Characteristics of the HR/DSSS PHY

Table 10-13 shows the values of a number of parameters in the HR/DSSS PHY. Like the DS PHY, the HR/DSSS PHY has a number of parameters that can be adjusted to compensate for delays in any part of a real system.

Table 10-13. HR/DSSS PHY parameters

Parameter	Value	Notes
Maximum MAC frame length	4,095 bytes	
Slot time	20 µs	
SIFS time	10 µs	The SIFS is used to derive the value of the other interframe spaces (DIFS, PIFS, and EIFS).
Contention window size	31 to 1,023 slots	
Preamble duration	144 µs	Preamble symbols are transmitted at 1 MHz, so a symbol takes 1 s to transmit; 96 bits require 96 symbol times.
PLCP header duration	48 bits	The PLCP header transmission time depends on whether the short preamble is used.

One other item of note is that the total aggregate throughput of all HR/DSSS networks in an area is still lower than the total aggregate throughput of all nonoverlapping frequency-hopping networks in an area. The total aggregate throughput is a function of the number of nonoverlapping channels. In North America and most of Europe, three HR/DSSS networks can be deployed in an area at once. If each network is run at the optional 11-Mbps rate, the area can have a total of 33-Mbps throughput, which is less than the frequency-hopping total aggregate throughput.

Exhibit 1019
Page 201 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

CHAPTER 11

802.11a: 5-GHz OFDM PHY

Many of the wireless devices that are currently on the market use the 2.4-GHz ISM band, which is rapidly becoming crowded. In an attempt to attain higher data rates and avoid overcrowding, the 802.11 working group is looking at the unlicensed bands around 5 GHz. In the U.S., the 5-GHz bands that are reserved for unlicensed use are designated as the Unlicensed National Information Infrastructure (U-NII). The U-NII bands provide more spectrum space than the 2.4-GHz bands and are much less heavily used; there are very few devices on the market that operate in these bands. The 802.11a working group is responsible for developing physical layers for high-rate wireless service on the 5-GHz bands.

802.11a hardware hit the market in late 2001. There are three main chipset vendors: Intersil, the developer of the widely used PRISM chipsets for the 2.4-GHz ISM band; the ever acquisitive Cisco, through its acquisition of chip-maker Radiata in late 2000; and Atheros Communications, a start-up founded by several notable Stanford researchers. 802.11a products hit the market about the time this book went to press, so I have not evaluated any vendor claims about increased range, throughput, or performance. Most of the products on the market look very much like the current 802.11b products: many are PC Cards and have similar configuration and installation routines.

In spite of its many advantages, 802.11a does not look like a sure-fire commercial success. Higher frequencies have higher path losses; some observers estimate that access points will have to be deployed much more densely than in an 802.11b network; some observers have estimated that one and a half times as many (or more) access points may be required. Most of the testing on 802.11a has been conducted by vendors in offices dominated by cubicles. How 11a systems will fare in other environments can be answered only by more deployment and testing. Higher-frequency RF systems require higher power, which may mean that 802.11a cards are not well-suited for mobile battery-powered systems. Many of the questions swirling around 802.11a can be answered only by the delivery of commercial products to the marketplace in

Exhibit 1019
Page 202 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

volume. The 802.11a standard as it currently stands is only for the U.S., but a task group is working on extending the standard to other regulatory domains.

This chapter begins with a qualitative introduction to the basis of OFDM. When all the mathematical formalism is stripped away, OFDM is a method for chopping a large frequency channel into a number of subchannels. The subchannels are then used in parallel for higher throughput. I anticipate that many readers will skip the first section, either because they are already familiar with OFDM or are interested only in how the frequency bands are used and how the PCLP wraps up frames for transmission.

Orthogonal Frequency Division Multiplexing (OFDM)

802.11a is based on *orthogonal frequency division multiplexing* (OFDM). OFDM is not a new technique. Most of the fundamental work was done in the late 1960s, and U.S. patent number 3,488,445 was issued in January 1970. Recent DSL work (HDSL, VDSL, and ADSL) and wireless data applications have rekindled interest in OFDM, especially now that better signal-processing techniques make it more practical.* OFDM does, however, differ from other emerging encoding techniques such as code division multiple access (CDMA) in its approach. CDMA uses complex mathematical transforms to put multiple transmissions onto a single carrier; OFDM encodes a single transmission into multiple subcarriers. The mathematics underlying the code division in CDMA is far more complicated than in OFDM.

OFDM devices use one wide frequency channel by breaking it up into several component subchannels. Each subchannel is used to transmit data. All the "slow" subchannels are then multiplexed into one "fast" combined channel.

Carrier Multiplexing

When network managers solicit user input on network build-outs, one of the most common demands is for more speed. The hunger for increased data transmissions has driven a host of technologies to increase speed. OFDM takes a qualitatively similar approach to Multilink PPP: when one link isn't enough, use several in parallel.

OFDM is closely related to plain old frequency division multiplexing (FDM). Both divide the available bandwidth into slices called *carriers* or *subcarriers* and make those carriers available as distinct channels for data transmission. OFDM boosts

* The lack of interest in OFDM means that references on it are sparse. Readers interested in the mathematical background that is omitted in this chapter should consult *OFDM for Wireless Multimedia Applications* by Richard van Nee and Ramjee Prasad (Artech House, 2000).

Exhibit 1019
Page 203 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

throughput by using several subcarriers in parallel and multiplexing data over the set of subcarriers.

Traditional FDM was widely used by first-generation mobile telephones as a method for radio channel allocation. Each user was given an exclusive channel, and guard bands were used to ensure that spectral leakage from one user did not cause problems for users of adjacent channels. Figure 11-1 illustrates the traditional FDM approach.

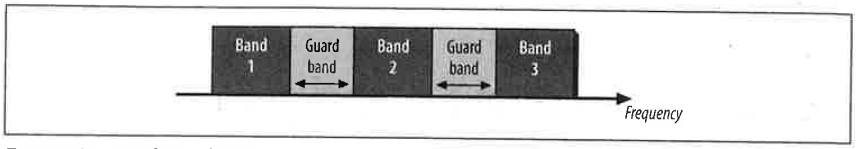


Figure 11-1. Traditional FDM

The trouble with traditional FDM is that the guard bands waste bandwidth and thus reduce capacity. To wasting transmission capacity with unused guard bands, OFDM selects channels that overlap but do not interfere with each other. Figure 11-2 illustrates the contrast between traditional FDM and OFDM.

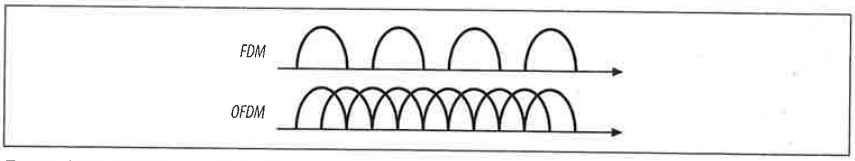


Figure 11-2. FDM versus OFDM

Overlapping carriers are allowed because the subcarriers are defined so that they are easily distinguished from one another. The ability to separate the subcarriers hinges on a complex mathematical relationship called *orthogonality*.

Orthogonality Explained (Without Calculus)

Orthogonal is a mathematical term derived from the Greek word *orthos*, meaning straight, right, or true. In mathematics, the word "orthogonal" is used to describe independent items. Orthogonality is best seen in the frequency domain, looking at a spectral breakdown of a signal. OFDM works because the frequencies of the subcarriers are selected so that at each subcarrier frequency, all other subcarriers do not contribute to the overall waveform. One common way of looking at orthogonality is shown in Figure 11-3. The signal has been divided into its three subcarriers. The peak of each subcarrier, shown by the heavy dot at the top, encodes data. The subcarrier

Exhibit 1019
Page 204 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

set is carefully designed to be orthogonal; note that at the peak of each of the subcarriers, the other two subcarriers have zero amplitude.

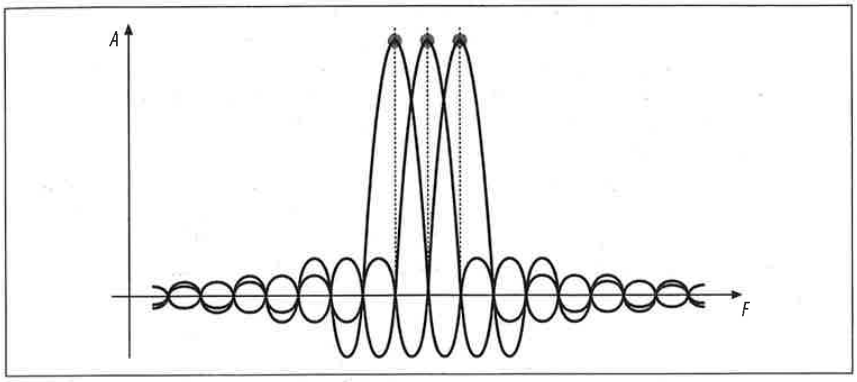


Figure 11-3. Orthogonality in the frequency domain

OFDM takes the coded signal for each subchannel and uses the inverse fast Fourier transform (IFFT) to create a composite waveform from the strength of each subchannel. OFDM receivers can then apply the FFT to a received waveform to extract the amplitude of each component subcarrier.

Guard Time

With the physical layers discussed in Chapter 10, the main problem for receivers was inter-symbol interference (ISI) (Figure 11-4). ISI occurs when the delay spread between different paths is large and causes a delayed copy of the transmitted bits to shift onto a previously arrived copy.

With OFDM, inter-symbol interference does not pose the same kind of problem. The Fourier transform used by OFDM distills the received waveform into the strengths of the subcarriers, so time shifts do not cause dramatic problems. In Figure 11-4, there would be a strong peak for the fundamental low-frequency carrier, and the late-arriving high-frequency component could be ignored.

As with all benefits, however, there is a price to pay. OFDM systems use multiple subcarriers of different frequencies. The subcarriers are packed tightly into an operating channel, and small shifts in subcarrier frequencies may cause interference between carriers, a phenomenon called *inter-carrier interference* (ICI). Frequency shifts may occur because of the Doppler effect or because there is a slight difference between the transmitter and receiver clock frequencies.

To address both ISI and ICI, OFDM transceivers reserve the beginning portion of the symbol time as the *guard time* and perform the Fourier transform only on the

Exhibit 1019
Page 205 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Fourier Analysis, the Fourier Transform, and Signal Processing

The Fourier transform is often called "the Swiss Army knife of signal processing." Signal processing often defines actions in terms of frequency components. Receivers, however, process a time-varying signal amplitude. The *Fourier transform* is a mathematical operation that divides a waveform into its component parts. Fourier analysis takes a time-varying signal and converts it to the set of frequency-domain components that make up the signal.

Signal-processing applications often need to perform the reverse operation as well. Given a set of frequency components, these applications use them like a recipe to build the composite waveform with these frequency components. The mathematical operation used to build the composite waveform from the known ingredients in the frequency domain is the *inverse Fourier transform*.

Strictly speaking, Fourier analysis is applied to smooth curves of the sort found in physics textbooks. To work with a set of discrete data points, a relative of Fourier transform called the *discrete Fourier transform* (DFT) must be used. Like the Fourier transform, the DFT has its inverted partner, the inverse DFT (IDFT).

The DFT is a computationally intensive process of order N^2, which means that its running time is proportional to the square of the size of the number of data points. If the number of data points is an even power of two, however, several computational shortcuts can be taken to cut the complexity to order N log N. On large data sets, the reduced complexity boosts the speed of the algorithm. As a result, the "short" DFT applied to 2^n data points is called the fast Fourier transform (FFT). It also has an inverted relative, the inverse fast Fourier transform (IFFT).

Fast Fourier transforms used to be the domain of supercomputers or special-purpose, signal-processing hardware. But with the microprocessor speeds available today, sophisticated signal processing is well within the capabilities of a PC. Specialized digital signal processors (DSPs) are now cheap enough to be used in almost anything—including the chip sets on commodity 802.11 cards.

non–guard time portion of the symbol time. The non–guard time portion of the symbol is often called the *FFT integration time* because the Fourier transform is performed only on that portion of the symbol.

Delays shorter than the guard time do not cause ICI because they do not allow frequency components to leak into successive symbol times. Selecting the guard time is a major task for designers of OFDM systems. The guard time obviously reduces the overall throughput of the system because it reduces the time during which data transmission is allowed. A guard time that is too short does not prevent interference but does reduce throughput, and a guard time that is too long reduces throughput unnecessarily.

Exhibit 1019
Page 206 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

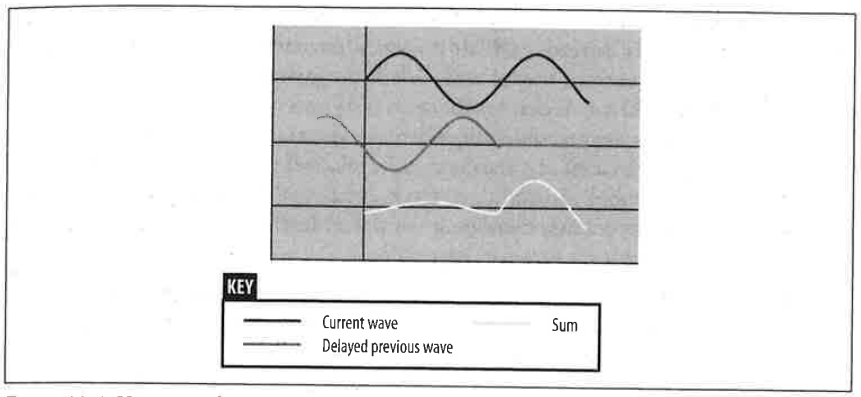


Figure 11-4. ISI reviewed

Cyclic Extensions (Cyclic Prefixes)

The most straightforward method of implementing the guard time would be simply to transmit nothing during the guard time, as shown in Figure 11-5.

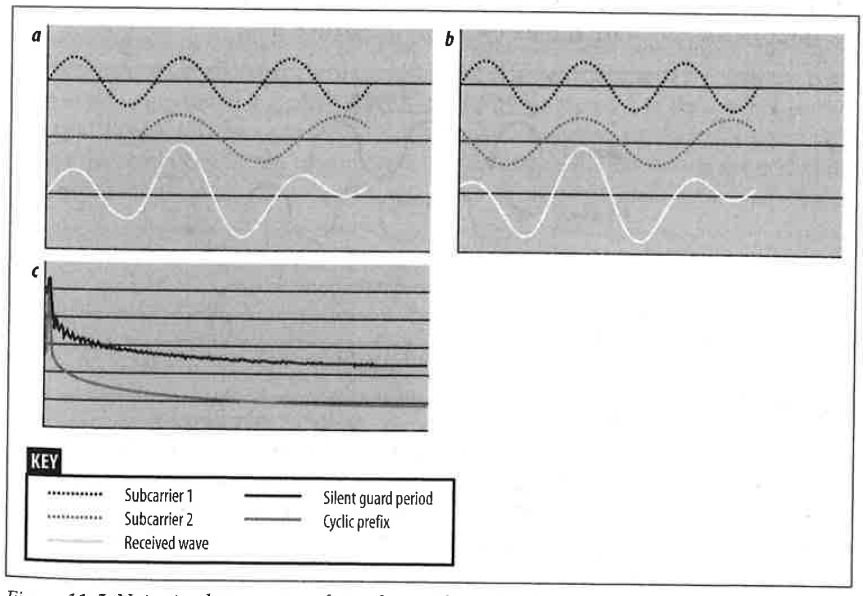


Figure 11-5. Naive implementation of guard time (do not do this!)

Exhibit 1019
Page 207 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Simplistic implementations of the guard time can destroy orthogonality in the presence of common delay spreads. OFDM depends on having an integer number of wavelengths between each of the carriers. When the guard time is perfectly quiet, it is easy to see how a delay can destroy this necessary precondition, as in Figure 11-5. When the two subcarriers are added together, the spectral analysis shows subcarrier 1 (two cycles/symbol) as a strong presence and a relatively smaller amount of subcarrier 2 (three cycles/symbol). In addition, the spectral analysis shows a large number of high-frequency components, muddying the waters further. These components are the consequence of suddenly turning a signal "on."

Solving the problems associated with a quiet guard time is quite simple. Each subcarrier is extended through the FFT integration period back through the preceding guard time. Extending each subcarrier (and hence the entire OFDM symbol) yields a Fourier transform that shows only the amplitudes of the subcarrier frequencies. This technique is commonly called *cyclic extension*, and it may be referred to as the "cyclic prefix extension." The guard time with the extended prefix is called the *cyclic prefix*.

In Figure 11-6, the cyclic prefix preserves the spectral analysis. Subcarrier 1 was not shifted and is not a problem. Subcarrier 2 is delayed, but the previous symbol appears only in the guard time and is not processed by the Fourier transform. Thanks to the cyclic prefix extension, when subcarrier 2 is processed by the Fourier transform, it is a pure wave at three cycles per integration time.

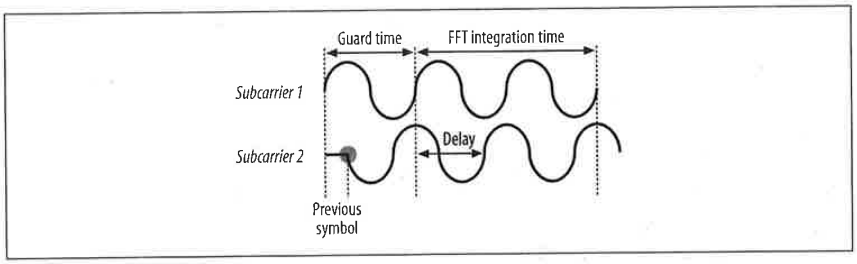


Figure 11-6. Cyclic prefix extension

Convolution Coding

Strictly speaking, convolution coding is not part of OFDM. However, OFDM is used in applications for which the signal is subject to narrowband interference or frequency-specific narrowband fading, also known as *deep fading*. When fading occurs, a channel's ability to carry data may go to zero because the received amplitude is so small. To keep a few faded channels from driving the bit error rate to the sky, OFDM implementations often apply an error correction code across all the subchannels. Implementations that use an error correction code in conjunction with OFDM are sometimes called *coded OFDM* (COFDM).

Exhibit 1019
Page 208 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

One common technique applied by OFDM transceivers is *convolution coding*, which is a specialized form of a forward error-correcting code. Full details of convolution coding are beyond the scope of this book. One important point is that the convolution code is described by a rate (R) that specifies the number of data bits transmitted per code bit. A convolution code with R=1/2 transmits one data bit for every two code bits. As the code rate decreases, more code bits are available to correct errors, and the code becomes more robust. However, the price of robustness is decreased throughput. 802.11a uses convolution codes extensively. The most conservative data rates use codes with a rate of 1/2, and the aggressive codes used for the highest data rates use a coding rate of 3/4.

Windowing

One further enhancement helps OFDM transceivers cope with real-world effects. Transitions can be abrupt at symbol boundaries, causing a large number of high-frequency components (noise). To make OFDM transmitters good radio citizens, it is common to add padding bits at the beginning and end of transmissions to allow transmitters to "ramp up" and "ramp down" from full power. Padding bits are frequently needed when error correction coding is used. Some documentation may refer to the padding as "training sequences."

Windowing is a technique used to bring the signal for a new symbol gradually up to full strength while allowing the old symbol to fade away. Figure 11-7 shows a common windowing function based on a cosine curve. At the start of the symbol period, the new function is brought up to full strength according to the cosine function. When the symbol ends, the cosine curve is used to fade out the bits at the end of the symbol.

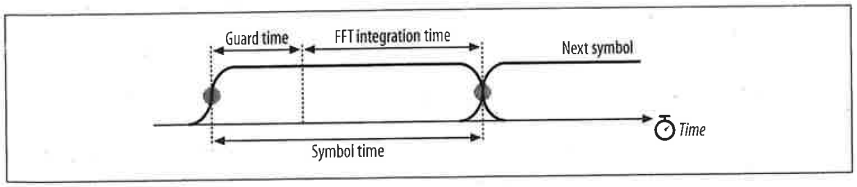


Figure 11-7. Cosine windowing technique

OFDM as Applied by 802.11a

802.11a is not a radical application of OFDM. The task group responsible for standardizing OFDM took the middle ground to apply OFDM to wireless LAN.

Exhibit 1019
Page 209 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

OFDM Parameter Choice for 802.11a

When choosing OFDM parameters, there are usually three given items of information. Bandwidth is fixed, often by regulatory authorities. Delay is determined by the environment in which the OFDM system will operate; most office buildings generally show a delay spread of 40–70 ns, though in some environments, the delay spread can approach 200 ns. Finally, the bit rate is usually a design goal, though the goal is usually "make the bit rate as high as possible, given the constraints of the other parameters."

One common guideline is that the guard time should be two to four times the average delay spread. As a result, the 802.11a designers selected a guard time of 800 ns. Symbol duration should be much larger than the guard time, but within reason. Larger symbol times mean that more subcarriers can fit within the symbol time. More subcarriers increase the signal-processing load at both the sender and receiver, increasing the cost and complexity of the resulting device. A practical choice is to select a symbol time at least five times the guard time; 802.11a matches the 800-ns guard time with a 4-µs symbol time. Subcarrier spacing is inversely related to the FFT integration time. 802.11a has a 3.2-µs integration time and a subcarrier spacing of 0.3125 MHz (1/3.2 µs).

Operating channels in 802.11a are specified as 20 MHz wide. The bandwidth of an operating channel is a design decision. Wider operating channels have higher throughput, but fewer operating channels fit into the assigned frequency spectrum. The use of a 20-MHz operating channel allows for reasonable speeds on each channel (up to 54 Mbps), as well as a reasonable number of operating channels in the assigned spectrum. 802.11a offers a wide variety of choices in modulation and coding to allow for a trade-off between robust modulation and conservative coding, which yields low, reliable throughput and finer-grained modulation and aggressive coding, resulting in higher, yet somewhat more fragile, throughput.

Structure of an Operating Channel

Like the DS PHYs, the OFDM physical layer organizes the spectrum into operating channels. Each 20-MHz channel is composed of 52 subcarriers. Four of the subcarriers are used as *pilot carriers* for monitoring path shifts and ICI, while the other 48 subcarriers are used to transmit data. Subcarriers are spaced 0.3125 MHz apart. As shown in Figure 11-8, channels are numbered from –26 to 26. Subcarrier 0 is not used for signal-processing reasons.

Pilot subcarriers are assigned to subcarriers –21, –7, 7, and 21. To avoid strong spectral lines in the Fourier transform, the pilot subcarriers transmit a fixed bit sequence specified in 802.11a using a conservative modulation technique.

Exhibit 1019
Page 210 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

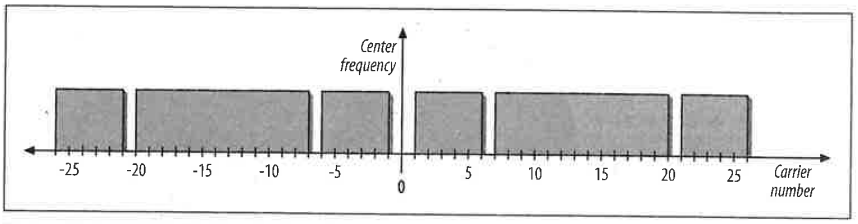


Figure 11-8. Structure of an OFDM channel

Operating Channels

In the U.S., the channels in the 5-GHz band are numbered starting every 5 MHz according to the following formula:

$$\text{center frequency (MHz)} = 5{,}000 + 5 \times n,\ n=0,1,2,\ldots 200$$

Obviously, each 20-MHz 802.11a channel occupies four channels in the U-NII bands. The recommended channel use is given in Table 11-1.

Table 11-1. United States channels for 802.11a

Band	Allowed power[a]	Channel numbers	Center frequency (GHz)
U-NII lower band	40 mW	36	5.180
(5.15–5.25 GHz)	(2.5 mW/MHz)	40	5.200
		44	5.220
		48	5.240
U-NII mid-band	200 mW	52	5.260
(5.25–5.35 GHz)	(12.5 mW/MHz)	56	5.280
		60	5.300
		64	5.320
U-NII upper band	800 mW	149	5.745
(5.725–5.825 GHz)	(50 mW/MHz)	153	5.765
		157	5.785
		161	5.805

[a] The allowed power is the maximum output power using a 6-dBi antenna gain.

There is one other feature of note about the operating bands. As with the DS PHYs, a transmit mask limits power leakage into the side bands. The mask is shown in Figure 11-9.

Figure 11-10 gives an overall view of the 802.11a channels available in the U.S. Four channels are available in each of the U-NII bands in the U.S. In the two lower U-NII

Exhibit 1019
Page 211 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

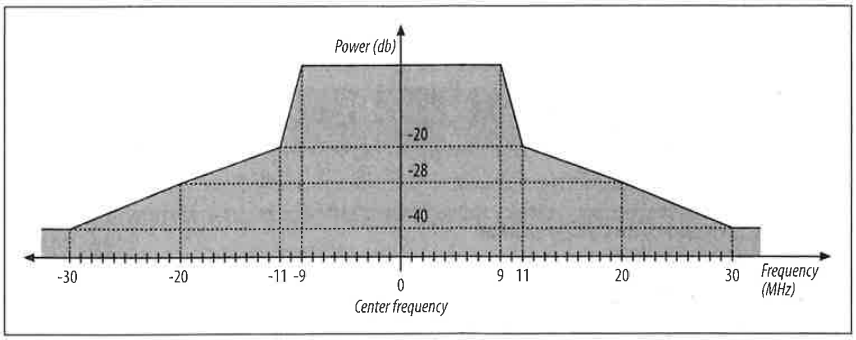


Figure 11-9. Transmit spectrum mask for 802.11a

bands, channels are allowed to overlap, and a 30-MHz guard band is present at both the lower end of the low U-NII band and the upper end of the mid U-NII band.

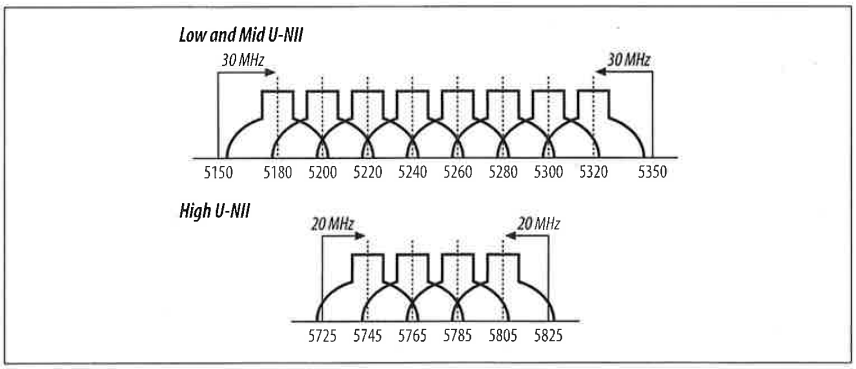


Figure 11-10. Operating bands from Table 11-1

OFDM PLCP

Like all the other physical layers, the OFDM PHY includes its own PLCP, which adds physical layer–specific framing parameters.

Framing

The OFDM PHY adds a preamble and a PLCP header. It also adds trailing bits to assist the encoding schemes used. This section divides the PLCP frame logically, but some components span different fields in the protocol unit. Figure 11-11 is the jumping-off point for discussion of the OFDM frame.

Exhibit 1019
Page 212 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

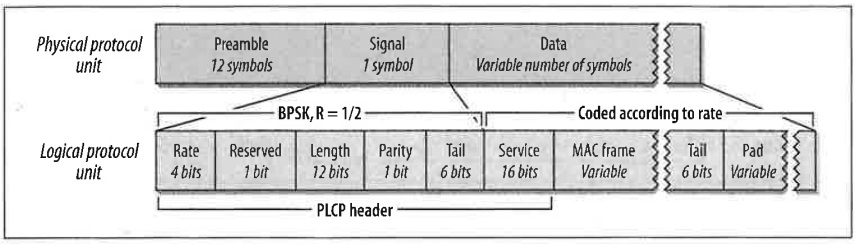


Figure 11-11. OFDM PLCP framing format

Figure 11-12 shows the start of a frame, but includes the guard intervals and windowing used by the transmitter. The preamble lasts 16 µs, which is evenly divided between short and long training sequences; the difference between the two is described in the next section. After the preamble, one OFDM symbol carries the Signal field, then a variable number of data symbols carry the end of the PLCP header, the MAC payload, and the trailer. All symbols use a modified cosine window to ensure smooth transitions. After the short preamble, which is used to synchronize frequencies, a guard time protects against multipath fading.

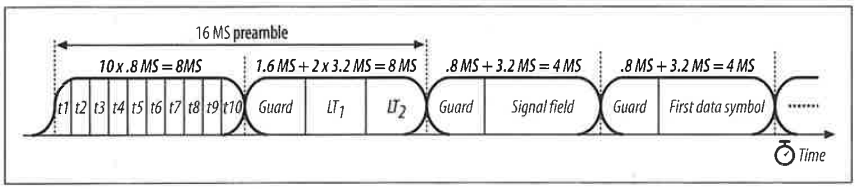


Figure 11-12. Preamble and frame start

Preamble

As with all other common IEEE 802 networks, and certainly all 802.11 physical layers, the OFDM physical protocol unit begins with a preamble. It is composed of 12 OFDM symbols that synchronize various timers between the transmitter and the receiver. The first 10 symbols are a *short training sequence*, which the receiver uses to lock on to the signal, select an appropriate antenna if the receiver is using multiple antennas, and synchronize the large-scale timing relationships required to begin decoding the following symbols. The short training sequences are transmitted without a guard period. Two *long training sequences* follow the short training sequences. Long training sequences fine-tune the timing acquisition and are protected by a guard interval.

Header

The PLCP header is transmitted in the Signal field of the physical protocol unit; it incorporates the Service field from the Data field of the physical protocol unit. As

Exhibit 1019
Page 213 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

shown in Figure 11-13, the Signal field incorporates the Rate field, a Length field, and a Tail field.

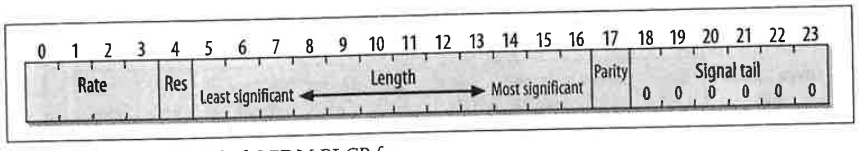


Figure 11-13. Signal field of OFDM PLCP frame

Rate (4 bits)
: Four bits encode the data rate. Table 11-2 shows the bits used to encode each of the data rates. See the section "OFDM PMD" for details on the encoding and modulation scheme used for each data rate.

Table 11-2. Rate bits

Data rate (Mbps)	Bits (transmission order)
6	1101
9	1111
12	0101
18	0111
24	1001
36	1011
48	0001
54	0011

Length (12 bits)
: Twelve bits encode the number of bytes in the embedded MAC frame. Like most fields, it is transmitted least-significant bit to most-significant bit. The length is processed by a convolution code to protect against bit errors.

Parity (1 bit) and Reserved (1 bit)
: Bit 4 is reserved for future use and must be set to 0. Bit 17 is an even parity bit for the first 16 Signal bits to protect against data corruption.

Tail (6 bits)
: The Signal field ends with six 0 tail bits used to unwind the convolution code. As such, they must by definition be processed by the convolution code.

Service (16 bits)
: The final field in the PLCP header is the 16-bit Service field. Unlike the other components of the PLCP header, it is transmitted in the Data field of the physical protocol unit at the data rate of the embedded MAC frame. The first eight bits are set to 0. As with the other physical layers, MAC frames are scrambled before transmission; the first six bits are set to 0 to initialize the scrambler. The

Exhibit 1019
Page 214 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

remaining nine bits are reserved and must set to 0 until they are adopted for future use.

Data

The encoding scheme used for the data depends on the data rate. Before transmission, data is scrambled, as it is with the other physical layers. The Service field of the header is included in the Data field of the physical protocol unit because it initializes the scrambler.

Trailer

The Data field of the physical protocol unit ends with a trailer. (The 802.11a specification does not call the ending fields a trailer, but it is a descriptive term.) It is composed of two fields:

Tail (6 bits)
: Like the tail bits in the PLCP header, the tail bits appended to the end of the MAC frame bring the convolution code smoothly to an end.

Pad (variable)
: As used by 802.11a, OFDM requires that fixed-size blocks of data bits be transferred. The Data field is padded so that its length is an integer multiple of the block size. The block size depends on the modulation and coding used by the data rate; it is discussed in the next section.

OFDM PMD

The OFDM PHY uses a cocktail of different modulation schemes to achieve data rates ranging from 6 Mbps to 54 Mbps. In all cases, the physical layer uses a symbol rate of 250,000 symbols per second across 48 subchannels; the number of data bits per symbol varies. An OFDM symbol spans all 48 subchannels.

There are four rate tiers with the OFDM PHY: 6 and 9 Mbps, 12 and 18 Mbps, 24 and 36 Mbps, and 48 and 54 Mbps. Support is required for 6, 12, and 24 Mbps, which are lowest speeds in each of the first three tiers, and therefore the most robust in the presence of interference. The lowest tier uses binary phase shift keying (BPSK) to encode 1 bit per subchannel, or 48 bits per symbol. The convolution coding means that either half or one quarter of the bits are redundant bits used for error correction, so there are only 24 or 36 data bits per symbol. The next tier uses quadrature phase shift keying (QPSK) to encode 2 bits per subchannel, for a total of 96 bits per symbol. After subtracting overhead from the convolution code, the receiver is left with 48 or 72 data bits. The third and fourth tiers use generalized forms of BPSK and QPSK known as quadrature amplitude modulation (QAM). 16-QAM encodes 4 bits using 16 symbols, and 64-QAM encodes 6 bits using 64 symbols. The third tier uses 16-QAM along with the standard R=1/2 and R=3/4 convolution codes. To achieve

Exhibit 1019
Page 215 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

higher rates with 64-QAM, however, the convolution codes use R=2/3 and R=3/4. Table 11-3 summarizes the coding methods used by each data rate in the OFDM PHY.

Table 11-3. Encoding details for different OFDM data rates

Speed (Mbps)	Modulation and coding rate (R)	Coded bits per carrier[a]	Coded bits per symbol	Data bits per symbol[b]
6	BPSK, R=1/2	1	48	24
9	BPSK, R=3/4	1	48	36
12	QPSK, R=1/2	2	96	48
18	QPSK, R=3/4	2	96	72
24	16-QAM, R=1/2	4	192	96
36	16-QAM, R=3/4	4	192	144
48	64-QAM, R=2/3	6	288	192
54	64-QAM, R=3/4	6	288	216

[a]Coded bits per subchannel is a function of the modulation (BPSK, QPSK, 16-QAM, or 64-QAM).
[b]The data bits per symbol is a function of the rate of the convolution code.

Clear Channel Assessment

The OFDM PHY specification leaves implementers a great deal of latitude in selecting techniques for noting a busy channel. Received signal strength thresholds determine whether the channel is in use, but the main guideline for 802.11a equipment is that it must meet certain performance standards. Implementations are free to use the Packet Length field from the PLCP header to augment clear channel assessment, but this is not required.

An Example of OFDM Encoding

OFDM encoding, as you can no doubt see by now, is an intense, multistep process. One of the additions that 802.11a made to the original specification was Annex G, an encoding of Schiller's *Ode to Joy* for transmission over an 802.11a network.* Shortly after 802.11a was published, the IEEE 802.11 working group discovered several errors in the example and published a correction. If you are interested in learning about OFDM encoding in detail, you can refer to this example.

Characteristics of the OFDM PHY

Parameters specific to the OFDM PHY are listed in Table 11-4. Like the physical layers presented in Chapter 10, the OFDM PHY also incorporates a number of parameters to

* Well, an English translation, anyway...

Exhibit 1019
Page 216 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

adjust for the delay in various processing stages in the electronics. As a final note, the extra radio bandwidth provided by the U-NII bands offers a great deal of throughput. There are eight overlapping channels available for the OFDM PHY, so it can offer up to 432 Mbps in an area where all eight channels are co-located.

Table 11-4. OFDM PHY parameters

Parameter	Value	Notes
Maximum MAC frame length	4,095 bytes	
Slot time	9 µs	
SIFS time	16 µs	The SIFS is used to derive the value of the other interframe spaces (DIFS, PIFS, and EIFS).
Contention window size	15 to 1,023 slots	
Preamble duration	20 µs	
PLCP header duration	4 µs	

Like the other physical layers, the OFDM PHY has a number of attributes that can be adjusted by a vendor to balance delays in various parts of the system. It includes variables for the latency through the MAC, the PLCP, and the transceiver, as well as variables to account for variations in the transceiver electronics.

Exhibit 1019
Page 217 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

CHAPTER 12
Using 802.11 on Windows

Whether you've made it to this point by skipping Chapters 3–11, or whether you've read all the theory, we're now going to get our hands dirty and start installing equipment.

From the standpoint of practical system and network administration, working with 802.11 is similar to working with Ethernet. Installing 802.11 drivers is nearly identical to installing Ethernet drivers, and the network interfaces behave almost exactly like Ethernet interfaces. 802.11 interfaces cause an ARP cache to be brought into existence, and other software may even perceive the wireless interface as an Ethernet interface. Unlike many Ethernet drivers, however, 802.11 drivers can have a number of advanced knobs and features that reflect the additional management features presented in Chapter 7.

This chapter is not intended to be a definitive guide to Windows drivers for 802.11 network cards. There are two major development lines in Windows (9x versus NT and progeny), and adding additional software such as a VPN client can further complicate matters. (My advice is to install the wireless LAN card before any VPN client software.) There are a number of vendors, and, as you'd expect, the driver software varies from one vendor to the next. The examples show how to install a driver on Windows and explain the non-Ethernet driver features in some detail. I selected two 802.11 cards as examples: the Nokia C110/C111 and the Lucent ORiNOCO. While not particularly common, the Nokia card is interesting because it has a number of advanced features and exposes a number of the network parameters that were discussed in Chapters 2–10. The Lucent card (which is sold under a number of different labels) probably has the lion's share of the market, and it hides most of the exotic configuration parameters from the user. Most cards that are available fall somewhere between these two extremes.

Exhibit 1019
Page 218 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Nokia C110/C111

Nokia's 802.11b solution comes in two similar form factors. The C110 is a Type 2 PC Card with an integrated antenna; the C111 is basically the same, but with two external antenna connectors. The card ships with a CD to enable basic installation, and updated drivers are available from *http://forum.nokia.com/* after registering.

Installation

Driver installation begins before inserting the card, so start by inserting the CD-ROM into the CD-ROM drive. After a splash screen, an installation program begins. Its main screen is shown in Figure 12-1. Select Installing Nokia C110/C111 to launch the installer.

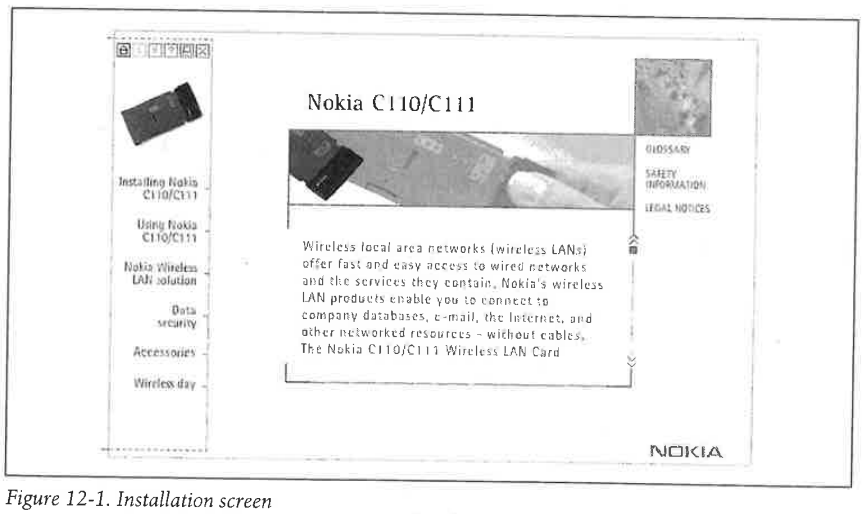


Figure 12-1. Installation screen

The next steps are very familiar. Selecting the install option launches InstallShield and brings up an admonition to close all other programs. Like all commercial software, the Nokia driver is licensed. Accept the license agreement to proceed. Next, the driver asks for the country in which the card is being used. This information is used to set the regulatory domain, which affects the radio channels that can be used. See Figure 12-2.

The setup program then asks where it should put the files that it installs. The default location is *C:\Program Files\Nokia C110*. Next, the driver asks which components should be installed. The software package is composed of three major components: drivers, help files, and administrator components. Typical users install only the first

Exhibit 1019
Page 219 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

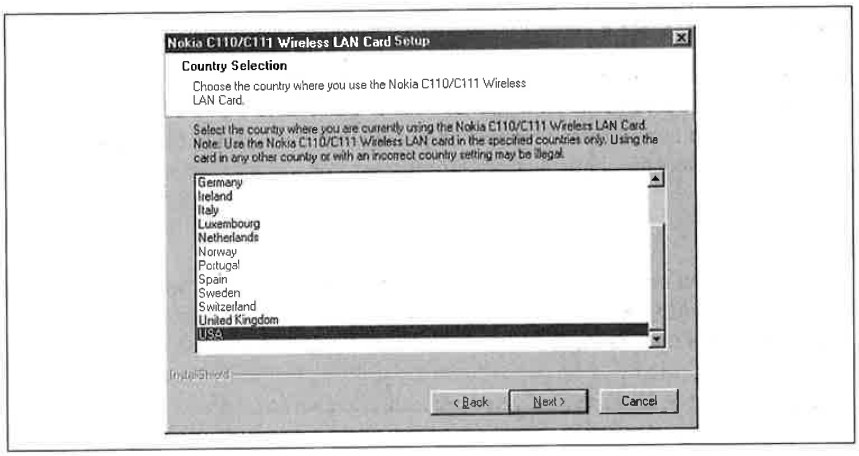


Figure 12-2. Country selection

two, but network administrators can use features in the software to streamline installation procedures for large numbers of users. A Typical installation consists of only the first two, and an Administrator installation uses all three. For good measure, a Custom installation allows any subset of the three components to be installed. In a Custom installation, the administrator component is not installed by default and must be selected explicitly.

At this point, the installation program has collected all the information necessary to install the driver. It copies files that were unpacked during the installation and makes registry changes to activate the new driver. After that completes, a dialog box appears and prompts you to insert the card into the computer to complete the installation. When the card is inserted into the PC Card slot, the message shown in Figure 12-3 appears, and the installation is complete.

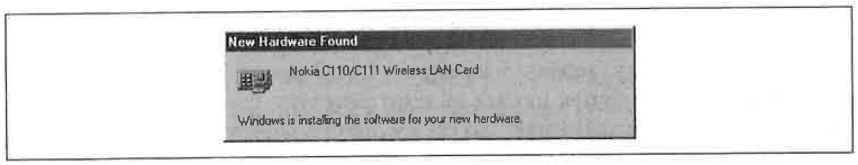


Figure 12-3. Driver installation prompt

The installer prompts you for the creation of a network profile before restarting the computer. Profiles are one of the card's advanced features; they are optional, but they make card management much simpler. It will be interesting to see whether other vendors pick up on this idea.

Exhibit 1019
Page 220 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Network Profiles

The Nokia card groups settings into *profiles*, which allow users to switch easily between networks. Immediately after installing the driver, the user is prompted to create a profile if none exists. Administrators may create customized driver installation disks or smart cards to distribute settings more easily. In addition to the run-of-the-mill network settings, the Nokia driver can control whether the system attempts to log in to a domain and a Microsoft workgroup. Profiles can also contain WEP keys as well as a number of 802.11 parameters. Figure 12-4 shows the initial Profile Wizard screen. Profiles are assigned text names; the name need not have anything to do with the SSIDs in use.

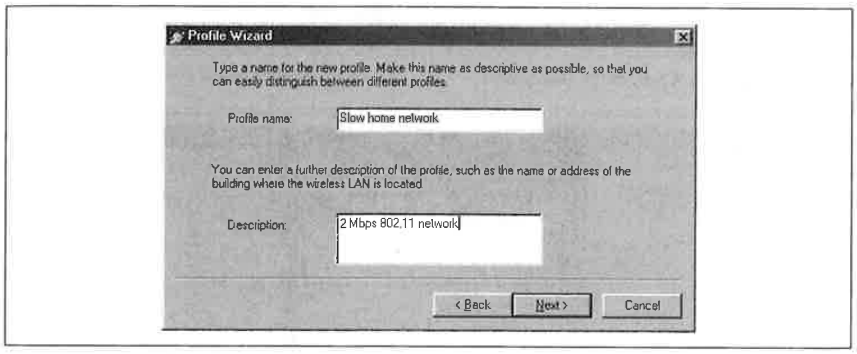


Figure 12-4. Initial Profile Wizard screen

After naming the profile and entering a detailed description, the user must then select the type of network in use. Infrastructure networks use access points, and ad hoc networks are independent BSSs.* After selecting the network operating mode, the user proceeds to a network selection window (Figure 12-5). Networks are distinguished by their service set IDs, which are called names for simplicity.

This window allows you to enter channel information. Unless you have an overwhelming reason to set the channel explicitly, leave the channel set to automatic, which means that the driver scans all channels when it is initialized. There are two ways to select the network name. One is to type the SSID for which the driver should search. To make it easier for basic users, the small unobtrusive button to the right of the network name field pops up a list of networks currently in range (Figure 12-6). This window is a nice touch; it shows you data rates and signal strengths to help you make an intelligent choice.

* Earlier Nokia products also had an operating mode called Instawave, which allowed direct station-to-station communication simultaneously with station-to-access point communication. Instawave was nonstandard and never found extensive use in production networks.

Exhibit 1019
Page 221 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

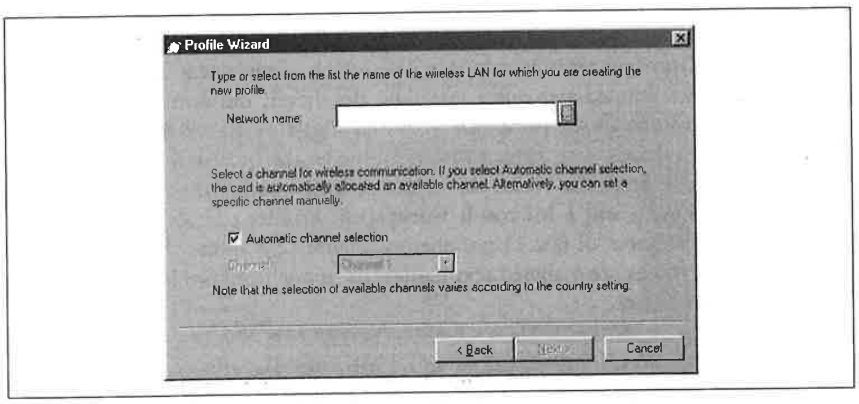


Figure 12-5. Network parameter dialog box

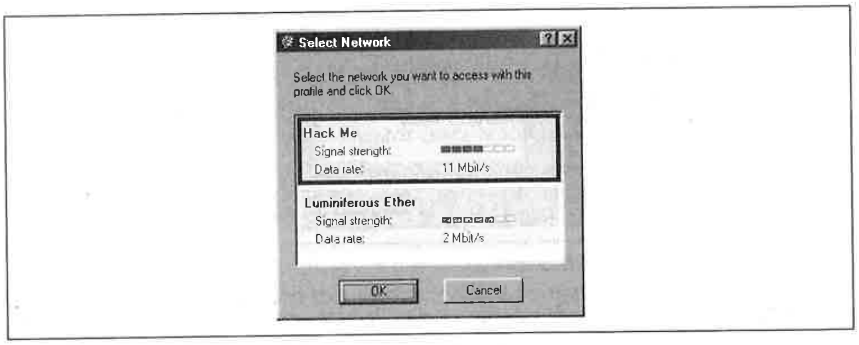


Figure 12-6. Network selection box

After selecting a network, the user is presented with the final screen for selecting basic network parameters (Figure 12-7). Most 802.11 networks use DHCP to assign IP addresses. If not, the profile can be modified later to specify an address explicitly. You can also specify Windows domain and workgroup names.

At this point, a network profile has been created. It appears on the main screen, as shown in Figure 12-8.

To select a profile, highlight it and click Apply. The selected profile gets a big green check mark to show that it has been selected. As part of choosing a profile, the driver maintains network configuration settings. These settings can be updated as the user changes profiles. When a new profile is selected, the dialog box in Figure 12-9 appears. My experience has been that you usually need to restart the system when you apply a new profile.

When the system comes back up, there will be a small default monitoring window, in addition to a taskbar icon. The small window, shown in Figure 12-10, displays the

Exhibit 1019
Page 222 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

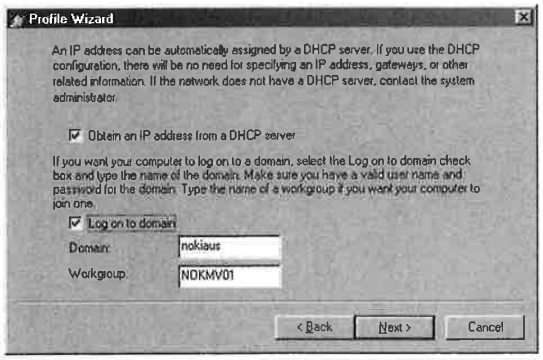


Figure 12-7. Addressing and login options

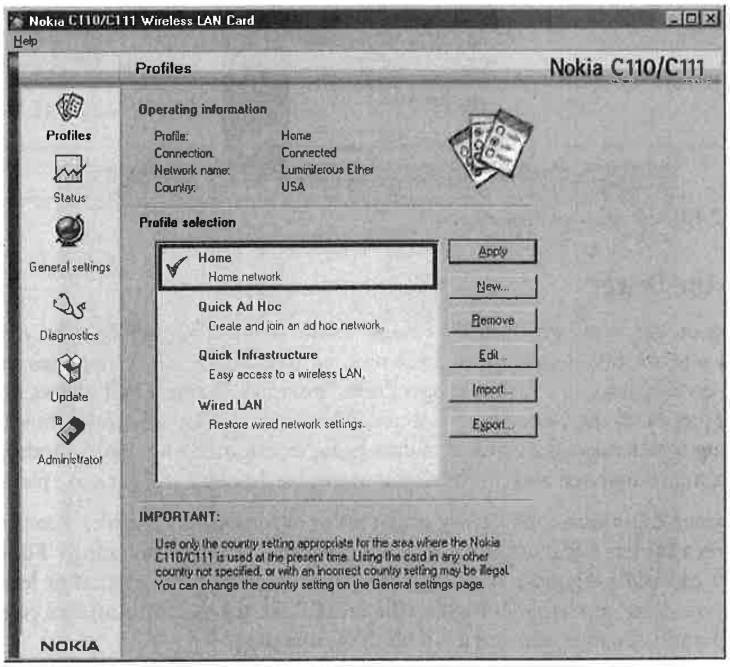


Figure 12-8. Profile selection screen

Exhibit 1019
Page 223 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

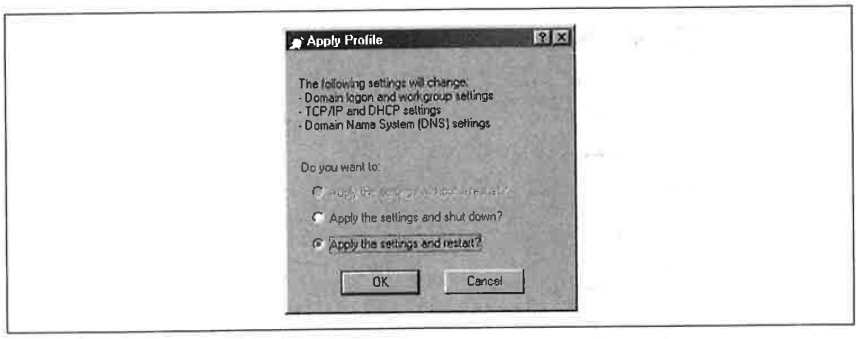


Figure 12-9. Options when changing profiles

network profile and a signal strength meter. It also provides a button next to the profile name for gaining access to the detailed configuration window.

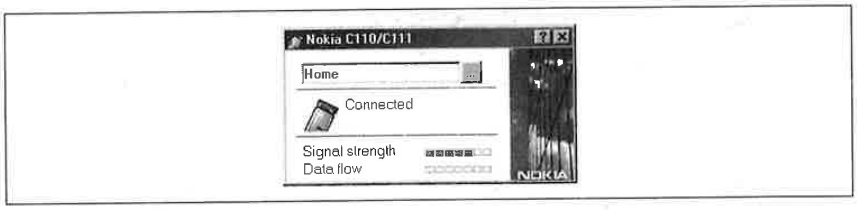


Figure 12-10. Default monitoring window

Using the Driver

Clicking on the small gray button brings up the main driver screen. The driver is divided into six broad categories, each with an icon at the left. Categories may be further divided into tabs for more specific information. Figure 12-11 shows the Status category, with the General tab selected. The Status → General screen shows a signal strength meter and the amount of data being transmitted on the BSS. Both graphs are continually updated and can be useful in troubleshooting and network planning.

When troubleshooting connectivity problems or expanding a network, it can be useful to run a card as a scanner, simply to see the access points within range. For example, you can walk around a building with your laptop and ensure that at least two access points are reachable in high-traffic areas. Like the general statistics page, the access point listing is reached as a tab off the status page.

Figure 12-12 shows two access points. The 11-Mbps access point is a Nokia A032, which is capable of transmitting its IP address and workload. The mechanism used to accomplish this will be discussed in more detail in Chapter 16.

Exhibit 1019
Page 224 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

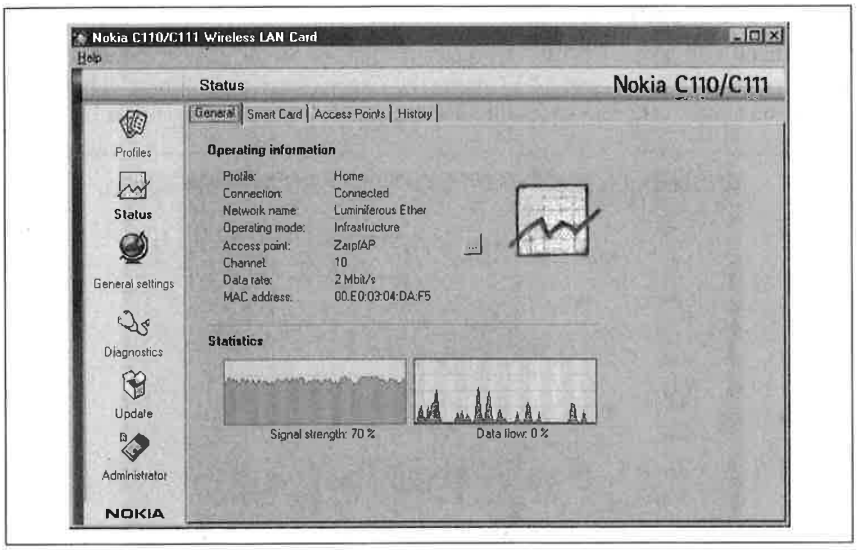


Figure 12-11. Status → General driver screen

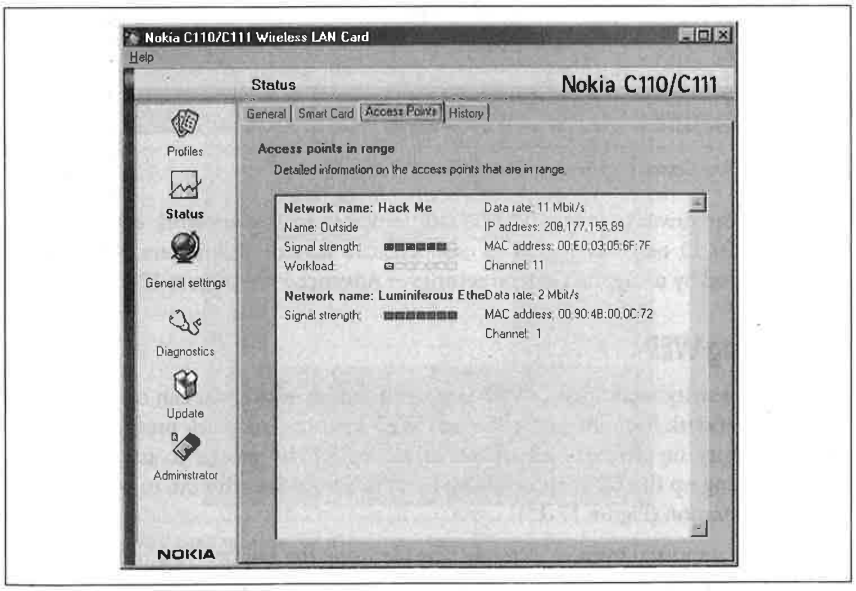


Figure 12-12. General → Access point tab

Exhibit 1019
Page 225 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Global driver configuration options

The General settings button can be used to gain access to the general driver settings used on a global basis. Figure 12-13 shows the General settings → General tab, which allows you to select the regulatory domain and enable low-power operation.

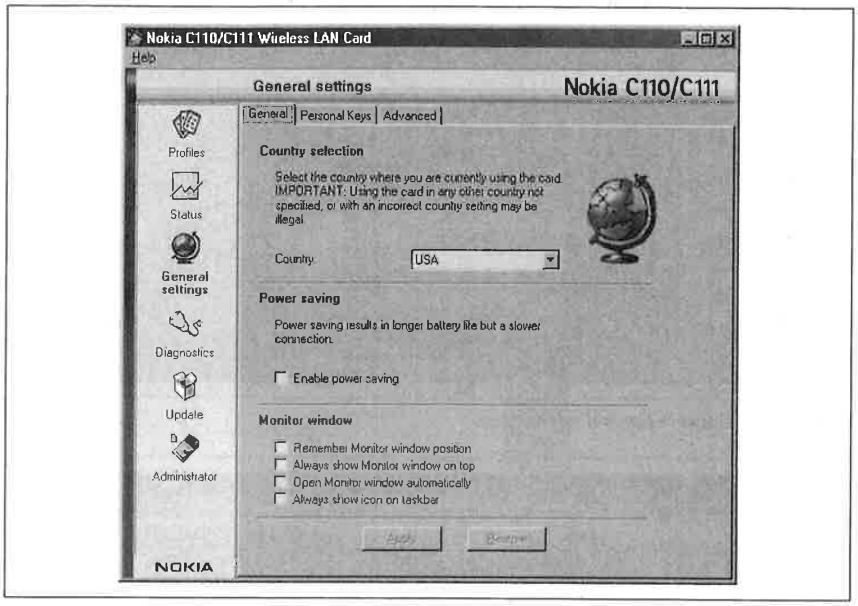


Figure 12-13. The General settings → General tab

By default, the driver manages TCP/IP properties and other network configuration and treats 802.11 network names as case-sensitive network identifiers. All of these can be changed by using the General settings → Advanced tab (Figure 12-14).

Configuring WEP

Despite its security weaknesses, WEP is a significant measure you can take to secure an 802.11 network from intruders. To add WEP keys to a network profile, go to the profile category on the left side of the driver, select the profile to use, and click **Edit...** to bring up the Edit Profile dialog box. Select the Security tab to bring up the WEP configuration (Figure 12-15).

Keys may be imported from an external text file using the Import... button or may be added from scratch using the Add... button. Figure 12-16 shows the dialog box used to edit a shared WEP key.

Exhibit 1019
Page 226 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

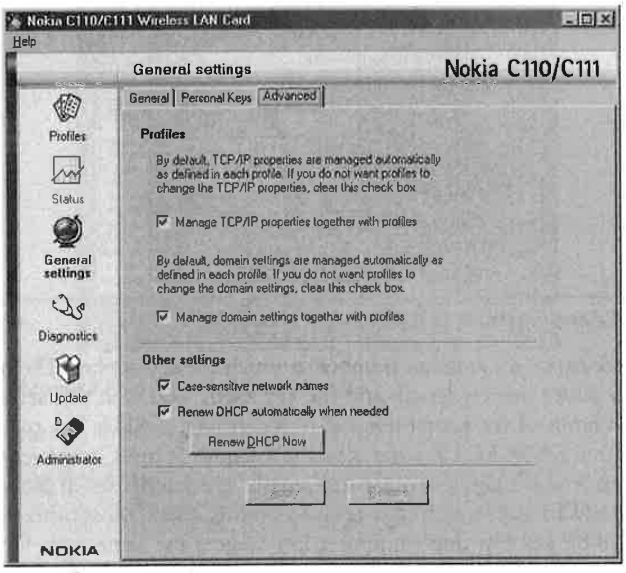


Figure 12-14. General settings → Advanced tab

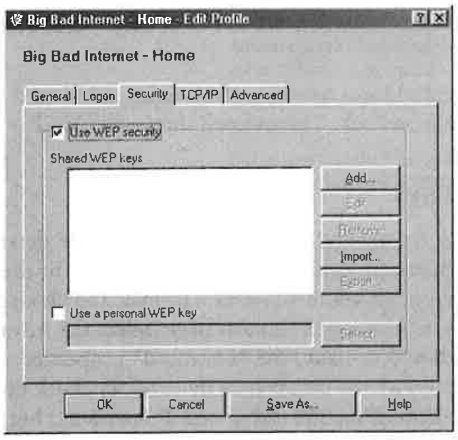


Figure 12-15. WEP configuration

802.11 permits four shared keys per SSID. Not all drivers support using all four, but Nokia's driver does. At the top of the dialog box there are fields for selecting the

Exhibit 1019
Page 227 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

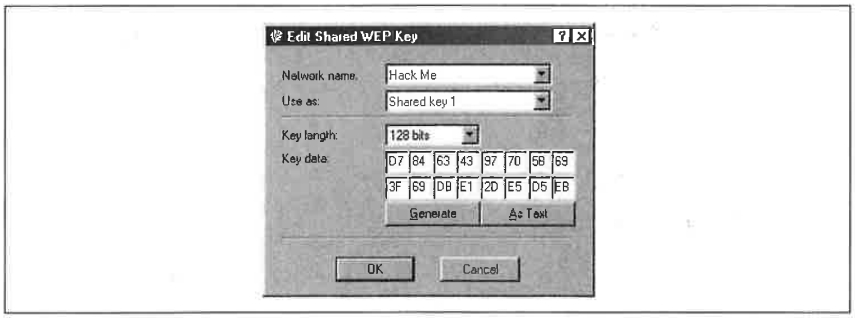


Figure 12-16. Editing a shared WEP key

SSID (network name) and the key number to which the key applies. The bottom part of the screen shows the key length and the key itself. Note that this screen specifies key length in terms of the actual number of secret bits; a 40-bit key corresponds to the standardized 64-bit RC4 key described in Chapter 5; most other vendors would refer to this as a 64-bit key. No other key lengths are standardized; Nokia also supports a 128-bit WEP key, which requires a 152-bit RC4 key. Most other vendors also support a 128-bit key but don't interpret key lengths the same way; for most vendors, 128 bits means a 128-bit RC4 key with 104 secret bits. The result is that Nokia cards won't interoperate with other vendors' products at the 128-bit key length.

For the first station on a network, the Generate button may be used to generate a random key. The As Text button allows you to cut and paste a hexadecimal string to use as the key.

When WEP is configured for a network, the wireless LAN card icon in the monitor window appears with a padlock. The padlock shows that WEP is in use.

Advanced Properties

The Nokia driver allows network administrators to make detailed changes to the 802.11 parameters discussed in previous chapters. To get to the advanced settings, go to the main configuration window, select a profile, and choose **Edit...**, which displays the main profile editing page. On this page, select the Advanced tab to display the advanced properties. Normally, the "advanced" properties are configured automatically. If you want to set the parameters by hand, uncheck the Automatic configuration checkbox. Clicking on the Advanced Properties... button takes you to a window that lets you modify several of the basic 802.11 properties (Figure 12-17). Table 12-1 describes the available options and their defaults.

Exhibit 1019
Page 228 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

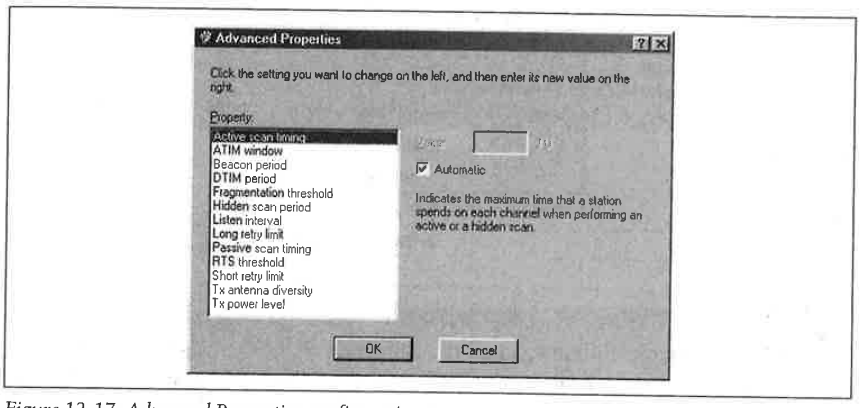


Figure 12-17. Advanced Properties configuration screen

Table 12-1. Parameter defaults used by the Nokia driver

Property	Default ("automatic") value	Measurement units	Description	Reference
Active scan timing	30	TU	Number of time units spent monitoring each channel during an active scan.	Chapter 7;802.11 clause 11.1.3
ATIM window	0	TU	Amount of time that stations in an ad hoc network must remain awake after a beacon.	Chapter 7;802.11 clause 11.2.2
Beacon period	100	TU	Amount of time between target beacon transmissions for an ad hoc BSS .	Chapter 7;802.11 clause 11.1.2
DTIM period	10	beacon periods	Number of beacon periods between DTIM messages.	Chapter 7;802.11 clause 11.2.1.3
Fragmentation threshold	2,346	bytes	Packets larger than the threshold are fragmented at the MAC layer for transmission.	Chapter 3;802.11 clause 9.4
Hidden scan period	30	seconds		
Listen interval	10	beacon periods	Number of beacon periods between station waking up to listen to DTIMs for buffered traffic delivery.	Chapter 7;802.11 clause 11.2
Long retry limit	4	attempts	Maximum number of attempts to transmit a frame bigger than the RTS threshold.	Chapter 3;802.11 clause 9.5.2.3
Passive scan timing	250	TU	Amount of time spent listening for traffic on each radio channel during a passive scan.	Chapter 7;802.11 clause 11.1
RTS threshold	0	bytes	Packets larger than the RTS are preceded by an RTS/CTS handshake.	Chapter 3;802.11 clause 9.2
Short retry limit	7	attempts	Maximum number of attempts to transmit a frame shorter than the RTS threshold.	Chapter 3;802.11 clause 9.5.2.3

Exhibit 1019
Page 229 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Table 12-1. Parameter defaults used by the Nokia driver (continued)

Property	Default ("automatic") value	Measurement units	Description	Reference
Tx antenna diversity	1	N/A	0–enabled 1–disabled	
Tx power level	1	N/A	1–high power 2–low power	

Smart Cards

When 802.11 networks initially gained prominence, one of the biggest concerns was how to distribute configuration information to the mobile computers on the network. One of the more novel solutions was Nokia's smart cards. The C11x cards have an integrated smart-card reader. Administrators can write profiles out to smart cards using the Administrator menu in the main configuration screen and distribute smart cards to users. Smart cards can store the entire profile, including WEP keys and TCP/IP configuration, which reduces the possibility of user error—users don't have to type network parameters or keys correctly. Smart cards also present a tamper-resistant barrier to sensitive information (such as WEP keys) that might otherwise be stored in a file on the hard disk.

Unlocking the smart card

Users with a smart card are presented with a challenge to unlock the card when the wireless card is placed in a PC Card slot. First, the dialog box of Figure 12-18 appears as the smart card is opened.

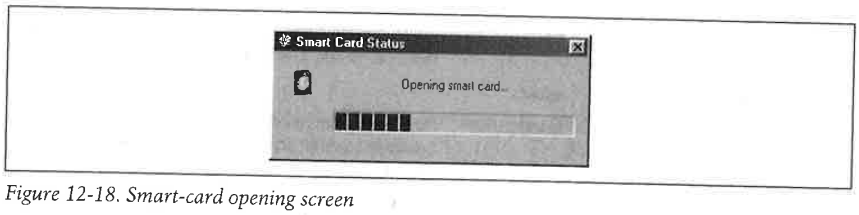


Figure 12-18. Smart-card opening screen

When the smart-card initialization completes, the user enters the PIN for the smart card, using the window in Figure 12-19.

After three unsuccessful attempts to open the smart card, it locks up and cannot be unlocked even by the correct PIN. Only the PIN Unlocking Key (PUK) distributed with the smart card, can be used to unlock the smart card after it locks. When the smart card is unlocked, any profiles that it stores can be used by the driver. Smart-card profiles cannot be edited directly and are identified with a special icon on the main configuration screen (Figure 12-20).

Exhibit 1019
Page 230 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

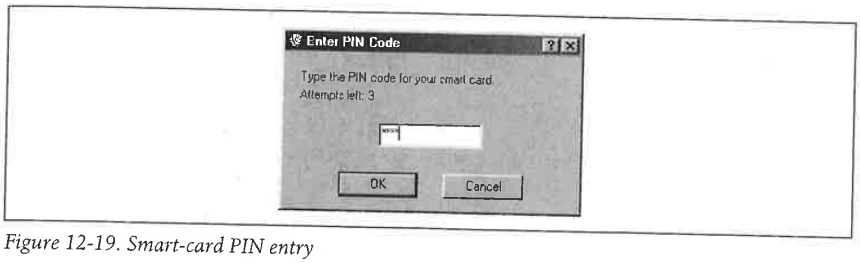


Figure 12-19. Smart-card PIN entry

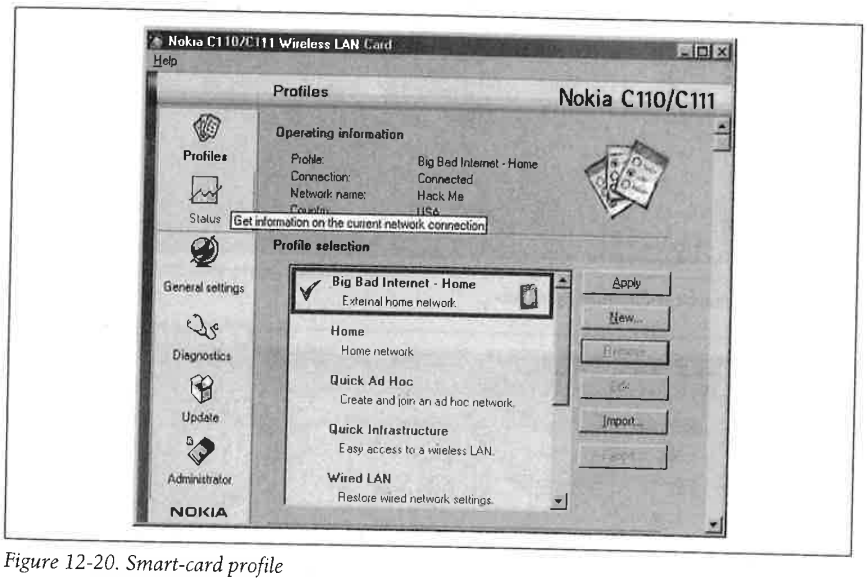


Figure 12-20. Smart-card profile

Locking the smart card and changing the PIN

If desired, the smart card can be locked from the driver configuration panel. Select Status, then choose the Smart Card tab, and click on the **Lock...** button shown in Figure 12-21. You can also use this screen to change the PIN by clicking the Change **PIN Code...** button at the bottom.

Moving profiles onto the smart card

Only driver installations that include the administrator routines can create or modify the data stored on a smart card. The **Administrator** button on the lefthand banner accesses administrative functions. Two main administrator functions are available: one to move profiles to smart cards and one to create installation disks with profiles readily available. See Figure 12-22.

Exhibit 1019
Page 231 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

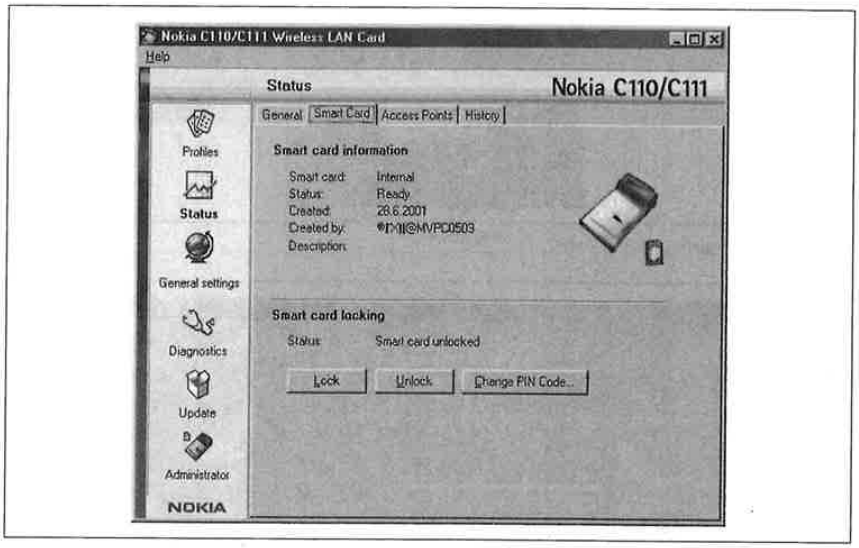


Figure 12-21. Smart-card management tab

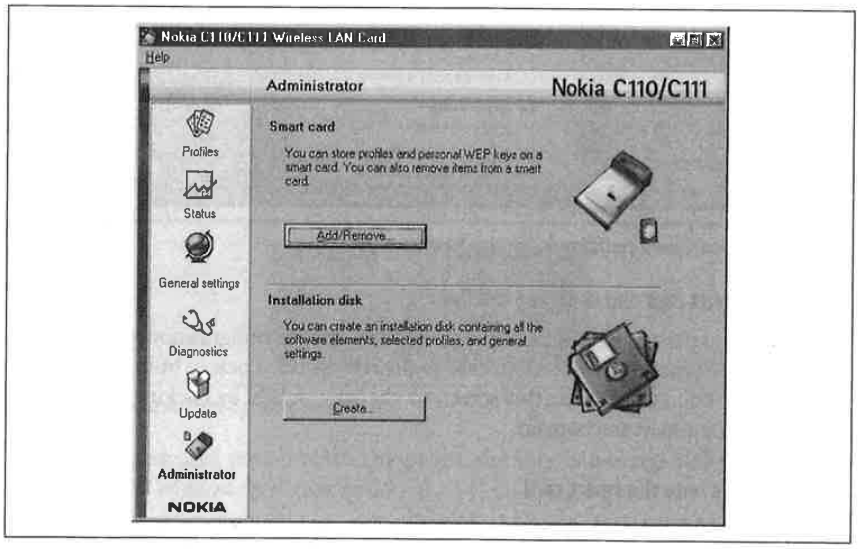


Figure 12-22. Administrator functions of the C11x driver

To access the smart-card functions, click on the **Add/Remove...** button at the top of the page. This brings you to Figure 12-23, which allows you to erase the smart card and move profiles from disk to smart card and vice versa.

Exhibit 1019
Page 232 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

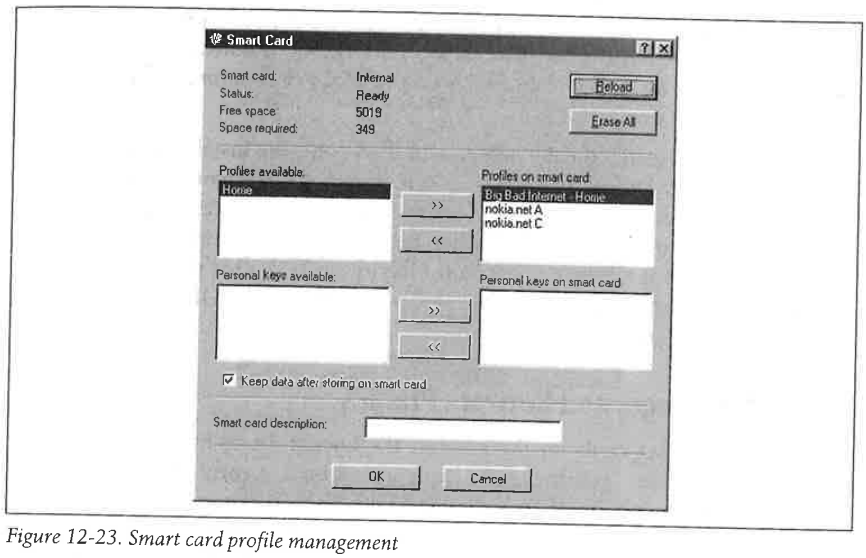


Figure 12-23. Smart card profile management

Lucent ORiNOCO

With the acquisition of WaveLAN, Lucent Technologies became an overnight leader in the wireless LAN market space. WaveLAN has been involved in wireless LAN development since the early 1990s. Naturally, standardized products were not available until the late 1990s with the adoption of the initial 802.11 standard. After experiencing financial trouble throughout much of 2001, Lucent's reorganization efforts have split the company into several pieces. As this book was written, the reorganization had yet to affect the physical appearance of the product—the card I ordered still had Lucent branding.

Lucent sells two flavors of the card. The cards are identical, except for the size of the WEP key supported. ORiNOCO Silver cards support WEP keys with 40 secret bits (marketed as 64-bit cards), and ORiNOCO Gold cards support WEP keys with 104 secret bits (marketed as 128-bit cards).

Installation

Lucent's card installation is more conventional. Begin by putting the card in an available PC Card slot and have the CD handy. When the card is inserted, Windows may identify it as a WaveLAN/IEEE card. Once 802.11 was ratified and products were brought to market, Lucent distinguished between the earlier proprietary cards and the 802.11-compliant cards by adding "IEEE" to the product name.

Exhibit 1019
Page 233 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Drivers for the ORiNOCO cards are bundled with a CD-ROM. Windows 2000 ships with Lucent drivers, but the bundled drivers have caused problems in a number of installations. Fetch an update from *http://www.orinocowireless.com* before inserting the card for the first time.

Allow Windows to search for the drivers and then point the installation program at the CD-ROM drive. Different versions of Windows have different drivers, so select the directory corresponding to your version of Windows (e.g., *D:\DRIVERS\WIN98*). Once the driver is installed in the network stack, you must install the Client Manager, which is the user frontend to the driver. The client manager is distributed on a CD with the card, and updates are available from the ORiNOCO web site at *http://www.orinocowireless.com*.

The Client Manager and Network Profiles

Like drivers for other cards on the market, the Lucent driver holds configuration information in profiles. Versions of the driver distributed in early 2001 were limited to four profiles, but fall 2001 revisions lifted that limit. This section covers the fall 2001 driver. After the driver and client manager are installed, you can create profiles to hold information about networks in the area. To get to the configuration, start the Client Manager. Figure 12-24 shows the main Client Manager screen.

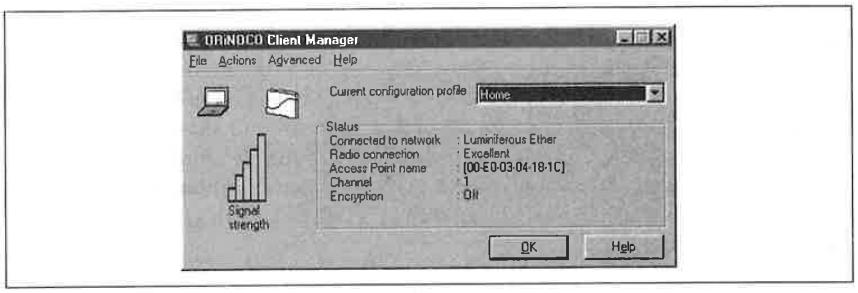


Figure 12-24. Client Manager main screen

Go to the **Actions** menu and select **Add/Edit Configuration Profile**. This brings up the main profile window, shown in Figure 12-25. Click on the **Add** button to begin creating the profile. The dialog box shown in Figure 12-26 appears. Name the profile on the lefthand side of the box and select the network type on the right. Access Point networks and Residential Gateways are infrastructure networks, and Peer-to-Peer networks are infrastructure networks.

Next, identify the network. As with other products, the network name is the SSID of the network. You can either type it into the network name field or use the Scan button to pull up a list of networks whose Beacons are currently being received.

Exhibit 1019
Page 234 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

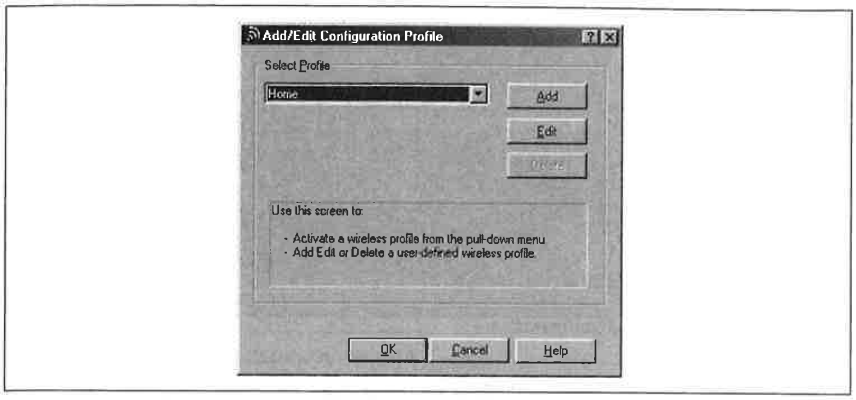


Figure 12-25. Add/Edit Configuration Profile window

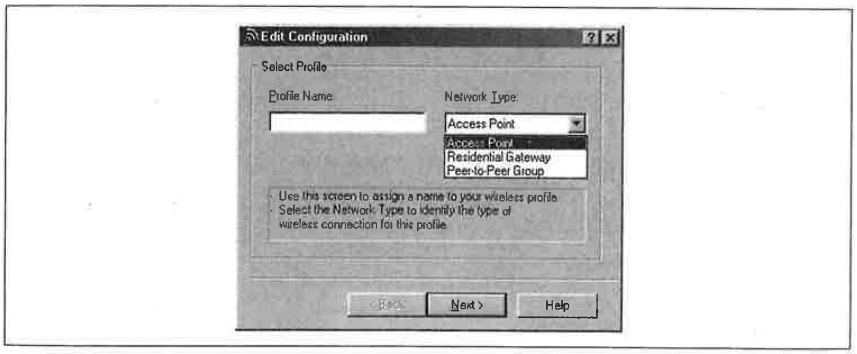


Figure 12-26. Profile creation

Figure 12-27 shows the network name configuration box, and Figure 12-28 shows the scan result box. Unfortunately, the scan result box gives no indication of signal strength.

Next, the user can configure WEP keys using the dialog box in Figure 12-29. Keys can be entered as alphanumeric strings and hashed into a bit string used as a key, or they can be entered directly. The algorithm used to generate a key from an alphanumeric string is not documented, which might lead to security questions. Much of the security of WEP resides in the key, and a simple key generator might compromise security by allowing dictionary attacks on the key.

Power management is controlled on a network-by-network basis and is configured after WEP using the dialog box in Figure 12-30.

The final item contained in a profile controls TCP/IP behavior and is set using the dialog box in Figure 12-31. The only option available with the Lucent driver is

Exhibit 1019
Page 235 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

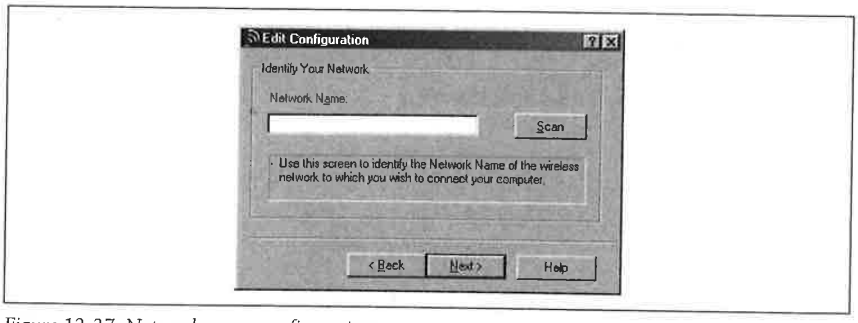


Figure 12-27. Network name configuration

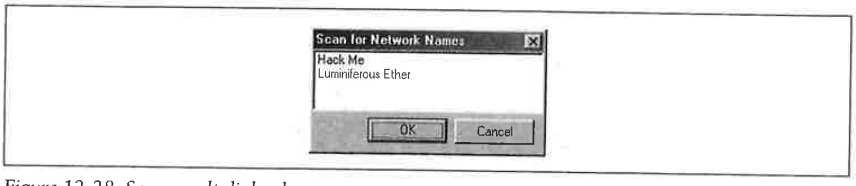


Figure 12-28. Scan result dialog box

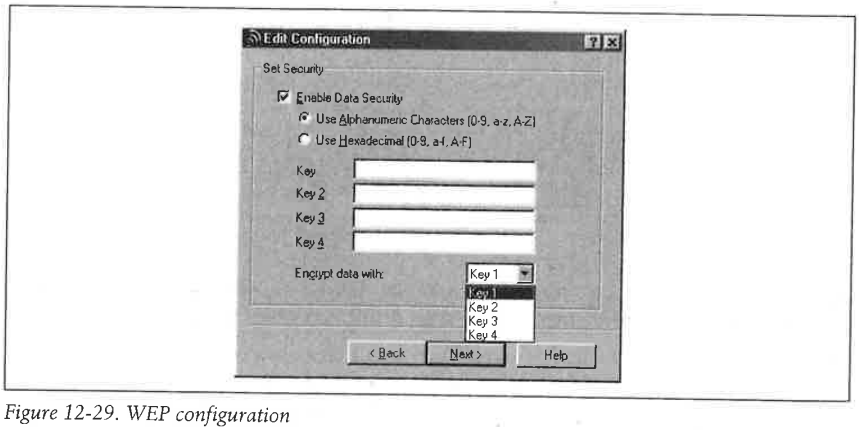


Figure 12-29. WEP configuration

whether a DHCP renewal will be issued when changing between profiles. No provisions are available for controlling Windows networking configuration.

Using the Driver

Miscellaneous system administration tasks are performed through the Client Manager. The radio can be disabled, even when the card is active, by going to the File menu and choosing Disable Radio.

Exhibit 1019
Page 236 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

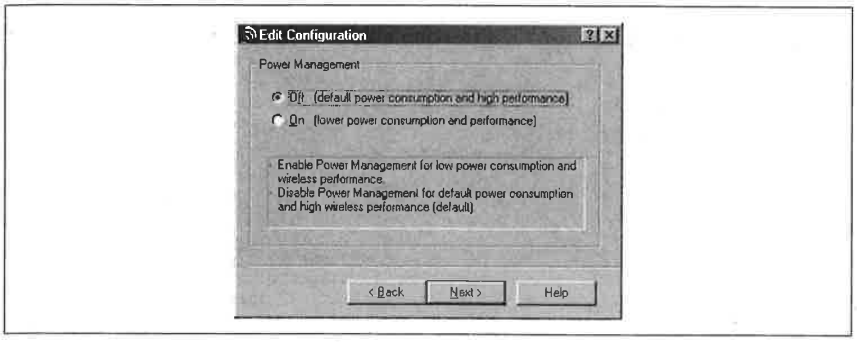


Figure 12-30. Power management configuration

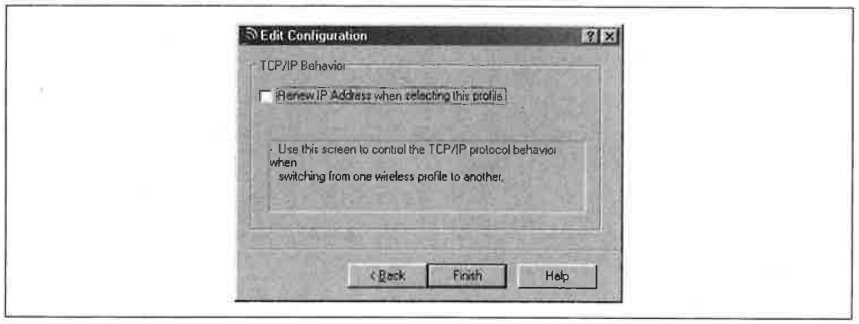


Figure 12-31. Network control

Changing between profiles

One of the most common configuration tasks is changing between profiles. The operating profile can be changed through the drop-down box on the right side of the Client Manager or through the Action menu. Changing profiles does not require a system reboot, even if the new profile requires an IP address renewal. See Figure 12-32.

Version information

Choosing Version Info from the Help menu brings up the version information. A version information window is shown in Figure 12-33. The driver and related software are implemented by several independent software pieces, all of which have their own version number. Most importantly, the driver for the card and the Client Manager user interface are separate and must be upgraded separately.

Exhibit 1019
Page 237 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

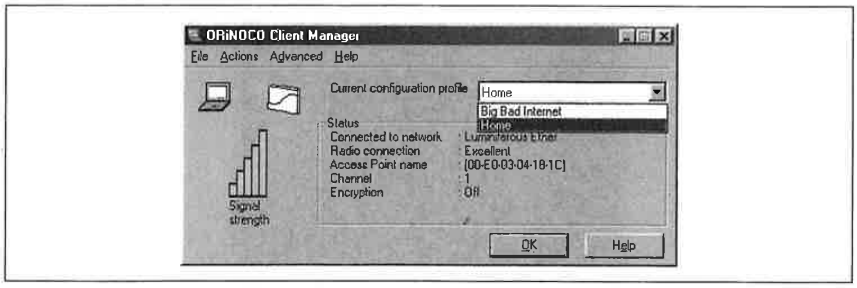


Figure 12-32. Changing profiles

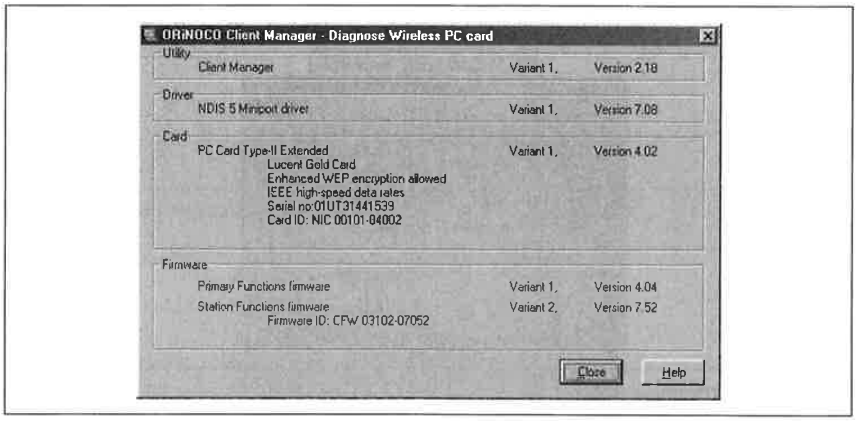


Figure 12-33. Version information

Site Monitor

The Site Monitor can be accessed by going to the Advanced Menu and choosing Site Monitor, which displays network information for open networks in the area. Networks that require WEP authentication are left off the list, which limits the use of Site Monitor to older networks not using WEP or networks with intentionally loose security, such as neighborhood or community networks.

If you are currently associated with an open network, the Site Monitor begins by displaying the AP list, which is shown in Figure 12-34. By default, the AP list shows only the signal-to-noise ratio (SNR), but it can be set to display the signal and noise figures themselves. One field not shown in the figure is the AP name, which is available only when the access point is also a Lucent access point.

Exhibit 1019
Page 238 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

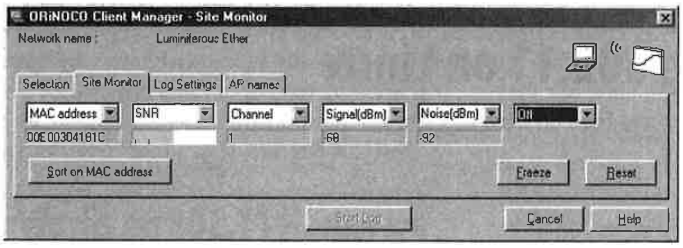


Figure 12-34. Network details

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

CHAPTER 13

Using 802.11 on Linux

Of the operating systems currently in wide use and active development, Unix environments offer the flexibility and stability required by power users and network administrators. When new hardware hits the market, the technical staff responsible for making purchase recommendations often asks about Linux support because of the additional functionality that can frequently be gained from the environment.

By now, though, the successful 802.11 hardware vendors have come to understand that Linux support increases sales and customer satisfaction without imposing significant additional costs. Linux friendliness among vendors was not always the case. One common strategy in past years was to release a fully featured driver as a binary module and a limited-functionality open source driver. What this strategy failed to take into account was the wide diversity of Linux hardware platforms and the difficulty in compiling and troubleshooting a closed-source binary driver without low-level documentation. Binary-only drivers are usually painful to install and troubleshoot, while open source counterparts are developed rapidly and are well-supported by the user community. Vendors that tried the two-driver approach often watched in shock as the open source driver quickly evolved to match the functionality of the closed-source driver, thanks to the magic of rapid collaborative development. Linux users have rejected the closed-source approach for a variety of reasons, and the marketplace has rejected products that do not have open source drivers available. This could be just a coincidence, but that is unlikely.

In recognition of the pain of installing a binary driver, this chapter describes only the open source solutions for 802.11 networking on Linux. Installing and troubleshooting open source drivers is possible on a wide variety of supported hardware, and it is a well-known procedure that many people use each day. Linux drivers for Lucent-based cards are now part of the standard Personal Computer Memory Card Industry Association (PCMCIA) distribution, and I expect that *linux-wlan* will eventually become part of the standard operating-system distribution. As with Windows drivers, installing wireless cards on Linux creates Ethernet interfaces. Many Linux drivers expose an Ethernet interface through the kernel. (Frequently, wireless interfaces

Exhibit 1019
Page 240 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

even have the *eth* prefix!) Programs can use the Ethernet interface to send and receive data at the link layer, and the driver handles Ethernet-to-802.11 conversions. Many of the things you would expect to see with an Ethernet interface remain the same. ARP works identically, and the IP configuration is done with the same utilities provided by the operating-system distribution. *ifconfig* can even be used to monitor the interface status and see the data sent and received.

A Few Words on 802.11 Hardware

As with other devices running under Linux, the more you know about the hardware, the better off you are. Only a handful of 802.11 chipset manufacturers exist. Most vendors use chipsets produced by Intersil (*http://www.intersil.com*, formerly known as Harris Semiconductor). Intersil's industry-leading position is the result of the success of its PRISM chipset. The initial PRISM, whose name is an acronym for Programmable Radio in the ISM band, was a common solution for vendors seeking a 2-Mbps DSSS 802.11 solution. When 802.11b was standardized in 1999, Intersil brought out the PRISM-2 chipset, which supported the 5.5-Mbps and 11-Mbps data rates.

Intersil's chipsets are used by Linksys, Nortel/Netgear, D-Link, and SMC for interface cards. Several new laptops with integrated 802.11 support are hitting the market, and many of these wireless-enabled laptops are powered by Intersil chipsets. Choosing between the supported Intersil-based cards is a personal trade-off between price, performance, range, and other factors.

Lucent's Hermes chipset is the major competitor to Intersil's PRISM. Unlike Intersil, Lucent also produces interface cards for the end user. Lucent's cards were first to market, so they are quite common. Several vendors seeking to bring a wireless solution to market chose OEM Lucent gear; the most notable example is Apple's AirPort, which is based on Lucent technology. OEM and rebranding relationships are quite common. Table 13-1 divides the major industry vendors by the radio chipset used in each vendor's cards, which should help you in choosing which driver to use. (In some cases, the OEM relationship is little more than a new sticker. Vendor B simply takes Vendor A's equipment and slaps a new logo on it.)

Table 13-1. The silicon behind the brand

Lucent chipset-based cards	Intersil PRISM-based cards
Lucent Wavelan/IEEE and Orinoco	Linksys
Cabletron/Enterasys RoamAbout	SMC
Apple AirPort	Compaq (WL100 and WL200)
Compaq (WL110, WL210, and WL215)	D-Link
IBM High Rate Wireless LAN	Nokia
HP 802.11b Wireless LAN	Cisco (Aironet)
Dell TrueMobile 1150 (earlier versions were OEM Aironet cards)	

Exhibit 1019
Page 241 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

In addition to the radio chipset, cards must have a MAC controller. Most cards on the market use an Intersil MAC controller. Several first-generation cards used an AMD Am930 MAC controller but have switched to the integrated MAC controller in the PRISM-2 chipset. Cisco's Aironet product line uses an Aironet-developed MAC controller with a PRISM-2 radio chipset.

The PC Card form factor is the dominant form factor, though it is not used exclusively. PC Cards can be used directly in portable computers and can be plugged into PCMCIA interfaces for other purposes. Many, but by no means all, access points use PC Card slots instead of fixed radio interfaces, which, conveniently, also allows for upgrades by swapping out the radio interface. Even PCI-based 802.11 solutions use PC Cards. Typically, a PCI solution consists of a PC Card plus a PCI carrier card with a PC Card interface.

PCMCIA Support on Linux

Most add-on 802.11 solutions for laptop computers are based on the PCMCIA form factor. Adding 802.11 support to Linux requires an understanding of how the PCMCIA subsystem in Linux is put together and how it works to enable drivers for PCMCIA cards.

PCMCIA Card Services Overview

Card Services grew out of an attempt to simplify system configuration. Rather than dedicating system resources to individual devices, the host system maintained a pool of resources for PC Cards and allocated resources as necessary. Figure 13-1 shows the procedure by which cards are configured on Linux.

When a card is inserted, the *cardmgr* process orchestrates the configuration of the device, as shown in Figure 13-1. The orchestration pulls together system resources, kernel components, and kernel driver modules through the configuration files stored in */etc/pcmcia*. Roughly speaking, the host takes the following steps:

1. A card is inserted into an empty PC Card socket, and *cardmgr* is notified of this event. In addition to any hardware operations (such as supplying power to the socket), *cardmgr* queries the *card information structure* (CIS) to determine the type of card inserted and the resources it needs. For more information on the CIS, see the sidebar "Card Information Structure."
2. *cardmgr* consults the card database stored in */etc/pcmcia/config* to determine which card was inserted. Part of the configuration is to associate cards with a *class*. For the purposes of configuring network cards, the important point to note is that items in the *network* class have additional network configuration operations performed on them later. The card is identified by the CIS data from step 1, and the class setting is set in the main system configuration file. At this point,

Exhibit 1019
Page 242 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

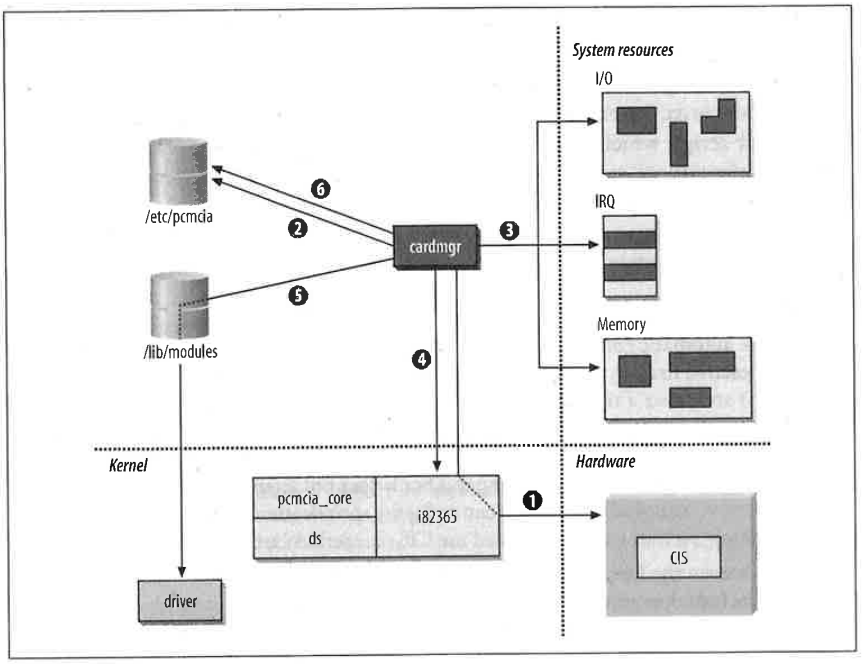


Figure 13-1. Linux PCMCIA configuration system

cardmgr beeps once. Successful identification results in a high-pitched beep; unsuccessful identifications are indicated by a beep of lower pitch.

3. *cardmgr* determines which resources are available to allocate to the card. Blocks of system resources are reserved for PCMCIA card use in the main configuration file, and *cardmgr* doles out resources as needed to cards. The number of I/O ports and the size of the memory window are obtained from the CIS.
4. Resources allocated by *cardmgr* are programmed into the PCMCIA controller, which is depicted in Figure 13-1 as interaction with the device driver. PCMCIA controllers implement *resource steering* to map resources required by the card onto available system resources. A card may ask for an interrupt, but the actual assigned interrupt is irrelevant. In operation, the card simply asks the PCMCIA controller to raise an interrupt, and the controller is responsible for looking up the interrupt assigned to the socket and firing the correct interrupt line.
5. Part of the configuration information obtained from the lookup in step 2 is the name of the device driver that should be loaded to use the newly inserted card. Drivers for PCMCIA cards are implemented as kernel modules. As part of the insertion process, the driver is informed of resources allocated in step 4. With

Exhibit 1019
Page 243 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

proper module dependencies, module stacking can be used to load multiple modules.

6. Further user-space configuration is performed based on the class of the device. Network cards, for example, have further configuration done by the */etc/pcmcia/network* script, which is configured by editing */etc/pcmcia/network.opts*. Successful configuration in this step generates a second high beep, and failure is reported with a low beep.

Card Information Structure

To enable automatic configuration, every PC Card has a blob of data that enables the card to describe itself to the host system. The blob is called the *card information structure* (CIS) and takes a relatively straightforward link-list format. The building blocks of the CIS are called *tuples* because they have three components: a type code to identify the type of tuple, a length field, and a series of data bytes. Tuple formats range from trivial to highly complex, which is why this book does not attempt to classify them any further. Brave, stout-hearted readers can order the specification from the PCMCIA and use the *dump_cis* tool on Linux to read the CIS of inserted cards.

The CIS assists the host in automatic configuration by reporting information about itself to the host operating system. For example, network interface cards identify themselves as such, and the CIS enables the Card Services software to allocate the appropriate resources such as I/O ports and interrupt request (IRQ) lines. On Linux, the system administrator uses configuration files to match the CIS data to the driver.

PCMCIA Card Services Installation

Installation of Card Services is documented in the PCMCIA-HOWTO. Both the software and documentation are available from *http://pcmcia-cs.sourceforge.net/*. Begin by unpacking the software and configuring it with *make config*.

Card Services asks for the current kernel source tree and picks relevant options out of it. In addition, it asks four questions. *Trusting* versions of the utilities allow operations by nonroot users, and are therefore set to no by default. *CardBus support* is included for newer cards, though no PRISM-2 cards currently use the CardBus interface. *PnP resource checking* is not required unless the computer hardware has plug-and-play hardware. Enabling this option can cause problems on computers that do not have PnP hardware and on several laptops. Finally, you must specify a module install directory. By default, modules are installed in */lib/modules/<version>/modules*, in which *<version>* is the version of the kernel source tree.

```
[root@bloodhound pcmcia-cs-3.1.28]# make config

    -------- Linux PCMCIA Configuration Script --------
```


Exhibit 1019
Page 244 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

```
The default responses for each question are correct for most users.
Consult the PCMCIA-HOWTO for additional info about each option.

Linux source directory [/usr/src/linux]:

The kernel source tree is version 2.2.19.
The current kernel build date is Sat Aug 25 07:11:52 2001.

Build 'trusting' versions of card utilities (y/n) [n]:
Include 32-bit (CardBus) card support (y/n) [y]:
Include PnP BIOS resource checking (y/n) [n]:
Module install directory [/lib/modules/2.2.19]:

Kernel configuration options:
    Symmetric multiprocessing support is disabled.
    PCI BIOS support is enabled.
    Power management (APM) support is enabled.
    SCSI support is disabled.
    IEEE 1394 (FireWire) support is disabled.
    Networking support is enabled.
     Radio network interface support is enabled.
     Token Ring device support is disabled.
     Fast switching is disabled.
     Frame Diverter is disabled.
    Module version checking is enabled.
    Kernel debugging support is disabled.
    /proc filesystem support is enabled.
    Maximum physical memory: 1GB

It looks like you have a System V init file setup.

X Windows include files found.
Forms library not installed.
    If you wish to build the 'cardinfo' control panel, you need the Forms
    library and the X Windows include files.  See the HOWTO for details.

Configuration successful.
```

After configuration, the installation can be accomplished with *make install*. Most configuration information is automatically detected from the environment without difficulty, though you need to set the PCIC environment variable to the correct type of PCMCIA controller. Two major types of controller are supported: the Databook TCIC-2 and compatibles, and the Intel i82365SL and compatibles. On RedHat and derived distributions, the variable is set in */etc/sysconfig/pcmcia*. Nearly all systems should set it to *i82365*.

Troubleshooting Resource Conflicts

One of the revolutionary developments hyped by PCMCIA card vendors was that users were no longer directly responsible for maintaining low-level hardware configurations on IBM-compatible hardware. In many respects, this hype was drastically

Exhibit 1019
Page 245 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

overblown because users are still responsible for maintaining the resource pools used by PCMCIA Card Services to draw from for automatic configuration, and therefore they must still be familiar with the hardware configuration. Three major resources are managed by Card Services for users: IRQ lines, I/O ports, and direct memory access (DMA) channels. DMA channels are not required by network cards, however, and are not discussed in this section.

IRQs

IRQs are used by devices that must use of the CPU periodically. Interfaces use IRQs so that when a buffer fills, the system CPU can be notified and drain the buffer. One limitation of the PC architecture is that it has only 15 available IRQs, and many are occupied by standard hardware. Table 13-2 shows common IRQ usage, which may help you determine which IRQs are available for PCMCIA cards. Disabling any extra components frees the IRQ. Table 13-2 also shows common IRQ settings on PC hardware. As a rule of thumb, IRQs 3, 5, and 10 are readily available on most machines.

Table 13-2. Common IRQ settings

IRQ number	Common usage	Purpose
0	System timer	Fires 18 times per second to maintain coarse clocking.
1	Keyboard	Allows operating system to monitor keyboard strokes by user.
2	Cascade	Two interrupt controller chips are used; the second controls IRQs 8–15 and is wired into IRQ 2 on the primary.
3	Second serial port	The second and fourth serial ports (COM2 and COM4 under Windows) both use IRQ 3. If only one serial port is present, IRQ 3 may be used by expansion devices.
4	First/third serial port	The first and third serial ports (COM1 and COM3 under Windows) both use IRQ 4. Generally, it is not a good idea to use IRQ 4 for expansion devices because loss of the serial port also means that terminal-emulation software cannot be used.
5	Second parallel port	Most systems have only one parallel port, but IRQ 5 is also commonly used for sound cards.
6	Floppy controller	All systems have floppy disks, which can be especially important on portable computers.
7	First parallel port	The first parallel port can frequently be disabled on laptops without an issue, unless the parallel port is used extensively for printing.
8	RTC	The Real-Time Clock maintains finer-grained timers
9	Video (older systems)	Older systems required an IRQ for the video controller, and it was typically assigned to IRQ 9. Most video controllers are now on the PCI bus and do not require a dedicated IRQ.
10		Usually available for expansion devices.
11	Usually PCI bus or SCSI controller	Generally not available for expansion devices.
12	Usually PS/2 mouse port	Generally not available.

Exhibit 1019
Page 246 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Table 13-2. Common IRQ settings (continued)

IRQ number	Common usage	Purpose
13	FPU	The floating-point unit IRQ is used by the math coprocessor, even on systems with a CPU with an integrated math coprocessor such as the Pentium series.
14	Primary IDE	The first IDE channel is used by the main system hard disk, and thus IRQ 14 is almost never available on a portable system.
15	Secondary IDE	Portable systems typically place the CD-ROM on IRQ 15, making this IRQ unavailable for use.

I/O ports

I/O addresses are used for bidirectional communication between the system and a peripheral device. They tend to be somewhat poorly organized, and many devices have overlapping defaults. Each I/O port can be used to transfer a byte between the peripheral device and the CPU. Most devices require the ability to transfer multiple bytes at a time, so a block of ports is assigned to the device. The lowest port number is also called the base I/O address. A second parameter describes the size of the I/O window. Table 13-3 lists some of the common port assignments. Refer to your hardware vendor's documentation for details on additional devices such as IR ports, USB controllers, the PCMCIA controller, and any resources that may be required by motherboard components.

Table 13-3. Common I/O ports

Device name	I/O range (size)
Communication ports	
First parallel port	0x3bc–0x3bf (4)
First serial port	0x3f8–0x3ff (8)
Second serial port	0x2f8–0x2ff (8)
Disk drives	
Primary IDE	master: 0x1f0–0x1f7 (8)
	slave: 0x3f6–0x3f7 (2)
Secondary IDE	master: 0x170–0x177 (8)
	slave: 0x376 (1)
Floppy controller	0x3f0–0x3f5 (6)
Input devices	
Keyboard	0x060 (1)
	0x064 (1)
Multimedia/gaming	
Sound card	0x220–0x22f (16)
	FM Synth: 0x388–0x38b (4)
	MIDI: 0x330–0x331 (2)

Exhibit 1019
Page 247 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Table 13-3. Common I/O ports (continued)

Device name	I/O range (size)
Joystick/Game port	0x200–0x207 (8)
System devices	
Interrupt controllers	0x020–0x021 (2)
	0x0a0–0x0a1 (2)
DMA controllers	DMA channels 0-3: 0x000–0x00f (16)
	Page registers: 0x080–0x08f (16)
	DMA channels 4-7: 0x0c0–0x0df
CMOS/real time clock	0x070–0x073 (4)
Speaker	0x061
Math coprocessor	0x0f0–0x0ff (16)

linux-wlan-ng for Intersil-Based Cards

The most commonly used driver for PRISM-based cards is the *linux-wlan* open source driver developed by Absolute Value Systems. In a shining example for other vendors, Intersil has actively supported the *linux-wlan* project. Two subprojects exist. The original *linux-wlan* code supports cards based on the original PRISM chipset, which generally run at 2 Mbps, and the newer *linux-wlan-ng* codebase supports cards based on the PRISM-2 chipset running at 11 Mbps. *linux-wlan-ng* supports most cards based on the PRISM-2 chipset.* The examples in this section were written with a Linksys WPC11 as the example card, largely because of its affordability relative to other PRISM-2 cards.

At this point, most users will be interested in 11-Mbps cards supported by *linux-wlan-ng*. Both projects aim toward developing a complete 802.11 layer for Linux. Currently, both projects support simple encapsulation of 802.11 frames and RFC 1042 encapsulation. In practice, this poses a problem only with strange access points that support only 802.1h translation.

Prerequisites

Before starting the installation of *linux-wlan-ng*, there are some housekeeping tasks to take care of. Compile-time configuration is taken care of in part by pulling some configuration information out of the kernel configuration and the PCMCIA utility configuration. Most distributions include the option of installing source code but do

* Nokia's C11x cards are based on the PRISM-2, but the addition of the smart-card reader changed the programming interface enough so that the C11x is not supported by *linux-wlan-ng*.

Exhibit 1019
Page 248 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

not include the configuration used to build the kernel or any of the additional software packages.

Officially, *linux-wlan* is developed against kernels later than 2.0.36 and PCMCIA Card Services (*pcmcia-cs*) later than 3.0.9, but *linux-wlan-ng* was largely developed against kernel 2.2 and PCMCIA services 3.1. Compiling *linux-wlan-ng* requires configured source trees for both the kernel and PCMCIA, so during the preinstallation process, it is a good idea to update both components to the latest versions available. Refer to the documentation bundled with *linux-wlan-ng* to get an idea of the software environments it has been tested against and upgrade to current versions. Kernel configuration and compilation has been extensively documented elsewhere; refer to the documentation for your distribution. Card Services software is now hosted at *http://pcmcia-cs.sourceforge.net*, and documentation is bundled with the source package. The examples in this section were written using kernel 2.2.19 and *pcmcia-cs* 3.1.28, though there is nothing specific about them.

Kernel compilation

Because the configuration in the kernel source tree is used by the subsequent configuration of both PCMCIA Card Services and *linux-wlan*, it must match the currently running kernel. Many problems can be avoided, and countless hours can be saved if you begin the process by downloading clean kernel source and configuring from scratch. *linux-wlan* does not require any specific options for support other than general network support, but it is useful for other reasons to enable certain options. The ISC Dynamic Host Configuration Protocol (DHCP) client, for example, requires that both the packet socket (*CONFIG_PACKET*) and socket filtering (*CONFIG_FILTER*) be enabled. Some applications of the driver also are enabled by the kernel/user network link driver (*CONFIG_NETLINK*). Kernel configuration and compilation is extensively documented in both online and offline resources and may have been installed with your documentation. Kernel source code is available from *ftp://ftp.kernel.org* and countless mirror sites worldwide.

Compiling and Installing linux-wlan-ng

The first step is to get the software from *http://www.linux-wlan.org*. Version numbering follows the Linux kernel convention, with odd minor numbers used for development versions and even minor numbers used for stable production versions. The software can be obtained by FTP from *ftp://ftp.linux-wlan.org/pub/linux-wlan-ng/*; the latest version as this book was written was 0.1.8-pre13.

Compile-time configuration

Run *make config* to configure the software. A configuration script is run, and you need to supply the location of your configured kernel source and PCMCIA Card Services

Exhibit 1019
Page 249 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

source trees. A few minor questions follow; these guidelines may help you answer those questions:

- Building with debugging information compiled in is a good idea when you're troubleshooting. Debugging can be disabled easily, and the code is a minuscule performance hit when compiled in and not used.
- Kernel PCMCIA is the PCMCIA software included with the 2.4 kernel series. Users of kernel Version 2.4 need to choose whether to use Kernel PCMCIA or PCMCIA Card Services; users of earlier kernels must use PCMCIA Card Services.
- PLX-based adapters are PCI wireless adapters. Most users will use PCMCIA cards and can disable PLX support.
- Card Services drivers support PCMCIA cards and should be enabled.

A sample configuration looks something like this:

```
[gast@bloodhound linux-wlan-ng-0.1.8-pre13]$ make config
-------------- Linux WLAN Configuration Script -------------
The default responses are correct for most users.
Linux source directory [/usr/src/linux]:
The kernel source tree is version 2.2.19.
The current kernel build date is Sat Aug 25 07:11:52 2001.
pcmcia-cs source dir [/usr/src/pcmcia-cs-3.1.28]:
Alternate target install root directory on host []:
  Module install directory [/lib/modules/2.2.19]:
PCMCIA script directory [/etc/pcmcia]:
It looks like you have a System V init file setup.
Target Architecture? (i386, ppc, or alpha) [i386]:
Prefix for build host compiler? (rarely needed) []:
Compiling with a cross compiler? (y/n) [n]:
Build for debugging (see doc/config.debug) (y/n) [n]: y
Build for Kernel PCMCIA? (y/n) [n]:
Build PLX???? based PCI (_plx) adapter drivers? (y/n) [n]:
Build PCMCIA Card Services (_cs) drivers? (y/n) [y]:
Configuration successful.
```

Building the software

Run *make all* to build the software. After some compilation messages, four main components are compiled:

wlanctl-ng
: The general administrative utility used to control the current running configuration and change the state of the hardware.

wlancfg
: Used to change values in the management information base and alter the configuration.

Exhibit 1019
Page 250 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

p80211.o

A generic 802.11 utility layer for Linux, implemented as a kernel module. It includes data structures required to handle buffers, build frames, translate between the supported frame types (RFC 1042-encapsulated, 802.1h, vanilla encapsulated, and the raw packet buffers used by the kernel), interact with the network devices in the kernel, and perform the functions placed in the MAC Layer Management Entity (MLME).

prism2_cs.o

The kernel module driver for PRISM-2–based cards. It implements the hardware interface to the Intersil HFA384x chipset used in supported cards, implements hardware interrupt service, allows user-space utilities to gather statistics and counters data from the hardware, and enables the management operations discussed in Chapter 7, such as scanning, joining, authentication, and association.

Installing the software

After compilation, become *root* and run *make install* to install the software. By default, the two administration programs (*wlanctl-ng* and *wlancfg*) are placed in */sbin*, and the two modules are installed in the module directory tree that corresponds to your current kernel version. On kernel Version 2.2.19, for example, *p80211.o* would be installed in */lib/modules/2.2.19/net* and *prism2_cs.o* would be placed in */lib/modules/2.2.19/pcmcia*.

The installation also places some configuration files in the main PCMCIA Card Services directory, which is */etc/pcmcia* by default. *wlan-ng.conf* contains card definitions for hardware known to work with *linux-wlan-ng*. It is copied to the Card Services directory, and a directive is put at the end of the main card definition file, */etc/pcmcia/config*, to include the PRISM-2 card definitions. The configuration script for wireless interfaces, *wlan-ng*, is also placed in */etc/pcmcia*. *wlan-ng* pulls in configuration options from an auxiliary option file, *wlan-ng.opts*. If either *wlan-ng* or *wlan-ng.opts* exists, the older versions will be backed up and will receive the suffix *.O*.

Administration of the 802.11 Interface

The main administrative tool in *linux-wlan-ng* is *wlanctl-ng*, a tool to control the internal state of the driver through *ioctl()* commands. As an example, consider the *wlanctl-ng* command used to join a network:

```
[root@bloodhound]# /sbin/wlanctl-ng wlan0 lnxreq_autojoin ssid="Luminferous Ether" \
authtype=opensystem
message=lnxreq_autojoin
  ssid=Luminferous Ether
  authtype=opensystem
  resultcode=success
```


Exhibit 1019
Page 251 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Commands operate on an interface, which is supplied as the first argument. The command to perform is the second argument. In the join command, two additional arguments are required. One is the SSID, so the kernel driver can identify the Beacon frames corresponding to the desired network. SSIDs with spaces can be used, provided that quotation marks are used to delimit the SSID. The other argument is the authentication type. Shared-key authentication is required if the WEP key is set.

wlanctl-ng commands fall into three broad classes. Commands that begin with *dot11req_*, such as *dot11req_mibset*, work with data structures specified in 802.11. The second class of commands, which begin with *p2req_*, work with the PRISM-2 chipset. A third class, prefaced with *lnxreq_*, are commands specific to the Linux driver.

WEP keys are set by writing the MIB variables corresponding to the default key on the interface the key must be set for:

```
[root@bloodhound]# wlanctl-ng wlan0 dot11req_mibset \
mibattribute=dot11WEPDefaultKey3=01:02:03:04:05
message=dot11req_mibset
  mibattribute=dot11WEPDefaultKey3=01:02:03:04:05
  resultcode=success
```

Some commands are not implemented and return a result code of *not implemented*. One such command is *dot11req_scan*, which should run a generalized active scan.

One command that network administrators are likely to find near and dear is the command that enables promiscuous mode packet capture.* The driver command is *lnxreq_wlansniff*, and it takes two additional arguments. One is the channel to sniff on, and the final argument is the enable flag, which can be used to enable and disable the sniffing functionality.

```
[root@bloodhound]# wlanctl-ng wlan0 lnxreq_wlansniff channel=1 enable=true
message=lnxreq_wlansniff
  enable=true
  channel=1
  resultcode=success
```

For convenience, I have written shell functions around the *lnxreq_wlansniff* command to cut down on typing. Both shell functions take two arguments: the interface name and the channel. Some users may not use multiple 802.11 cards and may choose to have the shell function take just one argument instead. In any case, here are the functions I use, which you are free to customize:

```
wlan-promisc ()
{
  /sbin/wlanctl-ng $1 lnxreq_wlansniff channel=$2 enable=true
}
```

* Promiscuous mode packet capture was disabled in *linux-wlan*-0.1.7 and later, but patches to enable the functionality are available. See Chapter 16 for details.

Exhibit 1019
Page 252 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

```
wlan-normal ()
{
  /sbin/wlanctl-ng $1 lnxreq_wlansniff channel=$2 enable=false
}
```

To activate promiscuous packet capture on channel 5, I run *wlan-promisc wlan0 5*. When promiscuous capture is activated, it is not possible to use the network for communication. To restore network connectivity, simply run *wlan-normal wlan0 5*. With only one wireless interface, the shell functions can be simplified even further by hardcoding *wlan0* as the interface used by the function and taking only the channel number as an argument.

IP addresing

Setting IP addresses is distribution-specific. On RedHat and derivative distributions, including Mandrake, configuration is stored in */etc/sysconfig/network-scripts*, with data for each interface shown in *ifcfg-interface*. *linux-wlan-ng* interfaces begin with *wlan*: *wlan0*, *wlan1*, and so forth. An example static IP configuration for the first wireless LAN interface would hold IP address and netmask information just as with any other interface:

```
[root@bloodhound network-scripts]# cat ifcfg-wlan0
DEVICE=wlan0
BOOTPROTO=static
IPADDR=192.168.200.125
NETMASK=255.255.255.0
NETWORK=192.168.200.0
BROADCAST=192.168.200.255
ONBOOT=no
```

Bringing up the interface is a matter of running *ifup* with the interface name as an argument as *root*. Most Linux kernels are built without bridging support and cannot by default be configured with two interfaces on the same logical IP subnet. This is often a danger for stations with wireless LAN interfaces because the wireless interface is sometimes used to replace an existing wired interface.

```
[root@bloodhound network-scripts]# ifup wlan0
```

Dynamic IP configuration with DHCP is allowed; refer to your distribution's documentation for details on enabling dynamic boot protocols. DHCP configuration depends on forming a successful association with an access point. If the association fails, the DHCP discovery packets result in "not associated" error messages in the log.

Configuring linux-wlan-ng

Card definitions are stored in */etc/pcmcia/config*. At the end of the file, there is a line that reads *source ./*.conf*. This line enables additional PCMCIA driver packages to add card definitions. *linux-wlan-ng* adds the file *wlan-ng.conf*, which is a series of

Exhibit 1019
Page 253 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

driver associations. As an example, consider the definition of the Linksys WPC11 card:

```
card "Linksys WPC11 11Mbps 802.11b WLAN Card"
   version "Instant Wireless ", " Network PC CARD", "Version 01.02"
   bind "prism2_cs"
```

The first line is a text string used to identify the card to the human user, and the *bind* directive specifies which driver will be loaded to support the card. The *version* strings are used to match the CIS of the card, which can be viewed using the *dump_cis* utility:

```
[root@bloodhound]# dump_cis
Socket 0:
  dev_info
    NULL 0ns, 512b
  attr_dev_info
    SRAM 500ns, 1kb
  vers_1 5.0, "Instant Wireless ", " Network PC CARD", "Version 01.02",
    ""
  manfid 0x0156, 0x002
  funcid network_adapter
  lan_technology wireless
  lan_speed 1 mb/sec
  lan_speed 2 mb/sec
  lan_speed 5 mb/sec
  lan_speed 11 mb/sec
  lan_media 2.4_GHz
  lan_node_id 00 04 5a 0d 0c 70
  lan_connector Closed connector standard
  config base 0x03e0 mask 0x0001 last_index 0x01
  cftable_entry 0x01 [default]
    Vcc Vmin 4750mV Vmax 5250mV Iavg 300mA Ipeak 300mA
    Idown 10mA
    io 0x0000-0x003f [lines=6] [16bit]
    irq mask 0xffff [level] [pulse]

Socket 1:
  no CIS present
```

Note that the CIS specifies how large an I/O window is necessary. In this case, 64 (0x3F) bytes of I/O space are required by the card. When the card is inserted, the kernel configuration prints out the resources assigned to the driver in */var/log/kernel/info*:

```
Oct 26 21:03:38 bloodhound kernel: prism2_cs: index 0x01: Vcc 5.0, irq 10, io
0x0100-0x13f
```

In addition to the card associations in *wlan-ng.conf*, the configuration script *wlan-ng* and its option file *wlan-ng.opts* are installed in */etc/pcmcia*. *wlan-ng* is called whenever a PRISM-2 wireless card is inserted. It identifies the card, associates with an access point or starts an independent BSS if desired, and initializes the network interface using whatever hooks are provided by the Linux distribution. *wlan-ng.opts* is used to configure the desired SSID and WEP keys.

Exhibit 1019
Page 254 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Selecting a network to join

To configure the network to join, some parameters in *wlan-ng.opts* must be changed. One section of the script is labeled "STA START" and contains the network parameters AuthType and DesiredSSID:

```
#========STA START=====================================
AuthType="opensystem"
DesiredSSID="linux-wlan"
```

AuthType can be set to *opensystem* for a network not employing access control, and *sharedkey* for networks that use the shared-key authentication mechanism of WEP. The *DesiredSSID* parameter must be set to the SSID announced by the Beacon frames for the network that the station will join.

Configuring WEP

Stations that use WEP must also configure WEP-related parameters in *wlan-ng.opts*:

```
#========WEP===========================================
dot11PrivacyInvoked=true
dot11WEPDefaultKeyID=2

dot11WEPDefaultKey0=01:02:03:04:05
dot11WEPDefaultKey1=01:02:03:04:05:06:07:08:09:0a:0b:0c:0d
dot11WEPDefaultKey2=
dot11WEPDefaultKey3=
```

Several settings matter only for access points; the settings that matter for stations are:

dot11PrivacyInvoked
: This must be set to `true` to enable WEP. If it is set to `false`, WEP is disabled.

dot11WEPDefaultKeyID
: This is set to 0, 1, 2, or 3, depending on which of the WEP keys are used on the network.

Four dot11WEPDefaultKeyXs (in which X is 0, 1, 2, or 3)
: These are used to program the WEP keys themselves. Keys are entered like MAC addresses, in which each byte is entered as a two-character hexadecimal number, and the bytes are separated by colons.

Enabling debugging output

If the *linux-wlan-ng* module was built with debugging code enabled, the PCMCIA Card Services system can be set up to pass configuration options to the driver. Module options are configured in */etc/pcmcia/config.opts*, using the *opts* parameter to the *module* directive, like this:

```
module "prism2_cs" opts "prism2_debug=3"
```

Debugging can be enabled for both the protocol module (*p80211*) and the PRISM-2 hardware driver (*prism2_cs*). Debugging of the protocol module is enabled with the

Exhibit 1019
Page 255 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

wlan_debug option, and debugging of the driver is enabled with the *prism2_debug* option. Five debugging levels are defined, with higher levels increasing the log output; Table 13-4 describes them in detail. Debugging levels are cumulative, so level 3 includes all messages from level 2 plus additional information.

Table 13-4. linux-wlan-ng debug levels

Level	Meaning
1	Error messages for rare exceptions.
2	More exceptions are logged.
3	Basic status output.
4	Additional status output.
5	Function entry and exit.

After making changes to *config.opts*, the card manager must be restarted so that the options are passed to the driver when the card is inserted. The easiest way to restart *cardmgr* is to send it the HUP signal:

```
kill -HUP `ps aux | grep cardmgr | grep -v grep | awk '{ print $2 }'`
```

Using linux-wlan-ng

Using wireless LAN interfaces driven by *linux-wlan-ng* is no different from using regular wired Ethernet interfaces. Configuration is identical to wired interfaces, and operations are similar. System administrators can, for the most part, treat 802.11 interfaces like Ethernet interfaces, but with higher latency. The higher latency is due partly to the lower bit rate of the wireless medium and partly to the requirement for fully acknowledged frame exchanges.

Common Problems

Most common problems are really resource conflicts, but resource conflicts may manifest themselves as a variety of error messages. Tracking down resource conflicts is a large task described in more detail in the section "Troubleshooting Resource Conflicts." Resource conflict tracking is often a tedious process of listing out the resources commonly used by a system and attempting to identify the conflict. It can be particularly annoying on laptop hardware because of the number of built-in devices that may be sucking up resources without extensive notification. This section describes the common problems that users may face before the next section delves into the hairy business of resolving resource conflicts.

Compilation difficulties

Most compilation difficulties can be traced to relying on an improper source tree for a dependent package. The most reliable way to install *linux-wlan-ng* is to first build

Exhibit 1019
Page 256 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

both a new kernel and new PCMCIA package from a clean source before attempting to build *linux-wlan-ng*. Although this takes time, it does guarantee that the compile-time configuration stored in the source tree matches the running version, and it will, therefore, eliminate most compilation problems.

Cards are identified as "anonymous memory card"

Card Services on Linux attempts to identify cards based on configuration information stored on the card itself in the CIS on the PCMCIA card. If Card Services is unable to identify a card, as a last resort, it identifies a card as an anonymous memory card. This error message almost always means that there is a resource conflict; as a result of the conflict, Card Services software is unable to communicate with the card to read the identifying data structures on the card. Vary the resource exclusion ranges to resolve the conflict; see "Troubleshooting Resource Conflicts" earlier in this chapter.

"Tx attempt prior to association, frame dropped" error message

This message is quite self-explanatory: a frame was queued for transmission before the station successfully associated with an access point. Several things might cause this error:

- If the desired SSID is not found, no association is made.
- Authentication is a precondition of association. If the authentication type is mismatched or the WEP key used for authentication is incorrect, the association fails.
- Resource conflicts can interfere with the sending and receiving of frames, which may cause transmissions (and therefore authentication or associations) to fail. Reconsider resource allocations and PCMCIA Card Services configuration.

Odd behavior from the lights on the card

linux-wlan-ng does not try to control the lights on the card in a consistent way between different card vendors. Generally speaking, one light becomes solid when the station associates with an access point, and a second light blinks to indicate traffic. If your card behaves differently, don't panic. Check at the access point for an association; if an association exists, then the card is functioning normally but uses the lights in a different manner. Naturally, the truly brave may attempt to rewrite the driver, but for most of the world, it is easier simply to note the behavior of the lights.

Dropped sessions

Associations may be dropped for a variety of reasons, many of which are not the fault of the client wireless LAN card. One frequent cause of this behavior is access

Exhibit 1019
Page 257 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

points that conserve resources by timing out idle associations. Nokia access points are notable for this behavior.

The original *linux-wlan* package for PRISM-1–based cards did not reassociate. *linux-wlan-ng* attempts to reassociate when an association times out, but it may not always be successful. If timeouts must always be avoided, run a script on the wireless station to send traffic periodically to prevent the association from timing out.

"MAC initialization failure, result=xxx"

The first major task for the PRISM-2 driver is to initialize the MAC controller chip on the card. The error code is one major clue. Positive result codes indicate that the firmware is at fault, and negative error codes are due to driver faults. Frequently, the error is due to a timeout period expiring; retry the operation to ensure that the error is caused by the system and not by a timeout expiration. Firmware problems are rare and generally seen only when the firmware loaded on the card was bad or the firmware load was incomplete. Driver problems are usually due to resource conflicts, especially with I/O ports. In some cases, a laptop with 5-volt PCMCIA slots may experience problems with a 3-volt card, though the failure usually happens later in the initialization process. I saw this message on an IBM ThinkPad T21 when the card was inserted, and the system was on battery power. If the T21 was plugged into AC power, however, the driver would load and configure without a hitch.

Agere (Lucent) Orinoco

Wireless networking is far older than the 802.11 standard. The first commercial wireless network hardware was manufactured by the WaveLAN division of AT&T and gained market acceptance in the early 1990s. When Lucent was spun off from AT&T in the late 1990s, the WaveLAN division, like most communications product manufacturing at AT&T, was made a part of Lucent.

Early WaveLAN hardware was a completely proprietary system. After 802.11 was finally standardized in 1997, new hardware that complied with the standard was sold under the WaveLAN brand. To distinguish the standards-compliant cards from the proprietary cards, the former were called "WaveLAN IEEE" cards, while the latter were simply "WaveLAN" cards.

As the market for 802.11 hardware continued to develop, Lucent decided to rename the WaveLAN division under another brand name. The new name, Orinoco, comes from the third largest river system in the world. During the rainy season in South America, the Orinoco swells with fresh rainfall flowing in from over 200 tributaries. At its peak, the river grows to over 10 miles wide and more than 300 feet deep. Nearly 1,000 miles of the Orinoco's 1,300 miles are navigable; it is no wonder that the river's name is derived from the native words for "a place to paddle."

Exhibit 1019
Page 258 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Lucent's initial strategy for support on open source platforms was to offer a choice of drivers. Two drivers were available. A closed-source proprietary binary driver, *wavelan2_cs*, provided full functionality largely equivalent to the drivers available for other platforms. A second, less functional open source driver, *wvlan_cs*, was made available under the GPL. For a variety of reasons, the proprietary driver was shunned in favor of the open source driver, and a devoted group of programmers enhanced the open source driver until it was equivalent to the proprietary driver.

Two open source drivers are now available for Lucent cards on Linux, depending on the kernel version employed. Kernel Versions 2.0 and 2.2 use the older *wvlan_cs* driver bundled with the *pcmcia-cs* package; kernel Version 2.4 uses the *orinoco_cs* driver in the kernel. The older *wvlan_cs* driver was based on a low-level library provided by Lucent, which was difficult to maintain and was a recurring source of bugs. Rather than continuing along an evolutionary dead-end, the maintainers of *wvlan_cs* learned from the *wlan-ng* driver and rewrote the lower layers of *wvlan_cs*. The result was *orinoco_cs*, a much more stable and robust driver. (*orinoco_cs* was originally known as *dldwd_cs*, which stood for David's Less Dodgy WaveLAN Driver!) *wvlan_cs* is no longer being maintained, and new installations should use *orinoco_cs* instead. Some Linux distributions may even ship with *orinoco_cs*.

Kernel 2.0/2.2: wvlan_cs

In kernel Version 2.2 and earlier, support for WaveLAN adapters is provided by the *wvlan_cs* driver bundled with PCMCIA Card Services for Linux.

Compiling and installing

Some wireless LAN support code is activated in the kernel by enabling the Wireless LAN extensions (*CONFIG_NET_RADIO*) kernel configuration option. *CONFIG_NET_RADIO*-enabled kernels collect wireless statistics and expose additional data structures used by the WaveLAN driver.

The driver for PCMCIA WaveLAN cards is part of the PCMCIA Card Services package. Previous sections of this chapter described how to compile and install the PCMCIA Card Services package and how PCMCIA cards are dynamically configured on Linux systems. After building a kernel with the wireless LAN extensions, you must rebuild the PCMCIA utilities as well.

The standard wireless tools are also required. They can be downloaded from *http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html*. Building the wireless tools is a straightforward task. After unzipping the file into a temporary directory, just run *make* in the root directory of the software distribution, and it will build the wireless tools.

Exhibit 1019
Page 259 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

PCMCIA configuration

Although the Orinoco card is rebadged on the outside, the CIS still identifies the card as a Lucent WaveLAN/IEEE, and the software is the same as it has always been for the WaveLAN/IEEE:

```
[root@bloodhound]# dump_cis
Socket 0:
  dev_info
    NULL 0ns, 512b
  attr_dev_info
    SRAM 500ns, 1kb
  vers_1 5.0, "Lucent Technologies", "WaveLAN/IEEE", "Version 01.01", ""
  manfid 0x0156, 0x002
  funcid network_adapter
  lan_technology wireless
  lan_speed 1 mb/sec
  lan_speed 2 mb/sec
  lan_speed 5 mb/sec
  lan_speed 11 mb/sec
  lan_media 2.4_GHz
  lan_node_id 00 02 2d 38 82 e4
  lan_connector Closed connector standard
  config base 0x03e0 mask 0x0001 last_index 0x01
  cftable_entry 0x01 [default]
    Vcc Vmin 4750mV Vmax 5250mV Iavg 300mA Ipeak 300mA
    Idown 10mA
    io 0x0000-0x003f [lines=6] [16bit]
    irq mask 0xffff [level] [pulse]

Socket 1:
  no CIS present
```

When the card is inserted, the */etc/pcmcia/wireless* script is run, using the configuration options in */etc/pcmcia/wireless.opts*. The wireless script is a frontend to the *iwconfig* program. Editing fields in *wireless.opts* sets the arguments to *iwconfig*. Therefore, knowing how *iwconfig* works allows you to set the appropriate fields in *wireless.opts*.

iwconfig

The main command-line tool for managing a WaveLAN wireless interface is *iwconfig*. When run with no parameters other than the interface name, *iwconfig* displays extended information about the radio interface, provided the kernel was built with radio extensions:

```
[root@bloodhound]# iwconfig wvlan0
wvlan0    IEEE 802.11-DS  ESSID:"Luminiferous Ether"  Nickname:"HERMES I"
          Mode:Managed  Frequency:2.457GHz  Access Point:00:E0:03:04:18:1C
          Bit Rate:2Mb/s   Tx-Power=15 dBm   Sensitivity:1/3
          RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
```


Exhibit 1019
Page 260 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

```
Link Quality:46/92 Signal level:-51 dBm  Noise level:-94 dBm
Rx invalid nwid:0  invalid crypt:0  invalid misc:0
```

The nickname is an internal name used by the driver. By default, it is set to "HERMES I" after the name of the Lucent chipset used in Lucent's wireless LAN cards.

Setting the network name

The basic task required to join a network is to select the appropriate network name, or SSID. *iwconfig* uses the *essid* parameter to set the desired network name. If the network name includes a space, it must be enclosed in quotation marks:

```
[root@bloodhound]# iwconfig wvlan0 essid "Luminiferous Ether"
```

Setting the network channel

The network-operating frequency can be selected in two ways. The *freq* parameter can take an operating frequency directly, or the *channel* parameter can be used with the appropriate channel number, and the driver will derive the frequency from the channel number. The following two commands are equivalent:

```
[root@bloodhound]# iwconfig wvlan0 freq 2.432G
[root@bloodhound]# iwconfig wvlan0 channel 4
```

Setting the network mode and associating with an access point

Most 802.11 stations are in either ad hoc networks or infrastructure networks. The *iwconfig* nomenclature for these two modes is *Ad-hoc* and *Managed*. Select between them by using the *mode* parameter:

```
[root@bloodhound]# iwconfig wvlan0 mode Ad-hoc
[root@bloodhound]# iwconfig wvlan0 mode Managed
```

For stations in an infrastructure network, the *ap* parameter may be used to request an association with the specified MAC address. However, the station is not required to remain associated with the specified access point and may choose to roam to a different access point if the signal strength drops too much:

```
[root@bloodhound]# iwconfig wvlan0 ap 01:02:03:04:05:06
```

Setting the data rate

Most cards support multiple bit rates. *iwconfig* allows the administrator to choose between them by using the *rate* parameter. Bit rates can be specified after the *rate* parameter, or the keyword *auto* can be used to specify that the card should fall back

* I use the term SSID in this book to refer to a network name. Some drivers, including the WaveLAN drivers, use ESSID instead. The distinction is that an ESSID is a network name assigned to an extended service set, not any old service set.

Exhibit 1019
Page 261 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

to lower bit rates on poor-quality channels. If *auto* is combined with a bit rate, the driver may use any rate lower than the specified rate:

```
[root@bloodhound]# iwconfig wvlan0 rate 11M auto
```

Configuring WEP

The key parameter controls the WEP function of the driver. Keys can be entered as hexadecimal strings as in the PRISM driver. Enter the string four digits at a time with dashes between the two-byte groups. (Although it is not mentioned in the documentation, I can enter keys in the colon-separated MAC address format as well.)

```
[root@bloodhound]# iwconfig wvlan0 key 0123-4567-89
[root@bloodhound]# iwconfig wvlan0 key 01:23:45:67:89
```

Multiple keys can be entered using a bracketed index number:

```
[root@bloodhound]# iwconfig wvlan0 key 0123-4567-89
[root@bloodhound]# iwconfig wvlan0 key 9876-5432-01 [2]
[root@bloodhound]# iwconfig wvlan0 key 5432-1678-90 [3]
```

Longer keys can be entered simply by using more bytes. If the key length is longer than 40 bits, the key is assumed to be a 104-bit key.

```
[root@bloodhound]# iwconfig wvlan0 key 0011-2233-4455-6677-8899-0011-22 [4]
```

Once multiple keys have been entered, select one by entering the index number without a key value:

```
[root@bloodhound]# iwconfig wvlan0 key [2]
```

Activate WEP processing using `key on` and disable WEP using `key off`. These can be combined with an index number to select a new WEP key:

```
[root@bloodhound]# iwconfig wvlan0 key [3] on
[root@bloodhound]# iwconfig wvlan0 key off
```

Finally, two types of WEP processing can be done. An *open* system accepts data frames sent in the clear, and a *restricted* system discards cleartext data frames. Both of these parameters can be combined with an index number:

```
[root@bloodhound]# iwconfig wvlan0 key [4] open
[root@bloodhound]# iwconfig wvlan0 key [3] restricted
```

The *key* parameter may also be accessed with the *encryption* parameter, which may be abbreviated as *enc*.

Tuning 802.11 parameters

iwconfig allows you to tune the RTS and fragmentation thresholds. The RTS threshold in the *wvlan_cs* driver is 2,347, which effectively disables RTS clearing. In an environment likely to have hidden nodes, it can be set using the *rts_threshold* parameter with *iwconfig*. *rts_threshold* can be abbreviated as *rts*.

```
[root@bloodhound]# iwconfig wvlan0 rts 500
```


Exhibit 1019
Page 262 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

The default value of the fragmentation threshold is 2,346. In noisy environments, it may be worth lowering the fragmentation threshold to reduce the amount of data, which must be retransmitted when frames are lost to corruption on the wireless medium. Set the parameter by using the *fragmentation_threshold* argument to *iwconfig*. It may be set anywhere from 256 to 2,356, but it may take on only even values. *fragmentation_threshold* may be abbreviated as *frag*.

```
[root@bloodhound]# iwconfig wvlan0 frag 500
```

802.11 stations maintain several retry counters. When frames are retransmitted "too many" times or wait for transmission for "too long," they are discarded. Two retry counters are maintained. The long retry counter, set by the *retry* parameter, is the number of times transmission is attempted for a frame longer than the RTS threshold. The short retry counter, set by the *retry min* parameter, is the number of times transmission will be attempted for a frame shorter than the RTS threshold. Unlike many drivers, *iwconfig* also allows for configuration of the maximum frame lifetime with the *retry lifetime* parameter. To specify a value in milliseconds or microseconds, append "m" or "u" to the value:

```
[root@bloodhound]# iwconfig wvlan0 retry 4
[root@bloodhound]# iwconfig wvlan0 retry min 7
[root@bloodhound]# iwconfig wvlan0 retry lifetime 400m
```

wvlan_cs driver parameters

Several options can be passed to the *wvlan_cs* module by *cardmgr* when it is loaded into the kernel. These options are most easily set in */etc/pcmcia/config.opts*. See Table 13-5.

Table 13-5. wvlan_cs driver parameters

Parameter	Value type	Description
irq_list	Comma-separated integer list	Specifies interrupts that may be used by driver.
port_type	integer, range 1–3	Sets network type to infrastructure (1), wireless distribution system (2), or ad hoc network (3).
station_name	string	Sets station name; defaults to card setting.
network_name	string	Configures name for ad hoc network or name of target infrastructure network.
channel	integer, range 0–14	Channel number/operating frequency for ad-hoc networks; not used for infrastructure networks. Default is 3.
ap_density	integer, range 1–3	Sets threshold for roaming based on a low-density (1), medium-density (2), or high-density (3) installation. The default is a low-density installation, which minimizes roaming activity.
medium_reservation	integer, range 0–2,347	Sets RTS/CTS threshold. Default is 2,347.
frag_threshold	even integer, range 256–2,346	Sets fragmentation threshold. Default is 2,346.
transmit_rate	integer, range 1–7	Each integer has a meaning; default is *auto select high speed* (3), which allows for fallback to lower speed.

Exhibit 1019
Page 263 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Table 13-5. wvlan_cs driver parameters (continued)

Parameter	Value type	Description
eth	integer, range 0–1	If set to 1, all devices are *ethN*. If set to 0, all devices are *wvlanN*. By default, this is set to 1, so all devices are *ethN*.
mtu	integer, range 256–2,296	Maximum transfer unit; default is 1,500.

Troubleshooting

Naturally, all the PCMCIA troubleshooting notes from the previous section apply.

Kernel 2.4: orinoco_cs

In addition to the WaveLAN cards and any OEM versions of WaveLAN cards, *orinoco_cs* contains basic support for some PRISM-2–based cards and Symbol cards that use the same MAC chipset. *orinoco_cs* has been part of the Linux kernel distribution since kernel Version 2.4.3.

Compiling and installing

Some wireless LAN support code is activated in the kernel by enabling the Wireless LAN extensions (*CONFIG_NET_RADIO*) kernel configuration option. *CONFIG_NET_RADIO*-enabled kernels collect wireless statistics and expose additional data structures used by the *orinoco_cs* driver. To compile the *orinoco_cs* driver itself, recompile the kernel with Hermes support (controlled by the *CONFIG_PCMCIA_HERMES* variable).

The standard wireless tools are also required. They can be downloaded from *http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html*. As with the previous driver, installation is a straightforward matter of running *make* in the right spot.

PCMCIA configuration

Most distributions ship with the *wvlan_cs* driver still enabled. To change the driver used by the distribution, it is sufficient to change the module binding the PCMCIA configuration. The author of *orinoco_cs* supplies a file, *hermes.conf*, which contains card definitions for the cards supported by *orinoco_cs*. Because *hermes.conf* ends in *.conf*, it is sourced by the line at the end of */etc/pcmcia/config* that reads all *.conf* files. However, to avoid binding conflicts, you must comment out all the lines that bind the older *wvlan_cs* driver to newly inserted cards. Alternatively, it is sufficient to edit the definition of your wireless card to bind the *orinoco_cs* driver after grabbing identification information from the output of *dump_cis*:

```
# in hermes.conf
#
card "Lucent Technologies Wavelan/IEEE"
   version "Lucent Technologies", "WaveLAN/IEEE"
   bind "orinoco_cs"
```


Exhibit 1019
Page 264 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

```
# from standard /etc/pcmcia/config
#
# card "Lucent Technologies WaveLAN/IEEE"
#    version "Lucent Technologies", "WaveLAN/IEEE"
#    bind "wvlan_cs"
```

Configuring the orinoco_cs interface

Configuration of *orinoco_cs* is identical to the configuration of *wvlan_cs*. When the card is inserted, the */etc/pcmcia/wireless* script is run, using the configuration options in */etc/pcmcia/wireless.opts*. The wireless script is a frontend to the *iwconfig* program. Editing fields in *wireless.opts* sets the arguments to *iwconfig*. For details on configuring the options to *iwconfig*, see the previous section on the *wvlan_cs* driver.

Exhibit 1019
Page 265 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

CHAPTER 14

Using 802.11 Access Points

In even the simplest 802.11 network, proper configuration of the access points is essential. Without properly configured network interfaces, no traffic will be bridged on to the wired network.

Access points can be divided into two groups. Home gateways and small office products are targeted at price-conscious users and do not offer much in the way of functionality beyond simple connectivity. Business-grade products are more expensive, but they possess functionality that is key to working with larger networks containing multiple access points. This chapter takes a look at two access points targeted at the high-end commercial user: the ORiNOCO AP-1000 and the Nokia A032. Extrapolating low-end product administration from the basic set of tasks is straightforward.

The major tasks facing a network administrator are to connect the access point to the wired network, configure the network interfaces, enable security features, and perform any configuration adjustments necessary to tune the wireless network to the area in which it is deployed. This chapter assumes that power and wired network connections are straightforward and have been taken care of.

General Functions of an Access Point

All access points provide a similar set of features to network users because the standard feature set is specified by 802.11. Configuration of these features, of course, is vendor-specific, but the products are fairly similar to each other.

Access points are bridges between the wireless world and the wired world. As bridges, then, all access points have features that one would expect to see on a network bridge. They have at least two network interfaces: a wireless interface that understands the details of 802.11 and a second interface to connect to wired networks. To take advantage of the installed base and expertise, the wired interface is almost always an Ethernet port. Many products also have a WAN port. Sometimes the WAN port is a serial port that can be connected to a modem for use with a dial-up ISP account. It

Exhibit 1019
Page 266 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

is also common to find DSL interfaces in access points. Still, other access points add a second Ethernet port for connection to a DSL modem or cable modem. Some access points have multiple wireless interfaces so network managers can increase hot-spot capacity by using two interfaces on nonoverlapping channels. Many access points also offer the option of using external antennas to further boost range and allow for fine-tuning of antenna placement. Bridges have some buffer memory to hold frames as they are transferred between the two interfaces, and they store MAC address associations for each port in a set of internal tables.

Bridging tables are, of course, highly implementation-specific, and there is no guarantee of similarities across the industry. Generally speaking, though, inexpensive consumer devices are fundamentally designed with the assumption that they will be the sole access point for a network, while the more expensive commercial devices include more advanced features to make large deployments and rollouts easier. Commercial-grade devices are also designed to work cooperatively; the most common feature is a vendor-proprietary Inter-Access Point Protocol (IAPP). An IAPP allows wireless stations to move from access point to access point without interrupting link-layer connectivity. At this point, no standard for an IAPP exists, though the 802.11 working group is addressing this shortcoming. (Some cheaper products may support an IAPP, however.) Network management is generally much more sophisticated on commercial-grade products to enable network engineers to manage the tens or hundreds of devices used to create a large-scale coverage area.

All but the cheapest access points have a TCP/IP network interface. TCP/IP interfaces are intended for remote management and typically accept only basic configurations; anything beyond an IP address, netmask, and a single static default route is atypical. Depending on the level of sophistication of the hardware and software, varying levels of low-level interface configuration are possible. Naturally, all products allow the configuration of the network name. Other low-level parameters, however, may or may not be configurable. For access points that use PCMCIA network interfaces, the configuration of the wireless interface may depend on the firmware present on the PCMCIA interface card.

Depending on the market for which an access point is developed, it may offer services to its wireless clients. The most popular service is DHCP; wireless stations may be assigned addresses automatically upon association. Many access points can also perform network address translation (NAT), especially the "home gateway"–type products that can connect to a modem and dial up an ISP.

Security has been a sore point for wireless network managers since before the advent of 802.11's success. Access points have a privileged position with respect to security concerns because they are the gateways to the wired network and are ideally positioned to implement security policies. As detailed elsewhere in this book, the tools that access points provide to enforce security polices are sorely lacking. Naturally, WEP implementations are fully configurable with new keys and can be set to either

Exhibit 1019
Page 267 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

open-system or shared-key authentication, but few other standardized access control tools exist. (Naturally, several vendors have implemented proprietary approaches.) The major access control method implemented by access points is *MAC address filtering*. With address filtering, network administrators can give each access point a list of the MAC addresses of clients that should be allowed to access the network. Many products that offer remote network management provide some tools to filter management access, but few of the filters are based on anything other than easily forged source IP addresses.

Management interfaces often leave something to be desired. Configuration of access points tends to be challenging because access points must be manufactured cheaply, and low-cost devices tend not to have the processing power to run an easy-to-use configuration engine. Most vendors use lightweight operating systems running on low-powered hardware, but one of the trade-offs of using a lightweight operating system is that it does not provide the programming environment necessary to build rich functionality. Early access points offered both a cryptic command-line interface and a web-based management interface. This is still a common model. It is not unheard of for a vendor to supply proprietary management software for just one operating-system platform. (Most offenders in this category are management applications confined to run on Microsoft Windows.) Typically, host software connects over a serial port, either RS-232 or USB, and runs a proprietary management protocol to change configuration variables. Moreover, it is only recently that vendors have even begun to address seriously the concerns of large-scale access point management.

Debugging and troubleshooting tools are as advanced as management tools, which unfortunately means that they often leave network administrators mired in inconclusive or irrelevant information. Ideally, products should maintain detailed logs of activities, but it is common to find vague logs of results that give very little insight into failures. Counters can be helpful, but only if the right counters are accurately maintained. Tools such as *ping* and *traceroute* are common, but network analyzers and packet capture tools are not.

Types of Access Points

Broadly speaking, there are two classes of access points in the marketplace. A low-cost tier is sold widely through retail channels directly to the end user. These low-cost devices are specialized computing platforms with only limited memory and storage. The higher-cost tier incorporates additional features required to support large deployments; frequently, these devices have additional memory and storage and resemble small general-purpose computing platforms in their design.

For the home: residential gateways

The low-cost tier is composed of devices often called *residential gateways*. Residential gateways are designed to be as low-cost as possible, so only the basic features

Exhibit 1019
Page 268 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

required for the typical small or home office are included. Residential gateways generally share the following characteristics:

- Most devices include a DHCP server to make plug-and-play configuration easier.
- They are often deployed by users with one routable IP address, so NAT implementations are common.
- Depending on the type of customer the residential gateway is aimed at, the WAN interface is a modem, a serial port, or even DSL. (Some residential gateway products may use an Ethernet port as the "WAN" connection to a cable modem or DSL modem.)
- They are often built as a single integrated unit, complete with a built-in antenna. If suitable coverage cannot be found, it is necessary to relocate the entire unit.
- Many products now claim to have an *IPSec pass-through* feature to allow the use of IPSec through NAT, which works with varying degrees of success depending on the IPSec VPN solution chosen.
- Configuration of residential gateways often relies on a Windows program that is installed and uses a proprietary protocol over serial or USB to configure the device.
- They are often sold directly to the end user and are designed to be aesthetically pleasing. Unfortunately for many end users, the improved visual design prevents the stacking of residential gateways with other network equipment.

As this book was written, residential gateways typically cost $150 to $300. Examples of the residential gateway class of device are the 3Com Home Wireless Gateway, the Apple AirPort, the D-Link DWL-1000AP access point, the Intel Wireless Gateway, the Linksys WAP11, and the the Orinoco RG-1100.

For the office: enterprise (corporate) gateways

Enterprise gateways, which often go by many other names that imply the buyer values features over cost, provide everything residential gateways do, plus additional features useful for larger-scale environments. Enterprise gateways generally share the following characteristics:

- The area over which mobility is required is much larger and requires several access points working in concert. Enterprise products support an IAPP so that a group of access points can be used to provide mobility through large areas. All IAPPs are proprietary at the time this book was written, but efforts to standardize the IAPP and enable roaming with devices from different vendors are underway.
- Upgrades are much easier with enterprise products. The wireless interfaces themselves are often PCMCIA cards, which makes the upgrade path much easier. As an example, many enterprise gateways do not require "forklift" upgrades to move to 802.11a. Instead, the wireless interface can be replaced

Exhibit 1019
Page 269 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

with an 802.11a-compatible interface, and a software upgrade provides a driver for the new card and the software features necessary for 802.11a.

- Enterprise-class products are designed to make deployment as easy as possible. Many high-end products can draw power over the unused pins in the Ethernet cable by complying with draft versions of the IEEE 802.3af standard.
- Frequently, site survey tools come bundled with enterprise-class products so network managers can plan large deployments by directly assessing coverage quality.
- Wireless interfaces on enterprise gateways usually allow for the possibility of using external antennas, and a wide selection of antennas is available to provide coverage in specific types of areas. External antennas may come standard with some enterprise products. Transmission power can be adjusted on many devices to enlarge or shrink the coverage area.
- Security developments appear on high-end products first. Enterprise gateways were the first to implement address filtering, and they are the test-bed for new security features implemented in software. Some vendors have enhanced the association process with proprietary key exchanges. 802.1x is beginning to appear in these products as well.
- Reflecting the administrative demands, configuration of enterprise-class devices is done with easily scripted command-line interfaces or SNMP, and monitoring and management capabilities are far more extensive than in residential gateways.

Naturally, these additional capabilities do not come without a price. As this book was written, enterprise gateways typically cost $800 to $1,100. Prices for enterprise-class products are not subject to the same downward pressure as residential gateways. Generally, enterprise-class products are made with much more generic hardware than residential gateways. Software upgrades are more common and can continue to add value through the life of the product. Examples of the enterprise gateway class of device are the 3Com AirConnect, the Cisco Aironet 350 Series Access Point, the Intel PRO/Wireless LAN Access Point, the Nokia A032, the Orinoco AP-1000 and AP-2000, and the Proxim RangeLAN2. This chapter describes the management of the Nokia A032 and Orinoco AP-1000 as examples. As you'll see, the mechanisms for configuring and managing these access points are different, but the information you need to supply and the configuration parameters that you can control are fundamentally similar. Even if you don't use Orinoco (a.k.a. Lucent, a.k.a. WaveLAN, a.k.a. Agere) or Nokia products, this chapter will show you what you need to do to configure any commercial access point.

Selecting Access Points

When choosing an access point, you should take a number of factors into account. With the emergence of 802.11 as the main vendor-neutral standard, standards compliance is generally not a big factor. Most 802.11 equipment has gone through

Exhibit 1019
Page 270 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

compatibility testing with the Wireless Ethernet Compatibility Alliance (WECA); equipment that has received WECA's "Wi-Fi" (short for "wireless fidelity") certification has proven successful interoperability and standards compliance. Originally, the Wi-Fi program was only for 802.11b equipment, though a successor certification program called Wi-Fi5 is in development for 802.11a gear. Wi-Fi certification is almost certainly something to look for because it is practically a guarantee of interoperability.

802.11 is a complex standard, however, with several optional features. External antennas are often useful for creating a dense coverage blanket over an area. Not all access points can connect to external antennas; it may be an extra-cost option. Even if an access point has an external antenna connector, there is no guarantee that you'll be able to find a wide range of antennas available for use. 802.11 only requires that any connectors for external antennas have a standard impedance of 50 ohms. The actual physical connector may be proprietary to the vendor. If external antennas are important for your deployment plans, make sure that a wide range of antennas is available, whether through the 802.11 vendor or another source.

Security has been an area of notable innovation. The standard specifies one method, WEP, with a short key length. Some vendors offer longer key length versions of WEP, which are nonstandard but generally interoperable. Longer key length WEP implementations may add cost, however. More importantly, though, WEP has been conclusively demonstrated to be severely flawed. Some attempts at fixing WEP have been made by vendors, but they are fundamentally constrained by the design. As a result, there are a number of proprietary prestandard security mechanisms based on 802.1x (which also appears to be flawed). Some products have even begun to support 802.1x, with a few enhancements for wireless use. Before you commit to an evolutionary dead end, ensure that any solution can be easily upgraded to incorporate the standard security framework being developed by the 802.11i task group.

If roaming is important, a single-vendor solution is mandatory. Most vendors ship products with a protocol that enables roaming between access points, but products from different vendors are not guaranteed to interoperate. In the absence of a standard, there cannot be compliance testing. While this is being addressed within the 802.11 working group, until a standard is finalized, only a single-vendor solution can provide roaming between access points.* Until a standardized protocol hits the market, you may want to investigate the situations in which the vendor claims to enable roaming. Can stations move between access points even when WEP is enabled? How quickly can stations move between access points? It may also be worthwhile to

* While this is generally true, there are some exceptions due to OEM relationships. The Apple AirPort has been reported as allowing roaming connections to and from Lucent access points, but the AirPort is essentially a rebadged Lucent access point with a far superior aesthetic appearance.

Exhibit 1019
Page 271 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

obtain a guarantee from the vendor that any access points you purchase now can be easily upgraded to the forthcoming standard.

802.11 includes a number of power-saving functions in the standard. Most are optional. If your deployment is based heavily on battery-powered devices, it may be worth evaluating which power-saving features are included with particular devices. It may also be worth experimenting with devices to see just how much longer batteries last with the power-saving functions enabled.

In some deployments, getting power to the access points can be a major headache. To blanket an area with the coverage required for a large implementation, access points often need to be placed in an area where power is not easily accessible. Long antenna runs can degrade signal quality unacceptably, so it is much better to bring power to the location. Installing new electrical conduits is often quite expensive. Work must be performed by licensed electricians, and building codes may impose additional restrictions. Some products can supply power over the unused pins in the Ethernet cable. Network wire is not subject to the same restrictions as electrical cable and can be installed by network administrators.

Device management is an important consideration. Wireless networks are a new service, and network staff will need to plan, evaluate, purchase, deploy, and maintain the additional hardware. Large deployments may have tens or hundreds of access points, which can easily make network management a headache without good tools. Does the vendor offer an access point manager to configure large numbers of devices in parallel? Can management of the access points be incorporated into your existing network management infrastructure using tools that you already have deployed? Are the management tools secure enough? Many products can be managed only with clear-text protocols, which may be just an annoyance or a major violation of a security policy. Experience with other network devices has shown that software upgrades are a frequent occurrence. How is the software upgraded, and how much functionality can upgrades add? Can new protocol features be added with firmware updates?

Depending on the size of the deployment, it may be possible to evaluate equipment before buying. See if you can get a feel for the range of each access point and test with a variety of common cards. Capacity on an 802.11 network is ultimately limited by the radio link, but you will want to make sure that there are no other capacity restrictions. Does the access point provide the processing power to run the wireless side at maximum capacity with WEP enabled? Some products incorporate cryptographic processors to help with the load of WEP, but many do not. Products that depend on a central processor to run WEP may run out of capacity if they are upgraded to the faster 802.11a standard. Products upgraded to 802.11a will also suffer if they do not have Fast Ethernet ports. Try to set up a test network and get a feel for the configuration required to integrate the access points with the rest of your network gear.

Exhibit 1019
Page 272 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

As with many other purchasing decisions, of course, there are a number of "soft" factors that may not be easily quantifiable. Warranties, a relationship with the vendor, and the quality of the technical support may all influence the purchasing decision. Soft factors are not technical nor easily quantifiable, however, so I will not discuss them.

Are Access Points Really Necessary?

Access points are not required for a wireless network. Wireless stations can be used in independent networks, which do not require an access point. Building a Unix box that routes between an Ethernet network and a wireless network is not difficult, and hardware can often be reused from the scrap pile. Why, then, would anybody use an access point?

Now that residential gateways have fallen well below the $200 mark, building a Unix router is no longer a cost-effective option for single–access point networks. Once you consider what staff time is worth, building a Unix router is a pretty silly use of staff time. Access point hardware has some advantages over redeployed general-purpose platforms, too. Access points are small devices with no moving parts. As a result, they do not consume a great deal of electrical power and do not generate much heat. There is one notable exception to this rule, though. Apple offers a "software base station" that transforms any desktop machine into a bridging access point. With a few mouse clicks and very little effort, a desktop computer can become a base station.

Unix-based routers have never been effective in larger deployments because of the lack of mobility support. Effective roaming requires transparent *bridged* access, not routed access, to the link layer at different physical locations. However, roaming with 802.11 is possible only when access points can communicate with each other to track the movement of a wireless station. In the future, it is likely that an open source Unix distribution will have the features necessary for an access point: low-level access to fundamental 802.11 parameters on the card, Ethernet bridging, and an IAPP. Until then, though, there is no substitute for commercial products.

ORiNOCO (Lucent) AP-1000 Access Point

The AP-1000 is the mid-range Orinoco product. The low end of the product line consists of products for the home market; the high-end products (for example, the AP-2000) has features such as enhanced security and upgrade ability to 802.11a.

In addition to purchasing the AP-1000 base unit, you must purchase wireless interfaces separately. Unlike many other products, the AP-1000 has two slots for PCMCIA wireless interface cards. (A less expensive version of the AP-1000, the AP-500, has only a single slot.)

Exhibit 1019
Page 273 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Unix-Based Access Points

One of the most basic preconditions for making a Unix-based access point is enabling access point functions in the wireless interface card. One of the major hurdles is rewriting the 802.11 headers. All traffic in an infrastructure network flows through the access point. Access points must rewrite the transmitter and receiver addresses in the 802.11 headers. Other management functions may be required as well. For example, 802.11 includes a number of power-saving mechanisms for infrastructure networks in the specification, but they can be used only on networks with access points that implement them.

There is also a nontechnical hurdle. Many vendors have actively supported the development of open source Unix drivers for their cards. After all, vendors make money selling hardware, and it is a good thing for them to sell cards for all client systems, even those that run open source Unix. Access points are a different story, however. Vendors have not been as forthcoming with the interface used to put cards into the access point mode. Access points are quite lucrative, and providing a driver interface in the access point mode in the card could potentially cannibalize access point sales.

At one point, the only way to get an Intersil-based card to act as an access point interface was to purchase the reference design from Intersil. (The reference design shipped with firmware that had access point functionality, and that firmware was not sold separately.) Intersil's shipping station firmware does, however, include something called a "Host AP Mode." In the Host AP Mode, the PRISM-2 chipset automatically takes care of "menial" tasks, such as transmitting Beacon frames and acknowledging incoming transmissions. Jouni Malinen has developed a driver to use the Host AP Mode with Linux kernel 2.4. In conjunction with the Ethernet bridging implementation in the kernel, this driver can be used to build an access point. It is available from *http://www.epitest.fi/Prism2/*.

With the present state of driver software, it is possible to build a Unix-based router. (I mean "router" pedantically, as "layer 3 network device.") One interface would connect to a wired network as it always has, and a second wireless interface could be run in IBSS mode. Ross Finlayson has established a community network at a coffee house in Mountain View, California using a FreeBSD-based router. The project's home page is at *http://www.live.com/danastreet/*, and there is a page devoted specifically to the router itself at *http://www.live.com/wireless/unix-base-station.html*.

The Lucent access points in this class do not include an internal DHCP server, but they are compatible with external DHCP servers. This is a reasonable assumption—if you are adding wireless capability to an existing network, it's almost certain that you already have a DHCP server on your network. Lucent's products for home use incorporate a DHCP server.

Exhibit 1019
Page 274 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

The Management Interface

The AP-1000 is managed almost exclusively with SNMP through a Windows-based SNMP tool designed specifically for the AP-1000. Command-line support was added in the Fall 2001 software release (Version 3.83). Only basic configuration can be done over the serial port. Three main items may be configured with the serial port: the IP network interface (address, mask, and router, or the use of DHCP), the wireless interface (network name, channel, and WEP settings), and the SNMP "password" (community string). The use of SNMP as a management protocol is a strange choice. A great deal of the configuration information on the AP-1000 should remain secret, but SNMP does not provide any facilities for protecting new settings as they travel across the wire. Even worse, the community string is easily recovered with a packet sniffer. Once the community string is recovered, a malicious attacker could change the settings on your access point.

The default configuration is to boot with the DHCP client enabled, so any DHCP server on the network can assign it an address. If you like, you can set up your DHCP server to assign a fixed address to the access point by associating the desired address with the Ethernet address printed on the AP-1000.

Introduction to the Orinoco AP Manager

The AP Manager software runs only on Windows (95/98/ME/NT/2000/XP). It is essentially a frontend to SNMP software, so you can build your own frontend on any SNMP-capable workstation.

The main AP Manager window is shown in Figure 14-1. When the software starts, it sends UDP probes to port 192 on the local network to locate existing access points. After an AP boots, it answers the requests, and the AP Manager displays a list organized by IP address:

```
[gast@bloodhound]$ tcpdump ip host 192.168.200.222
tcpdump: listening on eth0
19:33:18.190921 192.168.200.222.2159 > 255.255.255.255.192: udp 116
19:33:18.191196 192.168.200.100.192 > 192.168.200.222.2159: udp 116 (DF)
```

After locating access points, the AP Manager software monitors each AP it finds for reachability using ICMP echo requests.

Applying configuration changes

Many configuration changes require a restart. First, the changes are saved to the access point using SNMP Version 1 *SET* requests. Then the access point is restarted so it can reboot with the new configuration. Any change to the access point typically requires a reboot.

The access point's operating system is based on software from KarlNet, so it is not a surprise that the SNMP protocol operations direct the access point to set a large

Exhibit 1019
Page 275 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

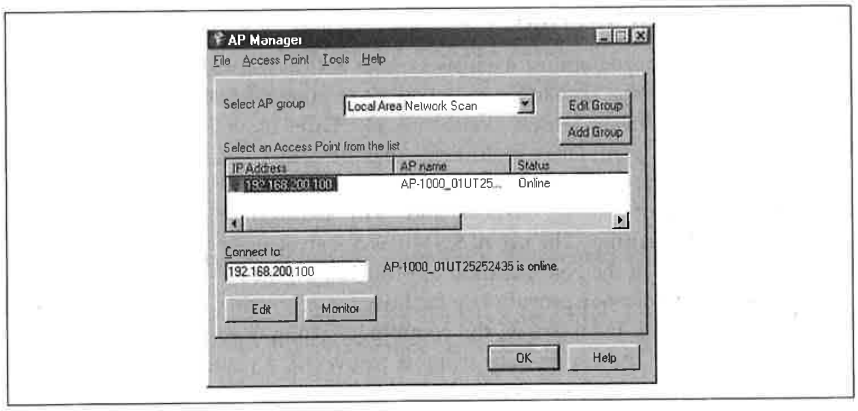


Figure 14-1. AP list

number of variables using KarlNet's SNMP enterprise number (762). Decoding the SNMP traffic and understanding it are two different matters, however. Just because it is easy to find where objects in the SNMP MIB are being set does not make it easy to find out what they do.

Basic Management Tasks

Running the access point is simple. Straight out of the box, the AP-1000 uses DHCP to obtain the address for the management interface.

Viewing the network configuration

Edit the access point from the main window and select the Access Point IP tab. Specify an IP address, network mask, and default gateway. Somewhat unusually, the AP-1000 also allows you to set the default TTL. See Figure 14-2.

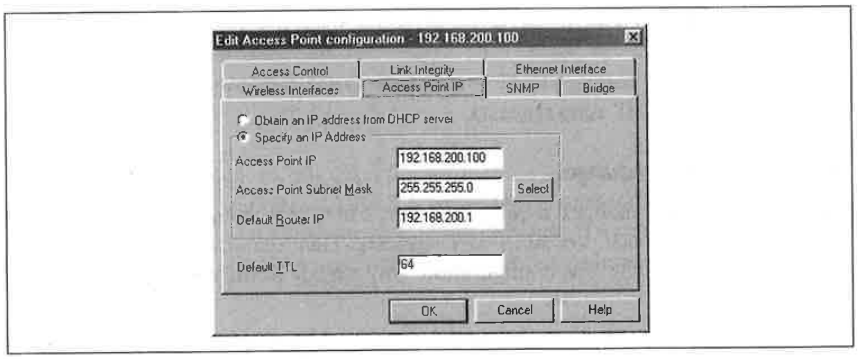


Figure 14-2. Access point IP interface

Exhibit 1019
Page 276 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

One nice thing about the AP Manager is that assigning the subnet mask is especially easy. The Select button raises a subnet mask panel that shows the mask in dotted decimal, hexadecimal, and mask-length notations. See Figure 14-3.

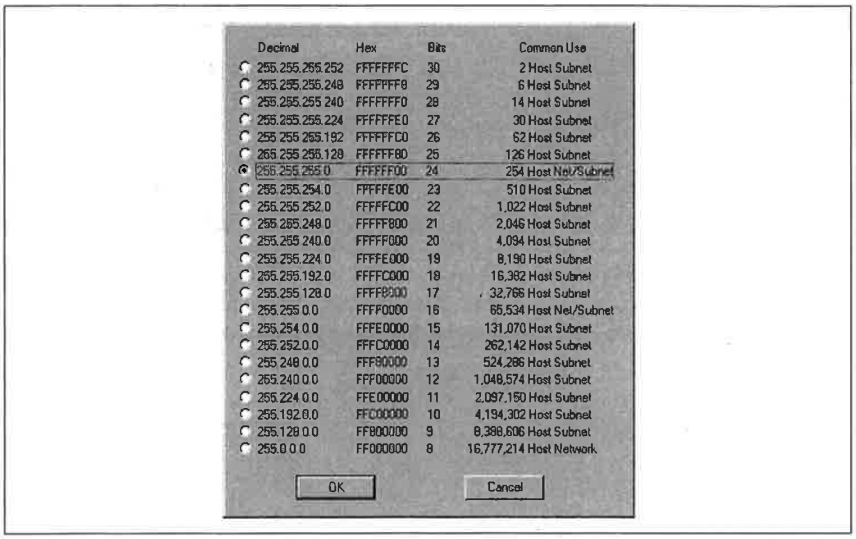


Figure 14-3. Selecting a subnet mask

Configuring the radio network

Configuration of the radio network is done on the Wireless Interfaces tab, shown in Figure 14-4. Each PCMCIA slot may have a wireless interface. Two configuration buttons are present for each interface. The Security button configures WEP, and the Advanced button configures the channel and radio parameters. The only parameter set on the main page is the network name (referred to in the specification as the ESSID).

The Advanced wireless interface configuration dialog box is shown in Figure 14-5. The channel, which is displayed both as a channel number and as a channel frequency, can be changed in the top dialog box. When you select the operating channel, you must be aware of any regulatory restrictions in your location.

Settings for the DTIM interval and RTS/CTS thresholds should be left as their default values unless they need to be changed to address a performance problem. Chapter 17 addresses changing 802.11 parameters in response to problems.

The multicast data rate is set with its own button in the advanced setup. Multicast messages are transmitted to all stations simultaneously, so it is vital that any multicast transmissions are made at a rate intelligible to all associated wireless stations.

Exhibit 1019
Page 277 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

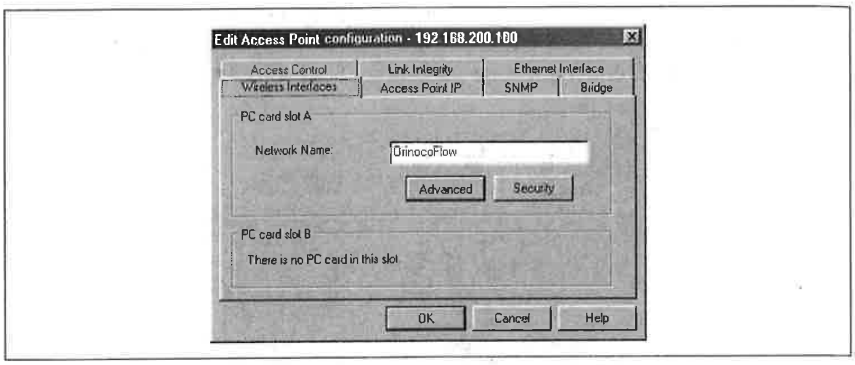


Figure 14-4. Wireless Interfaces tab



Figure 14-5. Advanced wireless interface configuration

The default value for the multicast rate is 2 Mbps because all commercial 802.11 direct-sequence cards are capable of 2-Mbps operation. You may change this to a higher rate, but you must ensure that network coverage is good and all stations are equipped with 802.11b cards compatible with the higher data rates.

The vendor cautions that the Interference Robustness setting should be used rarely with specific directions from an expert. There is a further warning that if the setting is needlessly enabled, connectivity and roaming may suffer. The documentation suggests that the Interference Robustness setting is used to work around interference close to the base station or in the path from the client to the base station. Based on their description, in many cases, it would be easier to redesign the wireless network than to suffer through what appears to be a nonstandard option.

Exhibit 1019
Page 278 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

The Distance Between APs option claims to tune the access point configurations to match the deployment. "Large" networks have very little overlap between BSSs and stations close to the edge of the BSS coverage area. "Medium" networks have 20% overlap, and "small" networks have an overlap area of 50% or more. It is not clear, however, which parameters are changed in response to setting this option.

Configuring the wired network

Configuring the wired network is done with the Ethernet Interface tab, shown in Figure 14-6. The interface is an RJ-45 jack, and the Ethernet transceiver is capable of full-duplex operation if it is connected to an Ethernet switch. By default, the port is set to automagically negotiate speed and duplex. Not all autonegotiation implementations are created equal, though, so you may need to pick a setting for compatible operation. One way to tell if the Ethernet interface is not negotiating properly is to monitor the number of collisions, especially late collisions, on the switch port. Late collisions are abnormal at any level and are a common indicator of mismatched duplex settings.

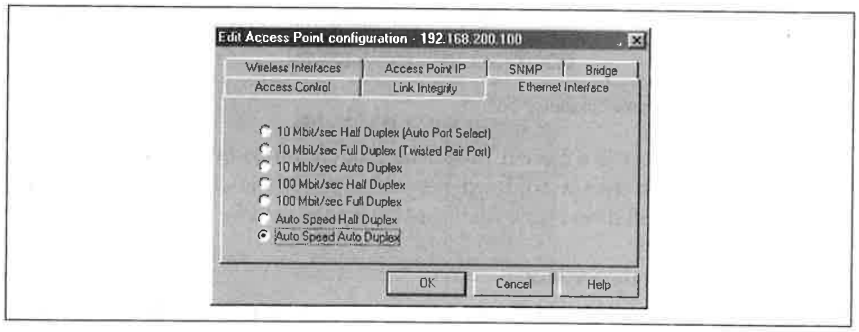


Figure 14-6. Ethernet interface configuration

Security Configuration

As with most 802.11 access points, security configuration is limited to configuring WEP, MAC address filtering, and limiting management access based on source IP address.

Configuring WEP

To enter WEP keys, click the Security button on the Wireless Interfaces tab of the main access point configuration screen. The security configuration is essentially WEP configuration; clicking the Security button brings up the WEP configuration dialog box shown in Figure 14-7. The AP-1000 supports both "64-bit WEP" (40

Exhibit 1019
Page 279 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

secret bits) and "128-bit WEP" (104 secret bits), depending on the wireless interface card used. "Silver" Orinoco cards support only the shorter length, and "Gold" cards support both. Keys can be entered either as alphanumeric strings or hexadecimal strings. Figure 14-7 shows a key being entered with a leading *0x* to denote that the key is being entered as a hexadecimal number.

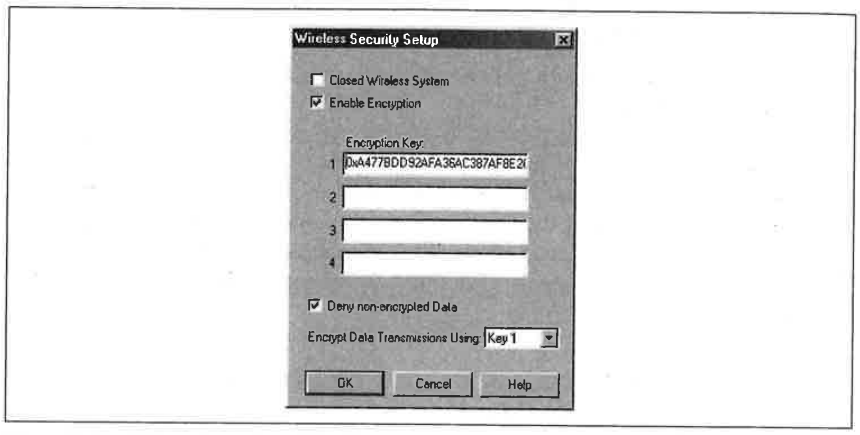


Figure 14-7. WEP configuration dialog box

Closed Wireless System is a Lucent proprietary extension to 802.11 that drops any frames from stations with a wildcard network name. It is compatible only with Orinoco cards and will drop traffic from cards manufactured by other vendors.

Filtering management connections

Management is done over SNMP and is controlled through the SNMP tab on the main access point configuration screen. The SNMP configuration box is shown in Figure 14-8. Read Password and Read/Write Password are community strings; both are set to public as the default. Traps are sent to the specified trap host; a setting of 0.0.0.0 disables traps.

Management access can be restricted using the SNMP IP Access List, which may contain up to five entries. (It's also based on source IP address filtering, which is a common but dubious security mechanism.) To add an entry to the list, click on the Add button to the right of the access list display. Doing so brings up the entry creation and edit screen, shown in Figure 14-9. Each entry is composed of an IP address and mask, and any source IP address matching one of the entries is permitted to access SNMP data. You can use the final field to restrict the interface on which SNMP processing takes place; 1 is used for the Ethernet interface, 2 and 3 for the two wireless interfaces, and X for any interface.

Exhibit 1019
Page 280 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

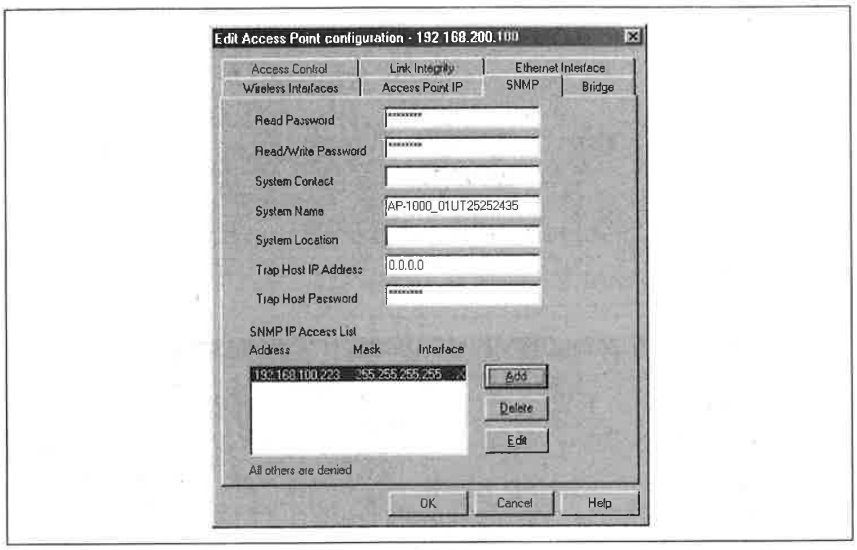


Figure 14-8. SNMP tab

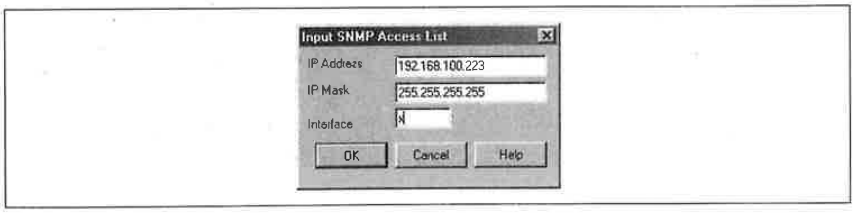


Figure 14-9. SNMP access filter dialog box

Restricting associations

Address filtering is configured under the Access Control tab, shown in Figure 14-10. AP Authentication is a local table stored inside the access point of MAC addresses that are allowed to connect. MAC addresses can also be stored in a RADIUS server. Adding addresses to the list of AP-authenticated addresses requires a system restart.

Monitoring Wireless Stations

Monitoring can also be performed using the AP Manager. Instead of editing the access point, monitor it. The main monitoring window is shown in Figure 14-11. By clicking on the various tabs across the top, you can see the bridge table, ARP table, various network statistics, and the list of associated stations.

Exhibit 1019
Page 281 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

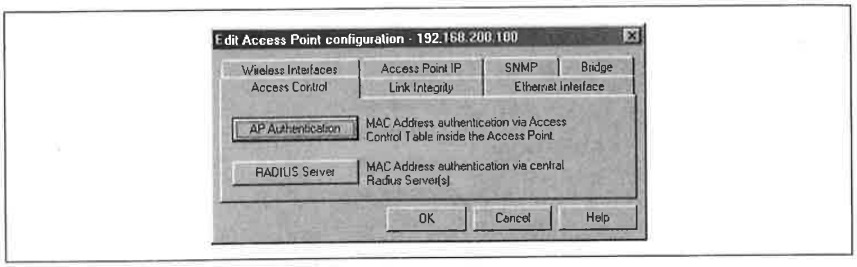


Figure 14-10. Access control tab

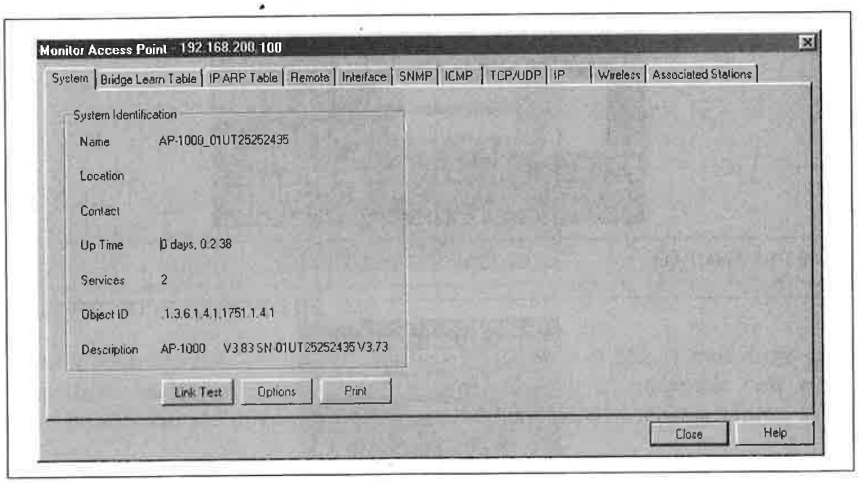


Figure 14-11. Access point monitoring

Monitoring associated stations

A common management task is to show the list of associated stations. Obtaining the list in the AP Manager is straightforward. Simply click on the Associated Stations tab at the top of the monitoring window, and the list of stations is displayed, as in Figure 14-12.

Naturally, associated stations must be given entries in the system bridge table. To view the bridging table, click on the Bridge Learn Table tab at the top. Figure 14-13 shows a simple example. Interface numbers have the same meaning as they do for SNMP access list purposes. Only one of the stations shown in Figure 14-13 is a wireless station. Beneath the display of this window is an implementation of the bridge group MIB.

Exhibit 1019
Page 282 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

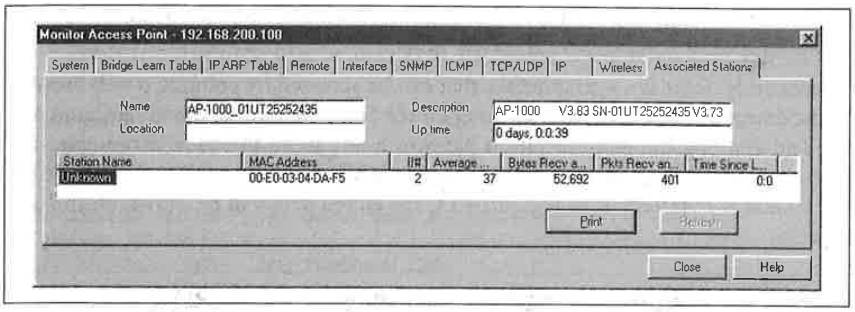


Figure 14-12. Associated stations

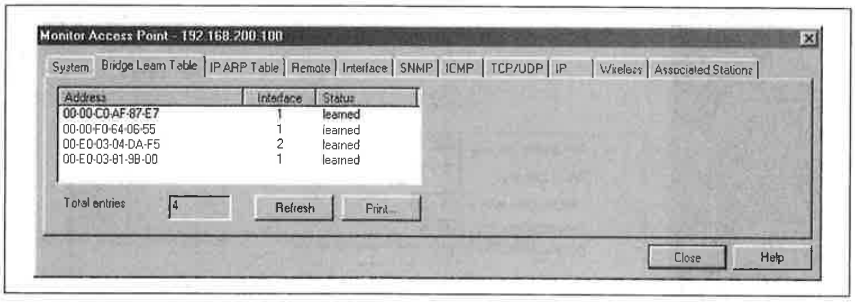


Figure 14-13. Bridge table

Nokia A032 Access Point

The A032 access point is the only access point in Nokia's current product line. It replaces the A020 and A021 access points. Unlike some products on the market, it comes with an interface card in the box and even ships with an external omnidirectional antenna. I found the external antenna to be problematic. Shortly after I started using it, the interface card stopped working, and I was unable to get a replacement.

A Tale of Two Management Interfaces

Like many network devices, the A032 has a console port that can be accessed using a null-modem cable. Configure the terminal emulator to the standard settings (9,600 bps, no parity, 8 data bits, and 1 stop bit) to access the console port. If a network address has already been configured, you can telnet to the device to gain access to the command-line interface.

The command-line interface lacks a great deal of the sophistication of the command line on more powerful devices. It has only limited help and does not assist with command

Exhibit 1019
Page 283 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

completion. Configuration is based on assigning values to parameters with the *set* command; its general format is set *parameter value*.

Alternatively, there is a web interface that can be accessed by pointing a web browser at the device's IP address. After clicking on the Setup link in the upper left-hand corner and entering the administration password, you'll see the basic setup screen in Figure 14-14. Changes to the device configuration frequently require a restart. Before the system is restarted, as in Figure 14-14, the edited data will be shown on the web screen.

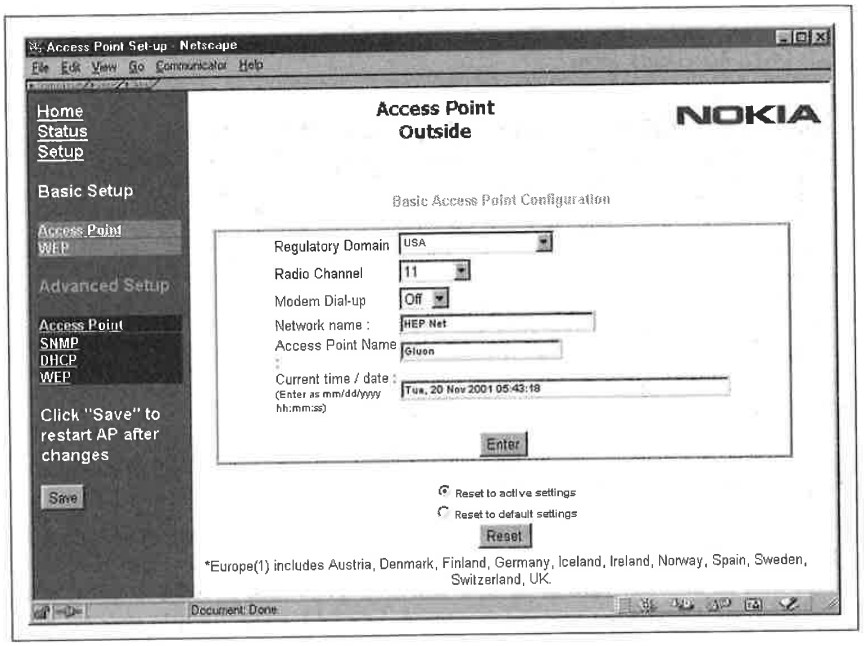


Figure 14-14. Web-based management interface

Web management is not for those concerned about security. Due to the limited computational power and storage space of the A032, it does not implement a secure web server, so the password is transmitted in the clear over the link layer. Of course, the link layer may be protected using WEP, but that is hardly reassuring.

Basic Management Tasks

To enable remote management and run the access point in the most basic way, you should configure the IP network interface for remote access and configure the wireless network interface with the necessary parameters. Bridging is implicitly enabled and needs no explicit configuration.

Exhibit 1019
Page 284 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Viewing the network configuration

The current configuration can be displayed by typing the *config* command at the command line:

```
CMD:config
=========Current Configuration =========
net_name: Photon        AP_name: Outside
channel:   5   domain: USA      lan: 10baseT
IP_address: 192.168.155.90  subnet_mask: 255.255.255.0
gateway: 192.168.155.1
RTS_threshold: 2301    Frag_threshold: 2346
short_retry: 15  long_retry: 15
sifs_time: default   protocols: all
telnet: On:23  web: On:80
lock: off
manager: Specific      admission: All
basic rates: 1000 2000 kbits/s
wep-key range: min 128 max 128
community_get: public   community_set: private
Specific Managers (manager_IP):
    192.168.155.91 <accept requests> <no traps>
    <empty>
    <empty>
    <empty>
Radio NID: 00E003056F7F  Management NID: 00E003802F88
```

Configuring the radio network

The most basic parameter for the radio network is the channel to be used. Several channels are defined by the standard, but not all are allowed in each geographical area where 802.11 equipment is sold. To ensure that a legal channel is selected, the A032 also includes a way to set the regulatory domain, which prevents inappropriate radio channels from being selected. This information can be set in the web interface from the basic setup screen shown in Figure 14-14.

The ESSID, or network name, is configured with the *net_name* parameter on the command line. Spaces are allowed if the network name is enclosed in quotation marks. The network name must be shared by every access point within the BSS.

```
CMD:set domain usa
Configuration has been updated
CMD:set channel 11
Configuration will be updated on next restart
CMD:set net_name "HEP Net"
Configuration will be updated on next restart
CMD:set ap_name Gluon
Configuration will be updated on next restart
```

Older Nokia access points had two modes of operation. One mode was in accord with 802.11 standards, and the second was a proprietary mode that has been eliminated in current shipping versions.

Exhibit 1019
Page 285 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Several 802.11 parameters can be set only at the command line:

Short retry counter (short_retry)
: By default, the short retry is set to 15, though it can be set as high as 31.

Long retry counter (long_retry)
: By default, the long retry is also set to 15, though it can also be set as high as 31.

RTS threshold (rts_threshold)
: By default, the RTS threshold is set to 2,301 bytes. It can be set as low as 1 byte or as high as 2,500 bytes.

Fragmentation threshold (frag_threshold)
: By default, the fragmentation threshold is 2,346 bytes as prescribed by the 802.11 standard. It may be set as low as 257 or as high as 2,346.

```
CMD:set short_retry 29
Configuration will be updated on next restart
CMD:set long_retry 7
Configuration will be updated on next restart
CMD:set rts_threshold 512
Configuration will be updated on next restart
CMD:set frag_threshold 1024
Configuration will be updated on next restart
```

The A032 also allows adjustment of the SIFS time, though this should not be necessary in practice because the interface is set to comply with the 802.11 standard.

Configuring the wired network, bridging, and management parameters

The predecessor to the A032 had both a 10baseT (twisted-pair RJ-45) and a 10base2 ("Thinnet") interface. In recognition of the dominance of twisted-pair physical connections, the A032 comes with only an RJ-45 port.

For management purposes, an IP address and default route can be assigned to the A032. The address may be used as a default gateway by associated wireless stations, but in practice, it often makes more sense to use the IP address simply for device management purposes. The IP interface is configured with the *ip_address*, *subnet_mask*, and *gateway* parameters:

```
CMD:set ip_address 10.1.2.100
Configuration will be updated on next restart
CMD:set subnet_mask 255.255.255.0
Configuration will be updated on next restart
CMD:set gateway 10.1.2.1
Configuration will be updated on next restart
```

One additional setting can be applied to the wired network. The A032 can be configured to bridge only TCP/IP frames. Many other link-layer protocols rely heavily on multicast frames, and if the A032 is connected to such a network, the available transmission time on the wireless network will dwindle as the multicasts are forwarded:

```
CMD:set protocols tcpip
Configuration will be updated on next restart
```


Exhibit 1019
Page 286 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

The command-line interface is normally accessible over Telnet connections on port 23, and the web interface is available on port 80. Both of these services can be run on arbitrary ports or disabled to prevent management. To disable a service, simply use the value off with the appropriate parameter:

```
CMD:set telnet 23
Configuration will be updated on next restart
CMD:set web off
Configuration will be updated on next restart
```

All of these parameters can be configured through the web interface by using the advanced access point setup link on the lefthand bar. The advanced access point page is shown in Figure 14-15.

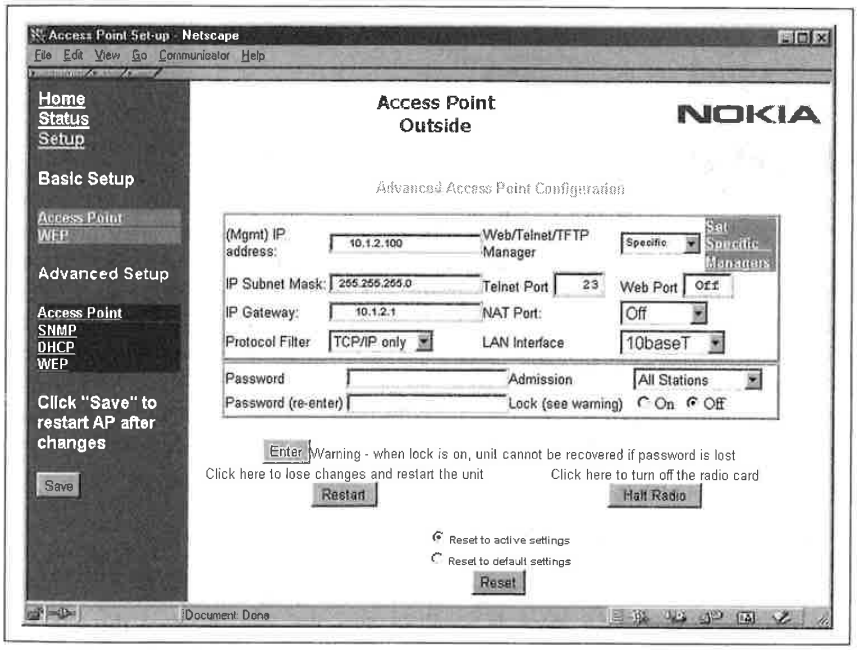


Figure 14-15. Network interface configuration

Viewing the next restart configuration

Configuration changes to the A032 do not take effect immediately. When the system starts up, it reads the configuration from nonvolatile memory. Many configuration commands can be used with + appended to show the settings that will be effective on the next restart. For example, *config+* can be used to show the network settings that will be effective on the next restart:

```
CMD:config+
=====Next Configuration (effective after restart)======
```


Exhibit 1019
Page 287 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

```
net_name: HEP Net      AP_name: Gluon
channel:  11   domain: USA      lan: 10baseT
IP_address: 10.1.2.100  subnet_mask: 255.255.255.0
gateway: 10.1.2.1
RTS_threshold: 512    Frag_threshold: 1024
short_retry: 29  long_retry: 7
sifs_time: default   protocols: TCPIP
telnet: On:23  web: Off
lock: off
manager: Specific      admission: All
basic rates: 1000 2000 kbits/s
wep-key range: min 128 max 128
community_get: public   community_set: private
Specific Managers (manager_IP):
    <empty>
    <empty>
    <empty>
    <empty>
Radio NID: 00E003056F7F  Management NID: 00E003802F88
```

Configuring DHCP

DHCP is useful with access points because it minimizes client configuration. The A032 includes a DHCP server that is suitable for small networks; larger networks may wish to take advantage of more generic solutions. The DHCP server can be allocated a pool of addresses. The administrator specifies both the DHCP base address (*dhcp_base*) and the size of the pool (*dhcp_pool*). The A032 DHCP server can also be configured to hand out a default gateway and DNS server information with the *dhcp_gateway* and *dhcp_dns* parameters:

```
CMD:set dhcp_base 10.1.2.200
Configuration will be set at next restart
CMD:set dhcp_pool 50
Configuration will be set at next restart
CMD:set dhcp_gateway 10.1.2.1
Configuration will be set at next restart
CMD:set dhcp_dns 192.168.100.5
Configuration will be set at next restart
```

The DHCP server configuration is accessed over the web interface through the DHCP link in the advanced setup section. The parameters that can be set over the web interface should be evident from Figure 14-16.

Security Configuration

The A032 supplies three main security tools for the network administrator. All are based on 802.11 and therefore inherit all the security limitations of 802.11. The three tools are WEP, filtering of management connections, and filtering of associated stations.

Exhibit 1019
Page 288 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

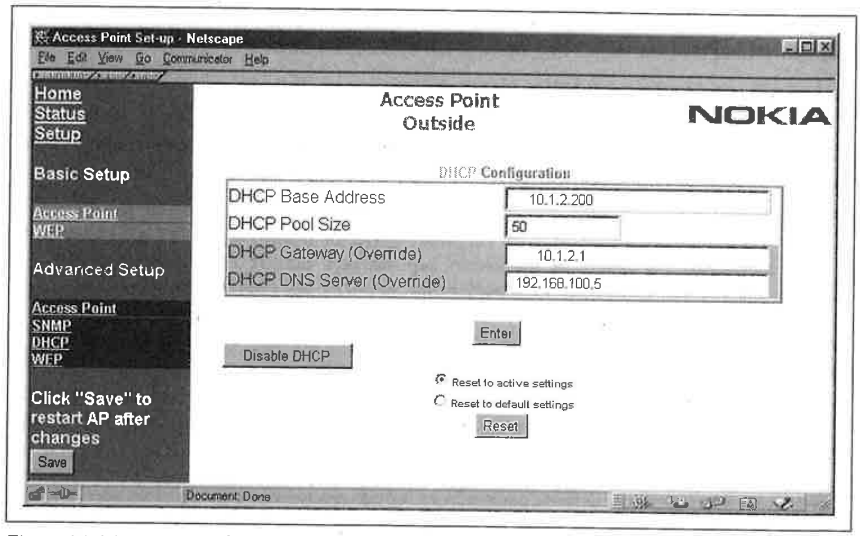


Figure 14-16. DHCP configuration

Configuring WEP

WEP keys can be stored locally on the A032 or fetched from a RADIUS server. Local keys are entered by the administrator and stored on the device. They are configured by setting the *wep_key* parameter to *set*. Unlike many other parameters, *wep_key* takes two values: the key number and its value. To erase a key, the value *n* can be used. As mandated by the specification, four keys can be entered; one must be set active by using the *wep_key_active* parameter:

```
CMD:set wep_key 1 BF8D2E9D0D6F669F4A36DA254FA651AC
Configuration will be set at next restart
CMD:set wep_key 2 n
Configuration will be set at next restart
CMD:set wep_key_active 1
Configuration will be set at next restart
```

The A032 provides two WEP-related settings to help control associations. One is the *WEP mode*. Normally, the WEP mode is set to *wep*, which means that stations can use either a shared WEP key or a key from RADIUS to authenticate the station to the access point. The mode can also be set to *open*, for no authentication, *personal*, to require a key from RADIUS, or *wifi*, for compatibility with equipment that attempts open authentication but transmits using a shared key for encryption.

```
CMD:set wep_mode wep
Configuration will be updated on next restart
```

Additionally, the administrator can specify a required key length so only longer keys are permitted. The *wep_key_range* parameter takes two arguments, the minimum

Exhibit 1019
Page 289 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

and maximum key lengths, specified in secret key bits. To allow all users to connect, allow anything from a 40-bit key up to a 128-bit secret key:

```
CMD:set wep_key_range 40 128
Configuration will be set at next restart
```

Alternatively, the key range can be set to *normal* to allow only 40-secret-bit keys or *high* to allow only 128-secret-bit keys:

```
CMD:set wep_key_range high
Configuration will be set at next restartAs noted in Chapter 5, Nokia quotes all WEP
key lengths as the length of the secret key. Nokia's 128-bit keys have 128 secret
bits and are not compatible with most other vendors' implementations.
```



As with all other command-line work, the settings described in this section can be made through the Advanced WEP configuration page. Figure 14-7 shows this page, set to use the local (device) key database. RADIUS keys are discussed in the next section.

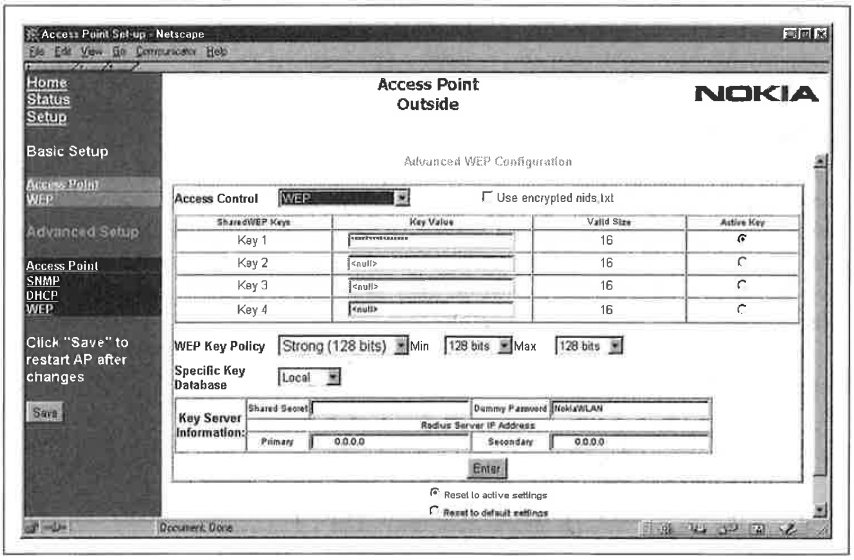


Figure 14-17. Advanced WEP configuration

Personal WEP and RADIUS

Personal WEP is something of a misnomer because it does not map keys to users. It maps keys to MAC addresses. Each MAC address must be configured as a user account on the specified RADIUS server. When the A032 receives an association request from a station, it queries the RADIUS server for the WEP key stored in the

Exhibit 1019
Page 290 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

RADIUS database for the MAC address in the association request. After receiving the WEP key, the association proceeds normally.

Keys are stored in the RADIUS server under the MAC addresses that are allowed. A RADIUS "account" for each card must be created. When the A032 receives an association request, it sends an authentication request to the server consisting of the client's MAC address and the dummy password in Figure 14-17. The RADIUS server returns an acceptance message if there is an "account" for the MAC address that includes the client's WEP key in its response.

Filtering management connections

It is conceivable that an access point may need to be configured to allow management connections only from selected stations. If access is restricted using the *set manager specific* command, the A032 checks each incoming management connection against the list and validates its access privileges. Stations can be added to the list using the *set manager_ip* command, which takes three arguments. The first is the IP address of the manager. The second and third arguments are for accepting requests and sending SNMP traps; a setting of 1 is used to enable the function, and a setting of 0 is used to disable it. To allow management connections from 10.1.2.88, use the following commands:

```
CMD:set manager_ip 1 10.1.2.88 1 0
Configuration will be set at next restart
CMD:set manager specific
Configuration will be updated on next restart
```

Specific manager configurations can be updated from the "Set Specific Managers" link in the Advanced Access Point Set-up, as shown in Figure 14-18.

Naming allowed stations

For ease of management of a number of features on the A032, MAC addresses can be associated with text entries called *network identifiers* (NIDs). Naming stations is done only at the command line using the *nid* command, which takes one of three arguments, all of which are self-explanatory:

```
CMD:nid add 00005e000145 Matthew
CMD:nid list
----NID----- ------Name------ M
00005e000145 Matthew         ,NH
CMD:nid delete 00005e000145 Matthew
CMD:nid list
<list empty>
```

NID mappings can be retrieved from or sent to the access point using TFTP. Simply fetch or send the file named *nids.txt* from the access point's IP address. If management filtering is on, the TFTP client must be on the manager list. In the file itself, the last column in the NID list is used to note whether the entry is a station (N) or a

Exhibit 1019
Page 291 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

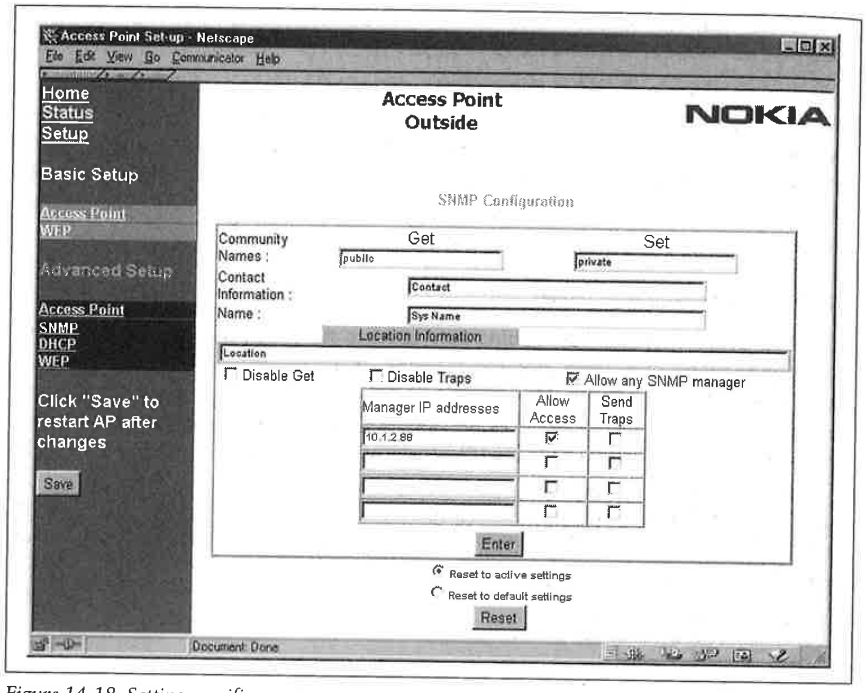


Figure 14-18. Setting specific managers

bridge (B). Bridge entries are relevant only when the A032 is deployed as a wireless bridge.

```
[gast@bloodhound]$ tftp a032
tftp> get nids.txt
Received 140 bytes in 0.0 seconds
tftp> quit
[gast@bloodhound]$ cat nids.txt
/ NID/Bridge list for AP(Outside) on Sat, 24 Nov 2001 23:12:39
49f7e6ca2f478073a1dcdf2cfeba08cc
00005e000145 Matthew          ,NH,
```

Up to 200 name/MAC pairs can be stored. No facility is provided by Nokia to synchronize the lists. System administrators must develop scripts to keep a single copy of *nids.txt* up to date and push it out to access points using TFTP.

Restricting associations

MAC address filtering is performed using the NIDs. Associations can be restricted to "named" stations. If a station does not have an entry in the NID list, it is not allowed to associate:

```
CMD:set admission named
Configuration will be updated on next restart
```


Exhibit 1019
Page 292 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Address filtering is enabled with the web interface by restricting admission to named stations, as shown in Figure 14-19.

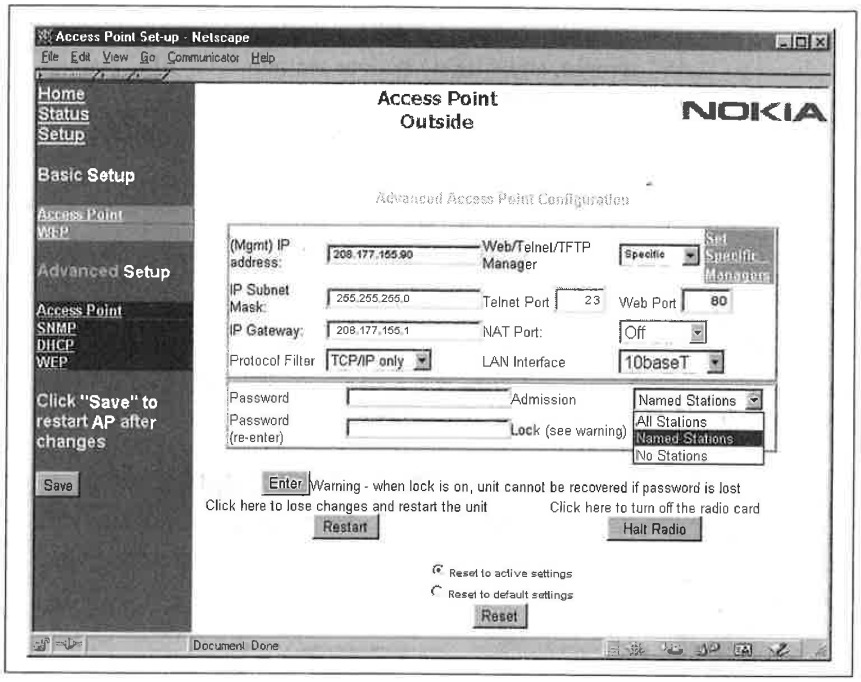


Figure 14-19. Enabling address filtering for associations

Monitoring Wireless Stations

802.11 is a link-layer protocol, and monitoring the link-layer associations is important to troubleshooting. The status menu in the web interface allows the administrator to see all the associated wireless stations, all the known wireless stations, and all the known MAC addresses.

All the currently associated stations are listed with *show a*. Stations with a name-to-MAC mapping are displayed by name:

```
CMD:show a
Total bridge entries     :6
Number Wireless Stations :1
   MAC Address   State       Channel Power IP Address
c110             Associated   11     On    10.1.2.91
```


Exhibit 1019
Page 293 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

The difference between "all associated stations" and "all wireless stations" is that the latter includes any wireless stations learned about from another access point. All wireless stations are displayed with *show s*.

All known stations are printed with *show g*. Stations connected to the Ethernet port are shown as "LAN" stations, and wireless stations are shown as "AIR" stations. A few MAC addresses are reserved for internal use.

```
CMD:show g
Total bridge entries     :6
Number Wireless Stations :1
   Net ID        Interface  IP Address
0000C06F37EE        LAN     208.177.155.88
001067003F86        LAN     208.177.155.1
c110                AIR     208.177.155.91
00E003056F7F       Internal
00E003000000       Internal
00E003802F88       Internal  208.177.155.90
```

show g also provides the content for the "Network Summary" in the web interface, as shown in Figure 14-20. Internal stations are not displayed in the web interface.

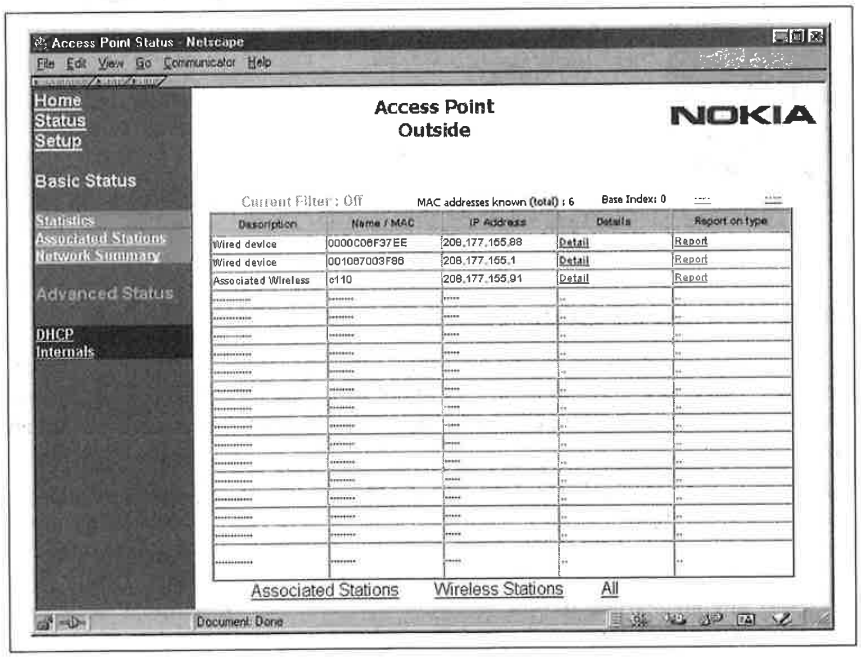


Figure 14-20. Network summary page

Exhibit 1019
Page 294 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Monitoring System Resources and Load

Counters can be monitored at the command line using the *stats* command. Statistics for both the wired and wireless interfaces are maintained, and they can be viewed by using *stats lan* and *stats air*:

```
CMD:stats lan
Stats last cleared : Sat, 24 Nov 2001 23:56:18
Accumulation time (secs) :       661

LAN Statistics:          Last 10s  Cumulative
tx lan frames     :       4358      23747
tx data bytes     :     371680    2039080
rx lan frames(all):       5952      36074
rx frames accepted:       5948      30980
rx data bytes     :     513936    2717364
rx discard frames :          0        311
CMD:stats air
Stats last cleared : Sat, 24 Nov 2001 23:56:18
Accumulation time (secs) :       663

Air Statistics:        Last 10s  Cumulative
tx frames :              5972      31657
tx bytes :             630273    3529370
tx fail  :                  0         62
rx frames(all) :         4360      23882
rx frames(mgmnt):           0          3
rx frames(data):         4360      23879
rx data bytes  :       441773    2433732
discard frames :            0        116
```

Software versions and basic system information are shown on the "Advanced Internals" status page in the web interface, reproduced in Figure 14-21. The bar graph at the bottom gives some indication of the system load. The buffer load shows the percentage of buffer memory used. Under normal situations, it should be 60% or less. System loading is the processor utilization. Radio usage is self-evident. Obviously, the access point shown in Figure 14-21 is under a heavy load.

Troubleshooting Tools

Troubleshooting is not the A032's strong point. The only network troubleshooting tool supplied with the A032 firmware is the lowly *ping*, and it works as expected. *traceroute* is not included.

Exhibit 1019
Page 295 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

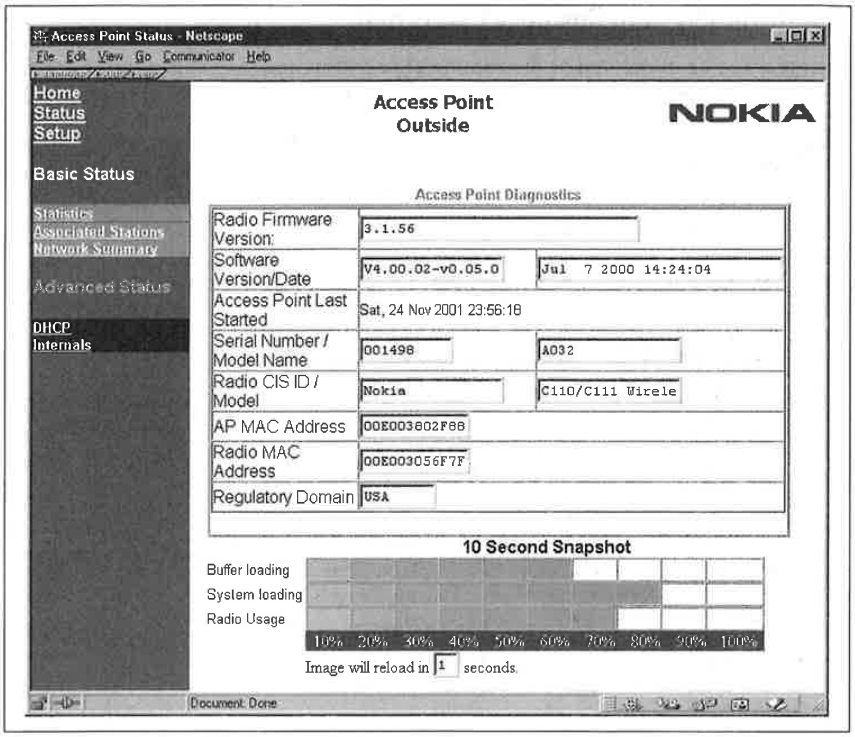


Figure 14-21. Advanced Internals status page

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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?

Exhibit 1019
Page 297 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

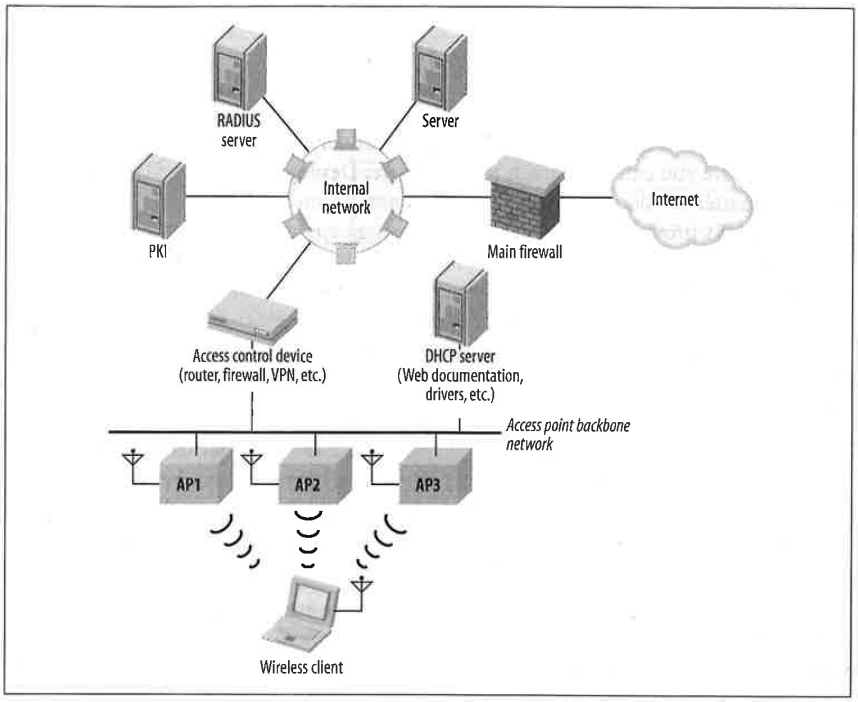


Figure 15-1. Standard wireless LAN deployment topology

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.

Exhibit 1019
Page 298 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 299 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Exhibit 1019
Page 300 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

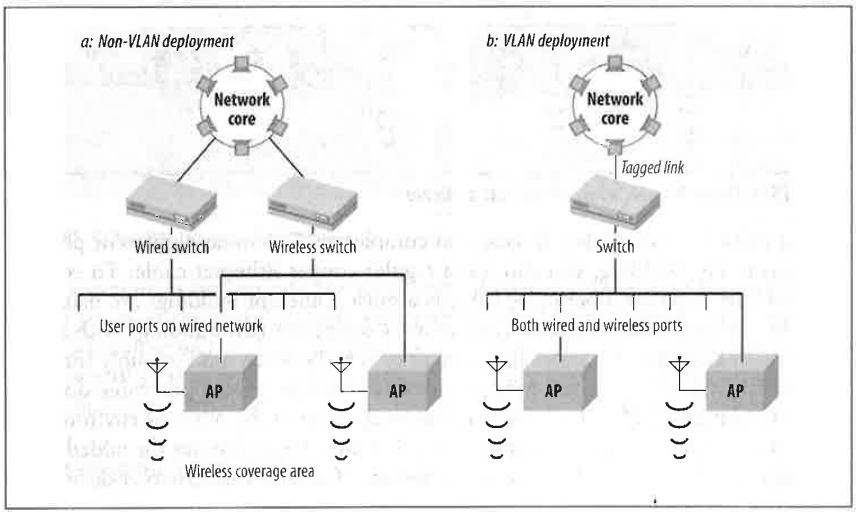


Figure 15-2. Physical topologies for 802.11 network deployment

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.

Exhibit 1019
Page 301 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

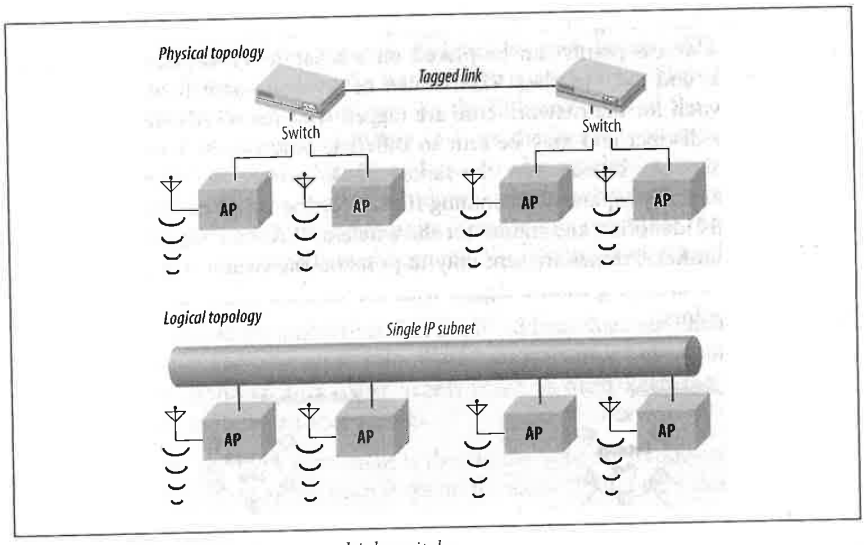


Figure 15-3. Using VLANs to span multiple switches

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

Exhibit 1019
Page 302 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

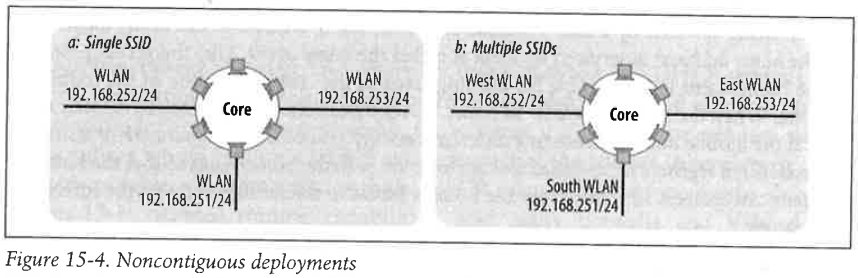


Figure 15-4. Noncontiguous deployments

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.

Exhibit 1019
Page 303 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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—

Exhibit 1019
Page 304 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 305 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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."

Exhibit 1019
Page 306 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 307 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 308 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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*.

Exhibit 1019
Page 309 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 310 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Exhibit 1019
Page 311 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 312 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Exhibit 1019
Page 313 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 314 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 315 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 316 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Exhibit 1019
Page 317 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 318 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Exhibit 1019
Page 319 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 320 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Exhibit 1019
Page 321 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

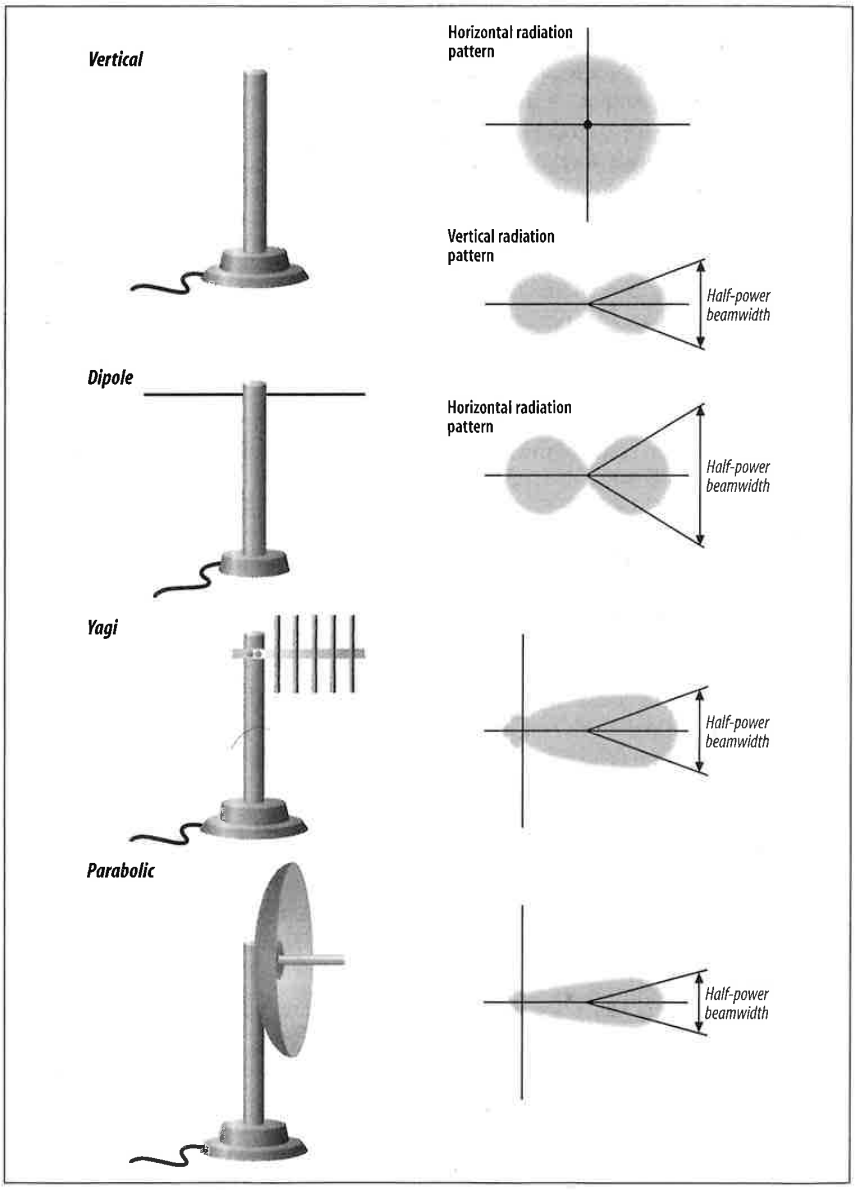


Figure 15-5. Antenna types

Exhibit 1019
Page 322 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 323 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 324 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 325 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148



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

Exhibit 1019
Page 326 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 327 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

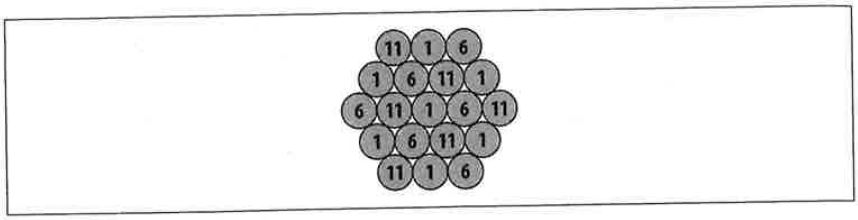


Figure 15-6. Frequency planning

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.

Exhibit 1019
Page 328 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 329 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

- 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).

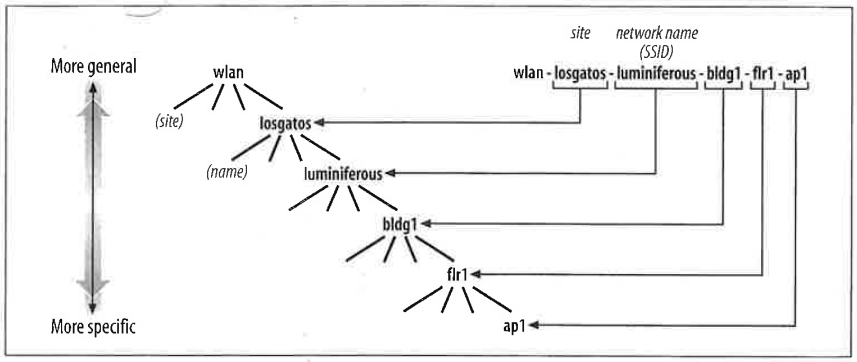


Figure 15-7. Wireless LAN naming convention for access points

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.

Exhibit 1019
Page 330 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

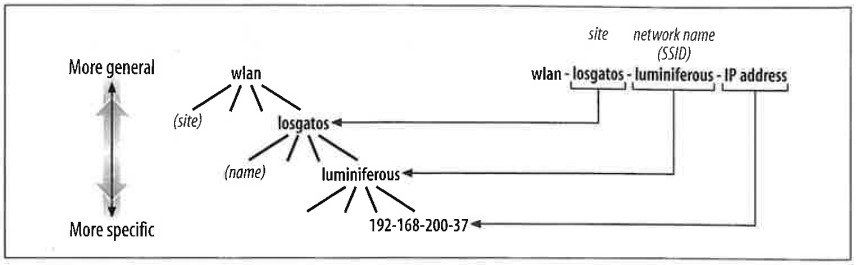


Figure 15-8. Convention for naming wireless LAN stations

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.

Exhibit 1019
Page 331 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 332 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Exhibit 1019
Page 333 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 334 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Exhibit 1019
Page 335 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 336 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 337 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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/
```


Exhibit 1019
Page 338 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

```
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

Exhibit 1019
Page 339 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Exhibit 1019
Page 340 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

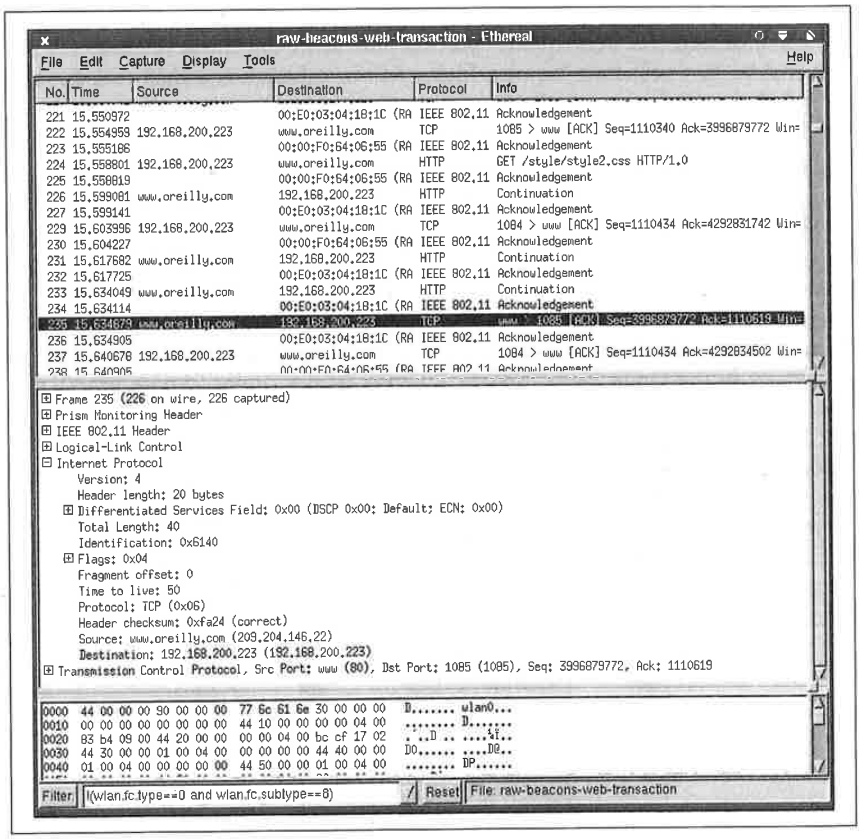


Figure 16-1. Main Ethereal window

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

Exhibit 1019
Page 341 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

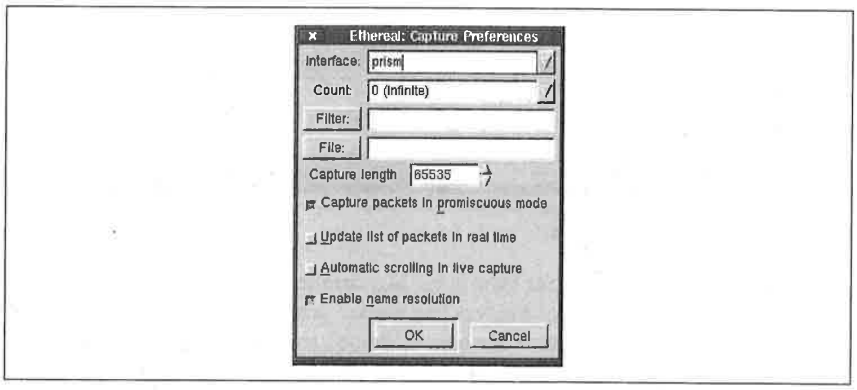


Figure 16-2. Selecting the prism pseudo-interface

Exhibit 1019
Page 342 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

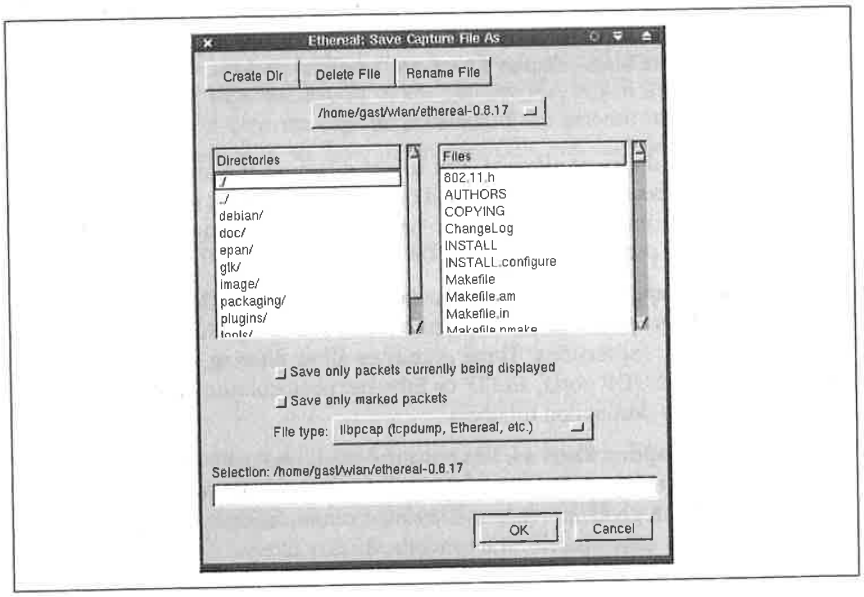


Figure 16-3. Save As dialog box

Exhibit 1019
Page 343 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Exhibit 1019
Page 344 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 345 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

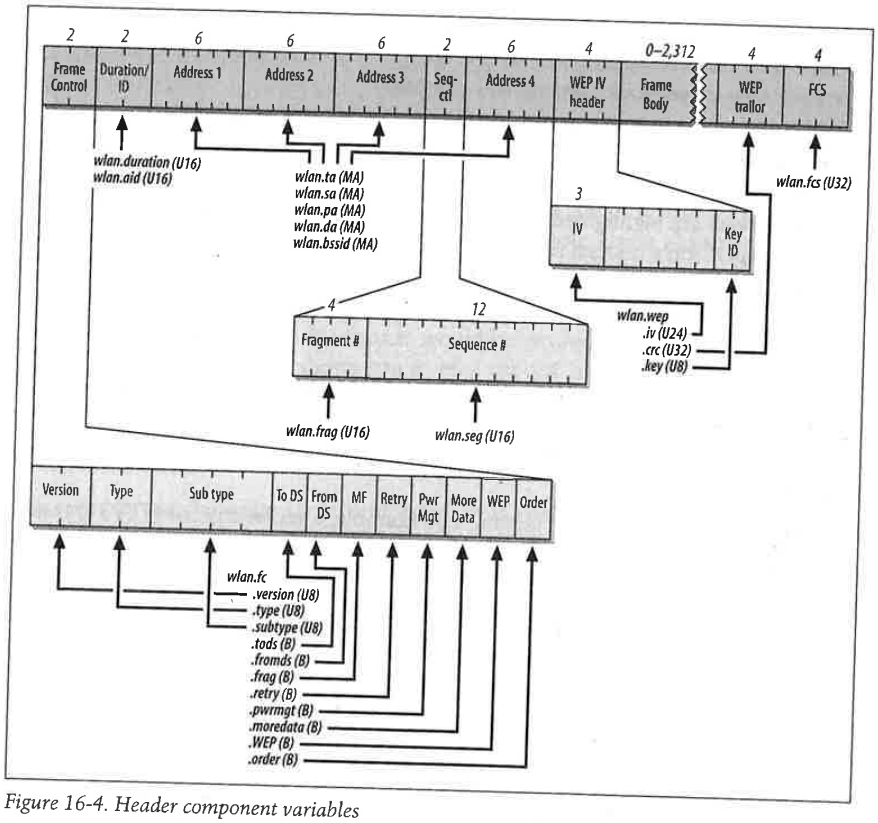


Figure 16-4. Header component variables

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*

Exhibit 1019
Page 346 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 347 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

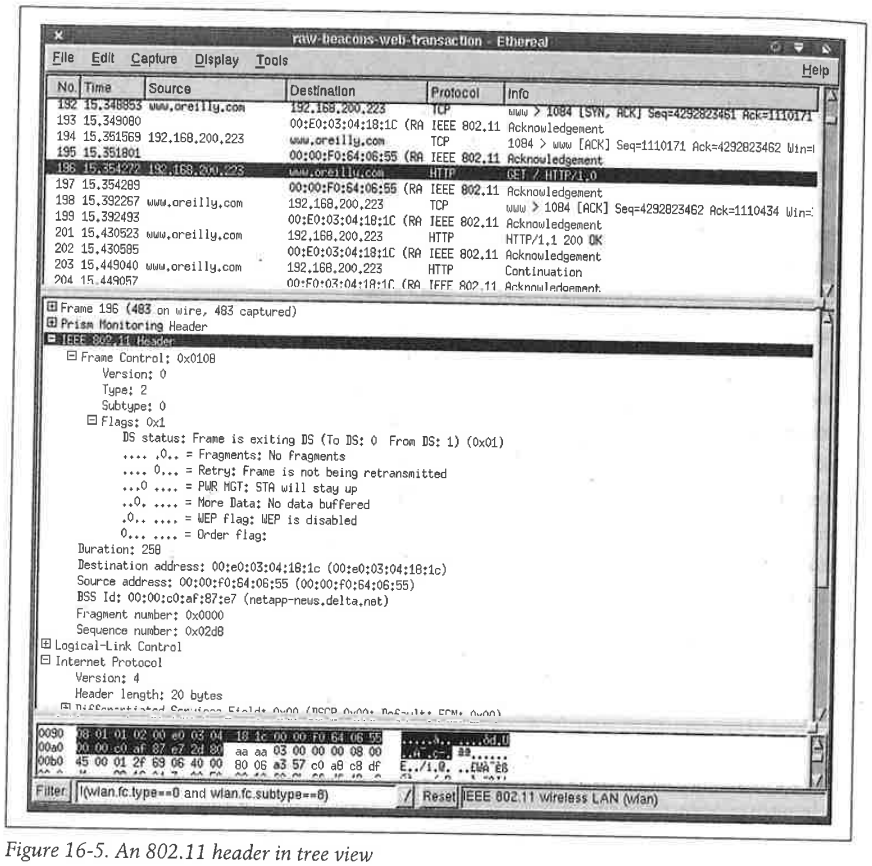


Figure 16-5. An 802.11 header in tree view

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*

Exhibit 1019
Page 348 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 349 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

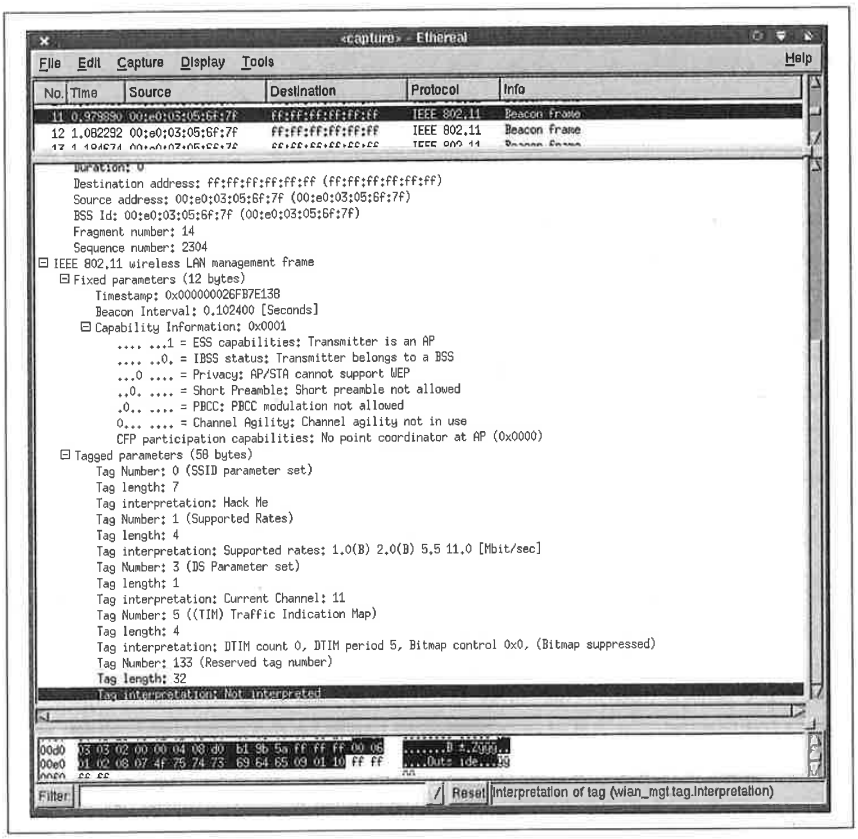


Figure 16-6. Beacon frame decode

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.

Exhibit 1019
Page 350 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

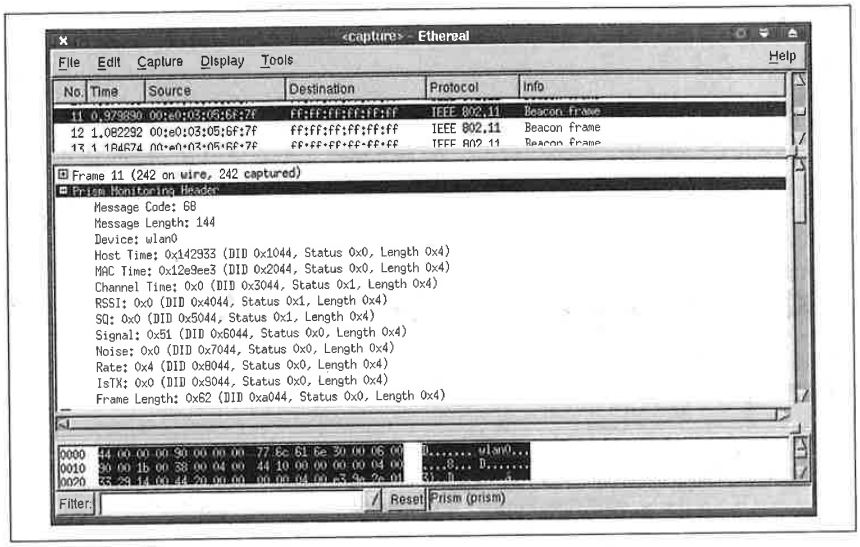


Figure 16-7. PRISM monitoring header

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.

Exhibit 1019
Page 351 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

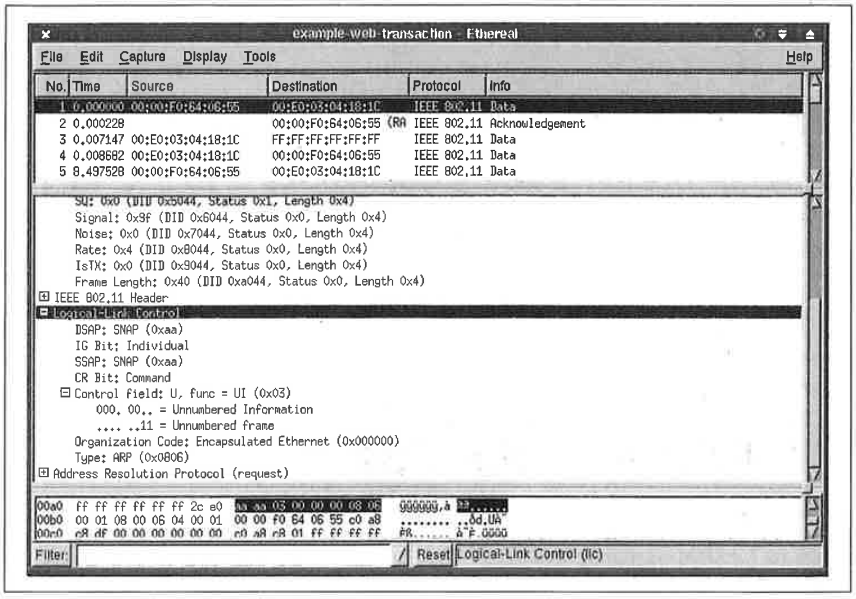


Figure 16-8. LLC SNAP header

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

Exhibit 1019
Page 352 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

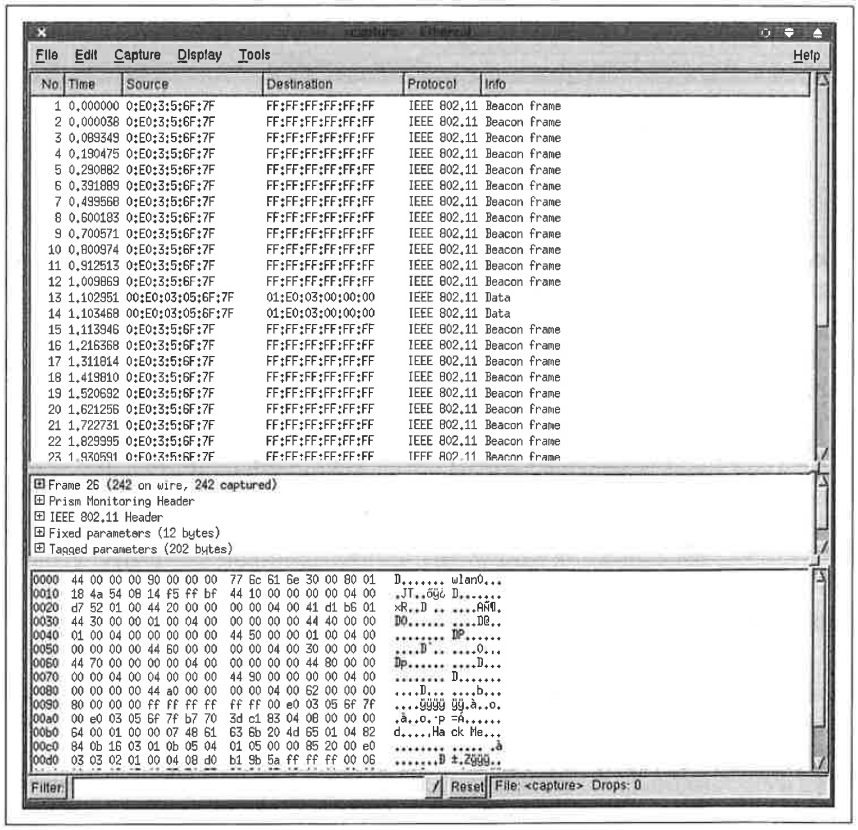


Figure 16-9. Trace of idle 802.11 network

Exhibit 1019
Page 353 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

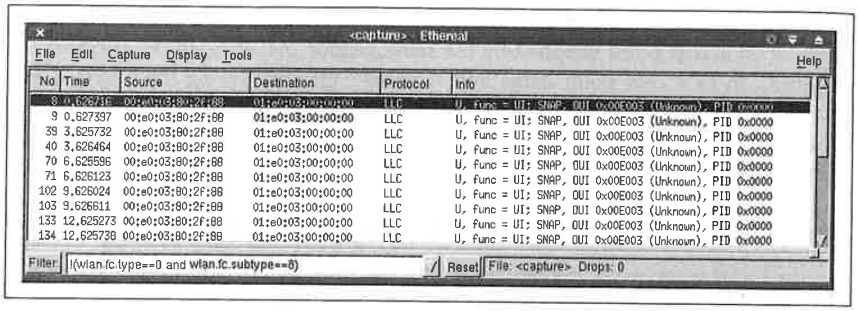


Figure 16-10. The leftover Data frames

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

Exhibit 1019
Page 354 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

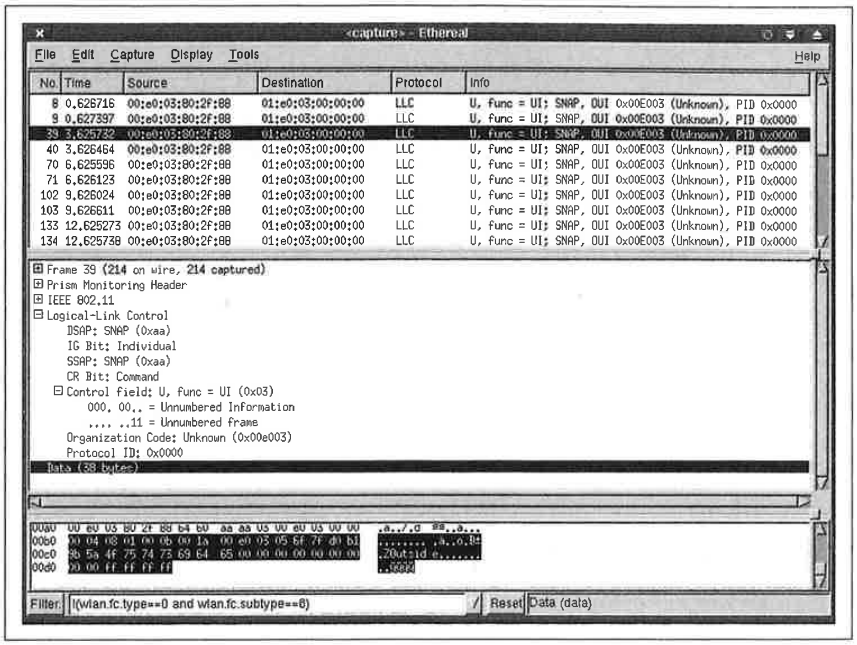


Figure 16-11. One of the random 802.11 Data frames and its LLC data

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.

Exhibit 1019
Page 355 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

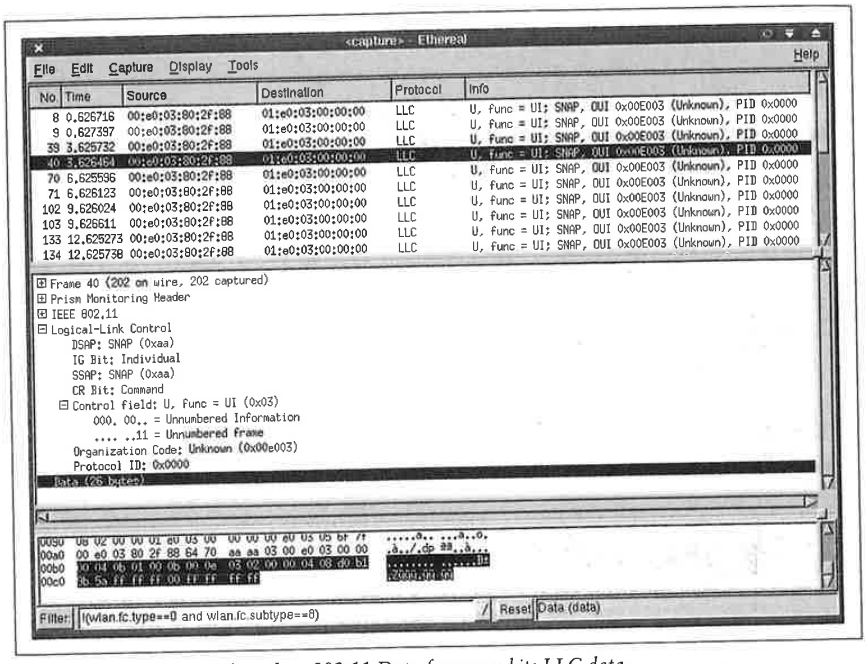


Figure 16-12. The second random 802.11 Data frame and its LLC data

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

Exhibit 1019
Page 356 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

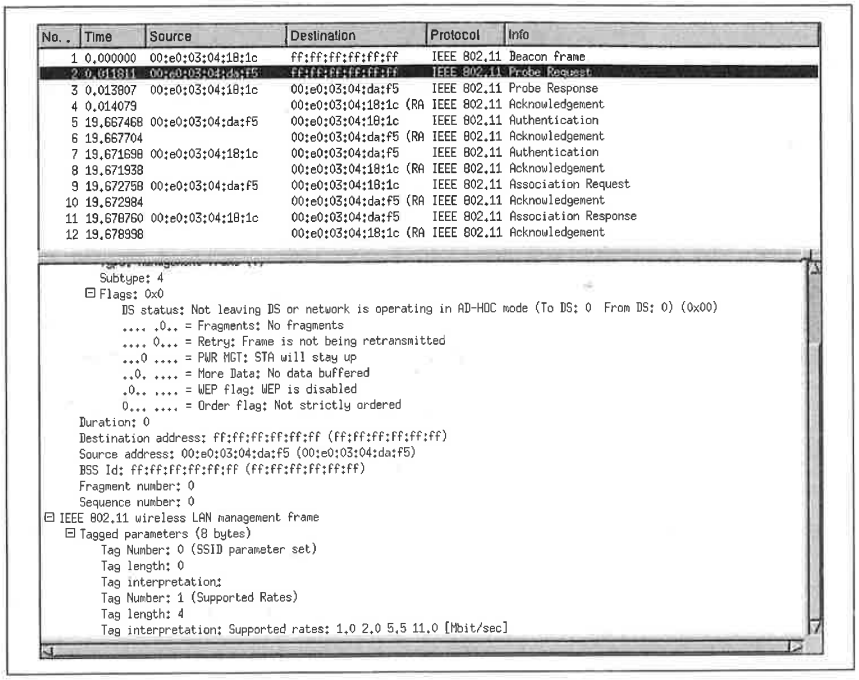


Figure 16-13. Probe Request frame

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.

Exhibit 1019
Page 357 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

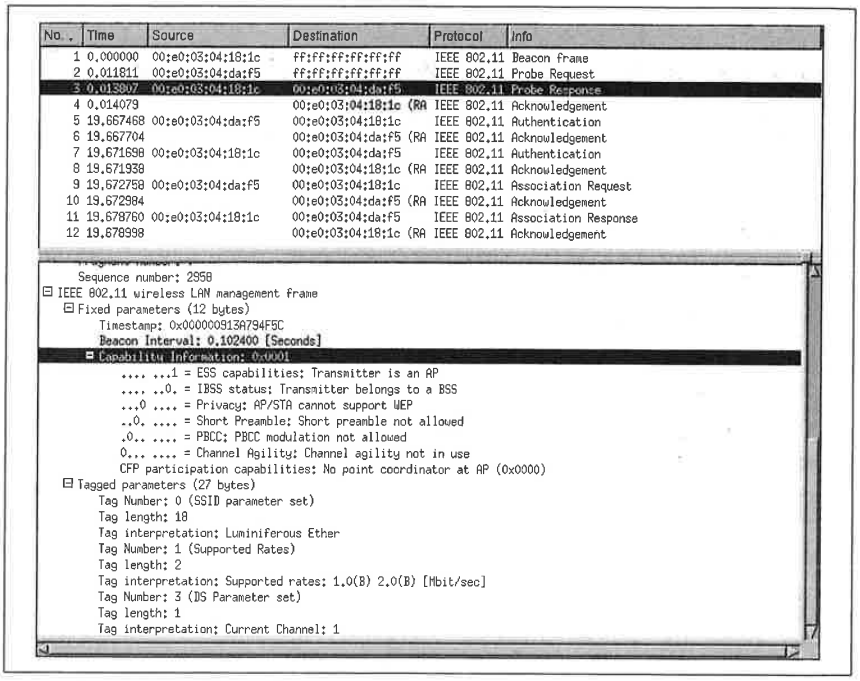


Figure 16-14. Expanded view of Probe Response frame

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

Exhibit 1019
Page 358 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

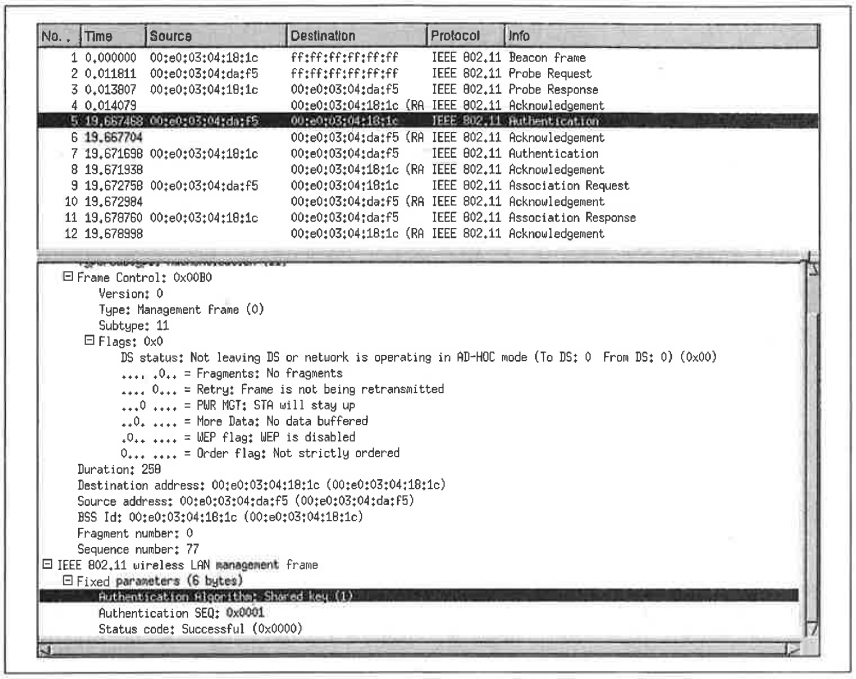


Figure 16-15. First authentication frame

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.

Exhibit 1019
Page 359 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

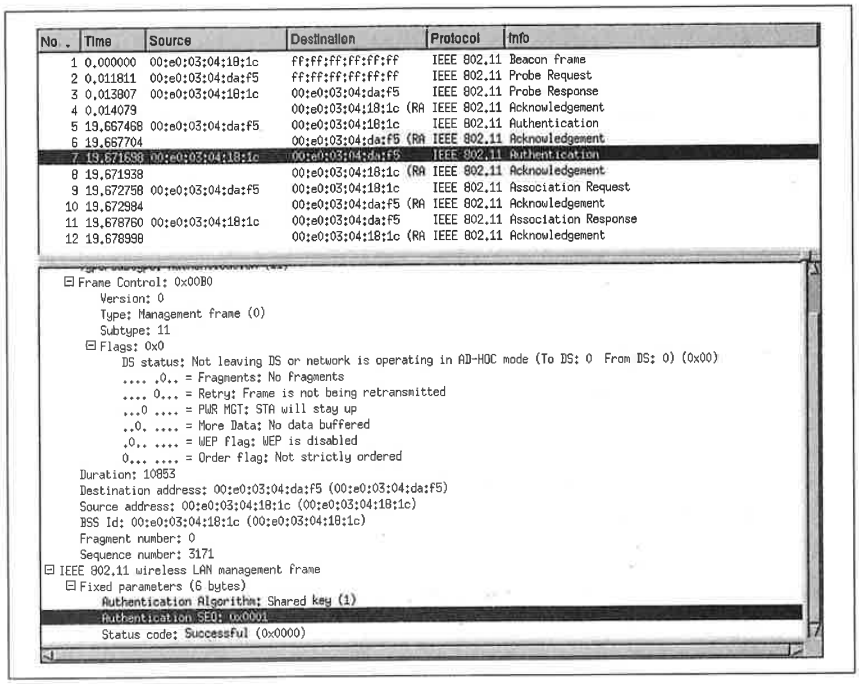


Figure 16-16. Second authentication frame

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.

Exhibit 1019
Page 360 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

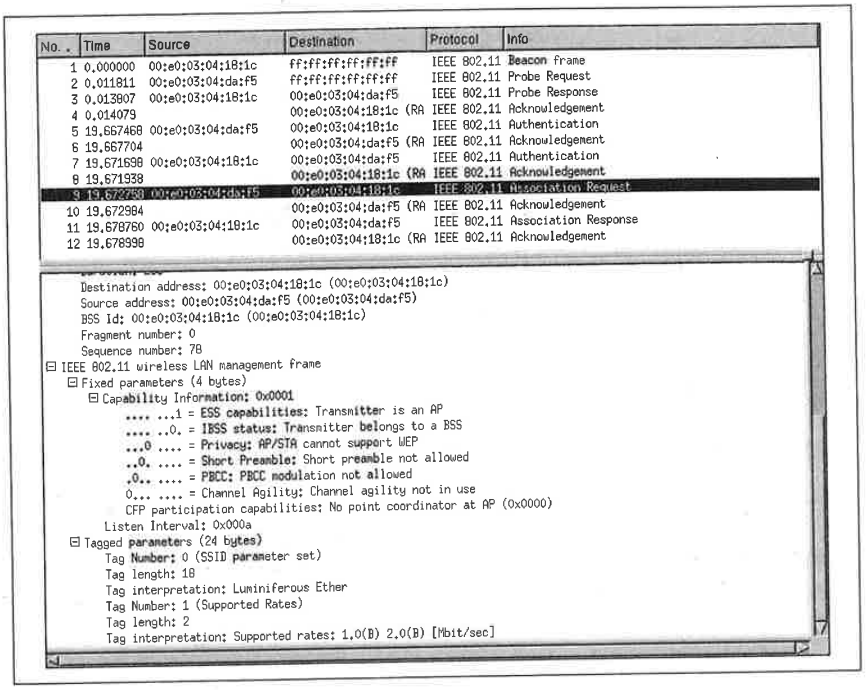


Figure 16-17. Association request

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.

Exhibit 1019
Page 361 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

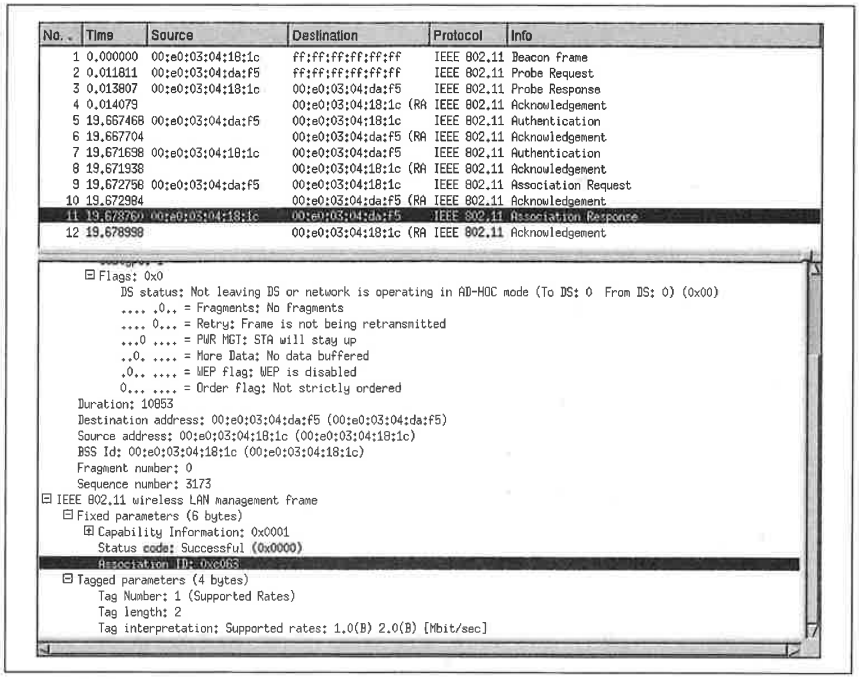


Figure 16-18. Association response

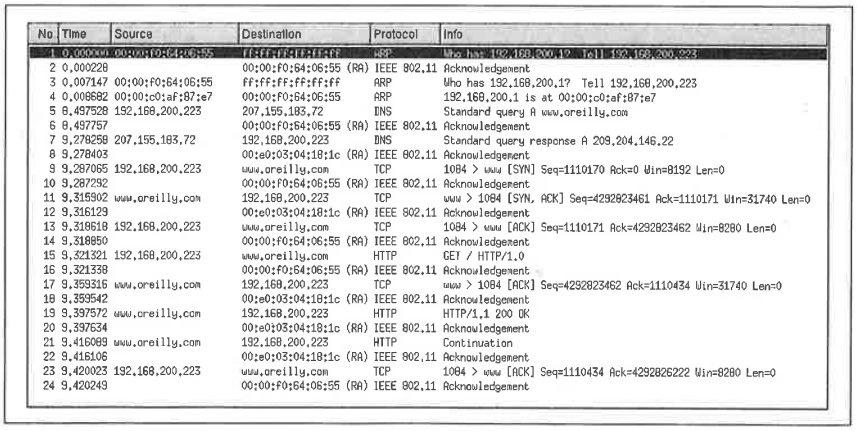


Figure 16-19. Full web site trace

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

Exhibit 1019
Page 362 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

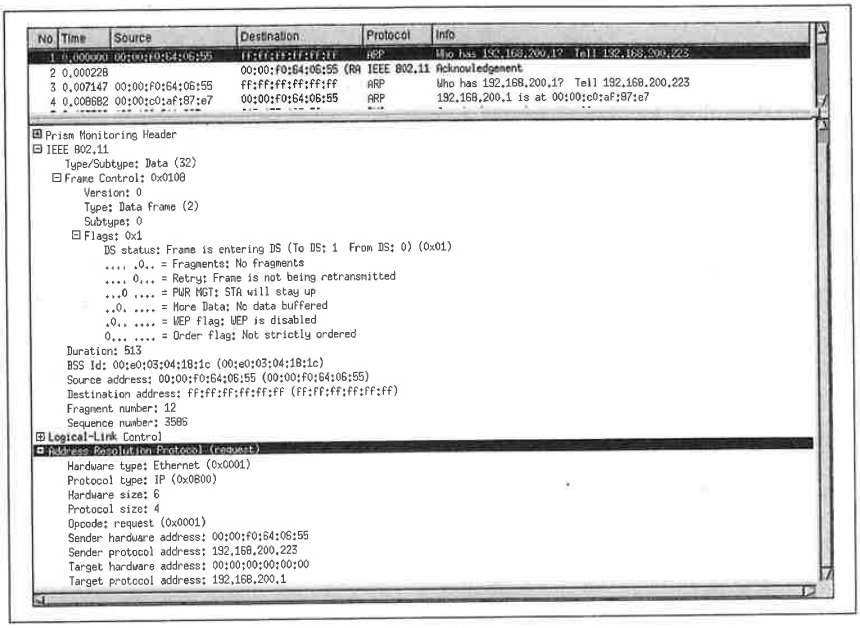


Figure 16-20. Initial ARP request

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*.

Exhibit 1019
Page 363 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

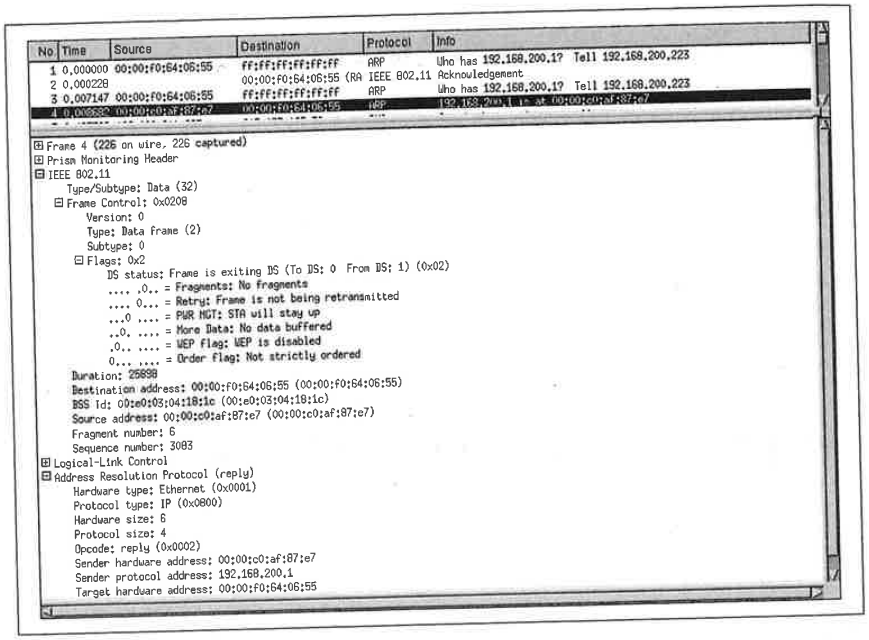


Figure 16-21. ARP reply

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.

Exhibit 1019
Page 364 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

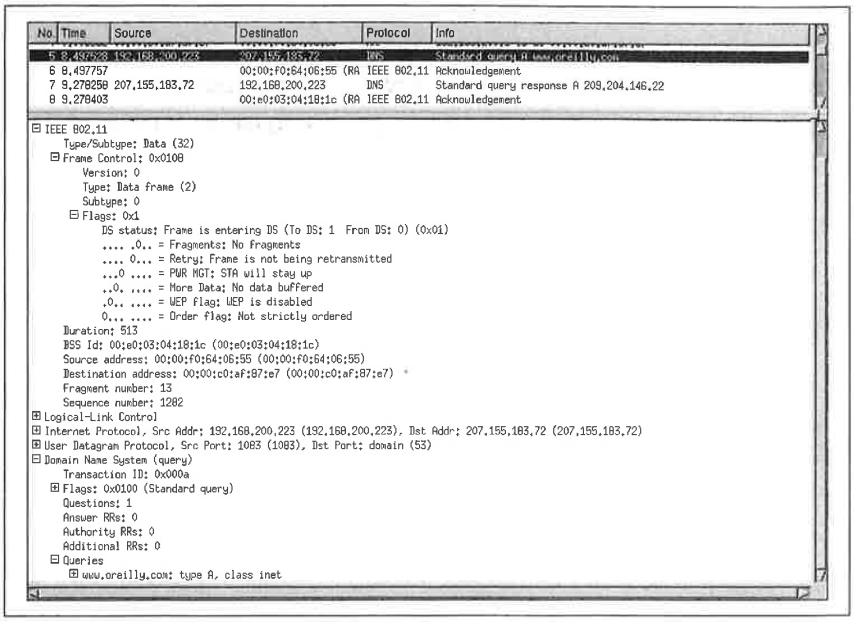


Figure 16-22. DNS request

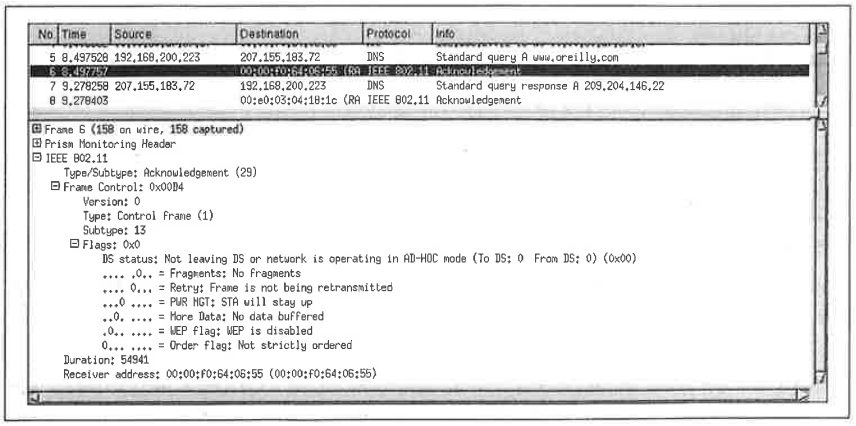


Figure 16-23. ACK of DNS request

Exhibit 1019
Page 365 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

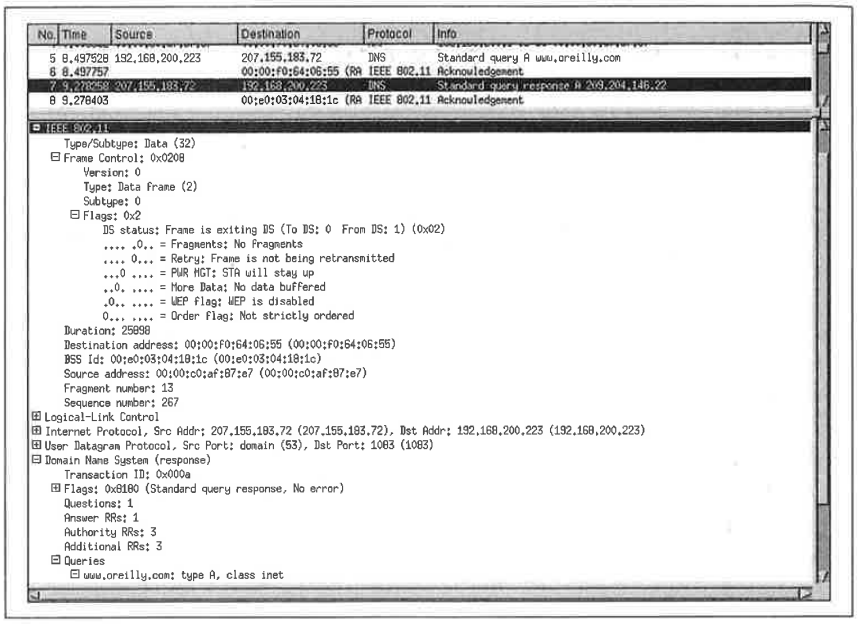


Figure 16-24. DNS reply

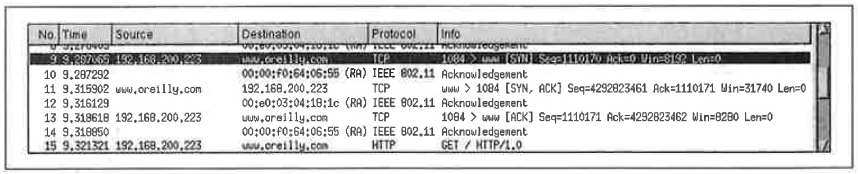


Figure 16-25. TCP three-way handshake

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.

Exhibit 1019
Page 366 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

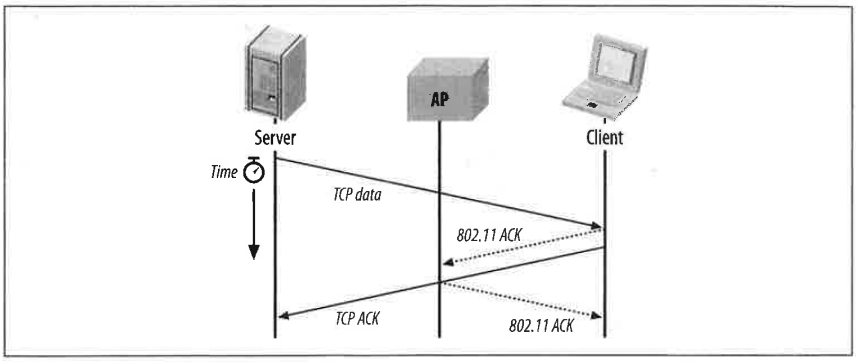


Figure 16-26. Frames corresponding to the bulk transfer of an HTTP connection

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.

Exhibit 1019
Page 367 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 368 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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
```


Exhibit 1019
Page 369 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

```
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.

Exhibit 1019
Page 370 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 371 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 372 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 373 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Exhibit 1019
Page 374 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 375 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Exhibit 1019
Page 376 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 377 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 378 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 379 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 380 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Exhibit 1019
Page 381 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 382 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Exhibit 1019
Page 383 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Exhibit 1019
Page 384 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Exhibit 1019
Page 385 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 386 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Exhibit 1019
Page 387 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

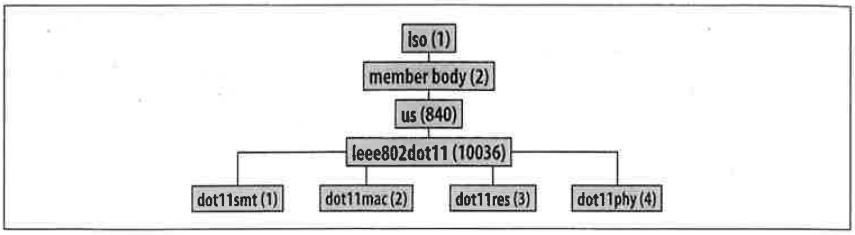


Figure A-1. The 802.11 MIB root and its main branches

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.

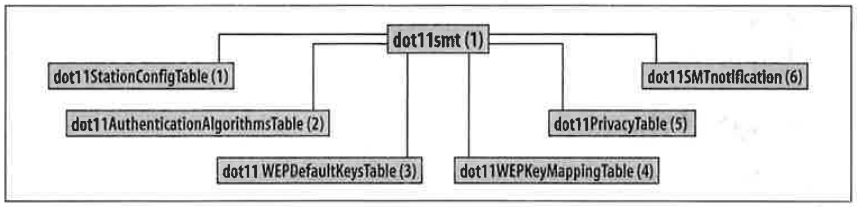


Figure A-2. Main branches of the station management (SMT) tree

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).

Exhibit 1019
Page 388 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

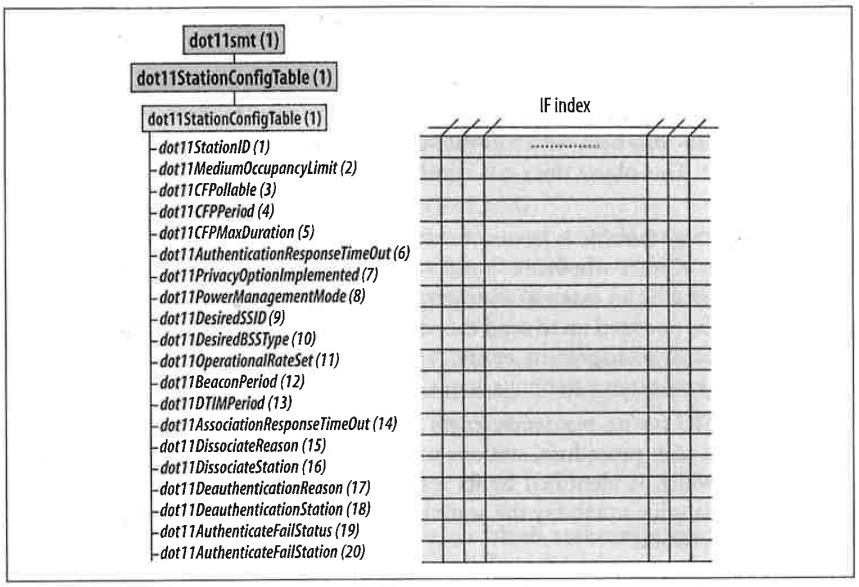


Figure A-3. The station configuration table dot11StationConfigTable

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

Exhibit 1019
Page 389 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 390 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

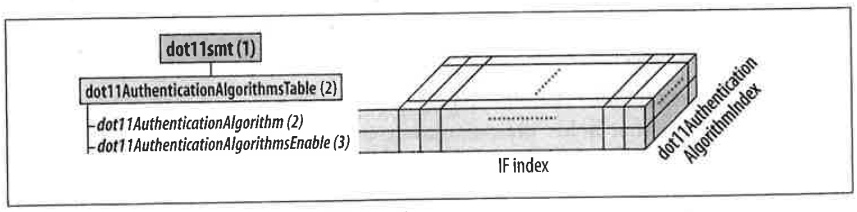


Figure A-4. The authentication algorithms table

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.

Exhibit 1019
Page 391 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

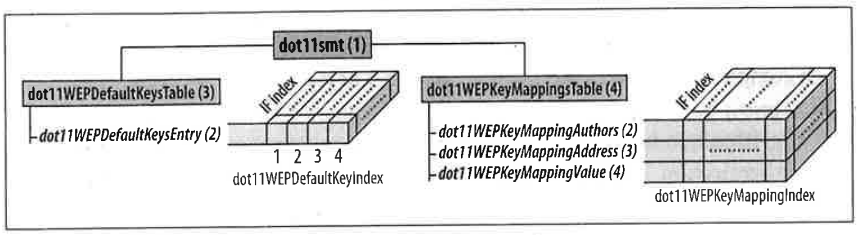


Figure A-5. WEP key tables

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.

Exhibit 1019
Page 392 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

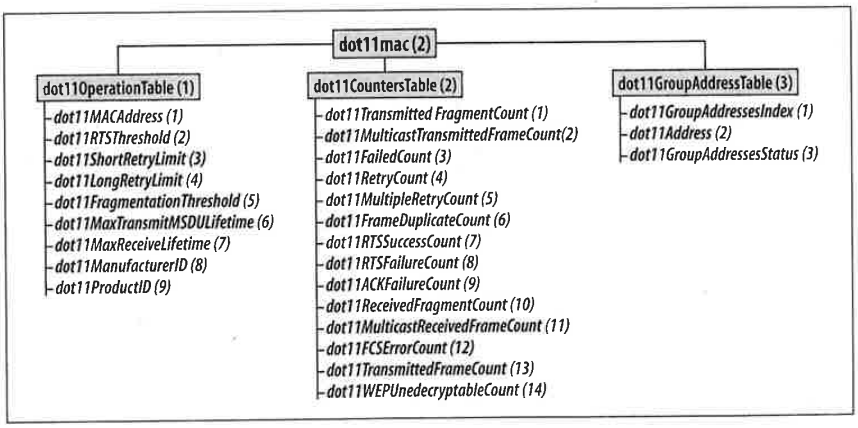


Figure A-6. The MAC attributes subtree

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.

Exhibit 1019
Page 393 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 394 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

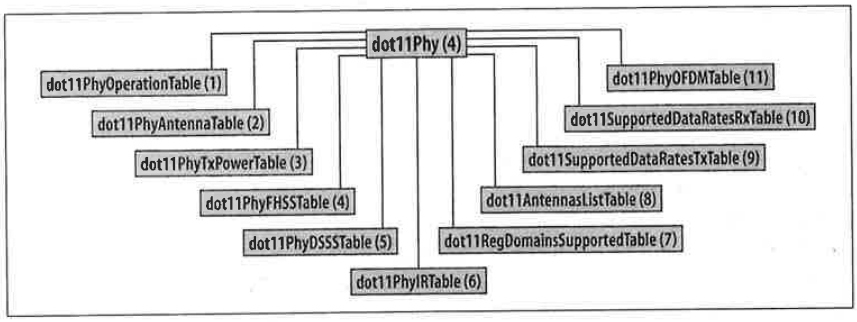


Figure A-7. Physical-layer MIB branches

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

Exhibit 1019
Page 396 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

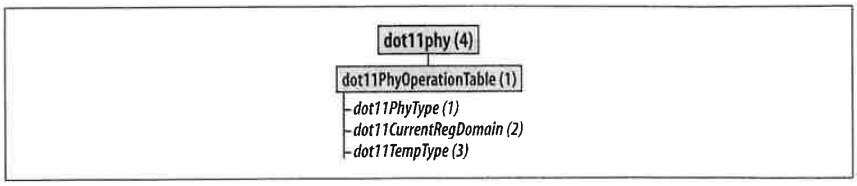


Figure A-8. Physical-layer operations table

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.

Exhibit 1019
Page 397 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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*.

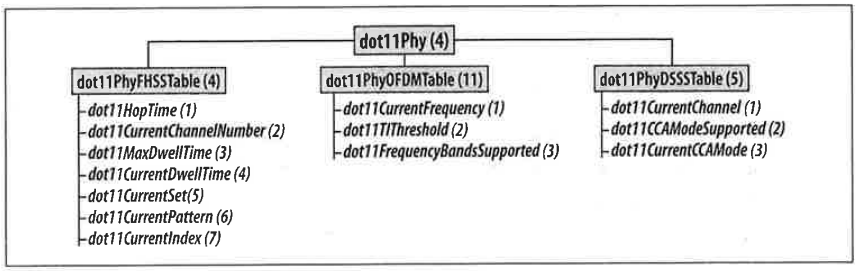


Figure A-9. Physical-layer attribute tables

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).

Exhibit 1019
Page 398 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 399 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 400 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 401 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

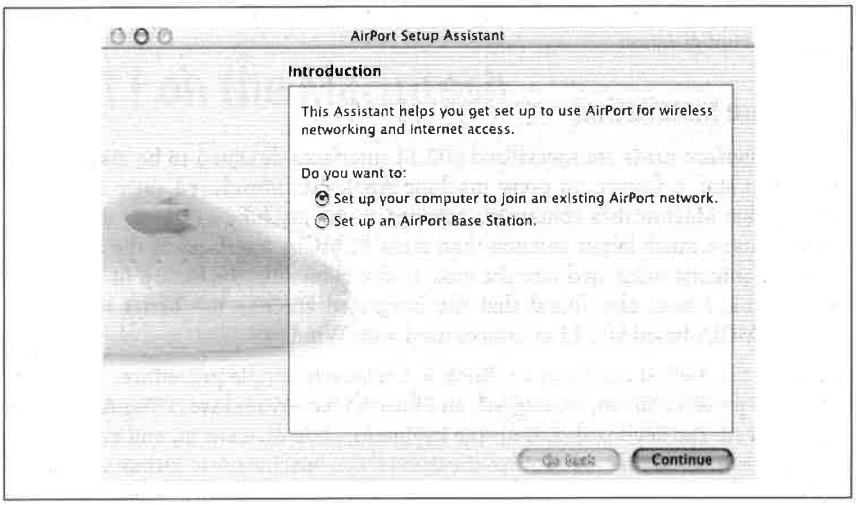


Figure B-1. Initial AirPort Setup Assistant screen

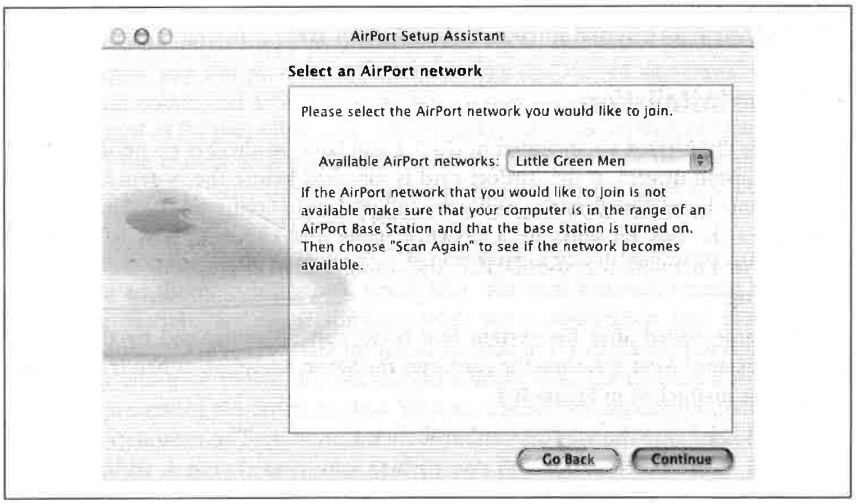


Figure B-2. AirPort card network selection

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

Exhibit 1019
Page 402 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Exhibit 1019
Page 403 of 451
Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

(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.

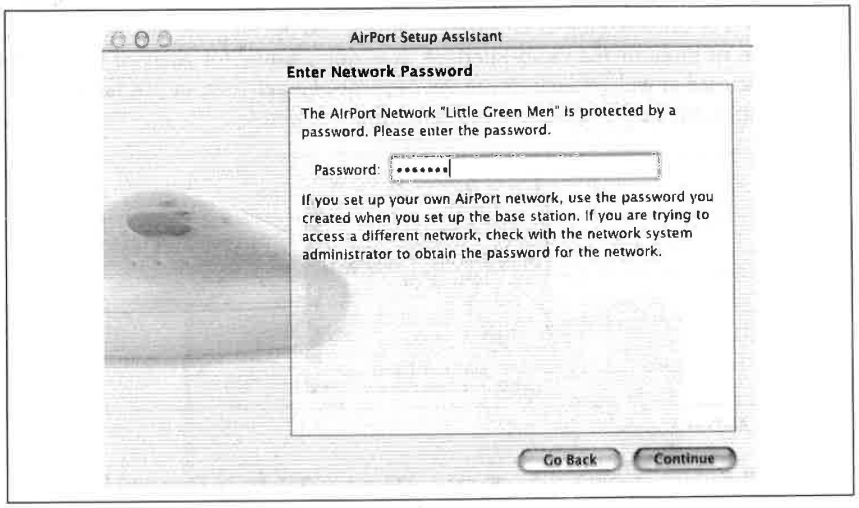


Figure B-3. AirPort network password entry

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.

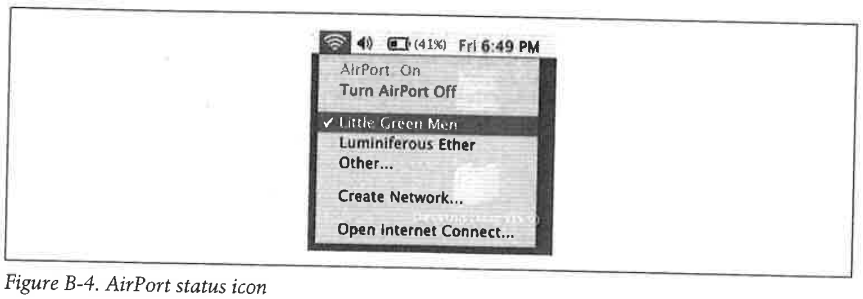


Figure B-4. AirPort status icon

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

Exhibit 1019
Page 404 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

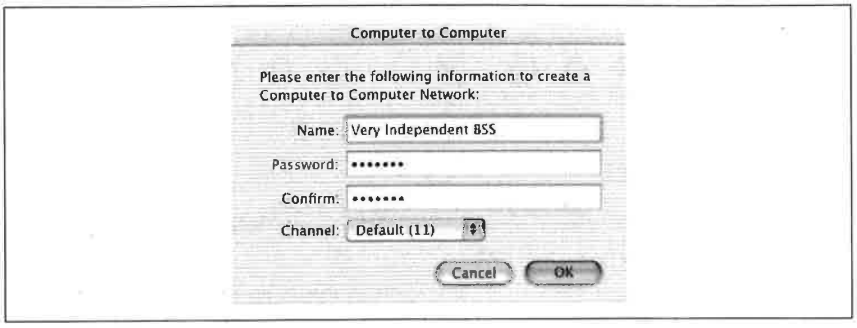


Figure B-5. IBSS parameter setup

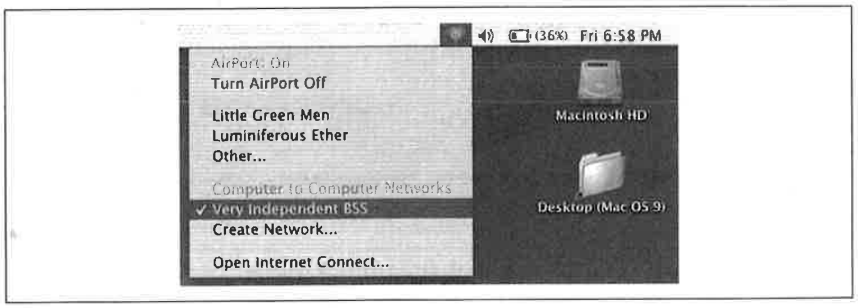


Figure B-6. Network preferences panel of the System Preferences application

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.

Exhibit 1019
Page 405 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

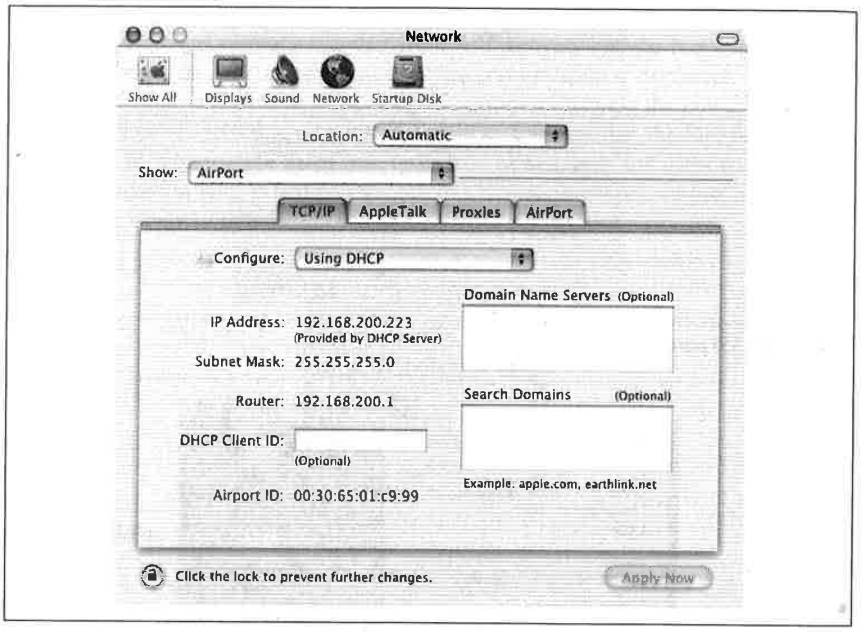


Figure B-7. TCP/IP preferences tab of the Network Preferences settings

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.

Exhibit 1019
Page 406 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

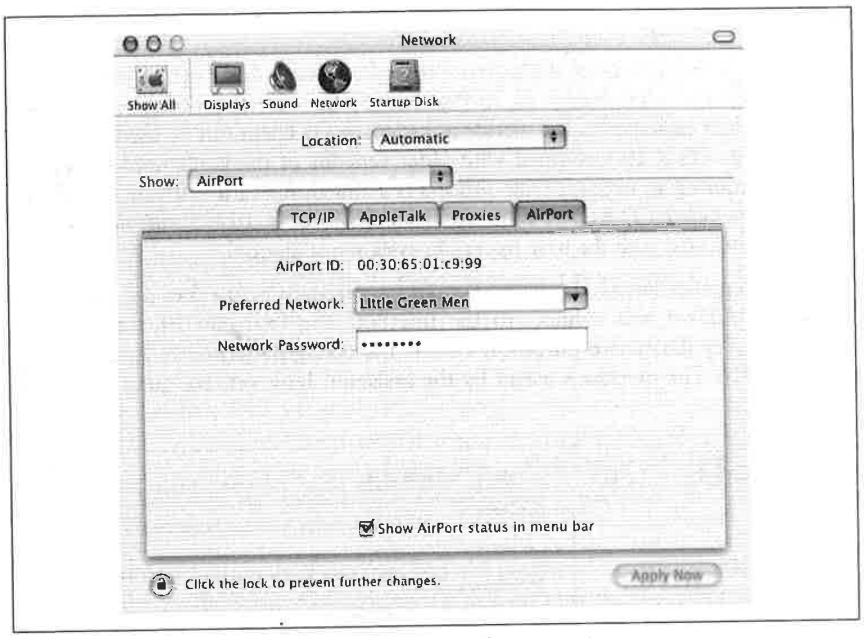


Figure B-8. AirPort preferences tab of the Network Preferences settings

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.

Exhibit 1019
Page 407 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

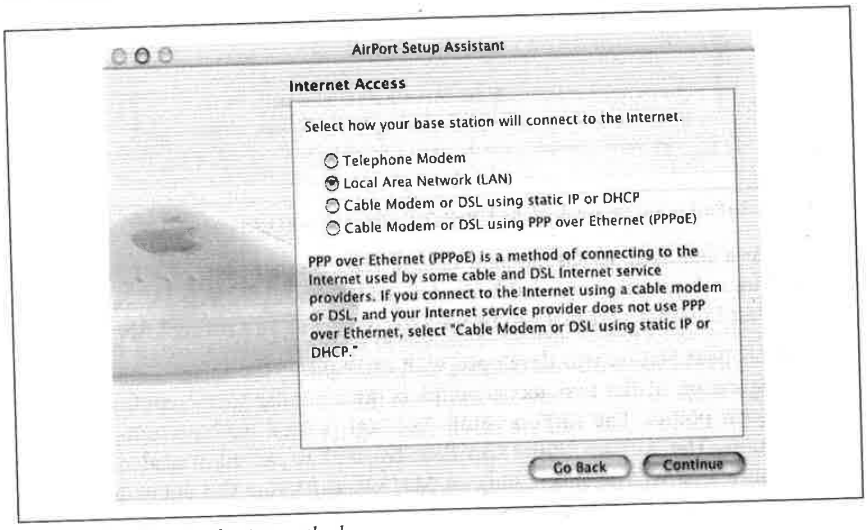


Figure B-9. Network selection methods

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.

Exhibit 1019
Page 408 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

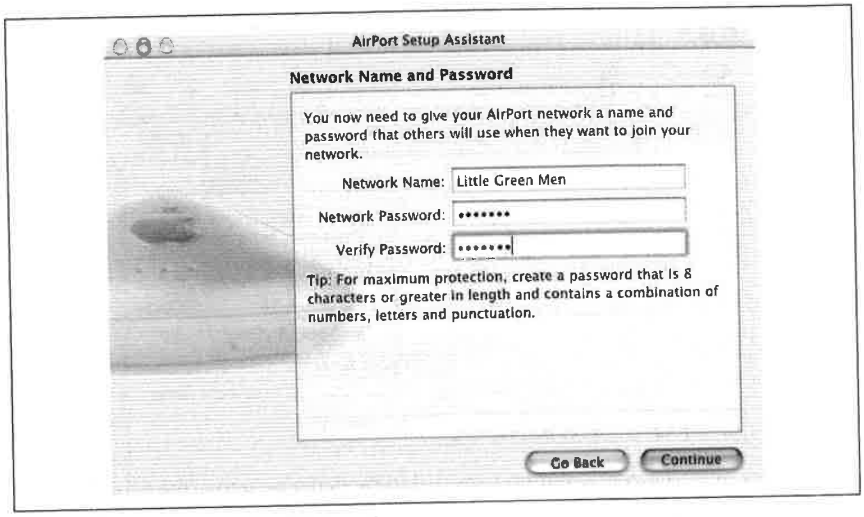


Figure B-10. Network name and password

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

Exhibit 1019
Page 409 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

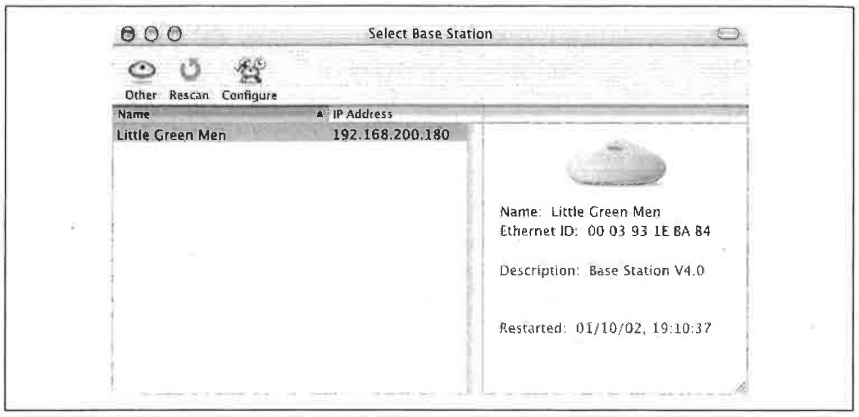


Figure B-11. AirPort Admin Utility main screen

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.

Exhibit 1019
Page 410 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

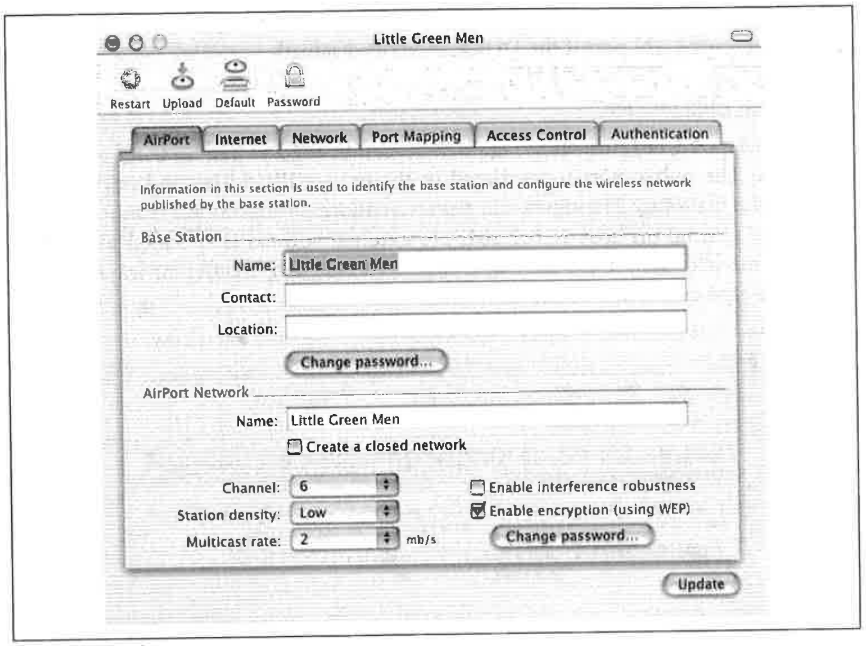


Figure B-12. Wireless interface configuration



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

Exhibit 1019
Page 411 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

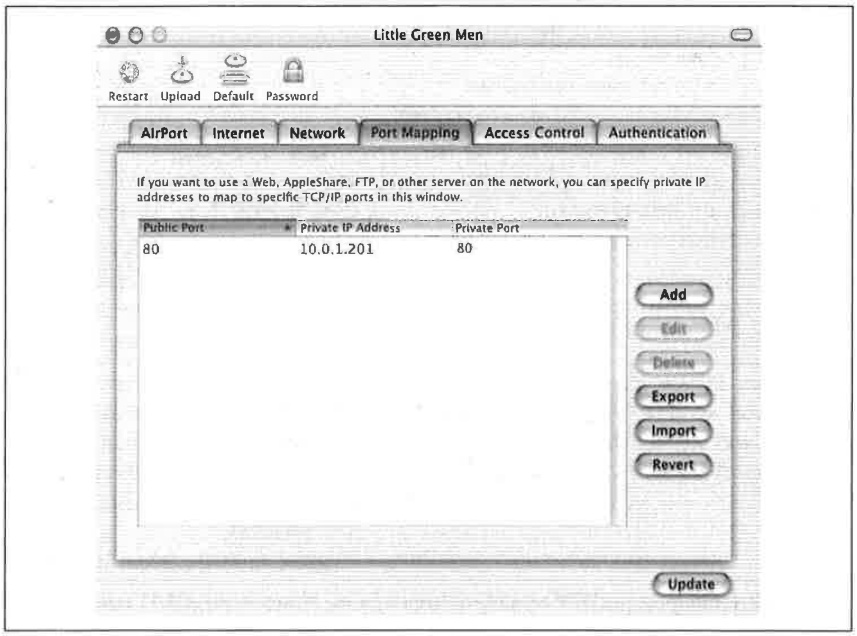


Figure B-14. Port Mapping tab

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,

Exhibit 1019
Page 412 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

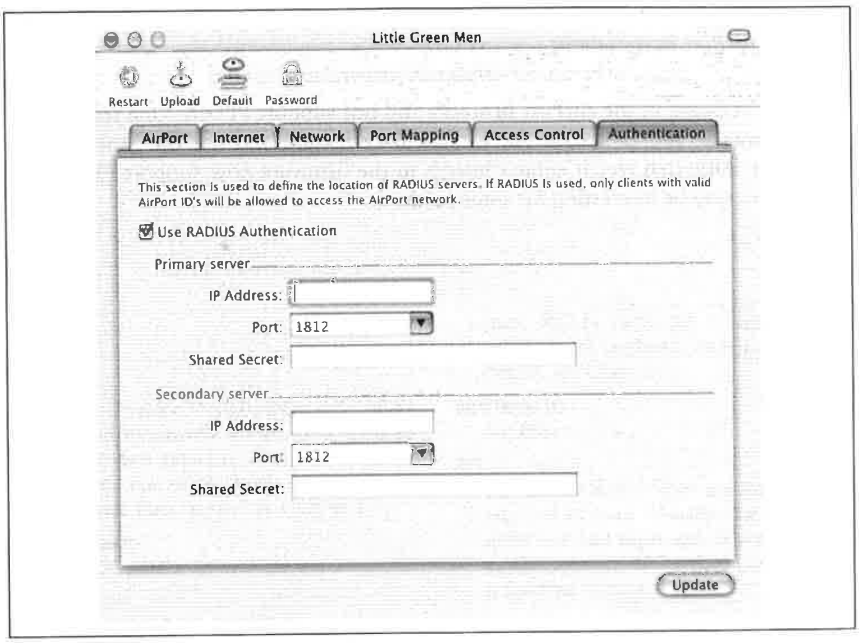


Figure B-15. RADIUS authentication tab

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.

Exhibit 1019
Page 413 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 414 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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).

Exhibit 1019
Page 415 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Exhibit 1019
Page 416 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Exhibit 1019
Page 417 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Exhibit 1019
Page 418 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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,

Exhibit 1019
Page 419 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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

Exhibit 1019
Page 420 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

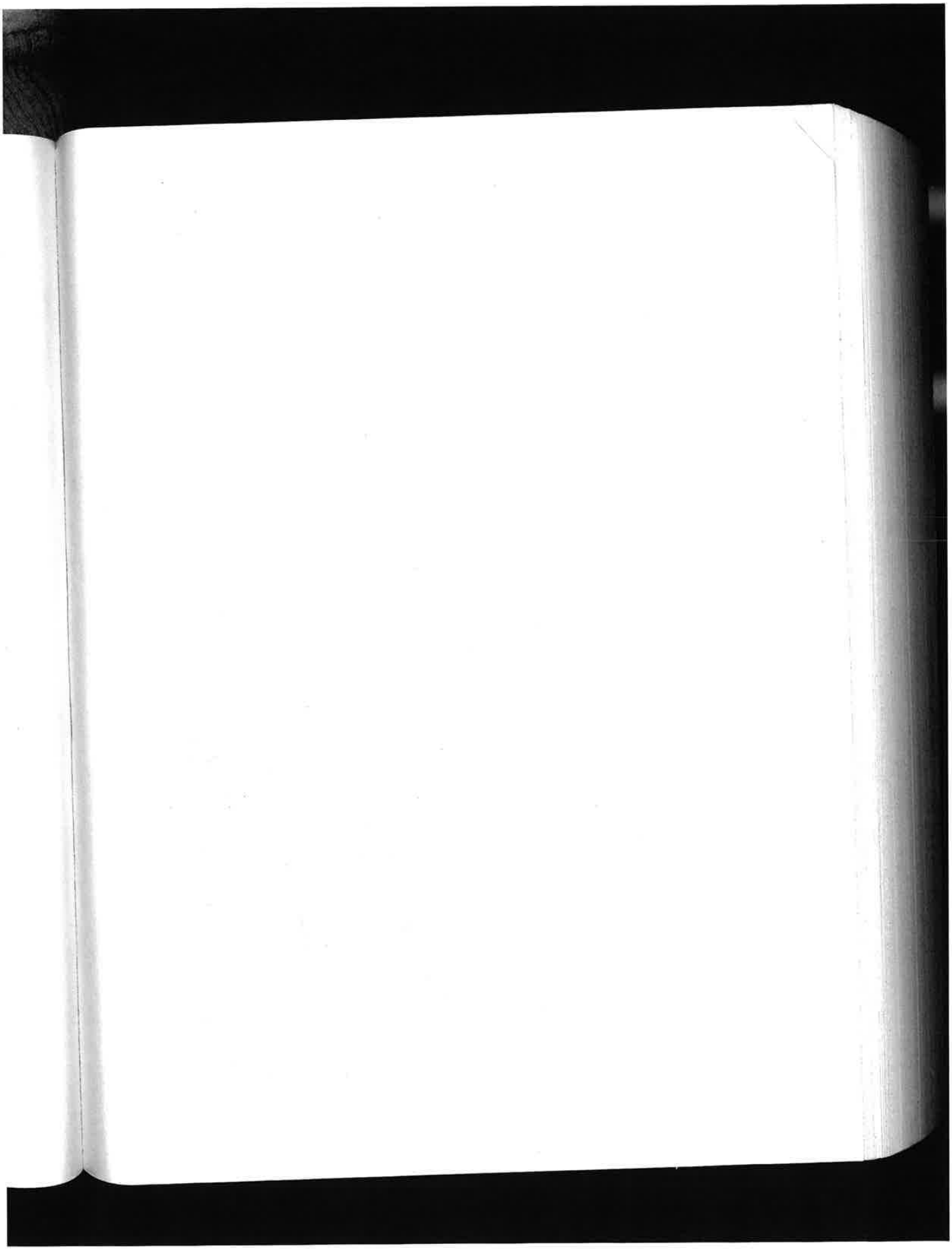
buffered unicast frames for each associated station as well as the presence of buffered multicast frames.

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.

Exhibit 1019
Page 421 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148



Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Index

Numbers

2GFSK (2-level GFSK), 170
3Com
  AirConnect, 266
  Home Wireless Gateway, 265
  site survey tools, 315
4GFSK (4-level GFSK), 171
802 family, 8–9
802.1 standard, 8
802.1d standard, 8
802.1h standard, 43
802.1q standard, 8
802.1x framework, vendor support for security, 327
802.11 standard
  Closed Wireless System extension, 276
  command-line parameters, 281
  comparisons, 6
  Ethereal and, 332
  features of, 8
  future for, 376–382
  Hierarchy of Network Development, 83
  Linux support for, 236
  MAC as key to, 24
  management architecture components, 114
  mobile network access and, 9
  open-system authentication, 120
  optional features, 267
  physical layers, 164
  RF and, 158–163
  station services and, 20
  success of, 6
802.11a standard
  features of, 6
  future of, 198
  OFDM and, 8, 199–208
  OFDM PHY, 152
  upgrade path from, 311
802.11b standard
  features of, 6
  HR/DS, 8
  HR/DS PHY, 152, 189–197
  Lucent ORiNOCO card, 229–234
  Nokia C110/C111 card, 215–228
  upgrade path from, 311
  Wi-Fi certification, 266
802.11g standard, 311
802.2 standard, 8
802.3 standard, 8
802.3af standard, 266
802.5 standard, 8

A

AAA (authentication, authorization, and accounting services), 299
Absolute Value Systems, 244
Abstract Syntax Notation 1 (ASN.1), 383
access control
  AirPort Base Station, 408
  authentication, 299
  project planning, 309
  security considerations and, 380
access deferral, 31
access modes, 8, 27–31
access point backbone, 295

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

Exhibit 1019
Page 423 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

access points
802.11 network component, 10
aging function, 131
AID and, 82
association and, 18
authentication and, 120
as bridges, 13, 14
BSS transitions and, 21
BSSIDs and, 67
buffering frames, 48
contention-free service and, 150
data frames and, 55
density of, 377
DTIM interval in, 132
frames from, 58
frames to, 59
general functions, 262–264
infrastructure BSSs and, 11
infrastructure networks and, 372
installing in wiring closets, 322
linux-wlan-ng problems, 253
Listen Interval field and, 82
naming conventions for, 326–327
network analysis, 349–350
Nokia A032 access point, 279–291
ORiNOCO (Lucent) AP-1000, 269–278
point coordinators and, 27
power management and, 38, 128, 268
reassociation, 18, 126
selecting, 266–269
site survey and, 315
station association, 12, 15
transmissions from, 142–143
types, 264–266
WEP design flaws, 95
Acknowledgment (ACK)
802.11 transmission rules and, 31
ATIM window and, 135
components of, 63
dot11ACKFailureCount, 391
dot11FrameDuplicateCount, 391
fragmentation and, 35, 47
frame buffering and, 129
integrity checks and, 42
Probe Response frames and, 353
unicast frames and, 44, 45
web transaction example, 360
active mode, 128, 131
active scan timer, 373
active scanning
features of, 116, 351
Probe Request frames and, 115
ad hoc BSSs (see IBSS)
ad hoc networks (see IBSS)
address fields
802.11 MAC and, 35
in data frames, 53
management frames and, 67
specifics, 40–41
address filtering
AP-1000 and, 277
NIDS and, 288
security considerations, 121
Address Resolution Protocol (see ARP)
addresses
assignment through DHCP, 299
group addresses table, 392
I/O addresses, 243, 254
multicast addresses, 40
private addresses, 299
translation considerations, 310
transmitter address, 41, 63, 66
(see also MAC addresses; receiver address)
addressing
DS bits and, 53–56
dynamic addressing, 299
static addressing, 299
Agere (see Lucent)
AID (association ID)
access points and, 82
defined, 40
frame buffering and, 129, 132
management frames and, 71
PS-Poll frame and, 48, 65
AirConnect (3Com), 266
Aironet (Cisco), 238, 266, 299
AiroPeek (WildPackets), 332
AirPort Base Station, 402–409
AirPort Card
features, 396–401
residential gateway, 265
AirSnort
network analysis, 363–367
WEP key recovery program, 96
algorithms
authentication algorithms table, 387
Challenge Text field and, 83

Exhibit 1019
Page 424 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Status Code field and, 83
whitening, 174
amplifiers
features of, 160
as power-hungry components, 48
ramp-up and ramp-down of, 174
transmitting power and, 321
amplitude
channels and, 204
interference and, 182
Annex D (see MIB)
announcement traffic indication map (see ATIM)
antenna diversity, 321, 374
antennas
AirPort Base Station, 402
diversity support, 174
external antennas, 266, 267, 279
installing, 322
Nokia C110/C111 card and, 215
omnidirectional antennas, 159, 161, 316
RF components, 158–160
specifications for, 316–322
standardization of, x
wireless network infrastructure, 2
Antheil, George, 156
AP Manager (see ORiNOCO AP Manager)
AP (see access points)
AP-1000 (see Orinoco AP-1000)
Apple Computer
802.11 equipment and, 396
Airport Card, 265
MacIntosh, 409–410
application layer, security at, 299
applications
fixed wireless, 3
performance characterization, 324
project planning, 308
site survey, 315
ARP (Address Resolution Protocol), 43, 358–359
ASN.1 notation, 383
association
802.11 networks, 124–127
access points and stations, 12
authentication and, 18, 124, 299
Class 2 frames and, 85
Class 3 frames and, 85
Current AP Address field and, 71
dot11AssociationResponseTimeOut, 386
Ethereal example, 355
IAPP and, 15
joining and, 119
linux-wlan-ng problems, 253
restricting for AP-1000, 277
restricting for Nokia A032, 288
setting for wvlan_cs driver, 257
(see also reassociation)
association ID (see AID)
Association Request frames
association procedure, 125
features of, 82
Association Response frames, 82
association states, 83–85
association timeout, 373
Atheros Communications chipset vendor, 198
ATIM (announcement traffic indication map), 78, 81
ATIM window
frames allowed during, 135
settings for, 134
as time window, 133
tunable parameter, 372
atomic operations
contention-based data service and, 44
defined, 25
DIFS and SIFS, 30
NAV and, 28
authentication
802.11 networks, 120–124
access control and, 299
access points and, 83
AirPort Base Station, 408
AP-1000 and, 277
data security and, 89
dot11AuthenticateFailStation, 387
dot11AuthenticateFailStatus, 387
dot11AuthenticationResponseTimeout, 385
Ethereal example, 354–355
frame transmission and, 83–85
Kerberos authentication, 299
linux-wlan-ng problems, 253
network service component, 18
project planning, 309
WEP limitations, 299
Authentication Algorithm Identification, 121
Authentication Algorithm Number field, 68, 83
authentication algorithms, 387
authentication, authorization, and accounting services (see AAA)
authentication states, 83–85

Exhibit 1019
Page 425 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

authentication timeout, 373
Authentication Transaction Sequence Number field, 69, 121
availability
  data security attribute, 299
  through redundancy, 299

B

backbone networks
  distribution system as, 10
  ESS and, 12
  limitations in choosing access points, 14
backoff timers, 138
backoff window, 33
bandwidth
  802.11a standard, 206
  carriers and, 199
  direct-sequence modulation and, 178
  FCC rules for frequency-hopping systems, 168
  frequencies and, 3
  limitations with wireless LANs, 3
  network speed and, 5
Barker sequence, 179
Barker words, 178, 189
basic rate set, 119
basic service area
  active scanning and, 117
  BSS and, 10
  no transition and, 20
  overlapping, 16
  reassociation and, 18
  SSIDs and, 75
  TSF and, 137
basic service set (see BSS)
basic service set ID (see BSSID)
batteries
  802.11 standard and, 268
  high-frequency devices and, 181
  maximizing life of, 133
  mobile devices and, 128
Beacon frames
  ATIM window and, 135
  BSS and, 369
  case study example, 349–350
  CF Parameter Set elements and, 149
  CFP and, 40, 141
  dot11CFPPeriod, 385
  excluding from Ethereal analysis, 345
  features of, 79
  as management frames, 45
  passive scanning and, 115, 116, 351
  purpose of, 67
  sleeping periods and, 48
  TBTT and, 138
  TIM and, 49, 129, 132
  timestamps, 168
  timing synchronization and, 137
Beacon interval
  frame buffering and, 130
  Listen intervals and, 371
  management frames and, 69
  tunable parameter, 369
Beacon period
  buffer management and, 137
  dot11BeaconPeriod, 386
  dot11DTIMPeriod, 386
  power management and, 128
Beacon Period scanning parameter, 118
BER (bit error rate), 204
Berra, Yogi, 376
binary phase shift keying (see BPSK)
bison, Ethereal and, 333
bit error rate (BER), 204
Bitmap Control field (TIM), 78
Bitmap Offset field (TIM), 78
bits
  defined, 177
  DS bits, 53–56
  order of, 36
  padding, 205
Bluetooth standard, 5, 378
Bohr, Niels, 86
BPSK (binary phase shift keying), 211
bridges
  access points as, 13, 14, 262
  wireless features, 15
bridging
  802.1d specification, 8
  access points and, 10
  example, 14
  Nokia A032 and, 280, 282
broadcast BSSID, 55
broadcast frames
  ATIM window and, 135
  contention-free service and, 150
  defined, 40
  DTIM and, 132
  frame exchanges and, 44
  mobile stations and, 133
broadcast SSID, 75, 117
Bruestle, Jeremy, 96

Exhibit 1019
Page 426 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

BSS (basic service set)
Beacon frames and, 369
BSSID in, 55
dot11BeaconPeriod, 386
dot11DesiredBSSType, 386
dot11FCSErrorCount, 391
dot11MaxDuration, 385
DTIM Period, 132
extended service sets and, 12
infrastructure BSS, 11, 55
passive scanning and, 116
stations joining, 119
BSSBasic Rate Set scanning parameter, 119
BSSID (basic service set ID)
access points as, 67
address fields in data frames, 54
defined, 55
IBSS frames and, 58
PS-Poll frame, 66
purpose of, 41
BSSID field, 147, 148
BSSID scanning parameter, 115
BSSType scanning parameter, 115
buffer memory, 131, 136
buffering frames (see frame buffering)

C

C++ compiler, AirSnort and, 364
cabling
antennas and, 320
fiber-optic cable, 298
installation constraints, ix
Capability Information field, 69, 82
capture program (AirSnort), 364, 365
card information structure (see CIS)
cardmgr process (PCMCIA Card Services), 238
carrier sense multiple access (see CSMA)
carrier sense multiple access with collision avoidance (see CSMA/CA)
carrier sense multiple access with collision detection(see CSMA/CD)
Carrier Sense/Clear Channel Assessment (see CS/CCA)
carrier-sensing functions, 28, 31
CCA (clear channel assessment)
dot11CCAModeSupported, 394
dot11CurrentCCAMode, 394
OFDM PHY, 212
physical layer and, 151
CCK (complementary code keying), 189, 193–195
CDMA (code division multiple access), 179, 199
CF Parameter Set information element
Beacon frames and, 79
contention-free service, 149
management frame information element, 78
scanning and, 119
CF-ACK frames, 143, 144, 145, 146
CF-ACK+CF-Poll frame, 143, 147
CF-End frames, 143, 147
CF-End+CF-ACK frame, 143, 148
CFP Count field, 149
CFP DurRemaining field, 149
CFP MaxDuration field, 149
CFP Period field, 149
CFP (contention-free period), 40, 141, 146
CFPMaxDuration, 141
CF-Poll frames
access points and, 58
contention-free period and, 145, 146
contention-free service and, 143
dot11CFPollable, 385
mobile stations and, 59
PCF operation and, 142
Challenge Text information element
algorithms and, 83
features of, 79
shared-key authentication and, 121
Channel Agility field, 70
channel agility option, 196
ChannelList scanning parameter, 115
channels
802.11 frequency hopping and, 166
802.11a standard, 206
amplitude and, 204
dot11CurrentChannel, 394
dot11CurrentChannelNumber, 393
dot11CurrentIndex, 394
dot11MaxDwellTime, 394
DS PHY, 179
energy spread, 180
hopping order, 168
HR/DS systems, 196
OFDM and, 199
operating channels, 206–208
overlapping coverage, 323, 324
PHY parameters, 119
regulatory domains, 167, 180
specifying, 115

Exhibit 1019
Page 427 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

chip, 177
chipsets
  interface cards, 237
  MAC controllers and, 238
  PRISM chipset, 198, 237, 244, 271
  vendors listed, 198
CIS (card information structure)
  linux-wlan-ng problems, 253
  PCMCIA Card Services, 238
  purpose of, 240
Cisco
  access control solution, 299
  Aironet 350 Series Access Point, 266
  Aironet chipset, 238, 299
  Radiata acquisition, 198
Class 1 frames, 84
Class 2 frames, 85
Class 3 frames, 85
classes (frames), 84
clear channel assessment (see CCA)
Clear to Send (see CTS)
Closed Wireless System, 276
clusters, 299, 307
code division multiple access (see CDMA)
coded OFDM (COFDM), 204
collision avoidance
  802.11 MAC and, 24
  interframe spacing and, 29
  random backoff, 27
collisions
  hidden nodes and, 26
  random backoffs and, 27
  troubleshooting Ethernet interface, 275
communications
  distribution system criticality in, 14
  infrastructure BSSs and, 11
  stations within ESSs, 13
comparison operators, 343
compilation
  AirSnort, 364
  Ethereal, 333–336
  kernel, 245
  libpcap library, 333
  linux-wlan-ng, 245–247, 252
  orinoco_cs driver, 260
  wvlan_cs driver, 255
complementary code keying (see CCK)
confidentiality
  data security and, 89
  ICV and, 90
  network deployment and, 299
  project planning, 309
  WEP limitations, 307
config command, 281
configuration
  AirPort interface, 399–401
  AP-1000, 271
  DHCP, 249, 284
  kernel, 245
  linux-wlan-ng, 245, 249–252
  Nokia A032, 283–289
  Nokia driver, 222, 224
  orinoco_cs driver, 260
  set command, 279
  WEP, 251, 258
  wvlan_cs driver, 256, 258
  (see also PCMCIA Card Services)
connectors
  AirPort cards and, 397
  antennas and, 321
contention
  data services based on, 44–50
  frame transmission and, 40
  PCF and, 27
contention-free period (see CFP)
contention window, 33, 34
contention-based service
  data and, 44–50
  data frames and, 52
contention-free polling bits, 70
contention-free service, 52, 140–150
Control field, 43, 347
Control frames
  Class 1 frames and, 84
  features of, 60–66
  matching for Ethereal analysis, 343
convolution code (OFDM), 205, 210
correlator, 177
coverage
  considerations, 322
  importance of, 3
  overlapping, 323, 324
  physical restrictions and, 311
  project planning, 308
  rule-of-thumb radius, 314
  site survey, 315
CRACK (Challenge/Response for Authenticated Control Keys), 304
crack program (AirSnort), 364, 365
CRC field, 186, 193

Exhibit 1019
Page 428 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

CRC (cyclic redundancy check)
FCS as, 42
WEP and, 92, 94
cryptography
confidentiality and, 299
decryption dictionaries, 94
dot11WEPUndecryptableCount, 391
key mapping relationships, 90
problems with WEP, 93–96
"Snake Oil Warning Signs: Encryption Software to Avoid", 305
WEP and, 39, 89–93
(see also encoding)
CS/CCA (carrier sense/clear channel assessment)
802.11 FH PHY, 175
DS PHY, 187
HR/DS, 196
CSMA (carrier sense multiple access), 24
CSMA/CA (carrier sense multiple access with collision avoidance), 24
CSMA/CD (carrier sense multiple access with collision detection), 8
CTS frames, 135, 391
CTS (Clear to Send)
components of, 63
interframe spacing and, 31
NAV and, 28
preventing collisions with, 26
virtual CTS, 64
(see also RTS/CTS exchange)
Current AP Address field, 71
cyclic extensions, 203–204
cyclic prefixes, 203–204
cyclic redundancy check (see CRC)

D

data
contention-based service, 44–50
Ethereal and, 338, 339
key recovery time estimates, 366
reducing amount captured, 340–341
WEP and, 90–92
wireless network access advantages, ix
Data field
OFDM PLCP, 211
specifics, 42
data frames
contention-free period, 144, 145
contention-free service and, 142
dot11ReceivedFragmentCount, 391
features of, 51–60
web transaction example, 362
data rates
configuring AP-1000, 273
dot11OperationalRateSet, 386
OFDM PHY, 211
setting for wvlan_cs driver, 257
Task Group G, 377
data transfer, 347, 362
Databook TCIC2 controller, 241
Data+CF-ACK frame, 143, 144
Data+CF-ACK+CF-Poll frame, 143, 145
Data+CF-Poll frame, 143, 145
dBm (HPA), 160
DBPSK (differential binary phase shift keying), 182–183, 187
DCF (distributed coordination function)
active scanning example, 117
ATIM window and, 135
contention-based access, 31–34
contention-based service, 44
Duration field, 67
Ethereal example, 354
throughput using, 311
wireless medium access, 27
DCF interframe space (see DIFS)
deauthentication, 18
Deauthentication frames
association procedure, 125
Class 2 frames and, 85
dot11DeauthenticateReason, 386
dot11DeauthenticateStation, 387
features of, 81
reason codes and, 72
reassociation procedure, 127
debugging
linux-wlan-ng, 251–254
ping and traceroute, 264
(see also error handling; troubleshooting)
decibels (dB), 160
decryption (see cryptography)
deep fading, 204
default keys (WEP), 90
deferred response, 49
delay (802.11a standard), 206
delay spread, 162
Delivery TIM (see DTIM)
deployment (see network deployment)
destination address, 40, 54
destination service access point (see DSAP)

Exhibit 1019
Page 429 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

device drivers
experimenting with, 23
Lucent ORiNOCO card, 229–234
Nokia C110/C111 card, 215–228
open source for Linux, 255
variability in features, 114
device management
802.11 standard and, 268
equipment purchases, 309
Nokia A032 and, 282
DFT (discrete Fourier transform), 202
DHCP (Dynamic Host Configuration Protocol)
address assignments through, 299
AP-1000 and, 270, 271, 272
configuring, 249, 284
popular service, 263
redundancy for, 299
standardized failover protocol, 299
differential binary phase shift keying (see DBPSK)
differential phase shift keying (see DPSK)
differential quadrature phase shift keying (see DQPSK)
DIFS (distributed interframe space), 33, 45
digital signal processors (DSPs), 202
dipole antennas, 317
direct sequence (DS)
802.11 standard and, 6
channel layout considerations, 322–324
dot11CCAModeSupported, 394
spread spectrum type, 156
direct-sequence PHY (see DS PHY)
direct-sequence PLCP (see DS PLCP)
direct-sequence PMD (see DS PMD)
direct-sequence spread spectrum (see DSSS)
disassociation, 18
Disassociation frames
dot11DisassociateReason, 386
dot11DisassociateStation, 387
features of, 81
reason codes and, 72
discrete Fourier transform (see DFT)
discrete Multi-Tone (see DMT)
distortion, noise as, 152
distributed coordination function (see DCF)
distributed interframe space (see DIFS)
distribution (network service component), 18
distribution system
802.11 networks, 10, 20
BSS transitions and, 21
considerations, 14–16
FromDS bit, 59
integration and, 18
ToDS and FromDS bits, 38
D-Link (DWL-1000AP), 265
DMT (discrete Multi-Tone), 156
DNS request/reply, 359, 360
dot11ACKFailureCount, 391
dot11Address, 392
dot11AssociationResponseTimeOut, 386
dot11AuthenticateFailStation, 387
dot11AuthenticateFailStatus, 387
dot11AuthenticationAlgorithm, 387
dot11AuthenticationAlgorithmsEnable, 387
dot11AuthenticationResponseTimeout, 385
dot11BeaconPeriod, 386
dot11CCAModeSupported, 394
dot11CFPollable, 385
dot11CFPPeriod, 385
dot11CountersTable, 390
dot11CurrentCCAMode, 394
dot11CurrentChannel, 394
dot11CurrentChannelNumber, 393
dot11CurrentFrequency, 395
dot11CurrentIndex, 394
dot11CurrentPattern, 394
dot11CurrentRegDomain, 393
dot11CurrentSet, 394
dot11DeauthenticateReason, 386
dot11DeauthenticateStation, 387
dot11DesiredBSSType, 386
dot11DesiredSSID, 386
dot11DisassociateReason, 386
dot11DisassociateStation, 387
dot11DTIMPeriod, 386
dot11FailedCount, 391
dot11FCSErrorCount, 391
dot11FragmentationThreshold, 390
dot11FrameDuplicateCount, 391
dot11FrequencyBandsSupported, 395
dot11HopTime, 393
dot11LongRetryLimit, 390
dot11mac objects, 383
dot11MACAddress, 389
dot11MaxDuration, 385
dot11MaxDwellTime, 394

Exhibit 1019
Page 430 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

dot11MaxReceiveLifetime, 390
dot11MaxTransmitMSDULifetime, 390
dot11MediumOccupancyLimit, 384
dot11MulticastReceivedFrameCount, 391
dot11MulticastTransmittedFrameCount, 390
dot11MultipleRetryCount, 391
dot11OperationalRateSet, 386
dot11OperationsTable, 389
dot11phy objects, 383
dot11PHYType, 392
dot11PowerManagementMode, 386
dot11PrivacyOptionImplemented, 386
dot11ReceivedFragmentCount, 391
dot11req_ command, 248
dot11req_mibset command, 248
dot11req_scan command, 248
dot11res objects, 383
dot11RetryCount, 391
dot11RTSFailureCount, 391
dot11RTSSuccessCount, 391
dot11RTSThreshold, 390
dot11ShortRetryLimit, 390
dot11StationID, 384
dot11TempType, 393
dot11TIThreshold, 395
dot11TransmittedFragmentCount, 390
dot11TransmittedFrameCount, 391
dot11WEPDefaultKeyValue, 388
dot11WEPKeyMappingAddress, 388
dot11WEPKeyMappingValue, 389
dot11WEPKeyMappingWEPOn, 389
dot11WEPUndecryptableCount, 391
dozing, power conservation and, 128
DPSK (differential phase shift keying), 182–185
DQPSK (differential quadrature phase shift keying), 183, 187, 189, 193
DSAP (destination service access point), 43, 347
DS bits, 53–56
DS Parameter Set information element, 77, 79
DS PHY
  direct-sequence channel layout, 322
  dot11PHYType, 392
  features of, 176–189
  physical layer, 152
DS PLCP, 185–186
DS PMD, 187–188
DS (see direct sequence)
DSSS (direct-sequence spread spectrum)
  802.11 specification and, 8
  contention window example, 33
  DSSS table, 394
  spread-spectrum type, 156
DTIM (Delivery TIM)
  dot11DTIMPeriod, 386
  frame delivery, 132
  purpose of, 372
DTIM Count field, 78
DTIM interval
  configuring AP-1000, 273
  contention free periods and, 149
  dot11CFPPeriod, 385
  tunable parameter, 369
DTIM Period scanning parameter
  basic service sets and, 132
  scan report, 118
  TIM and, 78
  tunable parameter, 372
dump_cis tool, 240
Duration field
  ACK frame and, 64
  CF-End frames and, 147
  CF-End+CF-ACK frame, 148
  CTS frame and, 63
  management frames and, 67
  RTS frame and, 62
  in data frames, 52–53
Duration/ID field, 39–40, 48
dwell time
  802.11 frequency-hopping systems, 167
  defined, 76, 165
  dot11MaxDwellTime, 394
  tunable parameter, 373
DWL-1000AP (D-Link), 265
Dynamic Host Configuration Protocol (see DHCP)

E

EAP (Extensible Authentication Protocol), 378
effective radiated power (see ERP)
EIFS (extended interframe space), 30
encapsulation
  802.2/LLC, 8
  frame header and, 36
  RFC 1042, 43, 244
  SNAP and 802.11, 347

Exhibit 1019
Page 431 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

encoding
802.11 direct-sequence networks and, 179
OFDM encoding, 212
via frequency, 169
encryption (see cryptography)
energy detection (ED) threshold, 187, 196
enterprise gateways, 265
equipment (see device management)
ERO (European Radiocommunications Office), 4, 153
ERP (effective radiated power), 156, 160
error handling
802.11 transmission rules and, 31
DCF and, 32
linux-wlan-ng problems, 253, 254
(see also debugging; troubleshooting)
ESS (extended service set)
Beacon frames and, 369
BSSs and, 12
ESS transitions and, 22
ESSID and, 257
information needs in, 15
joining, 119
Mobile IP and, 22
mobility within, 299
network addressing and, 313
ESS field, 70
ESSID, 257, 281
Ethereal network analyzers, 332–348
Ethernet
as backbone network, 10
as distribution system medium, 125, 14
Fast Ethernet connections, 298
Gigabit Ethernet connections, 298
reliability of, 24
similarity to 802.11, 7
WECA, 6, 266
wireless Ethernet, 6, 16
Ethernet interface
configuring AP-1000, 275
Linux drivers and, 236
linux-wlan-ng and, 252
ETSI (European Telecommunications Standards Institute), 153, 169
European Radiocommunications Office (see ERO)
European Telecommunications Standards Institute (see ETSI)
exclusive OR (see XOR)
Extended Authentication (see XAUTH)
extended interframe space (see EIFS)
extended service area
BSS transitions and, 21
characteristics of, 12–14
reassociation and, 18
SSID and, 75
extended service set (see ESS)
Extensible Authentication Protocol (see EAP)
external antennas
802.11standard and, 267
enterprise gateways and, 266
Nokia A032 and, 279

F

fading (see multipath fading)
Fast Ethernet connections, 298
fast Fourier transform (see FFT)
FCS (frame check sequence), 42, 57
FDM (frequency division multiplexing), 199
FDMA (frequency division multiple access), 165
Federal Communications Commission (FCC)
radio spectrum use and, 4
RF spectrum regulation, 153
rule enforcement, 322
rules regarding 802.11 frequency-hopping systems, 168
FFT (fast Fourier transform), 202
FFT integration time, 201
FH (see frequency hopping)
FH Parameter Set information element
802.11 frequency-hopping networks, 168
Beacon frames and, 79
fields in, 76
FH PHY
dot11PHYType, 392
dwell time and, 373
features of, 164–176
whitening and, 185
FH PLCP, 171–174
FH PMD, 174, 393
FHSS (frequency-hopping spread spectrum)
802.11 specification and, 8
FHSS table, 393
physical layer, 152
spread-spectrum type, 156
fiber-optic cable, 298
fields
address fields, 40–41
data field, 42

Exhibit 1019
Page 432 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Duration/ID field, 39–40
Frame Control subfield, 36–39
management frames, 68–73
Sequence Control field, 41
filtering
access points and, 59
Beacon frames, 345
BSSID and, 55
case study example, 349–350
data capture and, 340
displays, 340
management connections, 276, 287
network analyzers and, 332
packet filtering, 299
rejections and, 85
(see also address filtering)
Finlayson, Ross, 271
firewalls, 299
flex lexical scanner, 333
flexibility (wireless network advantage), ix, 2–3
Fluhrer, Scott, 95
foreign network, 299
foreshortening, contention-free period and, 144
Fourier analysis, 202
Fourier transform, 202, 204
fragmentation
802.11 MAC, 34–35
frames for group addressees, 45
at network layer, 46
preventing, 42
PS-Poll frame example, 49
RTS/CTS exchange and, 48
fragmentation threshold
changing, 47
dot11FragmentationThreshold, 390
parameter, 47, 282
setting, 48
tuning, 258, 370
fragments
802.11 transmission rules, 32
dot11MaxReceiveLifetime, 390
dot11ReceivedFragmentCount, 391
dot11TransmittedFragmentCount, 390
frame sequence numbers in, 35
lifetime of, 33
More Fragments bit, 38
Sequence Control field and, 42
virtual RTS and, 53
frame body (see Data field)
frame buffering
AID and, 129
association procedure and, 125
ATIM window and, 134
Beacon frames and, 50
contention and, 150
memory and, 136
power management and, 128
PS-Poll frames, 129
reassociation procedure, 127
TIM and, 372
frame check sequence (see FCS)
Frame Control field
ACK frame and, 64
CF-End frames and, 147
CF-End+CF-ACK frame, 148
Control frames and, 60
CTS frame and, 63
Data frames and, 52
Ethereal, 341
More Fragments bit, 47, 53
null frames and, 57
PS-Poll frame, 65
RTS frame and, 62
specifics, 36–39
frame sequence numbers, 35
frames
802.11 MAC format, 35–42
access points and, 58, 59
allowed during ATIM window, 135
broadcast and multicast data, 44
broadcast BSSID and, 55
BSS transitions and, 21
contention-free period and, 144–149
customizing lengths, 33
distribution system, 14
duplicate frames, 391
Integrity Check Value and, 90
manipulation on 802.11 networks, 10
NAV and, 28
processing by distribution systems, 14
retry counters and, 32
retry limits and, 371
RTS/CTS exchange, 27, 47
transmission and authentication, 83–85
transmission during contention-free periods, 40
WEP and, 60, 91
framing
DS PLCP, 185
HR/DS PLCP, 190–193

Exhibit 1019
Page 433 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

frames (*continued*)
OFDM PLCP, 208–211
PLCP and, 172–174
frequency
antenna sizes and, 158
dot11CurrentFrequency, 395
dot11FrequencyBandsSupported, 395
dwell time and, 165
encoding data via, 169
GFSK and, 170
listed for common bands, 4
wireless devices and, 3
wvlan_cs driver setting, 257
frequency allocation, 153–155
frequency division multiple access (see FDMA)
frequency division multiplexing (see FDM)
frequency hopping (FH)
802.11 standard and, 6
dot11CurrentChannelNumber, 393
spread-spectrum type, 156
timer synchronization and, 119, 137
transmission, 165–169
frequency-hopping PHY (see FH PHY)
frequency-hopping PLCP (see PH PLCP)
frequency-hopping PMD (see FH PMD)
frequency-hopping spread spectrum (see FHSS)
FromDS bit
ARP reply and, 359
ARP requests and, 357
control frames and, 61
distribution system, 59
DNS reply and, 360
specifics, 38

G

gain
amplifiers and, 160
defined, 316
regulatory requirements, 179
gateways
access point types, 264–266
residential gateways, 264, 269
Gaussian frequency shift keying (see GFSK)
GFSK (Gaussian frequency shift keying), 169–171
Gigabit Ethernet connections, 298
GNU Public License, 332
GTK+ library, 333
guard bands, 153, 200
guard time, 201–203, 206

H

half-duplex mode, 26
half-power beam width, 316
HDLC (high-level data link control), 43, 347
Header Error Check field (see HEC field)
HEC field, 174
Hegerle, Blake, 96
hidden nodes
challenges of, 25–27
physical carrier-sensing functions and, 28
RTS threshold and, 369, 370
high-level data link control (see HDLC)
high-power amplifiers (HPAs), 160
high-rate direct sequence (see HR/DS)
high-rate direct-sequence PHY (see HR/DS PHY)
high-rate direct-sequence PLCP (see HR/DS PLCP)
high-rate direct-sequence PMD (see HR/DS PMD)
HiperLAN, 155
home location, 299
Home Wireless Gateway (3Com), 265
Hop Index field, 77, 168
Hop Pattern field, 77, 168
hop patterns, 165, 394
Hop Set field, 77
hopping sequences, 166, 167
Host AP Mode (PRISM chipset), 271
HR/DS (high-rate direct sequence)
802.11b specification, 8
Barker words, 189
channel agility option, 196
CRC field, 193
CS/CCA, 196
DQPSK, 189
Length field, 192
Long Sync field, 191
preamble, 190
Service field, 192
SFD field, 191
Short Sync field, 191
Signal field, 191
transmission, 193
HR/DS PHY, 152, 189–197, 322
HR/DS PLCP, 190–193
HR/DS PMD, 193–196
HR/DSSS (see HR/DS)
HTTP (network performance requirements), 310
Hybrid Mode IKE, 299
HyperLink Technologies, 321

Exhibit 1019
Page 434 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

I

IAPP (inter-access point protocol)
  access points and, 15
  BSS transition and, 21
  functionality of, 299
  mobility and, 378
  purpose of, 263
  Task Group F, 377
iBooks, 397
IBSS (independent basic service set)
  address fields, 54
  ATIM and, 45, 78, 81
  BSSIDs and, 55
  frame features, 58
  overlapping, 16
  power management, 133–137
  states for, 83
  station communication within, 11
  timing synchronization and, 138
IBSS field, 70
IBSS Parameter Set, 78, 119
ICI (inter-carrier interference), 201–202
ICV (integrity check value), 90
IDFT (inverse DFT), 202
IETF Network Working Group, 299
IFFT (inverse fast Fourier transform), 201–202
iMacs, 397
independent basic service set (see IBSS)
Individual/Group bit, 40, 55
industrial, scientific, and medical bands (see ISM bands)
information elements
  CF Parameter Set, 78, 79, 119, 149
  Ethereal analysis and, 345
  management frames, 68, 74–79
  shared-key authentication and, 121
Infrared Data Association (IrDA), 152
infrared light
  compared with radio waves, 152
  as network medium, 3
  physical layer, 152
infrastructure BSS, 11, 55
infrastructure networks
  access points, 12, 372
  address fields in, 54
  authentication for, 120
  Class 1 frames and, 84
  contention-free services and, 27
  DCF and, 31
  distribution and, 18
  dot11DesiredBSSType, 386
  DTIM period and, 372
  power management in, 128–133
  Probe Requests in, 117
  timing synchronization in, 137
  transmitter address in, 58
  wvlan_cs driver, 257
initialization vector (see IV)
installation
  AirPort cards, 397–399
  Etheral, 333–336
  linux-wlan-ng, 245–247
  Lucent ORiNOCO card, 229
  network deployment, 325–328
  Nokia C110/C111 card, 215–216
  orinoco_cs driver, 260
  PCMCIA Card Services, 240–241
  wvlan_cs driver, 255
integration (network service component), 18
integrity
  compromising, 299
  data security and, 89
  data security attribute, 299
  ICV and, 90
integrity check value (see ICV)
Intel
  i82365SL controller, 241
  PRO/Wireless LAN Access Point, 266
  site survey tools, 315
  Wireless Gateway, 265
inter-access point protocol (see IAPP)
inter-carrier interference (see ICI)
interfaces
  802.11 management architecture and, 115
  chipsets used in cards, 237
  configuring for orinoco_cs driver, 261
  management interfaces, 264, 279
  wireless interfaces, x, 269
  (see also AirPort Card; Ethernet interface; web interface)
interference
  avoiding, 153
  challenges working around, 25
  direct sequence and, 180, 181
  frequency hopping and, 166, 169
  fragmentation and, 46
  fragmentation threshold and, 370
  guard time and, 202
  indirect effects of, 330
  multipath interference, 185
  noting potential sources of, 313
  primary sources, 34

Exhibit 1019
Page 435 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

interference (*continued*)
PSK and, 182
spectrum analyzer and, 316
spread spectrum and, 156
interframe spacing
fragments and, 35
PS-Poll frames and, 48
role of, 29–31
International Telecommunications Union (see ITU)
Internet Security, Applications, Authentication and Cryptography group (see ISAAC group)
Intersil
MAC controller, 238
OFDM proposal, 377
PRISM, 198, 237
Intersil-based cards, 244–254, 333
inter-symbol interference (see ISI)
inverse DFT (see IDFT)
inverse fast Fourier transform (see IFFT)
inverse Fourier transform, 202
I/O addresses, 243, 254
Ioannidis, John, 96
ioctl() command, 247
IP addresses
address translation and, 310
linux-wlan-ng and, 249
Mobile IP and, 299
mobility and, 296, 379
Nokia A032 and, 282
roaming and, 313
IPSec specification
access control and, 299
confidentiality and, 299
network performance requirements, 310
residential gateways and, 265
security and, 97, 311
IRQs, 242
ISAAC group, 93
ISI (inter-symbol interference), 162, 201
ISM bands
direct-sequence systems, 181
emission rules, 168
frequency ranges for, 4
higher data rates, 377
spread-spectrum technology and, 155–157
unlicensed use, 153, 154
ITU (International Telecommunications Union), 4, 153
IV (Initialization Vector), 90, 92, 95
iwconfig tool, 256, 258

J

joining
networks, 350–356, 373
frequency-hopping, 168
for linux-wlan-ng, 251
stations and, 119

K

Kerberos authentication, 299
kernel
configuration and compilation, 245
Ethereal prerequisite, 333
keys
management considerations, 92, 94
mapped keys, 90
public keys, 380
recovering, 366
(see also WEP keys)
keystream
one-time pads and, 88
RC4 cipher weakness, 95
reuse as weakness, 92

L

Lamarr, Hedy, 151, 156
latency
DS PHY, 188
FH PHY, 175
OFDM PHY, 213
in RTS/CTS transmission procedure, 27
Length field
DS PHY, 186
HR/DS, 192
OFDM PLCP, 210
libpcap library, 333
licensing
802.11 and, 321
ISM bands and, 4
Nokia driver, 215
radio communications and, 152–155
radio spectrum use and, 4
light waves (see infrared light)
link layer
802.11 specification, 8, 25
802.2 specification and, 8
fragmentation at, 46

Exhibit 1019
Page 436 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

IAPPs and, 299
mobility, 294, 296
Linksys (WAP11), 265
Linux
802.11 support, 236
AirPort Base Stations and, 410
Ethereal and, 333
linux-wlan-ng, 244–254
Lucent Orinoco, 254–261
open source drivers available, 255
PCMCIA support on, 238–244
linux-wlan driver, 244–254
linux-wlan-ng driver
AirSnort and, 363
common problems, 252–254
compiling and installing, 245–247
configuring, 249–252
Ethereal and, 333
prerequisites, 244
Listen Interval
access points and, 82
association procedure, 125
DTIM and, 372
management frames and, 71
power management and, 128
tunable parameter, 369, 371
LLC (logical link control)
802.2 specification, 8
Ethereal and, 332
header fields, 347
lnxreq_ command, 248
lnxreq_wlansniff command, 248
logical link control (see LLC)
logical operators, 343
long retry counter, 32, 33, 282
long retry limit, 370, 390, 391
Long Sync field, 191
long training sequences, 209
low-noise amplifiers (LNAs), 160
Lucent
access control solution, 299
AirPort Base Station, 403
Client Manager, 230
Closed Wireless System extension, 276
interface cards, 237
ORiNOCO card, 229–234, 254–261

M

MAC (medium access control)
802.11 specification and, 7, 8
access modes, 27–31
addresses
access control and, 299
AP-1000 authentication and, 277
authentication and, 121
BSS transitions and, 21
dot11AuthenticateFailStation, 387
dot11DeauthenticateStation, 387
dot11DisassociateStation, 387
dot11MACAddress, 389
dot11WEPKeyMappingAddress, 388
filtering, 121, 264
for stations, 17
using single, 13
address fields, 35
challenges for, 25–27
collision avoidance, 24
contention-based access, 31–34
contention-based services, 44–50
controllers, 238, 254
fragmentation, 34–35, 47
frame format, 35–42
frames
FH PLCP and, 171
PLCP and PMD, 9
PSDUs and, 185
whitening, 174
MIB and, 115, 389–392
physical-layer support, 10
MAC header, 57
MAC layer management entity (see MLME)
MAC (see MAC)
MAC Service Data Unit (see MSDU)
MAC Time field, 346
MacIntosh (Apple Computer), 409–410
make program, 333
Malinen, Jouni, 271
management frames
broadcast and multicast data, 44
contention-free service and, 143
dot11ReceivedFragmentCount, 391
Ethereal and, 343–345, 354
features of, 66–83
shared-key authentication and, 122
management functions, 36, 383
(see also Acknowledgment; RTS/CTS exchange)
management information base (see MIB)
management interfaces, 264, 279
Mantin, Itsik, 95
MaxChannelTime scanning parameter, 116–117

Exhibit 1019
Page 437 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

medium
as component of 802.11 networks, 10
as electromagnetic radation, 3
radio waves challenges as, 5
medium access control (see MAC)
MIB (management information base), 115, 383
MinChannelTime scanning parameter, 116, 117
MLME (MAC layer management entity), 114
Mobile IP
ESS transitions and, 22
mobility and, 379
RFC 2002, 296
roaming and, 299
mobile stations
association and, 125
authentication for, 120
CF-Poll functions and, 59
Current AP Address field and, 71
reassociation, 126
receiving frames, 133
roaming and, 313
TIM and, 129
mobile telephony, 1, 5
mobility
constraint in, 13
distribution system, 14
link layer and, 294
long-term considerations, 378
network deployment, 295–299
networks and, ix, 1, 128
project planning, 308
transition types and, 20–22
monitoring
AirPort interface, 399–401
Beacon frames, 369
wireless stations, 277, 289–291
More Data bit
buffered frames and, 132
control frames and, 61
frame buffering and, 129
specifics, 39
More Fragments bit
ACK frame and, 64
control frames and, 61
example, 47
Frame Control field and, 53
specifics, 38
MSDU (MAC Service Data Unit), 19
multicast frames
acknowledgments for ATIM, 135
contention-free service and, 150
dot11MulticastReceivedFrameCount, 391
dot11MulticastTransmittedFrameCount, 390
dot11TransmittedFragmentCount, 390
DTIM and, 132
duration field in, 53
exchanges and, 44
mobile stations and, 133
multipath fading
antennas and, 321, 374
challenges with, 25
defined, 161
ISI, 162
radio networks and, 161
multipath interference
defined, 161
DQPSK and, 185
features of, 161
radio waves and, 5
multipath time dispersion, 315

N

naming conventions
802.11 fields, 341–345
access points, 326–327
NAT (network address translation), 309
NAV (Network Allocation Vector)
Duration/ID field and, 39
fragments and, 35
PS-Poll frame and, 66
SIFS and, 45
stations seizing medium, 30
virtual carrier-sensing and, 28
network address translation (see NAT)
Network Allocation Vector (NAV), 309
network analysis
AirSnort, 363–367
joining a network, 350–356
web transactions, 356–??
workload information, 349–350
network analyzers
Ethereal, 332–348
reasons to use, 329–331
Network Associates (Sniffer Wireless), 332
network deployment
access points and, 263
batteries and, 268

Exhibit 1019
Page 438 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

enterprise gateways and, 266
external antennas and, 267
installation, 325–328
long-term considerations, 381
network analyzers and, 332
project planning, 307–314
roaming and mobility, 295–299
security, 299
site survey, 314–325
speed advantages, ix
supplying power, 268
topology, 294, 299
Unix-based routers and, 269
wireless LANs, 293
network identifiers (see NIDs)
network layer
confidentiality at, 299
fragmentation at, 46
network profiles (see profiles)
network services, 17–20
networks
backbone networks, 10, 12, 14
boundaries, 5, 16
changing fragmentation thresholds, 47
expansion of community networks, 3
foreign network, 299
joining, 350–356, 373
mobility and, 1, 13
operations and, 16–20
project planning, 313
scanning, 115–119, 351
throughput and capacity, 310
types of, 10–14
nid command, 287
NIDs (network identifiers), 287
nodes (see hidden nodes)
noise
defined, 152
direct sequence and, 177, 181
electrical storms and, 298
noise factor, 160
Noise field, 346
Nokia
A032
case study example, 349–350
enterprise gateway, 266
features of, 279–291
C110/C111 card, 215–228
P020 Public Access Controller, 299
Null frames, 57, 58

O

OFDM (orthogonal frequency division multiplexing)
802.11a standard and, 6, 8 205–208
encoding features, 212
features of, 199–205
Intersil proposal, 377
modulation technique, 311
OFDM table, 395
spread-spectrum type, 156
OFDM PHY
802.11a standard, 152
BPSK and, 211
CCA, 212
data rates, 211
dot11FrequencyBandsSupported, 395
dot11PHYType, 392
latency, 213
parameters, 212
QPSK, 211
OFDM PLCP, 208–211
OFDM PMD, 211
omnidirectional antennas
benefit of, 316
features of, 159
multipath interference and, 161
one-time pads, 88
open-system authentication, 120, 354
operating channels, 206–208
operators, 343
Order bit, 39, 61
organizationally unique identifier (see OUI)
Orinoco AP Manager, 271, 278
ORiNOCO AP-1000, 266, 269–278
ORiNOCO AP-2000, 266
ORiNOCO AS-2000, 299
ORiNOCO card, 229–234, 254–261
orinoco_cs driver, 255, 260
orthogonal frequency division multiplexing (see OFDM)
orthogonal frequency division multiplexing PHY (see OFDM PHY)
orthogonal frequency division multiplexing PLCP (see OFDM PLCP)
orthogonal frequency division multiplexing PMD (see OGDM PMD)
orthogonal hopping sequences, 166, 168
orthogonality, 200
OSI model, 8

Exhibit 1019
Page 439 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

OUI (organizationally unique identifier)
  Ethereal example, 350
  LLC header component, 348
  SNAP and, 43

P

p2req_ command, 248
packet binary convolution coding (see PBCC)
packet error rate (see PER)
packet sniffers, 23
packets
  filtering, 299
  key length and, 367
  marking, 341
Pad field, 211
parabolic antennas, 319
parameters
  802.11a choices, 206
  command-line, 281
  DS PHY, 189
  FH PHY, 175
  HR/DS PHY, 197
  Nokia driver, 224
  OFDM PHY, 212
  performance tuning, 374
  power management, 371–373
  radio management, 368–371
  for scanning, 115
  timing operations, 373
  timing parameters, 119
  tuning for 802.11 standard, 258
  wvlan_cs driver, 259
Parity bit, 210
passive scan timer, 373
passive scanning, 115, 116, 351
Path MTU Discovery (RFC 1191), 42
PBCC (packet binary convolution coding)
  802.11b option, 196
  management frames and, 70
  Task Group G, 377
PBCC field, 70
PCF (point coordination function)
  atomic exchanges and, 44
  contention-free service with, 27, 140–150
  dot11MediumOccupancyLimit, 384
  features of, 27
  throughput using, 311
PCF interframe space (see PIFS)
PCMCIA
  cards
    access points and, 263
    AP-1000 and, 269
    common I/O ports, 243
    common IRQ settings, 242
    enterprise gateways and, 265
  Card Services
    Linux support, 238–244
    linux-wlan and, 245
    linux-wlan-ng, 247, 249, 253
PER (packet error rate), 315
performance tuning
  dot11CountersTable, 390
  power management, 371–373
  project planning, 309
  radio management, 368–371
  timing operations, 373
  tunable parameters, 374
  wireless networks, 23
Perl, Ethereal and, 333
phase shift keying (PSK)
  DBPSK, 183
  DQPSK, 184, 194
  encoding data, 182
physical-layer convergence procedure (see PLCP)
physical-layer management entity (see PLME)
physical-medium dependent (see PMD)
physical (PHY) layer, 119, 392
  802.11 MAC and, 10, 24
  802.11 standard, 8, 152, 164
  interframe space times and, 30
  MIB and, 115, 392–395
  PMD and, 151
  radio waves as, 9
PIFS (PCF interframe space), 30
PIN Unlocking Key (PUK), 226
ping tool, 264, 291
PINs, 226, 227
planning, project, 307–314
PLCP (physical-layer convergence procedure)
  802.11 specification, 9
  DS PHY, 185–186
  frequency hopping and, 171–174
  HR/DS, 190–193
  MAC frames and, 9
  OFDM, 208–211

Exhibit 1019
Page 440 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

physical layer and, 9, 151
Probe Requests and, 190
PLCP service data units (PSDUs), 173, 185
PLCP Signaling field (PSF), 173
plus sign (+), 283
PLME (physical-layer management entity), 114
PMD (physical-medium dependent)
802.11 specification and, 9
direct sequence and, 187–188
frequency hopping and, 174–175
HR/DS, 193–196
MAC frames and, 9
OFDM PMD, 211
PHY sublayer, 151
physical layer and, 9
PN codes (pseudo-random noise codes)
Barker words and, 178
chipping streams, 177
point coordination function (see PCF)
point coordinator (PC), 27, 59
polling lists
contention free periods, 149
PCF operation and, 142
portability, 295
ports, serial, 264
power management
802.11 networks, 128–137
802.11 standard and, 268
Beacon generation and, 138
dot11PowerManagementMode, 386
IBSS, 133–137
Lucent ORiNOCO card, 231
PCF and, 149
performance tuning, 371–373
PS-Poll frames and, 48–50
Power Management bit
access points and, 60
control frames and, 61
null frames and, 57
specifics, 38
PowerBooks, 397
power-saving mode, 128
PPP, EAP and, 378
preamble
802.11 FH PHY, 173
DS PHY, 186
HR/DS, 190
OFDM PLCP, 209
preauthentication, 123
PRISM chipset
Ethereal analysis and, 345, 346
Host AP Mode, 271
Intersil-based, 198
linux-wlan support, 244
market leader, 237
Privacy bit, 70
privacy (network service component), 19
PRNG (pseudo-random number generator), 87
Probe Request frames
active scanning and, 115, 116, 351
Capability Information field, 69
features of, 79
as management frames, 45
PLCP headers and, 190
Probe Response frames
acknowledgment and, 353
active scanning and, 117, 351
Capability Information field, 69
features of, 80
timing synchronization and, 137
ProbeDelay scanning parameter, 116, 117
processing gain, 179
profiles
moving to smart cards, 227
Nokia card and settings, 217–220
ORiNOCO card, 230–232
project planning, 307–314
promiscuous capture, 248, 365
propagation
radio waves problems with, 5
in three dimensions, 11
Protocol field, 350
Protocol Type field, 348
PRO/Wireless LAN Access Point (Intel), 266
Proxim
RangeLAN2, 266
site survey tools, 315
PSDU length word (PLW) field, 173
pseudo-random noise codes (see PN codes)
pseudo-random number generator (see PRNG)
PS-Poll frames
deferred response and, 49
defined, 40
features, 65
frame buffering and, 48, 129
frame delivery, 132
State 3 and, 85
public-key infrastructure (PKI), 299
public keys, 380

Exhibit 1019
Page 441 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Q

quadrature amplitude modulation (QAM), 211
quadrature phase shift keying (QPSK), 211
quality of service (QoS), 377

R

radiation
  802.11 and, 25
  network mediums and, 3
  spectrum analyzers and, 313
radio communications
  licensing and, 152–155
  performance tuning, 368–371
radio frequency (see RF)
radio waves
  challenges as network medium, 5
  compared with IR PHY, 152
  as network medium, 3
  as physical layer, 9
RADIUS
  access control solution, 299
  Nokia A032 security, 285
  personal WEP and, 286
  security considerations, 311
radome, 319
random number generators, 54
RangeLAN2 (Proxim), 266
Rate field, 210, 346
RC4 cipher, 87, 95
Reason Code field, 72
reason codes, 72, 81
reassociation
  linux-wlan-ng problems, 254
  network service component, 18
  procedure for, 126–127
Reassociation Request frames, 82, 126
Reassociation Response frames, 82
received signal strength indication (see RSSI)
receiver address
  ACK frames and, 64
  CTS frame and, 63
  data frames and, 54
  purpose of, 41
  RTS frames and, 63
Receiver Address field
  CF-End frame and, 147
  CF-End+CF-ACK frame, 148
receivers
  ISI and, 201
  short training sequence, 209
redundancy, 299
regulations
  gain requirements, 179
  high-gain antennas and, 319
  radio spectrum use, 4
  (see also Federal Communications Commission)
regulatory domains
  802.11a standard, 198
  dot11CurrentRegDomain, 393
  hop sets in, 168
  radio channel usage, 323
rejections, filtering and, 85
reliability
  acknowledgment and, 45
  of Ethernet networks, 24
  frame delivery and, 17
reports, scan, 118
Request to Send (see RTS)
residential gateways, 264, 269
resource conflicts/constraints
  considerations, 380
  linux-wlan-ng, 252
  troubleshooting, 241–244
Retry bit, 38, 61
retry counters
  contention windows and, 34
  DCF error recovery and, 32
  tuning with iwconfig tool, 259
  using, 33
retry limits, 370
RF (radio frequency)
  802.11 specification and, 158–163
  FCC regulating, 153
  IR PHY and, 152
  link quality, 25
  wireless medium standard, 10
RFC 1042 (data encapsulation), 43, 244, 348
RFC 1191 (Path MTU Discovery), 42
RFC 1918 (private addresses), 299
RFC 1990 (Multilink PPP), 46
RFC 2002 (Mobile IP), 296
RFC 2284 (EAP), 378
RFC 2484 (EAP), 378
RG58 cable, 320
RJ-45 ports, 282
roaming
  802.11 standard and, 267
  limitations of, 313
  Mobile IP and, 299
  network deployment, 295–299
  requirements for effective, 269

Exhibit 1019
Page 442 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

RSA Security, Inc.
RC4, 88
SecurID, 299
RSSI (received signal strength indication), 315
RTS (Request to Send)
components of, 62
interframe spacing and, 31
NAV and, 28
preventing collisions with, 26
virtual RTS, 53
(see also RTS/CTS exchange)
RTS frames, 135, 391
RTS threshold
dot11LongRetryLimit, 390
dot11RTSThreshold, 390
dot11ShortRetryLimit, 390
parameter, 282
setting, 27
tuning, 258, 369
RTS/CTS exchange
atomic operations and, 28
controlling, 27
data transmission and, 47
dot11RTSThreshold, 390
fragmentation and, 35, 48
hidden nodes and, 27
overlapping networks and, 29
purpose of, 369
RTS threshold and, 32
SIFS and, 30
Rubin, Avi, 96

S

S-band ISM, 4
scan reports, 118
scan timers, 373
ScanType scanning parameter, 115
Schneier, Bruce, 299
secrecy (data security attribute), 299
secure socket layer (see SSL)
SecurID (RSA), 299
security
802.11 and, 23, 267
access points and, 263
address filtering, 121
AirPort Base Station, 405
authorization and, 89
configuring AP-1000, 275–277
enterprise gateways and, 266
IR-based LANs and, 152
long-term considerations, 380
network analyzers and, 331
network deployment, 299, 327
Nokia A032 configuration, 284–289
project planning, 309, 311
Task Group I, 378
web management and, 280
WEP and, 19
wireless networks, 5, 299
(see also WEP)
Sequence Control field, 41
serial ports, 264
Service field
DS PHY, 186
HR/DS, 192
OFDM PLCP, 210
Service Set ID (see SSID)
set command, 279
set manager specific command, 287
set manager_ip command, 287
SFD field, 173, 186, 190–191
Shamir, Adi, 95
shared-key authentication, 120, 121
short interframe space (see SIFS)
Short Preamble bit, 70
short retry count, 32, 282
short retry limit, 370, 390, 391
Short Sync field, 191
short training sequence, 209
show a command, 289
show g command, 290
shutdown process, 18
SIFS (short interframe space), 30, 45
signal boost, amplifiers and, 160
Signal field
DS PHY, 186
HR/DS, 191
PLCP header, 209
PRISM capture, 346
signal power
decibels and, 160
dot11TIThreshold, 395
effects of materials on, 313
reassociation and, 126
spread spectrum technology and, 155
signal-to-noise ratio (see SNR)
site survey, 314–325
sleeping mode
alternate names for, 133
frame transmission and, 131
power conservation and, 128
small offices or home offices (SOHO), 325
smart cards, 226–228

Exhibit 1019
Page 443 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

SME (system management entity), 114
Snake Oil Warning Signs: Encryption Software to Avoid, 299
SNAP (sub-network access protocol), 43, 347
Sniffer Wireless (Network Associates), 332
sniffing
  packet sniffers, 23
  WEP design flaws, 94
  wireless network security considerations, 5
SNMP
  AP-1000 and, 271, 276
  enterprise gateways and, 266
  security using WEP, 388
SNR (signal-to-noise ratio), 234
source service access point (see SSAP)
spectrum analyzers, 313, 316
Spectrum Managed 802.11a (SMa), 377
spread spectrum
  direct-sequence transmission, 176
  ETSI rules for, 169
  ISM bands and, 155–157
spreader, 177
spreading ratio, 178
SSAP (source service access point), 43, 347
SSB Electronics, 321
SSH
  confidentiality and, 299
  network performance requirements, 310
  security recommendations, 98
SSID (Service Set ID)
  broadcast SSID, 75, 117
  dot11DesiredSSID, 386
  information element, 75
  network connections and, 299
  Nokia driver network profiles and, 217
  scanning and, 352
SSID scanning parameter, 115
SSL (secure socket layer), 306
standards
  Bluetooth standard, 5, 378
  comparison of 802.11 standards, 6
  IAPP and, 21
  wireless interfaces and antennas, x
Start Frame Delimiter field (see SFD field)
states, frame types and, 83–85
station configuration table, 384–387, 389
stations
  802.11 compliance and, 20
  access points and, 12
  active scanning and, 116
  authentication, 83, 120
  Beacon frames, 369
  BSS transition and, 21
  communication within ESSs, 13
  as component in 802.11 network, 9
  distribution system, 14
  dot11StationID, 384
  importance of associating, 18
  joining BSSs, 119
  limitations, 15
  Listen Intervals and, 371
  MAC addresses for, 17
  memory and frame buffering, 136
  monitoring, 277
  MSDU delivery and, 19
  naming allowed, 287
  as point coordinators, 27
  polling list and, 142
  Probe Requests, 117
  retry counters and, 32
  set manager_ip command, 287
  SSIDs and, 75
  (see also mobile stations; wireless stations)
statistics, monitoring for Nokia A032, 291
stats command, 291
Status Code field, 73, 83
status codes
  dot11AuthenticateFailStatus, 387
  in Status Code field, 73
  shared-key authentication and, 123
Stream ciphers
  as compromise, 88
  RC4 as, 87
  vulnerability of, 94
Stubblefield, Adam, 96, 363
sub-network access protocol (see SNAP)
superposition, 161
Supported Rates information element
  Association Request frame and, 82
  basic rate set and, 119
  features of, 75
symbol rate, 170, 171
symbol time
  4GFSK, 171
  DBPSK, 183
  guard time and, 201, 202, 206
Sync field, 173, 186, 190
system management entity (see SME)

Exhibit 1019
Page 444 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

T

Tail field, 210, 211
target Beacon transmission time (see TBTT)
Task Group D (TGd), 376
Task Group E (TGe), 377
Task Group F (TGf), 377, 378
Task Group G (TGg), 377
Task Group H (TGh), 377
Task Group I (TGi), 378
TBTT (target Beacon transmission time), 138
TCP
  three-way handshake, 360
  web transaction example, 362
tcpdump, 339
TCP/IP
  cheap access points and, 263
  Ethereal and, 332
  Mobile IP and, 22
  network performance requirements, 309
telephony (see mobile telephony)
telnet connections, 283
temperature range, 393
Texas Instruments, 377
TFTP (trivial file transfer protocol)
  access control and, 299
  address filtering and, 121
  NID mappings and, 287
throughput
  direct-sequence modulation and, 178
  direct-sequence systems and, 181
  expectations for, 311
  frequency-hopping systems and, 168
  guard time and, 202
  project planning, 308
  retry limits and, 371
TIM (traffic indication map)
  Beacon frames and, 79
  buffered frames and, 372
  frame delivery, 129
  power and, 49
TIM information element, 77, 130
time units (TUs)
  active timer and, 373
  ATIM frames, 78
  ATIM window and, 372
  contention-free periods and, 149
  defined, 69
  dot11MaxTransmitMSDULifetime, 390
  dwell time and, 76
  scanning parameter and, 116, 118
timeouts
  dot11AssociationResponseTimeOut, 386
  dot11AuthenticationResponseTimeout, 385
  joining networks and, 373
  linux-wlan-ng problems, 254
timer synchronization
  802.11 networks, 137–139
  frequency hopping and, 119
Times Microwave (LMR-200), 320
Times Microwave (LMR-400), 320
Timestamp field
  management frames and, 72
  timing parameter, 119
  timing synchronization and, 137
timestamps
  Beacon frames on frequency-hopping networks, 168
  rules for, 138
  scanning and, 352
  TSF and, 137
timing operations
  802.11 MAC, 27–31
  performance tuning, 373
timing synchronization function (see TSF)
ToDS bit
  ARP requests and, 357
  control frames and, 61
  DNS request and, 359
  frames from distribution systems, 59
  specifics, 38
Token Ring, 8
traceroute tool, 264, 291
Traffic Indication Map (see TIM)
traffic indication map information element (see TIM information element)
transceivers
  802.11 direct-sequence networks, 187
  power conservation and, 128
transmission
  802.11 MAC rules for, 31
  from access points, 142–143
  acknowledgment and, 45
  authentication and, 83–85
  contention-free periods and, 40
  direct-sequence, 176–182
  dot11MaxTransmitMSDULifetime, 390
  dot11TransmittedFrameCount, 391
  DS PHY, 187
  DS PMD, 187
  FH PMD, 174
  frequency-hopping, 165–169

Exhibit 1019
Page 445 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

transmission (*continued*)
HR/DS, 193
licenses and, 153
licensing and, 156
MAC and, 8
OFDM and, 199
OFDM PLCP, 211
power levels and, 374
success defined, 33
using CCK, 193–195
transmitter address
PS-Poll frame, 66
purpose of, 41
RTS frames and, 63
trivial file transfer protocol (see TFTP)
troubleshooting
Ethernet interface, 275
linux-wlan-ng, 252–254
Nokia A032, 291
(see also debugging; error handling)
TSF (timing synchronization finction), 137
TUs (see time units)
Type code field
ARP example, 43
IP example, 43
listed, 36

U

UI (unnumbered information), 43
unauthorized users (see access control)
unicast frames
802.11 transmission rules, 31
acknowledgments and, 44
buffering, 48, 129
considerations, 45–48
dot11TransmittedFragmentCount, 390
UNII bands
5-GHz bands, 155, 198
dot11FrequencyBandsSupported, 395
operating channels, 207
Universal/Local bit, 41, 55
Unix operating system, 271
unlicensed bands, 153, 198
Unlicensed National Information Infrastructure bands (see UNII bands)
unnumbered information (see UI)

V

variable fields (see information elements)
VENONA cryptographic systems, 88
version information, 233
vertical antennas, 317
virtual bitmap, 77, 78
virtual LANs (see VLANs)
virtual private networks (see VPNs)
visited network (see foreign network)
VLANs (virtual LANs)
802.1q specification, 8
ESSs and, 13
mobility and, 296
network planning, 314
VPNs (virtual private networks), 311

W

WAP11 (Linksys), 265
WaveLAN
iwconfig tool and, 256
as proprietary system, 254
wireless LAN development and, 229
WDS (wireless distribution system), 56, 60
web interface
address filtering and, 289
DHCP and, 284
Nokia A032, 283
web servers, 356–363
WECA (Wireless Ethernet Compatibility Alliance), 6, 266
WEP (Wired Equivalent Privacy)
802.11 standard and, 267
AirSnort and, 364
configuring, 251, 275–277
cryptographic background to, 86–89
cryptographic operations, 89–93
defined, 39
design flaws in, 93, 299
dot11PrivacyOptionImplemented, 386
dot11WEPUndecryptableCount, 391
frames and, 60
keys
access control challenges and, 299
default keys, 90
dot11WEPDefaultKeyValue, 388
dot11WEPKeyMappingAddress, 388
dot11WEPKeyMappingValue, 389
dot11WEPKeyMappingWEPOn, 389
mapping table, 388
setting in linux-wlan-ng, 248
limitations of, 299
Lucent ORiNOCO, 229
MIB tables, 388
Nokia A032 security and, 285
personal WEP, 286

Exhibit 1019
Page 446 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

Privacy bit and, 70
privacy service and, 19
problems with, 93–96
recommendations, 96–98
security considerations, 380
shared-key authentication, 120, 121
WEP default key table, 388
wvlan_cs driver, 258
WEP bit
control frames and, 61
specifics, 39
WEP Information field, 341
whitening
FH PHY and, 185
PLCP and, 172–174
Wi-Fi (wireless fidelity), 6, 266
WildPackets (AiroPeek), 332
windowing technique, 205
Windows
Ethereal and, 336
installing drivers, 214
Lucent ORiNOCO card, 229–234
Nokia C110/C111 card, 215–228
WinPcap library, 336
Wired Equivalent Privacy (see WEP)
wired networks
association equivalency in, 125
authentication on, 120
configuring AP-1000, 275
physical access to, 19
wireless distribution system (see WDS)
wireless Ethernet, 6, 16
Wireless Ethernet Compatibility Alliance (see WECA)
wireless fidelity (see Wi-Fi)
Wireless Gateway (Intel), 265
wireless networks
advantages of, ix, 1–5
cost advantages, x
major components of, 9
names used to describe, 6
resource constraints, 380
wireless stations, 289–291, 296
wlanctl-ng command, 248
working groups
CCK, 189
IAPP standard, 263
IETF Network Working Group, 299
security, 96, 267
web site, 376
(see also entries beginning with Task Group)
workload information, 349–350
wvlan_cs driver, 255–260

X

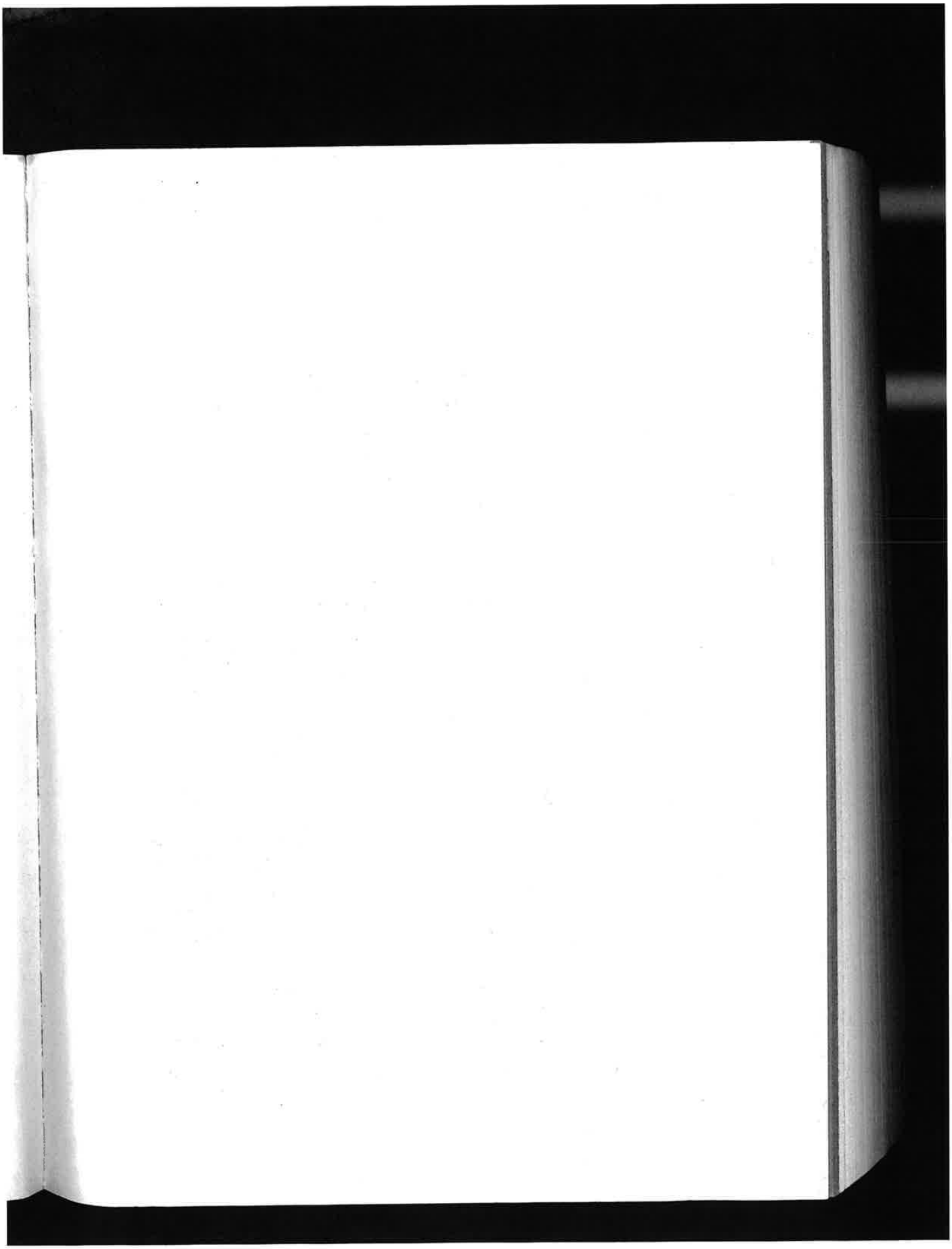
XAUTH (Extended Authentication), 304
XOR (exclusive OR), 87, 90

Y

yacc, Ethereal and, 333
Yagi antennas, 317, 319

Exhibit 1019
Page 447 of 451

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148



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.

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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.

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148

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*

Ubiquiti Inc. v. XR. Comm.
IPR2024-00148