UG103.11

APPLICATION DEVELOPMENT FUNDAMENTALS: THREAD

This document includes a brief background on the emergence of Thread, provides a technology overview, and describes some key features of Thread to consider when implementing a Thread solution.

Contents

1 Introduction

1.1 Silicon Labs and the Internet of Things

IPv4 (Internet Protocol version 4) was defined in 1981[1]. Using 32-bit (4-byte) addressing, IPv4 provided 2^{32} unique addresses for devices on the internet, a total of approximately 4.3 billion addresses. However, as the number of users and devices grew exponentially, it was clear that the number of IPv4 addresses would be exhausted and there was a need for a new version of the IP. Hence the development of IPv6 in the 1990s and its intention to replace IPv4. With 128-bit (16-byte) addressing, IPv6 allows for 2^{128} addresses, more than 7.9×10^{28} addresses than IPv4.[2]

The challenge for companies in the embedded industry like Silicon Labs is to address this technology migration and more importantly the demands of customers as we move to an ever-connected world of devices in the home, what is often referred to as the Internet of Things (IoT). At a high level the goals of IoT for Silicon Labs are to:

- Connect all the devices in the home with best-in-class mesh networking, whether with Ember ZigBee PRO today or other emerging standards.
- Leverage the company's expertise in low-power, constrained devices.
- Enhance established low-power, mixed-signal chips.
- Provide low-cost bridging to existing Ethernet and Wi-Fi devices.
- Enable cloud services and connectivity to smartphones and tablets that will promote ease of use and a common user experience for customers.

Achieving all of these goals will increase adoption rates and user acceptance for IoT devices in the connected home.

1.2 Silicon Labs and Wireless Mesh Networking

ZigBee (www.zigbee.org) is the preeminent wireless mesh networking solution on the market today with a global installed base of more than 50 million units shipped in 2014. Silicon Labs is an industry leading supplier and innovator of ZigBee solutions. While ZigBee has been the dominant 802.15.4 standard for many years, there has been a market pull for a new IP-based wireless mesh networking standard. Customers are demanding direct connectivity to all their devices, specifically in their Connected Home. At the same time, application developers are seeking a simpler IP development model for the cloud, smartphones, and tablets.

Thread Group (www.threadgroup.org) and Silicon Labs have developed Thread to meet these needs. Common hardware platforms can be utilized for both ZigBee and Thread devices. The Silicon Labs EM35x 802.15.4 wireless SoCs (Systems on Chips) support both mesh networking protocols.

There are two common hardware platforms and application development environments:

- EM35x 802.15.4 wireless SoC using a C-based API
- NCP (Network Co-Processor) IP Modem using UART (Universal Asynchronous Receiver/Transmitter) or SPI (Serial Peripheral Interface)

Because of this common hardware, there is a defined migration path for ZigBee PRO solutions to Thread, as well as for 802.15.4 chips that have no applications built on top of them.

Silicon Labs is committed to supplying and supporting best-in-class Thread and ZigBee solutions. Ember Desktop and Insight Adapter will be available on both platforms. Application developers will be able to use App Framework to develop their applications efficiently. In addition, Silicon Labs will continue to demonstrate leadership and involvement in both the ZigBee Alliance and Thread Group.

[1] http://tools.ietf.org/html/rfc791: *DARPA Internet Program Protocol Specification*
[2] http://en.wikipedia.org/wiki/IPv6

1.3 Thread Group

Thread Group was launched on July 15, 2014 and Silicon Labs is a founding company along with six other companies. Refer to http://www.threadgroup.org/About.aspx for a complete list. Thread Group is a market education group that offers product certification and promotes the use of Thread-enabled D2D (device-to-device) and M2M (machine-to-machine) products. Membership in Thread Group is open (www.threadgroup.org) and offers these benefits to members:

- Access to technology and technical documentation
- Use of the Thread Certification Program and test suite
- Participation in marketing and public relations campaigns

1.4 What is Thread?

Thread is a secure, wireless mesh networking protocol. The Thread stack is an open standard that is built upon a collection of existing IEEE (Institute for Electrical and Electronics Engineers) and IETF (Internet Engineering Task Force) standards, rather than a whole new standard (see Figure 1).

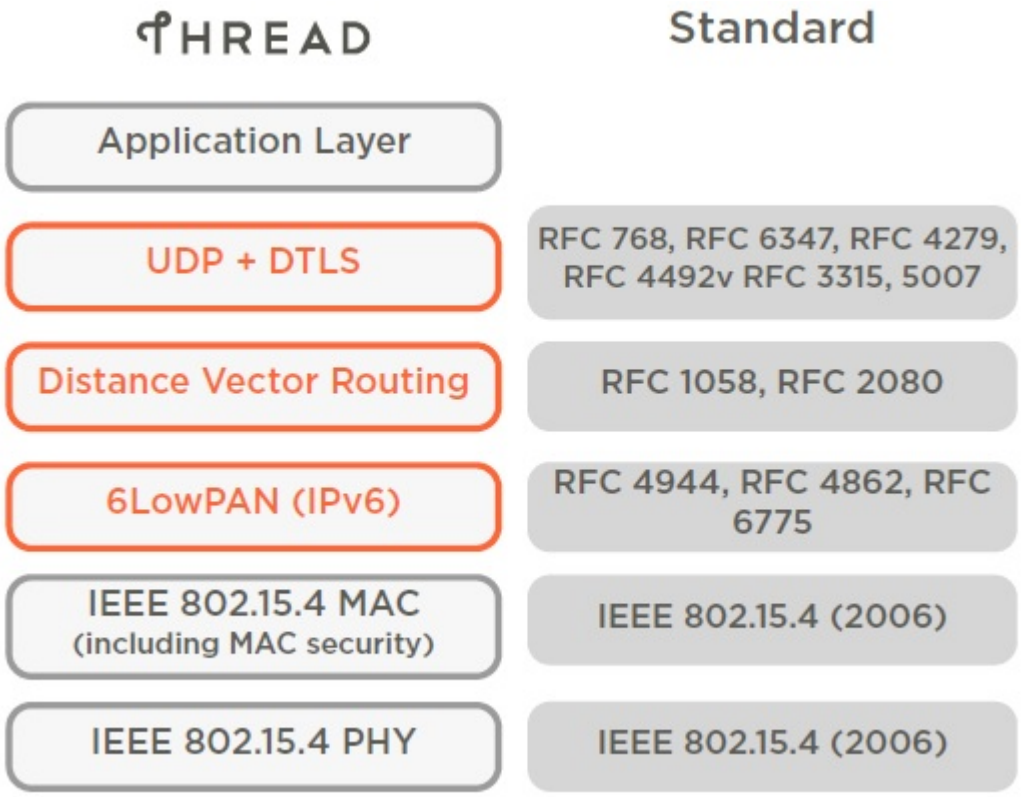

Figure 1. Thread Stack Overview

1.5 Thread General Characteristics

The Thread stack supports IPv6 addresses and provides low-cost bridging to other IP networks and is optimized for low-power / battery-backed operation, and wireless device-to-device communication. The Thread stack is designed specifically for Connected Home applications where IP-based networking is desired and a variety of application layers can be used on the stack.

These are the general characteristics of the Thread stack focused on the Connected Home:

- **Simple network installation, start-up, and operation:** The Thread stack supports several network topologies. Installation is simple using a smartphone, tablet, or computer. Product installation codes are used to ensure only authorized devices can join the network. The simple protocols for forming and joining networks allow systems to self-configure and fix routing problems as they occur.
- **Secure:** Devices do not join the network unless authorized and all communications are encrypted and secure. Security is provided at the network layer and can be at the application layer. All Thread networks are encrypted using a smartphone-era authentication scheme and AES (Advanced Encryption Standard) encryption. The security used in Thread networks is stronger than other wireless standards the Thread Group has evaluated.
- **Small and large networks:** Home networks vary from several devices to hundreds of devices communicating seamlessly. The networking layer is designed to optimize the network operation based on the expected use.
- **Range:** Typical devices provide sufficient range to cover a normal home. Readily available designs with power amplifiers extend the range substantially. A distributed spread spectrum is used at the PHY (Physical Layer) to be more immune to interference.
- **No single point of failure:** The Thread stack is designed to provide secure and reliable operations even with the failure or loss of individual devices.
- **Low power:** Devices efficiently communicate to deliver an enhanced user experience with years of expected life under normal battery conditions. Devices can typically operate for several years on AA type batteries using suitable duty cycles.
- **Cost-effective:** Compatible chipsets and software stacks from multiple vendors are priced for mass deployment, and designed from the ground up to have extremely low-power consumption. Typical home products run in the Connected Home include: normally powered (lighting, appliances, HVAC, fans); powered or battery-operated (thermostats, smoke detectors, CO and CO_2 detectors, security systems); and normally battery-operated (door sensors, window sensors, motion sensors, door locks).

2 Thread Technology Overview

2.1 IEEE 802.15.4

The IEEE 802.15.4-2006 specification[3] is a standard for wireless communication that defines the wireless MAC (Medium Access Control) and PHY (Physical) layers operating at 250 kbps in the 2.4 GHz band, with a roadmap to subGHz bands. Designed with low power in mind, 802.15.4 is suitable for applications usually involving a large number of nodes.

The 802.15.4 MAC layer is used for basic message handling and congestion control. This MAC layer includes a CSMA (Carrier Sense Multiple Access) mechanism for devices to listen for a clear channel, as well as a link layer to handle retries and acknowledgement of messages for reliable communications between adjacent devices. MAC layer encryption is used on messages based on keys established and configured by the higher layers of the software stack. The network layer builds on these underlying mechanisms to provide reliable end-to-end communications in the network.

One of the characteristics derived from the need for low power and limiting the BER (Bit Error Rate) is enforcing smaller sized packets to be sent over the air. These can be up to a maximum of 127 bytes at the PHY layer. The MAC layer payload can vary depending on the security options and addressing type as illustrated in Figure 2.

[3] IEEE 802.15.1-2006 Specification, http://standards.ieee.org/findstds/standard/802.15.4-2006.html

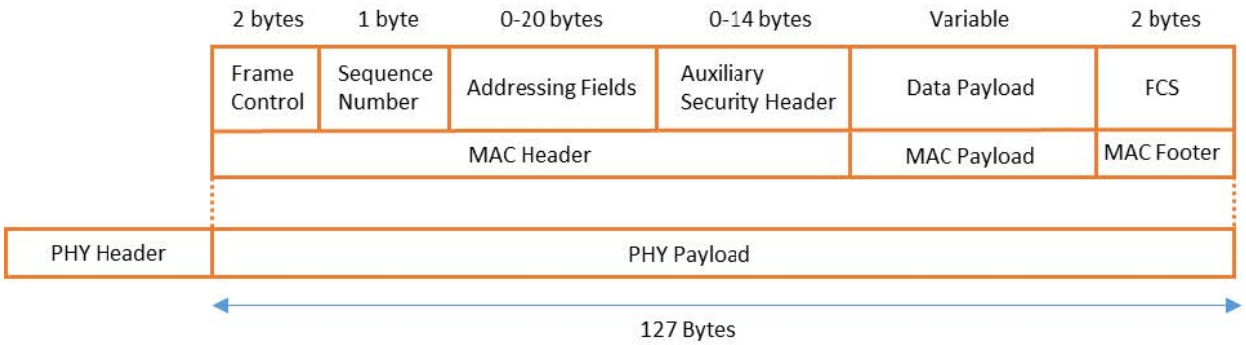

Figure 2. 802.15.4 MAC Payload

2.2 Thread Network Architecture

Users communicate with a Thread network from their own device (smartphone, tablet, or computer) via Wi-Fi on their HAN (Home Area Network) or using a cloud-based application. Figure 3 illustrates the key device types in the Thread network architecture.

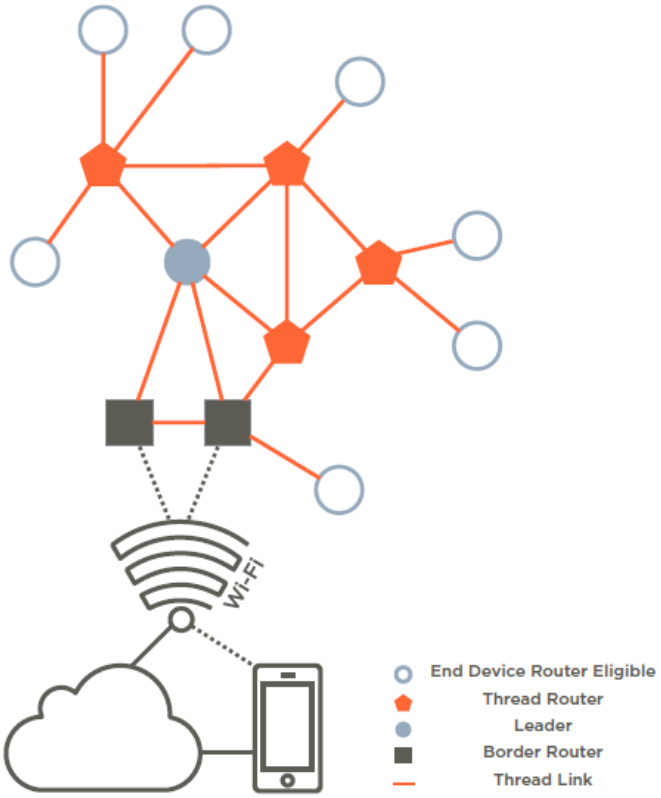

Figure 3. Thread Network Architecture

The following device types are included in a Thread network, starting from the Wi-Fi network:

- **Border Routers** provide connectivity from the 802.15.4 network to adjacent networks on other physical layers (Wi-Fi, Ethernet, etc.). Border Routers provide services for devices within the 802.15.4 network, including routing services for off network operations. There may be one or more Border Routers in a Thread network.
- A **Leader** manages a registry of assigned router IDs and accepts requests from router-eligible devices to become routers or vice versa. The Leader decides which should be routers. The Leader also assigns and manages router addresses using DHCPv6 (Dynamic Host Configuration Protocol version 6). However, all information contained in the Leader is present in the other Thread Routers. So, if the Leader fails or loses connectivity with the Thread network, another Thread Router is elected, and takes over as Leader without user intervention.
- **Thread Routers** provide routing services to network devices. Thread Routers also provide joining and security services for devices trying to join the network. Thread Routers are not designed to sleep and can downgrade their functionality and become router-eligible devices.
- **Router-eligible devices** can become a Thread Router or a Leader, but not necessarily a Border Router that has special properties, such as multiple interfaces. Because of the network topology or other conditions, router-eligible devices are not acting as routers. Router-eligible devices do not relay messages or provide joining or security services for other devices in the network. The network manages and promotes router-eligible devices to routers if necessary without user interaction.
- **Sleepy end devices** communicate only through their Thread Router and cannot relay messages for other devices.

2.3 No Single Point of Failure

The Thread stack is designed to not have a single point of failure. While there are a number of devices in the system that perform special functions, Thread is designed so they can be replaced without impacting the ongoing operation of the network or devices. For example, a sleepy end device requires a parent for communications so this parent represents a single point of failure for its communications. However, the sleepy end device can and will select another parent if its parent is unavailable. This transition should not be visible to the user.

While the system is designed for no single point of failure, under certain topologies there will be individual devices that do not have backup capabilities. For example, in a system with a single Border Router, if the Border Router loses power, there is no means to switch to an alternative Border Router.

3 IP Stack Fundamentals

3.1 Addressing

Devices in the Thread stack support IPv6 addressing architecture as defined in [RFC 4291].[4] ("RFC" stands for "Request for Comments.") Devices support one or more ULA (Unique Local Address) or GUA (Global Unicast Address) addresses based on their available resources.

The high-order bits of an IPv6 address specify the network and the rest specify particular addresses in that network. Thus, all the addresses in one network have the same first N bits. Those first N bits are called the "prefix". The "/64" indicates that this is an address with a 64-bit prefix. The device starting the network picks a /64 prefix that is then used throughout the network. The prefix is a ULA [RFC 4193].[5] The network may also have one or more Border Router(s) that each may or may not have a /64 that can then be used to generate a ULA or GUA. The

[4] https://tools.ietf.org/html/rfc4291: *IP Version 6 Addressing Architecture*
[5] https://tools.ietf.org/html/rfc4193: *Unique Local IPv6 Unicast Addresses*

device in the network uses its EUI-64 (64-bit Extended Unique Identifier) address to derive its interface identifier as defined in Section 6 of [RFC 4944][6]. The device will support a link local IPv6 address configured from the EUI-64 of the node as an interface identifier with the well-known link local prefix FE80::0/64 as defined in [RFC 4862][7] and [RFC 4944].

The devices also support appropriate multicast addresses. This includes link-local all node multicast, link local all router multicast, solicited node multicast, and a mesh local multicast.

Each device joining the network is assigned a 2 byte short address as per the IEEE 802.15.4-2006 specification. For routers, this address is assigned using the high bits in the address field. Children are then assigned a short address using their parent's high bits and the appropriate lower bits for their address. This allows any other device in the network to understand the child's routing location by using the high bits of its address field (see Figure 4).

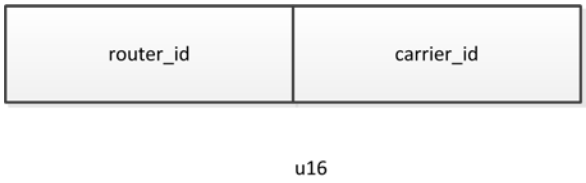

Figure 4. Thread Short Address

3.2 6LoWPAN

6LoWPAN stands for "IPv6 Over Low Power Wireless Personal Networks." The main goal of 6LoWPAN is to transmit and receive IPv6 packets over 802.15.4 links. In doing so it has to accommodate for the 802.15.4 maximum frame size sent over the air. In Ethernet links, a packet with the size of the IPv6 MTU (Maximum Transmission Unit) (1280 bytes) can be easily sent as one frame over the link. In the case of 802.15.4, 6LoWPAN acts as an adaptation layer between the IPv6 networking layer and the 802.15.4 link layer. It solves the issue of transmitting an IPv6 MTU by fragmenting the IPv6 packet at the sender and reassembling it at the receiver.

6LoWPAN also provides a compression mechanism that reduces the IPv6 header sizes sent over the air and thus reduces transmission overhead. The fewer bits that are sent over the air, the less energy is consumed by the device. Thread makes full use of these mechanisms to efficiently transmit packets over the 802.15.4 network. [RFC 4944] and [RFC 6282] describe in detail the methods by which fragmentation and header compression are accomplished.

All devices use 6LoWPAN as defined in [RFC 4944]. Header compression is used within the Thread network and devices transmitting messages compress the IPv6 header as much as possible to minimize the size of the transmitted packet.

The mesh header is supported for more efficient compression of messages within the Thread network and for link layer forwarding. The mesh header also allows end-to-end fragmentation of messages rather than hop-by-hop fragmentation specified in [RFC 4944]. The Thread stack uses route-over configuration which gives nodes IP-level visibility into the underlying radio connectivity.

The devices do not support neighbor discovery as specified in [RFC 6775][8] because DHCPv6 is used to assign short addresses to routers. Host devices are allocated addresses from their router parent and these are derived from the parent router address. With 6LoWPAN this short address is then used to generate the necessary ULA's that may be used.

[6] https://tools.ietf.org/html/rfc4944: *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*
[7] https://tools.ietf.org/html/rfc4862: *IPv6 Stateless Address Autoconfiguration*
[8] https://tools.ietf.org/html/rfc6775: *Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)*

3.3 Link Layer Forwarding

Another important feature of the 6LoWPAN layer is link layer packet forwarding. This provides a very efficient and low overhead mechanism for forwarding multi hop packets in a mesh network. Thread uses IP layer routing with link layer packet forwarding.

Thread uses the link layer forwarding to forward packets based on the IP routing table. In order to accomplish this, the 6LoWPAN mesh header is used in each multi-hop packet (Figure 5).

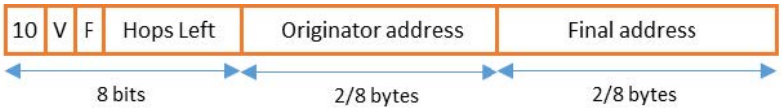

Figure 5. Mesh Header Format

In Thread, the 6LoWPAN layer fills the Mesh Header information with the originator 16-bit short address and final destination 16-bit source address. The transmitter looks-up the next hop 16-bit short address in the Routing Table, and then sends the 6LoWPAN frame to the next hop 16-bit short address as destination. The next hop device receives the packet, looks up the next hop in the Routing Table / Neighbor Table, decrements the hop count in the 6LoWPAN Mesh Header, and then sends the packet to the next hop or final destination 16-bit short address as destination.

3.4 6LoWPAN Encapsulation

6LoWPAN packets are constructed on the same principle as IPv6 packets and contain stacked headers for each added functionality. Each 6LoWPAN header is preceded by a dispatch value that identifies the type of header (Figure 6).

Figure 6. General Format of a 6LoWPAN Packet

Thread uses the following types of 6LoWPAN headers:

- Mesh Header (used for link layer forwarding)
- Fragmentation Header (used for fragmenting the IPv6 packet into several 6LoWPAN packets)
- Header Compression Header (used for IPv6 headers compression)

The 6LoWPAN specification mandates that if more than one header is present they must appear in the order mentioned above.

The following are examples of 6LoWPAN packets sent over the air.

In Figure 7, the 6LoWPAN payload is composed of the compressed IPv6 header and the rest of the IPv6 payload.

Figure 7. 6LoWPAN Packet Containing IPv6 Payload with Compressed IPv6 Header

In Figure 8, the 6LoWPAN payload contains the IPv6 header and part of the IPv6 payload.

Figure 8. 6LoWPAN Packet Containing Mesh Header, a Fragmentation Header, and a Compression Header

The rest of the payload will be transmitted in subsequent packets per the format in Figure 9.

Figure 9. 6LoWPAN subsequent fragment

3.5 ICMP

Thread devices support the ICMPv6 (Internet Control Message Protocol version 6) protocol as defined in [RFC 4443][9] and ICMPv6 error messages. They also support the echo request and echo reply messages.

3.6 UDP

The Thread stack supports UDP (User Datagram Protocol) as defined in [RFC 768][10] for messaging between devices.

4 Network Topology

4.1 Network Address and Devices

The Thread stack supports full mesh connectivity between all routers in the network. The actual topology is based on the number of routers in the network. If there is only one router then the network forms a star. If there is more than one router then a mesh is automatically formed (see Figure 3).

4.2 Mesh Networks

Embedded mesh networks make radio systems more reliable by allowing radios to relay messages for other radios. For example, if a node cannot send a message directly to another node, the embedded mesh network relays the message through one or more intermediary nodes. As discussed in Section 5.3, **Routing**, all router nodes in the Thread stack maintain routes and connectivity with each other so the mesh is constantly maintained and connected. There is a limit of 64 router addresses in the Thread network, but they cannot all be used at once. This allows time for the addresses of deleted devices to be reused.

In a mesh network, the sleepy end devices or router-eligible devices do not route for other devices. These devices send messages to a parent that is a router. This parent router handles the routing operations for its child devices.

5 Routing and Network Connectivity

The Thread network has up to 32 active routers that use next-hop routing for messages based on the routing table. The routing table is maintained by the Thread stack to ensure all routers have connectivity and up-to-date paths for any other router in the network. All routers exchange with other routers their cost of routing to other routers in the network in a compressed format using MLE (Mesh Link Establishment).

[9] https://tools.ietf.org/html/rfc4443: *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*
[10] https://tools.ietf.org/html/rfc768: *User Datagram Protocol*

5.1 MLE Messages

MLE messages are used to establish and configure secure radio links, detect neighboring devices, and maintain routing costs between devices in the network. MLE operates below the routing layer and uses one hop link local unicasts and multicasts between routers.

MLE messages are used to identify, configure, and secure links to neighboring devices as the topology and physical environment change. MLE is also used to distribute configuration values that are shared across the network such as the channel and PAN (Personal Area Network) ID. These messages can be forwarded with simple flooding as specified by MPL (Multicast Protocol for Low Power and Lossy Networks).[11]

MLE messages also ensure asymmetric link costs are considered when establishing routing costs between two devices. Asymmetric link costs are common in 802.15.4 networks. To ensure two-way messaging is reliable, it is important to consider bidirectional link costs.

5.2 Route Discovery and Repair

On-demand route discovery is commonly used in low-power 802.15.4 networks. However, on-demand route discovery is costly in terms of network overhead and bandwidth because devices broadcast route discovery requests through the network. In the Thread stack, all routers exchange one-hop MLE packets containing cost information to all other routers in the network. All routers have up-to-date path cost information to any other router in the network so on-demand route discovery is not required. If a route is no longer usable, the routers can select the next most suitable route to the destination.

Routing to child devices is done by looking at the high bits of the child's address to determine the parent router address. Once the device knows the parent router, it knows the path cost information and next hop routing information for that device.

As route cost or the network topology change, the changes propagate through the network using the MLE single-hop messages. Routing cost is based on bidirectional link quality between two devices. The link quality in each direction is based on the link margin on incoming messages from that neighboring device. This incoming RSSI (Received Signal Strength Indicator) is mapped to a link quality from 0 to 3. A value of 0 means unknown cost.

When a router receives a new MLE message from a neighbor, either it already has a neighbor table entry for the device or one is added. The MLE message contains the incoming cost from the neighbor so this is updated in the router's neighbor table. The MLE message also contains updated routing information for other routers which is updated in the routing table.

The number of active routers is limited to the amount of routing and cost information that can be contained in a single 802.15.4 packet. This limit is currently 32 routers.

5.3 Routing

Devices use normal IP routing to forward packets. A routing table is populated with network addresses and the appropriate next hop.

Distance vector routing is used to get routes to addresses that are on the local network. When routing on the local network, the upper six bits of this 16-bit address define the router destination. This routing parent is then responsible for forwarding to the final destination based on the remainder of the 16-bit address.

For off network routing, a Border Router notifies the Router Leader of the particular prefixes it serves and distributes this information as network data within the MLE packets. The network data includes prefix data which is the prefix itself, the 6lo context, the Border Routers and the DHCPv6 server for that prefix. If a device is to configure an address using that prefix, it contacts the appropriate DHCP server for this address. The network data also includes a list of routing servers that are the 16-bit addresses of the default Border Routers.

[11] https://tools.ietf.org/html/draft-ietf-roll-trickle-mcast-11: *Multicast Protocol for Low power and Lossy Networks (MPL)*

A Leader is designated to keep track of router-eligible devices becoming routers or allowing routers to downgrade to router-eligible devices. This Leader also assigns and manages the router addresses using DHCPv6. However, all information contained in this Leader is present in the other routers. If the Leader goes off the network, another router is elected, and takes over as Leader without user intervention.

The Border Router is responsible for handling 6LoWPAN compression or expansion and addressing to off network devices.

5.4 Retries and Acknowledgements

While UDP messaging is used in the Thread stack, reliable message delivery is required and completed by these lightweight mechanisms:

- MAC-level retries–each device uses MAC acknowledgements from the next hop and will retry a message at the MAC layer if the MAC ACK message is not received.
- Application-layer retries– the application layer can determine if message reliability is a critical parameter. If so, an end-to-end acknowledgement and retry protocol can be used.

6 Joining a Network

Thread allows two joining methods:

- Share commissioning information directly to a device using an out-of-band method. This allows steering the device to the proper network using this information.
- Establish a commissioning session between a joining device and a commissioning application on a smartphone, tablet, or the web.

The frequently used 802.15.4 method of joining with the permit joining flag in the beacon payload is not used in Thread networks. This method is most commonly used for push button type joining where there is no user interface or out-of-band channel to devices. This method has issues with device steering in situations where there are multiple networks available and it can also pose security risks.

In Thread networks, all joining is user-initiated. After joining, a security authentication is completed at the application level with a commissioning device. This security authentication is discussed in Section 9, **Security**.

Devices join a network as either a sleepy end device or a REED (Router-Eligible End Device). Only after a REED has joined and learned the network configuration, can it potentially request to become a Thread Router. Upon joining, a device is provided a 16-bit short address based on its parent. If a router-eligible device becomes a Thread Router, it is assigned a router address by the Leader. Duplicate address detection for Thread Routers is ensured by the centralized router address distribution mechanism which resides on the Leader. The parent is responsible for avoiding duplicate addresses for host devices because it assigns addresses to them upon joining.

6.1 Network Discovery

Network discovery is used by a joining device to determine what 802.15.4 networks are within radio range. The device scans all channels, issues a beacon request on each channel, and waits for beacon responses. The 802.15.4 beacon response contains a payload with network parameters, including the network SSID (Service Set Identifier) and a permit joining beacon that indicates if the network is accepting new members.

Network discovery is not required if the device has been commissioned onto the network because it knows the channel and extended PAN ID for the network. These devices attach to the network using the commissioning material provided.

6.2 MLE Data

Once a device has attached to a network, there is a variety of information required for it to participate in the network. MLE provides services for a device to send a unicast to a neighboring device to request network parameters and update link costs to neighbors. When a new device joins, it also conducts a challenge response to set security frame counters as discussed in Section 9, **Security**.

All devices support transmission and reception of MLE link configuration messages. This includes link request, link accept, link accept and request, and link reject messages.

The MLE exchange is used to configure or exchange the following information:

- The 16-bit short and 64 bit EUI 64 long address of neighboring devices
- Device capabilities information, including if it is a sleepy end device and the sleep cycle of the device
- Neighbor link costs if a Thread Router
- Security material and frame counters between devices
- Routing costs to all other Thread Routers in the network
- Updates to network parameters such as the channel, PAN ID, permit joining parameter, and beacon payload parameter used in the MAC.

Note: MLE messages are encrypted except during the initial node bootstrapping operations when the new device has not obtained the security material.

6.3 DHCPv6

DHCPv6 (Dynamic Host Configuration Protocol for IPv6) as defined in [RFC 3315][12] is used as a client-server protocol to manage configuration of devices within the network. DHCPv6 uses UDP to request data from a DHCP server.

The DHCPv6 service is used for configuration of:

- Network addresses
- Multicast addresses required by devices

Because short addresses are assigned from the server using DHCPv6, duplicate address detection is not required. DHCPv6 is also used by Border Routers that are assigning addresses based on the prefix they provide.

7 Management

7.1 ICMP

All devices support ICMPv6 (Internet Control Message Protocol for IPv6) error messages, as well as the echo request and echo reply messages.

7.2 Device Management

The application layer on a device has access to a set of device management and diagnostics information that can be used locally or collected and sent to other management devices.

At the 802.15.4 PHY and MAC layers, the device provides the following information to the management layer:

- EUI 64 address
- 16-bit short address

[12] https://www.ietf.org/rfc/rfc3315.txt: *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*

- Capability information
- PAN ID
- Packets sent and received
- Octets sent and received
- Packets dropped on transmit or receive
- Security errors
- Number of MAC retries

7.3 Network Management

The network layer on the device also provides information on management and diagnostics that can be used locally or sent to other management devices. The network layer provides the IPv6 address list, the neighbor and child table, and the routing table.

8 Persistent Data

Devices operating in the field may be reset accidentally or on purpose for a variety of reasons. Devices that have been reset need to restart network operations without user intervention. For this to be done successfully, non-volatile storage must store the following information:

- Network information such as PAN ID
- Security material (each key used)
- Addressing information from the network to form the IPv6 addresses for the devices

9 Security

Thread Networks are wireless networks that need to be secured against OTA (over-the-air) attacks. They are also connected to the internet and therefore must be secured against internet attacks.

For wireless security, Thread uses standard IEEE 802.15.4 authentication and encryption. For increased security, Thread implements an additional handshake between each pair of communicating nodes. (See Section 2.2, **Thread Network Architecture** for a description of the key device types in the Thread network architecture.) For internet security, Thread provides end-to-end addressing and message forwarding. The two endpoints are in direct communication, and any internet security protocol can be used.

Network management also needs to be secure. A Thread network management application can be run on any internet device. If that device is not itself a member of a Thread network, it must first establish a secure TLS (Transit Layer Security) connection with a Thread Border Router. Every Thread network has a management passphrase that is used for this connection. Once a management application has been connected to the Thread network, new devices can be added to the network.

10 Application Layer

Thread is a wireless mesh networking stack that is responsible for routing messages between different devices in the Thread network described in Section 2.2, **Thread Network Architecture**. A standard definition of an application layer is an “abstraction layer that specifies the shared protocols and interface methods used by hosts in a communications network.”[13] Put more simply, an application layer is the “language of devices,” for example, how a switch talks to a light bulb.

[13] https://en.wikipedia.org/wiki/Application_layer

Using these definitions, an application layer does not exist in Thread. Customers build the application layer based on the requirements in the Thread specification and their own user requirements. Figure 10 illustrates the layers in the Thread protocol.

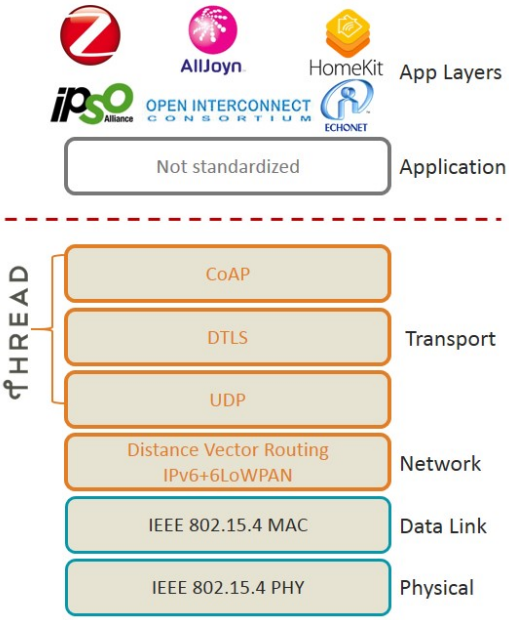

Figure 10. Thread Protocol Layers

The Thread specification defines standard methods for forming and joining a network (called commissioning) and custom applications are responsible for interoperability. Thread does, however, provide these basic application services:

- UDP messaging

  UDP offers a way to send messages using a 16-bit port number and an IPv6 address. UDP is a simpler protocol than TCP and has less connection overhead (for example, UDP does not implement keep-alive messages). As a result, UDP enables a faster, higher throughput of messages and reduces the overall power budget of an application. UDP also has a smaller code space than TCP, which leaves more available flash on the chip for custom applications.

- Multicast messaging

  Thread provides the ability to broadcast messages, that is, sending the same message to multiple nodes on a Thread network. Multicast allows a built-in way to talk to neighbor nodes, routers, and an entire Thread network with standard IPv6 addresses.

- Application layers using IP services

  Thread allows the use of application layers such as UDP and CoAP (Constrained Application Protocol) to allow devices to communicate interactively over the Internet. Non-IP application layers will require some adaptation to work on Thread. (See RFC 7252 for more information on CoAP.)

Silicon Labs has developed two sample applications–client and server–that demonstrate basic interoperability features of the Thread network and how to build a simple client and server example, including a sleepy end device.

Customers can leverage the Silicon Labs AppBuilder tool to help create and configure these two sample applications. AppBuilder provides a number of application building blocks, called plugins, such as debug message support and a CLI (Command Line Interface). These plugins save customers time from custom development for common tasks.

11 Next Steps

The Thread software includes a certified Thread networking stack and sample applications that demonstrate basic network and application behavior. Customers are encouraged to use the included sample applications to gain familiarity with Thread in general and the Silicon Labs offering in particular. Each of the applications demonstrates how devices form and join networks, as well as how messages are sent and received. The applications are available for use after loading the Thread installation in Ember AppBuilder. Ember Desktop includes support for decoding the network- and application-layer messages in Thread and provides additional insight into the operation of Thread networks.

CONTACT INFORMATION

Silicon Laboratories Inc.

400 West Cesar Chavez
Austin, TX 78701
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032

For additional information please visit the Silicon Labs Technical Support page:
http://www.silabs.com/support/Pages/default.aspx.

Patent Notice

Silicon Labs invests in research and development to help our customers differentiate in the market with innovative low-power, small size, analog-intensive mixed-signal solutions. Silicon Labs' extensive patent portfolio is a testament to our unique approach and world-class engineering team.