GAP

1 扫描和广播

1.2 函数作用

Hci\_core.c

**bt\_le\_adv\_start()**：设置正在广播标志（bt\_dev.flags），设置广播数据（set\_ad），若为可连接广播，则设置扫描响应数据（set\_ad），创建设置广播参数命令，设置广播参数。若为可连接广播且使能了蓝牙隐私（CONFIG\_BT\_PRIVACY），则设置私有地址（le\_set\_private\_addr），否则将设备的身份地址（bt\_dev.id\_addr.a）设置为设备的随机地址。若设备使用了静态随机地址，则将其设置为设备的随机地址。若为不可连接广播且参数地址（param->own\_addr）为NRPA（不可解析私有地址），则将其设置为设备的随机地址，否则设置私有地址。发送设置广播参数HCI命令，开启广播（set\_advertise\_enable）。若广播参数（param->options）中指定了只广播一次，则设备建立连接后不会重新开始广播。

**le\_set\_private\_addr()**：若RPA（可解析私有地址）无效，则创建RPA（bt\_rpa\_create）并将其设置为设备的随机地址（set\_random\_address），设置RPA有效标志。开启RPA更新定时器（提交一个延迟工作项，延迟时间是900秒，即每隔15分钟更新一次随机地址）。

**bt\_le\_scan\_start()**：若正在扫描，则停止扫描（bt\_hci\_stop\_scanning）。然后开始扫描（start\_le\_scan），并设置扫描发现设备回调函数（scan\_dev\_found\_cb）。

**start\_le\_scan()**：设置扫描参数（间隔、扫描窗、过滤策略、地址），发送设置扫描参数HCI命令。发送扫描使能HCI命令，若成功则设置正在扫描标志。

**le\_adv\_report()**：首先获取报告数量（num\_reports），然后循环处理报告数据。从数据缓冲中取出广播信息部分（共9字节，包括广播事件类型、广播地址类型、广播地址和数据长度，详见Spec Vol2 PartE 7.7.65.2）。获取RSSI，根据广播设备地址查找可用的身份地址（find\_id\_addr）？。若发现设备回调函数不为空，则调用它。检查是否有待处理的连接（check\_pending\_conn）。

**注意**：协议栈建立连接的步骤是先创建一个连接，然后将其设置为连接扫描状态（CONNECT\_SCAN），这表示此连接是待处理的（pending），之后开启被动广播，此时控制器会产生广播报告事件，在广播报告事件中再处理待定的连接（即发送连接请求）。

之所以这么做，我想是因为要确保连接的另一方处于广播状态。

**check\_pending\_conn()**：若为显示扫描（主动扫描），则不允许连接。若广播是不可连接的，则返回。根据广播设备的身份地址查找待处理的连接（bt\_conn\_lookup\_state\_le），若找到连接，则停止扫描。若使能了蓝牙隐私，则设置私有地址，否则若使用了静态地址，则将其设置为随机地址。创建连接（hci\_le\_create\_conn），将连接设置为连接状态（CONNECT），实际上就是提交了一个延迟工作项来更新连接。

**注意**：这里只是设置了本机的地址类型（Own\_Address\_Type），而实际上地址的设置（公共、静态或RPA）是由LL层完成的，在处理HCI命令的时候。

**hci\_le\_create\_conn()**：设置连接参数（见Spec Vol2 PartE 7.8.12），发送创建连接HCI命令。

**Hci.c->le\_create\_connection()**：LL层创建连接函数。

**le\_conn\_complete()**：若命令执行错误，则找到处于连接状态的连接（只能有一个连接处于连接状态），将其设置为断开连接状态（DISCONNECTED）。获取对方设备的身份地址（因为LL层可能解析了对方的RPA），找到处于连接状态的连接。若为从角色，则添加新的连接（bt\_conn\_add\_le）。设置连接的相关信息（handle、le.dst、interval、latency、timeout、role）。若为从角色，则设置连接双方的地址，若设置了保持广播标志（是否只广播一次），则使能广播。设置连接状态为已连接（CONNECTED）。若为主角色，则读取远程设备的特性（hci\_le\_read\_remote\_features）。若支持2M物理层，则设置物理层（hci\_le\_set\_phy）。若支持数据长度，则设置数据长度（hci\_le\_set\_data\_len）。更新连接参数（update\_conn\_param），更新扫描。

**bt\_conn\_set\_state()**：若状态为已连接，则初始化发送队列（conn->tx\_queue、conn->tx\_notify），初始化连接通道（conn->channels），L2CAP连接（bt\_l2cap\_connected），调用已连接回调函数（cb->connected）。

**bt\_l2cap\_connected()**：遍历LE通道链表（le\_channels），调用通道的接受函数（fchan->accept），将通道添加到连接中（l2cap\_chan\_add），调用通道的连接函数（chan->ops->connected）。

注意：le\_channels的初始化是在蓝牙初始化的时候进行的，共添加了ATT、SMP和LE\_SIG三个固定通道。

MASTER

SLAVE

开始扫描

bt\_le\_scan\_start()

开始广播

bt\_le\_scan\_start()

设置扫描参数并使能扫描

start\_le\_scan()

处理LE广播报告事件

le\_adv\_report()

开始扫描

bt\_le\_scan\_start()

调用扫描回调函数

scan\_dev\_found\_cb()

2 建立连接

要理解LE建立连接的过程，就要理解HCI的事件和命令。

2.1当扫描者发起主动扫描时，若有广播者作出响应，则扫描者会收到控制器生成的LE Advertising Report事件，事件中包含了广播者的扫描响应数据。当扫描者发起被动扫描时，若有广播者在进行广播，则扫描者会收到LE Advertising Report事件，事件中包含广播者的广播数据。

2.2当扫描者发出连接请求时，扫描者和广播者都会收到控制器生成的LE Connection Complete事件，此事件向形成连接的两台主机表示一个新连接已经创建，若连接成功，控制器会分配一个连接句柄（handle），利用这个句柄用来标识一个连接。若连接建立失败，则只有扫描者会收到此事件。

2.3 函数作用

**bt\_conn\_create\_le()**：判断连接参数是否为空，判断是否为显示扫描状态。先根据对等设备地址（peer）查找对应的连接，若连接不为空，则当连接状态为连接扫描时，设置连接参数（bt\_conn\_set\_param\_le）并返回；当连接状态为正在连接或已连接时，直接返回。若连接为空，则添加新的连接（bt\_conn\_add\_le）。复制对等设备地址，设置连接参数，设置连接状态为连接扫描（CONNECT\_SCAN），扫描更新（bt\_le\_scan\_update）。

**bt\_conn\_add\_le()**：获取新连接（conn\_new），设置目标地址（le.dst），设置安全等级（SECURITY\_LOW），设置连接类型（TYPE\_LE），设置连接间隔，初始化连接参数更新（le\_conn\_update）工作项。

**conn\_new()**：遍历连接数组（conns），若连接的引用为0，说明连接未被使用，将连接清空，设置连接的引用为1，返回连接指针。

bt\_le\_scan\_update()：若设备正在扫描，则停止扫描。找到处于扫描状态的连接，将其引用减1（在设置连接为扫描状态时已将连接引用加1）。若为快速扫描，则设置扫描间隔为60ms，扫描窗为30ms。开始被动扫描（start\_le\_scan）。

bt\_conn\_disconnect()：若为主机发起断开连接，则禁止自动连接。当连接处于扫描状态时，直接将连接设置为断开连接状态。当连接处于正在连接状态时，则发送取消创建连接HCI命令。当连接处于已连接状态时，发送断开连接HCI命令（OP\_DISCONNECT）。

MASTER

SLAVE

创建LE连接

bt\_conn\_create\_le()

开始广播

bt\_le\_adv\_start()

设置广播参数

set\_ad()

创建一个新连接

bt\_conn\_add\_le()

开启被动扫描

bt\_le\_scan\_update()

开启被动扫描

bt\_le\_scan\_update()

使能广播

set\_advertise\_enable()

处理广播报告事件

le\_adv\_report()

检查待定的连接

check\_pending\_conn()

发送连接请求

hci\_le\_create\_conn()

更新连接参数

k\_delayed\_work\_submit()

处理连接请求

bt\_le\_scan\_start()

ADV\_IND

CONN\_REQ

处理连接完成事件

le\_conn\_complete()

处理连接完成事件

le\_conn\_complete()

1 初始化

**bt\_enable()**：若蓝牙设备未使能，则设置使能标志。设置设备就绪回调函数。创建HCI发送线程（hci\_tx\_thread），创建HCI接收线程（hci\_rx\_thread）。调用蓝牙设备驱动的打开函数（bt\_dev.drv->open）。进行蓝牙初始化（bt\_init），若回调不为空，则调用回调函数。

**bt\_init()**：HCI初始化（hci\_init）。若使能了蓝牙连接，则进行蓝牙连接初始化（bt\_conn\_init）。若使能了蓝牙隐私，则初始化IRK（irk\_init）。初始化RPA更新工作项（rpa\_timeout），扫描更新。

**hci\_init()**：共同部分初始化（common\_init），LE初始化（le\_init），设置事件掩码（set\_event\_mask）。若没有公共地址，则设置为静态地址（set\_static\_addr）。显示设备信息（show\_dev\_info）。

**common\_init()**：发送重启控制器HCI命令，发送读取本地设备特性HCI命令，将结果复制到蓝牙设备特性（bt\_dev.features）。发送读取本地版本信息HCI命令，将结果复制到蓝牙设备。发送读取蓝牙地址HCI命令，将结果复制到蓝牙设备的身份地址。发送读取支持命令HCI命令，将结果复制到蓝牙设备。随机数初始化（prng\_init），设置流控制（set\_flow\_control）。

**bt\_conn\_init()**：蓝牙ATT初始化（bt\_att\_init），蓝牙SMP初始化（bt\_smp\_init），蓝牙L2CAP初始化（bt\_l2cap\_init），遍历连接链表，若连接设置了自动连接标志，则将其设置为连接扫描状态（这样一旦指定的设备开始广播就能自动连接）。

**bt\_att\_init()**：定义ATT固定通道（CID\_ATT、bt\_att\_accept），注册固定通道（bt\_l2cap\_le\_fixed\_chan\_register），蓝牙GATT初始化（bt\_gatt\_init）。

bt\_gatt\_init()：

**bt\_smp\_init()**：定义SMP固定通道（CID\_SMP、bt\_smp\_accept），定义公钥回调（bt\_smp\_pkey\_ready），是否支持安全连接（le\_sc\_supoorted），注册固定通道，生成公钥（bt\_pub\_key\_gen），测试加密函数（smp\_self\_test）。

**bt\_l2cap\_init()**：定义L2CAP固定通道（CID\_LE\_SIG、l2cap\_accept），注册固定通道。

3 接收数据

在Hci\_core.c文件中，创建了接收数据线程hci\_rx\_thread来处理接收到的数据，在这个处理函数中，只包含了hci\_acl()和hci\_event()两个处理方法，分别用来处理来自外部设备发送的数据和来自内部控制器的事件。？在建立连接之前如广播、扫描、连接等请求包如何处理，在哪里处理，以及LL\_Control\_PDU。