ZS110A BLE 用户使用指南

发布 ***1.0.0***

2018 年 **11** 月 **02** 日

目录

CHAPTER 1

文档介绍

1.1 文档目的

为低功耗蓝牙协议栈开发提供快速指引. 该文档只提供简单开发指引.

1.2 术语说明

表 1.1: 术语说明

术语	说明
BLE	蓝牙低功耗技术
GAP	通用接入规范
GATT	通用属性协议
ATT	属性协议
SMP	安全管理协协议
L2CAP	链路控制和适配协议
HCI	主机接口规范
LL	链路层
RF	物理层, 射频发射

1.3 参考文档

- http://docs.zephyrproject.org/1.9.0/subsystems/bluetooth/bluetooth.html

1.4 版本历史

表 1.2: 版本历史

日期	版本	注释	作者
2018-08-22	1.0	初始版本	ZS110A 项目组

CHAPTER 2

BLE 协议栈简介

2.1 BLE 协议栈层次架构

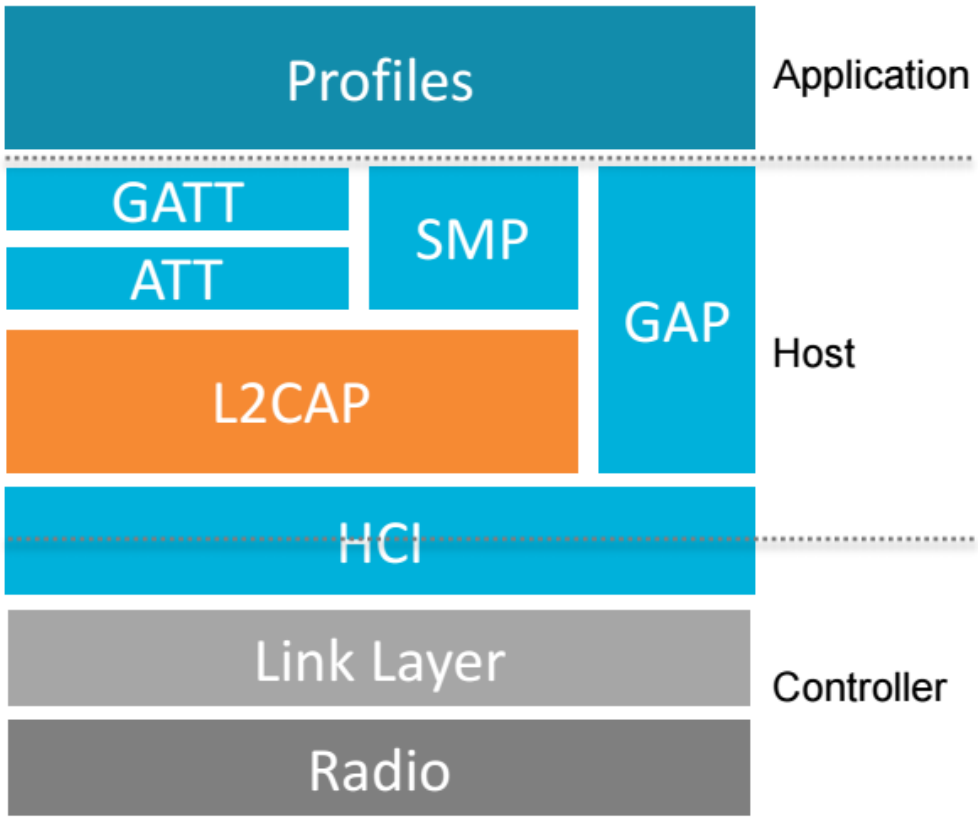

图 2.1: BLE 协议栈层次框图

- 基于 GATT 的应用规范 (Application)
  - 即时报警服务、FindMe 规范、链路丢失服务、健康体温计规范、电池服务等
- 通用接入规范 (Generic Access Profile, GAP)
  - 蓝牙设备如何发现、建立连接及绑定远端蓝牙设备的通用程序
- 通用属性协议 (Generic Attribute Protocol, GATT)
  - 定义了服务的流程、格式及其所包含的特征，包含特征的发现、读取、写入、通知、指示
  - Client/Server Model
- 属性协议 (Attribute Protocol, ATT)
  - 用于发现、读取和写入对端设备上的属性的规范
  - Attribute database
- 安全管理协协议 (Security Manager, SMP)
  - 配对和密钥分发
- 链路控制和适配协议 (Logical Link Control and Adaption Protocol, L2CAP)
  - 数据分组交换
- 主机接口规范 (Host controller Interface, HCI)
  - 主机和控制器之间的接口
- 链路层 (Link layer, MAC)
  - 链路管理，执行基带协议和其它低级的链路程序
- 物理层 (physical layer, RF&BandBand)
  - 空中包的收发

2.2 physical layer

- 低功耗蓝牙设备工作在 2.4 GHz ISM 频段
- 1 Mbps PHY 吞吐量
- 40 信道, 3 个广播信道，37 个数据信道
- 自适应调频

2.3 Link layer

- 角色
  - Master

  - Slave
- 主要责任
  - 信道管理
  - 广播和扫描
  - 创建和保持连接
  - 空中包收发
  - 加密链路
- 状态机
  - Standby：低功耗模式
  - Advertising: 在 3 个广播信道发送广播包
  - Scaning：主动和被动扫描
  - Initiating：创建连接
  - Connection：数据通讯

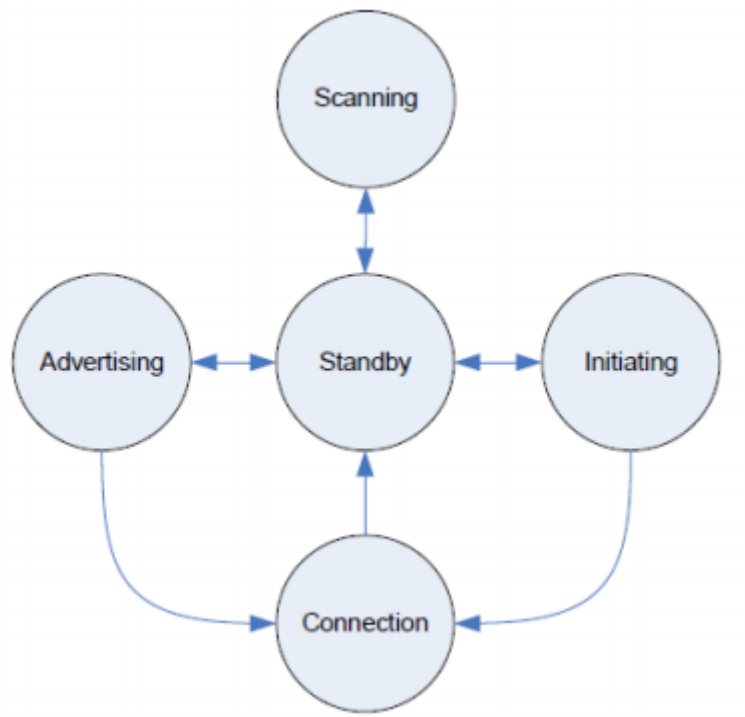

图 2.2: 链路层状态机

- 空中接口包的格式
  - 前导数据
  - 地址
  - PDU 头

- PDU 长度
- PDU 数据
- CRC

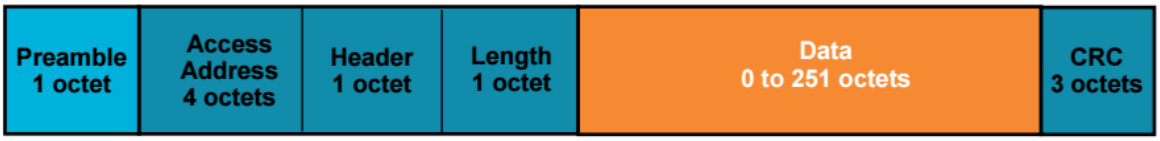

图 2.3: 空中包格式

- 广播和扫描

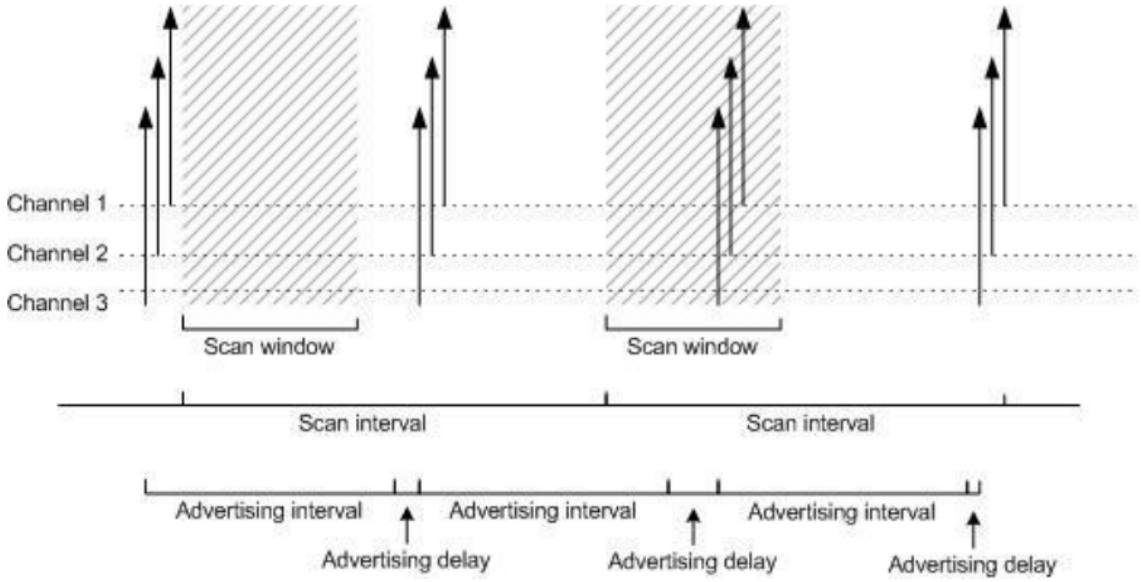

图 2.4: 广播和扫描过程

- 数据通讯

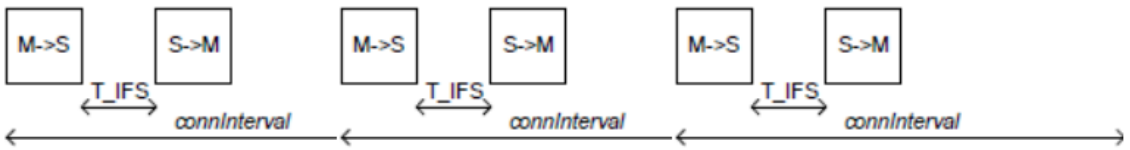

图 2.5: 数据通讯过程

2.4 HCI

- 主机和控制器之间的接口

  - 一种标准的通讯的机制
  - 允许主机将命令和数据发送到控制器
  - 允许控制器将事件和数据发送到主机

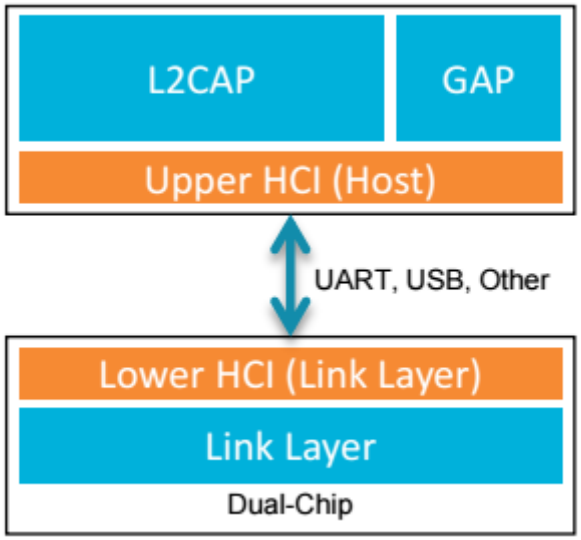

图 2.6: HCI 双芯片架构

2.5 GAP

- 角色
  - 广播角色：发送广播包
  - 观察者角色: 接收广播包
  - 中心角色：建立连接的发起者 (LL Master)
  - 外围角色：接受连接的建立 (LL Slave)

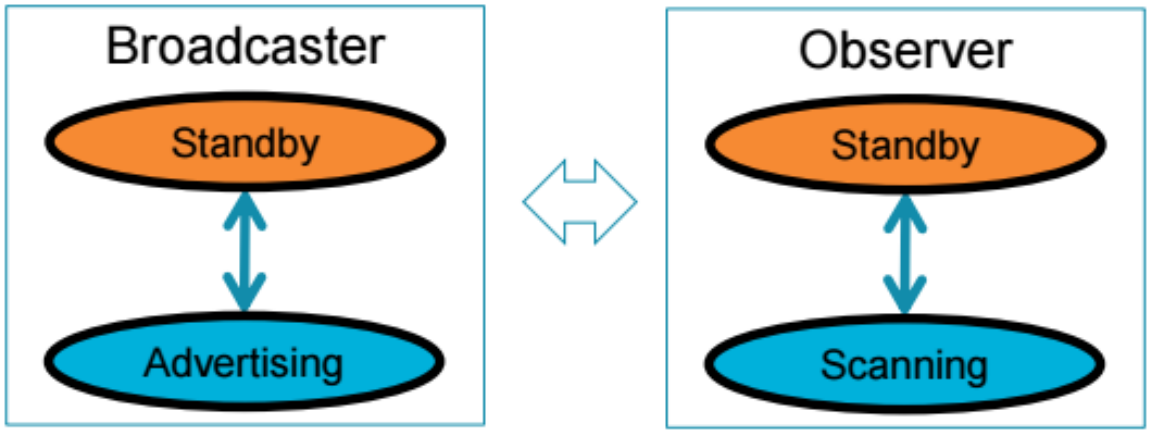

图 2.7: 广播者角色和观察者角色

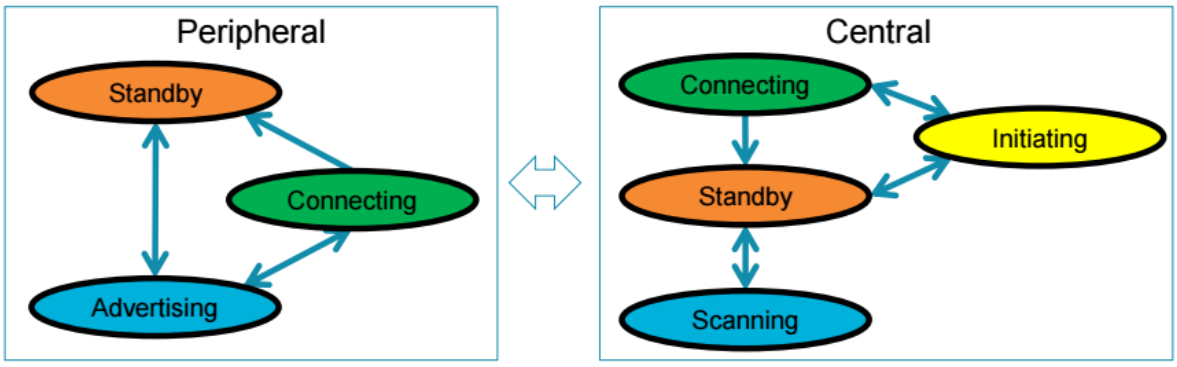

图 2.8: 中心角色和外围角色

- 连接管理
  - 配对：交换安全特征
  - 绑定：long term key 保存
  - 发现：服务和特征的发现
- 设备管理
  - 广播
  - 扫描
- 安全
  - 加密
  - 授权

2.6 L2CAP

- 数据分割和重合
  - 支持最大传输单元来提供传输效率，最大传送单元长度可以大于基带数据支持的长度
- 信道复用
  - 属性协议，即 CID=0x0004
  - 安全协议，即 CID=0x0006
  - L2CAP signal 信道，即 CID=0x0005

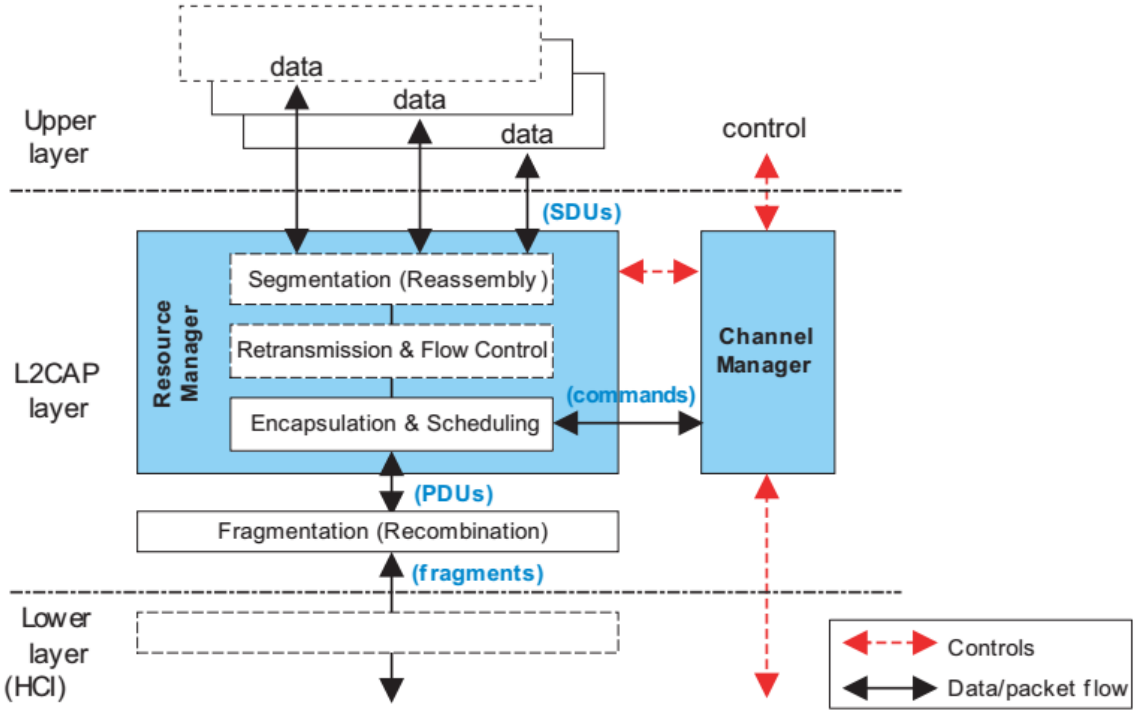

图 2.9: L2CAP 层框图

2.7 SMP

配对和密钥分发

- 加密和授权
- 配对过程
  1. 共享双方设备的 I/O 能力
  2. 生成 Short Term Key(STK)
  3. 特定的密钥分发
- 配对方法
  - Just works
  - Passkey entry
  - Out of band
  - Numeric comparion(for LE Secure connections)
- 绑定
  - 把配对过程中生成的 long term key 存储起来，这个过程叫绑定

	No Output	Display
No Input	No Input No Output	Display Only
Yes / No	No Input No Output	Display Yes / No
Keyboard	Keyboard Only	Keyboard Display

图 2.10: 配对中的 IO 组合情况

2.8 ATT

- 角色
  - 客户端
  - 服务端
- 属性
  - 属性类型：UUID(16bit or 128 bit)
  - 属性句柄：服务器上的所有属性都会分配一个唯一非零的属性句柄
  - 属性许可：使用许可、认证许可、授权许可
  - 属性值：0-512 byte

Handle	Type	Value
0x0001	Primary Service	GAP
0x0002	Characteristic	Device Name
0x0003	Device Name	“Tag”
0x0007	Primary Service	Tx Pwr
0x0008	Characteristic	Tx Pwr
0x0009	Tx Power	-4
0x0010	Primary Service	Battery
0x0011	Characteristic	Batt Level
0x0012	Battery Level	75%
0x0013	Char. Format	Uint8, 0, percent

图 2.11: att 属性储存数据库结构

- 方法类型
  - 请求 (客户端到服务端)
  - 应答 (服务端到客户端，是对请求的回应)
  - 命令 (客户端到服务端，不需要应答)
  - 通知 (服务端到客户端，不要确认)
  - 指示 (服务端到客户端)
  - 确认 (客户端到服务端，是对指示的回应)

2.9 GATT

- 角色
  - GATT 客户端：Read，write，notify，or indicate operations
  - GATT 服务端：储存数据和提供访问数据的方法
  - 独立于 LL Master/Slave

  - GATT 客户端和服务端能够同时存在

- 为应用提供属性支持
  - 发现所有服务、特征及特征描述符
  - 客户端发起过程 (writing values)
  - 服务端发起过程 (notifications, indications)

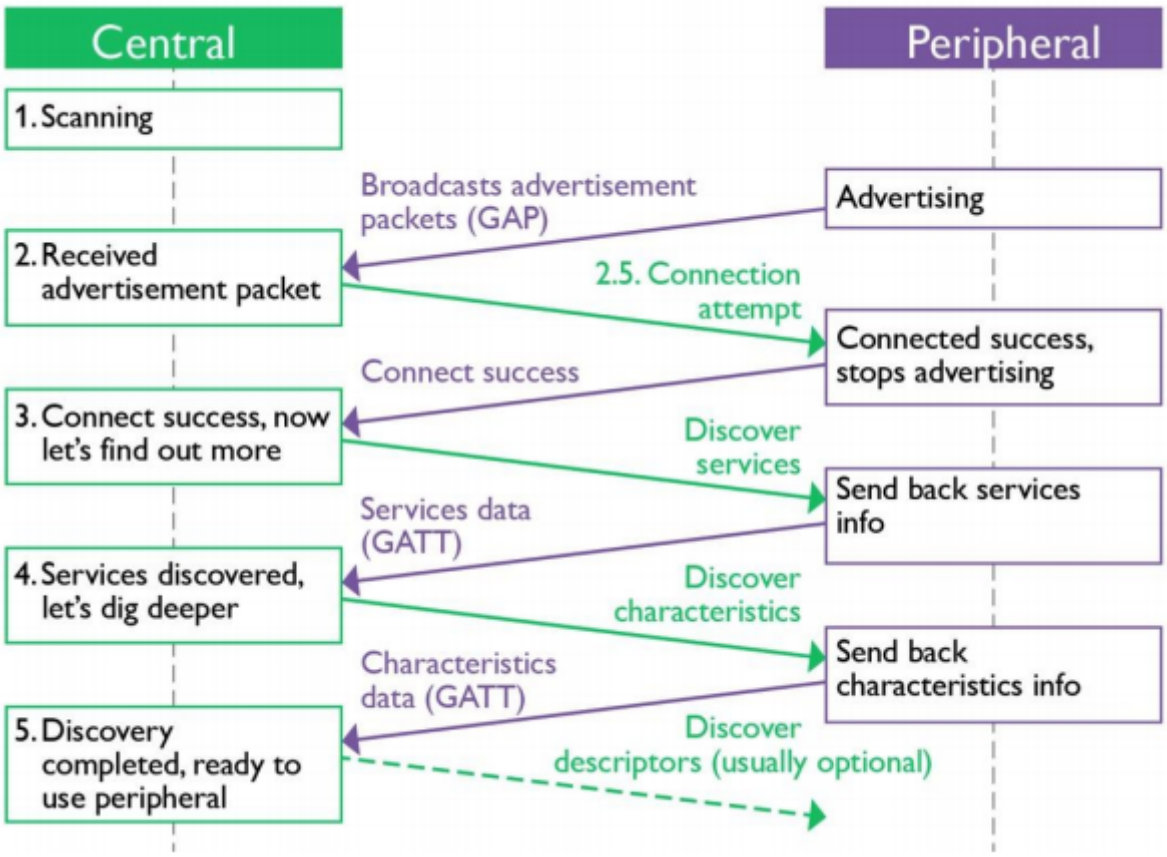

图 2.12: BLE 连接示图

2.10 Profiles

- 基础概念
  - 描述 2 个或者更多设备的行为规范
  - 描述设备的可发现性和可连接性
  - Profile 客户端实现如何使用服务 (管理)
  - Profile 服务端实现具体服务 (数据库)
- 例子: 心跳 Profile
  - 产品 (心跳监视)

– 角色：心跳数据采集者和收集者
– 用途：心跳采集设备测量心跳和其它信息，收集者接收采集者测量的数据

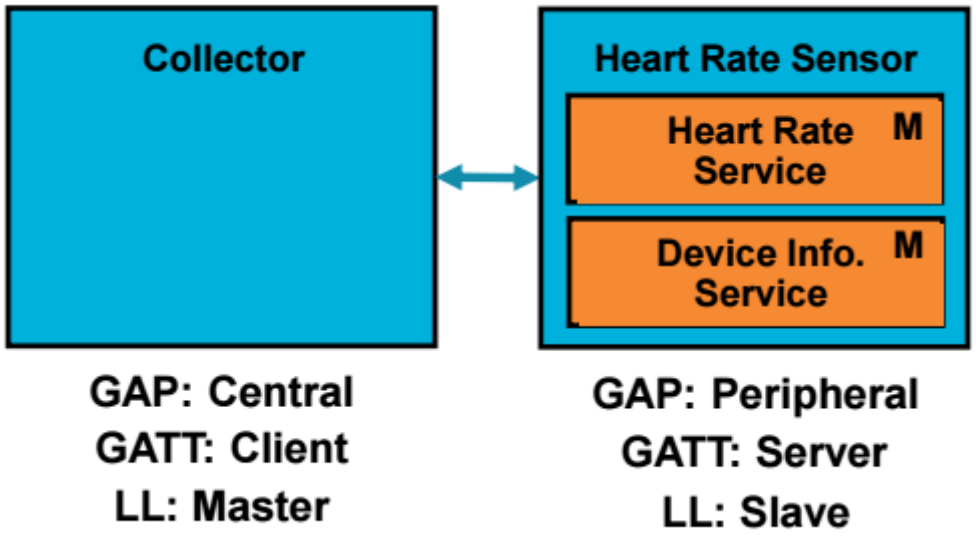

图 2.13: 心率 profile

CHAPTER 3

ROM 中的 Zephyr BLE 协议栈

3.1 ROM 的协议栈内容简介

- Host stack
  - BLE Host stack 绝大部分功能
  - 没有 mesh 代码
  - 没有经典蓝牙 Host 代码
  - 对应代码目录是 subsys/bluetooth/host，Zephyr 版本是 1.9.0
- Hci transport drivers
  - 标准 HCI 接口
  - 应用层直接可以调用
- Bluetooth Controller
  - 单模 LE 控制器实现
  - 对应代码不是 subsys/bluetooth/controller 目录的代码

3.2 ROM 中的 Zephyr Host 源码结构简介

- 源码结构如下表所示

表 3.1: 源码结构

源文件	API 接口文件	功能
subsys/bluetooth/host/gatt.c	include/gatt.h	GATT 层的实现
subsys/bluetooth/host/att.c	include/att.h	ATT 层的实现
subsys/bluetooth/host/smp.c subsys/bluetooth/host/key.c subsys/bluetooth/host/storage	include/storage.h	SMP 层的实现
subsys/bluetooth/host/l2cap.c	include/l2cap.h	L2CAP 层的实现
subsys/bluetooth/host/conn.c subsys/bluetooth/host/hci_core.c	include/conn.h include/bluetooth.h	GAP 层的实现

CHAPTER 4

Zephyr BLE 协议栈接口调用流程简介

4.1 Zephyr 协议栈初始化

- 协议栈初始化函数
  - bt_enable(bt_ready_cb_t cb)
  - 每个 Zephyr BLE 应用使用协议栈之前，都必须调用此函数
  - bt_enable 通常是在主线程被调用，但是其它回调函数是 WorkQueue 被调用的，对一些时序有要求的初始化，需要注意

4.2 GATT 接口使用

4.2.1 GATT Server

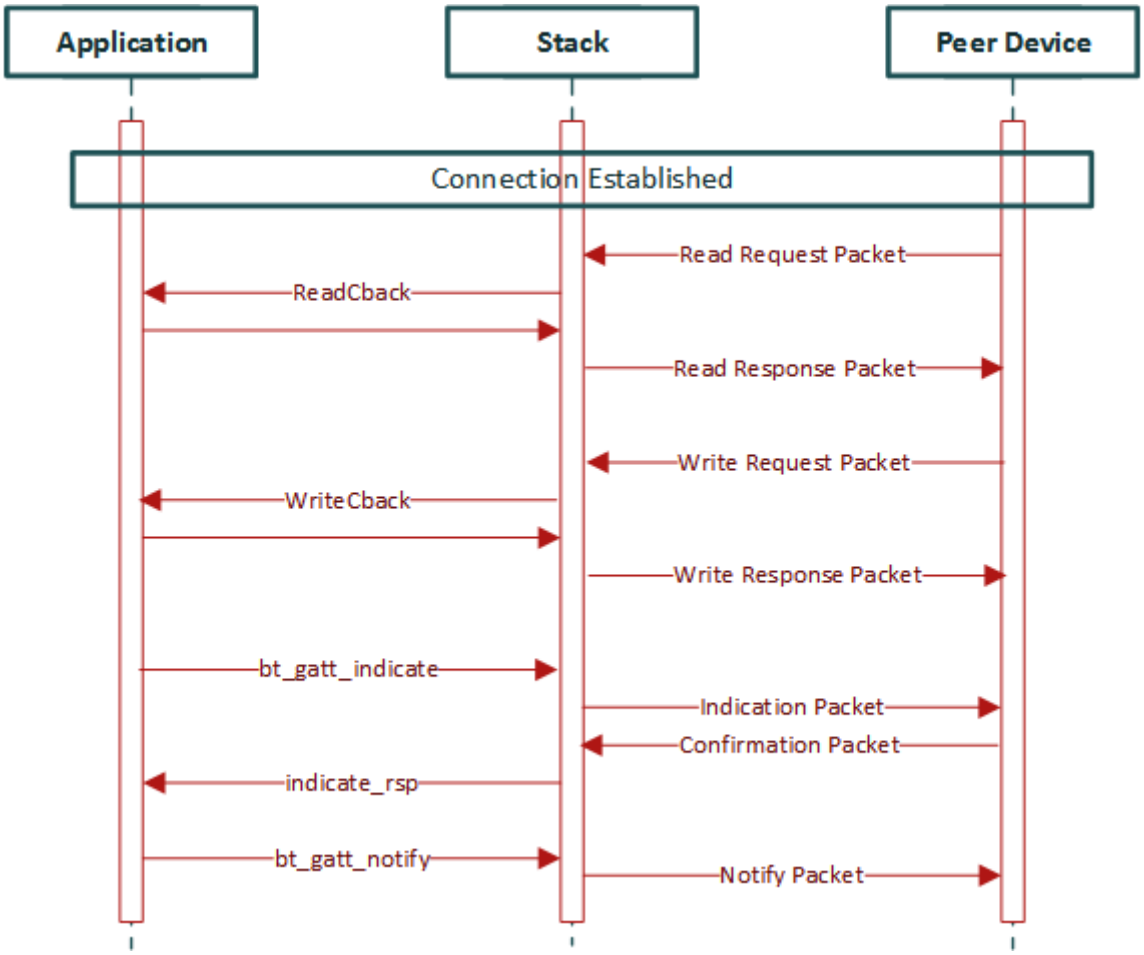

图 4.1: GATT Server 接口调用示意图

- 读回调函数
  - (* read)(conn, attr, buf, len, offset)
  - 实现一个来自 GATT client 的读操作
  - 可以不向 att 注册，stack 返回 att database 属性值
- 写回调函数
  - (* write)(conn,attr,buf,len,offset,flags)
  - 实现一个来自 GATT Client 的写操作
  - 可以不向 att 注册，stack 将写到 att database 属性值
- 发送指示
  - bt_gatt_indicate(conn, params)

- 发送通知
  - bt_gatt_notify(conn, attr, data, len)

4.2.2 GATT Client

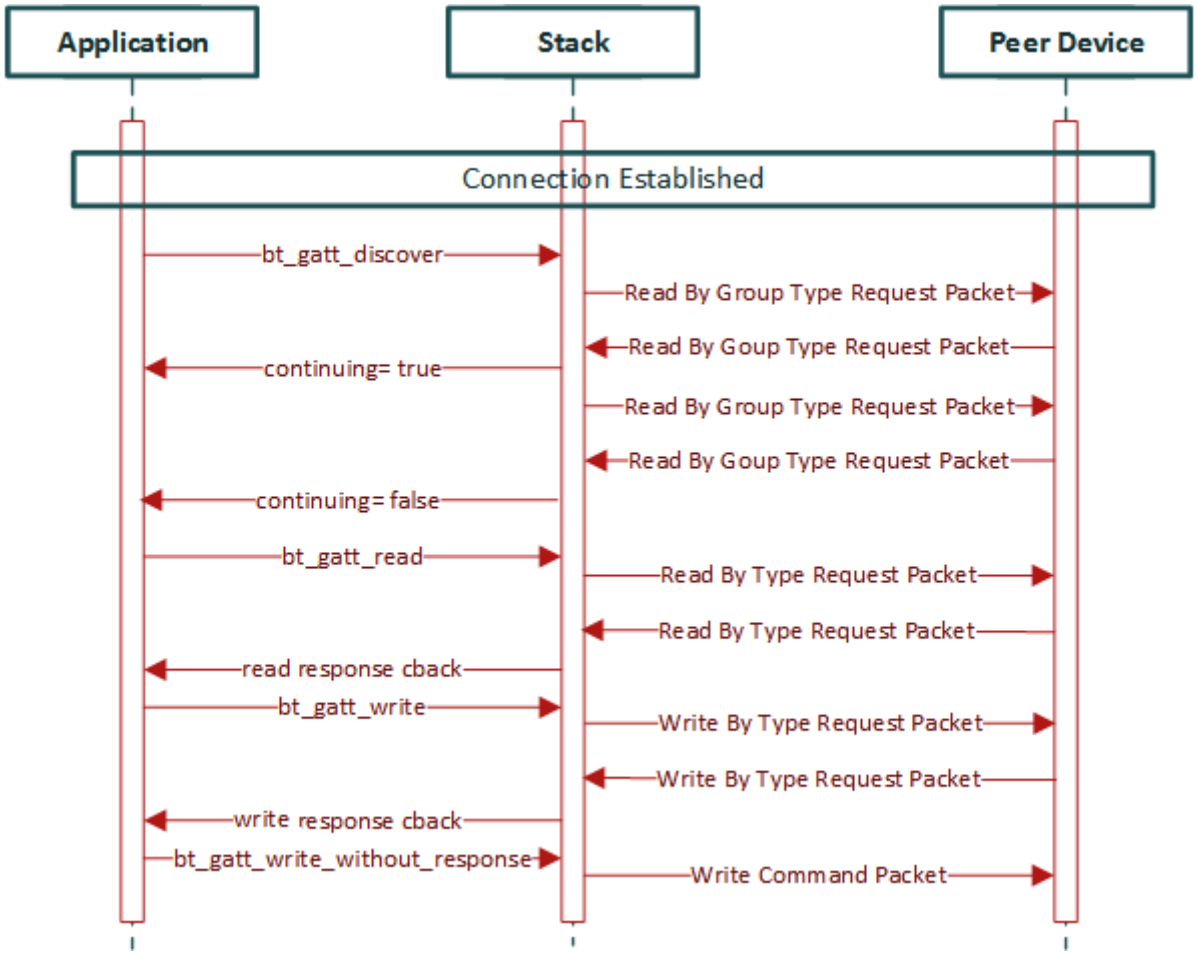

图 4.2: GATT Client 接口调用示意图

- 服务发现
  - bt_gatt_discover(conn, params)
  - primary Service 发现
  - Include service 发现
  - Characteristic 发现
  - Descriptors 发现
- 读请求
  - bt_gatt_read(conn, params)
- 写请求

  - bt_gatt_write(conn, params)

- 写命令
  - bt_gatt_write_without_response(conn, handle, data, length, sign)

4.3 GAP 接口使用

4.3.1 广播和扫描

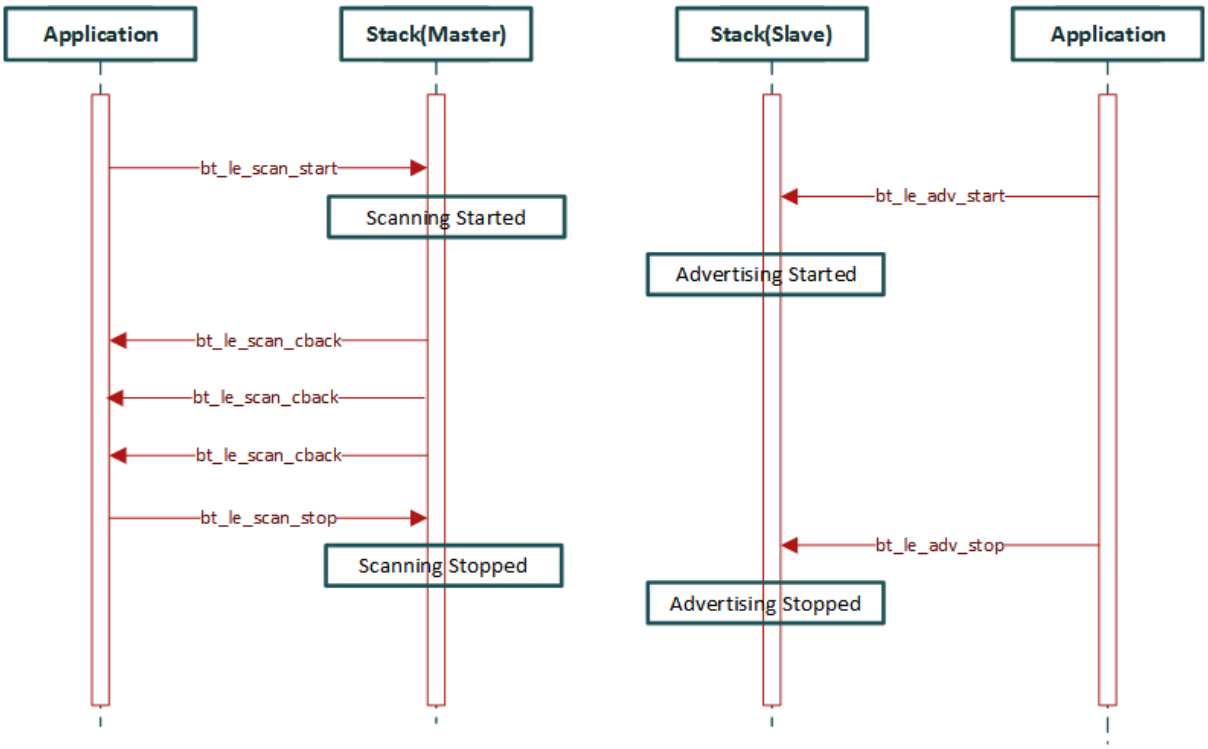

图 4.3: 广播和扫描接口调用示意图

- 开始广播
  - bt_le_adv_start(param, ad, ad_len, sd, sd_len)
- 停止广播
  - bt_le_adv_stop()
- 开始扫描
  - bt_le_scan_start(param, cb)
- 停止扫描
  - bt_le_scan_stop()

4.3.2 连接管理

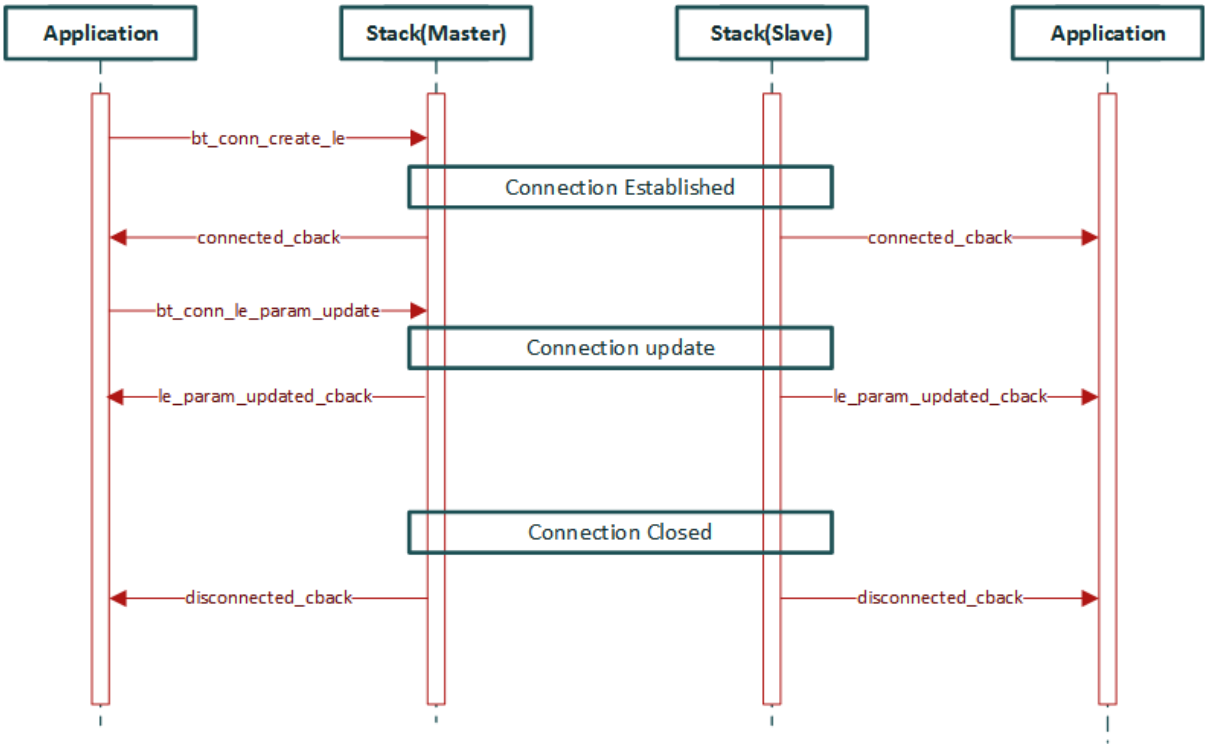

图 4.4: 连接过程示意图

- 建立连接
  - bt_conn_create_le(peer, param)
- 断开连接
  - bt_conn_disconnect(conn, reason)
- 连接参数更新
  - bt_conn_le_param_update(conn, param)

4.3.3 配对和加密

bt_conn_security(conn, sec)

- 配对
  - Master Side: bt_smp_send_pairing_req(conn)
  - Slave Side : bt_smp_send_security_req(conn)

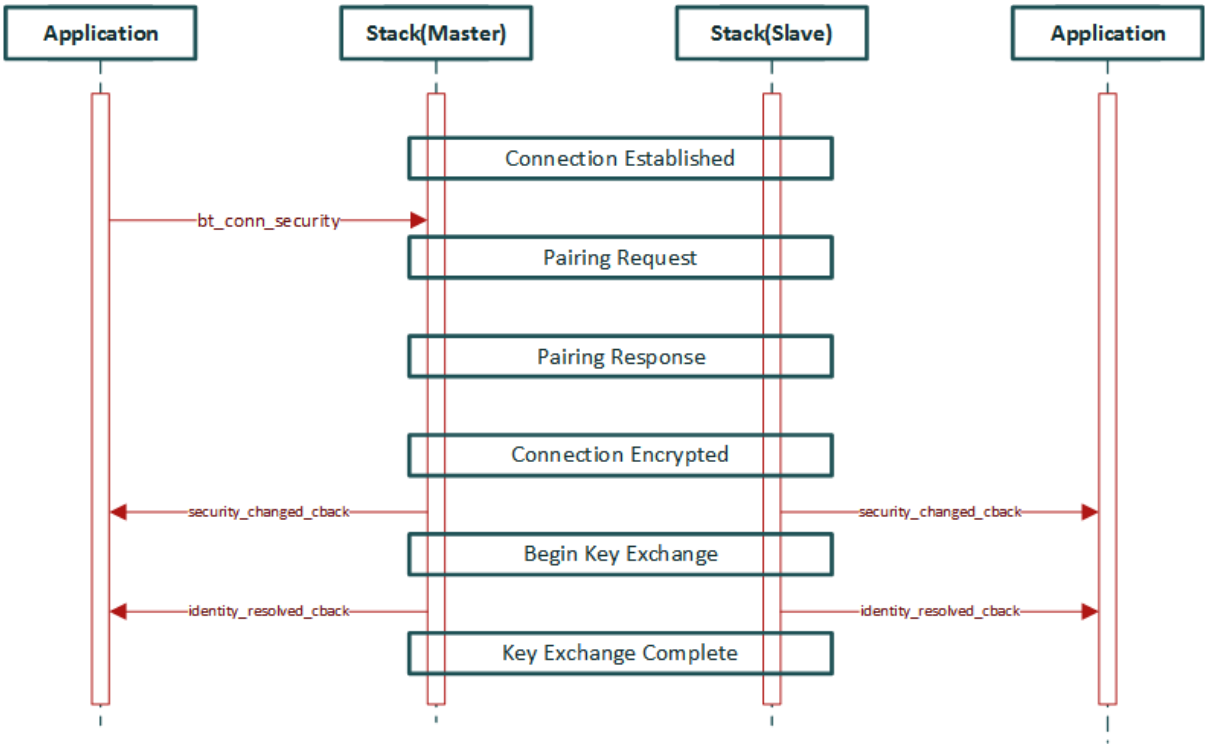

图 4.5: 配对过程示意图

- 加密
  - bt_conn_le_start_encryption

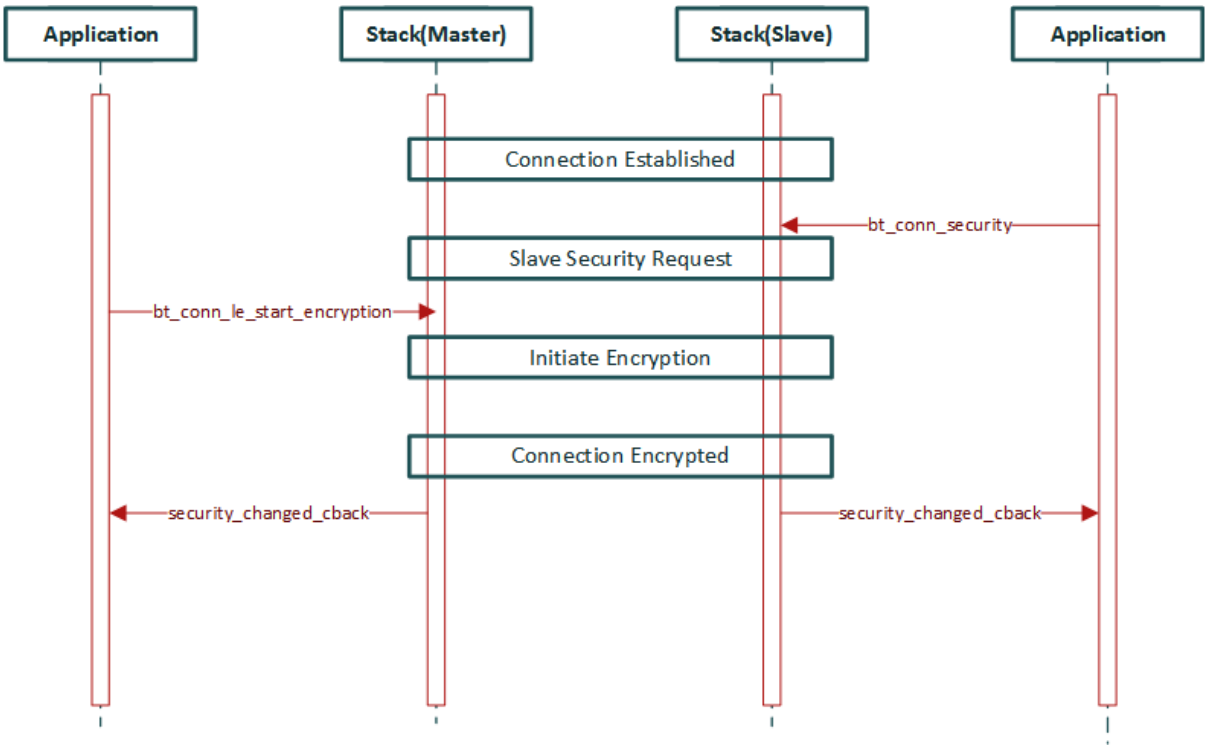

图 4.6: 加密过程示意图

CHAPTER 5

Zephyr BLE 协议栈数据接收和发送简介

5.1 BLE 协议栈的线程交互

- main 线程
- Tx 线程
- Rx 线程
- Hci driver 中断

5.2 BLE 协议栈的数据发送

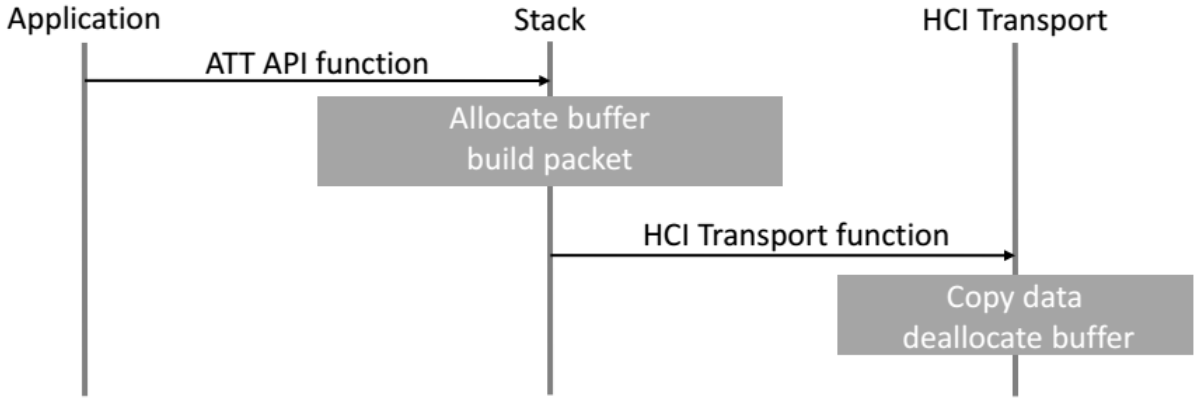

图 5.1: 发送数据流程图

- Host 协议栈分配 Tx buffer
- 应用层的数据拷贝到 Host 协议栈 Tx buffer
- HCI transport 收到 LL 层的数据包完成事件释放 buffer

5.3 BLE 协议栈的数据接收

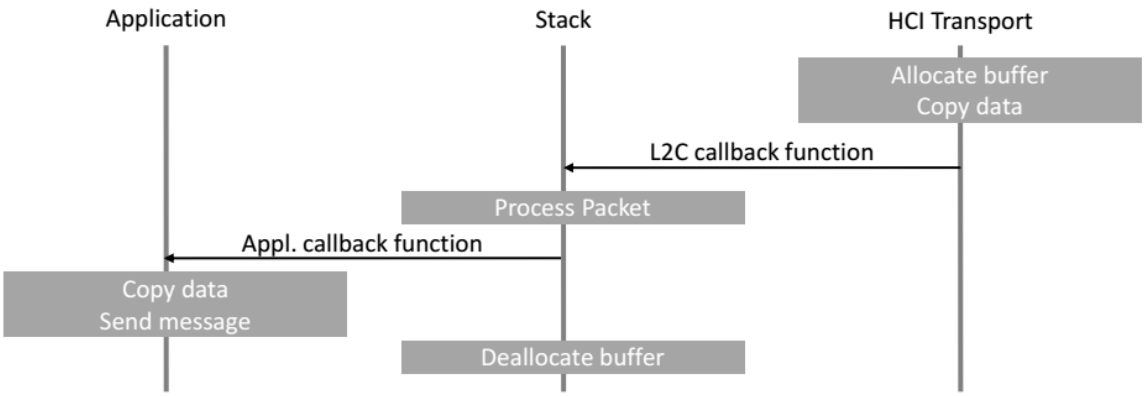

图 5.2: 接收数据流程图

- HCI transport 分配 Rx buffer
- 应用层从协议栈拷贝数据
- 协议栈释放 buffer

5.4 BLE 协议栈的数据接发使用注意事项

- 线程栈空间大小
  - Zephyr 协议栈 RX 线程栈较其它线程会大一些，不建议在 RX 线程的回调函数继续消耗更多的栈
- 数据接收过程中的流控
  - 发送流程有流控
  - 接收流程没有流控，但接收 buffer 分配完后，BLE 控制器的数据因为没有 buffer，会一直进行重发，直到 Host CPU 接收有 buffer 空闲
- Tx 和 Rx 流程资源竞争
  - Tx path 的 buffer 释放依赖 HCI transport Rx 包完成事件
  - Rx 完成事件 buffer 和 Rx 数据 buffer 共享 HCI transport 的 RX buffer
  - BLE 接收过程，控制器没有流控，如果对端设备发送数据速率过快，就会将 HCI transport RX buffer 分配耗尽

- Tx path 可能会因为 RX buffer 耗尽而阻塞住
- 因此，在应用程序设计时，尽量避免 Rx 线程直接发送数据，导致死锁。

CHAPTER 6

Zephyr BLE 协议栈配置

- BT_HCI_HOST
  - ROM 协议栈已经配置为支持，应用开发者不需要配置
- BT_HCI_CMD_COUNT
  - 用于 HCI 层发送命令的 buffer 个数，默认配置为 2
- BT_RX_BUF_COUNT
  - 用于接收控制器的 ACL 包或 HCI 事件的 buffer 个数，默认配置为 3
- BT_RX_BUF_LEN
  - 用于接收控制器的 ACL 包或 HCI 事件的 buffer 大小，默认配置为 76
- BT_HCI_TX_STACK_SIZE
  - 协议栈 TX 线程栈大小配置，默认为 256
- BT_HCI_TX_PRIO
  - 协议栈 TX 线程的优先级配置，默认为 7
- BT_HCI_RESERVE
  - 用于接收控制器的 ACL 包或 HCI 事件的 buffer，是否需要预留空间给 HCI 驱动使用，ROM HCI 驱动不需要，配置为 0
- BT_RECV_IS_RX_THREAD
  - ROM 协议栈已经配置为支持，应用开发者不需要配置
- BT_RX_STACK_SIZE
  - 协议栈 RX 线程栈大小配置，默认配置为 1024
- BT_RX_PRIO

  - 协议栈 RX 线程优先级配置，默认配置为 8
- CONFIG_BT_HOST_CRYPTO
  - 加密库支持
  - ROM 协议栈已经配置为支持，应用开发者不需要配置
- BT_CONN
  - ROM 协议栈已经配置为支持，应用开发者不需要配置
- BT_HCI_ACL_FLOW_CONTROL
  - ROM 协议栈已经配置为不支持，应用开发者不需要配置
- BT_L2CAP_TX_BUF_COUNT
  - 用于 L2CAP TX buffers 的个数，默认配置为 4
- BT_L2CAP_TX_MTU
  - L2CAP 层发送 MTU 大小配置，由于支持 SMP，默认配置为 65
- BT_CONN_TX_MAX
  - 最大允许发送给控制器 buffer 的个数，默认配置为 4
- BT_L2CAP_TX_USER_DATA_SIZE
  - 用于 L2CAP 层 TX buffers 用户数据大小，默认为 4
- BT_ATT_PREPARE_COUNT
  - 用于支持 ATT prepare write bufffers 的个数，默认为 0，即不支持。
- BT_ATT_TX_MAX
  - ATT 层可以缓冲的最大请求 buffer 个数，默认为 2
  - 设置范围为 1~BT_L2CAP_TX_BUF_COUNT
  - 如果应用尝试发送更多 ATT 请求过去，将会被阻塞。
- BT_SMP
  - 安全协议支持，默认已经支持，用户不需要配置
- BT_PRIVACY
  - 私有特征支持，使能后，将使用私有地址
- BT_RPA_TIMEOUT
  - 私有地址更新时间，默认为 900 秒 (15 分钟)
- BT_SIGNING
  - 数据签名支持，默认已经支持
- BT_L2CAP_DYNAMIC_CHANNEL
  - L2CAP 动态通道支持，默认支持

- BT_GATT_CLIENT
  - GATT 客户端支持，默认支持
- BT_MAX_PAIRED
  - 最大配对设备的支持
- BT_DEVICE_NAME
  - 蓝牙设备名配置
- BT_DEVICE_APPEARANCE
  - 蓝牙设备 appearance 配置

List of Figures

List of Tables