Weave Message Layer

Protocol Specification

Revision 1.1.1
2017-09-03

Abstract

This document is a technical specification of the format and operational semantics of the messages exchanged in the *Weave Message Layer*.

Introduction

This document is an informal specification of the Weave Message Layer, used in the Weave System as a common lightweight transport and session layer suitable for both machine-to-machine and machine-to-service communication.

Weave Message consists of A) an array of octets beginning with a common header structure of fixed size comprising a series of transport- and session-layer values each encoded in portable machine types, followed by B) a variable length array of octets, possibly encrypted, comprising the application payload data. In one typical usage scenario, messages are expected to be transmitted over User Datagram Protocol (UDP) on lossy links offering less than “best-effort” message delivery. The Weave Message Layer protocol provides additional mechanisms for reliable delivery, message integrity, confidentiality, and authenticity for cases where the underlying transport does not provide them.

Special Requirements Language

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are meant to be interpreted in accordance with “Key words for use in RFCs to Indicate Requirement Levels” [RFC2119].

Table Of Contents

Overview

This section introduces the Weave Message Layer at a conceptual level as a message transport and session layer to facilitate machine-to-machine and machine-to-service communication in the Weave System.

There are three different sorts of underlying network layer for the Weave Message Layer:

- a simple point-to-point packet exchange link, e.g., a framed full-duplex serial data connection,
- a multicast-capable shared access link, e.g., Ethernet, Wi-Fi™,
- a low-power, lossy network (LLN) that uses 6LoWPAN compression and mesh routing, e.g., Thread™.

Message Format

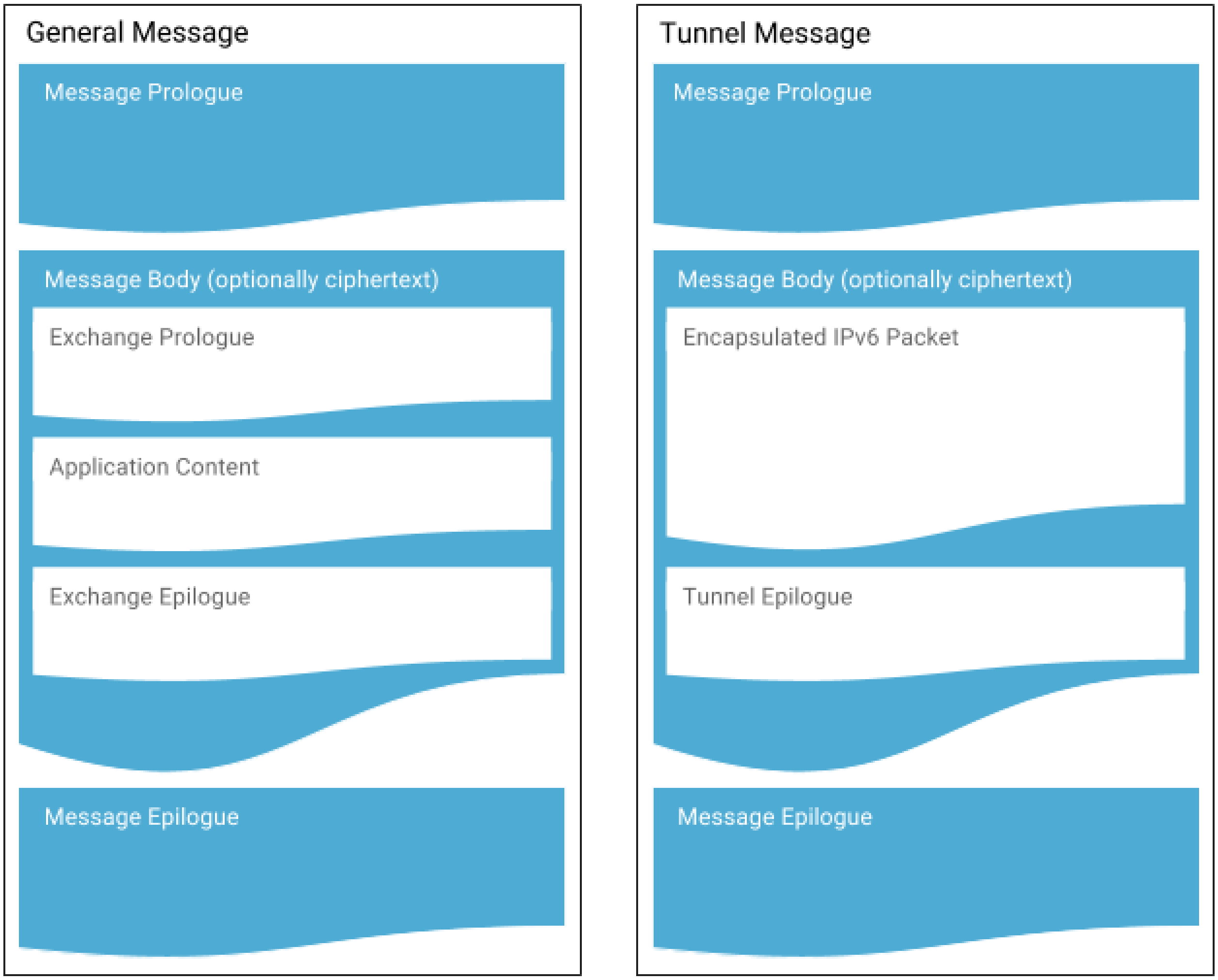
There are two forms of message in the Weave Message Layer, as follows:

- **General** — a message encapsulating general Weave content.
- **Tunnel** — a message encapsulating an IP packet.

Both forms of message consist of a Message Prologue, followed by an optionally encrypted Message Body, and optionally finished with Message Epilogue which may be either plaintext, ciphertext, or a mix of both, as the encryption strategy entails.

The content of the Message Prologue specifies the form and content of the Message Epilogue and whether the plaintext of the Message Body is general content or an encapsulated IP packet.

The following diagram gives a simplified overview of the two forms:



The various prologue and epilogue parts comprise fixed structures of fields followed by optional additional fields whose presence is signaled by other content of the message.

The Weave Common Profile is provided to define the format of the Application Body for certain message types used in the Weave Message Layer itself. These messages are typically encoded in the Weave TLV [WTLV] structured data interchange language.

Further description of these components is presented in Technical Detail below.

Framing Mechanisms

The Weave Message Layer relies, where available, on the underlying transport for message framing. For example, when transported by User Datagram Protocol (UDP) [RFC768], each datagram carries a single Weave Message of length that appears in the UDP header. For an alternative example, an unbounded series of Weave Messages is transported over the octet stream of a single Transmission Control Protocol (TCP) [RFC793] connection by prepending each message with a 16-bit Message Length field (see the Message Stream Structure subsection below).

Ordering Of Bits And Octets

To accommodate very constrained computing environments with “little-endian” machine representations of integer types, all the multi-octet integer fields in the Weave Message Layer format are encoded with least-significant octet first. Accordingly, this specification uses octet-aligned (as opposed to word-aligned) layout diagrams in the style of Guidelines for Internet Standards Writers [RFC2360], in which the bit order within octets is the so-called MSB 0 order, i.e., bits of integer fields are presented left to right, from most to least significant bit.

Node Identification

The Weave Message Layer identifies each node with a value in EUI-64 format [EUI] called its Node Identifier. The Weave Message Prologue includes fields for both Source and Destination Node Identifiers when messages are sent over links where they cannot be inferred from the network addresses in use.

Two distinguished Node Identifier values are reserved for special purposes: the Unspecified Node identifier 00:00:00:00:00:00:00:00, and the Any Node identifier ff:ff:ff:ff:ff:ff:ff:ff. The first is is never transmitted, i.e., it is only used in message processing logic as a placeholder when no valid node identifier is available. The second is used as a destination for messages sent to any receiving correspondent.

Fabric Identification

The Weave Message Layer introduces the concept of a Fabric Id to indicate the community of devices and service endpoints authorized to communicate freely with one another at the transport level. In the typical case, a fabric consists of all the authorized devices on a home area network, including devices that provide border routing functions over virtual network tunnels to the service, together with all the various service endpoints in the Weave Service, and also any mobile devices reachable via tunnels connected to the service.

The commissioning protocol, used to securely join a device with an existing fabric, is an application profile of the Weave Message Layer. Its technical specification is provided in *Weave Fabric Provisioning Protocol* [COMMISSION].

Each fabric is identified by its unique 64-bit Fabric Id that MUST be generated by a method that satisfies “Randomness Requirements for Security” [RFC4086] to facilitate its mapping to a prefix for use with Unique Local IPv6 Unicast Addresses [RFC4193]. For detail, see the format of the Message Prologue below.

Message Layer Forwarding

There are three types of nodes in a Weave Message Layer fabric that communicate freely with one another at the level of IPv6 networking:

- Devices: On-site sensors and actuators.
- Service Endpoints: Logically addressable server entities in the Weave Service.
- Remotes: Mobile handsets with human interface applications.

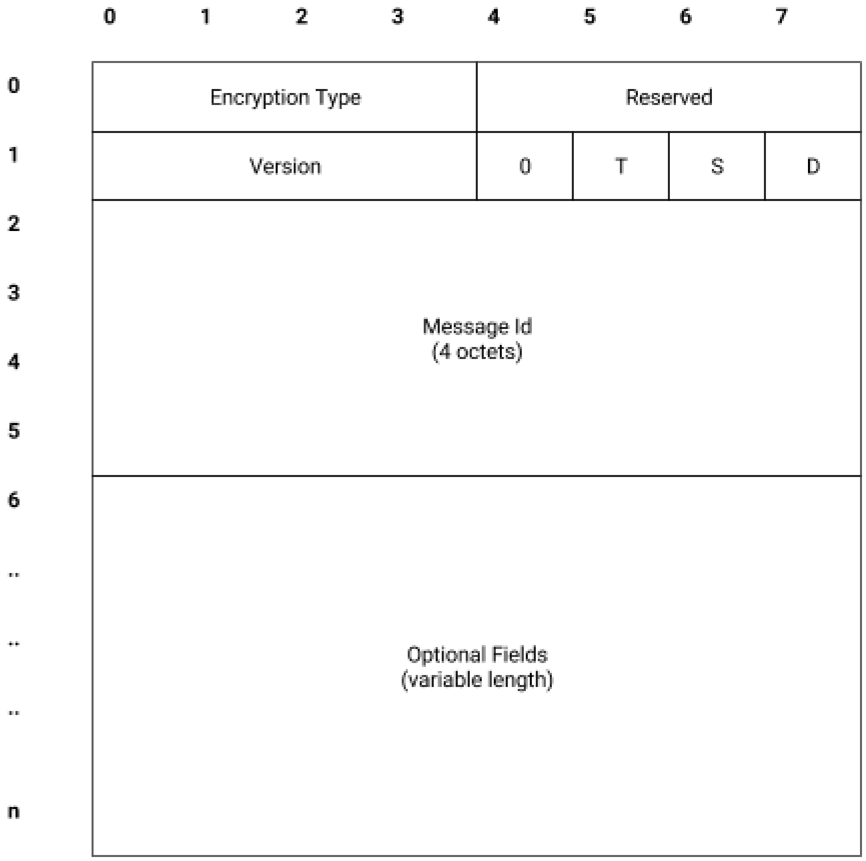
A rudimentary IPv6 router in the service forwards messages between all three types of nodes. Each fabric is logically divided into IPv6 subnetworks according to the node type and home area network location. Further details are provided in the Fabric Topology and Message Routing section below.

Technical Detail

This section defines the structure of Weave messages and serialized Weave message streams.

Message Prologue

The octets at the start of every Weave message MUST consist of the Message Prologue, which comprises a series of required and optional fields, shown in the following diagram:



The following table describes each required field in the Message Prologue:

Field Name	**Description**
Encryption Type	This integer field identifies the encryption algorithm in use for the Message Body. (See Encryption Of Message Body for details.)
Reserved, 0	This field is reserved for definition in future versions. In versions 1 and 2, Weave nodes MUST NOT transmit messages with a non-zero value in this field.
Version	This integer field identifies compatibility barriers between legacy versions of the Weave message format. The current version is 2. In the previous version 1, there is no support for cryptographic signatures or tunnel messages. All other values of the Version field are reserved.
T	If T=0 then the Message Prologue is followed by the general form of the Message Body. Otherwise if T=1 then it is followed by the tunnel form, i.e., an encapsulated IP packet. (See below.)
S	If S=0 then the Source Node Id field is not present in the

	Message Prologue. Otherwise if S=1, then the Source Node Id field is present. (See below.)
D	If D=0 then the Destination Node Id field is not present in the Message Prologue. Otherwise if D=1, then the Destination Node Id field is present. (See below.)
Message Id	This field is little-endian unsigned integer used in duplicate detection for identifying the transmission of a message and in some encryption modes as a nonce. Depending on other factors, values of this type are required to have additional properties, e.g., serial number arithmetic. (See below.)

In some historical documents and reference implementations, the reliable delivery mechanism is called the Weave Reliable Message Protocol (WRMP). When the reliable delivery mechanism is in use, the values appearing in the Message Id field in successively transmitted messages in an exchange MUST exhibit the ordering property described in Serial Number Arithmetic [RFC1982]. Further details about the semantics of the Message Id values is described in the Reliable Delivery and Encryption Of Message Body sections below.

Immediately following the required fields of the Message Prologue and immediately preceding the Message Body, zero or more octets of additional fields are formatted according to the values of the S, D, and Encryption Type fields in the Message Prologue.

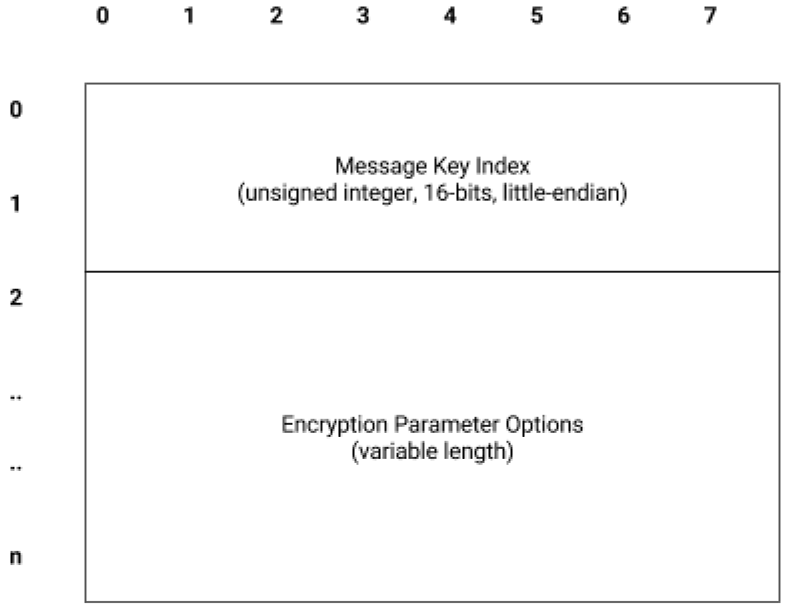
The first additional field, the Source Node Id, is present if S=1. It consists of an EUI-64 identifier for the Weave node that originated the message. The Source Node Id MUST NOT be the Anycast Node Id.

The second additional field, the Destination Node Id, is present if D=1. It consists of an EUI-64 identifier for the Weave node addressed to be the recipient.

The Unspecified Node Id MUST NOT be sent in the Source Node Id or the Destination Node Id fields of the Message Prologue.

To facilitate the transport of Weave messages over 6LoWPAN compression [RFC4944], source and destination Unique Local IPv6 Addresses [RFC4193] are composed by using the Node Id as the Interface Identifier (IID) and the least significant 40 bits of the Fabric Id as the ULA Global Id field. The Source and Destination Node Id fields MAY be elided in the Message Prologue when they can be inferred from the source and/or destination addresses in an encapsulating IPv6 header because they are ULAs with network prefixes that match the Fabric Id. This compression method MAY be used even when the enclosing IPv6 packet is not compressed with 6LoWPAN.

The third additional field, the Encryption Parameters Structure, is present if Encryption Type is non-zero. This structure contains the fields shown in the following diagram:


0
1
2
3
4
5
6
7
0
1
2
..
..
n
Message Key Index
(unsigned integer, 16-bits, little-endian)
Encryption Parameter Options
(variable length)

The Message Key Index field in the Encryption Parameters Structure is a 16-bit little-endian unsigned integer. Nodes capable of encrypting and decrypting Weave messages incorporate a Weave Key Service that can produce the necessary key material when provided with the Encryption Type, the Source Node Id, the Message Key Index, and potentially other parameters, depending on the Encryption Type.

Encryption and decryption of the body of a Weave message entails acquiring the necessary key material from the node's Weave Key Service and using it with any additional parameters, which appear as necessary in the Encryption Parameter Options.

No further content in the Message Prologue is currently defined.

Encryption Parameter Options

First in the Encryption Parameter Options fields, when the encryption method requires plaintext to be extended with padding octets—for example with block mode ciphers—the Plaintext Length field is present. The field is a 16-bit little-endian unsigned integer that indicates the number of plaintext octets present before the padding octets when decryption is finished.

Second, when the encryption method requires initializing octets, the Initialization Vector field is present. This is a variable length array of octets, sized according to the encryption method. No methods are currently defined that require an Initialization Vector field.

No further content is currently defined in the Encryption Parameter Options.

Message Body

The octets immediately following the Message Prologue, if present, MUST consist of the Message Body, encrypted or not according to the method specified by the Encryption Type in the Message Prologue.

If T=0, then the plaintext of the body is general Weave content (see below). Otherwise, if T=1 in the Message Prologue, then the plaintext of the body is Weave Tunnel content (see further below).

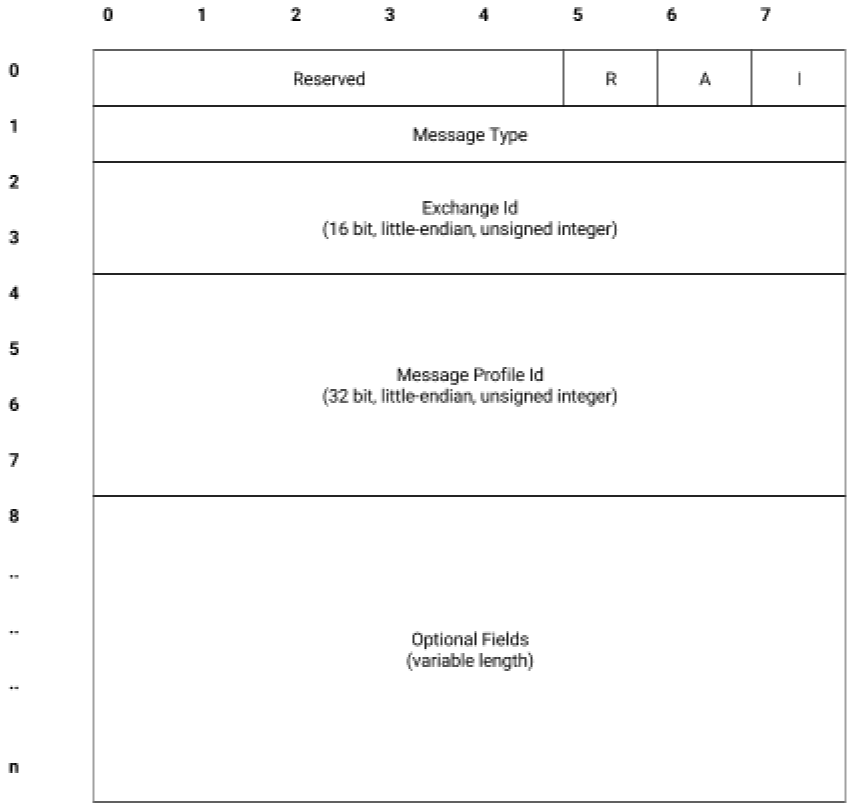
The end of the Message Body marks the end of a Weave Message. There is no epilogue after the body currently defined.

General Content Body

When T=0 in the Message Prologue, the plaintext of the Message Body is a variable length array of octets comprising the required and optional fields of an Exchange Prologue, followed by an array of zero or more octets formatted according to application requirements, and finally zero or more octets comprising the optional Exchange Epilogue fields.

Exchange Prologue

The format of the Exchange Prologue is shown in the following layout diagram:



	0	1	2	3	4	5	6	7
0	Reserved					R	A	I
1	Message Type							
2	Exchange Id (16 bit, little-endian, unsigned integer)							
3								
4	Message Profile Id (32 bit, little-endian, unsigned integer)							
5								
6								
7								
8	Optional Fields (variable length)							
..								
..								
..								
n								

The following table describes each required field in the Exchange Prologue:

Field Name	**Description**
R	If R=0, then the receiver is not required to perform reliable delivery processing (see Reliable Delivery below). Otherwise, if R=1, then reliable delivery processing is necessary.
A	If A=0, then no Acknowledgement Message Id field is present in the Exchange Prologue (see below). Otherwise, if A=1, then the Acknowledgement Message Id field is present.
I	If I=1, then the sender is operating in the Initiator role for the exchange. Otherwise, if I=0, then the sender is operating in a Responder role for the exchange.
Reserved	This field is reserved for future use. In versions 1 and 2, its value in transmitted packets MUST be 00010 and its value SHOULD NOT be recognized by receivers.
Message Type	An unsigned integer code identifying the specific type of content according to the application specified by the Message Profile Id field.
Exchange Id	An unsigned little-endian integer code used for identifying the context of an exchange of messages.
Message Profile Id	An unsigned little-endian integer code identifying the registered application that defines how the Message Type field is used. Nest Labs, Inc. is the authority maintaining the top level of the federated registry of profile identifiers [PROFILES].

With versions 1 and 2 of the Weave Message Layer, as indicated by the Version field in the Message Prologue, Weave nodes MUST NOT transmit messages with a value in the Reserved field of the Exchange Prologue other than the reserved value 00010, and they MUST NOT process messages differently depending on the content of the Reserved field.

The context of an exchange of Weave messages is identified by the Exchange Id field in the Exchange Prologue. The first node to send a Weave message in an exchange context is said to be in the Initiator role, and all the other nodes that subsequently participate in the exchange are said to be in a Responder role.

The node in the Initiator role MUST always set I=1 in the Exchange Prologue of every message it sends in that exchange. The first message a node in the Initiator role sends in a new exchange MUST contain a fresh value for the Exchange Id field. Each of the one or more other nodes participating in the exchange MUST use the same Exchange Id value for each of the zero or more messages that it sends in that exchange.

Each node in a Responder role for an exchange MUST use the Exchange Id code received in a previous message for the exchange. Each node in the Responder role MUST set I=0 in the Exchange Prologue of every message it sends in that exchange. Each node in a Responder role MUST NOT use a Destination Node Id field of the Message Options Prologue that identifies any node other than the node in the Initiator role for the exchange.

Immediately following the required fields of the Exchange Prologue and immediately preceding the application content, zero or more octets of additional fields of the Exchange Prologue appear, formatted according to the values of the A field in the Exchange Prologue.

First among the additional Exchange Prologue fields, if A=1 in the Exchange Prologue, a 32-bit unsigned little-endian integer Acknowledged Message Id field is present. Otherwise, if A=0, then the Acknowledged Message Id field is not present. (See Reliable Delivery for the operational semantics of this field.)

Zero is a distinguished value of the Message Profile Id, referring to the Weave Common Profile, which contains the following four message types:

Message Type	**Name**	**Description**
1	Status Report	A numeric status code.
2	Null	Empty message.
3	WRMP Delayed Delivery	Notification from a forwarding node to the sender of a message that delivery to its sleeping destination is delayed for a long time.
4	WRMP Throttle Flow	Notification from a receiving node to the sender to delay further transmissions for some time.

These values of the Message Type field are described in further detail below, in the Operational Semantics section.

Application Content

Zero or more octets of plaintext comprising the application content follows immediately after the Exchange Prologue and immediately before any optional Exchange Epilogue fields. The format of the application content is implied by the Message Profile Id. Application protocols MAY specify that the format of application content depends additionally on the content of other fields in the message, e.g., the Message Type, et cetera.

Exchange Epilogue

Immediately following the Application Content, and immediately preceding the end of the plaintext of the body, space is reserved for zero or more octets of Exchange Epilogue, to be formatted according to the content of the Message Prologue fields, usually the Encryption Type field (see Encryption Of Message Body below) and the Exchange Prologue fields.

No fields are currently defined in the Exchange Epilogue. Its size is always zero.

Tunnel Content Body

If T=1 in the Message Prologue, then the plaintext of the message body is not an Exchange Data Structure. Instead, the plaintext of the message body is an encapsulated IPv6 packet [RFC2460] followed immediately by the Tunnel Content Epilogue.

If T=1 in the Message Prologue, then the Destination Node Id in the Message Options Prologue MUST NOT be the Any Node Id.

Tunnel Content Epilogue

Zero or more octets of space after the end of the IPv6 packet up to the end of the Message Body comprise the Tunnel Content Epilogue, to be formatted according to the content of the Message Prologue fields, usually the Encryption Type field (see Encryption Of Message Body below) and the content of the encapsulated IPv6 packet.

No fields are currently defined in the Tunnel Content Epilogue.

Message Epilogue

Immediately following the Message Body, and immediately preceding the end of the message, space is reserved for zero or more octets of Message Epilogue, to be formatted according to the content of the Message Prologue fields, usually the Encryption Type field (see Encryption Of Message Body below).

No further content in the Message Epilogue is currently defined.

Message Size Limits

The size of a Weave message is not strictly limited. However, a practical limit applies to the size of messages according to the underlying transports on every path from its source to all of its destinations. Various underlying transports entail different limits. Some transports offer guarantees of minimum path MTU across routes with multiple links and varying MTU. Others do not. It is incumbent on applications to be cognizant of the MTU for the paths on which destinations are expected to be reachable.

Some examples of the MTU applied to Weave messages by typical underlying transports are provided in the following list:

- Single-hop UDP/IPv4 (no IP fragments or IP options) — On links where MTU is 1500 octets, the maximum Weave message size is 1472 octets.

- Single-hop UDP/IPv6 (no IP fragments or other extension headers) — On links where MTU is 1500 octets, the maximum Weave message size is 1452 octets.

- Multi-hop UDP/IPv4 (public routes) — Due to the standard minimum MTU for IPv4 defined as 576 octets, the maximum Weave message size on all paths routed over such links is 548 octets. (Larger messages are possible on paths with MTU larger than the minimum.)

- Multi-hop UDP/IPv6 (public routes) — Due to the standard minimum MTU for IPv6 defined as 1280 octets, the maximum Weave message size on all paths routed over such links is 1232 octets. (Larger messages are possible on paths with MTU larger than the minimum.)

- UDP over Thread™ (no IP fragments or other extension headers) — Thread™ uses 6LoWPAN compression [RFC4944] over IEEE 802.15.4 wireless links, where the MTU is defined as 1280 octets. Accordingly, the maximum Weave message size on Thread™ networks is 1232 octets.

- UDP over Bluetooth™ Low Energy (no IP fragments or other extension headers) — As described in “IPv6 over Bluetooth® Low Energy” [RFC7668], the 6LoWPAN Fragmentation mechanism admits reassembly of IPv6 packets up to 2047 octets in length, so the maximum Weave message size here is 1999 octets.

- UDP/IPv6 in AES-256-CTR mode encrypted Weave Tunnel in TCP Serialized Message Stream — The overhead of the outer Weave tunnel encapsulation in this encryption mode is 26 octets, so the maximum Weave message size here is 65510 octets.

As a practical convention to provide optimal forwarding over all transports, the length of any Weave message SHOULD NOT exceed 1452 octets, which is the size of messages when transported in the payload of a single UDP/IPv6 packet without extension headers on a path with 1500 octet MTU. Some links commonly found in residential LAN networks and public Internet routes cannot transport or forward packets of larger size.

No support for Packetization Layer Path MTU Discovery (PLPMTUD) [RFC4821] is provided.

Serialized Message Stream

A stream of messages may be serialized in any stream of octets, e.g., a flat file, one side of a TCP/IP connection, an Internet Message Body [RFC2045], i.e., a MIME entity of Content-type: application/octet-stream. Accordingly, a Serialized Message Stream consists of one or more Weave Messages when each message is prepended by a 16-bit little-endian unsigned integer indicating the length in octets of the message that follows.

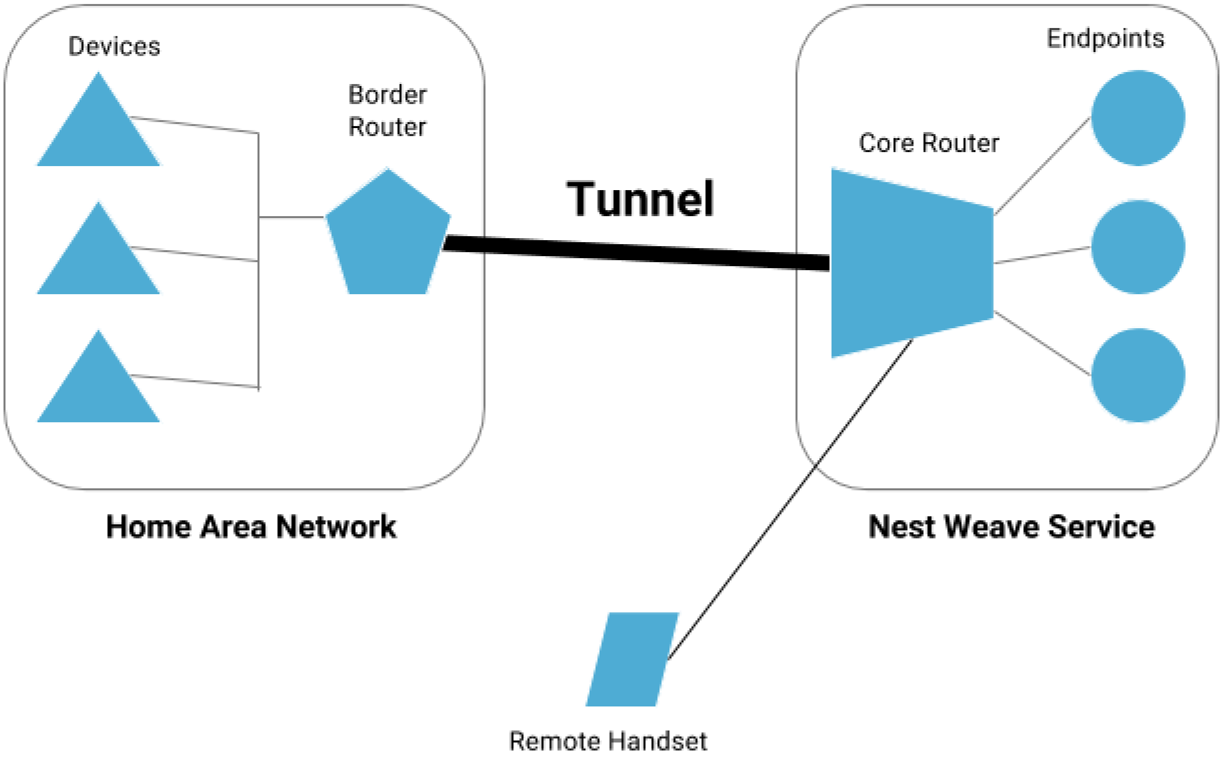
Accordingly, the length of any message presented for carriage in a Serialized Message Stream MUST NOT exceed 65,535 octets.

Operational Semantics

This section describes the operational semantics of the Weave Message Layer and defines the functional requirements of nodes when sending, receiving, and processing messages.

Fabric Topology and Message Routing

The Weave Message Layer admits the concept of a message router, which forwards messages within a fabric from one transport to another as shown below:



In this diagram, one IPv6 subnet in each fabric is assigned for the Weave Service Endpoints, another is assigned for each of the mobile devices, and additional subnets are assigned as necessary to one or more networks in each home area network associated with the fabric. The technical details of subnet number assignment are not in the scope of this document.

A distinguished category of device node is the Border Router, which establishes long-lived connections to the Weave Service and forwards messages between the service and other Weave device nodes connected to on-site networks using Weave tunnel form messages encapsulated in serialized message stream to carry IPv6 packets associated with the Weave fabric.

Mutable Fields In The Message Prologue

Forwarding nodes, i.e., intermediate nodes in Weave fabrics that forward messages in transit from their source to one or more destinations, SHOULD NOT decrypt, process, or modify the plaintext of the Message Body of forwarded messages.

The S and D fields in the Message Prologue MAY be modified by forwarding nodes provided that the corresponding Source Node Id and Destination Node Id fields are inserted or elided accordingly. Forwarding nodes MUST NOT elide the Source Node Id field or the Destination Node Id field when the underlying transport or some other ancillary channel of information does not carry sufficient information for the recipient to reconstruct their elided values. Forwarding nodes MUST NOT insert the Source Node Id or the Destination Node Id field if the value inserted is not equal to the value carried in any underlying transport and all other ancillary channels of information used for carrying the previously elided value.

Forwarding nodes MUST NOT modify, insert, or elide any fields in the Message Prologue other than the S, D, Source Node Id and Destination Node Id fields.

Reliable Delivery

The Weave Message Layer implements an optional reliable message delivery mechanism suitable for use with unreliable transport layers to retransmit messages until their reception is explicitly acknowledged in a subsequent response message. To optimize for suitability to constrained resource computing environments, the mechanism addresses only the problem of reliable delivery over lossy underlying transports using a positive acknowledgement carried in the optionally encrypted Exchange Prologue to protect against acknowledgement spoofing. Accordingly, it currently provides no rate adaptation, congestion avoidance, ordered delivery guarantee, or any support for path MTU discovery.

The reliable message delivery mechanism is defined for use when the underlying transport provides only best effort unreliable datagram service. Messages MUST NOT use the reliable delivery mechanism when they originate at a node where the underlying transport layer for any destination would provide reliable delivery, e.g., when a destination is reachable via a Serial Message Stream over a TCP connection.

The reliable message delivery mechanism is conceptually very simple. When the sender explicitly requires it, the sender MAY retransmit any unacknowledged previously sent message where R=1 in the Exchange Prologue.

When retransmitting, the sender MUST NOT change any octets in the original message. Also, the sender MUST NOT retransmit a message if A) the Destination Node Id field is equal to the Source Node Id of a previously received message M, and B) the Message Id field is equal to the Acknowledged Message Id field in M.

On receiving a message with R=1 in the Exchange Prologue, a node MUST reply with at least one response with A=1 in the Exchange Prologue and the Acknowledged Message Id field filled with the Message Id copied from the received message.

A node that receives a message with A=1 in the Exchange Prologue MUST NOT retransmit any message with Message Id equal to the Acknowledged Message Id in the received message.

Message Identifier Arithmetic

When the reliable delivery mechanism is used, the values used in the Message Id and Acknowledged Message Id fields are sequence numbers ranging from 0 to $2^{32} - 1$. Since the space is finite, all arithmetic dealing with Message Id values MUST be performed $modulo\ 2^{32}$. This unsigned arithmetic preserves the relationship between Message Id values as they cycle from $2^{32} - 1$ to 0 again. There are some subtleties to modulo arithmetic in typical computer programming environments, so great care is necessary in programming the comparison and addition of such values.

Comparison and addition of Message Id values SHOULD use Serial Number Arithmetic [RFC1982] where SERIAL_BITS = 32.

Detecting Duplicates of Received Messages

On receiving a message with R=1 in the Exchange Prologue, a node MAY use the method described here to detect whether the message is a duplicate. Application profiles MAY require duplicate messages to be handled differently from originals. For example, it’s typical that applications silently drop duplicate messages.

To detect duplicates, a Weave node maintains within each exchange context a list of previously received, potentially out of order, Message Id values in decreasing order according to Serial Number Arithmetic. This list is called the Reordering Window because it facilitates the discovery of duplicates when messages are received out of order, not the reordering of messages to recover the original order of transmission. When a message is received with a Message Id value already present in the window, the message is marked as a duplicate when presented to the application.

To permit constraining the size of the Reordering Window to a small finite length, a node MAY, after receiving the first message in an exchange, logically include in the window all values of Message Id for which addition of the finite size of the window produces a value that compares less than the Message Id of the most recently received message.

Upon creation of a fresh exchange, the Reordering Window is initially empty.

Standalone ACK

A node MAY reply immediately upon receiving a message with R=1 in the Exchange Prologue respond by sending a message with Message Profile Id zero, i.e., the Weave Common Profile, and with Message Type 2, i.e., the NULL message. When sent with R=0 and A=1 with the Acknowledged Message Id copied from the received message, this message is conventionally called a Standalone ACK message.

When a node receives the final application message in an exchange and R=1 is set in the Exchange Prologue, the receiver MUST reply to it with a Standalone ACK message.

Status Report

A common protocol design pattern is for servers to respond to those client requests that are not well-formed, or that request functions that cannot be completed accordingly, by sending a numeric status code indicating the application specific reason for the protocol failure.

As the final application message in an exchange, a node MAY send a Status message with Message Profile Id zero, i.e., the Weave Common Profile, and Message Type 1, i.e., the Status Report message. When sending a Status message, the server SHOULD set R=1 in the Exchange Prologue.

The Application Body of a Status message is encoded with Weave TLV [WTLV] as a structure container using the anonymous tag with two fields according to the following table:

Tag	**Description**
Profile Identifier	A 32-bit profile identifier for the Weave Profile that defines the scope of the status report code.
System Error Code	A 16-bit little endian integer code, with meaning defined by the Weave Profile.

Delivery Delayed Notification

At any time, a node MAY send to any other node a message with Message Profile Id zero, i.e., the Weave Common Profile, and with Message Type 3, i.e., the WRMP Delivery Delayed Notification message, to indicate that delivery of unacknowledged messages that were previously sent to a specific destination node is delayed by a specific number of milliseconds.

The Application Body of a WRMP Delivery Delayed Notification message comprises a 32-bit little endian unsigned integer specifying the number of milliseconds after message reception that delivery is known to be delayed by the network, followed by the eight octets of the Node Id of the final destination of the delayed messages.

Upon receipt of a WRMP Delivery Delayed Notification message, a node SHOULD NOT send any messages for the destination specified in the Application Body until the specified number of milliseconds have elapsed.

A node MUST NOT send a WRMP Delivery Delayed Notification message with R=1 in the Exchange Prologue.

Explicit Flow Throttle

At any time, a node MAY send to any other node a message with Message Profile Id zero, i.e., the Weave Common Profile, and with Message Type 4, i.e., the WRMP Flow Throttle message, to indicate that it will not be receiving messages for the number of milliseconds encoded in the Application Body as a 32-bit little-endian unsigned integer.

Upon receipt of a WRMP Flow Throttle message, a node SHOULD NOT send any messages destined for its Source Node Id until the number of milliseconds specified in the application body have elapsed.

A node MUST NOT send a WRMP Flow Throttle message with R=1 in the Exchange Prologue.

Encryption Of Message Body

The Weave Layer defines an extensible system of cryptographic algorithms for protecting the confidentiality, authenticity, and integrity of messages. The keys for encrypting and decrypting messages are managed by a conceptual internal database in every Weave node called the Weave Key Service. Negotiation and exchange of keys is described further in Weave Security Architecture [SECURITY] and Weave Fabric Provisioning Protocol [COMMISSION].

The following table defines the methods available for providing the body of Weave messages with cryptographic security properties.

Encryption Type	Description
0	No encryption or message integrity
1	HMAC-SHA-1 message integrity check, then AES-128-CTR encryption.

Further explanation is provided in the subsections below.

Type 0: No Encryption Or Message Integrity Check

With this value of Encryption Type in the Message Prologue, the Encryption Parameters Structure fields are not present in the Message Prologue. The entire Message Body appears in plaintext. No fields are present in either the Message Epilogue, the Exchange Epilogue, or the Tunnel Content Epilogue.

Type 1: HMAC-SHA-1 message integrity check, then AES-128-CTR encryption

With this value of Encryption Type in the Message Prologue, the Encryption Parameters Structure fields are present and used according to the following table:

Field Name	Operational Semantics
Message Key Index	One of the parameters used to obtain the AES-128 key from the Weave Key Service. A 16-bit little-endian int

The 16 octets for the AES-128-CTR counter is initialized as follows: the first 8 octets are the Source Node Id, the next 4 octets are the Message Id, and the last 4 octets are the block count, which are initially zero.

The 8 octets for the AES-128 key is obtained from the Weave Key Service by querying for the AES-128-CTR type with the Source Node Id and the Message Key Index as search parameters.

Decryption processes the ciphertext, consisting of the extent of both the Message Body and the Message Epilogue, with the counter and key obtained above to produce the plaintext.

Next, the HMAC-SHA-1 message integrity check code (20 octets) is computed for the message from the following input presented in order:

1. The Source Node Id value (which may be inferred from IID of the source address in the IPv6 packet header if its network prefix is the fabric ULA).

2. The Destination Node Id value (which may be inferred from IID of the destination address in the IPv6 packet header if its network prefix is the fabric ULA).

3. The Message Prologue with S and D reset to zero and the Source and Destination Node Id fields elided. (Not included in Version 1.)

4. The Message Id field. (Not included in Version 1.)

5. The Message Body.

The computed message integrity check code is compared with the value found in the Message Epilogue. If they are not identical, then the message received was not the message sent, and the plaintext of the Message Body MUST NOT be processed further.

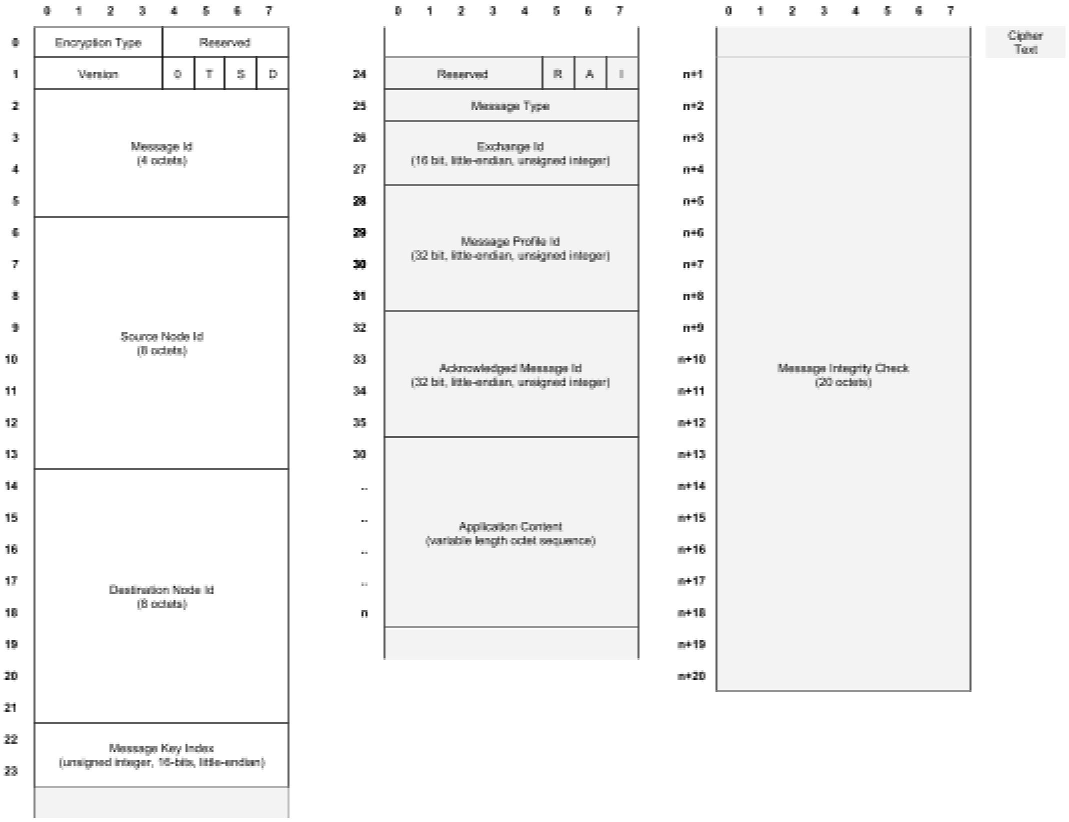
Encryption proceeds in the reverse order of operations, i.e., generate the message integrity check (MIC), append the MIC to the application content to obtain the plaintext, encrypt the plaintext, and produce the Encryption Parameters Structure with the Message Key Index required for decryption.

No additional Encryption Parameter Option fields are present in the Encryption Parameters Structure.

No fields are present in the Exchange Epilogue or the Tunnel Content Epilogue.

Example Message Layout

In the layout diagram below, the following values are set in the named fields of the Message Prologue: T=0, S=1, D=1 and Encryption Type=1. The message shown contains the optional Source Node Id and Destination Node Id fields. The Message Body is an encrypted General Content Body, and in the Exchange Prologue, the following values are set in the named fields: R=1 and A=1. The plaintext of the encrypted body (shown in grey) contains the optional Acknowledged Message Id field used by the reliable delivery mechanism.



References

Normative References

EUI	Guidelines for 64-bit Global Identifier
PROFILES	Weave: Common Profile Identifier Registry, Revision 14, August 2016
RFC1982	Elz R. and Bush R., “Serial Number Arithmetic”, RFC 1982, August 1996

RFC2119	Bradner S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
RFC2460	Deering S. and Hinden B., “Internet Protocol, Version 6 (IPv6) Specification”, RFC 2460, December 1998
RFC4086	Eastlake 3rd D., Schiller J. and Crocker S., "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005
RFC4193	Hinden R. and Haberman B., “Unique Local IPv6 Unicast Addresses”, RFC 4193, October 2005
WTLV	Weave TLV, Specification

Informative References

COMMISSION	Weave Fabric Provisioning Protocol
RFC768	Postel J., “User Datagram Protocol (UDP)”, RFC 768, August 1980
RFC793	Defense Advanced Research Projects Agency, “Transmission Control Protocol (TCP)”, RFC 793, September 1981
RFC2045	Freed N. and Borenstein N., “Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies”, RFC 2045, November 1996
RFC2360	Scott G., “Guide for Internet Standards Writers”, RFC 2360, June 1998
RFC4821	Mathis M. and Heffner J., “Packetization Layer Path MTU Discovery”, RFC 4821, March 2007
RFC4944	Montenegro G., Kushalnagar N., Hui J., and Culler D., “Transmission of IPv6 Packets over IEEE 802.15.4 Networks”, RFC 4944, September 2007
RFC7668	Nieminen J., Savolainen T., Isomaki M., Patil B., Shelby Z. and Gomez C. “IPv6 over BLUETOOTH® Low Energy”, RFC 7668, October 2015
SECURITY	Weave Security Architecture