CHAPTER 9

Related Technologies

ZigBee is not the only standard that adopts the IEEE 802.15.4 PHY and MAC layers to offer a wireless networking protocol. This chapter provides an overview of two other standards (6LoWPAN and WirelessHART) that reuse the IEEE 802.15.4 PHY and MAC layers as part of their wireless networking protocol. This chapter also reviews the basics of two wireless networking standards that are not based on IEEE 802.15.4 and compete with ZigBee in some application scenarios (Z-wave and ULP Bluetooth).

9.1 IPv6 over IEEE 802.15.4 (6LoWPAN)

The Internet Protocol (IP) version 6 (IPv6) [1] is a protocol developed by the Internet Engineering Task Force (IETF) [2]. IP is a network layer protocol for communication of data packets in a wired or wireless network. IP is the protocol used in the public Internet and many commercial networks. IPv6 replaces IP version 4 (IPv4). One of the major improvements in IPv6 compared to IPv4 is the address space. IPv4 supports 32-bit addressing, which translates to approximately 4.3 billion unique addresses. Although this might seem like a large address space, it will not be sufficient in the near future based on the expected growth rate of the nodes that will be connected to the Internet around the globe. IPv6 fixed this issue by supporting 128-bit addressing instead of 32-bit addressing in IPv4. This significant increase in the address space will ensure availability of unique addresses for all nodes in any practical growth-rate scenario.

In implementing a wireless sensor network, if the packet format is kept the same as the IP packet format, the interface between the wireless sensor network and the Internet can become simpler. This was the motivation behind development of the 6LoWPAN (IPv6 over low-power WPAN) standard. The 6LoWPAN is a standard that allows transmission of IPv6 packets over an IEEE 802.15.4 network.

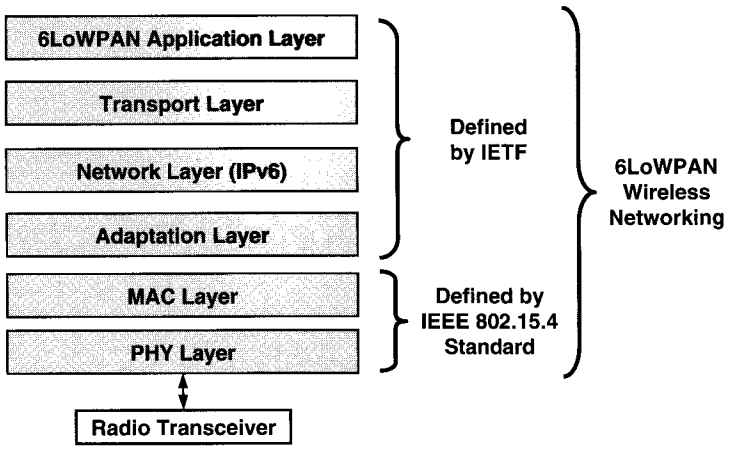

Figure 9.1: Protocol Layers in 6LoWPAN

Figure 9.1 shows the protocol layers in 6LoWPAN. The PHY and MAC layers are defined by IEEE 802.15.4. The packet format is different in IEEE 802.15.4 and IPv6 standards. The adaptation layer is created on top of the MAC layer to adapt IEEE 802.15.4 packets to IPv6, and vice versa. The next higher layer in 6LoWPAN is the IPv6 Network layer, which is compatible with any other IPv6 network regardless of its physical layer. The protocol layers above MAC are defined by IETF. The protocol layers in Figure 9.1 indicate that the user at the application layer always receives and transmits IPv6-compliant packets, whereas the packets transmitted over air are in IEEE 802.15.4 format. The 6LoWPAN standard, similar to ZigBee, supports mesh networking.

The 6LoWPAN standard is developed specifically for the nodes that have limited memory space and processing capabilities. The IPv6 requires support of packet sizes much larger than the largest IEEE 802.15.4 packet size. The size of the header in IPv6 is 40 octets. The 6LoWPAN uses a header compression method to reduce the size of the IPv6 packet header. But this compression is not sufficient to bring the size of IPv6 packets anywhere close to IEEE 802.15.4 packets. The maximum physical layer packet size in IEEE 802.15.4 is 127 octets (*aMaxPHYPacketSize*). The maximum size of the payload is less than 127 octets because of all necessary overheads in a packet. The *minimum* IPv6 packet size is 1280 octets, which is much larger than the *maximum* packet size in IEEE 802.15.4. Therefore, the IPv6 packets need to be fragmented by the transmitter and reassembled by the receiver to accommodate the requirements of both standards. The adaptation layer in 6LoWPAN is responsible for fragmenting and reassembling the packets.

One of the expected advantages of 6LoWPAN is the interoperability of the network with all other IP network links (wired and wireless). A 6LoWPAN node is capable of communicating with other IP-enabled devices. A wireless node that implements 6LoWPAN can be accessed and managed similarly to any other IP device. A user who is familiar with IPv6 can use tools and resources developed for IPv6 to implement its wireless sensor networking application while minimizing the interactions with lower layers of the protocol (IEEE 802.15.4).

6LoWPAN supports both 64-bit extended addressing and 16-bit short addressing in IEEE 802.15.4, but 6LoWPAN imposes additional constraints (beyond IEEE 802.15.4) on the format of the 16-bit short addressing. For example, in multicasting, the first three bits of the short address must be 100. This leaves 13 bits for the actual multicast address in a 6LoWPAN network. ZigBee, in contrast, uses all 16 bits for addressing.

6LoWPAN provides an alternative way of implementing a wireless network. The battery life in the 6LoWPAN and ZigBee standards should be comparable because of the similarities of their hardware and bottom two layers of their protocols. This is based on the assumption that both standards have comparable routing efficiency and are tested in similar use-case scenarios. The decision to select one standard versus another is determined by the target application. Consider an application for which there is no need to interface with IP-enabled devices and the average size of the packets is small. In this case, it is not necessary to implement 6LoWPAN, which performs fragmentation and reassembly of the packets to ensure their compatibility with IPv6.

9.2 WirelessHART

Highway Addressable Remote Transducer (HART) is a communications protocol for applications such as process control, equipment and process monitoring, advanced diagnostics, and closed loop control in wired industrial networks. HART supports a data rate of 1.2 Kbps using FSK modulation. HART uses a master/slave mechanism, and a slave device only transmits data when it is asked by a master device. HART is widely used in process control applications, but it is limited to wired networks.

WirelessHART is a wireless networking standard based on HART that adds wireless flexibility to an existing HART network. WirelessHART operates at the 2.4 GHz ISM band and is backward compatible with existing HART devices, commands, and tools [3]. WirelessHART supports mesh networking. For security, WirelessHART uses AES-128 block ciphers similar to the ZigBee standard.

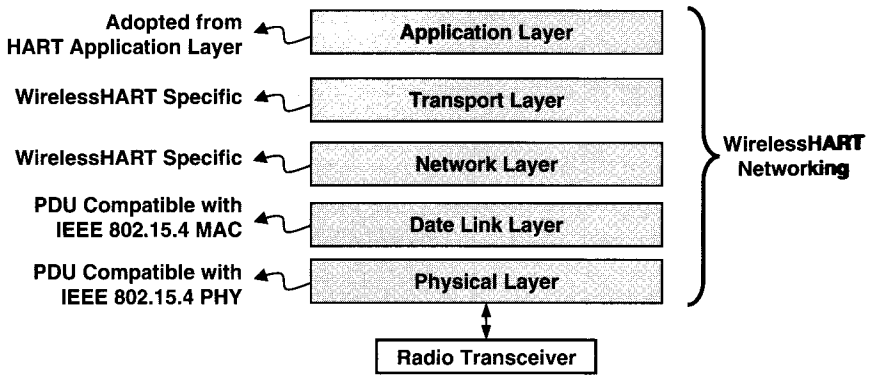

Figure 9.2: Protocol Layers in WirelessHART

The WirelessHART standard defines the protocol layers shown in Figure 9.2. The P layer of WirelessHART uses packet data units (PDUs) that are compatible with IEE 802.15.4 PHY at 2.4 GHz. WirelessHART uses the same frequency channels as IEE 802.15.4 with O-QPSK/DSSS modulation and supports a data rate of 250 Kbps. The difference between WirelessHART PHY and IEEE 802.15.4 PHY is that WirelessHA PHY hops over 16 channels defined by IEEE 802.15.4 on a packet-by-packet basis. WirelessHART can avoid hopping into certain channels by placing these channels in "blacklist." WirelessHART also performs CCA before each transmission to avoid co or experiencing interferences. WirelessHART can be implemented on the commerci available radios developed for the IEEE 802.15.4 standard because WirelessHART a IEEE 802.15.4 PHYs are compatible. The nominal transmitted power in WirelessHA is 10 dBm compared to 0 dBm (typical) in a ZigBee network.

The PDU in the Data Link layer of WirelessHART is compatible with IEEE 802.15.4 MAC PDU. WirelessHART uses a superframe to provide TDMA for communicatio between network devices. There are 100 time slots per second in WirelessHART. The nodes in WirelessHART may have dedicated time slots or use a contention-based ch access mechanism. The network layer is capable of mesh networking and supports broadcast, multicast, and unicast transmissions. The devices in the network maintain a record that includes information such as received signal strength from neighbors and t list of discovered neighbor devices.

The transport layer supports both acknowledged and unacknowledged communicatio and performs automatic retries a limited number of times if the initial data transmissio

is not successful. The WirelessHART application layer is based on the HART application layer to ensure compatibility of WirelessHART nodes with a HART network.

Both ZigBee and WirelessHART can be used in industrial monitoring and control applications. If a wired HART network is present, WirelessHART can be a better choice than ZigBee because of backward compatibility with the HART network. For typical wireless sensor networking applications where no interface with the HART network is required, ZigBee can be used instead of WirelessHART.

9.3 Z-wave

Z-wave is a wireless networking protocol developed by Z-wave alliance for 900MHz ISM band operation [4]. Unlike ZigBee, Z-wave defines all protocol layers and does not adopt IEEE 802.15.4 PHY and MAC layers. Z-wave supports 9.6 Kbps and 40 Kbps data rates using frequency shift keying (FSK) modulation. The Z-wave signals are narrowband, and no spreading method such as DSSS or FHSS is used to help Z-wave mitigate the presence of interferences and multipath nulls.

One of the differentiating factors between ZigBee and Z-wave is address space. ZigBee supports 64-bit and 16-bit addressing, whereas Z-wave supports only 8-bit addressing. Therefore, in a single Z-wave network, there can be up to 232 nodes. This can be sufficient in many applications. However, if a larger number of nodes is required, a ZigBee network can be a better alternative because even in 16-bit short addressing, there can be up to 65,536 nodes in a single ZigBee network. In Z-wave, each network is identified by a 32-bit value called *HomeID*.

Z-wave, similar to ZigBee, supports mesh networking, broadcasting, and multicasting. The collision avoidance in a Z-wave network is achieved by making sure the channel is available before each transmission. If the channel is not available, the node will use a random back-off mechanism between transmission attempts.

For security, Z-wave relays on the Triple Data Encryption Standard (TDES) [5]. The Data Encryption Standard (DES) is considered to be insecure for many applications due to small key size (56 bits). Triple DES uses DES three times to improve security. DES is superseded by the Advanced Encryption Standard (AES). The size of the key in AES can be 128 bits. ZigBee uses AES-128 to ensure communication security in the network.

In summary, a ZigBee network operating in the same sub-GHz frequency band as Z-wave can support higher (or comparable) data rates, a larger number of nodes, and a superior

security method than a Z-wave network. Z-wave nodes have lower complexity and have the potential to cost less than a comparable ZigBee node.

9.4 Ultra-Low-Power Bluetooth (Wibree)

The Ultra-Low-Power (ULP) Bluetooth standard [6], originally known as Wibree [7], is a short-range wireless networking standard developed for point-to-point and very low-duty-cycle wireless communications. The ULP is a simplified version of the Bluetooth standard that expects to have an order of magnitude longer battery life compared to a typical Bluetooth device. For example, ULP has only one packet type, whereas Bluetooth has 28 packet types. ULP does not support mesh networking and therefore does not compete with ZigBee in applications that require mesh networking. ULP and ZigBee compete in short-range point-to-point wireless networking applications. ULP operates in the 2.4 GHz ISM band and defines 40 channels with 2 MHz channel spacing. The signal bandwidth is 1 MHz.

The Bluetooth standard is a well-established wireless networking protocol. ULP-enabled devices can communicate with a Bluetooth-enabled device only if the Bluetooth-enabled device has implemented a dual-mode Bluetooth/ULP protocol stack (see Figure 9.3).

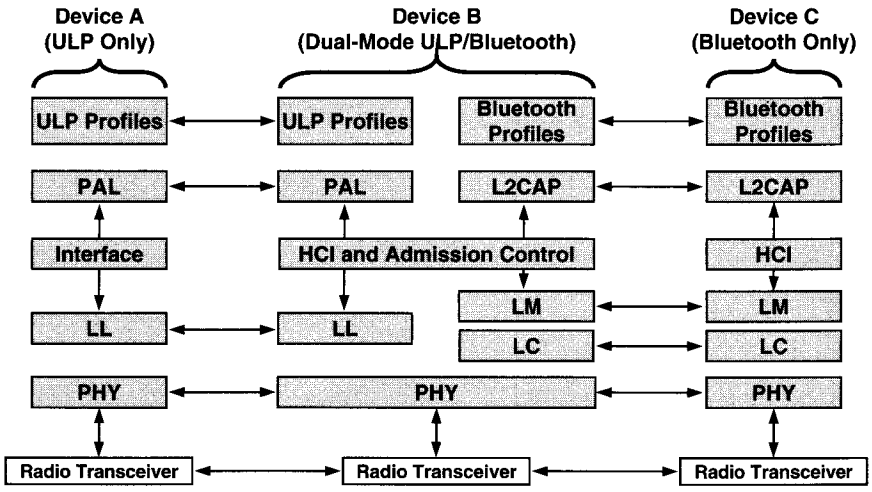

Figure 9.3: Interactions between Protocol Layers in ULP, Dual-Mode ULP/ Bluetooth, and Bluetooth Devices

This dual-mode protocol stack allows the dual-mode device to communicate with both traditional Bluetooth devices and ULP nodes. The common features between ULP and traditional Bluetooth standards simplify the implementation of the dual-mode protocol stack. The cost (e.g., additional memory space) of upgrading a Bluetooth device to a dual-mode ULP/Bluetooth device should be small.

In Figure 9.3, device A contains only a single-mode ULP stack. The PHY layer in ULP is the same as traditional Bluetooth. The next higher layer above PHY is the Link Layer (LL). In Bluetooth there is a separate link controller (LC) and a link manager (LM). The ULP replaces both the LC and LM with a simple link layer (LL). The next upper layer in ULP is PAL, which is a subset of the Logical Link Control and Adaptation Protocol (L2CAP) available in Bluetooth. The quality of service (QoS), segmentation, and reassembly of packets are examples of duties performed by the L2CAP. The ULP, similar to ZigBee, does not support QoS. The protocol layers in device A can communicate with their corresponding layers in a dual-mode device (device B). Device C has only a Bluetooth stack and cannot communicate directly with device A. In the Bluetooth stack, there is a host controller interface (HCI) below L2CAP. The ULP uses a simpler interface mechanism.

9.5 TinyOS

An operating system (OS) in a large network provides an interface for a user (e.g., a programmer) to manage the resources in a network. Typical operating systems that are developed for low-power wireless sensor networks may require large memory space and high-performance microprocessors, which can be beyond the capabilities of resource-limited wireless sensor nodes. TinyOS is an open-source operating system from the TinyOS alliance that's specifically developed for low-power and resource-limited sensor networks in which the nodes spend the majority of their time in sleep mode [8].

TinyOS can be implemented as the operating system in a wireless sensor network where the PHY and MAC layers are defined by the IEEE 802.15.4 standard and the NWK and APL are implemented using the ZigBee standard. TinyOS is not limited to ZigBee networking and can be implemented as an operating system in other wireless sensor networking protocols as well. TinyOS provides a common programming environment for users. In ZigBee applications the users may choose to develop their own application-specific codes without using TinyOS. TinyOS uses the network embedded systems C (nesC) to develop the applications. nesC is a programming language developed specifically for battery-powered nodes with low memory capacity and processing capabilities.

References

[1] Ipv6 over IEEE 802.15.4 (6LoWPAN), available at http://6lowpan.net/.
[2] Internet Engineering Task Force (IETF), available at www.ietf.org.
[3] WirelessHART, available at www.hartcomm.org.
[4] Z-wave Alliance, available at www.z-wavealliance.org.
[5] Data Encryption Standard (DES), Federal Information Processing Standard Publication (FIPS PUB) 46-3, Oct. 1999, available at http://csrc.nist.gov.
[6] Bluetooth Standard, available at www.Bluetooth.com.
[7] Wibree, available at www.wibree.com.
[8] TinyOS Alliance, available at www.tinyOS.net.