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 Usage of 6LoWPAN

July 2015

Revision History

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

Contents

Introduction

Intelligent devices around the home make our lives easier and more enjoyable. Connecting these devices and enabling them to communicate and share information is essential for the smart home of tomorrow. Wireless communication technology provides an elegant solution for the communication links between these objects. Wireless networks in the smart home need to satisfy specific requirements such as the ability to provide reliable communication and at the same time consume very little energy. Thread uses as its RF (Radio Frequency) connectivity protocol the IEEE 802.15.4 communication standard [IEEE802154] which is specifically designed for low-rate, low-power WPANs (Wireless Personal Area Networks). To further satisfy the connectivity requirements of the smart home, Thread employs IPv6 connectivity that allows devices to communicate with one another, access services in the cloud, or interact with the user through Thread mobile applications. The need to unify IPv6 and 802.15.4 technologies was resolved by the development of a layer that provides smooth adaptation between the IPv6 networking layer requirements and 802.15.4 link layer capabilities. This layer is called 6LoWPAN and is illustrated in Figure 1.

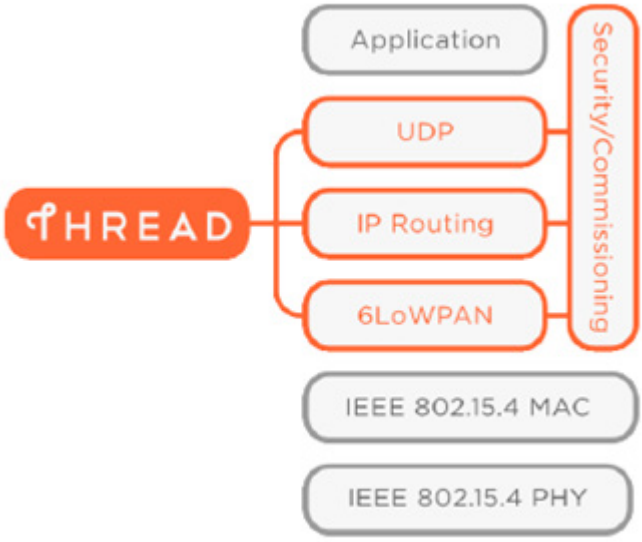

Figure 1. Thread Communication Stack

Using IPv6 as the network layer protocol requires that a minimum MTU (Maximum Transmission Unit) of 1280 bytes must be supported over the link [RFC 2460]. Considering the fact that [IEEE802154] defines a maximum PHY (Physical) packet size of 127 bytes, it becomes apparent

that an adaptation layer is needed. To satisfy these requirements, the 6LoWPAN adaptation layer employs several different techniques such as packet fragmentation and header compression (for the IPv6 header and transport headers such as UDP (User Datagram Protocol)) to ensure the successful delivery and reception of IPv6 packets coming to and from devices in the home.

IEEE 802.15.4

[IEEE802154] is a standard for wireless communication that defines the PHY and MAC (Media Access Control) layers and was issued by the IEEE (Institute for Electrical and Electronics Engineers). Designed with low power in mind, this wireless communication protocol is suitable for applications usually involving a large number of nodes, most of which can function on battery power for many years. 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 be as low as 88 bytes, depending on the security options and addressing type as illustrated in Figure 2.

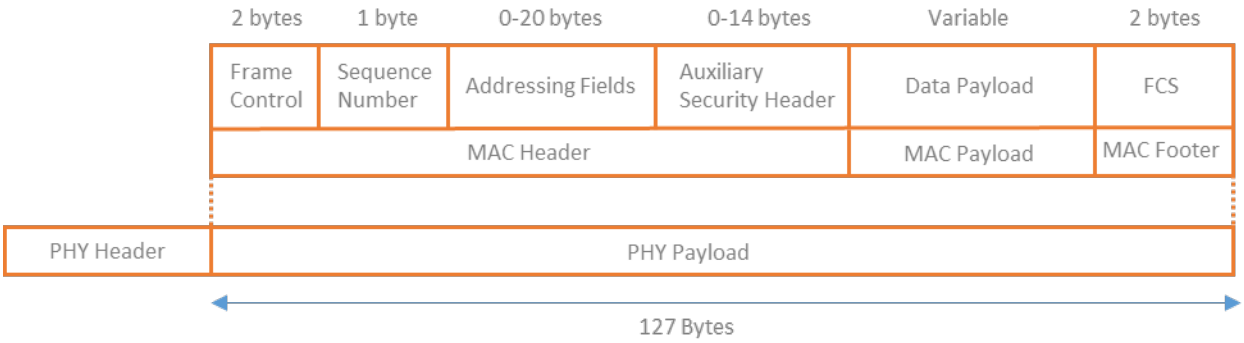

Figure 2. 802.15.4 PHY and MAC Payloads

IPv6 over 802.15.4

To efficiently send IPv6 packets over 802.15.4, one needs to concentrate on the issues that arise from the design of the underlying low power MAC and PHY protocols: the small payload support and the low reliability of the transmission of packets.

6LoWPAN stands for "IPv6 Over Low Power Wireless Personal Networks". It is designed specifically to handle the limitations when sending and receiving IPv6 packets over 802.15.4 links. In doing so, it has to accommodate for the 802.15.4 maximum frame size that can be sent

over-the-air. In Ethernet links, a packet with the size of the IPv6 MTU (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 headers sizes sent over-the-air and thus reduces transmission overhead. The fewer bits 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. The methods by which fragmentation and header compression is accomplished is described in detail by [RFC 4944] and [RFC 6282].

Another important feature of the 6LoWPAN layer is the ability to provide link layer packet forwarding. It 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. It makes use of the 6LoWPAN link layer forwarding capabilities to forward the packet by not having to send it up to the network layer. Thread makes full use of the ability of the MAC layer to provide addressing based on short addresses (16-bit length) to further reduce the information bits needed to be sent over-the-air to provide efficient packet forwarding. This saves processing cycles and improves power consumption at the same time while still using an IP based routing protocol.

Looking at the OSI (Open Systems Interconnection) model one can notice that the MAC is considered to be Layer 2 and the IPv6 Network is considered to be Layer 3. Being an adaptation layer, 6LoWPAN sits between these two and provides the necessary mechanisms and interfaces for them to interconnect.

To sum up, the 6LoWPAN adaptation layer provides the following:

- IPv6 packet encapsulation
- IPv6 packet fragmentation and reassembly
- IPv6 header compression
- Link layer packet forwarding

Each of these capabilities is described in more detail in the sections that follow.

6LoWPAN IPv6 Packet Encapsulation

To accomplish the functionalities described above, the 6LoWPAN layer takes the IPv6 packets, wraps them using encapsulation headers, and then subsequently sends them over-the-air using the 802.15.4 MAC and PHY layers.

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 as illustrated in Figure 3.

Figure 3. 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 [RFC 4944] mandates that if more than one header is present they must appear in the order mentioned above as illustrated in Figure 4.

Figure 4. 6LoWPAN Packet Containing IPv6 Payload with Compressed IPv6 Header

In the example above, the 6LoWPAN payload is composed of the compressed IPv6 header and the rest of the IPv6 payload as illustrated in Figure 5.

Figure 5. 6LoWPAN Packet Containing Mesh Hader for Layer 2 Forwarding, a Fragmentation Header and a Compression Header

In this example, the 6LoWPAN payload contains the IPv6 header and part of the IPv6 payload. The rest of the payload will be transmitted in subsequent packets that will have the format illustrated in Figure 6.

Figure 6. 6LoWPAN Packet Representing Subsequent Fragments that Do Not Contain Any Information about the IPv6 Header

6LoWPAN IPv6 Headers Compression

6LoWPAN uses header compression mechanisms to reduce the packet overhead when sending data over 802.15.4 links. It effectively reduces the size of the IPv6 header and transport headers sent over-the-air by relying on a set of assumptions (specific to 802.15.4 links) and information present in the link layer. Thread uses two types of compression both of which are described in [RFC 6282]:

- IPHC [Improved Header Compression]
- NHC [Next Header Compression]

The number of compressed bytes from the IPv6 and UDP transport headers varies depending on several factors such as what IPv6 addresses are used, the type of 802.15.4 addressing modes, and whether or not network contexts are available. The best case compression is illustrated in Figure 7 and Figure 8 (from 48 bytes to 6 bytes):

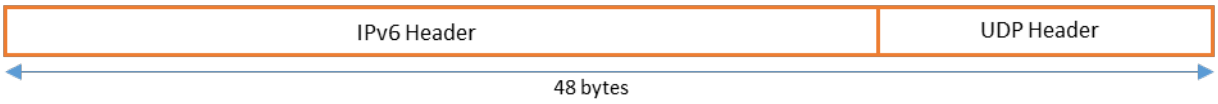

Figure 7. Full IPv6 and UDP Headers

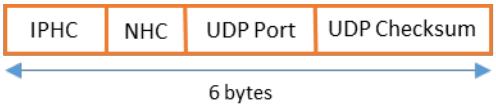

Figure 8. 6LoWPAN Compression of IPv6 Header and UDP Header

IPv6 Header Compression

IPHC is used for IPv6 header compression and brings an improvement on the HC1 header compression described in [RFC 4944]. It provides an efficient compression technique for global IPv6 addresses, enabling the reduction of IP information overhead between nodes residing more than one IP hop away. It makes use of shared contexts to fully elide the IP source and destination addresses in most cases. This alone can improve the compression efficiency substantially taking into consideration the fact that one IPv6 addresses is 16 bytes long.

This method of compression relies on the fact that context are distributed between nodes in the 802.15.4 network via a mechanism which is out of the scope of [RFC 4944].

IPHC compression assumes certain IPv6 header parameters will be common when communicating within a 6LoWPAN network as follows:

- The version field will have a value of 6.
- Traffic class and flow label are assumed to be 0.
- IP payload length can be calculated from the length specified in the lower layers (fragmentation header or IEEE 802.15.4 header).
- Hop limit will be set to a known value by the source.
- The prefixes of addresses will be known through the network by mean of context sharing.
- The IIDs of these addressed are constructed from either the 64-bit extended or 16-bit short IEEE 802.15.4 addresses.

The best compression of the IPv6 header (40 bytes) that can be achieved is 2 bytes in the case of link-local communication. When routing over multiple IP hops however, the compressed header can be no fewer than 7 bytes.

The structure of the IPHC header is illustrated in Figure 9 and explained in Table 1.

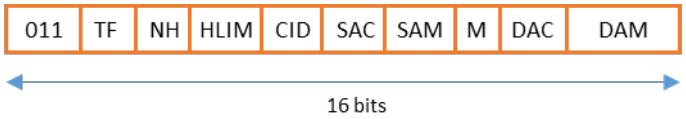

Figure 9. IPHC base encoding

Table 1. IPHC Header Field Definitions

Field	Definition
011	Represents the dispatch value for the IPHC header.
TF	Traffic Class and Flow Control. Specifies compression options for Traffic Class and Flow Label fields.
NH	Next Header. One bit specifying whether or not the next header is encoded using NHC.

Field	Definition
HLIM	Hop Limit. Two bits presenting information about how the hop limit is compressed.
CID	Context Identifier Extension. If this bit is 1, an 8 bit CIE (Context Identifier Extension) field follows after the DAM (Destination Address Mode) field.
SAC	Source Address Compression. One bit that specifies whether the compression is stateless of state full.
SAM	Source Address Mode. Two bits that are used with SAC to determine the source address compression type.
M	Multicast Compression. One bit that determines whether or not the destination address is a multicast address.
DAC	Destination Address Compression. One bit that specifies whether the compression is stateless or state full.
DAM	Destination Address Mode. Two bits that are used with M and DAC to determine the source address compression type.

6LoWPAN Contexts

In a Thread Network, IPv6 addressing is the basis of any node communication. An IPv6 header is present in most of the packets sent over-the-air.

There can be several types of IPv6 addresses that a node can assign such as link local, mesh local, and global addresses.

The 6LoWPAN layer reduces the size of the IPv6 header transmitted over-the-air by making use of network wide known information, such as the mesh local and global prefixes.

Mesh local and global prefixes used in a Thread Network are managed by the Leader which also manages network contexts by assigning a one-to-one mapping between prefixes and contexts as follows:

Mesh Local Prefix

FD00:0DB8::/64 -> Context Id 0 (the first non-link local prefix in the network has ID 0)

Global Prefixes

2001::/64 -> Context Id 1
2003::/64 -> Context Id 2

Contexts have a length and a context ID associated that must be synchronized in the entire Thread Network. The first step in using context compression is for the Leader to distribute the context to all the nodes in the Thread Network by means of propagating the Thread Network Data. More information about this procedure can be found in the "Thread Border Routers" white paper. Once propagation is done, a device can start sending packets with compressed IPv6 headers and set the CID field to '1' in the IPHC compression header. This signals that a context identifier extension will be present immediately following the IPHC DAM field. The context identifier extension has the structure illustrated in Figure 10 and explained in Table 2.

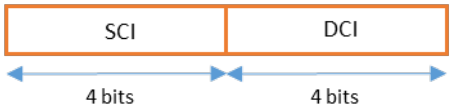

Figure 10. Context Identifier Extension

Table 2. Context Identifier Extension Field Definitions

Field	Definition
SCI	Source Context Identifier. Identifies the prefix associated with the IPv6 source address.
DCI	Destination Context Identifier. Identifies the prefix associated with the IPv6 destination address.

UDP Header Compression

UDP header compression in Thread is achieved by using the NHC [Next Header Compression] [RFC 4944]. The compression of the next header after the IPv6 header is specified using the NHC header. An example of a typical compression headers configuration is illustrated in Figure 11.

Figure 11. Compression Headers

The NHC header for UDP compression has the format illustrated in Figure 12 and explained in Table 3.

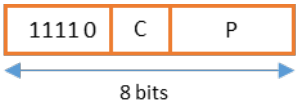

Figure 12. UDP Header Encoding

Table 3. UDP Header Encoding Field Definitions

Field	Definition
1111 0	NHC ID for UDP header compression
C	Checksum. One bit specifying whether the UDP checksum is carried inline or elided.
P	Ports. Two bits specifying the compression scheme for the UDP ports.

6LoWPAN IPv6 Packet Fragmentation

As mentioned above, the IEEE 802.15.4 MAC layer provides a worst-case scenario payload length of 88 bytes. Taking into consideration the fact that the IPv6 MTU that needs to be supported is 1280 bytes, a fragmentation and reassembly mechanism is needed. Also from the viewpoint of a practical application, the remaining payload size available for sending data that fits into a single 802.15.4 packet can be very short. If, for example, one uses UDP as a transport layer, then the effective payload for an application layer would be calculated as:

MAC Payload (88 bytes) – IPv6 Header (40 bytes) – UDP Header (8) = 40 Bytes

Using the header compression technique described in [RFC 6282] will increase the number of bytes available for the application payload but that will still only be 80 bytes as illustrated in Figure 13.

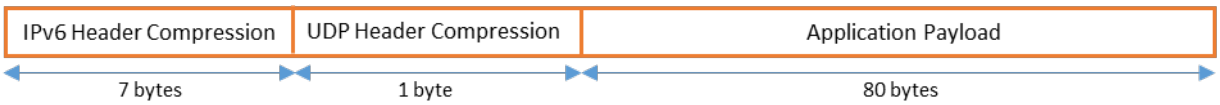

Figure 13. Application Payload in the Case of Multi-hop Transmission with IPv6 Header Compression

When an IPv6 packet does not fit into a single IEEE 802.15.4 frame, the 6LoWPAN layer creates fragments that encapsulate the original IPv6 packet and sends them one by one over-the-air. The 6LoWPAN packets are then received at the other end and the IPv6 packet is reassembled and delivered to the IP layer as illustrated in Figure 14.

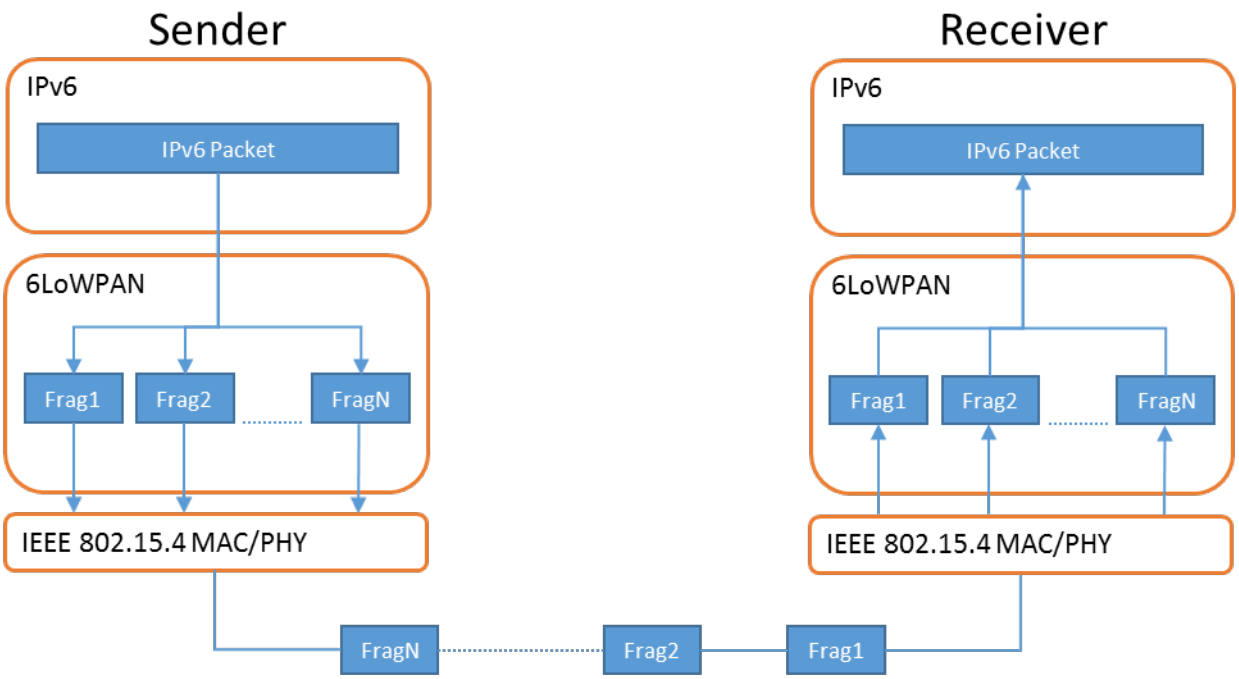

Figure 14. Fragmenting and Reassembling an IPv6 Packet

It is important to consider the fact that all fragments must arrive at the destination for the IPv6 packet to be reassembled. To avoid the risk of running out of memory due to possible fragment

loss, the receiving device keeps a timer that is activated when it receives the first 6LoWPAN fragment corresponding to an IPv6 packet. If the timer expires, the device frees all the allocated memory that it used to store the received fragments for that particular packet.

A More Detailed Look into Fragmentation

There are two types of fragments that are used by the 6LoWPAN layer—FRAG1 and FRAGN.

FRAG1 contains the IPv6 compressed header and can contain a part of the payload. FRAGN packets are the fragments that are send subsequently which must contain the rest of the IPv6 payload. The format of FRAG1 and FRAGN type headers are illustrated in Figure 15 and Figure 16.

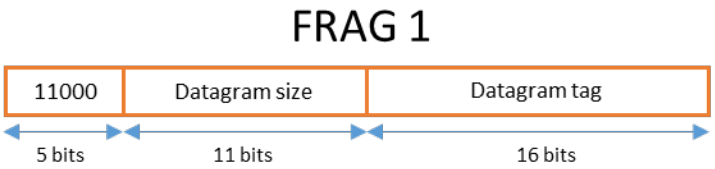

Figure 15. First Fragment Header

Figure 16. Subsequent Fragment Header

The fields present in the fragmentation headers are used to reconstruct the original IPv6 packet at the receiving end. Table 4 summarizes the field definitions.

Table 4. Fragmentation Header Field Definitions

Field name	Definition
Datagram size	The size of the original IPv6 packet.
Datagram tag	A unique identifier of the IPv6 fragmented packet. Usually it is used in conjunction with the source address of the device to establish network wide uniqueness of the packet. It links all the fragments together.
Datagram offset	This field is present only in the subsequent fragments and contains the offset of the fragment relative to the original packet in multiples of 8 bytes. This way the receiver know where to copy the information in the reconstruction of the original IPv6 packet.

Another important characteristic of fragmentation is that fragments may not necessarily arrive in the order they were sent. 6LoWPAN can reliably reassemble the original packet regardless of the order in which the fragments arrive. The only necessary condition is that all fragments are received.

6LoWPAN Mesh Forwarding

Thread uses the mesh header mechanism defined in the 6LoWPAN specification to forward packets between devices in the Thread Network. The 6LoWPAN mesh header is used in each multi-hop packet as illustrated in Figure 17.

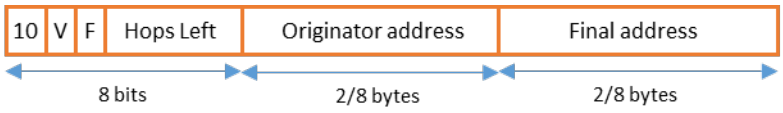

Figure 17. Mesh Header Format

The 6LoWPAN layer in Thread fills the Mesh Header information with the originator 16-bit short address and final destination 16-bit source address, 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.

References

Document	Title
[IEEE802154]	Wireless Personal Area Networks
[RFC 4944]	Transmission of IPv6 Packets over IEEE 802.15.4 Networks
[RFC 6282]	Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks
N/A	“Thread Border Routers” white paper