Wireless Networking in the Developing World

A practical guide to planning and building low-cost telecommunications infrastructure

Wireless Networking in the Developing World

For more information about this project, visit us online at *http://wndw.net/*

First edition, January 2006

Contents

Networking Hardware

Security

Building an Outdoor Node

Troubleshooting

Case Studies

About This Book

This book is part of a set of related materials about the same topic: Wireless Networking in the Developing World. While not all materials are available at the time of first printing, these will include:

- Printed books
- A DRM-free PDF version of the book
- An archived mailing list for discussion of the concepts and techniques described in the book
- Additional case studies, training course material and related information

For all of this material and more, see our website at *http://wndw.net/*

 Printed copies may be ordered from Lulu.com, a print-on-demand service. Consult the website (*http://wndw.net/*) for details on ordering a printed copy. The PDF will be updated periodically, and ordering from the print-on-demand service ensures that you will always receive the latest revision.

The website will include additional case studies, currently available equipment, and more external website references. Volunteers and ideas are welcome. Please join the mailing list and send ideas.

The training course material was written for courses given by the Association for Progressive Communications and the Abdus Salam International Center for Theoretical Physics. See *http://www.apc.org/wireless/* and *http://wireless.ictp.trieste.it/* for more details on those courses and their material. Additional information was provided by the International Network for the Availability of Scientific Publications, *http://www.inasp.info/*. Some of this material has been incorporated directly into this book.

Credits

This book was started as the BookSprint project at the 2005 session of WSFII, in London, England (*http://www.wsfii.org/*). A core team of seven people built the initial outline over the course of the event, presented the results at the conference, and wrote the book over the course of a few months. Rob Flickenger served as the lead author and editor. Throughout the project, the core group has actively solicited contributions and feedback from the wireless networking community.

Core group

- **Corinna "Elektra" Aichele**. Elektra's main interests include autonomous power systems and wireless communication (antennas, wireless long shots, mesh networking). She made a small linux distro based on slackware geared to wireless mesh networking. This information is of course redundant if one reads the book... *http://www.scii.nl/~elektra*

- **Rob Flickenger** was the lead author, editor, and illustrator of this book. Rob has been writing professionally since 2002. He has written and edited several books, including *Building Wireless Community Networks* and *Wireless Hacks*, published by O'Reilly Media. He co-founded Metrix Communication LLC (*http://metrix.net/*), a wireless hardware company dedicated to open source software, open standards, and ubiquitous wireless networking. Prior to becoming an active member of SeattleWireless (*http://seattlewireless.net/*), he was a founding father of the NoCat project (*http://nocat.net/*).

  Rob's ultimate goal is the realization of Infinite Bandwidth Everywhere for Free. He publishes some of his adventures along the path toward realizing this goal at *http://constructiveinterference.net/*

- **Carlo Fonda** is a member of the Radio Communications Unit at the Abdus Salam International Center for Theoretical Physics in Trieste, Italy.

- **Jim Forster** has spent his career in software development, mostly working on operating systems and networking in product companies. He has experience with several failed startup companies in Silicon Valley, and one successful one, Cisco Systems. After a lot of product development work there, his more recent activities involve projects and policies for improving Internet access in developing countries. He can be reached at *jrforster@mac.com*.

- **Ian Howard.** After flying around the world for seven years as a paratrooper in the Canadian military, Ian Howard decided to trade his gun for a computer.

  After finishing a degree in environmental sciences at the University of Waterloo he wrote in a proposal, "Wireless technology has the opportunity to

bridge the digital divide. Poor nations, who do not have the infrastructure for interconnectivity as we do, will now be able to create a wireless infrastructure." As a reward, Geekcorps sent him to Mali as the Geekcorps Mali Program Manager, where he led a team equipping radio stations with wireless interconnections and designed content sharing systems.

He is now a consultant on various Geekcorps programs.

- **Tomas Krag** spends his days working with *wire.less.dk*, a registered non-profit, based in Copenhagen, which he founded with his friend and colleague Sebastian Büttrich in early 2002. wire.less.dk specialises in community wireless networking solutions, and has a special focus on low-cost wireless networks for the developing world.

  Tomas is also an associate of the Tactical Technology Collective *http://www.tacticaltech.org/*, an Amsterdam-based non-profit "to strengthen social technology movements and networks in developing and transition countries, as well as promote civil society's effective, conscious and creative use of new technologies." Currently most of his energy goes into the Wireless Roadshow (*http://www.thewirelessroadshow.org/*), a project that supports civil society partners in the developing world in planning, building and sustaining connectivity solutions based on license-exempt spectrum, open technology and open knowledge.

- **Marco Zennaro**, aka marcusgennaroz, is an electronic engineer working at the ICTP in Trieste, Italy. He has been using BBSes and ham radios since he was a teenager, and he is happy to have merged the two together working in the field of wireless networking. He still carries his Apple Newton.

In addition to the core group, several others have contributed their writing, feedback, editing, and other skills to make this project what it is.

Contributors

- **Sebastian Büttrich** (*http://wire.less.dk/*) is a generalist in technology with a background in scientific programming and physics. Originally from Berlin, Germany, he worked with IconMedialab in Copenhagen from 1997 until 2002. He holds a Ph.D. in quantum physics from the Technical University of Berlin. His physics background includes fields like RF and microwave spectroscopy, photovoltaic systems, and advanced maths.

  He is also a performing and recording musician.

- **Kyle Johnston,** *http://www.schoolnet.na/*

- **Adam Messer**. Originally trained as an insect scientist, Adam Messer metamorphosed into a telecommunications professional after a chance conversation in 1995 led him to start one of Africa's first ISPs. Pioneering wireless data services in Tanzania, Messer worked for 11 years in eastern

and southern Africa in voice and data communications for startups and multinational cellular carriers. He now resides in Amman, Jordan.

- **Ermanno Pietrosemoli** has been involved in planning and building computer networks for the last twenty years. As president of the Latin American Networking School, Escuela Latinoamericana de Redes “EsLaRed”, *www.eslared.org.ve*, he has been teaching wireless data communications in several countries while keeping his base at Mérida, Venezuela.
- **Dana Spiegel** is an independent software consultant and founder of sociableDESIGN (*www.sociableDESIGN.com*), a consulting firm that specializes in social software and wireless technologies. He serves as the Executive Director and a member of the Board of Directors of NYCwireless (*www.nycwireless.net*), a New York City non-profit organization that advocates and enables the growth of free, public wireless networks. He also writes the Wireless Community blog (*www.wirelesscommunity.info*).

Support

- **Lisa Chan** (*http://www.cowinanorange.com/*) was the lead copy editor.
- **Richard Lotz** (*http://greenbits.net/~rlotz/*) provided technical review and suggestions. He works on SeattleWireless projects and would like to take his node (and his house) off the grid.
- **Casey Halverson** (*http://seattlewireless.net/~casey/*) provided technical review and suggestions.
- **Catherine Sharp** (*http://odessablue.com/*) provided copy edit support.
- **Matt Westervelt** (*http://seattlewireless.net/~mattw/*) provided technical review and copy edit support. Matt is the founder of SeattleWireless (*http://seattlewireless.net/*) and an evangelist for FreeNetworks worldwide. He left the corporate world to start Metrix Communication LLC (*http://metrix.net/*), a company created to supply FreeNetworkers with high quality, standards-based wireless networking products. As a child, he watched a lot of Sesame Street and has a firm (perhaps misguided) belief that cooperation can solve a lot of the world's problems.

Special thanks

The core team would like to thank the organizers of WSFII for providing the space, support, and occasional bandwidth that served as the incubator for this project. We would especially like to thank community networkers everywhere, who devote so much of their time and energy towards fulfilling the promise of the global Internet. Without you, community networks could not exist.

1
Where to Begin

This book was created by a team of individuals who each, in their own field, are actively participating in the ever-expanding Internet by pushing its reach farther than ever before. The massive popularity of wireless networking has caused equipment costs to continually plummet, while equipment capabilities continue to sharply increase. We believe that by taking advantage of this state of affairs, people can finally begin to have a stake in building their own communications infrastructure. We hope to not only convince you that this is possible, but also show how we have done it, and to give you the information and tools you need to start a network project in your local community.

Wireless infrastructure can be built for very little cost compared to traditional wired alternatives. But building wireless networks is only partly about saving money. By providing people in your local community with cheaper and easier access to information, they will directly benefit from what the Internet has to offer. The time and effort saved by having access to the global network of information translates into wealth on a local scale, as more work can be done in less time and with less effort.

Likewise, the network becomes all the more valuable as more people are connected to it. Communities connected to the Internet at high speed have a voice in a global marketplace, where transactions happen around the world at the speed of light. People all over the world are finding that Internet access gives them a voice to discuss their problems, politics, and whatever else is important to their lives, in a way that the telephone and television simply cannot compete with. What has until recently sounded like science fiction is now becoming a reality, and that reality is being built on wireless networks.

But even without access to the Internet, wireless community networks have tremendous value. They allow people to collaborate on projects across wide distances. Voice communications, email, and other data can be exchanged

for very little cost. By involving local people in the construction of the network, knowledge and trust are spread throughout the community, and people begin to understand the importance of having a share in their communications infrastructure. Ultimately, they realize that communication networks are built to allow people to connect with each other.

In this book we will focus on wireless data networking technologies in the 802.11 family. While such a network can carry data, voice, and video (as well as traditional web and Internet traffic), the networks described in this book are data networks. We specifically do not cover GSM, CDMA, or other wireless voice technologies, since the cost of deploying these technologies is well beyond the reach of most community projects.

Purpose of this book

The overall goal of this book is to help you build affordable communication technology in your local community by making best use of whatever resources are available. Using inexpensive off-the-shelf equipment, you can build high speed data networks that connect remote areas together, provide broadband network access in areas that even dialup does not exist, and ultimately connect you and your neighbors to the global Internet. By using local sources for materials and fabricating parts yourself, you can build reliable network links with very little budget. And by working with your local community, you can build a telecommunications infrastructure that benefits everyone who participates in it.

This book is not a guide to configuring a radio card in your laptop or choosing consumer grade gear for your home network. The emphasis is on building infrastructure links intended to be used as the backbone for wide area wireless networks. With that goal in mind, information is presented from many points of view, including technical, social, and financial factors. The extensive collection of case studies present various groups' attempts at building these networks, the resources that were committed to them, and the ultimate results of these attempts.

Since the first spark gap experiments at the turn of the last century, wireless has been a rapidly evolving area of communications technology. While we provide specific examples of how to build working high speed data links, the techniques described in this book are not intended to replace existing wired infrastructure (such as telephone systems or fiber optic backbone). Rather, these techniques are intended to augment existing systems, and provide connectivity in areas where running fiber or other physical cable would be impractical.

We hope you find this book useful for solving your particular communication challenges.

Fitting wireless into your existing network

If you are a network administrator, you may wonder how wireless might fit into your existing network infrastructure. Wireless can serve in many capacities, from a simple extension (like a several kilometer Ethernet cable) to a distribution point (like a large hub). Here just a few examples of how your network can benefit from wireless technology.

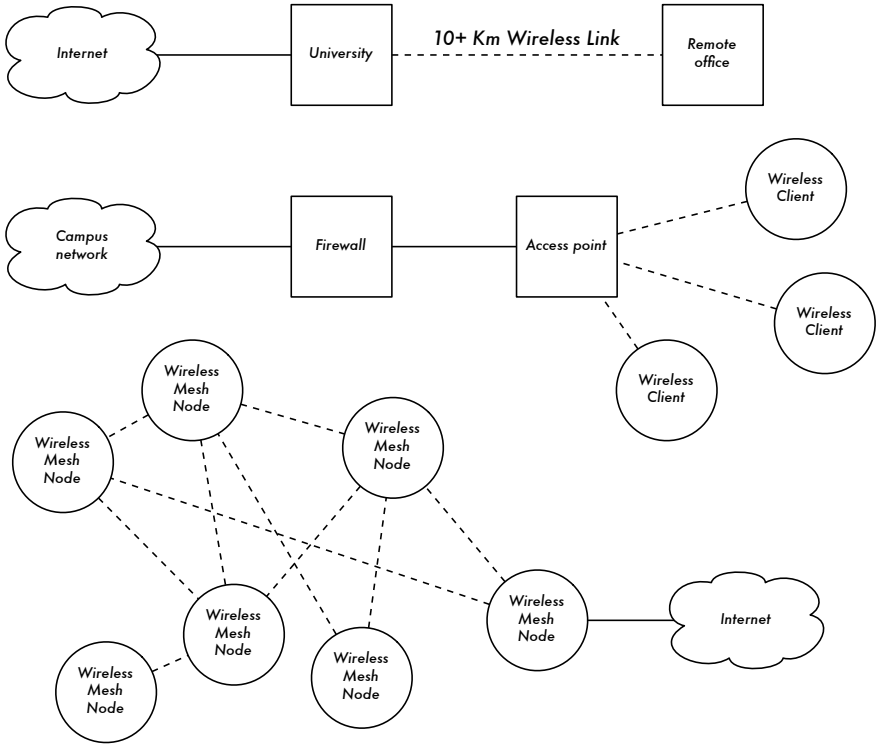

Figure 1.1: Some wireless networking examples.

Wireless networking protocols

The primary technology used for building low-cost wireless networks is currently the 802.11 family of protocols, also known in many circles as ***Wi-Fi***. The 802.11 family of radio protocols (802.11a, 802.11b, and 802.11g) have enjoyed an incredible popularity in the United States and Europe. By implementing a common set of protocols, manufacturers world wide have built highly interoperable equipment. This decision has proven to be a significant

boon to the industry and the consumer. Consumers are able to use equipment that implements 802.11 without fear of "vendor lock-in". As a result, consumers are able to purchase low-cost equipment at a volume which has benefitted manufacturers. If manufacturers had chosen to implement their own proprietary protocols, it is unlikely that wireless networking would be as inexpensive and ubiquitous as it is today.

While new protocols such as 802.16 (also known as WiMax) will likely solve some difficult problems currently observed with 802.11, they have a long way to go to match the popularity and price point of 802.11 equipment. As this equipment that supports WiMax is just becoming available at the time of this writing, we will focus primarily on the 802.11 family.

There are many protocols in the 802.11 family, and not all are directly related to the radio protocol itself. The three wireless standards currently implemented in most readily available gear are:

- **802.11b**. Ratified by the IEEE on September 16, 1999, 802.11b is probably the most popular wireless networking protocol in use today. Millions of devices supporting it have shipped since 1999. It uses a modulation called ***Direct Sequence Spread Spectrum*** (***DSSS***) in a portion of the ISM band from 2.412 to 2.484GHz. It has a maximum rate of 11Mbps, with actual usable data speeds up to about 5Mbps.
- **802.11g**. As it wasn't finalized until June 2003, 802.11g is a relative latecomer to the wireless marketplace. Despite the late start, 802.11g is now the de facto standard wireless networking protocol as it now ships as a standard feature on virtually all laptops and most handheld devices. 802.11g uses the same ISM range as 802.11b, but uses a modulation scheme called ***Orthogonal Frequency Division Multiplexing*** (***OFDM***). It has a maximum data rate of 54Mbps (with usable throughput of up to 25Mbps), and can fall back to 11Mbps DSSS or slower for backwards compatibility with the hugely popular 802.11b.
- **802.11a**. Also ratified by the IEEE on September 16, 1999, 802.11a uses OFDM. It has a maximum data rate of 54Mbps, with actual throughput of up to 27Mbps. 802.11a operates in the ISM band between 5.745 and 5.805GHz, and in a portion of the UNII band between 5.170 and 5.320GHz. This makes it incompatible with 802.11b or 802.11g, and the higher frequency means shorter range compared to 802.11b/g at the same power. While this portion of the spectrum is relatively unused compared to 2.4GHz, it is unfortunately only legal for use in a few parts of the world. Check with your local authorities before using 802.11a equipment, particularly in outdoor applications. 802.11a equipment is still quite inexpensive, but is not nearly as popular as 802.11b/g.

In addition to the above standards, there are a number of vendor-specific extensions to equipment, touting speeds of 108Mbps, stronger encryption, and increased range. Unfortunately these extensions will not operate between equipment from different manufacturers, and purchasing them will effectively lock you into that vendor for every part of your network. New equipment and standards (such as 802.11n, 802.16, MIMO, and WiMAX) promise significant increases in speed and reliability, but this equipment is just starting to ship at the time of this writing, and availability and vendor interoperability is unclear.

Due to the ubiquity of equipment, better range, and unlicensed nature of the 2.4GHz ISM band, this book will concentrate building networks using 802.11b and 802.11g.

Question & Answer

If you are new to wireless networking, you likely have a number of questions about what the technology can do and what it will cost. Here are some commonly asked questions, with answers and suggestions on the listed page.

Power

- How can I supply power to my radio equipment, if there is no power available? **Page 184.**
- Do I need to run a power cable all the way up the tower? **Page 180.**
- How can I use solar panel to power my wireless node while keeping it online overnight? **Page 184.**
- How long will my access point run on a battery? **Page 186.**

Management

- How can I monitor and manage remote access points from my office? **Page 165.**
- What do I do when the network breaks? **Page 166, 198.**
- What are the most common problems encountered on wireless networks, and how do I fix them? **Page 199.**

Distance

- How good is the range of my access point? **Page 51.**
- Is there any formula I can use to know how far I can go with a given access point? **Page 51.**

- How can I know if a remote place can be connected to Internet using a wireless link? **Page 58.**
- The manufacturer says my access point has a range of 300 meters. Is that true? **Page 51.**
- How can I provide wireless connectivity to many remote clients, spread all around the city? **Page 29.**
- Is it true that I can reach a much greater distance adding a tin can or aluminum foil to my AP's antenna ? **Page 101.**
- Can I use wireless to connect to a remote site and share a single central Internet connection? **Page 28.**
- My wireless link looks like it will be too long. Can I put a repeater in the middle to make it better? **Page 62.**
- Should I use an amplifier instead? **Page 60, 99.**

Installation

- How can I install my indoor AP on the top of a mast on my roof? **Page 171.**
- Is it really useful to add a lightning protector and proper grounding to my antenna mast, or can I go without them? **Page 131, 182.**
- Can I build an antenna mast by myself? How high can I go? **Page 181.**
- Why does my antenna work much better when I mount it “sideways”? **Page 93.**
- Which channel should I use? **Page 15.**
- Will radio waves travel through buildings and trees? What about people? **Page 17.**
- Will radio waves travel through a hill that is in the way? **Page 22.**
- How do I build a mesh network? **Page 41.**
- What kind of antenna is the best one for my network? **Page 94.**
- Can I build an access point using a recycled PC? **Page 133.**
- How can I install Linux on my AP? Why should I do so? **Page 142.**

Money

- How can I know if a wireless link is achievable with a limited amount of money? **Page 125.**
- Which is the best AP with the lowest price? **Page 123.**
- How can I track and bill customers for using my wireless network? **Page 153, 165.**

Partners and Customers

- If I am supplying connectivity, do I still need service from an ISP? Why? **Page 27.**
- How many customers do I need to cover my costs? **Page 228.**
- How many customers will my wireless network support? **Page 49.**
- How do I make my wireless network go faster? **Page 64.**
- Is my Internet connection as fast as it can be? **Page 74.**

Security

- How can I protect my wireless network from unauthorized access? **Page 150.**
- Is it true that a wireless network is always insecure and open to attacks by hackers? **Page 148.**
- Is it true that the use of open source software makes my network less secure? **Page 156.**
- How can I see what is happening on my network? **Page 164.**

Information and Licensing

- What other books should I read to improve my wireless networking skills? **Page 242.**
- Where can I find more information online? **Page 237.**
- Can I use parts of this book for my own teaching? Can I print and sell copies of this book? **Yes. See *About This Book* for more details.**

2
A Practical Introduction to Radio Physics

Wireless communications make use of electromagnetic waves to send signals across long distances. From a user's perspective, wireless connections are not particularly different from any other network connection: your web browser, email, and other applications all work as you would expect. But radio waves have some unexpected properties compared to Ethernet cable. For example, it's very easy to see the path that an Ethernet cable takes: locate the plug sticking out of your computer, follow the cable to the other end, and you've found it! You can also be confident that running many Ethernet cables alongside each other won't cause problems, since the cables effectively keep their signals contained within the wire itself.

But how do you know where the waves emanating from your wireless card are going? What happens when these waves bounce off of objects in the room or other buildings in an outdoor link? How can several wireless cards be used in the same area without interfering with each other?

In order to build stable high-speed wireless links, it is important to understand how radio waves behave in the real world.

What is a wave?

We are all familiar with vibrations or oscillations in various forms: a pendulum, a tree swaying in the wind, the string of a guitar - these are all examples of oscillations.

What they have in common is that something, some medium or object, is swinging in a periodic manner, with a certain number of cycles per unit of

time. This kind of wave is sometimes called a ***mechanical*** wave, since it is defined by the motion of an object or its propagating medium.

When such oscillations travel (that is, when the swinging does not stay bound to one place) then we speak of waves propagating in space. For example, a singer singing creates periodic oscillations in his or her vocal cords. These oscillations periodically compress and decompress the air, and this periodic change of air pressure then leaves the singers mouth and travels, at the speed of sound. A stone plunging into a lake causes a disturbance, which then travels across the lake as a ***wave***.

A wave has a certain ***speed***, ***frequency***, and ***wavelength***. These are connected by a simple relation:

```
Speed = Frequency * Wavelength
```

The wavelength (sometimes referred to as **lambda**, **λ**) is the distance measured from a point on one wave to the equivalent part of the next, for example from the top of one peak to the next. The frequency is the number of whole waves that pass a fixed point in a period of time. Speed is measured in meters/second, frequency is measured in cycles per second (or Hertz, abbreviated ***Hz***), and wavelength is measured in meters.

For example, if a wave on water travels at one meter per second, and it oscillates five times per second, then each wave will be twenty centimeters long:

```
1 meter/second = 5 cycles/second * W
W = 1 / 5 meters
W = 0.2 meters = 20 cm
```

Waves also have a property called ***amplitude***. This is the distance from the center of the wave to the extreme of one of its peaks, and can be thought of as the “height” of a water wave.. The relationship between frequency, wavelength, and amplitude are shown in Figure 2.1.

Waves in water are easy to visualize. Simply drop a stone into the lake and you can see the waves as they move across the water over time. In the case of electromagnetic waves, the part that might be hardest to understand is: “What is it that is oscillating?”

In order to understand that, we need to understand electromagnetic forces.

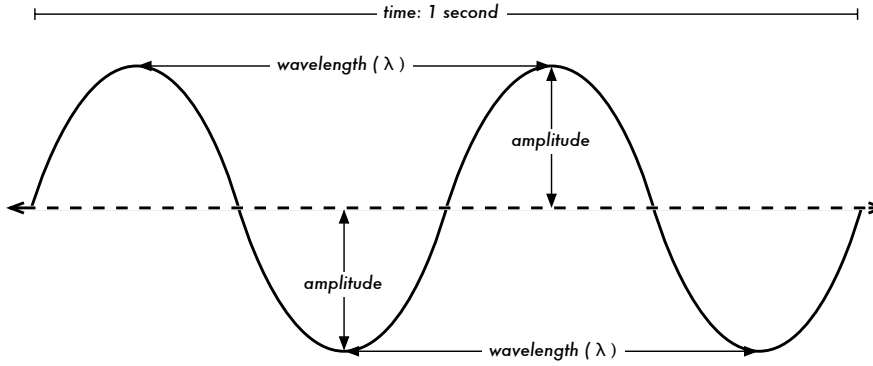

Figure 2.1: Wavelength, amplitude, and frequency. For this wave, the frequency is 2 cycles per second, or 2 Hz.

Electromagnetic forces

Electromagnetic forces are the forces between electrical charges and currents. Our most direct access to those is when our hand touches a door handle after walking on synthetic carpet, or brushing up against an electrical fence. A more powerful example of electromagnetic forces is the lightning we see during thunderstorms. The ***electrical force*** is the force between electrical charges. The ***magnetic force*** is the force between electrical currents.

Electrons are particles that carry a negative electrical charge. There are other particles too, but electrons are responsible for most of what we need to know about how radio behaves.

Let us look at what is happening in a piece of straight wire, in which we push the electrons from one and to the other and back, periodically. At one moment, the top of the wire is negatively charged - all the negative electrons are gathered there. This creates an electric field from plus to minus along the wire. The next moment, the electrons have all been driven to the other side, and the electric field points the other way. As this happens again and again, the electric field vectors (arrows from plus to minus) are leaving the wire, so to speak, and are radiated out into the space around the wire.

What we have just described is known as a dipole (because of the two poles, plus and minus), or more commonly a ***dipole antenna***. This is the simplest form of omnidirectional antenna. The motion of the electric field is commonly referred to as an ***electromagnetic wave***.

Let us come back to the relation:

```
Speed = Frequency * Wavelength
```

In the case of electromagnetic waves, the speed is **c**, the speed of light.

$$c = 300{,}000 \text{ km/s} = 300{,}000{,}000 \text{ m/s} = 3*10^8 \text{ m/s}$$
$$c = f * \lambda$$

Electromagnetic waves differ from mechanical waves in that they require no medium in which to propagate. Electromagnetic waves will even propagate through the vacuum of space.

Powers of ten

In physics, math, and engineering, we often express numbers by powers of ten. We will meet these terms again, e.g. in Giga-Hertz (GHz), Centi-meters (cm), Micro-seconds (μs), and so on.

Powers of Ten			
Nano-	10^{-9}	1/1000000000	n
Micro-	10^{-6}	1/1000000	μ
Milli-	10^{-3}	1/1000	m
Centi-	10^{-2}	1/100	c
Kilo-	10^{3}	1 000	k
Mega-	10^{6}	1 000 000	M
Giga-	10^{9}	1 000 000 000	G

Knowing the speed of light, we can calculate the wavelength for a given frequency. Let us take the example of the frequency of 802.11b wireless networking, which is

$$f = 2.4 \text{ GHz}$$
$$= 2{,}400{,}000{,}000 \text{ cycles / second}$$

$$\text{wavelength lambda } (\lambda) = c / f$$
$$= 3*10^8 / 2.4*10^9$$
$$= 1.25*10^{-1} \text{ m}$$
$$= 12.5 \text{ cm}$$

Frequency and wavelength determine most of an electromagnetic wave's behaviour, from antennas that we build to objects that are in the way of the networks we intend to run. They are responsible for many of the differences

between different standards we might be choosing. Therefore, an understanding of the basic ideas of frequency and wavelength helps a lot in practical wireless work.

Polarization

Another important quality of electromagnetic waves is ***polarization***. Polarization describes the direction of the electrical field vector.

If you imagine a vertically aligned dipole antenna (the straight piece of wire), electrons only move up and down, not sideways (because there is no room to move) and thus electrical fields only ever point up or down, vertically. The field leaving the wire and traveling as a wave has a strict linear (and in this case, vertical) polarization. If we put the antenna flat on the ground (horizontal, we would find horizontal linear polarization.

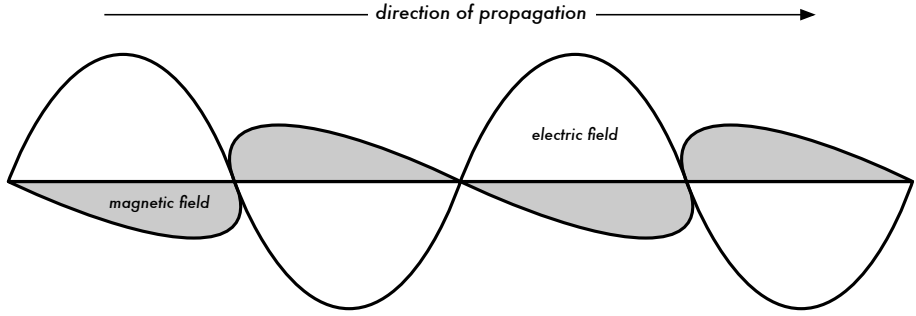

Figure 2.2: Electric field and complementary magnetic field components of an electromagnetic wave. Polarization describes the orientation of the electric field.

Linear polarization is just one special case, and is never quite so perfect: in general, we will always have some component of the field pointing other directions too. The most general case is elliptic polarization, with the extremes of linear (only one direction) and circular polarizations (both directions at equal strength).

As one can imagine, polarization becomes important when aligning antennas. If you ignore polarization, you might have very little signal even though you have the strongest antennas. We call this polarization mismatch.

The electromagnetic spectrum

Electromagnetic waves span a wide range of frequencies (and, accordingly, wavelengths). This range of frequencies and wavelengths is called the ***electromagnetic spectrum***. The part of the spectrum most familiar to humans is probably light, the visible portion of the electromagnetic spectrum.

Light lies roughly between the frequencies of $7.5*10^{14}$ Hz and $3.8*10^{14}$ Hz, corresponding to wavelengths from circa 400 nm (violet/blue) to 800 nm (red).

We are also regularly exposed to other regions of the electromagnetic spectrum, including ***AC*** (Alternating Current) or grid electricity, at 50/60 Hz, X-Rays / Roentgen radiation, Ultraviolet (on the higher frequencies side of visible light), Infrared (on the lower frequencies side of visible light) and many others. ***Radio*** is the term used for the portion of the electromagnetic spectrum in which waves can be generated by applying alternating current to an antenna. This is true for the range from 3 Hz to 300 GHz, but in the more narrow sense of the term, the upper frequency limit would be 1 GHz.

When talking about radio, many people think of FM radio, which uses a frequency around 100 MHz. In between radio and infrared we find the region of microwaves - with frequencies from about 1 GHz to 300 GHz, and wavelengths from 30 cm to 1 mm.

The most popular use of microwaves might be the microwave oven, which in fact works in exactly the same region as the wireless standards we are dealing with. These regions lie within the bands that are being kept open for general unlicensed use. This region is called the ***ISM band***, which stands for Industrial, Scientific, and Medical. Most other parts of the electromagnetic spectrum are tightly controlled by licensing legislation, with license values being a huge economic factor. This goes especially for those parts of the spectrum that are suitable for broadcast (TV, radio) as well as voice and data communication. In most countries, the ISM bands have been reserved for unlicensed use.

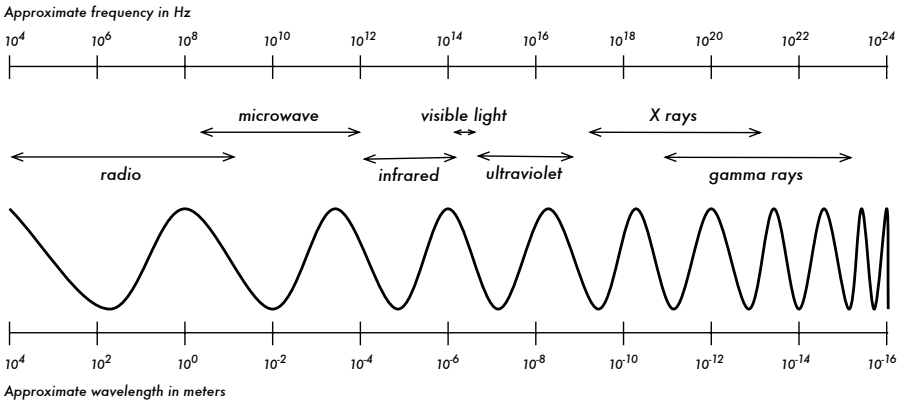

Figure 2.3: The electromagnetic spectrum.

The frequencies most interesting to us are 2.412 - 2.484 GHz, which is used by the 802.11b and 802.11g radio standards (corresponding to wavelengths

of about 12.5 cm). Other commonly available equipment uses the 802.11a standard, which operates at 5.170 - 5.805 GHz (corresponding to wavelengths of about 5 to 6 cm).

Bandwidth

A term you will meet often in radio physics is ***bandwidth***. Bandwidth is simply a measure of frequency range. If a range of 2.40 GHz to 2.48 GHz is used by a device, then the bandwidth would be 0.08 GHz (or more commonly stated as 80MHz).

It is easy to see that the bandwidth we define here is closely related to the amount of data you can transmit within it - the more room in frequency space, the more data you can fit in at a given moment. The term bandwidth is often used for something we should rather call a data rate, as in “my Internet connection has 1 Mbps of bandwidth”, meaning it can transmit data at 1 megabit per second.

Frequencies and channels

Let us look a bit closer at how the 2.4GHz band is used in 802.11b. The spectrum is divided into evenly sized pieces distributed over the band as individual ***channels***. Note that channels are 22MHz wide, but are only separated by 5MHz. This means that adjacent channels overlap, and can interfere with each other. This is represented visually in Figure 2.4.

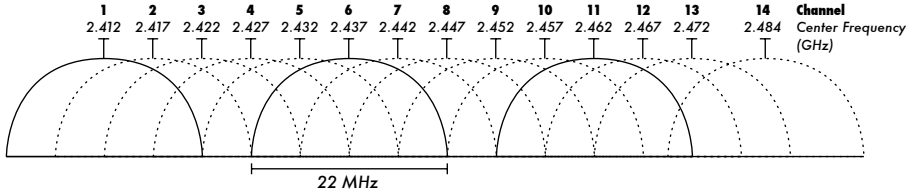

Figure 2.4: Channels and center frequencies for 802.11b. Note that channels 1, 6, and 11 do not overlap.

For a complete list of channels and their center frequencies for 802.11b/g and 802.11a, see Appendix A.

Behaviour of radio waves

There are a few simple rules of thumb that can prove extremely useful when making first plans for a wireless network:

- The longer the wavelength, the further it goes

- The longer the wavelength, the better it travels through and around things
- The shorter the wavelength, the more data it can transport

All of these rules, simplified as they may be, are rather easy to understand by example.

Longer waves travel further

Assuming equal power levels, waves with longer wavelengths tend to travel further than waves with shorter wavelengths. This effect is often seen in FM radio, when comparing the range of an FM transmitter at 88MHz to the range at 108MHz. Lower frequency transmitters tend to reach much greater distances than high frequency transmitters at the same power.

Longer waves pass around obstacles

A wave on water which is 5 meters long will not be stopped by a 5 mm piece of wood sticking out of the water. If instead the piece of wood were 50 meters big (e.g. a ship), it would be well in the way of the wave. The distance a wave can travel depends on the relationship between the wavelength of the wave and the size of obstacles in its path of propagation.

It is harder to visualize waves moving “through” solid objects, but this is the case with electromagnetic waves. Longer wavelength (and therefore lower frequency) waves tend to penetrate objects better than shorter wavelength (and therefore higher frequency) waves. For example, FM radio (88-108MHz) can travel through buildings and other obstacles easily, while shorter waves (such as GSM phones operating at 900MHz or 1800MHz) have a harder time penetrating buildings. This effect is partly due to the difference in power levels used for FM radio and GSM, but is also partly due to the shorter wavelength of GSM signals.

Shorter waves can carry more data

The faster the wave swings or beats, the more information it can carry - every beat or cycle could for example be used to transport a digital bit, a '0' or a '1', a 'yes' or a 'no'.

There is another principle that can be applied to all kinds of waves, and which is extremely useful for understanding radio wave propagation. This principle is known as the ***Huygens Principle***, named after Christiaan Huygens, Dutch mathematician, physicist and astronomer 1629 - 1695.

Imagine you are taking a little stick and dipping it vertically into a still lake's surface, causing the water to swing and dance. Waves will leave the center

of the stick - the place where you dip in - in circles. Now, wherever water particles are swinging and dancing, they will cause their neighbour particles to do the same: from every point of disturbance, a new circular wave will start. This is, in simple form, the Huygens principle. In the words of *wikipedia.org*:

> *"The Huygens' principle is a method of analysis applied to problems of wave propagation in the far field limit. It recognizes that each point of an advancing wave front is in fact the center of a fresh disturbance and the source of a new train of waves; and that the advancing wave as a whole may be regarded as the sum of all the secondary waves arising from points in the medium already traversed. This view of wave propagation helps better understand a variety of wave phenomena, such as diffraction."*

This principle holds true for radio waves as well as waves on water, for sound as well as light - only for light the wavelength is far too short for human beings to actually see the effects directly.

This principle will help us to understand diffraction as well as Fresnel zones, the need for line of sight as well as the fact that sometimes we seem to be able to go around corners, with no line of sight.

Let us now look into what happens to electromagnetic waves as they travel.

Absorption

When electromagnetic waves go through 'something' (some material), they generally get weakened or dampened. How much they lose in power will depend on their frequency and of course the material. Clear window glass is obviously transparent for light, while the glass used in sunglasses filer out quite a share of the light intensity and also the ultraviolet radiation.

Often, an absorption coefficient is used to describe a material's impact on radiation. For microwaves, the two main absorbent materials are:

- **Metal**. Electrons can move freely in metals, and are readily able to swing and thus absorb the energy of a passing wave.
- **Water**. Microwaves cause water molecules to jostle around, thus taking away some of the wave's energy[1].

---

1. A commonly held myth is that water "resonates" at 2.4GHz, which is why that frequency is used in microwave ovens. Actually, water doesn't appear to have any particular "resonant" frequency. Water spins and jostles around near radio, and will heat when in the presence of high power radio waves at just about any frequency. 2.4GHz is an unlicensed ISM frequency, and so was a good political choice for use in microwave ovens.

For the purpose of practical wireless networking, we may well consider metal and water perfect absorbers: we will not be able to go through them (although thin layers of water will let some power pass). They are to microwave what a brick wall is to light. When talking about water, we have to remember that it comes in different forms: rain, fog and mist, low clouds and so forth all will be in the way of radio links. They have a strong influence, and in many circumstances a change in weather can bring a radio link down.

There are other materials that have a more complex effect on radio absorption.

For **trees** and **wood**, the amount of absorption depends on how much water they contain. Old dead dry wood is more or less transparent, wet fresh wood will absorb a lot.

Plastics and similar materials generally do not absorb a lot of radio energy- but this varies depending on the frequency and type of material. Before you build a component from plastic (e.g. weather protection for a radio device and its antennas), it is always a good idea to measure and verify that the material does not absorb radio energy around 2.4GHz. One simple method of measuring the absorption of plastic at 2.4GHz is to put a sample in a microwave oven for a couple of minutes. If the plastic heats up, then it absorbs radio energy and should not be used for weatherproofing.

Lastly, let us talk about ourselves: humans (as well as other animals) are largely made out of water. As far as radio networking is concerned, we may well be described as big bags of water, with the same strong absorption. Orienting an office access point in such a way that its signal must pass through many people is a key mistake when building office networks. The same goes for hotspots, cafe installations, libraries, and outdoor installations.

Reflection

Just like visible light, radio waves are reflected when they come in contact with materials that are suited for that: for radio waves, the main sources of reflection are metal and water surfaces. The rules for reflection are quite simple: the angle at which a wave hits a surface is the same angle at which it gets deflected. Note that in the eyes of a radio wave, a dense grid of bars acts just the same as a solid surface, as long as the distance between bars is small compared to the wavelength. At 2.4GHz, a one cm metal grid will act much the same as a metal plate.

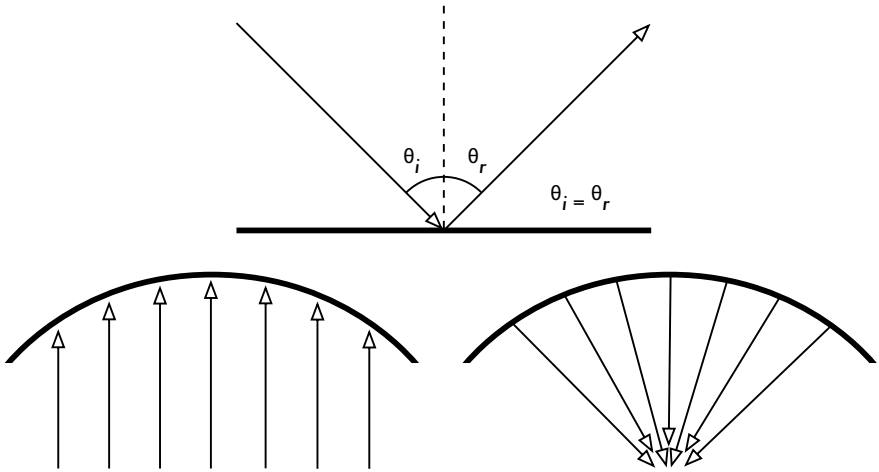

Figure 2.5: Reflection of radio waves. The angle of incidence is always equal to the angle of reflection. A parabolic uses this effect to concentrate radio waves spread out over its surface in a common direction.

Although the rules of reflection are quite simple, things can become very complicated when you imagine an office interior with many many small metal objects of various complicated shapes. The same goes for urban situations: look around you in city environment and try to spot all of the metal objects. This explains why ***multipath effects*** (i.e. signal reaching their target along different paths, and therefore at different times) play such an important role in wireless networking. Water surfaces, with waves and ripples changing all the time, effectively make for a very complicated reflection object which is more or less impossible to calculate and predict precisely.

We should also add that polarization has an impact: waves of different polarization in general will be reflected differently.

We use reflection to our advantage in antenna building: e.g. we put huge parabolas behind our radio transmitter/receiver to collect and bundle the radio signal into a fine point.

Diffraction

Diffraction is the apparent bending of waves when hitting an object. It is the effect of “waves going around corners”.

Imagine a wave on water traveling in a straight wave front, just like a wave that we see rolling onto an ocean beach. Now we put a solid barrier, say a wooden solid fence, in its way to block it. We cut a narrow slit opening into that wall, like a small door. From this opening, a circular wave will start, and it will of course reach points that are not in a direct line behind this opening, but

also on either side of it. If you look at this wavefront - and it might just as well be an electromagnetic wave - as a beam (a straight line), it would be hard to explain how it can reach points that should be hidden by a barrier. When modeled as a wavefront, the phenomenon makes sense.

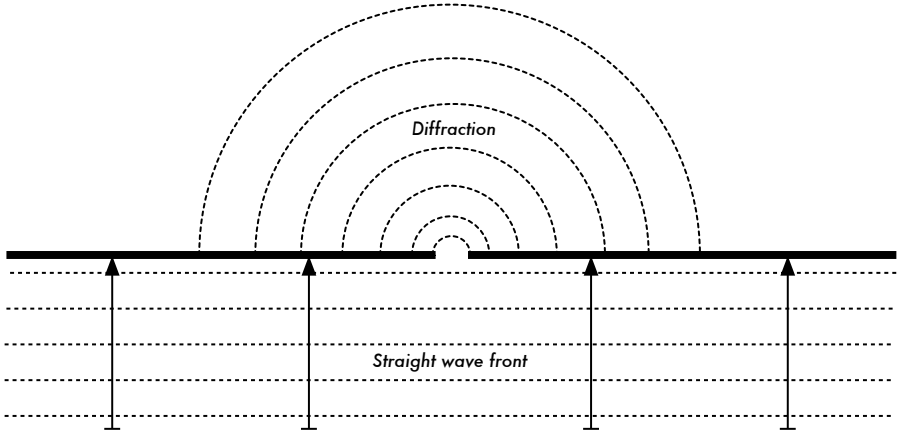

Figure 2.6: Diffraction through a narrow slit.

The Huygens Principle provides one model for understanding this behavior. Imagine that at any given instant, every point on a wavefront can be considered the starting point for a spherical "wavelet". This idea was later extended by Fresnel, and whether it adequately describes the phenomenon is still a matter of debate. But for our purposes, the Huygens model describes the effect quite well.

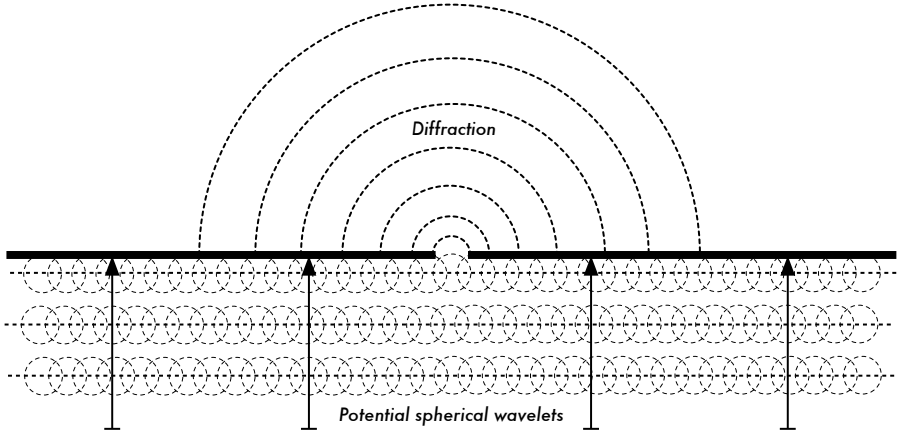

Figure 2.7: The Huygens Principle.

Through means of the effect of diffraction, waves will "bend" around corners or through an opening in a barrier. The wavelengths of visible light are far too small for humans to observe this effect directly. Microwaves, with a wave-

length of several centimeters, will show the effects of diffraction when waves hit walls, mountain peaks, and other obstacles. It seems as if the obstruction causes the wave to change its direction and go around corners.

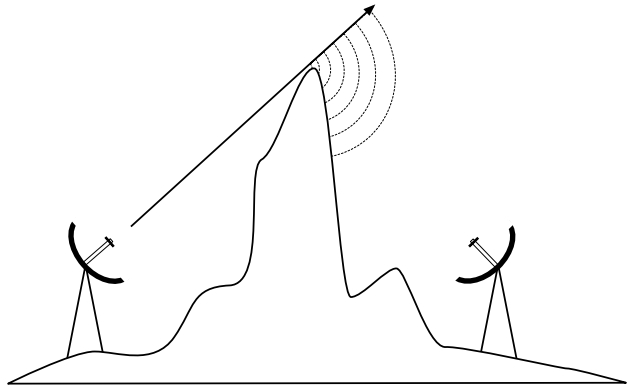

Figure 2.8: Diffraction over a mountain top.

Note that diffraction comes at the cost of power: the energy of the diffracted wave is significantly less than that of the wavefront that caused it. But in some very specific applications, you can take advantage of the diffraction effect to circumvent obstacles.

Interference

When working with waves, one plus one does not necessarily equal two. It can also result in zero.

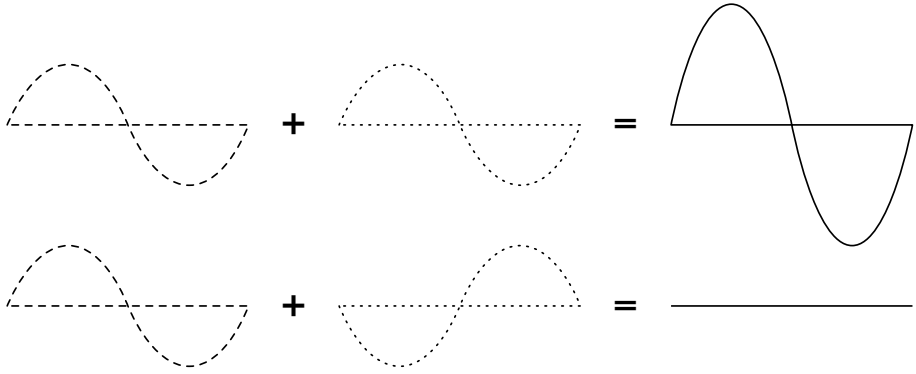

Figure 2.9: Constructive and destructive interference.

This is easy to understand when you draw two sine waves and add up the amplitudes. When peak hits peak, you will have maximum results (1 + 1 = 2). This is called ***constructive interference***. When peak hits valley, you will have complete annihilation ((1 + (-)1 = 0) - ***destructive interference***.

You can actually try this with waves on water and two little sticks to create circular waves - you will see that where two waves cross, there will be areas of higher wave peaks and others that remain almost flat and calm.

In order for whole trains of waves to add up or cancel each other out perfectly, they would have to have the exact same wavelength and a fixed phase relation, this means fixed positions from the peaks of the one wave to the other's.

In wireless technology, the word Interference is typically used in a wider sense, for disturbance through other RF sources, e.g. neighbouring channels. So, when wireless networkers talk about interference they typically talk about all kinds of disturbance by other networks, and other sources of microwave. Interference is one of the main sources of difficulty in building wireless links, especially in urban environments or closed spaces (such as a conference space) where many networks may compete for use of the spectrum.

Whenever waves of equal amplitude and opposite phase cross paths, the wave is annihilated and no signal can be received. The much more common case is that waves will combine to form a completely garbled waveform that cannot be effectively used for communication. The modulation techniques and use of multiple channels help to deal with the problem of interference, but does not completely eliminate it.

Line of sight

The term ***line of sight***, often abbreviated as ***LOS***, is quite easy to understand when talking about visible light: if we can see a point B from point A where we are, we have line of sight. Simply draw a line from A to B, and if nothing is in the way, we have line of sight.

Things get a bit more complicated when we are dealing with microwaves. Remember that most propagation characteristics of electromagnetic waves scale with their wavelength. This is also the case for the widening of waves as they travel. Light has a wavelength of about 0.5 micrometers, microwaves as used in wireless networking have a wavelength of a few centimeters. Consequently, their beams are a lot wider - they need more space, so to speak.

Note that visible light beams widen just the same, and if you let them travel long enough, you can see the results despite of their short wavelength. When pointing a well focussed laser at the moon, its beam will widen to well over 100 meters in radius by the time it reaches the surface. You can see this effect for yourself using an inexpensive laser pointer and a pair of binoculars

on a clear night. Rather than pointing at the moon, point at a distant mountain or unoccupied structure (such as a water tower). The radius of your beam will increase as the distance increases.

The line of sight that we need in order to have an optimal wireless connection from A to B is more than just a thin line - its shape is more like that of a cigar, an ellipse. Its width can be described by the concept of Fresnel zones.

Understanding the Fresnel zone

The exact theory of Fresnel (pronounced “Fray-nell”) zones is quite complicated. However, the concept is quite easy to understand: we know from the Huygens principle that at each point of a wavefront new circular waves start, We know that microwave beams widen. We know that waves of one frequency can interfere with each other. Fresnel zone theory simply looks at a line from A to B, and then at the space around that line that contributes to what is arriving at point B. Some waves travel directly from A to B, while others travel on paths off axis. Consequently, their path is longer, introducing a phase shift between the direct and indirect beam. Whenever the phase shift is one full wavelength, you get constructive interference: the signals add up optimally. Taking this approach and calculating accordingly, you find there are ring zones around the direct line A to B which contribute to the signal arriving at point B.

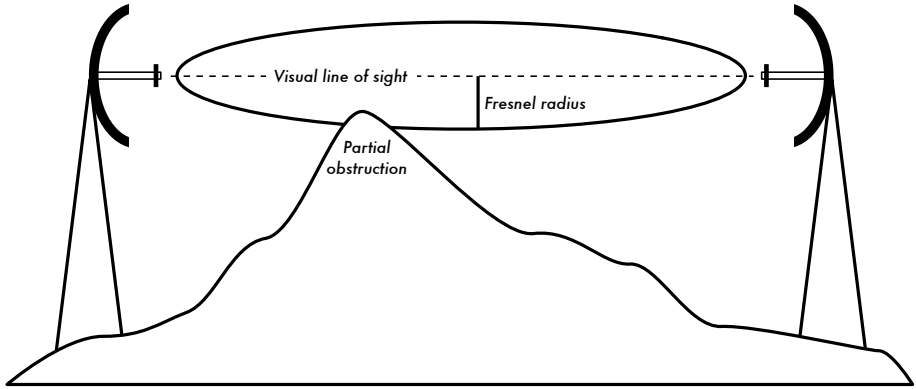

Figure 2.10: The Fresnel zone is partially blocked on this link, although the visual line of sight appears clear.

Note that there are many possible Fresnel zones, but we are chiefly concerned with zone 1. If this area were blocked by an obstruction, e.g. a tree or a building, the signal arriving at the far end would be diminished. When building wireless links, we therefore need to be sure that these zones be kept free of obstructions. Of course, nothing is ever perfect, so usually in wireless networking we should check that that the area containing about 60 percent of the first Fresnel zone should be kept free.

Here is one formula for calculating the first Fresnel zone:

```
r = 17.31 * sqrt(N(d1*d2)/(f*d))
```

...where **r** is the radius of the zone in meters, **N** is the zone to calculate, **d1** and **d2** are distances from obstacle to the link end points in meters, **d** is the total link distance in meters, and **f** is the frequency in MHz. Note that this gives you the radius of the zone. To calculate the height above ground, you need to subtract the result from a line drawn directly between the tops of the two towers.

For example, let's calculate the size if the first Fresnel zone in the middle of a 2km link, transmitting at 2.437GHz (802.11b channel 6):

```
r = 17.31 sqrt(1 * (1000 * 1000) / (2437 * 2000))
r = 17.31 sqrt(1000000 / 4874000)
r = 7.84 meters
```

Assuming both of our towers were ten meters tall, the first Fresnel zone would pass just 2.16 meters above ground level in the middle of the link. But how tall could a structure at that point be to clear 60% of the first zone?

```
r = 17.31 sqrt(0.6 *(1000 * 1000) / (2437 * 2000))
r = 17.31 sqrt(600000 / 4874000)
r = 6.07 meters
```

Subtracting the result from 10 meters, we can see that a structure 3.93 meters tall at the center of the link would block up to 60% of the first Fresnel zone. To improve the situation, we would need to position our antennas higher up, or change the direction of the link to avoid the obstacle.

Power

Any electromagnetic wave carries energy, or power - we can feel that when we enjoy (or suffer from) the warmth of the sun. The power **P** is of key importance for making wireless links work: you need a certain minimum power in order for a receiver to make sense of the signal.

We will come back to details of transmission power, losses, gains and radio sensitivity in chapter three. Here we will briefly discuss how the power P is defined and measured.

The electric field is measured in V/m (potential difference per meter), the power contained within it is proportional to the square of the electric field

$$P \sim E^2$$

Practically, we measure the power by means of some form of receiver, e.g. an antenna and a voltmeter, power meter, oscilloscope, or even a radio card and laptop. Looking at the signal's power directly means looking at the square of the signal in Volts.

Calculating with dBs

By far the most important technique when calculating power is calculating with ***decibels*** (***dB***). There is no new physics hidden in this - it is just a convenient method which makes calculations a lot simpler.

The decibel is a dimensionless unit[2], that is, it defines a relationship between two measurements of power. It is defined by:

```
dB = 10 * Log (P1 / P0)
```

where **P1** and **P0** can be whatever two values you want to compare. Typically, in our case, this will be some amount of power.

Why are decibels so handy to use? Many phenomena in nature happen to behave in a way we call exponential. For example, the human ear senses a sound to be twice as loud as another one if it has ten times the physical signal.

Another example, quite close to our field of interest, is absorption. Suppose a wall is in the path of our wireless link, and each meter of wall takes away half of the available signal. The result would be:

```
0 meters    =     1 (full signal)
1 meter     =     1/2
2 meters    =     1/4
3 meters    =     1/8
4 meters    =     1/16
n meters    =     1/2^n   = 2^-n
```

This is exponential behaviour.

But once we have used the trick of applying the logarithm (log), things become a lot easier: instead of taking a value to the n-th power, we just multiply by n. Instead of multiplying values, we just add.

---

2. Another example of a dimensionless unit is the percent (%) which can also be used in all kinds of quantities or numbers. While measurements like feet and grams are fixed, dimensionless units represent a relationship.

Here are some commonly used values that are important to remember:

```
 +3 dB = double power
 -3 dB = half the power
+10 dB = order of magnitude (10 times power)
-10 dB = one tenth power
```

In addition to dimensionless dBs, there are a number of relative definitions that are based on a certain base value P0. The most relevant ones for us are:

```
dBm   relative to P0 = 1 mW
dBi   relative to an ideal isotropic antenna
```

An ***isotropic antenna*** is a hypothetical antenna that evenly distributes power in all directions. It is approximated by a dipole, but a perfect isotropic antenna cannot be built in reality. The isotropic model is useful for describing the relative power gain of a real world antenna.

Another common (although less convenient) convention for expressing power is in ***milliwatts***. Here are equivalent power levels expressed in milliwatts and dBm:

```
  1 mW      = 0 dBm
  2 mW      = 3 dBm
100 mW      = 20 dBm
  1 W       = 30 dBm
```

Physics in the real world

Don't worry if the concepts in this chapter seem challenging. Understanding how radio waves propagate and interact with the environment is a complex field of study in itself. Most people find it difficult to understand phenomenon that they can't even see with their own eyes. By now you should understand that radio waves don't travel in a straight, predictable path. To make reliable communication networks, you will need to be able to calculate how much power is needed to cross a given distance, and predict how the waves will travel along the way.

There is much more to learn about radio physics than we have room for here. For more information about this evolving field, see the resources list in Appendix A. Now that you have an idea of how to predict how radio waves will interact in the real world, you are ready to start using them for communications.

3
Network Design

Before purchasing equipment or deciding on a hardware platform, you should have a clear idea of the nature of your communications problem. Most likely, you are reading this book because you need to connect computer networks together in order to share resources and ultimately reach the larger global Internet. The network design you choose to implement should fit the communications problem you are trying to solve. Do you need to connect a remote site to an Internet connection in the center of your campus? Will your network likely grow to include several remote sites? Will most of your network components be installed in fixed locations, or will your network expand to include hundreds of roaming laptops and other devices?

When solving a complex problem, it is often useful to draw a picture of your resources and problems. In this chapter, we will look at how other people have built wireless networks to solve their communication problems, including diagrams of the essential network structure. We will then cover the networking concepts that define TCP/IP, the primary networking language currently spoken on the Internet. We will then demonstrate several common methods for getting your information to flow efficiently through your network and on to the rest of the world.

Designing the physical network

It may seem odd to talk about the "physical" network when building wireless networks. After all, where is the physical part of the network? In wireless networks, the physical medium we use for communication is obviously electromagnetic energy. But in the context of this chapter, the physical network refers to the mundane topic of where to put things. How do you arrange the equipment so that you can reach your wireless clients? Whether

they fill an office building or stretch across many miles, wireless networks are naturally arranged in these three logical configurations:

- Point-to-point links
- Point-to-multipoint links
- Multipoint-to-multipoint clouds

The physical network layout you choose will depend on the nature of the problem you are trying to solve. While different parts of your network can take advantage of all three of these configurations, any individual link will fall into one of the above topologies. The application of each of these topologies is best described by example.

Point-to-point

Point-to-point links typically provide an Internet connection where such access isn't otherwise available. One side of a point-to-point link will have an Internet connection, while the other uses the link to reach the Internet. For example, a university may have a fast frame relay or VSAT connection in the middle of campus, but cannot afford such a connection for an important building just off campus. If the main building has an unobstructed view of the remote site, a point-to-point connection can be used to link the two together. This can augment or even replace existing dial-up links. With proper antennas and clear line of sight, reliable point-to-point links in excess of thirty kilometers are possible.

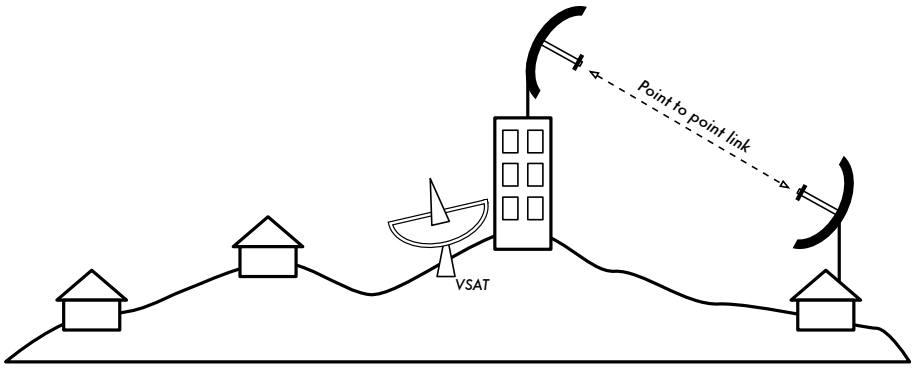

Figure 3.1: A point-to-point link allows a remote site to share a central Internet connection.

Of course, once a single point-to-point connection has been made, more can be used to extend the network even further. If the remote building in our

example is at the top of a tall hill, it may be able to see other important locations that can't be seen directly from the central campus. By installing another point-to-point link at the remote site, another node can join the network and make use of the central Internet connection.

Point-to-point links don't necessarily have to involve Internet access. Suppose you have to physically drive to a remote weather monitoring station, high in the hills, in order to collect the data which it records over time. You could connect the site with a point-to-point link, allowing data collection and monitoring to happen in realtime, without the need to actually travel to the site. Wireless networks can provide enough bandwidth to carry large amounts of data (including audio and video) between any two points that have a connection to each other, even if there is no direct connection to the Internet.

Point-to-multipoint

The next most commonly encountered network layout is ***point-to-multipoint***. Whenever several nodes[1] are talking to a central point of access, this is a point-to-multipoint application. The typical example of a point-to-multipoint layout is the use of a wireless access point that provides a connection to several laptops. The laptops do not communicate with each other directly, but must be in range of the access point in order to use the network.

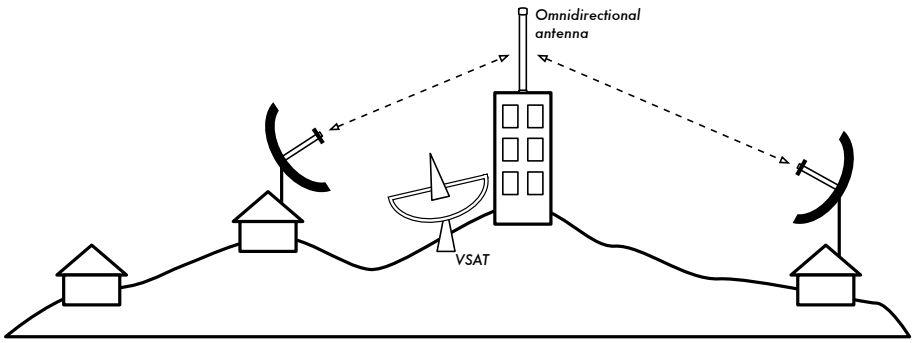

Figure 3.2: The central VSAT is now shared by multiple remote sites. All three sites can also communicate directly at speeds much faster than VSAT.

Point-to-multipoint networking can also apply to our earlier example at the university. Suppose the remote building on top of the hill is connected to the central campus with a point-to-point link. Rather than setting up several point-to-point links to distribute the Internet connection, a single antenna

1. A ***node*** is any device capable of sending and receiving data on a network. Access points, routers, computers, and laptops are all examples of nodes.

could be used that is visible from several remote buildings. This is a classic example of a wide area ***point*** (remote site on the hill) ***to multipoint*** (many buildings in the valley below) connection.

Note that there are a number of performance issues with using point-to-multipoint over very long distance, which will be addressed later in this chapter. Such links are possible and useful in many circumstances, but don't make the classic mistake of installing a single high powered radio tower in the middle of town and expecting to be able to serve thousands of clients, as you would with an FM radio station. As we will see, data networks behave very differently than broadcast radio.

Multipoint-to-multipoint

The third type of network layout is ***multipoint-to-multipoint***, which is also referred to as an ***ad-hoc*** or ***mesh*** network. In a multipoint-to-multipoint network, there is no central authority. Every node on the network carries the traffic of every other as needed, and all nodes communicate with each other directly.

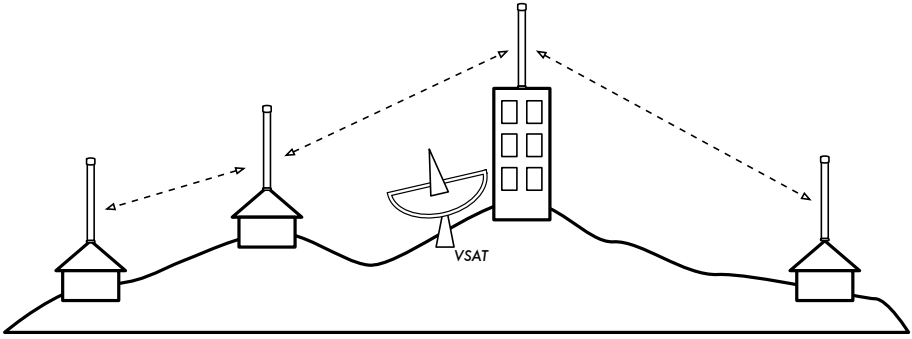

Figure 3.3: A multipoint-to-multipoint mesh. Every point can reach each other at very high speed, or use the central VSAT connection to reach the Internet.

The benefit of this network layout is that even if none of the nodes are in range of a central access point, they can still communicate with each other. Good mesh network implementations are self-healing, in that they automatically detect routing problems and fix them as needed. Extending a mesh network is as simple as adding more nodes. If one of the nodes in the "cloud" happens to be an Internet gateway, then that connection can be shared among all of the clients.

Two big disadvantages to this topology are increased complexity and lower performance. Security in such a network is also a concern, since every participant potentially carries the traffic of every other. Multipoint-to-multipoint networks tend to be complicated to troubleshoot, due to the large number of

changing variables as nodes move around. Multipoint-to-multipoint clouds typically do not have the same capacity as point-to-point or point-to-multipoint networks, due to the additional overhead of managing the network routing and increased contention in the radio spectrum.

Nevertheless, mesh networks are useful in many circumstances. We will see an example of how to build a multipoint-to-multipoint mesh network using a routing protocol called OLSR at the end of this chapter.

Use the technology that fits

All of these network designs can be used to complement each other in a large network, and can obviously make use of traditional wired networking techniques whenever possible. It is a common practice, for example, to use a long distance wireless link to provide Internet access to a remote location, and then set up an access point on the remote side to provide local access. One of the clients to this access point may also act as a mesh node, allowing the network to spread organically between laptop users who all ultimately use the original point-to-point link to access the Internet.

Now that we have a clear idea of the way that wireless networks are typically arranged, we can begin to understand how communication is possible over such networks.

The logical network

Communication is only possible when the participants speak a common language. But once the communication becomes more complex than a simple ongoing broadcast, ***protocol*** becomes just as important as language. All of the people in an auditorium may speak English, but without a set of rules in place to establish who has the right to use the microphone, the communication of an individual's ideas to the entire room is nearly impossible. Now imagine an auditorium as big as the world, full of all of the computers that exist. Without a common set of communication protocols to regulate when and how each computer can speak, the Internet would be a chaotic mess where every machine tries to speak at once.

TCP/IP refers to the suite of protocols that permit conversations to happen on the global Internet. By understanding TCP/IP, you can build networks that will scale to virtually any size, and will ultimately become part of the global Internet.

The TCP/IP model

Data networks are often described as being built on many layers. Each layer depends on the operation of all of the underlying layers before communication can take place, but only needs to exchange data with the layer above or beneath it. The TCP/IP model of networking[2] describes five layers, as shown in this diagram:

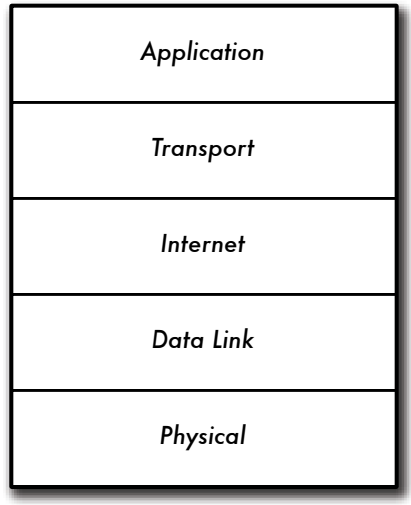

Figure 3.4: The TCP/IP networking model.

The previous section on network layouts described layer one: the ***physical layer***. This is the 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.

The next layer up is referred to as the ***data link layer***. Whenever two or more nodes share the same physical medium (for example, several computers plugged into a hub, or a room full of laptops all using the same radio channel) they use the data link layer to determine whose turn it is to transmit on the medium. 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 can communicate with each other directly. On networks modeled after Ethernet, nodes are referred to by their ***MAC address***, which is a unique 48 bit number assigned to every networking device when it is manufactured.

---

2. The TCP/IP model isn't an international standard, and its definition varies. It is included here as a pragmatic model used for understanding and troubleshooting Internet networks.

Just above the data link layer is the ***Internet layer***. For TCP/IP, this is the Internet Protocol (***IP***). At the Internet layer, packets can leave the link local network and be retransmitted on other networks. Routers perform this function on a network by having at least two network interfaces, one on each of the networks to be interconnected. Nodes on the Internet are reached by their globally unique ***IP address***.

Once Internet routing is possible, a method is needed to reach a particular service at a given IP address. This function is filled by the next layer, the ***transport layer***. TCP and UDP are common examples of transport layer protocols. Some protocols at the transport layer (such as TCP) ensure that all of the data has arrived at the destination, and is reassembled and delivered to the next layer in the proper order.

Finally, at the top of the pile we have the ***application layer***. This is the layer that most network users are exposed to, and is the level at which human communication happens. HTTP, FTP, and SMTP are all application layer protocols. The human sits at the top of all of the layers, and needs little or no knowledge of the layers beneath to effectively use the network.

One way to look at the TCP/IP model is to think of a person delivering a letter to an office building downtown. They first need to interact with the road itself (the physical layer), pay attention to other traffic on the road (the data link layer), turn at the proper place to connect to other roads and arrive at the correct address (the Internet layer), go to the proper floor and room number (the transport layer), and finally find the recipient or a receptionist who can take the letter from there (the application layer). The five layers can be easily remembered by using the mnemonic "**P**lease **D**on't **L**ook **I**n **T**he **A**ttic," which of course stands for "**P**hysical / **D**ata **L**ink / **I**nternet / **T**ransport / **A**pplication."

802.11 wireless networks

Before packets can be forwarded and routed to the Internet, layers one (the physical) and two (the data link) need to be connected. Without link local connectivity, network nodes cannot talk to each other and route packets.

To provide physical connectivity, wireless network devices must operate in the same part of the radio spectrum. As we saw in chapter two, this means that 802.11a radios will talk to 802.11a radios at around 5GHz, and 802.11b/g radios will talk to other 802.11b/g radios at around 2.4GHz. But an 802.11a device cannot interoperate with an 802.11b/g device, since they use completely different parts of the electromagnetic spectrum.

More specifically, wireless cards must agree on a common channel. If one 802.11b radio card is set to channel 2 while another is set to channel 11, then the radios cannot communicate with each other.

When two wireless cards are configured to use the same protocol on the same radio channel, then they are ready to negotiate data link layer connectivity. Each 802.11a/b/g device can operate in one of four possible modes:

1. ***Master mode*** (also called ***AP*** or ***infrastructure mode***) is used to create a service that looks like a traditional access point. The wireless card creates a network with a specified name (called the ***SSID***) and channel, and offers network services on it. While in master mode, wireless cards manage all communications related to the network (authenticating wireless 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, chapter six) 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.

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.

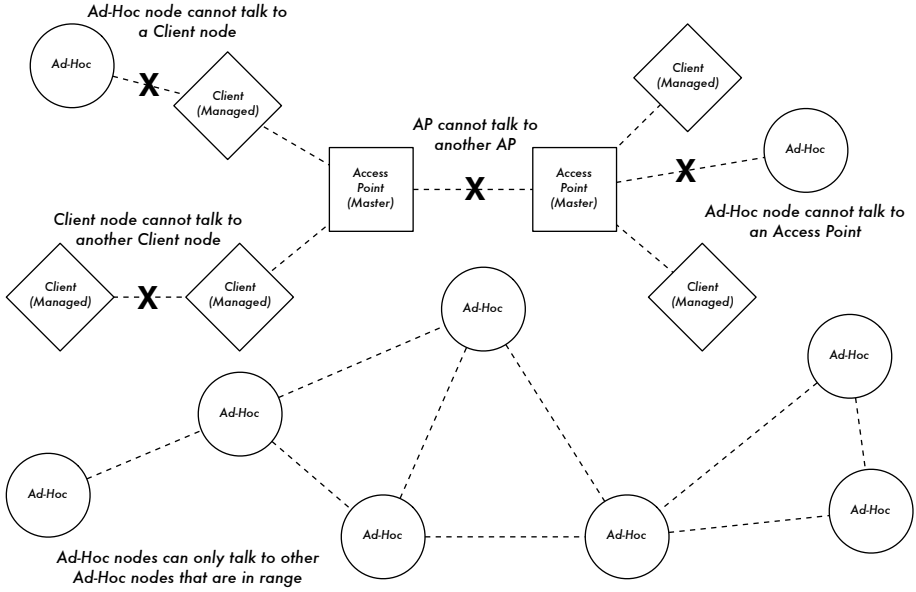

Figure 3.5: APs, Clients, and Ad-Hoc nodes.

It is important to keep these modes in mind when designing your network layout. Remember that managed mode clients cannot communicate with each other directly, so it is likely that you will want to run a high repeater site in master or ad-hoc mode. As we will see later in this chapter, ad-hoc is more flexible but has a number of performance issues as compared to using the master / managed modes.

Now that your wireless cards are providing physical and data link connectivity, they are ready to start passing around packets on layer 3: the internetworking layer.

Internet networking

IP addresses, network addressing, routing, and forwarding are important and related concepts in Internet networking. An ***IP address*** is an identifier for a network node such as a PC, server, router, or bridge. ***Network addressing*** is the system used to assign these identifiers in convenient groups. ***Routing*** keeps track of where in the network these groups may be found. The results of the routing process is kept in a list called a ***routing table***. ***Forwarding*** is the action of using the routing table to send a data packet to either the final destination or to the “next hop” which is closer to the destination.

IP addresses

In an IP[3] network, the address is a 32-bit number, normally written as four 8-bit numbers expressed in decimal form, separated by periods. Examples of IP addresses are 10.0.17.1, 192.168.1.1, or 172.16.5.23.

Network addressing

Interconnected networks must agree on an IP addressing plan. In the global Internet, committees of people allocate groups of IP addresses with a consistent, coherent method to ensure that duplicate addresses are not used by different networks and so that a shorthand can be used to refer to groups of addresses. These groups of addresses are called sub-networks, or ***subnets*** for short. Larger subnets can be further subdivided into smaller subnets. Sometimes a group of related addresses is referred to as an ***address space***.

On the Internet, no person or organization really owns these groups of addresses because the addresses only have meaning if the rest of the Internet community agrees with their usage. By agreement, the addresses are allocated to organizations according to their need and size. An organization which has been allocated an address range may then allocate a portion of that address range to another organization as part of a service agreement. Addresses which have been allocated in this manner, starting with internationally recognized committees, and then broken down hierarchically by national or smaller regional committees are referred to as ***globally routed IP addresses***.

Sometimes it is inconvenient or impossible to get more than one globally routed IP address allocated to an individual or organization. In this case a technique knows as Network Address Translation, or ***NAT*** can be used. A NAT device is a router with two network ports. The outside port uses one globally routed IP address, while the inside port uses an IP address from a special range known as ***private addresses***[4]. The NAT router allows the single global address to be shared with all of the inside users, who all use private addresses. It converts the packets from one form of addressing to the other as the packets pass through it. As far as the network users can tell, they are directly connected to the Internet and require no special software or drivers to share the single globally routed IP address.

---

3. In this book we deal primarily with IPv4, the version of the Internet Protocol in most common use today. While IPv6 will likely replace IPv4 at some point in the future, discussion of IPv6 is currently outside the scope of this book.

4. Private addresses are defined in RFC 1918, http://www.ietf.org/rfc/rfc1918

Routing

The Internet is constantly changing and growing. New networks are continually added, and links between networks are added and removed, fail and come back. It is the job of ***routing*** to determine the best path to the destination, and to create a routing table listing the best path for all the different destinations.

Static routing is the term used when the routing table is created by manual configuration. This is sometimes convenient for small networks but can easily become very difficult and error prone for large networks. Worse, if the best path to a network becomes unusable because of equipment failure or other reasons, static routing will not make use of the next best path.

Dynamic routing is a method in which network elements, in particular routers, exchange information about their state and the state of their neighbours in the network, and then use this information to automatically pick the best path and create the routing table. If something changes, such as a router failing or a new router being put into service, then the dynamic routing protocols make adjustments to the routing table. The system of packet exchanges and decision making is known as a ***routing protocol***. There are many routing protocols that are used in the Internet today, including OSPF, BGP, RIP, and EIGRP.

Wireless networks are like wired networks in that they need dynamic routing protocols, but they are also different enough from wired networks that they need different routing protocols. In particular, wired network connections typically work well or don't work at all (eg., either an Ethernet cable is plugged in, or it isn't). Things are not so clear when working with wireless networks. Wireless communication can be affected by objects moving into the path of the signal, or by interfering signals. Consequently, links may work well, or poorly, or vary between the two extremes. Since existing network protocols don't take the quality of a link into account when making routing decisions, the IEEE 802.11 committees and the IETF are working on standardizing protocols for wireless networks. It is currently unclear when a single standard for dealing with variable link quality will emerge.

In the meantime, there are many ongoing ad-hoc programming attempts to address the problem. Some examples include ***Hazy Sighted Link State*** (***HSLS***), ***Ad-hoc On-demand Distance Vector*** (***AODV***), and ***Optimized Link State Routing*** (***OLSR***). Another is ***SrcRR***, a combination of DSR and ETX implemented by the M.I.T. Roofnet project. Later in this chapter we will see an example of how to implement a network using OLSR to make routing decisions.

Forwarding

Forwarding is straightforward compared to addressing and routing. Each time a router receives a data packet, it consults its internal routing table. Starting with the high order (or most significant) bit, the routing table is searched for the entry that matches the most number of bits in the destination address. This is called the address ***prefix***. If an entry with a matching prefix is found in the routing table, then the ***hop count*** or ***time to live*** (***TTL***) field is decremented. If the result is zero, then the packet is dropped and an error packet is returned to the sender. Otherwise, the packet is sent to the node or interface specified in the routing table. For example, if the routing table contains these entries

```
Destination     Gateway         Genmask         Flags Metric  Iface
10.15.6.0       0.0.0.0         255.255.255.0   U     0       eth1
10.15.6.108     10.15.6.7       255.255.255.255 UG    1       eth1
216.231.38.0    0.0.0.0         255.255.255.0   U     0       eth0
0.0.0.0         216.231.38.1    0.0.0.0         UG    0       eth0
```

...and a packet arrives with the destination address of 10.15.6.23, then the router would send it out on interface eth1. If the packet has a destination of 10.15.6.108, then it would be forwarded to the gateway 10.15.6.7 (since it is more specific and matches more high-order bits than the 10.15.6.0 network route).

A destination of 0.0.0.0 is a special convention referred to as the ***default gateway***. If no other prefixes match the destination address, then the packet is sent to the default gateway. For example, if the destination address was 72.1.140.203, then the router would forward the packet to 216.231.38.1 (which would presumably send it closer to the ultimate destination, and so on).

If a packet arrives and no entry is found (i.e., there is no default gateway defined and no prefix matches a known route), then the packet is dropped and an error packet is returned to the sender.

The TTL field is used to detect routing loops. Without it, a packet could endlessly be sent back and forth between two routers who each list the other as the next best hop. These kinds of loops can cause so much unnecessary traffic on a network that they threaten its stability. Use of the TTL field doesn't fix routing loops, but it does help to prevent them from destroying a network due to simple misconfiguration.

Putting it all together

Once all network nodes have an IP address, they can send data packets to the IP address of any other node. Through the use of routing and forward-

ing, these packets can reach nodes on networks that are not physically connected to the originating node. This process describes much of what "happens" on the Internet. This is illustrated in the following figure:

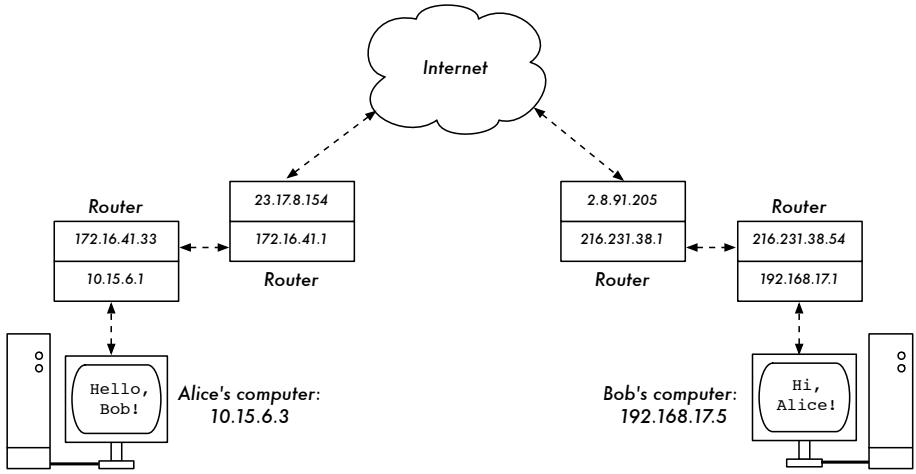

Figure 3.6: Internet networking. Each network segment has a router with two IP addresses, making it "link local" to two different networks. Packets are forwarded between routers until they reach their ultimate destination.

In this example, you can see the path that the packets take as Alice chats with Bob using an instant messaging service. Each dotted line represents an Ethernet cable, a wireless link, or any other kind of physical network. The cloud symbol is commonly used to stand in for "The Internet", and represents any number of intervening IP networks. Neither Alice nor Bob need to be concerned with how those networks operate, as long as the routers forward IP traffic towards the ultimate destination. If it weren't for Internet protocols and the cooperation of everyone on the net, this kind of communication would be impossible.

Now that we have seen how packets flow on IP networks, let's look at a very specialized kind of IP network: an OLSR mesh.

Mesh networking with OLSR

Most WiFi networks operate in infrastructure mode - they consist of an access point somewhere (with a radio operating in master mode), attached to a DSL line or other large scale wired network. In such a ***hotspot*** the access point usually acts as a master station that is distributing Internet access to its clients, which operate in managed mode. This topology is similar to a mobile phone (GSM) service. Mobile phones connect to a base station - without the presence of such a base station mobiles can't communicate with each other. If you make a joke call to a friend that is sitting on the other side of the table,

your phone sends data to the base station of your provider that may be a mile away - the base station then sends data back to the phone of your friend.

WiFi cards in managed mode can't communicate directly, either. Clients - for example, two laptops on the same table - have to use the access point as a relay. Any traffic between clients connected to an access point has to be sent twice. If client A and C communicate, client A sends data to the access point B, and then the access point will retransmit the data to client C. A single transmission may have a speed of 600 kByte/sec (thats about the maximum speed you could achieve with 802.11b) in our example - thus, because the data has to be repeated by the access point before it reaches its target, the effective speed between both clients will be only 300 kByte/sec.

In ad-hoc mode there is no hierarchical master-client relationship. Nodes can communicate directly as long as they are within the range of their wireless interfaces. Thus, in our example both computers could achieve full speed when operating ad-hoc, under ideal circumstances.

The disadvantage to ad-hoc mode is that clients do not repeat traffic destined for other clients. In the access point example, if two clients A and C can't directly “see” each other with their wireless interfaces, they still can communicate as long as the AP is in the wireless range of both clients.

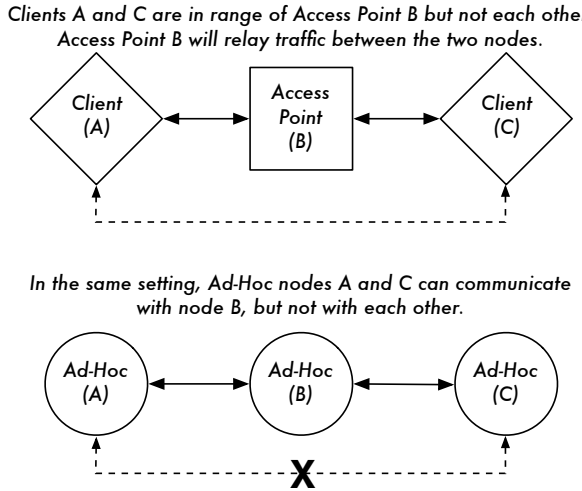

Figure 3.7: Access point B will relay traffic between clients A and C. In Ad-Hoc mode, node B will not relay traffic between A and C by default.

Ad-hoc nodes do not repeat by default, but they can effectively do the same if ***routing*** is applied. Mesh networks are based on the strategy that every mesh-enabled node acts as a relay to extend coverage of the wireless net-

work. The more nodes, the better the radio coverage and range of the mesh cloud.

There is one big tradeoff that must be mentioned at this point. If the device only uses one radio interface, the available bandwidth is significantly reduced every time traffic is repeated by intermediate nodes on the way from A to B. Also, there will be interference in transmission due to nodes sharing the same channel. Thus, cheap ad-hoc mesh networks can provide good radio coverage on the last mile(s) of a community wireless network at the cost of speed-- especially if the density of nodes and transmit power is high.

If an ad-hoc network consists of only a few nodes that are up and running at all time, don't move and always have stable radio links - a long list of ifs - it is possible to write individual routing tables for all nodes by hand.

Unfortunately, those conditions are rarely met in the real world. Nodes can fail, WiFi enabled devices roam around, and interference can make radio links unusable at any time. And no one wants to update several routing tables by hand if one node is added to the network. By using routing protocols that automatically maintain individual routing tables in all nodes involved, we can avoid these issues. Popular routing protocols from the wired world (such as OSPF) do not work well in such an environment because they are not designed to deal with lossy links or rapidly changing topology.

Mesh routing with olsrd

The Optimized Link State Routing Daemon - olsrd - from *olsr.org* is a routing application developed for routing in wireless networks. We will concentrate on this routing software for several reasons. It is a open-source project that supports Mac OS X, Windows 98, 2000, XP, Linux, FreeBSD, OpenBSD and NetBSD. Olsrd is available for access points that run Linux like the Linksys WRT54G, Asus Wl500g, AccessCube or Pocket PCs running Familiar Linux, and ships standard on Metrix kits running Metrix Pebble. Olsrd can handle multiple interfaces and is extensible with plug-ins. It supports IPv6 and it is actively developed and used by community networks all over the world.

Note that there are several implementations of Optimized Link State Routing, which began as an IETF-draft written at INRIA France. The implementation from *olsr.org* started as a master thesis of Andreas Toennesen at UniK University. Based on practical experience of the free networking community, the routing daemon was modified. Olsrd now differs significantly from the original draft because it includes a mechanism called Link Quality Extension that measures the packet loss between nodes and calculates routes according to this information. This extension breaks compatibility to routing daemons that follow the INRIA draft. The olsrd available from *olsr.org* can be configured to behave according to the IETF draft that lacks this feature - but there is no

reason to disable Link Quality Extension unless compliance with other implementations is required.

Theory

After olsrd is running for a while, a node knows about the existence of every other node in the mesh cloud and which nodes may be used to route traffic to them. Each node maintains a routing table covering the whole mesh cloud. This approach to mesh routing is called ***proactive routing***. In contrast, ***reactive routing*** algorithms seek routes only when it is necessary to send data to a specific node.

There are pros and cons to proactive routing, and there are many other ideas about how to do mesh routing that may be worth mentioning. The biggest advantage of proactive routing is that you know who is out there and you don't have to wait until a route is found. Higher protocol traffic overhead and more CPU load are among the disadvantages. In Berlin, the Freifunk community is operating a mesh cloud where olsrd has to manage more than 100 interfaces. The average CPU load caused by olsrd on a Linksys WRT54G running at 200 MHz is about 30% in the Berlin mesh. There is clearly a limit to what extent a proactive protocol can scale - depending on how many interfaces are involved and how often the routing tables are updated. Maintaining routes in a mesh cloud with static nodes takes less effort than a mesh with nodes that are constantly in motion, since the routing table has to be updated less often.

Mechanism

A node running olsrd is constantly broadcasting 'Hello' messages at a given interval so neighbours can detect it's presence. Every node computes a statistic how many 'Hellos' have been lost or received from each neighbour - thereby gaining information about the topology and link quality of nodes in the neighbourhood. The gained topology information is broadcasted as topology control messages (TC messages) and forwarded by neighbours that olsrd has chosen to be 'multipoint' relays.

The concept of multipoint relays is a new idea in proactive routing that came up with the OLSR draft. If every node rebroadcasts topology information that it has received, unnecessary overhead can be generated. Such transmissions are redundant if a node has many neighbours. Thus, an olsrd node decides which neighbours are favorable multipoint relays that should forward its topology control messages. Note that multipoint relays are only chosen for the purpose of forwarding TC messages. Payload is routed considering all available nodes.

Two other message types exist in OLSR that announce information: whether a node offers a gateway to other networks (HNA messages) or has multiple interfaces (MID messages). There is not much to say about what this messages do apart from the fact that they exist. HNA messages make olsrd very convenient when connecting to the Internet with a mobile device. When a mesh node roams around it will detect gateways into other networks and always choose the gateway that it has the best route to. However, olsrd is by no means bullet proof. If a node announces that it is an Internet gateway - which it isn't because it never was or it is just offline at the moment - the other nodes will nevertheless trust this information. The pseudo-gateway is a black hole. To overcome this problem, a dynamic gateway plugin was written. The plugin will automatically detect at the gateway if it is actually connected and whether the link is still up. If not, olsrd ceases to send false HNA messages. It is highly recommended to build and use this plugin instead of statically enabling HNA messages.

Practice

Olsrd implements IP-based routing in a userland application - installation is pretty easy. Installation packages are available for OpenWRT, AccessCube, Mac OS X, Debian GNU/Linux and Windows. OLSR is a standard part of Metrix Pebble. If you have to compile from source, please read the documentation that is shipped with the source package. If everything is configured properly all you have to do is start the olsr program.

First of all, it must be ensured that every node has a unique statically assigned IP-Address for each interface used for the mesh. It is not recommended (nor practicable) to use DHCP in an IP-based mesh network. A DHCP request will not be answered by a DHCP server if the node requesting DHCP needs a multihop link to connect to it, and applying dhcp relay throughout a mesh is likely impractical. This problem could be solved by using IPv6, since there is plenty of space available to generate a unique IP from the MAC address of each card involved (as suggested in "IPv6 Stateless 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 act compliant to the 802.11 standard for ad-hoc networking and thus 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 rather 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 to do 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 nodes on the same wireless channel, and RTS will mitigate this. RTS/CTS adds a handshake before each packet transmission to make sure that the channel is clear. This adds overhead, but increases performance in case of hidden nodes - and hidden nodes are the default in a mesh! This parameter sets the size of the smallest packet (in bytes) for which the node sends RTS. The RTS threshold value must be smaller than the IP-Packet size and the 'Fragmentation threshold' value - here set to 256 - otherwise it will be disabled. TCP is very sensitive to collisions, so it is important to switch RTS on.

Fragmentation allows to split an IP packet in a burst of smaller fragments transmitted on the medium. This adds overhead, but in a noisy environment

this reduces the error penalty and allows packets to get through interference bursts. Mesh networks are very noisy because nodes use the same channel and therefore transmissions are likely to interfere with each other. This parameter sets the maximum size before a data packet is split and sent in a burst - a value equal to the maximum IP packet size disables the mechanism, so it must be smaller than the IP packet size. Setting fragmentation threshold is recommended.

Once a valid IP-address and netmask is assigned and the wireless interface is up, the configuration file of olsrd must be altered in order that olsrd finds and uses the interfaces it is meant to work on.

For Mac OS-X and Windows there are nice GUI's for configuration and monitoring of the daemon available. Unfortunately this tempts users that lack background knowledge to do stupid things - like announcing black holes. On BSD and Linux the configuration file **`/etc/olsrd.conf`** has to be edited with a text editor.

A simple olsrd.conf

We are not going to provide a complete configuration file. Here are some essential settings that should be checked.

```
UseHysteresis           no
TcRedundancy            2
MprCoverage             3
LinkQualityLevel        2
LinkQualityWinSize      20

LoadPlugin "olsrd_dyn_gw.so.0.3"
{
        PlParam     "Interval"   "60"
        PlParam     "Ping"       "151.1.1.1"
        PlParam     "Ping"       "194.25.2.129"
}

Interface "ath0" "wlan0" {
        Ip4Broadcast 255.255.255.255
}
```

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 neighbours 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 then 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.

Many people use olsrd on wired and wireless interfaces - they don't think about network architecture. They just connect antennas to their WiFi cards, connect cables to their Ethernet cards, enable olsrd to run on all computers and all interfaces and fire it up. That is quite an abuse of a protocol that was designed to do wireless networking on lossy links - but - why not?

They expect olsrd to be an ueberprotocol. Clearly it is not necessary to send 'Hello' messages on a wired interface every two seconds - but it works. This

should not be taken as an recommendation - it is just amazing what people do with such a protocol and have success with it. In fact the idea of having a protocol that does everything for newbies that want to have a small to medium sized routed LAN is very appealing...

Plugins

A number of plugins are available for olsrd. Check out the *olsr.org* website for a complete list. Here a little HOWTO for the network topology visualization plugin **`olsrd_dot_draw`**.

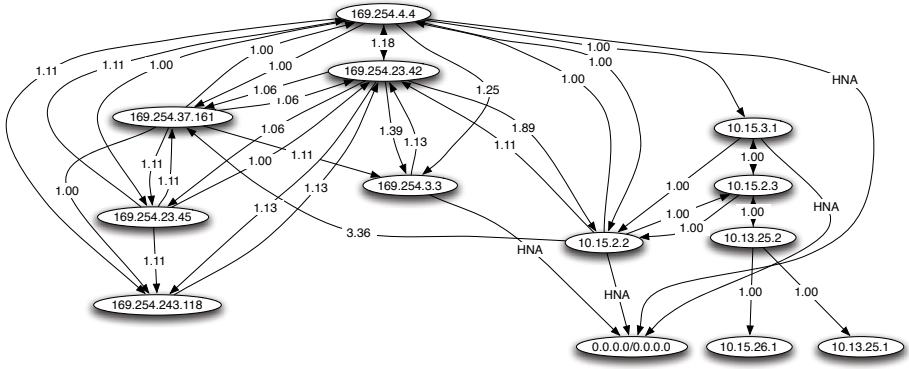

Figure 3.8: An automatically generated OLSR network topology.

Often it is very good for the understanding of a mesh network to have the ability to show the network topology graphically. Olsrd_dot_draw outputs the topology in the dot file format on TCP port 2004. The graphviz tools can then be used to draw the graphs.

Installing the dot_draw Plugin

Compile the olsr plugins separately and install them. To load the plugin add the following lines to **`/etc/olsrd.conf`**

```
LoadPlugin     "olsrd_dot_draw.so.0.3"
{
       PlParam "accept" "192.168.0.5"
       PlParam "port" "2004"
}
```

The parameter "accept" specifies which host is accepted to view the Topology Information (currently only one) and is "localhost" by default. The parameter "port" specifies the TCP port.

Then restart olsr and check if you get output on TCP Port 2004

```
telnet localhost 2004
```

After a while you should get some text output.

Now you can save the output graph descriptions and run the tools **`dot`** or **`neato`** form the graphviz package to get images.

Bruno Randolf has written a small perl script which continuously gets the topology information from olsrd and displays it using the graphviz and ImageMagick tools.

First install the following packages on your workstation:

- graphviz, *http://www.graphviz.org/*
- ImageMagick, *http://www.imagemagick.org/*

Download the script at: *http://meshcube.org/nylon/utils/olsr-topology-view.pl*

Now you can start the script with **`./olsr-topology-view.pl`** and view the topology updates in near-realtime.

Troubleshooting

As long as the WiFi-cards can 'see' each other directly with their radios, doing a ping will work whether olsrd is running or not. This works because the large netmasks effectively make every node link-local, so routing issues are sidestepped at the first hop. This should be checked first if things do not seem to work as expected. Most headaches people face with WiFi in Ad-Hoc mode are caused by the fact that the ad-hoc mode in drivers and cards are implemented sloppily. If it is not possible to ping nodes directly when they are in range it is most likely a card/driver issue, or your network settings are wrong.

If the machines can ping each other, but olsrd doesn't find routes, then the IP-addresses, netmask and broadcast address should be checked.

Are you running a firewall? Make sure it doesn't block UDP port 698.

Have fun!

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 54Mbps radios, but it will only provide up to 22Mbps 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. 22Mbps 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.

Application	BW / User	Notes
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.
Streaming audio	96 - 160 Kbps	Each user of a streaming audio service will use a constant amount of relatively large bandwidth for as long as it plays. It can tolerate some transient latency by using large buffers on the client. But extended periods of lag will cause audio "skips" or outright session failures.
Voice over IP (VoIP)	24 - 100+ Kbps	As with streaming audio, VoIP commits a constant amount of bandwidth to each user for the duration of the call. But with VoIP, the bandwidth is used roughly equally in both directions. Latency on a VoIP connection is immediate and annoying to users. Lag greater than a few milliseconds is unacceptable for VoIP.
Streaming video	64 - 200+ Kbps	As with streaming audio, some intermittent latency is avoided by using buffers on the client. Streaming video requires high throughput and low latency to work properly.
Peer-to-peer filesharing applications (BitTorrent, KaZaA, Gnutella, eDonkey, etc.)	0 - infinite Mbps	While peer to peer applications will tolerate any amount of latency, they tend to use up all available throughput by transmitting data to as many clients as possible, as quickly as possible. Use of these applications will cause latency and throughput problems for all other network users unless you use careful bandwidth shaping.

To estimate the necessary throughput you will need for your network, multiply the expected number of users by the sort of application they will likely use. For example, 50 users who are chiefly browsing the web will likely consume 2.5 to 5Mbps or more of throughput at peak times, and will tolerate some latency. On the other hand, 50 simultaneous VoIP users would require 5Mbps or more of throughput **in both directions** with absolutely no latency. Since 802.11g wireless equipment is ***half duplex*** (that is, it only transmits or receives, never both at once) you should accordingly double the required

throughput, for a total of **10Mbps**. Your wireless links must provide that capacity every second, or conversations will lag.

Since all of your users are unlikely to use the connection at precisely the same moment, it is common practice to ***oversubscribe*** available throughput by some factor (that is, allow more users than the maximum available bandwidth can support). Oversubscribing by a factor of 2 to 5 is quite common. In all likelihood, you will oversubscribe by some amount when building your network infrastructure. By carefully monitoring throughput throughout your network, you will be able to plan when to upgrade various parts of the network, and how much additional resources will be needed.

Expect that no matter how much capacity you supply, your users will eventually find applications that will use it all. As we'll see at the end of this chapter, using bandwidth shaping techniques can help mitigate some latency problems. By using bandwidth shaping, web caching, and other techniques, you can significantly reduce latency and increase overall network throughput.

To get a feeling for the lag felt on very slow connections, the ICTP has put together a bandwidth simulator. It will simultaneously download a web page at full speed and at a reduced rate that you choose. This demonstration gives you an immediate understanding of how low throughput and high latency reduce the usefulness of the Internet as a communications tool. It is available at *http://wireless.ictp.trieste.it/simulator/*

Link planning

A basic communication system consists of two radios, each with its associated antenna, the two being separated by the path to be covered. In order to have a communication between the two, the radios require a certain minimum signal to be collected by the antennas and presented to their input socket. Determining if the link is feasible is a process called ***link budget*** calculation. Whether or not signals can be passed between the radios depends on the quality of the equipment being used and on the diminishment of the signal due to distance, called ***path loss***.

Calculating the link budget

The power available in an 802.11 system can be characterized by the following factors:

- **Transmit Power**. It is expressed in milliwatts or in dBm. Transmit Power ranges from 30mW to 200mW or more. TX power is often dependent on the transmission rate. The TX power of a given device should be specified in the literature provided by the manufacturer, but can sometimes be diffi-

cult to find. Online databases such as the one provided by SeattleWireless (*http://www.seattlewireless.net/HardwareComparison*) may help.

- **Antenna Gain**. Antennas are passive devices that create the effect of amplification by virtue of their physical shape. Antennas have the same characteristics when receiving and transmitting. So a 12 dBi antenna is simply a 12 dBi antenna, without specifying if it is in transmission or reception mode. Parabolic antennas have a gain of 19-24 dBm, omnidirectional antennas have 5-12 dBi, sectorial antennas have roughly a 12-15 dBi gain.
- **Minimum Received Signal Level,** or simply, the sensitivity of the receiver. The minimum RSL is always expressed as a negative dBm (- dBm) and is the lowest power of signal the radio can distinguish. The minimum RSL is dependent upon rate, and as a general rule the lowest rate (1 Mbps) has the greatest sensitivity. The minimum will be typically in the range of -75 to -95 dBm. Like TX power, the RSL specifications should be provided by the manufacturer of the equipment.
- **Cable Losses**. Some of the signal's energy is lost in the cables, the connectors and other devices, going from the radios to the antennas. The loss depends on the type of cable used and on its length. Signal loss for short coaxial cables including connectors is quite low, in the range of 2-3 dB. It is better to have cables as short as possible.

When calculating the path loss, several effects must be considered. One has to take into account the ***free space loss***, ***attenuation*** and ***scattering***. Signal power is diminished by geometric spreading of the wavefront, commonly known as free space loss. Ignoring everything else, the further away the two radios, the smaller the received signal is due to free space loss. This is independent from the environment, depending only on the distance. This loss happens because the radiated signal energy expands as a function of the distance from the transmitter.

Using decibels to express the loss and using 2.45 GHz as the signal frequency, the equation for the free space loss is

```
Lfsl = 40 + 20*log(r)
```

where Lfsl is expressed in dB and r is the distance between the transmitter and receiver, in meters.

The second contribution to the path loss is given by attenuation. This takes place as some of the signal power is absorbed when the wave passes through solid objects such as trees, walls, windows and floors of buildings. Attenuation can vary greatly depending upon the structure of the object the signal is passing through, and it is very difficult to quantify. The most convenient way to express its contribution to the total loss is by adding an "allowed loss" to the free space. For example, experience shows that trees add

10 to 20 dB of loss per tree in the direct path, while walls contribute 10 to 15 dB depending upon the construction.

Along the link path, the RF energy leaves the transmitting antenna and energy spreads out. Some of the RF energy reaches the receiving antenna directly, while some bounces off the ground. Part of the RF energy which bounces off the ground reaches the receiving antenna. Since the reflected signal has a longer way to travel, it arrives at the receiving antenna later than the direct signal. This effect is called ***multipath***, fading or signal dispersion. In some cases reflected signals add together and cause no problem. When they add together out of phase, the received signal is almost worthless. In same cases, the signal at the receiving antenna can be zeroed by the reflected signals. This is known as ***nulling***. There is a simple technique that is used to deal with multipath, called ***antenna diversity***. It consists in adding a second antenna to the radio. Multipath is in fact a very location-specific phenomenon. If two signals add out of phase at one location, they will not add destructively at a second, nearby location. If there are two antennas, at least one of them should be able to receive a usable signal, even if the other is receiving a distorted one. In commercial devices, antenna switching diversity is used: there are multiple antennas on multiple inputs, with a single receiver. The signal is thus received through only one antenna at a time. When transmitting, the radio uses the antenna last used for reception. The distortion given by multipath degrades the ability of the receiver to recover the signal in a manner much like signal loss. A simple way of applying the effects of scattering in the calculation of the path loss is to change the exponent of the distance factor of the free space loss formula. The exponent tends to increase with the range in an environment with a lot of scattering. An exponent of 3 can be used in an outdoor environment with trees, while one of 4 can be used for an indoor environment.

When free space loss, attenuation, and scattering are combined, the path loss is:

```
L(dB) = 40 + 10*n*log(r) + L(allowed)
```

For a rough estimate of the link feasibility, one can evaluate just the free space loss. The environment can bring further signal loss, and should be considered for an exact evaluation of the link. The environment is in fact a very important factor, and should never be neglected.

To evaluate if a link is feasible, one must know the characteristics of the equipment being used and evaluate the path loss. Note that when performing this calculation, you should only add the TX power of one side of the link. If you are using different radios on either side of the link, you should calculate the path loss twice, once for each direction (using the appropriate TX power

for each calculation). Adding up all the gains and subtracting all the losses gives

```
  TX Power Radio 1
+ Antenna Gain Radio 1
- Cable Losses Radio 1
+ Antenna Gain Radio 2
- Cable Losses Radio 2
_____________________________

= Total Gain
```

Subtracting the Path Loss from the Total Gain:

```
  Total Gain
- Path Loss
_____________

= Signal Level at one side of the link
```

If the resulting signal level is greater than the minimum received signal level, then the link is feasible! The received signal is powerful enough for the radios to use it. Remember that the minimum RSL is always expressed as a negative dBm, so -56dBm is greater than -70dBm. On a given path, the variation in path loss over a period of time can be large, so a certain margin (difference between the signal level and the minimum received signal level) should be considered. This margin is the amount of signal above the sensitivity of radio that should be received in order to ensure a stable, high quality radio link during bad weather and other atmospheric disturbances. A margin of error of 10-15 dB is fine. To give some space for attenuation and multipath in the received radio signal, a margin of 20dB should be safe enough.

Once you have calculated the link budget in one direction, repeat the calculation for the other direction. Substitute the transmit power for that of the second radio, and compare the result against the minimum received signal level of the first radio.

Example link budget calculation

As an example, we want to estimate the feasibility of a 5km link, with one access point and one client radio. The access point is connected to an omnidirectional antenna with 10dBi gain, while the client is connected to a sectorial antenna with 14dBi gain. The transmitting power of the AP is 100mW (or 20dBm) and its sensitivity is -89dBm. The transmitting power of the client is 30mW (or 15dBm) and its sensitivity is -82dBm. The cables are short, with a loss of 2dB at each side.

Adding up all the gains and subtracting all the losses for the AP to client link gives:

```
  20 dBm (TX Power Radio 1)
+ 10 dBi (Antenna Gain Radio 1)
-  2 dB  (Cable Losses Radio 1)
+ 14 dBi (Antenna Gain Radio 2)
-  2 dB  (Cable Losses Radio 2)
----------
  40 dB = Total Gain
```

The path loss for a 5km link, considering only the free space loss is:

```
Path Loss = 40 + 20log(5000) = 113 dB
```

Subtracting the path loss from the total gain

```
40 dB - 113 dB = -73 dB
```

Since -73dB is greater than the minimum receive sensitivity of the client radio (-82dBm), the signal level is just enough for the client radio to be able to hear the access point. There is only 9dB of margin (82dB - 73dB) which will likely work fine in fair weather, but may not be enough to protect against extreme weather conditions.

Next we calculate the link from the client back to the access point:

```
  15 dBm (TX Power Radio 2)
+ 14 dBi (Antenna Gain Radio 2)
-  2 dB  (Cable Losses Radio 2)
+ 10 dBi (Antenna Gain Radio 1)
-  2 dB  (Cable Losses Radio 1)
----------
  35 dB = Total Gain
```

Obviously, the path loss is the same on the return trip. So our received signal level on the access point side is:

```
35 dB - 113 dB = -78 dB
```

Since the receive sensitivity of the AP is -89dBm, this leaves us 11dB of fade margin (89dB - 78dB). Overall, this link will probably work but could use a bit more gain. By using a 24dBi dish on the client side rather than a 14dBi sectorial antenna, you will get an additional 10dBi of gain on both directions of the link (remember, antenna gain is reciprocal). A more expensive option would be to use higher power radios on both ends of the link, but note that adding an amplifier or higher powered card to one end does not help the overall quality of the link.

Online tools can be used to calculate the link budget. For example, the Green Bay Professional Packet Radio's Wireless Network Link Analysis (*http://my.athenet.net/~multiplx/cgi-bin/wireless.main.cgi*) is an excellent tool. The Super Edition generates a PDF file containing the Fresnel zone and radio path graphs. The calculation scripts can even be downloaded from the website and installed locally. We will look at one excellent online tool in more detail in the next section, **Link planning software**.

The Terabeam website also has excellent calculators available online (*http://www.terabeam.com/support/calculations/index.php*).

Tables for calculating link budget

To calculate the link budget, simply approximate your link distance, then fill in the following tables:

Free Space Path Loss at 2.4GHz

Distance (m)	100	500	1,000	3,000	5,000	10,000
Loss (dB)	80	94	100	110	113	120

Antenna Gain:

Radio 1 Antenna (dBi)	+ Radio 2 Antenna (dBi)	= Total Antenna Gain

Losses:

Radio 1 + Cable Loss (dB)	Radio 2 + Cable Loss (dB)	Free Space Path Loss (dB)	= Total Loss (dB)

Link Budget for Radio 1 ➞ Radio 2:

Radio 1 TX Power	+ Antenna Gain	- Total Loss	= Signal	> Radio 2 Sensitivity

Link Budget for Radio 2 ➞ Radio 1:

Radio 2 TX Power	+ Antenna Gain	- Total Loss	= Signal	> Radio 1 Sensitivity

If the received signal is greater than the minimum received signal strength in both directions of the link, then the link is feasible.

Link planning software

While calculating a link budget by hand is straightforward, there are a number of tools available that will help automate the process. In addition to calculating free space loss, these tools will take many other relevant factors into account as well (such as tree absorption, terrain effects, climate, and even estimating path loss in urban areas). In this section, we will discuss two free tools that are useful for planning wireless links: Green Bay Professional Packet Radio's online interactive network design utilities, and RadioMobile.

Interactive design CGIs

The Green Bay Professional Packet Radio group (GBPRR) has made a variety of very useful link planning tools available for free online. You can browse these tools online at *http://www.qsl.net/n9zia/wireless/page09.html* . Since the tools are available online, they will work with any device that has a web browser and Internet access.

We will look at the first tool, **Wireless Network Link Analysis**, in detail. You can find it online at *http://my.athenet.net/~multiplx/cgi-bin/wireless.main.cgi* .

To begin, enter the channel to be used on the link. This can be specified in MHz or GHz. If you don't know the frequency, consult the table in Appendix B. Note that the table lists the channel's center frequency, while the tool asks for the highest transmitted frequency. The difference in the ultimate

result is minimal, so feel free to use the center frequency instead. To find the highest transmitted frequency for a channel, just add 11MHz to the center frequency.

Next, enter the details for the transmitter side of the link, including the transmission line type, antenna gain, and other details. Try to fill in as much data as you know or can estimate. You can also enter the antenna height and elevation for this site. This data will be used for calculating the antenna tilt 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.

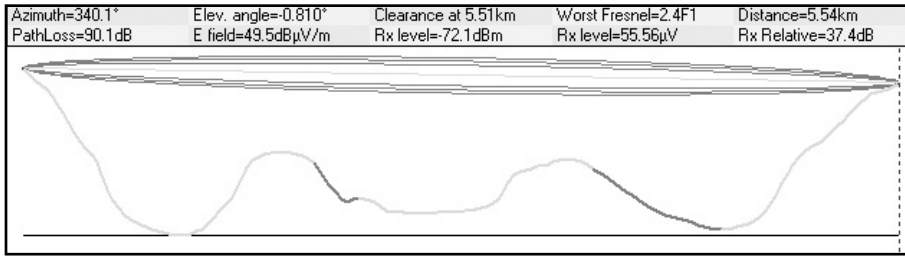

Figure 3.9: Link feasibility, including Fresnel zone and line of sight estimate, using RadioMobile.

The software calculates the coverage area from the base station in a point-to-multipoint system. It works for systems having frequencies from 20 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 scanned maps, satellite photos and Internet map services (such as Mapquest) to produce accurate prediction plots.

Download Radio Mobile here: *http://www.cplus.org/rmw/download.html*

The main Radio Mobile webpage, with examples and tutorials, is available at: *http://www.cplus.org/rmw/english1.html*

RadioMobile under Linux

Radio Mobile will also work using Wine under Ubuntu Linux. While the application runs, some button labels may run beyond the frame of the button and can be hard to read.

We were able to make Radio Mobile work with Linux using the following environment:

- IBM Thinkpad x31
- Ubuntu Breezy (v5.10), *http://www.ubuntu.com/*
- Wine version 20050725, from the Ubuntu Universe repository

There are detailed instructions for installing RadioMobile on Windows at *http://www.cplus.org/rmw/download.html*. You should follow all of the steps except for step 1 (since it is difficult to extract a DLL from the VBRUN60SP6.EXE file under Linux). You will either need to copy the MSVBVM60.DLL file from a Windows machine that already has the Visual

Basic 6 run-time environment installed, or simply Google for MSVBVM60.DLL, and download the file.

Now continue with step 2 at from the above URL, making sure to unzip the downloaded files in the same directory into which you have placed the downloaded DLL file. Note that you don't have to worry about the stuff after step 4; these are extra steps only needed for Windows users.

Finally, you can start Wine from a terminal with the command:

```
# wine RMWDLX.exe
```

You should see RadioMobile running happily in your XWindows session.

Avoiding noise

The unlicensed ISM and U-NII bands represent a very tiny piece of the known electromagnetic spectrum. Since this region can be utilized without paying license fees, many consumer devices use it for a wide range of applications. Cordless phones, analog video senders, Bluetooth, baby monitors, and even microwave ovens compete with wireless data networks for use of the very limited 2.4GHz band. These signals, as well as other local wireless networks, can cause significant problems for long range wireless links. Here are some steps you can use to reduce reception of unwanted signals.

- **Increase antenna gain on both sides of a point-to-point link.** Antennas not only add gain to a link, but their increased directionality tends to reject noise from areas around the link. Two high gain dishes that are pointed at each other will reject noise from directions that are outside the path of the link. Using omnidirectional antennas will receive noise from all directions.
- **Don't use an amplifier.** As we will see in chapter four, amplifiers can make interference issues worse by indiscriminately amplifying all received signals, including sources of interference. Amplifiers also cause interference problems for other nearby users of the band.
- **Use sectorials instead of using an omnidirectional.** By making use of several sectorial antennas, you can reduce the overall noise received at a distribution point. By staggering the channels used on each sectorial, you can also increase the available bandwidth to your clients.

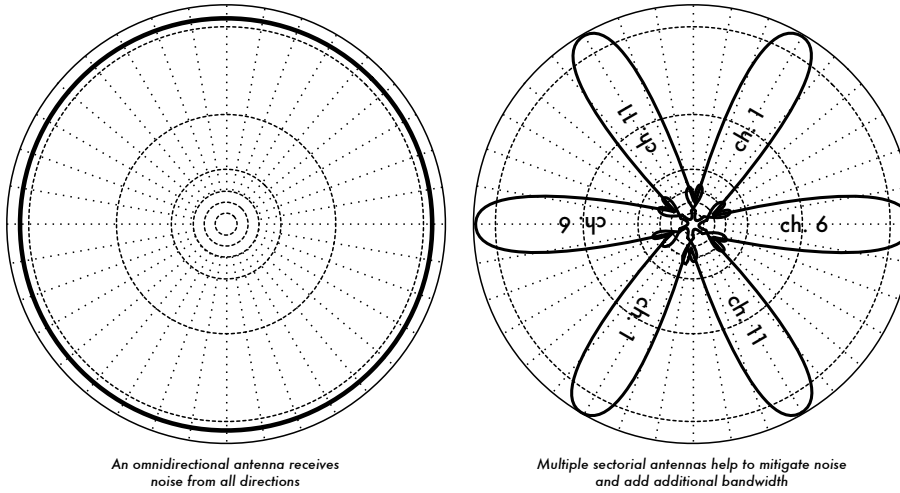

Figure 3.10: A single omnidirectional antenna vs. multiple sectorials.

- **Use the best available channel.** Remember that 802.11b/g channels are 22Mhz wide, but are only separated by 5MHz. Perform a site survey (as detailed in chapter eight), and select a channel that is as far as possible from existing sources of interference. Remember that the wireless landscape can change at any time as people add new devices (cordless phones, other networks, etc.) If your link suddenly has trouble sending packets, you may need to perform another site survey and pick a different channel.
- **Use smaller hops and repeaters, rather than a single long distance shot.** Keep your point-to-point links as short as possible. While it may be possible to create a 12km link that cuts across the middle of a city, you will likely have all kinds of interference problems. If you can break that link into two or three shorter hops, the link will likely be more stable. Obviously this isn't possible on long distance rural links where power and mounting structures are unavailable, but noise problems are also unlikely in those settings.
- **If possible, use 5.8GHz, 900MHz, or another unlicensed band**. While this is only a short term solution, there is currently far more consumer equipment installed in the field that uses 2.4GHz. Using 802.11a or a 2.4GHz to 5.8GHz step-up device will let you avoid this congestion altogether. If you can find it, some old 802.11 equipment uses unlicensed spectrum at 900MHz (unfortunately at much lower bit rates). Other technologies, such as Ronja (*http://ronja.twibright.com/*) use optical technology for short distance, noise-free links.
- **If all else fails, use licensed spectrum**. There are places where all available unlicensed spectrum is effectively used. In these cases, it may make sense to spend the additional money for proprietary equipment that

uses a less congested band. For long distance point-to-point links that require very high throughput and maximum uptime, this is certainly an option. Of course, these features come at a much higher price tag compared to unlicensed equipment.

To identify sources of noise, you need tools that will show you what is happening in the air at 2.4GHz. We will see some examples of these tools in chapter six.

Repeaters

The most critical component to building long distance network links is ***line of sight*** (often abbreviated as ***LOS***). Terrestrial microwave systems simply cannot tolerate large hills, trees, or other obstacles in the path of a long distance link. You must have a clear idea of the lay of the land between two points before you can determine if a link is even possible.

But even if there is a mountain between two points, remember that obstacles can sometimes be turned into assets. Mountains may block your signal, but assuming power can be provided they also make very good ***repeater*** sites.

Repeaters are nodes that are configured to rebroadcast traffic that is not destined for the node itself. In a mesh network, every node is a repeater. In a traditional infrastructure network, nodes must be configured to pass along traffic to other nodes.

A repeater can use one or more wireless devices. When using a single radio (called a ***one-arm repeater***), overall efficiency is slightly less than half of the available bandwidth, since the radio can either send or receive data, but never both at once. These devices are cheaper, simpler, and have lower power requirements. A repeater with two (or more) radio cards can operate all radios at full capacity, as long as they are each configured to use non-overlapping channels. Of course, repeaters can also supply an Ethernet connection to provide local connectivity.

Repeaters can be purchased as a complete hardware solution, or easily assembled by connecting two or more wireless nodes together with Ethernet cable. When planning to use a repeater built with 802.11 technology, remember that nodes must be configured for master, managed, or ad-hoc mode. Typically, both radios in a repeater are configured for master mode, to allow multiple clients to connect to either side of the repeater. But depending on your network layout, one or more devices may need to use ad-hoc or even client mode.

Typically, repeaters are used to overcome obstacles in the path of a long distance link. For example, there may be buildings in your path, but those

buildings contain people. Arrangements can often be worked out with building owners to provide bandwidth in exchange for roof rights and electricity. If the building owner isn't interested, tenants on high floors may be able to be persuaded to install equipment in a window.

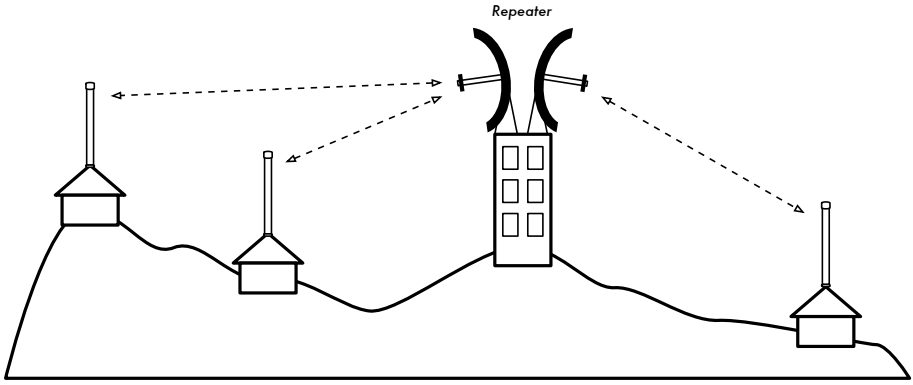

Figure 3.11: The repeater forwards packets over the air between nodes that have no direct line of sight.

If you can't go over or through an obstacle, you can often go around it. Rather than using a direct link, try a multi-hop approach to avoid the obstacle.

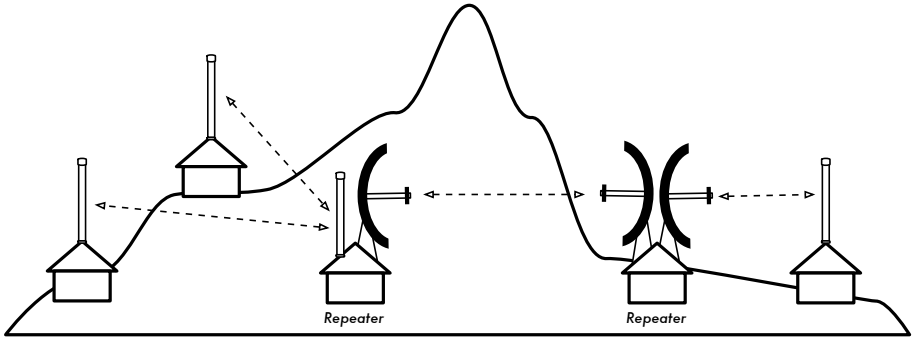

Figure 3.12: No power was available at the top of the hill, but it was circumvented by using multiple repeater sites around the base.

Finally, you may need to consider going backwards in order to go forwards. If there is a high site available in a different direction, and that site can see beyond the obstacle, a stable link can be made via an indirect route.

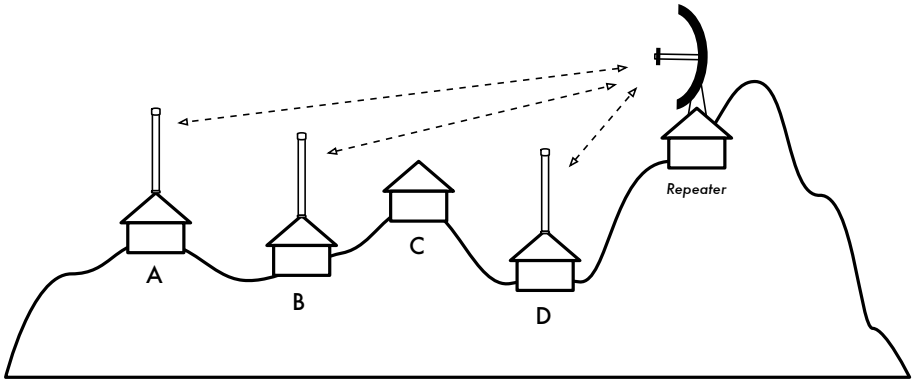

Figure 3.13: Site D could not make a clean link to site A or B, since site C is in the way and is not interested in hosting a node. By installing a high repeater, nodes A, B, and D can communicate. Note that traffic from node D actually travels further away from the rest of the network before the repeater forwards it along.

Repeaters in networks remind me of the "six degrees of separation" principle. This idea says that no matter who you are looking for, you need only contact five intermediaries before finding the person. Repeaters in high places can "see" a great deal of intermediaries, and as long as your node is in range of the repeater, you can communicate with any node the repeater can reach.

Traffic optimization

Bandwidth is measured as a bit rate over a time interval. This means that over time, bandwidth available on any link approaches infinity. Unfortunately, for any given period of time, the bandwidth provided by any given network connection is not infinite. You can always download (or upload) as much traffic as you like; you need only wait long enough. Of course, human users are not as patient as computers, and are not willing to 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 many common techniques for making sure that your network carries only the traffic that must traverse it.

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 *http://www.squid-cache.org/* (most Linux distributions such as Debian, as well as other versions of Unix such as NetBSD and FreeBSD come with Squid). A good Squid configuration guide can be found at *http://squid-docs.sourceforge.net/latest/book-full.html*.
- **Microsoft Proxy server 2.0.** Not available for new installations because it has been superseded by Microsoft ISA server and is no longer supported. It is nonetheless used by some institutions, although it should perhaps not be considered for new installations.
- **Microsoft ISA server.** ISA server is a very good proxy server program, that is arguably too expensive for what it does. However, with academic discounts it may be affordable to some institutions. It produces its own graphical reports, but its log files can also be analyzed with popular analyzer software such as Sawmill (*http://www.sawmill.net/*). Administrators at a site with MS ISA Server should spend sufficient time getting the configuration right; otherwise MS ISA Server can itself be a considerable bandwidth user. For example, a default installation can easily consume more bandwidth than the site has used before, because popular pages with short expiry dates (such as news sites) are continually being refreshed. There-

fore it is important to get the pre-fetching settings right, and to configure pre-fetching to take place mainly overnight. ISA Server can also be tied to content filtering products such as WebSense. For more information, see: *http://www.microsoft.com/isaserver/* and *http://www.isaserver.org/* .

Preventing users from bypassing the proxy server

While circumventing Internet censorship and restrictive information access policy may be a laudable political effort, proxies and firewalls are necessary tools in areas with extremely limited bandwidth. Without them, the stability and usability of the network are threatened by legitimate users themselves. Techniques for bypassing a proxy server can be found at *http://www.antiproxy.com/* . This site is useful for administrators to see how their network measures up against these techniques.

To enforce use of the caching proxy, you might consider simply setting up a network access policy and trusting your users. In the layout below, the administrator has to trust that his users will not bypass the proxy server.

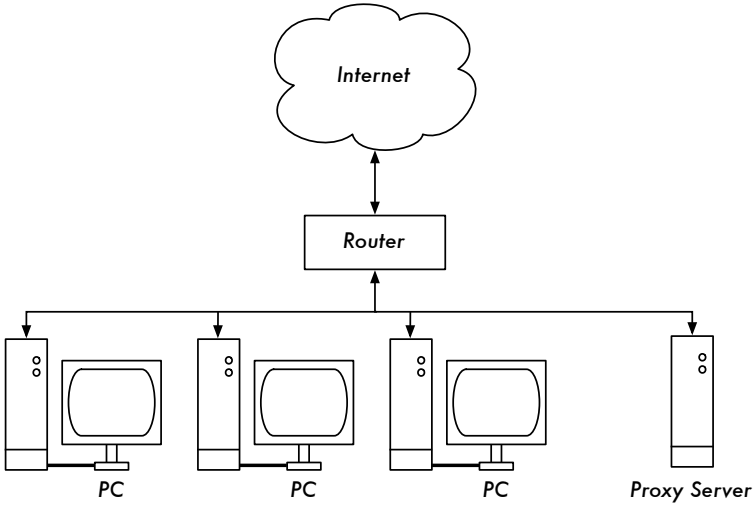

Figure 3.14: This network relies on trusted users to properly configure their PCs to use the proxy server.

In this case the administrator typically uses one of the following techniques:

- **Not giving out the default gateway address through DCHP.** This may work for a while, but some network-savvy users who want to bypass the proxy might find or guess the default gateway address. Once that happens, word tends to spread about how to bypass the proxy.
- **Using domain or group policies.** This is very useful for configuring the correct proxy server settings for Internet Explorer on all computers in the

domain, but is not very useful for preventing the proxy from being bypassed, because it depends on a user logging on to the NT domain. A user with a Windows 95/98/ME computer can cancel his log-on and then bypass the proxy, and someone who knows a local user password on his Windows NT/2000/XP computer can log on locally and do the same.

- **Begging and fighting with users.** This is never an optimal situation for a network administrator.

The only way to ensure that proxies cannot be bypassed is by using the correct network layout, by using one of the three techniques described below.

Firewall

A more reliable way to ensure that PCs don't bypass the proxy can be implemented using the firewall. The firewall can be configured to allow only the proxy server through, i.e. to make HTTP requests to the Internet. All other PCs are blocked, as shown in the diagram below.

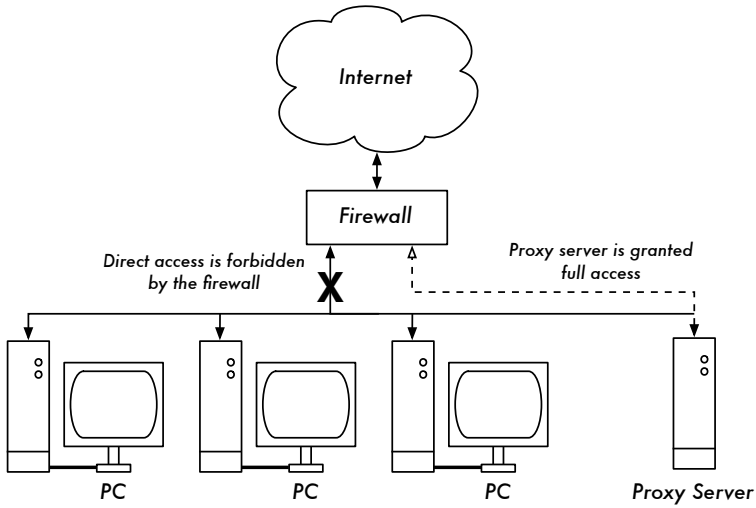

Figure 3.15: The firewall prevents PCs from accessing the Internet directly, but allows access via the proxy server.

Relying on a firewall, as in the above diagram, may or may not be sufficient, depending on how the firewall is configured. If it only blocks access from the campus LAN to port 80 on web servers, there will be ways for clever users to find ways around it. Additionally, they will be able to use other bandwidth hungry protocols such as Kazaa.

Two network cards

Perhaps the most reliable method is to install two network cards in the proxy server and connect the campus network to the Internet as shown below. In this way, the network layout makes it physically impossible to reach the Internet without going through the proxy server.

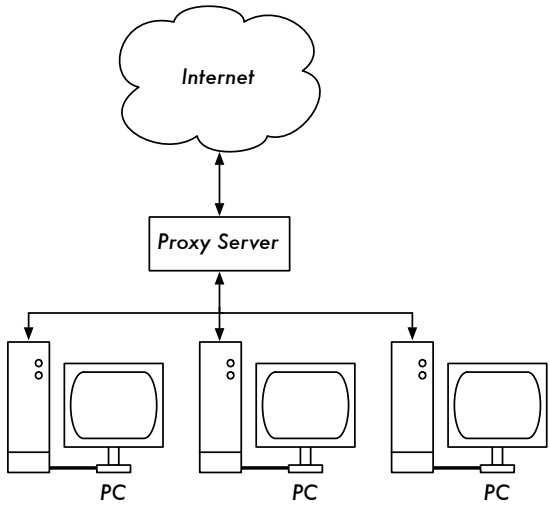

Figure 3.16: The only route to the Internet is through the proxy.

The proxy server in this diagram should not have IP forwarding enabled, unless the administrators knows exactly what they want to let through.

One big advantage to this design is that a technique known as ***transparent proxying*** can be used. Using a transparent proxy means that users' web requests are automatically forwarded to the proxy server, without any need to manually configure web browsers to use it. This effectively forces all web traffic to be cached, eliminates many chances for user error, and will even work with devices that do not support use of a manual proxy. For more details about configuring a transparent proxy with Squid, see:

- *http://www.squid-cache.org/Doc/FAQ/FAQ-17.html*
- *http://en.tldp.org/HOWTO/mini/TransparentProxy-2.html*

Policy-based routing

One way to prevent bypassing of the proxy using Cisco equipment is with policy routing. The Cisco router transparently directs web requests to the proxy server. This technique is used at Makerere University. The advantage

of this method is that, if the proxy server is down, the policy routes can be temporarily removed, allowing clients to connect directly to the Internet.

Mirroring a website

With permission of the owner or web master of a site, the whole site can be mirrored to a local server overnight, if it is not too large. This is something that might be considered for important websites that are of particular interest to the organization or that are very popular with web users. This may have some use, but it has some potential pitfalls. For example, if the site that is mirrored contains CGI scripts or other dynamic content that require interactive input from the user, this would cause problems. An example is a website that requires people to register online for a conference. If someone registers online on a mirrored server (and the mirrored script works), the organizers of the site will not have the information that the person registered.

Because mirroring a site may infringe copyright, this technique should only be used with permission of the site concerned. If the site runs ***rsync***, the site could be mirrored using rsync. This is likely the fastest and most efficient way to keep site contents synchronized. If the remote web server is not running rsync, the recommended software to use is a program called ***wget***. It is part of most versions of Unix/Linux. A Windows version can be found at *http://xoomer.virgilio.it/hherold/*, or in the free Cygwin Unix tools package (*http://www.cygwin.com/*).

A script can be set up to run every night on a local web server and do the following:

- Change directory to the web server document root: for example, **`/var/www/`** on Unix, or **`C:\Inetpub\wwwroot`** on Windows.
- Mirror the website using the command:

```
wget --cache=off -m http://www.python.org
```

The mirrored website will be in a directory **`www.python.org`**. The web server should now be configured to serve the contents of that directory as a name-based virtual host. Set up the local DNS server to fake an entry for this site. For this to work, client PCs should be configured to use the local DNS server(s) as the primary DNS. (This is advisable in any case, because a local caching DNS server speeds up web response times).

Pre-populate the cache using wget

Instead of setting up a mirrored website as described in the previous section, a better approach is to populate the proxy cache using an automated process. This method has been described by J. J. Eksteen and J. P. L. Cloete of

the CSIR in Pretoria, South Africa, in a paper entitled **Enhancing International World Wide Web Access in Mozambique Through the Use of Mirroring and Caching Proxies**. In this paper (available at *http://www.isoc.org/inet97/ans97/cloet.htm*) they describe how the process works:

> "*An automatic process retrieves the site's home page and a specified number of extra pages (by recursively following HTML links on the retrieved pages) through the use of a proxy. Instead of writing the retrieved pages onto the local disk, the mirror process discards the retrieved pages. This is done in order to conserve system resources as well as to avoid possible copyright conflicts. By using the proxy as intermediary, the retrieved pages are guaranteed to be in the cache of the proxy as if a client accessed that page. When a client accesses the retrieved page, it is served from the cache and not over the congested international link. This process can be run in off-peak times in order to maximize bandwidth utilization and not to compete with other access activities.*"

The following command (scheduled to run at night once every day or week) is all that is needed (repeated for every site that needs pre-populating).

```
wget --proxy-on --cache=off --delete after -m http://www.python.org
```

Explanation:

- **-m**: Mirrors the entire site. wget starts at *www.python.org* and follows all hyperlinks, so it downloads all subpages.
- **--proxy-on**: Ensures that wget makes use of the proxy server. This might not be needed in set-ups where a transparent proxy is employed.
- **--cache=off**: Ensures that fresh content is retrieved from the Internet, and not from the local proxy server.
- **--delete after**: Deletes the mirrored copy. The mirrored content remains in the proxy cache if there is sufficient disk space, and the proxy server caching parameters are set up correctly.

In addition, wget has many other options; for example, to supply a password for websites that require them. When using this tool, Squid should be configured with sufficient disk space to contain all the pre-populated sites and more (for normal Squid usage involving pages other than the pre-populated ones). Fortunately, disk space is becoming ever cheaper and disk sizes are far larger than ever before. However, this technique can only be used with a few selected sites. These sites should not be too big for the process to finish before the working day starts, and an eye should be kept on disk space.

Cache hierarchies

When an organization has more than one proxy server, the proxies can share cached information among them. For example, if a web page exists in server A's cache, but not in the cache of server B, a user connected via server B might get the cached object from server A via server B. ***Inter-Cache Protocol*** (***ICP***) and ***Cache Array Routing Protocol*** (***CARP***) can share cache information. CARP is considered the better protocol. Squid supports both protocols, and MS ISA Server supports CARP. For more information, see *http://squid-docs.sourceforge.net/latest/html/c2075.html*. This sharing of cached information reduces bandwidth usage in organizations where more than one proxy is used.

Proxy specifications

On a university campus network, there should be more than one proxy server, both for performance and also for redundancy reasons. With today's cheaper and larger disks, powerful proxy servers can be built, with 50 GB or more disk space allocated to the cache. Disk performance is important, therefore the fastest SCSI disks would perform best (although an IDE based cache is better than none at all). RAID or mirroring is not recommended.

It is also recommended that a separate disk be dedicated to the cache. For example, one disk could be for the cache, and a second for the operating system and cache logging. Squid is designed to use as much RAM as it can get, because when data is retrieved from RAM it is much faster than when it comes from the hard disk. For a campus network, RAM memory should be 1GB or more:

- Apart from the memory required for the operating system and other applications, Squid requires 10 MB of RAM for every 1 GB of disk cache. Therefore, if there is 50 GB of disk space allocated to caching, Squid will require 500 MB extra memory.
- The machine would also require 128 MB for Linux and 128 MB for X-windows.
- Another 256 MB should be added for other applications and in order that everything can run easily. Nothing increases a machine's performance as much as installing a large amount of memory, because this reduces the need to use the hard disk. Memory is thousands of times faster than a hard disk. Modern operating systems keep frequently accessed data in memory if there is enough RAM available. But they use the page file as an extra memory area when they don't have enough RAM.

DNS caching and optimization

Caching-only DNS servers are not authoritative for any domains, but rather just cache results from queries asked of them by clients. Just like a proxy server that caches popular web pages for a certain time, DNS addresses are cached until their ***time to live*** (***TTL***) expires. This will reduce the amount of DNS traffic on your Internet connection, as the DNS cache may be able to satisfy many of the queries locally. Of course, client computers must be configured to use the caching-only name server as their DNS server. When all clients use this server as their primary DNS server, it will quickly populate a cache of IP addresses to names, so that previously requested names can quickly be resolved. DNS servers that are authoritative for a domain also act as cache name-address mappings of hosts resolved by them.

Bind (named)

Bind is the de facto standard program used for name service on the Internet. When Bind is installed and running, it will act as a caching server (no further configuration is necessary). Bind can be installed from a package such as a Debian package or an RPM. Installing from a package is usually the easiest method. In Debian, type

```
apt-get install bind9
```

In addition to running a cache, Bind can also host authoritative zones, act as a slave to authoritative zones, implement split horizon, and just about everything else that is possible with DNS.

dnsmasq

One alternative caching DNS server is ***dnsmasq***. It is available for BSD and most Linux distributions, or from *http://freshmeat.net/projects/dnsmasq/*. The big advantage of dnsmasq is flexibility: it easily acts as both a caching DNS proxy and an authoritative source for hosts and domains, without complicated zone file configuration. Updates can be made to zone data without even restarting the service. It can also serve as a DHCP server, and will integrate DNS service with DHCP host requests. It is very lightweight, stable, and extremely flexible. Bind is likely a better choice for very large networks (more than a couple of hundred nodes), but the simplicity and flexibility of dnsmasq makes it attractive for small to medium sized networks.

Windows NT

To install the DNS service on Windows NT4: select Control Panel ➜ Network ➜ Services ➜ Add ➜ Microsoft DNS server. Insert the Windows NT4 CD

when prompted. Configuring a caching-only server in NT is described in Knowledge Base article 167234. From the article:

> "*Simply install DNS and run the Domain Name System Manager. Click on DNS in the menu, select New Server, and type in the IP address of your computer where you have installed DNS. You now have a caching-only DNS server.*"

Windows 2000

Install DNS service: Start ➜ Settings ➜ Control Panel ➜ Add/Remove Software. In Add/Remove Windows Components, select Components ➜ Networking Services ➜ Details ➜ Domain Name System (DNS). Then start the DNS MMC (Start ➜ Programs ➜ Administrative Tools ➜ DNS) From the Action menu select "Connect To Computer..." In the Select Target Computer window, enable "The following computer:" and enter the name of a DNS server you want to cache. If there is a . [dot] in the DNS manager (this appears by default), this means that the DNS server thinks it is the root DNS server of the Internet. It is certainly not. Delete the . [dot] for anything to work.

Split DNS and a mirrored server

The aim of split DNS (also known as ***split horizon***) is to present a different view of your domain to the inside and outside worlds. There is more than one way to do split DNS; but for security reasons, it's recommended that you have two separate internal and external content DNS servers (each with different databases).

Split DNS can enable clients from a campus network to resolve IP addresses for the campus domain to local RFC1918 IP addresses, while the rest of the Internet resolves the same names to different IP addresses. This is achieved by having two zones on two different DNS servers for the same domain.

One of the zones is used by internal network clients and the other by users on the Internet. For example, in the network below the user on the Makerere campus gets *http://www.makerere.ac.ug/* resolved to 172.16.16.21, whereas a user elsewhere on the Internet gets it resolved to 195.171.16.13.

The DNS server on the campus in the above diagram has a zone file for *makerere.ac.ug* and is configured as if it is authoritative for that domain. In addition, it serves as the DNS caching server for the Makerere campus, and all computers on the campus are configured to use it as their DNS server.

The DNS records for the campus DNS server would look like this:

```
makerere.ac.ug
www     CNAME   webserver.makerere.ac.ug
ftp     CNAME   ftpserver.makerere.ac.ug
mail    CNAME   exchange.makerere.ac.ug
mailserver      A       172.16.16.21
webserver       A       172.16.16.21
ftpserver       A       172.16.16.21
```

But there is another DNS server on the Internet that is actually authoritative for the *makerere.ac.ug* domain. The DNS records for this external zone would look like this:

```
makerere.ac.ug
www     A 195.171.16.13
ftp     A 195.171.16.13
mail    A 16.132.33.21
        MX mail.makerere.ac.ug
```

Split DNS is not dependent on using RFC 1918 addresses. An African ISP might, for example, host websites on behalf of a university but also mirror those same websites in Europe. Whenever clients of that ISP access the website, it gets the IP address at the African ISP, and so the traffic stays in the same country. When visitors from other countries access that website, they get the IP address of the mirrored web server in Europe. In this way, international visitors do not congest the ISP's VSAT connection when visiting the university's website. This is becoming an attractive solution, as web hosting close to the Internet backbone has become very cheap.

Internet link optimization

As mentioned earlier, network throughput of up to 22Mbps can be achieved by using standard, unlicensed 802.11g wireless gear. This amount of bandwidth will likely be at least an order of magnitude higher than that provided by your Internet link, and should be able to comfortably support many simultaneous Internet users.

But if your primary Internet connection is through a VSAT link, you will encounter some performance issues if you rely on default TCP/IP parameters. By optimizing your VSAT link, you can significantly improve response times when accessing Internet hosts.

TCP/IP factors over a satellite connection

A VSAT is often referred to as a ***long fat pipe network***. This term refers to factors that affect TCP/IP performance on any network that has relatively large bandwidth, but high latency. Most Internet connections in Africa and other parts of the developing world are via VSAT. Therefore, even if a university gets its connection via an ISP, this section might apply if the ISP's connection is via VSAT. The high latency in satellite networks is due to the long distance to the satellite and the constant speed of light. This distance adds about 520 ms to a packet's round-trip time (RTT), compared to a typical RTT between Europe and the USA of about 140 ms.

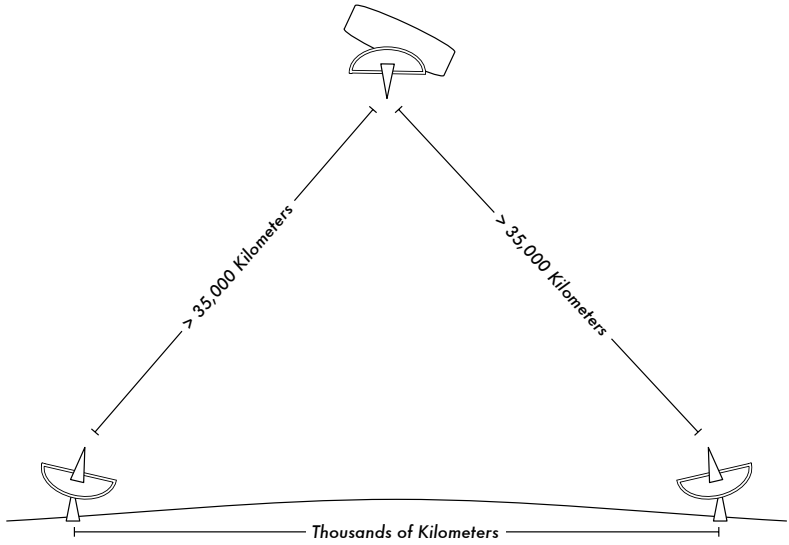

Figure 3.17: Due to the speed of light and long distances involved, a single ping packet can take more than 520ms to be acknowledged over a VSAT link.

The factors that most significantly impact TCP/IP performance are **long RTT**, **large bandwidth delay product**, and **transmission errors**.

Generally speaking, operating systems that support modern TCP/IP implementations should be used in a satellite network. These implementations support the RFC 1323 extensions:

- The ***window scale*** option for supporting large TCP window sizes (larger than 64KB).
- ***Selective acknowledgement*** (***SACK***) to enable faster recovery from transmission errors.
- Timestamps for calculating appropriate RTT and retransmission timeout values for the link in use.

Long round-trip time (RTT)

Satellite links have an average RTT of around 520ms to the first hop. TCP uses the slow-start mechanism at the start of a connection to find the appropriate TCP/IP parameters for that connection. Time spent in the slow-start stage is proportional to the RTT, and for a satellite link it means that TCP stays in slow-start mode for a longer time than would otherwise be the case. This drastically decreases the throughput of short-duration TCP connections. This is can be seen in the way that a small website might take surprisingly long to load, but when a large file is transferred acceptable data rates are achieved after a while.

Furthermore, when packets are lost, TCP enters the congestion-control phase, and owing to the higher RTT, remains in this phase for a longer time, thus reducing the throughput of both short- and long-duration TCP connections.

Large bandwidth-delay product

The amount of data in transit on a link at any point of time is the product of bandwidth and the RTT. Because of the high latency of the satellite link, the bandwidth-delay product is large. TCP/IP allows the remote host to send a certain amount of data in advance without acknowledgment. An acknowledgment is usually required for all incoming data on a TCP/IP connection. However, the remote host is always allowed to send a certain amount of data without acknowledgment, which is important to achieve a good transfer rate on large bandwidth-delay product connections. This amount of data is called the ***TCP window size***. The window size is usually 64KB in modern TCP/IP implementations.

On satellite networks, the value of the bandwidth-delay product is important. To utilize the link fully, the window size of the connection should be equal to the bandwidth-delay product. If the largest window size allowed is 64KB, the maximum theoretical throughput achievable via satellite is (window size) / RTT, or 64KB / 520 ms. This gives a maximum data rate of 123KB/s, which is 984 Kbps, regardless of the fact that the capacity of the link may be much greater.

Each TCP segment header contains a field called ***advertised window***, which specifies how many additional bytes of data the receiver is prepared to accept. The advertised window is the receiver's current available buffer size. The sender is not allowed to send more bytes than the advertised window. To maximize performance, the sender should set its send buffer size and the receiver should set its receive buffer size to no less than the bandwidth-delay

product. This buffer size has a maximum value of 64KB in most modern TCP/IP implementations.

To overcome the problem of TCP/IP stacks from operating systems that don't increase the window size beyond 64KB, a technique known as ***TCP acknowledgment spoofing*** can be used (see Performance Enhancing Proxy, below).

Transmission errors

In older TCP/IP implementations, packet loss is always considered to have been caused by congestion (as opposed to link errors). When this happens, TCP performs congestion avoidance, requiring three duplicate ACKs or slow start in the case of a timeout. Because of the long RTT value, once this congestion-control phase is started, TCP/IP on satellite links will take a longer time to return to the previous throughput level. Therefore errors on a satellite link have a more serious effect on the performance of TCP than over low latency links. To overcome this limitation, mechanisms such as ***Selective Acknowledgment*** (***SACK***) have been developed. SACK specifies exactly those packets that have been received, allowing the sender to retransmit only those segments that are missing because of link errors.

The Microsoft Windows 2000 TCP/IP Implementation Details White Paper states

> "*Windows 2000 introduces support for an important performance feature known as Selective Acknowledgment (SACK). SACK is especially important for connections using large TCP window sizes.*"

SACK has been a standard feature in Linux and BSD kernels for quite some time. Be sure that your Internet router and your ISP's remote side both support SACK.

Implications for universities

If a site has a 512 Kbps connection to the Internet, the default TCP/IP settings are likely sufficient, because a 64 KB window size can fill up to 984 Kbps. But if the university has more than 984 Kbps, it might in some cases not get the full bandwidth of the available link due to the "long fat pipe network" factors discussed above. What these factors really imply is that they prevent a single machine from filling the entire bandwidth. This is not a bad thing during the day, because many people are using the bandwidth. But if, for example, there are large scheduled downloads at night, the administrator might want those downloads to make use of the full bandwidth, and the "long fat pipe network" factors might be an obstacle. This may also become critical

if a significant amount of your network traffic routes through a single tunnel or VPN connection to the other end of the VSAT link.

Administrators might consider taking steps to ensure that the full bandwidth can be achieved by tuning their TCP/IP settings. If a university has implemented a network where all traffic has to go through the proxy (enforced by network layout), then the only machines that make connections to the Internet will be the proxy and mail servers.

For more information, see *http://www.psc.edu/networking/perf_tune.html* .

Performance-enhancing proxy (PEP)

The idea of a Performance-enhancing proxy is described in RFC 3135 (see *http://www.ietf.org/rfc/rfc3135*), and would be a proxy server with a large disk cache that has RFC 1323 extensions, among other features. A laptop has a TCP session with the PEP at the ISP. That PEP, and the one at the satellite provider, communicate using a different TCP session or even their own proprietary protocol. The PEP at the satellite provider gets the files from the web server. In this way, the TCP session is split, and thus the link characteristics that affect protocol performance (long fat pipe factors) are overcome (by TCP acknowledgment spoofing, for example). Additionally, the PEP makes use of proxying and pre-fetching to accelerate web access further.

Such a system can be built from scratch using Squid, for example, or purchased "off the shelf" from a number of vendors.

4
Antennas & Transmission Lines

The transmitter that generates the RF[1] power to drive the antenna is usually located at some distance from the antenna terminals. The connecting link between the two is the ***RF transmission line***. Its purpose is to carry RF power from one place to another, and to do this as efficiently as possible. From the receiver side, the antenna is responsible for picking up any radio signals in the air and passing them to the receiver with the minimum amount of distortion, so that the radio has its best chance to decode the signal. For these reasons, the RF cable has a very important role in radio systems: it must maintain the integrity of the signals in both directions.

There are two main categories of transmission lines: cables and waveguides. Both types work well for efficiently carrying RF power at 2.4GHz.

Cables

RF cables are, for frequencies higher than HF, almost exclusively coaxial cables (or ***coax*** for short, derived from the words "of common axis"). Coax cables have a core ***conductor*** wire surrounded by a non-conductive material called ***dielectric,*** or simply ***insulation***. The dielectric is then surrounded by an encompassing shielding which is often made of braided wires. The dielectric prevents an electrical connection between the core and the shielding. Finally, the coax is protected by an outer casing which is generally made

---

1. Radio Frequency. See chapter two for discussion of electromagnetic waves.

from a PVC material. The inner conductor carries the RF signal, and the outer shield prevents the RF signal from radiating to the atmosphere, and also prevents outside signals from interfering with the signal carried by the core. Another interesting fact is that the electrical signal always travels along the outer layer of the central conductor: the larger the central conductor, the better signal will flow. This is called the "skin effect".

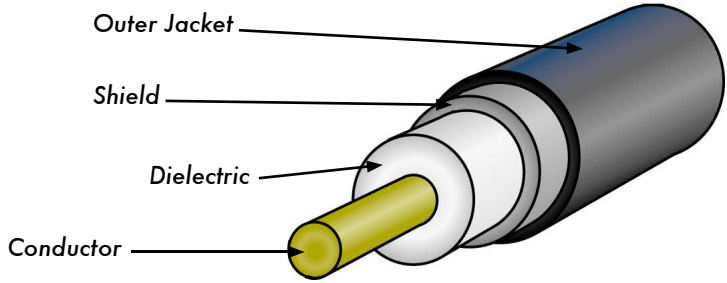

Figure 4.1: Coaxial cable with jacket, shield, dielectric, and core conductor.

Even though the coaxial construction is good at containing the signal on the core wire, there is some resistance to the electrical flow: as the signal travels down the core, it will fade away. This fading is known as ***attenuation***, and for transmission lines it is measured in decibels per meter (***dB/m***). The rate of attenuation is a function of the signal frequency and the physical construction of the cable itself. As the signal frequency increases, so does its attenuation. Obviously, we need to minimize the cable attenuation as much as possible by keeping the cable very short and using high quality cables.

Here are some points to consider when choosing a cable for use with microwave devices:

1. "The shorter the better!" The first rule when you install a piece of cable is to try to keep it as short as possible. The power loss is not linear, so doubling the cable length means that you are going to lose much more than twice the power. In the same way, reducing the cable length by half gives you more than twice the power at the antenna. The best solution is to place the transmitter as close as possible to the antenna, even when this means placing it on a tower.
2. "The cheaper the worse!" The second golden rule is that any money you invest in buying a **good quality** cable is a bargain. Cheap cables are intended to be used at low frequencies, such as VHF. Microwaves require the highest quality cables available. All other options are nothing but a dummy load[2].

2. A dummy load is a device that dissipates RF energy without radiating it. Think of it as a heat sink that works at radio frequencies.

3. Always avoid RG-58. It is intended for thin Ethernet networking, CB or VHF radio, not for microwave.
4. Always avoid RG-213. It is intended for CB and HF radio. In this case the cable diameter does not imply a high quality, or low attenuation.
5. Whenever possible, use ***Heliax*** (also called "Foam") cables for connecting the transmitter to the antenna. When Heliax is unavailable, use the best rated LMR cable you can find. Heliax cables have a solid or tubular center conductor with a corrugated solid outer conductor to enable them to flex. Heliax can be built in two ways, using either air or foam as a dielectric. Air dielectric heliax is the most expensive and guarantees the minimum loss, but it is very difficult to handle. Foam dielectric heliax is slightly more lossy, but is less expensive and easier to install. A special procedure is required when soldering connectors in order to keep the foam dielectric dry and uncorrupted. LMR is a brand of coax cable available in various diameters that works well at microwave frequencies. LMR-400 and LMR-600 are a commonly used alternative to Heliax.
6. Whenever possible, use cables that are pre-crimped and tested in a proper lab. Installing connectors to cable is a tricky business, and is difficult to do properly even with the proper tools. Unless you have access to equipment that can verify a cable you make yourself (such as a spectrum analyzer and signal generator, or time domain reflectometer), troubleshooting a network that uses homemade cable can be difficult.
7. Don't abuse your transmission line. Never step over a cable, bend it too much, or try to unplug a connector by pulling directly the cable. All of those behaviors may change the mechanical characteristic of the cable and therefore its impedance, short the inner conductor to the shield, or even break the line. Those problems are difficult to track and recognize and can lead to unpredictable behavior on the radio link.

Waveguides

Above 2 GHz, the wavelength is short enough to allow practical, efficient energy transfer by different means. A waveguide is a conducting tube through which energy is transmitted in the form of electromagnetic waves. The tube acts as a boundary that confines the waves in the enclosed space. The skin effect prevents any electromagnetic effects from being evident outside the guide. The electromagnetic fields are propagated through the waveguide by means of reflections against its inner walls, which are considered perfect conductors. The intensity of the fields is greatest at the center along the X dimension, and must diminish to zero at the end walls because the existence of any field parallel to the walls at the surface would cause an infinite current to flow in a perfect conductor. Waveguides, of course, cannot carry RF in this fashion.

The X, Y and Z dimensions of a rectangular waveguide can be seen in the following figure:

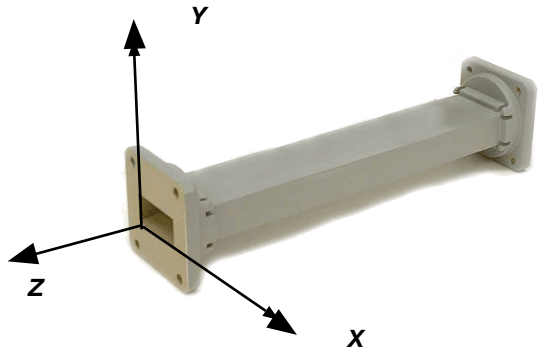

Figure 4.2: The X, Y, and Z dimensions of a rectangular waveguide.

There are an infinite number of ways in which the electric and magnetic fields can arrange themselves in a waveguide for frequencies above the low cutoff frequency. Each of these field configurations is called a ***mode***. The modes may be separated into two general groups. One group, designated ***TM*** (Transverse Magnetic), has the magnetic field entirely transverse to the direction of propagation, but has a component of the electric field in the direction of propagation. The other type, designated ***TE*** (Transverse Electric) has the electric field entirely transverse, but has a component of magnetic field in the direction of propagation.

The mode of propagation is identified by the group letters followed by two subscript numerals. For example, TE 10, TM 11, etc. The number of possible modes increases with the frequency for a given size of guide, and there is only one possible mode, called the ***dominant mode***, for the lowest frequency that can be transmitted. In a rectangular guide, the critical dimension is X. This dimension must be more than 0.5 λ at the lowest frequency to be transmitted. In practice, the Y dimension usually is made about equal to 0.5 X to avoid the possibility of operation in other than the dominant mode. Cross-sectional shapes other than the rectangle can be used, the most important being the circular pipe. Much the same considerations apply as in the rectangular case. Wavelength dimensions for rectangular and circular guides are given in the following table, where X is the width of a rectangular guide and r is the radius of a circular guide. All figures apply to the dominant mode.

Type of guide	Rectangular	Circular
Cutoff wavelength	**2X**	**3.41r**
Longest wavelength transmitted with little attenuation	**1.6X**	**3.2r**
Shortest wavelength before next mode becomes possible	**1.1X**	**2.8r**

Energy may be introduced into or extracted from a waveguide by means of either an electric or magnetic field. The energy transfer typically happens through a coaxial line. Two possible methods for coupling to a coaxial line are using the inner conductor of the coaxial line, or through a loop. A probe which is simply a short extension of the inner conductor of the coaxial line can be oriented so that it is parallel to the electric lines of force. A loop can be arranged so that it encloses some of the magnetic lines of force. The point at which maximum coupling is obtained depends upon the mode of propagation in the guide or cavity. Coupling is maximum when the coupling device is in the most intense field.

If a waveguide is left open at one end, it will radiate energy (that is, it can be used as an antenna rather than as a transmission line). This radiation can be enhanced by flaring the waveguide to form a pyramidal horn antenna. We will see an example of a practical waveguide antenna for WiFi later in this chapter.

Cable Type	Core	Dielectric	Shield	Jacket
RG-58	0.9 mm	2.95 mm	3.8 mm	4.95 mm
RG-213	2.26 mm	7.24 mm	8.64 mm	10.29 mm
LMR-400	2.74 mm	7.24 mm	8.13 mm	10.29 mm
3/8" LDF	3.1 mm	8.12 mm	9.7 mm	11 mm

Here is a table contrasting the sizes of various common transmission lines. Choose the best cable you can afford with the lowest possible attenuation at the frequency you intend to use for your wireless link.

Connectors and adapters

Connectors allow a cable to be connected to another cable or to a component of the RF chain. There are a wide variety of fittings and connectors designed to go with various sizes and types of coaxial lines. We will describe some of the most popular ones.

BNC connectors were developed in the late 40s. BNC stands for Bayonet Neill Concelman, named after the men who invented it: Paul Neill and Carl Concelman. The BNC product line is a miniature quick connect / disconnect connector. It features two bayonet lugs on the female connector, and mating is achieved with only a quarter turn of the coupling nut. BNC's are ideally suited for cable termination for miniature to subminiature coaxial cable (RG-58 to RG-179, RG-316, etc.) They have acceptable performance up to few GHz. They are most commonly found on test equipment and 10base2 coaxial Ethernet cables.

TNC connectors were also invented by Neill and Concelman, and are a threaded variation of the BNC. Due to the better interconnect provided by the threaded connector, TNC connectors work well through about 12GHz. TNC stands for Threaded Neill Concelman.

Type N (again for Neill, although sometimes attributed to “Navy”) connectors were originally developed during the Second World War. They are usable up to 18 Ghz, and very commonly used for microwave applications. They are available for almost all types of cable. Both the plug / cable and plug / socket joints are waterproof, providing an effective cable clamp.

SMA is an acronym for SubMiniature version A, and was developed in the 60s. SMA connectors are precision, subminiature units that provide excellent electrical performance up to 18 GHz. These high-performance connectors are compact in size and mechanically have outstanding durability.

The ***SMB*** name derives from SubMiniature B, and it is the second subminiature design. The SMB is a smaller version of the SMA with snap-on coupling. It provides broadband capability through 4 GHz with a snap-on connector design.

MCX connectors were introduced in the 80s. While the MCX uses identical inner contact and insulator dimensions as the SMB, the outer diameter of the plug is 30% smaller than the SMB. This series provides designers with options where weight and physical space are limited. MCX provides broadband capability though 6 GHz with a snap-on connector design.

In addition to these standard connectors, most WiFi devices use a variety of proprietary connectors. Often, these are simply standard microwave connectors with the center conductor parts reversed, or the thread cut in the opposite direction. These parts are often integrated into a microwave system using a short jumper called a ***pigtail*** that converts the non-standard connector into something more robust and commonly available. Some of these connectors include:

RP-TNC. This is a TNC connector with the genders reversed. These are most commonly found on Linksys equipment, such as the WRT54G.

U.FL (also known as ***MHF***). The U.FL is a patented connector made by Hirose, while the MHF is a mechanically equivalent connector. This is possibly the smallest microwave connector currently in wide use. The U.FL / MHF is typically used to connect a mini-PCI radio card to an antenna or larger connector (such as an N or TNC).

The ***MMCX*** series, which is also called a MicroMate, is one of the smallest RF connector line and was developed in the 90s. MMCX is a micro-miniature connector series with a lock-snap mechanism allowing for 360 degrees rotation enabling flexibility. MMCX connectors are commonly found on PCMCIA radio cards, such as those manufactured by Senao and Cisco.

MC-Card connectors are even smaller and more fragile than MMCX. They have a split outer connector that breaks easily after just a few interconnects. These are commonly found on Lucent / Orinoco / Avaya equipment.

Adapters, which are also called coaxial adapters, are short, two-sided connectors which are used to join two cables or components which cannot be connected directly. Adapters can be used to interconnect devices or cables with different types. For example, an adapter can be used to connect an SMA connector to a BNC. Adapters may also be used to fit together connectors of the same type, but which cannot be directly joined because of their gender. For example a very useful adapter is the one which enables to join two Type N connectors, having socket (female) connectors on both sides.

Figure 4.3: An N female barrel adapter.

Choosing the proper connector

1. "The gender question." Virtually all connectors have a well defined gender consisting of either a pin (the "male" end) or a socket (the "female" end). Usually cables have male connectors on both ends, while RF devices (i.e. transmitters and antennas) have female connectors. Devices such as directional couplers and line-through measuring devices may have both male and female connectors. Be sure that every male connector in your system mates with a female connector.
2. "Less is best!" Try to minimize the number of connectors and adapters in the RF chain. Each connector introduces some additional loss (up to a few dB for each connection, depending on the connector!)
3. "Buy, don't build!" As mentioned earlier, buy cables that are already terminated with the connectors you need whenever possible. Soldering connectors is not an easy task, and to do this job properly is almost impossible for small connectors as U.FL and MMCX. Even terminating "Foam" cables is not an easy task.
4. Don't use BNC for 2.4GHz or higher. Use N type connectors (or SMA, SMB, TNC, etc.)
5. Microwave connectors are precision-made parts, and can be easily damaged by mistreatment. As a general rule, you should rotate the outer sleeve to tighten the connector, leaving the rest of the connector (and cable) stationary. If other parts of the connector are twisted while tightening or loosening, damage can easily occur.
6. Never step over connectors, or drop connectors on the floor when disconnecting cables (this happens more often than what you may imagine, especially when working on a mast over a roof).
7. Never use tools like pliers to tighten connectors. Always use your hands. When working outside, remember that metals expand at high temperatures and reduce their size at low temperatures: a very tightened connector in the summer can bind or even break in winter.

Antennas & radiation patterns

Antennas are a very important component of communication systems. By definition, an antenna is a device used to transform an RF signal traveling on a conductor into an electromagnetic wave in free space. Antennas demonstrate a property known as ***reciprocity***, which means that an antenna will maintain the same characteristics regardless if whether it is transmitting or receiving. Most antennas are resonant devices, which operate efficiently over a relatively narrow frequency band. An antenna must be tuned to the same frequency band of the radio system to which it is connected, otherwise

the reception and the transmission will be impaired. When a signal is fed into an antenna, the antenna will emit radiation distributed in space in a certain way. A graphical representation of the relative distribution of the radiated power in space is called a ***radiation pattern***.

Antenna term glossary

Before we talk about specific antennas, there are a few common terms that must be defined and explained:

Input Impedance

For an efficient transfer of energy, the ***impedance*** of the radio, antenna, and transmission cable connecting them must be the same. Transceivers and their transmission lines are typically designed for 50Ω impedance. If the antenna has an impedance different than 50Ω, then there is a mismatch and an impedance matching circuit is required. When any of these components are mismatched, transmission efficiency suffers.

Return loss

Return loss is another way of expressing mismatch. It is a logarithmic ratio measured in dB that compares the power reflected by the antenna to the power that is fed into the antenna from the transmission line. The relationship between SWR and return loss is the following:

$$\text{Return Loss (in dB)} = 20\log_{10}\frac{\text{SWR}}{\text{SWR-1}}$$

While some energy will always be reflected back into the system, a high return loss will yield unacceptable antenna performance.

Bandwidth

The ***bandwidth*** of an antenna refers to the range of frequencies over which the antenna can operate correctly. The antenna's bandwidth is the number of Hz for which the antenna will exhibit an SWR less than 2:1.

The bandwidth can also be described in terms of percentage of the center frequency of the band.

$$\text{Bandwidth} = 100 \times \frac{F_H - F_L}{F_C}$$

...where F_H is the highest frequency in the band, F_L is the lowest frequency in the band, and F_C is the center frequency in the band.

In this way, bandwidth is constant relative to frequency. If bandwidth was expressed in absolute units of frequency, it would be different depending upon the center frequency. Different types of antennas have different bandwidth limitations.

Directivity and Gain

Directivity is the ability of an antenna to focus energy in a particular direction when transmitting, or to receive energy from a particular direction when receiving. If a wireless link uses fixed locations for both ends, it is possible to use antenna directivity to concentrate the radiation beam in the wanted direction. In a mobile application where the transceiver is not fixed, it may be impossible to predict where the transceiver will be, and so the antenna should ideally radiate as well as possible in all directions. An omnidirectional antenna is used in these applications.

Gain is not a quantity which can be defined in terms of a physical quantity such as the Watt or the Ohm, but it is a dimensionless ratio. Gain is given in reference to a standard antenna. The two most common reference antennas are the ***isotropic antenna*** and the ***resonant half-wave dipole antenna***. The isotropic antenna radiates equally well in all directions. Real isotropic antennas do not exist, but they provide useful and simple theoretical antenna patterns with which to compare real antennas. Any real antenna will radiate more energy in some directions than in others. Since antennas cannot create energy, the total power radiated is the same as an isotropic antenna. Any additional energy radiated in the directions it favors is offset by equally less energy radiated in all other directions.

The gain of an antenna in a given direction is the amount of energy radiated in that direction compared to the energy an isotropic antenna would radiate in the same direction when driven with the same input power. Usually we are only interested in the maximum gain, which is the gain in the direction in which the antenna is radiating most of the power. An antenna gain of 3dB compared to an isotropic antenna would be written as **3dBi**. The resonant half-wave dipole can be a useful standard for comparing to other antennas at one frequency or over a very narrow band of frequencies. To compare the dipole to an antenna over a range of frequencies requires a number of dipoles of different lengths. An antenna gain of 3dB compared to a dipole antenna would be written as **3dBd**.

The method of measuring gain by comparing the antenna under test against a known standard antenna, which has a calibrated gain, is technically known as a ***gain transfer*** technique. Another method for measuring gain is the 3

antennas method, where the transmitted and received power at the antenna terminals is measured between three arbitrary antennas at a known fixed distance.

Radiation Pattern

The ***radiation pattern*** or ***antenna pattern*** describes the relative strength of the radiated field in various directions from the antenna, at a constant distance. The radiation pattern is a reception pattern as well, since it also describes the receiving properties of the antenna. The radiation pattern is three-dimensional, but usually the measured radiation patterns are a two-dimensional slice of the three-dimensional pattern, in the horizontal or vertical planes. These pattern measurements are presented in either a ***rectangular*** or a ***polar*** format. The following figure shows a rectangular plot presentation of a typical ten-element Yagi. The detail is good but it is difficult to visualize the antenna behavior in different directions.

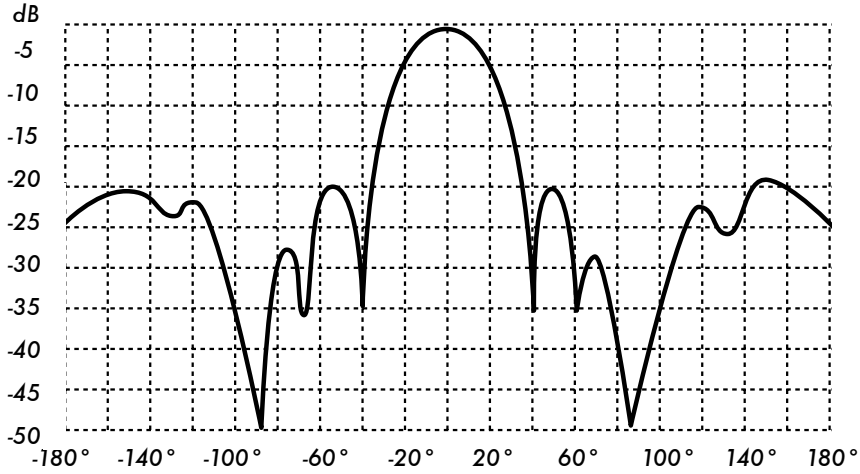

Figure 4.4: A rectangular plot of a yagi radiation pattern.

Polar coordinate systems are used almost universally. In the polar-coordinate graph, points are located by projection along a rotating axis (radius) to an intersection with one of several concentric circles. The following is a polar plot of the same 10 element Yagi antenna.

Polar coordinate systems may be divided generally in two classes: ***linear*** and ***logarithmic***. In the linear coordinate system, the concentric circles are equally spaced, and are graduated. Such a grid may be used to prepare a linear plot of the power contained in the signal. For ease of comparison, the equally spaced concentric circles may be replaced with appropriately placed circles representing the decibel response, referenced to 0 dB at the outer edge of the plot. In this kind of plot the minor lobes are suppressed. Lobes

with peaks more than 15 dB or so below the main lobe disappear because of their small size. This grid enhances plots in which the antenna has a high directivity and small minor lobes. The voltage of the signal, rather than the power, can also be plotted on a linear coordinate system. In this case, too, the directivity is enhanced and the minor lobes suppressed, but not in the same degree as in the linear power grid.

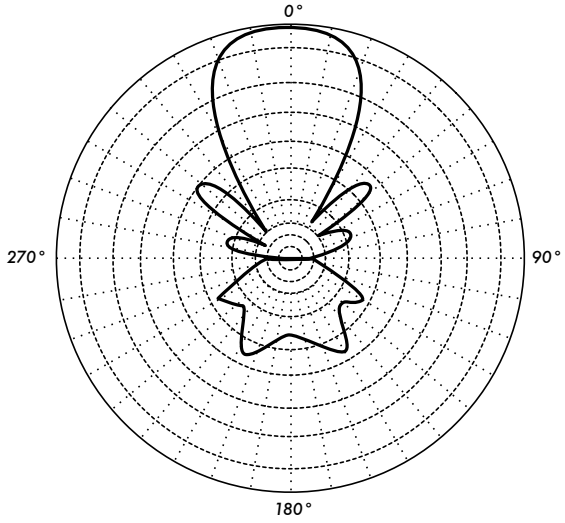

Figure 4.5: A linear polar plot of the same yagi.

In the logarithmic polar coordinate system the concentric grid lines are spaced periodically according to the logarithm of the voltage in the signal. Different values may be used for the logarithmic constant of periodicity, and this choice will have an effect on the appearance of the plotted patterns. Generally the 0 dB reference for the outer edge of the chart is used. With this type of grid, lobes that are 30 or 40 dB below the main lobe are still distinguishable. The spacing between points at 0 dB and at -3 dB is greater than the spacing between -20 dB and -23 dB, which is greater than the spacing between -50 dB and -53 dB. The spacing thus correspond to the relative significance of such changes in antenna performance.

A modified logarithmic scale emphasizes the shape of the major beam while compressing very low-level (>30 dB) sidelobes towards the center of the pattern.

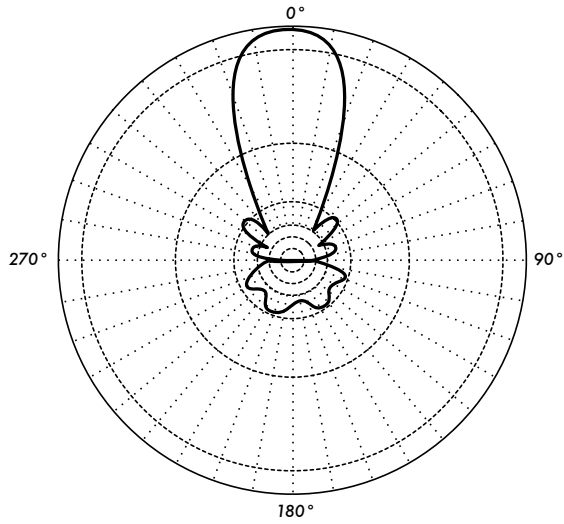

Figure 4.6: The logarithmic polar plot

There are two kinds of radiation pattern: ***absolute*** and ***relative***. Absolute radiation patterns are presented in absolute units of field strength or power. Relative radiation patterns are referenced in relative units of field strength or power. Most radiation pattern measurements are relative to the isotropic antenna, and the gain transfer method is then used to establish the absolute gain of the antenna.

The radiation pattern in the region close to the antenna is not the same as the pattern at large distances. The term near-field refers to the field pattern that exists close to the antenna, while the term far-field refers to the field pattern at large distances. The far-field is also called the radiation field, and is what is most commonly of interest. Ordinarily, it is the radiated power that is of interest, and so antenna patterns are usually measured in the far-field region. For pattern measurement it is important to choose a distance sufficiently large to be in the far-field, well out of the near-field. The minimum permissible distance depends on the dimensions of the antenna in relation to the wavelength. The accepted formula for this distance is:

$$r_{min} = \frac{2d^2}{\lambda}$$

where r_{min} is the minimum distance from the antenna, d is the largest dimension of the antenna, and λ is the wavelength.

Beamwidth

An antenna's ***beamwidth*** is usually understood to mean the half-power beamwidth. The peak radiation intensity is found, and then the points on either side of the peak which represent half the power of the peak intensity are located. The angular distance between the half power points is defined as the beamwidth. Half the power expressed in decibels is -3dB, so the half power beamwidth is sometimes referred to as the 3dB beamwidth. Both horizontal and vertical beamwidths are usually considered.

Assuming that most of the radiated power is not divided into sidelobes, then the directive gain is inversely proportional to the beamwidth: as the beamwidth decreases, the directive gain increases.

Sidelobes

No antenna is able to radiate all the energy in one preferred direction. Some is inevitably radiated in other directions. These smaller peaks are referred to as ***sidelobes***, commonly specified in dB down from the main lobe.

Nulls

In an antenna radiation pattern, a ***null*** is a zone in which the effective radiated power is at a minimum. A null often has a narrow directivity angle compared to that of the main beam. Thus, the null is useful for several purposes, such as suppression of interfering signals in a given direction.

Polarization

Polarization is defined as the orientation of the electric field of an electromagnetic wave. Polarization is in general described by an ellipse. Two special cases of elliptical polarization are ***linear polarization*** and ***circular polarization***. The initial polarization of a radio wave is determined by the antenna.

With linear polarization, the electric field vector stays in the same plane all the time. The electric field may leave the antenna in a vertical orientation, a horizontal orientation, or at some angle between the two. ***Vertically polarized*** radiation is somewhat less affected by reflections over the transmission path. Omnidirectional antennas always have vertical polarization. With ***horizontal polarization***, such reflections cause variations in received signal strength. Horizontal antennas are less likely to pick up man-made interference, which ordinarily is vertically polarized.

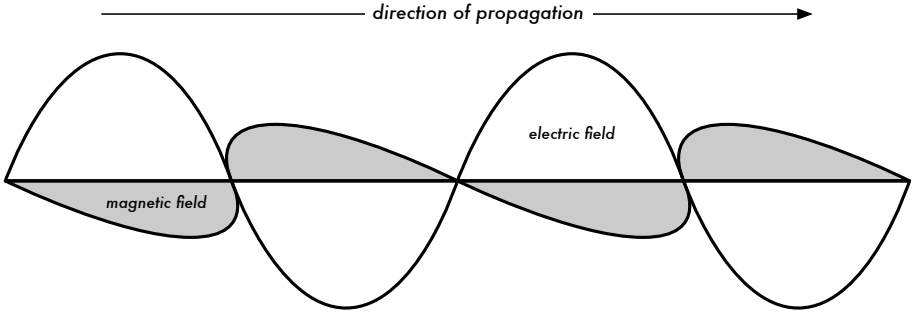

Figure 4.7: The electrical sine wave moves perpendicular to magnetic wave in the direction of propagation.

In circular polarization the electric field vector appears to be rotating with circular motion about the direction of propagation, making one full turn for each RF cycle. This rotation may be right-hand or left-hand. Choice of polarization is one of the design choices available to the RF system designer.

Polarization Mismatch

In order to transfer maximum power between a transmit and a receive antenna, both antennas must have the same spatial orientation, the same polarization sense, and the same axial ratio.

When the antennas are not aligned or do not have the same polarization, there will be a reduction in power transfer between the two antennas. This reduction in power transfer will reduce the overall system efficiency and performance.

When the transmit and receive antennas are both linearly polarized, physical antenna misalignment will result in a polarization mismatch loss, which can be determined using the following formula:

$$\text{Loss (dB)} = 20 \log (\cos \theta)$$

...where θ is the difference in alignment angle between the two antennas. For 15° the loss is approximately 0.3dB, for 30° we lose 1.25dB, for 45° we lose 3dB and for 90° we have an infinite loss.

In short, the greater the mismatch in polarization between a transmitting and receiving antenna, the greater the apparent loss. In the real world, a 90° mismatch in polarization is quite large but not infinite. Some antennas, such as yagis or can antennas, can be simply rotated 90° to match the polarization of the other end of the link. You can use the polarization effect to your advantage on a point-to-point link. Use a monitoring tool to observe interference from adjacent networks, and rotate one antenna until you see the low-

est received signal. Then bring your link online and orient the other end to match polarization. This technique can sometimes be used to build stable links, even in noisy radio environments.

Front-to-back ratio

It is often useful to compare the ***front-to-back ratio*** of directional antennas. This is the ratio of the maximum directivity of an antenna to its directivity in the opposite direction. For example, when the radiation pattern is plotted on a relative dB scale, the front-to-back ratio is the difference in dB between the level of the maximum radiation in the forward direction and the level of radiation at 180 degrees.

This number is meaningless for an omnidirectional antenna, but it gives you an idea of the amount of power directed forward on a very directional antenna.

Types of Antennas

A classification of antennas can be based on:

- **Frequency and size**. Antennas used for HF are different from antennas used for VHF, which in turn are different from antennas for microwave. The wavelength is different at different frequencies, so the antennas must be different in size to radiate signals at the correct wavelength. We are particularly interested in antennas working in the microwave range, especially in the 2.4 GHz and 5 GHz frequencies. At 2.4 GHz the wavelength is 12.5cm, while at 5 GHz it is 6cm.
- **Directivity.** Antennas can be omnidirectional, sectorial or directive. ***Omnidirectional antennas*** radiate roughly the same pattern all around the antenna in a complete 360° pattern. The most popular types of omnidirectional antennas are the ***dipole*** and the ***ground plane***. ***Sectorial antennas*** radiate primarily in a specific area. The beam can be as wide as 180 degrees, or as narrow as 60 degrees. ***Directional*** or ***directive antennas*** are antennas in which the beamwidth is much narrower than in sectorial antennas. They have the highest gain and are therefore used for long distance links. Types of directive antennas are the Yagi, the biquad, the horn, the helicoidal, the patch antenna, the parabolic dish, and many others.
- **Physical construction.** Antennas can be constructed in many different ways, ranging from simple wires, to parabolic dishes, to coffee cans.

When considering antennas suitable for 2.4 GHz WLAN use, another classification can be used:

- **Application**. Access points tend to make point-to-multipoint networks, while remote links are point-to-point. Each of these suggest different types of antennas for their purpose. Nodes that are used for multipoint access will likely use omni antennas which radiate equally in all directions, or sectorial antennas which focus into a small area. In the point-to-point case, antennas are used to connect two single locations together. Directive antennas are the primary choice for this application.

A brief list of common type of antennas for the 2.4 GHz frequency is presented now, with a short description and basic information about their characteristics.

1/4 wavelength ground plane

The 1⁄4 wavelength ground plane antenna is very simple in its construction and is useful for communications when size, cost and ease of construction are important. This antenna is designed to transmit a vertically polarized signal. It consists of a 1⁄4 wave element as half-dipole and three or four 1⁄4 wavelength ground elements bent 30 to 45 degrees down. This set of elements, called radials, is known as a ground plane. This is a simple and effective antenna that can capture a signal equally from all directions. To increase the gain, the signal can be flattened out to take away focus from directly above and below, and providing more focus on the horizon. The vertical beamwidth represents the degree of flatness in the focus. This is useful in a Point-to-Multipoint situation, if all the other antennas are also at the same height. The gain of this antenna is in the order of 2 - 4 dBi.

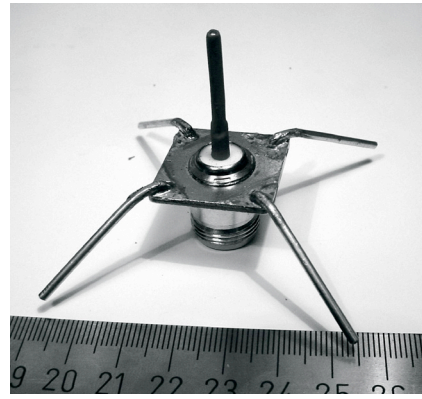

Figure 4.8: Quarter wavelength ground plane antenna.

Yagi antenna

A basic Yagi consists of a certain number of straight elements, each measuring approximately half wavelength. The driven or active element of a Yagi is the equivalent of a center-fed, half-wave dipole antenna. Parallel to the

driven element, and approximately 0.2 to 0.5 wavelength on either side of it, are straight rods or wires called reflectors and directors, or simply passive elements. A reflector is placed behind the driven element and is slightly longer than half wavelength; a director is placed in front of the driven element and is slightly shorter than half wavelength. A typical Yagi has one reflector and one or more directors. The antenna propagates electromagnetic field energy in the direction running from the driven element toward the directors, and is most sensitive to incoming electromagnetic field energy in this same direction. The more directors a Yagi has, the greater the gain. As more directors are added to a Yagi, it therefore becomes longer. Following is the photo of a Yagi antenna with 6 directors and one reflector.

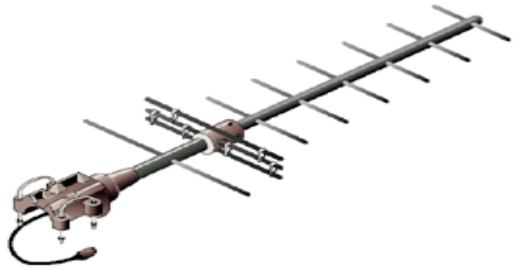

Figure 4.9: A Yagi antenna.

Yagi antennas are used primarily for Point-to-Point links, have a gain from 10 to 20 dBi and a horizontal beamwidth of 10 to 20 degrees.

Horn

The horn antenna derives its name from the characteristic flared appearance. The flared portion can be square, rectangular, cylindrical or conical. The direction of maximum radiation corresponds with the axis of the horn. It is easily fed with a waveguide, but can be fed with a coaxial cable and a proper transition. Horn antennas are commonly used as the active element in a dish antenna. The horn is pointed toward the center of the dish reflector. The use of a horn, rather than a dipole antenna or any other type of antenna, at the focal point of the dish minimizes loss of energy around the edges of the dish reflector. At 2.4 GHz, a simple horn antenna made with a tin can has a gain in the order of 10 - 15 dBi.

Figure 4.10: Feed horn made from a food can.

Parabolic Dish

Antennas based on parabolic reflectors are the most common type of directive antennas when a high gain is required. The main advantage is that they can be made to have gain and directivity as large as required. The main disadvantage is that big dishes are difficult to mount and are likely to have a large windage.

Dishes up to one meter are usually made from solid material. Aluminum is frequently used for its weight advantage, its durability and good electrical characteristics. Windage increases rapidly with dish size and soon becomes a severe problem. Dishes which have a reflecting surface that uses an open mesh are frequently used. These have a poorer front-to-back ratio, but are safer to use and easier to build. Copper, aluminum, brass, galvanized steel and iron are suitable mesh materials.

Figure 4.11: A solid dish antenna.

BiQuad

The BiQuad antenna is simple to build and offers good directivity and gain for Point-to-Point communications. It consists of a two squares of the same size of 1⁄4 wavelength as a radiating element and of a metallic plate or grid as reflector. This antenna has a beamwidth of about 70 degrees and a gain in the order of 10-12 dBi. It can be used as stand-alone antenna or as feeder for a Parabolic Dish. The polarization is such that looking at the antenna from the front, if the squares are placed side by side the polarization is vertical.

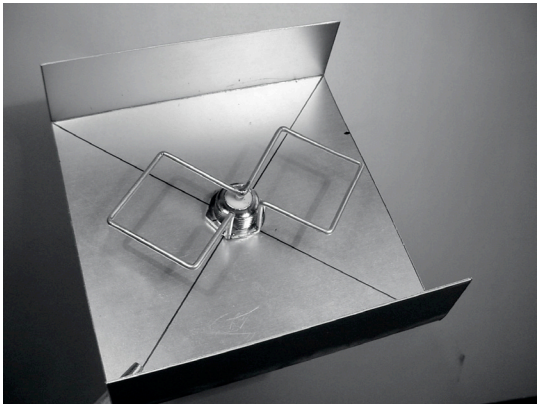

Figure 4.12: The BiQuad.

Other Antennas

Many other types of antennas exist and new ones are created following the advances in technology.

- Sector or Sectorial antennas: they are widely used in cellular telephony infrastructure and are usually built adding a reflective plate to one or more phased dipoles. Their horizontal beamwidth can be as wide as 180 degrees, or as narrow as 60 degrees, while the vertical is usually much narrower. Composite antennas can be built with many Sectors to cover a wider horizontal range (multisectorial antenna).
- Panel or Patch antennas: they are solid flat panels used for indoor coverage, with a gain up to 20 dB.

Reflector theory

The basic property of a perfect parabolic reflector is that it converts a spherical wave irradiating from a point source placed at the focus into a plane wave. Conversely, all the energy received by the dish from a distant source is

reflected to a single point at the focus of the dish. The position of the focus, or focal length, is given by:

$$f = \frac{D^2}{16 \times c}$$

...where D is the dish diameter and c is the depth of the parabola at its center.

The size of the dish is the most important factor since it determines the maximum gain that can be achieved at the given frequency and the resulting beamwidth. The gain and beamwidth obtained are given by:

$$\text{Gain} = \frac{(\pi \times D)^2}{\lambda^2} \times n$$

$$\text{Beamwidth} = \frac{70\ \lambda}{D}$$

...where D is the dish diameter and n is the efficiency. The efficiency is determined mainly by the effectiveness of illumination of the dish by the feed, but also by other factors. Each time the diameter of a dish is doubled, the gain is four times, or 6 dB, greater. If both stations double the size of their dishes, signal strength can be increased of 12 dB, a very substantial gain. An efficiency of 50% can be assumed when hand-building the antenna.

The ratio f / D (focal length/diameter of the dish) is the fundamental factor governing the design of the feed for a dish. The ratio is directly related to the beamwidth of the feed necessary to illuminate the dish effectively. Two dishes of the same diameter but different focal lengths require different design of feed if both are to be illuminated efficiently. The value of 0.25 corresponds to the common focal-plane dish in which the focus is in the same plane as the rim of the dish.

Amplifiers

As mentioned earlier, antennas do not actually create power. They simply direct all available power into a particular pattern. By using a ***power amplifier***, you can use DC power to augment your available signal. An amplifier connects between the radio transmitter and the antenna, and has an additional lead that connects to a power source. Amplifiers are available that work at 2.4GHz, and can add several Watts of power to your transmission. These devices sense when an attached radio is transmitting, and quickly

power up and amplify the signal. They then switch off again when transmission ends. When receiving, they also add amplification to the signal before sending it to the radio.

Unfortunately, simply adding amplifiers will not magically solve all of your networking problems. We do not discuss power amplifiers at length in this book because there are a number of significant drawbacks to using them:

- **They are expensive**. Amplifiers must work at relatively wide bandwidths at 2.4GHz, and must switch quickly enough to work for Wi-Fi applications. These amplifiers do exist, but they tend to cost several hundred dollars per unit.
- **You will need at least two**. Whereas antennas provide reciprocal gain that benefits both sides of a connection, amplifiers work best at amplifying a transmitted signal. If you only add an amplifier to one end of a link with insufficient antenna gain, it will likely be able to be heard but will not be able to hear the other end.
- **They provide no additional directionality.** Adding antenna gain provides both gain and directionality benefits to both ends of the link. They not only improve the available amount of signal, but tend to reject noise from other directions. Amplifiers blindly amplify both desired and interfering signals, and can make interference problems worse.
- **Amplifiers generate noise for other users of the band.** By increasing your output power, you are creating a louder source of noise for other users of the unlicensed band. This may not be much of an issue today in rural areas, but it can cause big problems in populated areas. Conversely, adding antenna gain will improve your link and can actually decrease the noise level for your neighbors.
- **Using amplifiers probably isn't legal.** Every country imposes power limits on use of unlicensed spectrum. Adding an antenna to a highly amplified signal will likely cause the link to exceed legal limits.

Using amplifiers is often compared to the inconsiderate neighbor who wants to listen to the radio outside their home, and so turns it up to full volume. They might even "improve" reception by pointing their speakers out the window. While they may now be able to hear the radio, so must everyone else on the block. This approach may scale to exactly one user, but what happens when the neighbors decide to do the same thing with their radios? Using amplifiers for a wireless link causes roughly the same effect at 2.4GHz. Your link may "work better" for the moment, but you will have problems when other users of the band decide to use amplifiers of their own.

By using higher gain antennas rather than amplifiers, you avoid all of these problems. Antennas cost far less than amps, and can improve a link simply

by changing the antenna on one end. Using more sensitive radios and good quality cable also helps significantly on long distance shots. These techniques are unlikely to cause problems for other users of the band, and so we recommend pursuing them long before adding amplifiers.

Practical antenna designs

The cost of 2.4GHz antennas has fallen dramatically since the introduction of 802.11b. Innovative designs use simpler parts and fewer materials to achieve impressive gain with relatively little machining. Unfortunately, availability of good antennas is still limited in many areas of the world, and importing them can be prohibitively expensive. While actually designing an antenna can be a complex and error-prone process, constructing antennas from locally available components is very straightforward, and can be a lot of fun. We present four practical antenna designs that can be built for very little money.

USB dongle as dish feed

Possibly the simplest antenna design is the use of a parabola to direct the output of a USB wireless device (known in networking circles as a ***USB dongle***). By placing the internal dipole antenna present in USB wireless dongles at the apex of a parabolic dish, you can provide significant gain without the need to solder or even open the wireless device itself. Many kinds of parabolic dishes will work, including satellite dishes, television antennas, and even metal cookware (such as a wok, round lid, or strainer). As a bonus, inexpensive and lossless USB cable is then used to feed the antenna, eliminating the need for expensive coaxial cable or heliax.

To build a USB dongle parabolic, you will need to find the orientation and location of the dipole inside the dongle. Most devices orient the dipole to be parallel with the short edge of the dongle, but some will mount the dipole perpendicular to the short edge. You can either open the dongle and look for yourself, or simply try the dongle in both positions to see which provides more gain.

To test the antenna, point it at an access point several meters away, and connect the USB dongle to a laptop. Using the laptop's client driver or a tool such as Netstumbler (see chapter six), observe the received signal strength of the access point. Now, slowly move the dongle in relation to the parabolic while watching the signal strength meter. You should see a significant improvement in gain (20 dB or more) when you find the proper position. The dipole itself is typically placed 3 to 5 centimeters from the back of the dish, but this will vary depending on the shape of the parabola. Try various posi-

tions while watching your signal strength meter until you find the optimum location.

Once the best location is found, securely fix the dongle in place. You will need to waterproof the dongle and cable if the antenna is used outdoors. Use a silicone compound or a piece of PVC tubing to seal the electronics against the weather. Many USB-fed parabolic designs and ideas are documented online at *http://www.usbwifi.orcon.net.nz/* .

Collinear omni

This antenna is very simple to build, requiring just a piece of wire, an N socket and a square metallic plate. It can be used for indoor or outdoor Point-to-MultiPoint short distance coverage. The plate has a hole drilled in the middle to accommodate an N type chassis socket that is screwed into place. The wire is soldered to the center pin of the N socket and has coils to separate the active phased elements. Two versions of the antenna are possible: one with two phased elements and two coils and another with four phased elements and four coils. For the short antenna the gain will be around 5dBi, while the long one with four elements will have 7 to 9 dBi of gain. We are going to describe how to build the long antenna only.

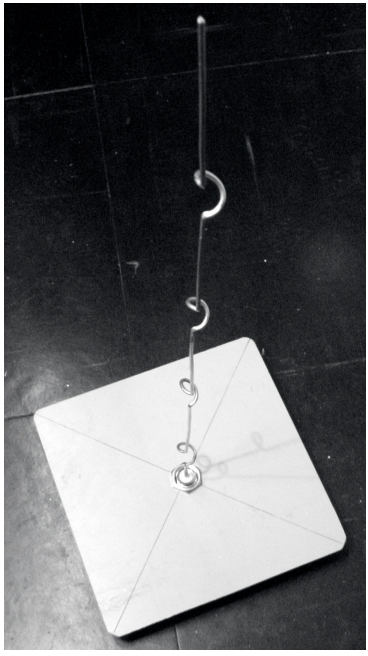

Figure 4.13: The completed colinear omni

Parts list

- One screw-on N-type female connector
- 50 cm of copper or brass wire of 2 mm of diameter
- 10x10 cm or greater square metallic plate

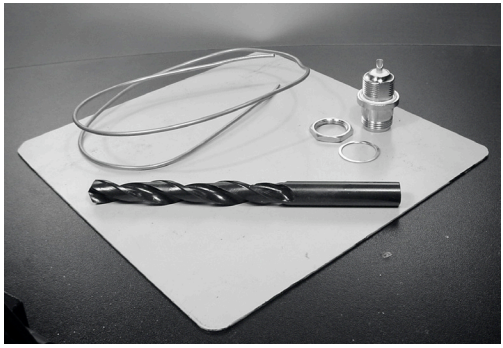

Figure 4.14: 10 cm x 10 cm aluminum plate.

Tools required

- Ruler
- Pliers
- File
- Soldering iron and solder
- Drill with a set of bits for metal (including a 1.5 cm diameter bit)
- A piece of pipe or a drill bit with a diameter of 1 cm
- Vice or clamp
- Hammer
- Spanner or monkey wrench

Construction

1. Straighten the wire using the vice.

Figure 4.15: Make the wire as straight as you can.

2. With a marker, draw a line at 2.5 cm starting from one end of the wire. On this line, bend the wire at 90 degrees with the help of the vice and of the hammer.

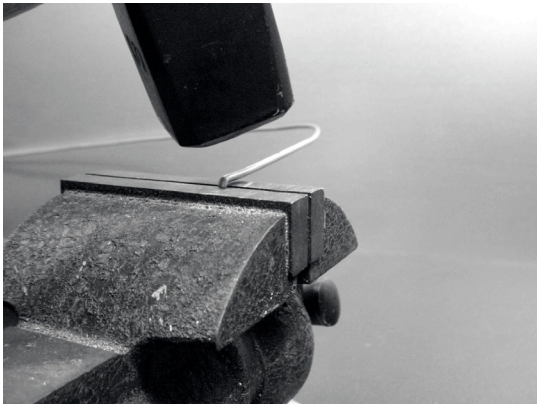

Figure 4.16: Gently tap the wire to make a sharp bend.

3. Draw another line at a distance of 3.6 cm from the bend. Using the vice and the hammer, bend once again the wire over this second line at 90 degrees, in the opposite direction to the first bend but in the same plane. The wire should look like a 'Z'.

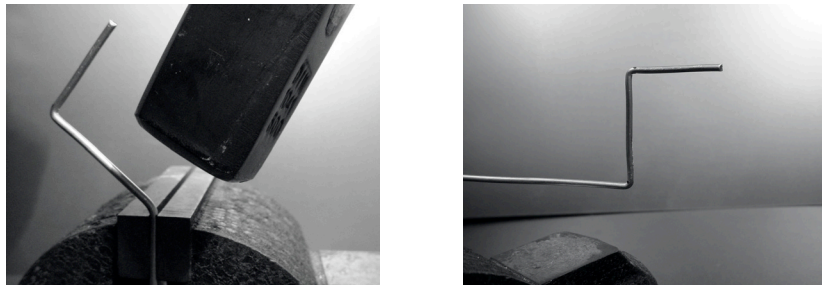

Figure 4.17: Bend the wire into a "Z" shape.

4. We will now twist the 'Z' portion of the wire to make a coil with a diameter of 1 cm. To do this, we will use the pipe or the drill bit and curve the wire around it, with the help of the vice and of the pliers.

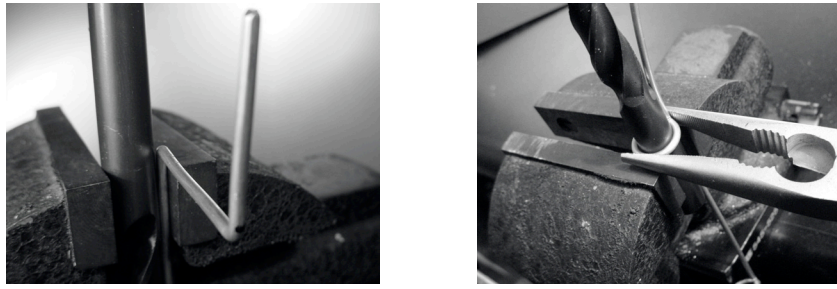

Figure 4.18: Bend the wire around the drill bit to make a coil.

The coil will look like this:

Figure 4.19: The completed coil.

5. You should make a second coil at a distance of 7.8 cm from the first one. Both coils should have the same turning direction and should be placed on the same side of the wire. Make a third and a fourth coil following the

same procedure, at the same distance of 7.8 cm one from each other. Trim the last phased element at a distance of 8.0 cm from the fourth coil.

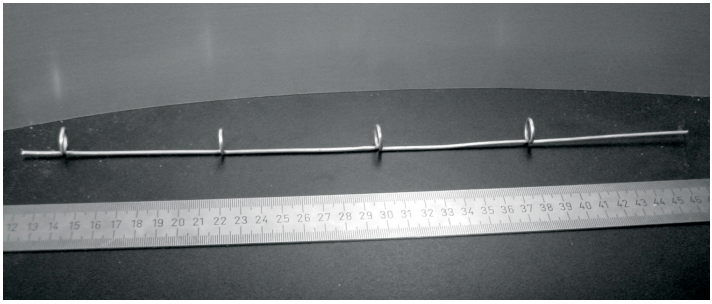

Figure 4.20: Try to keep it as straight possible.

If the coils have been made correctly, it should now be possible to insert a pipe through all the coils as shown.

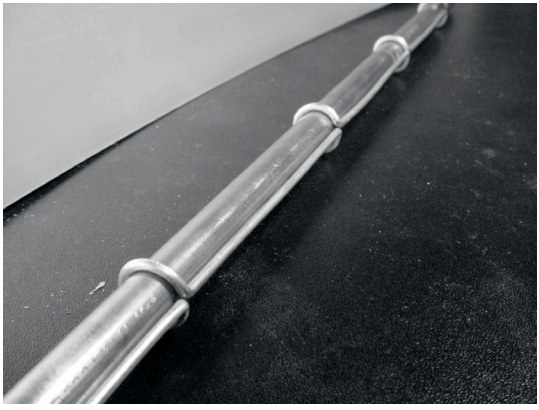

Figure 4.21: Inserting a pipe can help to straighten the wire.

6. With a marker and a ruler, draw the diagonals on the metallic plate, finding its center. With a small diameter drill bit, make a pilot hole at the center of the plate. Increase the diameter of the hole using bits with an increasing diameter.

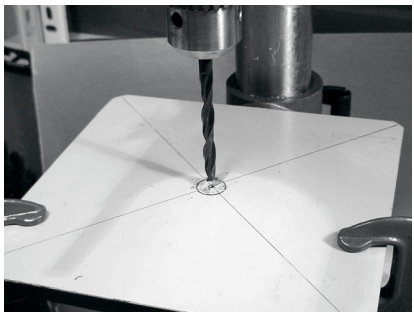


Figure 4.22: Drilling the hole in the metal plate.

The hole should fit the N connector exactly. Use a file if needed.

Figure 4.23: The N connector should fit snugly in the hole.

7. To have an antenna impedance of 50 Ohms, it is important that the visible surface of the internal insulator of the connector (the white area around the central pin) is at the same level as the surface of the plate. For this reason, cut 0.5 cm of copper pipe with an external diameter of 2 cm, and place it between the connector and the plate.

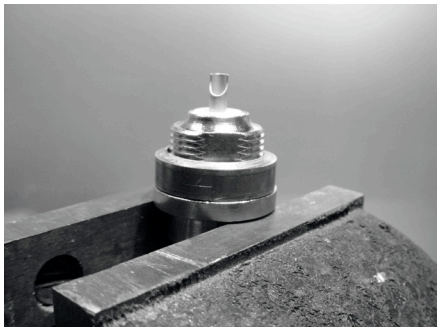

Figure 4.24: Adding a copper pipe spacer helps to match the impedance of the antenna to 50 Ohms.

8. Screw the nut to the connector to fix it firmly on the plate using the spanner.

Figure 4.25: Secure the N connector tightly to the plate.

9. Smooth with the file the side of the wire which is 2.5 cm long, from the first coil. Tin the wire for around 0.5 cm at the smoothed end helping yourself with the vice.

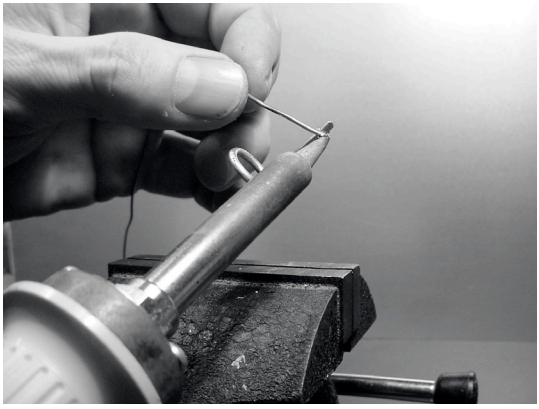

Figure 4.26: Add a little solder to the end of the wire to "tin" it prior to soldering.

10. With the soldering iron, tin the central pin of the connector. Keeping the wire vertical with the pliers, solder its tinned side in the hole of the central pin. The first coil should be at 3.0 cm from the plate.

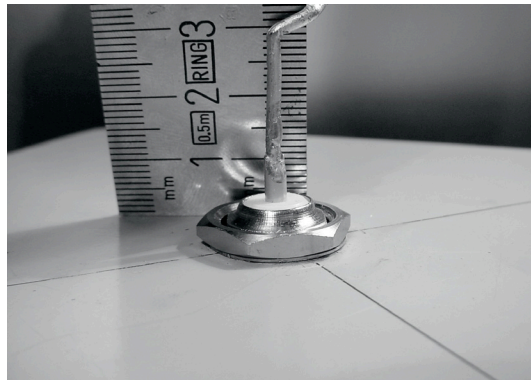

Figure 4.27: The first coil should start 3.0 cm from the surface of the plate.

11. We are now going to stretch the coils extending the total vertical length of the wire. Using the use the vice and the pliers, you should pull the cable so that the final length of the coil is of 2.0 cm.

Figure 4.30: Stretching the coils. Be very gentle and try not to scrape the surface of the wire with the pliers.

12. Repeat the same procedure for the other three coils, stretching their length to 2.0 cm.

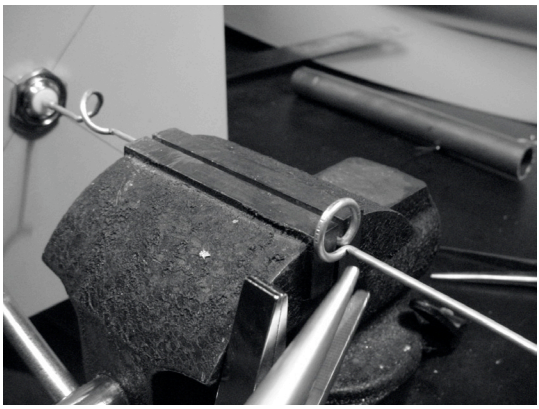

Figure 4.29: Repeat the stretching procedure for all of the remaining coils.

13. At the end the antenna should measure 42.5 cm from the plate to the top.

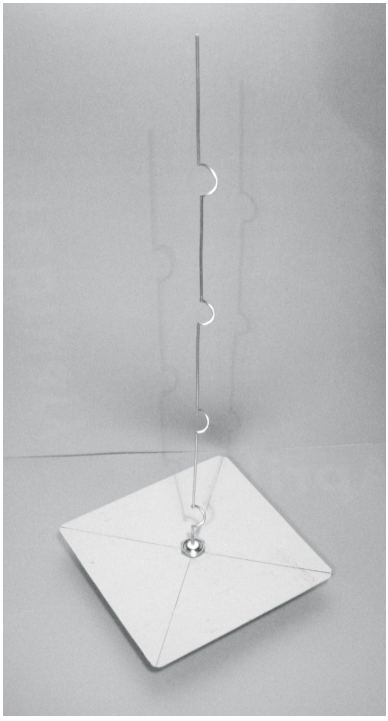

Figure 4.30: The finished antenna should be 42.5 cm from the plate to the end of the wire.

14. If you have a Spectrum Analyzer with Tracking Generator and a Directional Coupler, you can check the curve of the reflected power of the antenna. The picture below shows the display of the Spectrum Analyzer.

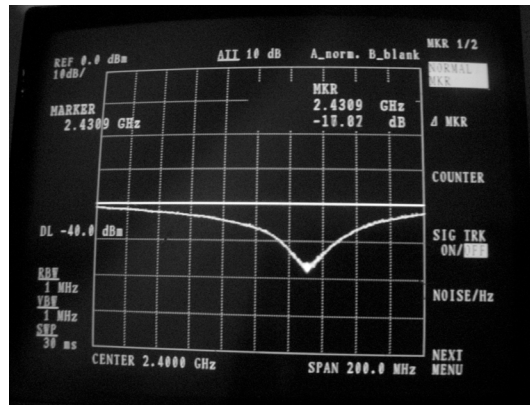

Figure 4.31: A spectrum plot of the reflected power of the collinear omni.

If you intend to use this antenna outside, you will need to weatherproof it. The simplest method is to enclose the whole thing in a large piece of PVC pipe closed with caps. Cut a hole at the bottom for the transmission line, and seal the antenna shut with silicone or PVC glue.

Cantenna

This antenna, sometimes called a Cantenna, uses a tin can as a waveguide and a short wire soldered on an N connector as a probe for coaxial-cable-to-waveguide transition. It can be easily built at just the price of the connector, recycling a food, juice, or other tin can. It is a directional antenna, useful for short to medium distance point-to-point links. It may be also used as a feeder for a parabolic dish or grid.

Not all cans are good for building an antenna because there are dimensional constraints:

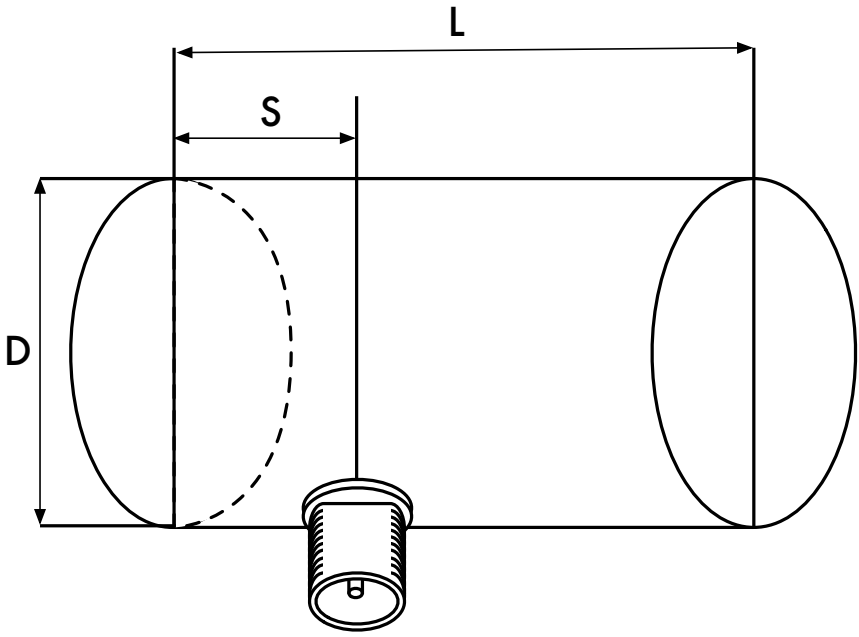

Figure 4.32: Dimensional constraints on the cantenna

1. The acceptable values for the diameter D of the feed are between 0.60 and 0.75 wavelength in air at the design frequency. At 2.44 GHz the wavelength λ is 12.2 cm, so the can diameter should be in the range of 7.3 - 9.2 cm.

2. The length L of the can preferably should be at least 0.75 λ_G , where λ_G is the guide wavelength and is given by:

$$\lambda_G = \frac{\lambda}{\text{sqrt}(1 - (\lambda / 1.706D)^2)}$$

For D = 7.3 cm, we need a can of at least 56.4 cm, while for D = 9.2 cm we need a can of at least 14.8 cm. Generally the smaller the diameter, the longer the can should be. For our example, we will use oil cans that have a diameter of 8.3 cm and a height of about 21 cm.

3. The probe for coaxial cable to waveguide transition should be positioned at a distance S from the bottom of the can, given by:

$$S = 0.25\ \lambda_G$$

Its length should be 0.25 λ, which at 2.44 GHz corresponds to 3.05 cm.

The gain for this antenna will be in the order of 10 to 14 dBi, with a beam-width of around 60 degrees.

Figure 4.33: The finished cantenna.

Parts list

- one screw-on N-type female connector
- 4 cm of copper or brass wire of 2 mm of diameter
- an oil can of 8.3 cm of diameter and 21 cm of height

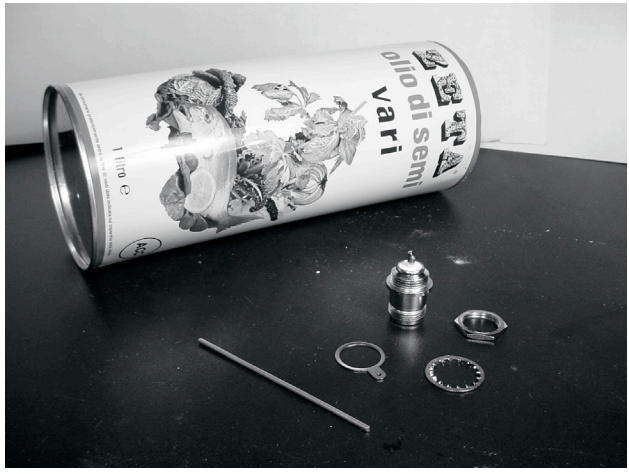

Figure 4.34: Parts needed for the can antenna.

Tools required

- Can opener
- Ruler
- Pliers
- File
- Soldering iron
- Solder
- Drill with a set of bits for metal (with a 1.5 cm diameter bit)
- Vice or clamp
- Spanner or monkey wrench
- Hammer
- Punch

Construction

1. With the can opener, remove carefully the upper part of the can.

Figure 4.35: Be careful of sharp edges when opening the can.

The circular disk has a very sharp edge. Be careful in handling it! Empty the can and wash it with soap. If the can contained pineapple, cookies, or some other tasty treat, have a friend serve the food.

2. With the ruler, measure 6.2 cm from the bottom of the can and draw a point. Be careful to measure from the inner side of the bottom. Use a punch (or a small drill bit or a Phillips screwdriver) and a hammer to mark

the point. This makes it easier to precisely drill the hole. Be careful not to change the shape of the can doing this by inserting a small block of wood or other object in the can before tapping it.

Figure 4.36: Mark the hole before drilling.

3. With a small diameter drill bit, make a hole at the center of the plate. Increase the diameter of the hole using bits with an increasing diameter. The hole should fit exactly the N connector. Use the file to smooth the border of the hole and to remove the painting around it in order to ensure a better electrical contact with the connector.

Figure 4.37: Carefully drill a pilot hole, then use a larger bit to finish the job.

4. Smooth with the file one end of the wire. Tin the wire for around 0.5 cm at the same end helping yourself with the vice.

Figure 4.38: Tin the end of the wire before soldering.

5. With the soldering iron, tin the central pin of the connector. Keeping the wire vertical with the pliers, solder its tinned side in the hole of the central pin.

Figure 4.39: Solder the wire to the gold cup on the N connector.

6. Insert a washer and gently screw the nut onto the connector. Trim the wire at 3.05 cm measured from the bottom part of the nut.

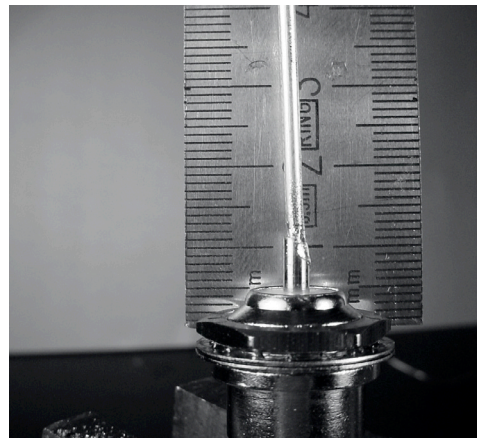

Figure 4.40: The length of the wire is critical.

7. Unscrew the nut from the connector, leaving the washer in place. Insert the connector into the hole of the can. Screw the nut on the connector from inside the can.

Figure 4.41: Assemble the antenna.

8. Use the pliers or the monkey wrench to screw firmly the nut on the connector. You are done!

Figure 4.42: Your finished cantenna.

As with the other antenna designs, you should make a weatherproof enclosure for the antenna if you wish to use it outdoors. PVC works well for the can antenna. Insert the entire can in a large PVC tube, and seal the ends with caps and glue. You will need to drill a hole in the side of the tube to accommodate the N connector on the side of the can.

Cantenna as dish feed

As with the USB dongle parabolic, you can use the cantenna design as a feeder for significantly higher gain. Mount the can on the parabolic with the opening of the can pointed at the center of the dish. Use the technique described in the USB dongle antenna example (watching signal strength changes over time) to find the optimum location of the can for the dish you are using.

By using a well-built cantenna with a properly tuned parabolic, you can achieve an overall antenna gain of 30dBi or more. As the size of the parabolic increases, so does the potential gain and directivity of the antenna. With very large parabolas, you can achieve significantly higher gain.

For example, in 2005, a team of college students successfully established a link from Nevada to Utah in the USA. The link crossed a distance of over 200 kilometers! The wireless enthusiasts used a 3.5 meter satellite dish to establish an 802.11b link that ran at 11Mbps, without using an amplifier. Details about this achievement can be found at *http://www.wifi-shootout.com/*

NEC2

NEC2 stands for ***Numerical Electromagnetics Code*** (version 2) and is a free antenna modeling package. NEC2 lets you build an antenna model in 3D, and then analyzes the antenna's electromagnetic response. It was developed more than ten years ago and has been compiled to run on many different computer systems. NEC2 is particularly effective for analyzing wire-grid models, but also has some surface patch modeling capability.

The antenna design is described in a text file, and then the model is built using this text description. An antenna described in NEC2 is given in two parts: its ***structure*** and a sequence of ***controls***. The structure is simply a numerical description of where the different parts of the antenna are located, and how the wires are connected up. The controls tell NEC where the RF source is connected. Once these are defined, the transmitting antenna is then modeled. Because of the reciprocity theorem the transmitting gain pattern is the same as the receiving one, so modeling the transmission characteristics is sufficient to understand the antenna's behaviour completely.

A frequency or range of frequencies of the RF signal must be specified. The next important element is the character of the ground. The conductivity of the earth varies from place to place, but in many cases it plays a vital role in determining the antenna gain pattern.

To run NEC2 on Linux, install the NEC2 package from the URL below. To launch it, type **`nec2`** and enter the input and output filenames. It is also worth installing the **`xnecview`** package for structure verification and radiation pattern plotting. If all went well you should have a file containing the output. This can be broken up into various sections, but for a quick idea of what it represents a gain pattern can be plotted using xnecview. You should see the expected pattern, horizontally omnidirectional, with a peak at the optimum angle of takeoff. Windows and Mac versions are also available.

The advantage of NEC2 is that we can get an idea of how the antenna works before building it, and how we can modify the design in order to get the maximum gain. It is a complex tool and requires some research to learn how to use it effectively, but it is an invaluable tool for antenna designers.

NEC2 is available from Ray Anderson's "Unofficial NEC Archives" at *http://www.si-list.org/swindex2.html*

Online documentation can be obtained from the "Unofficial NEC Home Page" at *http://www.nittany-scientific.com/nec/ .*

5
Networking Hardware

In the last couple of years, an unprecedented surge in interest in wireless networking hardware has brought a huge variety of inexpensive equipment to the market. So much variety, in fact, that it would be impossible to catalog every available component. In this chapter, we'll look at the sort of features and attributes that are desirable in a wireless component, and see several examples of commercial and DIY gear that has worked well in the past.

Wired wireless

With a name like "wireless", you may be surprised at how many wires are involved in making a simple point-to-point link. A wireless node consists of many components, which must all be connected to each other with appropriate cabling. You obviously need at least one computer connected to an Ethernet network, and a wireless router or bridge attached to the same network. Radio components need to be connected to antennas, but along the way they may need to interface with an amplifier, lightning arrestor, or other device. Many components require power, either via an AC mains line or using a DC transformer. All of these components use various sorts of connectors, not to mention a wide variety of cable types and thicknesses.

Now multiply those cables and connectors by the number of nodes you will bring online, and you may well be wondering why this stuff is referred to as "wireless". The diagram on the next page will give you some idea of the cabling required for a typical point-to-point link. Note that this diagram is not to scale, nor is it necessarily the best choice of network design. But it will introduce you to many common interconnects and components that you will likely encounter in the real world.

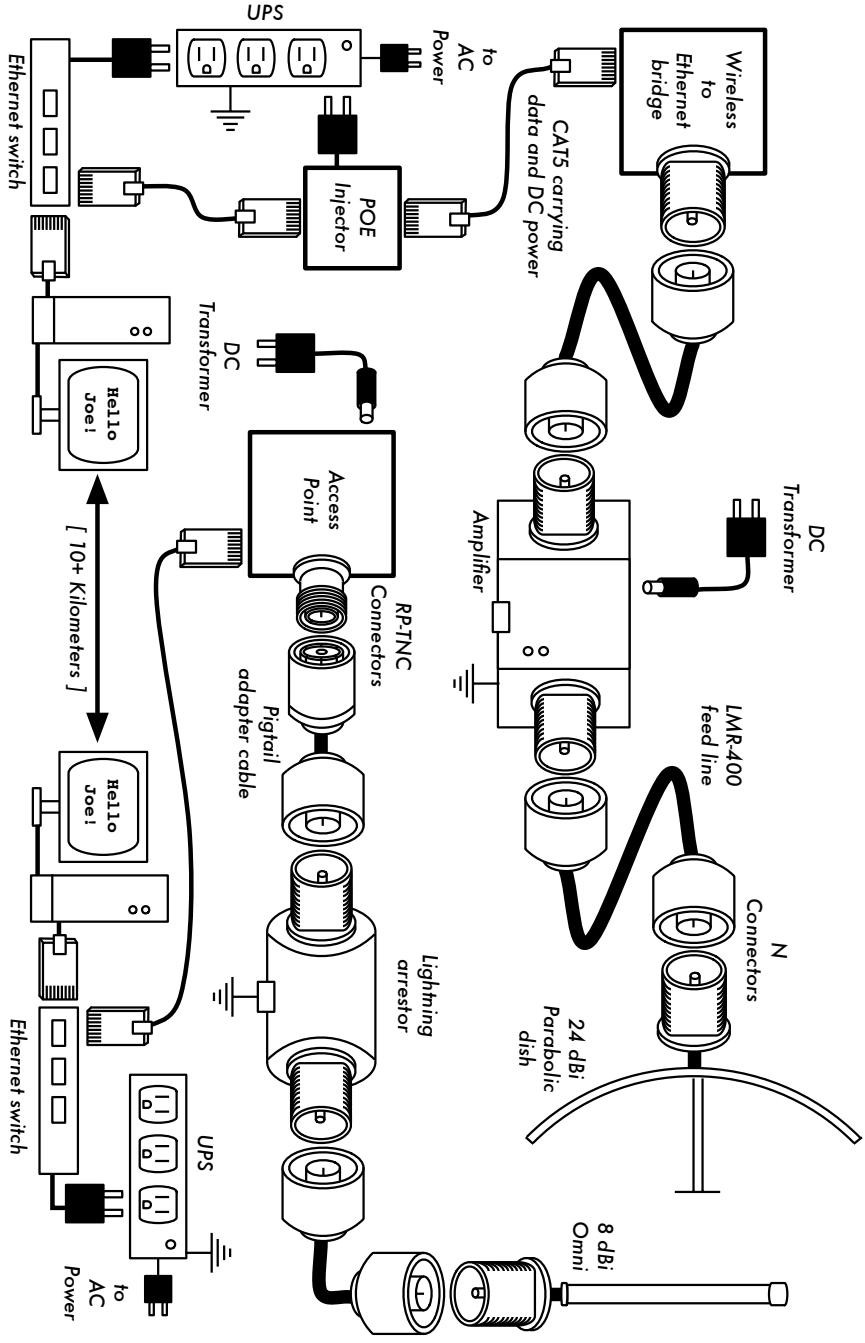

Figure 5.1: Component interconnects.

While the actual components used will vary from node to node, every installation will incorporate these parts:

1. An existing computer or network connected to an Ethernet switch.
2. A device that connects that network to a wireless device (a wireless router, bridge, or repeater).
3. An antenna that is connected via feed line, or is integrated into the wireless device itself.
4. Electrical components consisting of power supplies, conditioners, and lightning arrestors.

The actual selection of hardware should be determined by establishing the requirements for the project, determining the available budget, and verifying that the project is feasible using the available resources (including providing for spares and ongoing maintenance costs). As discussed in chapter one, establishing the scope of your project is critical before any purchasing decisions are made.

Choosing wireless components

Unfortunately, in a world of competitive hardware manufacturers and limited budgets, the price tag is the single factor that usually receives the most attention. The old saying that "you get what you pay for" often holds true when buying high tech equipment, but should not be considered an absolute truth. While the price tag is an important part of any purchasing decision, it is vital to understand precisely what you get for your money so you can make a choice that fits your needs.

When comparing wireless equipment for use in your network, be sure to consider these variables:

- **Interoperability.** Will the equipment you are considering work with equipment from other manufacturers? If not, is this an important factor for this segment of your network? If the gear in question supports an open protocol (such as 802.11b/g), then it will likely interoperate with equipment from other sources.
- **Range.** As we saw in chapter four, range is not something inherent in a particular piece of equipment. A device's range depends on the antenna connected to it, the surrounding terrain, the characteristics of the device at the other end of the link, and other factors. Rather than relying on a semi-fictional "range" rating supplied by the manufacturer, it is more useful to know the ***transmission power*** of the radio as well as the ***antenna gain*** (if

an antenna is included). With this information, you can calculate the theoretical range as described in chapter three.

- **Radio sensitivity.** How sensitive is the radio device at a given bit rate? The manufacturer should supply this information, at least at the fastest and slowest speeds. This can be used as a measure of the quality of the hardware, as well as allow you to complete a link budget calculation. As we saw in chapter three, a lower number is better for radio sensitivity.
- **Throughput.** Manufacturers consistently list the highest possible bit rate as the "speed" of their equipment. Keep in mind that the radio symbol rate (eg. 54Mbps) is never the actual throughput rating of the device (eg. about 22Mbps for 802.11g). If throughput rate information is not available for the device you are evaluating, a good rule of thumb is to divide the device "speed" by two, and subtract 20% or so. When in doubt, perform throughput testing on an evaluation unit before committing to purchasing a large amount of equipment that has no official throughput rating.
- **Required accessories.** To keep the initial price tag low, vendors often leave out accessories that are required for normal use. Does the price tag include all power adapters? (DC supplies are typically included; power over Ethernet injectors typically are not. Double-check input voltages as well, as equipment is often provided with a US-centric power supply). What about pigtails, adapters, cables, antennas, and radio cards? If you intend to use it outdoors, does the device include a weatherproof case?
- **Availability.** Will you be able to easily replace failed components? Can you order the part in large quantity, should your project require it? What is the projected life span of this particular product, both in terms of useful running time in-the-field and likely availability from the vendor?
- **Other factors.** Be sure that other needed features are provided for to meet your particular needs. For example, does the device include an external antenna connector? If so, what type is it? Are there user or throughput limits imposed by software, and if so, what is the cost to increase these limits? What is the physical form factor of the device? How much power does it consume? Does it support POE as a power source? Does the device provide encryption, NAT, bandwidth monitoring tools, or other features critical to the intended network design?

By answering these questions first, you will be able to make intelligent buying decisions when it comes time to choose networking hardware. It is unlikely that you will be able to answer every possible question before buying gear, but if you prioritize the questions and press the vendor to answer them before committing to a purchase, you will make the best use of your budget and build a network of components that are well suited to your needs.

Commercial vs. DIY solutions

Your network project will almost certainly consist of components purchased from vendors as well as parts that are sourced or even fabricated locally. This is a basic economic truth in most areas of the world. At this stage of human technology, global distribution of information is quite trivial compared to global distribution of goods. In many regions, importing every component needed to build a network is prohibitively expensive for all but the largest budgets. You can save considerable money in the short term by finding local sources for parts and labor, and only importing components that must be purchased.

Of course, there is a limit to how much work can be done by any individual or group in a given amount of time. To put it another way, by importing technology, you can exchange money for equipment that can solve a particular problem in a comparatively short amount of time. The art of building local telecommunications infrastructure lies in finding the right balance of money to effort needed to be expended to solve the problem at hand.

Some components, such as radio cards and antenna feed line, are likely far too complex to consider having them fabricated locally. Other components, such as antennas and towers, are relatively simple and can be made locally for a fraction of the cost of importing. Between these extremes lie the communication devices themselves.

By using off-the-shelf radio cards, motherboards, and other components, you can build devices that provide features comparable (or even superior) to most commercial implementations. Combining open hardware platforms with open source software can yield significant “bang for the buck” by providing custom, robust solutions for very low cost.

This is not to say that commercial equipment is inferior to a do-it-yourself solution. By providing so-called “turn-key solutions”, manufacturers not only save development time, but they can also allow relatively unskilled people to install and maintain equipment. The chief strengths of commercial solutions are that they provide ***support*** and a (usually limited) ***equipment warranty***. They also provide a ***consistent platform*** that tends to lead to very stable, often interchangeable network installations.

If a piece of equipment simply doesn’t work or is difficult to configure or troubleshoot, a good manufacturer will assist you. Should the equipment fail in normal use (barring extreme damage, such as a lightning strike) then the manufacturer will typically replace it. Most will provide these services for a limited time as part of the purchase price, and many offer support and warranty for an extended period for a monthly fee. By providing a consistent

platform, it is simple to keep spares on hand and simply "swap out" equipment that fails in the field, without the need for a technician to configure equipment on-site. Of course, all of this comes at comparatively higher initial cost for the equipment compared to off-the-shelf components.

From a network architect's point of view, the three greatest hidden risks when choosing commercial solutions are ***vendor lock-in***, ***discontinued product lines***, and ***ongoing licensing costs***.

It can be costly to allow the lure of ill-defined new "features" drive the development of your network. Manufacturers will frequently provide features that are incompatible with their competition by design, and then issue marketing materials to convince you that you simply cannot live without them (regardless of whether the feature contributes to the solution of your communications problem). As you begin to rely on these features, you will likely decide to continue purchasing equipment from the same manufacturer in the future. This is the essence of vendor lock-in. If a large institution uses a significant amount of proprietary equipment, it is unlikely that they will simply abandon it to use a different vendor. Sales teams know this (and indeed, some rely on it) and use vendor lock-in as a strategy for price negotiations.

When combined with vendor lock-in, a manufacturer may eventually decide to discontinue a product line, regardless of its popularity. This ensures that customers, already reliant on the manufacturer's proprietary features, will purchase the newest (and nearly always more expensive) model. The long term effects of vendor lock-in and discontinued products are difficult to estimate when planning a networking project, but should be kept in mind.

Finally, if a particular piece of equipment uses proprietary computer code, you may need to license use of that code on an ongoing basis. The cost of these licenses may vary depending on features provided, number of users, connection speed, or other factors. If the license fee is unpaid, some equipment is designed to simply stop working until a valid, paid-up license is provided! Be sure that you understand the terms of use for any equipment you purchase, including ongoing licensing fees.

By using generic equipment that supports open standards and open source software, you can avoid some of these pitfalls. For example, it is very difficult to become locked-in to a vendor that uses open protocols (such as TCP/IP over 802.11a/b/g). If you encounter a problem with the equipment or the vendor, you can always purchase equipment from a different vendor that will interoperate with what you have already purchased. It is for these reasons that we recommend using proprietary protocols and licensed spectrum **only** in cases where the open equivalent (such as 802.11a/b/g) is not technically feasible.

Likewise, while individual products can always be discontinued at any time, you can limit the impact this will have on your network by using generic components. For example, a particular motherboard may become unavailable on the market, but you may have a number of PC motherboards on hand that will perform effectively the same task. We will see some examples of how to use these generic components to build a complete wireless node later in this chapter.

Obviously, there should be no ongoing licensing costs involved with open source software (with the exception of a vendor providing extended support or some other service, without charging for the use of the software itself). There have occasionally been vendors who capitalize on the gift that open source programmers have given to the world by offering the code for sale on an ongoing licensed basis, thereby violating the terms of distribution set forth by the original authors. It would be wise to avoid such vendors, and to be suspicious of claims of "free software" that come with an ongoing license fee.

The disadvantage of using open source software and generic hardware is clearly the question of support. As problems with the network arise, you will need to solve those problems for yourself. This is often accomplished by consulting free online resources and search engines, and applying code patches directly. If you do not have team members who are competent and dedicated to designing a solution to your communications problem, then it can take a considerable amount of time to get a network project off the ground. Of course, there is never a guarantee that simply "throwing money at the problem" will solve it either. While we provide many examples of how to do much of the work yourself, you may find this work very challenging. You will need to find the balance of commercial solution and do-it-yourself approach that works for project.

In short, always define the scope of your network first, identify the resources you can bring to bear on the problem, and allow the selection of equipment to naturally emerge from the results. Consider commercial solutions as well as open components, while keeping in mind the long-term costs of both.

Professional wireless products

There is an abundance of equipment on the market for long distance, point-to-point (P2P) links. Most of this equipment is ready to go right out of the box, only the antenna cables need to be attached and sealed. When thinking about a long distance link, there are three main factors to consider: total link distance, uptime requirements, and of course, link speed requirements.

Most of the commonly available commercial products for longer range links now use OFDM technology and operate in the 5.8 GHz ISM band. There are

some products available that use open standards, but most use a proprietary protocol of some sort. This does mean that in order to form a link, the radios on both sides will have to be from the same manufacturer. For mission critical links it is a good idea to choose a system that uses the identical equipment on both sides of the link. This way only one spare unit needs to be stocked, and if need be, can replace either side of the link. There are some good products on the market that use different equipment at either end of a link. These can be used in a network as long as it is done with care, or else spares will need to be available in both kinds of radios.

This is not meant to be a sales pitch for any radio, or complaints about them either. These are just some notes that have come from more then five years of field experience all over the world with unlicensed commercial products. There is unfortunately no way to review every product, so some favorites are listed below.

Redline Communications

Redline first came to market with its AN-50 line of products. This was the first point-to-point product available with data rates above 50 Mbps that small operators could actually afford. They only use 20 MHz of spectrum per channel. There are three different models available in their AN-50 line. All three have the same basic feature sets, only the total bandwidth changes. The standard model has 36 Mbps throughput, the economy model has 18 Mbps, and the full version has 54 Mbps. The bandwidth controls are software upgradeable and can be added into the system as the demand for bandwidth increases.

Redline radios consist of an indoor unit, an outdoor unit, and an antenna. The indoor unit fits in a standard 19 inch rack, and occupies 1U. The outdoor unit mounts on the same bracket that holds the antenna in place. This outdoor unit is the actual radio. The two units are linked by a coax interface cable. Beldon RG6 or RG11 cable is used for this interface cable. This is the same cable used for satellite TV installations. It is inexpensive, easy to find, and eliminates the need for expensive low loss cable, like the Times Microwave LMR series or Andrew Corporation Heliax. Also, keeping the radio mounted so close to the antenna keeps the cable related loss to an absolute minimum.

There are two features to note on the Redline radios. The first is the ***General Alignment Mode***, which turns on a beeper that changes tone as the modulation technique changes. Faster beeping means a faster connection. This allows for a much easier alignment as the link can be mostly aligned by the tones alone. Only a final tuning will be needed, and a graphical Windows application is available to help with this. The other feature is a ***Test*** button. Whenever radio changes are made but are not sure to be correct, pressing

the test button instead of the ***Save*** button will make the new changes active for five minutes. After five minutes, the configuration reverts back to the setting before the test button was pushed. This allows the changes to be tried out, and if things don't work out and the link goes down, the link will come back after five minutes. Once the changes have been tried out, simply confirm the new settings in the configuration, and press the save button instead of the test button.

Redline has other models available. The AN-30 has four T1/E1 ports, in addition to a 30 Mbps Ethernet connection. The AN-100 follows the 802.16a standard, and the upcoming RedMax promises WiMax compliance.

For more information about Redline Communications products, see http://*www.redlinecommunications.com/*

Alvarion

One of the biggest advantages of working with Alvarion products is Alvarion's very well established worldwide distribution network. They also have one of the largest worldwide market shares for all kinds of wireless Internet connectivity hardware. There are distributors and resellers within most regions. For longer distance links there are two products of interest: The VL series, and the Link Blaster.

While the VL series is actually a point-to-multipoint system, a single client radio connecting to a single access point will function just fine for a point-to-point link. The only thing that should be considered is using a more directional antenna at the access point, unless there is a future link planned that could connect to that access point. There are two speeds available for the VL series, 24 Mbps and 6 Mbps. Budget, uptime, and speed requirements will guide the decision between which CPE to use.

The Link Blaster looks and feels a lot like a Redline AN-50. That's because it is one. Very soon after the Redline AN-50 came on the market, an OEM agreement between the two companies was signed, and the Link Blaster was born. Although the indoor unit is in a different case, and the antennas are marked differently, the electronics inside the units are identical. The Link Blaster does cost more then a Redline; this money buys you a more rugged design and an additional level of support. In many cases, an Alvarion reseller may be closer and easier to ship product from then some Redline resellers. This will be something that will have to be locally researched. It may be worth the extra money to have a product that is locally available and supported.

Alvarion does have some 2.4 GHz point-to-point products available. Most of their product range in the 2.4 GHz ISM band uses frequency hopping spread

spectrum (FHSS) and will create a lot of noise for local direct sequence spread spectrum (DSSS) on the same tower. If a DSSS based distribution system is being planned for, then a FHSS backhaul is not going to be an effective option.

For more information about Alvarion products, see *http://www.alvarion.com/*

Rad Data Communications

The Rad Airmux product line is relatively new to the market, and has some great potential. The Airmux 200 is a 48 Mbps radio, uses CAT5 cable, and comes with one of the most friendly price tags of any commercial solution. The units are small and easy to handle on a tower. The downside that may be found is a lack of a local distribution system in the developing world. There are two models available within the Airmux line. One uses internal antennas, and the other uses external antennas.

Experience with Airmux radios in early 2005 shows there is an issue in the timing configurations. This only becomes apparent when the link distance is more then 12 miles, or 19 km. It doesn't matter which antennas are being used. Until this bug is fixed, these radios should only be used for links under 19 km. When that guide is followed these radios preform very well, especially for their price point.

For more information about Rad Data Communications products, see *http://www.rad.com/*

Cisco Systems

Cisco wireless solutions have two big advantages to their credit. They have a very well established distribution, support, and training network throughout most of the world. There are distributors and resellers all over the place. This can be a big help when it comes time to procure equipment, and even more important if equipment breaks and needs replacing. The next big advantage is that for the most part, they use open standards. Most of their available equipment follows 802.11a/b/g standards.

Experience has shown that their web based configuration tools are not as easy to understand as those found in many other products, and the equipment tends to come with a price tag that makes other non-commercial, open standard solutions more viable.

More information about Cisco can be found at *http://www.cisco.com/*

Any others?

There are many more solutions available on the market now, and more arriving all of the time. Good solutions are available from companies like Trango Broadband (*http://www.trangobroadband.com/)* and Waverider Communications (*http://www.waverider.com/*). When considering which solution to use, always remember the three main factors; distance, uptime and speed. Be sure to check and make sure that the radios operate in an unlicensed band where you are installing them.

Professional lightning protection

The only natural predator of wireless equipment is lightning. There are two different ways lightning can strike or damage equipment: direct hits or induction hits. Direct hits are when lightning actually hits the tower or antenna. Induction hits are caused when lightning strikes near the tower. Imagine a negatively charged lightning bolt. Since like charges repel each other, that bolt will cause the electrons in the cables to move away from the strike, creating current on the lines. This is much more current the the sensitive radio equipment can handle. Either type of strike will usually destroy unprotected equipment.

Figure 5.2: A tower with a heavy copper grounding wire.

Protecting wireless networks from lightning is not an exact science, and there is no guarantee that a lightning strike will not happen, even if every single precaution is taken. Many of the methods used will help prevent both direct

and induction strikes. While it is not necessary to use every single lightning protection method, using more methods will help further protect the equipment. The amount of lightning historically observed within a service area will be the biggest guide to how much needs to be done.

Start at the very bottom of the tower. Remember, the bottom of the tower is below the ground. After the tower foundation is laid, but before the hole is backfilled, a ring of heavy braided ground wire should have been installed with the lead extending above ground surfacing near a tower leg. The wire should be American Wire Gauge (AWG) #4 or thicker. In addition, a backup ground or earthing rod should be driven into the ground, and a ground wire run from the rod to the lead from the buried ring.

It is important to note that not all steel conducts electricity the same way. Some types of steel act as better electrical conductors then others, and different surface coatings can also affect how tower steel handles electrical current. Stainless steel is one of the worst conductors, and rust proof coatings like galvanizing or paint lessen the conductivity of the steel. For this reason, a braided ground wire is run from the bottom of the tower all the way to the top. The bottom needs to be properly attached to the leads from both the ring and the backup ground rod. The top of the tower should have a lightning rod attached, and the top of that needs to be pointed. The finer and sharper the point, the more effective the rod will be. The braided ground wire from the bottom needs to be terminated at this grounding rod. It is very important to be sure that the ground wire is connected to the actual metal. Any sort of coating, such as paint, must be removed before the wire is attached. Once the connection is made, the exposed area can be repainted, covering the wire and connectors if necessary to save the tower from rust and other corrosion.

The above solution details the installation of the basic grounding system. It provides protection for the tower itself from direct hits, and installs the base system to which everything else will connect.

The ideal protection for indirect induction lightning strikes are gas tube arrestors at both ends of the cable. These arrestors need to be grounded directly to the ground wire installed on the tower if it is at the high end. The bottom end needs to be grounded to something electrically safe, like a ground plate or a copper pipe that is consistently full of water. It is important to make sure that the outdoor lightning arrestor is weatherproofed. Many arresters for coax cables are weatherproofed, while many arresters for CAT5 cable are not.

In the event that gas arrestors are not being used, and the cabling is coax based, then attaching one end of a wire to the shield of the cable and the other to the ground wire installed on the towers will provide some protection.

This can provide a path for induction currents, and if the charge is weak enough, it will not affect the conductor wire of the cable. While this method is by no means as good of protection as using the gas arrestors, it is better then doing nothing at all.

Building an AP from a PC

Unlike consumer operating systems (such as Microsoft Windows), the GNU/Linux operating system gives a network administrator the potential for full access to the networking stack. One can access and manipulate network packets at any level from the data-link layer through the application layer. Routing decisions can be made based on any information contained in a network packet, from the routing addresses and ports to the contents of the data segment. A Linux-based access point can act as a router, bridge, firewall, VPN concentrator, application server, network monitor, or virtually any other networking role you can think of. It is freely available software, and requires no licensing fees. GNU/Linux is a very powerful tool that can fill a broad variety of roles in a network infrastructure.

Adding a wireless card and Ethernet device to a PC running Linux will give you a very flexible tool that can help you deliver bandwidth and manage your network for very little cost. The hardware could be anything from a recycled laptop or desktop machine to an embedded computer, such as a Linksys WRT54G or Metrix networking kit.

In this section we will see how to configure Linux in the following configurations:

- As a wireless access point with Masquerading/NAT and a wired connection to the Internet (also referred to as a wireless gateway).
- As a wireless access point that acts as a transparent bridge. The bridge can be used either as a simple access point, or as a repeater with 2 radios.

Consider these recipes as a starting point. By building on these simple examples, you can create a server that fits precisely into your network infrastructure.

Prerequisites

Before proceeding, you should already be familiar with Linux from a users perspective, and be capable of installing the Gnu/Linux distribution of your choice. A basic understanding of the command line interface (terminal) in Linux is also required.

You will need a computer with one or more wireless cards already installed, as well as a standard Ethernet interface. These examples use a specific card and driver, but there are a number of different cards that should work equally well. Wireless cards based on the Atheros and Prism chipsets work particularly well. These examples are based on Ubuntu Linux version 5.10 (Breezy Badger), with a wireless card that is supported by the HostAP or MADWiFi drivers. For more information about these drivers, see *http://hostap.epitest.fi/* and *http://madwifi.org/* .

The following software is required to complete these installations. It should be provided in your Linux distribution:

- Wireless Tools (iwconfig, iwlist commands)
- iptables firewall
- dnsmasq (caching DNS server and DHCP server)

The CPU power required depends on how much work needs to be done beyond simple routing and NAT. For many applications, a 133MHz 486 is perfectly capable of routing packets at wireless speeds. If you intend to use a lot of encryption (such as WEP or a VPN server), then you will need something faster. If you also want to run a caching server (such as Squid, see chapter three) then you will need a computer with plenty of fast disk space and RAM. A typical router that is only performing NAT will operate will with as little as 64MB of RAM and storage.

When building a machine that is intended to be part of your network infrastructure, keep in mind that hard drives have a limited lifespan compared to most other components. You can often use solid state storage, such as a flash disk, in place of a hard drive. This could be a USB flash drive (assuming your PC will boot from USB), or a Compact Flash card using a CF to IDE adapter. These adapters are quite inexpensive, and will make a CF card appear act like standard IDE hard drive. They can be used in any PC that supports IDE hard drives. Since they have no moving parts, they will operate for many years through a much wider range of temperatures than a hard disk will tolerate.

Scenario 1: Masquerading access point

This is the simplest of the scenarios, and is especially useful in situations where you want a single access point for an office setting. This is easiest in a situation where:

1. There is an existing dedicated firewall and gateway running Linux, and you just want to add a wireless interface.

2. You have an old refurbished computer or laptop available, and prefer to use that as an access point.

3. You require more power in terms of monitoring, logging and/or security than most commercial access points provide, but don't want to splurge on an enterprise access point.

4. You would like a single machine to act as 2 access points (and firewall) so that you can offer both a secure network access to the intranet, as well as open access to guests.

Initial setup

Start of with an already configured computer running GNU/Linux. This could be an Ubuntu Server installation, or Fedora Core. The computer must have at least 2 interfaces for this to work, and at least one of these interfaces should be wireless. The rest of this description assumes that your cabled Ethernet port (eth0) is connected to the Internet, and that there is a wireless interface (wlan0) that will provide the access point functionality.

To find out if your chipset supports master mode, try the following command as root:

```
# iwconfig wlan0 mode Master
```

...replacing wlan0 with the name of your interface.

If you get an error message, then your wireless card doesn't support access point mode. You can still try the same setup in Ad-hoc mode, which is supported by all chipsets. This requires that you to set all the laptops that are connecting to this "access point" into Ad-hoc mode as well, and may not work quite the way you are expecting. It is usually better to find a wireless card that will support AP mode. See the HostAP and MADWiFi websites mentioned earlier for a list of supported cards.

Before continuing, make sure dnsmasq is installed on your machine. You can use the graphical package manager of your distribution to install it. In Ubuntu you can simply run the following as root:

```
# apt-get install dnsmasq
```

Setting up the interfaces

Set up your server so that eth0 is connected to the Internet. Use the graphical configuration tool that came with your distribution.

If your Ethernet network uses DHCP, you could try the following command as root:

```
# dhclient eth0
```

You should receive an IP address and default gateway. Next, set your wireless interface to Master mode and give it a name of your choice:

```
# iwconfig wlan0 essid "my network" mode Master enc off
```

The **`enc off`** switch turns off WEP encryption. To enable WEP, add a hex-key string of the correct length:

```
# iwconfig wlan0 essid "my network" mode Master enc 1A2B3C4D5E
```

Alternately, you can use a readable string by starting with "s:"

```
# iwconfig wlan0 essid "my network" mode Master enc "s:apple"
```

Now give your wireless interface an IP address in a private subnet, but make sure it is not the same subnet as that of your Ethernet adapter:

```
# ifconfig wlan0 10.0.0.1 netmask 255.255.255.0 broadcast 10.0.0.255 up
```

Setting up masquerading in the kernel

In order for us to be able to translate addresses between the two interfaces on the computer, we need to enable masquerading (NAT) in the linux kernel. First we load the relevant kernel module:

```
# modprobe ipt_MASQUERADE
```

Now we will flush all existing firewall rules to ensure that the firewall is not blocking us from forwarding packets between the two interfaces. If you have an existing firewall running, make sure you know how to restore the existing rules later before proceeding.

```
# iptables -F
```

Enable the NAT functionality between the two interfaces

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

Finally we need to enable the kernel to forward packets between interfaces:

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

On Debian-based Linux distributions such as Ubuntu, this change can also be made by editing the file **/etc/network/options**, and changing the line

```
ip_forward=no
```

to

```
ip_forward=yes
```

and then restarting the network interfaces with:

```
# /etc/init.d/network restart
```

or

```
# /etc/init.d/networking restart
```

Setting up the DHCP server

At this point we actually should have a working access point. It can be tested by connecting to the wireless network "my network" with a separate machine and giving that machine an address in the same address range as our wireless interface on the server (10.0.0.0/24 if you followed the examples). If you have enabled WEP, be sure to use the same key that you specified on the AP.

In order to make it easier for people to connect to the server without knowing the IP address range, we will set up a DHCP server to automatically hand out addresses to wireless clients.

We use the program dnsmasq for this purpose. As the name indicates, it provides a caching DNS server as well as a DHCP server. This program was developed especially for use with firewalls performing NAT. Having a caching DNS server is especially helpful if your Internet connection is a high-latency and/or low-bandwidth connection, such as a VSAT or dial-up. It means that many DNS queries can be resolved locally, saving a lot of traffic on the Internet connection, and also making the connection feel noticeably faster for those connecting.

Install dnsmasq with your distributions package manager. If dnsmasq is not available as a package, download the source code and install it manually. It is available from *http://thekelleys.org.uk/dnsmasq/doc.html*.

All that is required for us to run dnsmasq is to edit a few lines of the dnsmasq configuration file, **`/etc/dnsmasq.conf`**.

The configuration file is well commented, and has many options for various types of configuration. To get the basic DHCP server up and running we just need to uncomment and/or edit two lines.

Find the lines that starts:

```
interface=
```

...and make sure it reads:

```
interface=wlan0
```

...changing wlan0 to match name of your wireless interface. Then find the line that starts with:

```
#dhcp-range=
```

Uncomment the line and edit it to suit the match addresses being used, i.e.

```
dhcp-range=10.0.0.10,10.0.0.110,255.255.255.0,6h
```

Then save the file and start dnsmasq:

```
# /etc/init.d/dnsmasq start
```

That's it, you should now be able to connect to the server as an access point, and get an IP address using DHCP. This should let you connect to the Internet through the server.

Adding extra security: Setting up a Firewall

Once this is set up and tested, you can add extra firewall rules using whatever firewall tool is included in your distribution. Some typical front-ends for setting up firewall rules include:

- ***firestarter*** - a graphical client for Gnome, which requires that your server is running Gnome
- ***knetfilter*** – a graphical client for KDE, which requires that your server is running KDE
- ***Shorewall*** – a set of scripts and configuration files that will make it easier to setup an iptables firewall. There are also frontends for shorewall, such as webmin-shorewall
- ***fwbuilder*** - a powerful, but slightly complex graphical tool that will let you create iptables scripts on a machine separate from your server, and then transfer them to the server later. This does not require you to be running a graphical desktop on the server, and is a strong option for the security conscious.

Once everything is configured properly, make sure that all settings are reflected in the system startup scripts. This way, your changes will continue to work should the machine need to be rebooted.

Scenario 2: Transparent Bridging access point

This scenario can either be used for a two-radio repeater, or for an access point connected to an Ethernet. We use a bridge instead of routing when we want both interfaces on the access point to share the same subnet. This can be particularly useful in networks with multiple access points where we prefer to have a single, central firewall and perhaps authentication server. Because all clients share the same subnet they, can easily be managed with a single DHCP server and firewall without the need for DHCP relay.

For example, you could setup a server as the first scenario, but use two wired Ethernet interfaces instead of one wired and one wireless. One interface would be your Internet connection, and the other would connect to a switch. Then connect as many access points as you require to the same switch, set them up as transparent bridges, and everyone will pass through the same firewall and use the same DHCP server.

The simplicity of bridging comes at a cost of efficiency. Since all clients share the same subnet, broadcast traffic will be repeated throughout the network. This is usually fine for small networks, but as the number of clients increases, more wireless bandwidth will be wasted on broadcast network traffic.

Initial setup

The initial setup for a bridging access point is similar to that of a masquerading access point, without the requirement of dnsmasq. Follow the initial setup instructions from the previous example.

In addition, the ***bridge-utils*** package is required for bridging. This package exists for Ubuntu and other Debian-based distributions, as well as for Fedora Core. Make sure it is installed and that the command **`brctl`** is available before proceeding.

Setting up the Interfaces

On Ubuntu or Debian we set up the interfaces by editing the file **`/etc/network/interfaces`**

Add a section like the following, but change the names of interfaces and the IP addresses accordingly. The IP address and netmask must match that of your existing network. This example assumes you are building a wireless

repeater with two wireless interfaces, wlan0 and wlan1. The wlan0 interface will be a client to the "office" network, and wlan1 will create a network called "repeater".

Add the following to **/etc/network/interfaces**:

```
auto br0
iface br0 inet static
  address 192.168.1.2
  network 192.168.1.0
  netmask 255.255.255.0
  broadcast 192.168.1.255
  gateway 192.168.1.1
  pre-up ifconfig wlan 0 0.0.0.0 up
  pre-up ifconfig wlan1 0.0.0.0 up
  pre-up iwconfig wlan0 essid "office" mode Managed
  pre-up iwconfig wlan1 essid "repeater" mode Master
  bridge_ports wlan0 wlan1
  post-down ifconfig wlan1 down
  post-down ifconfig wlan0 down
```

Comment out any other sections in the file that refer to wlan0 or wlan1 to make sure that they don't interfere with our setup.

This syntax for setting up bridges via the **interfaces** file is specific to Debian-based distributions, and the details of actually setting up the bridge are handled by a couple of scripts: **/etc/network/if-pre-up.d/bridge** and **/etc/network/if-post-down.d/bridge**. The documentation for these scripts is found in **/usr/share/doc/bridge-utils/**.

If those scripts don't exist on your distribution (such as Fedora Core), here is an alternative setup for **/etc/network/interfaces** which will achieve the same thing with only marginally more hassle:

```
iface br0 inet static
  pre-up ifconfig wlan 0 0.0.0.0 up
  pre-up ifconfig wlan1 0.0.0.0 up
  pre-up iwconfig wlan0 essid "office" mode Managed
  pre-up iwconfig wlan1 essid "repeater" mode Master
  pre-up brctl addbr br0
  pre-up brctl addif br0 wlan0
  pre-up brctl addif br0 wlan1
  post-down ifconfig wlan1 down
  post-down ifconfig wlan0 down
  post-down brctl delif br0 wlan0
  post-down brctl delif br0 wlan1
  post-down brctl delbr br0
```

Starting the bridge

Once the bridge is defined as an interface, starting the bridge is as simple as typing:

```
# ifup -v br0
```

The "-v" means verbose output and will give you information to what is going on.

On Fedora Core (i.e. non-debian distributions) you still need to give your bridge interface an ip address and add a default route to the rest of the network:

```
#ifconfig br0 192.168.1.2 netmask 255.255.255.0 broadcast 192.168.1.255
#route add default gw 192.168.1.1
```

You should now be able to connect a wireless laptop to this new access point, and connect to the Internet (or at least to the rest of your network) through this box.

If you want more information about what your bridge and what it is doing, take a look at the brctl command. For example try this command:

```
# brctl show br0
```

That should show you some information about what your bridge is doing.

Scenario 1 & 2 the easy way

Instead of setting up your computer as an access point from scratch, you may wish to use a dedicated Linux distribution that is specially tailored for this purpose. These distributions can make the job as simple as booting from a particular CD on a computer with a wireless interface. See the following section, "Wireless-friendly operating systems" for more information.

As you can see, it is straightforward to provide access point services from a standard Linux router. Using Linux gives you significantly more control over how packets are routed through your network, and allows for features that simply aren't possible on consumer grade access point hardware.

For example, you could start with either of the above two examples and implement a private wireless network where users are authenticated using a standard web browser. Using a captive portal such as Chillispot, wireless users can be checked against credentials in an existing database (say, a Windows domain server accessible via RADIUS). This arrangement could

allow for preferential access to users in the database, while providing a very limited level of access for the general public.

Another popular application is the prepaid commercial model. In this model, users must purchase a ticket before accessing the network. This ticket provides a password that is valid for a limited amount of time (typically one day). When the ticket expires, the user must purchase another. This ticketing feature is only available on relatively expensive commercial networking equipment, but can be implemented using free software such as Chillispot and phpMyPrePaid. We will see more about captive portal technology and ticketing systems in the **Authentication** section in chapter six.

Wireless-friendly operating systems

There are a number of open source operating system that provide useful tools for working with wireless networks. These are intended to be used on repurposed PCs or other networking hardware (rather than on a laptop or server) and are fine-tuned for building wireless networks. Some of these projects include:

- **Freifunk**. Based on the OpenWRT project (*http://openwrt.org/*), the Freifunk firmware brings easy OLSR support to MIPS-based consumer access points, such as the Linksys WRT54G / WRT54GS / WAP54G, Siemens SE505, and others. By simply flashing one of these APs with the Freifunk firmware, you can rapidly build a self-forming OLSR mesh. Freifunk is not currently available for x86 architecture machines. It is maintained by Sven Ola of the Freifunk wireless group in Berlin. You can download the firmware from *http://www.freifunk.net/wiki/FreifunkFirmware* .
- **Metrix Pebble**. The Pebble Linux project was started in 2002 by Terry Schmidt of the NYCwireless group. It was originally a stripped-down version of the Debian Linux distribution that included wireless, firewall, traffic management, and routing tools. Since 2004, Metrix Communication has been extending Pebble to include updated drivers, bandwidth monitoring, and a web-based configuration tool. The aim of Metrix Pebble is to provide a complete platform for wireless development. It works on x86 hardware with at least 64MB of flash or hard disk storage. You can download Metrix Pebble from *http://metrix.net/metrix/howto/metrix-pebble.html .*
- **m0n0wall**. Based on FreeBSD, m0n0wall is a very tiny but complete firewall package that provides AP services. It is configured from a web interface and the entire system configuration is stored in a single XML file. Its tiny size (less than 6MB) makes it attractive for use in very small embedded systems. Its goal is to provide a secure firewall, and as such does not include userspace tools (it is not even possible to log into the machine over the network). Despite this limitation, it is a popular choice for wireless net-

workers, particularly those with a background in FreeBSD. You can download m0n0wall from *http://www.m0n0.ch/ .*

All of these distributions are designed to fit in machines with limited storage. If you are using a very large flash disk or hard drive, you can certainly install a more complete OS (such as Ubuntu or Debian) and use the machine as a router or access point. It will likely take a fair amount of development time to be sure all needed tools are included, without installing unnecessary packages. By using one of these projects as a starting point for building a wireless node, you will save yourself considerable time and effort.

The Linksys WRT54G

One of the most popular consumer access points currently on the market is the Linksys WRT54G. This access point features two external RP-TNC antenna connectors, a four port Ethernet switch, and an 802.11b/g radio. It is configured through a simple web interface. While it is not designed as an outdoor solution, it can be installed in a large sprinkler box or plastic tub for relatively little cost. As of this writing, the WRT54G sells for about $60.

Back in 2003, network hackers realized that the firmware that shipped with the WRT54G was actually a version of Linux. This led to a tremendous interest in building custom firmware that extended the capabilities of the router significantly. Some of these new features include client radio mode support, captive portals, and mesh networking. Two popular alternative firmware packages for the WRT54G are OpenWRT (*http://openwrt.org/*) and Freifunk (*http://www.freifunk.net/wiki/FreifunkFirmware*).

Unfortunately, in the fall of 2005, Linksys released version 5 of the WRT54G. This hardware revision eliminated some RAM and flash storage on the motherboard, making it practically impossible to run Linux (it ships with VxWorks, a much smaller operating system that does not allow easy customization). Since the WRT54G v5 cannot run custom Linux-based firmware, this makes it a less attractive alternative for network builders. Linksys has also released the WRT54GL, which is essentially the WRT54G v4 (which runs Linux) at a slightly higher price tag.

A number of other Linksys access points also run Linux, including the WRT54GS and WAP54G. While these also have relatively low price tags, the hardware specifications may change at any time. It is difficult to know which hardware revision is used without opening the packaging, making it risky to purchase them at a retail store and practically impossible to order online. While the WRT54GL is guaranteed to run Linux, Linksys has made it known that it does not expect to sell this model in large volume, and it is unclear how long it will be offered for sale.

If you can find a source of earlier revision WRT54Gs or WRT54GLs, they are handy and inexpensive routers. With custom firmware, they can be configured to work as an OLSR mesh or in client mode, and work very well as a cheap customer side solution. While the newer v5 model will work as an access point, it cannot be configured is a client, and it has mixed performance reviews compared to the v4 and earlier models.

For more information, see one of these websites:

- *http://linksysinfo.org/*
- *http://seattlewireless.net/index.cgi/LinksysWrt54g*

6
Security

In a traditional wired network, access control is very straightforward: If a person has physical access to a computer or network hub, then they can use (or abuse) the network resources. While software mechanisms are an important component of network security, limiting physical access to the network devices is the ultimate access control mechanism. Simply put, if all terminals and network components are only accessible to trusted individuals, then the network can likely be trusted.

The rules change significantly with wireless networks. While the apparent range of your access point may seem to be just a few hundred meters, a user with a high gain antenna may be able to make use of the network from several blocks away. Should an unauthorized user be detected, is impossible to simply "trace the cable" back to the user's location. Without transmitting a single packet, a nefarious user can even log all network data to disk. This data can later be used to launch a more sophisticated attack against the network. Never assume that radio waves simply "stop" at the edge of your property line.

Of course, even in wired networks, it's never quite possible to completely trust all users of the network. Disgruntled employees, uneducated network users, and simple mistakes on the part of honest users can cause significant harm to network operations. As the network architect, your goal should be to facilitate private communication between legitimate users of the network. While a certain amount of access control and authentication is necessary in any network, you have failed in your job if legitimate users find it difficult to use the network to communicate.

There's an old saying that the only way to completely secure a computer is to unplug it, lock it in a safe, destroy the key, and bury the whole thing in concrete. While such a system might be completely "secure", it is useless for

communication. When you make security decisions for your network, remember that above all else, the network exists so that its users can communicate with each other. Security considerations are important, but should not get in the way of the network's users.

Physical security

When installing a network, you are building an infrastructure that people will depend on. And thus, the network must be reliable. For many installations, outages often occur due to human tampering, accidental or not. Networks are physical, wires and boxes, things that are easily disturbed. In many installations, people will not know what the equipment is that you have installed, or, curiosity leads them to experiment. They will not realize the importance that a cable goes to a port. People might move an Ethernet cable so that they can connect their laptop for 5 minutes, or move a switch because it is in their way. A plug might be removed from a power bar because someone needs that receptacle. Assuring the physical security of an installation is paramount. Signs and labels will only be useful to few, whom can read, or speak your language. Putting things out of the way, and limiting access is the best means to assure that accidents, or tinkering does not occur.

In less developed economies proper fasteners, ties, or boxes will not be as easy to find. You should be able to find electrical supplies that will work just as well. Custom enclosures are also easy to manufacture and should be considered essential to any installation. It is often economical to pay a mason to make holes and install conduit, where this would be an expensive option in the developed world, this type of labour intensive activity can be affordable in Southern countries. PVC can be embedded in cement walls for passing cable from room to room, this avoids smashing holes every time a cable needs to be passed. To insulate, plastic bags can be stuffed into the conduit around the cables.

Small equipment should be mounted on the wall and larger equipment should be put in a closet or in a cabinet.

Switches

Switches, hubs or interior access points can with a wall plug be screwed directly onto a wall. Best to put this equipment as high as possible to reduce the chance that someone will touch the device or its cables.

Cables

Cables should be hidden and fastened. Better to bury cables, than to leave them hanging across a yard, where it might be used for drying clothes, or

simply snagged by a ladder etc. To avoid vermin and insects find plastic electrical conduit. The marginal expense will be well worth the trouble. The conduit should be buried about 30cm deep (below the frost in cold climates). It is also worth buying larger conduit than is presently required, so that future cables can be run through the same tubing. It is also possible to find plastic cable conduit that can be used in buildings. If not, simple cable attachments, nailed into the wall can be used to secure the cable and to make sure that it doesn't hang where it can be snagged, pinched or cut.

Power

It is best to have power bars locked in a cabinet. If that is not possible, Mount the power bar under a desk, or on the wall and use duct tape (gaffer tape, a strong adhesive tape) to secure the plug into the receptacle. On the UPS and power bar, do not leave empty receptacles, tape them if necessary. People will have the tendency to use the easiest receptacle, so make these critical ones difficult to use. If you do not, you might find a fan or light plugged into your UPS; though it is nice to have light, it is nicer to keep your server running!

Water

Protect your equipment from water and moisture. In all cases make sure that your equipment, including your UPS is at least 30cm from the ground, to avoid flooding. Also try to have a roof over your equipment, so that water and moisture will not fall onto it. In moist climates, it is important that the equipment has proper ventilation to assure that moisture can be exhausted. Small closets need to have ventilation, or moisture and heat can degrade or destroy your gear.

Masts

Equipment installed on a mast is often safe from thieves. Nonetheless, to deter thieves and to keep your equipment safe from winds it is good to overengineer mounts. Equipment should be painted a dull, white or grey colour to reflect the sun and to make it look plain and uninteresting. Panel antennas are much more subtle and less interesting than dishes and thus should be preferred. Any installation on walls, should require a ladder to reach. Try choosing well lit but not prominent places to put equipment. Also avoid antennae that resemble television antennae, as those are items that will attract interest by thieves, where a wifi antenna will be useless to the average thief.

Threats to the network

One critical difference between Ethernet and wireless is that wireless networks are built on a ***shared medium***. They more closely resemble the old network hubs than modern switches, in that every computer connected to the network can "see" the traffic of every other user. To monitor all network traffic on an access point, one can simply tune to the channel being used, put the network card into monitor mode, and log every frame. This data might be directly valuable to an eavesdropper (including data such as email, voice data, or online chat logs). It may also provide passwords and other sensitive data, making it possible to compromise the network even further. As we'll see later in this chapter, this problem can be mitigated by the use of encryption.

Another serious problem with wireless networks is that its users are relatively ***anonymous***. While it is true that every wireless device includes a unique MAC address that is supplied by the manufacturer, these addresses can often be changed with software. Even given the MAC address, it can be very difficult to judge where a wireless user is physically located. Multipath effects, high gain antennas, and widely varying radio transmitter characteristics can make it impossible to determine if a malicious wireless user is sitting in the next room or is in an apartment building a mile away.

While unlicensed spectrum provides a huge cost savings to the user, it has the unfortunate side effect that ***denial of service*** (***DoS***) attacks are trivially simple. By simply turning on a high powered access point, cordless phone, video transmitter, or other 2.4GHz device, a malicious person could cause significant problems on the network. Many network devices are vulnerable to other forms of denial of service attacks as well, such as disassociation flooding and ARP table overflows.

Here are several categories of individuals who may cause problems on a wireless network:

- ***Unintentional users***. As more wireless networks are installed in densely populated areas, it is common for laptop users to accidentally associate to the wrong network. Most wireless clients will simply choose any available wireless network when their preferred network is unavailable. The user may then make use of this network as usual, completely unaware that they may be transmitting sensitive data on someone else's network. Malicious people may even take advantage of this by setting up access points in strategic locations, to try to attract unwitting users and capture their data.

  The first step in avoiding this problem is educating your users, and stressing the importance of connecting only to known and trusted networks. Many wireless clients can be configured to only connect to trusted net-

works, or to ask permission before joining a new network. As we will see later in this chapter, users can safely connect to open public networks by using strong encryption.

- ***War drivers***. The "war driving" phenomenon draws its name from the popular 1983 hacker film, "War Games". War drivers are interested in finding the physical location of wireless networks. They typically drive around with a laptop, GPS, and omnidirectional antenna, logging the name and location of any networks they find. These logs are then combined with logs from other war drivers, and are turned into graphical maps depicting the wireless "footprint" of a particular city.

  The vast majority of war drivers likely pose no direct threat to networks, but the data they collect might be of interest to a network cracker. For example, it might be obvious that an unprotected access point detected by a war driver is located inside a sensitive building, such as a government or corporate office. A malicious person could use this information to illegally access the network there. Arguably, such an AP should never have been set up in the first place, but war driving makes the problem all the more urgent. As we will see later in this chapter, war drivers who use the popular program NetStumbler can be detected with programs such as Kismet. For more information about war driving, see sites such as *http://www.wifimaps.com/*, *http://www.nodedb.com/*, or *http://www.netstumbler.com/*.

- ***Rogue access points***. There are two general classes of rogue access points: those incorrectly installed by legitimate users, and those installed by malicious people who intend to collect data or do harm to the network. In the simplest case, a legitimate network user may want better wireless coverage in their office, or they might find security restrictions on the corporate wireless network too difficult to comply with. By installing an inexpensive consumer access point without permission, the user opens the entire network up to potential attacks from the inside. While it is possible to scan for unauthorized access points on your wired network, setting a clear policy that prohibits them is very important.

  The second class of rogue access point can be very difficult to deal with. By installing a high powered AP that uses the same ESSID as an existing network, a malicious person can trick people into using their equipment, and log or even manipulate all data that passes through it. Again, if your users are trained to use strong encryption, this problem is significantly reduced.

- ***Eavesdroppers***. As mentioned earlier, eavesdropping is a very difficult problem to deal with on wireless networks. By using a passive monitoring tool (such as Kismet), an eavesdropper can log all network data from a great distance away, without ever making their presence known. Poorly

encrypted data can simply be logged and cracked later, while unencrypted data can be easily read in real time.

If you have difficulty convincing others of this problem, you might want to demonstrate tools such as Etherpeg (*http://www.etherpeg.org/*) or Driftnet (*http://www.ex-parrot.com/~chris/driftnet/*). These tools watch a wireless network for graphical data, such as GIF and JPEG files. While other users are browsing the Internet, these tools simply display all graphics found in a graphical collage. I often use tools such as this as a demonstration when lecturing on wireless security. While you can tell a user that their email is vulnerable without encryption, nothing drives the message home like showing them the pictures they are looking at in their web browser.

Again, while it cannot be completely prevented, proper application of strong encryption will discourage eavesdropping.

This introduction is intended to give you an idea of the problems you are up against when designing a wireless network. Later in this chapter, we will look at tools and techniques that will help you to mitigate these problems.

Authentication

Before being granted access to network resources, users should first be ***authenticated***. In an ideal world, every wireless user would have an identifier that is unique, unchangeable, and cannot be impersonated by other users. This turns out to be a very difficult problem to solve in the real world.

The closest feature we have to a unique identifier is the MAC address. This is the 48-bit number assigned by the manufacturer to every wireless and Ethernet device. By employing ***mac filtering*** on our access points, we can authenticate users based on their MAC address. With this feature, the access point keeps an internal table of approved MAC addresses. When a wireless user tries to associate to the access point, the MAC address of the client must be on the approved list, or the association will be denied. Alternately, the AP may keep a table of known "bad" MAC addresses, and permit all devices that are not on the list.

Unfortunately, this is not an ideal security mechanism. Maintaining MAC tables on every device can be cumbersome, requiring all client devices to have their MAC addresses recorded and uploaded to the APs. Even worse, MAC addresses can often be changed in software. By observing MAC addresses in use on a wireless network, a determined attacker can "spoof" an approved MAC address and successfully associate to the AP. While MAC filtering will prevent unintentional users and even most curious individuals from accessing the network, MAC filtering alone cannot prevent attacks from determined attackers.

MAC filters are useful for temporarily limiting access from misbehaving clients. For example, if a laptop has a virus that sends large amounts of spam or other traffic, its MAC address can be added to the filter table to stop the traffic immediately. This will buy you time to track down the user and fix the problem.

Another popular authentication feature of wireless the so-called ***closed network***. In a typical network, APs will broadcast their ESSID many times per second, allowing wireless clients (as well as tools such as NetStumbler) to find the network and display its presence to the user. In a closed network, the AP does not beacon the ESSID, and users must know the full name of the network before the AP will allow association. This prevents casual users from discovering the network and selecting it in their wireless client.

There are a number of drawbacks to this feature. Forcing users to type in the full ESSID before connecting to the network is error prone and often leads to support calls and complaints. Since the network isn't obviously present in site survey tools like NetStumbler, this can prevent your networks from showing up on war driving maps. But it also means that other network builders cannot easily find your network either, and specifically won't know that you are already using a given channel. A conscientious neighbor may perform a site survey, see no nearby networks, and install their own network on the same channel you are using. This will cause interference problems for both you and your neighbor.

Finally, using closed networks ultimately adds little to your overall networks security. By using passive monitoring tools (such as Kismet), a skilled user can detect frames sent from your legitimate clients to the AP. These frames necessarily contain the network name. A malicious user can then use this name to associate to the access point, just like a normal user would.

Encryption is probably the best tool we have for authenticating wireless users. Through strong encryption, we can uniquely identify a user in a manner that is very difficult to spoof, and use that identity to determine further network access. Encryption also has the benefit of adding a layer of privacy by preventing eavesdroppers from easily watching network traffic.

The most widely employed encryption method on wireless networks is ***WEP encryption***. WEP stands for ***wired equivalent privacy***, and is supported by virtually all 802.11a/b/g equipment. WEP uses a shared 40-bit key to encrypt data between the access point and client. The key must be entered on the APs as well as on each of the clients. With WEP enabled, wireless clients cannot associate with the AP until they use the correct key. An eavesdropper listening to a WEP-enabled network will still see traffic and MAC addresses, but the data payload of each packet is encrypted. This provides a fairly good authentication mechanism while also adding a bit of privacy to the network.

WEP is definitely not the strongest encryption solution available. For one thing, the WEP key is shared between all users. If the key is compromised (say, if one user tells a friend what the password is, or an employee is let go) then changing the password can be prohibitively difficult, since all APs and client devices need to be changed. This also means that legitimate users of the network can still eavesdrop on each others' traffic, since they all know the shared key.

The key itself is often poorly chosen, making offline cracking attempts feasible. Even worse, the implementation of WEP itself is broken in many implementations, making it even easier to crack some networks. While manufacturers have implemented a number of extensions to WEP (such as longer keys and fast rotation schemes), these extensions are not part of the standard, and will not interoperate between equipment from different manufacturers. By upgrading to the most recent firmware for all of your wireless devices, you can prevent some of the early attacks found in WEP.

WEP can still be a useful authentication tool. Assuming your users can be trusted not to give away the password, you can be fairly sure that your wireless clients are legitimate. While WEP cracking is possible, it is beyond the skill of most users. WEP is extremely useful for securing long distance point-to-point links, even on generally open networks. By using WEP on such a link, you will discourage others from associating to the link, and they will likely use other available APs instead. WEP is definitely a handy "keep out" sign for your network. Anyone who detects the network will see that a key is required, making it clear that they are not welcome to use it.

WEPs greatest strength is its interoperability. In order to comply with the standards, all wireless devices support basic WEP. While it isn't the strongest method available, it is certainly the most commonly implemented feature. We will look at other more advanced encryption techniques later in this chapter.

For more details about the state of WEP encryption, see these papers:

- http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html
- http://www.cs.umd.edu/~waa/wireless.pdf
- http://www.crypto.com/papers/others/rc4_ksaproc.ps

Another data-link layer authentication protocol is ***Wi-Fi Protected Access***, or ***WPA***. WPA was created specifically to deal with the known problems with WEP mentioned earlier. It provides a significantly stronger encryption scheme, and can use a shared private key, unique keys assigned to each user, or even SSL certificates to authenticate both the client and the access point. Authentication credentials are checked using the 802.1X protocol,

which can consult a third party database such as RADIUS. Through the use of Temporal Key Integrity Protocol (TKIP), keys can be rotated quickly over time, further reducing the likelihood that a particular session can be cracked. Overall, WPA provides significantly better authentication and privacy than standard WEP.

The difficulty with WPA is that, as of this writing, interoperability between vendors is still very low. WPA requires fairly recent access point hardware and up-to-date firmware on all wireless clients, as well as a substantial amount of configuration. If you are installing a network in a setting where you control the entire hardware platform, WPA can be ideal. By authenticating both clients and APs, it solves the rogue access point problem and provides many significant advantages over WEP. But in most network settings where the vintage of hardware is mixed and the knowledge of wireless users is limited, WPA can be a nightmare to install. It is for this reason that most sites continue to use WEP, if encryption is used at all.

Captive portals

One common authentication tool used on wireless networks is the ***captive portal***. A captive portal uses a standard web browser to give a wireless user the opportunity to present login credentials. It can also be used to present information (such as an Acceptable Use Policy) to the user before granting further access. By using a web browser instead of a custom program for authentication, captive portals work with virtually all laptops and operating systems. Captive portals are typically used on open networks with no other authentication methods (such as WEP or MAC filters).

To begin, a wireless user opens their laptop and selects the network. Their computer requests a DHCP lease, which is granted. They then use their web browser to go to any site on the Internet.

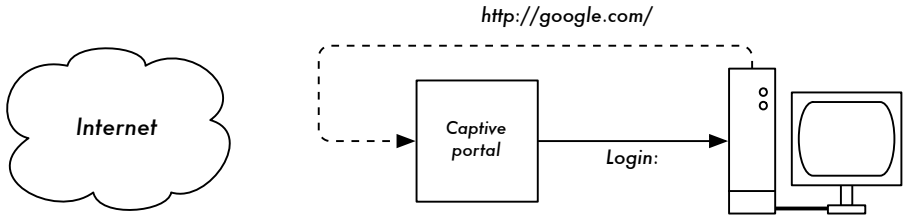

Figure 6.1: The user requests a web page and is redirected.

Instead of receiving the requested page, the user is presented with a login screen. This page can require the user to enter a user name and password, simply click a “login” button, type in numbers from a pre-paid ticket, or enter any other credentials that the network administrators require. The user then enters their credentials, which are checked by the access point or another

server on the network. All other network access is blocked until these credentials are verified.

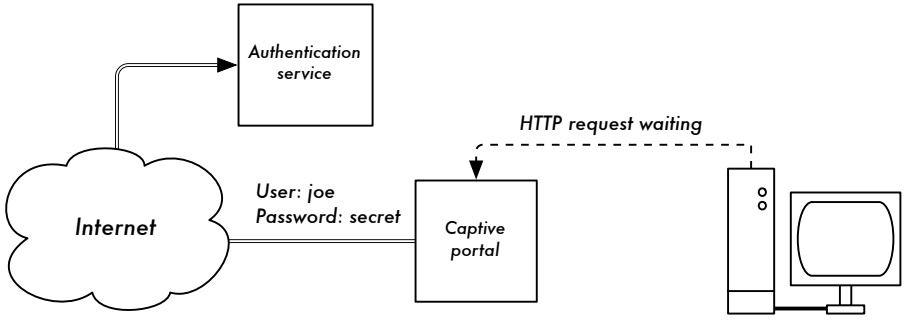

Figure 6.2: The user's credentials are verified before further network access is granted. The authentication server can be the access point itself, another machine on the local network, or a server anywhere on the Internet.

Once authenticated, the user is permitted to access network resources, and is typically redirected to the site they originally requested.

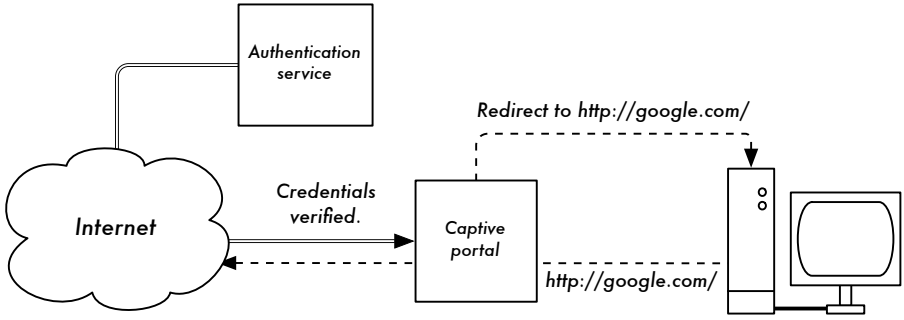

Figure 6.3: After authenticating, the user is permitted to access the rest of the network.

Captive portals provide no encryption for the wireless users, instead relying on the MAC and IP address of the client as a unique identifier. Since this is not necessarily very secure, many implementations will require the user to re-authenticate periodically. This can often be automatically done by minimizing a special pop-up browser window when the user first logs in.

Since they do not provide strong encryption, captive portals are not a very good choice for networks that need to be locked down to only allow access from trusted users. They are much more suited to cafes, hotels, and other public access locations where casual network users are expected.

In public or semi-public network settings, encryption techniques such as WEP and WPA are effectively useless. There is simply no way to distribute

public or shared keys to members of the general public without compromising the security of those keys. In these settings, a simple application such as a captive portal provides a level of service somewhere between completely open and completely closed.

Two popular open source captive portal implementations are NoCatSplash and Chillispot.

NoCatSplash

If you need to simply provide users of an open network with information and an acceptable use policy, take a look at NoCatSplash. It is available online at *http://nocat.net/download/NoCatSplash/ .*

NoCatSplash provides a customizable splash page to your users, requiring them to click a "login" button before using the network. This is useful for identifying the operators of the network and displaying rules for network access.

NoCatSplash is written in C, and will run on just about any Unix-like operating system including Linux, BSD, and even embedded platforms such as OpenWRT. It has a simple configuration file and can serve any custom HTML file as the splash page. It is typically run directly on an access point, but can also work on a router or proxy server. For more information, see *http://nocat.net/* .

Other popular hotspot projects

NoCatSplash is just one simple captive portal implementation. Many other free implementations exist that support a diverse range of functionality. Some of these include:

- Chillispot (*http://www.chillispot.org/*). Chillispot is a captive portal designed to authenticate against an existing user credentials database, such as RADUIS. Combined with the application phpMyPrePaid, pre-paid ticket based authentication can be implemented very easily You can download phpMyPrePaid from *http://sourceforge.net/projects/phpmyprepaid/*.
- WiFi Dog (*http://www.wifidog.org/*). WiFi Dog provides a very complete captive portal authentication package in very little space (typically under 30kb). From a user's perspective, it requires no pop-up or javascript support, allowing it to work on a wider variety of wireless devices.
- m0n0wall (*http://m0n0.ch/wall/*). As mentioned in chapter five, m0n0wall is a complete embedded operating system based on FreeBSD. It includes a captive portal with RADIUS support, as well as a PHP web server.

Privacy

Most users are blissfully unaware that their private email, chat conversations, and even passwords are often sent "in the clear" over dozens of untrusted networks before arriving at their ultimate destination on the Internet. However mistaken they may be, users still typically have some expectation of privacy when using computer networks.

Privacy can be achieved, even on untrusted networks such as public access points and the Internet. The only proven effective method for protecting privacy is the use of strong ***end-to-end encryption***.

Encryption techniques such as WEP and WPA attempt to address the privacy issue at layer two, the data-link layer. While this does protect eavesdroppers from listening in on the wireless connection, protection ends at the access point. If the wireless client uses insecure protocols (such as POP or simple SMTP for receiving and sending email), then users beyond the AP can still log the session and see the sensitive data. As mentioned earlier, WEP also suffers from the fact that it uses a shared private key. This means that legitimate wireless users can eavesdrop on each other, since they all know the private key.

By using encryption to the remote end of the connection, users can neatly sidestep the entire problem. These techniques work well even on untrusted public networks, where eavesdroppers are listening and possibly even manipulating data coming from the access point.

To ensure data privacy, good end-to-end encryption should provide the following features:

- ***Verified authentication of the remote end***. The user should be able to know without a doubt that the remote end is who it claims to be. Without authentication, a user could give sensitive data to anyone claiming to be the legitimate service.
- ***Strong encryption methods***. The encryption algorithm should stand up to public scrutiny, and it should not be easily decrypted by a third party. There is no security in obscurity, and strong encryption is even stronger when the algorithm is widely known and subject to peer review. A good algorithm with a suitably large and protected key can provide encryption that is unlikely to be broken by any effort in our lifetimes using current technology.
- ***Public key cryptography***. While not an absolute requirement for end-to-end encryption, the use of public key cryptography instead of a shared key can ensure that an individual user's data remains private, even if the key of

another user of the service is compromised. It also solves certain problems with distributing keys to users over untrusted networks.

- ***Data encapsulation***. A good end-to-end encryption mechanism protects as much data as possible. This can range from encrypting a single email transaction to encapsulation of all IP traffic, including DNS lookups and other supporting protocols. Some encryption tools simply provide a secure channel that other applications can use. This allows users to run any program they like and still have the protection of strong encryption, even if the programs themselves don't support it.

Be aware that laws regarding the use of encryption vary widely from place to place. Some countries treat encryption as munitions, and may require a permit, escrow of private keys, or even prohibit its use altogether. Before implementing any solution that involves encryption, be sure to verify that use of this technology is permitted in your local area.

In the following sections, we'll take a look at some specific tools that can provide good protection for your users' data.

SSL

The most widely available end-to-end encryption technology is ***Secure Sockets Layer***, known simply as ***SSL***. Built into virtually all web browsers, SSL uses public key cryptography and a trusted ***public key infrastructure*** (***PKI***) to secure data communications on the web. Whenever you visit a web URL that starts with https, you are using SSL.

The SSL implementation built into web browsers includes a collection of certificates from trusted sources, called ***certificate authorities*** (***CA***). These certificates are cryptographic keys that are used to verify the authenticity of websites. When you browse to a website that uses SSL, the browser and the server first exchange certificates. The browser then verifies that the certificate provided by the server matches its DNS host name, that it has not expired, and that it is signed by a trusted certificate authority. The server optionally verifies the identity of the browser's certificate. If the certificates are approved, the browser and server then negotiate a master session key using the previously exchanged certificates to protect it. That key is then used to encrypt all communications until the browser disconnects. This kind of data encapsulation is known as a ***tunnel***.

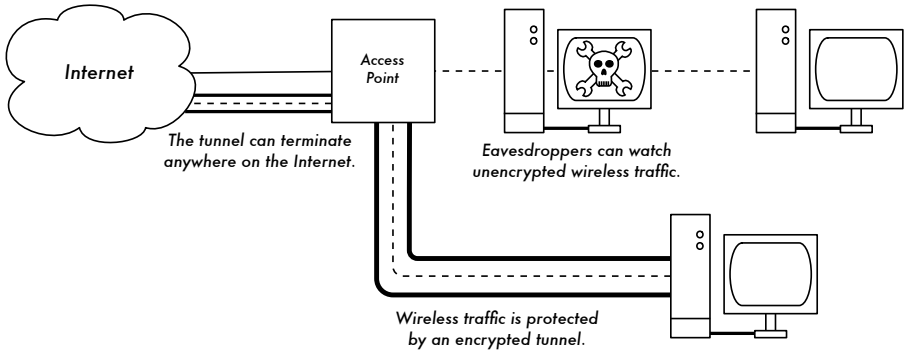

Figure 6.4: Eavesdroppers must break strong encryption to monitor traffic over an encrypted tunnel. The conversation inside the tunnel is identical to any other unencrypted conversation.

The use of certificates with a PKI not only protects the communication from eavesdroppers, but prevents so-called ***man-in-the-middle*** (***MITM***) attacks. In a man-in-the-middle attack, a malicious user intercepts all communication between the browser and the server. By presenting bogus certificates to both the browser and the server, the malicious user could carry on two simultaneous encrypted sessions. Since the malicious user knows the secret on both connections, it is trivial to observe and manipulate data passing between the server and the browser.

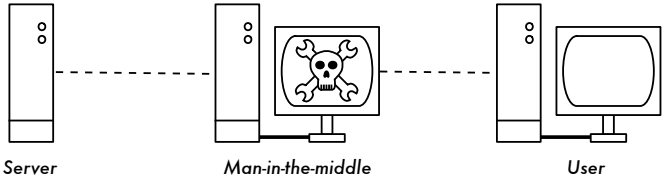

Figure 6.5: The man-in-the-middle effectively controls everything the user sees, and can record and manipulate all traffic. Without a public key infrastructure to verify the authenticity of keys, strong encryption alone cannot protect against this kind of attack.

Use of a good PKI prevents this kind of attack. In order to be successful, the malicious user would have to present a certificate to the client that is signed by a trusted certificate authority. Unless a CA has been compromised (very unlikely) or the user can be tricked into accepting the bogus certificate, then such an attack is not possible. This is why it is vitally important that users understand that ignoring warnings about expired or bogus certificates is very dangerous, especially when using wireless networks. By clicking the “ignore” button when prompted by their browser, users open themselves up to many potential attacks.

SSL is not only used for web browsing. Insecure email protocols such as IMAP, POP, and SMTP can be secured by wrapping them in an SSL tunnel. Most modern email clients support IMAPS and POPS (secure IMAP and POP) as well as SSL/TLS protected SMTP. If your email server does not provide SSL support, you can still secure it with SSL using a package like Stunnel (*http://www.stunnel.org/*). SSL can be used to effectively secure just about any service that runs over TCP.

SSH

Most people think of SSH as a secure replacement for **`telnet`**, just as **`scp`** and **`sftp`** are the secure counterparts of **`rcp`** and **`ftp`**. But SSH is much more than encrypted remote shell. Like SSL, it uses strong public key cryptography to verify the remote server and encrypt data. Instead of a PKI, it uses a key fingerprint cache that is checked before a connection is permitted. It can use passwords, public keys, or other methods for user authentication.

Many people do not know that SSH can also act as a general purpose encrypting tunnel, or even an encrypting web proxy. By first establishing an SSH connection to a trusted location near (or even on) a remote server, insecure protocols can be protected from eavesdropping and attack.

While this technique may be a bit advanced for many users, network architects can use SSH to encrypt traffic across untrusted links, such as wireless point-to-point links. Since the tools are freely available and run over standard TCP, any educated user can implement SSH connections for themselves, providing their own end-to-end encryption without administrator intervention.

OpenSSH (*http://openssh.org/*) is probably the most popular implementation on Unix-like platforms. Free implementations such as Putty (*http://www.putty.nl/*) and WinSCP (*http://winscp.net/*) are available for Windows. OpenSSH will also run on Windows under the Cygwin package (*http://www.cygwin.com/*). These examples will assume that you are using a recent version of OpenSSH.

To establish an encrypted tunnel from a port on the local machine to a port on the remote side, use the **-L** switch. For example, suppose you want to forward web proxy traffic over an encrypted link to the squid server at *squid.example.net*. Forward port 3128 (the default proxy port) using this command:

```
ssh -fN -g -L3128:squid.example.net:3128 squid.example.net
```

The **-fN** switches instruct ssh to fork into the background after connecting. The **-g** switch allows other users on your local segment to connect to the lo-

cal machine and use it for encryption over the untrusted link. OpenSSH will use a public key for authentication if you have set one up, or it will prompt you for your password on the remote side. You can then configure your web browser to connect to localhost port 3128 as its web proxy service. All web traffic will then be encrypted before transmission to the remote side.

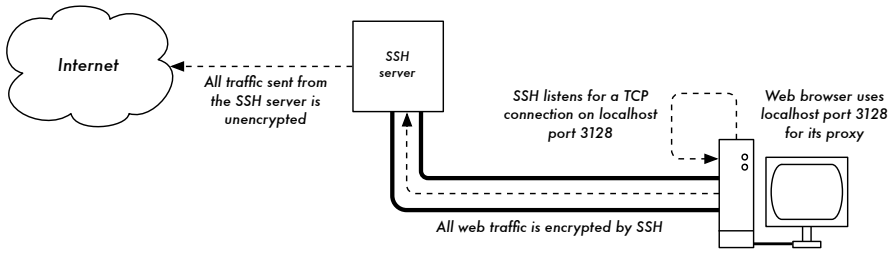

Figure 6.6: The SSH tunnel protects web traffic up to the SSH server itself.

SSH can also act as a dynamic SOCKS4 or SOCKS5 proxy. This allows you to create an encrypting web proxy, without the need to set up squid. Note that this is not a caching proxy; it simply encrypts all traffic.

```
ssh -fN -D 8080 remote.example.net
```

Configure your web browser to use SOCKS4 or SOCKS5 on local port 8080, and away you go.

SSH can encrypt data on any TCP port, including ports used for email. It can even compress the data along the way, which can decrease latency on low capacity links.

```
ssh -fNCg -L110:localhost:110 -L25:localhost:25 mailhost.example.net
```

The **-C** switch turns on compression. You can add as many port forwarding rules as you like by specifying the **-L** switch multiple times. Note that in order to bind to a local port less than 1024, you must have root privileges on the local machine.

These are just a few examples of the flexibility of SSH. By implementing public keys and using the ssh forwarding agent, you can automate the creation of encrypted tunnels throughout your wireless network, and protect your communications with strong encryption and authentication.

OpenVPN

OpenVPN is a free, open source VPN implementation built on SSL encryption. There are OpenVPN client implementations for a wide range of operating systems, including Linux, Windows 2000/XP and higher, OpenBSD,

FreeBSD, NetBSD, Mac OS X, and Solaris. Being a VPN, it encapsulates all traffic (including DNS and all other protocols) in an encrypted tunnel, not just a single TCP port. Most people find it considerably easier to understand and configure than IPSEC.

OpenVPN also has some disadvantages, such as fairly high latency. Some amount of latency is unavoidable since all encryption/decryption is done in user space, but using relatively new computers on either end of the tunnel can minimize this. While it can use traditional shared keys, OpenVPN really shines when used with SSL certificates and a certificate authority. OpenVPN has many advantages that make it a good option for providing end-to-end security.

- It is based on a proven, robust encryption protocol (SSL and RSA)
- It is relatively easy to configure
- It functions across many different platforms
- It is well documented
- It's free and open source.

Like SSH and SSL, OpenVPN needs to connect to a single TCP port on the remote side. Once established, it can encapsulate all data down to the Networking layer, or even down to the Data-Link layer, if your solution requires it. You can use it to create robust VPN connections between individual machines, or simply use it to connect network routers over untrusted wireless networks.

VPN technology is a complex field, and is a bit beyond the scope of this section. It is important to understand how VPNs fit into the structure of your network in order to provide the best possible protection without opening up your organization to unintentional problems. There are many good on-line resources that deal with installing OpenVPN on a server and client, I recommend this article from Linux Journal: *http://www.linuxjournal.com/article/7949* as well as the official HOWTO: *http://openvpn.net/howto.html*

Tor & Anonymizers

The Internet is basically an open network based on trust. When you connect to a web server across the Internet, your traffic passes through many different routers, owned by a great variety of institutions, corporations and individuals. In principle, any one of these routers has the ability to look closely at your data, seeing as a minimum the source and destination addresses, and quite often also the actual content of the data. Even if your data is encrypted using a secure protocol, it is possible for your Internet provider to monitor the

amount of data and the source and destination of that data. Often this is enough to piece together a fairly complete picture of your activities on-line.

Privacy and anonymity are important, and closely linked to each other. There are many valid reasons to consider protecting your privacy by ***anonymizing*** your network traffic. Suppose you want to offer Internet connectivity to your local community by setting up a number of access points for people to connect to. Whether you charge them for their access or not, there is always the risk that people use the network for something that is not legal in your country or region. You could plead with the legal system that this particular illegal action was not performed by yourself, but could have been performed by anyone connecting to your network. The problem is neatly sidestepped if it were technically infeasible to determine where your traffic was actually headed. And what about on-line censorship? Publishing web pages anonymously may also be necessary to avoid government censorship.

There are tools that allow you to anonymize your traffic in relatively easy ways. The combination of ***Tor*** (*http://tor.eff.org/*) and ***Privoxy*** (*http://www.privoxy.org/*) is a powerful way to run a local proxy server that will pass your Internet traffic through a number of servers all across the net, making it very difficult to follow the trail of information. Tor can be run on a local PC, under Microsoft Windows, Mac OSX, Linux and a variety of BSD's, where it anonymizes traffic from the browser on that particular machine. Tor and Privoxy can also be installed on a gateway server, or even a small embedded access point (such as a Linksys WRT54G) where they provides anonymity to all network users automatically.

Tor works by repeatedly bouncing your TCP connections across a number of servers spread throughout the Internet, and by wrapping routing information in a number of encrypted layers (hence the term ***onion routing***), that get peeled off as the packet moves across the network. This means that, at any given point in the network, the source and destination addresses cannot be linked together. This makes traffic analysis extremely difficult.

The need for the Privoxy privacy proxy in connection with Tor is due to the fact that name server queries (DNS queries) in most cases are not passed through the proxy server, and someone analyzing your traffic would easily be able to see that you were trying to reach a specific site (say *google.com*) by the fact that you sent a DNS query to translate google.com to the appropriate IP address. Privoxy connects to Tor as a SOCKS4a proxy, which uses hostnames (not IP addresses) to get your packets to the intended destination.

In other words, using Privoxy with Tor is a simple and effective way to prevent traffic analysis from linking your IP address with the services you use online. Combined with secure, encrypted protocols (such as those we have

seen in this chapter), Tor and Privoxy provide a high level of anonymity on the Internet.

Monitoring

Computer networks (and wireless networks in particular) are incredibly entertaining and useful inventions. Except, of course, when they don't work. Your users may complain that the network is "slow" or "broken", but what does that really mean? Without insight into what is actually happening, administering a network can be very frustrating.

In order to be an effective network administrator, you need access to tools that show you exactly what is happening on your network. There are several different classes of monitoring tools. Each shows you a different aspect of what is "going on", from the physical radio interaction all the way to how user applications interact with each other. By watching how the network performs over time, you can get an idea of what is "normal" for your network, and even be notified automatically when things seem to be out of the ordinary. The tools listed in this section are all quite powerful, and are freely available for download from the sources listed.

Network detection

The simplest wireless monitoring tools simply provide a list of available networks, along with basic information (such as signal strength and channel). They let you quickly detect nearby networks and determine if they are in range or are causing interference.

- **The built-in client.** All modern operating systems provide built-in support for wireless networking. This typically includes the ability to scan for available networks, allowing the user to choose a network from a list. While virtually all wireless devices are guaranteed to have a simple scanning utility, functionality can vary widely between implementations. These tools are typically only useful for configuring a computer in a home or office setting. They tend to provide little information apart from network names and the available signal to the access point currently in use.
- **Netstumbler** (*http://www.netstumbler.com/*). This is the most popular tool for detecting wireless networks using Microsoft Windows. It supports a variety of wireless cards, and is very easy to use. It will detect open and encrypted networks, but cannot detect "closed" wireless networks. It also features a signal/noise meter that plots radio receiver data as a graph over time. It also integrates with a variety of GPS devices, for logging precise location and signal strength information. This makes Netstumbler a handy tool to have for an informal site survey.

- **Ministumbler** (*http://www.netstumbler.com/*). From the makers of Netstumbler, Ministumbler provides much of the same functionality as the Windows version, but works on the Pocket PC platform. Ministumbler is handy to run on a handheld PDA with a wireless card for detecting access points in the field.
- **Macstumbler** (*http://www.macstumbler.com/*). While not directly related to the Netstumbler, Macstumbler provides much of the same functionality but for the Mac OS X platform. It works with all Apple Airport cards.
- **Wellenreiter** (*http://www.wellenreiter.net/*). Wellenreiter is a nice graphical wireless network detector for Linux. It requires Perl and GTK, and supports Prism2, Lucent, and Cisco wireless cards.

Protocol analyzers

Network protocol analyzers provide a great deal of detail about information flowing through a network, by allowing you to inspect individual packets. For wired networks, you can inspect packets at the data-link layer or above. For wireless networks, you can inspect information all the way down to individual 802.11 frames. Here are several popular (and free) network protocol analyzers:

- **Ethereal** (*http://www.ethereal.com/*). Ethereal is probably the most popular protocol analyzer available. It works with Linux, Windows, Mac OS X, and the various BSD systems. Ethereal will capture packets directly "from the wire" and display them in an intuitive graphical interface. It can decode over 750 different protocols, everything from 802.11 frames to HTTP packets. It will reassemble fragmented packets and follow entire TCP sessions easily, even if other data has broken up the sample. Ethereal is very valuable for troubleshooting tricky network problems, and figuring out exactly what is happening when two computers converse "on the wire".
- **Kismet** (*http://www.kismetwireless.net/*). Kismet is a powerful wireless protocol analyzer for Linux, Mac OS X, and even the embedded OpenWRT Linux distribution. It works with any wireless card that supports passive monitor mode. In addition to basic network detection, Kismet will passively log all 802.11 frames to disk or to the network in standard PCAP format, for later analysis with tools like Ethereal. Kismet also features associated client information, AP hardware fingerprinting, Netstumbler detection, and GPS integration.

  Since it is a passive network monitor, it can even detect "closed" wireless networks by analyzing traffic sent by wireless clients. You can run Kismet on several machines at once, and have them all report over the network back to a central user interface. This allows for wireless monitoring over a

large area, such as a university or corporate campus. Since it uses the passive monitor mode, it does all of this without transmitting any data.

- **KisMAC** (*http://kismac.binaervarianz.de/*). Exclusively for the Mac OS X platform, KisMAC does much of what Kismet can do, but with a slick Mac OS X graphical interface. It is a passive scanner that will log data to disk in PCAP format compatible with Ethereal. It does not support passive scanning with AirportExtreme cards (due to limitations in the wireless driver), but it supports passive mode with a variety of USB wireless cards.
- **Driftnet** and **Etherpeg**. These tools decode graphical data (such as GIF and JPEG files) and display them as a collage. As mentioned earlier, tools such as these are of limited use in troubleshooting problems, but are very valuable for demonstrating the insecurity of unencrypted protocols. Etherpeg is available from *http://www.etherpeg.org/*, and Driftnet can be downloaded at *http://www.ex-parrot.com/~chris/driftnet/*.

Bandwidth monitoring

The network is slow. Who is hogging all of the bandwidth? By using a good bandwidth monitoring tool, you can easily determine the source of spam and virus flooding problems. Such tools can also help you to plan for future capacity as the network users outgrow the available pipe. These tools will give you a visual representation of how traffic is flowing throughout your network, including traffic coming from a particular machine or service.

- **MRTG** (*http://people.ee.ethz.ch/~oetiker/webtools/mrtg/*). Most network administrators have encountered MRTG at some point in their travels. Originally written in 1995, MRTG is possibly the most widely used bandwidth monitoring application. Using Perl and C, it builds a web page full of graphs detailing the inbound and outbound traffic used on a particular network device. MRTG makes it simple to query network switches, access points, servers, and other devices and display the results as graphs that change over time.
- **RRDtool** (*http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/*). Developed by the same people who wrote mrtg, rrdtool is a more powerful generic monitoring application. RRD is short for “round-robin database”. It is a generic data format that allows you to easily track any particular data point as a set averaged over time. While rrdtool does not directly monitor interfaces or devices, many monitoring packages rely on it to store and display the data they collect. With a few simple shell scripts, you can easily monitor your network switches and access points, and plot the bandwidth used as a graph on a web page.
- **ntop** (*http://www.ntop.org/*). For historical traffic analysis and usage, you will want to investigate ntop. This program builds a detailed real-time report on observed network traffic, displayed in your web browser. It inte-

grates with rrdtool, and makes graphs and charts visually depicting how the network is being used. On very busy networks, ntop can use a lot of CPU and disk space, but it gives you extensive insight into how your network is being used. It runs on Linux, BSD, Mac OS X, and Windows.

- **iptraf** (*http://iptraf.seul.org/*). If you need to instantly take a snapshot of network activity on a Linux system, give iptraf a try. It is a command-line utility that gives you an up-to-the-second look at connections and network flows, including ports and protocols. It can be very handy for determining who is using a particular wireless link, and how heavily it is loaded. For example, by showing the detailed statistical breakdown for an interface, you can instantly find peer-to-peer client users, and determine exactly how much bandwidth they are currently using.

Troubleshooting

What do you do when the network breaks? If you can't access a web page or email server, and clicking the reload button doesn't fix the problem, then you'll need to be able to isolate the exact location of the problem. These tools will help you to determine just where a connection problem exists.

- **ping**. Just about every operating system (including Windows, Mac OS X, and of course Linux and BSD) includes a version of the ping utility. It uses ICMP packets to attempt to contact a specified host, and tells you how long it takes to get a response.

  Knowing what to ping is just as important as knowing how to ping. If you find that you cannot connect to a particular service in your web browser (say, *http://yahoo.com/*), you could try to ping it:

```
$ ping yahoo.com
PING yahoo.com (66.94.234.13): 56 data bytes
64 bytes from 66.94.234.13: icmp_seq=0 ttl=57 time=29.375 ms
64 bytes from 66.94.234.13: icmp_seq=1 ttl=56 time=35.467 ms
64 bytes from 66.94.234.13: icmp_seq=2 ttl=56 time=34.158 ms
^C
--- yahoo.com ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 29.375/33.000/35.467/2.618 ms
```

  Hit control-C when you are finished collecting data. If packets take a long time to come back, there may be network congestion. If return ping packets have an unusally low ttl, you may have routing problems between your machine and the remote end. But what if the ping doesn't return any data at all? If you are pinging a name instead of an IP address, you may be running into DNS problems.

Try pinging an IP address on the Internet. If you can't reach it, it's a good idea to see if you can ping your default router:

```
$ ping 216.231.38.1
PING 216.231.38.1 (216.231.38.1): 56 data bytes
64 bytes from 216.231.38.1: icmp_seq=0 ttl=126 time=12.991 ms
64 bytes from 216.231.38.1: icmp_seq=1 ttl=126 time=14.869 ms
64 bytes from 216.231.38.1: icmp_seq=2 ttl=126 time=13.897 ms
^C
--- 216.231.38.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 12.991/13.919/14.869/0.767 ms
```

If you can't ping your default router, then chances are you won't be able to get to the Internet either. If you can't even ping other IP addresses on your local LAN, then it's time to check your connection. If you're using Ethernet, is it plugged in? If you're using wireless, are you connected to the proper wireless network, and is it in range?

Network debugging with ping is a bit of an art, but it is useful to learn. Since you will likely find ping on just about any machine you will work on, it's a good idea to learn how to use it well.

- **traceroute** and **mtr** (*http://www.bitwizard.nl/mtr/*). As with ping, traceroute is found on most operating systems (it's called **tracert** in some versions of Microsoft Windows). By running traceroute, you can find the location of problems between your computer and any point on the Internet:

```
$ traceroute -n google.com
traceroute to google.com (72.14.207.99), 64 hops max, 40 byte packets
 1  10.15.6.1  4.322 ms  1.763 ms  1.731 ms
 2  216.231.38.1  36.187 ms  14.648 ms  13.561 ms
 3  69.17.83.233  14.197 ms  13.256 ms  13.267 ms
 4  69.17.83.150  32.478 ms  29.545 ms  27.494 ms
 5  198.32.176.31  40.788 ms  28.160 ms  28.115 ms
 6  66.249.94.14  28.601 ms  29.913 ms  28.811 ms
 7  172.16.236.8  2328.809 ms  2528.944 ms  2428.719 ms
 8  * * *
```

The **-n** switch tells traceroute not to bother resolving names in DNS, and makes the trace run more quickly. You can see that at hop seven, the round trip time shoots up to more than two seconds, while packets seem to be discarded at hop eight. This might indicate a problem at that point in the network. If this part of the network is in your control, it might be worth starting your troubleshooting effort there.

My TraceRoute (mtr) is a handy program that combines ping and traceroute into a single tool. By running mtr, you can get an ongoing average of latency and packet loss to a single host, instead of the momentary snapshot that ping and traceroute provide.

```
                          My traceroute  [v0.69]
tesla.rob.swn (0.0.0.0)        (tos=0x0 psize=64 bitpatSun Jan  8 20:01:26 2006
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                  Packets               Pings
 Host                           Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. gremlin.rob.swn              0.0%     4    1.9   2.0   1.7   2.6   0.4
 2. er1.sea1.speakeasy.net       0.0%     4   15.5  14.0  12.7  15.5   1.3
 3. 220.ge-0-1-0.cr2.sea1.speakeasy.  0.0%     4   11.0  11.7  10.7  14.0   1.6
 4. fe-0-3-0.cr2.sfo1.speakeasy.net   0.0%     4   36.0  34.7  28.7  38.1   4.1
 5. bas1-m.pao.yahoo.com         0.0%     4   27.9  29.6  27.9  33.0   2.4
 6. so-1-1-0.pat1.dce.yahoo.com  0.0%     4   89.7  91.0  89.7  93.0   1.4
 7. ae1.p400.msr1.dcn.yahoo.com  0.0%     4   91.2  93.1  90.8  99.2   4.1
 8. ge5-2.bas1-m.dcn.yahoo.com   0.0%     4   89.3  91.0  89.3  93.4   1.9
 9. w2.rc.vip.dcn.yahoo.com      0.0%     3   91.2  93.1  90.8  99.2   4.1
```

The data will be continuously updated and averaged over time. As with ping, you should hit control-C when you are finished looking at the data. Note that you must have root privileges to run mtr.

While these tools will not revel precisely what is wrong with the network, they can give you enough information to know where to continue troubleshooting.

Throughput testing

How fast can the network go? What is the actual usable capacity of a particular network link? You can get a very good estimate of your throughput capacity by flooding the link with traffic and measuring how long it takes to transfer the data. While there are web pages available that will perform a "speed test" in your browser (such as *http://www.dslreports.com/stest*), these tests are increasingly inaccurate as you get further from the testing source. Even worse, they do not allow you to test the speed of a particular link, but only the speed of your link to the Internet. Here are two tools that will allow you to perform throughput testing on your own networks.

- **ttcp** (*http://ftp.arl.mil/ftp/pub/ttcp/*). Now a standard part of most Unix-like systems, ttcp is a simple network performance testing tool. One instance is run on either side of the link you want to test. The first node runs in receive mode, and the other transmits:

```
node_a$ ttcp -r -s

node_b$ ttcp -t -s node_a
ttcp-t: buflen=8192, nbuf=2048, align=16384/0, port=5001  tcp -> node_a
ttcp-t: socket
```

```
ttcp-t: connect
ttcp-t: 16777216 bytes in 249.14 real seconds = 65.76 KB/sec +++
ttcp-t: 2048 I/O calls, msec/call = 124.57, calls/sec = 8.22
ttcp-t: 0.0user 0.2sys 4:09real 0% 0i+0d 0maxrss 0+0pf 7533+0csw
```

After collecting data in one direction, you should reverse the transmit and receive partners to test the link in the other direction. It can test UDP as well as TCP streams, and can alter various TCP parameters and buffer lengths to give the network a good workout. It can even use a user-supplied data stream instead of sending random data. Remember that the speed readout is in kilobytes, not kilobits. Multiply the result by 8 to find the speed in kilobits per second.

The only real disadvantage to ttcp is that it hasn't been developed in years. Fortunately, the code has been released in the public domain and is freely available. Like ping and traceroute, ttcp is found as a standard tool on many systems.

- **iperf** (*http://dast.nlanr.net/Projects/Iperf/*). Much like ttcp, iperf is a commandline tool for estimating the throughput of a network connection. It supports many of the same features as ttcp, but uses a "client" and "server" model instead of a "receive" and "transmit" pair. To run iperf, launch a server on one side and a client on the other:

```
node_a$ iperf -s

node_b$ iperf -c node_a
------------------------------------------------------------
Client connecting to node_a, TCP port 5001
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  5] local 10.15.6.1 port 1212 connected with 10.15.6.23 port 5001
[ ID] Interval       Transfer     Bandwidth
[  5]  0.0-11.3 sec   768 KBytes   558 Kbits/sec
```

The server side will continue to listen and accept client connections on port 5001 until you hit control-C to kill it. This can make it handy when running multiple test runs from a variety of locations.

The biggest difference between ttcp and iperf is that iperf is under active development, and has many new features (including IPv6 support). This makes it a good choice as a performance tool when building new networks.

Network health

By tracking information over time, you can get an overall idea of the general health of the network and its services. These tools will show you network trends and even notify a human when problems present themselves. More often than not, the systems will notice trouble before a person has a chance to call tech support.

- **cacti** (*http://www.cacti.net/*). As mentioned earlier, many tools use RRDtool as a back-end to build graphs for data that they collect. Cacti is such a tool. It is a PHP-based network management tool that simplifies data gathering and graph generation. It stores its configuration in a MySQL database, and is integrated with SNMP. This makes it very straightforward to map out all of the devices on your network, and monitor everything from network flows to CPU load. Cacti has an extensible data collection scheme that lets you collect just about any kind of data you can think of (such as radio signal, noise, or associated users) and plot it on a graph over time. Thumbnail views of your graphs can be combined into a single web page. This lets you observe the overall state of your network at a glance.
- **SmokePing** (*http://people.ee.ethz.ch/~oetiker/webtools/smokeping/*). Yet another tool by Tobias Oetiker, SmokePing is a tool written in Perl that shows packet loss and latency on a single graph. It is very useful to run SmokePing on a host with good connectivity to your entire network. Over time, trends are revealed that can point to all sorts of network problems. Combined with MRTG or Cacti, you can observe the effect that network congestion has on packet loss and latency. SmokePing can optionally send alerts when certain conditions are met, such as when excessive packet loss is seen on a link for an extended period of time.
- **Nagios** (*http://www.nagios.org/*). Nagios is a service monitoring tool. In addition to tracking the performance of simple pings (as with SmokePing), Nagios can watch the performance of actual services on any number of machines. For example, it can periodically query your web server, and be sure that it returns a valid web page. If a check should fail, Nagios can notify a person or group via email, SMS, or IM.

  While Nagios will certainly help a single admin to monitor a large network, Nagios is best used when you have a troubleshooting team with responsibilities divided between various members. Trouble events can be configured to ignore transient problems, then escalate notifications only to people who are responsible for fixing them. If the problem goes on for a predefined period of time without being acknowledged, other people can additionally be notified. This allows temporary problems to be simply logged without bothering people, and for real problems to be brought to the attention of the team.

7
Building an Outdoor Node

There are many practical considerations when installing electronic equipment outdoors. Obviously, it has to be protected from the rain, wind, sun, and other harsh elements. Power needs to be provided, and the antenna should be mounted at a sufficient height. Without proper grounding, nearby lightning strikes, fluctuating mains power, and even a light winds in the proper climate can annihilate your wireless links. This chapter will give you some idea of the practical problems you will be up against when installing wireless equipment outdoors.

Waterproof enclosures

Suitable waterproof enclosures come in many varieties. Metal or plastic may be used to create a watertight container for outdoor embedded equipment.

Of course, equipment needs power to work, and will likely need to connect to an antenna and Ethernet cable. Each time you pierce a watertight enclosure, you provide another potential place for water to seep in.

The National Electrical Manufacturers Association (NEMA) provides guidelines for protection of electrical equipment from rain, ice, dust, and other contaminants. An enclosure with a rating of ***NEMA 3*** or better is suitable for outdoor use in a fair climate. A ***NEMA 4X*** or ***NEMA 6*** provides excellent protection, even from hose driven water and ice. For fixtures that pierce the body of an enclosure (such as cable glands and bulkhead connectors), NEMA assigns an ingress protection (IP) rating. An ingress protection rating of ***IP66*** or ***IP67*** will protect these holes from very strong jets of water. A good outdoor enclosure should also provide UV protection to prevent breakdown

of the seal from exposure to the sun, as well as to protect the equipment inside.

Of course, finding NEMA rated enclosures may be a challenge in your local area. Often, locally available parts can be repurposed for use as enclosures. Rugged plastic or metal sprinkler boxes, electrical conduit housings, or even plastic food containers can be used in a pinch. When piercing an enclosure, use quality gaskets or o-rings along with a cable gland to seal the opening. UV stabilized silicone compound or other sealant can be used for temporary installations, but remember that cables flex in the wind, and glued joints will eventually weaken and allow moisture to seep in.

You can greatly extend the life of a plastic enclosure by providing some protection from the sun. Mounting the box in the shade, either beneath existing equipment, solar panel, or thin sheet of metal specifically for this purpose, will add to the life span of the box as well as the equipment contained inside.

Before putting any piece of electronics in a sealed box, be sure that it has minimal heat dissipation requirements. If your motherboard requires a fan or large heat sink, remember that there will be no airflow, and your electronics will likely bake to death on the tower. Only use electronic components that are designed to be used in an embedded environment.

Providing power

Obviously, DC power can be provided by simply poking a hole in your enclosure and running a wire. If your enclosure is large enough (say, an outdoor electrical box) you could even wire an AC outlet inside the box. But manufacturers are increasingly supporting a very handy feature that eliminates the need for an additional hole in the box: ***Power over Ethernet*** (***POE***).

The 802.3af standard defines a method for supplying power to devices using the unused pairs in a standard Ethernet cable. Nearly 13 Watts of power can be provided safely on a CAT5 cable without interfering with data transmissions on the same wire. Newer 802.3af compliant Ethernet switches (called ***end span injectors***) supply power directly to connected devices. End span switches can supply power on the same wires that are used for data (pairs 1-2 and 3-6) or on the unused wires (pairs 4-5 and 7-8). Other equipment, called ***mid span injectors,*** are inserted between Ethernet switches and the device to be powered. These injectors supply power on the unused pairs.

If your wireless router or CPE includes support for 802.3af, you could in theory simply connect it to an injector. Unfortunately, some manufacturers (notably Cisco) disagree on power polarity, and connecting mismatching gear can damage the injector and the equipment to be powered. Read the fine

print and be sure that your injector and wireless equipment agree on which pins and polarity should be used for power.

If your wireless equipment doesn't support power over Ethernet, you can still use the unused pairs in a CAT5 cable to carry power. You can either use a ***passive POE injector***, or simply build one yourself. These devices manually connect DC power to the unused wires on one end of the cable, and connect the other end directly to a barrel connector inserted in the device's power receptacle. A pair of passive POE devices can typically be purchased for under $20.

To make your own, you will need to find out how much power the device requires to operate, and provide at least that much current and voltage, plus enough to account for loss in the Ethernet run. You don't want to supply too much power, as the resistance of the small cable can present a fire hazard. Here is an online calculator that will help you calculate the voltage drop for a given run of CAT5 : *http://www.gweep.net/~sfoskett/tech/poecalc.html*

Once you know the proper power and electrical polarity needed to power your wireless gear, crimp a CAT5 cable only using the data wires (pairs 1-2 and 3-6). Then simply connect the transformer to pairs 4-5 (usually blue / blue-white) and 7-8 (brown / brown-white) on one end, and a matching barrel connector on the other. For a complete guide to building your own POE injector from scratch, see this terrific guide from NYCwireless: *http://nycwireless.net/poe/*

Mounting considerations

In many cases, equipment can be located inside a building, provided there is a window with ordinary glass through which the beam can travel. Normal glass will introduce little attenuation, but tinted glass will introduce unacceptable attenuation. This greatly simplifies mounting, power, and weatherproofing problems, but is obviously only useful in populated areas.

When mounting antennas on towers, it is very important to use a stand off bracket, and not mount the antennas directly to the tower. These brackets help with many functions including antenna separation, antenna alignment and protection.

Stand off brackets need to be strong enough to support the weight of the antenna, and also hold it in place on windy days. Remember, antennas can act like small sails, and can put a lot of force on to their mounts in strong winds. When estimating wind resistance, the total surface of the antenna structure must be considered, as well as the distance from the centre of the antenna to the point of attachment to the building. Large antennas such as

solid dishes or high gain sectorial panels can have considerable wind load. Using a slotted or mesh parabolic, rather than a solid dish, will help reduce the wind load without much affect on antenna gain. Be sure that the mounting brackets and supporting structure are solid, or your antennas will become misaligned over time (or worse, fall off the tower entirely!)

Mounting brackets must have enough clearance from the tower to allow for aiming, but not too much clearance that the antennas become too hard to reach if any service or maintenance is required.

Figure 7.1: An antenna with a standoff bracket being lifted onto a tower.

The pipe on the standoff bracket that the antenna will be mounted on needs to be round. This way the antenna can be pivoted on the pipe for aiming. Secondly, the pipe must also be vertical. If it is being mounted on a tapered tower, the standoff bracket will have to be designed to allow for this. This can be done using different lengths of steel, or by using combinations of threaded rod and steel plates.

As the equipment will be outside for all of its service life, it is important to be sure that the steel used is weatherproofed. Stainless steel often has too high a price tag for tower installations. Hot galvanizing is preferred, but may not be available in some areas. Painting all steel with a good rust paint will also

work. If paint is chosen, it will be important to plan a yearly inspection of the mount and repaint when necessary.

Guyed towers

A climbable guyed tower is an excellent choice for many installations, but for very tall structures a self supporting tower might be required.

When installing guyed towers, a pulley attached to the top of a pole will facilitate the tower installation. The pole will be secured to the lower section already in place, while the two tower sections are attached with an articulated joint. A rope passing through the pulley will facilitate the raising of the next section. After the cantilever section becomes vertical, bolt it to the lower section of the pole. The pole (called a gin pole in the trade) can then be removed, and the operation may be repeated, if required. Tighten the guy wires carefully, ensuring that you use the same tension at all suitable anchoring points. Chose the points so that the angles, as seen from the center of the tower, are as evenly spaced as possible.

Figure 7.2: A climbable guyed tower.

Self-supporting towers

Self supporting towers are expensive but sometimes needed, particularly when greater elevation is a requirement. This can be as simple as a heavy

pole sunk into a concrete piling, or as complicated as a professional radio tower.

Figure 7.3: A simple self-supporting tower.

An existing tower can sometimes be used for subscribers, although AM transmitting station antennas should be avoided because the whole structure is active. FM station antennas are acceptable, provided that at least a few of meters of separation is kept between the antennas. Be aware that while adjacent transmitting antennas may not interfere with your wireless connection, high powered FM may interfere with your wired Ethernet cable. Whenever using a heavily populated antenna tower, be very scrupulous about proper grounding and consider using shielded cable.

Figure 7.4: A much more complicated tower.

Rooftop assemblies

Non-penetrating roof mount antenna assemblies can be used on flat roofs. These consist of a tripod mounted to a metal or wooden base. The base is then weighed down with bricks, sandbags, water jugs, or just about anything heavy. Using such a rooftop "sled" eliminates the need to pierce the roof with mounting bolts, avoiding potential leaks.

Figure 7.5: This metal base can be weighed down with sandbags, rocks, or water bottles to make a stable platform without penetrating a roof.

Wall mount or metal strap assemblies can be used on existing structures such as chimneys or the sides of a buildings. If the antennas have to be mounted more than about 4 meters above the rooftop, a climbable tower may be a better solution to allow easier access to the equipment and to prevent antenna movement during high winds.

Dissimilar metals

To minimize electrolytic corrosion when two different metals are in moist contact, their electrolytic potential should be as close as possible. Use dielectric grease on the connection between two metals of different type to prevent any electrolysis effect.

Copper should never touch galvanized material directly without proper joint protection. Water shedding from the copper contains ions that will wash away

the galvanized (zinc) tower covering. Stainless steel can be used as a buffer material, but you should be aware that stainless steel is not a very good conductor. If it is used as a buffer between copper and galvanized metals, the surface area of the contact should be large and the stainless steel should be thin. Joint compound should also be used to cover the connection so water can not bridge between the dissimilar metals.

Protecting microwave connectors

Moisture leakage in connectors is likely the most observed cause of radio link failure. Be sure to tighten connectors firmly, but never use a wrench or other tool to do so. Remember that metals expand and contract as temperature changes, and an over-tightened connector can break in extreme weather changes.

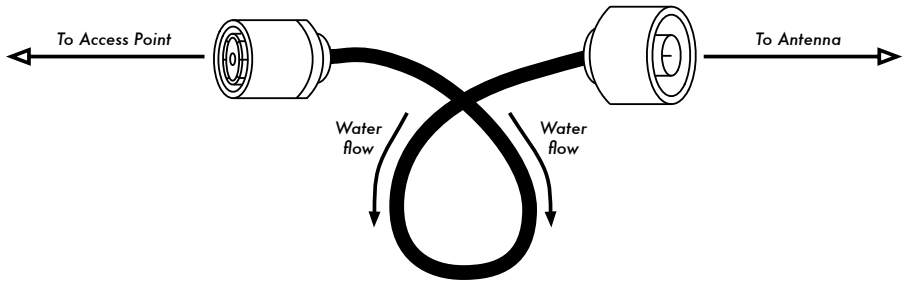

Figure 7.6: A drip loop forces rainwater away from your connectors.

Once tight, connectors should be protected by applying a layer of electrical tape, then a layer of sealing tape, and then another layer of electrical tape on top. The sealant protects the connector from water seepage, and the tape layer protects the sealant from ultraviolet (UV) damage. Cables should have an extra drip loop to prevent water from getting inside the transceiver.

Safety

Always use a harness securely attached to the tower when working at heights. If you have never worked on a tower, hire a professional to do it for you. Many countries require special training for people to be allowed to work on towers above a certain height.

Avoid working on towers during strong winds or storms. Always climb with a partner, and only when there is plenty of light. Tower work will likely take longer than you think it will. Remember that it is **extremely** hazardous to work in the dark. Give yourself plenty of time to complete the job long before the sun sets. If you run out of time, remember that the tower will be there in

the morning, when you can start on the problem again after a good night's sleep.

Aligning antennas on a long distance link

The key to successfully aligning antennas on a very long distance link is communication. If you change too many variables at once (say, one team starts wiggling an antenna while the other tries to take a signal strength reading), then the process will take all day and will probably end with misaligned antennas.

You will have two teams of people. Ideally, each team should have at least two people: one to take signal readings and communicate with the remote end, the other to manipulate the antenna. Keep these points in mind while working on long distance links.

1. **Test all equipment ahead of time.** You don't want to fiddle with settings once you're in the field. Before separating the equipment, power everything on, connect every antenna and pigtail, and make sure you can establish a connection between the devices. You should be able to return to this known good state by simply powering on the device, without having to log in or change any settings. Now is a good time to agree on antenna polarity (see chapter two if you don't understand what polarity means).
2. **Bring backup communications gear.** While mobile phones are usually good enough for working in cities, mobile reception can be bad or nonexistent in rural areas. Bring a high powered FRS or GMRS radio, or if your teams have amateur radio licenses, use a ham rig. Working at a distance can be very frustrating if you are constantly asking the other team "can you hear me now?" Pick your communication channels and test your radios (including the batteries) before separating.
3. **Bring a camera.** Take some time to document the location of each site, including surrounding landmarks and obstructions. This can be very useful later to determine the feasibility of another link to the location without having to travel there in person. If this is your first trip to the site, log the GPS coordinates and elevation as well.
4. **Start by estimating the proper bearing and elevation.** To begin, both teams should use triangulation (using GPS coordinates or a map) to get a rough idea of the direction to point. Use a compass to roughly align the antenna to the desired bearing. Large landmarks are also useful for pointing. If you can use binoculars to see the other end, all the better. Once you have made your guess, take a signal strength reading. If you are close enough and have made a good guess, you may already have signal.

5. **If all else fails, build your own landmark.** Some kinds of terrain make it difficult to judge the location of the other end of a link. If you are building a link in an area with few landmarks, a self-made landmark such as a kite, balloon, flood light, flare, or even smoke signal might help. You don't necessarily need a GPS to get an idea of where to point your antenna.

6. **Test signal in both directions, but only one at a time.** Once both ends have made their best guess, the end with the lowest gain antenna should make fix their antenna into position. Using a good monitoring tool (such as Kismet, Netstumbler, or a good built-in wireless client), the team with the highest gain antenna should slowly sweep it horizontally while watching the signal meter. Once the best position is found, try altering the elevation of the antenna. After the best possible position is found, lock the antenna firmly into place and signal the other team to begin slowly sweeping around. Repeat this process a couple of times until the best possible position for both antennas is found.

7. **Don't touch the antenna when taking a reading.** Your body will affect the radiation pattern of the antenna. Do not touch the antenna, and don't stand in the path of the shot, when taking signal strength readings. The same goes for the team on the other side of the link, too.

8. **Don't be afraid to push past the best received signal.** As we saw in chapter four, radiation patterns incorporate many smaller sidelobes of sensitivity, in addition to a much larger main lobe. If your received signal is mysteriously small, you may have found a sidelobe. Continue sweeping slowly beyond that lobe to see if you can find the main lobe.

9. **The antenna angle may look completely wrong.** The main lobe of an antenna often radiates slightly to one side or the other of the visual dead center of the antenna. Don't worry about how the antenna looks; you are concerned with finding the best possible position to achieve the greatest possible received signal.

10. **Double-check polarization.** It can be frustrating to attempt aligning a dish only to discover that the other team is using the opposite polarization. Again, this should be agreed upon before leaving home base, but if a link stays stubbornly weak, a double check doesn't hurt.

11. **If nothing works, check all components one at a time.** Are the devices on both ends of the link powered on? Are all pigtails and connectors properly connected, with no damaged or suspect parts? As outlined in chapter eight, proper troubleshooting technique will save you time and frustration. Work slowly and communicate your status well with the other team.

By working methodically and communicating well, you can complete the job of aligning high gain antennas in just a short while. If done properly, it should be fun!

Surge and lightning protection

Power is the greatest challenge for most installations in the developing world. Where there are electrical networks, they are often poorly controlled, fluctuate dramatically and are susceptible to lightning. Proper surge protection is critical to not only protect your wireless equipment, but all of the equipment connected to it.

Fuses and circuit breakers

Fuses are critical, but very often neglected. In rural areas, and even in many urban areas of developing countries, fuses are difficult to find. Despite the added cost, it is always prudent to use circuit breakers instead. These may need to be imported, but shouldn't be overlooked. Too often, replaceable fuses are removed and pocket change is used instead. In a recent case, all of the electronic equipment at at rural radio station was destroyed when a lightning strike went through the circuit, without circuit breaker or even a fuse to protect it.

How to ground

Proper grounding doesn't have to be a complicated job. When grounding, you are trying to accomplish two things: provide a short-circuit for a lightning strike, and provide a circuit for excess energy to be dissipated.

The first step is to protect equipment from a direct or near direct lightning hit, while the second provides a path to dissipate excess energy that would otherwise cause a build-up of static electricity. Static can cause significant degradation to signal quality, particularly on sensitive receivers (VSATs for example). Providing the short-circuit is simple. The installer simply needs to make the shortest path from the highest conductive surface (a lightning rod) to the ground. When a strike hits the rod, the energy will travel the shortest path and thus by-pass the equipment. This ground should be able to handle high-voltage (i.e. you need thick gauge wire, like 8 gauge braided copper).

To ground the equipment, mount a lightning rod above the equipment on a tower or other structure. Then use a thick gauge conductive wire to connect the rod to something that itself is well grounded. Underground copper pipes can be very well grounded (depending on their depth, the moisture, salinity, amount of metal and organic content of the soil). In many sites in West Africa, pipes aren't yet in the ground, and previous grounding equipment is

often inadequate due to ill-conductive soil (typical of seasonally arid, tropical soils). There are three easy ways to measure the efficiency of your ground:

1. The least accurate is to simply plug a good quality UPS or power strip into the circuit that has a ground detect indicator (a LED light). This LED is lit by energy that is being diffused to the ground circuit. An effective ground will dissipate small amounts of energy to the ground. Some people actually use this to pirate a bit of free light, as this energy does not turn an electrical counter!
2. Take a light socket and a low-wattage bulb (30 Watts), connect one wire to the ground wire and the second to the positive current. If the ground is working, the bulb should shine slightly.
3. The more sophisticated way is to simply measure the impedance between the positive circuit and the ground.

If your ground is not efficient you will need to bury a grounding stake deeper (where the soil is more moist, has more organic matter and metals) or you need to make the ground more conductive. A common approach where there is little soil is to dig a hole that is 1 meter in diameter and 2 meters deep. Drop in a highly conductive piece of metal that has some mass to it. This is sometimes called a ***plomb***, which literally means lead but can be any heavy piece of metal weighing 500 kg or more, such as an iron anvil or steel wheel. Then fill the hole with charcoal and mix in salt, then top with soil. Soak the area, and the charcoal and salt will diffuse around the hole and make a conductive area surrounding your plomb, improving the efficiency of the ground.

If radio cable is being used, it too can be used to ground the tower, though a more resilient design is to separate the ground for the tower from the cable. To ground the cable, simply peel back a bit of cable at the point closest to the ground before it goes into the building, then attach a ground cable from that point, either by soldering or using a very conductive connector. This then needs to be waterproofed.

Power stabilizers & regulators

There are many brands of power stabilizers, but most are either digital or electromechanical. The latter are much cheaper and more common. Electromechanical stabilizers take power at 220V, 240V, or 110V and use that energy to turn a motor, which always produces the desired voltage (normally 220V). This is normally effective, but these units offer little protection from lightning or other heavy surges. They often burn out after just one strike. Once burnt, they can actually be fused at a certain (usually wrong) output voltage.

Digital regulators regulate the energy using resistors and other solid state components. They are more expensive, but are much less susceptible to being burnt.

Whenever possible, use a digital regulator. They are worth the added cost, and will offer better protection for the rest of your equipment. Be sure to inspect all components of your power system (including the stabilizer) after lightning activity.

Solar and wind power

The applications described in this chapter use DC voltage. DC - Direct Current - has a polarity. Confusing the polarity will very likely immediately and irreversibly damage your equipment! I'll assume that you can handle a digital multimeter (DMM) to check out polarity. The DC voltages that are used in the described applications are not harmful when you touch conductors - but big lead-acid batteries can provide very high currents. A cable that creates a short between the terminals will immediately start to glow and burn its insulation. To prevent fire, there must be a fuse near the positive terminal of the battery at all times. That way the fuse will burn out before the cables do.

Lead acid batteries contain sulfuric acid that can cause severe burns. They release hydrogen when they are charged or have a short between terminals - even when they are the sealed acid type. Proper venting is necessary to prevent explosions, especially if the batteries are of the flooded cell acid type. It’s a good idea to protect your eyes with safety glasses when handling these batteries. I once met a battery “expert” that blew off three batteries during his career. Lead is toxic - make sure you dispose of worn out batteries properly. This may be difficult in countries that don't have any recycling infrastructure.

Off-the-grid power

There are many situations where you want to install a wireless node in an area where the grid providing mains power is unstable or just not existing. This could be a remote wireless relay, or a developing country where the grid fails often.

An autonomous power system consists basically of a battery which stores electric energy that is produced by a wind, solar and/or gasoline generator. Furthermore, electronic circuitry that controls the charging/discharging process is necessary.

It is important to choose a device that draws a minimum of energy when designing an system for operation on solar energy or wind power. Every watt that is wasted on the consumer side causes high costs at the side of the

power source. More power consumption means that larger solar panels and bulkier batteries will be necessary to provide sufficient energy. Saving power by choosing the right gear saves a lot of money and trouble. For example , a long distance link doesn't necessarily need a strong amplifier that draws a lot of power. A Wi-Fi card with good receiver sensitivity and a fresnel zone that is at least 60% clear will work better than an amplifier, and save power consumption as well. A well known saying of radio amateurs applies here, too: The best amplifier is a good antenna. Further measures to reduce power consumption include throttling the CPU speed, reducing transmit power to the minimum value that is necessary to provide a stable link, increasing the length of beacon intervals, and switching the system off during times it is not needed.

Most autonomous solar systems work at 12 or 24 volts. Preferably, a wireless device that runs on DC voltage should be used, operating at the 12 Volts that most lead acid batteries provide. Transforming the voltage provided by the battery to AC or using a voltage at the input of the access point different from the voltage of the battery will cause unnecessary energy loss. A router or access point that accepts 8-20 Volts DC is perfect.

Most cheap access points have a switched mode voltage regulator inside and will work through such a voltage range without modification or becoming hot (even if the device was shipped with a 5 or 12 Volt power supply).

WARNING: Operating your access point with a power supply other than the one provided by your manufacturer will certainly void any warranty, and may cause damage to your equipment. While the following technique will typically work as described, remember that should you attempt it, you do so at your own risk.

Open your access point and look near the DC input for two relatively big capacitors and an inductor (toroid with copper wire wrapped around it). If they are present, the device has a switched mode input, and the maximum input voltage should be somewhat below the voltage printed on the capacitors. Usually the rating of these capacitors is 16 or 25 volts. Be aware that an unregulated power supply has a ripple and may feed a much higher voltage into your access point than the typical voltage printed on it may suggest. So, connecting an unregulated power supply with 24 Volts to a device with 25 Volt-capacitors is not a good idea. Of course, opening your device will void any existing warranty. Do not try to operate an access point at higher voltage if it doesn't have a switched mode regulator. It will get hot, malfunction, or burn.

The popular Linksys WRT54G runs at any voltage between 5 and 20 volts DC and draws about 6 Watts, but it has an Ethernet switch onboard. Having a switch is of course nice and handy - but it draws extra power. Linksys also

offers a Wi-Fi access point called WAP54G that draws only 3 Watts and can run OpenWRT and Freifunk firmware. The 4G Systems Accesscube draws about 6 Watts when equipped with a single WiFi interface. If 802.11b is sufficient, mini-PCI cards with the Orinoco chipset perform very well while drawing a minimum amount of power.

Another important strategy for saving power is keeping DC power cables short and using a good quality, thick cable. This will keep voltage loss at a minimum.

Calculating and measuring power consumption

The design of an autonomous system always begins with the calculation of how much power is consumed. The easiest way to measure your device is a laboratory power supply that features a voltage and ampere meter. The nominal voltage provided by a lead acid battery typically varies between 11 Volts (empty) and about 14.5 Volt (charging, voltage at charging limit). You can tune the voltage at the laboratory power supply and see how much current the device draws at different voltages. If a laboratory power supply is not available, measurement can be performed by using the supply shipped with the device. Interrupt one cable that goes to the DC input of your device and insert an ***ampere-meter*** (or ***ammeter***). Note that a ammeter will burn itself or your power supply if applied between the positive and negative terminal because it behaves like a simple cable between the probes - thus creating a short. Many ammeters have an unfused input, so exercise caution as they can be easily damaged.

The amount of power consumed can be calculated with this formula:

```
P = U * I
```

P being Power in Watts, U being voltage in Volts, I being current in Ampere. For example:

```
6 Watts = 12 Volts * 0.5 Ampere
```

The result is the rating of the device. If the device of the example is operating for an hour it will simply consume 6 Watt-hours (Wh), respectively 0.5 Ampere-hours (Ah). Thus the device will draw 144 Wh or 12 Ah a day.

To simplify things, I will use the nominal voltage rating of batteries for calculations and not take into account that the voltage provided by the battery varies depending on its state of charge. Batteries are rated at their capacity in Ah - so it is easier to calculate using Ah instead of Wh. A battery from a big truck has typically 170 Ah - thus a 100% charged truck battery would power the device for about 340 hours during a 100% discharging cycle.

Discharging characteristics - Rule of thumb

A 12 Volt lead-acid battery that delivers energy to a consumer provides a voltage depending on it's state of charge. When the battery is 100% charged it has a output voltage of 12.8 Volts which is quickly dropping to 12.6 Volts under load. Given that the battery has to provide constant current the output voltage is now linear, dropping from 12.6 Volt to 11.6 Volts over a long period. Beneath 11.6 Volt the output voltage is dropping down quickly over time. Since the battery provides approximately 95% of it's power within this linear voltage drop, the charging state could be estimated by measuring the voltage under load. The assumption is that the battery is 100% full at 12.6 Volts and has 0% charge at 11.6 Volts. So, when measuring a battery that is currently discharged, the status can be estimated with a digital multimeter. For example a reading of 12.5 Volts corresponds 90% charge, 12.3 Volts corresponds 70% charge, etc.

Lead acid batteries degrade quickly when charging cycles go down to 0% charge. A battery from a truck will lose 50% of it's design capacity within 50 - 150 cycles if it is fully charged and discharged during each cycle. At 0% charge the battery still has 11 Volt at the terminals under load. Never discharge a 12 Volt lead acid battery beneath this value. It will forfeit a huge amount of storage capacity. Discharging to 0 Volt will utterly ruin it. To avoid this, a low voltage disconnect circuit (LVD) should be used to build a battery powered system. In cycle use it is not advisable to discharge a simple truck battery beneath 70%. Not going beneath 80% will significantly increase it's durability. Thus a 170 Ah truck battery has only a usable capacity of 34 to 51 Ah!

A battery from a car or truck should stay beyond 12.3 Volts in the system. In rare cases it may be allowed to drop down beneath this value - an unexpected long period of bad weather for example. This is tolerable if the battery is fully charged after such an incident. Charging to 100% charge takes quite a while because the charging process slows down when approaching the charging end even if there is plenty of energy from the power source. A weak power source may seldom achieve a full charge and thus wear out batteries quickly. It is recommended to charge aggressively to keep cost of ownership low. A wind/solar charging regulator or automatic battery charger (with advanced charge characteristic) will help save money. Best is IUIa-characteristic, IU characteristic is second choice.

Starter batteries are the cheapest batteries available, but they may not be the best option. There are special solar batteries on the market which are designed for use in solar systems. They allow deeper recharging cycles (down to 50% charge, depending on type) and have a low self-discharge current. The same applies to most sealed lead acid batteries. Sealed lead acid batteries are more expensive but safer to handle.

Truck or car batteries that carry the label ***maintenance-free*** should have neglectable low self-discharge current. However, maintenance-free batteries still need maintenance. The level of the electrolyte fluid must be checked frequently, especially in hot climate. If there is loss of electrolyte, distilled water has to be used to fill up the fluid. Neglecting this will ruin the battery.

Charging your batteries too much will destroy them too! The charging current in a battery buffered system must be regulated. Excessive and unlimited charging will destroy the battery. If the voltage in the battery is too high, the water component of the sulfuric acid will be cracked up by electrolysis, causing an atmosphere which contains a concentrated amount of oxygen. Oxygen is very corrosive and will destroy internal connectors.

Designing a battery buffered system

Things are less complicated if there is an unstable mains grid available that does its job every now and then. In that case, all that is needed is a decent automatic charger that is capable of fully charging a battery of sufficient size. A switched mode charger with a wide range voltage input and sophisticated charging characteristics is desirable. This will help protect against the grid, which may provide varying voltages. Cheap chargers that feature a simple transformer may never charge your battery at all if the voltage of the grid is too low. A simple charger designed for 230 Volts AC will provide little to no charging current when operated at 200 Volts or lower. No matter how long it operates, it will never achieve a full charge. On the other hand, it will burn out if the voltage is a little higher than expected - or it will simply ruin the batteries after a while. An AC voltage stabilizer that prevents your charger from burning out due excessive high voltage may be a really good idea in many situations.

A battery buffered system looks like this:

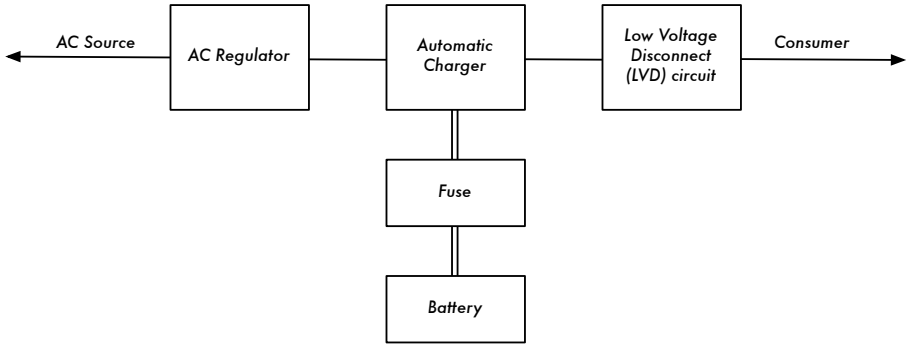

Figure 7.7: The complete battery buffered system.

Suppose our device draws 7 Watts at 12 Volts. We need the service 24 hours a day - so the device will draw:

```
168 Wh = 24h * 7 W
```

At 12 Volt the current in ampere would be:

```
14 Ah = 168 Wh / 12 Volt
```

Now, lets assume that occasionally we get a situation where the grid fails for one week.

```
98 Ah = 14 Ah/day * 7 days
1176 Wh = 98 Ah * 12 Volt
```

If we allow our battery to get discharged from 100% to 30% charge, thus consuming 70% of the capacity, we need a storage capacity of:

```
140 Ah = 98 ah / 0.7
```

A truck battery is available with this size.

Usually power comes back for 5 hours a day, thus the system will run 19 hours on battery.

```
133 Wh = 19h * 7 Watt
```

Charging and discharging a battery is never 100% efficient. There will always be energy loss in the battery, so we have to charge with more energy than we get. Charging/discharging efficiency usually is about 75%.

```
177.4 Wh = 133 Wh / 0.75
```

We want to charge aggressively and achieve a full charge within 5 hours.

Considering charging efficiency:

```
166 Wh = 148 Wh / 0.75
```

Converting to Ah:

```
14.8 Ah = 177.4 Wh / 12 Volt
```

Considering charging time:

```
2.96 A = 14.8 Ah / 5h
```

While we are charging the access point/router still draws power. 7 Watts equals 0.6 Ampere at 12 Volts:

```
3.56 A = 2.96 A + 0.6 A
```

We should consider that the charging process slows down near the end of the charge period. It would be better to have a higher initial charging current than calculated to achieve a 100% charge. A charging time of 5 hours is quite short, so a IUIa-charger with 8 Amperes or more is a good investment.

Even a cheap truck battery should last for 5 years, given that the electrolyte is checked frequently. Don't forget to use a low voltage disconnect circuit. It is not a mistake to oversize such a system to some degree. No matter how well designed the system is, the battery component will wear out and need replacement. In general, it is more cost effective to oversize the power source rather than batteries.

Designing a solar or wind powered system

The amount of energy that you can harvest with a solar or wind powered system depends on the area where you are and the time of the year. Usually you'll find information about the energy of the sun radiation or wind speed from administrative bodies competent for weather. They collect such information over the years and can tell you what to expect for each time of the year. Simulation and calculation programs for solar systems are available, PVSOL being one commercial (and expensive) program. A demo version is available in several languages.

Calculating exactly how much energy a solar powered system will produce at a certain site is a lot of work. Involved in the calculation are factors like temperature, number of sun hours, intensity of radiation, reflections in the environment, alignment of the solar panels and so on. A simulation program and weather data are a good place to start, but remember that in the real world, something as simple as dirt on the solar panels can completely spoil the results of your theoretical calculation.

Estimating the amount of energy produced by a wind generator is hard if there are obstacles around the wind generator. The empiric approach would be to measure the actual wind speed at the site over a year - which is rather impractical.

This should be a practical guide. If a fancy computer program and detailed weather data is not available for your country, I would suggest building a pilot system. If the battery does not get sufficiently charged, it is time to increase the number or size of the solar panels. As mentioned before, keeping the

power consumption at a minimum is really important to avoid unexpected high costs.

If the system needs to have 100% uptime, considerations will obviously start with the worst time of the year. You have to decide whether the system will need an oversized storage capacity or an oversized power source to provide power through calm periods. It may be much cheaper if someone manually charges the system with a generator running on gasoline in a time of a long dead calm.

Combining wind and solar energy makes the most sense in areas with seasons that provide wind energy when solar energy is weak. For example, in Germany the sun provides only 10% of the energy in winter time compared to summer. In spring and autumn there is not much solar power either, but it is quite windy. Huge batteries are necessary since it is possible that neither solar panels or a wind generator will provide much energy during wintertime.

Under such conditions, a system designed for 100% uptime needs a decent safety margin and a lot of storage capacity. Charging should be done aggressively to achieve full charge as often as possible during periods of good weather. In the long run, solar panels may need replacement every 25 years - while a battery in a system that lacks sufficient charging power may need replacement every year!

Circuit

An autonomous solar system consists of:

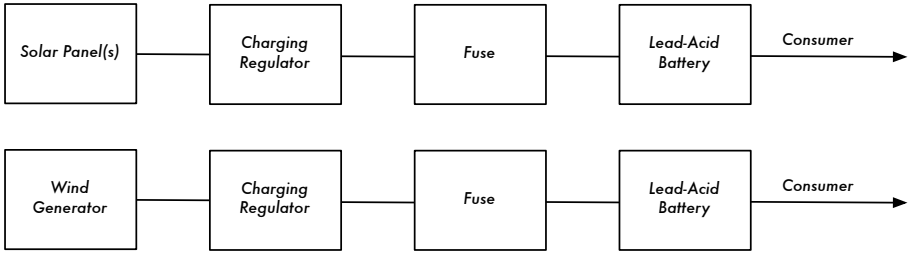

Figure 7.8: A solar powered or wind powered system.

Both systems are connected to the same battery if wind and solar power is combined.

Wind power

A wind generator is a clear option when an autonomous system is being designed for a wireless relay to be built on a hill or mountain. A concern for wind power is that the wind speed must be high enough at a site which may

be surrounded by objects. The average wind speed over the year should be at least 3 - 4 meter per second, and the wind generator should be 6 meters higher than other objects within a distance of 100 meters. A location far away from the coast usually lacks sufficient wind energy to support a wind powered system.

Solar power

In most cases, a system using only solar panels is the best solution. It is usually pretty easy to find a location suitable for solar panels, and they contain no mechanical moving parts that need maintenance.

It is important for a solar system that the solar panels are mounted with the best alignment and angle to the sun. The best angle may vary over the year and is dependent on the location of the site. It is a good idea to take into account that dust, leaves or birds may defile a solar panel. The optimum mounting angle may be quite flat, causing dirt to settle on the solar panel, making frequent cleaning necessary.

Shade must not wander over the solar panel during the day, because solar panels consist of a number of solar cells that are connected in a daisy chain. A chain is as strong as its weakest element. If something covers one cell of a solar panel completely - a leaf for example - the entire solar panel will produce no power. Even the shade from a cable will significantly reduce the amount of energy produced by the solar system!

Charging regulators

Charging regulators for wind generators are different from regulators for solar panels. If the system features wind and solar energy two regulators are needed. Each regulator has to be connected to the terminals of the battery directly (via a fuse, of course!).

Influence of maximum power point tracking

Manufacturers of solar panels are optimistic when calculating the power rating of their panels. Thus, the power that is effectively produced by a panel is significantly lower than claimed on the data sheet. The power rating is only achieved at a certain voltage, at a panel temperature of 20 degrees Celsius and at a sun radiation of 1000 Watt per square meter. This is not realistic because a solar panel gets really hot at 1000 Watt radiation per square meter. High temperature reduces the effective power output of a panel. There is not much that can be done about it apart from keeping in mind that a panel never achieves the claimed power rating.

The influence of the panel output voltage is more important to consider in a autonomous system. If a simple charging regulator is used, the voltage in the panel drops down to the level of the battery voltage. A solar panel may have the best efficiency at 18 Volts - it may produce 1 Ampere at 300 Watt/m at 30 degrees Celsius. This point of best efficiency is called ***Maximum Power Point*** or ***MPP***.

Thus, our panel would produce:

```
18 Watt = 18 Volt * 1 Ampere
```

If this panel is connected to a battery at 12.3 Volt the current will be slightly higher than in the MPP, maybe 1.1 Ampere, but the panel voltage will drop down to the level of the battery:

```
13.5 Watt = 12.3 Volt * 1.1 Ampere
```

The efficiency in our example would be only 75% with a simple charging regulator. This problem could be addressed by using a solar regulator with maximum power point tracking. A well designed MPP-regulator achieves an efficiency of 90%. A system with a simple regulator may never achieve more than 70% of the power rating given by the manufacturer.

Increasing battery and solar panel capacity

If you want to combine two (or more) batteries to increase capacity, interconnect them parallel - that is, interconnect both positive terminals with a heavy gauge cable. There must be a fuse in the cable near every positive terminal. Interconnect the negative terminals without fuses. Interconnecting solar panels can be done accordingly without fuses.

Low voltage disconnect circuit

Consumers (your access point, wireless router, or other device) will be connected to the charging regulator. Most charging regulators come with a low voltage disconnect circuit. The low voltage disconnect circuit should never need to switch off, otherwise there is a serious design flaw or damage present. If it happens that there are two or more regulators in the system that have a Low Voltage Disconnect Circuit, then connect the consumers to one regulator only. Otherwise the regulators could be damaged.

Calculation

The calculation of a solar system is not much different than the battery buffered system (as detailed earlier). Obviously, the times when no energy is

available for charging could be very long, and there is no fixed charging current that could be used for calculation.

A well designed system should be able to fully recharge an empty battery within a few days in good weather conditions while delivering power to the consumers.

8
Troubleshooting

How you establish the support infrastructure for your network is as important as what type of equipment you use. Unlike wired connections, problems with a wireless network are often invisible, and can require more skill and more time to diagnose and remedy. Interference, wind, and new physical obstructions can cause a long-running network to fail. This chapter details a series of strategies to help you build a team that can support your network effectively.

Building your team

Every village, company or family has individuals who are intrigued by technology. They are the ones found splicing the television cable, re-wiring a broken television or welding a new piece to a bicycle. These people will take interest in your network and want to learn as much about it as possible. Though these people are invaluable resources, you must avoid imparting all of the specialized knowledge of wireless networking to only one person. If your only specialist loses interest or finds better paying work somewhere else, they take the knowledge with them when they go.

There may also be many young and ambitious teenagers or young adults who will be interested and have the time to listen, help, and learn about the network. Again, they are very helpful and will learn quickly, but the project team must focus their attention on those who are best placed to support the network in the coming months and years. Young adults and teenagers will go off to university or find employment, especially the ambitious youth who tend to want to be involved. These youth also have little influence in the community, where an older individual is likely to be more capable of making decisions that positively affect the network as a whole. Even though these indi-

viduals might have less time to learn and might appear to be less interested, their involvement and proper education about the system can be critical.

Therefore, a key strategy in building a support team is to balance and to distribute the knowledge among those who are best placed to support the network for the long term. You should involve the youth, but do not let them capitalize use or knowledge of these systems. Find people who are committed to the community, who have roots in the community, who can be motivated, and teach them. A complementary strategy is to compartmentalize functions and duties, and to document all methodology and procedures. In this way, people can be trained easily, and substituted with little effort.

For example, in one project site the training team selected a bright young university graduate who had returned to his village. He was very motivated and learned quickly. Because he learned so quickly, he was taught more than had been foreseen, and he was able to deal with a variety of problems, from fixing a PC to rewiring Ethernet cable. Unfortunately, two months after the project launch he was offered a government job and left the community. Even a better salary could not keep him, since the prospect of a stable government job was too appealing. All of the knowledge about the network and how to support it left with him. The training team had to return and begin the training again. The next strategy was to divide functions, and to train people who were permanently rooted in the community: people who had houses and children, and were already employed. It took three times as long to teach three people as it took to train the young university grad, but the community will retain this knowledge for much longer.

Though this might seem to suggest that you should hand-pick who is to be involved, that is not often the best approach. It is often best to find a local partner organization or a local manager, and work with them to find the right technical team. Values, history, local politics, and many other factors will be important to them, while remaining completely unfathomable to people who are not from that community. The best approach is to coach your local partner, to provide them sound criteria, make sure that they understand that criteria, and to set firm boundaries. Such boundaries should include rules about nepotism and patronage, though these rules must consider the local situation. It may be impossible to say that you cannot hire kin, but it is best to provide a means of checks and balances. Where a candidate is kin, there should be clear criteria and a second authority in deciding upon their candidacy. It is also important that the local partner is given this authority and is not undermined by the project organizers, thus compromising their ability to manage. They will be best able to judge who will work best with them. If they are well educated in this process, then your requirements should be satisfied.

Troubleshooting and support of technology is an abstract art. The first time you look at an abstract painting, it may just look to you like a bunch of ran-

dom paint splatters. After reflecting on the composition for a time, you may come to appreciate the work as a whole, and the “invisible” coherence becomes very real. The neophyte looking at a wireless network may see the antennas and wires and computers, but it can take a while for them to appreciate the point of the “invisible” network. In rural areas, it can often take a huge leap of understanding before locals will appreciate an invisible network that is simply dropped into their village. Therefore, a phased approach is needed to ease people into supporting technology systems. The best method is involvement. Once the participants are chosen and committed to the project, involve them as much as possible. Let them "drive". Give them the cable crimper or keyboard and show them how to do the work. Even if you do not have time to explain every detail and even if it will take longer, they need to be involved physically and see not only what has been done, but how much work was done.

The scientific method is taught in virtually all western schools. Many people learn about it by the time they reach high-school science class. Simply put, you take a set of variables, then slowly eliminate those variables through binary tests until you are left with one or only a few possibilities. With those possibilities in mind, you complete the experiment. You then test to see if the experiment yields something similar to the expected result. If it did not, you re-calculate your expected result and try again. The typical agrarian villager may have been introduced to the concept, but likely will not have had the opportunity to troubleshoot complex problems. Even if they are familiar with the scientific method, they might not think to apply it to resolving real problems.

This method is very effective, although time consuming. It can be sped up by making logical assumptions. For example, if a long-running access point suddenly stops working after a storm, you might suspect a power supply related problem and thus skip most of the procedure. People charged with supporting technology should be taught how to troubleshoot using this method, as there will be times when the problem is neither known nor evident. Simple decision trees or flow charts can be made that test these variables, and try to eliminate the variables to isolate the problem. Of course, these charts should not be followed blindly.

It is often easier to teach this method using a non technological problem first. For example, have your student develop a problem resolution procedure on something simple and familiar, like a battery powered television. Start by sabotaging the television. Give them a battery that is not charged. Disconnect the aerial. Insert a broken fuse. Test the student, making it clear that each problem will show specific symptoms, and point the way as to how to proceed. Once they have fixed the television, have them apply this procedure to a more complicated problem. In a network, you can change an IP address, switch or damage cables, use the wrong SSID, or orient the

antenna in the wrong direction. It is important that they develop a methodology and procedure to resolve these problems.

Proper troubleshooting technique

No troubleshooting methodology can completely cover all problems you will encounter when working with wireless networks. But often, problems come down to one of a few common mistakes. Here are a few simple points to keep in mind that can get your troubleshooting effort working in the right direction.

- **Don't panic.** If you are troubleshooting a system, that means that it was working at one time, probably very recently. Before jumping in and making changes, survey the scene and assess exactly what is broken. If you have historical logs or statistics to work from, all the better. Be sure to collect information first, so you can make an informed decision before making changes.
- **Is it plugged in?** This step is often overlooked until many other avenues are explored. Plugs can be accidentally (or intentionally) unplugged very easily. Is the lead connected to a good power source? Is the other end connected to your device? Is the power light on? It may sound silly, but you will feel even sillier if you spend a lot of time checking out an antenna feed line only to realize that the AP was unplugged the entire time. Trust me, it happens more often than most of us would care to admit.
- **What was the last thing changed?** If you are the only person with access to the system, what is the last change you made? If others have access to it, what is the last change they made and when? When was the last time the system worked? Often, system changes have unintended consequences that may not be immediately noticed. Roll back that change and see what effect it has on the problem.
- **Make a backup.** This applies before you notice problems, as well as after. If you make a complicated software change to a system, having a backup means that you can quickly restore it to the previous settings and start again. When troubleshooting very complex problems, having a configuration that "sort-of" works can be much better than having a mess that doesn't work at all (and that you can't easily restore from memory).
- **The known good**. This idea applies to hardware, as well as software. A ***known good*** is any component that you can replace in a complex system to verify that its counterpart is in good, working condition. For example, you may carry a tested Ethernet cable in a tool kit. If you suspect problems with a cable in the field, you can easily swap out the suspect cable with the known good and see if things improve. This is much faster and less error-prone than re-crimping a cable, and immediately tells you if the change

fixes the problem. Likewise, you may also pack a backup battery, antenna cable, or a CD-ROM with a known good configuration for the system. When fixing complicated problems, saving your work at a given point lets you return to it as a known good, even if the problem is not yet completely solved.

- **Change one variable at a time.** When under pressure to get a failed system back online, it is tempting to jump ahead and change many likely variables at once. If you do, and your changes seem to fix the problem, then you will not understand exactly what led to the problem in the first place. Worse, your changes may fix the original problem, but lead to more unintended consequences that break other parts of the system. By changing your variables one at a time, you can precisely understand what went wrong in the first place, and be able to see the direct effects of the changes you make.
- **Do no harm.** If you don't fully understand how a system works, don't be afraid to call in an expert. If you are not sure if a particular change will damage another part of the system, then either find someone with more experience or devise a way to test your change without doing damage. Putting a penny in place of a fuse may solve the immediate problem, but it may also burn down the building.

It is unlikely that the people who design your network will be on call twenty-four hours per day to fix problems when they arise. Your troubleshooting team will need to have good troubleshooting skills, but may not be competent enough to configure a router from scratch or crimp a piece of LMR-400. It is often much more efficient to have a number of backup components on-hand, and train your team to be able to swap out the entire broken part. This could mean having an access point or router pre-configured and sitting in a locked cabinet, plainly labeled and stored with backup cables and power supplies. Your team can swap out the failed component, and either send the broken part to an expert for repair, or arrange to have another backup sent in. Assuming that the backups are kept secure and are replaced when used, this can save a lot of time for everyone.

Common network problems

Often, connectivity problems come from failed components, adverse weather, or simple misconfiguration. Once your network is connected to the Internet or opened up to the general public, considerable threats will come from the network users themselves. These threats can range from the benign to the outright malevolent, but all will have impact on your network if it is not properly configured. This section looks at some common problems found once your network is used by actual human beings.

Locally hosted websites

If a university hosts its website locally, visitors to the website from outside the campus and the rest of the world will compete with the university's staff for Internet bandwidth. This includes automated access from search engines that periodically ***spider*** your entire site. One solution to this problem is to use split DNS and mirroring. The university mirrors a copy of its websites to a server at, say, a European hosting company, and uses split DNS to direct all users from outside the university network to the mirror site, while users on the university network access the same site locally. Details about how to set this up are provided in chapter three.

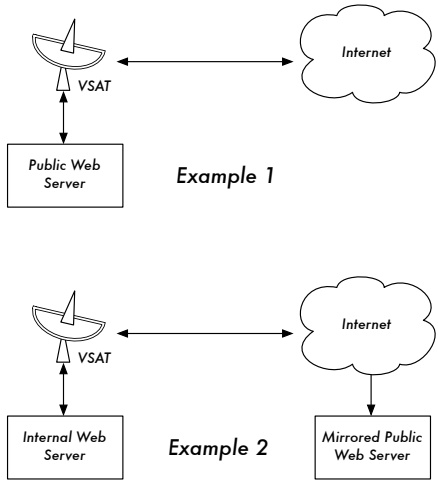

Figure 8.1: In Example 1, all website traffic coming from the Internet must traverse the VSAT. In Example 2, the public web site is hosted on a fast European service, while a copy is kept on an internal server for very fast local access. This improves the VSAT connection and reduces load times for web site users.

Open proxies

A proxy server should be configured to accept only connections from the university network, not from the rest of the Internet. This is because people elsewhere will connect and use open proxies for a variety of reasons, such as to avoid paying for international bandwidth. The way to configure this depends on the proxy server you are using. For example, you can specify the IP address range of the campus network in your **`squid.conf`** file as the only network that can use Squid. Alternatively, if your proxy server lies behind a border firewall, you can configure the firewall to only allow internal hosts to connect to the proxy port.

Open relay hosts

An incorrectly configured mail server will be found by unscrupulous people on the Internet, and be used as a relay host to send bulk email and spam. They do this to hide the true source of the spam, and avoid getting caught. To test for an open relay host, the following test should be carried out on your mail server (or on the SMTP server that acts as a relay host on the perimeter of the campus network). Use ***telnet*** to open a connection to port 25 of the server in question (with some Windows versions of telnet, it may be necessary to type 'set local_echo' before the text is visible):

```
telnet mail.uzz.ac.zz 25
```

Then, if an interactive command-line conversation can take place (for example, as follows), the server is an open relay host:

```
MAIL FROM: spammer@waste.com
250 OK - mail from <spammer@waste.com>
RCPT TO: innocent@university.ac.zz
250 OK - rcpt to spammer@waste.com
```

Instead, the reply after the first MAIL FROM should be something like:

```
550 Relaying is prohibited.
```

An online tester is available at sites such as *http://www.ordb.org/*. There is also information about the problem at this site. Since bulk emailers have automated methods to find such open relay hosts, an institution that does not protect its mail systems is almost guaranteed to be found and abused. Configuring the mail server not to be an open relay consists of specifying the networks and hosts that are allowed to relay mail through them in the MTA (eg., Sendmail, Postfix, Exim, or Exchange). This will likely be the IP address range of the campus network.

Peer-to-peer networking

Bandwidth abuse through peer-to-peer (P2P) file-sharing programs such as Kazaa, Morpheus, WinMX and BearShare can be prevented in the following ways:

- **Make it impossible to install new programs on campus computers.** By not giving regular users administrative access to PC workstations, it is possible to prevent the installation of programs such as Kazaa. Many institutions also standardize on a desktop build, where they install the required operating system on one PC. They then install all the necessary applications on it, and configure these in an optimal way. The PC is also configured in a way that prevents users from installing new applications. A disk

image of this PC is then cloned to all other PCs using software such as Partition Image (see *http://www.partimage.org/*) or Drive Image Pro (see *http://www.powerquest.com/*).

From time to time, users may succeed in installing new software or otherwise damaging the software on the computer (causing it to hang often, for example). When this happens, an administrator can simply put the disk image back, causing the operating system and all software on the computer to be exactly as specified.

- **Blocking these protocols is not a solution.** This is because Kazaa and other protocols are clever enough to bypass blocked ports. Kazaa defaults to port 1214 for the initial connection, but if that is not available it will attempt to use ports 1000 to 4000. If these are blocked, its uses port 80, making it look like web traffic. For this reason, ISPs don't block it but "throttle it", using a bandwidth-manager product (see chapter three).
- **If rate-limiting is not an option, change the network layout.** If the proxy server and mail servers are configured with two network cards (as described in chapter three) and these servers are not configured to forward any packets, this would block all P2P traffic. It would also block all other types of traffic, such as Microsoft NetMeeting, SSH, VPN software, and all other services not specifically permitted by the proxy server. In low bandwidth networks it may be decided that the simplicity of this design will outweigh the disadvantages. Such a decision may be necessary, but shouldn't be taken lightly. Network administrators simply cannot predict how users will make innovative use of a network. By preemptively blocking all access, you will prevent users from making use of any services (even low-bandwidth services) that your proxy does not support. While this may be desirable in extremely low bandwidth circumstances, it should never be considered as a good access policy in the general case.

Programs that install themselves (from the Internet)

There are programs that automatically install themselves and then keep on using bandwidth - for example, the so-called Bonzi-Buddy, the Microsoft Network, and some kinds of worms. Some programs are spyware, which keep sending information about a user's browsing habits to a company somewhere on the Internet. These programs are preventable to some extent by user education and locking down PCs to prevent administrative access for normal users. In other cases, there are software solutions to find and remove these problem programs, such as Spychecker (*http://www.spychecker.com/*), Ad-Aware (*http://www.lavasoft.de/*), or xp-antispy (*http://www.xp-antispy.de/*).

Windows updates

The latest Microsoft Windows operating systems assume that a computer with a LAN connection has a good link to the Internet, and automatically downloads security patches, bug fixes and feature enhancements from the Microsoft Web site. This can consume massive amounts of bandwidth on an expensive Internet link. The two possible approaches to this problem are:

- **Disable Windows updates on all workstation PCs.** The security updates are very important for servers, but whether workstations in a protected private network such as a campus network need them is debatable.
- **Install a Software Update Server.** This is a free program from Microsoft that enables you to download all the updates from Microsoft overnight on to a local server and distribute the updates to client workstations from there. In this way, Windows updates need not use any bandwidth on the Internet link during the day. Unfortunately, all client PCs need to be configured to use the Software Update Server for this to have an effect. If you have a flexible DNS server, you can also configure it to answer requests for *windowsupdate.microsoft.com* and direct the updater to your update server. This is only a good option for large networks, but can save untold amounts of Internet bandwidth.

Blocking the Windows updates site on the proxy server is not a good solution because the Windows update service (Automatic Updates) keeps retrying more aggressively, and if all workstations do that, it places a heavy load on the proxy server. The extract below is from the proxy log (Squid access log) where this was done by blocking Microsoft's cabinet (.cab) files.

Much of the Squid log looks like this:

```
2003.4.2 13:24:17 192.168.1.21 http://windowsupdate.microsoft.com/ident.cab
  *DENIED* Banned extension .cab GET 0
2003.4.2 13:24:18 192.168.1.21 http://windowsupdate.microsoft.com/ident.cab
 *DENIED* Banned extension .cab GET 0
2003.4.2 13:24:18 192.168.1.21 http://windowsupdate.microsoft.com/ident.cab
 *DENIED* Banned extension .cab HEAD 0
2003.4.2 13:24:19 192.168.1.21 http://windowsupdate.microsoft.com/ident.cab
 *DENIED* Banned extension .cab GET 0
2003.4.2 13:24:19 192.168.1.21 http://windowsupdate.microsoft.com/ident.cab
 *DENIED* Banned extension .cab GET 0
2003.4.2 13:24:20 192.168.1.21 http://windowsupdate.microsoft.com/ident.cab
 *DENIED* Banned extension .cab GET 0
2003.4.2 13:24:21 192.168.1.21 http://windowsupdate.microsoft.com/ident.cab
 *DENIED* Banned extension .cab GET 0
2003.4.2 13:24:21 192.168.1.21 http://windowsupdate.microsoft.com/ident.cab
 *DENIED* Banned extension .cab GET 0
2003.4.2 13:24:21 192.168.1.21 http://windowsupdate.microsoft.com/ident.cab
 *DENIED* Banned extension .cab HEAD 0
```

While this may be tolerable for a few PC clients, the problem grows significantly as hosts are added to the network. Rather than forcing the proxy

server to serve requests that will always fail, it makes more sense to redirect the Software Update clients to a local update server.

Programs that assume a high bandwidth link

In addition to Windows updates, many other programs and services assume that bandwidth is not a problem, and therefore consume bandwidth for reasons the user might not predict. For example, anti-virus packages (such as Norton AntiVirus) periodically update themselves automatically and directly from the Internet. It is better if these updates are distributed from a local server.

Other programs, such as the RealNetworks video player, automatically download updates and advertisements, as well as upload usage patterns back to a site on the Internet. Innocuous looking applets (like Konfabulator and Dashboard widgets) continually poll Internet hosts for updated information. These can be low bandwidth requests (like weather or news updates), or very high bandwidth requests (such as webcams). These applications may need to be throttled or blocked altogether.

The latest versions of Windows and Mac OS X also have a time synchronization service. This keeps the computer clock accurate by connecting to time servers on the Internet. It is better to install a local time server and distribute accurate time from there, rather than to tie up the Internet link with these requests.

Windows traffic on the Internet link

Windows computers communicate with each other via ***NetBIOS*** and ***Server Message Block*** (***SMB***). These protocols work on top of TCP/IP or other transport protocols. It is a protocol that works by holding ***elections*** to determine which computer will be the ***master browser***. The master browser is a computer that keeps a list of all the computers, shares and printers that you can see in ***Network Neighbourhood*** or ***My Network Places***. Information about available shares are also broadcast at regular intervals.

The SMB protocol is designed for LANs and causes problems when the Windows computer is connected to the Internet. Unless SMB traffic is filtered, it will also tend to spread to the Internet link, wasting the organization's bandwidth. The following steps might be taken to prevent this:

- **Block outgoing SMB/NetBIOS traffic on the perimeter router or firewall.** This traffic will eat up Internet bandwidth, and worse, poses a potential security risk. Many Internet worms and penetration tools actively scan for open SMB shares, and will exploit these connections to gain greater access to your network.

- **Install ZoneAlarm on all workstations (not the server)**. A free version can be found at *http://www.zonelabs.com/*. This program allows the user to determine which applications can make connections to the Internet and which ones cannot. For example, Internet Explorer needs to connect to the Internet, but Windows Explorer does not. ZoneAlarm can block Windows Explorer from doing so.
- **Reduce network shares.** Ideally, only the file server should have any shares. You can use a tool such as SoftPerfect Network Scanner (from *http://www.softperfect.com/*) to easily identify all the shares in your network.

Worms and viruses

Worms and viruses can generate enormous amounts of traffic. The W32/Opaserv worm, for example, is still prevalent, even though it is an old one. It spreads through Windows shares and is detected by other people on the Internet because it attempts to spread further. It is therefore essential that anti-virus protection is installed on all PCs. Furthermore, user education about executing attachments and responding to unsolicited email is essential. In fact, it should be a policy that no workstation or server should run unused services. A PC should not have shares unless it is a file server; and a server should not run unnecessary services either. For example, Windows and Unix servers typically run a web server service by default. This should be disabled if that server has a different function; the fewer services a computer runs, the less there is to exploit.

Email forwarding loops

Occasionally, a single user making a mistake can cause a problem. For example, a user whose university account is configured to forward all mail to her Yahoo account. The user goes on holiday. All emails sent to her in her absence are still forwarded to her Yahoo account, which can grow to only 2 MB. When the Yahoo account becomes full, it starts bouncing the emails back to the university account, which immediately forwards it back to the Yahoo account. An email loop is formed that might send hundreds of thousands of eemailmails back and forth, generating massive traffic and crashing mail servers.

There are features of mail server programs that can recognize loops. These should be turned on by default. Administrators must also take care that they do not turn this feature off by mistake, or install an SMTP forwarder that modifies mail headers in such a way that the mail server does not recognize the mail loop.

Large downloads

A user may start several simultaneous downloads, or download large files such as 650MB ISO images. In this way, a single user can use up most of the bandwidth. The solutions to this kind of problem lie in training, offline downloading, and monitoring (including real-time monitoring, as outlined in chapter six). Offline downloading can be implemented in at least two ways:

- At the University of Moratuwa, a system was implemented using URL redirection. Users accessing **ftp://** URLs are served a directory listing in which each file has two links: one for normal downloading, and the other for offline downloading. If the offline link is selected, the specified file is queued for later download and the user notified by email when the download is complete. The system keeps a cache of recently downloaded files, and retrieves such files immediately when requested again. The download queue is sorted by file size. Therefore, small files are downloaded first. As some bandwidth is allocated to this system even during peak hours, users requesting small files may receive them within minutes, sometimes even faster than an online download.
- Another approach would be to create a web interface where users enter the URL of the file they want to download. This is then downloaded overnight using a ***cron job*** or scheduled task. This system would only work for users who are not impatient, and are familiar with what file sizes would be problematic for download during the working day.

Sending large files

When users need to transfer large files to collaborators elsewhere on the Internet, they should be shown how to schedule the upload. In Windows, an upload to a remote FTP server can be done using an FTP script file, which is a text file containing FTP commands, similar to the following (saved as **`c:\ftpscript.txt`**):

```
open ftp.ed.ac.uk
gventer
mysecretword
delete data.zip
binary
put data.zip
quit
```

To execute, type this from the command prompt:

```
ftp -s:c:\ftpscript.txt
```

On Windows NT, 2000 and XP computers, the command can be saved into a file such as **`transfer.cmd`**, and scheduled to run at night using the Sched-

uled Tasks (Start ➜ Settings ➜ Control Panel ➜ Scheduled Tasks). In Unix, the same can be achieved by using ***at*** or ***cron***.

Users sending each other files

Users often need to send each other large files. It is a waste of bandwidth to send these via the Internet if the recipient is local. A file share should be created on the local Windows / Samba /web Novell server, where a user can put the large file for others to access.

Alternatively, a web front-end can be written for a local web server to accept a large file and place it in a download area. After uploading it to the web server, the user receives a URL for the file. He can then give that URL to his local or international collaborators, and when they access that URL they can download it. This is what the University of Bristol has done with their FLUFF system. The University offers a facility for the upload of large files (FLUFF) available from *http://www.bristol.ac.uk/fluff/*. These files can then be accessed by anyone who has been given their location. The advantage of this approach is that users can give external users access to their files, whereas the file share method can work only for users within the campus network. A system like this can easily be implemented as a CGI script using Python and Apache.

9
Case Studies

No matter how much planning goes into building a link or node location, you will inevitably have to jump in and actually install something. This is the moment of truth that demonstrates just how accurate your estimates and predictions prove to be.

It is a rare day when everything goes precisely as planned. Even after you install your 1st, 10th, or 100th node, you will still find that things do not always work out as you might have intended. This chapter describes some of our more memorable network projects. Whether you are about to embark on your first wireless project or you are an old hand at this, it is reassuring to remember that there is always more to learn.

General advice

The economies of developing countries are very different from the developed world, and thus a process or solution designed for a more developed country may not be suitable in West Africa, or Southern Asia. Specifically, the cost of locally produced materials and the cost of labour will be negligible, whereas imported goods can be much more expensive when compared to its cost in the developed world. For example, one can manufacture and install a tower for a tenth of the cost of a tower in the United States, but the price of an antenna might be double. Solutions that capitalize on local competitive advantages, namely cheap labour and locally found materials, will be the easiest to replicate.

Finding the right equipment is one of the most difficult tasks in developing markets. Because transportation, communication and economic systems are not developed, the right materials or equipment can be difficult and often impossible to find. A fuse, for example, is difficult to find, thus finding wire that has a burn-up at a certain amperage and can substitute is a great advantage.

Finding local substitutes for materials also encourages local entrepreneurship, ownership, and can save money.

Equipment enclosures

Cheap plastics are everywhere in the developing world, but they are made of poor materials and are thin, thus mostly unsuitable for enclosing equipment. PVC tubing is far more resilient and is made to be waterproof. In West Africa, the most common PVC is found in plumbing, sized from 90mm to 220mm. Access points such as the Routerboard 500 and 200 can fit into such tubing, and with end-caps that are torched-on, they can make very robust waterproof enclosures. They also have the added benefit of being aerodynamic and uninteresting to passers-by. The resulting space left around the equipment assures adequate air circulation. Also, it is often best to leave an exhaust hole at the bottom of the PVC enclosure. The author did find that leaving open holes can become a problem. In one instance ants decided to nest 25 meters above ground inside the PVC holding the access point. Using a wire mesh cover made from locally available screen material is advised to secure the exhaust hole from infestations.

Antenna masts

Recovering used materials has become an important industry for the poorest countries. From old cars to televisions, any material that has value will be stripped, sold, or re-used. For example, you will see vehicles torn apart piece by piece and day by day. The resulting metal is sorted and then tossed into a truck to be sold. Local metal workers will already be familiar with how to make television masts from scrap metal. A few quick adaptations and these same masts can be re-purposed for wireless networks.

The typical mast is the 5 meter pole, comprised of a single 30mm diameter pipe which is then planted into cement. It's best to construct the mast in two parts, with a removable mast that fits into a base which is slightly larger in diameter. Alternately, the mast may be made with arms that can be securely cemented into a wall. This project is easy, but requires the use of a ladder to complete and therefore some caution is suggested.

This type of mast can be augmented by several meters with the use of guy lines. To sturdy the pole, plant three lines 120 degrees apart, at a decline of at least 33 degrees from the tip of the tower.

Above all: involve the local community

Community involvement is imperative in assuring the success and sustainability of a project. Involving the community in a project can be the greatest challenge, but if the community is not involved the technology will not serve

their needs, nor will it be accepted. Moreover, a community might be afraid and could subvert an initiative. Regardless of the complexity of the undertaking, a successful project needs support and buy-in from those it will serve.

An effective strategy in gaining support is to find a respected champion whose motives are palatable. Find the person, or persons whom are most likely to be interested in the project. Often, you will need to involve such champions as advisors, or as members of a steering committee. These people will already have the trust of the community, will know who to approach, and can speak the language of the community. Take your time and be selective in finding the right people for your project. No other decision will affect your project more than having effective, trusted local people on your team.

In addition, take note of key players in an institution, or community. Identify those people whom are likely to be opponents and proponents of your project. As early as possible, attempt to earn the support of the potential proponents and to diffuse the opponents. This is a difficult task and one that requires intimate knowledge of the institution or community. If the project does not have a local ally, the project must take time to acquire this knowledge and trust from the community.

Be careful in choosing your allies. A "town-hall" meeting is often useful to see local politics, alliances, and feuds in play. Thereafter, it is easier to decide on whom to ally, champion and whom to diffuse. Try to not build unwarranted enthusiasm. It is important to be honest, frank, and not to make promises that you cannot keep.

In largely illiterate communities, focus on digital to analog services such as Internet for radio stations, printing on-line articles and photos, and other non-textual applications. Do not try to introduce a technology to a community without understanding which applications will truly will serve the community. Often the community will have little idea how new technologies will help their problems. Simply providing new features is useless without an understanding of how the community will benefit.

When gathering information, verify the facts that you are given. If you want to know the financial status of a company/organization, ask to see an electricity bill, or phone bill. Have they been paying their bills? At times, potential beneficiaries will compromise their own values in hopes of winning funds or equipment. Most often, local partners who trust you will be very frank, honest, and helpful.

Another common pitfall is what I call "divorced parents" syndrome, where NGOs, donors, and partners are not told of each others involvement with the beneficiary. Savvy beneficiaries can earn handsome rewards by letting NGOs and donors lavish them with equipment, training and funds. It is im-

portant to know which other organizations are involved so you can understand how their activities might impact your own. For example, I once designed a project for a rural school in Mali. My team installed an open source system with used computers and spent several days training people how to use it. The project was deemed a success, but shortly after the installation, another donor arrived with brand-new Pentium 4 computers running Windows XP. The students quickly abandoned the older computers and lined-up to use the new computers. It would have been better to negotiate with the school in advance, to know their commitment to the project. If they had been frank, the computers that are now sitting unused could have been deployed to another school where they would be used.

In many rural communities in under-developed economies, law and policies are weak, and contracts can be effectively meaningless. Often, other assurances must be found. This is where pre-paid services are ideal, as they do not require a legal contract. Commitment is assured by the investment of funds before service is given.

Buy-in also requires that those involved invest in the project themselves. A project should ask for reciprocal involvement from the community.

Above all, the “no-go” option should always be evaluated. If a local ally and community buy-in cannot be had, the project should consider choosing a different community or beneficiary. There must be a negotiation; equipment, money, and training cannot be gifts. The community must be involved and they too must contribute.

—Ian Howard

Case study: Crossing the divide with a simple bridge in Timbuktu

Networks ultimately connect people together, and therefore always involve a political component. The cost of Internet in less developed economies is high and the ability to pay is low, which adds to the political challenges. Attempting to superimpose a network where human networks are not fully functioning is nearly impossible in the long term. Trying to do so can leave a project on unstable social ground, threatening its existence. This is where the low cost and mobility of a wireless network can be advantageous.

The author's team was asked by funders to determine how to connect a rural radio station with a very small (2 computer) telecentre to the Internet in Timbuktu, the desert capital of Mali. Timbuktu is widely known as an outpost in the most remote area of the world. At this site, the team decided to imple-

ment a model which has been called the ***parasitic wireless model***. This model takes a wireless "feed" that is spliced from an existing network, and extends that network to a client site using a simple bridged network. This model was chosen because it requires no significant investment by the supporting organization. While it added a source of revenue for the telecentre, it did not add a significant operational cost. This solution meant that the client site could get cheap Internet, albeit not as fast or as reliable as a dedicated solution. Because of opposed usage patterns between an office and a telecentre there was no perceptible slowing of the network for either party. Though in an ideal situation it would be best to encourage more development of the small telecentre into an ISP, neither the telecentre nor the market were deemed ready. As is often the case, there were serious concerns about whether this telecentre could become self-sustaining once its funders departed. Thus, this solution minimized the initial investment while achieving two goals: first, it extended the Internet to the target beneficiary, a radio station, at an affordable cost. Second, it added a small additional revenue source for the telecentre while not increasing its operational costs, or adding complexity to the system.

The people

Timbuktu is remote, though having a world renowned name. Being a symbol of remoteness, many projects have wanted to "stake a flag" in the sands of this desert city. Thus, there are a number of information and communications technologies (ICT) activities in the area. At last count there were 8 satellite connections into Timbuktu, most of which service special interests except for the two carriers, SOTELMA and Ikatel. They currently use VSAT to link their telephone networks to the rest of the country. This telecentre used an X.25 connection to one of these telcos, which then relayed the connection back to Bamako. Relative to other remote cities in the country, Timbuktu has a fair number of trained IT staff, three existing telecentres, plus the newly installed telecentre at the radio station. The city is to some degree over saturated with Internet, precluding any private, commercial interests from being sustainable.

Design Choices

In this installation the client site is only 1 km away directly by line of sight. Two modified Linksys access points, flashed with OpenWRT and set to bridge mode, were installed. One was installed on the wall of the telecentre, and the other was installed 5 meters up the radio station's mast. The only configuration parameters required on both devices were the ssid and the channel. Simple 14 dBi panel antennas (from *http://hyperlinktech.com/*) were used. At the Internet side, the access point and antenna were fastened using cement plugs and screws onto the side of the building, facing the client site. At the client site, an existing antenna mast was used. The access point and antenna were mounted using pipe rings.

To disconnect the client, the telecentre simply unplugs the bridge on their side. An additional site will eventually be installed, and it too will have its own bridge at the telecentre so that staff can physically disconnect the client if they have not paid. Though crude, this solution is effective and reduces risk that the staff would make a mistake while making changes to the configuration of the system. Having a bridge dedicated to one connection also simplified installation at the central site. as the installation team was able to choose the best spot for connecting the client sites. Though it is not optimal to bridge a network (rather than route network traffic), when technology knowledge is low and one wants to install a very simple system this can be a reasonable solution for small networks. The bridge makes systems installed at the remote site (the radio station) appear as though they are simply connected to the local network.

Financial model

The financial model here is simple. The telecentre charges a monthly fee, about $30 per connected computer to the radio station. This was many times cheaper than the alternative. The telecentre is located in the court of the Mayor's office, so the principle client of the telecentre is the Mayor's staff. This was important because the radio station did not want to compete for clientele with the telecentre and the radio station's systems were primarily intended for the radio station staff. This quick bridge reduced costs, meaning that this selective client base could support the cost of the Internet without competing with the telecentre, its supplier. The telecentre also has the ability to easily disconnect the radio station should they not pay. This model also allowed sharing of network resources. For example, the radio station has a new laser printer, while the telecentre has a colour printer. Because the client systems are on the same network, clients can print at either site.

Training

To support this network, very little training was required. The telecentre staff were shown how to install the equipment and basic trouble shooting, such as rebooting (power cycling) the access points, and how to replace the unit should one fail. This allows the author's team to simply ship a replacement and avoid the two day trek to Timbuktu.

Summary

The installation was considered an interim measure. It was meant to serve as a stop-gap measure while moving forward with a more complete solution. While it can be considered a success, it has not yet led to building more physical infrastructure. It has brought ICTs closer to a radio solution, and reenforced local client/supplier relationships.

As it stands, Internet access is still an expensive undertaking in Timbuktu. Local politics and competing subsidized initiatives are underway, but this simple solution has proven to be an ideal use case. It took the team several months of analysis and critical thought to arrive here, but it seems the simplest solution provided the most benefit.

—Ian Howard

Case study: Finding solid ground in Gao

One day's drive east from Timbuktu, in Eastern Mali, is Gao. This rural city, which seems more more like a big village, sits up the the river Niger just before it dips South crossing into Niger and onto Nigeria. The city slopes into the river gently, and has few buildings taller than two stories. In 2004, a telecentre was installed in Gao. The project's goal was to provide information to the community in the hope that a better informed community would yield a healthier and more educated citizenry.

The centre provides information via CD-ROMs, films and radio, but the cornucopic source of information for the centre is the Internet. It is a standard telecentre, with 8 computers, an all-in-one printer, scanner, fax, a telephone and a digital camera. A small two room building was built to house the telecentre. It is located a bit outside of downtown, which is not an ideal location for attracting customers, but the site was chosen because of its sympathetic host. The site received funding for all construction needed, and equipment and initial training was supplied as well. The telecentre was expected to be self-sustaining after one year.

Several months after its opening, the telecentre was attracting few customers. It used a modem to dial-up to connect to an Internet provider in the capital. This connection was too slow and unreliable, and so the funder sponsored the installation of a VSAT system. There are a number of VSAT systems now available to the region; most of these services have just recently become available. Previously only C-band (which cover a larger area than Ku-band) systems were available. Recently, fiber has been laid in almost every subway tunnel and canal throughout Europe, and thus it has supplanted the more expensive satellite services. As a result, providers are now redirecting their VSAT systems to new markets, including middle and Western Africa, and South Asia. This has led to a number of projects which use satellite systems for an Internet connection.

After the VSAT was installed, the connection provided 128 Kbps down and 64 Kbps up, and cost about $400 per month. The site was having trouble earning enough revenue to pay for this high monthly cost, so the telecentre asked for help. A private contractor was hired, who had been trained by the

author to install a wireless system. This system would split the connection between three clients: a second beneficiary, a radio station, and the telecentre, each paying $140. This collectively covered the costs of the VSAT, and the extra revenue from the telecentre and the radio station would cover support and administration of the system.

The people

Though capable and willing, the author's team did not do the actual installation. Instead, we encouraged the telecentre to hire the local contractor to do it. We were able to reassure the client by agreeing to train and support the contractor in the fulfillment of this installation. The premise of this decision was to discourage a reliance on a short-term NGO, and rather to build trust and relationships between domestic service providers and their clients. This design proved to be fruitful. This approach took much more time from the author's team, perhaps twice as much, but this investment has already begun to pay-off. Networks are still being installed and the author and his team are now home in Europe and North America.

Design choices

Initially, it was conceived that a backbone connection would be made to the radio station, which already had a 25 meter tower. That tower would be used to relay to the other clients, avoiding the need to install towers at the client sites, as this tower was well above any obstacles in the city. To do this, three approaches were discussed: installing an access point in repeater mode, using the WDS protocol, or using a mesh routing protocol. A repeater was not desirable as it would introduce latency (due to the one-armed repeater problem) to an already slow connection. VSAT connections need to send packets up to the satellite and back down, often introducing up to 3000 ms in delay for a round trip. To avoid this problem, it was decided to use one radio to connect to clients, and a second radio for to the dedicated backbone connection. For simplicity it was decided to make that link a simple bridge, so that the access point at the radio station would appear to be on the same physical LAN as the telecentre.

In testing this approach functioned, though in the real world, its performance was dismal. After many different changes, including replacing the access points, the technician decided that there must be a software or hardware bug affecting this design. The installer then decided to place the access point at the telecentre directly using a small 3 meter mast, and to not use a relay site at the radio station. The client sites also required small masts in this design. All sites were able to connect, though the connections were at times too feeble, and introduced massive packet loss.

Later, during the dust season, these connections became more erratic and even less stable. The client sites were 2 to 5 km away, using 802.11b. The team theorized that the towers on either side were too short, cutting off too much of the Fresnel zone. After discussing many theories, the team also realized the problem with the performance at the radio station: the radio frequency 90.0 MHz was about the same as the frequency of the high-speed (100BT) Ethernet connection. While transmitting, the FM signal (at 500 watts) was completely consuming the signal on the Ethernet cable. Thus, shielded cable would be required, or the frequency of the Ethernet link would need to be changed. The masts were then raised, and at the radio station the speed of the Ethernet was changed to 10 Mbps. This changed the frequency on the wire to 20 MHz, and so avoided interference from the FM transmission. These changes resolved both problems, increasing the strength and reliability of the network. The advantage of using mesh or WDS here would be that client sites could connect to either access point, either directly to the telecentre to the radio station. Eventually, removing the reliance on the radio station as a repeater likely made the installation more stable in the longer-term.

Financial model

The satellite system used at this site cost approximately $400 per month. For many IT for Development projects this expensive monthly cost is difficult to manage. Typically these projects can purchase equipment and pay for the establishment of a wireless network, but most are not able to pay for the cost of the network after a short period of time (including the recurring Internet costs and operational costs). It is necessary to find a model where the monthly costs for a network can be met by those who use. For most community telecenters or radio stations, this is simply too expensive. Often, the only feasible plan is to share the costs with other users. To make the Internet more affordable, this site used wireless to share the Internet to the community, allowing a greater number of organizations to access the Internet while reducing the cost per client.

Typically in Mali, a rural community has only a few organizations or companies that could afford an Internet connection. Where there are few clients, and the Internet connection cost is high, the model developed by his team included ***anchor clients***: clients whom are solid and are low-risk. For this region, foreign NGOs (Non Governmental Organizations), the United Nations Agencies and large commercial enterprises are among the very few whom qualify.

Among the clients selected for this project were three anchor clients, who collectively paid the entire monthly cost of the satellite connection. A second beneficiary, a community radio station, was also connected. Any revenue earned from the beneficiaries contributed to a windfall, or deposit for future costs, but was not counted upon due to the small margins that both of these

community services operated on. Those clients could be disconnected and could resume their service once they can afford it again.

Training needed: who, what, for how long

The contractor taught the telecentre technician the basics of supporting the network, which was fairly rudimentary. Any non-routine work, such as adding a new client, was contracted out. Therefore it was not imperative to teach the telecentre staff how to support the system in its entirety.

Lessons learned

By sharing the connection, the telecentre is now self-sustaining, and in addition, three other sites have Internet access. Though it takes more time and perhaps more money, it is valuable to find the right local talent and to encourage them to build relationships with clients. A local implementor will be able to provide the follow-up support needed to maintain and expand a network. This activity is building local expertise, and demand, which will allow subsequent ICT projects to build on this base.

—Ian Howard

Case Study: Spectropolis, New York

In September 2003 and October 2004, NYCwireless produced Spectropolis. This event celebrated the availability of open wireless (Wi-Fi) networks in Lower Manhattan and explored their implications for art, community, and shared space. Spectropolis is the world's first wireless arts festival, and was envisioned as a way to bring the techno-centric nature of Wi-Fi into a more accessible form. The idea was to create a way for average residents and visitors to New York to "see" and "feel" the wireless signals that permeate the city (especially the free Wi-Fi that NYCwireless provides in many city parks) that are otherwise invisible.

The idea for Spectropolis came from a series of discussions in the winter of 2003 between Dana Spiegel, then a member-at-large for NYCwireless, and Brooke Singer, an independent New Media artist and associate professor at SUNY Purchase.

Spectropolis took place at City Hall Park, a well-known free wireless hotspot in New York City, New York. The festival featured works of art from 12 international artists. Each art piece integrated and made use of one or more forms of wireless technology, including Wi-Fi, Bluetooth, Radio, GPS, and others. Each piece was intended to explore how wireless technologies affect our everyday urban experiences. The pieces were exhibited outdoors in the

park for three days, and the artists were out exhibiting artwork and explaining their work to park visitors.

In addition to the works of art, Spectropolis offered five workshops and three panel discussions. The workshops offered an up-close look at wireless communication technologies and an opportunity for hands-on play and participation. The workshops aimed to educate both the technical and non-technical public and demystify a range of technologies through engaging presentations.

The panels explored the larger scale implications of wireless technologies for society, public policy, activism, and art. Each panel focussed on a particular area of influence for wireless technology, with commentary by a number of recognized leaders.

An outdoor park/public space was chosen for the event primarily because this location provided a way to both attract a large number of attendees as well as situate the event in a space that many people pass through both during the workday and on the weekend. One of the goals of the event was to reach out to local residents and people who wouldn't otherwise attend a technology-centric event. During the time that Spectropolis was in City Hall Park, thousands of people came through the park each day, and many stopped to look at one or more artworks.

From a visibility point of view, holding Spectropolis in an outdoor public space was important, and the foot traffic around the area definitely resulted in attracting a number of people into the park who would otherwise not have come to the event. In addition, New York City has a long history of outdoor public art, however this art is almost entirely sculptural in form, and meant to participate in the landscape but not really be interactive. Bringing highly interactive new media art from a museum or gallery into an outdoor public space created discordance with people's expectations.

Why Spectropolis is important

Spectropolis is an attempt to give wireless technology and Wi-Fi in particular a life beyond email and websurfing. The interactive works of art showcased at Spectropolis are engaging beyond the “work-use” that is associated with Wi-Fi by the general public. By introducing wireless technologies via “play” and “exploration”, Spectropolis removes much of the fear that people have about new technologies, and enables people to consider the larger implications of wireless technologies and their lives without getting caught up in the “how” of the technology itself.

Spectropolis is a unique event because it focuses on the social impact of wireless technologies, as opposed to the technologies themselves. The vast

majority of people are either scared by raw technology (this is common in adults more than children) or are merely disinterested. While Wi-Fi and cellular technologies have made significant inroads into general society, they have done so by riding on the coat-tails of two well established social activities: talking on the phone and accessing the Internet (email, web, IM, etc.)

In addition, Spectropolis puts a face on the ethereal nature of wireless signals. That Wi-Fi is available in a park may be indicated by signs and stickers on windows, but creating a tangible artifact in the form of works of art drives this concept home in the same way that benches, trees, and grass showcase the public amenities that a park provides. Wi-Fi in public spaces isn't a gated community, but rather a public resource that can be shared and appreciated by all just like the shade of a large tree.

Participating organizations

NYCwireless, through Dana Spiegel, took on the role of producing Spectropolis. NYCwireless is a non-profit organization that advocates and enables the growth of free, public wireless Internet access in New York City and surrounding areas. NYCwireless, founded in 2001, is an all-volunteer organization with seven board members, five special interest working groups and approximately sixty active members.

NYCwireless partnered with other local organizations and prominent individuals from the New York Arts community who volunteered their time to help curate and produce the event. Spectropolis was sponsored by the Alliance for Downtown New York (DTA), a Business Improvement District (BID) company. The DTA also sponsors a number of free, public, wireless hotspots in downtown New York, including the hotspot at City Hall Park, where Spectropolis was held. The Lower Manhattan Cultural Council (LMCC), an arts funding and promotion organization, sponsored the curation of Spectropolis. LMCC hosted a number of meetings and oversaw the process of inviting and evaluating artists and their works in preparation of the event. In addition, a number of individuals contributed a significant amount of time to Spectropolis: Wayne Ashley (Curator, LMCC), Yury Gitman (Curator), Jordan Silbert (Producer), and Jordan Schuster (Producer)

Community Reception

The local community received Spectropolis quite well. The primary groups of people who attended the event were: wireless researchers, wireless proponents, artists, and the general public.

Leading up to the event, we reached out to the local artist and local university communities to generate interest. We received a large number of email inquiries from people both locally and around the continent (primarily US and

Canada) about attending the event. Some wireless enthusiasts even traveled from Europe in order to attend. The local university community was particularly interested, with students from NYU, SUNY, New School, Parsons, and other nearby schools attending. During the event we even had a few people bring their own projects to the park and set them up.

We also sent out a press release to local media outlets and websites to inform the general NYC community about the event. While we weren't contacted prior to the event by anyone from the general public, there were some people from this group who signed up for our workshops and our panels who had not ever handled wireless equipment. Primarily, local residents and visitors just showed up to the event to experience the artwork. We had thousands of people each day come through the park and experience at least a few of the works.

In addition to the art, we had a number of people ask questions about wireless technology in general, and public Wi-Fi in specific. Many of these people were directed to the NYCwireless information booth that was set up in the middle of the park. A number did speak directly to the artists (we had expected this, and this was one of the reasons why we wanted artists to show their own work) about the works they created, and ask about how they worked and why the artist created the work.

For a number of attendees, Spectropolis was the first time they experienced Wi-Fi as something more than just an Internet technology. Many were surprised that wireless technologies could be more than just a cell phone call or a web page in a cafe, and they were pleased to get a better grasp on the alternative uses for Wi-Fi that the art works explored. In some instances, the relationship between wireless signals and the works of art were hidden and obscure--such as Akitsugu Maebayashi's Sonic Interface. In other works, like Upper Air by the DSP Music Syndicate, the art was designed to support the existence of the wireless technology, and the piece explored the the technology's relationship to both the viewer and the art.

Some pieces, such as Jabberwocky by Eric John Paulos and Elizabeth Goodman, made use of the technology to explore social relationships in urban environments. These works were important and meaningful because they related wireless technology to something that is clearly a human experience, such as seeing familiar strangers in a crowd. In Jabberwocky in particular, the viewer is forced to see also the limits of the wireless technology, and make use of human abilities to fill in the gaps.

GPS drawings, a workshop held by Jeremy Wood, extended the notion of humans + technology equaling something greater than the sum of its parts. Wood actually led groups of people around parts of downtown New York City

to create large scale drawings out of their movements. This artwork personalized the experience of wireless technologies more than any other project.

All of the projects forced people to re-evaluate their relationships with their technologies. More than just seeing public spectrum and wireless networks in a new light, Spectropolis caused people to think about how these technologies enrich and permeate their lives. In speaking with artists after the event, all of them were surprised by how engaged people were. People who interacted with the artworks had a better understanding of the otherwise ephemeral nature of wireless signals. For visitors to the event, Spectropolis made abstract concepts of spectrum and public wireless much more concrete, and gave them a way to understand these concepts in a way that merely using a cell phone or Wi-Fi laptop could not, and in this way, Spectropolis was a complete success.

Projects

Spectropolis featured the following projects and artists:

- **WiFi Ephemera Cache** by Julian Bleecker,
- **UMBRELLA.net** by Jonah Brucker-Cohen and Katherine Moriwaki
- **Microrardio Sound Walk** by free103point9 Transmission Artists
- **Urballoon** by Carlos J. Gomez de Llarena
- **Bikes Against Bush** by Joshua Kinberg
- **InterUrban** by Jeff Knowlton and Naomi Spellman
- **Hotspot Bloom** by Karen Lee
- **Sonic Interface** by Akitsugu Maebayashi
- **Jabberwocky** by Eric John Paulos and Elizabeth Goodman
- **Upper Air** by The DSP Music Syndicate
- **Twenty-Four Dollar Island** by Trebor Scholz
- **Text Messaging Service** and **Following 'The Man of the Crowd'** by Dodgeball + Glowlab

Planning

The planning for Spectropolis began about one year prior to the event. At the outset, representatives from NYCwireless, LMCC, and DTA, as well as the producers and curators, met on a monthly basis to establish the plan and execute the event. The cost of producing Spectropolis was about $11,000 USD.

More information can be found on the Spectropolis 2004 website at *http://www.spectropolis.info/* and at my Wireless Community blog at *http://www.wirelesscommunity.info/spectropolis.*

—Dana Spiegel

Case study: The quest for affordable Internet in rural Mali

For several years the international development community has promoted the idea of closing the digital divide. This invisible chasm that has formed separating access to the wealth of information and communications technologies (ICT) between the developed and the developing world. Access to information and communications tools has been shown to have a dramatic impact on quality of life. For many donors fatigued by decades of supporting traditional development activities, the installation of a telecentre in the developing world seems like a realizable and worthwhile effort. Because the infrastructure does not exist, this is much more expensive and difficult to do in the developing world than it is in the West. Moreover, few models have been shown to sustain these activities. To help mitigate some of the cost of bringing the Internet to rural areas of the developed world, the author's team has promoted the use of wireless systems to share the cost of an Internet connection. In November of 2004, an affiliated project asked the author's team to pilot such a wireless system at a recently installed telecentre in rural Mali, 8 hours South-West by four-by-four from Bamako, the capital.

This rural city, located on the margin of a man-made reservoir, holds water for the Manitali dam that powers a third of the country. This location is fortunate as hydroelectric power is much more stable and available than diesel generated power. While diesel generated power is far less stable, some rural communities are lucky to have any electricity at all.

The city is also endowed to be in one of the most fertile regions of the country, in its cotton belt, Mali's main cash crop. It was believed that this site would be the least difficult of the rural areas in Mali to make a self-sustaining telecentre. Like many experiments, this pilot was fraught with challenges.

Technologically it was a simple task. In 24 hours the team installed an 802.11b wireless network that shares the telecenter's VSAT Internet connection with 5 other local services: the Mayor, the Governor, the health service, the district's Mayor's council (CC) and the community advisory service (CCC).

These clients had been selected during a reconnaissance two months prior. During that visit the team had interviewed potential clients and determined which clients could be connected without complicated or expensive installations. The telecentre itself is housed at the community radio station. Radio stations tend to be great sites to host wireless networks in rural Mali as they are often well placed, have electricity, security and people who understand at least the basics of radio transmissions. They are also natural hubs for a village. Providing Internet to a radio station provides better information to its listeners. And for a culture which is principally oral, radio happens to be the most effect means to provide information.

From the list of clients above, you will note that the clients were all government or para-governmental. This proved to be a difficult mix, as there is considerable animosity and resentment between the various levels of government, and there were continuing disputes regarding taxes and other fiscal matters. Fortunately the director of the radio station, the network's champion, was very dynamic and was able to wade through most of these politics, though not all.

Design choices

The technical team determined that the access point would be installed at 20 meters up the radio station tower, just below the FM radio dipoles, and not so high as to interfere with coverage to client sites below in the bowl-like depression where most were found. The team then focused on how to connect each client site to this site. An 8 dBi omni (from Hyperlinktech, *http://hyperlinktech.com/*) would suffice, providing coverage to all client sites. The 8 dBi antenna that was chosen has a 15 degree down-tilt, assuring that the two clients less than a kilometer away could still receive a strong signal. Some antennae have very narrow beam width and thus "overshoot" sites that are close. Panel antennae were considered, though at least two would be required and either a second radio or a channel splitter. It was deemed unnecessary for this installation. The following calculation shows how to calculate the angle between the client site's antenna and the base station's antenna, using standard trigonometry.

```
tan(x) = difference in elevation
       + height of base station antenna
       - height of CPE antenna
       / distance between the sites

    tan(x) = 5m + 20m - 3m / 400m
         x = tan-1 (22m / 400m)
         x =~ 3 degrees
```

In addition to the equipment in the telecentre (4 computers, a laser printer, 16 port switch), the radio station itself has one Linux workstation installed by the

author's project for audio editing. A small switch was installed in the radio station, an Ethernet cable was run through plastic tubing buried at 5 cm across to the telecentre, across the yard.

From the main switch, two cables run up to a Mikrotik RB220, access point. The RB220 has two Ethernet ports, one that connects to the VSAT through a cross-over cable, and the second that connects to the radio station's central switch. The RB 220 is housed in a D-I-Y PVC enclosure and an 8 dBi omni (Hyperlink Technologies) is mounted directly to the top of the PVC cap.

The RB220, runs a derivative of Linux, Mikrotik version 2.8.27. It controls the network and provides DHCP, firewall, DNS-caching and routes traffic to the VSAT, using NAT. The Mikrotik comes with a powerful command line and a relatively friendly and comprehensive graphical interface. It is a small x86 based computer, that is designed for use as an access point or embedded computer. These access points are POE capable, have two Ethernet ports, a mini-pci port, two PCMCIA slots, a CF reader (which is used for its NVRAM), are temperature tolerant and support a variety of x86 operating systems. Despite that the Mikrotik software requires licensing, there was already a substantial install base in Mali and the system has a powerful and friendly graphical interface that was superior to other products. Due to the above factors the team agreed to use these systems, including the Mikrotik software to control these networks. The total cost of the RB220, with License Level 5, Altheros mini-pci a/b/g and POE was $461. You can find these parts at Mikrotik online at *http://www.mikrotik.com/routers.php#linx1part0.*

The network was designed to accommodate expansion by segregating the various sub-networks of each client; 24 bit private subnets were alloted. The AP has a virtual interface on each subnet and does all routing between, also allowing fire-walling at the IP layer. Note: this does not provide a firewall at the network layer, thus, using a network sniffer like tcpdump one can see all traffic on the wireless link.

To limit access to subscribers, the network uses MAC level access control. There was little perceived security risk to the network. For this first phase, a more thorough security system was left to be implemented in the future,, when time could be found to find an easier interface for controlling access. Users were encouraged to use secure protocols, such as https, pops, imaps etc.

The affiliate project had installed a C-band VSAT (DVB-S) system. These satellite systems are normally very reliable and are often used by ISPs. It is a large unit, in this case the dish was 2.2 meters in diameter and expensive, costing approximately $12,000 including installation. It is also expensive to operate. A 128 kbps down and 64 kbps up Internet connection costs approximately $700 per month. This system has several advantages compared

to a Ku system though, including: greater resilience to bad weather, lower contention rates (number of competing users on the same service) and it is more efficient at transferring data.

The installation of this VSAT was not ideal. Since the system ran Windows, users were able to quickly change a few settings, including adding a password to the default account. The system had no UPS or battery back up, so once a power outage occurred the system would reboot and sit waiting for a password, which had since been forgotten. To make this situation worse, because the VSAT software was not configured as an automatic background service it did not automatically launch and establish the link. Though the C-band systems are typically reliable, this installation caused needless outages which could have been resolved with the use of a UPS, proper configuration of the VSAT software as a service, and by limiting physical access to the modem. Like all owners of new equipment, the radio station wanted to display it, hence it was not hidden from view. Preferably a space with glass doors would have kept the unit secure while keeping it visible.

The wireless system was fairly simple. All of the client sites selected were within 2 km of the radio station. Each site had a part of the building that could physically see the radio station. At the client site, the team chose to use commercial, client grade CPEs: Based on price, the Powernoc 802.11b CPE bridge, small SuperPass 7 dBi patch antennas and home-made Power Over Ethernet (POE) adaptors. To facilitate installation, the CPE and the patch antenna were mounted on a small piece of wood that could be installed on the outside wall of the building facing the radio station.

In some cases the piece of wood was an angled block to optimize the position of the antenna. Inside, a POE made from a repurposed television signal amplifier (12V) was used to power the units. At the client sites there were not local networks, so the team also had to install cable and hubs to provide Internet for each computer. In some cases it was necessary to install Ethernet adapters and their drivers (this was not determined during the assessment). It was decided that because the client's networks were simple, that it would be easiest to bridge their networks. Should it be required, the IP architecture could allow future partitioning and the CPE equipment supported STA mode. We used a PowerNOC CPE bridge that cost $249 (available at *http://powernoc.us/outdoor_bridge.html*).

Local staff were involved during the installation of the wireless network. They learned everything from wiring to antenna placement. An intensive training program followed the installation. It lasted several weeks, and was meant to teach the staff the day to day tasks, as well as basic network troubleshooting.

A young university graduate who had returned to the community was chosen to support the system, except for the cable installation, which the radio sta-

tion technician quickly learned. Wiring Ethernet networks is very similar to coaxial cable repairs and installations which the radio technician already performed regularly. The young graduate also required little training. The team spent most of its time helping him learn how to support the basics of the system and the telecentre. Soon after the telecentre opened, students were lined up for the computer training, which offered 20 hours of training and Internet use per month for only $40, a bargain compared to the $2 an hour for Internet access. Providing this training was a significant revenue and was a task that the young computer savvy graduate was well suited for.

Unfortunately, and somewhat unsurprisingly, the young graduate left for the capital, Bamako, after receiving an offer for a government job. This left the telecentre effectively marooned. Their most technically savvy member, and the only one who was trained in how to support the system, had left. Most of the knowledge needed to operate the telecentre and network left with him. After much deliberation, the team determined that it was best not to train another tech savvy youth, but rather to focus on the permanent local staff, despite their limited technical experience. This took much more time. Our trainers have had to return for a total of 150 hours of training. Several people were taught each function, and the telecentre support tasks were divided among the staff.

Training did not stop there. Once the community services were connected, they too needed access. It seemed that although they were participating, the principals, including the mayor, were not using the systems themselves. The team realized the importance of assuring that the decision makers used the system, and provided training for them and their staff. This did remove some of the mystique of the network and got the city's decision makers involved.

Following training, the program monitored the site and began to provide input, evaluating ways that this model could be improved. Lessons learned here were applied to other sites.

Financial Model

The community telecentre was already established as a non-profit, and was mandated to be self-sustaining through the sale of its services. The wireless system was included as a supplementary source of revenue because early financial projections for the telecentre indicated that they would fall short of paying for the VSAT connection.

Based on the survey, and in consultation with the radio station whom manages the telecentre, several clients were selected. The radio station negotiated contracts with some support from its funding partner. For this first phase, clients were selected based on ease of installation and expressed

ability to pay. Clients were asked to pay a subscription fee, as described later.

Deciding how much to charge was a major activity which required consultation and expertise that the community did not have in financial projections. The equipment was paid for by the grant, to help offset the costs to the community, but clients were still required to pay a subscription fee, which served to assure their commitment. This was equivalent to one month of the service fee.

To determine the monthly cost for an equal slice of bandwidth we started with the following formula:

```
VSAT + salaries + expenses (electricity, supplies) =
telecentre revenue + wireless client revenue
```

We had estimated that the telecentre should earn about $200 to $300 per month in revenue. Total expenses were estimated to be $1050 per month, and were broken down as: $700 for the VSAT, $100 for salaries, $150 for electricity, and about $100 for supplies. About $750 in revenue from the wireless clients was required to balance this equation. This amounted to roughly $150 from each client. This was just tolerable by the clients, and looked feasible, but required fair weather, and had no room for complications.

Because this was becoming complicated, we brought in business geeks, who modified the formula as such:

```
Monthly expenses + amortization + safety funds = total
revenue
```

The business experts were quick to point out the need of amortization of the equipment, or one could say "re-investment funds" as well as safety funds, to assure that the network can continue if a client defaults, or if some equipment breaks. This added about $150 per month for amortization (equipment valued at about $3,000, amortized over 24 months) and the value of one client for default payments, at $100. Add another 10% to account for currency devaluation ($80), and that equals an expense of $1380 per month. In trying to implement this model, it was finally determined that amortization is a concept that was too difficult to convey to the community, and that they would not consider that clients might default on payment. Thus, both formulae were used, the first by the telecentre and the second for our internal analysis.

As was soon discovered, regular payments are not part of the culture in rural Mali. In an agrarian society everything is seasonal, and so too is income. This means that the community's income fluctuates wildly. Moreover, as many public institutions were involved, they had long budget cycles with little flexibility. Although they theoretically had the budget to pay for their service, it

would take many months for the payments to be made. Other fiscal complications arose as well. For example, the mayor signed on and used the back-taxes owed by the radio to pay for its subscription. This of course did not contribute to cash flow. Unfortunately, the VSAT providers have little flexibility or patience, as they have limited bandwidth and only have room for those that can pay.

Cash flow management became a primary concern. First, the revenue foreseen in financial projections showed that even with an optimistic outlook, they would not only have trouble earning enough revenue on time to pay the fee, but getting the money to the Bamako-based bank also presented a problem. Roads near the village can be dangerous, due to the number of smugglers from Guinea and wayward rebels from the Ivory Coast. As projected, the telecentre was not able to pay for its service and its service was suspended, thereby suspending payment from their clients as well.

Before the project was able to find solutions to these problems, the cost of the VSAT already began to dig the telecentre into debt. After several months, due to technical problems, as well as concerns raised in this analysis, the large C-band VSAT was replaced with a cheaper Ku band system. Although cheaper, it still sufficed for the size of the network. This system was only $450, which by ignoring amortization and safety margins is affordable by the network. Unfortunately, due to default payments, the network was not able to pay for the VSAT connection after the initial subsidized period.

Conclusions

Building a wireless network is relatively easy, but making it work is much more of a business problem than a technical problem. A payment model that considers re-investment and risk is a necessity, or eventually the network will fail. In this case, the payment model was not appropriate as it did not conform to fiscal cycles of the clients, nor did it conform to social expectations. A proper risk analysis would have concluded that a $700 (or even a $450) monthly payment left too narrow a margin between revenue and expenses to compensate for fiscal shortcomings. High demand and education needs limited the expansion of the network.

Following training the network operated for 8 months without significant technical problems. Then, a major power surge caused by a lightning strike destroyed much of the equipment at the station, including the access point and VSAT. As a result, the telecentre was still off-line at the time that this book was written. By that time this formula was finally deemed an unsuitable solution.

—Ian Howard

Case study: Commercial deployments in East Africa

Describing commercial wireless deployments in Tanzania and Kenya, this chapter highlights technical solutions providing solid, 99.5% availability Internet and data connectivity in developing countries. In contrast to projects devoted to ubiquitous access, we focused on delivering services to organizations, typically those with critical international communications needs. I will describe two radically different commercial approaches to wireless data connectivity, summarizing key lessons learned over ten years in East Africa.

Tanzania

In 1995, with Bill Sangiwa, I founded CyberTwiga, one of the first ISPs in Africa. Commercial services, limited to dialup email traffic carried over a 9.6 kbps SITA link (costing over $4000/month!), began in mid-1996. Frustrated by erratic PSTN services, and buoyed by a successful deployment of a 3-node point-multipoint (PMP) network for the Tanzania Harbours authority, we negotiated with a local cellular company to place a PMP base station on their central mast. Connecting a handful of corporations to this WiLan proprietary 2.4 GHz system in late 1998, we validated the market and our technical capacity to provide wireless services.

As competitors haphazardly deployed 2.4 GHz networks, two facts emerged: a healthy market for wireless services existed, but a rising RF noise floor in 2.4 GHz would diminish network quality. Our merger with the cellular carrier, in mid-2000, included plans for a nationwide wireless network built on the existing cellular infrastructure (towers and transmission links) and proprietary RF spectrum allocations.

Infrastructure was in place (cellular towers, transmission links, etc.) so wireless data network design and deployment were straightforward. Dar es Salaam is very flat, and because the cellular partner operated an analog network, towers were very tall. A sister company in the UK, Tele2, had commenced operations with Breezecom (now Alvarion) equipment in 3.8/3.9 GHz, so we followed their lead.

By late 2000, we had established coverage in several cities, using fractional E1 transmission circuits for backhaul. In most cases the small size of the cities connected justified the use of a single omnidirectional PMP base station; only in the commercial capital, Dar es Salaam, were 3-sector base stations installed. Bandwidth limits were configured directly on the customer radio; clients were normally issued a single public IP address. Leaf routers at each base station sent traffic to static IP addresses at client locations, and

prevented broadcast traffic from suffocating the network. Market pressures kept prices down to about $100/month for 64 kbps, but at that time (mid/late 2000) ISPs could operate with impressive, very profitable, contention ratios. Hungry applications such as peer-peer file sharing, voice, and ERPs simply did not exist in East Africa. With grossly high PSTN international charges, organizations rapidly shifted from fax to email traffic, even though their wireless equipment purchase costs ranged from $2000-3000.

Technical capabilities were developed in-house, requiring staff training overseas in subjects such as SNMP and UNIX. Beyond enhancing the company skills set, these training opportunities generated staff loyalty. We had to compete in a very limited IT labor market with international gold mining companies, the UN, and other international agencies.

To insure quality at customer sites, a top local radio and telecoms contractor executed installations, tightly tracking progress with job cards. High temperatures, harsh equatorial sunlight, drenching rain, and lightning were among the environmental insults tossed at outside plant components; RF cabling integrity was vital.

Customers often lacked competent IT staff, burdening our employees with the task of configuring many species of network hardware and topology.

Infrastructure and regulatory obstacles often impeded operations. The cellular company tightly controlled towers, so that if there was a technical issue at a base station hours or days could pass before we gained access. Despite backup generators and UPS systems at every site, electrical power was always problematic. For the cellular company, electrical mains supplies at base stations were less critical. Cellular subscribers simply associated with a different base station; our fixed wireless data subscribers went offline.

On the regulatory side, a major disruption occurred when the telecoms authority decided that our operation was responsible for disrupting C-band satellite operations for the entire country and ordered us to shut down our network.

Despite hard data demonstrating that we were not at fault, the regulator conducted a highly publicized seizure of our equipment. Of course the interference persisted, and later was determined to emanate from a Russian radar ship, involved in tracking space activities. We quietly negotiated with the regulator, and ultimately were rewarded with 2 x 42 MHz of proprietary spectrum in the 3.4/3.5 GHz bands. Customers were switched over to dialup in the month or so it took to reconfigure base stations and install new CPE.

Ultimately the network grew to about 100 nodes providing good, although not great, connectivity to 7 cities over 3000+km of transmission links. Only the

merger with the cellular operator made this network feasible—the scale of the Internet/data business alone would not have justified building a data network of these dimensions and making the investments needed for proprietary frequencies. Unfortunately, the cellular operator took the decision to close the Internet business in mid-2002.

Nairobi

In early 2003 I was approached by a Kenyan company, AccessKenya, with strong UK business and technical backup to design and deploy a wireless network in Nairobi and environs. Benefiting from superb networking and business professionals, improved wireless hardware, progress in internetworking, and bigger market we designed a high availability network in line with regulatory constraints.

Two regulatory factors drove our network design. At the time in Kenya, Internet services were licensed separately from public data network operators, and a single company could not hold both licenses. Carrying traffic of multiple, competing ISPs or corporate users, the network had to operate with total neutrality. Also, "proprietary" frequencies, namely 3.4/3.5 GHz, were not exclusively licensed to a single provider, and we were concerned about interference and the technical ability/political will of the regulator to enforce. Also, spectrum in 3.4/3.5 GHz was expensive, costing about USD1000 per MHz per year per base station. Restated, a base station using 2 x 12 MHz attracted license fees of over $10,000 year. Since Nairobi is a hilly place with lots of tall trees and valleys, wireless broadband networks demanded many base stations. The licensing overheads simply were not sensible. In contrast, 5.7/5.8 GHz frequencies were subject only to an annual fee, about USD 120, per deployed radio.

To meet the first regulatory requirement we chose to provide services using point-point VPN tunnels, not via a network of static IP routes. An ISP would deliver a public IP address to our network at their NOC. Our network conducted a public-private IP conversion, and traffic transited our network in private IP space. At the customer site, a private-public IP conversion delivered the globally routable address (or range) to the customer network.

Security and encryption added to network neutrality, and flexibility, as unique sales properties of our network. Bandwidth was limited at the VPN tunnel level. Based on the operating experience of our sister UK company, VirtualIT, we selected Netscreen (now subsumed under Juniper Networks) as the vendor for VPN firewall routers.

Our criteria for wireless broadband equipment eliminated big pipes and feature-rich, high performance gear. Form factor, reliability, and ease of installation and management were more important than throughput. All inter-

national Internet connections to Kenya in 2003, and at this writing, are carried by satellite. With costs 100X greater than global fiber, satellite connectivity put a financial ceiling on the amount of bandwidth purchased by end-users. We judged that the bulk of our user population required capacity on the order of 128 to 256 kbps. We selected Motorola's recently introduced Canopy platform in line with our business and network model.

Broadband Access, Ltd., went live in July 2003, launching the "Blue" network. We started small, with a single base station. We wanted demand to drive our network expansion, rather than relying on a strategy of building big pipes and hoping we could fill them.

Canopy, and third-party enhancements such as omnidirectional base stations, permitted us to grow our network as traffic grew, softening initial capital expenditures. We knew the tradeoff was that as the network expanded, we would have to sectorize traffic and realign client radios. The gentle learning curve of a small network paid big dividends later. Technical staff became comfortable with customer support issues in a simple network environment, rather than have to deal with them on top of a complex RF and logical framework. Technical staff attended two-day Motorola training sessions.

A typical PMP design, with base stations linked to a central facility via a Canopy high-speed microwave backbone, the network was deployed on building rooftops, not antenna towers. All leases stipulated 24x7 access for staff, mains power and, critically, protected the exclusivity of our radio frequencies. We did not want to restrict landlords from offering roof space to competitors, rather to simply guarantee that our own services would not be interrupted.

Rooftop deployments provided many advantages. Unlimited physical access, unconstrained by night or rain, helped meet the goal of 99.5% network availability. Big buildings also housed many big clients, and it was possible to connect them directly into our core microwave network. Rooftop sites did have the downside of more human traffic—workers maintaining equipment (a/c) or patching leaks would occasionally damage cabling. As a result all base stations were set up with two sets of cabling for all network elements, a primary and a spare.

Site surveys confirmed radio path availability and client requirements. Survey staff logged GPS positions for each client, and carried a laser rangefinder to determine height of obstacles. Following receipt of payment for hardware, contractors under the supervision of a technical staffer performed installations. Canopy has the advantage that the CPE and base station elements are light, so that most installations do not need extensive civil works or guying. Cabling Canopy units was also simple, with outdoor UTP connecting radios directly to customer networks. Proper planning enabled completion of

many installations in less than an hour, and contractor crews did not need any advanced training or tools.

As we compiled hundreds of customer GPS positions we began to work closely with a local survey company to overlay these sites on topographical maps. These became a key planning tool for base station placement.

Note that the point-point VPN tunnel architecture, with its separate physical and logical layers, required clients to purchase both wireless broadband and VPN hardware. In order to tightly control quality, we categorically refused to permit clients to supply their own hardware—they had to buy from us in order to have service and hardware guarantees. Every client had the same hardware package. Typical installations cost on the order of USD 2500, but that compares to the $500-600 monthly charges for 64 to 128 kbps of bandwidth. A benefit of the VPN tunnel approach was that we could prevent a client's traffic from passing over the logical network (i.e. if their network was hit by a worm or if they didn't pay a bill) while the radio layer remained intact and manageable.

As it grew from one base station to ten, and service was expanded to Mombasa, the network RF design evolved and wherever possible network elements (routers) were configured with fallover or hot swap redundancy. Major investments in inverters and dual conversion UPS equipment at each base station were required to keep the network stable in the face of an erratic power grid. After a number of customer issues (dropped VPN connections) were ascribed to power blackouts, we simply included a small UPS as part of the equipment package.

Adding a portable spectrum analyzer to our initial capital investment was costly, but hugely justified as we operated the network. Tracing rogue operators, confirming the operating characteristics of equipment, and verifying RF coverage enhanced our performance.

Fanatical attention to monitoring permitted us to uptweak network performance, and gather valuable historical data. Graphed via MRTG or Cacti (as described in chapter six), parameters such as jitter, RSSI, and traffic warned of rogue operators, potential deterioration of cable/connectors, and presence of worms in client networks. It was not uncommon for clients to claim that service to their site had been interrupted for hours/days and demand a credit. Historical monitoring verified or invalidated these claims.

The Blue network combined a number of lessons from Tanzania with improved RF and networking technologies.

Lessons learned

For the next few years satellite circuits will provide all international Internet connectivity in East Africa. Several groups have floated proposals for submarine fiber connectivity, which will energize telecommunications when it happens. Compared to regions with fiber connectivity, bandwidth costs in East Africa will remain very high.

Wireless broadband networks for delivery of Internet services therefore do not need to focus on throughput. Instead, emphasis should be placed on reliability, redundancy, and flexibility.

Reliability for our wireless networks was our key selling point. On the network side this translated into sizable investments in infrastructure substitution, such as backup power, and attention to details such as crimping and cabling. The most ordinary reasons for a single customer to lose connectivity were cabling or crimping issues. Radio failures were essentially unheard of. A key competitive advantage of our customer installation process is that we pushed contractors to adhere to tight specifications. It was common for well-managed customer sites to remain connected for hundreds of days with zero unscheduled downtime. We controlled as much of our infrastructure as possible (i.e building rooftops).

As attractive as potential alliances with cellular providers seem, in our experience they raise more problems than they solve. In East Africa, Internet businesses generate a fraction of the revenue of mobile telephony, and so are marginal to the cellular companies. Trying to run a network on top of infrastructure that doesn't belong to you and is, from the point of view of the cellular provider, a goodwill gesture, will make it impossible to meet service commitments.

Implementing fully redundant networks, with fail-over or hotswap capability is an expensive proposition in Africa. Nonetheless the core routers and VPN hardware at our central point of presence were fully redundant, configured for seamless fail-over, and routinely tested. For base stations we took the decision not to install dual routers, but kept spare routers in stock. We judged that the 2-3 hours of downtime in the worst case (failure at 1AM Sunday morning in the rain) would be acceptable to clients. Similarly weekend staff members had access to an emergency cupboard containing spare customer premises equipment, such as radios and power supplies.

Flexibility was engineered into both the logical and RF designs of the network. The point-to-point VPN tunnel architecture rolled out in Nairobi was extraordinarily flexible in service of client or network needs. Client connections could be set to burst during off-peak hours to enable offsite backup, as a single example. We could also sell multiple links to separate destinations,

increasing the return on our network investments while opening up new services (such remote monitoring of CCTV cameras) to clients.

On the RF side we had enough spectrum to plan for expansion, as well as cook up an alternative radio network design in case of interference. With the growing number of base stations, probably 80% of our customer sites had two possible base station radios in sight so that if a base station were destroyed we could restore service rapidly.

Separating the logical and RF layers of the Blue network introduced an additional level of complexity and cost. Consider the long-term reality that radio technologies will advance more rapidly than internetworking techniques. Separating the networks, in theory, gives us the flexibility to replace the existing RF network without upsetting the logical network. Or we may install a different radio network in line with evolving technologies (Wimax) or client needs, while maintaining the logical network.

Finally, one must surrender to the obvious point that the exquisite networks we deployed would be utterly useless without unrelenting commitment to customer service. That is, after all, what we got paid for.

More information

- Broadband Access, Ltd. *http://www.blue.co.ke/*
- AccessKenya, Ltd. *http://www.accesskenya.com/*
- VirtualIT *http://www.virtualit.biz/*

—Adam Messer, Ph.D.

Appendix A: Resources

We recommend these resources for learning more about the various aspects of wireless networking. For more links and resources, see our website at *http://wndw.net/.*

Antennas and antenna design

- Cushcraft technical papers on antenna design and radio propagation, *http://www.cushcraft.com/comm/support/technical-papers.htm*
- Free antenna designs, *http://www.freeantennas.com/*
- Hyperlink Tech, *http://hyperlinktech.com/*
- Pasadena Networks LLC, *http://www.wlanparts.com/*
- SuperPass, *http://www.superpass.com/*
- Unofficial NEC-2 code archives, *http://www.si-list.org/swindex2.html*
- Unofficial NEC-2 radio modeling tool home page, *http://www.nittany-scientific.com/nec/*
- USB WiFi dish designs, *http://www.usbwifi.orcon.net.nz/*

Network troubleshooting tools

- Cacti network monitoring package, *http://www.cacti.net/*
- DSL Reports bandwidth speed tests, *http://www.dslreports.com/stest*
- Ethereal network protocol analyzer, *http://www.ethereal.com/*
- Iperf network performance testing tool, *http://dast.nlanr.net/Projects/Iperf/*
- iptraf network diagnostic tool, *http://iptraf.seul.org/*
- MRTG network monitoring and graphing tool, *http://people.ee.ethz.ch/~oetiker/webtools/mrtg/*
- My TraceRoute network diagnostic tool, *http://www.bitwizard.nl/mtr/*
- Nagios network monitoring and event notification tool, *http://www.nagios.org/*
- Ntop network monitoring tool, *http://www.ntop.org/*

- RRDtool round robin database graphing utility, *http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/*
- SmokePing network latency and packet loss monitor, *http://people.ee.ethz.ch/~oetiker/webtools/smokeping/*
- SoftPerfect network analysis tools, *http://www.softperfect.com/*
- Squid transparent http proxy HOWTO, *http://en.tldp.org/HOWTO/mini/TransparentProxy-2.html*
- ttcp network performance testing tool, *http://ftp.arl.mil/ftp/pub/ttcp/*

Security

- AntiProxy http proxy circumvention tools and information, *http://www.antiproxy.com/*
- Anti-spyware tools, *http://www.spychecker.com/*
- Driftnet network monitoring utility, *http://www.ex-parrot.com/~chris/driftnet/*
- Etherpeg network monitoring utility, *http://www.etherpeg.org/*
- Introduction to OpenVPN, *http://www.linuxjournal.com/article/7949*
- Lavasoft Ad-Aware spyware removal tool, *http://www.lavasoft.de/*
- OpenSSH secure shell and tunneling tool, *http://openssh.org/*
- OpenVPN encrypted tunnel setup guide, *http://openvpn.net/howto.html*
- Privoxy filtering web proxy, *http://www.privoxy.org/*
- PuTTY SSH client for Windows, *http://www.putty.nl/*
- Sawmill log analyzer, *http://www.sawmill.net/*
- Security of the WEP algorithm, *http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html*
- Spyware prevention for Windows XP (German), *http://www.xp-antispy.de/*
- Stunnel Universal SSL Wrapper, *http://www.stunnel.org/*
- TOR onion router, *http://tor.eff.org/*
- Weaknesses in the Key Scheduling Algorithm of RC4, *http://www.crypto.com/papers/others/rc4_ksaproc.ps*
- Windows SCP client, *http://winscp.net/*
- Your 802.11 Wireless Network has No Clothes, *http://www.cs.umd.edu/~waa/wireless.pdf*
- ZoneAlarm personal firewall for Windows, *http://www.zonelabs.com/*

Bandwidth optimization

- Cache heirarchies with Squid, *http://squid-docs.sourceforge.net/latest/html/c2075.html*
- dnsmasq caching DNS and DHCP server, *http://thekelleys.org.uk/dnsmasq/doc.html*
- Enhancing International World Wide Web Access in Mozambique Through the Use of Mirroring and Caching Proxies, *http://www.isoc.org/inet97/ans97/cloet.htm*
- Fluff file distribution utility, *http://www.bristol.ac.uk/fluff/*
- Microsoft Internet Security and Acceleration Server, *http://www.microsoft.com/isaserver/*
- Microsoft ISA Server Firewall and Cache resource site, *http://www.isaserver.org/*
- Pittsburgh Supercomputing Center's guide to Enabling High Performance Data Transfers, *http://www.psc.edu/networking/perf_tune.html*
- RFC 3135: Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations, *http://www.ietf.org/rfc/rfc3135*
- Squid web proxy cache, *http://squid-cache.org/*

Mesh networking

- Champaign-Urbana Community Wireless Network software, *http://cuwireless.net/download*
- Freifunk OLSR mesh firmware for the Linksys WRT54G, *http://www.freifunk.net/wiki/FreifunkFirmware*
- MIT Roofnet Project, *http://pdos.csail.mit.edu/roofnet/doku.php*
- OLSR mesh networking daemon, *http://www.olsr.org/*
- Real-time OLSR topology viewer, *http://meshcube.org/nylon/utils/olsr-topology-view.pl*

Wireless operating systems and drivers

- HostAP wireless driver for the Prism 2.5 chipset, *http://hostap.epitest.fi/*
- m0n0wall wireless router OS, *http://m0n0.ch/wall/*
- MadWiFi wireless driver for the Atheros chipset, *http://madwifi.org/*
- Metrix Pebble wireless router OS, *http://metrix.net/metrix/howto/metrix-pebble.html*

- OpenWRT wireless router OS for Linksys access points, *http://openwrt.org/*
- Pebble Linux, *http://nycwireless.net/pebble/*

Wireless tools

- Chillispot captive portal, *http://www.chillispot.org/*
- Interactive Wireless Network Design Analysis Utilities, *http://www.qsl.net/n9zia/wireless/page09.html*
- KisMAC wireless monitor for Mac OS X, *http://kismac.binaervarianz.de/*
- Kismet wireless network monitoring tool, *http://www.kismetwireless.net/*
- MacStumbler wireless network detection tool for Mac OS X, *http://www.macstumbler.com/*
- NetStumbler wireless network detection tool for Windows and Pocket PC, *http://www.netstumbler.com/*
- NoCatSplash captive portal, *http://nocat.net/download/NoCatSplash/*
- PHPMyPrePaid prepaid ticketing system, *http://sourceforge.net/projects/phpmyprepaid/*
- RadioMobile radio performance modeling tool, *http://www.cplus.org/rmw/*
- Terabeam wireless link calculation tools, *http://www.terabeam.com/support/calculations/index.php*
- Wellenreiter wireless network detection tool for Linux, *http://www.wellenreiter.net/*
- WiFiDog captive portal, *http://www.wifidog.org/*
- Wireless Network Link Analysis tool by GBPRR, *http://my.athenet.net/~multiplx/cgi-bin/wireless.main.cgi*

General wireless related information

- DefCon long distance WiFi shootout, *http://www.wifi-shootout.com/*
- Homebrew wireless hardware designs, *http://www.w1ghz.org/*
- Linksys wireless access point information, *http://linksysinfo.org/*
- Linksys WRT54G resource guide, http://seattlewireless.net/index.cgi/LinksysWrt54g
- NoCat community wireless group, *http://nocat.net/*
- POE guide by NYCWireless, *http://nycwireless.net/poe/*
- Ronja optical data link hardware, *http://ronja.twibright.com/*

- SeattleWireless community wireless group, *http://seattlewireless.net/*
- SeattleWireless Hardware comparison page, *http://www.seattlewireless.net/HardwareComparison*
- Stephen Foskett's Power Over Ethernet (PoE) Calculator, *http://www.gweep.net/~sfoskett/tech/poecalc.html*

Networking hardware vendors

- Alvarion wireless networking equipment, *http://www.alvarion.com/*
- Cisco wireless networking equipment, *http://www.cisco.com/*
- Metrix outdoor wireless networking kits, *http://metrix.net/*
- Mikrotik wireless network equipment, *http://www.mikrotik.com/routers.php#linx1part0*
- PowerNOC outdoor wireless networking equipment, *http://powernoc.us/outdoor_bridge.html*
- RAD Data Communications networking hardware, *http://www.rad.com/*
- Redline Communications WiMax wireless networking equipment, *http://www.redlinecommunications.com/*
- Trango wireless networking hardware, *http://www.trangobroadband.com/*
- WaveRider wireless hardware, *http://www.waverider.com/*

Networking services

- Access Kenya ISP, *http://www.accesskenya.com/*
- Broadband Access Ltd. wireless broadband carrier, *http://www.blue.co.ke/*
- Virtual IT outsourcing, http://www.virtualit.biz/
- wire.less.dk consultancy and services, *http://wire.less.dk/*

Training and education

- Association for Progressive Communications wireless connectivity projects, *http://www.apc.org/wireless/*
- International Network for the Availability of Scientific Publications, *http://www.inasp.info/*
- Makere University, Uganda, *http://www.makerere.ac.ug/*
- Radio Communications Unit of the Abdus Salam International Center for Theoretical Physics, *http://wireless.ictp.trieste.it/*
- World Summits on Free Information Infrastructures, *http://www.wsfii.org/*

Miscellaneous links

- Cygwin Linux-like environment for Windows, *http://www.cygwin.com/*
- Graphvis graph visualization tool, *http://www.graphviz.org/*
- ICTP bandwidth simulator, *http://wireless.ictp.trieste.it/simulator/*
- ImageMagick image manipulation tools and libraries, *http://www.imagemagick.org/*
- NodeDB war driving map database, *http://www.nodedb.com/*
- Open Relay DataBase, *http://www.ordb.org/*
- Partition Image disk utility for Linux, *http://www.partimage.org/*
- RFC 1918: Address Allocation for Private Internets, *http://www.ietf.org/rfc/rfc1918*
- Spectropolis NYC art project, *http://www.spectropolis.info/*
- Ubuntu Linux, *http://www.ubuntu.com/*
- wget web utility for Windows, *http://xoomer.virgilio.it/hherold/*
- WiFiMaps war driving map database, *http://www.wifimaps.com/*

Books

- *802.11 Networks: The Definitive Guide, 2nd Edition*. Matthew Gast, O'Reilly Media. ISBN #0-596-10052-3
- *802.11 Wireless Network Site Surveying and Installation.* Bruce Alexander, Cisco Press. ISBN #1-587-05164-8
- The *ARRL Antenna Book, 20th Edition.* R. Dean Straw (Editor), American Radio Relay League*.* ISBN #0-87259-904-3
- The *ARRL UHF/Microwave Experimenter's Manual*. American Radio Relay League*.* ISBN #0-87259-312-6
- *Building Wireless Community Networks, 2nd Edition.* Rob Flickenger, O'Reilly Media. ISBN #0-596-00502-4
- *Deploying License-Free Wireless Wide-Area Networks.* Jack Unger, Cisco Press. ISBN #1-587-05069-2
- TCP/IP Illustrated, Volume 1. W. Richard Stevens, Addison-Wesley. ISBN #0-201-63346-9
- *Wireless Hacks, 2nd Edition*. Rob Flickenger and Roger Weeks, O'Reilly Media. ISBN #0-596-10144-9

Appendix B: Channel Allocations

The following tables list the channel numbers and center frequencies used for 802.11a and 802.11b/g. Note that while all of these frequencies are in the unlicensed ISM and U-NII bands, not all channels are available in all countries. Many regions impose restrictions on output power and indoor / outdoor use on some channels. These regulations are rapidly changing, so always check your local regulations before transmitting.

Note that these tables show the center frequency for each channel. Channels are 22MHz wide in 802.11b/g, and 20MHz wide in 802.11a.

802.11b / g			
Channel #	Center Frequency (GHz)	Channel #	Center Frequency (GHz)
1	2.412	8	2.447
2	2.417	9	2.452
3	2.422	10	2.457
4	2.427	11	2.462
5	2.432	12	2.467
6	2.437	13	2.472
7	2.442	14	2.484

802.11a	
Channel Number	Center Frequency (GHz)
34	5.170
36	5.180
38	5.190
40	5.200
42	5.210
44	5.220
46	5.230
48	5.240
52	5.260
56	5.280
60	5.300
64	5.320
149	5.745
153	5.765
157	5.785
161	5.805