The LightBee Protocol Stack for Wireless Sensor Networks

The LightBee Protocol Stack for Wireless Sensor Networks

1 Overview of the LightBee Protocol Stack

Wireless Sensor Networks (WSNs) are envisioned to consist of hundreds to thousands of small-footprint autonomous nodes each with sensing, computation, and wireless communication capabilities. These nodes are provided with the ability to form dynamic network topologies and multi-hop routing algorithms. The communication mode in WSNs is mostly many-to-one where sensor nodes collect the information and send it to a sink node. WSNs are used in both civilian and military applications such as environment/habitat monitoring, industrial control, intelligent buildings, vehicle traffic monitoring, target tracking and medical diagnostics. Most WSN applications require basic computational abilities and require low-power battery operation. The enabling technologies for such networks are the availability of low-cost processors, energy-efficient wireless transceivers, and advanced sensors..

The SySDSoft LightBee is a light-weight WSN stack comprising the transport, network, and data link layers of the 7-layer OSI model. It is an innovatively-designed set of protocols that provide the required functionality of a WSN at a much-reduced footprint than products based on the Zigbee Alliance protocol specifications.

The key requirements driving the design of the LightBee protocols are:

1. Enable implementation that extends battery life
2. Provide a small footprint implementation
3. Support of communication between large numbers of devices in a multi-application environment
4. Offer a low cost of implementation and low complexity.

2 Technical Overview

2.1 LightBee Protocol Architecture

LightBee is comprised of the data link (MAC), network, and transport layers as shown in Figure 1. It is compatible with the IEEE 802.15.4 physical layer, however, it can also be used with other proprietary physical layer offerings. The data link (MAC) layer is responsible for accessing the wireless medium, the network layer is responsible for routing the data across the network to the destination supporting multi-hop operation, and the transport layer provides end-to-end reliable data delivery and logical addressing.

The applications envisioned for LightBee are mainly building automation, meter reading, industrial control, and leak detection in long-haul water and oil pipelines.

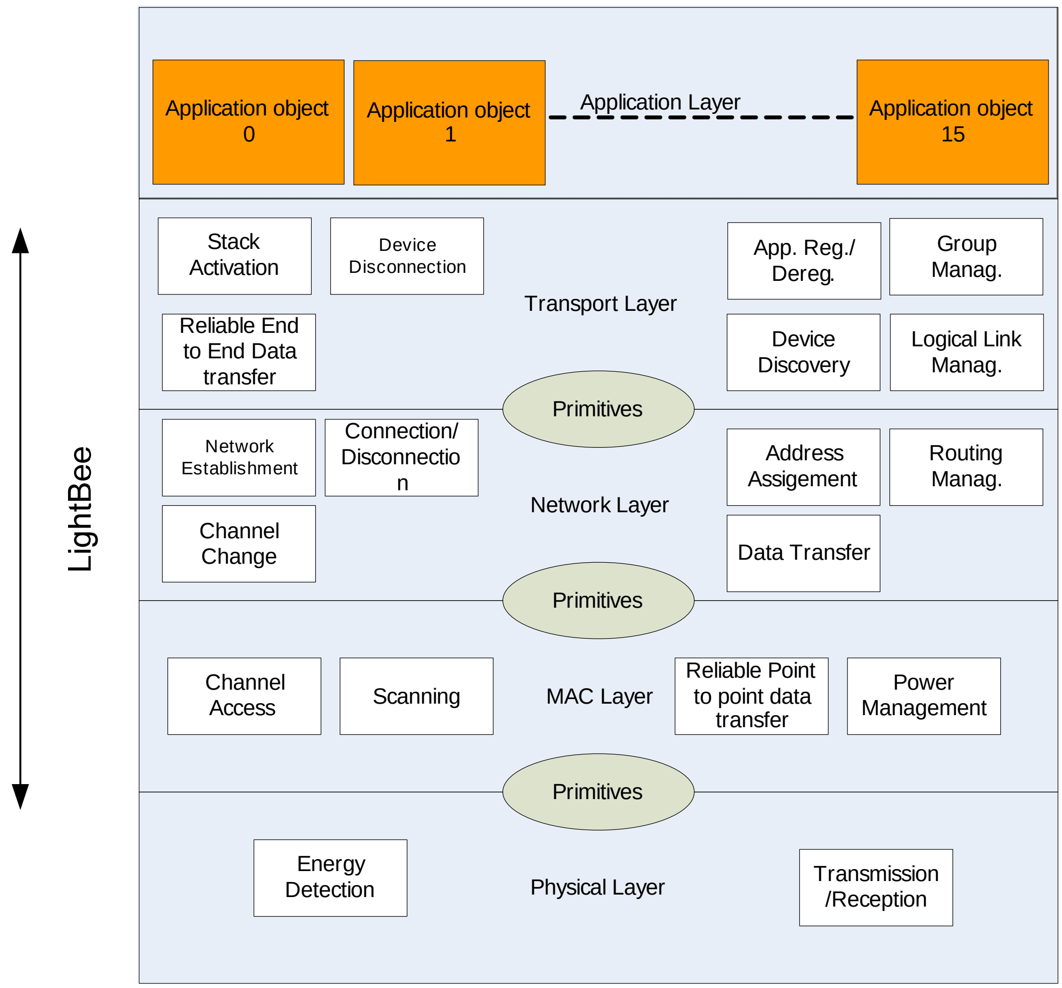

Figure 1: The LightBee Protocol Stack Architecture

MAC Layer Functions

The MAC defines two different device types that can participate in a LightBee sensor network: a Full-Function Device (FFD) and a Reduced-Function Device (RFD). The FFD can serve as coordinator or a router. An FFD can talk to RFDs or other FFDs, while an RFD can talk only to an FFD. An RFD is intended for applications that are extremely simple, such as a light switch; they do not have the need to send large amounts of data and may only communicate with a single FFD at a time. Consequently, the RFD can be implemented using minimal resources and memory capacity.

The MAC sub-layer provides two services: the MAC data service and the MAC management service. The main functionalities that the MAC Layer provides are:

– Beacon management.

– Carrier sense multiple access with collision avoidance (CSMA-CA) channel access.

– Frame validation.

– Maitaining a reliable link between two peer MAC entities.

– Low power consumption.

– Scanning and energy detection.

Network Layer Functions

The network layer is required to form a self-organizing topology and to handle the forwarding of data across the sensor network in a multi-hop fashion. The network layer includes two service entities that provide the necessary functionality to interface with the transport layer. These service entities are the data service and the management service entities. The Network Layer Data Entity (NLDE) provides data transfer service and the Network Layer Management Entity (NLME) provides the management service.

The **NLDE** provides the following services:

– **Generation of the network level protocol data unit** (NPDU): The NLDE is capable of generating an NPDU from the transport layer data through the addition of a Network Layer header.

– **Topology-specific routing**: The NLDE transmits an NPDU to an appropriate device that is either the final destination of the communication or the next hop toward the final destination.

The **NLME** provides the following services:

– Starting a network: this is the ability of the LightBee coordinator to establish a new network.

- Connect/disconnect to/from a network: this is the ability to connect/disconnect a device to/from the network.
- Addressing: this is the ability of the coordinator and router nodes to assign addresses to devices connected to the network.
- Route discovery: this is the ability of the coordinator or a router to discover and record paths through the network, whereby messages may be efficiently routed

Transport layer Functions

The transport layer provides an interface between the network layer and the application layer through a set of services. The services are provided by two entities:

- **Transport layer data entity** unit, which provides data transmission/reception service between two or more application entities located on the same network.
- **Transport layer service management** unit, which provides management services to application through which it can perform device activation and disconnection, application registration/deregistration, device discovery and maintaining multicast groups.

2.2 Device Types

The LightBee network includes different types of devices. These devices are:

- Coordinator: the coordinator node starts up and establishes the network. After network establishment, it acts as a router. The coordinator can be source or sink of data. The network can have only one coordinator node.
- Routers: A router routes messages until it reaches the final destination. A router can be the sink or source of data. A router can act as the coordinator in case the established network coordinator fails. In order to join the network, a router needs to connect to another router that is already connected to the network or connect directly to the coordinator. A router node supports the communication needs of end devices. End devices connected to a particular router are called its children.
- End Device: An end device must connect to a connected router or the coordinator in order to join the network. End devices cannot communicate directly with each other at the link level, i.e. no direct point-to-point connection is possible between two devices. Therefore any data exchanged between end devices must flow through at least one router.

2.3 Network Topologies

LightBee supports three network topologies as shown in Figure 2.

- ***Star Topology:*** all devices are within direct communication range to the coordinator, through which all messages are routed. This topology provides **high performance** as a packet passes at maximum two hops to reach its destination.
- ***Cluster-Tree Topology:*** In this topology routers are allowed to connect to the network. End devices connect to routers (or coordinator). In a cluster tree network, all of the messages sent through the network follow the path of the tree structure.
- ***Mesh Topology:*** In this topology the routers are interconnected. This provides fault-tolerance in case of link failures. Alternate routes to the destination can be identified in case of node failures. This **self-healing** capability provides high **reliability**. Furthermore, using routers extends the network range and enhances network **scalability**.

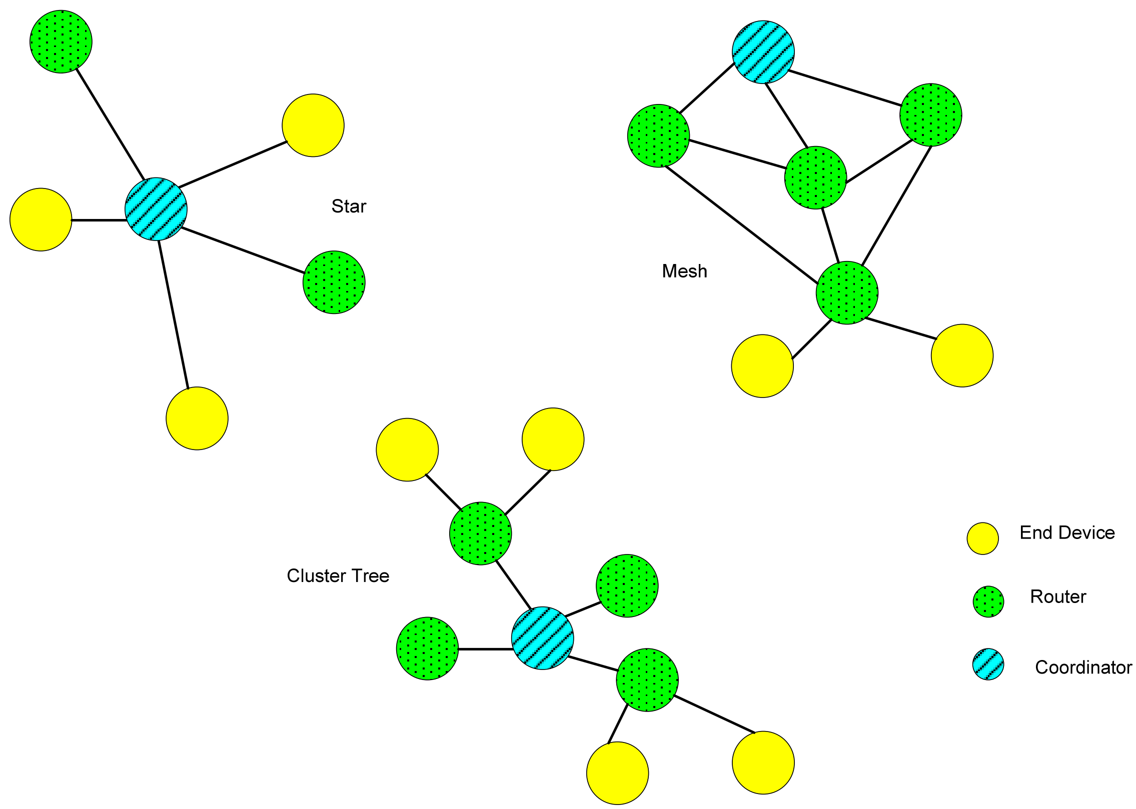

Figure 2: Network Topologies Supported by LightBee

2.4 Addressing

Each device is uniquely identified by 64-bit Extended Unique Identifier (EUI-64). This is the layer 2 (MAC) address of the device that uniquely identifies each device in the network. Upon joining the network, a device is assigned a 10-bit address considered its network layer address. The assignment of the network address is managed by the coordinator and the router devices in order to enable routing in the three aforementioned topologies. The address is changed if the end device reconnects to another router or coordinator node.

The application layer recognizes the devices by their EUI addresses. Upon network establishment, the nodes transport layer keeps a mapping between the EUI-64 address and the current assigned 10-bit network address.

2.5 Packet Format

Figure 3 shows the overall LightBee packet format as delivered to the PHY layer for transmission. The packet is comprised of the transport, network, and MAC layer headers and the MAC layer frame check sequence (FCS). The figure shows the fields aligned to 8-bit boundary. Each of the transport, network, and MAC layers has a control field that contains control information specific to that layer.

The MAC header contains the network ID field which uniquely identifies the sensor network. The next node address and current node network address identify the 10-bit network address of the node to receive the packet and the current node forwarding the packet respectively. A network address of all 1's identifies the broadcast address. The MAC FCS is inserted by the MAC layer and contains the checksum of all bytes in the packet.

The network layer header contains another set of source and destination addresses but these are to identify the final destination and original source of the packet to aid the end-to-end routing. The hop count field contains the maximum number of allowed hops between the source and destination and is decremented at each node forwarding the packet.

The transport layer contains the 4-bit source and destination port numbers to identify the applications. Alternatively, the destination port can be used to identify a group address depending on the transport control field. The 8-bit sequence number is used for reliable delivery mode.

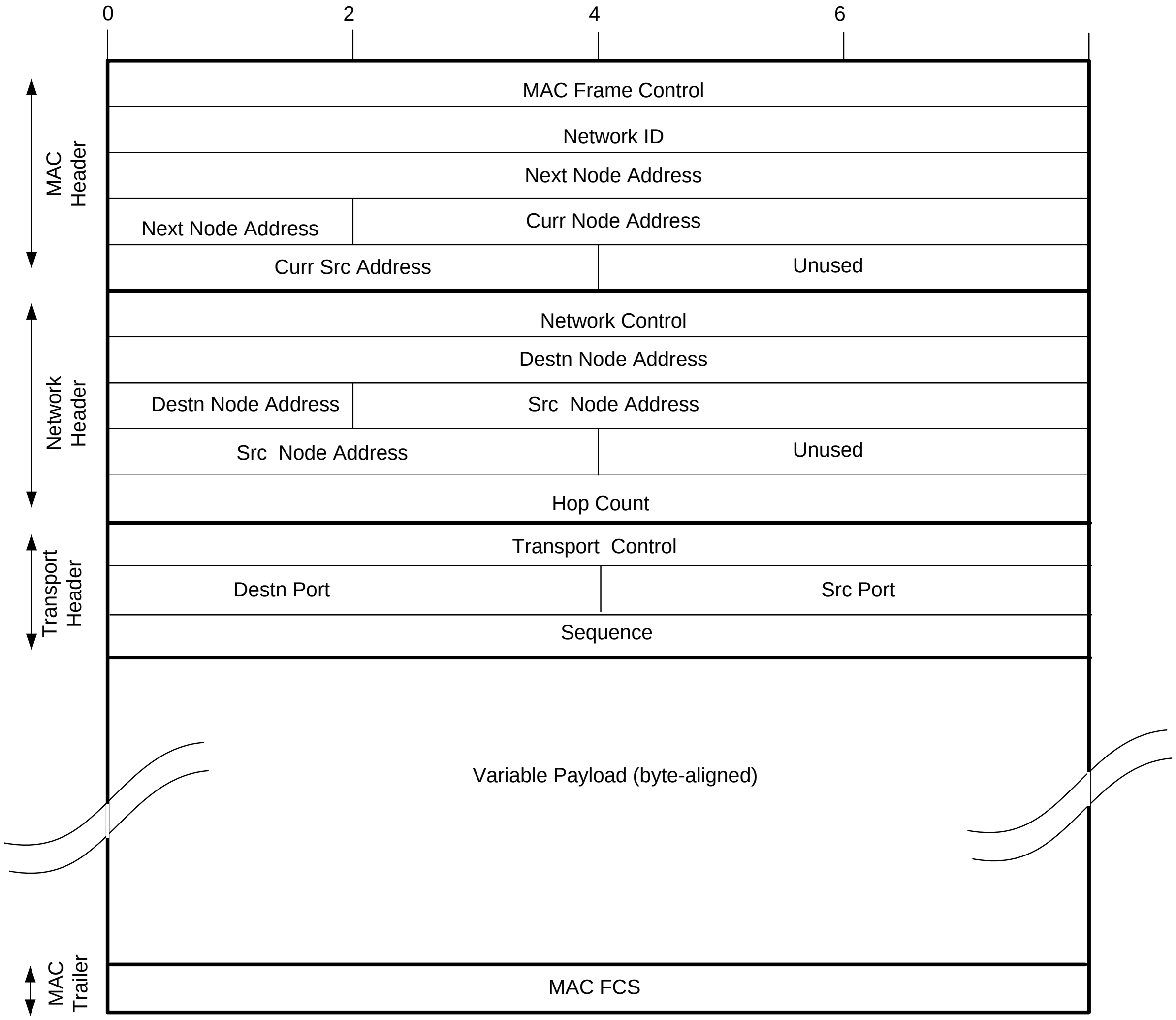

Figure 3: LightBee Packet Format

2.6 Method of Operation

Network Establishment and Device Discovery

LightBee network establishment begins when a pre-configured node selected to act as the network coordinator is powered up and scans the different available channels and selects the channel with the best condition. The physical layer transmits beacon signals on the channel of operation. When a device (end device or router) is powered up, it tunes to the different channels and upon detecting a beacon signal on a certain channel, it shall send a Beacon Request message to solicit beacons from devices that can host it on the network (coordinator or router). The device selects the best 'parent' to connect to. The parent and device exchange messages that allow the device to connect to the network. The parent assigns the device a 10-bit network address.

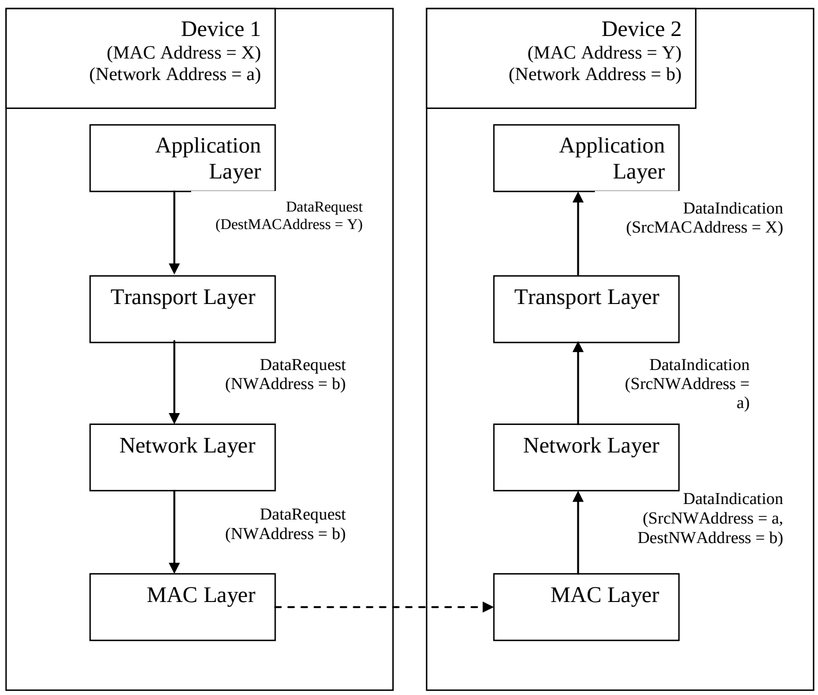

Figure 4: Mapping between EUI-64 and Network Addresses

To support communication at the application layer, LightBee supports application discovery by providing a service to make devices that relate to the same application 'discover' each other. Each device is pre-configured with the profile of applications it should support. Once a device joins the network, it should let other devices know of its existence and its applications profile. It sends its EUI-64 address and 10-bit network address to the coordinator node which routes this message to the other nodes in the network. This message is sent periodically so the devices recognise each other. The devices keep a table mapping the network and EUI-64 bit addresses of other devices in the network.

Routing

The network address is a 10-bit address and thus supports 1024 devices in the network (the actual number is slightly less as there are some special addresses). The network address is a structured address where part of it identifies the device address and the other part identifies the subnetwork to which the device belongs and thus its parent router. During network configuration by the coordinator, it is possible to configure the number of devices per subnetwork and the number of routers per subnetwork.

The data is routed between routers till reaching the parent router in one of two possible ways. In a cluster-tree topology, hierarchical routing based on network address structure is used. In the mesh topology, the routers forward the packets in the best possible path towards the parent router of the device. The parent forwards the data to the end device.

In the mesh topology an intelligent scheme is used to find the best path. If the route to a certain destination is not known, the parent router receiving the packet broadcasts a Route Discovery Request (RDReq) message and specifies the desired destination. The RDReq contains the destination address, the link cost in number of hops, and a sequence number identifying it. As the RDReq message keeps progressing in the network, the recipient routers keep updating their routing tables. A router receiving the RDReq message checks if the original source of the RDReq message has an entry in its routing table. If it does not exist, it then knows that can reach it through the router that forwarded the RDReq message. If it exists, it checks to see if the link cost is better than the current estimate and if so, it would update its table.

Upon reception of the RDReq at the final parent router of the destination device, it responds with a Route Discovery Response (RDRsp) message. The RDRsp contain the source address and path cost. The RDRsp message is unicast back to the source of the RDReq. Each router that the RDRsp traverses updates its routing table with the information contained in the RDRsp. The path cost field in the message is updated and forwarded to the next hop to reach the destination of the RDRsp message.

When a route fails due to high interference for instance or because a device is mobile and has changed its location, a new route discovery to the destination is performed.

Power Management

The coordinator and router nodes are assumed to be mains powered and assumed to be on all the time. In order to extend battery life, battery-operated RFD's are put into sleep mode by default. The transceiver of a RFD is turned off until it is activated by the upper layer processor to send data. Also, the RFD is configured with a polling interval according to which the device periodically polls its parent for data reception. The transceiver enters the sleep state again if the parent acknowledgement to the request is not received within a certain period or is received and indicates no pending data. In case the acknowledgement indicated pending data to the device, the MAC shall remain awake till data reception is completed. The parent of the device is responsible for storing packets destined to sleeping devices.

Media Access

The protocol uses unslotted carrier sense multiple access with collision avoidance (CSMA/CA) to access the medium. Any devices in the network is allowed to transmit at any time as long as the channel is idle. The CSMA/CA is mainly chosen since LightBee is designed with low bit-rate metering application in mind. In such a case CSMA/CA would provide reasonable performance and preserve battery power. Due to the extreme simplicity of the unslotted CSMA/CA, this resulted in small footprint implementation.

Data Transfer

LightBee supports unreliable and reliable data transfer. It implements an acknowledged data transfer in the MAC to make sure that data has been received by peer. If the acknowledgement is not received within a certain time-out period, the transmitter will retry the transmission for a fixed number of times before declaring an error. Reliable delivery is supported at the transport layer level as well to make sure that the data reached the end destination device. The LightBee stack also supports flagging packets as 'high priority' and processing them accordingly.

LightBee also support broadcast data transfer and group addressing that allows sending data to multiple devices.

Interference Mitigation

Transmissions are monitored by the coordinator and router devices. If the percentage of transmission failures at a router node reaches a certain threshold, it sends an interference detection announcement message to the coordinator. The coordinator may then select another channel for operation and reset the network by broadcasting the new channel to all the nodes in the network.

3 Footprint

The LightBee stack for the end-device, router, and coordinator nodes is implemented as ANSI-C code. It has been developed for an 8-bits micro-controller. The footprint and required memory are provided in Table 1. The code is portable and can be ported to multiple micro-controller platforms.

Table 1: Footprint and RAM requirement for the different types of nodes

	Footprint	**RAM**
End Device	6K	< 200 bytes
Router	18K	< 800 bytes
Coordinator	16K	< 800 bytes

4 Summary

The LightBee stack is a highly optimized stack developed as a small footprint yet feature-rich stack that is targeted for low-cost large scale deployment of sensor network applications. The stack is suitable for deployment in several markets such as home automation, building automation, industrial plant monitoring…etc. The LightBee protocol stack is feature-packed and supports the following:

– Definition of various device types: Network Coordinator, Router and End Devices.
– Support of applications that require low latency through the star network topology.
– Support of applications that require reliability and scalability through the mesh network topology and the associated mesh routing method.
– Support of the cluster-tree network topology.
– Support of multiple applications objects or multiple objects of the same application in the same device.
– Support of "Device Discovery" service that allows devices of the same applications in the network to discover each other and have a logical link.
– Reliable data delivery at the MAC and transport layers.
– Flagging data as high priority.
– Application data fragmentation/defragmentation.
– Support of broadcast group addressing (multicast) services.
– Supports of device roaming.
– Proactive interference detection and mitigation.
– Allowing devices to connect and disconnect from the network dynamically.
– The stack is available as ANSI C code and has been developed using Cypress PSoC® platform. Extensive Documentation and training material are provided.