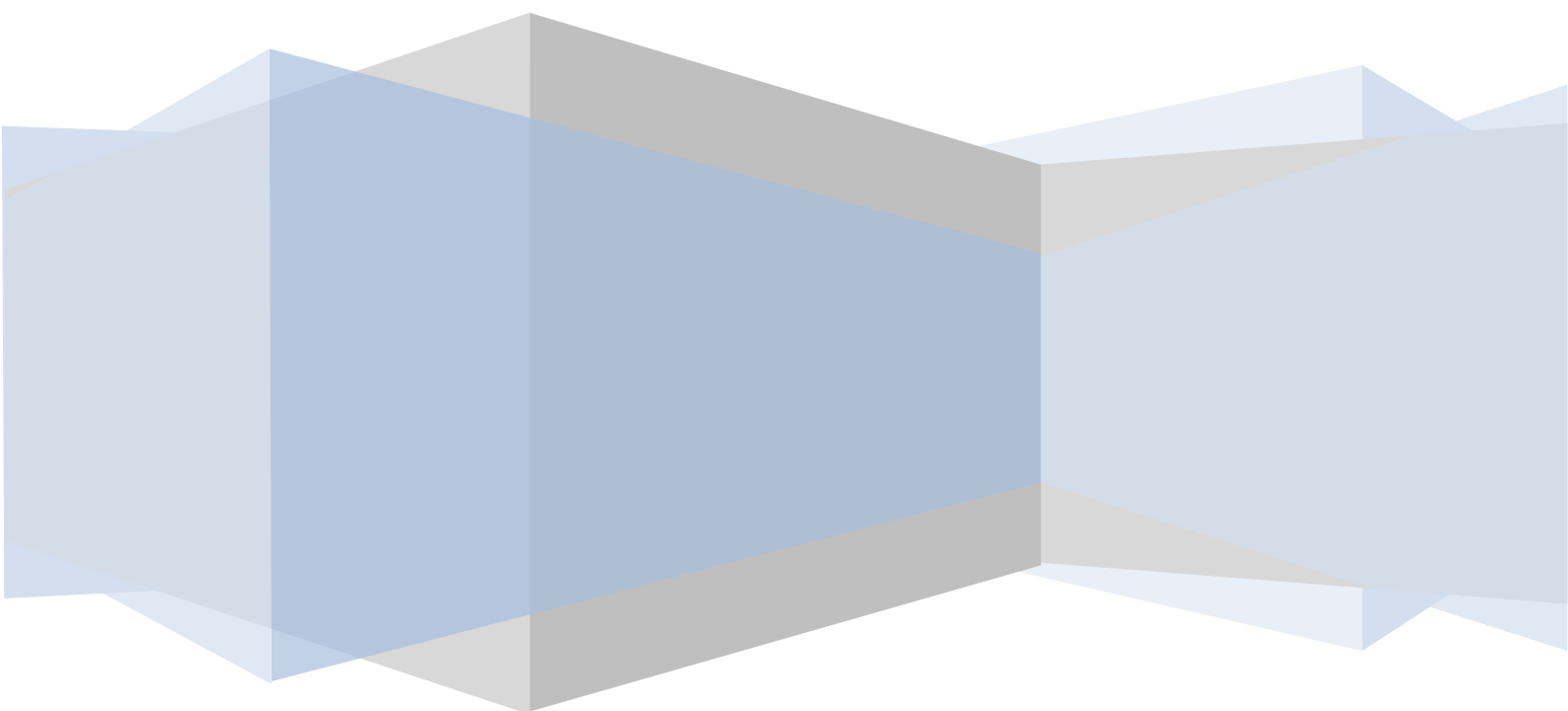


奥格无线蓝牙标准 BLE 部分协议译文 V1.0

译者: rf_smart



VOLUME 6 - CORE SYSTEM PACKAGE [LOW ENERGY CONTROLLER VOLUME]	11
PART A 物理层规范	11
1、简介	11
2、频率带宽和信道分配	11
3、发送特性	12
3.1 调制特性	12
3.2 杂散	12
3.2.1 调制频谱	12
3.2.2 带内杂散	12
3.2.3 带外杂散	13
3.3 射频频率误差	13
4、接收特性	13
4.1 实际灵敏度	13
4.2 干扰性能	13
4.3 带外阻塞	14
4.4 互调特性	14
4.5 最大使用功率电平	14
4.6 基准信号	14
5、附录	14
5.1 正常工作条件（NOC）	14
5.1.1 正常工作温度及湿度	14
5.1.2 正常供电电压	15
5.2 极限供电电压	15
5.2.1 极限温度和湿度	15
5.2.2 极限供电电压	15
6 附录 B-工作条件	15
PART B 链路层规范	16
1、一般性描述	16
1.1 链路层状态	16
1.1.1 状态与角色组合限制	17
1.2 位定义	17
1.3 设备地址	17
1.4 物理信道	18
1.4.1 广播及数据信道索引	18
2、空中接口数据包	18
2.1 数据包格式	18
2.1.1 引导码	18

2.1.2 访问地址	19
2.1.3 PDU.....	19
2.1.4 CRC.....	19
2.2 预留以备用	19
2.3 广播信道 PDU	19
2.3.1 广播 PDU 类型.....	20
2.3.1.1 ADV_IND(无目的地广播).....	20
2.3.1.2 ADV_DIRECTED (指定地址广播)	20
2.3.1.3 ADV_NONCONN_IND	20
2.3.1.4 ADV_SCAN_IND= ADV_DISCOVER_IND	21
2.3.2 扫描 PDUs.....	21
2.3.2.1 SCAN_REQ	21
2.3.2.2 SCAN_RSP	21
2.3.3 初始化 PDUs.....	22
2.3.3.1 CONNECT_REQ.....	22
2.4 数据信道 PDU.....	23
2.4.1 LL DATA PDU.....	24
2.4.2 LL CONTROL PDU	24
2.4.2.1 LL_CONNECTION_UPDATE_REQ.....	24
2.4.2.2 LL_CHANNEL_MAP_REQ	25
2.4.2.3 LL_TERMINATE_IND	25
2.4.2.4 LL_ENC_REQ.....	25
2.4.2.5 LL_ENC_RSP	25
2.4.2.6 LL_START_ENC_REQ.....	26
2.4.2.7 LL_START_ENC_RSP.....	26
2.4.2.8 LL_UNKNOW_RSP	26
2.4.2.9 LL_FEATURE_REQ.....	26
2.4.2.10 LL_FEATURE_RSP.....	26
2.4.2.11 LL_PAUSE_ENC_REQ.....	26
2.4.2.12 LL_PAUSE_ENC_RSP	26
2.4.2.13 LL_VERSION_IND.....	26
2.4.2.14 LL_REJECT_IND.....	27
3、位流处理.....	27
3.1 差错检测	27
3.1.1 CRC 产生	27
3.2 数据白化.....	28
4、空中接口协议	28
4.1 帧间间隔.....	28
4.2 时序要求.....	28
4.2.1 时钟精度	28
4.2.2 睡眠时钟精度	28
4.3 链路层设备过滤	28
4.3.2 广播过滤策略	29

4.3.3 扫描过滤策略	29
4.3.4 初始化者过滤策略.....	29
4.4 非连接状态	29
4.4.1 待机状态	29
4.4.2 广播状态	29
4.4.2.1 广播信道选择	30
4.4.2.2 广播间隔.....	30
4.4.2.3 可连接非定向事件类型	30
4.4.2.4 可连接定向事件类型.....	31
4.4.2.5 可扫描非定向事件类型	32
4.4.2.6 不可连接非定向事件类型	33
4.4.3 扫描状态	33
4.4.3.1 被动扫描.....	33
4.4.3.2 主动扫描.....	34
4.4.4 初始化状态	34
4.5 连接状态.....	35
4.5.1 连接事件	35
4.5.2 管理超时	35
4.5.3 连接事件传输窗口	36
4.5.5 连接设置——从角色.....	37
4.5.6 关闭连接事件	37
4.5.7 窗口扩展	38
4.5.8 数据信道索引选择.....	38
4.5.8.1 信道分类.....	38
4.5.8.2 信道选择.....	38
4.5.9 应答和流控	39
4.5.9.1 流控	39
4.6 属性支持.....	39
4.6.1 LE 加密.....	40
5.链路层控制.....	40
5.1 链路层控制流程	40
5.1.1 连接更新流程	40
5.1.2 信道表更新过程.....	41
5.1.3 加密流程.....	42
5.1.3.1 加密开始流程.....	42
5.1.3.2 加密暂停处理	43
5.1.4 Feature 交换流程.....	44
5.1.5 版本交换	44
5.1.6 终止处理	44
5.2 流程应答超时	45
VOLUM 3 - CORE SYSTEM PACKAGE[HOST VOLUME]	46
PART A L2CAP(LOGICAL LINK CONTROL AND DAPTATION PROTOCOL PECIFICATION)	46

1、介绍	46
1.1 L2CAP 特征	46
1.2 假设	47
1.3 范围	48
1.4 术语	48
2、一般操作	51
2.1 管道 ID	51
2.2 设备间的操作	51
2.3 协议层间操作	52
2.4 模式间操作	53
2.5 映射信道到逻辑链路	53
3、数据包格式	54
3.1 基本 L2CAP 面向连接管道	54
3.2 基本 L2CAP 模式中的无连接数据管道	54
3.3 重传、流控、数据流模式的面向连接管道	55
3.3.1 L2CAP 头域	55
3.3.2 控制域, 2 或 4 字节	55
3.3.3 L2CAP SDU 长度域, 2 字节	57
3.3.4 信息负载域	57
3.3.5 帧校验序列, 2 字节	58
3.3.6 不可用帧检测	58
3.3.7 不可用帧检测算法	58
4、信令包格式	59
4.1 拒绝命令: CODE = 0x01	61
4.2 连接请求 CODE = 0x02	62
4.3 连接应答 CODE = 0x03	63
4.4 配置请求 CODE = 0x04	64
4.5 配置应答 CODE = 0x05	65
4.6 断开连接请求	66
4.7 断开连接应答 CODE = 0x07	67
4.8 反射请求 CODE = 0x08	67
4.9 反射应答 CODE = 0x09	67
4.10 信息请求 CODE = 0x0A	67
4.11 信息应答 CODE = 0x0B	68
4.12 扩展特征掩码	69
4.13 固定管道支持	70
4.14 创建管道请求 CODE = 0x0C	70
4.15 创建信道应答 CODE = 0x0D	71
4.16 移动管道请求 CODE = 0x0E3	72
4.17 移动管道应答 CODE = 0x0F	72

4.18 移动管道认证 CODE = 0X10	73
4.19 移动管道确认请求 CODE = 0x11.....	73
4.20 连接参数更新请求 CODE = 0x12.....	73
4.21 连接参数更新应答 CODE = 0x13.....	74
5、配置参数选项	75
5.1 最大传输单元, MTU	75
PARTC – GAP(GENERIC ACCESS PROFILE)	76
1 介绍	76
1.1 概述	76
1.2 符号和规约	76
1.2.1 必要的状态符号.....	76
1.2 信号框图约定	76
1.2.3 定时器和计数器符号	77
2、协议概览.....	77
2.1 协议栈.....	77
2.2 协议角色	77
2.2.1 工作在 BR/EDR 物理层时的角色.....	77
2.2.2 工作在 LE 物理信道的角色.....	77
2.2.2.1 Broadcaster 角色.....	78
2.2.2.2 Observer 角色	78
2.2.2.3 Peripheral 角色.....	78
2.2.2.4 Central 角色.....	78
2.2.2.5 同时工作在多个 GAP 角色.....	78
2.3 用户要求和情境.....	79
2.4 PROFILE 基本原理.....	79
2.5 一致性.....	80
3、用户接口定义	80
3.1 用户接口层	80
3.2 蓝牙参数描述.....	80
3.2.1 蓝牙设备名(BD_ADDR).....	80
3.2.1.1 定义	80
3.2.1.1.1 LE-only 设备类型蓝牙设备地址.....	80
3.2.1.1.2 BR/EDR/LE 设备（双模）的蓝牙地址.....	80
3.2.1.2 用户接口层术语	80
3.2.1.3 描述	80
3.2.2 蓝牙设备名	80
3.2.2.1 定义.....	80
3.2.2.1.1 BR/EDR/LE 设备的蓝牙设备名.....	81
3.2.2.2 用户接口层术语	81
3.2.2.3 描述	81
3.2.3.2 用户层接口术语	81

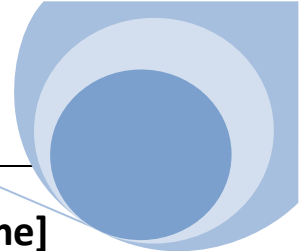
3.2.3.3 描述	81
3.2.4 设备种类	82
3.2.4.1 定义	82
3.2.4.2 用户接口术语	82
3.2.4.3 描述	82
3.2.4.4 用法	82
3.3 配对	82
9、LE 物理信道使用的工作模式和处理流程	82
9.1 广播模式和观察处理流程	82
9.1.1 广播模式	83
9.1.1.1 定义	83
9.1.1.2 条件	83
9.1.2 观察处理	83
9.1.2.1 定义	83
9.1.2.2 条件	83
9.2 发现模式和处理流程	83
9.2.1 要求	83
9.2.2 不可发现模式	84
9.2.2.1 描述	84
9.2.2.2 条件	84
9.2.3 限制发现模式	84
9.2.3.1 描述	84
9.2.3.2 条件	84
9.2.4 通用可发现模式	84
9.2.4.1 描述	84
9.2.4.2 条件	85
9.2.5 限制可发现处理	85
9.2.5.1 描述	85
9.2.5.2 条件	86
9.2.6 通用发现处理	86
9.2.6.1 描述	86
9.2.6.2 条件	86
9.2.7 设备名发现处理	87
9.2.7.1 描述	87
9.2.7.2 条件	87
9.3 连接模式和处理流程	87
9.3.2 不可连接模式	88
9.3.2.1 描述	88
9.3.2.2 条件	88
9.3.3 定向连接模式	88
9.3.3.1 描述	88
9.3.3.2 条件	88
9.3.4 非定向可连接模式	89
9.3.4.1 描述	89

9.3.4.2 条件	89
9.3.5 自动连接建立流程	90
9.3.5.1 描述	90
9.3.5.2 条件	90
9.3.6 通用连接建立过程	91
9.3.6.1 描述	91
9.3.6.2 条件	91
9.3.7 选择性连接建立流程	92
9.3.7.1 描述	92
9.3.7.2 条件	92
9.3.8 定向连接建立流程	92
9.3.8.1 描述	92
9.3.8.2 条件	92
9.3.9 连接参数更新流程	93
9.3.9.1 描述	93
9.3.9.2 条件	93
9.3.10 中断连接流程	93
9.3.10.1 描述	93
9.3.10.2 条件	93
9.4 绑定模式和处理	93
9.4.2.1 不可绑定模式	93
9.4.2.2 条件	93
9.4.3 可绑定模式	93
9.4.3.1 描述	93
9.4.3.2 条件	94
9.4.4 绑定处理	94
9.4.4.1 描述	94
9.4.4.2 条件	94
10 LE 加密处理	94
10.2 LE 加密模式	95
11、广播和扫描应答数据格式	96
11.1 AD 类型定义	96
11.1.1 服务 UUIDs	96
11.1.2 本地名	96
11.1.3 标志	96
11.1.4 装配标准数据	97
11.1.5 发射功率电平	97
11.1.6 加密管理 OOB	97
11.1.7 加密管理 TK 值	97
11.1.8 从机连接间隔范围	97
11.1.9 服务请求	97
11.1.10 服务数据	97
11.2 广播数据示例	97

12、GAP 特性	98
12.1 设备名特性	98
12.2 外观特征	98
12.3 外设隐私标志特征	98
12.4 重连接地址特征	99
12.5 外设优先连接参数特征	99
PARTG GATT(GENERIC ATTRIBUTE PROFILE)	101
1、介绍	101
1.1 简介	101
1.2 PROFILE 依赖	101
2、PROFILE 概述	101
2.1 协议栈	101
2.2 配置和角色	101
2.3 用户要求和脚本	102
2.4 特征值基本原理	102
2.5 属性协议	102
2.5.1 概述	102
2.5.2 属性缓冲	103
2.5.3 属性组	104
2.5.4 UUIDs	104
2.6 GATT 框架层级	104
2.6.1 概述	104
2.6.2 服务	104
2.6.3 包括服务	105
2.6.4 特征	105
2.7 配置广播	105
3 服务通用性要求	105
3.1 服务定义	105
3.2 包括定义	106
3.3 特征定义	106
3.3.1 特征声明	106
3.3.1.1 特征属性	107
3.3.1.2 特征值属性句柄	107
3.3.1.3 特征 UUID	107
3.3.2 特征值声明	107
3.3.3 特征描述声明	108
3.3.3.1 特征扩展属性	108
3.3.3.2 特征用户描述	108
3.3.3.3 客户端特征配置	109
3.3.3.4 服务端特征配置	109
3.3.3.5 特征描述格式	110

3.3.3.5.1 位序.....	110
3.3.3.5.2 格式化.....	110
3.3.3.5.3 指数.....	111
3.3.3.5.4 单元.....	111
3.3.3.5.5 命名空间.....	111
3.3.3.5.6 描述.....	111
3.3.3.5.6 特征聚合格式.....	112
3.4 GATT PROFILE 属性类型总结.....	112
4、GATT 特征要求.....	113
4.1 概述.....	113
4.2 特征支持和处理映射.....	113
4.3 服务配置.....	115
4.3.1 交换 MTU.....	115
4.4 根服务发现.....	115
4.4.1 发现所有根服务.....	115
4.4.2 通过服务 UUID 发现根服务.....	116
4.5 关系发现.....	117
4.5.1 发现包括服务.....	117
4.6 特征发现.....	117
4.6.1 发现服务所有特征.....	117
联系我们.....	118

历史	说明
V1.0	新建版本。



Volume 6 - Core System Package [Low Energy Controller volume]

PART A 物理层规范

1、简介

蓝牙低功耗(LE)设备工作在免费 2.4G ISM 频带。使用跳频收发器避免冲突和衰落。

LE 射频应该有发送器或接收器，或者都有。

LE 射频应该满足设备制造商声明的工作条件规定的要求。参考章节 5.1 和 5.2。

射频参数应该按 LE RF PHY 测试规范描述的方法进行测试。

本规范基于已为以直地区建立的规则：欧洲，日本，北美，台湾，韩国和日本。以下列出的标准文档只做为提供的信息，并可以任何时候修改或变更。

蓝牙 SIG 维护用于工作在 2.4G ISM 带宽的蓝牙技术的在线规则数据库，相关链接：

<https://www.bluetooth.org/regulatory/newindex.cfm>

欧洲：

通过的标准：European Telecommunications Standards Institute, ETSI

文档：EN 300 328, EN 301 489, ETS 300-826

日本：

通过的标准：Japanese Radio Law, JRL

文档：Japanese Radio Law: Article 4.3, Article 28, Article 29, Article 38

Radio Equipment Regulations: Article 5, Article 6, Article 7, Article 14,

Article 24, Article 9.4, Article 49.20.1.C.2, Article 49.20.1.E.3

Radio Law Enforcement Regulations: Article 6.2, Article 6.4.4.1, Article 7

北美：

通过的标准：Federal Communications Commission, FCC, USA

文档：CFR47, Part 15: Sections 15.205, 15.209 and 15.247

认证标准：Industry Canada, IC, Canada

文档：RSS-210 and RSS139

台湾：

认证标准：National Communications Commission, NCC

文档：Low Power 0002 (LP0002); Low-power Radio-frequency Devices Technical Regulations

韩国：

认证标准：Korea Communications Commission, KCC

文档：Rules on Radio Equipment 2008-116

中国：

认证标准：Ministry of Industry and Information Technology, MIIT

文档：MIIT regulation [2002]353

2、频率带宽和信道分配

LE 系统工作在 2.4G ISM 频带，频率范围：2400-2483.5MHz。LE 系统使用 40 个 RF 信道。这些 RF 信道以 $2402+k*2\text{MHz}$, $k=0, \dots, 39$ 为中心频点。

Regulatory Range	RF Channels
2.400-2.4835 GHz	$f=2402+k*2\text{ MHz}$, $k=0, \dots, 39$

Table 2.1: Operating frequency bands

3、发送特性

本章节所描述的要求是指定 LE 设备上天线连接器上的功率等级。如果设备没有连接器，则以 0dBi 天线做为参考。

由于很难对辐射功能进行准确测量，集成天线的系统在 LE RF 质量测试中需要临时提供天线连接器。对于发送器，最大输出功率等级设置需要在表 3.1 定义的范围內：

Minimum Output Power	Maximum Output Power
0.01 mW (-20 dBm)	10 mW (+10 dBm)

Table 3.1: Transmission power

设备输出功率控制可以由本地修改，比如，为了优化功耗消耗或减少与其它设备的冲突。

3.1 调制特性

调制方式为高斯频移键控(GFSK)，带宽时间积 $BT=0.5$ 。调制系数在 0.45-0.55 之间。二进制 1 代表正频率偏移，0 代表负频率偏移。

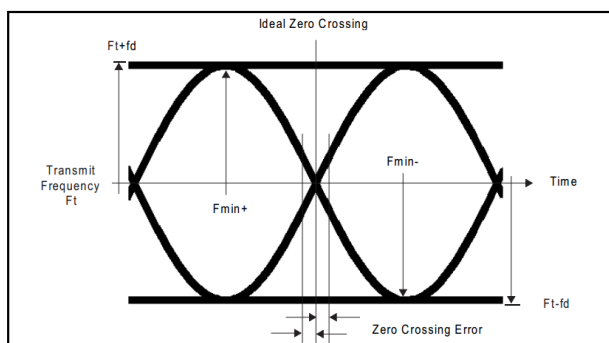


Figure 3.1: GFSK parameters definition

对于每一个传输的频偏：

$$F_{min} = \min\{|F_{min+}|, |F_{min-}|\}$$

相对于一个 1010 序列，应该不小于对应传输频率频偏的 $\pm 80\%$ ，相对于一个 00001111 序列。

此外，最小频偏绝对不能小于 185KHz。数据传输速度为 1Mbps。位符号时序精度应该好于 $\pm 50\text{ppm}$ 。

零交叉误差是理想符号周期和测试交叉时间的差值。这个值必须小于 $\pm 1/8$ 个位符号周期。

3.2 杂散

3.2.1 调制频谱

对于产品需要满足 FCC 15.247 部分的标准，最小传输频谱的 6dB 带宽应该小于 500kHz，使用 100KHz 带宽分辨率测量。

3.2.2 带内杂散

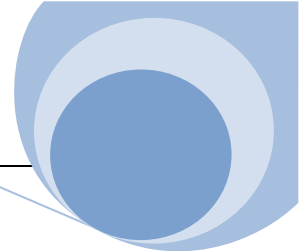
邻道功率定义为从载波外至少 2MHz 信道的功能。邻道功率定义为 1MHz 带宽内功率的总和。

频谱测量在 100KHz 分辨率及平均功率检测模式下进行。设备应该在一个 RF 信道上传输，这个信道中心频率为 M，邻道功率应该在相距中心频率 1MHz RF 频率 N 上测量。传输应该传输一个伪随机负载数据作为测试。

Frequency offset	Spurious Power
2 MHz ($ M-N = 2$)	-20 dBm
3 MHz or greater ($ M-N \geq 3$)	-30 dBm

Table 3.2: Transmit Spectrum mask

中心频率外超过 3 倍 1MHz 带宽的功率异常是允许的。但这些异常功率频点的极限值为 -20dBm。



3.2.3 带外杂散

设备生产厂家负责根据销售的目标厂家制定 ISM 带外杂散要求。

3.3 射频频率误差

数据包间的中心频率误差不能超过 150KHz，包括初始频率偏差和频率漂移。任何数据包内的频率偏移必须小于 50KHz。偏移率必须小于 400Hz/μs。

数据包内传输中心频率漂移限制如表 3.3：

Parameter	Frequency Drift
Maximum drift	±50 kHz
Maximum drift rate ¹	400 Hz/μs

Table 3.3: Maximum allowable frequency drifts in a packet

1. The maximum drift rate is allowed anywhere in a packet.

4、接收特性

本章节基准的灵敏度为-70dBm。数据包错误率定义为位误码率(BER)，应该用于所有接收器特性测试中。

4.1 实际灵敏度

实际灵敏度定义为接收器 BER 为 0.1%时的输入功率。

接收器实际灵敏度应该小于或等于-70dBm，必须在章节 3 发送器规范中定义的以下参数条件范围中保证这个灵敏度：

初始化频率偏移
频率漂移
符号速率
频偏

4.2 干扰性能

干扰性能需要目标信号基准电平以上 3dB 的值测量。如果干扰信号超出 2400–2483.5MHz 范围，此时需要遵守带外阻塞规范。测量分辨率为 1MHz。目标信号及干扰信号都应该是章节 4.6 定义的基准信号。在表 4.1 列出的所有信号干扰比下，BER 都需要小于等于 0.1%。

Frequency of Interference	Ratio
Co-Channel interference, $C/I_{\text{co-channel}}$	21 dB
Adjacent (1 MHz) interference ¹ , $C/I_{1 \text{ MHz}}$	15 dB
Adjacent (2 MHz) interference ¹ , $C/I_{2 \text{ MHz}}$	-17 dB
Adjacent (≥ 3 MHz) interference ¹ , $C/I_{\geq 3 \text{ MHz}}$	-27 dB
Image frequency Interference ^{12 3} , C/I_{Image}	-9 dB
Adjacent (1 MHz) interference to in-band image frequency ¹ , $C/I_{\text{Image} \pm 1 \text{ MHz}}$	-15 dB

Table 4.1: Interference performance

1. If two adjacent frequency specifications from Table 4.1 are applicable to the same frequency, the more relaxed specification applies.
2. In-band image frequency
3. If the image frequency $\neq n \cdot 1 \text{ MHz}$, then the image reference frequency is defined as the closest $n \cdot 1 \text{ MHz}$ frequency for integer n .

任何不能满足要求的频率称为伪响应 RF 信道。目标频率 2MHz 外的频带，不包括镜像频率和镜像频率 $\pm 1\text{MHz}$ ，允许五个伪响应 RF 信道。在这些伪响应 RF 信道，需要满足 $C/I = -17\text{dB}$ 的要求。

4.3 带外阻塞

带外阻塞用于 2400–2483.5MHz 外的干扰信号。带外抑制能力需要在基准灵敏度 3dB 以上测量。目标信号应该是章节 4.6 描述的基准信号，其中心频率为 2440MHz。误码率必须小于等于 0.1%。带外阻塞应该满足以下要求：

Interfering Signal Frequency	Interfering Signal Power Level	Measurement resolution
30 MHz – 2000 MHz	-30 dBm	10 MHz
2003 – 2399 MHz	-35 dBm	3 MHz
2484 – 2997 MHz	-35 dBm	3 MHz
3000 MHz – 12.75 GHz	-30 dBm	25 MHz

Table 4.2: Out-of-band suppression (or rejection) requirements.

超过 10 个异常是允许的，需要根据给定的 RF 信道并且是 1MHz 整数倍频率的中心：

- 对这些伪响应频率的最小 7 个，在最少 -50dBm 基础上减少干扰是允许的，以达到 BER 小于等于 0.1%。
- 对这些伪响应频率的最大 3 个，干扰将更低。

4.4 互调特性

实际灵敏度性能，BER 小于等于 0.1%，必须满足以下条件：

- 目标信号必须在功率为基准灵敏度以上 6dB 的 f_0 频率上。目标信号必须是章节 4.6 定义的基准信号。
- 静态正弦波信号必须在功率为 -50dBm 的 f_1 频率上。

干扰信号必须在功率为 -50dBm 的 f_2 频率上。干扰信号必须是章节 4.6 定义的基准信号。

频率 f_0, f_1, f_2 必须按下式选择： $f_0 = 2 * f_1 - f_2$ ，且 $|f_2 - f_1| = n * 1$ ，其中 $n = 3, 4$ 或 5。系统必须使用三个中的一个。

4.5 最大使用功率电平

接收者可以使用的最大可用的输入电平必须大于 -10dBm。并且 BER 必须小于或等于 0.1% 在 -10dBm 输入功率模式下。输入信号必须是章节 4.6 定义的基准信号。

4.6 基准信号

基准信号定义为：

调制模式=GFSK

调制系数=0.5 $\pm$ 1%

BT=0.5 $\pm$ 1%

码率=1Mbps $\pm$ 1ppm

目标信号调制数据=PRBS9

干扰信号高数数据=PRBS15

频率精度好于 $\pm 1\text{ppm}$

5、附录

5.1 正常工作条件（NOC）

5.1.1 正常工作温度及湿度

正常工作温度必须产品厂家进行说明。正常测试温度应该在正常温度范围的 $\pm 10^\circ\text{C}$ 以内。正常工作常温 $\pm 10^\circ\text{C}$ 的温度必须不能超过章节 5.2.1 的限制。

正常工作湿度必须产品厂家进行说明。正常测试条件的空中湿度需要在 NOC 范围内。

5.1.2 正常供电电压

正常测试条件下设备的正常测试电压必须按生产厂家描述的进行供电。

5.2 极限供电电压

5.2.1 极限温度和湿度

极限温度限制由厂家定义最小和最大的温度范围。

对于极限条件，空中湿度必须在厂家声明的工作湿度范围内，参考 5.1.1 章节。

5.2.2 极限供电电压

极限供电电压根据产品电源电压限定。

如果产品设计工作另一个系统的一部分或一个产品的一部分，极限供电电压将使用产品或系统的极限电压。

适合上层和底层的极限供电电压必须由产品厂家声明。

6 附录 B-工作条件

LE 射频参数必须兼容以下条件：

Parameter	Temperature	Power supply
Output power	EOC	EOC
In-band emissions	EOC	EOC
Modulation characteristics	NOC	NOC
Carrier frequency offset and drift	EOC	EOC
Receiver sensitivity	EOC	EOC
C/I and selectivity performance	NOC	NOC
Blocking performance	NOC	NOC
Intermodulation performance	NOC	NOC
Maximum input signal level	NOC	NOC

EOC = Extreme Operating Conditions

NOC = Normal Operating Conditions

注意：LE 接收器参数的确认使用直接测试模式执行。参考 Vol.6,Part F。

PARTB 链路层规范

1、一般性描述

1.1 链路层状态

链路层操作可以描述为以下五种状态的状态机：

- Standby State
- Advertising State
- Scanning State
- Initiating State
- Connection State

链路层状态机只允许同一时刻处于一个状态。链路层必须有至少一个支持广播状态或扫描状态的链路层状态机。链路层可能有多个链路层状态机实例。多个状态机内的一些状态机和角色的组合被禁止(参考章节 1.1.1)。

处于 Standby State 的链路层不传输和接收任何数据，可以由其它任何状态进入。

处于 Advertising State 的链路层可以发送广播信道包，并可能监听和应答由这些广播信道包触发的响应，设备处于该状态的称为广播者，该状态由 standby state 进入。

处于 Scanning State 的链路层将监听正在广播的设备的广播信道包。处于该状态被称为扫描者，该状态由 standby state 进入。

处于 Initiating State 的链路层将监听特定设备的广播信道包并应答这些数据以初始化一个连接，设备处于该状态被称为初始化者，该状态由 standby state 进入。

处于 Connection State 可以由 Initiating State 或 Advertising State 进入，设备在该状态可称为一个连接者。

在连接状态中两种角色被定义：

- Master 角色
- Slave 角色

如果从 Initiating State 进入的，称为主机，从 Advertising State 进入的称为从机。

主机将与从机通信，并定义传输时序。从机将与一个主机通信。

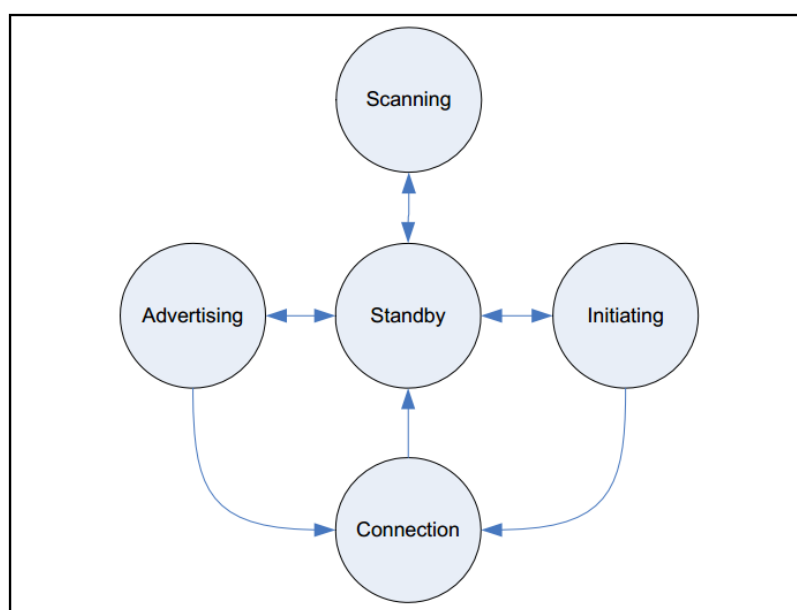


Figure 1.1: State diagram of the Link Layer state machine

1.1.1 状态与角色组合限制

链接层可以支持多个状态机，如果设计其不支持多个状态机，下面的状态和角色限制需要被使用：

- 连接状态的设备只能工作在主机或从机
- 工作在从机的设备只能有一个连接
- 工作在主机的设备可以有多个连接
- 从机不能由连接状态退到初始化状态
- 如果链路层已经工作在连接或初始化状态，从机不再接收能进入连接状态的广播帧。

所有其它状态组合可能被支持。链路层的实现不需要支持本标准定义的所有可能的状态组合。下表定义了允许和禁止的链路层状态机状态组合：

Multiple State Machine State and Roles		Advertising	Scanning	Initiating	Connection	
					Master Role	Slave Role
Advertising		Prohibited	Allowed	Allowed*	Allowed*	Allowed*
Scanning		Allowed	Prohibited	Allowed	Allowed	Allowed
Initiating		Allowed*	Allowed	Prohibited	Allowed	Prohibited
Connection	Master Role	Allowed*	Allowed	Allowed	Allowed	Prohibited
	Slave Role	Allowed*	Allowed	Prohibited	Prohibited	Prohibited

Table 1.1: Allowable and prohibited Link Layer state machine state combinations

* Only advertising packets that shall not result in the Link Layer entering Connection State in the Slave Role allowed.

1.2 位定义

在链接层定义的数据包或协议数据单元中，位顺序遵循小端格式。以下是规则：

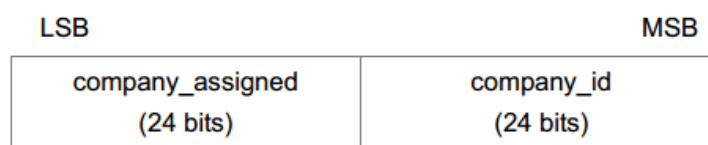
- 最低有效位对应 b0
- RF 发送时，先发最低位
- 印刷显示时，最低位安排在左边

1.3 设备地址

每个设备必须有一个设备地址，设备地址可以是一个公共的地址，也可以是一个随机地址，地址长度为 48 位。

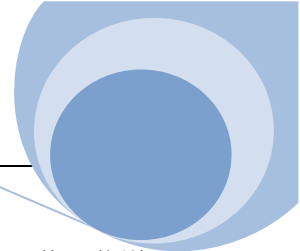
一个设备至少有一个公共地址或随机地址，或者可以两者都拥有。

公共地址定义如下：



随机地址定义如下：hash 在 [Vol. 3] Part C, Section 10.8.2.3. 有定义；random 在 [Vol. 3] Part C, Section 10.8.2.2. 中有定义。





1.4 物理信道

如 Part A 章节 2 描述的, 在 2.4G ISM 带宽中定义了 40 个 RF 信道。信道被分为两种 LE 物理信道: 广播信道和数据信道, 广播信道使用三个信道用于发现设备, 初始化连接, 和广播数据。多到 37 个信道用于两个已连接设备进行数据通信, 每一个信道都分配了一个唯一的信道索引。

两个设备如果要在同一信道上通信, 必须同时调谐在同一个信道。

当多个设备在同一范围内工作时肯定会产生冲突, 为了避免同一信道上的设备通信冲突, 设备在信道上传输时会使用访问地址 (Access address)。

一旦一个设备在某一个信道上在时序, 频率及访问地址同步即可称为已连接。

1.4.1 广播及数据信道索引

表 1.2 显示了 RF 信道与数据信道和广播信道索引的映射。也显示了每一个 RF 信道分配的类型。

RF Channel	RF Center Frequency	Channel Type	Data Channel Index	Advertising Channel Index
0	2402 MHz	Advertising channel		37
1	2404 MHz	Data channel	0	
2	2406 MHz	Data channel	1	
...	...	Data channels	...	
11	2424 MHz	Data channel	10	
12	2426 MHz	Advertising channel		38
13	2428 MHz	Data channel	11	
14	2430 MHz	Data channel	12	
...	...	Data channels	...	
38	2478 MHz	Data channel	36	
39	2480 MHz	Advertising channel		39

Table 1.2: Mapping of RF Channel to Data Channel Index and Advertising Channel Index

2、空中接口数据包

2.1 数据包格式

链路层只有一种数据包格式, 可用于广播信道及数据信道

LSB		MSB	
Preamble (1 octet)	Access Address (4 octets)	PDU (2 to 39 octets)	CRC (3 octets)

Figure 2.1: Link Layer packet format

最短包长为 10 个字节, 最长为 47 个字节。

2.1.1 引导码

引导码长 8 位, 在接收中用于频率同步, 数据速率同步和 AGC 调整。

在广播通道, 引导码为 10101010

在数据通道, 如果访问地址 (Access Address) 最低位为 1, 引导码为 01010101, 否则为 10101010

2.1.2 访问地址

访问地址长 32 位

所有的广播信道包访问地址为 10001110100010011011111011010110b (0x8E89BED6)

数据信道访问地址必须与其它任何已连接信道访问地址不一样，初始化者在初始化状态产生数据信道访问地址，并保证满足以下要求：

- 不能有连续超过 6 个以上的连续 0 或 1
- 不能与广播信道访问地址相同
- 不能与广播信道访问地址只有 1 位之差
- 数据信道访问地址不能四个字节都相同
- 不能有 24 位以上的变化
- 高 6 位必须有超过 2 位以上的变化

2.1.3 PDU

引导码与访问地址后紧跟的就是 PDU。

当数据包在一个广播信道上传输时，PDU 将成为广播信道 PDU，如章节 2.3 定义的。当数据包在一个数据信道上传输时，PDU 将成为数据信道 PDU，如章节 2.4 所定义的。

2.1.4 CRC

每一个链路层数据包最后是一个 24 位的 CRC。它通过 PDU 计算。CRC 算法在章节 3.1.1 中定义。

2.2 预留以备用

2.3 广播信道 PDU

广播信道 PDU 包括一个 16 位的数据头及可变长度的负载。

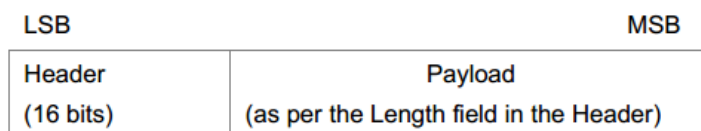


Figure 2.2: Advertising channel PDU

数据头定义如下：

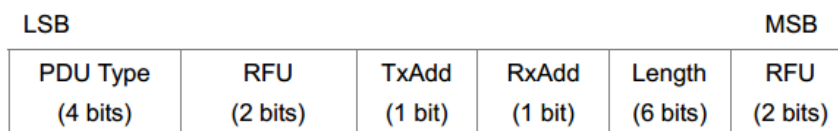


Figure 2.3: Advertising channel PDU Header

TxAdd: 0,表示发送方地址为公共地址;1,表示发送方地址为随机地址

RxAdd: 0,表示接收方地址为公共地址;1,表示接收方为随机地址

Length: 负载数据的字节长度，6-37.

PDU Type: 决定了 PDU 的类型

PDU Type $b_3b_2b_1b_0$	Packet Name
0000	ADV_IND
0001	ADV_DIRECT_IND
0010	ADV_NONCONN_IND
0011	SCAN_REQ
0100	SCAN_RSP
0101	CONNECT_REQ
0110	ADV_SCAN_IND
0111-1111	Reserved

Table 2.1: Advertising channel PDU Header's PDU Type field encoding

2.3.1 广播 PDU 类型

ADV_IND 可连接非定向广播事件

ADV_DIRECT_IND 可连接定向广播事件

ADV_NONCONN_IND 不可连接非定向广播事件

ADV_SCAN_IND 可扫描非定向广播事件

2.3.1.1 ADV_IND (无目的地广播)

ADV_IND 用于可连接非定向广播事件中，具有以下格式，

Payload	
AdvA (6 octets)	AdvData (0-31 octets)

Figure 2.4: ADV_IND PDU Payload

数据头中的 TxAdd 决定了在 AdvA 中的地址是公共地址 (TxAdd = 0) 还是随机地址 (TxAdd = 1)，AdvData 中可包含广播者的数据。

2.3.1.2 ADV_DIRECT_IND (指定地址广播)

ADV_DIRECT_IND 用于可连接定向广播事件中，具有以下格式：

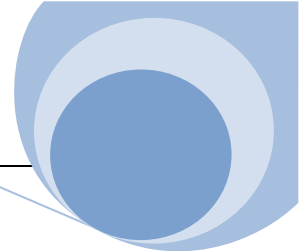
Payload	
AdvA (6 octets)	InitA (6 octets)

Figure 2.5: ADV_DIRECT_IND PDU Payload

数据头中的 TxAdd 决定了在 AdvA 中的地址是公共地址 (TxAdd = 0) 还是随机地址 (TxAdd = 1)，RxAdd 决定了在 InitA 中的地址是公共地址 (RxAdd = 0) 还是随机地址 (RxAdd = 1)。

2.3.1.3 ADV_NONCONN_IND

ADV_NONCONN_IND 用于不可连接非定向广播事件，具有以下格式：



Payload	
AdvA (6 octets)	AdvData (0-31 octets)

Figure 2.6: ADV_NONCONN_IND PDU Payload

数据头中的 TxAdd 决定了在 AdvA 中的地址是公共地址 (TxAdd = 0) 还是随机地址 (TxAdd = 1), AdvData 中可包含广播者的数据。

2.3.1.4 ADV_SCAN_IND= ADV_DISCOVER_IND

ADV_SCAN_IND 用于可扫描非定向广播事件, 具有以下格式:

Payload	
AdvA (6 octets)	AdvData (0-31 octets)

Figure 2.7: ADV_SCAN_IND PDU Payload

数据头中的 TxAdd 决定了在 AdvA 中的地址是公共地址 (TxAdd = 0) 还是随机地址 (TxAdd = 1), AdvData 中可包含广播者的数据。

2.3.2 扫描 PDUs

下面的广播信道 PDU 类型称为扫描 PDUs, 它们用于以下两种状态

- SCAN_REQ: 扫描请求, 由扫描状态的设备发出, 被广播状态的设备接收
- SCAN_RSP: 扫描应答, 由广播状态设备发出, 被扫描状态设备接收

2.3.2.1 SCAN_REQ

SCAN_REQ 具有以下格式

Payload	
ScanA (6 octets)	AdvA (6 octets)

Figure 2.8: SCAN_REQ PDU Payload

数据头中的 TxAdd 指示了扫描者的地址域 ScanA 是公共地址 (TxAdd = 0) 还是随机地址 (TxAdd = 1); RxAdd 指示了广播者的地址域 AdvA 是公共地址 (RxAdd = 0) 还是随机地址 (RxAdd = 1)。

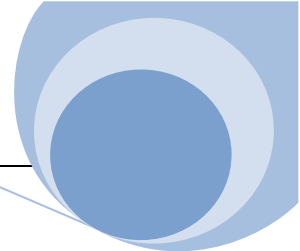
2.3.2.2 SCAN_RSP

SCAN_RSP 具有以下格式:

Payload	
AdvA (6 octets)	ScanRspData (0-31 octets)

Figure 2.9: SCAN_RSP PDU payload

数据头中的 TxAdd 指示了广播者的地址域 AdvA 是公共地址 (TxAdd = 0) 还是随机地址 (TxAdd = 1)。



2.3.3 初始化 PDUs

以下的广播信道 PDU 类型称为初始化 PDUs

● CONNECT_REQ 连接请求

连接请求 PDU 由初始化状态的设备发出，被广播状态的设备接收。

2.3.3.1 CONNECT_REQ

CONNECT_REQ 具有以下格式：

Payload		
InitA (6 octets)	AdvA (6 octets)	LLData (22 octets)

Figure 2.10: CONNECT_REQ PDU payload

数据头中的 TxAdd 指示了初始化者的地址域 InitA 是公共地址 (TxAdd = 0) 还是随机地址 (TxAdd = 1)。RxAdd 指示了广播者的地址域 AdvA 是公共地址 (RxAdd = 0) 还是随机地址 (RxAdd = 1)，以上 LLData 域有如下格式：

LLData									
AA (4 octets)	CRCInit (3 octets)	WinSize (1 octet)	WinOffset (2 octets)	Interval (2 octets)	Latency (2 octets)	Timeout (2 octets)	ChM (5 octets)	Hop (5 bits)	SCA (3 bits)

Figure 2.11: LLData field structure in CONNECT_REQ PDU's payload

LLData 由 10 个域组成：

AA:包括了连接者访问地址 (Access Address)

CRCInit:连接者使用的 CRC 运算初始值，是一个随机值

WinSize:决定传输窗口大小 (transmitWindowSize)， $\text{transmitWindowSize} = \text{WinSize} * 1.25 \text{ ms}$ 。

WinOffset:决定传输窗口偏置值 (transmitWindowOffsetvalue)， $\text{transmitWindowOffset} = \text{WinOffset} * 1.25 \text{ ms}$ 。

Interval:决定连接间隔 (connInterval)， $\text{connInterval} = \text{Interval} * 1.25 \text{ ms}$ 。

Latency:决定连接从机等待时间 (connSlaveLatencyvalue)， $\text{connSlaveLatency} = \text{Latency}$ 。

Timeout:决定连接从机超时值 (connSupervisionTimeoutvalue)， $\text{connSupervisionTimeout} = \text{Timeout} * 10 \text{ ms}$ 。

ChM: ChM 描述了使用和没有使用的数据信道 (Used and Unused data channels)，每一个数据信道由一个位表示，数据信道 0 由 bit0 表示，对应位为 0 表示信道没有被使用，如果为 1 表示信道已经被占用，位 37, 38, 39 预留使用，由该域查找数据信道索引时，应该注意排除广播信道。

Hop: 决定了下一跳增加的数据信道 (hopIncrement)，该值为一个随机值，范围为 5-16。

SCA:决定了主机睡眠系统时钟最大误差 (masterSCA)

SCA	masterSCA
0	251 ppm to 500 ppm
1	151 ppm to 250 ppm
2	101 ppm to 150 ppm
3	76 ppm to 100 ppm
4	51 ppm to 75 ppm
5	31 ppm to 50 ppm
6	21 ppm to 30 ppm
7	0 ppm to 20 ppm

Table 2.2: SCA field encoding

2.4 数据信道 PDU

数据信道的 PDU 包括了一个 16 位的数据头，一个可变长度的负载，还可能包括一个消息完整性校验域。

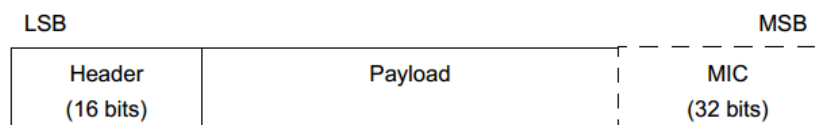


Figure 2.12: Data Channel PDU

数据头格式如下：

Header						
LLID (2 bits)	NESN (1 bit)	SN (1 bit)	MD (1 bit)	RFU (3 bits)	Length (5 bits)	RFU (3 bits)

Figure 2.13: Data channel PDU header

其中有效域定义如下：

Field name	Description
LLID	The LLID indicates whether the packet is an LL Data PDU or an LL Control PDU. 00b = Reserved 01b = LL Data PDU: Continuation fragment of an L2CAP message, or an Empty PDU. 10b = LL Data PDU: Start of an L2CAP message or a complete L2CAP message with no fragmentation. 11b = LL Control PDU
NESN	Next Expected Sequence Number
SN	Sequence Number
MD	More Data
Length	The Length field indicates the size, in octets, of the Payload and MIC, if included.

Table 2.3: Data channel PDU Header field

非加密连接及数据信道 PDU 中 0 字节负载长度的加密连接中不包括 MIC，数据信道 PDU 负载不为 0 的加密连接必须使用 MIC。

负载格式由 LLID 决定，如果 LLID = 01b 或 10b，数据信道 PDU 负载域是一个 LL DATA PDU，如果 LLID = 11b，数据信道 PDU 负载域是一个 LL CONTROL PDU，LLID = 00b 预留。

The NESN bit of the Header is defined in Section 4.5.9.

The SN bit of the Header is defined in Section 4.5.9.

The MD bit of the Header is defined in Section 4.5.6.

数据头的长度字节决定了负载和 MIC 长度，范围为 0-31，如果存在 MIC（4 字节），负载长度最大为 27。

2.4.1 LL DATA PDU

LL DATA PDU 是数据信道 PDU 中用于发送 L2CAP 数据，此时 LLID 需要被设置为 01b 或 10b。

如果 LLID = 01b，并且数据头的负载长度域为 0，则这是一个空 PDU。主机的连接可能发送一个空 PDU 给从机，以允许从机回复任意一个数据信道 PDU，包括空 PDU。

如果 LLID = 10b，则数据头的负载长度域不能设置为 0

2.4.2 LL CONTROL PDU

LL CONTROL PDU 在数据信道 PDU 中用于控制链路层的连接。

LL CONTROL PDU 具有以下数据格式：

Payload	
Opcode (1 octet)	CtrlData (0 – 22 octets)

Figure 2.14: LL control PDU payload

所有 LL CONTROL PDU 的数据头中的长度域不能为 0，所有 LL CONTROL PDU 都是固定长度，长度由 Opcode 决定。Opcode 决定了不同类型的 LL CONTROL PDU。

Opcode	Control PDU Name
0x00	LL_CONNECTION_UPDATE_REQ
0x01	LL_CHANNEL_MAP_REQ
0x02	LL_TERMINATE_IND
0x03	LL_ENC_REQ
0x04	LL_ENC_RSP
0x05	LL_START_ENC_REQ
0x06	LL_START_ENC_RSP
0x07	LL_UNKNOWN_RSP
0x08	LL_FEATURE_REQ
0x09	LL_FEATURE_RSP
0x0A	LL_PAUSE_ENC_REQ
0x0B	LL_PAUSE_ENC_RSP
0x0C	LL_VERSION_IND
0x0D	LL_REJECT_IND
0x0E-0xFF	Reserved for Future Use

Table 2.4: LL Control PDU Opcodes

CtrlData 由 LL CONTROL PDU 的类型决定。

如果接收到的 LL CONTROL PDU 是不被使用或不支持的，设备将返回 LL_UNKNOWN_RSP PDU，LL_UNKNOWN_RSP PDU 的未知类型域设置为没有使用或不支持的 Opcode。

如果接收到的 LL CONTROL PDU 的 Opcode 是无效的，如设置为预留值，或者 CtrlData 域是无效值，设备将返回 LL_UNKNOWN_RSP PDU，LL_UNKNOWN_RSP PDU 的未知类型域设置为没有使用或不支持的 Opcode。

2.4.2.1 LL_CONNECTION_UPDATE_REQ

该 PDU 具有以下格式：

CtrlData					
WinSize (1 octet)	WinOffset (2 octets)	Interval (2 octets)	Latency (2 octets)	Timeout (2 octets)	Instant (2 octets)

Figure 2.15: CtrlData field of the LL_CONNECTION_UPDATE_REQ PDU

WinSize: 决定了传输时间窗口时间 (transmitWindowSize), $\text{transmitWindowSize} = \text{WinSize} * 1.25$ ms.

WinOffset: 决定了传输时间窗口偏置值 (transmitWindowOffset value), $\text{transmitWindowOffset} = \text{WinOffset} * 1.25$ ms.

Latency: 决定连接从机等待时间 (connSlaveLatencyvalue), $\text{connSlaveLatency} = \text{Latency}$.

Timeout: 决定连接从机超时值 (connSupervisionTimeoutvalue), $\text{connSupervisionTimeout} = \text{Timeout} * 10$ ms.

Instant: 决定连接时间 (connInstant), 1-32767

2.4.2.2 LL_CHANNEL_MAP_REQ

该 PDU 具有以下数据格式:

CtrData	
ChM (5 octets)	Instant (2 octets)

Figure 2.16: CtrData field of the LL_CHANNEL_MAP_REQ PDU

ChM: ChM 描述了使用 and 没有使用的数据信道 (Used and Unused data channels), 每一个数据信道由一个位表示, 数据信道 0 由 bit0 表示, 对应位为 0 表示信道没有被使用, 如果为 1 表示信道已经被占用, 位 37, 38, 39 预留使用, 由该域查找数据信道索引时, 应该注意排除广播信道。

Instant: 决定连接时间 (connInstant), 1-32767

2.4.2.3 LL_TERMINATE_IND

该 PDU 具有以下数据格式:

CtrData
Error Code (1 octet)

Figure 2.17: CtrData field of the LL_TERMINATE_IND PDU

错误代码参考第二卷第四部分

2.4.2.4 LL_ENC_REQ

该 PDU 具有以下数据格式:

CtrData			
Rand (8 octets)	EDIV (2 octets)	SKDm (8 octets)	IVm (4 octets)

Figure 2.18: CtrData field of the LL_ENC_REQ PDU

Rand: 由主机提供的随机数, 并使用 EDIV (see [Vol. 3] Part H, Section 2.4.4).

EDIV: 加密域

SKDm: 包括了主机会议密钥认证。

IVm: 包括了主机初始化向量

2.4.2.5 LL_ENC_RSP

该 PDU 具有以下数据格式:

CtrData	
SKDs (8 octets)	IVs (4 octets)

Figure 2.19: CtrData field of the LL_ENC_RSP PDU

SKDs:从机会话密钥认证

IVs:从机初始化向量

2.4.2.6LL_START_ENC_REQ

无数据域

2.4.2.7LL_START_ENC_RSP

无数据域

2.4.2.8LL_UNKNOW_RSP

该 PDU 具有以下数据格式:

CtrData
UnknownType (1 octet)

Figure 2.20: CtrData field of the LL_UNKNOWN_RSP PDU

UnknownType: 其值为接收到的 LL_CONTROL_PDU 的 Opcode 值

2.4.2.9LL_FEATREQ_REQ

该 PDU 具有以下数据格式:

CtrData
FeatureSet (8 octets)

Figure 2.21: CtrData field of the LL_FEATURE_REQ PDU

FeatureSet: 描述了主机所支持的特性集合。

2.4.2.10LL_FEATURE_RSP

该 PDU 具有以下数据格式:

CtrData
FeatureSet (8 octets)

Figure 2.22: CtrData field of the LL_FEATURE_RSP PDU

FeatureSet: 描述了从机所支持的特性设置

2.4.2.11 LL_PAUSE_ENC_REQ

该 PDU 无数据域

2.4.2.12 LL_PAUSE_ENC_RSP

该 PDU 无数据域

2.4.2.13 LL_VERSION_IND

该 PDU 具有以下数据格式:

CtrData		
VersNr (1 octet)	Compld (2 octets)	SubVersNr (2 octets)

Figure 2.23: CtrData field of the LL_VERSION_IND PDU

VersNr: 描述了蓝牙控制器标准的版本

Compld: 描述了蓝牙控制器制造厂商认证

SubVersNr: 描述了蓝牙控制器的唯一序列号或蓝牙控制器版本号

2.4.2.14 LL_REJECT_IND

该 PDU 具有以下数据格式:

CtrData
Error Code (1 octet)

Figure 2.24: CtrData field of the LL_REJECT_IND

Error Code: 错误代码, 参考[Vol 2] Part D。

3、位流处理

以下处理流程必须在 PDU 上执行

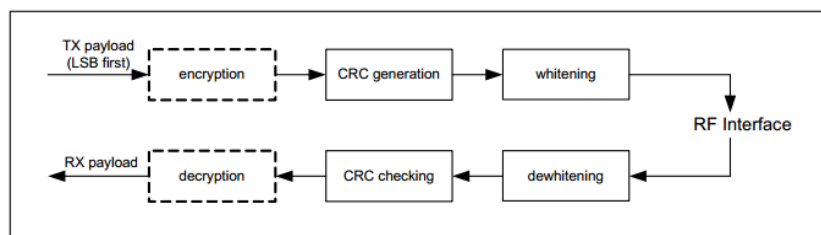


Figure 3.1: Payload bit processes

3.1 差错检测

当一个包接收到时, 首先需要检查访问地址 (ACCESS ADDRESS), 如果访问地址不正确, 则该包被丢弃, 否则该包将被接收。如果包 CRC 不正确, 则该包被丢弃, 否则该包被认为正确, 只有有效的数据包才能被处理。一个 CRC 错误的数据包可能引起一个继续连接事件, 参考 4.5.1。

3.1.1 CRC 产生

所有链路层的数据包都需要计算 PDU 的 CRC。如果 PDU 是加密的, 则先加密后进行 CRC 计算。

CRC 多项式是一个 24 位的 CRC, 并且 PDU 从最低位起所有位都需要参与计算。多项式为:

$$x^{24} + x^{10} + x^9 + x^6 + x^4 + x^3 + x + 1$$

对于所有数据信道的 PDU, CONNECT_REQ PDU 中链路层的通信与连接中所有的移位寄存器被设置为 CRC 初始值, 每一个广播信道的 PDU, 移位寄存器被设置为 0x555555。

CRC 传输时从最高位 bit23 开始传输。

下图描述了如果产生一个 CRC:

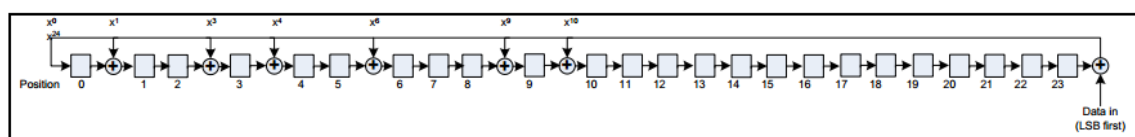


Figure 3.2: The LFSR circuit generating the CRC

3.2 数据白化

数据白化是为了避免数据流中长序列的 0 或 1，链路层数据包中的 PDU 及 CRC 都需要数据白化，并且数据白化的执行在 CRC 运算之后。接收方解数据白化执行在 CRC 运算之前。

数据白化与解白化使用相同的方式，使用了一个 7 位的线性反馈移位寄存器，其多项式为

$$x^7 + x^4 + 1$$

在数据白化与解白化之前，移位寄存器需要被初始化，初始化值是具有以下格式，并且由数据包传输的信道索引推导而来：

- 位 0 为 1
- 位 1-6 设置为收发时所使用的信道索引，位 1 为最高有效位，位 6 为最低有效位。

如信道索引为 23=0X17=1110 101B。

下图显示了数据白化的产生：

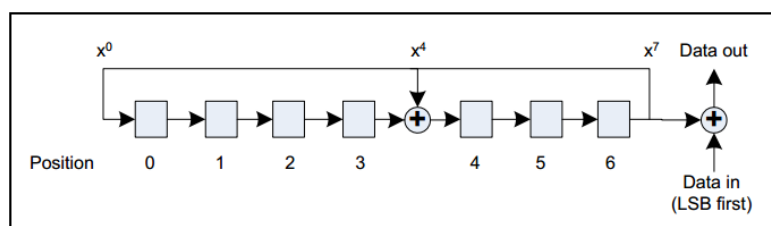


Figure 3.3: The LFSR circuit to generate data whitening

4、空中接口协议

4.1 帧间间隔

第一个包最后一个发送位与第二个包第一个发送位之间的时间，即 T_IFS，固定为 150uS。

4.2 时序要求

链路层要求使用两种时钟精度之一，在连接事件及广播事件中使用活跃时钟精度，否则使用睡眠时钟精度。

4.2.1 时钟精度

连接事件中的包传输平均时间使用激活时钟精度，误差应小于等于+50ppm。总共的误差时间不能超过 2us，那么包与包之间的时间间隔为 150+-2us。

4.2.2 睡眠时钟精度

广播事件及连接事件的时序使用睡眠时钟精度，误差偏移小于等于+500ppm。

引导点的瞬间在平均时序中不能超过 16us。

注意：在一个 1s 总误差为+1000ppm 的睡眠精度的连接间隔将扩展引导点两边的时间窗口，扩展的时间为 1ms+16us，以上假设从机在全部的连接间隔中都使用睡眠时钟。

The instantaneous timing of the anchor point (see Section 4.5.7) shall not deviate more than 16 μs from the average timing.

Note: This means that a 1 sec connection interval with a total ±1000ppm sleep clock accuracy will give a window widening either side of the anchor point of 1ms plus 16us, assuming that the slave controller was using its sleep clock for almost the complete connection interval.

4.3 链路层设备过滤

链路层可以通过设备地址对设备进行过滤，链路层设备过滤可用于减少响应的从机数量。

链路层需要支持链路层设备过滤，除非它只支持不可连接的广播

广播状态，扫描状态，初始化状态的过滤策略是相互独立的，在不同地状态有其对应的过滤策略，如果广播过滤策略没有使用，则其它过滤策略没必要存在。

链路层用于设备过滤的设备集称为怀特表。

怀特表包括了用于设备过滤的设备集，怀特表内包括了设备地址及地址类型（公共的还是随机的），支持链路层设备过滤的链路至少支持一个怀特表记录。复位后，怀特表为空。

怀特表由主机配置，在链路层用于过滤广播者，扫描者或初始化者。这样主机可以不激活请求就可以在链路层发出请求。

所有的设备过滤策略都使用相同的怀特表。

4.3.2 广播过滤策略

广播过滤策略定义了广播者如何处理扫描以及或连接请求。

当链路层用于可连接定向广播时，广播过滤策略将被忽略，否则链路层将使用以下由主机配置的广播过滤策略模式。

- 链路层只处理怀特表中的设备的扫描及连接请求。
- 链路层处理所有设备的扫描及连接请求，上电复位的默认状态，此时怀特表没有用。
- 链路层处理所有设备的扫描请求，但只处理怀特表中设备的连接请求。
- 链接层处理所有设备的连接请求，但只处理怀特表中设备的扫描请求。

同一时刻只能使用一种过滤策略。

4.3.3 扫描过滤策略

扫描过滤策略定义了扫描者如何处理广播数据包。链路层使用以下由主机配置的扫描过滤策略：

- 链路层只处理怀特表中设备的广播数据包
- 链路层将处理所有设备的广播数据包，上电复位功能。

对于扫描者过滤策略，不包含扫描者地址的可连接定向广播数据包将被忽略。

同一时刻只能使用一种过滤策略。

4.3.4 初始化者过滤策略

初始化者过滤策略定义了初始化者如何处理广播数据包，链路层使用以下由主机配置的初始化者过滤策略：

- 链路层将处理怀特表中所有设备的可连接广播数据包
- 链路层将忽略怀特表，并只处理由主机确定地址的设备的广播数据包。

链路层接将忽略不包括在怀特表内的设备或主机指定地址的设备发来的可连接定向广播数据包。

同一时刻只能使用一种过滤策略。

4.4 非连接状态

4.4.1 待机状态

待机状态是链路层的默认状态，在待机状态，链路层没有收发数据，从待机状态可以进入广播状态，扫描状态及初始化状态。

4.4.2 广播状态

链路层由主机指定进入广播状态，当进入广播状态，链路层将在广播事件发送广播 PDUs。每一个广播事件可以由一个或多个广播 PDU 组成。当一个广播 PDU 在每一个广播信道都发送后广播事件将被关闭，广播者也可以过早关闭广播事件，以实现其它功能。

两个连续广播事件的时间由 Section 4.4.2.2 定义。

广播事件可以是以下类型之一：

- 可连接非定向事件
- 可连接定向事件
- 不可连接非定向事件
- 扫描非定向事件

对于不同的事件类型使用对应的广播信道 PDU。

每一个广播事件的第一个 PDU 将由可用广播信道索引值最低的发送。

广播事件的类型决定了允许的应答 PDUs，下表确定了每一个广播事件可允许的应答。

Advertising Event Type	PDU used in this advertising event type	Allowable response PDUs for advertising event	
		SCAN_REQ	CONNECT_REQ
Connectable Undirected Event	ADV_IND	YES	YES
Connectable Directed Event	ADV_DIRECT_IND	NO	YES*
Non-connectable Undirected Event	ADV_NONCONN_IND	NO	NO
Scannable Undirected Event	ADV_SCAN_IND	YES	NO

Table 4.1: Advertising event types, PDUs used and allowable response PDUs

* Only the correctly addressed initiator may respond.

如果广播者从广播事件中接收到一个不允许的 PDU，该 PDU 将被忽略，如果没有收到 PDU 或 PDU 被忽略，广播者将在下一个可用广播信道发送一个广播 PDU 或关闭广播事件。

4.4.2.1 广播信道选择

广播事件使用三个预定义的广播信道，广播信道索引可使用或不使用。

链路层将使用主机确定的广播信道索引值，一旦进入广播状态，已使用的广播信道索引值将起作用。

4.4.2.2 广播间隔

对于所有非定向广播事件，两个连续的广播事件间隔时间由下式确定：

$$T_{advEvent} = advInterval + advDelay$$

advInterval 在 20ms-10.24s 之间，并且是 0.625ms 的整数倍。如果广播事件是扫描非定向事件类型或不可连接非定向事件类型，advInterval 将不小于 100ms，如果广播事件是可连接非定向事件类型，advInterval 可以是 20ms 或更大。

advDelay 是一个在 0-10ms 之间的预设随机值，由链路层在每一个广播事件中产生。

如下图，使用 advDelay 后广播事件在时序上将被打乱：

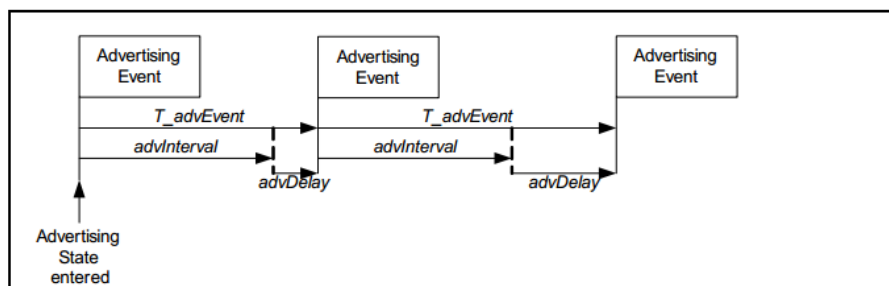


Figure 4.1: Advertising events perturbed in time using advDelay

4.4.2.3 可连接非定向事件类型

当使用可连接非定向事件类型时，广播（ADV_IND PDU）将被发送。

可连接非定向事件类型允许扫描者或初始化者应答扫描请求或连接请求。扫描者发送扫描请求以了解广播者更多的信息。初始化者可以发送一个连接请求（CONNECT_REQ PDU）以使链路层进入连接状态。

链路层将监听相同的广播信道扫描者或初始化者请求。

如果广播者接收到来自通过广播过滤策略的扫描者的包含广播者地址的 SCAN_REQ PDU，广播者将在相同的信道回复一个 SCAN_RSP PDU。SCAN_RSP PDU 发送后，或者广播过滤策略阻止了 SCAN_REQ PDU，广播者将移动到下一个广播信道发送 ADV_IND PDU 或关闭广播事件。

如果广播者接收到来自通过广播过滤策略的初始化者的包含广播者地址的 CONNECT_REQ PDU, 广播者将退出广播状态, 并进入连接状态, 进入从机角色。如果广播过滤策略阻止了 CONNECT_REQ PDU, 广播者将移动到下一个广播信道发送 ADV_IND PDU 或关闭广播事件。

各信道两个 ADV_IND PDU 的起始头的间隔应小于等于 10ms, 在广播间隔内, 将关闭广播状态。

下图显示了在所有的广播信道没有接收到 SCAN_REQ PDU 或 CONNECT_REQ PDU 时广播事件的使用。

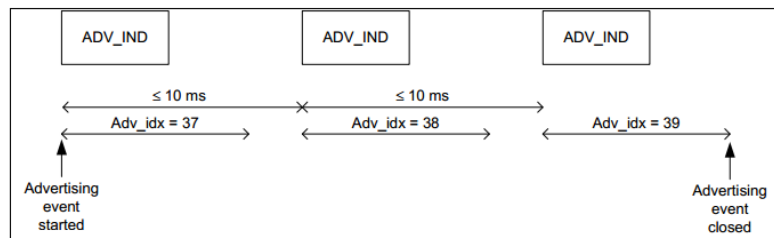


Figure 4.2: Connectable undirected advertising event with only advertising PDUs

下面两图说明了在某一个广播信道接收到 SCAN_REQ PDU 并且发出一个 SCAN_RSP PDU 时广播事件的使用:

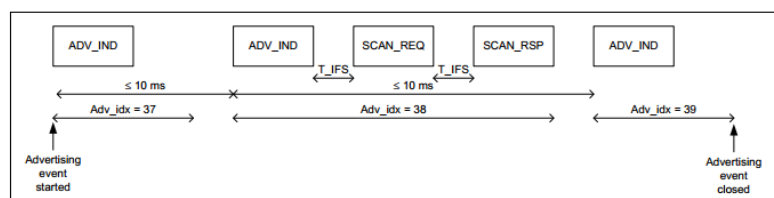


Figure 4.3: Connectable undirected advertising event with SCAN_REQ and SCAN_RSP PDUs in the middle of an advertising event

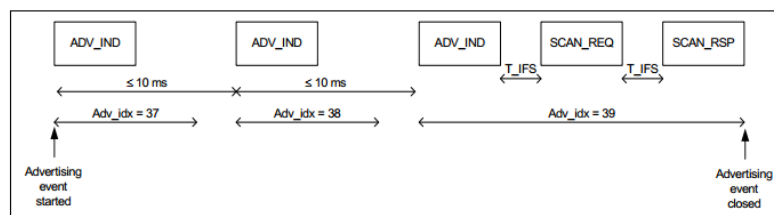
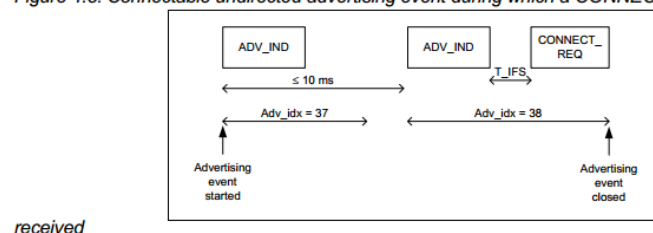


Figure 4.4: Connectable undirected advertising event with SCAN_REQ and SCAN_RSP packets at the end of an advertising event

下图描述了在第二个广播信道接收到一个 CONNECT_REQ PDU.

Figure 4.5: Connectable undirected advertising event during which a CONNECT_REQ PDU is



4.4.2.4 可连接定向事件类型

当可连接定向事件类型被使用时, 链路层将发送定向广播数据 (ADV_DIRECT_IND PDU)。

可连接定向事件类型允许初始化者使用连接请求做为应答, 初始化者将发送一个连接请求 (CONNECT_REQ PDU) 以请求链路层进入连接状态。

ADV_DIRECT_IND PDU 包含了初始化者和广播者的设备地址, 只有已编址的初始化者可以通过 CONNECT_REQ PDU 与广播者初始化链路层连接。

当广播者发送每一个 ADV_DIRECT_IN PDU 之后, 广播者将在同信道监听 CONNECT_REQ PDU。任何接收到的 SCAN_REQ PDUs 将被忽略。

如果广播者接收到一个包含其地址的 `CONNECT_REQ` PDU，并且 `ADV_DIRECT_IND` PDU 包含初始化者的地址，链路层将退出广播状态，并进入连接状态，做为从机角色。否则广播者将进入下一广播信道发送 `ADV_DIRECT_IND` PDU 或关闭广播事件。

相同信道上两个 `ADV_DIRECT_IND` PDU 起始头之间时间间隔必须小于等于 3.75ms。

进入广播状态后，链路层必须在 1.28s 内退出广播状态。

下图显示五个 `ADV_DIRECT_IND` PDUs，两个广播事件序列中，没有收到 `CONNECT_REQ` PDUs：

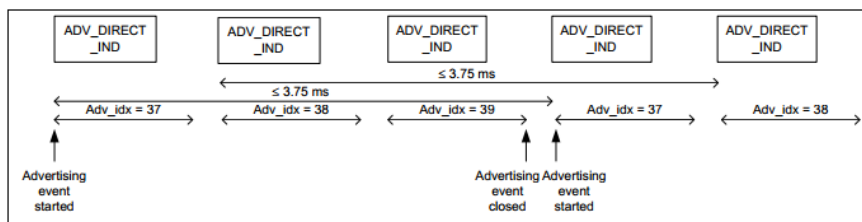


Figure 4.6: Connectable directed advertising event type with only advertising PDUs

注意：可连接定向广播用于快速连接，比如原来连接过的设备再次连接。

4.4.2.5 可扫描非定向事件类型

当扫描非定向事件类型被使用，扫描非定向数据（`ADV_SCAN_IND` PDUs）将被发送。

扫描非定向事件类型允许扫描者以扫描请求（`SCAN_REQ` PDU）应答广播者，请求广播者更多的信息。

链路层应该监听相同广播信道上扫描者的请求。

如果广播者接收到来自通过广播过滤策略的扫描者的包含广播者设备地址的 `SCAN_REQ` PDU，广播者将在相同信道回复 `SCAN_RSP` PDU。`SCAN_RSP` PDU 发送后或者广播过滤策略不再处理 `SCAN_REQ` PDU，广播者将进入下一个广播信道发送 `ADV_SCAN_IND` PDU，或关闭广播事件。

在两个 `ADV_SCAN_IND` PDU 起始头之间的间隔应该小于等于 10ms，广播事件在该间隔时间内应该被关闭。

广播事件中没能收到 `SCAN_REQ` PDU 的时序如下：

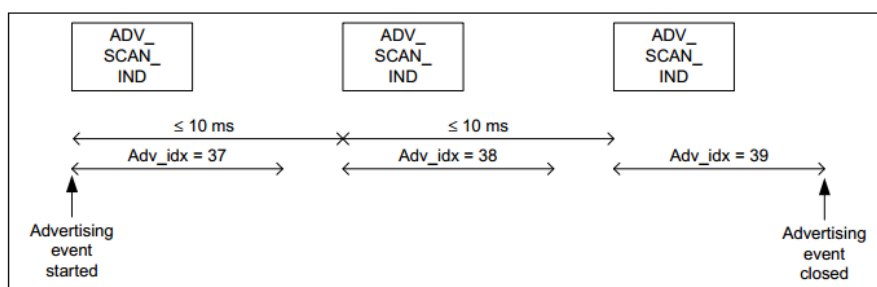


Figure 4.7: Scannable undirected advertising event with only advertising PDUs

下两图中描述了某广播信道中接收到一个 `SCAN_REQ` PDU，并发送一个 `SCAN_RSP` PDU：

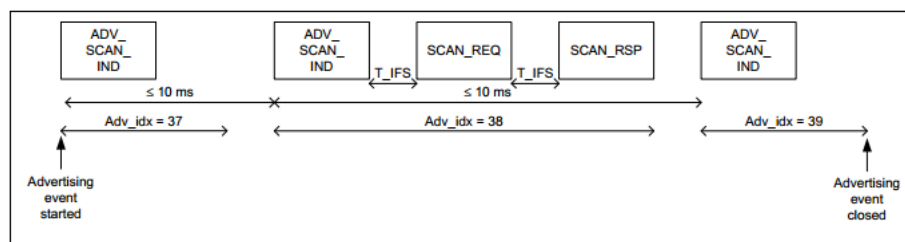


Figure 4.8: Scannable undirected advertising event with `SCAN_REQ` and `SCAN_RSP` PDUs in the middle of an advertising event

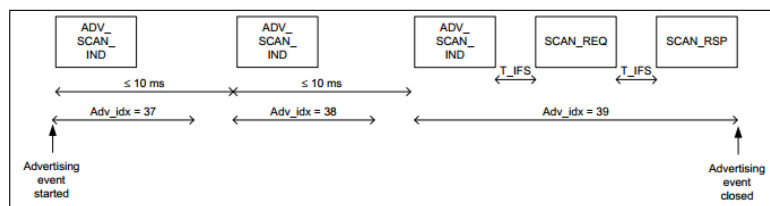


Figure 4.9: Scannable undirected advertising event with SCAN_REQ and SCAN_RSP PDUs at the end of an advertising event

4.4.2.6 不可连接非定向事件类型

当不可连接非定向事件类型使用时，不可连接广播数据（ADV_NONCONN_IND PDU）被发送。

不可连接非定向事件类型允许扫描者从广播者接收 ADV_NONCONN_IND PDU。

当广播者发送 ADV_NONCONN_IND PDU 后，将进入到下一个广播信道或关闭广播事件，因为链路层不监听，因此广播者不能从扫描者及初始化者接收任何请求。

两个 ADV_NONCONN_IND PDUs 头部之间的间隔小于等于 10ms，广播事件应该在广播间隔中被关闭。

在所有可使用的广播信道上，不可连接广播事件如下：

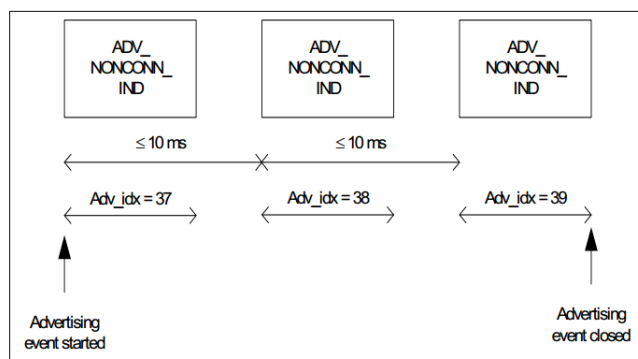


Figure 4.10: Non-connectable undirected advertising event

4.4.3 扫描状态

链路层将被主机指定进入扫描状态，在扫描状态，链路层将在广播信道上监听。有两种由主机决定的扫描类型：主动模式和被动模式。

针对扫描没有严格的时序或广播信道索引选择。在扫描状态，链路层将在扫描窗口（scanWindow）持续监听，扫描间隔（scanInterval）定义为连接两个扫描窗口（scanWindow）之间的间隔。链路层在由主机指定的每一个扫描间隔监听完整的扫描窗口（scanWindow），除非有时序冲突。在每一个扫描窗口，链路层应该扫描不同的广播信道，链路层应该使用所有的广播信道。

扫描窗口（scanWindow）和扫描间隔（scanInterval）参数应该小于等于 10.24s，扫描窗口（scanWindow）应该小于等于扫描间隔（scanInterval），如果扫描窗口（scanWindow）等于扫描间隔（scanInterval），链路层将连续扫描。

在扫描状态，接收到一个广播 PDU 时，扫描过滤策略将被执行。

从广播者接收到每一个包含链路层设备地址的非重复的 ADV_DIRECT_IND PDU，链路层将发送一个广播报告给主机。对于每一个来自广播者的非重复的 ADV_IND, ADV_SCAN_IND, ADV_NONCONN_IND, 或 SCAN_RSP PDU 链路层应该发送一个广播报告给主机。广播报告应该至少包含广播者设备地址和广播数据或扫描应答数据。重复的广播报告不需要发送给主机。当链路层保持在扫描状态，一个重复广播报告是一个包含相同设备地址的广播报告。广播数据可以改变，当确定为重复广播报告时，广播数据或扫描应答数据被认为没有意义。

4.4.3.1 被动扫描

当在被动扫描时，链路层只接收数据包，将不发送任何数据。

4.4.3.2 主动扫描

在主动扫描下，链路层将监听广播 PDU，并根据广播 PDU 类型可能需要请求广播者发送一些额外信息。

链路层将不发送 SCAN_REQ PDU 给接收到 ADV_DIRECT_IND PDU 或 ADV_NONCONN_IND PDU 的广播者。链路层进入扫描状态后将发送至少一个 SCAN_REQ PDU 给接收到 ADV_IND or ADV_SCAN_IND PDUs 的广播者，链路层将进一步发送 SCAN_REQ PDUs 给已经接收到 ADV_IND 或 ADV_SCAN_IND PDUs 的广播者。链路层将交错多个广播者的 SCAN_RSP PDUs。

扫描者将执行一个回退以减少多个扫描者发送 SCAN_REQ PDUs。

当冲突发生在 SCAN_RSP PDUs 时，回退过程使用两个参数 backoffCount 和 upperLimit 限制 SCAN_REQ PDUs 的发送数量。

一旦进入扫描状态 backoffCount 和 upperLimit 需要被设置为 1。

当接收到通过扫描过滤策略的 ADV_IND PDU 或 ADV_SCAN_IND PDU，且 SCAN_REQ PDU 被发送，backoffCount 应该减 1 直到 0。SCAN_REQ PDU 只能在 backoffCount 为 0 时才能被发送。

发送 SCAN_REQ PDU 后，链路层将监听来自广播者的 SCAN_RSP PDU。如果没有接收到 SCAN_RSP PDU 将被认为失败，否则成功。在每两个连续的失败，upperLimit 应该翻倍，直到达到 256，在每两个连续的成功，upperLimit 应该折半，直接达到 1。在接收 SCAN_RSP PDU 成功或失败后，链路层将设置 backoffCount 为一个在 1 到 upperLimit 之间的随机值。

广播信道中接收到 SCAN_REQ PDU 并发送 SCAN_RSP PDU 时序如下两图

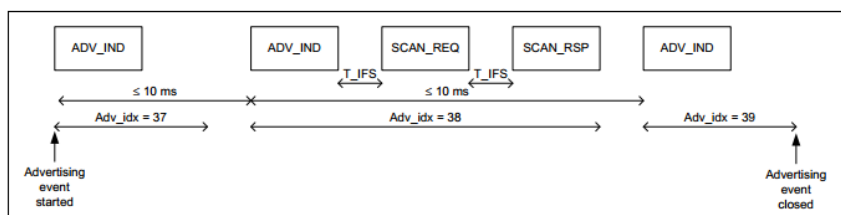


Figure 4.3: Connectable undirected advertising event with SCAN_REQ and SCAN_RSP PDUs in the middle of an advertising event

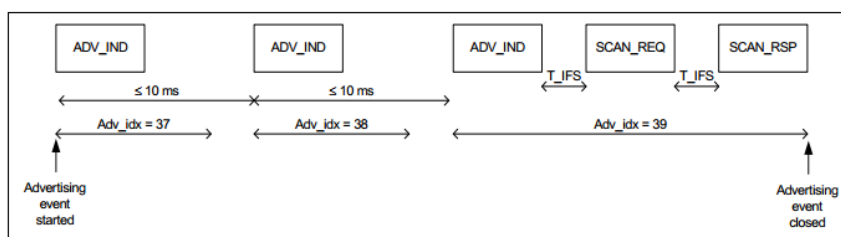


Figure 4.4: Connectable undirected advertising event with SCAN_REQ and SCAN_RSP packets at the end of an advertising event

4.4.4 初始化状态

链路层将在主机的指定下进入初始化状态。在初始化状态，链路层将监听广播信道。

对于初始化者没有严格的时序及广播信道选择规则。

在初始化状态，链路层将在扫描窗口（scanWindow）持续监听，扫描间隔（scanInterval）定义为连接两个扫描窗口（scanWindow）之间的间隔。链路层在由主机指定的每一个扫描间隔监听完整的扫描窗口（scanWindow），除非有时序冲突。在每一个扫描窗口，链路层应该扫描不同的广播信道，链路层应该使用所有的广播信道。

扫描窗口（scanWindow）和扫描间隔（scanInterval）参数应该小于等于 10.24s，扫描窗口（scanWindow）应该小于等于扫描间隔（scanInterval），如果扫描窗口（scanWindow）等于扫描间隔（scanInterval），链路层将连续扫描。

如果接收到一个通过初始化过滤策略的 ADV_IND PDU, 初始化者将发送一个 CONNECT_REQ PDU 给广播者。如果接收到一个带有链路层设备地址的并通过初始化过滤策略的 ADV_DIRECT_IND PDU, 初始化者将发送一个 CONNECT_REQ PDU 给广播者, 否则将忽略。

发送 CONNECT_REQ PDU 后, 链路层将退出初始化状态, 并进入连接状态, 做为主机角色。

4.5 连接状态

当初始化者发送一个 CONNECT_REQ PDU 后, 或者一个广播者接收到 CONNECT_REQ PDU, 将进入连接状态。

进入连接状态后即认为连接已经创建, 但该连接并不被认为是已连接的。当从节点接收到数据信道的数据包时, 连接才被认为已连接。

创建连接与已连接的差别是链接层连接管理超时值是否被使用。

当两个设备实现连接, 两个设备参与不同的角色, 主角色称为主机, 从角色称为从机, 主机控制连接事件的时序, 连接事件是主机和从机同步的要点。

4.5.1 连接事件

连接状态下的链路层只能在连接事件中传输数据信道数据包, 每一个连接事件下, 主机和从机必须确定数据信道索引, 在连接事件中, 所有的数据包使用相同的数据信道索引。每个连接事件至少包含主机发送的一个数据包。

在连接事件中, 主机和从机交替发送和接收数据包, 当设备双方连续发送数据时连接事件被认为是开放的。从机收到主机的数据包时, 无论 CRC 是否正确, 均需要返回一个数据包, 但连续多个 CRC 错误除外。主机收到从机的数据包时, 无论 CRC 是否正确, 也需要返回一个数据包。数据包中 CRC 错误也认为数据头的长度域是正确的, 如果主机没有收到从机的数据包, 主机则关闭连接事件。

连接事件可以被主机或从机关闭。

连接事件的时序由两个参数决定连接事件间隔 (connInterval) 和从机等待延时 (connSlaveLatency)。

连接事件的起始称为定位点 (anchor point), 在定位点主机将开始传输一个数据信道 PDU 给从机, 连接事件的起始将由连接事件间隔 (connInterval) 等间隔区分并且不会重叠。在下一个连接事件定位点之前, 主机应该至少保证在 T_IFS 时连接事件是关闭的。在定位点, 从机监听由它的主机发送来的数据包。

connInterval 应该在 7.5ms-4s, 并且是 1.25ms 的整数倍, connInterval 由初始化者链路层设置, 其值为主机发送的 CONNECT_REQ PDU 内定义的范围。

从机延时允许从机减少连接事件的数量, connSlaveLatency 参数决定了从机不需要监听来自主机的连续连接事件。connSlaveLatency 值不应该引起管理超时 (Supervision Timeout)。

connSlaveLatency 是一个 0 到 $((\text{connSupervisionTimeout} / \text{connInterval}) - 1)$ 的整数, 同时 connSlaveLatency 要小于 500。当 connSlaveLatency 为 0 时, 从机将监听每个数据包的定位点。当使用从机延时 (slave latency) 后, 从机没有收到主机的数据包, 它将继续监听每一个数据包的定位点, 并且不使用从机延时, 直到接收到主机的数据包。

主从机在每一个连接都有一个 16 位的连接事件计数器 (connEventCounter), 连接者主机发送第一个连接事件后, 连接事件计数器将被设置为 0。主机每发送一个新的连接事件, 该计数值将加 1, connEventCounter 值为 0xffff-0x0000, 该计数器用于同步链路层控制过程。

从机应该对于所有的连接事件都要加 1, 即使从机由于延时没有监听主机。

4.5.2 管理超时

连接可能因为多种原因断开, 比如设备超出连接范围, 被严重干扰或掉电。由于这些原因的发生没有任何的前兆, 主从机监视连接的状态是非常重要的。

为了能够检测到连接丢失, 主从机需要使用链路层连接管理定时器, T_LLconnSupervision, 该定时器由接收到的数据包复位。

如果在连接已建立之前链路管理定时器达到 $6 * \text{connInterval}$, 连接将被认为丢失, 这样可以快速的中断失败的已连接。

连接管理超时（connSupervisionTimeout）定义了连接丢失前两个接收数据包之间最大的时间。connSupervisionTimeout 应该在 10ms-32s 之间，且为 10ms 的整数倍，并大于 $(1 + \text{connSlaveLatency}) * \text{connInterval}$ 。

在任何时候，已连接设备中的管理定时器达到 `connSupervisionTimeout`，连接被被认为已丢失。

如果连接丢失，链路层将不再发送任何数据包，链路层退出连接状态，并进入待机状态。应该通知主机连接已丢失。

4.5.3 连接事件传输窗口

为了允许主机有效的在多连接者或其它与之有关的活跃者之间调度连接事件，主机在他选择的时间点可以灵活地安排第一个连接事件定位点。**CONNECT_REQ** PDU 内的参数决定了主机在连接状态什么时候可以发送第一个数据包，以设置引导点，什么时候从机必须监听。

CONNECT_REQ PDU 包含了三个参数以决定传输窗口，传输窗口起始于 CONNECT_REQ PDU 结束后 transmitWindowOffset+ 1.25 ms，并且 transmitWindowSize 定义了传输窗口的尺寸，connInterval 用于计算传输窗口的最大偏置和尺寸。transmitWindowOffset 和 transmitWindowSize 由链路层决定。

transmitWindowOffset 应该在 0-connInterval 之间，并且是 1.25ms 的整数倍，transmitWindowSize 应该在 1.25ms 到 $\min(10\text{ms}, (\text{connInterval} - 1.25\text{ms}))$ 之间，并且是 1.25ms 的整数倍。

所以在广播信道内 `CONNECT_REQ` PDU 传输结束之后第一个数据包的起始不能早于 $1.25\text{ ms} + \text{transmitWindowOffset}$ ，且不能晚于 $1.25\text{ ms} + \text{transmitWindowOffset} + \text{transmitWindowSize}$ 。

4.5.4 连接设置——主角色

初始化者发送 CONNECT_REQ PDU 后，链路层进入连接状态做为主角色。主机将复位链路层连接管理时间 T_LLconnSupervision，链路层将通知主机连接已经被创建。第一个连接事件应该使用 Section 1.4.1 定义的数据信道。

主机将在 Section 4.5.3 定义的传输窗口中发送第一个数据包，主机第一个数据包超过传输窗口时间是被禁止的。

处于连接状态的主机发送的第一个数据包决定了第一个连接事件的引导点及该连接下所有连接事件的时序。

第二个连接事件引导点需要在第一个连接事件引导点的连接间隔（connInterval）之后，所有的正常连接事件传输规则在 Section 4.5.1 时有定义。

以下是两个具有 LL connection 设置过程的主机时序图:

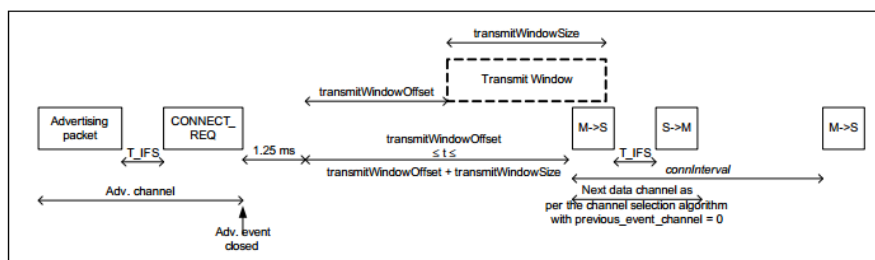


Figure 4.11: Master's view on LL connection setup with a non-zero transmitWindowOffset

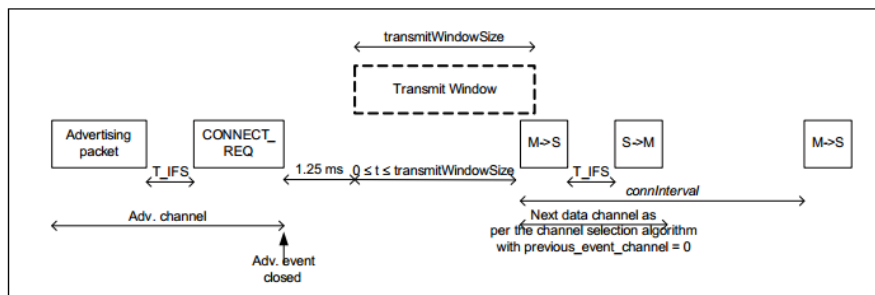


Figure 4.12: Master's view on LL connection setup with `transmitWindowOffset` set to zero

4.5.5 连接设置——从角色

当广播者接收到一个 CONNECT_REQ PDU 后，链路层将进入连接状态的从模式。从机将复位链路层连接管理定时器 T_LLconnSupervision，链路层将通知主机连接已经创建，第一个连接事件将使用 Section 1.4.1 定义的数据信道。

从机将在由 Section 4.5.3 定义的传输窗口中监听第一个数据包，主机的第一个数据包不允许超出传输窗口，从机必须对此进行考量。

第一个数据包接收后，无论 CRC 是否正确，从机的连接状态决定了第一个连接事件的引导点及该连接下所有连接事件的时序。

如果有一个传输窗口中没有接收到一个数据包，从机将尝试在下一个传输窗口中接收数据包，第二个传输窗口开始于第一个传输窗口的连接间隔（connInterval）之后，与第一个传输窗口大小一样。传输信道将跳转到下一个 Section 1.4.1 定义的数据信道。连接事件计数器（connEventCount）将自加 1。

下图显示了从机第一次没有接收到数据包，并在第二次接收到数据包的时序图：

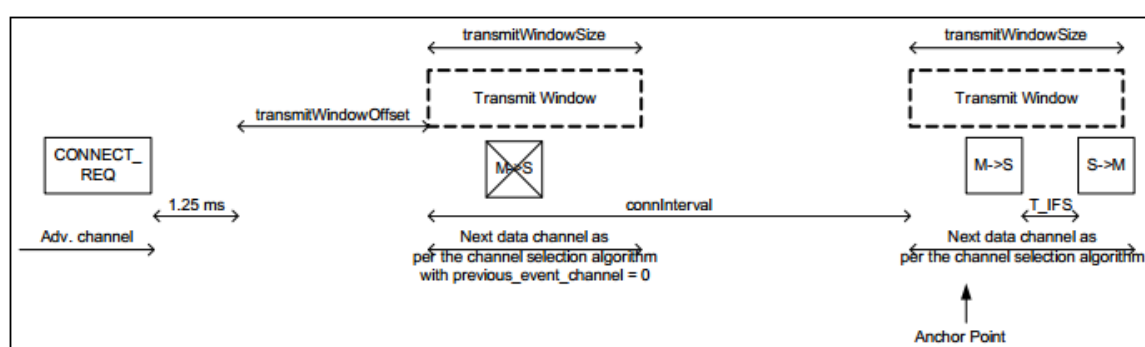


Figure 4.13: Slave closing LL connection setup in the second LL connection event

从机将在任何一个连接事件中被激活，直到通过 Section 4.5.1 定义的相关主机数据包将 NESN 设置为 1。

4.5.6 关闭连接事件

数据信道 PDU 中头部的 MD 位指示了是否有更多的数据将被发送，无论设备是否将 MD 置位，从机发送的数据包将关闭连接事件。如果其中一个设备将 MD 置位，主机将通过发送另一个数据包以维持连接，从机将在发送数据包后继续监听，如果主机没有接收到从机的数据包，主机将关闭连接事件。如果从机没有接收到主机的数据包，从机将关闭连接事件。

连续两个数据包的 CRC 错误将关闭连接事件。

MD 位使用如下表

		Master	
		MD = 0	MD = 1
Slave	MD = 0	Master shall not send another packet, closing the connection event. Slave does not need to listen after sending its packet.	Master may continue the connection event. Slave should listen after sending its packet.
	MD = 1	Master may continue the connection event. Slave should listen after sending its packet.	Master may continue the connection event. Slave should listen after sending its packet.

Table 4.2: MD bit usage for closing connection events

4.5.7 窗口扩展

由于睡眠时钟的精度问题，从机在主机引导点数确时序中有不确定性。因此从机需要在第一个所监听到的连接事件中进行重同步（re-synchronize），如果从机接收到一个 CRC 错误的数据包，从机将更新其引导点。

当主机发送第一个带有时钟抖动的数据包(slaveExpectedAnchorPoint)，或连接设置，或更新连接窗口设置时，从机将计算其时间。从机将使用 CONNECT_REQ PDU 中的主机睡眠时钟精度(masterSCA)，从机自己的时钟精度(slaveSCA)，以及他接收到的最后一个连接事件的数据包的引导点以计算它需要接收的时间。

监听时间的增加称为窗口扩展，假设时钟精度以 ppm 来定，窗口扩展按下式计算：

$$\text{windowWidening} = ((\text{masterSCA} + \text{slaveSCA}) / 1000000) * \text{timeSinceLastAnchor}$$

在一个连接事件或连接参数更新中，从机将监听窗口扩展，从传输窗口的起始到主机引导点传输窗口的最后。

在每一个随后的连接事件，从机将监听窗口扩展，从从机扩展引导点（slaveExpectedAnchorPoint）起始到从机扩展引导点的结束。

窗口扩展将小于 $((\text{connInterval}/2) - T_{\text{IFS}} \text{ us})$ ，如果窗口扩展达到 $((\text{connInterval}/2) - T_{\text{IFS}} \text{ us})$ ，将认为连接丢失。

4.5.8 数据信道索引选择

主机的链路层应该将数据信道分为已使用和未使用的，这称为信道表，最小的可用信道号为 2。

4.5.8.1 信道分类

主机应该为链路层提供信道分类信息，链路层将使用主机提供的信息，从机将从 CONNECT_REQ PDU 中接收信道映射，如果主机改变信道映射，将按 Section 5.1.2 定义的通知从机。

4.5.8.2 信道选择

信道选择策略有两部分组成：1、计算未映射的信道索引，2、将该信道索引映射到已使用信道的数据信道索引中。未映射信道（unmappedChannel）和最后未映射信道（lastUnmappedChannel）是两个连续连接事件的未映射信道索引。未映射信道是当前连接事件的未映射信道索引。最后未映射信道是前一个连接事件的未映射信道索引。连接的第一个连接事件的最后未映射信道应该为 0。

在连接事件的开始，未映射信道应该按下式计算：

$$\text{unmappedChannel} = (\text{lastUnmappedChannel} + \text{hopIncrement}) \bmod 37$$

当连接丢失时，lastUnmappedChannel 将设置为 unmappedChannel。

如果 unmappedChannel 是一个已使用的信道，信道选择策略将使用 unmappedChannel 做为连接事件的数据信道索引。

如果 unmappedChannel 是一个未使用信道，unmappedChannel 将被按下式重映射到一个已使用信道：
 $\text{remappingIndex} = \text{unmappedChannel} \bmod \text{numUsedChannels}$

其中 numUsedChannels 是信道映射中已使用信道的数量。

重映射表包括所有已使用信道，按升序排列，从 0 开始。remappingIndex 用于选择数据信道索引。整个完整的过程如下：

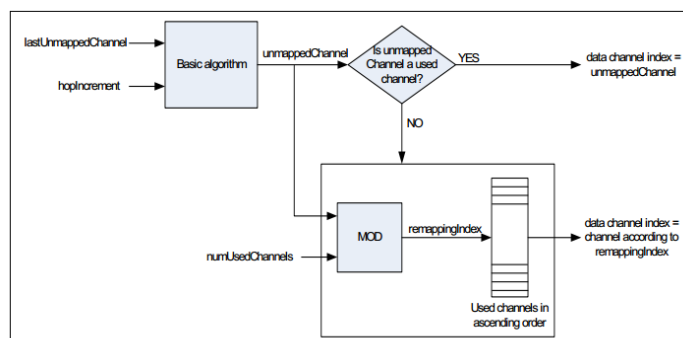


Figure 4.14: Block diagram of data channel selection algorithm

4.5.9 应答和流控

在所有链路层连接中，链路层应答和流控都应该被使用。

链路层中的所有连接有两个参数，`transmitSeqNum` 和 `nextExpectedSeqNum`，每个参数为一个位宽。`transmitSeqNum` 用于确定数据包是否由链路层发送，`nextExpectedSeqNum` 被对方使用，用于应答最后一个 Data Channel PDU 或请求重发 Data Channel PDU。

一旦进入连接状态 `transmitSeqNum` 和 `nextExpectedSeqNum` 应该被设置为 0。

一个新的 Data Channel PDU 是由链路层第一次发送的。最后一个 Data Channel PDU 是由链路层积压的。当一个 Data Channel PDU 被积压，其 LLID 域，SN 域，负载应该等于由链路层最后发送的 Data Channel PDU。

对于第一个新的 Data Channel PDU，头部的 SN 域应该设置为 `transmitSeqNum`，如果被积压，SN 域将不变。

一旦接收到一个 Data Channel PDU，SN 域将与 `nextExpectedSeqNum` 相比较，如果不等，表示这是一个积压数据 PDU，此时 `nextExpectedSeqNum` 不变。如果相等，表示这是一个新数据 PDU，并且 `nextExpectedSeqNum` 将加 1。

如一个数据信道 PDU 发送，头部的 NESN 域设置为 `nextExpectedSeqNum`。

一旦接收到一个数据信道 PDU，NESN 域与 `transmitSeqNum` 一样，则最后发送的数据信道 PDU 没有应答，并且将被积压。NESN 域与 `transmitSeqNum` 不一样，最后发送的数据信道 PDU 已应答，并且 `transmitSeqNum` 将加 1，且一个新的数据信道 PDU 将发送。

以上所有过程如下：

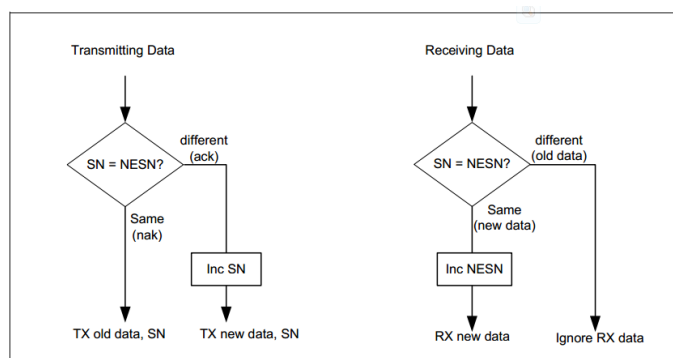


Figure 4.15: Transmit and Receive SN and NESN flow diagram

如果接收到的数据信道 PDU 的 CRC 匹配失效，NESN 不会改变，这意味着该数据信道 PDU 将不会被应答，这是引起对方重新发送数据信道 PDU 的原因。由于接收到的数据信道 PDU 被拒绝，对方发来的 NESN 不可信，因此，该设备最后发送的数据信道 PDU 不会被应答，并且必须重新发送。

SN, NESN 和 MD 位必须在每一个接收到的通过 CRC 校验的数据信道 PDU 中使用。数据信道 PDU 负载应该忽略每一个接收到的与前一接收到的具有相同 SN 值的数据信道 PDU。

4.5.9.1 流控

链路层可能因为某些原因不会更新 NESN，包括但不限于缓冲区的不足。这将导致对方稍后重传数据信道 PDU，因此使能流控。

4.6 属性支持

当这个信息从控制器发送到 HOST，对应位为 0 表示链路层 Feature 在控制器中不支持。对应位设置为 1 表示链路层 feature 在控制器中支持。

当这个信息从本地控制器发送到对方控制器，对应位为 0 表示链路层 feature 控制器不使用，对应位为 1，表示链路层 feature 在控制器中使用。

链路层 feature 对应位位置显示如表 4.3。表中显示了这些位可用于的意图。如果位显示不可用，使用 N 表示，则这个位在接收时被忽略。

Bit position	Link Layer Feature	Valid from Controller to Host	Valid from Host to Controller	Valid from Controller to Controller
0	LE Encryption	Y	Y	Y
1 – 63	RFU			

Table 4.3: FeatureSet field's bit mapping to Controller features

4.6.1 LE 加密

支持 LE 加密的控制器必须支持下部分：

Bit position	Link Layer Feature	Valid from Controller to Host	Valid from Host to Controller	Valid from Controller to Controller
0	LE Encryption	Y	Y	Y
1 – 63	RFU			

Table 4.3: FeatureSet field's bit mapping to Controller features

5.链路层控制

链路层控制协议用于控制及两个链路层之间的连接运行商议。这包括了连接控制的过程，开始和暂停加密和其它连接过程。

流程定义了超时规则，如 5.2 节所述。终端的处理过程可能被同时初始化，即使其它链路层控制流程当前是活跃的。对于所有其它的链路层控制过程，每一个设备的每一个连接同一时刻只能有一个链路层控制流程被初始化。一个新的链路层控制流程只可以在前一个链路层控制流程完成后才能被初始化。

链路层控制 PDU 和链路层数据 PDU 的优先级专门实现。比如：当链路终端被请求，数据 PDU 没有发送完成，主机不能假设未决数据将被发送。

5.1 链路层控制流程

5.1.1 连接更新流程

进入连接状态后，链路层连接参数(connInterval, connSlaveLatency and connSupervisionTimeout)可能被更新。

主机可以通过发送一个 LL_CONNECTION_UPDATE_REQ PDU 更新连接参数。从机不能发送这个 PDU，从机可能使用 L2CAP LE 信号信道请求更改连接参数。

主机链路层应该决定 connInterval，connInterval 范围为主机给定的间隔范围(connIntervalmin and connIntervalmax)。链路层应该指示主机已经选择的间隔值。LL_CONNECTION_UPDATE_REQ PDU 的即时字段用于指定 connEventCount 当更新参数被应用时，这被称为 instans。主机允许最少 6 个连接事件，这些事件从机必须在 instans 发生前监听。

Instant 之间使用的连接间隔称为 connInterval_OLD.连接间隔包括在 LL_CONNECTION_UPDATE_REQ PDU 中，而在 instant 时和后使用的连接间隔称为 connInterval_NEW.

Instant 前使用的连接从机等待称为 connSlaveLatency_OLD。连接从机等待包含在 LL_CONNECTION_UPDATE_REQ PDU 中，而在 instant 时和后使用的连接从机等待称为 connSlaveLatency_NEW.

Instant 前使用的连接超时管理称为 connSupervisionTimeout_OLD。连接从机等待包含在 LL_CONNECTION_UPDATE_REQ PDU 中，而在 instant 时和后使用的连接从机等待称为 connSupervisionTimeout_NEW.

比如：前一个连接事件和 instant 所在的连接事件中的连接间隔为 connInterval_OLD。Instant 所在的连接事件与下一个连接事件的连接间隔为 connInterval_NEW.

当从机接收到一个 LL_CONNECTION_UPDATE_REQ PDU，该 PDU 的 connEventCount 模 65536 小于 32767 并且 instant 不等于 connEventCount。从机将监听所有的连接事件，直到确认主机已经接收到 LL_CONNECTION_UPDATE_REQ PDU 的应答或 connEventCount 等于 instant.从机也会监听 connEventCount 等于 instant 的连接事件，或在其之前的连接事件。

当从机接收到一个 LL_CONNECTION_UPDATE_REQ PDU，该 PDU 的 connEventCount 模 65536 大于或等于 32767，从机链路层将认为连接丢失，链路层将退出连接状态，并切换到 Standby 状态并通知主机。

注意：connEventCount 和接收到的 instant 域之间的比较使用模 65536 计算，以处理 connEventCount 域被封装的情况。

当主机确定新连接参数第一个包传输的时序时，主机可能调整定位点。传输窗口将会被使用。传输窗口开始于 $\text{connIntervalOLD} + \text{transmitWindowOffset}$ ，在连接事件定位点之后，instant 之前。

transmitWindowOffset 应该是 1.25ms 的倍数，从 0ms- connIntervalNEW。transmitWindowSize 是 1.25ms 的整数倍，并在 $1.25\text{ms} - \min(10\text{ms}, \text{connIntervalNEW} - 1.25\text{ms})$ 。

主机应该在传输窗口中发送第一个数据包。主机的第一个数据包禁止超出传输窗口发送。

主机在 Instant 发送的第一个包决定连接事件新的定位点。因此，连接事件的所有特性的时序在这个连接中。

下一个连接事件定位点应该是在 instant 连接事件定位点之后的 connInterval_NEW。所有常用连接事件传输规则在 1.4.1 章节定义。

下图是连接更新过程的示例：

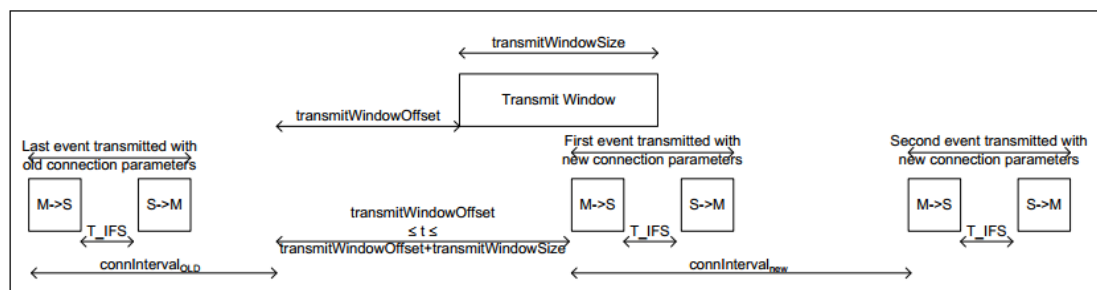


Figure 5.1: Connection event timing in the case of connection parameter update

传输窗口的开始，链路层将复位 T_LLconnSupervision。

如果主机链路层独立发送一个 LL_CONNECTION_UPDATE_REQ PDU，比如没有主机(HOST)发送请求，等待和超时参数不应该被改变，必须保持与 LL_CONNECTION_UPDATE_REQ 或 CONNECT_REQ PDU 之后相同，任何其它参数(transmitWindowSize, transmitWindowOffset, connInterval, Instant)可能在以上的限制中被改变。

注意：独立更新可以用于更新定位点以允许主机改变其它活跃者的连接调度。

链路层将通知他的 Host 三个连接参数中是否有改动。如果没有连接参数的改变，将不会通知 Host，这称为定位点移动。

当 instant 通过时流程完成。新的连接事件参数已经使用。

5.1.2 信道表更新过程

进入连接状态后，channelMap 的链路层参数可能被更新。主机将通过发送 LL_CHANNEL_MAP_REQ PDU 更新 channelMap。从机不允许发送该 PDU。

当 channelMapNEW 使用时，LL_CHANNEL_MAP_REQ PDU 的 Instant 域用于指示 connEventCount，这称为 instant。主机应该允许最少 6 个连接事件，从机将在 instant 发生前监听这些事件。

Instant 之前使用的 channelMap 称为 channelMapOLD. channelMap 包含在 LL_CHANNEL_MAP_REQ PDU 之内，而在 instant 时和后使用的 channelMap 称为 channelMapNEW。

当从机接收到一个 LL_CHANNEL_MAP_REQ PDU，该 PDU 的 connEventCount 模 65536 小于 32767，从机将监听所有的连接事件，直到确认主机已经接收到它对 LL_CHANNEL_MAP_REQ PDU 的应答或 connEventCount 等于 Instant。

当从机接收到一个 LL_CHANNEL_MAP_REQ PDU，该 PDU 的 connEventCount 模 65536 大于或等于 32767，从机链路层将认为连接丢失，并通知 Host。

注意: connEventCount 和接收到的 instant 域之间的比较使用模 65536 计算, 以处理 connEventCount 域被封装的情况。

当 connEventCount 等于 instant 域, channelMapNEW 将为当前 channelMap, lastUnmappedChannel 不会被复位。如果 unmappedChannel 是未使用的信道, channelMapNEW 将在重映射后使用。改变的参数只有 ChannelMap。

示例:

在连接设置时:

-初始化 channelMapOLD:0x1FFFFFFFFF(所有信道使能)

-初始化 hopincrement:10(十进制)

某 LL_CHANNEL_MAP_REQ PDU 具有以下参数:

-instant:100(十进制)。假设由开是连接开始, 没有连接事件计数。

-channelMapNEW: 0x1FFFFFF7FF(除信道 11 外其它信道都使能)

使用的信道:

-connEventCount 99 → 数据信道索引 1(channelMapOLD)

-connEventCount 100 → 数据信道索引 12(remapped from 11, channelMapNEW)

-connEventCount 101 → 数据信道索引 21(channelMapNEW)

当 instant 通过后, 流程完成。新的 channelmap 被使用。

5.1.3 加密流程

在链路层, 在进入连接状态后, HOST 端的请求可以使能数据包加密。

如果连接没有加密, 链路层只使用加密开始流程。

如果连接加密, 链路层首先使用加密暂停处理, 然后使用加密开始处理。

5.1.3.1 加密开始流程

为了使能加密, 必须交换 IV 和 SKD 两个参数。这两个参数由两个部分组成: 主部分和从部分, 且在 LL_ENC_REQ 和 LL_ENC_RSP PDUs 中交换。当这些被交换后, 且 HOST 已经通知链路层在该连接上已经使用 LTK, 加密可以使用一个三路握手开始, 使用 LL_START_ENC_REQ 和 LL_START_ENC_RSP PDUs。

为了开始加密, 链路层 MASTER 必须产生 IV 和 SKD 的主要部分(IVm 和 SKDm)。

链路层 MASTER 必须完成当前信道 PDU 的发送, 并完成控制器中额外数据信道 PDU 队列的发送。当这些数据信道 PDU 被应答, 链路层 MASTER 应该只发送空 PDUs 或 LL_ENC_REQ, LL_START_ENC_REQ, LL_START_ENC_RSP, LL_TERMINATE_IND or LL_REJECT_IND PDUs。

链路层 MASTER 随后必须发送一个 LL_ENC_REQ PDU; 随机数及 EDIV 域由 HOST 提供。

如果链路层的 SLAVE 不支持加密, 链路层 SLAVE 必须发送 LL_REJECT_IND PDU, 其错误代码设置为 "Unsupported Remote Feature / Unsupported LMP Feature" (0x1A)。链路层 MASTER 接收 LL_REJECT_IND PDU 必须通知 HOST。链路层 MASTER 从现在能发送 LL 数据包和 LL 控制包。这些包不会被加密。当 MASTER 从 SLAVE 接收到 LL_REJECT_IND PDU 时, 这个过程在 MASTER 中完成。当 SLAVE 从 MASTER 接收到 LL_REJECT_IND PDU 时, 这个处理在从机中完成。

然而, 当链路层 SLAVE 接收到一个 LL_ENC_REQ PDU 时, 它将产生 IV 和 SKD 的从部分 (IVs 和 SKDs), 并用随机域和 EDIV 域通知 HOST。

链路层从机必须完成当前信道 PDU 的发送, 且建议完成控制器中额外数据信道 PDU 队列的发送。当这些数据信道 PDU 被应答, 链路层从机只允许发送空 PDU 或 LL_ENC_RSP, LL_START_ENC_REQ, LL_START_ENC_RSP, LL_TERMINATE_IND or LL_REJECT_IND PDUs。

链路层从机随后必须发送一个 LL_ENC_RSP PDU。

每一个链路层必须以下面的方式组合初始化向量部分和会话密钥:

SKD = SKDm | SKDs

IV = IVm | IVs

SDKm 串联 SKDs, SKDm 的最低有效字节成为 SKD 的最低有位字节, SDKs 的最高有效字节成为 SKD 的最高有效字节。

IVm 串联 IVs. IVm 的最低有效字节成为 IV 的最低有位字节, IVs 的最高有效字节成为 IV 的最高有效字节。

LTK 由链路层 MASTER 和 SLAVE 的 HOST 提供, 以下三个行为之一会发生:

1、如果这个处理在暂停加密处理后被执行, 且 HOST 不提供 LTK, slave 将执行错误代码为 “PIN or key Missing” 的中断处理。

2、如果 HOST 不提供 LTK, 可能由于 HOST 事件被标志出来或 HOST 标志密钥不可用, slave 必须发送一个错误代码为 “PIN or key Missing” 的 LL_REJECT_IND PDU。一旦接收到一个 LL_REJECT_IND PDU, 链路层将通知 HOST。链路层从此时起可以发送 LL 数据 PDU 和 LL 控制 PDU。这些包不会被加密。当 MASTER 从 SLAVE 接收到 LL_REJECT_IND PDU 时, 这个过程在主机中完成。当 slave 从 master 接收到 LL_REJECT_IND PDU 的应答时, 这个过程在 slave 中完成。

3、如果 HOST 不提供 LTK, 链路层 SLAVE 必须从 MASTER 使用一个 LL_ENC_RSP PDU 应答 LL_ENC_REQ PDU。链路层必须使用以 LTK 为密钥的加密引擎计算会话密钥, 且 SKD 做为简单文本输入。会话密钥必须设置到加密引擎的输出。

会话密钥必须用于所有加密数据包的加密引擎的密钥。

当会话密钥被计算出后, 链路层 slave 必须发送一个 LL_START_ENC_REQ PDU。这个包将未加密发送, 且链路层必须设置在应答中接收一个加密数据包。

当链路层 master 接收到 LL_START_ENC_REQ PDU, 它将发送一个 LL_START_ENC_RSP PDU。这个 PDU 必须加密发送且设置为加密接收。

当链路层 slave 接收到 LL_START_ENC_RSP PDU, 它将传输一个 LL_START_ENC_RSP PDU。这个包将被加密发送。

当链路层 master 接收到 LL_START_ENC_RSP PDU, 连接将被加密。链路层从现在起发送 LL 数据 PDU 和 LL 控制 PDU。这些 PDU 将被加密发送。

链路层将通知 HOST 连接已经加密。

当 master 从 slave 接收到 LL_START_ENC_RSP PDU, 这个过程在 master 中完成。当 slave 从 master 接收到 LL_START_ENC_RSP PDU, 这个过程在 slave 中完成。

5.1.3.2 加密暂停处理

为了使能一个新的不需要断开连接的加密密钥被使用, 加密必须被禁止且然后再使能。在暂停时, 数据 PDU 必须加密发送以保护数据。

链路层 MASTER 必须完成当前数据信道 PDU 的发送, 建议完成控制器中额外数据信道 PDU 队列的发送。当这些数据信道 PDU 被应答, 链路层 master 必须只能发送空 PDU 或 LL_PAUSE_ENC_REQ or LL_TERMINATE_IND PDUs。

链路层 master 随后必须发送一个 LL_PAUSE_ENC_REQ PDU。

当链路层 slave 接收到 LL_PAUSE_ENC_REQ PDU, 它将完成当前的数据发送, 且建议完成控制器中额外数据信道 PDU 队列发送。当这些数据信道 PDU 被应答, 链路层 slave 只允许发送空 PDU 或 LL_PAUSE_ENC_RSP or LL_TERMINATE_IND PDUs。

链路层 slave 随后必须发送 LL_PAUSE_ENC_RSP PDU。这个包必须加密发送, 且链路层必须设置为加密接收。

当链路层 master 接收到 LL_PAUSE_ENC_RSP PDU, 它将设置为加密收发。它随后必须未加密发送一个 LL_PAUSE_ENC_RSP PDU 给 slave。

当链路层 slave 接收到 LL_PAUSE_ENC_RSP PDU, 它也将设置为未加密发送。

加密开始流程必须从现在起用于用新的会话密钥重新使能加密。

5.1.4 Feature 交换流程

链路层当前支持的 **feature** 设置 (**featureSet**) 参数建议在进入连接状态后交换。**Master** 可以使用 **LL_FEATURE_REQ** PDU 初始化这个流程, 且 **slave** 以一个 **LL_FEATURE_RSP** PDU 应答。**slave** 不能初始化这个流程。

featureSet 信息可能被缓存。如果这个信息没有为这些设备缓存, 链路层不应该在每一个连接上请求这些信息。设备的缓存信息可能不是正式的, 因此如果假设当前不支持 **feature** 的使用或由对方使用, 实例必须能够接受 **LL_UNKNOWN_RSP** PDU。

featureSetm 是链路层 **master** 的 **feature** 功能。当链路层 **master** 发送 **LL_FEATURE_REQ** PDU, **featureSet** 域必须设置为 **featureSetm**。

featureSets 是链路层 **slave** 的 **feature** 功能。

featureSet_used 是 **featureSetm** 和 **featureSets** 的逻辑与。当链路层 **slave** 发送 **LL_FEATURE_RSP** PDU, **featureSet** 域被设置为 **featureSet_used**。

链路层 **master** 发送 **LL_FEATURE_REQ** PDU。这可以在 **HOST** 请求或自动发送。

当链路层 **slave** 接收到 **LL_FEATURE_REQ** PDU, 它必须发送 **LL_FEATURE_RSP** PDU。链路层 **slave** 必须只使用在 **featureSet_used** 指示的处理。

当链路层 **master** 接收到 **LL_FEATURE_RSP** PDU, 它必须只使用 **featureSet_used** 指示的处理。

feature 交换示例如下图 5.2:

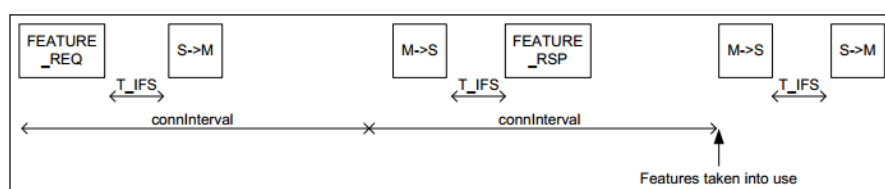


Figure 5.2: Feature Exchange Procedure

当 **master** 从 **slave** 接收到 **LL_FEATURE_RSP** PDU, 这个处理将在主机中完成。

5.1.5 版本交换

链路层版本信息 (**companyID**, **subVerNum**, **linkLayerVer**, as defined in Section 2.4.2.13)

参数在进入连接状态后可能被交换。链路层 **master** 或 **slave** 都可以通过发送 **LL_VERSION_IND** PDU 初始化这个处理。当由 **HOST** 请求时, 建议使用这个处理。这个处理可能被链路层自动初始化。

链路层必须在连接中排队以传输最多一个 **LL_VERSION_IND** PDU。

如果链路层接收一个 **LL_VERSION_IND** PDU 且还没有发送一个 **LL_VERSION_IND**, 则链路层必须发送一个 **LL_VERSION_IND** PDU 给对方设备。

如果链路层接收到 **LL_VERSION_IND** PDU 且已经接收到 **LL_VERSION_IND** PDU, 则链路层禁止发送另一个 **LL_VERSION_IND** PDU 给对方设备。

当从对方设备接收到一个 **LL_VERSION_IND** PDU, 则处理完成。

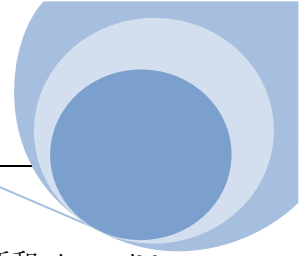
5.1.6 终止处理

这个处理用于主动终止连接。当 **HOST** 请求链路层终止连接时, 主动终止发生。链路层的 **master** 或 **slave** 可以通过发送 **LL_TERMINATE_IND** PDU 初始化这个处理。终止处理不能用于连接丢失事件中, 比如当连接管理超时后或处理超时后。

链路层必须起动一个定时器, **Tterminate**, 当 **LL_TERMINATE_IND** PDU 已经排队发送。初始化链路层必须发送 **LL_TERMINATE_IND** PDU 直到接收到应答或定时器 **Tterminate** 溢出。**Tterminate** 初始值必须设置为 **connSupervisionTimeout** 的值。

当链路层接收到 **LL_TERMINATE_IND** PDU 时, 它必须发送应答, 退出连接状态并切换到待机状态。

这个处理在接收到应答时完成。



5.2 流程应答超时

本章规定了流程超时规则，本规则将用于 5.1 规定的所有链路层控制流程，除连接更新和 channelMap 流程外，这两个流程不需要超时规则。

为了能够删除没有应答的链路层控制流程，主机和从机应该使用一个流程应答超时定时器——T_{prt}。根据流程的初始化，流程应答超时定时器将复位和启动。

每一个传输队列中的链路层控制 PDU 都会复位流程应答超时定时器。

如果流程应答超时定时器达到 40s，连接认为丢失。链路层退出连接状态且应该切换到待机状态。主机(Host)应该通知连接的丢失。

Volum 3 - Core System Package[Host volume]

PART A L2CAP (LOGICAL LINK CONTROL AND ADAPTATION PROTOCOL SPECIFICATION)

1、介绍

蓝牙标准本部分定义了逻辑链路控制和适应层协议(LLCAP 或 L2CAP)，L2CAP 向上层协议提供了支持协议多路复用，分割及重组操作的双向连接和无连接数据服务。L2CAP 允许上层协议和应用传输的接收高达 64K 字节的数据包。L2CAP 也允许每一个信道数据量和数据重传。

1.1 L2CAP 特征

L2CAP 功能包括协议信道复用，数据分割和重组(SAR)，每一个信道数据流控制和错误控制，L2CAP 位于以下其中一个底层之上：

- 1、BR/EDR 控制和 0 或多个 AMP 控制器或
- 2、BR/EDR/LE(支持 BR/EDR 和 LE)控制器和 0 或多个 AMP 控制器或
- 3、LE 控制器（只支持 LE）

L2CAP 与上层协议接口。

31 图的图 1.1 描述了 L2CAP 的组成部分。其中信道管理提供了控制面板功能并对所有内部信令、L2CAP 点对点信令和上层及下层的信令负责。执行 98 页第六章描述的状态机并使用 54 页第四章描述的消息格式。重传和流控部分提供了每一个信道的数据流控制并使用包重传恢复数据错误。资源管理部分负责为信道管理，重传和流控块和不需要重传和流控服务的应用数据流提供中继服务。它负责协调在多个 L2CAP 信道中数据包的传输和接收。

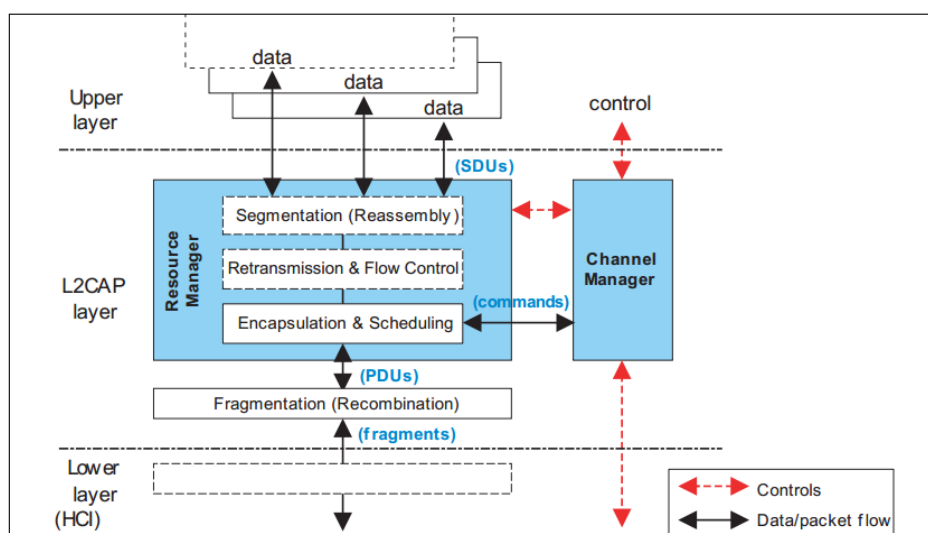


Figure 1.1: L2CAP architectural blocks

● 协议信道复用：

L2CAP 支持在单个控制器或多个控制器间的复用。同一时刻，一个 L2CAP 信道必须在工作在一个控制器上。在信道设置中，协议信道复用功能用于路由连接到正确的上层协议。

对于数据传输，逻辑信道必须与多个上层实体间区分。在同一个协议中可能存在多个上层实体。

● 分割与重组：

帧中继服务由资源管理器提供，传输帧的长度由独立运行在 L2CAP 的应用控制。如果 L2CAP 可以控制 PDU 的长度可以更好的服务大多数复用应用。主要有以下好处：

- a. 分割将允许应用数据单元的交叉以满足等待要求
- b. 当 L2CAP 控制包长度时，内存的缓冲管理更容易实现
- c. 重传错误校正更高效
- d. 当 L2CAP PDU 损坏或丢失时，被释放的数据量双应用数据单元小
- e. 降低应用程序在分割要求与数据包映射到底层数据包之间的耦合度

● 第一个 L2CAP 信道流控

控制器提供为空中传输的数据和因 HCI 数据传输而存在的 HCI 流控提供出错和流控管理。当同一个控制器的数据流使用不同的 L2CAP 信道，每一个信道需要独立实现流控。基于流控框架的窗口将提供。

● 出错控制和重传：

当 L2CAP 信道从一个控制器移动到另外一个控制器可能会丢失。同样，一些应用需要一个小于控制器可实现的残余错误比率。L2CAP 提供错误检查和 L2CAP PDUs 的重传。在 L2CAP 中的错误检查保护数据免于因控制器错误接收包含错误但通过控制器基本的数据完整性检查而造成出错。L2CAP 错误检查和重传同样保护数据免于因控制器清除而丢失数据包。出错控制工作在流控节点以感知每一次传输及流控机制将节制重传。

● 数据流支持：

数据流应用如音频使用双方协定的数率以设置一个 L2CAP 信道并不希望流控机制及控制器修改数据流中的数据。清除超时用于保持数据在传输方。数据流模式用于禁止 HCI 和基于流控的控制器在接收方被使用。

● 分割及重组：

一些控制器可能有传输能力的限制或可能需要分割成与 L2CAP 分割创建的数据不同的大小。因此，L2CAP 以下的协议层可能进一步分割和重组 L2CAP PDUs 以满足本层传输能力。在一个 L2CAP PDU 传输中，多个不同等级的分割和重组可能在两个对等设备中发生。

HCI 驱动或控制器可能分割 L2CAP PDUs 以满足主机控制器包尺寸限制。这会导致 HCI 数据包负载带有起始和连续分割包。相似的，控制器可能分割 L2CAP PDUs 以映射到控制器包中。这可能导致在控制器数据包中有起始及连续分割包。

每一个协议栈层可能传输不同尺寸的 L2CAP PDUs，且由每一个协议层建立的分割包大小可能在每一个设备中都不一样。然而 PDU 在协议栈内被分割，L2CAP 实体接收仍然重组分割包以获得完整的 L2CAP PDU。

● 服务质量：

L2CAP 连接建立流程允许在两个目标蓝牙设备间忽略服务质量（QoS）的信息交换。每一个 L2CAP 实现监控协议层使用的资源并保证 QoS 协定是被遵循的。

对于 BR/EDR 或 BR/EDR/LE 控制器，L2CAP 可能支持同步和异步数据流，通过相同的 ACL 逻辑链路，标志包包为自动清除或非自动清除通过设置 HCI ACL 数据包中的 Packet_Boundary_Flag 为对应值。自动清除 L2CAP 数据包通过为已在 L2CAP 信道映射的 ACL 逻辑链路设置自动清除超时清除。非自动清除 L2CAP 数据包不受自动清除超时影响，即不会清除。所有的 L2CAP 数据包能够被 HCI 清除命令清除。

对于 AMP 控制器，L2CAP 通过单独逻辑链路放置所有异步数据流到同一个远程设备。L2CAP 放置第一个同步数据流到其自己的逻辑链路。

1.2 假设

本协议基于以下假设而设计

1、控制器提供数据包的顺序传输，尽管单个数据包可能存在损坏和重复。对于 BR/EDR 或 BR/EDR/LE 控制器设备，在两个设备中不超过一个 ACL-U 逻辑链路存在。对于一个给定的 AMP 控制器设备，两个设备间可能存在多个 AMP-U 逻辑链路。对于 BR/EDR/LR 或 LE 控制器，两个设备间不超过一个 LE-U 逻辑链路存在。

2、控制器总是提供全双向通信信道的的印记。这并不是指所有的 L2CAP 通信是双向的。非双向通信不需要双重的信道。

3、L2CAP 层提供一种信道，这种信道在基于在控制器中可用机制中有一定程序的可靠性且带有额外的包分割，出错检测及重传，这些可在增加 L2CAP 层中使能。一些控制器执行数据完整性检查并重新发送数据直接数据被成功应答或超时发生。其它控制器将重传数据，直到一个确定的次数，以致数据被清除。因为应答可能丢失，即使数据已经成功发送后超时也可能发生。注意，在 BR/EDR 或 BR/EDR/LE 控制器中使用基带广播数据包是不可靠的，且所有的广播开始第一个相同 序列位的 L2CAP 数据包的分割包。

4、控制器为数据提供出错和流控，这些数据是在空中传输和 HCI 传输数据中的流控，但一些应用将希望有比控制器提供的更好的出错控制。也就是说，在控制器之间移动信道需要 L2CAP 流控及出错控制。流控和出错控制块提供四种模式，增强重传模式和重传模式提供分包，流控和 L2CAP PDU 重传。流控模式提供分包的流控功能。数据流模式提供分包的接收包清除。

1.3 范围

以下特征超出 L2CAP 负责的范围。

L2CAP 不传输为 SCO 和 eSCO 逻辑传输设计的同步数据

L2CAP 不支持可靠广播通道

1.4 术语

Term	Description
Upper layer	The system layer above the L2CAP layer, which exchanges data with L2CAP in the form of SDUs. The upper layer may be represented by an application or higher protocol entity known as the Service Level Protocol. The interface of the L2CAP layer with the upper layer is not specified.
Lower layer	The system layer below the L2CAP layer, which exchanges data with the L2CAP layer in the form of PDUs, or fragments of PDUs. The lower layer is mainly represented within the Controller, however a Host Controller Interface (HCI) may be involved, such that an HCI host driver could also be seen as the lower layer. Except for the HCI functional specification (in case HCI is involved) the interface between L2CAP and the lower layer is not specified.
L2CAP channel	The logical connection between two endpoints in peer devices, characterized by their Channel Identifiers (CID), which is multiplexed over one or more Controller based logical links.
SDU, or L2CAP SDU	Service Data Unit: a packet of data that L2CAP exchanges with the upper layer and transports transparently over an L2CAP channel using the procedures specified here. The term SDU is associated with data originating from upper layer entities only, i.e. does not include any protocol information generated by L2CAP procedures.
Segment, or SDU segment	A part of an SDU, as resulting from the Segmentation procedure. An SDU may be split into one or more segments. Note: this term is relevant only to Enhanced Retransmission mode, Streaming mode, Retransmission Mode and Flow Control Mode, not to the Basic L2CAP Mode.

Table 1.1: Terminology

Term	Description
Segmentation	A procedure used in the L2CAP Retransmission and Flow Control Modes, resulting in an SDU being split into one or more smaller units, called Segments, as appropriate for the transport over an L2CAP channel. Note: this term is relevant only to the Enhanced Retransmission mode, Streaming mode, Retransmission Mode and Flow Control Mode, not to the Basic L2CAP Mode.
Reassembly	The reverse procedure corresponding to Segmentation, resulting in an SDU being re-established from the segments received over an L2CAP channel, for use by the upper layer. Note that the interface between the L2CAP and the upper layer is not specified; therefore, reassembly may actually occur within an upper layer entity although it is conceptually part of the L2CAP layer. Note: this term is relevant only to Enhanced Retransmission mode, Streaming mode, Retransmission Mode and Flow Control Mode, not to the Basic L2CAP Mode.
PDU, or L2CAP PDU	Protocol Data Unit a packet of data containing L2CAP protocol information fields, control information, and/or upper layer information data. A PDU is always started by a Basic L2CAP header. Types of PDUs are: B-frames, I-frames, S-frames, C-frames and G-frames.
Basic L2CAP header	Minimum L2CAP protocol information that is present in the beginning of each PDU: a length field and a field containing the Channel Identifier (CID).
Basic information frame (B-frame)	A B-frame is a PDU used in the Basic L2CAP mode for L2CAP data packets. It contains a complete SDU as its payload, encapsulated by a Basic L2CAP header.
Basic information frame (B-frame)	A B-frame is a PDU used in the Basic L2CAP mode for L2CAP data packets. It contains a complete SDU as its payload, encapsulated by a Basic L2CAP header.
Information frame (I-frame)	An I-frame is a PDU used in Enhanced Retransmission Mode, Streaming mode, Retransmission mode, and Flow Control Mode. It contains an SDU segment and additional protocol information, encapsulated by a Basic L2CAP header
Supervisory frame (S-frame)	An S-frame is a PDU used in Enhanced Retransmission Mode Retransmission mode, and Flow Control Mode. It contains protocol information only, encapsulated by a Basic L2CAP header, and no SDU data.
Control frame (C-frame)	A C-frame is a PDU that contains L2CAP signaling messages exchanged between the peer L2CAP entities. C-frames are exclusively used on the L2CAP signaling channel.
Group frame (G-frame)	A G-frame is a PDU exclusively used on the Connectionless L2CAP channel. It is encapsulated by a Basic L2CAP header and contains the PSM followed by the completed SDU. G-frames may be used to broadcast data to multiple slaves (either to all slaves via Piconet Broadcast or to only active slaves via Active Broadcast) or to send unicast data to a single remote device.

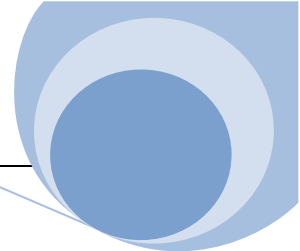
Table 1.1: Terminology

Term	Description
Fragment	A part of a PDU, as resulting from a fragmentation operation. Fragments are used only in the delivery of data to and from the lower layer. They are not used for peer-to-peer transportation. A fragment may be a Start or Continuation Fragment with respect to the L2CAP PDU. A fragment does not contain any protocol information beyond the PDU; the distinction of start and continuation fragments is transported by lower layer protocol provisions. Note: Start Fragments always begin with the Basic L2CAP header of a PDU.
Fragmentation	A procedure used to split L2CAP PDUs to smaller parts, named fragments, appropriate for delivery to the lower layer transport. Although described within the L2CAP layer, fragmentation may actually occur in an HCI host driver, and/or in a Controller, to accommodate the L2CAP PDU transport to HCI data packet or Controller packet sizes. Fragmentation of PDUs may be applied in all L2CAP modes. Note: in version 1.1, Fragmentation and Recombination was referred to as "Segmentation and Reassembly".
Recombination	The reverse procedure corresponding to fragmentation, resulting in an L2CAP PDU re-established from fragments. In the receive path, full or partial recombination operations may occur in the Controller and/or the Host, and the location of recombination does not necessarily correspond to where fragmentations occurs on the transmit side.
Maximum Transmission Unit (MTU)	The maximum size of payload data, in octets, that the upper layer entity is capable of accepting, i.e. the MTU corresponds to the maximum SDU size.
Maximum PDU payload Size (MPS)	The maximum size of payload data in octets that the L2CAP layer entity is capable of accepting, i.e. the MPS corresponds to the maximum PDU payload size. Note: in the absence of segmentation, or in the Basic L2CAP Mode, the Maximum Transmission Unit is the equivalent to the Maximum PDU payload Size and shall be made equal in the configuration parameters.
Signaling MTU (MTU _{sig})	The maximum size of command information that the L2CAP layer entity is capable of accepting. The MTU_{sig} refers to the signaling channel only and corresponds to the maximum size of a C-frame, excluding the Basic L2CAP header. The MTU_{sig} value of a peer is discovered when a C-frame that is too large is rejected by the peer.
Connectionless MTU (MTU _{cni})	The maximum size of the connection packet information that the L2CAP layer entity is capable of accepting. The MTU_{cni} refers to the connectionless channel only and corresponds to the maximum G-frame, excluding the Basic L2CAP header and the PSM which immediately follows it. The MTU_{cni} of a peer can be discovered by sending an Information Request.

Table 1.1: Terminology

Term	Description
MaxTransmit	In Enhanced Retransmission mode and Retransmission mode, MaxTransmit controls the number of transmissions of a PDU that L2CAP is allowed to try before assuming that the PDU (and the link) is lost. The minimum value is 1 (only 1 transmission permitted). In Enhanced Retransmission mode a value 0 means infinite transmissions. Note: Setting MaxTransmit to 1 prohibits PDU retransmissions. Failure of a single PDU will cause the link to drop. By comparison, in Flow Control mode, failure of a single PDU will not necessarily cause the link to drop.

Table 1.1: Terminology



2、一般操作

L2CAP 是基于 ‘Channels’ 管道的概念。每一个 L2CAP 管道端点通过管道 ID (CID) 引用。

2.1 管道 ID

一个 CID 是一个本地名，以代表在设备上的一个逻辑管道端点。空 ID (0X0000) 不能用于目标端点。ID 号从 0X0000 到 0X003F 预留为特殊的 L2CAP 功能。这些管道称为固定管道。同时，L2CAP 信令管道（固定管道 0x0001）或 L2CAP LE 信令管道（固定管道 0X0005）应该支持。如果固定管道 0X0005 支持，固定管道 0X0004 和 0X0006 必须支持。其它固定管道可能支持。请求/应答机制信息 (described in Section 4.10 and Section 4.11) 必须用于确定远程设备哪一个固定信道通过 ACL-U 逻辑链路支持。

每一个固定信道特征定义在每一个信道基础上。固定信道特征包括配置参数（比如可靠性，MTU 大小，QoS），加密和使用 L2CAP 配置机制修改参数的能力。表 2.1 列出了定义的固定管道，提供了相关管道特征定义和通过可能工作的管道逻辑链路的定义的参考。固定管道一旦 ACL-U 或 LE-U 逻辑链路建立即可用。当管道建立时正常执行的所有初始化过程将在每一个支持的固定管道中被执行。固定管道将只运行在 ACL-U 或 LE-U 逻辑链路中，且不能被移动。

实现是自由的，以使用一种最适合实际应用的方式管理剩余的 CID，规定：两个相似的激活 L2CAP 管道不能共享相同的 CID。表 2.1 总结了 CID 命令空间的定义和划分。

动态分配 CID 与实际设备有关，且设备可能从其它设备安排 CID 实现。因此，即使相同的 CID 值已经被一些与单独本地设备连接的远端设备分配到管道端点，本地设备可以和一个不同的设备唯一关联每一个远程 CID。

CID	Description	Channel Characteristics	Logical Link Supported
0x0000	Null identifier	Not allowed	
0x0001	L2CAP Signaling channel	See Section 4 on page 54	ACL-U
0x0002	Connectionless channel	See Section 7.6 on page 133	ACL-U
0x0003	AMP Manager Protocol	See [Part E] Section 2.2 on page 444.	ACL-U
0x0004	Attribute Protocol	See [Vol. 3], Part F	LE-U
0x0005	Low Energy L2CAP Signaling channel	See Section 4 on page 54	LE-U
0x0006	Security Manager Protocol	See [Vol. 3], Part H	LE-U
0x0007-0x003E	Reserved	Not applicable	
0x003F	AMP Test Manager	See Part D, Section 1.2.3	ACL-U
0x0040-0xFFFF	Dynamically allocated	Communicated using L2CAP configuration mechanism (see Section 7.1 on page 120.	ACL-U, AMP-U

Table 2.1: CID name space

2.2 设备间的操作

图 2.1 描述了 CID 在不同设备对应的 L2CAP 实现的通信中的使用。面向连接的数据信道代表了两个设备间的连接，CID 识别管道的每一个端点。当用于广播传输，无连接管道约束数据流为单向的。无连接管

道可能用于传输数据到微网内所有从机，或所有激活的从机。当用于单一广播传输时，无连接管道可能使用在主机和从机的任意一方。

也有一部分 CID 预留为特殊目的。L2CAP 信令是预留管道的一个例子。这个管道用于创建和建立面向连接的数据管道并商议面向连接管道中特征的改变且发现工作在 ACL-U 逻辑链路上的无连接管道特征。

L2CAP 信令管道和所有支持的固定管道在 ACL-U 逻辑链路在两个设备间建立后立即可用。另外一个 CID(0X0002)预留做为所有输入的输出无连接数据通道，无论是全向广播还是单一广播。无连接数据通道可能立即流动一旦 ACL-U 逻辑链路在两个设备之间建立且一旦传输设备已经确定远端设备支持无连接通道。

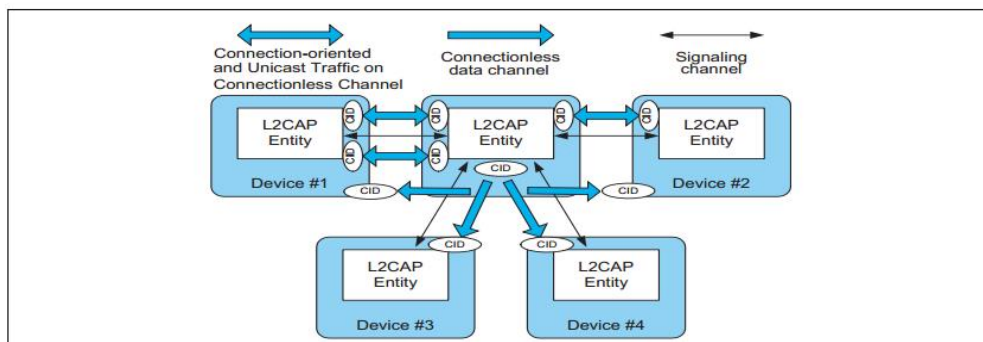


Figure 2.1: Channels between devices

表 2.2 描述了可变管道类型和他们的资源及目标识别。一个动态分配 CID 被分配为识别本地端点且必须在 0X0040 到 0XFFFF 之间。98 页第 6 章描述了每一个动态分配 CID 的面向连接管道间连接的状态机。44 页 3.1 节和 45 页 3.3 节描述了面向连接管道的数据包格式。45 页 3.2 节描述了无连接管道的数据包格式。

Channel Type	Local CID (sending)	Remote CID (receiving)
Connection-oriented	Dynamically allocated and fixed	Dynamically allocated and fixed
Connectionless data	0x0002 (fixed)	0x0002 (fixed)
L2CAP Signaling	0x0001 and 0x0005 (fixed)	0x0001 and 0x0005 (fixed)

Table 2.2: Types of Channel Identifiers

2.3 协议层间操作

L2CAP 实现应该遵循以下描述的的一般架构。L2CAP 实现上层协议和下层协议间的数据传输。这个文档列出了一些通过任何 L2CAP 实现扩展的服务。每一个实现应该支持在 L2CAP 实现间使用的信令命令集。L2CAP 实现应该准备接收来自下层协议的确定类型的事件并向上层协议产生事件。这些事件如何在层间传输有专门的定义。

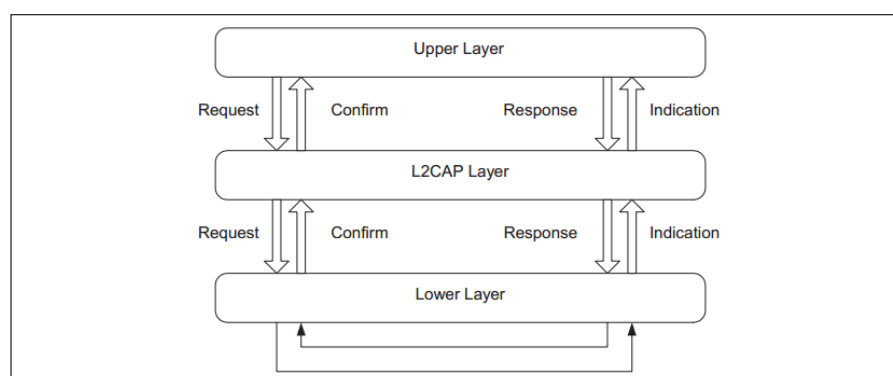
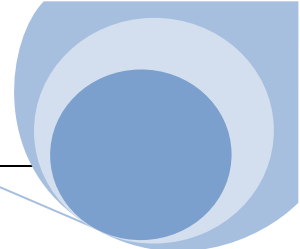


Figure 2.2: L2CAP transaction model



2.4 模式间操作

L2CAP 可能工作在五个不同的模式之一，这些模式如下：

- a、基本的 L2CAP 模式
- b、流控模式
- c、重传模式
- d、增强重传模式
- e、数据流模式

这些模式使用 7.1 章节描述的配置流程使能。基本 L2CAP 模式必须是默认模式，当没有其它模式约定时使用。增强重传模式必须用于所有的通过 AMP-U 逻辑链路建立的可靠管道及 7.10 章节描述的 ACL-U 逻辑链路操作。增强重传模式应该为通过不工作在章节 7.10 描述的 ACL-U 逻辑链路建立的可靠管道而使能。数据流模式必须用于 AMP-U 逻辑链路建立的数据流应用和工作在章节 7.10 描述的 ACL-U 逻辑链路。数据流模式应该为不工作工作在章节 7.10 描述的由 ACL-U 逻辑链路建立的数据流应用而使能。无论增强重传模式或数据流模式应该在支持 L2CAP 实例中使能。流控模式和重传模式必须只有在不支持增强重传模式或数据流模式的 L2CAP 实例的通信中使能。

在流控模式，重传模式和增强重传模式中，在实例中传输的 PDU 被计数并应答。PDU 中的序列号用于控制缓冲，且 TxWindow 大小用于限制缓冲空间的要求且或提供流控的方法。

在流控模式，没有重传产生，但丢失的 PDU 同样被检测且能被报告为丢失。

在重传模式，有一个定时器用于保证所有的 PDU 被传输到对方，一个回退 n 重传机制用于简化协议并限制缓冲要求。

增强重传模式与重传模式相似。它增加了设置 POLL 位以从远端 L2CAP 实例征求应答的能力，增加了 SREJ S 帧以改善出错恢复的效率，还增加了 RNR S 帧以替换 R 位来报告本地忙条件。

数据流模式用于实时同步管道。PDU 被计数但不会应答。一个限定的清除超时在发送方设置以定时的方式清除没有发送的数据包。在接收方，如果在一个新的 PDU 接收到时接收缓冲满，则前面接收到的 PDU 被覆盖。丢失的 PDU 可以被检测到并报告为丢失。TxWindow 大小不能在数据流模式中使用。

注意：

尽管 L2CAP 基本模式可能用于工作在 ACL-U 逻辑链路的 L2CAP 管道，只有已经配置为使用增强传输模式或数据流模式的 L2CAP 管道可能移动到工作在 AMP-U 逻辑链路。参考 4.16 章节。

L2CAP 管道用于复用协议层，比如 RFCOMM 能够服务多个更高协议层。在这种情况下，本地和远程设备共同支持一个 AMP 和补充支持配置文件在一个多路复用层需要可靠性（例如 RFCOMM），多路复用层应该配置使用增强的传输模式以确保 profile 利用多路复用器允许移动 AMP-U 逻辑链接。同样的，在这种情况下，本地设备和远程设备有共同支持一个 AMP 和补充支持配置文件在一个多路复用层不需要可靠性、多路复用层应该配置为使用流媒体模式以保证 profile 利用多路复用器并不阻止被搬到 AMP-U 逻辑链接。

当用于 profile 实例下方以确保性能没有压缩，应该小心的选择参数用于增强重传模式和数据流模式。当绝对不能相互支持 AMP 时，互补支持利益于 AMP 的 PROFILE，它可以更好的配置基本模式以最小化性能压缩的风险。

2.5 映射信道到逻辑链路

L2CAP 映射管道到控制逻辑链路，依次运行在控制器物理链路。所有的逻辑链路运行在本地控制器和运行单独物理链路的远端控制器之间。每一个 BR/EDR 物理链路有一个 ACL-U 逻辑链路，而每一个 LE 物理链路有一个 LE-U 逻辑链路。但每一个 AMP 物理链路可能有一个或多个 AMP-U 逻辑链路。

两个设备的 BR/EDR 物理链路努力管道和保证管道映射到一个单独的 ACL-U 逻辑链路。两个控制器间的 AMP 物理链路的努力管道映射到单独的 AMP-U 逻辑链路，而每一个保证管道映射到其自己的 AMP-U 逻辑链路，即每一个保证管道对应一个 AMP-U 逻辑管道。两个设备的所有 LE 物理链路管道必须认为是努力管道并映射到一个单独的 LE-U 逻辑链路。

当一个保证管道建立或移到一个控制器，一个对应的保证逻辑链路应该被创建以搭载管道通信。保证逻辑链路的创建包含许可控制。许可控制验证保证可以不需要对应的已经存在保证而达到。对于一个 AMP 控制器，L2CAP 必须告诉控制器创建一个保证逻辑链路且许可控制必须由控制器执行。对于一个 BR/EDR 控制器，许可控制必须由 L2CAP 层执行。

3、数据包格式

L2CAP 是基于数据包的，但通信模式是基于管道的。在无端设备间，一个管道代表一个 L2CAP 实例间的数据流。管道可能是面向连接的或无连接的。固定管道（除 L2CAP 无连接管道（CID 0X0002）之外）和两个信令管道（CID 0X0001 和 0X0005）被认为是面向连接的。所有动态分配 CID 的管道是面向连接的。所有 L2CAP 层包必须使用字节小端对齐，但信息负载域除外。高层协议层 L2CAP 信息负载的字节封装对齐方式是协议定义的。

3.1 基本 L2CAP 面向连接管道

下图描述了在面向连接管道中使用的 L2CAP PDU 格式。在基本 L2CAP 模式，在面向连接管道上的 L2CAP PDU 也可称为“B-frame”。

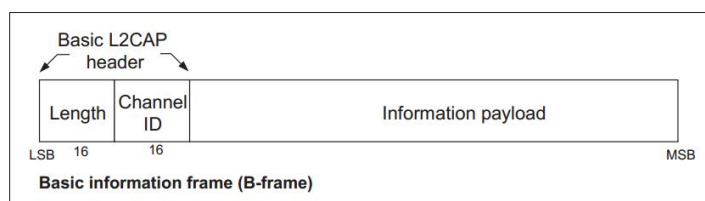


Figure 3.1: L2CAP PDU format in Basic L2CAP mode on connection-oriented channels (field sizes in bits)

其中各个位域如下：

长度：2 字节，长度指示了信息负载域的字节数，不包括 L2CAP 头长度。信息负载域的长度可以高达 65535 字节。在接收端，长度域用于重组和简单的验证 L2CAP 重组包的完整性

CID：2 字节，CID 识别数据包的目标管道端点。

信息负载：0-65535 字节，该域包含从上层协议接收到或将传输给下层协议的负载。动态分配 CID 管道的 MTU 在管道配置期间决定。参考 79 页 5.1 章节。信令管道 PDUS 最小支持的 MTU 值在 54 页的表 4.1 描述。

3.2 基本 L2CAP 模式中的无连接数据管道

图 3.2 描述在无连接数据管道中的 L2CAP PDU 格式。这里 L2CAP PDU 也称为“G-frame”。

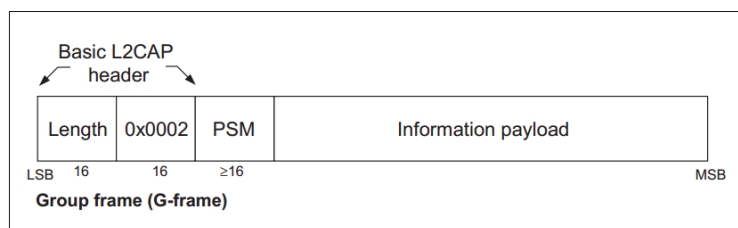


Figure 3.2: L2CAP PDU format on the Connectionless channel

相关域描述如下：

长度：2 字节，长度字节标志了信息负载的字节数，包括 PSM 域。

CID：2 字节，CID 0X0002 预留为无连接通信。

PSM, 协议服务复用器：至少 2 字节，参考 58 页 4.2 章节

信息负载：0-65533 字节，该域包含负载信息以分配微微网内所有从机的广播无连接通信，或为远端特定设备从 L2CAP 无连接管道发送数据。实例应该在无连接管道支持 48 字节的无连接 MTU。设备可能明确的改变成一个更大或更小的无连接 MTU。

注意：信息负载域的最大尺寸将减小，如果 PSM 域扩展超过两个字节。

3.3 重传、流控、数据流模式的面向连接管道

为了支持流控，重传和数据流，L2CAP PDU 类型在基本 L2CAP 头的基础上加入了协议元素定义。信息帧（I-frame）用于在 L2CAP 实例间传输信息。超帧（S-frame）用于应答 I-frames 和请求 I-frame 的重传。

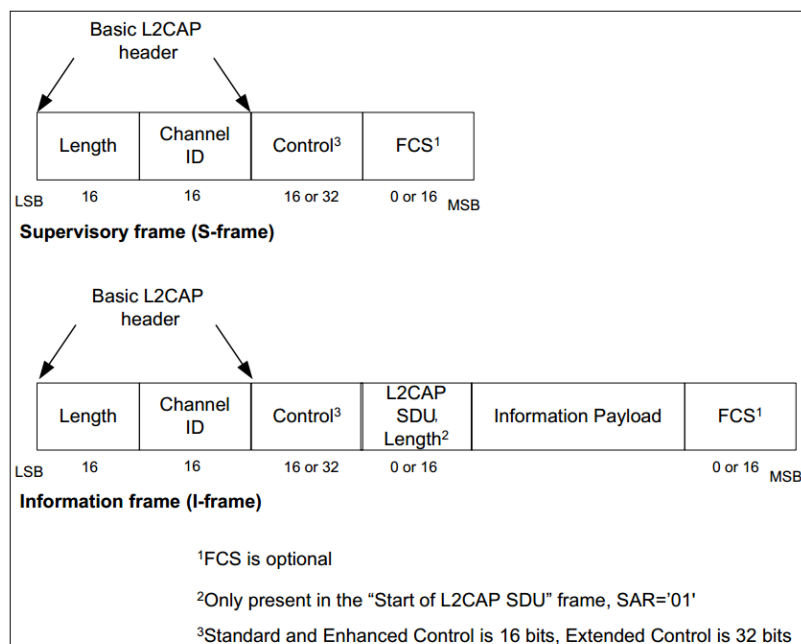


Figure 3.3: L2CAP PDU formats in Flow Control and Retransmission Modes

3.3.1 L2CAP 头域

长度：2 字节，L2CAP PDU 头两个字节包括整个 L2CAP PDU 的字节长度，除了长度和 CID 域以外。对于 I-frames 和 S-frames，长度域包括控制域长度，L2CAP SDU 长度，信息字节长度和帧校验序列长度（FCS）。I-frame 信息最大字节数基于相关域是否存在及控制域的类型。带有标准控制域的 I-frame 的信息域最大字节数如下：

L2CAP SDU 长度域存在，FCS 存在	65529 字节
L2CAP SDU 长度域存在，FCS 不存在	65531 字节
L2CAP SDU 长度域不存在，FCS 存在	65531 字节
L2CAP SDU 长度域不存在，FCS 不存在	65533 字节

带有扩展控制域的 I-frame 的信息域最大字节数如下：

L2CAP SDU 长度域存在，FCS 存在	65527 字节
L2CAP SDU 长度域存在，FCS 不存在	65529 字节
L2CAP SDU 长度域不存在，FCS 存在	65529 字节
L2CAP SDU 长度域不存在，FCS 不存在	65531 字节

CID：2 字节，该域包含 CID 内容。

3.3.2 控制域，2 或 4 字节

控制域用于识别帧类型。有三种控制域格式：标准控制域，增强控制域和扩展控制域。标准控制域必须用于重传模式和流控模式。增加控制域必须用于增强重传模式的数据流模式。扩展控制域应该用于增强

重传模式的数据流模式。控制域将包含合适的序列号。序列号如 48 页表 3.1, 49 页表 3.2 和 49 页表 3.3 所示编码。有两种不同的帧类型, 信息帧类型和超帧类型。信息帧和超帧类型由控制域的最后一位区分, 如表 3.1, 49 页表 3.2 和表 3.3 所示。

信息帧格式 (I-frame)

I-frame 用于在 L2CAP 实例间传输信息。每一个 I-frame 有一个 TxSeq (发送序列号), ReqSeq (接收序列号), 可能应答从数据链路层实例接收到的额外 I-frame。第一个带有标准控制域的 I-frame 有一个重传位 (R 位), R 位决定 I-frame 是否重传。每一个带有增强控制域或扩展控制域的 I-frame 有一个 F 位, 用于 Poll/Final 位功能。

I-frame 中的 SAR 域用于分割和重组控制。L2CAP SDU 长度域决定了 SDU 的长度, 包括所有分割包的总长度。

管理帧 (S-frame)

S-frame 用于应答 I-frame 和请求 I-frame 的重传。每一个 S-frame 有一个 ReqSeq 序列号, 可用于应答数据链路层实例接收到的 I-frame。每一个带有标准控制域的 S-frame 有一个 R 位, 用于决定 I-frame 是否需要重传。每一个带有增强控制域或扩展控制域的 S-frame 有一个 Poll 位 (P 位) 和一个 Final 位 (F 位) 但没有 R 位。

S-frame 定义的类型有 RR (Receiver Ready), REJ (Reject), RNR (Receiver Not Ready) 和 SREJ (Selective Reject)。

Frame type	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
I	SAR		ReqSeq						R	TxSeq						0
S	X	X	ReqSeq						R	X	X	X	S		0	1

X denotes reserved bits, which shall be set to 0.

Table 3.1: Standard Control Field formats

Frame type	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
I	SAR		ReqSeq						F	TxSeq						0
S	X	X	ReqSeq						F	X	X	P	S		0	1

X denotes reserved bits, which shall be set to 0.

Table 3.2: Enhanced Control Field formats

Frame type	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
I	ReqSeq														F	0
	TxSeq														SAR	
S	ReqSeq														F	1
									X	X	X	X	X	P	S	

X denotes reserved bits, which shall be set to 0.

Table 3.3: Extended Control Field formats

发送序列号: TxSeq, 6 位或 14 位

发送序列号用于计数每一个 I-frame, 以能序列化和重传。

接收序列号: ReqSeq, 6 位或 14 位

接收序列号由接收方使用, 以应答 I-frame, 并且在 REJ 和 SREJ 帧中使用特定的发送序列号请求 I-frame 重传。

重传禁止位: R 位, 1 位, 重传位用于实现流控。接收者在其内部接收缓冲区满时设置该位, 这种情况发生在一个或多个 I-frame 已经接收到位 SDU 重组功能还没有在所有接收到的帧中执行。当发送者接收到一个 R 位置 1 的的帧, 它将禁止重传定时器, 这将导致发送者停止重发 I-frame。

R=0: 正常模式。发送者使用重传定时器控制 I-frame 的重传。发送者不使用监控定时器。

R=1: 接收方请发送者重传 I-frame. 发送使用监控定时器监控信令。发送使用重传定时器。

ReqSeq 和 R 位的功能是独立的。

分割与重组, SAR, 2 位, SAR 位定义了 L2CAP SDU 是否被分割。对于分割的 SDU, SAR 位定义了该帧在 I-frame 中属于哪部分, 因此允许一个 L2CAP SDU 包含多个 I-frame。

SAR="Start of L2CAP SDU"的 I-frame 包含长度域, 在整个 L2CAP SDU 中特定数量的信息字节。SAR 的编码如表 3.4。

00	Unsegmented L2CAP SDU
01	Start of L2CAP SDU
10	End of L2CAP SDU
11	Continuation of L2CAP SDU

Table 3.4: SAR control element format.

管理功能, S 域, 2 位:

S 域标志了 S-frame 的类型。有四种类型定义 RR (Receiver Ready), REJ (Reject), RNR (Receiver Not Ready) 和 SREJ (Selective Reject), 编码如下:

00	RR - Receiver Ready
01	REJ - Reject
10	RNR - Receiver Not Ready
11	SREJ - Select Reject

Table 3.5: S control element format: type of S-frame.

Poll, P 位, 1 位, P 位设置为 1, 以向接收者请求一个应答。接收者应该使用 F 位为 1 的帧立即应答。

Final, F 位, 1 位, F 位 设置为 1, 以应答一个 P 位为 1 的 S-frame。

3.3.3 L2CAP SDU 长度域, 2 字节

当一个 SDU 分成多个 I-frame 时, 序列中的第一个 I-frame 必须被 SAR=0X01, 即"Start of L2CAP SDU"识别。L2CAP SDU 长度域必须决定在 SDU 中字节的总长度。L2CAP SDU 长度域必须存在在 I-frame 中, 其 SAR=01 (Start of L2CAP SDU), 且长度域不能存在于其它 I-frame 中。当 SDU 没有被分割, L2CAP SDU 长度域不需要, 且不应该存在。

3.3.4 信息负载域

信息负载域由整数字节数据组成。该域的最大字节数与商定的 MPS 配置参数相同。该域的最大字节数由受基本 L2CAP 头长度域的范围限制。这个范围从 65533 字节的带有标准或增强控制域, 无 SDU 长度域和 FCS 域的 I-frame 到 65527 字节的带有增强控制域, SDU 长度域和 FCS 域的 I-frame。因此, 即使商定 MPS 为 65533 字节, 基本 L2CAP 头长度域也将限制字节数。比如, 当一个增强控制域, 一个 SDU 长度域和 FCS 域存在, 该域的字节数将被限制为 65529。

3.3.5 帧校验序列, 2 字节

FCS 序列是 2 字节。FCS 使用如下公式生成:

$$g(D) = D^{16} + D^{15} + D^2 + 1$$

其中 16 位 LFSR 初始值为 0x0000, 如图 3.5 所示。开关 S 设置到位置 1, 则数据移入, 每一个字节以低位移入先。当最后一位进入到 LFSR, 开始 S 设置到位置 2, 且寄存器内容从右到左传输。FCS 桥头基本 L2CAP 头, 控制域, L2CAP SDU 长度和信息域, 如 46 页图 3.3 所示。

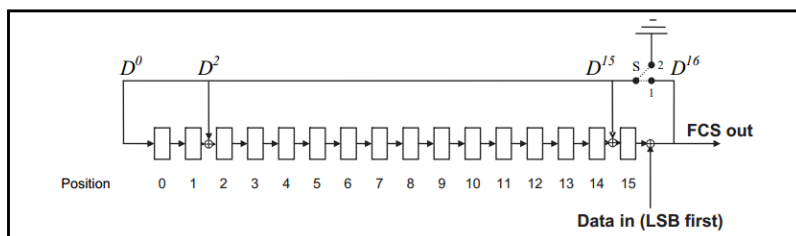


Figure 3.4: The LFSR circuit generating the FCS.

[illegible]

Figure 3.5: Initial state of the FCS generating circuit.

FCS 计算示例:

$$q(D) = D^{16} + D^{15} + D^2 + 1$$

1. I Frame

Length = 14

Control = 0x0002 (SAR=0, ReqSeq=0, R=0, TxSeq=1)

Information Payload = 00 01 02 03 04 05 06 07 08 09 (10 octets, hexadecimal notation)

==> FCS = 0x6138

==> Data to Send = 0E 00 40 00 02 00 00 01 02 03 04 05 06 07 08 09 38 61
(hexadecimal notation)

2. *RR Frame*

Length = 4

Control = 0x0101 (ReqSeq=1, R=0, S=0)

==> FCS = 0x14D4

==> Data to Send = 04 00 40 00 01 01 D4 14 (hexadecimal notation)

3.3.6 不可用帧检测

对于重传模式和流控模式，接收到的 PDU 必须忽略不可用包，以下是出现不可用包的情况：

- 1、包含未知的 CDI
- 2、包含 FCS 错误
- 3、包含长度值大于 MPS
- 4、I-frame 长度小于 8 字节
- 5、Iframe 的 SAR=01 (Startof L2CAP SDU)，但其长度小于 10 字节
- 6、I-frame 的 SAR 位值不是正常序列值或分割起始值或分割连续值或分割结束值。
- 7、S-frame 长度域不为 4

出错条件可能用于出错报告。

3.3.7 不可用帧检测算法

对于增强重传模式和数据流模式，下面的算法将用于 PDU 接收。它可能用于重传模式和流控模式。

- 1、检查 CDI，如果 PDU 包含未知的 CID,则忽略该帧。
- 2、检查 FCS。如果 PDU 包含一个 FCS 错误，则忽略该帧。如果管道配置为不使用 FCS，则 PDU 认为包含一个正确的 FCS。
- 3、检查如下条件，如果其中一个条件发生，管道将关闭或在固定管道的情况下，ACL 将失去连接。
 - a、PDU 包含长度大于 MPS 值。
 - b、I-frame 长度小于目标值。如果管道配置为使用标准或增强控制域，则要求的字节数为 6，此时 FCS 没有使用，否则，字节数为 8. 如果管道配置为使用扩展控制域，则要求的字节数为 8，这时 FCS 没有使用，否则，字节数为 10.
 - c、长度域可用的 S-frame. 如果管道配置为使用标准或增强控制域，则长度为 2，此时 FCS 没有使用，否则，字节数为 4. 如果管道配置为使用扩展控制域。则长度为 4，此时 FCS 域没有使用，否则，长度为 6。
- 4、检查 SAR 位。SAR 位检查在帧已经被正常接收后执行。下面情况之一发生时，PDU 不可用：
 - a、SAR=01 的 I-frame 长度小于目标长度。如果管道配置为使用标准或增强控制域，则要求的字节数为 8，此时 FCS 没有使用，否则，字节数为 10.如果管道配置为使用扩展控制域，则要求的字节数为 10，此时 FCS 没有使用，否则，字节数为 12.
 - b、I-frame 的 SAR 值不为正常序列，分割的起始帧，连续帧，结束帧之一的。
 - c、SAR=01 的 I-frame, 其 L2CAP SDU 长度域超出配置的 MTU 值。
- 5、如果 I-frame 以正确的序列接收且如 4 所描述的可用，则管道应该关闭或在固定管道情况下，ACL 应该失去连接。对于数据流模式和流控模式，如果一个或多个使用 SAR 位为开始，连续和结束的 I-frame 丢失，则序列中接收到的 I-frame 将被忽略。对于流控模式和数据流模式，未分割的超出序列带有 SAR 位的 I-frame 接收将被接受。

如果算法用于重传模式或流控模式，则必须使用 3.3.6 描述的帧检测。

这些错误条件可能用于出错报告。

4、信令包格式

本节描述两个设备中的 L2CAP 实例信令命令传输。所有信令命令通过信令管道发送。通过 ACL-U 逻辑链路管理管道的信令管道必须使用 CID 为 0x0001, 而通过 LE-U 逻辑链路管理管道的信令管道必须使用 CID 为 0x0005。一旦底层逻辑传输建立且 L2CAP 通信使能，信令管道即可用。54 页图 4.1 描述了包含信令命令的 L2CAP PDU 格式。多个命令可能通过固定管道 CID 为 0x0001 的单独 C-frame 发送，而每一个带有单独命令的 C-frame 将通过固定管道 CID 为 0x0005 发送。命令决定了请求和应答的形式。所有 L2CAP 实现必须支持带有不超过 MTU 负载长度的 C-frame 的接收。C-frame 支持的最小负载长度由 54 页表 4.1 定义。L2CAP 实现不能使用超出 MTU_SIG 的 C-frame. 如果设备接收到超出 MTU_SIG 的 C-frame, 则它将发慈禧太后一个包含可支持的 MTU_SIG 拒绝命令。实现必须能够处理通过 CID 为 0x0001 的固定管道发送的 L2CAP 包中多个命令的接收。

Logical Link	Minimum Supported Payload Length for the C-frame (MTU _{sig})
ACL-U not supporting Extended Flow Specification	48 octets
ACL-U supporting the Extended Flow Specification feature	672 octets
LE-U	23 octets

Table 4.1: Minimum Signaling MTU

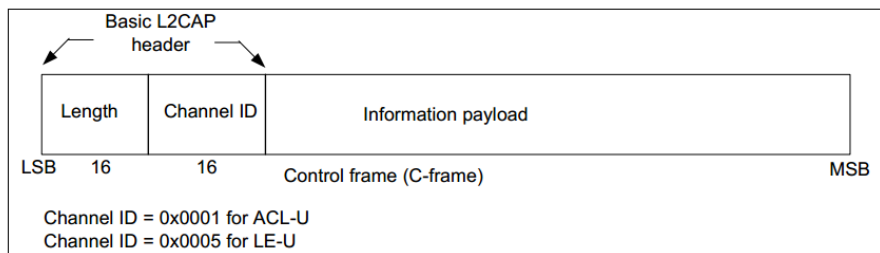


Figure 4.1: L2CAP PDU format on a signaling channel

图 4.2 描述了所有信令命令的一般格式：

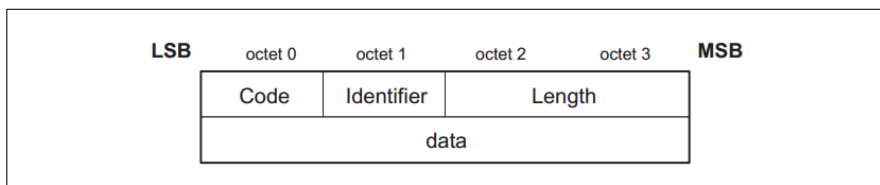
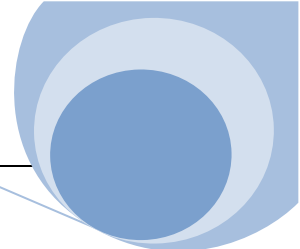


Figure 4.2: Command format

相关域描述如下：

Code，1 字节，**Code** 域为一字节长，并识别命令的类型。当 **Code** 域未知或禁止的数据包被接收，一个拒绝命令数据包将被发送以做为做答。

55 页的表 4.2 列出了本文档定义的代码。



Code	Description	CIDs on which Code is Allowed
0x00	Reserved	Any
0x01	Command reject	0x0001 and 0x0005
0x02	Connection request	0x0001
0x03	Connection response	0x0001
0x04	Configure request	0x0001
0x05	Configure response	0x0001
0x06	Disconnection request	0x0001
0x07	Disconnection response	0x0001
0x08	Echo request	0x0001
0x09	Echo response	0x0001
0x0A	Information request	0x0001
0x0B	Information response	0x0001
0x0C	Create Channel request	0x0001
0x0D	Create Channel response	0x0001
0x0E	Move Channel request	0x0001
0x0F	Move Channel response	0x0001
0x10	Move Channel Confirmation	0x0001
0x11	Move Channel Confirmation response	0x0001

Table 4.2: Signaling Command Codes

Code	Description	CIDs on which Code is Allowed
0x12	Connection Parameter Update request	0x0005
0x13	Connection Parameter Update response	0x0005

Table 4.2: Signaling Command Codes

Identifier, 1 字节, 识别域为一个字节长度, 匹配应答和请求。请求设备设置该域而应答设备在它的应答中使用相同的值。在任意两个设备间, 不同的识别使用在每一个成功的命令中。下面是一个命令中的识别原始传输, 识别可能被回收, 如果所有其它的随后的识别被使用。

RTX 和 ERTX 定时器用于决定什么时候远程端点不应答信令请求。在 RTX 或 ERTX 溢出时, 相同的识别号将被使用, 如果一个多重请求重发。如 110 页, 6.1.7 章节所强调。

设备接收一个多重请求应该使用一个多重应答回应。带有无效识别号的命令应答默认丢弃。信令识别 0x00 是非法识别号, 且必须不能在任何命令中使用。

长度, 2 字节, 长度域为两字节长度, 且指示了命令中的数据域的长度。它不包括 Code, 识别域及长度域。

数据, 0 或更多字节, 数据域由长度决定可用多少。Code 域决定数据域的格式, 长度域决定数据域的长度。

4.1 拒绝命令: Code = 0x01

拒绝命令数据包必须在应答未知命令代码命令数据包或当发送相应的应答不相符时发送。57 页图 4.3 显示了数据包格式。标识符必须匹配拒绝命令数据包的标识符。实例总是发送这些数据包应答未识别的信令包。拒绝命令不应该发送以应答一个未识别的应答数据包。

当多重命令包括在一个 L2CAP 数据包中且数据包超出接收者的信令 MTU，在应答中一个单独的拒绝命令将被发送。标识符必须与 L2CAP 数据包中的第一个请求命令相同。如果只有应答被识别，数据包默认被丢弃。

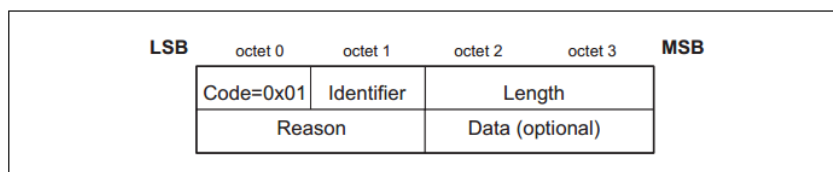


Figure 4.3: Command Reject packet

图 4.3 显示了拒绝命令数据包的数据格式，数据域为：

Reason, 2 字节，Reason 域描述了请求包为什么被拒绝，且设置为表 4.3 中的一个原因代码。

Reason value	Description
0x0000	Command not understood
0x0001	Signaling MTU exceeded
0x0002	Invalid CID in request
Other	Reserved

Table 4.3: Reason Code Descriptions

Data, 0 或多个字节，数据域的长度和内容独立于 Reason 代码。如果 Reason 代码为 0x0000，则为命令不能识别，数据域没有内容。如果 Reason 代码为 0x0001，则为信令 MTU 超出，此时 2 字节数据域代表数据包发送者能接受的最大的信令 MTU。

如果命令涉及一个不可用的管道，则 Reason 代码 0x0002 将被返回。典型的，一个管道不可用是因为它不存在。数据域为 4 字节包含了本地和远端管道的端点。远端端点是从拒绝命令而来的源 CID，本地端点是来自拒绝命令的目标 CID。如果拒绝命令只包含管道端点中的一个，另外一个将被空 CID 0x0000 代替。

Reason value	Data Length	Data value
0x0000	0 octets	N/A
0x0001	2 octets	Actual MTU _{sig}
0x0002	4 octets	Requested CID

Table 4.4: Reason Data values

4.2 连接请求 Code = 0x02

连接请求包被发送以在两个设备中创建一个 L2CAP 管道。L2CAP 管道必须建立在配置开始之前。图 4.4 描述了连接请求数据包：

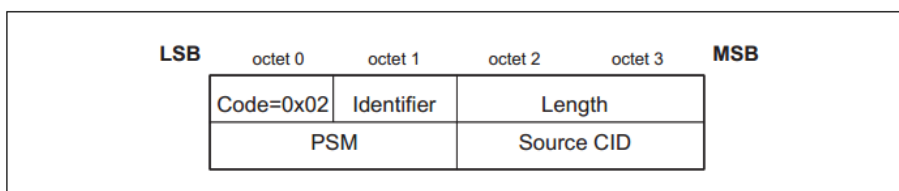


Figure 4.4: Connection Request Packet

相关数据域如下：

PSM，协议服务复用器，PSM 域至少两个字节长度。PSM 域的结构基于地址域的 ISO3309 扩展机制。所有 PSM 值必须是 ODD，也就是最低有效字节的最低有效位必须为 1。即而所有的 PSM 值的最高有效字节的最低有效位等于 0。这允许 PSM 域可以扩展超出 6 位。PSM 值分成两个部分。每一部分的有效部分由蓝牙标准技术联盟安排，且指示出协议。第二部分的值可动态分配，并在服务发现协议（SDP）节点中使用。动态分配值可能用于支持实现协议的多重实现。

PSM 值由 Assigned Numbers 文档定义：

Range	Type	Server Usage	Client Usage
0x0001-0x0EFF (Note 1)	Fixed, SIG assigned	PSM is fixed for all implementations.	PS may be obtained via SDP or may be assumed for a fixed service. Protocol used is indicated by the PSM as defined in the Assigned Numbers page.
>0x1000	Dynamic	PSM may be fixed for a given implementation or may be assigned at the time the service is registered in SDP.	PSM shall be obtained via SDP upon every reconnection. PSM for one direction will typically be different from the other direction

1. PSMs shall be odd and the least significant bit of the most significant byte shall be zero, hence the following ranges do not contain valid PSMs: 0x0100-0x01FF, 0x0300-0x03FF, 0x0500-0x05FF, 0x0700-0x07FF, 0x0900-0x09FF, 0x0B00-0x0BFF, 0x0D00-0x0DFF, 0x0F00-0x0FFF. All even values are also not valid as PSMs.

源 CID-SCID，2 字节，源 CID 为两个字节长度 并在发送请求的设备端表示一个管道端点。一旦管道被配置，流向请求的发送者的数据包必须发送到该 CID。因此，源 CID 代表发送请求和接收应答的设备的管道端点。

4.3 连接应答 Code = 0x03

当设备接收到一个连接请求数据包，它将发送一个连接应答数据包。连接应答数据包格式如图 4.5 所示：

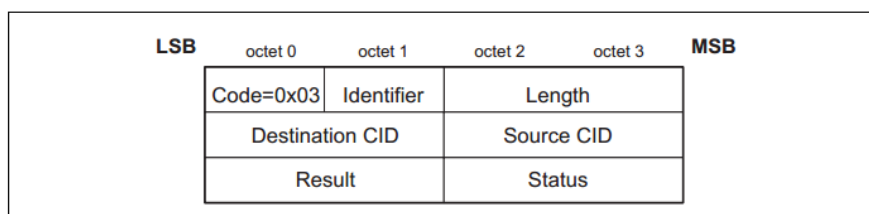


Figure 4.5: Connection Response Packet

相关数据域如下：

目标 CID，DCID，2 字节，该域包括发送该应答数据包设备的管道端点。因此，目标 CID 代表接收请求和发送应答的设备的管道端点。

源 CID，SCID，2 字节，该域包含接收应答数据包设备的管道端点。这是从连接请求数据包中的 SCID 域中拷贝而来。

结果，2 字节，结果域指示连接请求的输出。结果值为 0x0000 标志成功，而非 0 值标志连接请求失败或挂起。逻辑管道建立在成功的结果被接收到。59 页表 4.5 定义了该域的值。当结果域标志连接被拒绝，DCID 和 SCID 域必须忽略。

Value	Description
0x0000	Connection successful.
0x0001	Connection pending

0x0002	Connection refused – PSM not supported.
0x0003	Connection refused – security block.
0x0004	Connection refused – no resources available.
Other	Reserved.

Table 4.5: Result values

状态，2 字节，只为结果=挂起而定义。标志连接的状态。状态域设置为以下的值之一：

Value	Description
0x0000	No further information available
0x0001	Authentication pending
0x0002	Authorization pending
Other	Reserved

Table 4.6: Status values

4.4 配置请求 Code = 0x04

配置请求数据包被发送以在两个 L2CAP 实例间建立一个初始化逻辑链路传输协议，且重新商议这种协议是否合适。协议由 79 页第 5 章定义的配置参数选项集组成。所有参数选项有缺省的值并可以有先前商定的值，该先前商定的值在前一个配置流程中或在当前配置流程中的前一步被接受。必须被包括在配置请求数据包中的参数是那些与默认值或先前的值不同的值。

如果没有参数需要被商议或确认，则没有选项将被插入，且连续标志（C）必须设置为 0。任何遗漏的配置参数被假定最近显式或隐式接受的值。即使所有缺省的值是可接受的，一个没有选项的配置请求包必须被发送。隐性接收的值是配置参数的缺省值，这个配置参数还没有在配置下显式协定确定管道。参考 120 页 7.1 章节有关配置流程的详细描述。

图 4.6 定义了配置请求数据包的格式：

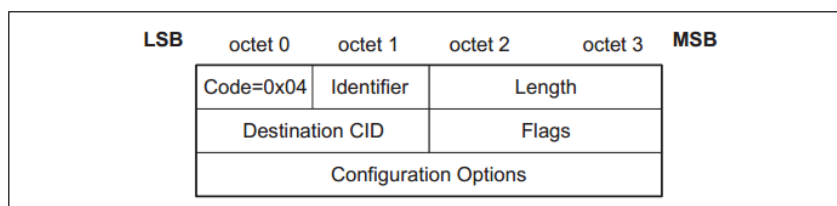


Figure 4.6: Configuration Request Packet

相关数据域如下：

目标 CID，DCID，2 字节，该域包括接收该请求数据包的设备的管道端点。

标志，2 字节，图 4.7 显示了两字节标志域，注意最高有效位显示在左边。

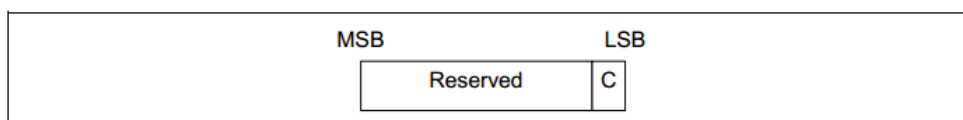


Figure 4.7: Configuration Request Flags field format

只有一个标志被定义，即连续标志（C）：

当 L2CAP 实例都支持扩展流确认选项，连续标志不能使用且必须在所有配置请求和应答数据包中设置为 0。

当所有配置参数不能适合带有不超出接收 MTU_SIG 的配置参数，选项将在多重配置命令数据包中传输。如果所有选项适合接收者的 MTU_SIG，则他们将被以连续标志设置为 0 的单独配置请求的方式发送。每一个配置请求必须包含一个整数个的选项，部分形式选项将不会被发送。每一个请求必须使用一个不同的标识符标志，且必须使用相同的标识符应答。

当使用在配置请求中，连续标志标志应答者应该期望接收接收到多重请求包。应答者应该回应每一个配置请求包。应答者可能使用包含请求中相同的选项的配置应答回应配置请求，或者应答者可能使用一个不包含选项的"Success"配置应答数据包回应。延时这些选项直到全部请求被接收。带有连续标志清除的配置请求数据包必须在管道状态机中做为配置请求事件。

当使用在配置请求应答中，如果连续标志在请求中被设置为 1，应答连续标志也必须被设置到 1。当匹配请求已经设置为 0 的应答中，如果连续标志被设置为 1，则它标志应答者有额外的选项以发送给请求者。在这种情况下，请求者应该发送空选项配置请求到应答者直到应答者使用连续标志设置为 0 的配置请求应答。连续标志设置为 0 的配置请求包必须做为管道状态机中的配置请求事件。

配置交易的结果是所有值的结果集合。所有的结果值必须是配置交易中成功的值。

其它标志预留必被设置为 0。L2CAP 实现必须忽略这些位。

配置选项：参数列表及其商定的值必须在配置选项域中提供。这些在 79 页章节 5 中定义。配置请求可能不包含选项（参考空配置请求或无效配置请求）且可能用于请求一个应答。对于空配置请求，长度域设置为 0X0004。

4.5 配置应答 Code = 0X05

配置应答包必须被发送以应答配置请求包，除非错误条件被拒绝命令应答覆盖。在配置应答中的每一个配置参数值（如果存在）反射在对应的配置请求中已经被发送的配置参数的一个“adjustment”。比如：如果配置请求从设备 A 到设备 B 的通信流，配置应答的发送者可能为相同的通信流调整这个值，但应答可能在相反的方向不调整这个值。

选项在应答中发送基于在结果域中的值，63 页的图 4.8 定义了配置应答包的格式。参考 120 页 7.1 章节以了解更多的配置过程。

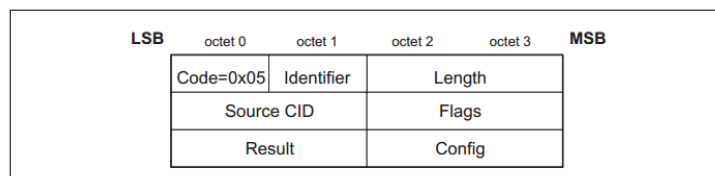


Figure 4.8: Configuration Response Packet

相关数据域如下：源 CID，SCID，2 字节，该域包含在设备上接收该应答包的管道端点。接收该应答的设备必须检查标识符域匹配对应配置请求命令的相同域，且 SCID 匹配其本地 CID 配上原 DCID。

标志：2 字节，图 4.9 显示了两字节标志域。注意最高有效位显示在左边。

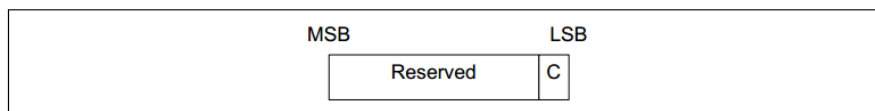


Figure 4.9: Configuration Response Flags field format

只有一个标志被定义，连续标志（C）。

当 L2CAP 实例都支持扩展流定义选项时，连续标志必须不能使用且应该设置为 0 在所有的配置请求和应答数据包中。

当 C 位设置为 1 时，更多的配置应答将紧跟。这个标志指示应答中的参数是发送应答包设备的参数集的一部分。其它的标志位预留必须测试为 0。L2CAP 实例必须忽略这些位。

结果，2 字节，结果域标志请求是否被接受，参考 63 页表 4.7 的相关编码：

Result	Description
0x0000	Success

0x0001	Failure – unacceptable parameters
0x0002	Failure – rejected (no reason provided)
0x0003	Failure – unknown options
0x0004	Pending
0x0005	Failure - flow spec rejected
Other	Reserved

Table 4.7: Configuration Response Result codes

配置选项：该域包含将被配置的参数列表。这些在 79 页章节 5 中定义。在一个成功结果（Result = 0x0000）和挂起结果（Result = 0x0004），这些参数包含任何参数值的返回结果（参考 82 页章节 5.3）和包含在请求中的非商定配置参数值的“adjustments”。结果代码为 0x0000 的应答也称为积极应答。

在一个不可接受的参数失败（Result = 0x0001）时，拒绝参数必须被发送在应答中其值已经接收如果发送在原始请求中。任何在配置请求中的丢失配置参数被假设有他们自己的缺省值或先前可靠的值，且他们应该包括在配置应答中，如果他们需要被改变。结果代码 0x0001 的应答也称为消极应答。

在未知选项失败（Result = 0x0003），包含不被接收到的请求理解的选项类型域的选项必须包含在应答中，除非他们是隐藏的。如果不能理解，在请求中将路过这些隐藏的选项。隐藏的选项将不包括在应答中，且不应该成为拒绝这个请求的唯一原因。

在流定义拒绝失败时（Result = 0x0005），一个扩展流定义选项可能被包含以影响可接受的 QoS 等级。参考 122 页 7.1.3 章节。

在中断商定之前，仲裁管道参数的用时标准独立定义。

4.6 断开连接请求

中断 L2CAP 管道请求需要一个断开连接请求被发送且由一个断开连接请求应答。65 页的图 4.10 显示了一个断开连接请求。接收者必须保证在初始化管道断开连接前 SCID 和 DCID 匹配。

一旦断开连接请求被执行，所有在 L2CAP 管道的输入数据必须被忽略且任何新的额外输出数据必须被忽略。一旦一个管道的断开连接请求被接收到，所有其管道上的将被发送出去的数据队列必须被忽略。

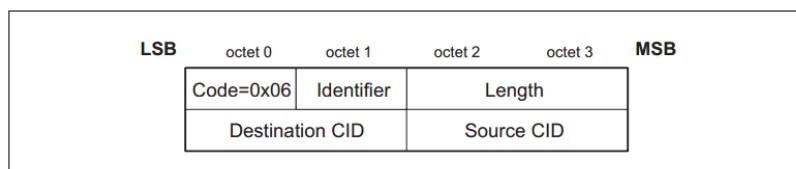


Figure 4.10: Disconnection Request Packet

相关数据域定义如下：

DCID，2 字节，该域定义了接收该请求的设备上的断开连接的管道端点。

SCID，2 字节，该域定义了发送该请求的设备上的断开连接的管道端点。

SCID 和 DCID 与请求的发送者有关，且必须匹配那些将被断开连接的管道。如果 DCID 不被该消息的接收者识别，带有不可用 CID 结果代码的命令拒绝消息将在应答中被发送。如果接收者发现 DCID 匹配但 SCID 不能找到相同的匹配，请求将被默认忽略。

4.7 断开连接应答 Code = 0x07

断开连接应答必须被发送以应答每一个可用的断开连接请求

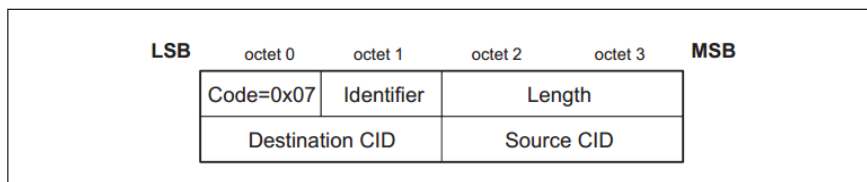


Figure 4.11: Disconnection Response Packet

相关数据域如下：

DCID，2 字节，该域标识发送该应答的设备上的管道端点。

SCID，2 字节，该域标志接收该应答的设备上的管道端点。

DCID 和 SCID 及标识符域必须匹配那些应答的断开连接请求命令。如果 CID 不匹配，在接收方，应答默认被忽略。

4.8 反射请求 Code = 0x08

反射请求用于从一个无端 L2CAP 实例中请求一个应答。这些请求可能用于 测试连接或使用可选数据域验证厂家特定信息。L2CAP 实例必须以反射应答包应答一个可用的反射请求包。数据域是可选的并单独定义。如果存在这个域，L2CAP 实例应该忽略这个域的内容。

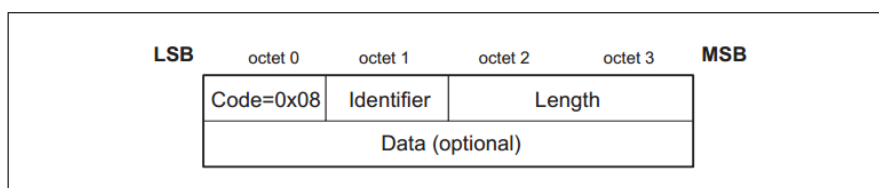


Figure 4.12: Echo Request Packet

4.9 反射应答 Code = 0x09

一个反射应答必须被发送，一旦接收到一个可用反射请求。应答中的标识符必须匹配在请求中的标识符。可选和单独定义的数据域可能包含请求中的数据域的内容，不同的数据或没有数据。

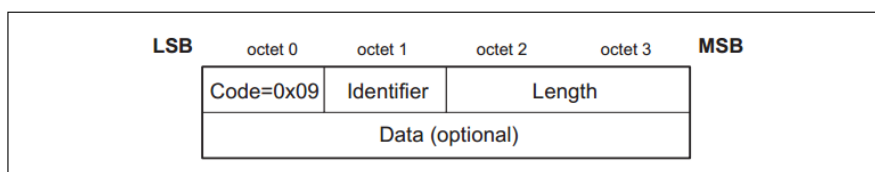


Figure 4.13: Echo Response Packet

4.10 信息请求 Code = 0x0a

信息请求用于从远端 L2CAP 实例中请求实例特定信息。L2CAP 实例必须应答用信息应答回应一个可用的信息请求。发送信息请求是可选的。

必须为远端 L2CAP 实例已经标志支持信息应答的 L2CAP 实例使用可选特征或属性范围。直到标志支持可靠特征或范围的信息应答已经被接收仅有的强制特征和范围必须被使用。

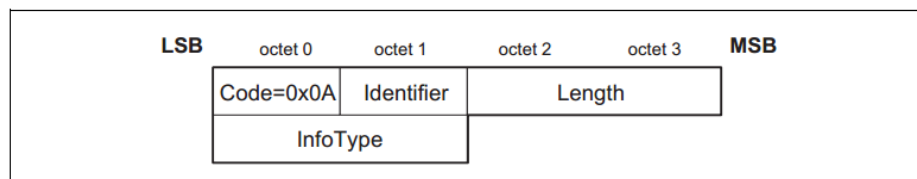


Figure 4.14: Information Request Packet

数据域定义如下：

信息类型，2 字节，信息类型定义了被请求特定信息实例的类型。参考 67 页 4.11 章节：

Value	Description
0x0001	Connectionless MTU
0x0002	Extended features supported
0x0003	Fixed Channels supported
Other	Reserved

Table 4.8: InfoType definitions

L2CAP 实例禁止在固定管道 CID=0X0001 上发送信息类型为 0X0003 的信息请求，直到第一个确定固定信道位在远端设备扩展特征掩码被设置。支持固定管道是强制的对于 BR/EDR/LE 或 LE 控制器设备。信息请求和信息应答禁止用在固定管道 CDI 为 0X0005 上。

4.11 信息应答 Code = 0X0B

一旦接收到一个可用信息请求帧，信息应答必须被发送。应答中的标识符必须匹配请求中的标识符。数据域必须包含信息类型域相关的值，或必须为空（如果信息类型不支持）。

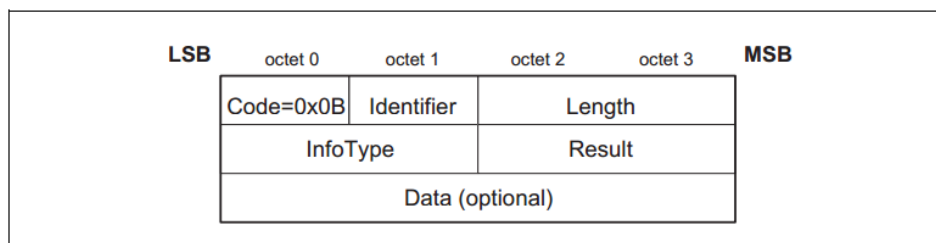


Figure 4.15: Information Response Packet

相关数据域如下，

信息类型：2 字节，信息类型定义被请求特定信息的实例类型。该值必须从信息请求中的信息类型域中复制。

结果：2 字节，结果包含关于请求成功的信息。如果结果成功，数据域包含 69 页表 4.10 定义的信息。如果结果是不支持，没有数据被返回！

Value	Description
0x0000	Success
0x0001	Not supported
Other	Reserved

Table 4.9: Information Response Result values

数据：0 或多个字节，数据域的内容由信息类型不同而不同。
对于信息类型为 0X0001，数据域包含远端实例的两字节可接受无连接 MTU，缺省值在 45 页，3.2 章节定义。

对于信息类型为 0X0002，数据域包含 4 字节 L2CAP 扩展特征掩码。特征掩码参考发送信息应答支持的 L2CAP 实例扩展特征。特征位包含在 69 页 4.12 定义的 L2CAP 特征掩码中。

对于信息类型为 0X0003，数据域包含一个 8 字节位映射，标志支持的固定 L2CAP 管道。可用固定管道列表在 2.1 章节的表 2.1 中提供。详细的有关信息类型的描述在 70 页 4.13 章节中结出。

注意：L2CAP 实例版本在版本 1.2 之前，为 L2CAP 特征发现接收一个信息类型为 0X0002 的信息请求，返回一个结果代码为不支持的信息应答。L2CAP 实例在版本 1.2 到 2.1+EDR 有一个全 0 扩展特征掩码可能返回一个结果代码为不支持的信息应答。

InfoType	Data	Data Length (octets)
0x0001	Connectionless MTU	2
0x0002	Extended feature mask	4
0x0003	Fixed Channels Supported	8

Table 4.10: Information Response Data fields

4.12 扩展特征掩码

在信息应答数据域中，特征代表为位掩码，参考 67 页 4.11 章节。对于每一个特征安排一个单独的位与之对应。如果设置为 1，该特征被支持，如果设置为 0，该特征不被支持。所有未知，预留或未安排的特征位必须设置为 0。

69 页表 4.11 显示的特征掩码由四个字节组成，每一个位编码为 0-7。在信息应答包数据域内，字节 0 的位 0 安排在最左边。字节 3 的位 7 安排在最右边。

注意：L2CAP 特征掩码在蓝牙 V1.2 中介绍，且在蓝牙 V1.1 之后包含特征介绍。

No.	Supported feature	Octet	Bit
0	Flow control mode	0	0
1	Retransmission mode	0	1
2	Bi-directional QoS ¹	0	2
3	Enhanced Retransmission Mode	0	3
4	Streaming Mode	0	4
5	FCS Option	0	5
6	Extended Flow Specification for BR/EDR	0	6
7	Fixed Channels	0	7
8	Extended Window Size	1	0
9	Unicast Connectionless Data Reception	1	1
31	Reserved for feature mask extension	3	7

Table 4.11: Extended feature mask.

1. Peer side supports upper layer control of the Link Manager's Bi-directional QoS, see [Section 5.3 on page 82](#) for more details.

4.13 固定管道支持

每一个固定管道由 8 字节位掩码中的一个单独位代表。如果 L2CAP 实例支持在该信道支持信令，信息相关位必须被设置为 1。L2CAP 信令管道是强制的且因此管道相关位必须设置为 1。表 4.12 显示了位掩码的格式。

CID	Fixed Channel	Value	Octet	Bit
0x0000	Null identifier	Shall be set to 0	0	0
0x0001	L2CAP Signaling channel	Shall be set to 1	0	1
0x0002	Connectionless reception	0 – if not supported 1 – if supported	0	2
0x0003	AMP Manager Protocol Channel	0 – if not supported 1 – if supported	0	3
0x0004 - 0x003E	Reserved	Shall be set to 0 and ignored upon receipt.	0 1-6 7	4-7 0-7 0-6
0x003F	AMP Test Manager	0 – if not supported 1 – if supported	7	7

Table 4.12: Fixed Channels Supported bit mask

一个 L2CAP 实例禁止在固定管道上传输，直到他接收到一个固定管道支持的信息类型从 L2CAP 实例标志管道支持，或已经接收到一个可用信令包从远端设备在固定管道。在不支持的固定管道上接收到的所有包必须被忽略。

4.14 创建管道请求 Code = 0x0c

创建管道请求包，如果图 4.16 所示，被发送通过控制器用控制器 ID 识别以在两个设备间创建一个 L2CAP 管道。

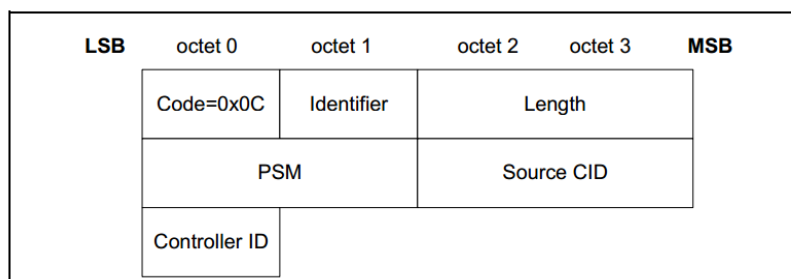


Figure 4.16: Create Channel Request Packet

相关数据域定义如下：

PSM: 协议服务利用器，最小 2 字节。PSM 域结构是基于 ISO 3309 地址扩展域机制。所有 PSM 值都应该是奇数，也就是说有效字节的最低有效位必须为 1。同样，所有 PSM 值最高有效字节的最低有效位应该等于 0。这样允许 PSB 域可以扩展到 16 位。PSM 值分为两个范围，第一个范围由蓝牙 SIG 分配并指示协议。第二个范围值是动态分配的，并且与服务发现协议相关联（SDP）。动态分配可能用于支持多种指定协议的实现。PSM 值在 Assigned Numbers 文档中定义。

SCID: 源 CID，2 字节。源 CID 为两字节长度，并代表发送请求的设备的一个信道端点。一旦信道被配置，紧随发送者的请求数据包将被发送到这个 CID。因此，源 CID 代表了设备发送请求和接收应答的端点。

控制标识符: 控制 ID，1 字节，代表了通过哪一个信道将创建控制器物理连接。控制 ID 是远端设备的一个标识符，并通过 AMP 管理发现可用 AMPs 请求(AMP Manager Discover Available AMPs request)获得。参考 PARTE，AMP Manager Protocol 规范第 439 页，控制器物理连接在这里详细说明。

4.15 创建信道应答 Code = 0x0d

当设备接收到一个创建管道请求包，它将发送一个创建管道应答包。创建管道应答包格式如图 4.17 所示。

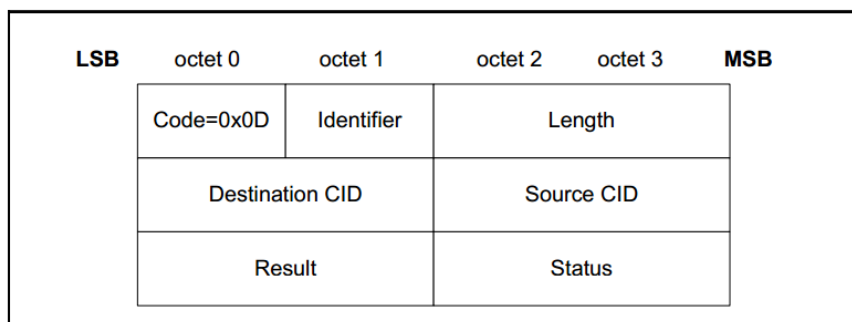


Figure 4.17: Create Channel Response Packet

相关数据域如下：

DCID: 2 字节，该域包含了设备上发送这个应答包的信道端点。因此，目的 CID 代表接收请求和发送应答的端点。

SCID: 2 字节，该域包含了设备上接收这个应答包的信道端点。这个值是创建信道请求数据包中 SCID 域的备份。

结果: 2 字节，结果域标志了创建管道请求的输出。结果值为 0x0000 标志成功，而非 0 值表示创建管道请求失败或挂起。一旦成功结果被接收，逻辑管道即建立。表 4.13 定义了该域的值，DCID 和 SCID 域必须被忽略，当结果域标志连接被拒绝。

Result	Description
0x0000	Connection successful
0x0001	Connection pending
0x0002	Connection refused - PSM not supported
0x0003	Connection refused - security block
0x0004	Connection refused - no resources available
0x0005	Connection refused - Controller ID not supported
Other	Reserved

Table 4.13: Result values

状态，2 字节，只定义结果为挂起时。标志连接状态，状态值为表 4.14 之一：

Value	Description
0x0000	No further information available
0x0001	Authentication pending
0x0002	Authorization pending
Other	Reserved

Table 4.14: Status values

4.16 移动管道请求 **cdoe = 0X0E3**

移动管道请求包被发送以移动一个存在的 L2CAP 管道从一个控制器上的物理链路到另外一个控制器的物理链路连接。L2CAP 的 CID 不被移动操作改变。图 4.18 描述了一个移动管道请求包。如果配置使用增强重传模式或数据流模式，管道必须被移动。固定管道不能被移动。

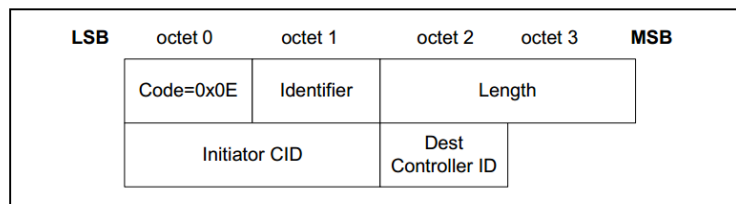


Figure 4.18: Move Channel Request Packet

相关数据域如下：

初始华管道标识符——ICID，2 字节，该域包含发送该请求包的设备上的管道端点。

目的控制器标识符——目的控制器 ID，1 字节，目的控制器 ID 是一个字节长度并代表必须移动的控制器的物理链路。控制器 ID 是通过 AMP 管理发现可用 AMPs 请求获得的远端设备上的控制器标识符。参考：Part E, AMP Manager Protocol Specification on page 439。控制器物理链路必须已经存在。

4.17 移动管道应答 **Code = 0x0f**

当设备接收到一个移动管道请求包，它将发送一个移动管道应答包。如果设备已经发送他自己的移动管道请求，则发生冲突。其中一个请求将被拒绝而另外的将被处理。被拒绝的移动管道请求基于 7.7 章节的算法。

移动管道应答包如图 4.19 所示：

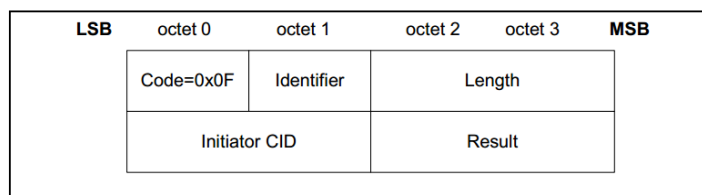


Figure 4.19: Move Channel Response Packet

初始化管道标识符——ICID：2 字节，该域包含发送移动管道请求设备上的管道端点。该值与移动管道请求数据包中的值相同。

结果：2 字节，结果域表示移动请求的输出。结果值为 0x0000 表示成功，而非 0 值表示移动请求失败。一旦成功的结果被接收到，逻辑管道移动成功。表 4.15 定义了该域的可能值。ICID 域必须被忽略，当结果域表示移动被拒绝。

Result	Description
0x0000	Move Success
0x0001	Move Pending
0x0002	Move refused - Controller ID not supported
0x0003	Move refused - new Controller ID is same as old Controller ID
0x0004	Move refused - Configuration not supported
0x0005	Move refused - Move Channel collision
0x0006	Move refused - Channel not allowed to be moved

Result	Description
Other	Reserved

Table 4.15: Result values

4.18 移动管道认证 Code = 0X10

当移动管道请求的初始化者从应答者接收到一个结果码不是挂起的移动管道应答，它将发送一个移动管道确认包，这推断移动管道过程，通知应答者，所有的 QoS 参数是否成功商定。移动管道确认包格式如图 4.20 所示：

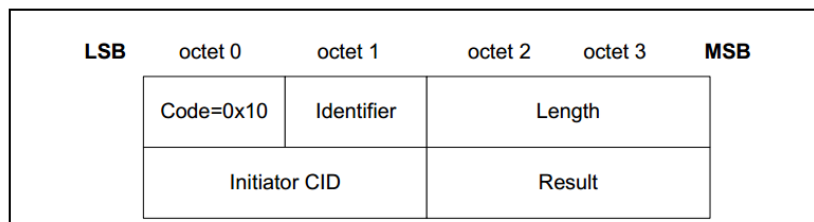


Figure 4.20: Move Channel Confirmation Packet

初始化管道标识符——ICID：2 字节，该域包含发送移动管道请求设备上的管道端点。该值与移动管道请求数据包中的值相同。

结果：结果域表示移动请求的输出。结果值为 0X0000 表示成功，而非 0 值表示移动请求失败。

Result	Description
0x0000	Move success - both sides succeed
0x0001	Move failure - one or both sides refuse
Other	Reserved

Table 4.16: Result values

4.19 移动管道确认请求 Code = 0x11

当设备接收到一个移动管道确认包，它将发送一个移动管道确认应答包。移动管道确认应答包的目的是保证移动管道确认和命令应答形式是一致的。移动管道确认应答包的格式如图 4.21 所示：

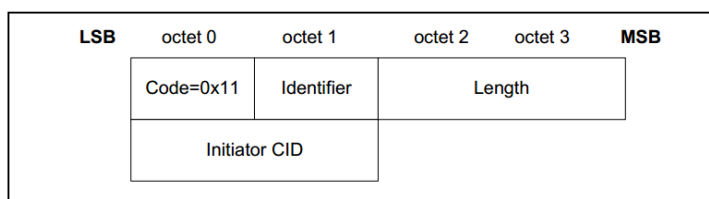


Figure 4.21: Move Channel Confirmation Response Packet

相关数据域如下：

初始化管道标识符——ICID：2 字节，该域包含发送移动管道确认包的设备上的管道端点。

4.20 连接参数更新请求 Code = 0x12

该命令由 LE 从机设备发送到 LE 主机设备。如果 LE 从机 HOST 接收到连接参数更新请求包，它必须用 reason 值为 0X0000（命令不能解析）的命令拒绝包应答。

连接参数更新请求允许 LE 从机 HOST 请求一个新的连接参数集合。当 LE 主机 HOST 接收到一个连接参数更新请求，根据其它连接参数，LE 主机 HOST 可能接受请求参数并传输请求参数到其控制器或拒绝请求。在支持 HCI 的设备中，LE 主机 HOST 使用 HCI_LE_Connection_Update 命令传输请求参数到其控制器(see [Vol. 2] Part E, Section 7.8.18)。如果 LE 主机 HOST 接受请求参数，它将发送一个结果值为 0X0000(参数被接受)连接参数更新应答包，否则它将发送结果值为 0X0001(请求被拒绝)的应答包。

LE 从机 HOST 将从 LE 从机控制器接收标志，当连接参数已经更新。在支持 HCI 的设备中，这通过 LE 连接更新完成事件的形式通知。(see [Vol. 2] Part E, Section 7.7.65.1)。如果 LE 主机控制器拒绝更新连接参数，从 LE 从机控制器将不会发送标志到 LE 从机 HOST。

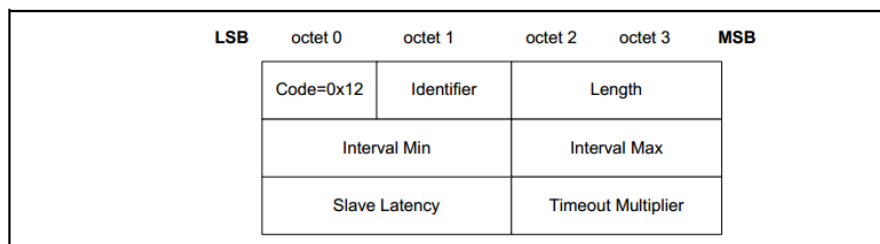


Figure 4.22: Connection Parameters Update Request packet

相关数据域如下：

Interval Min: 2 字节，以下面的方式定义最小连接事件间隔的值：

$\text{connIntervalMin} = \text{Interval Min} * 1.25 \text{ ms}$ 。间隔最小值范围：6 到 3200 帧，其中一帧为 1.25ms，即等于 2 个 BR/EDR 时隙。超出该范围的值为预留。最小间隔必须小于等于最大间隔。

Interval Max: 2 字节，以下面的方式定义最大连接事件间隔的值：

$\text{connIntervalMax} = \text{Interval Max} * 1.25 \text{ ms}$ 。间隔最大值范围：6 到 3200 帧，其中一帧为 1.25ms，即等于 2 个 BR/EDR 时隙。超出该范围的值为预留。最大间隔必须大于等于最小间隔。

从机等待：2 字节，以下面的方式定义人机等待参数：

$\text{connSlaveLatency} = \text{Slave Latency}$ ，人机等待域必须是 0 到 $((\text{connSupervisionTimeout} / \text{connIntervalMax}) - 1)$ 的一个值，从机等待延时域必须小于 500。

超时复用器：2 字节，以下面的方式定义连接超时参数：

$\text{connSupervisionTimeout} = \text{Timeout Multiplier} * 10 \text{ ms}$

超时复用器必须是 10-3200 之间的一个值。

4.21 连接参数更新应答 Code = 0x13

这个应答只能从 LE 主机设备发送到 LE 从机设备。

连接参数更新请求包必须由主机 HOST 发送当它接收到一个连接参数更新请求包时。如果 LE 主机 HOST 接收这个请求，它必须发送连接参数更新给其控制器。

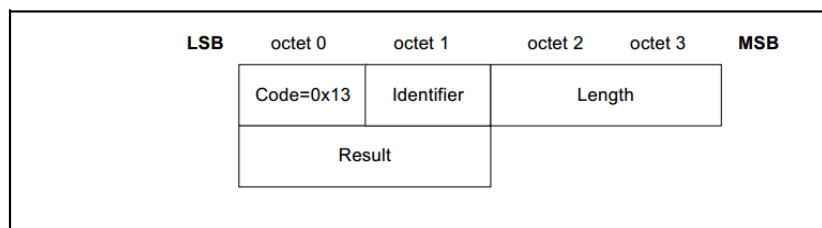


Figure 4.23: Connection Parameters Update Response packet

相关数据域如下：

结果：2 字节，结果域标志了连接参数更新请求应答的。数值 0x0000 标志了 LE 主机 HOST 已经接收了连接参数，而 0x0001 标志 LE 主机 HOST 拒绝连接参数。

Value	Description
0x0000	Connection Parameters accepted
0x0001	Connection Parameters rejected
Other	Reserved

5、配置参数选项

选项是扩展配置参数的机制。选项必须做为包含选项类型，选项长度，一个或多个选项数据域的信息元素传输。图 5.1 描述了选项的格式：

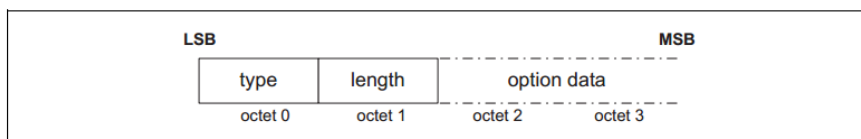


Figure 5.1: Configuration option format

相关数据域如下：

类型：1 字节，选项类型域定义了被配置的参数。类型的最高位定义了选项被识别时的动作。

0-选项必须被识别，如果选项没有被识别，由拒绝配置请求。

1-选项隐藏，如果选项没有被识别，则路过该选项并继续处理。

长度：1 字节，长度域定义了选项数据的字节长度。因此，一个没有选项数据的选项类型长度为 0。

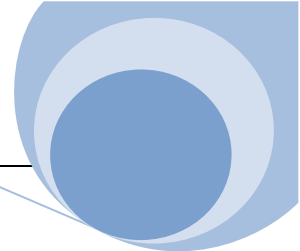
选项数据：该域的内容因选项类型而不同。

5.1 最大传输单元，MTU

该选项定义了最大 SDU 大小，该选项的发送者有能力接收一个信道。其类型为 0x01，且负载长度为 2 字节，内容为两字节的 MTU 大小值做为仅有的信息元素(see Figure 5.2 on page 81)。不像 B-frame 长度域，I-frame 长度域可能大于配置的 MTU，因为他包括了控制，L2CAP SDU 长度和 FCS 域的长度，当然也包括信息字节。

MTU 不是一个商定的值，它是每一个调协可以单独定义的信息参数。它向远端设备标志本地设备可以在管道接收一个大于最小需求的 MTU。所有 L2CAP 实例必须支持 ACL-U 逻辑链路上最小 48 字节的 MTU 和 LE-U 逻辑链路上最小 23 字节的 MTU。然而，一些协议和 profile 明确需要支持更大的 MTU。管道最小 MTU 大于 L2CAP 最小 48 字节 MTU 和任何协议的 Profile 要求的 MTU。(注意：MTU 只能被直接使用管道的 profile 影响。比如，如果一个服务发现传输由一个无服务发现 profile 初始化，profile 不影响服务发现服务的 L2CAP 管道的 MTU。)

未完，待续。。。。。



PARTC – GAP(GENERIC ACCESS PROFILE)

1 介绍

1.1 概述

GAP 的目标如下：

介绍传输和应用 profile 模式和访问过程相关的定义，推荐规范和一般性需求。

描述设备在 standby 和 connecting 状态如何运行，以保证链路和信道在两个设备之间是确定的，并且可以实现多 profile 的运行。特别专注于发现，连接确定及加密过程。

为了强调用户接口方面的需求，主要代码方案，过程及参数的名字需要保证良好的用户体验。

文档定义了三种设备类型，如 [Vol. 0], Part B Section 3.1 所述，这三种设备类型如下：

Device Type	Description
BR/EDR	Devices that support the “Basic Rate” Core Configuration (see [Vol. 0], Part B Section 4.1)
LE only	Devices that support the “Low Energy” Core Configuration (see [Vol. 0], Part B Section 4.4)
BR/EDR/LE	Devices that support the “Basic Rate and Low Energy Combined” Core Configuration (see [Vol. 0], Part B Section 4.5)

Table 1.1: Device Types

仅支持 LE 的设备和 BR/EDR/LE 设备类型可以工作在 LE 物理信道。

设备类型说明如下：

Device Types	Sections	Support
BR/EDR	1-8, 15-18	C1
LE-only	1-3, 9-12, 15-18	C1
BR/EDR/LE	1-18	C1
C1: Mandatory to support only one device type		

Table 1.2: Requirements for device types

1.2 符号和规约

1.2.1 必要的状态符号

在本文档，尤其是 profile 要求表中，下面的符号将被使用：

M，代表强制要求支持

O，代表可选择性支持

C，代表有条件性支持

E，代表排除在 profile 角色之外

N/A，代表空

在本标准中，shall 的使用表示这是强制要求，should 表示推荐要求，may 表示可选择。

1.2 信号框图约定

下面的箭头用于在框图中描述处理流程：

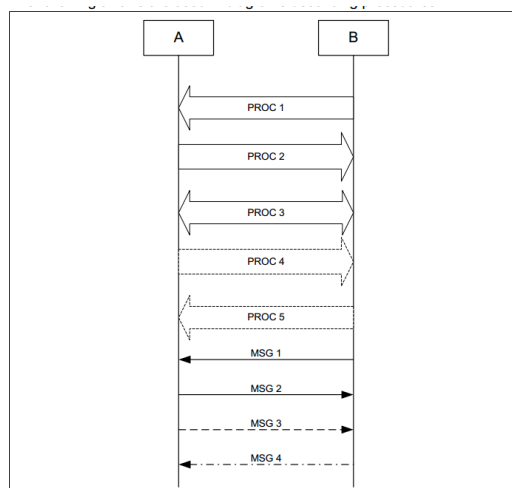


Figure 1.1: Arrows used in signaling diagrams

1.2.3 定时器和计数器符号

GAP 协议层有其对应的定时器，为了与 BLE 其它协议层的定时器区分，本协议内的定时器命名为 "Tgap(nnn)"

2、协议概览

2.1 协议栈

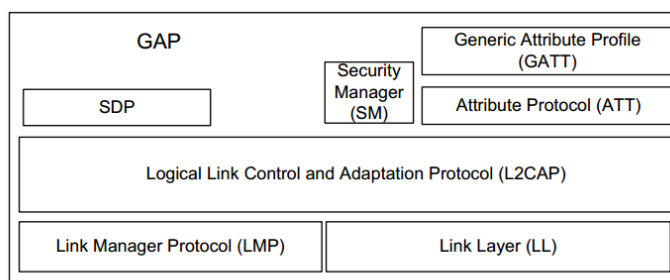


Figure 2.1: Relationship of GAP with lower layers of the Bluetooth architecture

本协议的意图是为了描述以下几点：

- 协议角色
- 搜索模式和处理流程
- 连接模式和处理流程
- 加密模式和处理流程

2.2 协议角色

2.2.1 工作在 BR/EDR 物理层时的角色

与 BLE 无关，跳过。

2.2.2 工作在 LE 物理信道的角色

工作在 LE 物理信道的设备有 4 种 GAP 角色

- 1、广播角色(Broadcaster)
- 2、观察角色(Observer)
- 3、外设角色(Peripheral)
- 4、主设角色(Central)

2.2.2.1 Broadcaster 角色

如 [Vol. 6], Part B Section 4.4.2 所定义的发送广播事件的设备工作在 Broadcaster 角色, 可称为广播者, 该广播者必须有发送能力, 某些条件下有接收能力。

2.2.2.2 Observer 角色

如 [Vol. 6], Part B Section 4.4.3 所定义的接收广播事件的设备工作在 Observer 角色, 可称为观察者, 该观察者必须有接收能力, 某些条件下有发送能力。

2.2.2.3 Peripheral 角色

任何能接收 Section 13.3 定义的 LE 物理链路层连接流程处理机制的设备工作在 Peripheral 角色。工作在 Peripheral 的设备做为 [Vol. 6], Part B Section 4.5 描述的链路层连接状态中的从机。该设备可称为外设或从机, 必须具有接收和发送的能力。

2.2.2.4 Central 角色

支持 Central 角色的设备可以初始化物理连接机制。工作在 Central 角色的设备做为 [Vol. 6], Part B Section 4.5 描述的链路层连接状态中的主机。主机必须上具有接收和发送的能力。

2.2.2.5 同时工作在多个 GAP 角色

如果控制器支持, 一个设备可以同时工作在多个 GAP 角色。建议 Host 在处理流程和模式使用前, 读取控制器所支持的链路层状态和状态组合。

只有在 [Vol. 6], Part B Section 1.1.1 中支持和允许的链路层状态和状态组合可以被使用。每一个 GAP 角色的物理层和链路层功能如下表显示:

M:强制支持

O:可选择性支持

C:满足条件才支持

E:除非在外, 不支持

	GAP Roles when Operating over an LE Physical Channel			
	Broadcaster	Observer	Peripheral	Central
Physical Layer functionality				
• Transmitter Characteristics	M	O	M	M
• Receiver Characteristics	O	M	M	M
Link Layer functionality				
States:				
• Standby State	M	M	M	M
• Advertising State	M	E	M	E
• Scanning State	E	M	E	M
• Initiating State	E	E	E	M
Connection State:				

• Slave Role	E	E	M	E
• Master Role	E	E	E	M
Advertising event types:				
• Connectable undirected event	E	E	M	E
• Connectable directed event	E	E	O	E
• Non-connectable undirected event	M	E	O	E
• Scannable undirected event	O	E	O	E
Scanning type:				
• Passive scanning	E	M	E	O
• Active scanning	E	O	E	C1
Link Layer control procedures:				
• Connection Update procedure	E	E	M	M
• Channel Map Update procedure	E	E	M	M
• Encryption procedure	E	E	O	O
• Feature Exchange procedure	E	E	M	M
• Version Exchange procedure	E	E	M	M
• Termination procedure	E	E	M	M
C1: If passive scanning is supported then active scanning is optional, otherwise active scanning is mandatory.				

Table 2.1: Physical Layer and Link Layer functionality for GAP roles when operating over an LE physical channel

2.3 用户要求和情境

蓝牙用户肯定希望能够与其它任何蓝牙设备连接。即使这两个连接的设备没有使用一样的应用，也应该能力找到其它设备并使用基本的蓝牙功能。当两个不同厂家的设备共用一样的应用，它们之间不能连接的原因只有各厂家在用户接口层用不同的名字调用蓝牙功能，或实现的处理流程不同。

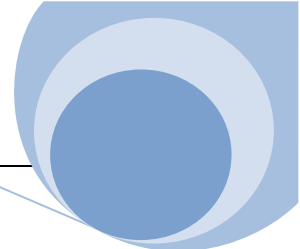
2.4 profile 基本原理

GAP 需要有名称，值和编码方案用于参数命名，并且熟练在用户接口层处理。

GAP 定义的是非服务及应用 profile 工作模式，但应用 profile 也可以使用 GAP, 并实现多种应用 profile。

GAP 定义的处理流程可以用于发现确认，命名和在发现状态的蓝牙设备的基本功能。只定义了没有信道或没有使用连接机制的处理过程。

GAP 定义了如何在两个设备间建立连接的一般处理流程，比如链接密码的交换。



2.5 一致性

蓝牙设备必须保证与 GAP 的一致性及互通性。

蓝牙设备若遵守其它蓝牙标准，应该使用相应的适配器，以保证与 GAP 在通用处理层的兼容。

GAP 所有强制要求的功能必须按定义方式实现。同样，可选支持和有条件才支持的功能也要按定义实现。所有强制支持，可选支持和有条件支持必须包含在蓝牙认证处理的一部分。

2.6 其它要求

BR/EDR 和 BR/EDR/LE 设备类型必须在 BR/EDR 传输上使用 SDP 做服务发现。实现 ATT 的设备 BR/EDR 设备类型必须也要实现 GAP。

BR/EDR/LE 和 LE-only(单模)设备类型应该在 LE 传输上使用 GATT 实现服务发现。LE-only 设备必须实现 ATT。

3、用户接口定义

3.1 用户接口层

在本标准的定义中，用户接口层用于放置蓝牙设备连接名，蓝牙术语和参数的数值和数值表示。

本节定义了用户接口应该使用的一般术语。

BR/EDR 和 LE 物理信道应该有统一的用户接口，以隐藏不同用户间的潜在的技术差异。设备应该支持在这样的 BR/EDR 和 LE 物理信道上实现发现和连接。

3.2 蓝牙参数描述

3.2.1 蓝牙设备名(BD_ADDR)

3.2.1.1 定义

设备必须能通过蓝牙设备地址识别。当设备工作在 BR/EDR 物理信道，设备地址就是蓝牙地址 (BD_ADDR)，参考 [Vol. 2], Part B Section 1.2。设备地址在执行设备发现过程时从对方获得。

BD_ADDR 根据 Section 9.2 建立。

3.2.1.1.1 LE-only 设备类型蓝牙设备地址

LE-only 设备类型应该使用公共地址或随机地址。公共地址应该设置到 BD_ADDR。随机地址定义在 Section 10.8。LE-only 应该支持 Section 10.7 定义的私有特性，这些私有特性需要 Section 10.8.2 定义的私有设备地址支持。

3.2.1.1.2 BR/EDR/LE 设备（双模）的蓝牙地址

BR/EDR/LE 设备应该在 BR/EDR 物理信道上使用 BD_ADDR，并在 LE 物理信道使用公共设备地址。公共设备地址应该设置为 BD_ADDR。BR/EDR/LE 设备类型应该能够通过随机设备地址与 BLE 单模设备或另一个 BR/EDR/LE 双模设备连接。

3.2.1.2 用户接口层术语

当蓝牙地址在 UI 层被指定，Bluetooth Device Address 术语将被使用。

3.2.1.3 描述

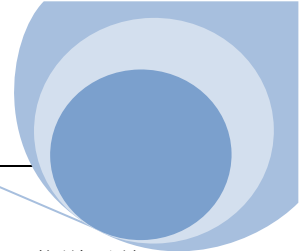
在 BB 层，蓝牙地址做为 48 位存在，如 [Vol. 2], Part B Section 1.2 定义。在链接层，公共地址和随机地址以 48 位地址存在。

在 UI 层，蓝牙地址应该以 12 个 HEX 字条存在，以 ':' 分成几个部分，如 '000C3E3A4B69' 或 '00:0C:3E:3A:4B:69'。在 UI 层，任何数据以自然排列顺序，如 16，表示为 0x10。

3.2.2 蓝牙设备名

3.2.2.1 定义

蓝牙设备名用于向对方显示自己。对于 BR/EDR 设备，蓝牙名由 LMP_name_req 的应答数据 LMP_name_res 返回。而 LE-only 设备，蓝牙设备名保存在 Section 12.1 定义的设备名特性中。



3.2.2.1.1 BR/EDR/LE 设备的蓝牙设备名

BR/EDR/LE 设置必须有一个单独的蓝牙设备名，该设备名应该与用于设备名发现的物理信道无关。

对于 BR/EDR 物理信道，设备名在 LMP_name_res 中。对于 LE 物理信道，可以从 Section 12.1 中定义的设备名特性中读取。

注意：如果设备支持 BR/EDR 的 ATT，可以通过 BR/EDR 的 ATT 读取设备名。

3.2.2.2 用户接口层术语

如果 UI 层指定蓝牙设备名，Bluetooth Device Name 这个术语将被使用。

3.2.2.3 描述

蓝牙设备名可以高达 248 字节，参考 [Vol. 2], Part C Section 4.3.5。设备名应该以 UTF-8 编码。

设备不能希望一般的远程设备能够处理大于前 40 个字符的蓝牙设备名。如果远程设备在显示能力上有限制，它可能只使用前 20 个字符。

3.2.3 蓝牙密钥

3.2.3.1 定义

蓝牙密钥用于在通过配对过程建立连接时两个蓝牙设备的相互认证。密钥用于配对过程上，以产生初始化连接密码。

PIN 密码可以从 UI 层输入，也可能存储在设备中，比如防止一个设备没有人机界面以输入和显示数字。

3.2.3.2 用户层接口术语

如果蓝牙 PIN 在用户接口层被指定，术语 Bluetooth Passkey 将被使用。

3.2.3.3 描述

蓝牙密钥是一串不同的数字。在上层有两种不同的表现，1、用于加密配对和加密管理，2、用于遗留配对

对于简单加密配对和加密管理，蓝牙密钥是 6 个数字是 000000-999999 中的一个整数。这个数值在认证的第一阶段被输入，用于加密配对入口，或 [Vol. 3], Part H Section 2.3.5 定义的加密管理中 TK 值。

对于遗留配对，参考 Section 17.2 on page 399，蓝牙 PIN 在不同的协议层上有不同的表现。PIN_{BB} 用于基底层，PIN_{UI} 用于用户接口层。PIN_{BB} 用 [1] 用于在配对过程中计算初始化密码。

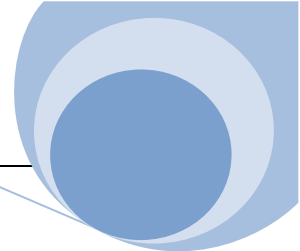
PIN 码可能高达 16 个字符。为了利用所有协议层的加密，PIN 应该是 16 个字符长。可变的 PIN 应该由 Unicode 0x00-0x7f 之间的字母或数字组成。如果 PIN 包括 10 进制数据，这些数据应该使用 Unicode 中的拉丁文表示。

为了设备的数据键盘的兼容性，固定的 PIN 和可变的 PIN 应该只由 10 进制数字组成。

如果设备支持输入超出 Unicode 0x00-0x7f 范围的字符，其它 Unicode 代码可以使用，除了 0xff00-0xffef 之间的半宽和全宽格式不能使用外。

示例：

User-entered code	Corresponding PIN _{BB} [0..length-1] (value as a sequence of octets in hexadecimal notation)
'0196554200906493'	length = 16, value = 0x30 0x31 0x39 0x36 0x35 0x35 0x34 0x32 0x30 0x30 0x39 0x30 0x36 0x34 0x39 0x33
'Børnelitteratur'	length = 16, value = 0x42 0xC3 0xB8 0x72 0x6e 0x65 0x6c 0x69 0x74 0x74 0x65 0x72 0x61 0x74 0x75 0x72



3.2.4 设备种类

3.2.4.1 定义

设备种类是设备在发现过程中接收到的一个参数，标志了支持哪一种设备类型和哪一种服务类型。

3.2.4.2 用户接口术语

设备种类中的参数信息应该称为 Bluetooth Device Class(如主要和次要的设备种类域)和 Bluetooth Service Type(服务类型域)。蓝牙设备种类和蓝牙服务类型在如下网址中的 GAP 中定义：

<https://www.bluetooth.org/en-us/specification/assigned-numbers>

当使用在蓝牙设备种类和蓝牙服务类型中的混合信息时，Bluetooth Device Type 将被使用。

3.2.4.3 描述

设备种类是 <https://www.bluetooth.org/en-us/specification/assigned-numbers> 中定义的一个位域。U 设备种类信息在 UI 层的描述特别定义。

3.2.4.4 用法

一些设备可能提供多于一个的服务，并且某一个给定的服务可能由不同的设备类型提供。因此，设备类型和服务支持不是一对一的关系。主要和次要的设备种类域应该用于确认设备是否支持某个特定的服务。在服务发现请求执行前，它可能被用于指示设备最可能支持希望的服务，并可能用于指导用户在一些设备中选择相同的服务。

3.3 配对

通过 BR/EDR 物理链路配对由 LMP 层定义。通过 LE 物理链路配对由加密管理定义，详细请参考 [Vol. 3], Part H Section 2.3。无论用户初始化连接和在两个设备之间创建特有目的的连接时的密钥认证，或用户由于事先没有共享一个共同的连接密钥而请求进入密钥认证过程，首先用户要主动要求执行连接，然后主动要求使用密钥认证。

9、LE 物理信道使用的工作模式和处理流程

一些不同的模式和处理流程可能在 LE 物理信道上同时执行。下面在 LE 物理信道定义的模式和处理流程将被使用：

- 1、广播模式和观察处理流程
- 2、发现模式和处理流程
- 3、连接模式和处理流程
- 4、绑定模式和处理流程

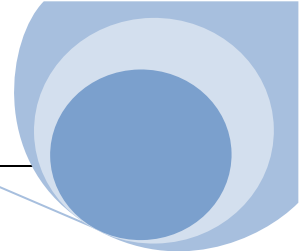
上面每一个模式和流程相互之间是独立的，但又是强相关的，因为对于大多数设备来说相互通信都需要模式和流程的组合。模式和流程可以由于用户直接的操作或设备的自主行为分别进入和退出。在执行以上任何模式和流程前，Host 应该使用 [Vol. 6], Part B Section 4.6 定义的本地链路层特性信息设置控制器。

9.1 广播模式和观察处理流程

广播模式和观察处理流程允许两个设备用广播事件以非定向无连接方式通信。支持广播和观察处理流程的工作在指定 GAP 角色的设备要求如下：

Broadcast Mode and Observation procedure	Ref.	Broadcaster	Observer
Broadcast mode	9.1.1	M	E
Observation procedure	9.1.2	E	M

Table 9.1: Broadcast mode and Observation procedure Requirements



9.1.1 广播模式

9.1.1.1 定义

广播模式提供了让设备使用广播事件发送无连接数据的方法。

9.1.1.2 条件

在广播模式的设备必须可以在不可连接非定向广播事件或可扫描非定向广播事件中发送数据。

广播数据应该用 Section 11 定义的广播数据(AD)类型格式化。在广播模式的设备应该设置 Section 11.1.3 定义的 AD(广播数据)类型中的'LE General Discoverable Mode' 标志或'LE Limited Discoverable Mode'。

当启动隐私, 如 Section 10.7 所定义, 且设备工作在广播模式, Host 应该按 Section 10.8.2.2 定义生成可分解的私有地址。Host 应该配置控制器使用可分解的私有地址做为广播事件中的广播设置地址。若禁止隐私, Host 应该 Tgap(private_addr_init) 定时器, 并配置控制器使用公共/静态地址做为广播设备地址。当 Tgap(private_addr_int) 定时器溢出, Host 应该重复上面的过程。设备不推荐在广播过程中发送设备名。如果设备接收到扫描请求, 如果扫描请求来自已经设备, 设备必须做出唯一应答。

注意: 只支持广播模式的设备不支持发送加密或数据签名数据。因为设备不能区分 [Vol. 3], Part H 定义的加密管理的密钥。密钥区分可能由使用外设或主设角色的设备执行。

9.1.2 观察处理

9.1.2.1 定义

观察处理提供了从发送广播事件的设备接收无连接数据的方法。

9.1.2.2 条件

执行观察处理的设备可能使用被动扫描或主动扫描以接收广播事件。

执行观察处理的设备可能使用主动扫描以接收工作在广播模式使用可扫描非定向广播事件的设备发送的扫描应答数据。当执行观察处理的设备在广播事件中接收接收到一个可分解的私有地址, 设备可能通过 Section 10.8.2.3 定义的分解私有地址解析流程解析该私有地址。

Section 10.7 定义的执行观察处理使能隐私的设备, 当主动扫描时, 应该使用一个非分解私有地址做为设备地址。生成非分解地址如 Section 10.8.2.1 定义。

注意: 在广播模式的设备发送动态数据的情况下, 接收设备应该禁止控制器重复过滤功能以使 Host 接收控制器接收到的所有广播数据包。

注意: 工作在观察流程的设备不支持接收加密的签名数据, 因为这个设备不能区分加密管理器在 [Vol. 3], Part H Section 3.6.1 定义的密钥。密钥区分可以由使用外设或主设角色的设备执行。

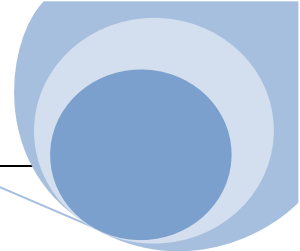
9.2 发现模式和处理流程

所有的设备应该在不可发现模式或其中一个可发现模式。在可发现模式的设备应该在通用可发现模式或限制可发现模式。在不可发现模式的设备不可以被发现。工作在通用发现模式或限制可发现模式的设备可以被执行发现设备找到。找到其它设备的设备执行 Section 9.2.5 定义的限制发现流程或 Section 9.2.6 定义的通用发现流程。

9.2.1 要求

Discovery modes and procedures	Ref.	Peripheral	Central
Non-Discoverable mode	9.2.2	M	E
Limited Discoverable mode	9.2.3	O	E
General Discoverable mode	9.2.4	C1	E
Limited Discovery procedure	9.2.5	E	O
General Discovery procedure	9.2.6	E	M
Name Discovery procedure	9.2.7	O	O
C1: if limited discoverable mode is not supported then general discoverable mode is mandatory, else optional.			

Table 9.2: Device discovery requirements



9.2.2 不可发现模式

9.2.2.1 描述

配置为不可发现模式的设备不可以被执行通用发现流程和限制发现流程的设备找到。

9.2.2.2 条件

工作在不可发现模式的设备发送广播事件将不会置位 AD 类型的 ‘LE General Discoverable Mode’ 标志或 ‘LE Limited Discoverable Mode’ 标志。在不可连接模式的外设可能发送不可连接非定向事件或可扫描非定向广播事件或不发送广播包。

9.2.3 限制发现模式

9.2.3.1 描述

配置为限制发现模式的设备可以被执行限制或通用设备发现流程的设备在一个限制的时间周期内发现。当用户执行使设备在一定时间内被发现时，限制可发现模式被使用。

9.2.3.2 条件

当设备在外设角色模式，设备可能支持限制可发现模式。当设备在广播角色，观察角色或主设角色，设备必须不能支持限制发现模式。

在限制可发现模式的设备发送不可连接广播事件，可扫描非定向广播事件或可连接非定向事件。

当设备在限制文件模式，设备将发送 Section 11.1.3 定义的带有标志的 AD 类型广播事件，这些标志描述如下：

- 1、LE 限制可发现模式标志必须设置为 1
- 2、LE 通用可发现模式标志必须设置为 0
- 3、LE-only 设备应该设置 ‘BR/EDR Not Supported’ 位为 1，设置 ‘Simultaneous LE and BR/EDR to Same Device Capable (Controller)’ 为 0，设置 ‘Simultaneous LE and BR/EDR to Same Device Capable (Host)’ 为 0。

广播数据建议包括下面的 AD 类型以使能更快速的连接。

- 1、Section 11.1.5 定义的 AD 类型发射功率等级
- 2、Section 11.1.2 定义的 AD 类型本地名
- 3、Section 11.1.1 定义的 AD 类型服务 UUID
- 4、Section 11.1.7 定义的 AD 类型从机连接间隔范围

设备保持在限制可发现模式不能超过 $T_{gap}(lim_adv_timeout)$ 。

当设备在限制可发现模式，Host 按下面配置控制器：

- 1、Host 必须设置广播过滤策略以 “为所有设备处理扫描和连接请求”
- 2、Host 建议设置最小广播间隔为 $T_{GAP}(lim_disc_adv_int_min)$
- 3、Host 建议设置最大广播间隔为 $T_{GAP}(lim_disc_adv_int_max)$

设备应该保持在限制可发现模式，直到确定连接或 Host 中断模式。

如果设备在限制可发现模式，且在非定向可连接模式，广播间隔值建议按 Section 9.3.4 设置。

设备在限制可发现模式必须不能设置 LE Limited Discoverable 标志和 LE General Discoverable 标志为 1。

注意：周期性改变数据建议放在广播数据中，静态数据建议放在扫描应答数据中。

注意：广播间隔的选择是功率消耗与设备发现时间的平衡。

9.2.4 通用可发现模式

9.2.4.1 描述

设备配置在通用发现模式目的在于被执行通用发现流程的设备找到。通用可发现模式常用于长时间的可发现模式。

9.2.4.2 条件

当设备在外设角色，可能支持通用可发现模式。当设备在广播，观察或主设角色，设置将不支持通用可发现模式。

设备在通用可发现模式发送不可连接广播事件，可扫描非定向广播事件或可连接非定向广播事件。

当在通用可发现模式，设备将发送 Section 11.1.3 定义的包含 AD 类型标志的广播数据，这些标志如下：

- 1、LE 限制可发现模式标志设置为 0
- 2、LE 通用可发现模式标志设置为 1
- 3、LE-only 设备类型将设置 ‘BR/EDR Not Supported’ 标志为 1，设置 ‘Simultaneous LE and BR/EDR to Same Device Capable (Controller)’ 为 0，设置 ‘Simultaneous LE and BR/EDR to Same Device Capable (Host)’ 为 0。

广播数据建议也包括如下的 AD 类型，以使能快速连接：

- 1、Section 11.1.5 定义的 AD 类型发射功率等级
- 2、Section 11.1.2 定义的 AD 类型本地名
- 3、Section 11.1.1 定义的 AD 类型服务 UUID
- 4、Section 11.1.7 定义的 AD 类型从机连接间隔范围

当设备在通用可发现模式，Host 按如下配置控制器：

- 1、Host 必须设置广播过滤策略 “以处理所有设备的扫描和连接请求”
- 2、Host 建议设置最小广播间隔为 $T_{GAP}(gen_disc_adv_int_min)$
- 3、Host 建议设置最大广播间隔为 $T_{GAP}(gen_disc_adv_int_max)$

设备必须保持在通用可发现模式，直到确定连接或 Host 中断该模式。

如果设备在通用可发现模式和定向可连接模式或不可连接模式，广播时间间隔值应按 Section 9.3.3 设置。

设备在通用可发现模式必须不能设置 LE Limited Discoverable 标志和 LE General Discoverable 标志为 1。

注意：周期性的数据改变建议放在广播数据中，静态数据建议放在扫描应答中。

注意：广播间隔的选择是功率损耗和设备可发现时间的平衡。

9.2.5 限制可发现处理

9.2.5.1 描述

设备执行限制可发现流程接收只在限制可发现模式设备的设备地址，广播数据和扫描应答数据。

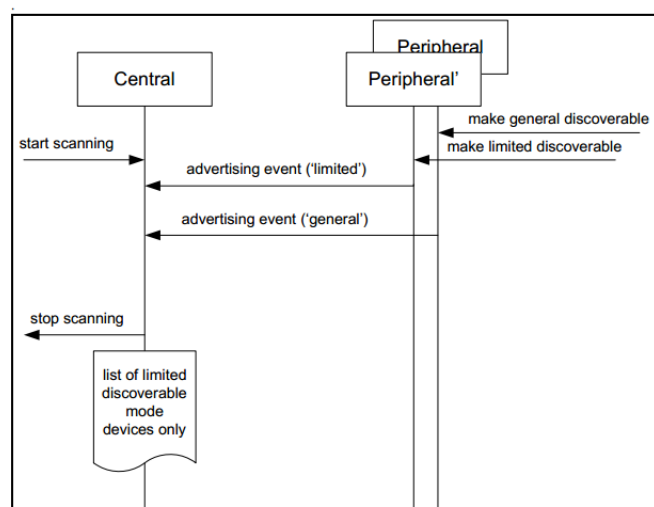


Figure 9.1: A Central performing limited discovery procedure discovering Peripherals in the limited discoverable mode

9.2.5.2 条件

当设备在主设角色，设备可能支持限制发现处理。当设备在广播，观察，或外设角色，设备决不能支持限制发现流程。

当 Host 执行限制发现流程，Host 按如下配置控制器：

- 1、Host 必须设置扫描过滤策略以“处理所有广播包”
- 2、Host 建议设置扫描间隔为 $TGAP(lim_disc_scan_int)$
- 3、Host 建议设置扫描窗口为 $TGAP(lim_disc_scan_wind)$
- 4、Host 建议配置控制器使用主动扫描

Host 必须开始扫描广播数据包并建议持续 $TGAP(lim_disc_scan_min)$ ，除非 Host 中断限制发现流程。

Host 必须校验广播数据中的 AD 类型标志。如果 AD 类型标志存在，且 LE 限制发现可连接标志为 1，则 Host 必须认为设备为一个可发现设备，否则广播数据将被忽略。Section 11.1.3 定义的 AD 类型标志。广播设备广播数据可能包含其它 AD 类型数据，比如 UUID AD 类型，发射功率电平 AD 类型，本地名 AD 类型，从机连接间隔范围 AD 类型。Host 可能在 Section 9.3 定义的任何连接建立流程处理中使用这些数据。

9.2.6 通用发现处理

9.2.6.1 描述

设备执行通用发现处理从在限制可发现模式或通用可发现模式的设备接收设备地址，广播数据和扫描应答数据。

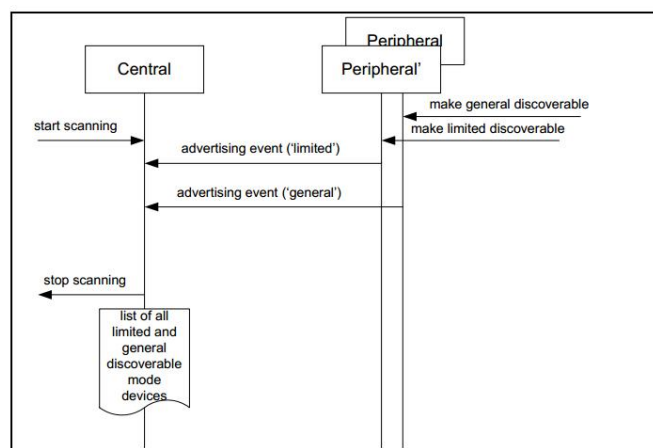


Figure 9.2: A Central performing general discovery procedure discovering Peripherals in the limited discoverable mode and general discoverable mode.

9.2.6.2 条件

当设备在主设角色，设备将支持通用发现处理。当设备在广播，观察或外设角色决不能支持通用发现处理。

当 Host 执行通用发现流程，Host 配置控制器如下：

- 1、Host 必须设置扫描过滤策略以“处理所有广播数据包”
- 2、Host 建议设置扫描间隔为 $TGAP(gen_disc_scan_int)$
- 3、Host 建议设置扫描窗口为 $TGAP(gen_disc_scan_wind)$
- 4、Host 建议配置控制器使用主要扫描

Host 必须开始扫描广播包且建议持续至少 $TGAP(gen_disc_scan_min)$ 。这个处理流程可能被 Host 过早中断。

Host 必须检查广播数据包中的 AD 类型标志。如果 AD 类型标志存在且 LE General Discoverable Mode 标志或 LE Limited Discoverable Mode 标志设置为 1，则 Host 必须认为设备是可发现设备，否则广播数

据将被忽略。可发现设备广播数据包可能包含其它 AD 类型，比如 UUID AD 类型，发射功率电平 AD 类型，本地名 AD 类型，人机连接间隔范围 AD 类型。Host 可能在 Section 9.3 定义的任何连接建立流程处理中使用这些数据。

9.2.7 设备名发现处理

9.2.7.1 描述

设备名发现处理用于获取远程可连接设备的蓝牙设备名。

9.2.7.2 条件

如果执行限制发现流程或通用发现流程后没有获得完整的设备名，则设备名发现处理可能被执行。

设备名发现流程必须按以下流程执行

- 1、Host 必须使用 Section 9.3 定义的流程建立方法之一建立一个连接
- 2、Host 必须使用 UUID 字符读取的 GATT 流程读取设备名字符。
- 3、连接可能在 GATT 流程完成后被中断。

9.3 连接模式和处理流程

连接模式和处理允许设备与另一个设备建立一个连接。

在定向和非定向模式中，外设设备可能支持 Section 10.7 定义的隐私特性。主设可能支持与不支持只用于通用连接建议流程的隐私特性的外设连接。

当设备已经连接，连接参数可能通过连接参数更新流程更新。连接设备可能使用中断连接流程中断连接。支持连接模式和处理流程的设备要求定义如下表：

9.3.1 Requirements

Connection Modes and Procedures	Ref.	Peripheral	Central	Broadcaster	Observer
Non-connectable mode	9.3.2	M	E	M	M
Directed connectable mode	9.3.3	O	E	E	E
Undirected connectable mode	9.3.4	M	E	E	E
Auto connection establishment procedure	9.3.5	E	O	E	E
General connection establishment procedure	9.3.6	E	C1	E	E
Selective connection establishment procedure	9.3.7	E	O	E	E
Direct connection establishment procedure	9.3.8	E	M	E	E
Connection parameter update procedure	9.3.9	O	M	E	E
Terminate connection procedure	9.3.10	M	M	E	E
C1: Mandatory if privacy feature is supported, else optional					

Table 9.3: Connection modes and procedures requirements

9.3.2 不可连接模式

9.3.2.1 描述

设备在不可连接模式必须禁止连接的建立。

9.3.2.2 条件

当设备在外设角色，设备将支持不可连接模式。设备在主设角色不能发送广播事件，因此设备被认为支持不可连接模式。

设备在广播，观察角色不能建立连接，因此，设备被认为支持不可连接模式。

外设设备在不可连接模式可能发送不可连接非定向广播事件或可扫描非定向广播事件。

9.3.3 定向连接模式

9.3.3.1 描述

设备在定向可连接模式必须能从能执行自动连接建立流程或通用连接建立流程的已知设备接收连接请求。

9.3.3.2 条件

当设备在外设角色可能支持定向可连接模式。这个模式必须只能用于对方设备地址是已知的情况下。当设置在广播，观察或主设角色，设备不支持定向可连接模式。

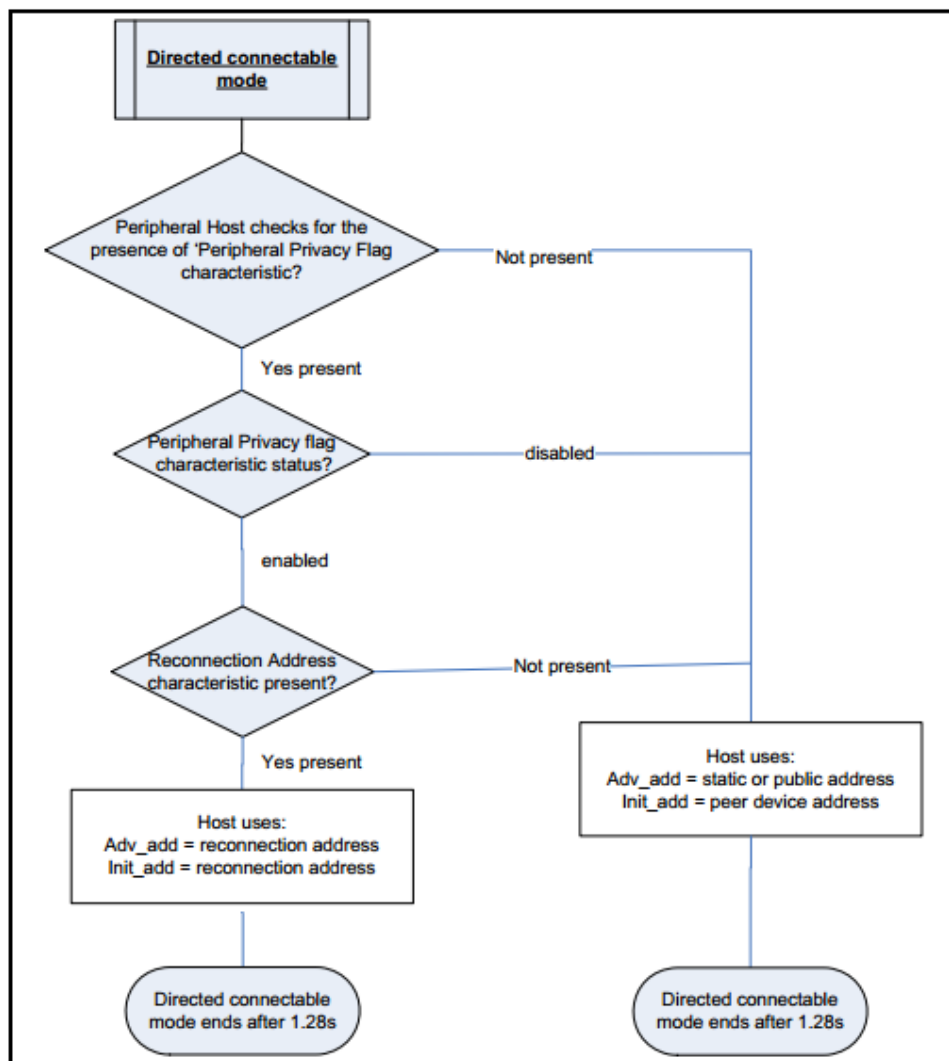


Figure 9.3: Flow chart for a device entering the directed connectable mode

Host 进入定向可连接模式必须执行以下几步：

- 1、如果外设隐私标志存在于本地设备，Host 必须处理步骤 2，否则 Host 必须处理步骤 5
- 2、如果外设隐私标志设置为 1，Host 必须处理步骤 3，否则处理步骤 5
- 3、如果重连接地址存在于本地设备，Host 必须处理步骤 4，否则处理步骤 5
- 4、Host 使用重连接地址在定向可连接广播事件中做为初始化者设备地址和广播者设备地址。Host 必须处理步骤 6.
- 5、Host 使用静态或公共地址做为广播设备地址。Host 使用已经设备地址做为初始化者地址，Host 必须处理步骤 6.
- 6、Host 建议按如下配置控制器：最小广播间隔设置为 $TGAP(dir_conn_adv_int_min)$ ，最大广播间隔设置为 $TGAP(dir_conn_adv_int_max)$ 。Host 必须配置控制器以发送定向可连接广播事件。

注意：当设备在定向可连接模式建议了一个连接，设备将退出该模式必进入到不可连接模式（已连接）。

9.3.4 非定向可连接模式

9.3.4.1 描述

设备在非定向可连接模式将接收能执行自动连接建议流程或通用连接建立流程的设备发连接请求。

9.3.4.2 条件

当设备在外设角色，设备必须支持非定向可连接模式。当设备在广播，观察或主设角色，设备禁止支持非定向可连接模式。

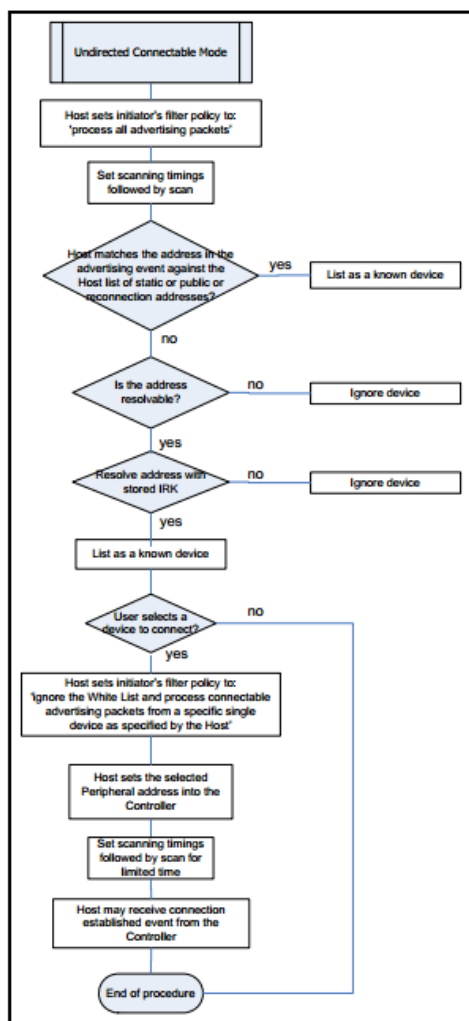


Figure 9.4: Flow chart for a device entering the undirected connectable mode

Host 进入非定向可连接模式必须执行下面的步骤:

- 1、如果外设隐私标志存在, Host 必须处理步骤 2, 否则处理步骤 4
- 2、如果外设隐私标志值为 1, 即隐私使能, Host 必须处理步骤 3, 否则处理步骤 4
- 3、Host 必须使用 Section 10.8.2.2 定义的可分解私有地址生成过程生成可分解私有地址。Host 必须设置定时器大于等于 TGAP(private_addr_int), Host 必须在可连接非定向广播事件中配置控制器使用分解私有地址做为广播地址。Host 建议设置广播过滤策略以“只处理怀特表中的设备的扫描和连接请求”或“处理所有设备的扫描和连接请求”。Host 必须配置控制器发送非定向可连接广播事件, 以设置最小广播间隔为 TGAP(undir_conn_adv_int_min), 最大广播事件间隔为 TGAP(undir_conn_adv_int_max)。如果隐私禁止, 则 Host 必须林彪 TGAP(private_addr_int) 定时器并处理步骤 4。当 TGAP(private_addr_int) 定时器溢出, Host 必须配置控制器停止广播并重复步骤 3
- 4、Host 使用本地设备静态或公共地址做为广播设备地址。Host 建议配置控制最小广播间隔为 TGAP(undir_conn_adv_int_min), 最大广播间隔为 TGAP(undir_conn_adv_int_max)。Host 必须配置控制器发送非定向可连接广播事件。

注意: 当设备在非定向可连接模式建立一个连接, 设备将退出该模式并进入到不可连接模式(已经连接)。

9.3.5 自动连接建立流程

9.3.5.1 描述

自动连接建议过程允许 Host 配置控制器在定向可连接模式或非定向可连接模式自动与一个或多个设备建立连接。控制器中的怀特表用于存储设备地址。这个过程使用初始化者控制器中的怀特表。控制器自动与设备地址与怀特表里存储的设备地址匹配的设备建立一个连接。

9.3.5.2 条件

当设备在主设角色, 设备可能支持自动连接建立流程。当设备在广播, 观察或外设角色, 设备禁止支持自动连接建立流程。

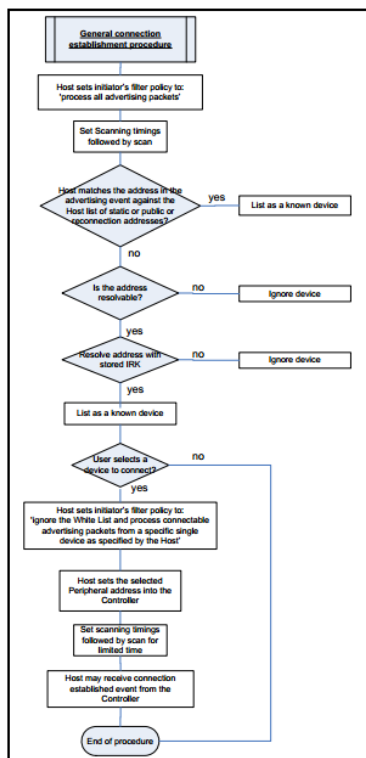


Figure 9.5: Flow chart for a device performing the auto connection establishment procedure

当 Host 执行自动连接建立流程，Host 按如下配置控制器：

- 1、Host 必须将自动连接的设备地址列表写入怀特表
- 2、Host 必须设置初始化者过滤策略以“处理来自怀特表中所有设备的可连接广播数据包”
- 3、Host 建议设置最小连接间隔为 TGAP(conn_est_int_min).
- 4、Host 建议设置最大连接间隔为 TGAP(conn_est_int_max)
- 5、Host 建议设置从机等待为 TGAP(conn_est_latency)
- 6、Host 建议设置扫描间隔为 TGAP(auto_conn_est_scan_int)，且设置扫描窗口为 TGAP(auto_conn_est_scan_wind)
- 7、Host 必须初始化连接。对于 Section 10.7 定义的隐私使能的设备，Host 必须使用非分解私有地址。

当连接建立或 Host 中断流程时，这个流程被中断。

9.3.6 通用连接建立过程

9.3.6.1 描述

通用连接建立过程允许 Host 与在定向可连接模式或非定向可连接模式中已知设备集合中的设备建立连接。

9.3.6.2 条件

当设备在主设模式，如果设备支持隐私特性，设备必须支持通用连接建立流程，否则设备在主设角色可能支持通用连接建立流程。当设备在广播，观察或外设角色，设备必须支持通用连接建立流程。

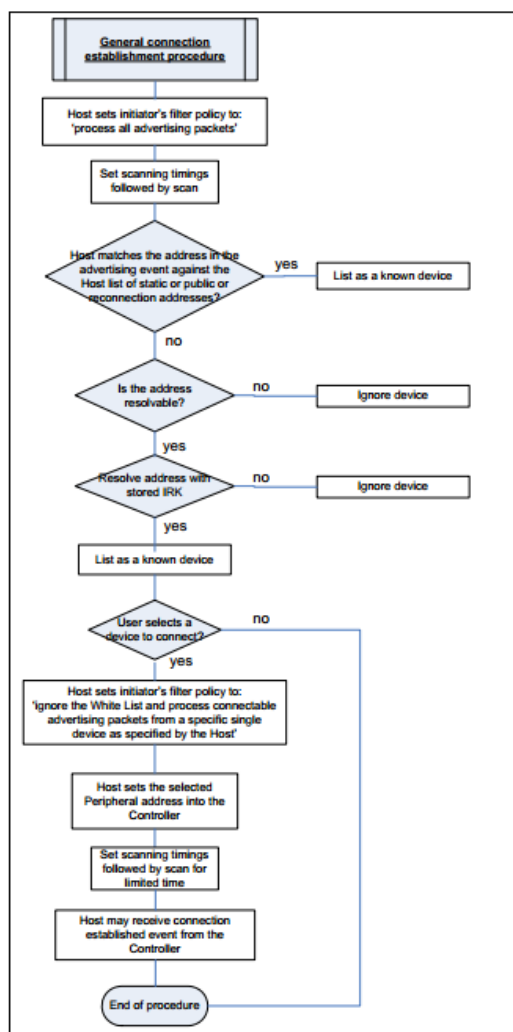
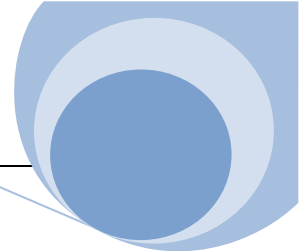


Figure 9.6: Flow chart for a device performing the general connection establishment procedure



当 Host 执行通用连接建立流程，Host 按如下配置控制器

- 1、Host 必须设置扫描过滤策略以“处理所有广播数据包”
- 2、Host 建议设置扫描间隔为 TGAP(gen_conn_est_scan_int).
- 3、Host 建议设置扫描窗口为 TGAP(gen_conn_est_scan_wind).
- 4、Host 必须启动扫描。对于一个 Section 10.7 定义的隐私使能的外设，如果 Host 在主动扫描，Host 必须使用非分解私有地址。

如果本地设备支持隐私属性，且 Host 发现一个可分解私有地址的设备，Host 可能通过执行 Section 10.8.2.3 定义的“解析私有地址流程”解析这个可分解私有地址。如果解析成功，Host 认为发现的设备是已知设备，否则发现的设备被忽略。

当 Host 发现一个设备，Host 将尝试连接，Host 必须停止扫描，并使用定向连接建立流程初始化连接。

当支持隐私的主设的 Host 和‘Peripheral Privacy Flag’ 设置为 1 的设备建立了连接，主设 Host 将生成一个新的重连接地址，并在主设 Host 发送任何应用数据前更新外设重连接地址特征值。外设应该使用老的重连接地址，如果重连接地址没有更新。

如果主设不支持隐私，当主设 Host 和‘Peripheral Privacy Flag’ 设置为 1 的设备建立连接，主设 Host 可能设置 Peripheral Privacy 标志为 0。如果主设 Host 不能禁止外设隐私标志，主设将不通用与外设连接。

当连接建立或 Host 中断流程，这个流程被中断。

9.3.7 选择性连接建立流程

9.3.7.1 描述

选择性连接建立流程允许 Host 和怀特表中带有连接配置参数的设备建立连接。

9.3.7.2 条件

当设备在主设角色，设备可能支持选择性建立流程。当设备在广播，观察或外设角色，设备禁止支持选择性连接建立流程。

当 Host 执行选择性连接流程，Host 按如下配置控制器：

- 1、Host 必须把选择性连接的设备地址列表写入怀特表。
- 2、Host 必须设置扫描过滤策略以“只处理怀特表中的广播事件”
- 3、Host 建议设置扫描间隔为 TGAP(sel_conn_est_scan_int)
- 4、Host 建议设置扫描窗口为 TGAP(sel_conn_est_scan_wind)
- 5、Host 必须启动主动扫描或被动扫描。

当 Host 发现一个将连接的设备，Host 将停止扫描，并使用带有该设备连接配置参数的定向连接建立流程初始化连接。

当连接已经建立或 Host 中断流程，这个流程将中断。

9.3.8 定向连接建立流程

9.3.8.1 描述

定向连接建立流程允许 Host 和单个带有选择连接配置参数的设备连接。

9.3.8.2 条件

当设备在主设角色，设备必须支持定向连接建立流程。当设备在广播，观察或外设角色，设备禁止支持定向连接建立流程。

当 Host 执行定向连接建立流程，Host 按如下配置控制器：

- 1、Host 必须设置初始化过滤策略以“忽略怀特表内的设备并只处理由 Host 确定的单个设备的可连接广播包”
- 2、Host 必须设置对等地址为连接设备的地址。
- 3、Host 必须设置连接设备的连接配置参数

Host 必须初始化一个连接，对于 Section 10.7 定义的隐私使能，Host 必须使用不可解析私有地址。当连接建立后或 Host 中断流程，该流程中断。

9.3.9 连接参数更新流程

9.3.9.1 描述

连接参数更新流程允许外设或主设更新已建立连接的链路层连接参数。

9.3.9.2 条件

当设备在主设角色，设备必须支持连接参数更新流程。当设备在外设角色，设备可能支持连接参数更新流程。当设备在广播或观察角色，设备禁止支持连接参数更新流程。

主设初始化连接参数更新流程必须使用 [Vol. 6], Part B Section 5.1.1 定义的带有需求的连接参数的链路层连接更新流程。

外设初始连接参数更新流程必须使用 [Vol. 1], Part A Section 4.2 定义的带有要求的连接参数 L2CAP 连接参数更新请求命令。外设禁止在一个 L2CAP 连接参数更新应答接收到的 T GAP(conn_param_timeout) 内发送 L2CAP 连接参数更新请求命令。如果连接参数不是外设所期望的，外设可能断开连接，并回应出错代码。外设应该允许主设给定的连接间隔的误差。

9.3.10 中断连接流程

9.3.10.1 描述

中断连接流程处理允许 Host 中断与对方的连接。

9.3.10.2 条件

主设和外设必须支持中断连接流程，广播和观察禁止支持中断连接流程。

Host 初始化中断连接流程必须使用 [Vol. 6], Part B Section 5.1.6 定义的链路层中断流程。

9.4 绑定模式和处理

绑定，允许两个设备交换和存储密钥和认证信息以创建一个信任的关系。 [Vol. 3], Part H Section 2.4.1 定义的密钥和认证信息也称为绑定信息。当设备存储绑定信息，称为设备已经绑定或绑定已经建立。

绑定有两个模式，不可绑定模式，绑定模式。绑定发生在两个设备的可绑定模式中。支持绑定模式和处理的要求如下：

9.4.1 Requirements

Bonding	Ref.	Peripheral	Central
Non-Bondable mode	9.4.2	M	M
Bondable mode	9.4.3	O	O
Bonding procedure	9.4.4	O	O

Table 9.4: Bonding Compliance Requirements

9.4.2.1 不可绑定模式

设备在不可绑定模式不允许与另外一个设备创建绑定。

9.4.2.2 条件

当设备在外设或主设角色，设备将支持不可绑定模式。如果设备不支持加密管理定义的配对，则该设备认为在不可绑定模式。

如果支持加密管理配对，Host 必须设置 Bonding_Flags 为 [Vol. 3], Part H Section 3.5.1 定义的“无绑定”，且绑定信息不能交换或存储。

9.4.3 可绑定模式

9.4.3.1 描述

设备在可绑定模式允许与另外一个在可绑定模式的设备建立绑定。

9.4.3.2 条件

Host 必须在过程中设置 Bonding_Flags 为“通用绑定”。

9.4.4 绑定处理

9.4.4.1 描述

当一个没有绑定的设备尝试访问需要绑定的服务时，绑定处理将被执行。绑定处理执行的目的在于在两个设备间建立绑定。

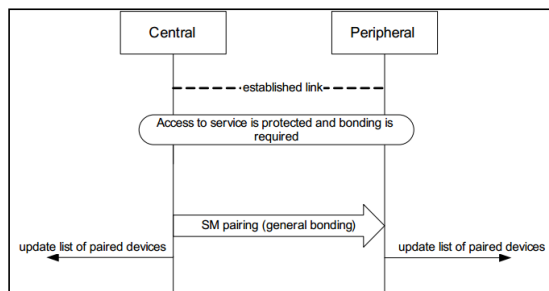


Figure 9.7: The Bonding procedure

9.4.4.2 条件

当设备在外设或主设角色，设备可能支持绑定处理。当设备在广播或观察模式，设备禁止支持绑定处理。

主设的 Host 将 [Vol. 3], Part H Section 3.5.1 定义的 Bonding_Flags set 为 Bonding 以初始化 [Vol. 3], Part C Section 2.1 定义的配对流程。如果对方在可绑定模式，设备必须在加密数据里交换并存储绑定信息。

如果设备按 Table 10.7 定义使能隐私，Host 应该在配对过程中发送它的 IRK 给对方设备，并应答对方设备的 IPK。如果认证要求不能区分 IRK，Host 可以退出配对流程。如果配对流程因为认证要求和 IRK 区分请求失败，配对过程应该不带 IRK 区分请求继续重试。

10 LE 加密处理

本章节定义了与连接加密相关的模式和处理。以下是定义的模式和处理：

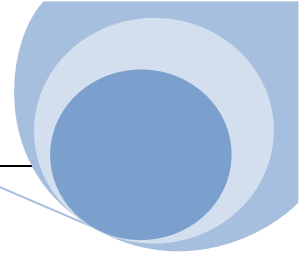
- 1、LE 加密模式 1
- 2、LE 加密模式 2
- 3、认证处理
- 4、授权处理
- 5、连接数据签署处理
- 6、认证签署数据处理

设备支持 LE 加密模式和处理的需求如表 10.1 所示：

10.1 REQUIREMENTS

Security Modes and Procedures	Ref.	Broadcaster	Observer	Peripheral	Central
LE Security mode 1	10.2.1	E	E	O	O
LE Security mode 2	10.2.2	E	E	O	O
Authentication procedure	10.3	E	E	O	O
Authorization procedure	10.5	E	E	O	O
Connection data signing procedure	10.4.1	E	E	O	O
Authenticate signed data procedure	10.4.2	E	E	O	O

Table 10.1: Requirements related to security modes and procedures



10.2 LE 加密模式

待译。。。。

11、广播和扫描应答数据格式

广播数据和扫描应答数据格式如下图所示。数据由有定义和无定义部分组成，有定义的由 AD 结构序列组成。每一个 AD 结构应该有一个字节的长度域，这个长度域包括了长度和数据的总长度。数据域的第一个字节包含了 AD 类型域。数据域保存的 Length-1 个字节内容由 AD 类型域决定，这部分称为 AD 数据。无定义部分用于扩展广播和扫描应答数据达到 31 字节，并用 0 填充。

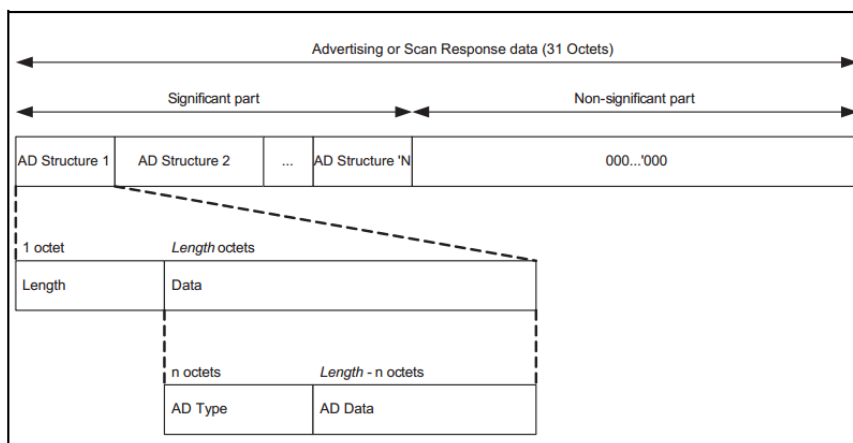


Figure 11.1: Advertising and Scan Response data format

如果长度域设置为 0，则数据域没有数据。这只能用于过早中断广播或扫描应答数据。

只有广播或扫描应答数据有定义的部分需要通过空中发送。

广播和扫描应答数据在广播事件中发送。广播数据放在 ADV_IND, ADV_NONCONN_IND 和 ADV_SCAN_IND 包的 AdvData 域中。扫描应答数据在 SCAN_RSP 中的 ScanRspData 域中发送。

11.1 AD 类型定义

所有的 AD 类型在蓝牙数据分配文档中列出。

所有多字节数据使用小端模式。

11.1.1 服务 UUIDs

设备可能在广播数据时列出自己实现的服务 UUIDs。服务 UUIDs 可能是 16 位或 128 位。每一类型的服务 UUIDs 如表 Table 18.2。

11.1.2 本地名

本地名 AD 类型包括设备名，包括 Section 3.2.2 定义的完全或精简设备名。该 AD 类型标志了设备名是完全的还是精简型。如果设备名是精简的，完整的设备名可以通过 Section 12 定义的设备名特征读取。广播和扫描应答数据禁止包括一个以上的本地名 AD 类型实例。精简设备名必须只包含完整设备名的开始字符。比如，如果设备名是 BT_Device_Name，则 BR/EDR 的精简名可能是 BT_Device，LE 的精简名可能是 BT_Dev。该数据格式如 Table 18.3. 定义。

11.1.3 标志

标志 AD 类型包括一些标志位，以布尔值表示。

LE 物理信道使用的标志如下：

- Limited Discoverable Mode
- General Discoverable Mode
- BR/EDR Not Supported
- Simultaneous LE and BR/EDR to Same Device Capable (Controller)
- Simultaneous LE and BR/EDR to Same Device Capable (Host)

标志 AD 类型禁止包括扫描应答数据。广播数据禁止包括一个以上的标志 AD 类型实例。如果以上标志位任何一位为 1，标志 AD 类型必须包括广播数据。如果以上所有位为 0，标志 AD 类型可能从广播数据中省略。标志 AD 类型如 Table 18.1 定义。

11.1.4 装配标准数据

装配标准 AD 类型用于装配标准数据。参考 Section 18.11.

11.1.5 发射功率电平

发射功率电平 AD 类型指示了广播包的发送功率电平。发射功率电平 AD 格式定义如 Table 18.4。

11.1.6 加密管理 OOB

加密管理 OOB 机制用于加密管理持续发现信息和其它与配对流程相关的其它信息。加密管理 OOB AD 格式由 Section 18.7 定义。这必须只能通过 OOB 机制使用。

11.1.7 加密管理 TK 值

加密管理 TK 值 AD 类型定义如 Section 18.6。这必须在 OOB 机制下使用。

11.1.8 从机连接间隔范围

从机连接间隔范围 AD 类型包括所有逻辑连接的外设优先连接间隔范围。

注意：最小值基于外设的电池考虑，最大值由外设的可用缓冲决定。

主设在建立连接时建议使用外设的从机连接间隔范围 AD 类型中的信息。

从机连接间隔范围 AD 类型定义如 Section 18.8

11.1.9 服务请求

服务请求 AD 类型之一可能被发送以邀请其它在服务请求数据中显示一个或多个服务定义的设备相连接。设备建立在非定向可连接模式和其中一个可发现模式。这样使得主设提供一个或多个服务以连接到该外设，以至于外设可以使用主设上的服务。

服务请求 AD 类型如 Section 18.9 定义。

11.1.10 服务数据

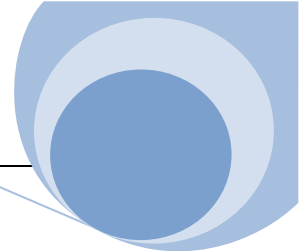
服务数据 AD 类型由带有与其服务相关的数据 服务 UUID 组成。

服务数据 AD 类型如 Section 18.10 所定义。

11.2 广播数据示例

以下是 AD 类型广播数据示例

Value	Notes
0x02	Length of this Data
0x01	AD type = Flags
0x01	LE Limited Discoverable Flag set
0x0A	Length of this Data
0x09	AD type = Complete local name
0x50	'P'
0x65	'e'
0x64	'd'
0x6F	'o'
0x6D	'm'
0x65	'e'
0x74	't'
0x65	'e'
0x72	'r'



12、GAP 特性

UUID 必须是 GAP 的 UUID。

每一个 GAP 角色的特性要求如 Table 12.1 所示。设备必须只能有一个 GAP 服务实例。GAP 服务是蓝牙数据分配文档定义的 GATT 的 UUID 基础服务。

Characteristics	Ref.	Broadcaster	Observer	Peripheral	Central
Device Name	12.1	E	E	M	M
Appearance	12.2	E	E	M	M
Peripheral Privacy Flag	12.3	E	E	C1	E
Reconnection Address	12.4	E	E	C2	E
Peripheral Preferred Connection Parameters	12.5	E	E	O	E
C1: Mandatory if privacy feature is supported, else excluded C2: Optional if privacy feature and Peripheral privacy Flag characteristic are supported, else excluded					

Table 12.1: Requirements related to characteristics

支持多个 GAP 角色的设备包含所有特性以满足所支持的角色要求。当设备在对应的角色时，必须连续检查没有使用的特性以发现相应特性。

12.1 设备名特性

设备名特性必须包含一个 Section 3.2.2 定义的 UTF-8 格式的设备名。设备各特征值必须可以不经认证和授权即可读取。设备名特征值可能可写。

Attribute Handle	Attribute Type	Attribute Value
0xMMMM	0x2A00 – UUID for «Device Name»	Device Name

Table 12.2: Device Name characteristic

设备名特征值必须在 0-248 个字符内。设备必须只有一个设备名特征值实例。

12.2 外观特征

外观特征定义了设备外在表现。这使得正在发现的设备使用一个图标或字符串或类似的描述设备。外观特征值必须可以不经认证或授权即可读取。外观特征可能被写。

Attribute Handle	Attribute Type	Attribute Value
0xMMMM	0x2A01 – UUID for «Appearance»	Appearance

Table 12.3: Appearance characteristic

外观特征值必须是一个枚举值，这些值在蓝牙数字安排文件里有定义。外观特征值应该是两个字节长度。设备必须只有一个外观特征实例。

12.3 外设隐私标志特征

外向隐私标志特征定义了当前的设备是否使用隐私功能，详情 Table 10.7.2。

Attribute Handle	Attribute Type	Attribute Value
0xMMMM	0x2A02 – UUID for «Peripheral Privacy Flag»	Peripheral Privacy Flag

Table 12.4: Peripheral Privacy Flag characteristic

外设隐私标志特征值应该为一个字节长度：

0：设备中隐私禁止

1：设备中隐私使能

2-255：预留

设备必须只有一个外设隐私标志特征实例。

12.4 重连接地址特征

重连接特征定义了重连接地址，参考 Table 10.7.2。重连接地址是一个非分解私有地址。

Attribute Handle	Attribute Type	Attribute Value
0xMMMM	0x2A03 – UUID for «Reconnection Address»	Reconnection Address

Table 12.5: Reconnection Address characteristic

重连接特征值必须是 6 字节长度。设备必须只有一个重连接地址特征实例。

12.5 外设优先连接参数特征

外设优先连接参数（PPCP）特征包括外设优先连接参数。

Attribute Handle	Attribute Type	Attribute Value
0xMMMM	0x0x2A04 – UUID for «Peripheral Preferred Connection Parameters»	Peripheral Preferred Connection Parameter

Table 12.6: Peripheral Preferred Connection Parameters characteristic

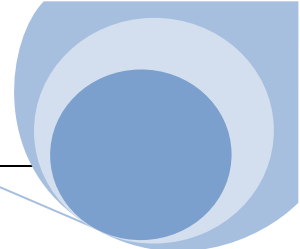
外设优先连接参数特征值必须是 8 字节长度，设备必须只有一个外设优先连接参数特征实例。

优先连接参数结构数据定义如下：

Name	Size (Octet)	Description
Minimum connection interval	2	Defines minimum value for the connection interval in the following manner: $connInterval_{min} = Conn_Interval_Min * 1.25 \text{ ms}$ Conn_Interval_Min range: 0x0006 to 0x0C80 Value of 0xFFFF indicates no specific minimum. Values outside the range are reserved. (excluding 0xFFFF)

Name	Size (Octet)	Description
Maximum connection interval	2	Defines maximum value for the connection interval in the following manner: $connInterval_{max} = Conn_Interval_Max * 1.25 \text{ ms}$ Conn_Interval_Max range: 0x0006 to 0x0C80 Shall be equal to or greater than the Conn_Interval_Min. Value of 0xFFFF indicates no specific maximum. Values outside the range are reserved. (excluding 0xFFFF)
Slave latency	2	Defines the slave latency for the connection in number of connection events. Slave latency range: 0x0000 to 0x03E8 Values outside the range are reserved.
Connection Supervision timeout multiplier	2	Defines the connection supervisor timeout multiplier as a multiple of 10ms. Range: 0xFFFF indicates no specific value requested. Range: 0x000A to 0x0C80 Time = N * 10 msec Time Range: 100 msec to 32 seconds Values outside the range are reserved. (excluding 0xFFFF)

Table 12.7: Format of the preferred connection parameters structured data



PARTG GATT(GENERIC ATTRIBUTE PROFILE)

1、介绍

1.1 简介

GATT(Generic Attribute Profile)定义了使用 ATT 协议的服务框架。这个框架定义了服务和特征的处理过程和格式。处理过程的定义包括：发现，读，写，通知和指示特征，也配置广播特征。

1.2 Profile 依赖

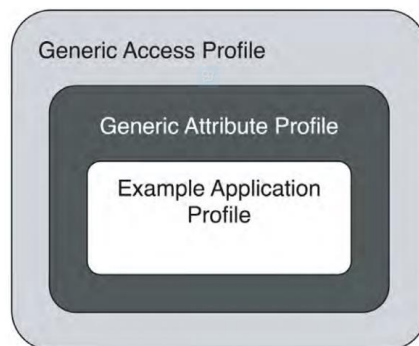


Figure 1.1: Profile dependencies

图 11 描述了 Profile 的结构及依赖。如果 Profile 将依赖于其隐含或直接引用另外一个 Profile 部分内容的 Profile。

1.3, 1.4, 1.5 与技术无关，只是版权信息，不译。

2、Profile 概述

GATT 框架设计目的是让应用程序或其它 profile 使用，所以客户端可以和服务器端通信。服务器端包含了多个属性，GATT 框架定义了如何使用这些属性协议来发现，读，写和获得这些属性的标志，同时也配置属性的广播。

2.1 协议栈

下图显示了使用此框架的对等协议：

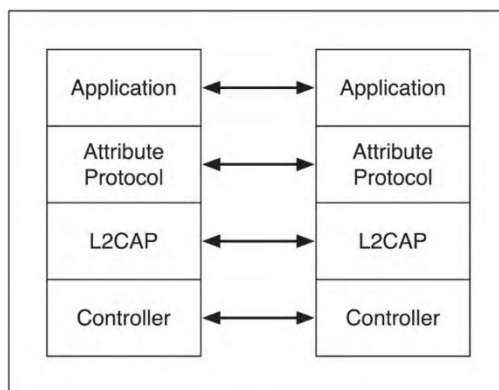


Figure 2.1: Protocol model

2.2 配置和角色

下面在本框架中实现的角色是为设备定义的：

客户端——这个设备是向服务器端初始化命令和请求，并能接收服务器发送的应答，标志和通知。

服务端——这个设备可以接受客户端输入的命令和请求，并发送应答，标志和通知给客户端。

注意：设备的角色并不固定。当设备初始化一个定义的流程时，角色被确定，当流程结束时，角色被释放。

设备可以在同一时间做为两个角色。

一个配置的示例用于描述该框架，如下：

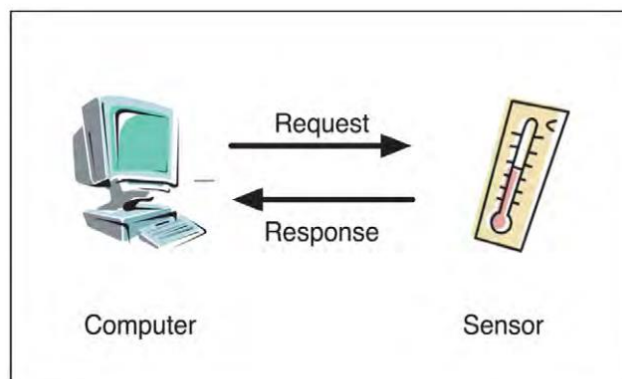


Figure 2.2: Examples of configuration

在图 2.2 中，电脑是温度计服务的客户端，而传感器是温度服务方。电脑初始华流程以配置传感器或读取传感器值。在这个例子中，传感器提供了部分温度服务特征信息，并可能禁止一些特征被写入。同样，传感器以合适的值应答读请求。

2.3 用户要求和脚本

本框架包括了以下脚本

- 1、交换配置
- 2、设备上的服务和特征发现
- 3、读特征值
- 4、写特征值
- 5、特征值通知
- 6、特征值表示

2.4 特征值基本原理

这个框架可以用在任何物理层，使用属性协议 L2CAP 信道，称为 ATT Bearer。下面是底层客户端与服务端通信要求的简单总结：

- 1、ATT Bearer 通过使用 “Channel Establishment” 建立，如 Section 6 定义
- 2、在 LE 物理链路层，加密特性，比如授权，认证和加密是可选的。在 BR/EDR 物理链路层，加密是强制的。
- 3、GATT 框架多个字节域必须先发送低字节，即小端模式。

2.5 属性协议

GATT 框架需要属性协议的实现和 Section 4.2 所示的属性操作码要求。

2.5.1 概述

GATT 框架使用属性协议在两个设备间以命令，请求，应答，指示，通知和确认的格式传输数据。这些数据包含在属性协议 PDU 中。

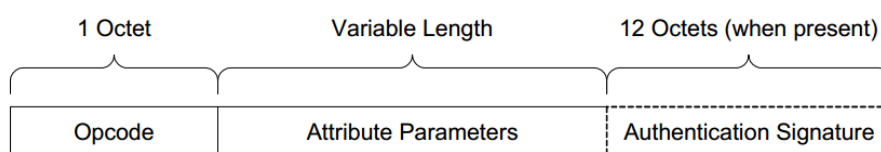


Figure 2.3: Attribute Protocol PDU

Opcode 中包括了明确的命令，请求，应答，指示，通知或确认操作码和认证标志。
属性参数包含了明确的命令或请求或应答返回的数据，指示或通知。

验证是可选的，如 [Vol. 6] Part F, Section 3.3.1 定义。

属性协议命令和请求作用于服务端设备属性存储的值。属性由四个部分组成：属性句柄，属性类型，属性值和属性权限。一个给定的实际实现是明确的实现。

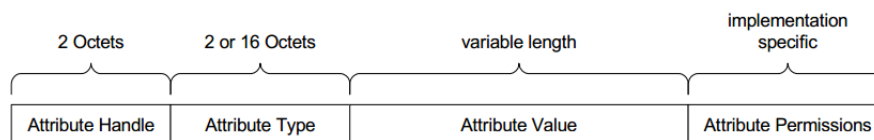


Figure 2.4: Logical Attribute Representation

属性句柄是与明确属性对应的索引。

属性类型是描述属性值的 UUID。

属性值是一个被属性类型描述和被属性句柄索引的数据。

属性以增加属性句柄值的方式安排。属性句柄值可能开始于 0x0001-0xffff 之间的值。即使属性句柄值按增加安排，但是后续的属性句柄相差可能多于 1. 也就是说，在两个连接的属性句柄间有空隙。

属性权限是属性的一部分，不能使用属性协议读取或写。由服务端使用以决定给定的属性的读写访问是否被禁止。属性权限由 GATT 框架建立，如果没有定义，由更高层协议实现。

2.5.2 属性缓冲

属性缓冲允许客户端对发现属性信息的优化，比如服务端一次性使用的属性处理和通过无重搜索重连接使用相同的属性，客户端应该在每一次重连接时重搜索属性信息。使用缓冲，时间被保存且客户端与服务端有意义的包交换数量是不需要的。客户端必须缓存属性信息是所有服务属性和 GATT 服务特征值的属性处理。

服务使用的属性处理不应该随时时间改变。这意味着一旦客户端发现一个属性处理，属性的属性处理将不会改变。

一此情况下可能导致服务端改变用于服务的属性处理，可能的原因是出厂复位或执行固件更新流行。如果服务端的服务可被增加，修改或移除以下是仅有的需求。如果 GATT 在服务端的基础服务在设备的可用生命周期内不能改变，服务改变特性必须不能存在在服务端，且客户端不需要为该服务端在初始化服务发现后执行服务发现。

当服务端支持在 GATT 基础服务中改变时，为了支持缓存，当服务在服务端被增加，移除或修改时，一个指示由服役端发送给客户端。如果在一个服务端属性处理和对应的属性组的绑定被改变，则认为 GATT 基础服务被修改。GATT 服务定义的特征值的改变而不是服务自己改变特征值也被认为是一个修改。

对客户端，和服务端有一个信任的关系，通过连接，属性缓存是可用的。对于有信任关系且没有在连接状态的客户端当服务改变发生时，当客户端重新连接到服务端时，服务端必须发送一个指示。若客户端和服务端没有信任关系，属性缓存只能在连接中可用。当服务端只在当前连接改变发生时，没有信任关系的客户端必须接收一个指示。

注意：如果服役端支持服役改变特性，无信任关系的客户端必须在每一个连接上执行服务发现。

服务端必须发送一个包含有效属性处理的句柄值指示，该属性处理必须在客户端属性缓存中是可用的。属性处理的开始必须是服务端定义的包含改变的属性处理开始，属性处理的结束必须是服务端定义的包含改变的属性处理的最后。指示的值由两个 16 位属性处理串行组成，以指示受影响属性处理的范围。

注意：服役端可能设备受影响属性处理的范围从 0x0001-0xffff 以指示客户端后发现服务端所有的属性处理集合。

一旦接收到包含受影响属性处理范围的句柄值指示，客户端必须认为属性缓存在受影响处理范围是可用的。如果属性处理包含在受影响的属性处理范围，任何未解决的请求传输必须认为是不可用的。客户端

必须在客户端使用任何在受影响属性处理范围内有一个属性的服务前执行服务发现。一旦服务已经接收到句柄值确认，服务端可以认为客户端知道更新属性处理。

客户端必须认为受影响属性处理范围在他的属性缓存是可用的，并执行发现流程以重建属性缓存。服务必须为所有的绑定设备存储服务改变信息。

2.5.3 属性组

GAP 定义了三种属性组类型：

初级服务，次级服务和特征。一个组以一个声明开始，并以 Section 3.1 定义的服务和 Section 3.3 定义的特征结束。不是所有的组属性可以用在以组类型请求的 ATT 读。次级服务和第二级服务组类型可能用在以组类型请求的读。特征组类型必须不能用在组类型请求的 ATT 读。

2.5.4 UUIDs

所有 16 位 UUID 必须包含在 2 个字节内，所有的 128 位 UUID 必须包含在 16 字节内。

2.6 GATT 框架层级

2.6.1 概述

GATT 框架定义了框架数据交换的结构。这个结构定义了在使用框架中基本的元素，比如服务和特征。所有的元素由属性包含。使用在属性协议的属性是搭载这个协议数据的容器。

层级的最高层是一个框架。框架由一个或多个需要满足用户使用情境的服务组成。服务由特征或其它服务的参考组成。每一个特征包含一个值，并可能包含可选的值信息。这个服务和特征的组成部分包含了框架数据并存储在服务端的属性中。

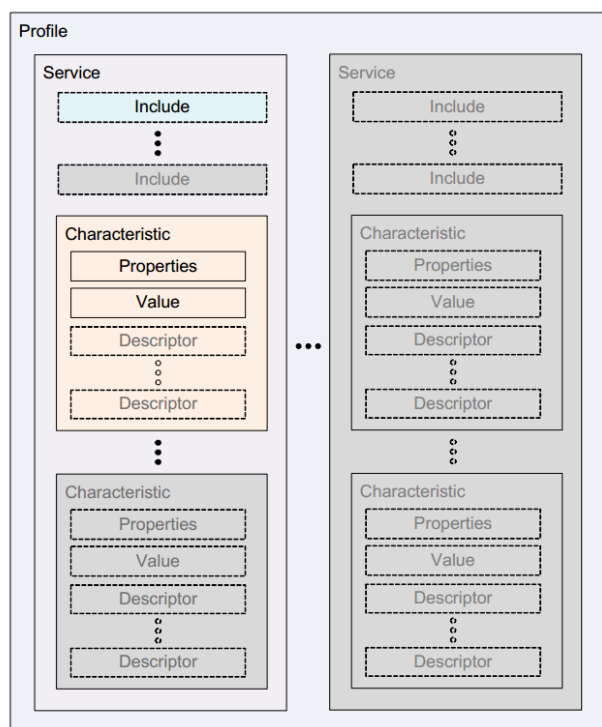


Figure 2.5: GATT Profile hierarchy

2.6.2 服务

一个服务是数据的采集和完成一个特定任务或特征的相关行为。在 GATT，服务由他的服务端定义。服务的定义可能包含了参考服务，强制特征和可选特征。

为了持续向后兼容更早的客户端，新版本的服务可以只增加参考服务和可选服务。从先前服务定义的版本改变行为以定义新版本的服务是禁止的。

有两种类型的服务，主要服务和次要服务。主要服务用于揭示本设备主要可用功能。一个主要服务可以被另外一个服务包含。主要服务可以使用主要服务发现流程发现。次要服务是一种只为了从主要服务或另一个次要服务或其它更高层标准引用的设备。次要服务只与参考它的实体的上下文相关。

一个服务是主要服务还是次要服务的决定权由更高层标准托管。

服务可能用在一个或多个更高层标准，以满足特殊的用户情境。

服务的定义如 Section 3.1。

2.6.3 包括服务

包括服务是另一个现有服务定义一个方法来引用在服务器上定义的服务。为了包括另一个服务，包括定义用于服务定义的开始。当服务定义使用包括定义引用一个包括服务，全部的包括服务定义成为新服务定义的一部分。这包括所有的包括服务和包括服务的特征。包括服务仍然存在为一个独立的服务。一个被其它服务包括的服务不能被包含在内的服务改变也不能被包括服务改变。包括定义的数量或在服务定义内的嵌套包括深度没有限制。

包括定义如 Section 3.2 定义

2.6.4 特征

特征值用于服务属性和如何访问这个值的配置信息和值如何显示和表示的信息。特征定义包括一个特征申明，特征属性和一个值，并可能包括值的描述符或访问特征值的服务许可配置。

特征定义如 Section 3.3 所示。

2.7 配置广播

对于 LE 物理连接，当服务正在执行广播模式流程时，配置广播是客户端表明服务的特征值将在广播数据内广播的方法。

当在广播模式时，服务配置配置特征值为广播。客户端设置 Section 3.3.3.3. 描述的广播配置位为 1。广播频率是服务或特征行为定义的一部分。

3 服务通用性要求

3.1 服务定义

服务定义必须包含服务声明，且可能包含包括定义和特征定义。服务定义在下一个服务声明前或最大属性处理达到之后结束。服务定义出现在属性处理定单中的服务。

所有在服务定义内的包括定义和特征定义被称为服务的一部分。所有包括定义必须立即紧跟服务声明且在任何特征定义之前。服务定义可能没有或多个包括定义，所有特征定义必须立即紧跟最后的包括定义或没有发生包括定义，立即紧跟服务声明。服务定义可能没有或有多个特征定义，包括或特征定义没有上限。服务声明是属性的属性类型设置为主要服务或将要服务的 UUID。属性值必须是 16 位的蓝牙 UUID 或 128 位的服务 UUID，统称为服务 UUID。客户端必须支持 16 位 UUID 和 128 位 UUID 的使用。客户端可能忽略任何未知服务 UUID 的服务定义。未知服务 UUID 是一个不支持服务的 UUID。属性许可必须是只读的，且不行请求认证的授权。

当多个服务存在，使用 16 位蓝牙 UUID 声明的服务定义应该组合在一起，而使用 128 位 UUID 声明的服务应该组合在一起。

Attribute Handle	Attribute Type	Attribute Value	Attribute Permission
0xNNNN	0x2800 – UUID for «Primary Service» OR 0x2801 for «Secondary Service»]	16-bit Bluetooth UUID or 128-bit UUID for Service	Read Only, No Authentication, No Authorization

Table 3.1: Service Declaration

一个设备或高层定义可能有多个服务定义，也可能使用同一个服务 UUID 定义多个服务。所有属性必须包含一个服务或存在于一个服务定义之中。

包含在服务者中的服务定义可能以任意次序出来，客户端不能假设服务端服务定义的次序。

3.2 包括定义

所有包括定义必须只能包含一个包括声明。

包括声明是一个属性类型设置 UUID 为《Include》属性。属性值必须设置为包括服务属性句柄，结束群句柄和服务 UUID 的值。服务 UUID 只能存在于 UUID 是 16 位时。属性权限必须只读明需要认证或授权。

Attribute Handle	Attribute Type	Attribute Value			Attribute Permission
0xNNNN	0x2802 – UUID for «Include»	Included Service Attribute Handle	End Group Handle	Service UUID	Read Only, No Authentication, No Authorization

Table 3.2: Include Declaration

服务者不能包含这样的服务定义：一个带包括定义对其它服务引用原始服务。这将应用每一个服务到包括定义引用，这也称为环形引用。

如果客户端检测到一个环形引用或检测到比其预期的高层包括声明，它将中断 ATT 过程。

3.3 特征定义

特征定义必须包括特征声明和特征值声明且可能包括特征值描述声明。特征定义结束于下一个特征或服务声明的开始或结束于最大属性句柄之后。特征定义出现在以属性句柄排列服务定义的服务者中。

以上每一个声明包含在一个分开的属性中。其中特征声明和属性值声明是必须的。特征值声明必须存在于紧跟特征声明之后。任何可选特征描述声明放在特征值声明之后。可选特征描述表声明的次序没有意义。

特征定义可能定义为连接一些特征值到单个聚合特征值。这可以用于优化通过读写单个聚合特征值以读写多个特征值。这类特征定义与普通特征定义是一样的。特征声明必须使用唯一特征 UUID 以聚合特征定义。聚合特征定义可能包含特征聚合格式化描述表，这个描述表描述了聚合特征值的显示格式化。

3.3.1 特征声明

特征声明是属性类型 UUID 设置为 «Characteristic» 的属性，且属性值设置为特征属性，特征值属性句柄和特征 UUID。属性权限必须为可读且不需要认证或授权。

特征声明属性值在服务端与客户端建立信任关系时不能修改。

Attribute Handle	Attribute Type s	Attribute Value			Attribute Permissions
0xNNNN	0x2803–UUID for «Characteristic»	Characteristic Properties	Characteristic Value Attribute Handle	Characteristic UUID	Read Only, No Authentication, No Authorization

Table 3.3: Characteristic declaration

特征声明的属性值只读：

Attribute Value	Size	Description
Characteristic Properties	1 octets	Bit field of characteristic properties
Characteristic Value Handle	2 octets	Handle of the Attribute containing the value of this characteristic
Characteristic UUID	2 or 16 octets	16-bit Bluetooth UUID or 128-bit UUID for Characteristic Value

Table 3.4: Attribute Value Field in characteristic declaration

服务可能有多个相同特征 UUID 的特征定义。

在一个服务定义中，一些特征可能被强制支持，且这些特征必须位于包括声明之后，任何可选特征之前。客户端不能假设强制支持特征的次序或服务中可选特征的次序。无论什么可能的时候和在要求规定早期，使用 16 位蓝牙 UUID 特征声明的特征定义必须组合在一起，且使用 128 位蓝牙 UUID 特征声明的特征定义必须组合在一起。

3.3.1.1 特征属性

特征属性位域决定了特征值如何被使用，或特征描述表如何被访问。如果表 3.5 定义的位被设置，所描述行为被禁止，多个特征属性可以同时设置。

这些位通过处理这些特征支持设置，不用理会加密要求。

Properties	Value	Description
Broadcast	0x01	If set, permits broadcasts of the Characteristic Value using Characteristic Configuration Descriptor
Read	0x02	If set, permits reads of the Characteristic Value using procedures defined in Section 4.8
Write Without Response	0x04	If set, permit writes of the Characteristic Value without response using procedures defined in Section 4.9.1 .
Write	0x08	If set, permits writes of the Characteristic Value with response using procedures defined in Section 4.9.3 or Section 4.9.4 .
Notify	0x10	If set, permits notifications of a Characteristic Value without acknowledgement using the procedure defined in Section 4.10 .
Indicate	0x20	If set, permits indications of a Characteristic Value with acknowledgement using the procedure defined in Section 4.11 .
Authenticated Signed Writes	0x40	If set, permits signed writes to the Characteristic Value using the procedure defined in Section 4.9.2 .
Extended Properties	0x80	If set, additional characteristic properties are defined in the Characteristic Extended Properties Descriptor defined in Section 3.3.3.1 .

Table 3.5: Characteristic Properties bit field

3.3.1.2 特征值属性句柄

特征值属性句柄域是包含特征值的属性的属性句柄。

3.3.1.3 特征 UUID

特征 UUID 域是一个 16 位的 UUID 或 128 位的 UUID，描述了特征值的类型。客户端必须支持这两种 UUID 形式。客户端可能忽略未知特征 UUID 的特征定义。未知特征 UUID 是不支持的特征 UUID。

3.3.2 特征值声明

特征值声明包括特征的值。它是特征声明后的第一个属性。所有特征定义必须有一个特征值声明。特征值声明是属性类型设置为在特征声明中使用特征值为 16 位 UUID 或 128 位 UUID 的属性。属性值设置为特征值，属性权限由服务或其它定义确定。

Attribute Handle	Attribute Type	Attribute Value	Attribute Permissions
0xNNNN	0xuuuu – 16-bit Bluetooth UUID or 128-bit UUID for Characteristic UUID	Characteristic Value	Higher layer profile or implementation specific

Table 3.6: Characteristic Value declaration

3.3.3 特征描述声明

特征描述通常包含与特征值相关的信息，GATT Profile 定义了标准的特征描述集合，可由更高层 profile 使用。更高层 profile 可能定义了额外的特征描述。每一个特征描述通过特征描述 UUID 识别。客户端应该支持 16 位或 128 位的特征描述 UUID。客户端忽略任何未知特征描述 UUID 的特征描述。未知特征描述 UUID 是不支持的特征描述的 UUID。

如果特征描述存在于特征定义中，则必须紧跟特征值声明后。特征描述声明可能在特征定义中以任意次序出现。客户端不能假设特征描述声明出现的次序。

特征描述声明许可由高层协议定义或单独定义。客户端不能假设所有特征描述是可读的。

3.3.3.1 特征扩展属性

特征扩展属性声明是定义客户特征属性的描述。如果特征属性的扩展特征位置位，则特征描述必须存在。特征描述可能在现在特征值后的特征定义中的任意位置。在一个特征定义中，只能有一个扩展属性声明存在。

特征描述包含在属性中，且属性类型 UUID 设置为«Characteristic Extended Properties»，且属性值必须为特征扩展属性位域。属性许可必须可读，不需要认证和授权。

Attribute Handle	Attribute Type	Attribute Value	Attribute Permissions
0xNNNN	0x2900 – UUID for «Characteristic Extended Properties»	Characteristic Extended Properties Bit Field	Read Only, No Authentication, No Authorization

Table 3.7: Characteristic Extended Properties declaration

特征扩展属性位域描述了额外属性，如如何使用特征值或如果访问特征描述。如果表 3.8 定义的位置位，相就的行为被禁止。多个特征属性可能被设置。

Properties	Value	Description
Reliable Write	0x0001	If set, permits reliable writes of the Characteristic Value using the procedure defined in Section 4.9.5 .
Writable Auxiliaries	0x0002	If set, permits writes to the characteristic descriptor defined in Section 3.3.3.2
Reserved for Future Use	0xFFFC	Reserved for Future Use

Table 3.8: Characteristic Extended Properties bit field

3.3.3.2 特征用户描述

特征用户描述声明是一个可先的特征描述，包含 UTF-8 字符串，该字符串是特征值描述用的用户文本。如果特征属性的可写辅助位置 1，则该特征描述可被写。特征描述可能出现在特征值后的物定义内的任何位置。在一个特征定义中只能一个特征用户描述声明。

特征描述包含属性类型 UUID 为«Characteristic UserDescription» 的属性，且属性值必须设置为用户描述 UTF-8 字符串。属性权限由 profile 定义或可能单独定义。

Attribute Handle	Attribute Type	Attribute Value	Attribute Permissions
0xNNNN	0x2901 – UUID for «Characteristic User Description»	Characteristic User Description UTF-8 String	Higher layer profile or implementation specific

Table 3.9: Characteristic User Description declaration

3.3.3.3 客户端特征配置

客户端特征配置声明是一个可选的特征描述，定义了特征可以如何被定义的客户端配置。客户特征配置描述值必须在绑定设备间保持。客户特征配置描述值必须在第一个未绑定设备连接中设置为初始值。特征描述值是一个位域，当位置位，相应的动作将被使能，否则将不能使用。客户端特征配置描述可能出现在特征值扣的特征定义中的任意位置。在一个特征定义中，只有一个客户端特征配置声明存在。

客户端可能写配置描述以控制客户端服务上的特征的配置。每一个客户端有他自己的客户端特征配置实例。客户端特征配置的读只显示客户端的配置且并只写客户端的影响配置。验证和授权可能被服务者要求以写配置描述。客户特征配置声明必须可读写。

特征描述包含在属性中。属性类型 UUID 必须设置为«Client Characteristic Configuration»。属性值必须设置为特征描述值。属性许可由 profile 定义也可能单独实现。

下面定义了客户端特征配置位，缺省的客户端特征配置描述值是 0x0000。

Attribute Handle	Attribute Type	Attribute Value	Attribute Permissions
0xNNNN	02902 – UUID for «Client Characteristic Configuration»	Characteristic Configuration Bits	Readable with no authentication or authorization. Writable with authentication and authorization defined by a higher layer specification or is implementation specific.

Table 3.10: Client Characteristic Configuration declaration

Configuration	Value	Description
Notification	0x0001	The Characteristic Value shall be notified.
Indication	0x0002	The Characteristic Value shall be indicated.
Reserved for Future Use	0xFFFF4	Reserved for future use.

Table 3.11: Client Characteristic Configuration bit field definition

3.3.3.4 服务端特征配置

服务端特征配置声明是可选特征描述，定义了如何为服务端配置特征。特征描述值是一个位域，当位置位，相应的动作被使能，否则将不使用。服务特征配置描述可能出现在特征值后的特征定义中的任何位置。在特征定义中只能有一个服务特征配置声明。服务特征配置声明必须可读写。

客户端可能写这个配置描述以控制服务端上的特征配置。对所有的客户端只有一个单独的服务特征配置实例。服务特征的读显示所有客户端的配置，而写影响所有客户端的配置。服务端可能要求认证和授权以写配置描述。

特征描述包含在属性中，属性类型 UUID 设置为 «Server Characteristic Configuration»。属性值必须设置为特征描述值，属性许可由 profile 定义或可能单独实现。

下面定义了服务特征配置位：

Attribute Handle	Attribute Type	Attribute Value	Attribute Permissions
0xNNNN	0x2903 – UUID for «Server Characteristic Configuration»	Characteristic Configuration Bits	Readable with no authentication or authorization. Writable with authentication and authorization defined by a higher layer specification or is implementation specific.

Table 3.12: Server Characteristic Configuration declaration

Configuration	Value	Description
Broadcast	0x0001	The Characteristic Value shall be broadcast when the server is in the broadcast procedure if advertising data resources are available.
Reserved for Future Use	0xFFFF2	Reserved for future use.

Table 3.13: Server Characteristic Configuration bit field definition

3.3.3.5 特征描述格式

特征描述格式化声明是一个可选的特征描述，定义了特征值的格式化。特征描述可能出现在特征值后的特征定义的任何位置。如果多于一个特征描述格式化声明存在，在一个特征定义中，则特征聚合格式化声明将存在并做为特征定义的一个部分。

特征格式化值由五个部分组成：格式化，指数，单元，命名空间和描述。

特征描述包含在属性中，属性类型 UUID 设置为 «Characteristic Format»。属性值必须设置为特征描述值。属性许可必须只读且不需要认证或授权。

Attribute Handle	Attribute Type	Attribute Value					Attribute Permissions
0xNNNN	0x2904 – UUID for «Characteristic Format»	Format	Exponent	Unit	Name Space	Description	Read only No Authentication, NO authorization

Table 3.14: Characteristic Format declaration

特征描述格式化属性值域定义如下：

Field Name	Value Size	Description
Format	1 octet	Format of the value of this characteristic.
Exponent	1 octet	Exponent field to determine how the value of this characteristic is further formatted.
Unit	2 octets	The unit of this characteristic as defined in [1]
Name Space	1 octet	The name space of the description as defined in [1]
Description	2 octets	The description of this characteristic as defined in a higher layer profile.

Table 3.15: Characteristic Format Value definition

3.3.3.5.1 位序

用于特征格式化描述的位序必须为小端模式。

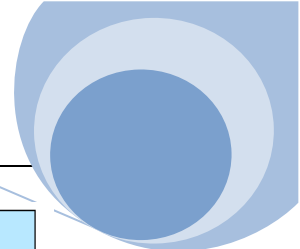
3.3.3.5.2 格式化

格式化域定义了包含在特征值中的单独的值是什么样的格式化。如果格式化不是整数字节，则数据必须包含值的最低有效位，而其它位设置为 0。如果特征值小于一个字节，它占用一个完整字节。

如果下格式化被定义：

Format	Short Name	Description	Exponent Value
0x00	rfu	Reserved for Future Used	No
0x01	boolean	unsigned 1-bit; 0 = false, 1 = true	No
0x02	2bit	unsigned 2-bit integer	No
0x03	nibble	unsigned 4-bit integer	No
0x04	uint8	unsigned 8-bit integer	Yes
0x05	uint12	unsigned 12-bit integer	Yes
0x06	uint16	unsigned 16-bit integer	Yes
0x07	uint24	unsigned 24-bit integer	Yes
0x08	uint32	unsigned 32-bit integer	Yes
0x09	uint48	unsigned 48-bit integer	Yes
0x0A	uint64	unsigned 64-bit integer	Yes

Table 3.16: Characteristic Format types



Format	Short Name	Description	Exponent Value
0x0B	uint128	unsigned 128-bit integer	Yes
0x0C	sint8	signed 8-bit integer	Yes
0x0D	sint12	signed 12-bit integer	Yes
0x0E	sint16	signed 16-bit integer	Yes
0x0F	sint24	signed 24-bit integer	Yes
0x10	sint32	signed 32-bit integer	Yes
0x11	sint48	signed 48-bit integer	Yes
0x12	sint64	signed 64-bit integer	Yes
0x13	sint128	signed 128-bit integer	Yes
0x14	float32	IEEE-754 32-bit floating point	No
0x15	float64	IEEE-754 64-bit floating point	No
0x16	SFLOAT	IEEE-11073 16-bit SFLOAT	No
0x17	FLOAT	IEEE-11073 32-bit FLOAT	No
0x18	duint16	IEEE-20601 format	No
0x19	utf8s	UTF-8 string	No
0x1A	utf16s	UTF-16 string	No
0x1B	struct	Opaque structure	No
0x1C – 0xFF	rfu	Reserved for Future Use	No

Table 3.16: Characteristic Format types

当编码为 IPV4 地址，将使用 uint32 格式。

当编码为 IPV6 地址，将使用 uint128 格式。

当编码为蓝牙地址，将使用 uint48 格式。

一个 duint16 是两个 uint16 值串联起来。

3.3.3.5.3 指数

指数域为整数数据，用于决定数据进一步格式化。扩展域只使用表 3.16 所示的整数格式化类型。扩展域是一个有符号整数。

$$\text{actual value} = \text{Characteristic Value} * 10^{\text{Exponent}}$$

如上式所知，实际值是特征值和 10 的 exponent 次方的乘积。这也称为定点数。

比如，如果指数为 2，特征值为 23，则实际值为 2300。

比如，如果指数是-3，特征值为 3892，则实际值为 3.892

3.3.3.5.4 单元

单元是 Assigned Numbers document 定义的 UUID。

3.3.3.5.5 命名空间

命令空间域用于决定 Assigned Numbers document 定义的组织，负责定义描述字段的枚举。

3.3.3.5.6 描述

描述是 Assigned Numbers document 定义的枚举值，通过命令空间域识别。

3.3.3.5.6 特征聚合格式

特征聚合格式化声明是可选 特征描述，定义了聚合特征值的格式化。

特征描述可能出现在特征值后的特征定义的任何位置。在一个特征定义中只能有一个特征聚合格式化声明。

特征聚合格式化值由特征描述模式声明属性句柄列表组成，每一个属性句柄指向对应的特征描述格式化声明。

属性许可必须只读，并不需要认证或授权。

属性句柄列表是多个 16 位属性句柄值的串联，组成一个单独的属性值。列表必须包含至少两个特征描述格式化声明的属性句柄。特征值必须被属性句柄指向的每一个特征描述格式化声明分离。属性句柄的次序是有定义的。

Attribute Handle	Attribute Type	Attribute Value	Attribute Permissions
0xNNNN	0x2905 – UUID for «Characteristic Aggregate Format»	List of <i>Attribute Handles</i> for the Characteristic Presentation Format Declarations	Read only No authentication No authorization

Table 3.17: Characteristic Aggregate Format declaration

如果多于一个特征描述格式化声明存在于一个特征定义中，必须对应有一个特征聚合 格式化声明。在属性句柄列表中，特征聚合格式化声明必须包括每一个特征描述格式化声明在特征定义中。来自其它特征定义的特征描述格式化声明也被使用。

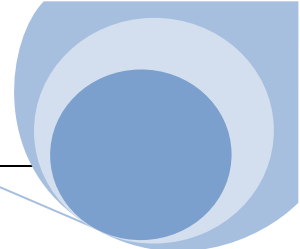
特征描述格式化声明可能存在于没有特征描述格式化声明的特征定义中。特征聚合格式化声明可能使用来自其它特定定义的特征描述格式化声明。

3.4 GATT profile 属性类型总结

下面的表格总结了 GATT profile 定义的属性类型

Attribute Type	UUID	Description
«Primary Service»	0x2800	Primary Service Declaration
«Secondary Service»	0x2801	Secondary Service Declaration
«Include»	0x2802	Include Declaration
«Characteristic»	0x2803	Characteristic Declaration
«Characteristic Extended Properties»	0x2900	Characteristic Extended Properties
«Characteristic User Description»	0x2901	Characteristic User Description Descriptor
«Client Characteristic Configuration»	0x2902	Client Characteristic Configuration Descriptor
«Server Characteristic Configuration»	0x2903	Server Characteristic Configuration Descriptor
«Characteristic Format»	0x2904	Characteristic Format Descriptor
«Characteristic Aggregate Format»	0x2905	Characteristic Aggregate Format Descriptor

Table 3.18: Summary of GATT Profile Attribute types



4、GATT 特征要求

4.1 概述

在 GATT profile 中有 11 个特征定义

1. Server Configuration
2. Primary Service Discovery
3. Relationship Discovery
4. Characteristic Discovery
5. Characteristic Descriptor Discovery
6. Reading a Characteristic Value
7. Writing a Characteristic Value
8. Notification of a Characteristic Value
9. Indication of a Characteristic Value
10. Reading a Characteristic Descriptor
11. Writing a Characteristic Descriptor

每一个特征映射到对应的处理流程和子处理流程。这些处理流程和子处理流程描述了如何使用属性协议完成对应的特征。

4.2 特征支持和处理映射

下表映射每一个特征到对应的使用这些特征的处理流程，并指示了这些处理是否为可选或强制的。这些处理在对应的参考章节中描述。

Item No.	Feature	Sub-Procedure	Ref.	Support in Client	Support in Server
1	Server Configuration	Exchange MTU	4.3.1	O	O
2	Primary Service Discovery	Discover All Primary Services	4.4.1	O	M
		Discover Primary Services By Service UUID	4.4.2	O	M
3	Relationship Discovery	Find Included Services	4.5.1	O	M

Item No.	Feature	Sub-Procedure	Ref.	Support in Client	Support in Server
4	Characteristic Discovery	Discover All Characteristic of a Service	4.6.1	O	M
		Discover Characteristic by UUID	4.6.2	O	M
5	Characteristic Descriptor Discovery	Discover All Characteristic Descriptors	4.7.1	O	M
6	Characteristic Value Read	Read Characteristic Value	4.8.1	O	M
		Read Using Characteristic UUID	4.8.1	O	M
		Read Long Characteristic Values	4.8.2	O	O
		Read Multiple Characteristic Values	4.8.3	O	O
7	Characteristic Value Write	Write Without Response	4.9.1	O	C.1
		Signed Write Without Response	4.9.2	O	O
		Write Characteristic Value	4.9.3	O	C.2
		Write Long Characteristic Values	4.9.4	O	O
		Characteristic Value Reliable Writes	4.9.5	O	O
8	Characteristic Value Notification	Notifications	4.10.1	O	O
9	Characteristic Value Indication	Indications	4.11.1	M	C3
10	Characteristic Descriptor Value Read	Read Characteristic Descriptors	4.12.1	O	O
		Read Long Characteristic Descriptors	4.12.2	O	O
11	Characteristic Descriptor Value Write	Write Characteristic Descriptors	4.12.3	O	O
		Write Long Characteristic Descriptors	4.12.4	O	O

Table 4.1: GATT feature mapping to procedures

Item No.	Feature	Sub-Procedure	Ref.	Support in Client	Support in Server
C1: Write Without Response is mandatory if Signed Write Without Response is supported otherwise optional C2: Write Characteristic Value is mandatory if Write Long Characteristic Values is supported otherwise optional C3: If <i>Service Change Characteristic</i> is present, this feature is mandatory, otherwise optional.					

Table 4.1: GATT feature mapping to procedures

4.3 服务配置

这个过程由客户端使用，以配置属性协议。这个过程有一个子过程，用于设置 MTU 大小。

4.3.1 交换 MTU

当客户端支持的 MTU 大于默认的 ATT_MTU 时，该子过程由客户端使用以设置 ATT_MTU 至双方都支持的最大可能的值。这个子过程必须在连接中只初始化一次。

这个子过程禁止在 BR/EDR 物理连接中使用，因为其 MTU 大小使用 L2CAP 信道配置过程协商确定。

该子过程使用属性协议交换 MTU 请求。客户端接收 MTU 参数必须设置为客户端能接收的最大 MTU。

服务端有两个可能的应答可以被发送，以应答交换 MTU 请求：交换 MTU 应答和错误应答。

如果客户端发生错误，错误应答将返回。

服务端必须以交换 MTU 应答回答这个消息，应答中的服务接收 MTU 参数设置为服务端能接收的最大 MTU 值。

如果服务端发送错误应答，应答中会带有请求不支持，属性代码不支持和缺省 MTU 等之一。

一旦消息被交换，ATT_MTU 必须设置为客户端接收 MTU 和服务端接收 MTU 值最小的那个。

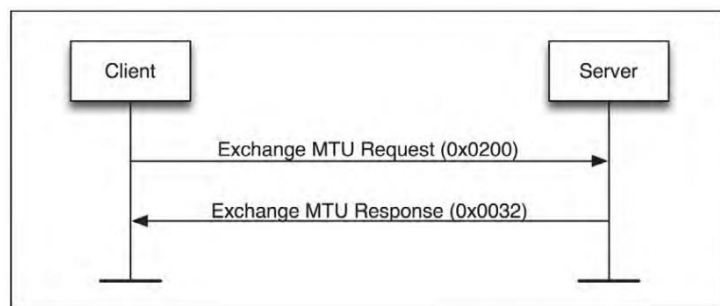


Figure 4.1: Exchange MTU

举例，如图 4.1，ATT_MTU 最终值设置为 0x0032。

4.4 根服务发现

这个过程由客户端使用，以发现服务端的根服务。一旦根服务被发现，可以使用其它处理来访问根服务的其它信息，包括特征发现和关系发现以找到其它相关的根服务和次服务。

有两个子处理可以用于根服务的发现，发现所有根服务和使用服务 UUID 发现根服务。

通过 BR/EDR 传输的 GATT 服务发现将列出运行在 LE 传输的服务，因此，在 BR/EDR 上必须使用 SDP 服务发现。

4.4.1 发现所有根服务

该子处理由客户端使用以发现服务端的所有根服务。

该协议将使用以组类型读请求，其属性类型参数设置为根服务的 UUID。起始句柄为 0x0001, 线束句柄设置为 0xFFFF。

服务端会发送两个可能的应答：以组类型读应答和错误应答。

如果服务端发生错误，将发送错误应答。

以组类型读应答返回一个属性句柄列表，结束组句柄和服务端支持的属性值数组。应答中的每一个属性值是服务支持的一个服务 UUID。属性句柄是服务声明的句柄。结束组句柄是服务定义中最后一个属性的句柄。以组类型读请求必须再次被调用，其开始句柄设置为大于以组类型读应答中的最后一个结束组句柄。

当接收到错误应答接收到且错误代码设置为属性未发现时，该子处理完成。

在发现所有服务端的根服务前，如果一个预期的根服务被发现，禁止过早结束该处理。

注意：在 3.1 章节中描述的服务声明定义了服务声明是可读的，且不需要认证或授权，因此，缺少认证或禁止读错误不会发生。

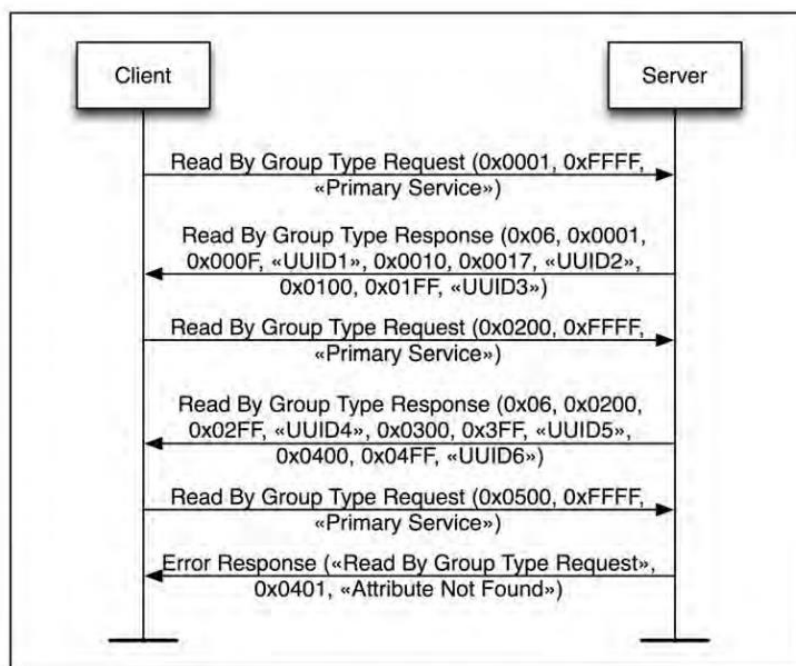


Figure 4.2: Discover All Primary Services example

4.4.2 通过服务 UUID 发现根服务

当服务 UUID 是已知时，该子处理由客户端使用以发现服务端特定的根服务。特别的，一个服务端可能有多个根服务。根服务通过服务 UUID 识别并发现。

以类型值发现请求属性协议将被使用，其属性类型参数设置为根服务 UUID，且属性值设置为 16 位蓝牙 UUID 或 128 位 UUID 的特定根服务。起始句柄设置为 0x0001，结束句柄设置为 0xFFFF。

服务端可能回应两个可能的应答：以类型值发现应答和出错应答。

以类型值发现应答返回一个属性句柄范围列表。属性句柄范围从服务定义的开始句柄到结束句柄。如果被搜索的服务 UUID 属性句柄范围被返回，且结束句柄不为 0xFFFF，以类型值发现请求可能再一次被调用，此时其起始句柄设置为大于以类型值发现应答中最后一个属性句柄范围的值。

当接收到出错应答且错误代码为属性未发现时，该子过程完成。

在发现服务端所有由服务 UUID 决定的根服务之前，如果发现了期望的根服务，禁止过早结束该子过程。

注意：章节 3.1 描述的服务声明决定了服务声音是可读的，并不需要认证或授权，因此，认证不足或禁止读错误不会发生。

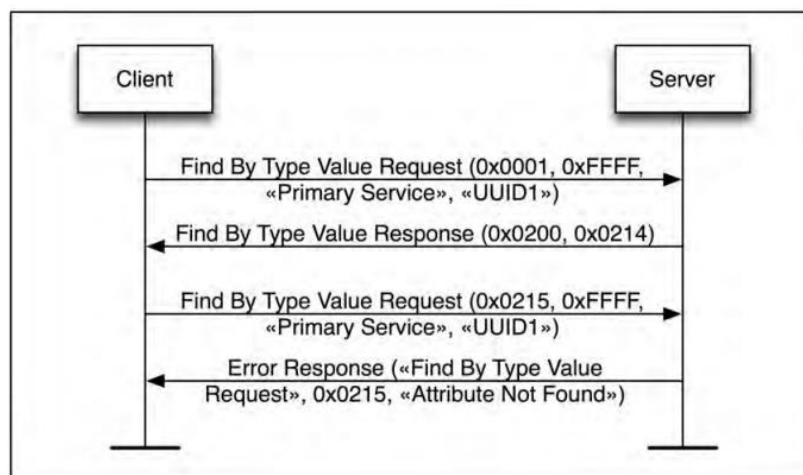


Figure 4.3: Discover Primary Service by Service UUID example

4.5 关系发现

该处理由客户端使用，以发现服务与其它服务的关系。

有一个子处理可用于关系发现：发现包括服务。

4.5.1 发现包括服务

该子处理由客户端使用，以发现服务端服务定义中的包括服务声明。服务定义由服务句柄范围分别。

以类型 读请求属性协议将被使用，其属性类型参数设置为《Include》UUID，起始句柄设置为确定服务的开始句柄，结束句柄尾市为确定服务的结束句柄。在发现服务端支持的所有服务的包括服务前，如果发现期望的包括服务，禁止过早结束该过程。

从服务端可返回两个以类型读请求的应答：以类型读应答和出错应答。

以类型读应答返回一个属性句柄集合，属性值对对应于服务定义中的包括服务。每一个应答中的属性值由包括服务声明的属性句柄和结束组句柄组成。如果服务 UUID 是 16 位蓝牙 UUID，UUID 也会在应答中被返回。以类型读请求必须再交被调用，其起始句柄设置为大于以类型读应答中最后一个属性句柄。

当接收到错误代码为属性未发现的出错应答或以类型读应答中包括服务属性句柄等于请求的结束句柄时，该过程完成。

当包括服务是 128 位 UUID 时，为了获得其包括服务 UUID，读请求被使用。读请求的属性句柄为包括服务的属性句柄。

注意：章节 3.2 中包括声明描述决定了包括声明是可读的且不需要认证或授权，因此认证不足或禁止读错误不会发生。

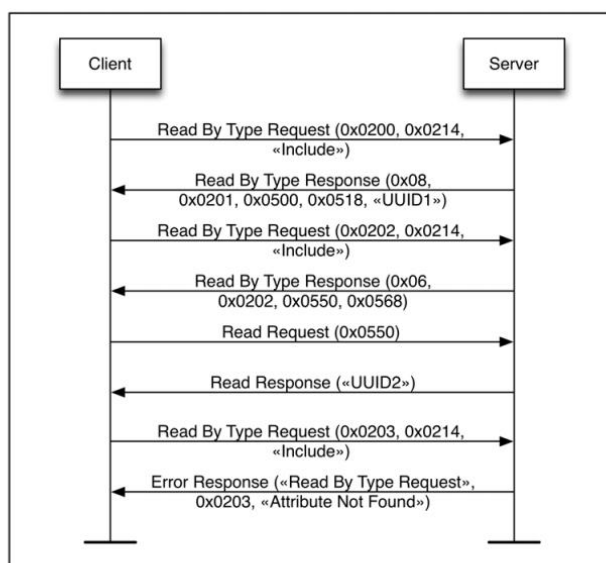


Figure 4.4: Find Included Services example

4.6 特征发现

该过程由客户端使用，以发现服务端的服务特征。一旦特征被发现，可以使用其它处理以发现特征相关的其它信息。

有两个子处理可以用于特征发现：发现所有特征及以 UUID 发现特征。

4.6.1 发现服务所有特征

当服务句柄范围已知时，该过程由客户端使用，以发现服务定义中的所有特征声明。服务定义通过服务句柄范围区分。

以类型读请求属性协议将被使用，其属性类型参数设置为《Characteristic》UUID. 起始句柄必须设置为确定服务的的开始句柄，结束句柄必须设置为确定服务的结束句柄。

服务端可能返回两个以类型读请求应答：以类型读应答和出错应答。

如果服务端发生错误，将返回出错应答。

以类型读应答返回一个属性句柄列表，属性值对对应服务定义的特征。属性句柄是特征声明的句柄。属性值是特征属性，特征值句柄和特征 UUID。以类型读请求必须再调被调用，其开始句柄设置为大于以类型读应答中的最后一个属性句柄。

当接收到错误代码设置为属性未发现的出错应答或以类型读应答中的属性句柄等于请求的结束句柄时，该子处理完成。

在发现服务端支持的所有服务的特征前，如果期望的特征已被发现，仍禁止过早结束子处理。

注意：章节 3.3 描述的特征声明决定了特征声明是可读的，且不需要认证或授权，因此，认证不足或禁止读错误不会发生。

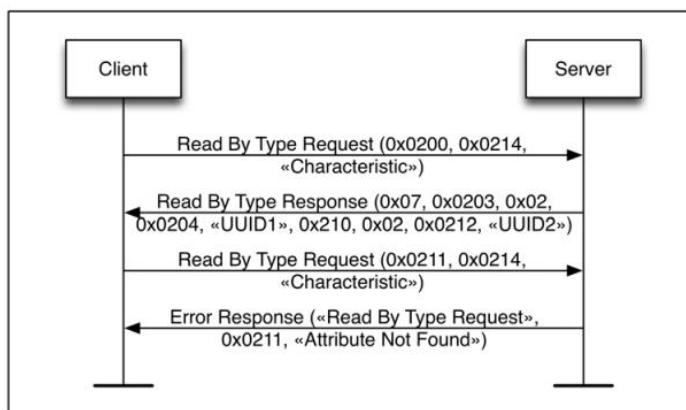


Figure 4.5: Discover All Characteristics of a Service example

未完待续。。。。。

联系我们

深圳市奥格无线科技有限公司

<http://rf-smart.taobao.com>

QQ: 375797387

手机: 15919408243