



Chapter 7B: Bluetooth Mesh Protocol

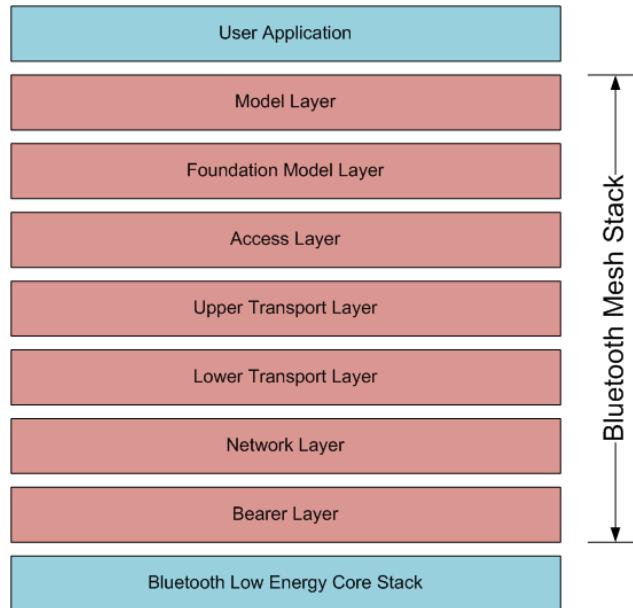
Time 1 Hour

This chapter covers the details of the Bluetooth Mesh Stack and Packets.

7B.1 MESH STACK ARCHITECTURE	2
7B.1.1 BEARER LAYER.....	2
7B.1.2 NETWORK LAYER	2
7B.1.3 LOWER TRANSPORT LAYER	3
7B.1.4 UPPER TRANSPORT LAYER.....	3
7B.1.5 ACCESS LAYER	3
7B.1.6 FOUNDATION MODEL LAYER.....	3
7B.1.7 MODEL LAYER	3
7B.2 (ADVANCED) PACKET DETAILS.....	4
7B.2.1 ACCESS MESSAGES	4
7B.2.2 CONTROL MESSAGES	6
7B.2.3 PACKET SEGMENTATION AND REASSEMBLY	8
7B.3 EXERCISES.....	9
EXERCISE 7B.1 MESH PROFILE SPEC.....	9
EXERCISE 7B.2 MESH MODEL SPEC	9

7B.1 MESH Stack Architecture

Like all complex systems, Bluetooth mesh uses a layered approach. In this case, the stack is broken into 7 layers as shown below. The lowest layer sits on top of the BLE stack.



7B.1.1 Bearer Layer

The lowest level defines how mesh messages get to/from the BLE stack. Presently, there are two bearer layers – the Advertising Bearer and the GATT Bearer. Most devices in a mesh network will use the Advertising Bearer to send messages using BLE advertising packets.

The GATT Bearer allows devices that don't support the Advertising Bearer (such as most smartphones) to communicate indirectly with the network via GATT Proxy nodes using the Proxy Protocol.

7B.1.2 Network Layer

The network layer defines message address types and network message format. It can support multiple bearers, each of which may have multiple network interfaces including the local interface which is used for communication between elements in the same node. Then network layer determines which network interface(s) to output messages over.

An input filter is used to determine if messages from the bearer layer should be delivered to the network layer for processing or not. An output filter is used to determine if messages should be delivered to the bearer layer or should be dropped.

The Relay and Proxy features are implemented in the network layer. Therefore, any messages not intended for this node are relayed to the appropriate network interface(s) but are not sent to the next layer in the stack.



7B.1.3 Lower Transport Layer

The lower transport layer is responsible for segmenting outgoing messages so that they will fit in the required transport PDU (Protocol Data Unit) and reassembling segmented incoming messages.

7B.1.4 Upper Transport Layer

The upper transport layer is responsible for encryption, decryption, and authentication of application data being passed to/from the access layer. It is also responsible for generating and dealing with transport control messages such as friendship and heartbeat messages.

7B.1.5 Access Layer

The access layer defines how higher-level applications can make use of the upper transport layer. It includes defining the format of the application data, defining and controlling the encryption and decryption process performed in the upper transport layer, and verifying data from the upper transport layer is intended for the right network and application before sending it further up the stack.

7B.1.6 Foundation Model Layer

The foundation model layer implements models required to configure and manage a mesh network. Specifically, the two required models: device configuration and device health.

7B.1.7 Model Layer

The model layer implements the behaviors, messages, states, etc. as defined in the mesh model specification. The user application interacts with devices by using the models. That is, this layer is the main interface between your application and the rest of the mesh stack.



7B.2 (Advanced) Packet Details

Bluetooth mesh networks are accomplished using Advertising packets. The Advertising packet is 31 octets, but much of that space is used up by network overhead. That overhead includes information from the network layer, information from the lower transport layer, and information from the upper transport layer.

As was discussed in the prior chapter, the number of octets available for a message payload depends on whether it is a control or access messages and on whether the message is segmented or unsegmented.

The packets include message integrity checks for the network layer (for all messages) called the NetMIC and at the upper level transport layer (only for access messages) called the TransMIC. These are used to verify that a message was not modified (either intentionally or unintentionally) during transmission. The two fields for integrity checks are as follows:

Name	Message Type	Size (Octets)
NetMIC	Control	8
NetMIC	Access	4
TransMIC	Unsegmented Access	4
TransMIC	Segmented Access	4 or 8

7B.2.1 Access Messages

Unsegmented Access Messages

Unsegmented access messages can contain up to 11 octets of payload. The full 31 octets in the advertising packet (assuming a max size packet) are allocated as shown in the table below including the level in the stack that is responsible for each item.

Octet	# of Octets	Field Name	Level	Notes
0	1	Length	BLE ADV Packet	Advertising Packet Length
1	1	Type	BLE ADV Packet	Advertising Packet Type = Mesh
2	1	IVI NID	Network	1 bit: LSB of IV Index 7 bits: value derived from NetKey to identify the encryption and privacy keys used for the packet
3	1	CTL TTL	Network	1 bit: indicate an access message (0) 7 bits: Time to Live (TTL)
4 – 6	3	SEQ	Network	Sequence Number
7 – 8	2	SRC	Network	Source Address
9 - 10	2	DST	Network	Destination Address
11	1	SEG AKF AID	Lower Level Transport	1 bit: unsegmented message (0) 1 bit: application key flag 6 bits: application key identifier
12 – 22	11	Payload	Payload	Message Payload including 1, 2, or 3 bytes of Opcode
23 - 26	4	TransMic	Upper Level Transport	
27 - 30	4	NetMIC	Network	Message Integrity Check for Network layer (4 octets for Access messages)



Segmented Access Messages

Segmented access messages can contain 12 octets of payload for each packet except the last one which can contain up to either 4 or 8. The maximum payload for the last packet depends on the size of the Transport Message Integrity Check (TransMIC) which can be either 4 or 8 octets depending on the message. This allows a maximum of 380 or 376 octets of payload in a single access message depending on the last packet. One, two, or three of those octets is an opcode for the message.

Number of Packets	Max Payload Size (Octets)	
	4 octet TransMIC	8 octet TransMIC
1 (unsegmented)	11	n/a
1 (segmented)	8	4
2	20	16
n	(nx12)-4	(nx12)-8
32	380	376

The full 31 octets in each advertising packet (assuming all max size packets) are allocated as shown in the table below including the level in the stack that is responsible for each item.

Octet	# of Octets	Field Name	Level	Notes
0	1	Length	BLE ADV Packet	Advertising Packet Length
1	1	Type	BLE ADV Packet	Advertising Packet Type = Mesh
2	1	IVI NID	Network	1 bit: LSB of IV Index 7 bits: Value derived from NetKey to identify the encryption and privacy keys used for the packet
3	1	CTL TTL	Network	1 bit: Indicates an access message (0) 7 bits: Time to Live (TTL)
4 – 6	3	SEQ	Network	Sequence Number
7 – 8	2	SRC	Network	Source Address
9 - 10	2	DST	Network	Destination Address
11	1	SEG AKF AID	Lower Level Transport	1 bit: segmented message (1) 1 bit: application key flag 6 bits: application key identifier
12 – 14	3	SZMIC SeqZero SegO SegN	Lower Level Transport	1 bit: Size of TransMIC 13 bits: Least significant bits of SeqAuth 5 bits: Segment offset number 5 bits: Last segment number
15 – 18 15 – 22 15 – 26	4 8 12	Payload	Payload	Message Payload including 1, 2, or 3 bytes of Opcode. The last packet is a max of either 4 or 8 octets depending on the size of TransMic. The other packets are 12 octets each since TransMic is sent only at the end of the last packet.
19 – 26 23 – 26	8 4	TransMIC	Upper Level Transport	Either 4 or 8 octets depending on value of SZMIC. This is only included in the last packet.
27 – 30	8	NetMIC	Network	Message Integrity Check for Network layer



7B.2.2 Control Messages

Unsegmented Control Messages

Like unsegmented access messages, unsegmented control messages can contain up to 11 octets of payload, but the overall contents of the packet are slightly different. The full 31 octets in the advertising packet (assuming a max size packet) are allocated as shown in the table below including the level in the stack that is responsible for each item.

Octet	# of Octets	Field Name	Level	Notes
0	1	Length	BLE ADV Packet	Advertising Packet Length
1	1	Type	BLE ADV Packet	Advertising Packet Type = Mesh
2	1	IVI NID	Network	1 bit: LSB of IV Index 7 bits: Value derived from NetKey to identify the encryption and privacy keys used for the packet
3	1	CTL TTL	Network	1 bit: Indicates a control message (1) 7 bits: Time to Live (TTL)
4 – 6	3	SEQ	Network	Sequence Number
7 – 8	2	SRC	Network	Source Address
9 – 10	2	DST	Network	Destination Address
11	1	SEG Opcode	Lower Level Transport	1 bit: Unsegmented message (0) 7 bits: Opcode
12 – 22	11	Payload	Payload	Message Payload
23 – 30	8	NetMIC	Network	Message Integrity Check for Network layer



Segmented Control Messages

Segmented control messages can contain up to 8 octets of payload each. Therefore, a control message can be at most 256 octets long.

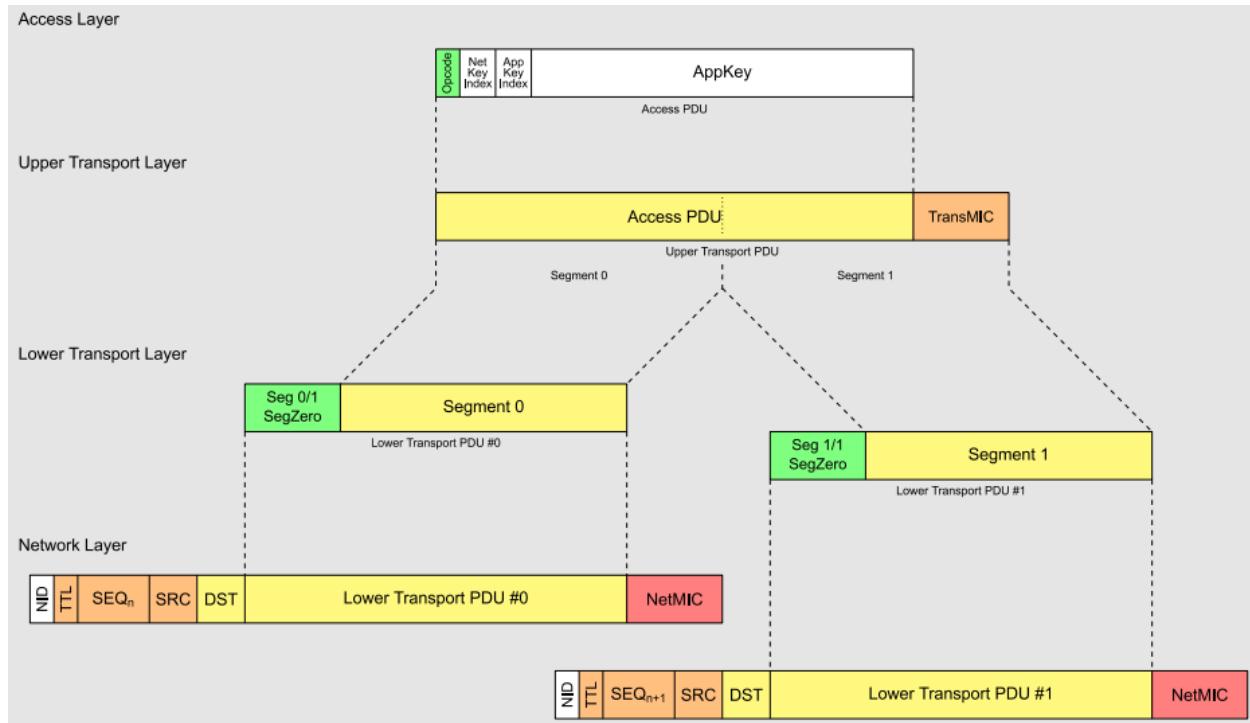
Number of Packets	Max Payload Size (Octets)
1 (unsegmented)	11
1 (segmented)	8
2	16
n	n*8
32	256

The full 31 octets in each of the advertising packets (assuming all max size packets) are allocated as shown in the table below including the level in the stack that is responsible for each item.

Octet	# of Octets	Field Name	Level	Notes
0	1	Length	BLE ADV Packet	Advertising Packet Length
1	1	Type	BLE ADV Packet	Advertising Packet Type = Mesh
2	1	IVI NID	Network	1 bit: LSB of IV Index 7 bits: Value derived from NetKey to identify the encryption and privacy keys used for the packet
3	1	CTL TTL	Network	1 bit: Indicates a control message (1) 7 bits: Time to Live (TTL)
4 – 6	3	SEQ	Network	Sequence Number
7 – 8	2	SRC	Network	Source Address
9 – 10	2	DST	Network	Destination Address
11	1	SEG Opcode	Lower Level Transport	1 bit: Segmented message (1) 7 bits: Opcode
12 – 14	3	RFU SeqZero SegO SegN	Lower Level Transport	1 bit: Unused 13 bits: Least significant bits of SeqAuth 5 bits: Segment offset number 5 bits: Last segment number
15 – 22	8	Payload	Payload	Message Payload
23 – 30	8	NetMIC	Network	Message Integrity Check for Network layer

7B.2.3 Packet Segmentation and Reassembly

As mentioned previously, the segmentation and reassembly of packets are handled by the lower transport layer. An example of how segmentation is done can be seen in the figure below.



(This figure is taken from the Bluetooth Mesh Profile Specification)



7B.3 Exercises

Exercise 7B.1 Mesh Profile Spec

Download and review the Mech Profile Spec. Answer the following questions:

1. What is the endianness for the network layer, lower transport layer, upper transport layer, and access layer?
 2. What is a mesh gateway?
 3. For a Mesh Proxy Service, what characteristics are required and what properties must they have?

Exercise 7B.2 Mesh Model Spec

Download and review the Mesh Model Spec. Answer the following questions:

1. What is the difference between the states "Light Lightness Actual" and "Light Lightness Linear"? How are they related?



2. How many lighting server models are defined? Where can you find them described in the Mesh model spec?
 3. List one set of messages that can be used for setting the Hue, Saturation (i.e. color) and Lightness of a light. Is this the only way to do it?