CHAPTER 15

802.11 Network Deployment

Deploying a wireless LAN is a considerable undertaking. Significant planning is required before you can even touch the hardware. Deploying a wireless network is not simply a matter of identifying user locations and connecting them to the backbone. Wireless LANs provide mobility through roaming capabilities, but this feature comes with a price. Wireless LANs are much more susceptible to eavesdropping and unauthorized access. Working to mitigate the security problems while offering high levels of service makes large wireless LAN deployments topologically more complex, especially because solving security problems means that a great deal of integration work may be required to get all the different pieces of the solution working in concert.

Wireless networks require far more deployment planning because of the nature of the radio link. Every building has its own personality with respect to radio transmissions, and unexpected interference can pop up nearly everywhere because of microwave ovens, electrical conduits, or severe multipath interference. As a result, each wireless LAN deployment is unique in many respects, and careful planning and a meticulous site survey are required before removing any equipment from the box.

Beyond considerations due to the physical environment, wireless networks often extend an existing wired infrastructure. The wired infrastructure may be quite complex to begin with, especially if it spans several buildings in a campus setting. Wireless networks depend on having a solid, stable, well-designed wired network in place. If the existing network is not stable, chances are the wireless extension is doomed to instability as well.

This chapter is about deployment considerations for wireless LANs, written from a technical perspective. How do the features of wireless LANs influence network topology? Besides the 802.11 equipment, what do you need to deploy a network? How should the logical network be constructed for maximum mobility? What do you need to look for in a site survey to make a deployment successful?

The Topology Archetype

Figure 15-1 shows how many wireless LAN deployments evolve. This figure serves as the road map for this chapter. The guiding principle of Figure 15-1 is that mobility must be limited to the link layer, because network-layer mobility is not generally available on IP networks. The other design decisions help augment the access control of the wireless device and lower management overhead by taking advantage of existing services, each of which will be considered in turn.

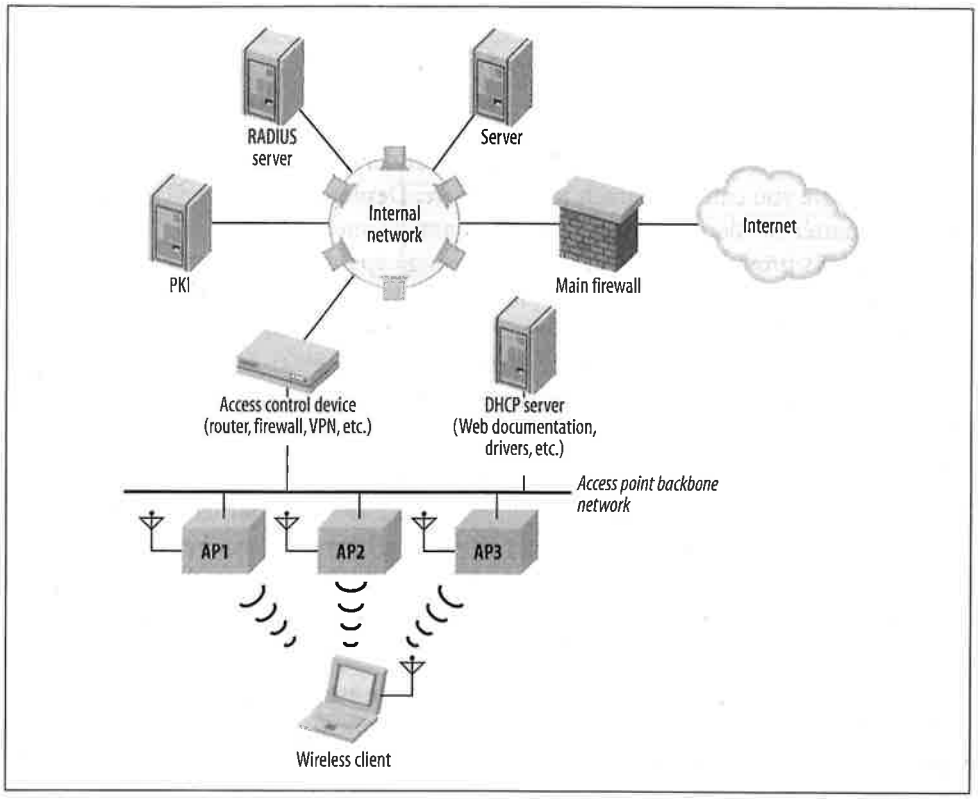

Figure 15-1. Standard wireless LAN deployment topology

Some deployments may look like multiple instances of Figure 15-1. The topology shown in the figure provides seamless mobility between the access points connected to the access point backbone network. In very large deployments, such as a campus-wide deployment across a large number of buildings, it may be desirable to limit the coverage areas in which seamless roaming is provided. One common strategy is to provide seamless mobility within individual buildings, but not provide roaming between buildings. Each building would have a wireless LAN that looked something like Figure 15-1, and all the access point backbone networks would ultimately connect to a campus backbone.

Roaming and Mobility

In Figure 15-1, the network linking all the access points, which I call the *access point backbone*, is a single IP subnet. To allow users to roam between access points, the network should be a single IP subnet, even if it spans multiple locations, because IP does not generally allow for network-layer mobility. To understand this design restriction, it is important first to appreciate the difference between true *mobility* and mere *portability*.*

Portability certainly results in a net productivity gain because users can access information resources wherever it is convenient to do so. At the core, however, portability removes only the physical barriers to connectivity. It is easy to carry a laptop between several locations, so people do. But portability does not change the ritual of connecting to networks at each new location. It is still necessary to physically connect to the network and reestablish network connections, and network connections cannot be used while the device is being moved.

Mobility, on the other hand, is a far more powerful concept: it removes further barriers, most of which are based on the logical network architecture. Network connections stay active even while the device is in motion. This is critical for tasks requiring persistent, long-lived connections, which may be found in database applications. Support personnel frequently access a tracking database that logs questions, problems, and resolutions. The same argument can be made for a number of tracking applications in a health care setting. Accessing the database through a wireless network can boost productivity because it allows people to add small amounts of information from different locations without needing to reconnect to the database each time. Inventory applications are another example and one of the reasons why retail and logistics are two of the markets that have been quicker to adopt 802.11. When taking inventory, it makes far more sense to count boxes or products where they sit and relay data over a wireless network than to record data on paper and collate the data at the end of the process.

Traditional wired Ethernet connections provide portability. I can take my laptop computer anywhere on the campus at work and plug in. (If I'm willing to tolerate slow speeds, I can even make a phone call and access my corporate network from anywhere in the world.) Each time I access the network, though, I'm starting from scratch. I have to reestablish connections, even if I only moved a few feet. What I'd really like is to walk into the conference room and connect to the corporate network without doing anything.

* The exception to this general rule is, of course, a network in which Mobile IP is deployed. I am enthusiastic about Mobile IP, especially on wireless networks, but it is far from ubiquitous as I write this book. Most network engineers are, therefore, designing networks without the benefit of network-layer mobility.

And therein lies the rub. 802.11 is implemented at the link layer and provides link-layer mobility. IP affords the network designer no such luxury. 802.11 hosts can move within the last network freely, but IP, as it is currently deployed, provides no way to move across subnet boundaries. To the IP-based hosts of the outside world, the VPN/access control boxes of Figure 15-1 are the last-hop routers. To get to an 802.11 wireless station with an IP address on the wireless network, simply go through the IP router to that network. It doesn't matter whether a wireless station is connected to the first or third access point because it is reachable through the last-hop router. As far as the outside world can tell, the wireless station might as well be a workstation connected to an Ethernet.

A second requirement for mobility is that the IP address does not change when connecting to any of the access points. New IP addresses interrupt open connections. If a wireless station connects to the first access point, it must keep the same address when it connects to the third access point.

A corollary to the second requirement is that all the wireless stations must be on the same IP subnet. As long as a station stays on the same IP subnet, it does not need to reinitialize its networking stack and can keep its TCP connections open. If it leaves the subnet, though, it needs to get a IP new address and reestablish any open connections. The purpose of the design in Figure 15-1 is to assign a single IP subnet to the wireless stations and allow them to move freely between access points. Multiple subnets are not forbidden, but if you have different IP subnets, seamless mobility between subnets is not possible.

The "single IP subnet backbone" restriction of the design in Figure 15-1 is a reflection on the technology deployed within most organizations. Mobile IP was standardized in late 1996 in RFC 2002, but it has yet to see widespread deployment. (See the sidebar for a description of how Mobile IP allows stations to change IP addresses without interrupting connections.) Until Mobile IP can be deployed, network designers must live within the limitations of IP and design networks based on fixed locations for IP addresses. In Figure 15-1, the backbone network may be physically large, but it is fundamentally constrained by the requirement that all access points connect directly to the backbone router (and each other) at the link layer.

Spanning multiple locations with an 802.11 network

Access points that cooperate in providing mobility need to be connected to each other at layer 2. One method of doing this, shown in Figure 15-2a, builds the wireless infrastructure of Figure 15-1 in parallel to the existing wired infrastructure. Access points are supported by a separate set of switches, cables, and uplinks in the core network. Virtual LANs (VLANs) can be employed to cut down on the required physical infrastructure, as in Figure 15-2b. Rather than acting as a simple layer-2

repeater, the switch in Figure 15-2b can logically divide its ports into multiple layer-2 networks. The access points can be placed on a separate VLAN from the existing wired stations, and the "wireless VLAN" can be given its own IP subnet. Frames leaving the switch for the network core are tagged with the VLAN number to keep them logically distinct and may be sent to different destinations based on the tag. Multiple subnets can be run over the same uplink because the VLAN tag allows frames to be logically separated. Incoming frames for the wired networks are tagged with one VLAN identifier, and frames for the wireless VLAN are tagged with a different VLAN identifier. Frames are sent only to ports on the switch that are part of the same VLAN, so incoming frames tagged with the wireless VLAN are delivered only to the access points.

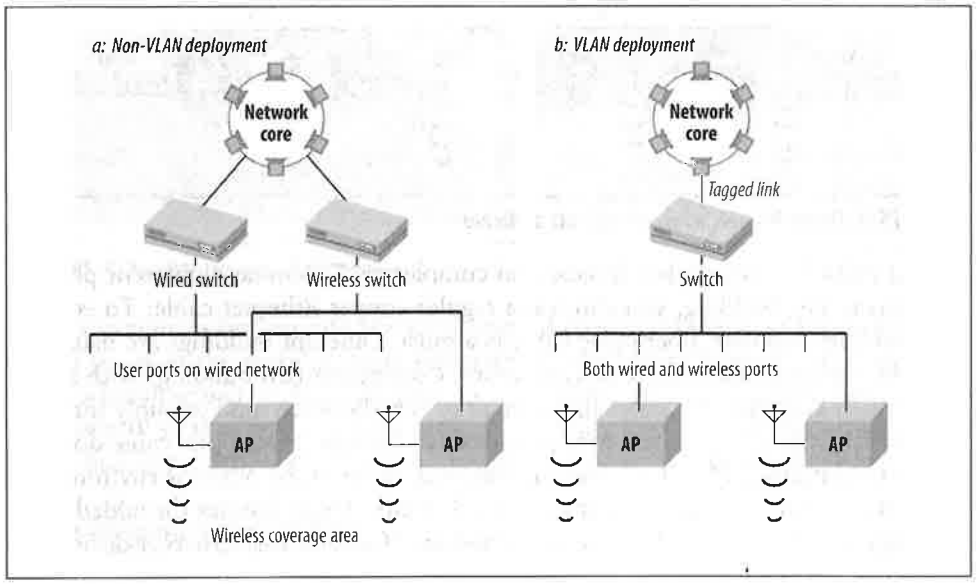

Figure 15-2. Physical topologies for 802.11 network deployment

Even better, VLANs can easily span long distances. VLAN-aware switches can be connected to each other, and the tagged link can be used to join multiple physical locations into a single logical network. In Figure 15-3, two switches are connected by a tagged link, and all four access points are assigned to the same VLAN. The four access points can be put on the same IP subnet and will act as if they are connected to a single hub. The tagged link allows the two switches to be separated, and the distance can depend on the technology. By using fiber-optic links, VLANs can be made to go between buildings, so a single IP subnet can be extended across as many buildings as necessary.

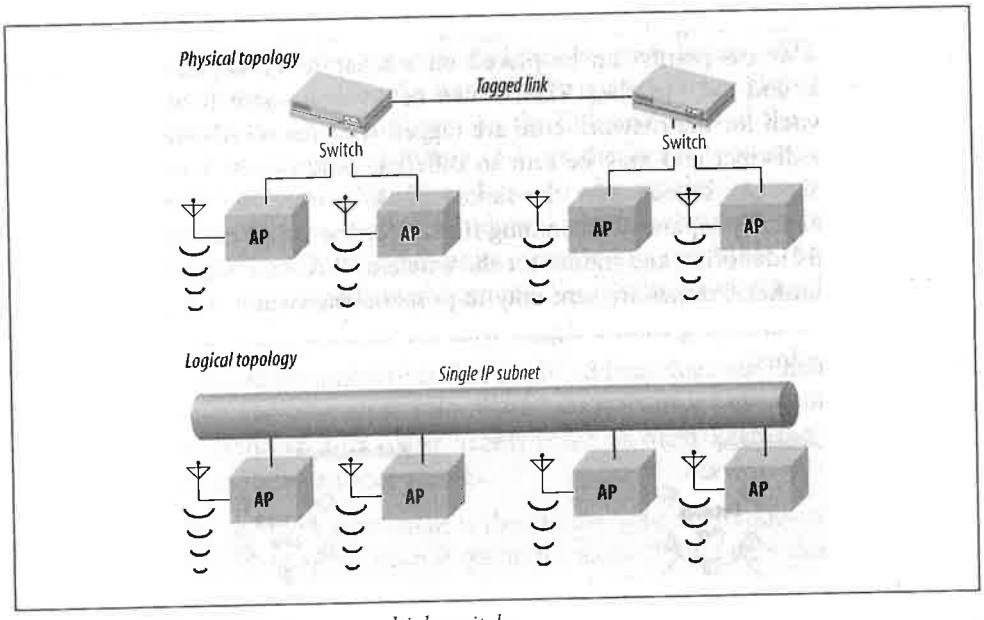

Figure 15-3. Using VLANs to span multiple switches

Tagged links can vary widely in cost and complexity. To connect different physical locations in one building, you can use a regular copper Ethernet cable. To connect two buildings together, fiber-optic cable is a must. Different buildings are usually at different voltage levels relative to each other. Connecting two buildings with a conductor such as copper would enable current to flow between (and possibly through) the two Ethernet switches, resulting in expensive damage. Fiber-optic cable does not conduct electricity and will not pick up electrical noise in the outdoor environment, which is a particular concern during electrical storms. Fiber also has the added benefit of high speeds for long-distance transmissions. If several Fast Ethernet devices are connected to a switch, the uplink will be a bottleneck if it is only a Fast Ethernet interface. For best results on larger networks, uplinks are typically Gigabit Ethernet.

For very large organizations with very large budgets, uplinks do not need to be Ethernet. One company I have worked with uses a metro-area ATM cloud to connect buildings throughout a city at the link layer. With appropriate translations between Ethernet and ATM, such a service can be used as a trunk between switches. Computer trade shows such as Comdex and Interop regularly use metro-area networks to showcase both the metro-area services and the equipment used to access those services.

Limits on mobility

The access point backbone network must be a single IP subnet and a single layer-2 connection throughout an area over which continuous coverage is provided. It may span multiple locations using VLANs. Large campuses may be forced to break up the

access point backbone network into several smaller networks, each of which resembles Figure 15-1.

802.11 allows an ESS to extend across subnet boundaries, as in Figure 15-4a. Users can roam throughout each "island" of connectivity, but network connections will be interrupted when moving between islands. One solution is to teach users one SSID and let them know that mobility is restricted; another alternative is to name each SSID separately. Both solutions have advantages. In the first case, there is only one SSID and no user confusion, but there may be complaints if the coverage areas do not provide mobility in the right ways. In the second case, mobility is always provided within an SSID, but there are several SSIDs and more opportunity for user confusion.

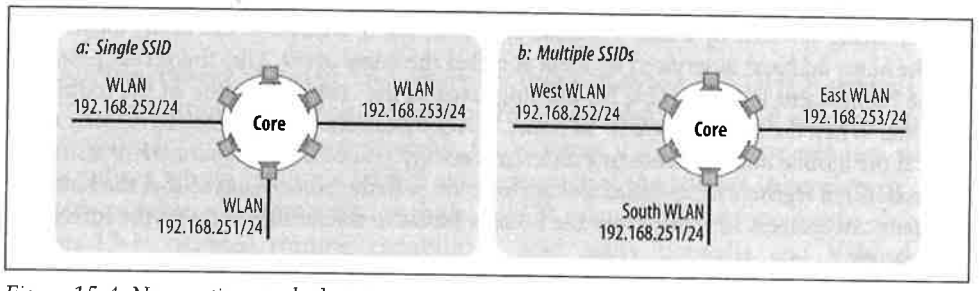

Figure 15-4. Noncontiguous deployments

When a campus is broken into several disjointed coverage areas as in Figure 15-4, be sure to preserve the mobility most important to the users. In most cases, mobility within a building will be the most important, so each building's wireless network can be its own IP subnet. In some environments, mobility may be restricted to groups of several buildings each, so the islands in Figure 15-4 may consist of multiple buildings.

Address assignment through DHCP

Multiple independent data sets that must be synchronized are an accident waiting to happen in any field. With respect to wireless LANs, they present a particular problem for DHCP service. To make life as easy as possible for users, it would be best if stations automatically configured themselves with IP network information. DHCP is the best way to do this. Access points frequently include DHCP servers, but it would be folly to activate the DHCP server on every access point. Multiple independent DHCP lease databases are the network equivalent of a tanker-truck pile-up waiting to happen. To avoid the "multiple independent database" problem, have a single source for IP addressing information. Furthermore, some access points may reclaim addresses if an association lapses, regardless of whether the lease has expired. For these reasons, I recommend using an existing DHCP server or installing a new server specifically to support wireless clients. Depending on the importance of the wireless infrastructure, it may be worth considering a backup server as well.

Mobile IP and Roaming

802.11 performs a sleight-of-hand trick with MAC addresses: stations communicate with a MAC address as if it were fixed in place, just like any other Ethernet station. Instead of being fixed in a set location, however, access points note when the mobile station is nearby and relay frames from the wired network to it over the airwaves. It does not matter which access point the mobile station associates with because the appropriate access point performs the relay function. The station on the wired network can communicate with the mobile station as if it were directly attached to the wire.

Mobile IP performs a similar trick with IP addresses. The outside world uses a single IP address that appears to remain in a fixed location, called the *home location*. Rather than being serviced by a user's system, however, the IP address at the home location (the *home address*) is serviced by what is called the *home agent*. Like the access point, the home agent is responsible for keeping track of the current location of the mobile node. When the mobile node is "at home," packets can simply be delivered directly to it. If the mobile node attaches to a different network (called a *foreign network* or *visited network*), it *registers* its so-called foreign location with the home agent so that the home agent can redirect all traffic from the home address to the mobile node on the foreign network.

Consider an example in which two wireless LANs are built on different IP subnets. On its home subnet, a wireless station can send and receive traffic "normally," since it is on its home network.

When the wireless station moves from its home subnet to the second subnet, it attaches to the network using the normal procedure. It associates with an access point and probably requests an IP address using DHCP. On a wireless station that is unable to use Mobile IP, connections are interrupted at this point because the IP address changes suddenly, invalidating the state of all open TCP connections.

Wireless stations equipped with Mobile IP software, however, can preserve connection state by registering with the home agent. The home agent can accept packets for the mobile station, check its registration tables, and then send the packets to the mobile station at its current location. The mobile station has, in effect, two addresses. It has its home address, and it can continue to use this address for connections that were established using the home address. It may also use the address it has been assigned on the foreign network. No TCP state is invalidated because the mobile station never stopped using its home address.

—continued—

Naturally, this sidebar has omitted a great deal of the detail of the protocol operations. Designing a protocol to allow a station to attach anywhere in the world and use an address from its home network is a significant engineering endeavor. Several security problems are evident, most notably the authentication of protocol operations and the security of the redirected packets from the home network to the mobile station's current location. Maintaining accurate routing information, both the traditional forwarding tables at Internet gateways and the Mobile IP agents, is a major challenge. And, of course, the protocol must work with both IPv4 and IPv6. For a far more detailed treatment of Mobile IP, I highly recommend *Mobile IP: Design Principles and Practices* by Charles Perkins (Prentice Hall).

Within the context of Figure 15-1, there are two places to put a DHCP server. One is on the access point backbone subnet itself. A standalone DHCP server would be responsible for the addresses available for wireless stations on the wireless subnet. Each subnet would require a DHCP server as part of the rollout. Alternatively, most devices capable of routing also include DHCP relay. The security device shown in Figure 15-1 includes routing capabilities, and many firewalls and VPN devices include DHCP relay. With DHCP relay, requests from the wireless network are bridged to the access point backbone by the access point and then further relayed by the access controller to the main corporate DHCP server. If your organization centralizes address assignment with DHCP, take advantage of the established, reliable DHCP service by using DHCP relay. One drawback to DHCP relay is that the relay process requires additional time and not all clients will wait patiently, so DHCP relay may not be an option.

Static addressing is acceptable, of course. The drawback to static addressing is that more addresses are required because all users, active or not, are using an address. To minimize end user configuration, it is worth considering using DHCP to assign fixed addresses to MAC addresses.

As a final point, there may be an interaction between address assignment and security. If VPN solutions are deployed, it is possible to use RFC 1918 (private) address space for the infrastructure. DHCP servers could hand out private addresses that enable nodes to reach the VPN servers, and the VPN servers hand out routable addresses once VPN authentication succeeds.

Use a single DHCP server per access point backbone or DHCP relay at the access point network router to assign addresses to wireless stations. Static addressing or fixed addressing through DHCP is also acceptable.

Security

Informally, data security is defined in terms of three attributes, all of which must be maintained to ensure security:*

Integrity
: Broadly speaking, integrity is compromised when data is modified by unauthorized users. ("Has somebody improperly changed the data?")

Secrecy
: Of the three items, secrecy is perhaps the easiest to understand. We all have secrets and can easily understand the effect of a leak. ("Has the data been improperly disclosed?")

Availability
: Data is only as good as your ability to use it. Denial-of-service attacks are the most common threat to availability. ("Can I read my data when I want to?")

Wireless LAN technology has taken a fair number of knocks for its failures in all three areas. Most notably, though, wireless LANs have two major failings with respect to the informal definition of security. First, secrecy is difficult on a wireless network. Wireless networks do not have firm physical boundaries, and frames are transmitted throughout a general area. Attackers can passively listen for frames and analyze data. To defeat attacks against secrecy, network security engineers must employ cryptographic protocols to ensure the confidentiality of data as it travels across the wireless medium. WEP has been a failure in this respect, but other protocols and standards may be employed instead of or in addition to WEP.

Second, integrity may be compromised by wireless hosts. Quick wireless LAN deployments are often connected directly to a supposedly secure internal network, allowing attackers to bypass the firewall. In many institutions, internal stations are afforded higher levels of access privileges. Physical security may have made some additional privileges rational or inevitable, but wireless stations may not necessarily be trusted hosts run by internal users. Attacks against integrity may frequently be defeated by strong access control.

Vendors often tout WEP as a security solution, but the proven flaws in the design of WEP should give even the most freewheeling security administrators cause for concern. WEP is, in the words of one industry observer, "unsafe at any key length."† Future approaches based on 802.1x and EAP may improve the picture, but current deployments must depend on solutions that are available now. Although products

* My definitions here are not meant to be formal. In this section, I'm trying to take a fundamental approach to security by showing how wireless LAN security fails and how some of the failures can be solved by applying solutions the industry has already developed.

† Or, in the words of one reviewer, "WEP is trash that just gets in the way."

claiming to support 802.1x are currently appearing on the market, they have yet to establish a track record with respect to either security or interoperabilty.

Access control and authentication

Connecting to wireless networks is designed to be easy. In fact, the ease of connection is one of the major advantages to many newer wireless technologies. 802.11 networks announce themselves to anybody willing to listen for the Beacon frames, and access control is limited by the primitive tools supplied by 802.11 itself. To protect networks against the threat of unauthorized access, strong access control should be applied. A helpful rule of thumb is to treat wireless access points like open network drops in the building lobby. 802.11 networks can benefit from access control at two points:

- Before associating with an access point, wireless stations must first authenticate. At present, this process is either nonexistent or based on WEP.
- After association with the access point, the wireless station is attached to the wireless network. However, strong authentication can be applied to any wireless stations to ensure that only authorized users are connecting to protected resources. This form of access control is no different from the access control widely enforced by firewalls today.

At the present time, the initial authentication during the association process is pitifully weak. Current deployments must depend on two methods, one of which was never specified by the standard but is widely used.

One approach is to allow only a specified set of wireless LAN interface MAC addresses to connect to access points. Maintaining the list is its own administrative headache. Distributing the list to access points may be even worse. In a network with access points from multiple vendors, the script may need to massage the list into different file formats to cope with what different products require. Frequently, the list of allowed devices must be distributed by TFTP. Even if the distribution is automated by administrative scripts, TFTP comes with its own security woes. Furthermore, like wired Ethernet cards, 802.11 cards may change the transmitter MAC address, which totally undermines the use of the MAC address as an access control token. Attackers equipped with packet sniffers can easily monitor successful associations to acquire a list of allowed MAC addresses.

A second approach is to allow connections from stations that possess a valid WEP key. Stations that pass the WEP challenge are associated, and stations that fail are not. As noted in Chapter 5, this method is not very strong because WEP is based on RC4, and it is possible to fake a legitimate response to a WEP challenge without any knowledge of the WEP key. In spite of its limitations, WEP makes a useful speed bump for attackers to jump over. Use it, but be aware of its limitations. Or disable it, but be cognizant of the fact that association is unrestricted.

In some products, these methods may be combined. However, both are easily defeated. Maintaining strong security over a wireless LAN requires solutions outside the scope of 802.11, in large part to augment the relatively weak access control supplied by 802.11.

Many networks deploy firewalls to protect against unauthorized access and use of systems by outsiders. In many respects, wireless stations should be considered untrusted until they prove otherwise, simply because of the lack of control over the physical connection. In the network topology shown in Figure 15-1, an access control device is used to protect the internal network from wireless stations. This access control device could be one of several things: a firewall, a VPN termination device, or a custom solution tailored to the requirements of 802.11 networks.

At the time this book was written, many security-conscious organizations opted to use existing firewalls or VPN devices or build systems to meet their own internal requirements. Firewalls are well-known for providing a number of strong authentication mechanisms, and they have a proven ability to integrate with one-time password systems such as RSA's SecurID tokens. New releases of IPSec VPN devices also increasingly have this capability. Initial versions of the IPSec specification allowed authentication only through digital certificates. Certificates work well for site-to-site VPNs, but the idea of rolling out a public-key infrastructure (PKI) to support remote access was frightening for most users. As a result, several new approaches allow for traditional ("legacy") user authentication mechanisms by passing VPN user authentication requests to a RADIUS server. Several mechanisms were in draft form as this book was written: Extended Authentication (XAUTH), Hybrid Mode IKE, and CRACK (Challenge/Response for Authenticated Control Keys).

Several wireless LAN vendors have also stepped up to the plate to offer specialized "wireless access controller" devices, which typically combine packet filtering, authentication, authorization, and accounting services (AAA), and a DHCP server; many devices also include a DNS server and VPN termination. AAA features are typically provided by an interface to an existing corporate infrastructure such as RADIUS, which frequently has already been configured for remote access purposes. Some products may also include dynamic DNS so that a domain name is assigned to a user, but the IP number can be assigned with DHCP.

Several vendors have access controller solutions. Cisco offers an external access control server for the Aironet product line. Lucent's ORiNOCO AS-2000 access server has an integrated RADIUS server. Nokia's P020 Public Access Zone Controller is an integrated network appliance with a RADIUS client and DHCP server, and the companion P030 Mobility Services Manager offers the RADIUS server and billing functions.

Recognize the limitations of WEP. Treat wireless stations as you would treat untrusted external hosts. Isolate wireless LAN segments with firewalls, and use strong authentication for access control. Consider using existing user databases as part of the authentication roll-out.

Confidentiality: WEP, IPSec, or something else?

Confidentiality is the second major goal in wireless LAN deployments. Traffic is left unprotected by default, and this is an inappropriate security posture for most organizations. Users can choose among three options:

- Use WEP.
- Use a proven cryptographic product based on open protocols.
- Use a proprietary protocol.

Option three locks you into a single vendor and leaves you at their mercy for upgrades and bug fixes. Proprietary cryptographic protocols also have a poor track record at ensuring security. In the end, the choice really comes down to whether WEP is good enough. Given the insecurity of WEP, there are two questions to ask:

"Does the data on this network need to stay secret for more than a week?" WEP is not strong encryption by any stretch of the imagination, and you should assume that a sufficiently motivated attacker could easily capture traffic from the wireless network, recover the WEP key, and decrypt any data.

"Do users need to be protected from each other?" In most WEP deployments, keys are distributed to every authorized station. When all users have access to the key, the data is protected from outsiders only. WEP does not protect an authorized user with the key from recovering the data transmitted by another authorized user. If users need to be protected from each other, which is a common requirement in many computing environments, then additional security precautions are required.

Choosing a cryptographic protocol or product is subject to a few basic ground rules, conveniently summarized in the Cryptographic Snake Oil FAQ.* While looking for signs of snake oil is not sure protection, it should filter out the most egregious duds. Cryptography is like a fine wine—it gets better with age. If a protocol or algorithm has withstood extensive public analysis, it is probably better than something just invented.

There are only a few non–snake oil solutions that are worth considering. To provide confidentiality at the network layer, there is only one standard: IPSec. Unfortunately, IPSec is not without its drawbacks. It is a complex system to understand and use, so you must be prepared for a learning curve for network administrators. The complexity of IPSec contributes to a relatively high management overhead, at least at the beginning of deployment. IPSec solutions require the installation of client software on wireless stations to protect outbound traffic, and desktop software management is

* The full document title is "Snake Oil Warning Signs: Encryption Software to Avoid." Get your very own copy from *http://www.interhack.net/people/cmcurtin/snake-oil-faq.html*, among many other places. Bruce Schneier wrote a nice summary of the Snake Oil FAQ in the February 15, 1999, Crypto-Gram newsletter, available from *http://www.counterpane.com/crypto-gram-9902.html*.

always unpleasant at best. Perhaps the most frustrating attribute of IPSec is the difficulty in configuring two different systems to be interoperable. Extensive testing of IPSec interoperability can be a huge burden, one that should not be taken lightly.

An alternative is to allow only applications with strong built-in cryptographic systems. Web-based systems can be secured with the secure socket layer (SSL). Host logins can be secured with SSH. SSH can also be used to secure many types of TCP-based network traffic, though the port-forwarding configuration may be too complex for many users. Some environments may have already deployed a framework such as Kerberos for application layer security, in which case, it can probably be extended to wireless stations without great difficulty.

Consider the wireless network to be transmitting data in the clear if you are not using strong, proven cryptographic solutions such as IPSec or SSH.

Availability through redundancy

Nothing in Figure 15-1 requires single points of failure. Clustered solutions for all the major components exist, so availability is not necessarily compromised by the failure of any of the security components. Clusters are composed of several independent devices that get together and share state information among multiple machines. When any member of the cluster fails, survivors pick up the workload with no interruption. If you elect to use either firewalls or VPNs, consider using a clustered product. Clustering is particularly important for VPN access because you own the infrastructure, and users are far less forgiving of internal problems than flaky Internet connectivity.

One item to watch for in this area is redundancy for DHCP servers. No standard exists for synchronizing the data held by DHCP servers. However, the Network Working Group of the IETF is working towards a standardized DHCP failover protocol, which will increase the reliability of the address allocation service.

Summary and Analysis of Archetypal Topology

Several points about the archetypal topology of Figure 15-1 are important and bear repeating:

1. 802.11 provides for mobility only within an extended service set, and Inter-Access Point Protocols (IAPPs) cooperate only when directly connected at the link layer. Until a standard is developed, proprietary IAPPs are not guaranteed to interoperate; you need to select a single access point vendor for each area of continuous coverage. Each extended service set (ESS) should therefore be a single IP subnet. It is acceptable to use VLANs and other bridging technologies to achieve this goal.

2. Address assignment is best done as dynamic addressing to minimize end user configuration. Only one DHCP server should be responsible for handing out addresses to wireless stations because it is important to prevent accidental readdressing. That server may be placed on the access point backbone. DHCP relay can also be used to take advantage of DHCP servers that are already deployed.
3. Consider the use of WEP. In many cases, it is not recommended because it does not significantly bolster security, but it may complicate roaming, and it is just another configuration item to get wrong. Some vendors charge more for cards that implement the "strong" 128-bit WEP, too.
4. Consider the security policy and goals for your wireless network. WEP is problematic, but it may be better than running open access points. In many environments, though, WEP should be disabled. With large numbers of users, WEP is just another configuration item to get wrong. Deploying WEP may also complicate roaming between access points.
5. Carefully consider the limitations of WEP and deploy additional solutions to enhance:

   Authentication

   WEP does not provide strong user authentication, so treat any wireless stations as you would untrusted external hosts. Your security policy may offer guidance here—how do you protect against threats from existing mobile users, such as telecommuters and road warriors? Isolate wireless LAN segments with firewalls and use strong authentication for access control. Existing user databases such as RADIUS servers can probably be used as part of the access control deployment.

   Confidentiality

   If you do not want to broadcast data to the world, use strong, proven cryptographic solutions to protect data as it traverses the airwaves. IPSec is the best standardized solution, especially since it now provides strong user authentication with existing token-based systems.
6. To maintain availability and increase uptime, use clustered solutions whenever possible and cost-effective.

Project Planning

Site survey work is the heart of installing a wireless LAN. To successfully run a site survey, though, preparation is very important.

Gathering Requirements

Before "breaking cable" on a network expansion, gather end user requirements and information to find out which expectations are important. Use the following checklist

to flesh out the customer requirements; each point is detailed further in a subsequent section:

Throughput considerations
: How much throughput is required? This is partly dependent on the type of device that will be used on the wireless LAN, though if it is a PC-like device with the ability to display large and complex graphics, you will want your wireless LAN to be as fast as possible. In most cases, this will lead to choosing 802.11b-based networks to use the 11-Mbps physical layer. If you like leading-edge technology, you may want to consider 802.11a products, several of which appeared just as this book went to press.

Coverage area
: Where should coverage be provided, and what is the density of users in particular areas?

Mobility
: How much movement between coverage is needed? Does it need to be full mobility, with continuous connections as the wireless station moves around the network? Or can the network simply enable effective portability by facilitating automatic reconfiguration when the mobile station moves between coverage areas?

User population
: How many people will use the wireless network, and what quality of service do they expect? As always, be sure to allow for growth!

Physical network planning
: Will new network cabling be needed to supply the wireless LAN backbone, or can you make do with existing cabling? Are an adequate number of outlets available in the correct locations? Can the access points and antennas be installed in the open or must they be confined to wiring closets or other hidden locations?

Logical network planning
: How many IP addresses will be set aside for wireless users? Is a large enough address block available, or will the existing network need to be renumbered to accommodate the wireless network? If the necessary IP address space is not available, it may mean cutting back on the level of seamless mobility on the wireless LAN.

Application characteristics
: Can address translation be used to save IP address space? Are any applications sensitive to high or variable delays? Do any applications provide time-critical data? If so, consider looking for products that support the point coordination function and contention-free delivery, but be aware that many products do not support the PCF.

Security requirements

Wireless LANs have been subject to a number of security concerns. The two main goals of wireless LAN security planning are ensuring adequate access control and preserving the confidentiality of data as it traverses the wireless network. Security requirements may be dictated by legal requirements or the legal threat of unauthorized data disclosure.

Authentication has long been a weak point of 802.11 networks. The two main options provided by 802.11 are to filter on the MAC addresses of wireless stations allowed to connect or use shared WEP keys for stronger authentication. In practice, MAC address filtering is too cumbersome and error-prone, so the choices are to use WEP authentication or depend on external solutions.

Data confidentiality is provided by encryption services. One option is the WEP standard, though higher-security sites may opt for additional VPN technology on top of the 802.11 layer.

Site environmental considerations

A number of factors can affect radio propagation and signal quality. Building materials, construction, and floor plan all affect how well radio waves can move throughout the building. Interference is a fact of life, but it is more pronounced in some buildings than in others. Temperature and humidity have minor effects. Early site visits can assist in anticipating several factors, and a detailed site survey can spot any real problems before installation begins in earnest.

Purchasing wireless LAN hardware and software

At some point, wireless LAN hardware and software must be purchased. Many vendors exist, and the decision can be based on a number of criteria. (Selecting an access point vendor was discussed in Chapter 14.) Selecting cards may depend on your institution's policies. Some organizations may choose to purchase all cards centrally from a single vendor. Others may choose to select a small set of officially "supported" vendors but allow users to select alternative hardware if they are willing to forego official support from the network staff. At least one wireless network analyzer should be part of the budget. Depending on the size of the wireless LAN and the number of network administrators, you may wish to budget for more than one.

Project management

As with many other projects, drawing up a schedule and budget is a necessary component. This chapter does not provide any guidance on nontechnical factors because they are often organization-specific.

Network performance requirements

Depending on the applications used on the wireless network, different requirements are imposed. One of the most important items, and the one that is least under the control of the network architect, is the characteristics of the application. Most applications

can now be run over TCP/IP, but they may require widely varying throughput, delay, or timing characteristics. More importantly, though, is how an application reacts to network address translation (NAT)—translating its IP addresses with intermediate devices.

Single TCP connections, such as those used by HTTP and SSH, are easily translated with no side effects. Other network protocols, most notably those in the Microsoft Networking family, embed the source IP address in the data portion of the packet and cannot be used in conjunction with address translation without great difficulty.* NAT also causes problems for most videoconferencing applications. At the time this book was written, standardized IPSec also did not work when the IPSec packet was passed through an address translator because IPSec authenticates the source IP address of the packet. Translating the source IP address caused the integrity check to fail.

The remaining three factors are under direct control of the end user. A coverage area must be defined, and a form of mobility between the coverage areas is a likely companion requirement. (Mobility imposes its own requirements on the IP addressing architecture, which was discussed previously.)

Finally, end users will have a target throughput requirement. Any throughput goals must be carefully considered because wireless LANs are a shared medium, but one without an upgrade path similar to dropping in an Ethernet switch.

Table 15-1 shows the number of users that can be served on 11-Mbps 802.11b networks, with different sustained loads per user. Wireless LAN bit rates are low, and the extra management features limit throughput to a relatively low fraction of the available bit rate. As you can see from the table, though, 11-Mbps networks are likely to be practical for office environments, which are mainly email, web browsing, and intermittent file access. It is likely that 20–30 users per access point is a reasonable estimate for capacity planning.

Table 15-1. Network capacity compared to sustained throughput per user

Connection method and speed	Effective number of simultaneous users on 11-Mbps networks (6 Mbps data throughput)
Cellular modem, 9.6 kbps	625
Modem, 50 kbps	120
Single ISDN B channel, 64 kbps	93
Dual ISDN, 128 kbps	46
100 kbps sustained LAN usage	60
150 kbps sustained LAN usage	40

* NAT devices can block logon traffic and the inter-domain controller chatter used by NT-based networks. See articles Q172227 and Q186340 in the Microsoft Knowledge Base.

Table 15-1. Network capacity compared to sustained throughput per user (continued)

Connection method and speed	Effective number of simultaneous users on 11-Mbps networks (6 Mbps data throughput)
200 kbps sustained LAN usage	30
300 kbps sustained LAN usage	20

Realistic throughput expectations for 802.11b networks

In terms of throughput, the performance of 802.11 LANs is similar to shared Ethernet. As more users are added, the available capacity per user is divided up. A practical rule of thumb is that the highest throughput that can be attained using the DCF is 75% of the nominal bit rate. The 75% figure is a theoretical result derived from the protocol itself; it includes overhead such as the preamble, interframe spaces, and framing headers. However, throughput rates as low as 50% may be observed. A target of 65% of the nominal bit rate is commonly observed.

For 2-Mbps networks, this translates to a top speed of 1.5 Mbps, though rates as low as 1.3 Mbps are common. Applying similar percentages to 11-Mbps networks yields a practical throughput range of 6 to 8 Mbps.

For networks under the operation of the PCF, throughput is higher because it uses shorter interframe spaces and more efficient acknowledgments. Implementation of the PCF is not required by the standard, so implementations are quite uncommon.

Security

The security trade-offs were discussed in the previous "Security" section. In many cases, an IPSec-based VPN is the logical choice. IPSec was designed for precisely the environment that wireless LANs typify. Intruders can easily capture traffic and perform extensive offline attacks with stored data. Now that IPSec is evolving to support remote clients connecting to central sites, it can also be used to provide strong authentication without a difficult PKI rollout. Many products now support one of the various standards to allow an IPSec termination device to perform user authentication through RADIUS, which allows administrators to take advantage of existing authentication databases. The new 802.1x standard incorporates RADIUS; unless there's a critical problem in 802.1x (as there was in WEP), future wireless security is likely to be based on the 1x standard.

Coverage and physical installation restrictions

Part of the end user requirement is a desired coverage area, and possibly some physical restrictions to go along with it. Physical restrictions, such as a lack of available electrical power and network connections, can be mundane. Some institutions may also require that access points and antennas are hidden; this may be to maintain the physical security of the network infrastructure, or it may be simply to preserve the aesthetic appeal of the building.

11a, 11b, 11g, and more?

Where do you go past 11 Mbps? That's the question people have been asking with increasing frequency over the past year. Right now, the leading standard for higher wireless data rates is 802.11a; it provides 54 Mbps in the 5-GHz band. 802.11a products are on the market now, though it's really too early to say anything substantial about them. Some vendors are announcing that their access points can be upgraded to 802.11a by purchasing a new card and installing new firmware. Software upgrades may be helpful, but only if the hardware is ready for 802.11a. Some vendors have boasted about the easy software upgrade to the 54-Mbps performance of 802.11a, but the access point in question had only a 10-Mbps Ethernet port. Any access points you consider software upgrades for should have Fast Ethernet ports. 802.11a is somewhat more expensive than 11b, though prices should begin to drop soon.

Because 802.11a uses the same MAC layer as 802.11b, with the OFDM PHY layer discussed in Chapter 12, I expect the installation and administration of products to be essentially the same as it is for 802.11b products. In short, just about everything discussed in this book still applies. I am guessing that the range should be similar; OFDM looks like a superior modulation technique, but at the higher frequency, there should be greater problems with path loss, multipath fading, reflections, etc. One estimation was that the radius of 802.11a access points would be 20–25% shorter. Because the 5GHz band is much larger than the ISM band and isn't already occupied by microwave ovens and other devices, there should be fewer problems with interference.

Another standard waiting in the wings is 802.11g. 11g is a 2.4-GHz standard, like 11b, but it uses the OFDM modulation technique of 11a. It also operates at 54 Mbps. The standard isn't finalized yet, and it's hard to imagine products appearing before the end of 2002. The upgrade path from 11b to 11g might be easier than that from 11b to 11a; because both standards use the same frequency band, you should be able to upgrade your access points without worrying about changing their coverage. (For better or worse, the RF characteristics of your site will be different at 2.4 and 5 GHz, so you may find you need to move or add access points if you migrate to the higher-frequency band.) 802.11g also promises to be less expensive than 11a, though in practice this probably means only that the existence of 11g will drive down the price of 11a products.

Some organizations may want to provide coverage outdoors as well, though this is confined to mild climates. Any equipment placed outdoors should be sturdy enough to work there, which is largely a matter of waterproofing and weather resistance. One solution is to install access points inside and run antennas to outdoor locations, but external antenna cables that are long enough are not always available. Outdoor network extensions can be difficult because most 802.11 equipment is not suited to outdoor use, and even if it was, power and Ethernet connections are not readily available outdoors. The best approach to providing outdoor coverage is to keep the access points inside and use external weatherproof antennas on the roof.

The Building

It is a great help to get blueprints or floor plans and take a tour of the installation site as early as possible in the process. Based on a walk-through with the floor plans, you can note where coverage must be provided, nearby network and power drops, and any relevant environmental factors. Most importantly, you can correct the blueprints based on any changes made to the structure since the blueprints were drawn. Many minor changes will not be reflected on the blueprints.

Different materials have different effects on the radio link. Signal power is most affected by metal, so elevator shafts and air ducts cause significant disruption of communications. Tinted or coated windows frequently cause severe disruption of radio signals. Some buildings may have metal-coated ceilings or significant amounts of metal in the floor. Wood and most glass panes have only small effects, though bullet-proof glass can be quite bad. Brick and concrete have effects somewhere between metal and plain untreated glass. To a large extent, though, the expected drop in signal quality due to building construction is a judgment call that improves with experience.

During a pre-survey walk-though, also note any potential sources of interference. The 2.4-GHz ISM band is unlicensed, so many types of devices using the band can be deployed without central coordination. Newer cordless phones operate in the 2.4-GHz band, as well as Bluetooth-based devices and a number of other unlicensed radio devices. Depending on the quality and amount of shielding, microwave ovens may also emit enough radiation to disrupt 802.11 communications. If you anticipate a large amount of interference, testing tools called *spectrum analyzers* can identify the amount of radiation in the wireless LAN frequency band. If your organization does RF testing, it may be necessary to shield any labs where testing is done to avoid interference with the wireless LAN. As a rule of thumb, keep access points at least 25 feet away from any strong interference sources. End user devices also suffer if they are located too close to sources of interference, but only end user communications are interrupted in that case.

The Network

There are two components to network planning. The first, physical planning, is largely legwork. In addition to the building map, it helps to obtain a physical network map, if one exists. It is much easier to install wireless LAN hardware when no expensive and time-consuming wiring needs to be done. Knowing the location and contents of all the wiring closets is an important first step.

The second component of network planning is the plan for changes to the logical network. How will mobile stations be addressed? How will access points be reconnected to their firewall or router?

Network addressing

802.11 provides mobility between access points as long as both access points are part of the same ESS. Roaming works only when mobile stations can transfer from one

access point to another and keep the same IP address. All access points of the same ESS must therefore be connected to the same IP subnet so that wireless stations can keep addresses as they associate with different access points.

To get the IP space allocated, you will probably need to work with a network administrator. The administrator will want to know how many addresses you need and why. In addition to the planned number of wireless stations, be sure to include an address for each access point and any servers and security devices on the wireless subnet.

After tentatively locating access points on a blueprint or sketch of the area, work with the physical network map to plug the access points into the nearest wiring closet. If the access device is a switch with VLAN capability, the access point can probably be placed on the access point backbone VLAN. If not, it may be necessary to patch the access point back to a switch capable of VLAN connections or replace the access device with a small multi-VLAN switch.

Preliminary Plan

Based on the floor plans, use the map to come up with a preliminary plan. The preliminary plan is based on the coverage area required and the typical coverage radius from an access point. At this point, detailed radio channel use planning is not yet necessary. The main use of the preliminary plan is to come up with trial access point locations to begin signal quality measurements as part of the detailed site survey. Table 15-2 is based on the best-case coverage radius from a typical omnidirectional antenna.

Table 15-2. "Rule-of-thumb" coverage radius for different types of space

Type of space	Maximum coverage radius
Closed office	up to 50–60 feet
Open office (cubicles)	up to 90 feet
Hallways and other large rooms	up to 150 feet
Outdoors	up to 300 feet

The Site Survey

After coming up with the preliminary plan, it is time to move on to the heart of the deployment routine: the site survey. Several options exist for performing a site survey. Vendors may provide site surveys to early adopters who agree to be reference accounts. Value-added resellers may also have the skills to perform detailed site surveys; resellers may sell site survey consulting services or use site surveys as a way of coming up with a wireless LAN deployment bid. Some companies that specialize in technical education also offer classes on performing site surveys.

Refining the preliminary design is the purpose of the site survey. Radio transmission is complicated, and some things must be done by experiment. All sites will require adjustments to the preliminary design as part of the site survey. In many cases, use of site survey tools can help eliminate access points from a network design and result in substantial cost savings. The major goal of a site survey is to discover any unforeseen interference and redesign the network accordingly. In most cases, interference problems can be repaired by relocating an access point or using a different antenna.

The site survey should assess the following:

- The actual coverage of the access points and the optimum location of access points in the final network
- Actual bit rates and error rates in different locations, especially locations with a large number of users
- Whether the number of access points is sufficient—more or fewer may be required, depending on the characteristics of the building with respect to radio waves
- The performance characteristics of customer applications on the wireless LAN

Tools

Site survey work consists mostly of seemingly endless signal quality measurements. Depending on the tool used, the signal quality measurements may be any of the following:

Packet Error Rate (PER)
: The fraction of frames received in error, without regard to retransmissions. A common rule of thumb is that the PER should be less than 8% for acceptable performance.

Received Signal Strength Indication (RSSI)
: A value derived from the underlying mathematics. Higher values correspond to stronger (and presumably better) signals.

Multipath time dispersion
: Some software or instruments may be able to measure the degree to which a signal is spread out in time by path differences. Higher delay spreads make the correlation of the wideband signals more difficult. Devices need to accept either a higher error rate at high delay spreads or fall back to a more conservative coding method. Either way, throughput goes down. The higher the delay spread, the more throughput suffers.

Signal quality measurements can be carried out by a dedicated hardware device or a software program running on a laptop with the card vendor's site survey tool. Several wireless LAN vendors, such as Intel, Proxim, and 3Com, bundle site survey tools

with their access points. Handheld site survey tools designed specifically for 802.11 networks also exist.

Patience and comfortable shoes are among the most important items to bring to a site survey. Measuring signal quality in an area is a painstaking process, requiring many measurements, often taken after making minor changes to the antenna or access point configuration. You will spend a lot of time walking, so wear shoes that you can walk all day in.

Particularly stubborn interference may require the use of a *spectrum analyzer* to locate the source of interference from a non-802.11 network. Devices that can scan a wide frequency band to locate transmissions are not cheap. Expect to pay several thousand dollars, or you can hire a consultant. In either case, a spectrum analyzer is the tool of last resort, necessary for only the most stubborn problems.

Antenna Types

Wireless cards all have built-in antennas, but these antennas are, at best, minimally adequate. If you were planning to cover an office—or an even larger area, such as a campus—you will almost certainly want to use external antennas for your access points. When considering specialized antennas, there are only a few specifications that you need to pay attention to:

Antenna type
: The antenna type determines its radiation pattern—is it omnidirectional, bidirectional, or unidirectional? Omnidirectional antennas are good for covering large areas; bidirectional antennas are particularly good at covering corridors; unidirectional antennas are best at setting up point-to-point links between buildings, or even different sites.

Gain
: The gain of the antenna is the extent to which it enhances the signal in its preferred direction. Antenna gain is measured in dBi, which stands for decibels relative to an isotropic radiator. An isotropic radiator is a theoretical beast that radiates equally in all directions. To put some stakes in the ground: I've never seen a specification for the gain of the built-in antenna on a wireless card, but I would guess that it's negative (i.e., worse than an isotropic radiator). Simple external antennas typically have gains of 3 to 7 dBi. Directional antennas can have gains as high as 24 dBi.*

Half-power beam width
: This is the width of the antenna's radiation pattern, measured in terms of the points at which the antenna's radiation drops to half of its peak value. Understanding the half-power beam width is important to understanding your

* If you want one more stake, the radio telescope at Arecibo has a gain in excess of 80 dBi.

antenna's effective coverage area. For a very high-gain antenna, the half-power beam width may be only a couple of degrees. Once you get outside the half-power beam width, the signal typically drops off fairly quickly, though that depends on the antenna's design. Don't be fooled into thinking that the half-power beam width is irrelevant for an omnidirectional antenna. A typical omnidirectional (vertical) antenna is only omnidirectional in the horizontal plane. As you go above or below the plane on which the antenna is mounted, the signal decreases.

We've discussed antennas entirely in terms of their properties for transmitting, largely because most people find that easier to understand. Fortunately, an antenna's receiving properties are identical to its transmitting properties—an antenna enhances a received signal to the same extent that it enhances the transmitted signal. This result is probably what you would expect, but proving it is beyond the scope of this book.

Now, let's talk about some of the antenna types that are available; Figure 15-5 shows a number of different antenna types:

Vertical

This is a garden variety omnidirectional antenna. Most vendors sell several different types of vertical antenna, differing primarily in their gain; you might see a vertical antenna with a published gain as high as 10 dBi or as low as 3 dBi. How does an omnidirectional antenna generate gain? Remember that a vertical antenna is omnidirectional only in the horizontal plane. In three dimensions, its radiation pattern looks something like a donut. A higher gain means that the donut is squashed. It also means that the antenna is larger and more expensive, though no antennas for 802.11 service are particularly large.

If you want to cover a confined outdoor area—for example, a courtyard between several buildings of a corporate campus—note that the half-power beam width means that a roof-mounted vertical antenna might be less than ideal, particularly if the building is tall. Vertical antennas are good at radiating out horizontally; they're not good at radiating down. In a situation like this, you would be better off mounting the antenna outside a first- or second-story window.

Dipole

A dipole antenna has a figure eight radiation pattern, which means it's ideal for covering a hallway or some other long, thin area. Physically, it won't look much different from a vertical—in fact, some vertical antennas are simply vertically mounted dipoles.

Yagi

A Yagi antenna is a moderately high-gain unidirectional antenna. It looks somewhat like a classic TV antenna. There are a number of parallel metal elements at right angles to a boom. However, you are not likely to see the elements on a Yagi for 802.11 service; the commercially made Yagis that I have seen are all enclosed

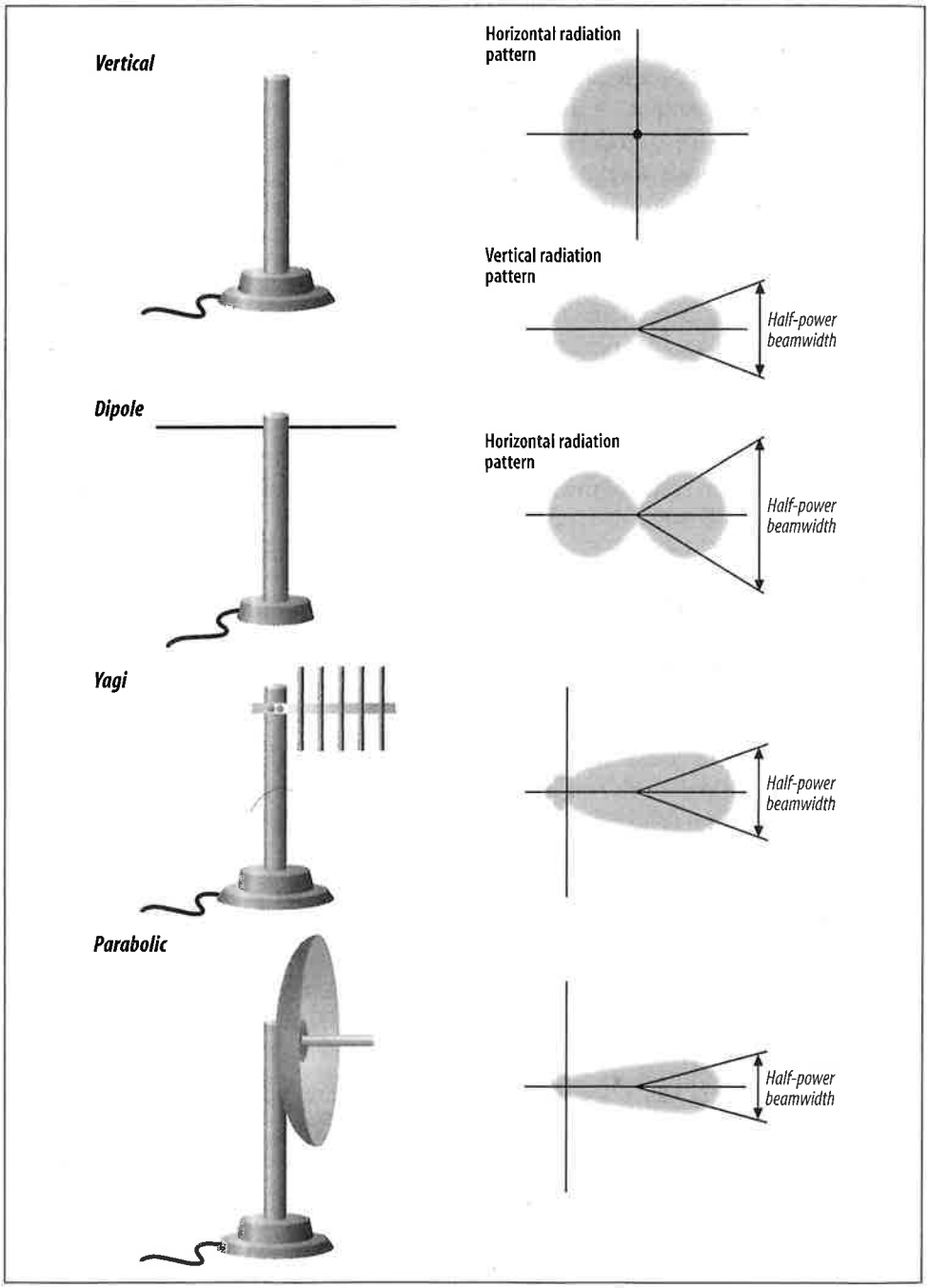

Figure 15-5. Antenna types

in a *radome*, which is a plastic shell that protects the antenna from the elements in outdoor deployments. Yagi antennas for 802.11 service have gains between 12 and 18 dBi; aiming them is not as difficult as aiming a parabolic antenna, though it can be tricky.

Parabolic

This is a very high-gain antenna. Because parabolic antennas have very high gains (up to 24 dBi for commercially made 802.11 antennas), they also have very narrow beam widths. You would probably use a parabolic antenna only for a link between buildings; because of the narrow beam width, they are not very useful for providing services to end users. Vendors publish ranges of up to 20 miles for their parabolic antennas. Presumably, both ends of the link are using a similar antenna. Do not underestimate the difficulty of aiming a parabolic antenna properly—one commercial product has a published beam width of only 6.5 degrees. If you decide to install a parabolic antenna, make sure that you have it mounted firmly. You do not want a bad storm to nudge it a bit and take down your connection.

Some vendors make an issue of the distinction between "mesh" or "grid" parabolas (in which the antenna's reflector looks like a bent barbecue grill) and solid parabolas. Don't sweat it—if the antenna is well-designed, the difference in performance between a mesh and a solid reflector is not worth worrying about. A mesh does have an advantage, though, in areas subject to high winds.

Parabolic and Yagi antennas are useful primarily for links between buildings. The biggest problem is aiming them properly. If the two sites are visible to each other, you can play some tricks with gunsights—though if you can see one site from the other, you probably don't need such a sophisticated antenna system. Otherwise, buy a good compass and a topographical map from the U.S. Geological Survey, and compute the heading from one site to the other. Remember to correct for magnetic north. If you can spend some extra money, you might be able to simplify the setup by installing a high-gain vertical antenna at one site; then you need to aim only one antenna. If the signal is marginal, replace the vertical with a parabolic antenna once you have the first antenna aimed correctly.

High-gain antennas can become a regulatory problem, particularly in Europe (where power limits are lower than in the U.S.). Lucent notes that their high-gain parabolic antenna cannot be used legally on channels 1, 2, 10, and 11 in the U.S., though it can be used on channels 3 through 9. But that limitation probably assumes that you're using a Lucent wireless card, and Lucent's transceiver produces less output power than the Intersil chip set used by most other vendors. If you connect the Lucent parabolic antenna to a Nokia wireless card, you'll be be way beyond the maximum legal effective radiated power.

Cabling

Having put so much effort into thinking about antennas, we have to spend some time thinking about how to connect the antennas to the access points or wireless cards. Most vendors sell two kinds of cable: relatively inexpensive thin cable (typically 0.1 inch in diameter) and "low-loss cable" that's substantially thicker (typically 0.4 inch) and much more expensive. The thin cable is usually available only in lengths of a couple of feet, and that's as it should be: it is very lossy, and more than a few feet can easily eat up your entire signal. It's intended for connecting a wireless card in a laptop to a portable antenna on your desktop, and that's all. To put numbers behind this: one vendor specifies a loss of 2.5 dB for a 2-meter cable. That means that close to half of your signal strength is disappearing in just two meters of cable. One cable vendor, for a cable that would typically be used in this application, specifies a loss of 75 dB per 100 feet at 2.4 GHz. That means that your signal strength will drop by a factor of 2^{25} (roughly 33 million), clearly not something you want to contemplate. I know of one vendor that recommends using RG58 cable with medium-gain antennas. RG58 is better than the really thin cable intended for portable use, but not much better (35 dB per 100 feet); if you use RG58 cable, keep the cable run as short as possible. Better yet, ditch the RG58 and see if you can replace it with LMR-200 (a high-quality equivalent with half the loss).

What does the picture look like when you're using a *real* low-loss cable? Significantly better, but maybe not as better as you would like. A typical cable for this application—used by at least one 802.11 vendor—is Times Microwave LMR-400. LMR-400 is a very high-quality cable, but it still has a loss of 6.8 dB per 100 feet at 2.4 GHz. This means that, in a 100-foot length of cable, over three quarters of your signal is lost. The moral of the story is clear: keep your access points as close as possible to your antennas. Minimize the length of the transmission line. If you want a roof-mounted antenna, perhaps to cover a courtyard where people frequently have lunch, don't stick your access point in a wiring closet in the basement and run a cable to the roof. If possible, put your access point in a weatherproof enclosure on the roof. If that's not possible, at least put the access point in an attic or crawlspace. There is no substitute for keeping the transmission line as short as possible. Also, keep in mind that transmission lines have a strange ability to shrink when they are routed through walls or conduits. I've never understood why, but even if you measure carefully, you're certain to find that your cable is two feet short. More to the point: the straight-line distance from your access point to the antenna may be only 20 feet, but don't be surprised if it takes a 50-foot cable to make the trip. The cable will probably have to go around corners and through conduits and all sorts of other misdirections before it arrives at its destination.

If you decide to use an 802.11a product, which operates at 5-GHz, be aware that cable loss will be an even more significant issue. Losses increase with frequency, and coaxial cable isn't particularly effective at 2.4 GHz, let alone 5 GHz.

Finally, there's the matter of antenna connectors. All wireless vendors sell cables in various length with the proper connectors and adapters. I strongly recommend taking the easy way out and buying cables with the connectors preinstalled. Connector failure is one of the most common causes for outages in radio systems, particularly if you don't have a lot of experience installing RF connectors.

Antenna diversity

One common method of minimizing multipath fading is to have *antenna diversity*. Rather than making the antenna larger, radio systems can use multiple antennas and choose the signal from the antenna with better reception. Using multiple antennas does not require sophisticated mathematical theory or signal-processing techniques.

Several wireless LAN vendors have built multiple antennas into wireless network cards. Some vendors even offer the ability to connect multiple external antennas to network cards intended for access points. Antenna diversity is recommended by the 802.11 standard, but it is not required. For environments with large amounts of interference, antenna diversity is a worthwhile option to consider when selecting vendors.

Bring on the heat

Amplifiers are available for increasing your transmitting power. Transmitting amplifiers often incorporate preamplifiers for receiving, helping to improve your weak signal sensitivity. Is an amplifier in your future? It depends. The basic problem is that, as you cover a larger and larger territory, there are more and more stations that can potentially join your network. However, the number of stations that can be handled by any given access point is fairly limited (see the following section). All in all, more low-power access points provide better service than a smaller number of access points with high-power amplifiers. There may be some applications that are exceptions to the rule (community networks or ISPs in remote areas, for example), but in most situations, high power sounds like a better idea than it really is.

However, if you want to check around and see what's available, SSB Electronics (*www.ssbusa.com/wireless.html*) and HyperLink Technologies (*http://www.hyperlinktech.com/web/amplifiers_2400.html*) sell high-power amplifiers for 802.11b service. However, remember:

- To stay within the legal power limit, both for absolute power and ERP.
- That 802.11 is an unlicensed service. If you interfere with another service, it's your problem, by definition. And if a licensed service interferes with you, it's your problem, by definition. Interference is more likely to be a problem if your network covers a large service area and if you are using high power.
- To use equipment that is approved for 802.11 service. Other amplifiers are available that cover the frequency range, but using them is illegal.

The FCC does enforce their rules, and their fines are large. If you're in violation of the regulations, they won't be amused, particularly if you're in excess of the power limit or using unapproved equipment.

A word about range

It's tempting to think that you can put up a high-gain antenna and a power amplifier and cover a huge territory, thus economizing on access points and serving a large number of users at once. This isn't a particularly good idea. The larger the area you cover, the more users are in that area—users your access points must serve. Twenty to 30 users for each wireless card in your access points looks like a good upper bound. A single access point covering a large territory may look like a good idea, and it may even work well while the number of users remains small. But if your network is successful, the number of users will grow quickly, and you'll soon exceed your access point's capacity.

Conducting the Site Survey

When working on the site survey, you must duplicate the actual installation as much as possible. Obstacles between wireless LAN users and access points decrease radio strength, so make an effort to replicate exactly the installation during the site survey.

If access points need to be installed in wiring closets, make sure the door is closed while testing so the survey accounts for the blocking effect of the door on radio waves. Antennas should be installed for the test exactly as they would be installed on a completed network. If office dwellers are part of the user base, make sure that adequate coverage is obtained in offices when the door is closed. Even more important, close any metal blinds, because metal is the most effective radio screen.

Signal measurements should be identical to the expected use of the network users, with one exception. Most site survey tools attempt to determine the signal quality at a single spatial point throughout a sequence of several points in time, and thus it is important to keep the laptop in one location as the measurement is carried out. Taking large numbers of measurements is important because users will move with untethered laptops, and also because the multipath fading effects may lead to pronounced signal quality differences even between nearby locations.

Have several copies of the map to mark signal quality measurements at different tentative access point locations, and note how the antenna must be installed at the location. If multiple antennas were used, note the type and location of each antenna.

Direct-Sequence Channel Layout

Most locations are deploying 802.11 products based on direct-sequence technology because the high–data rate products are based on direct-sequence techniques. Direct

sequence underlies both the 2-Mbps DS PHY and the 11-Mbps HR/DSSS PHY. Both standards use identical channels and power transmission requirements.

Direct-sequence products transmit power across a 25-MHz band. Any access points must be separated by five channels to prevent inter-access point interference. Selecting frequencies for wireless LAN operation is based partly on the radio spectrum allocation where the wireless LAN is installed. See Table 15-3.

Table 15-3. Radio channel usage in different regulatory domains

Channel number	Channel frequency (GHz)	US/Canada[a]	ETSI[b]	France
1	2.412	✓	✓	
2	2.417	✓	✓	
3	2.422	✓	✓	
4	2.427	✓	✓	
5	2.432	✓	✓	
6	2.437	✓	✓	
7	2.442	✓	✓	
8	2.447	✓	✓	
9	2.452	✓	✓	
10[c]	2.457	✓	✓	✓
11	2.462	✓	✓	✓
12	2.467		✓	✓
13	2.472		✓	✓

[a] 802.11 allows different rules regarding the use of radio spectrum in the U.S. and Canada, but the U.S. Federal Communications Commission and Industry Canada have adopted identical rules.
[b] Not all of Europe has adopted the recommendations of the European Telecommunications Standards Institute (ETSI). Spain, which does not appear in the table, allows the use of only channels 10 and 11.
[c] Channel 10 is allowed by all regulatory authorities and is the default channel for most access points when they are initially powered on.

Access points can have overlapping coverage areas with full throughput, provided the radio channels differ by at least five. Only wireless LANs in the U.S., Canada, and Europe that have adopted the ETSI recommendations can operate access points with overlapping coverage areas at full throughput.

After locating the access points, make sure that any access points with overlapping coverage are separated by at least five channels. The cellular-telephone industry uses the "hex pattern" shown in Figure 15-6 to cover large areas.

Part of the site survey is to establish the boundaries of access point coverage to prevent more than three access points from mutually overlapping, unless certain areas use multiple channels in a single area for greater throughput.

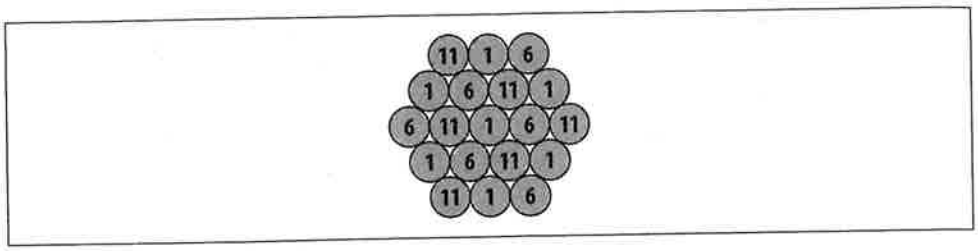

Figure 15-6. Frequency planning

Limitations of direct-sequence channel layout

One of the problems with 802.11 direct-sequence networks and 802.11b Direct-sequence networks is that there are only three nonoverlapping channels. Four channels are required for nonoverlapping coverage in two dimensions, and more channels are required for three dimensions. When laying out frequency channels in three dimensions, always keep in mind that radio signals may penetrate the floor and ceiling.

Application Performance Characterization

As part of the site survey, take some time to work with the "power users" to ensure that the application performance is adequate. Most applications use web frontends and are relatively tolerant with respect to long delays or coverage dropouts because web browsers retry connections. Terminal emulation and other state-oriented client/server applications may be less tolerant of poor coverage. Part of the engineering in installing a wireless LAN is to tailor the areas of overlapping coverage to offer denser coverage when the applications are less tolerant of momentary drops.

The End of the Site Survey: The Report

After the completion of the site survey, the technical details must be provided to installers to complete the network build-out. Consultants may use the site survey in different ways. Some consultants charge for the site survey and allow the customer or a third party to finish the installation. Value-added resellers may take the same approach or use the site survey to put together an installation bid for the customer.

Depending on the customer's requirements, some or all of the following details may be included in a site survey report:

1. A summary of the requirements from the initial preliminary work.
2. Estimated coverage areas based on the site survey measurements. This may be divided into areas with good coverage, marginal coverage, and weak coverage. It may also site potential trouble spots if the signal strength measurements allow for it.

3. A description of the locations of all access points, along with their configuration. Some elements of this configuration are the following:
   a. The access point name
   b. Its operating channel
   c. Approximate coverage area
   d. IP configuration
   e. Antenna type and configuration (including direction for directional antennas)
   f. Any other vendor-specific information
4. If the customer supplied detailed floor plans or physical network maps, those maps can be returned with detailed access point placement information. Estimated coverage areas can be noted on the map and serve as the basis for frequency reuse planning. Any antenna requirements (external antennas, antenna types, and adjustments to default transmission power) for achieving the noted coverage area should be recorded as well.
5. Many customers appreciate an estimate of the work necessary to install drivers onto any affected laptops. The scope of this item depends a great deal on the sophistication of the management tools used by the customer. For many, it will be sufficient merely to include a copy of the driver installation instructions as an appendix to the report. Some clients may require low-level details on the driver installation so that the driver installation can be completely automated down to any necessary registry changes on Windows systems.

Installation and the Final Rollout

After the site survey is finished, there should be enough information available to install a wireless LAN. Actual cabling and physical installation may be contracted out or performed by internal staff.

Recordkeeping

Careful documentation is an important part of any network build-out, but it is especially important for wireless LANs because the network medium is invisible. Finding network components is not always a simple matter of cable tracing! To document a wireless LAN, keep the following list in a safe place with the rest of the network maps:

- The site survey report.
- The annotated building blueprints with access point locations, names, and their associated coverage areas. If possible, the blueprints should also indicate areas of marginal or no coverage.

- A separate list of information about the access points in tabular form, which includes the location, name, channel, IP network information, and any other administrative information.

On the Naming of Access Points

Many institutions have naming policies that may dictate DNS names for wireless LAN access points. Device names should be as descriptive as possible, within reason. Companies that provide network service to other users, such as a "hot spot" provider, may wish to keep information about the detailed location of access points secret from users to keep the physical location of access points secret. Figure 15-7 illustrates a DNS naming convention in which the secrecy of access point locations is not a particular concern, so each name includes the geographic site location, the building name and floor, and the access point number and location on that floor. It also includes the wireless LAN name (SSID).

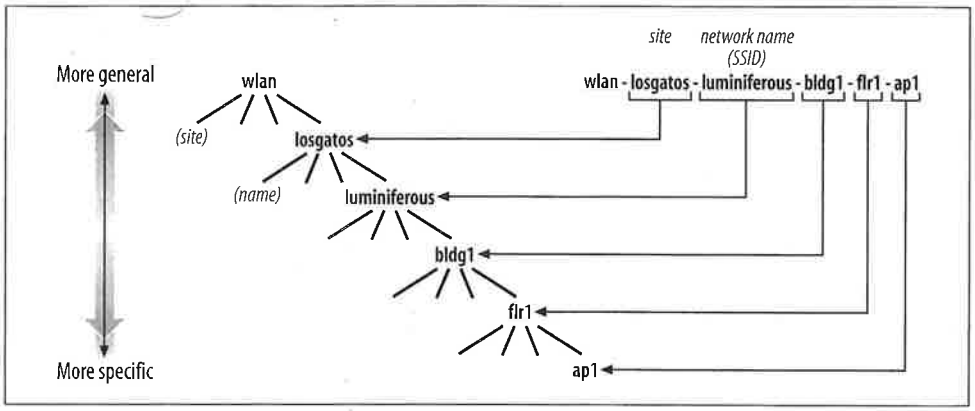

Figure 15-7. Wireless LAN naming convention for access points

All names are prefixed with *wlan-* to indicate clearly that they are associated with an 802.11 network. The next level of hierarchy is the geographic location of the wireless network; it can often be derived from existing site codes in large companies. Each site may compose a campus and have several buildings, but a single extended service area may offer coverage throughout the entire site at anything up through even midsized installations. The next level of hierarchy is the building identifier, which is often clear from existing conventions. Within a building, the floor number and the location of an access point within the floor can be used to further identify an access point. Figure 15-8 shows how the DNS name can be structured for an access point on the luminiferous network on the second floor of building one at a site in Los Gatos, California (*wlan-losgatos-luminiferous-bldg1-flr2-ap1*). In very large buildings, the access point number might even be replaced by a description of the location of the access point on the floor, such as *-ap-nw-4* for the fourth access point in the northwest corner.

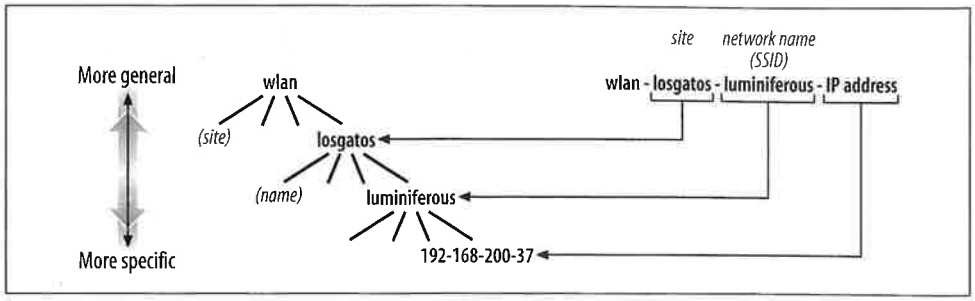

Figure 15-8. Convention for naming wireless LAN stations

To make troubleshooting easier, follow a convention for the naming of wireless LAN stations. A station on the same network as the access point described previously might be *wlan-losgatos-luminiferous-192-168-200-37*, in which the last set of numbers is the IP address.

Security

After installation is complete, you should execute a second test solely for security. Configuring access points can be time-consuming, detail-oriented work, and it is possible to forget to set a software option here and there. Check the following items to be sure your network is as secure as possible:

- If desired, WEP is enabled on all access points to prevent unauthorized association with the network.
- Lists of MAC addresses allowed to associate with the network have been distributed to each access point.
- Any access controllers in place are properly configured to block initial connections, and they can reach authentication servers to allow user access.
- Any VPN software is properly configured to accept connections from associated stations.
- Access points enforce restrictions on stations allowed to connect for management, and passwords are set.

Some vendors are advertising products that support 802.1x security, particularly in high-end access points. 802.1x requires that you set up a RADIUS authentication server, but the additional security is well worth the effort. On the other hand, 1x products are only just appearing as this book goes to press. Because 802.1x is a standard, products should interoperate, in theory—but we don't yet know whether they interoperate in practice. If you're not comfortable buying all your equipment from one vendor, you may want to stay away from 802.1x for a while. It's hard to make predictions that mean anything, but here's a guess: given the amount of attention that security has received lately, I expect that the 802.1x situation will be stabilized by the middle of 2002.

Several vendors have proprietary security solutions to replace or supplement WEP—some appear to be preliminary versions of 802.1x. I do not recommend locking yourself into a proprietary solution when a standard solution is available, or nearly available.

CHAPTER 16

802.11 Network Analysis

In the 1990s, computer professionals joined doctors as People With Answers. Just as doctors are asked medical questions by complete strangers, computer professionals are asked a bewildering variety of technical questions by complete strangers. When these strangers learn that I work with networks, I am often asked, "Why does the Internet break so often?" The more I contemplate the question, the more I believe that the question should be, "Why doesn't the Internet break *more* often?"

While I could never hope to answer either question in a single chapter of a book, it is obvious that network problems are a fact of life. Networks break, and wireless networks are no exception. Wireless LANs can improve productivity, but they also carry a larger risk of complete outage, and the limited bandwidth is almost sure to be overloaded. After building a wireless LAN, network engineers must be ready to investigate any problems that may arise.

As in many other network types, the trusty network analyzer is a key component in the engineer's toolbox. Network analyzers already exist for the wired backbone side of the wireless network and can be used productively in many troubleshooting scenarios. Wireless network troubleshooting depends on having a network analyzer for exactly the same reason. Sometimes, you just need to have a way of seeing what is on the airwaves. This chapter is devoted to tools that allow network engineers to do just that. Several commercial analyzers are available, and there are free tools that run on Linux. Before diving into the tools, though, it may help to consider why wireless network analyzers are a practical requirement for the network administrator.

Why Use a Network Analyzer?

In spite of the shared heritage, 802.11 is not Ethernet. It has a number of additional protocol features, each of which can cause problems. Fixing problems on 802.11 networks sometimes requires that a network administrator get down to the low-level

protocol details and see what is happening over the airwaves. Network analyzers have long been viewed as a useful component of the network administrator's toolkit on wired networks for their ability to report on the low-level details. Analyzers on wireless networks will be just as useful, and possibly even more important. More things can go wrong on an 802.11 network, so a good analyzer is a vital tool for quickly focusing troubleshooting on the likely culprit.

Avoiding problems begins at the planning stages. Some analyzers can report detailed statistics on RF signal strength, which can help place access points. Analyzers can help network administrators avoid creating dead zones by ensuring that there is enough overlap at the edges of BSSs to allow for timely transitions. As wireless networks grow in popularity, they may need to support more users. To avoid performance problems, administrators may consider shrinking the size of access point coverage areas to get more aggregate throughput in a given area. In the process of shrinking the coverage areas, network administrators may go through large parts of the deployment plan all over again and depend once again on their analyzer.

With the limited bit rates of wireless networks, performance is likely to be a problem sooner or later. Performance problems can be caused by cramming too many users into too few access points, or they can be related to problems happening at the radio layer. The designers of 802.11 were aware of the problems that could be caused by the radio transmission medium. Frame transmissions succeed reliably. Most implementations will also retransmit frames with simpler (and slower) encoding methods and fragment frames in the presence of persistent interference.

Interference is a major problem for 802.11 network performance. In addition to the direct effect of trashing transmitted frames that then require retransmission, interference has two indirect effects. Poor transmission quality may cause a station to step down to a lower bit rate in search of more reliable radio link quality. Even if slower transmissions usually succeed, some measure of throughput is lost at the lower bit rates. 802.11 stations may also attempt to fragment pending frames to work around interference, which reduces the percentage of transmissions that carry end user data. 802.11 headers are quite large compared to other LAN protocols, and fragmentation increases the amount of header information transmitted for a fixed amount of data.

On many networks, however, only a few applications are used. Do performance complaints indicate a general network problem, or a problem with a specific application? Network analyzers can help you find the cause of the problem by examining the distribution of packet sizes. More small packets may indicate excessive use of fragmentation in the face of interference. Some analyzers can also report on the distribution of frames' transmission rates on a wireless network. 802.11b networks are capable of transmitting at 11 Mbps, but frames may be transmitted at slower rates (5.5 Mbps, 2 Mbps, or even 1 Mbps) if interference is a problem. Stations capable of high-rate operation but nonetheless transmit at lower rates may be subject to a large amount of interference.

To solve interference problems, you can attempt to reorient the access point or its antenna, or place a new access point in a zone with poor coverage. Rather than waiting for users to report on their experience with the changes in place, you can use an analyzer to get a quick idea of whether the changes will help alleviate the problem. Some analyzers can provide extensive reports on the RF signal quality of received frames, which can help you place hardware better the first time around. Avoiding repeated experimentation directly with end users makes you look better and makes users happier. Shortening troubleshooting cycles has always been a strength of network analyzers.

Analyzers also help network administrators check on the operation of unique features of the 802.11 MAC. While it is possible to capture traffic once it has been bridged on to a wireless backbone network and analyze it there, the problem could always be on the wireless link. Are frames being acknowledged? If they are not, there will be retransmissions. Are the direct-sequence bits set correctly? If they are not, then address fields will be misinterpreted. If a malformed packet is seen on the wired side of an access point, it could be mangled at several points. A wireless analyzer can look at frames as they travel through the air to help you pin down the source of the mangled packet. Malformed frames may be transmitted by the client or mangled by the access point, and it is helpful to pin down the problem before requesting assistance from the vendor.

Security is a major concern for wireless networks, and wireless analyzers can be used to monitor wireless networks for security problems. They can look at the MAC addresses of all stations to look for unknown addresses, though this may or may not be all that useful in practice. It is probably impossible to know all the MAC addresses used on your network, though it might be possible to spot cards from manufacturers of hardware that is not part of a standard build. It may be more effective to look for failed attempts to authenticate to your access points.

Some installations will rely on WEP for security, either totally or in part. Some commercial analyzers offer the ability to decrypt frames processed by WEP, provided they are given the shared WEP key. 802.11 frames have enough information to enable a sniffer in possession of the key to do real-time decryption. This ability allows network administrators to peer into frames protected by WEP to perform higher-level protocol analysis and to check that WEP processing is not mangling frames. Real-time decryption is a byproduct of the poor design of WEP. Once WEP is replaced by a real security system, real-time decryption may become much more complicated.

802.11 Network Analyzers

802.11 network analyzers are now quite common and should be a part of any wireless LAN administrator's toolbox. Most 802.11 network analyzers are software packages

that use an 802.11 network card. No special hardware is required because commodity 802.11 network cards supply all the RF hardware needed to grab packets. The only catch is that each software package usually only works with a limited number of cards on the market.

Commercial Network Analyzers

With the stunning growth of wireless networks, commercial software vendors rushed to introduce wireless versions of successful wired network analyzers. The two main commercial wireless network analyzers are Sniffer Wireless from Network Associates (*http://www.sniffer.com*) and AiroPeek from WildPackets (*http://www.wildpackets.com*). Sniffer Wireless requires the use of wireless LAN cards from either Cisco or Symbol. AiroPeek supports a much broader range, including cards made by 3Com, Cisco, Nortel, Intel, Symbol, and Lucent.

Like their wired relatives, the commercial wireless LAN analyzers have a host of features. In addition to decoding captured frames, they can filter the captured frames based on anything in the 802.11 header, report statistical data on the packet size and speed distributions, monitor the real-time network utilization, and quickly scan all the available channels to detect all networks in the area. If given the key, both can decrypt frames protected by WEP.

A network analyzer should be part of the deployment budget for any wireless network. The choice to buy or build is up to you, though I anticipate that most institutions will rely primarily on commercial products and leave development and bug fixes to the network analyzer vendors, especially because commercial analyzers can be purchased, installed, and made useful much more quickly.

Ethereal

Ethereal is the standard open source network analyzer. Like the proprietary analyzers, it supports a long list of protocols and can capture live data from a variety of network interfaces. Unlike the proprietary analyzers, Ethereal comes complete with a slogan ("Sniffing the glue that holds the Internet together").

Ethereal runs on most Unix platforms as well as Windows. Source code is freely available for both, but modifications are easier to make on Unix because of the availability of free compilers for the Unix programming environment. Like many open source projects, Ethereal is distributed under the terms of the GNU Public License. Protocol decodes are included for many common networking protocols. For the purpose of this section, the important protocols are IEEE 802.11 and LLC, both of which are used on every 802.11 frame. Of course, the TCP/IP suite is included as well.

For 802.11 network analysis with Ethereal, Linux is the platform of choice. The *linux-wlan-ng* driver can be modified to feed raw 802.11 packets to Ethereal, and Ethereal can be modified to decode them in real time. For data capture on Linux, only Intersil-based cards are supported. Of course, Intersil cards can be used to monitor any 802.11 network, including those that use other chipsets.

Compilation and Installation

Binary packages are available from the main project web site at *http://www.ethereal.com*. Because of the required modifications to the code, a binary package is not sufficient.

Prerequisites

Before compiling Ethereal, both *libpcap* and the GTK+ library must be installed. *libpcap* is the packet capture library from *tcpdump*, and the GTK+ library is the GIMP tool kit. For 802.11 network analysis, *libpcap* must be modified, so it needs to be built from source. GTK+ is used unmodified, however, and can typically be installed from a package that came with your Linux distribution. GTK+ source is available from *http://www.gtk.org/* and is built with the standard *./configure*, *make*, and *make install* sequence familiar to most open source software users. If you opt to go the package route, it is necessary to install both the libraries themselves and the *-devel* versions to install the required header files.

Several utility programs must be installed on the system for Ethereal to build properly. Ethereal depends on GNU *make*, which is the default *make* program on Linux systems. Perl is used to assemble the manpages. The modified *libpcap* that grabs raw 802.11 frames needs to build a parser, which requires the *flex* lexical scanner and either *yacc* or *bison*. Users intending to do NetWare Core Protocol (NCP) analysis must have Python installed.*

Ethereal depends on one kernel function. Because it interfaces directly with the kernel to grab packets, it requires that the kernel be built with Packet Socket support (*CONFIG_PACKET*).

Compiling the modified libpcap

The next step is the installation of the slightly modified *libpcap* library. *libpcap* is available from *http://www.tcpdump.org*. Tim Newsham's monitoring patch is intended to be applied against 0.6.2, the latest version as of the writing of this book. At slightly over 200 lines, the patch is fairly lightweight. Its main purpose is to add another type of packet capture to the *libpcap* library for 802.11-specific captures. Essentially, it adds the functionality of *prismdump* to *libpcap*, which makes it much

* Network administrators who deal with NCP also have my sympathy.

easier to write programs that manipulate raw 802.11 frames. The patch also defines a data structure that holds miscellaneous data of interest to network analyzers, such as the received signal strength and information on the signal-to-noise ratio.

First, fetch the wireless LAN monitoring patches from *http://www.lava.net/~newsham/wlan/*. Get the *wlan-mods.tgz* package, which includes the patch for *libpcap*, the patch for Ethereal, and a new system header file to describe the PRISM capture type:

```
$ patch -p1 < ../wlan-mods/libpcap.patch
patching file Makefile.in
patching file gencode.c
patching file pcap-linux.c
patching file pcap-prism.c
patching file pcap-prism.h
```

The newly modified libpcap depends on the *802.11h* file included in the *wlan-mods* package. *802.11h* includes definitions for the raw PRISM monitoring. Copy it to the main system *include* file location (probably */usr/include*). After applying the patch, building *libpcap* follows the usual routine of configuration, building, and installing:

```
$ ./configure
creating cache ./config.cache
checking host system type... i686-pc-linux-gnu

(many lines of configuration output omitted)

updating cache ./config.cache
creating ./config.status
creating Makefile
creating config.h
$ make
gcc -O2 -I.  -DHAVE_CONFIG_H -c ./pcap-linux.c

(many lines of compilation output omitted)

ar rc libpcap.a pcap-linux.o pcap.o inet.o gencode.o optimize.o nametoaddr.o
etherent.o savefile.o bpf_filter.o bpf_image.o bpf_dump.o pcap-prism.o scanner.o
grammar.o version.o
ranlib libpcap.a
```

Finally, run *make install* to put *libpcap* in its place. With the installation complete, you can proceed to link the modified *libpcap* with Ethereal, which gives Ethereal the ability to work with 802.11 frames.

Building Ethereal itself

You can build Ethereal after installing the modified *libpcap* and the associated header file. Grab the source code for Ethereal from *http://www.ethereal.com* and uncompress it into a working directory:

```
$ tar -xzvf ../ethereal-0.8.17-a.tar.gz
ethereal-0.8.17/
```

```
ethereal-0.8.17/Makefile.in
ethereal-0.8.17/debian/
ethereal-0.8.17/debian/README.debian

(many other filenames omitted)
```

Next, apply Tim's monitoring patch, which is written against Ethereal Version 0.8.17. The patch does not apply cleanly against later versions, though with some effort it could undoubtedly be made to do so.

```
$ patch -p1 < ../wlan-mods/ethereal.patch
patching file Makefile.am
patching file capture.c
patching file packet-ieee80211.c
patching file packet-prism.c
patching file packet-prism.h
patching file wiretap/libpcap.c
patching file wiretap/wtap.h
```

You're now ready to begin building the source code. Ethereal ships with an autoconfiguration script that you run from the root directory of the source tree. The autoconfiguration script performs a series of tests to assist in the compilation process. On one of my systems, the configure script had trouble finding the GLIB library, so I had to point it at the correct location. On Mandrake 7.2, GLIB is installed in */usr/lib*. The GTK prefix is used to find the *gtk-config* script. I specified */usr* because */bin* will be appended to the specified path. As a result of my configuration, the Ethereal configuration script looked in */usr/bin* for *gtk-config* and found it:

```
$ ./configure --with-gtk-prefix=/usr/lib --with-gtk-exec-prefix=/usr
creating cache ./config.cache
checking for a BSD compatible install... /usr/bin/install -c
checking whether build environment is sane... yes

(many other test results omitted)

creating ./config.status
creating Makefile
creating config.h

The Ethereal package has been configured with the following options.
                    Build ethereal : yes
                   Build tethereal : yes
                     Build editcap : yes
                     Build randpkt : no
                      Build dftest : no

                    Install setuid : no
                   Use pcap library : yes
                   Use zlib library : yes
          Use IPv6 name resolution : no
                   Use SNMP library : no
```

Note the summary at the end of the configuration script. When I failed to identify the location of the GLIB library, Ethereal was perfectly happy to configure for terminal-only

analysis. Check the summary to be sure you are building the features you want. After configuration, run *make*, which builds:

Ethereal
: The X-based graphical analyzer described in the rest of this section.

tethereal
: A terminal-based analyzer that uses the same core packet-analysis code.

editcap
: A program that manipulates capture files and can translate between several common capture formats. By default, it uses the file format used by *libpcap*, though it also supports snoop, Sniffer traces, NetXray, and Microsoft Network Monitor captures. For a complete list of the supported file types, see the manual page for *editcap*.

Finally, install the executables using *make install*, which puts the executable in the location specified by the *Makefile*. If no install directory is explicitly specified in the configuration step, Ethereal is installed in the */usr/local* hierarchy, with executables in */usr/local/bin* and man pages in */usr/local/man*.

Ethereal on Windows

Binary packages are more important in the Windows world because of the lack of a high-quality, free-development environment. Though Ethereal does not provide the same level of 802.11 support under Windows, it can still be a valuable program to have, especially in a day job that requires use of Windows systems. Binary versions of Ethereal for Windows are available from *http://www.ethereal.com*. They require the WinPcap library to provide the *libpcap*-type support on Windows. WinPcap can be downloaded from *http://netgroup-serv.polito.it/winpcap/*. WinPcap is supported only on 32-bit Windows systems (95, 98, ME, NT, and 2000) and is licensed under a BSD-style license. Interestingly enough, WinPcap was supported in part by Microsoft Research.

Running Ethereal

To start Ethereal, run it from the command line. Any user may start Ethereal, but root privileges are required to capture packets. Other users may load Ethereal to analyze capture files and perform analysis, though.

```
[gast@bloodhound ethereal-0.8.17]$ ethereal &
```

Starting Ethereal pops up the main window, which is shown in Figure 16-1. The main window has three panes. The top pane, called the *packet list pane*, gives a high-level view of each packet. It displays each packet's capture time, source and destination address, the protocol, and a basic decode of the packet. The Protocol field is

filled in with the final decode, or *dissector*, used to analyze the frame. On 802.11 networks, the final decode may be IEEE 802.11 for management frames, or it may go all the way to the final TCP protocol for analysis, as in the case of an 802.11 frame holding an LLC-encapsulated IP packet with a TCP segment carrying HTTP.

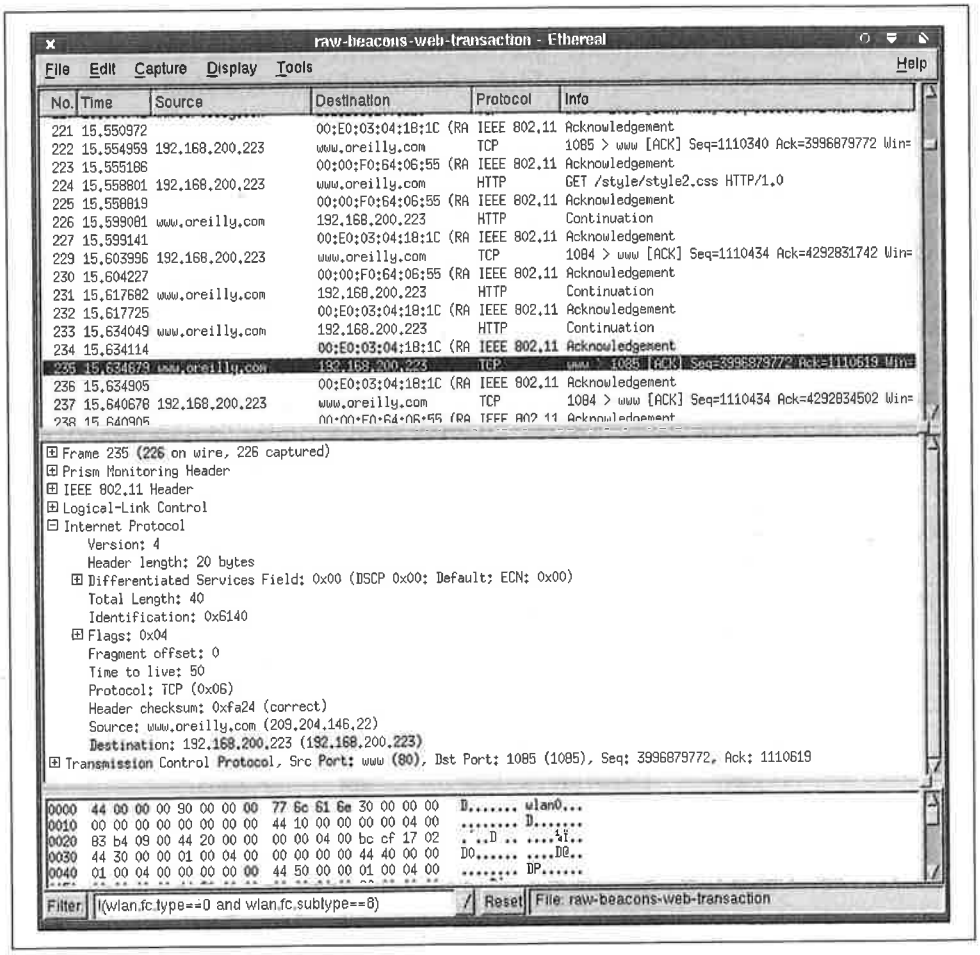

Figure 16-1. Main Ethereal window

The middle pane, called the *tree view pane*, is a detailed view of the packet selected in the packet list. All the major headers in a packet are shown and can be expanded for more detail. All packets have the basic "Frame" tree, which contains details on arrival time and capture length. On 802.11 networks, all frames have the Prism Monitoring header, which contains radio-link data. Data packets on 802.11 networks also have a Logical Link Control (LLC) header. From there, the LLC may contain ARP packets, IP packets, TCP segments, and so on. Ethereal includes dissectors for all the commonly used protocols, so 802.11 frames are fully decoded most of the time. The

bottom pane is called the *data view pane*. It shows the raw binary data in the selected packet. It also highlights the field selected in the tree view pane.

At the bottom of the Ethereal window is a bar with four important elements. The leftmost button, **Filter:**, is used to create filters that reduce the captured packet list to the packets of interest. The text box just to the right allows you to enter filters without going through the construction process. Ethereal maintains a filter history list that enables easy switching between filters. At the right is a text field that displays several kinds of information, depending on what Ethereal is doing. It may indicate that Ethereal is currently capturing data, display the name of the capture file loaded, or display the field name currently highlighted in the tree view.

Capturing data

Capturing data is straightforward. Go to the **Capture** menu and choose **Start**. The Capture Preferences window, shown in Figure 16-2, opens. The first thing to do is select the interface you want to monitor. Interface selection has one small wrinkle. Naturally, Ethereal can use any interface it detects, even *wlan0*. However, all Ethernet drivers in the Linux kernel present Ethernet frames. If you choose *wlan0*, you miss out on all 802.11 control and management frames, and the Data frames lose the 802.11 headers by the time they get to Ethereal's capture engine. The reason for applying the patch to Ethereal during compilation is that it creates a pseudo-interface, *prism*, which allows Ethereal to read data directly from the hardware. That way, Ethereal uses the same capture code that *prismdump* does and can display all the 802.11 traffic in the air. *prism* does not appear as an interface in the drop-down box, but it can be typed into the field directly, as in Figure 16-2.

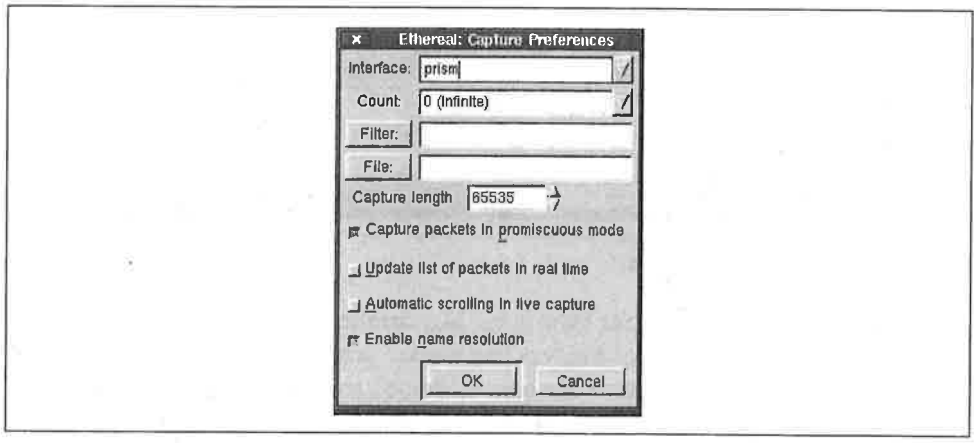

Figure 16-2. Selecting the prism pseudo-interface

Ethereal accepts the *–i* command-line option to specify an interface. If you plan to do all of your analysis on one interface, you can define a shell mapping of *ethereal* to *ethereal –i prism*.

Capture Length is what *tcpdump* calls the *snap length*. It is the amount of data from each packet that will be captured. *tcpdump* uses a short snap length and saves only IP and TCP headers. By default, Ethereal grabs the entire packet, which facilitates much more detailed offline analysis.

I typically turn on "Update list of packets in real time" and "Automatic scrolling in live capture". If the former is left unselected, the trace appears only when the capture stops. If the latter is left unselected, the trace does not scroll to the bottom. Speed is important to real-time analysis. Disabling name resolution eliminates overhead for every packet captured and may allow a station to avoid missing frames in the air.

Saving data to a file

To save data for offline analysis, use the File → Save As option from the menu bar. Selecting it displays the dialog box shown in Figure 16-3.

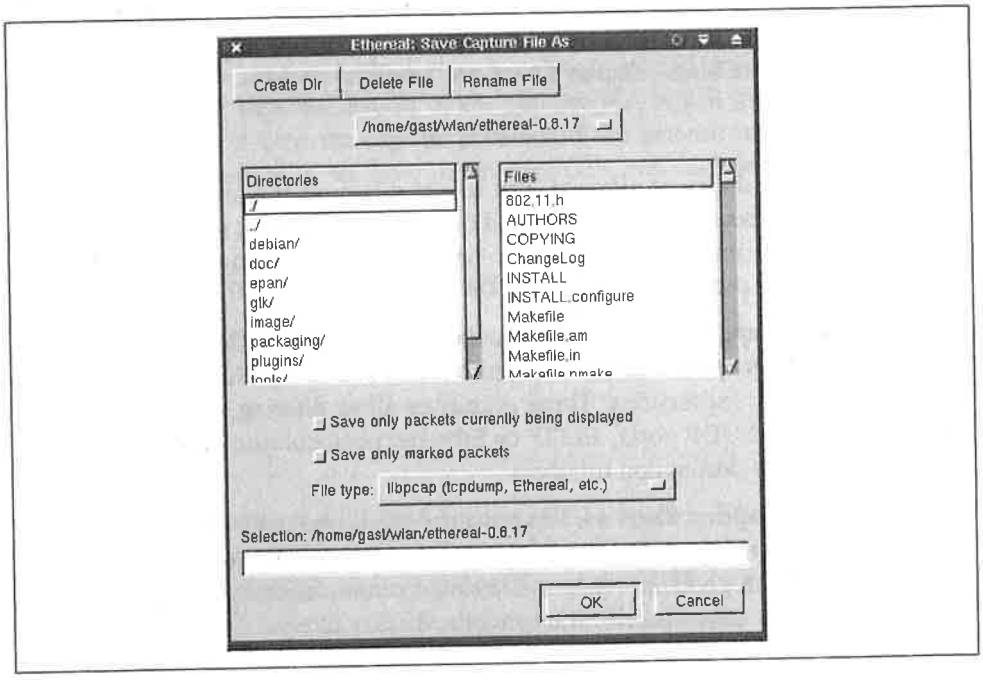

Figure 16-3. Save As dialog box

Choosing a filename is straightforward. The real power of Ethereal is that extraneous packets can be eliminated by using the two checkboxes below the file chooser. "Save only packets currently being displayed" saves only the packets that made it through the display filter, greatly reducing the amount of data in the capture file. "Save only marked packets" saves only packets that you have selected. If neither option is checked, Ethereal saves the entire trace. Saving an entire trace may be necessary at times, but it's worth avoiding; an entire trace could include all the packets that have crossed your network over an extended period. Using a display filter and saving only the displayed packets makes the trace far more manageable.

The file type selection allows the file to be saved to a *libpcap* file format. To imply that there is only one *libpcap* file format would be wrong, though. Four choices are available, including one that allows you to read capture files collected from *tcpdump* on Nokia's IPSO-based network appliances.

Data Reduction

Raw captures can be quite large, and extraneous packets can make finding wheat among the chaff a challenge. One of the keys to successful use of a network analyzer is to winnow the torrent of packets down to the few packets at the heart of the matter. Ethereal provides three ways to reduce the amount of data captured to a manageable amount: capture filters, display filters, and marking packets.

Capture filters

Capture filters are the most efficient way to cut down on the amount of data processed by Ethereal because they are pushed down into the packet sniffing interface. If the packet capture interface discards the packet, that packet does not make it to Ethereal for further processing.

Ethereal uses *libpcap*, so the capture filter language is exactly the same as the language used by *tcpdump*. A number of primitives are available, which can be grouped into arbitrarily long expressions. These primitives allow filtering on Ethernet and IP addresses, TCP and UDP ports, and IP or Ethernet protocol numbers. Many can be applied to source or destination numbers.

All in all, though, capture filters are less powerful than display filters for a simple reason: capture filters must operate in real time (i.e., as the packets are arriving over the network interface). A good approach to filtering is to use the capture filters to make a rough cut, then fine-tune the selection using the display filters.

Display filters

Display filters can be used on any field that Ethereal identifies, which makes them far more powerful than capture filters. Display filters inherit the knowledge of all the dissectors compiled into Ethereal, so it is possible to filter on any of the fields in any

of the protocols that Ethereal is programmed to recognize. Wireless LAN administrators can filter frames based on anything in the 802.11 or LLC headers. Examples specific to 802.11 are presented later in this chapter.

Marking packets

You can mark any packet by pressing Ctrl-M. Marking is essentially a manual filter. Marked packets are highlighted in the packet view pane. The only reason to mark a packet is to perform a manual reduction of a trace to a few interesting packets. When eliminating large amounts of data, automated filters are faster.

Analysis Tools

Several additional tools are available for Ethereal users. Packets can be colorized according to protocol. This chapter does not take advantage of the feature because the traces presented in the case study sections are short, and colorization doesn't help much.

Ethereal can reconstruct a TCP stream. For example, a reconstructed HTTP transaction would likely show several objects being fetched from a web page, as well as the HTML text used to create the page.

Summary statistics are available for each capture loaded into Ethereal as well. Administrators can view the details of the capture file, which includes information such as the length of time the capture covers and the amount of traffic on the network. It is also possible to see how much data falls into each level of the protocol hierarchy.

Using Ethereal for 802.11 Analysis

Several Ethereal features are handy when applied to 802.11 networks. This section lists several tips and tricks for using Ethereal on wireless networks, in no particular order.

Display filters

Ethereal allows filtering on all fields in the 802.11 header. Frame fields are structured hierarchically. All 802.11 fields begin with *wlan*. Two subcategories hold information on the Frame Control field (*wlan.fc*) and the WEP Information (*wlan.wep*) field. Figure 16-4 shows the variable names for 802.11 header components; in the figure, each field is labeled with a data type. Boolean fields are labeled with a B, MAC addresses with MA, and unsigned integers with U plus the number of bits. Table 16-1 shows the same information, omitting the Ethereal display fields that are unlikely to be useful for filtering.

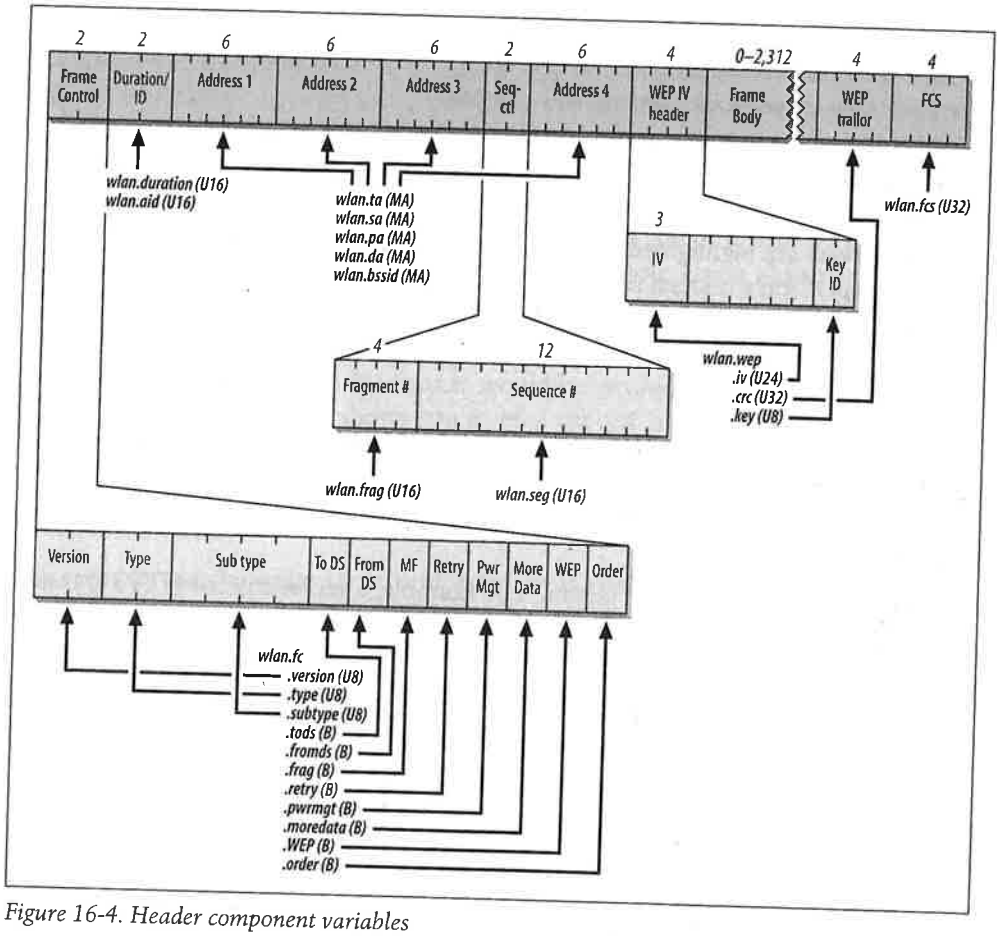

Figure 16-4. Header component variables

Table 16-1. Ethereal fields for 802.11 header components

802.11 header field	Ethereal field
Header fields	
Duration	*wlan.duration*
Association ID	*wlan.aid*
Transmitter address	*wlan.ta*
Source address	*wlan.sa*
Receiver address	*wlan.ra*
Destination address	*wlan.da*
BSSID	*wlan.bssid*
Fragment number	*wlan.frag*

Table 16-1. Ethereal fields for 802.11 header components (continued)

802.11 header field	Ethereal field
Sequence number	*wlan.seq*
Frame control subfields	
Version	*wlan.fc.version*
Frame type	*wlan.fc.type*
Frame subtype	*wlan.fc.subtype*
ToDS flag	*wlan.fc.tods*
FromDS flag	*wlan.fc.fromds*
Fragment flag	*wlan.fc.frag*
Retry flag	*wlan.fc.retry*
Power management flag	*wlan.fc.pwrmgt*
More Data flag	*wlan.fc.moredata*
WEP flag	*wlan.fc.wep*
Order flag	*wlan.fc.order*
WEP fields	
Initialization vector	*wlan.wep.iv*
Key identifier	*wlan.wep.key*

Fields can be combined using operators. Ethereal supports a standard set of comparison operators: `==` for equality, `!=` for inequality, `>` for greater than, `>=` for greater than or equal to, `<` for less than, and `<=` for less than or equal to. An example of a display filter would be `wlan.fc.type==1` to match Control frames.

Logical operators `and` and `or` are supported; as in many programming languages, the exclamation point is used for logical negation. Boolean fields can be tested for existence, so Control frames with WEP enabled would be matched by the display filter `wlan.fc.type==1 and wlan.fc.wep`.

Figure 16-5 shows a complete 802.11 header in the tree view. Selecting the 802.11 header in the tree view highlights the bits that comprise the 802.11 header in the ASCII view at the bottom. Expanding the 802.11 header tree decodes all the fields in the 802.11 header.

Compared to Control and Data frames, 802.11 Management frames have a great deal of structure. Ethereal decodes Management frames into two parts. *Fixed Parameters* in the tree view pane correspond to the fixed fields of 802.11 management frames. *Tagged Parameters* are the variable fields and are decoded in the tree view pane. Table 16-2 shows the fixed fields that can be searched on in Ethereal, as well as the capability flags.

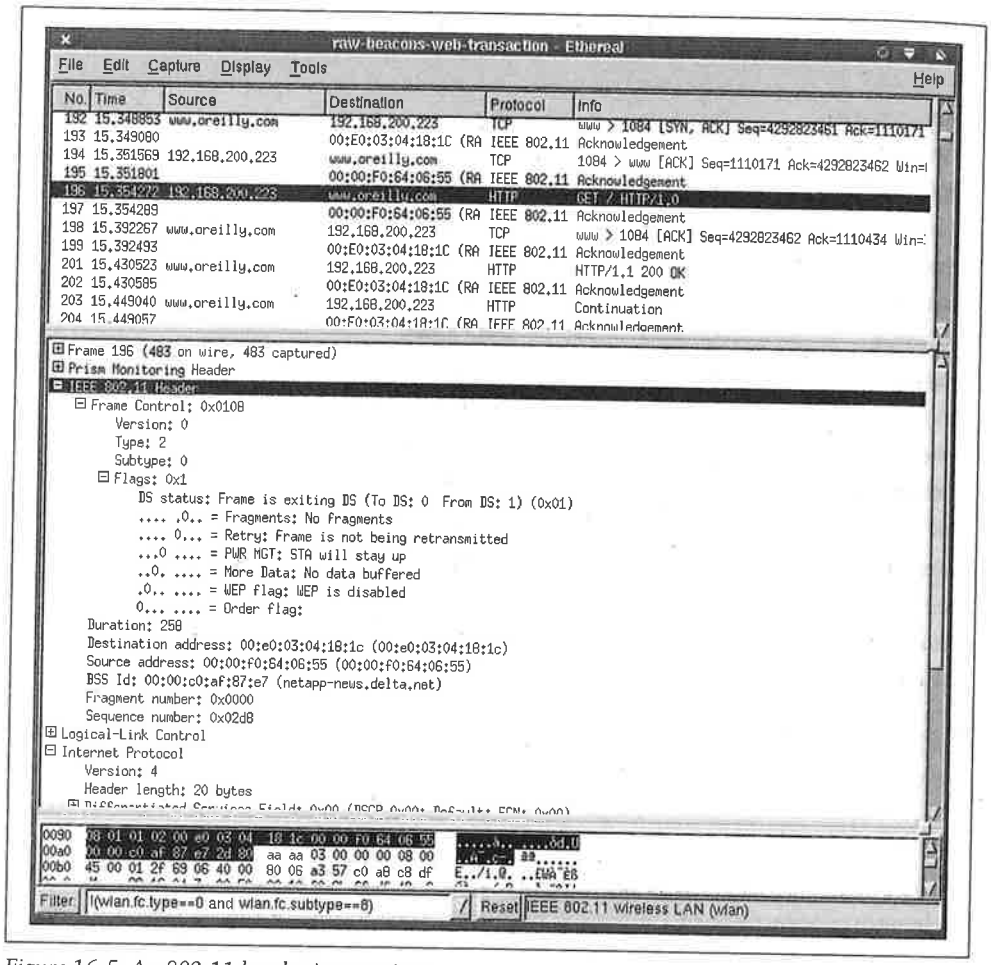

Figure 16-5. An 802.11 header in tree view

Table 16-2. Fixed Management frame components

802.11 management field	Ethereal field
Fixed fields	
Authentication Algorithm Number	*wlan_mgt.fixed.auth.alg*
Authentication Transaction Sequence Number	*wlan_mgt.fixed.auth_seq*
Beacon Interval	*wlan_mgt.fixed.beacon*
Current AP address	*wlan_mgt.fixed.current_ap*
Listen Interval	*wlan_mgt.fixed.listen_ival*
Association ID	*wlan_mgt.fixed.aid*
Timestamp	*wlan_mgt.fixed.timestamp*

Table 16-2. Fixed Management frame components (continued)

802.11 management field	Ethereal field
Reason Code	*wlan_mgt.fixed.reason_code*
Status Code	*wlan_mgt.fixed.status_code*
Capability info	
ESS flag (is the BSS part of an ESS?)	*wlan_mgt.fixed.capabilties.ess*
IBSS flag (is the BSS independent?)	*wlan_mgt.fixed.capabilities.ibss*
Contention-free station polling bit	*wlan_mgt.fixed.capabilities.cfpoll.sta*
Contention-free AP polling bit	*wlan_mgt.fixed.capabilities.cfpoll.ap*
Privacy flag (is WEP implemented?)	*wlan_mgt.fixed.capabilities.privacy*
Preamble (is 802.11b short preamble implemented?)	*wlan_mgt.fixed.capabilities.preamble*
PBCC (is 802.11b PBCC coding implemented?)	*wlan_mgt.fixed.capabilities.pbcc*
Channel Agility (is 802.11b Channel Agility implemented?)	*wlan_mgt.fixed.capabilties.agility*

In the tree view pane, the fixed and variable fields show up in different trees. Variable information elements are decoded under the tagged tree item according to their type. Figure 16-6 shows the decoding of the information elements in a Beacon frame. The access point generating the Beacons is indeed an access point for the Hack Me network on direct-sequence channel 11. It supports both the older 1-Mbps and 2-Mbps encodings in addition to the faster 802.11b encoding schemes.

At the end of the Beacon is a variable field with a reserved tag number that the wireless card's vendor uses to hold the access point's name, and possibly the access point's load factor. The ASCII dump displays the word Outside clearly, which is, in fact, the name of this access point.

Excluding Beacon frames

Beacon frames can get in the way when working with a raw 802.11 trace. The sheer number of frames obscures patterns in the data. Therefore, it's common to exclude those frames from display. Frame type information is carried in the Frame Control field of the 802.11 header. Beacon frames are identified by a Type code of 0 for Management frames, with a subtype of 8 for Beacon. The filter matching the Type code is `wlan.fc.type==0`, and the filter matching the Subtype code is `wlan.fc.subtype==8`. Therefore, to discard frames that match both these conditions, one possible filter is `!(wlan.fc.type==0 and wlan.fc.subtype==8)`.

PRISM monitoring header

The modifications to *libpcap* add a PRISM pseudo-header to any captured frames. Some of the information in this header corresponds to the information that would be kept in the PLCP header. Figure 16-7 shows the pseudo-header on a frame.

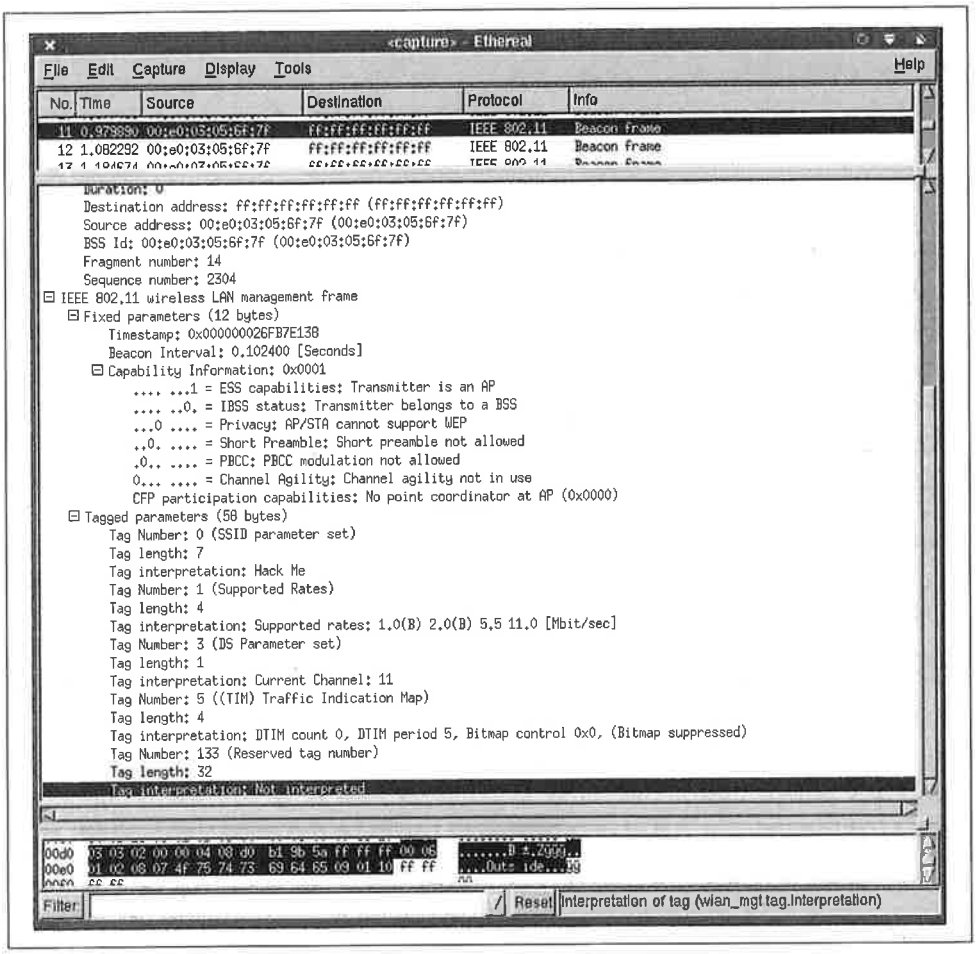

Figure 16-6. Beacon frame decode

Four fields of note are reported by the PRISM capture:

MAC Time
: A timestamp added by the MAC counter to each received frame.

Signal
: Indicates the signal strength of the received packet.

Noise
: Quantifies background noise during the packet time. Calculating the signal-to-noise ratio is straightforward from these two fields.

Rate
: Encoded according to the MAC framing conventions described in Chapter 4. In Figure 16-7, the network is operating at 2 Mbps because the rate field is 0x04.

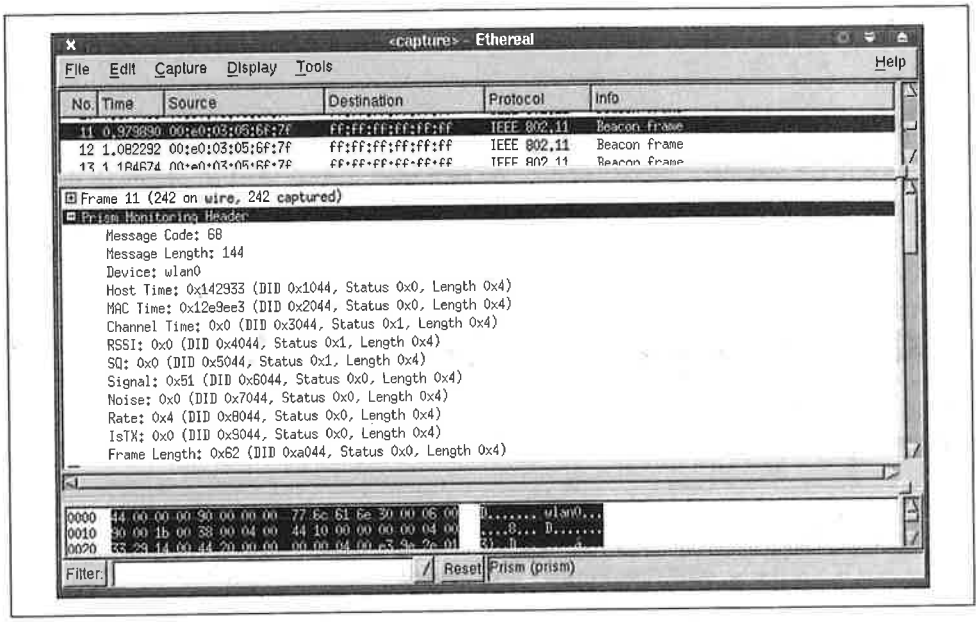

Figure 16-7. PRISM monitoring header

Understanding the LLC header

To multiplex higher-level protocol data over the wireless link, 802.11 uses the LLC SNAP encapsulation. (SNAP encapsulation was described at the end of Chapter 3.) 802.11 does not include a protocol field, so receivers cannot discriminate between different types of network protocols. To allow multiple protocols, an 8-byte SNAP header is added. The SNAP header is decoded in Ethereal's tree view, as shown in Figure 16-8.

Highlighting the LLC header in the tree view shows the corresponding 8-byte header in the packet dump. The eight bytes in the SNAP header are clearly visible in the data view pane. Five fields make up the header:

The destination service access point (DSAP)
: This is always set to 0xAA for SNAP encapsulation.

The source service access point (SSAP)
: This is always set to 0xAA for SNAP encapsulation.

Control
: This is derived from HDLC. Like all data transfer using HDLC, it labels the data following the LLC header as unnumbered information. Unnumbered information indicates the use of a connectionless data transport and that the data need not be sequenced or acknowledged.

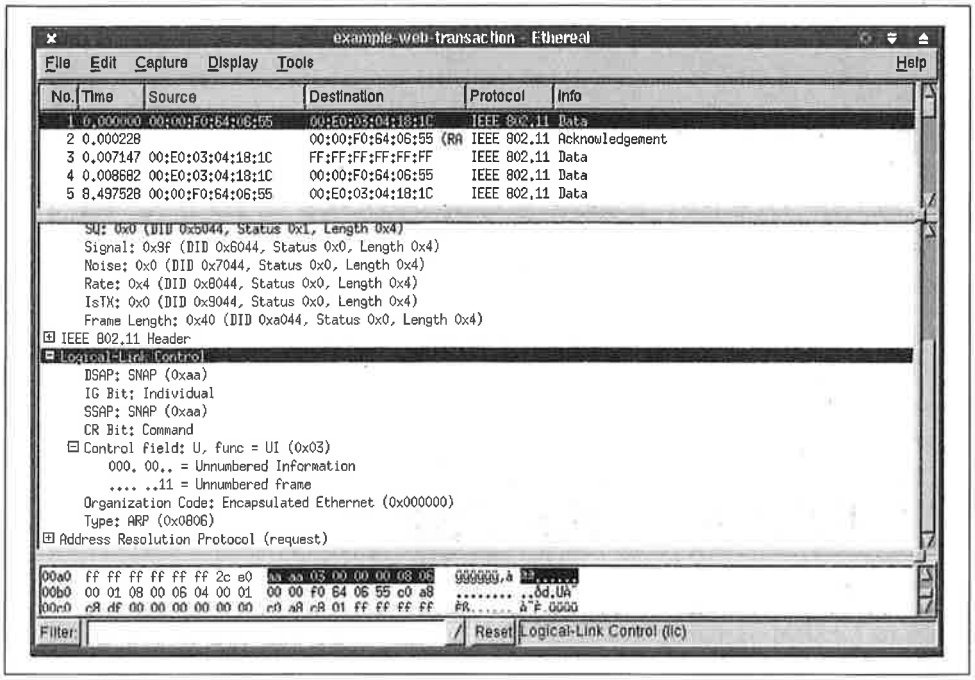

Figure 16-8. LLC SNAP header

An organizationally unique identifier (OUI)
: This is used to determine how to interpret the following bytes. IP is encapsulated in LLC using the standard in RFC 1042, which specifies the use of the OUI 0x00-00-00. (Some vendors may use an assigned OUI for proprietary communications.)

Protocol Type
: This is copied from the corresponding Ethernet frame. The Type field matches the Ethernet type codes. On IP networks, it will be either 0x0800 for IP or 0x0806 for ARP.

LLC encapsulation is required by the 802.11 specification because it saves the 802.11 frame from having to carry protocol information directly.

802.11 Network Analysis Examples

To illustrate how a network analyzer can aid network engineers in looking at wireless LAN traffic, this section presents three detailed examples network analyzers to answer questions about a wireless LAN. The examples are contrived, but nonetheless show the types of operations that are common with network analyzers. The

examples are described specifically for Ethereal but can be carried out with any of the commercial alternatives described earlier.

Case Study 1: Access Point Name and Workload Information

As an example of how to use Ethereal, consider the typical driver display on Windows. Usually, there is a function within the driver that allows a card to display the name of an access point supplied by the same vendor. As an example, consider the transmission of access point name and workload information by the Nokia A032. The trace in Figure 16-9 was taken in the vicinity of a Nokia A032 with no stations associated with it. Naturally, there are many Beacon frames announcing the A032's existence.

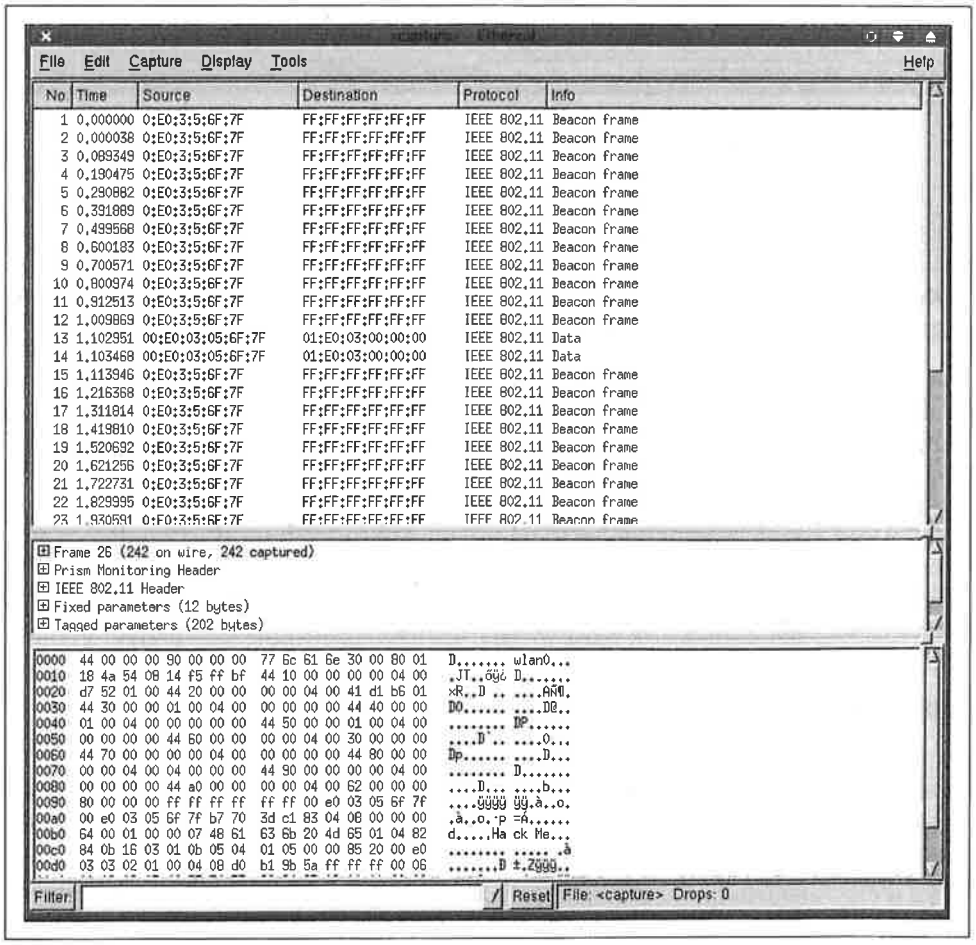

Figure 16-9. Trace of idle 802.11 network

At this point, it is best to get rid of the Beacon frames because they are getting in the way of what we actually want to see. The easiest way to do this is to use the filter expression that excludes Beacon frames: `!(wlan.fc.type==0 and wlan.fc.subtype==8)`. After adding the filter, a few stray Data frames are left; Figure 16-10 shows the remaining Data frames in couplets approximately every three seconds.

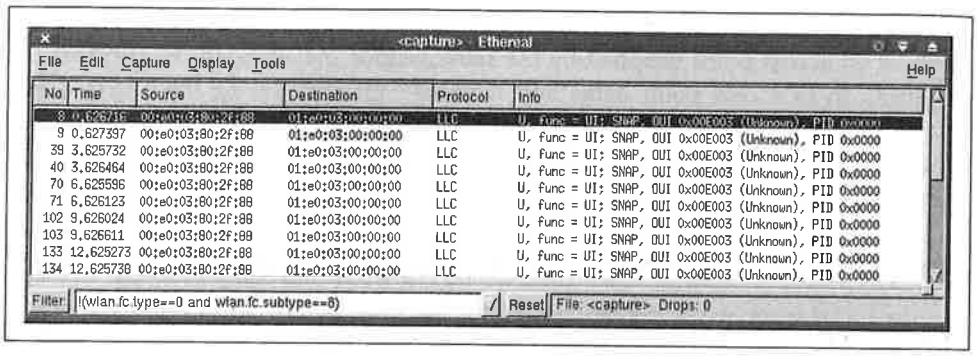

Figure 16-10. The leftover Data frames

With a filter active, the 802.11 Data frames are fully decoded, and the Protocol field is given the value of the protocol dissector for the highest possible layer. In this case, the Protocol field in Ethereal is set to LLC because the data is contained as raw data within an LLC header. After selecting one of the Data frames for further analysis, the tree view displays Figure 16-11.

The data is encapsulated in a raw LLC frame using the Unnumbered Information header. The OUI used is assigned to Nokia Wireless Business Communications, the wireless LAN product unit of Nokia. (The listing of assigned OUIs can be found at *http://standards.ieee.org/regauth/oui/oui.txt*.) In the LLC data, the name of the access point, Outside, is plainly clear.

The other Data frame, shown in Figure 16-12, also encapsulates a blob of data. I can only assume that this Data frame encapsulates workload information, though the data doesn't make obvious which protocol is in use.

Case Study 2: Joining a Network

Joining a network is a more complicated affair than many people ever need to think about. Stations wishing to access network services must locate a desirable network, prove their identity, and connect to the network to start using its services. On a wired network, the joining process seems simple because the location and identification steps are performed by the network administrators. The network has been built out to network jacks throughout the building, so locating a network consists of looking

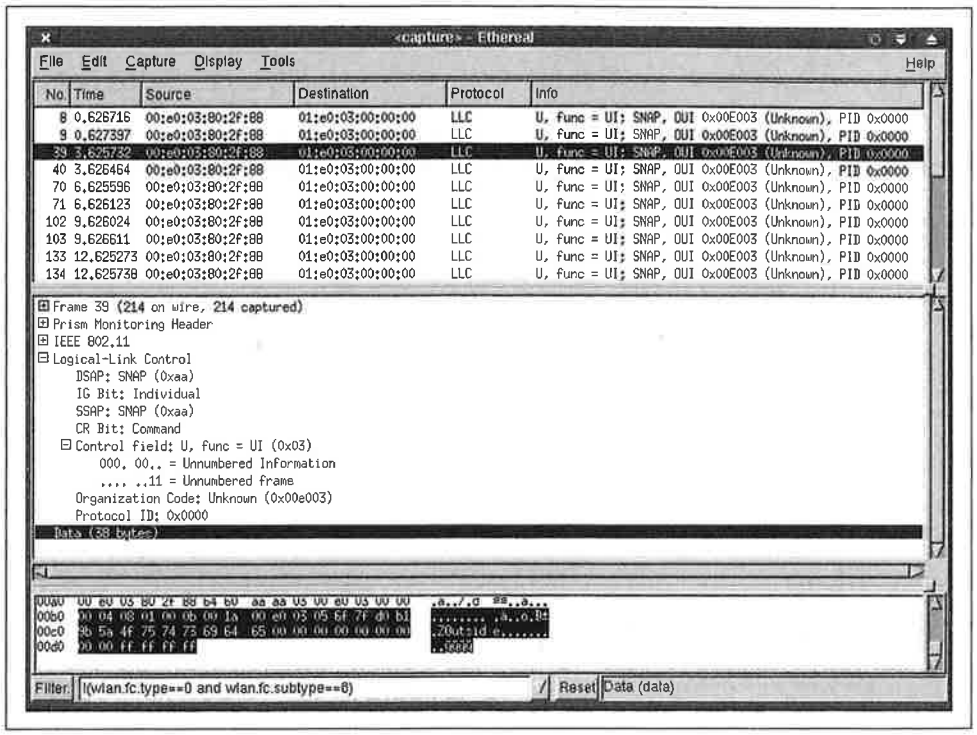

Figure 16-11. One of the random 802.11 Data frames and its LLC data

around for a jack. Proving identity is based on physical access control and personal interaction. Connecting to the network consists of plugging in the cable. Joining an 802.11 network consists of the same steps, but the elimination of the cable means the steps take a different form. To find the network, you must actively scan for other stations to link up with. Authentication cannot be based on physical access control but on cryptographic authentication. Finally, the connection to the network is not based on establishing a physical connection but on the logical connection of an association. The second case study examines the trace of a station joining a network after filtering out extraneous frames to present a clear picture of the exchange.

Scanning

Scanning may be active or passive. Passive scans are carried out by listening to Beacon frames in the area and are not be displayed on a network analyzer because the analyzer cannot tell whether a given Beacon was heard by a wireless station. Active scans rely on Probe Request and Probe Response frames, which are recorded by an analyzer.

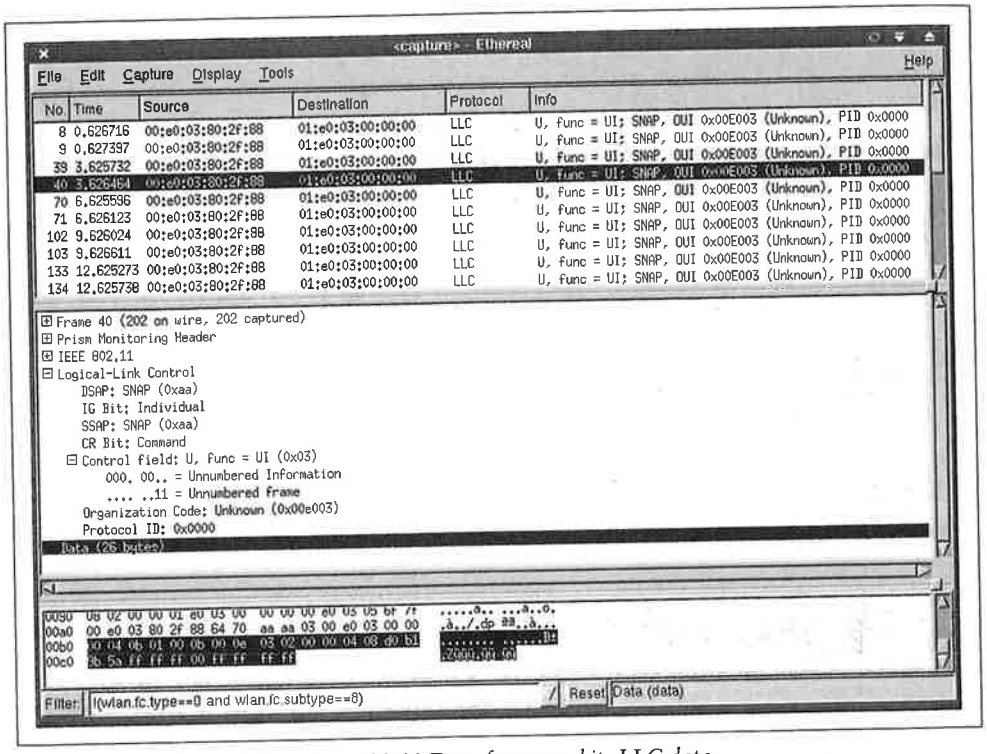

Figure 16-12. The second random 802.11 Data frame and its LLC data

Figure 16-13 shows a Probe Request frame sent by a station seeking access to a network. Probe Request frames carry only two variable information elements in the frame body: the desired SSID and the supported data rates. Probe Requests are broadcast in two senses. First, they are sent to the all-1s destination address `ff:ff:ff:ff:ff:ff`. However, the frame filtering rules in 802.11 would prevent such a frame from being passed to higher protocol layers if the BSSID did not match. Therefore, the frame is also a broadcast in the sense that it is sent to BSSID `ff:ff:ff:ff:ff:ff`. Ethereal has also decoded the information elements to determine that the SSID parameter was set to `Luminiferous Ether`. Probe Requests frequently use a zero-length SSID to indicate that they are willing to join any available network, and any receiving network is responsible for sending a Probe Response. However, this Probe Request is sent specifically to one network. Finally, the Probe Request indicates that the station supports 1-Mbps, 2-Mbps, 5.5-Mbps, and 11-Mbps operation, which is what is expected from an 802.11b card.

In response to the probe request, access points in the SSID named `Luminiferous Ether` should respond with a Probe Response. Indeed, that is the next frame in the trace. Figure 16-14 shows the expanded view of the Probe Response frame in Ethereal's tree

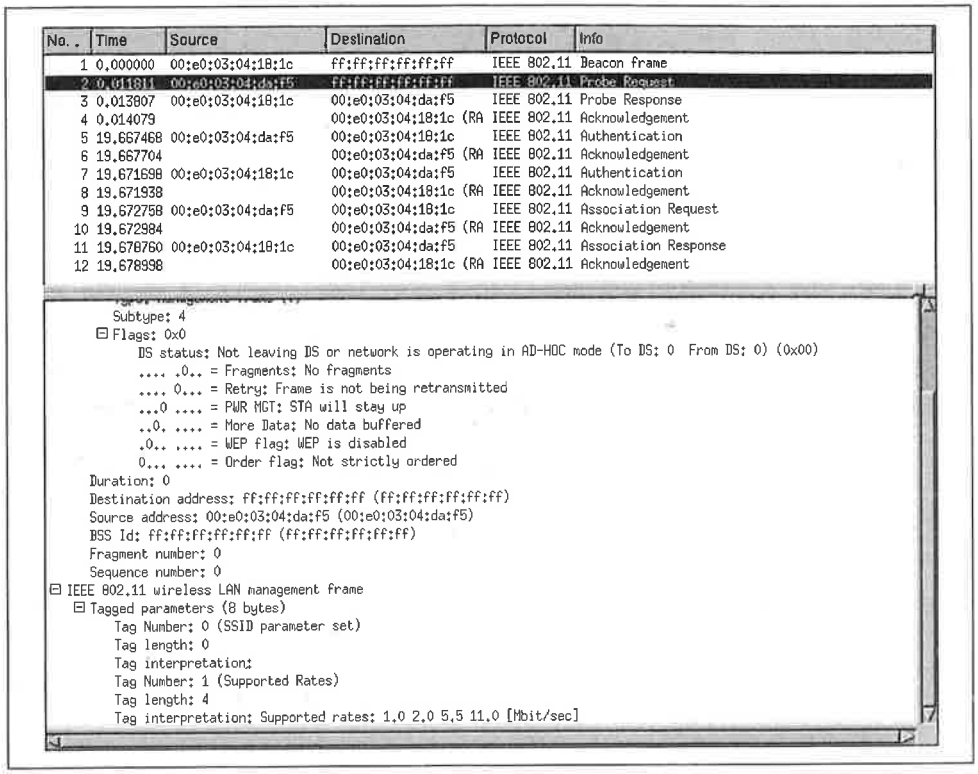

Figure 16-13. Probe Request frame

view pane. Probe Response frames contain three fixed fields. Timestamps are included so that the probing station can synchronize its timer to the access point timer. The Beacon interval is included because it is a basic unit of time for many power-saving operations. It is expressed in time units, though later versions of Ethereal convert it to seconds. The third fixed field is capability information. The access point transmitting the frame does not implement WEP and is not an 802.11b access point, so most of the Capability field flags are set to 0. The CFP capabilities are set to 0 because this access point, like all access points I am aware of, does not implement contention-free service. After the fixed-length fields, the variable-length fields describe the network. Parameters are the same as in the Probe Request frame, with the addition of the DS Parameter Set to identify the current operating channel.

Probe Responses are unicast frames and must be acknowledged by the receiver. Following the Probe Response, the receiver sends an acknowledgment to the access point. No source address is listed because the only address included in an 802.11 acknowledgment frame is the receiver address.

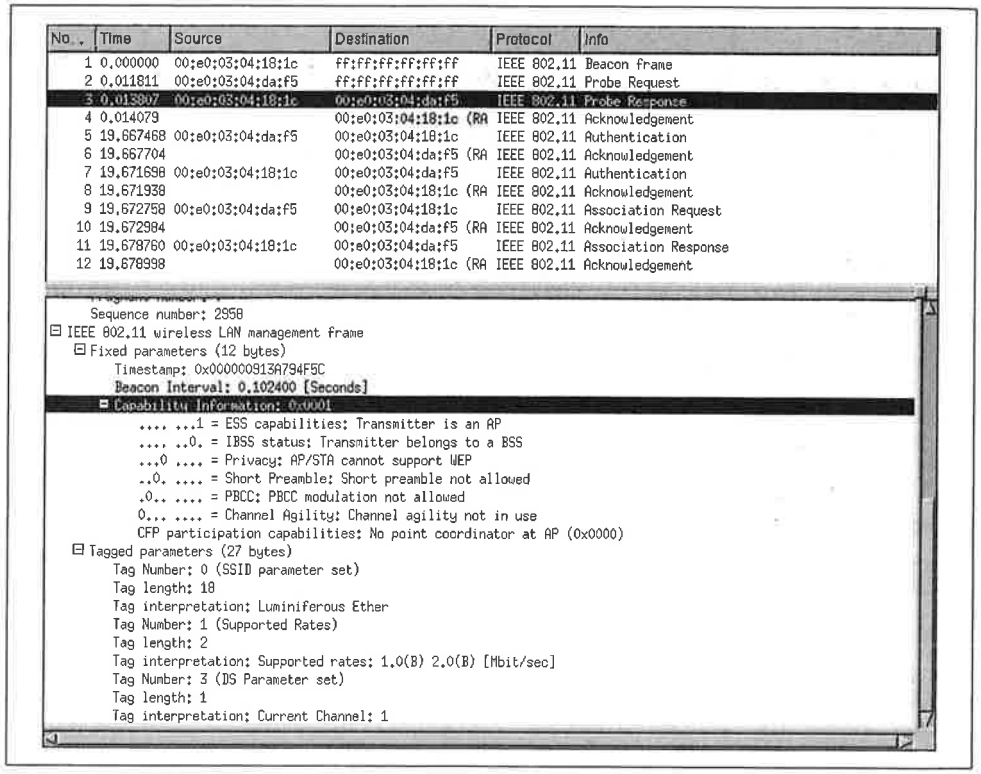

Figure 16-14. Expanded view of Probe Response frame

Authentication

After finding a network, the next step is to authenticate to it. The network in this example is using simple open-system authentication. This greatly simplifies interpreting the trace because open-system authentication requires exchanging only two authentication frames.

First, the station requests authentication with the first frame in the authentication sequence, shown in Figure 16-15. Authentication requests occur only after the station has matched parameters with the network; note that the BSSID now matches the source on the Beacon frames. The first frame specifies authentication algorithm 0 for open system. (Ethereal decodes the field improperly; when highlighted, the data pane clearly shows that the algorithm number is set to 0.) Finally, the status code indicates success because it is too early in the sequence to fail. There is a big time gap between the fourth and fifth frames in the sequence because the driver on the client station was configured to prompt the user to determine which ESS should be joined.

Authentication requests are unicast Management frames and must be acknowledged under the rules of the DCF. The access point sends an acknowledgment immediately after receiving the first Authentication frame. The access point may then choose to

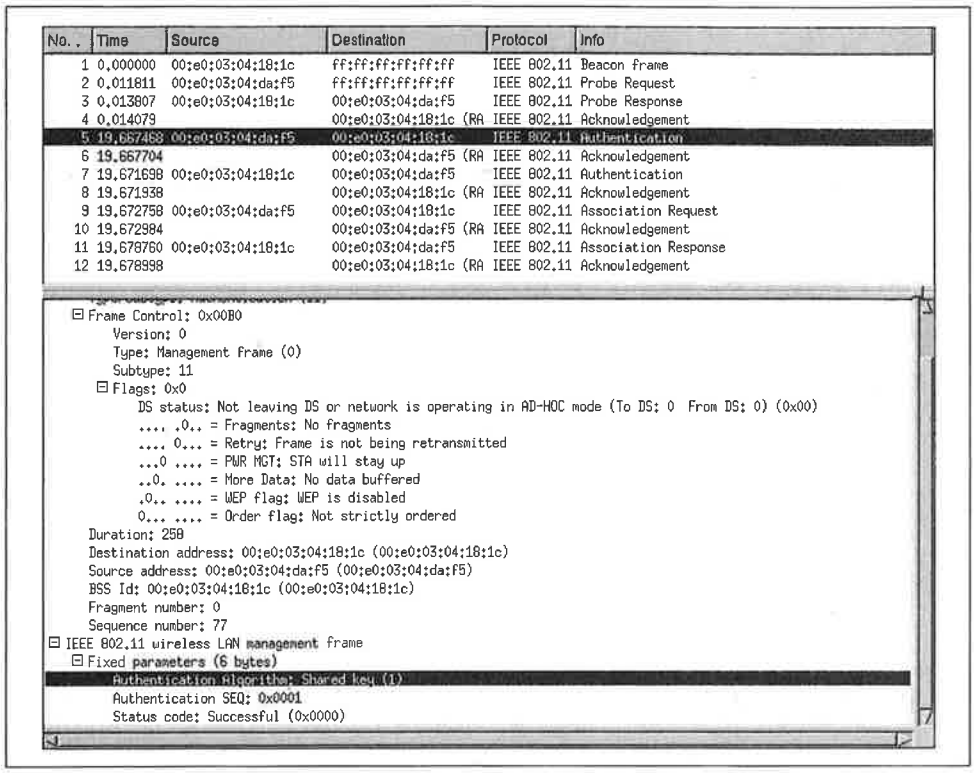

Figure 16-15. First authentication frame

allow or deny the request. Open systems are supposed to accept any authentication request, as this one does (Figure 16-16). The second frame concludes the open-system authentication exchange, and, as expected, the result is successful. Once again, the sequence number is highlighted in the data view pane. Like the previous Management frame, the second Authentication frame is a unicast management frame and must be acknowledged.

Association

After authentication is complete, the station is free to attempt association with the access point. Figure 16-17 shows the frame expanded in the tree view. Most of the parameters in the association request are familiar by this point. Capability information is present, along with the SSID and supported rates of the station.

Association Request frames must be acknowledged. Following the 802.11 acknowledgment, the access point decides whether to allow the association. The main reason for rejecting an association is that a busy access point may not have sufficient resources to support an additional node. In this case, the association was successful.

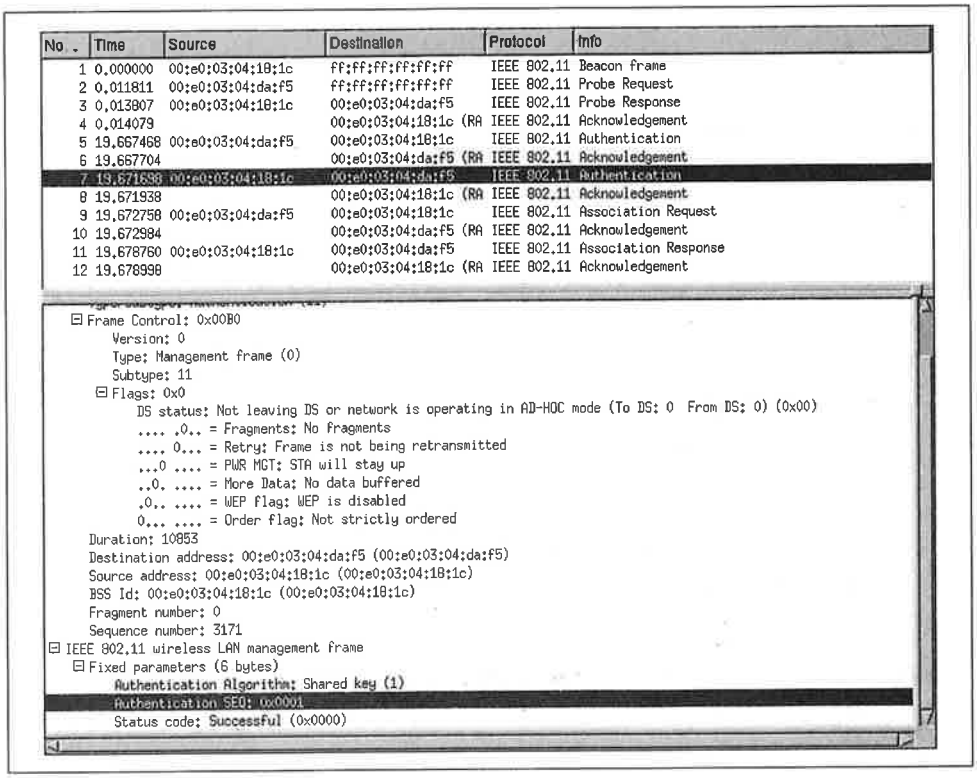

Figure 16-16. Second authentication frame

As part of the response, which is shown decoded in Figure 16-18, the access point assigns an Association ID.

Case Study 3: A Simple Web Transaction

Now we'll look at a higher-level operation: a trace of a machine browsing the Web. If you ask most network engineers how a connection to a web server works, the reply would go something like this:

1. Before it can make an HTTP request, a host must locate the HTTP server by making a DNS request. Because the DNS server is probably not attached to the same IP subnet as the host, the host issues an ARP request to find its default gateway.
2. Once it receives the ARP reply, the host sends a DNS query to resolve the name of the web server (*www.oreilly.com*). It receives a reply with the IP address of the web server (209.204.146.22).
3. The client opens a standard TCP connection to port 80 on the web server and sends an HTTP request for the specified page.

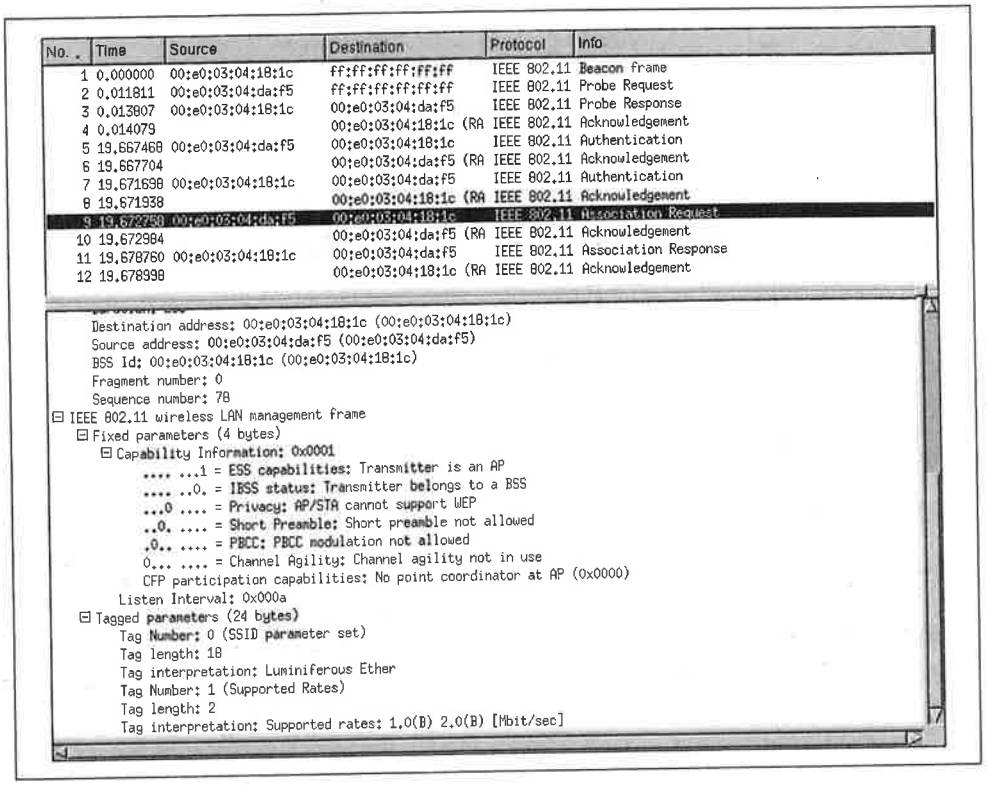

Figure 16-17. Association request

On an 802.11 network, this simple, well-understood process requires 24 frames. Figure 16-19 shows these frames in the packet summary pane. A display filter has been applied to remove SSH traffic that was in the air at the same time, as well as the vendor-specific access point name advertisements using the raw LLC encapsulation. References to frame numbers throughout this section use the frame numbers from Figure 16-19.

The ARP request

The first step in the process is an ARP request to get the MAC address of the default router, 192.168.200.1. ARP requests are normally broadcast to the local network. On a wireless network, though, different procedures apply. To start with, the ARP request takes more than one frame on the wireless network. The first frame, shown in Figure 16-20, kicks off the exchange. The 802.11 header has several expected fields. The frame is a Data frame because it is carrying a higher-layer packet. It is bound for the distribution system in the AP, so the ToDS bit is set, and the FromDS bit is clear. Like all ARP requests, it is sent to the broadcast address. However, the BSSID keeps the broadcast from being replicated to wireless stations attached to other BSSs in the area.

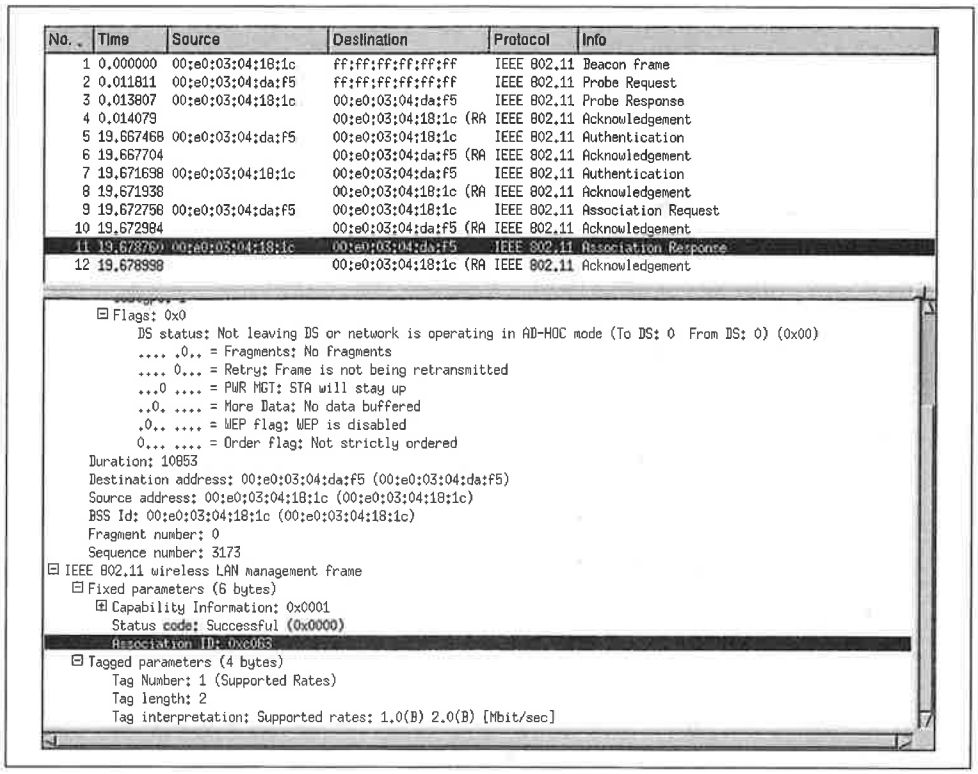

Figure 16-18. Association response

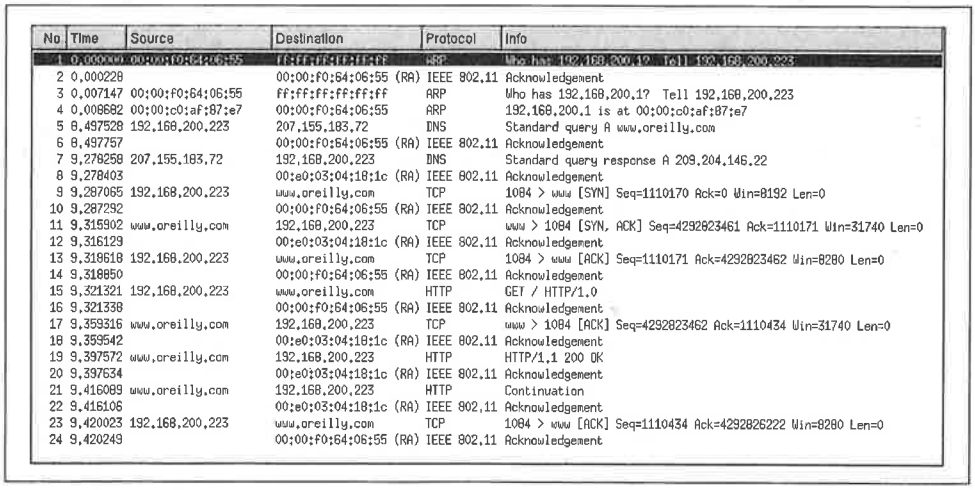

Figure 16-19. Full web site trace

Although the frame is destined for a broadcast address, the wireless LAN station has no way of sending a broadcast directly onto the wired network. The access point converts the frame into a broadcast on the wired network. Like any packet on the

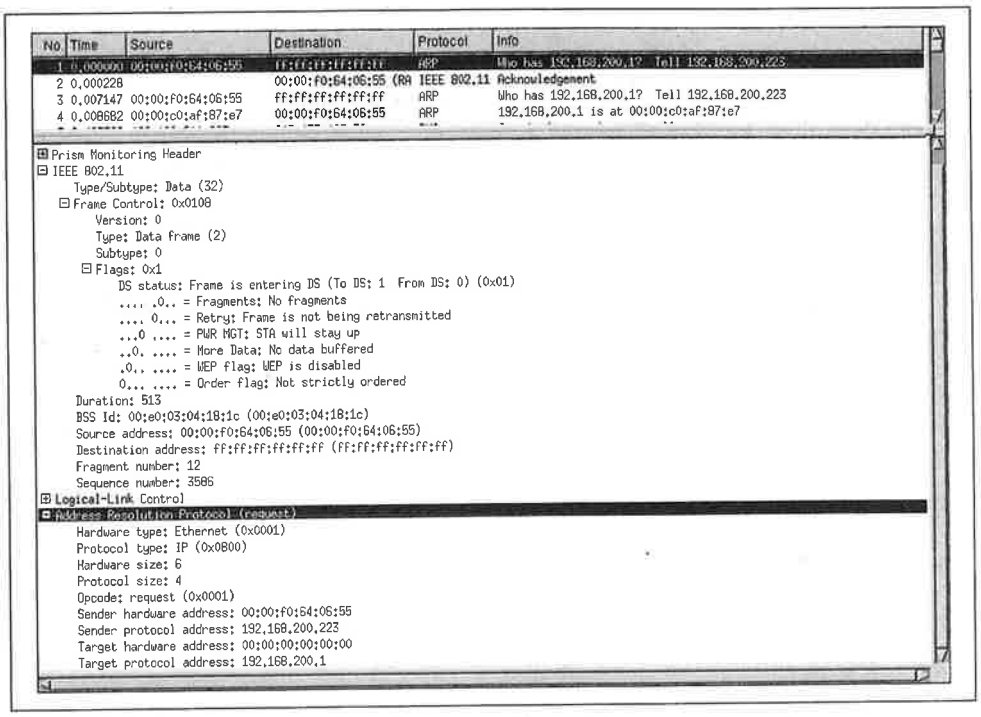

Figure 16-20. Initial ARP request

wired network, the access point checks to see whether the broadcast must be relayed to the wireless network. Frame 3 is the ARP request after it is processed by the access point. Note that the FromDS bit is set to indicate that the frame originated on the distribution system. In this case, the frame originated on the wired network (the distribution system medium).

The ARP reply

Once the frame reaches the wired network, the default router can reply. The ARP reply of Frame 4 is shown in Figure 16-21. The Reply frame originates from the wired network, so the FromDS bit is set. The frame retains its source address from the wired network.

The DNS request

The DNS request of Frame 5 is shown in Figure 16-22. It is sent using the same BSSID used throughout this example, with a source address of the wireless LAN station and a destination of the default router. The frame originates on the wireless network and must be bridged to the wired side, so the ToDS bit is set. Ethereal's tree view shows summary decodes on the source and destination IP addresses and UDP ports. Finally, the DNS decode is called, which shows the request for the address of *www.oreilly.com*.

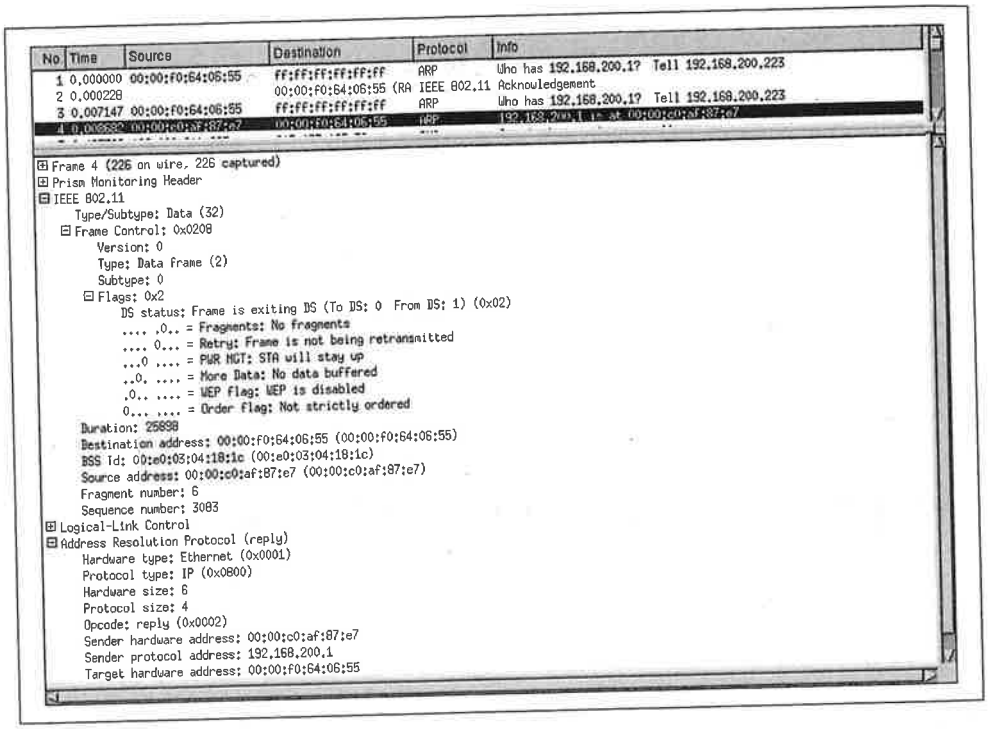

Figure 16-21. ARP reply

Frame 6 is an 802.11 acknowledgment of the DNS request. It is a Control frame of type Acknowledgment. Acknowledgment frames are extraordinarily simple, containing only the address of the sender of the previous frame. It is shown in Figure 16-23.

The DNS reply

The DNS system queried in the previous step responds with Frame 7, which is shown in Figure 16-24. The frame comes from the distribution system, so the FromDS bit is set. The source MAC address of the frame is the default router and is transmitted using the BSSID of the network. Like all unicast Data frames, the DNS reply must be acknowledged by the 802.11 MAC layer. Frame 8 is the required acknowledgment.

The TCP three-way handshake

The TCP three-way handshake is shown in the packet view pane in Figure 16-25. Note that each TCP segment involved is embedded in a unicast 802.11 data and must be acknowledged, so the exchange requires six frames.

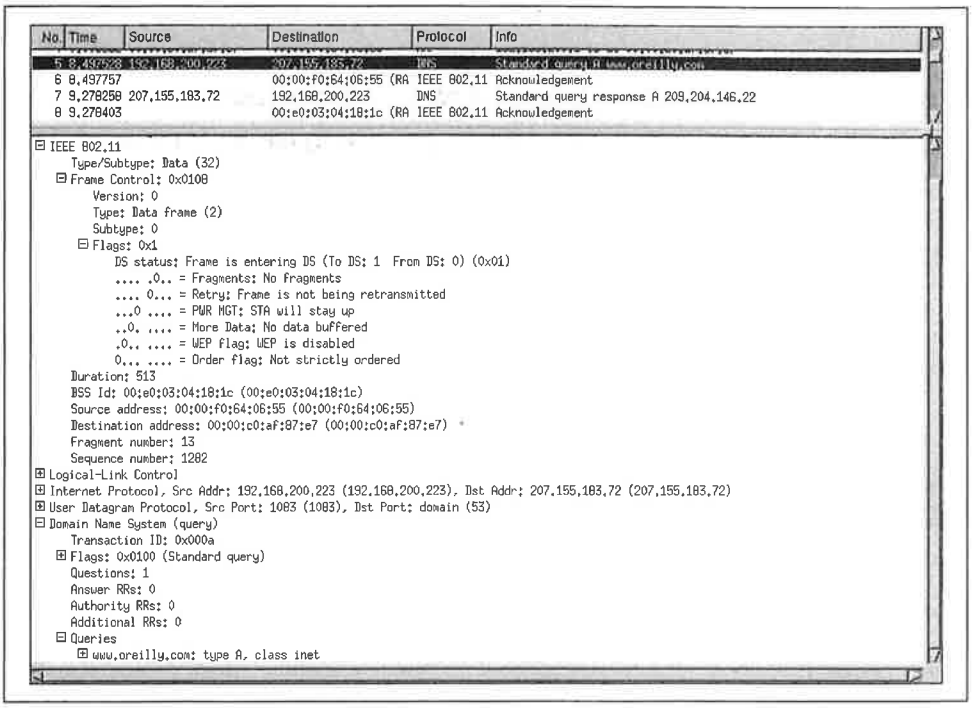

Figure 16-22. DNS request

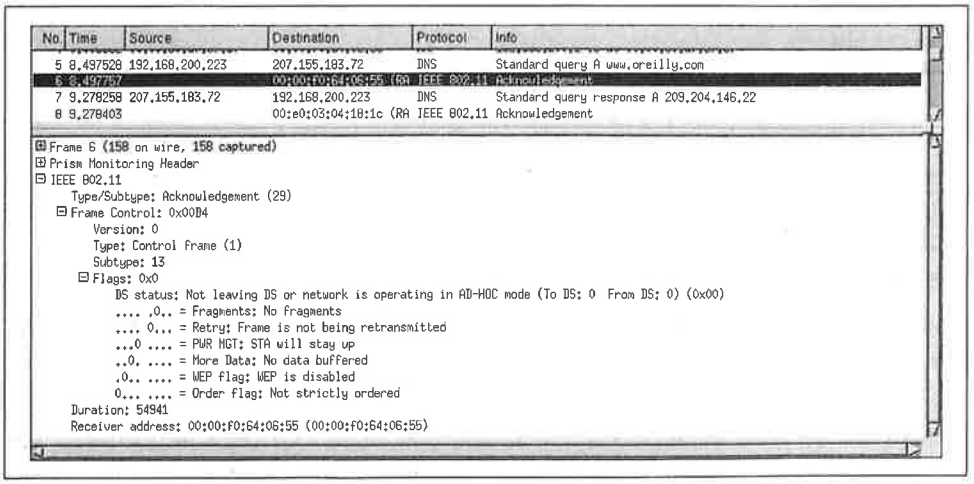

Figure 16-23. ACK of DNS request

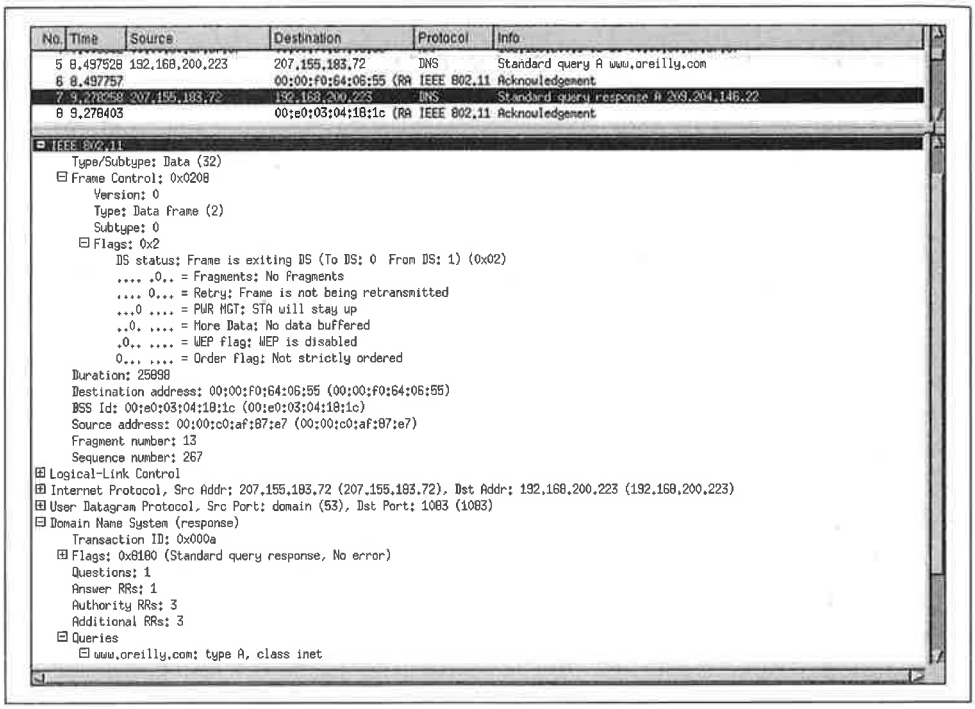

Figure 16-24. DNS reply

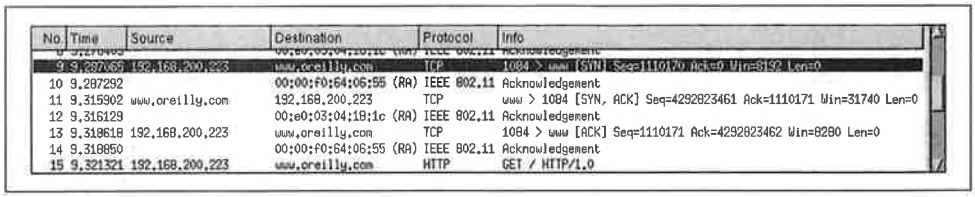

Figure 16-25. TCP three-way handshake

The TCP data transfer

The HTTP connection itself lasts much longer than everything we've seen so far, but everything past the three-way TCP handshake is boring. Figure 16-26 illustrates the exchange, which consists of a series of packets in the following pattern:

1. An 802.11 Data frame containing a maximum-sized TCP segment from the web server, carrying data from TCP port 80 on the server to the random high-numbered port chosen by the client. The frame's source address is the default router, but its transmitter address is the access point.
2. An 802.11 acknowledgment from the client to the access point to acknowledge receipt of the frame in step 1.

3. An 802.11 Data frame carrying a TCP acknowledgment from the client to the server acknowledges the receipt of the data in the frame in step 1. However, the acknowledgment in this step is a TCP segment, which is higher-layer data to the MAC layer. The source MAC address of this frame is the client, and the destination MAC address is the default router. However, the default router is on a wired Ethernet, and an access point is required for bridging, so the receiver address is the MAC address of the wireless interface in the access point.
4. The access point sends an 802.11 acknowledgment for the frame in step 3 to the client.

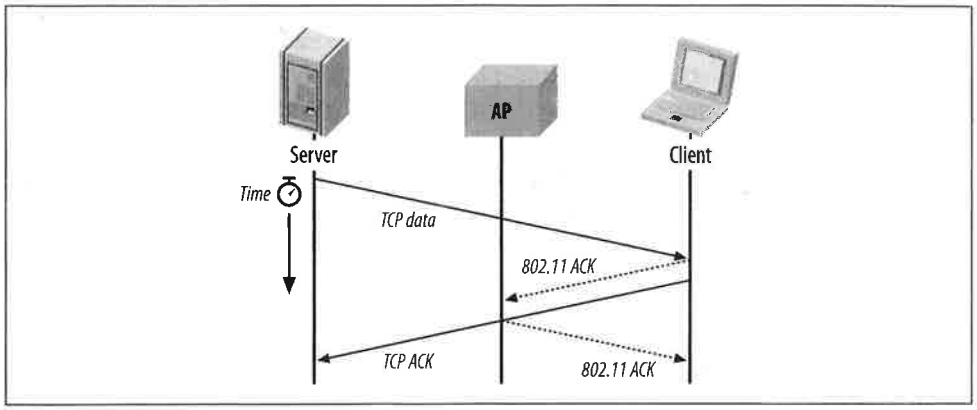

Figure 16-26. Frames corresponding to the bulk transfer of an HTTP connection

AirSnort

Technically, AirSnort is not a network analysis tool. It is a tool that every wireless network administrator should be familiar with, though, because it allows a malicious attacker to retrieve the WEP keys from an active network simply by collecting enough traffic. AirSnort was released with some fanfare in August 2001 as the first public implementation of the Fluhrer/Mantin/Shamir attack against WEP.

Prerequisites

Before attempting to configure AirSnort, you need to have a configured kernel source tree, a configured PCMCIA source tree, and a compiled *linux-wlan-ng* driver capable of raw packet captures. If you have already built Ethereal for use with a PRISM-2 card, all the prerequisites should be met. Download and unpack the code from *http://airsnort.sourceforge.net*; the current version as this chapter was written was 0.0.9. Unlike Adam Stubblefield's attack, the code for AirSnort is publicly available under the GPL.

AirSnort can be used to recover either standard 40-bit keys or what it refers to as 128-bit keys. However, as noted in Chapter 5, the label of 128-bit WEP is a slight misnomer because it refers to the size of the RC4 key. After subtracting the 24-bit initialization vector, only 104 bits are secret. To be accurate, then, AirSnort can be used to recover 40-bit secret keys and 104-bit secret keys, which correspond to the two most common WEP implementations currently on the market. Some vendors have WEP implementations that use longer keys; for example, Nokia uses 128 secret bits in addition to the 24-bit IV, and this vendor's long key-length WEP cannot currently be attacked with AirSnort. However, the longer key length doesn't provide any real protection against this attack. If the history of open source development is a guide, future versions of AirSnort will be able to attack longer keys.

Compiling

AirSnort is written in C++. Some Linux distributions do not install a C++ compiler by default, so you may need to fetch the C++ compiler package before proceeding. Simply run *make* in the top-level directory to build the package:

```
$ make
make -C src
make[1]: Entering directory `/home/gast/wlan/airsnort-0.0.9/src'
g++  -g  -c -o capture.o capture.cc
g++  -g  -c -o PacketSource.o PacketSource.cc
g++ -o capture capture.o PacketSource.o  -lncurses
g++  -g  -c -o crack.o crack.cc
g++  -g  -c -o RC4.o RC4.cc
g++  -g  -c -o crc-32.o crc-32.cc
g++ -o crack crack.o RC4.o crc-32.o
g++  -g  -c -o gencases.o gencases.cc
g++ -o gencases gencases.o RC4.o crc-32.o
g++  -g  -c -o decrypt.o decrypt.cc
g++  -g  -c -o utils.o utils.cc
g++ -o decrypt decrypt.o RC4.o crc-32.o utils.o
make[1]: Leaving directory `/home/gast/wlan/airsnort-0.0.9/src'
```

The final product of the compilation is two executable files:

capture
: A program that interfaces with low-level device drivers to capture raw packets as they come in. The sole argument to *capture* is the filename used to store "interesting" packets for later processing.

crack
: A program that takes the interesting packets saved by *capture* and runs them through the cryptographic attack.

Running AirSnort

AirSnort depends on promiscuous capture, so start by using *wlanctl-ng* to enable promiscuous mode capture on the channel you want to monitor:

```
# wlanctl-ng wlan0 lnxreq_wlansniff channel=5 enable=true
message=lnxreq_wlansniff
  enable=true
  channel=5
  resultcode=success
```

Next, fire up AirSnort's *capture* program. *capture* grabs all the incoming frames and looks for frames with a weak initialization vector. Those frames are saved for further analysis. *capture* takes one argument: the name of the file for saving captured frames. *capture* always appends data to the saved file, so a file can collect the traces from several sniffing sessions before attempting analysis. The *–c* option displays statistics on the number of packets captured, the number of encrypted packets analyzed, and the number of weak IVs saved. It also displays the last IV received, which allows you to check for vulnerability to the "IV increments from zero" design flaw identified by the Berkeley team. *capture* won't run if the driver is not in sniffing mode.

```
$ ./capture -c 40bits.1

 AirSnort Capture v0.0.9
 Copyright 2001, Jeremy Bruestle and Blake Hegerle

 Total Packets:       70550
 Encrypted Packets:   22931
 Interesting Packets: 5
 Last IV = 8f:1b:1e
```

To attempt key recovery, run the *crack* program against the capture output file. *crack* takes one argument, which is the name of the capture file. There are two options. The *–b* option controls the *breadth* of the analysis and ranges from 1 to 5. Wider breadths take more CPU time but may be able to guess the key earlier. The second option is the key length, specified as *–l*. *crack* can attempt to recover 40-bit keys and 128-bit keys (with 104 secret bits), which are specified with *–l 40* and *–l 128*, respectively. Note that higher breadths can require vastly more CPU time at long key lengths. As a practical matter, all breadths are fine for 40-bit keys. For 128-bit keys, a breadth of 2 requires about 30 seconds of run time, and a breadth of 3 takes so long to complete, it's impractical.

When *crack* is unsuccessful, it simply prints the number of samples received. Each IV is associated with the key byte it is used to attack. *crack* categorizes the samples against each key byte; the authors of AirSnort suggest that approximately 115 samples are needed for each key byte to mount a successful key recovery:

```
[gast@bloodhound airsnort-0.0.9]$ time ./crack -b 5 40bits.1
Reading packets
```

```
Performing crack, keySize=40 bit, breadth=5
Key Byte 0: 8 samples
Key Byte 1: 12 samples
Key Byte 2: 9 samples
Key Byte 3: 8 samples
Key Byte 4: 11 samples
Check samples: 10

FAILED! r=1
0.29user 0.14system 0:00.42elapsed 100%CPU (0avgtext+0avgdata 0maxresident)k
0inputs+0outputs (152major+16minor)pagefaults 0swaps
```

When a *crack* run is successful, it prints the key at the end of the run. Some vendors simply take an ASCII string and use the ASCII representation for the WEP key, so *crack* prints both the hexadecimal and textual representations of the key string:

```
[gast@bloodhound airsnort-0.0.9]$ time ./crack -b 5 40bits.1

Reading packets
Performing crack, keySize=40 bit, breadth=5
Key Byte 0: 143 samples
Key Byte 1: 149 samples
Key Byte 2: 132 samples
Key Byte 3: 127 samples
Key Byte 4: 128 samples
Check samples: 10

GOT KEY!
Hex = 9c:3b:46:20:97
String = '.;F .'
0.00user 0.01system 0:00.01elapsed 62%CPU (0avgtext+0avgdata 0maxresident)k
0inputs+0outputs (153major+21minor)pagefaults 0swaps
```

(Although it is probably meaningless to do so, I have changed the key on my wireless LAN from 9c:3b:46:20:97.)

Key Recovery Time Estimates

There are two components to recovering a key. First, enough frames with weak IVs must be gathered to mount an attack, which I refer to as the *gathering time*. Second, a successful attack must be run against the stored frames, which I refer to as the *analysis time*.

In my experience, the time required to gather enough data to mount the attack dominates the CPU time required to run the attack. With enough samples to successfully attack, the analysis time is only a few seconds. The analysis time scales linearly, so the protection afforded by longer keys is only a few seconds. By doubling the key length, the CPU time required for the attack will double, but doubling a few seconds is still only a few seconds.

In my real-world trials against my home network, only a few hours of gathering time were required, and longer keys did not dramatically increase the gathering time. While longer keys require more weak IVs, more IVs are weak at longer key lengths. WEP IVs are three bytes long and are often written in the byte-delimited, colon-separated format commonly used for MAC addresses. Weak IVs have a middle byte that is all 1s; that is, they can be written in the form B+3:FF:N, in which B and N may take on different values. In the Fluhrer/Mantin/Shamir attack, N may be any value from 0 to 255 and is not constrained. Each weak key helps recover a particular byte in the secret portion of the key. B is the byte number, starting from 0, of the secret portion of the key. Thus, a weak IV of the form 4:FF:N helps recover key byte 1. The number of weak IVs is the product of the key length in bytes multiplied by 256; Table 16-3 shows how the fraction of weak keys increases with key size. As the fraction of weak keys increases, the requisite number of weak keys may be gathered more quickly.

Table 16-3. Number of weak packets as a function of key length

Secret key length	Values of B+3 in weak IV (B+3:FF:N)	Number of weak IVs	Fraction of IV space
40 bits	3 <= B+3 < 8 (0 <= B < 5)	1,280	0.008%
104 bits	3 <= B+3 < 16 (0 <= B < 13)	3,328	0.020%
128 bits	3 <= B+3 < 19 (0 <= B < 16)	4,096	0.024%

Two opposing forces work on the gathering time. Longer keys require more samples of weak IVs but also expose more IVs as weak. In my experience, these two effects offset for the most part, though longer keys may slightly increase the gathering time. In theory, the two effects should offset each other. In fact, the number of weak IVs that need to be collected is directly proportional to the key size. In my testing, about 100 weak IVs per key byte were necessary before the key recover could be successful. The fraction of weak IVs is also a direct function of the key size in bytes, so there is reason to believe that the gathering time is independent of key size.

In my experimental runs, I generated a 30% load on an 11-Mbps 802.11b network. I was always able to recover keys in eight hours, though I could occasionally recover shorter keys in as few as five. Many production wireless LANs may run at higher loads, which would certainly decrease the required gathering time. If the network load was 60%, which would not be outrageous for an 11-Mbps shared medium, keys could easily be recovered within a single working day. AirSnort conclusively demonstrates that any information flowing over a wireless LAN and protected with standard WEP should be treated as fully exposed.

CHAPTER 17

802.11 Performance Tuning

Until now, wireless network administrators have probably received a bit of a free ride. Wireless is new and cool, and people do not know what sort of service they should expect. Users are happy that it works at all, and it is both easy and correct to tell them that they should not expect the same performance they would see on a 100BaseT Ethernet. Most wireless installations do not have large user communities and therefore do not have dozens or hundreds of stations trying to associate with a small number of access points. Furthermore, most wireless networks are logically subordinate to existing wired networks. 802.11 was designed to complement existing LANs, not replace them. When the wired LAN is the primary network, people can still get the job done without the wireless network, and it is seen as less critical. Most likely, your biggest problems are positioning your access points so you have coverage everywhere you want it, installing drivers, and keeping your WEP keys up to date.

However, networks have a way of growing, and users have a way of becoming more demanding. Your network's performance "out of the box" is probably fairly poor, even if no one but you notices. Changing the physical environment (by experimenting with access point placement, external antennas, etc.) may alleviate some problems, but others may best be resolved by tuning administrative parameters. This chapter discusses some of the administrative parameters that can tuned to improve the behavior of your wireless network.

Tuning Radio Management

As with other types of wireless networks, the most precious resource on an 802.11 network is radio bandwidth. Radio spectrum is constrained by regulatory authority and cannot be easily enlarged. Several parameters allow you to optimize your network's use of the radio resource.

Beacon Interval

Beacon frames serve several fundamental purposes in an infrastructure network. At the most basic level, Beacon frames define the coverage area of a basic service set (BSS). All communication in infrastructure networks is through an access point, even if the frame is sent between two stations in the same BSS. Access points are stationary, which means that the distance a Beacon frame can travel reliably won't vary over time.* Stations monitor Beacon frames to determine which Extended Service Sets (ESSs) offer coverage at a their physical location and use the received signal strength to monitor the signal quality.

Transmitting Beacon frames, however, eats up radio capacity. Decreasing the Beacon interval makes passive scanning more reliable and faster because Beacon frames announce the network to the radio link more frequently. Smaller Beacon intervals may also make mobility more effective by increasing the coverage information available to mobile nodes. Rapidly moving nodes benefit from more frequent Beacon frames because they can update signal strength information more often.† Increasing the Beacon interval indirectly increases the power-saving capability of attached nodes by altering the listen interval and the DTIM interval, both of which are discussed in the section "Tuning Power Management." Increasing the Beacon interval may add an incremental amount of throughput by decreasing contention for the medium. Time occupied by Beacon frames is time that can't be used for transmitting data.

RTS Threshold

802.11 includes the RTS/CTS clearing procedure to help with large frames. Any frame larger than the RTS threshold must be cleared for departure from the antenna by transmission of an RTS and reception of a CTS from the target. RTS/CTS exists to combat interference from so-called hidden nodes. The RTS/CTS exchange minimizes interference from hidden nodes by informing all stations in the immediate area that a frame exchange is about to take place. The standard specifies that the RTS threshold should be set to 2,347 bytes. If network throughput is slow or there are high numbers of frame retransmissions, enable RTS clearing by decreasing the RTS threshold.

In Chapter 3, I said that a hidden node was a node that wasn't visible to all the stations on the network. Under what sorts of situations can you expect hidden nodes? Just about any, really. In almost any network, there are bound to be places where two nodes can reach the access point but not each other. Let's consider the simplest

* Multipath interference may cause odd time-dependent interference patterns. A particular spot may be within range of an access point at one instant in time and subject to multipath fading seconds later. However, such a spot has marginal coverage and should not be considered a part of a basic service area.

† 802.11 is not designed to support high-speed mobility, though. Cellular-based, wide-area technologies are more effective.

network imaginable: one access point in the middle of a large field with nothing to cause reflections or otherwise obstruct the signal. Take one mobile station, start at the access point, and move east until the signal degrades so that communication is just barely possible. Now take another station and move west. Both stations can communicate with the access point, but they are invisible to each other.

The previous thought experiment should convince you that invisible nodes are a fact of life. I would expect invisible nodes to be common in buildings with a great deal of metal in the walls and floors, lots of surfaces capable of reflecting radio waves, and lots of noise sources. In general, the harsher the radio environment, the greater the probability that a significant number of nodes are invisible to each other. All environments have hidden nodes. The question is how many there are and how many collisions result. Hidden nodes are likely to be more of a problem on highly populated networks, where more stations have the opportunity to transmit and cause unwanted collisions, and on busy networks, where there are more network communications that can be interrupted by unintended collisions.

Fragmentation Threshold

MAC-layer fragmentation is controlled by the fragmentation threshold variable. Any frames longer than the fragmentation threshold are sliced into smaller units for transmission. The default fragmentation threshold is the smaller of 2,346 or the maximum MAC frame length permitted by the physical layer. However, the RF-based physical layers usually have a maximum MAC frame length of 4,096 bytes, so this parameter generally defaults to 2,346. The common value immediately implies that fragmentation and RTS/CTS clearing are often used in tandem.

In environments with severe interference, encouraging fragmentation by decreasing this threshold may improve the effective throughput. When single fragments are lost, only the lost fragment must be retransmitted. By definition, the lost fragment is shorter than the entire frame and thus takes a shorter amount of time to transmit. Setting this threshold is a delicate balancing act. If it is decreased too much, the effective throughput falls because of the additional time required to acknowledge each fragment. Likewise, setting this parameter too high may decrease effective throughput by allowing large frames to be corrupted, thus increasing the retransmission load on the radio channel.

Retry Limits

Every station in a network has two retry limits associated with it. A retry limit is the number of times a station will attempt to retransmit a frame before discarding it. The *long retry limit*, which applies to frames longer than the RTS threshold, is set to 4 by default. A frame requiring RTS/CTS clearing is retransmitted four times before it is

discarded and reported to higher-level protocols. The *short retry limit*, which applies to frames shorter than the RTS threshold, is set to 7 by default.

Decreasing the retry limit reduces the necessary buffer space on the local system. If frames expire quicker, expired frames can be discarded, and the memory can be reclaimed quicker. Increasing the retry limits may decrease throughput due to interactions with higher-layer protocols. When TCP segments are lost, well-behaved TCP implementations perform a slow start. Longer retry limits may increase the amount of time it takes to declare a segment lost.

Tuning Power Management

From the outset, 802.11 was designed for mobile devices. To be useful, though, mobile devices cannot be constrained by a power cord, so they usually rely on an internal battery. 802.11 includes a number of parameters that allow stations to save power, though power saving is accomplished at the expense of the throughput or latency to the station.

Listen Interval

When stations associate with an access point, one of the parameters specified is the listen interval, which is the number of Beacon intervals between instances when the station wakes up to received buffered traffic. Longer listen intervals enable a station to power down the transceiver for long periods. Long power-downs save a great deal of power and can dramatically extend battery life. Each station may have its own listen interval.

Lengthening the listen interval has two drawbacks. Access points must buffer frames for sleeping stations, so a long listen interval may require more packet buffer space on the access point. Large numbers of clients with long listen intervals may overwhelm the limited buffer space in access point hardware. Second, increasing the listen Interval delays frame delivery. If a station is sleeping when its access point receives a frame, the frame must be buffered until the sleeping station is awake. After powering up, the station must receive a Beacon frame advertising the buffered frame and send a PS-Poll to retrieve the frame. This buffering and retrieval process can delay the time the frame spends in transit. Whether this is acceptable depends on the traffic requirements. For asynchronous communications such as email, lengthening the listen Interval isn't likely to be a problem. But in other applications that require synchronous, time-sensitive communications (such as securities market data feeds today or an IP phone with an 802.11 interface in the future), a longer interval might not be acceptable. Certain applications may also have trouble with the increased latency. Database applications, in particular, are significantly affected by increased latency. A task group is working on MAC enhancements to provide quality of service for transmissions on 802.11 networks, but no standard has emerged yet.

DTIM Period

The DTIM period is a parameter associated with an infrastructure network, shared by all nodes associated with an access point. It is configured by the access point administrator and advertised in Beacon frames. All Beacon frames include a traffic indication map (TIM) to describe any buffered frames. Unicast frames buffered for individual stations are delivered in response to a query from the station. This polled approach is not suitable for multicast and broadcast frames, though, because it takes too much capacity to transmit multicast and broadcast frames multiple times. Instead of the polled approach, broadcast and multicast frames are delivered after every Delivery TIM (DTIM).

Changing the DTIM has the same effect as changing the listen Interval. (That should not be a surprise, given that the DTIM acts like the listen Interval for broadcast and multicast frames.) Increasing the DTIM allows mobile stations to conserve power more effectively at the cost of buffer space in the access point and delays in the reception. Before increasing the DTIM, be sure that all applications can handle the increased delay and that broadcasts and multicasts are not used to distribute data to all stations synchronously. If the application uses broadcast or multicast frames to ensure that all mobile stations receive the same blob of data simultaneously, as would be the case with a real-time data feed, increasing the DTIM will likely have adverse effects.

ATIM Window

In an infrastructure network, access points provide most of the power-saving support functions. In an independent or ad hoc 802.11 network, many of those functions move into the network interface driver. In ad hoc networks, stations are required to power up for every Beacon transmission and remain powered up for the duration of the Announcement TIM (ATIM) window, which is measured in time units (TUs).

Decreasing the ATIM window increases the power savings because the required power-on time for the mobile stations is reduced. Stations can power down quickly and are not required to be active during a large fraction of the time between Beacons. Increasing the ATIM window increases the probability a power-saving station will be awake when a second station has a frame. Service quality is increased, and the required buffer space is potentially smaller.

Decreasing or disabling the ATIM window would probably have the same effect on synchronous or real-time applications as increasing the DTIM timer on an infrastructure network—that is, it is likely to cause problems with less reliable communications or applications that depend on real-time data. One of the most obvious examples of a real-time application of ad hoc networking is gaming, but it is far more

likely that ad hoc gaming networks would be tuned for low delay and high throughput than for low-power operation.

Timing Operations

Timing is a key component of 802.11 network operations. Several management operations require multistep processes, and each has its own timer.

Scan Timing

To determine which network to join, a station must first scan for available networks. Some products expose timers to allow customization of the scanning process. In products that expose timers, both an *active scan timer* and a *passive scan timer* may be exposed. The active timer is the amount of time, in TUs, that a station waits after sending a Probe Request frame to solicit an active response from access points in the area. Passive scanning is simply listening for Beacon frames and can take place on several radio channels; the passive scan timer specifies the amount of time the receiver spends listening on each channel before switching to the next.

Timers Related to Joining the Network

Once a station has located an infrastructure network to join, it authenticates to an access point and associates with it. Each of these operations has a timeout associated with it. The *authentication timeout* is reset at each stage of the authentication process; if any step of the process exceeds the timeout, authentication fails. On busy networks, the timeout may need to be increased. The *association timeout* serves a similar function in the association process.

Dwell Time (Frequency-Hopping Networks Only)

The amount of time that an FH PHY spends on a single hop channel is called the dwell time. It is set by local regulatory authorities and is generally not tunable, except by changing the network card driver to a different regulatory domain.

Physical Operations

Most wireless LAN hardware on the market includes antenna diversity and user-selectable power levels, though neither option is strictly required by the specification. Some products implementing these features offer configuration options to control them.

Antenna Diversity

Multiple antennas are a common way of combating multipath fading. If the signal level at one antenna is bad due to multipath effects, a second antenna a short distance away may not be subject to the same fade. Some products that implement antenna diversity allow it to be disabled, though there is little reason to do so in practice.

Transmit Power Levels

All common 802.11 products implement multiple transmission power levels. Boosting the power can overcome fading because the power does not fall below the detection threshold. Power level selection may not be available in all products and may depend on the country in which you use the product; the transmitted power must comply with local regulatory requirements. Consult your manufacturer's documentation or support organization for details.

Lower power can be used to intentionally increase the access pont density. Increasing the access point density can increase the aggregate wireless network throughput in a given area by shrinking the size of the BSSs providing coverage and allowing for more BSSs. Depending on the number of access points required by 802.11a equipment, reducing the transmit power on access points might also be useful in reducing the coverage area as part of a move to 802.11a.

Summary of Tunable Parameters

For quick reference, Table 17-1 summarizes the contents of this chapter, including the effect of changing each of the tuning parameters.

Table 17-1. Summary of common tunable parameters

Parameter	Meaning and units	Effect when decreased	Effect when increased
Beacon Interval	Number of TUs between transmission of Beacon frames.	Passive scans complete more quickly, and mobile stations may be able to move more rapidly while maintaining network connectivity.	Small increase in available radio capacity and throughput and increased battery life.
RTS Threshold	Frames larger than the threshold are preceded by RTS/CTS exchange.	Greater effective throughput if there are a large number of hidden node situations .	Maximum theoretical throughput is increased, but an improvement will be realized only if there is no interference.
Fragmenttion Threshold	Frames larger than the threshold are transmitted using the fragmentation procedure.	Interference corrupts only fragments, not whole frames, so effective throughput may increase.	Increases throughput in noise-free areas by reducing fragmentation acknowledgment overhead.

Table 17-1. Summary of common tunable parameters (continued)

Parameter	Meaning and units	Effect when decreased	Effect when increased
Long Retry Limit	Number of retransmission attempts for frames longer than the RTS threshold.	Frames are discarded more quickly, so buffer space requirement is lower.	Retransmitting up to the limit takes longer and may cause TCP to throttle back on the data rate.
Short Retry Limit	Number of retransmission attempts for frames shorter than the RTS threshold.	Same as long retry limit.	Same as long retry limit.
Listen Interval	Number of Beacon intervals between awakenings of power-saving stations.	Latency of unicast frames to station is reduced. Also reduces buffer load on access points.	Power savings are increased by keeping transceiver powered off for a larger fraction of the time.
DTIM Window	Number of Beacon intervals between DTIM transmissions (applies only to infrastructure networks).	Latency of multicast and broadcast data to power-saving stations is reduced. Also reduces buffer load on access points.	Power savings are increased by keeping transceiver powered off for a larger fraction of the time.
ATIM Window	Amount of time each station remains awake after a Beacon transmission in an independent network.	Increases power savings by allowing mobile stations to power down more quickly after Beacon transmission.	Latency to power-saving stations is reduced, and the buffer load may be decreased for other stations in the network.
Active Scan Timer	Amount of time a station waits after sending a Probe Response frame to receive a response.	Station moves quickly in its scan.	Scan takes longer but is more likely to succeed.
Passive Scan Timer	Amount of time a station monitors a channel looking for a signal.	Station may not find the intended network if the scan is too short.	Scan takes longer but is more likely to succeed.
Authentication Timeout	Maximum amount of time between successive frames in authentication sequence.	Authentications must proceed faster; if the timeout is too low, there may be more retries.	No significant effect.
Association Timeout	Maximum amount of time between successive frames in association sequence.	Associations must proceed faster; if the timeout is too low, there may be more retries.	No significant effect.

CHAPTER 18

The Future, at Least for 802.11

It's hard to make predictions,
especially about the future.
—Yogi Berra

This completes our picture of the current state of 802.11 networks. In this chapter, we'll get out a crystal ball and look at where things are heading. First, we'll look at standards that are currently in the works and close to completion. Then we'll take a somewhat longer-term look and try to draw conclusions about where wireless local networks are heading.

Current Standards Work

Publication of the 802.11 standard was only the beginning of wireless LAN standardization efforts. Several compromises were made to get the standard out the door, and a great deal of work was deferred for later. The 802.11 working group conducts its business publicly, and anybody can view their web site at *http://grouper.ieee.org/groups/802/11/* to get an update on the progress of any of these revisions to 802.11.

Revisions to the standard are handled by Task Groups. Task Groups are lettered, and any revisions inherit the letter corresponding to the Task Group. For example, the OFDM PHY was standardized by Task Group A (TGa), and their revision was called 802.11a.

Task Group D

802.11 was written with only North American and European regulatory agencies in mind. Expanding the wireless LAN market beyond these two continents required that the 802.11 working group study regulatory requirements for other locations. TGd was chartered for this purpose, and the required revisions were approved in June 2001 as 802.11d.

Task Group E: Quality of Service

TGe works on generic MAC enhancements, including both quality of service (QoS) and security. Security issues took on a life of their own, though, and were eventually split off into TGi, leaving only quality of service revisions for TGe. Work on the draft is currently being letter balloted; once any outstanding issues are worked out, the draft will be sent to the IEEE Standards Association for review.

Task Group F: A Standard IAPP

Right now, roaming between access points is possible only if all the access points in the BSS come from the same vendor. Before roaming between access points supplied by different vendors can become a reality, a standardized Inter-Access Point Protocol (IAPP) is required. With most of the bugs worked out of existing wireless LAN products on the market, customers are now turning to the challenge of expanding existing networks. Lack of a vendor-neutral IAPP is hindering these efforts and is a major driver behind the standardization work.

Task Group G: Higher ISM Data Rates

Now that we have reached the third millenium, data rates of 2 Mbps or even 11 Mbps can be considered "slow" for LAN applications. Many customers have invested in site surveys to identify potential problems with the RF environment on an ISM-band wireless LAN. Building a wireless LAN in the 2.4-GHz spectrum is reasonably well understood. Customers would like to take advantage of this existing work in understanding the ISM band with even higher data rates. The project authorization request that set up TGg specifies a target data rate of at least 20 Mbps.

The encoding mechanisms used by 802.11b are more sophisticated than the initial 802.11 draft, but they still leave a great deal of room for improvement. In May 2001, the Texas Instruments proposal to use Packet Binary Convolution Coding (PBCC) was removed from consideration; the surviving proposal to use OFDM in the ISM band made by Intersil looks likely to succeed, although it has not yet attained enough support to reach draft status.

Task Group H: Spectrum Managed 802.11a

Spectrum Managed 802.11a (SMa) incorporates two additional features into the 802.11a standard: dynamic channel selection and transmit power control. Both features are designed to enhance 802.11a networks; additionally, both are required to obtain regulatory approval for 802.11a devices in Europe. Dynamic channel selection improves the ability to coexist with other users of the 5-GHz bands because devices can select channels based on real-time feedback. One likely application of dynamic channel selection is selecting lower-powered channels for short-range

indoor situations while transparently switching to a higher-power channel when longer ranges are required. Transmit power control enables dense network deployments by allowing administrators to control the area that an access point services by tuning the power to achieve the desired size. Preliminary implementations of TGh enhancements have already been developed by vendors and will likely be revised to comply with the final standard.

Task Group I: Improving 802.11 Security

Security has been the most visible flaw in current 802.11 implementations and, as a result, has grabbed the headlines. Weak security was long suspected, and these suspicions became reality with the successful attempt to break WEP's cryptosystems. Initially, TGi had contemplated simply lengthening the key, but the success of the Fluhrer/Mantin/Shamir attack means that such an approach is also doomed to fail. TGi has refused to mandate any authentication protocols or key distribution mechanisms but has adopted the 802.1x framework and is moving forward.* Several proposals that have been made involve the use of PPP's Extensible Authentication Protocol (EAP) to authenticate users; EAP is specified in RFC 2284, which was in turn updated by RFC 2484. It is unclear what will come out of TGi, but the possibility that manual keying will remain the only method for key distribution leaves a great deal of room for third-party security products, especially if TGi does not finish standardization before 802.11a products are produced in volume.

The Longer Term

What does the picture look like over the longer term? 802.11 has already killed off other wireless efforts aimed at the home market (such as HomeRF), and it seems likely to severely constrain deployment of Bluetooth as well—which is unfortunate, because Bluetooth and 802.11 can coexist and serve different needs.

The larger issues for the long term are in areas such as wireless mobility and security, both of which present problems that aren't easily solved.

Mobility

Wireless networks are fundamentally about mobility. 802.11 deployments have successfully demonstrated that users are interested in mobility and that mobile connectivity is a long-felt need in the networking world. After all, users move, but network jacks do not.

However, 802.11 offers only link-layer mobility, and that is possible only when the access points can all communicate with each other to keep track of mobile stations.

* The recent discovery of flaws in 802.1x makes it seem likely that this story is far from over.

Standardization of the IAPP by Task Group F should make it easier to deploy networks by facilitating interoperability between multiple vendors. Standardization of the IAPP will also make it easier to merge multiple distinct wireless networks into each other.

A lot of planning is required if you want to deploy a wireless LAN with a substantial coverage area, especially if seamless mobility through the entire coverage area is a requirement. In most deployments, the requirement for centralized planning is not in and of itself a hurdle. With community networks, however, loose cooperation is the watchword, and centralized command-and-control planning makes it hard to build a functional network. Furthermore, community networking groups do not have the resources to purchase big Ethernet switches, extend VLANs between houses, and create a neighborhood fiber backbone. Community networks are typically a federation of access points that offer Internet access through NAT. Without centralized addressing, mobile users will need to obtain new DHCP addresses when roaming throughout a network.

It's easy to overlook the problems that mobility causes at the IP level. It's easy to say that a wireless node should receive a new IP address when it moves from one part of a larger network to another, but in practice, it isn't simple. How do you get the node to notice that it should abandon its current address and ask for a new one? What happens to network connections that are open when the node's IP address changes?

Avoiding the NAT requirement and the need to periodically change addresses requires mobility at the network layer. Mobile IP has been standardized to a reasonable degree, but an open source Mobile IP implementation of choice has yet to emerge. Barring a large commercial driving force, it is likely that community networking will drive future open source Mobile IP implementation. When Mobile IP can be deployed without significant headaches, there may be another burst in the market as network managers discover that deployment of wireless networks can happen without painful readdressing or Ethernet switching games.

Wireless LANs are likely to borrow a few concepts from mobile telephony. When European telephony experts wrote the standards on which modern second-generation cellular networks are built, it was explicitly recognized that no single telecommunications carrier had the resources to build a pan-European cellular network. Wireless telephony had previously been held back by a plethora of incompatible standards that offer patchwork coverage throughout parts of Europe. Experts realized that the value of carrying a mobile telephone was proportional to the area in which it could be used. As a result, the GSM standards that were eventually adopted emphasized roaming functions that would enable a subscriber to use several networks while being billed by one network company. As public-access wireless LANs become larger, authentication and roaming functions are likely to be a larger focus in both the 802.11 working group and the IETF. Network sharing is another concept that will likely be borrowed from the cellular world. The incredible cost of third-generation licenses has pushed a

less expensive), so wireless networks tend to have relatively few users, and the networks themselves are physically relatively far apart. What happens when they're stressed? What would a wireless network be like if it had, say, 1,000 users (which can easily be supported by a well-designed wired network)? What would it be like in a large office building, where you might have half a dozen companies, each with its own network, in the space of two or three floors?

We don't really have the answers to these questions yet. As wireless becomes more common, we'll be forced to answer them. It is clear, though, that there are resource constraints. Current technologies will suffer from overcrowding within the unlicensed 2.4-GHz band. 802.11a and other technologies will move to the 5-GHz band, but crowding will eventually become an issue there, too. Meanwhile, commercial users are fighting for additional frequency space, and it's not likely that governments will allocate more spectrum to unlicensed users.

There are a few ways out of this problem. Improved encoding techniques will help; the use of directional antennas may make it possible for more devices to coexist within a limited space. Directional antennas aren't without cost: the more effective a directional antenna is, the harder it is to aim. It's all very nice to imagine sitting at a picnic table in front of your company's office with your portable antenna aimed at the access point on the roof; but what if somebody else sits down and knocks over the antenna? How much of a pain will it be to reorient it?

Other solutions have already been mentioned in the discussion of continuing standards work. We'll certainly see cards that can switch between high-power and low-power transmission, possibly even changing channels on the fly as power requirements change.

Deployment

One of the major problems faced by the architects of public-access wireless networks is that the service is quite generic. In the absence of any constraints, anybody can put up an antenna and offer Internet access via 802.11 equipment. Mobile telephony coped with this problem by licensing the spectrum so that only one carrier had the right to transmit in a given frequency band. A similar solution is not possible with 802.11 because it explicitly uses the unlicensed bands. Competition between network providers therefore shifts to the political layer of the OSI model; an example is a network provider that leases space from the building owner for the exclusive right to deploy a wireless network in a given area. Airports are a leading proponent of this new approach, though many observers believe that allowing genuine competition would better serve the interests of users.

Other deployment problems faced by wireless LAN builders are common throughout the industry: obtuse command lines, the need for extensive attention to individual desktops, and the lack of large-scale automation. Vendors are concentrating on

number of cellular carriers to the brink of bankruptcy. In reponse, groups of carriers are now planning to share the data-carrying infrastructure to share the cost and risk of an expensive third-generation network build-out. The high costs of building a wide-scale 802.11 network are likely to have the same effect, with the surviving "hot zone" players regrouping around one set of multitenant access points in the coverage area.

Security

Wireless networks have all been tarred with the brush of poor security. Weaknesses in the Wired Equivalent Privacy (WEP) standard made the news with a great deal of regularity in 2001, culminating with a total break partway through the year. Even against this backdrop, though, the market for 802.11 network equipment has exploded. Better security mechanisms are needed to usher in centrally coordinated rollouts at large, security-conscious institutions, but the apparent security weaknesses in current equipment have not prevented the market from forming and growing rapidly.

Many observers long suspected that WEP was fundamentally broken. (It was humorously derided as "Wiretap Equivalence, Please" by several commentators.) WEP suffers from several fatal design flaws, and it is now clear that something far better is needed. One common tactic to address the shortfalls of WEP is to require the use of stronger encryption technology over the wireless link. Unfortunately, this only shifts the problem elsewhere. Adding VPN client software moves the problem from security weaknesses in the wireless link to managing desktop software images. System integration of VPN client software is difficult on the best of days. Integration so far has proven too large a hurdle for most users. Moving forward, 802.11 will need to incorporate public-key mutual authentication of stations and access points and random session keys. Neither is a new idea, and both have been used for years in both *ssh* and IPSec.

Most importantly, though, different access controls are needed for the future. Right now, most wireless LANs are used to extend corporate LANs throughout the office. Existing authentication concepts were designed for a known, static user group, such as a group of employees. Just as in cellular telephony, the promise of wireless networks is installation in hard-to-reach spots or places where users are on the move. Designing an 802.11 network for an airport or train station requires dealing with the question of who is allowed to use the network. These problems aren't trivial; service providers need to think about how to protect users from each other and how to authenticate users to access points and other services on the network.

Radio Resources

So far, wireless networks have had a free ride. They are fairly exotic, and wireless cards still aren't common (though they're selling quickly and, as a result, becoming

large-scale management frameworks to ease the pain of network administrators stuck with managing tens or hundreds of access points. Better analysis tools, both for the site survey phase and the troubleshooting/deployment phase, are required by network engineers.

The End

And that's it. The future of wireless networking isn't without its problems, but that's no different from any other technology. The price of network cards is dropping rapidly—they now cost about half what they did when I started writing, and it wouldn't be surprising if they were eventually little more expensive than the typical Ethernet card. While it's probably true that equipment that stays fixed will benefit from a fixed network—it's hard to imagine backing up a large database over a wireless network, for example—we will certainly see more and more computer users accessing their networks through wireless links, leaving their desks and working in the park, the library, or wherever they're most comfortable.

APPENDIX A
802.11 MIB

802.11 contains extensive management functions to make the wireless connection appear much like a regular wired connection. The complexity of the additional management functions results in a complex management entity with dozens of variables. For ease of use, the variables have been organized into a management information base (MIB) so that network managers can benefit from taking a structured view of the 802.11 parameters. The formal specification of the 802.11 MIB is Annex D of the 802.11 specification.

The Root of the Matter

The 802.11 MIB is designed by the 802.11 working group. Like other MIBs, it is based on a global tree structure expressed in Abstract Syntax Notation 1 (ASN.1) notation. Unlike SNMP MIBs, though, the 802.11 MIB has a different root: *.iso.member-body.us.ieee802dot11* (.1.2.840.10036). (For simplicity, the prefix will be omitted from any objects described in this appendix, much as *.iso.org.dod.internet* is omitted from SNMP object names.)

The basic structural overview of the 802.11 MIB is shown in Figure A-1. Four main branches compose the MIB:

dot11smt
: Contains objects related to station management and local configuration

dot11mac
: Composed of objects that report on the status of various MAC parameters and allow configuration of them.

dot11res
: Contains objects that describe available resources

dot11phy
: Report on the status of the various physical layers

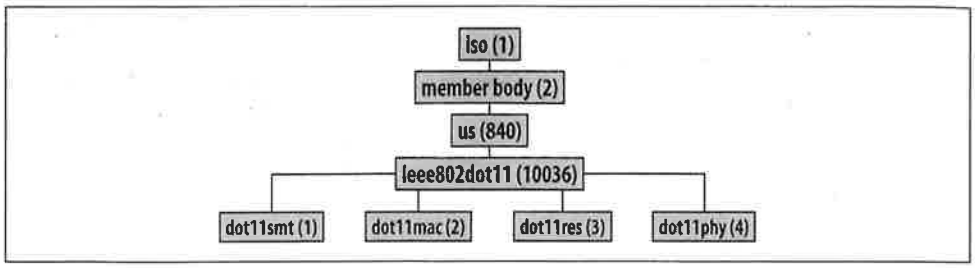

Figure A-1. The 802.11 MIB root and its main branches

Station Management

Station management is the term used to describe the global configuration parameters that are not part of the MAC itself. Figure A-2 shows a high-level view of the station management branch of the MIB. Six subtrees organize information on global configuration, authentication, and privacy, and provide a means for automated notification of significant events.

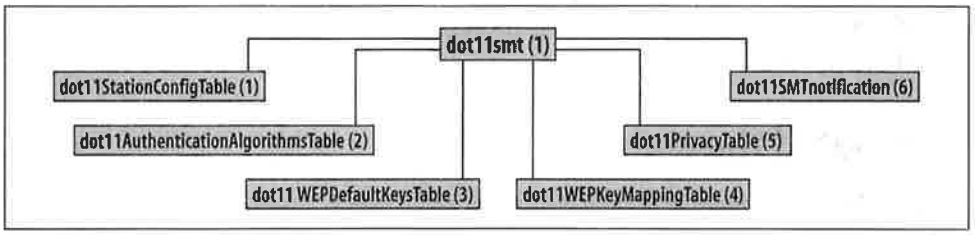

Figure A-2. Main branches of the station management (SMT) tree

The Station Configuration Table

The main table in the station management tree is the global configuration table, *dot11StationConfigTable*, which is shown in Figure A-3. All entries in the station configuration table start with the prefix *dot11smt.dot11StationConfigTable.dot11StationConfigEntry* (1.1.1):

dot11StationID (MacAddress)
: The bit string in this object is used to identify the station to an external manager. By default, it is assigned the value of the station's globally unique programmed address.

dot11MediumOccupancyLimit (integer, range 0–1,000)
: This object has meaning only for stations that implement contention-free access using the point coordination function. It measures the number of continuous time units (TUs) that contention-free access may dominate the medium. By default, it is set to 100 TUs, with a maximum value of 1,000 (1.024 seconds).

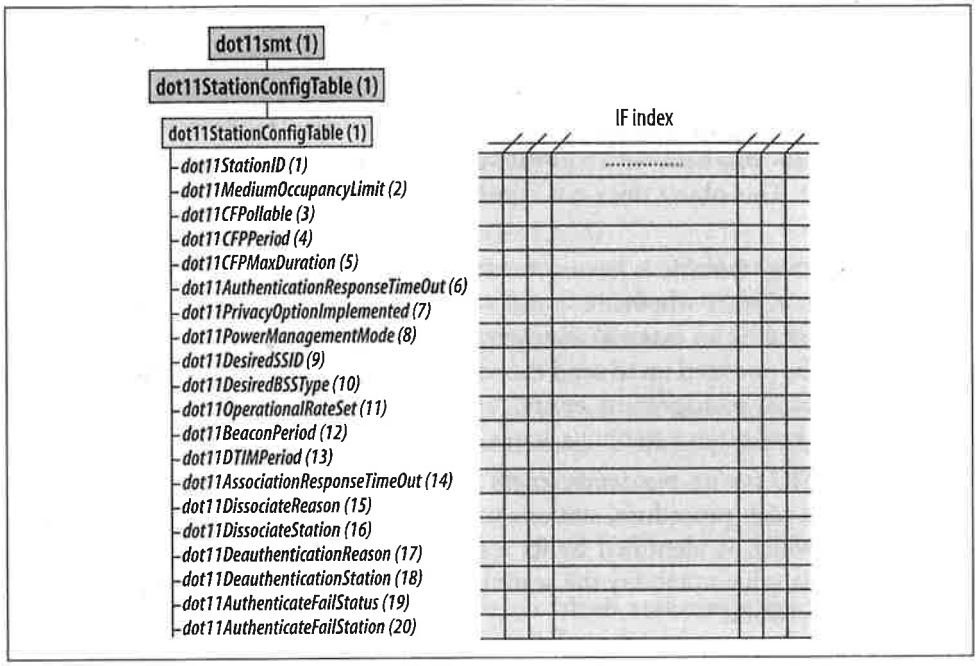

Figure A-3. The station configuration table dot11StationConfigTable

Making this value larger increases the capacity allocated to contention-free service. Decreasing it reduces the amount of time available for contention-free service. Network managers can tune this value appropriately once products implementing contention-free service are produced.

dot11CFPollable (TruthValue: a defined type set to 1 for true or 2 for false)

This object reports whether a station can respond to contention-free polling messages. Because the ability to respond to CF-Poll messages is a function of the software installed, this object is read-only.

dot11CFPPeriod (integer, range 0–255)

Contention-free periods always begin on a DTIM message. This object is the number of DTIM intervals between contention-free periods. DTIM messages are always a part of a Beacon frame, so the time between contention-free period starts can be obtained by multiplying the CFP period value in this object by the DTIM interval.

dot11MaxDuration (integer, range 0–65,535)

This object describes the maximum duration of the contention-free period when a new BSS is created. The reason for the difference in the allowed range of values between this object and the *dot11MediumOccupancyLimit* is a mystery.

dot11AuthenticationResponseTimeout (integer, range 1–4,294,967,295)

At each step of the station authentication process, the station will wait for a timeout period before considering the authentication to have failed. This object

contains the number of TUs that each step is allowed to take before failing the authentication.

dot11PrivacyOptionImplemented (TruthValue)

If the station implements WEP, this object will be *true* (1). By default, it is *false* (2). It is read-only because a software update to the station is required to implement WEP. This object does not report whether WEP is in use, only whether it is available.

dot11PowerManagementMode (enumerated type)

This object reports whether a station is *active* (1) or in a *powersave* (2) mode. When queried by an external manager, it will always return active because a station must be powered up to send the response frame. This object is far more useful for a local management entity, which can poll the object periodically to determine how often a station is active.

dot11DesiredSSID (string, maximum length 32)

During the scan procedure, stations may be configured to look for a particular network, which is identified by its service set ID (SSID). Management entities may set this value to modify the scanning process to preferentially associate with a certain network.

dot11DesiredBSSType (enumerated)

Like the previous object, this object is also used to configure the scanning procedure. By setting this object, a manager can force association with an *infrastructure* (1) network, an *independent* (2) BSS, or *any* (3) BSS.

dot11OperationalRateSet (string, maximum length 126)

Initially, this described the data rate as a number that was the number of 500-kbps increments for the data rate. Recent standardization work is likely to make them simple labels because the range of 1–127 allows only a maximum rate of 63.5 Mbps.

dot11BeaconPeriod (integer, range 1–65,535)

This object contains the length of the Beacon interval, in TUs. Once a BSS has been established with the Beacon interval, the value may be changed. However, it will not take effect until a new BSS is created.

dot11DTIMPeriod (integer, range 1–255)

This object contains the number of Beacon intervals between DTIM transmissions.

dot11AssociationResponseTimeOut (integer, range 1–4,294,967,295)

At each step of the association process, the station will wait for a timeout period before considering the association attempt to have failed. This object contains the number of TUs that each step is allowed to take before failing the association.

dot11DisassociateReason and dot11DeauthenticateReason (integer, range 0–65,535)

This object contains the reason code from the most recently transmitted Disassociation or Deauthentication frame. If no such frame has been transmitted, the value is 0. For a complete list of reason codes, see Chapter 3 or clause 7.3.1.7 of 802.11.

dot11DisassociateStation and dot11DeauthenticateStation (MacAddress)
: This object contains the MAC address of the station to which the most recently Disassociation or Deauthentication frame was transmitted. If there is no such frame, the value is 0. By using either of these objects in combination with the corresponding Reason object described previously, a manager can track which station was kicked off the network and why.

dot11AuthenticateFailStatus (integer, range 0–65,535)
: This object contains the status code from the most recently transmitted Authentication Failure frame. If no such frame has been transmitted, the value is 0. For a complete list of status codes, see Chapter 3 or clause 7.3.1.9 of 802.11.

dot11AuthenticateFailStation (MacAddress)
: This object contains the MAC address of the station to which the most recent Authentication Failure frame was transmitted. If there is no such frame, the value is 0.

Authentication Algorithms Table

The authentication algorithms table reports on which authentication algorithms are supported by the station on each interface. It is best thought of as the three-dimensional array shown in Figure A-4.

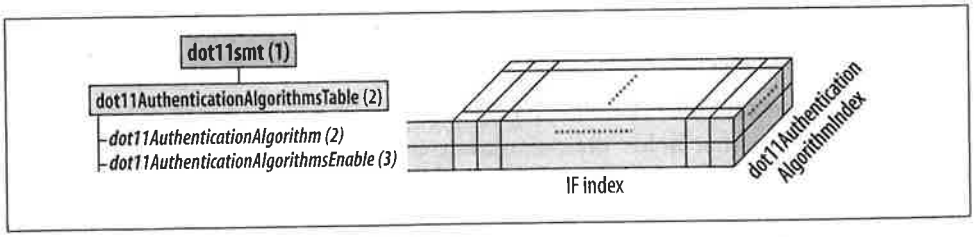

Figure A-4. The authentication algorithms table

In addition to the ifIndex index variable, an auxiliary index variable is used. The reason for the auxiliary variable is that multiple authentication algorithms exist, and the table should report on all of them. For each index, the auxiliary index allows several cells to report on one authentication algorithm each. Each cell has two component objects:

dot11AuthenticationAlgorithm (enumerated type)
: This object is set either to *openSystem* (1) or *sharedKey* (2) to report on the authentication algorithm.

dot11AuthenticationAlgorithmsEnable (TruthValue)
: This object reflects whether the corresponding authentication algorithm noted by the previous object is supported by the interface that indexes the table. By default, it is set to *true* (1) for open-system authentication and *false* (2) for shared-key authentication.

WEP Key Tables

Two tables report on the status of WEP key information. Both use the *WEPKeytype* defined data type, which is simply a 40-bit string. The WEP key tables are shown in Figure A-5. WEP key tables are supposed to be write-only, but a security advisory was issued in June 2001 against one vendor who exposed the keys to queries using SNMP.

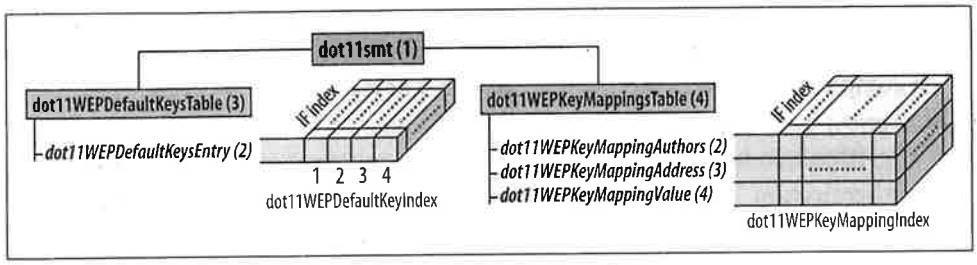

Figure A-5. WEP key tables

WEP default key table

The default key table is quite simple. Each interface has a maximum of four default keys associated with it because the WEP specification allows for four default keys per network. When transmitted in frames over the air, the WEP key ID for the default key runs from 0–3. In this table, however, the index runs from 1–4. Each cell in the table has just one object:

dot11WEPDefaultKeyValue (WEPKeytype)
: A 40-bit string that holds the default key for the interface and default key ID specified by the location in the table.

WEP key-mapping table

WEP supports using a different key for every MAC address in the world. Keys can be mapped to the unique pair of (transmitter address, receiver address). For each interface on the system, an arbitrary number of address and key pairs can be associated with that interface. Each interface uses an auxiliary index to identify all the MAC addresses associated with keys, plus information about each address. The following three objects are used to describe a key-mapping relationship. A fourth object in the row gives the row status, but the row status indicator is used only when creating or deleting table rows.

dot11WEPKeyMappingAddress (MacAddress)
: This is the "other" address of the address pair for which a key-mapping relationship exists.

dot11WEPKeyMappingWEPOn (TruthValue)
: This object is set to *true* (1) when WEP should be used when communicating with the key-mapping address. If WEP should not be used, this object is set to *false* (2).

dot11WEPKeyMappingValue (WEPKeytype)
: This is the 40-bit keying information used as the shared secret for WEP. Logically, this cell in the table should be write-only, but not all implementations will obey that convention.

MAC Management

The MAC branch of the 802.11 MIB provides access to objects that allow administrators to tune and monitor MAC performance and configure multicast processing. It is divided into three main groups as shown in Figure A-6.

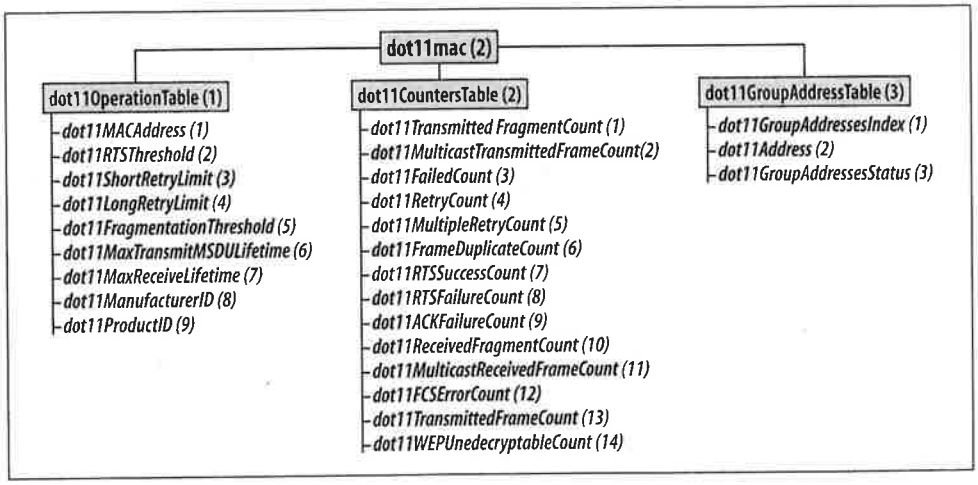

Figure A-6. The MAC attributes subtree

The dot11OperationsTable

The main table in the station management tree is the global configuration table, *dot11OperationTable*, which is shown in Figure A-6. All entries in the station configuration table start with the prefix *dot11mac.dot11OperationTable.dot11OperationEntry* (2.1.1) and are indexed by a system interface:

dot11MACAddress (MacAddress)
: This object is the MAC address of the station. By default, it is the globally unique address assigned by the manufacturer. It may, however, be overridden by a local manager.

dot11RTSThreshold (integer, range 0–2,347; default value is 2,347)

Any unicast data or management frames larger than the RTS threshold must be transmitted using the RTS/CTS handshake exchange. By default, the value is 2,347, which has the effect of deactivating RTS/CTS clearing before transmission. Setting the object to 0 activates the RTS/CTS handshake before every transmission.

dot11ShortRetryLimit (integer, range 1–255; default value is 7)

The short retry limit is applied to frames that are shorter than the RTS threshold. Short frames may be retransmitted up to the short retry limit before they are abandoned and reported to higher-layer protocols.

dot11LongRetryLimit (integer, range 1–255; default value is 4)

The long retry limit is analogous to the short retry limit but applies to frames longer than the RTS threshold.

dot11FragmentationThreshold (integer, range 256–2,346; default value is vendor-specific)

When a unicast frame exceeds the fragmentation threshold, it is broken up. MAC headers and trailers count against the threshold limits.

dot11MaxTransmitMSDULifetime (integer, range 1–4,294,967,295; default value is 512)

This object is the number of TUs that the station will attempt to transmit a frame. If the frame is held by the MAC longer than the lifetime, it is discarded, and the failure is reported to higher-level protocols. By default, it is set to 512 TUs, or about 524 ms.

dot11MaxReceiveLifetime (integer, range 1–4,294,967,295; default is 512)

When MAC-layer fragmentation is used, the receiver uses a timer to discard "old" fragments. After the reception of the first fragment of a frame, the clock starts running. When the timer expires, all fragments buffered for reassembly are discarded.

dot11CountersTable

Counters allow a local or external management entity to monitor the performance of a wireless interface. One of the major uses of data from the counters is to make informed performance-tuning decisions. All entries in this table begin with *dot11mac.dot11CountersTable.dot11CountersEntry* (2.2.1):

dot11TransmittedFragmentCount (Counter32)

This counter is incremented for all acknowledged unicast fragments and multicast management or multicast data fragments. Frames that are not broken up into pieces but can be transmitted without fragmentation also cause this counter to be incremented.

dot11MulticastTransmittedFrameCount (Counter32)

This counter is incremented every time a multicast frame is sent. Unlike the previous counter, acknowledgement is not necessary.

dot11FailedCount (Counter32)
When frames are discarded because the number of transmission attempts has exceeded either the short retry limit or the long retry limit, this counter is incremented. It is normal for this counter to rise with increasing load on a particular BSS.

dot11RetryCount (Counter32)
Frames that are received after requiring a retransmission will increment this counter. Any number of retransmissions will cause the counter to increment.

dot11MultipleRetryCount (Counter32)
Frames that require two or more retransmissions will increment this counter. As such, it will always be lesser than or equal to the retry count.

dot11FrameDuplicateCount (Counter32)
Duplicate frames arise when acknowledgements are lost. To provide an estimate of the number of lost acknowledgements from the station, this counter is incremented whenever a duplicate frame is received.

dot11RTSSuccessCount (Counter32)
Reception of a CTS in response to an RTS increments this counter.

dot11RTSFailureCount (Counter32)
When no CTS is received for a transmitted RTS, this counter is incremented.

dot11ACKFailureCount (Counter32)
This counter directly tracks the number of inbound acknowledgements lost. Whenever a frame is transmitted that should be acknowledged, and the acknowlegment is not forthcoming, this counter is incremented.

dot11ReceivedFragmentCount (Counter32)
This counter tracks all incoming data and management fragments. Full frames are considered fragments for the purpose of incrementing this counter.

dot11MulticastReceivedFrameCount (Counter32)
This counter tracks incoming multicast frames.

dot11FCSErrorCount (Counter32)
If the frame check calculation fails, this counter is incremented. This is one of the main counters that give network administrators an idea of the health of a BSS.

dot11TransmittedFrameCount (Counter32)
The transmitted frame count is the number of successfully transmitted frames. Part of the definition of successful transmission is that an expected acknowledgement is received.

dot11WEPUndecryptableCount (Counter32)
This counter is incremented when an incoming frame indicates that it is encrypted using WEP, but no decryption is possible. Naturally, this number will increment rapidly on stations that do not implement WEP when WEP frames are

used on the network. It can also increment rapidly when the key mapping key is invalid or the default key is incorrect.

Group Addresses Table

The group addresses table maintains a list of multicast addresses from which the station will accept frames. The table is a list of *dot11Address* objects of type *MacAddress*, indexed by an auxiliary variable and the interface index. The table is shown in Figure A-6.

Physical-Layer Management

Physical-layer management is divided into 11 tables, as shown in Figure A-7. Certain tables are specific to a certain product, or the implementation highly vendor-dependent, so those tables are not described in this appendix. There is a table for the infrared physical layer, but that table is omitted as well because no products on the market implement the infrared physical layer.

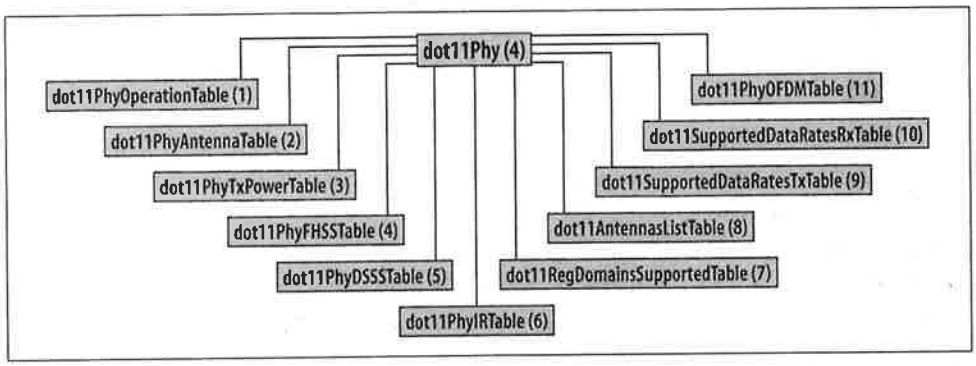

Figure A-7. Physical-layer MIB branches

Physical-Layer Operations Table

Overall operational status is reported by the physical-layer operations table, *dot11phy.dot11PhyOperationTable* (4.1) (Figure A-8). Each interface creates corresponding entries in the operation table, which allows a manager to determine the physical-layer type and access the appropriate related branch in the physical-layer subtree. All entries in the physical-layer operation table begin with *dot11phy.dot11PhyOperationTable.dot11PhyOperationEntry* (4.1.1):

dot11PHYType (enumerated)

This object is set to *fhss* (1) for frequency-hopping radio systems and *dsss* (2) for direct-sequence radio systems. It may also be set to *irbaseband* (3) if IR systems

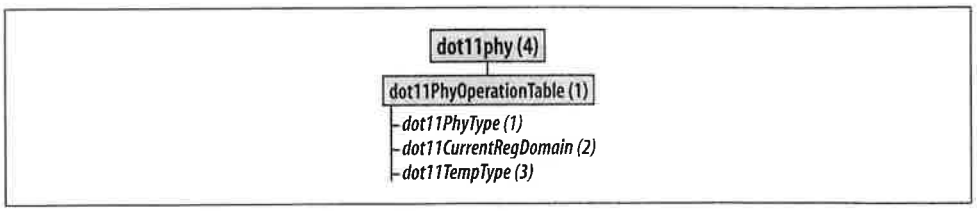

Figure A-8. Physical-layer operations table

are ever implemented and will be set to *ofdm* (4) for the OFDM PHY specified in 802.11a.

dot11CurrentRegDomain (enumerated)
: This object is set to the current regulatory domain of the system. Several values are possible, as shown in Table A-1.

Table A-1. Regulatory domains in the 802.11 MIB

Value	Regulatory authority	Regulatory area
fcc (16)	U.S. Federal Communications Commission	United States
doc (32)	Industry Canada	Canada
etsi (48)	European Telecommunications Standards Institute	Europe, excluding France and Spain
spain (49)		Spain
france (50)		France
mkk (64)	Radio Equipment Inspection and Certification Institute	Japan

dot11TempType (enumerated)
: This object is set to *tempType1* (1) for a commercial operating temperature range of 0–40 degrees Celsius and *tempType2* (2) for an industrial operating temperature range of 0–70 degrees Celsius.

FHSS Table

The MIB table for FH PHYs is shown in Figure A-9. Entries in the frequency-hopping table begin with *dot11phy.dot11PhyFHSSTable.dot11PhyFHSSEntry* (4.4.1):

dot11HopTime (integer, set to 224)
: The time, in microseconds, required for the frequency-hopping PMD to change from channel 2 to channel 80. This is fixed by the specification, so the corresponding MIB definition is also fixed.

dot11CurrentChannelNumber (integer, range 0–99)
: This object is set to the current operating channel. In a frequency-hopping system, the operating channel changes frequently.

dot11MaxDwellTime (integer, range 1–65,535)
: The maximum amount of time the transmitter is permitted to use a single channel, in TUs.

dot11CurrentSet (integer, range 1–255)
: The number of the hop pattern set currently employed by the station.

dot11CurrentPattern (integer, range 0–255)
: The current hopping pattern in use by the station.

dot11CurrentIndex (integer, range 1–255)
: The index in the current hop pattern that determines the current channel number. This index will select the hop frequency in *dot11CurentChannelNumber*.

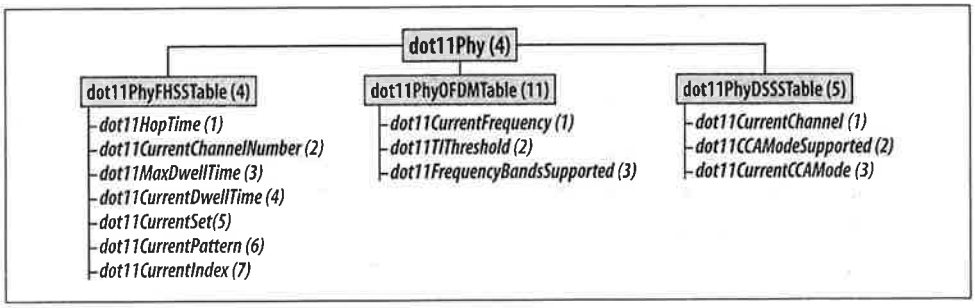

Figure A-9. Physical-layer attribute tables

DSSS Table

The direct-sequence table is also shown in Figure A-9. All entries in the direct-sequence table are indexed by the interface number and begin with *dot11phy.dot11PhyDSSSTable.dot11PhyDSSSEntry* (4.5.1):

dot11CurrentChannel (integer, range 1–14)
: This object reports the current operating channel. Valid channels range from 1–14.

dot11CCAModeSupported (integer, 1–7)
: Three main clear-channel assessment options can be used to determine the status of a direct-sequence operating channel: energy detection (ED), carrier sense (CS), or a combination of both. The three options are assigned numerical values and added up to produce the value reported by this object. Only energy detection is assigned the value 1. Only carrier sense is assigned the value 2. Using both in combination is assigned the value 4. A system that supports all three modes reports a value of 7 in this object.

dot11CurrentCCAMode (enumerated)
: This object is set to *edonly* (1) if only energy detection is used to assess channel clarity. If only the MAC carrier sense functions are used, the object is set to *csonly* (2). If both are employed, the object is set to *edandcs* (4).

OFDM Table

When the OFDM PHY was standardized in 802.11a, a corresponding table was added in the MIB to report on its status. All entries in the OFDM table, which is also shown in Figure A-9, begin with *dot11phy.dot11PhyOFDMTable.dot11PhyOFDMEntry* (4.11.1). The table is indexed by interface number.

dot11CurrentFrequency (integer, range 0–99)

This is the current operating-frequency channel number of the OFDM PHY.

dot11TIThreshold (Integer32)

This medium will report as busy if a received signal strength is above this threshold.

dot11FrequencyBandsSupported (integer, range 1–7)

The OFDM PHY is designed for the U-NII frequency bands. The lowest band (5.15–5.25 GHz) is assigned the value 1, the midband (5.25–5.35 GHz) is assigned the value 2, and the high band (5.725–5.825 GHz) is assigned the value 4. The values corresponding to the supported frequency bands are added to produce the value returned by a query to this object.

APPENDIX B

802.11 on the Macintosh

Apple Computer has been a key player in establishing the market for 802.11 equipment. Most companies in the 802.11 market saw their contributions in terms of standards committee activity and technology development. Apple contributied by distilling complex technology into an easy-to-use form factor and applying its mass-marketing expertise.

In 1999, 802.11 was a promising technology that had demonstrated its value in a few narrow markets. 802.11 interfaces cost around $300, and access points were around $1,000. Apple saw the promise in the technology and moved aggressively, releasing $300 access points and $99 interfaces. With a new competitor suddenly pricing the gear at a third of the prevailing price, other vendors were forced to drop prices dramatically, and the market took off. Prices have been dropping over the last year. 802.11 will probably never be as cheap as Ethernet, but it's easy to imagine 802.11 interfaces under $50 and access points under $100.

This appendix was made possible by a generous equipment loan from Apple. Apple loaned me an iBook running Mac OS X 10.1, the dual Ethernet version of the AirPort Base Station, and their software tools for configuration and management. Unfortunately, it arrived too late for me to include in the discussion of Apple's hardware; that will have to wait for the second edition. The late delivery also prevented me from presenting the Software Base Station, a MacOS 9 application that turns any Macintosh with an AirPort card into a base station. (It was not available for MacOS X as this book went to press, so time constraints prevented its inclusion.)

The AirPort Card

Apple offers tightly integrated systems because the hardware and the software are designed in tandem. Unlike the chaotic IBM-compatible world, with Apple one company is responsible for both the hardware and software, and it shows. You can install the hardware and software and connect to an existing network in only a few minutes.

If an AirPort card is plugged in during system installation, this can be done as part of the initial configuration

Hardware Installation

AirPort interface cards are specialized 802.11 interfaces designed to be inserted into an AirPort slot, a feature on every machine Apple has introduced since 2000. AirPort-capable Macintoshes contain an antenna in the machine's case, so the AirPort card can use a much larger antenna than most PCMCIA interfaces in the PC world. With the antenna integrated into the case, it also eliminates the hazard of a protruding antenna. I have also found that the integrated antenna has better range than many PCMCIA-based 802.11 interfaces used with Windows.

Installing the AirPort card into an iBook is a relatively simple procedure. The iBook should be powered down, unplugged, and have its battery removed. The AirPort slot is underneath the keyboard. Lift up the keyboard, slide the card in, and connect the computer's built-in antenna cable to the card. Other machines are just as straightforward. PowerBooks work the same as iBooks. The AirPort slot on iMacs is under the bottom panel; tower machines have a slot readily available under the cover. The external antenna connector is the same connector used by Lucent products, so users who really want an external antenna can use Lucent gear.

Software Installation

Drivers for the AirPort are included in OS 9.1 and later, so there is no need to download and install drivers. If the AirPort card is installed before the system first boots, the first-time boot configuration utility will allow you to configure the AirPort interface out of the box by selecting a network name and choosing how to configure TCP/IP. For a network that uses DHCP, the configuration instructions are only a few screens long.

AirPort cards added after the system first boots can be configured by the AirPort Setup Assistant.* After inserting the card, run the Setup Assistant. When it starts, you will see the dialog box in Figure B-1.

Choose to configure the AirPort card and click Continue. The next step, shown in Figure B-2, is to select the network you wish to join. Every network within range is displayed in the pop-up menu. Figure B-2 shows the user selecting the Little Green Men network.

* Cards can also be configured with the System Preferences application. For completeness, this appendix discusses the Setup Assistant first and the System Preferences application later in terms of monitoring and changing the configuration. However, there is no reason why you cannot configure the card straight from the System Preferences application.

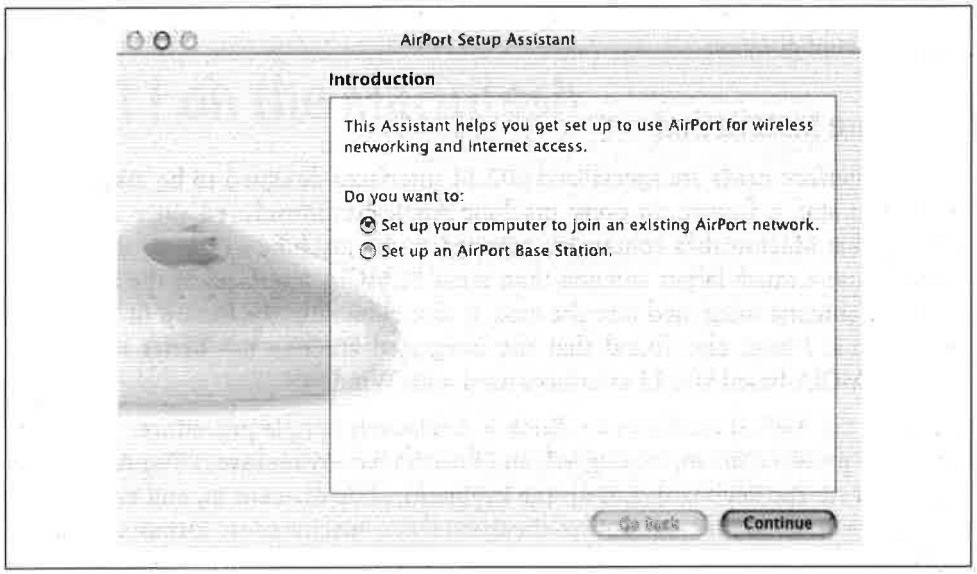

Figure B-1. Initial AirPort Setup Assistant screen

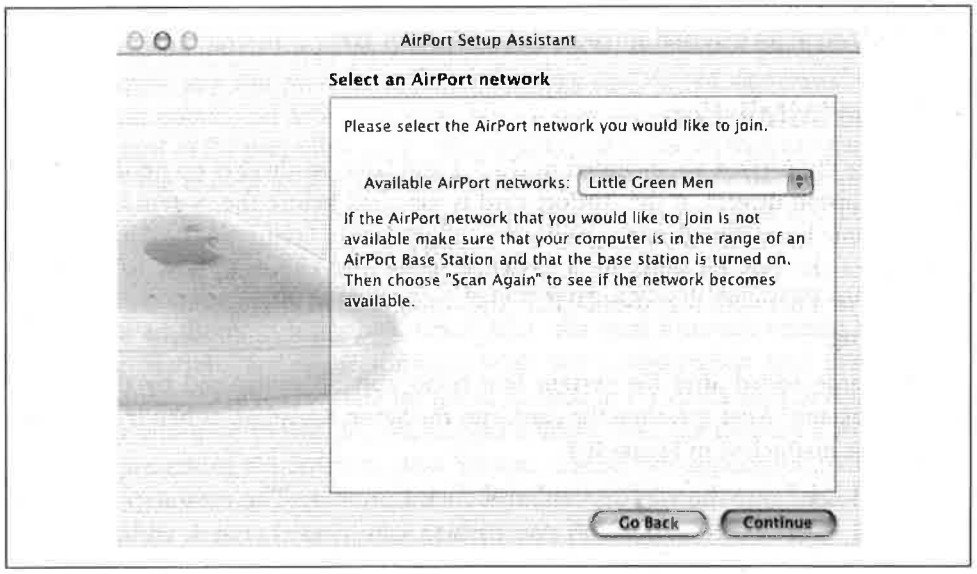

Figure B-2. AirPort card network selection

Once the network is selected, you can move on to the third step: entering the network password. This is the WEP configuration. To make matters easier for users, Apple has allowed administrators to input WEP keys as variable-length passwords. The ASCII text of the password is then hashed into a WEP key of the appropriate length. WEP keys can also be entered in hexadecimal by prefacing them with a dollar sign, such as `$EB102393BF`. Hex keys are either 10 hex digits (40-bit) or 26 hex digits

(104-bit). Interpretation of the input string can be forced to ASCII by enclosing the key in double quotes.* Input the key, if any, into the box in Figure B-3, then move on by clicking the Continue button.

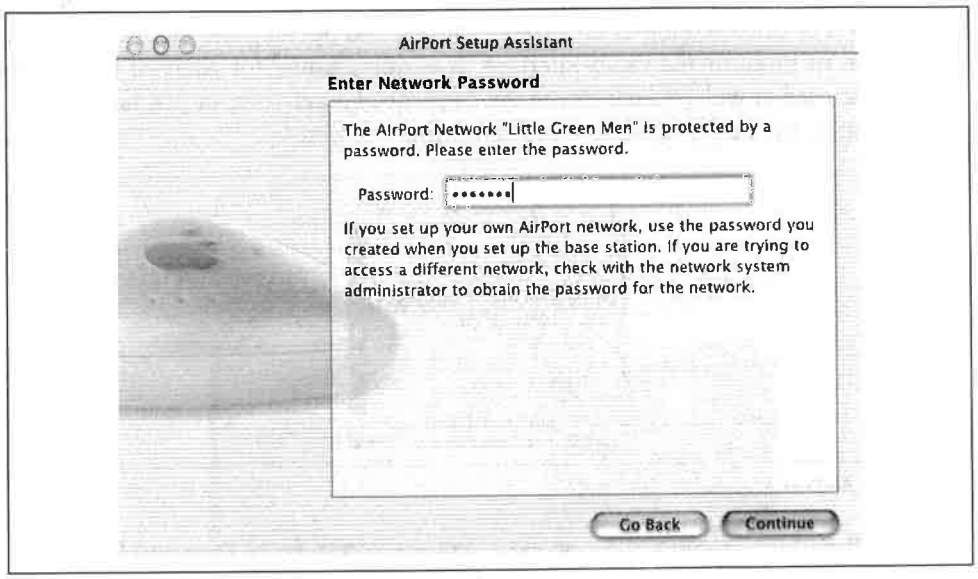

Figure B-3. AirPort network password entry

After these three steps are performed, the setup is complete, and you will be joined to the wireless network. Subsequent screens inform the user that the network configuration is complete.

Configuring and Monitoring an AirPort Interface

You may need to change the AirPort configuration from time to time. You may move between different 802.11 networks (ESSs) or need to change the WEP keys or IP settings. Once the card is installed, you can change its configuration with the configuration tools provided with OS X. Apple's configuration programs do not allow users to change any of the more complex 802.11 parameters. All the information needed to join a network, for example, is broadcast in the Beacon frames. Apple decided that in most cases, it is sufficient to simply present the user with network names and prompt for a password if needed.

Basic configuration with the AirPort status icon

Once configuration is complete, the AirPort status icon is displayed in the upper-right corner of the screen, next to the speaker volume, battery, and clock icons, provided

* For more information, see article 106250 in Apple's Knowledge Base at *http://kbase.info.apple.com*.

you haven't turned off those icons. The AirPort icon also indicates radio strength. In Figure B-4, there are several solid wavefronts on the icon. As you move farther from the access point and the signal degrades, the number of bars decreases. When it is clicked, a drop-down command list offers the option of turning the power to the AirPort card on or off, selecting or creating networks, and opening the Internet Connect application to monitor the radio interface. It is quite handy for users to be able to turn off the card at will. When you are out of range of a network, or just not using it, the card can easily be powered down to save battery power.

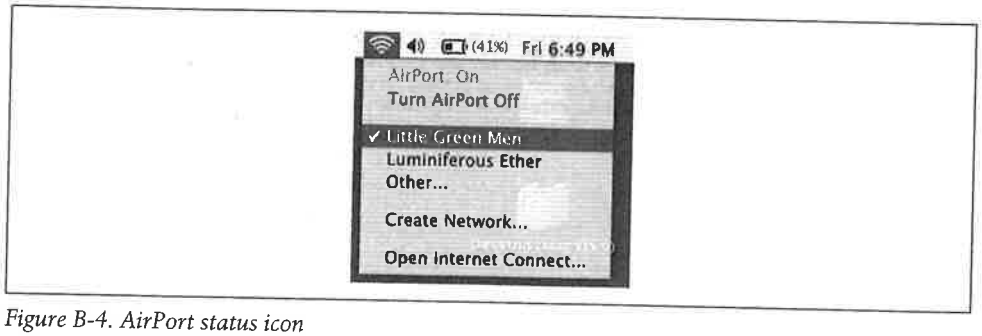

Figure B-4. AirPort status icon

In Figure B-4, there are two networks within range: Little Green Men and Luminiferous Ether. The checkmark by Little Green Men indicates that it is the network to which the user is currently connected, though the user can switch between the two networks simply by selecting a preference. Any other network can be selected by using the Other... option and entering the name of the network.

An IBSS can be created by going to the Create Network option and selecting the basic radio parameters shown in Figure B-5. The computer is set to create an IBSS with the network name of Very Independent BSS; the radio channel defaults to 11 but can be changed to any of the 11 channels acceptable in North America and Europe. Every computer taking part in the IBSS must use the same channel. After you've set up an independent network, the system adds a new section titled "Computer to Computer Networks" to the drop-down list, as shown in Figure B-6. The AirPort status icon also changes to a computer in the pie wedge shape to indicate that the network is an IBSS rather than an infrastructure network.

Configuration with the System Preferences application

If you move between different ESSs, you can create a "location" for each and use this to configure the ESS/password pair, which you can then pick from a menu as you move to a different location. You can also preconfigure an ESS/password pair if you're not currently on the network for which you are setting up.

The System Preferences application allows you to configure many system attributes, including those that are network-related. Figure B-7 shows the network panel of the

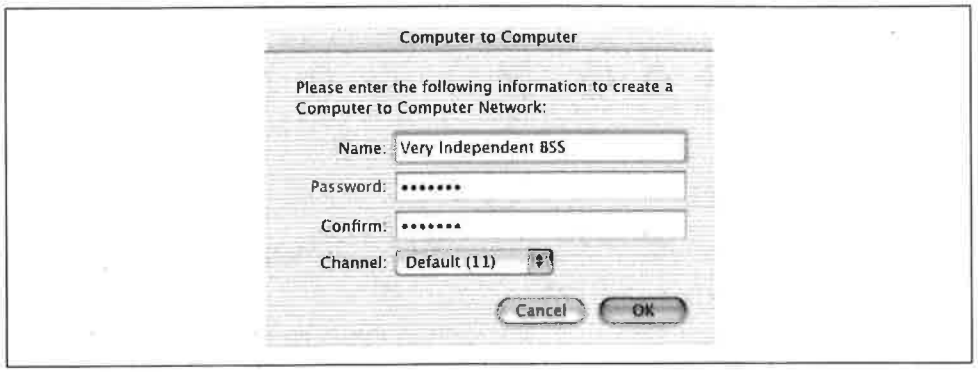

Figure B-5. IBSS parameter setup

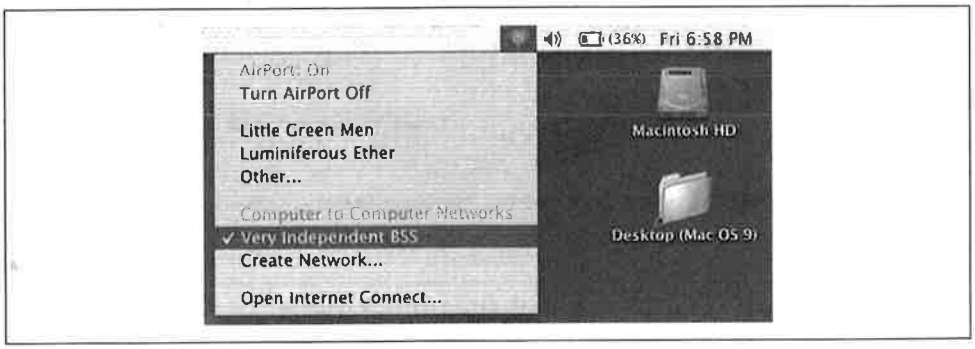

Figure B-6. Network preferences panel of the System Preferences application

System Preferences application. The Show pop-up list can be set to any of the network interfaces in the system. Naturally, when set to AirPort, it enables the fourth tab, as shown in the figure. The default tab is TCP/IP (Figure B-7), which can be set to configure interfaces manually or with the assistance of DHCP or BootP. When set to DHCP, as in the figure, the leased address is shown. Although the DHCP server on my network provided DNS servers, the server IP addresses are not shown. (They are, however, placed in */etc/resolv.conf*, as with any other common Unix system.)

The other network panel worthy of note is the AirPort tab (Figure B-8), which can set the network and password, though it involves more pointing and clicking than the menu associated with the AirPort status icon.

Monitoring the wireless interface

The wireless-interface status can be monitored with the Internet Connect application. The Internet Connect application can be launched from either the Applications folder on the hard disk or from the AirPort status icon. It can be used to display the signal strength of the nearest access point, as well as change the network with which the station is associated.

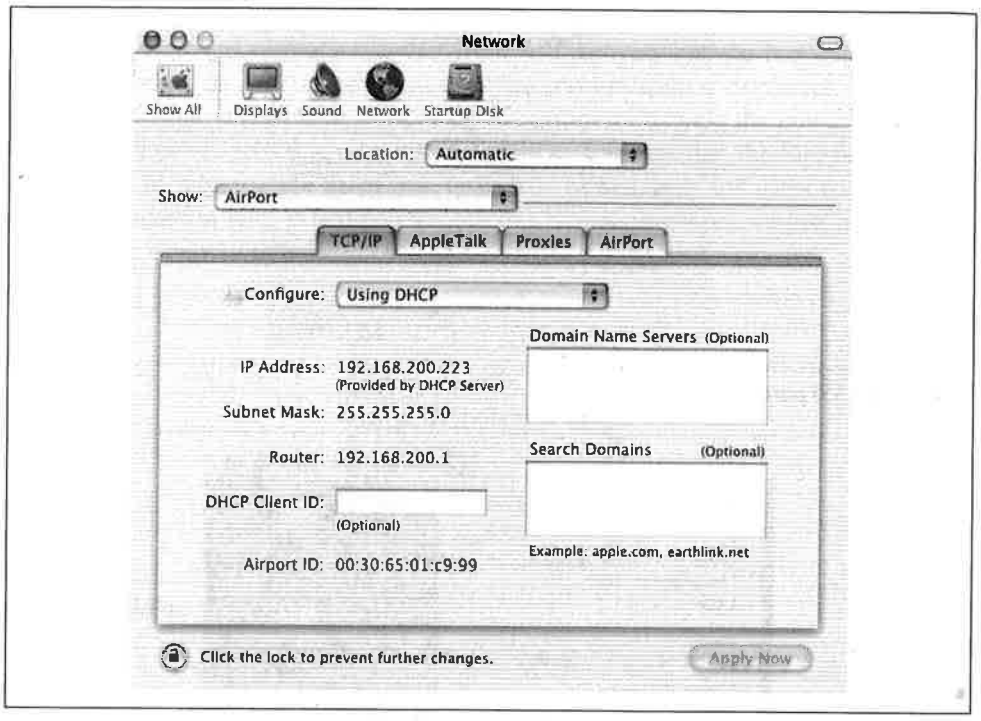

Figure B-7. TCP/IP preferences tab of the Network Preferences settings

The AirPort Base Station

The AirPort Base Station is a model of industrial design. It looks like a flying saucer. There are three lights: for power, wireless activity, and Ethernet activity. The most recent AirPort Base Station model has two Ethernet ports: a 56-KB modem port and a built-in wireless interface. On first-generation AirPort Base Stations, the wireless interface was a Lucent-branded card, complete with a Lucent label; second-generation AirPort Base Stations use Apple-branded AirPort cards.* The AirPort Base Station has a built-in antenna that is good, though not as good as some of the external antennas used with other access points.

In addition to bridging between a wireless and a wired network, the Airport Base Station is designed to connect all network users to the Internet via a cable modem, DSL

* Although the AirPort card is an Apple-branded product, a lookup of the FCC ID, IMRWLPC24H, indicates that it was made by Agere (Lucent). In first-generation AirPort Base Stations, the wireless interface was a Lucent Silver card (64-bit WEP). Several web pages show how to dismantle the AirPort Base Station to replace the Silver card with the Gold card (128-bit WEP) for improved security. Many of the same sites also show how to hook up an external antenna by drilling through the AirPort's plastic case. That is not likely to be as big a deal with the second-generation AirPort, since it uses the external antenna connector to use a larger antenna just under the top of the external case.

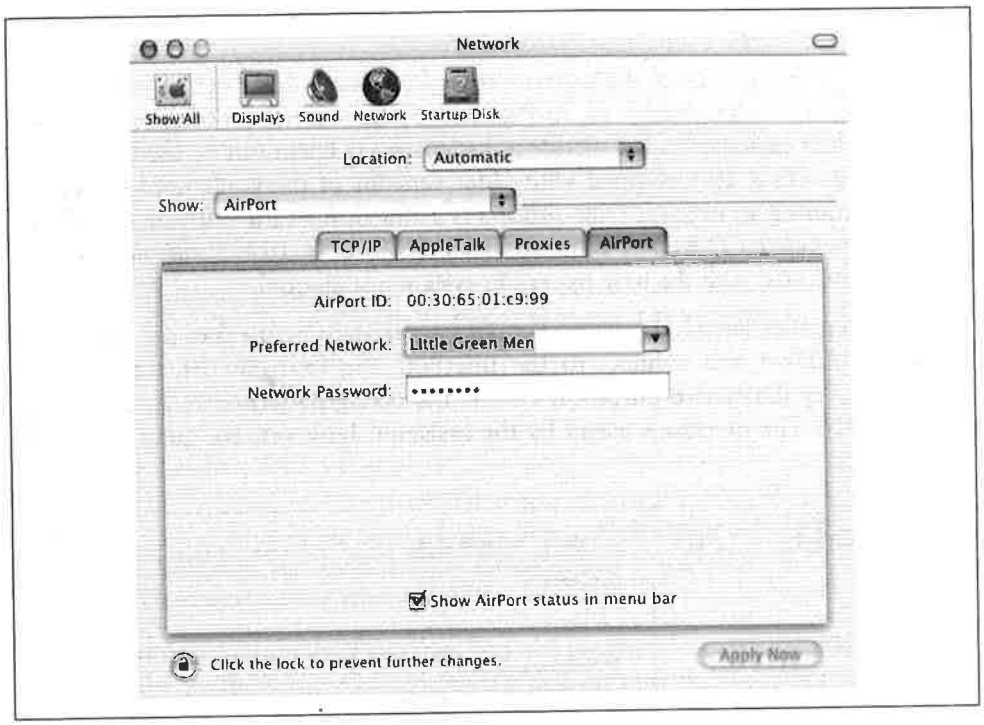

Figure B-8. AirPort preferences tab of the Network Preferences settings

modem, or a standard 56-KB modem. One of the Ethernet ports is a 10baseT port designed for a connection to a cable modem or DSL modem. The second Ethernet port is a Fast Ethernet port that can be used to connect to an existing LAN.

The AirPort Base Station was developed with assistance from Lucent, and it shows. The configuration of the two access points is quite similar to the configuration of Lucent access points. The AirPort Setup Assistant is used for basic out-of-the-box configuration. The AirPort Utility can then be used to set additional parameters. (Both applications were available only on Mac OS as I wrote this appendix, though Apple announced a Windows configuration application after the submission of the manuscript.) The Lucent access point hardware is commonly used, and there are configuration tools available for both Windows and Linux.

The low price of the AirPort has led many observers to classify it as a consumer device. Certainly, it has a great deal in common with the generic profile of a consumer access point: the price is relatively low, and it is a single unit with an integrated antenna. However, it is more capable than a number of consumer devices and support roaming users. From that standpoint, the AirPort would not be out of place in a small office.

First-Time Setup

The first-time setup is done with the same AirPort Setup Assistant application that configures new wireless interfaces. AirPort Base Stations fresh out of the shrink wrap broadcast their existence to the world so that a MacOS client can be used for configuration. Mac OS X 10.1 shipped with older versions of the configuration utilities, and they required an upgrade. The process is a straightforward software installation from the included CD-ROM. With the release of OS X 10.1.1, the new utilities should be included with the base operating-system installation.

The first step after launching the AirPort Setup Assistant is to select the method by which the AirPort will connect to the Internet. The four choices are shown in Figure B-9. For illustrative purposes, I show the configuration of an AirPort on an existing LAN. The questions asked by the assistant, however, are quite similar for each method.

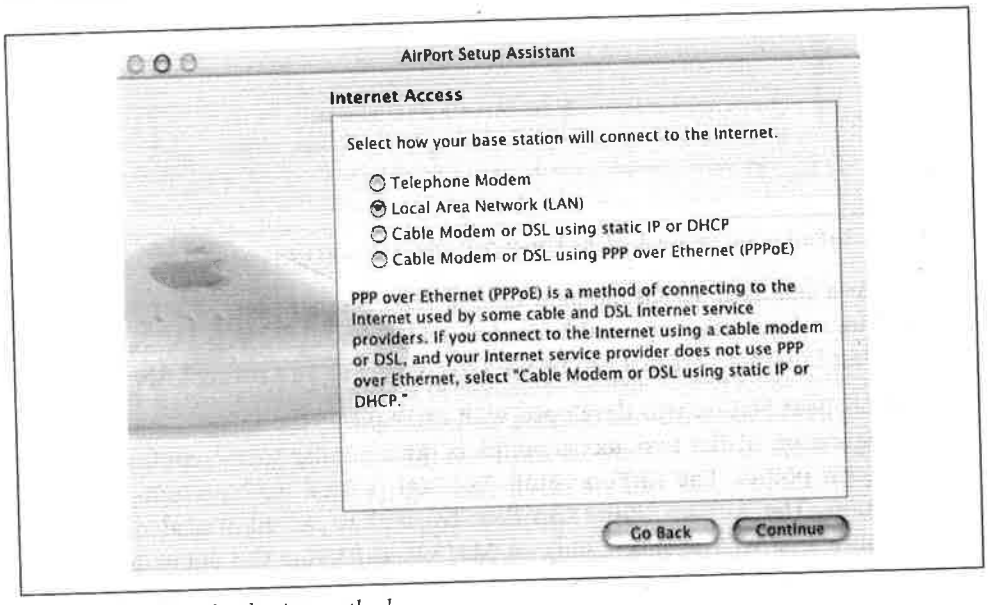

Figure B-9. Network selection methods

After selecting the connection type, the next step is to configure the Ethernet interface for the AirPort. It can be configured automatically with DHCP, or manually. Setting these parameters should be familiar to anyone who has set up a network.

After configuring the Ethernet interface, you proceed to configuring the wireless interface. You must supply the network name and a password. The password is an ASCII seed for the WEP key and can be left blank to run an open network that does not require WEP authentication. Figure B-10 shows the configuration screen for the network name and password.

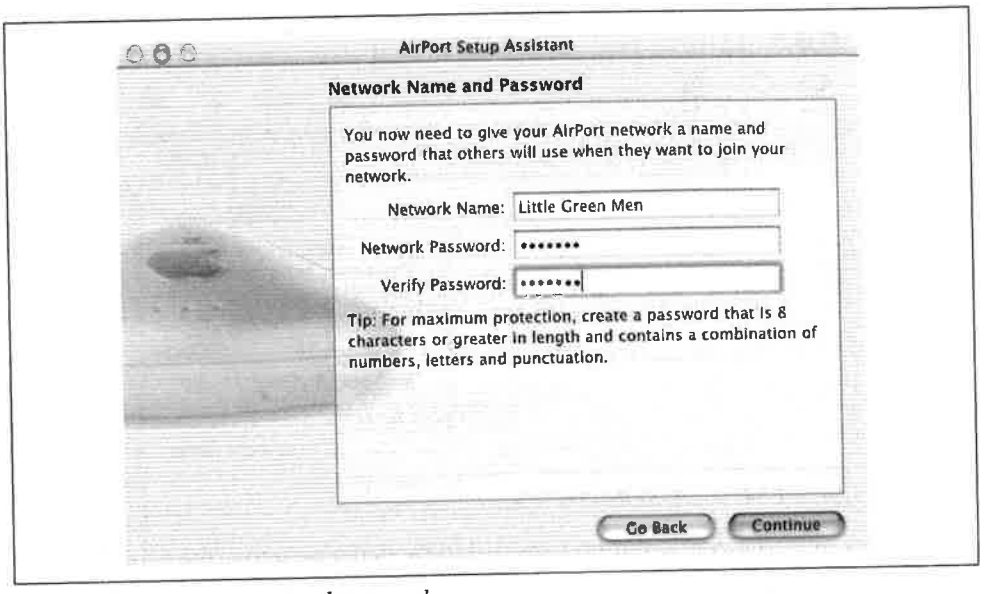

Figure B-10. Network name and password

The AirPort Base Station itself can use the same password as the network, or a different password. As the Setup Assistant itself notes, keeping the passwords separate is a practical requirement for network administrators who must maintain security separate from users. A dialog box lets you make this choice. If a separate password is configured, there is one additional screen in which to enter the base station password before the Setup Assistant terminates.

The Management Interface

Once the bootstrap configuration is done with the Setup Assistant, the AirPort Base Station is on the network and must be configured with the AirPort Utility (Figure B-11). This is a separate configuration utility that will feel vaguely familiar after seeing Lucent's AP Manager. When it is started, the AirPort Admin Utility searches all the AirPort base stations on the network and displays them in a list. Individual base stations can be selected for further configuration. When changes are made, the base station must be restarted for the changes to take effect. The **Other** button at the top allows configuration of any AirPort Base Station that the manager can send IP packets to. Far-away base stations may not appear on the browse list, but by clicking on **Other** and entering an IP address, the Manager can configure any base station to which it has IP connectivity.

Configuring the wireless interface

When a base station is selected for configuration, the configuration screen will pop up. Several tabs are used to group configuration information into logical subsets, and

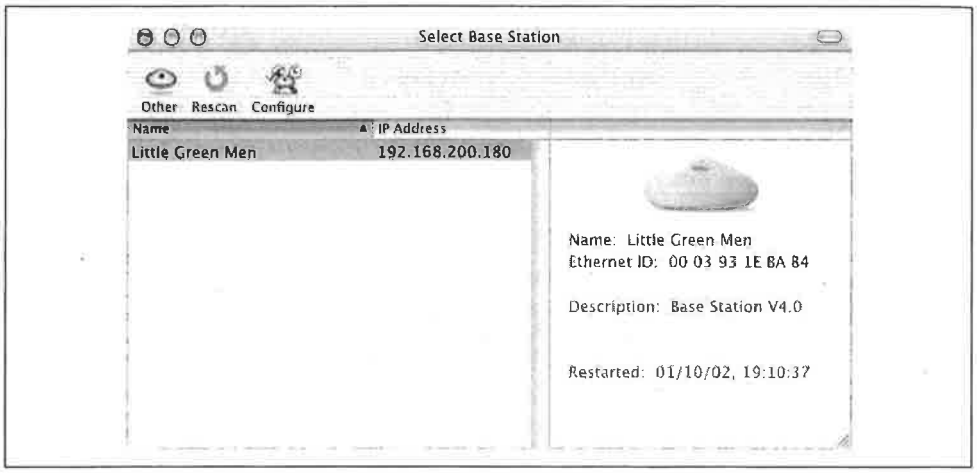

Figure B-11. AirPort Admin Utility main screen

the wireless interface configuration is available by default. Across the top, there are buttons to restart the access point, upload new firmware, return the base station to factory defaults, and change the password. (New firmware is distributed as part of the Admin Utility package.)

The AirPort configuration is shown in Figure B-12. The "AirPort Network" settings are comparable to the Lucent settings of the same name, with the exception of the WEP configuration. Only one password is supported by the AirPort Base Station.

AirPorts were designed to be compatible with other vendors' 802.11 equipment. Most other vendors, however, require that users enter WEP keys as hexadecimal strings. The Password icon at the top of the toolbar will print out the raw WEP key for use in other products. Figure B-13 shows the WEP key that results from entering an ASCII string of `Book Key` as the text seed.*

Configuration of the WAN interface

The Internet configuration tab presents no surprises. When set to Ethernet, it uses the "WAN" Ethernet port to connect to the outside world. You can configure the standard network settings (IP address, network mask, DNS server, router) manually, or you can select automatic configuration via DHCP. The AirPort is compatible with cable modems that use the Point to Point Protocol over Ethernet (PPPoE) and DSL modems. PPPoE configuration requires a username and password supplied by your ISP. Modem access can be set up using either a generic dial-in or a script provided by Apple to make the AirPort Base Station compatible with America Online.

* Chally Microsolutions (*http://www.chally.net*) distributes a tool called WEP Key Maker that will generate a WEP key of a specified length from a long pass phrase.

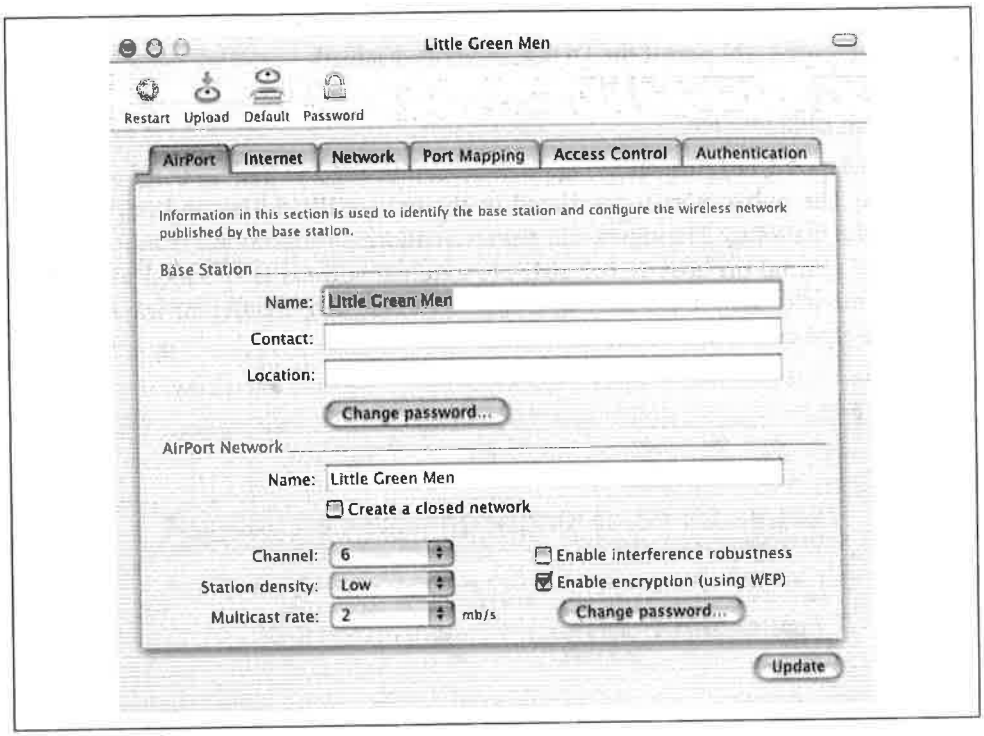

Figure B-12. Wireless interface configuration

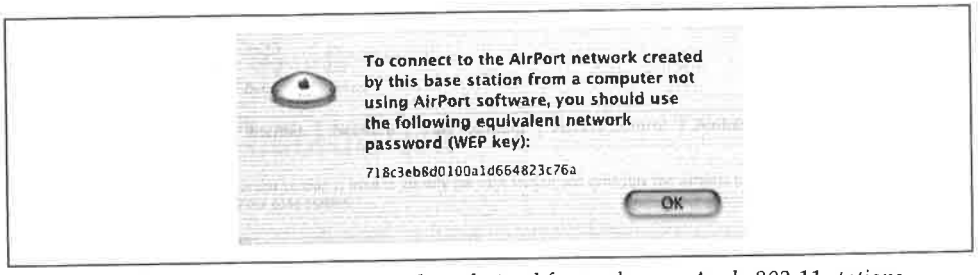

Figure B-13. Printing out the WEP key in hexadecimal for use by non-Apple 802.11 stations

Configuration of the LAN interface

The AirPort can be used as the central connection device in a home network by connecting the 10-Mbps Ethernet port to a broadband connection and using the 100-Mbps Ethernet port to connect other LAN stations. Client computers can be statically addressed by the network administrator, or the built-in DHCP server can be turned on and assigned a range of addresses to assign to clients. When NAT is used to hide several computers behind one IP address, the address specified in the Internet properties is used as the public address. The wired LAN interface is given the private address 10.0.1.1, and DHCP is used to lease out addresses from 10.0.1.2 to 10.0.1.200 (in this case, you can't select the range of addresses for the DHCP

server to give out). The AirPort base station can be connected to wired networks by its Fast Ethernet LAN port if the DHCP server is disabled.

Inbound NAT configuration

The Port Mapping tab, shown in Figure B-14, can be used to add inbound static port mappings. The public port is translated to the private IP address and port number listed in the mapping. The figure illustrates an address translation for inbound web services to port 80 on host 10.0.1.201. No external address is specified in the Port Mapping tab because the external address can be assigned in different ways depending on the type of Internet connection used.

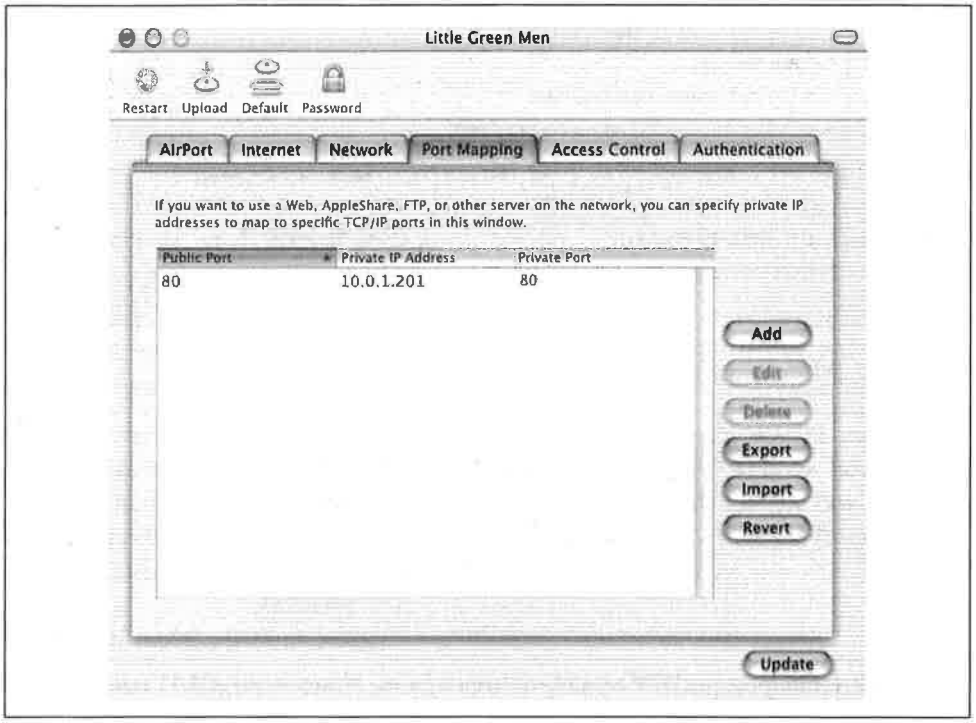

Figure B-14. Port Mapping tab

Access control

Like most other products, the AirPort Base Station supports filtering by client MAC address. The Access Control tab lets you identify clients by their AirPort ID (MAC address) and add them to a list of allowed clients, together with a description.

Authentication

The AirPort allows a RADIUS server, which can provide an external authentication mechanism for MAC addresses. When authentication of a client MAC is required,

the base station will pass the request on to the RADIUS servers defined in the tab shown in Figure B-15.

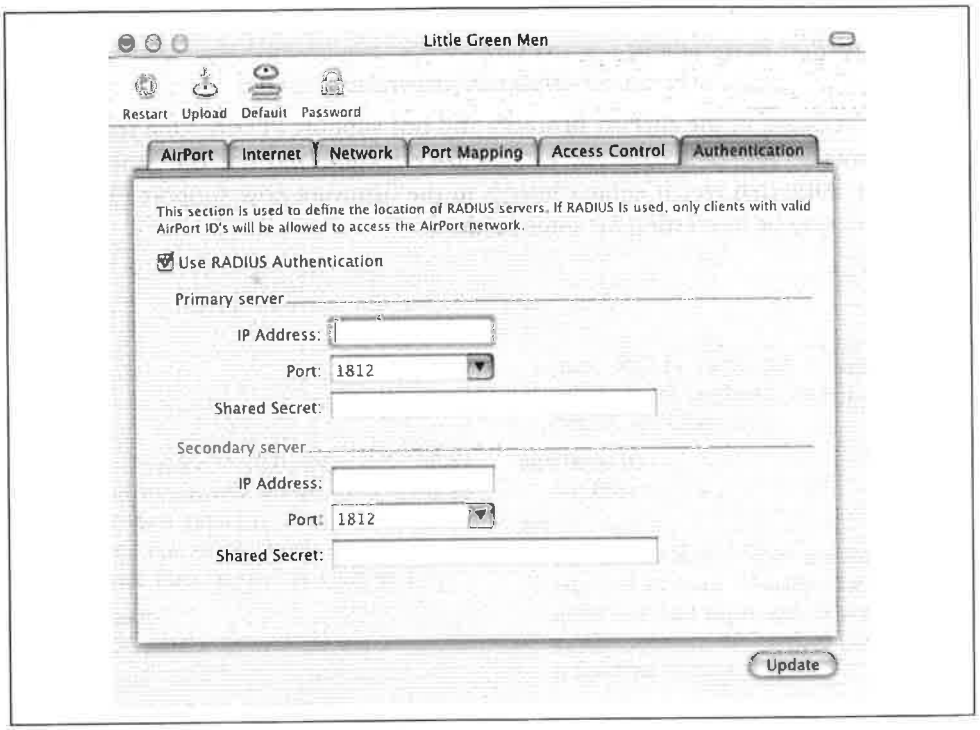

Figure B-15. RADIUS authentication tab

Links to More Information

Official sites

http://www.apple.com/airport/ (sales and marketing)

http://www.info.apple.com/usen/airport/ (support; registration required)

Repairing first-generation AirPorts

http://www.vonwentzel.net/ABS/Repair.html

First-generation AirPort Base Stations tend to fail due to capacitor failure. This site explains the problem and shows how to repair the base station, though the procedure almost certainly voids the warranty.

AirPort hardware hacking

http://www.msrl.com/airport-gold/

Older AirPort base stations use Lucent cards. By changing the card from the stock Silver card to a Gold card, longer WEP keys can be used.

http://jim.chronomedia.com/Airportmod/

First-generation AirPort Base stations run hot. This site shows how to add a cooling fan to the case.

Linux on AirPort Base Stations

http://www-hft.ee.tu-berlin.de/~strauman/airport/airport.html

Early versions of the AirPort firmware did not support PPPoE. One response to this was to run Linux on the AirPort Base Station and use the Linux PPPoE driver. Although recent enhancements to the firmware now support PPPoE, the project may be interesting for some readers.

Glossary

access point
See AP.

ACK
Abbreviation for "Acknowledgment." ACKs are used extensively in 802.11 to provide reliable data transfers over an unreliable medium. For more details, see "Contention-Based Data Service" in Chapter 3.

Acknowledgment
See ACK.

ad hoc
A network characterized by temporary, short-lived relationships between nodes. See also IBSS.

AID
Association Identifier. A number that identifies data structures in an access point allocated for a specific mobile node.

AP
Access Point. Bridge-like device that attaches wireless 802.11 stations to a wired backbone network. For more information on the general structure of an access point, see Chapter 14.

ASN
Abstract Syntax Notation. The formal description of the grammar used to write MIB files.

association identifier
See AID.

ATIM
Announcement Traffic Indication Message. ATIMs are used in ad hoc (independent) 802.11 networks to announce the existence of buffered frames. For more details, see Chapter 7.

basic service set
See BSS.

BER
Bit Error Rate. The number of bits received in error. Usually, the number is quite low and expressed as a ratio in scientific notation. 10^{-2} means one bit in 100 is received in error.

BPSK
Binary Phase Shift Keying. A modulation method that encodes bits as phase shifts. One of two phase shifts can be selected to encode a single bit.

BSS
Basic Service Set. The building block of 802.11 networks. A BSS is a set of stations that are logically associated with each other.

BSSID
Basic Service Set Identifier. A 48-bit identifier used by all stations in a BSS in frame headers.

CCITT
Comité Consultatif International Télégraphique et Téléphonique. A UN body responsible for telephone standardization. Due to a reorganization, it is now called the International Telecommunication Union-Telecommunication Standardization Sector (ITU-T).

CCK
Complementary Code Keying. A modulation scheme that transforms data blocks into complex codes and is capable of encoding several bits per block.

CF
Contention Free. Services that do not involve contention for the medium are contention-free services. Such services are implemented by a Point Coordinator (PC) through the use of the Point Coordination Function (PCF). Contention-free services are not widely implemented.

CFP
Contention-Free Period. Even when 802.11 provides contention-free services, some contention-based access to the wireless medium is allowed. Periods controlled by a central authority are called contention-free periods (CFP).

CRC
Cyclic Redundancy Check. A mathematical checksum that can be used to detect data corruption in transmitted frames.

CSMA
Carrier Sense Multiple Access. A "listen before talk" scheme used to mediate the access to a transmission resource. All stations are allowed to access the resource (multiple access) but are required to make sure the resource is not in use before transmitting (carrier sense).

CSMA/CA
Carrier Sense Multiple Access with Collision Avoidance. A CSMA method that tries to avoid simultaneous access (*collsions*) by deferring access to the medium. 802.11 and AppleTalk's LocalTalk are two protocols that use CSMA/CA.

CTS
Clear to Send. The frame type used to acknowledge receipt of a Request to Send and the second component used in the RTS-CTS clearing exchange used to prevent interference from hidden nodes.

DA
Destination Address. The MAC address of the station the frame should be processed by. Frequently, the destination address is the receiver address. In infrastructure networks, however, frames bridged from the wireless side to the wired side will have a destination address on the wired network and a receiver address of the wireless interface in the access point.

DBPSK
Differential Binary Phase Shift Keying. A modulation method in which bits are encoded as phase shift differences between successive symbol periods. Two phase shifts are possible for an encoding rate of one data bit per symbol.

DCF
Distributed Coordination Function. The rules for contention-based access to the wireless medium in 802.11. The DCF is based on exponentially increasing backoffs in the presence of contention as well as rules for deferring access, frame acknowledgment, and when certain types of frame exchanges or fragmentation may be required.

DHCP
Dynamic Host Configuration Protocol. An IETF standard used by network administrators to automatically configure hosts. Hosts needing configuration information may broadcast a request that is responded to by a DHCP server. DHCP was the Internet community's admission that the Internet was growing so fast that network administrators had lost control over what was plugged into networks.

DIFS
Distributed Inter-Frame Space. The interframe space used to separate atomic exchanges in contention-based services. See also DCF.

distributed coordination function
See DCF.

distributed inter-frame space
See DIFS.

DQPSK
Differential Quadrature Phase Shift Keying. A modulation method in which bits are encoded as phase shift differences

between successive symbol periods. Four phase shifts are possible for an encoding rate of two data bits per symbol.

DS
Distribution System. The set of services that connects access points together. Logically composed of the wired backbone network plus the bridging functions in most commercial access points. See Figure 2-6

DSSS
Direct-Sequence Spread Spectrum. A transmission technique that spreads a signal over a wide frequency band for transmission. At the receiver, the widespread signal is correlated into a stronger signal; meanwhile, any narrowband noise is spread widely. Most of the 802.11-installed base at 2 Mbps and 11 Mbps is composed of direct-sequence interfaces.

DTIM
Delivery Traffic Indication Map. Beacon frames may contain the DTIM element, which is used to indicate that broadcast and multicast frames buffered by the access point will be delivered shortly.

EIFS
Extended Inter-Frame Space. The longest of the four inter-frame spaces, the EIFS is used when there has been an error in transmission.

EIRP
Effective Isotropic Radiated Power. An antenna system will have a footprint over which the radio waves are distributed. The power inside the footprint is called the effective isotropic radiated power.

ERP
Effective Radiated Power. Used to describe the strength of radio waves transmitted by an antenna.

ESS
Extended Service Set. A logical collection of access points all tied together. Link-layer roaming is possible throughout an ESS, provided all the stations are configured to recognize each other.

ETSI
European Telecommunications Standards Institute. ETSI is a multinational standardization body with regulatory and standardization authority over much of Europe. GSM standardization took place under the auspices of ETSI. ETSI has taken the lead role in standardizing a wireless LAN technology competing with 802.11 called the High Performance Radio LAN (HIPERLAN).

extended inter-frame space
See EIFS.

FCC
Federal Communications Commission. The regulatory agency for the United States. The FCC Rules in Title 47 of the Code of Federal Regulations govern telecommunications in the United States. Wireless LANs must comply with Part 15 of the FCC rules, which are written specifically for RF devices.

FCS
Frame Check Sequence. A checksum appended to frames on IEEE 802 networks to detect corruption. If the receiver calculates a different FCS than the FCS in the frame, it is assumed to have been corrupted in transit and is discarded.

FH
Frequency Hopping. See FHSS.

FHSS
Frequency Hopping Spread Spectrum. A technique that uses a time-varying narrowband signal to spread RF energy over a wide band.

GFSK
Gaussian Frequency Shift Keying. A modulation technique that encodes data based on the frequency of the carrier signal during the symbol time. GFSK is relatively immune to analog noise because most analog noise is amplitude-modulated.

HR/DSSS
High-Rate Direct-Sequence Spread Spectrum. The abbreviation for signals transmitted by 802.11b equipment. Although similar to the earlier 2-Mbps transmissions

in many respects, advanced encoding enables a higher data rate.

IAPP
Inter-Access Point Protocol. The protocol used between access points to enable roaming. In late 2001, each vendor used a proprietary IAPP, though work on a standardized IAPP was underway.

IBSS
Independent Basic Service Set. An 802.11 network without an access point. Some vendors refer to IBSSs as ad hoc networks; see also ad hoc.

ICV
Integrity Check Value. The checksum calculated over a frame before encryption by WEP. The ICV is designed to protect a frame against tampering by allowing a receiver to detect alterations to the frame. Unfortunately, WEP uses a flawed algorithm to generate the ICV, which robs WEP of a great deal of tamper-resistance.

IEEE
Institute of Electrical and Electronics Engineers. The professional body that has standardized the ubiquitous IEEE 802 networks.

IR
Infrared. Light with a longer wavelength and lower frequency than visible red light. The wavelength of red light is approximately 700 nm.

ISI
Inter-Symbol Interference. Because of delays over multiple paths, transmitted symbols may interfere with each other and cause corruption. Guarding against ISI is a major consideration for wireless LANs, especially those based on OFDM.

ISM
Industrial, Scientific, and Medical. Part 15 of the FCC Rules sets aside certain frequency bands in the United States for use by unlicensed Industrial, Scientific, and Medical equipment. The 2.4-GHz ISM band was initially set aside for microwave ovens so that home users of microwave ovens would not be required to go through the burdensome FCC licensing process simply to reheat leftover food quickly. Because it is unlicensed, though, many devices operate in the band, including 802.11 wireless LANs.

ITU
International Telecommunications Union. The successor to the CCITT. Technically speaking, the ITU issues recommendations, not regulations or standards. However, many countries give ITU recommendations the force of law.

IV
Initialization Vector. Generally used as a term for exposed keying material in cryptographic headers; most often used with block ciphers. WEP exposes 24 bits of the secret key to the world in the frame header, even though WEP is based on a stream cipher.

LLC
Logical Link Control. An IEEE specification that allows further protocol multiplexing over Ethernet. 802.11 frames carry LLC-encapsulated data units.

MAC
Medium Access Control. The function in IEEE networks that arbitrates use of the network capacity and determines which stations are allowed to use the medium for transmission.

MIB
Management Information Base. An ASN specification of the operational and configuration parameters of a device; frequently used with SNMP or other network management systems.

MPDU
MAC Protocol Data Unit. A fancy name for frame. The MPDU does not, however, include PLCP headers.

MSDU
MAC Service Data Unit. The data accepted by the MAC for delivery to another MAC on the network. MSDUs are composed of higher-level data only. For example, an 802.11 management frame does not contain an MSDU.

NAV
Network Allocation Vector. The NAV is used to implement the virtual carrier sensing function. Stations will defer access to the medium if it is busy. For robustness, 802.11 includes two carrier-sensing functions. One is a *physical* function, which is based on energy thresholds, whether a station is decoding a legal 802.11 signal, and similar things that require a physical measurement. The second function is a *virtual* carrier sense, which is based on the NAV. Most frames include a nonzero number in the NAV field, which is used to ask all stations to politely defer from accessing the medium for a certain number of microseconds after the current frame is transmitted. Any receiving stations will process the NAV and defer access, which prevents collisions. For more detail on how the NAV is used, see "Contention-Based Data Service" in Chapter 3.

OFDM
Orthogonal Frequency Division Multiplexing. A technique that splits a wide frequency band into a number of narrow frequency bands and inverse multiplexes data across the subchannels. Both 802.11a and the forthcoming 802.11g standards are based on OFDM.

OSI
Open Systems Interconnection. A baroque compendium of networking standards that was never implemented because IP networks actually existed.

PBCC
Packet Binary Convolution Coding. An alternative method of encoding data in 802.11b networks that has not been widely implemented. PBCC was also proposed for consideration for 20+ Mbps networks, but was rejected.

PC
Point Coordinator. A function in the access point responsible for central coordination of access to the radio medium during contention-free service.

PCF
Point Coordination Function. The set of rules that provides for centrally coordinated access to the medium by the access point.

PCMCIA
Personal Computer Memory Card International Association. An industry group that standardized the ubiquitous "PCMCIA card" form factor and made it possible to connect a wide variety of peripherals to notebook computers. 802.11 interfaces are available almost exclusively in the PCMCIA form factor. Also expanded humorously as People Who Can't Manage Computer Industry Acronyms because of its unwieldy length and pronunciation.

PDU
See protocol data unit.

PER
Packet Error Rate. Like the bit error rate, but measured as a fraction of packets with errors.

PHY
Common IEEE abbreviation for the physical layer.

physical-layer convergence procedure
The upper component of the PHY in 802.11 networks. Each PHY has its own PLCP, which provides auxiliary framing to the MAC.

PIFS
PCF Inter-Frame space. During contention-free service, any station is free to transmit if the medium is idle for the duration of one PCF inter-frame space.

PLCP
See physical-layer convergence procedure.

PMD
Physical Medium Dependent. The lower component of the MAC, responsible for transmitting RF signals to other 802.11 stations.

PPDU
PLCP Protocol Data Unit. The complete PLCP frame, including PLCP headers,

MAC headers, the MAC data field, and the MAC and PLCP trailers.

protocol data unit
Layers communicate with each other using protocol data units. For example, the IP protocol data unit is the familiar IP packet. IP implementations communicate with each other using IP packets. See also service data unit.

PS
Power Save. Used as a generic prefix for power-saving operations in 802.11.

PSDU
PLCP Service Data Unit. The data the PLCP is responsible for delivering, i.e., one MAC frame with headers.

PSK
Phase Shift Keying. A method of transmitting data based on phase shifts in the transmitted carrier wave.

QPSK
Quadrature Phase Shift Keying. A modulation method that encodes bits as phase shifts. One of four phase shifts can be selected to encode two bits.

RA
Receiver Address. MAC address of the station that will receive the frame. The RA may also be the destination address of a frame, but not always. In infrastructure networks, for example, a frame destined for the distribution system is received by an access point.

RC4
A proprietary cipher algorithm developed by RSA Data Security and licensed for a great deal of money. Also used as the basis for WEP and prevents open source WEP implementations from existing because of the fear of lawsuits by RSA.

RF
Radio Frequency. Used as an adjective to indicate that something pertains to the radio interface ("RF modulator," "RF energy," and so on).

RTS
Request to Send. The frame type used to begin the RTS-CTS clearing exchange. RTS frames are used when the frame that will be transmitted is larger than the RTS threshold.

SA
Source Address; as disinct from TA. Station that generated the frame. Different when frame originates on the distrbution system and goes to the wireless segment.

SDU
See service data unit.

Service Data Unit
When a protocol layer receives data from the next highest layer, it is sending a service data unit. For example, an IP service data unit can be composed of the data in the TCP segment plus the TCP header. Protocol layers access service data units, add the appropriate header, and push them down to the next layer. See also protocol data unit.

SFD
Start of Frame Delimiter. The component of the frame header that indicates when synchronization has concluded and the actual frame is about to start.

SIFS
Short Inter-Frame Space. The shortest of the four inter-frame spaces. The SIFS is used between frames in an atomic frame exchange.

SSID
Service Set Identity. A string used to identify a service set. Typically, the SSID is a recognizable character string for the benefit of users.

SYNC
Short for Synchronize. Bits transmitted by the PLCP to allow senders and receivers to synchronize bit timers.

TA
Transmitter Address. Station that actually put the frame in the air. Often the access point in infrastructure networks.

TIM
Traffic Indication Map. A field transmitted in Beacon frames used to inform associated stations that the access point has buffered. Bits are used to indicate both

buffered unicast frames for each associated station as well as the presence of buffered multicast frames.

WEP
: Wired Equivalent Privacy. Derided as Wiretap Equivalence Protocol by its critics. A standard for ciphering individual data frames. It was intended to provide minimal privacy and has succeeded in this respect. In August 2001, WEP was soundly defeated, and public code was released.

Wi-Fi and Wi-Fi5
: The Wireless Ethernet Compatibility Alliance started the Wi-Fi ("wireless fidelity") certification program to ensure that equipment claiming 802.11 compliance was genuinely interoperable. Wi-Fi–certified equipment has demonstrated standards compliance in an interoperability lab. Originally, the term was applied to devices that complied with 802.11b (11-Mbps HR/DSSS). The newer term, Wi-Fi5, is applied to 802.11a (54-Mbps OFDM) equipment that passes a similar certification test suite.

Index

Numbers

A

We'd like to hear your suggestions for improving our indexes. Send email to *index@oreilly.com*.

B

C

D

E

F

G

H

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

About the Author

Matthew S. Gast is a renaissance technologist. In addition to his demonstrated expertise on a variety of network technologies, he is relentlessly inquisitive about the interconnected and interdependent world around him. After graduating from college, his interests in routing, security, and cryptography pulled him towards Silicon Valley to participate in scaling the mountainous network engineering challenge called the Internet. In addition to his technology interests, Matthew is a voracious reader on science and economics and a lifelong supporter of the scientific method.

Matthew is also a Registered Patent Agent before the United States Patent and Trademark Office. Patent agents assist in the drafting and prosecution of patent applications, which has been called the most demanding task in the United States legal system by the Supreme Court. Matthew has co-written two patent applications, one of which was for his own invention.

Colophon

Our look is the result of reader comments, our own experimentation, and feedback from distribution channels. Distinctive covers complement our distinctive approach to technical topics, breathing personality and life into potentially dry subjects.

The animal on the cover of *802.11 Wireless Netowrks: The Definitive Guide* is a horseshoe bat (*Rhinolophus hipposideros*). This rare and globally endangered species is the smallest of the European horseshoe bats; they typically weigh only 4–10 grams and have a wingspan of 19–25 centimeters. Horseshoe bats get their name from the horseshoe-shaped, leaflike plate of skin around the nose. This nose-leaf helps modify and direct the ultrasonic sounds they emit through their nostrils (a method of sensory perception known as echolocation) to orient themselves to their surroundings, detect obstacles, communicate with each other, and find food. Bats' echolocation systems are so accurate that they can detect insects the size of gnats and objects as fine as a human hair.

Lesser horseshoe bats are found in a variety of habitats, ranging from the British Isles to the Arabian Peninsula and Central Asia, and from Morocco to Sudan. The lesser horseshoe bat was originally a cave-roosting bat, but many summer maternity colonies now occupy the roofs of old rural houses and farm buildings. These bats also sometimes roost in hedgerows and hollow trees. Maternity colonies of 30 to 70 are normal, but roosting mothers have been known to form colonies of as many as 200 bats. Lesser horseshoe bats hibernate, sometimes in large groups, from October until late April or early May. Their winter roosts are usually underground, in caves or tunnels. They hang by their feet with their wings wrapped around their bodies, often in open and exposed positions but rarely in large clusters.

Matt Hutchinson was the production editor and proofreader, and Leanne Soylemez was the copyeditor for *802.11 Wireless Networks: The Definitive Guide*. Sarah Sherman and Darren Kelly provided quality control. Lucie Haskins wrote the index.

Ellie Volckhausen designed the cover of this book, based on a series design by Edie Freedman. The cover image is a 19th-century engraving from the Dover Pictorial Archive. Emma Colby produced the cover layout with QuarkXPress 4.1 using Adobe's ITC Garamond font.

Melanie Wang designed the interior layout, based on a series design by David Futato. Neil Walls converted the files from Microsoft Word to FrameMaker 5.5.6 using tools created by Mike Sierra. The text font is Linotype Birka; the heading font is Adobe Myriad Condensed; and the code font is LucasFont's TheSans Mono Condensed. The illustrations that appear in the book were produced by Robert Romano and Jessamyn Read using Macromedia FreeHand 9 and Adobe Photoshop 6. The tip and warning icons were drawn by Christopher Bing. This colophon was written by Rachel Wheeler.

Network Administration

O'REILLY®

802.11 Wireless Networks: The Definitive Guide

Using a wireless network is a liberating experience. But underneath the experience lies a complex protocol, and even more complex issues arise when your data isn't limited to traveling on physical wires. How do you structure your network so mobile users can move around effectively? How do you extend wireless coverage so it's available everywhere you need it? What kinds of security issues do wireless networks raise? How do you tune your network for optimal performance? How do you provide enough capacity to support the users you expect initially, and how do you deal with the problems that arise as more users join the network?

802.11 Wireless Networks: The Definitive Guide discusses all these issues, and more. This book is for the serious system or network administrator who is responsible for deploying or maintaining a wireless network. It discusses how the 802.11 protocols work, with a view towards understanding which options are available and troubleshooting problems that arise. It contains an extensive discussion of wireless security issues, including the problems with the WEP standard and a look at the 802.1x security standard. Since network monitoring is essential to any serious network administrator, and commercial packet sniffers for wireless applications are scarce and expensive, this book shows how to create a wireless packet sniffer from a Linux system and open source software.

In addition to the current 802.11b standard, *802.11 Wireless Networks: The Definitive Guide* also looks forward to the newest developments in wireless networking, including the two new 54-Mbps standards: 802.11a and 802.11g. It also surveys other efforts moving through the standards track, including work to facilitate mobility between access points, quality of service, spectrum management, and power control.

Finally, *802.11 Wireless Networks: The Definitive Guide* shows you how to configure wireless cards and Linux, Windows, and OS X systems, and how to work with access points. Few books in any field combine the theory you need to know with the practical experience and advice you need to get things working. *802.11 Wireless Networks: The Definitive Guide* is one of those books. If you are responsible for a wireless network, you need this book.

ISBN 0-596-00183-5
US $44.95
CAN $69.95
90000
9 780596 001834
6 36920 00183 6

Visit O'Reilly on the Web at *www.oreilly.com*