It looks like Ethernet will continue to expand in its applications for some time. 10-gigabit Ethernet has freed it from the distance constraints of CSMA/CD. Much effort is being put into **carrier-grade Ethernet** to let network providers offer Ethernet-based services to their customers for metropolitan and wide area networks (Fouli and Maler, 2009). This application carries Ethernet frames long distances over fiber and calls for better management features to help operators offer reliable, high-quality services. Very high speed networks are also finding uses in backplanes connecting components in large routers or servers. Both of these uses are in addition to that of sending frames between computers in offices.

4.4 WIRELESS LANS

Wireless LANs are increasingly popular, and homes, offices, cafes, libraries, airports, zoos, and other public places are being outfitted with them to connect computers, PDAs, and smart phones to the Internet. Wireless LANs can also be used to let two or more nearby computers communicate without using the Internet.

The main wireless LAN standard is 802.11. We gave some background information on it in Sec. 1.5.3. Now it is time to take a closer look at the technology. In the following sections, we will look at the protocol stack, physical-layer radio transmission techniques, the MAC sublayer protocol, the frame structure, and the services provided. For more information about 802.11, see Gast (2005). To get the truth from the mouth of the horse, consult the published standard, IEEE 802.11-2007 itself.

4.4.1 The 802.11 Architecture and Protocol Stack

802.11 networks can be used in two modes. The most popular mode is to connect clients, such as laptops and smart phones, to another network, such as a company intranet or the Internet. This mode is shown in Fig. 4-23(a). In infrastructure mode, each client is associated with an **AP (Access Point)** that is in turn connected to the other network. The client sends and receives its packets via the AP. Several access points may be connected together, typically by a wired network called a **distribution system**, to form an extended 802.11 network. In this case, clients can send frames to other clients via their APs.

The other mode, shown in Fig. 4-23(b), is an **ad hoc network**. This mode is a collection of computers that are associated so that they can directly send frames to each other. There is no access point. Since Internet access is the killer application for wireless, ad hoc networks are not very popular.

Now we will look at the protocols. All the 802 protocols, including 802.11 and Ethernet, have a certain commonality of structure. A partial view of the 802.11 protocol stack is given in Fig. 4-24. The stack is the same for clients and

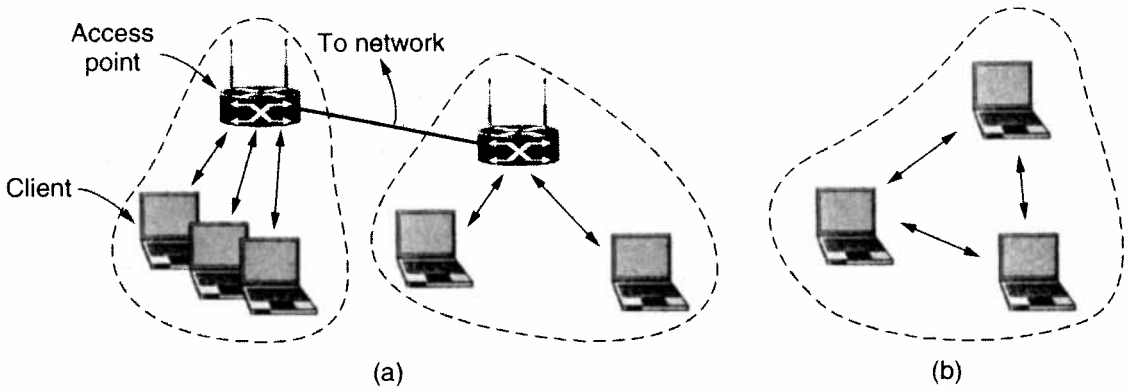

Figure 4-23. 802.11 architecture. (a) Infrastructure mode. (b) Ad-hoc mode.

APs. The physical layer corresponds fairly well to the OSI physical layer, but the data link layer in all the 802 protocols is split into two or more sublayers. In 802.11, the MAC (Medium Access Control) sublayer determines how the channel is allocated, that is, who gets to transmit next. Above it is the LLC (Logical Link Control) sublayer, whose job it is to hide the differences between the different 802 variants and make them indistinguishable as far as the network layer is concerned. This could have been a significant responsibility, but these days the LLC is a glue layer that identifies the protocol (e.g., IP) that is carried within an 802.11 frame.

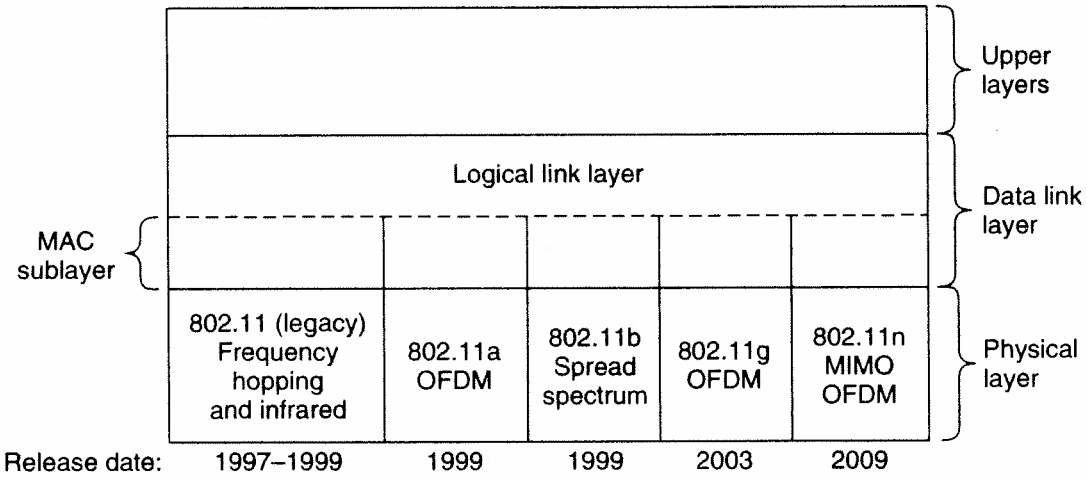

Figure 4-24. Part of the 802.11 protocol stack.

Several transmission techniques have been added to the physical layer as 802.11 has evolved since it first appeared in 1997. Two of the initial techniques, infrared in the manner of television remote controls and frequency hopping in the 2.4-GHz band, are now defunct. The third initial technique, direct sequence spread spectrum at 1 or 2 Mbps in the 2.4-GHz band, was extended to run at rates up to 11 Mbps and quickly became a hit. It is now known as 802.11b.

To give wireless junkies a much-wanted speed boost, new transmission techniques based on the OFDM (Orthogonal Frequency Division Multiplexing) scheme we described in Sec. 2.5.3 were introduced in 1999 and 2003. The first is called 802.11a and uses a different frequency band, 5 GHz. The second stuck with 2.4 GHz and compatibility. It is called 802.11g. Both give rates up to 54 Mbps.

Most recently, transmission techniques that simultaneously use multiple antennas at the transmitter and receiver for a speed boost were finalized as 802.11n in Oct. 2009. With four antennas and wider channels, the 802.11 standard now defines rates up to a startling 600 Mbps.

We will now examine each of these transmission techniques briefly. We will only cover those that are in use, however, skipping the legacy 802.11 transmission methods. Technically, these belong to the physical layer and should have been examined in Chap. 2, but since they are so closely tied to LANs in general and the 802.11 LAN in particular, we treat them here instead.

4.4.2 The 802.11 Physical Layer

Each of the transmission techniques makes it possible to send a MAC frame over the air from one station to another. They differ, however, in the technology used and speeds achievable. A detailed discussion of these technologies is far beyond the scope of this book, but a few words on each one will relate the techniques to the material we covered in Sec. 2.5 and will provide interested readers with the key terms to search for elsewhere for more information.

All of the 802.11 techniques use short-range radios to transmit signals in either the 2.4-GHz or the 5-GHz ISM frequency bands, both described in Sec. 2.3.3. These bands have the advantage of being unlicensed and hence freely available to any transmitter willing to meet some restrictions, such as radiated power of at most 1 W (though 50 mW is more typical for wireless LAN radios). Unfortunately, this fact is also known to the manufacturers of garage door openers, cordless phones, microwave ovens, and countless other devices, all of which compete with laptops for the same spectrum. The 2.4-GHz band tends to be more crowded than the 5-GHz band, so 5 GHz can be better for some applications even though it has shorter range due to the higher frequency.

All of the transmission methods also define multiple rates. The idea is that different rates can be used depending on the current conditions. If the wireless signal is weak, a low rate can be used. If the signal is clear, the highest rate can be used. This adjustment is called **rate adaptation**. Since the rates vary by a factor of 10 or more, good rate adaptation is important for good performance. Of course, since it is not needed for interoperability, the standards do not say how rate adaptation should be done.

The first transmission method we shall look at is **802.11b**. It is a spread-spectrum method that supports rates of 1, 2, 5.5, and 11 Mbps, though in practice the operating rate is nearly always 11 Mbps. It is similar to the CDMA system we

examined in Sec. 2.5, except that there is only one spreading code that is shared by all users. Spreading is used to satisfy the FCC requirement that power be spread over the ISM band. The spreading sequence used by 201.11b is a **Barker sequence**. It has the property that its autocorrelation is low except when the sequences are aligned. This property allows a receiver to lock onto the start of a transmission. To send at a rate of 1 Mbps, the Barker sequence is used with BPSK modulation to send 1 bit per 11 chips. The chips are transmitted at a rate of 11 Mchips/sec. To send at 2 Mbps, it is used with QPSK modulation to send 2 bits per 11 chips. The higher rates are different. These rates use a technique called **CCK (Complementary Code Keying)** to construct codes instead of the Barker sequence. The 5.5-Mbps rate sends 4 bits in every 8-chip code, and the 11-Mbps rate sends 8 bits in every 8-chip code.

Next we come to **802.11a**, which supports rates up to 54 Mbps in the 5-GHz ISM band. You might have expected that 802.11a to come before 802.11b, but that was not the case. Although the 802.11a group was set up first, the 802.11b standard was approved first and its product got to market well ahead of the 802.11a products, partly because of the difficulty of operating in the higher 5-GHz band.

The 802.11a method is based on **OFDM (Orthogonal Frequency Division Multiplexing)** because OFDM uses the spectrum efficiently and resists wireless signal degradations such as multipath. Bits are sent over 52 subcarriers in parallel, 48 carrying data and 4 used for synchronization. Each symbol lasts 4μs and sends 1, 2, 4, or 6 bits. The bits are coded for error correction with a binary convolutional code first, so only 1/2, 2/3, or 3/4 of the bits are not redundant. With different combinations, 802.11a can run at eight different rates, ranging from 6 to 54 Mbps. These rates are significantly faster than 802.11b rates, and there is less interference in the 5-GHz band. However, 802.11b has a range that is about seven times greater than that of 802.11a, which is more important in many situations.

Even with the greater range, the 802.11b people had no intention of letting this upstart win the speed championship. Fortunately, in May 2002, the FCC dropped its long-standing rule requiring all wireless communications equipment operating in the ISM bands in the U.S. to use spread spectrum, so it got to work on **802.11g**, which was approved by IEEE in 2003. It copies the OFDM modulation methods of 802.11a but operates in the narrow 2.4-GHz ISM band along with 802.11b. It offers the same rates as 802.11a (6 to 54 Mbps) plus of course compatibility with any 802.11b devices that happen to be nearby. All of these different choices can be confusing for customers, so it is common for products to support 802.11a/b/g in a single NIC.

Not content to stop there, the IEEE committee began work on a high-throughput physical layer called **802.11n**. It was ratified in 2009. The goal for 802.11n was throughput of at least 100 Mbps after all the wireless overheads were removed. This goal called for a raw speed increase of at least a factor of four. To make it happen, the committee doubled the channels from 20 MHz to 40 MHz and

reduced framing overheads by allowing a group of frames to be sent together. More significantly, however, 802.11n uses up to four antennas to transmit up to four streams of information at the same time. The signals of the streams interfere at the receiver, but they can be separated using **MIMO (Multiple Input Multiple Output)** communications techniques. The use of multiple antennas gives a large speed boost, or better range and reliability instead. MIMO, like OFDM, is one of those clever communications ideas that is changing wireless designs and which we are all likely to hear a lot about in the future. For a brief introduction to multiple antennas in 802.11 see Halperin et al. (2010).

4.4.3 The 802.11 MAC Sublayer Protocol

Let us now return from the land of electrical engineering to the land of computer science. The 802.11 MAC sublayer protocol is quite different from that of Ethernet, due to two factors that are fundamental to wireless communication.

First, radios are nearly always half duplex, meaning that they cannot transmit and listen for noise bursts at the same time on a single frequency. The received signal can easily be a million times weaker than the transmitted signal, so it cannot be heard at the same time. With Ethernet, a station just waits until the ether goes silent and then starts transmitting. If it does not receive a noise burst back while transmitting the first 64 bytes, the frame has almost assuredly been delivered correctly. With wireless, this collision detection mechanism does not work.

Instead, 802.11 tries to avoid collisions with a protocol called **CSMA/CA (CSMA with Collision Avoidance)**. This protocol is conceptually similar to Ethernet's CSMA/CD, with channel sensing before sending and exponential back off after collisions. However, a station that has a frame to send starts with a random backoff (except in the case that it has not used the channel recently and the channel is idle). It does not wait for a collision. The number of slots to backoff is chosen in the range 0 to, say, 15 in the case of the OFDM physical layer. The station waits until the channel is idle, by sensing that there is no signal for a short period of time (called the DIFS, as we explain below), and counts down idle slots, pausing when frames are sent. It sends its frame when the counter reaches 0. If the frame gets through, the destination immediately sends a short acknowledgement. Lack of an acknowledgement is inferred to indicate an error, whether a collision or otherwise. In this case, the sender doubles the backoff period and tries again, continuing with exponential backoff as in Ethernet until the frame has been successfully transmitted or the maximum number of retransmissions has been reached.

An example timeline is shown in Fig. 4-25. Station *A* is the first to send a frame. While *A* is sending, stations *B* and *C* become ready to send. They see that the channel is busy and wait for it to become idle. Shortly after *A* receives an acknowledgement, the channel goes idle. However, rather than sending a frame right away and colliding, *B* and *C* both perform a backoff. *C* picks a short backoff,

and thus sends first. *B* pauses its countdown while it senses that *C* is using the channel, and resumes after *C* has received an acknowledgement. *B* soon completes its backoff and sends its frame.

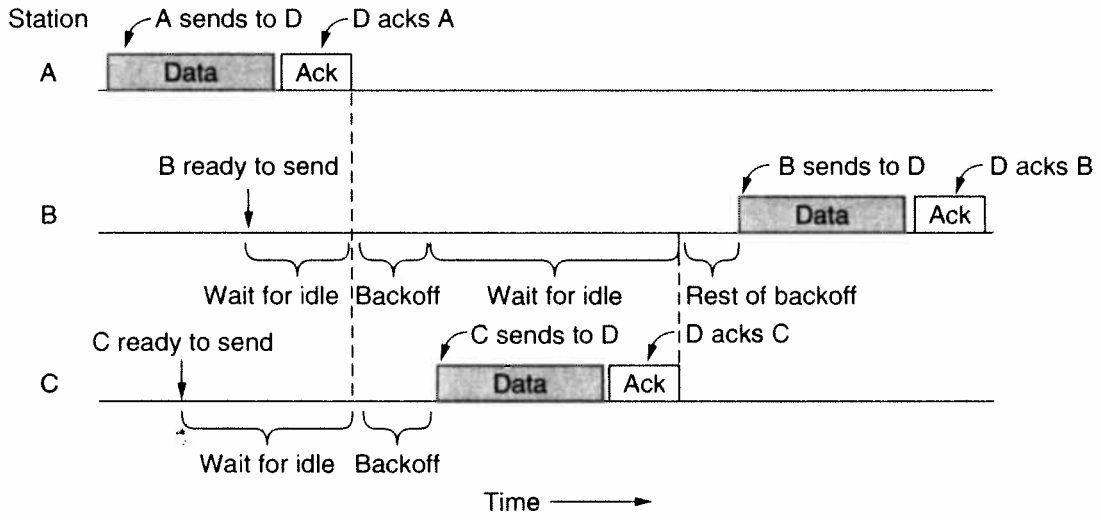

Figure 4-25. Sending a frame with CSMA/CA.

Compared to Ethernet, there are two main differences. First, starting backoffs early helps to avoid collisions. This avoidance is worthwhile because collisions are expensive, as the entire frame is transmitted even if one occurs. Second, acknowledgements are used to infer collisions because collisions cannot be detected.

This mode of operation is called **DCF (Distributed Coordination Function)** because each station acts independently, without any kind of central control. The standard also includes an optional mode of operation called **PCF (Point Coordination Function)** in which the access point controls all activity in its cell, just like a cellular base station. However, PCF is not used in practice because there is normally no way to prevent stations in another nearby network from transmitting competing traffic.

The second problem is that the transmission ranges of different stations may be different. With a wire, the system is engineered so that all stations can hear each other. With the complexities of RF propagation this situation does not hold for wireless stations. Consequently, situations such as the hidden terminal problem mentioned earlier and illustrated again in Fig. 4-26(a) can arise. Since not all stations are within radio range of each other, transmissions going on in one part of a cell may not be received elsewhere in the same cell. In this example, station *C* is transmitting to station *B*. If *A* senses the channel, it will not hear anything and will falsely conclude that it may now start transmitting to *B*. This decision leads to a collision.

The inverse situation is the exposed terminal problem, illustrated in Fig. 4-26(b). Here, *B* wants to send to *C*, so it listens to the channel. When it hears a

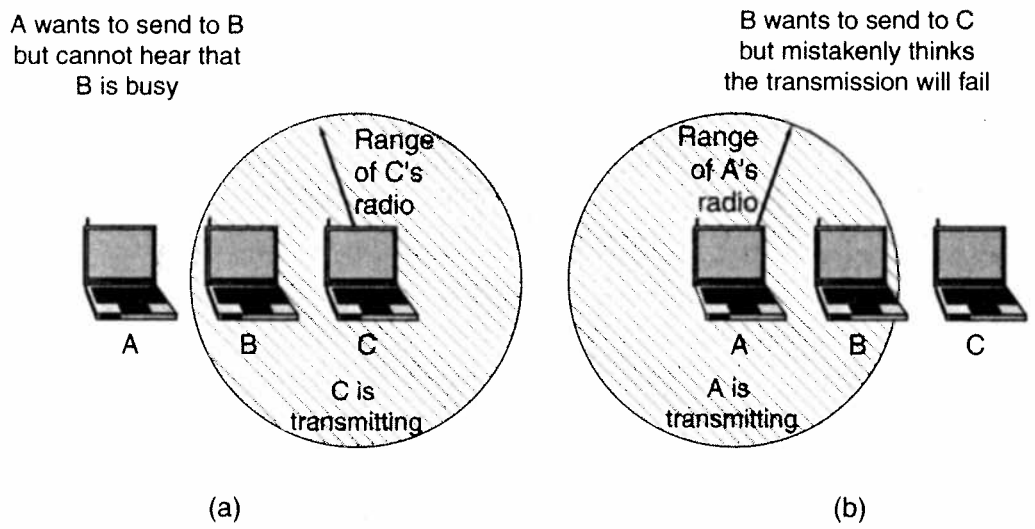

Figure 4-26. (a) The hidden terminal problem. (b) The exposed terminal problem.

transmission, it falsely concludes that it may not send to *C*, even though *A* may in fact be transmitting to *D* (not shown). This decision wastes a transmission opportunity.

To reduce ambiguities about which station is sending, 802.11 defines channel sensing to consist of both physical sensing and virtual sensing. Physical sensing simply checks the medium to see if there is a valid signal. With virtual sensing, each station keeps a logical record of when the channel is in use by tracking the **NAV (Network Allocation Vector)**. Each frame carries a NAV field that says how long the sequence of which this frame is part will take to complete. Stations that overhear this frame know that the channel will be busy for the period indicated by the NAV, regardless of whether they can sense a physical signal. For example, the NAV of a data frame includes the time needed to send an acknowledgement. All stations that hear the data frame will defer during the acknowledgement period, whether or not they can hear the acknowledgement.

An optional RTS/CTS mechanism uses the NAV to prevent terminals from sending frames at the same time as hidden terminals. It is shown in Fig. 4-27. In this example, *A* wants to send to *B*. *C* is a station within range of *A* (and possibly within range of *B*, but that does not matter). *D* is a station within range of *B* but not within range of *A*.

The protocol starts when *A* decides it wants to send data to *B*. *A* begins by sending an RTS frame to *B* to request permission to send it a frame. If *B* receives this request, it answers with a CTS frame to indicate that the channel is clear to send. Upon receipt of the CTS, *A* sends its frame and starts an ACK timer. Upon correct receipt of the data frame, *B* responds with an ACK frame, completing the exchange. If *A*'s ACK timer expires before the ACK gets back to it, it is treated as a collision and the whole protocol is run again after a backoff.

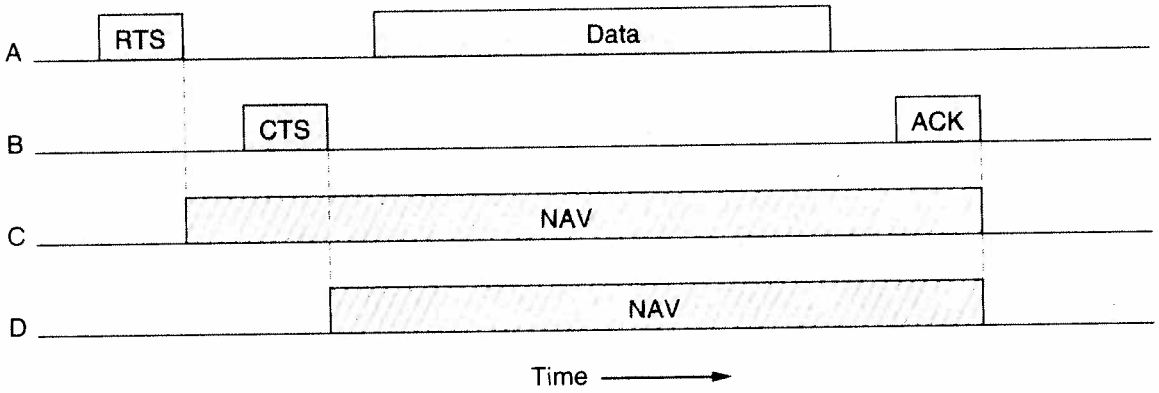

Figure 4-27. Virtual channel sensing using CSMA/CA.

Now let us consider this exchange from the viewpoints of *C* and *D*. *C* is within range of *A*, so it may receive the RTS frame. If it does, it realizes that someone is going to send data soon. From the information provided in the RTS request, it can estimate how long the sequence will take, including the final ACK. So, for the good of all, it desists from transmitting anything until the exchange is completed. It does so by updating its record of the NAV to indicate that the channel is busy, as shown in Fig. 4-27. *D* does not hear the RTS, but it does hear the CTS, so it also updates its NAV. Note that the NAV signals are not transmitted; they are just internal reminders to keep quiet for a certain period of time.

However, while RTS/CTS sounds good in theory, it is one of those designs that has proved to be of little value in practice. Several reasons why it is seldom used are known. It does not help for short frames (which are sent in place of the RTS) or for the AP (which everyone can hear, by definition). For other situations, it only slows down operation. RTS/CTS in 802.11 is a little different than in the MACA protocol we saw in Sec 4.2 because everyone hearing the RTS or CTS remains quiet for the duration to allow the ACK to get through without collision. Because of this, it does not help with exposed terminals as MACA did, only with hidden terminals. Most often there are few hidden terminals, and CSMA/CA already helps them by slowing down stations that transmit unsuccessfully, whatever the cause, to make it more likely that transmissions will succeed.

CSMA/CA with physical and virtual sensing is the core of the 802.11 protocol. However, there are several other mechanisms that have been developed to go with it. Each of these mechanisms was driven by the needs of real operation, so we will look at them briefly.

The first need we will look at is reliability. In contrast to wired networks, wireless networks are noisy and unreliable, in no small part due to interference from other kinds of devices, such as microwave ovens, which also use the unlicensed ISM bands. The use of acknowledgements and retransmissions is of little help if the probability of getting a frame through is small in the first place.

The main strategy that is used to increase successful transmissions is to lower the transmission rate. Slower rates use more robust modulations that are more likely to be received correctly for a given signal-to-noise ratio. If too many frames are lost, a station can lower the rate. If frames are delivered with little loss, a station can occasionally test a higher rate to see if it should be used.

Another strategy to improve the chance of the frame getting through undamaged is to send shorter frames. If the probability of any bit being in error is p, the probability of an n-bit frame being received entirely correctly is $(1-p)^n$. For example, for $p = 10^{-4}$, the probability of receiving a full Ethernet frame (12,144 bits) correctly is less than 30%. Most frames will be lost. But if the frames are only a third as long (4048 bits) two thirds of them will be received correctly. Now most frames will get through and fewer retransmissions will be needed.

Shorter frames can be implemented by reducing the maximum size of the message that is accepted from the network layer. Alternatively, 802.11 allows frames to be split into smaller pieces, called **fragments**, each with its own checksum. The fragment size is not fixed by the standard, but is a parameter that can be adjusted by the AP. The fragments are individually numbered and acknowledged using a stop-and-wait protocol (i.e., the sender may not transmit fragment $k + 1$ until it has received the acknowledgement for fragment k). Once the channel has been acquired, multiple fragments are sent as a burst. They go one after the other with an acknowledgement (and possibly retransmissions) in between, until either the whole frame has been successfully sent or the transmission time reaches the maximum allowed. The NAV mechanism keeps other stations quiet only until the next acknowledgement, but another mechanism (see below) is used to allow a burst of fragments to be sent without other stations sending a frame in the middle.

The second need we will discuss is saving power. Battery life is always an issue with mobile wireless devices. The 802.11 standard pays attention to the issue of power management so that clients need not waste power when they have neither information to send nor to receive.

The basic mechanism for saving power builds on **beacon frames**. Beacons are periodic broadcasts by the AP (e.g., every 100 msec). The frames advertise the presence of the AP to clients and carry system parameters, such as the identifier of the AP, the time, how long until the next beacon, and security settings.

Clients can set a power-management bit in frames that they send to the AP to tell it that they are entering **power-save mode**. In this mode, the client can doze and the AP will buffer traffic intended for it. To check for incoming traffic, the client wakes up for every beacon, and checks a traffic map that is sent as part of the beacon. This map tells the client if there is buffered traffic. If so, the client sends a poll message to the AP, which then sends the buffered traffic. The client can then go back to sleep until the next beacon is sent.

Another power-saving mechanism, called **APSD (Automatic Power Save Delivery)**, was also added to 802.11 in 2005. With this new mechanism, the AP buffers frames and sends them to a client just after the client sends frames to the

AP. The client can then go to sleep until it has more traffic to send (and receive). This mechanism works well for applications such as VoIP that have frequent traffic in both directions. For example, a VoIP wireless phone might use it to send and receive frames every 20 msec, much more frequently than the beacon interval of 100 msec, while dozing in between.

The third and last need we will examine is quality of service. When the VoIP traffic in the preceding example competes with peer-to-peer traffic, the VoIP traffic will suffer. It will be delayed due to contention with the high-bandwidth peer-to-peer traffic, even though the VoIP bandwidth is low. These delays are likely to degrade the voice calls. To prevent this degradation, we would like to let the VoIP traffic go ahead of the peer-to-peer traffic, as it is of higher priority.

IEEE 802.11 has a clever mechanism to provide this kind of quality of service that was introduced as set of extensions under the name 802.11e in 2005. It works by extending CSMA/CA with carefully defined intervals between frames. After a frame has been sent, a certain amount of idle time is required before any station may send a frame to check that the channel is no longer in use. The trick is to define different time intervals for different kinds of frames.

Five intervals are depicted in Fig. 4-28. The interval between regular data frames is called the **DIFS (DCF InterFrame Spacing)**. Any station may attempt to acquire the channel to send a new frame after the medium has been idle for DIFS. The usual contention rules apply, and binary exponential backoff may be needed if a collision occurs. The shortest interval is **SIFS (Short InterFrame Spacing)**. It is used to allow the parties in a single dialog the chance to go first. Examples include letting the receiver send an ACK, other control frame sequences like RTS and CTS, or letting a sender transmit a burst of fragments. Sending the next fragment after waiting only SIFS is what prevents another station from jumping in with a frame in the middle of the exchange.

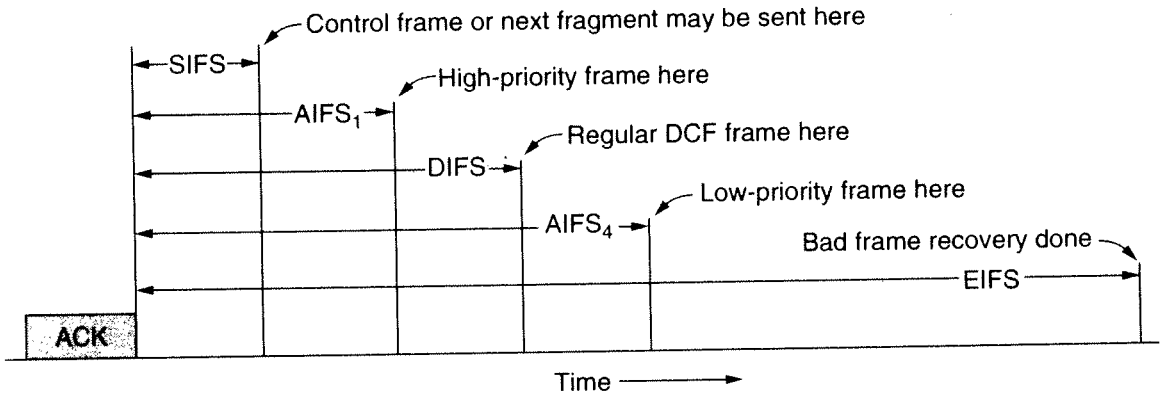

Figure 4-28. Interframe spacing in 802.11.

The two **AIFS (Arbitration InterFrame Space)** intervals show examples of two different priority levels. The short interval, $AIFS_1$, is smaller than DIFS but longer than SIFS. It can be used by the AP to move voice or other high-priority

traffic to the head of the line. The AP will wait for a shorter interval before it sends the voice traffic, and thus send it before regular traffic. The long interval, $AIFS_4$, is larger than DIFS. It is used for background traffic that can be deferred until after regular traffic. The AP will wait for a longer interval before it sends this traffic, giving regular traffic the opportunity to transmit first. The complete quality of service mechanism defines four different priority levels that have different backoff parameters as well as different idle parameters.

The last time interval, **EIFS (Extended InterFrame Spacing)**, is used only by a station that has just received a bad or unknown frame, to report the problem. The idea is that since the receiver may have no idea of what is going on, it should wait a while to avoid interfering with an ongoing dialog between two stations.

A further part of the quality of service extensions is the notion of a **TXOP** or **transmission opportunity**. The original CSMA/CA mechanism let stations send one frame at a time. This design was fine until the range of rates increased. With 802.11a/g, one station might be sending at 6 Mbps and another station be sending at 54 Mbps. They each get to send one frame, but the 6-Mbps station takes nine times as long (ignoring fixed overheads) as the 54-Mbps station to send its frame. This disparity has the unfortunate side effect of slowing down a fast sender who is competing with a slow sender to roughly the rate of the slow sender. For example, again ignoring fixed overheads, when sending alone the 6-Mbps and 54-Mbps senders will get their own rates, but when sending together they will both get 5.4 Mbps on average. It is a stiff penalty for the fast sender. This issue is known as the **rate anomaly** (Heusse et al., 2003).

With transmission opportunities, each station gets an equal amount of airtime, not an equal number of frames. Stations that send at a higher rate for their airtime will get higher throughput. In our example, when sending together the 6-Mbps and 54-Mbps senders will now get 3 Mbps and 27 Mbps, respectively.

4.4.4 The 802.11 Frame Structure

The 802.11 standard defines three different classes of frames in the air: data, control, and management. Each of these has a header with a variety of fields used within the MAC sublayer. In addition, there are some headers used by the physical layer, but these mostly deal with the modulation techniques used, so we will not discuss them here.

We will look at the format of the data frame as an example. It is shown in Fig. 4-29. First comes the *Frame control* field, which is made up of 11 subfields. The first of these is the *Protocol version*, set to 00. It is there to allow future versions of 802.11 to operate at the same time in the same cell. Then come the *Type* (data, control, or management) and *Subtype* fields (e.g., RTS or CTS). For a regular data frame (without quality of service), they are set to 10 and 0000 in binary. The *To DS* and *From DS* bits are set to indicate whether the frame is going to or coming from the network connected to the APs, which is called the distribution

system. The *More fragments* bit means that more fragments will follow. The *Retry* bit marks a retransmission of a frame sent earlier. The *Power management* bit indicates that the sender is going into power-save mode. The *More data* bit indicates that the sender has additional frames for the receiver. The *Protected Frame* bit indicates that the frame body has been encrypted for security. We will discuss security briefly in the next section. Finally, the *Order* bit tells the receiver that the higher layer expects the sequence of frames to arrive strictly in order.

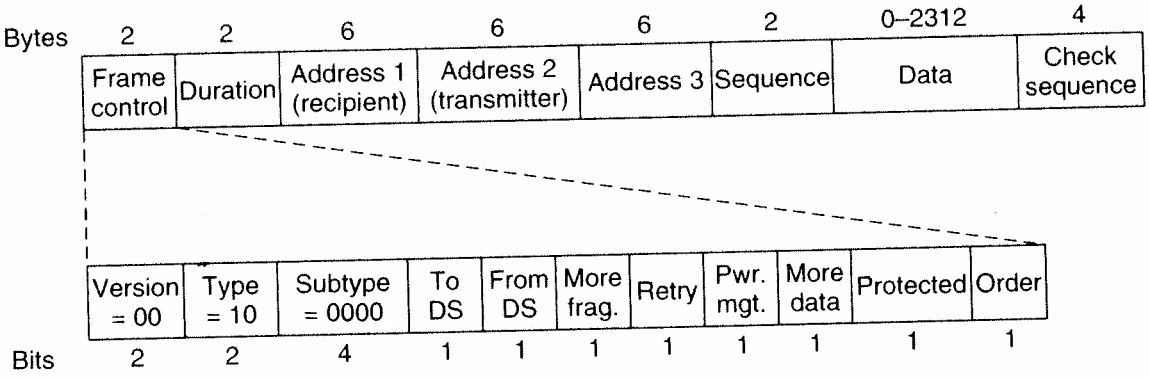

Figure 4-29. Format of the 802.11 data frame.

The second field of the data frame, the *Duration* field, tells how long the frame and its acknowledgement will occupy the channel, measured in microseconds. It is present in all types of frames, including control frames, and is what stations use to manage the NAV mechanism.

Next come addresses. Data frames sent to or from an AP have three addresses, all in standard IEEE 802 format. The first address is the receiver, and the second address is the transmitter. They are obviously needed, but what is the third address for? Remember that the AP is simply a relay point for frames as they travel between a client and another point on the network, perhaps a distant client or a portal to the Internet. The third address gives this distant endpoint.

The *Sequence* field numbers frames so that duplicates can be detected. Of the 16 bits available, 4 identify the fragment and 12 carry a number that is advanced with each new transmission. The *Data* field contains the payload, up to 2312 bytes. The first bytes of this payload are in a format known as **LLC (Logical Link Control)**. This layer is the glue that identifies the higher-layer protocol (e.g., IP) to which the payloads should be passed. Last comes the *Frame check sequence*, which is the same 32-bit CRC we saw in Sec. 3.2.2 and elsewhere.

Management frames have the same format as data frames, plus a format for the data portion that varies with the subtype (e.g., parameters in beacon frames). Control frames are short. Like all frames, they have the *Frame control*, *Duration*, and *Frame check sequence* fields. However, they may have only one address and no data portion. Most of the key information is conveyed with the *Subtype* field (e.g., ACK, RTS and CTS).

4.4.5 Services

The 802.11 standard defines the services that the clients, the access points, and the network connecting them must be a conformant wireless LAN. These services cluster into several groups.

The **association** service is used by mobile stations to connect themselves to APs. Typically, it is used just after a station moves within radio range of the AP. Upon arrival, the station learns the identity and capabilities of the AP, either from beacon frames or by directly asking the AP. The capabilities include the data rates supported, security arrangements, power-saving capabilities, quality of service support, and more. The station sends a request to associate with the AP. The AP may accept or reject the request.

Reassociation lets a station change its preferred AP. This facility is useful for mobile stations moving from one AP to another AP in the same extended 802.11 LAN, like a handover in the cellular network. If it is used correctly, no data will be lost as a consequence of the handover. (But 802.11, like Ethernet, is just a best-effort service.) Either the station or the AP may also **disassociate**, breaking their relationship. A station should use this service before shutting down or leaving the network. The AP may use it before going down for maintenance.

Stations must also **authenticate** before they can send frames via the AP, but authentication is handled in different ways depending on the choice of security scheme. If the 802.11 network is "open," anyone is allowed to use it. Otherwise, credentials are needed to authenticate. The recommended scheme, called **WPA2 (WiFi Protected Access 2)**, implements security as defined in the 802.11i standard. (Plain WPA is an interim scheme that implements a subset of 802.11i. We will skip it and go straight to the complete scheme.) With WPA2, the AP can talk to an authentication server that has a username and password database to determine if the station is allowed to access the network. Alternatively a pre-shared key, which is a fancy name for a network password, may be configured. Several frames are exchanged between the station and the AP with a challenge and response that lets the station prove it has the right credentials. This exchange happens after association.

The scheme that was used before WPA is called **WEP (Wired Equivalent Privacy)**. For this scheme, authentication with a preshared key happens before association. However, its use is discouraged because of design flaws that make WEP easy to compromise. The first practical demonstration that WEP was broken came when Adam Stubblefield was a summer intern at AT&T (Stubblefield et al., 2002). He was able to code up and test an attack in one week, much of which was spent getting permission from management to buy the WiFi cards needed for experiments. Software to crack WEP passwords is now freely available.

Once frames reach the AP, the **distribution** service determines how to route them. If the destination is local to the AP, the frames can be sent out directly over the air. Otherwise, they will have to be forwarded over the wired network. The

integration service handles any translation that is needed for a frame to be sent outside the 802.11 LAN, or to arrive from outside the 802.11 LAN. The common case here is connecting the wireless LAN to the Internet.

Data transmission is what it is all about, so 802.11 naturally provides a **data delivery** service. This service lets stations transmit and receive data using the protocols we described earlier in this chapter. Since 802.11 is modeled on Ethernet and transmission over Ethernet is not guaranteed to be 100% reliable, transmission over 802.11 is not guaranteed to be reliable either. Higher layers must deal with detecting and correcting errors.

Wireless is a broadcast signal. For information sent over a wireless LAN to be kept confidential, it must be encrypted. This goal is accomplished with a **privacy** service that manages the details of encryption and decryption. The encryption algorithm for WPA2 is based on **AES (Advanced Encryption Standard)**, a U.S. government standard approved in 2002. The keys that are used for encryption are determined during the authentication procedure.

To handle traffic with different priorities, there is a **QOS traffic scheduling** service. It uses the protocols we described to give voice and video traffic preferential treatment compared to best-effort and background traffic. A companion service also provides higher-layer timer synchronization. This lets stations coordinate their actions, which may be useful for media processing.

Finally, there are two services that help stations manage their use of the spectrum. The **transmit power control** service gives stations the information they need to meet regulatory limits on transmit power that vary from region to region. The **dynamic frequency selection** service give stations the information they need to avoid transmitting on frequencies in the 5-GHz band that are being used for radar in the proximity.

With these services, 802.11 provides a rich set of functionality for connecting nearby mobile clients to the Internet. It has been a huge success, and the standard has repeatedly been amended to add more functionality. For a perspective on where the standard has been and where it is heading, see Hiertz et al. (2010).

4.5 BROADBAND WIRELESS

We have been indoors too long. Let us go outdoors, where there is quite a bit of interesting networking over the so-called "last mile." With the deregulation of the telephone systems in many countries, competitors to the entrenched telephone companies are now often allowed to offer local voice and high-speed Internet service. There is certainly plenty of demand. The problem is that running fiber or coax to millions of homes and businesses is prohibitively expensive. What is a competitor to do?

The answer is broadband wireless. Erecting a big antenna on a hill just outside of town is much easier and cheaper than digging many trenches and stringing

cables. Thus, companies have begun to experiment with providing multimegabit wireless communication services for voice, Internet, movies on demand, etc.

To stimulate the market, IEEE formed a group to standardize a broadband wireless metropolitan area network. The next number available in the 802 numbering space was **802.16**, so the standard got this number. Informally the technology is called **WiMAX (Worldwide Interoperability for Microwave Access)**. We will use the terms 802.16 and WiMAX interchangeably.

The first 802.16 standard was approved in December 2001. Early versions provided a wireless local loop between fixed points with a line of sight to each other. This design soon changed to make WiMAX a more competitive alternative to cable and DSL for Internet access. By January 2003, 802.16 had been revised to support non-line-of-sight links by using OFDM technology at frequencies between 2 GHz and 10 GHz. This change made deployment much easier, though stations were still fixed locations. The rise of 3G cellular networks posed a threat by promising high data rates *and* mobility. In response, 802.16 was enhanced again to allow mobility at vehicular speeds by December 2005. Mobile broadband Internet access is the target of the current standard, IEEE 802.16-2009.

Like the other 802 standards, 802.16 was heavily influenced by the OSI model, including the (sub)layers, terminology, service primitives, and more. Unfortunately, also like OSI, it is fairly complicated. In fact, the **WiMAX Forum** was created to define interoperable subsets of the standard for commercial offerings. In the following sections, we will give a brief description of some of the highlights of the common forms of 802.16 air interface, but this treatment is far from complete and leaves out many details. For additional information about WiMAX and broadband wireless in general, see Andrews et al. (2007).

4.5.1 Comparison of 802.16 with 802.11 and 3G

At this point you may be thinking: why devise a new standard? Why not just use 802.11 or 3G? In fact, WiMAX combines aspects of both 802.11 and 3G, making it more like a 4G technology.

Like 802.11, WiMAX is all about wirelessly connecting devices to the Internet at megabit/sec speeds, instead of using cable or DSL. The devices may be mobile, or at least portable. WiMAX did not start by adding low-rate data on the side of voice-like cellular networks; 802.16 was designed to carry IP packets over the air and to connect to an IP-based wired network with a minimum of fuss. The packets may carry peer-to-peer traffic, VoIP calls, or streaming media to support a range of applications. Also like 802.11, it is based on OFDM technology to ensure good performance in spite of wireless signal degradations such as multipath fading, and on MIMO technology to achieve high levels of throughput.

However, WiMAX is more like 3G (and thus unlike 802.11) in several key respects. The key technical problem is to achieve high capacity by the efficient use of spectrum, so that a large number of subscribers in a coverage area can all get

high throughput. The typical distances are at least 10 times larger than for an 802.11 network. Consequently, WiMAX base stations are more powerful than 802.11 Access Points (APs). To handle weaker signals over larger distances, the base station uses more power and better antennas, and it performs more processing to handle errors. To maximize throughput, transmissions are carefully scheduled by the base station for each particular subscriber; spectrum use is not left to chance with CSMA/CA, which may waste capacity with collisions.

Licensed spectrum is the expected case for WiMAX, typically around 2.5 GHz in the U.S. The whole system is substantially more optimized than 802.11. This complexity is worth it, considering the large amount of money involved for licensed spectrum. Unlike 802.11, the result is a managed and reliable service with good support for quality of service.

With all of these features, 802.16 most closely resembles the 4G cellular networks that are now being standardized under the name **LTE (Long Term Evolution)**. While 3G cellular networks are based on CDMA and support voice and data, 4G cellular networks will be based on OFDM with MIMO, and they will target data, with voice as just one application. It looks like WiMAX and 4G are on a collision course in terms of technology and applications. Perhaps this convergence is unsurprising, given that the Internet is the killer application and OFDM and MIMO are the best-known technologies for efficiently using the spectrum.

4.5.2 The 802.16 Architecture and Protocol Stack

The 802.16 architecture is shown in Fig. 4-30. Base stations connect directly to the provider's backbone network, which is in turn connected to the Internet. The base stations communicate with stations over the wireless air interface. Two kinds of stations exist. Subscriber stations remain in a fixed location, for example, broadband Internet access for homes. Mobile stations can receive service while they are moving, for example, a car equipped with WiMAX.

The 802.16 protocol stack that is used across the air interface is shown in Fig. 4-31. The general structure is similar to that of the other 802 networks, but with more sublayers. The bottom layer deals with transmission, and here we have shown only the popular offerings of 802.16, fixed and mobile WiMAX. There is a different physical layer for each offering. Both layers operate in licensed spectrum below 11 GHz and use OFDM, but in different ways.

Above the physical layer, the data link layer consists of three sublayers. The bottom one deals with privacy and security, which is far more crucial for public outdoor networks than for private indoor networks. It manages encryption, decryption, and key management.

Next comes the MAC common sublayer part. This part is where the main protocols, such as channel management, are located. The model here is that the base station completely controls the system. It can schedule the downlink (i.e., base to subscriber) channels very efficiently and plays a major role in managing

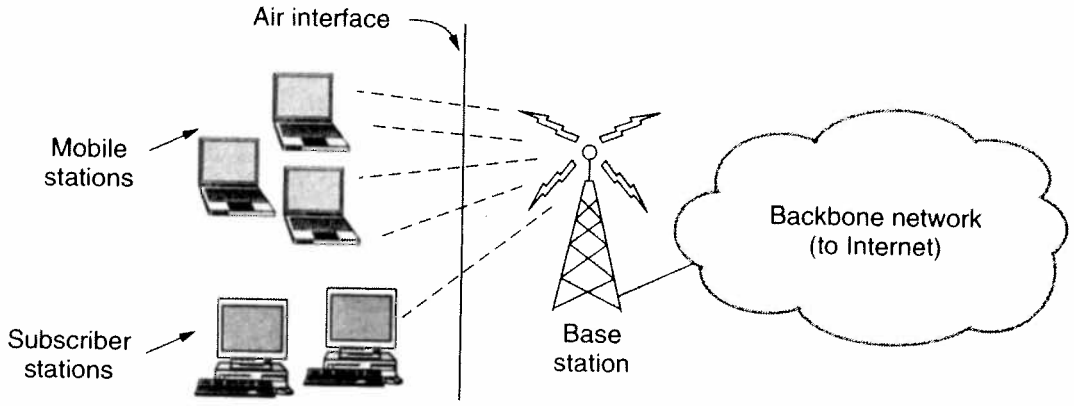

Figure 4-30. The 802.16 architecture.

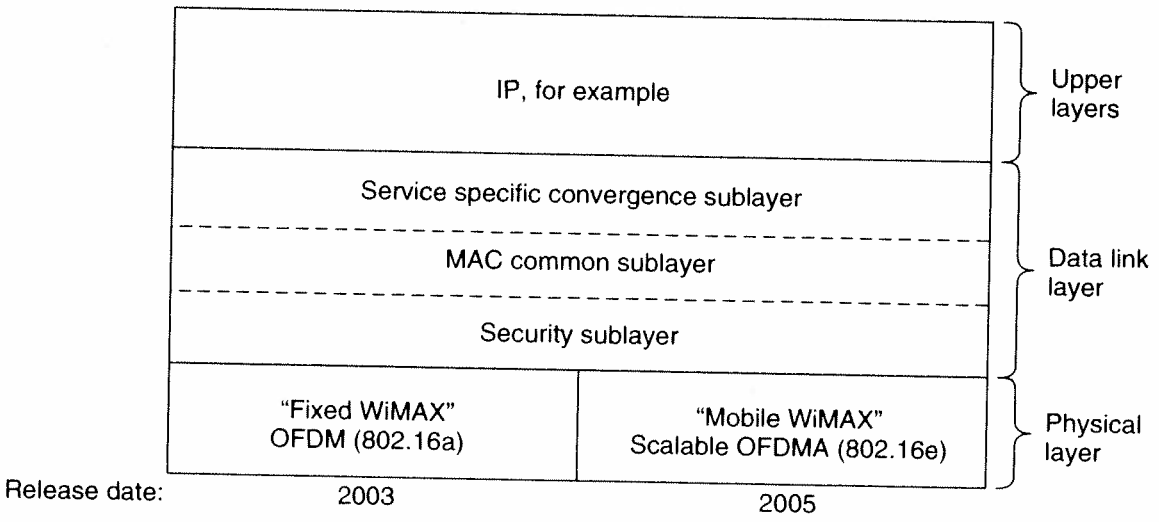

Figure 4-31. The 802.16 protocol stack.

the uplink (i.e., subscriber to base) channels as well. An unusual feature of this MAC sublayer is that, unlike those of the other 802 protocols, it is completely connection oriented, in order to provide quality of service guarantees for telephony and multimedia communication.

The service-specific convergence sublayer takes the place of the logical link sublayer in the other 802 protocols. Its function is to provide an interface to the network layer. Different convergence layers are defined to integrate seamlessly with different upper layers. The important choice is IP, though the standard defines mappings for protocols such as Ethernet and ATM too. Since IP is connectionless and the 802.16 MAC sublayer is connection-oriented, this layer must map between addresses and connections.

4.5.3 The 802.16 Physical Layer

Most WiMAX deployments use licensed spectrum around either 3.5 GHz or 2.5 GHz. As with 3G, finding available spectrum is a key problem. To help, the 802.16 standard is designed for flexibility. It allows operation from 2 GHz to 11 GHz. Channels of different sizes are supported, for example, 3.5 MHz for fixed WiMAX and from 1.25 MHz to 20 MHz for mobile WiMAX.

Transmissions are sent over these channels with OFDM, the technique we described in Sec. 2.5.3. Compared to 802.11, the 802.16 OFDM design is optimized to make the most out of licensed spectrum and wide area transmissions. The channel is divided into more subcarriers with a longer symbol duration to tolerate larger wireless signal degradations; WiMAX parameters are around 20 times larger than comparable 802.11 parameters. For example, in mobile WiMAX there are 512 subcarriers for a 5-MHz channel and the time to send a symbol on each subcarrier is roughly 100 μsec.

Symbols on each subcarrier are sent with QPSK, QAM-16, or QAM-64, modulation schemes we described in Sec. 2.5.3. When the mobile or subscriber station is near the base station and the received signal has a high signal-to-noise ratio (SNR), QAM-64 can be used to send 6 bits per symbol. To reach distant stations with a low SNR, QPSK can be used to deliver 2 bits per symbol. The data is first coded for error correction with the convolutional coding (or better schemes) that we described in Sec. 3.2.1. This coding is common on noisy channels to tolerate some bit errors without needing to send retransmissions. In fact, the modulation and coding methods should sound familiar by now as they are used for many networks we have studied, including 802.11 cable, and DSL. The net result is that a base station can support up to 12.6 Mbps of downlink traffic and 6.2 Mbps of uplink traffic per 5-MHz channel and pair of antennas.

One thing the designers of 802.16 did not like was a certain aspect of the way GSM and DAMPS work. Both of those systems use equal frequency bands for upstream and downstream traffic. That is, they implicitly assume there is as much upstream traffic as downstream traffic. For voice, traffic is symmetric for the most part, but for Internet access (and certainly Web surfing) there is often more downstream traffic than upstream traffic. The ratio is often 2:1, 3:1, or more:1.

So, the designers chose a flexible scheme for dividing the channel between stations, called **OFDMA (Orthogonal Frequency Division Multiple Access)**. With OFDMA, different sets of subcarriers can be assigned to different stations, so that more than one station can send or receive at once. If this were 802.11, all subcarriers would be used by one station to send at any given moment. The added flexibility in how bandwidth is assigned can increase performance because a given subcarrier might be faded at one receiver due to multipath effects but clear at another. Subcarriers can be assigned to the stations that can use them best.

As well as having asymmetric traffic, stations usually alternate between sending and receiving. This method is called **TDD (Time Division Duplex)**. The

alternative method, in which a station sends and receives at the same time (on different subcarrier frequencies), is called **FDD (Frequency Division Duplex)**. WiMAX allows both methods, but TDD is preferred because it is easier to implement and more flexible.

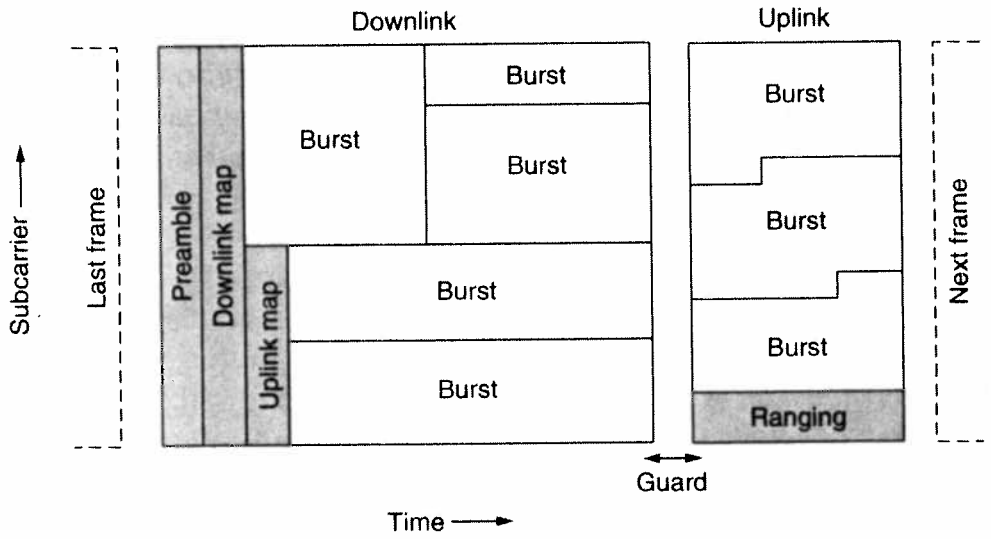

Figure 4-32. Frame structure for OFDMA with time division duplexing.

Fig. 4-32 shows an example of the frame structure that is repeated over time. It starts with a preamble to synchronize all stations, followed by downlink transmissions from the base station. First, the base station sends maps that tell all stations how the downlink and uplink subcarriers are assigned over the frame. The base station controls the maps, so it can allocate different amounts of bandwidth to stations from frame to frame depending on the needs of each station.

Next, the base station sends bursts of traffic to different subscriber and mobile stations on the subcarriers at the times given in the map. The downlink transmissions end with a guard time for stations to switch from receiving to transmitting. Finally, the subscriber and mobile stations send their bursts of traffic to the base station in the uplink positions that were reserved for them in the map. One of these uplink bursts is reserved for **ranging**, which is the process by which new stations adjust their timing and request initial bandwidth to connect to the base station. Since no connection is set up at this stage, new stations just transmit and hope there is no collision.

4.5.4 The 802.16 MAC Sublayer Protocol

The data link layer is divided into three sublayers, as we saw in Fig. 4-31. Since we will not study cryptography until Chap. 8, it is difficult to explain now how the security sublayer works. Suffice it to say that encryption is used to keep secret all data transmitted. Only the frame payloads are encrypted; the headers

are not. This property means that a snooper can see who is talking to whom but cannot tell what they are saying to each other.

If you already know something about cryptography, what follows is a one-paragraph explanation of the security sublayer. If you know nothing about cryptography, you are not likely to find the next paragraph terribly enlightening (but you might consider rereading it after finishing Chap. 8).

When a subscriber connects to a base station, they perform mutual authentication with RSA public-key cryptography using X.509 certificates. The payloads themselves are encrypted using a symmetric-key system, either AES (Rijndael) or DES with cipher block chaining. Integrity checking uses SHA-1. Now that was not so bad, was it?

Let us now look at the MAC common sublayer part. The MAC sublayer is connection-oriented and point-to-multipoint, which means that one base station communicates with multiple subscriber stations. Much of this design is borrowed from cable modems, in which one cable headend controls the transmissions of multiple cable modems at the customer premises.

The downlink direction is fairly straightforward. The base station controls the physical-layer bursts that are used to send information to the different subscriber stations. The MAC sublayer simply packs its frames into this structure. To reduce overhead, there are several different options. For example, MAC frames may be sent individually, or packed back-to-back into a group.

The uplink channel is more complicated since there are competing subscribers that need access to it. Its allocation is tied closely to the quality of service issue. Four classes of service are defined, as follows:

1. Constant bit rate service.
2. Real-time variable bit rate service.
3. Non-real-time variable bit rate service.
4. Best-effort service.

All service in 802.16 is connection-oriented. Each connection gets one of these service classes, determined when the connection is set up. This design is different from that of 802.11 or Ethernet, which are connectionless in the MAC sublayer.

Constant bit rate service is intended for transmitting uncompressed voice. This service needs to send a predetermined amount of data at predetermined time intervals. It is accommodated by dedicating certain bursts to each connection of this type. Once the bandwidth has been allocated, the bursts are available automatically, without the need to ask for each one.

Real-time variable bit rate service is for compressed multimedia and other soft real-time applications in which the amount of bandwidth needed at each instant may vary. It is accommodated by the base station polling the subscriber at a fixed interval to ask how much bandwidth is needed this time.

Non-real-time variable bit rate service is for heavy transmissions that are not real time, such as large file transfers. For this service, the base station polls the subscriber often, but not at rigidly prescribed time intervals. Connections with this service can also use best-effort service, described next, to request bandwidth.

Best-effort service is for everything else. No polling is done and the subscriber must contend for bandwidth with other best-effort subscribers. Requests for bandwidth are sent in bursts marked in the uplink map as available for contention. If a request is successful, its success will be noted in the next downlink map. If it is not successful, the unsuccessful subscriber have to try again later. To minimize collisions, the Ethernet binary exponential backoff algorithm is used.

4.5.5 The 802.16 Frame Structure

All MAC frames begin with a generic header. The header is followed by an optional payload and an optional checksum (CRC), as illustrated in Fig. 4-33. The payload is not needed in control frames, for example, those requesting channel slots. The checksum is (surprisingly) also optional, due to the error correction in the physical layer and the fact that no attempt is ever made to retransmit real-time frames. If no retransmissions will be attempted, why even bother with a checksum? But if there is a checksum, it is the standard IEEE 802 CRC, and acknowledgements and retransmissions are used for reliability.

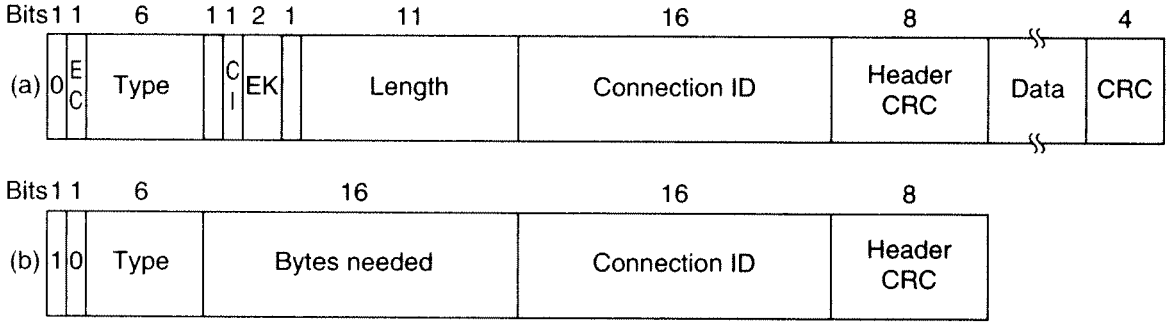

Figure 4-33. (a) A generic frame. (b) A bandwidth request frame.

A quick rundown of the header fields of Fig. 4-33(a) follows. The *EC* bit tells whether the payload is encrypted. The *Type* field identifies the frame type, mostly telling whether packing and fragmentation are present. The *CI* field indicates the presence or absence of the final checksum. The *EK* field tells which of the encryption keys is being used (if any). The *Length* field gives the complete length of the frame, including the header. The *Connection identifier* tells which connection this frame belongs to. Finally, the *Header CRC* field is a checksum over the header only, using the polynomial $x^8 + x^2 + x + 1$.

The 802.16 protocol has many kinds of frames. An example of a different type of frame, one that is used to request bandwidth, is shown in Fig. 4-33(b). It

starts with a 1 bit instead of a 0 bit and is otherwise similar to the generic header except that the second and third bytes form a 16-bit number telling how much bandwidth is needed to carry the specified number of bytes. Bandwidth request frames do not carry a payload or full-frame CRC.

A great deal more could be said about 802.16, but this is not the place to say it. For more information, please consult the IEEE 802.16-2009 standard itself.

4.6 BLUETOOTH

In 1994, the L. M. Ericsson company became interested in connecting its mobile phones to other devices (e.g., laptops) without cables. Together with four other companies (IBM, Intel, Nokia, and Toshiba), it formed a SIG (Special Interest Group, i.e., consortium) in 1998 to develop a wireless standard for interconnecting computing and communication devices and accessories using short-range, low-power, inexpensive wireless radios. The project was named **Bluetooth**, after Harald Blaatand (Bluetooth) II (940–981), a Viking king who unified (i.e., conquered) Denmark and Norway, also without cables.

Bluetooth 1.0 was released in July 1999, and since then the SIG has never looked back. All manner of consumer electronic devices now use Bluetooth, from mobile phones and laptops to headsets, printers, keyboards, mice, gameboxes, watches, music players, navigation units, and more. The Bluetooth protocols let these devices find and connect to each other, an act called **pairing**, and securely transfer data.

The protocols have evolved over the past decade, too. After the initial protocols stabilized, higher data rates were added to Bluetooth 2.0 in 2004. With the 3.0 release in 2009, Bluetooth can be used for device pairing in combination with 802.11 for high-throughput data transfer. The 4.0 release in December 2009 specified low-power operation. That will be handy for people who do not want to change the batteries regularly in all of those devices around the house. We will cover the main aspects of Bluetooth below.

4.6.1 Bluetooth Architecture

Let us start our study of the Bluetooth system with a quick overview of what it contains and what it is intended to do. The basic unit of a Bluetooth system is a **piconet**, which consists of a master node and up to seven active slave nodes within a distance of 10 meters. Multiple piconets can exist in the same (large) room and can even be connected via a bridge node that takes part in multiple piconets, as in Fig. 4-34. An interconnected collection of piconets is called a **scatternet**.

In addition to the seven active slave nodes in a piconet, there can be up to 255 parked nodes in the net. These are devices that the master has switched to a low-power state to reduce the drain on their batteries. In parked state, a device cannot

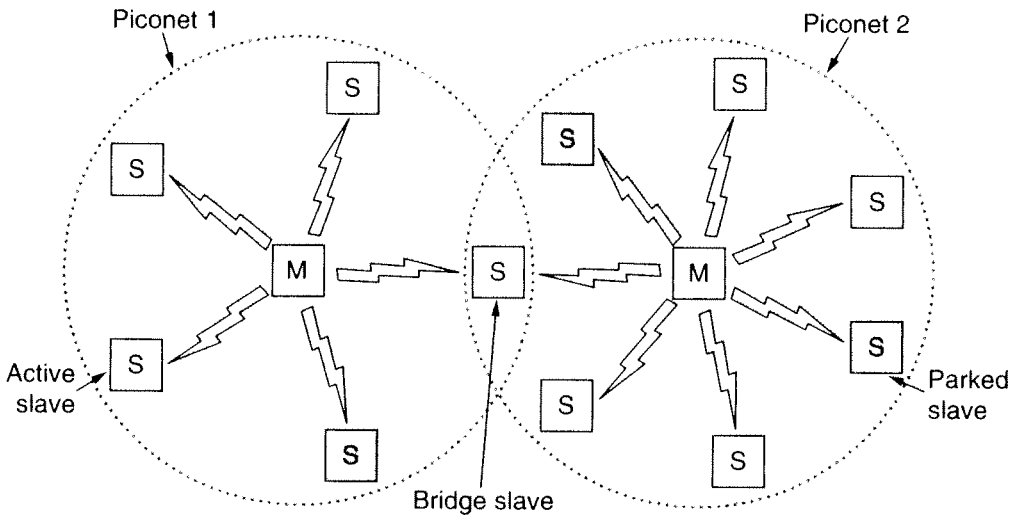

Figure 4-34. Two piconets can be connected to form a scatternet.

do anything except respond to an activation or beacon signal from the master. Two intermediate power states, hold and sniff, also exist, but these will not concern us here.

The reason for the master/slave design is that the designers intended to facilitate the implementation of complete Bluetooth chips for under $5. The consequence of this decision is that the slaves are fairly dumb, basically just doing whatever the master tells them to do. At its heart, a piconet is a centralized TDM system, with the master controlling the clock and determining which device gets to communicate in which time slot. All communication is between the master and a slave; direct slave-slave communication is not possible.

4.6.2 Bluetooth Applications

Most network protocols just provide channels between communicating entities and let application designers figure out what they want to use them for. For example, 802.11 does not specify whether users should use their notebook computers for reading email, surfing the Web, or something else. In contrast, the Bluetooth SIG specifies particular applications to be supported and provides different protocol stacks for each one. At the time of writing, there are 25 applications, which are called **profiles**. Unfortunately, this approach leads to a very large amount of complexity. We will omit the complexity here but will briefly look at the profiles to see more clearly what the Bluetooth SIG is trying to accomplish.

Six of the profiles are for different uses of audio and video. For example, the intercom profile allows two telephones to connect as walkie-talkies. The headset and hands-free profiles both provide voice communication between a headset and its base station, as might be used for hands-free telephony while driving a car.

Other profiles are for streaming stereo-quality audio and video, say, from a portable music player to headphones, or from a digital camera to a TV.

The human interface device profile is for connecting keyboards and mice to computers. Other profiles let a mobile phone or other computer receive images from a camera or send images to a printer. Perhaps of more interest is a profile to use a mobile phone as a remote control for a (Bluetooth-enabled) TV.

Still other profiles enable networking. The personal area network profile lets Bluetooth devices form an ad hoc network or remotely access another network, such as an 802.11 LAN, via an access point. The dial-up networking profile was actually the original motivation for the whole project. It allows a notebook computer to connect to a mobile phone containing a built-in modem without using wires.

Profiles for higher-layer information exchange have also been defined. The synchronization profile is intended for loading data into a mobile phone when it leaves home and collecting data from it when it returns.

We will skip the rest of the profiles, except to mention that some profiles serve as building blocks on which the above profiles are built. The generic access profile, on which all of the other profiles are built, provides a way to establish and maintain secure links (channels) between the master and the slaves. The other generic profiles define the basics of object exchange and audio and video transport. Utility profiles are used widely for functions such as emulating a serial line, which is especially useful for many legacy applications.

Was it really necessary to spell out all these applications in detail and provide different protocol stacks for each one? Probably not, but there were a number of different working groups that devised different parts of the standard, and each one just focused on its specific problem and generated its own profile. Think of this as Conway's Law in action. (In the April 1968 issue of *Datamation* magazine, Melvin Conway observed that if you assign n people to write a compiler, you will get an n-pass compiler, or more generally, the software structure mirrors the structure of the group that produced it.) It would probably have been possible to get away with two protocol stacks instead of 25, one for file transfer and one for streaming real-time communication.

4.6.3 The Bluetooth Protocol Stack

The Bluetooth standard has many protocols grouped loosely into the layers shown in Fig. 4-35. The first observation to make is that the structure does not follow the OSI model, the TCP/IP model, the 802 model, or any other model.

The bottom layer is the physical radio layer, which corresponds fairly well to the physical layer in the OSI and 802 models. It deals with radio transmission and modulation. Many of the concerns here have to do with the goal of making the system inexpensive so that it can become a mass-market item.

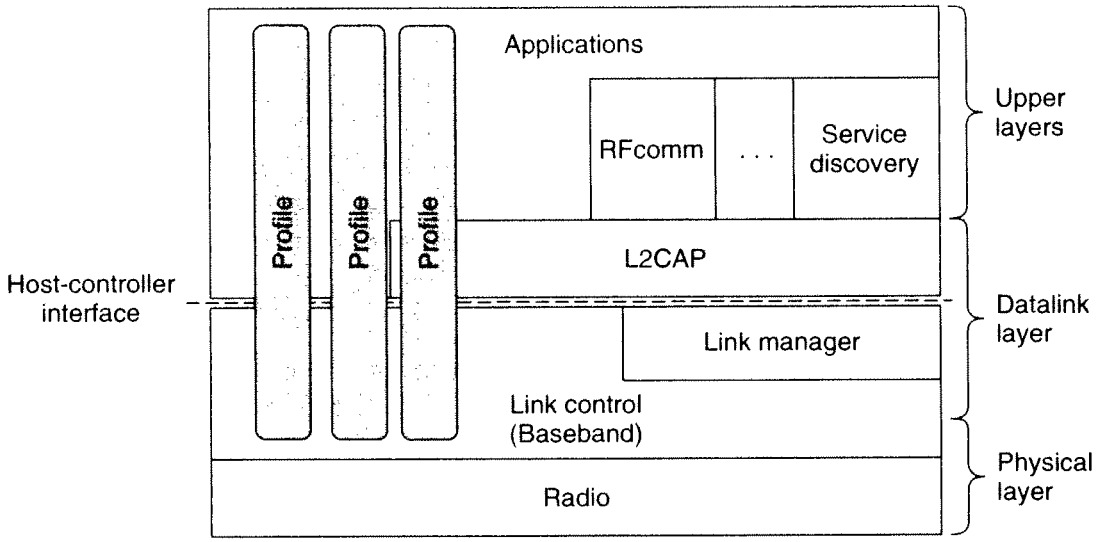

Figure 4-35. The Bluetooth protocol architecture.

The link control (or baseband) layer is somewhat analogous to the MAC sublayer but also includes elements of the physical layer. It deals with how the master controls time slots and how these slots are grouped into frames.

Next come two protocols that use the link control protocol. The link manager handles the establishment of logical channels between devices, including power management, pairing and encryption, and quality of service. It lies below the host controller interface line. This interface is a convenience for implementation: typically, the protocols below the line will be implemented on a Bluetooth chip, and the protocols above the line will be implemented on the Bluetooth device that hosts the chip.

The link protocol above the line is **L2CAP (Logical Link Control Adaptation Protocol)**. It frames variable-length messages and provides reliability if needed. Many protocols use L2CAP, such as the two utility protocols that are shown. The service discovery protocol is used to locate services within the network. The RFcomm (Radio Frequency communication) protocol emulates the standard serial port found on PCs for connecting the keyboard, mouse, and modem, among other devices.

The top layer is where the applications are located. The profiles are represented by vertical boxes because they each define a slice of the protocol stack for a particular purpose. Specific profiles, such as the headset profile, usually contain only those protocols needed by that application and no others. For example, profiles may include L2CAP if they have packets to send but skip L2CAP if they have only a steady flow of audio samples.

In the following sections, we will examine the Bluetooth radio layer and various link protocols, since these roughly correspond to the physical and MAC sublayers in the other procotol stacks we have studied.

4.6.4 The Bluetooth Radio Layer

The radio layer moves the bits from master to slave, or vice versa. It is a low-power system with a range of 10 meters operating in the same 2.4-GHz ISM band as 802.11. The band is divided into 79 channels of 1 MHz each. To coexist with other networks using the ISM band, frequency hopping spread spectrum is used. There can be up to 1600 hops/sec over slots with a dwell time of 625 μsec. All the nodes in a piconet hop frequencies simultaneously, following the slot timing and pseudorandom hop sequence dictated by the master.

Unfortunately, it turned out that early versions of Bluetooth and 802.11 interfered enough to ruin each other's transmissions. Some companies responded by banning Bluetooth altogether, but eventually a technical solution was devised. The solution is for Bluetooth to adapt its hop sequence to exclude channels on which there are other RF signals. This process reduces the harmful interference. It is called **adaptive frequency hopping**.

Three forms of modulation are used to send bits on a channel. The basic scheme is to use frequency shift keying to send a 1-bit symbol every microsecond, giving a gross data rate of 1 Mbps. Enhanced rates were introduced with the 2.0 version of Bluetooth. These rates use phase shift keying to send either 2 or 3 bits per symbol, for gross data rates of 2 or 3 Mbps. The enhanced rates are only used in the data portion of frames.

4.6.5 The Bluetooth Link Layers

The link control (or baseband) layer is the closest thing Bluetooth has to a MAC sublayer. It turns the raw bit stream into frames and defines some key formats. In the simplest form, the master in each piconet defines a series of 625-μsec time slots, with the master's transmissions starting in the even slots and the slaves' transmissions starting in the odd ones. This scheme is traditional time division multiplexing, with the master getting half the slots and the slaves sharing the other half. Frames can be 1, 3, or 5 slots long. Each frame has an overhead of 126 bits for an access code and header, plus a settling time of 250–260 μsec per hop to allow the inexpensive radio circuits to become stable. The payload of the frame can be encrypted for confidentiality with a key that is chosen when the master and slave connect. Hops only happen between frames, not during a frame. The result is that a 5-slot frame is much more efficient than a 1-slot frame because the overhead is constant but more data is sent.

The link manager protocol sets up logical channels, called **links**, to carry frames between the master and a slave device that have discovered each other. A pairing procedure is followed to make sure that the two devices are allowed to communicate before the link is used. The old pairing method is that both devices must be configured with the same four-digit PIN (Personal Identification Number). The matching PIN is how each device would know that it was connecting to

the right remote device. However, unimaginative users and devices default to PINs such as "0000" and "1234" meant that this method provided very little security in practice.

The new **secure simple pairing** method enables users to confirm that both devices are displaying the same passkey, or to observe the passkey on one device and enter it into the second device. This method is more secure because users do not have to choose or set a PIN. They merely confirm a longer, device-generated passkey. Of course, it cannot be used on some devices with limited input/output, such as a hands-free headset.

Once pairing is complete, the link manager protocol sets up the links. Two main kinds of links exist to carry user data. The first is the **SCO (Synchronous Connection Oriented)** link. It is used for real-time data, such as telephone connections. This type of link is allocated a fixed slot in each direction. A slave may have up to three SCO links with its master. Each SCO link can transmit one 64,000-bps PCM audio channel. Due to the time-critical nature of SCO links, frames sent over them are never retransmitted. Instead, forward error correction can be used to increase reliability.

The other kind is the **ACL (Asynchronous ConnectionLess)** link. This type of link is used for packet-switched data that is available at irregular intervals. ACL traffic is delivered on a best-effort basis. No guarantees are given. Frames can be lost and may have to be retransmitted. A slave may have only one ACL link to its master.

The data sent over ACL links come from the L2CAP layer. This layer has four major functions. First, it accepts packets of up to 64 KB from the upper layers and breaks them into frames for transmission. At the far end, the frames are reassembled into packets. Second, it handles the multiplexing and demultiplexing of multiple packet sources. When a packet has been reassembled, the L2CAP layer determines which upper-layer protocol to hand it to, for example, RFcomm or service discovery. Third, L2CAP handles error control and retransmission. It detects errors and resends packets that were not acknowledged. Finally, L2CAP enforces quality of service requirements between multiple links.

4.6.6 The Bluetooth Frame Structure

Bluetooth defines several frame formats, the most important of which is shown in two forms in Fig. 4-36. It begins with an access code that usually identifies the master so that slaves within radio range of two masters can tell which traffic is for them. Next comes a 54-bit header containing typical MAC sublayer fields. If the frame is sent at the basic rate, the data field comes next. It has up to 2744 bits for a five-slot transmission. For a single time slot, the format is the same except that the data field is 240 bits.

If the frame is sent at the enhanced rate, the data portion may have up to two or three times as many bits because each symbol carries 2 or 3 bits instead of 1

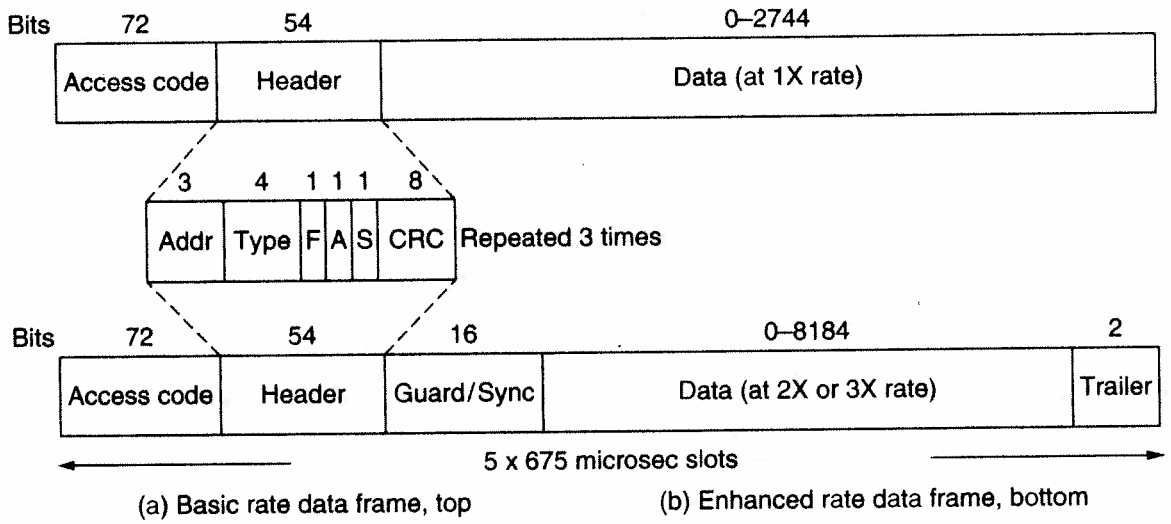

Figure 4-36. Typical Bluetooth data frame at (a) basic and (b) enhanced, data rates.

bit. These data are preceded by a guard field and a synchronization pattern that is used to switch to the faster data rate. That is, the access code and header are carried at the basic rate and only the data portion is carried at the faster rate. Enhanced-rate frames end with a short trailer.

Let us take a quick look at the common header. The *Address* field identifies which of the eight active devices the frame is intended for. The *Type* field identifies the frame type (ACL, SCO, poll, or null), the type of error correction used in the data field, and how many slots long the frame is. The *Flow* bit is asserted by a slave when its buffer is full and cannot receive any more data. This bit enables a primitive form of flow control. The *Acknowledgement* bit is used to piggyback an ACK onto a frame. The *Sequence* bit is used to number the frames to detect retransmissions. The protocol is stop-and-wait, so 1 bit is enough. Then comes the 8-bit header *Checksum*. The entire 18-bit header is repeated three times to form the 54-bit header shown in Fig. 4-36. On the receiving side, a simple circuit examines all three copies of each bit. If all three are the same, the bit is accepted. If not, the majority opinion wins. Thus, 54 bits of transmission capacity are used to send 10 bits of header. The reason is that to reliably send data in a noisy environment using cheap, low-powered (2.5 mW) devices with little computing capacity, a great deal of redundancy is needed.

Various formats are used for the data field for ACL and SCO frames. The basic-rate SCO frames are a simple example to study: the data field is always 240 bits. Three variants are defined, permitting 80, 160, or 240 bits of actual payload, with the rest being used for error correction. In the most reliable version (80-bit payload), the contents are just repeated three times, the same as the header.

We can work out the capacity with this frame as follows. Since the slave may use only the odd slots, it gets 800 slots/sec, just as the master does. With an 80-bit

payload, the channel capacity from the slave is 64,000 bps as is the channel capacity from the master. This capacity is exactly enough for a single full-duplex PCM voice channel (which is why a hop rate of 1600 hops/sec was chosen). That is, despite a raw bandwidth of 1 Mbps, a single full-duplex uncompressed voice channel can completely saturate the piconet. The efficiency of 13% is the result of spending 41% of the capacity on settling time, 20% on headers, and 26% on repetition coding. This shortcoming highlights the value of the enhanced rates and frames of more than a single slot.

There is much more to be said about Bluetooth, but no more space to say it here. For the curious, the Bluetooth 4.0 specification contains all the details.

4.7 RFID

We have looked at MAC designs from LANs up to MANs and down to PANs. As a last example, we will study a category of low-end wireless devices that people may not recognize as forming a computer network: the **RFID (Radio Frequency IDentification)** tags and readers that we described in Sec. 1.5.4.

RFID technology takes many forms, used in smartcards, implants for pets, passports, library books, and more. The form that we will look at was developed in the quest for an **EPC (Electronic Product Code)** that started with the Auto-ID Center at the Massachusetts Institute of Technology in 1999. An EPC is a replacement for a barcode that can carry a larger amount of information and is electronically readable over distances up to 10 m, even when it is not visible. It is different technology than, for example, the RFID used in passports,which must be placed quite close to a reader to perform a transaction. The ability to communicate over a distance makes EPCs more relevant to our studies.

EPCglobal was formed in 2003 to commercialize the RFID technology developed by the Auto-ID Center. The effort got a boost in 2005 when Walmart required its top 100 suppliers to label all shipments with RFID tags. Widespread deployment has been hampered by the difficulty of competing with cheap printed barcodes, but new uses, such as in drivers licenses, are now growing. We will describe the second generation of this technology, which is informally called **EPC Gen 2** (EPCglobal, 2008).

4.7.1 EPC Gen 2 Architecture

The architecture of an EPC Gen 2 RFID network is shown in Fig. 4-37. It has two key components: tags and readers. RFID tags are small, inexpensive devices that have a unique 96-bit EPC identifier and a small amount of memory that can be read and written by the RFID reader. The memory might be used to record the location history of an item, for example, as it moves through the supply chain.