THREAD GROUP

July 13, 2015

This Thread Technical white paper is provided for reference purposes only.

The full technical specification is available to Thread Group Members. To join and gain access, please follow this link: http://threadgroup.org/Join.aspx.

If you are already a member, the full specification is available in the Thread Group Portal: http://portal.threadgroup.org.

If there are questions or comments on these technical papers, please send them to help@threadgroup.org.

THREAD GROUP

Thread Stack Fundamentals

July 2015

Revision History

Revision	Date	Comments
1.0	November 29, 2014	Initial Release
2.0	July 13, 2015	Public Release

Contents

Introduction

General Characteristics

The Thread stack is an open standard for reliable, cost-effective, low-power, wireless D2D (device-to-device) communication. It 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 and network:

- Simple network installation, start up and operation: The simple protocols for forming, joining, and maintaining Thread Networks allow systems to self-configure and fix routing problems as they occur.
- Secure: Devices do not join the Thread Network unless authorized and all communications are encrypted and secure.
- Small and large networks: Home networks vary from several devices to hundreds of devices communicating seamlessly. The network layer is designed to optimize the network operation based on the expected use.
- Range: Typical devices in conjunction with mesh networking provide sufficient range to cover a normal home. Spread spectrum technology is used at the physical layer to provide good immunity to interference.
- No single point of failure: The stack is designed to provide secure and reliable operations even with the failure or loss of individual devices.
- Low power: Host devices can typically operate for several years on AA type batteries using suitable duty cycles.

Figure 1 illustrates an overview of the Thread stack.

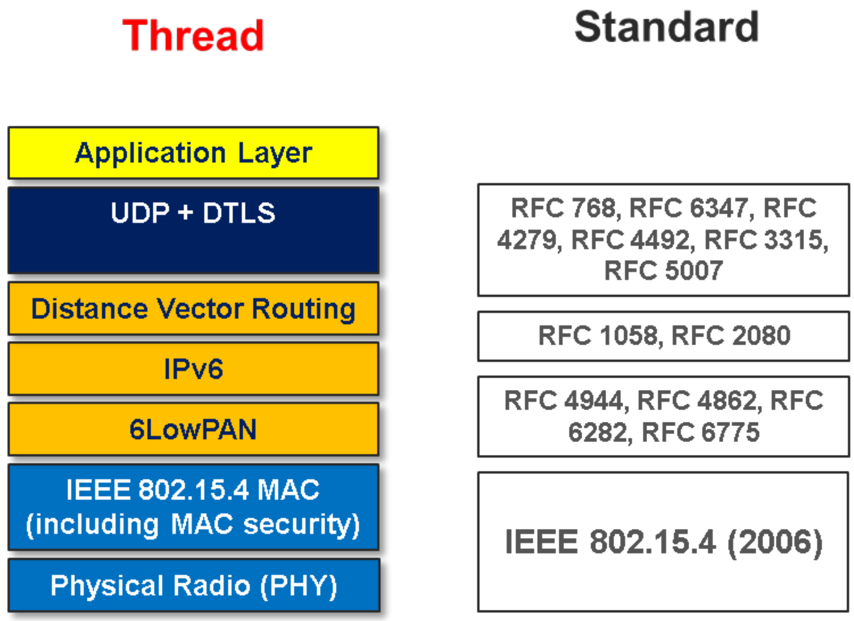

Figure 1. Overview of Thread Stack

IEEE 802.15.4

This standard is based on the IEEE 802.15.4 [IEEE802154] PHY (Physical) and MAC (Media Access Control) layers operating at 250 kbps in the 2.4 GHz band. The IEEE 802.15.4-2006 version of the specification is used for the Thread stack.

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 and integrity protection 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.

No Single Point of Failure

In a system comprised of devices running the Thread stack, none of these devices represents a single point of failure. While there are a number of devices in the system that perform special functions, the design of the Thread stack is such that they can be replaced without impacting the ongoing communication within the Thread Network. For example, a sleepy Child requires a Parent for communications so this Parent represents a single point of failure for its communications. However, the sleepy device can and will select another Parent if its Parent is unavailable so 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 gateway, if the gateway loses power, there is no means to switch to an alternative gateway.

A Router or Border Router can assume a Leader role for certain functions in the Thread Network. This Leader is required to make decisions within the network. For example, the Leader assigns Router addresses and allows new Router requests. The Leader role is elected and if the Leader fails, another Router or Border Router assumes the Leader role. It is this autonomous operation that ensures there is no single point of failure.

Device Types

Border Routers

A Border Router is a specific type of Router that provides connectivity from the 802.15.4 network to adjacent networks on other physical layers (for example, Wi-Fi and Ethernet). 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.

Routers

Routers provide routing services to network devices. Routers also provide joining and security services for devices trying to join the network. Routers are not designed to sleep. Routers can downgrade their functionality and become REEDs (Router-eligible End Devices).

Router-eligible End Devices

REEDs have the capability to become Routers but due to the network topology or conditions these devices are not acting as Routers. These devices do not generally forward messages or provide joining or security services for other devices in the Thread Network. The Thread Network manages REEDs becoming Routers if necessary without user interaction.

Sleepy End devices

Sleepy end devices are host devices. They communicate only through their Parent Router and cannot forward messages for other devices.

IP Stack Fundamentals

Addressing

Devices in the Thread stack support IPv6 addressing architecture specified in [RFC 4291]. Devices configure 1 or more ULA (Unique Local Address) or GUA (Global Unicast Address) addresses.

The device starting the network picks a /64 prefix that is then used throughout the Thread Network. The prefix is a Locally Assigned Global ID, often known as a ULA prefix [RFC 4193], and can be referred to as the mesh local ULA prefix. The Thread Network may also have one or more Border Routers that each may or may not have a prefix that can then be used to generate additional GUAs. The device in the Thread Network uses its Extended MAC address to derive its interface identifier as defined in section 6 of [RFC 4944] and from this configures a link local IPv6 address with the well-known local prefix FE80::0/64 as described in [RFC 4862] and [RFC 4944].

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

Each device joining the Thread Network is assigned a 16-bit short address as specified in [IEEE802154]. For Routers, this address is assigned using the high bits in the address field with the lower bits set to 0, indicating a Router address. Children are then allocated a 16-bit short address using their Parent's high bits and the appropriate lower bits for their address. This allows any other device in the Thread Network to understand the Child's routing location simply by using the high bits of its address field.

Figure 2 illustrates the Thread short address.

router_id	child_id
u16	

Figure 2. Thread Short Address

6LoWPAN

All devices use 6LoWPAN as defined in [RFC 4944] and [RFC 6282].

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 mesh and for link layer forwarding as discussed in the **Routing and Network Connectivity** section. 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.

The devices do not support neighbor discovery as specified in [RFC 6775] as DHCPv6 is used to allocate addresses to Routers. End devices and REEDs are allocated short addresses by their Router Parent. This short address is then used to configure the mesh local ULA that is used for intra-network communications.

Further details on 6LoWPAN usage and configuration are contained in the **"Thread Usage of 6LoWPAN"** white paper. Chapter 3 of the Thread specification details the specific 6LoWPAN configuration used.

ICMP

Devices support the ICMPv6 (Internet Control Message Protocol version 6) protocol [RFC 4443] and ICMPv6 error messages, as well as the echo request and echo reply messages.

UDP

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

Network Topology

Network Address and Devices

The Thread stack supports full mesh connectivity between all Routers in the Thread Network.

The actual topology is based on the number of Routers in the Thread Network. If there is only one Router or Border Router, then a basic star topology with a single Router is formed. If there is more than one Router then a mesh topology is automatically formed. Figure 3 illustrates the basic topology of a Thread Network and the types of devices.

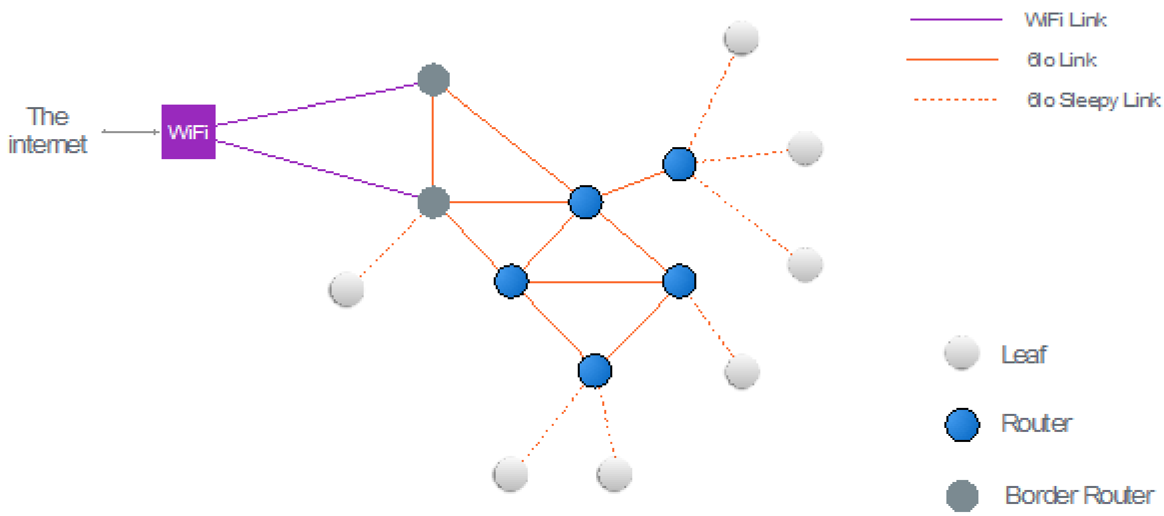

Figure 3. Basic Thread Network Topology and Devices

Mesh Networks

Mesh networks make radio systems more reliable by allowing radios to forward messages for other radios. For example, if a node cannot send a message directly to another node, the mesh network forwards the message through one or more intermediary nodes. As discussed in the **Routing and Network Connectivity** section, the nature of the Thread Network is that all Router nodes maintain routes and connectivity with each other so the mesh is constantly maintained and connected. There is typically a limit of 32 active Routers in the Thread Network. However, 64 Router addresses are used to allow recycling of Router addresses.

In a mesh network, the sleepy end devices or REEDs 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.

Routing and Network Connectivity

The Thread Network typically has up to 32 active Routers that use next-hop routing for messages based on the device routing table. The device routing table is maintained by the stack to ensure all Routers have connectivity and up-to-date paths for any other Router in the Thread Network. The RIP (Routing Information Protocol) algorithm is used (the algorithms from [RFC 1058] and [RFC 2080] but not the specific message formats). All Routers exchange with other Routers their cost of routing to other Routers in the Thread Network in a compressed format using MLE (Mesh Link Establishment).

Note: From an IP standpoint, the Thread Network supports Routers and hosts. Hosts are either sleepy end devices or REEDs.

MLE Messages

MLE messages (see [draft-kelsey-intarea-mesh-link-establishment-06] which is extended for Thread in Chapter 4, Message Link Establishment, of the Thread specification) are used for establishing and configuring secure radio links, detecting neighboring devices, and maintaining routing costs between devices in the Thread Network. MLE messages are transported using single-hop link local unicasts and multicasts between Routers.

MLE messages are used for identifying, configuring, and securing links to neighboring devices as the topology and physical environment change. MLE is also used to distribute configuration values that are shared across the Thread 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). (See [draft-ietf-roll-trickle-mcast-09] for more information.)

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 the costs of bidirectional links.

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 due to route discovery requests flooding the network.

In a Thread Network, all Routers periodically exchange single-hop MLE advertisement packets containing link cost information to all neighbor Routers, and path costs to all other Routers in the Thread Network. Through these periodic, local updates, all Routers have up-to-date path cost information to any other Router in the Thread Network, so on-demand route discovery is not required. If a route is no longer usable, Routers can make a selection on the next most suitable route to the destination. This self-healing routing mechanism allows Routers to quickly detect when other Routers have dropped off the Thread Network, and calculate the best path to maintain connectivity to all other devices in the Thread Network.

The link quality in each direction is based on the link cost on incoming messages from that neighboring device. This incoming link cost is mapped to a link quality from 0 to 3. A value of 0 means unknown cost. The link cost is a measure of RSSI (Received Signal Strength Indicator) of received messages above the receive level.

Table 1 summarizes the link quality and link cost.

Table 1. Link Quality and Link Cost

Link Quality	Link Cost
0	unknown
1	6
2	2
3	1

Figure 4 illustrates examples of various link costs on a Thread Network.

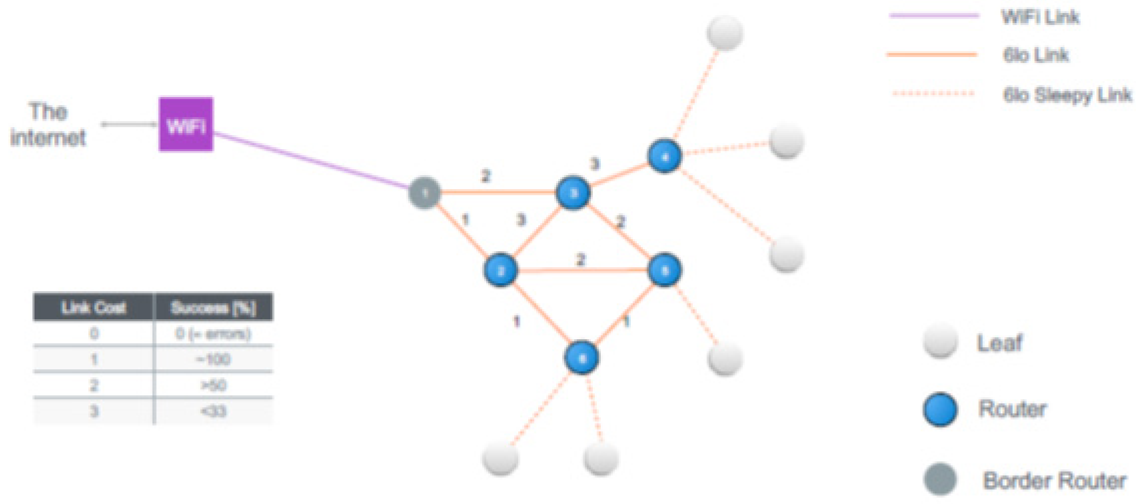

Figure 4. Examples of Various Link Costs on a Thread Network

The path cost to any other node in the Thread Network is then the minimum sum of link costs to reach that node. Routers monitor these costs, even as the radio link quality or topology of the network changes, and propagate the new costs through the Thread Network using the periodic MLE advertisement messages. Routing cost is based on bi-directional link quality between two devices.

To illustrate through a simplified example, imagine a pre-commissioned network with shared security material where all devices are powered on at the same time. Each Router would periodically send an advertisement initially populated only with costs to single-hop neighbors. Internally, each Router would store next hop information that is not sent in the advertisement.

The first few advertisements would have path cost equal to link cost, because the only Routers that are known are immediate neighbors as illustrated in Figure 5.

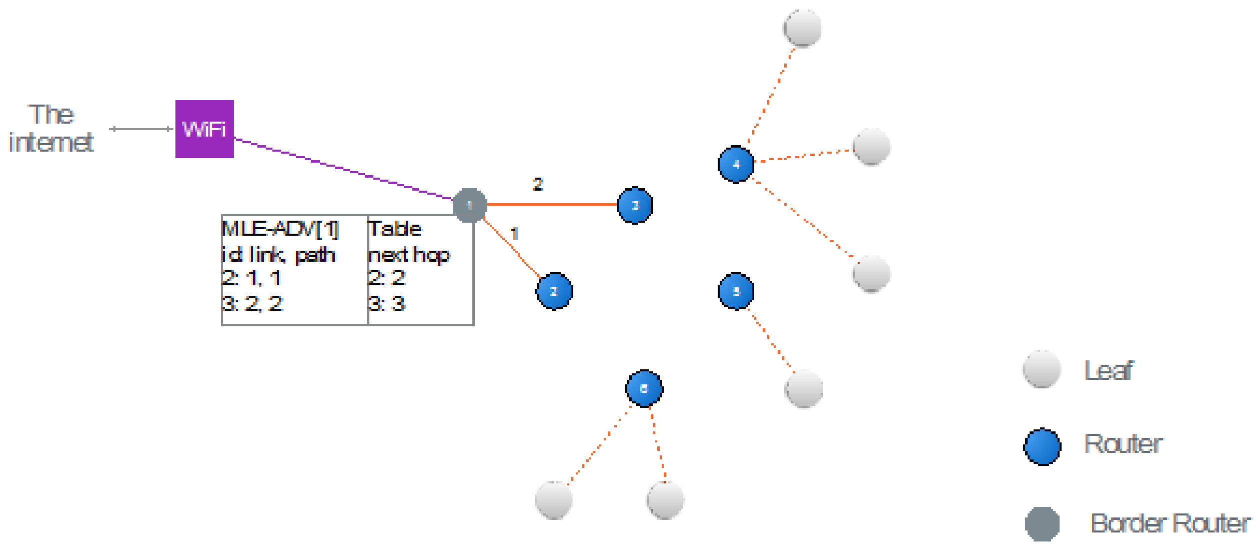

Figure 5. Forming a Thread Network after Power Cycle

But as Routers start hearing advertisements from their neighbors that contain costs to other Routers that are two or more hops away, their tables populate with multi-hop path costs which then propagate even farther out, until eventually there is connectivity information between all Routers in the network as illustrated in Figure 6 and Figure 7.

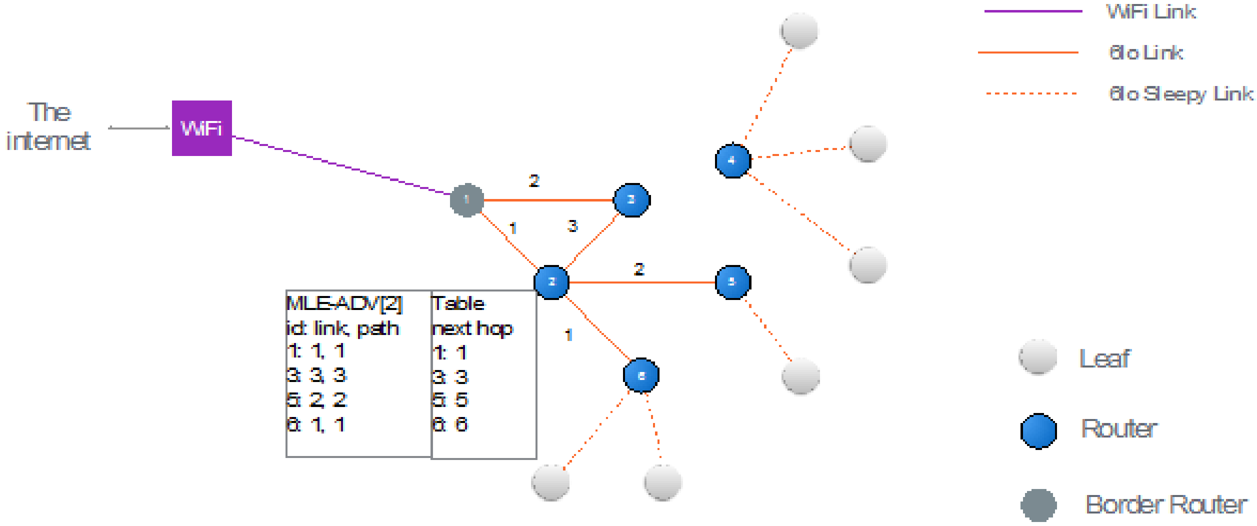

Figure 6. Thread Network Formation: Route Advertisements

Figure 7. Thread Network Formation: Multi-hop

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

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 has the path cost information and next-hop routing information for that device.

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 but 64 active Router addresses are provided to allow aging out of Router addresses.

Routing

Devices use IP routing to forward packets. A device routing table is populated with a compressed form of a mesh local ULA address for Routers and the appropriate next hop.

Distance vector routing is used to get routes to Router addresses that are on the Thread Network. When routing on the Thread Network, the upper 6 bits of this 16-bit address define the Router address of the Destination Router. If the lower bits of the destination address are 0, then the Router is the final destination. Otherwise, the Destination Router is responsible for forwarding to the final destination based on the lower bits of the 16-bit destination address.

For routing beyond the Thread Network, a Border Router notifies the Leader of the particular prefix(es) it serves and this information is distributed as Thread Network data within the MLE packets. The Thread Network data includes: prefix data, which is the prefix itself, the 6LoWPAN context, the Border Routers, and the DHCPv6 server for that prefix. If a device is to configure an IPv6 address using that prefix, it uses SLAAC (Stateless Address Autoconfiguration) or contacts the appropriate DHCP (Dynamic Host Configuration Protocol) server for this. The Thread Network data also includes a list of routing servers, which are the Router addresses of the default Border Routers.

A Leader is designated to make decisions on selecting REEDs to become Routers or allowing Routers to downgrade to REEDs. The Leader also assigns and manages the Router addresses. However, all information contained in this routing Leader is present in the other Routers and if the routing Leader becomes unreachable, another Router is autonomously elected and takes over as Leader without user intervention.

Retries and Acknowledgements

While UDP messaging is used in the Thread stack, reliable message delivery is still a requirement. This is done using a series of lightweight mechanisms as follows:

- 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 level retries: The application level can determine if message reliability is a critical parameter and may implement its own retry mechanism if necessary.

Joining a Thread Network

There are various phases a joining device has to go through before it can participate in a Thread Network:

- Discovery
- Commissioning
- Attaching

All joining is user-initiated in Thread Networks. Once joined, a device is fully participating in the Thread Network and can exchange application layer information with other devices and services within and beyond the Thread Network.

Discovery

A joining device has to discover the Thread Network and establish contact with a Router for commissioning. The joining device scans all channels, issues a beacon request on each channel, and waits for beacon responses. The beacon contains a payload including the network SSID (Service Set Identifier) and a permit joining beacon indicating if the Thread Network is accepting new members. Once a device has discovered the Thread Network, it uses MLE messages to establish a neighboring Router through which it can perform commissioning.

Discovery is not required if the device has already obtained commissioning information because it already has sufficient information to directly attach to the Thread Network.

Commissioning

Thread provides two commissioning methods:

- Configuring commissioning information directly onto a device using an out-of-band method. The commissioning information allows the joining device to attach to the proper Thread Network as soon as it is introduced to the network.
- Establishing a commissioning session between a joining device and a commissioning application on a smartphone, tablet, or the web. The commissioning session securely delivers commissioning information to the joining device, allowing it to attach to the proper Thread Network after having completed the commissioning session.

Note: The typical 802.15.4 method of joining solely via the permit joining flag in the beacon payload is not used in a Thread Network. 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 may have issues with device steering in situations where there are multiple networks available and also may have security issues.

Attaching

A joining device with commissioning information makes contact with a Parent Router and then attaches to the Thread Network via the Parent Router through exchanging MLE link configuration messages. Devices attach to a Thread Network as either an end device or a REED and are allocated a 16-bit short address by the Parent Router as illustrated in Figure 8.

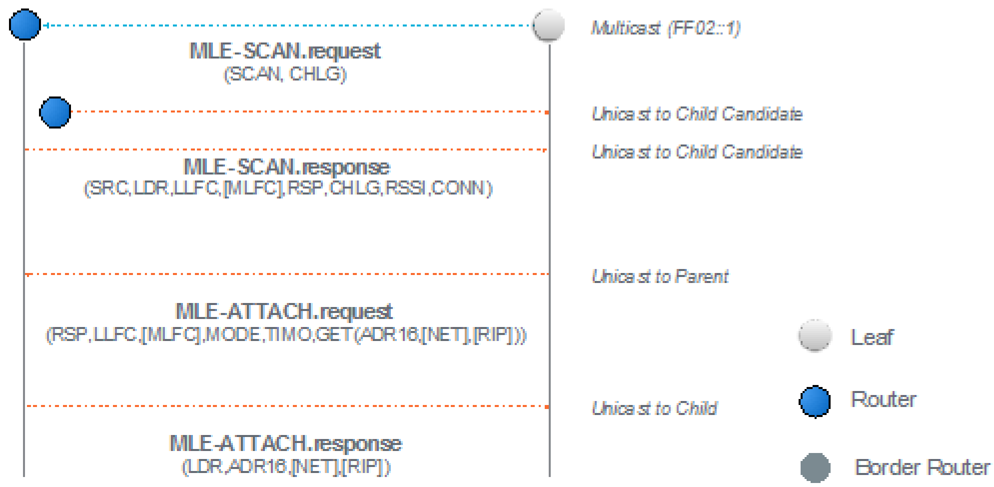

Figure 8. Attaching to a Thread Network with a Known Key

Once an REED has attached, it can potentially issue an address request to become a Router whereupon it is allocated a Router address by the Leader.

MLE Messages

Once a device has attached to a Thread Network, there is a variety of information required for it to maintain its participation in the network. MLE provides services to distribute network data throughout the network and exchange link costs and security frame counters between neighbors.

The MLE messages distribute 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 sleepy host device
- Neighbor link costs (if a Router)
- Security material and frame counters between devices
- Routing costs to all other Routers in the Thread Network
- Updates to network data such as the channel, PAN ID, and the beacon payload parameter used in the MAC

Note: MLE messages are encrypted except during discovery before the joining device has obtained the required security material.

DHCPv6

DHCPv6 [RFC 3315] is a UDP-based client-server protocol to manage configuration of devices within a network. DHCPv6 uses UDP to request data from a DHCP server.

The DHCPv6 service on a Border Router is used for configuration of:

- Network addresses
- Multicast addresses required by devices
- Hostname services

Management

ICMP

All devices support ICMPv6 error messages, as well as the echo request and echo reply messages.

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.

The information used by Thread from the 802.15.4 MAC layer includes:

- EUI 64 address
- 16 bit short address
- Capability information
- PAN ID
- Packets sent and received
- Packets dropped on transmit or receive
- Security errors
- Number of MAC retries

The information used by Thread from the network layer includes:

- IPv6 address lost
- Neighbor table
- Child table
- Routing table

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 a set of information needs to be stored in non-volatile storage. This includes:

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