# 一、蓝牙学习一（简介）

————————————————

版权声明：本文为CSDN博主「t\_guest」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/qq\_26226375/article/details/127906179

————————————————

1.简介

蓝牙分为经典蓝牙（BT-Bluetooth）和低功耗蓝牙（BLE-Bluetooth Low Energy），本次主要学习BLE。

BLE分了很多个版本，现在用的比较多的就是4.2和5.X。那4.2到5.0之间有哪些升级呢？

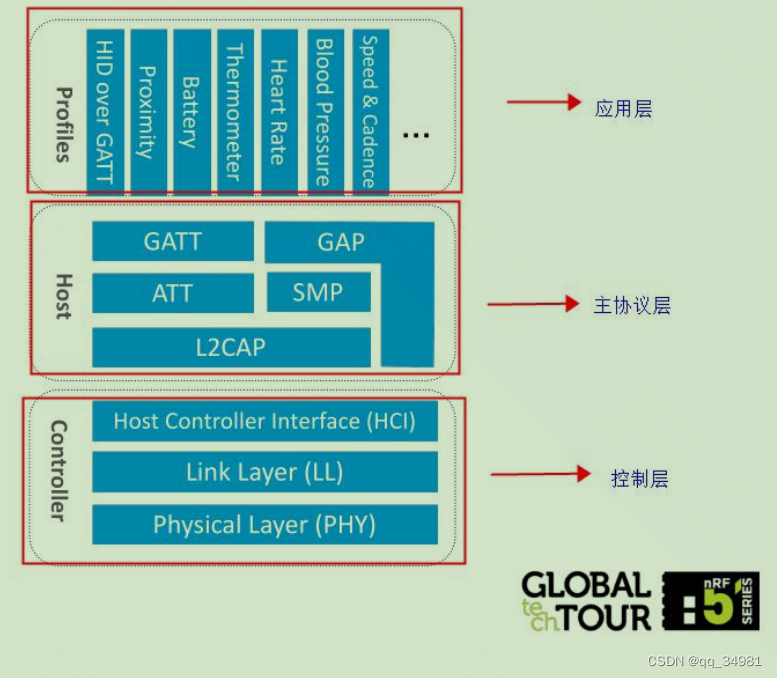
首先，4.2版本传输速度只有1Mbps，广播包最大长度为31字节。而5.0版本开始，传输速度就已经增加到2Mbps了，并且广播包的最大长度也增加为254字节。除此之外，5.X版本还增加了Mesh的功能，且通讯距离也增加至300米以上。

2.架构

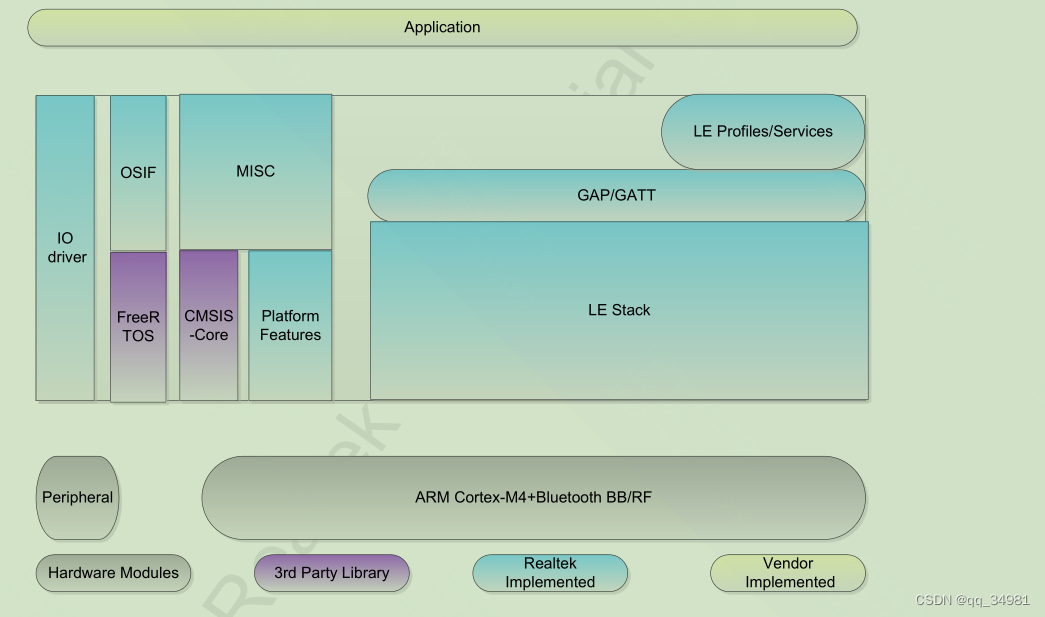
蓝牙里把**蓝牙协议的实现代码称为协议栈(protocol stack)**，BLE协议栈就是实现低功耗蓝牙协议的代码，理解和掌握BLE协议是实现BLE协议栈的前提。

每个厂商提供的SDK中包含的BLE协议栈都会有些许不同。这里主要以Nordic和RTL两个厂商来进行分析。

先看Nordic的架构



再看RTL的架构



乍一看两者的区别很大。但是再定睛一看，区别还是很大。开玩笑，其实两者区别不大，只要搞懂底层基础的概念，这些东西就都是纸老虎。

先看Nordic的架构，从上图可以看到。**一个完整的BLE程序被分为了三层，分别是应用层、主协议层和控制层。**这些层中分别还包含了各自的内容。

其中的术语，这里大概解释一下。

PHY层（Physical layer-物理层）。PHY层用来指定BLE所用的无线频段，调制解调方式和方法等。PHY层做的好不好，直接决定了整了BLE芯片的功耗、灵敏度等射频指标。

LL层（Link Layer-链路层）。**LL层是整个协议栈的核心**。也是BLE协议栈的难点和重点。像Nordic的BLE协议栈能同时支持20个Link连接，就是LL层的功劳。

**LL层要做的事情非常多**，比如具体选择哪个射频通道进行通信。怎么识别空中数据包。具体在哪个时间点把数据包发送出去。怎么保证数据的完整性。ACK如何接收。如何进行重发。以及如何对链路进行管理和控制等等。

LL层只负责把数据发送出去或者接收回来，对数据进行怎样的解析则由GAP或ATT来负责。

HCI（Host controller interface）。HCI是可选择的。HCI主要用于2颗芯片实现BLE协议栈的场合，用来规范两者之间的通讯协议和通信命令等。

（主机与蓝牙芯片之间的接口。）

**GAP层（Generic access profile-通用访问配置文件）**。GAP是对LL层payload（有效数据包）如何进行解析的两种方式的一种，而且也是最简单的一种。

GAP简单的对LL payload进行一些规范和定义，因此GAP能实现的功能极其有限。GAP目前主要用来进行广播，扫描和发起连接。

L2CAP层（logic link control and adaptation protocol-**逻辑链路控制和适配协议**）.L2CAP对LL进行了一次简单的封装，LL只关心传输的数据本身，

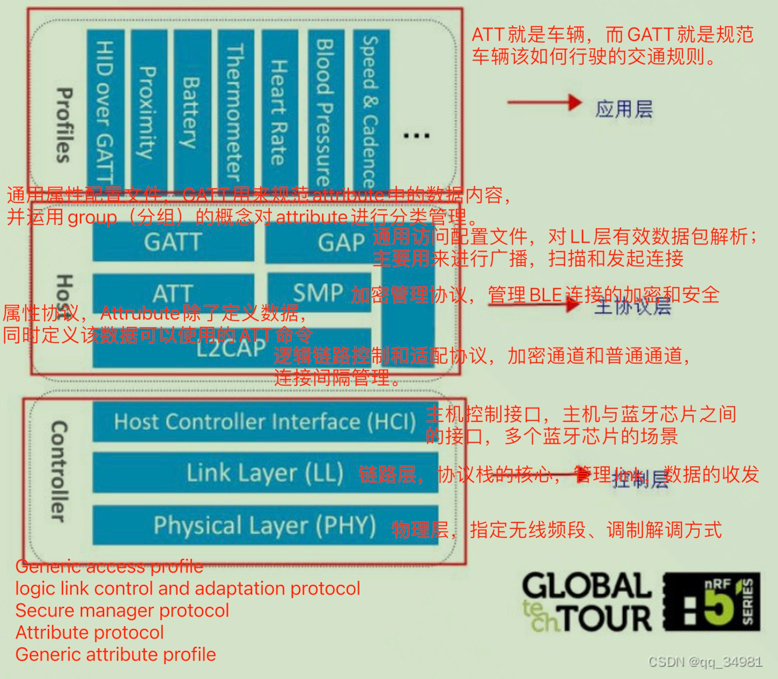
**L2CAP就要区分是加密通道还是普通通道**，同时还要对连接间隔进行管理。

SMP（Secure manager protocol-加密管理协议）。**SMP用来管理BLE连接的加密和安全。**如何保证连接的安全性，同时不影响用户的体验，这些都是SMP要考虑的工作。

ATT（Attribute protocol-属性协议）。简单来说，**ATT层用来定义用户命令及命令操作的数据。**比如读取某个数据或者写某个数据。BLE协议栈中，开发者接触最多的就是ATT。BLE引入了attribute概念，用来描述一条一条的数据。Attrubute除了定义数据，同时定义该数据可以使用的ATT命令，因此这一层被称为ATT层。

GATT（Generic attribute profile-通用属性配置文件）。GATT用来规范attribute中的数据内容，并运用group（分组）的概念对attribute进行分类管理。没有GATT，BLE协议栈也能跑。但互联互通就会出问题，也正是因为有了GATT和各种各样的应用profile，BLE摆脱了Zigbee等无线协议的兼容性困境，成为了出货量最大的2.4G无线通信产品。

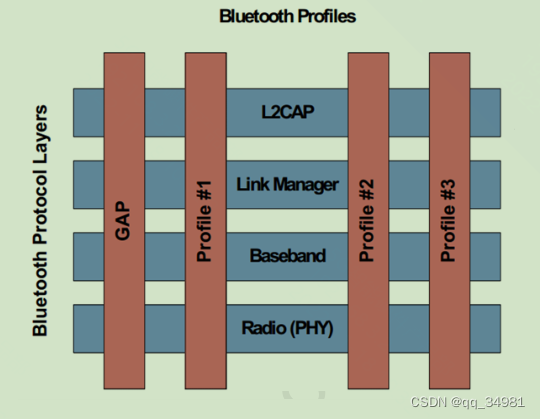
ATT和GATT的关系就像车辆和交通规则一样。ATT就是车辆，而GATT就是规范车辆该如何行驶的交通规则。只有GATT存在的时候，多个ATT见面才知道该如何形式，到底是拐弯让直行还是坐车道超车。



从上边的名字可以看到，主要分为protocol和profile。其中L2CAP、SMP、ATT属于protocol，而GAP和GATT属于profile。

profile和protocol有什么区别呢? 在蓝牙核心规范（Bluetooth Core Specification）中，profile的定义不同于protocol的定义。Protocol被定义为各层协议。而Profile从使用蓝牙核心规范中各层协议的角度，定义蓝牙应用互操作性的实现。Profile定义Protocol中的可用特性和功能，以及蓝牙设备互操作性的实现，使蓝牙协议栈适用于各种场景的应用开发。

在蓝牙核心规范中，Profile和Protocol的关联如下图:



Profile由红色矩形框表示，包括GAP、Profile#1、Profile#2和Profile#3。蓝牙核心规范中的Profile分为两种类型：GAP，途中红色矩形框所示的GAP；基于GATT的Profile，图中红色矩形框所示的Profile#123。

可能初学者看完上述之后还是云里雾里的。没有关系，我们在后续的文章中，会对这些进行单独讲解，也会再重复提到。只要跟着走，后续这些问题都会弄清楚的。加油！

# 蓝牙学习二（连接和通讯简述）

————————————————

版权声明：本文为CSDN博主「t\_guest」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/qq_26226375/article/details/127919515>

————————————————

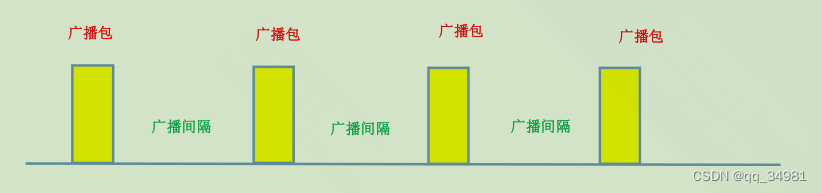
1.简介

蓝牙的通信是双向的，为了创建和维护一个BLE通信连接，在蓝牙中引入了“角色”这一概念，一个BLE设备不是主机（集中器）就是从机（外围设备）角色，这是根据是谁发起这个连接来确定的。主机（集中器）设备总是连接的发起者，而从机（外围设备）总是被连接者。整个访问与连接过程都是在GAP（Generic Access Profile-通用访问规范）进行实现的。

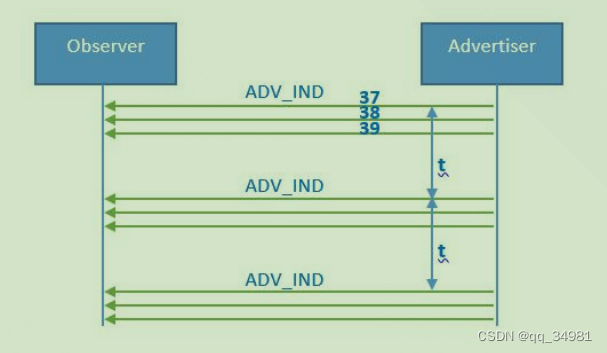
2.从机广播

从机要被主机连接，那么它就必须先被主机发现。这个时候，从机设备把自身信息通过广播的形式发射出去。

比如设备A需要先进行广播，即设备A（Advertiser）不断发送广播信号，t 为广播间隔。每发送一次广播包，我们称为一次广播事件（advertising event），因此 t 也称为广播事件间隔。广播事件时间是一段一段的，每次都会持续一段时间，蓝牙芯片也只有在广播事件期间才会打开射频模块发射广播，这个时间功耗比较高（几十毫安），其余时间蓝牙芯片都处于idle待机状态，因此平均功耗就非常低（几微安）。



当广播发出的时候，每个广播事件包含了三个广播包，分别在37/38/39三个广播信道上进行发送。广播的内容是相同的。下图observer为主机观察者，advertiser就是从机广播者。

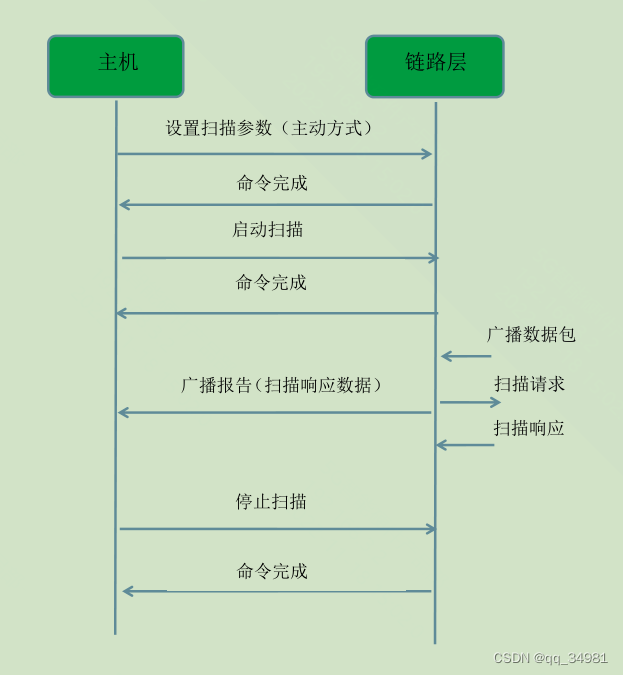


3.主机扫描

设备A不断发送广播给主机（Observer），如果主机不开启扫描窗口，是收不到设备A的广播的。主机不但要开启射频接收窗口，而且主机的射频接收窗口要跟从机的广播发送窗口匹配才行。由于这种配对成功时一个概率事件，所以手机扫到设备A也是一个概率事件。也就是说，从机（Advertiser）可能很快就被主机（Observer）发现，也有可能从机（Advertiser）要很久才能被主机（Observer）发现。

下图表示了主机主动扫描广播的过程：

主机设置自身的扫描参数，设置成功后，控制器根据设置参数开启扫描窗口。当控制器收到扫描数据包后，向主机发送给一个广播报告事件（adv\_report），该事件同样包括了链路层数据包的广播类型。因此，主机能够判断对端设备是否可以连接或扫描，并且区分出广播数据包和扫描响应数据包。

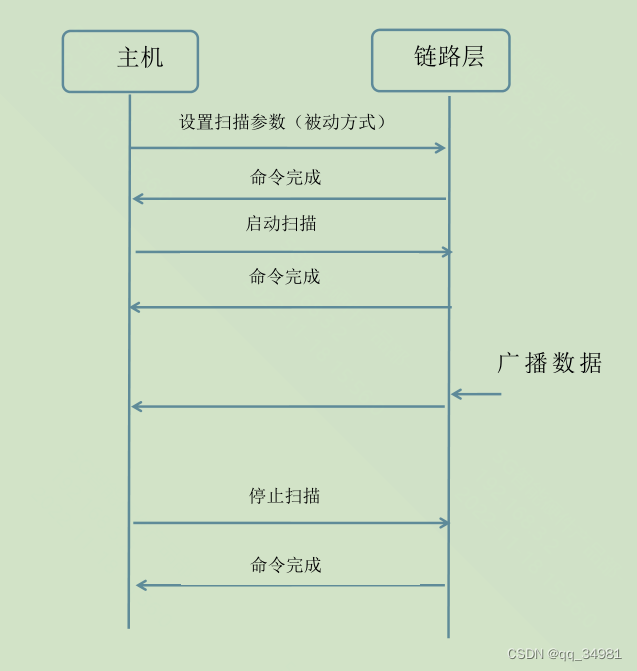


4.被动扫描

既然有主动扫描，当然也就有被动扫描。在被动扫描中，扫描者设备应该仅仅去监听广播包，而不向广播者设备发送任何数据。

一旦上面的扫描参数设置完毕，主机就可以启动扫描。如果控制器接收到的符合过滤策略和其他规划的广播数据包，则发送一个Advertising Report事件给主机。除了广播者的设备地址外，报告事件还包括广播数据包中的数据，以及接收广播数据包时的信号接收强度。可以利用该信号强度以及位于广播数据包中的发射功率，共同确定信号的路径损失，从而给出大致的范围，这个应用就是防丢器和蓝牙定位。被动扫描不需要向主机发送任何数据。

下图为被动扫描广播的过程：



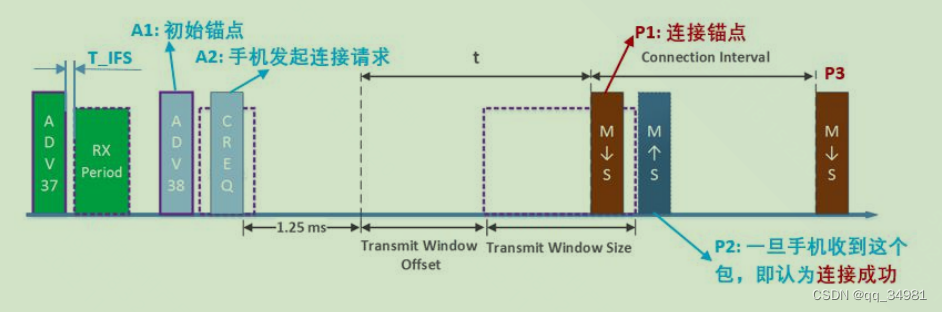
5.建立连接

手机（主机）在收到A1广播包ADV\_IND后，以此为初始点，T\_IFS时间后，给Advertiser（从机）发送一个connection request命令，即A2数据包，告诉advertiser（从机）将要进行连接，做好准备。Advertiser（从机）根据connect\_req命令信息做好接收准备。connect\_req其实就是告诉advertiser手机（主机）将在Transmit Window期间发送第一个同步包（P1），请advertiser（从机）在这段时间内打开接收窗口。设备B收到P1后，T\_IFS时间后将给手机（主机）回复数据包P2。一旦手机（主机）收到数据包P2，连接即认为建立成功。后续手机（主机）将以P1为锚点（原点），Connection Interval为周期，周期性地给设备B发送Packet。

主 A1 A2 P1 P2

↑ ↓ ↓ ↑

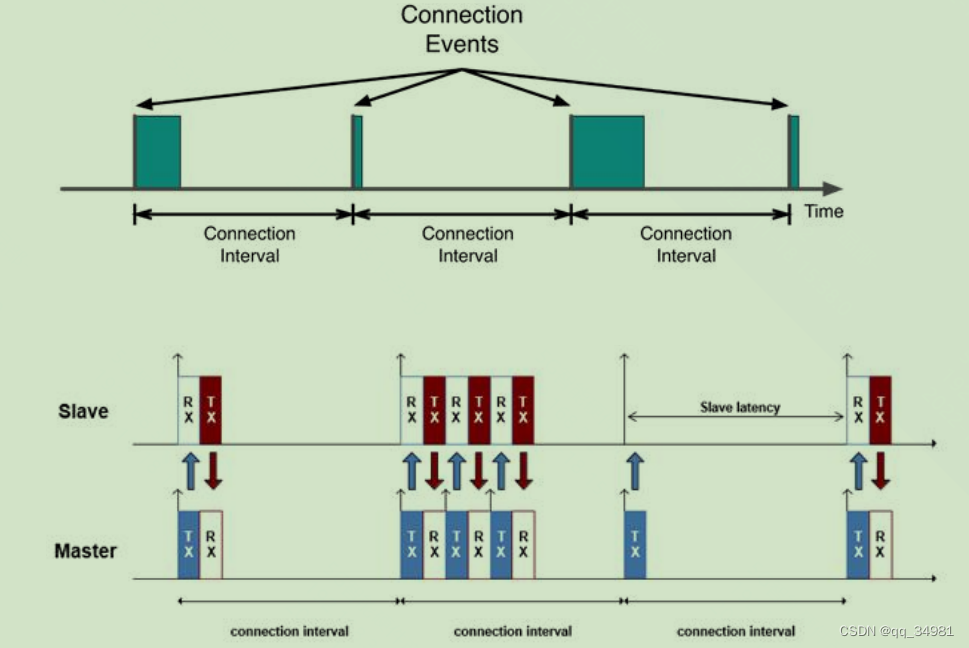
从



6.发送与接收数据

连接成功后，主机和从机在每一个connection interval开始的时候，都必须交互一次，即主机给从机发一个包，从机再给主机发一个包。整个交互过程被称为一个connection event。蓝牙芯片只有在connection event期间才把射频模块打开，此时功耗比较高，其余时间蓝牙芯片都处于idle状态，因此平均功耗非常低。

主机不可能时时刻刻都有数据给从机，所以主机大部分时间都是发的空包（empty packet）给从机。同样从机也不是时时刻刻都有数据给主机。因此从机回复给主机的包，大部分也是空包。**另外，一个connection event期间，主机可以发多个包给从机，以提高吞吐率（IOS一个连接间隔最多交互4次，安卓一个连接间隔最多交互6次）。**综上所述，连接成功后的通信时序图如下：



图中，主从数据发送的数据包TX和RX表示方向性的数据通道，也就是蓝牙的空中属性，空中操作时间都是采用蓝牙操作句柄来进行的。因为句柄能够唯一表示各个属性。空中特性的性质包括：

从机->主机方向：

通知：从机端上报数据给主机，不需要主机回复一个响应

指示：从机端上报数据给主机，需要主机端回复一个确认

通知和指示之间不同之处在于指示有应用层上的确认，而通知没有。

主机->从机方向：

写

没有回应的写

读

# 蓝牙学习三（GAP）

————————————————

版权声明：本文为CSDN博主「t\_guest」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/qq\_26226375/article/details/128017160

————————————————

1.简介

GAP（Generic Access Profile-通用访问配置文件）与应用层紧密相连，所以要想了解BLE，GAP是必须认识的东西。

在第一章中我们说过GAP层，GAP层目前主要用来进行广播、扫描和发起连接。GAP保证了不同的BLE设备可以互相发现对方并建立连接。有不清楚的，建议再回去看一下。

GAP定义了蓝牙设备如何发现和建立与其他设备的安全或不安全连接。它处理一些一般模式的业务，比如询问、命名和搜索，还处理一些安全问题，比如担保。同时还处理一些有关连接的业务，比如链路建立、信道和连接建立。GAP规定的是一些一般性的运行任务。因此，它具有强制性，并作为所有其他蓝牙应用规范的基础。GAP是所有其他配置文件的基础，它定义了在蓝牙设备之间建立基带链路的通用方法，除此之外，GAP还定义了下列内容：

必须在所有蓝牙设备中实施的功能

发现和链接设备的通用步骤

基本用户界面术语

**GAP是应用层能够直接访问BLE协议栈的最底层，它包括管理广播和连接事件的有关参数。**GAP模块代表了所有蓝牙设备的共用基础功能，如传输，协议或者应用规范所使用的模式和访问过程。**GAP的服务包括设备发现，连接方式，安全，认证，关联模型和服务发现等。**

总之一句话，GAP是保证BLE设备能互相发现和连接的规范。GAP就好比一种语言，大家都说同一种语言才能相互听懂，相互知道说的是什么。否则就只能大眼瞪小眼。

BLE协议栈的GAP层负责处理设备的接入方式和过程，包括设备发现，链路建立，链路终止，启动安全功能，设备配置。

GAP还定义了以下四种角色类型：

**广播者（Broadcaster）：**用于只通过广播发送数据的应用。广播发送者，是不可连接的设备。广播者角色只是广播信息而不接受信息（ibeacon类似应用）。

**观察者（Observer）**：用于接收广播数据的应用。扫描广播，不能够启动连接。观察者只监听空气中的事件（抓包器）。

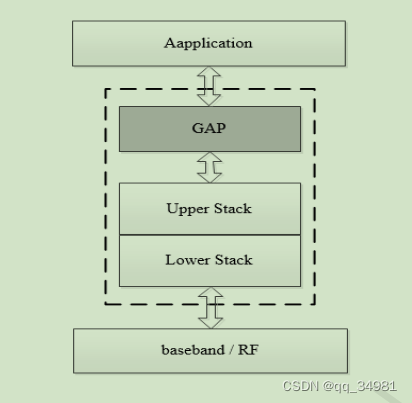
**外围设备（Peripheral）**：用于通过广播发送数据并且可以建立链路的应用。广播发送者，是可连接的设备，在单一链路层连接时作为从机。

**中央设备（Central）**：用于接收广播数据并且建立一条或多条链路的应用。扫描广播启动连接，在单一或多链路层连接时作为主机，支持三个同时连接。是吗？？

2. GAP结构

2.1GAP的位置

GAP层作为蓝牙协议层的组成模块，如下图所示，虚线框内的部分为蓝牙协议层。Application在蓝牙协议层之上，baseband/RF位于蓝牙协议层之上。GAP层给application提供访问Upper stack的接口。



2.2GAP的功能

GAP提供如下功能

广播（Advertising）：设置和获取广播参数，启动和停止

扫描（Scan）：设置和获取扫描参数，启动和停止

连接（Connection）：设置连接参数，创建连接，终止连接，更新连接参数

配对：设置配对参数，启动配对。使用passkey entry方式时输入/显示passkey，删除绑定设备密钥

密钥管理：根据设备地址和地址类型查找密钥，保存/加载绑定设备信息的密钥，解析random address

其他：

设置GAP公共参数，例如外貌和名字。

获取支持的最大BLE链路数目

修改白名单（white list）

生成/设置本地设备random address

配置本地设备identiy address

广播

中央设备能够与外围设备建立连接，外围设备必须处于广播状态，它每经过一个时间间隔发送一次广播数据包，这个时间间隔被称为广播间隔。他的范围是20ms到10.24s。广播间隔影响建立连接的时间。

中央设备在发送一个连接请求来发起连接之前，必须接收到一个广播数据包。外围设备发送一个广播数据包之后一小段时间内只监听连接请求。

一个广播数据包最多能携带31字节的数据，它通常包含用户可读的名字、关于设备发送数据包的有关信息、用于表示此设备是否可被发现的标志等类似的标志。

当中央设备接收到广播数据包后，它可以发送请求更多数据包的请求，称为扫描回应。如果它被设置成主动扫描。外围设备将会发送一个扫描响应作为对集中器请求的回应。扫描回应最多可以携带31字节的数据。

扫描

扫描是中央设备监听广播数据包和发送扫描请求的过程。它有2个定时参数需要特别注意：**扫描窗口和扫描间隔。**

对于每一个扫描间隔，中央设备扫描的时间等于一个扫描窗口，这就意味着如果扫描窗口等于扫描间隔，那么集中器将处于连接扫描之中。扫描窗口和扫描间隔之比称为扫描占空比。

连接

中央设备和外围设备第一次交换数据定义为连接状态。在一个连接状态中，中央设备将会在一个特定定义的间隔从外围设备请求数据，这个间隔称为连接间隔，它由中央设备决定并应用于连接，但是外围设备可以发送连接参数更新请求给中央设备。根据蓝牙核心规范，连接间隔必须在7.5ms到4S之间。

如果外围设备在一个时间帧内没有回应集中器的数据包，被称为连接监管超时，连接则被认为丢失。

可以通过在每一个连接间隔中传输多个数据包以获得更高的数据吞吐量（IOS一次连接间隔最多可以交互4次，安卓为6次）。在蓝牙5.0里，每个传输数据包最多可以写到247个字节的应用数据。但是如果电流消耗是重点，同时外围设备也没有数据要发送，则可以忽略一定数量的连接间隔，这个忽略连接间隔的数目称为从机延迟（从机潜伏）。

**在一个连接中，除了广播信道，设备间在频带的所有信道中进行通信。**

【广播、扫描、连接】

2.3 GAP层设备状态

GAP层设备状态由advertising、scan、connection组成。每个状态都有相应的子状态，

advertising

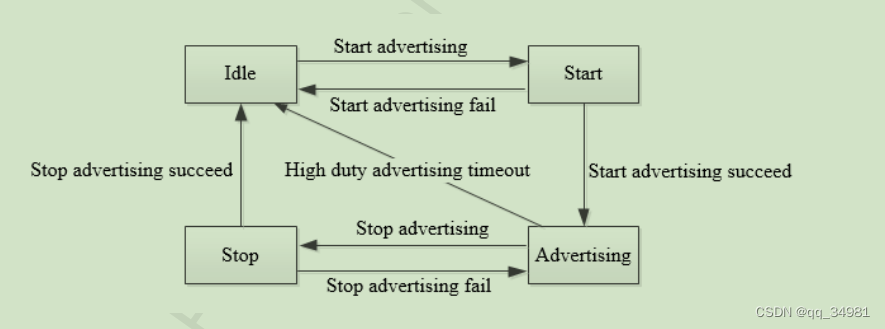
advertising有四个子状态，分别为idle、start、advertising和stop。

idle：默认状态，不发送广播

start：在idle启动广播后，启动广播的流程尚未完成。start为临时状态，若成功启动广播，则进入广播状态。若启动失败，则回到idle状态

advertising：成功启动广播后，设备发送广播。若广播类型为快速广播，一旦超时，则进入idle状态。

stop：在停止广播之后，停止广播的流程尚未完成。stop为临时状态，若成功停止广播，则进入idle状态。若停止失败，则回到advertising状态。



scan

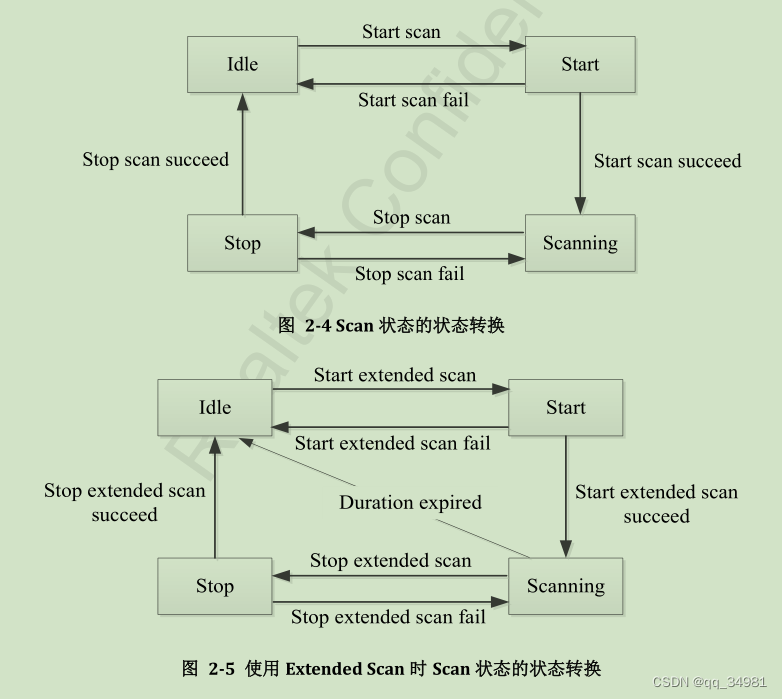
scan状态有四个子状态，idle、start、scanning和stop。

idle：不进行scan

start：在idle状态启动scan之后，启动scan的流程尚未完成。start状态为临时状态，若成功启动scan，则进入scanning状态。若启动scan失败，则返回idle状态。

scanning：成功启动scan。在此状态下，设备进行scan，接收advertisement。使用延时扫描时，若持续时间参数非零，则scan时间超时后，进入idle状态。

stop：在scanning停止之后，停止scan的流程尚未完成。stop状态为临时状态，若成功停止scan，则进入idle状态。若停止scan失败，则返回scanning状态。



3.GAP配置

3.1GAP安全模式

GAP在启动连接时会负责BLE连接的安全功能。只有对通过身份验证的连接而言的某些数据是可读或可写的。一旦形成一个连接，两个设备可以通过一个过程进行连接，称为配对。**进行配对时，密钥建立加密或认证的链接。**例如当两个设备要建立配对关系时，外围设备需要中央设备提供密钥以完成配对。这个密钥可以是固定值或随机值。中央设备发送正确的密钥后，两台设备交换安全密钥并验证链接。

在许多情况下，两个设备会经常建立连接和断开连接。BLE具有一个安全功能，允许两个设备在配对的时候给对方一个长久的安全密钥。此功能被称为绑定。绑定后的两个设备每次重连时能够迅速重新确立加密和认证，并且不需要经过充分的配对过程，只需要提供他们存储的长期密钥信息即可。

加密方式大概分为一下几种

不允许连接

无安全要求（可直接连接）

加密连接

MITM（man in the middle-中间人攻击）保护

LESC（LE secure connections-）加密

签名

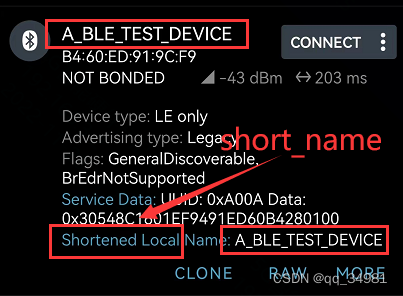
【配对、绑定】

3.2蓝牙设备名称与外观

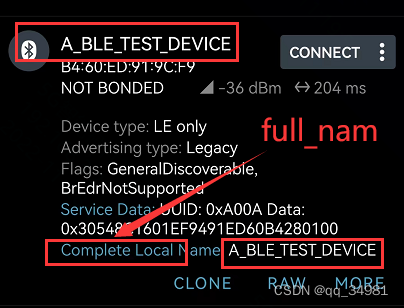
设备名称

当我们需要查找蓝牙设备的时候，大部分情况都是通过设备名称来查找的。协议栈也是支持用户自行修改设备名称的。在设置的时候可以设置short\_name和full\_name。

short\_name：可以成为可选名称，字面意思上可以理解为名称的简写。当一个名称比较长的时候，由于广播的信息容量是有限的。要留空间放置其他信息，没有那么多空间显示全名，因此APP显示的时候为简称。



full\_name：如果名称不是很长，广播数据可以容纳。那么可以通过GAP设置全名显示。



设备外观

当我们在搜寻蓝牙的时候，不但可以找到蓝牙设备的名称，还可以查看到蓝牙设备的外观。外观是什么，就决定了我们找的设备是什么。例如未知、键盘、鼠标、温度计、手表等等。

4.GAP连接

在一个典型的蓝牙系统中，外围设备发送具体的广播数据让所有中央设备知道它是一个可连接的设备。广播内容包含设备地址，还包含了一些额外的数据，比如设备名称。中央设备接收到广播之后发送一个搜索请求给外围设备，外围设备答复一个搜索应答，这就是设备发现的过程。

【（发现过程：）从机：->广播，<-搜索请求，->搜索应答：主机】

设备发现之后，中央设备就知道外围设备是一个可连接的设备。中央设备可以发送一个建立链接的请求给外围设备，开始进行主机和从机设备的互联。

在一个连接中，包含了如下参数：

连接间隔：

在一个BLE连接中，跳频机制需要被使用，这样两个设备之间可以在一个特定的通道上进行数据收发。在一个特定的时间之后，会跳到一个新的通道上，LL层负责通道切换。这个发现设备并收到数据的事件，**就被称为连接事件**。

尽管没有应用程序数据需要收发，两个设备之间仍然需要交换链路层数据来保持连接。连接间隔是两个连接事件之间的时间，使用一个单元是1.25ms的步进。连接间隔从最小值6（7.5ms）到最大值3200（4.0S）。

不同的应用需要不同的连接间隔，一个长时间的连接间隔将会节约更多的能量，因为设备可以在两个连接事件之间睡眠更长的时间。但是他会导致数据发送不及时。如果有数据要发送，那么只能等下一个连接事件到来时才能被发送。

值得注意的是，**连接间隔参数是中央设备进行设置，外围设备只是负责执行。【怎么设置的？】**

从机潜伏周期：

这个参数描述了从机跳过连接事件的次数。这使外围设备具有一定的灵活性，如果它不具有任何数据传送，它可以选择跳过连接事件，并保持睡眠，从而提供了一些积蓄力量。这一决定取决于外围设备。也可以成为从机延迟，从设备能够忽略主设备的链接事件的最大值。比如：从设备延迟为6的话，那么从设备每隔6个锚点可以监听到主设备发送过来的数据包。如果从设备延迟为0，从设备在每个锚点都可以监听主设备的数据包。

监督超时：

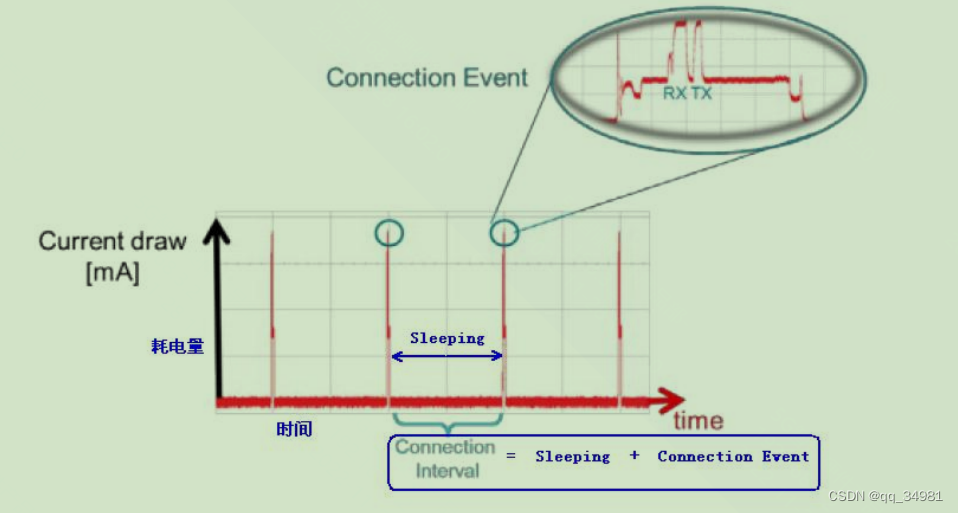
这是两个成功的连接事件之间间隔的最大值。如果超过这个时间还未出现成功的连接事件，那么设备将会考虑失去连接，返回一个未连接状态。这个参数值使用10ms的步进。监督超时时间从最小10（100ms）到最大3200（32.0s）。监督超时时间必须大于有效连接事件。

有效连接事件时间 = 连接间隔 \* （1 + 从机延迟值）。

连接间隔、潜伏周期的不同组合，有好有坏，而如何组合，需要根据实际的应用场景而定。例如短连接间隔，功耗高，数据吞吐量高，发送等待时间短。长链接间隔，功耗低，数据吞吐量低，发送等待时间长。低潜伏值，从机在没有数据发送的情况下功耗高，但从机可以快速的收到主机的数据。高潜伏值，从机在没有数据发送的情况下可以低功耗。但从机无法及时收到主机的数据，但主机能及时收到从机的数据。

在某些情况下，中央设备请求与外围设备建立链接包含的连接参数对外围设备而言是不利的。此时，外围设备可以在连接过程中改变连接参数。这个取决于外围设备的应用程序。**外围设备可以通过设置连接参数更新请求，请求中央设备改变连接参数。**这个请求包含四个参数：最小链接间隔、最大链接间隔、从机潜伏周期、连接超时时间。这些值代表了外围设备针对连接的期望参数，连接间隔是以范围的形式提供的。当中央设备接收到这个请求后，中央设备有权利决定是接受还是拒绝这些参数。

4.1 连接间隔



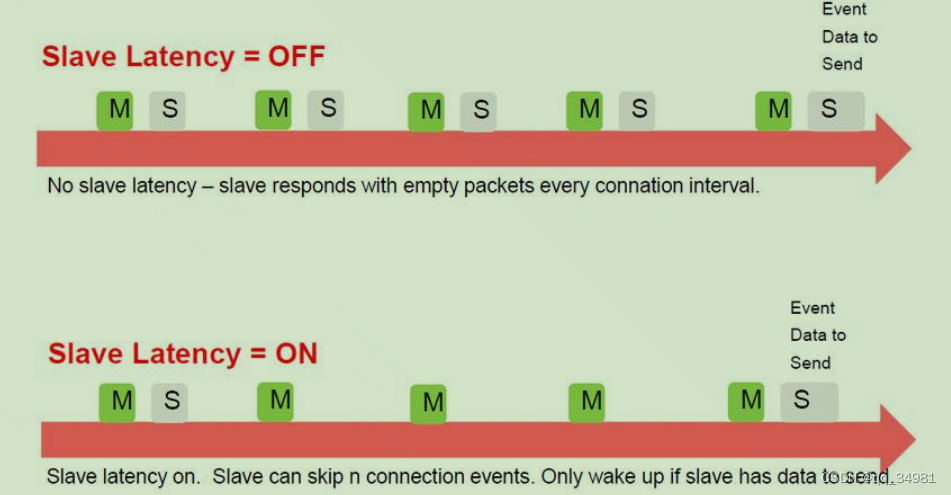
每个尖刺就是一次连接事件（Connection events），两个连接事件的中间为sleeping休眠时间。从图中也可以看出，在每次从机的连接事件中，首先是射频接收，然后才是射频发送。其实也是为了保证主机和从机的射频时序是相匹配的。设备在建立链接之后的大多数事件都是处于sleeping。这种情况下耗电量比较低，只有在连接事件（connection events）中，耗电量才相对较高，但持续时间很短。这也是为什么BLE非常省电的原因之一。

在两个connection interval连接间隔之间，BLE会使用跳频机制。两个设备使用特定的信道发送和接收数据，然后过一段时间后，再使用新的信道（BLE协议栈的LL链路层处理信道的切换）。两个设备在切换信道后发送和接收数据被称为一个连接事件。尽管没有应用数据被发送或接收，两个设备仍旧会交换链路层数据（空包 Empty PDU）来维持连接。在每个连接事件中，都是由主机发送，从机回复。

当大数据传递时，通讯数据包是连续传递的。主机会选择最小连接间隔来进行通讯。无数据传递时，主机会选择最大链接间隔来定期询问从机状态，以保持连接不中断。当从机向主机发送一个连接参数更新请求后，主机会按照请求，修改连接间隔。但这个时间需要在主机设置的最大和最小连接间隔之内。

4.2从机潜伏周期

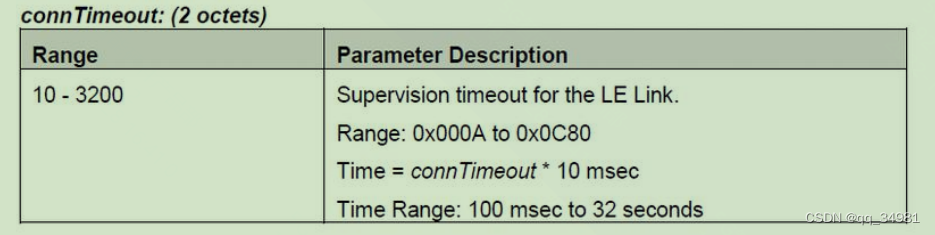
从机设备没有数据要发时，跳过一定数目的连接事件Connection Event。范围0-499。设置为0则每次连接间隔都连接。数字越小，通讯速度越快，但功耗越高。注：如果在距离远或干扰大的情况下，当无数据通信时，这个值设置大，可以减少掉线的概率。比如防丢器：如果这个参数设置为0，表示规定时间内必须响应从机，不然就以为是蓝牙断开了。如果设置为3，假设信号不好时，中间丢了3个包，只接受到了1个就表示连接正常。因为他会跳过其中3个，保证了不掉线的概率。



允许从机在没有数据发送的情况下，跳过一定数目的连接事件。在这些连接事件中，不必回复主机的包。这样就更省电。

4.3连接超时

超时时间，就是两个设备在连接的这段时间没有发生通讯而导致连接自动断开的值。Range(10ms-32s)。用在信号不太好的情况下，给对方一点时间，超过这个时间通讯就建立失败。



连接间隔、从机潜伏周期以及超时时间这三者必须满足以下公式：

超时时间 > (1 + 从机潜伏周期) \* 连接间隔。否则连接就会不正常断开。

这三个连接参数不同情况下对通信速率和功耗的影响：

连接间隔缩短，主机和从机通信更加频繁。提高了数据吞吐速度，缩短了数据发送的时间，增加了功耗。

连接间隔增长，通信频率降低，数据吞吐速度降低，增加了数据发送的时间，降低了功耗。

潜伏间隔减少，每次连接事件中都回复主机包，功耗增加，数据发送速度提高。潜伏间隔增加，功耗降低，数据发送速度降低。

蓝牙协议栈定义了主机决定连接参数的值（连接间隔、潜伏周期、超时时间），从机可以请求更新这些参数，但是由主机决定是否接受，接受的值是多少。所以会出现手机接受参数后和从机请求的参数有偏差的情况，甚至是拒绝（IOS）。安卓和IOS都是在手机和设备建立链接时就会默认设置这些参数，**APP开发时无法修改这些参数，这些默认参数是由手机厂商决定的。**

# 蓝牙学习四（广播）

————————————————

版权声明：本文为CSDN博主「t\_guest」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/qq_26226375/article/details/128091893>

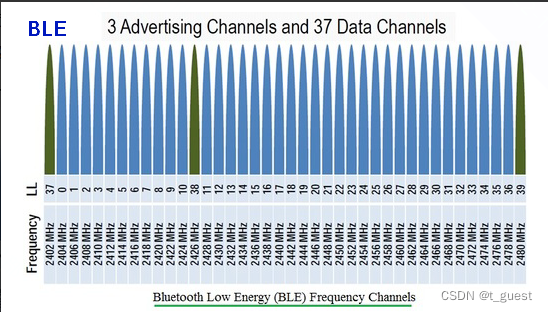
————————————————

1.简介

什么叫做广播，顾名思义就像广场上的大喇叭一样，不停的向外传输着信号。不同的是，大喇叭传输的是音频信号，而蓝牙传输的是射频信号。

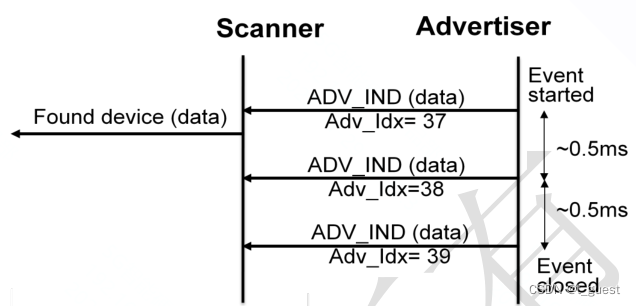
BLE使用的是无线电波传递信息，就是将数据编码，调制到射频信号中发射。BLE使用的射频频率是2.4GHz。跟WIFI、Zigbee等协议使用的是同一频段。

那如何做到使用同一频段而有不相互干扰呢？首先要知道的是2.4G指的不是某一个频率，而指的是一个频段（2400MHz-2483.5MHz）。在这个频段内每隔2M为一个信道，共40个信道。2.4G频段是一个用于短距离，无须执照使用的开放频段。意思就是可以免费使用。为了不占用更多的资源从而造成相互干扰，每个设备在使用时，同一时刻，只会在一个信道进行工作，不会占用其他信道。一个BLE设备，在任一时刻，只能选择40个信道之中的一个进行发射或监听。



BLE将信道划分为广播信道和数据信道。广播信道只有3个，37、38、39。剩下的37个信道全都是数据信道0-36。

在广播事件中，每一个广播事件都会在3个广播信道中进行数据传输，而且每一个事件都是从最小的信道编号开始传输。也就是说当广播事件来了，数据包从广播信道37、38、39中依次进行传输。



2. 广播间隔

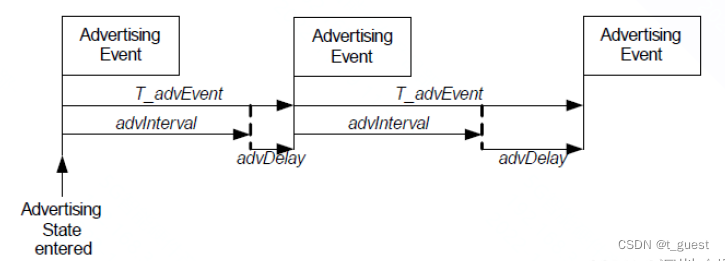
设备每次广播时，会在3个广播信道发送相同的报文。这些报文报文的动作被称为一个广播事件。除了定向广播外，其他广播事件均可以选择20ms-10.28s不等的间隔。通常，一个广播中的设备会每一秒广播一次。两个相邻的广播事件之间的时间被称为“广播间隔”。

设备周期性的发送广播会有一个问题：由于设备间的时钟会不同程度的漂移，**两个设备可能在很长一段时间同时广播而造成干扰。**为了防止这一情况的发生，除定向广播外的其他广播类型，发送时间均会有些许波动。实现方式为，在上一次广播事件后加入“0-10ms"的随机延迟。这意味着，即使两个设备广播间隔相同，并在相同信道及时间点上发送造成了冲突，但他们发送下一个广播事件时也很大概率不会冲突。

所以，两个相邻的广播事件之间的时间间隔T\_advEvent为：

T\_AdvEvent = advInterval + advDelay

其中，advInterval必须是0.625ms的整数倍，范围是20ms-10.24s之间。对于可扫描非定向广播和不可连接非定向广播者两种广播类型，该值最好不小于100ms，即160个0.625ms。advDelay是LL层分配的一个随机数，范围为0-10ms。



在实际的设置中，通过设置Advertising\_Interval\_Min（最小广播间隔）和Advertising\_Interval\_Max（最大广播间隔）这两个参数来调整广播间隔。都是以0.625ms为单位。如果要固定广播间隔为某一个值，需要将这两个参数设置为同一个值即可。

3.广播类型

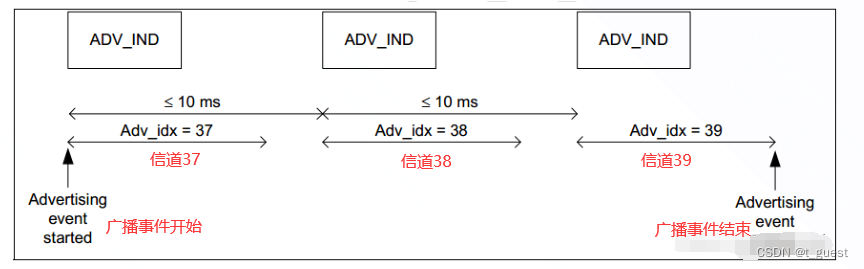
3.1 非定向可连接广播事件（ADV\_IND）

ADV\_IND就是链路层通过广播信道发送广播的事件。发送的PDU（Protocol Data Unit-协议数据单元）是ADV\_IND\_PDU-通用广播报文。这个报文发送之后可以接收由扫描者发送的SCAN\_REQ\_PDU-扫描请求，或者由发起者发送的CONNECT\_REQ\_PDU-连接请求。而接收后链路层需要在同一个信道上进行扫描或回复发起者的应答。当接收的数据报文不符合广播滤波协议，要么就用下一个广播信道进行广播，要么就停止广播事件。如果接收到的SCAN\_REQ\_PDU通过了滤波协议，那么广播者需要在150±2us内在同一信道回复SCAN\_RSP\_PDU-扫描应答报文。如果接收到CONNECT\_REQ\_PDU，则进入连接状态，这个时候并不需要进行应答。

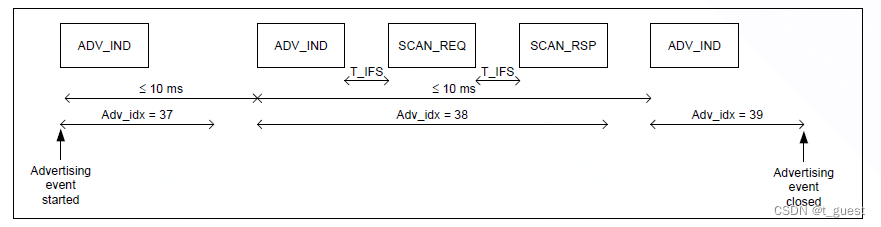
需要注意的是，一个广播事件中，相邻两个ADV\_IND\_PDU之间的时间需要不大于10ms。

接下来分类一下此类广播事件中广播包的发送情况。

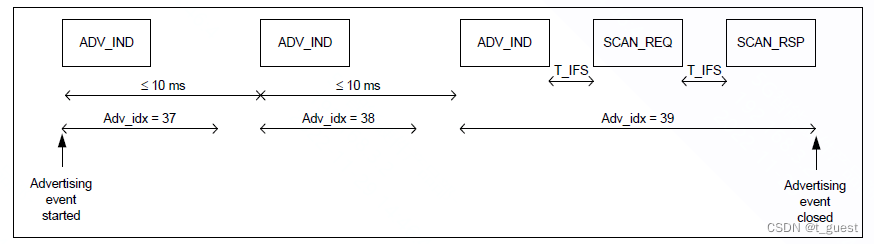
（1）仅仅有广播PDUS



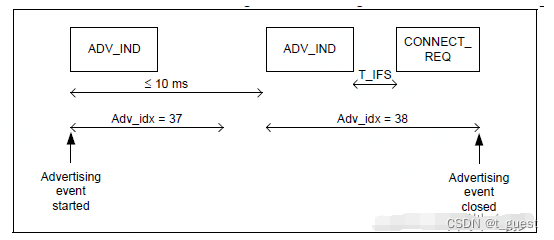
（2）在广播事件中有SCAN\_REQ\_PDUS和SCAN\_RSP\_PDUS。

 注：当有扫描请求包再广播事件中的中间信道上收到时，T\_IFS（帧间隔）为150us。

（3）在广播事件的结尾有SCAN\_REQ和SCAN\_RSP。



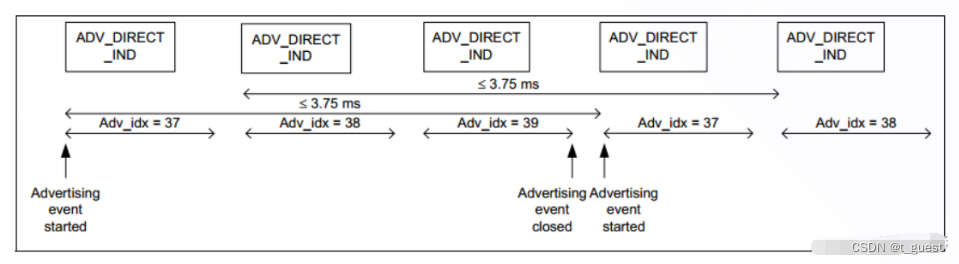
（4）在广播事件的中间接收到CONNECT\_REQ-连接请求包。没有应答



3.2 定向可连接广播事件（ADV\_DIRECT\_IND）

这个广播是为了快速建立链接。这种报文包含两个地址：广播者地址和发起者地址。发起者收到发给自己的定向广播报文后，可以立刻发送连接请求事件作为回应，并立刻进入连接状态。

定向广播事件有特殊的时序要求。完整的广播事件必须每3.75ms之内重复一次，即3.75ms内在37、38、39三个广播信道上全部发送一次报文。这样的方法使得扫描设备只需要扫描3.75ms即可收到定向广播设备的消息。



如果按照定向广播的要去持续发送报文的话，广播信道将充斥着大量的定向广播报文。如此其他设备将无法进行广播。所以，蓝牙协议规定，定向广播不能持续1.28S以上。如果主机没有主动要求停止，或者连接没有建立，控制器会自动停止广播。一旦超过1.28S，主机只能使用通用广播让其他设备连接。

3.3 非定向不可连接事件（ADV\_NONCONN\_IND）

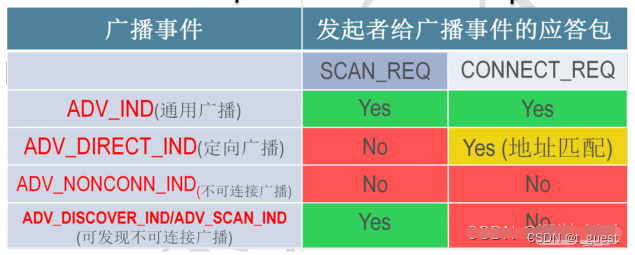
该广播的时间要求与通用广播事件相同。此外，该事件只向外发射广播报文，但是不可以被连接，也不接收任何信息。是唯一一个只有发射而没有接受的广播类型。ibeacon发出的就是这种类型的广播。

3.4 非定向可发现不可连接事件（ADV\_DISCOVER\_IND/ADV\_SCAN\_IND）

该广播的时间要求与通用广播事件相同，应答也是SCAN\_REQ和SCAN\_RSP。这个广播和通用广播的区别是，它不能建立连接。

注：所谓的定向和非定向针对的是广播对象，如果是针对特定的对象进行广播（在广播包PDU中包含目标对象的MAC），则为定向广播。反之为非定向广播。可连接和不可连接指的是是否接受连接请求。如果是不可连接的广播类型，它将不应答连接请求报文。可扫描广播类型会回应扫描请求。

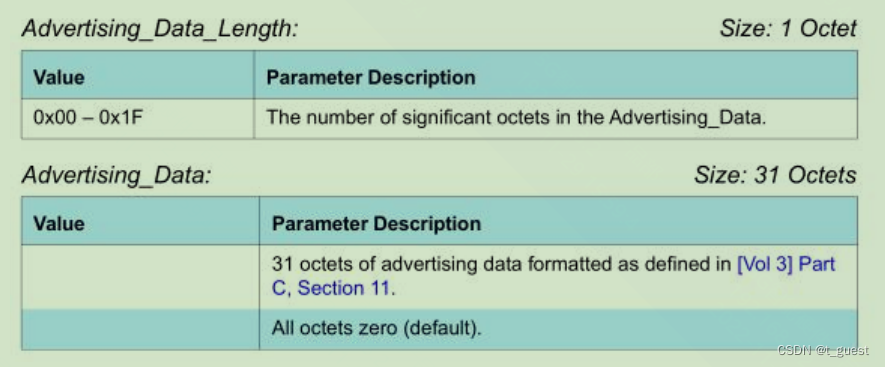
不同的广播类型对应的扫描请求和连接请求如下图：



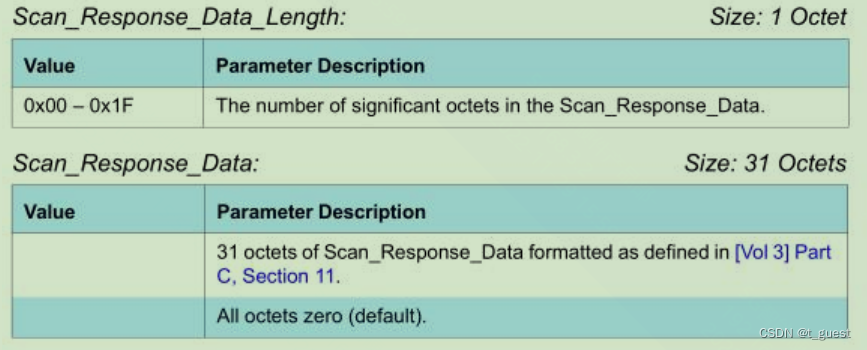
4. 广播响应包

广播包有两种：广播包（Advertising Data）和响应包（Scan Response）。其中广播包是每个设备必须广播的，而响应包是可选的。

广播包在蓝牙5.0协议栈核心中介绍如下：



应答包介绍如下：



每个包都是31字节，数据包中分为有效数据（significant）和无效数据（non-significant）

有效数据部分：包含若干个广播数据单元，称为AD Structure。AD Structure的组成是：第一个字节是长度值Len，表示接下来的Len个字节是数据部分。数据部分的第一个自己表示数据的类型AD Type，剩下的len-1个字节是真正的数据AD Data。

无效数据部分：因为广播包的长度必须是31个字节，如果有效数据部分不到31字节，则剩余部分用0补全。

广播响应包是为了给广播一个额外的31字节数据，用于主机在主动扫描情况下，反馈数据使用。