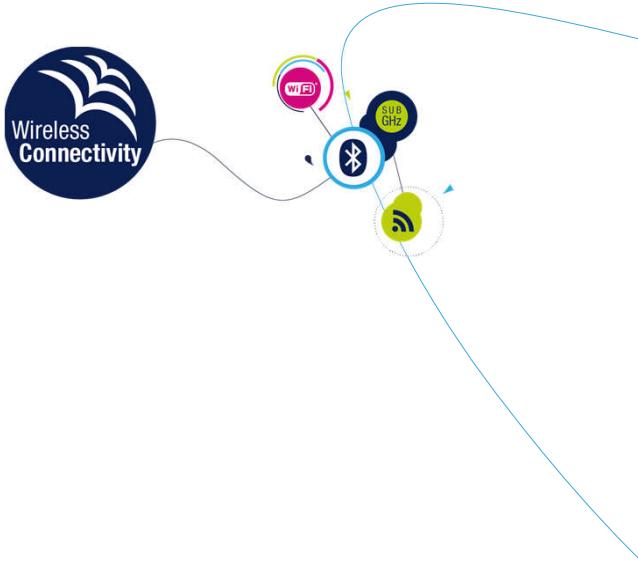
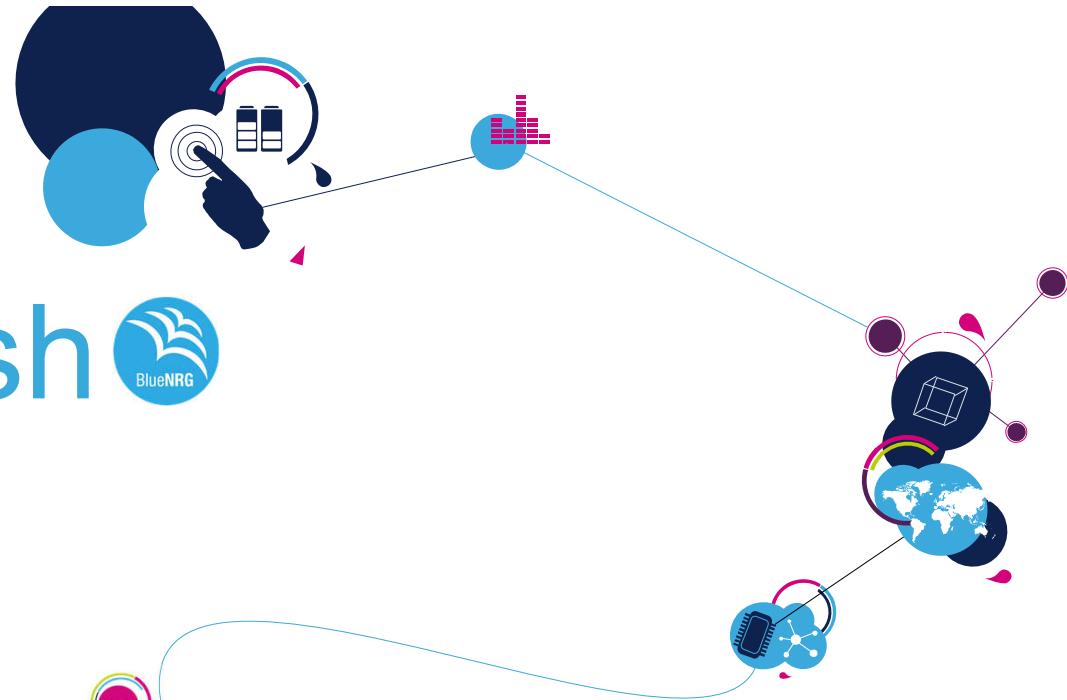


# BlueNRG-Mesh



# BlueNRG-Mesh



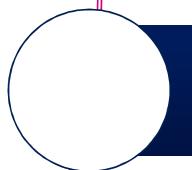
# BlueNRG-Mesh Agenda



Bluetooth Mesh Introduction



BlueNRG-Mesh Solution



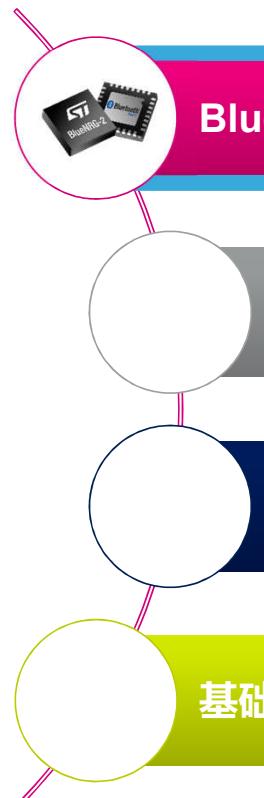
Getting started with BlueNRG-Mesh



Basics and Hands on Session



# BlueNRG-Mesh 日程



# Video & Demo

5

- <https://www.youtube.com/watch?v=5EKoEcSnXP0>

# 视频和演示

6

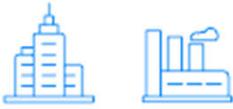
- <https://www.youtube.com/watch?v=5EKoEcSnXP0>

# Bluetooth Mesh official announcement

7

## July 18, 2017 - Bluetooth SIG Announced Mesh Networking Capability

Brings proven, global interoperability and the mature, trusted ecosystem of Bluetooth technology to industrial-grade device networks

industrial-grade solution	proven, global interoperability	mature, trusted technology
 <p>Bluetooth mesh uniquely meets the reliability, scalability and security requirements of building and factory automation markets that demand true industrial-grade solutions.</p> <ul style="list-style-type: none"><li>• <b>Reliability:</b> Enables inherently self-healing networks with no single points of failure</li><li>• <b>Scalability:</b> Supports thousands of nodes with industrial-level performance</li><li>• <b>Security:</b> Provides industrial-grade security for protection against all known attacks</li></ul>	 <p>Only Bluetooth mesh delivers the proven multi-vendor interoperability that enables markets to flourish and assures that products from different vendors across the globe work together.</p> <ul style="list-style-type: none"><li>• <b>A full-stack solution:</b> A unique full stack approach that defines everything from the low-level radio to the high-level application layer, ensuring all levels of the technology are fully specified</li><li>• <b>An interop-centric spec:</b> Comprehensive interoperability testing conducted prior to specification release, not after</li><li>• <b>Time-tested tools and processes:</b> A 20-year history of delivering the qualification tools and processes necessary to ensure global, multi-vendor interoperability</li></ul>	 <p>The value-added capabilities, mature ecosystem and global brand awareness that Bluetooth wireless technology provides enable the creation of much richer solutions with a faster time to market.</p> <ul style="list-style-type: none"><li>• <b>Value-added services:</b> A mesh network built on Bluetooth can also provide localized information, asset tracking and way-finding services</li><li>• <b>A mature ecosystem:</b> The best enabling technology, along with the development and test tools and services needed to shrink your time to market</li><li>• <b>Global brand awareness:</b> A trusted global brand that stands for simple, secure wireless connectivity</li></ul>

# Bluetooth Mesh 官方通告

2017年7月18日 - Bluetooth SIG 通告 Mesh 组网

给工业级设备组网带来可验证的，全球互通和成熟的，可信任的生态系统

## industrial-grade solution



Bluetooth mesh uniquely meets the reliability, scalability and security requirements of building and factory automation markets that demand true industrial-grade solutions.

- **Reliability:** Enables inherently self-healing networks with no single points of failure
- **Scalability:** Supports thousands of nodes with industrial-level performance
- **Security:** Provides industrial-grade security for protection against all known attacks

## proven, global interoperability



Only Bluetooth mesh delivers the proven multi-vendor interoperability that enables markets to flourish and assures that products from different vendors across the globe work together.

- **A full-stack solution:** A unique full stack approach that defines everything from the low-level radio to the high-level application layer, ensuring all levels of the technology are fully specified
- **An interop-centric spec:** Comprehensive interoperability testing conducted prior to specification release, not after
- **Time-tested tools and processes:** A 20-year history of delivering the qualification tools and processes necessary to ensure global, multi-vendor interoperability

## mature, trusted technology



The value-added capabilities, mature ecosystem and global brand awareness that Bluetooth wireless technology provides enable the creation of much richer solutions with a faster time to market.

- **Value-added services:** A mesh network built on Bluetooth can also provide localized information, asset tracking and way-finding services
- **A mature ecosystem:** The best enabling technology, along with the development and test tools and services needed to shrink your time to market
- **Global brand awareness:** A trusted global brand that stands for simple, secure wireless connectivity

PRESS RELEASES

## Bluetooth SIG Announces Mesh Networking Capability

7/18/2017 8:00:00 AM

9

# BlueNRG-Mesh is here



companies supporting the launch  
of Bluetooth mesh networking



PRESS RELEASES

## Bluetooth SIG Announces Mesh Networking Capability

7/18/2017 8:00:00 AM

10

# BlueNRG-Mesh 在这里



companies supporting the launch  
of Bluetooth mesh networking





# Extending Bluetooth Capabilities

11

## PAIRING one-to-one



### DATA TRANSFER

- Sports & fitness devices
- Health and wellness devices
- Peripherals and accessories

## BROADCASTING one-to-many

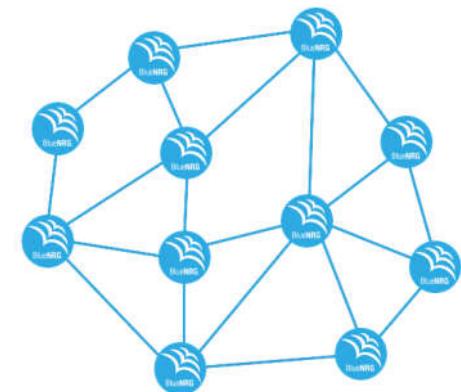


### LOCALIZED INFORMATION

- Point of interest beacons
- Item finding beacons
- Way finding beacons

NEW

## MESH many-to-many



### LARGE DEVICE NETWORKS

- Building automation
- Wireless sensor networks
- Asset tracking



# 扩展的蓝牙能力

12

## 配对 单点到单点



### 数据传输

- 运动穿戴装置
- 健康装置
- 外设和附件

## 广播 单点到多点

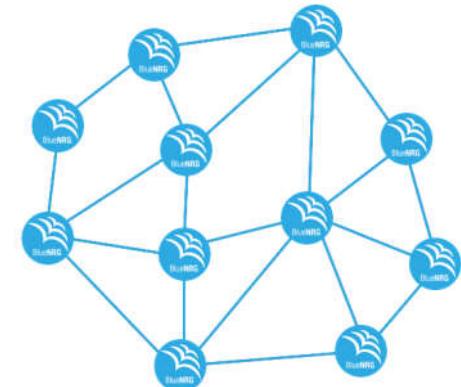


### 本地信息

- 单点接入关注的Beacon装置
- 条目搜寻 beacons
- 方式搜寻 beacons

NEW

## MESH 多点到多点



### 大设备网络

- 搭建自动化
- 无线传感器网络
- 资产追踪

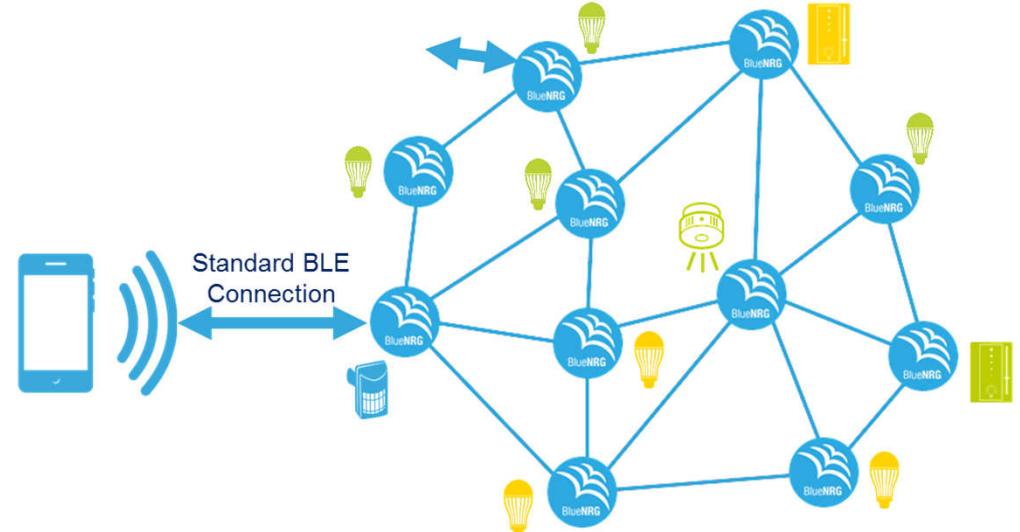
# Bluetooth Mesh Overview

13

- Mesh created over BLE connected nodes
- Key Features
  - Mesh Profile specifications defined by Bluetooth SIG
  - Data transfers are Flooding based
  - Multicast data transfer (Custom groups defined by end-user)
  - Continue to work even when one or more devices are moved or stop operating
  - Low power consumption

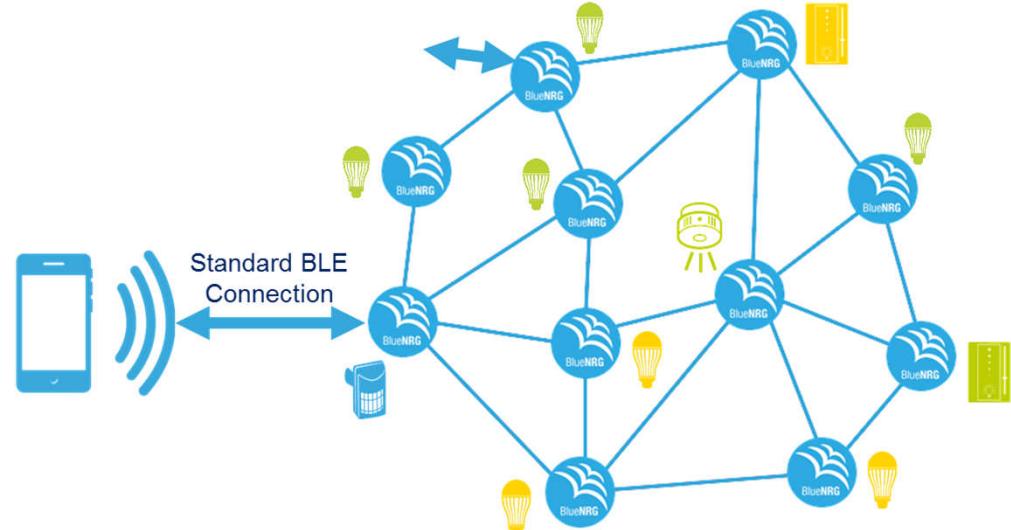
- Security

- All messages in the network are authenticated
- Messages are sent with sequence numbers to protect against replay attacks



# 蓝牙 Mesh 简介

- Mesh 是基于连接BLE的节点
- 关键特性
  - Mesh Profile 规格是由蓝牙SIG 定义的
  - 数据传输时基于Flooding 型
  - 多广播数据传输 (用户定义分组)
  - 即使有一个或多个设备拿走或停止操作，系统仍然可持续工作
  - 低功耗
- 安全
  - 网络内所有的信息都需要认证
  - 防止重传信息发送的时候都带有序列号



# Bluetooth Mesh Applications

The Bluetooth SIG Mesh Specification intends to extend the capabilities of Bluetooth Smart chips to answer **more and more complex applications**. The protocol has been developed with the **Smart Lighting industry** in mind.

- Building automation
- Wireless sensor networks
- Asset tracking
- Smart home
- Street lighting
- Industry 4.0
- ...



# 蓝牙 Mesh 应用

16

蓝牙特别兴趣小组Mesh规范 扩展了Bluetooth Smart复杂的应用场景. 该协议规范同智慧照明工业的理念发展而来。

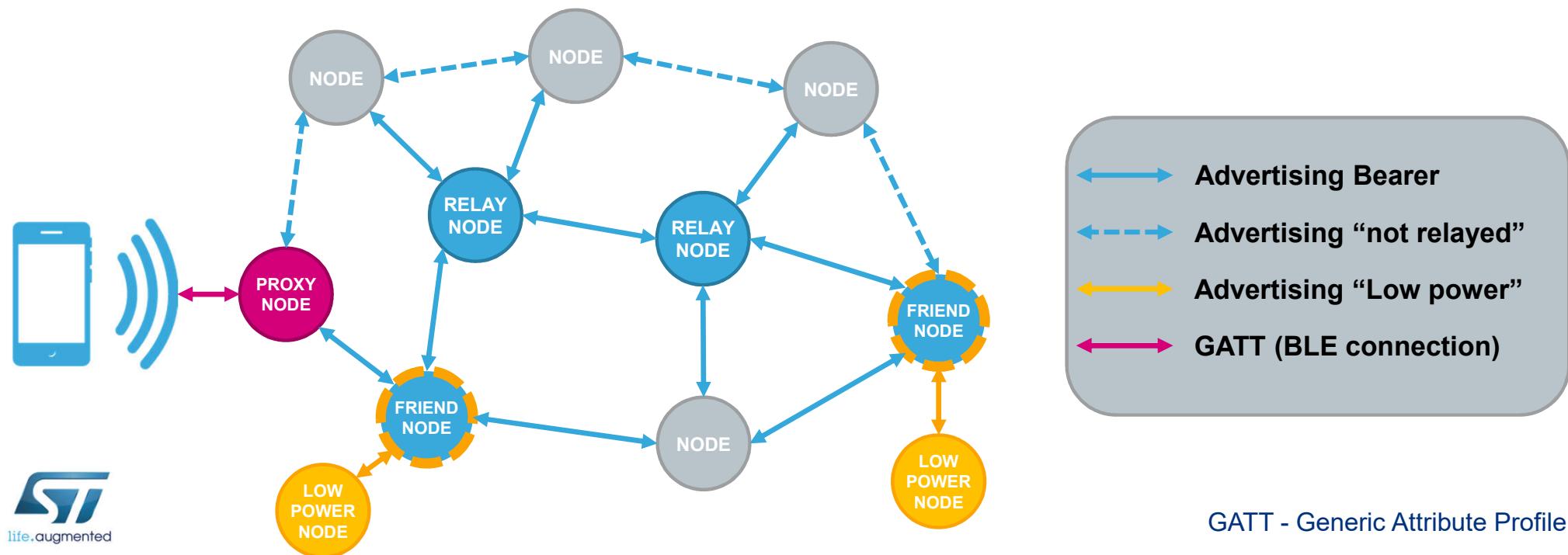
- 搭建自动化
- 无线传感网络
- 设备追踪
- 智能家庭
- 道路照明
- 工业4.0
- ...



# Bluetooth Mesh Topology

17

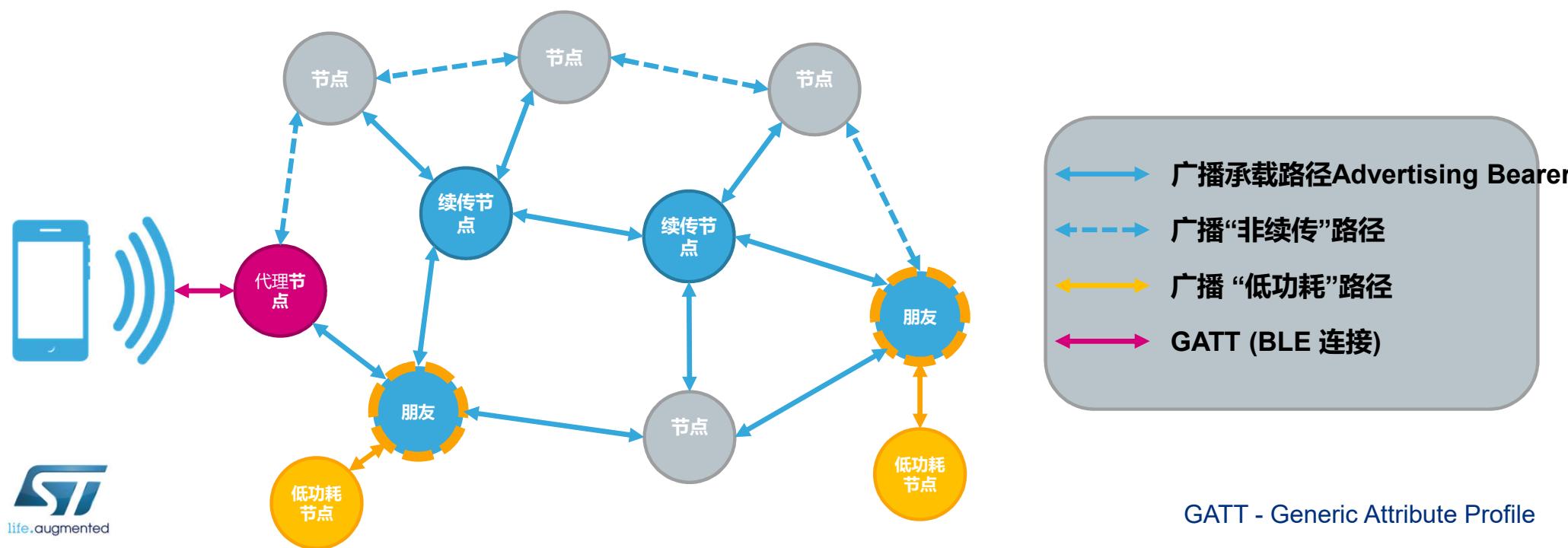
The Bluetooth Mesh working group chose for mesh network mechanism a **flooding protocol**. Compared to routed protocols, it is **much simpler** to deploy. To stay efficient, the BLE Mesh take advantage of a **managed flooding network**.



# 蓝牙Mesh 拓扑结构

18

蓝牙 Mesh 工作组选择的 mesh 是**泛洪协议**. 对比路由协议, 它**更加简单** 适用于开发. 为了保持效率, BLE Mesh 吸收了**可管理的泛洪网络**的优势.



GATT - Generic Attribute Profile

# Mesh Technologies Comparison

Technology	Advantage	Disadvantage	Comments
BLE (Mesh)  	<ul style="list-style-type: none"> <li>✓ Available in phones</li> <li>✓ No single point of failure</li> <li>✓ High security</li> <li>✓ Low-Power</li> <li>✓ SIG standard</li> </ul>	<ul style="list-style-type: none"> <li>○ Low bandwidth</li> </ul>	<ul style="list-style-type: none"> <li>▪ Has key the advantages over other technologies for smart-home</li> <li>▪ Widely adopted technology (iOS, Android, Windows, ..)</li> </ul>
ZigBee  	<ul style="list-style-type: none"> <li>✓ Stable and Mature</li> </ul>	<ul style="list-style-type: none"> <li>○ Not available in phones</li> <li>○ Need a Gateway to phone</li> <li>○ Gateway may be single point of failure</li> <li>○ 250kbps data-rate</li> </ul>	<ul style="list-style-type: none"> <li>▪ Disadvantaged for home-automation</li> </ul>
6LoWPAN  	<ul style="list-style-type: none"> <li>✓ IEEE standard</li> </ul>	<ul style="list-style-type: none"> <li>○ Not available in phones</li> <li>○ Needs a gateway to phone</li> <li>○ No standard for all communication layers</li> </ul>	<ul style="list-style-type: none"> <li>▪ Wi-SUN is aligned more for Smart-City applications</li> </ul>
Wi-Fi  	<ul style="list-style-type: none"> <li>✓ Available in Phone</li> <li>✓ High Bandwidth</li> <li>✓ Web connectivity</li> <li>✓ Some Mesh examples available</li> </ul>	<ul style="list-style-type: none"> <li>○ High power requirement limits usability for home-automation</li> <li>○ Expensive</li> </ul>	<ul style="list-style-type: none"> <li>▪ Widely adopted technology</li> </ul>
ANT+  	<ul style="list-style-type: none"> <li>✓ Available in limited Phones</li> <li>✓ Few mesh demo</li> </ul>	<ul style="list-style-type: none"> <li>○ Only limited number of phones support it</li> </ul>	

# Mesh 技术对比

技术	优点	缺点	注释
BLE (Mesh) 	<ul style="list-style-type: none"> <li>✓ 手机端适用</li> <li>✓ 单点失败不会影响系统网络</li> <li>✓ 高安全性</li> <li>✓ 低功耗</li> <li>✓ SIG 标准</li> </ul>	<ul style="list-style-type: none"> <li>○ 窄带</li> </ul>	<ul style="list-style-type: none"> <li>▪ 针对智慧家庭和其他技术对比有关键的优势</li> <li>▪ 被广泛兼容的技术(iOS, Android, Windows, ..)</li> </ul>
ZigBee 	<ul style="list-style-type: none"> <li>✓ 稳定和成熟</li> </ul>	<ul style="list-style-type: none"> <li>○ 手机端不适用</li> <li>○ 需要一个网关才能接入手机</li> <li>○ 网关会因一个节点失败而失效</li> <li>○ 250kbps 速率</li> </ul>	<ul style="list-style-type: none"> <li>▪ 不利于智慧家庭的构建</li> </ul>
6LoWPAN 	<ul style="list-style-type: none"> <li>✓ IEEE 标准</li> </ul>	<ul style="list-style-type: none"> <li>○ 手机端不适用</li> <li>○ 需要一个网关才能接入手机</li> <li>○ 对于所有的通信层标准</li> </ul>	<ul style="list-style-type: none"> <li>▪ Wi-SUN适用于更多的智慧城市的应用</li> </ul>
Wi-Fi 	<ul style="list-style-type: none"> <li>✓ 手机端适用</li> <li>✓ 宽带</li> <li>✓ Web 可连接</li> <li>✓ 一些 Mesh 实例可实现</li> </ul>	<ul style="list-style-type: none"> <li>○ 功耗高限制应用</li> <li>○ 昂贵</li> </ul>	<ul style="list-style-type: none"> <li>▪ 广泛被采纳的技术</li> </ul>
ANT+ 	<ul style="list-style-type: none"> <li>✓ 部分手机支持</li> <li>✓ 几乎没有 mesh 实例</li> </ul>	<ul style="list-style-type: none"> <li>○ 只有少数手机支持</li> </ul>	



# Built-in Security

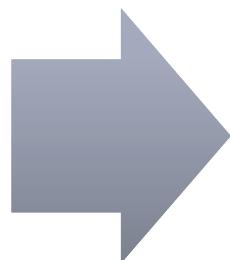
21

## Unconfigured device

### Provisioning

- 256-bit ECDH algorithm (public/private keys)
- Exchange several **security keys**
- Set a **unique address** for the device
- **Two separate BLE connections** to secure the communication link

Device added  
to the network



## Configured node

### Mesh communications

- Encrypted with 128-bit AES-CCM
- Multiple layers security thanks to the network/application keys shared during provisioning
- **Privacy through obfuscation**
- **Protection from multiple types of attack:**
  - Replay attack
  - Bit-Flipping attack
  - Eaves Dropping attack
  - Man-in-the-middle attack
  - Trashcan attack



## 未配置的设备

### Provisioning

- 256位 ECDH 算法 (公开/私有密钥)
- 交换几个安全密钥
- 设置设备**唯一地址**
- **2个独立BLE连接** 保障安全通信链路

设备加进网络



## 配置的节点

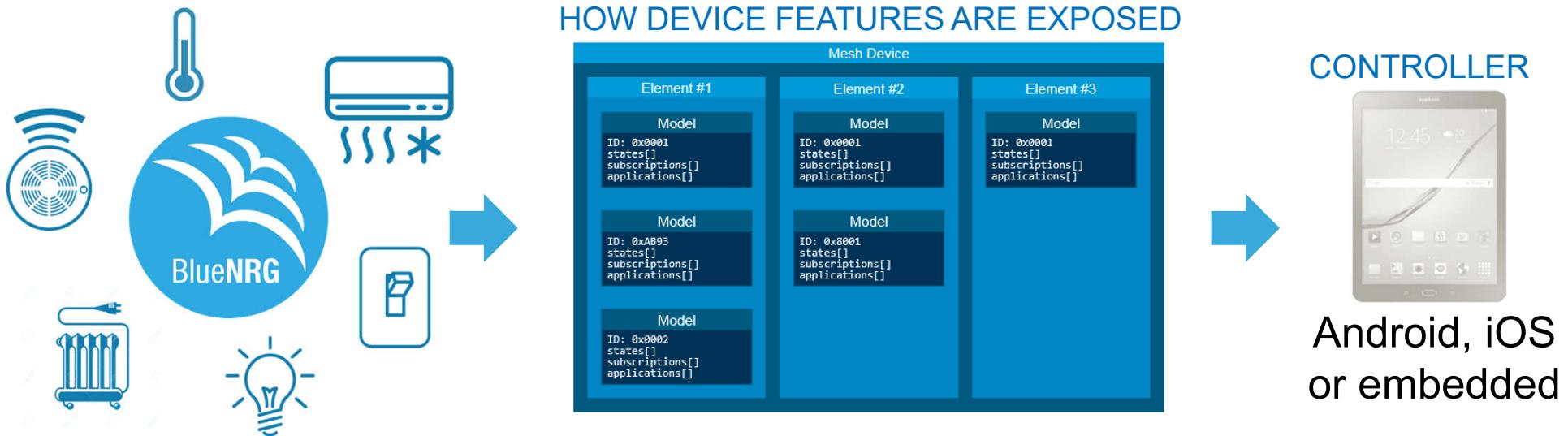
### Mesh 通信

- 128-位 AES-CCM 加密
- 基于provisioning期间分享的网络/应用密钥的多层安全机制
- **私有模糊处理**
- **保护多种类型的攻击:**
  - Replay 攻击
  - Bit-Flipping 攻击
  - Eaves Dropping 攻击
  - Man-in-the-middle 攻击
  - Trashcan 攻击

# Proven, Global Interoperability

23

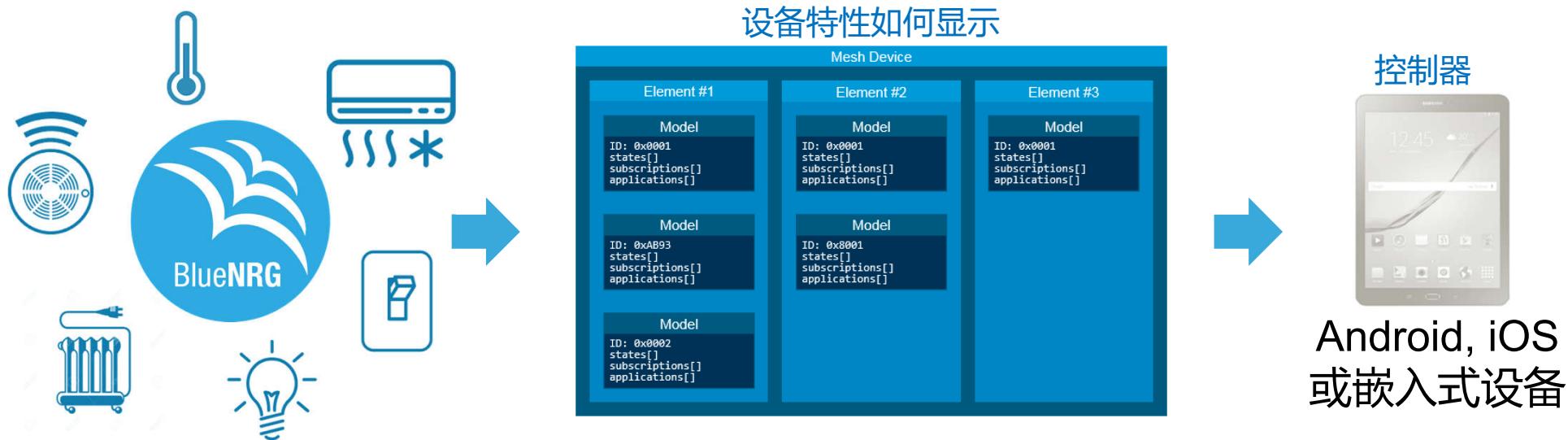
The Bluetooth SIG Mesh Profile defines a **standard access layer** to **exchange messages** between a BLE device and a Smart Application (Controller) for simple **control and monitoring applications** using models.



A **model** represents a specific **behavior or service** and defines a set of **states and messages** that act on these states.  
Model examples are **Device Configuration, Sensor Reading, Light Control or Vendor Specific** models.

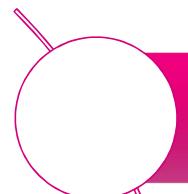
# Proven, Global Interoperability

蓝牙 SIG Mesh 规范定义了简单**控制和监控应用的**模型，就是在BLE设备和智慧应用（控制器）之间的加入一个**交换信息的标准接入层**.



一个模型代表了一个特殊的行为和服务的规范，同时定义了为了实现该规范的一整套状态和信息。模型的例子是设备的配置，传感器数据读取，以及控制灯或者厂商规格的模型。

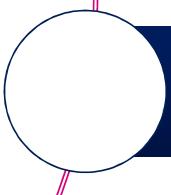
# BlueNRG-Mesh



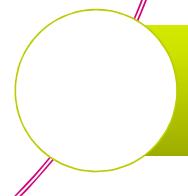
Bluetooth Mesh Introduction



BlueNRG-Mesh Solution



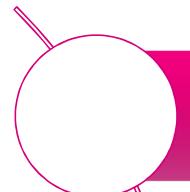
Getting started with BlueNRG-Mesh



Basics and Hands on Session



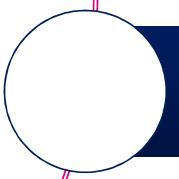
# BlueNRG-Mesh



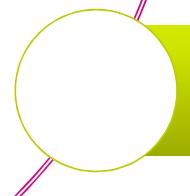
Bluetooth Mesh 介绍



BlueNRG-Mesh 方案



开启BlueNRG-Mesh之旅



基础和动手实践环节



# BlueNRG-Mesh

Bluetooth® Low Energy (BLE) System-on-Chip  
accelerates the spread of connected objects



Easily connecting appliances to iOS/Android, out-of-the-box



# BlueNRG-Mesh

Bluetooth® Low Energy (BLE) System-on-Chip  
accelerates the spread of connected objects

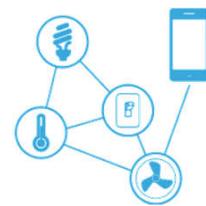


► Discover more about BlueNRG-2



简易连接到iOS/Android, 拆箱即用





# STSW-BNRG-Mesh

- SDK includes: Firmware, Android, iOS
- [www.st.com/blemesh](http://www.st.com/blemesh)

## GET SOFTWARE

Part Number	Software Version	Marketing Status	Supplier	
BlueNRG-Mesh for Android		Active	ST	<a href="#">GO TO SITE</a>
BlueNRG-Mesh for iOS		Active	ST	<a href="#">GO TO SITE</a>
STSW-BNRG-Mesh	1.03.000	Active	ST	<a href="#">Get Software</a>



ST life.augmented

≡ Menu

Home > Embedded Software > Wireless Connectivity Software > STSW-BNRG-Mesh

**STSW-BNRG-Mesh** ACTIVE

Mesh over Bluetooth Low Energy

[Download Databrief](#)

[QUICK VIEW](#) [RESOURCES](#) [TOOLS AND SOFTWARE](#) [GET SOFTWARE](#)

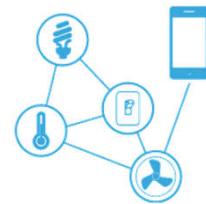
BlueNRG-Mesh is a software solution for connecting multiple BLE (Bluetooth low energy) devices in Mesh networks for Internet of Things (IoT) solutions. It enables true two-way communication between Bluetooth-enabled devices in powerful, secure, integrated and range-extending Mesh networks.

The solution is compatible with the ST BlueNRG product family range.

**Key Features**

- Mesh network with Bluetooth low energy (BLE) nodes enabling communication between a BLE device and a Smartphone
- Control and monitor applications involving short packets and infrequent communication
- Advertising packets used for data communication using managed flooding method
- Multi-hop data transmission
- Network node support up to 32,767 nodes and up to 126 hops
- Multiple communication scenario
  - Smartphone to node communication with unicast addressing
  - Smartphone to node communication with multicast (Group) addressing
  - Smartphone to node communication with broadcast addressing
  - Node to node communication
- Secure communication
  - Devices added to a network are provisioned using proven security algorithms using 256-bit elliptic curves
  - All messages in the network are encrypted with AES-128 CCM mode
  - Privacy through obfuscation

# STSW-BNRG-Mesh



- SDK 包含: 固件, Android, iOS
- [www.st.com/blemesh](http://www.st.com/blemesh)

## GET SOFTWARE

Part Number	Software Version	Marketing Status	Supplier	
BlueNRG-Mesh for Android		Active	ST	<a href="#">GO TO SITE</a>
BlueNRG-Mesh for iOS		Active	ST	<a href="#">GO TO SITE</a>
STSW-BNRG-Mesh	1.03.000	Active	ST	<a href="#">Get Software</a>



☰ Menu

Home > Embedded Software > Wireless Connectivity Software > STSW-BNRG-Mesh

## STSW-BNRG-Mesh ACTIVE

### Mesh over Bluetooth Low Energy

Download Databrief

QUICK VIEW

RESOURCES

TOOLS AND SOFTWARE

GET SOFTWARE

BlueNRG-Mesh is a software solution for connecting multiple BLE (Bluetooth low energy) devices in Mesh networks for Internet of Things (IoT) solutions. It enables true two-way communication between Bluetooth-enabled devices in powerful, secure, integrated and range-extending Mesh networks.

The solution is compatible with the ST BlueNRG product family range.

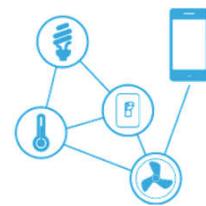
#### Key Features

- Mesh network with Bluetooth low energy (BLE) nodes enabling communication between a BLE device and a Smartphone
- Control and monitor applications involving short packets and infrequent communication
- Advertising packets used for data communication using managed flooding method
- Multi-hop data transmission
- Network node support up to 32,767 nodes and up to 126 hops
- Multiple communication scenario
  - Smartphone to node communication with unicast addressing
  - Smartphone to node communication with multicast (Group) addressing
  - Smartphone to node communication with broadcast addressing
  - Node to node communication
- Secure communication
  - Devices added to a network are provisioned using proven security algorithms using 256-bit elliptic curves
  - All messages in the network are encrypted with AES-128 CCM mode
  - Privacy through obfuscation

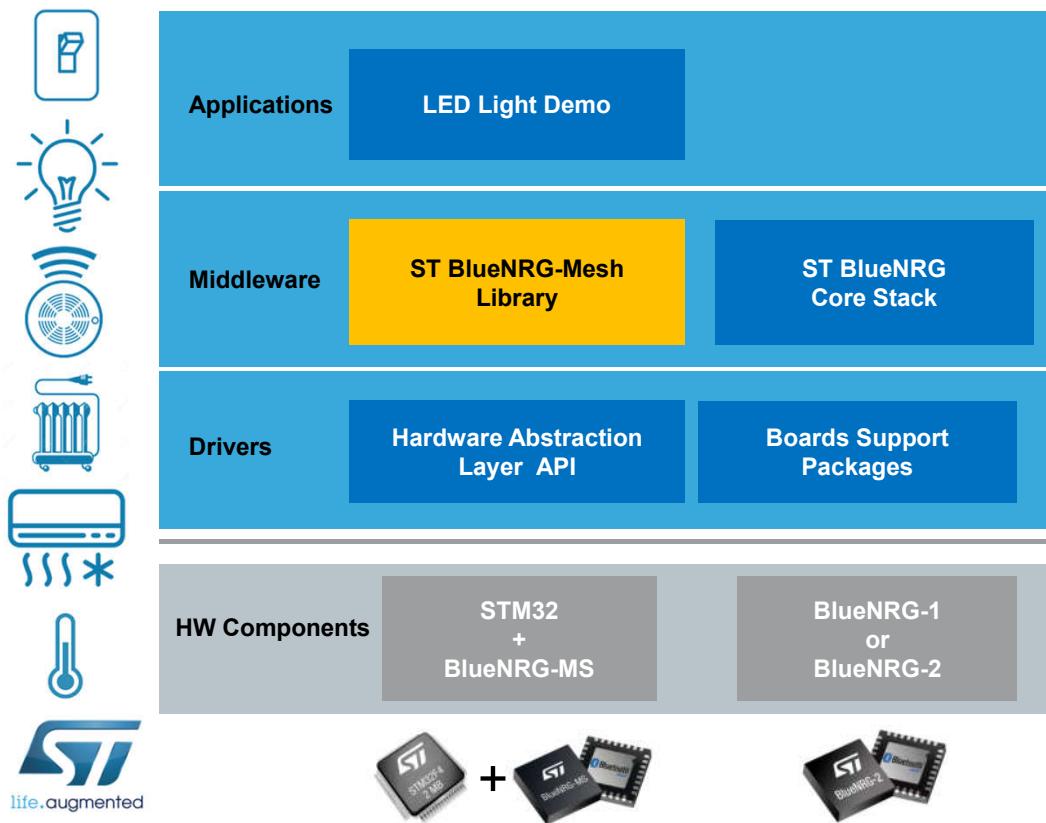


# BlueNRG-Mesh SDK

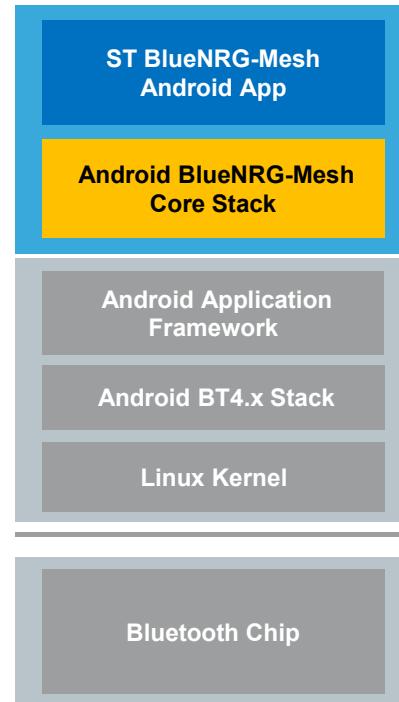
## for Embedded, Android and iOS



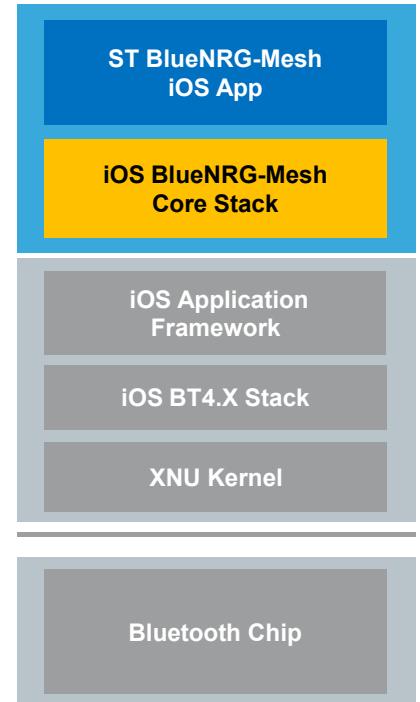
### BlueNRG-Mesh SDK



### Android SDK



### iOS SDK



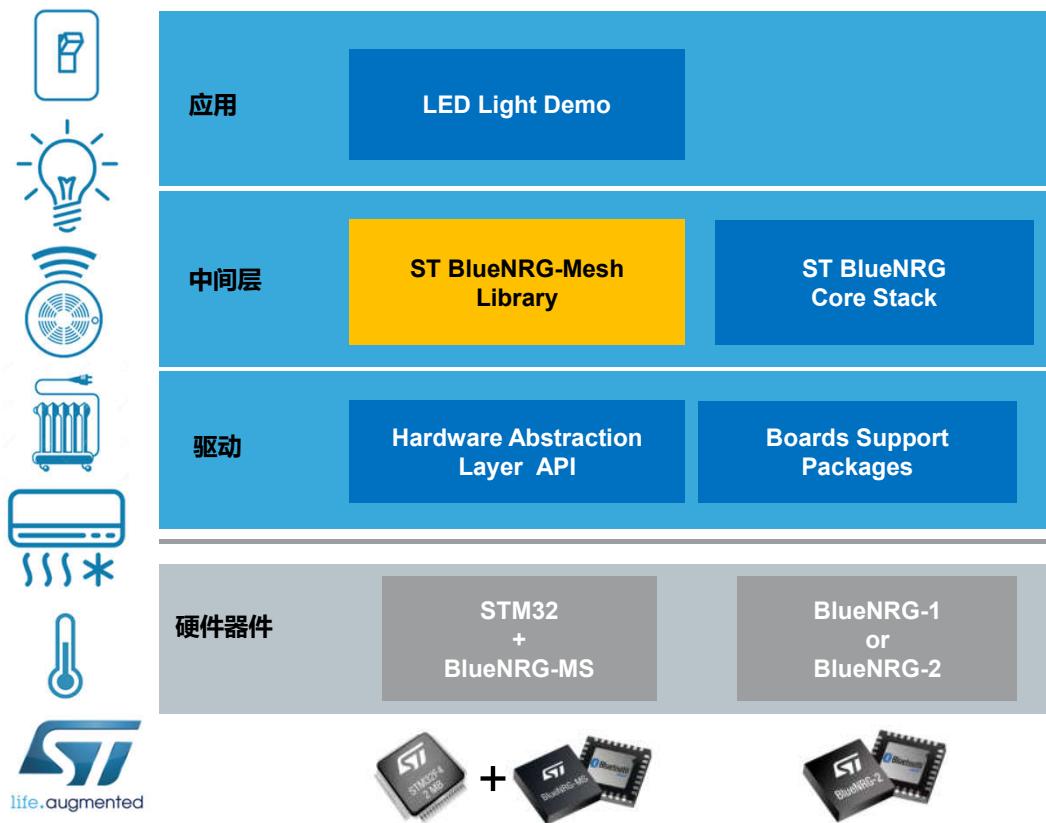
Available on SoC and network processor

# BlueNRG-Mesh SDK

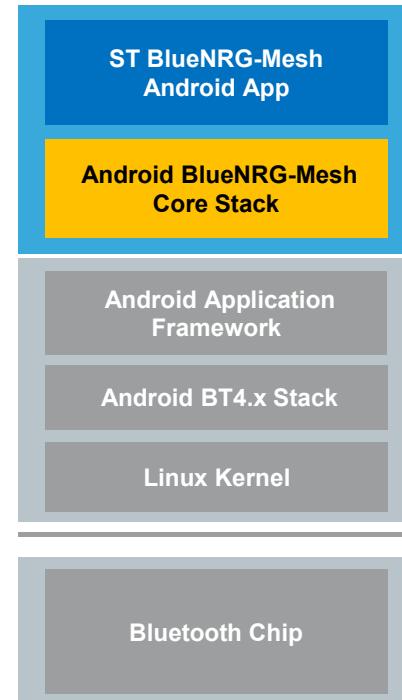
## 嵌入式, Android and iOS



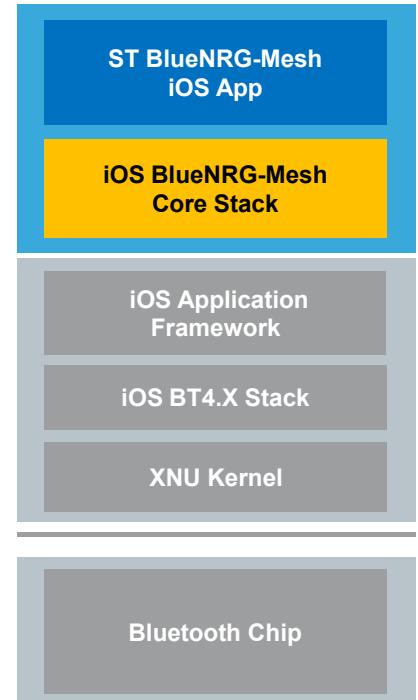
### BlueNRG-Mesh SDK



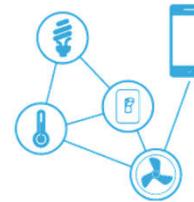
### Android SDK



### iOS SDK



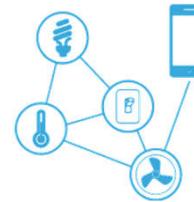
SoC和网络处理器皆可提供



# X-CUBE-BLEMESH1 @ STM32ODE

33

- Firmware
  - X-CUBE-BLEMESH1 to be published inside STM32ODE
- Documents
  - Data brief approved
  - User Manuals under update using STM32ODE templates
- Supported Platforms
  - NUCLEO-L152RE : STM32L1
  - NUCLEO-L476RG : STM32L4
  - NUCLEO-F401RE : STM32F4
  - X-NUCLEO-IDB05A1: Bluetooth Low Energy expansion board based on SPBTLE-RF module for STM32 Nucleo

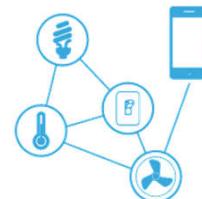


# X-CUBE-BLEMESH1 @ STM32ODE

34

- 固件
  - STM32ODE 里包含X-CUBE-BLEMESH1
- 文档
  - 数据简介
  - 更新使用STM32ODE模板的用户手册
- 支持平台
  - NUCLEO-L152RE : STM32L1
  - NUCLEO-L476RG : STM32L4
  - NUCLEO-F401RE : STM32F4
  - X-NUCLEO-IDB05A1: 基于SPBTLE-RF模块的STM32 Nucleo的BLE扩展板

# BlueNRG-Mesh On Store



 Download on the  
App Store



GET IT ON  
Google play



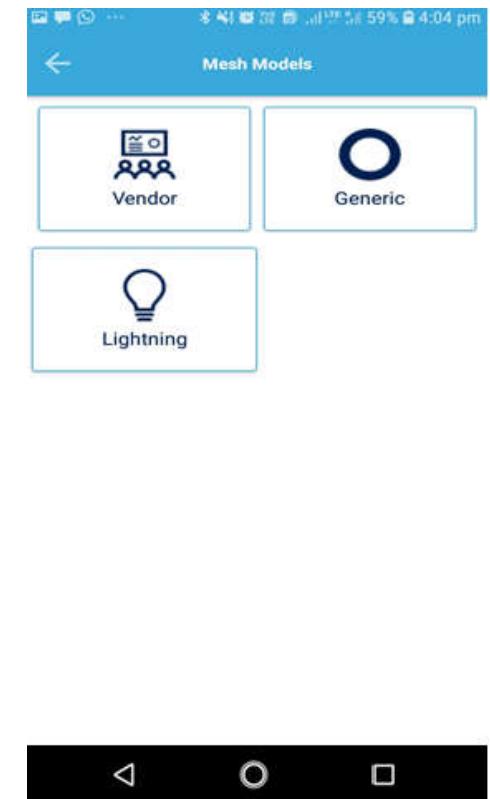
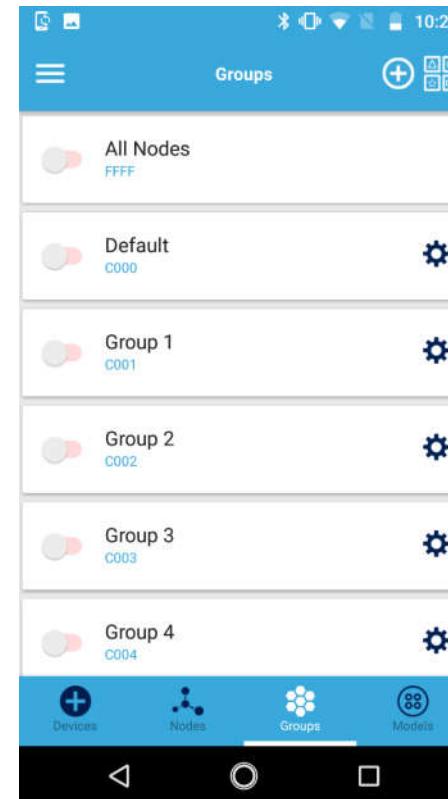
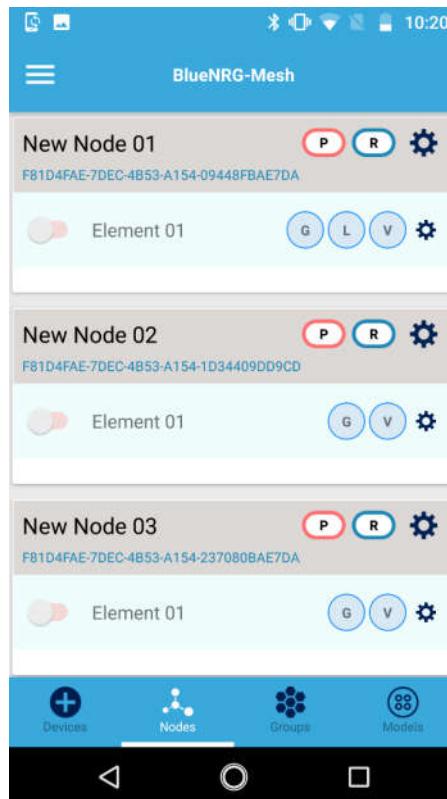
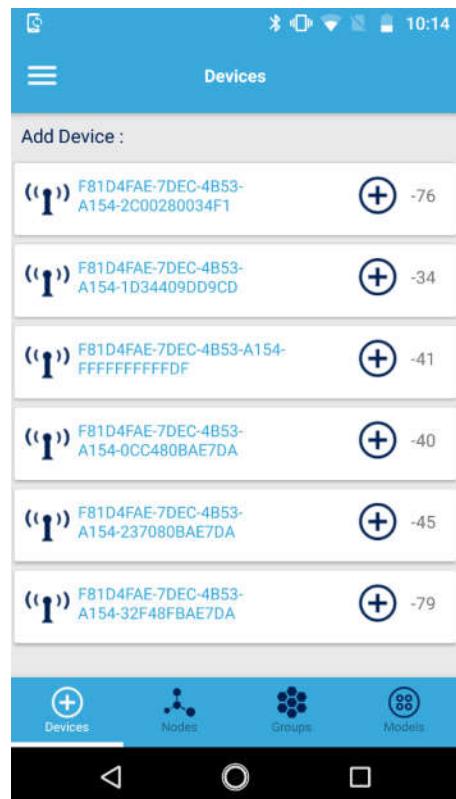
# BlueNRG-Mesh 商店



# STSW-BNRG-Mesh v1.05

37

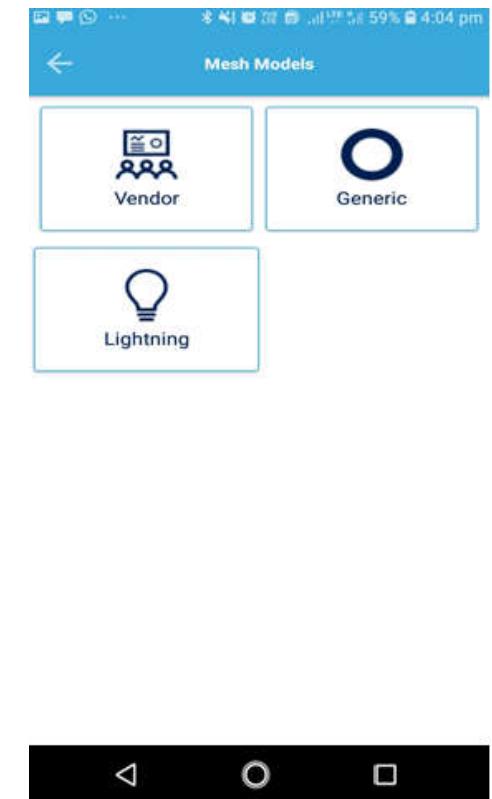
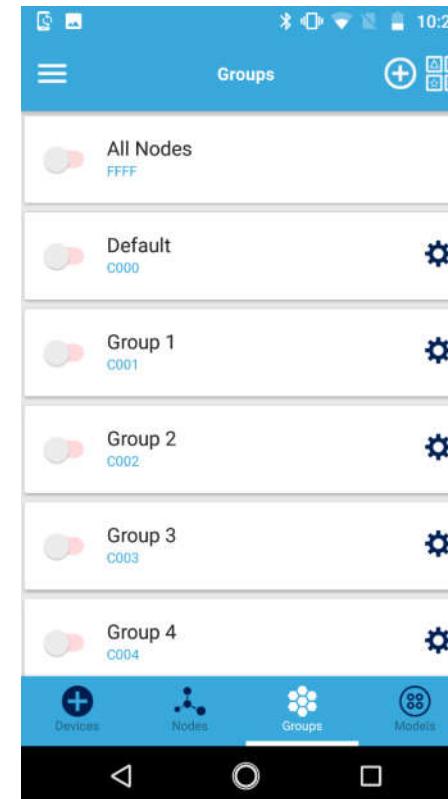
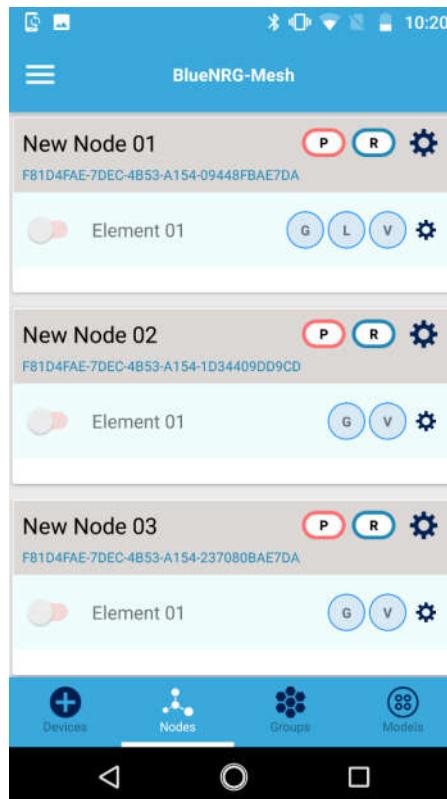
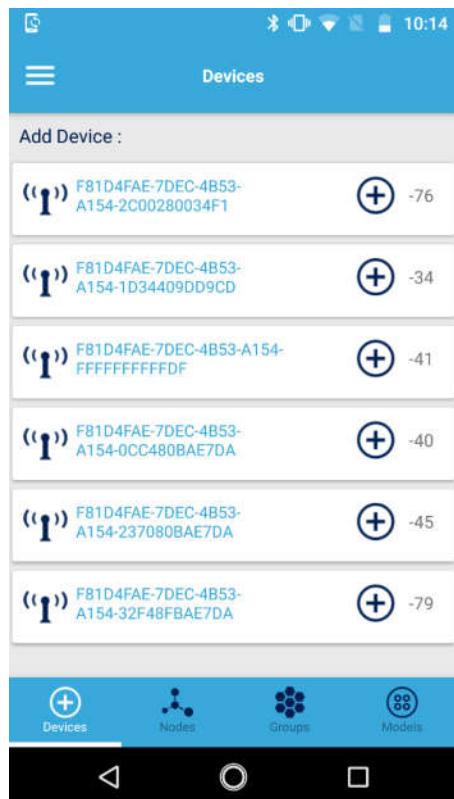
Updated App interface for the Android & iOS

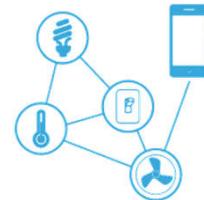


# STSW-BNRG-Mesh v1.05

38

更新App Android & iOS接口





# STSW-BNRG-Mesh Documentation

39

[UM2290 - BlueNRG Mesh Android API guide for Mesh over Bluetooth low energy](#)

[UM2417 - BlueNRG Mesh iOS API guide for Mesh over Bluetooth low energy](#)



## Start developing Bluetooth Low Energy Mesh applications on Android devices

- Network creation
- List un-provisioned nodes
- Provision a new node
- Send unicast/broadcast commands
- Save/Restore network configuration
- Register/Un-register callbacks



Android  
Studio



Xcode



[UM2180 - Getting started with the ST BlueNRG-Mesh Android application \(\\*\)](#)

[UM2361 - Getting started with the ST BlueNRG-Mesh iOS application \(\\*\)](#)

## Provision, un-provision, and control nodes in Bluetooth Low Energy Mesh network

[UM2295 - Getting started with the ST BlueNRG-Mesh embedded firmware](#)



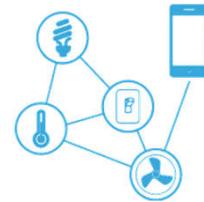
## Start developing Bluetooth Low Energy Mesh applications on available platforms

- Core functionality for secure provisioning and communication
- Smart Light sample application



(\*) Differences due to available control and libraries to design the App  
Unified User Interface available since v1.04.000

# STSW-BNRG-Mesh 文档



[UM2290 - BlueNRG Mesh Android API 基于BLE Mesh的指南](#)

[UM2417 - BlueNRG Mesh iOS API 基于BLE Mesh的指南](#)

- 开始基于Android 设备开发BLE Mesh

- 网络设备

- 列出un-provisioned 节点

- Provision 一个新节点

- 发送单点广播/广播 命令

- 保存/存储 网络配置

[UM2180 – 开始 ST BlueNRG-Mesh Android 应用之旅 \(\\*\)](#)

[UM2361 – 开始ST BlueNRG-Mesh iOS 应用之旅 \(\\*\)](#)

Provision, un-provision, 和在BLE Mesh网络下控制节点

[UM2295 – 开始ST BlueNRG-Mesh 嵌入式固件开发](#)



开始可用平台下的BLE Mesh开发

- 安全provisioning和通信的核心功能
- 智慧灯控应用

(\*) Differences due to available control and libraries to design the App  
Unified User Interface available since v1.04.000



# Bluetooth SIG Certification

- Mandatory Features: 177 Tests
  - Mesh v1.0 Compliance
  - Role: Node
  - Bearer: Advertising, GATT
  - Provisioning: PB-GATT
  - Provisioning Protocol: Provisioning Server
  - Provisioning: Public Key No OOB
  - Network layer: All
  - Foundation Mesh Models: Configuration Server, Health Server
  - Node features: Proxy, Relay
- Optional Feature: 38 tests
  - Friendship
  - Low Power Node

Bluetooth Qualification  
Certified

BlueNRG-Mesh QDID  
: 116029

<https://launchstudio.bluetooth.com/ListingDetails/65504>

# Bluetooth SIG 认证

- 强制特性: 177 测试项
  - Mesh v1.0 兼容
  - 角色: 节点
  - 承载: 广播, GATT
  - Provisioning: PB-GATT
  - Provisioning 协议: Provisioning 服务
  - Provisioning: 非 OOB 公钥
  - 网络层: 所有
  - 模型基础: 配置服务, 健康服务
  - 节点特性: 网关, 续传

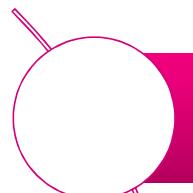
- 可选特性: 38 测试项
  - 朋友
  - 低功耗特性



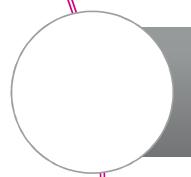
蓝牙认证  
**BlueNRG-Mesh QDID**  
**: 116029**

<https://launchstudio.bluetooth.com/ListingDetails/65504>

# BlueNRG-Mesh



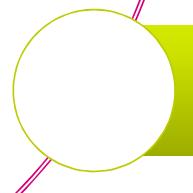
Bluetooth Mesh Introduction



BlueNRG-Mesh Solution



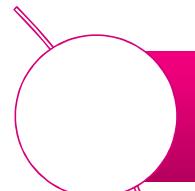
Getting started with BlueNRG-Mesh



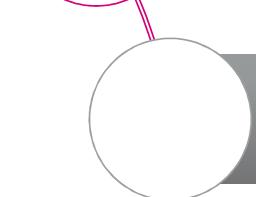
Basics and Hands on Session



# BlueNRG-Mesh



Bluetooth Mesh 介绍



BlueNRG-Mesh 解决方案



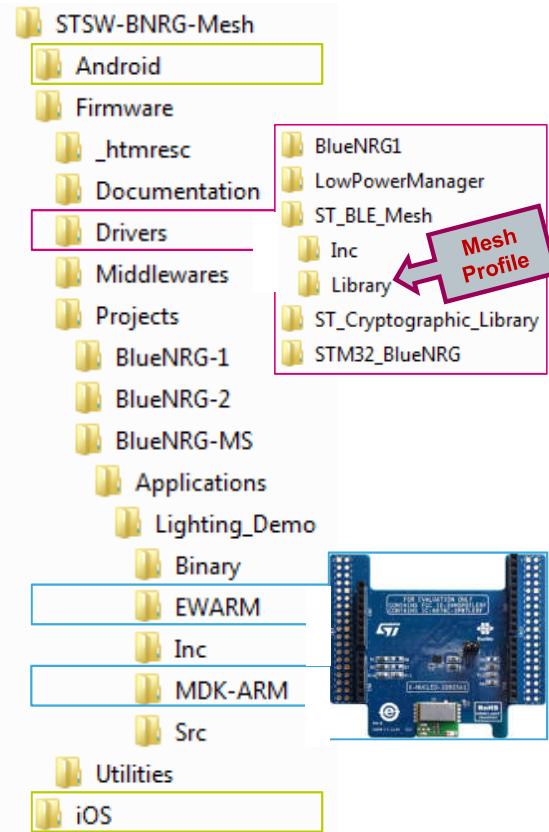
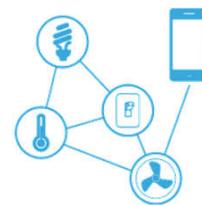
开启BlueNRG-Mesh之旅



基础和动手实践环节

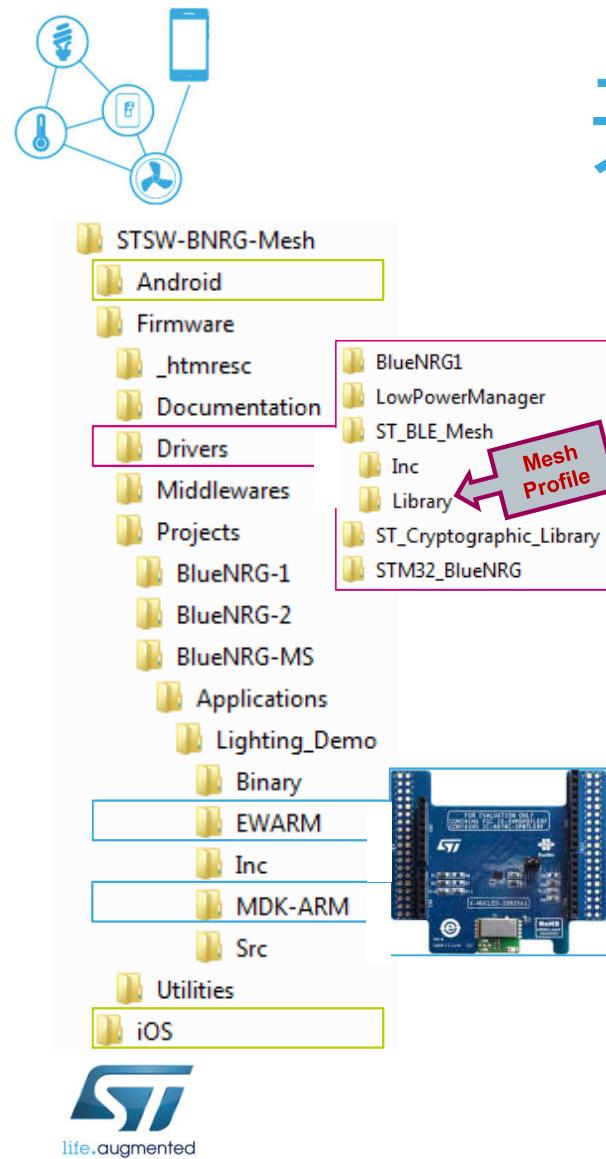


# STSW-BNRG-Mesh Getting Started



1. Download STSW-BNRG-Mesh software package  
[>>> www.st.com/blemesh <<<](http://www.st.com/blemesh)
2. Unzip & Install it
3. Open sample project in STSW-BNRG-Mesh folder
  - APP Development on selected device
    - iOS 
    - Android 
  - FW Development on available platforms
    - IAR / Keil Toolchains 
4. Compile & Flash it
5. Enjoy the Mesh!

# 开启STSW-BNRG-Mesh之旅



## 1. 下载 STSW-BNRG-Mesh 软件包

>>> [<<<](http://www.st.com/blemesh)

## 2. 解压, 安装

## 3. 打开STSW-BNRG-Mesh文件夹下的项目

- APP 开发工具选择
  - Android Studio / Xcode IDEs
  
- 固件爱你开发平台工具选择
  - IAR / Keil 工具链

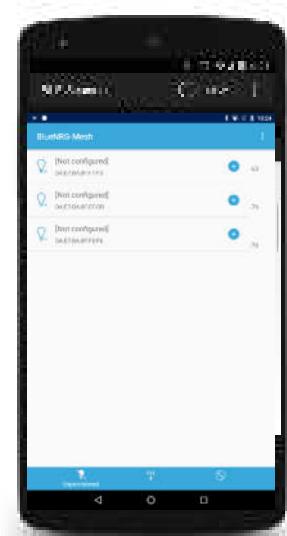


## 4. 编译, 烧录

## 5. 享受 Mesh !

# Setup & Demo Example

Smartphone

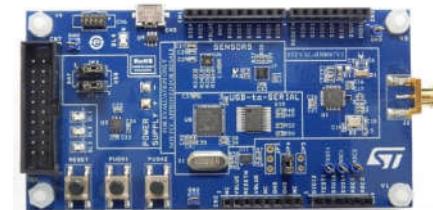


App for Demo



BlueNRG-Mesh Android , iOS

BLE Hardware



STEVAL-IDB007V1: evaluation platform  
based on the BlueNRG-1

STEVAL-IDB008V1, STEVAL-IDB008V2:  
evaluation platform  
based on the BlueNRG-2

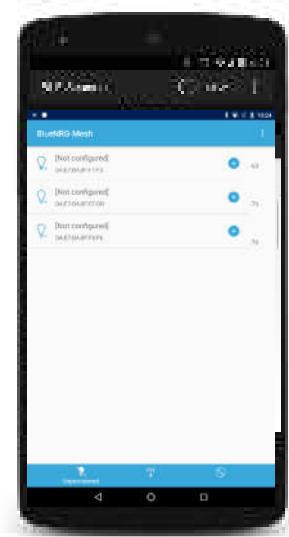


X-NUCLEO-IDB05A1 + NUCLEO-L152RE

## Prerequisites

# 安装 和演示

智能手机



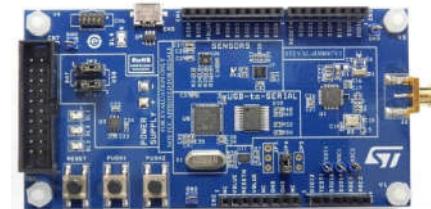
预备

演示的App



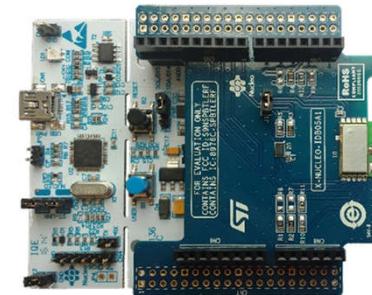
BlueNRG-Mesh Android , iOS

BLE 硬件



STEVAL-IDB007V1: 基于BlueNRG-1 的开发平台

STEVAL-IDB008V1, STEVAL-IDB008V2:  
基于BlueNRG-2 的开发平台



X-NUCLEO-IDB05A1 + NUCLEO-L152RE

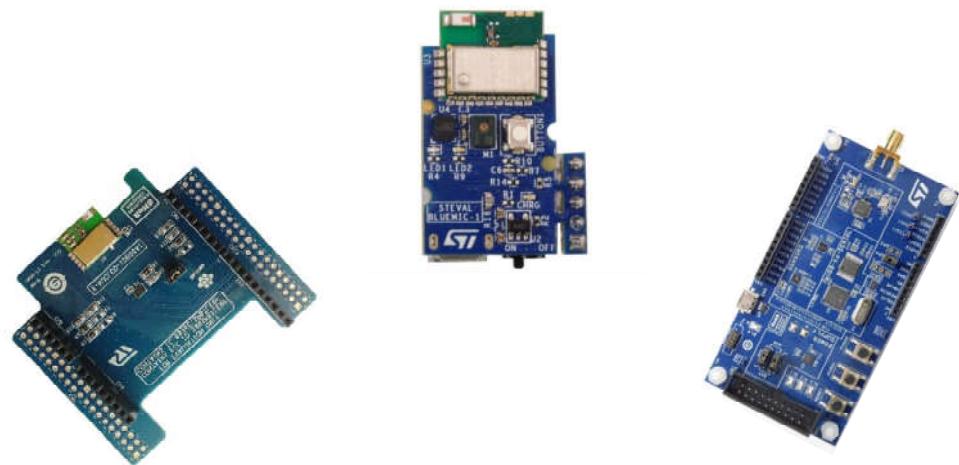
# Lighting Demo Requirements

- **Hardware Requirement**

- [NUCLEO-L152RE + X-NUCLEO-IDB05A1](#) (firmware 7.2C)
- [STEVAL-IDB007V1 Board with BlueNRG-1 SoC](#)
- [STEVAL-IDB008V1 Board with BlueNRG-2 SoC](#)
- USB type A to Mini-B USB cable to connect the Nucleo board to the PC
- USB type A to Micro-B USB cable to connect the IDB007V1 board to the PC
- (Software on demand) [STEVAL-BLUEMIC1 Board](#), small form-factor and battery powered
- (Software on demand) [STEVAL-IDB007V1M Board](#) with SPBLTE-1S

- **Software Requirement**

- [USB driver](#) for Android smartphones
- [Android SDK Platform Tools\(adb\)](#)
- [ST-Link Utility](#) to install ST-Link USB driver
- [IDB007V1 BlueNRG-1 Flasher](#)
- [STSW-BNRGUI](#) PC tool to upgrade BlueNRG-MS firmware



# 演示预备

50

## • 硬件预备

- NUCLEO-L152RE + X-NUCLEO-IDB05A1 (固件7.2C)
- STEVAL-IDB007V1 Board with BlueNRG-1 SoC
- STEVAL-IDB008V1 Board with BlueNRG-2 SoC
- 用于连接Nucleo板到PC的USB type A 转 Mini-B USB 线
- 用于连接IDB007V1 板到PC的USB type A 转 Micro-B USB 线
- (软件需求) STEVAL-BLUEMIC1 Board, 供电小板
- (软件需求) STEVAL-IDB007V1M Board w带有SPBLTE-1S模块

## • 软件预备

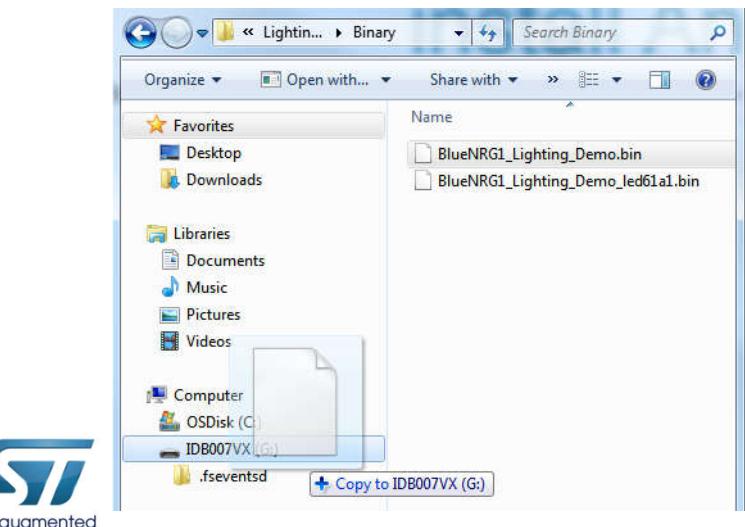
- Android 智能手机的USB 驱动
- Android SDK 平台工具(adb)
- ST-Link Utility 安装ST-Link USB 驱动
- IDB007V1 BlueNRG-1 Flasher
- STSW-BNRGUI PC 工具升级固件



# Flashing the board & Installing Android application

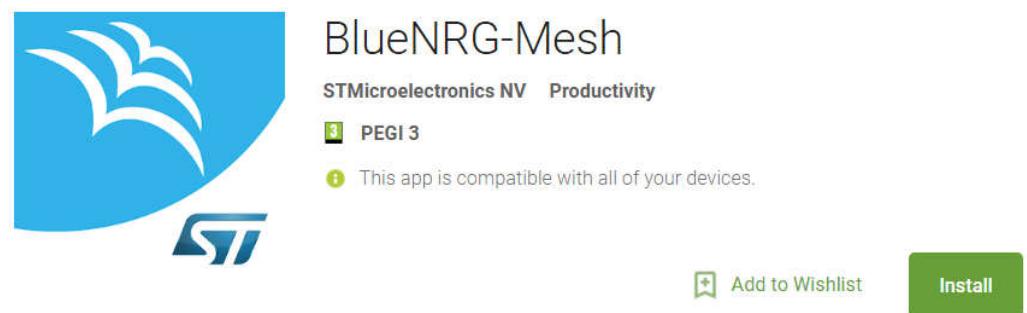
## Embedded

- Once the package is installed, Go to BlueNRG-Mesh\Embedded\Projects\BlueNRG-1\Applications\Lighting\_Demo\Binary
- Drag-and-drop the BlueNRG1\_Lighting\_Demo.bin in the mass storage named IDB007VX or BlueNRG-1



## Android

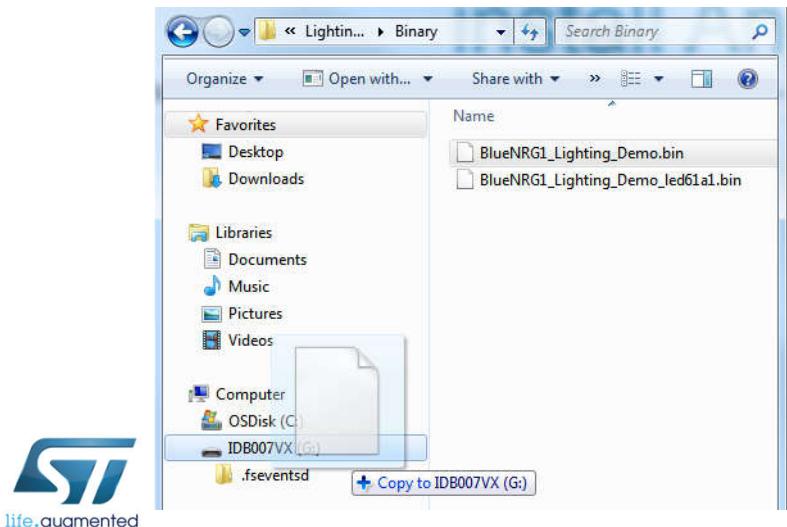
- First option: Once the package installed, Go to BlueNRG-Mesh\Android\apk, Android apk file is available
- Second option: go on the Play Store and search for BlueNRG-Mesh. Install BlueNRG-Mesh Android app
- Make sure the Bluetooth and positioning features are activated



# 烧录单板和安装 Android 应用

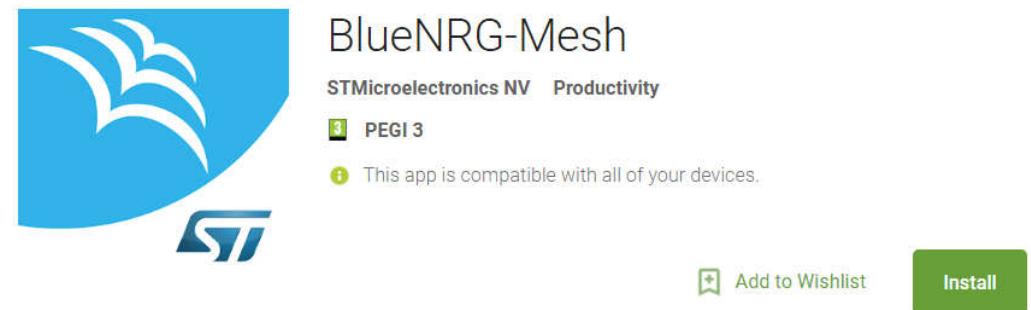
## 嵌入式系统

- 一旦安装完成开发包后，请去路径BlueNRG-Mesh\Embedded\Projects\BlueNRG-1\Applications\Lighting\_Demo\Binary
- 把BlueNRG1\_Lighting\_Demo.bin 文件拖到名为IDB007VX 或BlueNRG-1大容量存储设备中

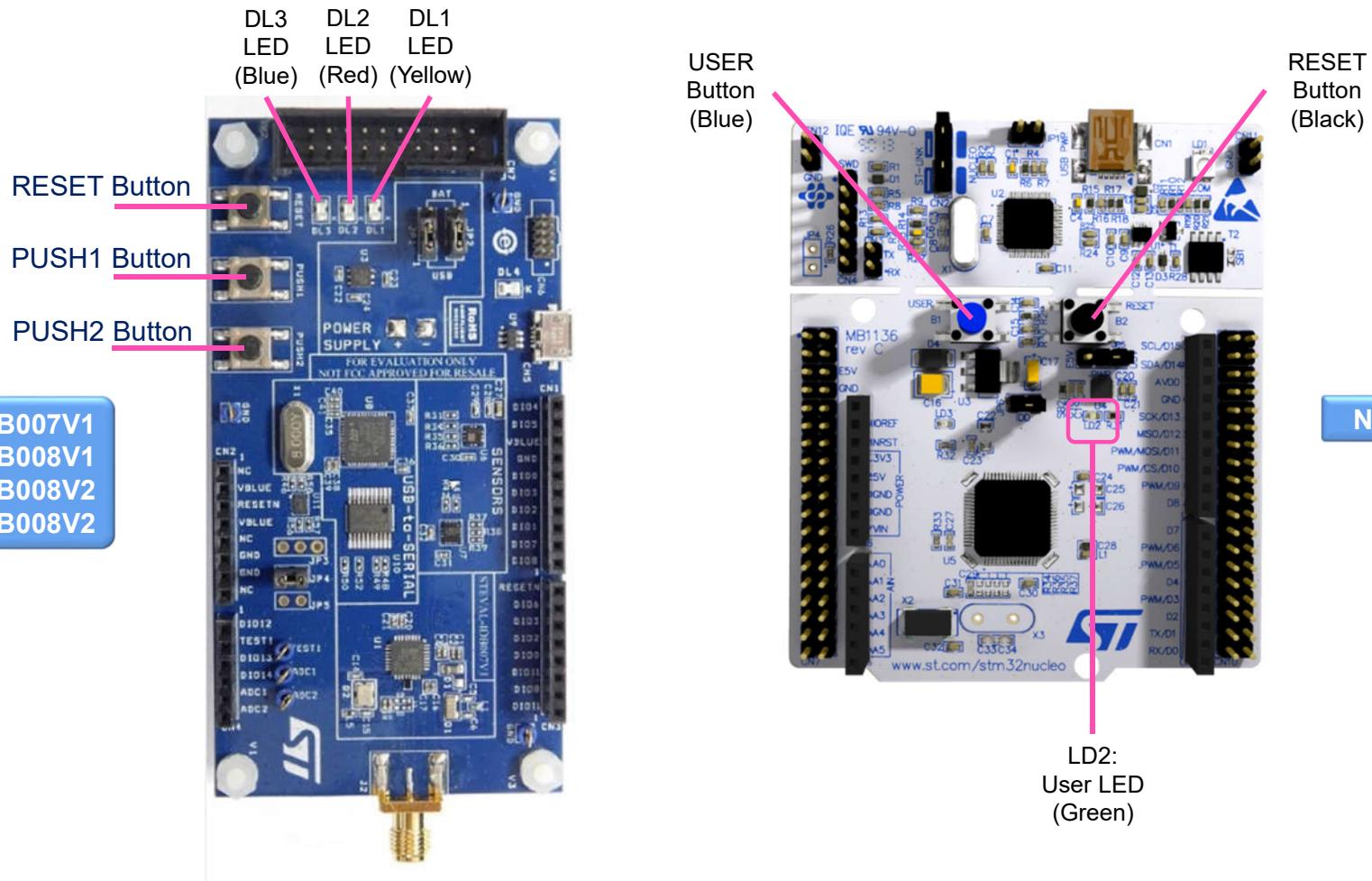


## Android

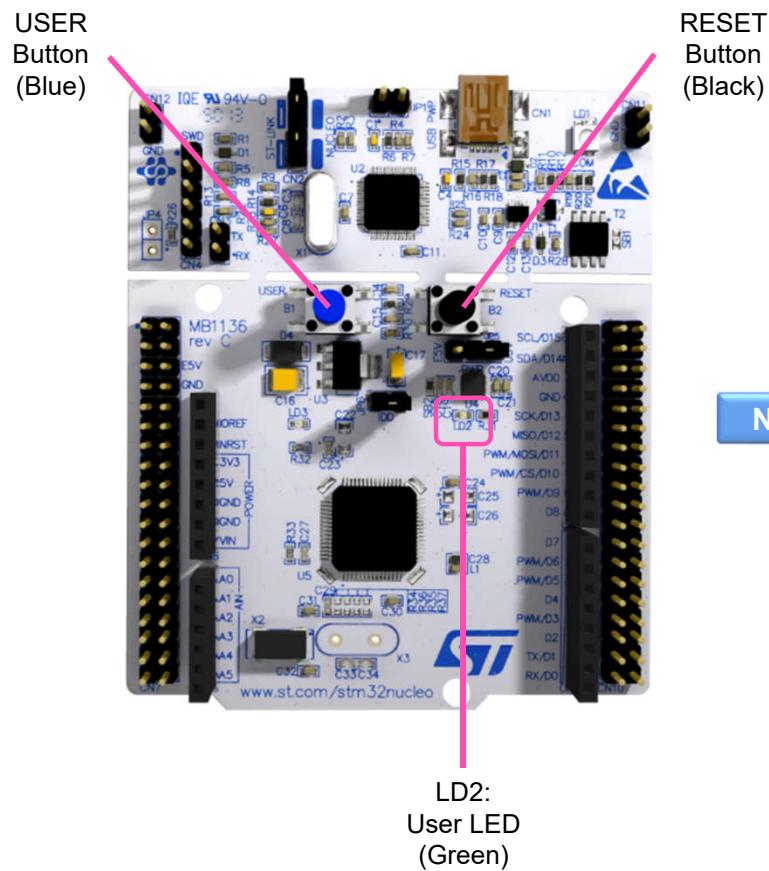
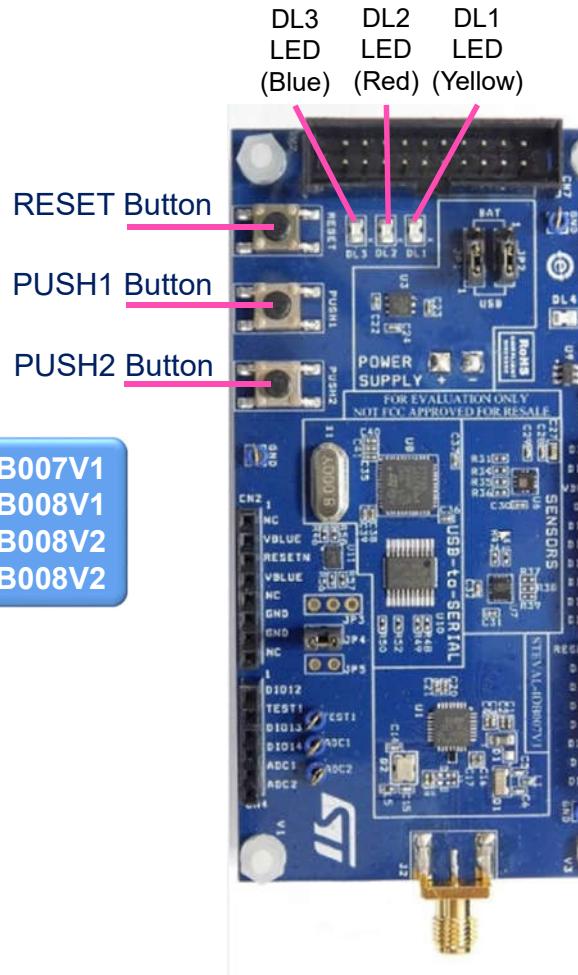
- 首先选项:一旦安装完成开发包后，请去路径BlueNRG-Mesh\Android\apk, Android apk 文件就在这里
- 然后: 去Play Store 并且搜索BlueNRG-Mesh. 安装BlueNRG-Mesh Android app
- 请确保蓝牙和定位功能是激活的



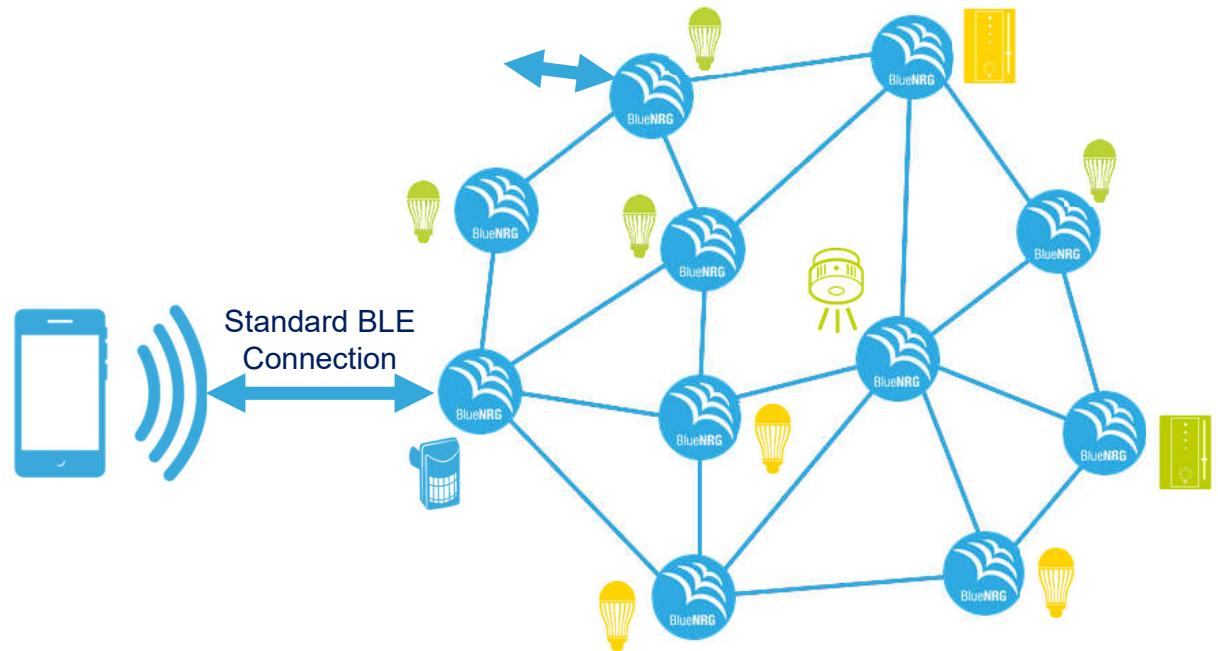
# User interface quick reference



# 快速参考用户接口

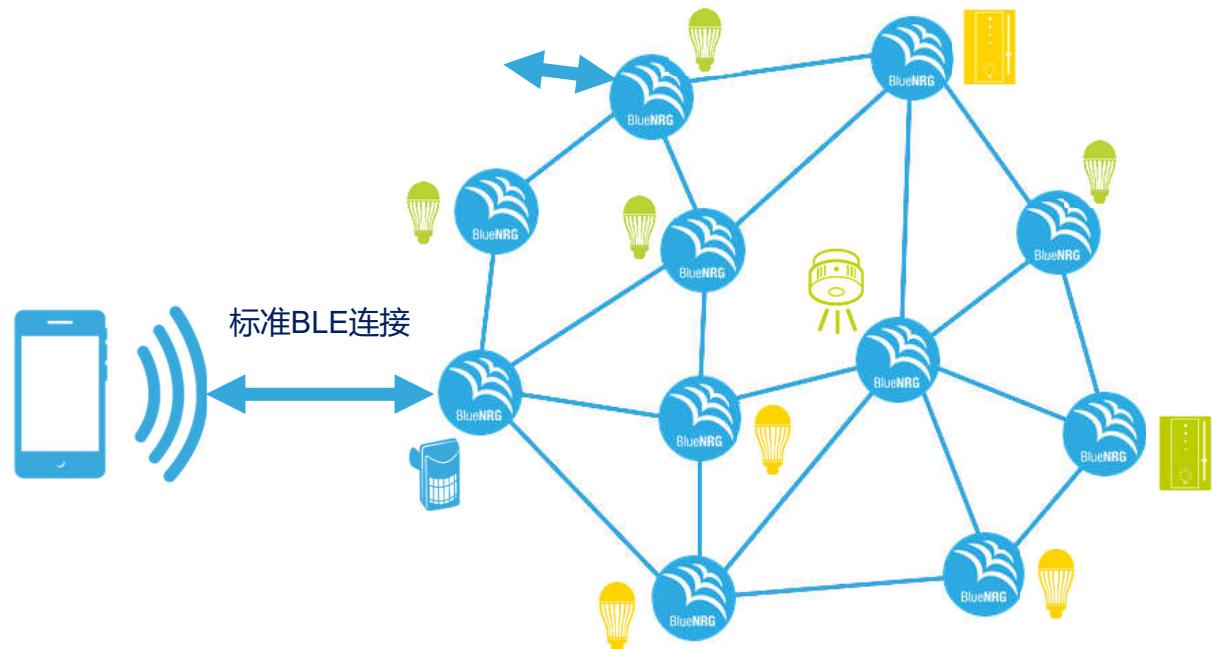


# Lighting Application Demonstration

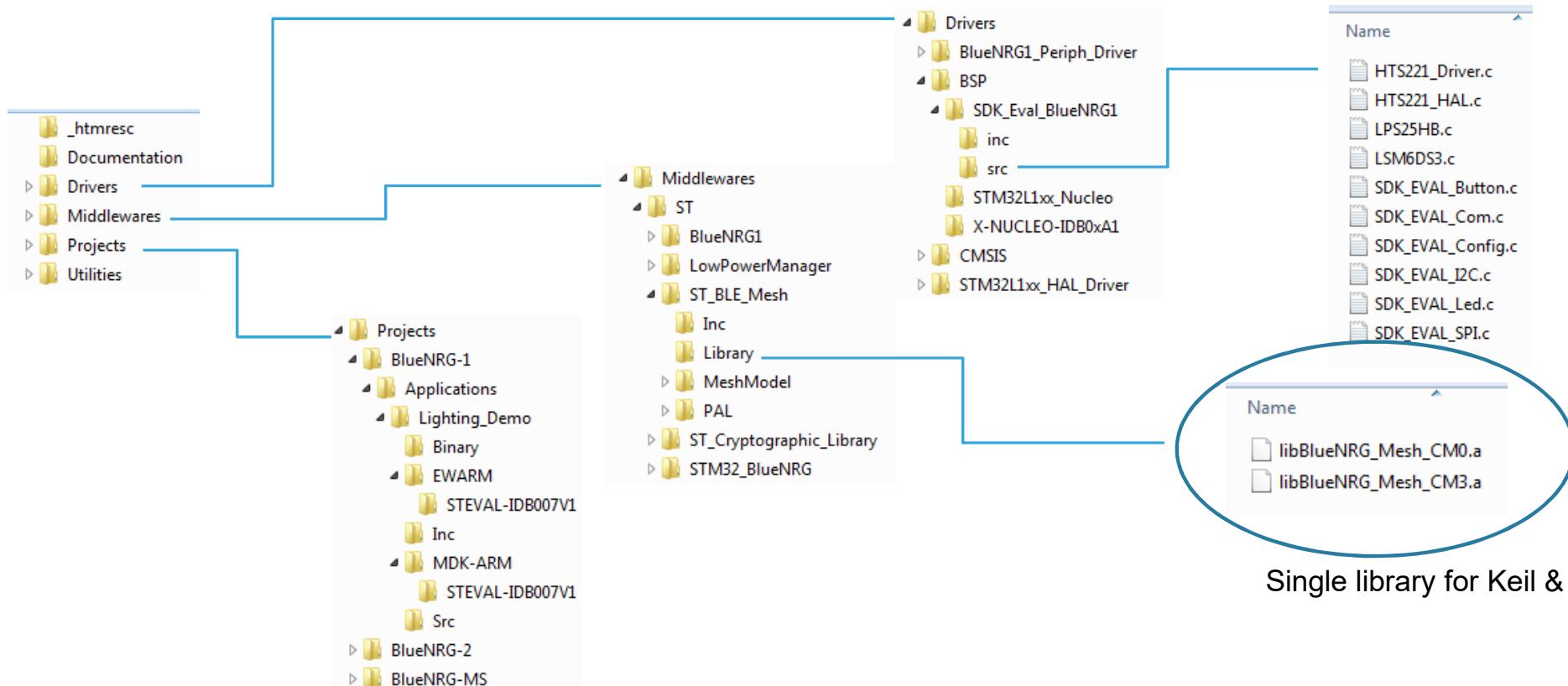
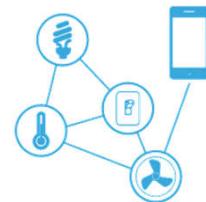


# 灯控演示

56



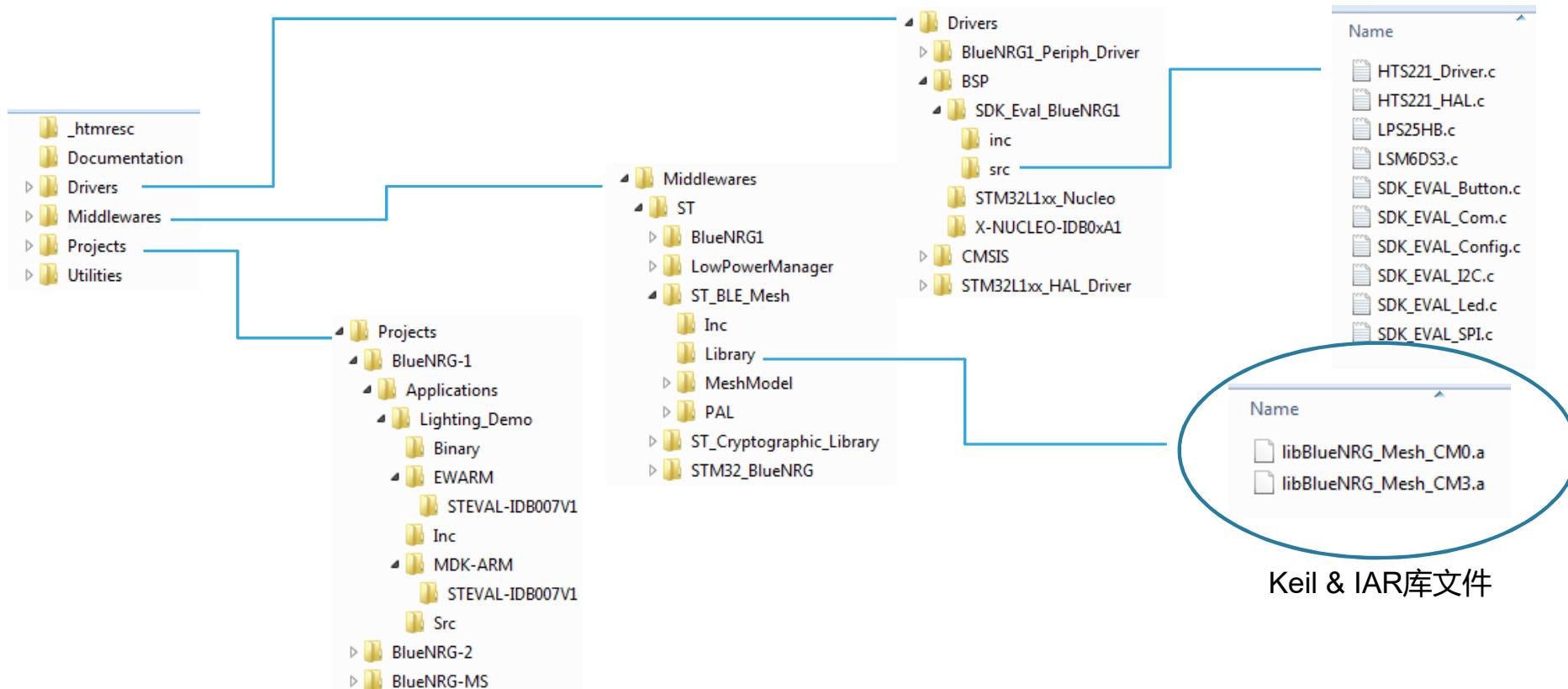
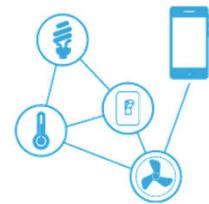
# BlueNRG-Mesh Firmware SDK



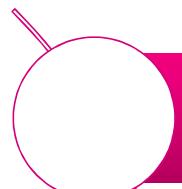
Name  
 libBlueNRG\_Mesh\_CM0.a  
 libBlueNRG\_Mesh\_CM3.a

Single library for Keil & IAR

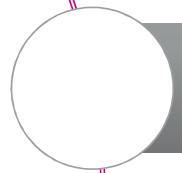
# BlueNRG-Mesh固件SDK



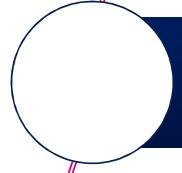
# BlueNRG-Mesh



Bluetooth Mesh Introduction



BlueNRG-Mesh Solution



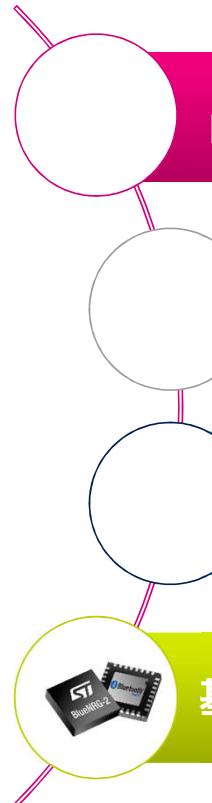
Getting started with BlueNRG-Mesh



Basics and Hands on Session



# BlueNRG-Mesh



Bluetooth Mesh 介绍

BlueNRG-Mesh 解决方案

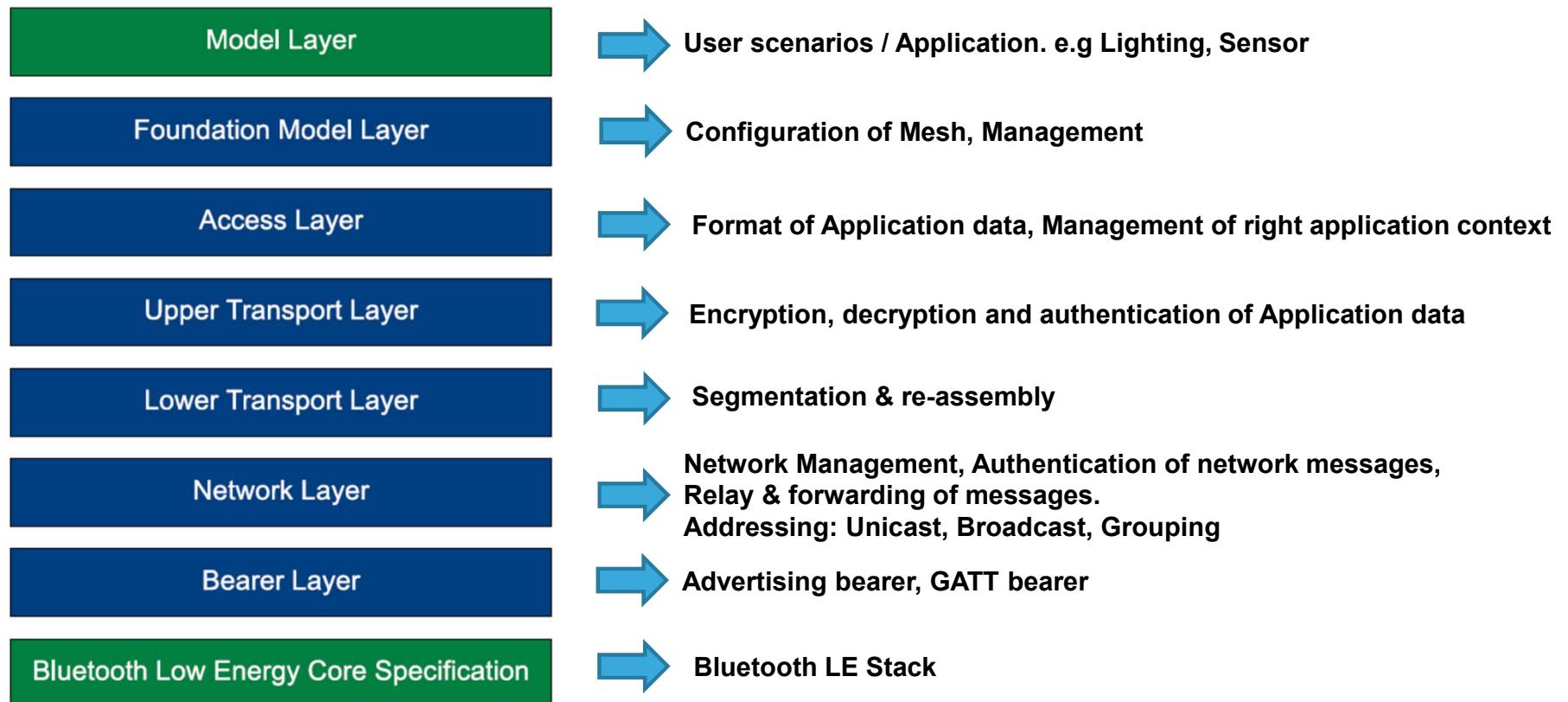
开启BlueNRG-Mesh之旅



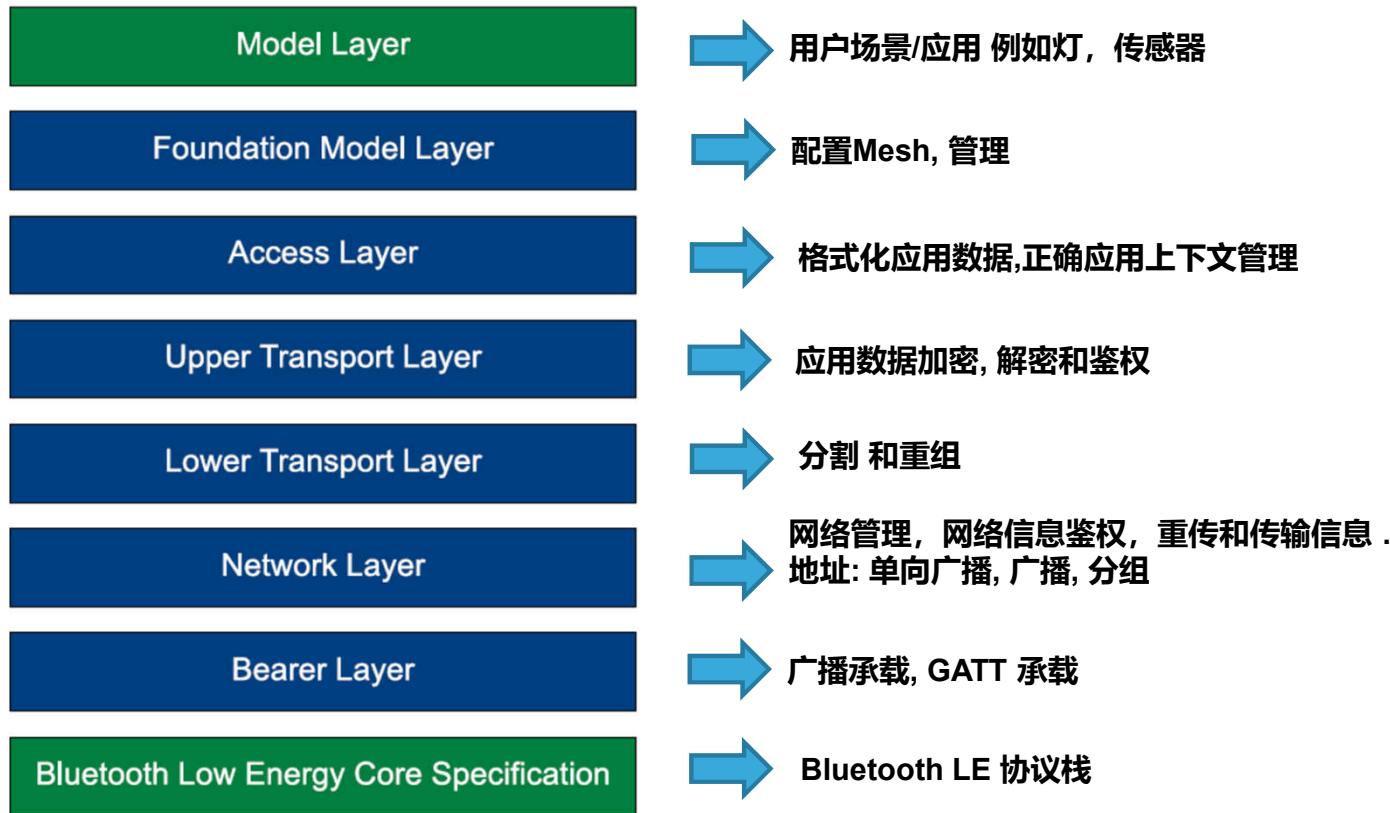
基础和动手实践环节



# BLE Mesh system architecture



# BLE Mesh系统架构



# Security, Attacks and Protection

- Devices are added to the mesh network to make them mesh nodes is a secure process
- All mesh messages are encrypted and authenticated
- Mesh security protects the network against replay attacks.

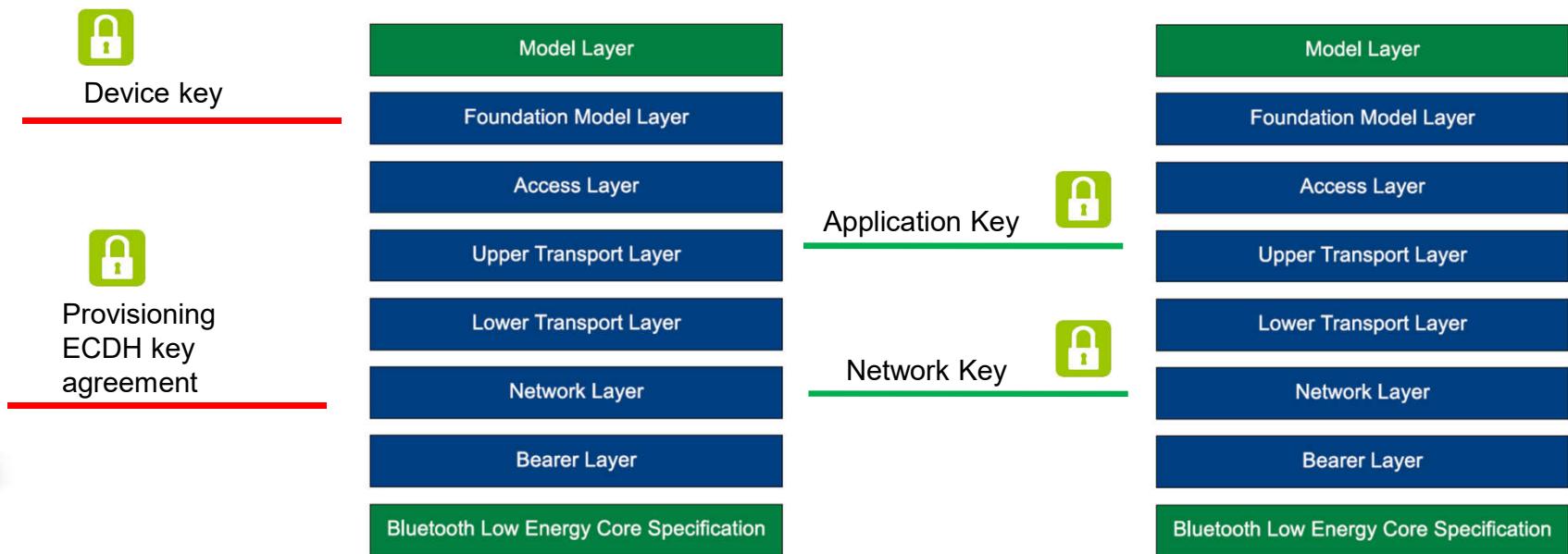
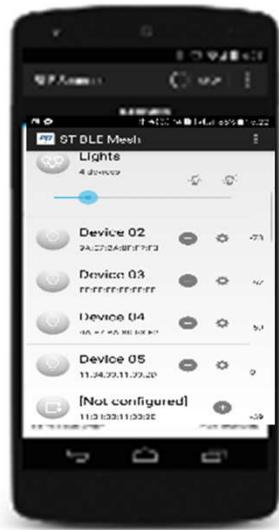


# 安全，攻击和保护

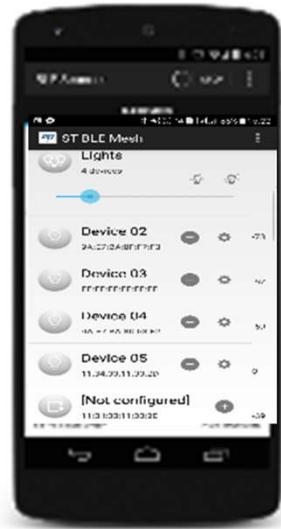
- 设备被添加到mesh网络并保证这些mesh节点的添加是一个安全的过程
- 所有的mesh消息是被加密和认证的
- Mesh安全保护网络不受重放攻击



# Bluetooth Mesh Keys



# 蓝牙mesh密钥



设备密钥



配置ECDH密钥协商



应用密钥



网络密钥



# Type of Keys

67

- Following Type of Keys used

- Network Key : Provided by Provisioner
  - Same network key in a single network
  - Single node can have multiple Network keys if it is part of several subnets
  - Smart-phone can be member of multiple networks (Home, Office, Car-garage...)
- Application Key : Provided by Provisioner
  - Application specific keys
  - Lights and bell are different applications and may have different keys

- Device Key : Never transmitted over air
  - Calculated using the shared secret derived from the ECDH (Elliptic Curve Diffie-Hellman) key agreement between the provisioner and the device being provisioned
  - No other nodes come to know about it

# 密钥类型

68

- 常用密钥类型如下

- 网络密钥：由启动配置设备提供
  - 在单一网络里有相同网络密钥
  - 如果单一节点是子网的一部分，那么它能够有多重网络密钥
  - 智能手机可以成为多重网络的一员（家庭，办公室，停车场……）
- 应用密钥：由启动配置设备提供
  - 应用特定密钥
  - 灯和钟是不同的应用，可能有不同的密钥

- 设备密钥：永不从空中传输

- 使用由启动配置设备和被配置设备之间的ECDH（椭圆曲线密钥交换算法）密钥协商产生的共享安全信息计算得出
- 没有其他节点会知道它

## SEQ

- Sequence Number (SEQ)
  - Messages have 24-bit Sequence Number
  - Must be unique for every message sent
  - Exhaustion (@ 2 Hz) after 97 days

## IV Index

- IVI (Initialization Vector Index)
  - Each node has a 32-bit IV Index
  - Incremented only when sequence number is about to exhaust
  - Only least significant bit of IV Index is included in every mesh message
  - Exhaustion (@ 1 Hz) after 2 billion years

## SEQ

- 序列码 (SEQ)
  - 消息具有24比特序列码
  - 对于每条发出的消息都必须是唯一的
  - 耗尽 (@ 2 Hz) 97 天之后

## IV Index

- IVI (初始向量索引)
  - 每一个节点具有32比特初始向量索引
  - 只有当序列码大约耗尽时才会增加
  - 只有IVI的最低有效位是包含在所有 mesh消息中的
  - 耗尽 (@ 1 Hz) 20亿年之后

# Network Packet Protection

71



Nonce for Security

Authenticated

Encrypted

Obfuscated



# 网络数据包保护

72



临时保护

认证

加密

模糊



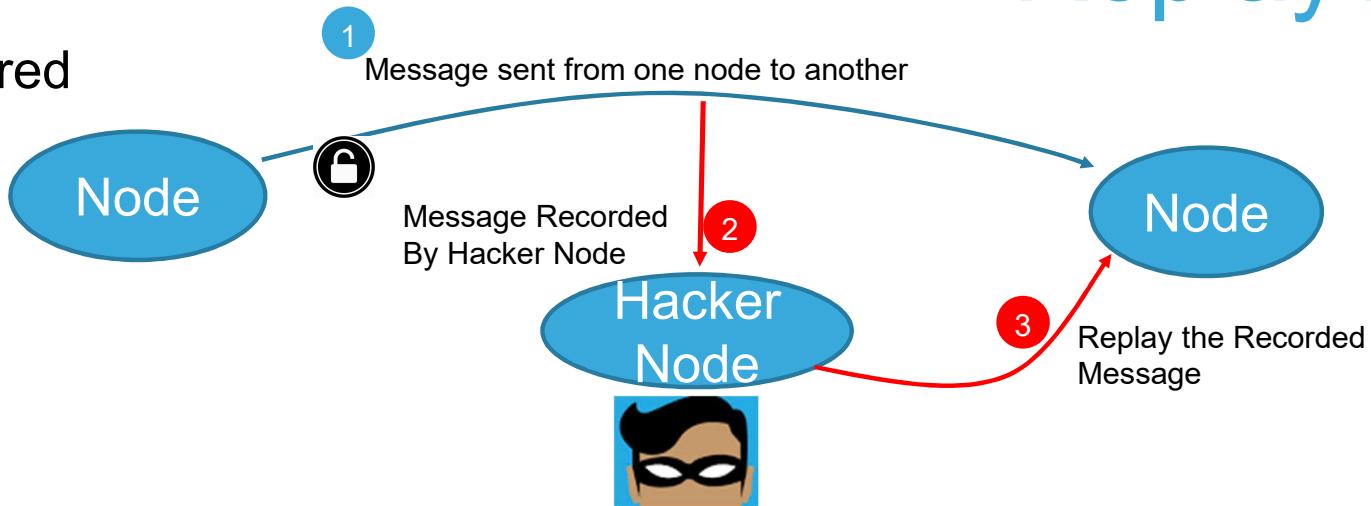
模糊

加密

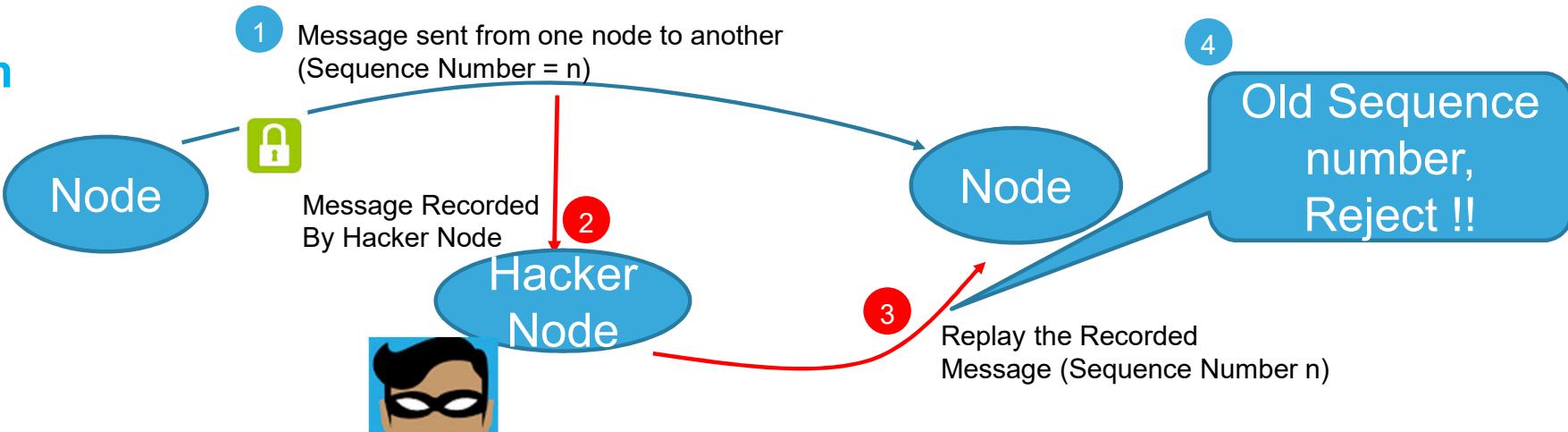
# Replay Attack

73

# Un-secured Network



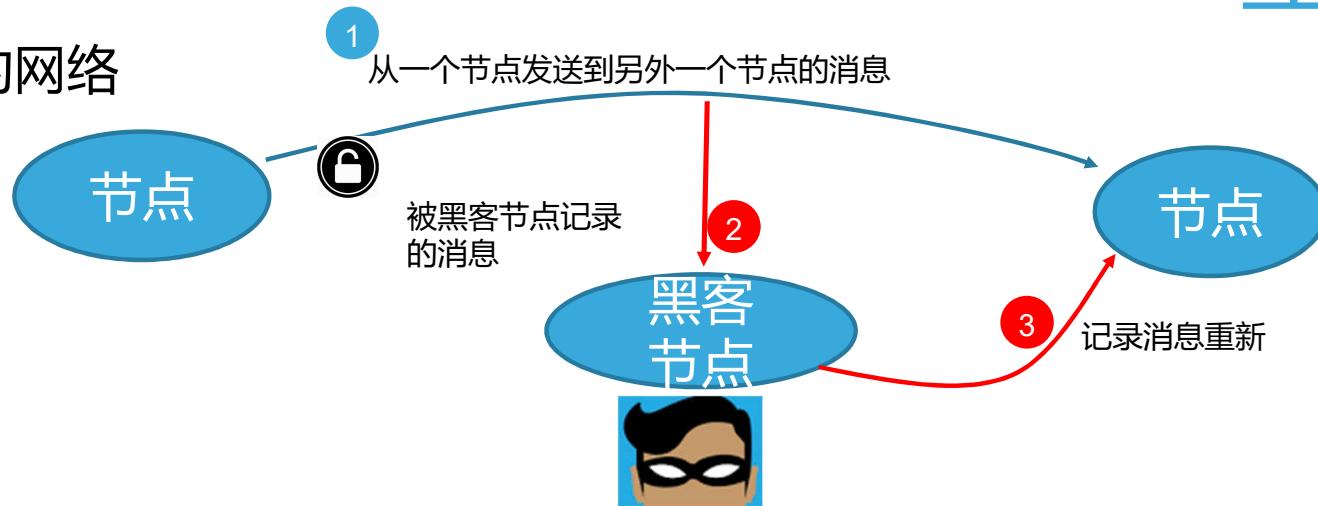
# Bluetooth Mesh



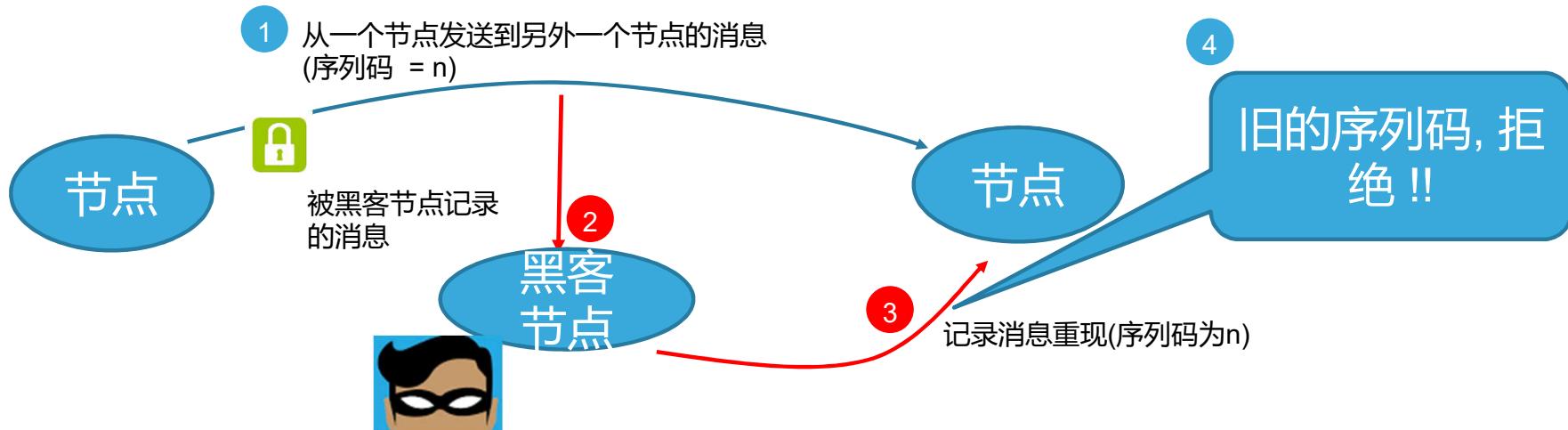
# 重现攻击

74

## 不安全的网络



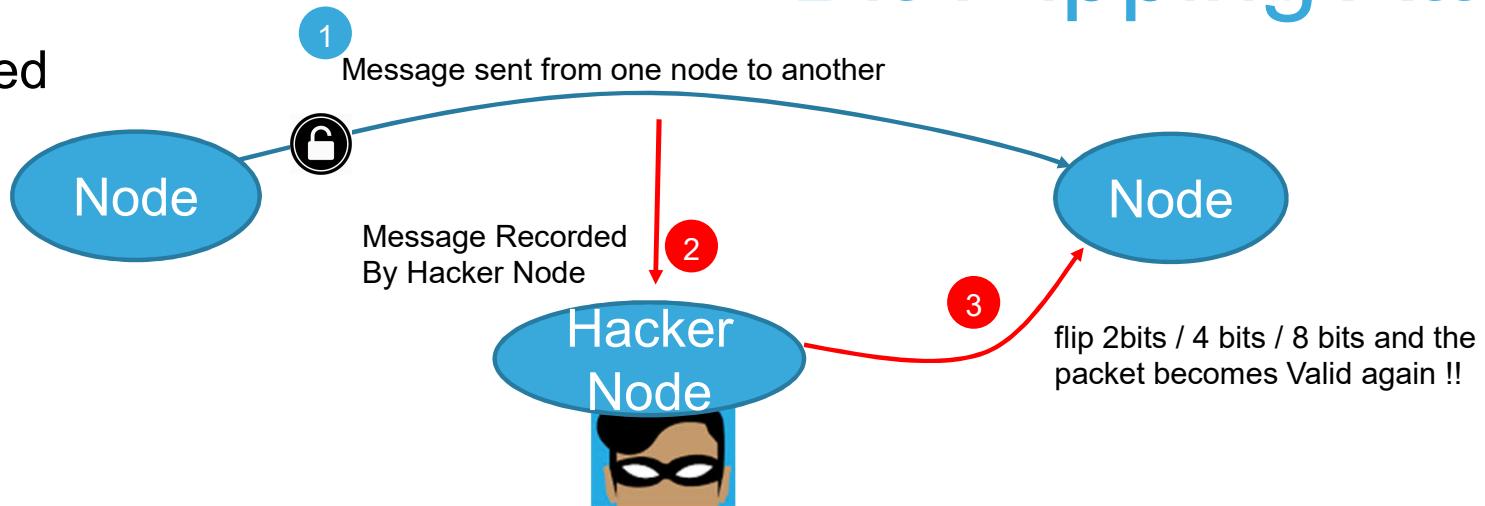
## 蓝牙 Mesh



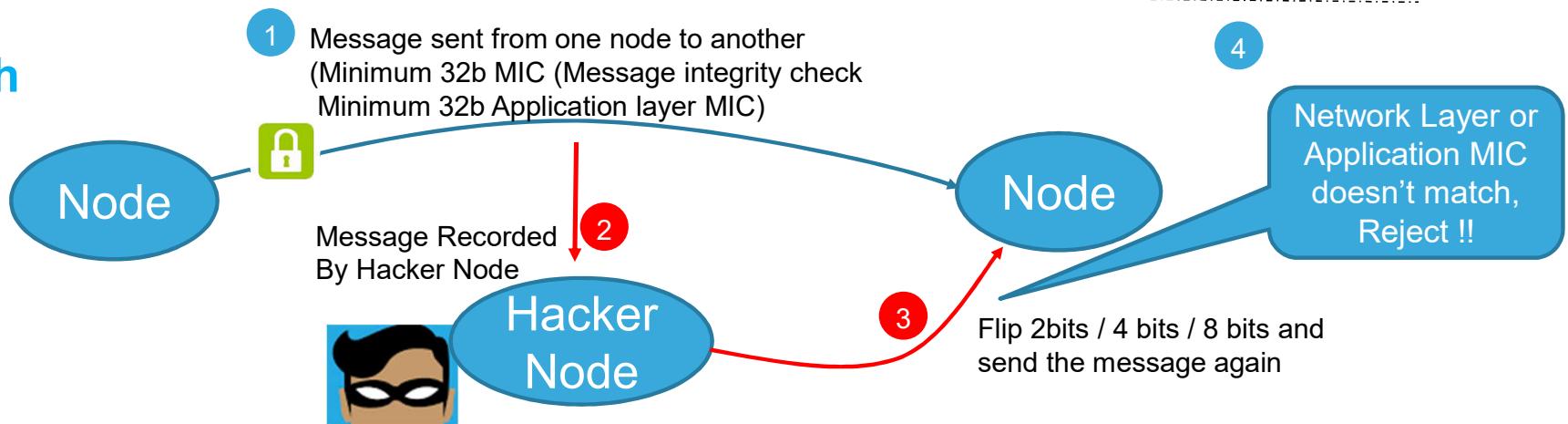
# Bit-Flipping Attack

75

Un-secured  
Network



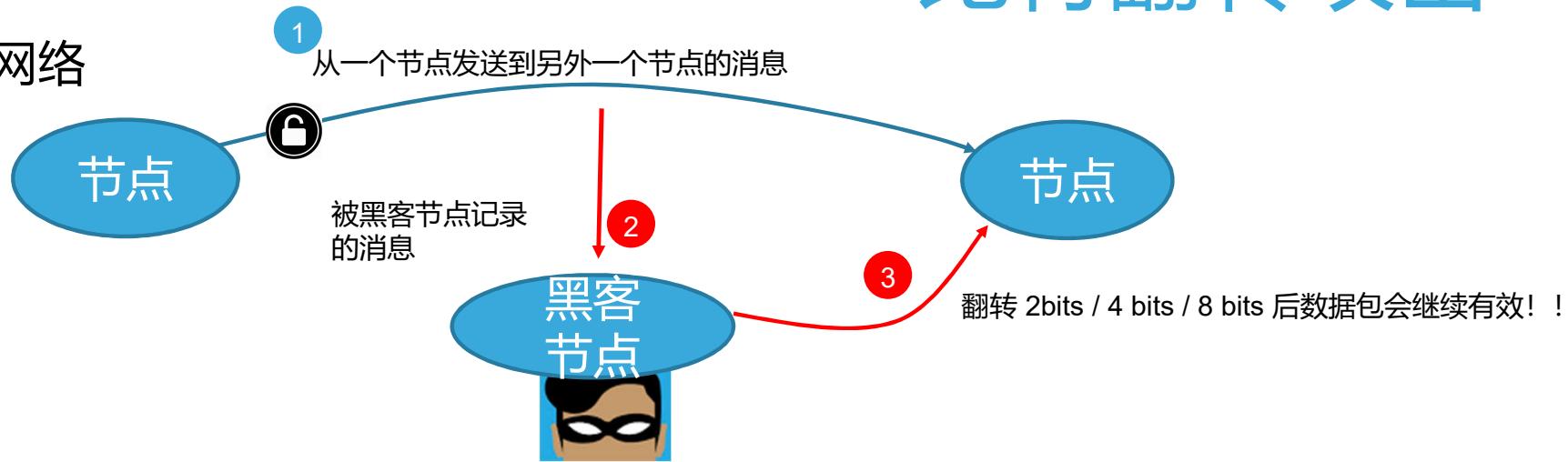
Bluetooth  
Mesh



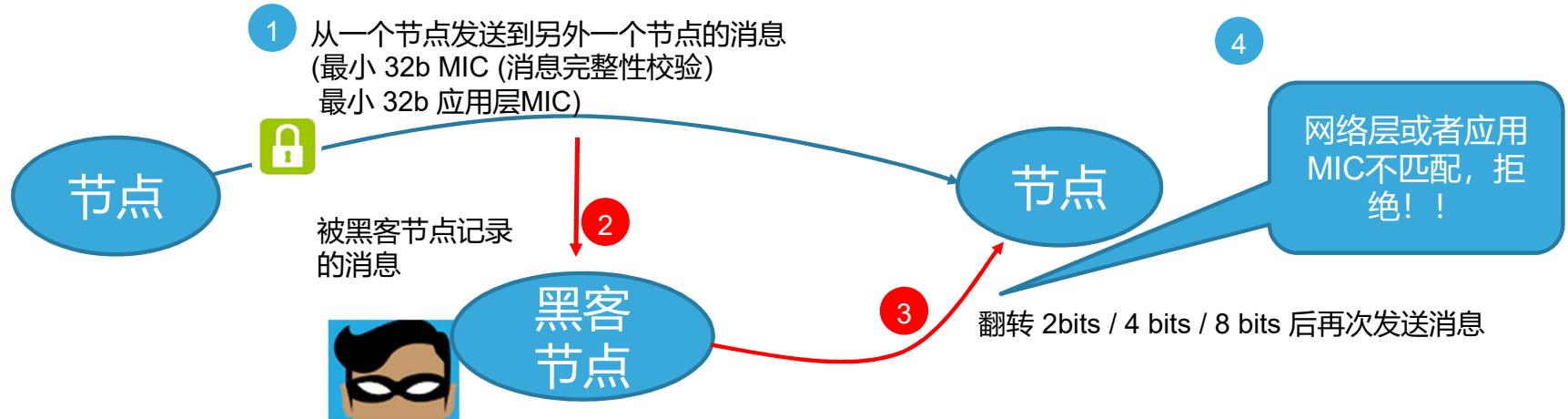
# 比特翻转攻击

76

## 不安全的网络



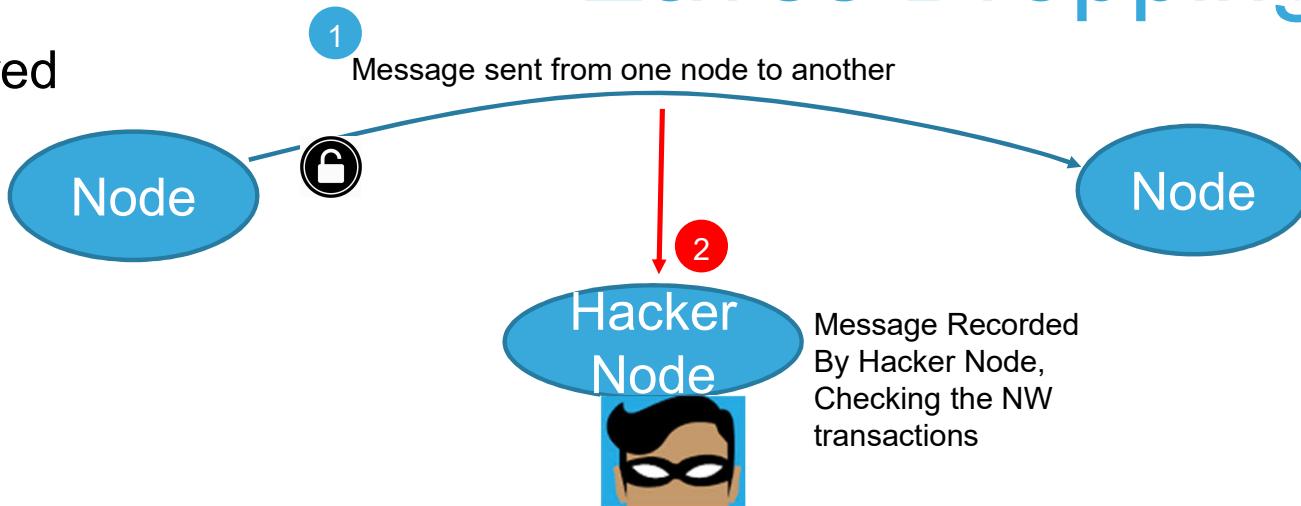
## 蓝牙 Mesh



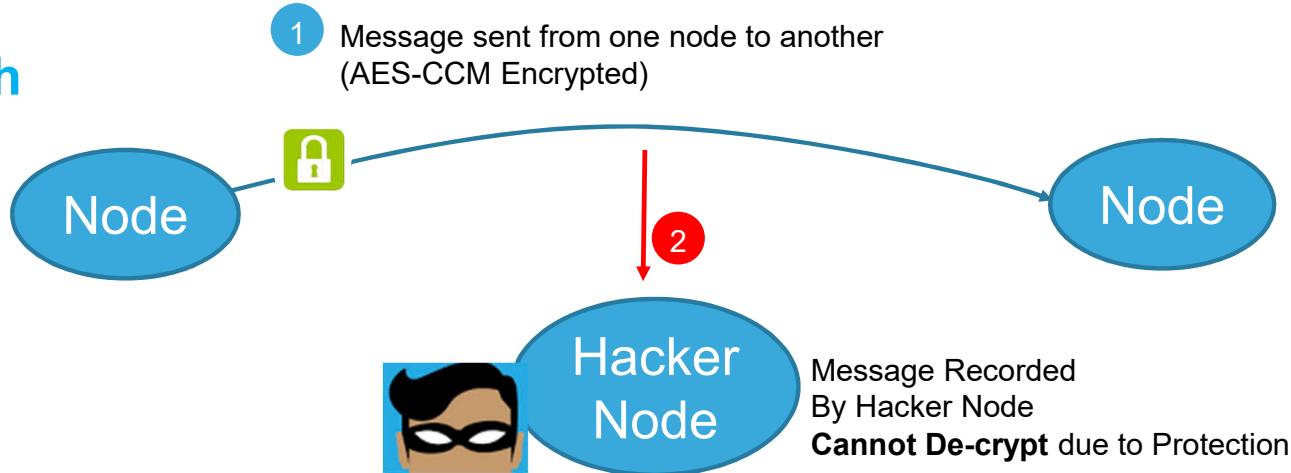
# Eaves Dropping Attack

77

Un-secured  
Network



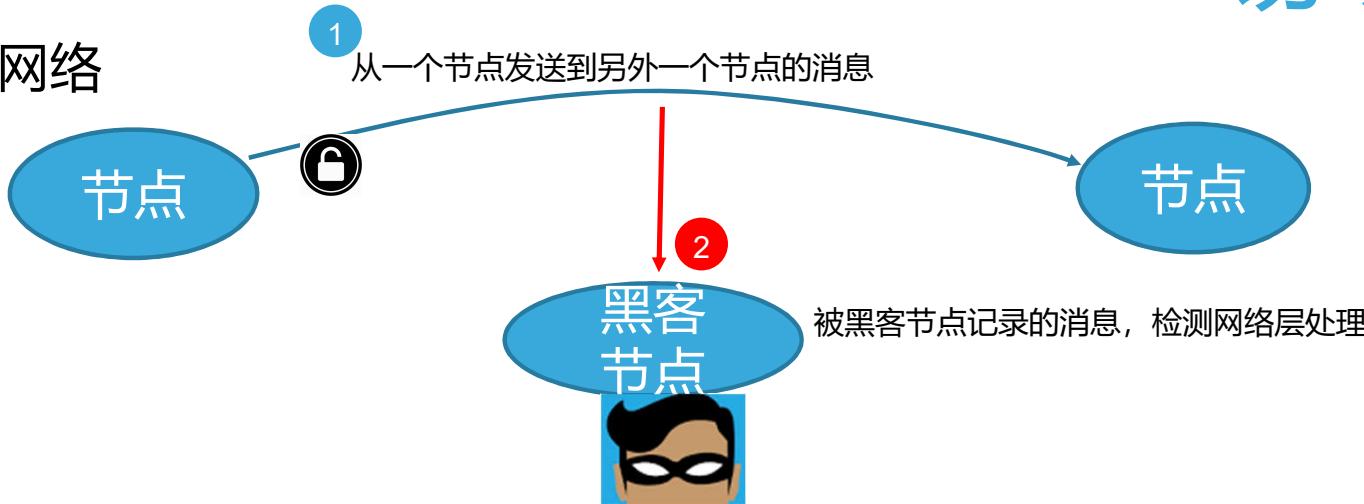
Bluetooth  
Mesh



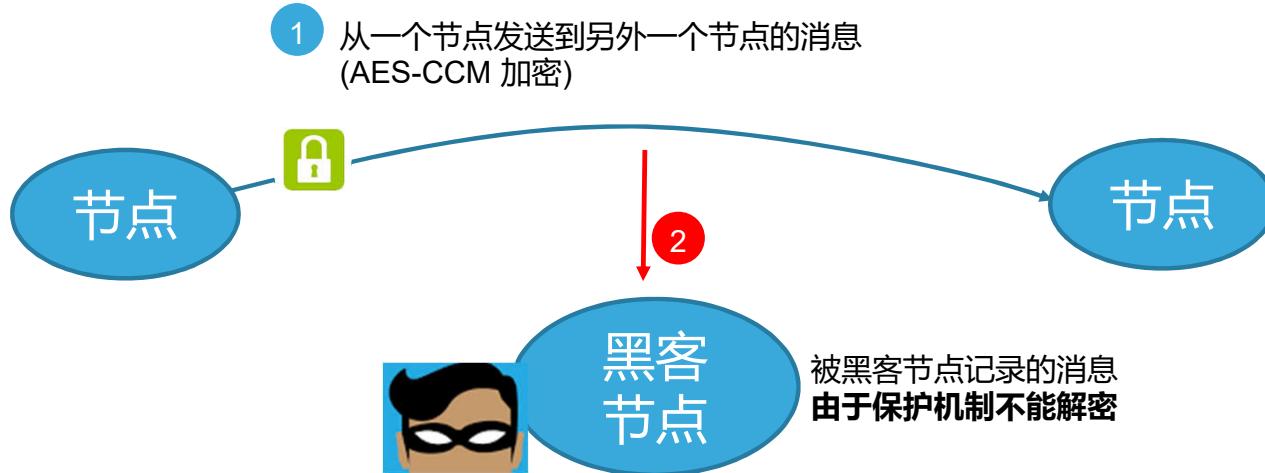
# 窃听攻击

78

不安全的网络

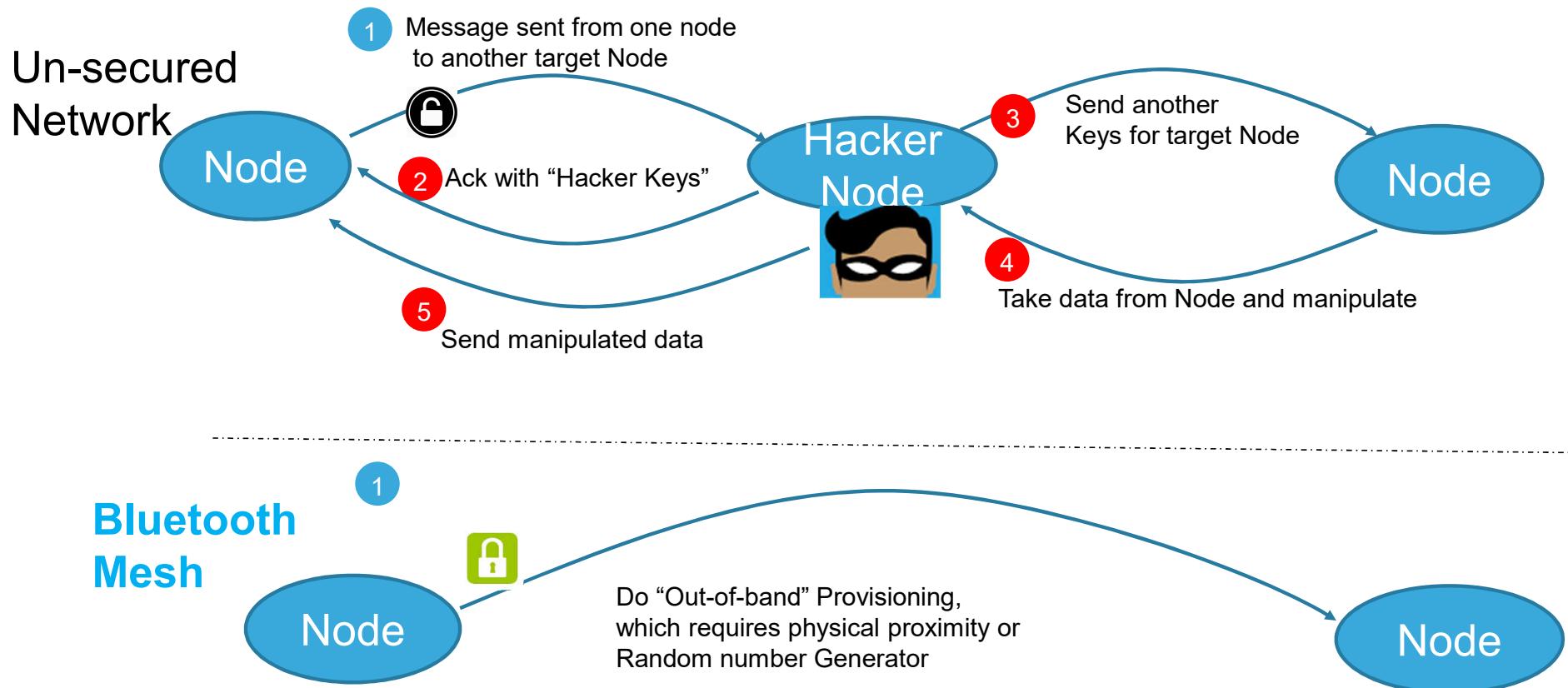


蓝牙  
Mesh



# Man-in-the-middle attack

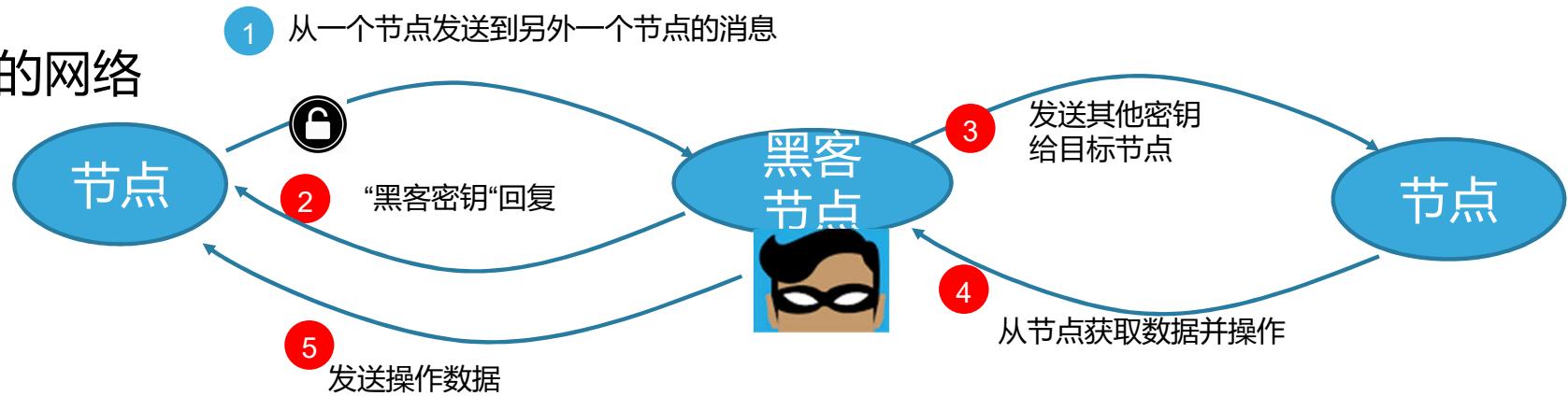
79



# 中间人攻击

80

不安全的网络



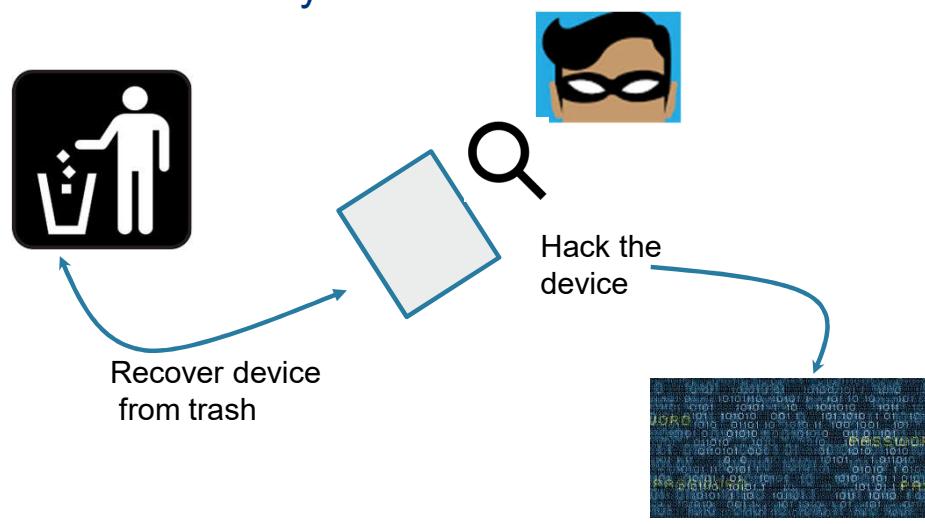
蓝牙  
Mesh



# Trash-can Attack

- Attack

- Recover the “faulty” node from trash
- Reverse-engineer to find the keys
- Recover the Network key, Application Key



- Checks

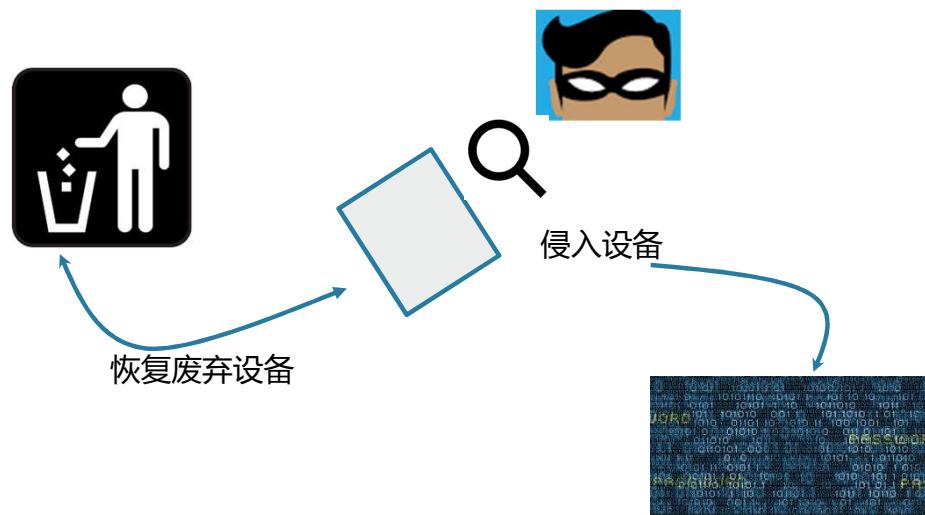
- Use of a device key is designed to protect against the “trash-can” attack by allowing the distribution of new network and application keys to selected devices only.
- UN-provision the node to delete the network, application keys via App
- Unprovision via hardware (if BLE connection not working) !!
- Blacklist the node
- Do key-refresh on the remaining used Network



# 垃圾桶攻击

- 攻击

- 恢复废弃节点
- 反向处理去获取密钥
- 恢复网络密钥，应用密钥



- 校验

- 设备密钥的使用是为防止“垃圾桶”攻击而设计的，通过允许新的网络和应用密钥发布到只被选择的设备的方式。
- 通过app解除节点以删除网络、应用密钥
- 通过硬件解除配置（如果BLE没有连接）
- 将节点设置为黑名单
- 在剩余的可用网络中执行密钥更新

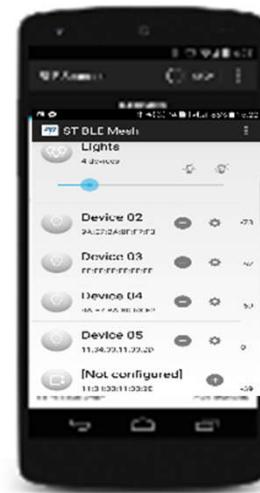


# Provisioning

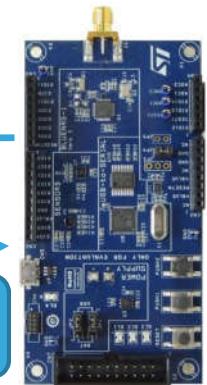
83

- Provisioning is a process of adding an unprovisioned device to a mesh network
- Managed by a Provisioner
- Provisioner is typically a smart phone or other mobile computing device
- Device can be identified to a Provisioner by its Device UUID

Provisioner



New Device



Advertising

Provision

Node becomes  
part of network

# Provisioning

84

- Provisioning 是一个把没有 Provisioning 设备添加到网络的过程
- 由 Provisioner 来管理
- Provisioner 是典型的智能手机或者移动电脑装置
- 装置由设备的通用唯一标识符来区分

Provisioner

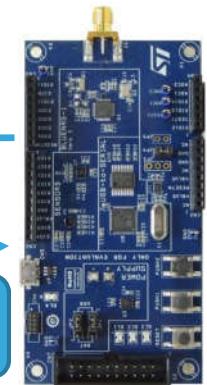


广播

Provision

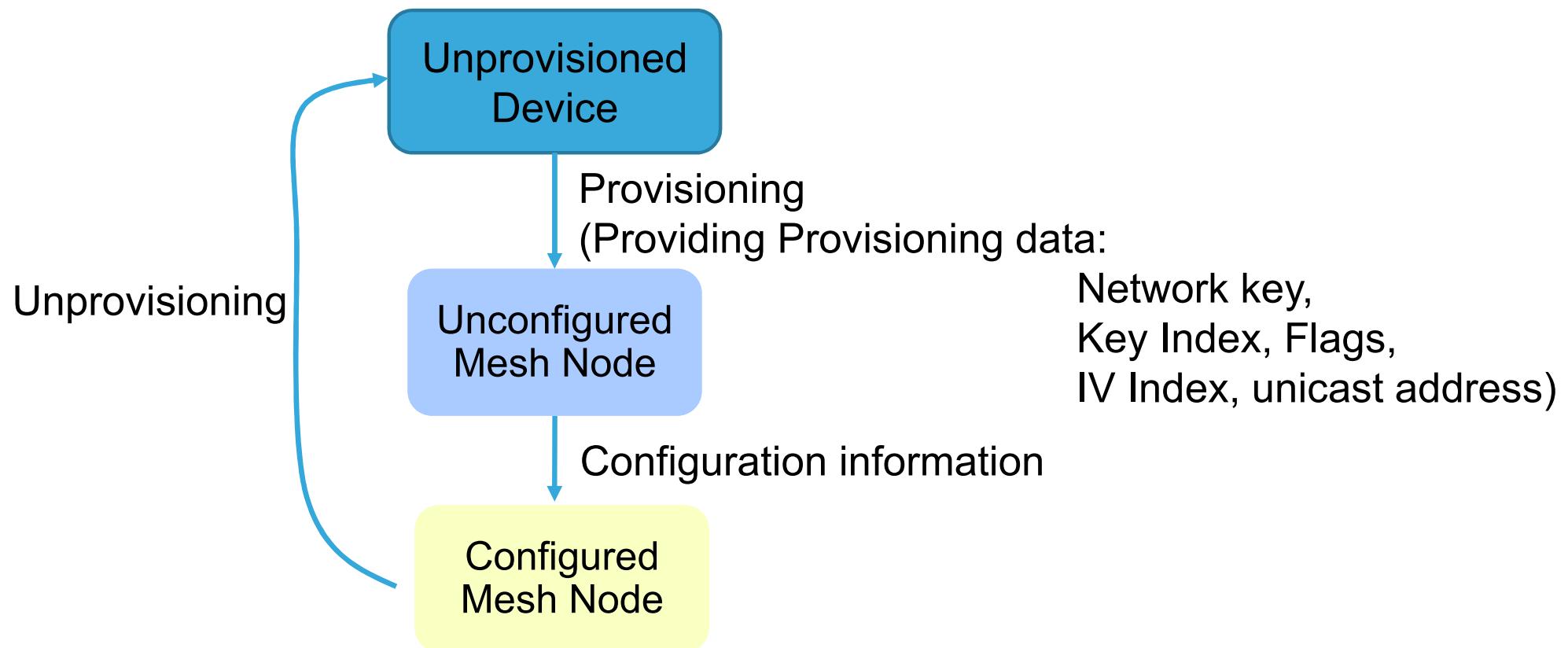
节点加入网络

新设备

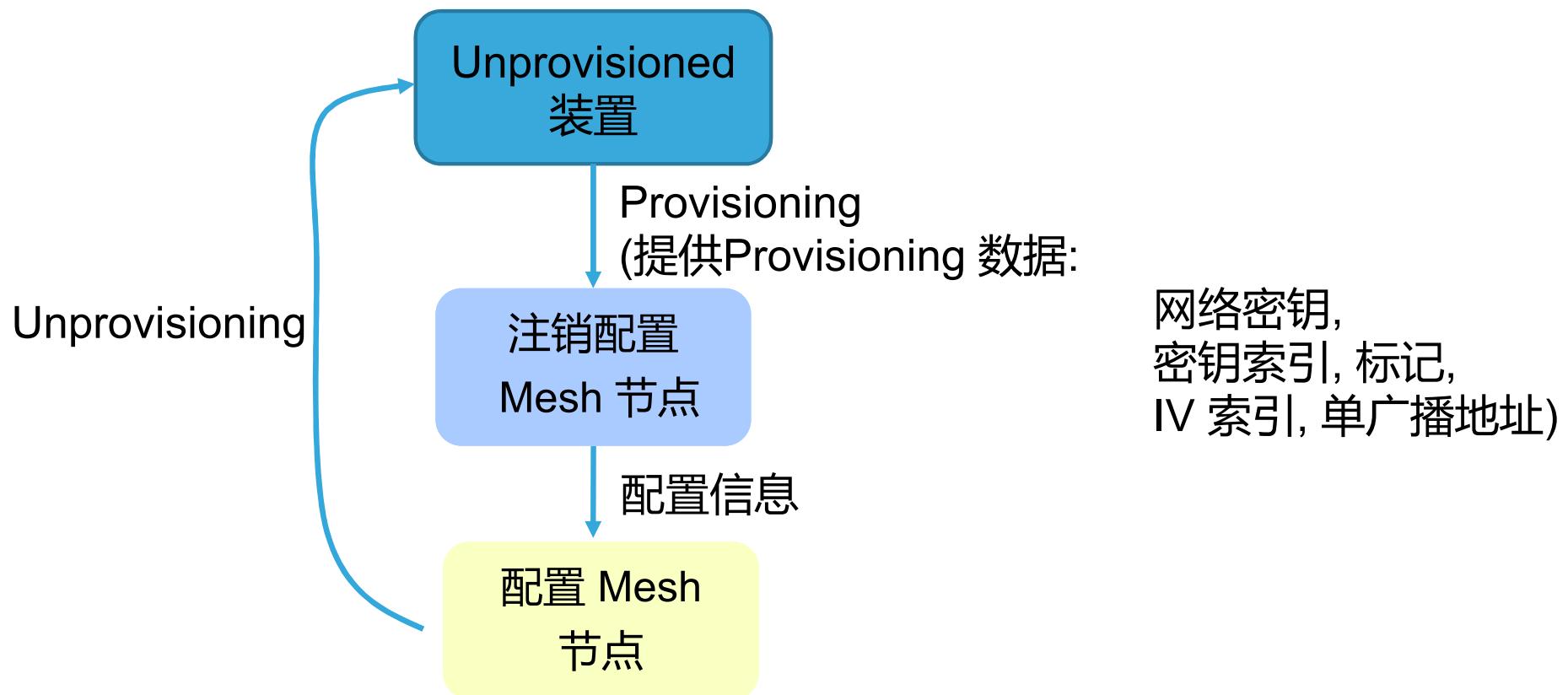


# Provisioning & Configuring Mesh Nodes

85



# Provisioning & 配置 Mesh 节点



# Mesh Provisioning Bearers

87

- How can you provision devices?

- PB-ADV

- Uses LE advertising packets
- Must be within range of device
- Requires OS support to send advertisements

- PB-GATT

- Uses GATT service
- Must be within range of device
- Supported by all oses “smart ready”



# Mesh Provisioning 承载

88

- 怎样provision 设备?

- PB-ADV

- 使用LE广播数据包
- 必须在设备工作范围内
- 需要 OS 支持发送广播

- PB-GATT

- 使用 GATT 服务
- 必须在设备工作范围内
- 所有的“smart ready”都支持



Used in  
BlueNRG-  
Mesh

# Provisioning Steps

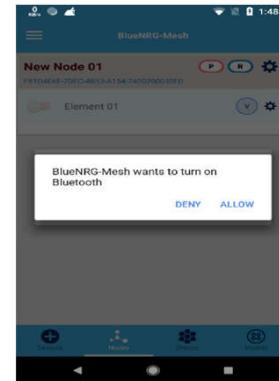
1

Launch ST BlueNRG Mobile  
Android Application



2

App Prompts to turn ON  
the Bluetooth(if not open)



3

Identify Un-provisioned  
Node

4

Device UUID For  
Node Identification.

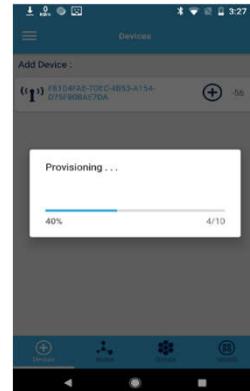


5

Click Here For Node  
Provisioning

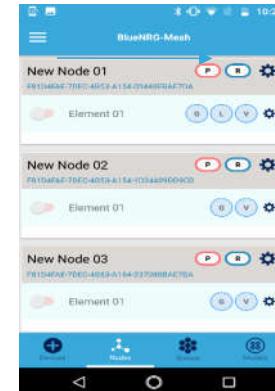
6

Provisioning  
Process Starts



7

Provisioned  
Node



# 启动配置 步骤

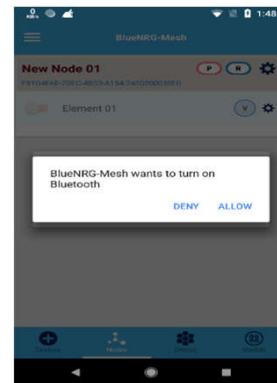
1

启动 ST BlueNRG 手机安卓  
应用程序



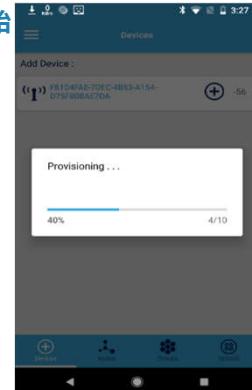
2

APP会提示打开蓝牙功能  
(如果没开的话)



6

启动配置过程开始



3

确定未配置的节点

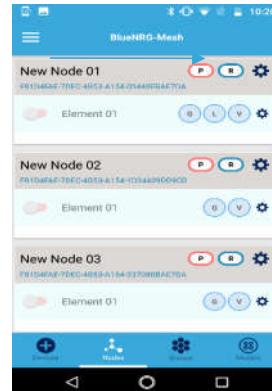


4

提供设备 UUID 给节点  
进行确认.

7

节点配置完成



# Provisioning Steps

91

- How is provisioning done?

- Setup

- Unprovisioning Device Beacon
  - Includes DeviceUUID
- Invite new device
- Agree on authentication algorithm to be used
- Share public keys

- Authentication

- No OOB Authentication

- Key Distribution

- Create session key
- Encrypt network key, device address and send to new device
- Creates pair-wise devicekey

# Provisioning 步骤

92

- 怎样provision 设备?

- 步骤

- Unprovisioning 信标设备
  - Includes 包含设备通用唯一标识符
- 邀请新设备
- 同意使用鉴权算法
- 分享公共密钥

- 鉴权

- 非 OOB认证

- 密钥分配

- 创建段密钥
- 加密网络密钥，设备地址并发送给新节点
- 创建智能配对设备密钥

# Configuration Steps

93

- A node cannot do anything until it is configured...

- Device Configuration

- Every device has a configuration block
- Includes:
  - Company ID
  - Product ID
  - “Model” support
- Reading configuration block necessary to allow provisioner to configure new nodes

- Adding Keys / Binding

- Provisioning does not add app keys to nodes (during proxy)
- Installation of application keys
- Application keys are bound to network keys

- Publish / Subscribe

- Configure how nodes will publish and subscribe
- Configuration by model
  - Some will publish
  - Some will subscribe
  - Some will be configured to do neither

# 配置步骤

94

- 配置完成之前的节点做不了任何事情...

## 设备配置

- 每一个设备都有一个配置区间
- 包含:
  - 公司 ID
  - 产品 ID
  - 支持的“模型”
- 允许手机通过读取配置区间来配置新节点

## 添加密钥/绑定

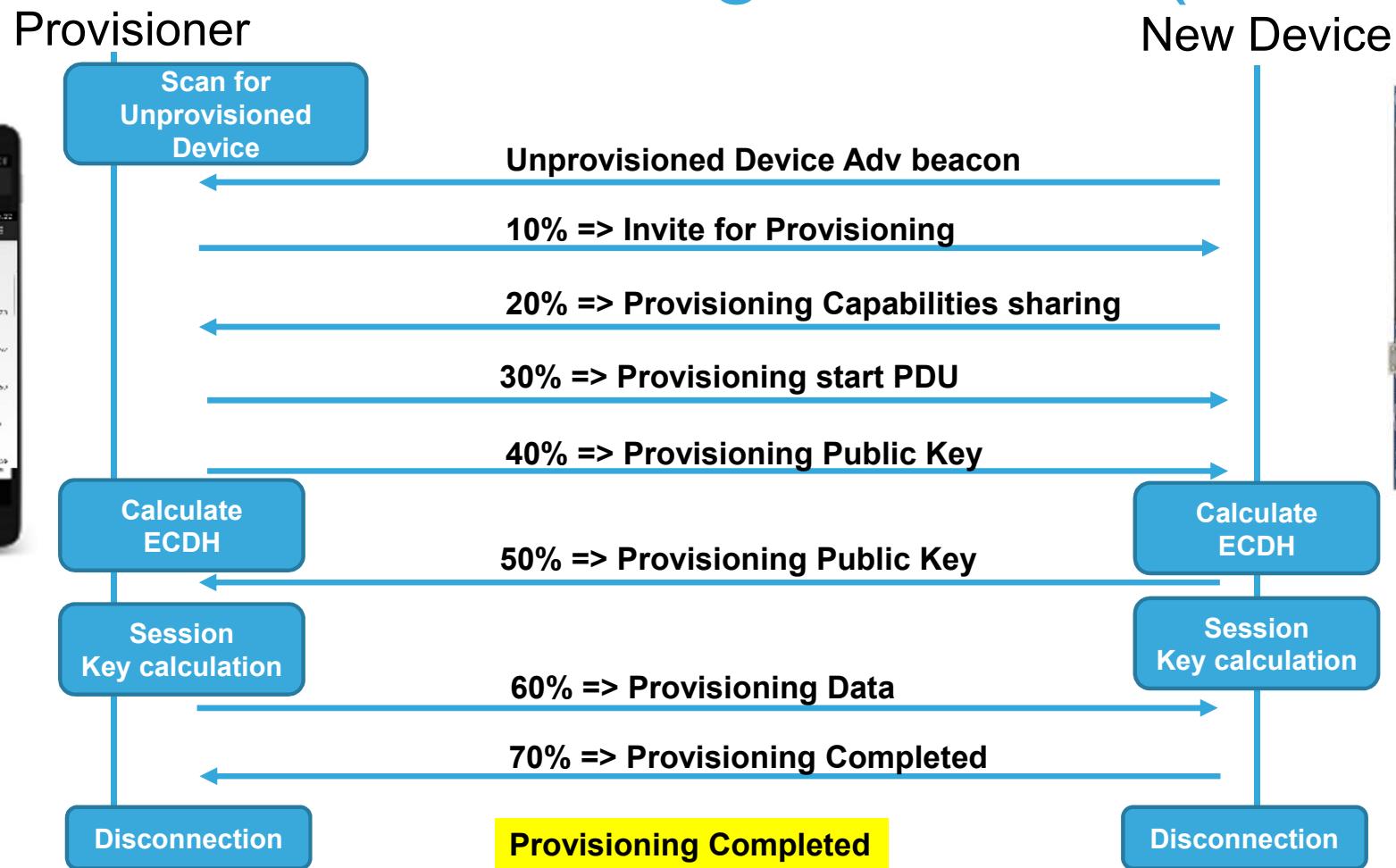
- Provisioning 不添加app密钥到节点(设置网关的过程中)
- 安装应用密钥
- 应用密钥绑定到网络上

## 公开 / 订阅

- 配置节点如何公开/订阅
- 模型配置
  - 部分公开
  - 部分订阅
  - 部分配置成既不公开也不订阅

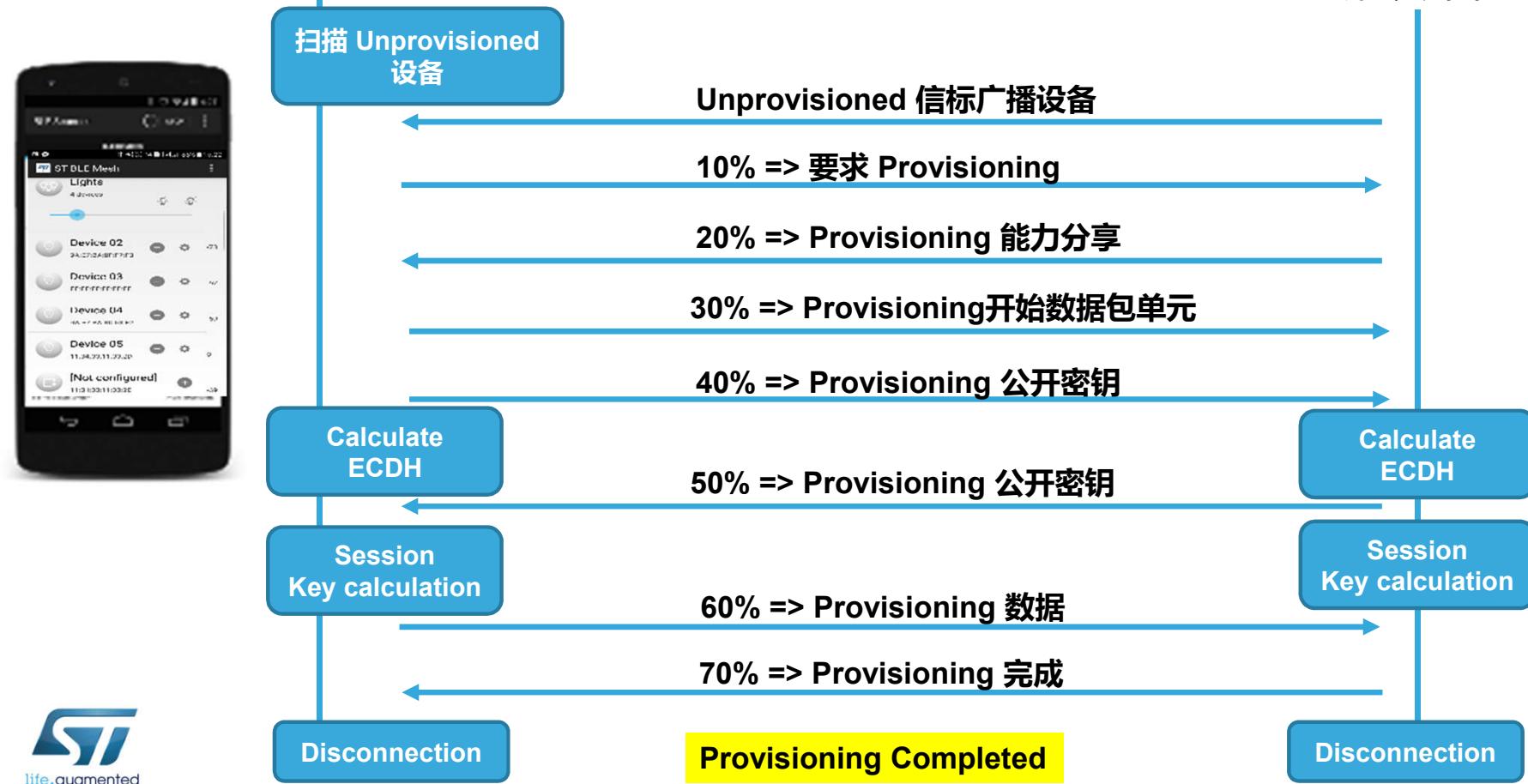
# Provisioning Process (0-70%)

95

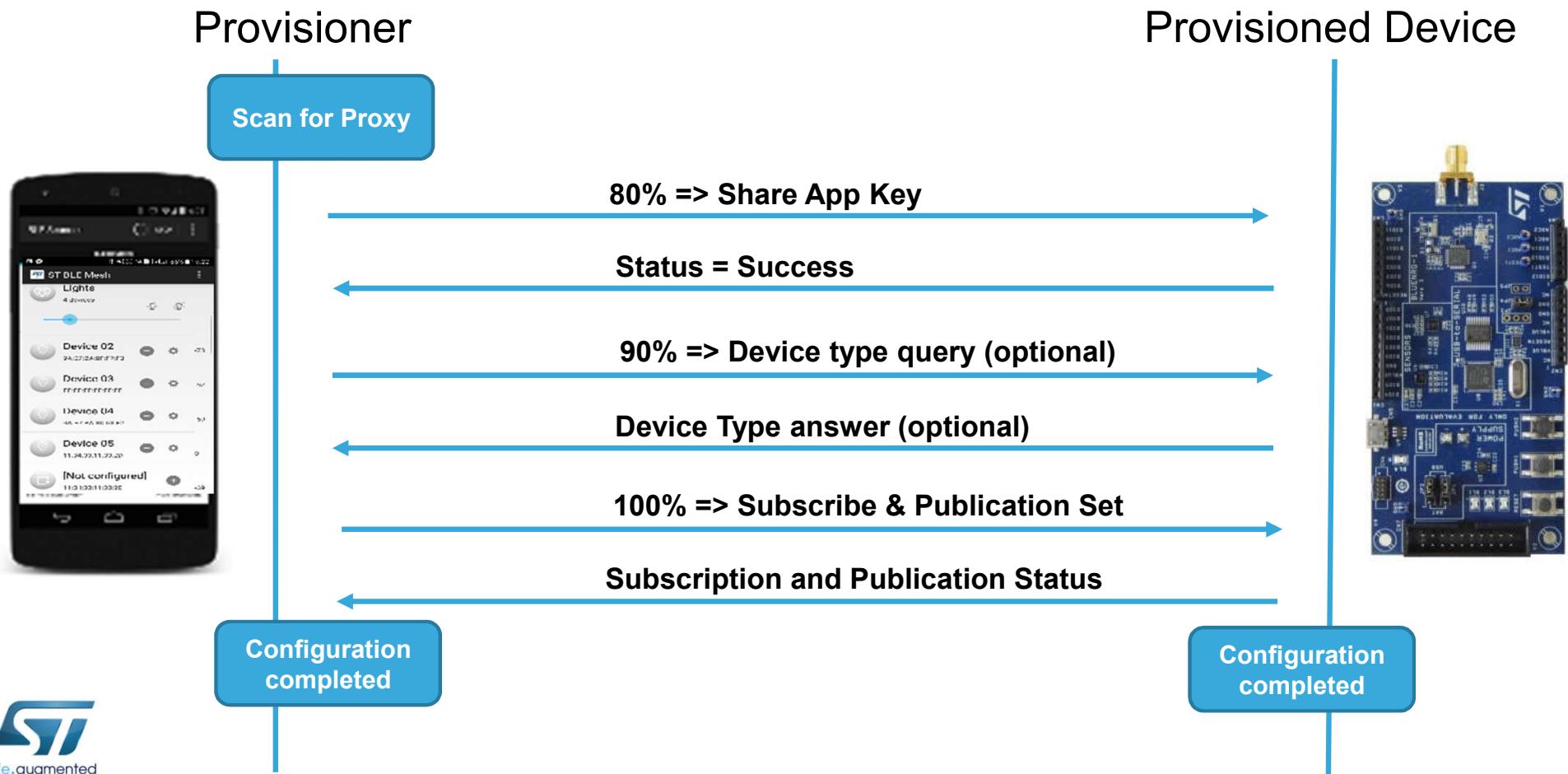


# Provisioning 流程 (0-70%)

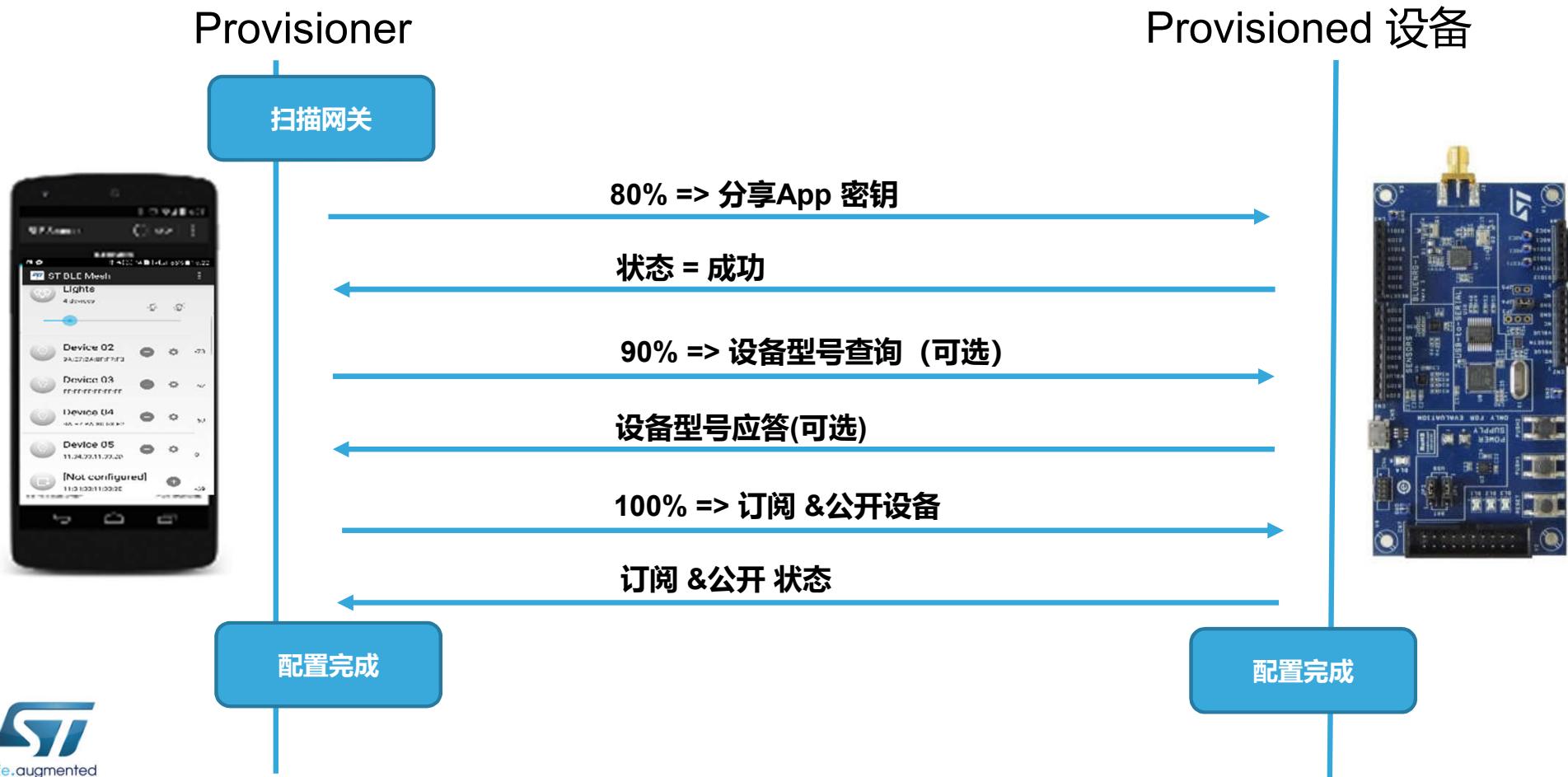
Provisioner 新设备



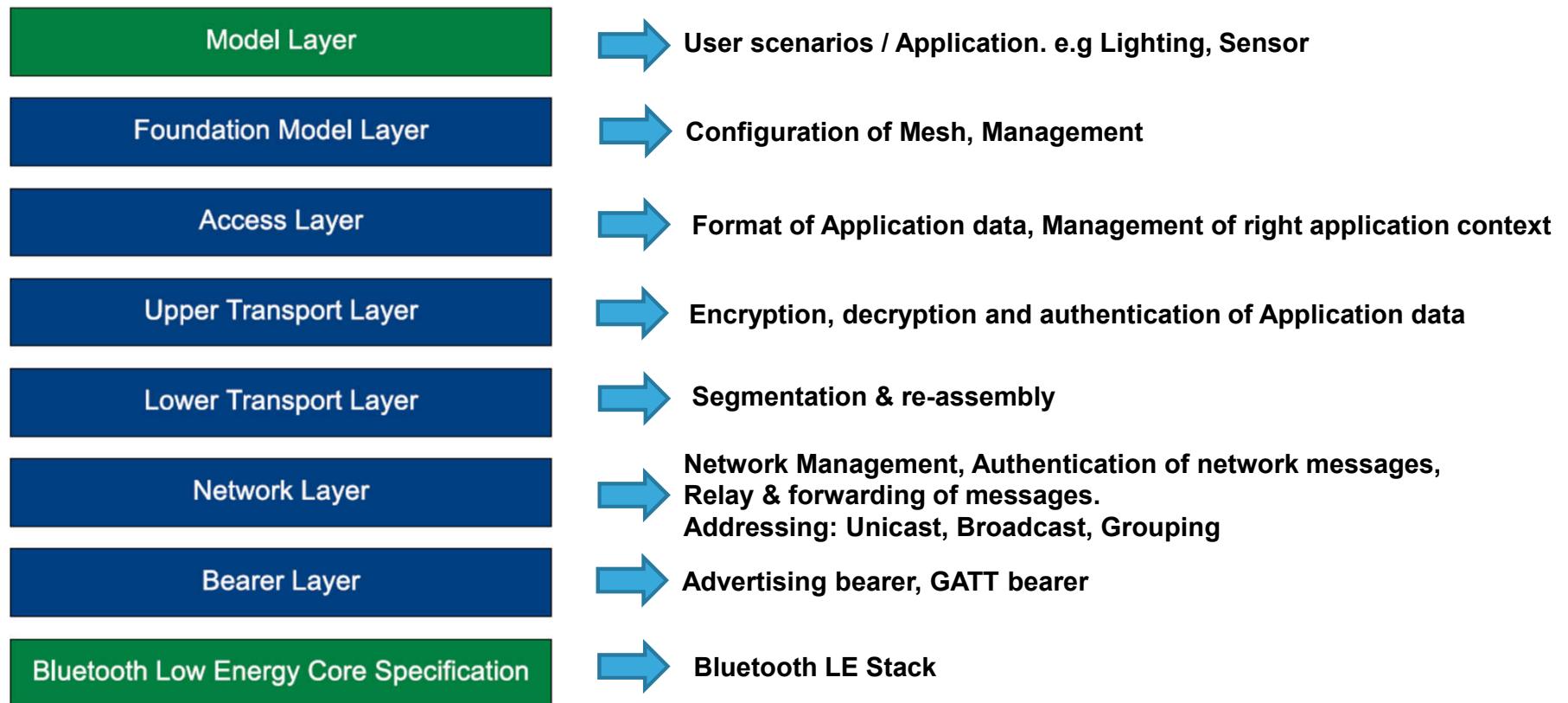
# Provisioning Process (70-100%)



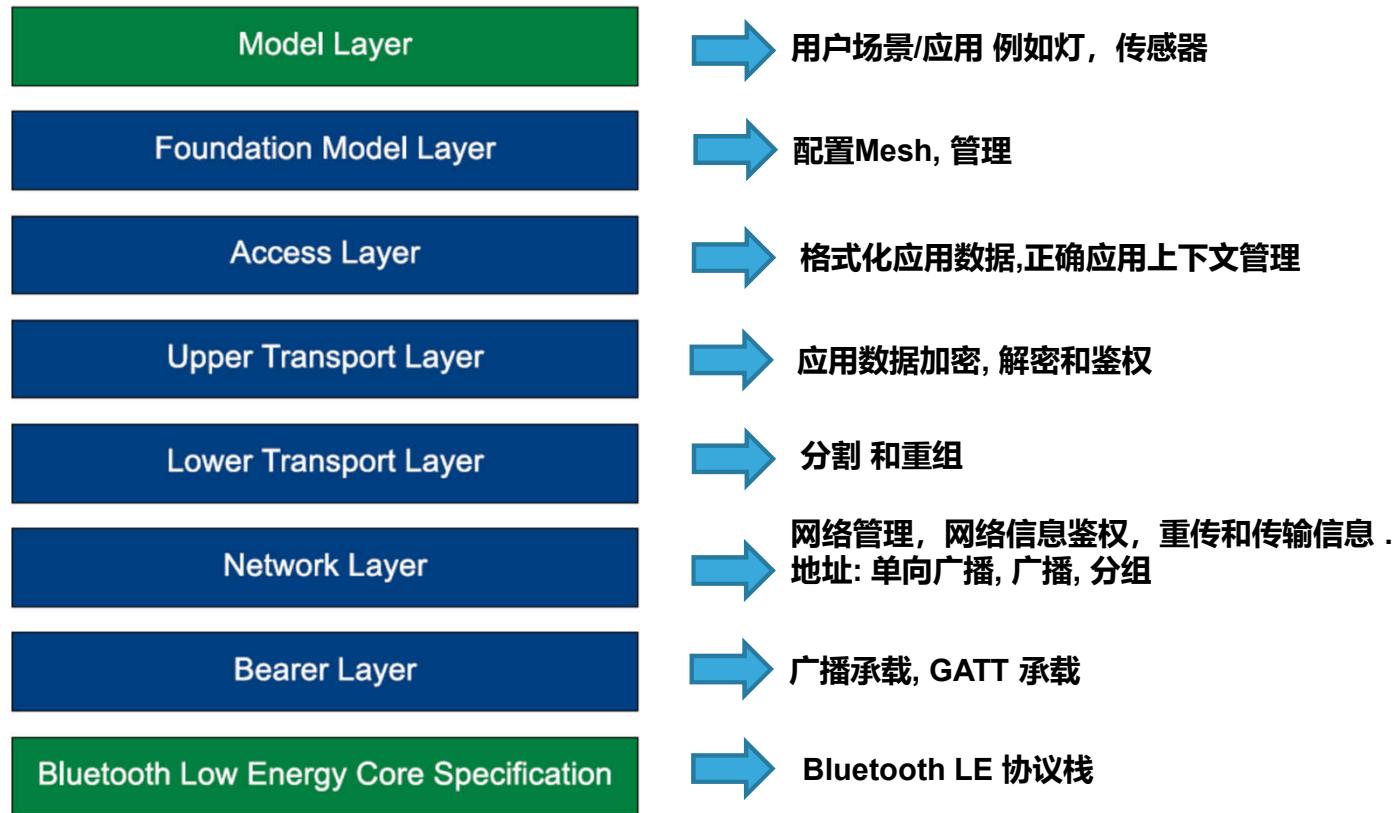
# Provisioning 流程 (70-100%)



# BLE Mesh system architecture



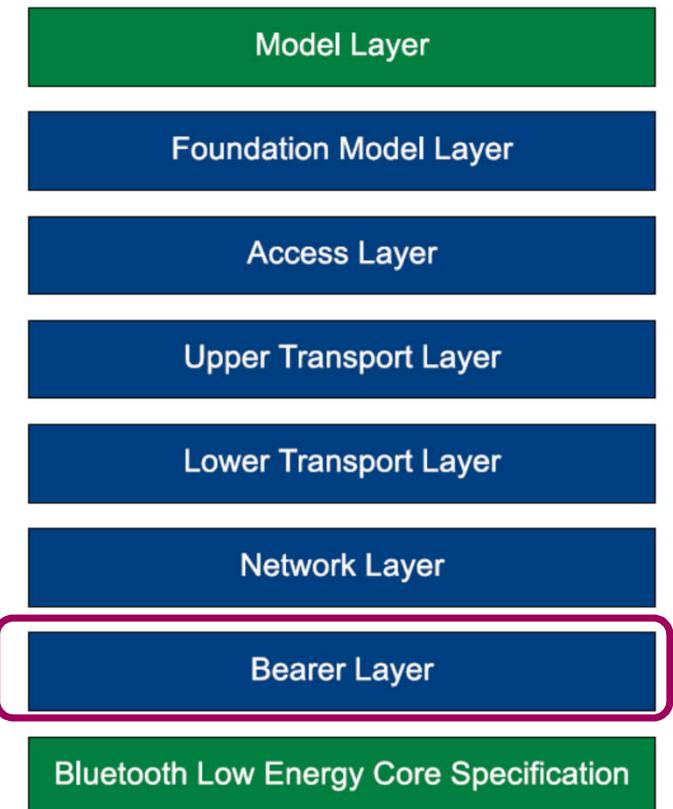
# BLE Mesh系统架构



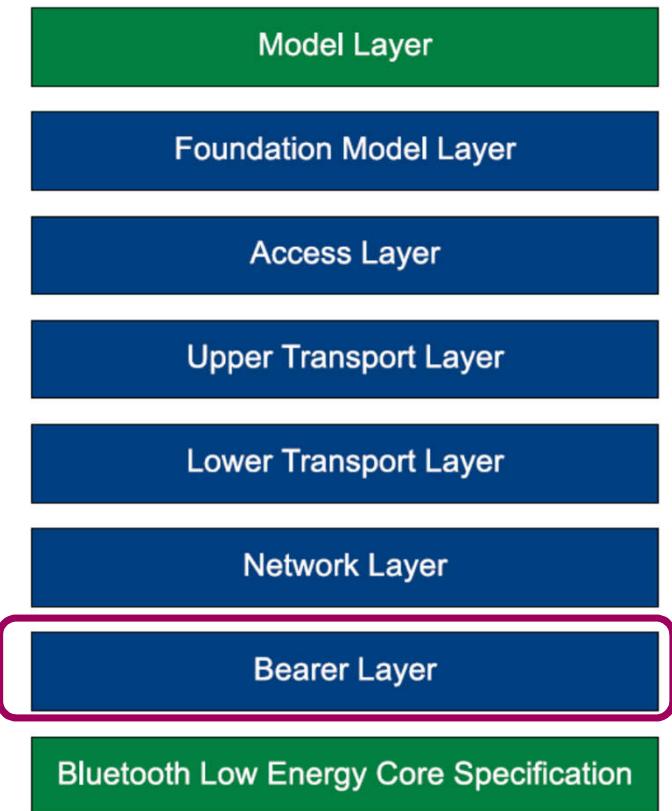
# Bearer Layer

101

- Communication system for transmission and receipt of Mesh messages
- Advertising bearer
  - Uses LE advertising packets : non-connectable advertising
  - Leverages Bluetooth LE's GAP advertising and scanning features
  - Limited to 29 octets of network message
- GATT bearer
  - Uses LE connection to send messages point to point
  - Allows a device which does not support the Advertising Bearer to communicate indirectly with nodes of a mesh network
  - Uses GATT service
  - Supported by all OSes "Smart Ready"

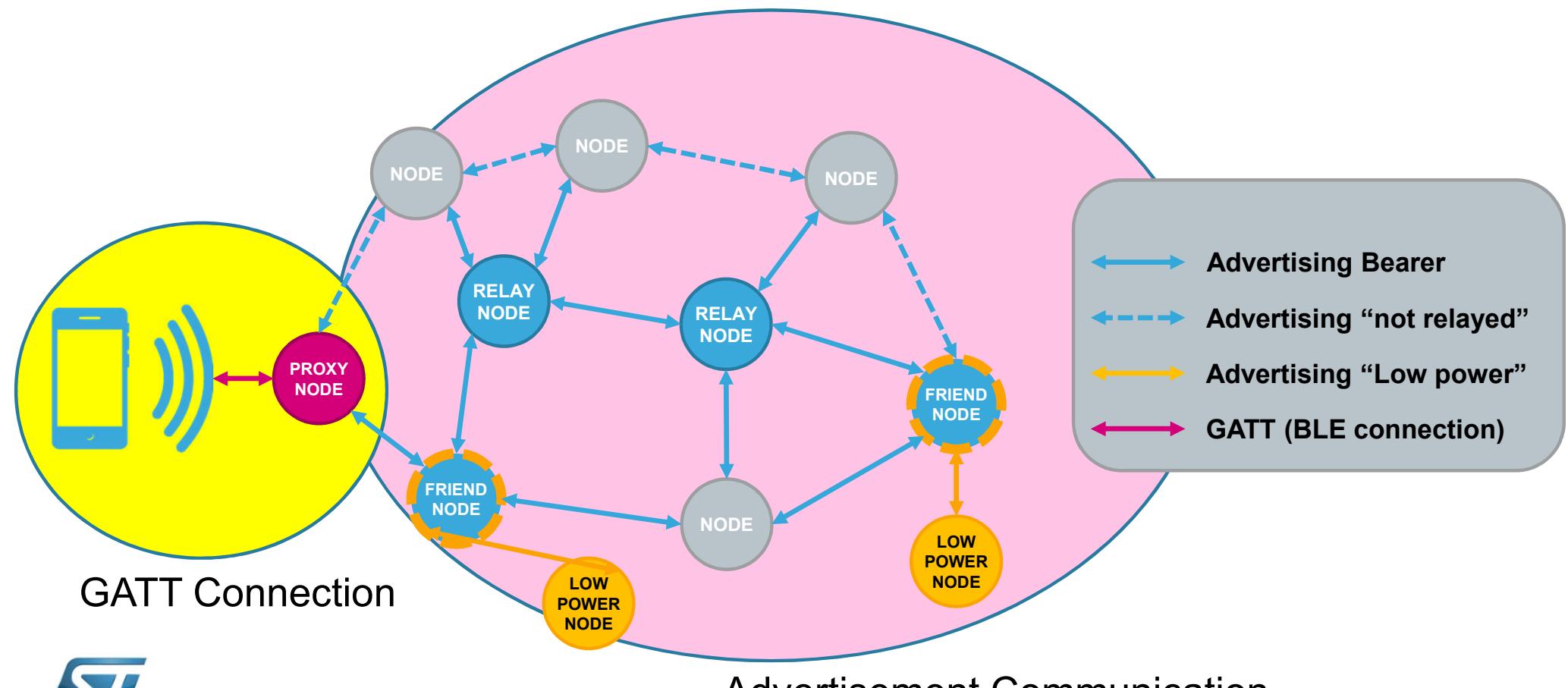


- 通信系统传输和接收Mesh信息
- 广播承载
  - 使用 LE 广播数据包 : 不可连接的广播
  - 蓝牙LE's GAP 广播扫描特性
  - 仅限于 29 八位网络信息
- GATT 承载
  - 使用LE连接发送点到点数据
  - 允许一个不支持广播承载的设备和Mesh网络节点间接的通信
  - 使用GATT 服务
  - 支持所有的“Smart Ready”设备

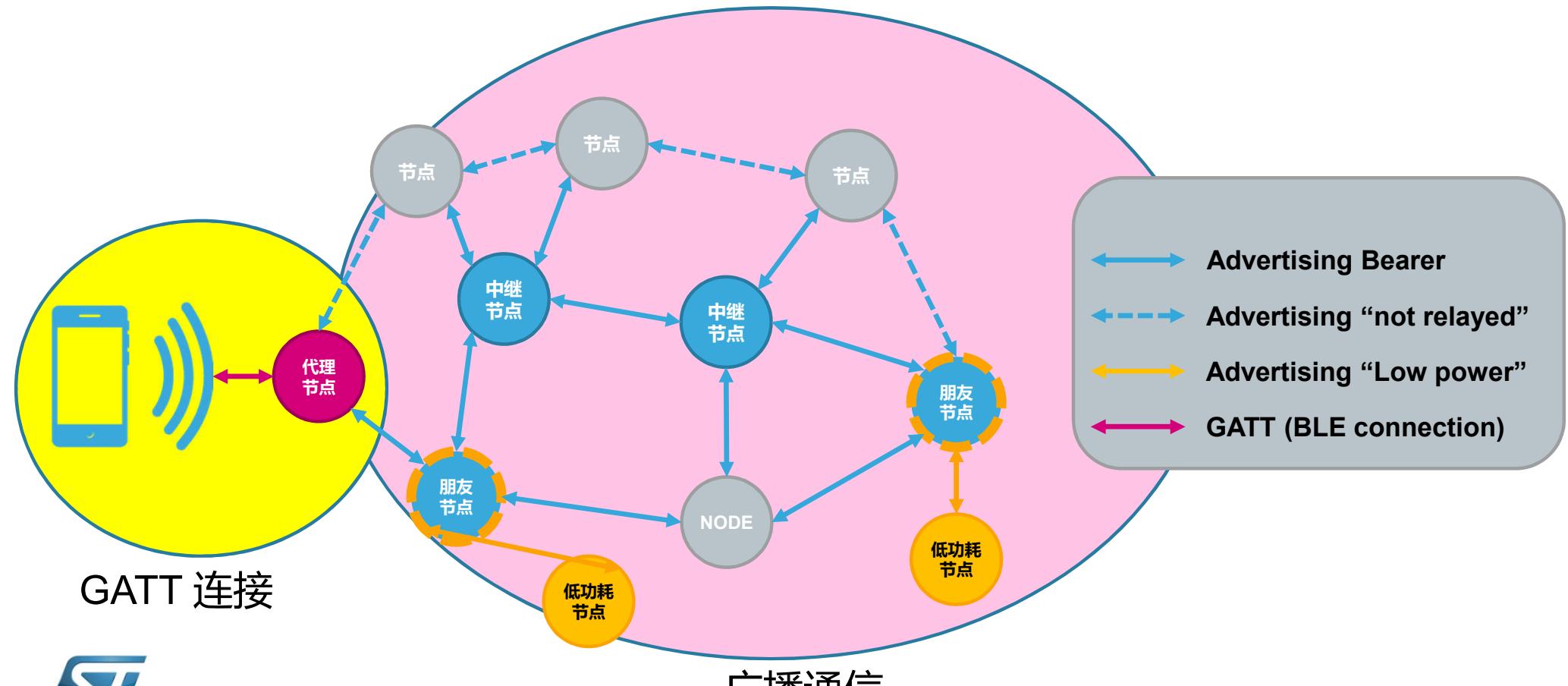


# BLE Mesh Topology: Bearer

103

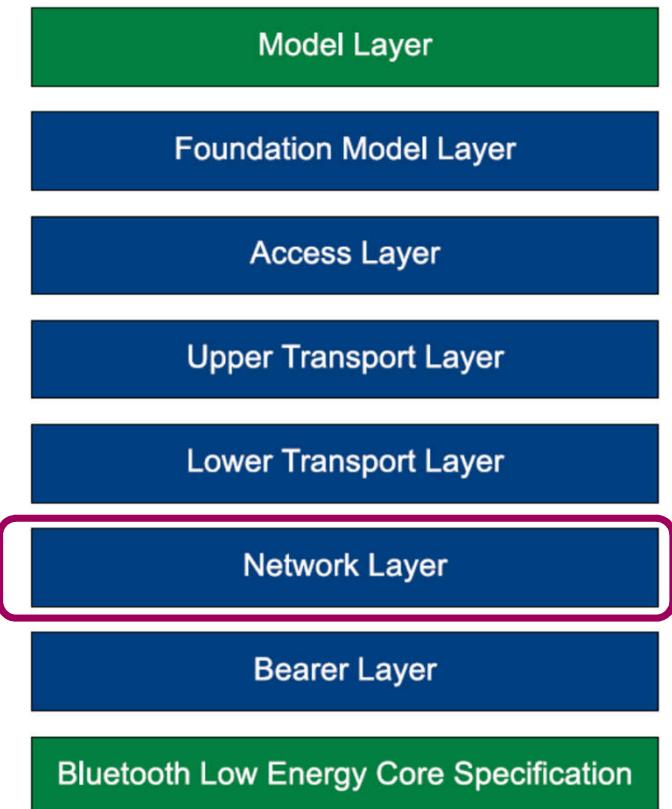


# BLE Mesh 拓扑: 承载层



# Network Layer

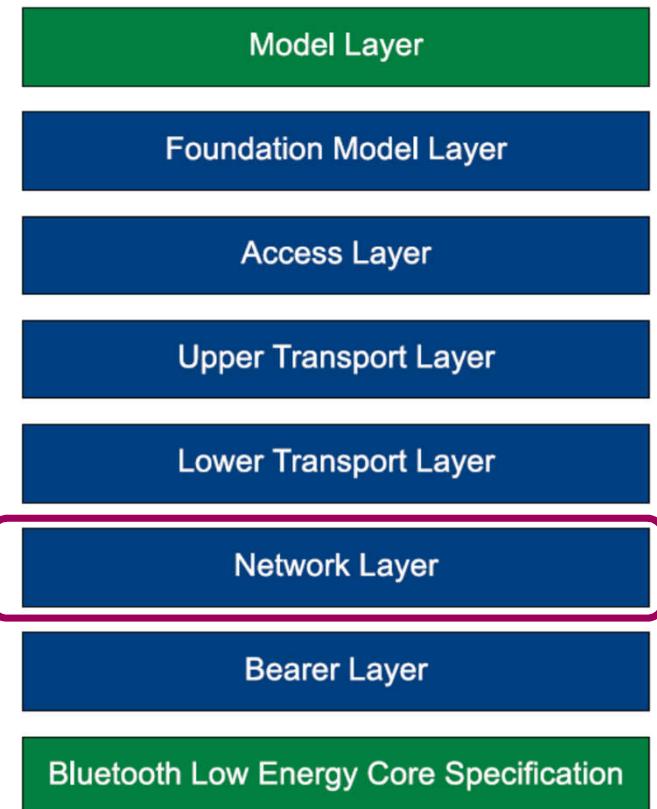
- Defines the various message address types and a network message format
- Allows transport layer PDUs to be transported by the bearer layer
- Filter messages arriving from the bearer layer to determine whether or not they should be delivered for further processing



# 网络层

106

- 定义多种信息的地址类型和网络信息格式
- 允许传输层数据包单元用承载层传输
- 过滤从承载层传输到的信息是否需要继续传递



# Network PDU

107

- What does a mesh network message look like?



- **IVI – Initialization Vector Index**
  - 1-bit value
  - Least significant bit of the IV index
  - Allows nodes to receive messages from last “IV index” and “current IV index”
  - Allows for efficient and seamless transition from one IV index to another

- **NID – Network Identifier**
  - 7-bit value
  - Determines which mesh network this message is sent under
  - Removes requirement to brute force all known network keys against this message
  - $2^{121}$  network keys for each NID – will not be unique

# 网络数据包单元

108

- Mesh 的信息数据包包含哪些内容?



- **IVI – 初始化矢量索引**

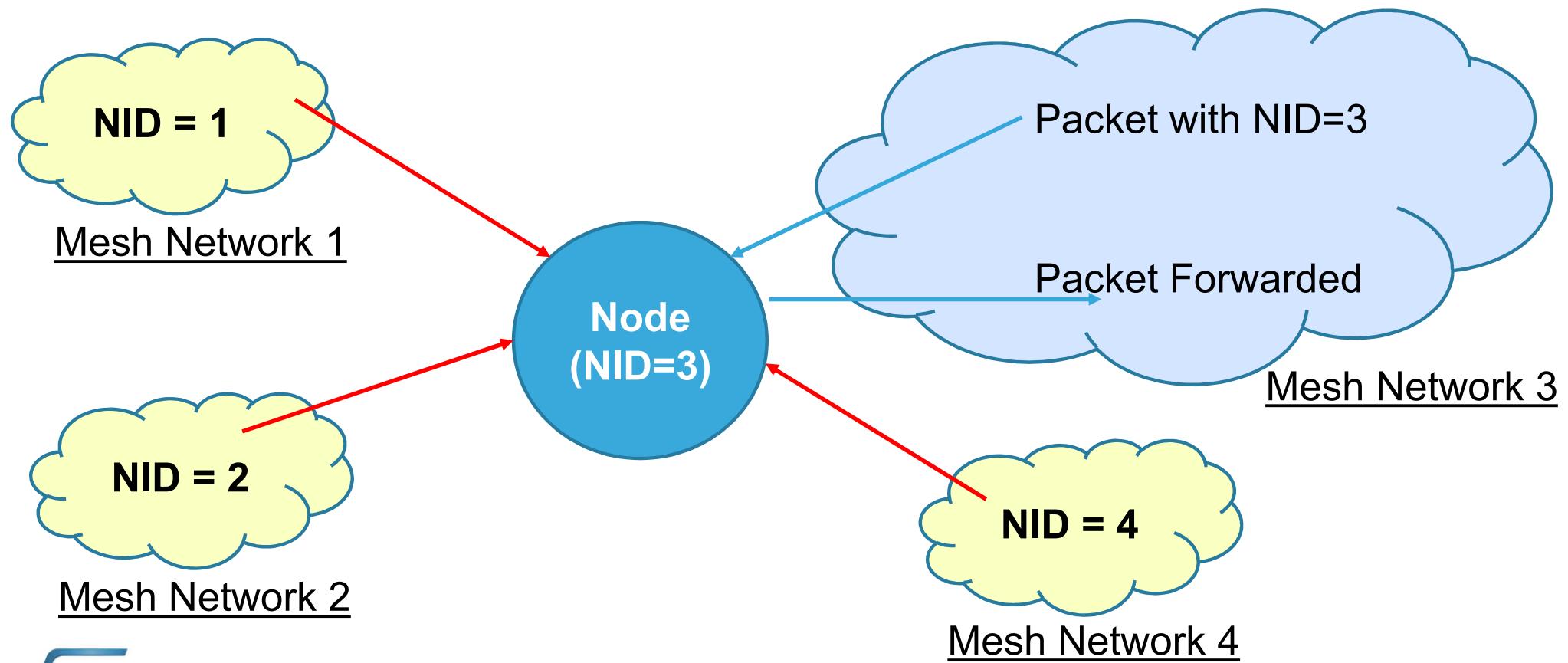
- 1 位 值
- IV索引的最小值位
- 允许节点从最新的“IV索引”和“当前的IV索引”接收信息
- 允许有效且无缝的从一个IV索引传输到另一个索引

- **NID – 网络身份**

- 7-位 值
- 决定使用哪个mesh网络传递数据
- 删除和该信息矛盾的请求
- 网络身份有 $2^{121}$  个网络密钥

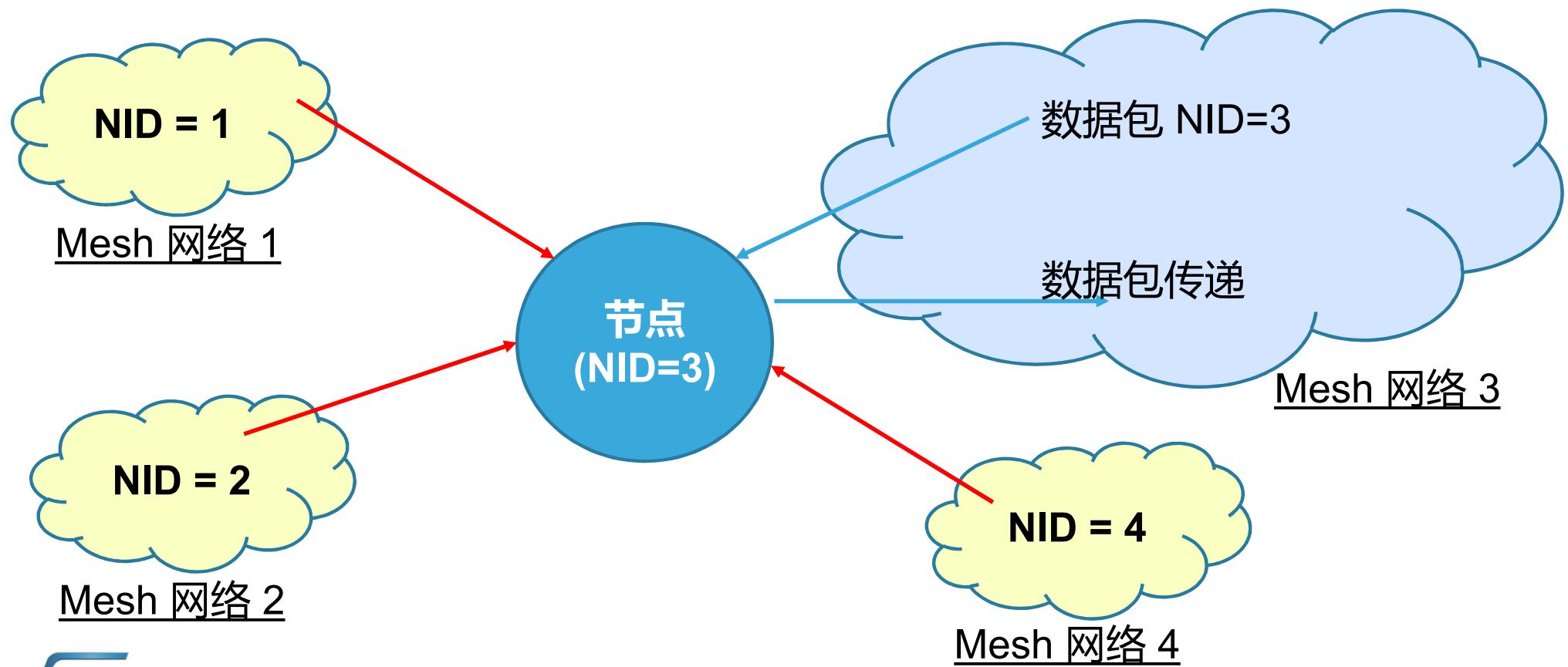
# NID: Network Identification

109



# NID: 网络身份

110



# Network PDU

111

- What does a mesh network message look like?



- **CTL – Control**

- 1-bit value
- Determines if this mesh message contains an “access message” or is a “control message”
- Access messages are delivered to access layer (via lower and upper transport)
- Control messages are delivered to lower and upper transport layer

# 网络数据单元

112

- Mesh 的信息数据包包含哪些内容?



- **CTL – 控制**

- 1-位值
- 决定是否mesh信息包含“接入信息”或者“控制信息”
- 接入信息在接入层传递(通过更低或更高层传输)
- 控制信息会被传输到更低和更高的传输层

# Network PDU

113

- What does a mesh network message look like?



- **TTL – Time to Live**

- 7-bit value
- Determines how many more times this message can be relayed
- 0 = do not relay – never been relayed
- 1 = do not relay - has been relayed
- $\geq 2$  = can be relayed

# 网络数据包单元

114

- Mesh 的信息数据包包含哪些内容?



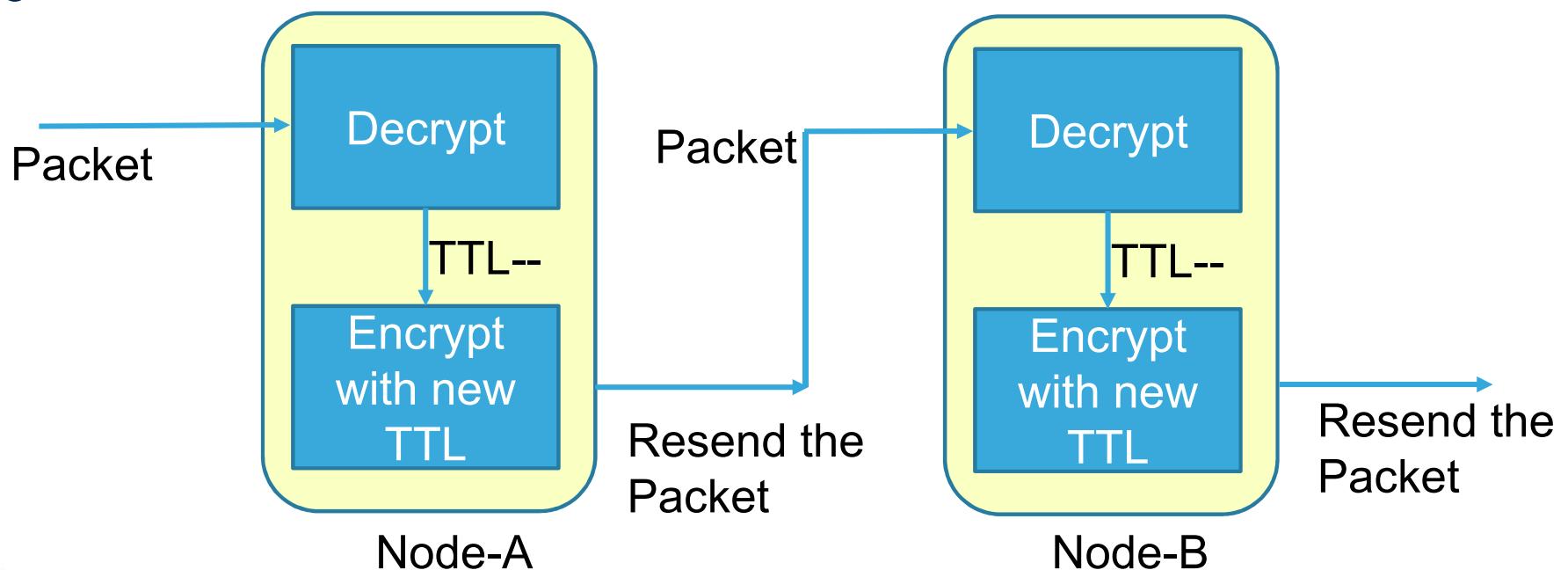
- TTL – 活动时间

- 7 位的值
- 决定该信息可以被续传几次
- 0 = 不续传 – 从未被续传
- 1 = 不续传 – 之前被续传过
- $\geq 2$  = 可以被续传

# TTL :Time to Live

115

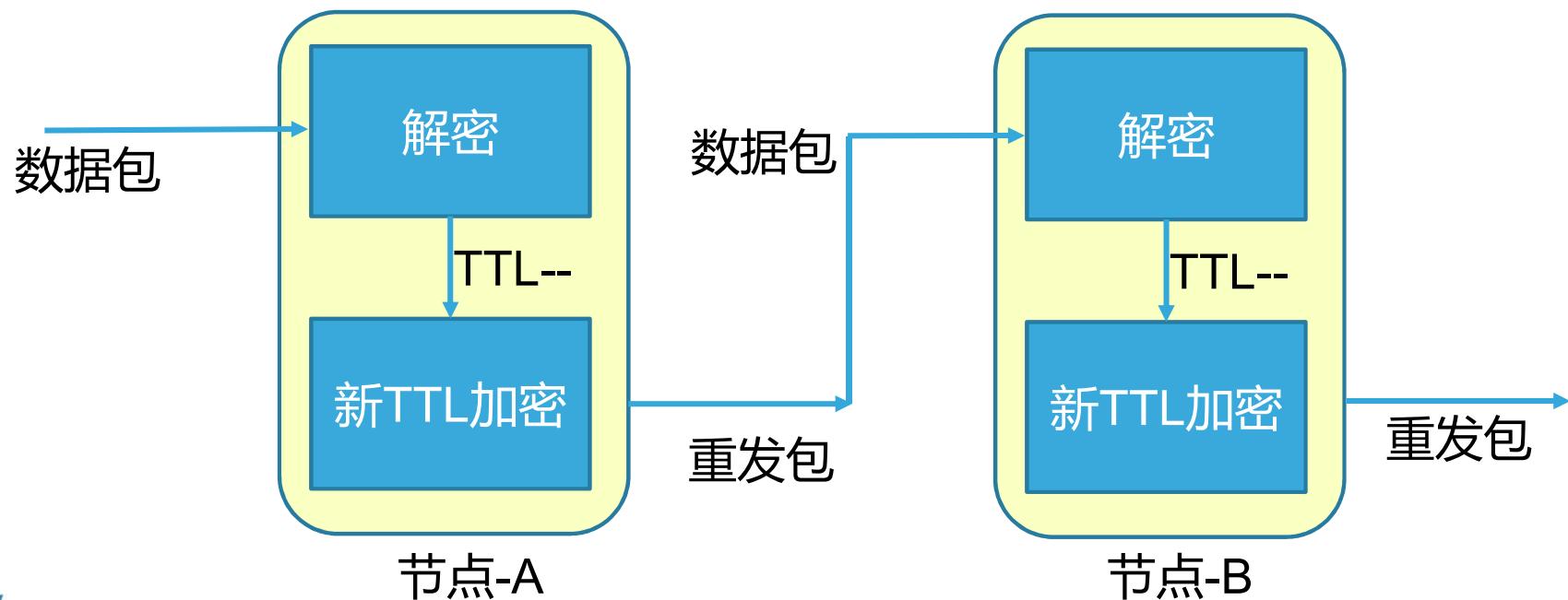
- Useful for managed flooding for relayed packets
- Whole packet is decrypted at Network Layer => Decrement the TTL => Encrypt again and send



# TTL : 活动时间

116

- 有效管理flooding的续传数据包
- 整个包在网络层解密 => 解密活动时间 => 再次加密然后发送



# Network PDU

117

- What does a mesh network message look like?



- **SEQ – Sequence Number**

- 24-bit value
- Must be unique for every mesh message sent
- Together with SRC, uniquely identifies mesh message
  - Can be used for “replay attack protection”

- Mesh 的信息数据包包含哪些内容?



- **SEQ – 序列号**

- 24 位值
- 每个mesh的信息该值必须不同
- 要有源地址, 唯一身份识别mesh 信息
  - 就可被用来“重发攻击防护”

# Network PDU

119

- What does a mesh network message look like?



- **SRC – Source Address**

- 16-bit value
- Identifies the original source of this message
- Must be a unicast address

- **DST – Destination Address**

- 16-bit value
- Identifies the destination(s) of this message
- Can be a unicast, virtual, group or the broadcast address
- Dst is encrypted using network key

# 网络数据包单元

120

- What does a mesh network message look like?



- **SRC – 源地址**

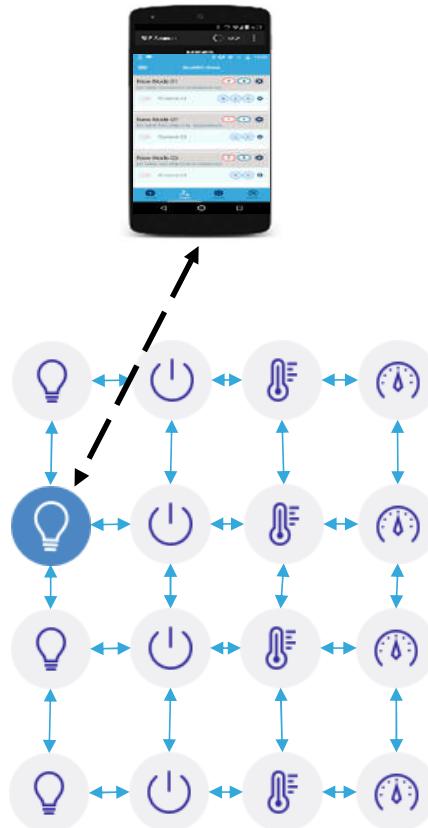
- 16位值
- 识别信息原始源
- 必须是一个单广播地址

- **DST – 目标地址**

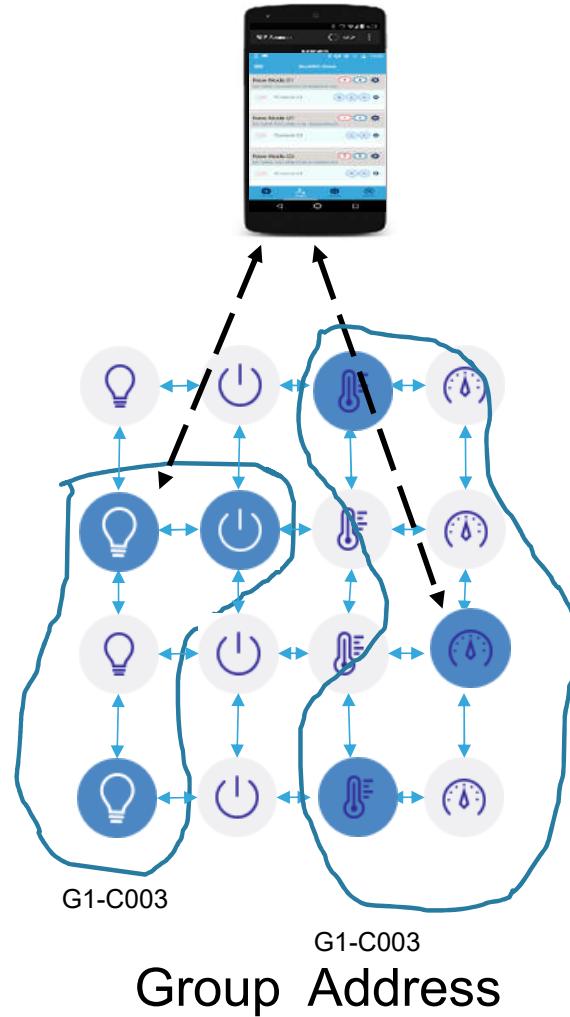
- 16比特数值
- 定义了这条消息的目的地
- 可以是一个单播地址、虚拟地址、分组地址或者全体广播地址
- 目标地址使用网络密钥加密

# Mesh Addresses

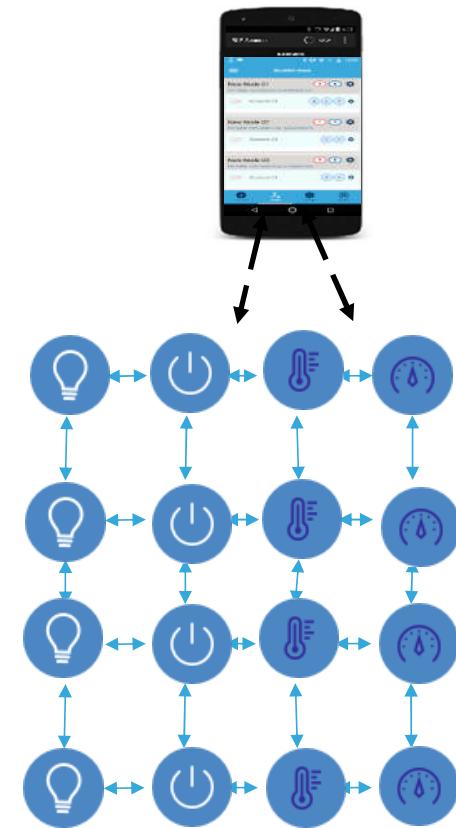
121



Unicast Address



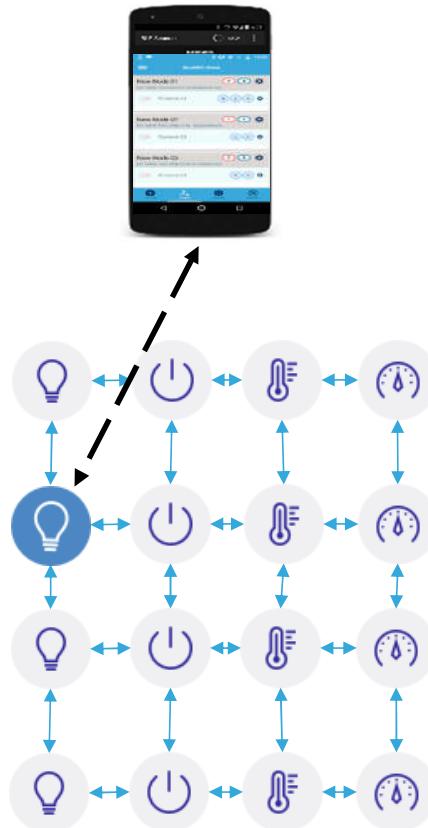
Group Address



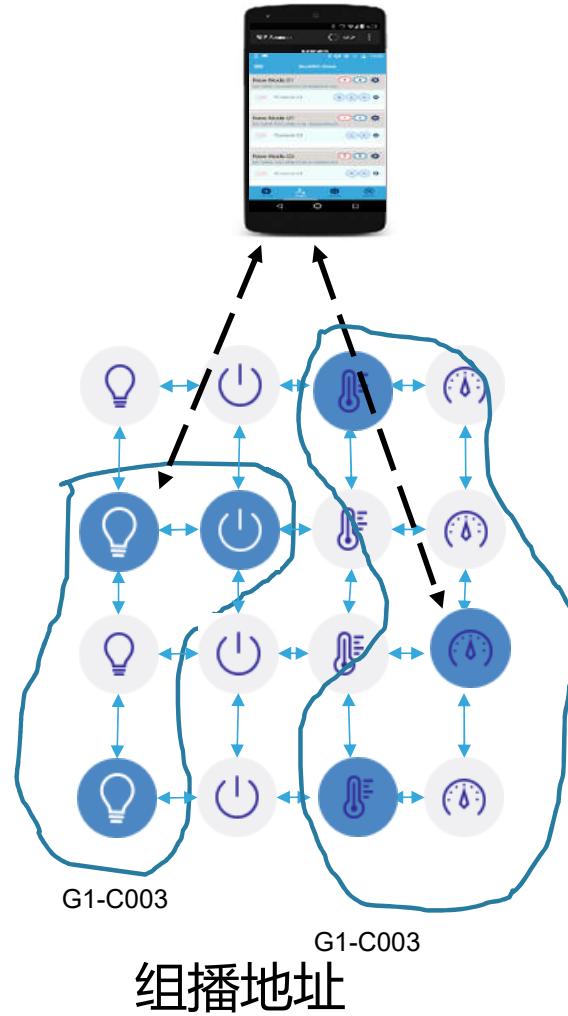
Broadcast

# Mesh 地址

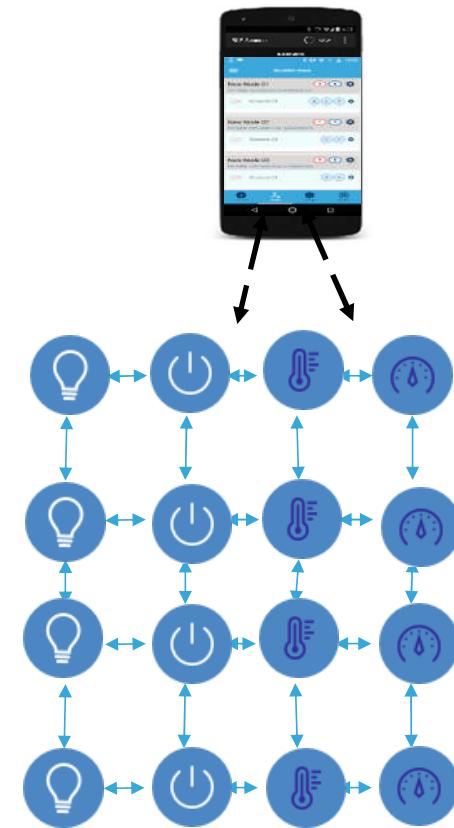
122



单播地址



组播地址



全体广播地址

# Mesh Addresses

123

Unassigned	0000000000000000	No address assigned (typically used when not publishing or subscribing)
Unicast	0xxxxxxxxxxxxxxx	Address of a mesh node in a network
Virtual	10xxxxxxxxxxxxxx	Virtual address of a single / group of devices
Group	11xxxxxxxxxxxxxx	Group address of a number of mesh nodes (typically used for publish and subscribe)
Broadcast	1111111111111111	Every node in a network

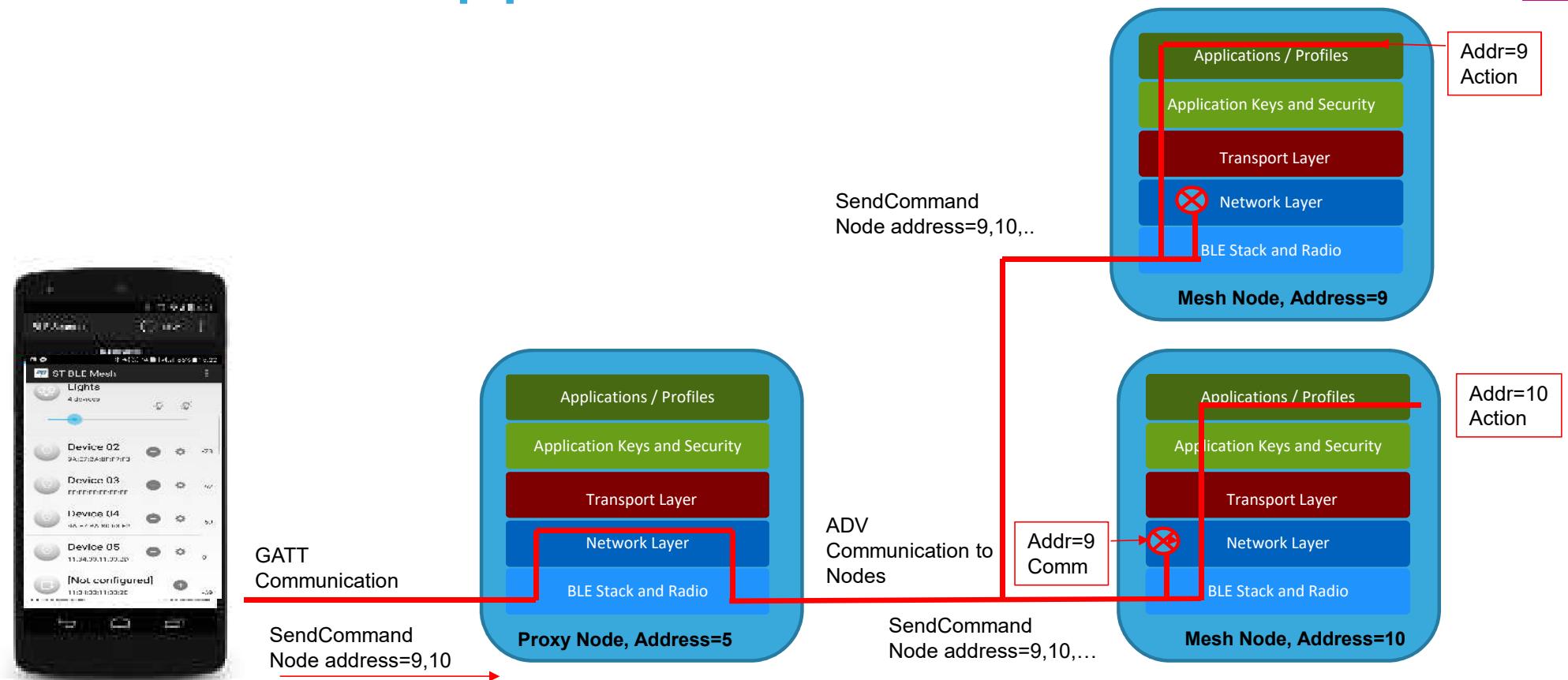
# Mesh 地址

124

未分配	0000000000000000	没有分配地址(无发布或订阅时被特定使用)
单一的	0xxxxxxxxxxxxxxx	网络里的一个mesh节点的地址
虚拟的	10xxxxxxxxxxxxxx	一个或一组设备的虚拟地址
分组的	11xxxxxxxxxxxxxx	一组mesh节点的地址 (发布或订阅时被特定使用)
全体的	1111111111111111	网络中的每一个节点

# Addr: App to Nodes Communication

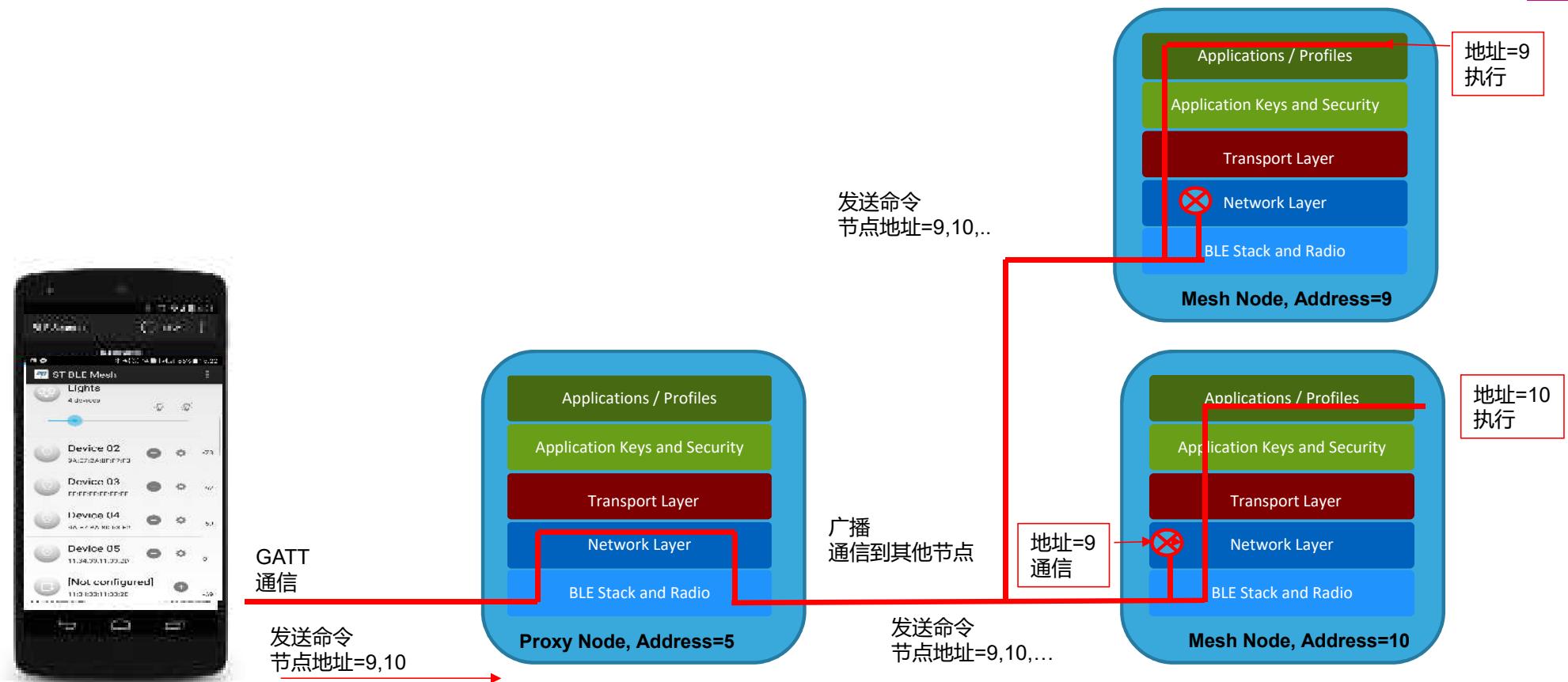
125



Note: Address= 5,9,10 are for indication purposes.

It doesn't necessarily mean Address=5 will always be Proxy Node

# 地址: APP和节点之间的通信

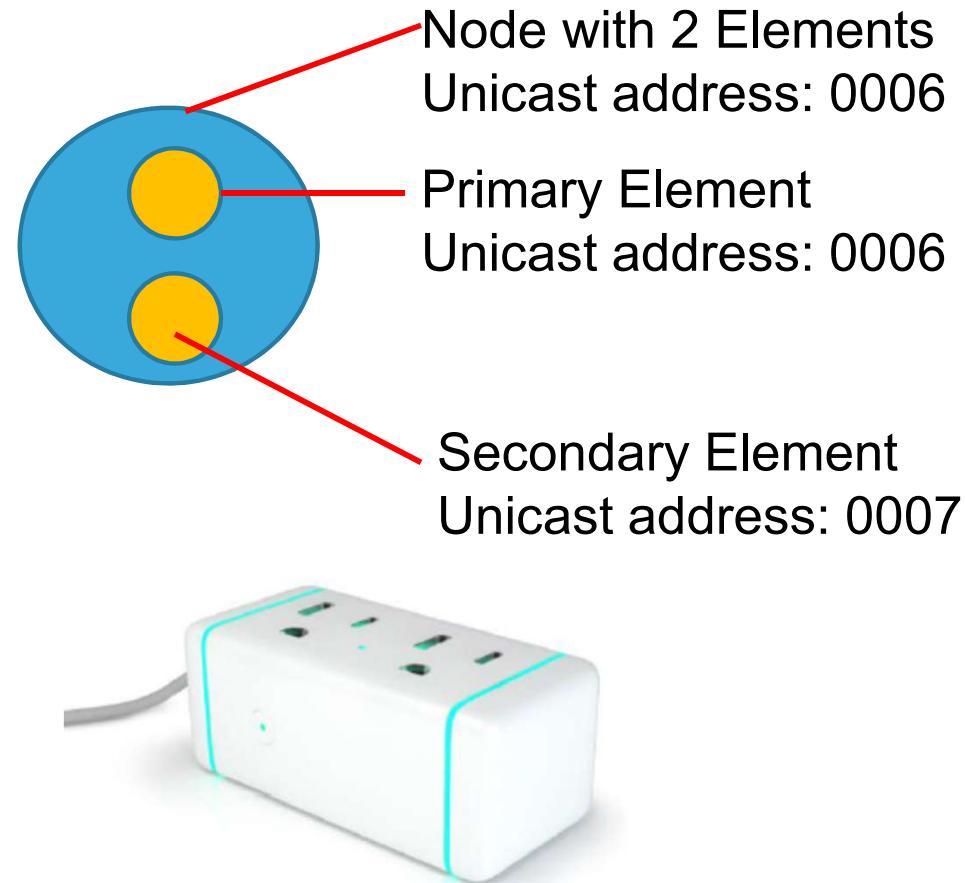


备注: 地址= 5,9,10 目的是用于指示.  
这并不一定意味着地址=5总是代理节点

# Elements in a Node

127

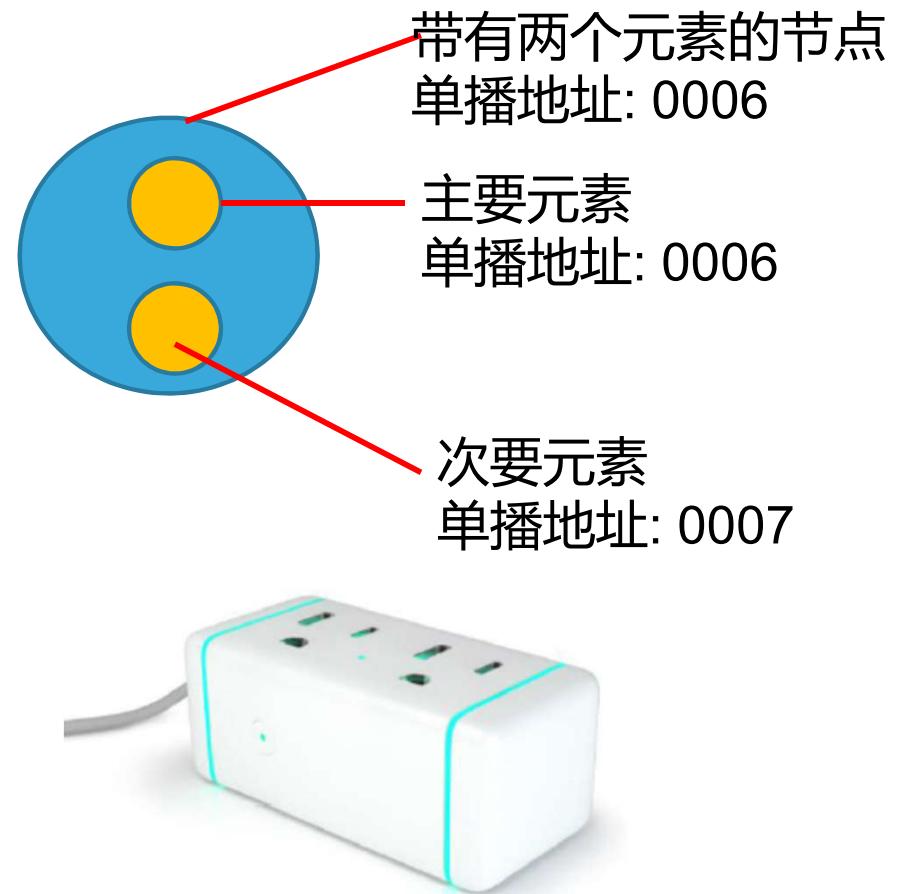
- Each node has atleast 1 Element called Primary Element
- Other elements are secondary elements
- Primary element has 1<sup>st</sup> Unicast address of node
- Each element has a GATT Bluetooth Namespace Descriptor value that helps identify which part of the node this element represents



# 一个节点中的元素

128

- 一个节点至少含有一个元素称之为主要元素
- 其他元素称之为次要元素
- 主要元素含有节点的第一单播地址
- 每个元素都有一个GATT的蓝牙命名空间描述符的值，可以帮助确定这个元素代表了节点的哪一部分



# Network PDU

129

- What does a mesh network message look like?



- **Transport PDU**

- 1 to 16 octets
- Contains information from lower transport layer
- Size of Transport PDU determined by CTL
- Transport PDU is encrypted using the Network Key

# 网络 数据包单元

130

- Mesh网络消息看起来像什么？



## • 传输 协议数据单元

- 1 到 16 个字节
- 包含了低传输层的信息
- CTL决定了传输PDU的大小
- 传输 PDU 时使用了网络密钥进行加密

# Network PDU

131

- What does a mesh network message look like?



- **NetMIC – Network Message Integrity Check Value**

- 32-bit or 64-bit value
- Authenticates the message as being sent by a mesh node that knows the network key
- If CTL is 0 (access message) NetMIC is a 32-bit value - Transport pdu limited to 16 octets
- If CTL is 1 (control message) NetMIC is a 64-bit value - Transport pdu limited to 12 octets

# 网络数据包单元

132

- Mesh网络消息看起来像什么？



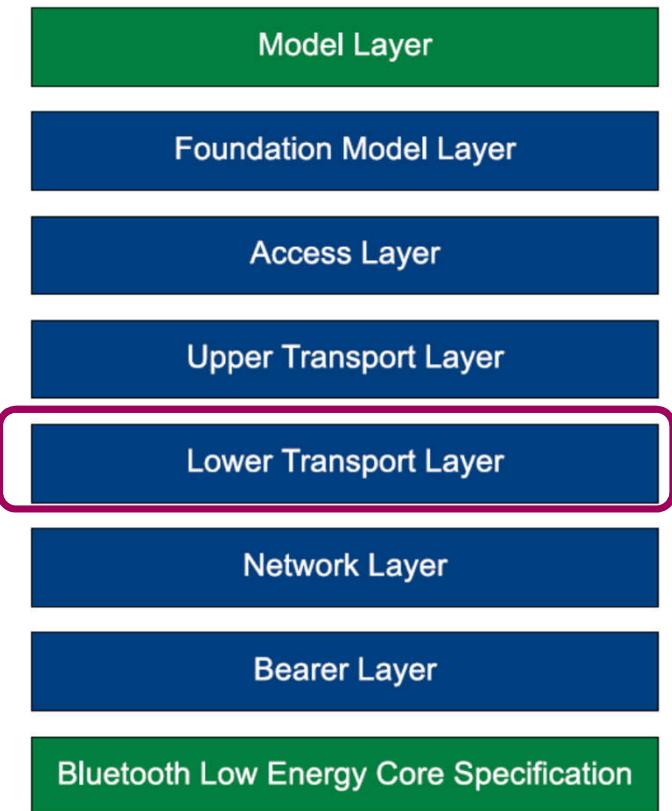
- **NetMIC – 网络消息完整校验值**

- 32位或64位值
- 认证符合网络密钥的节点发出的消息
- 如果 CTL 是0 (接入消息) NetMIC 是一个 32-bit 数值 – 传输 pdu 限制在16 字节
- 如果 CTL 是 1 (控制消息) NetMIC 是一个64-bit 数值 – 传输 pdu 限制在12 字节

# Lower Transport Layer

133

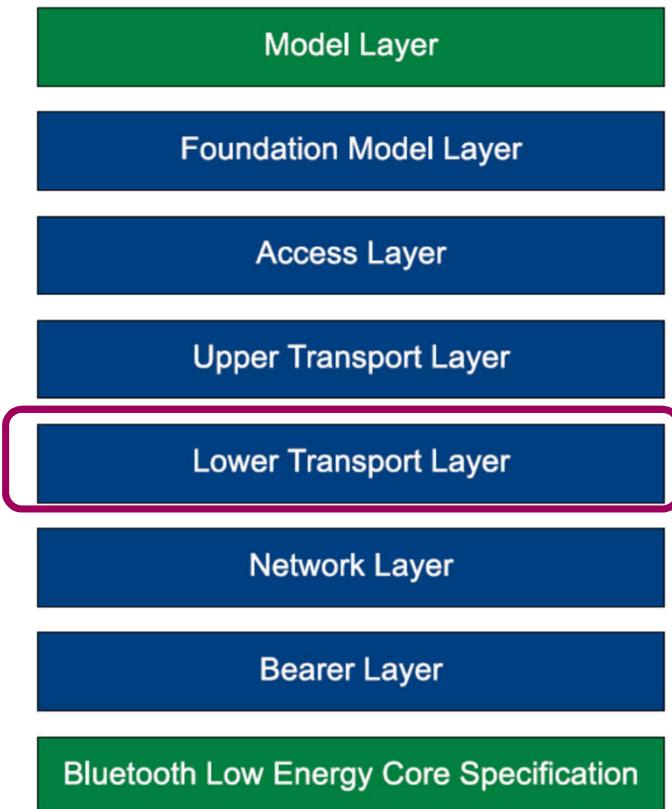
- Helps in segmentation and reassembly of PDUs
- Takes PDUs from upper transport layer and sends them to the lower transport layer on a peer device
- Performs segmentation and reassembly of longer PDUs



# 低层传输层

134

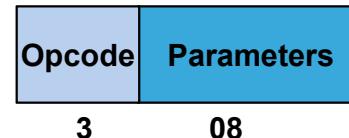
- PDUs的分割重组帮助
- 在对等设备上从高传输层获取PDUs发送到低传输层
- 对更长的PDUs进行分割和重组



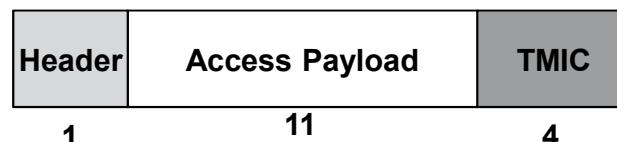
# Mesh packet structure: Unsegmented

135

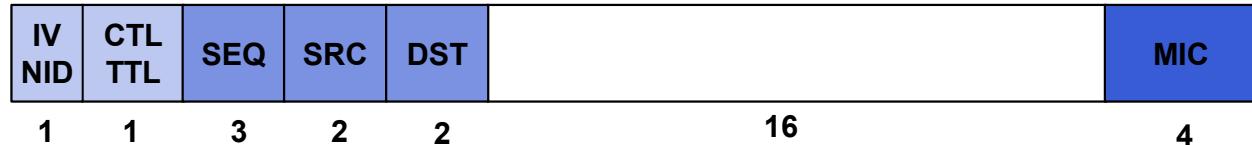
Model / Access layer



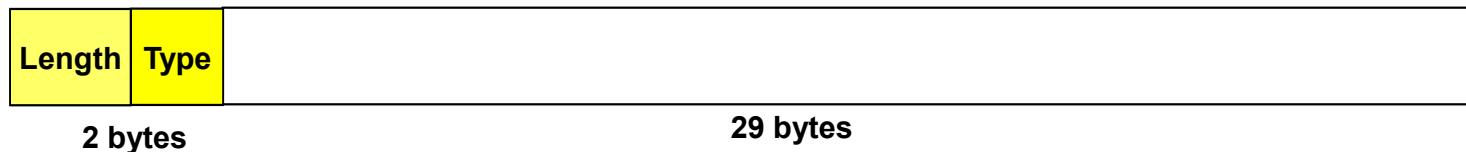
Upper / Lower Transport layer



Network layer



Bearer layer



16 octets available at NW layer, 11 at transport layer (including Trans Header & Trans MIC)

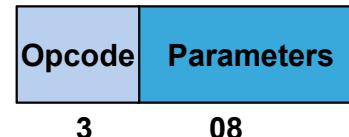
Opcode can be 1/2/3 octets. 1 & 2 octet opcodes are reserved by SIG.

3 Byte Opcode can be used as vendor specific opcode.

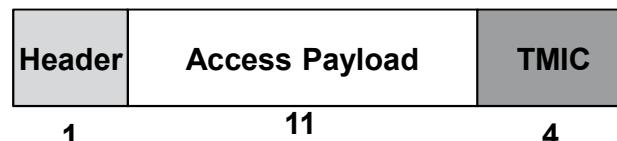
# Mesh 包结构: 未分段

136

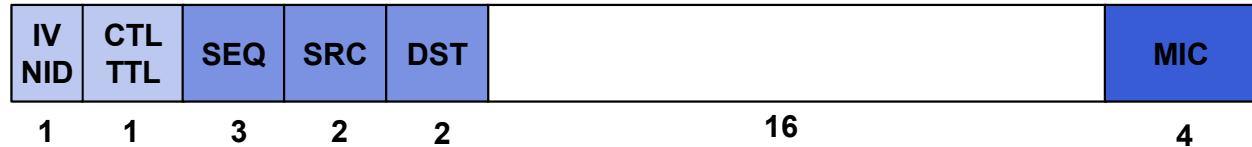
模型 / 接入层



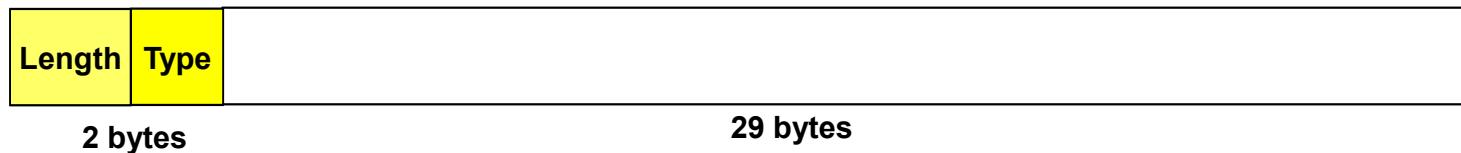
高 / 低 传输层



网络层



承载层

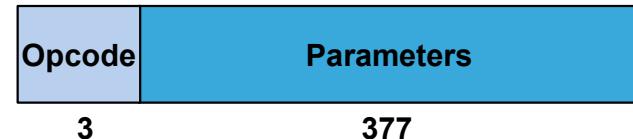


网络层有16可用字节, 11个在传输层 (包含传输头和传输MIC) 操作码可以是1/2/3 字节. 1 & 2 字节 操作码 是SIG组织预留的  
3 字节 操作码 可以被用作厂商特定操作码.

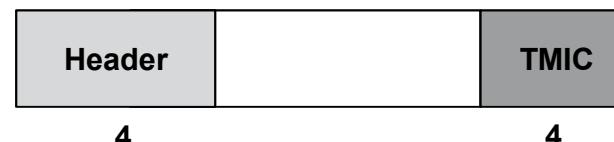
# Mesh packet structure: Segmented

137

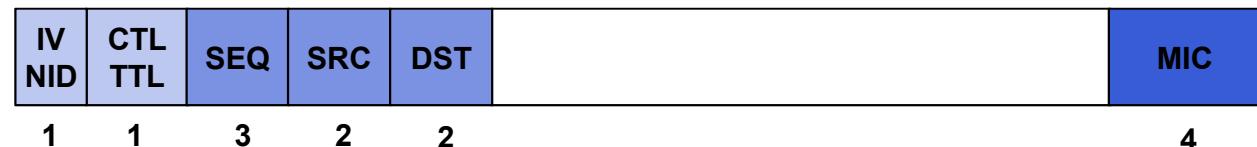
Model / Access layer



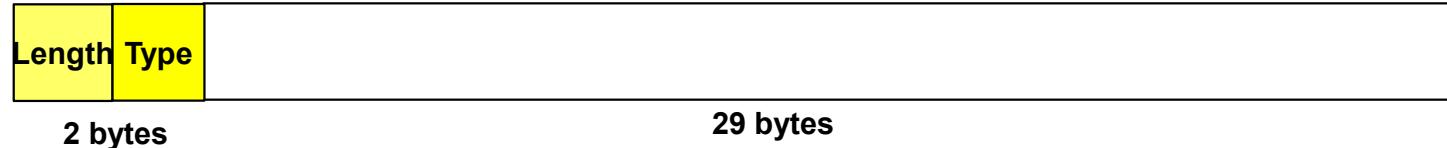
Upper / Lower Transport layer



Network layer



Bearer layer



When using segmented message, 12 bytes are available at transport layer payload (Trans MIC is not included here) Max 32 segments possible. However, Opcode and TransMIC is transferred in only once (in first and last).

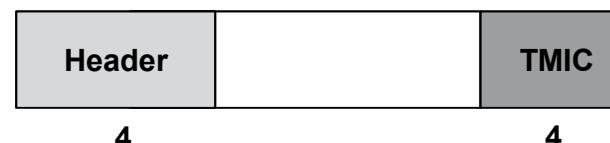
# Mesh 包结构: 分段

138

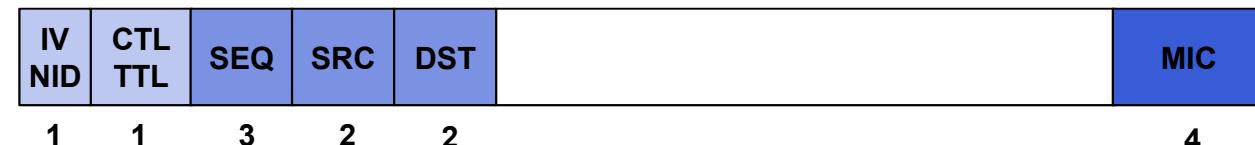
模型 / 接入层



高 / 低传输层



网络层



承载层



当使用分段消息时，传输层负荷有12可用字节（这里不包含传输MIC） 最大可达到32段。然而，操作码和传输MIC仅被传输一次（在开始和最后）。

# Message / Transport and Reassembly

139

UN SEGMENTED MESSAGE			SEGMENTED MESSAGE		
Opcode	Message type	Max size	Opcode	Message type	Max size
1 Octet/Byte	Special message	10 Octet	1Octet/Byte	Special message	379 Octet
2 Octet/Byte	Standard message	9 Octet	2 Octet/Byte	Standard message	378 Octet
3 Octet/Byte	Vendor specific message	8 Octet	3 Octet/Byte	Vendor specific message	377 Octet

- Transport layer provides 11 Octet/Bytes for unsegmented message

- The transport layer provides a mechanism of SAR capable of transporting up to 32 segments
- Opcode is sent in 1<sup>st</sup> packet
- Application MIC is transferred in last message

# 消息 / 传输和重组

140

未分段消息			分段消息		
操作码	消息类型	最大长度	Opcode	消息类型	最大长度
1 Octet/Byte	特殊消息	10 Octet	1Octet/Byte	特殊消息	379 Octet
2 Octet/Byte	标准消息	9 Octet	2 Octet/Byte	标准消息	378 Octet
3 Octet/Byte	厂商特定消息	8 Octet	3 Octet/Byte	厂商特定消息	377 Octet

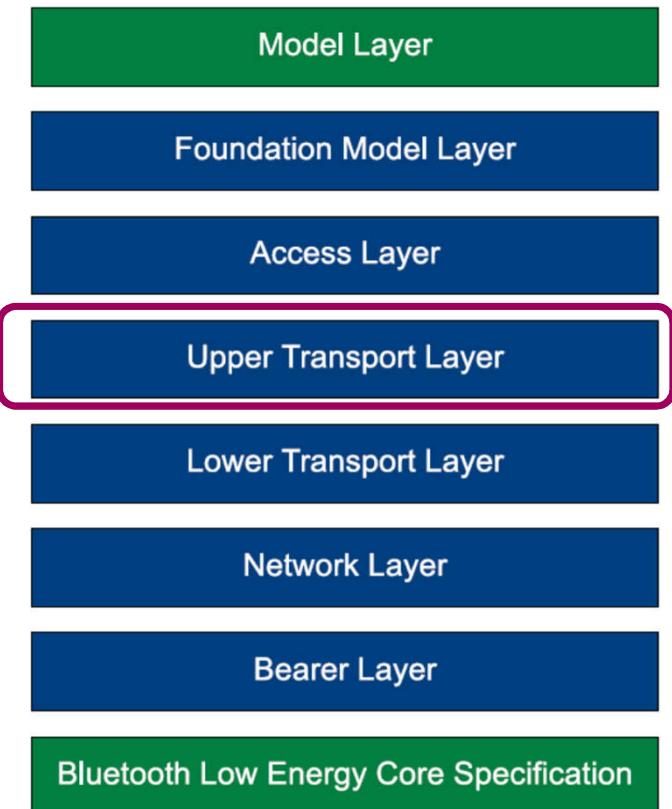
- 传输层给未分段消息提供11字节

- 传输层提供了SAR机制使得传输容量能够达到32段
- 操作码在第一包数据中被发送
- 应用MIC在最后一条消息中被传出

# Upper Transport Layer

141

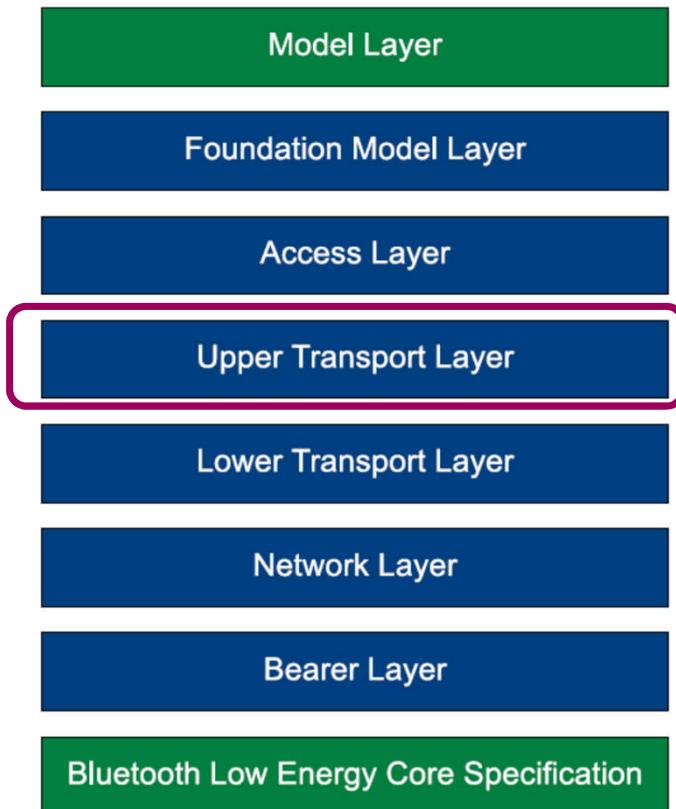
- Helps in encryption, decryption and authentication
- Responsible for encryption, decryption and authentication of application data passing to and from the access layer
- Handles Transport control messages, which are internally generated and sent between the upper transport layers on different peer nodes



# 上层传输层

142

- 加解密和认证帮助
- 负责对来自接入层的应用数据进行加解密和认证
- 处理在不同的对等节点的高传输层内部产生并发送的传输控制消息  
(处理不同的对等节点的传输控制消息，该消息由他们的高传输层内部产生并发送的)



# Upper Transport Layer

143

- Helps in encryption, decryption and authentication

- Access Messages

- Defines the format of application data
- Application Key or Device Key used to encrypts and authenticates message
- Application Key
  - Used for model data
- Device Key
  - Used for configuration and key management
  - Pairwise key
  - Only known by provisioner and a single device

- Control Messages

- Used to establish and manage friendship and heartbeats operations
- Friendship
  - Used to enable low power operation
- Heartbeat
  - Helps in detecting if device is still alive
  - Can be used to determine network topology

# 上层传输层

144

## • 加解密和认证帮助

### ❖ 接入消息

- 定义了应用数据的格式
- 应用密钥或者设备密钥用于加密和认证信息
- 应用密钥
  - 用于模型数据
- 设备密钥
  - 用于配置和密钥管理
  - 配对密钥
  - 只被启动配置设备和单一设备知晓

### ❖ 控制消息

- 用于创建和管理友谊和心跳操作
- 友谊
  - 用于使能低功耗操作
- 心跳
  - 帮助检测设备是否仍然工作
  - 可被用于终结网络拓扑

# Upper Transport PDU

145

- TransMIC size
  - Segmented messages use SZMIC
  - Can be 32-bit or 64-bit TransMIC
  - Unsegmented messages always 32-bit

Access Payload

TransMIC

Encrypted

# 上层传输协议数据单元

- TransMIC 大小
  - 分段消息使用SZMIC
  - 可以是32-bit或64-bit TransMIC
  - 未分段信息始终是32-bit

Access Payload

TransMIC

Encrypted

# Friendship/Low Power Nodes

- Problem with all time active nodes
  - Must listen all the time
  - Listening Uses Lots Of Power
  - Duty Cycles must be near 100%
  - Some devices can use coin cell batteries or energy harvesting for some application
  - But still need to stay synchronized for security updates to be part of network
- Low Power Feature can help
  - Duty cycles can be very low to save power
  - Low power nodes (LPN) find a friend
  - Friend node cache all incoming messages
  - LPNs then poll friend for messages
  - friends reply with messages

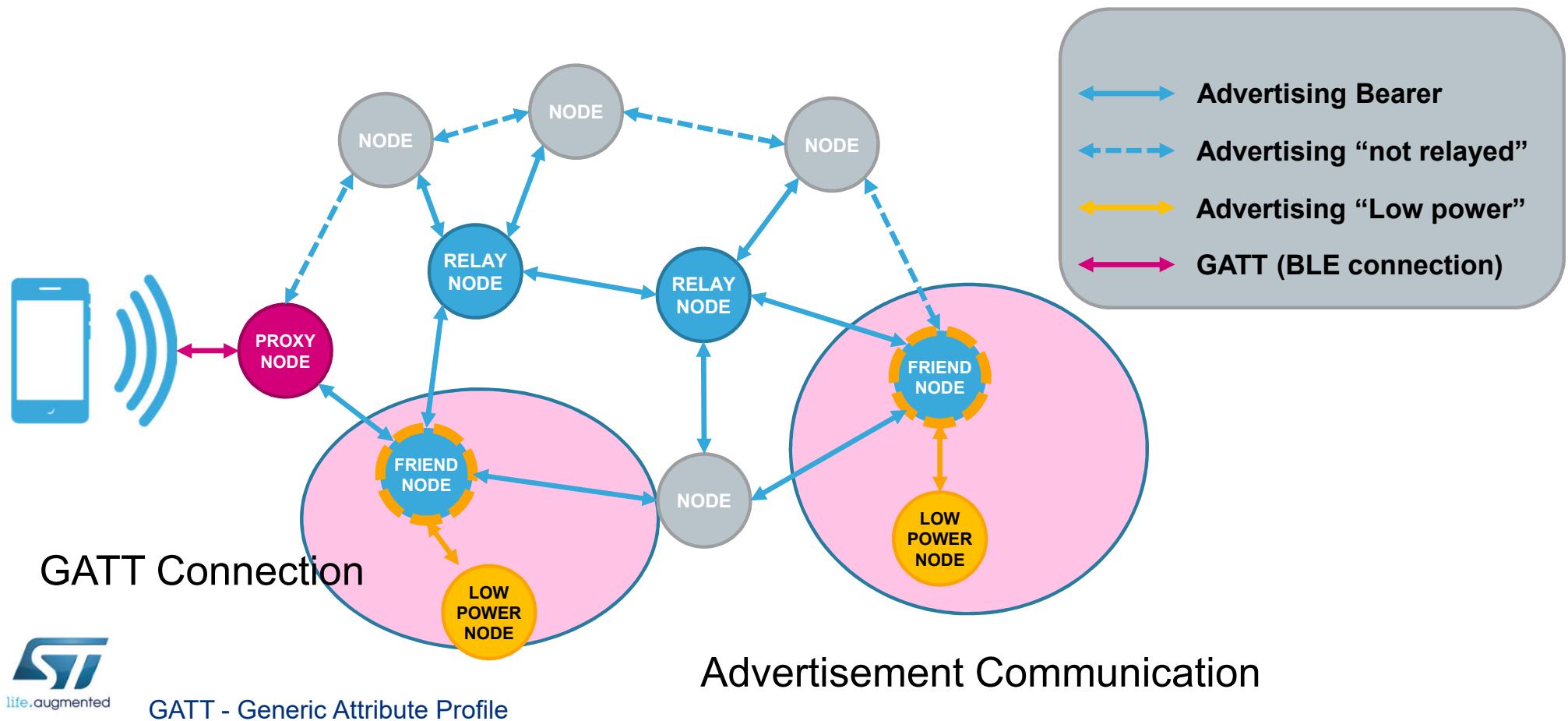
# 友谊/低功耗节点

- 始终工作的节点存在的问题
  - 必须始终监听
  - 监听消耗巨大能量
  - 工作周期必须接近100%
  - 在某些应用中一些设备可以使用纽扣电池或者能量采集供电
  - 但是仍然需要保持安全更新同步否则将会退出网络

- 低功耗特性可以有所帮助
  - 工作周期可以很低以便省电
  - 低功耗节点会发现一个朋友节点
  - 朋友节点会缓存所以接收的信息
  - 低功耗节点会轮询朋友节点获取信息
  - 朋友节点会回复信息

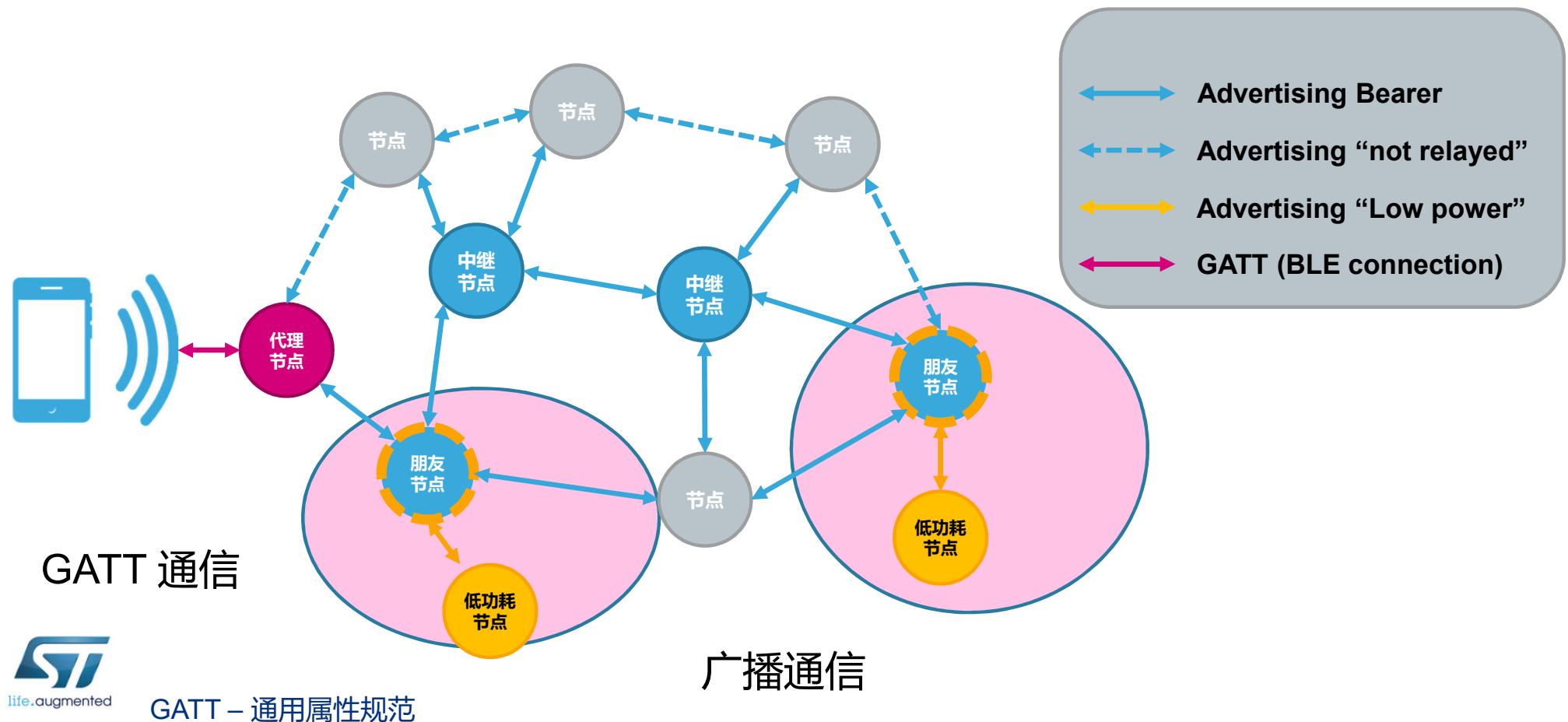
# BLE Mesh Topology: Friend & LP Nodes

149



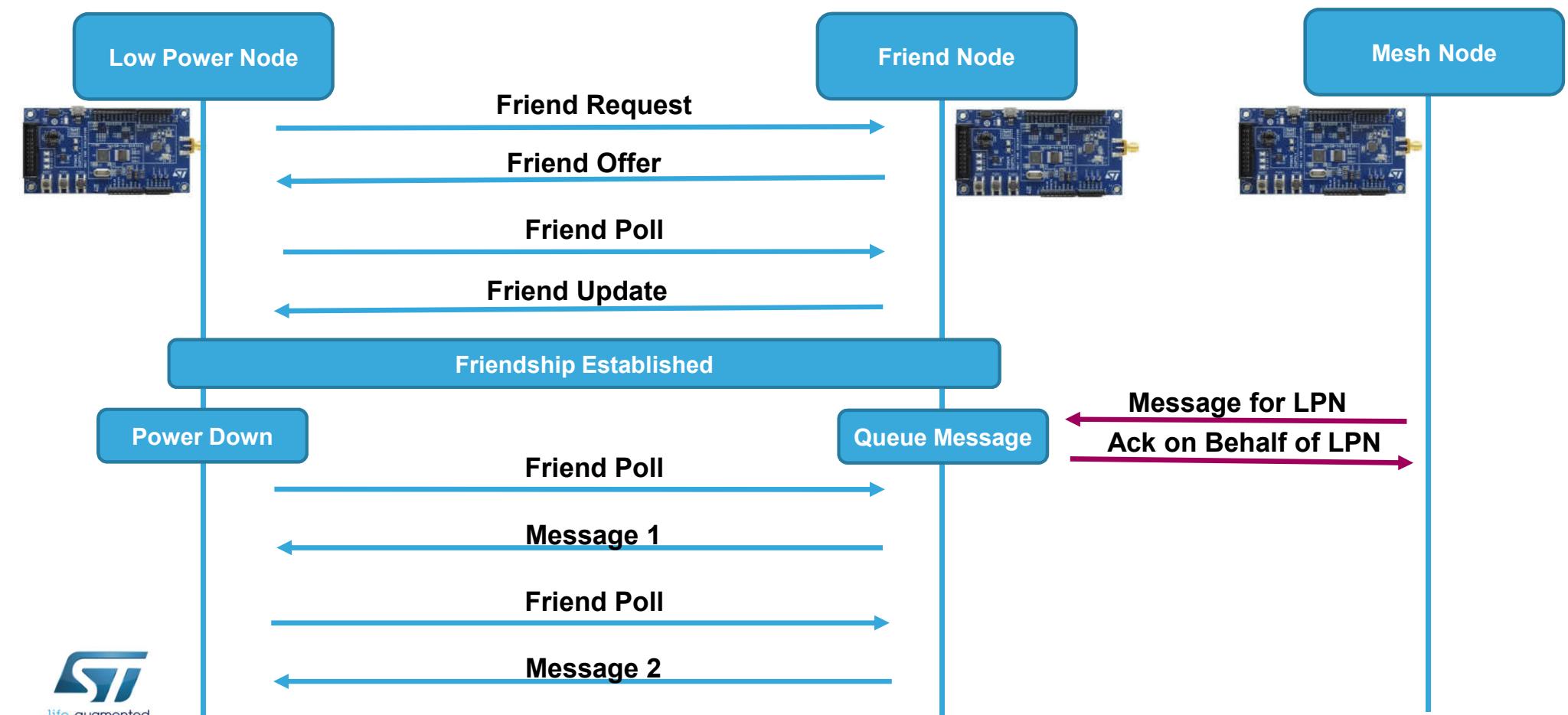
# BLE Mesh 拓扑: 朋友 & 低功耗节点

150



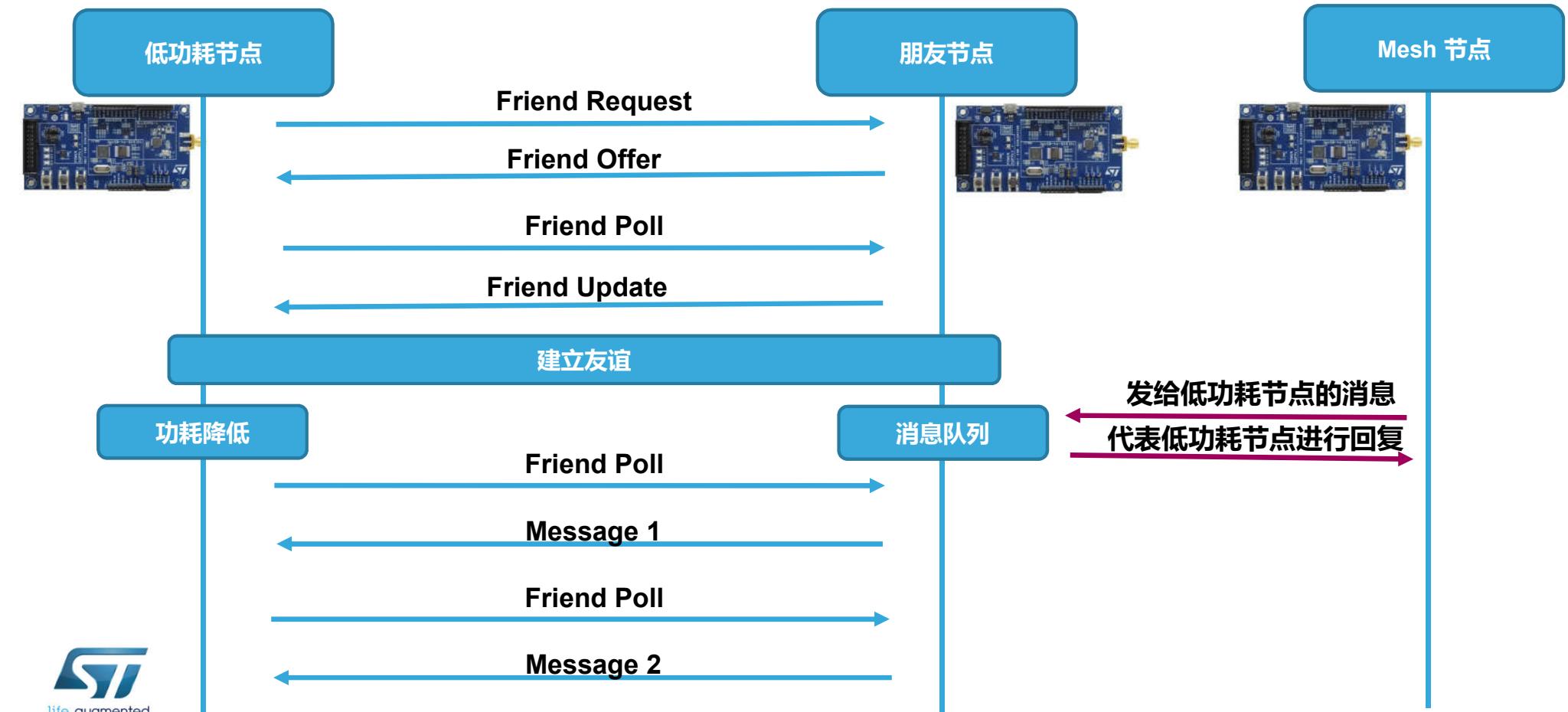
# Friendship Messaging

151



# 友谊信息

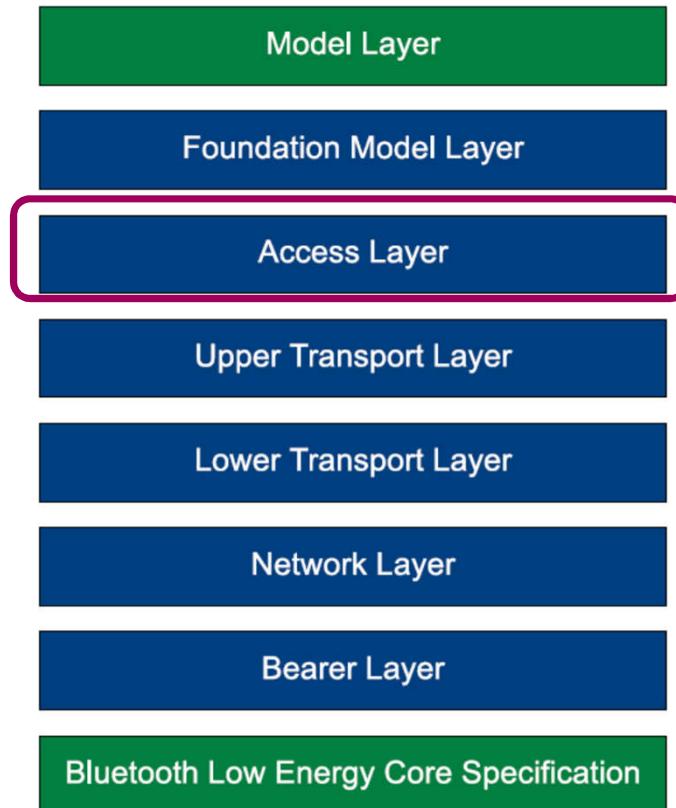
152



# Access Layer

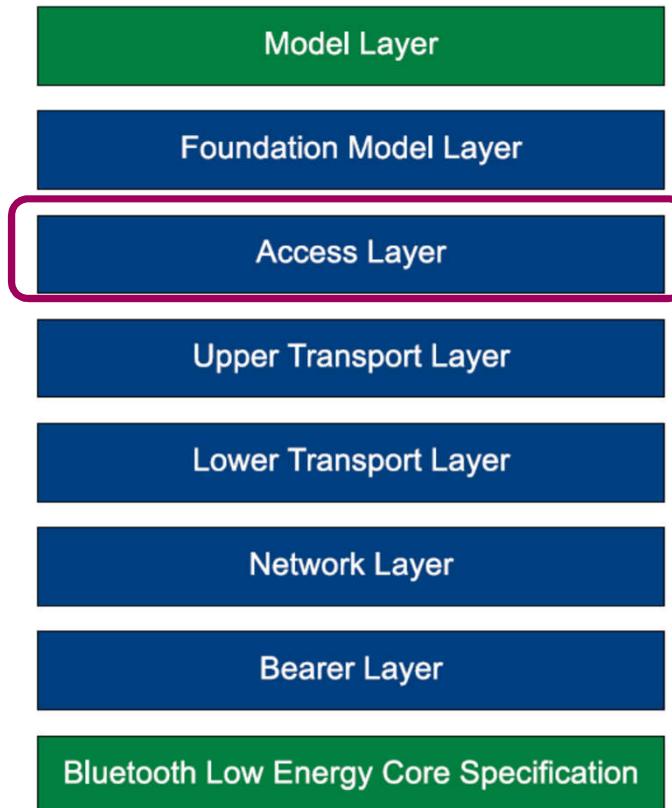
153

- Helps models access mesh networks
- All communication within a mesh network is accomplished by sending messages
- Messages operate on states
- Defined set of messages that a server supports and a client may use to request a value of a state or to change a state.



- Mesh网络中模型读取帮助

- 在一个mesh网络里面所有的通信都是靠发送消息完成的
- 不同状态下的消息处理
- 定义了一组消息，服务器支持此类消息并且客户端可以使用此类消息去请求状态值或者改变状态



# Access Message

- Opcodes
  - One, two, or three octet opcodes
    - 0xxxxxx - Bluetooth SIG defined application opcodes.
    - 10xxxxxxxx - Bluetooth SIG defined application opcodes.
    - 11xxxxxxxx yyyy-yyyy - Manufacturer-specific opcodes.
  - Three octet opcodes are for vendors
    - each vendor given 64 opcodes (x)
    - identified with a company\_id (y)

Operation Code	Application Parameters
Opcode	Parameters
1, 2 or 3 octets	379 octets with 1 octet opcode

**Access payload may be sent in up to 32 segments of 12 octets each. This implies that the maximum number of octets is 384 including the TransMIC.**

- 操作码

- 1个, 2个或3个字节的操作码
  - 0xxxxxxx – 蓝牙SIG定义的应用操作码
  - 10xxxxxxxx xxxxxxxx – 蓝牙SIG定义的应用操作码
  - 11xxxxxxxx yyyy yyyy yyyy yyyy – 制造商特定操作码
- 厂商专用三字节操作码
  - 提供64字节给每个厂商 (X)
  - 使用公司ID去定义 (Y)

操作码	应用参数
Opcode	Parameters
1, 2 or 3 octets	379 octets with 1 octet opcode

读取的负荷最高能以32段（每段12字节）被发送。这意味着包含TransMIC的最大字节数是384。

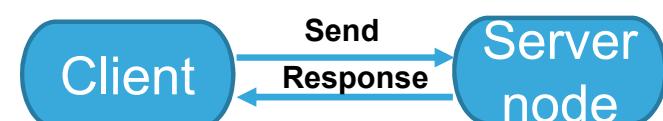
# Unacknowledged & Acknowledged Messages

157

- Unacknowledged message
  - No response to an unacknowledged message
  - Not possible for the sending element to determine if that message has been delivered or processed
  - Peer nodes should be notified about a state change using a status message
  - Status message is sent to the publish address of a model associated with the changed state



- Acknowledged message
  - Acknowledged by each receiving element by responding to that message
  - Response is typically a status message
  - Retransmit the message if a response is not received within a defined time period
  - Message timeout should be set to a minimum of 30 seconds
  - Not recommended to send an acknowledged message to the all-nodes address



# 未经确认的 & 经确认的消息

- 未经确认的消息

- 对未经确认的消息没有回应
- 如果消息已经被传递或处理，不太可能发送元素去终结
- 对等节点会被状态消息通知状态变化
- 状态消息被发送到模型的发布地址，该模型与改变的状态相关联

- 经确认的消息

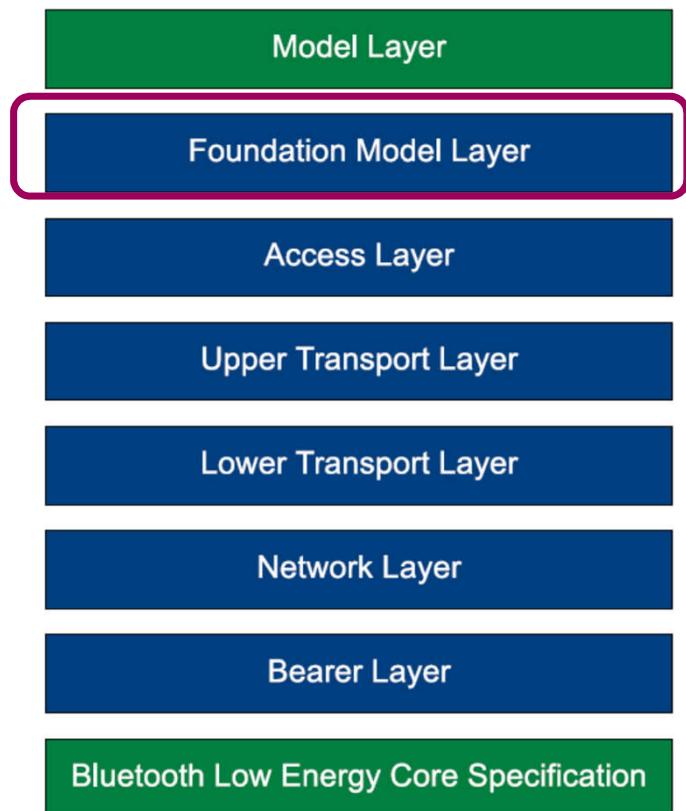
- 每个接收元素通过回应消息进行确认
- 回应是典型的状态消息
- 在规定时间周期内没有收到消息会进行重传
- 消息超时时间最小应被设置为30秒
- 不推荐向所有节点地址发送确认消息



# Foundation Model Layer

159

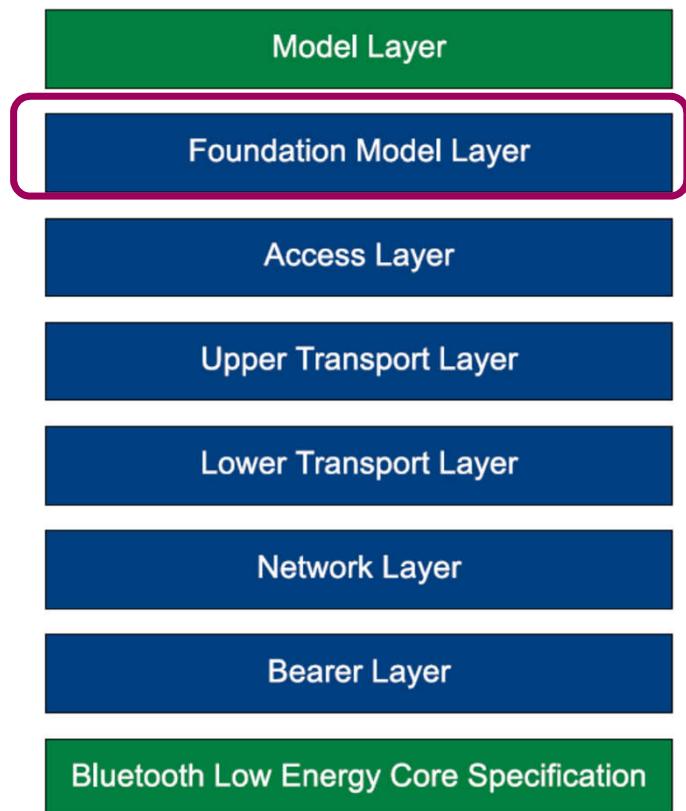
- Helps configuration and management of a mesh network
- Responsible for implementation of models concerned with the configuration and management of a mesh network



# 基础模型层

160

- Mesh网络的配置管理帮助
- 负责模型的实现，该模型与mesh网络的配置和管理相关



# Root Models

161

- Configuration Server model
  - Used to represent a mesh network configuration of a device
  - Supported by a primary element and shall not be supported by any secondary elements
  - Uses the device key for application-layer security
- Configuration Client model
  - Used to represent an element that can control and monitor the configuration of a node

- Health Server model
  - Used to represent a mesh network diagnostics of a device.
  - Shall be supported by a primary element and may be supported by any secondary elements
  - Uses the application key for application-layer security
- Health Client model
  - Used to represent an element that can monitor the health messages of a node

- 配置服务器模型

- 用于代表一个设备的mesh网络配置
- 由主要元素支持并且不被其他任何次要元素支持
- 使用设备密钥保证应用层安全

- 配置客户端模型

- 用于代表能够控制和监听节点配置的元素

- 健康服务器模型

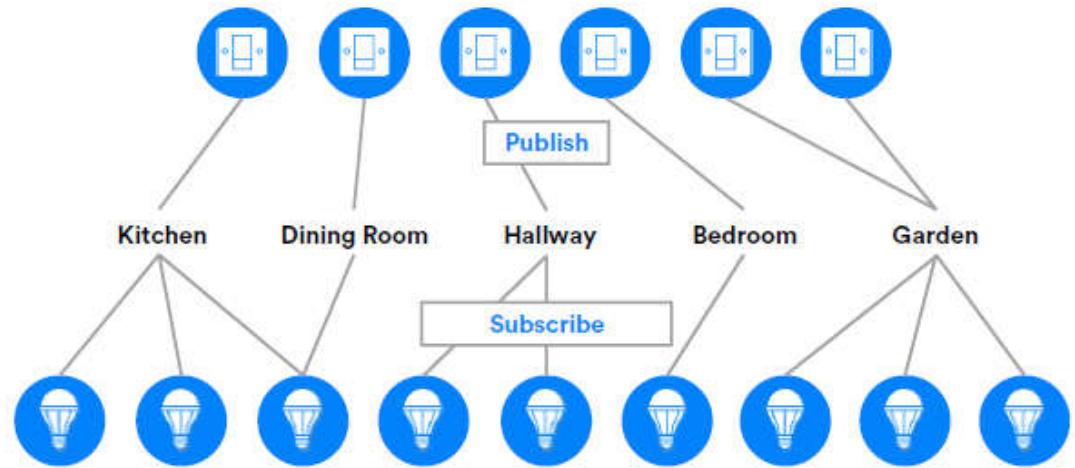
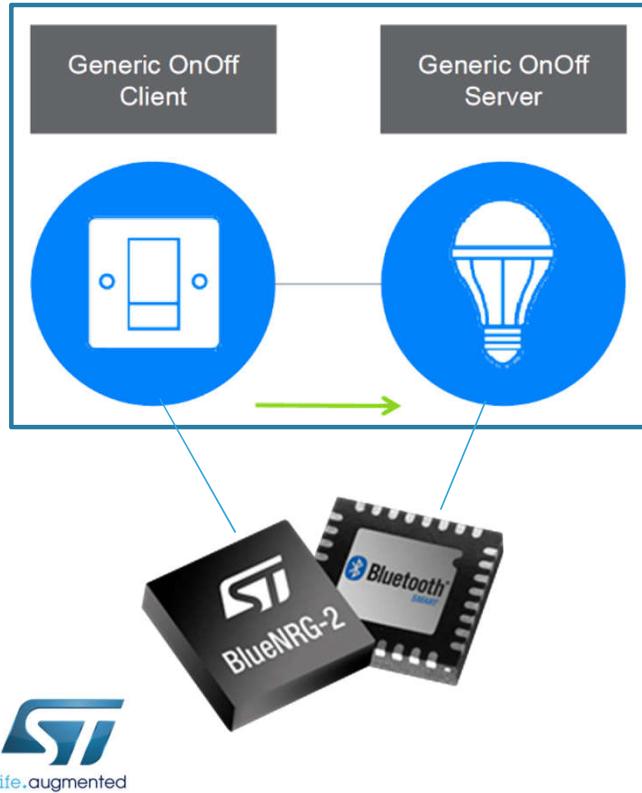
- 用于代表一个设备的mesh网络诊断
- 应当被主要元素支持并可能被一些次要元素支持
- 使用应用密钥保证应用层安全

- 健康客户端模型

- 用于代表能够监听节点健康消息的元素

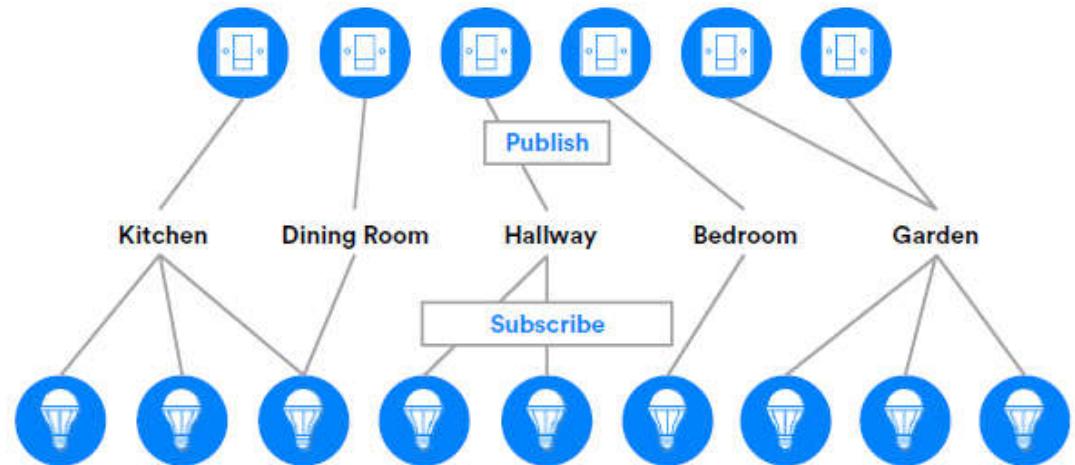
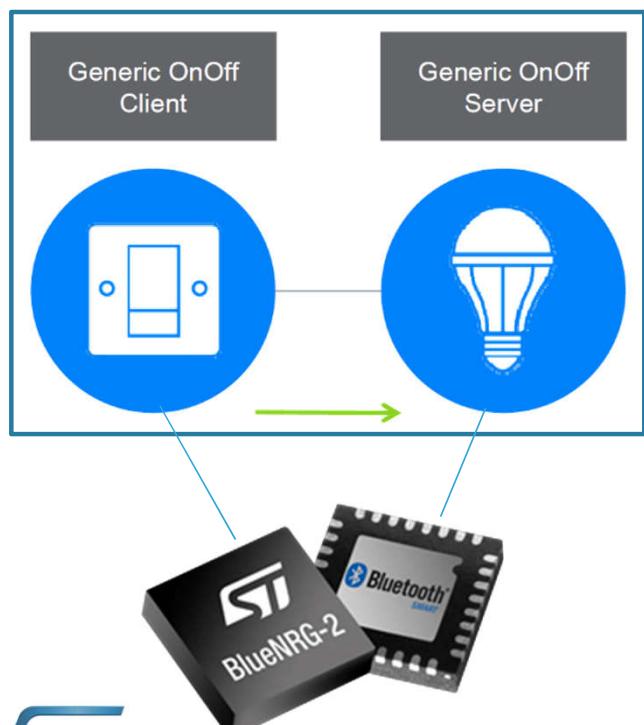
# Mesh Messaging Model

## Publish and Subscribe



Example: **client device** (switch) can post messages and **server device** (light bulb) can be notified about new command arrival.

# Mesh 消息模型 发布和订阅



示例: 客户端设备 (开关) 可以投送消息并且服务器设备 (照明灯泡) 会被通知新命令的到达

# Publish & Subscribe

165

- Publish
  - Model state is published to a single DST
  - Can be a unicast / virtual / group address
- State is published
  - Periodically
  - When reliable message is received
  - When state has changed on its own
  - When state was changing
- Subscribe
  - Nodes subscribe to dst addresses
  - A node can subscribe to multiple addresses
- Plug and Play
  - Matching publish and subscribe addresses is the way that mesh network is configured
  - Allows replacement of devices without having to reconfigure all other nodes

# 发布 & 订阅

166

- 发布

- 模型状态被发布到一个单一的目标地址
- 可以被单播、虚拟广播、组播

- 发布的状态

- 周期性地
- 当收到可靠消息时
- 当状态自己发生变化后
- 当状态正在改变时

- 订阅

- 节点订阅到目标地址
- 一个节点能够订阅到多个地址

- 即插即用

- mesh网络是通过匹配发布和订阅地址的途径被配置的
- 允许不经重新配置其他节点即可替换设备

# Features of Node

- Relay Feature
  - Receive and retransmit mesh messages over the advertising bearer to enable larger networks
- Proxy Feature
  - Receive and retransmit mesh messages between GATT and advertising bearers
- Low Power feature
  - Operate within a mesh network at significantly reduced receiver duty cycles only in conjunction with a node supporting the Friend feature
- Friend feature
  - Helps a node supporting the low power feature to operate by storing messages destined for those nodes.

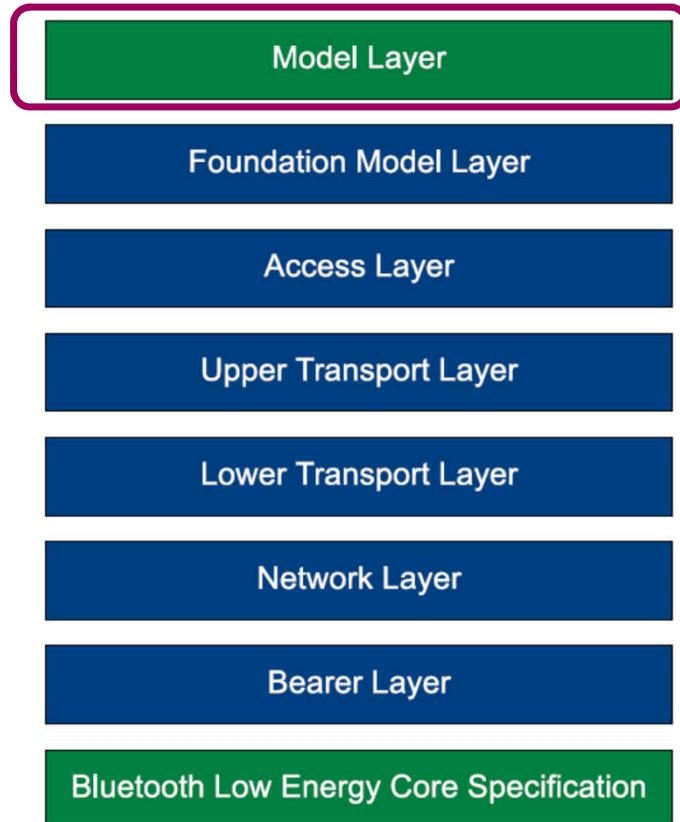
# 节点特性

- 转播特性
  - 通过广播承载层接收和转发mesh消息来扩大组网规模
- 代理特性
  - 在GATT和广播承载层之间接收和转发mesh消息
- 低功耗特性
  - 在mesh网络里，通过只在节点（支持友邻特性）连接时才进行操作的方式，显著地降低了接收节点的工作周期
- 友谊特性
  - 帮助支持低功耗特性的节点存储发往他们的消息

# Model Layer

169

- Defines the basic functionality of a node
- Model defines
  - Required states
  - Messages that act upon states
  - Associated behavior
- A node may include multiple models
- Uses a client-server architecture communicating with a publish-subscribe paradigm



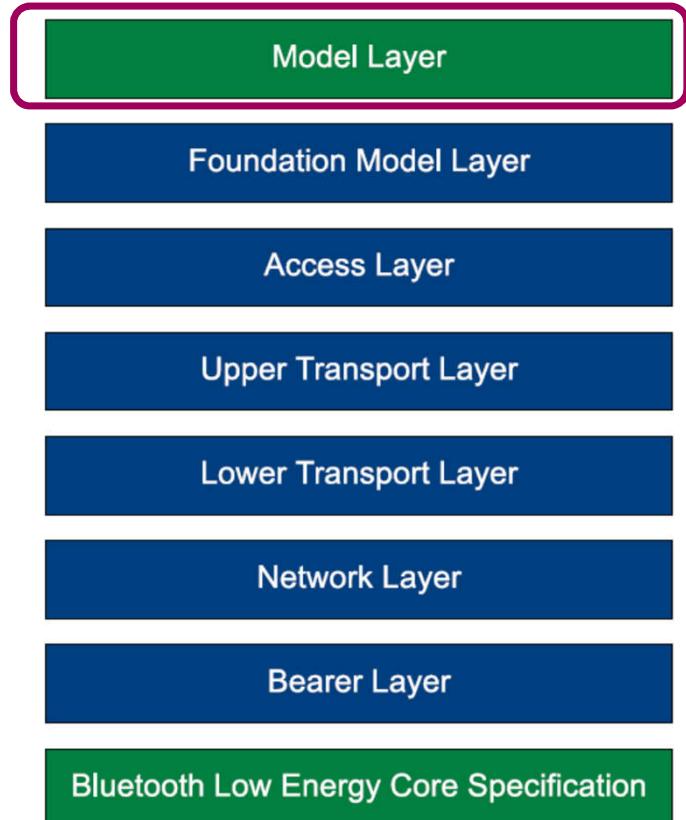
- 定义了节点的基础功能

- 模型定义

- 请求状态
- 根据状态动作的消息
- 相关行为

- 一个节点可能包含多种模型

- 使用客户端-服务器的架构，通过发布-订阅的形式进行通信



# Types of Model

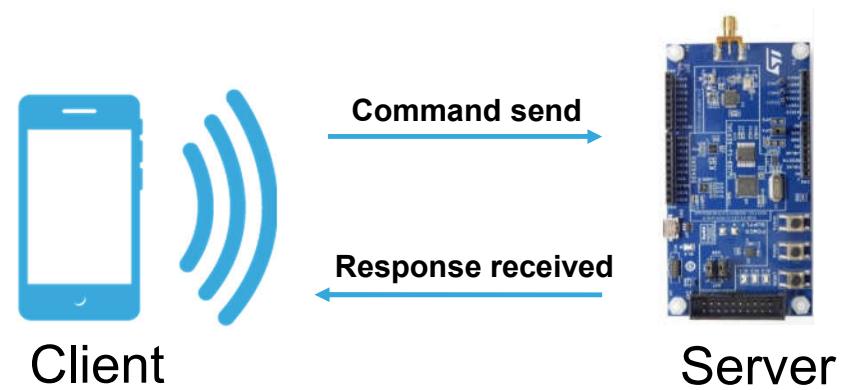
171

- Server Model

- Defines a set of mandatory messages that it can transmit or receive to set required behavior

- Client Model

- A client model defines a set of messages (both mandatory and optional) that a client uses to request, change, or consume corresponding server states. It does not have state.

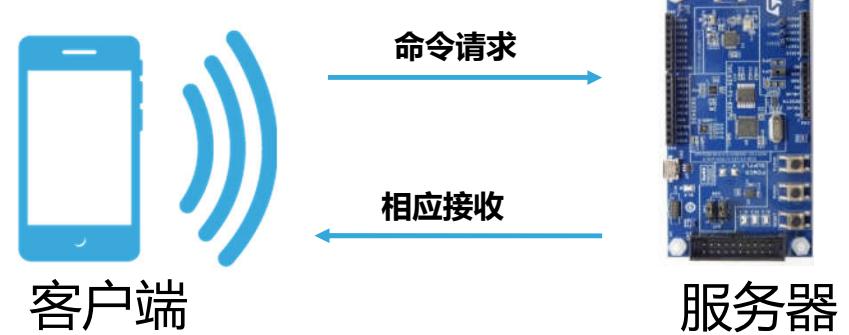


- 服务器模型

- 定义了一套强制的消息，能够通过发送或接受来设置请求的行为

- 客户端模型

- 一个客户端模型定义一套消息（包括强制的和可选的），一个客户端可以使用这些消息进行请求，改变，或者销毁相应的服务器状态。客户端本身没有状态。

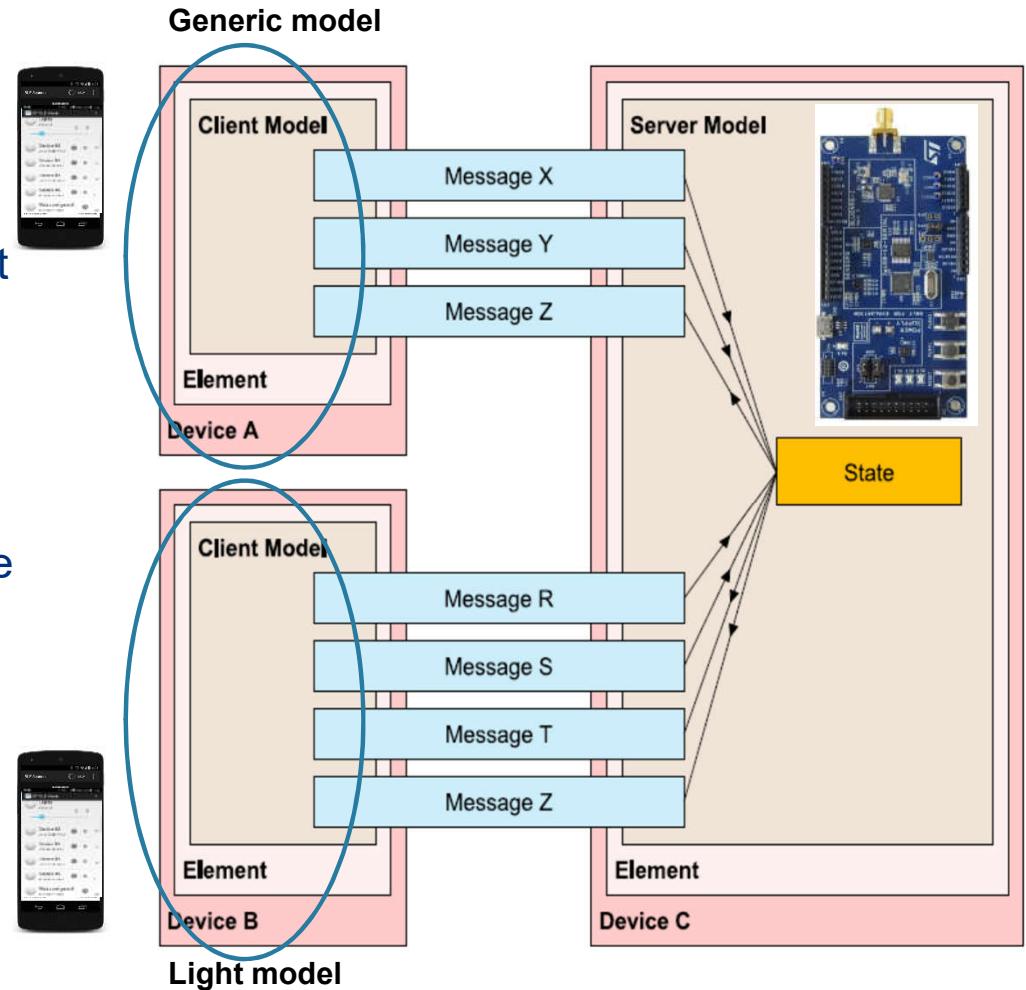


# Model Example

173

- Multiple Server Model Implementation

- Node of Bluetooth network can support multiple user models
- Message X,Y,Z are supported by one particular client device A along with node C
- Message R,S,T,Z are supported by one particular client device B along with node C

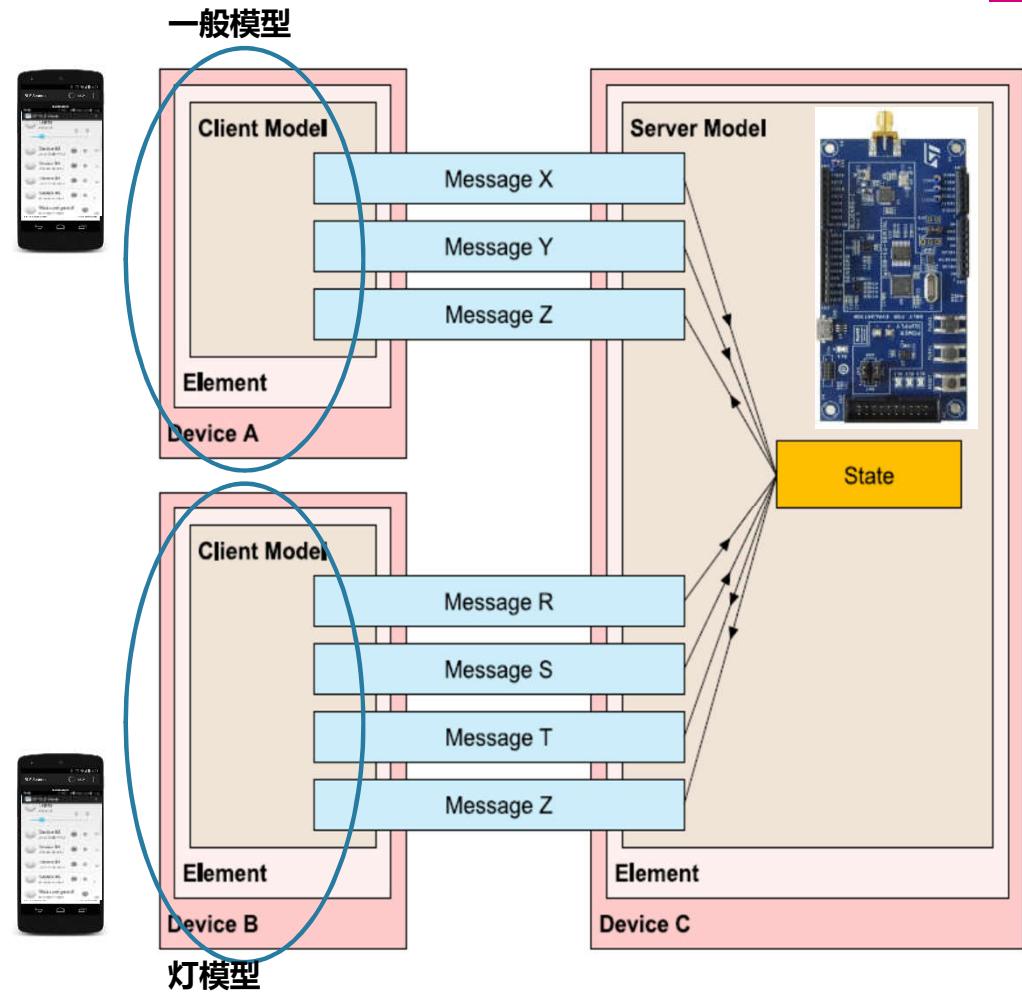


# 模型示例

174

## • 多重服务器模型的实现

- 蓝牙网络节点能够支持多重用户模型
- 消息X,Y,Z是被特殊客户端设备A以及节点C支持的
- 消息R,S,T,Z是被特殊客户端设备B以及节点C支持的



# Model Messages

175

- Types of Model Messages

- Set Reliable

- This message is used to send message to a node expecting status or response from that node

- Set Unreliable

- This message is used to send message to a node but not expecting status or response from that node

- Get Status

- This message is used to get the response from the node



Set Reliable



Set Unreliable



Get Status

# 模型消息

176

- 模型消息种类

- 设置为可信赖

- 这个消息被用于发送到一个节点，并希望该节点回复状态或者给出响应

- 设置为不可信赖

- 这个消息被用于发送到一个节点，并不一定希望该节点回复状态或者给出响应

- 获取状态

- 这个消息被用于获取节点响应



设置为可信赖



设置为不可依赖



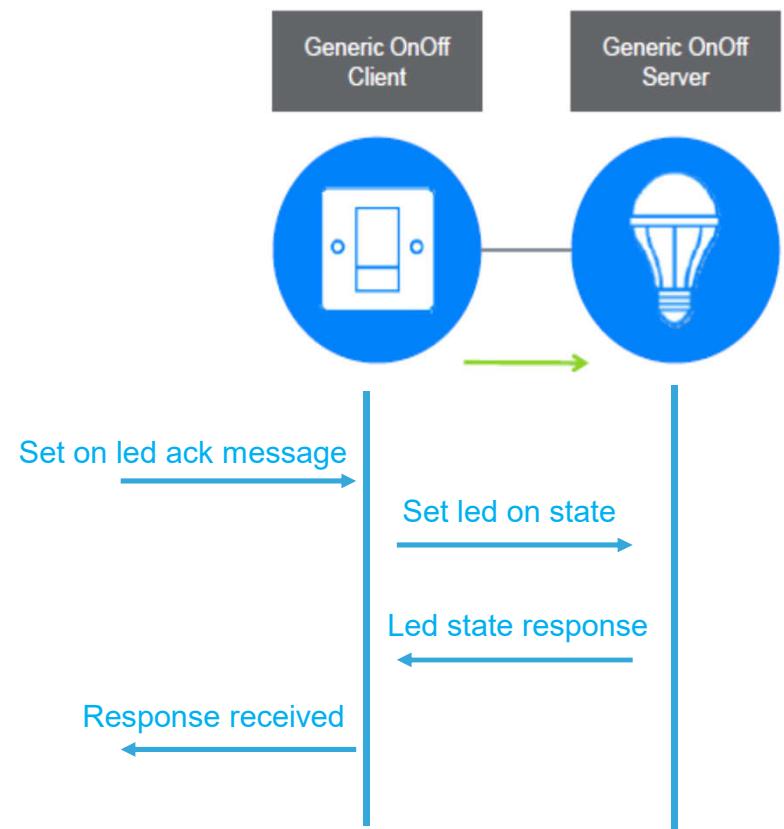
获取状态

# SIG or Vendor Models

177

- Models may be defined and adopted by Bluetooth SIG and may be defined by vendors
- SIG adopted model
  - Models defined by Bluetooth SIG
  - Models are identified by unique identifiers is 16 bits
- Vendor Models
  - Models defined by vendors
  - SIG adopted models, or 32 bits, for vendor model

## Generic Model Example

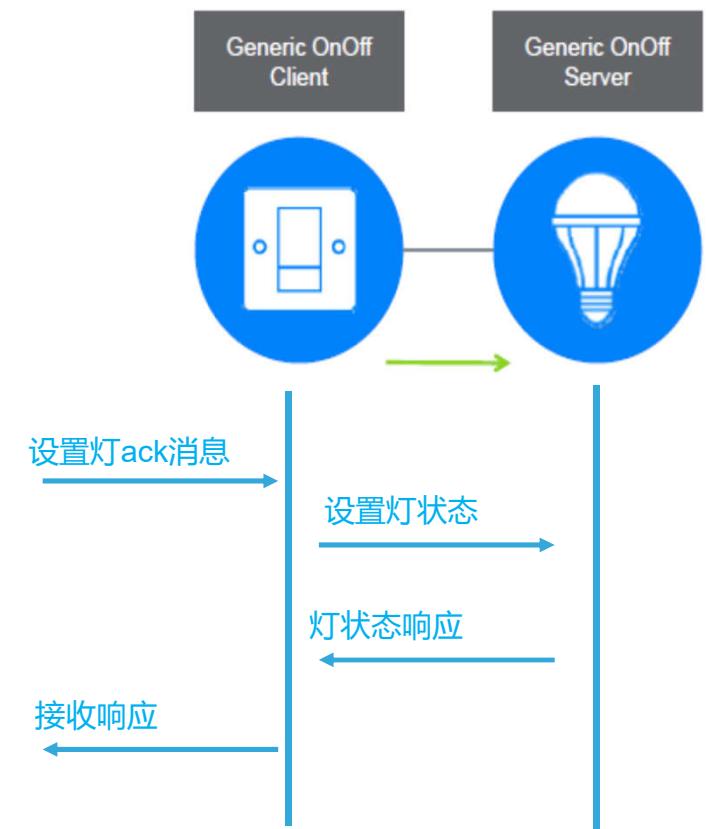


# SIG 或厂商模型

178

- 这些模型可能被SIG或者厂商定义和采用
- SIG采用模型
  - 蓝牙SIG定义的模型
  - 模型通过唯一的16位身份认定
- 厂商模型
  - 厂商定义的模型
  - SIG采用的模型，或者32比特，提供给厂商

## 通用模型示例



# Adding Model in Firmware

179

- All the supported models has been mentioned in the mesh\_cfg.h file
- Supported models can be enabled and disabled using macros
- Each macro defines corresponding model Id provided by SIG

Model Group	Model Name	SIG Model ID
Generic	Generic OnOff Server	0x1000
	Generic OnOff Client	0x1001
	Generic Level Server	0x1002
	Generic Level Client	0x1003

```
/* Define the following Macros to enable the usage of the Generic Models */
#define ENABLE_GENERIC_MODEL_SERVER_ONOFF
#define ENABLE_GENERIC_MODEL_SERVER_LEVEL
//#define ENABLE_GENERIC_MODEL_SERVER_DEFAULT_TRANSITION_TIME
//#define ENABLE_GENERIC_MODEL_SERVER_POWER_ONOFF
//#define ENABLE_GENERIC_MODEL_SERVER_POWER_ONOFF_SETUP
//#define ENABLE_GENERIC_MODEL_SERVER_POWER_LEVEL
```

# 在固件中增加模型

- 所有支持的模型在mesh\_cfg.h 文件中都被提及
- 支持的模型可以通过宏定义使能或失能
- 每个宏都定义了SIG提供的相关ID

Model Group	Model Name	SIG Model ID
Generic	Generic OnOff Server	0x1000
	Generic OnOff Client	0x1001
	Generic Level Server	0x1002
	Generic Level Client	0x1003

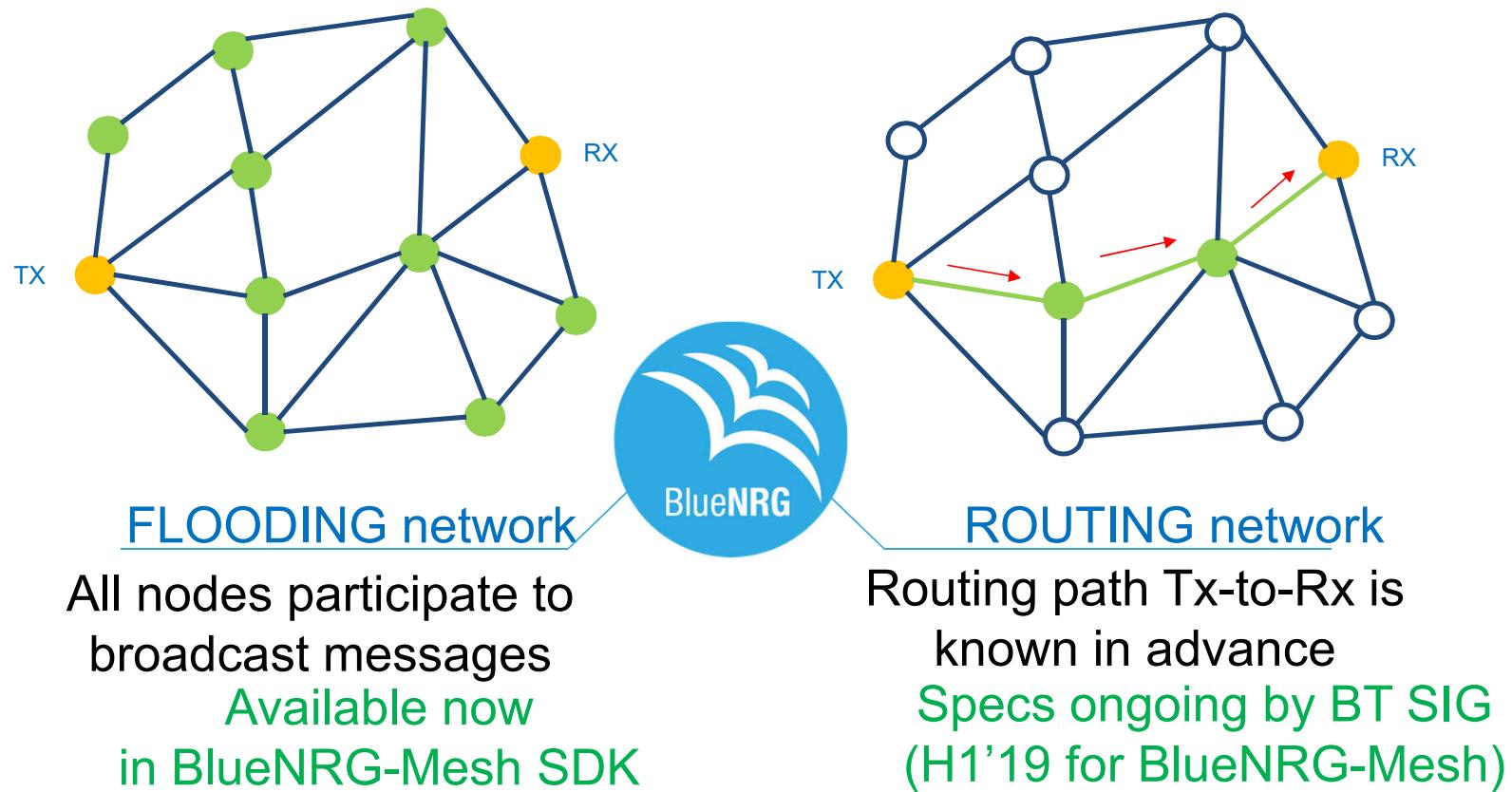
```

/* Define the following Macros to enable the usage of the Generic Models */
#define ENABLE_GENERIC_MODEL_SERVER_ONOFF
#define ENABLE_GENERIC_MODEL_SERVER_LEVEL
//#define ENABLE_GENERIC_MODEL_SERVER_DEFAULT_TRANSITION_TIME
//#define ENABLE_GENERIC_MODEL_SERVER_POWER_ONOFF
//#define ENABLE_GENERIC_MODEL_SERVER_POWER_ONOFF_SETUP
//#define ENABLE_GENERIC_MODEL_SERVER_POWER_LEVEL

```

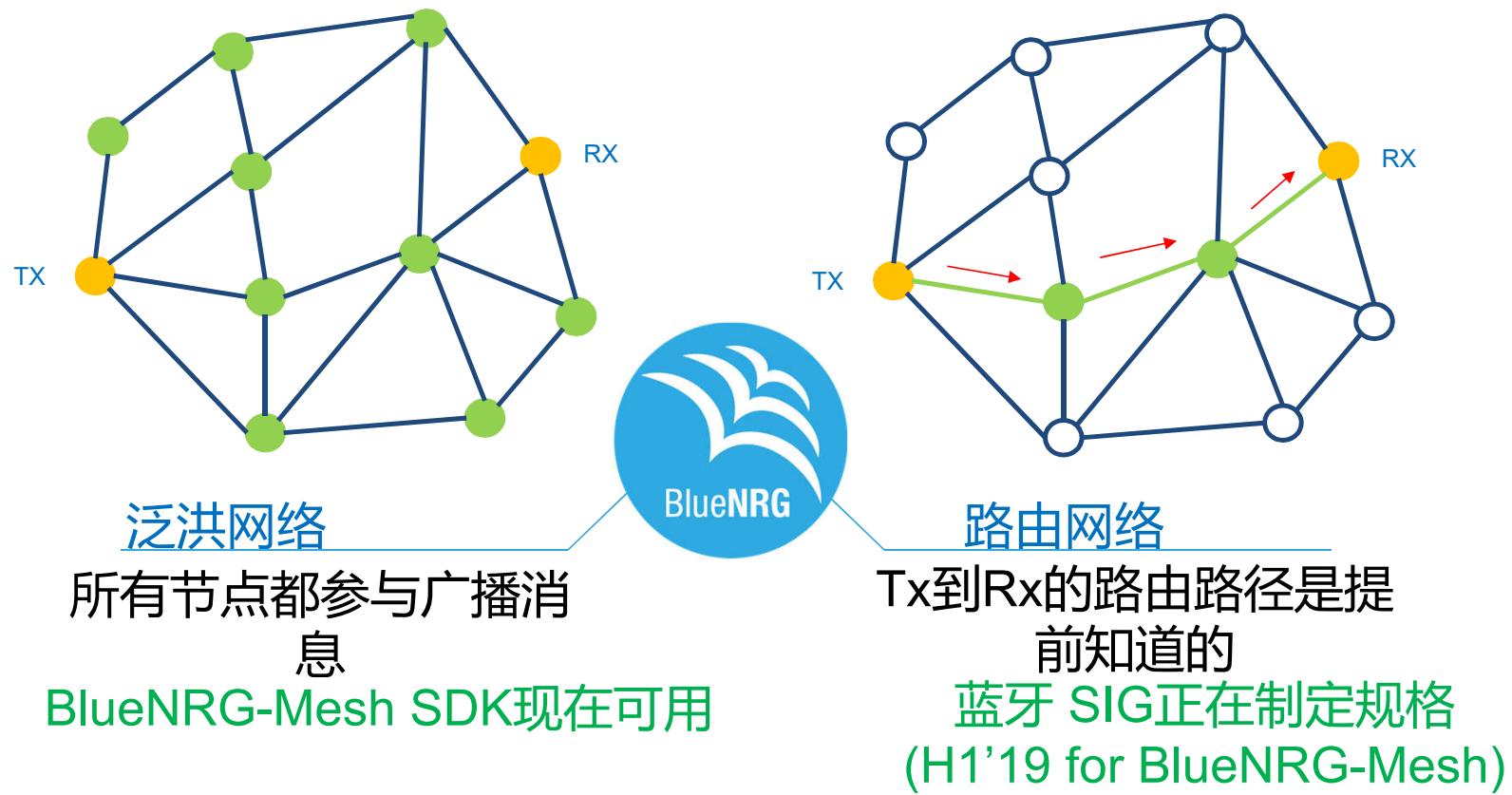
# Flooding vs Routing

181



# 泛洪 vs 路由

182



# Additional resources

183

- Bluetooth Mesh deep dive:
  - [Bluetooth Mesh specification](#)
  - [Bluetooth Mesh overview](#)
  - [Bluetooth Mesh FAQ](#)
- Software resources available this week (firmware, Android):  
<http://www.st.com/en/embedded-software/stsw-bnrg-mesh.html>
- BlueNRG-Mesh Android application available on Play Store:  
<https://play.google.com/store/apps/details?id=com.st.bluenrgmesh&hl=en>

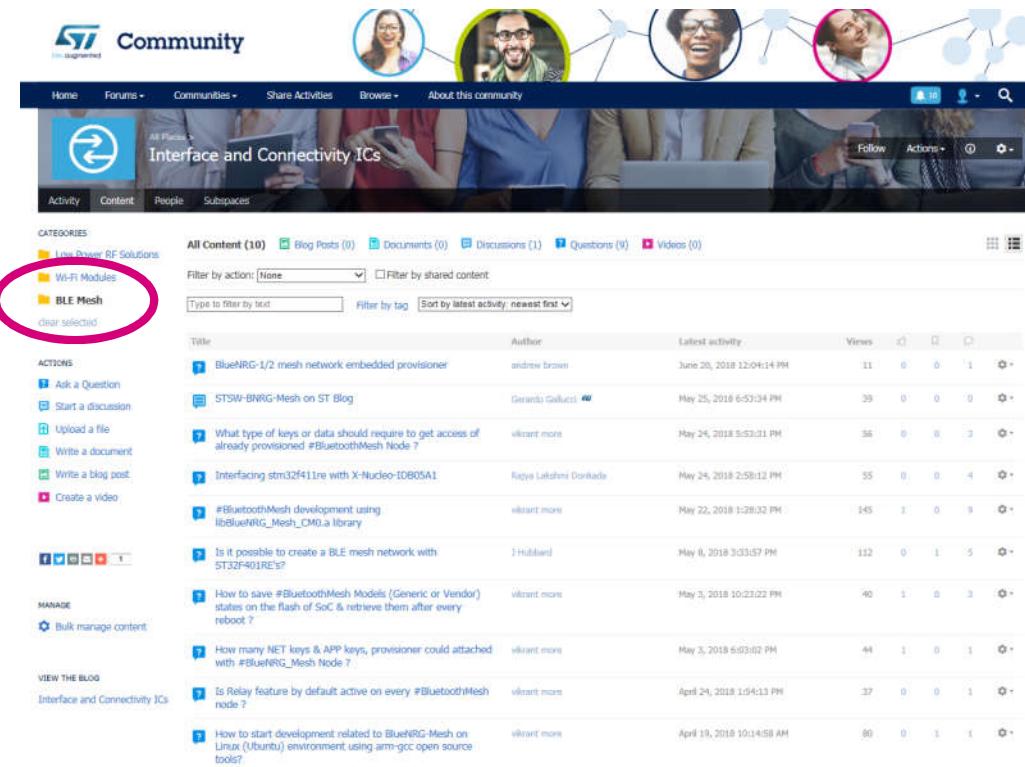
- 蓝牙 Mesh 深度学习:
  - [Bluetooth Mesh specification](#)
  - [Bluetooth Mesh overview](#)
  - [Bluetooth Mesh FAQ](#)
- 这周内软件资源可用(固件, 安卓): [http://www.st.com/en/embedded-software/stsw-bnrg-mesh.html](#)
- BlueNRG-Mesh 安卓应用程序在PlayStore中可用:  
[https://play.google.com/store/apps/details?id=com.st.bluenrgmesh&hl=en](#)

# 社区 @www.st.com

The screenshot shows the ST Community interface. At the top, there's a navigation bar with links for Home, Forums, Communities, Share Activities, Browse, and About this community. Below the navigation is a banner for 'Interface and Connectivity ICs'. The main content area displays a list of discussions under the 'BLE Mesh' category. A red circle highlights the 'BLE Mesh' category link in the sidebar. The list includes threads like 'BlueNRG-I/2 mesh network embedded provisioner', 'STSW-BNRG-Mesh on ST Blog', and 'How to save #BluetoothMesh Models (Generic or Vendor) states on the flash of SoC & retrieve them after every reboot ?'. Each thread shows the author, latest activity, views, and a settings icon.

- ST BLE Mesh community is Live and active
- <http://community.st.com/blemesh> is short link for discussions on e2e community
- 9 discussion threads + 28 messages

# Community @www.st.com



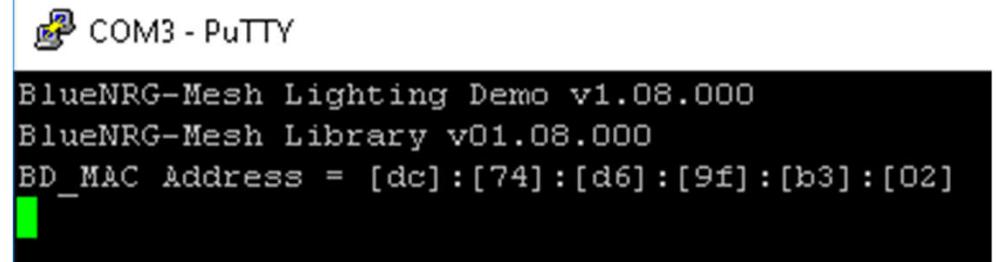
The screenshot shows the ST Community interface. At the top, there's a navigation bar with links for Home, Forums, Communities, Share Activities, Browse, and About this community. Below the navigation is a banner for 'Interface and Connectivity ICs'. The main content area displays a list of posts under the 'BLE Mesh' category, which is highlighted with a red oval. The list includes various topics such as 'BlueNRG-I/2 mesh network embedded provisioner', 'STSW-BNRG-Mesh on ST Blog', and 'What type of keys or data should require to get access of already provisioned #BluetoothMesh Node ?'. Each post has details like author, date, views, and interaction counts.

- ST BLE Mesh社区活力四射
- <http://community.st.com/blemesh> 是一个在e2e社区讨论的短链接
- 9个讨论主线+28条信息

# Exercise 1: MAC address

- Check the MAC address
- Correspondence in UUID
- Use of external MAC address
- Use of internal unique number
  - The unique serial number is a six bytes value stored at address 0x100007F4: it is stored as two words (8 bytes) at address 0x100007F4 and 0x100007F8 with unique serial number padded with 0xAA55.

```
#if !defined(DISABLE_TRACES)
/* Prints the MAC Address of the board */
printf("BlueNRG-Mesh Lighting Demo v%s\n\r", BLUENRG_MESH_APPLICATION_VERSION);
printf("BlueNRG-Mesh Library v%s\n\r", BluenrgMesh_GetLibraryVersion());
printf("BD_MAC Address = [%02x]:[%02x]:[%02x]:[%02x]:[%02x]:[%02x] \n\r",
       bdaddr[5],bdaddr[4],bdaddr[3],bdaddr[2],bdaddr[1],bdaddr[0] );
#endif
```



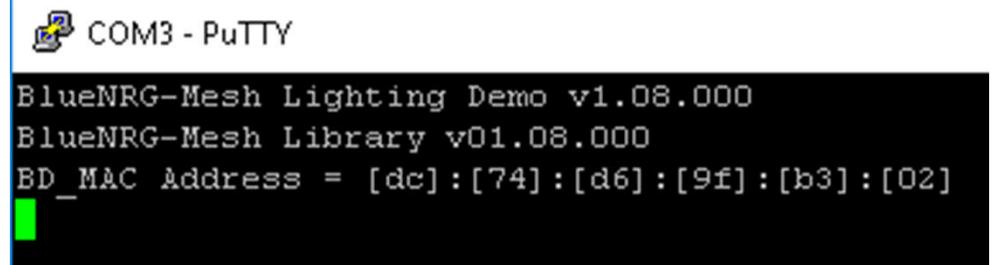
COM3 - PuTTY

```
BlueNRG-Mesh Lighting Demo v1.08.000
BlueNRG-Mesh Library v01.08.000
BD_MAC Address = [dc]:[74]:[d6]:[9f]:[b3]:[02]
```

# 练习 1: MAC 地址

- 检查MAC地址
- 使用UUID 通信
- 外部MAC地址的使用
- 内部唯一码的使用
  - 唯一序列号是一个存储在0x100007F4 地址的6字节数值：它被以2个字（8字节）存储在0x100007F4 地址和 0x100007F8 地址，以0xAA55作为补充。

```
#if !defined(DISABLE_TRACES)
    /* Prints the MAC Address of the board */
    printf("BlueNRG-Mesh Lighting Demo v%s\n\r", BLUENRG_MESH_APPLICATION_VERSION);
    printf("BlueNRG-Mesh Library v%s\n\r", BluenrgMesh_GetLibraryVersion());
    printf("BD_MAC Address = [%02x]:[%02x]:[%02x]:[%02x]:[%02x] \n\r",
           bdaddr[5],bdaddr[4],bdaddr[3],bdaddr[2],bdaddr[1],bdaddr[0] );
#endif
```



```
COM3 - PuTTY
BlueNRG-Mesh Lighting Demo v1.08.000
BlueNRG-Mesh Library v01.08.000
BD_MAC Address = [dc]:[74]:[d6]:[9f]:[b3]:[02]
```

# MAC address Management

- BlueNRG2\_Lighting\_Demo: This binary uses static random address of the controller to generate MAC address, Other options to generate MAC address are as below
- BlueNRG2\_Lighting\_Demo\_Ext\_MAC : Use this binary to use configure the MAC addresses externally.
  - Note: When using external MAC address, it is recommended to firstly program the BlueNRG2\_Lighting\_Demo\_Ext\_MAC.bin file and program the MAC addresses from Utilities\BlueNRG2\_MAC later
- BlueNRG2\_Lighting\_Demo\_UuidSno: This binary uses internal unique serial number of the controller to generate MAC address.

# MAC address 管理

190

- BlueNRG2\_Lighting\_Demo: 这个bin文件使用控制器的静态随机地址生成MAC地址，其他生成MAC地址的选项如下
- BlueNRG2\_Lighting\_Demo\_Ext\_MAC : 使用bin去配置外部MAC地址
  - 备注：当使用外部MAC地址时，推荐先烧录BlueNRG2\_Lighting\_Demo\_Ext\_MAC.bin 然后使用Utilities\BlueNRG2\_MAC工具 烧录MAC地址
- BlueNRG2\_Lighting\_Demo\_UniqueSno: 这个bin文件使用控制器内部唯一序列号生成MAC地址

# Exercise 2: UUID changes

- How to modify the UUID
- UUID is 128 bits (16 bytes) and can guarantee uniqueness across space and time.
- Can be “Time-based” UUID or “truly-random or pseudo-random numbers”

```

MOBLE_RESULT Appli_BleSetUUIDCb(MOBLEUINT8 *uuid_prefix_data)
{
    /* UUID is 128 bits (16 bytes) and can guarantee uniqueness across space and time.
     It can be "Time-based" UUID or "truly-random or pseudo-random numbers"

    In this buffer user need to fill 10 bytes of UUID data.
    Remaining 6 bytes, library fill this data with BDaddress.

    For example :
        F81D4FAE-7DEC-XBC4-Y12F-17D1AD07A961 (16 bytes)
        <any number> |_ |_ |_ <BDAddress>

        X = 1 i.e. "Time Base UUID"
        X = 4 i.e. "Random Number"
        Y = 4 i.e. Conforming to the current spec
        For UUID information refer RFC4122
    */

    /* copy UUID prefix data*/
    uint8_t uuid_buff[10] = {0xF8,0x1D,0x4F,0xAE,0x7D,0xEC};/*random value by user */

    uuid_buff[6] = 0x4B; /* 0x4B => 4 : Random */
    uuid_buff[7] = 'S' ; /* User define value */
    uuid_buff[8] = 0xA1; /* 0xA1 => A : Conforming to the current Spec */
    uuid_buff[9] = 'T' ; /* User define value */

    memcpy(uuid_prefix_data,uuid_buff,sizeof(uuid_buff));

    return MOBLE_RESULT_SUCCESS;
}

```

## 练习 2: 改变UUID

- 如何修改UUID
- UUID是128比特（16字节）并且能够在时间和空间上保证唯一性。
- 可以是“时间片”UUID或者“真随机数或伪随机数”

```

MOBLE_RESULT Appli_BleSetUUIDCb(MOBLEUINT8 *uuid_prefix_data)
{
    /* UUID is 128 bits (16 bytes) and can guarantee uniqueness across space and time.
     It can be "Time-based " UUID or "truly-random or pseudo-random numbers"

     In this buffer user need to fill 10 bytes of UUID data.
     Remaining 6 bytes, library fill this data with BDaddress.

     For example :
     F81D4FAE-7DEC-XBC4-Y12F-17D1AD07A961 (16 bytes)
     <any number> |_ |_ |_ <BDAddress>

     X = 1 i.e. "Time Base UUID"
     X = 4 i.e. "Random Number"
     Y = 4 i.e. Conforming to the current spec
     For UUID information refer RFC4122
 */

    /* copy UUID prefix data*/
    uint8_t uuid_buff[10] = {0xF8,0x1D,0x4F,0xAE,0x7D,0xEC};/*random value by user */

    uuid_buff[6] = 0x4B; /* 0x4B => 4 : Random */
    uuid_buff[7] = 'S' ; /* User define value */
    uuid_buff[8] = 0xA1; /* 0xA1 => A : Conforming to the current Spec */
    uuid_buff[9] = 'T' ; /* User define value */

    memcpy(uuid_prefix_data,uuid_buff,sizeof(uuid_buff));

    return MOBLE_RESULT_SUCCESS;
}

```

# Exercise 3: TxPower changes

193

- How to modify the Transmit Power
- By default, +4 dbm is configured in the nodes for BlueNRG-1 and BlueNRG-2

```
MOBLE_RESULT Appli_BleSetTxPowerCb(void)
{
    tBleStatus status;
    /* API to change the Transmission power of BlueNRG Device */
    status = aci_hal_set_tx_power_level(POWER_LEVEL_HIGH,TX_POWER_LEVEL_PLUS_4DBM);
    /* Check if command executed successfully */
    if (status)
        return MOBLE_RESULT_FAIL;
    else
        return MOBLE_RESULT_SUCCESS;
}
```

```
/* MACROS for Power Level definitions */
#define POWER_LEVEL_LOW          0
#define TX_POWER_LEVEL_MINUS_18DBM 0 // = -18 dBm,
#define TX_POWER_LEVEL_MINUS_15DBM 1 // = -15 dBm,
#define TX_POWER_LEVEL_MINUS_12DBM 2 // = -12 dBm,
#define TX_POWER_LEVEL_MINUS_9DBM  3 // = -9 dBm,
#define TX_POWER_LEVEL_MINUS_6DBM  4 // = -6 dBm,
#define TX_POWER_LEVEL_MINUS_2DBM  5 // = -2 dBm,
#define TX_POWER_LEVEL_0DBM       6 // = 0 dBm,
#define TX_POWER_LEVEL_PLUS_5DBM  7 // = 5 dBm.
#define POWER_LEVEL_HIGH          1
#define TX_POWER_LEVEL_MINUS_14DBM 0 // = -14 dBm,
#define TX_POWER_LEVEL_MINUS_11DBM 1 // = -11 dBm,
#define TX_POWER_LEVEL_MINUS_8DBM  2 // = -8 dBm,
#define TX_POWER_LEVEL_MINUS_5DBM  3 // = -5 dBm,
//#define TX_POWER_LEVEL_MINUS_2DBM 4 // = -2 dBm,
#define TX_POWER_LEVEL_PLUS_2DBM  5 // = 2 dBm,
#define TX_POWER_LEVEL_PLUS_4DBM  6 // = 4 dBm,
#define TX_POWER_LEVEL_PLUS_8DBM  7 // = 8 dBm
```

# 练习 3: 改变发射功率

- 如何修改发射功率
- BlueNRG-1 和 BlueNRG-2 的节点默认配置为 +4dbm

```
MOBLE_RESULT Appli_BleSetTxPowerCb(void)
{
    tBleStatus status;
    /* API to change the Transmission power of BlueNRG Device */
    status = aci_hal_set_tx_power_level(POWER_LEVEL_HIGH,TX_POWER_LEVEL_PLUS_4DBM);
    /* Check if command executed successfully */
    if (status)
        return MOBLE_RESULT_FAIL;
    else
        return MOBLE_RESULT_SUCCESS;
}
```

```
/* MACROS for Power Level definitions */
#define POWER_LEVEL_LOW 0
#define TX_POWER_LEVEL_MINUS_18DBM 0 // = -18 dBm,
#define TX_POWER_LEVEL_MINUS_15DBM 1 // = -15 dBm,
#define TX_POWER_LEVEL_MINUS_12DBM 2 // = -12 dBm,
#define TX_POWER_LEVEL_MINUS_9DBM 3 // = -9 dBm,
#define TX_POWER_LEVEL_MINUS_6DBM 4 // = -6 dBm,
#define TX_POWER_LEVEL_MINUS_2DBM 5 // = -2 dBm,
#define TX_POWER_LEVEL_0DBM 6 // = 0 dBm,
#define TX_POWER_LEVEL_PLUS_5DBM 7 // = 5 dBm.
#define POWER_LEVEL_HIGH 1
#define TX_POWER_LEVEL_MINUS_14DBM 0 // = -14 dBm,
#define TX_POWER_LEVEL_MINUS_11DBM 1 // = -11 dBm,
#define TX_POWER_LEVEL_MINUS_8DBM 2 // = -8 dBm,
#define TX_POWER_LEVEL_MINUS_5DBM 3 // = -5 dBm,
// #define TX_POWER_LEVEL_MINUS_2DBM 4 // = -2 dBm,
#define TX_POWER_LEVEL_PLUS_2DBM 5 // = 2 dBm,
#define TX_POWER_LEVEL_PLUS_4DBM 6 // = 4 dBm,
#define TX_POWER_LEVEL_PLUS_8DBM 7 // = 8 dBm
```

# Exercise 4: GATT Conn Indication

195

- How to modify the GATT connection LED
- Proxy node in the network has the capability to make connection with the smartphone through GATT interface.
- Callback function can be used to indicate that node is connected/Disconnected with the smartphone

```
void Appli_BleGattConnectionCompleteCb(void)
{
    /* Proxy Node, will be called whenever Gatt connection is established */
    /* Turn ON Red LED*/
    SdkEvalLedOn(LED2);
}

void Appli_BleGattDisconnectionCompleteCb(void)
{
    /* Proxy Node, will be called whenever Gatt disconnected */
    /* Turn Off Red LED*/
    SdkEvalLedOff(LED2);
}
```

# 练习 4: GATT 连接指示

- 如何修改GATT连接LED
- 网络中的代理节点有能力通过GATT接口与智能手机机进行连接。
- 回调功能能够被用于指示节点是否跟智能手机连接。

```
void Appli_BleGattConnectionCompleteCb(void)
{
    /* Proxy Node, will be called whenever Gatt connection is established */
    /* Turn ON Red LED*/
    SdkEvalLedOn(LED2);
}
```

```
void Appli_BleGattDisconnectionCompleteCb(void)
{
    /* Proxy Node, will be called whenever Gatt disconnected */
    /* Turn Off Red LED*/
    SdkEvalLedOff(LED2);
}
```

# Exercise 5: Generic Models

- How to modify generic model messages
- Switch On/Off LED using Generic On/Off set message
- Intensity control using generic level set message

```
MOBLE_RESULT Appli_Generic_OnOff_Set(Generic_OnOffStatus_t* pGeneric_OnOffParam,
                                      MOBLEUINT8 OptionalValid)
{
    AppliOnOffSet.Present_OnOff = pGeneric_OnOffParam->Present_OnOff;
    SetLed(AppliOnOffSet.Present_OnOff);

    return MOBLE_RESULT_SUCCESS;
}
```

```
MOBLE_RESULT Appli_Generic_Level_Set(Generic_LevelStatus_t* plevelParam,
                                      MOBLEUINT8 OptionalValid)
{
    uint16_t duty;
    AppliLevelSet.Present_Level16= plevelParam->Present_Level16;

    /* For demo, if Level is more than 100, switch ON the LED */

    /* increment of 20 percent at each level */
    duty = PwmValueMapping(AppliLevelSet.Present_Level16 , 0x7FFF ,0);

    Modify_PWM(2, duty); /* 2 is for GPIO_6. Software PWM */

    return MOBLE_RESULT_SUCCESS;
}
```

# 练习 5: 通用 模型

- 如何修改通用模型消息
- 使用通用开关消息开关LED
- 使用通用水平消息控制强度

```
MOBLE_RESULT Appli_Generic_OnOff_Set(Generic_OnOffStatus_t* pGeneric_OnOffParam,  
                                      MOBLEUINT8 OptionalValid)  
{  
    AppliOnOffSet.Present_OnOff = pGeneric_OnOffParam->Present_OnOff;  
    SetLed(AppliOnOffSet.Present_OnOff);  
  
    return MOBLE_RESULT_SUCCESS;  
}
```

```
MOBLE_RESULT Appli_Generic_Level_Set(Generic_LevelStatus_t* plevelParam,  
                                      MOBLEUINT8 OptionalValid)  
{  
    uint16_t duty;  
    AppliLevelSet.Present_Level16= plevelParam->Present_Level16;  
  
    /* For demo, if Level is more than 100, switch ON the LED */  
  
    /* increment of 20 percent at each level */  
    duty = PwmValueMapping(AppliLevelSet.Present_Level16 , 0x7FFF ,0);  
  
    Modify_PWM(2, duty); /* 2 is for GPIO_6. Software PWM */  
  
    return MOBLE_RESULT_SUCCESS;  
}
```

# Exercise 6: Vendor Models

- Putting logs and checking commands and response
- Checking the addressing:  
Unicast/Group/broadcast

```
MOBLE_RESULT Appli_Vendor_LEDControl( MOBLEUINT8 const *data, MOBLEUINT32 length,
                                         MOBLEUINT8 elementNumber)

{
    MOBLE_RESULT status = MOBLE_RESULT_SUCCESS;
    MOBLEUINT8 subCommand;
    subCommand = data[0];
    MOBLEUINT16 duty;
    MOBLEUINT16 intensityValue;

    switch(subCommand)
    {
        /*
         Message Received      B0      B1      B2      B3      B4      B5      B6      B7
                           B0 - Sub-Cmd LED
                           B1-B7 - Data Bytes
        */
        case APPLI_CMD_LED_BULB:
        {
            if(elementNumber == FIRST_ELEMENT)
            {
                Appli_LedState = *(data+1); /* Toggle the state of the Blue LED */
            }
        }
    }
}
```

# 练习 6: 厂商模型

- 打印日志以及检查指令和响应
- 检查地址：单播、组播和全体广播

```
MOBLE_RESULT Appli_Vendor_LEDControl( MOBLEUINT8 const *data, MOBLEUINT32 length,
                                         MOBLEUINT8 elementNumber)

{
    MOBLE_RESULT status = MOBLE_RESULT_SUCCESS;
    MOBLEUINT8 subCommand;
    subCommand = data[0];
    MOBLEUINT16 duty;
    MOBLEUINT16 intensityValue;

    switch(subCommand)
    {
        /*
         Message Received      B0      B1      B2      B3      B4      B5      B6      B7
                           B0 - Sub-Cmd LED
                           B1-B7 - Data Bytes
        */
        case APPLI_CMD_LED_BULB:
        {
            if(elementNumber == FIRST_ELEMENT)
            {
                Appli_LedState = *(data+1); /* Toggle the state of the Blue LED */
            }
        }
    }
}
```

# Exercise7: Publish/Subscribe

201

- Playing with Publish messages

```
void Vendor_Publish(void)
{
    MOBLE_ADDRESS publishAddress;
    MOBLEUINT8 elementNumber;

    /*Select the Element Number for which publication address is required*/

    if (NumberOfElements == 1)
    {
        elementNumber = 0x01;
    }

    else if(NumberOfElements == 2)
    {
        elementNumber = 0x02; /*Element 2 is configured as switch*/
    }

    else if(NumberOfElements == 3)
    {
        elementNumber = 0x03; /*Element 3 is configured as switch*/
    }

    publishAddress = BluenrgMesh_GetPublishAddress(elementNumber);

    if(publishAddress)
    {
#define !defined(DISABLE_TRACES)
        printf("Published Address is= 0x%2x \n\r", publishAddress);
#endif

        /* changes the LED status on other nodes in the network */
        if(CommandStatus == 1)
        {
            AppliBuffer[0] = 2;
        }
        else
        {
            AppliBuffer[0] = 1;
        }
        BluenrgMesh_SetRemoteData(publishAddress,
                                APPLI_LED_CONTROL_STATUS_CMD ,
                                AppliBuffer, sizeof(AppliBuffer), NULL);
    }
}
```

# 练习: 发布、订阅

- 练习使用发布消息

```

void Vendor_Publish(void)
{
    MOBLE_ADDRESS publishAddress;
    MOBLEUINT8 elementNumber;

    /*Select the Element Number for which publication address is required*/

    if (NumberOfElements == 1)
    {
        elementNumber = 0x01;
    }

    else if(NumberOfElements == 2)
    {
        elementNumber = 0x02; /*Element 2 is configured as switch*/
    }

    else if(NumberOfElements == 3)
    {
        elementNumber = 0x03; /*Element 3 is configured as switch*/
    }

    publishAddress = BluenrgMesh_GetPublishAddress(elementNumber);

    if(publishAddress)
    {
#define !defined(DISABLE_TRACES)
        printf("Published Address is= 0x%2x \n\r", publishAddress);
#endif

        /* changes the LED status on other nodes in the network */
        if(CommandStatus == 1)
        {
            AppliBuffer[0] = 2;
        }
        else
        {
            AppliBuffer[0] = 1;
        }
        BluenrgMesh_SetRemoteData(publishAddress,
                                APPLI_LED_CONTROL_STATUS_CMD ,
                                AppliBuffer, sizeof(AppliBuffer), NULL);
    }
}

```

# Exercise8: LED Intensity Control

- Five PWMs available in BlueNRG1 & BlueNRG2 boards to control LEDs intensity
- Three are software PWMs & rest two are hardware PWMs
- Software PWM GPIOs can be modified using macros declared in `PWM_config.h` provided in firmware at below path
  - `Embedded\Drivers\BSP\SDK_Eval_BlueNRG1\inc`

```

/* PWM pins for MFT1 and MFT2 respectively */
#define PWM0_PIN           GPIO_Pin_4
#define PWM1_PIN           GPIO_Pin_3

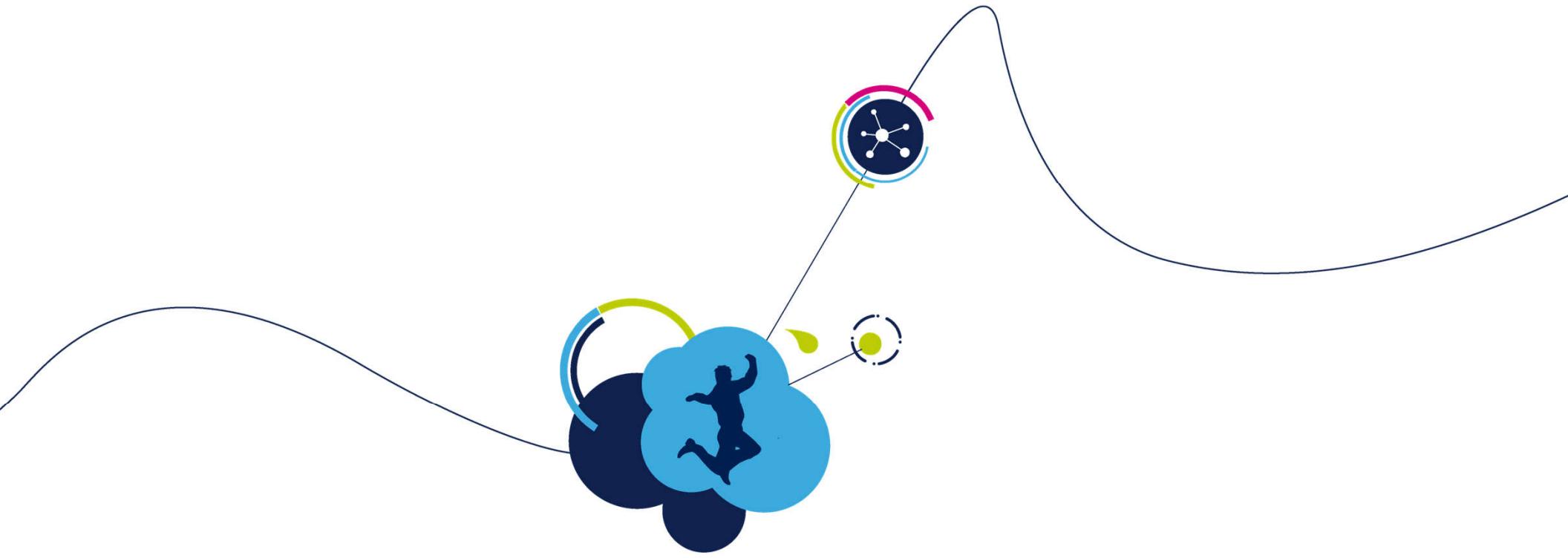
/* GPIOs pins (SOFTWARE PWM) used to output a PWM signal */
#define PWM2_PIN           GPIO_Pin_6 //YELLOW LED
#define PWM3_PIN           GPIO_Pin_2
#define PWM4_PIN           GPIO_Pin_1

```

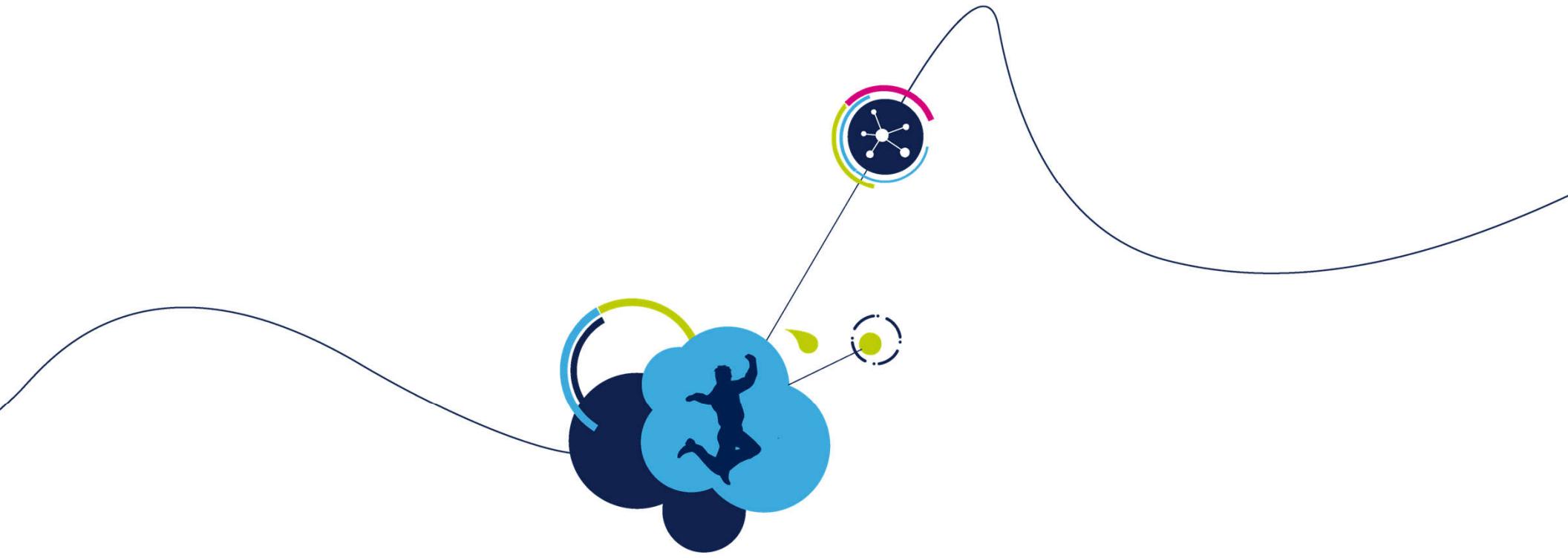
# 练习 8: LED 强度控制

- BlueNRG1和BlueNRG2 开发板有5路PWM可用来控制LED强度
- 有软件PWM和其余的2路硬件PWM
- 软件PWM的GPIO可以使用宏进行修改，宏在如下路径提供的固件PWM\_config.h 中声明
  - *Embedded\Drivers\BSP\SDK\_Eval\_BlueNRG1\inc*

```
/* PWM pins for MFT1 and MFT2 respectively */  
#define PWM0_PIN GPIO_Pin_4  
#define PWM1_PIN GPIO_Pin_3  
  
/* GPIOs pins (SOFTWARE PWM) used to output a PWM signal */  
#define PWM2_PIN GPIO_Pin_6 //YELLOW LED  
#define PWM3_PIN GPIO_Pin_2  
#define PWM4_PIN GPIO_Pin_1
```



Thanks



谢谢!