

目 录

第 1 章 概述	(1)
1.1 蓝牙——驱动新经济的引擎	(1)
1.2 蓝牙技术及产品发展现状分析	(2)
1.3 蓝牙技术介绍	(4)
1.4 蓝牙协议体系结构	(6)
1.5 蓝牙应用模型及协议栈	(8)
1.6 蓝牙技术的应用	(11)
第 2 章 基带层协议	(12)
2.1 概述	(12)
2.2 物理信道	(13)
2.3 物理链路	(14)
2.4 分组	(15)
2.4.1 通用格式	(15)
2.4.2 识别码	(16)
2.4.3 分组头	(17)
2.4.4 分组类型	(19)
2.4.5 有效载荷格式	(23)
2.5 纠错	(25)
2.5.1 前向纠错码	(25)
2.5.2 ARQ(自动重复请求)方案	(26)
2.5.3 错误校验	(29)
2.6 逻辑信道	(30)
2.7 数据加噪	(31)
2.8 收/发规则	(32)
2.8.1 TX 规则	(32)
2.8.2 RX 规则	(34)
2.8.3 流控制	(35)
2.8.4 比特流处理	(35)
2.9 发/收定时	(36)
2.9.1 主/从定时同步	(36)
2.9.2 连接状态	(37)
2.9.3 退出保持模式	(38)
2.9.4 唤醒休眠状态	(38)
2.9.5 呼叫状态	(38)

2.9.6	FHS 分组	(39)
2.9.7	多从单元操作	(40)
2.10	信道控制	(41)
2.10.1	主 - 从定义	(41)
2.10.2	蓝牙时钟	(41)
2.10.3	状态综述	(42)
2.10.4	识别过程	(42)
2.10.5	查询过程	(47)
2.10.6	连接状态	(50)
2.10.7	散射网	(56)
2.10.8	节能管理	(58)
2.10.9	链路监测	(58)
2.11	跳频选择	(59)
2.11.1	通用选择方案	(59)
2.11.2	选择内核	(60)
2.11.3	控制字	(61)
2.12	蓝牙音频	(63)
2.12.1	对数 PCM 编译码器(CODEC)	(64)
2.12.2	连续变化斜率增量调制编译码器(CVSD CODEC)	(64)
2.12.3	错误处理	(65)
2.12.4	一般音频要求	(65)
2.13	蓝牙编址	(65)
2.13.1	蓝牙设备地址(BD-ADDR)	(65)
2.13.2	识别码	(66)
2.13.3	活动成员地址(AM-ADDR)	(68)
2.13.4	休眠成员地址(PM-ADDR)	(68)
2.13.5	访问请求地址(AR-ADDR)	(68)
2.14	蓝牙安全性	(69)
2.14.1	随机数发生器	(69)
2.14.2	字管理	(70)
2.14.3	加密	(71)
2.14.4	鉴权	(75)
第 3 章	链路管理器协议	(77)
3.1	概述	(77)
3.2	链路管理器协议格式(LMP)	(77)
3.3	过程规则与 PDU	(78)
3.3.1	通用应答消息	(79)
3.3.2	鉴权	(79)
3.3.3	匹配	(80)
3.3.4	改变链接字	(82)

3.3.5	改变当前链接字	(83)
3.3.6	加密	(84)
3.3.7	请求时钟补偿	(86)
3.3.8	时钟补偿信息	(87)
3.3.9	计时精度信息请求	(87)
3.3.10	LMP 版本	(88)
3.3.11	蓝牙支持特性	(89)
3.3.12	主、从角色切换	(90)
3.3.13	请求命名	(90)
3.3.14	断开连接	(91)
3.3.15	保持模式	(91)
3.3.16	呼吸模式	(92)
3.3.17	休眠模式	(94)
3.3.18	功率控制	(97)
3.3.19	在 DM 和 DH 之间基于质量的信道变化	(98)
3.3.20	服务质量(QoS)	(99)
3.3.21	SCO 链路	(100)
3.3.22	多时隙分组控制	(101)
3.3.23	呼叫方案	(102)
3.3.24	链路监控	(103)
3.4	建立连接	(103)
3.5	测试模式	(104)
3.5.1	激活和解除测试模式	(104)
3.5.2	测试模式的控制	(105)
3.6	出错处理	(105)
第 4 章	逻辑链路控制和适配协议	(107)
4.1	概述	(107)
4.2	主要操作	(109)
4.2.1	信道标识符	(109)
4.2.2	设备间操作	(109)
4.2.3	层间操作	(110)
4.2.4	分段和重组	(111)
4.3	状态机	(112)
4.3.1	事件	(114)
4.3.2	动作	(116)
4.3.3	信道操作状态	(117)
4.3.4	事件到行为的映射	(118)
4.4	数据分组格式	(121)
4.4.1	面向连接信道	(121)
4.4.2	无连接数据信道	(121)

4.5	信令	(122)
4.5.1	指令拒绝(代码 0x01)	(123)
4.5.2	连接请求(代码 0x02)	(124)
4.5.3	连接应答(代码 0x03)	(124)
4.5.4	配置请求(代码 0x04)	(125)
4.5.5	设置应答(代码 0x05)	(126)
4.5.6	断开请求(代码 0x06)	(127)
4.5.7	连接断开应答(代码 0x07)	(127)
4.5.8	回应请求(代码 0x08)	(128)
4.5.9	回应应答(代码 0x09)	(128)
4.5.10	信息请求(代码 0x0A)	(128)
4.5.11	信息应答(代码 0x0B)	(129)
4.6	配置参数选项	(129)
4.6.1	最大传输单位(MTU)	(130)
4.6.2	刷新超时选择	(130)
4.6.3	服务质量(QoS)选项	(131)
4.6.4	配置处理	(132)
4.7	小结	(134)
第 5 章	服务搜索协议(SDP)	(135)
5.1	引言	(135)
5.2	SDP 概述	(136)
5.2.1	客户服务器交互	(136)
5.2.2	服务记录	(137)
5.2.3	服务属性	(137)
5.2.4	服务类	(138)
5.2.5	服务搜索	(139)
5.2.6	服务浏览	(140)
5.3	数据表示	(141)
5.4	协议说明	(143)
5.4.1	协议数据单元格式	(143)
5.4.2	局部应答和后续状态	(144)
5.4.3	出错处理	(144)
5.4.4	服务搜索处理	(145)
5.4.5	服务属性事务	(147)
5.4.6	服务搜索属性事务	(149)
5.5	服务属性定义	(152)
5.5.1	通用属性定义	(152)
5.5.2	“服务搜索服务器”服务类属性定义	(157)
5.5.3	“浏览组描述符”服务类属性定义	(158)
第 6 章	基于 TS 07.10 的 RFCOMM 协议	(160)

6.1	引言	(160)
6.2	RFCOMM 服务	(161)
6.2.1	RS-232 控制信令	(161)
6.2.2	空 Modem 仿真	(161)
6.2.3	多串口仿真	(162)
6.3	服务接口描述	(163)
6.4	RFCOMM 支持的 TS07.10 子集	(164)
6.5	根据蓝牙对 TS07.10 的修正	(165)
6.5.1	介质调整	(165)
6.5.2	TS07.10 多路复用器的启用和关闭过程	(166)
6.5.3	系统参数	(166)
6.5.4	利用 RFCOMM 服务器通道进行 DLCI 定位	(167)
6.5.5	多路复用控制指令	(167)
6.6	流控制	(168)
6.7	与其他实体的互操作	(169)
6.7.1	端口仿真和端口代理实体	(169)
6.7.2	服务注册和搜索	(170)
6.7.3	低层约束	(170)
第 7 章	IrDA 互操作性	(172)
7.1	概述	(172)
7.2	OBEX 对象和协议	(173)
7.2.1	对象模型	(173)
7.2.2	会话协议	(174)
7.3	OBEX over RFCOMM	(176)
7.4	OBEX over TCP/IP	(177)
7.5	利用 OBEX 的蓝牙应用概述	(178)
第 8 章	电话控制二进制协议	(179)
8.1	概述	(179)
8.1.1	设备间操作	(179)
8.1.2	层间操作	(180)
8.2	呼叫控制(CC)	(182)
8.2.1	呼叫状态	(182)
8.2.2	呼叫建立	(182)
8.2.3	呼叫清除	(185)
8.3	组管理(GM)	(187)
8.3.1	无线用户组(WUG)	(188)
8.3.2	获取访问权限	(189)
8.3.3	配置分布	(189)
8.3.4	成员间快速访问	(190)
8.4	无连接 TCS(CI)	(191)

8.5	补充服务(SS)	(192)
8.5.1	呼叫线路识别	(192)
8.5.2	DTMF 启动和终止	(192)
8.5.3	注册重呼	(193)
8.6	报文格式	(193)
8.6.1	呼叫控制报文格式	(194)
8.6.2	组管理报文格式	(197)
8.6.3	CL INFO	(199)
8.7	报文编码	(199)
8.7.1	协议标识和报文类别	(199)
8.7.2	其他信息元	(201)
8.8	报文出错处理	(211)
8.9	协议参数	(212)
第 9 章	WAP 信道的蓝牙互操作性要求	(213)
9.1	蓝牙环境中的 WAP 应用	(213)
9.1.1	增值服务	(213)
9.1.2	应用实例	(213)
9.2	WAP 服务概述	(214)
9.2.1	WAP 实体	(214)
9.2.2	WAP 协议	(214)
9.2.3	WAP 和 INTERNET 间的协议转换	(215)
9.3	WAP 在蓝牙匹克网中的应用	(216)
9.3.1	WAP 服务器通信	(216)
9.3.2	蓝牙环境下的 WAP 应用	(217)
9.3.3	对 WAP 的网络支持	(218)
9.4	互操作性要求	(219)
9.5	服务搜索	(219)
9.5.1	SDP 服务记录	(219)
9.5.2	服务搜索过程	(221)
第 10 章	主控制器接口功能规范	(222)
10.1	概述	(222)
10.1.1	蓝牙软件栈底层	(222)
10.1.2	蓝牙硬件块描述	(222)
10.1.3	物理总线体系结构	(223)
10.1.4	主控制器层概述	(224)
10.2	HCI 流控制	(225)
10.3	HCI 指令	(226)
10.3.1	引言	(226)
10.3.2	数据和参数格式	(226)
10.3.3	HCI 信息交换	(226)

10.3.4	链路控制指令	(230)
10.3.5	链接策略命令	(246)
10.3.6	主控制器与基带命令	(253)
10.3.7	信息参数	(286)
10.3.8	状态参数	(289)
10.3.9	测试指令	(292)
10.4	事件	(294)
10.4.1	事件	(294)
10.4.2	事件说明	(296)
10.5	错误码表	(310)
10.5.1	错误码表	(310)
10.5.2	错误码用法描述	(311)
第 11 章	HCI 传输层	(316)
11.1	HCI USB 传输层	(316)
11.1.1	HCI 终端要求	(316)
11.1.2	类别码	(321)
11.1.3	设备固件升级	(321)
11.1.4	限制	(321)
11.2	HCI RS232 传输层	(322)
11.2.1	概述	(322)
11.2.2	协商协议	(323)
11.2.3	分组传输协议	(325)
11.2.4	使用含有 COBS 的分界符同步	(325)
11.2.5	使用 RTS/CTS 同步	(327)
11.3	HCI UART 传输层	(330)
11.3.1	协议	(330)
11.3.2	RS232 设置	(330)
11.3.3	纠错	(331)
第 12 章	蓝牙测试模式	(332)
12.1	概述	(332)
12.2	测试环境	(333)
12.2.1	发送端测试	(333)
12.2.2	回送测试	(336)
12.3	LMP 消息概览	(339)
第 13 章	蓝牙兼容性要求	(342)
13.1	概述	(342)
13.2	蓝牙认证计划	(343)
13.3	蓝牙产品许可要求	(345)
13.3.1	蓝牙无线链路要求	(345)
13.3.2	蓝牙协议要求	(345)

13.3.3 蓝牙框架要求	(345)
13.3.4 蓝牙信息请求	(346)
13.3.5 蓝牙外设产品要求	(346)
13.5.6 蓝牙部件要求	(346)
13.3.7 蓝牙许可条款	(347)
13.4 有关蓝牙产品功能信息的建议	(347)
13.5 质量管理、配置管理和版本控制	(347)
第 14 章 测试控制接口	(348)
14.1 概述	(348)
14.2 描述	(348)
14.2.1 基带和链路管理验证	(348)
14.2.2 HCI 验证	(349)
14.2.3 逻辑链路控制和适配验证	(350)
14.3 测试配置	(351)
14.4 TCI - L2CAP 描述	(353)
14.4.1 事件	(353)
14.4.2 命令	(354)
14.4.3 数据传输	(357)
缩略语	(359)

第1章 概述

1.1 蓝牙——驱动新经济的引擎

1999年11月,IT时代“软件王国”的缔造者比尔·盖茨专程来到拉斯维加斯一间只有11名员工的小公司。为什么?只因这家公司已研制成功一种含蓝牙技术的胸卡。

1999年12月,微软宣布全面支持“蓝牙”技术。到2000年初,蓝牙SIG(Special Interest Group,特殊利益集团)已有3com、爱立信、IBM、英特尔、朗讯、微软、摩托罗拉、诺基亚、东芝等9大集团公司和2000多家成员企业。蓝牙技术到底如何,竟让盖茨如此动心,让IT行业的巨头们和众多的厂商走到一起?

蓝牙的英文名称是Bluetooth,是1998年5月由爱立信、IBM、英特尔、诺基亚、东芝等5家公司联合制定的近距离无线通信技术标准,其目的是实现最高数据传输速率1Mb/s(有效传输速率为721kb/s)、最大传输距离为10m的无线通信。Bluetooth原为欧洲中世纪的丹麦国王Harald II的名字,他为统一四分五裂的瑞典、芬兰、丹麦立下了不朽的功劳。瑞典爱立信公司为这种即将成为全球通用的无线技术命此名,也许大有一统天下的含义。

1999年7月,蓝牙SIG公布正式规范1.0版本,而遵从这一规范的移动电话和笔记本电脑也将在2000年底上市,声称要把蓝牙技术产品化的企业也与日俱增。

2000年6月在新加坡召开的“Communic Asia”展览会上,爱立信公司推出了全球第一部使用蓝牙技术的GPRS手机——R520,R520手机融合了GPRS、高速数据(HSCSD)、蓝牙技术和WAP,除了高速率外,R520还可以借助其内置蓝牙芯片提供全面无线连接解决方案,从而避免了在电话和其他移动设备(如PC和免提设备)之间铺设线缆。据了解,除了R520外,另一款采用蓝牙技术的手机——T36适用于GSM标准的3种制式(900/1800/1900MHz),支持高速数据通信HSCSD技术,可与同时上市的无线耳机连接,当手机收到信号时,只要轻按一下耳机上的按钮即可用无线方式接通耳机以及进行蓝牙对话。据了解,这两款手机发送信息时,可按耳机上的按钮用语音识别方式通话,手机均内置了蓝牙收发信号模块和WAP浏览器。

作为蓝牙技术的另一倡导者,IBM也宣布了一系列对蓝牙计划的支持,主要体现在拳头产品ThinkPad笔记本电脑上。IBM已在第二季度出台了一系列新的无线增强技术,以与IBM成功的ThinkPad笔记本电脑的线路设计相配套,同时在2000年5月推出应用蓝牙技术的全新ThinkPad笔记本电脑。这款笔记本电脑带有Portofino端口,能方便地接到无线调制解调器、照相机和其他设备上。新款ThinkPad支持IEEE802.11规程,所以只要给笔记本插上这种规格的网卡就可以进行无线网络通信。IBM有关负责人表示,在推出新产品的同时也会考虑ThinkPad老用户的需要。第三季度,IBM已为使用老式ThinkPad的用户推出一种蓝牙PC卡和一种连接到较新式机型的蓝牙收发器,同时发布的还有用于Palm便携设备上的调制解调器。通过蓝牙技术,笔记本电脑将不再需要无线调制解调器或是单独的无线ISP账号,而是将来自笔记本电脑的数据通过无线电设备发送到蜂窝电话,然后再由蜂窝电话进

行传输。

业界人士认为爱立信、IBM 将使蓝牙技术产品化具有战略意义，他们在很多方面具有优势，广泛的合作伙伴关系将为他们提供很大的发展空间。除爱立信、IBM 外，东芝、摩托罗拉、英特尔等公司也将纷纷推出基于蓝牙技术的笔记本电脑芯片等。一直难有突破性进展的掌上电脑，如果运用蓝牙技术，毫无疑问，则可以形成一个很大的市场，也许能使“掌上时代”的到来成为现实。

据 Frost & Sullivan 公司发布的市场调查和预测报告显示，1999 年蓝牙技术产品的全球销售量几乎为零，2000 年猛增至 3670 万美元，2001 年将达 1.26 亿美元，2006 年有望高达 6.99 亿美元；2002 年，全球使用蓝牙技术的调制解调器等外围设备将达 1.5 亿台，使用蓝牙技术的笔记本电脑将达 2500 万部；2003 年，全球 90% 以上的笔记本电脑将使用蓝牙技术；2005 年，全球将推出 6.7 亿台使用蓝牙技术的信息家电。

1.2 蓝牙技术及产品发展现状分析

蓝牙 (Bluetooth) 技术自提出以来，在短短 2 年时间里已风靡全球。目前，全球已有 2 000 多家企业推出了蓝牙芯片、蓝牙平台、应用程序、测试设备等产品。在摩纳哥蓝牙 2000 年大会上，有公司预测，今后 2 年内使用蓝牙技术的设备将达到 5 000 万台，到 2005 年蓝牙设备产量将超过 14 亿台。

客观地说，蓝牙采用的技术中有些并非是当前该领域最先进的技术。蓝牙的目标是全球通用、价格低廉、结构紧凑，因此它并不强调技术的先进性。比如纠错编码方式，蓝牙采用的是 1/3 率的重复码、2/3 率的汉明码，而没有采用相同编码速率的卷积码、TURBO 码或其他更先进的编码方式。作为用户，总希望使用的产品所采用的技术越先进越好，而对规范实现者和产品生产商而言，总希望产品的制造成本越低越好。

Micrologic 的 Quinn 说：“蓝牙芯片必须具有小巧、廉价、结构紧凑和功能强大的特点才能放进蜂窝电话中”。蓝牙芯片的价格和大小下不来，既有经济原因，也有技术原因。从技术角度看，蓝牙芯片集成了无线、基带和链路管理层的功能，事实上，链路管理层既可通过硬件实现，也可通过软件实现，如果由软件实现链路管理层的功能，那么芯片将被简化，其价格和大小将变得合理。

索尼在日本无线展览会的现场进行了蓝牙和 IEEE802.11b 与微波炉之间的相互干扰实验。结果表明，在无干扰的情况下，数据传送速度为 500 kb/s~600 kb/s。一旦使用微波炉，由于干扰的出现，数据传送速度降至 300 kb/s，此时再使用对应 IEEE802.11b 规格的无线 LAN，由于干扰的加大，数据传送速度下降至 100 kb/s~299 kb/s。未来的蓝牙产品应用环境包括扩频设备、跳频设备、无线 LAN、微波炉等。又据 SIG 英特尔公司在京的一次会议上谈到，国际 SIG 在各种环境中做过实验，低功率蓝牙产品对其他同类产品的干扰微乎其微，相反，其他产品对蓝牙产品的干扰可通过软件或硬件方法解决。

安全问题包括信息安全和生态安全。信息安全问题更多是在软件协议栈中加以强调。OEM 希望知道说明特殊应用（如商务、国防等）中的安全要求，以便由软件工程师去解决它。生态安全问题是当蓝牙设备靠近人体时是否带来危害，对此人们非常关心。蜂窝电话业多年来一直在这个问题上进行讨论，但是到目前为止一直不能证明是否真正有危害，也不能给出造成危害的根据。不可避免地，蓝牙产品的主要问题是由于蓝牙产品使用和微波炉一

样的频率范围,这是否会带来不良后果,目前也尚无定论。一些组织认为蓝牙产品输出功率很小(只有 1 mW),是微波炉使用功率的百万分之一,是移动电话的一小部分。而在这些输出中,仅仅有一小部分被物体吸收,根本检测不到温度的增加。

互操作性是蓝牙产品的重要特性。从理论上说,只要通过了产品的一致性和互连性测试,互操作性问题就可以得到解决。目前蓝牙协议中的许多互连测试规范尚未推出,即使推出了,其测试的完备性也有一个过程。国际 SIG 对蓝牙互操作性非常重视,因为它涉及到蓝牙产品的进一步应用,各大公司接连不断开会进行沟通、测试、试验,目的就是使其产品具有相互可操作性。

以下是近期蓝牙产品研发的几个“第一”:

(1) Motorola 引入新的蓝牙产品,使这一新技术第一次用于移动电话

2000 年 9 月 25 日, Motorola 又推出新的蓝牙产品,一种应用蓝牙技术的移动电话。这是将 Motorola Timeport270 电话与蓝牙智能模块和 PC 卡组合构成的新产品。

(2) Toshiba 第一个蓝牙 PC 卡投放市场

2000 年 9 月 25 日,计算机系统集团(CSG)的东芝(Toshiba)美国信息系统公司率先在美国推出全集成的蓝牙 PC 卡。东芝是 1998 年成立的蓝牙 SIG 九个发起人之一,致力于开发无线电产品和服务,是第一个把蓝牙 PC 卡投放市场的公司。利用东芝的蓝牙 PC 卡及其 SPANworks 软件,用户可以在 100 英尺范围内共享信息,即时交换信息和传送文件,以及交换商务卡。

(3) Motorola 的 PC 卡和 USB 适配器成为第一批得到认证的产品

2000 年 9 月 28 日, Motorola 宣布它是接受 Allied Business Intelligence(ABI)蓝牙产品全面认证的第一个公司。其认证的产品有 PC 卡和 USB 适配器,它们通常用于笔记本电脑和台式计算机。由于 Motorola 在市场时间上的领先,因此能使其下家(如 Toshiba 和 IBM 公司)首先将它们的产品送到用户手中。蓝牙产品的产值会很快增长,预计会从 2001 年的 5600 万美元增至 2005 年的 14 亿美元,其中半导体产业的商机约为 5.3 亿美元。

(4) 世界上第一个蓝牙无线网络

英国的 Red-M 公司在 2000 年 10 月 20 日宣布推出第一个网络产品。该公司是一个无线因特网服务开发商,它的新的接入服务器称为 Red-M 3 000AS(接入系统),它利用蓝牙技术实现短距离无线通信。服务器提供对因特网和本地互联网的移动接入,有关带蓝牙功能的设备有 PC 机、电话、PDA 和 WAP 电话等。服务器可与 WAN 和 LAN 接口匹配,也可以作为 Web 的高速缓存器、安全防火墙和虚拟个人网络,还可以作为主机发送电子邮件,作为网络服务器向蓝牙设备发送 mail 和 Web 内容。这样的蓝牙应用远远代替并超出了电缆的作用,开拓了一种新的移动通信应用,这就是无线局域网(PAN)系统。

(5) 第一个有望冲击 5 美元价格极限的消息

从 1998 年启动蓝牙行动至今,其市场迟迟不能起来,关键是在价格。谁又能跨越这个门槛呢? Cambridge Silicon Radio(CSR)相信他们能超越 5 美元这个价格极限。CSR 的 Bluecore02 芯片提供给 OEM 的是基于 CMOS 的无线电、基带,以及与全集成的蓝牙软件栈在一起的微控制器,每个芯片为 5 美元。蓝牙芯片用于移动电话、笔记本电脑、台式计算机和打印机,估计 2001 年底蓝牙芯片销售额将达到 5 600 万美元。Fujitsu 媒体设备公司最近利用 CSR 的蓝牙 CMOS 产品开发成一种智能卡,它可用于 PC 机、便携机、PDA 和数字照相机。

(6) 第一次证明蓝牙是世界上最小、价钱最低的服务器

Madge 网络公司属下的 Red-M 于 2000 年 10 月 17 日证实, 能够实现智能机间连接的低价格小服务器可由蓝牙技术来应对。服务器可以是内置的, 也可以是操作台, 使工作区内的蓝牙设备能及时进入因特网和个人互联网。Red-M 服务器是利用蓝牙技术将电子设备连接起来的一个行动, 目标是实现蓝牙网络解决方案。

(7) 第一个直接变换的蓝牙单芯片

全球通信技术(GCT)公司是前卫的新一代芯片开发商, 致力于无线电通信和因特网。该公司在 2000 年 10 月 18 日宣布正式进入蓬勃发展的无线电芯片市场, 并推出了第一种专门为蓝牙应用设计的无线电芯片 GDM1100, 它是直接变频单片蓝牙产品, 这一具有自主知识产权和专利技术的射频(RF)设计给 OEM 厂商建立了一个出类拔萃的蓝牙 RF CMOS 平台。尽管 GDM1100 很小, 它却把无线电前端和 MODEM 集成到了一个 CMOS 芯片上, 其无线电作用距离为 10 米, 可实现点对点和点对多点的无线通信, 邻道选择性为 -6 dB/MHz, 60dB 的接收机动态范围, 是多种多样的手持设备的理想配套设备。

(8) 世界上第一个得到认证的蓝牙单芯片

Silicon Wave 公司在 2000 年 10 月 19 日宣布, 它的 Odyssey SiW1502 无线电 MODEM 集成电路(IC)得到蓝牙认证, 成为世界上第一个得到认证的用于蓝牙无线电通信的单片无线电。作为无线电产品领先的开发商, Silicon Wave 得到认证的蓝牙产品还有 Odyssey 无线开发系统(WDS)和无线电 MODEM 评价系统(WMES)。

1.3 蓝牙技术介绍

蓝牙是一种低功耗的无线技术, 目的是取代现有的 PC、打印机、传真机和移动电话等设备上的有线接口。主要优点是: 可以随时随地用无线接口来代替有线电缆连接; 具有很强的移植性, 可应用于多种通信场合, 如 WAP、GSM、DECT 等, 引入身份识别后可以灵活实现漫游; 功耗低, 对人体危害小; 蓝牙集成电路应用简单, 成本低廉, 实现容易, 易于推广。

蓝牙技术提供低成本、近距离的无线通信, 构成固定与移动设备通信环境中的个人网络, 使得近距离内各种信息设备能够实现无缝资源共享。

蓝牙技术作为一种无线数据与语音通信的开放性标准, 它以低成本的近距离无线连接为基础, 为固定与移动设备通信环境建立一个特别连接。如果把蓝牙技术引入到移动电话和便携型电脑中, 就可以去掉移动电话与便携型电脑之间连接电缆的不便, 而通过无线建立通信。打印机、PDA、桌上型电脑、传真机、键盘、游戏操纵杆及所有其他的数字设备都可以成为蓝牙技术系统的一部分。除此之外, 蓝牙无线技术还为已存在的数字网络和外设提供通用接口, 以组建一个远离固定网络的个人特别连接设备群。

蓝牙技术工作在全球通用的 2.4 GHz ISM (工业、科学、医学) 频段, 蓝牙的数据速率为 1 Mb/s。从理论上讲, 以 2.45 GHz ISM 频段运行的技术能够使相距 30 米以内的设备互相连接, 传输速率可达到 2 Mb/s, 但实际上很难达到。应用了蓝牙技术——PLUG&PLAY 的概念 (有点类似“即插即用”的概念), 任意蓝牙技术设备一旦搜寻到另一个蓝牙技术设备, 马上就可以建立联系, 而无需用户进行任何设置, 可以解释成“即连即用”。在无线电环境非常嘈杂的环境下, 其优势更加明显。

蓝牙技术的另一大优势是它应用了全球统一的频率设定，这就消除了“国界”的障碍，而在蜂窝式移动电话领域，这个障碍已经困扰用户多年。

另外，ISM 频段是对所有无线电系统都开放的频段，因此使用其中的某个频段都会遇到不可预测的干扰源，例如某些家电、无绳电话、汽车房开门器、微波炉等，都可能是干扰源。为此，蓝牙技术特别设计了快速确认和跳频方案以确保链路稳定。跳频技术是把频带分成若干个跳频信道，在一次连接中，无线电收发器按一定的码序列不断地从一个信道跳到另一个信道，只有收发双方按这个规律通信，而其他的干扰源不可能按同样的规律进行干扰。跳频的瞬时带宽很窄，但通过扩展频谱技术可使这个窄带成倍地扩展成宽频带，使可能干扰的影响变得很小。与其他工作在相同频段的系统相比，蓝牙跳频更快，数据包更短，这使蓝牙技术系统比其他系统更稳定。

蓝牙技术目前主要以满足美国 FCC 要求为目标。对于在其他国家的应用，需要做一些适应性调整。蓝牙 1.0 规范已公布的主要技术指标和系统参数如表 1.1 所示。

表 1.1 蓝牙技术指标和系统参数

工作频段	ISM 频段: 2.402 GHz~2.480 GHz
双工方式	全双工, TDD 时分双工
业务类型	支持电路交换和分组交换业务
数据速率	1 Mb/s
非同步信道速率	非对称连接 721 kb/s、57.6 kb/s, 对称连接: 432.6 kb/s
同步信道速率	64 kb/s
功率	美国 FCC 要求小于 0 dbm(1 mW), 其他国家可扩展为 100 mW
跳频频率数	79 个频点/MHz
跳频速率	1600 次/秒
工作模式	PARK/HOLD/SNIFF
数据连接方式	面向连接业务 SCO, 无连接业务 ACL
纠错方式	1/3FEC, 2/3FEC, ARQ
鉴权	采用反应逻辑算术
信道加密	采用 0 位、40 位、60 位加密字符
语音编码方式	连续可变斜率调制 CVSD
发射距离	一般可达 10 m, 增加功率情况下可达 100 m

蓝牙支持点对点和一点对多点的通信。蓝牙最基本的网络组成是匹克网。匹克网实际上是一种个人区域网，这是一种以个人区域（即办公室区域）为应用环境的网络架构。需要指出的是，匹克网并不能够代替局域网，它只是用来代替或简化个人区域中的电缆连接。

匹克网由主设备单元和从设备单元构成。主设备单元负责提供时钟同步信号和跳频序列，而从设备单元一般是受控同步的设备单元，并接受主设备单元的控制。在同一匹克网中，所有设备单元均采用同一跳频序列。一个匹克网中一般只有一个主设备单元，而从设备单元目前最多可以有 7 个。

蓝牙协议模型主要包括：

- 物理层，即蓝牙无线接口层；

- 核心协议，基带（Baseband）协议、LMP、L2CAP、SDP 等；
- 电缆替代协议，RFCOMM；
- 电话传送控制协议，TCS 二进制、AT 命令集等。

由于蓝牙技术独立于不同的操作系统和通信协议之外，可以移植到许多应用领域，因而应用场合很普遍。蓝牙力求与不同的操作系统和通信协议有良好的接口，从而保证一定的兼容性。蓝牙技术适用于任何数据、图像、声音等短距离通信场合。目前所能看到的应用有：替换蜂窝电话和远端网络之间的通信时所用的有线电缆；提供新的多功能耳机，并可在 PC、蜂窝电话、随身听中共用；笔记本、PDA、蜂窝电话之间的名片数据交换等。

1.4 蓝牙协议体系结构

蓝牙特殊利益集团（SIG）开发的蓝牙规范 Version1.0，允许开发人员开发基于具有互操作的无线模块和数据通信协议的交互式服务和应用。本节主要对该规范中的协议和它们各自的功能及其相互关系进行概括性描述。

蓝牙协议规范的目标是允许遵循规范的应用能够进行相互间操作。为了实现互操作，在远程设备上的对应应用程序必须以同一协议栈运行。下述协议列表就是一个支持业务卡片交换应用的协议栈（自顶向下）实例：vCard→OBEX→RFCOMM→L2CAP→基带。该协议栈包括一个内部对象表示规则、vCard、无线传输协议和其他部分。不同应用可运行于不同协议栈。但是，每一协议栈都使用同一公共蓝牙数据链路和物理层。图 1.1 就是互操作应用支持的蓝牙应用模型之上的完整蓝牙协议栈。

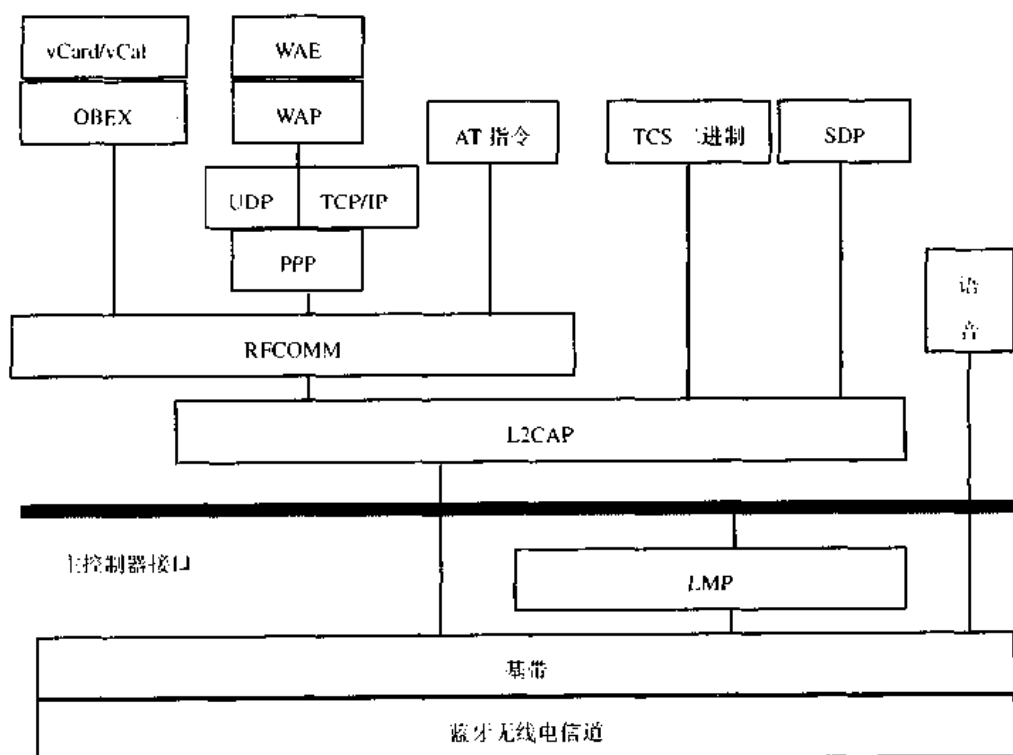


图 1.1 蓝牙协议栈

并不是所有应用程序都利用全部协议。相反，应用程序往往只利用协议栈中的某些部分。并且，协议栈中的某些附加垂直协议子集恰恰是用于支持主要应用的服务，比如说 TCS

(语音控制规范)或 SDP(服务搜索协议)等。实际上,上述示意图描述的是当需要无线传输数据有效载荷时,利用其他协议服务过程中的协议间关系。这些协议应具有与其他协议之间的关联。例如,一些协议(如 L2CAP、TCS 二进制)当需要控制链路管理器时,可以使用 LMP(链路管理器协议)。

如图 1.1 所示,整个蓝牙协议栈包括蓝牙指定协议(LMP 和 L2CAP)和非蓝牙指定协议(如对象交换协议 OBEX 和用户数据报协议 UDP)。设计协议和协议栈的主要原则是尽可能利用现有的各种高层协议,保证现有协议与蓝牙技术的融合以及各种应用之间的互通性,充分利用兼容蓝牙技术规范的软硬件系统。蓝牙技术规范的开放性保证了设备制造商可自由地选用其专利协议或常用的公共协议,在蓝牙技术规范基础上开发新的应用。

1. 蓝牙体系结构中的协议

蓝牙体系结构中的协议可分为四层。

- 核心协议:基带、LMP、L2CAP、SDP;
- 电缆替代协议:RFCOMM;
- 电话传送控制协议:TCS 二进制、AT 命令集;
- 可选协议:PPP、UDP/TCP/IP、OBEX、WAP、vCard、vCal、IrMC、WAE。

除上述协议层外,规范还定义了主机控制器接口(HCI),它为基带控制器、链路管理器、硬件状态和控制寄存器提供命令接口。在图 1.1 中,HCI 位于 L2CAP 的下层,但 HCI 也可位于 L2CAP 上层。

蓝牙核心协议由 SIG 制定的蓝牙指定协议组成,绝大部分蓝牙设备都需要核心协议(加上无线部分),而其他协议根据应用的需要而定。

2. 蓝牙核心协议

a. 基带协议

基带和链路控制层确保匹克网内各蓝牙设备单元之间由射频构成物理连接。蓝牙的射频系统是一个跳频系统,其任一分组在指定时隙、指定频率上发送,它使用查询和寻呼进程来使不同设备间的发送频率和时钟同步。基带数据分组提供两种物理连接方式:面向连接(SCO)和无连接(ACL),而且在同一射频上可实现多路数据传送。ACL 适用于数据分组,SCO 适用于语音及数据/语音的组合,所有语音与数据分组都附有不同级别的前向纠错(FEC)或循环冗余校验(CRC),而且可进行加密。此外,不同数据类型(包括连接管理信息和控制信息)都分配一个特殊通道。

可使用各种用户模式在蓝牙设备间传送语音,面向连接的语音分组只需经过基带传输,而不到达 L2CAP。语音模式在蓝牙系统内相对简单,只需开通语音连接,就可传送语音。

b. 链路管理协议(LMP)

链路管理协议(LMP)负责蓝牙各设备间连接的建立和设置。它通过连接的发起、交换、核实,进行身份验证和加密,通过协商确定基带数据分组大小;它还控制无线设备的节能模式和工作周期,以及匹克网内设备单元的连接状态。

c. 逻辑链路控制和适配协议(L2CAP)

逻辑链路控制和适配协议(L2CAP)是基带的上层协议,可以认为它与 LMP 并行工作。它们的区别在于当业务数据不经过 LMP 时,L2CAP 为上层提供服务。L2CAP 向上层提供

面向连接的和无连接的数据服务时,采用了多路复用技术、分段和重组技术及组概念。L2CAP 允许高层协议以 64K 字节收发数据分组。虽然基带协议提供了 SCO 和 ACL 两种连接类型,但 L2CAP 只支持 ACL。

d. 服务搜索协议 (SDP)

服务在蓝牙技术框架中起到至关重要的作用,它是所有用户模式的基础。使用 SDP,可以查询到设备信息和服务类型,从而在蓝牙设备间建立相应的连接。

3. 电缆替代协议

RFCOMM 是基于 ETSI 07.10 规范的串行仿真协议。“电缆替代”协议在蓝牙基带协议上仿真 RS232 控制和数据信号,为使用串行线传送机制的上层协议(如 OBEX)提供服务。

4. 电话控制协议

电话控制协议(TCS 二进制或 TCS BIN)是面向比特的协议。它定义了蓝牙设备间建立语音和数据呼叫的控制信令,定义了处理蓝牙 TCS 设备群的移动管理进程。基于 ITU-T Q.931 建议的 TCS 二进制被指定为蓝牙的二元电话控制协议规范。

另外, SIG 还根据 ITU-T V.250 建议和 GSM 07.07 定义了控制多用户模式下移动电话和调制解调器和可用于传真业务的 AT 命令集。

5. 选用协议

a. 点对点协议 (PPP)

在蓝牙技术中, PPP 位于 RFCOMM 上层,完成点对点的连接。

b. UDP/IP/TCP

UDP/IP/TCP 协议由 Internet 工作任务组(IETF)制定,广泛应用于互联网通信,在蓝牙设备中使用这些协议是为了与互联网相连接的设备进行通信。

c. 对象交换协议 (OBEX)

IrOBEX(简称为 OBEX)是由红外数据协会(IrDA)制定的会话层协议,它采用简单的和自发的方式交换对象。OBEX 是一种类似于 HTTP 的协议,这里假设传输层是可靠的,采用客户机/服务器模式,独立于传输机制和传输应用程序接口(API)。

d. 电子名片交换格式(vCard)、电子日历及日程交换格式(vCal)都是开放性规范,它们都没有定义传输机制,而只是定义了数据传输模式。SIG 采用 vCard/vCal 规范,是为了进一步促进个人信息交换。

e. 无线应用协议 (WAP)

无线应用协议由无线应用协议论坛制定,它融合了各种广域无线网络技术,其目的是将互联网内容和电话债券的业务传送到数字蜂窝电话和其他无线终端上。选用 WAP 可以充分利用为无线应用环境(WAE)开发的高层应用软件。

1.5 蓝牙应用模型及协议栈

蓝牙 SIG 定义许多蓝牙应用模型,下面分别进行简述。

1. 文件传输应用模型

文件传输应用模型提供两个终端之间的数据通信功能，可传输.xls、.ppt、.wav、.jpg 和.doc 文件及其他文件，以及完整的文件夹或目录或多媒体数据流等并提供远程文件夹浏览功能。文件传输协议栈如图 1.2 所示。

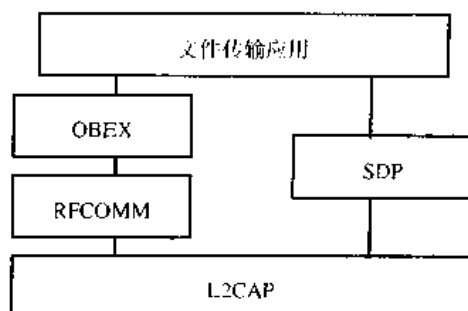


图 1.2 文件传输协议栈

2. 互联网网桥模式

在这种用户模式下，由手机或无线调制解调器向 PC 提供拨号入网和收发传真的功能，而不必与 PC 建立物理连接。拨号上网需要两个协议栈（不包括 SDP），如图 1.3 所示。

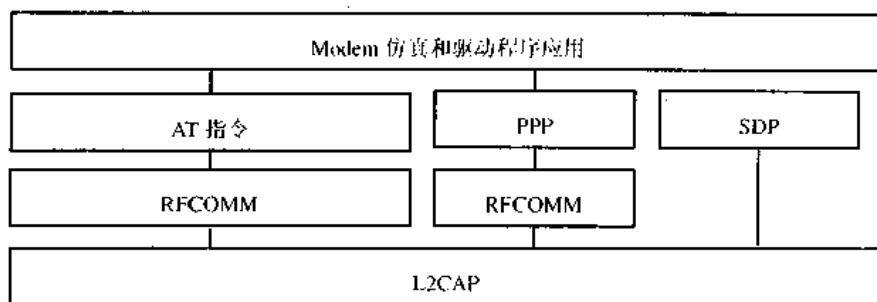


图 1.3 互联网网桥模式

AT 命令集用来控制移动电话或调制解调器以及传送其他业务数据的协议栈。传真采用类似协议栈，但不使用 PPP 及基于 PPP 的其他网络协议，而由应用软件利用 RFCOMM 直接发送。

3. 局域网访问模式

在此用户模式下，多功能数据终端（DT）经局域网访问点（LAP）无线接入局域网，接入后 DT 的操作与通过拨号方式接入局域网设备的操作一样，其协议栈如图 1.4 所示。

4. 同步模式

同步用户模式提供设备到设备的个人资料管理（PIM）的同步更新功能，其典型应用如电话簿、日历、通知和记录等，它要求微机、蜂窝电话和个人数字助理（PDA）在传输和处理名片、日历及任务通知时，使用通用的协议和格式。协议栈如图 1.5 所示，其中同步应用模式代表红外移动通信（IrMC）客户机或服务器。

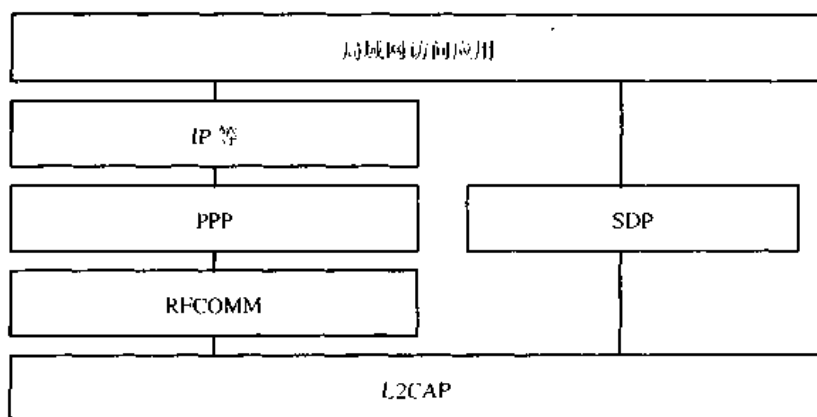


图 1.4 局域网访问模式

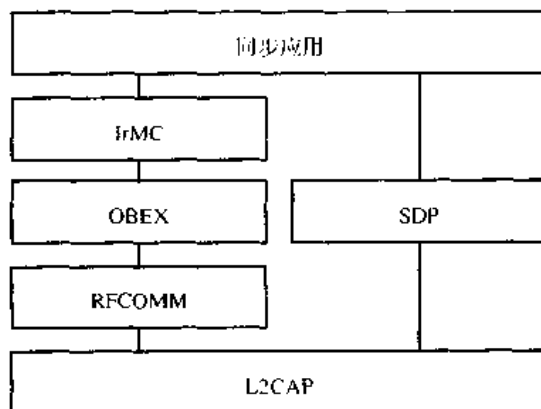


图 1.5 同步模式

5. 三合一电话模式

手持电话机有三种使用方法：接入公用电话网，作为普通电话使用；作为不计费的内部电话使用；作为蜂窝移动电话使用。无线电话和内部电话使用相同的协议栈：语音数据直接与基带协议连接,不经过 L2CAP 层，如图 1.6 所示。

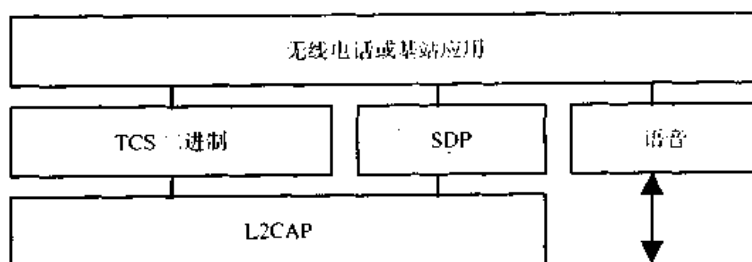


图 1.6 三合一电话模式

6. 头戴式设备模式

使用该模式，用户打电话时可自由移动。通过无线连接，头戴式设备通常作为蜂窝电话、无线电话或个人微机的音频输入输出设备。头戴式设备协议栈如图 1.7 所示，语音数据流不经过 L2CAP 层而直接接入基带协议层。头戴式设备必须能收发并处理 AT 命令。

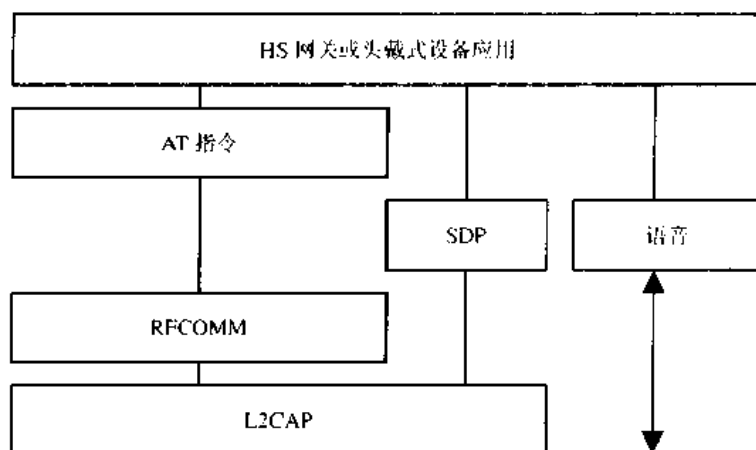


图 1.7 头戴式设备模式

1.6 蓝牙技术的应用

手机与计算机相连。目前手机多数通过 IrDA 红外线或 RS232 串口线与计算机相连，蓝牙技术可以取而代之，不仅方便，而且资料传送的速度更快（有些情况下 IrDA 的速度更快些），也许将来手机下部的连接器也会消失，或是变得更简单。

可作无绳电话使用。内置蓝牙芯片的手机，在家里可以当作无绳电话使用，不用双向收费，节省手机费用。当然离开屋子一段距离后便回自动切换至无线网络基站上。

数据共享，办公更方便。无论是手机、计算机、PDA、打印机，或是数码相机、MP3 播放器、D 播放器等都可以利用蓝牙技术互传语音、文字、图像、文件等。蜘蛛网式的会议室将不复存在，白板纪录仪、摄影机等都可以利用蓝牙技术来简化操作。

Internet 接入。内置蓝牙芯片的笔记本型计算机或手机等，不仅可以使 PSTN（公用电话交换网）、ISDN、LAN、xDSL（如 ADSL）等接入，而且可以使用蜂窝式移动网络进行高速连接。

无线免提。笔记本型电脑具有话筒和喇叭，用蓝牙技术连接将来的手机（也许是宽带网），可使多人视频会议更为容易。免提手机也不再会汽车独有。

同步资料。无论在办公室或家里，你的 Note Book、手机或是 PDA 可通过蓝牙产品及相应程序，与其他设备同步。内部信息永保最新。当然 E-mail 也可以实时接收并同步输入计算机，而且 E-mail 可以在飞机上完成，下机后自动发出。

影像传递。这有点类似 NODIA9110 的影像传输方式，但更加简单。带有蓝牙功能的数码相机在拍摄完成后，影像传至手机后可直接送至世界任何一个角落，记者特别需要这一功能。当然也可以直接将影像送入打印机，即拍即现。

蓝牙技术还可应用于键盘、鼠标、家庭网络、高速无线内部网络、电子名片等方面。

第2章 基带层协议

2.1 概述

蓝牙是一种用于替代移动设备或固定电子设备之间连接电缆的近距离无线链路。其关键特性是健壮性、低复杂性、低功耗和低成本。

蓝牙工作在全球通用的 2.4GHz 的 ISM 频段, 并采用跳频收发信机来达到抗干扰和抑制信号衰减的作用, 采用二进制调频(FM)模式降低收发信机的复杂性, 其符号速率为 1ms/s。划分为时隙的信道采用 625 μ s 的标称时隙长度。蓝牙系统采用全双工分时(TDD)传输方案实现双工传输。在信道中, 信息可以分组方式进行交换。各信息分组可采用不同跳频实现传输。理论上讲, 一个分组覆盖一个单时隙, 而实际上一个分组可扩展至覆盖 5 个时隙。

蓝牙协议使用电路交换和分组交换的混合方式。时隙保留用于同步分组。同时, 蓝牙能够支持一条异步数据信道, 乃至三个同步语音信道, 或一条同时支持异步数据和同步语音的信道。每一语音信道在每一方向上支持 64kb/s 同步语音信道连接。异步信道最大可不对称支持 723.2kb/s (回程为 57.6kb/s), 或对称支持 433.9kb/s 的传输速率。

蓝牙系统由无线部分、链路控制部分、链路管理支持部分和主终端接口组成, 如图 2.1 所示。本章详细阐述执行基带协议和其他低层链路规则的蓝牙链路器内容。用于链路设置及控制的链路层信息在链路管理协议章节中再作详细介绍。

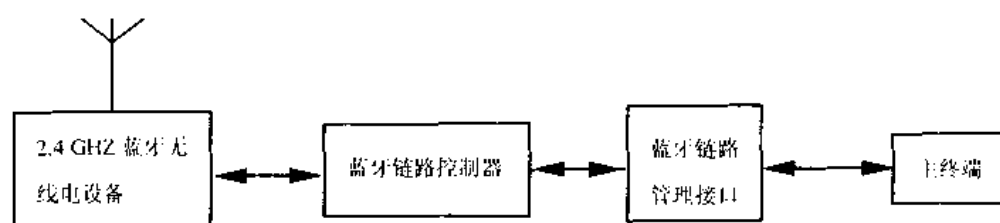


图 2.1 蓝牙系统结构

蓝牙系统提供点对点连接方式或一对多连接方式, 其连接方式如图 2.2 所示。

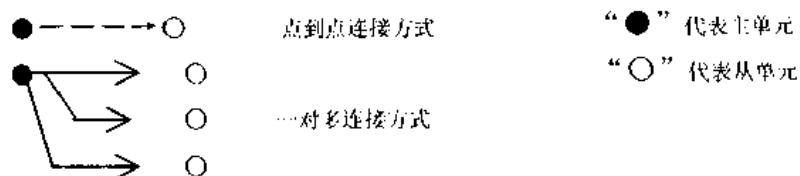


图 2.2 蓝牙系统连接方式

在一对多连接方式中, 多个蓝牙单元之间共享一条信道。共享同一信道的两个或两个以上的单元形成一个匹克网(Piconet)。其中, 一个蓝牙单元作为匹克网的主单元, 其余则为从单元。在一个匹克网中最多可有 7 个活动从单元。另外, 更多的从单元可被锁定于某一主单元, 该状态称为休眠状态。在该信道中, 不能激活这些处于休眠状态的从单元, 但仍可使之与主单元之间保持同步。对处于激活或休眠状态的从单元而言, 信道访问都是由主单元

进行控制。

具有重叠覆盖区域的多个匹克网构成一个散射网络（Scatternet）结构，如图 2.3 所示。每一匹克网只能有一个主单元，从单元可基于时分复用参加不同的匹克网。另外，在一个匹克网中的主单元仍可作为另一个匹克网的从单元，各匹克网间不必以时间或频率同步。各匹克网各有自己的跳频信道。

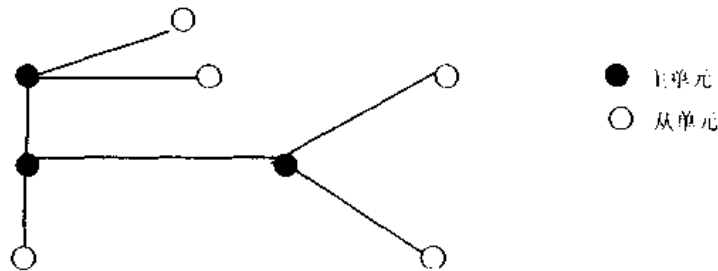


图 2.3 散射网络结构

2.2 物理信道

1. 频带及射频（RF）信道

蓝牙技术工作在 2.4 GHz 的 ISM 频段。虽然该频段为全球通用，但实际上准确的频率和带宽在各国有一些差异。在美国和欧洲，使用的带宽为 83.5 MHz，在该频段里，以 1 MHz 的带宽为间隔设立了 79 个射频跳频点。在日本、西班牙和法国，缩减了带宽，在该频段里设立了 23 个射频跳频点，其带宽仍以 1 MHz 为间隔，如表 2.1 所示。

表 2.1 可用射频信道

地 区	频 率 范 围	射 频 信 道
欧洲及美国	2400~2485 MHz	$F=2402+k \text{ MHz}$ $k=0, 1, \dots, 78$
日 本	2471~2497 MHz	$F=2473+k \text{ MHz}$ $k=0, 1, \dots, 22$
西 班 牙	2445~2475 MHz	$F=2449+k \text{ MHz}$ $k=0, 1, \dots, 22$
法 国	2446.5~2483.5 MHz	$F=2454+k \text{ MHz}$ $k=0, 1, \dots, 22$

2. 信道定义

信道使用一组伪随机跳频序列，经 79 或 23 个射频跳频点的跳频序列来表示。跳频序列对匹克网是惟一的，而且由主单元蓝牙设备编址确定，跳频序列的相位由主单元蓝牙时钟确定。信道被划分为时隙（时间片）的形式，且每一时隙对应一个 RF 跳频点。连续跳频则对应于不同 RF 跳频模式，理论跳频速率为 1600 跳/秒，参加匹克网的全部蓝牙单元与信道保持时间和跳频同步。

3. 时隙

信道被分成长度为 625μs 的时隙。时隙依据匹克网主单元蓝牙时钟来编号。时隙编号区域为 $0 \sim 2^{27}-1$ 且循环周期是 2^{27} 。在各时隙中，主单元和从单元都能够传输分组。

由于蓝牙系统中主、从单元的分组传输采用分时双工（TDD）交替传输方式，所以在

系统中主单元采用偶数编号时隙开始信息传输，而从单元则采用奇数编号时隙开始信息传输。分组起始位置与时隙起始点相吻合。由主或从单元传输的分组可以扩展到 5 个时隙，TDD 和定时工作方式如图 2.4 所示

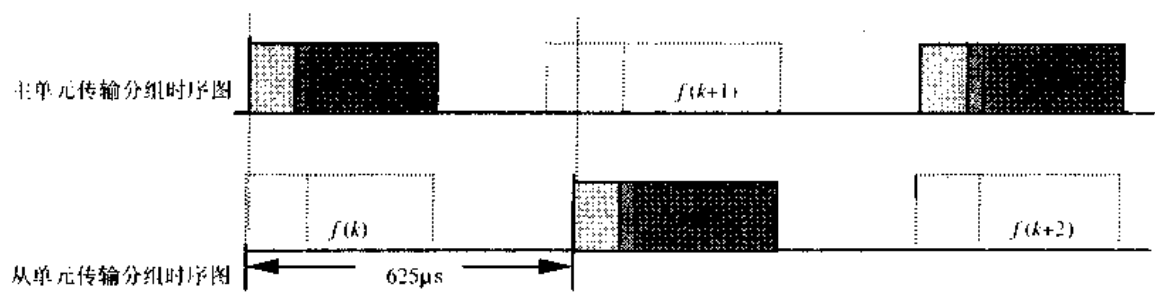


图 2.4 主从单元传输分组时序图

射频（RF）跳频在分组传输期间保持不变。对于单时隙分组，RF 跳频以当前蓝牙时钟值作为基点。对于多时隙分组来讲，RF 跳频以蓝牙中第一个分组时隙的时钟值作为整个分组基点。在多时隙分组的第一个时隙里的 RF 跳频将被认为由当前蓝牙时钟值确定的频率。图 2.5 举例说明了单时隙分组和多时隙分组的跳频定义。若分组占有的时隙多于一个时，采用的跳频频率就是用于以分组开始传输的时隙采用的跳频频率。

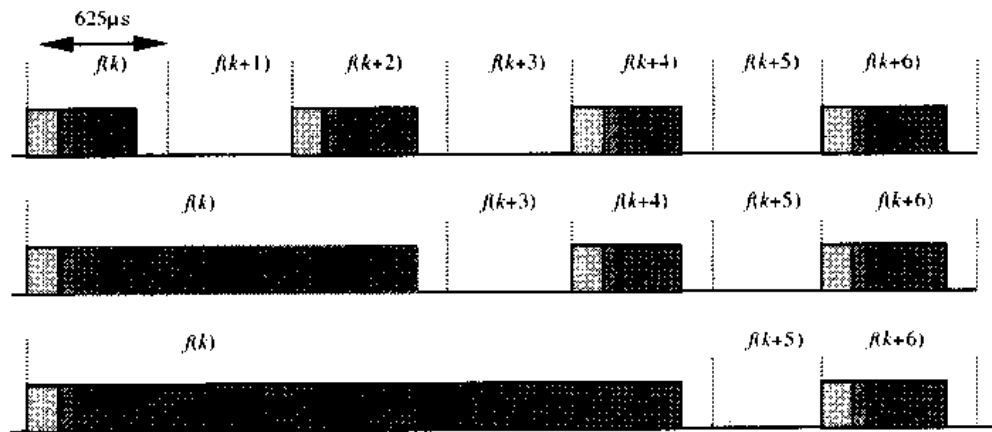


图 2.5 多时隙分组

4. 调制与比特率

数据以 1Mb/s 的速率进行传输，使用高斯型二进制 FSK 模式。二进制“1”代表正频偏，二进制“0”代表负频偏，最大频偏在 140 kHz~175 kHz 之间。

2.3 物理链路

在主单元和从单元之间，可以建立不同类型的链路，如同步面向链接（SCO）链路、异步无链接（ACL）链路。

同步面向链接（SCO）的目的是在匹克网中的主单元和从单元之间实现点到点链接，主单元通过在规则间隔上使用保留时隙保持 SCO 链接。而 ACL 链接是主单元与共存于匹克网中的所有从单元之间实现一对多的链接方式。在非 SCO 链接保留时隙上，主单元可以以时

隙为单位建立到任何其他从单元的 ACL 链接，且连接的从单元包括已处于 SCO 链接方式中的从单元。

1. SCO 链接

SCO 链接是在主单元与指定的从单元之间实现的对称的、点到点链接。SCO 链接方式采用保留时隙来传输分组，因此该方式可看作是在主单元和从单元之间实现的电路交换链接。SCO 链接主要用于支持类似于象语音这类时限信息。从主单元方面看，它可以支持多达 3 路的指向相同从单元或不同从单元的 SCO 链接。而从从单元方面看，针对同一主单元可以支持多达 3 路的 SCO 链接。若链接来自于不同主单元，此时从单元只能支持 2 路 SCO 链接。

主单元在规则间隔上发送分组，该规则间隔称为主从保留时隙中的到从单元的 SCO 间隔 T_{SCO} （以时隙为单位）。如果另一从单元未在前一主从时隙中标识，则在后面的主从时隙中，SCO 从单元通常应允许应答 SCO 分组。如果 SCO 从单元对分组头中从单元地址解码失败，在保留 SCO 时隙里它仍允许返回一个 SCO 分组。

SCO 链接由通过 LMP 协议发送 SCO 设置消息的主单元建立。该消息含定时参数（如 SCO 间隔 T_{SCO} 和时隙补偿 D_{SCO} ），用于定义保留时隙。

为防止时钟卷绕问题，在 LMP 中设置信息的初始化标志时应指出是采用初始化方式 1 还是初始化方式 2。从单元将通过初始化标志指示采用的初始化模式。若当前主时钟（CLK₂₇）的 MSB 是 0，主单元使用初始化模式 1；当前主时钟（CLK₂₇）的 MSB 是 1 时，主单元使用初始化模式 2。由主从单元保留的主从 SCO 时隙将在满足下述等式的时隙上被初始化。

$$CLK_{27-i} \bmod T_{SCO} = D_{SCO} \quad \text{初始化方式 1}$$

$$(CLK_{27-1}, CLK_{26-1}) \bmod T_{SCO} = D_{SCO} \quad \text{初始化方式 2}$$

从主 SCO 时隙紧跟保留主从 SCO 时隙。初始化后，作为下一主从 SCO 时隙的时钟值 CLK(k+1)，可以通过增加固定间隔 T_{SCO} 到当前主从 SCO 时隙时钟值来建立。

$$CLK(k+1) = CLK(k) + T_{SCO}$$

2. ACL 链接

在非 SCO 链接保留时隙里，主单元可以时隙为单位与任何从单元交换分组。ACL 链接提供在主单元与所有在匹克网中活动从单元的分组交换链接，并可采用异步和等时两种服务方式。在一个主单元和一个从单元之间，只能存在一个 ACL 链接。对于大多数 ACL 分组，分组重传的目的在于确保数据的完整性。

在从一主时隙里，当且仅当先前的主一从时隙已被编址时，从单元允许返回一个 ACL 分组。如果在分组头的从单元地址解码失败，则不允许传输该分组。

未指定目的从单元的 ACL 分组可视为广播分组，且可由各从单元读出。如果在 ACL 链接上没有数据可传输且未进行轮询申请，那么就不会发生任何传输过程。

2.4 分组

2.4.1 通用格式

在基带里定义分组和消息时，编码序列必须遵循下列规则（即 Little Endian 格式）。

- b_0 代表最低标识位 (LSB);
- LSB 是第一个无线发送位;
- 在示例中, LSB 置于最左边。

基带控制器认为来自高层软件层的第一位是 b_0 即: 这是经无线发送的第一位。各数据段在基带内部生成, 如分组头信息和有效载荷头长度信息等, 都以 LSB 首先发送。例如: $X=3$ 的 3 位参数, 其传输码值是 $b_0 b_1 b_2=110$, 位“1”首先经无线发送, 最后才是位“0”。

匹克网信道中的数据以分组形式传输, 标准格式如图 2.6 所示。每一分组由三个部分组成: 识别码、头和有效载荷, 图中给出了每个部分所占的位数。



图 2.6 标准分组格式

识别码和头是一个固定值, 分别用 72 位和 54 位表示。有效载荷长度范围从 0 到 2745 位。分组具有几种不同的类型格式, 如分组可仅由识别码组成 (压缩格式, 可参看本书中有关 ID 分组的内容), 也可以用识别码和头组成, 或识别码—头—有效载荷分组。

2.4.2 识别码

每个分组都是用识别码作开始, 若头信息紧随其后, 则识别码长度是 72 位, 否则识别码长度是 68 位。识别码主要用于同步、DC 补偿平衡和识别。识别码识别所有在匹克网信道上交换的分组。在同一匹克网中发送的所有分组都先于同一信道识别码。在蓝牙单元接收机中, 滑动相关器关联于识别码, 且当超过门限电平时被激发, 该激发信号被用于确定接收定时。识别码也可用于呼出和查询过程。在这种情况下, 识别码自身就被当作一个信令消息, 而不必给出头和有效载荷。

识别码由头、同步字组成, 有时也包括尾, 如图 2.7 所示。

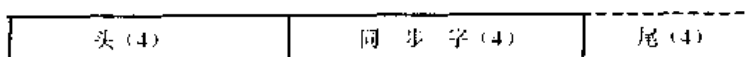


图 2.7 识别码格式

1. 识别码类型

存在三种不同类型的识别码:

- 信道识别码 (CAC);
- 设备识别码 (DAC);
- 查询识别码 (IAC)。

各识别码类型用于处于不同操作模式的蓝牙单元中。信道识别码标识一个匹克网, 代码包含在匹克网信道上所有交换的分组中。设备识别码用作一个指定信令过程, 如呼出或呼出应答过程。对于查询识别码, 存在两个变量: 一个称为通用查询识别码 (GIAC), 为所有设备通用, 可用于检测在指定范围内有否其他蓝牙单元; 另一个称为专用查询识别码 (DIAC), 该查询识别码为在蓝牙系统中具有公共属性的专用设备组使用, 可以用于发现在指定范围内符合条件的专用蓝牙单元。

CAC 由头、同步字和尾组成, 而且它的整个长度是 72 位。当 CAC 当作没有头的自包含消息使用时, 则 DAC 和 IAC 就不能包含尾且长度值是 68 位。

不同识别码类型使用不同的低地址部分（LAP）来创建同步字。关于基带（BD）地址的 LAP 段问题参见蓝牙设备的编址部分内容，表 2.2 列出不同识别码类型概要。

表 2.2 识别码类型小结

代码类	LAP	代码长度	注释
CAC	主单元	72	具体内容见 识别码部分
DAC	呼入单元	68/72 ⁺	
CIAC	保留	68/72 ⁺	
DIAC	专用	68/72 ⁺	

※ 注释长度为 72，只能配合用于 FHS 分组。

2. 头

头是用于 DC 补偿的为固定 0-1 模式的 4 位符号标志。该序列可以是 1010 或是 0101，这取决于后续同步字的 LSB 是 1 或 0。如果 LSB 是 1，头序列就是 1010；如果 LSB 是 0，头序列就是 0101。

3. 同步字

同步字是一个来自于 24 位地址（LAP）的 64 位代码字。对于 CAC，使用主单元 LAP；对于 GIAC 和 DIAC，使用保留和专用的 LAP；对于 DAC，使用从单元 LAP。对于不同的 LAP，该构成在同步字之间建立一个大汉明空间。另外，同步字的良好自相关特性可以改善定时同步进程。同步字的推演过程参见识别码部分内容。

4. 尾

这是一种 CAC 的典型情况。然而尾也可用于 DAC 和 IAC，这时这些代码用于在呼叫应答和查询应答过程中的 FHS 分组交换。

尾也是一个固定 0-1 模式的四个符号标志。尾与同步字的 3 位 MSB 一起形成一个用于扩展 DC 补偿的 0, 1 交替的 7 位模式。尾序列究竟是 1010 或是 0101，取决于同步字的 MSB 是 0 还是 1。当同步字的 MSB 是“0”时，CAC 中的尾是 1010；当同步字的 MSB 是“1”时，CAC 中的尾是 0101。

2.4.3 分组头

分组头包含链路控制（LC）信息，由六个段组成。

- AM_ADDR 3 位，活动成员地址；
- TYPE 4 位，类型码；
- FLOW 1 位，流控制；
- ARQN 1 位，确认指示；
- SEQN 1 位，序列号；
- HEC 8 位，头错误校验。

包含 HEC 的整个头信息由 18 位组成，如图 2.8 所示。该头信息以 1/3 比例前向纠错码编码（有关内容详见本书的纠错部分），因此，头信息最后成为 54 位编码格式。但要注意，AM_ADDR 和 TYPE 信息段的 LSB 首先发送。

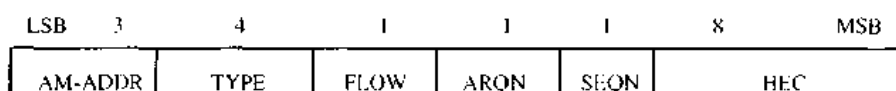


图 2.8 头格式

1. AM_ADDR

AM_ADDR 代表成员地址用来区分参加匹克网的不同活动成员。匹克网中，可以有一个或多个从单元与主单元相连。为了区分识别每个从单元，各从单元在它们处于活动状态时都将分配一个临时的 3 位地址值。分组在主单元和从单元之间进行交换时，都将携带该从单元的 AM_ADDR 信息，即在主—从分组和从—主分组里都要使用从单元的 AM_ADDR。

若主—从单元之间采用广播分组方式，那么保留使用各位全为 0 的地址。但 FHS 分组格式例外，FHS 分组格式可以使用“全 0”成员地址，但它不是广播消息。失去连接或休眠的从单元将放弃它们的 AM_ADDR。但当它们重新进入匹克网时，则必须重新分配新的 AM_ADDR。

2. TYPE

分组可以有 16 种不同类型。4 位类型码正好给出了这十六种不同类型结构。特别值得注意的是，类型码的解释取决于与分组相关的物理链路类型。首先它需要确认分组是以 SCO 链接或是以 ACL 链接发送，其次它还需要确认是以 SCO 分组或 ACL 分组的哪种类型接收。同时类型码也表示当前分组将占有多少个时隙。这种方法使未编址的接收设备不得在其余时隙持续期间中侦听信道。

3. FLOW

FLOW 用于 ACL 链接上的分组流量控制。当接收方用于 ACL 链接的 RX 缓冲区已满或非空时，停止 (STOP) 指示 (FLOW=0) 将暂时停止数据传输。注意，STOP 信令只涉及 ACL 分组。包含链路控制信息 (ID, POLL 和 NULL 分组) 的分组或 SCO 分组仍可继续接收。当 RX 缓冲区为空时，将返回继续 (GO) 指示信息 (FLOW=1)。当没有接收到分组或接收头信息有错时，则 GO 信息将隐式给出。

4. ARQN

ARQN 指示用于将含有 CRC 的有效载荷数据通知成功转发源单元。ARQN 可以是一个以 ACK 表示的有效确认或是一个以 NAK 表示的无效确认。若接收成功，则返回 ACK (ARQN=1)，否则返回 NAK (ARQN=0)。当没有接收到涉及确认返回信息时，系统将以 NAK 形式隐含指示出来，实际上我们可将 NAK 看作是默认的信息。

ARQN 在返回分组头信息里稍带确认。接收正确的校验由循环冗余校验 (CRC) 码来校验。使用未编码的 ARQ 方式意指 ARQN 与来自同一源的最后接收到的分组有关。

5. SEQN

SEQN 提供一个序列码方式来排列分组流的顺序。对每个包含 CRC 数据的传输分组，SEQN 位将反相。这就要求在接收点滤出重传过程。重传过程是因为 ACK 失败，导致收端将重复接收一次同样的分组。通过对相邻分组的 SEQN 比较，正确接收的重传过程就可以

不考虑。SEQN 必须存在头格式里，究其原因是因为在未编码的 ARQ 方式里缺少分组编号。关于 SEQN 位的初始化及如何合理使用该信息位，在后面有关章节中叙述。

6. HEC

为检测头的完整性，每个头都有一个“头校验错”信息字。HEC 由 8 位字组成，该字由多项式 647（八进制数）生成。HEC 生成器用 8 位值进行初始化。若 FHS 分组以主呼叫应答状态发送，从单元使用高地址部分（UAP）。若 FHS 分组以查询应答方式发送，就使用缺省校验初始化（DCI）。在其他情况，主设备都采用 UAP 方法。关于蓝牙系统中设备编址的定义，请参看蓝牙编址章节内容。在初始化后，HEC 形成 10 位头。在校验 HEC 之前，接收装置必须以适当的 8 位 UAP（或 DCI）来初始化 HEC 校验电路。如果 HEC 没有校验，则忽略整个分组。

2.4.4 分组类型

在匹克网上使用的分组类型与它们使用的物理链接方式有关。直到现在我们仅涉及到 SCO 和 ACL 两种链接方式。针对这两种链接方式的任一方式，都有 12 种不同类型的分组能被使用；另外四种控制分组为所有链接模式公用，它们的类型码是惟一的且与用什么链接类型方式无关。为区分链接分组上的不同分组类型，采用了四位类型码的不同组合来表示，并将分组类分成四个段。第一段为四个控制分组公用所有物理链接类；第二段为占有单时隙的分组，有六种分组类型；第三段为占有三时隙的分组，有两种分组类型；第四段为占有五时隙的分组，有两种分组类型。时隙占有在分段上反映出来，而且能直接从类型码得到。表 2.3 描述了上述讨论的 SCO 和 ACL 两种链接方式分组。

表 2.3 分组类型

段	类型码	占有时隙	SCO 链接	ACL 链接
1	0000	1	NULL	NULL
	0001	1	POLL	POLL
	0010	1	FHS	FHS
	0011	1	DM1	DM1
2	0100	1	未定义	DH1
	0101	1	HV1	未定义
	0110	1	HV2	未定义
	0111	1	HV3	未定义
	1000	1	DV	未定义
	1001	1	未定义	AUX1
3	1010	3	未定义	DM3
	1011	3	未定义	DH3
	1100	3	未定义	未定义
	1101	3	未定义	未定义
4	1110	5	未定义	DM5
	1111	5	未定义	DH5

1. 公用分组类型

公用分组共有五个，除在表 2.3 中段 1 所列的类型外，还有 ID 分组。下面对各个分组分别进行介绍。

a. ID 分组

ID 或身份分组由设备识别码 (DAC) 或查询识别码 (IAC) 构成，其长度为 68 位。由于接收设备使用位环行解调电路来匹配接收分组以确认 ID 分组的位序列，所以 ID 分组是一种非常可靠的分组。为此，ID 分组常用于呼叫，查询及应答过程中。

b. NULL 分组

NULL 分组是没有携带有效载荷的分组，它仅由信道识别码和分组头组成，它的总长度 (固定) 为 126 位。NULL 分组用来返回链接信息给发送端。用先前传输 (ARQN) 是否成功或当前收端 RX 缓冲区 (FLOW) 的状态来说明。NULL 分组自身并不需要确认。

c. POLL 分组

POLL 分组非常类似于 NULL 分组，它也不携带有效载荷。与 NULL 分组相比，它需要一个从收端来的确认。POLL 分组并不影响 ARQN 和 SEQN 段。在 POLL 分组的收端从单元必须用一个分组来应答，该返回分组是 POLL 分组的一个隐含答复。这种分组可用于在匹克网中主单元查询从单元的过程。在这种过程中，就是主单元没有任何信息送出，从单元也必须应答。

d. FHS 分组

FHS 分组是一种专用控制启动分组，其中包括蓝牙设备地址和发端时钟，有效载荷包含 144 位信息和 16 位 CRC 码。有效载荷采用 2/3 比例前向纠错码，因此总有效载荷长度达 240 位。FHS 分组覆盖一个单时隙。FHS 有效载荷及其格式如图 2.9 表示。

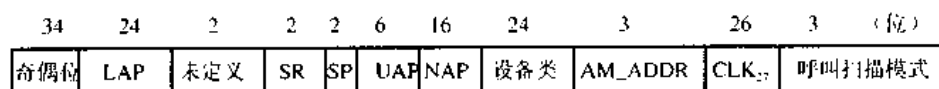


图 2.9 FHS 有效载荷及其格式

在 FHS 分组中有效载荷有 11 个段，这样 FHS 分组一般常用于呼叫主应答、查询应答和主从切换。在呼叫主应答和主从切换中，在应答被确认或超时超出之前，它都是一个可重复传输的分组。在查询应答中，对 FHS 分组可不必确认。FHS 分组含有实时时钟信息，时钟信息在每次重传之前被修正。它们的相同部分在每次重传过程中，FHS 有效载荷的重传稍有点不同于普通有效载荷的重传。

FHS 分组在匹克网信道被确定之前或当从现有的一个匹克网转到一个新的匹克网时，使用同步跳频技术。在前一种情况里，收端若还没有被分配一个活动成员地址，FHS 分组头里的 AM_ADDR 段就设成一个全“0”。然而，FHS 分组就不再考虑作为广播分组。在后一种情况里，在当前匹克网中从单元已有一个 AM_ADDR 时，FHS 分组的同步信息可以使用在下面列出的 FHS 中各段的描述。

奇偶位：34 位。该段发送 FHS 分组主体识别码同步字第一部分的奇偶位，这些位来自于 LAP。具体内容参见本书中识别码部分内容。

LAP：24 位。该段含有发送 FHS 分组主体的低地址部分。

未定义：2 位。该段作为将来应用保留且被清除为“0”。

SR：2 位。该段用来扫描重复段和指示在两个连续扫描窗口之间的间隔，其内容如表 2.4

所示。具体内容见本书中识别码部分内容。

SP: 2 位。该段为扫描周期段并指出在查询应答信息被传输后，命令呼叫扫描模式的周期。SP 段内容如表 2.5 所示。

UAP: 8 位。该段含有发送 FHS 分组单元的高地址部分。

NAP: 16 位。该段含有发送 FHS 分组单元的非有效地址部分。

设备类: 24 位。该段含有发送 FHS 分组单元的设备类。

AM_ADDR: 3 位。若 FHS 分组用于呼叫建立或主—从交换，它含有将被用作接收的成员地址。若仅发送 FHS 分组，从单元回复主单元应答或单元应答查询申请分组就含了全“0” AM_ADDR 段。

CLK₂₇₋₂: 26 位。该段用于全发送 FHS 分组部件的本地系统时钟值。就 FHS 分组识别码传输示例来说，该时钟值有 1.25ms（两个时隙间隔）的分辨率。每次新的传输，该段都被修改，所以它准确反映了实时时钟值。

呼叫扫描模式: 3 位。呼叫扫描模式的详细定义如表 2.6 所示。这里是基本支撑一种命令扫描模式及多达三种可选扫描模式。

表 2.4 SR 段内容

SR 位格式 b ₁ b ₀	SR 模式
00	R0
01	R1
10	R2
11	保留

表 2.5 SP 段内容

SP 位格式 b ₁ b ₀	SP 模式
00	P0
01	P1
10	P2
11	保留

表 2.6 呼叫扫描模式段内容

位格式 b ₂ b ₁ b ₀	呼叫扫描模式
000	命令扫描模式
001	选择扫描模式 I
010	选择扫描模式 II
011	选择扫描模式 III
100	保留备用
101	保留备用
110	保留备用
111	保留备用

LAP、UAP 和 NAP 共同形成发送 FHS 分组单元的 48 位 IEEE（国际电子电气工程师协会）地址。使用奇偶位和 LAP，接收单元可直接创建 FHS 分组发送者的信道识别码。

e. DM1 分组

在任何链接类型里为支持控制信息，DM1 服务作为段 1 的部分。尽管如此，它也经常传输用户数据。由于 DM1 分组在 SCO 链接上被认可，所以它也可中断同步信息去传送控制信息。关于 DM1 分组作为 ACL 分组认可问题，可参见本书 ACL 分组内容。

2. SCO 分组

SCO 分组用于同步 SCO 链接，分组不包括循环冗余检测（CRC）码，而且不允许重传。SCO 分组发送到同步 I/O（语音）端口。另外，SCO 分组除含有同步语音段外，还含有同步数据段。SCO 分组到目前为止主要用于 64kb/s 的语音传输。

a. HV1 分组

HV1 分组含有 10 个信息字节。该字节使用 1/3 比例前向纠错码保护，未使用 CRC 码。有效载荷长度被固定在 240 位，无有效载荷头。

HV1 分组的典型应用是语音传输，HV1 支持高保真语音，语音分组不可重复传输且不需要 CRC 码。HV1 分组可载有 64kb/s 速率的 1.25ms 语音信息，在这种情况下，HV1 分组每两个时隙 ($T_{SCO}=2$) 必须进行一次传输。

b. HV2 分组

HV2 分组含有 20 个信息字节。该字节使用 2/3 比例前向纠错码保护，未使用 CRC 码。有效载荷长度被固定在 240 位，无有效载荷头。

若 HV2 分组用作 64kb/s 速率的语音传输，它可载有 2.5ms 的语音信息，在这种情况下，HV2 分组每四个时隙 ($T_{SCO}=4$) 必须进行一次传输。

c. HV3 分组

HV3 分组含有 30 个信息字节。该字节没用比例前向纠错码保护，也未使用 CRC 码。有效载荷长度被固定在 240 位，无有效载荷头。

若 HV3 分组用作 64kb/s 速率的语音传输，它可载有 3.75ms 的语音信息，在这种情况下，HV3 分组每六个时隙 ($T_{SCO}=6$) 必须进行一次传输。

d. DV 分组

DV 分组由数据—语音分组组成。有效载荷被分成 80 位的语音段和高达 150 位的数据段，其格式如图 2.10 所示。

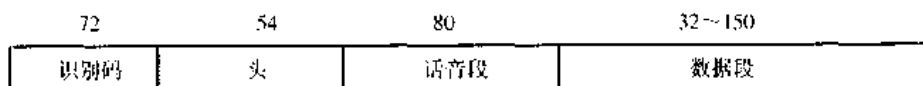


图 2.10 DV 分组格式

语音段不由 FEC 保护，数据段含有 10 个信息字节（包含 1 字节有效载荷头）和 CRC。数据段用 2/3 比例前向纠错码编码，必要时使用填入附加“0”的方法来保证有效载荷位的总数先于比例前向纠错码，且是 10 的整倍数。因 DV 分组含有同步（语音）内容，所以 DV 分组必须是以有规律的间隔进行传输，所以它在 SCO 分组类下列出。语音和数据段完全分别处理。语音段如一般 SCO 数据一样处理而且不允许重复传输，即：语音段传输的总是新的信息。数据段可以进行错误校验工作，必要时数据信息可重复传输。

3. ACL 分组

ACL 分组用于异步链接方式。分组内信息可以是用户数据或是控制数据。目前已定义了七种 ACL 分组，其中前六种含有 CRC 码，若非正常接收情况确认已收到，ACL 分组可重传（执行刷新操作过程除外）；第七种分组（AUX1 分组），没有 CRC 码且不能重传。

a. DM1 分组

DM1 分组是一种只能携带数据信息的分组。DM 是中速数据的表示。有效载荷包括多到 18 个字节信息的（其中一个字节是有效载荷头）和 16 位 CRC 码。DM1 分组可覆盖一个单时隙。有效载荷加上 CRC 位用 2/3 前向比例纠错码编码，形成每 10 位信息段加上五位奇偶位。必要时，CRC 位后增补一些“0”来保证总数位（信息位、CRC 位和尾位）为 10 的整倍数。DM1 分组内的有效载荷头仅一字节长，在有效载荷头里的长度指示器指出了用户字节数（有效载荷头和 CRC 码除外）。

b. DH1 分组

该分组类似于 DM1 分组。除分组里有效载荷外，其余信息都没有用 FEC 编码。因此，DH1 分组可多达 28 个信息字节加 16 位 CRC 码。DH 是高速数据的表示。DH1 可以覆盖单时隙。

c. DM3 分组

DM3 分组是一种使用扩展有效载荷的 DM1 分组。DM3 分组可覆盖 3 个时隙。有效载荷包括多达 123 个字节的信息（其中两个字节是有效载荷头）和 16 位 CRC 码。DM3 分组的有效载荷头仅两字节长，在有效载荷头里的长度指示器指出了用户字节量（有效载荷头和 CRC 码除外）。当 DM3 分组进行发送或接收时，在三时隙持续期间，RF（射频）跳频不发生改变（第一个时隙是信道识别码传输时隙）。

d. DH3 分组

该分组类似于 DM3 分组。除分组里有效载荷外，其余信息都没用 FEC 编码。为此，DH3 分组可多达 185 个信息字节（包括两字节信息头）加 16 位 CRC 码。

DH3 分组可复盖三个时隙。当 DH3 分组被发送或接收时，在三时隙持续期间，跳频不发生改变（第一个时隙是信道识别码传输时隙）。

e. DM5 分组

DM5 分组是一种使用扩展有效载荷的 DM1 分组。DM5 分组可覆盖五个时隙。有效载荷多达 226 个信息字节（其中两个字节是有效载荷头）加 16 位 CRC 码。DM5 分组的有效载荷头仅两字节长，在有效载荷头里的长度指示器指出了用户字节量（有效载荷头和 CRC 码除外）。当 DM5 分组进行发送或接收时，在五时隙持续期间 RF（射频）跳频不发生改变（第一个时隙是信道识别码传输时隙）。

f. DH5 分组

该分组类似于 DM5 分组。除分组里有效载荷外，其余信息都没用 FEC 编码。为此，DH5 分组可多达 341 个信息字节（包括两字节信息头）加 16 位 CRC 码。

DH5 分组可覆盖五个时隙。当 DH5 分组被发送或接收时，在五时隙持续期间，跳频不发生改变（第一个时隙是信道识别码传输时隙）。

g. AUX1 分组

该分组类似于 DM5 分组，但没有 CRC 码。AUX1 分组可含有多达 30 个信息字节（包括一字节有效载荷头）。AUX1 分组可覆盖单时隙。

2.4.5 有效载荷格式

在有效载荷里，有两个数据段应作区分：（同步）语音段和（异步）数据段。ACL 分组只有数据段，SCO 分组只有语音段，而 DV 分组则兼有两种数据段。

1. 语音段

语音段是一个定长数据段。对于 HV 分组，语音段长度是 240 位；对于 DV 分组，语音段长度是 80 位。不需带有效载荷头。

2. 数据段

数据段由三个部分组成：有效载荷头、有效载荷主体和 CRC 码（仅 AUX1 分组不具有 CRC 码）。

(1) 有效载荷头

只有数据段具有有效载荷头。有效载荷头长度为一个或两个字节。若第一个和第二个段的分组只有一字节有效载荷头，在第三个或第四个段内的分组有两字节有效载荷头。有效载荷头用于指示逻辑信道（两位 L_CH 表示），控制逻辑信道中的数据流（一位 FLOW 表示），并含有有效载荷长度指示器（分别用五位或九位表示一字节或两字节有效载荷头）。在两字节有效载荷头的情况下，长度指示器有四位扩展到下一字节，第二字节的其余四位则留作备用，并设置为“0”。一字节或两字节有效载荷头的格式如图 2.11 所示。图中，L_CH 段最先传输，最后是长度段。表 2.7 描述了有关 L_CH 段的详细内容。



图 2.11 有效载荷头格式

表 2.7 逻辑信道的 L_CH 段内容

L_CH 码 (b ₁ b ₀)	逻辑信息	信息
00	NA	未定义
01	UA/UI	L2CAP 消息的后续分段
10	UA/UI	L2CAP 消息的开始或非分段
11	LM	LMP 消息

一个 L2CAP 消息可以分成几个分组。代码 10 用于携带该消息第一分段的 L2CAP 分组，代码 01 用于后续分段。若没有分段，对每个分组来说都采用代码 10。代码 11 表示了 LMP 消息。代码 00 留作备用。

有效载荷中的流控制指示器（FLOW）用来在 L2CAP 层次上控制流量。在实际应用中，它用来控制每个逻辑信道中的流量。FLOW=1 表示流控制开启（“发送正常”），FLOW=0 表示流控制关闭（“发送停止”）。有效载荷头中流控制位没有严格的实时要求。由最后正确接收有效载荷头的流控制位确定流量状态。链路管理器负责有效载荷头流控制位的设置及处理。由链路控制器通过分组头里的流控制位在分组层次上执行实时流量控制。通过使用有效载荷流位，可以控制来自远程终端的通信。允许生成和发送有效载荷长度为 0 的 ACL 分组。当有效载荷长度为 0 时（即：在 L2CAP 分组发送过程的中间，不得发送空起始段），L2CAP 起始和后续分段指示（L_CH=10 及 L_CH=01）也将保留它们的信息。有效载荷长度等于 0 和 L_CH=0 的 ACL 分组的发送总能确保安全。有效载荷流控制位对于每个逻辑信道（UA/UI 或 LM）都有其含义，如表 2.8 所示。

表 2.8 逻辑信道上有效载荷头流位的使用

L_CH 码 (b ₁ b ₀)	ACL 有效载荷头流位的用法及语义
00	无定义，备用
01 或 10	UA/UI 信道（用做发送 L2CAP 信息）流控制
11	通常传输时 FLOW 总被置“1”，且在接收时忽略该位

另外，在 LM 信道上，不使用流控制，且有效载荷流控制位总是置为“1”。

除有效载荷头和 CRC 码以外的有效载荷（即只包括有效载荷主体），长度指示器指出有效载荷里的字节数（即 8 位字）。在 1 字节长度的数据头中的 MSB 长度段是有效载荷头里的最后（最右）一位。在 2 字节长度的数据头中的 MSB 长度段是有效载荷头里的第二字节自左边数起的第四位。

（2）有效载荷主体

有效载荷主体包括用户主机信息，并用于确定有效用户吞吐量。有效载荷主体的长度由有效载荷头的长度指示段指出。

（3）CRC 代码发生器

有效载荷中的 16 位循环冗余校验码使用 CRC-CCITT 多项式 210041（8 进制表示）产生，与 HEC 生成方法相似。在确定 CRC 码之前，用一个 8 位值来初始化 CRC 发生器。对于在主单元呼叫应答状态中发送的 FHS 分组中的 CRC 码，使用从单元 UAP。对于在查询应答状态中发送的 FHS 分组，使用 DCI。而对所有其他分组，使用主单元 UAP。

8 个二进制数置于 LFSR 回路的 8 个最低位（最左边）的位置，而其余 8 位设置为 0。其后，CRC 码与信息进行运算。CRC 码附加信息后，UAP（或 DCI）则被忽略。对于接收方，在接收信息校验前，CRC 回路以相同方法使用 8 位 UPA（DCI）初始化。更详细的内容参看错误校验章节内容。

2.5 纠错

在蓝牙技术中使用了三种纠错方案：

- 1/3 比例前向纠错码；
- 2/3 比例前向纠错码；
- 用于数据的 ARQ 方案。

对数据有效载荷使用 FEC 方案的目的是减少重发次数。然而，对于纠错要求不高的情况，FEC 将增加不必要开销，从而导致数据吞吐量下降。因此，分组定义中对于在有效载荷中采用或不采用 FEC 给出了相当的灵活度，由此才定义了 ACL 链接中使用的 DM 和 DH 分组和 SCO 链接中使用的 HV 分组。分组头通常采用 1/3 比例前向纠错码保护，它含有很重要的链接信息，能容忍多位错误。

有关话音解码器中用于屏蔽错误的纠错措施不包含在这部分内容中，该问题将在本书的有关章节中讨论。

2.5.1 前向纠错码

1. 1/3 比例前向纠错码

这是一种较简单的纠错码方式，它用一种简单的 3 倍重复格式，即实现时对每位信息重复三次，如图 2.12 所示。

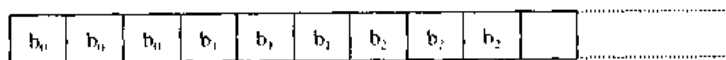


图 2.12 1/3 比例前向纠错码

在整个分组头里都采用了三位重复码，同时在 HVI 分组里的话音段中也采用了这种编

码格式。

2. 2/3 比例前向纠错码

另一种 FEC 方案采用了一种 (15, 10) 精简的汉明码表示方式。生成多项式为: $g(D) = (D+1)(D^4+D+1)$ 。生成这种代码的线性反馈移位寄存器 (LFSR) 如图 2.13 所示。

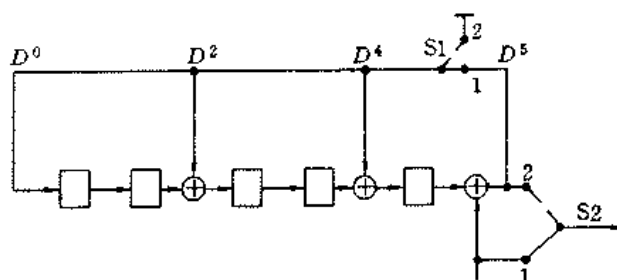


图 2.13 线性反馈移位寄存器

所有的寄存器单元在初始化时被设置为“0”。通过设置在位置 1 上的开关 S1 和 S2，将 10 个信息位顺序载入 LFSR。在最后一位输入完毕后，开关 S1 和 S2 被置于位置 2 上，且移出了五个奇偶位。这些奇偶位附加在信息位后。由此，每个 10 位信息块就被编码成为 15 位代码字。该代码字能够在各代码字中纠正所有奇数位错和检测所有偶数位错。2/3 比例前向纠错码可用于 DM 分组、DV 分组中的数据段、FHS 分组和 HV2 分组中。由于编码器采用长度为 10 的信息段，所以值为“0”的尾位可附加在 CRC 位之后。所有需要编码的位数（即：有效载荷头、用户数据、CRC 和尾部数位）必须是 10 的整倍数。

2.5.2 ARQ（自动重复请求）方案

使用自动重复请求方案，DM、DH 和 DV 分组的数据段可以进行传输或重发，直到收端返回成功接收确认信息（或超时）为止。该确认信息包含在返回分组头里，故称为稍带确认（Piggy-backing）。为了确定有效载荷正确与否，循环冗余校验码应加载于有效载荷中。ARQ 方案只工作在分组的有效载荷上（仅针对具有 CRC 的有效载荷）。分组头和语音有效载荷不受 ARQ 保护。

1. 无编号 ARQ

蓝牙使用快速、无编号确认方案。为了应答前次接收分组，应返回 ACK (ARQN=1) 或 NAK (ARQN=0)。从单元将在主—从时隙后紧跟在从—主时隙中进行应答。主单元则将在下一个事件中应答，该事件将给出同一从单元地址（即：在有关从单元最后接收的分组和主单元对该分组的应答之间，主单元可能已经给出了其他从单元的地址）。为了成功完成分组重发，至少应对 HEC 进行校验。另外，必要时 CRC 也必须校验。

在一次可能是呼叫、呼叫扫描、主—从切换或解除休眠结果的新连接开始时，主单元将发送 POLL（轮询）分组来检查连接。在该分组里，主单元初始化 ARQN 位为无效确认（NAK）。而由从单元发送的应答分组，也将 ARQN 位设置成 NAK。随后的分组将遵循下列规则。

ARQ 位只能由包含 CRC 和空时隙的分组改变。在 CRC 分组接收成功时，ARQN 位置为正确确认（ACK）。如果，在从单元的任何接收时隙里或在主单元分组发送后的接收时隙

中，没有检测到识别码，而且 CRC 分组的 CRC 校验或 HEC 校验失败，此时 ARQN 位将被置成 NAK。

具有正确 HEC 但却被指定为其他从单元的分组，或者除了 DH、DM 或 DV 以外的分组，不影响 ARQN 位。在这些情况下，当 ARQN 位在分组接收之前时，该位将被忽略。如果具有正确头的 CRC 分组同先前接收的分组具有相同 SEQN，则 ARQN 位设置为 ACK，而且将在不进行 CRC 校验的情况下忽略该有效载荷。

在 FHS 分组里的 ARQ 位没有意义。FHS 分组里的 ARQN 位的内容不作校验。

广播分组使用 CRC 校验错误，但不使用 ARQ 方案。广播分组不需要确认。

非激活链接模式 HOLD（保持）和 SNIFF（呼吸）不影响 ARQN 方案。从这些模式返回后，分组继续使用设置成保持/呼吸模式开始前使用的值。

2. 重传过滤

在收到主动确认前（或超时超出前），有效载荷数据一直都处于重发状态。重发是一个执行过程，它用于分组自身传输失败或因在返回分组中稍带确认传输失败（在后一种情况中，故障率较低的原因是在头中有相当多的附加编码）。后一种情况中，收端反复持续接收相同有效载荷。在收端为了过滤重传数据，头将附加 SEQN 位。通常，每次新的 CRC 数据有效载荷传输，SEQN 位将交替发生变化。在重传过程中，SEQN 位不发生改变。所以，收端可以将当前 SEQN 位与以前的 SEQN 位进行比较。如果比较结果不同，表明新的信息已到达；当比较结果相同时，表明是相同有效载荷，且该有效载荷可以放弃。只有新的数据有效载荷才能转发给链路管理器。值得注意的是：CRC 数据有效载荷只能由 DM、DH 或 DV 分组携带。

作为呼叫、呼叫扫描、主-从切换或解除休眠的结果的新连接开始时，主单元发送 POLL 分组检查连接。从单元则以分组形式表示回答。第一个 CRC 分组的 SEQN 位在主从单元两端都被置为“1”，随后的分组遵循如下规则：

每当一次新的 CRC 分组发送时，SEQN 位都反相一次。在收到 ACK 或分组刷新之前，CRC 分组重发时都使用相同的 SEQN 值。当收到 ACK 时，SEQN 位产生反相，并将发送一个新的有效载荷。当分组刷新时，将立即发送一个新的有效载荷，而 SEQN 位不必改变。然而如果在新分组发送前收到 ACK，则 SEQN 位反相。由于重传过滤，该过程将防止消息（在刷新命令已给出后）的第一个分组丢失。

FHS 分组里的 SEQN 位没有实际意义，该位可以设置为任何值，内容不必校验。在另外所有其他类型的分组传输期间，SEQN 位仍保持与先前分组中的相同值不变。

非激活模式 HOLD/SNIFF（保持/呼吸）不影响 SEQN 方案。在从这些状态返回后，分组继续使用在进入保持/呼吸模式前使用的值。

SEQN 位只受 CRC 数据分组影响，而 CRC 分组传输过滤如图 2.14 所示。

3. 有效载荷刷新

由于插入重发以确保数据无错转发，ARQ 方案将在通信流中引起可变延迟。对于某些通信链路，只能允许有限延迟。所以重传允许设置某个超时限制，用以忽略当前有效载荷，而考虑下一有效载荷。该数据转发称为等时通信（isochronous traffic）。这意味着要继续发送和接收下一个数据有效载荷，重传过程必须中止。放弃重传过程通过刷新旧数据和强制蓝

牙控制器转而读取下一个数据来实现。

刷新将导致逻辑链路控制和适配协议（L2CAP）消息的其余部分丢失。因此，刷新后的分组将在分组头中包含一个 L_CH（逻辑信道）=10 的起始分组指示，以指示下一个 L2CAP 消息。它将刷新通知接收方。同时，刷新不会改变 SEQN 位的取值。

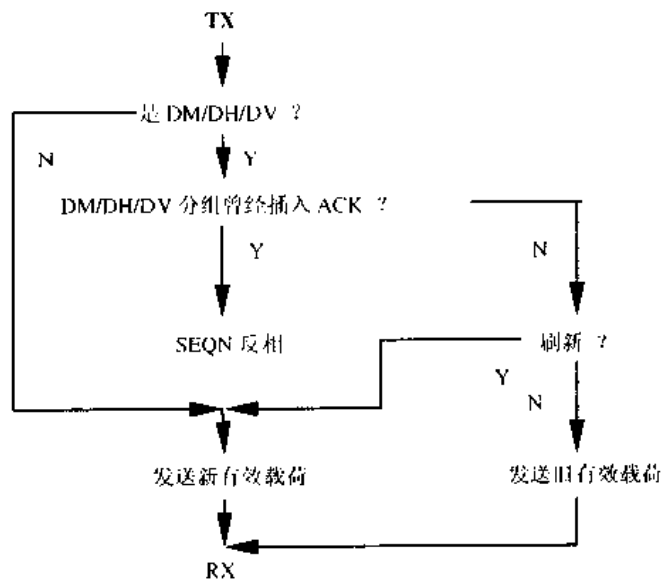


图 2.14 CRC 分组重发过滤

4. 多从单元和广播分组

在具有多个从单元的匹克网里，主单元独立于各从单元执行 ARQ 协议。

广播分组是由主单元同时发往所有从单元的分组。广播分组通过全为“0”的成员地址（AM_ADDR）标识，广播分组不需确认（至少不在 LC 层上）。但需注意，FHS 分组是惟一一类可以拥有全“0”地址编码而不是广播分组的分组。

由于广播分组不需确认，所以每次广播分组都可以按照一固定重复发送次数重复发送。在同一消息的下一广播分组重复发送之前，一个广播分组可以重复发送 N_{BC} 次。重复广播方案如图 2.15 所示。

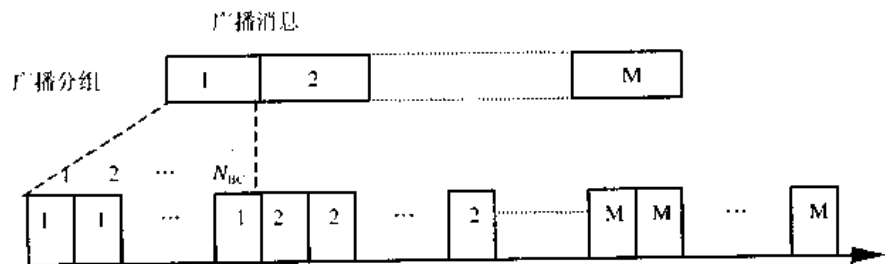


图 2.15 广播重复方案

使用 CRC 的广播分组具有它自身的序列号。第一个使用 CRC 的广播分组的 SEQN 由主单元设置为 SEQN=1，此后每次发送 CRC 的新分组，都将使 SEQN 反相。不含 CRC 的广播分组不影响该序列号。从单元接受它在连接中收到的第一个广播分组的 SEQN，并将在检查后续广播分组中的 SEQN 变化。由于广播消息不需要确认，而且无结束标志指示，所以正确接收起始分组显得尤其重要。为了确保起始分组正确接收，将不过滤作为 L2CAP 起

始分组和链路管理协议（LMP）分组广播分组的重发分组。这些分组由有效载荷头里的 L.CH=1X 指出。只有 L2CAP 后续分组的重发分组需被过滤。

2.5.3 错误校验

使用头 HEC 和有效载荷中的 CRC，可以校验分组差错或传输差错。在接收分组时，首先校验识别码。由于在信道识别码中的 64 位同步字源自于 24 位主单元的低地址部分（LAP）。这样就可以校验 LAP 是否正确，并可防止接收方接收来自其他匹克网的分组。

HEC 和 CRC 用来检测数据错及地址错。为了以八位增加地址空间，高地址部分（UAP）通常也被包含在 HEC 和 CRC 检测中。甚至当具有相同识别码的分组——即：具有相同 LAP 但不同 UAP 的设备识别码——通过识别码测试时，它也将在 UAP 位不匹配时在 HEC 和 CRC 测试后被放弃。但是有一种情况例外，就是在收发两端都没有可用的公共 UAP。这种情况发生在查询应答状态中的 FHS 分组生成 HEC 和 CRC 时。此时，将使用缺省检测初始化（DCI）值，该 DCI 定义为 0X00（十六进制）。

在计算 HEC 和 CRC 之前，HEC / CRC 发生器中的移位寄存器用 8 位 UAP（或 DCI）值初始化。头和有效载荷信息分别（LSB 在先）移入 HEC 和 CRC 发生器。HEC 和 CRC 的产生及检测如图 2.16、图 2.17 所示。

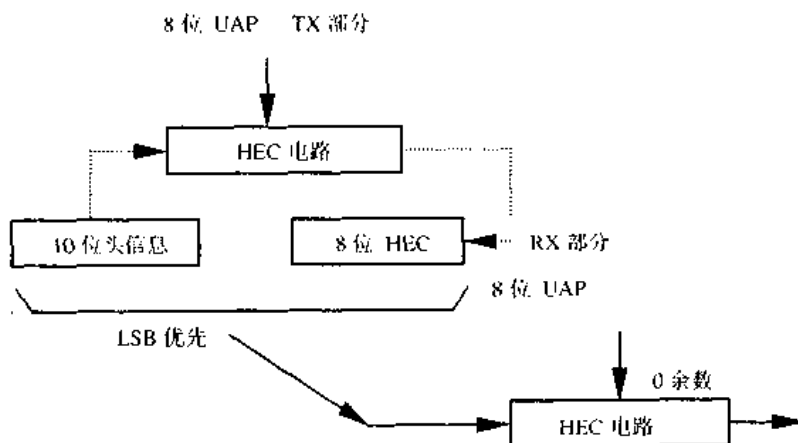


图 2.16 HEC 发生器和检测

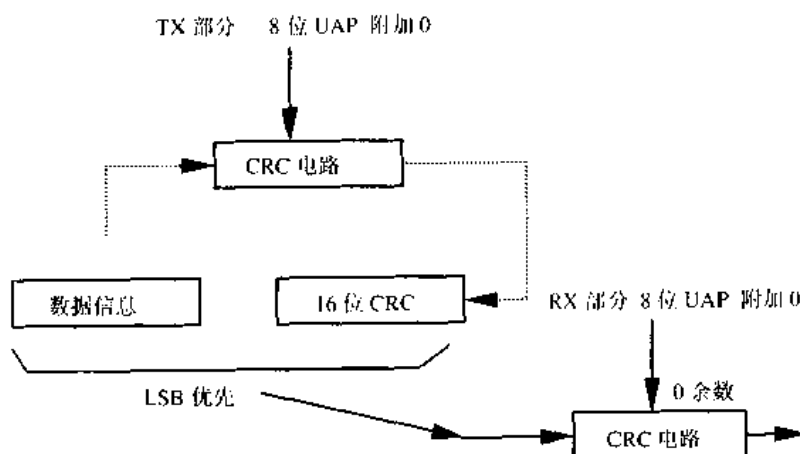


图 2.17 CRC 发生器和检测

HEC 产生 LFSR 过程如图 2.18 所示。

在以上过程中将产生一个多项式：

$$g(D) = (D+1)(D^7+D^4+D^3+D^2+1) = D^8+D^7+D^5+D^2+D+1。$$

首先，该电路预先载入。在该过程中，使用 UAP (UAP₀) 的 LSB 进入最左边单元的 8 位 UAP，以及 UAP₇ 进入最右边的单元。然后，数据将使用设置在位置 1 上的开关 S 移入。当最后一个数据位已进入 LFSR 时，开关 S 设置在位置 2，且 HEC 可以从寄存器中读出。LFSR 位从右到左被读出（即：位置 7 的位首先被传输，紧随其后是在位置 6 上位）。

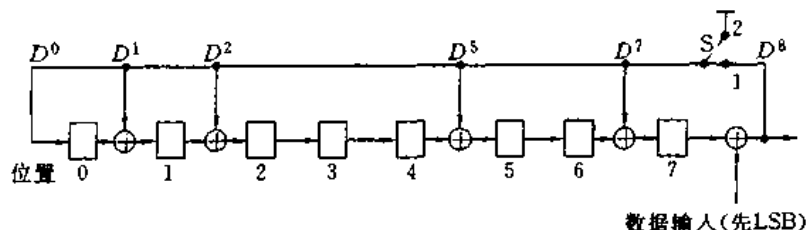


图 2.18 HEC 产生 LFSR 电路

用于 CRC 的 16 位 LFSR 同样使用 CRC-CCITT 生成多项式： $g(D) = D^{16}+D^{12}+D^5+1$ ，如图 2.19 所示

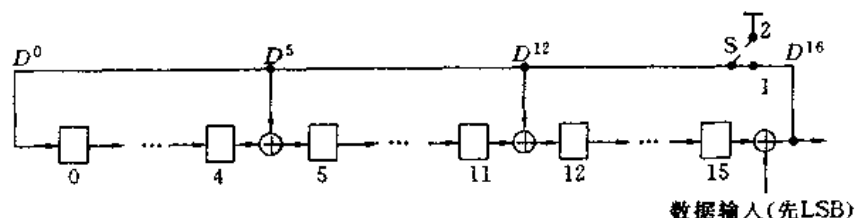


图 2.19 生成 CRC 的 LFSR 电路

对于这种情况，最左边 8 位将首先使用 8 位 UAP 载入 (UAP₀ 在左，UAP₇ 在右)，而最右边的 8 位设置为 0。

当数据移入时，开关 S 设置在位置 1。在最后一位进入 LFSR 后，开关 S 设置在位置 2，且寄存器内容从右至左传输（即：开始用位置 15，然后是位置 14.....）。

2.6 逻辑信道

在蓝牙系统中定义了五种逻辑信道：链路控制器 (LC) 控制信道、链路管理器 (LM) 控制信道、UA 用户信道、UI 用户信道、US 用户信道。

LC 和 LM 控制信道分别用于链路控制层次和链路管理层次。UA、UI 和 US 用户信道分别用于传输异步、等时和同步用户信息。LC 信道在分组头携带，而其他信道则在分组有效载荷中携带。LM、UA 和 UI 信道在有效载荷头里的 L_CH 段给出指示。US 信道只能由 SCO 链接传输，UA 和 UI 信道一般由 ACL 链接传输。然而，它们也可在 SCO 链接上以 DV 分组的数据传输，LM 信道既可以用 SCO 链接传输也可用 ACL 链接传输。

1. LC 信道（链路控制）

该信道携带类似于 ARQ、流控制和有效载荷特征的低层链路控制信息。除了没有分组头的 ID 分组以外，LC 信道可传输各种分组。

2. LM 信道（链路管理）

LM 控制信道用来传送主单元和从单元链路管理器之间的互换控制信息。LM 信道使用保护 DM 分组。LM 信道由有效载荷头里 L_CH 代码 11 指定。

3. UA/UI 信道（用户异步/等时数据）

UA 信道传送 L2CAP 透明异步用户数据。该数据可以以一个和多个基带分组形式传输。对于分段消息，起始分组使用值为 10 的有效载荷头 L_CH 代码，其余后续分组使用 L_CH 代码 01。如果没有分段，则所有分组都使用 L2CAP 开始代码 10。

等时数据信道由高层正确定时起始分组支持。在基带层次，L_CH 代码用法同 UA 信道一样。

4. US 信道（用户同步数据）

US 信道传输透明同步用户数据。该信道在 SCO 链接上传输。

5. 信道映射

LC 信道被映射于分组头，其他的信道则映射于有效载荷。US 信道只能映射到 SCO 分组，而所有其他的信道映射到 ACL 分组或 SCO DV 分组。如果涉及到较高优先权信息，LM、UA 和 UI 信道可以中断 US 信道。

2.7 数据加噪

在传输之前，头和有效载荷使用数据噪声字加扰，其目的是使来自较高冗余模式的数据随机化，并最小化分组中的 DC 偏差。这种加扰过程先于 FEC 编码完成。在接收端，接收数据使用与发送端相同噪声字发生器进行解扰。解扰过程在 FEC 解码后完成。

噪声字利用多项式 $g(D) = D^7 + D^4 + 1$ （即八进制数 221）生成，并随即使使用头和有效载荷的异或操作。噪声字使用线性反馈移位寄存器生成，如图 2.20 所示。

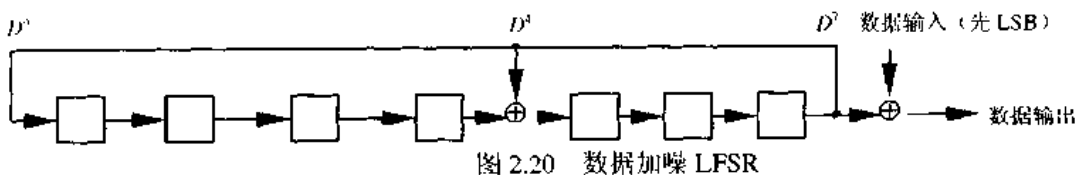


图 2.20 数据加噪 LFSR

在每次传输前，移位寄存器使用扩展 MSB 位值为 1 的主蓝牙时钟 $CLK_{6.1}$ 的一部分进行初始化。该初始化使用写入位置 0 的 CLK_1 ，使用写入位置 1 的 CLK_2 等方式执行。在跳频访问窗口间 FHS 分组发送是一种例外，加噪寄存器的初始化执行方式不同。作为主单元时钟代替，将使用在查询或呼叫应答（取决于当前状态）规则中使用的使用 X—输入，具体内容分别见后续有关 79 跳和 23 跳系统的表格所示。在 79 跳系统情况下，5 位值用两个值为 1 的 MSB 扩展；在 23 跳系统情况下，4 位值用 3 位扩展，两个 MSB 置成 1，而第三个最重要的位置为“0”。在寄存器初始化期间，X（即： X_0 ）的 LSB 位写入位置 0， X_1 位置写入位置 1 等。

初始化后，分组头和有效载荷（包含 CRC）进行加扰。有效载荷加噪取自在 FEC 结束时的加噪 LFSR 的状态。在分组和有效载荷之间不存在移位寄存器上的再次初始化。数据输

入序列的第一位是分组头的最低位 (LSB)。

2.8 收/发规则

为了支持 ACL 及 SCO 链路通信, 本节描述前面所述的分组使用方法。另外对使用 TX 及 RX 缓冲区的规则也作了说明。下面内容中对 TX 和 RX 常规使用方法的介绍仅是一种描述性特征, 而最终执行结果可以不同。

2.8.1 TX 规则

TX 规则可分别在 ACL 链路和 SCO 链路上执行。用于 TX 规则的 ACL 缓冲区和 SCO 缓冲区如图 2.21 所示。

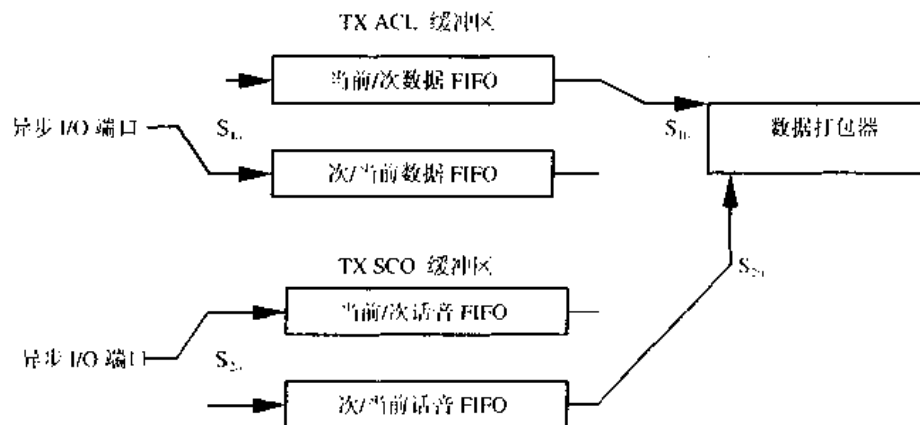


图 2.21 TX 缓冲区的功能图

在图中, 只对单 TX ACL 缓冲区和单 TX SCO 缓冲区进行了描述。在主单元中, 对于每一从单元都分别有一个 TX ACL 缓冲区。另外, 对每个 SCO 从单元 (不同的 SCO 链路可以重用相同的 TX SCO 缓冲区或各自带有的独立 TX SCO 缓冲区) 可以有一个或多个 TX SCO 缓冲区。每个 TX SCO 缓冲区由两个 FIFO (先进先出) 寄存器组成。其中, 由蓝牙控制器访问和读出的用于组成分组的寄存器称为现态寄存器, 而由蓝牙链路管理器访问和载入新信息的寄存器称为次态寄存器。开关 S_1 和 S_2 的位置确定哪一个为现态寄存器或次态寄存器。开关由蓝牙链路控制器控制。在 FIFO 寄存器里输入和输出开关不得同时和同一寄存器相连。

在 ACL 和 SCO 链路上公用的分组 (ID、NULL、POLL、FHS、DMI) 当中, 只有 DMI 分组带有可在链路控制器和链路管理器之间交换的有效载荷。该公用分组可实现对 ACL 缓冲区的利用。所有 ACL 分组都是利用 ACL 缓冲区, 所有 SCO 分组都是利用 SCO 缓冲区, 而 DV 分组例外。对于 DV 分组, 其语音部分由 SCO 缓冲区处理, 而数据部分由 ACL 缓冲区处理。

1. ACL 通信

在纯 (异步) 数据情况下, 只考虑使用 TX ACL 缓冲区。此时, 只有 DM 和 DH 分组可以使用, 且它们具有不同长度值。分组长度在有效载荷头里指出。选用高速数据或中速数据取决于链路质量。当链路质量较好时, 可以忽略数据有效载荷里的 FEC, 可使用 DH 分组, 否则只能用 DM 分组。

纯数据通信中的默认类型是 NULL。它意味着如果没有数据可传输 (数据通信是异步的,

且在没有有效载荷时出现暂停状态)或没有从单元需轮询时,可发送 NULL 分组——目的是为了发送链路控制信息到其他蓝牙单元(如用于接收数据的 ACK/STOP 信息)。当无链路控制信息可用(无须确认或不必要停止 RX 流)时,则完全不必发送分组。

TX 规则工作方法如下所述:

蓝牙链路管理器将新的信息载入到开关 S_{1a} 指定的寄存器。紧接着,它将发送刷新命令到蓝牙链路控制器。该链路控制器强制开关 S_1 发生变化(S_{1a} 和 S_{1b} 开关同时执行)。当需发送有效载荷时,分组打包器读取当前寄存器,并根据分组类型创建一个附加在信道识别码和头后的有效载荷,然后进行传输。在应答分组里(如果涉及到主单元传输,则将在后面的 RX 时隙到达,如果涉及到从单元传输,它可能被推迟到以后的 RX 时隙),传输结果将返回。在 ACX 情况中,开关 S_1 应改变位置;如果(显式或隐式)接收到 NAK,开关 S_1 就不改变位置。在这种情况下,相同的有效载荷将在下一 TX 时间完成重传。

只要链路管理器持续使用新信息载入寄存器,蓝牙链路控制器将自动传输有效载荷。另外,在错误情况下将自动执行重传过程。当没有新信息载入时,链路控制器将发送 NULL 或什么都不发。如果没有新信息载入次态寄存器,则在最后一次传输期间,在上次传输被确认后且次态寄存器变成现态寄存器后,分组打包器将指向空寄存器。如果将新数据载入次态寄存器,将要求刷新命令转换 S_1 开关到合适的寄存器。在各 TX 时隙之前,只要链路管理器持续载入数据和类型寄存器,由于 S_1 开关由应答时接收的 ACK 信息进行控制,数据将由链路控制器自动处理。然而,如果来自于链路管理器的通信曾经被中断过,而且将发送默认分组,则需要刷新命令继续链路控制器中的通信流。

刷新命令也可以用于时限(等时)数据。在链路质量较差的情况下,有必要进行多次重传。在某些应用中,数据有时间限制。如果因链路出错导致有效载荷始终重发,则其有效载荷可能过时,且此时系统可能决定使用最近的数据来代替原数据,同时跳过不能正常传输的有效载荷。这一过程也可由刷新命令完成。使用刷新命令,开关 S_1 强制改变且链路控制器也强制考虑下一数据有效载荷,并取消 ACK 控制。

2. SCO 通信

在 SCO 链接方式下,我们只使用 HV 分组类型。同步端口连续在 SCO 缓冲区中载入次态寄存器。 S_2 开关则根据 T_{SCO} 间隔变化。该 T_{SCO} 间隔在 SCO 链接建立时的主单元和从单元之间进行协商。

对于每一新的 SCO 时隙,在 S_2 开关发生改变后,分组打包器将从现态寄存器读出数据。如果 SCO 时隙用于在主单元和相关从单元(或其他从单元)之间,发送具有较高优先权的有关一个控制分组的控制信息时,分组打包器将放弃 SCO 信息并使用控制信息。该控制信息必须在 DM1 分组里发送。通过使用 DV 或 DM1 分组,主单元和 SCO 从单元之间可交换数据或链路控制信息。任何 ACL 类型的分组都能用于发送数据或链路控制信息给其他的 ACL 从单元。

3. 数据—语音混合通信

DV 分组可以在 SCO 链接中同时支持数据和语音。当类型为 DV 时,链路控制器将读取数据寄存器以填充数据段,读取语音寄存器以填充语音段。其后,开关 2 被改变。然而 S_1 的位置取决于类似于 ACL 链接上的传输结果。只有当接收到 ACK 时, S_1 开关才改变位置。在各个 DV 分组里,语音信息总是新的。但数据信息因以前传输失败,可能会重发。如果没

有数据发送，在数据一语音混合传输之前，SCO 链接将自动地从 DV 分组变成当前 HV 分组类型使用。注意当数据流被中断且新数据到达时，必须使用刷新命令。

如果信道容量允许，混合数据一语音传输（除 SCO 链接外）也能单独使用 ACL 链接来完成。

4. 默认分组类型

在 ACL 链接上，主单元和从单元中的默认类型都是 NULL。它意味着没有需要发送的用户信息，或者如果具有 ACK 或 STOP 信息则将发送 NULL 分组，或者完全没有分组发送。NULL 分组可由主单元使用，以将下一从一主时隙分配到某一从单元（一个明确编址的从单元）。但是，从单元可以不必强制应答来自于主单元的 NULL 分组。如果主单元要求应答，它可发送一个 POLL 分组。

当 SCO 链接建立时，SCO 分组类型在 LM 层上进行协商。协商认可的分组类型也就是 SCO 时隙的缺省分组类型。

2.8.2 RX 规则

RX 规则可分别为 ACL 链接和 SCO 链接执行。但是，与主单元 TX ACL 缓冲区相比，单个 RX 缓冲区将在所有从单元中共享。对于 SCO 缓冲区，它取决于如何区分不同 SCO 链接方式，而不论是否需要额外的 SCO 缓冲区。图 2.22 描述了 RX 规则中 ACL 和 SCO 缓冲区的使用方式。

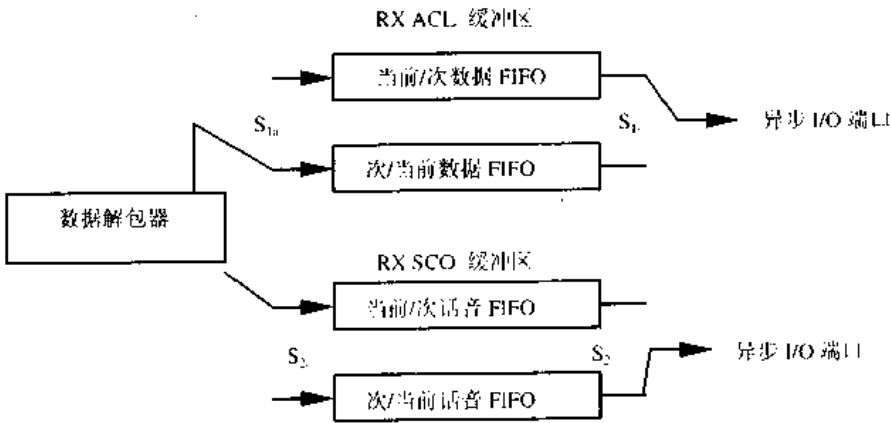


图 2.22 RX 缓冲区功能图

RX ACL 缓冲区由两个 FIFO 寄存器组成。一个寄存器可由含最近一个 RX 分组的蓝牙链路控制器识别和载入，另一个寄存器可由蓝牙链路管理器访问并用于读出先前的有效载荷。RX SCO 缓冲区也可由两个 FIFO 寄存器组成，一个寄存器由新到的语音信息填入，而另一个寄存器则可使用语音处理单元进行读取。

由于接收的分组头含有 TYPE 指示器，它指出有效载荷是否含有数据和/或语音信息。分组解包器能够直接自动地与合适缓冲区通信。链路管理器每读一次旧的寄存器，开关 S₁ 就改变一次。如果在 RX 寄存器为空前，下一个有效载荷到达，STOP 指示必须包括在返回的下一个 TX 分组的分组头里。一旦 RX 寄存器空闲，STOP 指示将再次移走。SEQN 段在新的 ACL 有效载荷被载入 ACL 寄存器前进行校验(L_CH 中的刷新指示和广播消息影响 SEQN 段的译码)。

在每个 T_{SCO} 上都将改变 S₂ 开关。如果由于头错误，没有新的语音信息到达，开关仍

将发生改变。为了解释丢失的语音部分，语音处理单元应处理该语音信号。

2.8.3 流控制

流控制用来解决新的有效载荷到达时 RX ACL 缓冲区的填满问题。如前所述，在返回 TX 分组里的头段 FLOW 用 STOP 或 GO 来控制新数据传输。

1. 收端控制

只要数据不能接收，将发送 STOP 指示，该指示由链路控制器自动将它插入返回分组的头中。只要 RX ACL 缓冲区不被链路管理器清空，就返回 STOP。当新的数据可以再次接收时，返回 GO 指示。GO 是一个系统默认值。

注意：不包含数据的所有分组类型仍可以被接收。例如话音通信就不受流控制影响。同时也要注意，虽然蓝牙单元不能接收新信息，但它仍能传输信息。流控制可以在每一方向上分别应用。

2. 发端控制

在接收 STOP 信令时，链路控制器将自动切换到默认分组类型，并冻结当前 TX ACL 缓冲区状态。只要一收到 STOP 指示，默认分组就将被发送。当没有收到分组时，就隐式假定为 GO。

注意：默认分组含有接收方向上（可以一直开放）的链路控制信息和语音（HV 分组）。当收到 GO 指示时，链路控制器将继续传输数据，就如在当前 TX ACL 缓冲区里一样。

在多个从单元配置中，只停止到发出 STOP 信号的从单元的信息传输。这就说明以前讨论的主单元实现规则只涉及到对应于暂时不能接收数据的从单元的 TX ACL 缓冲区。

2.8.4 比特流处理

在使用无线接口发送信息之前，为增加发送信息的可靠性和安全性，应在发送方进行适当的处理。对于分组头，要增加 HEC。头位使用噪声字加扰，并采用 FEC 编码。在收端，执行相反过程。

图 2.23 描述收、发两端分组头处理的执行过程，在处理过程中两端的所有头位处理过程是强制性的。

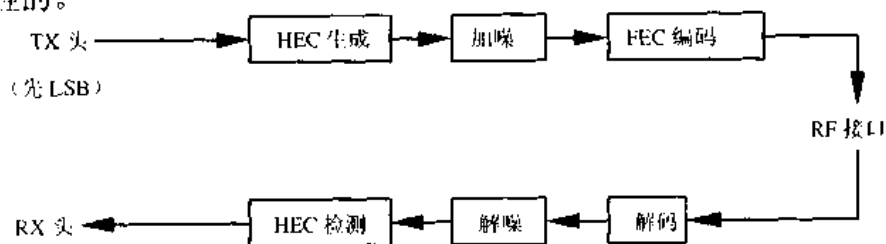


图 2.23 头位处理过程

对有效载荷来讲，将执行类似的处理过程。但执行过程还要取决于分组类别。其执行过程如图 2.24 所示。

另外，除了为分组头定义的处理过程之外，还将对有效载荷进行加密。只有加噪和解噪为每个有效载荷必须强制执行，其他所有处理都是可选的，这取决于分组类型和可用模式。图中凡属可选处理皆用虚线描出。

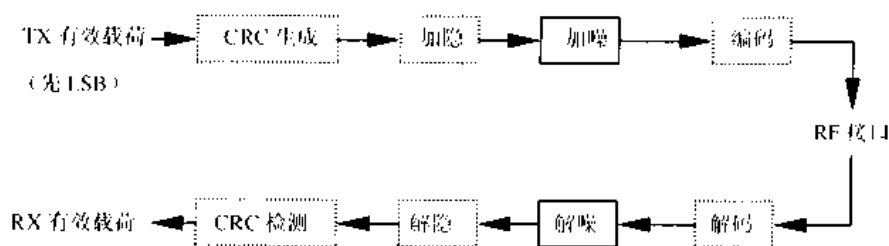


图 2.24 有效载荷位处理过程

2.9 发 / 收定时

蓝牙发射机采用分时双工（TDD）方案。这表明在蓝牙技术中使用同步交替收发方式。采用该定时方案是由于蓝牙单元的 TDD 方案的精确定时模式。

在常规链接模式中，主单元传输总是用偶数时隙（主 CLK1=0）作为起始，而从单元传输总是采用奇数时隙（主 CLK1=1）来作为起始。由于分组类型覆盖多个单时隙，主单元传输将可以在奇数时隙里持续进行，而从单元传输将可以在偶数时隙中持续进行。

主单元分组传输的平均定时的偏差不得快于相对于 625μs 的理想时隙定时的 20ppm。瞬间定时偏差不能大于来自于平均定时的 1μs。于是时隙边界 k 的绝对分组传输定时 t_k 必须满足等式：

$$t_k = \sum_{i=1}^k [(i + d_i)T_N] + t_{\text{offset}}j_k + t_{\text{offset}} \quad (2-1)$$

式中， T_N 是一个标称时隙长度（625μs）， j_k 是在时隙 K 里的抖动（ $|j_k| \leq 1\mu\text{s}$ ），而 d_k 是在时隙 K 的偏差（ $|d_k| \leq 20\text{ppm}$ ）。在给定的各时隙限制里，抖动偏差是非常随机的。而 t_{offset} 是一种随机但固定的常量，对于活动、休眠和呼吸模式，偏差和抖动参数将在后面的链路管理协议内容中具体说明。

2.9.1 主 / 从定时同步

匹克网由主单元的系统时钟同步。在匹克网的存在期间，主单元决不会调整它的系统时钟。在连续两次传输期间，它维持 $M \times 625 \mu\text{s}$ 的准确间隔（此处 M 是一个大于 0 的正整数同时也是一个偶数）。为了匹配主时钟，从单元使用定时补偿来调整它们自身的时钟值。这种定时补偿在每完成一次从主单元接收分组后就应修改。通过用接收分组的精确 RX 定时值与估计 RX 定时值相比较，从单元能够正确地补偿任何定时失调。注意：由于从单元只需要信道识别码进行同步，所以从单元 RX 定时可以用在任何主-从时隙中发送的分组来校准。

从单元 TX 定时基于最近从单元的 RX 定时。RX 定时又基于在主-从时隙期间最后一次成功触发。对于 ACL 链接来讲，该触发必须在主-从时隙里发生，而且要先于当前从单元传输。对 SCO 链接来讲，即使在前一主从时隙中没有收到任何分组，在从单元允许发送一个 SCO 分组前，触发可能已在多个主从时隙中发生。只要定时不匹配值保持在不稳定窗口 $\pm 10 \mu\text{s}$ 内，从单元就可以接收分组和调整 RX 定时。

主单元 TX 定时严格地依照主时钟确定。主单元维持 $M \times 1250 \mu\text{s}$ 准确间隔（此处 M 是一个大于 0 的正整数）。在连续传输开始之间，RX 定时基于准确 $N \times 625 \mu\text{s}$ 移位（此处 N 为

奇数且是一个大于 0 的正整数) 的 TX 定时。

在主单元 RX 周期中, 主单元也可使用 $\pm 10 \mu\text{s}$ 误差窗口允许从单元的失调。主单元将根据相应分组的 RX 处理过程来调整。但对于随后的 TX 和 RX 周期, 主单元将不再调整它的 RX/TX 定时。定时行为可根据单元当前状态稍有差异。

2.9.2 连接状态

在蓝牙连接模式中, 收、发信机的发送和接收过程是交替进行的。其运行模式如图 2.25 和图 2.26 所示。

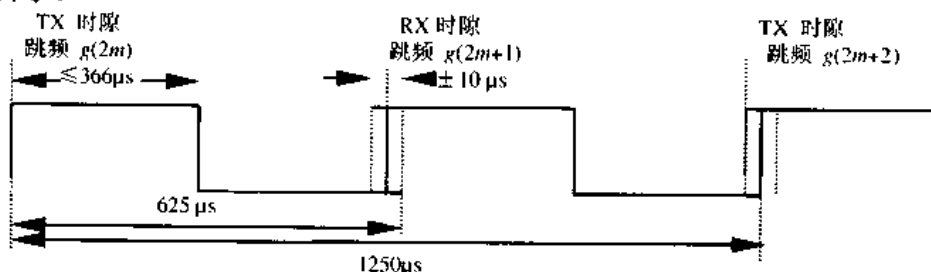


图 2.25 在单时隙有效载荷标准模式里蓝牙主收发信机的 RX/TX 周期

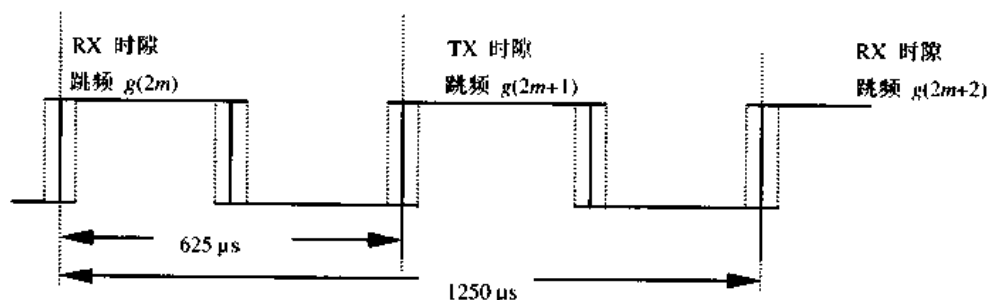


图 2.26 在单时隙有效载荷标准模式里蓝牙从收发信机的 RX/TX 周期

在这两个图中都是用单时隙分组来举例说明。在收、发过程中, 每次 RX 和 TX 传输都以不同的跳频点来实现, 并且传输过程取决于分组类型和有效载荷长度。分组大小可达 $366 \mu\text{s}$ 。对多时隙分组来说, 同一分组可能覆盖 n 个时隙, 因此第一个时隙使用的跳频将在整个传输过程中使用。

$g(m)$ 指出了信道跳频点。传输后, 在 TX 上升沿的起始点处, 返回分组期望为 $N \times 625 \mu\text{s}$, 此处的 N 是一个奇数且为正整数。 N 取决于传输分组的类型。为了解释有些时间误差, 不稳定窗口与准确接收定时之间偏差很小。在正常操作中, 时隙不稳定窗口宽度为 $20 \mu\text{s}$ 。这就是说允许 RX 上升沿可提前早到 $10 \mu\text{s}$ 或延后推迟晚到 $10 \mu\text{s}$ 。在 RX 开始的周期中, 识别相关器搜索在不稳定窗口中的正确信道识别码。如果没有触发事件产生, 接收方进入休眠状态直到下一次 RX 事件发生。如果在搜索中, 相互输出关系明显决不会超出最终阈值, 接收方可以提前进入休眠转态。如触发事件一旦产生, 接收方将保持开放并准备接收剩余信息。

当前主单元传输基于先前的主单元传输。在前主单元 TX 上升沿开始处, 预定为 $M \times 1250 \mu\text{s}$, 此处的 M 取决于传输和接收分组类型。注意, 主单元 TX 定时不会受从单元的时间偏移影响。如果在若干连续时隙期间没有传输过程发生, 主单元将取最后一次 TX 上升沿的 TX 定时作为基准。

从单元传输在从单元的 RX 上升沿开始后预定为 $N \times 625 \mu\text{s}$ 。若从单元的 RX 定时发生

漂移，结果将影响从单元 TX 定时。如果在若干连续时隙期间没有接收产生，从单元将取最后一次 RX 上升沿的 RX 定时作为基准。

2.9.3 退出保持模式

在连接状态里，蓝牙单元可以置于保持模式。此时，蓝牙的收、发信机既不发送信息也不接收信息。当返回到正常操作时，蓝牙从单元退出保持模式后，从单元必须在可以发送信息之前侦听主单元。在这种情况下，从单元里的搜索窗口可以从 $\pm 10 \mu\text{s}$ 增加到较大值 $X \mu\text{s}$ 。我们在此用图 2.27 来说明这个问题。

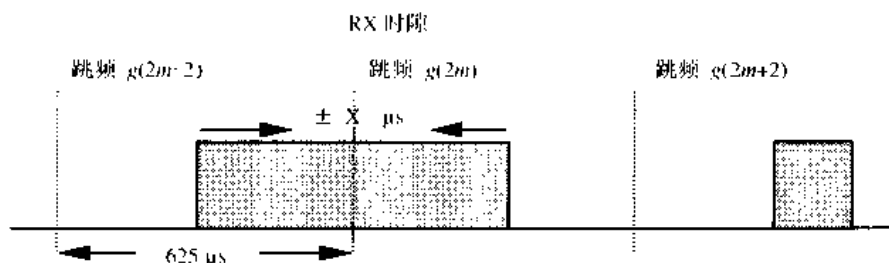


图 2.27 从单元从保持模式中返回 RX 定时

注意：该图仅使用 RX 跳频。用于主从 (RX) 时隙的跳频也扩展到用于从主 (TX) 时隙的前述时间间隔。

如果搜索窗口超过 $625 \mu\text{s}$ ，连续窗口将不在 RX 跳频的开始处： $g(2m)$ ， $g(2m+2)$ …… $g(2m+2i)$ （此处“ i ”是一个整数）； $g(2m)$ ， $g(2m+4)$ …… $g(2m+4i)$ ；甚至 $g(2m)$ ， $g(2m+6)$ …… $g(2m+6i)$ 等的中心点上。为避免搜索窗口相互覆盖，将使用对应于 RX 时隙数的 RX 跳频。

单时隙分组用于从保持模式返回以最小化同步时间。尤其在长期的保持模式中，要求搜索窗口超过 $625 \mu\text{s}$ 。

2.9.4 唤醒休眠状态

休眠模式类似于保持模式。休眠从单元由主单元周期性唤醒以侦听来自主单元的信息单元，并进行同步时钟补偿。如从保持模式返回相似，唤醒的休眠从单元可以将搜索窗口从 $\pm 10 \mu\text{s}$ 增加到较大值 $X \mu\text{s}$ 。其关系表现在从单元从保持模式返回的 RX 定时图中。

2.9.5 呼叫状态

在呼叫状态里，主单元可以很快地以大量不同跳频点，传输对应于要连接的从单元的设备识别码 (ID 分组)。因 ID 分组是一个很短的分组，而跳频速率可以从 1600 跳/秒增加到 3200 跳/秒。在单 TX 时隙间隔里，呼出主单元在两个不同跳频点上进行传输。在单 RX 时隙间隔里，呼出收发信机在两个不同跳频点上进行侦听。

在 TX 时隙期间，呼出单元以 TX 跳频 $f(k)$ 和 $f(k+1)$ 发送一个 ID 分组。在 RX 时隙里，它在相应的 RX 跳频点 $f'(k)$ 和 $f'(k+1)$ 上进行应答。在收到相应呼出分组后，侦听周期精确定时为 $625 \mu\text{s}$ ，其中包括 $\pm 10 \mu\text{s}$ 不稳定期，如图 2.28 所示。

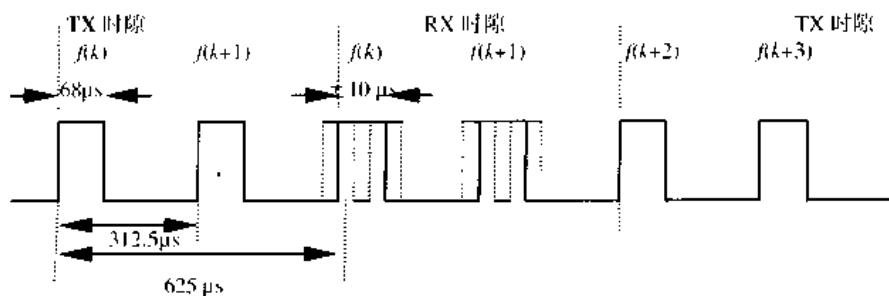


图 2.28 呼叫模式蓝牙收发信机的 RX/TX 周期

2.9.6 FHS 分组

在连接建立和主-从切换期间，FHS 分组由主单元转发到从单元。该分组将建立定时和频率同步。在从单元已收到呼叫消息后，它将返回一个应答消息。该消息是在收到呼叫消息后，由 ID 分组和紧随其后的 625 μs 准确值重新组成。根据主单元的 RX/TX 定时，主单元将在它接收从单元应答的 RX 时隙之后的 TX 时隙中发送 FHS 分组。在应答和 FHS 消息之间的时差取决于从单元接收的呼叫消息的定时。

从单元接收到呼叫消息后将首先在主-从时隙里发送。然后它将在从-主时隙里前部分中以 ID 分组格式应答。FHS 分组的定时基于前次主-从时隙里的最先发送的呼叫消息的定时。在第一次呼叫消息和 FHS 分组之间，有一个准确的 1250 μs 延迟。分组在跳频点 $f(k+1)$ 发送，该跳频点紧随在呼叫消息被接收处的跳频点 $f(k)$ 之后。

从单元接收呼叫消息位于之后的主-从时隙里。这个用 ID 分组的应答过程是在收到呼叫消息后的从-主时隙 625 μs 准确值的后半部分。FHS 分组的定时仍基于前次主-从时隙里首先发送呼叫消息的定时。在第一次呼叫消息和 FHS 分组之间，有一个准确的 1250 μs 延迟，分组在跳频点处 $f(k+2)$ 发送，该跳频点紧随在呼叫消息接收到的 $f(k+1)$ 的跳频点处。如图所示：

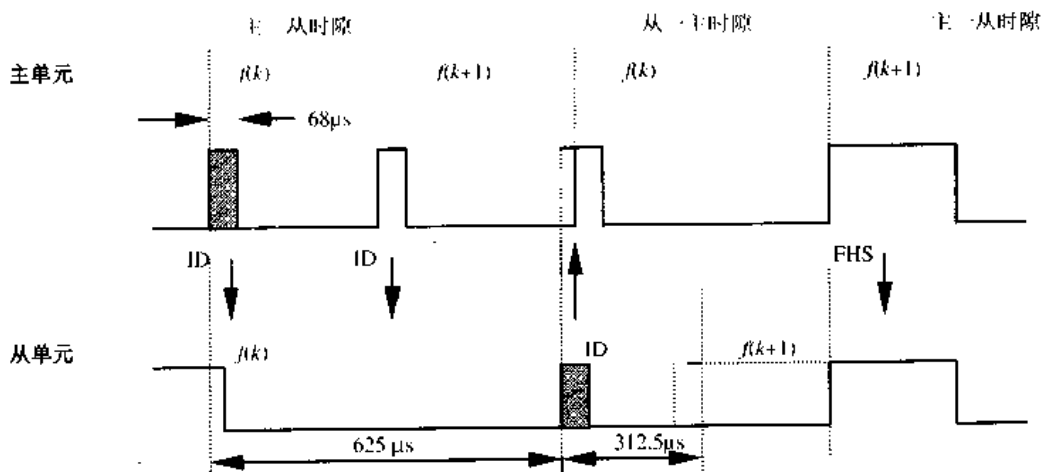


图 2.29 成功呼叫 FHS 分组的定时前半部分时隙

从单元调整它的 RX/TX 定时取决于 FHS 分组的接收，而不是根据呼叫消息的接收。即 FHS 分组接收确认的二次应答消息应在 FHS 分组开始后的 625 μs 进行传输。

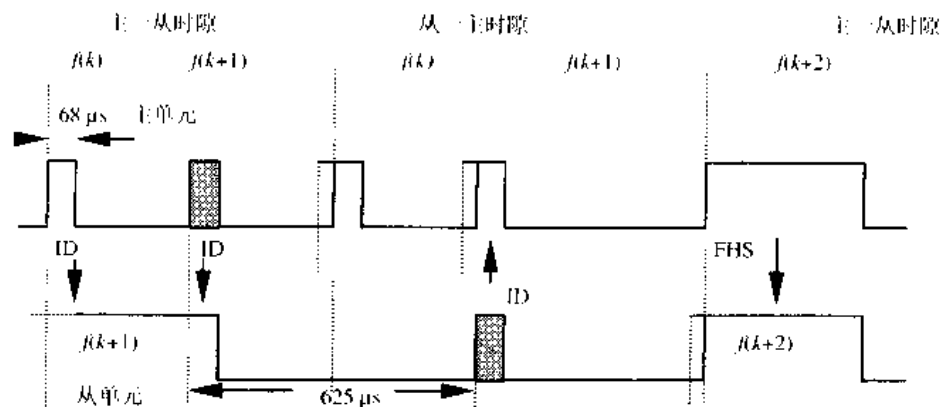


图 2.30 成功呼叫 FHS 分组的定时后半部分时隙

2.9.7 多从单元操作

正如在开始处所提到的，主单元总是以偶数时隙开始传输，而从单元总是以奇数时隙开始传输。这就表明主单元的定时是经过一个时隙（625 μs）发生转换，如图 2.31 所示。

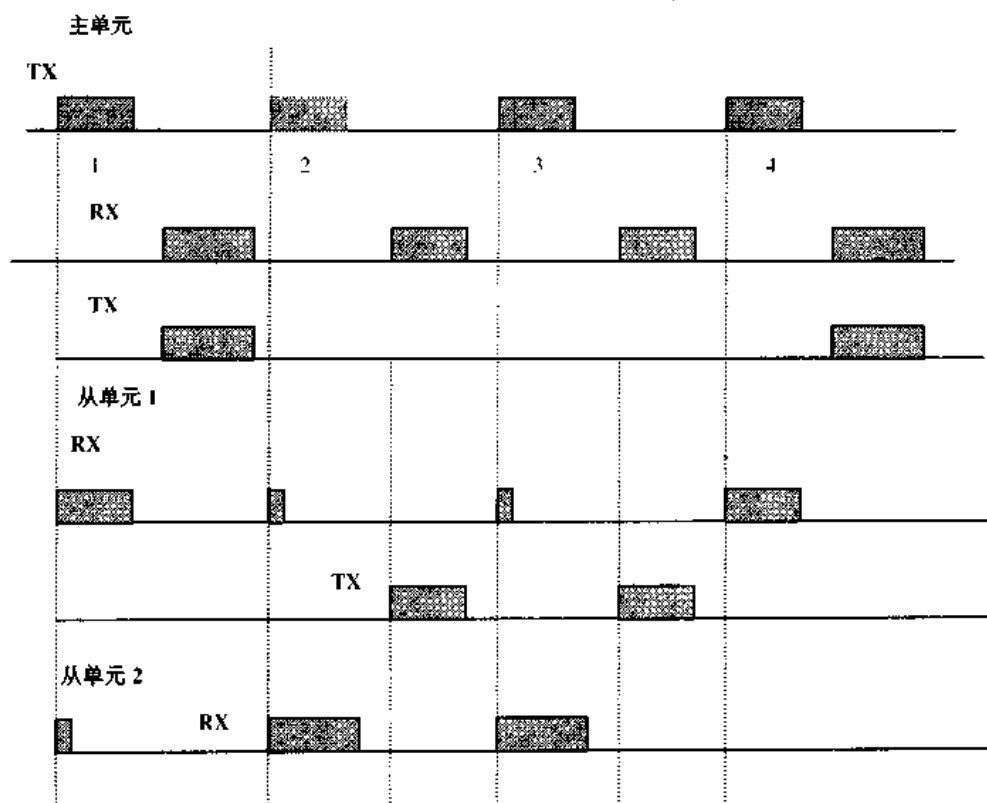


图 2.31 多从结构 RX/TX 定时

在下一次从一主时隙里，只有经过它自身 AM_ADD 编址的从单元可以返回。如果没有收到有效的 AM_ADD 值且涉及到保留的 SCO 从一主时隙，则只有从单元可以应答。在发送广播消息的情况下，从单元不得返回分组。在休眠模式中，用于访问请求的识别窗口是一个例外。

2.10 信道控制

本节描述如何实现匹克网的信道建立、单元增加及释放过程。另外，对共享同一区域的多匹克网操作也作了讨论。

2.10.1 主-从定义

匹克网中的信道特性完全由匹克网的主单元来确定。主单元蓝牙设备地址 (BD_ADDR) 确定跳频序列及信道识别码。主单元的系统时钟确定跳频序列的状态和定时设置。另外，主单元通过轮询 (POLL) 方式控制信道通信。

初始化连接的蓝牙单元定义为主单元 (与一个或多个从单元相连接)。注意：主单元和从单元的取名只是就信道上的协议而言。由于蓝牙每个单元的权重是完全一样的，也就是说任何一个单元都可能成为匹克网的主单元。所以一旦匹克网建立，主-从角色就完全可以进行互换。

2.10.2 蓝牙时钟

每一个蓝牙单元都有一个内部系统时钟，该时钟决定收、发信机的定时和跳频。因蓝牙时钟取自一个自由运转的本地时钟，且该时钟永远不会调整和关闭。对于与其他单元的同步，加到本地时钟的时钟补偿值可提供用于相互同步的临时蓝牙时钟。应当注意：蓝牙时钟与每天的时间无关。因此，它可用任何值初始化。蓝牙时钟提供蓝牙收、发信机的心脏脉搏。它的分辨率至少是 TX 或 RX 的时隙长度的一半或者 $312.5\mu\text{s}$ 。时钟周期约为一天。如果时钟用计数器来实现，那么 28 位计数器的计数值范围是 $2^{28}-1$ 。LSB 点以 $312.5\mu\text{s}$ 为单位跳动，并且给出的时钟频率是 3.2kHz。

在匹克网信道上的定时和跳频由主单元的蓝牙时钟来确定。当匹克网确立时，主单元时钟通过通信链路传送给从单元。各从单元在自己的本地时钟上增加一个补偿值以求得与主时钟同步。由于时钟不能受控，所以该补偿值必须有规律的进行更新。

在蓝牙接收机里，时钟确定临界时间并激发事件。对蓝牙系统来说有四个时间段非常重要： $312.5\mu\text{s}$ 、 $625\mu\text{s}$ 、 1.25ms 和 1.28s 。这些时间段分别与定时器位 CLK0、CLK1、CLK2 和 CLK3 对应。

当 CLK0、CLK1 都为“0”时，主-从传输以偶数时隙开始。

在不同的模式和状态里的蓝牙单元可具有不同的时钟特性：

- CLKN 本地时钟
- CLKE 预计时钟
- CLK 主时钟

CLKN 是一个自由运转的本地时钟，而且是所有其他时钟的参考。在高度活跃状态下，本地时钟用精度为 $\pm 20\text{ppm}$ 的晶体振荡器产生。在低度活跃状态下，如待机、保持、休眠，本地时钟可以用相对精度较差的 $\pm 250\text{ppm}$ 低功耗振荡器 (LOP) 产生。

CLKE 是一个呼叫单元用于利用接收方本地时钟的时钟估算值。CLKE 通过增加一个补偿值从 CLKN 参考时钟值得到。通过使用接收的 CLKN，呼叫加速了链接建立。

CLK 是匹克网的主时钟，用于匹克网中所有定时和时序安排。所有的蓝牙设备都使用 CLK 来安排它们的传输和接收时序。CLK 通过在本地图钟 CLKN 的基础上增加一个补偿值

获得。因为 CLK 同它自己的本地时钟 CLKN 唯一对应，所以对主单元来说，补偿值是“0”。而面对各个从单元来说，都对自身的 CLKN 加上一个适当的补偿值，以求得与主单元的 CLKN 一致。虽然在蓝牙设备里所有 CLKN 都以相同的标称速率运行，但相互之间的漂移将仍会导致 CLK 的不准确性。因此，在从单元里的补偿必须定期修改。CLK 近似于主单元的 CLKN。

2.10.3 状态综述

下面我们来说明用于蓝牙链路控制器的不同状态，如图 2.32 所示。这里有两种主要状态：STANDBY（待机）和 CONNECTION（连接）。另外还有七种子状态：呼叫、呼叫扫描、查询、查询扫描、主应答、从应答和从查询。子状态用来在匹克网中增加新的从单元的过渡状态。

要从一个状态转到另一个状态，要么使用蓝牙链路管理器命令，要么使用链路控制器的内部命令（类似于来自相关器或超时信号的激发信号）。

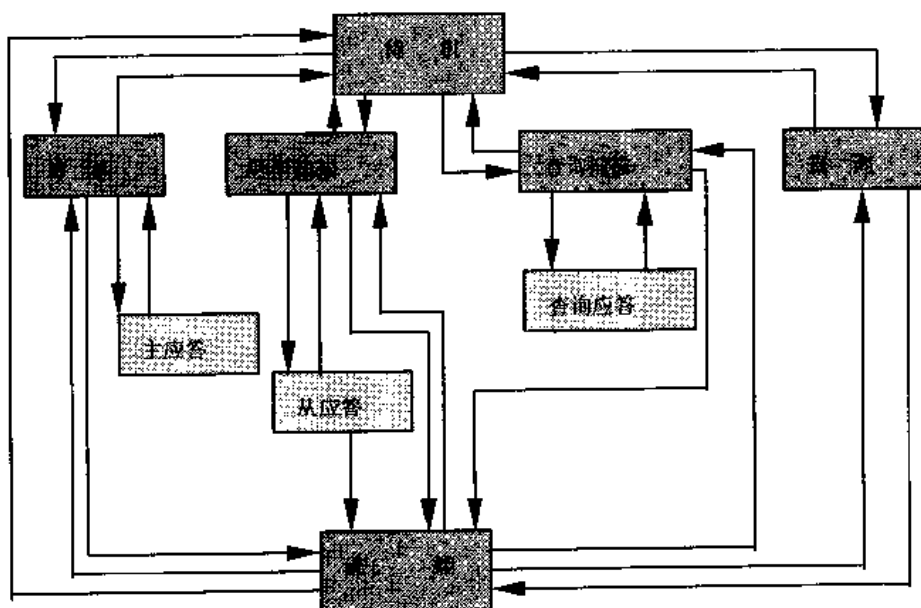


图 2.32 蓝牙链路控制器状态图

待机状态是蓝牙单元中的默认状态。在这个状态下，蓝牙单元处于低功耗模式，只有本地时钟以 LOP 精度（或更好的）运行。

控制器可以脱离待机状态进行扫描呼叫、查询消息、呼叫或自身查询。当对一个呼叫消息应答时，该单元不会再返回到待机状态，而是作为一个从单元进入连接状态。当执行呼叫成功时，该单元将作为一个主单元进入连接状态。有关扫描活动间隔来执行的内容将在呼叫扫描和查询扫描中讨论。

2.10.4 识别过程

为了建立新的连接，应使用查询和呼出过程。查询过程使一个单元能发现那些在范围内的单元，以及它们的设备地址和时钟。通过呼出过程，能够确定实际连接。只有通过蓝牙设备地址才能要求建立连接。时钟的确认将加速连接建立过程。建立连接的单元将可以执行呼叫过程并自动成为连接的主单元。

在呼出和查询过程中，分别使用设备识别代码（DAC）和查询识别代码（IAC）。在呼

叫扫描和查询扫描子状态里的单元根据各自的识别码和匹配相关器相关。

对于呼出过程，可以使用几种呼出方案。有一种必须是由各蓝牙设备支持的强制呼出方案。当单元是第一次使用时，就使用强制呼出方案，而且呼出过程紧跟查询过程。曾经使用强制呼出 / 扫描方案连接的两个单元可以考虑选择强制呼出/扫描方案。

1. 呼叫扫描

在呼叫扫描子状态里，单元在 $T_{w_page_scan}$ 扫描窗口内侦听自身设备识别码，并在该扫描期间单元以单跳频方式获取与设备识别码的相关器匹配。扫描窗口应有足够宽度来完成 16 跳呼叫扫描频率。

当单元进入呼叫扫描子状态时，它依据对应于该单元的呼叫跳频序列来选择扫描频率。该序列是一个 32 跳的跳频序列（或假设为一个 16 跳的简化跳频系统），在这里面每一个跳频点都是惟一的。呼叫跳频序列通过单元的蓝牙设备地址（BD_ADDR）来确定。在序列中的时段由单元的本地时钟 $CLKN_{16,12}$ 来确定（ $CLKN_{16,12}$ 假设为一个 16 跳的简化跳频系统）。也就是说，每隔 1.28s 就要选择一个不同的频率。

如果在呼叫扫描期间相关器超过了触发门限，单元将进入从单元应答子状态，有关内容参看后续的从应答描述。

呼叫扫描子状态可以从待机状态或连接状态进入。在待机状态下，没有建立任何连接，且该单元可以使用全部容量来执行呼叫扫描。在从连接状态进入呼叫扫描子状态前，单元将尽可能多地保留扫描容量。如有可能的话，单元可以在保持模式设置的 ACL 连接甚至休眠模式中使用。SCO 连接更希望不要被呼叫扫描中断。在这种情况下，呼叫扫描模式可以由具有更高优先权的 SCO 保留时隙中断。SCO 分组应当要求用最小容量 (HV3 分组)。扫描窗口将增加以最小化设置延迟。如果一个 SCO 链接使用 HV3 分组，且 $T_{sc0}=6$ 时隙，建议使用一个至少为 36 时隙 (22.5ms) 的总扫描窗口；如果两个 SCO 链接使用 HV3 分组，且 $T_{sc0}=6$ 时隙，建议使用一个至少为 54 时隙 (33.75ms) 的总扫描窗口。

扫描间隔 T_{page_scan} 定义为两个连续的呼叫扫描之间的间歇。这样就造成在扫描间隔等于扫描期间 $T_{w_page_scan}$ (连续扫描) 与最大为 1.28s 的扫描间隔或最大为 2.56s 的扫描间隔之间的差异。这三种情形将决定呼出单元的特性。也就是说，呼出单元使用 R0, R1 还是 R2。下面用表的形式来阐明 T_{page_scan} 和模式 R0, R1 和 R2 之间的关系。

虽然在 R0 模式下扫描是连续的，但扫描可以被类似于保留 SCO 时隙中断。扫描间隔信息包含于 FHS 分组的 SR 段中。

表 2.9 扫描间隔、重复序列及 R0、R1 和 R2 呼出模式之间的关系

SR 模式	T_{page_scan}	N_{page}
R0	连续	≥ 1
R1	$\leq 1.28s$	≥ 128
R2	$\leq 2.56s$	≥ 256
保留	..	..

在呼叫扫描过程中，蓝牙单元可以选择使用任意扫描方案(但查询应答消息返回后的呼叫扫描情况除外)。具体内容参看查询应答章节。

2. 呼叫

呼叫状态由主单元（源地址）用于激活并与一个在呼叫扫描子状态中周期性唤醒的从单元（目的地址）建立连接。主单元在不同的跳频信道反复地传输从单元的设备识别码（DAC）来追踪从单元。由于主单元和从单元的蓝牙时钟并不同步，主单元就不能确切地知道从单元什么时间该唤醒且工作在什么跳频点上。因此，它在不同的跳频点上传送一系列相同的设备识别码，并在传输间隔中侦听来自从单元接收应答。

主单元的呼叫过程由几个步骤构成。首先，从单元的设备地址用于确定呼叫的跳频序列，这是主单元用于找到从单元的序列。对于序列中的各阶段，主单元使用从单元的时钟估算值。这个估算值可能取自最后一次遇到该特定设备所交换的定时信息（那时，该特定设备可能是作为活动主单元），或者取自查询过程。通过使用从单元蓝牙时钟估算值 $CLKN$ ，主单元能够预测从单元在什么时间唤醒和处在哪个跳频点上。

从单元中的蓝牙时钟估算值可能完全是错误的。虽然主单元和从单元使用相同的跳频序列，但是它们在序列里使用不同阶段而且可能永远不能彼此相遇。为了补偿时钟漂移，在若干唤醒频率上的短时间间隔期间，主单元将发送自己的呼叫消息。实际上主单元也在当前时间的前后跳频点上传送预测的跳频。在各个 TX 时隙里，主单元顺序地在两个不同的跳频点上传送信息。由于呼叫消息是一个长度仅有 68 位的 ID 分组，在下面的 RX 时隙中有充足的时间（最短 224.5 μ s）切换频率合成器，接收机将连续地侦听对应于 ID 分组的两个 RX 跳频。RX 跳频又将依据呼叫应答跳频序列来选择。呼叫应答跳频序列也严格地与呼叫跳频序列相关。也就是说，对于每一呼叫跳频点都有一个对应的呼叫应答跳频点。在下一个 TX 时隙中，主单元将传送不同于前面跳频序列的两个跳频序列。频率合成器跳频的速率可增加到 3200 跳/秒。

在 79 跳系统和 23 跳系统之间应有差别。首先考虑 79 跳系统。正如以上所述，增加跳频速率，发射机能覆盖 16 时隙或 10 毫秒内覆盖 16 个不同的跳频。呼叫跳频序列被分成 16 种频率点的 A 和 B 两个呼叫序列。序列 A 中包括围绕当前预测跳频 $f(k)$ 的 16 种跳频。这里 k 由时钟估算值 $CLKN_{16-12}$ 来决定。所以第一个序列由跳频 $f(k-8), f(k-7), \dots, f(k), \dots, f(k+7)$ 组成。当主单元和从单元的蓝牙时钟差别在 -8×1.28 s 和 $+7 \times 1.28$ s 之间时，由主单元使用的频点之一将恰恰是从单元所侦听的跳频点。然而，由于主单元不知道从单元什么时间将进入呼叫扫描状态，主单元就必须重复 A 序列 N_{page} 次或直到收到应答为止。如果从单元扫描间隔与 R1 一致，重复次数至少是 128；如果从单元扫描间隔与 R2 一致，重复次数至少是 256。注意， $CLKN_{16-12}$ 每 1.28 秒改变一次。因此，每 1.28 秒，序列包含为呼叫跳频设置的不同频率。当主单元和从单元的蓝牙时钟差别小于 -8×1.28 秒或大于 $+7 \times 1.28$ 秒时，就要试探更多的跳频点。总的来说，由于只有 32 种专用唤醒跳频点，所以更多的跳频点是仍未试探过的跳频点。余下的 16 个跳频点用于新的 10ms 队列 B。第二个队列由跳频 $f(k-16), \dots, f(k-15), \dots, f(k-9), \dots, f(k+8), \dots, f(k+15)$ 组成。队列 B 重复 N_{page} 次。如果仍然没有得到回答，队列 A 应再重复 N_{page} 次，然后序列 A 和序列 B 交替使用，直到收到一个回答或超过呼叫超时。在侦听场合下，从单元返回应答，主单元进入主单元应答子状态。

对于在日本及某些欧洲国家所使用的 23 跳系统，该过程稍有不同。在 23 跳系统的情况下，呼叫跳频序列的长度被压缩为 16，为此，仅有一个单独的序列（序列 A）包含所有的呼叫跳频频率。该呼叫跳频序列阶段不是 $CLEK_{16-12}$ 而是 $CLEK_{15-12}$ 。不一定非要对从单元时钟作一个估算。

呼叫子状态可以从待机状态或连接状态进入。在待机状态下，不会建立连接且单元可以使用所有的容量来呼叫。在从连接状态进入呼叫子状态之前，该单元应当尽可能多地释放所有容量来扫描。为了保证这点，建议在保持或休眠状态上设置 ACL 链接。然而，SCO 链接不应被呼叫打扰，这就意味着呼叫只能被具有更高优先级保留 SCO 时隙中断。为了使呼出获得更多的容量，建议使用具有最小容量的 SCO 分组 (HV3 分组)。若是 SCO 链接，单队列的重复次数 N_{page} 将增加。此处假设是使用的具有 $T_{scn}=6$ 的 HV3 分组，该分组对应于 64kb/s 语音链路，如表 2.10 所示。

呼叫队列的建立不依赖于 SCO 链接的存在。即，SCO 分组在保留时隙上传送但并不影响用于非保留时隙上的跳频。

表 2.10 当为 SCO 链接时在队列重复和 R0、R1 和 R2 呼出模式的相互关系

SR 模式	非 SCO 链接	单 SCO 链接 (HV3)	双 SCO 链接 (HV3)
R0	$N_{page} \geq 1$	$N_{page} \geq 2$	$N_{page} \geq 3$
R1	$N_{page} \geq 128$	$N_{page} \geq 256$	$N_{page} \geq 384$
R2	$N_{page} \geq 256$	$N_{page} \geq 512$	$N_{page} \geq 768$

3. 呼叫应答过程

当呼叫消息由从单元成功接收时，在主单元和从单元之间存在近似 FH 同步。主从单元双方进入交换重要信息的应答规范过程，以继续连接设置。对于匹克网连接重要的是双方蓝牙单元都使用相同信道识别码、相同信道跳频序列，且它们的时钟同步。这些参数取自主单元。初始化连接的单元 (启动呼叫) 作为主单元 (仅在匹克网存在期有效)；信道识别码和信道跳频序列取自主单元的蓝牙设备地址(BD_ADDR)；定时取决于主单元时钟，从单元的本地时钟在加上补偿后应临时与主单元时钟同步。在开始时，必须把主单元的参数发送给从单元。主单元和从单元之间的初始化消息如表 2.11 及图 2.33、图 2.34 所示。

表 2.11 开始过程的初始化消息

步骤	消 息	方 向	跳频序列	识别码及时钟
1	从单元 ID	主→从	呼叫	从单元
2	从单元 ID	从→主	呼叫应答	从单元
3	FHS	主→从	呼叫	从单元
4	从单元 ID	从→主	呼叫应答	从单元
5	第一分组主单元	主→从	信道	主单元
6	第一分组从单元	从→主	信道	主单元

图 2.33 和图 2.34 描述了 $f(k)$, $f(k+1)$ 等频率是由从单元的 BD_ADDR 确定的呼叫跳频序列的频率。频率 $f'(k)$, $f'(k+1)$ 等是相应的呼叫—应答频率(从→主)。频率 $g(m)$ 属于信道跳频序列。

在步骤 1 中，主单元处于呼叫子状态，而从单元处于呼叫扫描子状态。假设在这一步里，呼叫消息（从单元的设备识别码）由主单元传送到从单元。经辨认从单元设备识别码，从单元进入第二步的从单元应答状态。此时主单元等待从单元的应答且当在第二步时得到从

单元的回答，主单元就进入第三步的主单元应答。注意，在最初的信息交换中,所有的参数都取自从单元的 BD_ADDR，而且只能使用呼叫跳频和呼叫应答跳频序列（它们也取自于从单元的 BD_ADDR）。

当主单元和从单元进入应答状态时，将固定它们输入的呼叫时钟和呼叫应答跳频选择，具体内容见呼叫应答部分的描述。

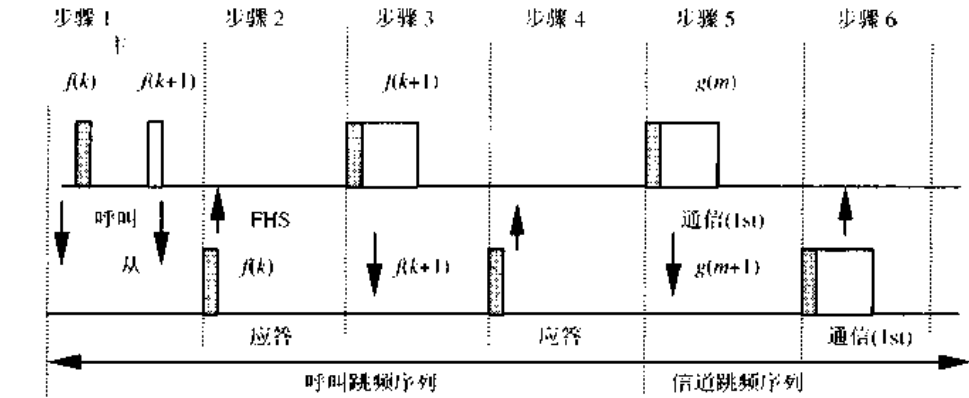


图 2.33 当从单元应答第一次呼叫消息时的连接通信初态

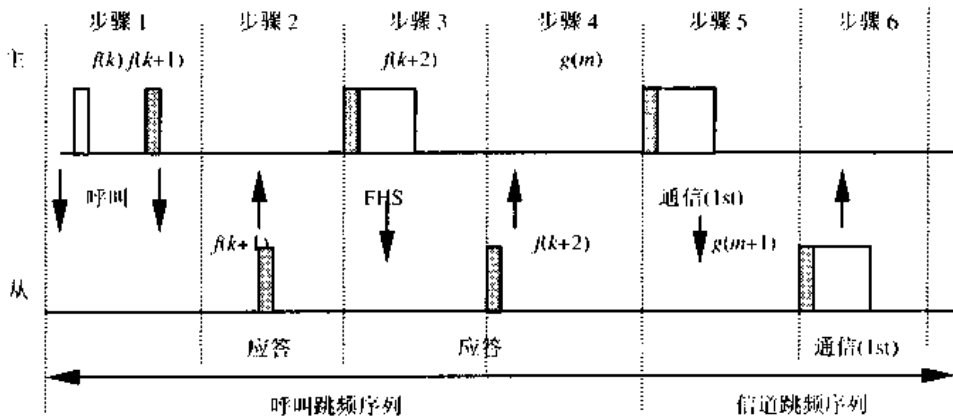


图 2.34 当从单元应答第二次呼叫消息时连接通信初态

a. 从单元应答

从单元在第一步中收到自身的设备识别码后，第二步就传送一个应答消息。该应答消息也仅由从单元的设备识别码构成。在开始接收呼叫消息（从单元 ID 分组）和在应答跳频符合呼叫消息接收的跳频后，从单元将以 625μs 传送该应答消息。因此从单元传输时间应与主单元传输时间相匹配。在初始化通信中，从单元仍然使用呼叫应答跳频序列向主单元返回信息。时钟输入 CLKN_{16,12} 被固定为呼叫消息接收到的时间值。

在发出应答消息后，将激活从单元的接收器（在应答消息开始后的 312.5μs），并等待 FHS 分组的到来。注意：FHS 分组在呼叫消息到达后经 312.5μs 到达。而且不像通常 RX/TX 定时情况下那样在 625μs 后。

如果进入连接状态之前设置失败，则执行以下过程。只要在呼叫应答超出前没收到 FHS 分组，从单元将持续侦听。然而，每隔 1.25ms，它就要根据呼叫跳频序列来选取下一个主-从跳频。如果在呼叫应答后没有收到任何信息，从单元在一个扫描周期内返回到呼叫扫描状态。扫描周期的长度取决于 SCO 时隙。如果在这个附加的扫描周期内，没收到呼叫消息，

从单元将以它固有的扫描间歇继续扫描并返回到第一次呼叫扫描状态之前的状态。

如果 FHS 分组由从单元在从应答状态下收到，从单元将在第四步中返回一个应答（只能是从单元的设备识别码）以确认 FHS 分组的接收（仍使用呼叫应答跳频序列）。该应答分组的传输将基于 FHS 分组的接收，然后从单元改变为 FHS 分组中接收到的信道（主单元的）识别码和时钟。只有主时钟的 MSB 26 位进行转发。定时假设只有当主单元在偶数时隙上进行传输，且收到 FHS 分组时，CLK₁ 和 CLK₀ 才为“0”。通过 FHS 分组的主时钟，主单元时钟和从单元时钟之间的补偿取值将确定并报告给从单元的链路管理器。

最后从单元将在第 5 步进入连接状态，从那时起，从单元将使用主单元的时钟和主单元的 BD_ADDR 以决定信道跳频序列和信道识别码。连接模式由主单元传输的一个 POLL 分组作为开始。从单元以任何类型的分组应答。如果从单元没收到 POLL 分组或主单元没收到应答分组，那么，在 FHS 分组确认后并在新的连接时隙数内，主单元和从单元将分别返回到呼叫与呼叫扫描状态。

b. 主应答

当主单元在第二步里收到来自于从单元的应答消息，它将进入主单元应答规则。它将冻结输入到呼叫跳频选择方案的当前时钟。然后主单元将在第三步传送一个 FHS 分组，该分组包含主单元的实时蓝牙时钟、主单元的 48 位 BD_ADDR 地址、BCH 奇偶位和设备类型。FHS 分组包含所有信道识别码的信息，而不需要与主单元设备地址的准确偏移。FHS 分组将在从单元应答的时隙后的主-从时隙的开始进行传送。FHS 的 TX 定时并不以主单元从单元的应答分组接收为基础。因此，FHS 分组将可以在接收到应答分组后的 312.5μs 后送出，而不像通常 RX/TX 定时情况下在接收分组的 625μs 后。

主单元送出 FHS 分组后，在 FHS 分组接收确认的第四步中等待来自主单元的第二应答。如果没有收到应答，主单元将重发 FHS 分组，但是时钟将要被修改且仍使用从单元的参数。主单元将重传（每次重传时，时钟都要作修改）直到第二次收到从单元应答或超出呼叫应答超时 PagerespTO。在后一种情况下，主单元将返回到呼叫状态，并向链路管理器传送一个出错信息。在 FHS 分组的重传过程中，主单元将一直使用呼叫跳频序列。

如果确实收到从单元的应答，主单元将改变为主单元参数，如信道识别码和主单元时钟。在 FHS 分组传送开始时，低时钟位 CLK₀ 和 CLK₁ 都为“0”而且不包含在 FHS 分组中。最后，在第 5 步主单元将进入连接状态。主单元 BD_ADDR 可用于改变为新的跳频序列和信道跳频序列。信道跳频序列在（伪）随机方式中使用所有 79 个跳频信道。现在，主单元将使用新（主单元）参数确定的跳频把它的第一个通信分组传送出去。该状态里的第一分组是一个由主单元传送的 POLL 分组。该分组将在 FHS 分组确认收到后的 newconnectionTO 时隙内发送。从单元将以任何类型的分组作为应答。如果 POLL 分组没有被从单元收到，或 POLL 分组的应答没有被主单元在 newconnectionTO 时隙内收到，那么主单元和从单元将分别返回到呼叫和呼叫扫描状态。

2.10.5 查询过程

在蓝牙系统中，查询过程定义为用于收端设备地址不为发端所知的应用。人们可以认为是类似于如打印机、传真机或与 LAN 的网关，或其他设备。而且，查询过程也可以用于发现其他蓝牙单元是否在范围内。在查询子状态中，发现单元搜集所有应答查询消息的蓝牙单元地址和时钟。如果需要的话，它可以通过前述呼叫过程同其中任何一个建立联系。

由发端广播的查询消息不包含任何有关发端的信息。然而，它将指出应答设备的类型。使用通用查询设备识别码(GIAC)可以查询任何蓝牙设备，使用一些专用查询识别码(DIAC)可以查询指定类型设备。查询识别码取自于保留的蓝牙设备字。

希望发现有其他蓝牙单元的单元将进入查询子状态。在该子状态中，它将连续以不同的跳频传送查询消息(即ID分组)。查询跳频序列总是取自GIAC的LAP。这样，当使用DIAC时，应用的跳频序列将可由GIAC的LAP产生。允许发现的单元将有规律的进入查询扫描子状态以便应答查询消息。查询应答是可选的，单元并不强迫要求应答查询消息。

1. 查询扫描

查询扫描子状态非常类似于呼叫扫描子状态。然而，与单元设备识别码扫描不同，接收方将扫描查询识别码，并有足够时间以完成16种查询频率扫描。扫描周期的长度称为 $T_{w_inquiry_scan}$ 。扫描在单跳频点上执行。就像在呼叫过程中一样，根据查询跳频序列扫描过程将使用32种专用查询跳频频率。这些频率由通用查询地址来确定。该状态由执行查询扫描的单元的本地时钟来确定。这个阶段每1.28秒改变一次。

除了通用查询识别码以外，该单元将可以扫描一个或更多专用查询识别码。然而，扫描将紧随由通用查询地址决定的查询跳频序列之后。如果查询消息在查询唤醒期被识别出来，那么蓝牙单元将进入查询应答子状态。

查询扫描状态可以从待机状态或连接状态进入。在待机状态下，没有建立任何连接，且该单元可以使用所有容量执行查询扫描。在从连接状态进入查询扫描子状态之前，单元将尽可能多的保留用于扫描的容量。如果希望该单元可以将ACL链接置于保持模式，甚至使用休眠模式，请见保持模式(HOLD)内容。SCO链接不得由查询扫描中断。在这种情况下，查询扫描可以被具有更高优先级的SCO时隙中断。SCO分组应当具有最低限度的请求容量(HV3分组)。扫描窗口 $T_{w_inquiry_scan}$ 应当增加，以增加向查询消息作应答的可能性。如果存在一条使用HV3分组和 $T_{SCO}=6$ 的SCO链接，建议使用至少36时隙(22.5ms)的完整扫描窗口。如果存在两条使用HV3分组和 $T_{SCO}=6$ 的SCO链接，建议使用一个至少达54时隙的(33.75ms)扫描窗口。

扫描间歇 $T_{inquiry_scan}$ 定义为在连续的两个查询扫描之间的间隔，最多为2.56秒。

2. 查询

查询子状态由企图找到新设备的单元使用。该子状态非常类似于呼叫子状态，TX/RX定时也同样类似于呼出过程。TX和RX频率紧跟在查询跳频序列和查询应答跳频序列之后，而且由通用查询识别码和发现设备的本地时钟确定。在查询传输期间，蓝牙接收方将扫描查询应答消息。当找到时，将读出整个应答分组(实际上就是FHS分组)。此后，查询单元继续执行查询传输。所以，在查询子状态中的蓝牙单元并不要求确认查询应答消息，它继续试着使用不同的跳频信道并侦听应答分组。如在呼叫子状态一样，将定义两个10ms的A和B队列，并把32个频率的查询跳频序列分为两个16跳频的部分。每个队列在一个队列使用之前，必须重复至少 $N_{inquiry}=256$ 次。为了搜集对通信质量要求不高情况下的所有应答，至少需要发生3次序列切换。结果，查询子状态至少应持续10.24秒，除非查询方搜集到足够的应答并决定放弃早期查询子状态。为了在对通信质量要求不高的情况下接收到所有的应答，查询方可延长查询子状态。如果一个查询过程自动定期启动(比如说每分钟10秒周期)，那么

在两个查询过程之间的间隔的确定必须随机确定。这样做是为了避免两个蓝牙单元同步它们的查询过程。查询子状态将一直持续到被蓝牙链路管理器停止（当它决定它具有足够应答数时），或达到超时时。

查询子状态可以从待机状态或连接状态进入。在待机状态下，没有建立任何连接，该单元可以使用所有的容量执行查询。在从连接状态进入查询子状态之前，该单元尽可能多的保留用于扫描的容量。为了作到这点，建议将 ACL 链接置于保持和休眠状态。然而，SCO 链接不得被查询干扰。这就意味着查询可以被具有更高优先级的保留 SCO 时隙中断。建议使用采用最小容量（HV3 分组）的 SCO 分组。如果是 SCO 链接，就要增加 $N_{inquiry}$ 重复次数。这里假设使用具有 $T_{SCO}=6$ 时隙间隔的 HV3 分组，该 T_{SCO} 对应于一个 64kb/s 的语音链路，如表 2.12 所示。

表 2.12 重复次数

重复次数	无 SCO 链接	单 SCO 链接 (HV3)	双 SCO 链接 (HV3)
$N_{inquiry}$	≥ 256	≥ 512	≥ 768

3. 查询应答

对于查询操作，只有一个从单元应答，而没有主单元应答。主单元将在用于应答的查询消息之间侦听。但读取应答后，它将继续传输查询信息。从单元查询应答规则完全不同于应用于呼叫的从单元应答规则。当在查询扫描子状态下收到查询消息时，应返回一个含有接受方地址的应答消息。该应答消息是一个普通的携带单元参数的 FHS 分组。然而，当几个蓝牙单元在物理层上与查询单元处于近距离链接，并且都同时向查询单元作出应答时，就会出现一个竞争问题。首先，每个蓝牙单元都有自己的运行时钟。因此，他们几乎不可能使用同一查询跳频序列的同一段。然而，为了避免在两个单元之间在同一个查询跳频信道真的发生同时唤醒的冲突。从单元的查询应答必须使用以下协议。如果从单元收到一个查询消息，它将在 0 到 1023 之间产生一个随机数。另外，它将固定当前输入值（阶段）为跳频选择方案。然后，从单元将在 RAND 时隙期间返回到连接或待机状态。在返回到连接或待机状态之前，该单元可以浏览呼叫扫描子状态。该呼叫扫描必须使用强制呼叫扫描方案。在 RAND 时隙之后，该单元将返回到查询应答子状态。收到第一个查询消息时，从单元将向主单元返回一个 FHS 应答分组。如果在扫描期间，在 $inqrespTO$ 超时限制内没有发生触发事件，从单元就返回到待机或连接状态。如果从单元收到查询消息并返回一个 FHS 分组，它将在查询跳频序列（这个阶段分辨率为 1.28 秒）阶段增加一个值为 1 的补偿，并且再次进入查询扫描子状态。如果从单元被再次激发，它就使用一个新的 RAND 来重复上述过程。时钟补偿将在每次 FHS 分组返回时累加。在一个 1.28 秒的探查窗口中，从单元平均应答 4 次，但每次都将在不同的时间和不同的频率上应答。可能的 SCO 时隙应当比应答分组具有更高的优先级。那就是说，如果一个应答分组被 SCO 时隙覆盖时，它就不能被传送，而且需等到下一个查询消息到来。

查询规则执行期间的消息发送如表 2.13 所示。在第一步中，主单元将使用查询识别码和它自己的时钟传输一个查询消息信息。从单元将以包含从单元设备地址，本地时钟和其他从单元信息的 FHS 分组作应答。该 FHS 分组在一个半随机时间上返回。FHS 分组在查询规则执行过程中不需确认，但只要主单元正以查询消息作搜索，它就应在其他时间或其他频率

上执行重传过程。

如果扫描单元使用可选扫描方案，在使用 FHS 分组应答查询应答后，它将用强制呼叫扫描方案在强制呼叫扫描周期 $T_{\text{mandatory_pscan}}$ 内完成呼叫扫描。每次发送查询应答后，该单元都将启动限制在一定强制扫描超时时间内的定时器。每次进行新查询应答时，定时器都将被重置。当单元进行呼叫扫描时，该单元将 SR 模式下使用强制呼叫方式。SR 模式在该单元全部呼叫扫描间隔周期当中使用，一直到定时器溢出为止。在查询过程后使用强制扫描模式将使得所有单元都可连接，即使所有单元都仍然不支持可选扫描模式。另外，在使用强制呼叫扫描时间的同时，可在 $T_{\text{mandatory_pscan}}$ 内并行使用可选呼叫扫描方案。强制呼叫扫描时间 $T_{\text{mandatory_pscan}}$ 包含于查询应答规则返回的 FHS 分组的 SP 段中，如表 2.14 所示。

表 2.13 查询规则

步骤	消息	方 向	跳频序列	识别码
1	ID	主-从	查询	查询
2	FHS	从-主	查询应答	查询

表 2.14 P0、P1、P2 扫描区间模式的强制扫描期

SP 模式	强制呼叫扫描时间
P0	$\geq 20\text{S}$
P1	$\geq 40\text{S}$
P2	$\geq 60\text{S}$
保留	—

2.10.6 连接状态

连接状态下，连接已建立，并且已经可以接收和发送分组。在通信两端的两个单元中，将使用信道（主单元）识别码和主单元蓝牙时钟，跳频方案使用信道跳频序列。主单元将在偶数时隙（ $\text{CLK}_{1-0}=00$ ）上开始传输，从单元则在奇数信道（ $\text{CLK}_{1-0}=10$ ）上开始传输。

连接状态以主单元用以检查主单元定时和信道跳频切换而发送的 POLL 分组为起点，从单元可以任何类型的分组应答。如果从单元没有收到 POLL 分组，或者主单元没有在 newconnectionTO 时隙数内收到新的应答分组，两设备都将返回到呼叫/呼叫扫描子状态。

在连接状态中第一个信息分组包含描述链路特性和有关蓝牙单元更多细节的控制消息。这些消息在各单元的链路管理器间进行交换。例如，它定义 SCO 链接和呼吸参数。然后用户信息的传输可由传送和接收的分组交替执行。

连接状态通过 detach 和 reset 命令结束。如果链路以正常方式断开连接，则使用 detach 命令。此时，蓝牙链路控制器中的所有设置数据仍然有效。reset 命令用于整个控制器处理强制清除。清除后，控制器必须重新设置。

在连接状态中，蓝牙单元可以进入多种操作模式，如活动模式、呼吸模式、保持模式和休眠模式。

1. 活动模式

在活动模式中，蓝牙单元积极参与共享信道。主单元基于来自和发送到不同从单元的通信要求对传输进行调度。另外，它还支持有序传输以保持从单元和信道同步。活动从单元则在一主一从时隙中侦听分组。如果活动从单元没有编址，它可以在下一次新的主单元传输之前处于睡眠状态。根据分组类型指示，可以得到主单元为它的传输所保留的时隙数。在该时间期间内，未编址的从单元不得侦听主一从时隙。同时，为了使从单元和信道保持同步，必须进行周期性主单元传输。由于从单元只需要信道识别码用于同步，因此可以使用任何类型

的分组。

2. 呼吸方式

在呼吸方式下，可减少由从单元负责的侦听周期。如果从单元参与 ACL 链接，它必须在每一 ACL 时隙中侦听主单元通信。通过呼吸模式，可以减少主单元与某一指定从单元开始传输所占用的时隙数。也就是说，主单元只能在特定时隙上开始传输。所以在呼吸时隙之间将存在长为 T_{sniff} 的规则间隔。

从单元必须在 D_{sniff} 呼吸时隙上对重复 $N_{\text{sniff_attempt}}$ 次的 T_{sniff} 进行侦听。如果从单元在 $T_{\text{sniff_attempt}}$ RX 时隙上的任一时隙接收到一个分组，它应继续侦听，直到它收到所有含有它自己 AM_ADDR 的分组。一旦它停止接收分组，它应在 $N_{\text{sniff_timeout}}$ RX 时隙内，或 RX 时隙的其余 $N_{\text{sniff_attempt}}$ 时隙内继续侦听。后者在占用时隙数上大于前者。

为了进入呼吸模式，主单元将通过 LM 协议发出呼吸命令。该消息包含呼吸间隔 T_{sniff} 和补偿值 D_{sniff} 。呼吸模式定时确定方法与 SCO 链接相似。另外，还采用一个初始化标志表明是使用初始化过程 1 还是过程 2。当当前主单元时钟(CLK₂₇)MSB 是“0”时，它将使用初始化过程 1；当当前主单元时钟(CLK₂₇)MSB 是“1”时，它将使用初始化过程 2。从单元根据初始化标志表示选用不同初始化方法，而与其自身时钟位值 CLK₂₇ 无关。由主从单元共同确定的主-从呼吸时隙将在时钟满足以下等式的时隙上进行初始化。

$$\text{CLK}_{27} \bmod T_{\text{sniff}} = D_{\text{sniff}} \quad \text{初始化过程 1}$$

$$(\text{CLK}_{27}, \text{CLK}_{26}) \bmod T_{\text{sniff}} = D_{\text{sniff}} \quad \text{初始化过程 2}$$

由主从单元共同确定的从-主呼吸时隙将在上面定义的主-从呼吸时隙之后的时隙上进行初始化。初始化后，可以通过在当前主-从的呼吸时隙时钟上增加固定间隔 T_{sniff} 获得下一主-从呼吸时隙的时钟值 CLK(k+1)：

$$\text{CLK}(k+1) = \text{CLK}(k) + T_{\text{sniff}} \quad (2-2)$$

3. 保持模式

在连接状态中，可以将指向某一从单元的 ACL 链接置于 HOLD 模式。这就意味着从单元暂时不能支持信道上的 ACL 分组（注意：可能仍将支持 SCO 链接）。使用保持模式，从单元仍可将容量空闲出来用于其他事务，如扫描、呼叫、查询或加入另一匹克网。处于保持模式的单元也可进入低功耗睡眠模式。在保持模式期间，从单元仍可维持其活动成员地址 (AM_ADDR)。

在进入保持模式前，主从单元应就从单元保持模式的持续时间达成一致，并对定时器以 holdTO 值初始化。当到达定时时间时，从单元将被唤醒与信道通信同步，并等待主单元的进一步指示。

4. 休眠模式

当从单元不必加入匹克网信道时，仍需保持与信道的同步，此时可进入休眠模式。该模式是指从单元处于一种低活性低功耗模式。在休眠模式下，从单元将放弃其活动成员地址 AM_ADDR，并接收两个用于休眠模式的新地址：8 位休眠成员地址：PM_ADDR，8 位识别请求地址：AR_ADDR。

PM_ADDR 能够将一个休眠从单元与其他休眠从单元区别开来。该地址用于由主单元初始化的解除休眠过程。除 PM_ADDR 以外，休眠从单元也可由它的 48 位 BD_ADDR 解除

休眠。全 0 的 PM_ADDR 是一个保留地址：如果一个休眠单元的 PM_ADDR 全为 0，它就只能被 BD_ADDR 解除休眠。在这种情况下，PM_ADDR 没有意义。而 AR_ADDR 则由从单元在由从单元初始化的解除休眠过程中使用。由于缺少 AM_ADDR，所有送往休眠从单元的消息必须由广播分组（全 0 的 AM_ADDR）携带。

休眠从单元为了重新同步和检查广播消息，将被在定期间隔上醒来并侦听信道。为了支持休眠从单元的信道识别和同步，主单元将提供信标信道。当从单元处于休眠状态时，信标结构将传送给从单元。当标志结构改变时，休眠从单元可通过广播消息来修改。

休眠模式除用于节能外，还可用于向一个主单元提供多于 7 个从单元的连接。但是无论何时最多只能同时激活 7 个从单元。然而通过从单元在匹克网内进行休眠和活动模式的切换，事实上可以连接的从单元数目可以很大（若使用 PM_ADDR 时，从单元为 255 个，如使用 BD_ADDR 时，从单元数目可更大）。而可被休眠的从单元数目也是无限的。

a. 信标信道

为了支持休眠从单元，当一个或多个从单元休眠时，主单元将建立一信标信道。信标信道由一个信标时隙或一个等距信标时隙队列构成。该队列以固定时间间隔作周期性传输。信标信道如图 2.35 所示。

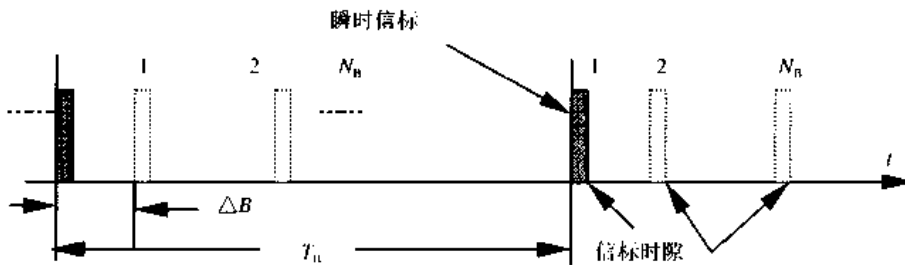


图 2.35 通用信标信道格式

时隙数为 N_B ($N_B \geq 1$) 的信标时隙队列由时隙数 T_B 定义。

队列中的信标时隙由 ΔB 分开。首个信标时隙的起点叫做瞬时信标，同时作为信标定时参照。这样，就可以在信道质量较差的环境当中的某一时间窗口内，选取信标参数 N_B 和 T_B ，以保证有充足的信标时隙用于休眠从单元同步。

休眠时，从单元将通过 LMP 指令接收信标参数。另外，瞬时信标的定时可以通过补偿 D_B 指明。正如 SCO 链接一样，可以使用初始化过程 1 或初始化过程 2。如果当前主单元时钟 (CLK_{27}) 的 MSB 是 0 时，主单元使用初始化过程 1；如果当前主单元时钟 (CLK_{27}) 的 MSB 是 1 时，主单元使用初始化过程 2。选择的初始化过程由 LMP 指令中的初始化标志携带。从单元将按照初始化标志指出的初始化方法进行初始化，而不必考虑它的时钟位 CLK_{27} 。定位在瞬时信标上的主-从时隙将在满足以下等式的时隙上进行初始化。

$$\begin{aligned} \overline{CLK_{27-1}} \bmod T_B &= D_B && \text{初始化 1} \\ (CLK_{27}, CLK_{26-1}) \bmod T_B &= D_B && \text{初始化 2} \end{aligned}$$

初始化后，下一次信标瞬时的时钟值 $CLK(k+1)$ 通过加一个固定间隔 T_B 到当前信标瞬时的时钟值上得到：

$$CLK(k+1) = CLK(k) + T_B \quad (2-3)$$

信标信道用于以下四个目的：

- 休眠从单元用于重新同步的主-从分组的传输；
- 携带休眠从单元用于改变信标参数的消息；

- 携带发往休眠从单元的通用广播消息；
- 解除一个或多个休眠从单元的休眠状态。

因为从单元可以与任何在适当信道识别码之后的分组同步，所以在信标时隙上传输的分组不必包含用于休眠从单元的指定广播消息，以能够用于同步；置于信标时隙上的唯一要求是存在现有主——从传输。如果无消息可以传送，则应由主单元发送 NULL 分组。如果确实有广播信息要发送给休眠从单元，则应在信标时隙队列中的每个信标时隙上重复传输广播消息的第一个分组。然而，类似于 SCO 链接上的同步通信可以中断信标传输。

b. 信标访问窗口

除了信标时隙外，访问窗口则定义为休眠从单元发送请求解除休眠的时间期间。为了增加可靠性，访问窗口可以重复 M_{access} 次 ($M_{\text{access}} \geq 1$)。在瞬时信标之后，访问窗口以一个固定延迟 D_{access} 开始。访问窗口的宽度也就是 T_{access} ，如图 2.36 所示。

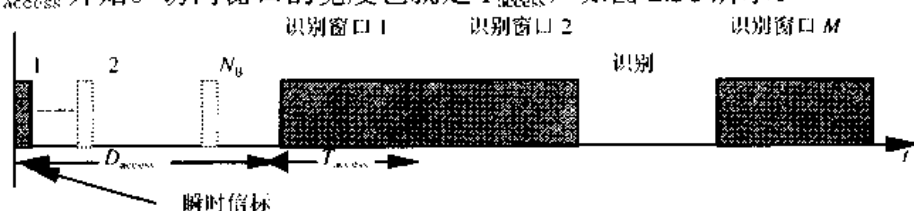


图 2.36 访问窗口定义

访问窗口可以支持不同的从单元访问技术，如轮询、随机访问或其他的访问方式。现在，只对轮询进行定义。轮询（POLL）技术如图 2.37 所示。

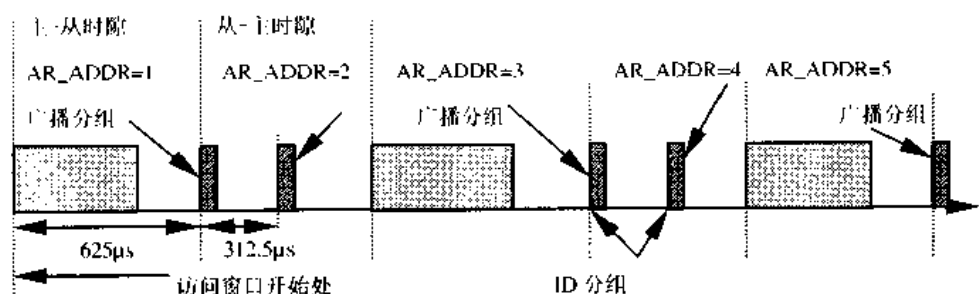


图 2.37 采用轮询技术的访问过程

在匹克网信道中使用同样的 TDD 结构，即：主-从的传输由从-主传输所替换。从-主时隙被分为两个 312.5μs 的半时隙。休眠从单元允许应答的半时隙对应于其识别请求地址 (AR_ADDR)。为了计算半时隙数以确定访问请求时隙，应使用访问窗口的起始点。如果在先前的主-从时隙上已接收到一个广播分组，那么只允许从单元在适当的从-主半时隙上发送一个访问请求。通过该方法，主单元轮询休眠从单元。

然而，如果必要，访问窗口内的时隙也可用于匹克网通信。例如，如果必须支持 SCO 连接，保留用于 SCO 链接的时隙可以用于携带 SCO 信息，而不是访问请求，即：如果访问窗口内的主-从时隙包含一个不同于广播分组的分组，那么以后的从-主时隙就不能再用于从单元访问请求。据定义的访问结构，没有进行通信的访问窗口内的时隙仍然可以使用。图 2.38 中，如果没有发生中断，则继续访问过程。

当从单元休眠时，将给出要使用何种访问方案。对于轮询方案，需要给出从-主访问时隙数 $N_{\text{acc-slot}}$ 。缺省给出访问窗口。然而，其动作取决于主单元在访问窗口内的适当时隙上

向从单元发送广播消息。信标时隙上广播 LMP 指令可指出下一访问窗口将不被激活。这将避免对请求访问的休眠从单元进行不必要扫描。

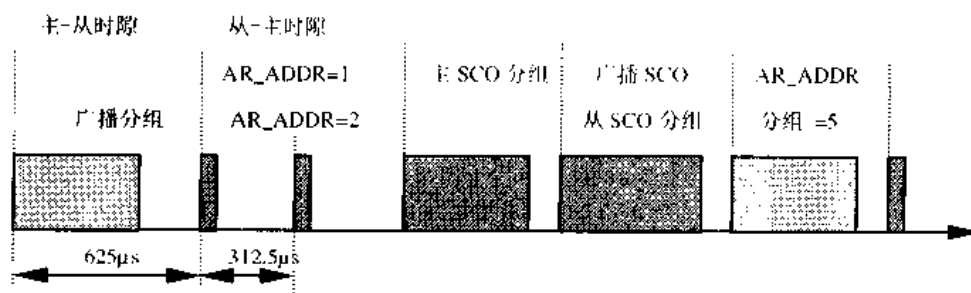


图 2.38 SCO 通信访问窗口的中断

c. 休眠从单元同步

休眠从单元大多数情况下都处于睡眠状态。然而，他们将周期地唤醒以与信道重新同步。信道上任何交换的分组都可用于同步。由于在信标时隙上必须进行主单元传输，休眠从单元将采用该信标信道以重新同步。休眠从单元将在该瞬时信标上唤醒，以读取在首个信标时隙上发送的分组。如果失败，它将在信标队列的下一时隙上重试。对于每一信标时隙，总共存在 N_B 个机会进行重新同步。在查找过程中，从单元可以扩大其搜索窗口，如图 2.39 所示。信标队列中信标时隙之间的间隔 T_B 应正确选择，以使连续搜索窗口之间不会相互覆盖。休眠从单元不必在每个瞬时信标上唤醒。反而，采用的睡眠间隔可以比信标间隔 T_B 更长。

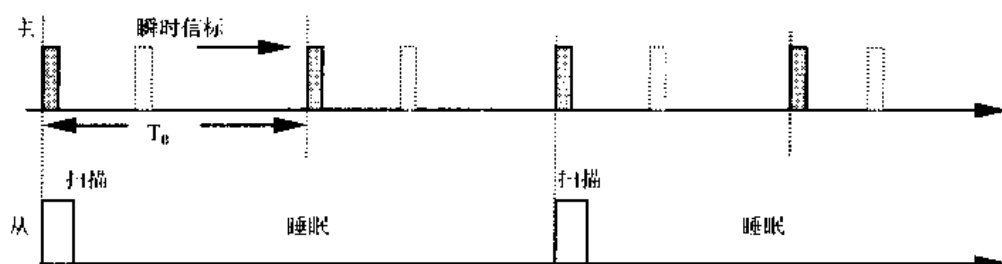


图 2.39 休眠从单元的扩充睡眠间隔

从单元睡眠窗口必须是 T_B 与 $D_{B\text{ sleep}}$ 的乘积。从单元唤醒所在的准确瞬时信标由使用 $D_{B\text{ sleep}}$ 的主单元指出。 $D_{B\text{ sleep}}$ 指出有关瞬时信标 ($0 < D_{B\text{ sleep}} < N_{B\text{ sleep}} - 1$) 的补偿值 (T_B 的乘积)。为了初始化唤醒周期，必须使用下列等式：

$$\begin{aligned} \text{LK}_{27-1} \bmod (D_{B\text{ sleep}} * T_B) &= D_B + D_{B\text{ sleep}} * T_B && \text{初始化 1} \\ (\text{CLK}_{27}, \text{CLK}_{26-1}) \bmod (N_{B\text{ sleep}} * T_B) &= D_B + D_{B\text{ sleep}} * T_B && \text{初始化 2} \end{aligned}$$

其中，如果当前主单元时钟的 MSB 是“0”时，选择初始化过程 1，在当前主时钟的 MSB 为“1”时，选择初始化过程 2。

当主单元需传送广播消息给休眠从单元时，它将使用信标时隙以用于广播消息。然而，如果 $N_B < N_{BC}$ ，在信标队列里的紧跟在最后一个信标时隙后的时隙将用于其余的 $N_{BC} - N_B$ 广播分组。如果 $N_B > N_{BC}$ ，广播消息将在所有 N_B 信标时隙上重复发送。

休眠从单元至少应读取在唤醒信标时隙中传输的广播信息。最小唤醒活动是读取用于重新同步的信道识别码和用于广播消息的分组头。

d. 休眠

主单元可以通过一个或几个 LMP 命令的交换休眠一个活动从单元。在进入休眠模式之前，从单元需分配 PM_ADDR 和 AR_DDR。每个休眠从单元都具有一个惟一的 PM_ADDR，

而 AR_DDR 却不一定是惟一的。当从单元休眠时, 信标参数由主单元给出。随后, 从单元将放弃其 AM_ADDR 而进入休眠模式。主单元一次只能休眠一个从单元。休眠消息由普通数据分组携带发送, 且利用 AM_DDR 地址标识从单元。

e. 由主单元发起的解除休眠过程

主单元可以通过发送含有休眠从单元地址的专用 LMP 解除休眠命令, 来激活一个休眠的从单元。该消息在信标时隙上的广播分组中发送。其中, 可以使用从单元 PM_ADDR, 也可以使用它的整个 BD_ADDR。消息也可以包含在从单元重新进入匹克网后从单元使用的活动成员地址 AM_ADDR。解除休眠消息可包含多个从单元地址, 以便可同时唤醒多个从单元。对于每一从单元, 都赋予了一个不同的 AM_ADDR。

收到解除休眠消息后, 匹配 PM_ADDR 或 BD_ADDR 的休眠从单元将退出休眠模式而进入活动模式。它将持续侦听主单元, 直到主单元利用 AM_ADDR 对它进行了编址操作。由主单元发送的第一个分组是一个轮询 (POLL) 分组。对应于 POLL 分组的返回分组将对从单元已解除休眠进行确认。如果在信标重复周期结束后, 在 newconnectionTO 个时隙中, 还没有收到从单元发回的任何应答分组, 主单元将再次将从单元解除休眠 (唤醒)。如果从单元在信标重复周期内的 newconnectionTO 个时隙中, 没有收到 POLL 分组, 从单元将使用相同信标参数返回休眠状态。在证实从单元为活动状态后, 主单元将决定从单元继续采用什么模式。

f. 由从单元发起的解除休眠

从单元能够通过访问窗口请求访问信道。访问窗口包含多个从一主半时隙。在这些半时隙中, 从单元可发出访问请求消息。从单元允许在指定半时隙中应答。该从单元处于休眠时, 对应于它接收到的访问请求地址 (AM_ADDR)。如果半时隙的次序是不固定的 (AR_ADDR 数可线性从 1 增加到 5), 也就是说, 在信标时隙中发送的 LMP 命令可以对访问窗口进行重新设置。当从单元希望访问信道时, 它可在一个合适的从一主半时隙中发出一个访问请求消息。该从单元访问请求消息是一个包含主单元设备识别码 (DAC) 的 ID 分组 (该情况下, ID 分组是一个没有分组尾的信道识别码)。休眠从单元只允许在某半时隙中发送访问请求消息。该半时隙位于前面的主一从时隙中, 且休眠从单元已收到一个广播分组。该广播信息可以包含任何一种不必与各休眠从单元有关的广播信息。如果没有可用的广播信息, 则应发送一个广播 NULL 分组或广播 POLL 分组。

在已发出访问请求后, 休眠从单元将侦听来自主单元的解除休眠消息。只要没有接收到解除休眠消息, 从单元就将在后面的访问窗口中重复访问请求。在最后一个访问窗口 (总共有 M_{access} 个窗口) 之后, 休眠从单元将在附加的 N_{poll} 个时隙内继续侦听解除休眠消息。如果在 N_{poll} 个时隙内, 在上次访问窗口结束后, 仍然没收到解除休眠消息, 从单元可以返回睡眠状态并在下一个瞬时信标后重新尝试进行访问。

收到解除休眠消息后, 匹配 PM_ADDR 或 BD_ADDR 的休眠从单元将退出休眠模式而进入活动模式。它将继续侦听主单元, 直到主单元通过 AM_ADDR 对该从单元进行编址。由主单元发送的第一个分组应是一个 POLL 分组。而对应于该 POLL 分组的返回分组将对从单元已被解除休眠进行确认。如果在最后一个访问窗口结束后的 N_{poll} 个时隙后的 newconnectionTO 个时隙内, 仍然没有接收到来自从单元的任何应答消息, 主单元将再次发送解除休眠消息给从单元, 从单元将以相同信标参数返回休眠状态。在确认从单元处于活动状态后, 主单元将决定从单元继续工作于何种模式。

g. 广播扫描窗口

在信标队列中，主单元可以支持对各休眠从单元广播消息。然而，通过向各休眠从单元指明在信标队列后有更多的广播信息，主单元将可以扩充其广播容量。在有限的时间窗口内，这一过程可由特定的用于控制活动从单元（或休眠从单元）的 LMP 指令序列实现。该时间窗口以瞬时信标作为起始，其持续周期在所发送的信标队列中的 LMP 指令中给出。

5. 轮询（Polling）方案

a. 活动模式下的轮询

一般，主单元完全控制匹克网。由于严格的 TDD 方式，从单元只能同主单元进行通信而不能与其他从单元进行通信。为了避免在 ACL 链接上发生冲突，从单元只有由主—从时隙分组头中的 AM_ADDR 编址时，才允许在从—主时隙上传送信息。如果在先前时隙中的 AM_ADDR 不匹配，或 AM_ADDR 不能从先前时隙中取出来，就不允许从单元传输信息。

在 SCO 链接上，轮询规则可稍作修改。从单元允许在 SCO 链接保留时隙上传输信息，除非先前时隙里的（合法）AM_ADDR 指出另一个不同的从单元。如果没有能从先前时隙里取出有效 AM_ADDR，从单元仍允许在保留 SCO 时隙上发送信息。

b. 休眠模式下的轮询

在休眠模式中，允许休眠从单元在访问窗口中传输访问请求。该访问窗口提供在先前主—从时隙中接收到的广播分组。活动模式下的从单元不再在广播分组后的从—主时隙上传输信息，而只是允许它们以指定地址发送信息。

6. 时隙保留方案和广播方案

SCO 链接的建立是通过链路管理器之间的协商来完成的。该链路管理器可通过 LMP 消息进行 T_{SCO} 和 D_{SCO} 等重要 SCO 定时参数交换。

匹克网主单元可以广播能够到达所有从单元的消息。广播分组由全零 AM_ADDR 进行标识。每个新的广播消息（可由多个分组携带）都以刷新指示作为起始（L_CH=10）。

广播分组不需确认。在通信质量要求不高的环境中，主单元可以执行 N_{BC} 次重传，以增加无差错传送的可能性。

为了支持休眠模式，主单元传输将以固定间隔进行。主单元传输可作为一个从单元可同步的信标。如果在信标事件中没有发生通信，将发送广播分组。

2.10.7 散射网

同一区域可支持多个匹克网。各匹克网都有自己的主单元，且其跳频相互独立。各匹克网的信道跳频序列和状态也由各自的主单元确定。另外，信道上传输的分组由不同信道识别码作为开始，并由主单元设备地址决定。随着匹克网的增加，冲突的可能性也将增加，在跳频扩频系统里，有些性能的降低是正常的。

如果多匹克网覆盖相同区域，一个单元可通过时间多路复用参与两个或两个以上的匹克网。为了加入合适的信道，它将使用相关主单元设备地址和适当时钟补偿来获取正确的状态。一个蓝牙单元可以在多个匹克网中作为从单元活动，但一个匹克网里只能有一个主单元。由于使用同一主单元的两个匹克网之间是同步的，而且使用相同的跳频序列，所以它们也就是同一个匹克网。构成不同匹克网间连接的一组匹克网就称为散射网。

主单元或从单元通过被另外匹克网的主单元呼叫，可成为另一匹克网中的从单元。另

一方面,一个匹克网的单元也可以呼叫另一匹克网中的主单元或从单元。由于呼叫单元总是以主单元身份出现,如果需要一个从单元,将需要进行主-从角色切换。

多路复用技术可用于匹克网间切换。假使只有 ACL 链接,一个单元可以在当前匹克网中进入保持或休眠模式。在此期间,它可通过改变信道参数而加入其他匹克网。处于呼吸模式中的各单元,在两个呼吸时隙之间,具有充足的时间访问另外的匹克网。如果 SCO 链接已经建立,则只能在两者之间的非保留时隙上访问其他匹克网。如果只有一条使用 HV 分组的 SCO 链接,则只能这样。在四时隙的链接期间,则可以访问另外一个匹克网。由于多匹克网不能进行同步,则必须采用保护时间以解释多匹克网间未对准的情况。也就是说,在任两个 HV3 分组间,只可有二个时隙能有效用于访问其他匹克网。

因为不同匹克网的两主单元时钟之间不同步,则加入两个匹克网的一个从单元必须兼顾两个加到它自身本地时钟的补偿,并创建一个或另一个主单元时钟。由于两个主时钟独立发生时间漂移,为了保证从单元与两个主单元同步,必须定期修改补偿值。

原则上讲,创建匹克网的单元是主单元。当从单元企图变成主单元时,则产生主-从单元(MS)切换。对于进行切换的两个单元,MS 切换将导致它们 TX 和 RX 定时的颠倒,即 TDD 切换。然而,由于匹克网参数取自设备地址和主时钟地址,主-从单元切换也自然将引起匹克网的重新定义:匹克网切换。新匹克网取自从单元的设备地址和时钟。匹克网切换的结果是不包含在该切换中的匹克网中的其他从单元必须转移到新的匹克网中,同时也改变它们的定时和跳频方式。新匹克网参数必须发给每一个从单元。假设单元 A 欲成为主单元,且单元 B 是以前的主单元。采取步骤如下:

- 从单元 A 和主单元 B 同意互换角色。
- 当两单元确认后,从单元 A 和主单元 B 将进行 TDD 切换,但将保留以前跳频方案(仍然使用单元 B 的设备地址和时钟),所以未发生匹克网切换。
- 单元 A 现在是匹克网的主单元。由于新旧主单元时钟异步,FHS 分组中给出的 1.25ms 时钟信息分辨率还不足以给出两匹克网的时隙边界。在发送 FHS 分组之前,新主单元 A 将发送一个给出新旧匹克网信道上主-从时隙起始之间延迟的 LMP 分组。该定时信息可取 0 到 1249 μ s 之间的值,其分辨率为 1 μ s。当在 FHS 分组确认后切换到新主单元定时器时,它将与 FHS 分组的时钟信息一起用于精确定位关联窗口。
- 在时间对准 LMP 消息后,主单元 A 仍将使用原匹克网参数,发送一个包含新 AM_ADDR 的 FHS 分组给从单元 B (FHS 分组头中的 AM_ADDR 为零地址)。在构成 ID 分组和由从单元在原跳频序列上传送的 FHS 分组确认后,主单元 A 和从单元 B 将转向 FHS 和时间校准 LMP 分组(至少对 A-B 连接)指出的新匹克网的新信道参数。
- 匹克网切换在各从单元上分别实施。主单元 A 将发送时间校准基准和一个 FHS 分组,并等待确认。FHS 分组的传送和确认将继续按单元 B 的旧匹克网参数进行(可将其与在连接建立过程中的呼叫跳频方案相比)。在使用由从单元发送的 ID 分组的 FHS 确认后,从单元将继续使用单元 A 的新设备地址和时钟进行通信。发送给每个从单元的 FHS 分组具有包含于 FHS 分组头里原 AM_ADDR 和包含于 FHS 分组有效载荷中的新 AM_ADDR (新 AM_ADDR 可能与原 AM_ADDR 相同)。
- 在接收到 FHS 分组确认后,新主单元 A 将切换为其自身时钟,并发送一个 POLL 分组以校验该切换。主单元和从单元将在 FHS 分组确认时启动超时为 $T_{\text{connectionTO}}$ 的时钟。如果没有接收到应答,主单元将重发 POLL 分组直到达到 newconnectionTO 超时。在超时达到后,

从单元和主单元将返回到原匹克网定时（但是 TDD 切换将保留不变）。主单元则再次发送 FHS 分组，并重复执行该过程。

- 在原匹克网中，新主单元将对每个从单元重复以上过程。

总之，MS 切换分两步完成：首先进行指定主单元和从单元的 TDD 切换，然后进行所有匹克网单元的匹克网切换。当所有的从单元都确认收到 FHS 分组时，每个单元将使用由新主单元定义新匹克网参数，这时匹克网切换完成。原从单元的 AM_ADDR、PM_ADDR 和其他属性信息，由原主单元发送给新主单元。该转发过程范围不在本过程范围之内。休眠从单元将使用原休眠参数来激活，并将改变为新匹克网参数，然后再使用该新休眠参数返回休眠模式。

2.10.8 节能管理

该特性用于保证低功耗操作，既用于处理分组的低层，也用于使用某种操作模式的高层。

1. 分组处理

为了降低功耗，分组处理将在 TX 和 RX 两端被最小化。在 TX 一端，功率通过仅发送有效载荷实现最小化。这意味着如果只有链路控制信息需要交换，将使用 NULL 分组。如果没有链路控制信息或者只具有 NAK（NAK 不需应答），就不会执行任何传输过程。如果有数据传送，则裁减有效载荷长度以只传送数据字节。在 RX 一端，分组处理以不同步步骤进行。如果在搜索窗口中没有发现有效识别码，收、发信机将返回到睡眠状态。如果发现识别码，则唤醒接收单元并开始处理分组头。如果 HEC 失败，则该单元应在分组头处理后将返回到睡眠状态。有效分组头指出该分组是否含有有效载荷，以及覆盖多少时隙。

2. 时隙占用

分组类型指出分组可占用多少时隙。在第一个时隙中未编址的从单元可在所占用的剩余时隙中进入睡眠状态。这可从 TYPE 代码中读取。

3. 低功耗模式

在降低功耗的连接状态中，存在三种模式。如果我们把这些模式按功耗递增顺序进行排序的话，那么呼吸模式功耗较高，然后是保持模式，具有最低功耗的是休眠模式。

2.10.9 链路监测

有很多原因能够引起连接中断，例如一台超出功率故障限制的设备。由于该情况的发生没有任何提前报警，所以当 AM_ADDR 重新分配给另一个从单元时，在主单元和从单元两端对链路进行监测，以尽可能避免冲突，则显得非常重要。

为了能够监测链路丢失，主单元和从单元将使用链路监测定时器 $T_{\text{supervision}}$ 。一旦收到经过 HEC 校验的分组和正确的 AM_ADDR，定时器就进行复位。如果在连接状态的任何时间上，定时器达到 supervisionTO 值，则连接复位。SCO 和 ACL 连接使用该同一超时值。超时时间 supervisionTO 可在 LM 层上进行协商。该监测超时取值应比保持和呼吸周期更长。休眠从单元的链路监测可通过对从单元解除休眠和重新休眠来实现。

2.11 跳频选择

总共存在 10 类跳频序列，其中 79 跳和 23 跳系统各有 5 类。

(1) 呼叫跳频序列：该序列含有 32 (16) 个唯一的唤醒频率，并均匀地分布在 79 (23) MHz 上，其周期长度为 32 (16)。

(2) 呼叫应答序列：该序列覆盖 32 (16) 个唯一的与当前呼叫跳频序列一一对应的应答频率。主单元和从单元可使用不同规则以获得相同序列。

(3) 查询序列：该序列含有唯一的在 79 (23) MHz 上均匀分布的 32 (16) 唤醒频率，其周期长度为 32 (16)。

(4) 查询应答序列：该序列覆盖 32 (16) 个唯一与当前查询跳频一一对应的应答频率。

(5) 信道跳频序列：该序列具有一个非常长的周期长度，该周期在一个较短间隔上并不会呈现重复模式，但在一个较短时间间隔内该序列将均匀分布在 79 (23) 跳频频率上。

对于呼叫跳频序列，重要的是能很容易的使状态进行前后转换。所以，需要一个计数器与跳频序列之间的 1-1 映射。对于每种情况，都需要从一主和一从两种跳频序列。

查询和查询应答序列通常利用取自跳频序列的 GIAC LAP 作为低地址而 DCI 作高地址部分，即使它与 DIAC 查询有关。

2.11.1 通用选择方案

选择方案由两部分组成：选择一个序列；在跳频频率上映射该序列。

跳频选择方案的一般框图如图 2.40 所示。

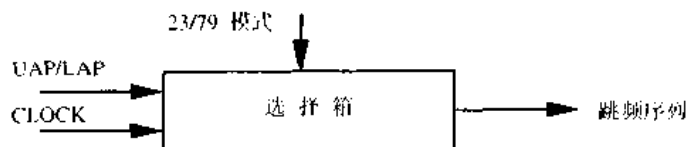


图 2.40 跳频选择方案的一般框图

从输入到特定的跳频序列映射在选择框内完成。基本上，输入是本地时钟和当前地址。在连接状态下，本地时钟 (CLKN) 由一个与主时钟 (CLK) 相等的补偿进行修改。其中，只能使用时钟的 27 位 MSB。而在呼叫和查询子状态下，将使用时钟的整个 28 位。在呼叫状态下，本地时钟将被修改为被叫单元对主单元的估算值。

地址输入由 28 位组成，即整个 LAP 和 UAP 的 4 位 LSB。在连接状态中，可使用主单元地址。在呼叫子状态下使用呼叫单元地址。而当为查询子状态时，将使用与 GIAC 对应的 UAP/LAP。输出则构成一个伪随机序列。覆盖 79 跳还是覆盖 23 跳系统，这取决于当前是什么状态。

对于 79 跳系统，选择方案将选择占用间隔为 64MHz 的 32 跳频段，并以随机次序访问这些跳频点一次。然后，选择一个不同的 32 跳频段，并依次类推。对于呼叫、呼叫扫描和呼叫应答子状态，将使用同一 32 跳频段（该段由地址进行选择，不同单元将具有不同呼出频段）。在连接状态下，输出构成在 79 跳或 23 跳之间变换的伪随机序列，这取决于所选择的跳频系统。对于 23 跳系统，频段大小为 16，其原理如图 2.41 所示。

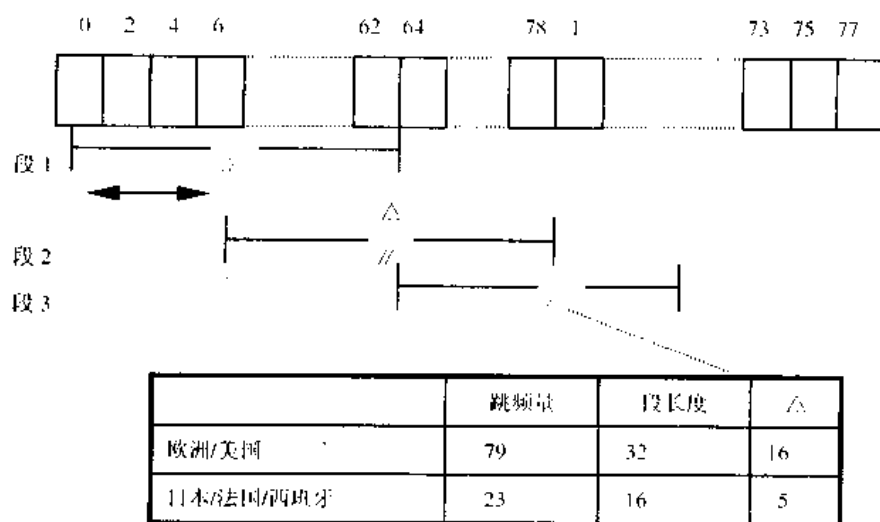


图 2.41 连接状态下的跳频选择方案

2.11.2 选择内核

79 跳和 23 跳系统的跳频选择内核分别如图 2.42 和图 2.43 所示。

X 输入决定 32 跳频段中的状态，而不论 Y1 和 Y2 在主—从及从—主传输之间选择哪一种传输方式。A 到 D 的输入决定段内顺序，E 到 F 的输入则决定对跳频频率的映射。该内核将代表含有跳频频率的寄存器。最终，将列有所有偶数跳频频率和所有奇数跳频频率的序列列表。这样，32 跳系统将以 64MHz 为一段，而 16 跳系统则将一 23MHz 为一段。

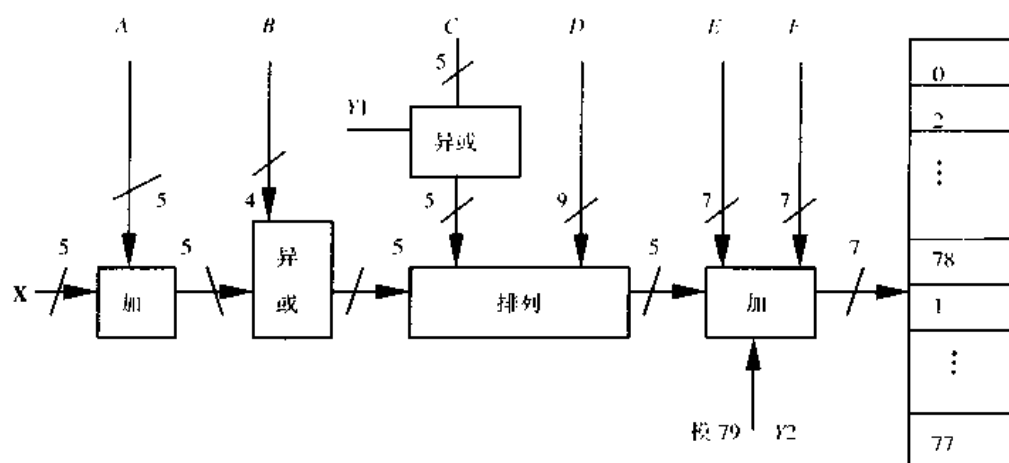


图 2.42 79 跳频系统跳频选择内核框图

该选择过程由一次加法运算、XOR 操作、排列操作、二次加法运算和寄存器选择顺序构成。在本节的其余部分，记号 A_j 用来表示 BD_ADDR 的第 j 位。

一次加法操作仅在该阶段上加一个常数，并对 32 或对 16 求模。对于呼叫跳频序列，一次加法运算是多余的，因为它仅在本频段内改变状态。然而，当不同频段前后连接起来（就像在信道跳频序列中一样）时，一次加法操作将对最终序列造成影响。

在 XOR 操作中，如果用 Z' 来表示一次加法运算的输出，则 Z' 的 4 个 LSB (Z'_0 、 Z'_1 、 Z'_2 、 Z'_3) 分别与地址位 $A_{22:19}$ 作模 2 的异或运算，输出分别为 Z_0 、 Z_1 、 Z_2 、 Z_3 。

排列操作，对于 79 跳系统包含从输入 5 到输出 5 的切换；对于 23 跳系统包含从输入

4 到输出 4 的切换，期间采用由控制字控制的操作方式。表 2.15 和表 2.16 说明如何利用控制信号 P 对蝶型操作进行控制。注意：P_{0,8} 对应 D_{0,8}，而 P_{i,9} 对应于 $C_i \oplus Y1$ ($i=0,1,\dots,4$)。蝶型操作可以前述的多路复用器实现

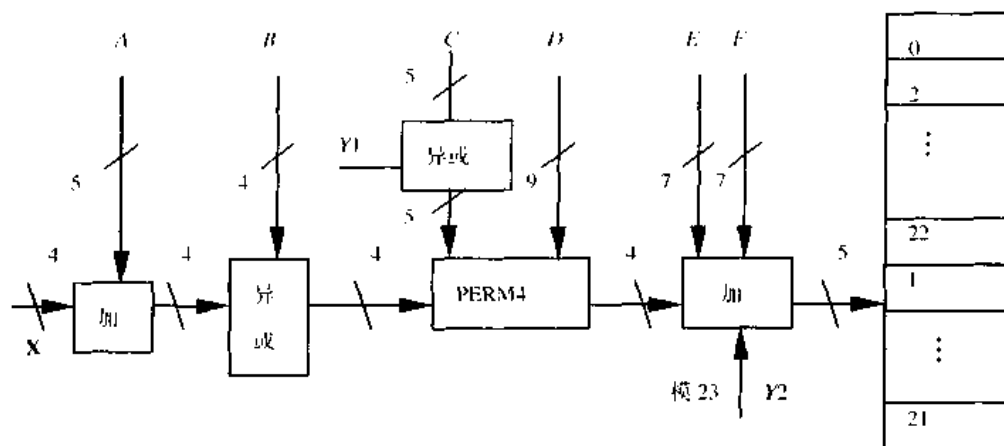


图 2.43 23 跳频系统跳频内核选择框图

表 2.15 79 跳系统蝶型控制

控制信号	蝶 型	控制信号	蝶 型
P0	{Z ₀ , Z ₄ }	P8	{Z ₁ , Z ₄ }
P1	{Z ₂ , Z ₄ }	P9	{Z ₀ , Z ₃ }
P2	{Z ₃ , Z ₂ }	P10	{Z ₂ , Z ₄ }
P3	{Z ₃ , Z ₄ }	P11	{Z ₁ , Z ₄ }
P4	{Z ₀ , Z ₃ }	P12	{Z ₀ , Z ₄ }
P5	{Z ₁ , Z ₄ }	P13	{Z ₁ , Z ₂ }
P6	{Z ₀ , Z ₂ }		
P7	{Z ₃ , Z ₄ }		

表 2.16 23 跳系统蝶型控制

控制信号	蝶 型	控制信号	蝶 型
P0	{Z ₀ , Z ₁ }	P8	{Z ₀ , Z ₂ }
P1	{Z ₂ , Z ₃ }	P9	{Z ₁ , Z ₃ }
P2	{Z ₀ , Z ₃ }	P10	{Z ₀ , Z ₄ }
P3	{Z ₁ , Z ₂ }	P11	{Z ₁ , Z ₂ }
P4	{Z ₀ , Z ₂ }	P12	{Z ₀ , Z ₄ }
P5	{Z ₁ , Z ₄ }	P13	{Z ₂ , Z ₃ }
P6	{Z ₀ , Z ₁ }		
P7	{Z ₂ , Z ₃ }		

二次加法操作只是在排列操作输出上增加一个常数。结果，16 跳或 32 跳频段将映射到不同的跳频频率上。二次加法操作采用模 79 或模 23，取决于系统类型（欧洲/美国或其他国家采用了不同系统）。

加法器的输出代表 79 或 23 跳系统寄存器组。寄存器采用对应于跳频频率 0 到 78 或 0 到 22 的同步代码字进行载入。注意：寄存器组的高半部分包含偶数跳频频率，而低半部分则包含奇数跳频频率。

2.11.3 控制字

内核控制字 P 由综合控制信号 X, Y1, Y2, 和 A 到 F 来控制。在呼叫和查询过程中，输入 A 到输出 E 使用上述两表中相应栏中所给出的地址值。另外，还将使用输入 X, Y1 和 Y2，但没有使用输入 F。在 79 跳系统中，时钟位 CLK_{6,2}（即输入 X）指定长度为 32 的序列中的状态。而在 23 跳系统中，CLK_{5,2} 指定长度为 16 的序列中的状态。对两种系统来说，CLK₁

(即输入 Y1 和 Y2) 都用于 TX 和 RX 之间的选择。输入地址决定段内的序列顺序。对跳频的最终映射由寄存器内容来决定。

以下，我们将对时钟的三种类型作出区分。这三种时钟类型是匹克网的主时钟、蓝牙单元的本地时钟和呼叫蓝牙单元的时钟估算值。这些类型形式标注以下：

- CLK_{27,0}: 当前匹克网的主时钟；
- CLKN_{27,0}: 单元的本地时钟；
- CLKE_{27,0}: 呼叫单元对被叫单元本地时钟的估算值。

在连接状态中，输入 A、C 和 D 是地址与时钟的位 XOR 运算的结果。在 XOR 操作中，每两位高位一起进行 XOR 运算。因而，在每 32 (16) 时隙后，在 79 跳 (23 跳频) 系统中，就选择一个新的长度为 32 (16) 的频段。在一个特定频段中的序列顺序将在一段很长的时间内不进行重复。于是，整个跳频序列将由各 32 跳频频段构成。由于每个 32 跳序列都占用了超过 80% 的 79MHz 频带，则可获得分布在一短时间间隔上所希望的频率。

在呼叫扫描中，扫描单元的蓝牙设备地址用作地址输入。在查询扫描中，GIAC LAP 和 DCI (如 A_{27,24}) 的四位 LSB 用作跳频序列的地址输入。很自然，为了发送识别码，应在接收方相关器中使用 GIAC 和 DIAC。使用哪一查询识别码则取决于查询的目的。

5 个 X 输入位的变化取决于该单元的当前状态。在呼叫扫描和查询扫描子状态中，将使用本地时钟(CLKN)。在连接状态中，主单元时钟 (CLK) 用作输入。这种情况将稍复杂于其他状态。

在 79 跳系统的呼叫状态中，呼出单元将使用 A 队列开始，即 {f(k-8), ..., f(k), ..., f(k+7), ...}，这里 f(k)是在呼入单元里当前接收器频率的估算值。很清楚，标志 k 是所有输入的一个函数。在每 1.28 秒间隔中，就有 32 个可能的呼出频率。这些频率中的一半属于 A 队列，剩下的 (即 {f(k+8), ..., f(k+15), f(k-16), ..., f(k-9)}) 属于 B 队列。为了实现 A 队列的 -8 补偿，可以在时钟位上加上一个常数 24 (这就相当于对模 32 减 8 求其补数)。很明显，B 队列可以通过对 8 加补偿来实现。为避免在呼出和扫描单元间可能重复的错误匹配，队列内顺序也要作周期性地变化。于是：

$$X_p^{(79)} = [\text{CLKE}_{16,12} + k_{\text{offset}} + (\text{CLKE}_{4,2,0} - \text{CLKE}_{16,12}) \bmod 16] \bmod 32 \quad (2-4)$$

其中， $k_{\text{offset}} = 24$ (A 队列)，或 8 (B 队列) (下同)。

A 和 B 队列的每次切换都可以通过在 k_{offset} 当前值上增加 16 来完成 (初始值为 24)。

在 23 跳系统的呼叫子状态中，呼出单元只使用 A 队列。为了使用 f(k-8)作为起始，将使用 8 作为常量补偿。此外，由于以 16 为模作加法运算，所以只需要 4 位。因此：

$$X_p^{(23)} = [\text{CLKE}_{15,12} + 8 + \text{CLKE}_{4,2,0}] \bmod 16 \quad (2-5)$$

在呼叫扫描子状态中的识别其自身识别码的单元将进入从单元应答子状态。为了消除由于不符合本地时钟 CLKN 和主单元时钟估算值 CLKE 而造成链路丢失的可能，应以当前值固定该四位的 CLKN_{16,12} 值。该值将固定为在检测到接受方识别码所在时隙当中所具有的内容。注意，实际本地时钟并没停止，且该值是为了用于创建固定一段时间的 X-输入的所用各位的值。被固定的值采用星号 (*) 标志。

对于每一应答时隙，被叫单元将使用一个 X-输入值，该值 (模 32 或 16) 只比先前应答时隙中的值大 1。然而，首个应答值将使用与识别码确认时值相同 X-输入值进行。设 N 是从 0 开始的计数器，那么，第 (N+1) 次应答时隙 (第一次应答时隙紧随在现在应答呼叫时隙后的时隙) 的 X-输入应为：

$$X_{\text{prc}}^{(79)} = [\text{CLKE}_{16:12}^* + N] \bmod 32 \quad (2-6)$$

$$X_{\text{prc}}^{(23)} = [\text{CLKE}_{15:12}^* + N] \bmod 16 \quad (2-7)$$

以上等式分别用于 79 跳和 23 跳系统。

在从单元确认呼叫的时隙中的计数器 N 被置为零。然后，每次 CLKN_i 被置为零时计数器的值就加 1。它对应于主单元 TX 的起始时隙。X-输入通过此方法进行构造，直到收到第一个 FHS 分组，且已将后续分组发送出去为止。在此之后，该从单元将使用在 FHS 分组里收到的参数进入连接状态。

呼出单元收到从单元的应答就进入主单元应答子状态。很明显，主单元也必须把从单元时钟估算值固定为触发被叫单元应答的值。当接收从单元应答时（因为只有 CLKE_i 与相应的呼叫传输不同），它将相当于使用时钟估算值。这样，当从单元 ID 分组接收到时，其值将被固定。除使用时钟各位以外， k_{offset} 的当前值也必须固定。主单元将采用与被叫单元相同的做法调整它的 X-输入，即：在每一次 CLKE_i 置为 0 时此值加 1。第一次增加应在发送 FHS 分组到呼入单元前完成。设 N 为从 1 开始的计数器，则形成 X-输入的规则为：

$$X_{\text{prm}}^{(79)} = [\text{CLKE}_{16:12}^* + k_{\text{offset}}^* + (\text{CLKE}_{4:2,0}^* - \text{CLKE}_{16:12}^* \bmod 16 + N) \bmod 32] \bmod 32 \quad (2-8)$$

$$X_{\text{prm}}^{(23)} = [\text{CLKE}_{15:12}^* + 8 + \text{CLKE}_{4:2,0}^* + N] \bmod 16 \quad (2-9)$$

以上等式分别用于 79 跳和 23 跳系统。

每次 CLKE_i 的值置为 0 时， N 值加 1，这与主单元的 TX 时隙的起始位置相对应。

查询子状态的 X-输入与呼叫子状态中使用的 X-输入十分相似。因为没有给出任何特定单元，所以将使用查询者的本地时钟 CLKN 。而且，使用两队列补偿中的哪一个作为开始在这个状态中无关紧要。因而，

$$X_i^{(79)} = [\text{CLKE}_{16:12} + k_{\text{offset}} + (\text{CLKE}_{4:2,0} - \text{CLKE}_{16:12} \bmod 16) \bmod 32] \bmod 32 \quad (2-10)$$

$$X_i^{(23)} = [\text{CLKE}_{15:12} + 8 + \text{CLKE}_{4:2,0}] \bmod 16 \quad (2-11)$$

以上等式分别用于 79 跳和 23 跳系统。

GIAC LAP 和 DCI 的四个 LSB ($A_{27:24}$) 用作跳频序列发生器的地址输入。

查询应答子状态类似于接收 X-输入的从单元应答状态。然而，计数器 N 并不是在 CLKN_i 的基础上增加的，而是在每次 FHS 分组被传送给查询者作为应答后进行增加。

GIAC LAP 和 DCI 的四位 LSB ($A_{27:24}$) 可用作跳频序列发生器的地址输入。发生器的其他输入位与呼叫应答情况形一样。

在连接状态中，在信道跳频序列生成过程中使用的时钟位总是依据主时钟 CLK 确定。地址位则取自主单元的蓝牙设备地址。

2.12 蓝牙音频

在蓝牙无线接口上，可以使用 64kb/s 的对数 PCM (A-规则或 μ -规则)，或者 64kb/s 的

表 2.17 在无线接口上支持的语音编码方式

语音编码	
线性	CVSD
8 位对数	A-规则
	μ -规则

CVSD（连续变化斜率增量调制器）。后一形式主要采用音节压扩增量调制算法。

在线路接口上的语音代码应当有和 64kb/s 的对数 PCM 一样或更好的质量。表 2.17 列出了无线接口所支持的语音编码方式。合适的语音编码将在链路管理器之间协商后进行选择。

2.12.1 对数 PCM 编译码器 (CODEC)

由于在无线接口上的语音信道可以支持 64kb/s 的信息流,所以在传输中可以使用 64kb/s 的对数 PCM 进行传输,也可使用 A-规则或 μ -规则进行压缩。在有线接口使用 A-规则和无线接口使用 μ -规则的情况下,可执行 A-规则到 μ -规则的转换。压缩方式遵循 ITU-T 建议 G.711。

2.12.2 连续变化斜率增量调制编译码器 (CVSD CODEC)

无线接口上语音的较健壮的格式为增量调制。该调制方案具有某种波形,其中输出各位将指出预计值是否大于输入波形。为了减少斜率过载影响,应使用音量压扩技术:步长可根据平均信号斜率进行修改。CVSD 编码器的输入是 64K 采样的线性 PCM。CVSD 编码器和 CVSD 译码器的结构如图 2.44 所示,系统以 64KHz 来定时。图 2.44 中的累加器如图 2.45 所示。

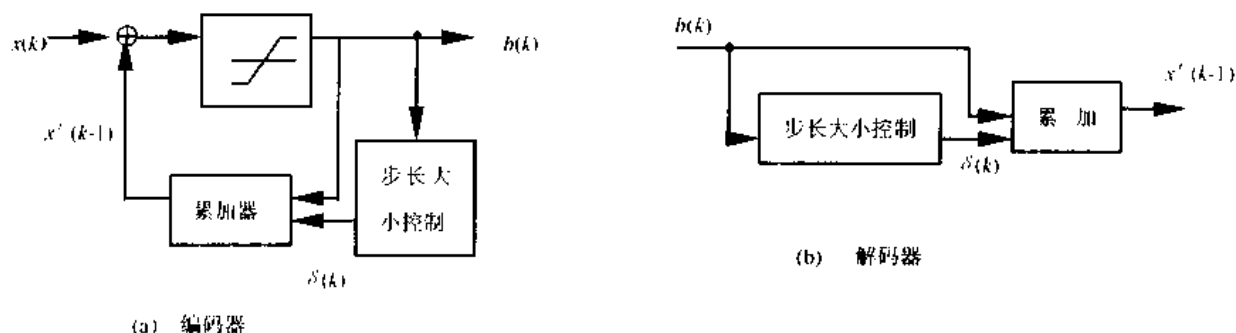


图 2.44 具有声音压扩的 CVSD 编解码器框图

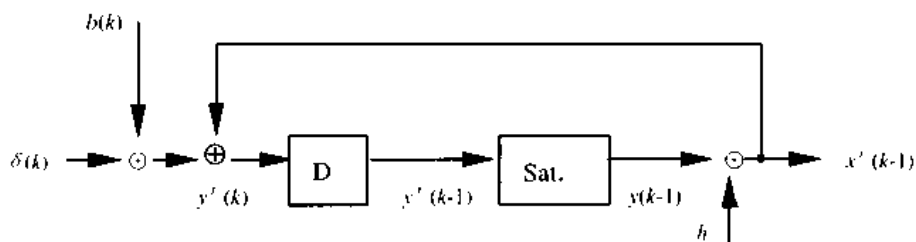


图 2.45 累加器过程

CVSD 输出位为 $b(k)$, 累加器内容为 $y(k)$, 步长为 $\delta(k)$ 。设 η 为累加器的延迟因子, β 表示步长的延迟因子, a 表示音节压扩参数。后一个参数通过考虑最近 K 的输出位监测斜率。

$$x'(k) = \eta y(k) \quad (2-12)$$

然后, CVSD 编码内部状态将依据以下等式来进行修改:

$$b(k) = \text{sgn} \{ x(k) - x'(k-1) \} \quad (2-13)$$

$$a = \begin{cases} 1, & \text{如果在上次 } K \text{ 输出位里的 } J \text{ 位是相等的} \\ \text{否则} \end{cases} \quad (2-14)$$

$$\delta(k) = \begin{cases} \min \{ \delta(k-1) + \delta_{\min}, \delta_{\max} \}, & a = 1 \\ \max \{ \beta \delta(k-1), \delta_{\min} \}, & a = 0 \end{cases} \quad (2-15)$$

$$y(k) = \begin{cases} \min \{ y'(k), y_{\max} \} \\ \max \{ y'(k), y_{\min} \} \end{cases} \quad (2-16)$$

$$\text{其中, } y'(k) = x'(k-1) + b(k) \delta(k) \quad (2-17)$$

表 2.18 CVSD 参数值

参 数	值
q	1-1/32
p	1-1/1024
J	4
K	4
$\delta_{\min}$	10
$\delta_{\max}$	1280
$y_{\min}$	-2^{15} 或 $-2^{15}+1$
$y_{\max}$	$2^{15}-1$

在这些等式中, $\delta_{\min}$ 和 $\delta_{\max}$ 分别是最小步长和最大步长; 而 $y_{\min}$ 和 $y_{\max}$ 分别是累加器的正、负饱和度。

对于 64 kb/s 的 CVSD, 必须使用表 2.18 列出的参数。

这些数字基于累加器输出的 16 位精度值。由这些值可得到累加器延迟的 0.5 毫秒时间和步长延迟的 16 毫秒常量时间。

2.12.3 错误处理

在 DV 和 HV3 分组中, 语音不受 FEC 保护。在通信质量要求不高的情况下, 语音质量取决于语音编码方式的稳定性。尤其 CVSD 在白噪声背景中对随机位错相当不敏感。然而, 由于信道识别码或 HEC 测试不成

功而拒绝分组时, 就必须采取措施来填补丢失的语音段。

HV2 分组中的语音有效载荷受 2/3 比例 FEC 的保护。如果发生不可纠正的错误, 这些错误应当被忽略。也就是说, 从含有未纠正错误的 15 位 FEC 段中, 应使用在 FEC 译码前发现的 10 位信息部分。HV1 分组由 1/3 比例 FEC 进行保护。在大部分检测方式中, 将不会再出现未纠正的错误。

2.12.4 一般音频要求

对于 A_规则或 η 规则的对数 PCM 编码信息来说, 要求信号遵循 ITU-T G. 711。

16 位线性 PCM 和 CVSD 编码器的接口处的完全摆幅定义为 3dBm。数字 CVSD 编码测试信号由合法测试文件提供。该信号由一个参考 CVSD 编码器的软件工具所产生。数字编码器输入信号 (1020Hz, 正弦波) 生成的测试信号有 -15 dBm 的标称功率。当 CVSD 编码的测试信号经 CVSD 接收链馈送时, 标称输出功率应为 -15 ± 1.0 dBm。

蓝牙的音频质量要求由发射方确定。64 kSPS 的线性 PCM 输入信号必须有 4KHz 以上的频谱功率强度。基准输入信号的设置由基准译码器 (在站点上有效) 传输和发送编码。64 kSPS 的线性 PCM 输出在译码信号的 4~32 kHz 带内的功率频谱强度, 应当比 0~4kHz 范围内的最大值低于 20dB。

2.13 蓝牙编址

2.13.1 蓝牙设备地址 (BD_ADDR)

每一蓝牙的收、发信机都分配有一个 48 位的蓝牙设备地址 (BD_ADDR)。该地址取自 IEEE802 标准, 分为三个部分: 由 24 位构成的低地址部分 (LAP 段); 由 8 位构成的高地址部分 (UAP 段); 由 16 位构成的非有效地址部分 (NAP 段)。LAP 和 UAP 构成 BD_ADDR

的有效部分, 获得的整个地址空间为 2^{32} , 如图 2.46 所示。

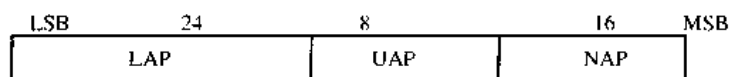


图 2.46 BD_ADDR 格式

2.13.2 识别码

在蓝牙系统中, 共定义了三种不同的识别码: 设备识别码 (DAC)、信道识别码 (CAC) 和查询识别码 (IAC)。

通用查询操作使用 GIAC, 而指定查询操作则使用 63 个指定的 IAC (DIAC)。所有代码都取自 BD_ADDR 中的 LAP。设备识别码用于呼叫、呼叫扫描和呼叫应答状态中。它包括一个取自单元的 BD_ADDR 代码。信道识别码用于标识匹克网信道, 并构成信道上所有交换分组的头。信道识别码取自主单元的 BD_ADDR 的 LAP。查询识别码用于查询操作。通用查询识别码为所有蓝牙单元公用, 指定查询识别码用于设备类型的查询。

识别码也可用于对接收方指出分组的到来。它可用于定时同步和偏移补偿。接收方与识别码中的整个同步字相关, 并提供一个非常健壮的信令。在信道设置过程中, 该代码本身可以作为一个 ID 分组来支持识别过程。另外, 在休眠状态下, 它还可用于随机识别过程。识别码由头, 同步字和尾构成, 下面描述同步字的生成过程。

1. 同步字

同步字基于 (64, 30), 使用覆盖 (按位 XOR 方式) 64 位全长 PN 序列的删除代码块。该删除代码保证基于不同地址上的同步字之间的大汉明空间 ($d_{\min} = 4$), PN 序列改善了识别码的自相关特性。以下各步描述如何产生同步字:

- 产生信息序列;
- 与 PN 覆盖序列的信息覆盖部分作 XOR 运算;
- 产生代码字;
- 采用 PN 覆盖序列的所有 64 位与代码字作 XOR 运算。

信息序列通过在 24 位 LAP 上附加 6 位生成 (第 1 步)。如果 LAP 的 MSB 等于 0, 那么附加位为 0011101; 如果 LAP 的 MSB 为 1, 那么附加位为 110010。LAP MSB 连同附加位一起构成了一个长度为 7 的 Barker 序列。在第 2 步中, 信息与伪随机噪声 (PN) 序列的 $P_{34} \cdots P_{63}$ 位通过 XOR 操作实现预加扰。在产生 BCH 代码字 (第 3 步) 后, 完整的 PN 序列与代码字作 XOR 运算 (第 4 步)。解扰代码字的信息部分, 同时代码字的部分位则被加扰。因此, 起始的 LAP 和 Barker 序列将保证作为识别码同步字的一部分, 而且将去掉 BCH 代码字的循环特性。原理如图 2.47 所示。

最后, 二进制序列将通过它们相应的 D -传输 (这里 D^i 在表示 i 次单元延时) 来表示。设 $P'(D) = P'_0 + P'_1D + \cdots + P'_{62}D^{62}$, D^{62} 为 63 位的伪随机序列。其中, P'_0 是 PRNG 之外的第一位 (LSB), 而 P'_{62} 是最后一位 (MSB)。为了获得 64 位, 在该序列的末尾还要附加一个额外的零 (这样, $P'(D)$ 就不会改变)。为了解释方便, 该扩展多项式的倒数 $P(D) = D^{63}P'(1/D)$ 将用于该序列。 $P(D)$ 是按照逆序排列的序列。我们使用 $a(D) = a_0 + a_1D + \cdots + a_{23}D^{23}$ (a_0 是蓝牙地址的 LSB) 表示 24 位蓝牙地址的低地址部分 (LAP)。

(64, 30) 块代码发生器符号使用 $g(D) = (1+D)g'(D)$ 表示。其中, $g'(D)$ 是初始二进制 (63, 30)

的 BCH 代码的发生器多项式 1557464165547（八进制数）。这样按照八进制符号解释，我们可以得到：

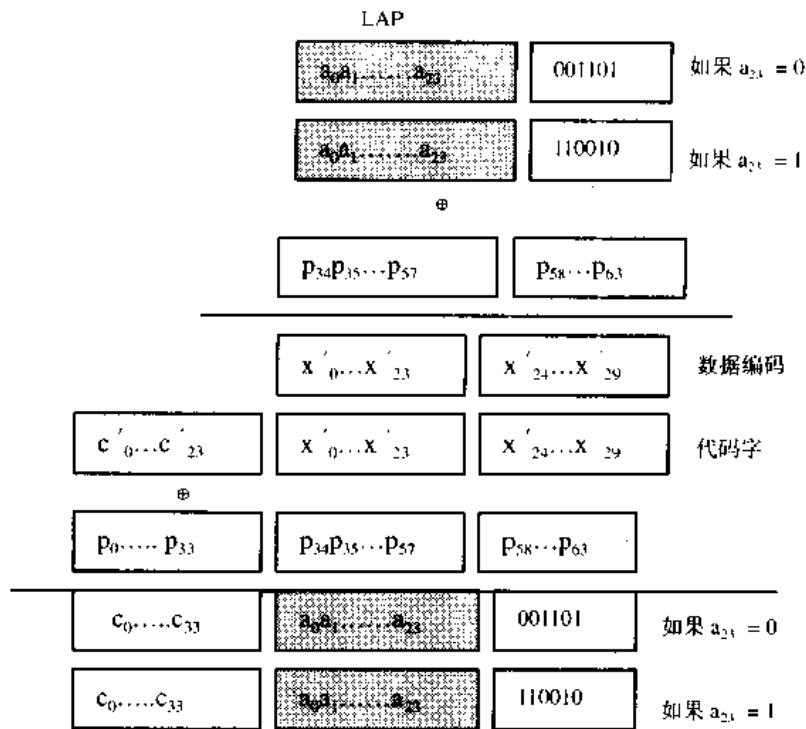


图 2.47 同步字的创建

$$g(D) = 260534236651 \quad (2-18)$$

最左位与和高序列 (g_{34})系数相对应。DC-空闲四位序列 0101 和 1010 可以分别为：

$$\begin{cases} F_0(D) = D + D^3, \\ F_1(D) = 1 + D^2, \end{cases} \quad (2-19)$$

另外：

$$\begin{cases} B_0(D) = D^2 + D^3 + D^5, \\ B_1(D) = 1 + D + D^4, \end{cases} \quad (2-20)$$

该等式用于建立一个长度为 7 的 Barker 序列。然后，识别码由以下过程产生：

- ① 初始化编码的 30 个信息位： $x(D) = \alpha(D) + D^{24} \beta a_{23}(D)$ ；
- ② 增加覆盖部分 PN 覆盖序列的信息： $x'(D) = x(D) + p_{34} + p_{35}D + \dots + p_{63}D^{29}$ ；
- ③ 产生扩展 BCH 代码的奇偶位： $c'(D) = D^{34}x'(D) \bmod g(D)$ ；
- ④ 产生 BCH 代码字： $s'(D) = D^{14}x'(D) + c'(D)$ ；
- ⑤ 增加 PN 序列： $s(D) = s'(D) + p(D)$ ；
- ⑥ 附加 (DC-空闲) 头和尾： $y(D) = Fc_0(D) + D^4 s(D) + D^{86} Fa_{23}(D)$ 。

2. 伪随机噪声序列发生器

我们使用基本多项式 $h(D) = 1 + D + D^3 + D^4 + D^6$ 生成伪随机噪声序列，LFSR 和它的初始状态如图 2.48 所示。

生成的 PN 序列（包括附加结束位“0”）为 83848D96BBCC54FC。LFSR 输出以 PN 序

列的最左位开始。这与前节的 $P'(D)$ 相对应。这样，将使用 $P(D)$ 的倒数覆盖给出的 64 位序列： $p \approx 3F2A33DD69B121C1$ 。

其中，最左位 $P_0=0$ (在 16 进制数字 3 的二进制表示中的有两个初始零)，最位 $P_{63}=1$ 。

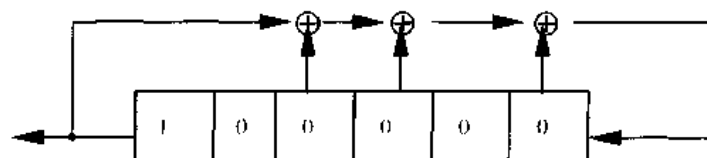


图 2.48 LFSR 及生成 $P'(D)$ 的初始状态

3. GIAC 和 DIAC 的保留地址

蓝牙查询操作保留一块 64 位连续 LAP 地址；通用查询则保留一块蓝牙设备公用 LAP 地址。其余的 63 位 LAP 地址则保留用作特定类型蓝牙设备的指定查询。可以使用同一 64 位块而不必关心 UAP 和 NAP 的内容。因此，这些 LAP 地址将不会作为用户的 BD_ADDR。

当为查询应答的 FHS 分组初始化 HEG 和 CRC 时，UAP 将由 DCI 替代。同样，无论何时如果有一个保留 BD_ADDR 用于生成一个跳频序列，UAP 都将由 DCI 替代。

保留 LAP 地址暂时为：0X9E8B00~0X9E8B3F。通用查询 LAP 暂时为：0X9E8B33。所有地址都具有在最右边位置上以十进制表示的 LSB 位。

2.13.3 活动成员地址 (AM_ADDR)

在匹克网中的每一个活动从单元都赋予一个 3 位活动成员地址 (AM_ADDR)。全零 AM_ADDR 则保留用于广播消息。主单元不具有 AM_ADDR，其定时将它与从单元区分开来。从单元只能采用匹配的 AM_ADDR 和广播消息分组接收一个分组。AM_ADDR 在分组头中携带。只有从单元在信道上处于活动状态时，AM_ADDR 才是合法的。一旦从单元脱离链路或进入休眠状态，它就将丢失其 AM_ADDR。

当从单元激活时，主单元就分配一个 AM_ADDR 给从单元。在连接建立时，AM_ADDR 置于 FHS 有效载荷中 (FHS 头携带带有全“0”AM_ADDR)。当处于唤醒状态时，AM_ADDR 置于唤醒消息中。

2.13.4 休眠成员地址 (PM_ADDR)

处于休眠模式的从单元能够通过其 BD_ADDR 或通过专用休眠成员地址 (PM_ADDR) 识别。后者地址是一个区分休眠从单元的八位成员地址。只有当从单元处于休眠状态时，PM_ADDR 才有效。当从单元激活时，它将被分配一个 AM_ADDR 地址，但将丢失 PM_ADDR。PM_ADDR 在休眠时分配给从单元。

2.13.5 访问请求地址 (AR_ADDR)

访问请求地址用于休眠从单元在访问窗口内确定从一主半时隙。在该半时隙中它可以允许发送访问请求消息。当从单元进入休眠模式时，AR_ADDR 分配给从单元。而且，只有从单元处于休眠状态，AR_ADDR 才有效。同时 AR_ADDR 并不必是惟一的，即不同休眠从单元可以共享同一个 AR_ADDR。

2.14 蓝牙安全性

蓝牙技术提供短距离的对等通信。为了提供使用保护和信息保密，系统必须在应用层和链路层上提供安全措施。该措施将适用于对等环境。也就是说，各蓝牙单元中的措施、鉴权和加密规则将以同样方法实现。链路层中存在四种不同的实体用来保证安全。每个用户具有一个惟一的公共地址、两个加密字和每次新事务处理中都不同的随机数。这四个实体及其长度如表 2.19 所示。

表 2.19 用于鉴权和加密的实体

实 体	长 度
BD_ADDR	48 位
私有用户字，鉴权	
私有用户字，加密配置长度（字节-方式）	8~128 位
随机数	128 位

蓝牙设备地址（BD_ADDR）是一个对每个蓝牙单元惟一的 48 位 IEEE 地址。蓝牙地址是公开的而且可经 MMI 交换，也可自动通过蓝牙单元查询规则获得。

加密字取自于初始化期间，且不会被公开。通常，加密字取自鉴权过程中的鉴权字。对于鉴权算法，该鉴权字一般为 128 位。加密字长度可在八进制数 1~16 间变化（8~128 位）。加密字长度可以两种原因进行配置。第一种原因是，不同国家具有很多不同的适用于加密算法的要求，一般情况下，取决于出口规定和官方对于加密的态度。第二个原因是，没有必要对算法及加密硬件进行重新设计。增加有效字长度应符合当前计算能力的发展状况。当前所给出的 64 位加密字长度足以满足大部分用户的保密要求。

加密字完全不同于鉴权字。每次加密的激活过程都将产生新的加密字。因此加密字的生命周期不必与鉴权字一致。鉴权字从本质上讲，与加密字相比具有静态性。一旦建立加密字，那么就将由运行在蓝牙设备上的具体应用决定在何时和是否需要改变加密字。鉴权字对于具体蓝牙链路也非常重要性，它常被当作链路字进行引用。

RAND 是一个取自蓝牙单元中的一个随机数或伪随机过程的随机数，它并不是一种静态参数，而是经常会改变。

2.14.1 随机数发生器

每个蓝牙单元都具有一个随机数发生器。随机数在安全功能方面有很多用途。例如竞争应答方案、鉴权和加密字生成过程等。理想情况下，将使用一个真正的基于使用非连续随机物理处理过程的随机数发生器。举例来讲，如来自于元器件、电阻及各种振荡器的不稳定性所产生的热噪声。由于实用的原因，一般使用基于随机数发生器的软件解决方案。总的来说，很难界定一个伪随机序列的随机过程。在蓝牙中，使用随机数的要求是它们必须非重复和随机生成。

非重复的意思是，在鉴权字生命期内，该值不得重复。例如：计数器的输出值可以是非重复值，因为它在鉴权字的整个生命周期内不会重复，另一个例子是时间戳。随机生成意思是，它不得以大于“0”的概率进行预测（如大于 $1/2^L$ ，其中 L 为字长度）。

显然，LM 可使用该发生器于各种目的，如 RAND、单元字、 K_{init} 、 K_{master} 及补偿或等待时间等。

2.14.2 字管理

在特定单元里，加密字的长度不能由用户设定，它必须由生产厂商预置。为了防止用户超出允许字长度，蓝牙基带处理不得接收由高层软件提供的加密字。无论何时需要新加密字，它都必须以加密算法来产生 E-3 字。

链接字的改变也应通过基带过程来完成。根据链接字的类型，可以采用不同方式。

1. 字类型

链接字是 128 位的随机数。它由两方或两方以上共享，而且它还是各方之间安全事务处理的基础。链接字本身用于鉴权规则。当导出加密字时，链接字也可作为其参数之一。

会话 (session) 定义为某单元作为一特定匹克网成员时所处的时间间隔。因此，当单元脱离匹克网链路时，会话将终止。

链接字可以是半永久性的或是临时的。半永久性链接字存放在随机存储器里，而且在当前会话终止后仍可使用。而且，一旦半永久性链接字被定义，它就可用于共享它的蓝牙单元之间的后续连接的鉴权过程中。临时链接字的生命周期由当前会话的生命周期作为限制。它不能在以后的会话中重新使用。在一对多点配置中，一条相同的信息可以安全地分发到多个接收方。而且，在一对多点配置中，可以采用公用加密字。为达到该目的，可以使用指定链接字（由主单元字表示）临时替换当前链接字。

最后，我们有时也会用到当前链接字。当前链接字是指一个用于当前时间的正在使用的链接字。它可以是半永久性的或临时性的。因此，当前链接字可用于当前连接中的所有鉴权过程和所有加密字生成过程。

为了适应不同类型的应用要求，链接字的四种类型定义如下：

- 组合字 K_{AB} ；
- 单元字 K_A ；
- 临时字 K_{master} ；
- 初始化字 K_{init} 。

除了这些加密字类型以外，还有一种加密字 K_C 。该字取自当前链接字。一旦加密由 LM 命令激活，则自动实现该加密字的修改。区分鉴权字和加密字的目的在于方便在不降低鉴权过程作用的情况下使用更短的加密字。

对一个蓝牙单元来说，组合字 K_{AB} 和单元字 K_A 从功能上不可区分。其不同在于它们生成的方法不同。单元字 K_A 生成取决于单元 A。单元字在蓝牙单元安装时生成，且此后几乎不变。组合字取自单元 A 和单元 B 中的信息，而且它也取决于这两个单元。组合字对于每两个蓝牙单元的每个新组合都不同。

单元字和组合字的使用取决于应用和设备。具有很少存储空间存储加密字的蓝牙单元，或者安装到可由大批用户访问设备中的蓝牙单元，将更倾向于采用它们自己的单元字。在这种情况下，它们只须把单个字存起来。而要求较高安全性的应用则倾向于采用组合字。由于要存储对应于不同蓝牙单元链路的组合字，这些应用将需要更多的存储空间。

主单元字 K_{master} 是只用于当前会话的链接字。它将只是临时性替换原链接字。例如当主单元希望能够使用相同加密字同时访问两个以上蓝牙单元时，就可使用该加密字。

当没有定义或交换组合字或单元字，或没有丢失链接字时，可使用初始化字 K_{init} 。初始

化字可用于保证初始化参数的传输。该字取自于一个随机数、一个八位字节 PIN 码，和请求单元的 BD_ADDR。该字只能用在初始化过程中。

PIN 可以是一个由蓝牙单元（如在 PSTN 插件里没有 MMI 时）提供的固定值。PIN 可由用户任意选择，还可由进入两个必须匹配的单元。当两个单元都具有 MMI 时（如电话或便携式计算机）可使用后一种过程。在两个单元中各输入 PIN 的方式比在任一单元中使用固定 PIN 更安全。只要可能，应尽可能使用第一种方式。即使使用固定 PIN，也应可改变该 PIN。这主要是防止曾经已具有 PIN 用户的再次初始化问题。如果 PIN 是无效的，将使用默认值“0”。

对于很多应用，PIN 代码将是一串相当短的数字。一般，它由四位十进制数组成。尽管很多情况下它可充分满足安全需要，但是还是存在无数更敏感的其他情况，对于这些情况，PIN 并不足够可靠。因此，PIN 长度可在 1~16 位（八进制）之间进行选择。对于更长的长度，我们认为各单元与其是通过机械（人工）交换 PIN，不如说是通过应用层软件来进行支撑。例如：它可能是 Diffie-Hellman 字协议。此处，字的互换在双方单元中都需经过 K_{init} 的生成过程，就如在较短的 PIN 代码的情况里一样。

2. 字生成和初始化

链接字需在各蓝牙单元中生成及分布，以便用于鉴权过程。由于链接字必须是安全的，所以它不能象蓝牙地址那样通过查询规则来获得。字生成在初始化过程中实现。初始化过程也就是分别在每两个正要实现鉴权和加密的单元上所执行的过程。整个初始化过程的实现由下面五步组成：

- 生成初始化字；
- 鉴权；
- 生成链接字；
- 交换链接字；
- 在各个单元中生成加密字。

初始化过程后，各单元可以进行通信，也可以与链路断开连接。若执行加密，应选取自当前链接字的合适加密字用于 E_0 算法。对于任一在单元 A 和单元 B 之间建立的新连接，它们将使用用于鉴权的公共链接字，以免再一次从 PIN 推导 K_{init} 。新的取自指定链接字的加密字将再下次加密过程激活时生成。

若无可链接字，LM 将在初始化过程中自动启动。

2.14.3 加密

用户信息可采用分组有效载荷的加密进行保护，但识别码和分组头不加密。有效载荷的加密采用 E_0 流密码实现。 E_0 将对每一有效载荷重新同步，其原理如图 2.49 所示。

流密码系统 E_0 由三部分组成。第一部分执行初始化（生成有效载荷字），第二部分生成字流位，第三部分完成加密和解密。有效载荷字发生器非常简单，它仅仅以适当序列对输入位进行组合，然后将它们转移到用于字流发生器的四位 FLSR 中。第二部分是该密码系统的主要部分，并也将用于初始化过程中。字流位通过取自于 Massey 和 Rueppel 流密码发生器的方法来生成。该方法与现有已知方法相比它具有加密分析方面的明显优点。尽管求和发生器仍具有关联攻击方面的弱点，但是频繁的重新同步将抑制这类攻击。

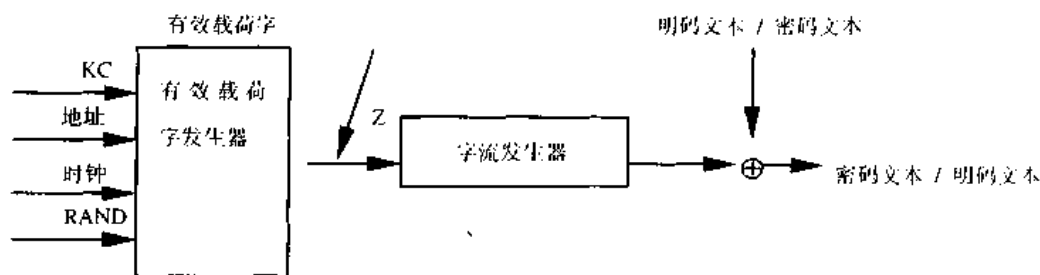


图 2.49 蓝牙的 E₀ 流加密

1. 加密字长度协商

执行基带规范各蓝牙设备需要一个定义最大字允许长度 $L_{\max}$ 的参数 ($1 \leq L_{\max} \leq 16$) (八进制数)。对于各种应用, 参数 $L_{\min}$ 定义为某一特定应用的最小可接受字长度。在加密字生成前, 有关单元必须协商决定实际可用的字的长度。

主单元将发送一个建议值 $L_{\text{sug}}^{(M)}$ 到从单元。最初建议值设成 $L_{\max}^{(M)}$ 。如果 $L_{\min}^{(S)} \leq L_{\text{sug}}^{(M)}$, 而且从单元支持该建议值长度, 则从单元确认且该值将成为该链路的加密字。但是, 如果两个条件都不满足, 从单元将发送一个新的建议值 $L_{\text{sug}}^{(S)} < L_{\text{sug}}^{(M)}$ 到主单元。该值将是所有可支持长度中的最大值, 但小于先前主单元的建议值。主单元按照从单元建议完成相应测试。该过程一直持续到字长度达到一致为止, 或一个单元放弃协商。放弃操作可由缺少 L_{sug} 支持和字长度太小, 或者在某个单元中 $L_{\text{sug}} < L_{\min}$ 等原因所引起。异常中止情况下, 不得使用蓝牙链接加密字。

建立安全链路可能失败, 是让应用决定是否接受或拒绝建议字长度所带来的不可避免的结果。然而这是一种必要的预防措施。否则一个非法用户单元将可以通过声明一个小的最大字长度进行非法侵入。

2. 加密模式

如果从单元具有半永久链接字 (组合字或单元字), 它将只能在自身的 (当然在主单元的相反方向) 的个别地址时隙上接受加密字。特别是假设广播消息没有加密的话, 可行通信模式如表 2.20 所示。

表 2.20 使用半永久链接字从单元的可行通信模式

广播通信	个别地址通信
无加密	无加密
无加密	加密, 半永久链接字

表 2.21 主单元的可行通信模式

广播通信	个别地址通信
无加密	无加密
无加密	加密, K_{master}
加密, K_{master}	加密, K_{enc}

当表中入口涉及到一链接字时, 它是指加密 / 解密引擎将使用从该链接字得到的加密字。如果从单元接收一主单元字, 就有如表 2.21 所示的三种可能的组合。

在这种情况下, 匹克网中的所有单元都使用公共链接字 K_{master} 。由于主单元为了保证匹克网上整个通信的安全而使用链接字中导出的加密字, 因此也就避免了使用加密字的从单元中的多义性。在这种情况下, 其缺省方式是广播消息不加密。对于广播和个别地址通信, 可使用指定 LM 命令激活加密。主单元能够发送 LM 命令到从单元, 并通知它们回到前半永久链接字。不管它们以前处于什么模式, 都将以非加密广播通信作为结束。

3. 加密概念

对于加密规则，流密码算法用于将加密位按位模 2 并加到数据流上，然后通过无线接口进行传输。有效载荷在附加 CRC 位之后加密，但在 FEC 编码之前。

各分组单独加密。密码算法 E_0 使用随机数 EN_RAND_A 、主单元蓝牙地址、主单元实时时钟 ($CLK_{26,1}$) 的 26 位和加密字 K_c 作为输入，如图 2.50 所示。

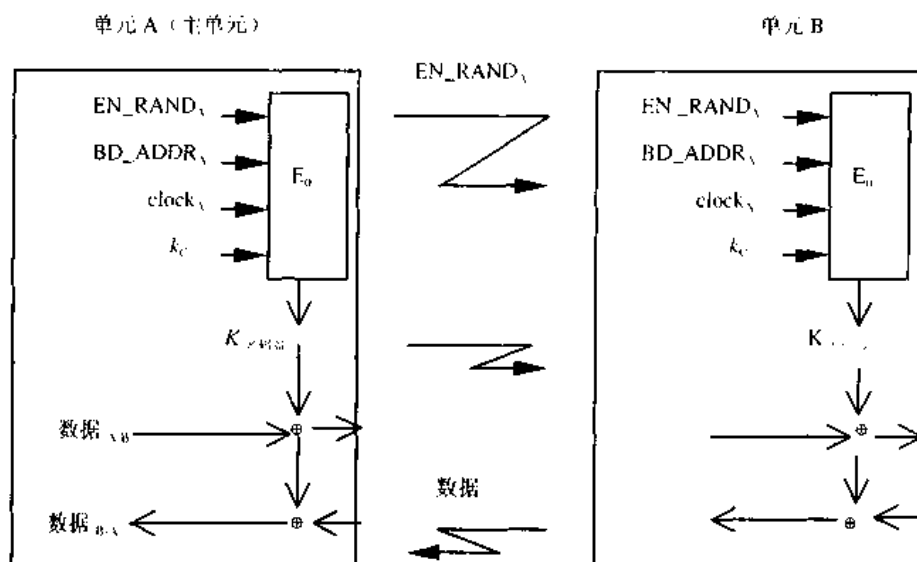


图 2.50 编码过程

K_c 由当前链接字、COF、一个随机数、 EN_RAND_A 推导得到。 K_c 在进入加密模式之前由主单元发布。注意： EN_RAND_A 是公开的，因为在无线模式中它是一个明码文本。

在 E_0 算法中，加密字 K_c 修改为另一加密字 K_c' 。该字的最大有效长度由厂商预置，并可设置为 8 的整数倍 (8~128 位)。

实时时钟在每一时隙上都将增长。 E_0 算法将在每一新分组的开始重新初始化 (即：主-从与从-主传输一致)。通过使用 $CLK_{26,1}$ ，在两次传输之间将至少改变一位。每次重新初始化后，将生成新的字流。对于覆盖不止一个时隙的分组，其第一时隙中的蓝牙时钟将用于整个分组。

加密算法 E_0 生成二进制字流 K_{cipher} 。该字流是模 2 运算并加到加密数据上的结果。密码是对称的。解密使用完全和加密相同的字和相同的方法实现。

4. 加密算法

线性反馈移位寄存器 ($LFSR_5$) 的系统输出是一个 16 状态的简单有限状态机 (称作求和合成器) 的组合。该状态机的输出为字流序列，或是初始化状态中的随机初始值。算法由加密字 K_c 、48 位蓝牙地址、主单元时钟位 $CLK_{26,1}$ 和 128 位 RAND 值表示，其设置过程如图 2.51 所示。

具有 $L_1=25$ ， $L_2=31$ ， $L_3=33$ 和 $L_4=39$ 长度的四个 $LFSR_5$ ($LFSR_1, \dots, LFSR_4$)，用如表 2.22 所示的指定反馈多项式生成。

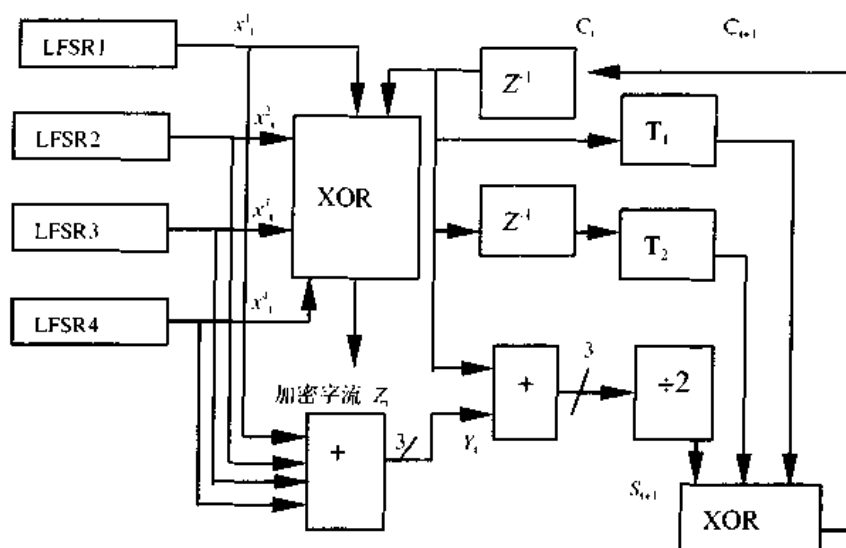


图 2.51 加密引擎概念

表 2.22 四个原语反馈多项式

i	L_i	反馈 $f_i(t)$	权
1	25	$t^{25} + t^{20} + t^{12} + t^8 + 1$	5
2	31	$t^{31} + t^{24} + t^{16} + t^{12} + 1$	5
3	33	$t^{33} + t^{26} + t^{24} + t^7 + 1$	5
4	39	$t^{39} + t^{36} + t^{28} + t^7 + 1$	5

寄存器组总长度是 128 位。这些多项式全是原语式的。整个反馈多项式的汉明权重都为 5，其理由是便于硬件实现，需减少 XOR 门的数量及为获得生成序列好的统计特性之间的折衷。

设让 X^i_i 表示 LFSR i 的符号，从四元组 $X^1_i, \dots, X^4_i$ ，可以导出值 y_i ：

$$y_i = \sum_{i=1}^4 X^i_i \quad (2-21)$$

其中该和大于所有这些整数。于是 y_i 可取值 0, 1, 2, 3 或 4。加法生成器的输出通过下列等式给出。

$$z_i = X^1_i \oplus X^2_i \oplus X^3_i \oplus X^4_i \oplus C^0_i \in \{0, 1\}, \quad (2-22)$$

$$s_{i+1} = (s^1_{i+1}, s^0_{i+1}) = (y_i + C_i) / 2 \in \{0, 1, 2, 3\}, \quad (2-23)$$

$$c_{i+1} = (c^1_{i+1}, c^0_{i+1}) = s_{i+1} \oplus T_1[c_i] \oplus T_2[c_{i-1}], \quad (2-24)$$

其中 $T_1[\cdot]$ 和 $T_2[\cdot]$ 是在 $GF(4)$ 上的两个不同的线性映射。假设 $GF(4)$ 是通过不能简化的多项式 x^2+x+1 ，而且让在 $GF(4)$ 里多项式的 α 为“0”时产生。映射 T_1 和 T_2 如下定义：

$$T_1: GF(4) \rightarrow GF(4)$$

$$x \mapsto x$$

$$T_2: GF(4) \rightarrow GF(4)$$

$$x \mapsto (\alpha + 1)x$$

我们可以用二进制矢量写 GF(4) 的元素，如表 2.23 所示。

$$T_1: (x_p, x_0) \rightarrow (x_p, x_0),$$

$$T_2: (x_p, x_0) \mapsto (x_p, x_0 \oplus x_0).$$

表 2.23 T_1 和 T_2 映射

X	$T_1[x]$	$T_2[x]$
00	00	00
01	01	11
10	10	01
11	11	10

2.14.4 鉴权

蓝牙中的实体鉴权采用竞争-应答方案。在该方案中，申请者对密钥字的确认使用对称密钥字经 2-MOV 协议进行校验。对称密钥指当前申请者/校验器共享同一密钥字，比如说 K。在竞争-应答方案里，校验器将竞争申请者鉴权随机数输入，该输入以含一鉴权码的 AU_RAND_A 标注，而该鉴权码则以 E1 标注，同时向校验器返回结果 SRES，如图 2.54 所示。图中说明了输入到 E1 的值由 AU_RAND_A 和申请者的蓝牙设备地址 (BD_ADDR) 组成。该地址防止简单反射攻击。由单元 A 和单元 B 共享的密钥 K 是当前链接字。

由于整个服务要求基于 FIFO，所以反射攻击实际上对蓝牙没形成危害。当抢先占有导出时，该攻击才具有潜在危害。

用于蓝牙里的对称字竞争-应答方案如图 2.52 所示。

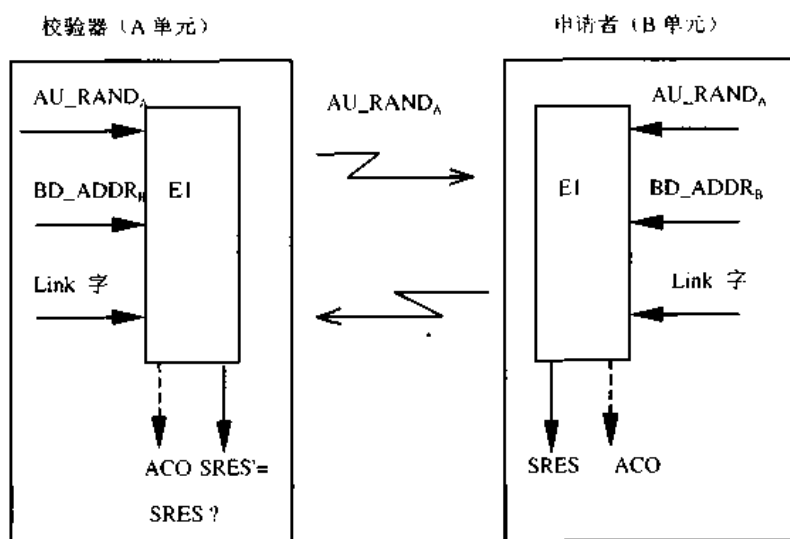


图 2.52 蓝牙竞争应答

在蓝牙中，校验器不必就是主单元。应用将指出谁将被谁鉴权。某些应用只要求进行单向鉴权。然而，在某些点对点的通信中，人们宁可相互鉴权，原因是在两个鉴权过程中，各单元将可顺序成为竞争方（校验器）。LM 通过申请者确定要进行哪个方向鉴权的应用协调给出的鉴权选择。对于共享单元相互鉴权，在单元 A 已成功鉴权单元 B 后，单元 B 通过发送 AU_RAND_B （不同于单元 A 发布的 AU_RAND_A ）到单元 A 来鉴权单元 A，并从新的

AU_RAND_B 中导出 SRES 和 SRES'、单元 A 地址和链接字 K_{AB}。
如果鉴权成功，则通过 E_i 过程产生的 ACO 值将被保留。

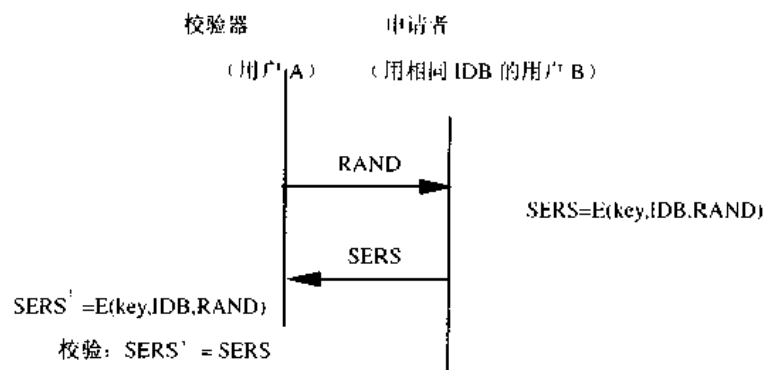


图 2.53 对称字竞争-应答方案

第3章 链路管理器协议

3.1 概述

本章主要描述链路管理器协议（LMP），该协议用来对链路进行设置和控制。收方链路管理器通过该协议对接收到的信号进行识别和筛选，而不再将接收的信号转发到更高的协议层。链路管理器全局视图如图 3.1 所示。LMP 协议用于链路的建立、链路的加密和控制。该协议可直接发送有效载荷而不用 L2CAP 方式来发送，同时通过有效载荷头的 L_CH 字段保留值来区别不同发送方式。该消息直接由接收方的 LM 过滤和解释，同样也不将它转发到更高的协议层。

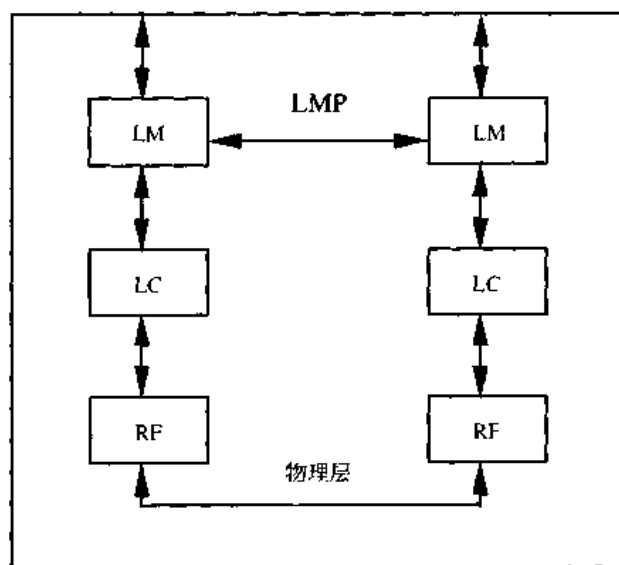


图 3.1 链路管理器全局视图

链路管理器消息发送比用户数据发送具有更高的优先级。也就是说，虽然链路管理器消息发送会被个别基带中的数据分组发送延迟，但是它不能被 L2CAP 的通信延迟。

由于 LC 对消息发送提供了可靠的连接，在此我们不必对 LMP 消息本身的结构作深入了解。

根据发送过程规定，在接收端收到带有 LMP PDU 基带数据分组与带有合法应答 PDU 基带发送数据分组之间的时间间隔不能大于 LMP 的最大应答延迟时间，最大应答延迟时间为 30 秒。

3.2 链路管理器协议格式（LMP）

LMP PDU 总是以单时隙分组的方式发送，因此有效载荷头只占一个字节。有效载荷头的两个最低位用来确定逻辑信道。对于 LMP PDU，这些位设置如表 3.1 所示。

表 3.1 逻辑信道 L_CH 域内容

L_CH 代码	逻辑信道	信 息
00	NA	未定义
01	UA/I	继续发送 L2CAP 消息
10	UA/I	开始发送 L2CAP 消息
11	LM	LMP 消息

一般情况下有效载荷头中的 FLOW 只有一位，并且该 FLOW 位可以被接收方忽略。每个 PDU 都分配了一个 7 位操作码，它用来标识不同类型的 PDU。操作码和只占有一位数据的事件 ID 共同设置成有效载荷的首字节，如图 3.2 所示。事件 ID 位于该字节的最低位。如果 PDU 属于由主单元发起的事件，则事件 ID 为 0；如果 PDU 属于由从单元发起的事件，则事件 ID 为 1。如果在 PDU 分组中含一个或多个参数，则这些参数都位于有效载荷的第二个字节中。字节数根据参数的长短来确定。假设有一条使用 HV1 数据分组的 SCO 链路，且数据内容长度不足 9 个字节，那么 PDU 可以用 DV 数据分组的格式来发送，否则必须使用 AM1 分组格式发送。所有的参数都使用小端格式，即最低位字节先发送。

协议数据单元的源地址和目的地址由消息头的 AM_ADDR 决定。

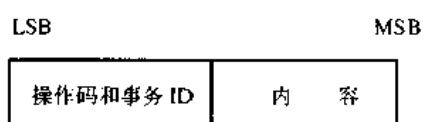


图 3.2 LMP PDU 被发送时的有效载荷

每个 PDU 可以被设置成必选或可选的，这要视使用情况而定。必选或可选项在下节表中列出。在发送过程中，LM 不能发送可选 PDU。如果发送过程需要应答，则按下节过程规定发送一个有效应答，LM 必须能识别所有收到的可选 PDU。如果不需要应答收到的可选 PDU，则不发送应答消息。

3.3 过程规则与 PDU

每一过程都以序列图形式进行描述，图 3.3 所示的符号用于序列图。

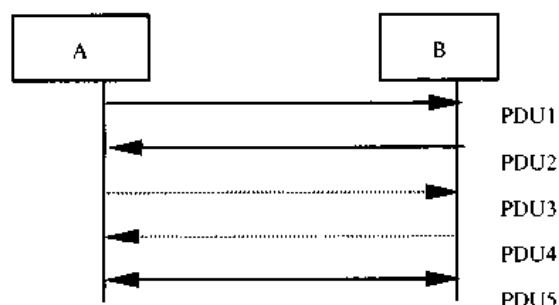


图 3.3 用于序列图的符号

图中，PDU 1 是由 A 传送到 B 的协议数据单元；PDU 2 是由 B 传送到 A 的协议数据单元；PDU 3 是从 A 传送到 B 的可选协议数据单元；PDU 4 是从 B 传送到 A 的可选协议数据

单元；PDU 5 是从 A 或 B 发出的协议数据单元；垂直线表示可选择发送更多的协议数据单元。

3.3.1 通用应答消息

LMP_accepted 与 LMP_not_accepted 协议数据单元在不同过程中用作其他 PDU 的应答消息。LMP_accepted 协议数据单元分组含有接收消息的操作码，LMP_not_accepted 协议数据单元分组含有未接收消息的操作码以及未接收该消息的原因，如表 3.2 所示。

表 3.2 通用应答消息

M/O	PDU	内容
M	LMP_accepted	操作码
M	LMP_not_accepted	操作码 原因

3.3.2 鉴权

鉴权过程基于竞争应答模式。校验器发送一个 LMP_au_rand PDU 分组给请求者，该 PDU 分组含有一个随机数（或称竞争码）。请求者根据获取的分组计算出应答值，该应答值是竞争码、请求者 BD_ADDR 和保密字三者的函数，然后将应答发回给校验器验证应答值是否正确。鉴权应答值的正确计算需要两个设备共享同一密钥。主单元和从单元都可以作为校验器，表 3.3 所示的协议数据单元可用作鉴权的过程。

表 3.3 用于鉴权的 PDU

M/O	协议数据单元	内容
M	LMP_au_rand	随机数
M	LMP_sres	鉴权应答

1. 请求者具有链接字

如果请求者具有与校验器相关联的链接字，则请求者将计算出的应答值并连带 LMP_sres 发送到校验器，由校验器检查其应答值。如果应答值不正确，校验器则发送附加原因码“鉴权失败”（authentication failure）的 LMP_detach 来终止连接，如图 3.4 所示。

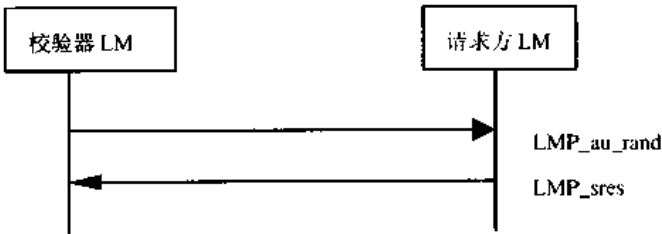


图 3.4 鉴权、请求者具有链接字

2. 请求者无链接字

如果请求者没有与校验器相关联的链接字，在收到 LMP_au_rand 后，请求者则发送附加原因码“字丢失”(key missing)的 LMP_not_accepted 消息，如图 3.5 所示。

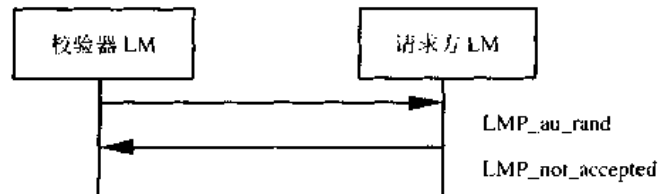


图 3.5 鉴权失败、请求者无链接字

验证失败后，将实施基带所述的一些保护方案，以防入侵者在短时间尝试多个字。

3.3.3 匹配

当两台设备无共用链接字时，则基于 PIN 和随机数创建初始化字 K_{init} 。创建 K_{init} 字在校验器向请求者发出 LMP_in_rand 的时候创建，然后进行鉴权，其计算过程基于 K_{init} 字，而不是链接字。通过鉴权后，链接字即被创建。用于匹配过程的 PDU 如表 3.4 所示。

表 3.4 用于匹配过程的 PDU

M/O	协议数据单元	内容
M	LMP_in_rand	随机数
M	LMP_au_rand	随机数
M	LMP_sres	鉴权应答值
M	LMP_comb_key	随机数
M	LMP_unit_key	键

1. 请求者接受匹配

在校验器发出 LMP_in_rand 后，请求者用 LMP_accepted 应答，如图 3.6 所示。两设备计算出 K_{init} 字，然后基于此字进行鉴权。在校验器检查鉴权应答值后，如正确，就创建链接字，否则，校验器则发送附加原因码“鉴权失败”的 LMP_detach 消息终止连接。

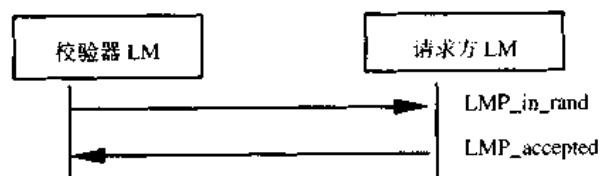


图 3.6 请求者接受匹配

2. 请求者请求成为校验器

如果请求者有一固定的 PIN，它可以通过生成一个随机数用以请求进行“请求者-校验器”的角色切换，并在 LMP_in_rand 中发回该随机数。如果启动匹配过程的设备具有可变 PIN，

则它必须接受这个可变 PIN 并用 LMP_accepted 消息应答，这样角色就成功地进行了切换，如图 3.7 所示。

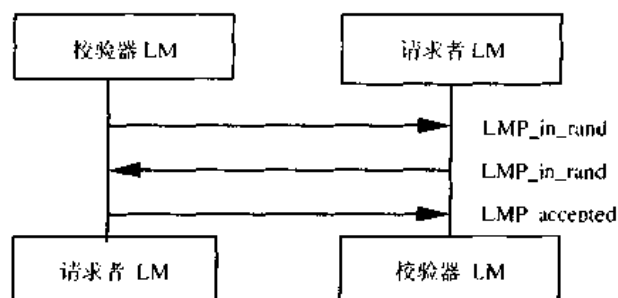


图 3.7 请求者接受匹配并请求成为校验器

如果启动匹配过程的设备具有固定的 PIN，而另一设备请求进行角色切换，则可通过发送的附加原因码匹配不允许（pairing not allowed）LMP_not_accepted 拒绝进行切换，然后终止该匹配过程，如图 3.8 所示。

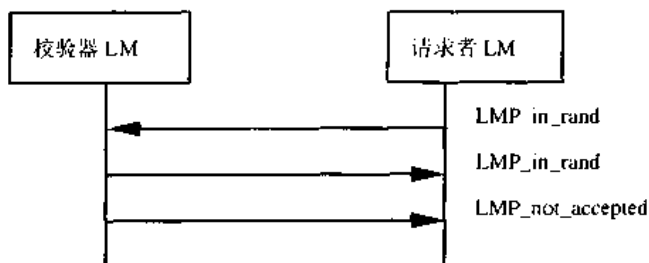


图 3.8 不成功的“请求者—校验器”角色切换过程

3. 请求者拒绝匹配

如果请求者拒绝匹配，在收到 LMP_in_rand 后发出附加原因码“不允许”（not allowed）的 LMP_not_accepted 消息，如图 3.9 所示。

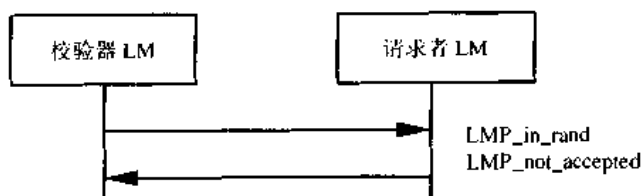


图 3.9 请求者拒绝匹配

4. 创建链接字

鉴权结束后，必须进行链接字的创建过程，如图 3.10 所示。该链接字用于两设备间的所有后续连接的鉴权，直到该链接字改变为止。匹配过程中创建的链接字可以是组合字，也可以是一个单元的单元字。以下规则用于链接字的选择。

- 如果一个单元发送 LMP_unit_key，另一个单元发送 LMP_comb_key，那么该单元字即为链接字；

- 如果两个单元都发送 LMP_unit_key，那么主单元字即为链接字；
- 如果两个单元都发送 LMP_comb_key，链接字将按基带规范所述过程进行计算。

LMP_unit_key 的内容是单元字与 K_{init} 进行 XOR 操作的结果值。LMP_comb_key 的内容是 LK_RAND 与 K_{init} 进行 XOR 操作的结果值。任何配置为使用组合字的设备都将该链接字存储在固定存储器中。

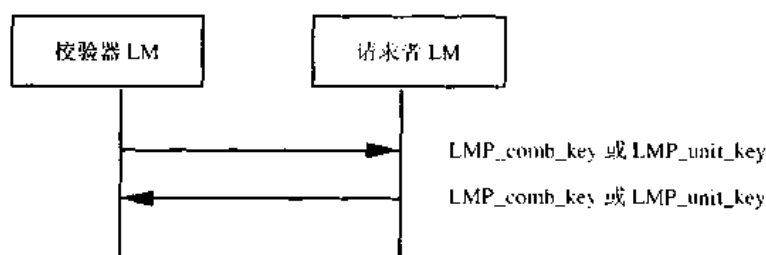


图 3.10 创建链接字

由鉴权应答出错导致处于匹配过程中的鉴权失败时，则执行重试方案。这样，可防止入侵者在短时间内使用大量不同 PIN 来达到侵入企图。

3.3.4 改变链接字

如果两设备匹配，且链接字来自于组合字，那么就可以改变链接字。如果链接字就是单元字，则在完成匹配过程后才能改变链接字。而 PDU 内容可以通过与当前链接字进行 XOR 运算得到保护。用于改变链接字的 PDU 如表 3.5 所示，链接字改变成功或失败分别如图 3.11 和图 3.12 所示。

表 3.5 用于改变链接字的 PDU

M/O	协议数据单元	内容
M	LMP_comb_key	随机数
M	LMP_unit_key	键

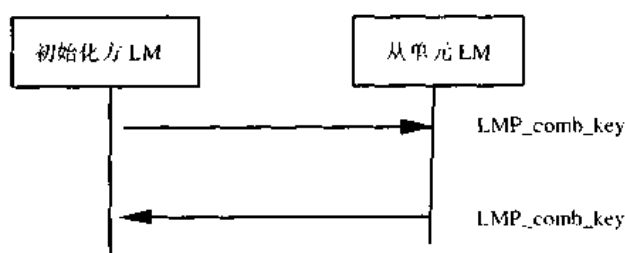


图 3.11 改变链接字

若一个单元使用单元字且不能进行链接字改变时，此时又需要成功改变链接字，那么就将新的链接字存储在固定存储器中，而旧链接字被删除，该过程将持续到链接字再次改变为止，这样新的链接字就将作为链接字用于两设备间的连接。在新链接字成为当前链接字后，新链接字一直作为当前链接字被使用，直到再次发生改变，或者创建临时链接字为止。

如果此时正在对该链路进行加密，且当前链接字为临时链接字，则可以调用改变链接字过程立即终止加密，此后，可重新启动加密过程。这样就可保证当半永久性链接字成为

当前链接字时，不会使匹克网中其他设备侦听到加密过程的加密参数。

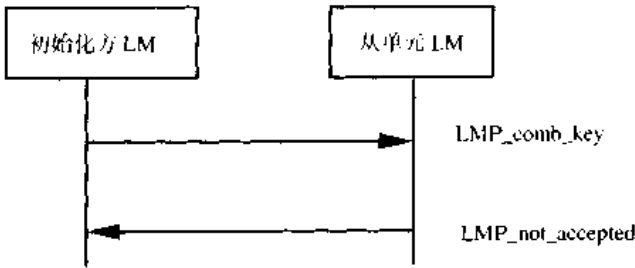


图 3.12 链接字改变失败

3.3.5 改变当前链接字

当前链接字可以是半永久性链接字或是临时链接字。如果临时改变当前链接字，其改变过程只对本次会话有效。如果匹克网支持加密广播，则必须将当前链接字改变为临时链接字。用于改变当前链接字的 PDU 如表 3.6 所示。

表 3.6 用于改变当前链接字的 PDU

M/O	协议数据单元	内容
M	LMP_temp_rand	随机数
M	LMP_temp_key	键
M	LMP_use_semi_permanent_key	

1. 改变为临时链接字

通过用创建主单元的初始字 K_{master} 启动主单元后，主单元发出一随机码 RAND，并将它的分组含在 LMP_temp_rand 中发往从单元，然后双方进行叠加运算得到 $OVL = E_{22}(\text{当前链接字}, RAND, 16)$ 后，主机再将附加的 OVL 以 2 为模的 K_{master} 在 LMP_temp_key 中发往从单元，识别 OVL 的从单元则计算 K_{master} 。这样 K_{master} 成为当前链接字，该链接字将一直保持到创建新的临时链接字或改变该链接字，如图 3.13 所示。

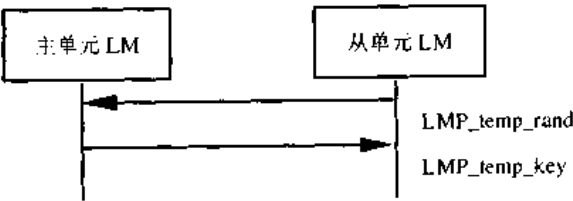


图 3.13 改变为临时链接字

2. 将半永久性链接字设置为当前链接字

在当前链接字变为 K_{master} 后，可以撤销该改变，并将半永久性链接字恢复为当前链接字，如图 3.14 所示。如果链路已加密，则可通过调用恢复半永久性字的过程立即停止加密，在此之后可重新启动加密过程。这样可保证当半永久性链接字成为当前链接字后，不会使匹克网中其他设备侦听到加密过程的加密参数。

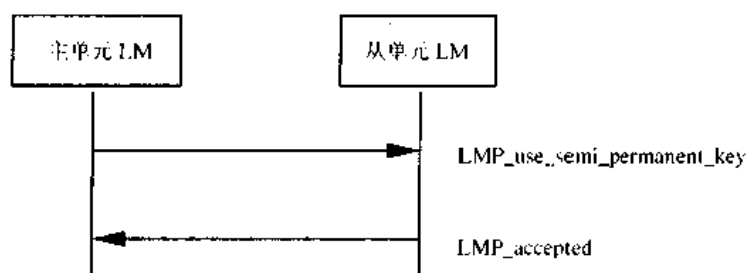


图 3.14 链接字改变为半永久性链接字

3.3.6 加密

只有在执行了至少一次的鉴权过程后，才可以进行加密。如果主单元要求匹克网中所有从单元使用相同的加密参数，那么在开始加密之前，主机必须发送临时字 K_{master} ，并将该字设置成匹克网中所有从单元的当前链接字。如果要对广播数据分组进行加密，则必须执行该过程。用于加密的 PDU 如表 3.7 所示。

表 3.7 用于加密的 PDU

M/O	协议数据单元	内容
O	LMP_encryption_mode_req	加密模式
O	LMP_encryption_key_size_req	字长度
O	LMP_start_encryption_req	随机码
O	LMP_stop_encryption_req	

1. 协商加密模式

首先，主、从单元对于是否使用加密、点对点数据分组加密或点对点与广播数据分组加密等方面必须保持一致。如果主、从单元一致采用加密模式，主单元将继续发出更多与加密有关的信息。主、从单元协商加密模式过程如图 3.15 所示。

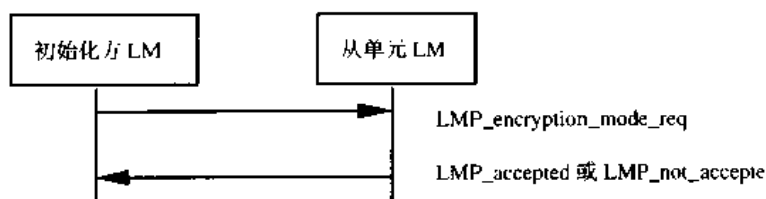


图 3.15 协商加密模式

2. 加密字长度

在确定加密字长度的过程中，应使用与基带部分相同的术语。主单元发出含有推荐值为 $L_{\text{sug}}^{(m)}$ 字长度的 LMP_encryption_key_size_req 分组，该字与 $L_{\text{sug}}^{(m)}$ 相等。如果 $L_{\text{min}}^{(s)} \leq L_{\text{sug}}^{(m)}$ ，并且从单元支持 $L_{\text{sug}}^{(m)}$ ，那么它将返回 LMP_accepted，字的长度则设置为 $L_{\text{sug}}^{(m)}$ 。如果两条件都不能满足，从单元将返回分组含有推荐值为 $L_{\text{sug}}^{(s)}$ 的长度 LMP_encryption_key_size_req 消息。该值是从单元所支持的最大的字长度，但它比 $L_{\text{sug}}^{(s)}$ 小。然后主单元将按照从单元的推荐值执

行相应测试。此过程将重复执行，直到主、从双方确定为一致的字长度，或明确知道不能确定字长度时为止。一旦确定了一致的字长度，任一单元最后将发送 `LMP_accepted` 以及字的长度，如图 3.16 所示。

接下来，将使用 `LMP_encryption_key_size_req`，然后开始加密。如果没有确定一致的字长度，任一单元都可以发送 `LMP_not_accepted`，以及该 `LMP_not_accepted` 附加原因码 `Unsupported parameter value`（不支持的参数值），这时主从单元将不能使用蓝牙链路加密进行通信，如图 3.17 所示。

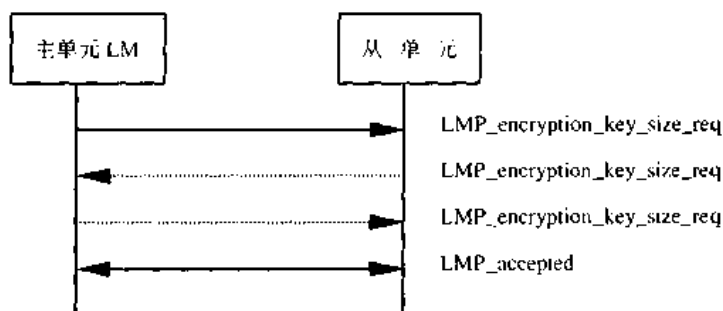


图 3.16 加密字长度协商成功

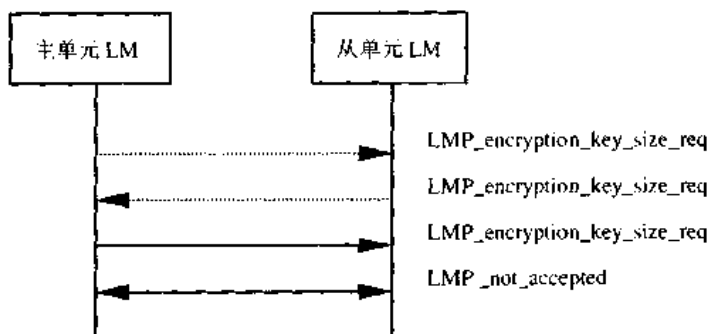


图 3.17 加密字长度协商失败

3. 开始加密

加密过程启动后，主机将发送随机数 `EN RAND` 并计算出加密字为 $Kc=E3$ （当前连接字，`EN RAND`，`COF`）。如果匹克网支持广播加密，该随机数必须在所有的从单元上都一样。然后，主单元发送含有 `EN RAND` 的 `LMP_start_encryption_req` 分组，如图 3.18 所示。当从单元收到该消息并通过 `LMP_accepted` 确认后，从单元就开始计算 Kc 值。对于主、从单元， Kc 和 `EN RAND` 都被用作加密运算法则的输入参数。

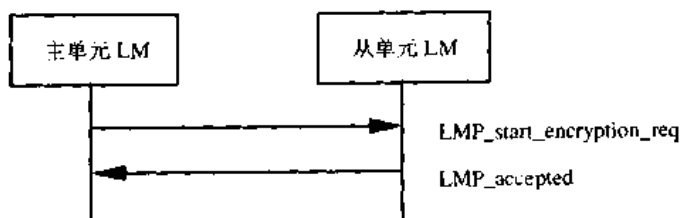


图 3.18 开始加密

在开始加密前，必须暂停高层数据通信，以防主、从单元收到错误数据。加密过程分

成三步来进行：

- 设置主单元，以传输未加密数据分组，并接收加密数据分组；
- 设置从单元，传输和接收加密数据分组；
- 设置主单元，传输和接收加密数据分组。

在 1、2 步之间传输 LMP_start_encryption_req 后，就可进行主单元到从单元的数据传输。当从单元收到 LMP_start_encryption_req 时，激活第二步。在第二、三步之间传输 LMP_accepted 后，可进行从单元对主单元传输。当主单元收到 LMP_accepted 时，激活第三步。

4.停止加密

停止加密模式如图 3.19 所示。在停止加密以前，必须停止高层数据通信，以防主、从单元收到错误数据。停止加密过程也分成三步来进行，它类似于开始加密的过程。

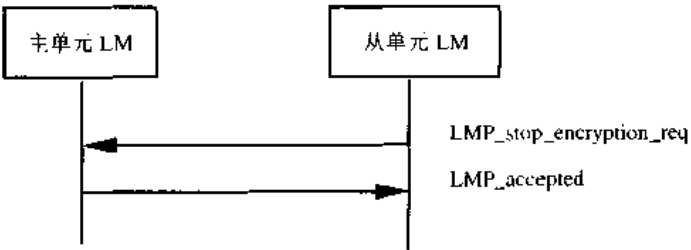


图 3.19 停止加密

- 设置主单元，传输未加密数据分组，并接收加密数据分组；
- 设置从单元，传输和接收加密数据分组；
- 设置主单元，传输和接收未加密数据分组。

在第一、二步之间发送 LMP_stop_encryption_req 后，可进行主单元到从单元的通信。当从单元收到 LMP_stop_encryption_req 时，激活第二步。在第二、三步之间发送 LMP_accepted 后，可进行从单元到主单元的通信。当主单元收到 LMP_accepted 时，激活第三步。

如果需要改变加密模式、加密字或加密随机数，必须首先停止加密，然后再用新的参数重新启动。

3.3.7 请求时钟补偿

当从单元收到跳频（FHS）分组时，就开始计算从单元时钟值与主单元时钟值之差。主单元时钟值分组包含在 FHS 有效载荷内。每次从主单元接收数据分组时都要计算该时钟补偿值。在连接过程中，主单元可在任意时间上请求时钟补偿。通过记录时间补偿，主单元就可知道当从单元离开匹克网后，在哪一条 RF 信道上被激发过。这样，在下次对该设备呼叫时，可以缩短呼叫时间。用于时钟补偿请求的 PDU 如表 3.8 所示；时钟补偿请求模式如图 3.20 所示。

表 3.8 用于时钟补偿请求的 PDU

M/O	协议数据单元	内容
M	LMP_clkoffset	
M	LMP_clkoffset_res	时钟补偿

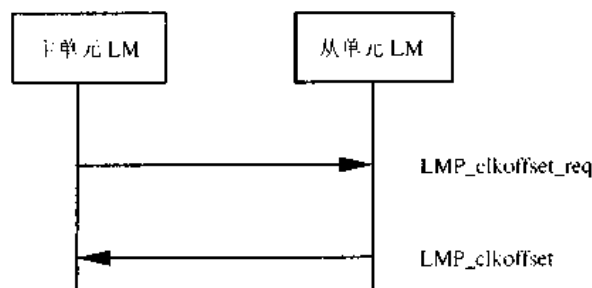


图 3.20 时钟补偿请求

3.3.8 时钟补偿信息

通过利用 LMP_slot_offset，可传输不同匹克网之间的时间补偿信息，如图 3.21 所示。协议数据单元中携带有时钟补偿信息和 BD_ADDR 参数，如表 3.9 所示。时钟补偿以 μs 为单位，从第一个匹克网中主单元 TX 时隙的开始时间到第二个匹克网主单元 TX 时隙的开始时间。其中，第一个匹克网用于传输 PDU，第二个匹克网的主单元地址为 BD_ADDR。

在进行主、从单元切换之前，可以从将被切换成主单元的设备发送 PDU。如果由主单元的初始化来实现切换过程，则从单元在发送 LMP_accepted 前，就应先发送 LMP_slot_offset。如果由从单元的初始化来实现切换过程，从单元在发送 LMP_switch_req 前，就应先发送 LMP_slot_offset。PDU 也可用于匹克网与匹克网之间的通信。

表 3.9 用于时钟补偿信息的 PDU

M/O	协议数据单元	内容
O	LMP_slot_offset	时隙补偿 BD_ADDR

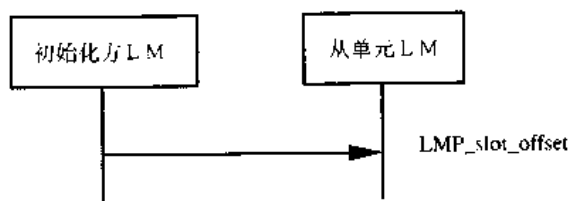


图 3.21 发送时钟补偿信息

3.3.9 计时精度信息请求

LMP 支持计时精度请求。当准备返回到保持状态并增加扫描窗口为最大保持时间时，该信息能够针对给定的保持时间为最小化扫描窗口。而且，该信息也可在扫描呼吸模式时隙和扫描休眠模式信标数据分组时，用于最小化扫描窗口。返回的计时精度参数是 drift 和 jitter。

其中 drift 以 ppm 为单位, jitter 在保持、呼吸和休眠模式中使用时钟值以 μs 为单位。这些参数对于一台特定设备将固定不变,而且要求在多次请求中都应保持一致。如果设备不支持计时精度信息,那么当它收到该信息请求时,就发送附带原因码 unsupported LMP feature (不支持的 LMP 特性) 的 LMP_not_accepted 消息。请求方设备在这种情况下必须采用计时精度低限值 (drift=250 ppm, jitter=10 μs)。用于请求计时精度信息的 PDU 如表 3.10 所示,请求设备支持或不支持计时精度信息时的处理过程如图 3.22 和图 3.23 所示。

表 3.10 用于请求计时精度信息的 PDU

M/O	协议数据单元	内容
O	LMP_timing_accuracy_req	
O	LMP_timing_accuracy_res	Drift jitter

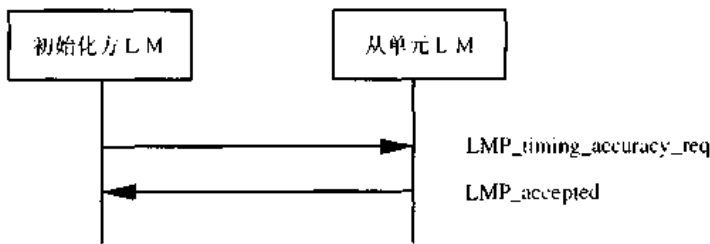


图 3.22 被请求设备支持计时精度信息时的处理过程

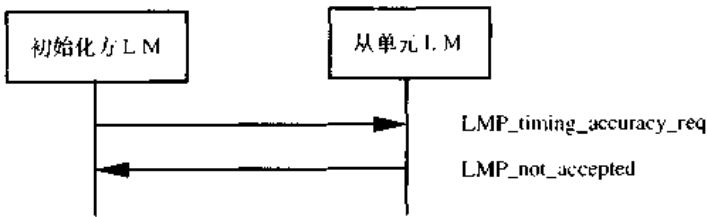


图 3.23 被请求设备不支持计时精度信息的处理过程

3.3.10 LMP 版本

LMP 支持对 LMP 协议版本的请求。若被请求设备发送响应消息,则该消息应具有三个参数: VersNr, CompId 和 SubVersNr。VersNr 说明设备支持的蓝牙 LMP 规范版本; CompId 用于跟踪低层蓝牙可能出现的问题,任何创建连接管理器各自执行版本的厂商都具有自己的 CompId; 同样这些公司也要负责管理和维护 SubVersNr。建议所有厂商为每一 RF/BB/LM 执行版本创建各自的 SubVersNr。对于某一给定 VersNr 和 CompId, 每发布一个新的执行版本, SubVersNr 的值就必须随之增加。对于 CompId 和 SubVersNr, 0xFFFF 的值表示没有使用合法值,不能针对 LMP 版本执行协商机制。表 3.11 所示是用于 LMP 版本请求的 PDU, 图 3.24 是 LMP 版本请求的处理过程。

表 3.11 用于 LMP 版本请求的 PDU

M/O	协议数据单元	内容
M	LMP_version_req	VersNr Compld SubVersNr
M	LMP_version_res	VersNr Compld SubVersNr

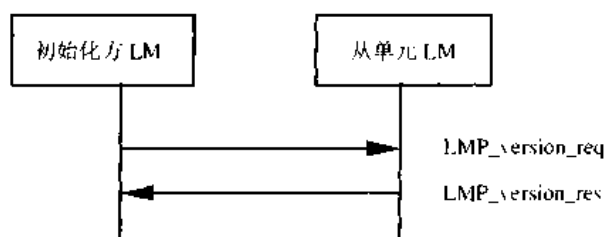


图 3.24 LMP 版本请求

3.3.11 蓝牙支持特性

蓝牙无线电和链路控制器只支持基带协议和无线电规范中所定义的数据分组类型和特性的部分子集。LMP_features_req 与 LMP_features_res 等 PDU 就用于交换这一信息，如表 3.12 所示。一台设备在获得其他设备支持特性信息之前，只能发送 ID、FHS、NULL、POLL 和 DMI 数据分组。只有在执行了特性请求后，才能够发送通信双方都可以共同支持的数据分组类型。一旦发出请求，该请求必须与其他设备的支持特性相兼容。例如，当建立一个 SCO 链路时，如果其他设备不支持 HV3 数据分组，最好建议初始化方也不要使用 HV3 数据分组。只有 LMP 切换注册和时钟偏移信息例外，当两蓝牙设备建立连接时，并且请求方还未获知另一方特性以前，它们可以作为第一个 LMP 消息发送出去（切换是可选特性）。请求支持特性的过程如图 3.25 所示。

表 3.12 用于特性请求的 PDU

M/O	协议数据单元	内容
M	LMP_features_req	features
M	LMP_features_res	features

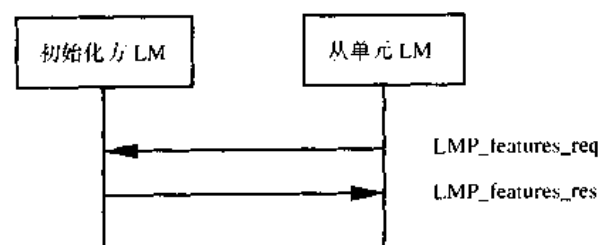


图 3.25 请求支持特性的过程

3.3.12 主、从角色切换

由于呼叫设备一般都是匹克网中的主单元, 因此有时需要进行主、从单元角色的切换, 用于主从切换的 PDU 如表 3.13 所示。假设设备 A 为从单元, 设备 B 为主单元, 在初始化切换设备时, 应结束当前 L2CAP 消息的传输, 然后再发送 LMP_switch_req。

如果接受切换, 另一设备也将结束当前 L2CAP 消息的传输, 并以 LMP_accepted 应答。然后, 执行基带协议的过程, 如图 3.26 所示。

如果拒绝切换, 另一设备以 LMP_not_accepted 应答, 不进行切换, 如图 3.27 所示。

表 3.13 用于主从切换的 PDU

M/O	协议数据单元	内容
O	LMP_switch_req	

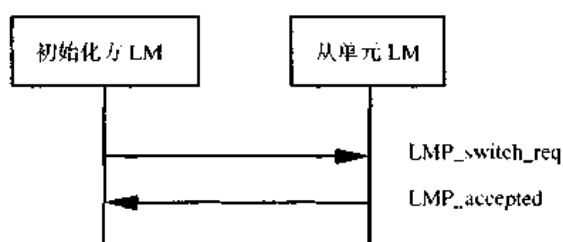


图 3.26 接受主从切换

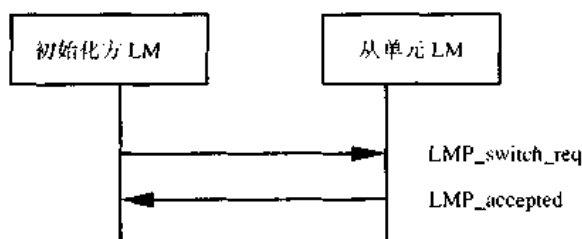


图 3.27 未接受的主从切换

3.3.13 请求命名

LMP 支持向另一蓝牙设备请求命名, 用于名字请求的 PDU 如表 3.14 所示。名字是与蓝牙设备相关联的名字。根据 UTF-8 标准, 名字最多可由 248 个编码字节构成。名字分为一个或多个 DM1 数据分组。当发送 LMP_name_req 时, 名字偏移表示需要哪一段。对应的 LMP_name_res 具有相同的名字偏移, 名字长度表示蓝牙设备名字的总字节数和名字段:

表 3.14 用于名字请求的 PDU

M/O	协议数据单元	内容
	LMP_name_req	名字偏移
	LMP_name_res	名字偏移 名字长度 名字段

- 如果 $(N + \text{名字偏移}) < \text{名字长度}$, 名字段 $(N) = \text{名字}(N + \text{名字偏移})$
- 否则, 名字段 $(N) = 0$

其中, $0 \leq N \leq 13$ 。在第一个发出的 LMP_name_req 中, 名字偏移等于零。同时重复序列 25, 直到初始化方收集到所有名字段。请求命名及其响应的过程如图 3.28 所示。

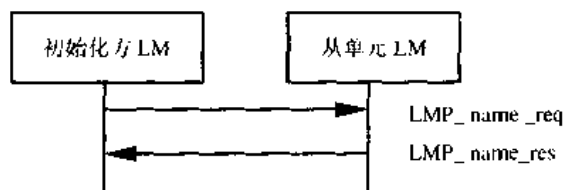


图 3.28 请求的设备名及其响应

3.3.14 断开连接

两蓝牙设备间的连接可在任意时间由主单元或从单元关闭。同时, 在通知另一方的消息分组中含有通信关闭原因的参数。用于断开连接的 PDU 如表 3.15 所示; 断开连接的处理过程如图 3.29 所示。

表 3.15 用于断开连接的 PDU

M/O	PDU	内容
M	LMP_detach	原因

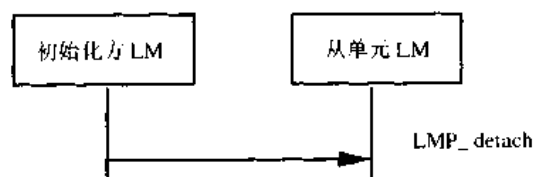


图 3.29 通过发送 LMP_detach 关闭通信

3.3.15 保持模式

两设备间的 ACL 链路可在指定保持时间内置为保持状态, 用于保持模式的 PDU 如表 3.16 所示。在保持状态下, 主单元不再发送 ACL 数据分组。一般单元如果在相当长的一段时间内不发送数据, 那么该单元就可进入保持模式。考虑到设备的节能问题, 在保持模式下应关闭收发器。但是如果设备要搜索其他设备或考虑到将被其他蓝牙设备搜索, 或它可能要加入其他匹克网时, 也可使用保持模式。实际上, 设备在保持时间内的动作不是由保持消息决定的, 而是由各设备自己决定。

表 3.16 用于保持模式的 PDU

M/O	协议数据单元	内容
O	LMP_hold	控制时间
O	LMP_hold_req	控制时间

1. 主机强制保持模式

如果存在预先已经接受的保持模式请求，就可使用主单元强制保持模式，如图 3.30 所示。当请求保持模式时，保持时间分组含 PDU，同时主单元强制模式不得长于从单元已接收的任何保持时间。

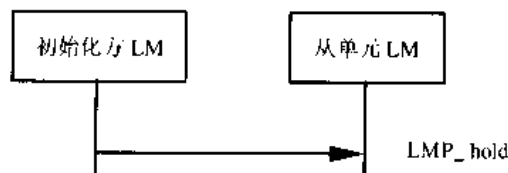


图 3.30 主单元强制从单元进入控制模式

2. 从机强制保持模式

如果存在预先已经接受的保持模式请求，就可使用从单元强制保持模式，如图 3.31 所示。当请求保持模式时，保持时间分组含 PDU，同时从单元强制模式不得长于主单元已接收的任何保持时间。

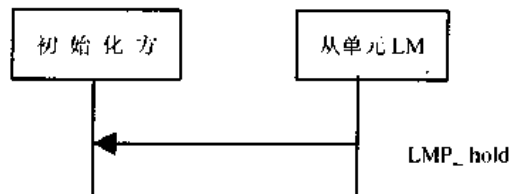


图 3.31 从单元强制保持模式

3. 保持模式请求

主单元或从单元可以请求进入保持模式。一旦收到该请求，将返回含有参数修改后的相同请求，否则将终止协商。如果达成一致，发出 LMP_accepted 后终止协商，并将 ACL 链路置为保持模式。如果未达成一致，发出 LMP_not_accepted 后终止协商（LMP_not_accepted 不含原因码 unsupported parameter value），不能进入保持模式。其处理过程如图 3.32 所示。

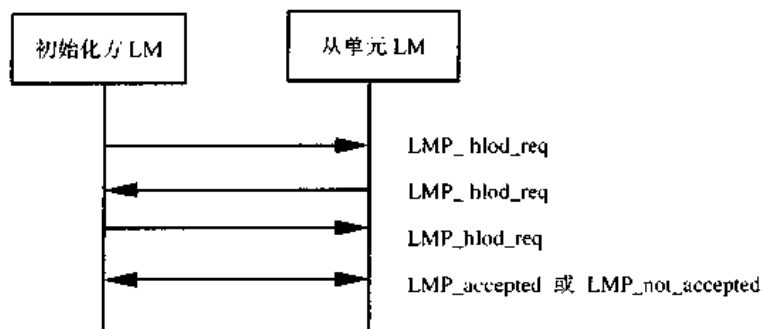


图 3.32 保持模式协商

3.3.16 呼吸模式

为了进入呼吸模式，主从单元将对呼吸间隔 T_{sniff} 和呼吸偏移 D_{sniff} 进行协商。二者说明了呼吸时隙的计时方式。呼吸偏移确定了第一个呼吸时隙的持续时间，然后按照呼吸间隔

T_{sniff} 周期性生成呼吸时隙。为了避免在初始化时有时钟隐含的问题，应针对第一次时隙的计算选择两个参数中的一个。由主单元发出消息中的计时控制标志就表示了使用哪一个参数。注意：域中只有第一位有效。当链路处于呼吸模式时，主单元只能在呼吸时隙中进行数据传输。同时由以上所述的两个参数来控制从单元的侦听动作。呼吸尝试确定了呼吸时隙开始计时时间，并且从单元必须侦听时隙的个数。即使还没有收到一个含有 AM 地址的数据分组，也必须如此。如果从单元继续接收只含有 AM 地址的数据分组，呼吸尝试参数将确定从单元必须侦听的时隙个数。用于呼吸模式的 PDU 如表 3.17 所示。

表 3.17 用于呼吸模式的 PDU

M/O	协议数据单元	内容
O	LMP_sniff	计时控制标志, D_{sniff} T_{sniff} 呼吸尝试 呼吸超时
O	LMP_sniff_req	呼吸控制标志, D_{sniff} T_{sniff} 呼吸尝试 呼吸超时
O	LMP_unsniff_req	-

1. 主单元或从单元请求呼吸模式

主单元或从单元可以请求进入呼吸模式。一旦收到该请求，将返回含有参数修改后的相同请求，否则将终止协商。如果达成一致，LMP_accepted 则终止协商，并将 ACL 连接置于呼吸模式。如果未达成一致，LMP_not_accepted 则终止协商，LMP_not_accepted 不含原因码 unsupported parameter value，且不能进入呼吸模式，如图 3.34 所示。

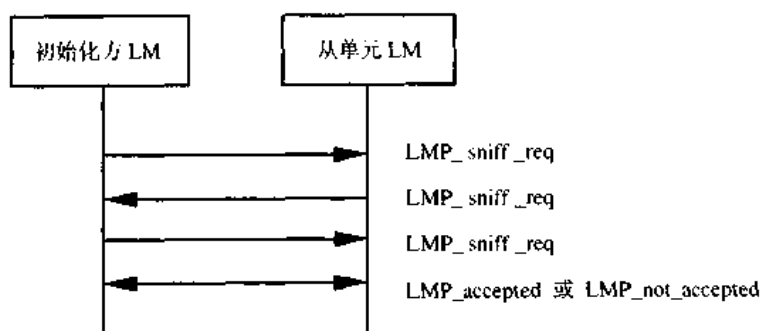


图 3.34 呼吸模式协商

2. 将从单元从呼吸模式转为活动模式

通过发送 LMP_unsniff_req 协议数据，从单元可结束呼吸模式。被请求设备必须以 LMP_accepted 应答。如果是从单元请求，则在收到 LMP_accepted 后进入活动模式。如果是主单元请求，从单元则在收到 LMP_unsniff_req 后进入活动模式，如图 3.35 所示。

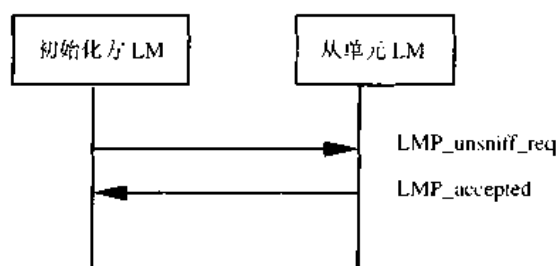


图 3.35 从单元从呼吸模式转为活动模式

3.3.17 休眠模式

即使从单元不加入信道，它仍然要执行同步跳频过程。这时它应设置成休眠模式。在这种模式中，设备将放弃它原来具有的 AM_ADDR，但该设备仍然可以通过信标间隔的信标实例被唤醒并与信道实现重新同步。信标间隔、信标偏移量和表示第一个信标实例计算方法的标志确定了第一个信标实例。此后，以预先定义的信标间隔周期性地发送信标实例。在信标实例中，休眠从单元能由主单元激活，主单元也能改变休眠模式的参数，并传输广播信息或使休眠从单元实现信道访问的请求。

广播消息的过程就是由匹克网中的主单元向所有的休眠从单元发送 PDU。而这些 PDU (由 LMP_set_broadcast_scan_window, LMP_modify_beacon, LMP_unpark_BD_addr_req 和 LMP_unpark_PM_addr_req 组成)是惟一能够发送到休眠从单元或用于广播的 PDU 的。为了增加广播消息的可靠性，数据分组应尽量短。所以，LMP 的协议数据单元格式允许有差异，且其参数并不总是以字节序列进行排列，同时 PDU 的长度值也可改变。

控制休眠模式的消息分组含有多个参数，这些参数都在基带协议中作了定义。一旦将某一从单元置于休眠模式，也就同时给它指定了惟一的 PM_ADDR。主单元利用这个 PM_ADDR 可以解除从单元的休眠模式。如果 PM_ADDR 各位全部为零，则表示该 PM_ADDR 不合法。如果给某一设备指定了 PM_ADDR，那么在该设备解除休眠模式时，该设备的 PM_ADDR 必须与 AM_ADDR 一致。

1. 主单元强制从单元进入休眠模式

主单元能够执行强制从单元进入休眠模式的操作。进行这种操作时，主单元首先要结束当前的 L2CAP 消息传输，并发送 LMP_park。从单元收到该 PDU 时，就将结束当前的 L2CAP 消息传输，并向主单元发送 LM_accepted 应答，如图 3.36 所示。

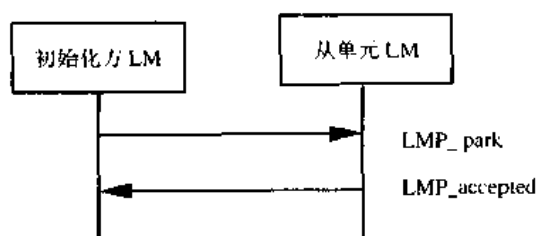


图 3.36 从单元强制休眠模式

2. 主单元请求从单元进入休眠模式

主单元请求从单元进入休眠模式时，主单元首先要结束当前的 L2CAP 消息传输，并发

送 LMP_park_req。如果从单元同意进入休眠模式，它也将结束当前 L2CAP 消息传输，并发送 LMP_accepted 应答。最后由主单元发出 LMP_park。如果从单元拒绝进入休眠模式，则从单元将向主单元发送 LMP_not_accepted 应答。同意和拒绝进入休眠模式的处理过程如图 3.37 和图 3.38 所示。

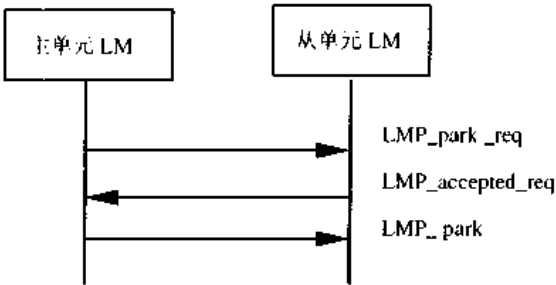


图 3.37 从单元同意进入休眠模式

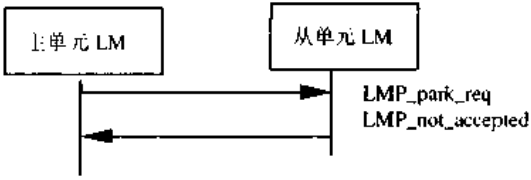


图 3.38 从单元拒绝进入休眠模式

3. 从单元请求置于休眠模式

若从单元希望置于休眠模式，从单元首先应结束当前的 L2CAP 消息传输，并发送 LMP_park_req。如果主单元接收从单元的休眠模式请求，就结束当前的 L2CAP 消息传输，并向从单元发送 LMP_park。如果主单元拒绝从单元的休眠模式请求，则主单元将向从单元发送 LMP_not_accepted，分别如图 3.39 和图 3.40 所示。

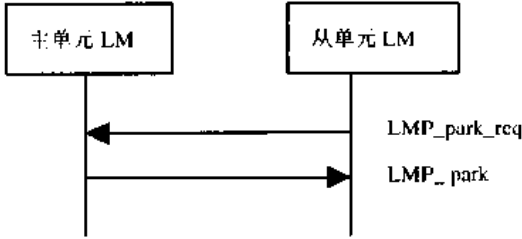


图 3.39 主单元同意将从单元置为休眠模式

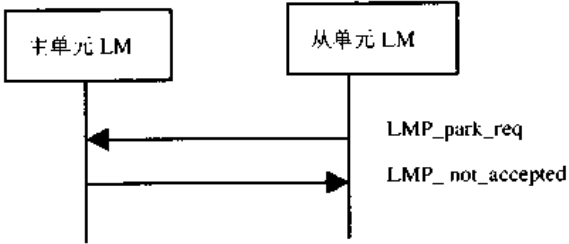


图 3.40 主单元拒绝将从单元置为休眠模式

4. 主单元建立广播扫描窗口

如果主单元希望建立广播扫描窗口，同时又需要有比信标队列具有更大的广播容量，主单元将发送 LMP_set_broadcast_scan_window，如图 3.41 所示，通知从单元将会有更多广播信息在信标队列后传递过来。LMP_set_broadcast_scan_window 通常是在信标时隙中以广播数据分组的形式发送。扫描窗口在信标实例中启动，并且只对当前信标有效。

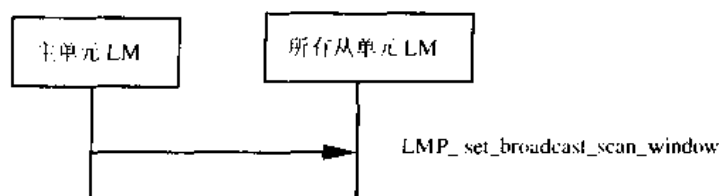


图 3.41 主单元通知所有从单元增加广播容量

5. 主单元修改信标参数

当信标参数发生变化时，主单元就通过发送 **LMP_modify_beacon** 把这个变化通知所有处于休眠状态的从单元，如图 3.42 所示。该消息通常在信标时隙中以广播数据分组形式传输。



图 3.42 主单元修改信标参数

6. 解除休眠的从单元

主单元能够通过发送广播 LMP 消息解除一个或多个从单元的休眠状态。该广播 LMP 消息分组含有设备的 **PM_ADDR** 或 **BD_ADDR**，而这些设备由主单元在信标时隙中来解除休眠的从单元。该广播 LMP 消息分组含有主单元分配给从单元的 **AM_ADDR**。在发送此消息后，主单元必须通过轮询每一个被解除休眠状态的从单元(也就是发送 **POLL** 分组)来检查解除休眠过程是否成功，以使该从单元获得可访问信道的授权。已解除休眠的从单元必须通过发送 **LMP_accepted** 进行应答。如果主单元发出解除休眠消息以后，在一定时间内没有收到 **LMP_accepted**，则解除休眠过程失败，而主单元必须考虑从单元是否仍需处于休眠模式。

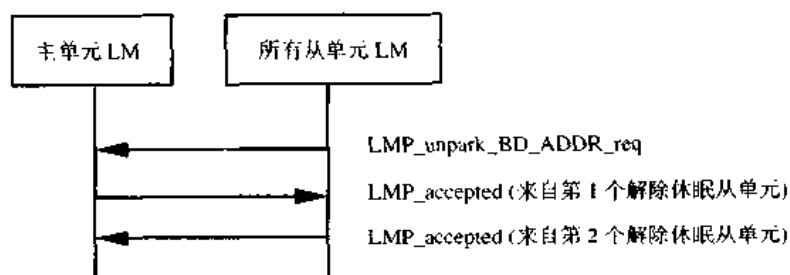


图 3.43 主单元将地址为 **BD_ADDR** 的从单元解除休眠

有两种消息可供使用。一种用于以 **PM_ADDR** 标识的休眠设备，另一种用于以 **BD_ADDR** 标识的休眠设备。两种消息都具有可变长度，其长度取决于主单元要解除休眠的从单元的个数。对于主单元要解除休眠的每一从单元，**AM_ADDR** 分组包含在有效载荷中，且 **AM_ADDR** 位于设备 **PM/BD_ADDR** 之后，而该设备地址则指定为 **AM_ADDR**。如果从单元由 **PM_ADDR** 标识，那么利用同一消息最多可将七个从单元解除休眠。如果从单元由

BD_ADDR 标识，那么利用同一消息只能将两个从单元解除休眠。分别如图 3.43 和图 3.44 所示。

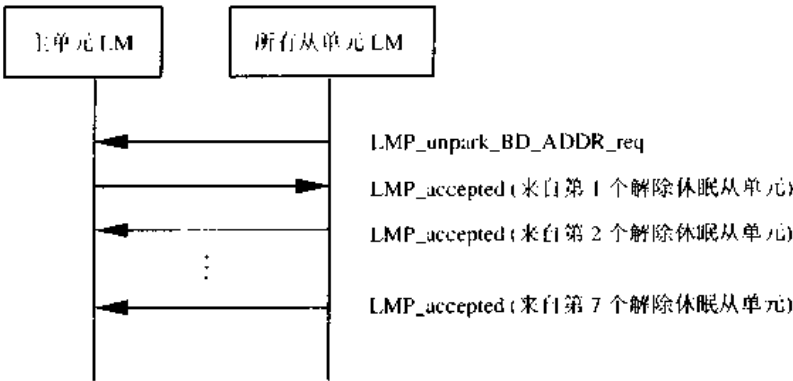


图 3.44 主单元将地址为 PM_ADDR 的从单元解除休眠

3.3.18 功率控制

如果接收信号场强指示（RSSI）值与蓝牙设备约定值差别太大，它可以增加或降低其他设备的 TX 功率。一旦收到这一消息，输出功率就会增加或降低一个等级。主单元侧的 TX 功率完全独立于其他从单元；从单元的请求只能影响相对于同一从单元的主单元 TX 功率。用于功率控制的 PDU 和处理过程分别如表 3.19 和图 3.45 所示。

表 3.19 用于功率控制的 PDU

M/O	协议数据单元	内容
O	LMP_incr_power_req	保留（1 个字节）
O	LMP_decr_power_req	保留（1 个字节）
O	LMP_max_power	-
O	LMP_min_power	-

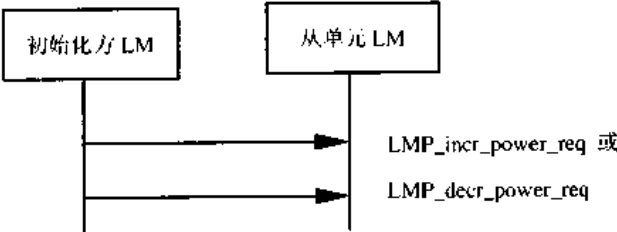


图 3.45 一台设备请求改变其他设备的 TX 功率

如果 LMP_incr_power_req 的接收者已经使用最大功率进行发送，则返回 LMP_max_power，如图 3.46 所示。如果设备已进行了至少一次的功率降低请求，那它只能再进行一次功率增大请求。同样，如果 LMP_decr_power_req 的接收者已经使用最小功率进行发送，则返回 LMP_min_power，如图 3.47 所示。如果设备已请求至少一次的功率增加，那它只能再请求一次功率降低。

在 LMP_incr/decr_power_req 中有一个字节保留起来以备将来使用。例如，它有可能用于作为约定 RSSI 与实际测试到的 RSSI 之间匹配的差值。LMP_incr/decr_power_req 的接受者就可以利用该值来马上将功率调节为正确值，而不是每收到一次请求才改变一个等级。在

没定义以前，参数值在所有的 LMP 版本中都必须为 0x00。

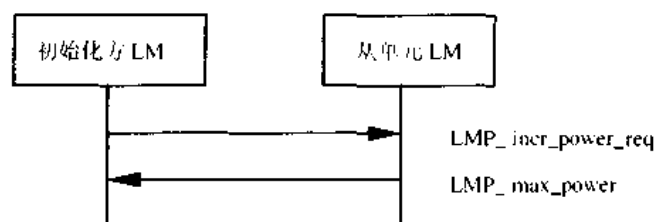


图 3.46 不能增强 TX 功率

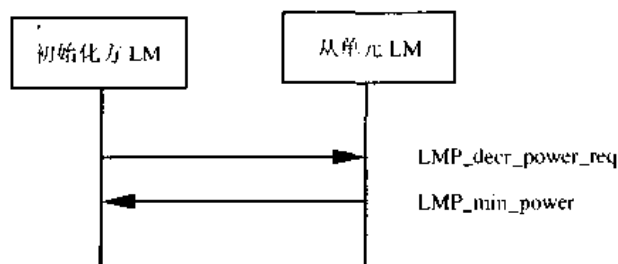


图 3.47 不能降低 TX 功率

3.3.19 在 DM 和 DH 之间基于质量的信道变化

一台设备通常可以设置为使用 DM 数据分组，或 DH 数据分组，或根据信道质量自动调整使用的数据分组类型。但是，所有设备都要求能够传送 DM 和 DH 数据分组。DM 与 DH 的区别在于 DM 的数据分组由 2/3 前向比例纠错码来保护有效载荷，而 DH 中的有效载荷不受任何 FEC 前向纠错码的保护。如果一设备需要自动调节使用 DM 或 DH 数据分组类型，它就要向其他设备发送 LMP_auto_rate，如图 3.48 所示。基于 LC 质量测试，设备可以通过数据分组类型的变化确定是否将增加吞吐量。如果要增加，就向其他设备发送 LMP_preferred_rate，如图 3.49 所示。其中，使用到的 PDU 如表 3.20 所示。

表 3.20 用于质量驱动的数据误码率变化的 PDU

M/O	协议数据单元	内容
O	LMP_auto_rate	-
O	LMP_preferred_rate	数据误码率

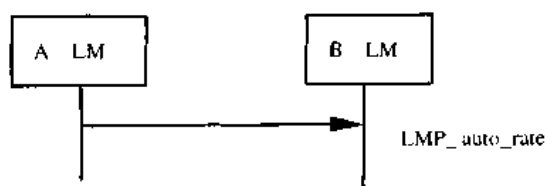


图 3.48 左手单元设置为自动改变使用 DM 或 DH

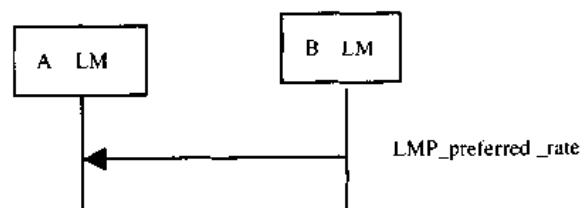


图 3.49 右手设备命令左手设备改变数据误码率

3.3.20 服务质量(QoS)

链路管理提供支持服务质量的能力。从主单元到特定从单元的连续传输之间的最大间隔时间称为轮询间隔时间。轮询间隔用于支持带宽分配和延迟控制。除了发生呼叫冲突、呼叫扫描冲突、查询冲突和查询扫描冲突的情况外，都应保证轮询间隔时间。另外，主从单元应就广播数据分组的重复次数 N_{BC} 进行协商。用于 QoS 的 PDU 如表 3.21 所示。

表 3.21 用于 QoS 的 PDU

M/O	协议数据单元	内容
M	LMP_qualityPof_service	Polling interval N_{BC}
M	LMP_quality_of_service_req	Poll interval N_{BC}

1. 主单元服务质量通知从单元

当主单元需要将服务质量通知从单元时，主单元必须将新的轮询间隔时间和 N_{BC} 通知从单元，如图 3.50 所示。在这种情况下任何从单元都不得拒绝该通知。

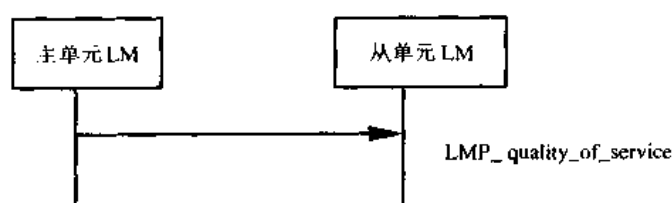


图 3.50 主单元将新的服务质量通知从单元

2. 设备请求新的服务质量

在这种情况下，主单元或从单元将请求一个新的轮询间隔时间和 N_{BC} 。 N_{BC} 参数只有在从主单元发往从单元时才有意义。对于从单元发来的 LMP_quality_of_service_req 协议数据单元，主单元将忽略该参数。同时请求可以被接收或被拒绝，这样，主从单元可以就所需服务质量进行动态协商，分别如图 3.51 和图 3.52 所示。

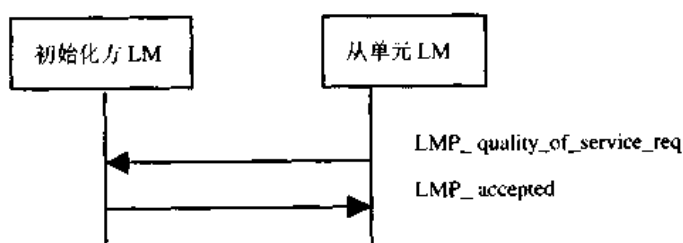


图 3.51 设备接受新的服务质量

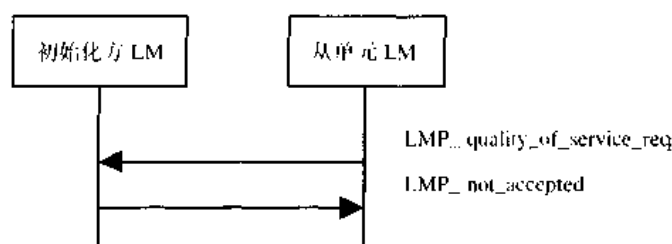


图 3.52 设备拒绝新的服务质量

3.3.21 SCO 链路

在两个不同的蓝牙设备之间建立连接时，该连接由 ACL 链路组成。然后，就可以建立起一条或多条 SCO 链路。SCO 链路保留由 SCO 间歇时间、 T_{SCO} 分开所得到的时隙。SCO 链路保留的第一个时隙由 T_{SCO} 、SCO 延迟、 D_{SCO} 来定义。此后，SCO 时隙将按 SCO 间歇时间进行周期性的发送。为了避免 SCO 链路初始化过程中由时钟带来的隐含问题，应在从主单元发出的消息分组中含有一个标志。该标志表示第一个 SCO 时隙是如何计算出来的。注意：只有该字段的第 0 位和第 1 位有效。SCO 链路通过 SCO 句柄相互区别。不能使用 SCO 零句柄。

1. 主单元初始化 SCO 链路

建立 SCO 链路时，主单元发送带参数的请求，这些参数用于对 SCO 链路的计时方式、数据分组类型和编码方式进行说明。对于蓝牙支持的每一个 SCO 数据分组，在无线接口方面，蓝牙应支持三种不同的声音编码格式： μ -law log PCM, A-law log PCM and CVSD。

用于 SCO 链路的时隙由主单元控制的三个参数决定。这三个参数是 T_{SCO} 、 D_{SCO} 和表示第一个 SCO 时隙计算方式的标志。发出第一个时隙以后，将按照 T_{SCO} 周期性发送 SCO 时间片。如果从单元不接受 SCO 链路，而是愿意考虑接受另一个 SCO 参数集，那么它就可以在 LMP_not_accepted 出错原因字段里标识它有哪些参数不能接收。而主单元就可能发送一个含有参数已修改后的新请求。主单元初始化 SCO 链路的处理过程如图 3.53 所示。

该消息中的 SCO 句柄必须区别于已存在的 SCO 链路。

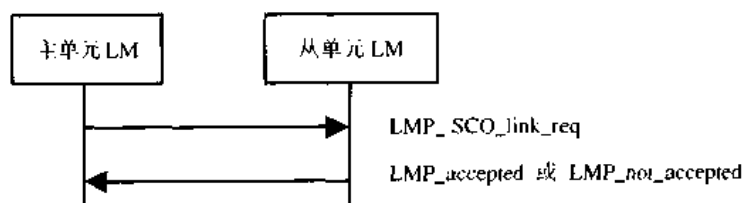


图 3.53 主单元请求 SCO 链路

2. 从单元初始化 SCO 链路

从单元也可以对 SCO 链路进行初始化。从单元发送 LMP_SCO_link_req，但其中计时控制标志、 D_{SCO} 都应置为无效，而 SCO 句柄值应为零。如果主机不能建立 SCO 链路，就通过发送 LMP_not_accepted 作出应答，如图 3.54 所示。否则它将返回 LMP_SCO_link_req。该消息分组含有指定的 SCO 句柄、 D_{SCO} 和计时控制标志。而且，主单元应尽量采用与从单

元请求中的其他相同参数；如果主单元不能满足该请求，主单元也可以使用其他参数值。但从单元必须使用 `LMP_accepted` 或 `LMP_not_accepted` 来进行响应，如图 3.55 所示。

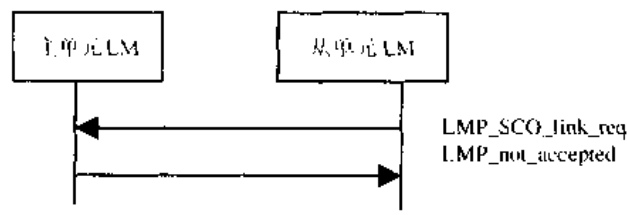


图 3.54 主单元拒绝从单元的建立 SCO 链路的请求

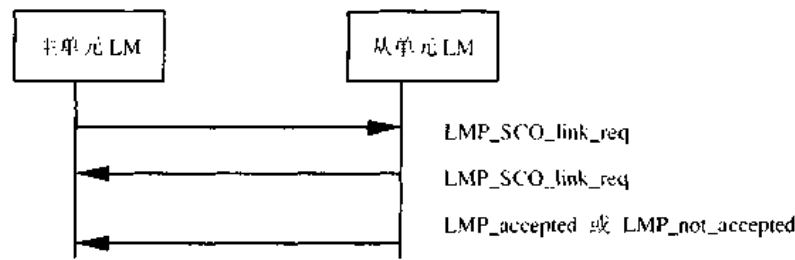


图 3.55 主单元接受从单元使用 SCO 链路的请求

3. 主单元请求改变 SCO 参数

在这种情况下，由主单元发送 `LMP_SCO_link_req`。其中 SCO 句柄是主单元希望改变参数的 SCO 链路的句柄。如果从单元同意接收这个新参数，它就用 `LMP_accepted` 来应答，并且 SCO 链路的旧参数将变为新参数。如果从单元不同意接收这个新参数，它就用 `LMP_not_accepted` 来应答，而 SCO 链路的原参数将继续保持不变。在这种情况下，从单元应以 `LMP_not_accepted` 来应答，并将在出错原因参数里表明哪些参数不能接收。然后，主单元再次用修改后的参数来达到对 SCO 链路的改变。

4. 从单元请求改变 SCO 参数

从单元发送 `LMP_SCO_link_req` 时，其中 SCO 句柄是从单元希望改变 SCO 链路的参数句柄。该消息中的计时控制标志和 D_{SCO} 参数都应置为无效。当主单元不接受从单元发送的新参数时，主单元将发送 `LMP_not_accepted` 对从单元作出应答，且原 SCO 的连接继续保持不变。如果主单元接受从单元所发送的新参数时，主单元将发送 `LMP_SCO_link_req` 对从单元作出应答，此时 SCO 的连接则必须使用与从单元请求中所提出的相同参数。

5. 撤销 SCO 连接

主从单元可以通过发送特定请求撤销 SCO 链路，该请求分组含有需要撤销的 SCO 链路句柄，以及撤销 SCO 链路的原因。接收方必须以 `LMP_accepted` 应答。

3.3.22 多时隙分组控制

从单元使用的时隙数量可在返回分组中进行限制。主单元允许从单元发送提供最大时隙参数的 `LMP_max_slots` 协议数据单元，来说明从单元使用的最大时隙量。同时从单元可以通过发送含有最大时隙参数的 `LMP_max_slot_req` 协议数据单元，来使用最大时隙量。系

统的默认值是 1 个时隙。也就是说，如果从单元没有利用时隙数量信息去通知主单元，从单元就只能使用单时隙分组。表 3.23 说明了两种使用多时隙分组的 PDU。

表 3.23 用于控制多时隙分组的 PDU

M/O	协议数据单元	内容
M	LMP_max_slot	最大时隙数量
M	LMP_max_slot_req	最大时隙数量

3.3.23 呼叫方案

除了强制呼叫方案外，蓝牙还设置了可选呼叫。LMP提供了一种协商呼叫方案的方法。该方法用于下次匹克网中的单元有被呼叫的情况。请求呼叫方案的PDU如表3.24所示。

表 3.24 请求呼叫方案的 PDU

M/O	协议数据单元	内容
O	LMP_page_mode_req	呼叫方案 呼叫方案设置
O	LMP_page_scan_mode_req	呼叫方案 呼叫方案设置

1. 呼叫模式

当设备 A 呼叫设备 B 时，由设备 A 启动该过程，并就呼叫方案进行协商。当设备 A 提出一个含有新参数的呼叫方案时，设备 B 可接受或可拒绝它，如图 3.60 所示。如拒绝则表示原设置保持不变。但设备 A 工作在强制模式下时，设备 B 必须接受切换并进入强制方案的请求。

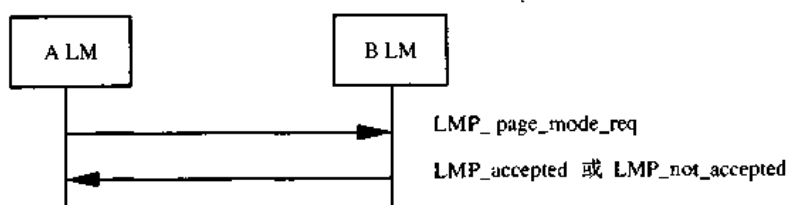


图 3.60 呼叫模式协商

2. 呼叫扫描模式

该过程完全类似于呼叫模式。当设备 A 呼叫设备 B 时，由设备 A 启动该过程，并就呼叫方案进行协商。设备 A 提出一个含有新参数的呼叫方案时，设备 B 可接受或可拒绝它，如图 3.61 所示。如拒绝则表示原设置保持不变。但设备 A 工作在强制模式下时，设备 B 必须接受切换并进入强制方案的请求。

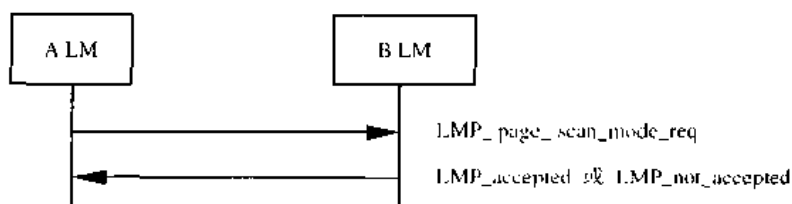


图 3.61 呼叫扫描模式协商

3.3.24 链路监控

每一蓝牙链路都具有一个用于链路监控的计时器。该计时器用于检测因设备移出范围而引起的链路丢失、设备关机、或者其他通信失败的情况。本方案在基带协议中作了一些说明，此处就不再进行深入讨论。该 LMP 过程用于对超时进行监控设置，如表 3.25 和图 3.62 所示。

表 3.25 用于设置监控最大持续时间的 PDU

M/O	协议数据单元	内容
M	LMP_supervision_timeout	监控超时

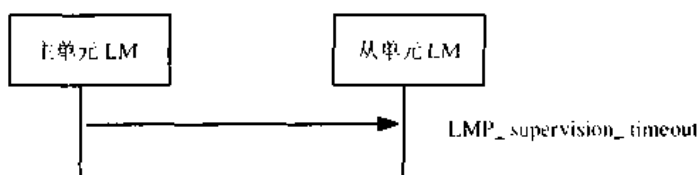


图 3.62 设置链路监控最大持续时间

3.4 建立连接

呼叫过程执行以后，主单元必须通过发送 POLL 或 NULL 分组对从单元进行轮询，然后再执行 LMP 的过程，同时该过程并不需要有介于 LM 和被叫方主机间的接口。

如果创建 LM 以上层次连接，主叫设备应发送 LMP_Host_Connection_req。一旦呼入方收到该消息，并通知主机呼入连接的建立。远程设备可通过发送 LMP_accepted 或 LMP_not_accepted 来接受或拒绝连接请求。

如果设备不需要进一步的连接建立过程，应发送 LMP_Setup_complete。而且，不需要建立进一步连接的设备仍然要对其他设备的请求作出应答。当其他设备准备建立连接时，应发送 LMP_setup_complete。然后在与 LMP 不同的逻辑信道上发送第一个分组。连接建立过程如图 3.63 所示，用于连接建立的 PDU 如表 3.26 所示。

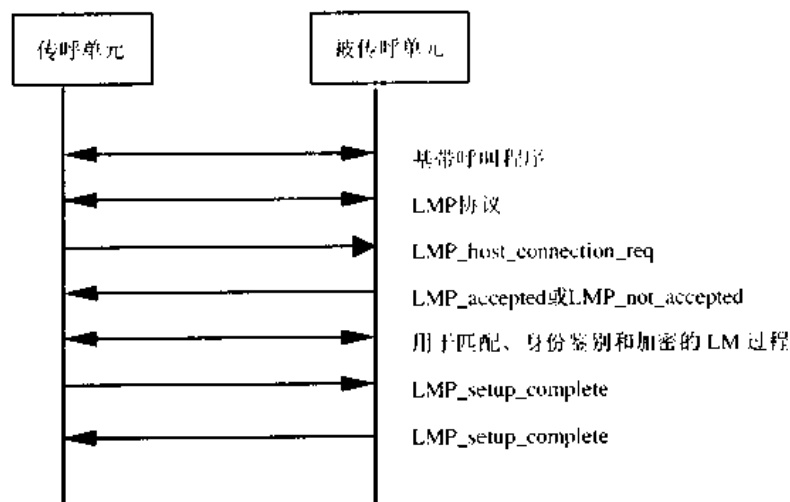


图 3.63 连接建立

表 3.26 用于建立连接的 PDU

M/O	协议数据单元	内容
M	LMP_host_connection_req	-
M	LMP_setup_complete	-

3.5 测试模式

LMP 具有支持不同蓝牙测试模式的 PDU。测试模式主要用于对蓝牙设备的鉴权和兼容蓝牙无线电和基带的测试。

3.5.1 激活和解除测试模式

通过向被测设备(DUT)发送 LMP_test_activate 来激活测试模式。DUT 通常作为从单元。链路管理器必须具有在任何时候都有接收该消息的能力。如果能在本地端就进入 DUT 的测试模式，那么被测设备(DUT)采用 LMP_accepted 作出应答，并进入测试模式，如图 3.64 所示。否则，被测设备(DUT)采用 LMP_not_accepted 作出应答，并且被测设备(DUT)保持正常的操作状态，如图 3.65 所示。LMP_not_accepted 的原因码应为 PDU not allowed。

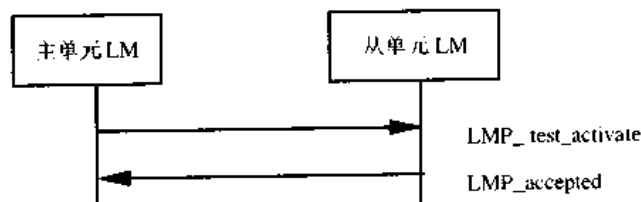


图 3.64 成功激活测试模式

测试模式能用两种方法来撤销。一种方法是：通过发送将测试环境设置为“退出测试模式”的 LMP_test_control 退出测试模式命令，从单元立即返回到与主单元保持正常连接的工作状态。另一种方法是：通过 LMP_detach 发送到被测设备(DUT)，从而同时实现对被

测设备的测试模式进行终止和连接。

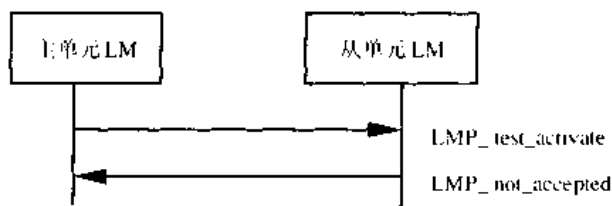


图 3.65 测试模式激活失败

3.5.2 测试模式的控制

当被测设备（DUT）进入测试模式时，可以向被测设备（DUT）发送协议数据单元 LMP_test_control，从而启动指定测试。该协议数据单元通过 LMP_accepted 确认。如果一个没有处于测试模式的设备收到 LMP_test_control，它就用 LMP_not_accepted 应答，应答消息的原因码为 PDU not allowed。分别如图 3.66 和图 3.67 所示。

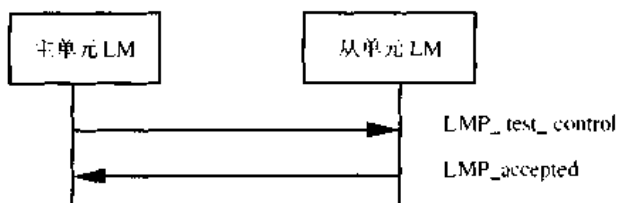


图 3.66 测试模式控制成功

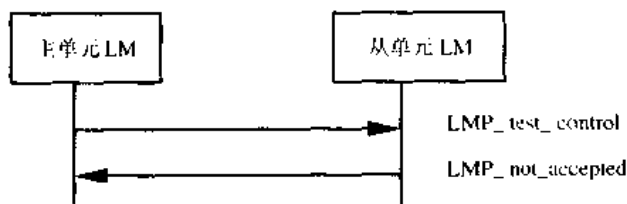


图 3.67 测试模式控制被拒绝

3.6 出错处理

如果链路管理器收到含有不能识别操作码的 PDU，就用 LMP_not_accepted 应答，并且 LMP_not_accepted 中含有原因码 unknown LMP PDU。并且返回的操作码参数同样也是不能识别的操作码。

如果在链路管理器中收到了含有非法参数的 PDU，就用 LMP_not_accepted 来作出应答，并且 LMP_not_accepted 中含有原因码 invalid LMP parameters（非法 LMP 参数）。

如果发现了最大响应时间超时或检测到链路丢失，等待应答的一方就可以认为该过程已经终止。

信道出错或发送方系统出错都会引起发送错误的消息。为了检测通信的最近状况，LM 应监测错误消息数量，一旦超过阈值就将其断开。该阈值应根据实际情况而设置成不同的值。

在链路双方的 LM 都对同一过程进行初始化且都没有成功的情况下，由于没有实时解

析 LMP PDU， 此时就会发生冲突。这时，主单元将通过发送含有原因码“LMP Error Transaction Collision”（即 LMP 出错处理冲突）的 LMP_not_accepted 消息，拒绝由从单元方尝试进行初始化的过程。然后，才终止由主单元方尝试初始化的过程。

第 4 章 逻辑链路控制和适配协议

4.1 概述

本章主要对逻辑链路控制和适配协议(L2CAP)做出描述, 主要针对协议状态自动机、分组格式及构成, 以及蓝牙测试和鉴权的测试接口做出详细阐述。

L2CAP 基于基带协议, 位于数据链路层中, 如图 4.1 所示。L2CAP 通过协议多路复用、分段重组操作和组概念, 向高层提供面向连接的和无连接的数据服务。L2CAP 允许高层协议和应用传输接收长达 64 KB 的 L2CAP 数据分组。

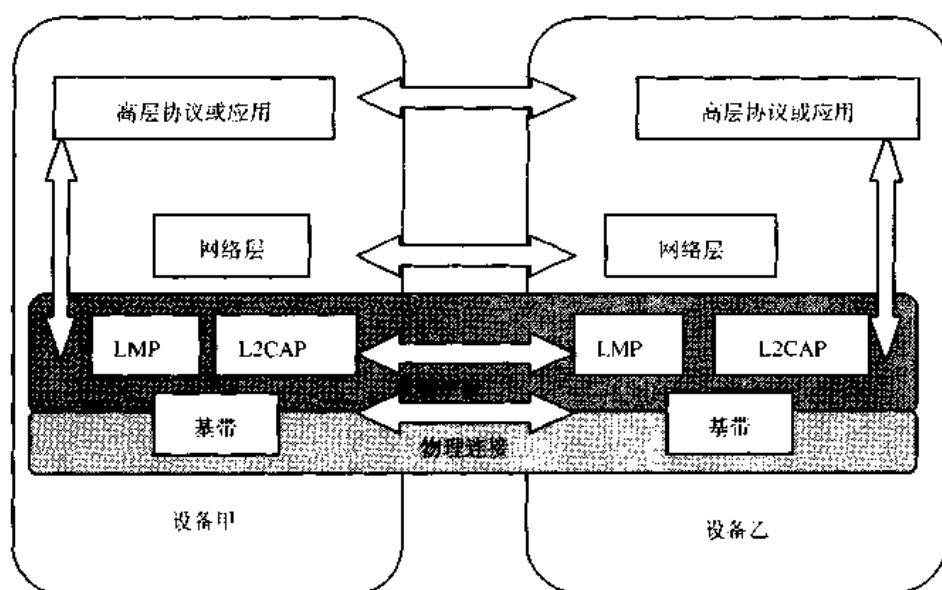


图 4.1 协议层内的 L2CAP

基带定义了两种链路类型：同步面向连接链路(SCO)和异步无连接链路(ACL)。SCO 链路采用保留带宽支持实时语音通信。ACL 链路支持最佳通信。L2CAP 规范仅定义 ACL 链路而不支持 SCO 链路。在 ACL 链路上禁止使用 AUX1 分组。该类型分组不支持数据完整性校验(无 CRC)。因为 L2CAP 在基带上依靠完整检查来保护传输的信息, 而 AUX1 分组却不能传输 L2CAP 分组。

ACL 有效载荷的格式, 如图 4.2 和图 4.3 所示, 分别表示单时隙分组头和多时隙分组头。它们的惟一差别是长度段的大小。分组类型(基带分组头中的一个段)用于区分单时隙分组与多时隙分组。

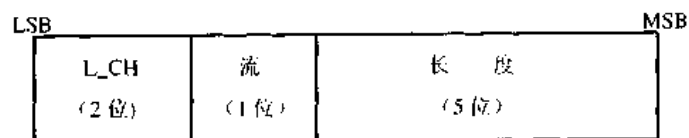


图 4.2 用于单时隙分组的 ACL 有效载荷头

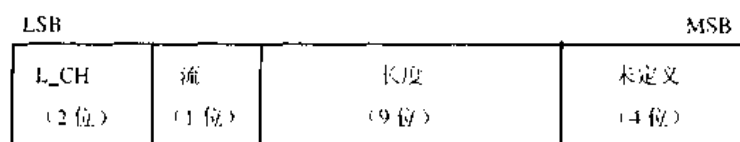


图 4.3 用于多时隙分组的 ACL 有效载荷头

2 位逻辑信道(L_CH)段用于区分 L2CAP 分组与链路管理器协议分组，其定义如表 4.1 所示。其余编码则保留以备将来使用。

表 4.1 逻辑信道 L_CH 段的内容

L_CH 信道	逻辑信道	信 息
00	RESERVED	保留
01	L2CAP	L2CAP包的延续
10	L2CAP	L2CAP分组起始位
11	LMP	链路管理协议

链路控制器(LC)作为一个基带执行实体，实施对 ACL 报文头的 FLOW（流）位的管理。当不再在 ACL 链路上进行 L2CAP 通信时，该位通常设置为 0(停止发送)。而 FLOW 位设置为 1 的 L2CAP 分组则意味着重新开始接收 L2CAP 分组流。

L2CAP 的功能要求包括协议复用、分段与重组(SAR)，以及组管理。图 4.4 具体说明 L2CAP。L2CAP 处于基带协议的上一层，并与蓝牙服务搜索协议(SDP)、RFCOMM 和电话控制(TCS)等其他通信协议具有通信接口。基带 SCO 链路常用作语音和电话应用的语音信道。经过分组的语音数据，如 IP 电话，通过使用 L2CAP 上层的通信协议进行发送。

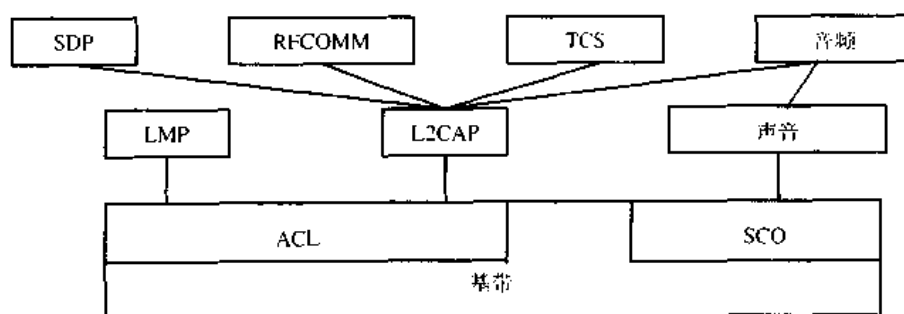


图 4.4 蓝牙协议体系结构中的 L2CAP

L2CAP 的必要协议要求包括简单和低拥塞。L2CAP 适用于具有计算资源有限的设备。由于蓝牙无线设备降低了功耗，L2CAP 应该不会过分耗费耗电。协议实施的内存要求也应保持最小化。协议复杂性应该适应由蓝牙支持的个人计算机、个人数字助理、数字蜂窝电话、无线耳机，游戏杆和其他无线设备。而且，协议应该能够达到相当高的带宽利用效率。

(1) 协议复用

L2CAP 应支持协议复用，因为基带协议不支持任何“类型”段，而这些类型段则用于标识要复用的更高层协议。L2CAP 必须能够区分高层协议，例如，服务搜索协议，RFCOMM，和电话控制。

(2) 分段与重组

与其他有线物理介质相比，由基带协议定义的分组在大小上受到限制。输出与最大基带有效载荷(DH5 分组中的 341 字节)关联的最大传输单位(MTU)限制了更高层协议带宽的有效使用，而高层协议要使用更大的分组。大 L2CAP 分组必须在无线传输前分段成为多个小基带分组。同样，收到多个小基带分组后也可以重新组装成大的单一的 L2CAP 分组。在使用比基带分组更大的分组协议时，必须使用分段与重组功能。

(3) 服务质量

L2CAP 连接建立过程，允许交换有关两蓝牙单元之间服务质量的信息。每个 L2CAP 设备必须监视由协议使用的资源并保证服务质量(QoS)的完整实现。

(4) 组

许多协议包含地址组的概念。基带协议支持匹克网，匹克网为能够使用同一时钟进行同步工作的一组设备。L2CAP 组概念可以实现在匹克网上的有效协议映射。如果没有组概念，为有效管理组，高层协议就必须直接与基带协议和链路管理器打交道。

L2CAP 协议是以下列假设为依据进行设计的：

- 使用链路管理器协议在两单元间建立 ACL 链路。基带提供数据分组的有序传输，但可能有个别分组损坏或重复。任两台设备之间只会有一条 ACL 链路。
- 基带通常提供全双工信道。但这并不是说所有 L2CAP 通信都是双向的。多点传送和单向通信(例如，视频)并不要求双工信道。
- 通过使用基带层提供的机制，L2CAP 提供了一条可靠的信道。当收到请求和重发数据时，基带通常要执行数据完整性校验，直到数据成功确认或发生超时。由于可能会丢失确认报文，所以甚至在数据成功发送后也会发生超时。基带协议使用长度为 1 位的序列号，该序列号用于删除重复发送的分组。由于所有广播的 L2CAP 数据分组的首段都以同一序列位为起始位，如果需要提供可靠传输，就应禁止使用基带广播分组。

4.2 主要操作

逻辑链路控制和适配协议(L2CAP)是以信道概念为基础的，它通过信道识别符引用每条 L2CAP 信道的端点。

4.2.1 信道标识符

信道标识符(CID)是表示逻辑信道本地端设备的名字。从 0x0001 到 0x003F 的标识符保留用于特定的 L2CAP 功能。空标识符(0x0000)则定义为一个非法标识符，并且不得用于目标端。可以根据实际应用目的和情况，以合适方式自由管理其余的 CID。但在本地设备与多个远端设备存在多个并发 L2CAP 信道的情况下，同一 CID 不得重新用作本地 L2CAP 信道端。

CID 的指定与特定设备有关，一台设备可以独立于其他设备指定 CID。这样，即使通过连接到一个本地设备的多个远程设备，将同一 CID 值指定给(远程)信道端，本地设备仍然能够将远端 CID 与每一不同的远程设备联系起来。

4.2.2 设备间操作

图 4.5 说明了 CID 在不同设备对等 L2CAP 实体间通信中的使用方式。面向连接的数据信道提供了两设备间的连接，而 CID 则用于标识信道的每一端。无连接信道限制数据向单

一方向的流动。这些信道用于支持一个信道“组”，在该信道“组”里发送端 CID 用于表示一个或多个远程设备。因此保留了一些 CID 以备将来特殊用途使用。信号信道是一个保留信道的实例。该信道用于创建和建立面向连接的数据信道，并可对这些信道的特性变化进行协商。L2CAP 实体必须支持信号信道。另一 CID 则保留用于呼入的无连接数据通信。在下面的例子中，CID 用于标识由设备#3 和设备#4 组成的组。而来自 ID 信道的数据则被发往保留用于无连接数据通信的远程信道。

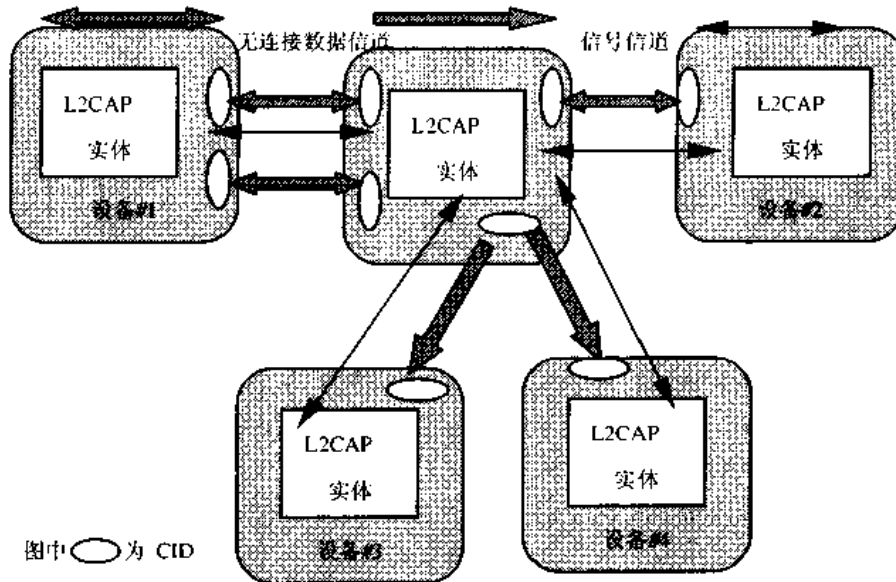


图 4.5 设备间信道

表 4.3 描述了不同信道及其主端和目标端标识。可以创建一条范围为 0x0040 到 0xFFFF 的“已分配”信道代表本地信道端。

表 4.3 信道标识类型

信道类型	当地的 CID	远程 CID
连接导向	动态分配	动态分配
无连接数据	动态分配	0x0002(固定值)
发信号	0x0001(固定值)	0x0001(固定值)

4.2.3 层间操作

L2CAP 的实施应遵循图 4.6 所示的总体体系结构，并可在高层协议和低层协议间传送数据。图中列出了一些 L2CAP 应用必须实现的服务。每个应用都必须支持一组用于 L2CAP 应用间通信的信令指令。L2CAP 应用还应准备从低层接受某类型的事件，并可向高层生成事件。事件如何在层间传递则根据实际应用情况而定。

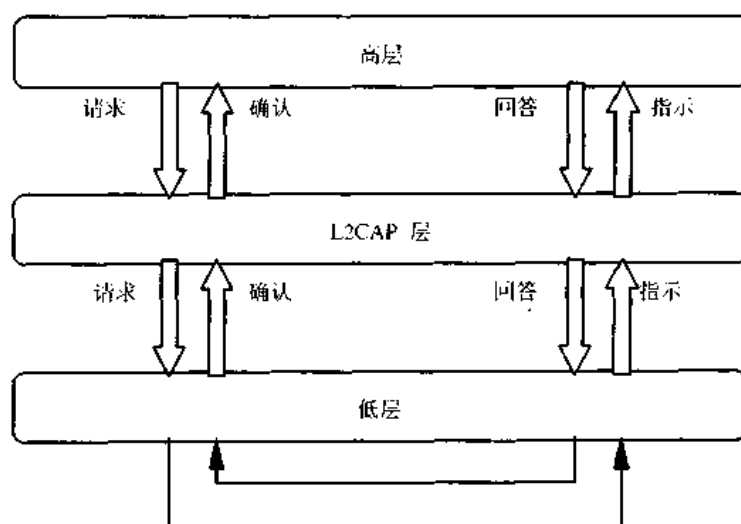


图 4.6 L2CAP 体系结构

4.2.4 分段和重组

分段和重组（SAR）操作用于通过支持最大传输单位（MTU）来提高传输效率。MTU 的长度大于最大的基带数据包。这样，就可以通过网络广播和传送高层协议分组降低拥塞。所有 L2CAP 分组都可以在基带分组基础上进行分段。L2CAP 协议并不执行任何分段和重组操作，但是其分组格式支持调整到更小的物理帧长度。L2CAP 发送出的（即，远程主机所接收的）MTU 把上层分组分为可通过主控制器接口（HCI）传送到链路管理器的“数据块”。在接收端，L2CAP 应用接收到来自 HCI 的“数据块”后，就可以利用 HCI 提供的来自分组头的信息，把这些“数据块”重组成 L2CAP 分组，如图 4.7 所示。

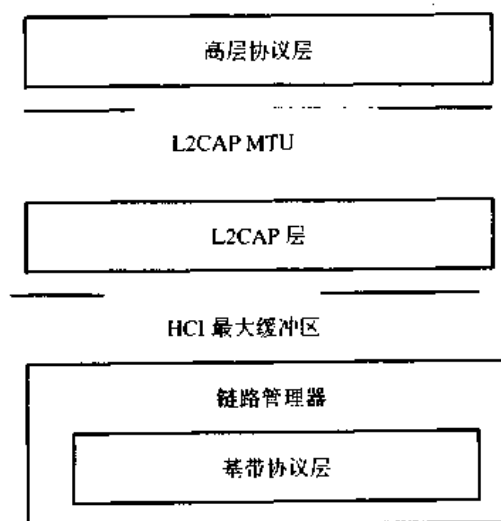


图 4.7 L2CAP SAR 变量

执行分组和重组只使用了很小的代价。位于基带分组有效载荷的第一个字节(也叫帧头)的两个 L_CH 位用于表示 L2CAP 分组的开始和附加部分。L_CH 为“10”表示 L2CAP 分组的第一段，而为“01”则表示它的其余部分。图 4.8 即是 SAR 的示例。

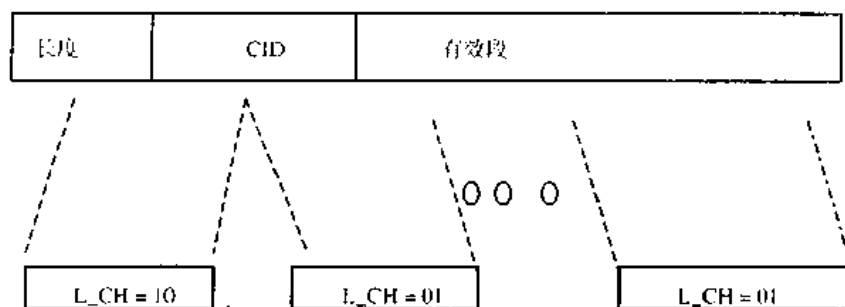


图 4.8 L2CAP 分段

1. 分段过程

通过使用专用服务接口可以输出 L2CAP 的最大传输单元(MTU)。高层协议负责在 MTU 区间内限制发往 L2CAP 层的分组大小。而 L2CAP 应用则将该分组分段成协议数据单元并送到下层。如果 L2CAP 直接位于基带的上一层, L2CAP 就应把分组分段成用于无线传输的基带数据分组。典型情况下, L2CAP 在主控制器接口上运行, 就应把整块“数据块”发送到主控制器, 再由主控制器将它们分段成为基带数据分组。在目的地址为同一单元的其他 L2CAP 分组发送以前, 所有与 L2CAP 分组相关联的 L2CAP 分组都必须先传送到基带。

2. 重组过程

基带协议按顺序发送 ACL 分组, 并利用 16 位 CRC 保证数据的完整性。基带也可以利用自动重复请求(ARQ)机制支持可靠连接。当基带控制器收到 ACL 分组时, 它可以在每个基带分组到达时通知 L2CAP 层, 也可以在接收缓冲区溢出或定时器失效之前收集一定数量的分组, 然后再通知 L2CAP 层。

L2CAP 应用必须使用 L2CAP 分组头里的长度段, 进行一致性校验, 并丢弃与长度段不匹配的 L2CAP 分组。如果不考虑信道可靠性, 将丢弃长度不合适的分组。如果考虑信道可靠性, L2CAP 必须通知上层信道已不可靠。通过具有无限刷新超时值定义可靠信道。

当存在高层 PDU 和 L2CAP 分组之间的一对一映射时, 分段和重组规则所用的段大小将取决于实际应用情况, 并且在发端和收端之间也可以不同。

4.3 状态机

本节描述 L2CAP 的面向连接信道的状态机, 定义了导致状态转换的状态和事件, 以及用于响应事件的动作。该状态机仅与双向 CID 有关, 它既不代表信号信道, 也不代表单向信道。

图 4.9 说明由 L2CAP 层应用所执行的事件和动作。客户和服务端分别代表请求的发起方和接收方。一个应用层次的客户可以发起, 也可以接受请求。两层间的接口(纵向接口)使用向高层提供服务的低层前缀, 如 L2CA。相同层实体之间的接口(横向接口)则使用协议前缀(把 P 加到协议层标识上), 如 L2CAP。来自高层的事件称作请求(Req), 而相应的答复则称为确认(Cfm)。来自低层的事件称为指示(Ind), 而相应的答复则称为应答(Rsp)。需要进一步处理的应答称为中间应答(Pnd)。用于确认和应答的概念表示肯定答复。否定答复

则应标有‘Neg’后缀，例如 L2CAP_ConnectCfmNeg。

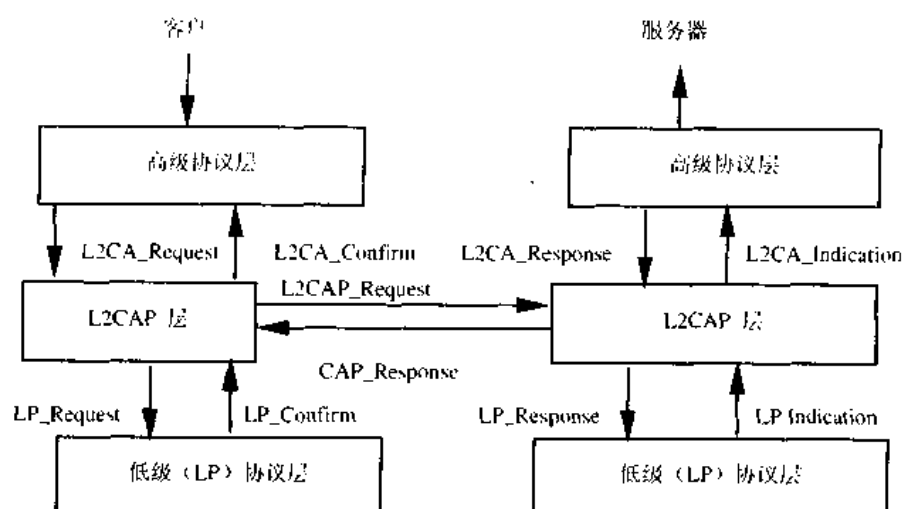


图 4.9 L2CAP 层互操作

一个来自高层动作请求的结果通常是相应的确认，无论对该动作的确认成功与否。而来自低层指示的结果却不总是相应的应答。如果指示知道本地的触发事件，则就会发生后一种情况。图 4.10 使用报文序列图(MSC)来解释事件的正常序列。两条外部垂直直线表示发起方(发出请求的设备)和接受方(应答发起方请求的设备)之间的 L2CA 接口。L2CA 接口的请求指令将导致发出协议定义请求。当协议向接受者传递请求时，远程 L2CA 实体将向上层协议发送指示。当接受方的上层协议应答时，应答将由协议打包并发回发起方。最后，再用确认报文把结果发回发起方的上层协议。

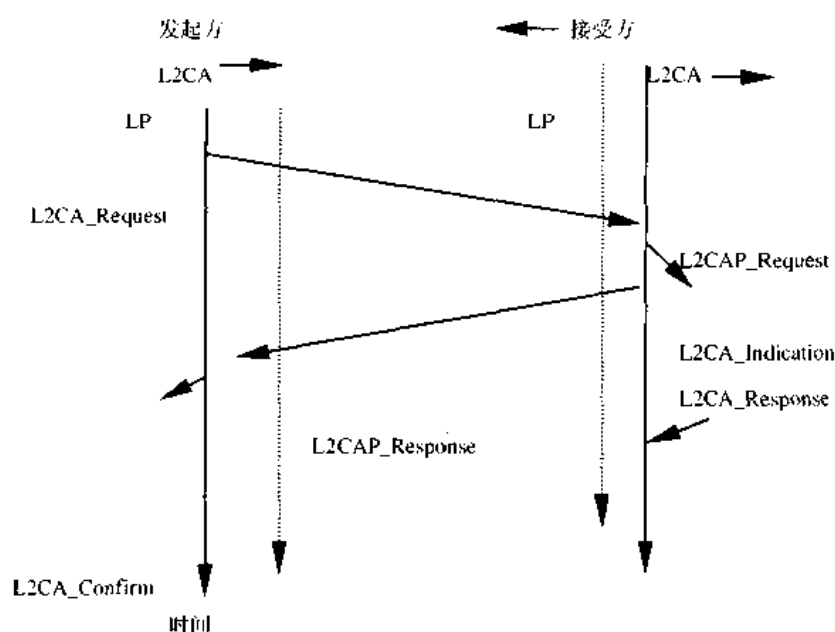


图 4.10 使用报文序列图解释事件的正常序列

4.3.1 事件

事件划分为 5 类：来自低层的指示和确认、来自高层的要求和应答、来自对等层的数据、来自对等层的信号请求和应答以及由定时器失效引起的事件。

1. 低层协议(LP)到 L2CAP 的事件

- LP_ConnectCfm

确认建立低层(基带)连接的请求(参见 LP_ConnectReq)。如果需要在建立物理链路时进行身份认证,就应包括对认证进行认证。

- LP_ConnectCfmNeg

确认建立低层(基带)连接的请求失败(参见 LP_ConnectReq)。这可能是因为设备联系不上,或请求被拒绝,或对 LMP 身份认证的认证失败。

- LP_ConnectInd

表示低层协议已成功建立连接。对于基带,是一条 ACL 链路。可以利用一个 L2CAP 实体跟踪物理链路信息。

- LP_DisconnectInd

表示低层协议(基带)已被 LMP 指令或一个超时事件关闭。

- LP_QoS Cfm

确认给定服务质量请求(参见 LP_QoSReq)。

- LP_QoS CfmNeg

确认给定服务质量请求失败(参见 LP_QoSReq)。

- LP_QoS ViolationInd

表示低层协议已检测到违反 LP_QoSReq 规定的 QoS 协定的情况。

2. 从 L2CAP 到 L2CAP 的信号、数据事件

在交换对应 L2CAP 信号 PDU 以后,由每个 L2CAP 实体生成从 L2CAP 到 L2CAP 的信号、数据事件。正如其他 L2CAP PDU 一样,该事件通过低层指示事件从低层进行接收。为阐述简洁,我们不对该过程进行详细说明,而且我们假定信号事件可以直接在 L2CAP 对等实体之间进行交换。

- L2CAP_ConnectReq

已收到一个连接请求报文。

- L2CAP_ConnectRsp

已收到连接应答报文,该报文指示连接已经建立。

- L2CAP_ConnectRspPnd

已收到连接应答报文,该报文指示远端已收到该请求并正在进行处理。

- L2CAP_ConnectRspNeg

已收到连接请求报文,该报文指示不能建立连接。

- L2CAP_ConfigReq

已收到配置请求报文,该报文指示远端希望参与有关信道参数的协商。

- L2CAP_ConfigRsp

已收到配置应答报文,该报文指示远端同意所有正在协商的参数。

- **L2CAP_ConfigRspNeg**

已收到配置应答报文，该报文表示远端不同意应答报文中的参数。

- **L2CAP_DisconnectReq**

已收到断开请求报文，信道必须启动连接断开过程。L2CAP 信道连接断开过程结束后，L2CAP 实体应将本地 CID 放回到“未分配”CID 库中。

- **L2CAP_DisconnectRsp**

已收到连接断开应答报文。收到该信号以后，接收方 L2CAP 实体可以将相应的本地 CID 返回到未分配的 CID 库中。由于该断开请求肯定成功，所以也就没有相应否定应答。

- **L2CAP_Data**

数据分组已收到。

3. 高层到 L2CAP 的事件

- **L2CA_ConnectReq**

请求与创建到远程设备的信道，该请求来自 L2CAP 的上层。

- **L2CA_ConnectRsp**

远程设备请求指示的应答，该应答来自协议高层，参见 L2CA_ConnectInd。

- **L2CA_ConnectRspNeg**

远程设备连接请求的否定应答，该应答来自协议高层(拒绝)，参见 L2CA_ConnectInd。

- **L2CA_ConfigReq**

请求(重新)设置信道，该请求来自协议上层。

- **L2CA_ConfigRsp**

对(重新)设置请求的应答，该应答来自协议高层，参见 L2CA_ConnectInd。

- **L2CA_ConfigRspNeg**

对(重新)设置请求的否定回答，该应答来自协议高层，参见 L2CA_ConnectInd。

- **L2CA_DisconnectReq**

请求信道立即断开，该请求来自协议高层。

- **L2CA_DisconnectRsp**

用于响应连接断开请求的指示，该响应来自协议上层，参见 L2CA_ConnectInd。由于没有相应的否定应答，因此必须执行连接断开指示。

- **L2CA_DataRead**

请求将收到的来自 L2CAP 实体的数据转发到高层，该请求来自协议高层。

- **L2CA_DataWrite**

请求在开放信道上将来自协议高层的数据转发到 L2CAP 实体。该请求来自协议高层。

4. 定时器事件

(1) RTX

当远端不对信令请求作出应答时，采用应答超时失效定时器终止信道。当发送信令请求到远程设备时，定时器开始工作。当收到应答时，该定时器失效。如果初始化方定时器失效，将发送一则完全相同的请求报文，否则将断开由请求所标识的信道。如果已发出同样的请求报文，则应将 RTX 超时设置为一个新值，该值至少应为原值的两倍。

在断开信道前，应用负责确定在 L2CAP 层次上执行的最大请求转发数。该决定应该基于信号连接的刷新超时。该决定应基于信令信道的刷新超时时间，超时时间越长，物理层次上执行转发的次数就越多。如果要减少 L2CAP 层次上的转发次数，应改进信道可靠性。例如，如果刷新超时无限大，那么在 L2CAP 层次上将不会执行转发操作。

该定时器值因实际应用情况而异，最小值为 1 秒，最大值为 60 秒。对于每一信令请求，包括每一应答请求，都存在一个 RTX 定时器。当收到应答，或物理链路丢失时，定时器在最后失效时消失。在定时器起始阶段和信道断开起始阶段（如果没有收到应答的话）之间的共用时间最大值为 60 秒。

(2) ERTX

当怀疑远端正在执行对请求信令的附加处理的时候，扩展应答超时终止定时器(ERTX) 用于替代应答超时终止定时器(RTX)。当远端应答说明请求为中间应答时，将启动该定时器。例如，当收到 L2CAP_ConnectRspPnd 事件时，定时器就会开始工作。当收到正式应答或物理链路丢失时，将停用该定时器。如果最初的定时器失效，将发送同一请求，或断开信道。如果发出同一请求，则 ERTX 定时器将消失，代之以一个新 RTX 定时器。然后，整个定时过程将如前述 RTX 定时器那样重新启动。

ERTX 定时器的值根据实际应用情况而不同，最小值为 60 秒，最大值为 300 秒。与 RTX 一样的是，对于每一收到中间应答的请求都必须至少有一个 ERTX 定时器，但对于每一请求最多只能有一个 RTX 或 ERTX。在该定时器起始阶段和信道断开起始阶段(如果应答没收到)之间的最长共用时间为 300 秒。

4.3.2 动作

动作分为五类：面向高层的确认和指示、面向低层的请求和应答、面向对等协议层的请求和应答、面向对等协议层的数据传输、定时器设置。

1. 从低层到 L2CAP 的动作

• LP_ConnectReq

L2CAP 请求低层协议创建连接。如果不存在到远程设备的物理链路，则应发送该报文到协议低层以建立物理连接。既然我们假定在两设备之间可能存在不止一条 ACL 链路，那么两设备之间的其他 L2CAP 信道必须共享同一基带 ACL 链路。

在处理请求之后，低层将返回 LP_ConnectCfm 或 LP_ConnectCfmNeg 报文，以指示是否已对该请求进行了确认。

• LP_QoSReq

L2CAP 请求协议低层以兼容一个特定 QoS 参数集。在处理该请求以后，低层将返回 LP_QoS Cfm 或 LP_QoS CfmNeg 以指示是否已对该请求进行了确认。

• LP_ConnectRsp

接受以前连接指示请求的主动回答，参见 LP_ConnectInd。

• LP_ConnectRspNeg

拒绝以前连接指示请求的否定应答，参见 LP_ConnectInd。

2. 从 L2CAP 到高层的动作

• L2CA_ConnectInd

标识已收到一个发自远程设备的连接请求，参见 L2CA_ConnectReq。

- L2CA_ConnectCfm

在收到来自远程设备的连接报文后，确认该连接请求已被接受，参看 L2CAP_ConnectReq。

- L2CA_ConnectCfmNeg

连接请求的否定确认，参见 L2CA_ConnectReq。对于该连接请求将触发一个 RTX 定时器失效事件参见 L2CA_TimeOutInd，不是一个消极连接应答，最后执行本动作。

- L2CA_ConnectPnd

确认已收到来自远程设备的连接应答（中间应答）。

- L2CA_ConfigInd

表示已收到从来自远程设备的配置请求。

- L2CA_ConfigCfm

在收到来自远程设备的配置应答后，确认已收到该配置请求，参见 L2CA_ConfigReq。

- L2CA_ConfigCfmNeg

配置请求的否定确认，参见 L2CA_ConfigReq。对于该连接请求将触发一个 RTX 定时器失效事件，参见 L2CA_TimeOutInd，而不是一个消极连接应答，最后再执行本动作。

- L2CA_DisconnectInd

表示已收到一个来自远程设备连接断开请求，或者是由于应答信令请求失败而导致远程设备断开。

- L2CA_DisconnectCfm

收到远程设备的连接断开应答后，确认断开请求已由远程设备处理，参见 L2CA_DisconnectReq。对于该连接请求将触发一个 RTX 定时器失效事件参见 L2CA_TimeOutInd，而不是一个消极连接应答，最后再执行本动作。一旦收到该事件，协议上层即获知 L2CAP 信道已被终止。没有对应的消极确认。

- L2CA_TimeOutInd

表示 RTX 或 ERTX 定时器已失效。本动作将在 L2CAP 放弃并发送 L2AC_DisconnectInd 以前多次执行，执行次数根据实际应用而定。

- L2CA_QoSViolationInd

表示违反服务质量协定。

4.3.3 信道操作状态

- CLOSED

在该状态下，不存在与 CID 关联的信道。本状态是不存在链路层次连接（基带）时的惟一状态。链路断开将强制其他状态转为 CLOSED 状态。

- W4_L2CAP_CONNECT_RSP

在该状态下，CID 代表本地终端，以及已经发送与本地终端有关的 L2CAP_ConnectReq 报文，并且该终端正在等待对应的 L2CAP_ConnectRsp 报文。

- W4_L2CA_CONNECT_RSP

在该状态下，存在远程终端，且本地 L2CAP 实体已收到 L2CAP_ConnectReq 连接请求报文。同时已发送 L2CA_ConnectInd 到协议上层，而正对收到的 L2CAP_ConnectReq 报

文进行处理的本地 L2CAP 实体，将等待相应应答。应答需要进行安全校验。

• CONFIG

在该状态下，已建立连接，但通信双方正在对信道参数进行协商。如果信道参数正在重新协商，也将进入该配置状态。进入 GONFIG 状态以前，由于要对数据通信参数重新协商，应暂停呼出的数据通信。但在远程终端进入 CONFIG 之前，应一直接收呼入的数据通信。

在该状态下，通信双方必须分发 L2CAP_ConfigReq 报文。如果只使用缺省值，则发送空报文，如果要对多数参数进行协商，则应发送多个报文以避免 MTU 限制，并进行增量协商。

从 CONFIG 状态迁移到 OPEN 状态需要双方都做好准备。当一个 L2CAP 实体收到对它的最后请求的肯定应答时，即表示它已做好准备。然后，该 L2CAP 实体将对远程设备的最后请求作出肯定应答。

• OPEN

在该状态下，已建立和配置连接，并且可以继续执行数据流。

• W4_L2CAP_DISCONNECT_RSP

该状态表示，正在关闭连接，而且 L2CAP_DisconnectReq 报文已发送。该状态表示正在等待对应应答。

• W4_L2CA_DISCONNECT_RSP

该状态表示，远程终端连接正在关闭，且已收到 L2CAP_DisconnectReq 报文。同时向协议上层发送 L2CAP_DisconnectInd 报文，以通知该 CID 的所有远程终端关闭。

4.3.4 事件到行为的映射

表 4.4 定义了在一定状态下，为响应事件而采取的动作。没在该表中列出的事件，或者没标记 N/C (没有变化)的动作，都假定为错误且丢弃。

数据输入输出事件仅由 OPEN 和 CONFIG 状态定义。不能在最初的 Configuration 状态期间接收数据，但当为了重新配置而再次进入 Configuraton 状态时，就可以接收数据。在其他状态下接收的数据将被丢弃。

表 4.4 L2CAP 信道状态机

事 件	当前状态	行 动	新 状 态
LP_ConnectCfm	关闭 (CLOSED)	如上所述标识物理链路，并初始化 L2CAP 连接	关闭 (CLOSED)
LP_connectCfmNeg	关闭 (CLOSED)	如下所述标识物理链路，并通过向上层发送 L2CA_ConnectCfmNeg 报文拒绝服务连接请求	关闭 (CLOSED)
LP_ConnectInd	关闭 (CLOSED)	如上所述标识链路	关闭 (CLOSED)
LP_DisconnectInd	关闭 (CLOSED)	如下所述标识连接	关闭 (CLOSED)
LP_DisconnectInd	除关闭 (CLOSED) 以外的任何状态	将 L2CA_DisconnectInd 报文发往上层	关闭 (CLOSED)
LP_QoSViolationInd	除打开 (OPEN) 以外的任何状态	丢弃	N/C
LP_QoSViolationInd	打开 (OPEN)	将 L2CA_QoSViolationInd 报文发往 上层，如果能够保证服务水平，则终止信道	打开 (OPEN) 或者

续表

事 件	当前状态	行 动	新 状 态
L2CAP_ConnectReq	关闭 (CLOSED)。 (CID 从免费的联 营分配来的)	将 L2CA_ConnectInd 发往上层。可选择 是否将 L2CAP_ConnectRspPnd 发往对等 协议层	W4_L2CA_ CONNECT_RSP
L2CAP_ConnectRsp	W4_L2CAP_CONNE CT_RSP	把 L2CA_ConnectCfm 报文发往上层。终 止 RTX 定时器	配置 (CONFIG)
L2CAP_ConnectRsp Pnd	W4_L2CAP_CONNE CT_RSP	将 L2CA_ConnectPnd 报文发往上层。终 止 RTX 定时器并启动 ERTX 定时器	N/C
L2CAP_ConnectRsp Neg	W4_L2CAP_CONNE CT_RSP	将 L2CA_ConnectCfmNeg 报文发往 上层。并将 CID 返回自由池。终止 RTX/ERTX 定时器	关闭 (CLOSED)
L2CAP_ConfigReq	关闭 (CLOSED)	将 L2CAP_ConfigRspNeg 报文发往对等 协议层	N/C
L2CAP_ConfigReq	配置 (CONFIG)	将 L2CA_ConfigInd 报文发往上层	N/C
L2CAP_ConfigReq	打开 (OPEN)	将 L2CA_ConfigInd 报文发往上层	配置 (CONFIG)
L2CAP_ConfigRsp	配置 (CONFIG)	将 L2CA_ConfigCfm 报文发往上层。终止 RTX 定时器。如果已收到 L2CAP_ConfigReq 报文并且已主动应答, 则进入 OPEN 状态, 否则仍保持 CONFIG 状态	N/C 或者 打 开 (OPEN)
L2CAP_ConfigRsp Neg	配置 (CONFIG)	将 L2CA_ConfigCfmNeg 报文发往上层。 终止 RTX 定时器	N/C
L2CAP_Disconnect Req	关闭 (CLOSED)	将 L2CAP_DisconnectRsp 报文发往对等协 议层	N/C
L2CAP_Disconnect Req	除关闭 (CLOSED) 以外的任何状态	将 L2CA_DisconnectInd 报文发往上层	W4_L2CA_ DISCONNECT_RSP
L2CAP_Disconnect Rsq	W4_L2CAP_ DISCONNECT_RSP	将 L2CA_DisconnectCfm 报文发往上层。 终止 RTX 定时器	关闭 (CLOSED)
L2CAP_Data	打开 (OPEN) 或配 置 (CONFIG)	如果收到完整 L2CAP 分组, 将 L2CA_Read 确认发往上层	N/C
L2CA_ConnectReq	关闭 (CLOSED)。 (CID 从免费的联 营分配来的。)	将 L2CAP_ConnectReq 报文发往对等协议 层。启动 RTX 定时器	W4_L2CAP_ CONNECT_RSP
L2CA_ConnectRsp	W4_L2CA_CONNEC T_RSP	将 L2CAP_ConnectRsp 报文发往对等协议 层	配置 (CONFIG)
L2CA_ConnectRsp Neg	W4_L2CA_CONNEC T_RSP	将 L2CAP_ConnectRspNeg 报文发往对等 协议层。将 CID 返回至自由池 (free pool)	关闭 (CLOSED)
L2CA_ConfigReq	关闭 (CLOSED)	将 L2CA_ConfigCfmNeg 报文发往上层	N/C
L2CA_ConfigReq	配置 (CONFIG)	将 L2CAP_ConfigReq 报文发往对等协议 层。启动 RTX 定时器	N/C
L2CA_ConfigReq	打开 (OPEN)	在合适的时候暂停数据传输。并将 L2CAP_ConfigReq 报文发往对等协议 层。启动 RTX 定时器	配置 (CONFIG)
L2CA_ConfigRsp	配置 (CONFIG)	将 L2CAP_ConfigRsp 报文发往对等协议 层。如果所有 L2CAP_ConfigReq 报文都收 到肯定应答, 则进入 OPEN 状态。否则仍 保持 CONFIG 状态	N/C 或者 打 开 (OPEN)

续表

事 件	当前状态	行 动	新 状 态
L2CA_ConfigRspNeg	配置 (CONFIG)	将 L2CAP_ConfigReqNeg 报文发往对等协议层	N/C
L2CA_DisconnectReq	打开 (OPEN) 或配置 (CONFIG)	将 L2CAP_DisconnectReq 报文发往对等协议层。启动 RTX 定时器	W4_L2CAP_DISCONNECT_RSP
L2CA_DisconnectRsp	W4_L2CAP_DISCONNECT_RSP	将 L2CAP_DisconnectRsp 报文发往对等协议层。将 CID 返回到自由池	关闭 (CLOSED)
L2CA_dataRead	打开 (OPEN)	如果完成有效载荷加载, 则将该有效载荷转发到 InBuffer	打开 (OPEN)
L2CA_dataWrite	打开 (OPEN)	将 L2CAP_Data 报文发往对等协议层	打开 (OPEN)
Timer_RTX	任何状态	将 L2CA_TimeOutInd 报文发往上层。如果最终失效, 则将 CID 返回到自由池, 否则重新发出请求	关闭 (CLOSED)
Timer_ERTX	任何状态	将 L2CA_TimeOutInd 报文发往上层。如果最终失效, 则将 CID 返回到自由池, 否则重新发出请求	关闭 (CLOSED)

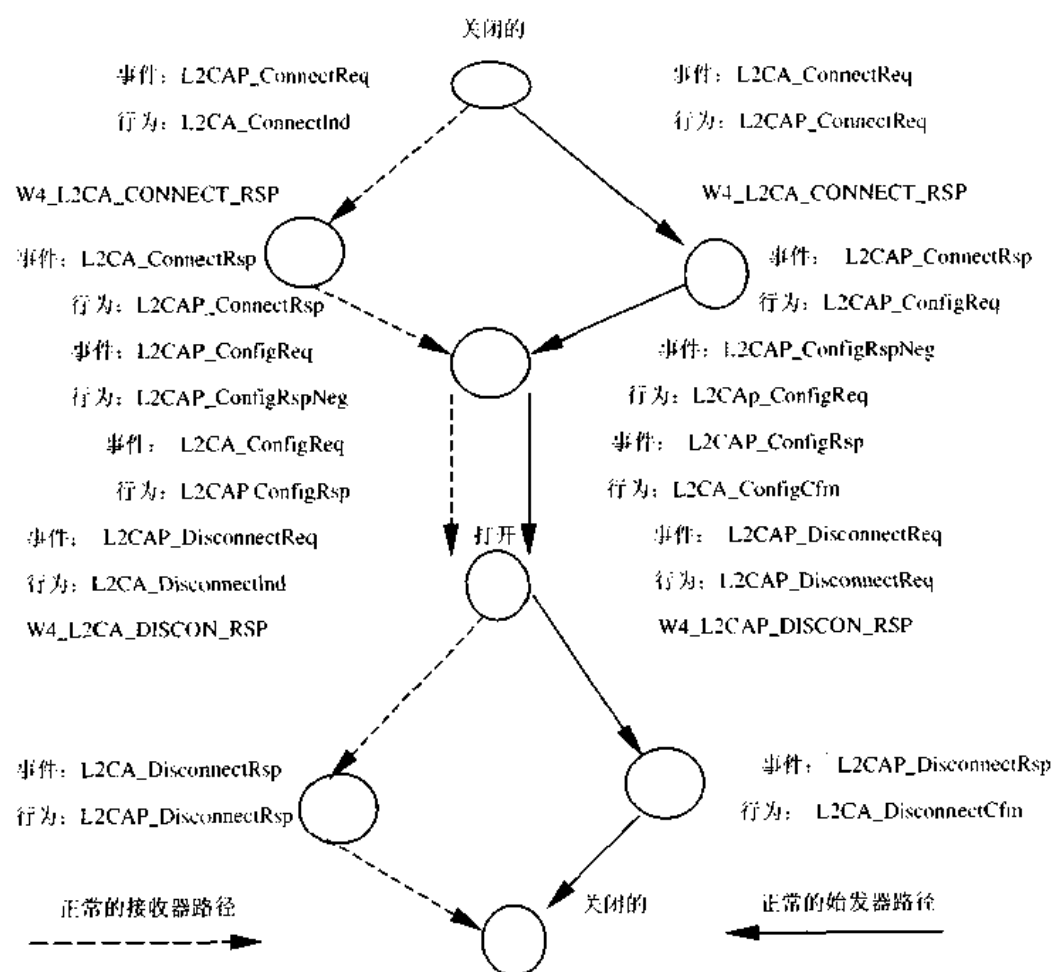


图 4.11 状态机实例

图 4.11 表示了一个简单状态自动机，以及由初始化方和接收方所采用的状态间典型转

换路径。该状态机显示由哪一事件导致状态转换，以及在状态转换发生时采取哪一动作。

4.4 数据分组格式

L2CAP 基于分组，但它实际上遵循的是一个基于信道的通信模型。一条信道代表远程设备上两 L2CAP 实体间的一数据流。信道可以是面向连接的，也可以是无连接的。

4.4.1 面向连接信道

图 4.12 说明了在面向连接信道内的 L2CAP 分组格式（也称之为 L2CAP PDU）。各段描述如下。

长度：2 个字节

长度指除了 L2CAP 报文头长度以外的信息有效载荷的大小，单位为字节。信息有效载荷的长度可长达 65535 个字节。

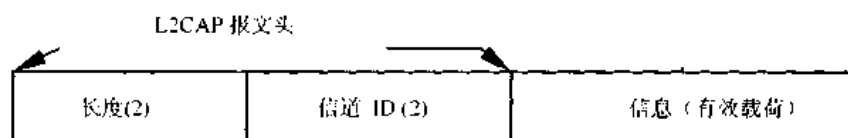


图 4.12 L2CAP 分组（各段以字节为单位）

信道 ID：2 个字节

信道 ID 用于标识分组的目标信道终端。信道 ID 取值与接收分组的设备相关。

信息：0 到 65535 个字节

信息分组是来自上层协议的有效载荷或者发送到协议上层的有效载荷。用于面向连接分组 MTU 的最小值将在信道设置期间进行协商。用于信令分组的 MTU 的最小值为 48 字节。

4.4.2 无连接数据信道

除了面向连接信道以外，L2CAP 也面向组信道。送到‘组’信道的数据将会同时送往组中的所有成员。由于组不提供服务质量，因此组信道通常并不可靠。这样 L2CAP 不能保证发往组的信息能够成功到达组中所有成员。如果需要进行可靠的组传输，它必须在协议高层中执行。

从理论上讲，面向组的数据传输会毫无例外地被传送到该组中所有成员，但因本地设备不能成为组成员之一，所以要由高层协议把任何数据流发回到本地设备上。这就意味着非组内成员也可以接收组传输，因此我们就可利用更高层次或链路层次的加密来支持私有通信。图 4.13 是无连接信息内的 L2CAP 分组格式，各段内容描述如下。

长度：两个字节

除 L2CAP 报文头的长度外，长度是信息有效载荷与 PSM 段长度的和。

信道 ID：两个字节

信道 ID (0x0002) 保留用于无连接通信。

协议/服务复用(PSM)：2 个字节(最小)

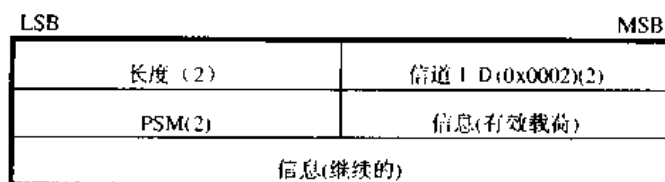


图 4.13 无连接分组（各段以字节为单位）

PSM 段以地址段 ISO 3309 扩展机制为基础。PSM 段的内容，即 PSM 值必须是奇数。也就是说，最低字节的最低位必须为 ‘1’。而且所有的 PSM 值的最高字节的最高位应等于 ‘0’。这样，PSM 段就可以扩充到 16 位以上。PSM 值定义主要针对 L2CAP，并由蓝牙 SIG 指定。

信息： 0~65533 个字节

该有效载荷信息将分发到组中所有成员。如果没有就其他值达成一致，应用应支持 670 字节的最小无连接 MTU(MTU_{cnl})。对于遵守某一使用特定无连接通信蓝牙标准的操作设备，其 MTU 将可小于 MTU_{cnl}。

L2CAP 组服务接口提供组管理机制，通过该机制可以创建组，给组增加成员，从组中删除成员。但不能够预先规定组。

4.5 信令

本节描述远程设备上两 L2CAP 实体间传递的信令指令。所有信令指令都将送至 CID 0x0001。L2CAP 应用必须能够确定发出该指令设备的蓝牙地址(BD_ADDR)。图 4.14 就包含信令指令的所有 L2CAP 分组通用格式做出描述。可在一个(L2CAP)分组中发送多条指令，该分组将送至 CID 0x0001。MTU 指令采用请求和应答方式。所有 L2CAP 应用都必须支持接收 MTU 小于 48 字节的信令分组。在没有对该应用是否支持更大信令分组进行测试的情况下，L2CAP 应用不得使用超过 48 字节的信令分组。

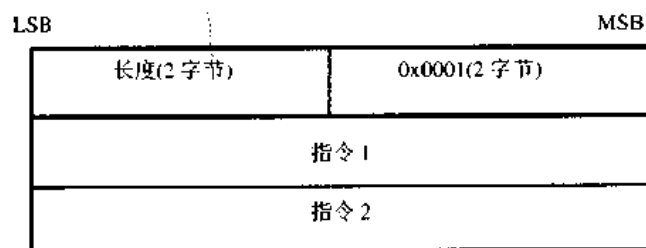


图 4.14 信令指令分组格式

图 4.15 表示所有信令分组的通用格式，各段描述如下。

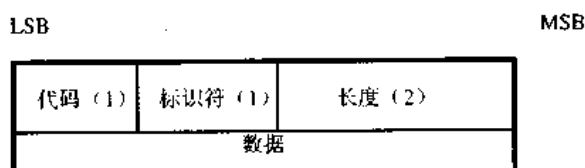


图4.15 指令格式

代码： 1 个字节

代码段长为一个字节，用于标识指令类型。在收到一个含有未知代码段的分组时，作为应答将发出一个指令拒绝分组。分配代码的当前值在最近的蓝牙“分配号码”文件中加以说明。所有代码在最左边的位置上以最高位指定。

标识符： 1 个字节

标识符段长为一个字节，用于请求与应答间的匹配。请求方设备设置该段，而应答设备在应答中使用相同值。每个最初的指令必须用不同的标识符。在使用该标识符的最初指令转换开始后的 360 秒里，不得重复使用标识符。RTX 或 ERTX 定时器终止时，如果重发同一请求，也应使用同一标识符。收到同一请求的设备也应以同一应答回答。含有非法标识符的应答则应被悄悄丢弃。信令标识符 0x0000 被定义为非法标识符，并且不得在任何指令中使用。

长度： 2 个字节

长度段长为 2 个字节，只用于以字节为单位表示的指令数据段的大小。也就是说，该数据段大小不包含代码、标识符和长度段在内。

数据： 任意个字节

数据段长度可变。可以使用长度段得到数据段长度。代码段决定数据段格式。

4.5.1 指令拒绝(代码 0x01)

为了响应含有未知指令代码的指令分组，或当不适于发送对应指令的时候，才可以发送指令拒绝分组。图 4.16 列出了分组格式。图中标识符应与含有未标识代码段的标识符相匹配。应用通常采用这些分组来应答未标识的信令分组。

当一个 L2CAP 分组里包含多条指令，并且该分组超过接收方的 MTU 时，将以一个指令拒绝分组应答。其标识符应与 L2CAP 分组的首条请求指令相匹配。如果只有应答被识别，那么将丢弃该分组。

指令拒绝分组的各段内容描述如下。

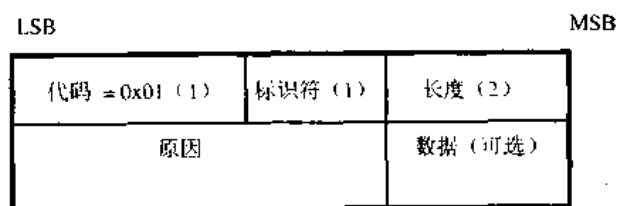


图 4.16 指令拒绝分组格式

长度： 0x0002 或更多的字节

原因： 2 个字节

原因段描述为什么拒绝请求分组。

数据： 任意字节

数据段的长度和内容取决于原因代码。如果原因代码是 0x0000，即“命令未被理解”，那么就不会使用数据段。如果原因代码是 0x0001，即“超出信令 MTU”，那么该两字节数据段将表示该分组接收方收到的最大信令 MTU。

如果指令指向一条错误的信道，那么将返回原因编码 0x0002。显然，由于该信道不存

在，则该信道为非法。指令拒绝的 4 字节长数据段将包含被争用信道的本地端和远端。远端从对应的被拒绝指令中获取。如果该拒绝指令只包含其中一个信道终端，则用无效的 CID 0x0000 代替另一端。

4.5.2 连接请求（代码 0x02）

连接请求分组用于创建一条介于两设备之间的信道。信道连接应在配置开始前建立。图 4.17 就连接请求分组做出说明。

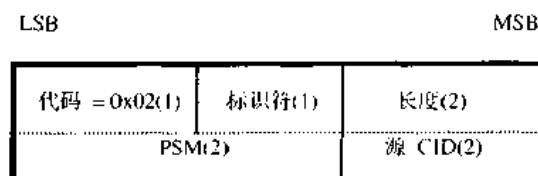


图 4.17 连接请求分组格式

长度： 0x0004 或更多字节

协议/服务复用(PSM)： 2 个字节(最小)

PSM 段长为 2 个字节(最小)。PSM 段结构以地址段的 ISO 3309 扩展机制为基础。所有 PSM 值都必须为奇数，也就是说，最低位字节的最低位必须为 ‘1’。而且，所有 PSM 值的最高字节的最低位应等于 ‘0’。这样，PSM 段将可以扩展到 16 位以上。PSM 值被分成两部分。第一部分的值由蓝牙 SIG 及其协议分配。第二部分的值则可以动态分配，并与服务搜索协议(SDP)一起使用。动态分配的值可以用于支持一个特定协议的多种执行版本，如 L2CAP 上层的 RFCOMM 或试验性协议原型。

源 CID(SCID)： 2 个字节

本地源 CID 长为 2 个字节，并可用于标识发送请求设备上的一个信道终端。一旦信道设置，则来自请求发送方的数据分组将被发往该 CID。在本节中，源 CID 表示发送请求和接收应答设备上的信道终端，而目标 CID 则表示接收请求和发送应答设备上的信道终端。

4.5.3 连接应答(代码 0x03)

当一个单元收到连接请求分组时，它必须发送一个连接应答分组。图 4.18 给出了连接应答分组格式，各段内容描述如下。

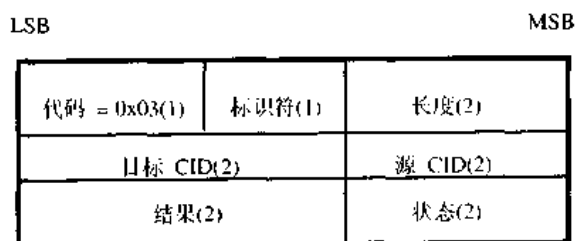


图 4.18 连接应答分组格式

长度： 0x0008 字节

目标信道标识符(DCID)： 2 个字节

该段包含发送应答分组设备上的信道终端。

源信道标识符(SCID)： 2 个字节

该段包含接收该应答分组设备上的信道终端。

结果: 2 个字节

结果段指示连接请求的结果。结果值 0x0000 表示连接请求成功,而非零值则表示连接请求失败。收到成功结果,即建立一条逻辑信道。如果结果段非零,那么就应忽略 DCID 和 SCID 段。

状态: 2 个字节

即中间应答,只有结果段对此做出定义,表示连接状态。

4.5.4 配置请求(代码 0x04)

配置请求分组用于在两个 L2CAP 实体间建立一个初始逻辑链路传输协定,如果合适,还可以重新对该协定进行协商。在重新协商会话期间,该信道上的所有数据通信都应在得到协商结果之前暂停。配置请求的每个配置参数只能与呼出数据通信和呼入数据通信两者之一有关。如果 L2CAP 实体在等待应答时收到配置请求,那么它也不能阻塞发送配置应答,否则该配置进程将会死锁。

如果没有需要协商的参数,那么也就没有必要插入任何选项,而 C-位应被清除。如果不接受缺省值,远程设备上的 L2CAP 实体必须就本文件中定义的所有参数进行协商,无论何时缺省值都不能被接受。任何丢失的配置参数都可被假定为它最近的接受值。即使可接受所有缺省值,都必须发送不含有选项段的配置请求分组。暗中接受的值可以是本文件中指定配置参数的任何缺省值,但这些缺省值并没有就设置中信道明确地进行协商。

每一配置参数都是单向的,并且都与配置请求发送方给出的方向有关。如果设备需要建立反向的配置参数值,而不是由配置请求给出的参数值,那么就要在与原连接请求相反的方向上发送新的配置请求,而该配置请求将含有希望得到的反向配置参数值。

在终止协商之前,如何确定仲裁信道参数所用的时间(报文)将根据实际情况而定,但不能超过 120 秒。

图 4.19 定义了配置请求分组的格式,其中各段内容描述如下。

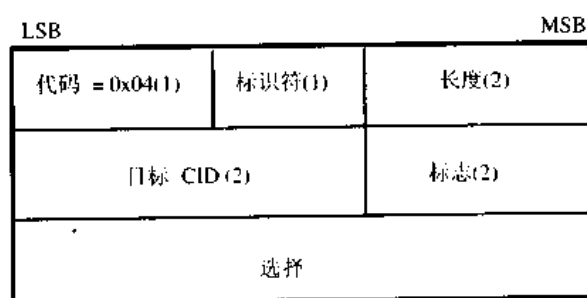


图 4.19 配置请求分组格式

长度: 0x0004 或更多字节。

目标 CID(DCID): 2 个字节。

该段包含接收该请求分组设备上的信道终端。

标志: 2 个字节。

图 4.20 表示了该两字节长的标志段,注意:左边是最高位。

C: 当设置为 1 时,将发送更多的配置请求。该标志表示正在发送更多的参数协商报文,在确认这些参数后,远程设备不应进入 OPEN 状态。如果参数超过 MTU_{sig},则有必要对配

置请求分组进行分段。其他标志保留，但应被清空。L2CAP 应用应忽略这些位。

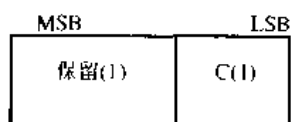


图 4.20 配置请求标志段格式

配置选项：

包括参数列表以及需要协商参数值。配置请求可以不包含任何选项（可以看作配置请求为空），并能用于请求一个应答。对于一个为空的配置请求，长度段应设置为 0x0004。

4.5.5 设置应答（代码 0x05）

为响应配置请求分组，必须发送设置应答分组。每一在配置应答中的配置参数值都反映了对已发出配置参数值的‘调整’。例如，如果配置请求中的配置参数与从设备 A 到设备 B 的通信有关，那么该配置应答发送方将根据从设备 A 到设备 B 的同一通信流，对该值作出调整。应答中的选项取决于结果段的值。图 4.21 对配置应答分组的格式进行了定义。

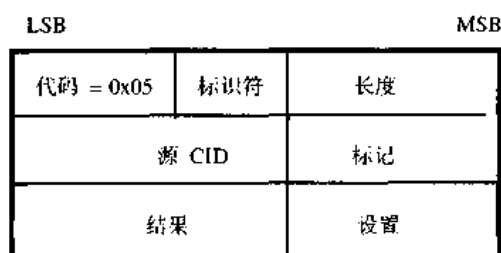


图 4.21 配置应答分组格式

长度：0x0006 或更多字节。

源 CID(SCID)：2 个字节

该段包含接受该应答分组设备上的信道终端。接收应答的设备必须检查标识符段是否与相应配置请求指令中的同一段匹配，SCID 是否与匹配于原 DCID 的本地 CID 相匹配。

标志：2 个字节

图 4.22 表示 2 个字节标记段，注意：左边为最高位。

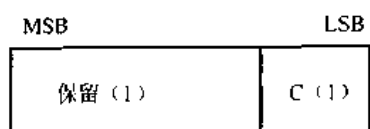


图 4.22 配置应答标志段格式

C：设置为 1 时，将发送更多配置应答。该标志表示应答参数是发送应答分组设备的一部分参数子集。其他标志应保留，但需清空。L2CAP 应用应忽略这些位。

结果：2 个字节

结果段指示是否可接受请求。

配置选项：

该段包含正在协商的参数表。

当发生参数不能被接受的情况(结果 = 0x0001)时, 肯定是返回参数中既包含了被拒绝参数也包含了可能被接受的值。任何丢失的配置参数都被假定为它们最近(互相)的接受值, 而且如果需要改变配置参数的话, 它们也可以包含在配置应答中。每一配置参数都为单向并与配置请求发送方的指示方向相关。如果配置应答发送方需要在反方向上建立一个配置参数值, 而不是原配置请求的指示方向, 那么就应在与原连接请求相反的方向上, 发送含有所希望的配置参数值的新配置请求。

当出现一个未知选项时(Result = 0x0003), 在应答中肯定包含了请求接受方所不理解的选项类型。注意由于不能理解而被跳过的请求选项, 不能包括在应答报文里, 而且也不能作为拒绝请求的惟一原因。

当应用决定终止协商之前, 花费在确定信道参数上的时间长短或报文数量。

4.5.6 断开请求(代码 0x06)

如果要终止一条 L2CAP 信道, 就需要发送连接断开请求分组, 并由断开连接应答分组进行确认。由于发往目的信道的所有其他 L2CAP 分组都将自动传递到协议上层, 则可以通过信令信道请求断开连接。图 4.23 表示了连接断开请求分组格式。在初始化连接断开过程以前, 接受方必须保证源 CIDs 和目标 CIDs 的匹配。一旦断开请求发出, 将丢弃在 L2CAP 信道中传输的所有呼入数据, 并且也不允许再对外发送新的数据。一旦接受信道的断开请求, 所有在该信道排队等待发送的数据也可能被丢弃。

LSB		MSB
代码 = 0x06(1)	标识符(1)	长度(2)
目标 CID(2)		源 CID(2)

图 4.23 连接断开请求分组格式

长度:0x0004 字节。

目标 CID(DCID): 2 个字节

该段用于标识在设备收到该请求时, 将被关闭的信道终端。

源 CID(SCID): 2 个字节

该段用于标识在设备发送该请求时, 将被关闭的信道终端。

SCID 和 DCID 与请求发送方有关, 并且必须与将被断开的信道相匹配。如果报文接收方没能识别 DCID, 那么就必须以含有‘无效 CID’结果码的 CommandReject 报文应答。如果接收方发现只有 DCID 匹配而 SCID 却不匹配, 则应丢弃该请求。

4.5.7 连接断开应答(代码 0x07)

为了响应每一连接断开请求, 应发送连接断开应答。连接断开应答的分组格式如图 4.24 所示。

长度:0x0004 个字节

目标 CID(DCID): 2 个字节

该段用于标识发送应答设备的信道终端。

源 CID(SCID): 2 个字节

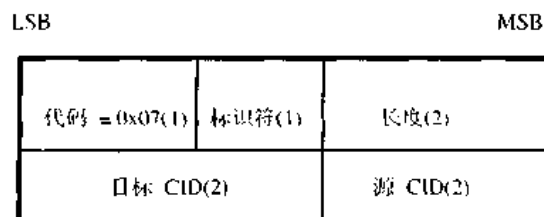


图 4.24 连接断开应答分组格式

该段用于标识接受应答设备的信道终端。

DCID 和 SCID (与请求发送器有关), 以及标识符段必须与对应的连接断开请求指令相匹配。如果 CID 不匹配, 应在接收方丢弃该应答。

4.5.8 回应请求(代码 0x08)

回应请求用于请求来自远程 L2CAP 实体的应答, 其格式如图 4.25 所示。该请求可以用于测试链路, 或利用可选数据段传递厂商指定信息。L2CAP 实体必须采用回应应答分组来对结构完整的回应请求分组进行应答。数据段可选, 并可根据实际情况而定。L2CAP 实体应该忽略该段内容。

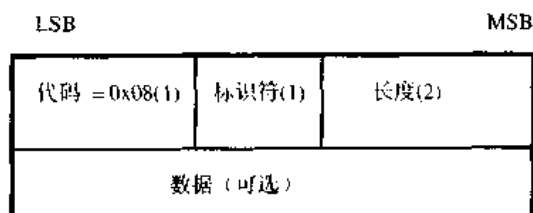


图 4.25 回应请求分组格式

4.5.9 回应应答(代码 0x09)

一收到回应请求分组就应发送回应应答分组, 其格式如图 4.26 所示。应答标识符必须

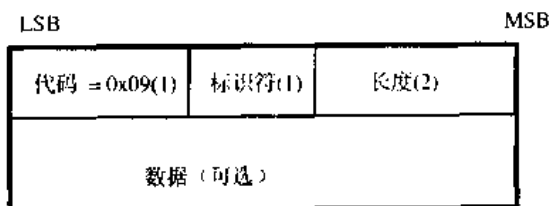


图 4.26 回应应答分组格式

与请求标识符匹配。可选的与根据应用而定的数据段可以包含于请求报文中数据段的内容, 也可以包含不同数据, 或者根本就不包含数据。

4.5.10 信息请求 (代码 0x0A)

信息请求用于从远程 L2CAP 实体请求应用指定信息。L2CAP 实体必须使用信息应答分组回应结构完整的信息请求分组。信息请求分组格式如图 4.27 所示。

长度: 0x0002 个字节

信息类型: 2 个字节

信息类型定义被请求的应用指定信息的类型。

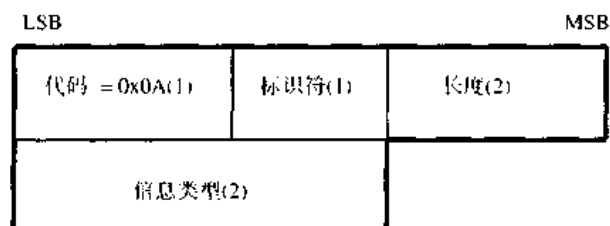


图 4.27 信息请求分组格式

4.5.11 信息应答（代码 0x0B）

一收到信息请求分组，就应发送信息应答。信息应答格式如图 4.28 所示。应答标识符必须与请求标识符匹配。可选的数据段可以包含请求数据段的内容，可以包含不同的数据，或根本就不包含数据。

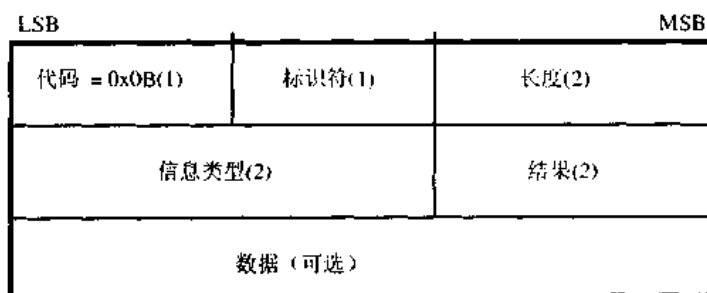


图 4.28 信息应答分组格式

信息类型: 2 个字节

在请求中发送的是同样的值。

结果: 2 个字节

结果包含请求成功与否的信息。如果结果是“成功”，那么数据段包含在下表说明的信息。如果结果为“不支持的信息”，则不会返回任何数据。

数据: 0 个以上个字节

数据段内容取决于信息类型。对于连接 MTU 请求，数据段包含远程实体的 2 字节可接受的无连接 MTU。

4.6 配置参数选项

选项是一种用于扩展不同连接要求协商能力的机制。选项以信息单元的形式传输，这些信息单元由选项类型、选项长度和一个以上的选项数据段组成，如图 4.29 所示。

类型: 1 个字节。

选项类型段定义正在设置的参数。如果没能识别出选项，则由选项类型的最高位决定要采取的动作。以下是该位取值的含义解释。

- 0 选择—— 必须识别该选项，表示拒绝配置请求；
- 1 选择—— 表示跳过该选项，继续处理。

长度：1 个字节

长度段定义选项有效载荷的字节数。无有效载荷的选项类型的长度段值为 0

选项数据：该段内容取决于选项类型。

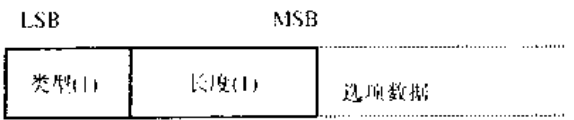


图 4.29 参数选项格式

4.6.1 最大传输单位(MTU)

该选项说明发送方能够接收的有效载荷大小。类型为 0x01，并且有效载荷长度为 2 个字节，如图 4.30 所示。有效载荷携带一两字节的 MTU 大小值作为惟一信息单元。由于所有 L2CAP 应用都能够支持最小 L2CAP 分组大小，MTU 不仅可以协商，而且还是发往远程设备的一个信息参数，该参数表示本地设备能够在本信道中接纳大于最小值的 MTU。只有在很少的情况下，远程设备才有可能在该信道中发送比本地设备给出的 MTU 大的 L2CAP 分组。然后，该配置请求将会收到一条否定应答。在该应答里，远程设备分组含要传输的 MTU 值。在这种情况下，本地设备是继续配置过程还是维护该信道，将因应用情况而定。

主动配置应答中的远程设备将包括用于本信道到本地设备的通信数据流的实际 MTU，该 MTU 为最小值(在 configReq 的 MTU，远程设备的输出 MTU 的大小)。当远程设备发送自己的配置请求时，将在该信道上，为相反方向通信数据流建立 MTU。

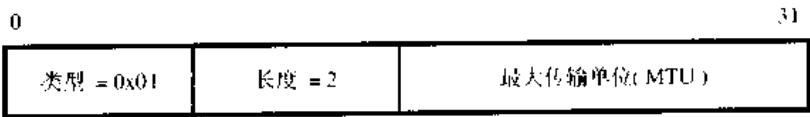


图 4.30 最大传输单位(MTU) 选项格式

最大传输单位(MTU)大小： 2 个字节

最大传输单位(MTU)段表示以字节为单位的最大 L2CAP 分组有效载荷，并且该 MTU 可以为请求发起方接收。MTU 是不对称的。而且请求发送方将定义可从信道接收的 MTU，其值与缺省值不同。L2CAP 应用必须支持至少 48 字节的 MTU 最小长度。缺省值为 672 字节。

如果选择缺省 MTU，则应以由两个基带 DH5 分组(2*341=682)减去基带 ACL 分组头(2*2=4)与 L2CAP 报文头(6)的和为基础。

4.6.2 刷新超时选择

本选项用于在放弃和刷新分组之前，起始链路控制器/链路管理器传输 L2CAP 段所耗时间通知接收方。其类型为 0x02， 并且有效载荷大小为 2 个字节，如图 4.31 所示。

刷新超时：

该值表示以毫秒为单位的时间单元。该值为 1 时，由于最小轮询间隔为 1.25ms，所以不能执行基带层次的重发操作。这也就是‘可靠信道’的特点。在这种情况下，链路管理器将持续重发一个段，直到物理链路丢失。该值为不对称值，并且如果它与 0xFFFF 缺省值不同，那么请求发送方将指定自己的刷新超时。

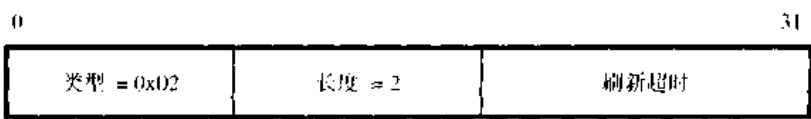


图 4.31 刷新超时

4.6.3 服务质量 (QoS) 选项

该选项说明与 RFC 1363 类似的流控制规范。如果没有协商 QoS 配置参数，那么链路就应假设为以下缺省参数。QoS 选项类型为 0x03。

当该选项包含于配置请求时，用于描述从发送请求设备到接收请求设备的呼出数据流。如果该选项包含于主动配置应答，从发送该应答的设备角度来看，它用于描述呼入数据流协定。当该选项包含于消极配置应答时，从发送应答设备的角度来看，它用于描述呼入数据流。

L2CAP 应用只需要支持‘最大化’服务；对于其他任何服务类型是可选的。最大化服务并不需要任何授权。如果没有将 QoS 选项置于请求中，必须假定为最大化服务。如果需要任何 QoS 授权，则必须发送 QoS 配置请求。

远程设备将取决于结果段值的信息置入配置应答。如果为了服务授权而请求，那么应答将包含在应答中的任何令牌参数的指定值(参见通信速率和令牌最大长度的描述)。如果该结果是“失败—不能接受该参数”，那么应答将可能包含一系列呼出流控制规范参数和参数值，这些参数值可以生成一个来自本地设备的新的连接请求，而本地设备必须能够为远程设备接受。明确定义的配置请求或暗含的配置参数中的引用值都将包含于配置应答。所有来自配置请求的丢失的配置参数将假定为最近接收的值。为了实现最大化和经授权的服务，当配置应答中包含 QoS 选项时，配置请求中将包含“无关项”值。

服务质量选项的说明规范如图 4.32 所示，各段内容描述如下。

0x03	长度 = 22	标志	服务类型
通信速率			
最大令牌长度 (字节)			
高峰带宽 (字节/秒)			
潜伏期 (微秒)			
迟滞变化			

图 4.32 服务质量说明规范

标志: 1 个字节
保留以备以后使用，并且必须设置为 0。

服务类型: 1 个字节

该段表示需要的服务层次。如果选择了缺省值‘最大化’,则其余段应由远程设备作为暗含要求。远程设备可能会忽略这些段,以努力满足该暗含要求,但不提供应答(在应答信息中省略了 QoS 选项),或者以要满足的设置来应答。

通信速率: 4 个字节

该段值用于表示通信速率,以字节/秒为单位。一个应用可以按照该速率连续发送数据。而突发数据可以以最大令牌长度发送(参看以下内容)。突发数据发送完毕后,应用必须限制其通信速率。0x00000000 值表示没有指定通信速率。该值为缺省值,它与通信速率并无不同。0xFFFFFFFF 值代表一张与最大可用通信速率相匹配的令牌。该值含义取决于与服务类型有关的语义。对于最大化服务,该值表示应用需要尽可能多的带宽。对于授权服务,在请求时该值表示最大可用带宽。

最大令牌长度: 4 个字节

该段值以字节为单位表示最大令牌长度。如果达到最大令牌长度,应用程序就必须等待,或者将数据丢弃。0x00000000 的值表示不需要最大令牌长度,该值为缺省值。0xFFFFFFFF 值表示与最大令牌长度相匹配的令牌。该值含义取决于与服务类型有关的语义。对于最大化服务,该值表示应用需要一个尽可能大的令牌长度。对于授权服务,该值表示请求时最大可用缓冲区。

带宽峰值: 4 个字节

该段值表示,以字节/秒为单位,来自应用的分组连续传输速率。中间系统可以利用该信息进行更有效的资源分配。缺省值 0x00000000 表示最大带宽是未知的。

延迟: 4 个字节

该段值是指发送端传输一位和首次无线传输之间的最大可接受延迟,以微秒为单位。对该数量的精确解释取决于服务类指定的授权水平。缺省值 0xFFFFFFFF 表示该值无关。

延迟变化: 4 个字节

单位为微秒,该段值是指分组的最大延迟时间和最小延迟时间之差。应用采用该值来确定接收方所需缓冲区的大小,以恢复原有数据传输模式。缺省值 0xFFFFFFFF 表示该值无关。

4.6.4 配置处理

对信道参数的协商包括以下 3 步。

- 将本地接受的非缺省参数通知远端;
- 使远端同意或拒绝这些参数值(包括缺省值);
- 在反方向上重复第(1)步和第(2)步动作。

该处理可抽象成为一条请求协商路径和应答协商路径。

1. 请求路径

请求路径就呼入 MTU、刷新超时和呼出流量规范进行协商。表 4.5 定义了配置选项,这些配置选项可置入配置请求报文及其语义中。

表 4.5 可置入请求的参数

参 数	说 明
最大传输单位 (MTU)	呼入MTU信息
FlushTo	呼出刷新超时
OutFlow	呼出流量信息

2. 应答路径

应答路径就呼出 MTU(即：远端呼入 MTU)、远端刷新超时和呼入流量规范(即：远端呼出流量规范)进行协商。表 4.6 定义了应答路径配置选项。如果请求报文中没有包含面向请求的参数(恢复为缺省值)，远端通过将推荐值包含在消极应答报文中，协商非缺省值。

表 4.6 允许在应答中使用的参数

参 数	说 明
最大传输单位 (MTU)	呼出MTU信息
FlushTo	呼入刷新超时
InFlow	呼出流量信息

3. 配置状态机

配置状态机图示了两条路径，如图4.33所示。在离开CONFIG状态并进入OPEN状态之前，两条路径必须闭合。请求路径要求本地设备接收主动应答以进入闭合状态，而应答路径要求本地设备发送主动应答以进入闭合状态。

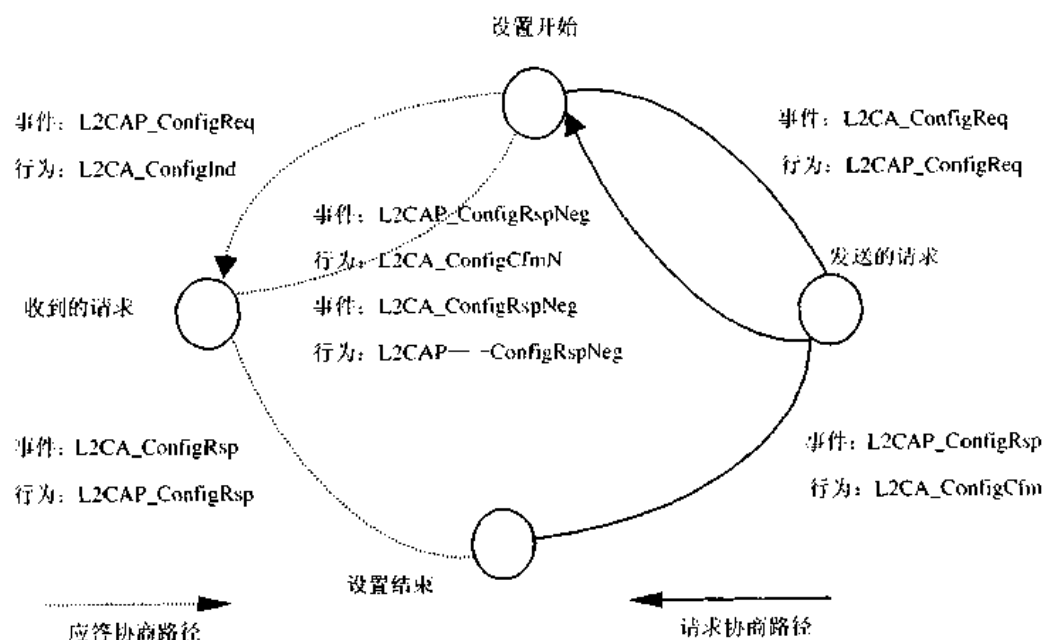


图4.33 配置状态机

4.7 小结

逻辑链路控制和适配协议(L2CAP)是在基带之上工作的两链路层协议之一。L2CAP 负责高层协议复用、提取 MTU、组管理, 以及将服务质量信息传送到链路层次。

定义信道支持协议复用。每一信道将采用多对一方式绑定于单一协议。复用信道可以被绑定在同一协议上, 但是一条信道不能绑定于多重协议。在信道上收到的每一 L2CAP 分组都指向合适的高层协议。

L2CAP 提取由基带协议使用的不同大小的分组。而且, 它可以使用低成本的分段重组机制, 支持达 64 K B 的分组。

组管理协议提供允许在匹克网成员与组之间有效映射的单元组概念。

组通信是无连接的和不可靠的。当组只由一对单元组成时, 组将提供无连接信道, 以代替 L2CAP 面向连接信道。

L2CAP 可以通过信道传输 QoS 信息, 并且提供授权控制, 以避免其他信道违反 QOs 协定。

第 5 章 服务搜索协议 (SDP)

5.1 引言

服务搜索协议(SDP)提供应用发现可用服务,以及确定可用服务特点的方法。

根据移动设备的邻频动态改变服务的蓝牙服务搜索,与传统的基于网络的服务搜索具有相当大的不同。服务搜索协议用于描述蓝牙环境的惟一特征。

下列性能被认定为服务搜索协议 1.0 版的必要条件。

- SDP 将为客户提供搜寻所需服务的能力;
 - SDP 允许基于服务类型搜索服务;
 - SDP 可以执行服务浏览,而不用预先知道服务特征;
 - SDP 将提供一种方法来搜索新的服务。当设备进入客户设备邻频或处于客户邻频的设备的新服务可用时,这些服务才可用;
 - SDP 将提供一种机制来确定在设备离开客户设备邻频时,或者当处于客户设备邻频的设备上的新服务不可用时,设备在何时变为不可用;
 - SDP 将提供对服务、服务类型和属性的惟一标识;
 - SDP 应允许一方设备上的客户在另一方设备上搜索服务,而不用查询第三方设备;
 - SDP 应适于在不太复杂的设备上使用;
 - SDP 应提供一种可增量搜索设备所提供服务信息的机制。这样将可以减少必须交换的数据量,以便确定客户是否不需要某一特定服务;
 - SDP 应可通过中介代理支持服务搜索信息缓存,以提高搜索进程的速度或效率;
 - SDP 应可独立传输;
 - SDP 将把 L2CAP 作为其传输协议时工作;
 - SDP 应允许搜索和使用能够提供对其他服务搜索协议访问的服务;
 - SDP 应支持和定义新的服务,而不需要向中心授权机构申请注册。
- 蓝牙 SIG 认为下列能力与服务搜索有关。
- SDP 1.0 不能存取服务。它仅仅提供对服务有关信息的访问;
 - SDP 1.0 不提供服务中介;
 - SDP 1.0 不提供对服务参数的协商;
 - SDP 1.0 不提供对服务使用的计费;
 - SDP 1.0 不向客户提供控制或改变服务操作的方法;
 - 在服务或服务有关信息不可用时,SDP 1.0 不提供事件通知;
 - 当服务属性被修改时,SDP 1.0 不提供事件通知;
 - SDP 1.0 不提供服务汇总、服务注册等服务代理功能。

5.2 SDP 概述

5.2.1 客户服务器交互

图 5.1 是客户服务器交互的一般性框图。

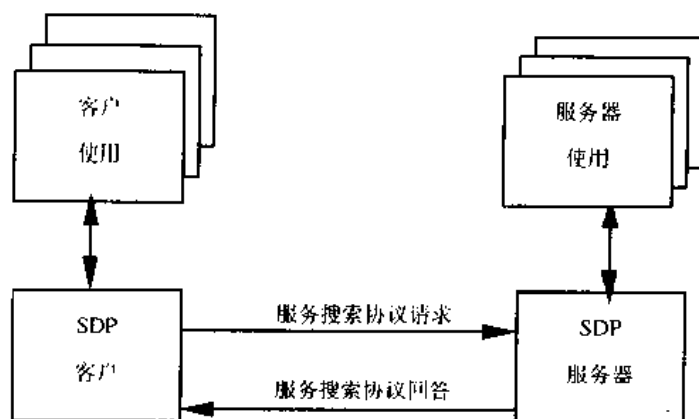


图 5.1 客户服务器交互框图

服务搜索机制为客户应用提供了搜索服务器应用所提供服务及其属性的一种方法。服务属性包括提供的服务类型，以及使用这些服务所需要的机制或协议。就服务搜索协议（SDP）而言，图 5.1 所示配置可以简化为图 5.2 所示配置。

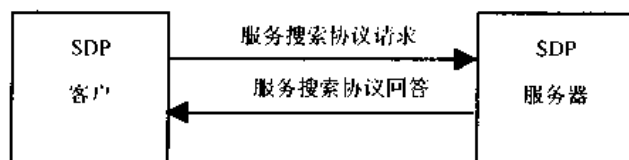


图 5.2 图 5.1 的简化

服务搜索协议(SDP)包括 SDP 服务器与 SDP 客户之间的通信。服务器保持一张描述服务器有关服务特征的服务记录表。每一服务记录都包含一项服务信息。客户可以通过发送 SDP 请求,从由 SDP 服务器维护的服务记录中检索信息。

如果客户或与客户有关的应用决定使用一种服务,它就必须建立一个到服务提供方的连接。SDP 提供一种搜索服务及其属性(包括相关服务访问协议)的机制,但它不提供使用这些服务的机制(如发送服务访问协议)。

蓝牙设备与 SDP 服务器应一一对应,且存在一个最大的对应数(如果一台蓝牙设备仅仅充当一个客户,那它就不需要 SDP 服务器。)。一台蓝牙设备既可以作为一个 SDP 服务器也可以同时作为一个 SDP 客户。如果一台设备的多个应用程序同时提供服务,那么可以将 SDP 服务器作为处理服务相关信息请求的服务提供方。

同样,多个客户应用也可以作为一个 SDP 客户,代表客户应用去查询服务器。

对于 SDP 客户可用的 SDP 服务器组,可以基于服务器到客户的邻频实现动态改变。当服务器可用时,应通过 SDP 以外的方法通知潜在客户,以使客户能够利用 SDP 查询服务器相

关服务。但是,当一个服务器离开邻频或者由于某些原因不可用时,将无法通过服务搜索协议显式通知客户。然而,客户可以使用 SDP 轮询服务器。如果服务器长时间无应答,客户就可推断出服务器不可用。

5.2.2 服务记录

服务是任何一个能够提供信息、执行动作或代表另一实体控制资源的实体。服务可以以软件、硬件或硬软件混合的形式执行。

由 SDP 服务器所维护服务的所有信息都包含于一条服务记录中。该服务记录全部由一张服务属性表组成,如图 5.3 所示。

服务属性 1
服务属性 2
服务属性 3
...
服务属性 N

图 5.3 服务记录

服务记录句柄是一个专门用于惟一标识 SDP 服务器内每一服务记录的 32 位的值。必须注意的是,每个句柄在每个 SDP 服务器内都是惟一的。如果 SDP 服务器 S1 和 SDP 服务器 S2 同时包含同一服务记录(代表同一服务),用于引用该相同服务记录的服务记录句柄则是完全独立的。如用于引用 S1 设备上服务的句柄指向 S2,那么该句柄并无任何含义。

当向 SDP 服务器增加或删除服务记录时,服务搜索协议并不提供通知客户的机制。当与服务器建立一条 L2CAP (逻辑链路控制与适配协议)连接时,从服务器获取的服务记录句柄将保持有效,除非它所代表的服务记录被删除。如果某一服务从服务器中删除,在从 L2CAP 连接获取服务记录句柄期间,又采用该服务记录句柄向服务器提出进一步请求,将会引起表示该服务记录句柄非法的出错应答。SDP 服务器应确保在保持 L2CAP 连接期间,不会重用任何服务记录句柄。

注意:当 ServiceDatabaseState 属性保持不变时,服务记录句柄在连续 L2CAP 连接中也仍然保持有效(参见属性定义里的 ServiceRecordState 和 ServiceDatabaseState 属性)。

存在一个适用于所有 SDP 服务器的服务记录处理句柄。该服务记录句柄值为 0x00000000,并且该句柄也就是代表 SDP 服务器的服务记录的句柄。该服务记录句柄包含 SDP 服务器属性及其支持的协议。例如,它的属性之一是服务器支持的 SDP 协议版本列表。服务记录句柄值 0x00000001~0x0000FFFF 保留使用。

5.2.3 服务属性

服务属性用于描述某一服务的一个特征。服务属性的实例如表 5.1 所示。

所有服务记录通用属性定义,可参见通用属性定义,服务供应方也可以定义其自己的服务属性。服务属性由两个部分组成:属性 ID 和属性值,如图 5.4 所示。

1. 属性 ID

属性 ID 是 16 位无精度整数,可用于在服务记录中将不同服务属性区分开来。属性 ID 也能够区分与属性值关联的不同语义。

表 5.1 服务属性实例

ServiceClassIDList	用于标识由服务记录代表的服务类型。也就是说，该服务属性也就是一个类列表，该服务仅是某一类的一个实例
ServiceID	惟一标识某一服务实例
ProtocolDescriptorList	表示可用于使用某一服务的协议栈
ProviderName	提供某一服务的个人或组织的文本名
IconURL	表示被某一图标引用的 URL，该图标可用于代表某一服务
ServiceName	含有服务名称的文本串
ServiceDescription	描述服务的文本串

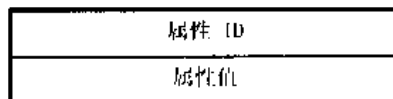


图 5.4 服务属性组成

服务类型定义是指服务类的属性 ID，并为关联于每一属性 ID 的属性值给出一定含义。例如，假设服务类 C 指出：与属性 ID 12345 关联的属性值是一个包含服务创建日期的文本串。并且，服务 A 是该服务类 C 的一个实例。如果服务 A 的服务记录包含一个属性 ID 为 12345 的服务属性，那么该属性值必然就是一个包含服务 A 创建日期的文本串。然而，非服务类 C 实例的服务可给 ID 12345 分配一个不同含义。

所有属于某一给定服务类将为每一特定属性 ID 分配同样的含义（参见服务类型）。

在服务搜索协议里，属性 ID 通常代表一个数据元。参见 5.3 节数据表示。

2. 属性值

属性值也就是可变长度段，其含义由与其关联的属性 ID 和包含该属性 ID 的服务的服务类决定。在服务搜索协议中，属性值代表一个数据元。通常，任一类型数据元都可以允许作为属性值，并受到服务类型定义的限制。该服务类型定义将属性 ID 指定给该属性，并为该属性值指定一个含义。参见服务属性定义的属性值实例。

5.2.4 服务类

每一服务都是服务类的一个实例。服务类定义提供对所有包含于服务记录中属性的定义，而这些服务记录就代表一个类实例。每一属性定义将给出该属性 ID 的数字值、该属性值的用法及其格式。服务记录包含服务类的专用属性，以及用于所有服务的通用属性。

每一服务类将指定一个惟一标识符。该服务类标识符包含于 ServiceClassIDList 属性的属性值，并表示为 UUID。由于服务记录中的某些属性的格式和含义依赖于服务记录的服务类，因此 ServiceClassIDList 属性非常重要。在使用任一类指定属性之前，应检查或验证它们的值。由于服务记录的所有属性必须遵循所有的服务类，包含于 ServiceClassIDList 属性中的服务类标识符也与此有关。特别要说明的是，每一服务类都是另外一类的子类，该父类的标识符包含在 ServiceClassIDList 列表中。服务子类定义与其父类不同，子类中包含其他子类特定的属性定义。ServiceClassIDList 属性中的服务类标识符，按照从底层类到高层类的顺序一一列出。

在定义本身是另一服务类子类的新服务类时，该新类将保留父类定义的所有属性。同

时，也将定义专用于新服务类的其他属性。换句话说，向已有服务类的某些实例添加新属性的机制将创建一个新的服务类，该服务类是已有服务类的子类。

例如，彩色打印机可遵循四个服务类定义，并具有一个含 UUID 的 ServiceClassIDList，包括以下服务类 ServiceClasses：

- DuplexColorPostscriptPrinterServiceClassID;
- ColorPostscriptPrinterServiceClassID;
- PostscriptPrinterServiceClassID;
- PrinterServiceClassID。

注意：本例仅仅是一个描述性例子，并不是实际的打印机类层次结构。

5.2.5 服务搜索

一旦 SDP 客户具有服务记录句柄，它将很容易地请求特定属性值，但该客户如何才能获取用于服务记录的服务记录句柄呢？服务搜索事务允许客户基于服务记录包含的属性值，搜索特定服务记录的服务记录句柄。

本协议并未提供基于强制值的服务记录搜索能力。相反，只提供了搜索值为通用定位符“UUID”的属性。可用于搜索服务的重要服务属性表示为 UUID。

1. UUID

UUID 是经授权可在所有时空中保持惟一的通用定位符。UUID 可以分布的方式独立创建，而且不需要指定 UUID 的中心注册机构。UUID 值长度为 128 位。

为了减少存储压力，并便于 128 位 UUID 值的转换，UUID 值段已经进行预先分配。在已分配段的第一个 UUID 作为蓝牙基 UUID，具有来自蓝牙号码分配文件的值 00000000-0000-1000-7007-00805F9B34FB。在已分配段的 UUID 值都具有 16 位或 32 位的别名。这些别名常被称为 16 位或 32 位 UUID。但每一别名实际上都代表一个 128 位 UUID。

可以通过一个简单的数学运算计算 16 位或 32 位 UUID 的全部 128 位值。

128 位值 = 16 位值 * 2^{96} + 蓝牙基 UUID

128 位值 = 32 位值 * 2^{96} + 蓝牙基 UUID

通过对 16 位值进行零扩展，将 16 位 UUID 转换成为 32 位 UUID 格式。另外一个转换方法是将 16 位值加到所有位都为零的 32 位 UUID。

注意：可以直接对两个 16 位 UUID 或 32 位 UUID 或 128 位 UUID 进行比较，如果要对不同大小的 UUID 进行比较，短 UUID 必须在比较前转换成为长 UUID 格式。

2. 服务搜索模式

服务搜索模式是一个用于定位匹配服务记录的 UUID 表。如果服务搜索模式中的 UUID 包含于任一服务记录属性值，服务搜索模式应可匹配一条服务记录。UUID 不必包含于任何特定属性，也不必在服务记录中以任何特定顺序排序。如果服务搜索模式包含的 UUID 在服务记录属性值中，构成了 UUID 子集，服务搜索模式将匹配一条服务记录。只有在服务搜索模式包含不止一个包含于服务记录属性值的 UUID 时，服务搜索模式才不与服务记录匹配。

必须注意：一个合法服务搜索模式必须含有至少一个 UUID。

¹ UUID 的格式由国际标准化组织 1996 年在 ISO/IEC 11578 里定义。

5.2.6 服务浏览

通常，一个客户基于某些服务特性（由 UUID 表示）搜索服务。然而有时候，也可以搜索由 SDP 服务器服务记录进行描述的服务类型，这些服务器服务记录不含服务预定义信息。寻找服务的过程叫做浏览。在 SDP 中，浏览服务机制以各服务类共享属性为基础。该属性称为 BrowseGroupList 属性，属性值包含一张 UUID 表。每一 UUID 采用出于浏览目的而关联的服务代表一个浏览组。

当客户需要浏览 SDP 服务器的服务时，它将创建包含 UUID 的服务搜索模式，而该 UUID 表示根浏览组。所有可以在顶层浏览的服务，可以通过把根浏览组的 UUID 作为 BrowseGroupList 属性值，从而构成根浏览组。如果 SDP 服务器没有多少服务，它所有的服务将被放在根浏览组里。然而，由 SDP 服务器提供的服务，可以通过在根浏览组的下层定义其他浏览组，以浏览组层次结构形式组织起来。这些下层浏览组都有服务记录，采用 BrowseGroupDescriptor 服务类进行描述。

浏览组描述符服务记录可以通过其组 ID 属性定义新浏览组。为了获取包含于可浏览的新定义浏览组的服务，必须可以对新浏览组的浏览组描述符服务记录进行浏览。由浏览组描述符服务记录提供的可浏览服务的结构层次，允许对 SDP 服务器提供的服务进行增量浏览。在服务包含许多服务记录时，该服务层次非常有用。

1. 服务浏览层次举例

图 5.5 所示是一个描述浏览组描述符用法的虚拟服务浏览层次结构。浏览组描述符服务记录标识为 G，其他服务记录标识为 S。表 5.2 列出了执行浏览层次所需的服务记录和服务属性。

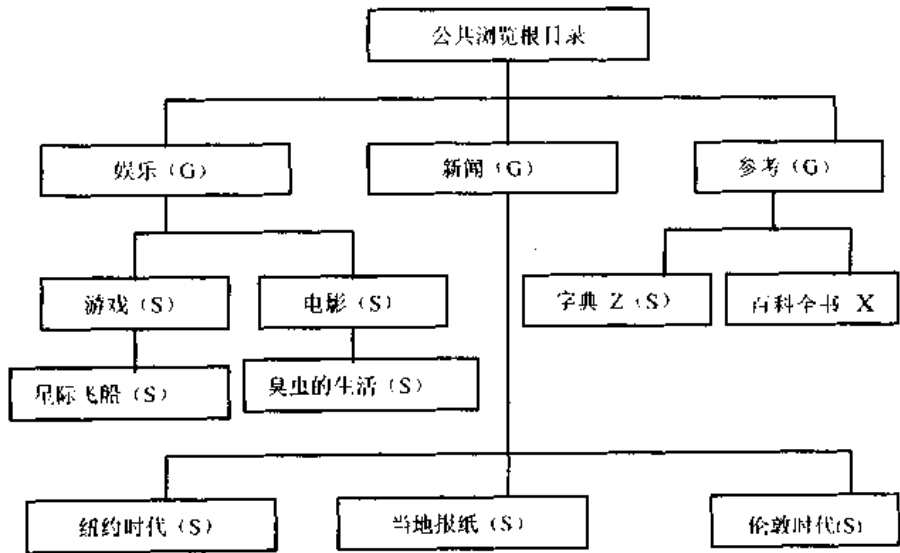


图 5.5 虚拟服务浏览层次结构

表 5.2 浏览层次所需的服务记录和服务属性

服 务 名	服 务 类 别	属 性 名	属 性 值
娱乐	BrowseGroupDescriptor	BrowseGroupList	PublicBrowseRoot
		GroupID	EntertainmentID

续表

服 务 名	服 务 类 别	属 性 名	属 性 值
新闻	BrowseGroupDescriptor	BrowseGroupList	PublicBrowseRoot
		GroupID	NewsID
参考	BrowseGroupDescriptor	BrowseGroupList	PublicBrowseRoot
		GroupID	ReferenceID
游戏	BrowseGroupDescriptor	BrowseGroupList	EntertainmentID
		GroupID	GamesID
电影	BrowseGroupDescriptor	BrowseGroupList	EntertainmentID
		GroupID	MoviesID
星际飞船	视频游戏类ID	BrowseGroupList	GamesID
臭虫的生活	电影类 ID	BrowseGroupList	MoviesID
字典Z	字典类 ID	BrowseGroupList	ReferenceID
百科全书X	百科全书类 ID	BrowseGroupList	ReferenceID
纽约时代	报纸类 ID	BrowseGroupList	NewspaperID
伦敦时代	报纸类 ID	BrowseGroupList	NewspaperID
地方报纸	报纸类 ID	BrowseGroupList	NewspaperID

5.3 数据表示

属性值可以包含具有强制复杂性的各种类型的信息，从而使得能够在不同服务类和环境中使用属性表。

SDP 定义了一个描述包含于属性值中数据的简单机制，其基本结构单元为数据元。

一个数据元表示一个打印数据。它由两个段组成：报文头段和数据段。报文头段又由两部分组成：一个类型描述符和一个尺寸描述符。数据段是一个字节序列，其长度在尺寸描述符中定义（参见数据尺寸描述符），其含义由类型描述符（部分）定义。

数据元类型由 5 位长的类型描述符代表。数据元头包含在数据元报文头首字节的最高 5 位中。表 5.3 是已被定义数据的类型。

数据元尺寸描述符由一个后面紧跟 0、8、16 或 32 位的 3 位尺寸索引字表示。该尺寸索引字包含于数据元头首字节的最低位中。尺寸索引编码如表 5.4 所示。

表 5.3 数据元类型描述符

类型描述符值	有效尺寸 描述符值	类 型 描 述
0	0	Nil, the null type
1	0,1,2,3,4	无精度整数
2	0,1,2,3,4	两位整数
3	1,2,4	UUID, 通用惟一标识符
4	5,6,7	文本串

续表

类型描述符值	有效尺寸描述符值	类型描述
5	0	逻辑
6	5,6,7	数据元序列，数据段是一个数据元序列的数据元。
7	5,6,7	可选数据元，数据段是一个数据元序列的数据元，从顺序中选出数据成分
8	5,6,7	URL，统一资源定位
9-31		保留

表 5.4 数据元尺寸描述符

尺寸索引	额外的字节	数据大小
0	0	1 个字节。除非数据成分分类为零，那么数据尺寸就为 0 个字节
1	0	2 个字节
2	0	4 个字节
3	0	8 个字节
4	0	16 个字节
5	8	数据大小包含在另外 8 位中，这 8 位为无精度整数
6	16	数据大小包含在另外 16 位中，这 16 位为无精度整数
7	32	数据大小包含在另外 32 位中，这 32 位为无精度整数

图 5.6 给出了几种数据元表示的实例。

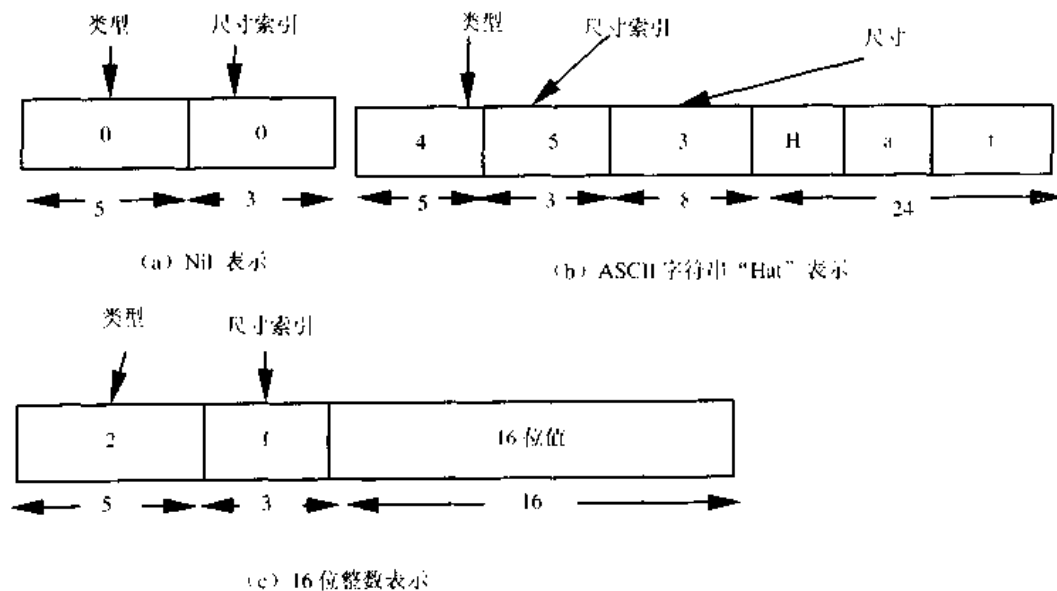


图 5.6 数据元表示示例

5.4 协议说明

SDP 是一个对通信要求最少的简单协议。它可工作于可靠分组传输模式（如果客户实现超时并且可在必要时进行重复请求，那么也可以工作于不可靠分组传输模式）。

SDP 使用一个请求/应答模型。在模型中，每一处理事务由请求协议数据单位(PDU)和应答协议数据单位(PDU)组成。然而，请求和应答实际上都可以不按次序进行传输。

在服务搜索协议使用蓝牙 L2CAP 传输协议的特定情况下，可以在一个 L2CAP 分组中传输多个 SDP PDU，在每一连接上只能发送一个这样的 L2CAP 给指定 SDP 服务器。限制 SDP 发送确认分组成为流控制形式的一种。服务搜索协议按 Big Endian 方式（即高位字节先于低位字节）进行传输。

5.4.1 协议数据单元格式

每一 SDP 协议数据单元(PDU)都由 PDU 头和 PDU 指定参数组成，如图 5.7 所示。报文头包含三个段：协议数据单元 ID、事务 ID 和 参数长度 ParameterLength；各个段的描述分别如表 5.5、表 5.6 和表 5.7 所示。

参数包括一个后续状态参数，下面给予描述：每一 PDU 类型的指定参数在后面的 PDU 描述中分别说明。

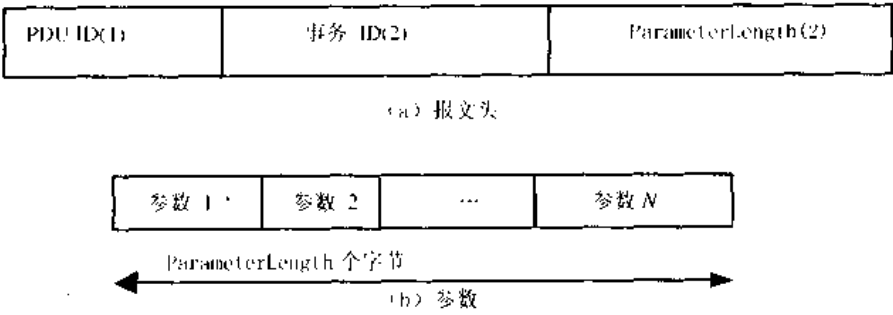


图 5.7 协议数据单元(PDU)格式

表 5.5 PDU ID

值	参数说明
N	PDU ID 段标识 PDU 类型，即其定义与指定参数
0x00	保留
0x01	SDP_ErrorResponse
0x02	SDP_ServiceSearchRequest
0x03	SDP_ServiceSearchResponse
0x04	SDP_ServiceAttributeRequest
0x05	SDP_ServiceAttributeResponse
0x06	SDP_ServiceSearchAttributeRequest
0x07	SDP_ServiceSearchAttributeResponse
0x07~0xFF	保留

表 5.6 事务 ID

值	参 数 说 明
N	TransactionID 段惟一标识请求 PDU，并被用于将应答 PDU 与请求 PDU 相匹配。SDP 客户可给请求 TransactionID 指定任意值，只要该值与所有发出请求不同。应答 PDU 的 TransactionID 要求与应答的请求 PDU 值一致。

表 5.7 ParameterLength

值	参 数 说 明
N	ParameterLength 段指定包含于 PDU 的所有参数，单位为字节

5.4.2 局部应答和后续状态

一些 SDP 请求可以要求比单个应答 PDU 更大的应答分组。这时，SDP 服务器将生成含有后续状态参数的局部应答。客户可以通过在以后的请求中提供后续状态参数，检索完全应答的下一部分。后续状态参数是可变长度段，其首字节中包含后续信息的附加字节数目，如图 5.8 所示。在 SDP 服务器中没有统一后续信息格式标准。每个后续状态参数只有对产生它的 SDP 服务器才有意义。

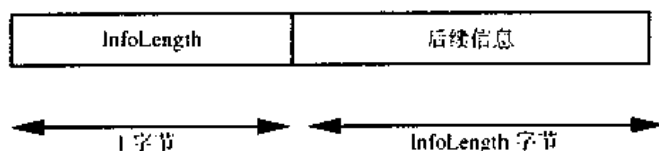


图 5.8 后续状态格式

在客户收到局部应答及其后续状态参数后，它将重发最初的请求（含新事务 ID），并在发往服务器的新请求中包含后续状态参数，以表示它想获得最初应答的其余部分。InfoLength 段中允许的最大值为 16 (0x10)。

注意当 SDP 服务器生成局部应答时，SDP 服务器可以在任意强制边界上分割应答。SDP 服务器根据应答内容选择进行分割的边界，但并不是必须要这样做。

5.4.3 出错处理

每一事务都由一个请求和一个应答 PDU 组成。通常，每种请求 PDU 类型都对应于一种应答 PDU 类型。但是，如果服务器确认请求格式不正确或由于某种原因，服务器不能采用合适的 PDU 类型进行应答时，该服务器将采用 SDP_ErrorResponse 协议数据单元应答，如图 5.9 所示。

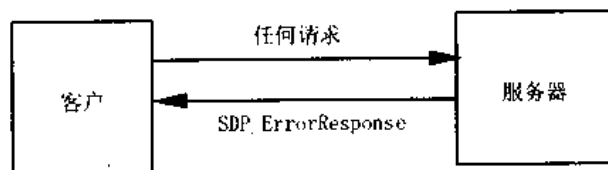


图 5.9

SDP 服务器生成本 PDU 类型，以对未正确格式化的请求 PDU 进行应答，或者在 SDP

服务器由于某些原因而不能生成合适的应答 PDU 时进行应答。SDP_ErrorResponse 协议数据单元的描述如表 5.8 所示，其中 Error Code 和 Error Info 的说明如表 5.9 和表 5.10 所示

表 5.8 SDP_ErrorResponse

PDU 类型	PDU ID	参 数
SDP_ErrorResponse	0x01	ErrorCode, ErrorInfo

表 5.9 ErrorCode

值	参 数 说 明
N	ErrorCode 标识 SDP_ErrorResponse PDU 生成的原因
0x0000	保留
0x0001	无效/不支持的 SDP 版本
0x0002	无效的服务记录句柄
0x0003	无效的请求语法
0x0004	无效的 PDU 尺寸
0x0005	无效的后续状态
0x0006	满足请求的资源不足
0x0007-0xFFFF	保留值

表 5.10 Errorinfo

值	参 数 说 明
Error-specific	ErrorInfo 是根据 ErrorCode 而定的参数，其含义取决于 ErrorCode 参数。当前定义的 ErrorCode 值不能指定 ErrorInfo 段的格式

5.4.4 服务搜索处理

服务搜索处理模式如图 5.10 所示。

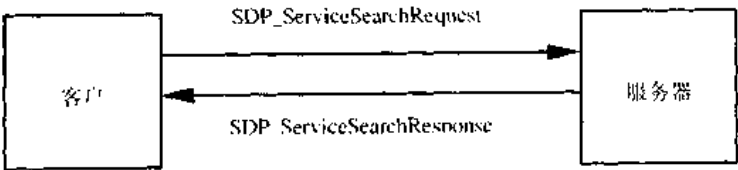


图 5.10 服务搜索处理模式

1. SDP_ServiceSearchRequest PDU

SDP 客户生成一个 SDP_ServiceSearchRequest 以定位匹配给定服务搜索模式的服务记录，该服务搜索模式是该 PDU 的首个参数。一收到该请求，SDP 服务器将检查其服务记录数据库，并将返回包含服务记录句柄的 SDP_ServiceSearchResponse，该服务记录匹配于给定服务搜索模式。SDP_ServiceSearchRequest PDU 的描述如表 5.11 所示，PDU 参数的描述如表 5.12、表 5.13 和表 5.14 所示。

注意：并未提供任何可以获取所有服务记录信息的机制。

表 5.11 SDP_ServiceSearchRequest PDU

PDU 类型	PDU ID	参 数
SDP_ServiceSearchRequest	0x02	ServiceSearchPattern, MaximumServiceRecordCount, Continuation State

表 5.12 ServiceSearchPattern

值	参 数 说 明
数据元序列	ServiceSearchPattern 是一个数据元序列, 在该序列中, 每一个数据元都是一个 UUID。该序列必须包含至少一个 UUID。该序列中最大的 UUID 值为 12*。服务搜索模式由 UUIDs 表构成
注: * 在服务搜索的范围和搜索请求协议数据单元 (PDU) 的大小之间, 值 12 为折衷值。在服务搜索模式使用的 UUID 不得多于 12 个	

表 5.13 MaximumServiceRecordCount

值	参 数 说 明
N	MaximumServiceRecordCount 是一个 16 位计数器, 该计数器指定用于应答该请求的返回服务记录句柄的最大数。SDP 服务器不应返回比此值更多的句柄。如果有多于 N 的服务记录与请求匹配, SDP 服务器需要确定采用哪些匹配的服务记录句柄应答。范围: 0x0001~0xffff

表 5.14 ContinuationState

值	参 数 说 明
连续状态	ContinuationState 由一个 8 位计数器 N、连续状态信息字节数以及服务器发回的 N 个字节连续状态信息构成。N 必须小于或等于 16。如果请求中不含连续状态参数, N 设置为 0

2. SDP_ServiceSearchResponse PDU

在 SDP 服务器收到一有效服务搜索请求 SDP_ServiceSearchRequest 时, 将生成一个服务请求应答 SDP_ServiceSearchResponse, 其定义如表 5.15 所示。该应答包含与请求服务搜索模式相匹配的服务记录的服务记录句柄表。

值得注意的是, 如果生成局部应答, 则它必须包含整数个完整的服务记录句柄, 而且不必将服务记录句柄值在多个 PDU 中分割开来。协议数据单元 (PDU) 参数如表 5.16~表 5.19 所示。

表 5.15 SDP_ServiceSearchResponse PDU

PDU	PDU ID	参 数
SDP_ServiceSearchResponse	0x03	TotalServiceRecordCount CurrentServiceRecordCount, ServiceRecordHandleList, ContinuationState

表 5.16 TotalServiceRecordCount

值	参 数 说 明
<i>N</i>	TotalServiceRecordCount 为一包含服务记录数目的整数，该服务记录须与请求的服务搜索模式匹配。如果无服务记录与请求的服务搜索模式匹配，则参数设置为 0。 <i>N</i> 不得大于 SDP_ServiceSearchRequest 的最大服务记录数 MaximumServiceRecordCount。当使用多个局部应答时，每一局部应答都应包含本参数的相同值。

表 5.17 CurrentServiceRecordCou

值	参 数 说 明
<i>N</i>	CurrentServiceRecordCount 是一个整数，用来指示包含下一个参数的服务记录句柄的数。如果无服务记录与请求的服务搜索模式匹配，这个参数就设置为 0。 <i>N</i> 决不能大于在当前应答的 CurrentServiceRecordCount 的值。 范围：0x0000-0xFFFF

表 5.18 ServiceRecordHandleList

值	参 数 说 明
32 位句柄表	ServiceRecordHandleList 包含一个服务记录句柄表。句柄数已在 CurrentServiceRecordCount 里列出。表中的每一句柄都指的是一个服务记录，该记录与请求的服务搜索模式匹配。注意该服务记录句柄表不包括数据元格式，不包括头段，而只包括 32 位服务记录句柄。

表 5.19 ContinuationState

值	参 数 说 明
连续状态	ContinuationState 由 8 位计数器 <i>N</i> 、连续状态信息字节数以及连续信息的 <i>N</i> 个字节构成。如果结束当前应答，本参数由为 0 的单个字节构成。如果 PDU 包括局部应答，ContinuationState 参数就包括在后续请求中，以检索应答的其余部分。

5.4.5 服务属性事务

服务属性事务模式如图 5.11 所示。

SDP 客户将生成一个 SDP_ServiceAttributeRequest 协议数据单元，以从一指定服务记录中检索指定属性值，并提供所需服务的服务记录句柄和从服务记录中检索的属性 ID 表作为参数。该协议数据单元的定义及其参数描述如表 5.20~表 5.24 所示。

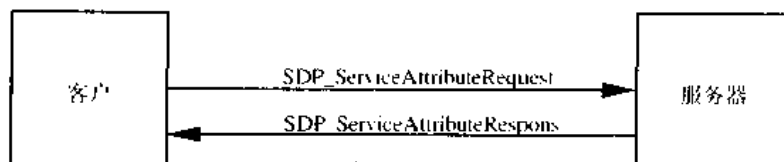


图 5.11 服务属性事务模式

表 5.20 SDP_ServiceAttributeRequest PDU

PDU 类型	PDU ID	参 数
SDP_ServiceAttributeRequest	0x04	ServiceRecordHandle, MaximumAttributeByteCount, AttributeIDList, ContinuationState

表 5.21 ServiceRecordHandle

值	参 数 说 明
32 位句柄	ServiceRecordHandle 参数根据检索到的属性值指明服务记录，可以通过前面的 SDP_ServiceSearch 事务中获取句柄

表 5.22 MaximumAttributeByteCount

值	参 数 说 明
N	MaximumAttributeByteCount 给出将在响应该请求的应答中返回的属性数据的最大字节数。SDP 服务器不得返回多于 N 个字节的应答。如果被请求属性多于 N 个字节，则由 SDP 服务器确定如何截断该表

表 5.23 AttributeIDList

值	参 数 说 明
数据元序列	AttributeIDList 为一数据元序列，其中，该表中每一数据元要么是属性 ID 要么是一个属性 ID 的取值范围。每一属性 ID 都编码为 16 位低精度整数数据元。每一属性取值范围都编码为 32 位低精度整数数据元，其中高段 16 位定为属性 ID 的起始段，低段 16 位定为属性 ID 的结束段。AttributeIDList 的属性 ID 必须以递增的顺序列表，且属性 ID 值不得重复。注意所有的被请求属性都指定在 0x0000-0xFFFF 范围内

表 5.24 ContinuationState

值	参 数 说 明
后续状态	ContinuationState 由 8 位计数的后续状态信息的字节数构成，后面是后续状态信息的 N 个字节，这个字节是从以前应答的服务器发回来。N 必须小于或等于 16；如果在请求中无后续状态，N 就设置为 0

2. SDP_ServiceAttributeResponse PDU

在 SDP 服务器收到有效 SDP_ServiceAttributeRequest 报文时，将生成一个 SDP_ServiceAttributeResponse 应答。该应答包含被请求服务记录属性列表(属性 ID 和属性值)。协议数据单元 (PDU) 的定义及其参数如表 5.25~5.28 所示。

表 5.25 SDP_ServiceAttributeResponse PDU

PDU 类型	PDU ID	参 数
SDP_ServiceAttributeResponse	0x05	AttributeListByteCount, AttributeList, ContinuationState

表 5.26 AttributeListByteCount

值	参 数 说 明
N	AttributeListByteCount 的值为 AttributeList 参数的字节数。 N 不得大于在 SDP_ServiceAttributeRequest 中定义的属性最大字节数 MaximumAttributeByteCount。范围：0x0002-0xFFFF

表 5.27 AttributeList

值	参 数 说 明
数据成分顺序	AttributeList 为一数据元序列，包含属性 ID 和属性值。该序列的第一个数据元包含第一个返回属性的属性 ID。序列的第二个数据元包含该属性对应的属性值。后面的数据元对将包含其他的属性 ID 和值对。AttributeList 中只包括 SDP_ServiceAttributeRequest 指定的服务记录非空属性值及其属性 ID。如果服务记录的属性 ID 或属性值为空，则不得包含于该 AttributeList 中

表 5.28 ContinuationState

值	参 数 说 明
后续状态	ContinuationState 由 8 位计数器 N 、后续状态信息字节数，以及后续信息的 N 个字节构成。如果当前应答结束，则该参数由为 0 的单个字节构成。如果 PDU 包含局部应答，则后续状态 ContinuationState 参数将包含在在后续请求中，以检索应答的其余部分

5.4.6 服务搜索属性事务

服务搜索属性事务模式如图 5.12 所示。

1. SDP_ServiceSearchAttributeRequest PDU

SDP_ServiceSearchAttributeRequest 事务综合 SDP_ServiceSearchRequest 和 SDP_ServiceAttributeRequest 二者功能于一个请求中。作为参数，它既包含服务搜索模式，又包含一张属性表，该属性表从与服务搜索模式匹配的服务记录中检索。SDP_ServiceSearchAttributeRequest 及其应答与 SDP_ServiceSearch 和 SDP_ServiceAttribute 两者相比，显得更复杂并且

可能需要更多的字节。但是，使用 SDP_ServiceSearchAttributeRequest 可以减少总的 SDP 事务量，特别是当检索多条服务记录时。该协议数据单元（PDU）的描述及其参数如表 5.29～5.33 所示。

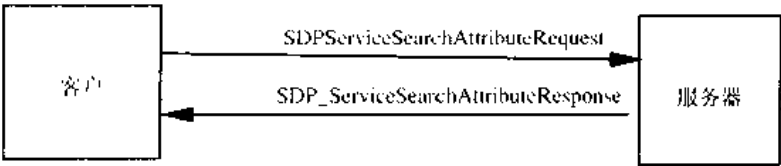


图 5.12 服务搜索属性事务模式

表 5.29 服务搜索属性请求

PDU 类型	PDU ID	参 数
SDP_ServiceSearchAttributeRequest	0x06	ServiceSearchPattern, MaximumAttributeByteCount, AttributeIDList ContinuationState

表 5.30 ServiceSearchPattern

值	参 数 说 明
数据元序列	ServiceSearchPattern 为一数据元序列，该序列中每一数据元都是一个 UUID。该序列必须包含至少一个 UUID，序列中 UUID 的最大值为 12*。 UUID 表构成服务搜索模式
* 12 作为服务搜索范围和搜索请求 PDU 之间的折衷值，并且服务搜索模式中使用的 UUID 不得多于 12 个	

表 5.31 MaximumAttributeByteCount

值	参 数 说 明
N	MaximumAttributeByteCount 指定请求应答所返回属性数据的最大字节数。SDP 服务器在应答中不得返回多 N 个字节的属性数据。如果被请求属性要求多于 N 个字节，则由 SDP 服务器决定如何截断该表。范围：0x0009-0xFFFF

表 5.32 AttributeIDList

值	参 数 说 明
数据元序列	AttributeIDList 为一数据元序列。其中，该表中数据元为属性 ID，或该属性的取值范围。每一属性 ID 都编码为 16 位低精度整数数据元。每一属性范围都编码为 32 位低精度整数数据元，其中高段 16 位为属性 ID 起始段，低段 16 位为属性 ID 的结束段。AttributeIDList 的属性 ID 必须以递增顺序在表中排列，且属性 ID 值不得重复，注意所有的属性值都应在 0x0000-0xFFFF 范围内

需要注意的是，对于每一服务记录，服务记录句柄将都包含于该服务的 ServiceRecordHandle 属性中，并且会与其他属性一起接受请求。

表 5.33 ContinuationState

值	参 数 说 明
连续状态	ContinuationState 由 8 位计数器 N 、后续状态信息字节数，以及服务器前一个应答中返回的 N 个字节的后续状态信息组成。 N 必须小于或等于 16。如果该请求不包括后续状态参数，则 N 设置为 0。

2. SDP_ServiceSearchAttributeResponse PDU

在 SDP 服务器有效 SDP_ServiceSearchAttributeRequest 时将生成一个 SDP_ServiceSearchAttributeResponse 应答。该应答包含一张服务记录属性表(属性 ID 和属性值)，该服务记录要求与所请求的服务搜索模式匹配。该协议数据单元(PDU)的定义及其参数描述如表 5.34~5.37 所示。

表 5.34 SDP_ServiceSearchAttributeResponse PDU

PDU 类型	PDU ID	参 数
SDP_ServiceSearchAttributeResponse	0x07	AttributeListsByteCount, AttributeLists, ContinuationState

表 5.35 AttributeListsByteCount

值	参 数 说 明
N	AttributeListsByteCount 包含 AttributeList 参数的字节数总值。 N 不得大于在 SDP_ServiceAttributeRequest 中定义的最大属性字节数 MaximumAttributeByteCount 值。范围：0x0002-0xFFFF。

表 5.36 AttributeLists

值	参 数 说 明
数据成分顺序	AttributeLists 为一数据元序列，该序列中每一数据元实际又是一个代表一张属性表的数据元序列。每一属性表都包含服务记录属性 ID 及其属性值。每一属性表中的第一个数据元都包含为该服务记录返回的第一个属性的属性 ID。属性表中的第二个数据元则包含该属性所对应属性值。后面的数据元则将包含其他的属性 ID 和值对。AttributeList 中只包括 SDP_ServiceAttributeRequest 指定的服务记录非空属性值及其属性 ID。如果服务记录的属性 ID 或属性值为空，则不得包含于该 AttributeList 中。每一属性列表中，属性都以属性 ID 值的增序列出。

表 5.37 ContinuationState

值	参 数 说 明
连续状态	ContinuationState 由 8 位计数器 N 、后续状态信息字节数，以及服务器前一个应答中返回的 N 个字节的后续状态信息组成。如果当前应答完成，则该参数由值为 0 单字节组成。如果给出局部应答，则该参数将在后续应答中给出，以检索应答的其余部分。

5.5 服务属性定义

本节的服务类和属性只包括直接支持 SDP 服务器的服务类，只是由 SDP 支持的服务类和属性表的一部分。

5.5.1 通用属性定义

通用属性是指其定义适用于所有服务记录的服务属性。但是，这并不意味着每一服务记录都必须包含所有这些服务属性值。然而，如果服务记录属性具有一个分配为通用属性的属性 ID，则该属性值必须符合通用属性定义。

在服务记录实例中，只允许存在两个属性：服务记录句柄 ServiceRecordHandle (属性 ID 0x0000) 和服务类 ID 列表 ServiceClassIDList (属性 ID 0x0001)。而所有其他服务属性对于服务记录都是可选项。

1. ServiceRecordHandle

服务记录句柄为一 32 位数，它惟一标识 SDP 服务器的每一服务记录，其定义如表 5.38 所示。特别要注意是，每一句柄在每一 SDP 服务器内是惟一的。如果 SDP 服务器 S1 和 SDP 服务器 S2 包含同一服务记录（代表同一服务），那么用来引用这些服务的句柄则是完全独立的。一般，用于应用 S1 服务的句柄将对于 S2 毫无意义。

表 5.38 ServiceRecordHandle

属 性 名	属性 ID	属性值类型
ServiceRecordHandle	0x0000	32-位低精度整数

2. ServiceClassIDList

ServiceClassIDList 属性由一个数据单元顺序构成，在该序列中每一数据元都是一个 UUID，该 UUID 代表某个服务记录所遵循的服务类。UUID 按照从具体类到通用类的顺序进行列表。服务类 ID 列表 ServiceClassIDList 必须包含至少一种服务类 UUID。其定义如表 5.39 所示。

表 5.39 ServiceClassIDList

属 性 名	属性 ID	属性值类型
ServiceClassIDList	0x0001	数据元序列

3. ServiceRecordState

ServiceRecordState 是一个用于缓存 ServiceAttributes 服务属性的 32 位的整数，其定义如表 5.40 所示。如果该属性包含于一条服务记录，那么在该服务记录中增删或改变其他属性值时，该值也一定会改变。如果该值从上次检查后就一直没发生变化，客户就可以推知服务记录的其他属性值也没有发生变化。

表 5.40 ServiceRecordState

属 性 名	属性 ID	属性值类型
ServiceRecordState	0x0002	32 位低精度整数

4. ServiceID

ServiceID 是一个可以普遍和惟一标识由服务记录描述的服务实例的 UUID，其定义如表 5.41 所示。如果同一服务在不只一个 SDP 服务器的服务记录中描述过，那么该服务属性将非常有用。

表 5.41 ServiceID

属 性 名	属性 ID	属性值类型
ServiceID	0x0003	UUID

5. ProtocolDescriptorList

ProtocolDescriptorList 属性描述可用于访问由服务记录所描述服务的一个或多个协议栈，其定义如表 5.42 所示。

表 5.42 ProtocolDescriptorList

属 性 名	属性 ID	属性值类型
ProtocolDescriptorList	0x0004	数据元序列或备选数据元

如果 ProtocolDescriptorList 只描述单个的协议栈，则它将采用数据元序列的形式，在该序列中每一序列数据元都是一个协议描述符。反过来，每一协议描述符又是一个数据元序列；其第一个数据元就是一个用于标识该协议的 UUID，而后面的数据元则是协议指定参数。潜在的协议指定参数是一个协议版本号和一个连接端口号。协议描述符按照从底层协议到高层协议的顺序列出，以用于对服务进行访问。

如果有一个以上的用于访问服务的协议栈，ProtocolDescriptorList 将采用备选数据元的形式，其中每一数据元都是一个数据元序列，如上段所述。

协议描述符惟一标识一则通信协议，并且提供协议指定参数，协议描述符以数据元序列表示。该序列中的第一个数据元必须是可惟一标识协议的 UUID；其他数据元则可以有选择性地提供协议指定信息。下面以 L2CAP 协议/服务多路复用器（PSM）和 RFCOMM 服务器信道号（CN）为例进行介绍。

假设 L2CAP 层的上层存在一个 RFCOMM 实例，L2CAP 协议指定信息（PSM）将指向该 RFCOMM 实例。如果在 L2CAP 层的上层存在两个不同但相互独立的 RFCOMM 实例，L2CAP 协议指定信息（PSM）指向可识别每一 RFCOMM 实例的惟一标识符。根据 L2CAP 规范，该标识符取值范围为 0x1000~0xFFFF。

IrDA-like 打印机：((L2CAP, PSM=RFCOMM)), (RFCOMM, CN=1), (Postscript-Stream));

IP 网络打印：((L2CAP, PSM=RF), (COMM RFCOMM, CN=2), (PPP), (IP), (TCP), (IPP));

同步协议描述符举例：((L2CAP, PSM=0x1001), (RFCOMM, CN=1), (Obex), (vCal)) ((L2CAP, PSM=0x1002), (RFCOMM, CN=1), (Obex), (其他同步应用 otherSynchronisationApplication))。

6. BrowseGroupList

BrowseGroupList 属性由一个数据元序列组成，在序列中每一数据元就是一个代表一个浏览组的 UUID，而服务记录就属于该浏览组。其定义如表 5.43 所示。

表 5.43 BrowseGroupList

属 性 名	属性 ID	属性值类型
BrowseGroupList	0x0005	数据元序列

顶级浏览组 ID 称为 PublicBrowseRoot 公共浏览根目录，即浏览层次结构的根，在蓝牙指定号码文件中该 ID 的值为：00001002-0000-1000-7007-00805F9B34FB (UUID16: 0x1002)。

7. LanguageBaseAttributeIDList

在一条服务记录中，为支持人们可读的多种自然语言属性，可以给服务记录中用到的每一自然语言分配一个基本属性 ID，其定义如表 5.44 所示。这样，就可以使用属性 ID 偏移，而不用绝对属性 ID 来定义人们可读的通用属性。该偏移来自于每一基本值。

表 5.44 LanguageBaseAttributeIDList

属 性 名	属性 ID	属性值类型
LanguageBaseAttributeIDList	0x0006	数据元序列

LanguageBaseAttributeIDList 属性实际就是一张列表。对于服务记录中的每一种自然语言，该表包括一个语言标识符、一个字符编码标识符和一个基本属性 ID。LanguageBaseAttributeIDList 属性由一个成员为 16 位低精度整数的数据元序列组成。该三个数据元组成一个三元组。

每一个三元组的第一个数据元包含一个可表示自然语言的惟一标识符，该语言基于 ISO 639:1988 (E/F)：“语言名字表示代码”。

每一个三元组的第二个数据元包含一个用于该语言字符编码方式的惟一标识符。字符编码值可在 IANA 数据库（注 1）中找到。这些编码的值可称作 MIBE-num 值。推荐字符编码方式为 UTF-8。

每一个三元组的第三个数据元包含作为服务记录中自然语言基本属性 ID 的一个属性 ID。同一服务器中的不同服务记录可对同一语言使用不同基本属性 ID 值。

为了便于采用一种主要语言实现对可读通用属性的检索，由服务记录支持的主要语言基本属性 ID 值必须为 0x0100。并且，如果服务记录包含 LanguageBaseAttributeIDList 属性，那么其第一个数据元的基本属性 ID 必须为 0x0100。

8. ServiceInfoTimeToLive

ServiceTimeToLive 属性为一个 32 位整数，如表 5.45 所示。它包含了以秒为单位的，

希望服务记录中信息保持有效和不变的时间长度。该时间间隔从 SDP 服务器检索属性值开始计起。但是，该值并不意味着服务记录将始终保持可用或不变。也可以说，客户可以用它来确定一个重新校验服务记录内容的轮询间隔时间。

表 5.45 ServiceInfoTimeToLive

属 性 名	属性 ID	属性值类型
ServiceInfoTimeToLive	0x0007	32 位低精度整数

9. ServiceAvailability

ServiceAvailability 属性为一个 8 位低精度整数，它用于表示服务是否具有接受其他客户的相关能力，其定义如表 5.46 所示。值 0xFF 表示服务现在没有在使用中并完全可用，而值 0x00 则意味着服务现在没有接受新的客户。对于支持多个并发客户的服务，中间值表示服务的相关线性利用率。

表 5.46 ServiceAvailability

属 性 名	属性 ID	属性值类型
ServiceAvailability	0x0008	8 位低精度整数

例如,可接受三个客户的服务应当提供服务利用率：当客户数为 0 时，该服务利用率值为 0xFF；当客户数为 1 时，该服务利用率值为 0xAA；当客户数为 2 时，该服务利用率值为 0x55；当客户数为 3 时，该服务利用率值为 0x00。0xAA 值近似于 $(2/3) * 0xFF$ ，表示 2/3 的利用率；而 0x55 值近似于 $(1/ 3) * 0xFF$ ，表示 1/3 的利用率。利用率值可以近似等于 $(1- (当前客户数 / 最大客户数)) * 0xFF$

如果最大客户数很大，则须修改该公式以确保服务利用率 ServiceAvailability 的值 0x00 和 0xFF 被保留，并分别用于表示利用率为 0 和利用率为 1 的情况。

注意根据服务可支持的最大客户数，可以根据服务当前客户使用资源的情况而不同。非零的 ServiceAvailability 并不保证可使用该服务，它只是表示可利用状态的近似值。

10. BluetoothProfileDescriptorList

BluetoothProfileDescriptorList 属性由一个数据元序列组成，该序列中的每一数据元都是一个包含有关该服务所遵循蓝牙标准信息的标准描述符。每一标准描述符都是一个数据元序列，其第一个数据元为赋给该标准的 UUID，其第二个单元是一个 16 位的标准版本号。该属性的定义如表 5.47 所示。

表 5.47 Blue tooth Profile Descriptorlist

属 性 名	属性 ID	属性值类型
BluetoothProfileDescriptorList	0x0009	数据元序列

标准的每一版本号都分配一个 16 位低精度整数，包含两个 8 位段。高端的 8 位包含主版本号段，低端的 8 位包含次版本号段。每一标准的最初版本具有为 1 的主要版本和一个为

0 的次要版本。标准作向上兼容的改变时，将增长次版本号。如果标准作不兼容的改变，则将增长主版本号。

11. DocumentationURL

该属性是一个指向服务记录所描述服务文档的 URL，其定义如表 5.48 所示。

表 5.48 DocumentationURL

属 性 名	属性 ID	属性值类型
DocumentationURL	0x000A	URL

12. ClientExecutableURL

该属性是一个指示应用所在位置的 URL，该应用可用于使用服务记录所代表的服务，其定义如表 5.49 所示。由于不同操作环境要求不同执行格式，那么就可以定义一种机制，以允许该属性可被用于定位适用于客户操作环境的执行程序。在使用该 URL 之前，客户应用程序可用代表操作环境的一个字符串，替代该 URL 属性的首字节，该字节值为 0x2A。

表 5.49 ClientExecutableURL

属 性 名	属性 ID	属性值类型
ClientExecutableURL	0x000B	URL

蓝牙号码分配文件中给出了代表操作环境的标准化字符串列表。例如，假设 ClientExecutableURL 属性值为 `http://my.fake/public/*/client.exe`。在一台能够执行 SH3 WindowsCE 文件的设备上，该 URL 将变为 `http://my.fake/public/sh3-microsoft-wince/client.exe`。在能够执行 Windows 98 二进制代码的设备上，该 URL 将变为 `http://my.fake/public/i86-microsoft-win98/client.exe`。

13. IconURL

该属性包含一个指向某图标位置的 URL，该图标用于标识由服务记录描述的服务，其定义如表 5.50 所示。由于不同硬件设备需要不同的图标格式，那么就可以定义一种机制，以使该属性可以用于定位适于该客户端设备的图标。在使用该 URL 之前，客户可采用一个代表所希望图标格式的字符串代替 URL 数性值中值为 0x2A 的首字节。

表 5.50 IconURL

属 性 名	属性 ID	属性值类型
IconURL	0x000C	URL

蓝牙号码分配文件中给出了代表图标格式的标准化字符串列表。例如，假设 IconURL 属性值为 `http://my.fake/public/icons/*`。在 256 色的 24×24 图标所在的设备上，该 URL 将变为 `http://my.fake/public/icons/24 \times 24 \times 8.png`。而在单色的 10×10 图标所在的设备上，该 URL 将变为 `http://my.fake/public/icons/10 \times 10 \times 1.png`。

14. ServiceName

ServiceName 属性包含一个由服务记录所表示服务的名称字符串，如表 5.51 所示。它应简短并便于表现由服务记录表示的服务图标。必须将偏移量 0x0000 与属性 ID 基地址相加(位于 LanguageBaseAttributeIDList 属性)，以便为本属性计算属性 ID。

表 5.51 ServiceName

属 性 名	属性 ID	属性值类型
ServiceName	0x0000	字符串

15. ServiceDescription

如表 5.52 所示，该属性是一个包括服务简短说明的字符串，其长度不到 200 个字符。应将偏移 0x0001 与属性 ID 基地址相加，以便为该属性计算属性 ID。

表 5.52 ServiceDescription

属 性 名	属性 ID	属性值类型
ServiceDescription	0x0001	字符串

16. ProviderName

该属性是一个包含提供服务的人名或组织名称的字符串，其定义如表 5.53 所示。应将偏移 0x0002 与属性 ID 相加(位于 LanguageBaseAttributeIDList 属性里)，以便为该属性计算属性 ID。

表 5.53 ProviderName

属 性 名	属性 ID	属性值类型
ProviderName	0x0002	字符串

17. 保留的通用属性 ID

0x000D~0x01FF 范围内的属性 ID 保留。

5.5.2 “服务搜索服务器”服务类属性定义

本服务类描述包含服务搜索服务器本身的服务记录。本节中所列属性只在服务类 ID 列表 (ServiceClassIDList) 属性包含服务搜索服务器服务类 ID (ServiceDiscoveryServer-ServiceClassID) 的情况下有效。注意，ServiceDiscoveryServer 类的服务记录包含所有通用属性。

1. ServiceRecordHandle 服务记录句柄属性

已在 ServiceRecordHandle 的通用属性定义里描述。

2. ServiceClassIDList 服务类 ID 列表属性

ServiceClassIDList 属性已在 ServiceClassIDList 的通用属性定义中描述。

3. VersionNumberList 版本号列表属性

VersionNumberList 是一个数据元序列，其定义如表 5.54 所示，该序列中的每个数据元都是 SDP 服务器支持的版本号。版本号是一个包含两个段的 16 位低精度整数。高段 8 位包含主版本号段，而低段 8 位则包含次版本号段。SDP 最初版本具有一个主版本号 1 和一个次版本号 0。当协议作向上兼容的改变时，将增长次版本号。如果 SDP 作非兼容改变时，将增长主版本号。这将保证客户和服务端是否支持一个通用主版本号。如果客户和服务端都支持一个次版本号，并只使用次版本规范特性，它们就可以相互通信。

表 5.54 VersionNumberList

属 性 名	属性 ID	属性值类型
VersionNumberList	0x0200	数据元序列

4. ServiceDatabaseState 服务数据库状态属性

ServiceDatabaseState 是一个便于实现服务记录缓存的 32 位整数，其定义如表 5.55 所示。如果存在该属性，则当从服务器数据库中添加或删除其他服务记录时，必须保证改变该属性值。如果该值自从上次客户查询以来一直没变，那么客户就可以推知：a) 没有增加或删除 SDP 所维护的所有其他服务记录；b) 服务器获得的任一服务记录句柄仍然有效。当到服务器的连接建立时，客户可以在使用前一连接期间获取的服务记录句柄之前，查询该值。

表 5.55 ServiceDatabaseState

属 性 名	属性 ID	属性值类型
ServiceDatabaseState	0x0201	32 位低精度整数

值得注意的是，当已有服务记录被修改时，包含增加、移动或服务属性修改，服务记录的服务数据库状态（ServiceDatabaseState）属性将保持不变。服务记录的服务记录状态（ServiceRecordState）属性则表示服务记录何时被修改。

5. 保留的属性 ID

在 0x0202~0x02FF 的范围内的属性 ID 将保留。

5.5.3 “浏览组描述符”服务类属性定义

本服务类描述由蓝牙设备支持，为每一浏览组描述符（BrowseGroupDescriptor）服务提供服务记录（ServiceRecord）。只有在服务类 ID 列表 ServiceClassIDList 属性包含浏览组描述符服务类 ID（BrowseGroupDescriptorServiceClassID）时，本节所列属性才有效。注意：所有通用属性都将包含在浏览组描述符（BrowseGroupDescriptor）类的服务记录里。

1. ServiceClassIDList 服务类 ID 列表属性

ServiceClassIDList 属性已在服务类 ID 列表的通用属性定义中描述。

2. GroupID 组 ID 属性

该属性包含一个可用于定位浏览组成员服务的 UUID，该服务由服务记录描述，如表 5.56 所示。

表 5.56 GroupID

属 性 名	属性 ID	属性值类型
GroupID	0x0200	UUID

3. 保留的属性 ID

在 0x0201~0x02FF 范围内的属性 ID 保留。

第 6 章 基于 TS 07.10 的 RFCOMM 协议

本章定义 RFCOMM 协议，其中包括根据蓝牙技术进行修正的 ETSI TS07.10 标准的一个子集。

6.1 引言

RFCOMM 协议提供对基于 L2CAP 协议的串口仿真，该协议基于 ETSI 标准 TS 07.10。本文并不详细描述完整的 TS 07.10 规范，而只利用 TS 07.10 标准一个子集，并根据蓝牙技术作出适当修正。详细内容可参见 TS 07.10 标准。

RFCOMM 是一个简单传输协议，其中针对 9 针 RS-232 (EIA/TIA-232-E) 串口仿真附加了部分条款。RFCOMM 协议可支持在两个 BT 设备之间同时保持高达 60 路的通信连接。可由 BT 设备同时利用的连接数量根据实际应用情况而定。

RFCOMM 的目的，是针对如何在两个不同设备（通信的两端）上的应用之间保证一条完整的通信路径，并在他们之间保持一通信段。图 6.1 表示一条完整通信路径。图中应用不只表示终端用户应用，也可以是高层协议或作为终端用户应用的其他服务。



图 6.1 RFCOMM 通信段

RFCOMM 准备把利用串口设备进行通信的应用覆盖在内。在一个简单配置实例当中，通信段就是设备之间的 BT 直接链路，如图 6.2 所示。如果通信段为另一网络，BT 用于在该设备和网络接入设备（如 Modem）之间建立路径。RFCOMM 只针对直接互连设备之间的连接，或者是设备与网络接入设备之间的互连。RFCOMM 支持其他的配置方式，如一端采用 BT 通信，另一端采用有线接口，如图 6.3 所示。这些设备不只是调制解调器，而且还提供简单服务。

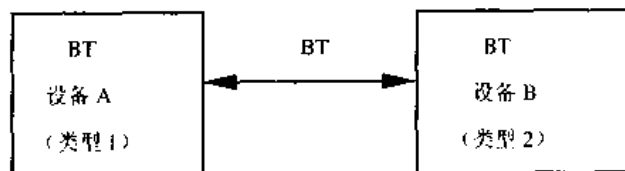


图 6.2 利用 COM 口的 RFCOMM

通信两端设备必须兼容于 RFCOMM 协议。第一类设备是诸如计算机、打印机等通信终端设备。第二类设备是通信段的一部分，如 Modem。但是为了简化协议内容，RFCOMM 协议对这两种设备不作区分。

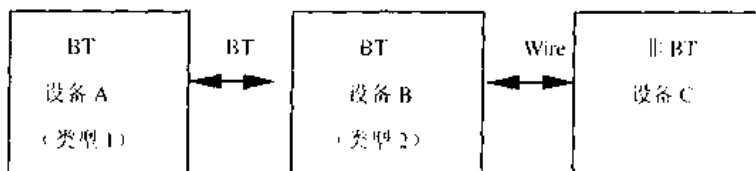


图 6.3 利用 COMM 设备的 RFCOMM

在两个 RFCOMM 实体间传输信息也都支持这两类设备，其中有些信息只用于第二类设备。本协议中也没有对两类设备所用信息进行严格区分。而是由本协议的用户决定使用哪些信息。由于一个设备并不知道通信路径上的其他设备的类型，所以每一个设备都应按照协议规定发送所有可用信息。

6.2 RFCOMM 服务

RFCOMM 仿真 RS-232 (EIA/TIA-232-E) 串口。仿真过程包括非数据通路状态的传输，RFCOMM 内置空 Modem 仿真标准框架。

如果通过 RFCOMM 服务接口设定指定端口的波特率，也不会影响 RFCOMM 的实际数据吞吐量。也就是说，RFCOMM 不限制人工速率或步长。但是，如果通信链路两端的设备都是负责将数据转发到其他通信介质的第二类设备，或在两端 RFCOMM 设备接口上进行数据传输，实际数据吞吐一般将反映波特率的设置。

RFCOMM 支持两个设备间的多串口仿真，也支持多个设备间多串口的仿真。

6.2.1 RS-232 控制信令

RFCOMM 提供对 9 针 RS-232 接口的仿真，其通路设置如表 6.1 所示。

表 6.1 RFCOMM 中的仿真 RS-232 通路

针	通路名称
102	公用信号
103	发送数据 (TD)
104	接收数据 (RD)
105	请求发送 (RTS)
106	清除发送信号 (CTS)
107	数据准备就绪 (DSR)
108	终端准备就绪 (DTR)
109	数据载波监听 (CD)
125	铃声报警 (RI)

6.2.2 空 Modem 仿真

RFCOMM 基于 TS 07.10。当设备准备传输非数据通路的状态信息时，TS 07.10 不区分 DTE 和 DCE 设备，而是通过 RS-232 控制信号来表示 DTE/DCE 各自的信号。表 6.2 反映了 TS07.10 信号与 RS-232 控制信号之间的对应关系。

表 6.2 TS07.10 串口控制信号

TS07.10 信号	对应的 RS-232 控制信号
RTK	DSR, DTR
RTR	RTS, CTS
IC	RI
DV	DCD

当两个同类设备互连时，TS07.10 传输 RS-232 控制信号的方式就会创建空 Modem。图 6.4 体现了在两个 DTE 设备通过 RFCOMM 互连时空 Modem 的创建过程。虽然，没有一种空 Modem 有线传输方案能够满足所有情况下的通信要求，但 RFCOMM 所提供的空 Modem 方案能够满足大多数情况下的通信要求。

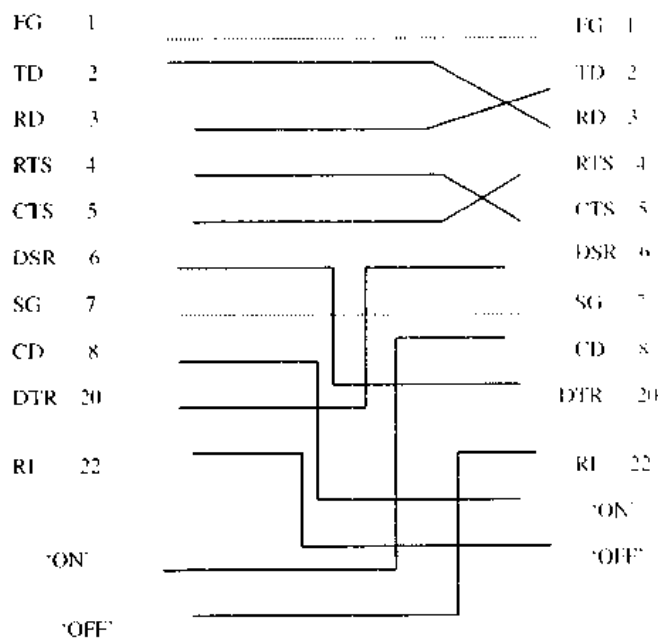


图 6.4 RFCOMM DCE-DTE 空 Modem 仿真

6.2.3 多串口仿真

1. 两设备间的多串口仿真

两设备间的多串口仿真，如图 6.5 所示。两个采用 RFCOMM 进行通信的 BT 设备有可能同时打开多个串口。RFCOMM 支持同时打开 60 个仿真端口。但是，一个设备打开端口数根据实际实现而不同。

一个数据链接标志（DLCI）惟一标识一对客户和服务端之间的持续连接。DLCI 长度为 6 字节，其无效值区间为 2 至 61。TS07.10 中，DLCI 0 为控制信道，DLCI 1 根据服务器信道概念不能使用，DLCI 62-63 保留使用。DLCI 在两个设备间的 RFCOMM 会话中保持一致。这部分内容将在下节进一步阐述。

在一次 RFCOMM 会话中，客户和服务端应用可以分布在通信的两端，每一端的客户都可以独立发起建立通信连接。因此，利用 RFCOMM 服务器信道的概念将 DLCI 值域空间在

两个正在进行通信的设备间进行划分。这部分内容将在 RFCOMM 服务器信道的 DLCI 定位节中进一步阐述

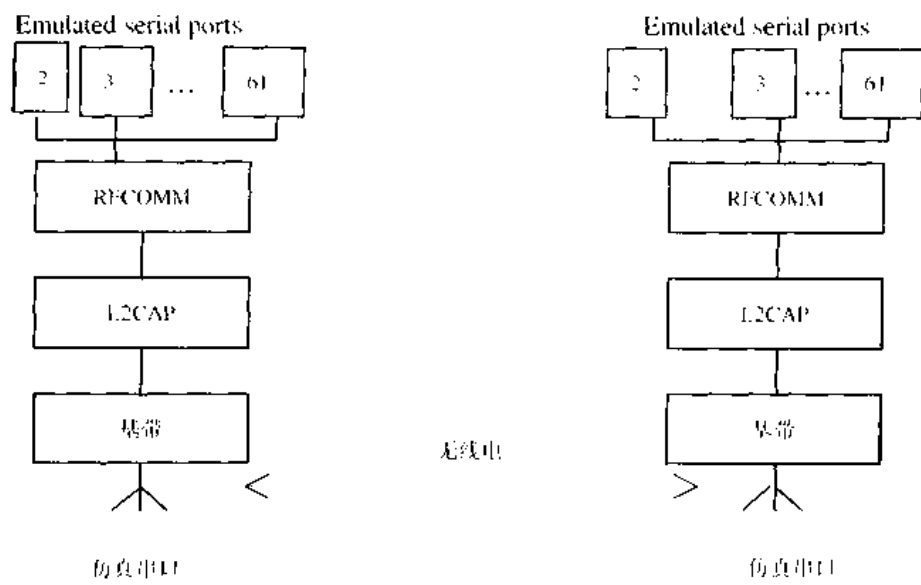


图 6.5 多串口仿真

2. 多仿真串口和多 BT 设备

如果 BT 设备支持多串口仿真，通信连接两端允许使用不同 BT 设备，那么 RFCOMM 实体必须能够运行多个 TS07.10 多路复用器会话，参见图 6.6。每一多路复用器都使用其 L2CAP 信道 ID (CID)。RFCOMM 可以选择支持 TS07.10 多路复用器的多个会话。

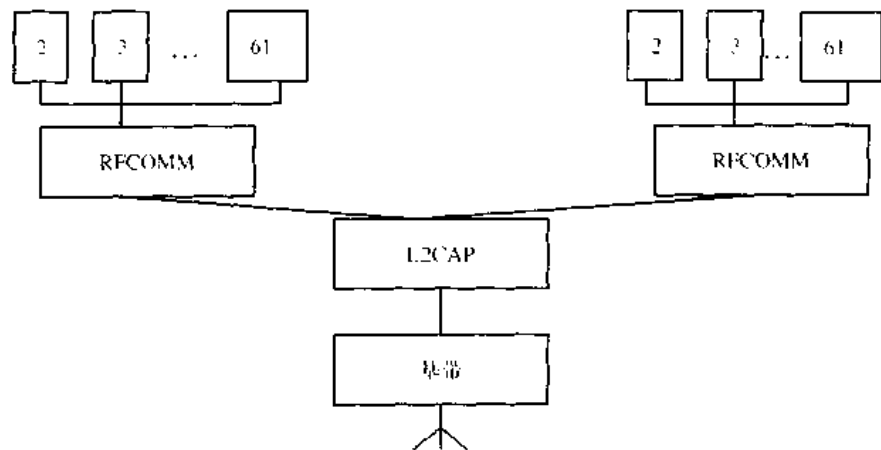


图 6.6 两个 BT 设备的多串口仿真

6.3 服务接口描述

RFCOMM 目的在于定义一个能够利用仿真串口的协议。大多数系统中，RFCOMM 将成为包括串口仿真实体的端口驱动程序的一部分。

图 6.7 给出了 RFCOMM 如何适应于典型系统的模型。此图提出了 RFCOMM 参考模型。RFCOMM 参考模型各组成部分描述如表 6.3 所示

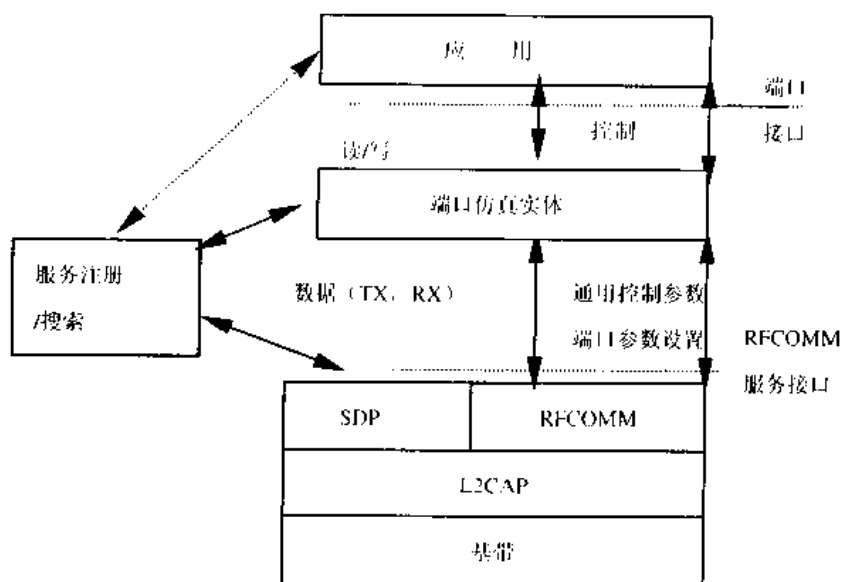


图 6.7 RFCOMM 参考模型

表 6.3 RFCOMM 参考模型的组成

组成部分	描 述
应用	利用端口通信接口
端口仿真实体	端口仿真实体将系统通信接口映射到 RFCOMM 服务、端口仿真实体与 RFCOMM 组成端口驱动程序
RFCOMM	基于 L2CAP 信道提供透明数据流和控制通道，复用多个仿真串口
服务注册/搜索	服务器应用注册在本地服务上，并向客户端应用提供获取其他服务上服务器端应用的服务
L2CAP	协议复用，SAR
基带	BT 定义的基带协议

6.4 RFCOMM 支持的 TS07.10 子集

1. 选项和模式

RFCOMM 利用 TS07.10 的基本选项进行定义。

2. 帧类型

表 6.4 列出了 RFCOMM 支持的帧类型。

RFCOMM 不支持“未加标记信息(UI)指令和应答”帧类型。另外，RFCOMM 的任何一种帧格式都不支持 TS07.10 协议的纠错机制。

表 6.4 RFCOMM 支持的帧类型

帧类型
异步平衡模式设置(SQBM)指令
未加标记的确认 (DM) 应答
断开模式(DM)指令
连接断开 (DISC) 指令
带头校验的未标记信息 (UIH) 指令和应答

3. 指令集

TS07.10 定义了一种可以具有完整控制通道 DLCI 0 的多路复用器。控制通道用于在两个多路复用器之间传递信息。表 6.5 所示的 TS07.10 指令得到 RFCOMM 的支持。

表 6.5 RFCOMM 支持的指令

支持的控制通道指令集
测试指令 (Test)
流控制打开指令 (Fcon)
流控制关闭指令 (Fcoff)
Modem 状态指令(Msc)
远程端口协商指令 (RPN)
远程通路状态 (RLS)
DLC 参数协商 (PN)
对未被支持的指令的应答 (NSC)

无论何时收到未支持的指令类型，NSC 帧就作为应答信息发出。

4. 聚集层

RFCOMM 只支持 TS07.10 中的第一种聚集层类型。

Modem 状态指令 (MSC) 应用于传递 RS-232 控制信号和仿真串口的断开信号。

6.5 根据蓝牙对 TS07.10 的修正

6.5.1 介质调整

RFCOMM 不使用 TS07.10 基本帧格式中的开始和结束标志，而仅仅使用包含在 L2CAP 层和 RFCOMM 层间交换标志中的那些域，如表 6.6 所示。

TS07.10 中，帧校验序列 (FCS) 根据不同帧类型在不同域集上进行运算。下面列出需要进行帧运算的域。

对于 SABM、DISC、UA、DM 帧，在地址、控制和长度标志域上进行运算；对于 UIH 帧，在地址和控制域上进行运算。

表 6.6 基本帧结构

标志	地址	控制	长度标志	信息	FCS	标志
01111101	8 字节	8 字节	8 字节或 16 字节	不定长，但是整数个 8 字节 L2E	8 字节	01111101

注：为了便于表达清楚和制定 RFCOMM 标准，FCS 运算中的域都在 TS07.10 7.0.0 版本中进行了修改，但是 RFCOMM 没有对上面的 FCS 运算方案进行改动。

6.5.2 TS07.10 多路复用器的启用和关闭过程

RFCOMM 不支持 TS07.10 第 5.7 节中定义的启用和关闭过程。也就是说，RFCOMM 不支持 AT 指令 AT+CMUX 和多路复用器的关闭指令。

任意两个设备之间最多只能保持一个 RFCOMM 会话。当建立一个新的 DLC 链路时，如果已经存在一个 RFCOMM 会话，应检查会话发起一方，然后在上面建立新 DLC。一个会话由两个通信终端的蓝牙 BD_ADDR 惟一标识。

1. 启用程序

由建立两个设备间的第一个仿真串口连接的设备负责建立多路复用控制通道。这包括以下步骤：

- 利用 L2CAP 基本服务，建立与对等 RFCOMM 实体之间的 L2CAP 通路，参见 L2CAP 基本服务。
- 通过在 DLCI 0 上发送 SABM 启动 RFCOMM 多路复用器，并等候对等实体的 UA 应答。当然也有可能进行进一步的协商。

经过以上步骤，用于用户数据通讯的 DLC 通路就被建立起来。

2. 关闭程序

关闭指定会话上最后一个连接（DLC）的设备负责通过关闭相应 L2CAP 通路关闭多路复用器。可以根据具体实现决定是否通过在 DLCI 0 上发送 DISC 指令帧关闭多路复用器，但必须直接用 UA 应答 DISC 指令。

3. 链路丢失处理

如果收到 L2CAP 链路丢失通知，本地 RFCOMM 实体应负责向每一条激活的 DLC 链路端口仿真/代理实体发送链路丢失通知，然后释放所有与此 RFCOMM 会话有关的资源。而端口仿真/代理实体采取的动作则取决于高层 API。例如，假设该设备为 DTE，对于一个仿真串口（vCOMM），就应撤销 CD、DSR 和 CTS 信号。

6.5.3 系统参数

表 6.7 包括了 TS07.10 多路复用的 RFCOMM 执行版本的所有应用系统参数。

其中，时钟 T1 是发送 P/F 位为 1 的帧的所需超时（在 RFCOMM 中只适用于 SABM 和 DISC 帧）。T2 是以 DLCI 0 上 UIH 帧格式发送指令所需超时。

因 RFCOMM 依赖于低层提供可靠传输，超时时其缺省动作为关闭多路复用会话。

惟一的例外是在已有会话上试图建立新的 DLC 链路时，也就是等待对 SABM 指令的 UA

应答时如果会话发起方知道通信延时主要来源于用户交互时，就会将会话超时不定期延长。

表 6.7 系统参数取值

系统参数	值
帧的最大尺寸 (N1)	缺省为 127，可以为 23~32767 之间的任意值
确认时钟 (T1)	60 秒
多路复用控制通路的应答时钟 (T2)	60 秒

无论如何，建立连接最终都要考虑超时问题。会话发起方应在同一 DLSI 通路上发送一个类似于 SABM 指令帧的 DISC 指令帧，目的在于通知其他实体该连接已被放弃。然后，会话发起方等待 DISC 指令的 UA 应答。

6.5.4 利用 RFCOMM 服务器通道进行 DLCI 定位

在一次 RFCOMM 会话中，服务器和客户可以同时位于会话的两端，每一客户端都可以独立建立连接。这样，DLCI 值域就将被两个利用 RFCOMM 服务器通道和方向位 (Direction bit) 概念的通信设备划分。

RFCOMM 服务器通道号实质上是 TS01.10 帧中地址域 DLCI 部分位的子集，如表 6.8 所示。

表 6.8 地址域格式

位	1	2	3	4	5	6	7	8
TS07.10	EA	C/R	DLCI					
RFCOMM	EA	C/R	D	服务器端通道				

注册为 RFCOMM 服务接口的服务器端应用应在 1 至 30 范围内，指定一个服务器端通道号(在 TS07.10 中，0 和 31 由相应 DLCI 保留使用)。该值应在服务搜索数据库中登记。

对于一次 RFCOMM 会话，发起方设备方向位(Direction bit)设为 D=1 (相反，把 D=0 赋给其他的设备)。当在已有的 RFCOMM 会话上建立一条新的数据链接时，方向位(Direction bit)用于与服务器端通道相关，以确定其 DLCI，从而建立到特定应用的连接。连接建立后，DLCI 就在两端间的两个方向上传输数据分组。

DLCI 值域实际上分为两部分：非发起方设备上的应用使用 DLCI 偶数号 (2, 4, ..., 60) 访问，发起方设备上的应用则使用 DLCI 奇数号 (3, 5, ..., 61) 访问。注意：对于一个支持多路同步 RFCOMM 会话的设备而言，方向位不一定在所有会话中都一致。一个在已有会话上建立新 DLC 的 RFCOMM 实体，将其他设备应用使用的服务器端通道和该会话方向位的求反值组合为 DLCI。RFCOMM 将 DLCI1 和 62-63 保留，并不使用。

6.5.5 多路复用控制指令

在 TS07.10 中，一些附属于 DLCI 的多路复用控制指令在相应 DLC 建立起来之前，可在 DLCI 0 上进行交换 (参见 PN 和 RPN 指令)。在收到 DISC 指令帧或从本地关闭 DLC 时，与单条 DLC 相关的所有状态都应重置为缺省值。也就是说，同一会话连接上的所有 DLC 的创建或重建都可以预见其结果，而与该会话历史无关。

1. 远程端口协商指令(RPN)

RPN 指令可以在新的 DLC 打开前使用, 并且只能在端口设置发生改变时使用。

RPN 指令在 TS07.10 中设置为可选项, 但在 RFCOMM 中则为协议正式内容。

2. 远程线路状态指令(RLS)

该指令用于指示远程端口状态。

RLS 指令在 TS07.10 中设置为可选项, 但在 RFCOMM 中则为协议正式内容。

3. DLC 参数协商(PN)

该指令于新的 DLC 打开之前使用。PN 指令在 TS07.10 中设置为可选项, 但在 RFCOMM 中则为协议正式内容。

用于传递不用于 RFCOMM 信息的 PN 指令具有一些参数。因此, 这些参数域必须由发送方预先赋值, 并且接受端必须认识该参数域。上述参数域包括:

- H1~H4 应设置为 0, 即使用 UIH 帧;
- CL1~CL4 应设置为 0, 即使用第一类聚集层;
- T1~T8 应设置为 0, 即确认定时器 T1, 在 RFCOMM 为硬性规定;
- NA1~NA8 应设置为 9, 转发 N2 代码, 在 RFCOMM 常置为 0;
- K1~K3 应设置为 0, 用于定义纠错模式下窗口大小, 不适用于 RFCOMM。

如果任一命令的任一参数域收到非法值或无法访问的值, 就应发出一个 DLC 参数协商应答, 该应答包含可由应答设备接受的值。

6.6 流控制

有线端口通常使用 RTS/CTS 等流控制控制通信。另一方面, RFCOMM 和低层 L2CAP 间的流控制则取决于实际支持的服务接口。而且, RFCOMM 也有其自己的流控制机制。

L2CAP 依赖于基带链路管理层提供的流控制机制。而 L2CAP 和 RFCOMM 层间的流控制机制根据具体实现定义。

有线串行端口流控制分为两大类: 利用 XON/XOFF 等字符的软件流控制, 和利用 RTS/CTS 或 DTR/DSR 电路的流控制。有线链路两端可能同时采用这两种方法, 也可能只在一端进行。

RFCOMM 协议提供两类流控制机制:

- RFCOMM 协议定义了能对两个 RFCOMM 实体之间全部数据流操作的流控制指令 Fcon 和 Fcoff, 对所有的 DLCI 都起作用。

- 调制解调器状态指令 Msc 实质就是可操作单个 DLCI 的流控制机制。

端口仿真实体串行流控制分两种情况:

对于第一类设备, 端口驱动程序 (即加载 RFCOMM 的端口仿真实体) 需要提供其仿真 API 中定义的流控制服务。应用可以请求 XON/XOFF 或 RTS/CTS 等指定流控制机制, 并可通过端口驱动程序处理流控制。

对于第二类设备, 端口驱动程序在通道的非 RFCOMM 部分 (即 RS-232 端口) 上执行流控制。该流控制通常是由第一类设备的对等 RFCOMM 实体发出的控制参数定义。本节中

流控制描述主要用于第一类设备的端口驱动程序。

由于 RFCOMM 已经具有流控制机制，因此端口驱动程序不再需要利用应用请求的方式执行流控制。理想情况下，由应用设定流控制机制，并假定其实现细节由 COMM 系统处理。这样，端口驱动程序就可以忽略该请求而直接利用 RFCOMM 的流控制。由此，应用就能够发送和接收数据，而不用担心端口驱动程序没有通过请求机制进行流控制。但是，在实际中这种方式存在一些问题：

- 基于 RFCOMM 的端口驱动程序在基于分组的协议上运行，数据可能会在通信链路中某个地方缓存。而在有线通信的同样情况下，端口驱动程序就不能执行流控制。
- 应用可以自己决定采用流控制机制，而不仅仅是通过端口驱动程序请求流控制。

这些问题说明端口驱动器必须为执行流控制仿真作更多的工作。下面是流控制仿真的基本规则：

- 端口驱动程序不能仅仅依靠应用请求的流控制机制，而且可以使用多种流控制混合机制；
- 端口驱动程序必须清楚应用请求的流控制机制。并且，当它发现非数据环路（硬件流控制方式）或输入数据中的流控制字符（软件流控制方式）时，端口驱动程序采取与有线通信相同的工作方式。例如，如果有线方式下需要卸载 XOFF 和 XON 字符，那么基于 RFCOMM 的端口驱动程序也必须这样做；
- 如果应用通过端口驱动程序接口设定流控制机制并触发该机制，端口驱动程序必须采取与有线方式相似的操作。例如，如果有线条件下需要传输 XOFF 和 XON 字符，那么端口驱动程序也必须传输这些字符。

这些基本规则可应用于每一种有线控制机制的仿真。而且，可以同时设定多种流控制。TS 07.10（v6.3.0）的 5.4.8 节对每种流控制机制都做出了定义。

6.7 与其他实体的互操作

6.7.1 端口仿真和端口代理实体

本节定义 RFCOMM 协议如何仿真串口。RFCOMM 协议支持的两类设备如图 6.8 所示。

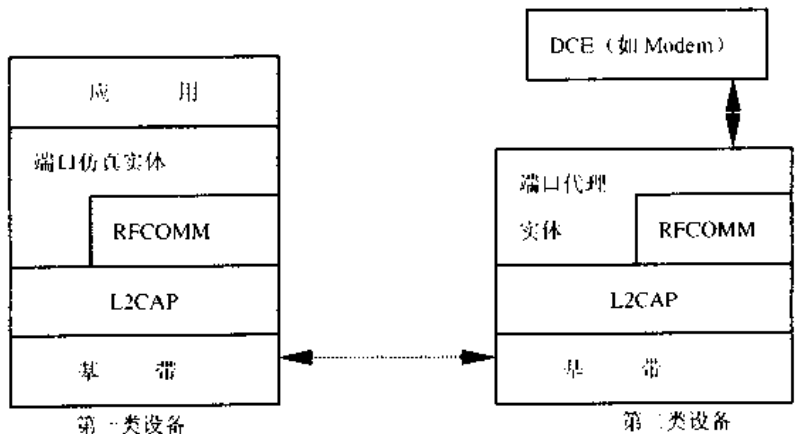


图 6.8 RFCOMM 通信模型

第一类设备是计算机和打印机等通信端设备，第二类设备是通信段的一部分，如

MODEM。端口仿真实体将系统指定通信接口（API）映射于 RFCOMM 服务。端口代理实体将数据从 RFCOMM 转发至链接 DCE 设备的外部 RS-232 接口。RS-232 接口的通信参数根据接收的 RPN 指令进行设置，参见 6.5.5 节。

6.7.2 服务注册和搜索

在原有非蓝牙应用上运行的蓝牙配置应用，主要负责一些应用或服务的注册。这些应用或服务包含可用于访问 RFCOMM 服务器通道信息。

下面是一个特定 RFCOMM 服务记录的开发模板或范例。它利用单个服务类和两个协议的协议描述列表（Protocol List）解释了服务类列表（Service Class List）的内涵。尽管在 RFCOMM 之上可能有更多的协议，本例表现了其他服务属性（Service Name）的使用。对于每个运行在 RFCOMM 之上的服务，应采用合适的已定义 SDP 的通用属性或服务属性。

那些客户端用于连接 RFCOMM 之上服务的属性至少应包括服务类列表和协议描述符列表（Protocol Descriptor List），如表 6.9 所示。

表 6.9 服务类列表

项目	定义	类型/大小	取值	属性 ID
ServiceClassList			1	0x001
ServiceClass0	⑤	UUID/32 bit	1	
ProtocolDescriptorList				0x004
Protocol0	L2CAP	UUID/32 bit	L2CAP 1	
Protocol1	RFCOMM	UUID/32 bit	RFCOMM 1	
ProtocolSpecificParameter0	服务器通道	Unit8	N= 服务器通道	
（其他协议）		UUID/32 bit	1	
[其他协议指定参数]	⑥	32	32	
ServiceName	可显示的文本名	数据元/字符串	“服务实例”	2
其他适合于服务的公共属性	④	1	142	1
[指定服务属性]	③	32	32	3

注释：

- ①定义于“蓝牙分配号码”；
- ②对于支持“可显示”文本字符串属性的其他语言，应根据该语言对语言基本属性列表（Language Base Attribute List）值增加偏移量；
- ③根据具体服务定义；
- ④对于具体服务，可以采用一些定义 SDP 的公共属性；
- ⑤表示服务类，对于大多数的服务类都只有一个入口或列表。

6.7.3 低层约束

1. 可靠性

RFCOMM 利用 L2CAP 服务建立连接到其他服务上。RFCOMM 实体的 L2CAP 通道

L2CAP 通道用于 RFCOMM/TS07.10 多路复用器会话。根据 5.1 节进行适当修正的 TS07.10 帧就通过该通道发出。

一些帧类型(SABM 和 DISC), 以及 DLCI 0 上发出的带有多路复用器控制指令的 UIH 帧, 通常需要远程实体的应答, 以便进行 RFCOMM 层次上的确认。(当无确认时不重发, 参见“系统参数”一节)。RFCOMM 协议中对数据帧不需要进行应答和确认。

因此, RFCOMM 需要 L2CAP 提供具有最大可靠性的通道, 以确认按次序和不重复地传输所有帧。如果 L2CAP 通道不能够提供该特性, RFCOMM 则需要由 RFCOMM 处理链路丢失通知, 参见“链路丢失处理”一节。

2. 节能模式

如果所有指向某设备的 L2CAP 通道在一段时间内空闲, 应将该设备置为节能模式。(即使用挂起、呼吸或休眠状态, 参见基带定义)。RFCOMM 对此未作解释, 但给出了 L2CAP 的潜在要求。低层可根据该信息决定使用何种节能模式。

然而, RFCOMM 协议不能接受节能模式引起的潜在通信延迟, 本文未注明 RFCOMM 操作最大延时。可能允许的延时取决于应用需求, 也就是说, RFCOMM 服务接口可以通过某种方式体现可能的延时要求, 并通过 RFCOMM 应用进行汇总并传输到 L2CAP。

第7章 IrDA 互操作性

在蓝牙技术中, IrOBEX 协议提供相对于 IrDA 协议层次的相应特性, 能够使应用在蓝牙协议栈上运行, 与在 IrDA 协议栈上相同。

7.1 概述

本章目的在于指导在短距离 RF 和 IR 介质上开发应用项目。这两类介质各有优缺点, 但目标都在于运行应用。因此, 与其说本章内容在于划分应用范围, 不如说在于定义蓝牙应用与 IrDA 应用间的互操作性, 其互操作点为 IrOBEX。

IrOBEX 是由 IrDA 定义的会话协议。蓝牙技术将该协议同时用于支持蓝牙无线技术和 IrDA IR 技术应用。IrDA 和蓝牙技术都是设计用于短距离无线通信, 但是他们仍在低层协议上存在很大的差异。因此, IrOBEX 才单独制定适用于蓝牙的低层映射协议。

本章定义 IrOBEX 在 RFCOMM 和 TCP/IP 间的映射方式。最初, OBEX(Object Exchange Protocol)对象交换协议的开发目的是在红外线链路上实现对象交换, 并置于 IrDA 协议层次内。但是, 它在目前的 RFCOMM 和 TCP/IP 协议层次中居于传输层之上。那么我们可以说, OBEX over TCP/IP 可以作为支持 OBEX 协议蓝牙设备的可选特性。

IrOBEX 提供了一个对象表示模型和一个会话协议。这两个协议确定了两设备间的会话框架。IrOBEX 协议遵循客户/服务器的请求/应答会话模式。

虽然 IrDA 同时定义了无连接 OBEX, 但蓝牙只使用面向连接的 OBEX。采用面向连接策略的原因在于:

- 在蓝牙体系结构中, OBEX 映射于面向连接的协议之上;
- 大部分采用蓝牙和 OBEX 技术的应用框架需要一个面向对象的 OBEX, 以提供这些应用框架中定义特性所描述的功能;
- 无连接的 OBEX 与面向连接的 OBEX 的通信都会带来互操作的问题。

图 1.1 解释了蓝牙体系结构的部分层次, 揭示了 OBEX 协议及其应用在该体系结构框架中所处的层次。该协议能够与服务搜索数据库进行通信。

蓝牙系统中, OBEX 协议的目的在于实现数据对象交换。典型例子是将业务卡片(business card)对象“推”到网络中其他实体。还有更复杂的例子, 如多个 OBEX 设备间的时钟同步问题。对于“推”对象和同步应用, 其内容格式可以是 vCard、vCal、vMessage 和 vNote 格式。vCard、vCal、vMessage 和 vNote 格式分别描述了电子业务卡片、电子时钟和计划、电子报文和邮件、电子便签等的格式。

蓝牙技术文本, 包括五个与 OBEX 及其应用有关的规定细则。

(1) 蓝牙 IrDA 互操作性规定, 包括:

- 定义应用如何同时在蓝牙和 IrDA 之上运行;
- 定义 OBEX 如何在 RFCOMM 和 TCP/IP 之间映射;
- 定义 OBEX over Bluetooth 的应用总体要求。

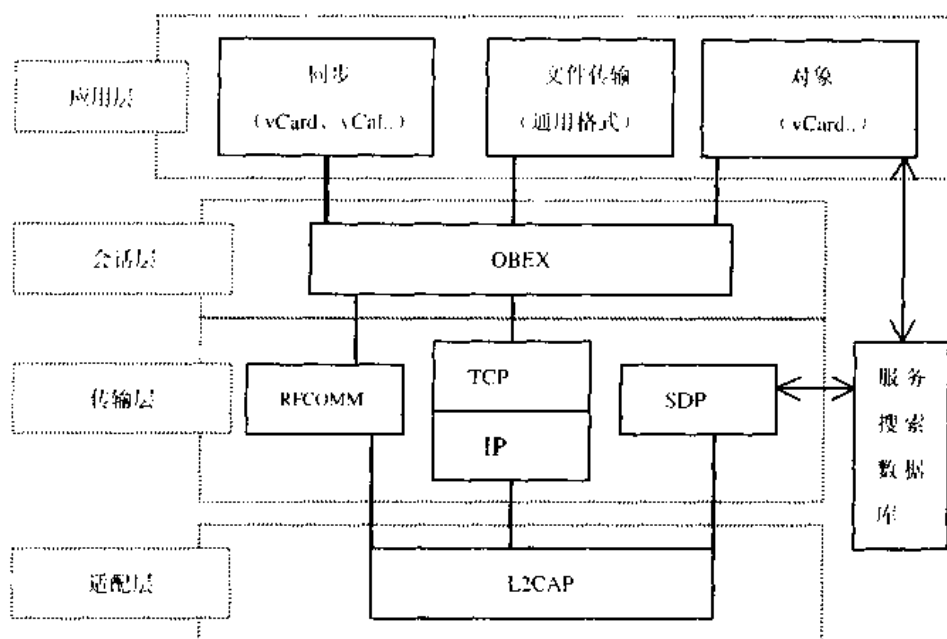


图 7.1 部分蓝牙协议层次

(2) 蓝牙通用对象交换框架规范，包括：

- OBEX 应用框架的一般性互操作的规范；
- 用于应用框架的协议低层（如基带和 LMP）的互操作性定义。

(3) 蓝牙同步应用框架规范，包括：

- 同步应用的应用框架；
- 对同步应用框架中的应用互操作性要求的定义；
- 不对基带、LMP、L2CAP、RFCOMM 的同步应用条件做出定义。

(4) 蓝牙文件传输框架规范，包括：

- 文件传输应用的应用框架；
- 对文件传输应用框架中的应用互操作性要求的定义；
- 不对基带、LMP、L2CAP、RFCOMM 的文件传输应用条件做出定义。

(5) 蓝牙“推”对象框架规定，包括：

- “推”对象应用的应用框架；
- “推”对象应用框架中的应用互操作性要求的定义；
- 不对基带、LMP、L2CAP、RFCOMM 的“推”对象应用条件做出定义。

除在 IP 之上，OBEX 还应用在 IrCOMM 和 Tiny IP 上。蓝牙技术并没有像针对 OBEX 定义传输层那样定义这些协议，但他们都可以得到独立软件供应商的支持。

7.2 OBEX 对象和协议

7.2.1 对象模型

OBEX 对象模型对 OBEX 对象做出描述。OBEX 协议能够通过“推”、“拉”操作传输对象。一个对象可以通过多个“推”请求和“拉”应答进行交换。

该模型处理对象及其有关信息。对象由对象头组成，对象头由若干个头 ID 和包含的值组成。头 ID 描述了对象头的组成及其格式，以及头 ID 所定义的各位值的格式和含义。头 ID 一般包括计数器(Count)、名字(Name)、类型(Type)、长度(Length)、时间(Time)、描述(Description)、目的地址(Target)、HTTP 协议、主体(Body)、主体结束标志 (End of Body)、宿主标识、连接 ID (Connection ID)、应用参数 (Application Parameter)、认证字 (Authenticate Challenge)、认证应答字 (Authenticate Response)、对象类别 (Object Class)，以及用户自定义头。细节参见 IrOBEX 规范的 2.2 节。

7.2.2 会话协议

OBEX 操作采用应答-请求模式。请求由客户端发出，由服务器端应答。在发出请求之后和发出下一个新的请求之前，客户端将等待服务器的应答。每一个请求分组由一个 1 字节长的操作码、一个 2 字节的长度标识和数据组成。每一个应答分组由一个 1 字节长的应答码、一个 2 字节长的长度标识和数据组成，其数据则可有可无。

1. 连接操作

当应用第一次请求发送 OBEX 对象时，则启动一次 OBEX 会话。一个 OBEX 客户启动一次 OBEX 会话建立过程。该会话自连接请求发出开始，请求格式如表 7.1 所示。

表 7.1 请求格式

0	1	2	3	4	5	6	7to n
0x80 位置码	连接请求分组长度		OBEX 版本号	标志	OBEX 分组最小长度		可选头

注释：PDU（请求和应答信息包）的比特序列格式在 OBEX 与在 IrOBEX 中一样，采用 Big Endian 码格式，即 MSB 在左边，LSB 在右边。

连接请求由在远程主机的 OBEX 服务器接收。服务器通过向客户端发出成功应答确认连接，通过发送其他应答信息到客户端表示建立连接失败。其应答格式如表 7.2 所示。

表 7.2 应答格式

0	1	2	3	4	5	6	7to n
应答码	连接请求分组长度		OBEX 版本号	标志	OBEX 分组最大长度		可选头

应答码如 IrOBEX 规范 3.2.1 节中所列。第 5 和 6 字节定义了 OBEX 报文最大长度，由服务器接收。该值与长度部分不同，以便能够为客户端接收。连接请求和应答报文尺寸和格式都应一致。

连接一旦建立便始终保持激活状态，只能通过由请求/应答或失败断开，也就是说，在所有 OBEX 对象完全传输后连接也不会自动断开。

2. 连接断开操作

当 OBEX 连接所需应用被关闭，或应用要改变目的主机的时候，OBEX 会话将断开。客户端将连接断开请求发往服务器。该请求格式如表 7.3 所示。

表 7.3 请求格式

0	Byte and 2	3byte
0x81	分组长	分组头 (可选)

服务器不能拒绝该请求，而且它还要发回应答，其格式如表 7.4 所示。

表 7.4 应答格式

0	1 Byte and 2	3byte
0xA0	应答分组长度	应答分组头 (可选)

3. PUT 操作

当服务器和客户端间的连接建立之后，客户端就可以向服务器“推”(push)对象了。“推”请求用于推一个 OBEX 对象。该请求格式如表 7.5 所示。

表 7.5 请求格式

0	Byte and 2	3byte
0x02 (当未位设定时为 0x82)	分组长度	分组序列

一个“请求”可由一个或多个请求分组组成，这取决于传送对象的大小和分组尺寸。每一个“推”请求分组都需要一个发白服务器的应答分组。组成一个 OBEX 对象的多个请求分组不能只有一个应答分组。其应答格式如表 7.6 所示。

表 7.6 应答格式

0	Byte and 2	3byte
应答码	应答分组长度	应答头 (可选)

4. GET 操作

连接在服务器和客户端间建立之后，客户端也可以从服务器拉 (pull) 对象。GET 操作就是用于“拉”OBEX 对象。该请求格式如表 7.7 所示。

表 7.7 请求格式

0	1 Byte and 2	3byte
0x03 (当未位设定时为 0x83)	应答分组长度	以名字起始的应答分组头 (可选)

对象以分组头序列方式返回，而客户端必须为每一个应答分组发送请求分组，其应答格式如表 7.8 所示。

表 7.8 应答格式

0	1 Byte and 2	3byte
应答码	应答分组长度	应答头（可选）

5. 其他操作

其他 OBEX 操作包括设置路径（SetPath）和放弃（Abort）。在 IrOBEX 规范 3.3.5-6 节对此做出了详细解释。需要说明的是，客户端可以在每一次应答后，甚至在请求/应答操作过程中发出放弃请求。而且，在发出放弃请求之前不必接收整个 OBEX 对象。

除了这些操作，IrOBEX 规范还可支持自定义操作，但在蓝牙技术中不作支持。

7.3 OBEX over RFCOMM

OBEX 在 RFCOMM 上的映射关系，基于 ETSI TS07.10 的多路复用和传输层，而且它提供了对串行电缆仿真的支持。支持 OBEX 协议的蓝牙设备需满足以下要求：

- 支持 OBEX 的设备可以单独作为服务器、客户端或者同时作为两者；
- 所有同时运行在一个设备上的服务器应用应各自使用其 RFCOMM 服务器通道。

使用 OBEX 的应用（服务/服务器）能够将信息在服务搜索库中注册，不同应用框架是在框架规范文件中定义的。

1. RFCOMM 上的 OBEX 服务器设置

当客户端发出一个连接请求时，服务器假定已经准备好接收请求。但是，在服务器准备接收和进入侦听状态之前，应满足以下前提条件：

- 服务器应打开一个 RFCOMM 服务器通道；
- 服务器必须将其容量注册到服务搜索库。

在此之后，主机才能找到所需服务器，服务器才能对客户端请求进行侦听。

2. 从串口接收 OBEX 分组

如上所述，一个对象可以通过一个或多个 PUT 请求和 GET 应答操作进行交换，也就是说，一个对象可以由一个或多个数据分组进行传输。然而，如果 OBEX 可以直接在串口运行，它就不会从 RFCOMM 数据分组。一个比特流则可以通过 OBEX 从 RFCOMM 仿真串口接收。

为了检测比特流中的一个数据分组，OBEX 查找是应答码还是操作码，取决于该数据分组是请求数据分组还是应答数据分组。操作码和应答码可以看作是数据分组的起始标志。OBEX 数据分组中不存在结束标志。数据分组长度信息则由操作码和应答码后的两个字节组成。因而，可以通过这样获得整个数据分组的长度，并确定两数据分组的边界。

所有未识别的数据都应被丢掉，而这会产生同步问题。但是根据 OBEX 协议的实质，这对于 RFCOMM 不是问题，反而提供了基于蓝牙的可靠传输。

3. 连接建立

由客户端初始化一个连接。但是，在客户端能够发出第一个数据请求前，需执行下列

任务：

- 通过使用 SDP 规定中的 SD 协议，客户端必须搜索到与要建立连接服务器相关的明确信息：

- 客户端利用搜索到的 RFCOMM 信道，建立 RFCOMM 连接；
- 客户端向服务器发出连接请求，以建立一个 OBEX 会话。客户端如接收到服务器发出的一个成功应答，会话就可以直接建立起来。

4. 连接断开

一个基于 RFCOMM 的 OBEX 会话可以直接通过连接断开请求断开。当客户端收到应答后，便关闭指定给 OBEX 客户的 RFCOMM 信道。

5. 在 RFCOMM 上推、拉 OBEX 分组

通过 PUT 请求在 RFCOMM 上利用 OBEX 数据分组传输数据。应答必须在每一次请求后和下一次请求之前发出。

通过发出 GET 请求从远程主机“拉”数据。数据分组含于 OBEX 应答数据分组。每次应答后，可以发出新的“拉”数据请求。

7.4 OBEX over TCP/IP

支持 OBEX Over TCP/IP 协议的蓝牙设备必须满足以下要求：

- 支持 OBEX 的设备可以单独作为服务器、客户端或同时作为两者；
- 服务器 TCP 端口号 650 由 IANA 指定。该端口号应小于 1023。一般推荐使用 IANA 定义的 TCP 端口号 650。0 和 1023 号由 IANA 保留使用；
- 客户端必须使用一个不在 0~1023 范围内的端口号；
- 使用 OBEX 的客户服务器应用必须能够将明确的信息注册到服务搜索库。不同应用的信息都在框架规定中阐述。

1. TCP/IP 的 OBEX 服务器设置

当客户端发出一个 PUT 或 GET 请求时，服务器假定已经准备好接收请求。但是，在服务器准备接收和进入侦听状态之前，应满足以下前提条件：

- 服务器应把 TCP 端口初始化为 650 或大于 1023 的值；
- 服务器必须将其容量注册到服务搜索库。

在此之后，主机才能找到所需的服务器，服务器才能对客户端请求进行侦听。

2. 连接建立

由客户端初始化一个连接。但是，在客户端能够发出第一个数据请求前，需执行下列任务：

- 通过 SDP 规范中的 SD 协议，客户端须搜索到与建立连接服务器相关的明确信息；
- 客户端初始化大于 1023 的 TCP 端口号相关套接字，并与服务器主机建立一个 TCP 连接；
- 客户端向服务器发出连接请求，以建立一个 OBEX 会话。客户端如果接收到服务器

发出的一个成功应答，会话可以直接建立起来。

3. 连接断开

一个基于 RFCOMM 的 OBEX 会话可以直接通过连接断开请求断开。当客户端收到应答后，便关闭指定给 OBEX 客户的 RFCOMM 通道。

7.5 利用 OBEX 的蓝牙应用概述

蓝牙 SIG 定义了三种不同的 OBEX 应用框架。下面就对这些框架进行简要介绍。

1. 同步

同步就是通过比较和调整两个对象存储的时钟操作并使之一致。支持同步的蓝牙设备可以是 PC、笔记本电脑、PDA、移动电话和无绳电话。

蓝牙同步框架面向兼容于 IrMC 同步(IrDA 制定)的服务器和客户应用。蓝牙同步服务器和客户支持 IrMC 规定的 LEVEL 4 同步功能。在客户端设备上实现同步运算的同步引擎机制可以根据实际情况制定。这一点由软件服务商提供，而不列入蓝牙标准。

同步服务不只限于一种类型的应用。蓝牙同步服务(IrMC 同步)支持四种不同的服务类型：

- (1) 通讯录——提供管理通讯录的服务；
- (2) 日历——便于用户对日程和任务进行管理；
- (3) 发消息——管理用户信息，如 E-mail；
- (4) 记事本——管理用户短信息。

蓝牙同步标准的互操作要求在同步标准和通用对象交换框架标准中进行定义。

2. 文件传输

文件传输标准用于向蓝牙设备发送和从蓝牙设备接收通用类型文件。文件传输服务支持浏览远程蓝牙设备文件夹。

蓝牙文件传输标准的互操作性要求在文件传输标准和通用对象交换标准中进行定义。

3. 对象“推”操作

对象推操作标准是用于发送对象和有选择地“拉”缺省对象的文件传输标准的特例。它可以提供业务卡片交换等服务。

蓝牙对象“推”操作框架的互操作性要求，在对象推操作标准和通用对象交换标准规范中进行定义。

第 8 章 电话控制二进制协议

本章采用面向比特的二进制协议规范，描述蓝牙电话控制协议二进制规范(TCS 二进制)。该协议定义用于在蓝牙设备间建立语音会话和数据呼叫的呼叫控制信号，以及用于处理蓝牙 TCS 设备的移动管理过程。

8.1 概述

蓝牙电话控制二进制协议（TCS 二进制）基于 ITU-T 推荐书 Q.931 制定，其先前版本发表在 Q.931 附录上。该最终文本不区分用户方和网络方，而只是区分呼出方（呼叫发起方）和呼入方（呼叫终止方）。同时，该文本只针对蓝牙和可预见的应用做出必要的修改，以最大限度地重用 Q.931 的内容。

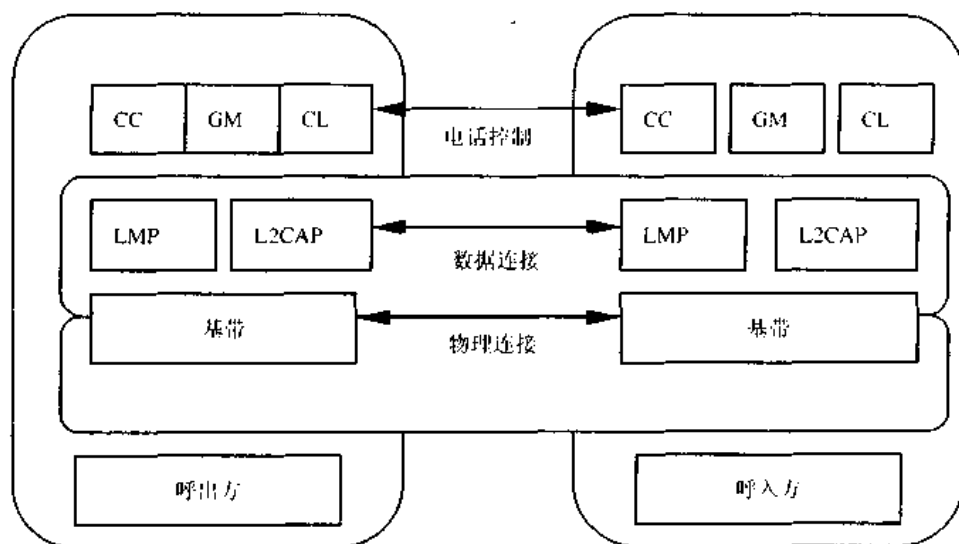


图 8.1 蓝牙协议栈中的 TCS

TCS 包括以下功能：

- 呼叫控制（CC）——指示蓝牙设备间语音会话和数据呼叫的建立和释放；
- 组管理（GM）——方便蓝牙设备组的处理；
- 无连接 TCS（CL）——与非正在进行的呼叫进行相关信令信息交换的条款。

8.1.1 设备间操作

TCS 采用点到点通信和点到多点的通信模式。在已知要建立呼叫的目标蓝牙设备的情况下，使用点到点信号。如果有多个可用于建立呼叫的目标蓝牙设备，可使用点到多点信号。

点对点信号映射于一个面向连接的 L2CAP 通道，点到多点信号映射于无连接的 L2CAP，但后者在匹克广播通道上以广播信息的形式发送。

图 8.2 是点对点信号在单点配置方式下建立语音和数据呼叫的过程。首先，利用点对点

信道 A 通知另一个设备有呼叫请求；第二步，在该信道上建立语音会话或数据信道。

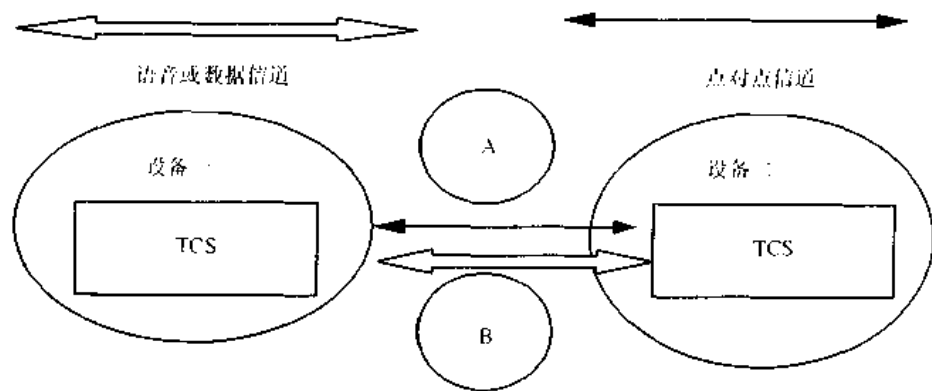


图 8.2 单点配置中点到点信令

图 8.3 表示了利用点到多点信号和点对点信号在多点配置方式下如何建立语音或数据呼叫的过程。首先，利用点对多点信道 A 通知另一个设备有呼叫请求；第二步，在该信道上建立会话或数据信道。

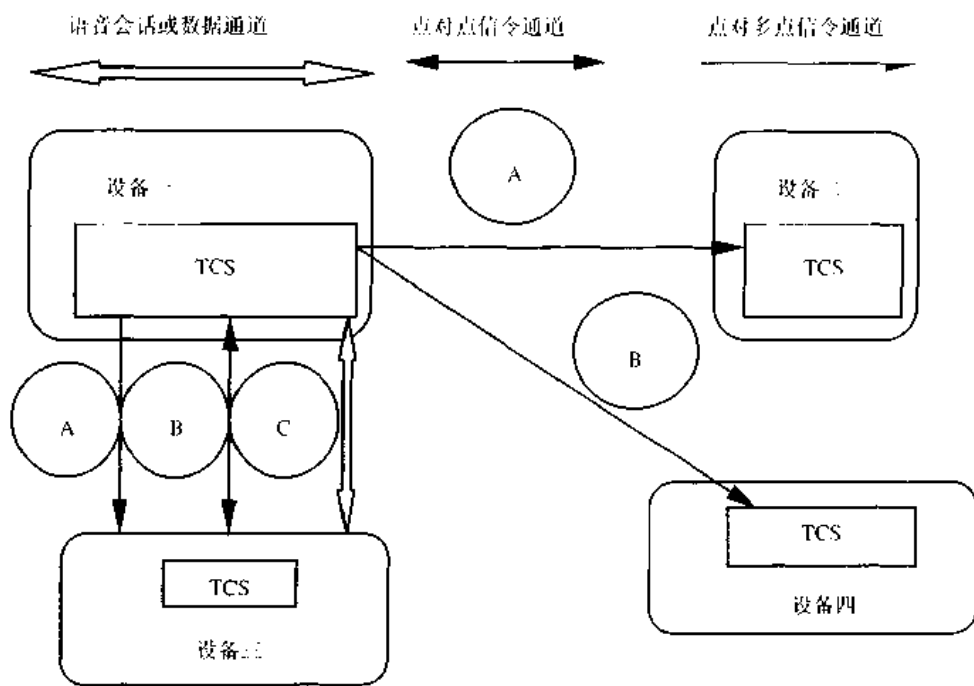


图 8.3 多点配置信令

8.1.2 层间操作

TCS 的执行版本在图 8.4 所示的通用体系结构中描述（为了简化，数据呼叫处理没有画入）。

二进制 TCS 内部结构包括功能实体呼叫控制、组管理和无连接。作为补充还包括 TCS 内部协议识别码不同的协议，以及到功能实体的路径流量控制。

为处理更多的并发呼叫，可以同时存在二进制 TCS 的多个实例。各实例之间根据 L2CAP 通道标识相互区别。

二进制 TCS 为多个蓝牙实体提供接口，以向应用提供电话服务。该接口如图 8.4 所示。

信息通过这些接口交换，以实现：

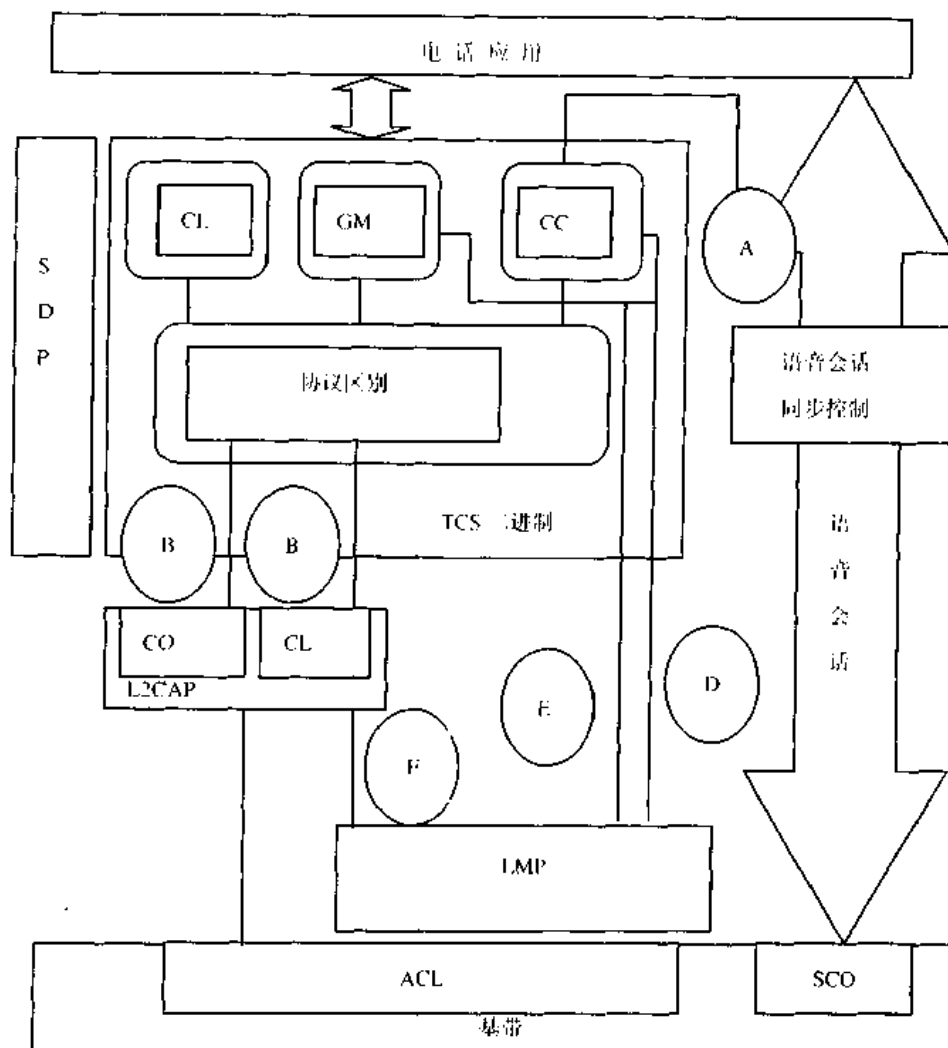


图 8.4 TCS 体系结构

A. 当连接到会话路径时，呼叫控制实体向会话同步控制提供信息。该信息基于呼叫控制信息，如接受连接确认指令（CONNECT ACKNOWLEDGE）和连接断开指令（DISCONNECT）。

B. 利用点到多点通信发送 SETUP 报文，通过 L2CAP 接口在无连接通道中进行传输，并利用该接口通知 TCS 已收到利用无连接通道传输的 SETUP 报文。无连接的 L2CAP 信道映射于匹克广播通道；

C. 无论何时利用点对点通信发出 TCS 报文，该报文都是通过 L2CAP 接口在面向连接通道上进行传输。在 L2CAP 信道建立过程中，必须指定连接服务质量，特别是节能模式的使用（L2CAP 将接口 F 有关信息通知 LMP）。

D. 为了建立和释放 SCO 链接，呼叫控制实体应直接控制 LMP；

E 和 G. 组管理实体为了在初始化过程中控制查询、呼叫访问和匹配而直接控制 LMP 和 LC/基带。

8.2 呼叫控制（CC）

8.2.1 呼叫状态

对于用户端，TCS 使用的呼叫状态在 Q.931 中定义。对于 TCS，由于计算资源限制，只要求使用该状态集的一个子集。该子集命名为精简 TCS。该协议集如下。

通用状态：

- Null(0) 空状态
- Active(10) 激活状态
- Disconnect request(11) 连接断开请求
- Disconnect indication(12) 连接断开指示
- Release Request(19) 释放请求

呼出端状态：

- Call Initiated (1) 初始化呼叫
- Overlap sending (2) 重发
- Outgoing call proceeding (3) 呼叫进行
- Call delivered (4) 呼叫转发

呼入端状态：

- Call present(6) 正在呼叫
- Call received(7) 接受状态
- Connect request(8) 连接请求
- Incoming Call proceeding (9) 呼入呼叫正在进行
- Overlap receiving(25) 重复接收

这些状态及它们之间的转换都在附录 1-TCS 呼叫状态中阐述。为了描述清楚，精简 TCS 对各状态信息又分别给予解释。

8.2.2 呼叫建立

面向连接的 L2CAP 必须在呼叫控制程序开始运作之后，在呼出端和呼入端之间建立通道。而且，在多点配置当中，必须在呼出端和呼入端之间建立无连接 L2CAP 通道。

1. 呼叫请求

发送端通过发送 SETUP 报文和启动 T303 定时器初始化呼叫，如图 8.5 所示。

在点对点配置情况下，通过面向连接的通道传输 SETUP 报文。

在多点配置情况下，通过无连接通道传输 SETUP 报文。而该 SETUP 报文在每一结点上以广播报文形式传输。

如果在 T303 定时器失效之前，没有收到从呼入端发回的应答报文，呼出端：

- 如果 SETUP 报文通过无连接通道传输，将返回 NULL 状态，终止传输 SETUP 报文；
- 如果 SETUP 报文通过面向连接的通道传输，将向呼入端发送 RELEASE COMPLETE 报文。该报文将包括#102 号事件 recovery on timer expiry（当定时器失效时恢复）。

SETUP 报文通常包括呼叫类别，以及呼入端需要的所有信息。如果被呼叫方号码信息位数不够，则需要重新发送。SETUP 报文应包括完整的号码信息。

在 SETUP 报文发送之后,呼出端则进入呼叫初始化 (Call Initiated) 状态。呼入方在接收到 SETUP 报文后进入呼叫进行 (Call present) 状态。

2. 选择信道类型

在呼叫请求中发送的 SETUP 报文可以包括信道容量信息元,以表示被请求信道。接收方通过 SETUP 报文的第一个应答报文中包含信道容量信息对被请求信道进行协商。

信道容量信息元表示如何在呼叫中利用低层资源 (信道)。如果信道为 ‘同步面向连接’ (SCO) 类别,将采用 SCO 链路,并使用给定数据分组类别和用于语音会话呼叫的语音编码。如果信道为 ‘异步无连接’ (ACL) 类别,将使用 ACL 链路。在此之上是用于数据呼叫的具有 QoS 要求的 L2CAP 通道。如果信道类别信息为 ‘NONE’,就不会单独建立信道。

3. 重发

如果接收到的 SETUP 报文不包括发送完成指示信息元,并存在以下两种情况中的一种:

- 被叫号码信息不完整;
- 呼入方不能确认被叫号码信息完整。

那么,呼入方将启动定时器 T302,并向呼出方发送 SETUP ACKNOWLEDGE 报文,并进入重复接收状态。

当接收到 SETUP ACKNOWLEDGE 报文时,呼出方将进入重复发送状态,并中止定时器 T302,而启动定时器 T304。

在接收到 SETUP ACKNOWLEDGE 报文后,呼出方将采用被叫号码发送其余信息。该信息可以是一条或多条 INFORMATION 报文。

当每条 INFORMATION 报文发出时,呼出方将重新启动定时器 T304。

完成信息发送任务的最后一条 INFORMATION 报文将包括一个发送结束标志。如果呼入方不能确定被叫号码是否完整,那么将在接收到每一条不包含发送结束标志的 INFORMATION 报文时重新启动定时器 T302。

在定时器 T304 失效时,呼出方将初始化呼叫清除过程。该过程同时触发#102 事件。

在定时器 T302 失效时,呼入方:

- 当呼入方无法确认呼叫信息是否完整时,将初始化呼叫清除过程,并触发#28 事件非法号码格式;
- 否则,呼入方将回复一个 CALL_PROCEEDING_ALERTING 或 CONNECT 报文。

4. 呼叫进行

a. 进行整块发送

如果使用整块发送(如呼入方能够确定它从呼出方接收到的 SETUP 报文中包含了全部建立呼叫所需信息),呼入方就会向呼出方发送一个 CALL PROCEEDING 报文,以确认收到 SETUP 报文和表示呼叫正在进行。当收到 CALL PROCEEDING 报文时,呼出方就进入呼出呼叫进行状态,并中止定时器 T302,启动定时器 T304。发送 CALL PROCEEDING 报文后,呼入方就会进入呼入呼叫进行状态。

b. 呼叫进行,重复发送

当以下情况发生时:

- 呼入方接收到报文发送完毕指示;

- 呼入方认为所有影响呼叫建立的呼叫信息都已收到。

呼入方就会向呼出方发送一个呼叫进行报文，并中止定时器 T302，进入呼入呼叫进行状态。

当收到 CALL PROCEEDING 报文时，呼出方就进入呼出呼叫进行状态，并中止定时器 T304，如果可能将启动定时器 T302。

c. 定时器 T310 失效

定时器 T310 失效时，如呼出方没有收到 ALERTING、CONNECT、DISCONNECT 或 PROGRESS 报文时，呼出方将按照 8.2.3 节中的#102 事件“定时器失效时恢复”初始化呼叫清除。

5. 呼叫确认

当呼入方接收到被叫地址上的用户报警时，就会发出 ALERTING 报文，并进入呼叫接收状态。当呼出方接收到 ALERTING 报文时，呼出方将启动一个内部生成的报警指示，并进入呼叫传递状态。呼出方将启动定时器 T304，以避免重复发送，中止定时器 T303 或 T310（如果正在运行），启动定时器 T301（如果没有另一内部报警优先级更高的定时器时）。

T301 定时器失效时，呼出方将按照 8.2.3 节中的#102 事件“定时器失效时恢复”初始化呼叫清除。

6. 呼叫连接

呼入方通过向呼出方发送 CONNECT 报文和终止用户报警表示接受呼入呼叫。当发送 CONNECT 报文时，呼入方将启动定时器 T313。

当接收到 CONNECT 报文时，呼出方将终止任一个内部生成的报警信息，终止定时器 T301、T303、T304 和 T310，建立到呼出方的被叫信道，并发送连接确认报文和进入激活（Active）状态。

CONNECT ACKNOWLEDGE 报文标志被叫信道的建立。当收到 CONNECT ACKNOWLEDGE 报文，呼入方将连接到信道，终止定时器 T313，并进入激活状态。

收到 CONNECT ACKNOWLEDGE 报文之前且时钟 T313 失效时，呼入方将按照#102 事件“定时器失效时恢复”初始化呼叫清除。

7. 呼叫信息

处于激活状态时，发送方和接收方可以交换与当前呼叫有关的信息，该呼叫使用信息（INFORMATION）报文。

8. 主动的用户清除

当在多点配置的无连接信道上传输呼叫时，除了向呼入方发送 CONNECT ACKNOWLEDGE 报文，呼出方还要向其他为应答 SETUP 报文而发送 SETUP ACKNOWLEDGE、CALL PROCEEDING、ALERTING 或 CONNECT 报文的呼入方发送 RELEASE 报文。该 RELEASE 报文通知这些呼入方将不再向它们提供呼叫。

9. 带内语音和广播

当呼入方提供带内语音和广播，并且如果被呼叫信道正在进行语音呼叫，呼入方将首

先立即建立信道。然后，则与带内语音和广播同时发送进度指示#8 带内信息或合适模式(in-band information or appropriate pattern)。该进度指示也可以包含在任一允许包含进度指示信息元的控制报文中，或者在呼叫状态没有变化的情况下也可包含在 PROGRESS 报文中。

当收到该报文后，呼出方就可以连接到通信通道以接收带内语音/广播信息。

10. 呼叫建立失败

在当前呼叫中如果有重复接收、呼叫进行或呼叫已接收状态的情况，呼入方将初始化呼叫清除过程。下面的值将在重复接收、占线情况下终止当前呼叫。

- #1 未分配的号码
 - #3 无到目的地址的路径
 - #17 用户忙
 - #18 无用户应答
 - #22 号码改变
 - #28 非法号码格式（非完整号码）
 - #34 无可回路/信道
 - #44 无可被请求回路/信道
 - #58 当前无可信道容量
 - #65 当前无信道容量
- 当呼入方处于呼叫已接收状态时，清除当前呼叫可使用下列事件值：
- #19 无来自用户的回答（提醒用户）
 - #21 用户拒绝呼叫

11. 呼叫建立报文流

图 8.5 提供了在成功建立呼叫过程中交换信息的完整视图。其中，实线表示精简 TCS 的一部分，即必需的报文；虚线表示可选报文；三角形表示正在运行的定时器。

8.2.3 呼叫清除

1. 正常呼叫清除

除了下节说明的例外情况以外，呼出方和呼入方都可以启动呼叫清除过程。为了叙述清楚，下面只对呼出方初始化呼叫清除过程进行描述。

当发送或接收到任何呼叫清除报文，应终止除了 T305 或 T308 以外的协议定时器。

呼出方通过发送 DISCONNECT 报文，启动定时器 T305，从信道断开，进入请求断开状态，从而初始化呼叫清除过程。

呼入方将在接收到 DISCONNECT 报文时进入断开指示状态。该报文通知呼入方断开与信道的连接。一旦用于呼叫的通道被断开，呼入方将向呼出方发送 RELEASE 报文，启动定时器 T308，并进入释放请求状态。

接收到 RELEASE 报文后，呼出方将取消定时器 T305，释放占用信道，发送 RELEASE COMPLETE 报文，返回 NULL 空状态。

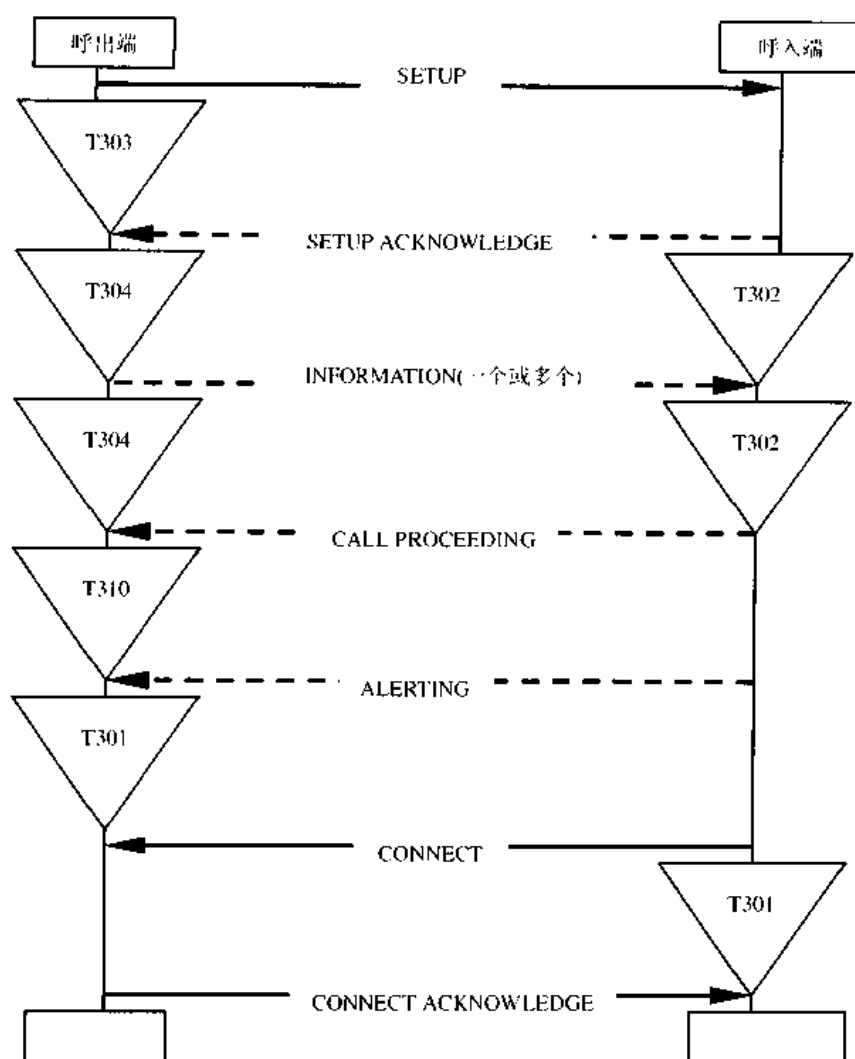


图 8.5 建立呼叫报文流

从呼出方接收到 **RELEASE COMPLETE** 报文后，呼入方将终止定时器 **T308**，释放占用信道，返回 **NULL** 空状态。

如果呼出方在定时器 **T305** 失效之前没有收到应答 **DISCONNECT** 报文的 **RELEASE** 报文，它将向呼入方发送包含 **DISCONNECT** 报文中呼叫号码的 **RELEASE** 报文，启动定时器 **T308**，进入释放请求状态。

如果一方处于释放请求状态，且在定时器 **T308** 失效之前没有收到 **RELEASE COMPLETE** 报文，它将返回 **NULL** 状态。

除了上述呼叫清除过程以外，如果被请求信道正在进行语音呼叫，呼出方将在呼叫清除阶段采取带内语音/广播信息。当提供了带内语音/广播信息时，呼出方将终止占用信道（如果该信道未使用），然后发送包含进度指示#8“带内信息或合适模式”的连接断开报文。

接收到该报文后，呼入方就可以接入信道，并接收带内语音/广播信息，进入连接断开指示状态。

呼入方在收到呼出方发出的 **RELEASE** 报文之前，将通过与信道断开连接，启动定时器 **T308**，进入释放请求状态来进行呼叫清除过程。

2. 非正常呼叫清除

正常情况下，呼叫清除由发送 DISCONNECT 报文的任何一方进行初始化，其过程在上节中进行定义。上述规则的惟一例外如下所列：

- 为应答 SETUP 报文，呼入方可以因为无可利用资源等原因而拒绝呼叫。如果没有其他应答信息发出的话，拒绝呼叫过程通过发送 RELEASE COMPLETE00 报文实现，然后呼入方进入 NULL 空状态；
- 在多点配置情况下，可以通过呼出方发出的 RELEASE 报文进行非用户主动选择的呼叫清除；
- 在多点配置情况下，SETUP 报文通过无连接通道发送。如果呼入方在呼叫建立过程中接收到远程呼叫用户的断开指令，无论呼入方已经应答还是正要应答，呼叫都将被 RELEASE 报文清除掉，呼叫清除过程如上节所述。呼出方将在呼叫清除过程完成后进入 NULL 空状态。

3. 清除冲突

当呼入方和呼出方同时发出 DISCONNECT 报文时将发生清除冲突。当任一方在连接断开请求状态下收到 DISCONNECT 报文时，该方将终止定时器 T305，如果信道连接没有断开就断开信道连接，并发送 RELEASE 报文，启动定时器 T308，进入释放请求状态。

清除冲突也有可能发生在呼叫双方同时发送 RELEASE 报文时发生。处于释放请求状态和接收到 RELEASE 报文的实体将终止定时器 T308，释放信道，并进入 NULL 空状态，而不再需要发送 RELEASE COMPLETE 报文。

4. 呼叫清除报文流

图 8.6 提供正常呼叫清除情况下报文交换的完整视图。所有的报文都是强制要求的。

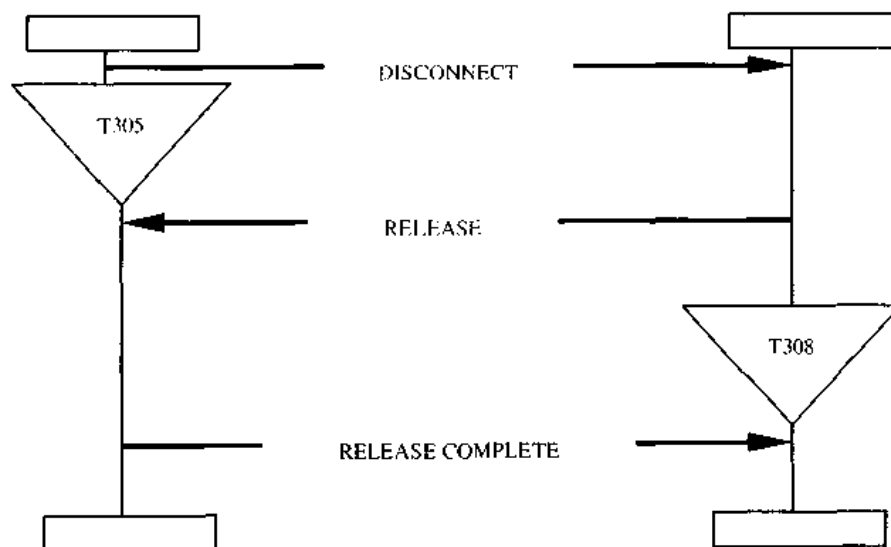


图 8.6 呼叫清除报文流

8.3 组管理 (GM)

组管理实体提供了管理一组设备的程序，如下所述。

- 获取访问权限，使被请求设备能够使用组里其他设备的电话服务；
- 配置分布，使处理和操作一组设备成为可能；
- 组员间快速访问，实现同组设备间的快速通信。

在任一组管理过程实现之前，应首先建立设备间面向连接的 L2CAP 通道。

对于组管理，将使用无线用户组（WUG）的概念。

8.3.1 无线用户组（WUG）

一个 WUG 由多个支持 TCS 的蓝牙单元组成，其中一台设备称为 WUG 管理员。WUG 管理员实质就是一个典型的网关，以提供给组内其他蓝牙设备（称为 WUG 成员）访问外部网络的能力。所有范围内 WUG 成员都是一个激活或休眠的匹克网成员。该匹克网管理员通常也就是 WUG 管理员。

WUG 的主要特点是：

- 所有 WUG 内的单元互相都知道谁是 WUG 管理员，谁是 WUG 成员。所有的 WUG 成员都从 WUG 管理员那里接收到这些配置信息；
- 当一个新的单元能够与 WUG 管理员通信时，它也能够与其他任一 WUG 成员通信和进行身份验证与加密，而不需要进一步的匹配/初始化。WUG 管理员为各成员提供必要的身份验证和加密参数。

所有有关特性都通过配置分布过程进行维护。

1. WUG 的加密

为在无连接 L2CAP 通道上进行加密传输，WUG 管理员发布了一个临时密钥（ K_{master} ），因为一个蓝牙单元不能在两个或多个加密字之间实时切换，该密钥通常也能在面向连接通道上进行加密传输。该通道实行单独编址通信。由于 WUG 管理员将可能不间断地进行周期操作，因此 K_{master} 将进行周期变动。

为了允许不同 WUG 成员间进行身份认证和加密，WUG 管理员将使用配置分布发放链接字，以使 WUG 成员能够用来相互通信。建立通信的两个 WUG 成员之间只能使用唯一的链接字。

配置分布通常通过加密链接进行。因此与其说 K_{master} 用于加密，不如说是给已知地址的 WUG 成员使用的类似于参数的密钥。

2. 随机匹配

对于 TCS，与 WUG 管理员匹配也就意味着与所有 WUG 成员匹配。这一点通过配置分布实现。同时，这也避免了该外部设备分别与 WUG 中的每一成员单独匹配。

在蓝牙中，匹配不只是与特定服务匹配，同时也是与特定设备匹配。建立匹配关系后，如果没有禁止特定的应用或设备，则可以访问所有由该设备提供的服务。

如果没有其他的问题，将一个设备与 WUG 管理员匹配，也就意味着该设备提供的服务可以由所有 WUG 成员访问。反之亦然，该设备也可以访问所有 WUG 成员提供的服务。

因此，在使用 TCS（特别是配置分布）时，建议加入以下规范条款：

- 一个进入 WUG 的新的设备，不必通过初始化获取访问权限过程而成为该 WUG 的成员，而只要能够使用由 WUG 管理员提供的服务就可以了；

- WUG 管理员可以拒绝一个要求获得访问权限的请求；
 - WUG 成员在配置分布过程中不必接收配对信息。
- 这些要求不只应用于提供 TCS 相关服务的设备。

8.3.2 获取访问权限

利用获取访问权限过程，一个设备能够获得使用另一个 WUG 设备电话服务的权限。

一个设备通过发送 ACCESS RIGHTS REQUEST 报文和启动定时器 T401 来请求访问权限。接收方设备在接收到 ACCESS RIGHTS REQUEST 报文后，通过发送 ACCESS RIGHTS ACCEPT 报文接受请求。

请求方设备接收到 ACCESS RIGHTS ACCEPT 报文时，启动定时器 T401。这样，整个访问权限过程就成功完成。

如果定时器 T401 失效前没有收到应答，请求方设备将重新考虑访问权限请求。

如果在收到 ACCESS RIGHTS REQUEST 报文时，接收方设备由于某种原因不能接受访问权限，它将应答 ACCESS RIGHTS REJECT 报文，请求方设备也将终止定时器 T401 并重新考虑访问权限请求。

图 8.7 给出了在获取访问权限过程中交换报文的完整视图。

8.3.3 配置分布

WUG 中发生的诸如添加或删除一个单元等变化都需要随时通知 WUG 中所有单元，配置分布即用于交换该数据。

当发生 WUG 配置变动时，WUG 管理员将启动每一个 WUG 成员的配置分布过程。WUG 管理员将追踪任何一个被通知到 WUG 配置发生变动的 WUG 成员。

可能会有一些 WUG 成员在通信范围之外而未被通知到，但当它们重新与 WUG 管理员建立关联后将更新这些成员。

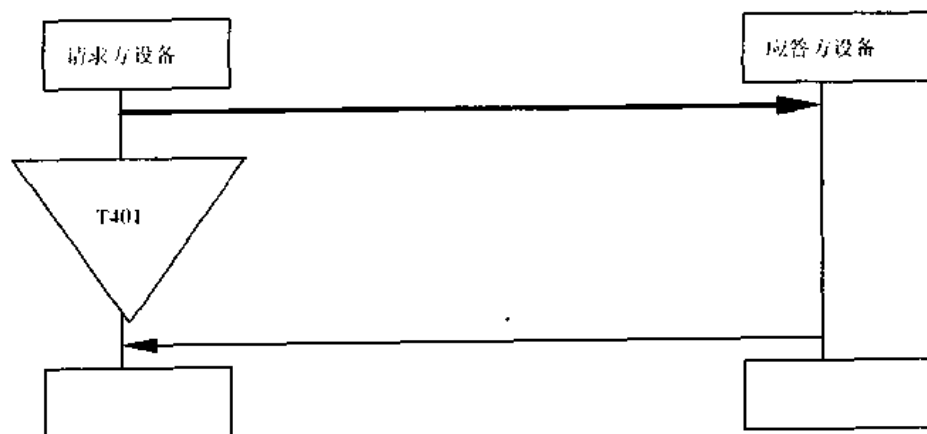


图 8.7 获取访问权限报文流

图 8.8 给出了在配置分布情况下进行消息交换的完整视图。

WUG 管理员通过启动定时器 T403 和发送 INFO SUGGEST 报文初始化配置分布过程。INFO SUGGEST 报文包含完整的 WUG 配置信息。收到 INFO SUGGEST 报文后，WUG 成员将发送 INFO ACCEPT 报文，以确认收到明确的 WUG 配置信息。

当 WUG 管理员收到 INFO ACCEPT 报文后，将终止定时器 T403，从而成功完成配置

分布过程。在定时器 T403 失效后，配置分布过程也将被终止。

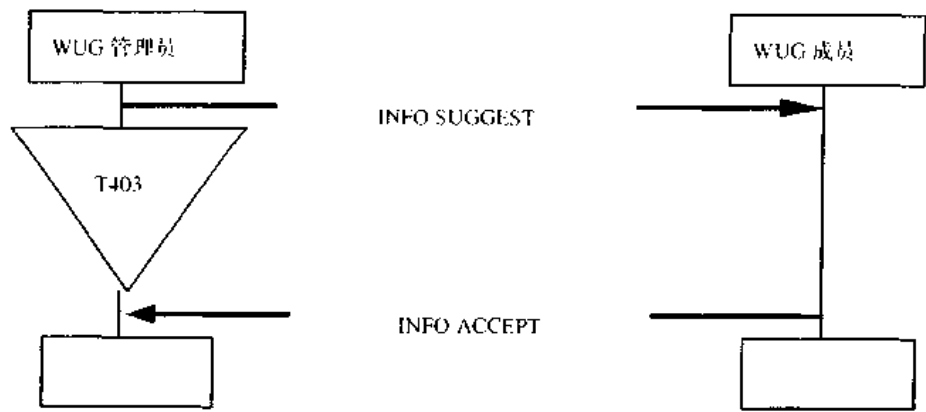


图 8.8 配置分布报文流

8.3.4 成员间快速访问

当 WUG 主匹克网中的两个成员都处于激活状态时，其中一个 WUG 成员就能够使用成员间快速访问过程快速访问另一方。通过成员间快速访问过程，宿主成员将从目的方获取时钟信息，并强制目的方在指定时间（T406）内进入 PAGE_SCAN 状态，如图 8.9 所示。

1. 请求侦听

宿主 WUG 成员通过启动定时器 T404 和向 WUG 管理员发送 LISTEN REQUEST 报文，指出希望建立联系的 WUG 成员，从而初始化成员间快速访问过程。

在定时器 T404 失效前，宿主方如果没有收到 LISTEN REQUEST 的应答报文，将终止该成员间快速访问过程。

2. 接收侦听

收到 LISTEN REQUEST 报文时，WUG 管理员会判断出被访问一方是否是 WUG 成员。如果是，WUG 管理员就会通过启动定时器 T405，向目的方发送 LISTEN SUGGEST 报文，从而初始化成员间快速访问。

当收到 LISTEN SUGGEST 报文时，目的 WUG 成员将通过向 WUG 管理员发送 LISTEN ACCEPT 报文确认该内部呼叫。该报文包括目的 WUG 成员的时隙信息。发送 LISTEN ACCEPT 报文后，目的 WUG 成员将进入呼叫扫描状态，持续 T406 时长，从而由宿主 WUG 成员建立连接。

收到 LISTEN ACCEPT 报文时，WUG 管理员终止定时器 T405，同时通过发送 LISTEN ACCEPT 报文通知宿主 WUG 成员间快速访问的结果。LISTEN ACCEPT 报文包括目的 WUG 成员的时隙信息。收到 LISTEN ACCEPT 消息时，宿主 WUG 成员将终止定时器 T404，并开始呼叫目的 WUG 成员。

定时器 T405 第一次失效前，如果 WUG 管理员没有收到 LISTEN SUGGEST 报文的应答报文，WUG 管理员将利用#102 事件，通过向宿主方和目的方 WUG 成员发送 LISTEN REJECT 报文，终止成员间快速访问过程。

3. 由 WUG 管理员执行的侦听拒绝过程

如果 WUG 管理员拒绝成员间快速访问过程，它就会向宿主 WUG 成员发送 LISTEN REJECT 报文。合法事件值为：

#1, Unallocated(unassigned)number(当给定 WUG 成员并非 WUG 成员时使用)

#17, User busy(在目的 WUG 成员正与外部呼叫关联时使用)

#20, Subscriber absent(在与目的 WUG 成员建立关联失败时使用)

以及任一由目的 WUG 成员发出或接收到的 LISTEN REJECT 报文中包括的事件值。

收到 LISTEN REJECT 报文时，宿主 WUG 成员将终止定时器 T404，并终止该过程。

4. 由 WUG 成员执行的侦听拒绝过程

如果目的 WUG 成员拒绝了收到的 LISTEN SUGGEST 报文中的建议动作，它就会向 WUG 管理员发一则 LISTEN REJECT 报文。合法事件值为#17“用户忙”。

接收到 LISTEN REJECT 消息时，WUG 管理员终止定时器 T405。

5. 报文流

图 8.9 给出了在内部成员快速访问过程中交换报文的一个视图。成功的内部成员快速访问过程终止于目的 WUG 成员进入呼叫扫描状态时，以便允许目的 WUG 成员直接访问它。

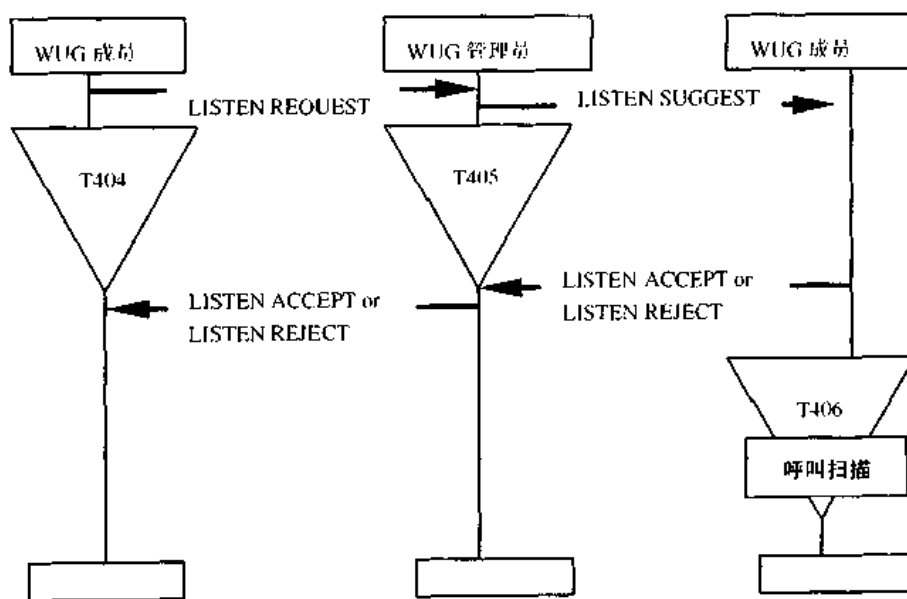


图 8.9 内部成员快速访问报文流

8.4 无连接 TCS (CL)

无连接 TCS 报文不需要建立 TCS 呼叫就能够用来交换信号信息。这也是一个 TCS 提供的无连接服务。一个无连接 TCS 报文就是一个 CL_INFO 报文（参见 8.6.3 节定义），如图 8.10 所示。

发送 CL_INFO 报文之前，可以利用呼出方和呼入方向的面向连接的 L2CAP 通道。

注：面向连接通道可以推迟通道的终止时间，以获得更多时间交换更多的 CL_INFO 报文。

在多点配置中，CL_INFO 发送前，可以利用无连接的 L2CAP。

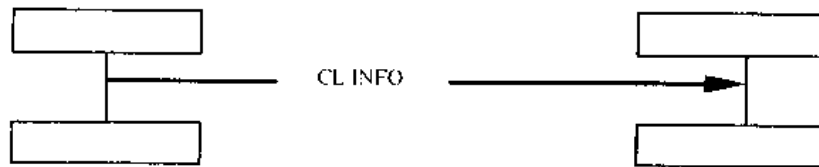


图 8.10 无连接 TCS 报文流

8.5 补充服务 (SS)

TCS 只明确提供一种补充服务，即呼叫线路识别。

对于外部网络提供的补充服务，可以利用 DTMF 序列实现附加服务的激活/失效和查询，支持 DTMF 启动/终止过程（见 8.5.2 节）。该过程支持完整和不完整的语音长度。

8.5.1 呼叫线路识别

为了通知呼入方呼叫发起方的唯一标识，呼出方将把呼叫请求的一部分——主叫号码信息单元包含在传输的 SETUP 报文中，如图 8.11 所示。

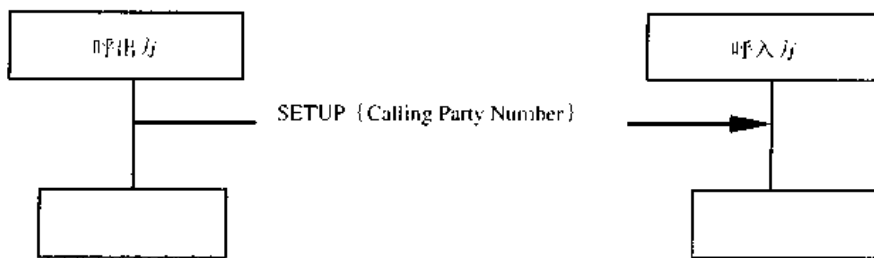


图 8.11 呼叫线路识别报文流

8.5.2 DTMF 启动和终止

DTMF 启动和终止过程支持在 PSTN 网络上，提供补充服务控制。

从 DTMF 原理上说，报文可以由呼入方和呼出方中任一方接收，但在实际应用中，一般都是由接入外部网络的一方（网关）接收。

DTMF 报文只能在呼叫激活状态下进行传输，语音生成过程在呼叫断开时也将终止。

1. 启动 DTMF 请求

用户可以通过某种手段生成 DTMF 语音，例如使用关键字解压缩。相关操作将作为要在已建立信道上以 START DTMF 报文形式发送 DTMF 数据位的请求进行解释。该报文包括需要传输的数据位的值(0, 1, 2, ..., 9, A, B, C, D, *, #)。

每一 START DTMF 报文中只传输单一数据位。

2. 启动 DTMF 应答

收到 START DTMF 报文的一方将重新将接收到的数据位恢复为远程用户可使用的 DTMF 语音，并向初始化端返回 START DTMF ACKNOWLEDGE 报文。该确认表示已成功传输。

3. 终止 DTMF 请求

当用户表示 DTMF 发送应结束时(如通过释放关键字), 初始化方将向其他方发送 STOP DTMF 报文。

4. 终止 DTMF 应答

收到 STOP DTMF 报文后, 接收方将停止发送 DTMF 语音, 并向初始化方返回 STOP DTMF ACKNOWLEDGE 报文。

5. 报文流

图 8.12 给出了在需要生成单个 DTMF 时交换报文的视图。

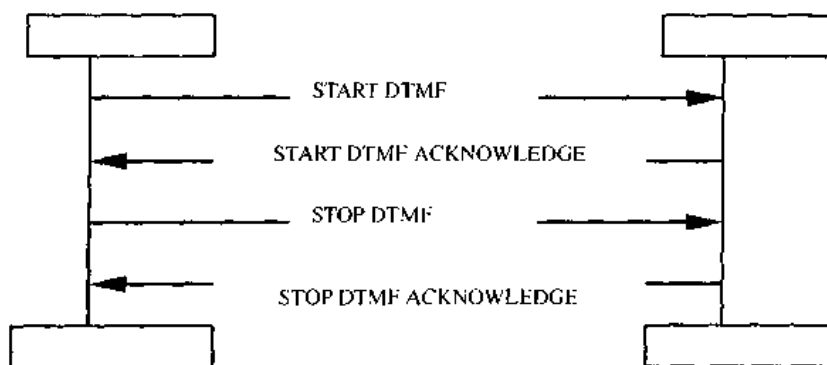


图 8.12 DTMF 启动/终止报文流

8.5.3 注册重呼

注册重呼指利用语音拨号获取注册, 以准许输入数据或其他操作。注册重呼通过发送含有键盘信息元素的 INFORMATION 报文, 实现注册重呼 (16 位)。以后的数据发送按照上节所示程序进行。

8.6 报文格式

本节提供报文结构完整视图, 定义每一类报文的功能和信息内容。

按照呼叫控制、组管理、无连接 TCS 的规定, 无论何时发出报文, 该报文都将含有必选信息元, 以及本节中定义的某些可选信息。

报文通常以单个 L2CAP 数据分组的形式进行传输。报文起始位置也就是 L2CAP 数据内容的起始位置。

定义内容包括:

- 报文流向和用途的简要描述;
- 报文中信息元顺序列表(所有报文类别具有一致顺序);
- 该表中各信息元的含义, 一般包括描述该信息元的规范的章节、是否标记必需信息 ‘M’ 或可选择信息 ‘O’、信息元长度, 这也可能因不同应用而定;
- 必要的注释信息。

所有报文都以 8 个字节为单位进行解释。

8.6.1 呼叫控制报文格式

1. ALERTING

该报文由呼入方发送，表示被叫用户报警已被初始化。报文格式如表 8.1 所示。

报文类别：ALERTING

流向：从呼入方到呼出方

表 8.1 ALERTING 报文内容

信息元	类别	长度
报文类别	M	1
信道容量	O*	4 (26)
进度指示	O	2
SCO 句柄	O	2
目的 CID	O	4
厂商指定信息	O	3

*：只允许在呼入方发出的第一个报文中使用

表 8.2 CALL PROCEEDING 报文内容

信息元	类别	长度
报文类别	M	1
信道容量	O*	4 (26)
进度指示	O	2
SCO 句柄	O	2
目的 CID	O	4
厂商指定信息	O	3-

*：只允许在呼入方发出的第一个报文中使用

2. CALL PROCEEDING

本报文（见表 8.2）由呼入方发送，表示请求呼叫建立过程已初始化，不再接收其他的呼叫建立信息。

报文类别：CALL PROCEEDING

流向：从呼入方到呼出方

3. CONNECT

本报文由呼入方发送，表示被叫用户已接收到呼叫。报文内容如表 8.3 所示。

报文类别：CONNECT

流向：从呼入方到呼出方

表 8.3 CONNECT 报文内容

信息元	类别	长度
报文类别	M	1
信道容量	O*	4 (26)
SCO 句柄	O	2
厂商指定信息	O	3-

*：只允许在呼入方发出的第一个报文中使用

表 8.4 CONNECT ACKNOWLEDGE 报文内容

信息元	类别	长度
报文类别	M	1
信道容量	O	4 (26)
SCO 句柄	O	4
厂商指定信息	O	3-

4. CONNECT ACKNOWLEDGE

本报文（见表 8.4）由呼出方发送，表示确认收到 CONNECT 报文。

报文类别：CONNECT ACKNOWLEDGE

流向：从呼出方到呼入方

5. DISCONNECT

本报文（见表 8.5）可由任意一方发送，请求终止呼叫。

报文类别：DISCONNECT

流向：双向

表 8.5 DISCONNECT 报文内容

信息元	类别	长度
报文类别	M	1
信道容量	O	2
进度指示	O	2
SCO 句柄	O	2
目的 CID	O	4
厂商指定信息	O	3-

表 8.6 INFORMATION 报文内容

信息元	类别	长度
报文类别	M	1
发送完成	O	1
键盘功能	O	2
被叫方号码	O	3-
语音控制	O	3-
厂商指定信息	O	3-

6. INFORMATION

本报文（见表 8.6）可由任一方发送，在重复呼叫情况下，用以在呼叫建立过程中提供附加信息。

报文类别：INFORMATION

流向：双向

7. PROGRESS

本报文（见表 8.7）由呼入方发送，表示工作情况下或在附加带内信息/模式时，任一方的呼叫进度。

报文类别：PROGRESS

流向：呼入方到呼出方

表 8.7 PROGRESS 报文内容

信息元	类别	长度
报文类别	M	1
进度指示	O	2
SCO 句柄	O	2
目的 CID	O	4
厂商指定信息	O	3-

表 8.8 RELEASE 报文内容

信息元	类别	长度
报文类别	M	1
原因	O	2
厂商指定信息	O	3-

*：必须在首次呼叫清除报文里。

8. RELEASE

本报文（见表 8.8）用于表示发送报文的设备已断开连接，并准备释放通道，而接收方也将在发送 RELEASE COMPLETE 后释放通道。

报文类别：RELEASE

流向：双向

9. RELEASE COMPLETE

本报文（见表 8.9）用于表示发送报文的设备已经释放通道，此通道可以被重用。

报文类别: RELEASE COMPLETE

流向: 双向

表 8.9 RELEASE COMPLETE 报文内容

信息元	类别	长度
报文类别	M	1
原因	O	2
厂商指定信息	O	3-

*: 在首次呼叫清除报文里必须使用。

表 8.10 SETUP 报文内容

信息元	类别	长度
报文类别	M	1
呼叫类	M	2
完成发送	O	1
信道容量	O	4(26)
信号	O	2
主叫方号码	O	3-
被叫方号码	O	3-
厂商指定信息	O	3-

10. SETUP

本报文（见表 8.10）由呼出方发送，以初始化呼叫建立。

报文类别: SETUP

流向: 呼出方到呼入方

11. SETUP ACKNOWLEDGE

本报文（见表 8.11）由呼入方发出，表示已初始化呼叫建立过程，但需要附加信息。

报文类别: SETUP ACKNOWLEDGE

流向: 呼入方到呼出方

表 8.11 SETUP ACKNOWLEDGE 报文内容

信息元	类别	长度
报文类别	M	1
信道容量	O	4(26)
进程指示	O	2
SCO 句柄	O	2
目的 CID	O	4
厂商指定信息	O	3-

*: 仅可用于呼入方发送的首个报文。

表 8.12 START DTMF 报文内容

信息元	类别	长度
报文类别	M	1
键盘功能	M	2

12. START DTMF

本报文（见表 8.12）包括其他方应转为远程用户使用的 DTMF 双音多频语音的数据位

报文类别: START DTMF

流向: 双向

13. START DTMF ACKNOWLEDGE

本报文（见表 8.13）表示 START DTMF 报文所需操作已成功初始化。

报文类别: START DTMF ACKNOWLEDGE

流向：双向

表 8.13 START DTMF ACKNOWLEDGE 报文内容

信息元	类别	长度
报文类别	M	1
键盘功能	M	2

表 8.14 START DTMF REJECT 报文内容

信息元	类别	长度
报文类别	M	1
原因	O	2

14. START DTMF REJECT

本报文（见表 8.14）表示不能接收 START DTMF 报文。

报文类别：START DTMF REJECT

流向：双向

15. STOP DTMF

本报文（见表 8.15）终止向远程用户发送 DTMF 语音信息。

报文类别：STOP DTMF

流向：双向

表 8.15 STOP DTMF 报文内容

信息元	类别	长度
报文类别	M	1

表 8.16 STOP DTMF ACKNOWLEDGE 报文内容

信息元	类别	长度
报文类别	M	1
键盘功能	M	2

16. STOP DTMF ACKNOWLEDGE

本报文（见表 8.16）表示终止向远程用户发送 DTMF 语音信息。

报文类别：STOP DTMF ACKNOWLEDGE

流向：双向

8.6.2 组管理报文格式

1. 请求访问权限

发送本报文（见表 8.17）表示初始化一方请求获取访问权限。

报文类别：ACCESS RIGHTS REQUEST

2. 接受访问权限

由应答方发送本报文（见表 8.18）表示访问权限授权。

报文类别：ACCESS RIGHTS ACCEPT

表 8.17 ACCESS RIGHTS REQUEST 报文内容

流向：双向信息元	类别	长度
报文类别	M	1
厂商指定信息	O	3-

图 8.18 ACCESS RIGHTS ACCEPT 报文内容

信息元	类别	长度
报文类别	M	1
厂商指定信息	O	3-

3. 拒绝访问权限

由应答方发送本报文（见表 8.19）表示拒绝访问权限。

报文类别：ACCESS RIGHTS REJECT

4. INFO SUGGEST

由 WUG 管理员发送本报文（见表 8.20）表示 WUG 配置已变化。

报文类别：INFO SUGGEST

流向：WUG 管理员到 WUG 成员

表 8.19 ACCESS RIGHTS REJECT 报文内容

信息元	类别	长度
报文类别	M	1
厂商指定信息	O	3-

表 8.20

信息元	类别	长度
报文类别	M	1
配置数据	M	8
厂商指定信息	O	3-

5. INFO ACCEPT

由 WUG 成员发送本报文（见表 8.21）表示已接收 WUG 配置更新。

报文类别：INFO ACCEPT

流向：WUG 成员到 WUG 管理员

表 8.21 INFO ACCEPT 报文内容

信息元	类别	长度
报文类别	M	1
厂商指定信息	O	3-

表 8.22 LISTEN REQUEST 报文内容

信息元	类别	长度
报文类别	M	1
被叫方号码	M	3-
厂商指定信息	O	3-

6. LISTEN REQUEST

由 WUG 成员发送本报文（见表 8.22），用于向 WUG 管理员请求对 WUG 成员进行成员间快速访问。

报文类别：LISTEN REQUEST

流向：WUG 成员到 WUG 成员

7. LISTEN SUGGEST

本报文（见表 8.23）由 WUG 管理员发送，表示存在允许 WUG 成员进行成员间快速访问的请求。

报文类别：LISTEN SUGGEST

流向：WUG 管理员到 WUG 成员

8. LISTEN ACCEPT

发送本报文（见表 8.24）表示接收先前的成员间快速访问请求。

报文类别：LISTEN ACCEPT

流向： 双向

表 8.23 LISTEN SUGGEST 报文内容

信息元	类别	长度
报文类别	M	1
厂商指定信息	O	3-

表 8.24 LISTEN ACCEPT 报文内容

信息元	类别	长度
报文类别	M	1
时隙	O	4
厂商指定信息	O	3-

9. LISTEN REJECT

发送本报文（见表 8.25）表示拒绝先前的成员间快速访问请求。

报文类别：LISTEN REJECT

流向： 双向

表 8.25 LISTEN REJECT 报文内容

信息元	类别	长度
报文类别	M	1
原因	O	2
厂商指定信息	O	3-

表 8.26 CL INFO 报文内容

信息元	类别	长度
报文类别	M	1
语音控制		
厂商指定信息		

8.6.3 CL INFO

由任何一方发送此报文（见表 8.26），表示将按照无连接模式提供附加信息。

报文类别：CL INFO

流向： 双向

8.7 报文编码

报文编码规则遵循 ITU-T 的 Q.931 建议，但按照 TCS 需要进行调整和修改。每一个报文都由协议标识、报文类别、其他所需信息元等三部分组成。其中协议标识和报文类别是每一个 TCS 报文的基本组成部分，其他信息元则根据报文类别而定。一个 TCS 报文的通用格式如表 8.27 所示。一个信息元只能在给定报文中出现一次。缺省值只有在不使用指定值或没有备选值协商机制时使用。

当一个域分布在几个字节上，每一位取值的顺序将递减，而字节号递增。该域的最小位由最高位字节的最低位比特表示。总之，每个字节的第一位包括该段的最低位。

表 8.27 通用报文格式

8	7	6	5	4	3	2	1
协议标识				报文类别			
其他信息元							

8.7.1 协议标识和报文类别

协议标识的作用是将报文划分为不同的功能组。报文第一部分的前 3 位就是协议标识。

协议标识按照表 8.28 方式编码。

表 8.28 协议标识

位			功 能
8	7	6	
0	0	0	蓝牙 TCS 呼叫控制
0	0	1	蓝牙 TCS 组管理
0	1	0	蓝牙 TCS 无连接
保留其他所有值			

报文类别用于标识发出报文的的功能。报文第一部分的 5 位就是报文类别标识。报文类别按照表 8.29 方式编码。

表 8.29 报文类别

位					功 能
5	4	3	2	1	
					呼叫控制报文
					建立呼叫
0	0	0	0	0	ALERTING
0	0	0	0	1	CALL PROCEEDING
0	0	0	1	0	CONNECT
0	0	0	1	1	CONNECT ACKNOWLEDGE
0	0	1	0	0	PROGRESS
0	0	1	0	1	SETUP
0	0	1	1	0	SETUP ACKNOWLEDGE
					呼叫清除
0	0	1	1	1	DISCONNECT
0	1	0	0	0	RELEASE
0	1	0	0	1	RELEASE COMPLETE
					混合
0	1	0	1	0	INFORMATION
1	0	0	0	0	START DTMF
1	0	0	0	1	START DTMF ACKNOWLEDGE
1	0	0	1	0	START DTMF REJECT
1	0	0	1	1	STOP DTMF
1	0	1	0	0	STOP DTMF ACKNOWLEDGE
					组管理报文
0	0	0	0	0	INFO SUGGEST
0	0	0	0	1	INFO ACCEPT
0	0	0	1	0	LISTEN REQUEST
0	0	0	1	1	LISTEN ACCEPT
0	0	1	0	0	LISTEN SUGGEST
0	0	1	0	1	LISTEN REJECT

续表

位					功 能
5	4	3	2	1	
0	0	1	1	0	ACCESS RIGHTS REQUEST
0	0	1	1	1	ACCESS RIGHTS ACCEPT
0	1	0	0	0	ACCESS RIGHTS REJECT
					无连接报文
0	0	0	0	0	CL INFO

8.7.2 其他信息元

1. 编码规则

信息元分为三大类：

- 1) 单字节信息元（见图 8.13）；
- 2) 双字节信息元（见图 8.14）；
- 3) 不定长位组信息元（见图 8.15）。

表 8.30 将本规范中用于信息元的信息元编码规则进行汇总。

8	7	6	5	4	3	2	1	字节
1	信息元标识							1

图 8.13 单字节信息元格式

8	7	6	5	4	3	2	1	字节
1	信息元标识							1
信息元内容							2	

图 8.14 单字节信息元格式

8	7	6	5	4	3	2	1	字节
1	信息元标识							1
信息元内容长度							2	
信息元内容							3	

图 8.15 双字节信息元格式

下面的信息元描述将按字符顺序排列，但报文中信息元的实际排列顺序与此不同。对于不定长信息元格式的信息元标识编码值将根据报文信息元的实际顺序按升序排列。信息接收设备可以直接检测出某一信息位是否缺少，而不用对整个报文进行扫描。

本规范中的信息元可以包含被赋值为‘0’的空位。为了适应将来的变化，报文不能由于某空位值置为‘1’而被拒绝。

不定长信息元的第二个字节标识信息元内容总长，与第一个字节的编码无关。也就是说，长度从第三个字节开始计算。长度值实际就是内容字节数的二进制编码。

报文可以包括一个可选的变长信息元，但它为空（长度为 0）。如果不包括该信息元，接受方也应能解释。简单地说，接受方应能对为空的信息元进行解释。

表 8.30 信息元标识编码

编 码									参考	最大长度 (字节)
8	7	6	5	4	3	2	1			
1								单字节信息元	15	1
	0	1	0	0	0	0	1	发送完成		
1								双字节信息元		
	1	0	0	0	0	0	0	呼叫类	4	2
	1	0	0	0	0	0	1	原因	7	2
	1	0	0	0	0	1	0	速度指示	13	2
	1	0	0	0	0	1	1	信号	16	2
	1	0	0	0	1	0	0	键盘功能	12	2
	1	0	0	0	1	0	1	SCO 句柄	14	2
0								变长信息元		
	0	0	0	0	0	0	0	时隙	8	4
	0	0	0	0	0	0	1	配置数据	2	*
	0	0	0	0	0	1	0	信道容量	3	4(26)
	0	0	0	0	0	1	1	目的 CID	11	4
	0	0	0	0	1	0	0	主叫号码	6	*
	0	0	0	0	1	0	1	被叫号码	5	*
	0	0	0	0	1	1	0	语音控制	2	*
	0	0	0	0	1	1	1	厂商指定信息	9	*

2. 语音控制

语音控制信息元用于表示与语音控制相关的信息，其格式和编码分别如图 8.16 和表 8.31 所示。

8	7	6	5	4	3	2	1	字节
0	0	0	0	0	0	0	0	1
信息元内容长度								2
控制信息								3

图 8.16 语音控制信息元格式

表 8.31 语音控制信息元编码

控制信息(第 3 字节)								
比特								
	7	6	5	4	3	2	1	
	0	0	0	0	0	0	0	音量增大
	0	0	0	0	0	0	1	音量减小
	0	0	0	0	0	1	0	麦克风音量增大
	0	0	0	0	0	1	1	麦克风音量减小
	0	X	X	X	X	X	X	蓝牙标准保留使用
	1	X	X	X	X	X	X	厂商指定信息

3. 信道容量

信道容量信息元用于指示一种被请求的或可用的服务，其格式和编码分别如图 8.17 和表 8.32 所示。

如果没有该信息元，缺省的信道容量为带有 HV3 类型包的同步面向连接链接，该链接采用用户信息层 CVSD 编码方式，如图 8.18 和图 8.19 所示。

8	7	6	5	4	3	2	1	字节
0	0	0	0	0	0	1	0	1
信息元内容长度								2
链接类型								3

图 8.17 信道容量信息元格式

用户信息层 1	数据分组类型	第 4 字节
---------	--------	--------

图 8.18 链接类型元编码=00000000（SCO）

标 志		4
服务类别		5
令牌率		6
		7
		8
		9
令牌尺寸(字节)		10
		11
		12
		13
带宽值(字节/秒)		14
		15
		16
		17
应答时间(微秒)		18
		19
		20
		21
延迟(微秒)		22
		23
		24
		25
用户信息层 3	用户信息层 2	26

注：由于只有 TCS 具有端到端服务质量条件，服务质量在 TCS 层次上重复。

图 8.19 链接类型元编码=00000001（ACL）

表 8.32 信道容量信息元编码

链接类型(第 3 字节)								
比特								
8	7	6	5	4	3	2	1	
0	0	0	0	0	0	0	0	同步面向连接
0	0	0	0	0	0	0	1	异步无连接
0	0	0	0	0	0	01	0	无
保留所有值								
字节 4 编码(链接类型信息元编码=000000000)								
数据分组类型(字节 4)								
	5	4	3	2	1			
	0	0	1	0	1	HV1		
	0	0	1	1	0	HV2		
	0	0	1	1	1	HV3		
	0	1	0	0	0	DV		
保留所有值								
用户信息层 1(字节 4)								
比特								
	8	7	6					
	0	0	1					
	0	1	0					
	0	1	1					
保留所有值								
字节 4-26 编码(链接类型信息元编码=000000001)								
字节 4-25 编码细节可在 L2CAP 中找到								
用户信息层 2(字节 26)								
比特								
4	3	2	1					
0	0	0	0	RFCOMM over L2CAP				
保留所有值								
用户信息层 3(字节 26)								
比特								
8	7	6	5					

续表

0	0	0	0	未定义
0	0	0	1	PPP
0	0	1	0	IP
保留所有值				
字节 4 编码(链接类型信息元编码=000000010)				
缺少字节 4				

4. 呼叫类别

呼叫类别用于表示被请求服务的基本情况。该信息元允许用户指出缺省属性的用途，并缩短 setup 报文的长度。其格式和编码分别如图 8.20 和表 8.33 所示。

8	7	6	5	4	3	2	1	字节
1	1	0	0	0	0	1	0	1
呼叫类别								2

图 8.20 呼叫类别信息元格式

表 8.33 呼叫类别信息元编码

呼叫类别(字节 2)								
比特								
8	7	6	5	4	3	2	1	
0	0	0	0	0	0	0	0	外部呼叫(①)
0	0	0	0	0	0	0	1	内部呼叫(②)
0	0	0	0	0	0	1	0	服务呼叫(③)
0	0	0	0	0	0	1	1	紧急呼叫(④)
保留所有值								

注:

- ①外部呼叫是指一个对外部网络（如 PSTN）的呼叫或来自于外部网络的呼叫；
- ②内部呼叫是指蓝牙设备之间的呼叫；
- ③服务呼叫是指出于配置目的的呼叫；
- ④紧急呼叫是指一个利用紧急呼叫号码和某些特性的外部呼叫。

5. 被叫号码

被叫号码信息元用于惟一标识呼叫的被叫方，其格式和编码分别如图 8.21、表 8.34 所示。

8	7	6	5	4	3	2	1	字节
0	0	0	0	0	1	0	1	1
信息元内容长度								2
0	号码类别			编号计划标识				3
0	号码位(IA5 字符)*							4

※注：多个第 4 位号码数字输入顺序一致，也就是说，第一个输入的数字位于第一个字节 4。

图 8.21 被叫方信息元格式

表 8.34 被叫方信息元编码

号码类别(字节 3)					
	比特				
	7	6	5		
	0	0	0	未知	
	0	0	1	国际号码	
	0	1	0	国内号码	
	0	1	1	网络指定号码	
	1	0	0	用户号码	
	1	1	0	缩写号码	
	1	1	1	保留	
	保留所有值				
	比特				
	4	3	2	1	
	0	0	0	0	未知
	0	0	0	1	ISDN/电话编号方案 E.164
	0	0	1	1	数据编号方案 REC.X.121
	0	1	0	0	保留
	1	0	0	0	国内标准编号方案
	1	0	0	1	私用编号方案
	保留所有值				

6. 主叫号码

主叫号码信息元用于标识呼叫的来源，其格式和编码分别如图 8.22、表 8.35 所示。

8	7	6	5	4	3	2	1	字节
0	0	0	0	0	1	0	0	1
信息元内容长度(字节)								2
0	号码类别				编号方案标识			3
0	标识		0 0 0			显示标识		4
	数据位(IA5 字符)							5 等

图 8.22 主叫方信息元格式

表 8.35 主叫方信息元编码

号码类别(字节 3)				
比特				
7	6	5		
0	0	0	未知	
0	0	1	国际号码	
0	1	0	国内号码	
0	1	1	网络指定号码	
1	0	0	用户号码	
1	1	0	缩写号码	
1	1	1	保留	
保留所有值				
编号方案标识(字节 3)				
比特				
4	3	2	1	
0	0	0	0	未知
0	0	0	1	ISDN/电话编号方案 E.164
0	0	1	1	数据编号方案 REC.X.121
0	1	0	0	保留
1	0	0	0	国内标准编号方案
1	0	0	1	私用编号方案
保留所有值				
标识(字节 4)				
比特				
7	6			
0	0	允许的标识		
0	1	限制的标识		
1	0	不能用于互操作的号码		
1	1	保留		
显示标识				
比特				
2	1			
0	0	用户提供, 不显示		
0	1	用户提供, 校验通过		
1	0	用户提供, 校验失败		
1	1	网络提供		
保留所有值				

7. 出错原因

出错原因用于指示引起被请求服务失败的远端原因,其格式和编码分别如图 8.23、表 8.36 所示。

8	7	6	5	4	3	2	1	字节
1	1	0	0	0	0	0	1	1
出错原因值								2

图 8.23 出错原因信息元格式

表 8.36 出错原因信息元编码

出错原因(字节 2)								
	比特							
8	7	6	5	4	3	2	1	
0	这 7 位采用类似于 ITU-T 建议 Q.850 中的原因值子域编码方法进行编码							

8. 时隙

时隙信息元用于表示使用的蓝牙时隙，其格式和编码分别如图 8.24、表 8.37 和图 8.25、表 8.38 所示。其中厂商指定信息元用于发送非标准信息。

8	7	6	5	4	3	2	1	字节
0	0	0	0	0	0	0	0	1
信息元内容长度								2
时 隙								3
								4

图 8.24 时隙信息元格式

表 8.37 厂商指定信息

号码类别(字节 3)																
	比特															
	字节 3								字节 4							
8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	
	0	0	0	包括蓝牙时钟的 2~16 位												

8	7	6	5	4	3	2	1	字节
0	0	0	0	0	1	1	1	1
信息元内容长度								2
厂商指定标识								3
厂商指定标识								4
厂商指定内容								
								L+2

图 8.25 厂商指定信息元格式

表 8.38 厂商指定信息元编码

厂商标识编码(字节 3 和字节 4)																
	比特								比特							
	字节 3								字节 4							

续表

I 商标识编码(字节 3 和字节 4)															
8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
保留所有值															

9. 配置数据

配置数据信息元用于表示配置数据，其格式和编码如图 8.26 所示。

8	7	6	5	4	3	2	1	字节
0	0	0	0	0	0	0	1	1
信息元内容长度								2
0	WUG 成员 1 的内部号码(IA5 字符)							3
0	WUG 成员 1 的内部号码(IA5 字符)							4
WUG 成员 1 的蓝牙地址								5
								...
								10
用于 WUG 成员 1 的链接关键字								11
								...
								26
.....								
0	WUG 成员 n 的内部号码(IA5 字符)							$3+((n-1)*24)$
0	WUG 成员 n 的内部号码(IA5 字符)							$4+((n-1)*24)$
WUG 成员 n 的蓝牙地址								$5+((n-1)*24)$
								...
								$10+((n-1)*24)$
用于 WUG 成员 n 的链接关键字								$11+((n-1)*24)$
								...
								$26+(n-1)*24)$

图 8.26 配置数据信息元格式与编码

注：内部号码(两位)出现在第 3 字节和第 4 字节，其顺序与输入顺序相同。也就是说，第一个输入的数据位位于字节 3。对于所有的 n，WUG 成员重复字节 3~26。

10. 目的 CID

目的 CID 信息元用于使远端能够将已建立的 L2CAP 通道与正在进行的呼叫相关联。目的 CID 唯一对应于以配置请求包形式交换目的 CID(参见 L2CAP)。其格式和编码如图 8.27 所示。

8	7	6	5	4	3	2	1	字节
0	0	0	0	0	0	1	1	1
信息元内容长度(字节)								2
DCID 字节1								3
DCID 字节0								4

图 8.27 目的 CID 信息元格式

11. 键盘设置

键盘功能信息元的目的在于传输 IA5 字符，如由终端键盘输入的字符，其格式和编码如图 8.28 所示。

8	7	6	5	4	3	2	1	字节
1	1	0	0	0	1	0	0	1
0	键盘功能信息(IA5 字符)							2

图 8.28 键盘功能信息元格式

12. 进度指示

进度指示信息元用于描述在呼叫生命周期中发生的事件，其格式和编码分别如图 8.29、表 8.39 所示。

8	7	6	5	4	3	2	1	字节
1	1	0	0	0	0	1	0	1
0	进度指示							2

图 8.29 进度指示信息元格式

表 8.39 进度指示信息元编码

进度信息(字节 2)							
比特							
7	6	5	4	3	2	1	
0	0	0	1	0	0	0	带内信息或合适模式现在可用
保留所有值							

13. SCO 句柄

SCO 句柄信息元用于使远端能够将已建立的 SCO 链接与正在进行的呼叫相关联。SCO 句柄惟一标识一个由网管发出的以 LMP_SCO_link_req 形式交换的 SCO 句柄，其格式如图 8.30 所示。

8	7	6	5	4	3	2	1	字节
1	1	0	0	0	1	0	1	1
0	SCO 句柄值							2

图 8.30 SCO 句柄信息元格式

14. 发送完成

发送完成信息元用于表示被叫方号码结束，其格式和编码如图 8.31 所示。

8	7	6	5	4	3	2	1	字节
1	0	1	0	0	0	0	1	1

图 8.31 发送完成信息元格式和编码

15. 信号

信号信息元用于向与语音和报警信号有关的用户发送信息，其格式和编码分别如图 8.32、表 8.40 所示。

8	7	6	5	4	3	2	1	字节
1	1	0	0	0	0	1	1	1
0	信号值							2

图 8.32 信号信息元格式

表 8.40 信号信息元编码

信号值(字节 2)								
比特								
8	7	6	5	4	3	2	1	
0	1	0	0	0	0	0	0	外部呼叫
0	1	0	0	0	0	0	1	内部呼叫
0	1	0	0	0	0	1	0	回呼
0	X	X	X	X	X	X	X	蓝牙标准保留
1	X	X	X	X	X	X	X	厂商指定

8.8 报文出错处理

1. 协议标识出错

当收到的报文除了包括 7.2 节中定义的信息元以外，还有已编码的协议标识，那么，该报文应被忽略。

2. 报文太短或未被识别

如果收到的报文太短，不足以包含完整的报文类别信息元，则忽略该报文；
如果收到的报文包含完整的报文类别信息元，但不能被识别，则忽略该报文。

3. 报文类别或报文顺序出错

除了在 NULL 状态下以外，无论何时收到除了 RELEASE 或 RELEASE COMPLETE 以外的报文，则应忽略该报文。

当收到一个未知的 RELEASE 报文，接收方应断开和释放已建立的信道，并返回 RELEASE COMPLETE 报文，终止所有定时器，进入 NULL 状态；

当收到一个未知的 RELEASE COMPLETE 报文，接收方应断开和释放已建立的信道，

并返回 RELEASE COMPLETE 报文，终止所有定时器，进入 NULL 状态。

4. 信息元出错

报文中的信息元应按第 6 节中所示的顺序进行排列。

当收到的报文缺少必需的信息元，或必需的信息元包括不合法内容，则忽略该报文。

为避免在 SETUP 报文中必需信息元出错，应返回 RELEASE COMPLETE 报文和#96 出错信息，‘缺少必需信息元’或#100 出错信息，‘信息元内容非法’。

当收到的报文包含未识别信息元，或包含含有非法内容的可选信息元，或包含未定义的可识别信息元，接收方应忽略该信息元。

信息元超长则视为该信息元含有非法内容。

8.9 协议参数

协议参数如表 8.41 所示。

表 8.41 定时器值

定时器名	值
T301	至少 3 分钟
T302	15 秒
T303	20 秒
T304	30 秒
T305	30 秒
T308	4 秒
T310	30~120 秒
T313	4 秒
T401	8 秒
T402	8 秒
T403	4 秒
T404	2.5 秒
T405	2 秒
T406	20 秒

第9章 WAP 信道的蓝牙互操作性要求

蓝牙在 WAP 客户和服务器之间提供通信物理介质和链路控制。本章对基于 PPP 的通信方式进行描述，主要包括：

- 概述蓝牙环境下对 WAP 的使用，解释符合蓝牙版本的增值服务概念，并提供利用 WAP 增值服务如何适应蓝牙应用模型的实例；
- WAP 服务概述的目的在于将 WAP 环境置于相同的环境当中，并将介绍 WAP 组件及其与同层次 INTERNET 协议的比较；
- 对蓝牙匹克网中的 WAP 进行论述，并描述蓝牙通信结构关联 WAP 服务的方式；
- 最后，阐述为在两个支持蓝牙技术的 WAP 设备间实现互操作而需提供的蓝牙特性。

9.1 蓝牙环境中的 WAP 应用

9.1.1 增值服务

一个设备的通信能力并不仅仅限于其自身。终端用户通常并不关心技术，而只是对技术能够帮助他们做什么感兴趣。

传统的电信技术只是依靠语音通信作为惟一的技术应用，而且该应用方式在市场中取得了成功。而当数据通信服务变得越来越广泛地得到应用时，如何更好更多地利用数据服务则变得越来越重要了。

无线应用协议论坛的目标就是创建一个标准框架，在该框架中对能够提供的增值服务和互操作层次做出定义。

9.1.2 应用实例

蓝牙用于增值服务的惟一优点是能够在有限范围内建立通信链路。理想情况下，支持蓝牙的设备能够满足与位置无关的服务需求。下面就是一些 WAP 客户/服务器模型应用于蓝牙用途的实例。

1. 短信息

短信息服务允许用户的笔记本电脑或移动电话进行无用户干扰的通信，主要用于更新用户电子邮件。用户可从手机中浏览收到的消息，而不再需要将笔记本电脑从皮包里取出来。

2. 强制信息

强制信息与短信息相似。用户可以将消息组织在一个没有拨号连接的环境里。过一段时间，等笔记本电脑唤醒以后，就检查移动电话，看是否可以发送刚才未发送的信息。如果存在可用通信链路，就可发出电子邮件。

3. WAP 网站信息

WAP 网站信息使用户能够连接到移动 PC 或手持设备，以与公共 WAP 网站相连。网站

可以根据该设备所处位置向该设备提供信息。例如，航班信息、机场登机口、购物中心的商店位置、火车时刻表或者铁路目的地。

9.2 WAP 服务概述

无线应用协议用于向那些在多方面受到限制的设备提供 Internet 或类似于 Internet 的访问。有限的通信带宽、内存、不间断电源、显示能力和输入设备都是促进 WAP 发展的因素。尽管某些设备具有上述限制，但 WAP 仍然为这些设备提供了大量可用之处。

典型的 WAP 环境由三类设备组成：WAP 客户端设备、WAP 代理/网关和 WAP 服务器。在某些情况下，代理/网关也包括服务器的一些功能。

9.2.1 WAP 实体

1. WAP 客户端

WAP 客户端设备通常都是用户手持设备。该设备可以是功能强大的便携计算机，也可以是移动电话。客户端的必要特性是必须要有显示和输入设备。

WAP 客户端通过无线网络与 WAP 代理/网关相连。该网络可能基于某些可用的技术。WAP 协议允许网络具有低可靠性和高延迟，但不能中断服务。

2. WAP 代理/网关

WAP 代理/网关的作用在于提供无线网络和 Internet 之间的接口。代理的主要功能在于能够向 WAP 客户端设备提供 DNS 域名解析服务，以及将 Internet 协议和内容格式翻译到 WAP 的相应层次。

3. WAP 服务器

WAP 服务器的功能类似于 Internet 上的服务器。实际上，WAP 服务器通常就是一个 HTTP 服务器。该服务器作为用户经常访问信息存放处，其内容包括文本、图像、甚至是允许客户端设备执行的服务器端代码。

WAP 服务器日志与代理/网关放在同一个物理设备上，或者在代理/网关能够访问到的网络中任何地方。

这样，一个服务器可以作为一个 HTTP 服务器，一个 WAP 服务器，或者两者兼而有之。

9.2.2 WAP 协议

WAP 环境包括一个分层的协议栈，它将用户代理与通信网络细节分开。图 9.1 表示 WAP 协议栈的综合结构。

1. 无线数据分组协议 (WDP)

WDP 层提供一个服务接口，该接口作为基于套接字 UDP 的执行版本。对于一个基于 IP 的信道服务，该层是 UDP。对于不能提供 UDP 服务接口的信道，应提供 WDP 应用作为适配层，以允许在本信道进行基于套接字的数据分组传输。

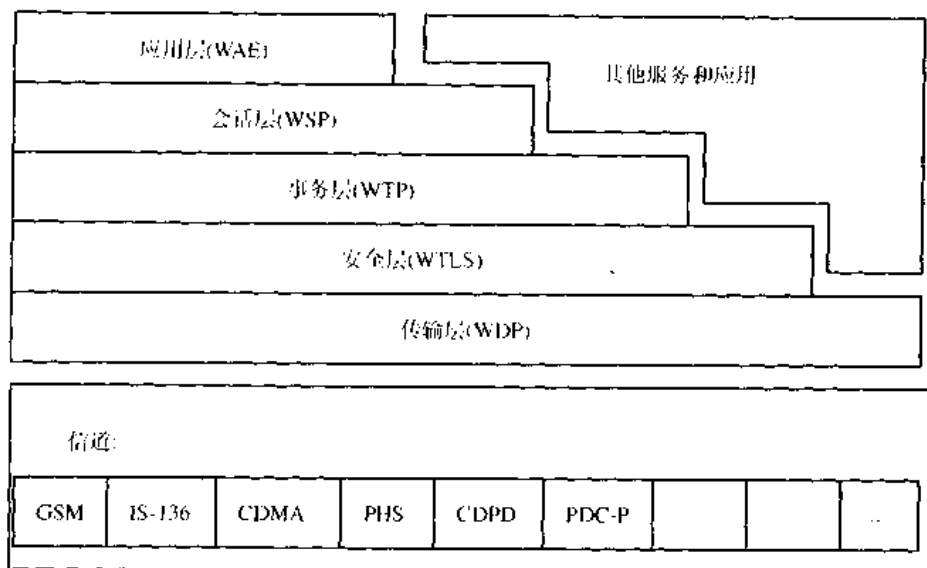


图 9.1 WAP 协议栈

2. 无线传输协议 (WTP)

WTP 层在 WDP (UDP) 协议层之上提供可靠的数据分组服务。

3. 无线传输层安全性 (WTLS)

WTLS 层对于能够在客户端 WSP 会话与其对应服务器 WSP 会话之间提供安全数据管道的协议栈而言,是一个可选的组件。在 WAP 规范的当前版本中,会话终止于服务器端。现在,在 WAP 论坛之前就有一个关于代理协议的提议。该提议将允许中间 WAP 代理能够在不对数据流解密的情况下,通过代理/网关传输 WTLS 数据分组。

4. 无线会话协议 (WSP)

WSP 层在客户端应用与 WAP 服务器之间建立关联。这样就可以长时间保持会话,而且在服务中断时仍能够保持。WSP 利用 WTP 服务提供到目的代理/网关的可靠传输。

9.2.3 WAP 和 INTERNET 间的协议转换

WAP 协议栈要解决的问题和应用都与 IETF 组织所定义的协议和应用相关。WAP 论坛的基本目标就是对那些因不能应用 INTERNET 协议而受到限制的设备提供 INTERNET 内容或服务。

本节分别就 WAP 协议栈各层与 IETF 中的对应协议层次进行比较。

1. UDP/WDP

在大多数的基本协议层次上, WAP 和 INTERNET 相同。与 INTERNET 中传输层相同, WAP 协议栈也利用基于套接字的数据分组模型。

一些 INTERNET 协议也采用 UDP 服务,但大多数协议都是采用面向连接报文流协议 (TCP)。

2. WTP/TCP

无线传输协议 (WTP) 在某些方面提供与 TCP 要求相同的服务。INTERNET 传输协议 (TCP) 提供基于 IP 服务的可靠、面向连接和字符流的协议。比较而言, WTP 能够提供单

向的可靠或非可靠的报文传输，也能够提供双向可靠的报文传输。这种经过优化的传输方式适于 WAP 的“短请求、长回应”的会话特点。WTP 提供报文拼接，以减少报文传输数量。

3. WTLS/SSL

无线传输层安全性（WTLS）是由安全套接字协议（SSL）发展而来的。因此，它也提供与 SSL 相同的鉴权和加密服务。

4. WSP/HTTP

WAP 中的会话服务由无线会话协议（WSP）提供。该协议与 HTTP 1.1 的术语和功能定义保持一致，但增加了对长时间会话、‘推’数据和会话挂起、会话继续的支持。另外，协议利用压缩编码方法以适应窄带通信要求。

5. WML/HTML

WAP 使用的标记语言是与 HTML 相同的经过精简的版本，但经过针对手持设备应用的优化。WML 成为一种符合 XML 定义的标记语言。

6. WML Script/JavaScript

WAP 具有与 JavaScript 相同的编码语言，但该语言根据 WAP 目标设备类型进行了修正。

9.3 WAP 在蓝牙匹克网中的应用

蓝牙技术在很多方面都可以应用在其他类似于 WAP 的无线网络中。蓝牙技术能够用于在 WAP 客户端与其邻近 WAP 服务器之间提供数据传输的信道。此外，蓝牙本质上可以提供只能由 WAP 协议统一浏览的能力。

9.3.1 WAP 服务器通信

WAP 通信的传统组成包括一个能够利用 WAP 协议与服务器/代理设备通信的客户端设备。在这种情况下，蓝牙中介可以提供由 WAP 体系结构规定的信道服务。

1. 客户端设备初始化

当一个 WAP 设备处于对蓝牙设备的有效侦听状态时，它就能利用蓝牙服务搜索协议搜索 WAP 服务器。

在图 9.2 中，WAP 客户端设备首先进入 WAP 代理/网关匹克网范围。当客户端检测到 WAP 代理/网关的存在时，它就能自动地或在客户端请求下与服务器建立连接。

客户端应能够确定已检测到 WAP 代理/网关的特性。蓝牙服务搜索协议应能够获取服务器的下列信息：

- 服务器名——应为用户能够理解的描述性名字；
- 服务器主页文本名字——即服务器的主页链接。本信息可选。
- 服务器/代理功能——表示该设备是否是一个 WAP 内容服务器，还是一个代理或两者都是。如果是一个代理，它必须能够解析不是服务器/代理设备地址的 URL。

在图 9.2 第二步中，该设备正在与 WAP 代理/网关通信，可以正常使用所有的 WAP 数据服务。

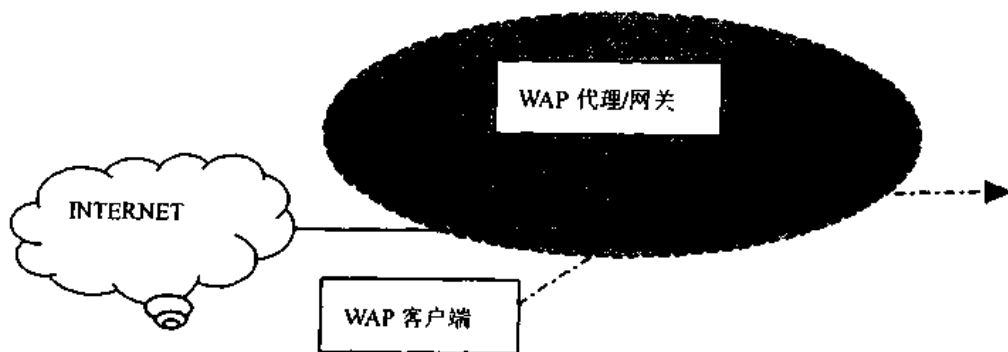


图 9.2 匹克网中的 WAP 代理/网关

2. 客户端设备终止

在图 9.2 第三步中, 该设备退出匹克网。当该设备检测到与 WAP 代理/网关的通信链接已经丢失, 它就会决定是否利用搜索到的信息决定是否继续保持通信连接。

例如, 支持多个信道的客户端设备, 在给出了服务器功能信息后, 将查询服务器的其他地址信息。由于客户端设备随时会离开匹克网而导致搜索的信息不再有用, 应将这些信息缓存起来以便以后的访问。

在前面的 WAP 网站例子里, 如果用户想要在超出蓝牙范围时仍能够接收信息, 该网站应向客户端设备提供 INTERNET 地址。当蓝牙通信连接失效时, 该设备就可以利用单元数据分组继续客户/服务器会话。

该功能根据实际情况不同而不同。

3. 服务器设备初始化

在客户与服务器之间初始化通信的另外一个方法就是服务器周期性轮询有无可用的客户端设备。当服务器设备发现一个客户具有 WAP 客户功能的时候, 服务器可以连接该设备并向它‘推’送数据。

客户端设备则由终端用户决定选择是否忽略被‘推’过来的数据。

通过蓝牙服务搜索协议, 该服务器能够确定以下有关客户端的信息:

- 客户名——经过格式化的描述客户端设备的描述性名字;
- 客户端特性——服务器可以通过该信息确定有关客户端蓝牙特性的信息。

9.3.2 蓝牙环境下的 WAP 应用

为了有效支持 WAP OVER BLUETOOTH, 应充分考虑设备特性。

1. WDP 管理实体

在 WAP 协议栈中, 与 WDP 协议层关联的实体负责管理由该层提供的服务。WDP 管理实体 (WDP-ME) 作为控制协议栈的扩展机制, 包括如下内容。

a. 异步通知

当某种事件发生时, WDP-ME 需要能够生成发往应用层的异步通知。异步通知的例子有:

- 侦测到新的客户端结点;
- 侦测到新的服务器端结点;

- 丢失客户端结点信号；
- 丢失服务器端结点信号；
- 侦测到服务器“推”操作，而并未请求该内容。

对于这些事件的支持因实际情况而异。所有列出的服务器“推”操作异常事件都从蓝牙主机控制器接口中引申而来。

b. 备选信道

某个设备的 WAP 应用可以支持多个信道。有关信道选择方法的讨论超出了本文本范围。接下来的过程根据实际应用情况的不同而不同。

2. 寻址

在 WAP 环境中使用两种基本的寻址方式：用户寻址和代理/网关寻址。用户寻址描述对象在网络中的位置，它与所在信道无关。代理/网关寻址描述与之通信的 WAP 代理/网关的位置，它取决于信道类型。

用户主要处理统一资源定位（URL）。这些地址是用于描述被访问文档的文本串。如代理/网关就与 INTERNET 域名紧密相关。

服务器将这些地址解析成为网络地址。

WAP 代理/网关地址通常是一个由用户或网络管理员配置的静态值。当用户输入 URL，也就是向配置好的 WAP 代理/网关发送请求时，如果 URL 在 WAP 代理/网关域之外，WAP 代理/网关就会利用 DNS 解析确定文本所在的服务器的 IP 地址。

客户端设备首先标识一个可以由蓝牙访问的代理/网关，然后就利用服务搜索协议向用户提供服务器名或对服务器的描述。当用户选择了一个服务器，客户端就会下载服务器主页。一旦用户浏览该服务器主页，那么所有的下级 URL 就都会与该主页相关联。该情况假设 WAP 代理/网关和 WAP 内容服务器都位于蓝牙设备上，尽管互操作性并不要求具有这种结构。

WAP 代理/网关/服务器提供含有服务器主页内容的缺省 URL。一个只作为代理的设备并不提供 URL 或相关内容。

9.3.3 对 WAP 的网络支持

下面对协议栈作出规定。该协议栈可以在 WAP 组件下进行利用，可以支持其他的协议栈，并且应通过蓝牙服务搜索协议体现出来。

作为利用 PPP 的 WAP 服务所在的信道，支持蓝牙的设备提供对下列协议栈的支持，如图 9.3 所示。

为实现互操作性，本文本假设 WAP 客户端扮演基于 PPP 的局域网访问规范所定义数据终端的角色。此外，还假设 WAP 服务器或代理设备作为局域网访问点。

基带、LMP 和 L2CAP 是 OSI 的第一层和蓝牙协议的第二层。RFCOMM 是 GSM TS07.10 的蓝牙版本。SDP 是蓝牙服务搜索协议。

PPP 即 IETF 点对点协议。WAP 是无线应用协议栈和应用环境。

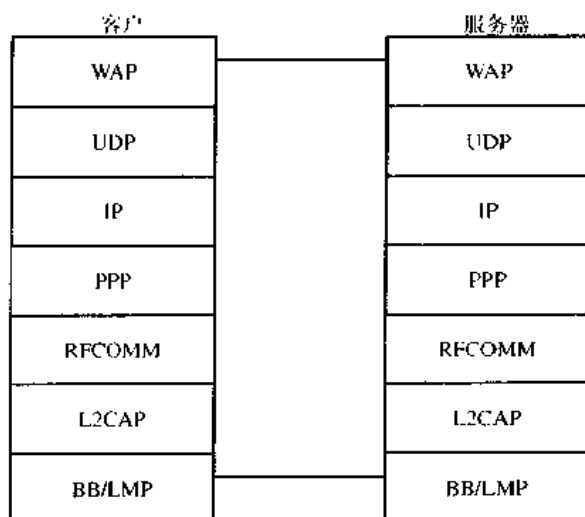


图 9.3 对 WAP 协议栈的支持

9.4 互操作性要求

对 WAP Over Bluetooth 的第一步互操作性（必须）包括：

- 提供对 A 级 WAP 设备的兼容；
- 通过服务搜索机制，为支持 WAP 代理/网关功能的设备提供网络地址。

对 WAP Over Bluetooth 的第二步互操作性（必须）包括：

- 支持所有第一步的互操作性要求；
- 通过服务搜索，提供有关服务器/代理性能的信息和服务器名；
- 通过服务搜索，提供有关客户特性信息和客户名；
- 对服务器的异步通知；
- 对客户的异步通知。

9.5 服务搜索

9.5.1 SDP 服务记录

服务记录是 WAP 客户端设备和代理/网关动态查找对方的一种机制。这种用途与其他在两设备间短暂的 WAP 信道不同。也就是说，蓝牙设备不会有一个配置或存储指定代理/网关的具体信道地址。

客户端和代理/网关会在它们互相接近时侦测到对方。蓝牙服务搜索协议使设备能够互相查询对方特性。

表 9.1 列出了 WAP 代理/网关设备的服务记录；表 9.2 和表 9.3 列出了 WAP 互操作性要求的 SDP 协议数据单元。

表 9.1 WAP 代理/网关设备的服务记录格式

项目	定义	类型	值	属性 ID	是否必须
ServiceClassIDList				0x0001	M

续表

项目	定义	类型	值	属性 ID	是否必须
ServiceClass0	WAP 代理/网关	UUID	WAP		M
BluetoothProfile DescriptorList					M
ProfileDescriptor0				0x0009	M
Profile	支持的标准	UUID	LANAccess UsingPPP[4]		M
Version	标准版本	UInt16			M
Protocol DescriptorList					O
Descriptor0	UDP	UUID	UDP		O
Parameter0	WSP 无连接 会话端口号	UInt16	9200(缺省值)		O
Parameter1	WTP 会话端口号	UInt16	9201		O
Parameter2	WSP 安全 无连接端口号	UInt16	9202		O
Parameter3	WTP 安全 会话端口号	UInt16	9203		O
Parameter4	WAP vCard 端口号	UInt16	9204		O
Parameter5	WAP vCard 端口号	UInt16	9205		O
Parameter6	WAP vCard 安全端口号	UInt16	9206		O
Parameter7	WAP vCard 安全端口号	UInt16	9207		O
ServiceName	可显示的文本名字	String	变量		
NetworkAddress	服务器网络 IP 地址	UInt32	变量		M
WAPGateway*	指示设备是原服务器 或代理	UInt8	0x01=原服务器 0x02=代理 0x03=原服务器 和代理		M
HomePageURL	主页 URL	URL			C1+
注: * 第二步互操作性要求 + 如果忽略参数,将指定原服务器的缺省 URL: http://networkaddress/index.wml					

表 9.2 SDP 协议数据单元

项目	定义	类型	值	属性 ID	是否必需
ServiceClassIDList				0x0001	M
ServiceClass0	WAP 客户	UUID	WAP_CLIENT		M
BluetoothProfile DescriptorList					M
ProfileDescriptor0	支持的标准		LANAccessUsing PPP		M
Profile	标准版本	UUID	变量		M
Version		UInt16	变量		M

续表

项目	定义	类型	值	属性 ID	是否必需
ServiceName	可显示的客户 端文本名字	String			M

表 9.3 SDP PDU

PDU 号	SDP PDU	发送能力		检索能力	
		WAP 客户	WAP 代理	WAP 客户	WAP 代理
1	SdpErrorResponse	M	M	M	M
2	SdpServiceSearchAttributeRequest	M	M	M	M
3	SdpServiceSearchAttributeResponse	M	M	M	M

9.5.2 服务搜索过程

信号发送过程，如图 9.4 所示。

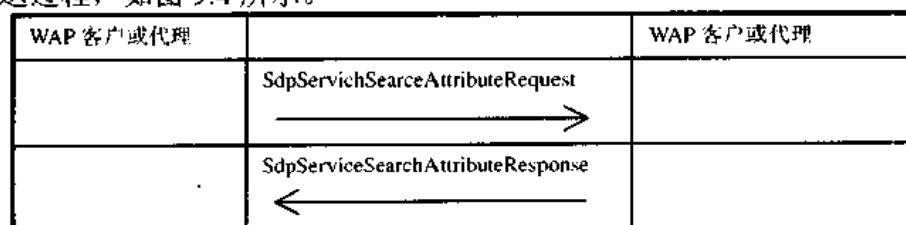


图 9.4 服务搜索过程

WAP 服务搜索过程是对称的。每一设备都必须能够处理 PDU，而与设备当前角色无关。至少必须返回服务名字字符串。

第 10 章 主控制器接口功能规范

本章对主控制器接口（HCI）的功能作出描述。HCI 提供对基带控制器和链路管理器的命令接口，以及对硬件状态和控制注册成员的访问。该接口还提供对蓝牙基带的统一访问模式。

10.1 节概要介绍蓝牙软件栈和底层蓝牙硬件；以及主机上低层 HCI 设备驱动器接口；10.2 节对主机和主控制器间的流控制进行描述；10.3 节详细阐述 HCI 命令，并标识出每一命令的参数，列出每一命令相关事件；10.4 节详细介绍事件及其参数描述；10.6 节介绍错误码表及其错误码的用法描述。

10.1 概述

10.1.1 蓝牙软件栈底层

图 10.1 提供了对软件层低层的概述。通过访问基带命令对链路管理器、硬件状态注册器、控制注册器、事件注册器等访问，HCI 规范实现了蓝牙硬件 HCI 命令。

在主机系统 HCI 驱动程序和蓝牙硬件 HCI 基础间存在许多层次。这些中间层和主控制器传输层提供了在没有数据描述信息的情况下传输数据的能力。

主机 HCI 驱动程序在蓝牙硬件上与 HCI 基础交换数据和命令。主机控制传输层的驱动程序（如物理总线）为 HCI 两层提供互相交换信息的能力。

主机将收到 HCI 事件的异步通知，而不管正在使用哪个主控制器传输层。HCI 事件用于在事件发生时通知主机。当主机发现某事件已经发生，它就会分析接收的事件分组以确定发生的是哪个事件。

10.1.2 蓝牙硬件块描述

蓝牙硬件由模拟部分和数字部分组成。模拟部分指蓝牙发射台，数字部分指主控制器。主控制器包括一个硬件信号处理部分——链路控制器（LC），一个 CPU 内核，以及主机环境接口。

1. 链路控制器

链路控制器（LC）由硬件部分、软件部分和物理层协议组成，前两部分执行蓝牙基带处理，后者如 ARQ 协议和 FEC 编码。

链路控制器的功能包括：

- 含有指定服务质量参数的传输类型；
- 利用使用硬件快速自动重新请求的授权传递进行异步传输。帧可从重发缓冲区中溢出，以利用等时数据；
- 同步传输；
- 语音编码。通过健壮的 64 位 CVSD 编码方式和 LOG-PCM 编码方式的硬件实现；

• 加密。

2. CPU 内核

CPU 内核使蓝牙模块处理查询和过滤呼叫请求。主控制器能够通过编程应答某些呼叫消息和鉴权远程链接。

链路管理器(LM)软件运行在 CPU 内核上。LM 通过链路管理协议找到远程 LM 并通信，以利用底层链路控制器提供其服务。具体细节详见“链路管理器协议”。

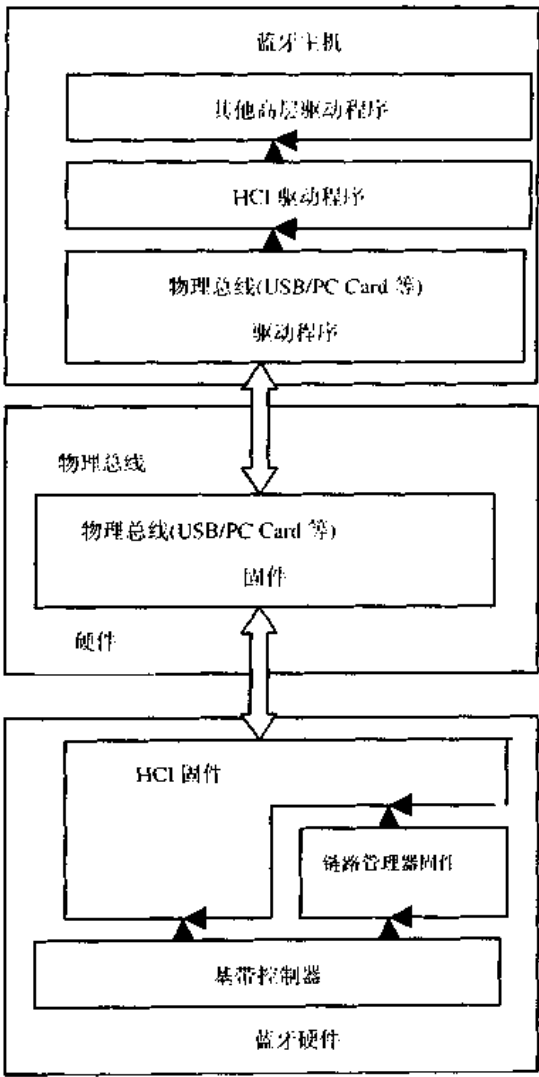


图 10.1 底层软件概览

10.1.3 物理总线体系结构

蓝牙设备具有多种能够用于链接蓝牙硬件的物理总线接口。这些总线具有不同的体系结构和参数。蓝牙主控制器初步支持两种物理总线体系结构：USB 和 PC 卡。

1. USB HCI 体系结构

图 10.2 中的方框表示通过 USB HCI 到远程计算机的链接。USB 支持在同一个物理通道

上处理多个逻辑通道。因此控制、数据和语音通道不再需要额外的物理接口。但注意不能通过 USB 直接访问蓝牙模块的注册表/内存。如果要做到这一点，则需要使用合适的 HCI 指令和主控制器传输层接口。

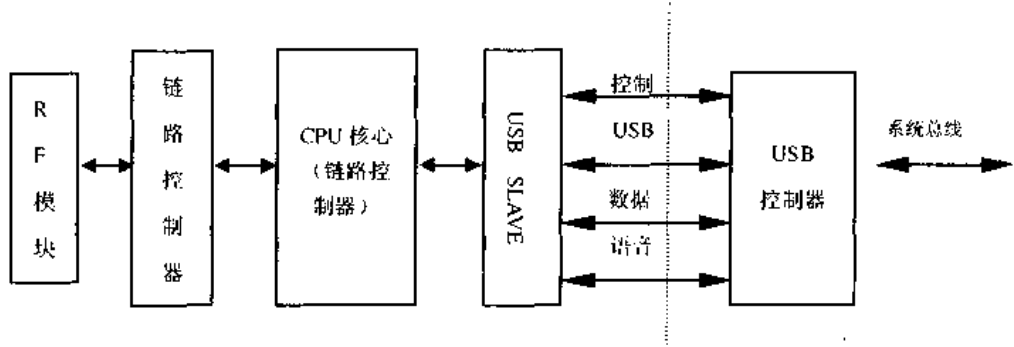


图 10.2 USB HCI 蓝牙模块配置图

2. PC Card HCI 体系结构

除了 USB 接口，改进的 ISA 总线也可以作为集成 PC 解决方案的选择。与 USB 不同，所有主机和蓝牙模块间的通信都将经过 PC 卡总线接口。在主机 PC 和蓝牙模块之间的通信将通过直接访问注册器/内存进行。图 10.3 中的方框表示 PC 卡 HCI 的数据流。

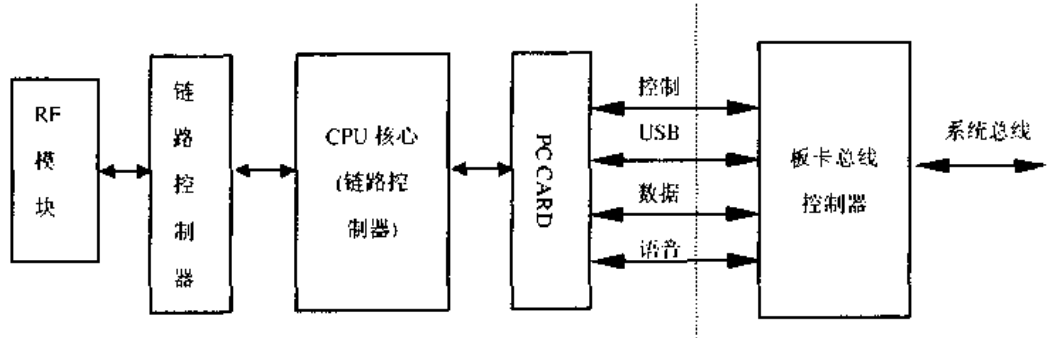


图 10.3 PC 卡蓝牙模块配置图

10.1.4 主控制器层概述

主驱动器栈是介于主控制器驱动程序和主控制器间的传输层。在笔记本电脑上，该传输层可以是 PC 卡或通用串口总线(USB)。

传输层主要目的是实现透明性。与主控制器对话的主控制器驱动程序不关心它是运行在 USB 上还是在 PC 卡上。无论 USB 还是 PC 卡都不需要对主控制器驱动程序传送给主控制器的数据可见。这就使接口(HCI)或主控制器能在不影响传输层的情况下升级。

主控制器传输层在下列物理介质的说明文档中进行描述：

- HCI USB 传输层；
- HCI RS232 传输层；
- HCI UART 传输层。

10.2 HCI 流控制

流控制在主机和主控制器之间，避免将传送到未应答远程设备的 ACL 数据溢出主控制器数据缓冲区。主机负责管理主控制器的数据缓冲区。

主机通过发送 `Read_Buffer_Size` 指令进行初始化。通过该指令返回的参数可以确定从主机发往主控制器的 HCI ACL 和 SCO 数据分组(不包括报头)的最大长度。另有两个返回参数表示主控制器为等待传输可以缓存的 HCI ACL 和 SCO 数据分组的数量。在只有一个链接或处于本地回送的情况下，主控制器利用已完成数据分组事件控制从主机发来的数据流。事件分组包括一个链接句柄列表，以及从事件返回后已经完成发送的 HCI 数据分组的应答数量。如果该事件没有返回指定链接句柄，则从链接创建开始。发送完成是指数据分组的传输、溢出和回送至主机。根据事件返回信息和存放在主控制器的 `Read_Buffer_Size` 指令返回参数，主机决定后面 HCI 数据分组发送哪一个链接句柄。每次发送 HCI 数据分组以后，主机都假定对应于链路类型的主控制器的一部分空闲缓存空间已被 HCI 数据分组占用。主机收到新的已完成数据分组事件，获取自上次事件返回以后减少的可用缓存空间大小，它就可以计算当前实际可用缓存。当主控制器在其缓存中存放有 HCI 数据分组时，它必须向主机周期性持续发送已完成数据分组事件，直到所有 ACL 数据分组都已发送完毕或溢出。事件发送频率由厂商指定。注意：如果 SCO 流控制失效，则已完成数据分组事件号就不能在 SCO 链接句柄中进行报告(参见 `Read/Write_SCO_Flow_Control_Enable`)。

对于每一链接句柄，数据都应按照它在主机内的创建顺序，以 HCI 数据分组的形式发送到主控制器。主控制器也以相同的顺序传输从主机收到的数据。同样，从其他设备收到的数据也可作相同处理。这就意味着应在链接句柄的基础上排序。对于每一链接句柄，数据顺序应与其创建时保持一致。

在某种情况下，必须在主控制器到主机的方向上采用流控制。一般采用 `Set_Host_Controller_To_Host_Flow_Control` 指令关闭或打开流控制。如果流控制已打开，工作方式如上所述。初始化时，主机利用 `Host_Buffer_Size` 指令通知主控制器发往主机的 HCI ACL 和 SCO 数据分组最大尺寸。该指令还包括其他两个参数，用以通知主控制器在主机数据缓存区中能够存储的 HCI ACL 和 SCO 数据分组的数量。主机就像主控制器利用已完成数据分组数量一样利用 `Host_Number_Of_Completed_Packets` 指令。`Host_Number_Of_Completed_Packets` 指令用于无流控制指令可用的情况下，只要存在链接或处于本地回送模式时就可以发送该指令。这就使得流控制可以以同样方式实现双向，而且不干扰正常指令流。

主机收到断开链接完成事件后，就可以认定相对于返回的 `Connection_Handle` 而发送到主控制器的所有 HCI 数据分组都已溢出，而且相应的数据缓存已被释放。主控制器不必再以完成数据分组事件数量的形式将此通知主机。如果在从主控制器到主机的方向上采用流控制，主控制器将在发送 `Disconnection_Complete` 后，认定主机收到 `Disconnection_Complete` 时将释放已发送的 `Connection_Handle` 所占用的缓存。主机不必再以 `Host_Number_Of_Completed_Packets` 的形式将该信息通知主控制器。

10.3 HCI 指令

10.3.1 引言

HCI 提供访问蓝牙硬件的统一指令方式。HCI 链路指令使主机能够控制到其他蓝牙设备的链路层链接。这些指令通过链路控制器(LM)与远程蓝牙设备交换 LMP 指令。具体细节请参见“链路管理器协议”。

HCI 策略指令用于本地或远程 LM。这些策略指令向主机提供影响 LM 管理匹克网的方法。主控制器和基带、信息和状态指令可为主机提供访问主控制器中不同注册表的能力。

执行 HCI 策略指令将耗费不同时间。因此，指令结果将以事件的形式返回给主机。例如，对于大多数 HCI 指令，主控制器在该指令完成时将生成命令完成事件。该事件分组包括已完成 HCI 指令的返回参数。为了在 HCI 传输层上检测出错信息，必须定义在主控制器收到命令和发出应答之间的应答时间。由于最大应答时间取决于所采用的 HCI 传输层，因此推荐采用 1 秒的缺省值。这个事件值也取决于在指令队列中未处理指令的数量。

为了便于描述，下面先介绍几个术语。

基带数据分组：数据的最小单位，它在各个设备之间进行传输。

数据分组：是比基带分组更高层次的协议报文，目前只定义了 L2CAP，参见‘逻辑链路控制和适配协议规范’。其他分组类型在今后逐步定义。

链接句柄：一个链接句柄就是一个用于惟一标识蓝牙设备之间数据或语音链接的 12 位标识符。链接句柄可以通过惟一标识两蓝牙设备间的数据管道进行访问。同时，无论设备进入空闲、休眠还是保持状态，都应在链接的整个生命周期内保持链接句柄。链接句柄值在主机和主控制器间取本地值。在两个蓝牙设备间可以拥有多个链接句柄，但只能保持一个 ACL 链接。

事件：HCI 用于通知主机命令完成和链路层状态变动等信息的一种机制。

10.3.2 数据和参数格式

- 如没有另外指定其他格式，所有值都应采用二进制或 Big Edian 码；
- 另外，当定义值时，所有具有负值的参数必须使用两位补码；
- 参数数组采用以下两个概念进行定义：ParameterA[i]，如果参数数组的一个集合定义为如 ParameterA[i]、ParameterB[i]的格式，那么参数数组顺序如下：ParameterA[0]、ParameterB[0]、ParameterA[1]、ParameterB[1].....ParameterA[n] ParameterB[n]；
- 如果没有另外给出说明，所有参数值都应按 Little Edian 码的格式发送和接收；
- 所有非数组的命令和指令参数，以及所有参数数组元素都具有固定长度。参数和非数组参数的长度都包含在一条指令里，并为每一命令或事件定义事件。参数数组内元素数量可以不定。

10.3.3 HCI 信息交换

主控制器传输层提供 HCI 信息的透明传输。该传输机制为主机提供向主控制器发送 HCI 指令、HCI 数据和 SCO 数据，以及从主控制器接收 HCI 事件、ACL 数据和 SCO 数据的能力。

力。

由于主控制器传输层提供 HCI 信息的透明传输，HCI 规范对主机和主控制器间交换的指令、事件、数据的格式进行定义。下面就 HCI 分组格式做出说明。

1. HCI 指令分组

HCI 指令分组用于从主机向主控制器发送指令。HCI 指令分组的格式如图 10.4 所示，其中每一段的定义如表 10.1 所示。当主控制器完成大多数指令的发送时，就向主机发送指令完成事件。当然，其中一些指令在它们完成后并不接收指令完成事件。相反，当主控制器收到一个 HCI 指令并准备执行时，它将向主机返回一个指令状态事件。然后，当与该指令相关联的动作执行后，对应于该发出指令的事件也将由主控制器发往主机。但是，如果由于参数错误或该指令非法等原因而导致该指令不能执行时，则不返回对应于该发出指令的事件。这时，指令状态事件将在状态参数中返回应答的出错码。开启电源或重新启动时，主机在收到指令完成或指令状态事件之前会一直发送一个最大长度的 HCI 指令数据分组。如果在返回指令完成事件时出错，Return_Parameter 段就不能包含该指令的返回参数，而是返回用于解释出错原因，同时也是第一个返回值的状态参数。如果在 Status 参数后紧接着是 Connection_Handle 或 BD_ADDR 参数，仍需返回该参数，以便主机能够判定指令完成事件属于哪个指令实例。这时，Connection_Handle 或 BD_ADDR 参数与对应指令参数具有相同值。出错时返回参数个数根据应用不同情况确定。

如果出错时，一条指令没有返回指令完成事件，那么与此指令关联事件的所有返回值都为非法值。主机必须关心哪个参数具有合法的取值，而这取决于关联于给定指令的指令完成事件的状态参数值。指令完成事件和指令状态事件分组包含一个叫做 Num_HCI_Command_Packets 的参数。该参数表示主机当前允许发往主控制器的 HCI 指令分组数量。主控制器可以缓存一个或多个 HCI 指令分组，主控制器应按照收到的顺序执行指令。但主控制器可以在前一条指令未执行完时开始执行下一条指令。因此，实际上指令并不一定按照它们最初收到时的顺序执行。主控制器必须能够接收 HCI 指令分组和除 HCI 指令报头以外的 255 字节数据。

每一指令都指定了一个 2 字节的操作码，用于惟一标识指令类型。操作码参数分为两段，操作组段(OGF)和操作码指令段(OCF)。OGF 占用操作码的上 6 位，OCF 占用其余 10 位。OGF 的 0x3E 保留用于厂商调试，0x3E 保留用于蓝牙标志测试。操作码的结构使得能够在不必对整个操作码完全解码的情况下即可获知附加信息。

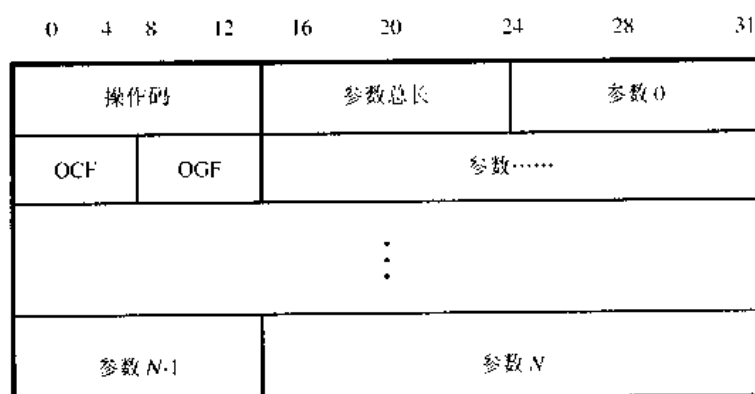


图 10.4 HCI 指令分组

表 10.1 HCI 指令分组各段定义

名 称	值	参 数 描 述
操作码 (2 字节)	0xXXXX	OCF 占用 6 位: 0x00~0x3F(0x3F 保留用于蓝牙标志测试, 0x3F 保留用于用户调试指令); OCF 占用 10 位: 0x0000~0x03FF
Parameter_Total_Length (1 字节)	0xXX	所有数据分组中的参数总长以字节度量 N.B.: 参数总长, 不是参数个数
参数 0~N	0xXX	每一指令都有几个与其关联的参数。这些参数及其大小由指令定义, 其大小一般为整数个字节

2. HCI 事件分组

主控制器利用 HCI 事件分组在事件发生时通知主机。主机必须能够接收 HCI 指令和除 HCI 指令报头以外的 255 字节数据。HCI 事件分组格式如图 10.5 所示, 各段定义如表 10.2 所示。

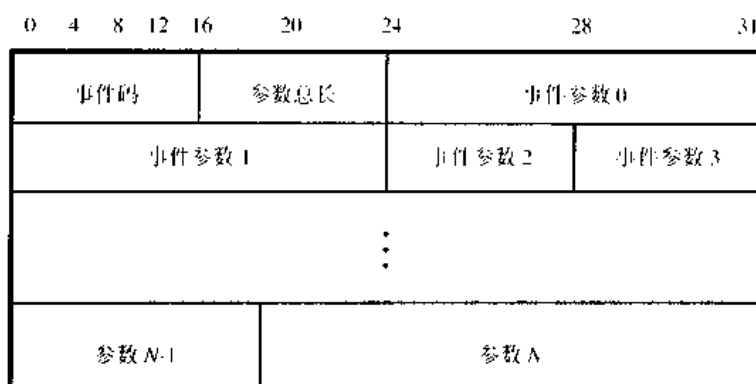


图 10.5 HCI 事件分组

表 10.2 HCI 事件分组各段定义

名 称	值	参 数 描 述
操作码 (2 字节)	0xXX	每一事件都指定 1 字节事件码, 以惟一标识不同事件类型 范围: 0x00~0xFF(事件码 0xFE 保留用于用户调试事件事件码。而且, 事件码 0xFE 保留用于蓝牙标志测试。)
参数总长 (1 字节)	0xXX	所有数据分组中的参数总长以字节度量
参数 0~N	0xXX	每一指令都有几个与其关联的参数。这些参数及其大小由指令定义, 其大小一般为整数个字节

3. HCI 数据分组

HCI 数据分组用于在主机和主控制器之间交换数据。数据分组根据 ACL 和 SCO 数据分组类型定义。HCI ACL 数据分组格式如图 10.6 所示, SCO 数据分组格式如图 10.7 所示。数据分组中各段定义分别如表 10.3 和表 10.4 所示。

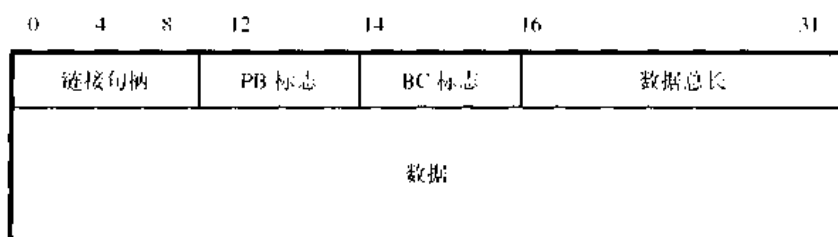


图 10.6 HCI ACL 数据分组

表 10.3 HCI ACL 数据分组各段定义

名 称	值	参 数 描 述
PB 标志 (2 位)	00	保留
	01	用于高层报文的数据分组成段
	10	高层报文的第一个数据分组(即: L2CAP 数据分组的开始)
	11	保留
BC 标志 (2 位)	00	没有广播, 只是点到点
	01	激活的广播: 包发往所有激活的从单元
	10	匹克网广播(数据分组发往所有从单元, 包括休眠单元)
	11	保留
数据总长 (2 字节)	0xFFFF	数据总长以字节度量
链接句柄 (12 位)	0xFFFF	链接句柄用于传输数据或段 范围: 0x0000~0xFFFF(0x0F00~0xFFFF 保留) 注: 主控制器一定不会发送包括用于广播的新 Connection_Handle 值的链接完成事件 注: 有些情况下, 会发生这样一种情况: 主控制器在解释从主机收到的广播包之前发送链接完成事件, 而且链接完成事件和 HCI 数据分组具有相同 Connection_Handle 值。为了避免这种冲突, 应作以下处理: 如果收到包含用于广播链接句柄的链接完成事件, 主机在为新的链接发送数据分组之前必须处于等待状态, 一直要等到它收到一些已完成包事件, 这些事件表示已没有属于链接句柄的广播包。此外, 主机必须由用于对应广播类型的 Connection_Handle 改变为不由主控制器指定的 Connection_Handle。Connection_Handle 必须用于下面的广播类型, 直到重新启动或发生相同的冲突, 然而这种情况很少发生 在上述冲突情况下, 主控制器必须能够区分由主机发出的广播报文和新链接发出的报文, 尽管其链接句柄值相同 对于从 Broadcast_Flag 为 01 或 10 的主机发送到主控制器的 HCI 数据分组, Connection_Handle 参数应包括到发送广播的主控制器的 ACL 链接的链接句柄 注: 用于广播的链接句柄不能判定一个 ACL 点到点链接, 所以这些句柄不能用于含有 Connection_Handle 参数的指令, 而且这些句柄不能在具有 Connection_Handle 参数的任何事件中返回, 除了已完成包事件的数量

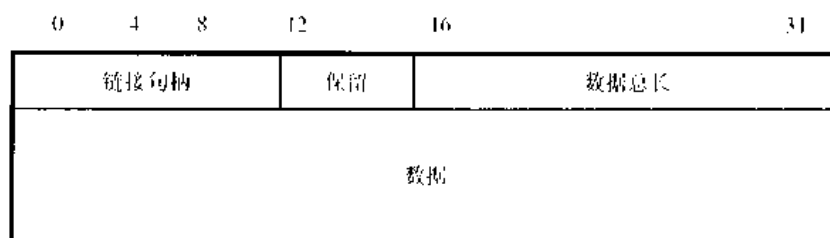


图 10.7 HCI SCO 数据分组

表 10.4 HCI SCO 数据分组各段定义

名 称	值	参 数 描 述
链接句柄 (12 位)	0xXXX	链接句柄用于传输 SCO 数据分组： 范围：0x0000~0x0EFF(0x0F00~0x0FFF 保留) 在开启电源或重新启动后，主机首次发送 Broadcast_Flag 置为 01b(广播激活)或 10b(匹克网广播)的 HCI 数据分组，Connection_Handle 值必须不是现在主控制器指定的值。主机对于激活广播和匹克网广播采用不同链接句柄。对于每一类链接，主控制器直到重新启动都要使用同一链接句柄。
保留字 (4 位)	XXX	保留
数据总长(2 字节)	0xFFFF	数据总长以字节度量

10.3.4 链路控制指令

链路控制指令（见表 10.5）允许主控制器控制到其他蓝牙设备的链接。使用链路控制指令时，链路管理器(LM)负责控制如何建立和保持蓝牙匹克网和散射网。这些指令指示 LM 创建和修改与蓝牙远程设备的链接层链接，执行对范围内其他蓝牙设备和 LM 指令的查询。对于链路控制指令，OGF 设为 0x01。下面分别对各条指令进行介绍。

表 10.5 链路控制指令一览表

指 令	指 令 描 述
Inquiry	该指令使蓝牙设备进入查询模式，查询模式用于搜索邻近的蓝牙设备。
Inquiry_Cancel	Inquiry_Cancel 使处于查询模式的蓝牙设备取消 Inquiry 模式。
Periodic_Inquiry_Mode	该指令用于将蓝牙设备配置为能够基于指定周期执行自动查询。
Exit_Periodic_Inquiry_Mode	该指令用于终止轮询模式，如果本地设备处于轮询模式。
Create_Connection	Create_Connection 指令使链路管理器能够利用指令参数定义的 BD_ADDR 创建到蓝牙设备的 ACL 链接。
Disconnect	本指令用于终止现有链接。
Add_SCO_Connection	该指令使链路管理器能够利用链接句柄指令参数指定的 ACL 链接创建 SCO 链接。
Accept_Connection_Request	Accept_Connection_Request 指令用于接收新的呼入链接请求。
Reject_Connection_Request	Reject_Connection_Request 指令用于拒绝新的呼入链接请求。
Link_Key_Request_Reply	该指令用于应答从主机控制器发出的链接字请求事件，并指定存储在主机上的链接字作为与 BD_ADDR 指定的蓝牙设备进行链接用的链接字。

续表

指 令	指 令 描 述
Link_Key_Request_Negative_Reply	如果主机上没有存储的链接字作为与 BD_ADDR 指定的蓝牙设备进行链接用的链接字, 该指令用于应答从主机控制器发出的链接字请求事件
PIN_Code_Request_Reply	该指令用于应答从主控制器发出的 PIN 编码请求事件, 并指定用于链接的 PIN 编码。
PIN_Code_Request_Negative_Reply	该指令指令用于在主机不能指定用于链接的 PIN 编码, 应答从主控制器发出的 PIN 编码请求事件
Change_Connection_Packet_Type	该指令用于改变用于正在建立的链接的包类型。
Authentication_Requested	该指令用于在与指定链接句柄关联的两个蓝牙设备之间建立身份鉴权。
Set_Connection_Encryption	该指令用于建立和取消链接层次的加密
Change_Connection_Link_Key	该指令用于强制关联到链接句柄的两个设备建立链接, 并生成一个新的链接字
Master_Link_Key	该指令用于强制关联到链接句柄的两个设备利用主设备临时链接字或常规字。
Remote_Name_Request	该指令用于获取另一蓝牙设备的用户名
Read_Remote_Supported_Features	该指令请求远程设备所支持特性的列表
Read_Clock_Offset	Read_Clock_Offset 指令允许主机读取远程设备时隙信息。

1. Inquiry

该指令使蓝牙设备进入查询模式, 用于搜索邻近的蓝牙设备。指令及其描述如表 10.6 和表 10.7 所示。其中, LAP 输入参数包含当进行查询过程时从查询访问码中得到的 LAP; Inquiry_Length 参数指定查询模式的持续时间, 超出该时间则终止查询; Num_Response 参数指定查询终止前能收到的应答数量。当查询指令由蓝牙设备开始启动时, 主控制器将指令状态事件发往主机。当查询进程结束时, 主控制器也向主机发送一指令状态事件, 表示已完成查询。查询指令事件的事件参数将从查询进程中得到一个结果集, 该结果集报告邻近发出应答的蓝牙设备数量。当蓝牙设备应答查询报文时, 将发生应答结果事件, 以通知主机搜索结果。一个在查询或查询周期内应答的设备, 通常应在查询结果事件中向主机报告已在当前查询或查询周期中报告该设备, 以及该设备是否已用 Set_Event_Filter 指令过滤掉。是否报告这些情况取决于实际执行情况。也就是说, 取决于先前结果是否已保存到主控制器, 以及已经保存了多少应答信息。推荐使用在一个查询或查询周期内主控制器只报告一个特定设备。

表 10.6 Inquiry 指令

指 令	OCF	指令参 数	返 回 参 数
HCI_Inquiry	0x0001	LAP, Inquiry_length, Num_Response	

表 10.7 Inquiry 指令描述

参 数	值	参 数 描 述
LAP (3 字节)	0x9E8B00~0X9E8B3F	当开始访问进程时, 访问识别码将从 LAP 引申而来。参见‘蓝牙分配号码’部分

续表

参 数	值	参 数 描 述
Inquiry_Length (1 字节)	N=0xXX	查询最长持续时间 大小: 1 字节; 范围: 0x01-0x30 时间=N*1.28 秒; 范围: 1.28 秒~61.44 秒
Num_Response	0x00	缺省值, 不限制应答次数
	0xXX	从查询开始的最大应答次数, 范围: 0x01-0xFF

当主控制器启动查询进程时, 从主控制器向主机发送一个指令状态事件。

对每一个应答查询消息的蓝牙设备都要创建一个查询结果集事件, 而且多个应答查询消息的蓝牙设备将生成同一事件。当查询进程结束时, 生成一个查询结束事件。

注意: 主控制器不是通过发出指令完成事件来表示指令完成, 而是通过发送查询完成事件来表示。取消查询进程不会生成查询完成事件。

2. Inquiry_Cancel

该指令使蓝牙设备能够终止当前查询。该指令允许主机中断蓝牙设备当前运行任务并请求蓝牙设备执行另一任务。该指令只能在查询指令发出和收到查询指令的指令状态事件后, 查询完成事件发生之前发出。指令及其描述如表 10.8 和表 10.9 所示。当查询取消事件完成时, 生成一个指令完成事件。对于已取消的查询进程不会生成查询完成事件。

表 10.8 Inquiry_Cancel 指令

指 令	OCF	指令参数	返回参数
HCI_Inquiry_Cancel	0x0002		Status

表 10.9 Inquiry_Cancel 指令描述

参 数	值	参 数 描 述
Status (1 字节)	0x00	Inquiry_Cancel 指令执行成功
	0x01-0xFF	Inquiry_Cancel 指令执行失败

3. Periodic_Inquiry_Mode

该指令用于配置蓝牙设备进入执行自动查询的轮询模式, 指令及其描述如表 10.10 和表 10.11 所示。其中 Max_Period_Length 和 Min_Period_Length 参数定义两个连续发生的查询之间的间歇时间长度, 该时间指从上一次查询的开始到下一次查询的开始之间的这段时间。主控制器利用这个时间范围在两次连续查询之间确定一次新的查询随机时间。当查询进程开始后, LAP 输入参数包括查询识别码引申来的 LAP。Inquiry_Length 参数指定查询模式的持续时间。时间超出时将终止查询。Num_Response 参数指定查询终止前能够接收的应答次数。当蓝牙设备一启动查询进程, 也就结束该指令。每一次轮询完成后, 主控制器将向主机发送查询完成事件, 表示最近一次轮询已经完成。查询完成事件的参数含有前一次轮询进程的结果集。该结果集报告邻近发出应答的蓝牙设备数量。当一个蓝牙设备对查询报文做出应答时, 就可以通过查询结果事件通知搜索到的主机。

表 10.10 Periodic_Inquiry_Mode 指令

指 令	OCF	指 令 参 数	返回参数
HCI_Periodic_Inquiry_Mode	0x0003	Max_Period_length, Min_Period_length, LAP, Inquiry_length, Num_Response	Status

表 10.11 Periodic_Inquiry_Mode 指令描述

参 数	值	参 数 描 述
Max_Period_Length (2 字节)	N=0xXXXX	两个连续查询之间的最大时间间隔 大小: 2 字节; 范围: 0x03 ~ 0xffff 时间=N*1.28 秒; 范围: 3.84 ~ 83884.8 秒 (0.0 ~ 23.3 小时)
Min_Period_Length (2 字节)	N=0xXXXX	两个连续查询之间的最小时间间隔 大小: 2 字节; 范围: 0x02 ~ 0xffff 时间=N*1.28 秒; 范围: 2.56 ~ 83883.52 秒 (0.0 ~ 23.3 小时)
LAP (3 字节)	0x9E8B00~0x9E8B3F	当查询进程开始后, LAP 输入参数包括查询访问码来源的 LAP.
Inquiry_Length (1 字节)	N=0xxx	指定查询模式的持续时间. 大小: 1 字节; 范围: 0x01 ~ 0x30 时间=N*1.28 秒; 范围: 1.28 ~ 61.44 秒
Num_Response (1 字节)	0x00	缺省值, 未限制应答次数
	0xxx	在查询终止前应答查询的最大次数, 范围: 0x01 ~ 0xff
Status (1 字节)	0x00	轮询模式指令成功
	0x01~0xFF	轮询模式指令失败

在查询或查询周期内应答的设备, 通常应在查询结果事件中向主机报告: 在当前查询或查询周期内是否已报告了该设备, 以及是否已经用 Set_Event_Filter 指令将该设备过滤。如果在当前查询或查询周期内已报告了该设备, 是否再报告它取决于实际应用情况。推荐在一次查询或查询周期内只对指定设备报告一次。

当主控制器向主机发送该指令的指令完成事件, 则轮询模式开始。应答查询报文的每一蓝牙设备都将创建一个查询结果事件。此外, 应答查询报文的多个蓝牙设备也将组合为同一事件。每次轮询进程结束时将生成一条查询完成事件。取消查询进程不生成任何查询完成事件。

4. Exit_Periodic_Inquiry_Mode

该指令用于在本地设备处于周期性查询模式时, 终止周期性查询模式。指令及其描述如表 10.12 和表 10.13 所示。如果本地设备当前处于查询进程中, 则将直接终止查询进程。而且主控制器将不再执行周期性查询, 直至周期性查询指令再次发出。

本指令的指令完成事件在本地设备不再处于周期性查询模式时发生。对于取消的查询进程不会生成查询完成事件。

表 10.12 Exit_Periodic_Inquiry_Mode 指令

指 令	OCF	指令参数	返回参数
HCI_Exit_Periodic_Inquiry_Mode	0x0004		Status

表 10.13 Exit_Periodic_Inquiry_Mode 指令描述

参 数	值	参 数 描 述
Status (1字节)	0x00	退出周期性查询指令成功
	0x01~0xFF	退出周期性查询失败

5. Create_Connection

本指令将使链路管理器创建与指令参数 BD_ADDR 指定的蓝牙设备的相互间链接，指令及其描述如表 10.14 和表 10.15 所示。本指令使本地蓝牙设备开始呼叫进程以创建一层链路层链接，链路管理器将确定新的 ACL 链接如何建立。ACL 链接由设备和匹克网的当前状态，以及要链接的设备状态决定。Packet_Type 指令参数指定链路管理器为 ACL 链接使用何种分组类型。链路管理器为了发送 HCI ACL 数据分组只能使用由 Packet_Type 指令参数指定的分组类型。可以通过执行不同分组类型间的位异或操作为分组类型参数指定多个分组类型，链路管理器将从可接受分组类型列表中选择哪一分组类型。Page_Scan_Repetition_Mode 和 Page_Scan_Mode 参数指定 BD_ADDR 代表设备的呼叫扫描模式。Clock_Offset 参数是本地时钟与 BD_ADDR 所代表的远程设备时钟之间的偏差，只能使用 2 到 16 位的偏差，且它们也将各自直接映射到参数的 0 到 14 位。Clock_Offset 的第 15 位 Clock_Offset_Valid_Flag 用于表示时钟偏差是否合法。在链接完成事件中将为该链接返回一链接句柄。Allow_Role_Switch 参数说明当远程设备在链接初始化过程中请求主从角色切换时（在本地主控制器返回链接完成事件之前），本地设备是接受还是拒绝该请求。不同分组类型定义详见“基带规范”。

注意：必须至少指定一种分组类型。主机最好能够启用尽量多的分组类型，以使链管理器能够高效执行。但是，主机不能启用本地设备不支持的分组类型。

表 10.14 Create_Connection 指令

指 令	OCF	指令参数	返回参数
HCI_Create_Connection	0x0005	BD_ADDR, Packet_Type, Page_Scan_Repetition_Mode, Page_Scan_Mode, Clock_Offset, Allow_Role_Switch	

表 10.15 Create_Connection 指令描述

参 数	值	参 数 描 述
BD_ADDR (6字节)	0xFFFFFFFFXXXX	要链接设备的 BD_ADDR
Packet_Type (2字节)	0x0001	保留
	0x0002	保留

续表

参 数	值	参 数 描 述
	0x0004	保留
	0x0008	D M 1
	0x0010	D H 1
	0x0020	保留
	0x0040	保留
	0x0080	保留
	0x0100	保留
	0x0200	保留
	0x0400	D M 3
	0x0800	D H 3
	0x1000	保留
	0x2000	保留
	0x4000	D M 5
	0x8000	D H 5
Page_Scan_Repetition_Mode (1 字节)	0x00	R 0
	0x01	R 1
	0x02	R 2
	0x03-0xFF	保留
Page_Scan_Mode (1 字节)	0x00	强制呼叫扫描模式
	0x01	呼叫扫描模式 1(可选)
	0x02	呼叫扫描模式 2(可选)
	0x03	呼叫扫描模式 3(可选)
	0x04-0xFF	保留
Clock_Offset (2 字节)	Bit 14	
	Bit15	Clock_Offset_Valid_Flag 非法时隙=0; 合法时隙=1
Allow_Role_Switch	0x00	本地设备为主设备,它不会在链接建立时接受由远程设备请求的主从互换
	0x01	本地设备为主设备,但在链接建立时可以接受由远程设备请求的主从互换,变为从设备
	0x02-0xFF	保留

当主控制器接收到链接创建指令时,主控制器向主机发送指令状态事件。而且当 LM 确定链接已经建立起来时,在两个蓝牙设备上形成链接的主控制器将向每一主机发送链接完成事件。如果该指令执行成功,则链接完成事件将包括链接句柄。

6. Disconnect

该指令用于终止现有链接,其指令参数表示要断开哪个链接。指令及其描述如表 10.16 和表 10.17 所示。Reason 指令参数表示终止链接的原因。远程蓝牙设备将在链接断开完成事件中接收 Reason 指令参数。在物理链接上 ACL 链接断开前,同一物理链接上所有 SCO 链

接都将被断开。

表 10.16 Disconnect 指令

指 令	OCF	指 令 参 数	返 回 参 数
HCI_Disconnect	0x0006	Connection_Handle; Reason	

表 10.17 Disconnect 指令描述

	值	参 数 描 述
链接句柄 (12 位)	0xXXXX	要断开的链接的链接句柄 范围: 0x0000-0x0FFF 保留使用
Reason (1 字节)	0x13~0x15, 0x1A	其他端终止链接出错码(0x13~0x15), 以及其他未给出的远 程部件出错码(0x1A); 范围: 0x0000~0x0FFF 保留使用

当主控制器接收到 Disconnect 指令时, 它向主机返回指令状态参数。当链接终止时每
一主机都将发生链接断开完成事件, 指示该指令已执行。主控制器不是通过发出指令完成事
件来表示指令完成, 而是通过发送查询完成事件来表示。取消查询进程不会生成查询完成事
件。

7. Add_SCO_Connection

该指令能够使链路管理器利用 Connection_Handle 指令参数指定的 ACL 链接, 创建一
个 SCO 链接。指令及其描述如表 10.18 和表 10.19 所示。该指令能够使本地蓝牙设备创建 SCO
链接, 由链路管理器确定如何建立新链接。该链接由设备和匹克网的当前状态, 以及要链接
设备的状态等因素确定。Packet_Type 指令参数用于说明链路管理器将在此链接上使用何种
报文类型。链路管理器只能使用由 Packet_Type 指令参数指定的报文类型, 以用于发送 HCI
SCO 数据分组。可以通过执行不同报文类型的逻辑或操作, 由 Packet_Type 指令参数指定多
个报文类型。链路管理器可以在多个可接收的报文类型中进行选择。在链接完成事件中返回
该链接的链接句柄。

注意, SCO 链接只能在 ACL 链接已经存在时创建, 而且至少应指定一种数据分组。主
机应尽量支持更多的报文类型, 以使链路管理器能够更有效工作。但是, 主机不能采用本地
设备不支持的报文类型。

表 10.18 Add_SCO_Connection 指令

指 令	OCF	指 令 参 数	返回参数
HCI_Add_SCO_Connection	0x0007	Connection_Handle; Packet_Type	

表 10.19 Add_SCO_Connection 指令描述

参 数	值	参 数 描 述
Connection_Handle (2 字节)	0xXXXX	用于 SCO 链接地 ACL 链接的句柄 范围: 0x0000-0x0EFF, (0x0F00~0x0FFF 保留)

续表

参 数	值	参 数 描 述
Packet_Type (2 字节)	0x0001	保留
	0x0002	保留
	0x0004	保留
	0x0008	保留
	0x0010	保留
	0x0020	HV1
	0x0040	HV2
	0x0080	HV3
	0x0100	保留
	0x0200	保留
	0x0400	保留
	0x0800	保留
	0x1000	保留
	0x2000	保留
	0x4000	保留
	0x8000	保留

当主控制器接收到 Add_SCO_Connection 指令时，将向主机发送指令状态事件。此外，当 LM 确认链接已建立，构成链接两设备的主控制器将向每一个主机发送链接完成事件。如果指令成功执行，则链接完成指令分组包括链接句柄。

注意：主控制器不是通过发出指令完成事件来表示指令完成，而是通过发送链接完成事件来表示。

8. Accept_Connection_Request

该指令用于接受最新的呼入链接请求，指令及其描述如表 10.20 和表 10.21 所示。该指令只能在链接请求事件发生后发出。链接请求事件将返回请求链接设备的 BD_ADDR。该指令使链路管理器能够利用由该指令参数指定的 BD_ADDR 创建到蓝牙设备的链接，由链路管理器确定如何建立新的链接，。该链接由设备和匹克网的当前状态，以及要链接设备的状态确定。Role 指令参数允许主机指定链路管理器是否执行主-从切换，以及是否成为链接的主设备。而且，在本地蓝牙模块上链接超时之前应确定是否接受链接。

表 10.20 Accept_Connection_Request 指令

指 令	OCF	指 令 参 数	返 回 参 数
HCI_Accept_Connection_Request	0x0009	BD_ADDR, Role	

表 10.21 Accept_Connection_Request 指令描述

参 数	值	参 数 描 述
BD_ADDR (6 字节)	0XXXXXX	要链接设备的 BD_ADDR
Role (1 字节)	0x00	成为链接的主设备,LM 将执行主/从切换
	0x01	保持为链接的从设备, LM 不执行主/从切换

当主控制器开始创建链接时，主控制器发送指令状态事件。此外，当链路管理器确认链接已建立，在链接的两端上蓝牙设备的主控制器将向主机发送链接完成事件。如果该指令成功完成，则链接完成事件分组包括链接句柄。

注意，主控制器不是通过发出指令完成事件来表示指令完成，而是通过发送链接完成事件来表示。

9. Reject_Connection_Request

该指令用于拒绝新的呼入请求，指令及其描述如表 10.22 和表 10.23 所示。只有在链接请求事件发生后才呼叫 Reject_Connection_Request。链接请求事件将返回请求链接设备的 BD_ADDR。Reason 指令参数将在状态参数中返回，表示拒绝的是哪条链接。该状态参数是指返回到链接设备主机的链接完成事件的状态参数。

表 10.22 Reject_Connection_Request 指令

指 令	OCF	指 令 参 数	返 回 参 数
HCI_Reject_Connection_Request	0x000A	BD_ADDR, Reason	

表 10.23 Reject_Connection_Request 指令描述

参 数	值	参 数 描 述
BD_ADDR (6 字节)	0xFFFFFFFFXXXX	拒绝链接宿主设备的 BD_ADDR
Reason (1 字节)	0x0D ~ 0x0F	主机拒绝出错码

当主控制器接收到 Reject_Connection_Request 指令时，主控制器向主机发送指令状态事件，并发送链接完成事件到本地主机和要建立链接的主机。链接完成事件的状态参数，用于发送到主机设备以建立链接，包括 Reason 指令参数。

注意，主控制器不是通过发出指令完成事件来表示指令完成，而是通过发送链接完成事件来表示。

10. Link_Key_Request_Reply

该指令用于应答从主控制器发送的链接字请求事件，并指定存储在主机的链接字，其指令及其描述如表 10.24 和表 10.25 所示。

当主控制器按序列为本地链路管理器生成链接字请求事件，以应答从远程链路管理器发送来的请求时（该链接字请求事件实质上也就是远程主机发送的 Create_Connection 或 Authentication 的结果集）本地主机必须在远程链路管理器侦测到 LMP 应答超时之前，利用 Link_Key_Request_Reply 或 Link_Key_Request_Negative_Reply 应答。

表 10.24 Link_Key_Request_Reply 指令

指 令	OCF	指 令 参 数	返 回 参 数
HCI_Reject_Connection_Request	0x000B	BD_ADDR, Link_Key	Status, BD_ADDR

表 10.25 Link_Key_Request_Reply 指令描述

参 数	值	参 数 描 述
BD_ADDR (6 字节)	0XXXXXXXXXXXXX	与链接字相关的设备的 BD_ADDR
Link_Key (16 字节)	0XXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXX	关联 BD_ADDR 的链接字
Status (1 字节)	0x00	Link_Key_Request_Reply 指令成功
	0x01 ~ 0xFF	Link_Key_Request_Reply 指令失败
BD_ADDR (6 字节)	0XXXXXXXXXXXXX	链接字请求应答完成设备的 BD_ADDR

11. Link_Key_Request_Negative_Reply

如果主机没有用于由 BD_ADDR 指定的其他蓝牙设备链接的链接字, 就可以利用 Link_Key_Request_Negative_Reply 指令应答从主控制器发出的链接字请求事件。指令及其描述如表 10.26 和表 10.27 所示。

当主控制器按序列为本地链路管理器生成链接字请求事件, 以应答从远程链路管理器发送来的请求时, (该链接字请求事件实质上也就是远程主机发送的 Create_Connection 或 Authentication 的结果集)本地主机必需在远程链路管理器侦测到 LMP 应答超时之前, 利用 Link_Key_Request_Reply 或 Link_Key_Request_Negative_Reply 应答。

表 10.26 Link_Key_Request_Negative_Reply 指令

指 令	OCF	指 令 参 数	返 回 参 数
HCI_Reject_Connection_Negative_Request	0x000C	BD_ADDR	状态, BD_ADDR

表 10.27 Link_Key_Request_Negative_Reply 指令描述

参 数	值	参 数 描 述
BD_ADDR (6 字节)	0XXXXXXXXXXXXX	与链接字相关的设备的 BD_ADDR
状态 (1 字节)	0x00	Link_Key_Request_Negative_Reply 指令成功
	0x01 ~ 0xFF	Link_Key_Request_Negative_Reply 指令失败
BD_ADDR (6 字节)	0XXXXXXXXXXXXX	链接字请求应答完成设备的 BD_ADDR

12. PIN_Code_Request_Reply

该命令用来答复主控制器的代码申请事件及说明用于链接的 PIN 码, 指令及其描述如表 10.28 和表 10.29 所示。当远程初始化设备有配对申请时, 将产生 PIN 码申请事件。当本地链路管理器响应远程链路管理器申请事件时, 主控制器产生 PIN 码申请事件(作为远程主设备创建链接 Create_Connection 或鉴权申请 Authentication_Requested 命令结果), 在远程链路管理器检测 LMP 响应超时前, 本地主设备必须响应 PIN_Code_Request_Reply 或 PIN_Code_Request_Negative_Reply 命令。

表 10.28 PIN_Code_Request_Reply 指令

命 令	OCF	命 令 参 数	返 回 参 数
HCI_PIN_Code_Request_Reply	0x000D	BD_ADDR, PIN_Code_Length, PIN_Code	Status, BD_ADDR

表 10.29 PIN_Code_Request_Reply 指令描述

参 数	值	参 数 说 明
BD_ADDR (6 字节)	0xFFFFFFFFXXXX	设备的 BD_ADDR 值
PIN_Code_Length (1 字节)	0xFF	被使用的 bd_addr 代码长度, 0x01~0xFF
PIN_Code (16 字节)	0xFFFFFFFFXXXX XXXXXXXXXXXX XXXXXX	PIN 码用于链接, 主设备应保证 PIN 码使用 PIN 码最大可到 128 位, PIN 码的 MSB 占 据字节零
Status (1 字节)	0x00	PIN_Code_Request_Reply 命令成功
	0x01~0xFF	PIN_Code_Request_Reply 命令失败
BD_ADDR (6 字节)	0xFFFFFFFFXXXX	BD_ADDR 代码申请答复完成

13. PIN_Code_Request_Negative_Reply

当主控制器不能指定用于链接的 PIN 码时, 该命令用于响应来自主控制器的 PIN 码申请事件。其指令及其描述如表 10.30 和表 10.31 所示。

当本地链路管理器响应远程链路管理器申请事件时, 主控制器产生 PIN 码申请事件作为远程主设备创建链接 (Create_Connection) 或鉴权申请 (Authentication_Requested 命令结果), 在远程链路管理器检测 LMP 响应超时前, 本地主设备必须响应 PIN_Code_Request_Reply 或 PIN_Code_Request_Negative_Reply 命令。

表 10.30 PIN_Code_Request_Negative_Reply 指令

命 令	OCF	命 令 参 数	返 回 参 数
HCI_PIN_Code_Request_Negative_Reply	0x000e	BD_ADDR	Status, BD_ADDR

表 10.31 PIN_Code_Request_Negative_Reply 指令描述

参 数	值	参 数 说 明
BD_ADDR (6 字节)	0xFFFFFFFFXXXX	该命令设备的 BD_ADDR 正在响应
Status (1 字节)	0x00	PIN_Code_Request_Negative_Reply 命令成功
	0x01~0xFF	PIN_Code_Request_Negative_Reply 命令失败
BD_ADDR (6 字节)	0xFFFFFFFFXXXX	消极应答 PIN 码设备的 BD_ADDR 完成

14. Change_Connection_Packet_Type

该命令用于当前确立链接的分组类型交换, 其指令及其描述如表 10.32 和表 10.33 所示。为支持不同类型的用户数据, 该命令允许当前链接动态修改。分组类型命令参数指定用于链接的链路管理器的分组类型。链路管理器只能使用由发送 HCI 数据分组 Packet_Type 命令参数指定的分组类型。Packet_Type 命令参数值的解释取决于在链接建立时, 链接完成事件里的 Link_Type 命令参数。

表 10.32 Change_Connection_Packet_Type 指令

命 令	OCF	命 令 参 数	返 回 参 数
HCI_Change_Connection_Packet_Type	0x000f	Connection_handle, Packet_type	

多分组类型可通过不同分组类型按位或操作的 **Packet_Type** 命令参数来指定。

注意：至少一种分组类型必须被确定。为保证链路管理器有效执行，主控制器应允许尽可能多的分组类型。然而，主机不允许本地设备不支持的分组类型。

表 10.33 Change_Connection_Packet_Type 指令描述

参 数	值	参 数 说 明
Connect_Handle 2 字节(12 位有意义)	0xxxxx	链接句柄用于发射和接收声音或数据。从创建链接里返回。范围：0x0000~0x0FFF (0x0F00~0x0FFF 保留)

ACL 链接类型

值	参数说明
0x0001	保留
0x0002	保留
0x0004	保留
0x0008	DM1
0x0010	DH1
0x0020	保留
0x0040	保留
0x0080	保留
0x0100	保留
0x0200	保留
0x0400	DM3
0x0800	DH3
0x1000	保留
0x2000	保留
0x4000	DM5
0x8000	DH5

SCO 链接类型

值	参数说明
0x0001	保留
0x0002	保留
0x0004	保留
0x0008	保留
0x0010	保留
0x0020	HV1
0x0040	HV2
0x0080	HV3
0x0100	保留
0x0200	保留
0x0400	保留
0x0800	保留
0x1000	保留
0x2000	保留
0x4000	保留
0x8000	保留

当主控制器收到变化链接分组类型命令时，主控制器发送命令状态事件到主机。另外，当链路管理器确定因链接分组已发生变化时，本地设备的主控制器将发送链接分组类型变化事件给主机。

15. Authentication_Requested

该命令用于期望鉴权指定链接句柄的远程设备，其指令及其描述如表 10.34 和表 10.35 所示。主机不发布使用与加密链接相符链接句柄的 **Authentication_Requested** 命令。在鉴权失败时，主控制器或链路管理器将不能自动分离。如果操作适当，主机可通过发布断开命令以终止链接。注意，链接句柄命令参数用来识别链接形式的其他蓝牙设备，链接句柄是 ACL 链接方式的链接句柄。

当主控制器收到 **Authentication_Requested** 命令时，它发送命令状态事件到主机。当链接的鉴权完成时，鉴权完成事件出现而且指出该命令已完成。

注意：通过主控制器传送的无命令完成事件指出该命令已完成，代替了鉴权完成事件指出的该命令已完成。当本地或远程主控制器没有链接字作为指定的 **Connection_Handle** 时，

在本地主机最后收到鉴权完成事件前，它将申请来自于主机的链接字。

表 10.34 Authentication_Requested 指令

命 令	OCF	命令参数	返回参数
HCI_Authentication_Requested	0X0011	Connection_Handle	

表 10.35 Authentication_Requested 指令描述

参 数	值	参 数 说 明
Connection_Handle 2 字节(12 位有意义)	0Xxxxx	用来设置鉴权的链接句柄，对于所有用相同蓝牙设备端点链接句柄的链接句柄。范围：0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)

16. Set_Connecion_Encryption

该命令用于允许和禁止链接层加密，其指令及其描述如表 10.36 和表 10.37 所示。链接句柄命令参数用来识别链接形式的其他蓝牙设备。链接句柄应是 ACL 链接句柄。当加密被改变时，由于链接句柄与远程设备有关，所以整个 ACL 通信必须关闭。

表 10.36 Set_Connecion_Encryption 指令

命 令	OCF	命 令 参 数	返回参数
HCI_Set_Connection_Encryption	0x0013	Connection_Handle, Encryption_Enable	

表 10.37 Set_Connecion_Encryption 指令描述

参 数	值	参 数 说 明
Connection_Handle 2 字节(12 位有意义)	0Xxxxx	使用相同蓝牙设备终端当作指定链接句柄的全部链接句柄用于允许和禁止链接层加密。范围：0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)
Encryption_Enable (1 字节)	0x00	关闭链接层加密
	0x01	打开链接层加密

当主控制器收到 Set_Connection_Encryption 命令时，主控制器发送命令状态事件到主机。当链路管理器为链接完成了允许 / 禁止加密时，本地蓝牙设备上的主控制器将发送加密变化事件到主机，而且远程设备上的主控制器也产生加密变化事件。

注意：通过主控制器传送的无命令完成事件指出该命令已完成，代替了加密变化事件指出的该命令已完成。

17. Change_Connection_Link_Key

该命令强制使用链接句柄的双方产生新的链接字，链接字用作链接的加密和鉴权。其指令及其描述如表 10.38 和表 10.39 所示。Connection_handle 命令参数用来识别链接形式的其他蓝牙设备。链接句柄是 ACL 链接形式的链接句柄。如果允许链接加密和使用当前临时链接字，则蓝牙主单元设备将自动重新加密。

表 10.38 Change_Connection_Link_Key 指令

命 令	OCF	命 令 参 数	返回参数
HCI_Change_Connection_Link_Key	0x0015	Connection_Handle	

表 10.39 Change_Connection_Link_Key 指令描述

参 数	值	参 数 说 明
Connection_Handle 2 字节(12 位有意义)	0Xxxxx	用来识别链接的链接句柄。 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)

当主机控制器收到 Change_Connection_Link_Key 命令时, 主机控制器发送命令状态事件到主机。当链路管理器为链接改变了链接字时, 本地蓝牙设备上的主控制器将发送链接标志信息事件和改变链接链接字完成事件到主机, 且远程设备上的主控制器也产生链接标志信息事件。链接字标志信息事件指出新链接字对于链接是有效的。

注意: 通过主控制器传送的无命令完成事件指出该命令已完成, 代替了改变链接字完成事件指出的该命令已完成。

18. Master_Link_Key

该命令强迫匹克网中主单元使用主单元设备的临时链接字或半永久链接字, 其指令及其描述如表 10.40 和表 10.41 所示。临时链接字用在匹克网内广播消息的加密, 而半永久链接字用于点对点通讯单独加密。Key_Flag 命令参数用来指出使用哪个链接字(主单元临时链接字, 或半永久链接字)。

表 10.40 Master_Link_Key 指令

命 令	OCF	命 令 参 数	返回参数
HCI_Master_Link_Key	0x0017	KEY_FLAG	

表 10.41 Master_Link_Key 指令描述

参 数	值	参 数 说 明
Key_Flag (1 字节)	0x00	使用半永久链接字
	0x01	使用临时链接键

当主控制器收到 Master_Link_Key 命令时, 主控制器发送命令状态事件到主机。当链路管理器改变链接字时, 在本地和远程设备上的主控制器发送主单元链接字完成事件到主机。主单元方的链接句柄是现有的链接从单元之一的链接句柄。从单元方的链接句柄是初始化主单元链接句柄。主单元链接字完成事件包含该命令的状态。

注意: 通过主控制器传送的无命令完成事件指出该命令已完成, 代替了主单元链接字完成事件指出的该命令已完成。

19. Remote_Name_Request

该命令用来获得其他蓝牙设备的用户界面友好名, 其指令及其描述如表 10.42 和表 10.43 所示。BD_ADDR 命令参数用来识别用户界面友好名获得的设备。Page_Scan_Repetition_Mode 和 Page_Scan_Mode 命令参数, 指出由使用 BD_ADDR 的远程设备支持

的呼叫扫描模式，该信息在查询过程期间获得。Clock_Offset 参数在本地时钟和用 BD_ADDR 远程设备的时钟之间有差异。Clock_Offset_Valid_Flag 定义在 Clock_Offset 命令参数的 15 位，用来指出时钟补偿是否有效。

注意：如果在本地设备和相应 BD_ADDR 的设备之间无链接存在，临时链接层链接将确立获得远程设备名。

表 10.42 Remote_Name_Request 指令

命 令	OCF	命 令 参 数	返回参数
HCI_Remote_Name_Request	0X0019	BD_ADDR, Page_Scan_Repetition_Mode, Page_Scan_Mode, Clock_Offset	

表 10.43 Remote_Name_Request 指令描述

参 数	值	参 数 说 明
BD_ADDR (6 字节)	0XXXXXXXXXXXXX	命名设备的 BD_ADDR
Page_Scan_Repetition_Mode (1 字节)	0x00	R0
	0x01	R1
	0x02	R2
	0x03 ~ 0xFF	保留
Page_Scan_Mode (1 字节)	0x00	强制呼叫扫描模式
	0x01	可选呼叫扫描模式 I
	0x02	可选呼叫扫描模式 II
	0x03	可选呼叫扫描模式 III
	0x04 ~ 0xFF	保留
Clock_Offset (2 字节)	位 14, 0	clkslave-clkmaster 的 16, 2 位
	位 15	Clock_Offset_Valid_Flag 无效时钟补偿 Offset = 0; 有效时钟补偿 Offset = 1

当主控制器收到 Remote_Name_Request 命令时，主控制器发送命令状态事件到主机。当链路管理器已完成 LMP 消息获得远程名时，本地蓝牙设备上的主控制器将发送远程名申请完成事件到主机。

注意，通过主控制器传送的无命令完成事件指出该命令已完成，代替了远程名申请事件指出的该命令已完成。

20. Read_Remote_Supported_Features

该命令申请通过链接句柄参数识别远程设备支持特征的一张表，其指令及其描述如表 10.44 和表 10.45 所示。链接句柄必须是 ACL 链接方式链接句柄。阅读远程支持特征完成事件将返回 LMP 特征的表。

表 10.44 Read_Remote_Supported_Features 指令

命 令	OCF	命 令 参 数	返回参数
HCI_Read_Remote_Supported_Features	0x001B	Connection_Handle	

表 10.45 Read_Remote_Supported_Features 指令描述

参 数	值	参 数 说 明
Connection_Handle 2 字节(12 位有意义)	0xFFFF	指出得到哪个链接句柄的 LMP 支持特征表 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)

当主控制器收到 Read_Remote_Supported_Features 命令时, 主控制器发送命令状态事件到主机。当链路管理器已完成 LMP 消息并确立远程特征时, 本地蓝牙设备上的主控制器将发送读远程支持特征完成的事件到主机。读远程支持特征完成事件包含该命令的状态, 而且参数描述远程设备的支持特征。

注意: 通过主控制器传送的无命令完成事件指出该命令已完成, 代替了读远程支持特征完成事件指出的该命令已完成。

21. Read_Remote_Version_Information

该命令通过链接句柄参数识别远程蓝牙设备版本信息的获取值, 其指令及其描述如表 10.46 和表 10.47 所示。链接句柄必须是 ACL 链接方式链接句柄。

表 10.46 Read_Remote_Version_Information 指令

命 令	OCF	命令参数	返回参数
HCI_Read_Remote_Version_Information	0x001D	链接句柄	

表 10.47 Read_Remote_Version_Information 指令描述

参 数	值	参 数 说 明
Connection_Handle 2 字节(12 位有意义)	0xFFFF	指出获得哪个链接句柄的版本信息 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)

当主控制器收到 Read_Remote_Version_Information 命令时, 主控制器发送命令状态事件到主机。当链路管理器完成 LMP 确定远程版本信息时, 本地蓝牙设备上的主控制器将发送读远程版本信息完成事件到主机。读远程版本信息完成事件包含了该命令的状态, 而且参数描述了通过远程设备使用 LMP 的版本及损坏。

注意: 通过主控制器传送的无命令完成事件指出该命令已完成, 代替了读远程版本信息完成事件指出的该命令已完成。

22. Read_Clock_Offset

使用系统时钟和远程设备的时钟补偿来确定用于呼叫扫描的远程设备采用的跳频频率。该命令允许主机读远程设备的时钟补偿, 其指令及其描述如表 10.48 和表 10.49 所示。链接句柄必须是 ACL 链接方式的链接句柄。

表 10.48 Read_Clock_Offset 指令

命 令	OCF	命令参数	返 回 参 数
HCI_Read_Clock_Offset	0x001F	链接句柄	

表 10.49 Read_Clock_Offset 指令描述

参 数	值	参 数 说 明
Connection_Handle 2 字节(12 位有意义)	0xFFFF	指出返回哪个链接时钟补偿参数 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)

当主控制器收到 Read_Clock_Offset 命令时, 主控制器发送命令状态事件到主机。如果该命令在主单元和链路管理器已完成获得时钟补偿信息的 LMP 消息时被申请, 本地蓝牙设备上的主控制器发送读时钟补偿完成事件到主机。

注意: 通过主控制器传送的无命令完成事件指出该命令已完成, 代替了只读时钟补偿完成事件指出的该命令已完成。如果命令在从单元方申请, 且没有 LMP PDU 的互换, LM 将直接发送命令状态事件和读时钟补偿事件到主机。

10.3.5 链接策略命令

链接策略命令 (见表 4.50) 提供了主控制器如何影响匹克网链路管理器消息的方式。当使用链接策略命令时, LM 控制蓝牙匹克网和散射网怎样建立和维持, 取决于可调的策略参数。这些策略命令修改了链路管理器的状态, 而且能导致用蓝牙远程设备链接层链接的变化。

注意: 在两个蓝牙设备之间, 仅有一种 ACL 链接方式存在, 因此对各个物理层链接来说, 仅存在着惟一的 ACL HCI 链接句柄。蓝牙主控制器提供策略调整机制来支持多种策略, 它允许用一种蓝牙模型来支持多种不同模型。对于链接策略命令, OGF 设为 0x02。

表 10.50 链接策略命令

命 令	命令说明汇总
Hold_Mode	该命令用来改变 LM 状态和本地及远程设备为主模式的 LM 位置
Sniff_Mode	该命令用来改变 LM 状态和本地及远程设备为呼吸模式的 LM 位置
Exit_Sniff_Mode	该命令用来结束链接句柄在当前呼吸模式里的呼吸模式
Park_Mode	该命令用来改变 LM 状态和本地及远程设备为休眠模式的 LM 位置
Exit_Park_Mode	该命令用来切换从休眠模式返回到活动模式的蓝牙设备
QoS_Setup	该命令用来指出链接句柄的服务质量参数
Role_Discovery	该命令用于蓝牙设备确定设备正在履行特殊链接句柄的角色
Switch_Role	该命令用于蓝牙设备切换当前正在履行指定蓝牙设备特殊链接的设备角色
Read_Link_Policy_Setting	该命令为指定链接句柄读链接策略设置。链接策略设置允许主控制器指定可用于指定链接句柄的 LM 链接模式
Write_Link_Policy_Setting	该命令为指定链接句柄写链接策略设置。链接策略设置允许主控制器指定可用于指定链接句柄的 LM 链接模式

1. Hold_Mode

该命令用来改变链路管理器的状态, 并通过指定链接句柄为保持模式相关的 ACL 基带链接位置, 其指令及其描述如表 10.51 和表 10.52 所示。Hold_Mode_Max_Interval 和 Hold_Mode_Min_Interval 命令参数确定主控制器置链接为保持模式的时间长度。

Hold_Mode_Max_Interval 参数用来指出主控制器在协调远程设备后实际进入保持模式的保持间隔最大长度。保持间隔定义为保持模式开始和保持模式完成之间的时间量。

Hold_Mode_Min_Interval 参数用来指出主控制器在协调远程设备后实际进入保持模式的保持间隔最小长度。因此 Hold_Mode_Min_Interval 不能大于 Hold_Mode_Max_Interval。如果命令是成功的，主控制器将在模式变化事件的间隔参数里返回实际的保持间隔。该命令使主控制器能够为自己或若干其他的蓝牙设备提供低功耗策略，而且允许设备进入查询扫描和许多其他的可能行为。

注意：链接句柄不能是 SCO 链接类型。

表 10.51 Hold_Mode 命令

命 令	OCF	命 令 参 数	返回参数
HCI_Hold_mode	0x0001	Connection_handle: Hold_mode_max_interval Hold_mode_min_interval	

表 10.52 Hold_Mode 命令描述

参 数	值	参 数 说 明
Connection_handle 2 字节(12 位有意义)	0xXXXX	用来识别链接的链接句柄。 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)
Hold_Mode_Max_Interval 2 字节	N = 0xXXXX	在保持模式基带时隙的最大可接收量。 保持时间长度 = N * 0.625ms (单基带时隙) N 的范围: 0x0001 ~ 0xFFFF, 时间: 0.625ms ~ 40.9 秒
Hold_Mode_Min_Interval 2 字节	N = 0xXXXX	在保持模式基带时隙最小可接收量 保持时间长度 = N * 0.625ms (单基带时隙) N 的范围: 0x0001 ~ 0xFFFF, 时间: 0.625ms ~ 40.9 秒

当收到了 Hold_Mode 命令时，主控制器发送命令状态事件到主机。当保持模式开始时，模式变化事件将发生。当保持模式为指定链接句柄完成时，模式变化事件甚至将再次发生。如果事件是远程蓝牙设备，则保持模式的模式变化时间信号结束是确定保持模式的终止。

注意：通过主控制器传送的无命令完成事件指出该命令已完成，代替了模式变化事件指出的该命令已完成。如果错误出现在命令状态事件发生后，那么在模式变化事件里的状态将指出其错误。

2. Sniff_Mode

呼吸模式命令用来改变链路管理器的状态，并通过指定链接句柄为呼吸模式相关的 ACL 基带链接位置。指令及其描述如表 10.53 和表 10.54 所示。链接句柄命令参数用来识别 ACL 链接链接在呼吸模式里的位置。Sniff_Max_Interval 和 Sniff_Min_Interval 命令参数用于指出在呼吸模式里申请可接受的最大和最小区间。Sniff_Min_Interval 不能大于 Sniff_Max_Interval。呼吸间隔定义了在各连续呼吸间隔之间的时间量。如果命令是成功的，主控制器应在模式改变事件的间隔参数里返回实际呼吸间隔。通过 Sniff_Attempt 命令参数作为区间指定，从单元在每个实际呼吸间隔的结束处监听。只要从单元接收分组，它将继续监听经 Sniff_Timeout 指出的附加期分组。该命令允许主控制器支持自己或几个其他的蓝牙设备的低功耗策略，而且允许设备进入查询扫描，呼叫扫描和一些其他的可能行为。

注意：链接句柄不能是 SCO 链接类型之一。

表 10.53 Sniff_Mode 命令

命 令	OCF	命 令 参 数	返回参数
HCI_Sniff_Mode	0x0003	Connection_Handle: Sniff_Max_Interval : Sniff_Min_Interval : Sniff_Attempt: Sniff_Timeout	

表 10.54 Sniff_Mode 命令描述

参 数	值	参 数 说 明
Connection_Handle 2 字节(12 位有意义)	0xXXXX	用来识别链接的链接句柄 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)
Sniff_Max_Interval 2 字节	N=0xXXXX	在各呼吸区间之间, 基带时隙的最大可接受值 长度= $N * 0.625\text{ms}$ (单基带时隙) N 的范围: 0x0001 ~ 0xFFFF, 时间: 0.625ms ~ 40.9s
Sniff_Min_Interval 2 字节	N = 0xXXXX	在各呼吸区间之间, 基带时隙的最大可接受值 长度= $N * 0.625\text{ms}$ (单基带时隙) N 的范围: 0x0001 ~ 0xFFFF, 时间: 0.625ms ~ 40.9s
Sniff_Attempt 2 字节	N = 0xXXXX	作为呼吸期望的基带时隙数 长度= $N * 0.625\text{ms}$ (单基带时隙) N 的范围: 0x0001 ~ 0xFFFF, 时间: 0.625ms ~ 40.9s
Sniff_Timeout 2 字节	N = 0xXXXX	作为呼吸超时的基带时隙数 长度= $N * 0.625\text{ms}$ (单基带时隙) N 的范围: 0x0001 ~ 0xFFFF, 时间: 0.625ms ~ 40.9s

当收到 Sniff_Mode 命令时, 主控制器发送命令状态事件到主机。对于指定的链接句柄, 当呼吸模式开始时, 将出现模式改变事件。

注意: 通过主控制器传送的无命令完成事件指出该命令已完成, 代替了模式变化事件指出的该命令已完成。如果错误出现在命令状态事件发生后, 那么在模式变化事件里的状态将指出其错误。

3. Exit_Sniff_Mode

该命令用于作为链接句柄的呼吸模式的结束处。链路管理器决定和发布适当的 LMP 命令与链接句柄有关的远程呼吸模式。指令及其描述如表 10.55 和表 10.56 所示。

注意: 链接句柄不能是 SCO 链接类型之一。

表 10.55 Exit_Sniff_Mode 命令

命 令	OCF	命 令 参 数	返回参数
HCI_Exit_Sniff_Mode	0x0004	Connection_Handle	

表 10.56 Exit_Sniff_Mode 命令描述

参 数	值	参 数 说 明
Connection_Handle 2 字节(12 位有意义)	0xXXXX	用来识别链接的链接句柄。 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)

当主控制器已收到 Exit_Sniff_Mode 命令时, 该命令的命令状态事件将出现。对于指定

的链接句柄，当呼吸模式已结束，模式改变事件出现。

注意：通过主控制器传送的无命令完成事件指出该命令已完成，代替了模式变化事件指出的该命令已完成。

4. Park_Mode

休眠模式命令用来改变链路管理器的状态，并通过指定链接句柄为休眠模式相关的ACL 基带链接位置。指令及其描述如表 10.57 和表 10.58 所示。链接句柄命令参数用来识别ACL 链接在休眠模式里的位置。链接句柄必须是 ACL 链接方式的链接句柄。信标间隔命令参数指出信标间可接受的最长间隔。然而，远程设备可申请较短的间隔。Beacon_Max_Interval 参数指出信标之间可接受的最长间隔长度。 Beacon_Min_Interval 参数指出信标之间可接受的最短间隔长度。最小信标间隔不能大于最大信标间隔。如果命令成功，主控制器将在模式变化事件中的间隔参数里返回一个实际的信标间隔。该命令允许支持主控制器自身或若干其他蓝牙设备的低功耗策略，允许设备进入查询扫描，呼叫扫描，在单个匹克网里提供支持大量的蓝牙设备和多种可行的活动。

表 10.57 Park_Mode 命令

命 令	OCF	命 令 参 数	返回参数
HCI_Park_Mode	0x0005	Connection_Handle; Beacon_Max_Interval Beacon_Min_Interval	

表 10.58 Park_Mode 命令描述

参 数	值	参 数 说 明
Connection 2 字节(12 位有意义)	0xXXXX	用来识别链接的链接句柄。 范围：0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)
Beacon_Max_Interval 2 字节	N =0xXXXX	在连续信标之间基带时隙最大可接受数。 间隔长度 = N * 0.625ms (单基带时隙) N 的范围：0x0001 ~ 0xFFFF, 时间区段：0.625ms ~ 40.9s
Beacon_Min_Interval 2 字节	N =0xXXXX	在连续信标之间基带时隙最小可接受数。 间隔长度= N * 0.625ms (单基带时隙) N 的范围：0x0001 ~ 0xFFFF, 时间区段：0.625ms ~ 40.9s

当收到 Park_Mode 命令时，主控制器发送命令状态事件给主机。对于指定的链接句柄，当休眠模式开始时，模式变化事件发生。

注意：通过主控制器传送的无命令完成事件指出该命令已完成，代替了模式变化事件指出的该命令已完成。 如果错误出现在命令状态事件发生后，那么在模式变化事件里的状态将指出其错误。

5. Exit_Park_Mode

该命令用于蓝牙设备从休眠模式切换到活动模式，其指令及其描述如表 10.59 和表 10.60 所示。当设备与在休眠模式里指定的链接句柄相关时，该命令才可以发出。链接句柄必须是 ACL 链接方式的链接句柄。

表 10.59 Exit_Park_Mode 命令

命 令	OCF	命令参数	返回参数
HCI_Exit_Park_Mode	0x0006	链接句柄	

表 10.60 Exit_Park_Mode 命令描述

	值	参 数 说 明
Connection_Handle (2 字节(12 位有意义))	0xFFFF	用来识别链接的链接句柄。 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)

当主控制器收到 Exit_Park_Mode 命令时, 该命令的状态事件发生。对于指定的链接句柄, 当休眠模式结束时, 模式变化事件发生。注意, 通过主控制器传送的无命令完成事件指出该命令已完成, 代替了模式变化事件指出的该命令已完成。

6. QoS_Setup

该命令用于指出链接句柄的服务质量参数, 其指令及其描述如表 10.61 和表 10.62 所示。链接句柄必须是 ACL 链接方式的链接句柄。这些 QoS 参数与 L2CAP QoS 参数一样。它允许链路管理器具有宿主机正在申请各个链接的所有信息。LM 将决定是否遇见 QoS 参数。主单元和从单元的蓝牙设备都能使用该命令。当设备是从单元时, 该命令将激发 LMP, 要求主单元提供一个经 LM 确定的指定 QoS 从单元。当设备是主单元时, 该命令用来申请接受指定 QoS 从设备, 该指定 QoS 通过主单元的 LM 确定。链接句柄命令参数用来识别哪个 QoS 申请链接。

表 10.61 QoS_Setup 命令

命 令	OCF	命 令 参 数	返回参数
HCI_QoS_Setup	0x0007	Connection_Handle, Flag, Service_Type, Token_Rate, Peak_Bandwidth, Latency, Delay_Variation	

当主控制器收到 QoS_Setup 命令时, 主控制器发送命令状态事件到宿主机。当链路管理器完成确立申请 QoS 参数的 LMP 消息时, 本地蓝牙设备的主控制器将发送 QoS 建立完成事件到主机, 如果有 LMP 消极应答, 事件可以也在远程方产生。然而 QoS 建立完成事件的参数值不同于初始化和远程方。QoS 建立完成事件通过在包含该命令状态的本地主控制器上返回, 而且返回的 QoS 参数描述了对于链接 QoS 的支持。

注意: 通过主控制器传送的无命令完成事件指出该命令已完成, 代替了 QoS 建立完成事件指出的该命令已完成。

表 10.62 QoS_Setup 命令描述

参 数	值	参 数 说 明
链接句柄 (12 位)	0xFFFF	用来识别 QoS 建立链接的链接句柄 范围: 0x0000~0x0EFF (0x0F00 ~ 0x0FFF 保留)
Flags (1 字节)	0x00-0xFF	保留
Service_Type (1 字节)	0x00	无传播

续表

参 数	值	参 数 说 明
	0x01	最大能力
	0x02	保证
	0x03-0xFF	保留
Token_Rate (4 字节)	0XXXXXXXX	每秒字节令牌率
Peak_Bandwidth (4 字节)	0XXXXXXXX	每秒字节带宽峰值
Latency (4 字节)	0XXXXXXXX	微秒等待时间
Delay_Variation (4 字节)	0XXXXXXXX	微秒延期变化

7. Role_Discovery

该命令用于确定蓝牙设备，该蓝牙设备的任务正在完成特殊链接句柄。该链接句柄必须是 ACL 链接方式的链接句柄。其指令及其描述如表 10.63 和表 10.64 所示。

表 10.63 Role_Discovery 命令

命 令	OCF	命 令 参 数	返 回 参 数
HCI_Role_Discovery	0x0009	Connection_Handle	Status, Connection_handle, Current_Role

表 10.64 Role_Discovery 命令描述

参 数	值	参 数 说 明
Connection_handle 2 字节(12 位有意义)	0XXXXX	用来识别链接的链接句柄 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)
Status (1 字节)	0x00	Role_Discovery 命令成功
	0x01~0xFF	Role_Discovery 命令失败
Connection_Handle 2 字节(12 位有意义)	0XXXXX	用来识别链接的链接句柄 Role_Discovery 命令被发出 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)
Current_Role (1 字节)	0x00	对于该链接句柄来说当前角色是主单元
	0x01	对于该链接句柄来说当前角色是从单元

8. Switch Role

该命令用于切换当前设备角色的蓝牙设备，该设备正在完成与另外指定的蓝牙设备特殊链接，其指令及其描述如表 10.65 和表 10.66 所示。BD_ADDR 命令参数指出哪种链接角色切换完成。角色指出本地设备完成的新角色申请。

注意：BD_ADDR 命令参数必须指出已存在链接的蓝牙设备。

表 10.65 Switch Role 命令

命 令	OCF	命 令 参 数	返 回 参 数
HCI_Switch_Role	0x000B	BD_ADDR, Role	

表 10.66 Switch Role 命令描述

参 数	值	参 数 说 明
BD_ADDR 6 字节	0XXXXXXXXXX XX	完成角色切换的链接设备的 BD_ADDR
Role (1 字节)	0x00	对于该 BD_ADDR 改变自身角色为主单元
	0x01	对于该 BD_ADDR 改变自身角色为从单元

当主控制器收到 Switch_Role 命令时, 该命令的命令状态事件发生。当角色切换完成时, 角色交换事件出现并指出角色已改变, 而且双方主机将进行通信。

9. Read_Link_Policy_Settings

该命令对于指定的链接句柄读链接策略设置, 其指令及其描述如表 10.67 和表 10.68 所示。当链路管理器或从远程设备接收申请, 或确定自身主-从角色改变或进入保持、呼吸、休眠模式时, Link_Policy_Settings 参数确定了本地链路管理器的行为。本地链路管理器可自动地接收或拒绝远程设备申请, 甚至可以自动申请自己, 取决于相应链接句柄的 Link_Policy_Settings 参数值。当确定的链接句柄的 Link_Policy_Settings 参数值改变时, 待该命令完成后, 新值仅用于远程设备或本地链路管理器自身提出的申请。链接句柄必须是 ACL 链接方式的链接句柄。通过分别启动各个模式, 主控制器能选择任何组合需要支持各种操作模式。对于 Link_Policy_Settings 参数, 通过不同活动类型的按位“或”操作, 可指定多重 LM 策略。

表 10.67 Read_Link_Policy_Settings 命令

命 令	OCF	命 令 参 数	返 回 参 数
HCI_Read_Link_Policy_Settings	0x000C	Connection_Handle	Status, Connection_Handle, Link_Policy_Settings

表 10.68 Read_Link_Policy_Settings 命令描述

参 数	值	参 数 说 明
Connection_handle 2 字节(12 位有意义)	0XXXXX	用来识别链接的链接句柄 范围: 0x0000~0x0EFF (0x0F00 ~ 0x0FFF 保留)
Status (1 字节)	0x00	Read_Link_Policy_Settings 命令成功
	0x01~0xFF	Read_Link_Policy_Settings 命令失败
Connection_Handle 2 字节(12 位有意义)	0XXXXX	用来识别链接的链接句柄 范围: 0x0000~0x0EFF (0x0F00 ~ 0x0FFF 保留)
Link_Policy_Settings (2 字节)	0x0000	禁止所有 LM 模式
	0x0001	启动主从切换
	0x0002	启动保持模式
	0x0004	启动呼吸模式
	0x0008	启动休眠模式
	0x0010- 0x8000	保留

10. write_link_policy_setting

该命令将为指定的链接句柄写链接策略设置，其指令及其描述如表 10.69 和表 10.70 所示。当链路管理器或从远程设备接收申请，或确定自身主-从角色改变或进入保持、呼吸、休眠模式时，Link_Policy_Settings 参数确定了本地链路管理器的行为。本地链路管理器可自动地接收或拒绝远程设备申请，甚至可以自动申请自己，取决于相应链接句柄的 Link_Policy_Settings 参数值。当确定的链接句柄的 Link_Policy_Settings 参数值改变时，待该命令完成后，新值仅用于远程设备或本地链路管理器自身提出的申请。链接句柄必须是 ACL 链接方式的链接句柄。通过分别启动各个模式，主控制器能选择任何组合需要支持各种操作模式。对于 Link_Policy_Settings 参数，通过不同活动类型的按位“或”操作，可指定多重 LM 策略。

表 10.69 write_link_policy_setting 命令

命 令	OCF	命令参数	返回参数
HCI_Write_Link_Policy_Settings	0x000D	Connect_Handle, Link_Policy_Settings	Status, Connection_Handle.

表 10.70 write_link_policy_setting 命令描述

	值	参数说明
Connection_Handle 2 字节(12 位有意义)	0xXXXX	用来识别链接的链接句柄 范围: 0x0000~0x0EFF (0x0F00 ~ 0x0FFF 保留)
Link_Policy_ Settings (2 字节)	0x0000	禁止所有 LM 模式
	0x0001	启动主从切换
	0x0002	启动保持模式
	0x0004	启动呼吸模式
	0x0008	启动休眠模式
	0x0010~0x8000	保留
Status (1 字节)	0x00	Write_Link_Policy_Settings 命令成功
	0x01~0xFF	Write_Link_Policy_Settings 命令失败
Connection 2 字节(12 位有意义)	0XXXXX	用来识别链接的链接句柄 范围: 0x0000~0x0EFF (0x0F00 ~ 0x0FFF 保留)

10.3.6 主控制器与基带命令

主控制器与基带命令提供识别和控制各种蓝牙硬件的能力。这些参数提供蓝牙设备的控制和主控制器、链路管理器及基带的能力。主机可利用这些命令修改本地设备的行为。对于 HCI 控制和基带命令，PGF 定义为 0x03。

表 10.71 主控制器与基带命令

命 令	命令说明
Set_Event_mask	控制主机经 HCI 产生的事件
Reset	复位蓝牙主控制器、链路管理器和无线设备

命 令	命 令 说 明
Set_event_Filter	经主机指定不同事件过滤器。对于不同类型或同类型的过滤器，主机可多次发送该命令申请多种条件
Flush	对于指定的链接句柄，放弃所有作为当前待传输数据，甚至当前是属于多个在主控制器里的 L2CAP 分组的数据块
Read_Pin_Type	主机读出由指定主机支持的可变 PIN 还是固定 PIN 值
Write_Pin_Type	主机写入由指定主机支持的可变 PIN 还是固定 PIN 值
Creat_New_Unit_Key	创建新单元字
Read_Store_Link_Key	提供读出存放在蓝牙主控制器里的单个或多个链接字的能力
Write_Store_Link_Key	提供写入存放在蓝牙主控制器里的单个或多个链接字的能力
Delete_Store_Link_Key	提供删除存放在蓝牙主控制器里的单个或多个链接字的能力
Change_Local_Name	提供修改蓝牙设备的用户友好名的能力
Read_Local_Name	提供读存储的蓝牙设备用户友好名的能力
Read_Connection_Accept_Timeout	读链接识别超时结构参数值，在指定区段出现后，该命令允许蓝牙硬件自动拒绝链接申请和拒绝新的链接
Write_Connection_Accept_Timeout	写链接识别超时结构参数值，在指定区段出现后，该命令允许蓝牙硬件自动拒绝链接申请和拒绝新的链接
Read_Page_Timeout	读呼叫响应超时结构参数，在本地设备返回链接失败前，该值是允许蓝牙硬件定义等待远程设备链接申请的时间量
Write_Page_Timeout	写呼叫响应超时结构参数，在本地设备返回链接失败前，该值是允许蓝牙硬件定义等待远程设备链接申请的时间量
Read_Scan_Enable	读出扫描允许结构参数值，该值的控制不管蓝牙设备是处于呼叫期望的周期性扫描或是其他蓝牙设备的查询申请。
Write_Scan_Enable	写入扫描允许结构参数值，该值的控制不管蓝牙设备是处于呼叫期望的周期性扫描或是其他蓝牙设备的查询申请
Read_Page_Scan_Activity	读出呼叫扫描间隔和呼叫扫描区间结构参数。呼叫扫描间隔定义为在连续呼叫扫描之间的时间量。呼叫扫描区间定义为在呼叫扫描的期间
Write_Page_Scan_Activity	写入呼叫扫描间隔和呼叫扫描区间结构参数。呼叫扫描间隔定义为在连续呼叫扫描之间的时间量。呼叫扫描区间定义为在呼叫扫描的期间
Read_Inquiry_Scan_Activity	读出查询扫描间隔和查询扫描区间的结构参数。查询扫描间隔定义为在连续查询扫描之间的时间量。查询扫描区间定义为在查询扫描的期间
Write_Inquiry_Scan_Activity	写入查询扫描间隔和查询扫描区间的结构参数。查询扫描间隔定义为在连续查询扫描之间的时间量。查询扫描区间定义为在查询扫描的期间
Read_Authentication_Enable	读出鉴权允许参数值。该值控制蓝牙设备使用其他蓝牙设备各种链接的鉴权申请
Write_Authentication_Enable	写入鉴权允许参数值。该值控制蓝牙设备使用其他蓝牙设备各种链接的鉴权申请
Read_Encryption_Mode	读出加密模型参数值。该值控制蓝牙设备使用其他蓝牙设备各种链接的加密申请
Write_Encryption_Mode	写入加密模型参数值。该值控制蓝牙设备使用其他蓝牙设备各种链接的加密申请
Read_Class_Of_Device	读出设备类参数值。用来指出别的设备能力

续表

命 令	命 令 说 明
Write_Class_Of_Device	写入设备类参数值。用来指出别的设备能力
Read_Voice_Setting	读出语音设置参数值。控制所有语音链接的各种设置
Write_Voice_Setting	写入语音设置参数值。控制所有语音链接的各种设置
Read_Automatic_Flush_Timeout	对指定的链接句柄, 读出刷新超时参数值
Write_Automatic_Flush_Timeout	对指定的链接句柄, 写入刷新超时参数值
Read_Num_Broadcast_Retransmissions	读出设备的广播重发次数值。广播分组不需确认而且不可靠。该参数通过多次重传广播消息来提高广播消息的可靠性
Write_Num_Broadcast_Retransmissions	写入设备的广播重发次数值。广播分组不需确认而且不可靠。该参数通过多次重传广播消息来提高广播消息的可靠性
Read_Hold_Mode_Activity	读出主控模型活动参数值。当设备为主控模型时, 该值用来确定设备的活动做什么
Write_Hold_Mode_Activity	写入主控模型活动参数值。当设备为主控模型时, 该值用来确定设备的活动做什么
Read_Transmit_Power_Level	对于指定的链接句柄, 读出传输功率电平参数值
Read_SCO_Flow_Control_Enable	读出 SCO 流控制允许设置。通过使用该设置, 主控制器能决定是否主控制器对于 SCO 链接句柄送出完成分组事件的数
Write_SCO_Flow_Control_Enable	写入 SCO 流控制允许设置。通过使用该设置, 主控制器能决定是否主控制器对于 SCO 链接句柄送出完成分组事件的数
Set_Host_Controller_To_Host_Control	用于主控制器直接打开或关闭主控制器到主机的流控制
Host_Buffer_Size	通过主机修改主控制器有关 ACL 和 SCO 数据缓冲区的大小。主控制器将从主控制器到主机分段传送数据, 所以包含在 HCI 数据分组里的数据将不会超出这个区段
Host_Number_Of_Completed_Packets	当主机对于任何链接句柄准备接收较多的 HCI 分组时, 该命令用于通过主机指出主控制器。
Read_Link_Supervision_Timeout	对于设备读出链接管理超时参数。该参数通过主或从蓝牙设备到损失监控链接使用。
Write_Link_Supervision_Timeout	对于设备写入链接管理超时参数。该参数通过主或从蓝牙设备到损失监控链接使用。
Read_Number_Of_Supported_ICA	读出在查询扫描期间本地蓝牙设备正同时扫描的查询识别码 (ICA) 数的值。
Read_Current_ICA_LAP	读出用于创建在查询扫描期间本地蓝牙设备正同时扫描的查询识别码 (ICA) 的 LAP
Write_Current_ICA_LAP	写入用于创建在查询扫描期间本地蓝牙设备正同时扫描的查询识别码 (ICA) 的 LAP
Read_Page_Scan_Period_Mode	用于读出本地蓝牙设备的强制呼叫扫描区间模式。
Write_Page_Scan_Period_Mode	用于写入本地蓝牙设备的强制呼叫扫描区间模式。
Read_Page_Scan_Mode	用于读出本地蓝牙设备的默认呼叫扫描模式。
Write_Page_Scan_Period_Mode	用于写入本地蓝牙设备的默认呼叫扫描模式。

1. Set_Event_Mask

该命令通过主机的 HCI 来控制哪个事件产生，其指令及其描述如表 10.72 和表 10.73 所示。如果在 Event_Mask 位置成 1，则与该位有关的事件产生。主机必须处理由蓝牙设备发生的每个事件。事件屏蔽允许主机控制中断数量，但命令完成事件、命令状态事件和完成分组事件数不能屏蔽。这些事件总是出现。事件屏蔽是整个事件指定的屏蔽位。

表 10.72 Set_Event_Mask 命令

命 令	OCF	命令参数	返回参数
HCI_Set_Event_Mask	0x0001	Event_Mask	状态

表 10.73 Set_Event_Mask 命令描述

参 数	值	参 数 说 明
Event_Mask (8 字节)	0x0000000000000000	无事件指定
	0x0000000000000001	查询完成事件
	0x0000000000000002	查询结果事件
	0x0000000000000004	链接完成事件
	0x0000000000000008	链接申请事件
	0x0000000000000010	断开完成事件
	0x0000000000000020	鉴权完成事件
	0x0000000000000040	远程名申请完成事件
	0x0000000000000080	加密变化事件
	0x0000000000000100	变化链接链接字完成事件
	0x0000000000000200	主单元链接字完成事件
	0x0000000000000400	读远程支持特征完成事件
	0x0000000000000800	读远程版本信息完成事件
	0x0000000000001000	QoS 建立完成事件
	0x0000000000002000	命令完成事件
	0x0000000000004000	命令状态事件
	0x0000000000008000	硬件错误事件
	0x0000000000010000	刷新发生事件
	0x0000000000020000	角色变化事件
	0x0000000000040000	完成分组事件的数
	0x0000000000080000	模式变化事件
	0x0000000000100000	返回链接字事件
	0x0000000000200000	PIN 码申请事件
	0x0000000000400000	链接字申请事件
	0x0000000000800000	链接字注释事件
	0x0000000001000000	反馈命令事件
	0x0000000002000000	数据缓冲区溢出事件
	0x0000000004000000	最大时隙变化事件
	0x0000000008000000	读时钟补偿完成事件
	0x0000000010000000	链接分组类型改变事件

续表

参 数	值	参 数 说 明
	0x00000000~0x00000001	QoS 违例事件
	0x00000000~0x00000002	呼叫扫描模式改变事件
	0x00000000~0x00000003	呼叫扫描重复模式变化事件
	0x00000000~0x00000004-0x80000000~0x80000000	保留
	0x00000000~0xFFFFFFFF	缺省(所有事件允许)
Status (1 字节)	0x00	Set_Event_Mask 命令成功
	0x01~0xFF	Set_Event_Mask 命令失败

2. Reset

Reset 命令复位蓝牙主控制器、链路管理器和无线设备，而且放弃当前的操作选择，同时也放弃分组排队。在复位完成后，蓝牙设备进入待机模式。指令及其描述如表 10.74 和表 10.75 所示。

表 10.74 Reset 命令

命 令	OCF	命令参数	返回参数
HCI Reset	0x0003		Status

表 10.75 Reset 命令描述

参 数	值	参 数 说 明
Status (1 字节)	0x00	复位命令成功，收到且将执行
	0x01~0xFF	复位命令失败

3. Set_Event_Filter

该命令用来通过主机指定不同的事件过滤器，其指令及其描述如表 10.76 和表 10.77 所示。对于同类事件过滤器或不同类事件过滤器，主机可多次发送各种链接申请。事件过滤器通过主机指定有关的对象，这些对象允许主控制器只发送与主机有关的事件。仅有一部分事件具有事件过滤器。默认（开机和复位后）方式无过滤器设置，而且自动识别标志关闭（引入的链接不是自动识别）。每次从主机发送该命令时，都加入事件过滤器，Filter_Condition_Type 不等于 0x00（旧的事件过滤器不会重写）。为清除所有事件过滤器，使用 Filter_Type = 0x00，Auto_Accept_Flag(自动识别标志)设置成关闭。

表 10.76 Set_Event_Filter 命令

命 令	OCF	命令参数	返回参数
HCI_Set_Event_Filter	0x0005	Filter_Type, Filter_Condition_Type, condition	Status

清除事件过滤器只能是确定的 Filter_Type，使用 Filter_Condition_Type = 0x00。查询结果过滤器允许主控制器滤出查询结果事件。如果查询结果事件遇到由主机设置的指定条件之一，查询结果过滤器允许主机指定的主控制器只发送查询结果到主机。对于查询结果过滤器，主机可指定一个或多个下列过滤器链接类型：新设备响应查询过程；使用指定设备类的设备

响应查询过程；使用指定 **BD_ADDR** 的设备响应查询过程。

查询结果过滤器使用查询和循环查询命令合取。如果事件遇到由主机指定的设置之一，则链接建立过滤器允许主机指出主控制器只发送链接完成和链接申请事件。作为链接建立过滤器，主机可指定一个或多个下列过滤器链接类型：允许与所有设备链接；允许与指定设备类的设备链接；允许用指定 **BD_ADDR** 的设备链接。

对于这些链接类型，自动识别标志参数允许主机指出链接实现时应作什么样的动作。自动识别标志参数允许主机指出引入链接是否是自动识别（在自动识别情况下，当链接完成时，主控制器送出链接完成事件给主机，或是否由主机作出抉择，在这种情况下，主控制器送出链接申请事件给主机，在链接上引出抉择）。

如果本地设备处于呼叫扫描的过程，而且是在由主机设置的链接上的另一设备呼入，同时自动识别标志针对该设备已处于关闭状态，则链接申请事件通过主控制器发送到主机。在主机已响应引入链接期望后，链接完成事件随后送出。以同样的例子，如果自动识别标志是开启的，则链接完成事件通过主控制器送给主机（该情况下，无链接申请事件发送）。主控制器在随机存储器里存储这些过滤器，直到主机使用 **Set_Event_Filter** 命令清除事件过滤器，或直到复位命令发出。主控制器存储的事件过滤器的次数是独立实现的。如果主机希望设立多于主控制器能存储的过滤器，主控制器将返回“存储器满”的错误码，同时过滤器不能再存入。

注意，在链接处于自动识别状态时，链接字申请事件、PIN 码申请事件和链接字标志事件通过主控制器在链接完成事件发送前实现发送。如果在事件过滤器之间有矛盾，则后发送的事件过滤器将复盖先发送的事件过滤器。

表 10.77 **Set_Event_Filter** 命令描述

参 数	值	参 数 说 明
Filter_Type (1 字节)	0x00	清除的所有滤波器。（注意：在这种情况下，Filter_Condition_Type 和链接参数将不给出，其长度为 0 字节。Filter_Type 仅是参数。）
	0x01	查询结果
	0x02	链接设置
	0x03~0xFF	保留
Inquiry_Result_ Filter_Condition_ Type (1 字节)	0x00	响应查询过程的新设备
	0x01	响应查询过程指定设备类的设备
	0x02	响应查询过程指定 BD_ADDR 的设备
	0x03~0xFF	保留
Connection_Setup_ _Filter_Condition_ _Type (1 字节)	0x00	允许所有的设备链接
	0x01	允许指定设备类的设备链接
	0x02	允许指定 BD_ADDR 的设备链接
	0x03~0xFF	保留
Status (1 字节)	0x00	Set_Event_Filter 命令成功
	0x01	Set_Event_Filter 命令失败

各过滤器链接类型定义了查询结果过滤器和链接设置过滤器，要求的无链接参数或多

个链接参数取决于过滤器链接类及过滤器类，如图 10.78 所示。

当主控制器允许事件过滤时，该命令完成事件发生。当满足条件之一时，指定事件发生。

表 10.78 Set_Event_Filter 命令的状态参数描述

条 件		参 数	值	参 数 说 明
Inquiry_	0x00	Condition		没使用条件参数
Result_	0x01	Class_of_Device	0x000000	缺省,返回所有的设备
Filter_		3 字节	0xFFFFFFFF	相关设备类
Condition_		Class_of_Device_	0xFFFFFFFF	屏蔽位决定设备参数类的哪些位是无关项。屏蔽零值位指出设备类的哪些位是无关项
Type		Mask 3 字节		
	0x02	BD_ADDR	0xFFFFFFFF XXXXXX	相关设备的 BD_ADDR
		6 字节		
Connection _Setup_ Filter_ Condition_ Type	0x00	Auto_Accept_Flag	0x01	非自动识别链接
			0x02	自动识别链接
			0x03~0xFF	保留
	0x01	Class_of_Device	0x000000	默认,返回所有设备
			0xFFFFFFFF	相关设备类
		Class_of_Device_	0xFFFFFFFF	屏蔽位决定设备参数类的哪些位是无关项。屏蔽零值位指出设备类的哪些位是无关项
		Mask 3 字节		
		Auto_Accept_Flag	0x01	非自动识别链接
			0x02	自动识别链接
	0x03~0xFF		保留	
	0x02	BD_ADDR	0xFFFFFFFF XXXXXX	相关设备的 BD_ADDR
		Auto_Accept_Flag	0x01	非自动识别链接
			0x02	自动识别链接
0x03~0xFF			保留	
		1 字节		

4. Flush

刷新命令用来放弃对于主控制器所指链接句柄当前待传输的数据，甚至当前是属于大于主控制器里 L2CAP 分组的数据块。在此之后，发送到相同链接句柄主控制器的所有数据通过主控制器放弃，直到使用 Packet_Boundary_Flag (0x02) 开始的 HCI 数据分组收到。其指令及其描述如表 10.79 和表 10.80 所示。

当此过程发生时，新的传输期望构成。在主控制器里的当前待传所有数据刷新前，该命令将允许高层软件控制链接句柄基带期望重传基带分组长度。刷新命令仅用于 ACL 链接方式。除刷新命令外，在指定刷新定时器终止后，自动刷新定时器能自动地刷新当前正发送的 L2CAP。

表 10.79 Flush 命令

命 令	OCF	命令参数	返回参数
HCI_Flush	0x0008	Connection_handle	Status, connection_handle

表 10.80 Flush 命令描述

参 数	值	参 数 说 明
Connection 2 字节 (12 位有意义)	0xXXXX	用来识别刷新链接的链接句柄 范围: 0x0000~0x0EFF(0x0F00~0x0FFF 保留)
Status (1 字节)	0x00	刷新成功的命令
	0x01~0xFF	刷新失败的命令
Connection 2 字节 (12 位有意义)	0xXXXX	用来识别发出刷新命令的链接句柄 范围: 0x0000~0x0EFF(0x0F00~0x0FFF 保留)

一旦刷新命令执行, 刷新发生事件发生。刷新发生事件由自动刷新引起或通过主机发出刷新命令引起。当刷新命令完成时, 命令完成事件产生, 并指出引起刷新的主机。

5. Read_PIN_Type

该命令用于主机读出链路管理器假设由主机支持可变的 PIN 码仅为固定 PIN 码。蓝牙硬件在配对期间使用 PIN 类信息。其指令及其描述如表 10.81 和表 10.82 所示。

表 10.81 Read_PIN_Type 命令

命 令	OCF	命令参数	返回参数
HCI_Read_PIN_Type	0x0009		Status, pin_type

表 10.82 Read_PIN_Type 命令描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_PIN_Type 命令成功
	0x01~0xFF	Read_PIN_Type 命令失败
PIN_Type 1 字节	0x00	可变 pin
	0x01	固定 Pin

6. Write_PIN_Type

该命令用于主机写入链路管理器假设主机支持可变的 PIN 码仅为固定 PIN 码。蓝牙硬件在配对期间使用 PIN 类信息。其指令及其描述如表 10.83 和表 10.84 所示。

表 10.83 Write_PIN_Type 命令

命 令	OCF	命令参数	返回参数
HCI_Write_PIN_Type	0x000a	Pin_type	Status

表 10.84 Write_PIN_Type 命令描述

	值	参数说明
PIN_Type 1 字节	0x00	可变 pin
	0x01	固定 Pin

7. Create_New_Unit_Key

该命令用来创建新单元字。蓝牙硬件产生一个用于产生新单元字的随机启动源。所有新的链接将使用新单元字，但旧单元字仍被用于所有的当前链接。其指令及其描述如表 10.85 和表 10.86 所示。该命令对不使用单元字的设备没有任何影响。

表 10.85 Create_New_Unit_Key 命令

命 令	OCF	命令参数	返回参数
HCI_Create_New_Unit_Key	0x000b		Status

表 10.86 Create_New_Unit_Key 命令描述

参 数	值	参数说明
Status 1 字节	0x00	创建新单元字命令成功
	0x01~0xFF	创建新单元字命令失败

8. Read_Stored_Link_Key

该命令提供读出一个或多个存储在蓝牙主控制器里的链接字，其指令及其描述如表 10.87 和表 10.88 所示。对于其他蓝牙设备的蓝牙主控制器只能提供存储有限的链接字。链接字为两个蓝牙设备共享，而且用于两设备间安全处理。当需要时，主机可增加存储量，为再装入蓝牙主控制器的链接字提供存储需要。Read_All_Flag 参数来指出是否整个链接字将返回。如果 Read_All_Flag 指出所有链接字将返回，那么命令参数的 BD_ADDR 必须忽略。BD_ADDR 命令参数用来识别读出哪个链接字。存储链接字通过一个或多个返回链接字事件来返回。

表 10.87 Read_Stored_Link_Key 命令

命 令	OCF	命令参数	返回参数
HCI_Read_Stored_Link_Key	0x000d	Bd_addr Read_all_flag	Status, Max_num_keys Num_keys_read

表 10.88 Read_Stored_Link_Key 命令描述

参 数	值	参数说明
BD_ADDR 6 字节	0XXXXXXXXXXXXX	读存储链接字的 BD_ADDR
Read_All_Flag 1 字节	0x00	返回指定 BD_ADDR 的链接字
	0x01	返回所有存储链接字
	0x02~0xFF	保留
Status 1 字节	0x00	Read_Stored_Link_Key 命令成功
	0x01~0xFF	Read_Stored_Link_Key 命令失败
Max_Num_Key 2 字节	0XXXXX	主控制器能存储链接字的最大量 范围：0x0000 ~ 0xFFFF
Num_Keys_Read 2 字节	0XXXXX	读链接字数，范围：0x0000 ~ 0xFFFF

命令发出后，没有或多个返回链接字事件发生。当没有链接字存储时，无返回链接字事件返回。当有链接字存储时，指定在各返回链接字事件里的链接字返回数执行。

9. write_store_link_key

该命令提供写入一个或多个存储在蓝牙主控制器里的链接字，其指令及其描述如表 10.89 和表 10.90 所示。对于其他蓝牙设备的蓝牙主控制器只能存储有限的链接字。如果在蓝牙主控制器里没有有效的附加空间，就不能存储附加的链接字。如空间有限，则所有链接字要存储在有限空间里是不合适的。正确的链接字表序列可确定哪些链接字被存储。首先在表开始处的链接字被存储。

Num_Keys_Write 参数返回成功存储的链接字数。如果没有附加的存储空间，在任何附加链接字存储前，主机必须删除一个以上的已存储链接字。

链接字替换算法通过主机完成，而不是主控制器来执行。链接字替换算法由主机执行，不由主机控制器执行。

链接字为两个蓝牙设备共享，而且用于两设备间安全处理。当需要时，主机可增加存储量，为再装入蓝牙主控制器的链接字提供存储需要。

表 10.89 write_store_link_key 命令

命 令	OCF	命 令 参 数	返 回 参 数
HCI_Write_Stored_Link_Key	0X0011	Num_Keys_To_Write BD_ADDR: Link_Key	Status Num_Keys_Written

表 10.90 write_store_link_key 命令描述

参 数	值	参 数 说 明
Num_Keys_To_Write 1 字节	0xXX	写入链接字数
BD_ADDR 6 字节	0XXXXXXXXX XXXX	与链接字有关的 BD_ADDR
Link_Key 16 字节	0XXXXXXXXX XXXXXXXXXX XXXXXXXX	与 BD_ADDR 有关的链接字
Status 1 字节	0x00	Write_Stored_Link_Key 命令成功
	0x01-0xFF	Write_Stored_Link_Key 命令失败
Num_Keys_Written 1 字节	0xXX	成功写入的链接字数，范围：0x00~0xFF

10. delete_store_link_key

该命令提供删除一个以上的存储在蓝牙主控制器里的链接字，其指令及其描述如表 10.91 和表 10.92 所示。蓝牙主控制器只能为其他蓝牙设备存储有限的链接字。链接字为两个蓝牙设备共享，而且用于两设备间安全处理。Delete_all_flag 参数用来指出是否所有的存储链接字都要删除。如果 Delete_All_Flag 指出所有的链接字被删除，那么 BD_ADDR 命令参数必须忽略该命令提供在设备之间达成安全协议能力。BD_ADDR 命令参数用来识别删除哪个链接字。如果链接字是用于当前的链接，当所有的链接断开时，则链接字被删除。

表 10.91 delete_store_link_key 命令

命 令	OCF	命 令 参 数	返 回 参 数
HCI_Delete_Stored_Link_Key	0x0012	Bd_addr Delete_all_flag	Status Num_keys_deleted

表 10.92 delete_store_link_key 命令描述

参 数	值	参 数 说 明
BD_ADDR (6 字节)	0XXXXXXXXX XXXX	被删除链接字的 BD_ADDR
Delete_All_Flag (1 字节)	0x00	仅删除指定 BD_ADDR 链接字
	0x01	删除所有存储链接字
	0x02~0xFF	保留
Status (1 字节)	0x00	Delete_Stored_Link_Key 命令成功
	0x01~0xFF	Delete_Stored_Link_Key 命令失败
Num_Keys_Deleted (2 字节)	0XXXXX	删除的链接字数

11. Change_Local_Name

该命令提供蓝牙设备用户友好名的修改能力，其指令及其描述如表 10.93 和表 10.94 所示。蓝牙设备可发送申请得到另外蓝牙设备的用户友好名。用户友好名提供用户把蓝牙设备与另外蓝牙设备区分开来的能力。名命令参数编码是长度可达 248 个字节的 UTF-8。如果 UTF-8 编码不到 248 个字节，名命令参数应是空终止(0x00)。名参数以名第一字节开始发送。传输多字节参数小 Endian 序列格式是一个例外。

表 10.93 Change_Local_Name 命令

命 令	OCF	命 令 参 数	返 回 参 数
HCI_Change_Local_Name	0x0013	Name	Status

表 10.94 Change_Local_Name 命令描述

参 数	值	参 数 说 明
Name 248 字节		UTF-8 用户友好编码描述了设备名称 UTF-8 编码名长度可达的 248 个字节。如果它 小于 248 个字节，用空字节(0x00)指出结束
		空终止零长度，默认
Status (1 字节)	0x00	Change_Local_Name 命令成功
	0x01~0xFF	Change_Local_Name 命令失败

12. Read_Local_Name

该命令提供读蓝牙设备存储用户友好名能力，其指令及其描述如表 10.95 和表 10.96 所示。用户友好名提供用户把蓝牙设备与另外蓝牙设备区分开来的能力。名返回参数编码是长度可达 248 个字节的 UTF-8。如果 UTF-8 编码不到 248 个字节，名返回参数应是空终止(0x00)。注意，名参数以名第一字节开始发送。传输多字节参数小 Endian 序列格式是一个

例外。

表 10.95 Read_Local_Name 命令

命 令	OCF	命令参数	返 回 参 数
HCI_Read_Local_Name	0x0014		Status; name

表 10.96 Read_Local_Name 命令描述

	值	参 数 说 明
Status	0x00	Read_Local_Name 命令成功
1 字节	0x01~0xFF	Read_Local_Name 命令失败
Name 248 字节		UTF-8 用户友好编码描述了设备名称。 UTF-8 编码名长度可达的 248 个字节。如果它小于 248 个字节，用空字节 (0x00)指出结束

13. Read_Connection_Accept_Timeout

该命令读出 Connection_Accept_Timeout 结构参数值，其指令及其描述如表 10.97 和表 10.98 所示。在指定周期已出现并且新的链接还没识别时，Connection_Accept_Timeout 参数允许蓝牙硬件自动地拒绝链接申请。该参数定义为从当主控制器发出链接申请事件起到主控制器自动拒绝引入链接止的时间区间。

表 10.97 Read_Connection_Accept_Timeout 命令

命 令	OCF	命令参数	返 回 参 数
HCI_Read_Connection_Accept_Timeout	0x0015		Status; conn_accept_timeout

表 10.98 Read_Connection_Accept_Timeout 命令描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_Connection_Accept_Timeout 命令成功
	0x01~0xFF	Read_Connection_Accept_Timeout 命令失败
Conn_Accept_Timeout 2 字节	N=0xFFFF	在基带时隙的数里链接识别超时测量 Interval Length = N * 0.625 msec(单基带时隙) 范围: 0.625ms ~ 29s

14. Write_Connection_Accept_Timeout

该命令写入 Connection_Accept_Timeout 结构参数值，其指令及其描述如表 10.99 和表 10.100 所示。在指定周期已出现并且新的链接还没识别后，Connection_Accept_Timeout 参数允许蓝牙硬件自动地拒绝链接申请。该参数定义为从当主控制器发出链接申请事件起到主控制器自动拒绝引入链接止的时间区间。

表 10.99 Write_Connection_Accept_Timeout 命令

命 令	OCF	命令参数	返回参数
HCI_Write_Connection_Accept_Timeout	0x0016		Status

表 10.100 Write_Connection_Accept_Timeout 命令描述

参 数	值	参 数 说 明
Conn_Accept_Timeout 2 字节	N = 0xXXXX	在基带时隙的数里链接识别超时测量 Interval Length = $N * 0.625 \text{ msec}$ (单基带时隙) N 的范围: 0x0001 ~ 0xb540 时间范围: 0.625ms ~ 29s 默认: N = 0x1FA0 , 时间=5s
Status 1 字节	0x00	Write_Connection_Accept_Timeout 命令成功
	0x01~0xFF	Write_Connection_Accept_Timeout 命令失败

15. Read_Page_Timeout

该命令读出 Page_Timeout 结构参数值, 其指令及其描述如表 10.101 和表 10.102 所示。Page_Timeout 结构参数定义本地链路管理器等待基带呼叫响应的最大时间, 该基带呼叫响应来自于本地初始化链接期望的远程设备。如果该时间终止和远程设备没在基带级上响应呼叫, 链接期望将认为是失败。

表 10.101 Read_Page_Timeout 命令

命 令	OCF	命令参数	返回参数
HCI_Read_Page_Timeout	0x0017		Status, Page_timeout

表 10.102 Read_Page_Timeout 命令描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_Page_Timeout 命令成功
	0x01~0xFF	Read_Page_Timeout 命令失败
Page_Timeout 2 字节	N = 0xXXXX	在基带时隙数里的呼叫超时测量 间隔长度 = $N * 0.625 \text{ ms}$ (单基带时隙) N 的范围: 0x0001 ~ 0xFFFF 时间范围: 0.625ms ~ 40.9s

16. Write_Page_Timeout

该命令写入 Page_Timeout 结构参数值, 其指令及其描述如表 10.103 和表 10.104 所示。Page_Timeout 结构参数定义本地链路管理器等待基带呼叫响应的最大时间, 该基带呼叫响应来自于本地初始化链接期望的远程设备。如果该时间终止和远程设备没在基带级上响应呼叫, 链接期望将认为是失败。

表 10.103 Write_Page_Timeout 命令

命 令	OCF	命 令 参 数	返回参数
HCI_Write_Page_Timeout	0x0018	Page_Timeout	Status

表 10.104 Write_Page_Timeout 命令描述

参 数	值	参 数 说 明
Page_Timeout 2 字节	0	非法呼叫超时 必须大于 0
	N = 0xXXXX	在基带时隙数里的呼叫超时测量 间隔长度 = $N * 0.625\text{ms}$ N 的范围: 0x0001 ~ 0xFFFF, 时间范围: 0.625ms ~ 40.9s 默认: N = 0x2000, 时间 = 5.12s
Status 1 字节	0x00	Write_Page_Timeout 命令成功
	0x01~0xFF	Write_Page_Timeout 命令失败

17. Read_Scan_Enable

该命令读出 Scan_Enable 参数值, 其指令及其描述如表 10.105 和表 10.106 所示。Scan_Enable 参数控制蓝牙设备是否周期性地扫描其他蓝牙设备的呼叫期望或查询申请。

如果 Page_Scan 允许, 则设备将基于 Page_Scan_Interval 和 Page_Scan_Window 参数进入呼叫扫描模式; 如果 Inquiry_Scan 允许, 则设备将基于 Inquiry_Scan_Interval 和 Inquiry_Scan_window 参数进入查询扫描模式。

表 10.105 Read_Scan_Enable 命令

命 令	OCF	命令参数	返回参数
HCI_Read_Scan_Enable	0x0019		Status; Scan_enable

表 10.106 Read_Scan_Enable 命令描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_Scan_Enable 命令成功了
	0x01~0xFF	Read_Scan_Enable 命令失败
Scan_Enable 1 字节	0x00	无扫描允许
	0x01	查询扫描允许; 呼叫扫描禁止
	0x02	查询扫描禁止; 呼叫扫描允许
	0x03	查询扫描允许; 呼叫扫描允许

18. Write_Scan_Enable

该命令写入 Scan_Enable 参数值, 其指令及其描述如表 10.107 和表 10.108 所示。Scan_Enable 参数控制蓝牙设备是否周期性地扫描其他蓝牙设备的呼叫期望或查询申请。

如果 Page_Scan 允许, 则设备将基于 Page_Scan_Interval 和 Page_Scan_Window 参数进入呼叫扫描模式; 如果 Inquiry_Scan 允许, 则设备将基于 Inquiry_Scan_Interval 和 Inquiry_Scan_window 参数进入查询扫描模式。

表 10.107 Write_Scan_Enable 命令

命 令	OCF	命令参数	返回参数
HCI_Write_Scan_Enable	0x001A	Scan_Enable	status

表 10.108 Write_Scan_Enable 命令描述

参 数	值	参 数 说 明
Scan_Enable 1 字节	0x0	无扫描允许，默认
	0x01	查询扫描允许；呼叫扫描禁止
	0x02	查询扫描禁止；呼叫扫描允许
	0x03	查询扫描允许；呼叫扫描允许
Status 1 字节	0x00	Write_Scan_Enable 命令成功
	0x01~0xFF	Write_Scan_Enable 命令失败

19. Read_Page_Scan_Activity

该命令读出 Page_Scan_Activity 结构参数值，其指令及其描述如表 10.109 和表 10.110 所示。Page_Scan_Interval 结构参数定义为连续呼叫扫描之间的时间量。该时间间隔被定义为主控制器上次呼叫扫描的开始处到下次呼叫扫描的开始。Page_Scan_Window 结构参数定义为呼叫扫描持续时间。Page_Scan_Window 只能小于或等于 Page_Scan_Interval。当 Page_Scan 允许时，仅执行呼叫扫描。改变 Page_Scan_Interval 能改变本地的 Page_Scan_Repetition_Mode。

表 10.109 Read_Page_Scan_Activity 命令

命 令	OCF	命令参数	返 回 参 数
HCI_Read_Page_Scan_Activity	0x001B		Status, Page_Scan_Interval, Page_Scan_Window

表 10.110 Read_Page_Scan_Activity 命令描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_Page_Scan_Activity 命令成功
	0x01~0xFF	Read_Page_Scan_Activity 命令失败
Page_Scan_Interval 2 字节	N=0xXXXX	长度：2 字节，范围：0x0012~0x1000 时间= N * 0.625ms，范围：11.25ms ~ 2560ms
Page_Scan_Window 2 字节	N=0xXXXX	长度：2 字节，范围：0x0012 ~ 0x1000 时间= N * 0.625ms，范围：11.25ms ~ 2560ms

20. Write_Page_Scan_Activity

该命令写入 Page_Scan_Activity 结构参数值，其指令及其描述如表 10.111 和表 10.112 所示。Page_Scan_Interval 结构参数定义为连续呼叫扫描之间的时间量。该时间间隔被定义为主控制器上次呼叫扫描的开始处到下次呼叫扫描的开始。Page_Scan_Window 结构参数定义为呼叫扫描持续时间。Page_Scan_Window 只能小于或等于 Page_Scan_Interval。当

Page_Scan 允许时，仅执行呼叫扫描。改变 Page_Scan _Interval 能改变本地的 Page_Scan_ Repetition_Mode。

表 10.111 Write_Page_Scan_Activity 命令

命 令	OCF	命 令 参 数	返回参数
HCI_Write_Page_Scan_Activity	0x001C	Page_Scan_Interval Page_Scan_Window	Status

表 10.112 Write_Page_Scan_Activity 命令描述

参 数	值	参 数 说 明
Page_Scan_Interval 2 字节	N=0xXXXX	长度：2 字节，范围：0x0012~ 0x1000， 时间 = $N * 0.625ms$ ，范围：11.25ms ~ 2560ms， 默认：N = 0x0800，时间 = 1.28s
Page_Scan_Window 2 字节	N=0xXXXX	长度：2 字节，范围：0x0012 ~ 0x1000 时间 = $N * 0.625ms$ ，范围：11.25ms ~ 2560ms 默认：N = 0x0012，时间 = 11.25ms
Status 1 字节	0x00	Write_Page_Scan_Activity 命令成功
	0x01~0xFF	Write_Page_Scan_Activity 命令失败

21. Read_Inquiry_Scan_Activity

该命令读出 Inquiry_Scan_Activity 结构参数值，其指令及其描述如表 10.113 和表 10.114 所示。Inquiry_Scan _Interval 结构参数定义为连续查询扫描之间的时间量。该时间间隔被定义为主控制器上次查询扫描的开始处到下次查询扫描的开始。Inquiry_Scan_Window 结构参数定义为呼叫查询持续时间。Inquiry_Scan_Window 只能小于或等于 Inquiry_Scan_Interval。当 Inquiry_Scan 允许时，仅执行查询扫描。

表 10.113 Read_Inquiry_Scan_Activity 命令

命 令	OCF	命令参数	返 回 参 数
HCI_Read_Inquiry_Scan_Activity	0x001D		Status, Inquiry_Scan_Interval, Inquiry_Scan_Window

表 10.114 Read_Inquiry_Scan_Activity 命令描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_Inquiry_Scan_Activity 命令成功
	0x01	Read_Inquiry_Scan_Activity 命令失败
Inquiry_Scan_Interval 2 字节	N=0xXXXX	长度：2 字节，范围：0x0012 ~ 0x1000 时间 = $N * 0.625ms$ ，范围：11.25ms ~ 2560ms
Inquiry_Scan_Window 2 字节	N=0xXXXX	长度：2 字节，范围：0x0012 ~ 0x1000 时间 = $N * 0.625ms$ ，范围：11.25ms~2560ms

22. Write_Inquiry_Scan_Activity

该命令写入 Inquiry_Scan_Activity 结构参数值，其指令及其描述如表 10.115 和表 10.116 所示。Inquiry_Scan_Interval 结构参数定义为连续查询扫描之间的时间量。该时间间隔被定义为主控制器上次查询扫描的开始处到下次查询扫描的开始。Inquiry_Scan_Window 结构参数定义为呼叫查询持续时间。Inquiry_Scan_Window 只能小于或等于 Inquiry_Scan_Interval。当 Inquiry_Scan 允许时，仅执行查询扫描。

表 10.115 Write_Inquiry_Scan_Activity 命令

命 令	OCF	命 令 参 数	返回参数
HCL_Write_Inquiry_Scan_Activity	0x001E	Inquiry_Scan_Interval Inquiry_Scan_Window	Status

表 10.116 Write_Inquiry_Scan_Activity 命令描述

参 数	值	参 数 说 明
Inquiry_Scan_Interval 2 字节	N=0xXXXX	长度: 2 字节, 范围: 0x0012 ~ 0x1000 时间 = $N * 0.625\text{ms}$, 范围: 11.25ms ~ 2560ms 默认: $N = 0x0800$, 时间 = 1.28s
Inquiry_Scan_Window 2 字节	N=0xXXXX	长度: 2 字节, 范围: 0x0012 ~ 0x1000 时间 = $N * 0.625\text{ms}$, 范围: 11.25ms ~ 2560ms 默认: $N = 0x0012$, 时间 = 11.25ms
Status 1 字节	0x00	Write_Inquiry_Scan_Activity 命令成功
	0x01~0xFF	Write_Inquiry_Scan_Activity 命令失败

23. Read_Authentication_Enable

该命令读出 Authentication_Enable 参数值，其指令及其描述如表 10.117 和表 10.118 所示。Authentication_Enable 参数控制是否由本地设备申请在链接设置（在创建链接命令或引入 ACL 链接的接收而且符合链接完成事件）下鉴权远程设备。在链接设置下，只有使用 Authentication_Enable 参数允许的设备可期望鉴权其他的设备。改变此参数不影响现存的链接。

表 10.117 Read_Authentication_Enable 命令

命 令	OCF	命令参数	返回参数
HCL_Read_Authentication_Enable	0x001F		Status, Authentication_enable

表 10.118 Read_Authentication_Enable 命令描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_Authentication_Enable 命令成功
	0x01~0xFF	Read_Authentication_Enable 命令失败
Authentication_Enable 1 字节	0x00	鉴权禁止
	0x01	允许所有的链接鉴权
	0x02~0xFF	保留

24. Write_Authentication_Enable

该命令写入 Authentication_Enable 参数值，其指令及其描述如表 10.119 和表 10.120 所示。Authentication_Enable 参数控制是否由本地设备申请在链接设置（在创建链接命令或引入 ACL 链接的接收而且符合链接完成事件）下鉴权远程设备。在链接设置下，只有使用 Authentication_Enable 参数允许的设备可期望鉴权其他的设备。改变此参数不影响现存的链接。

表 10.119 Write_Authentication_Enable 命令

命 令	OCF	命 令 参 数	返回参数
HCI_Write_Authentication_Enable	0x0020	Authentication_Enable	Status

表 10.120 Write_Authentication_Enable 命令描述

参 数	值	参 数 说 明
Authentication_Enable 1 字节	0x00	鉴权禁止。默认
	0x01	允许所有链接鉴权
	0x02~0xFF	保留
Status 1 字节	0x00	Write_Authentication_Enable 命令成功
	0x01~0xFF	Write_Authentication_Enable 命令成功

25. Read_Encryption_Mode

该命令读出 Encryption_Mode 参数值，其指令及其描述如表 10.121 和表 10.122 所示。Encryption_Mode 参数控制是否由本地设备申请在链接设置（在创建链接命令或引入 ACL 链接的接收而且符合链接完成事件）下加密远程设备。在链接设置下，只有使用 Authentication_Enable 参数允许和 Encryption_Mode 参数允许的设备可期望加密其他的设备。改变此参数不影响现存的链接。

表 10.121 Read_Encryption_Mode 命令

命 令	OCF	命令参数	返 回 参 数
HCI_Read_Encryption_Mode	0x0021		Status: Encryption_Mode

表 10.122 Read_Encryption_Mode 命令描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_Encryption_Mode 命令成功
	0x01	Read_Encryption_Mode 命令失败
Encryption_Mode 1 字节	0x00	加密禁止
	0x01	仅为点对点的分组加密
	0x02	为点对点 and 广播分组加密
	0x03~0xFF	保留

26. Write_Encryption_Mode

该命令写入加密模式参数值，其指令及其描述如表 10.123 和表 10.124 所示。

Encryption_Mode 参数控制是否由本地设备申请在链接设置（在创建链接命令或引入 ACL 链接的接收而且符合链接完成事件）下加密远程设备。在链接设置下，只有使用 Authentication_Enable 参数允许和 Encryption_Mode 参数允许的设备可期望加密其他的设备。改变此参数不影响现存的链接。

表 10.123 Write_Encryption_Mode 命令

命 令	OCF	命 令 参 数	返回参数
HCI_Write_Encryption_Mode	0x0022	Encryption_Mode	Status

当广播和点对点通信加密时，必须使用临时链接字。虽然该参数用于申请远程设备的加密能力，但主机不必指定比本地设备支持的加密能力更多的 Encryption_Mode 参数。

注意：当本地设备不支持加密或广播加密时，主机不要求使用 Encryption_Mode 参数的命令设成 0x01 或 0x02。当本地设备申请多于远程设备支持的加密能力时，对于只支持部分能力的新的链接（或链接完成事件），实际 Encryption_Mode 在事件里返回。例如，当远程设备不支持加密时，在事件里返回 0x00，当只支持点对点的加密时，返回 0x00 或 0x01。

表 10.124 Write_Encryption_Mode 命令描述

参 数	值	参 数 说 明
Encryption_Mode 1 字节	0x00	加密禁止，默认
	0x01	仅为点对点的分组加密
	0x02	为点对点和广播分组加密
	0x03~0xFF	保留
Status 1 字节	0x00	Write_Encryption_Mode 命令成功
	0x01~0xFF	Write_Encryption_Mode 命令失败

27. Read_Class_of_Device

该命令读出 Class_of_Device 参数值，其指令及其描述如表 10.125 和表 10.126 所示。Class_of_Device 参数用来指出本地设备到其他设备的能力。

表 10.125 Read_Class_of_Device 命令

命 令	OCF	命令参数	返 回 参 数
HCI_Read_Class_of_Device	0x0023		Status, Class_of_Device

表 10.126 Read_Class_of_Device 命令描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_Class_of_Device 命令成功
	0x01	Read_Class_of_Device 命令失败
Class_of_Device 3 字节	0XXXXXX	Class_of_Device 3 字节

28. Write_Class_of_Device

该命令写入 Class_of_Device 参数值，其指令及其描述如表 10.127 和表 10.128 所示。Class_of_Device 参数用来指出本地设备到其他设备的能力。

表 10.127 Write_Class_of_Device 命令

命 令	OCF	命令参数	返回参数
HCI_Write_Class_of_Device	0x0024	Class_of_Device	Status

表 10.128 Write_Class_of_Device 命令描述

参 数	值	参 数 说 明
Class_of_Device 3 字节	0xXXXX	设备类型
Status 1 字节	0x00	Write_Class_of_Device 命令成功
	0x01	Write_Class_of_Device 命令失败

29. Read_Voice_Setting

该命令读出 Voice_Setting 参数值，其指令及其描述如表 10.129 和表 10.130 所示。Voice_Setting 参数控制所有语音链接的各种设置：输入编码，无线编码格式，输入数据格式，输入采样容量和线性 PCM 参数。这些设置用于所有语音链接但不能设置成单个语音链接。

表 10.129 Read_Voice_Setting 命令

命 令	OCF	命令参数	返回参数
HCI_Read_Voice_Setting	0x0025		Status, Voice_Setting

表 10.130 Read_Voice_Setting 命令描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_Voice_Setting 命令成功了
	0x01	Read_Voice_Setting 命令失败了
Voice_Setting 2 字节(10 位有意义)	00XXXXXXXX	输入编码：线性
	01XXXXXXXX	输入编码：μ-law 输入编码
	10XXXXXXXX	输入编码：A-law 输入编码
	11XXXXXXXX	保留
	XX 00XXXXXX	输入数据格式：反码
	XX01XXXXXX	输入数据格式：补码
	XX10XXXXXX	输入数据格式：信号幅度
	XX11XXXXXX	保留
	XXXX0XXXXX	输入采样容量：8 位(仅为线性 PCM)
	XXXX1XXXXX	输入采样容量：16 位(仅为线性 PCM)
	XXXXXnnnXX	Linear_PCM_Bit_Pos: 采样的 MSB#位位置 总是从 MSB (仅为线性 PCM) 开始。
	XXXXXXXX00	无线电编码格式 CVSD
	XXXXXXXX01	无线电编码格式 μ-law
	XXXXXXXX10	无线电编码格式 A-law
	XXXXXXXX11	保留

30. Write_Voice_Setting

该命令写入 Voice_Setting 参数值，其指令及其描述如表 10.131 和表 10.132 所示。Voice_Setting 参数控制所有语音链接的各种设置：输入编码，无线编码格式，输入数据格式，输入采样容量和线性 PCM 参数。这些设置用于所有语音链接但不能设置成单个语音链接。

表 10.131 Write_Voice_Setting 命令

命 令	OCF	命令参数	返回参数
HCI_Write_Voice_Setting	0x0026	Voice_Setting	Status

表 10.132 Write_Voice_Setting 命令描述

参 数	值	参 数 说 明
Voice_Setting 2 字节(10 位有意义)	00XXXXXXXX	输入编码：线性
	01XXXXXXXX	输入编码：μ-law 输入编码
	10XXXXXXXX	输入编码：A-law 输入编码
	11XXXXXXXX	保留
	XX00XXXXXX	输入数据格式：反码
	XX01XXXXXX	输入数据格式：补码
	XX10XXXXXX	输入数据格式：信号幅度
	XX11XXXXXX	保留
	XXXX0XXXXX	输入采样容量：8 位(仅为线性 PCM)
	XXXX1XXXXX	输入采样容量：16 位(仅为线性 PCM)
	XXXXXnmXX	Linear_PCM_Bit_Pos：采样的 MSB#位位置总是从 MSB（仅为线性 PCM）开始。
	XXXXXXXX00	无线电编码格式 CVSD
	XXXXXXXX01	无线电编码格式 μ-law
	XXXXXXXX10	无线电编码格式 A-law
	XXXXXXXX11	保留
Status 1 字节	0x00	Read_Voice_Setting 命令成功了
	0x01	Read_Voice_Setting 命令失败了

31. Read_Automatic_Flush_Timeout

该命令对于指定的链接句柄读出 Flush_Timeout 参数值，其指令及其描述如表 10.133 和表 10.134 所示。Flush_Timeout 参数仅用于 ACL 链接。Flush_Timeout 参数定义在所有 L2CAP 分组块前的时间量，基带分组当前正在发送时，由主控制器自动刷新。当传输期望构造 L2CAP 分组的第一个基带分组时，超时区段开始。如果没有主机发送刷新命令时，它允许 ACL 分组自动地刷新。该命令提供对等式数据的支持，例如图象。当此时正在传输的 L2CAP 分组被自动地“刷新”时，失败计数器增加 1。指定链接句柄传输的下一个 L2CAP 分组的第一个块可预先存储在主控制器里。在这种情况下，含有 L2CAP 分组数据的第一个基带分组的传输可直接开始。如果下一个 L2CAP 分组没存在主控制器里，在相同链接句柄刷新后，发送给主控制器的所有数据通过主控制器放弃，直到收到具有 Packed_Boundary Packed_Boundary_Flag(0x02)开始的 HCI 数据分组。当该种情况发生时，一次新的传输期望

构成。

表 10.133 Read_Automatic_Flush_Timeout 命令

命 令	OCF	命令参数	返回参数
HCI_Read_Automatic_Flush_Timeout	0x0027	Connection_Handle	Status; Connection_Handle; Flush_Timeout

表 10.134 Read_Automatic_Flush_Timeout 命令描述

参 数	值	参 数 说 明
Connection_handle 2 字节(12 位有意义)	0xFFFF	指定读哪个链接句柄的刷新超时。 范围: 0x0000~0x0EFF (0x0F00 ~ 0x0FFF 保留)
Status 1 字节	0x00	Read_Automatic_刷新时间命令成功。
	0x01	Read_Automatic_刷新时间命令失败。
Connection_Handle 2 字节(12 位有意义)	0xFFFF	指定哪个链接句柄的刷新超时已读过 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)
Flush_Timeout 2 字节	0	超时 = ∞; 无自动刷新
	N=0xFFFF	刷新超时 = N * 0.625ms 长度: 11 字节, 范围: 0x0001 ~ 0x07FF

32. Write_Automatic_Flush_Timeout

该命令对于指定的链接句柄写入 Flush_Timeout 参数值，其指令及其描述如表 10.135 和表 10.136 所示.Flush_Timeout 参数仅用于 ACL 链接.Flush_Timeout 参数定义在所有 L2CAP 分组块前的时间量，基带分组当前正在发送时，由主控制器自动刷新。当传输期望构造 L2CAP 分组的第一个基带分组时，超时区段开始。如果没有主机发送刷新命令时，它允许 ACL 分组自动地刷新。该命令提供对等式数据的支持，例如图象。当此时正在传输的 L2CAP 分组被自动地“刷新”时，失败计数器增加 1。指定链接句柄的传输的下一个 L2CAP 分组的第一个块可预先存储在主控制器里。在这种情况下，含有 L2CAP 分组数据的第一个基带分组的传输可直接地开始。如果下一个 L2CAP 分组没存在主控制器里，在相同链接句柄刷新后，发送给主控制器的所有数据通过主控制器放弃，直到具有 Packed_Boundary_Flag(0x02)开始的 HCI 数据分组收到。当该种情况发生时，一次新的传输期望构成。

表 10.135 Write_Automatic_Flush_Timeout 命令

命 令	OCF	命令参数	返回参数
HCI_Write_Automatic_Flush_Timeout	0x0028	Connection_Handle; Flush_Timeout	Status 链接句柄

表 10.136 Write_Automatic_Flush_Timeout 命令描述

参 数	值	参 数 说 明
Connection_Handle 2 字节(12 位有意义)	0xFFFF	指定写哪个链接句柄的刷新超时 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)

续表

参 数	值	参 数 说 明
Flush_Timeout 2 字节	N=0xXXXX	超时 = ∞: 无自动刷新, 默认 刷新超时 = $N * 0.625ms$ 长度: 11 字节, 范围: 0x0001~0x07FF
Status 1 字节	0x00	Write_Automatic_刷新时间命令成功
	0x01	Write_Automatic_刷新时间命令失败
Connection_Handle 2 字节(12 位有意义)	0xXXXX	指定哪个链接句柄的刷新超时已写过 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)

33. Read_Num_Broadcast_Retransmissions

该命令读出作为广播重传次数设备的参数值, 其指令及其描述如表 10.137 和表 10.138 所示。广播分组不需确认且不可靠, 广播重传次数参数通过多次重传广播消息来提高广播消息的可靠性。该参数定义了设备重传广播数据分组的次数, 同时该参数随链接质量变化而被调整。

表 10.137 Read_Num_Broadcast_Retransmissions 命令

命 令	OCF	命令参数	返回参数
HCI_Read_Num_ Broadcast_Retransmission	0x0029		Status, Num_Broadcast_Retran

表 10.138 Read_Num_Broadcast_Retransmissions 命令描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_Num_Broadcast_Retransmissions 命令成功
	0x01~0xFF	Read_Num_Broadcast_Retransmissions 命令失败
Num_Broadcast_Retran 1 字节	N = 0xFF	广播重发次数 = N, 范围: 0x00-0xFF

34. Write_Num_Broadcast_Retransmissions

该命令写入作为广播重传次数设备的参数值, 其指令及其描述如表 10.139 和表 10.140 所示。广播分组不需确认, 广播重传次数参数通过多次重传广播消息来提高广播消息的可靠性。该参数定义了设备重传广播数据分组的次数, 同时该参数随链接质量测量变化而被调整。

表 10.139 Write_Num_Broadcast_Retransmissions 命令

命 令	OCF	命令参数	返回参数
HCI_Write_Num_Broadcast_ Retransmissions	0x002A	Num_Broadcast_Retran	Status

表 10.140 Write_Num_Broadcast_Retransmissions 命令描述

参 数	值	参 数 说 明
Num_Broadcast_Retran 1 字节	N = 0xFF	广播重发次数 = N, 范围: 0x00 ~ 0xFF 默认: N = 0x01
Status 1 字节	0x00	Write_Num_Broadcast_Retransmissions 命令成功
	0x01~0xFF	Write_Num_Broadcast_Retransmissions 命令失败

35. Read_Hold_Mode_Activity

该命令读出 Hold_Mode_Activity 参数值，其指令及其描述如表 10.141 和表 10.142 所示。当设备处于保持模式时，Hold_Mode_Activity 值用于确定活动是否挂起。在保持时期终止后，设备返回原操作模式。通过执行不同活动类型的按位“或”操作，多保持模式活动可由 Hold_Mode_Activity 参数指定。如果没有活动被挂起，则在保持模式期间，所有当前设置的定期查询、查询扫描和呼叫扫描仍然有效。如果 Hold_Mode_Activity 参数设置成挂起呼叫扫描、挂起查询扫描和挂起定期查询，则在保持模式期间，设备可进入低功耗状态，同时所有活动都被挂起。通过执行不同活动类型的按位“或”操作，多挂起模式活动可由 Hold_Mode_Activity 参数指定。如果整个链接处于保持模式，只有保持模式活动是有效的。

表 10.141 Read_Hold_Mode_Activity 命令

命 令	OCF	命令参数	返回参数
HCI_Read_Hold_Mode_Activity	0x002B		Status: Hold_Mode_Activity

表 10.142 Read_Hold_Mode_Activity 命令描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_Hold_Mode_Activity 命令成功
	0x01	Read_Hold_Mode_Activity 命令失败
Hold_Mode_Activity 1 字节	0x00	维持当前功率状态
	0x01	挂起呼叫扫描
	0x02	挂起查询扫描
	0x04	挂起定期查询
	0x08-0xFF	保留

36. Write_Hold_Mode_Activity

该命令写入 Hold_Mode_Activity 参数值。其指令及其描述如表 10.143 和表 10.144 所示。当设备处于保持模式时，Hold_Mode_Activity 值用于确定活动是否挂起。在保持时期终止后，设备返回原操作模式。通过执行不同活动类型的按位“OR”操作，多保持模式活动可由 Hold_Mode_Activity 参数指定。如果没有活动被挂起，则在保持模式期间，所有当前设置的定期查询、查询扫描和呼叫扫描仍然有效。如果 Hold_Mode_Activity 参数设置成挂起呼叫扫描、挂起查询扫描和挂起定期查询，则在保持模式期间，设备可进入低功耗状态，同时所有活动都被挂起。通过执行不同活动类型的按位“OR”操作，多挂起模式活动可由 Hold_Mode_Activity 参数指定。如果整个链接处于保持模式，只有保持模式活动是有效的。

表 10.143 Write_Hold_Mode_Activity 命令

命 令	OCF	命令参数	返回参数
HCI_Write_Hold_Mode_Activity	0x002C	Hold_Mode_Activity	Status

表 10.144 Write_Hold_Mode_Activity 命令描述

参 数	值	参 数 说 明
Hold_Mode_Activity 1 字节	0x00	维持当前功率状态，默认
	0x01	挂起呼叫扫描
	0x02	挂起查询扫描
	0x04	挂起定期查询
	0x08-0xFF	保留
Status 1 字节	0x00	Write_Hold_Mode_Activity 命令成功
	0x01	Write_Hold_Mode_Activity 命令失败

37. Read_Transmit_Power_Level

该命令对于指定的链接句柄读出 Transmit_Power_Level 参数值。链接句柄必须是 ACL 方式的链接句柄。其指令及其描述如表 10.145 和表 10.146 所示。

表 10.145 Read_Transmit_Power_Level 命令

命 令	OCF	命 令 参 数	返 回 参 数
HCI_Read_Transmit_Power_Level	0x002D	Connection_handle, Type	Status, Connection_Handle, Transmit_Power_Level

表 10.146 Read_Transmit_Power_Level 命令描述

参 数	值	参 数 说 明
Connection_handle 2 字节(12 位有意义)	0xFFFF	指定读出哪种链接句柄的功率电平设置 范围: 0x0000~0x0EFF (0x0F00 ~ 0x0FFF 保留)
Type 1 字节	0x00	读当前传输功率电平
	0x01	读最大传输功率电平
	0x02~0xFF	保留
Status 1 字节	0x00	Read_Transmit_Power_Level 命令成功
	0x01~0xFF	Read_Transmit_Power_Level 命令失败
Connection_handle 2 字节(12 位有意义)	0xFFFF	指定返回哪个链接句柄传输的功率电平设置 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)
Transmit_Power_Level 1 字节	N=0xFF	长度: 1 字节 (带符号整数), 范围: $-30 \leq N \leq 20$ 单位: dBm

38. Read_SCO_Flow_Control_Enable

该命令提供读出 SCO_Flow_Control_Enable 设置的能力，其指令及其描述如表 10.147 和表 10.148 所示。通过使用该设置，主机能决定对于 SCO 链接句柄，是否主控制器将送出完成分组事件数。该设置能允许主机启动及禁止 SCO 流控制。如果不存在链接情况，只能改变 SCO_Flow_Control_Enable 设置。

表 10.147 Read_SCO_Flow_Control_Enable 命令

命 令	OCF	命令参数	返 回 参 数
HCI_Read_SCO_Flow_Control_Enable	0x002E		Status: SCO_Flow_Control_Enable

表 10.148 Read_SCO_Flow_Control_Enable 命令描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_SCO_Flow_Control_Enable 命令成功
	0x01~0xFF	Read_SCO_Flow_Control_Enable 命令失败
SCO_Flow_Control_Enable 1 字节	0x00	禁止 SCO 流控制。对于 SCO 链接句柄，主控制器里无完成分组事件数送出
	0x01	允许 SCO 流控制。对于 SCO 链接句柄，主控制器里有完成分组事件数送出

39. Write_SCO_Flow_Control_Enable

该命令提供写入 SCO_Flow_Control_Enable 设置的能力，其指令及其描述如表 10.149 和表 10.150 所示。通过使用该设置，主机能决定对于 SCO 链接句柄，是否主控制器将送出完成分组事件数。该设置能允许主机启动及禁止 SCO 流控制。如果不存在链接情况，只能改变 SCO_Flow_Control_Enable 设置。

表 10.149 Write_SCO_Flow_Control_Enable 命令

命 令	OCF	命 令 参 数	返回参数
HCI_Write_SCO_Flow_Control_Enable	0x002F	SCO_Flow_Control_Enable	Status

表 10.150 Write_SCO_Flow_Control_Enable 命令描述

	值	参 数 说 明
SCO_Flow_Control_Enable 1 字节	0x00	禁止 SCO 流控制。对于 SCO 链接句柄，主控制器里无完成分组事件数送出，默认
	0x01	允许 SCO 流控制。对于 SCO 链接句柄，主控制器里有完成分组事件数送出
Status 1 字节	0x00	Write_SCO_Flow_Control_Enable 命令成功
	0x01	Write_SCO_Flow_Control_Enable 命令失败

40. Set_host_Controller_To_host_Flow_Control

该命令用于主机在主控制器到主机方向打开或关闭流控制。如果流控制关闭，主机就不发送 Host_Number_Of_Completed_Packed 命令。如果该命令已由主机发送且流控制是关闭的，则该命令通过主控制器忽略。其指令及其描述如表 10.151 和表 10.152 所示。

表 10.151 Set_host_Controller_To_host_Flow_Control 命令

命 令	OCF	命 令 参 数	返回参数
HCI_Set_host_Controller_To_host_Flow_Control	0x0031	Flow_Control_Enable	Status

表 10.152 Set_host_Controller_To_host_Flow_Control 命令描述

	值	参 数 说 明
Flow_Control_Enable 1 字节	0x00	从主控制器到主机方向关闭流控制，默认
	0x01	从主控制器到主机方向开启流控制
	0x02~	保留
	0xFF	
Status 1 字节	0x00	Set_host_Controller_To_host_Flow_Control 命令成功
	0x01	Set_host_Controller_To_host_Flow_Control 命令失败

41. Host_Buffer_Size

该命令用于主机通知主控制器有关从主控制器到主机 HCI ACL 和 SCO 数据分组发送的数据部分的最大长度。根据长度规定，主控制器将分段传输这些数据，所以 HCI 数据分组包含最大使用长度。该命令也通知主控制器能够存放在主机数据缓冲区的 HCI ACL 和 SCO 数据分组的总数。其指令及其描述如表 10.153 和表 10.154 所示。

表 10.153 Host_Buffer_Size 命令

命 令	OCF	命 令 参 数	返回参数
HCI_host_Buffer_Size	0x0033	host_ACL_Data_Packet_Length , host_SCO_Data_Packet_Length , host_Total_Num_ACL_Data_Packets, host_Total_Num_SCO_Data_Packets	Status

如果从主控制器到主机的控制被关闭，而且 Host_Buffer_Size 命令还没通过主机发布，这意味着主控制器可随意使用任何长度发送 HCI 数据分组到主机，同时可假设数据缓冲区是无限的。如果从主控制器到主机的流控制是打开的，则 Host_Buffer_Size 命令必须在电源打开或复位后通过主机在第一次 Host_Number_Of_Completed_Packets 命令发送前发送（Set_Host_Controller_To_Host_Flow_Control 命令用来打开或关闭流控制）。Host_ACL_Data_Packet_Length 命令参数用来确定包含在 ACL 数据分组内的 L2CAP 段的长度，该分组从主控制器传送到主机。Host_SCO_Data_Packet_Length 命令参数用来确定 HCI SCO 数据分组的最大容量。主机和主控制器双方都必须支持该命令和事件分组，在分组里的数据部分（含头）长度是 255 个字节。Host_Total_Num_ACL_Data_Packets 命令参数包含有可存储在主机数据缓冲区里的 HCI ACL 数据分组的总数。主控制器可确定在不同链接句柄间缓冲区如何划分问题。Host_Num_SCO_Data_Packets 命令参数给出 HCI SCO 数据分组的同样信息。

注意：Host_ACL_Data_Packet_Length 和 Host_SCO_Data_Packet_Length 命令参数不包括 HCI 数据分组头的长度。

表 10.154 Host_Buffer_Size 命令描述

参 数	值	参 数 说 明
Host_ACL_Data_Packet_Length 2 字节	0XXXXX	主机能接受的所有 HCI ACL 数据分组的数据部分最大长度（在字节里）
Host_SCO_Data_Packet_Length 1 字节	0XXX	主机能接受的所有 HCI SCO 数据分组的数据部分最大长度（在字节里）
Host_Total_Num_ACL_Data_Packets 2 字节	0XXXXX	能存储在主机数据缓冲区的 HCI ACL 数据分组总数
Host_Total_Num_SCO_Data_Packets 2 字节	0XXXXX	能存储在主机数据缓冲区的 HCI SCO 数据分组总数
Status 1 字节	0x00	host_Buffer_Size 命令成功
	0x01	host_Buffer_Size 命令失败

42. Host_Number_Of_Completed_Packets

该命令用于由主机指出主控制器完成每次链接句柄的 HCI 数据分组数，意指在主机里的相应缓冲区空间已释放。其指令及其描述如表 10.155 和表 10.156 所示。

基于 Host_Buffer_Size 命令的该信息，Host_Total_Num_Data_Packets 及 Host_Total_Num_SCO_Data_Packets 命令参数，主控制器可以确定紧随 HCI 数据分组的哪个链接句柄将送往主机。如果从主控制器到主机方向的流控制是打开的，而且至少有一个链接句柄，或主控制器处于本地回送模式，则该命令由主机发布。否则，该命令由主控制器忽略。当主机在自己的缓冲区中具有 HCI 数据分组时，它必须定期持续的发送 Host_Number_Of_Completed_Packets 命令到主控制器，直到最终报告在主机里的所有缓冲空间通过 ACL 数据分组已释放。使用该命令的频率由生产厂商指定。如果从主控制器到主机方向的流控制是打开的，则在 Host_Buffer_Size 命令总是在打开电源和复位后在第一个 Host_Number_Of_Completed_Packets 命令发送前由主机发送。

注意：Host_Number_Of_Completed_Packets 命令是一个特定命令，它意指在命令已完成后，一般无事件产生。当至少有一个链接，或主控制器处于独立于其他的命令本地回送模式，则该命令通过主机可在任何时候发送。

表 10.155 Host_Number_Of_Completed_Packets 命令

命 令	OCF	命 令 参 数	返回参数
HCI_host_Number_Of_Completed_Packets	0x0035	Number_Of_Handles , Connection_handle , Host_Num_Of_Completed_Packets	

表 10.156 Host_Number_Of_Completed_Packets 命令描述

参 数	值	参 数 说 明
Number_Of_Handles 1 字节	0xXX	链接句柄数及包含在该命令里的 Host_Number_Of_Completed_Packets 参数对。范围：0 ~ 255
Connection Number_Of_Handles * 2 字节(12 位有意义)	0XXXXX	链接句柄 范围：0x0000 ~ 0x0FFF (0x0F00 ~ 0x0FFF 保留)
Host_Num_Of_Completed_Packets Number_Of_Handles * 2 字节	N=0XXXXX	由于前次事件已返回，与链接句柄有关的 HCI 数据分组的数已完成 N 的范围：0x0000 ~ 0xFFFF

通常，在 Host_Number_Of_Completed_Packets 命令完成后无事件产生。然而，如果 Host_Number_Of_Completed_Packets 命令包含一个以上无效参数，则主机用失败状态返回一个命令完成事件，并指出无效 HCI 命令参数的错误代码。

当至少有一个链接或主控制器处于本地回送模式时，主机可在任何时候发送 Host_Number_Of_Completed_Packets 命令。通常命令流控制不作为该命令。

43. Read_Link_Supervision_Timeout

该命令为设备读出 Link_Supervision_Timeout 参数值，其指令及其描述如表 10.157 和表 10.158 所示。Link_Supervision_Timeout 参数由主单元或从单元蓝牙设备用来监视链接损失。若从链接句柄无基带分组收到的持续期大于 Link_Supervision_Timeout，则链接断开。对于由链接句柄指定的设备，SCO 和 ACL 链接方式都使用同样的超时值。

注意：用于该命令的链接句柄必须是 ACL 链接方式的适当设备。该命令针对该设备的其他 SCO 链接句柄设置 Link_Supervision_Timeout 值。通过设置 Link_Supervision_Timeout，No Link_Supervision_Timeout (0x0000)将禁止对指定的链接句柄 Link_Supervision_Timeout 校验。这就没有必要每约 40 秒就要使匹克网的主单元解除休眠及休眠每个蓝牙设备。通过使用 No Link_Supervision_Timeout 设置，休眠模式的可伸缩性是无限制的。

表 10.157 Read_Link_Supervision_Timeout 命令

命 令	OCF	命 令 参 数	返 回 参 数
HCI_Read_Link_Supervision_Timeout	0x0036	Connection_handle	Status, Connection_handle Link_Supervision_Timeout

表 10.158 Read_Link_Supervision_Timeout 命令描述

参 数	值	参 数 说 明
Connection_Handle 2 字节(12 位有意义)	0XxxxX	指定读出哪个链接句柄的链接监督超时值 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)
Status 2 字节	0XXXXX	指定读出哪个链接句柄的链接监督超时值 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)
Status 1 字节	0x00	Read_Link_Supervision_Timeout 命令成功
	0x01	Read_Link_Supervision_Timeout 命令失败
Link_Supervision_timeout 2 字节	0x0000	无 Link_Supervision_Timeout
	N=0xXXXX	测量基带时隙数, link_supervision_timeout = N * 0.625ms N 的范围: 0x0001 ~ 0xFFFF; 时间范围: 0.625ms ~ 40.9s

44. write_Link_Supervision_Timeout

该命令为设备写入 Link_Supervision_Timeout 参数值，其指令及其描述如表 10.159 和表 10.160 所示。Link_Supervision_Timeout 参数由主单元或从单元蓝牙设备用来监视链接损失。若从链接句柄无基带分组收到的持续期大于 Link_Supervision_Timeout，则链接断开。对于由链接句柄指定的设备，SCO 和 ACL 链接方式都使用同样的超时值。

注意：用于该命令的链接句柄必须是 ACL 链接方式的适当设备。该命令针对为该设备其他的 SCO 链接句柄设置 Link_Supervision_Timeout 值。通过设置 Link_Supervision_Timeout，No Link_Supervision_Timeout (0x0000)将禁止对指定的链接句柄

Link_Supervision_Timeout 校验。这就没有必要每约 40 秒就要使匹克网的主单元解除休眠及休眠每个蓝牙设备。通过使用 No Link_Supervision_Timeout 设置，休眠模式的可伸缩性是无限制的。

表 10.159 write_Link_Supervision_Timeout 命令

命 令	OCF	命 令 参 数	返回参数
HCI_Write_Link_Supervision_Timeout	0x0037	Connection_Handle Link_Supervision_Timeout	Status Connection_handle

表 10.160 write_Link_Supervision_Timeout 命令描述

参 数	值	参 数 说 明
Connection_Handle 2 字节(12 位有意义)	0xFFFF	指定写入哪个链接句柄的链接监督超时值 范围: 0x0000 - 0x0EFF (0x0F00 ~ 0x0FFF 保留)
Link_Supervision_Timeout 2 字节	0x0000	无 Link_Supervision_Timeout
	N=0xFFFF	测量基带时隙数 $link_supervision_timeout = N * 0.625ms$ N 的范围: 0x0001 ~ 0xFFFF, 范围: 0.625ms ~ 40.9s 默认: N = 0x7D00, link_supervision_timeout = 20s
Status 1 字节	0x00	Write_Link_Supervision_Timeout 命令成功
	0x01	Write_Link_Supervision_Timeout 命令失败
Connection_Handle 2 字节(12 位有意义)	0xFFFF	指定写入哪个链接句柄的链接监督超时值 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)

45. Read_Number_Of_Supported_IAC

该命令读出本地蓝牙设备在查询期间能同时监听的查询识别码数(IAC)的值。所有的蓝牙设备要求至少支持一种 IAC, (GIAC 或 UIAC), 但是有些蓝牙设备支持附加的 IACs 。其指令及其描述如表 10.161 和表 10.162 所示。

表 10.161 Read_Number_Of_Supported_IAC 命令

命 令	OCF	命令参数	返回参数
HCI_Read_Number_Of_Supported_IAC	0x0038		Status Num_support_IAC

表 10.162 Read_Number_Of_Supported_IAC 命令描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_Number_Of_Supported_IAC 命令成功
	0x01	Read_Number_Of_Supported_IAC 命令失败
Num_Support_IAC 1 字节	0xFF	指定本地蓝牙设备在查询期间可同时监听支持的 ICA 量。 范围: 0x01 ~ 0xFF

46. Read_Current_IAC_LAP

该命令读出用于创建查询期间本地蓝牙设备能同时扫描的查询识别码的 LAP (s)。所

有的蓝牙设备要求至少支持一种 IAC（GIAC 或 UIAC），但有些蓝牙设备支持附加的 IACs。其指令及其描述如表 10.163 和表 10.164 所示。

表 10.163 Read_Current_IAC_LAP 命令

命 令	OCF	命令参数	返回参数
HCI_Read_Current_IAC_LAP	0x0039		Status, Num_current_IAC, IAC_lap

表 10.164 Read_Current_IAC_LAP 命令描述

参 数	值	参数说明
Status 1 字节	0x00	Read_Current_IAC_LAP 命令成功
	0x01	Read_Current_IAC_LAP 命令失败
Num_Current_IAC 1 字节	0xXX	指定查询期间通过本地蓝牙设备当前使用的 IACs 的数。 范围：0x01 ~ 0x40
IAC_LAP 3 字节 * Num_Current_IAC	0xXXX XXX	用于创建 IAC 的 LAPs, IAC 为在查询期间当前用于本地 蓝牙设备同时监听的参数。范围：0x9E8B00~0x9E8B3F

47. Write_Current_IAC_LAP

该命令写入用于创建查询期间本地蓝牙设备能同时扫描的查询识别码的 LAP（S）。所有的蓝牙设备要求至少支持一种 IAC（GIAC 或 UIAC），但有些蓝牙设备支持附加的 IACs。因此，用于创建 GIAC 或 UIAC 的 LAP 必须是在该命令的 IAC_LAP 中。其指令及其描述如表 10.165 和表 10.166 所示。该命令通过蓝牙设备使用改写了当前的 IACs。如果 Num_Current_IAC 的值大于支持 IACs 的数，仅为“1”，X 查询识别码（X 等于支持 IACs 的数）以无任何错误的形式被存储。

表 10.165 Write_Current_IAC_LAP 命令

命 令	OCF	命 令 参 数	返回参数
HCI_Write_Current_IAC_LAP	0x003A	Num_Current_IAC, IAC_LAP	Status

表 10.166 Write_Current_IAC_LAP 命令描述

参 数	值	参 数 说 明
Num_Current_IAC 1 字节	0xXX	指定查询期间通过本地蓝牙设备当前使用的 IACs 的数。 范围：0x01 ~ 0x40
IAC_LAP 3 字节 * Num_Current_IAC	0XXXXXXXX	用于创建 IAC 的 LAPs, IAC 为在查询期间当前用于本地蓝 牙设备同时监听的参数。范围：0x9E8B00 ~ 0x9E8B3F GIAC 是使用的默认 IAC；如果支持附加的 IACs，附加默认 IAC 由生产厂商确定。
Status	0x00	Write_Current_IAC_LAP 命令成功
	0x01	Write_Current_IAC_LAP 命令失败

48. Read_Page_Scan_Period_Mode

该命令用来读出本地蓝牙设备的强制 Page_Scan_Period_Mode。每次查询响应消息发送时，蓝牙设备启动定时器(T_mandatory_pscan)，该定时器的值取决于 Page_Scan_Period_Mode。只要该定时器没终止，蓝牙设备将使用所有后面呼叫扫描的 Page_Scan_Period_Mode。其指令及其描述如表 10.167 和表 10.168 所示。

注意：在每次新的查询响应时，定时器 T_mandatory_pscan 将被复位。在传输一个或多个查询响应（FHS）分组作为查询扫描过程时，本地蓝牙设备使用强制呼叫扫描模式，而不管 scan_Enable 参数的设置，允许进入呼叫扫描状态。

表 10.167 Read_Page_Scan_Period_Mode 命令

命 令	OCF	命令参数	返回参数
HCI_Read_Page_Scan_Period_Mode	0x003B		Status: Page_Scan_Period_Mode

表 10.168 Read_Page_Scan_Period_Mode 命令描述

参 数	值	参数说明
Status 1 字节	0x00	Read_Page_Scan_Period_Mode 命令成功
	0x01	Read_Page_Scan_Period_Mode 命令失败
Page_Scan_Period_Mode 1 字节	0x00	P0
	0x01	P1
	0x02	P2
	0x03-0xFF	保留

49. Write_Page_Scan_Period_Mode

该命令用来写入本地蓝牙设备的强制 Page_Scan_Period_Mode。每次查询响应消息发送时，蓝牙设备启动定时器(T_mandatory_pscan)，该定时器的值取决于 Page_Scan_Period_Mode。只要该定时器没终止，蓝牙设备将使用所有后面呼叫扫描的 Page_Scan_Period_Mode。其指令及其描述如表 10.169 和表 10.170 所示。

表 10.169 Write_Page_Scan_Period_Mode 命令

命 令	OCF	命令参数	返回参数
HCI_Write_Page_Scan_Period_Mode	0x003C	Page_Scan_Period_Mode	Status

表 10.170 Write_Page_Scan_Period_Mode 命令描述

参 数	值	参数说明
Page_Scan_Period_Mode 1 字节	0x00	P0
	0x01	P1
	0x02	P2
	0x03-0xFF	保留
Status 1 字节	0x00	Write_Page_Scan_Period_Mode 命令成功
	0x01	Write_Page_Scan_Period_Mode 命令失败

注意：在每次新的查询响应时，定时器 T_mandatory_pscan 将被复位。在传输一个或多个查询响应（FHS）分组作为查询扫描过程时，本地蓝牙设备使用强制呼叫扫描模式，而不管 scan_Enable 参数的设置，允许进入呼叫扫描状态。

50. Read_Page_Scan_Mode

该命令用来读出本地蓝牙设备的默认呼叫扫描模式。Page_Scan_Mode 参数指出用于默认呼叫扫描的呼叫扫描模式。当前定义了一个强制呼叫扫描模式和 3 个选择呼叫扫描模式。如果基带定时器 T_mandatory_pscan 没终止，随后的查询响应必须使用强制呼叫扫描模式。其指令及其描述如表 10.171 和表 10.172 所示。

表 10.171 Read_Page_Scan_Mode 命令

命 令	OCF	命令参数	返回参数
HCI_Read_Page_Scan_Mode	0x003D		Status, Page-Scan-Mode

表 10.172 Read_Page_Scan_Mode 命令描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_Page_Scan_Mode 命令成功
	0x01	Read_Page_Scan_Mode 命令失败
Page_Scan_Mode 1 字节	0x00	强制呼叫扫描模式
	0x01	选择扫描模式 I
	0x02	选择扫描模式 II
	0x03	选择扫描模式 III
	0x04~0xFF	保留

51. Write_Page_Scan_Mode

该命令用来写入本地蓝牙设备的默认呼叫扫描模式。Page_Scan_Mode 参数指出用于默认呼叫扫描的呼叫扫描模式。当前定义了一个强制呼叫扫描模式和 3 个选择呼叫扫描模式。如果基带定时器 T_mandatory_pscan 没终止，随后的查询响应必须使用强制呼叫扫描模式。其指令及其描述如表 10.173 和表 10.174 所示。

表 10.173 Write_Page_Scan_Mode 命令

命 令	OCF	命令参数	返回参数
HCI_Write_Page_Scan_Mode	0x003E	Page_Scan_Mode	Status

表 10.174 Write_Page_Scan_Mode 命令描述

参 数	值	参 数 说 明
Page_Scan_Mode 1 字节	0x00	强制呼叫扫描模式，默认
	0x01	选择呼叫扫描模式 I
	0x02	选择呼叫扫描模式 II
	0x03	选择呼叫扫描模式 III
	0x04~0xFF	保留
Status 1 字节	0x00	Write_Page_Scan_Mode 命令成功
	0x01	Write_Page_Scan_Mode 命令失败

10.3.7 信息参数

信息参数（见表 10.175）由蓝牙硬件制造商固定。这些参数提供有关蓝牙设备的信息和主控制器、链路管理器及基带的容量。主机不能修改这些参数的任何东西。对于信息参数，OGF 定义为 0x04。

表 10.175 信息参数命令一览表

命 令	命令说明汇总
Read_Local_Version_Information	Read_Local_Version_Information 命令读出本地蓝牙设备的版本信息值
Read_Local_Supported_Features	Read_Local_Supported_Features 命令申请本地设备支持特征表
Read_Buffer_Size	Read_Buffer_Size 命令返回 HCI 缓冲区的容量。通过主控制器这些缓冲区用于传输缓冲数据
Read_Country_Code	Read_Country_Code 命令读出国家代码状态参数值。国家代码定义了 ISM2.4GHz 波道的那些频段由设备使用
Read_BD_ADDR	Read_BD_ADDR 读出 BD_ADDR 的参数值。BD_ADDR 是蓝牙设备的一个 48 位惟一标识符

1. Read_local_Version_Information

该命令读出本地蓝牙设备版本信息值，其指令及其描述如表 10.176 和表 10.177 所示。版本信息由 2 个参数组成：版本和修正参数。版本参数定义了蓝牙硬件的主要硬件版本。当蓝牙硬件新版本为新的蓝牙 SIG 说明生产时，只有版本参数改变。版本参数由 SIG 控制。修订参数由制造商控制，当需要时，可以修改。Manufacturer_Name 参数指出本地蓝牙模型的制造商，并通过 LMP 定义的该参数指定。子版本参数由制造商控制，当需要时可以修改。定义的蓝牙硬件各个版本的各种修正子版本参数，将作为设计进程变化和错误固定而通过。它允许由软件来确定正在使用什么样的蓝牙硬件，如有必要，硬件可在各类故障范围工作。

表 10.176 Read_local_Version_Information 命令

命 令	OCF	命令参数	返回参数
HCI_Read_local_Version_information	0x0001		Status: HCI Version; HCI Revision; LMP Version; Manufacturer_Name; LMP Subversion

表 10.177 Read_local_Version_Information 命令描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_local_Version_Information 命令成功
	0x01	Read_local_Version_Information 命令失败
HCI_Version 1 字节	0xXX	蓝牙硬件的当前 HCI 版本。 蓝牙 HCI 规范 1.0 0x01 ~ 0xFF: 保留

续表

参 数	值	参 数 说 明
LMP_Version 1 字节	0xXX	蓝牙硬件的当前 LMP 版本
Manufacturer_Name 2 字节	0XXXXX	蓝牙硬件制造商名
LMP_Subversion 2 字节	0XXXXX	蓝牙硬件的当前 LMP 子版本

2. Read_Local_Supported_Features

该命令为本地设备申请支持特征表。该命令返回 LMP 特征表。其指令及其描述如表 10.178 和表 10.179 所示。

表 10.178 Read_Local_Supported_Features 命令

命 令	OCF	命令参数	返回参数
HCI_Read_Local_Supported_Features	0x0003		Status, LMP_feature

表 10.179 Read_Local_Supported_Features 命令描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_Local_Supported_Features 命令成功
	0x01	Read_Local_Supported_Features 命令失败
LMP_Features 8 字节	0XXXXXXXXX XXXXXXXXXX	LMP 特征的位屏蔽表

3. Read_Buffer_Size

该命令用来读出从主机到主控制器发送 HCI ACL 和 SCO 数据分组的数据部分最大值。主机根据这些分组大小，分段从主机传输到主控制器，以便 HCI 数据分组包含这类大小的数据。其指令及其描述如表 10.180 和表 10.181 所示。

表 10.180 Read_Buffer_Size 命令

命 令	OCF	命令参数	返 回 参 数
HCI_Read_Buffer_Size	0x0005		Status, HCI_ACL_Data_Packet_Length, HCI_SCO_Data_Packet_Length, HCI_Total_Num_ACL_Data_Packets, HCI_Total_Num_SCO_Data_Packets

Read_Buffer_Size 命令也返回能存储在主控制器缓冲区里的 HCI ACL 和 SCO 数据分组的总数。Read_Buffer_Size 命令必须在主机发送任何数据到主控制器前由主机发布。

HCI_ACL_Data_Packet_Length 返回参数用来确定包含在 ACL 数据分组里的 L2CAP 段的大小，从主机传输到主控制器的 L2CAP 段通过链路管理器分散进入基带分组。

HCI_SCO_Data_Packet_Length 返回参数用来确定 HCI SCO 数据分组的最大容量。主机和主控制器双方都必须支持该命令及事件分组，此时，包含在分组里的数据部分(除头外)是 255 个字节。

HCI_Total_Num_ACL_Data_Packets 返回参数包含存储在主控制器数据缓冲区的 HCI ACL 数据分组总数。主机将确定在不同的链接句柄之间缓冲区如何进行划分。如果没有 HCI SCO 数据分组，HCI_Total_Num_SCO_Data_Packet 返回参数给出了相同信息。

注意：HCI_ACL_Data_Packet_Length 和 HCI_SCO_Data_Packet_Length 返回参数不包括 HCI 的数据分组头的长度。

表 10.181 Read_Buffer_Size 命令描述

参 数	值	参数说明
Status 1 字节	0x00	Read_Buffer_Size 命令成功
	0x01	Read_Buffer_Size 命令失败
HCI_ACL_Data_Packet_Length 2 字节	0xFFFF	主控制器可接受的各 HCI ACL 数据分组的数据部分最大长度
HCI_SCO_Data_Packet_Length 1 字节	0xFF	主控制器可接受的各 HCI SCO 数据分组的数据部分最大长度
HCI_Total_Num_ACL_Data_Packets 2 字节	0xFFFF	能存储在主控制器数据缓冲区里的 HCI ACL 数据分组总数
HCI_Total_Num_SCO_Data_Packets 2 字节	0xFFFF	能存储在主控制器数据缓冲区里的 HCI SCO 数据分组总数

4. Read_Country_Code

该命令读出 Country_Code 返回参数值。Country_Code 定义 ISM 2.4GHz 波道的哪些频带可被设备使用。各国根据自身的行规调整可使用的 2.4GHz 频率范围。其指令及其描述如表 10.182 和表 10.183 所示。

表 10.182 Read_Country_Code 命令

命 令	OCF	命令参数	返回参数
HCI_Read_Country_Code	0x0007		Status, Country_Code

表 10.183 Read_Country_Code 命令描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_Country_Code 命令成功
	0x01	Read_Country_Code 命令失败
Country_Code 1 字节	0x00	北美洲与欧洲（西班牙、法国除外）
	0x01	法国
	0x02	西班牙
	0x03	日本
	0x04~FF	保留

5. Read_BD_ADDR

该命令读出 BD_ADDR 参数值。BD_ADDR 是 48 位蓝牙设备的唯一标识符。当该命令完成时，命令完成事件产生。其指令及其描述如表 10.184 和表 10.185 所示。

表 10.184 Read_BD_ADDR 命令

命 令	OCF	命令参数	返回参数
HCI_Read_BD_ADDR	0x0009		Status, BD_ADDR

表 10.185 Read_BD_ADDR 命令描述

参 数	值	参数说明
Status 1 字节	0x00	Read_BD_ADDR 命令成功
	0x01	Read_BD_ADDR 命令失败
BD_ADDR 6 字节	0XXXXXXXXXXXXX	设备 BD_ADDR

10.3.8 状态参数

主控制器可修改所有状态参数。这些参数（见表 10.186）提供有关主控制器、链路管理器和基带的当前状态信息。主机不能修改这些参数的任何部分，除复位确实指定的参数。对于状态和基带，OGF 定义为 0x05。

表 10.186 状态参数命令一览表

命 令	命令说明汇总
Read_Failed_Contact_Counter	Read_Failed_Contact_Counter 读出对于其余设备特殊链接的 Failed_Contact_Counter 参数值。Failed_Contact_Counter 记录在刷新超时终止及当前正在传输的 L2CAP 数据分组被自动刷新后，主单元或从单元不能响应连续事件次数
Reset_Failed_Contact_Counter	Reset_Failed_Contact_Counter 复位对于其余设备特殊链接的 Failed_Contact_Counter 参数值。Failed_Contact_Counter 记录在刷新超时终止及当前正在传输的 L2CAP 数据分组被自动刷新后，主单元或从单元不能响应连续事件次数
Get_Link_Quality	Get_Link_Quality 命令读出指定链接句柄的 Link_Quality 的值
Read_RSSI	Read_RSSI 命令读出对于其他蓝牙设备链接句柄的场强值

1. Read_Failed_Contact_Counter

该命令读出其余设备特殊链接的 Failed_Contact_Counter 参数，其指令及其描述如表 10.187 和表 10.188 所示。链接句柄必须是 ACL 方式的链接句柄。Failed_Contact_Counter 记录了在刷新超时终止和当前正在传输的 L2CAP 分组被自动刷新后，主单元和从单元没响应的连续事件次数，当该情况出现时，Failed_Contact_Counter 的值增 1。在下列条件里，链接的 Failed_Contact_Counter 复位为“0”。

- 当一个新链接确立时；
- 当 Failed_Contact_Counter 大于“0”和作为链接的 L2CAP 分组被确认；
- 当 Reset_Failed_Contact_Counter 命令发出。

表 10.187 Read_Failed_Contact_Counter 命令

命 令	OCF	命令参数	返回参数
HCI_Read_Failed_Contact_Counter	0x0001	Connection_handle	Status, Connection_handle Failed_Contact_Counter

表 10.188 Read_Failed_Contact_Counter 命令描述

参 数	值	参 数 说 明
Connection_Handle 2 字节(12 位有意义)	0xFFFF	读出哪个 Failed_Contact_Counter 链接的链接句柄 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)
Status 1 字节	0x00	Read_Failed_Contact_Counter 命令成功
	0x01	Read_Failed_Contact_Counter 命令失败
Connection_Handle 2 字节(12 位有意义)	0xFFFF	已读出哪个 Failed_Contact_Counter 链接的链接句柄。 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)
Failed_Contact_Counter 2 字节	0xFFFF	相应链接句柄的链接连续失败的次数

2. Reset_Failed_Contact_Counter

该命令复位其余设备特殊链接的 Failed_Contact_Counter 参数，其指令及其描述如表 10.189 和表 10.190 所示。链接句柄必须 ACL 方式的链接句柄。Failed_Contact_Counter 记录了在刷新超时终止和当前正在传输的 L2CAP 分组被自动刷新后，主单元和从单元没响应的连续事件次数，当该情况出现时，Failed_Contact_Counter 的值增 1。在下列条件里，链接的 Failed_Contact_Counter 复位为“0”。

- 当一个新链接确立时；
- 当 Failed_Contact_Counter 大于“0”和作为链接的 L2CAP 分组被确认；
- 当 Reset_Failed_Contact_Counter 命令发出。

表 10.189 Reset_Failed_Contact_Counter 命令

命 令	OCF	命令参数	返回参数
HCI_Reset_Failed_Contact_Counter	0x0002	Connection_Handle	Connection_Handle, Status

表 10.190 Reset_Failed_Contact_Counter 命令描述

参 数	值	参 数 说 明
Connection_Handle 2 字节(12 位有意义)	0xFFFF	复位哪个 Failed_Contact_Counter 链接的链接句柄 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)
Status 1 字节	0x00	Reset_Failed_Contact_Counter 命令成功
	0x01	Reset_Failed_Contact_Counter 命令失败
Connection_Handle 2 字节(12 位有意义)	0xFFFF	已复位哪个 Failed_Contact_Counter 链接的链接句柄 范围: 0x0000-0x0EFF (0x0F00 ~ 0x0FFF 保留)

3. Get_Link_Quality

该命令返回指定链接句柄的 Link_Quality 值。链接句柄必须是 ACL 链接方式的链接句柄。该命令将返回在两个蓝牙设备之间表示的链接质量的从 0 ~ 255 的 Link_Quality 值。该值越高，链接质量就越好。各蓝牙模型供应商将决定怎样测量链接质量。其指令及其描述如表 10.191 和表 10.192 所示。

表 10.191 Get_Link_Quality 命令

命 令	OCF	命令参数	返回参数
HCI_Get_Link_Quality	0x0003	Connection_Handle	Status, Connection_Handle, Link_Quality

表 10.192 Get_Link_Quality 命令描述

参 数	值	参 数 说 明
Connection_Handle 2 字节(12 位有意义)	0xFFFF	读出哪种链接质量参数链接的链接句柄 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)
Status 1 字节	0x00	Get_Link_Quality 命令成功
	0x01	Get_Link_Quality 命令失败
Connection_Handle 2 字节(12 位有意义)	0xFFFF	已读出哪种链接质量参数链接的链接句柄 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)
Link_Quality 1 字节	0xFF	在本地设备和通过链接句柄指出的远程设备之间, 链接的当前质量。范围: 0x00 ~ 0xFF。值越高, 链接质量就越好

4. Read_RSSI

对于其他蓝牙设备，在测量场强和最佳接收电平区段限制之间，该命令读出的不同值。链接句柄必须是 ACL 链接方式的链接句柄。通过主控制器返回的任何 RSSI 正值指出超过 RSSI 上限的多少 dB，任何 RSSI 负值指出低于 RSSI 下限的多少 dB。值“0”指出在最佳设备功率区段内的 RSSI。其指令及其描述如表 10.193 和表 10.194 所示。

注意：dB 值精确度取决于蓝牙硬件。对硬件的惟一要求是蓝牙设备能判定 RSSI 是否在最佳接收功率范围内、上限或下限。

表 10.193 Read_RSSI 命令

命 令	OCF	命令参数	返回参数
HCI_Read_RSSI	0x0005	链接句柄	Status, 链接句柄, RSSI

表 10.194 Read_RSSI 命令描述

参 数	值	参 数 说 明
Connection_Handle 2 字节(12 位有意义)	0xFFFF	读出哪个 RSSI 链接的链接句柄 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留)
Status 1 字节	0x00	Read_RSSI 命令成功
	0x01	Read_RSSI 命令失败

续表

参 数	值	参 数 说 明
RSSI 1 字节	N = 0xXX	长度: 1 字节(带符号整数); 范围: -128 ~ -127 单位: dB
Connection_Handle 2 字节(12 位有意义)	0xFFFF	已读出哪个 RSSI 链接的链接句柄 范围: 0x0000 ~ 0x0FFF (0x0F00 ~ 0x0FFF 保留)

10.3.9 测试指令

测试指令（见表 10.195）用于提供测试蓝牙硬件不同功能的能力，并为测试提供安排不同条件的能力。对于测试指令，OGF 定义为 0x06。

表 10.195

指 令	指 令 综 述
Read_Loopback_Mode	Read_Loopback_Mode 将读取主控制器回送模式的设置值 回送模式设置可以确定信息发送路径
Write_Loopback_Mode	Write_Loopback_Mode 将写入主控制器回送模式的设置 值。回送模式设置可以确定信息发送路径
Enable_Device_Under_Test_Mode	Enable_Device_Under_Test_Mode 指令允许本地蓝牙模块通 过 LMP 测试指令进入测试模式。当主请求本地设备作 为待测试设备，实现蓝牙测试模式文件中规定测试情景时， 则发送该指令

1. Read_loopback_Mode

本指令将读取主控制器回送模式参数值，其指令及其描述如表 10.196 和表 10.197 所示。回送模式可以确定信息发送路径。在非测试模式操作中，回送模式设置为非测试模式，而其信息路径则由蓝牙规范指定。在本地回送模式中，每一数据分组(ACL 和 SCO)和从主机发送到主控制器的指令分组，也将由主控制器不加任何改变地返回。当蓝牙主控制器进入本地回送模式时，它可以以四种链接完成事件应答，其中一种用于 ACL 通道，三种用于 SCO 通道，以便当发送 ACL 和 SCO 数据时，主机能够获取链接句柄。当处于本地回送模式时，主控制器将向主机回送指令和数据。回送指令事件用于主机向主控制器发送回送指令。

表 10.196 Read_loopback_Mode 命令

指 令	OCF	指令参数	返 回 参 数
HCI_Read_loopback_Mode	0x0001		Status, Loopback_Mode

在本地回送模式中有一些指令不会被回送，包括 Reset, Host_Buffer_Size, Set_Host_Controller_To_host_Flow_Control, Host_Number_Of_Completed_Packets, Read_Buffer_Size, Read_loopback_Mode 和 Write_loopback_Mode。指令 Reset 和 Write_loopback_Mode 可用于退出本地回送模式。如果 Write_loopback_Mode 用于退出本地回送模式，则向主机发送四种链接断开完成事件，这四种事件对应于进入本地回送模式时发送的链接完成事件。而且，本地回送模式不得允许任何链接。如果存在一个链接，且存在设备进入本地回送模式的尝试，则主控制器将拒绝呼入链接尝试。这将不允许使用其他变量对主控制器传输层进行测试

试。如果一设备设置为远程回送模式，它将无线发回所有数据（ACL 和 SCO）。它最大可允许同时保持一条 ACL 链接和三条 SCO 链接。而这与远程设备相同。如果存在不止一条指向远程设备的链接，并且存在设置本地设备为远程回送模式的尝试，该尝试将被拒绝。可以不使用任何其他变量测试蓝牙无线链路。

表 10.197 Read_loopback_Mode 命令描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_loopback_Mode 指令成功
	0x01	Read_loopback_Mode 指令失败
Loopback_Mode 1 字节	0x00	未启用回送模式，缺省
	0x01	启用本地回送
	0x02	启用远程回送
	0x03-0xFF	保留

2. Write_Loopback_Mode

该命令将写入主控制器回送模式的设置值，其指令及其描述如表 10.198 和表 10.199 所示。回送模式设置可以确定信息发送路径。在非测试模式操作中，回送模式设置为非测试模式，而其信息路径则由蓝牙规范指定。在本地回送模式中，每一数据分组(ACL 和 SCO)和从主机发送到主控制器的指令分组，也将由主控制器不加任何改变地返回。当蓝牙主控制器进入本地回送模式时，它可以以四种链接完成事件应答，其中一种用于 ACL 通道，三种用于 SCO 通道，以便当发送 ACL 和 SCO 数据时，主机能够获取链接句柄。当处于本地回送模式时，主控制器将向主机回送指令和数据。

表 10.198 Write_Loopback_Mode 命令

指 令	OCF	指令参数	返回参数
HCI_Write_loopback_Mode	0x0002	Loopback_Mode	Status

在本地回送模式中有一些指令将不会被回送，包括 Reset，Host_Buffer_Size，Set_Host_Controller_To_host_Flow_Control，Host_Number_Of_Completed_Packets，Read_Buffer_Size，Read_loopback_Mode 和 Write_loopback_Mode。这些指令可以以常规执行方式执行。指令 Reset 和 Write_loopback_Mode 可用于退出本地回送模式。如果 Write_loopback_Mode 用于退出本地回送模式，则可向主机发送四种链接断开完成事件，以对应于进入本地回送模式时的链接完成事件。而且，本地回送模式不得允许任何链接。如果存在一个链接，且存在设备进入本地回送模式的尝试，则主控制器将拒绝呼入链接尝试。这将不允许使用其他变量对主控制器传输层进行测试。如果一设备设置为远程回送模式，它将无线发回所有数据（ACL 和 SCO）。它也最大可允许同时保持一条 ACL 链接和三条 SCO 链接。而这与远程设备相同。如果存在不止一条指向远程设备的链接，并且存在设置本地设备为远程回送模式的尝试，而该尝试将被拒绝。可以不使用任何其他变量测试蓝牙无线链路。

表 10.199 Write_Loopback_Mode 命令描述

参 数	值	参 数 说 明
Loopback_Mode 1 字节	0x00	未启用回送模式
	0x01	启用本地回送
	0x02	启用远程回送
	0x03-0xFF	保留
Status 1 字节	0x00	Write_loopback_Mode 指令成功
	0x01	Write_loopback_Mode 指令失败

3. Enable_Device_Under_Test_Mode

该指令将允许本地蓝牙模块通过 LMP 测试指令进入测试模式，其指令及其描述如表 10.200 和表 10.201 所示。当主机要求本地设备成为 DUT，并进入蓝牙测试模式中的测试情景时，主机将发送该指令。当主控制器收到该指令时，它将通过指令完成事件完成该指令。主控制器将正常操作，直至远程测试装置发出 LMP 测试指令将本地设备进入测试模式。为了终止并退出测试模式，主机将发送 HCI_Reset 指令。该指令将阻止远端蓝牙设备不先发出该指令就将本地蓝牙设备置为测试模式。

表 10.200 Enable_Device_Under_Test_Mode 命令

指 令	OCF	指令参数	返回参数
HCI_Enable_Device_Under_Test_Mode	0x0003		Status

表 10.201 Enable_Device_Under_Test_Mode 命令描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Enable_Device_Under_Test_Mode 指令成功
	0x01	Enable_Device_Under_Test_Mode 指令失败

10.4 事件

10.4.1 事件

除表 10.202 列出的事件以外，事件代码 0xFF 将保留作为厂商调试事件的事件代码，事件代码 0xFE 保留用于蓝牙标识测试。

表 10.202 支持事件列表

事 件	事 件 总 述
查询完成事件	查询完成事件表示查询已完成
查询结果事件	查询结果事件表示在当前查询进程中已有一个或多个蓝牙设备应答
链接完成事件	链接完成事件指示构成链接的两主机已建立一个新的链接。
链接请求事件	链接请求事件用于表示正在建立一个新的呼入链接
链接断开完成事件	链接断开完成事件当链接终止时发生

续表

事 件	事 件 总 述
鉴权完成事件	鉴权完成事件当指定链接鉴权完成时发生
远程命名请求事件	远程命名请求事件用于表示远程命名请求已完成。Remote_Name 事件参数为一长度可达 248 字节的 UTF-8 编码字符串
加密改变事件	加密改变事件用于表示对于由 Connection_Handle 事件参数指定链接句柄已完成加密改变
链接链接字改变完成事件	链接链接字改变完成事件用于表示由 Connection_Handle 事件参数指定链接句柄的链接字改变已完成
主单元链接字完成事件	主单元链接字完成事件用于表示蓝牙主单元一方的临时链接字或半永久链接字改变已完成
远端支持特性读取完成事件	远端支持特性读取完成事件用于表示链路管理器进程已完成，该链路管理器包含由 Connection_Handle 事件参数指定远程蓝牙设备支持的特性
远程版本信息读取完成事件	远程版本信息读取完成事件用于表示链路管理器进程已完成，该链路管理器包含由 Connection_Handle 事件参数指定远程蓝牙设备的版本信息
QoS 启用完成事件	QoS 启用完成事件用于表示启用 QoS 的链路管理器进程已完成，该过程由 Connection_Handle 事件参数指定远程蓝牙设备完成
指令完成事件	指令完成事件由主控制器用于为每一 HCI 指令传递指令返回状态和其他事件参数
指令状态事件	指令状态事件用于表示已收到 Command_Opcode 参数所描述指令，且主控制器正在执行该指令任务
硬件故障事件	该事件用于表示蓝牙设备硬件故障类别
刷新事件	该事件用于表示对于指定链接句柄，要传输的当前用户数据已删除
角色改变事件	角色改变事件用于表示与特定链接相关的当前蓝牙角色已改变
完成分组数事件	完成分组数事件由主控制器用于通知主机自前一完成分组数事件发送后，对于每一链接句柄已完成了多少 HCI 数据分组
模式改变事件	模式改变事件用于指示与链接句柄相关的设备何时在激活、挂起、呼吸和休眠模式间变化
返回链接字事件	返回链接字事件用于在使用 Read_Stored_Link_Key 指令后，返回保存的链接字
PIN 码请求事件	PIN 码请求事件用于表示需要一个 PIN 码以创建链接的新链接字
链接字请求事件	链接字请求事件用于表示需要一链接字以建立与 BD_ADDR 指定设备的链接
链接字通知事件	链接字通知事件用于通知主机与 BD_ADDR 指定设备链接的链接字已创建
回送指令事件	回送指令事件用于回送大多数主机发往主控制器的指令
数据缓冲区溢出事件	数据缓冲区溢出事件用于表示由于主机发出分组数超出允许数量，主控制器数据缓冲区已溢出
时隙读取完成事件	时隙读取完成事件用于表示包含时隙信息的 LM 进程已完成
链接分组类型改变事件	链接分组类型改变事件用于表示改变 Connection_Handle 所指定分组类型的链路管理器进程已完成
违反 QoS 事件	违反 QoS 事件用于表示链路管理器不能提供链接句柄的当前 QoS 要求
呼叫扫描模式改变事件	呼叫扫描模式改变事件表示采用指定 Connection_Handle 链接的远程蓝牙设备已成功改变 Page_Scan_Mode

续表

事 件	事 件 总 述
呼叫扫描竞争模式改变事件	呼叫扫描竞争模式改变事件表示采用指定 Connection_Handle 链接的远程蓝牙设备已成功改变 Page_Scan_Repetition_Mode
最大时隙改变事件	该事件用于在 LMP_Max_Slots 值改变时将该值通知主机

10.4.2 事件说明

这些事件提供返回参数和与每一事件有关数据的方法。

1. 查询完成事件

查询完成事件表示查询结束。该事件包含一个状态参数，该参数用于表示查询成功完成与否。另外，Num_Responses 参数包含在最近一次查询中应答的蓝牙设备的数量。查询完成事件及其描述如表 10.203 和 10.204 所示。

表 10.203 查询完成事件

事 件	OCF	事 件 代 码
Inquiry complete	0x01	Status: Num_Responses

表 10.204 查询完成事件描述

参 数	值	参 数 说 明
Status 1 字节	0x00	查询指令成功完成
	0x01	查询指令失败
Num_Responses 1 字节	0xXX	查询的应答数

2. 查询结果事件

查询结果事件表示在当前查询进程中已有一个或多个蓝牙设备应答，查询结果事件及其描述如表 10.205 和 10.206 所示。如果远程设备只支持强制呼叫方案，则一旦从远程设备收到一查询应答，主控制器将向主机发送该事件。主控制器可将查询回答排队，并在一个查询结果事件中发送多个蓝牙设备信息。该事件可用于在一个事件中返回一个或多个查询应答。该事件包括对应于应答上一次查询的蓝牙设备的 BD_ADDR，Page_Scan_Repetition_Mode, Page_Scan_Period_Mode, Page_Scan_Mode, Clock_Offset 和 Class of Device。

表 10.205 查询结果事件

事 件	OCF	事 件 代 码
查询结果事件	0x02	Num_Responses, BD_ADDR Page_Scan_Repetition_Mode, Page_Scan_Period_Mode, Page_Scan_Mode, Class_of_device, Clock_Offset

表 10.206 查询结果事件描述

参 数	值	参 数 说 明
Num_Responses 1 字节	0xXX	查询的应答数
BD_ADDR [1] 6 个字节*	0XXXXXXXXXXXXX	每一应答设备的 BD_ADDR
Page_Scan_Repetition_Mode [1] 1 字节*	0x00	R0
	0x01	R1
	0x02	R2
	0x03-0xFF	保留
Page_Scan_Period_Mode [1] 1 字节	0x00	P0
	0x01	P1
	0x02	P2
	0x03-0xFF	保留
Page_Scan_Mode 1 字节	0x00	强制呼叫扫描
	0x01	可选呼叫扫描模式 I
	0x02	可选呼叫扫描模式 II
	0x03	可选呼叫扫描模式 III
	0x04-0xFF	保留
Class_of_Device 3 字节	0XXXXXX	设备类型
Clock_offset 1 字节	14.0 位	16.2 位的 CLKslave-CLKmaster
	第 15 位	保留

3. 链接完成事件

链接完成事件指示构成链接的两个主机已建立一个新的链接。该事件也可通知发送 Create_Connection 或 Add_SCO_Connection 或 Accept_Connection_Request 或 Reject_Connection 指令的主机, 并接收表示指令是否成功完成的指令状态事件。该事件及其描述如表 10.207 和 10.208 所示。

表 10.207 链接完成事件

事 件	OCF	事件代码
链接完成事件	0x03	Status: Connection_handle BD_ADDR: Link_Type: Encryption_mode

表 10.208 链接完成事件描述

参 数	值	参 数 说 明
Status 1 字节	0x00	链接成功完成
	0x01	链接未完成, 参见表 6.1
Connection_Handle 2 个字节(其中 12 位有意义)	0XXXXX	链接句柄用于识别蓝牙设备间链接。链接柄也可用作发送和接收语音或数据的标识符。 范围: 0x0000 ~ 0x0EFF (0x0F00 ~ 0x0FFF 保留使用)
BD_ADDR 6 字节	0XXXXXXXXXXXXX	构成链接的另一链接设备的 BD_ADDR
Link_Type 1 字节	0x00	SCO 链接(语音通道)

续表

参 数	值	参 数 说 明
Encryption_Mod 1 字节	0x01	ACL 链接(数据通道)
	0x02-0xFF	保留使用
	0x00	加密停止
	0x01	只对点对点分组加密
	0x02	对点对点和广播分组加密
	0x03 ~ 0xFF	保留

4. 链接请求事件

链接请求事件用于表示尝试建立一个新的链接。链接可被接受，也可被拒绝。如果该事件被屏蔽，并且存在一呼入链接尝试，主控制器将自动拒绝该链接尝试。当主机收到该事件时，它将在 Conn_Accept_Timeout 定时器失效前，以 Accept_Connection_Request 或 Reject_Connection_Request 指令应答。该事件及其描述如表 10.209 和 10.210 所示。

表 10.209 链接请求事件

事 件	OCF	事 件 代 码
Connection request	0x04	BD-ADDR; Class_of_device; Link_Type

表 10.210 链接请求事件描述

参 数	值	参 数 说 明
BD_ADDR 6 字节	0xFFFFFFFFXXXXXX	请求链接设备的 BD_ADDR
Class_of_Device 3 字节	0xFFFFFFFF	请求链接设备的设备类型
Link_Type 1 字节	0x00	SCO 链接(语音通道)
	0x01	ACL 链接(数据通道)
	0x02-0xFF	保留使用

5. 链接断开完成事件

链接断开完成事件当链接终止时发生。状态参数表示链接断开是否成功。如果链接断开成功，则原因参数表示链接断开原因。如果链接断开失败，主机将忽略原因参数值。例如，如果主机已发出链接断开指令，并存在参数错误，则不允许该指令执行，或给出不能应答链接的链接句柄。该事件及其描述如表 10.211 和 10.212 所示。

注意：当物理链路失败时，将在物理链路上为每一逻辑通道返回链接断开完成事件，相应链接句柄作为其参数。

表 10.211 链接断开完成事件

事 件	OCF	事 件 代 码
Disconnection Complete	0x05	Status; Connection_handle; Reason

表 10.212 链接断开完成事件描述

参 数	值	参 数 说 明
Status 1 字节	0x00	链接断开完成
	0x01	链接断开未完成
Connection_Handle 2 字节(其中 12 位有意义)	0xFFFF	断开链接句柄 范围: 0x0000~0x0EFF(0x0F00~0x0FFF 保留)
Reason 1 字节	0x08 , 0x13~ 0x16 , 0x1A	链接超时(0x08), 其他终端链接终止错误代码 (0x13~0x15), 由本地主机终止的链接 (0x16), 以及未支持的远端特性错误代码(0x1A)

6. 鉴权完成事件

当指定链接的鉴权已完成时, 鉴权完成事件发生。Connection_Handle 是 ACL 链接的 Connection_Handle。该事件及其描述如表 10.213 和 10.214 所示。

表 10.213 鉴权完成事件

事 件	OCF	事 件 代 码
鉴权完成事件	0x06	Status, : Connection_Handle

表 10.214 鉴权完成事件描述

参 数	值	参 数 说 明
Status 1 字节	0x00	鉴权请求成功完成
	0x01	鉴权请求未完成
Connection_Handle 2 个字节(其中 12 位有意义)	0xFFFF	执行鉴权的链接句柄 范围: 0x0000 ~ 0x0EFF(0x0F00~0x0FFF 保留)

7. 远程命名请求完成事件

远程命名请求事件用于表示远程命名请求已完成。Remote_Name 事件参数为一长度可达 248 字节的 UTF-8 编码字符串。如果 UTF-8 编码字符串不到 248 个字节, 则 Remote_Name 事件参数尾段用空值(0x00)填充。BD_ADDR 事件参数则用于标识获取名字的设备。该事件及其描述如表 10.215 和 10.216 所示。

注意: Remote_Name 参数从名字的第一字节开始接收。

表 10.215 远程命名请求完成事件

事 件	OCF	事 件 代 码
远程命名请求完成事件	0x07	Status; BD_ADDR; Remote_name

表 10.216 远程命名请求完成事件描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Remote_Name_Request 指令成功
	0x01	Remote_Name_Request 指令失败

续表

参 数	值	参 数 说 明
BD_ADDR 6 字节	0XXXXXXX XXXXX	被请求设备的 BD_ADDR
Remote_Name 248 字节	名称[248]	远程设备的 UTF-8 编码的描述性名字 UTF-8 编码名字可长达 248 字节。如果它短 于 248 字节，则用 0x00 进行填充。

8. 加密模式改变事件

该事件用于表示已完成由 Connection_Handle 事件参数指定链接句柄的加密模式改变。Connection_Handle 为 ACL 链接的 Connection_Handle。Encryption_Enable 事件参数指定由 Connection_Handle 指定的链接句柄启用新的加密模式。该事件将在链接两端设备上发生，以通知两主机加密模式已改变。该事件及其描述如表 10.217 和 10.218 所示。

表 10.217 加密模式改变事件

事 件	OCF	事 件 代 码
加密模式改变事件	0x08	Status; Connection_handle; Encryption_enable

表 10.218 加密模式改变事件描述

参 数	值	参 数 说 明
Status 1 字节	0x00	加密模式改变失败
	0x01	加密模式改变成功
Connection_Handle 2 字节 (其中 12 位有意义)	0XXXXX	同一蓝牙设备终端的所有链接句柄中启用或 终止链路层加密的链接句柄。范围: 0x0000 ~ 0x0EFFF(0x0F00 ~ 0x0FFF 保留)
Encryption_Enable 1 字节	0x00	停用链路层加密
	0x01	启用链路层加密

9. 链接链接字改变完成事件

该事件用于表示由 Connection_Handle 事件参数指定的链接句柄的链接字改变已完成。Connection_Handle 为 ACL 链接的 Connection_Handle。该事件只发往发送 Change_Connection_Link_Key 指令的主机。该事件及其描述如表 10.219 和 10.220 所示。

表 10.219 链接链接字改变完成事件

事 件	OCF	事 件 代 码
链接链接字改变完成事件	0x09	Status; Connection_handle

表 10.220 链接链接字改变完成事件描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Change_Connection_Link_Key 指令成功
	0x01	Change_Connection_Link_Key 指令失败

续表

参 数	值	参 数 说 明
Connection_Handle 2 字节(其中 12 位有意义)	0xFFFF	同一蓝牙设备终端的所有链接句柄中改变链接字的链接句柄 范围: 0x0000 ~ 0x0EFF(0x0F00-0x0FFF 保留)

10. 主单元链接字完成事件

主单元链接字完成事件用于表示匹克网蓝牙主单元的临时链接字或半永久链接字改变已完成。Connection_Handle 为 ACL 链接句柄。链接所使用的链接字将为主设备的临时链接字, 或由 Key_Flag 表示的半永久链接字。Key_Flag 事件参数用于表示当前在匹克网中使用的是哪个链接字(主单元临时链接字或半永久链接字)。该事件及其描述如表 10.221 和 10.222 所示。

注意, 对于一主单元, 从临时链接字到半永久链接字的变化将影响所有与匹克网有关的所有链接句柄。对于一从单元, 此变化将仅仅影响某指定链接句柄。当广播和点对点通信都需要加密时, 则必须使用临时链接字。

表 10.221 主单元链接字完成事件

事 件	OCF	事 件 代 码
主单元链接字完成事件	0x0A	Status; Connection_handle; Key_flag

表 10.222 主单元链接字完成事件描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Master_Link_Key 指令成功
	0x01	Master_Link_Key 指令失败
Connection_Handle 2 字节(其中 12 位有意义)	0xFFFF	对于同一匹克网中所有设备, 链接字已改变的链接句柄 范围: 0x0000 ~ 0x0EFF(0x0F00-0x0FFF 保留)
Key_Flag 1 字节	0x00	使用半永久链接字
	0x01	使用临时链接字

11. 远程支持特性读取完成事件

该事件用于表示获取由 Connection_Handle 指定远程蓝牙设备支持特性的链路管理器进程的结束。Connection_Handle 为 ACL 链接的链接句柄。事件参数则包括 LMP 特性表。该事件及其描述如表 10.223 和 10.224 所示。

表 10.223 远程支持特性读取完成事件

事 件	OCF	事 件 代 码
远程支持特性读取完成事件	0x0b	Status; Connection_handle; LMP_features

表 10.224 远程支持特性读取完成事件描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_Remote_Supported_Features 指令成功
	0x01	Read_Remote_Supported_Features 指令失败
Connection_Handle 2 字节(其中 12 位有意义)	0xFFFF	链接句柄用于 Read_Remote_Supported_Features 指令，范围：0x0000~0x0EFF (0x0F00~0x0FFF 保留)
LMP_Features 8 字节	0xFFFFFFFF XXXXXXXX	LMP 屏蔽位列表，参见“链路管理器协议”

12. 远程版本信息读取完成事件

该事件表示用于获取远程蓝牙设备版本信息的链路管理器进程的结束。该版本信息由 Connection_Handle 指定。Connection_Handle 为一 ACL 链接的链接句柄。LMP_Version 事件参数定义蓝牙硬件的主要硬件方案。只有符合新蓝牙 SIG 规范的蓝牙新硬件版本出现时，该事件参数才变化，也就是说，该事件参数由 SIG 控制。Manufacturer_name 参数表示远程蓝牙模块制造商。LMP_Subversion 事件参数应由制造商控制，并且可根据需要改变。LMP_Subversion 事件参数定义了当设计进程变化和错误得到修改时，蓝牙硬件的每一修订版本。该事件允许软件确定正在使用哪种蓝牙硬件，并且如果必要，将能够在硬件出错的环境下工作。该事件及其描述如表 10.225 和 10.226 所示。

表 10.225 远程版本信息读取完成事件

事 件	OCF	事 件 代 码
远程版本信息读取完成事件	0x0c	Status: Connection_handle, LMP_version: Manufacturer_name, LMP_subversion

表 10.226 远程版本信息读取完成事件描述

参 数	值	参 数 说 明
Status 1 字节	0x00	Read_Remote_Version_Information 指令成功。
	0x01	Read_Remote_Version_Information 指令失败。
Connection_Handle 2 字节(其中 12 位有意义)	0xFFFF	链接句柄用于 Read_Remote_Version_Information 指令 范围：0x0000~0x0EFF(0x0F00 ~ 0x0FFF)保留)
LMP_Version 1 字节	0xFF	远程蓝牙硬件当前 LMP 版本
Manufacturer_name 2 字节	0xFFFF	远程蓝牙硬件制造商名称
LMP_Subversion 2 字节	0xFFFF	远程蓝牙硬件的当前 LMP 子版本

13. QoS 设置完成事件

该事件表示由 Connection_handle 事件参数指定的远程蓝牙设备 QoS 的链路管理器设置进程结束。Connection_handle 为 ACL 链接的链接句柄。该事件及其描述如表 10.227 和 10.228 所示。

表 10.227 QoS 设置完成事件

事 件	OCF	事 件 代 码
QoS 设置完成事件	0x0d	Status: Connection_handle, Flags: Service_type Token_rate: Peak_bandwidth: Latency: Delay_variation

表 10.228 QoS 设置完成事件描述

参 数	值	参 数 说 明
Status 1 字节	0x00	QoS_Setup 指令成功
	0x01	QoS_Setup 指令失败
Conection_handle 2 字节 (其中 12 位有意义)	0xXXXX	链接句柄用于 Qos_setup 指令 范围: 0x0000~0x0EFF(0x0F00~0x0FFF 保留)
Flags 1 字节	0x00~0xFF	保留
Service_Type 1 字节	0x00	无可用通信
	0x01	允许最大传输能力
	0x02	可用授权
	0x03~0xFF	保留
Token_Rate 4 字节	0xFFFFFFFF	允许 Token_Rate, 单位为字节/秒
Peak_Bandwidth 4 字节	0xFFFFFFFF	允许峰值带宽
Latency 4 字节	0xFFFFFFFF	允许延时, 单位微秒
Delay_Variation 4 字节	0xFFFFFFFF	允许延迟值, 单位微秒

14. 指令完成事件

该事件由主控制器用于传递大多数指令返回状态, 以及其他已发出 HCI 指令的事件参数。Num_HCI_Command_Packets 事件参数允许主控制器指出主机可发往主控制器 HCI 指令分组个数。如果主控制器要求停止发送指令, Num_HCI_Command_Packets 事件参数将设置为零。为了通知主机主控制器已准备好接收 HCI 指令分组, 主控制器将生成 Command_Opcode 为 0x0000 的指令完成事件, 并且 Num_HCI_Command_Packets 事件参数将设置为 1 或更大值。Command_Opcode (0x0000) 为 NOP(空操作), 并且用于改变主机进入等待状态前可发送的 HCI 指令分组的个数。该事件及其描述如表 10.229 和 10.230 所示。

表 10.229 指令完成事件

事 件	OCF	事 件 代 码
指令完成事件	0x0E	Num_HCI_command_packets: Command_opcode: Return_parameters

表 10.230 指令完成事件描述

参 数	值	参 数 说 明
Num_HCI_Command_Packets 1 字节	N=0xxx	可从 host 发往主控制器的 HCI 指令分组个数 N 取值范围: 0 ~ 255

续表

参 数	值	参 数 说 明
Command_Opcode 2 字节	0xFFFF	可引发该事件的指令的操作码
Return_Parameter 取决于代码 事件参数指令完全的 0x0E	0xFF	Command_Opcode 事件参数指定指令的返回参数, 参见与该指令相关的返回参数列表

15. 指令状态事件

该事件表示已收到由 Command_Opcode 参数描述的指令, 且主控制器正在执行该指令任务。该事件及其描述如表 10.231 和 10.232 所示。该事件需提供异步操作机制, 以防止主机一直处于等待指令完成的状态。如果指令不能执行(可能由于参数错误或指令不允许), 指令状态事件参数将包含相应出错代码, 并且由于指令未执行则不会生成该指令完成事件。Num_HCI_Command_Packets 事件参数允许主控制器表示主机能发送至主控制器的 HCI 指令分组个数。如果主控制器要求停止发送指令, Num_HCI_Command_Packets 事件参数将置为 0。为了通知主机主控制器已准备好接收 HCI 指令分组, 主控制器将生成状态为 0x00、Command_Opcode 为 0x0000, 以及 Num_HCI_Command_Packets 事件参数为 1 或更大值的指令状态事件。Command_Opcode (0x0000) 为 NOP(空操作), 并且用于改变主机进入等待状态前可接收 HCI 指令分组的个数。

表 10.231 指令状态事件

事 件	OCF	事 件 代 码
指令状态事件	0x0F	Status: Num_HCI_command_packets: Command_Opcode

表 10.232 指令状态事件描述

	值	参 数 说 明
Status 1 字节	0x00	指令当前正在执行
	0x01	指令失败, 参见出错代码列表
Num_HCI_Command_Packets 1 字节	N=0xFF	允许从主机发往主控制器的 HCI 指令分组个数 N 取值范围: 0~255
Command_Opcode 2 字节	0xFFFF	引发该事件, 且正在执行指令的操作码

16. 硬件故障事件

硬件故障事件用于表示蓝牙设备的硬件故障类型。该事件也用于通知主机蓝牙模块已发生故障。该事件及其描述如表 10.233 和 10.234 所示。

表 10.233 硬件故障事件

事 件	OCF	事 件 代 码
硬件故障事件	0x10	Hardware_code

表 10.234 硬件故障事件描述

参 数	值	参 数 说 明
Hardware_Code 1 字节	0x00	Hardware_Codes 因不同实现而定，且可指定为不同硬件故障

17. 刷新事件

刷新事件用于表示对于特定链接句柄，将用于传输的当前用户数据删除。Connection_handle 为 ACL 链接的链接句柄。在主控制器中，一个 L2CAP 分组的多个数据块已待定。如果 L2CAP 分组的基带分组部分被刷新，则 L2CAP 分组的 HCI 数据分组的其余部分也必须刷新。该事件及其描述如表 10.235 和 10.236 所示。

表 10.235 刷新事件

事 件	OCF	事 件 代 码
刷新事件	0x11	Connection_handle

表 10.236 刷新事件描述

参 数	值	参 数 说 明
Connection_handle 2 字节(其中 12 位有意义)	0xFFFF	已刷新链接句柄 范围: 0x0000 ~ 0x0FFF(0x0F00~0x0FFF 保留)

18. 角色变化事件

角色变化事件用于表示与特定链接有关的当前蓝牙角色已改变。只有当与 BD_ADDR 事件参数相关的远端和本地蓝牙设备已完成其角色变化时，该事件才发生。当角色改变后，该事件将通知链接两端设备。该事件及其描述如表 10.237 和 10.238 所示。

表 10.237 角色变化事件

事 件	OCF	事 件 代 码
角色变化事件	0x12	Status; BD_ADDR; New_Role

表 10.238 角色变化事件描述

参 数	值	参 数 说 明
状态 1 字节	0x00	发生角色变化
	0x01	角色变化失败，参见 745 页表 6.1 出错代码列表
BD_ADDR	0xFFFFFFFFXXXX	角色改变设备的 BD_ADDR
New_Role 1 字节	0x00	对应于指定 BD_ADDR 的主单元
	0x01	对应于指定 BD_ADDR 的从单元

19. 完成分组数事件

该事件由主控制器用于通知主机自从前一完成分组数事件发往主机后，对于每一链接句柄，已完成(发送或刷新)传输的 HCI 数据分组个数。也就意味着，主控制器相应缓冲区

空间也已释放。基于该信息，以及 Read_Buffer_Size 指令的返回参数 HCI_Total_Num_ACL_Data_Packets 和 HCI_Total_Num_SCO_Data_Packets，主机就能确定以后的 HCI 数据分组将使用何链接句柄发往主控制器。在相应链接完成事件发生前，不应发送该事件。当主控制器在其缓冲区缓存 HCI 数据分组时，它必须向主机周期性持续发送完成分组数事件，直到它最终报告所有待处理 ACL 数据分组都已发送或刷新为止。事件及其发送速率由制造商指定。该事件及其描述如表 10.239 和 10.240 所示。

注意：如果停用 SCO 流控制，将不能报告对应于 SCO 链接句柄的完成分组数事件。

表 10.239 完成分组数事件

事 件	OCF	事 件 代 码
完成分组数事件	0x13	Number_of_handles: Connection_handle HCI_num_of_completed_packets

表 10.240 完成分组数事件描述

参 数	值	参 数 说 明
Number_of_Handles 1 字节	0xXX	本事件包含的 Num_HCI_Data_Packets 参数对和链接句柄数量。范围：0 ~ 255
Connection_handle[] Number_of_Handles*2 字节(其中 12 位有意义)	0xFFFF	链接句柄 范围：0x0000 ~ 0x0EFF(0x0F00 ~ 0x0FFF 保留)
HCI_Num_Of_Completed_Packets 2 字节	N = 0xFFFF	自从前一事件返回后，对应于相关链接句柄的已完成(发送或刷新)的 HCI 数据分组数 N 取值范围：0x0000 ~ 0xFFFF

20. 模式变化事件

该事件用于表示与特定链接句柄相关联的设备在激活、保持、呼吸和休眠模式之间何时发生变化。Current_mode 为 ACL 链接的链接句柄。Current_mode 事件参数用于表示模式变化事件相对于哪一个链接句柄发生。该参数也可用于表示链接处于哪一状态。Interval 参数则用于指定每一状态的持续时间。每一与已发生模式变化的链接句柄相关联的主控制器都将发送一模式变化事件到主机。该事件及其描述如表 10.241 和 10.242 所示。

表 10.241 模式变化事件

事 件	OCF	事 件 代 码
模式变化事件	0x14	Status, Connection_handle, Current_mode, Interval

表 10.242 模式变化事件描述

参 数	值	参数说明
Status 1 字节	0x00	发生模式变化事件
	0x01~0xFF	Hold_Mode、Sniff_Mode、Exit_Sniff_Mode、Park_Mode，或 Exit_Park_Mode 指令失败
Connection_handle 2 字节(其中 12 位有意义)	0xFFFF	链接句柄 范围：0x0000 ~ 0x0EFF(0x0F00 ~ 0x0FFF 保留)

续表

参 数	值	参数说明
Current_Mode 1 字节	0x00	激活模式
	0x01	保持模式
	0x02	呼吸模式
	0x03	休眠模式
	0x04~0xFF	保留
间隔 2 个字节	N = 0xFFFF	保持：保持模式等待的基带时隙数 保持间隔= $N * 0.625 \text{ ms}$ N 取值范围：0x0000~0xFFFF， 时间范围：0 ~ 40.9 秒 呼吸：呼吸间隔之间的基带时隙数 呼吸间隔之间的时间= 0.625 ms N 取值范围：0x0000~0xFFFF， 时间范围：0 ~ 40.9 秒 休眠：连续信号灯之间的基带时隙数。 间隔时间= $N * 0.625 \text{ ms}$ N 取值范围：0x0000 ~ 0xFFFF， 时间范围：0 ~ 40.9 秒

21. 链接字返回事件

该事件由主控制器用于向主机发送一个或多个存储链接字。该事件及其描述如表 10.243 和 10.244 所示。在 Read_Stored_Link_Key 指令发出以后，将不发生或发生多个事件实例。当无存储链接字时，将不返回链接字事件。当有存储链接字时，在每一链接字返回事件中返回的链接字数量将根据具体实现而定。

表 10.243 链接字返回事件

事 件	OCF	事件代码
链接字返回事件	0x15	Num_keys; BD_ADDR; Link_key

表 10.244 链接字返回事件描述

参 数	值	参 数 说 明
Num_Keys 1 字节	0xFF	该事件所包含的链接字数 范围：0x01 ~ 0xFF
BD_ADDR [i] 6* Num_Keys 字节	0xFFFFFFFFXXXX	对应于相关链接字的 BD_ADDR
Num_Keys Link_Key [i] 16 字节	0xFFFFFFFFXXXX XXXXXXXXXXXX XXXXXXXXXXXX	对应于相关 BD_ADDR 的链接字

22. PIN 码请求事件

该事件用于表示需要一个 PIN 码以创建新的链接字。主机是采用 PIN 码请求应答，还是采用 PIN 码消极请求应答，取决于主机是否能够为主控制器提供 PIN 码。该事件及其描述如表 10.245 和 10.246 所示。如果 PIN 码请求事件被屏蔽，则主控制器就假设主机无 PIN 码。当主控制器生成一个 PIN 码请求事件，以便本地链路管理器对来自远程链路管理器的请求应答时(作为来自远程主机的 Create_Connection 或 Authentication_Requested 指令的结果)，本地主机必须在远程链路管理器检测到 LMP 应答超时之前，使用 PIN_Code_Request_Reply 或 PIN_Code_Request_Negative_Reply 指令应答。

表 10.245 PIN 码请求事件

事 件	OCF	事 件 代 码
Pin code request	0x16	BD_ADDR

表 10.246 PIN 码请求事件描述

参 数	值	参 数 说 明
BD_ADDR 6 字节	0XXXXXXXXXXXXX	新链接字所属设备的 BD_ADDR

23. 链接字请求事件

该事件用于表示需要针对指向 BD_ADDR 指定设备的链接创建一链接字。如果主机具有被请求链接字，则主机将使用 Link_Key_Request_Reply 指令将被请求链接字送往主控制器。如果主机不含被请求链接字，则主机将使用 Link_Key_Request_Negative_Reply 指令通知主控制器该主机不含被请求链接字。该事件及其描述如表 10.247 和 10.248 所示。如果链接字请求事件被屏蔽，则主控制器将假设主机不含其他链接字。当主控制器生成一链接字请求事件，以便本地链路管理器对来自远程链路管理器的请求应答时(作为来自远程主机的 Create_Connection 或 Authentication_Requested 指令的结果)，本地主机必须在远程链路管理器检测到 LMP 应答超时之前，使用 Link_Key_Request_Reply 或 Link_Key_Request_Negative_Reply 指令应答。

表 10.247 链接字请求事件

事 件	OCF	事 件 代 码
链接字请求事件	0x17	BD_ADDR

表 10.248 链接字请求事件描述

参 数	值	参 数 说 明
BD_ADDR 6 字节	0XXXXXXXXXXXXX	储存链接字所属设备的 BD_ADDR

24. 链接字通知事件

该事件用于通知主机已针对指向 BD_ADDR 指定设备的链接创建了一个新链接字。主机将在其存储设备中保存该新链接字，以便将来使用。并且，主机将使用 Link_Key_Request_Reply 指令将该链接字存储在主控制器的链接字存储设备中。该事件及其

描述如表 10.249 和 10.250 所示。

表 10.249 链接字通知事件

事 件	OCF	事 件 代 码
链接字通知事件	0x18	BD_ADDR, Link_key

表 10.250 链接字通知事件描述

参 数	值	参 数 说 明
BD_ADDR 6 字节	0XXXXXXXXXXXXX	生成新链接字设备的 BD_ADDR
Link_Key 16 字节	0XXXXXXXXXXXXX XXXXXXXXXXXXX XXXXXXXXXXXXX	与 BD_ADDR 相关联的链接字

25. 回送指令事件

处于本地回送模式时，主控制器将回送指令和数据到主机。回送指令事件用于返回含例外情况的所有从主机发往主控制器的命令。参见“Read_Loopback_Mode”中有关不能回送指令的情况。HCI_Command_Packet 事件参数包含所有包含头的 HCI 分组。该事件及其描述如表 10.251 和 10.252 所示。事件分组有效载荷最大值为 255 字节。由于 HCI 指令头数据长为 3 字节，则只返回开始 252 字节的指令参数。

表 10.251 回送指令事件

事 件	OCF	事 件 代 码
回送指令事件	0x19	HCI_command_packet

表 10.252 回送指令事件描述

	值	参 数 说 明
HCI_Command_Packet	0XXXXXXX	HCI 指令分组，包括头

26. 数据缓冲区溢出事件

表 10.253 数据缓冲区溢出事件

事 件	OCF	事 件 代 码
数据缓冲区溢出事件	0x1A	Link_type

表 10.254 数据缓冲区溢出事件描述

	值	参 数 说 明
Link_Type 1 字节	0X00	SCO 缓冲区溢出(语音信道)
	0X01	ACL 缓冲区溢出(数据信道)
	0x02 ~ 0xFF	保留

该事件用于表示主控制器数据缓冲区已溢出。如果主机发送分组数量超过限制时将引发该事件。Link_Type 参数用于表示溢出是由 ACL 数据还是由 SCO 数据引起的。该事件及

其描述如表 10.253 和 10.254 所示。

10.5 错误码表

10.5.1 错误码表

本节列出各种可能的错误码。当一指令失败时，将返回指示错误原因的错误码。错误码长度为一个字节，错误码允许范围是 0X01~0XFF。 下节将给出错误码的用法描述。

表 10.255 错误码表

错 误 码	描 述
0X01	未知的 HCI 指令
0X02	不能链接
0X03	硬件故障
0X04	呼叫超时
0X05	身份验证失败
0X06	键丢失
0X07	存储器已满
0X08	链接超时
0X09	最大链接数
0X0A	链接到设备 A 的最大 SCO 链接数
0X0B	ACL 链接已存在
0X0C	指令非法
0X0D	由于资源有限，主机被拒绝
0X0E	由于安全原因，主机被拒绝
0X0F	由于远程设备是一个人设备，主机被拒绝
0X10	主机超时
0X11	不支持的特性或参数值
0X12	非法的主控制器接口指令参数
0X13	由于另一端引起链接中断：用户中断链接
0X14	由于另一端引起链接中断：资源限制
0X15	由于另一端引起链接中断：关机
0X16	本地主机中断链接
0X17	重复尝试
0X18	不允许匹配
0X19	未知的 LMP PDU
0X1A	不支持的远程特性
0X1B	拒绝 SCO 补偿
0X1C	拒绝 SCO 间歇模式
0X1D	拒绝 SCO 无线模式
0X1E	非法链路管理器协议（LMP）参数
0X1F	未特别指明的错误

续表

错误码	描述
0X20	不支持的链路管理器协议参数值
0X21	不允许的角色改变
0X22	链路管理器协议响应超时
0X23	链路管理器协议错误处理事务冲突
0X24	不允许的 LMP PDU
0X25-0XFF	保留

10.5.2 错误码用法描述

本节目的是给出错误码的具体描述。使用方式根据具体实现而定。但是，一些特殊情况下的错误码的使用描述应更详尽、通俗易懂。下列错误码仅仅用于链路管理器协议报文，因此本节不作描述：

- 未知的链路管理器协议的协议数据单元 (0x19)；
- 拒绝同步面向链接 (0x1B)；
- 拒绝同步面向链接间隔 (0x1C)；
- 拒绝同步面向链接无线模式 (0x1D)；
- 错误的链路管理器协议参数(0x1E)。

可以根据具体实现，决定是在指令状态事件中还是在发出指令相关事件中（跟随一个带有状态=0x00 的状态指令）返回错误码。在这些情况下，不能由于错误而执行该指令，因此推荐使用指令状态事件。采用该事件的原因在于指令状态事件不可能适用于所有软件体系结构。

1. 未知的主控制器接口指令(0X01)

当主控制器收到带有不能识别操作码的主控制器接口指令分组时，主控制器在指令完成事件的状态参数中或在指令状态事件中返回“未知的主控制器接口指令”。给定操作码可能不对应于任何一个本文中定义的操作码，或任何厂商指定操作码，或是可能还未执行的指令。如果返回一个指令完成事件，状态参数是惟一包含在 **Return_parameters** 事件参数中的参数。使用何事件应根据具体实现而定。

2. 不能链接 (0X02)

当主机发出一个请求链接指令，并且当前不存在一个对应于指定链接句柄或 BD 地址的链接时，主控制器将在某一事件的状态参数中返回“不能链接”错误码。如果发出指令为一条要求必须返回指令完成事件的指令，则包含错误码的事件就是指令完成事件。否则，包含错误码的事件是指令状态事件或与发出指令有关的事件，这取决于实际情况。

3. 硬件故障(0X03)

当主机发出了一条指令，但该指令由于硬件故障不能执行时，主控制器将在事件状态参数中返回错误码——“硬件故障”。如果发出指令为一要求必须返回指令完成事件的指令，那么包含错误码的事件为指令完成事件。否则，包含错误码的事件为指令状态事件或与发出指令有关的事件（在收到状态=0x00 的指令状态事件之后）。

4. 呼叫超时(0X04)

如果主机发出一个 `Create_Connection` 指令，而且要链接的设备在呼叫定时器失效前没有在基带层次上进行应答，主控制器将在链接完成事件的状态参数中返回错误码“呼叫超时”。当主机已发出 `Remote_Name_Request` 指令以建立临时链接，但又发生呼叫超时的时候，也可以在 `Remote_Name_Request` 的状态参数中返回错误码（呼叫超时用 `Write_Page_Timeout` 指令设置）。

5. 验证失败(0X05)

当丢失 PIN 码或链接字导致匹配/验证计算结果错误，进而引起匹配或验证失败时，主控制器将在链接完成事件或验证完成事件的状态参数中返回“验证失败”错误码。

6. 字丢失(0X06)

当失去 PIN 码而导致匹配失败时，主控制器将在链接完成事件或验证完成事件的状态参数中返回“字丢失”错误码。

7. 存储器已满(0X07)

当主机发出一指令时，该指令要求主控制器保存新参数，但主控制器并无对该指令的存储能力，主控制器将在指令完成事件的状态参数中返回“存储器已满”错误码。这种情况可能是在 `Set_Event_Filter` 指令发出以后。对于 `Write_Stored_Link_Key` 指令，当主控制器不能够存储更多链接字时，将不返回错误码。主控制器根据可用于保存链接字的空闲空间存储链接字，并且将把可存储链接字数通知主机。

8. 链接超时(0X08)

该错误码可用于指示链接断开的原由。它通常在链接断开完成事件的原因参数中返回。因此在下述描述中它可称为原因码。

当链路监控定时器失效，并因此考虑释放链路时，主控制器将在一事件中发出“链接超时”原因码。链接监控超时可使用 `Write_Link_Supervision` 超时进行设置。返回该原因码的事件通常是链接断开完成事件。链接双方将返回该事件，而主控制器则将采用对应于链接到其他设备的物理链路链接句柄，向主机发送一链接断开完成事件。（当链接完成事件中返回原因码时，将可以在链接建立期间检测链路丢失）。

9. 最大链接数(0X09)

当蓝牙模块不能再设置链接时，主控制器将在指令状态事件、链接完成事件或远程命名请求完成事件的状态参数中返回“最大链接数”错误码。在链接完成事件，还是在指令状态事件（其中指令状态事件中状态为 0x00）之后的事件中返回该错误，取决于实际实现情况。该错误原因可能是由于硬件或固件限制而引起的。在返回错误以前，主机发出 `Create_Connection` 指令、`Add_SCO_Connection` 或 `Remote_Name_Request`。当需要建立临时链接以请求一名字时，可以在远程命名请求完成事件中返回“最大链接数”错误码。

10. 设备最大 SCO 链接数（0X0A）

当达到设备最大 SCO 链接数时，主控制器将在指令状态事件或链接完成事件的状态参

数中返回此错误码。而到底使用这两个事件中的哪一个取决于实际实现。该设备应是由先前发出的 Add_SCO_Connection 指令指定的设备。

11. ACL 链接已存在(0X0B)

当与一设备已有 ACL 链接, 并且主机又试图使用 Create_Connection 建立另外一个连接时, 主控制器将在指令状态事件或链接完成事件的状态参数中返回“ACL 链接已存在”错误码。具体使用哪一事件取决于实际实现。

12. 指令不允许(0X0C)

当主控制器处于准备接收带有某些操作码的指令, 并且收到的 HCI 指令分组不包含这些操作码时, 主控制器将在指令完成事件或指令状态事件的参数中返回“指令不允许”错误码。如果发出指令是一个要求返回指令完成事件的指令, 应使用指令完成事件。否则, 则使用指令状态事件。主控制器不必使用“未知的 HCI 指令”错误码, 因为这会需要对收到的操作码进行不必要的处理。何时使用“指令不允许”错误码主要根据实际实现情况而定。例如, 有些应用在链接请求、链路字请求或 PIN 码请求事件之后只能接受合适的 HCI 应答指令。通常允许复位指令。

13. 由于某种原因主机被拒绝 (0X0D~0X0F)

这些错误码通常用于指示拒绝呼入链接的原因。因此在下列描述中它们将被称为原因码。当主机已收到一个链接请求事件, 并且主机通过发出 Reject_Connection_Request 指令拒绝呼入链接时, 就可以使用一个原因码作为原因参数的值。在发出 Reject_Connection_Request 指令后, 由主控制器返回的指令状态事件 (STATUS=0x00) 将紧接着返回状态参数含已发出原因码的链接完成事件。而在 Reject_Connection_Request 指令的原因参数中的原因代码也将通过无线发出, 其目的就是使它能在初始化方的链接完成事件中返回。在此之前, 初始化方应已发出 Create_Connection 指令或 Add_SCO_Connection 指令, 并已收到指令状态事件(状态参数=0x00)。

14. 主机超时(0X10)

该错误码用于指示拒绝呼入链接的原因。因此在下列描述中它将被称为原因码。

假定主机已收到一个链接请求事件, 而且在链接定时器(其链接接受超时可使用 Write_Connection_Accept_Timeout 进行设置)终止前, 主机没有发出 Accept_Connection_Request 指令或 Reject_Connection_Request 指令。在这种情况下, 主控制器将发出一个状态参数中含“主机超时”原因码的链接完成事件。该原因码可通过无线发送, 以便可在初始化方链接完成事件中返回。在这以前, 初始化方已发出一条 Create_Connection 指令或 Add_SCO_Connection 指令, 并已收到一指令状态事件(状态参数=0x00)。

15. 不支持的特性或参数值(0X11)

当主控制器收到了含有一个或多个不被硬件支持的参数值的指令时, 主控制器将在事件状态参数中返回“不支持特性或参数值”错误码。但是, 这些参数应在本文件指定允许的参数范围以内。如果发出指令为一要求返回指令完成事件的指令, 那么包含错误码的事件为一指令完成事件。否则, 包含错误码的事件为一指令状态事件或与发出指令有关的事件(在

含有状态参数=0x00 的指令状态事件之后)。

16. HCI 指令参数非法(0X12)

当总的参数长度（或收到指令中的一个或多个参数值）不符合本文件所指定长度时，主控制器将在事件的状态参数中返回“错误的 HCI 指令参数”错误码。

尽管参数值在允许参数范围内，但如果该参数值当前不被允许，也将返回错误码。例如：当一指令需要一个 ACL 链接句柄，但主机已将 SCO 链接句柄作为参数的情况。还有，主控制器通过事件请求一个链接字、一个 PIN 码或一个呼入链接应答时，主机使用含未收到请求 BD_ADDR 的应答指令进行应答。

如果发出指令是要求返回指令完成事件的指令，则包含错误码的事件为指令完成事件。否则，包含错误码的事件为指令状态事件或与发出指令有关的事件(在 STATUS=0x00 指令状态事件跟随一个带有状态=0x00 的指令状态事件)。

17. 其他终端终止链接 (0X13~0X15)

该错误码用于指示链接断开原因。因此它们在以下描述中称为原因码。

当主单元发出链接断开指令时，将把一个原因码作为原因参数值使用。“本地主机终止链接”原因参数将于主控制器在发出链接断开指令之后，返回链接断开事件（状态=0x00），并在此之后在链接断开完成事件的原因参数中返回。链接断开指令的原因参数中的原因代码将通过无线发出，以便能够在远程链接断开完成事件的原因参数中再次返回。

18. 本地主机终止链接(0X16)

由于它在链接断开完成事件的原因参数中返回，该错误码称为原因码。

19. 重复尝试(0X17)

当设备由于验证或匹配失败的原因而没有更多时间再进行验证或匹配时，主控制器将在链接完成事件或验证完成事件的状态参数中返回“重试”错误码。

20. 不允许匹配(0X18)

当设备由于某些原因不允许匹配时，主控制器将在链接完成事件或验证完成事件的状态参数中返回“不允许匹配”错误码。例如：PSTN 适配器就只允许在按下该适配器的一个按键之后的某一时间段内进行匹配（在适配器的一个按键被按了以后）配对。

21. 不支持的远程特性(0X1A)

当指令参数中指定的远程设备不支持与发出指令有关的特性时，主控制器将在与发出指令有关事件的状态参数中返回“不支持的远程特性”错误码。该错误码也可用作链接断开指令的原因参数值。该错误码将通过无线方式发送，以便它能够在远端链接断开事件的原因参数中返回。在链接断开指令发出方的指令状态事件（状态=0X00）之后的链接断开完成事件中，原因参数将包含原因码“本地主机终止链接”。

22. 未定义错误(0X1F)

当本文没有指定适用错误码时，适用“未定义错误”错误码。

23. 不支持的 LMP 参数值(0X20)

当指令参数中指定的远程设备返回包含 LMP 错误码 0x20（即：不支持的 LMP 参数值）的 LMP 消息时，主控制器将在与发出指令有关事件的状态参数中返回“不支持的 LMP 参数值”错误码。

24. 不允许的角色变化(0X21)

当不允许角色变化时，主机在链接完成事件或角色变化事件状态参数中返回“不允许的角色变化”错误码。如果本地主机发出 Switch_Role 指令，但远程设备拒绝角色变化，那么错误码将在角色变化事件中返回。如果由于设备接受一个呼入 ACL 链接和角色变化请求，而角色变化又被初始化设备拒绝，从而引起链接失败时，错误码将在两端链接完成事件中返回。

25. LMP 应答超时(0X22)

当远程设备在 LMP 最大应答时间内，不能对来自本地设备的 LMP PDU 进行应答时，主控制器将在指令完成事件或与发出指令（在状态=0X00 的指令状态事件之后）有关事件的状态参数中返回“LMP 应答超时”错误码。

26. LMP 错误处理冲突(0X23)

当指令参数所指定的远程设备返回包含 LMP 错误码(0x23)的 LMP 消息，主控制器将在与发出信号有关的事件状态参数中返回“LMP 错误处理冲突”错误码。

27. 不允许的链路管理器协议 PDU(0X24)

当指令参数所指定的远程设备返回包含 LMP 错误码（0x24）的 LMP 消息时，主控制器将在与发出信号有关的事件状态参数中返回本错误码。

第 11 章 HCI 传输层

本章描述 USB 传输子层、RS232 传输子层、UART 传输子层（介于主机和主控制器之间），以及与三者相关的 HCI 指令和针对三者定义的事件和数据分组流，但 USB、UART 传输子层不对分组进行解码。

11.1 HCI USB 传输层

主机和蓝牙无线模块之间的关系如图 11.1 所示。本节主要讨论图 11.1 中标有“USB 功能”双向箭头的实现细节。

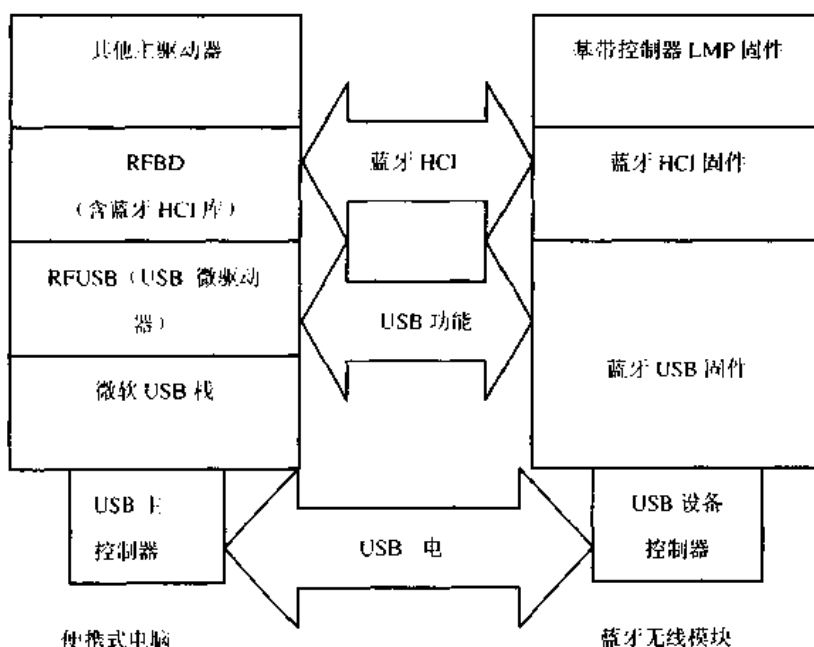


图 11.1 主机和蓝牙无线模块之间的关系

USB 硬件可以以两种方式嵌入：

- 作为一个 USB 加/解密芯片；
- 集成到笔记本主板。

11.1.1 HCI 终端要求

本节概述用于与主机更好工作的 USB 终端。终端号（以下称为“建议终端地址”）可在驱动程序初始化时动态识别，这将根据具体实现而定。

1. 描述符概述

USB 设备可看作高速设备，其固件配置由两个接口组成。第一个接口(接口 0)为固定设置，并包含 BULK 和中断终端。第二个接口(接口 1)提供可扩展的同步带宽占用方式，该接口方式提供四种设置，以提供基于同步带宽需求的占用方式。其缺省接口为空，以使设备能够支持非同步带宽。

一个 HCI 帧，包含一个 HCI 头和 HCI 数据，应包含于 USB 事务中。USB 事务为一个或多个包含 I/O 请求数据的 USB 帧。例如，包含 256 字节的 ACL 数据分组(包括 HCI 头和 HCI 数据)将在 I/O 请求中通过 BULK 终端发送。该 I/O 请求将需要四个 64 字节的 USB 帧，并组成一个事务。

当通过选择接口呼叫调整同步带宽占用方式时，终端可自由选择两种接口，以便不会中断或重新提交任何中间 BULK 或中断事务。表 11.1 给出了所需配置。

表 11.1

接口号	可选设置	推荐端地址	端类型	推荐最大分组尺寸
HCI 指令				
0	0	0X00	控制	8/16/32/64
HCI 事件				
0	0	0X81	中断 (IN)	16
ACL 数据				
0	0	0X82	BULK (IN)	32/64
0	0	0X02	BULK (OUT)	32/64
无激活语音信道 (兼容 USB)				
1	0	0X83	ISOCH (IN)	
1	1	0X03	ISOCH (OUT)	9
一条 8 位编码的语音信道				
1	1	0X83	ISOCH (IN)	9
1	1	0X03	ISOCH (OUT)	9
两条 8 位编码的语音信道和一条 16 位编码的语音信道				
1	2	0X83	ISOCH (IN)	17
1	2	0X03	ISOCH (OUT)	17
三条 8 位编码的语音信道				
1	3	0X83	ISOCH (IN)	25
1	3	0X03	ISOCH (OUT)	25
两条 16 位编码的语音信道				
1	4	0X83	ISOCH (IN)	33
1	4	0X03	ISOCH (OUT)	33
三条 16 位编码的语音信道				
1	5	0X83	ISOCH (IN)	49
1	5	0X03	ISOCH (OUT)	49

表 11.2 和表 11.3 描述了终端给定数据流。

表 11.2

语音信道数 1		语音数据持续时间 3ms I/O 请求		编码 8 位	
时间 (微秒)	USB 数据 (数据头参照 HCI 头, 发自主机)	数据队列 (读/写)	时间 (微秒)	无线数据	收/发数量 (微秒)
0	收到 0 个字节, 发送 9 个 字节 (3 个头, 6 个数据)	0/6	0	发送 0	0/0
		10/6	0. 625	收到 10	1. 25/0
1	收到 0 个字节, 发送 9 个 字节 (9 个字节 HCI 数 据)	10/15	1. 25	发送 0	1. 25/0
		20/15	1. 875	收到 10	2. 50/0
2	收到 0 个字节, 发送 9 个 字节 (9 个字节 HCI 数 据)	20/24	2. 50	SEND 0	2. 50/0
		30/24	3. 125	收到 10	3.75/0
3	收到 9 个字节(3 个头, 6 个数据), 发送 9 个字节(3 个头, 6 个数据)	24/20	3.75	发送 10	3.75/1.25
4	收到 9 个字节(9 个字节数 据), 发送 9 个字节(9 个 字节 HCI 数据)	25/29	4.375	收到 10	5.0/1.25
5	收到 9 个字节(9 个字节数 据), 发送 9 个字节(9 个 字节 HCI 数据)	16/28	5.0	发送 10	5.0/2.5
		26/28	5.625	收到 10	6.25/2.5
6	收到 9 个字节(3 头, 6 数 据), 发送 9 个字节(3 头, 6 数据)	20/24	6.25	发送 10	6.25/3.75
		30/24	6.875	收到 10	7.5/3.75
7	收到 9 个字节(9 个字节数 据), 发送 9 个字节(9 个 字节 HCI 数据)	21/23	7.5	发送 10	7.5/5.0
8	收到 9 个字节(9 个字节数 据), 发送 9 个字节(9 个 字节 HCI 数据)	22/32	8.125	收到 10	8.75/5.0
		22/22	8.75	发送 10	8.75/6.25
9	收到 9 个字节(9 个字节数 据), 发送 9 个字节(9 个 字节 HCI 数据)	26/28	9.375	收到 10	10.0/6.25

续表

语音信道数 1		语音数据持续时间 3ms/I/O 请求		编码 8 位	
10	收到 9 个字节(9 个字节数据), 发送 9 个字节(9 个字节 HCI 数据)	17/27	10	发送 10	10.0/7.5
		27/27	10.625	收到 10	11.25/7.5
11	收到 9 个字节(9 个字节数据), 发送 9 个字节(9 个字节 HCI 数据)	18/26	11.25	发送 10	11.25/8.75

因为无线发送器平均每 1 毫秒发送 8 字节语音数据, 而 USB 每 1 毫秒发送 8 字节语音数据, 所以将要求聚集。

表 11.3

语音信道数 2		语音数据持续时间 每 I/O 请求 3 毫秒		编码 8 位	
时间 (毫秒)	USB (数据头参照 HCI 头, 并发自主机)	数据队列 (读/写)	时间 (毫秒)	无线数据	收/发数量 (毫秒)
0	信道#1 收到 0 个字节, 信道#1 发送 17 个字节 (3 个头, 14 个数据)	C1-0/14 C2-0/0	0	发送 0 到 C1	C1-0/0 C2-0/0
		C1-20/14 C2-0/0	0.625	C1 收到 20	C1-2.5/0 C2-0/0
1	信道#1 收到 0 个字节, 信道#1 发送 17 个字节(17 个字节 S, HCI 数据)	C1-20/31 C2-0/0	1.25	发送 0 到 C2	C1-2.5/0 C2-0/0
		C1-20/31 C2-20/0	1.875	C2 收到 20	C1-2.5/0 C2-2.5/0
2	信道#1 收到 0 个字节, 信道#1 发送 17 个字节(17 个字节 S, HCI 数据)	C1-20/28 C2-20/0	2.50	发送 20 到 C1	C1-2.5/2.5 C2-2.5/0
		C1-40/28 C2-20/0	3.125	C1 收到 20	C1-5.0/2.5 C2-2.5/0
3	信道#2 收到 0 个字节, 信道#2 发送 17 个字节(3 个头, 14 个数据)	C1-40/28 C2-20/14	3.75	发送 0 到 C2	C1-5.0/2.5 C2-2.5/0
4	信道#2 收到 0 个字节, 信道#2 发送 17 个字节(17 个字节 S, HCI 数据)	C1-40/28 C2-40/31	4.375	C2 收到 20	C1-5.0/2.5 C2-5.0/0
5	信道#2 收到 0 个字节, 信道#2 发送 17 个字节(17 个字节 S, HCI 数据)	C1-40/8 C2-40/48	5.0	发送 20 到 C1	C1-5.0/5.0 C2-5.0/0

续表

语音信道数 2		语音数据持续时间 每一 I/O 请求 3 毫秒		编码 8 位	
		C1-60/8 C2-40/48	5.625	发送 20 到 C1	C1-7.5/5.0 C2-7.5/2.5
6	信道#1 收到 17 个字节, 信道#1 发送 17 个字节(3 个头, 14 个数据)	C1-46/22 C2-40/48	6.25	发送 20 到 C2	C1-7.5/5.0 C2-5.0/2.5
		C1-46/22 C2-60/48	6.875	C2 收到 20	C1-7.5/5.0 C2-7.5/2.5
7	信道#1 收到 17 个字节, 信道#1 发送 17 个字节(17 个字节 S, HCI 数据)	C1-29/19 C2-60/48	7.5	发送 20 到 C1	C1-7.5/7.5 C2-7.5/2.5
8	信道#1 收到 17 个字节, 信道#1 发送 17 个字节(17 个字节 S, HCI 数据)	C1-32/36 C2-60/28	8.125	C1 收到 20	C1-10/7.5 C2-7.5/5.0
		C1-32/36 C2-60/8	8.75	C2 发送 20	C1-10/7.5 C2-7.5/5.0
9	信道#2 收到 17 个字节, 信道#2 发送 17 个字节(3 个头, 14 个数据)	C1-32/36 C2-54/22	9.375	C2 收到 20	C1-10/7.5 C2-10/5.0
10	信道#2 收到 17 个字节(17 个字节数据), 信道#2 发 送 17 个字节(17 个字节 S HCI 数据)	C1-32/16 C2-37/39	10	发送 20 到 C1	C1-10/10 C2-10/5.0
		C1-52/16 C2-37/39	10.625	C1 收到 20	C1-12.5/10 C2-10/5.0
11	信道#1 收到 17 个字节, 信道#1 发送 17 个字节(17 个字节 S, HCI 数据)	C1-52/16 C2-20/36	11.25	发送 20 到 C2	C1-12.5/10 C2-10/7.5

2. 控制终端要求.

终端 0 用于配置和控制 USB 设备。终端 0 还可用于允许主机向主控制器发送特定 HCI 指令。当 USB 固件在具有蓝牙类别码的终端上收到一个分组时, 他应将该分组视为一个 HCI 指令分组。

3. BULK 终端要求

数据完整性是 ACL 数据的一个关键方面。他与带宽请求一起成为使用 BULK 终端的原因。每毫秒应通过总线传输多个 64 字节分组。推荐批最大分组尺寸为 64 字节。BULK 能够通过总线每毫秒传输多个 64 字节的分组帧。

BULK 能够进行检错和纠错。通过该管道的数据流可流向多个从设备。为了避免阻塞, 推荐主控制器采用类似于共享终端模型的流控制模型。

4. 中断终端要求

中断终端能够保证事件以可预测并及时的方式传递。事件分组可以在一定允许延时条件下通过 USB 发送。中断终端应有 1 毫秒的时间间隔。

USB 软件和固件不必对传送到主控制器的事件充分了解。

5. 同步终端要求

同步终端与无线主控制器相互传输 SCO 数据。时间是该数据类型的重要因素。USB 固件应将数据内容传递到主控制器的 SCO 先进先出队列 (FIFO)。如果 FIFO 满, 则应用新数据覆盖原有数据。终端应有 1 毫秒的时间间隔, 参见 USB 规范 1.0 和 1.1 要求。

无线收发器可支持 3 个 64kb/s 语音信道, 可以接收不同编码方式的数据——16 位线性音频编码要求最大数据量。该终端推荐最大分组尺寸至少为 64 字节。但是, 如果不需要支持 3 条 16 位编码的语音信道, 32 字节作为最大分组尺寸也可接受。

11.1.2 类别码

类别码将用于所有 USB 蓝牙设备。这将允许调用合适的驱动程序, 而不需要考虑设备由哪家厂商提供。他也允许通过控制终端区分 HCI 指令和 USB 指令。

类别码(bDeviceClass)为 0xE0——无线控制器

子类码(bDeviceSubClass)为 0X01——射频控制器

协议码(bDeviceProtocol)为 0X01——蓝牙编程

11.1.3 设备固件升级

固件升级能力并非必需的特性。但如果实现, 固件升级应兼容于“设备固件升级通用串行总线设备类别规范”(1.0 版, 1999/05/13), 详见 USB 论坛网址 <http://WWW.USB>。

11.1.4 限制

1. 功率限制

目前, 支持 USB 设备的主控制器被置于如 PIIX4 的芯片内。当系统处于 S3 或 S4 时, USB 主控制器将不能接收信号。只有当系统处于 S1 或 S2 时, USB 才能被唤醒。

USB 主控制器的另一特性是, 当一设备被连接上时, USB 主控制器将不断检查存储器, 以确认是否有需要完成的工作。检查存储器的时间间隔是 1 毫秒。这将阻止处理器进入 C3 节能状态。由于笔记本处理器不能进入 C3 状态而将导致显著的电能损失。这对商业用户是一个大问题, 因为一个典型的商业用户将在 C3 状态中花费 90% 的时间。

2. 其他限制

同步终端可能导致数据差错。终端 1 和终端 2 都可能会有数据差错。

USB 在所有数据传输时都提供 16 位循环冗余校验。USB 误码率为 10^{-3} 。

注意: 当加密狗从系统中取出时, 无线收发器将不再工作(假定这是一台总线驱动的设备)。这也就意味着设备将丢失连接。

11.2 HCI RS232 传输层

HCI RS232 传输层的目的在于在蓝牙主机和蓝牙主控制器之间的物理 RS232 接口上使用蓝牙 HCI，如图 11.2 所示。

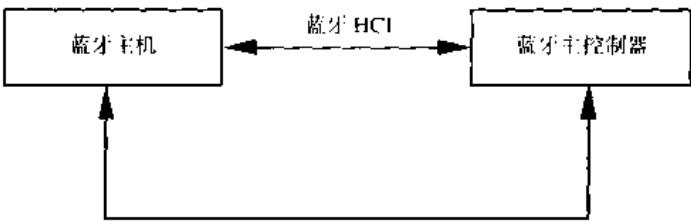


图 11.2 HCI RS232 传输层

11.2.1 概述

通过 RS232 传输层可发出四种 HCI 分组，包括：HCI 指令分组、HCI 事件分组、HCI ACL 数据分组和 HCI SCO 数据分组(参见“主控制器接口功能规范”)。HCI 指令分组仅能用于发送到蓝牙主控制器，HCI 事件分组仅能由蓝牙主控制器发送，HCI ACL/SCO 数据分组则可由蓝牙主控制器自由发送和接收。

但是，主控制器接口不能区分四种 HCI 类型。因此，如果通过一通用物理接口发出一个 HCI 分组，则 HCI 分组指示器必须根据表 11.4 执行加操作。

表 11.4 HCI RS232 分组头

HCI 分组类型	HCI 分组指示器
HCI 指令分组	0X01
HCI ACL 数据分组	0X02
HCI SCO 数据分组	0X03
HCI 事件分组	0X04
错误消息分组	0X05
协商分组	0X06

除上述四种 HCI 分组类型外，还有两种分组类型用于支持动态协商和错误报告。接收端使用错误消息分组(0x05)将错误报告给发送端。协商分组(0x06)则用于协商通信设置和协议。

当每次发送一个以上 HCI 分组时，HCI 分组指示器将在一个 8 位序列号上每次加 1，除非转发分组作为纠错的一部分。HCI 分组紧跟在该序列号段后。所有四种 HCI 分组都有一长度段，该段用于确定 HCI 分组长为多少字节。尽管协商分组能够达到 7 个以上字节，但错误消息分组和协商分组都是基于扩展段(Extension)值的定长分组。

基本 RS232 传输分组的帧结构如表 11.5 所示。

表 11.5 RS232 传输分组的帧结构

LSB		MSB
分组类型 (8 位)	序列号 (8 位)	HCI 分组 或错误消息/协商分组有效载荷

11.2.2 协商协议

在 RS232 链路上发送任何字节之前，应当在主控制器和主机之间对波特率、奇偶校验值类型、终止位和协议模式进行协商。 T_{detect} 是发射机检测 CTS 状态变化的最大时间，加上如果 RTS/CTS 用于发送器错误指示和重新同步时刷新传输缓冲区所需的时间。否则， T_{detect} 代表本地中断延时。主机将使用协议模式 0x13，首先发送一个含最大推荐值的协商分组，加上在缺省 UART 设置下 ACK 码为 000b 的主机 T_{detect} 值。同时，主控制器方也将其 UART 配置设置为初始化参数，并等待来自主机的协商分组。

如果主控制器方能够接受来自主机的推荐值，它将回送含同样 UART 设置值，以及 ACK 为 001b 的主控制器 T_{detect} 值的协商分组。然后，主机返回含同样 UART 设置值的协商分组，以及 ACK 为 001b 的 T_{detect} 值作为最终确认，然后将主机 UART 设置为新值。在收到来自主机的最终确认分组以后，主控制器也将其 UART 设置为新值。

另一方面，如果主控制器方不能接受推荐值，它应发送一新推荐值集，以及 ACK 为 010b 的 T_{detect} 值。每一方都将继续执行这些步骤，直到双方收到可接受的 ACK 代码值。初始化协商期间的出错检测和出错恢复将以协议模式 0x13 方式执行。

协商可由任何一方在任意时间重新初始化，以协商新值，或通知新的 T_{detect} 时间。当协商在数据传输期间重新初始化时，它将使用先前的协商设置，以交换新参数，而不是使用缺省值。初始参数为：

- 波特率：9600b/s；
- 奇偶校验值类型：无奇偶校验值；
- 数据位数：8 位(注：只允许 8 位数据长度)；
- 终止位：1 位；
- 协议模式：0x13 (HDLC，采用 COBS/CCITT-CRC 帧分方式)。

协商分组格式如表 11.6 所示。

表 11.6 协商分组格式

LSB			MSB		
分组类型头 0x06 (8 位)	序列号 8 位	UART 设置和 ACK (8 位)	波特率 16 位	T_{detect} 时间 (16 位)	协议模式 (8 位)

- (1) 序列号
8 位，每传输一分组增加 1，不包含重发分组。
- (2) UART 设置和 ACK 段 (见表 11.7)

表 11.7 UART 设置和 ACK 段格式

位 0-1	位 2	位 3	位 4	位 5-7
保留	终止位	启用奇偶校验	奇偶校验类型	ACK 码

- ① 终止位
0: 1 终止位
1: 2 终止位

② 启用奇偶校验

0: 奇校验

1: 偶校验

③ ACK 代码

• 000b: 请求;

• 001b: 接受;

• 010b: 不接受新推荐值;

• 011b~111b: 保留。

(3) 波特率

• 波特率=27, 648, 000/N 时输入 N;

• N=0 非法;

• 最大可能速率为 27.648Mb/s;

• 最小可能速率为 421.88b/s;

• 使用无类型小 Endian 格式, 即最低位最先发送。

(4) T_{detect} 时间

16 位段, 以发射机检测 CTS 变化所需最大时间, 加上如果 RTS 和 CTS 用于重新同步而用于刷新传输先进先出队列 (FIFO) 的时间进行填充。否则, 它将以本地中断延迟填充。 T_{detect} 以 100 微秒为时间单位。使用无类型小 ENDIAN 格式, 即最低位首先发送。

(5) 协议模式 (见表 11.8)

表 11.8 协议模式

位 0	位 1	位 2	位 3	位 4	位 5	位 6	位 7
使用 CRC	使用分界符	使用 RTS/CTS	RTS/CTS 模式	使用纠错	Ext0	Ext1	Ext2

① 使用 CRC

0: 分组末尾不附属 CRC-CCITT;

1: 分组末尾附属 CRC-CCITT;

16 位 CRC 使用 RTS/CTS 或分界符, 尽管当使用分界符时本规范只说明一种情况。

CRC 发生器的生成多项式为 $x^{16}+x^{12}+x^5+1$ 。

② 使用分界符

0: 分界符, 0x7E, 未使用;

1: 分界符, 0x7E, 与 COBS 一起使用 (缺省)。

③ 使用 RTS/CTS

0: 未使用 RTS/CTS; (缺省)

1: 使用 RTS/CTS。

④ RTS/CTS 模式

0: RTS/CTS 用于错误指示和重新同步 (缺省);

1: RTS/CTS 用于硬件流控制, 参见 HCI UART 传输层。

⑤ 使用纠错

0: 未支持纠错;

即使不支持纠错，也将发送错误消息。

1: 支持纠错（缺省）。

如果 RTS/CTS 用于同步，则纠错将重发含有错误的分组和所有其他分组。另一方面，如果 0x7E 用作分界符，其中 COBS 作为同步机制，然后纠错将只重发含有错误的分组。

⑥ Ext2, Ext1, Ext0

这 3 位为附属于协商分组后的额外字节，用于今后扩充。

11.2.3 分组传输协议

分组传输可以提供或不提供检错机制，如奇偶校验或无奇偶校验，CRC 校验或无 CRC 校验，这主要取决于应用环境。

可以选择 RTS/CTS 或分界符作为一种同步机制。RTS/CTS 的使用可以减少 COBS 编码计算时间，但它需要 2 根额外的铜芯，而铜丝则不适用于某些应用。如果必须使用 3 芯电缆，或不使用可编程 RTS/CTS，则分界符、0x7E 就能够与 COBS 一起使用。

这两种方案的纠错差别很小。如果 RTS/CTS 用于再次同步，他将简单重发所有分组，并以含有错误的分组作为起始分组。如果使用分界符，发送端将仅重发含有错误的分组。可以不使用纠错，但当接收端检测到错误时仍要将错误消息分组发到发送端。

HCI RS232 传输层通常使用 8 位数据长度，并且假定为 little Endian 格式。并且最低位先发。

主控制器可以仅支持一种协议模式，但主机必须能够支持任何形式。

11.2.4 使用含有 COBS 的分界符同步

1. 使用含 COBS 和 CRC 的分界符，协议模式 0X13

在不能使用 RTS/CTS，或者它们通过物理连接而作为硬件流控制时，将采用类似于 HDLC 的含 16 位 CRC(CRC-CCITT)的帧和含 COBS 的分界符 0x7E (COBS)，作为检错和重新同步的手段。

CRC-CCITT 将使用以下多项式生成 16 位的校验和： $x^{16}+x^{12}+x^5+1$ 。该 16 位 CRC 应附加于分组末尾，同时又正好在结束分界符 0x7E 之前。起始分界符 0x7E 之后为分组类型指示段。

CONSISTENT OVERHEAD BYTE STUFFING 是 PPP 最近的改进，不考虑数据模式，它将产生不到 0.5% 的开销。它将使用两个步骤替换分界符 0x7E。第一步骤将消除 0 并在起始和结束分界符之间用 0x00 替代所有 0x7E。

在此采用一种简单纠错方案以降低因支持纠错而产生的开销。当接收端检测到任何错误时，它将向发出方返回一个含错误类型的错误消息分组。错误消息分组包括一个含错误段的序列数（含错误信息的 SEQ NO），以标示是哪个分组检测到错误。每一分组的序列号段都是一个 8 位段。它们将在传输每一分组时增加 1，重发分组除外。重发分组应在序列号（SEQ NO）段中包含原序列号。

发送端应只重发含错误的 HCI 分组。该分组序列号段用于指示该错误。而接收端将负责记录分组的正确顺序。如果发送端在重发保持缓冲区内的分组序列号不正确，它应发送错误类型为 0x81 的错误消息分组，以及错误段为重发分组丢失序列号的序列号（SEQ No），以便接收端能够检测丢失分组。在这种情况下，不能执行完整的纠错过程。但是，接收端至

少能检测分组的丢失。

接收端能够在等待重发分组时和超时以前，等待的时间至少 4 倍于远程 T_{detect} 、本地 T_{detect} 、错误消息分组传输时间，加上该重发分组时间之和。当发生超时时，接收端可以通过发送另一错误信息分组（错误类型=0x09）重新请求或放弃，并将情况报告上层。

2. 帧分（Framing）

BOF(0x7E)、CRC-CCITT 和 EOF (0x7E) 将加入本文件所描述的基本分组中，如表 11.9 所示。当 CRC 送出时，应首先送出最低位字节。

表 11.9

LSB			MSB		
0X7E BOF (8 位)	分组类型 (8 位)	序列号 (8 位)	有效载荷	CRC (16 位)	0X7E EOF (8 位)

3. 错误消息分组

错误消息分组格式如表 11.10 所示。

表 11.10 可用错误类型

分组类型, 0X05 (8 位字段)	序列号 SEQ (8 位字段)	错误类型 (8 位字段)	含错误的序列号 (8 位字段)
错误类型		描述	
0X00		保留	
0X01		数据速率不匹配错或超时错	
0X02		奇偶校验错	
0X03		保留	
0X04		帧分错误	
0X05-0X07		保留	
0X08		CRC 错误	
0X09		丢失序列号	
0X0A-0X80		保留	
0X81		重发分组丢失	
0X82-0XFF		保留	

4. 一致开销字节填充法

COBS 要求两步编码。

第一步是消除零。如果启用 CRC 校验，则在增加起始和结束分界符（0x7E）以前，附加 16 位 CRC 之后，进行本步骤。每一 COBS 代码块包括后面为零或更多数据字节的 COBS 代码。代码字节 0x00、0xD1、0xD2 和 0Xff 不得使用。COBS 零消除过程查找首先出现零值的分组。为简化编码，将在 CRC 之后临时增加一个零值在分组末端作为临时占用位。字节数和是否包含首个零决定使用的编码。如果该数为 207 或更小，则它将作为 COBS 编码

字节，后面紧跟非零数据字节，但不包括为零的末字节。另一方面，如果该数大于 207，则使用编码字节 0xD0，且后面紧跟首个 207 非零字节。该过程将重复执行直至分组所有字节，（包括末位临时占用的零位）都已完成编码。如果在 0~30（8 进制）非零字节之后检测到一对 0x00，将使用字节数与 0xE0 的和作为 COBS 编码使用，其后为非零字节，但不包括这对零。如果检测到 3 个到 15 个 0x00 字节，则将该 0x00 字节数与 0xD0 的和用作编码，后面不再跟其他字节。

表 11.11

代码	后面内容	描述
0X00		未使用
0X01~0XCF	N-1 字节数据	N-1 字节数据加隐含的零
0XD0	N-1 字节数据	N-1 字节数据不含零
0XD1		未使用
0XD2		保留
0XD3~0XDF	缺省	一个 N-0XD0 零的运行
0XE0~0XFE	N-E0 字节数据	隐含两个零的数据
0XFF		未使用

第二步用 0x00 替代 0x7E。这两个步骤可以循环方式一起执行，以减少编码时间。具体细节和参考编码，参见“PPP 一致字节填充法（COBS）”。

11.2.5 使用 RTS/CTS 同步

1. 使用 RTS/CTS 同步，协议模式 0X14

HCI 分组传输流由两个 MODEM 控制/状态信号 RTS 和 CTS 处理。CTS 和 RTS 以空 MODEM 方式连接，也就意味着本地 RTS 应连接到远程 CTS，而本地 RTS 应连接到远程 RTS。这些 MODEM 控制/状态信号用于将错误检测结果通知其他方，同时在检测到错误后与分组发出端重新同步。

只有 CTS 位为 1 时，才发送 HCI 分组。如果在 HCI 分组传输期间或在末尾字节传输之后，CTS 位变为 0，这就表示有错误发生。接收端一旦检测到任何错误，将马上撤销 RTS，并将把含错误类型的错误分组返回发送端。该错误分组包括含错误段的序列号，该错误段指示是哪个分组错误。每一分组的序列号段都是一个 8 位段。该段在除重发分组以外的每次传输任何类型分组时都加 1。该重发分组应包括 SEQ No 字段中原来的序列号。

当发送端任何时候检测到 CTS 位从 1 变为 0 时，都将保持传输并等待，直至在恢复传输前收到错误分组。当接收端准备接收新数据时，它应在最小 T_{detect} 时间后确认 RTS。 T_{detect} 时间是发送端用于检测 CTS 位状态变化的最大时间，加上它刷新传输缓冲区时间的和。在协商期间，每一端的 T_{detect} 值都应互相通知对方。本地 T_{detect} 值和远端 T_{detect} 值，以及波特率，能够用于估算重发占用缓冲区所需队列长度。在接收端再次确认 RTS 之前，它应刷新 RX 缓冲区。

发送端应自错误分组开始重发所有 HCL 分组，错误分组的错误由错误段的序列号 SEQ No 指明。在重发之前，应刷新可能缓存自前一丢弃分组开始以后其它分组的传输缓冲区。

当它从重发占用缓冲区中重发分组时，应采用其 SEQ No 与错误相匹配的序列号所在分组来开始传输。如果发送端在重发占用缓冲区中不包含具有正确序列号的分组，发送端应发送一错误类型为 0x81 的错误消息分组，并且它将跳过序列号在缓冲区中可用的分组。在这种情况下，不能执行完整纠错过程。但接收端至少能够检测分组丢失。

2. 错误消息分组

错误消息分组格式参见表 11.10。

3. 流控制实例

实例 1：正常恢复处理（见表 11.12）

表 11.12 流控制实例 1：正常恢复处理

控制方	主机方
0) 声明 CTS, 且检测已声明 CTS	声明 RTS, 且检测已声明 CTS
	1) 送出控制/数据[n], 并在重发保持缓冲区中存储控制/数据[n]
2)收到错误的控制/数据[n]	
3) 撤销声明 RTS 4a)发送[n]错误消息, 并在TX 重发保持缓冲区中存储[n]的错误消息 4b)清除 RX 先进先出队列并等待 T_{detect} (主机)时间长度	4)检测到撤销声明的 CTS
	5a)停止进一步传输, 并等待直到 TX 先进先出队列清空(如果允许, 刷新先进先出队列) 5b)收到[n]错误消息
6)声明 RTS	
	7)检测到声明的 CTS 8)重发控制/数据[n]

实例 2：双方同时检测到错误（见表 11.13）

表 11.13 双方同时检测到错误

控制方	主机方
0) 声明 RTS 并检测已声明的 CTS 1) 发送控制/数据[n], 并在重发保持缓冲区中存储控制/数据[n] 2)收到错误的控制/数据[n] 3) 撤销声明 RTS 4) 检测到已撤销声明的 RTS 5a) 停止进一步传输, 并等待直到 TX 先进先出队列清空(如果允许, 可刷新该先进先出队列)	0) 声明 RTS, 并检测已声明的 CTS 1) 发送控制/数据[n], 并在重发保持缓冲区中存储控制/数据[n] 2) 收到错误的控制/数据[n] 3) 撤销声明 RTS 4) 检测到已撤销声明的 RTS 5a) 停止进一步传输, 并等待直到 TX 先进先出队列清空(如果允许, 可刷新该先进先出队列)

续表

5b) 清空 RX 先进先出队列并且等待 T_{detect} (控制器)的时间长度	5b) 清空 RX 先进先出队列并且等待 T_{detect} (控制器)的时间长度
6) 声明 RTS	6) 声明 RTS
7) 检测已声明的 CTS	7) 检测已声明的 CTS
8) 发送错误消息[x], 并在 TX 重发保持缓冲区中存储错误消息[x]	8) 发送错误消息[x], 并在 TX 重发保持缓冲区中存储错误消息[x]
9) 收到错误消息[x]	9) 收到错误消息[n]
10) 重发控制/数据[x]	10) 重发控制/数据[x]

实例 3: 错误信息 (见表 11.14)

表 11.14 错误信息

控制方	主机方
0) 声明 RTS 并检测已声明的 CTS	0) 声明 RTS 并检测已声明的 CTS
	1) 发送控制/数据[n], 并在重发保持缓冲区中存储控制/数据[n]
2)收到错误的控制/数据[n]	
3) 撤销声明 RTS	
4a) 发送错误消息[n] (Err[n]), 并在 TX 重发保持缓冲区中存储 Err[n] 4b) 清空 RX 先进先出队列并且等待 T_{detect} (主机)时间长度	
	5a) 停止进一步传输, 并等待直到 TX 先进先出队列清空(如果允许, 可刷新该先进先出队列) 5b) 收到错误消息[n]
6) 声明 RTS	6a) 撤销 RTS 声明 6b) 清空 RX 先进先出队列并且等待 T_{detect} (控制器)时间长度
7) 检测已撤销声明的 CTS	
8) 停止进一步传输, 并等待直到 TX 先进先出队列清空(如果允许, 可刷新该先进先出队列)	8) 检测到声明的 CTS
	9a) 发送 Err[n]的错误消息, 并在 TX 重发保持缓冲区中存储 Err[n]的错误消息 9b) 声明 RTS
10a) 收到 Err[n]的错误消息 10b) 检测到声明的 CTS	
11) 重发错误消息[n]	
	12) 收到错误消息[n]
	13) 重发控制/数据[n]

11.3 HCI UART 传输层

11.3.1 协议

通过 UART 传输层发送的 HCI 分组共有四种：HCI 指令分组、HCI 事件分组、HCI ACL 数据分组和 HCI SCO 数据分组(参见“HCI 功能规范”)。HCI 指令分组仅能发送到蓝牙主控制器，HCI 事件分组仅能从蓝牙主控制器发出，HCI ACL/SCO 数据分组则从蓝牙主控制器中既能发出，也能接收，如图 11.3 所示。

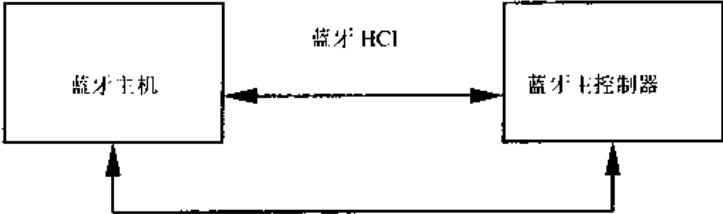


图 11.3 HCI UART 传输层

HCI 不能够区分四种 HCI 分组类型。因此，如果 HCI 分组通过一通用物理接口发出，HCI 分组指示器必须按表 11.15 内容执行。

表 11.15 HCI 分组

HCI 分组类型	HCI 分组指示
HCI 指令分组	0x01
HCI ACL 数据分组	0x02
HCI SCO 数据分组	0x03
HCI 事件分组	0x04

HCI 分组指示应在 HCI 分组前立即发出。所有四种 HCI 分组都有一个长度段，用于确定 HCI 分组有多少字节。当收到整个 HCI 分组时，将要求下一 HCI 分组的 HCI 分组指示器。在 UART 传输层上，只有后面跟 HCI 分组的 HCI 分组指示器可以允许使用。

11.3.2 RS232 设置

HCI UART 传输层对 RS232 的设置，如表 11.16 所示。

表 11.16 RS232 设置

波特率	厂商指定信息
数据位数	8
奇偶校验位	无奇偶校验值
终止位	1 终止位
流控制	RTS/CTS
流完成响应时间	3ms

含 RTS/CTS 的流控制用于阻止临时 UART 缓冲区溢出。由于 HCI 具有用于 HCI 指令、

HCI 事件和 HCI 数据的流控制机制，因此它不应用作 HCI 的流控制。

如果 CTS 为 1，则允许主机/主控制器发送。

如果 CTS 为 0，则禁止主机/主控制器发送。

流完成响应时间定义了从设置 RTS 为 0，到字节流真正结束之间的最大时间。

当处于空 Modem 模式时，RS232 信令应处于连接状态，即本地 TXD 应连接到远端 RXD，本地 RTS 应连接到远端 CTS，反之亦然。

11.3.3 纠错

如果主机或主控制器在 RS232 通信上失去同步，则需要复位。失去同步意味着已检测到错误的 HCI 分组指示器，或 HCI 分组的长度段超出范围。

如果在主机到主控制器的通信中丢失 UART 同步，那么主控制器将发送硬件故障事件，以便将同步错误告诉主机。主控制器将需要从主机接收一个 HCI_RESET 指令执行复位。主控制器也将在主机到主控制器的字节流中使用 HCI_RESET 指令，以实现重新同步。

如果在主控制器到主机的通信中失去 UART 同步，主机将发送 HCI_RESET 指令以复位主控制器。主机也将通过在主控制器到主机的字节流中查找 HCI_RESET 指令的 HCI 指令完成事件，进行重新同步。

第 12 章 蓝牙测试模式

12.1 概述

本章阐述蓝牙设备硬件和低层功能测试的测试模式。测试模式包括发送端测试（连续比特模式分组）和回送测试。

测试模式支持蓝牙发送端和接收端的测试。它主要用于进行无线和基带层的验证/兼容性测试，也可用于例行质检，或生产中和售后测试。

处于测试模式的设备不支持正常操作。基于安全原因，测试模式设计并不提供给用户。因此，不允许在硬件或软件接口上进行数据输出和接收。

1. 测试配置

配置包括一台待测试设备(DUT)和一台测试装置，如图 12.1 所示。当然，也可以使用其他附加测试设备。

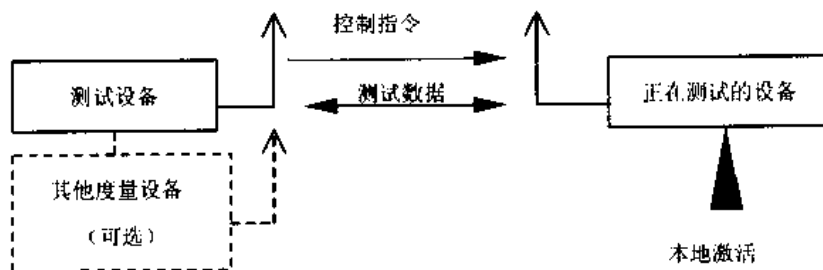


图 12.1 启动测试模式

测试装置和 DUT 组成一个匹克网。其中，测试装置作为主单元，并在测试过程中实现完全控制，而 DUT 则作为从单元。

控制使用 LMP 指令通过无线接口执行(参见“链路管理器协议”一节)。可能存在与 DUT 的硬件接口，但不从属于标准。

测试模式是蓝牙模型的一个特殊状态。由于安全和类型原因，处于测试模式的设备可能不支持正常操作。当 DUT 退出测试模式后将进入待机状态。关机后，蓝牙设备必须回到待机状态。

2. 激活操作

激活操作可以通过硬件或软件接口，或使用无线接口在本地执行。

对于无线接口上的激活操作，由于安全和类型要求的原因，进入测试模式必须在本地启用。该过程的本地实现不从属于标准。测试装置将发送 LMP 指令，以强制 DUT 进入测试模式。DUT 在进入测试模式前将终止所有正常操作。DUT 将在收到激活指令时返回 LMP_accepted 指令。如果本地没有启用 DUT，将返回 LMP_not_accepted。

如果使用硬件或软件接口在本地执行激活操作，DUT 将在进入测试模式前终止所有正

常操作，直到建立到测试装置的连接为止，设备将执行呼叫扫描和查询扫描。推荐使用扩展扫描操作。

3. 控制

可以使用特定 LMP 指令执行控制和配置。如果蓝牙设备没有处于测试模式，则必须拒绝这些指令。在这种情况下，将返回 LMP_not_accepted 指令。当 DUT 处于测试模式时，DUT 将在收到控制指令时返回 LMP_accepted。

处于测试模式的蓝牙设备必须忽略所有与测试模式控制无关的 LMP 指令。处理节能控制和 LMP 特征请求(LMP_features_req)的 LMP 指令将允许在测试模式中执行。常规过程也可用于测试自适应节能控制。

通过 LMP_detach 指令或发送测试情景设置为“退出测试模式”的 LMP_test_control 指令，DUT 将退出测试模式。

12.2 测试环境

12.2.1 发送端测试

蓝牙设备传输连续比特模式。该模式通过周期性按照由测试装置和 DUT 所组成匹克网的从单元 TX 定时器发送分组。每次都重复发送同样的测试分组。

当主单元发送首个 POLL 分组时，发送端测试开始工作。在非跳频模式下，POLL 分组将使用统一频率。

测试装置以其 TX 时隙(控制指令或 POLL 分组)执行发送操作。从单元将以后面的从单元 TX 时隙启动发送操作。主单元轮询间歇是固定值，并被预先定义。即使没有从测试装置接收到分组，正在测试的设备也将按照正常定时，进行其突发通信。

突发通信长度超过一个时隙分组的长度。测试装置可以轮询使用下一空闲主单元 TX 时隙。其定时过程如图 12.2 所示。

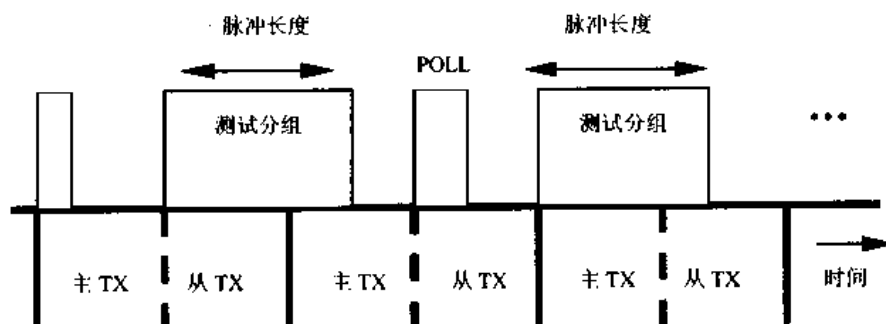


图 12.2 发送端测试定时

1. 分组格式

测试分组为一普通蓝牙分组，有效载荷如图 12.3 所示。

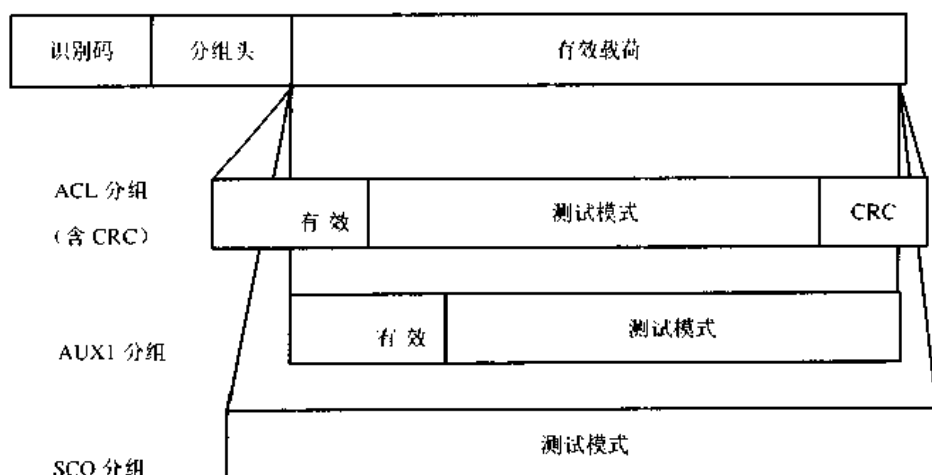


图 12.3 TX 分组通用格式

配置期间，测试装置将定义：

- 要使用的分组类型；
- 有效载荷长度。

对于有效载荷长度，将采用基带规范限制。如果是 ACL 分组，基带规范中定义的有效载荷结构也将保留。

对于发送端测试模式，只能使用没有前向纠错码（FEC）的分组，包括 HV3、DH1、DH3、DH5 和 AUX1 分组。

在发送端测试模式下，测试装置和 DUT 之间的交换分组不会采用伪噪声序列进行加密编码。当 DUT 进入发送端测试模式时将停止伪噪声加密，而当 DUT 退出发送端测试模式时则启用伪噪声加密，如图 12.4 所示。

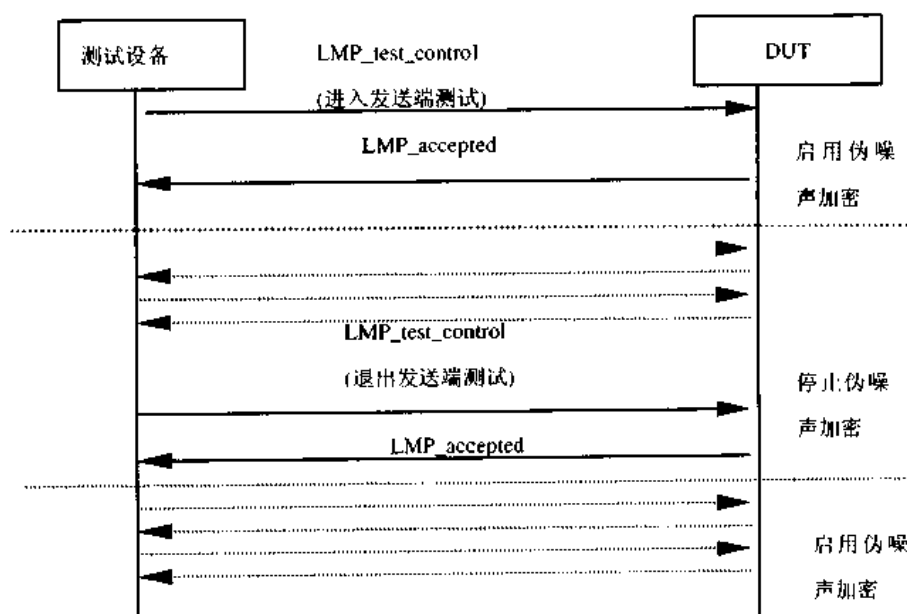


图 12.4 在传输模式中使用伪噪声

2. 伪随机序列

在伪随机序列情况下，每一次传输将使用同一位序列(即分组重复发送)。使用 PRBS-9 序列。该序列可以用 9 级移位寄存器生成，其中将第 5 级和第 9 级输出进行模 2 加，并且将结果又返回到第 1 级中，如图 12.5 所示。序列以 9 个连续 ONE 的首个 ONE 开始，即移位寄存器有 9 级初始化。

- 移位寄存器级数： 9
- 伪随机序列长度： $2^9 - 1 = 511$ 位
- 最长的零序列： 8 (NON-INVERTED 信号)

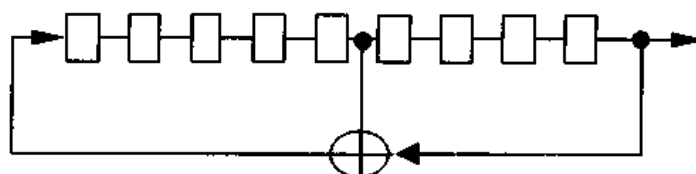


图 12.5 用于生成 PRBS 序列的线性反馈移位寄存器

3. 递减的跳频序列

为了在完整频率范围上支持快速无线测试，定义了递减跳频模式。该模式对于蓝牙设备和模块可选。

递减跳频只使用五种可执行时序跳频的频率(使用信道 0、23、46、69 和 93)，如图 12.6 所示。

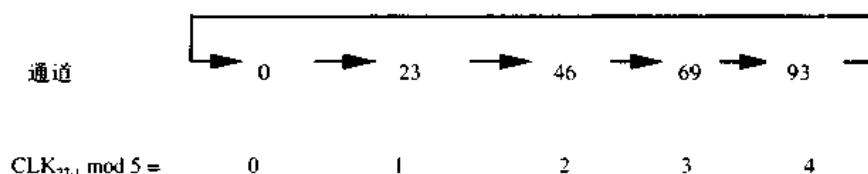


图 12.6 递减跳频方案

定时基于测试装置的蓝牙时钟。CLK_{27,1}（而不使用 CLK₀ 表示半个时隙）作 5 取模运算的结果值用于确定传输频率。

4. 控制传送的参数

下列参数用于设置发送端测试。

a. 位模式

- 常数零；
- 常数 1；
- 交互 1010...；
- 交互的 1111 0000 1111 0000...；
- 伪随机位模式；
- 传输完成。

b. 选择频率

- 单频率；

- 欧洲/美国跳频模式;
- 日本跳频模式;
- 法国跳频模式;
- 西班牙跳频模式;
- 递减跳频(对于蓝牙设备和模块可选)。

c. TX 频率

- $k \Rightarrow F : =(2402 + k) \text{ MHz}$

※注: 该频段用于测试覆盖正常 79 个通道的整个频段。西班牙、法国和日本跳频方案与此相同。频率指定规则与固定 TX 频率 ($f=(2402+k) \text{ MHz}$) 相同。

d. TDD 帧($n * 1.25 \text{ ms}$)的缺省轮询周期

e. 分组类型

f. 测试序列的长度(参见基带规范的用户数据分组定义)

5. 节能控制

如果测试自适应节能控制, 将使用常规 LMP 指令。DUT 将以最大功率开始传输, 并随着收到每一指令而减少/增加其所用功率。

6. 不同频率设置之间的切换

当 LMP 过程完成时, 频率选择切换将有效。

在收到 LMP_accepted 消息后, 测试装置将切换到新的频段或跳频模式。

在发出 LMP_accepted 消息后, DUT 将进行切换。

注意: LMP_accepted 分组丢失将最终导致频率同步丢失, 且不能恢复。改变跳频模式时, 正常操作中也会发生类似问题。

12.2.2 回送测试

处于测试的设备可接收常规基带分组。收到的分组将在 DUT 中解码, 并且将使用同一分组类型返回有效载荷。返回分组将在测试装置传输后的 TX 时隙中发回, 或者它将被推迟并将在测试装置下一次传输后时隙中发回。

当然, 也可以实现一个推迟回送。然后, 返回分组将被推迟到下一个 TX 时隙。没有指令可以确定或控制该模式。可以通过其他手段来固定和调整设备动作, 但不能随意改变。

测试装置能够选择启用或停止伪噪声加密。该设置可同时保持上行或下行链路。对于伪噪声状态切换, 采用图 12.4 的同样规则。

回送测试将使用下列规则, 参见图 12.7~图 12.10。

- 如果没有检测到同步字, 将不会应答;
- 如果头检错(HEC)失败, DUT 将使用含 ARQN 位的 NULL 分组, 该 ARQN 位设置为 NAK。但并不是必须返回该 NULL 分组, 也可以什么都不返回;
- 如果分组包含一条与测试模式控制有关的 LMP 消息, 则执行该指令, 且不返回该分组, 尽管作为正常程序也将返回 ACK 或 NAK。其他 LMP 指令将被忽略, 且不返回任何分组;
- 有效载荷的前向纠错码(FEC)可被解码, 并可由于传输而再次编码。这将允许进行 FEC 处理测试。如果可确定真正的误码率, 测试装置将选择一种不含前向纠错码(FEC)的

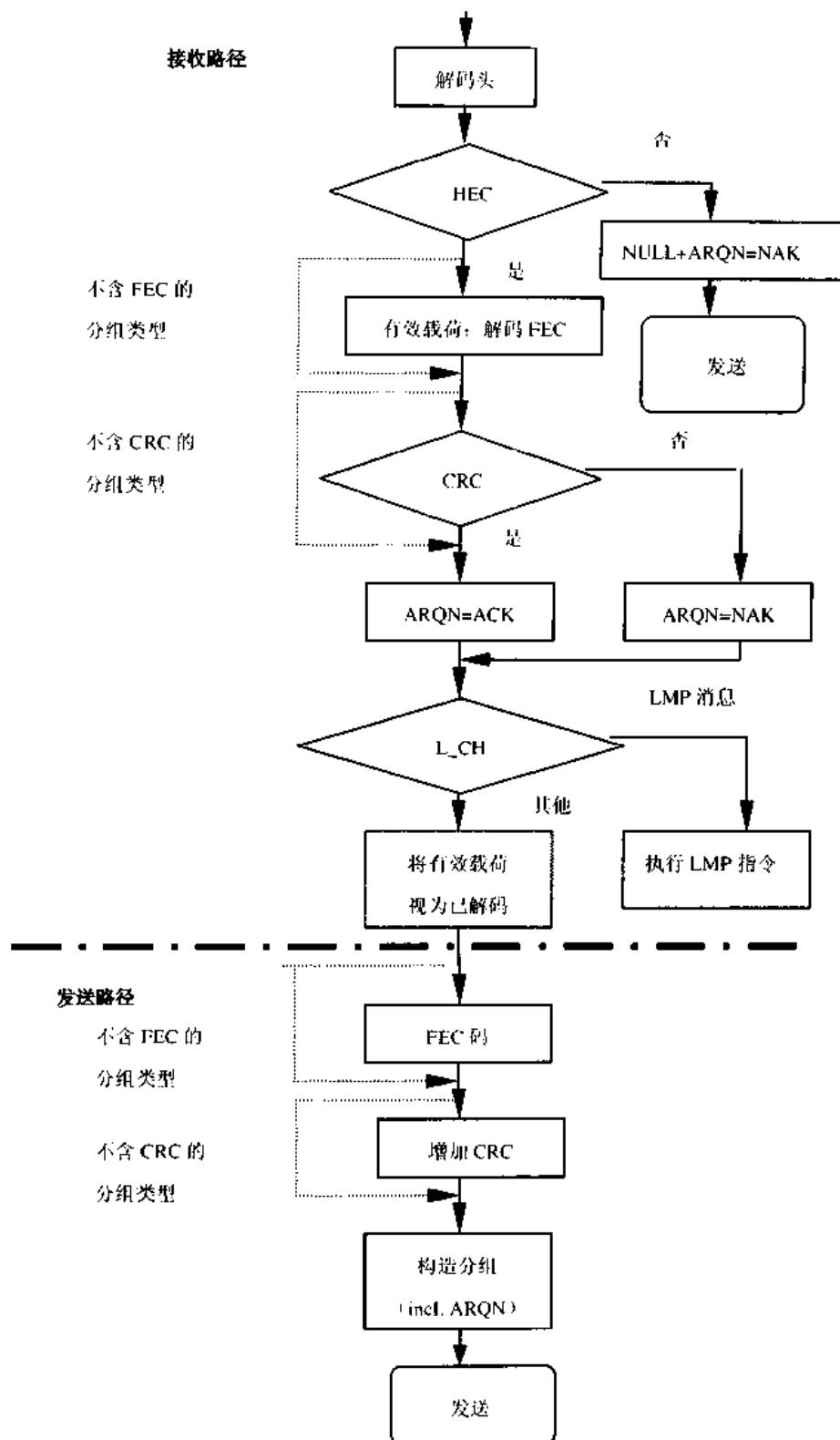


图 12.7 伪噪声加密可以与正常激活模式的相同方式执行

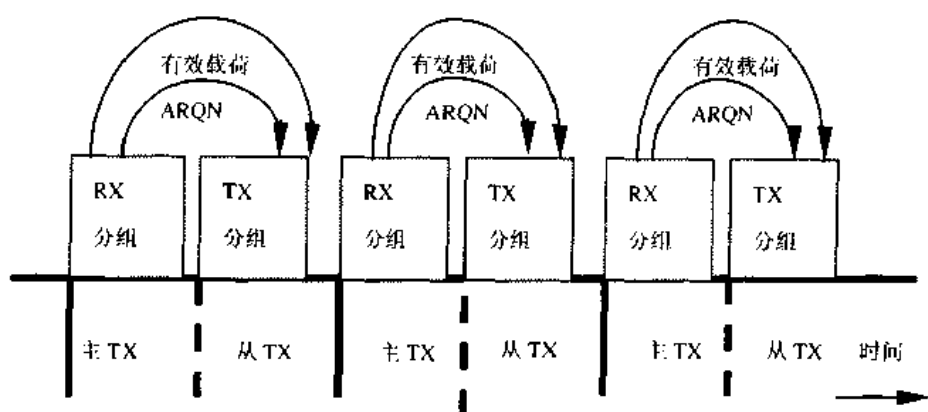


图 12.8 正常回送中有效载荷和 ARQN 的处理

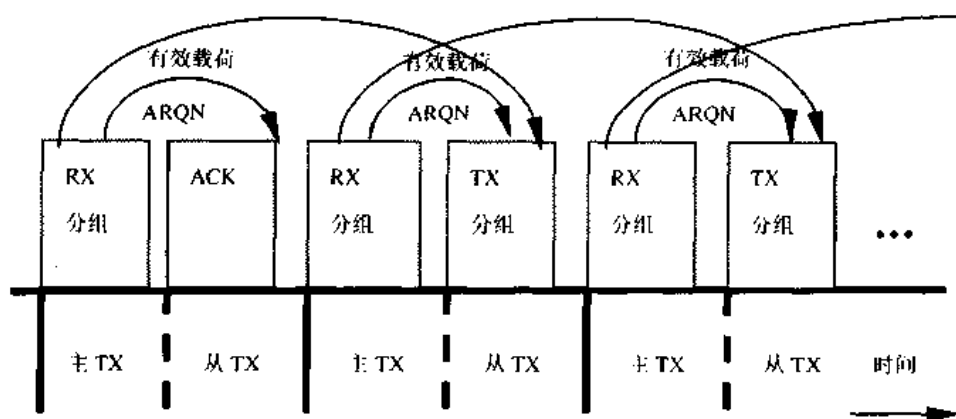


图 12.9 延迟回送中的有效载荷和 ARQN 的处理——开始

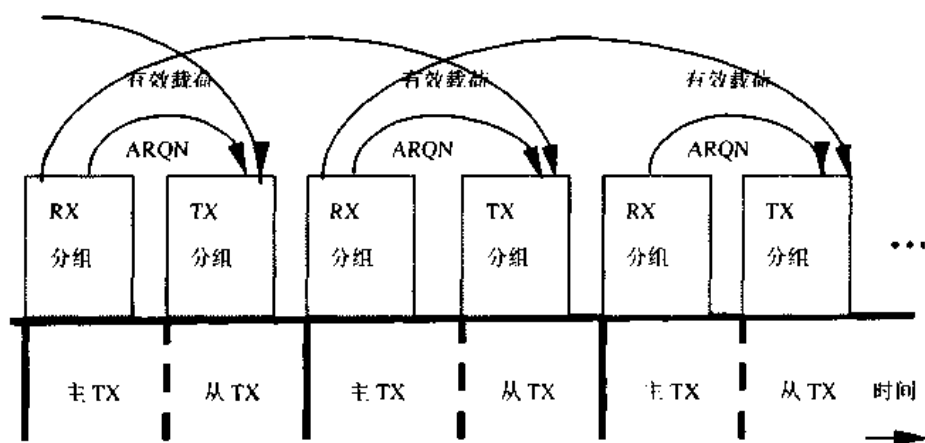


图 12.10 延迟回送中的有效载荷和 ARQN 的处理——结束

分组类型:

- CRC 的评价: 失败时, 将返回 ARQN=NAK 的有效载荷。返回分组的 CRC 将用于返

回有效载荷计算：

- 如果 CRC 失败，则回送在有效载荷头(可能错)中显示的字节数。

以下参数将用于配置回送测试。

a. 分组类型

- ACL 分组；
- SCO 分组；
- 不含伪噪声的 ACL 分组；
- 不含伪噪声的 SCO 分组。

b. 频率选择

- 单频率(独立于 RX 和 TX)；
- 欧洲/美国跳频；
- 日本跳频；
- 法国跳频；
- 西班牙跳频；
- 递减跳频(可选)。

c. 功率水平（根据无线规范要求而使用）

- 节能控制或设定 TX 功率。

d. 频率设置的切换可以参照发送端测试执行(参见 12.2.1 节)

12.3 LMP 消息概览

表 12.1、表 12.2 和表 12.3 分别列出了所有用于测试模式的 LMP 消息、测试参数和 PDU 参数设置的限制。

表 12.1 所有用于测试模式的 LMP 消息

LMP PDU	PDU 号	可能方向	内容	在有效载荷中的位置
LMP_Test_Activate	56	M → S		
LMP_Test_Control	57	M → S	测试情景 跳频模式 TX 频率 RX 频率 节能控制模式 轮询周期 分组类型 测试数据长度	2 3 4 5 6 7 8 9-10
LMP_Detach	7	M → S		
LMP_Accepted	3	M ← S		
LMP_not-accepted	4	M ← S		

表 12.2 用于测试模式的测试参数

名称	长度 (字节)	类型	单元	细节
测试说明书	1	U-INT8		0 暂停 (TX OFF) 1 发送端测试-0 模式 2 发送端测试-1 模式 3 发送端测试-1010 模式 4 伪随机位序列 5 回送关闭 ACL 分组 6 回送关闭 SCO 分组 7 不含伪噪声的 ACL 分组 8 不含伪噪声的 SCO 分组 9 发送端测试-11110000 模式 10~254 保留 255 退出测试模式
跳频模式	1	U-INT8		0 单频 RX/TX 1 欧洲/美国跳频 2 日本跳频 3 法国跳频 4 西班牙跳频 5 递减跳频 (可选) 6~255 保留
TX 频率 (对于 DUT)	1	U-INT8		$F=[2402+K]\text{MHz}$
RX 频率 (对于 DUT)	1	U-INT8		$F=[2402+K]\text{MHz}$
节能控制模式	1	U-INT8		0 固定 TX 输出功率 1 自适应功率控制
轮询周期	1	U-INT8	1.25ms	
分组类型	1	U-INT8		在包头中编号, 参见基带规范
测试序列长度 (=基带规范中用户数据长度)	2	U-INT16	1 个字节	无精度二进制数

表 12.3 用于 LMP_Test_Control PDU 的参数限制

参数	发送端测试限制	回送测试限制
TX 频率	$0 \leq K \leq 93$	$0 \leq K \leq 93$
RX 频率	与 TX 频率相同	$0 \leq K \leq 93$
轮询周期		不适用 (设置为 0)

续表

参数	发送端测试限制	回送测试限制
测试序列长度	取决于分组类型 DH1: ≤ 28 个字节 DH3: ≤ 181 个字节 DH5: ≤ 339 个字节 AUX1: ≤ 29 个字节 HV3: ≈ 30 个字节	不适用 (设置为 0)

第 13 章 蓝牙兼容性要求

13.1 概述

1. 适用范围

蓝牙销售厂商和技术采用厂商已分别签署了蓝牙推广协定和蓝牙采用者协定。这些协定授权销售厂商和技术采用厂商生产符合本规范的许可。

蓝牙兼容性要求规定了销售厂商和技术采用厂商必须遵守的要求，即产品必须符合的规范，而且该产品必须是由销售厂商和技术采用厂商协定各自授权认证的。

蓝牙认证是指由一销售厂商和技术采用厂商说明其产品符合要求规范的过程。本文件提供规范要求和蓝牙认证的介绍。其他细节可访问蓝牙网站。

一般要求和官方推荐要求不在本文件适用范围内。

2. 使用术语

蓝牙商标 (Trademark): 在销售厂商和技术采用厂商协定中定义。

蓝牙商标 (Brand): 包含在“蓝牙商标手册”中说明的所有商标元素，与 Bluetooth Trademark 等同。

蓝牙标志: 该商标参见“蓝牙商标手册”中的图形标志。

蓝牙许可: 由销售厂商和技术采用厂商协定定义，兼容于本规范并由本规范授权的所有权利，如蓝牙专利许可和蓝牙商标许可。

蓝牙专利许可: 构成专利权的，或由此在销售厂商和技术采用厂商协定中定义的蓝牙许可的应用部分。

蓝牙商标许可: 构成在销售厂商和技术采用厂商协定中定义的商标权的蓝牙许可的应用部分。

协议规范: 定义某层两个对等设备间的通信。

框架规范: 定义对应于某一蓝牙使用模型的协议栈的用法。

蓝牙认证进程: 由生产厂商展示其兼容于蓝牙规范的规则和步骤。

蓝牙认证计划: 蓝牙认证进程的实现。

蓝牙认证评价协会 (BQRB): 负责管理、评价和推进蓝牙认证计划。蓝牙 SIG 将指定 BQRB 早期成员。

蓝牙认证测试装置 (BOTF): 一个由 BQRB 官方授权以测试蓝牙产品的测试工具。

蓝牙认证人员 (BQB): 由 BQRB 授权的人员，以负责检查不符合规范的声明和文档，评价产品测试报告，在蓝牙授权产品官方数据库中列出该产品。

蓝牙认证管理员 (BQA): 一个代表 BQRB 的负责管理蓝牙认证计划的人员。

实现一致条款 (ICS): 由厂商提交用于认证的、附属于产品的文档。该文档详细阐述了实现的蓝牙功能。

蓝牙伙伴：等同于销售厂商和技术采用厂商。

3. 法律问题

如何使用蓝牙商标元素的规则和准则将在蓝牙网址的《蓝牙商标手册》文件中描述。

据我们了解，已建立了蓝牙规范来满足世界范围的一般性要求。一般性认证不是蓝牙认证要求的一部分，而是所有市场的要求。每个生产商的惟一任务是确保其产品具有市场所接受的一般性功能。

产品必须完成蓝牙认证，以满足“兼容于规范”的要求。由销售厂商和技术采用厂商协定授权的蓝牙许可只对经认证的产品有效，而不能转移到其他产品。

本文中，“蓝牙许可”有时由于操作性原因，可划分为“蓝牙专利许可”和“蓝牙商标许可”。这些条款分别与销售厂商和技术采用厂商协定中的“必要声明”和“商标”相对应。

正如销售厂商和技术采用厂商协定所规定，任何负责生产或销售包含蓝牙接口元素产品的厂商，如果有不遵循蓝牙的规范，或者任何未完成蓝牙认证的包含蓝牙接口的产品，都必须经正式批准后才能生效。

规范 1.0 发布后，蓝牙 SIG 保留增加蓝牙新标准的权利。

蓝牙商标许可由爱立信向使用与兼容于规范的产品相关联商标的成员授权。

而且，爱立信在它已注册该商标的国家，基于该商标的使用向各成员提供一定的成本和费用补偿。但爱立信不对该产品承担任何责任，无论该责任是由产品造成的人员或财产损失，或产品自身缺点。

4. 蓝牙商标价值

本文的目标是定义蓝牙兼容性要求。应时刻铭记基本蓝牙哲学：“让无线连接变得更轻松”。重要的最终用户实践范例有：

- 可靠的高质量无线链路；
- 不同品牌产品之间的互操作性；
- 易理解的产品功能。

可靠的无线链路取决于所有产品是否兼容于蓝牙无线链路性能规范。互操作性由协议和标准实现准则来实现。易用性则取决于蓝牙产品能力的清晰、一致的文档描述。以上所有元素都在蓝牙兼容性要求中有阐述。

13.2 蓝牙认证计划

本节阐述了蓝牙认证申请者必须执行的蓝牙认证计划框架。如果完成，完整蓝牙认证计划将在蓝牙网站公布。

蓝牙认证计划建立制造商显示其蓝牙规范兼容性的规则和程序，以及由产品制造商和分销商使用蓝牙许可的进程。该计划定义如下实体：

蓝牙认证评价协会 (BQRB)：负责管理、评价和推进蓝牙认证计划。蓝牙 SIG 将指定 BQRB 早期成员。

蓝牙认证管理员 (BQA)：一个代表 BQRB 的负责管理蓝牙认证计划的人员。

蓝牙认证测试装置 (BOTF)：一个由 BQRB 官方授权以测试蓝牙产品的测试工具。

蓝牙认证人员 (BQB)：由 BQRB 授权的人员，以负责检查不符合规范的声明和文档，

评价产品测试报告，在蓝牙授权产品官方数据库中列出该产品。

蓝牙认证计划的功能和关系如图 13.1 所示，认证进程综述如下。

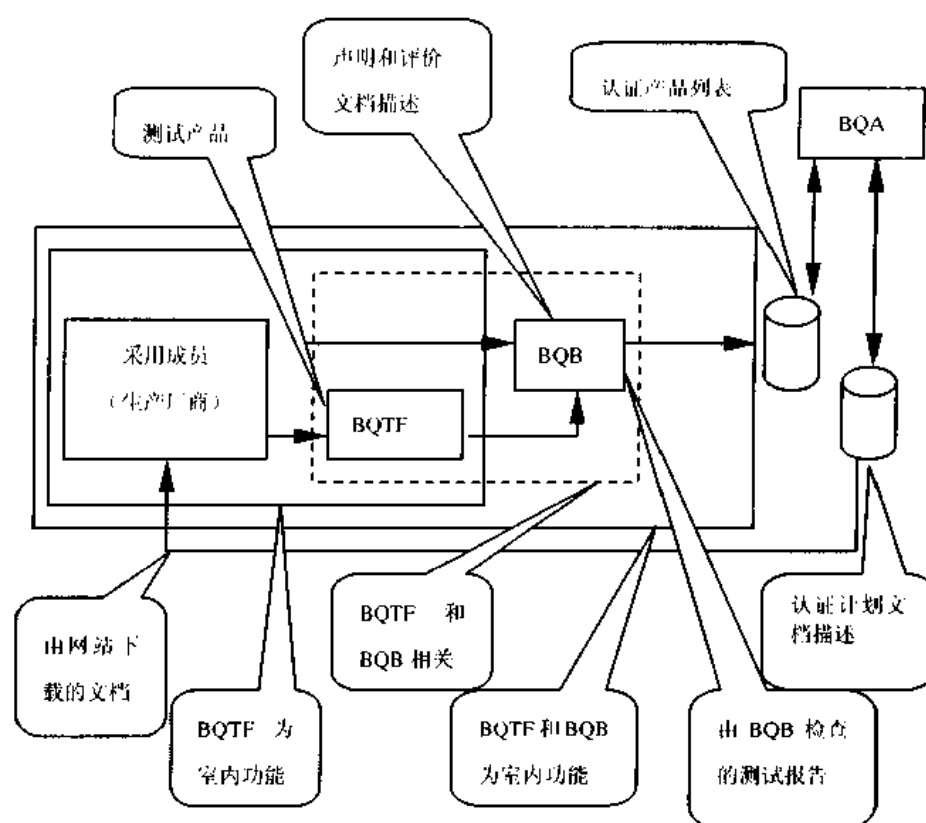


图 13.1 蓝牙认证进程

a. 使用成员将用于蓝牙认证的产品递交 BQTF。制造商必须增加临时接口或功能以便能够测试所有蓝牙功能实现。BQTF 不负责提供任何其他系统用于测试，如 LAN、PSTN 或 GSM 网络，同时应提供必要的文档。例如，产品说明、用户手册和技术实现说明(蓝牙站点有该类文件模板)。BQTF 将根据当前测试规范和 BQRB 政策，测试每一个在技术实现说明中声明的蓝牙特性，并准备一份测试报告。

b. 测试结果和产品文档将送往 BQB。采用成员将一申请发往 BQB，以请求该产品标识为“蓝牙认证”。申请应当包括：

- 准确的产品描述；
- 蓝牙产品规范兼容性声明（包含完整文档）和一册由采用成员当前负责人签署的蓝牙商标手册。

c. 当申请完成时，BQB 发出一认证产品通知，并经申请者同意在官方蓝牙认证产品数据库中列出该产品，以使所有采用成员查阅。

BQTF 可以是第三方测试机构，也可以是采用成员的内部机构。而且 BQB 也可以是内部或外部的。但 BQTF 和 BQB 都必须由 BQRB 授权。

制造商负责与 BQTF、BQB 和 BQA（如果必要）建立必要的保密协议。在 BQB 不能确定兼容性的情况下，BQB 在征得申请者同意后，可以向 BQA 提交有关信息接受指导。在必须咨询 BQRB 的情况下，根据 BQRB 指导，将要求申请者准备采纳。

采用成员将从 BQTF 和 BQB 得到各自服务和费用的发票。BQRB 将收取一定费用以资助与认证计划有关的管理机构。开始，该费用为每产品 3000 美元。为反映实际成本，将每年进行调整。

13.3 蓝牙产品许可要求

本节概述完全符合蓝牙认证的产品要求。产品要求分为：

- 蓝牙无线链路要求；
- 蓝牙协议要求；
- 蓝牙框架要求；
- 蓝牙信息要求。

13.3.1 蓝牙无线链路要求

蓝牙无线链路将满足测试规范文档中描述的最小化要求。这是建立和维持蓝牙技术作为短距离无线链路的最好选择。蓝牙无线链路要求的测试规范将基于蓝牙规范的无线规范两部分。

BQRB 将列出 BQTF，允许不按照蓝牙性能要求进行产品认证。

13.3.2 蓝牙协议要求

蓝牙协议栈低层实现应满足测试规范文档中描述的最小化要求。为了验证要求得到满足，将执行单独协议测试。该验证过程通过蓝牙测试控制器接口 TCI 访问协议高层接口实现。如何在验证过程中使用测试控制器接口在测试规范中有所阐述。

蓝牙协议要求的测试规范将基于蓝牙规范的基带、链路管理器、逻辑链路控制和适配三个部分，并且如果适用，还应包括主控制器接口部分。

BQRB 将列出 BQTFS，允许不按照蓝牙性能要求进行产品认证。

允许生产厂商修改产品硬件和软件，以执行协议测试。如果执行完毕，制造商必须保证，在实际产品中蓝牙规范的基带、链路管理器、逻辑链路控制和适配层部分，以及主控制器接口（如果可用）部分得到统一实现。

13.3.3 蓝牙框架要求

蓝牙产品将满足在测试规范中为每一框架定义的某个最小化蓝牙框架要求。这将保证最终用户能够受益于不同产品之间的互操作性。蓝牙框架要求的测试规范将基于蓝牙规范的框架规范部分。

下列通用蓝牙框架要求必须满足：

- 必须遵循“通用访问”框架；
- 在技术实现文档中必须就实现的蓝牙服务进行描述；
- 在技术实现文档中已声明的所有蓝牙框架，必须根据每一框架规范来实现；
- 应实现所有蓝牙标准角色的必须性，所有蓝牙标准的可选特性则根据框架规范来实现；
- 如果存在蓝牙框架的一种服务被实现，则它必须根据该框架进行。允许向框架改进或增加特性，只要它能够与其他实现上节所述标准框架的产品实现互操作性。改进的或新的特

性只能在两蓝牙设备之间适当协商后被激活。

注：采用成员如果要想实现一种新的服务，却没有足够的标准化蓝牙框架规范可用的话，则允许这样做。但是，该新的服务将绝对不能以符合标准蓝牙框架和部分蓝牙规范的方式进行引用。制造商必须告知市场，以使最终用户可以在清晰和一致的方式下了解该产品在通用互操作性方面的局限性。

BQRB 将列出 BQTFs，允许不按照蓝牙性能要求进行产品认证。

13.3.4 蓝牙信息请求

制造商必须以清晰和一致的方式告知市场和最终用户有关所实现蓝牙功能的信息。

允许不按照蓝牙性能要求进行产品认证。

13.3.5 蓝牙外设产品要求

蓝牙外设产品定义为“销售到最终用户的，至少包含蓝牙无线和基带的硬件，但并非独立产品的产品。在主机系统内安装后，该产品以完整蓝牙产品的方式运行”。蓝牙外设产品的例子有 PC 卡、串口加密狗、USB 加密狗。

蓝牙外设产品也必须通过完整蓝牙认证过程。为便于测试，蓝牙外设产品和提供的蓝牙软件将安装在一由制造商提供的主机设备中。

蓝牙外设产品的认证应遵循前面已描述的认证过程。

13.5.6 蓝牙部件要求

蓝牙部件定义为“一种设计和销售用于组成完整蓝牙产品（至少包含一已存在蓝牙标准子集）的，不能作为完整蓝牙产品作用的部件产品”。例如，蓝牙部件可能是一个设计用于集成于 PC 卡上的完整模块，或者是一块实现所有蓝牙基带和协议功能的集成电路。

蓝牙部件一般都是由 OEM 厂商购买和集成到一个销售到最终用户的产品中。

蓝牙部件制造商一般将获得有限的蓝牙许可，以使制造商能标识该部件的蓝牙功能。部件制造商也希望通过基于部件的参考性设计的一次性认证测试，最小化其 OEM 客户的认证测试要求。只要最终产品能够认证，部件认证并非必须。部件认证只是可以更好地促进销售。

蓝牙部件认证蓝牙部件必须以申请表中注明的参考性设计配置，并通过完整蓝牙认证过程。

配置有限的认证蓝牙部件的蓝牙产品也必须通过完整蓝牙认证过程。但是，如果某些测试已在部件的有限认证许可中通过的话可以得到免除。

部件的有限蓝牙许可可以说明某些认证测试已由使用最终用户产品部件的 OEM 制造商进行预先认证了。这些在其参考性设计中执行蓝牙部件认证测试的认证测试由 BQTF 说明。BQTF 经过咨询制造商基于部件的统一设计特点对这些测试作出说明。

蓝牙部件认证包括集成蓝牙部件的产品必须按以上所述进行认证，如果某些测试已在部件的有限认证许可中通过的话可以得到免除。

13.3.7 蓝牙许可条款

1. 早期产品的蓝牙许可条款

早期产品认证的过程和条件将在蓝牙网站定义并公布。

2. 特殊产品和市场营销的蓝牙商标许可条款

a. 蓝牙开发工具和演示

蓝牙开发工具和演示是用于开发商用的蓝牙产品，或者在某应用中展示蓝牙技术的产品。两者都不能出售给普通消费者。

产品的生产商或销售商应当明确通知顾客，这些产品只是用于开发或演示目的，而且它们不能通过蓝牙规范认证。

不需要由 BQTF 执行的认证测试。该认证基于申请者规范和商标手册的兼容性声明。

b. 市场营销

蓝牙商标元素可用于市场营销和产品宣传，但应遵循蓝牙商标手册的规则。

如果蓝牙商标用于一个已存在的产品，但并不是每一个人都必须明确该产品并不包含蓝牙发射器，但该产品必须有一个明确的声明(如：具有蓝牙商标的一计算器必须具有一可见声明，以使内置蓝牙的计算器具有意义)。

13.4 有关蓝牙产品功能信息的建议

除商标手册中给出的要求以外，建议至少应将以下信息提供给市场和最终用户。

- 至少应在产品外包装中简短声明产品的蓝牙功能；
- 用户手册(或相关信息)应当包含说明所有蓝牙功能的章节。如果适用，也应包括实现框架的修订号。对于早期产品，应包含一个互操作性产品列表，而不仅仅是标准。

最终用户信息应在用户手册(用户指南)、传单、包装盒和其他广告资源中。

可在附录 A 中找到用户手册信息实例。

蓝牙 SIG 虽尚未批准的新标准，但不可误认为包含了蓝牙规范的标准。对于新标准，制造商应告知市场和最终用户具有互操作性的其他产品。

13.5 质量管理、配置管理和版本控制

当大规模生产某一类产品时，各制造商必须保持高质量，并必须为此负责。投入市场的产品必须经过认证。

蓝牙认证覆盖不同产品硬件和软件版本。产品制造商通过保持适当质量管理和配置管理计划，保证所有产品部件符合统一认证版本。

与认证产品相关的主要硬件或软件的修订，应文档化并提交 BQB 进行评价。基于制造商的描述，BQB 可确认产品不需要进一步测试，并且允许更新许可可以包含新的版本。在其他情况下，BQB 可以区分一个测试的有限子集。该测试由 BQTF 执行以认证新的版本。

增加的蓝牙功能要求对产品进行新的认证。

第 14 章 测试控制接口

14.1 概述

测试控制接口 TCI 提供访问测试设备或在测试过程中实现高层接口的统一方法。

对于所有蓝牙外设产品、蓝牙部件和蓝牙产品，协议测试将用在底层验证实现的功能，即一致性测试。对于该类测试，需要高层测试设备(UT)来完成测试实现(IUT)。为了使测试设备避免随每个 IUT 或测试系统(SUT) 实现而变化，须强制使用标准控制接口。

制造商必须要做到：

- 采用独立于实现的与 TCI 的接口；
- 提供 IUT 所需适配器(可以是硬件、软件或固件)。

蓝牙测试控制接口，将用来验证蓝牙外设产品、蓝牙部件或蓝牙产品的协议要求。具体而言，TCI 将用于验证以下实现的功能：

- 基带层，BB(与协议有关的部分)；
- 链路管理器协议，LMP；
- 逻辑链路控制和适配协议，L2CAP；
- 主控制接口，HCI。

14.2 描述

测试设备和 SUT/IUT 之间的主控接口有两种类型：

(1) TCI-HCI

该接口在语义和句法上等同于主控制器接口。

(2) TCI-L2CAP

该接口基于 HCI 接口，将在 SUT/IUT 的 L2CAP 层验证中使用。

虽然期望物理通道能为 HCI 指定传输层中的一类，包括 USB、RS232 或 UART。但是，还可以有其他选择。

14.2.1 基带和链路管理验证

对于基带层的链路控制部分和链路管理器层的验证，TCI-HCI 接口将作为测试系统和 SUT/IUT 高层间的接口使用。测试系统通过发送 HCI 命令和从 SUT/IUT 接收 HCI 事件访问 SUT/IUT 高层接口，如“主控制器接口功能规范”所述。TCI-HCI 上的支持功能取决于 BB 和 LM 层的功能实现。

用于测试设备和 SUT/IUT 之间的传输层通道有两种类型：

(1) 物理通道为 USB、RS232 或 UART 三者之一。推荐在 SUT/IUT 和测试设备之间将这三类物理通道作为传输层通道使用。

(2) 软件传输通道，即测试设备和 SUT/IUT 之间没有物理连接。在这种情况下，SUT/IUT

制造商必须在设备发送测试数据时，提供能由测试操作员操作的测试软件。操作员将从测试设备接收命令，并在 SUT/IUT 上执行。软件必须能提供同样的功能，就如用物理通道使用 TCI-HCI 一样。软件接口的使用必须由 SUT/IUT 制造商和和执行验证的测试机构双方同意。测试机构可以自己制定对于该接口的要求。

图 14.1 实现了不支持 HCI 的蓝牙产品 BB 和 LM 验证的可能测试配置，该蓝牙产品为 TCI-HCI 使用一个物理传输通道。在该图中，当 TC（测试控制）软件用于验证时，制造商必须为 SUT/IUT 提供配置。TC 软件的功能是使独立于实现的接口适配于 TCI-IUT。

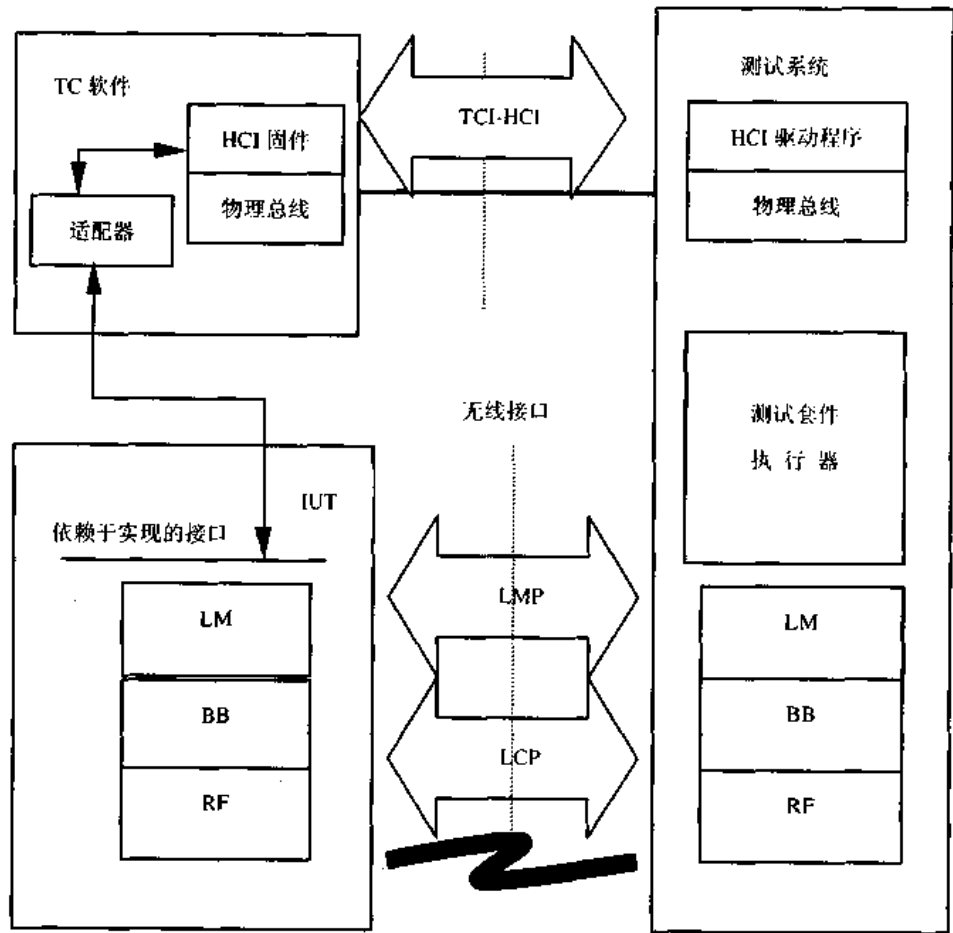


图 14.1 不含 HCI 的基带和 LM 验证——物理传输通道

图 14.2 表示使用 TCI-HCI 接口的软件传输通道的同一蓝牙产品的测试配置。这里，TC 软件的角色在于体现应用可以由测试操作员控制。

14.2.2 HCI 验证

TCI-HCI 接口也可用于 HCI 信令验证。只有制造商指明支持 HCI 功能，才能对 HCI 信令进行验证。

测试设备和 SUT/IUT 之间的传输通道应为 USB、RS232 或 UART 类型中的一种。

图 14.3 表示的是使用对应于 TCI-HCI 接口物理传输的通道时，蓝牙产品的 HCI 验证的可能测试配置。在图中可以看到，不需要额外测试控制软件。而是采用实现的 HCI 作为与测试设备的接口。

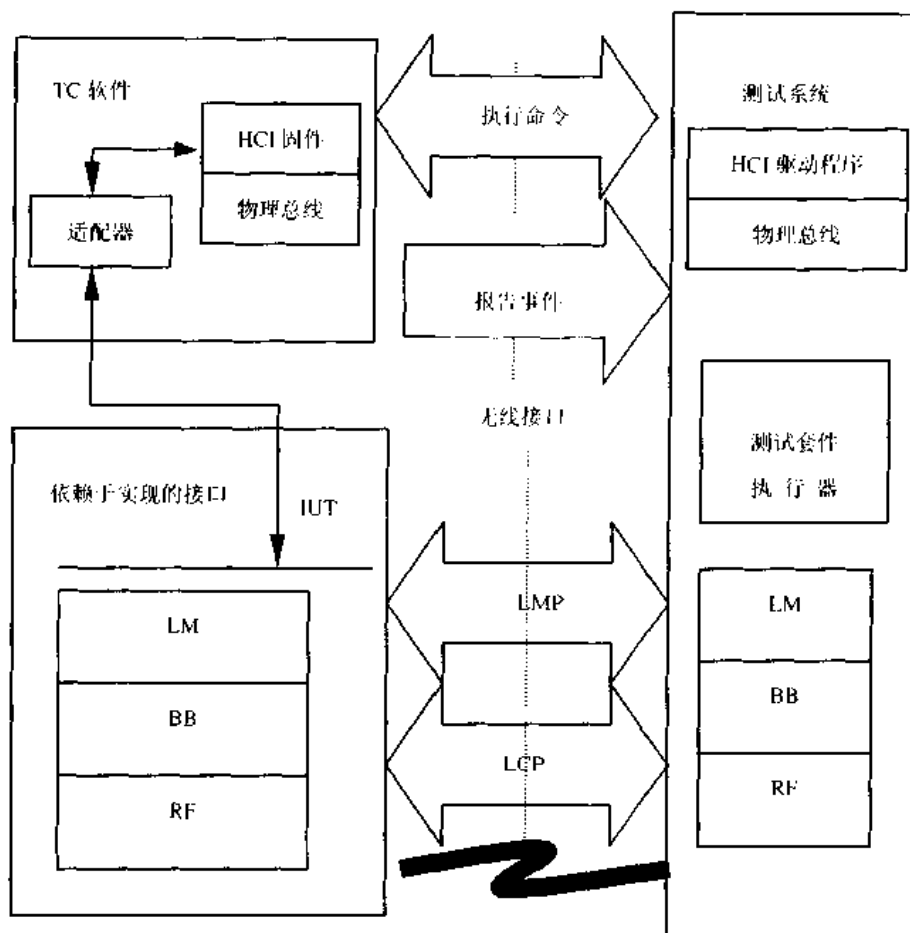


图 14.2 不含 HCI 的基带和 LM 验证——软件传输通道

14.2.3 逻辑链路控制和适配验证

TCI-L2CAP 接口基于 HCI，并且将在 SUT/IUT 的 L2CAP 层验证期间使用。它使用通用事件和命令语法。命令和事件根据 L2CAP 服务接口来定义。

在逻辑链路控制和适配层规范中定义的基本服务可作为参考。但是 L2CAP 基本事件和命令必须转换成与 HCI 事件和命令相同格式的报文。

图 14.4 展示如何使用 TCI-L2CAP 接口物理传输通路查找蓝牙产品的 L2CAP 验证。在该图中，TC 软件在用于验证时，生产商必须为 SUT/IUT 提供配置。TC 软件的功能在于使实现的接口适配于 TCI-L2CAP 接口。

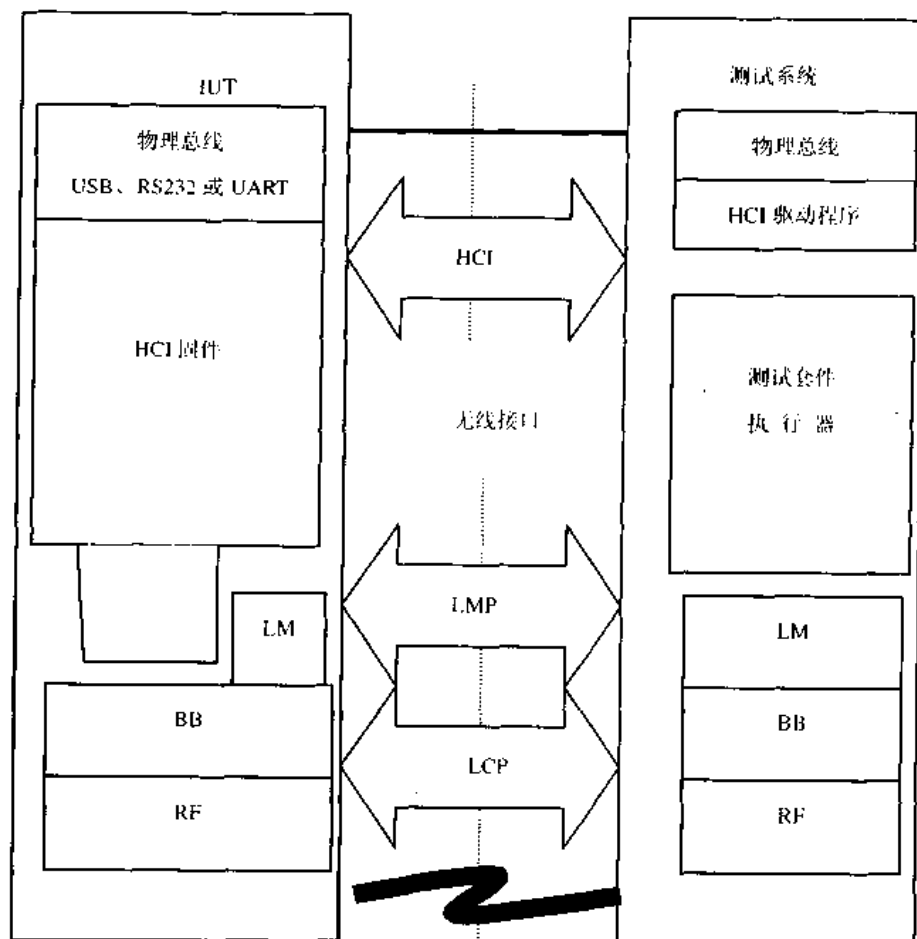


图 14.3 HCI 验证

14.3 测试配置

1. 蓝牙射频链路要求

蓝牙射频链路要求的测试规范基于蓝牙规范，并包含在 SUT/IUT 上执行的相关测试命令内。

对于蓝牙射频链路要求验证，将使用定义的测试模式。

对于此类型验证，只要求无线接口，见图 14.5。由于安全原因，可在本地启用测试模式。

2. 蓝牙协议要求

根据在产品验证中实现蓝牙基带、LM、HCI 或 L2CAP 的不同，需要验证蓝牙协议要求的测试次数也将有所不同。而且，在验证期间使用的 TCI 也可能不同。

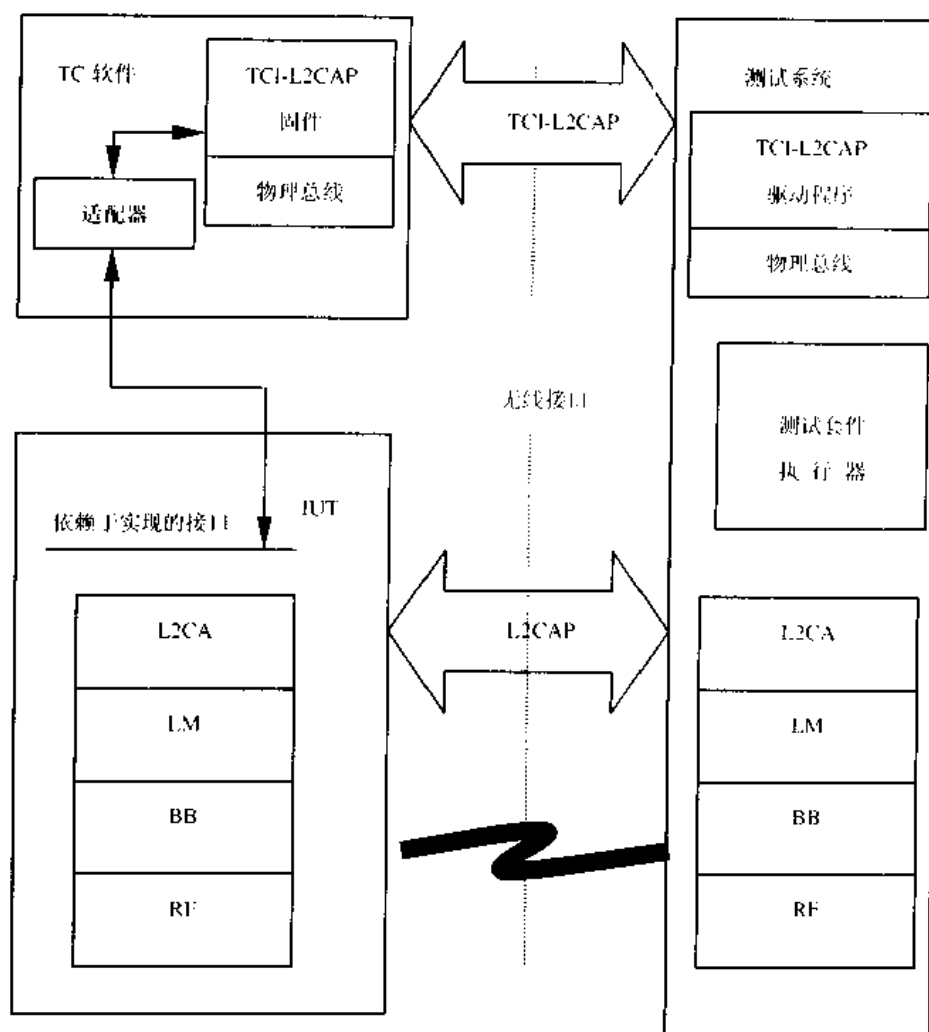


图 14.4 L2CAP 验证

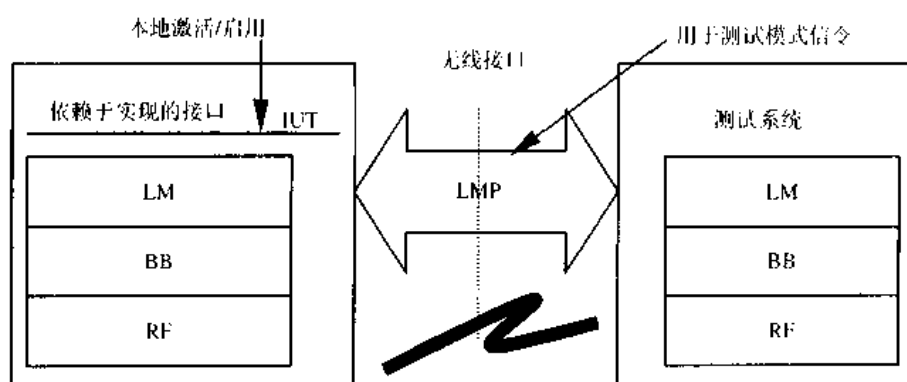


图 14.5 用于 RF 链路要求验证的测试配置

蓝牙协议要求的测试规范基于蓝牙规范，如果合适，它将包含在 SUT/IUT 上执行的相关测试命令内。

对于该验证类型，要求有 SUT/IUT 的无线接口和测试控制接口。

3. 蓝牙标准要求.

对于每一蓝牙产品声明遵循的标准, 将执行的标准测试要以验证蓝牙标准为准, 以保证产品间的互操作性, 例如互操作性测试。

蓝牙标准要求的测试规范基于蓝牙规范, 并包含在 SUT/IUT 上执行的相关测试命令。对于该类验证, SUT 无线接口和支持的 MMI 将用于验证, 见图 14.6。

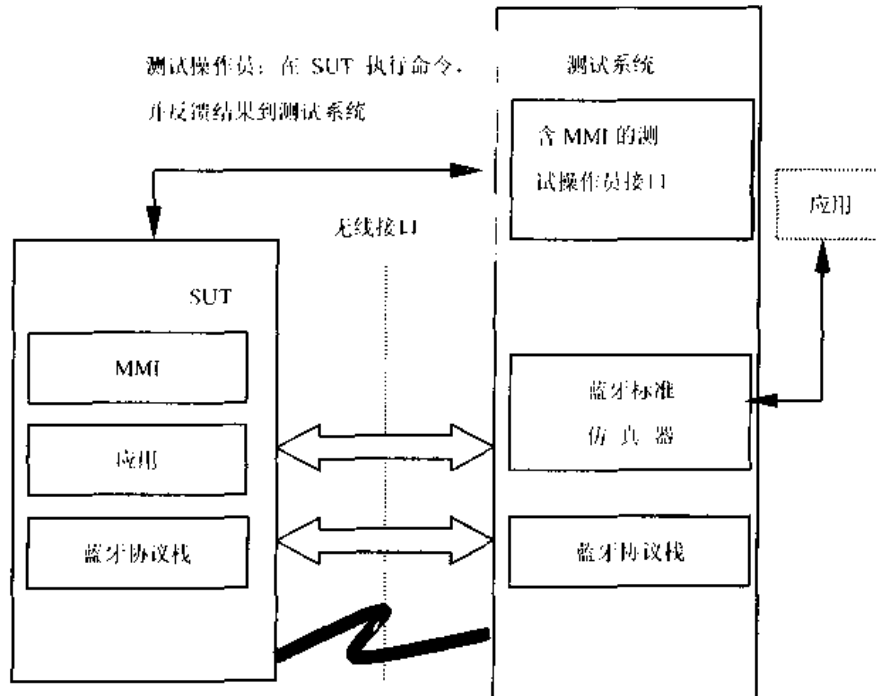


图 14.6 标准要求验证的测试配置

14.4 TCI-L2CAP 描述

14.4.1 事件

在 L2CAP 服务接口中, 命令映射到回送功能。对应应答参数在该功能的返回参数中提交。对于 TCI, 该命令映射为事件和命令应答。

有一个事件代码保留用于测试: 0XFE。为区分 L2CAP 事件, EVENT_ID 参数将作为首个参数提交。该参数为一个字节, 可以形成 256 种可能的事件, 表 14.1 给出分配结果。

测试接口中的事件必须遵循 HCI 语法, 如表 14.2~表 14.5 所示。

表 14.1 事件 ID 分配

EVENT_ID	L2CAP_EVENT
0X00	保留
0X01	逻辑链控制和适配-连接
0X02	逻辑链控制和适配-CONFIGING
0X03	逻辑链控制和适配-DISCONNECTING
0X04	逻辑链控制和适配-QOSVIOLATIONING
0X05~0XFF	保留

表 14.2 连接命令

事 件	事件代码	事件参数
L2CA_ConnectInd	0XFE	Event_ID、BD_ADDR、CID、PSM、标识符

表 14.3 配置命令

事 件	事件代码	事件参数
L2CA_ConnectInd	0XFE	Event_ID、CID、OutMTU、InFlow、FlushTO

表 14.4 连接断开命令

事 件	事件代码	事件参数
L2CA_ConnectInd	0XFE	Event_ID、CID

表 14.5 违反命令

事 件	事件代码	事件参数
L2CA_ConnectInd	0XFE	Event_ID、BD_ADDR

14.4.2 命令

为了将用于 L2CAP 测试的命令与 HCI 命令区分开来，系统将为 L2CAP 测试接口保留一个子集。图 14.7 表示如何对用于测试的 HCI 命令分组的操作域进行编码和解码。

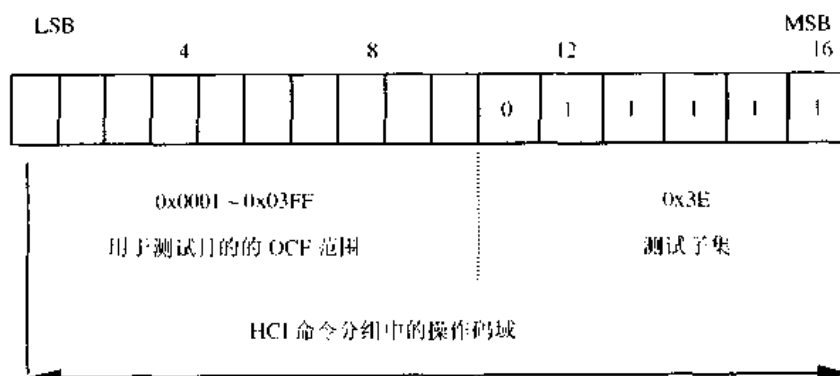


图 14.7 用于测试的 HCI 操作码域值

L2CAP 命令操作码的命令域，OCF 的分配包括在表 14.6 中。

测试接口中的命令遵循 HCI 语法，返回参数使用命令完成事件返回，如表 14.7~表 14.20 所示。

表 14.6 操作码域值分配

OCF	L2CAP 命令
0X0000	保留
0X0001	L2CA_ConnectReq
0X0002	L2CA_DisconnectReq
0X0003	L2CA_ConfigReq
0X0004	L2CA_DisableCLT
0X0005	L2CA_EnableCLT

续表

OCF	L2CAP 命令
0X0006	L2CA_GroupCreate
0X0007	L2CA_GroupClose
0X0008	L2CA_GroupAddMember
0X0009	L2CA_GroupRemoveMember
0X000A	L2CA_GroupMembership
0X000B	L2CA_WriteData
0X000C	L2CA_ReadData
0X000D	L2CA_Ping
0X000E	L2CA_GetInfo
0X000F	保留
0X0010	保留
0X0011	L2CA_ConnectRsp
0X0012	保留
0X0013	L2CA_ConfigRsp
0X0014	保留

表 14.7 连接建立

命 令	OCF	命令参数	返回参数
L2CA_ConnectRsp	0X0001	PSM、BD_ADDR	LCID、结果、状态
描述：请求创建代表一个到物理地址的逻辑连接通道			

表 14.8 连接应答

命 令	OCF	命令参数	返回参数
L2CA_ConnectRsp	0X0001	BD_ADDR、标识符、LCID、 应答、状态	结果
描述：发出连接请求事件命令的应答			

表 14.9 连接释放（断开连接）

命 令	OCF	命令参数	返回参数
L2CA_ConnectRep	0X0002	CID	结果
描述：请求信道断开连接。输入参数为代表本地通道终端的 CID			

表 14.10 配置

命 令	OCF	命令参数	返回参数
L2CA_ConnectRep	0X0003	CID、InMTU、OutFlow、 FlushTO、LinkTO	结果、InMTU、 OutFlow、FlushTO
描述：请求初始化信道配置，以得到新的信道参数设置			

表 14.11 配置应答

命 令	OCF	命令参数	返回参数
L2CAP_ConnectRsp	0X0003	CID、InFlow、OutMTU	结果
描述：发出一个配置请求事件命令的应答			

表 14.12 停止无连接通信

命 令	OCF	命令参数	返回参数
L2CAP_DisableCLT	0X0004	N、PSM 表	结果

表 14.13 启用无连接通信

命 令	OCF	命令参数	返回参数
L2CAP_EnableCLT	0X0005	N、PSM 表	结果

表 14.14 组创建

命 令	OCF	命令参数	返回参数
L2CAP_GroupCreate	0X0006	PSM	CID
描述：请求创建信道标识符，以为多个设备提供逻辑连接。在创建时，组内应为空			

表 14.15 组关闭

命 令	OCF	命令参数	返回参数
L2CAP_GroupClose	0X0007	CID	结果
描述：该命令关闭组			

表 14.16 增加组成员

命 令	OCF	命令参数	返回参数
L2CAP_GroupAddMember	0X0008	CID、BD_ADDR	结果
描述：该命令向组增加一新成员			

表 14.17 删除组成员

命 令	OCF	命令参数	返回参数
L2CAP_GroupRemoveMember	0X0009	CID、BD_ADDR	结果
描述：该命令从组中删除成员			

表 14.18 组成员资格

命 令	OCF	命令参数	返回参数
L2CAP_GroupMembership	0X000A	CID	结果、N、BD_ADDR_LST
描述：获取组成员资格			

表 14.19 PING

命 令	OCF	命令参数	返回参数
L2CAP_PING	0X000D	BD_ADDR、ECHO_DATA	结果、ECHO-DATA

表 14.20 Get Info

命 令	OCF	命令参数	返回参数
L2CAP_GetInfo	0X000E	BD_ADDR、InfoType	结果、INFO_FATA

14.4.3 数据传输

数据传输应以读、写功能来建模，其处理过程与 L2CAP 命令相似。

为了能够发送大量数据，该数据用于验证 L2CAP 实现如何处理大批量数据(如：分段和组装)，由于它不可能象 HCI 事件分组那样使用 HCI 命令分组发送数据，所以它将使用 HCI ACL 数据分组。用于 TCI-L2CAP 接口的发送数据分组的过程/信令将用报文序列图描述。

数据传输的写功能描述如表 14.21 所示。表中，输入参数为 CID、数据长度和数据本身。数据以 HCI ACL 数据分组的方式发送，如图 14.8 所示。

表 14.21 写

命 令	OCF	命令参数	返回参数
L2CAP_WriteData	0X000B	CID、Length、OutBuffer	结果、SIZE

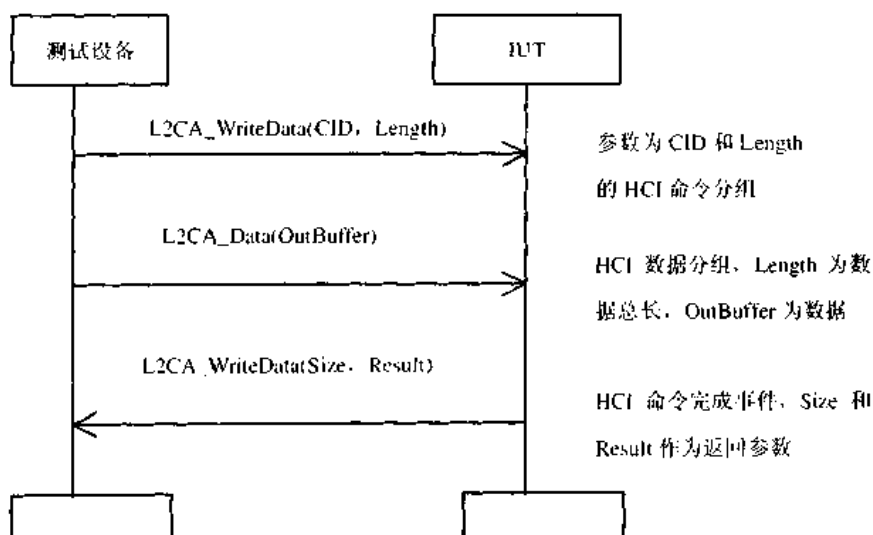


图 14.8 表示如何向 L2CAP 写数据的 MSC

基本 L2CAP_Data 作为在测试设备和 IUT 之间传输数据的抽象用名使用。测试设备将为该数据使用连接句柄 0X0001，且将 FLAGS-FIELD 设置为 0X02。数据总长域将包含 OutBuffer 的长度。

IUT 收到数据后，将返回设置为 0X01 的 N 参数，操作码参数设置为对应 OCF 和子集（即 OCF=0X00B，子集=0X3E）的 HCI 命令完成事件（图中命名为 L2CA_WriteData）。Size 和 Result 在 HCI ACL 数据分组的 Return_Parameters 参数中发送。

数据传输的读功能描述如表 14.22 所示。表中，输入参数为 CID、Length、InBuffer。输出参数是 Result。数据将以 HCI ACL 数据分组的形式发送，如图 14.9 所示。

表 14.22 读

命 令	OCF	命令参数	返回参数
L2CA_ReadData	0X000C	CID、Length、InBuffer	Result

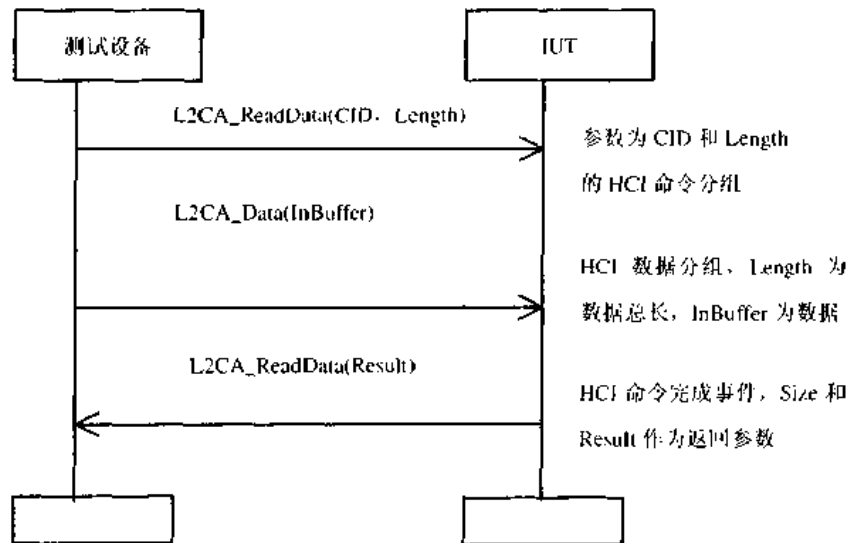


图 14.9 表示如何从 L2CAP 中读取数据的 MSC

基本 L2CAP_DATA 作为测试设备和 IUT 之间传输数据的抽象用名使用。IUT 为该数据使用连接句柄 0X0001，并将标志域设置为 0X02。数据总长域包含 InBuffer 的长度。

IUT 发出数据后，将返回包含设置为 0X01 的 N 参数，操作码参数设置为对应 OCF=0X00C 和子集=0X3E 的 HCI 命令完成事件（图中为 L2CA_ReadData）。其 Size 和 Result 在 HCI ACL 数据分组的 Return_Parameters 域中发出。

缩 略 词

OBEX Object Exchange Protocol 对象交换协议
Blue tooth SIG Bluetooth Special Interest Group 蓝牙特殊兴趣小组
IrDA Infra-red Data Association 红外数据联合会
IrMC Ir Mobile Communications 红外移动通信
IrCOMM Ir Communications 红外通信
IrOBEX Ir Object EXchange 红外对象交换协议
L2CA Logical Link Control And Adaptation 逻辑链路控制和适配
L2CAP Logical Link Control and Adaptation Protocol 逻辑链路控制和适配协议
LSB Least Significant Bit 最低位
MSB Most Significant Bit 最高位
PDU Protocol Data Unit 协议数据单元
SD Service Discovery 服务搜索
SDP Service Discovery Protocol 服务搜索协议
SDDb Service Discovery Database 服务搜索数据库
TCP/IP 传输控制协议/网间网协议
TCS Telephony Control Specification 电话控制协议
GM Group Management 组管理
LM Link Manager 链路管理
LMP Link Manager Protocol 链路管理协议
BB Baseband 基带
CC Call Control 呼叫控制
WUG Wireless User Group 无线用户组
DTMF 双音多频应用
RSSI Received Signal Strength Indication 接收信号场强指示
WAE Wireless Application Environment 无线应用环境
WML Wireless Markup Language 无线标记语言
WAP Wireless Application Protocol 无线应用协议
WDP Wireless Data Protocol 无线数据协议
WTP Wireless Transaction Protocol 无线事务协议
WSP Wireless Session Protocol 无线会话协议
SSL 安全套接字协议
URL Universal Resource Locator 统一资源定位
PPP IETF 点对点协议
HCI Host Controller Interface 主控制器接口
ACL Asynchronous Connectionless Link 异步无连接
BD_ADDR 蓝牙设备地址

DH Data-High Rate Data 高速率数据
 DM Data-Medium Rate Data 中速率数据
 DIAC Devoted Inquiry Access Code 专用查询识别码
 DUT Device Under Test 测试中设备
 DV Data Voice 数据语音
 GIAC General Inquiry Access Code 通用查询访问码
 LAP Low Address Part 低地址段
 LC Link Controller 链路控制器
 OCF 操作码指令域
 OGF 操作码组域
 RF 射频
 SCO Synchronous Connection-Oriented 同步面向连接
 USB Universal Serial Bus 通用串行接口
 COBS 一致开销字节填充法
 TCI Test Control Interface 测试控制接口
 TDD: Time Division Duplex 分时双工
 CAC Channel Access Code 信道访问码
 DAC Device Access Code 设备访问码
 IAC Inquiry Access Code 查询访问码
 CVSD Continuous Variable Slope Delta Modulation,连续可变斜率增量调制
 HV High quality Voice 高保真语音
 ACK Acknowledge 确认
 AG Audio Gateway 语音网关
 AM-ADDR Active Member Address 活动的成员地址
 AP Access Point 接入点
 API Application Programming Interface 应用编程接口
 AR-ADDR Access Request Address 访问请求地址
 BB Baseband 基带
 BCH Bose, Chaudhuri & Hocquenghem Type of code 博斯-乔赫里-霍克文黑姆码
 BER Bit Error Rate 误码率
 BT Bandwidth Time 带宽时间
 CODEC Coder Decoder 编解码器
 COF Cipherring Offset 加密补偿
 CRC Cyclic Redundancy Check 循环冗余校验码
 CTP Cordless Telephony Profile 无绳电话应用
 DC Bias Direct Current Bias 直流补偿
 DCE Data Communication Equipment 数据通信设备
 DCE Data Circuit-Terminating Equipmen 数据电路端接设备
 DCI Default Check Initialization 缺省的检错初始值
 DIAC Dedicated Inquiry Access Code 专用查询码

DT Data Terminal 数据终端
 DUT Device Under Test 在测试的设备
 ETSI European Telecommunications Standards Institute 欧洲电信标准组织
 FCC Federal Communications Commission 联邦通信委员会
 FEC Forward Error Correction Code 前向纠错码
 FH Frequency Hopping 跳频
 FHS Frequency Hop Synchronization 跳频同步
 FIFO First In First Out 先进先出
 FSK Frequency Shift Keying type of modulation 频移键控, 一种调制方式
 FTP File Transfer Protocol 文件传输协议
 FW Firmware 固件 (一般指硬件中的程序)
 GAP Generic Access Profile 通用访问应用
 GFSK Gaussian Frequency Shift Keying 高斯频移键控
 GM Group Management 主管理
 GOEP Generic Object Exchange Profile 通用对象交换应用
 GW Gateway 网关
 HA Host Application SW using Bluetooth 使用 Bluetooth 主机应用软件
 HEC Header-Error-Check 头校验码
 HID Human Interface Device 人机接口设备
 HS Headset 戴在头上的耳机或听筒
 HTTP Hyper Text Transfer Protocol 超文本传输协议
 HW Hardware 硬件
 IEEE Institute of Electronic and Electrical Engineering 电气和电子工程师协会
 IETF Internet Engineering Task Force Internet 工作任务组
 ISM Industrial, Scientific, Medical 工业、科学、医疗
 IUT Implementation Under Test 在测试实现
 L-CH Logical Channel 逻辑信道
 LAN Local Area Network 局域网
 LAP LAN Access Point 局域网接入点
 LAP Lower Address Part 低位地址部分
 LC Link Controller 链路控制器
 LCP Link Control Protocol 链路控制协议
 LCSP Link Controller Service Signaling 链路控制器服务信令
 LFSR Linear Feedback Shift Register 线性反馈移位寄存器
 LIAC Limited Inquiry Access Code 有限查询访问码
 LOP Low Power Oscillator 低功率振荡器
 LocDev Local Device 本地设备
 M Master or Mandatory Master 必须的 (用于帧格式)
 M-ADDR Medium Access Control Address 介质访问控制地址
 MAC Medium Access Control 介质访问控制

MAPI Messaging Application Procedure Interface 消息应用过程接口
ME Management Entity 管理实体
MM Mobility Management 移动管理
MMI Man Machine Interface 人机接口
MSC Message Sequence Chart 信息序列表
MTU Maximum Transmission Unit 最大传输单元
NAK Negative Acknowledge 消极确认
NAP Non-significant Address Part 不重要的地址部分
PCM Pulse Coded Modulation 脉冲编码调制
PCMCIA Personal Computer Memory Card International Association 个人计算机存储器卡国际联合会
PDA Personal Digital Assistant 个人数字助理
PIM Personal Information Management 个人信息管理
PIN Personal Identification Number 个人识别码
PM-ADDR Parked Member Address 休眠的 Bluetooth 单元的地址
PN Pseudo-random Noise 伪随机噪声
PnP Plug and Play 即插即用
POTS Plain Old Telephone System 简易老式电话系统
PPP Point-to-Point Protocol 点到点协议
PPM Part Per Million 百万分之一
PRBS Pseudo Random Bit Sequence 伪随机比特序列
PRNG Pseudo Random Noise Generation 伪随机噪声生成
PSTN Public Switched Telephone Network 公用交换电话网
QoS Quality of Service 服务质量
RAND Random number 随机数
RF Radio Frequency 射频
RFC Request For Comments 草案
RFCOMM Serial cable emulation protocol based on ETSI TS 07.10 基于 ETST TS 07.10 的串行电缆模拟协议
RX Receiver 接收器
SAP Service Access Points 服务接入点
SAR Segmentation and Reassembly sublayer 分组/装配子层
SEQN Sequential Numbering scheme 序列号码机制
SUT System Under Test 在测试系统
SW Software 软件
TAE Terminal Adapter Equipment 终端适配设备
TBD To Be Defined 未定义
TC Test Control 测试控制
TCI Test Control Interface 测试控制接口
UDP User Datagram Protocol 用户数据报协议

TCS Binary Telephony Control Specification 二进制电话控制标准
TX Transmitter 发送端
UA User Asynchronous user data 用户异步数据
UAP Upper Address Part 高位地址部分
UART Universal Asynchronous Receiver Transmitter 通用异步收发器
UC User Control 用户控制
UI User Isochronous user data 用户同步数据
UT Upper Tester 高层测试
UUID Universally Unique Identifier 通用唯一标识