15. Sensor Network Standards

Stefano Chessa, Dipartimento di Informatica, University of Pisa, Pisa, Italy.

15.1. Introduction

The standardization of wireless sensor network proceeds along two main directives: the IEEE 802.15.4 standard [1] and ZigBee [2]. As shown in Figure 1 these two standards specify different subsets of layers: the IEEE 802.15.4 defines the physical and Medium Access Control (MAC) layers, and ZigBee defines the network and application layers. The two protocol stacks can be combined to support low data rate, long lasting applications on battery powered, wireless devices. Application fields of these standards include sensors, interactive toys, smart badges, remote controls, and home automation.

The first release of the IEEE 802.15.4 was delivered in 2003 and it is freely distributed in [3]. This standard was revised in 2006 but the new release is not yet freely distributed. The ZigBee protocol stack was proposed at the end of 2004 by the ZigBee Alliance, an association of companies working together to develop standards (and products) for reliable, cost-effective, low-power wireless networking. The first release of ZigBee have been revised at the end of 2006 (both releases can be freely downloaded from [4]). The 2006 version introduces extensions relating the standardization of application profiles and some minor improvements to the network and application layers. In this chapter we will focus to the main functionalities shared by the two releases. A survey of the IEEE 802.15.4 and ZigBee standards against the research state of the art can be found in [5].

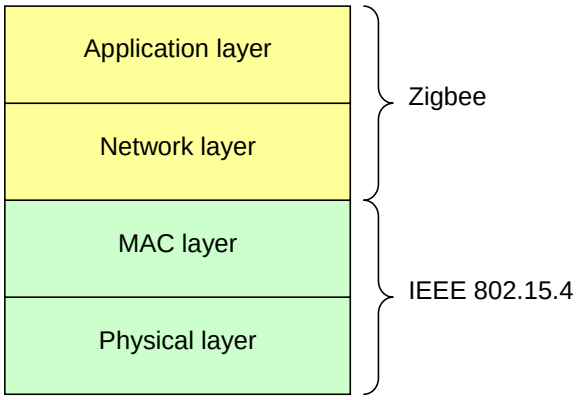

Figure 1. The protocol stack of the IEEE 802.15.4 and ZigBee standards.

This chapter presents the main features of the two standards with a bottom up approach. Section 15.2 introduces the physical and MAC layers of IEEE 802.15.4 and Section 15.3 presents the network and application layers of ZigBee. Section 15.4 draws the conclusions.

15.2. The IEEE 802.15.4 standard

The IEEE 802.15.4 standard [1] specifies the physical and MAC layers for low-rate wireless Personal Area Networks (PAN). Its protocol stack is simple and flexible, it does not require any infrastructure and it is suitable for short-range communications (typically within a range of 10 meters). For these reasons it features ease of installation, low cost, and a reasonable battery life of the devices.

The physical layer of the IEEE 802.15.4 has been designed to coexist with other IEEE standards for wireless networks such as IEEE 802.11 and IEEE 802.15.1 (Bluetooth). It features activation and deactivation of the radio transceiver and transmission of packets on the physical medium. It operates in one of the following three license-free bands:

- 868–868.6 MHz (e.g., Europe) with a data rate of 20 kbps;
- 902–928 MHz (e.g., North America) with a data rate of 40 kbps; or
- 2400–2483.5 MHz (worldwide) with a data rate of 250 kbps.

The MAC layer provides data and management services to the upper layers. The data service enables transmission and reception of MAC packets across the physical layer. The management services include synchronization of the communications, management of guaranteed time slots, and association and disassociation of devices to the network. In addition the MAC layer implements basic security mechanisms.

Since the description of physical layers for wireless sensor networks is out of the scope of this book, we limit this presentation to the MAC layer features of this standard. The acronyms used in this section are listed in Table 1.

Acronym	Definition
ACL	Access Control List
CAP	Contention Access Period
CFP	Contention Free Period
CSMA-CA	Carries Sense Multiple Access with Collision Avoidance
FFD	Full Function Devices
GTS	Guaranteed Time Slots
MAC	Medium Access Control layer
PAN	Personal Area Network
RFD	Reduced Function Devices

Table 1: Acronyms used in the IEEE 802.15.4 MAC layer.

15.2.1. Overview of the MAC layer

The MAC layer defines two types of nodes: Reduced Function Devices (RFDs) and Full Function Devices (FFDs). RFD are meant to implement end-devices with reduced processing, memory, and communication capabilities which implement a subset of the MAC layer functions. In particular the RFD can only associate to an existing network and they depend on FFDs for communication. One RFD can be associated to only one FFD at a time. Example of RFD devices are simple sensors or actuators like light switches, lamps and similar devices.

The FFDs implement the full MAC layer and they can act aither as the Personal Area Network (PAN) coordinator or as a generic coordinator of a set of RFDs. The PAN coordinator sets up and manages the network, in particular it selects the PAN identifier and manages associations or disassociation of devices. In the association phase the PAN coordinator assigns to the new device a 16 bit address. This address can be used in alternative to the standard 64 bit extended IEEE address which is statically assigned to each device.

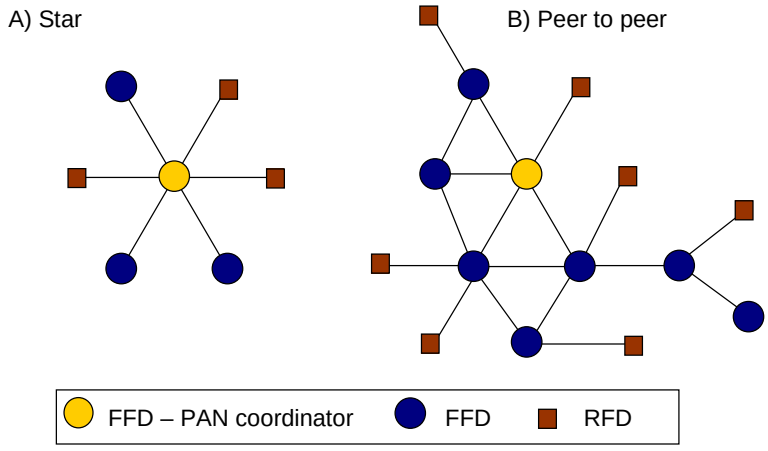

Figure 2. Network topologies supported by the IEEE 802.15.4 MAC layer.

The FFDs cooperate to implement the network topology. The actual network formation is performed at the network layer, but the MAC layer provides support to two types of network topologies: star and peer-to-peer.

In the star topology one FFD is the PAN coordinator and it is located in the star centre. All the other FFD and RFD behave as generic devices and they can only communicate with the coordinator which synchronizes all the communications in the network. Different stars operating in the same area have different PAN identifier and operate independently of each other. An example of star topology is shown in Figure 2A.

In the peer to peer topology each FFD is capable of communicating with any other device within its radio range. One FFD (normally the FFD which initiated the network) act as PAN coordinator, and the other FFDs act as routers or end devices to form a multihop network as shown in Figure 2B. The RFDs act as end devices and each RFD is connected only with one FFD.

15.2.2. The channel access

The MAC protocol have two channel access: with or without a superframe structure. The channel access with superframe structure is used in star topologies (it can also be used in peer to peer topologyes organized in trees) and provides synchronization between nodes to enable energy savings of the devices. On the other hand, the channel access without superframe structure is more general and can be used to support communications in arbitrary peer to peer topologies.

15.2.2.1. Communications with the superframe structure

A superframe is composed by an active and an inactive portion. All the communications happen during the active portion, hence the PAN coordinator (and the connected devices) may enter a low power (sleep) mode during the inactive portion. The active portion comprises up to 16 equally sized time slots. The first time slots is the beacon frame and is sent by the PAN coordinator to begin the superframe. The beacon frames are used to synchronize the attached devices, to identify the PAN, and to describe the structure of the superframes. The actual communications between the end devices and the coordinator take place in the remaining time slots. The time slots in the active portion are divided into a Contention Access Period (CAP) and a (optional) Contention Free Period (CFP).

In the CAP period the devices compete for channel access using a standard slotted CSMA-CA protocol (Carries Sense Multiple Access with Collision Avoidance). This means that a device wishing to transmit data frames first waits for the beacon frame and then it randomly selects a time slot for its transmission. If the selected time slot is busy because another communication is already ongoing (this is detected using carrier sense) then the device selects randomly another time slot. If the channel is idle, the device can begin transmitting on the next slot.

The CFP period is optional and it is used for low-latency applications or applications requiring specific data bandwidth. To this purpose the PAN coordinator may assign portions of the active superframe (called Guaranteed Time Slots or GTS) to specific applications. The GTSs form the contention-free period (CFP), which always begins at the end of the active superframe starting at a slot boundary immediately following the CAP. Each GTS may comprise more than one time slots and it is assigned to an individual application which access it without contention.

In any case the PAN coordinator always leave a sufficient number of frames for the CAP period for the other devices and to manage the association/disassociation protocols. Note also that all contention-based transactions shall be complete before the beginning of the CFP, and each device transmitting in a GTS shall complete its transmission within its GTS. The superframe structure is shown in Figure 3.

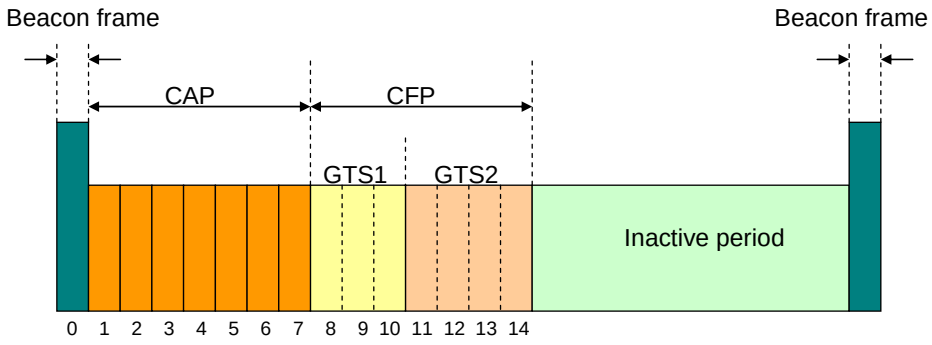

Figure 3. The superframe structure.

15.2.2.2. Communications without the superframe structure

The PAN coordinator may optionally avoid the the use of the superframe structure (thus the PAN is called *non beacon-enabled*). In this case the PAN coordinator never sends beacons and communication happens on the basis of the unslotted CSMA-CA protocol. The coordinator is always on and ready to receive data from an end-device while data transfer in the opposite direction is poll-based: the end device periodically wakes up and polls the coordinator for pending messages. The coordinator responds to this request by sending the pending messages or by signalling that no messages are available.

15.2.3. Data transfer models

The standard supports three models of data transfer: end device to the coordinator, coordinator to an end device, and peer to peer. The star topology use only the first two models because the data transfers can happen only between the PAN coordinator and the other devices. In the peer to peer topology all the three models are possible since data can be exchanged between any pair of devices. The actual implementation of the three data transfer models depends on whether the network supports the transmission of beacons.

15.2.3.1. Data transfers in beacon-enabled networks

Data transfer from an end device to a coordinator : The end device first waits for the network beacon to synchronize with the superframe. When the beacon is received, if it owns a GTS it directly use it, otherwise it transmits the data frame to the coordinator using the slotted CSMA-CA protocol in one of the frames in the CAP period. The coordinator may optionally acknowledge the successful reception of the

data by transmitting an acknowledgment frame in a successive time slot. This sequence is shown in Figure 4A.

Data transfer from a coordinator to an end device : The coordinator stores the message (a data frame) and it indicates in the network beacon that the data message is pending. The end-device usually sleeps most of the time and it periodically listens to the network beacon to check for pending messages. When it notice that a message is pending it explicitly requests the message to the coordinator using the slotted CSMA-CA in the CAP period. In turn, the coordinator sends the pending message in the CAP period using the slotted CSMA-CA. The device thus acknowledges the reception of the data by transmitting an acknowledgment frame in a successive time slot so that the coordinator can remove the pending message from its list. This protocol is shown in Figure 4B.

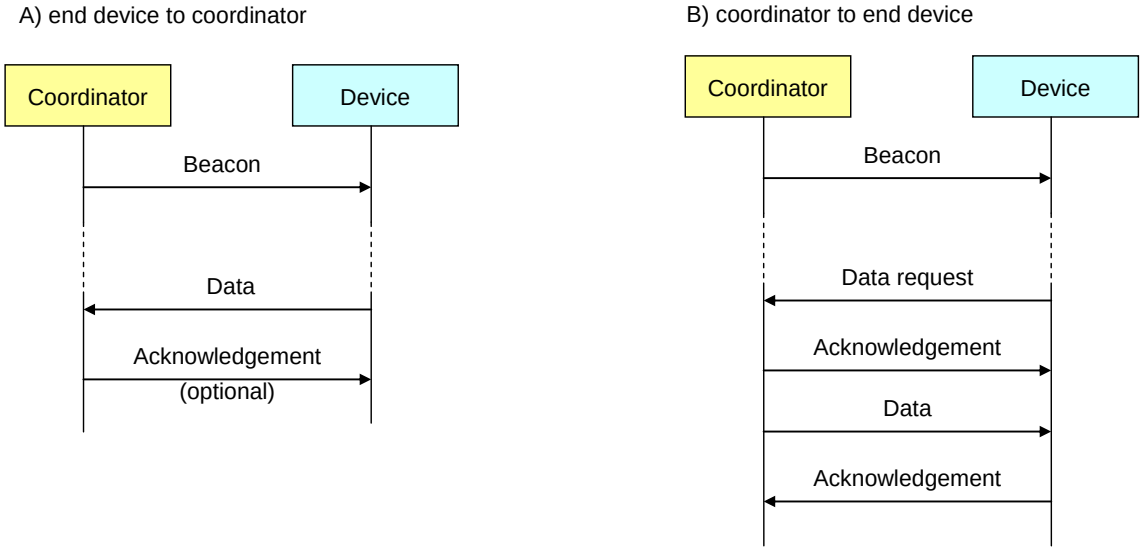

Figure 4. Data transfer modes in beacon-enabled networks.

Peer-to-peer data transfers : If the sender or the receiver is a end device then one of the above schemes is used. Otherwise both the source and the destination are coordinators and they issue their own beacons. In this case the sender must first synchronize with the beacon of the destination and act as an end device. The measures to be taken in order to synchronize coordinators are beyond the scope of the IEEE 802.15.4 standard and are thus left to the upper layers.

15.2.3.2. Data transfers in non beacon-enabled networks

Data transfer from an end device to a coordinator : In this case the end device transmits directly its data frame to the coordinator using the unslotted CSMA-CA protocol. The coordinator acknowledges the

successful reception of the data by transmitting an optional acknowledgment frame. This sequence is summarized in Figure 5A.

Data transfer from a coordinator to an end device : The coordinator stores the message (a data frame) and waits for the appropriate device to request for the data. A device can inquiry the coordinator for pending messages by transmitting a request using the unslotted CSMA-CA protocol (this request happens at an application-defined rate). The coordinator acknowledges the successful reception of the request by transmitting an acknowledgment frame. If there are pending messages, the coordinator transmits the messages to the device using the unslotted CSMA-CA protocol. Otherwise, if no messages are pending, the coordinator transmits a message with a zero-length payload (which indicates that no messages are pending). The device acknowledges the successful reception of the messages by transmitting an acknowledgment frame so that the coordinator can discard the pending messages. This protocol is shown in Figure 5B.

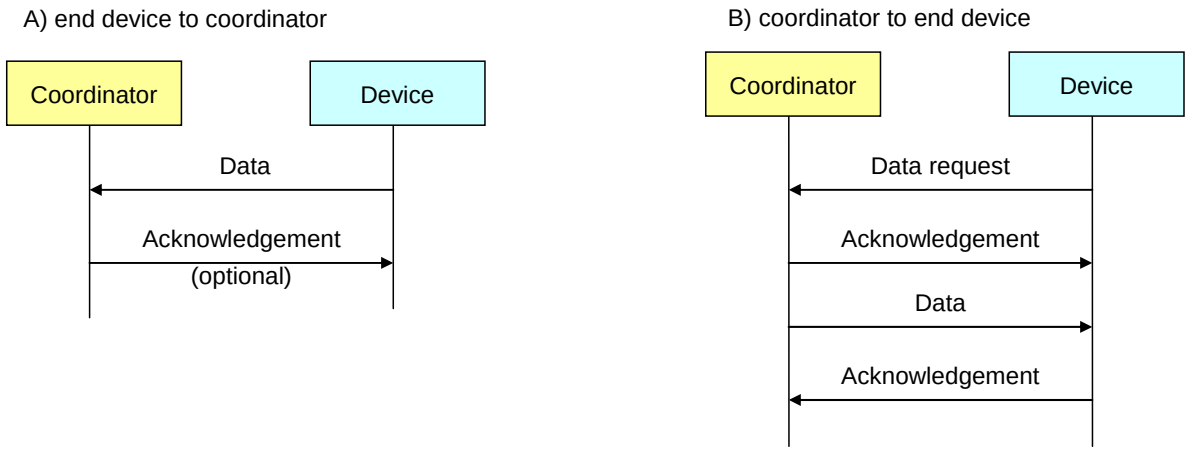

Figure 5. Data transfer modes in non beacon-enabled networks.

Peer-to-peer data transfers : In peer-to-peer PANs each device can communicate with each other device in its radio range. In order to do this effectively, the devices wishing to communicate need either to i) keep the radio constantly active in order to be ready to receive incoming messages or ii) to synchronize with each other. In the former case the device can directly transmit the data using unslotted CSMA-CA while in the second case the device has to wait until the destination device is ready to receive. Note however that the devices synchronization is beyond the scope of the IEEE 802.15.4 standard and it is left to the upper layers.

15.2.4. The MAC layer services

The MAC layer provide data and management services to the upper layer (normally the ZigBee network layer). Each service is specified by a set of primitives which can be of four generic types (as illustrated in Figure 6):

- **Request**: It is invoked by the upper layer to request for a specific service;
- **Indication**: It is generated by the MAC layer and it is directed to the upper layer to notify the occurrence of an event related to a specific service;
- **Response:** It is invoked by the upper layer to complete a procedure previously initiated by an indication primitive;
- **Confirm:** It is generated by the MAC layer and is directed to the upper layer to convey the results of one or more service requests previously issued.

Each service may use all or part of the four primitives depending on its needs.

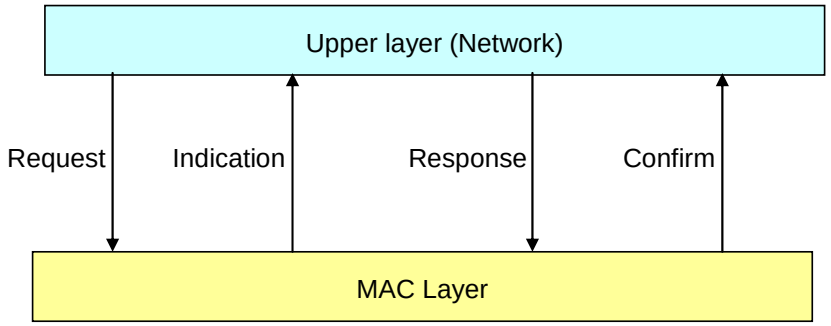

Figure 6. The four types of primitives used to implement the MAC services.

15.2.4.1. The data service

The data service comprises one main service which exploits only the request, confirm and indication primitives. The DATA.request primitive is invoked by the upper layer to send a message to another device. The result of a transmission requested with a previous DATA.request primitive is reported by the MAC layer to the upper layer by the DATA.confirm primitive, which returns the status of transmission (either success or an error code). The DATA.indication primitive corresponds to a “receive” primitive: it is generated by the MAC layer on receipt of a message from the physical layer to pass the received message to the upper layer.

Figure 7 illustrates the sequence of messages and primitives occurring during a data exchange between two nodes.

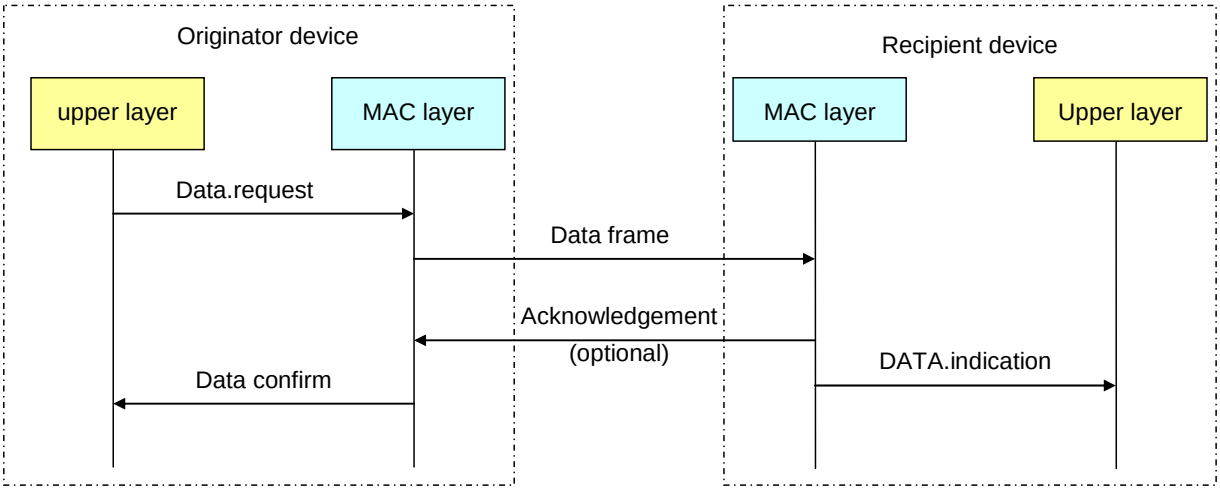

Figure 7. Implementation of the DATA servce.

Name	Request	Indication	Response	Confirm	Functionality
ASSOCIATE	X	O	O	X	Request of association of a new device to an existing PAN.
DISASSOCIATE	X	X		X	Leave a PAN.
BEACON-NOTIFY		X			Provides to the upper layer the received beacon.
GET	X			X	Reads the parameters of the MAC.
GTS	O	O		O	Request of GTS to the coordinator.
SCAN	X			X	Look for for active PANs.
COMM-STATUS		X			Notify the upper layer about the status of a transaction begun with a response primitive.
SET	X			X	Set parameters of the MAC layer.
START	O			O	Starts a PAN and begins sending beacons. Can also be used for device discovery.
POLL	X			X	Request for pending messages to the coordinator.

Table 2. Main management services of the MAC layer.

15.2.4.2. The management service

The management services of the MAC layer include functionalities for PAN initialization, devices association/disassociation, detection of existing PANs and other services exploiting some of the features of the MAC layer. The main management services are summarized in Table 2. In the table symbol X in a cell corresponding to service S and primitive P denotes that S uses the primitive P, while symbol O means that primitive P is optional for the RFDs.

As a matter of example we describe here the protocol and functionalities of the ASSOCIATE service. This service is invoked by a device wishing to associate with a PAN which it have already identified by preliminary invoking the SCAN service. The ASSOCIATE.request primitive takes as parameters (among others) the PAN identifier, the coordinator address, and the 64-bits extended IEEE address of the

device. The primitive sends an association request message to a coordinator (either the PAN coordinator or a router). Since the association procedure is meant for beacon-enabled networks, the association request message is sent during the CAP using the slotted CSMA-CA protocol.

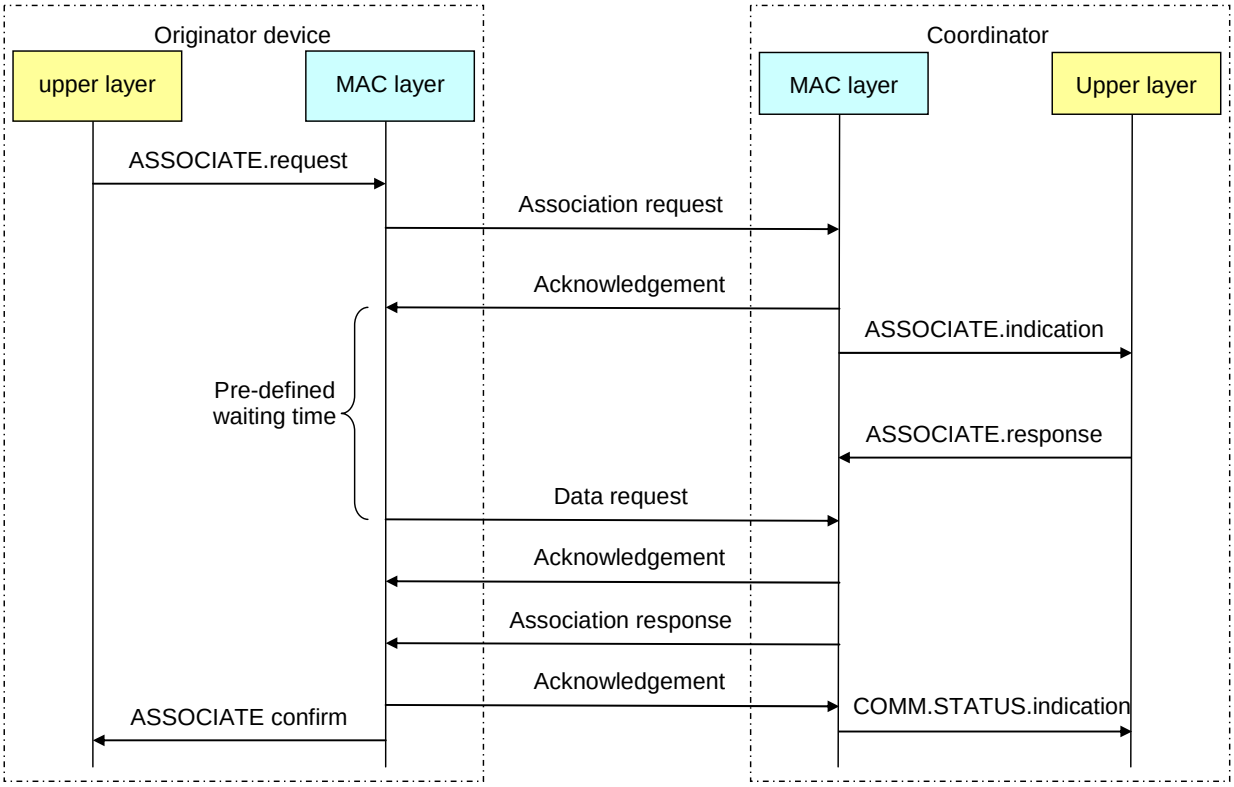

Figure 8. Implementation of the ASSOCIATE service.

The coordinator acknowledges immediately the reception of the association messages, however this acknowledgement does not mean that the request has been accepted.

On the coordinator side the association request message is passed to the upper layers of the coordinator protocol stack (using the ASSOCIATE.indication primitive) where the decision about the association request is actually taken. If the request is accepted the coordinator selects a short 16 bit address that the device may use later in place of the 64-bit extended IEEE address. The upper layers of the coordinator thus invoke the ASSOCIATE.response primitive of the coordinator MAC layer. This primitive takes as parameters the 64 bit address of the device, the new 16 bit short address and the status of the request (which can be association successful or an error code). The primitive thus generates an association response command message which is sent to the device requesting association using indirect transmission, i.e., the command message is added to the list of pending messages stored on the coordinator. The MAC layer of the device automatically issues a data request message to the coordinator after a pre-defined

period following the acknowledgement of the association request command[1]. The coordinator then sends the association response command message to the device.

Upon receiving the command message, the devices' MAC layer issues a ASSOCIATE.confirm primitive, while the MAC layer of the coordinator issues the COMM-STATUS.Indication primitive to inform the upper layer that the association protocol is concluded either with success or with an error code. The association protocol is shown in Figure 8.

15.2.5. Security

The IEEE 802.15.4 MAC layers provides a basic support for security, and it leaves advanced security features (such as keys management and device authentication) to the upper layers. All the security services are based on symmetric-keys and use keys provided by the higher layers. The MAC layer security services also assume that the keys are generated, transmitted, and stored by the upper layers in a secure manner. Note also that the security features of the MAC layer are optional and the applications can decide when and which functionality they use.

The security services provided by the MAC layer are the following:

- **Access control**: allows a device to keep a list of devices (called the Access Control List, or ACL) with which it is enabled to communicate. If this service is activated, each device in the PAN maintains its own ACL and it discards all the incoming packets received from devices not included in the ACL.
- **Data encryption**: uses symmetric cryptography to protect data from being read by parties without the cryptographic key. The key can be shared by a group of devices (typically stored as the default key) or it can be shared between two peers (stored in an individual ACL entry). Data encryption may be provided on data, command, and beacon payloads.
- **Frame integrity**: uses an integrity code to protect data from being modified by parties without the cryptographic key and to assure that the data comes from a device with the cryptographic key. As in the data encryption service the key can be shared by a group or by pairs of devices. Integrity may be provided on data, beacon and command frames.

[1] Note that there are two ways for a device to request a pending data message to the coordinator: by the POLL service or automatically after a pre-defined period following the acknowledgement of a previous request command (as in the ASSOCIATE service).

- **Sequential freshness**: orders the sequence of input frames to ensure that an input frame is more recent than the last received frame.

15.3. The ZigBee Standard

ZigBee builds upon the IEEE 802.15.4 standard. It specifies the network and the application layers. The network layer provides support to star, tree, and peer-to-peer multi-hop network topologies, and the application layer provides a framework for distributed application development and communication. The application layer comprises the Application Framework, the ZigBee Device Objects (ZDO), and the Application Support sublayer (APS). The Application Framework contains up to 240 Application Objects (APO), that is, user defined application modules which implement a ZigBee application. The ZDO provides services that allow the APOs to organize into a distributed application. The APS provides data and management services to the APOs and ZDO. An overview of the ZigBee protocol stack is shown in Figure 9, and Table 3 summarizes the acronyms used in this section.

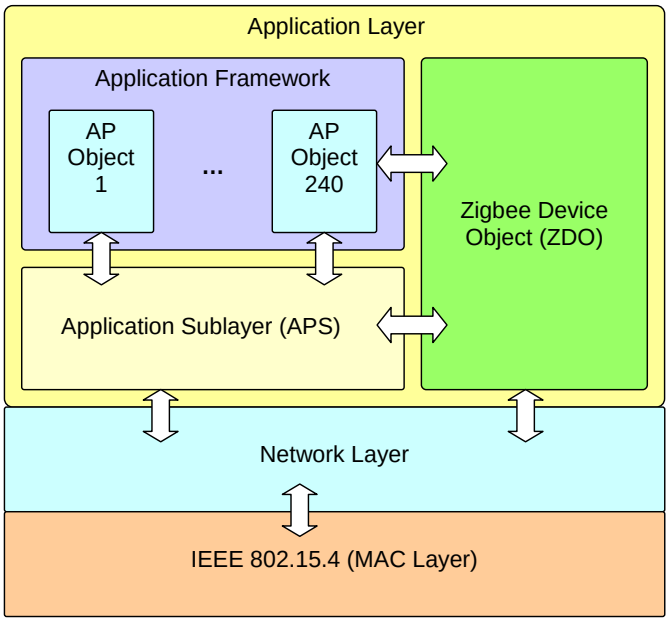

Figure 9: ZigBee functional layer architecture and protocol stack.

Acronym	Definition
APO	Application Objects
APS	Application Sublayer
RDT	Route Discovery Table of the network layer
RREQ	Route Request message (network layer)
RREP	Route Reply message (network layer)
RT	Routing Table of the network layer
ZDO	ZigBee Device Object

Table 3: Acronyms used in the ZigBee network and application layers.

15.3.1. The network layer

The network layer defines three types of devices: the end-device which corresponds to a RFD or to a FFD acting as a simple device, the router which is a FFD with routing capabilities, and the network coordinator which is a FFD managing the whole network. Besides the star topology (that naturally maps to the star topology of IEEE 802.15.4), the network layer also supports tree and mesh topologies (the ZigBee network topologies are shown in Figure 10). The network layer provides services for the initialization of the network, devices addressing, routes management, routing, and management of connections and disconnections of devices. Table 4 lists the set of services of the network layer. Differently than the IEEE 802.15.4 MAC layer, the network layer services are defined only in terms of request, indication and confirm primitives. In the following we describe in more details the main network protocols which implement the services for network creation, join, and routing.

Name	Request	Indication	Confirm	Description
DATA	X	X	X	Data transmission service
NETWORK-DISCOVERY	X		X	Look for existing PANs
NETWORK-FORMATION	X		X	Create a new PAN (invoked by a router or by a coordinator)
PERMIT-JOINING	X		X	Allows associations of new devices to the PAN (invoked by a router or by a coordinator)
START-ROUTER	X		X	(Re-)initializes the superframe of the PAN coordinator or of a router
JOIN	X	X	X	Request to join an existing PAN (invoked by any device
DIRECT-JOIN	X		X	Request to other devices to join the PAN (used by routers or by the coordinator)
LEAVE	X	X	X	Leave a PAN
RESET	X		X	Resets the network layer
SYNC	X	X	X	Allows the application layer to synchronize with the coordinator or a router and/or to extract pending data from it
GET	X		X	Reads the parameters of the network layer
SET	X		X	Set parameters of the network layer

Table 4. Services offered by the network layer.

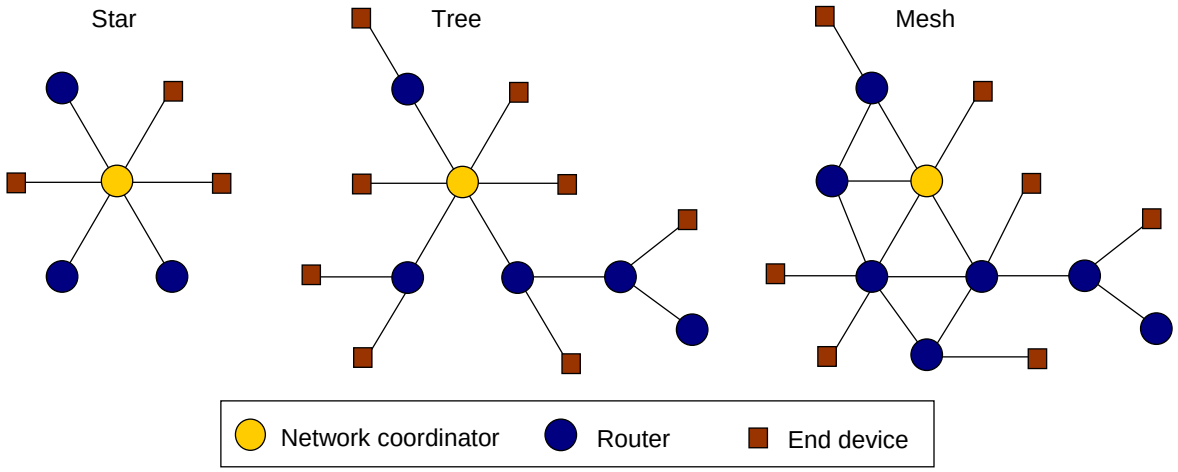

Figure 10: ZigBee network topologies.

15.3.1.1. Network formation

The procedure to establish a new network is initiated by the NETWORK-FORMATION.request primitive. This primitive can be invoked only by FFD devices that can behave as coordinator and that are not currently joined to another network. The primitive first uses the MAC layer services to look for a channel which does not conflict with other existing networks.

If a suitable channel is found the primitive selects a PAN identifier which is not already in use by other PANs, and assigns to the device (which is also the coordinator of the new PAN) the 16-bit network address 0x0000. The primitive then invokes the SET.request primitive of the MAC layer to set the PAN identifier and the device address; then it invokes the START.request primitive of the MAC layer to start the PAN. In response to this primitive the MAC layer starts generating the beacons.

15.3.1.2. Joining a network

The join procedure can be requested by a device wishing to join an existing network (*join through association*) or it can be requested by a router or by the coordinator to force a device to join its PAN (*direct join*). In this section we present the join through association procedure.

When the application layer running on a device *D* wishes to join an existing network, it first invokes the NETWORK-DISCOVERY service to look for existing PANs. This procedure exploits the MAC layer SCAN service to learn about neighbouring routers that announce their networks. Once this procedure is completed, the application layer is notified about the existing networks. In turn, the application layer selects one network (several ZigBee networks may spatially overlap, using different channels) and it

invokes the JOIN.request primitive with two parameters: the PAN identifier of the selected network and a flag indicating whether it joins as a router or as an end device.

The JOIN.request primitive in the network layer selects a "parent" node *P* (in the desired network) from his neighbourhood. The parent should be a device in the PAN allowing joins. For instance, in the case of the star topology, the parent is the coordinator and the devices join as an end device. The network layer then performs the MAC layer association procedure to *P*. Upon receiving an indication of the association request from the MAC layer, the *P*'s network layer assigns *D* a 16-bit short address and lets the MAC layer successfully reply to the association request. Node *D* will use the short address for any further network communication. Figure 11 shows the join procedure at the device side.

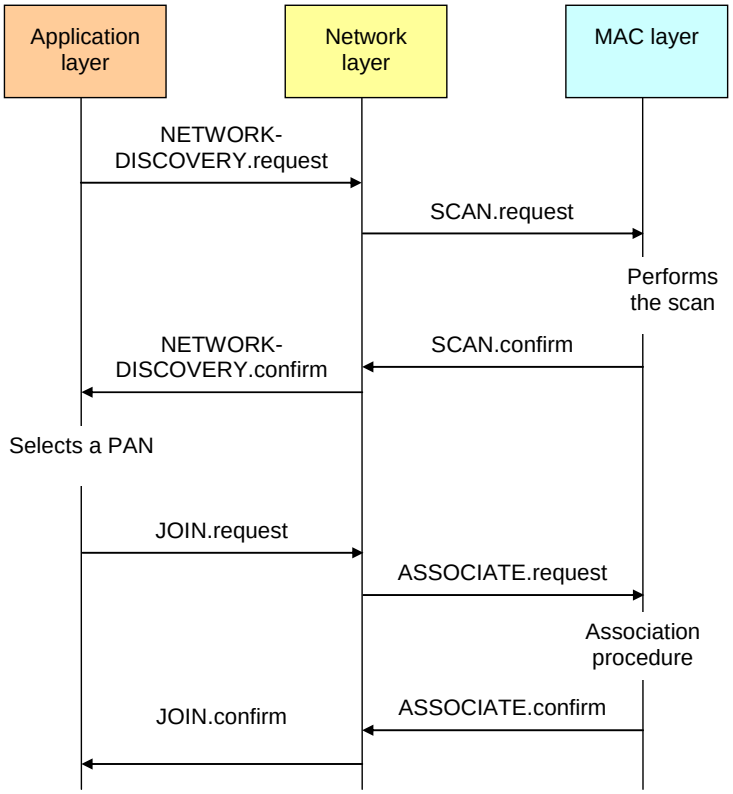

Figure 11. The JOIN protocol at the child's side.

The parent-child relationships established as a result of joins, shape the whole network in the form of a tree with the ZigBee coordinator as the root, the ZigBee routers as internal nodes and ZigBee end-devices as leaves. This tree structure is also at the basis of the distributed algorithm for network address assignment. The ZigBee coordinator fixes the maximum number of routers (R_m) and end-devices (D_m) that each router may have as children and also fixes the maximum depth of the tree (L_m). On the basis of its depth in the tree, a newly joined router is assigned a range of consecutive addresses (16-bit integers). This

range is such that the router will have enough addresses for all of its children and descendants and it is computed based on R_m, D_m, and L_m. Figure 12 shows an example of addresses assignment in a network with R_m=2, D_m=2 and L_m=3 where all addresses have been assigned to routers (white nodes) and end-devices (gray nodes). The address appears inside the circle representing each node, while the assigned address ranges are displayed in brackets next to each router.

Although the addresses are always assigned assuming a tree topology, the network layer can be configured by the application layer to implement a mesh or a tree topology. If the configuration is the mesh then the all the nodes (coordinator, routers and end devices) communicate without superframe structure (see Section 15.2.2.1).

Otherwise, if the topology is the tree the network can communicate with superframe structure. In this case all newly joined routers invoke the primitive START-ROUTER.request to begin transmitting their beacons. To avoid overlaps of the activity periods, the routers should have (relatively) long inactive periods and have neighboring routers start their superframe in the inactive period of the other routers to avoid overlapping. Communication from a child to a parent happens in the CAP (Contention Access Period) of the parent while communication from a parent to a child is indirect. In any case a node has to synchronize with the parent's beacon to exchange data with it, while it drives communication with its children according to its superframe.

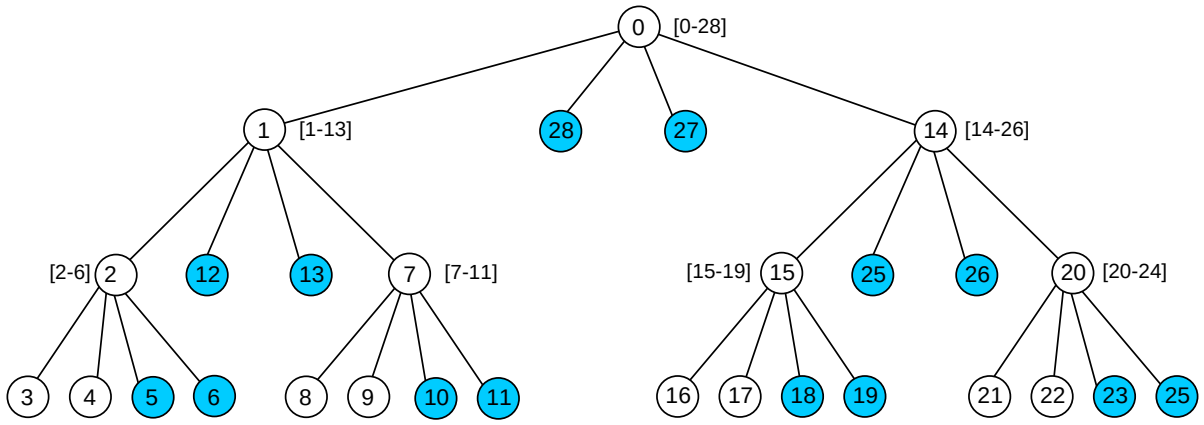

Figure 12: A tree topology and address allocations for R_m=2, D_m=2 and L_m=3.

15.3.1.3. Routing

On receipt of a data frame (a message) the network layer routes the message depending on the capability of the device. If the sender is an end device it forwards the message to its parent, which have routing capability. Otherwise the sender is a router (or the coordinator), thus it maintains a Routing Table RT (the fields of a RT entry are shown in Table 5) and it routes the message according to the following procedure.

If the destination is a child then the message is forwarded directly using the DATA service of the MAC layer. Otherwise the actual routing protocol depends on the topology used in the network (tree or mesh).

If the topology is the mesh the network layer look for an entry corresponding to the destination in RT. If the entry corresponding to the destination in RT is not active or such entry does not exists, then the network layer initiates a route discovery procedure (see Section 15.3.1.3) and the message is buffered until the discovery is complete. Otherwise, if the table entry for the destination is active then the table contains the address of the next hop towards the destination, and the message is forwarded to the destination through the next hop. Figure 13 shows the route followed by a message in a mesh topology.

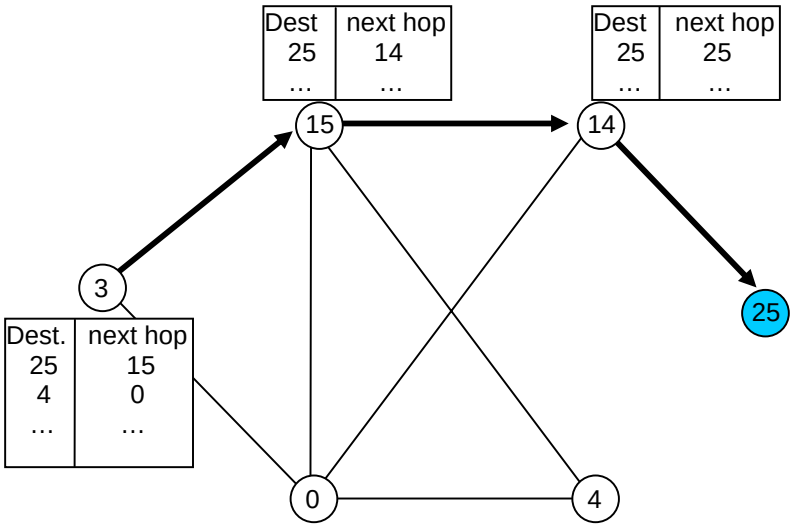

Figure 13: Routing of a message from node 3 to node 25 in a mesh.

If the topology is a tree then the network routes the packets along the tree. In the tree topology routers maintain their address and the address information associated with their children and parent. Given the way addresses are assigned, a router that needs to forward a message can easily determine whether the destination is one of its end-device children or if it belongs to the subtree rooted in one of its children. If so, it routes the packet to the appropriate child; otherwise it routes the packet to its parent. Figure 14 shows the route followed by a message in a tree topology.

Note that tree and mesh topologies may live together, that is, the routers can maintain both information for mesh and tree routing. In this case a router forwarding the message can switch between one routing algorithm to the other. For instance if a route to a destination in the mesh routing is not yet available the message can be forwarded through the tree.

It should also be observed that, while mesh routing is more complex to handle and it does not allow beaconing (it works in networks without the superframe structure), tree routing allows the routers to operate in beacon-enabled networks.

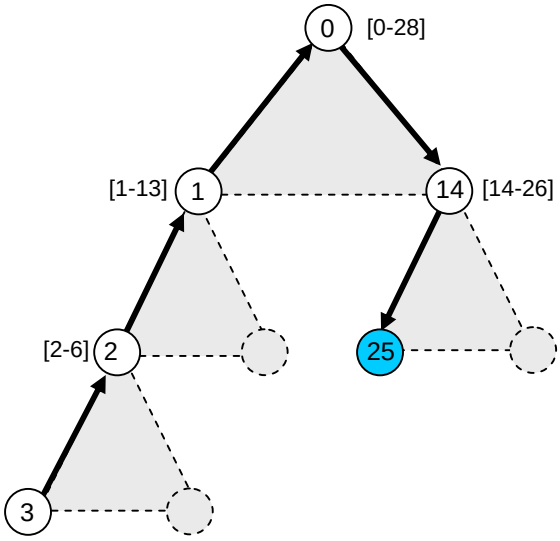

Figure 14: Routing of a message from node 3 to node 25 in a tree.

Field Name	Size	Description
Destination Address	16 bits	network address of the destination
Next-hop Address	16 bits	network address of next hop towards destination
Entry Status	3 bits	Route status: Active, Discovery_underway, Discovery_failed, or Inactive

Table 5: The fields of an entry of the Routing Table (RT) in a ZigBee router.

15.3.1.4. Route discovery

Route discovery is a protocol initiated by the network layer of a source device *S* when it needs to send a message to destination device *D* but its RT does not contain information suitable to route the message. A Route Discovery Table (RDT) (the fields of a RDT entry are shown in Table 6) is maintained by routers and the coordinator to implement route discovery.

To initiate the route discovery *S* broadcasts a route request (RREQ) message that contains the RREQ ID, the destination address and the path cost which is initially be set to 0. The RREQ ID is an integer which the device *S* increments every time it sends a new RREQ message. Thus the RREQ ID and the address of *S* can be used as a unique reference for a route discovery process.

As the RREQ propagates in the network, an intermediate device *I* receiving the RREQ does the following actions:

1. it updates the path cost field adding the cost of the last traversed link. The cost of a link can be a constant or it is a function of the link quality estimation provided by the IEEE 802.15.4 interface.
2. it searches within its RDT for an entry corresponding to the RREQ. If no match is found, a new RDT entry is created for the discovery process and a route request timer is started (upon timer expiration the RDT entry will be removed). Conversely if an entry is found in the RDT, the node compares the path cost for the RREQ message and the corresponding value in the RDT entry. If the former is higher it drops the RREQ message, otherwise it updates the RDT entry.
3. if *I* is not the route discovery destination, it allocates a RT entry for the destination, with status DISCOVERY_UNDERWAY, and rebroadcasts the RREQ after updating its path cost field.
4. else, if the node is the final destination, is replies to the originator with a route reply (RREP) message that travels back along the path.

The RREP message is sent in unicast to the route discovery originator and carries a residual cost value field that each node increments as it forwards the message.

Upon receipt of a route reply (RREP) message, a node does the following:

1. If the node is the RREQ originator and this is the first RREP it received, it sets the corresponding RT entry to ACTIVE and records the residual cost and next hop in the RDT entry.
2. Otherwise (the node is not the RREQ originator):
    a. if the residual cost of the RREP is higher than the residual cost of the corresponding RDT entry then the node discards the RREP message;
    b. otherwise it updates the RDT entry (residual cost) and the RT entry (next hop).
    c. it forwards the RREP towards the originator. Note that intermediate nodes never change the RT entry status to ACTIVE as a result of receiving a RREP message. They will only change the entry status upon reception of a data message for the given destination.

Field Name	Size	Description
RREQ ID	8 bits	Unique ID (sequence number) given to every RREQ message being broadcasted
Source Address	16 bits	Network address of the initiator of the route request
Sender Address	16 bits	Network address of the device that sent the most recent lowest cost RREQ
Forward Cost	8 bits	The accumulated path cost from the RREQ originator to the current device
Residual Cost	8 bits	The accumulated path cost from the current device to the RREQ destination
Expiration time	16 bits	A timer indicating the number of milliseconds until this entry expires.

Table 6: The fields of an entry of the Route Discovery Table (RDT).

15.3.2. The application layer

The Application layer defines the Application Framework under which the programmers develop applications in terms of Application Objects (APO). The APOs exploit the services offered by the ZDO and the APS sublayers which include data, binding, and discovery services.

15.3.2.1. The application framework

The Application Framework contains up to 240 APOs, each of which is interfaced with an application endpoint numbered from 1 to 240. The endpoint 0 is reserved for the ZDO. Each APO in the network is uniquely identified by combining its endpoint address and the network address of the hosting device. The APOs define the behaviour of Zigbee applications. They can have complex states and they communicate exploiting the data services of the APS.

Figure 14 shows an example of a simple ZigBee application. Device *A* contains two APOs (attached to endpoints 10 and 25, respectively), each of which controls a switch. Device *B* contains three APOs (attached to endpoints 5, 6, and 8, respectively), each controlling a lamp. One switch (10*A*) controls three lamps (5*B* and 6*B*) and switch (25*A*) controls lamp 8*B*. In this simple example the APOs 5*B*, 6*B*, and 8*B* could have a single attribute containing the status of the lamp (on/off) which can be set remotely from the APOs 10*A* and 25*A*.

To the purpose of specification of services and applications the ZigBee standard introduces the concept of clusters and profiles. A cluster is the specification in a standard format of the messages managed by an APO. Clusters are numbered within a given application profile with an 8 bit identifier.

An application profile is the specification in a standard format of the behaviour of an application possibly operating on several ZigBee devices. An application profile describes a set of devices and clusters. The application profiles are assigned with a unique identification number which is assigned by the ZigBee alliance.

15.3.2.2. The binding and discovery services

Binding and discovery of services and devices are the main services are provided to the APOs.

Device Discovery – Device discovery allows a device to obtain the (network or MAC) address of other devices in the network. A router (or the coordinator) responds to a device discovery query by returning its address and the address of all its associated end devices.

Service discovery – Service discovery exploits cluster descriptors ad cluster identifiers to determine the services offered by a given APO. It can be accomplished by issuing a query for each endpoint on a given device or by using a match service feature. In the example of Figure 15 device and service discovery can be used by device *A* to determine the address of *B* and the services offered by its APOs. Once device *A* has discovered the address and the services offered by *B* it can issue request messages to *B* according to the cluster descriptions of the APOs of *B*.

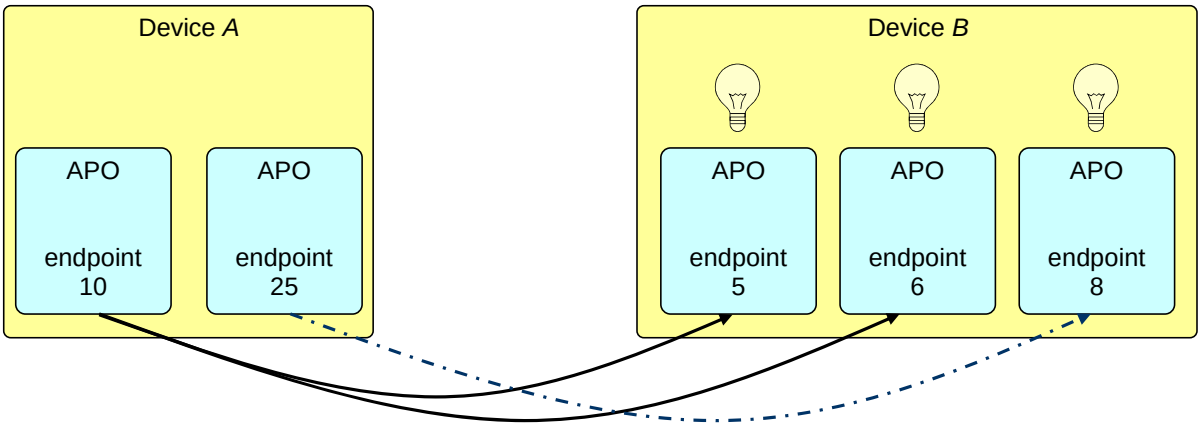

Figure 15. A simple ZigBee application.

Binding – A message is normally routed from the source to the destination APO based on the destination address pair <destination endpoint, destination network address>. However, this kind of addressing (called *direct* addressing) might be unsuitable for extremely simple devices which could be unable to store information about the address of the destination device. For this reason ZigBee also offers the *indirect* addressing which exploit binding tables to translate the source address (in terms of network and endpoint address) and the cluster identifier of the message into the pair <destination endpoint, destination network address>. The binding table is stored in the ZigBee coordinator and/or in the routers and it is updated on explicit request of the ZDO in the routers or in the coordinator. A binding table for the example of Figure 14 is shown in Table 7.

<Source address, endpoint address, cluster identifier>	<destination address, endpoint number>
<*A*,10,15>	<*B*,5>,<*B*,6>
<*A*,25,15>	<*B*,8>

Table 7. The binding table for the ZigBee application of Figure 14.

15.3.2.3. The Application Support Sublayer (APS)

The APS offers the binding service to the ZDO and the data service to both the APOs and the ZDO.

The data service enables the exchange of messages between two or more devices within the network using either direct or in direct addressing. The data service is defined in terms of the request, confirm and indication primitives. The request implements a send and the indication implements a receive. The primitive confirm returns to the sender the status of the transmission (either success or an error code).

The binding services comprises the BIND and UNBIND services, both defined in terms of the request and confirm primitives. These services can be invoked only by the ZDO of the coordinator or of a router. The BIND.request primitive take as input parameters the tuple <source address, source endpoint, cluster identifier, destination address, destination endpoint>, and creates in the binding table of the device in which it is invoked an entry corresponding to the tuple in input. The UNBIND.request primitive deletes the entry corresponding to its input parameters from the binding table. The BIND.confirm and UNBIND.confirm primitives return the result of the corresponding request primitive (either success or an error code).

15.3.2.4. The ZigBee Device Object (ZDO)

The ZDO behaves as a special application which uses network and APS primitives to implement ZigBee End Devices, ZigBee Routers, and ZigBee Coordinators. The ZDO is attached to the APS through endpoint 0 and it is specified by a special profile, the ZigBee Device Profile, which describes the clusters that must be supported by any ZigBee device. In particular the ZigBee Device Profile defines how the ZDO should implement the services of discovery and binding and how it should manage the network and the security.

Device and service discovery – The ZDO implements this services depending on the capability of the hosting device. In particular:

- The discovery of end devices and of their services is responsibility of the ZDO of the coordinator. This because the end devices may sleep most of the time and their ZDO may be unable to respond to discovery requests. However the ZDO of an end device should respond to the discovery requests when the device is active.

- The ZDOs in the coordinator and the routers should respond to discovery requests on behalf of their associated sleeping end devices.
- In any case, the ZDO of any device should offer the discovery services to the local APOs.

The device and service discovery requests can be conducted based on different input parameters. Typically the device discovery takes in input a 64 bit extended IEEE address of a device and returns its network address and/or the list of the network addresses of its associated devices. Service discovery is more complex and takes in input a network address and optionally a endpoint number, a cluster identifier, a profile identifier, or a device descriptor. The queried device returns the set of endpoints matching the query (for instance the endpoints which implement a given cluster).

Binding management – The ZDO processes the binding requests received from local or remote endpoint adding or deleting entries from the APS binding table. The ZDO of the coordinator supports the binding of end devices which are requested on the basis of button presses or other manual means.

Network management – This function implements the coordinator, a router or an end device according to configuration settings established either at run time by an application or during installation. If the device is a router or an end device, the network management function offers services for the selection of an existing PAN to join. If the device type is a coordinator or a router, this function provides the ability to create a new PAN. Note however that is also possible to deploy a network without a device pre-designated as coordinator if the first activated Full Function Device (FFD) assumes automatically the role of the coordinator.

Node management – The ZDO serves incoming requests aimed at performing network discovery, retrieving the routing and binding tables of the device and manage joins/disconnections of nodes to the network.

Security management – The ZDO determines whether security is enabled or disabled and, if enabled, manages the keys used for the encryption of the messages.

15.3.3. Security in ZigBee

The security model in ZigBee ensures protection of individual devices but not individual applications in the same device. This allows the re-use of the same keying material among the different layers on the same device, thus reducing the storage costs. The security requirements are message integrity, device authentication, message encryption, and message freshness (to avoid message duplicates). Authentication and encryption are possible either at the network-level and at device-level. Network-level authentication

and encryption are achieved by using a common network key. This prevents outsider attacks while adding very little in memory cost. Device-level authentication and encryption are achieved by using unique link keys between pairs of devices. This prevents insider and outsider attacks but has higher memory cost.

The ZigBee architecture includes security mechanisms at both network and application layers.

The network layer is responsible for securely transmit outgoing frames and securely receive incoming frames. It enforces security using symmetric encryption of outgoing messages and decription of incoming messages using keys provided by the application layer. It should be observed that, even if security is enabled, some command messages cannot be encrypted. This is the case, for instance, of the messages used to associate new devices to the network.

The application layer provides the services for the management of the security policies and of the keys. To this purpose ZigBee defines the *Trust Center* (assumed to be located in the ZigBee Coordinator) that is responsible for providing the keys to the other devices in the network. The trust center generates the keys for network-level and device-level authentication and encryption, and maintains a list of associated devices and keys in use. The devices in the network establish a secure communication link with the trust center using the master key which could be either pre-assigned or provided to the devices using special procedures (it could be manually inserted by the user), and use the secure link to request to the trust center the keys for their needs.

15.4. Conclusions

This chapter presents the most significant standards for wireless sensor networks. The standardization process of wireless sensor network is however far to be complete, and evolutions of the standards presented in this chapter as expected in the near future. Improvements to these standards are expected in the energy efficiency strategies, in particular with respect to mesh networking and synchronization to allow longer inactive period of the devices.

References

[1] ZigBee Alliance, “ZigBee Specifications”, December 2006

[2] Institute of Electrical and Electronics Engineers, Inc., “IEEE Std. 802.15.4-2003 “Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)”, New York, IEEE Press. October 1, 2003.

[3] http://www.ieee802.org/15/pub/TG4.html

[4] http://www.ZigBee.org/en/index.asp

[5] Paolo Baronti, Prashant Pillai, Vince Chook, Stefano Chessa, Alberto Gotta, and Y. Fun Hu, “Wireless Sensor Networks: a Survey on the State of the Art and the 802.15.4 and ZigBee Standards”, Computer Communications, 30, (2007), pp. 1655-1695.