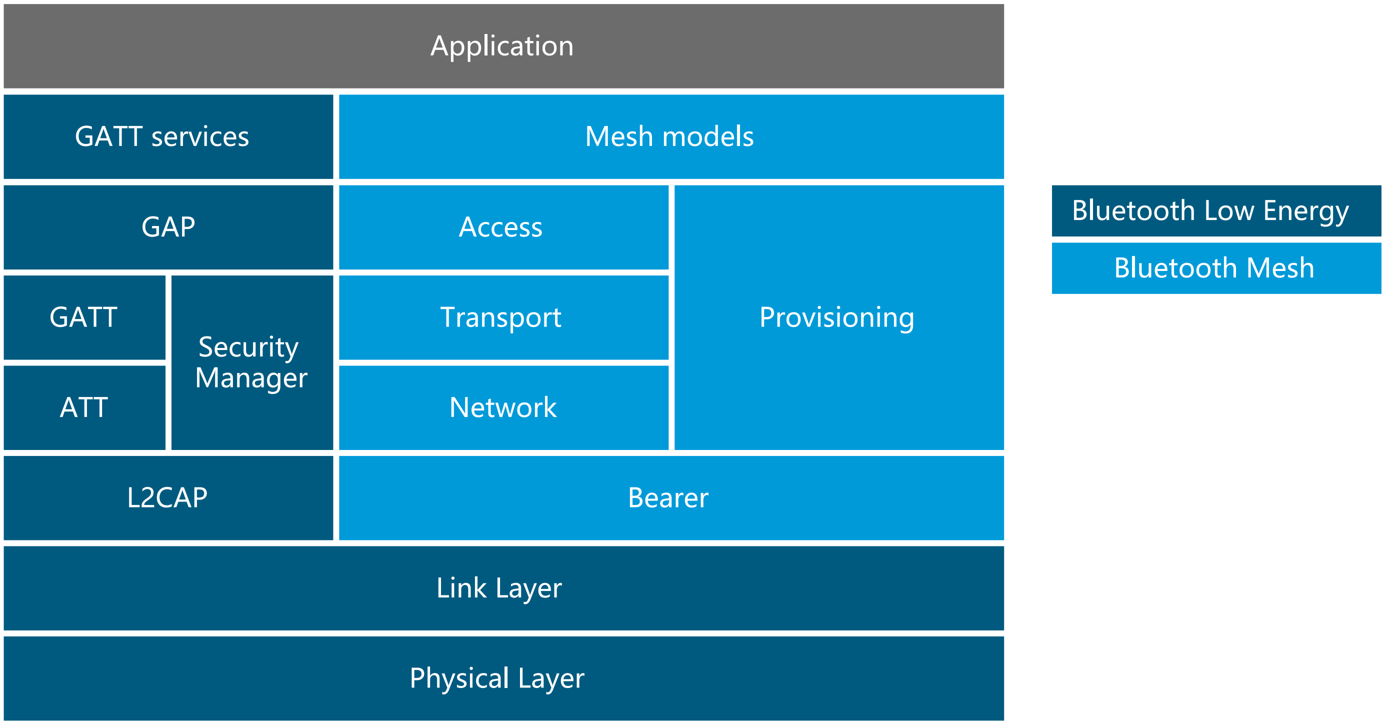
Copy URL <https://infocenter.nordicsemi.com/topic/com.nordic.infocenter.meshsdk.v3.1.0/md_doc_introduction_basic_concepts.html>

Basic Bluetooth Mesh concepts

Bluetooth Mesh is a profile specification developed and published by the Bluetooth SIG <http://www.bluetooth.org/>. This document explains the basic concepts of the Bluetooth Mesh and gives an overview of the operation and capabilities of the profile, as well as explaining the life cycle of a mesh device. For more specific information about Nordic Semiconductor's implementation of the Bluetooth Mesh, see mesh architecture documentation.

The Bluetooth Mesh is based on the Bluetooth low energy part of the Bluetooth 4.0 Specification and shares the lowest layers with this protocol. On-air, the Bluetooth Mesh physical representation is compatible with existing Bluetooth low energy devices, as mesh messages are contained inside the payload of Bluetooth low energy *advertisement* packets. However, Bluetooth Mesh specifies a completely new host layer, and although some concepts are shared, Bluetooth Mesh is incompatible with the Bluetooth low energy host layer.



Relationship between Bluetooth Mesh and Bluetooth low energy specifications

Read more about basic Bluetooth Mesh concepts in the following sections:

- Application areas
- Network topology and relaying, including information about transport, relays, power consumption, and GATT proxy.
- Addressing
- Models and elements
- Provisioning
- Network configuration
- Security, including information about authentication, message encryption, privacy key, and replay protection.

Application areas

Bluetooth Mesh primarily targets simple control and monitoring applications, like light control or sensor data gathering. The packet format is optimized for small control packets, issuing single commands or reports, and is not intended for data streaming or other high-bandwidth applications.

Using Bluetooth Mesh causes higher power consumption than traditional Bluetooth low energy applications. This is mainly due to the need for keeping the radio running constantly. Therefore, unlike Bluetooth low energy advertisers, active mesh devices cannot be run off coin-cell batteries for extended periods of time.

Bluetooth Mesh supports up to 32767 devices in a network, with a maximum network diameter of 126 hops.

Network topology and relaying

Bluetooth Mesh is a broadcast-based network protocol, where every device in the network sends and receives all messages to and from all devices within radio range. There is no concept of connections in a mesh network. Any device in the network may relay messages from any other device, which makes it possible for a mesh device to send a message to a device outside of radio range by having one or more other devices relay the message towards the destination. This property also allows devices to move around and drop in and out of the network at any time.

Mesh transport

Bluetooth Mesh utilizes the Bluetooth low energy advertiser and scanner roles, communicating through Bluetooth low energy advertisement packets. The advertisement packets are picked up by nearby mesh devices and handled like other Bluetooth low energy advertisement packets. The mesh packets are represented with a unique AD type and added to the advertisement packet payload.

Bluetooth low energy devices send advertisement packets at regular *advertisement intervals*, and mesh packets are no exception. However, unlike traditional advertisers, mesh devices will change their advertisement payload on every transmission, broadcasting new mesh packets as they are queued up in the stack. Every Bluetooth Mesh advertisement is transmitted only once for every device, and if there is no traffic in the mesh, the devices stay silent.

Relays

Bluetooth Mesh expands the range of the network by relaying messages. Any mesh device may be configured to act as a relay, and no dedicated relay devices are needed to build a network. Every device acting as a relay will decrement the Time To Live (TTL) value in received messages and forward them if the TTL is two or higher. This undirected relaying is referred to as message *flooding* and ensures a high probability of message delivery, without requiring any information on the network topology. The Mesh Profile Specification does not provide any routing mechanisms, and all messages are forwarded by all relays until the TTL value reaches zero. To avoid messages being forwarded by the same relays over and over, all mesh devices maintain a *message cache*. This cache is used for filtering out packets that the device has already handled.

The flooding based approach to message relaying can cause a lot of redundant traffic on air, which may impact the throughput and reliability of the network. Therefore, it is highly recommended to limit the number of relays in a network to restrict this effect. The rate of relay-enabled devices in the network is a trade-off between message route-redundancy and reliability. It should be tuned according to network density, traffic volumes, network layout, and requirements for reliability and responsiveness.

Power consumption

To enable broadcast-based communication, the devices must continuously keep their radio in listening mode, causing significantly higher power consumption than in a typical Bluetooth low energy device. To enable low power devices to take part in the mesh network, Bluetooth Mesh contains a low power *friendship* feature. This protocol lets low power devices establish a relationship with a regular mesh device, which will then cache and forward messages to the low power device at regular intervals. This saves the low power device from having to stay on to listen for incoming messages.

GATT proxy

To enable support for legacy Bluetooth low energy devices that do not support receiving mesh packets, Bluetooth Mesh defines a separate protocol for tunneling mesh messages over the Bluetooth low energy GATT protocol. For this purpose, the Mesh Profile Specification defines a GATT bearer and the corresponding GATT Proxy Protocol. This protocol allows legacy Bluetooth low energy devices to participate in the mesh network by establishing a GATT connection to a mesh device that has the proxy feature enabled.

The legacy device gets assigned an address and the necessary keys to become a full-fledged member of the network. The device receives the security credentials through the regular provisioning procedure or through some out-of-band mechanism.

Addressing

The Bluetooth Mesh addressing scheme is different from the Bluetooth low energy addressing scheme. It features three types of addresses:

- *Unicast addresses*: unique for every device
- *Group addresses*: allow forming a group of devices and addressing them all at once
- *Virtual addresses*: untracked UUID-based addresses with a large address space

When a device is added to a network, it is assigned a range of unicast addresses that represents it. A device's unicast addresses cannot be changed and are always sequential. The unicast address space supports having 32767 unicast addresses in a single mesh network. Unicast addresses can be used by any application to directly send a message to a device.

Group addresses are allocated and assigned as part of the network configuration procedure. A group address may represent any number of devices, and a device may be part of any number of groups. There can at most be 16127 general purpose group addresses in a mesh network.

The virtual addresses can be considered a special form of group addresses, and can be used to represent any number of devices. Each virtual address is a 128-bit UUID generated from a text label. The virtual addresses do not have to be tracked by a network configuration device, and in this way, users can generate virtual addresses prior to deployment or addresses can be generated ad-hoc between devices in the network.

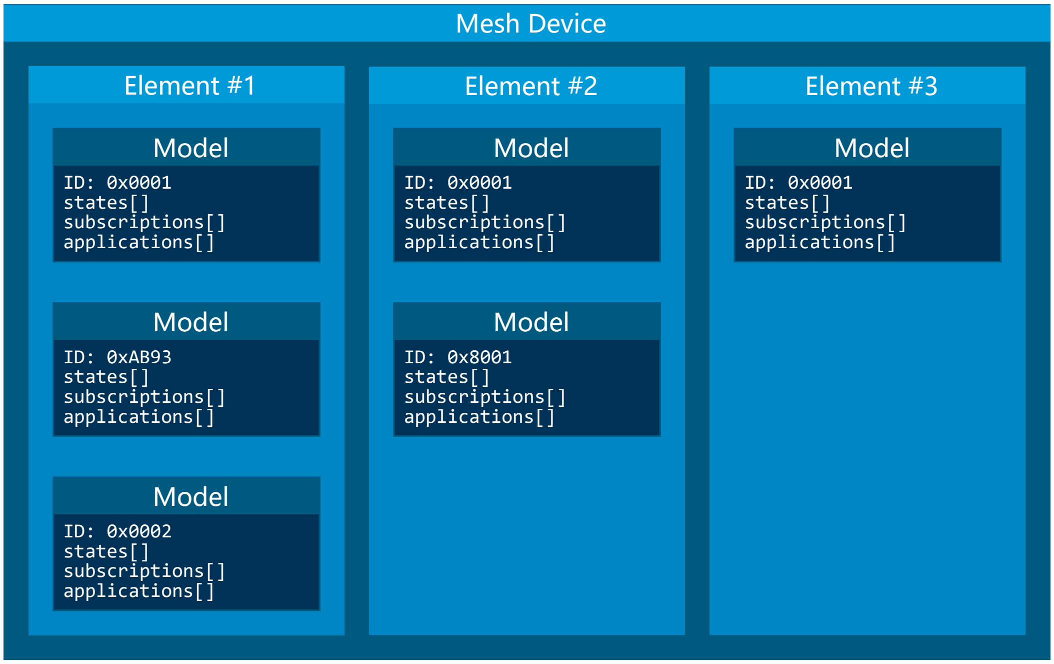
Models and Elements

To standardize communication between devices from different vendors, the Mesh Profile Specification defines an access layer, which routes mesh messages between the various models in a device. A model represents a specific behavior or service and defines a set of states and messages that act on these states. The Mesh Profile Specification and the Mesh Model Specification each define a set of models to cover typical usage scenarios like device configuration, sensor readings, and light control. In addition to these, vendors are free to define their own models with accompanying messages and states.

The models in a device belong in elements. Every device has one or more elements, each acting as a virtual entity in the mesh with its own unique unicast address. Each incoming message is handled by a model instance in an element. To make it possible to uniquely resolve how messages are handled, only one model instance per element can implement a handler for a specific message opcode. If a device has multiple instances of the same model, each instance must be assigned to a separate element. Similarly, if two models implement handlers for the same message, these models must be in separate elements.

To represent complex behavior with minimal message and state duplication, models can be made up of other models, potentially spanning multiple elements. These models are referred to as extended models. Models that are purely self-contained are referred to as root models.

Models talk to each other through a publish and subscribe system. Every model may subscribe to a set of group and virtual addresses, and the model will only handle messages that are published to one of its subscription addresses or the containing element's unicast address. Any model may maintain a publish address that it publishes messages to. This publish address can be of any type.



Access layer structure

Provisioning

Before a device can participate in normal mesh operation, it must be provisioned. During provisioning, a device gets added to the network and is assigned unicast addresses, a network key, and a device key. The provisioning is done by a *Provisioner*, which is a trusted device with access to the full list of devices in the network and their addresses. After the new device has been provisioned, the provisioner is expected to use the new device's device key to establish a secure channel to configure it.

Network configuration

Bluetooth Mesh leaves the network configuration to a central network configurator. Devices are not expected to do any sort of service discovery on their own. To control other devices, a device like a light switch must be configured by a provisioner, either through user interaction or by loading a predetermined configuration from a database. Every device must implement a mandatory Configuration Server model in their first element, which is used to configure the rest of its models.

As soon as provisioning is complete, the provisioner uses its instance of the Configuration Client model to give the new device a set of application keys and addresses. The device will use these keys and addresses for the duration of its lifetime on the network, unless it gets reconfigured.

Example scenario: A light bulb and a switch

After a new light switch has been provisioned, the Configuration Client model in the provisioner reads out a list of the new device's models and elements and presents them to the user. The user finds the light switch model in the device's model list and gives it the "Light Control" application key. Next, the user sets the model's publish address to the "Kitchen Area" group address, to which all the light bulbs in the kitchen subscribe. The next time the new light switch is pressed, all light bulbs in the kitchen turn on.

Security

Bluetooth Mesh employs several security measures to prevent third-party interference and monitoring:

- Authentication
- Message encryption
- Privacy key
- Replay protection

Authentication

Device authentication is part of the provisioning process and lets the user confirm that the device being added to the network is indeed the device they think it is. The Mesh Profile Specification defines a range of out-of-band authentication methods, such as blinking of lights, output and input of passphrases, and static authentication against a pre-shared key. To secure the provisioning procedure, elliptic curve Diffie-Helman (ECDH) public key cryptography is used. After a device has been provisioned, it is part of the network and all its messages are considered authenticated.

Message encryption

Bluetooth Mesh features two levels of AES-CCM encryption with 128-bit keys for all messages going across the network:

- **Network encryption**: The lowest layer that protects all messages in a mesh network from being readable by devices that are not part of the network. The encryption is done with a network encryption key, and any network may consist of up to 4096 different subnets, each with their own network key. All devices sharing a network key are considered part of the network and may send and relay messages across it. By using multiple network keys, a network administrator may effectively divide their network into multiple subnets, because a mesh relay only forwards messages that are encrypted with a known network key.
- **Transport encryption**: The second encryption layer that limits which devices can do what *within a network* by encrypting the application payload with an application or device key. As an example, consider a mesh network deployed in a hotel, where it is desirable to limit some features to be controlled by the staff (like configuration of key cards or access to storage areas) and some features to be available to guests (like controlling room lighting or air conditioning). For this, we can have one application key for the guests and one for the staff, allowing the messages to be relayed across the same network, while preventing the guests and the staff from reading each other's messages.

While application keys are used to separate access rights to different applications in the network, the device keys are used to manage devices in the network. Every device has a unique device key, which is only known to the provisioner and the device itself. The device key is used when configuring a device with new encryption keys (network or application keys) or addresses, in addition to setting other device-specific parameters. It can also be used to evict malicious devices from a network by transferring new keys to all the other devices in the network (using their individual device keys when transferring the keys). This process is called the *Key Refresh Procedure*.

Each encryption layer contains a message integrity check value that validates that the content of the message was encrypted with the indicated encryption keys.

Privacy key

All mesh message payloads are fully encrypted. Message metadata like source address and message sequence number is obfuscated with the privacy key, derived from the network key, providing limited privacy even for public header fields.

Replay protection

To guard against malicious devices replaying previous messages, every device keeps a running sequence number, which is used for outbound messages. Each mesh message is sent with a unique pair of sequence number and source address. When receiving a message, the receiving device stores the sequence number and makes sure that it is more recent than the last sequence number it received from the same source address.

Documentation feedback | Developer Zone <https://devzone.nordicsemi.com/questions/> | Subscribe | Updated 2019-04-26