

to the loudness rating, in ITU P.79 and at PGA value of 0 dB. Programmable Gain Amplifiers (PGAs) are used to control the audio level at the terminals by the user. For conversion between various PCM representations: A-law,  $\mu$ -law and linear PCM, ITU-T G.711, G.712, G.714 give guidelines and PCM value relationships. Zero-code suppression based on ITU-T G.711 is also recommended to avoid network mismatches.

### 1.7 FREQUENCY MASK

For interfacing a Bluetooth terminal to a digital cellular mobile terminal, a compliance of the CVSD decoder signal to the frequency mask given in the cellular standard, is recommended to guarantee correct function of the speech coders. A recommendation for a frequency mask is given in Table 1.1. Figure 1.3: shows a plot of the frequency mask for Bluetooth (solid line). The GSM frequency mask (dotted line) is shown in Figure 1.3: for comparison.

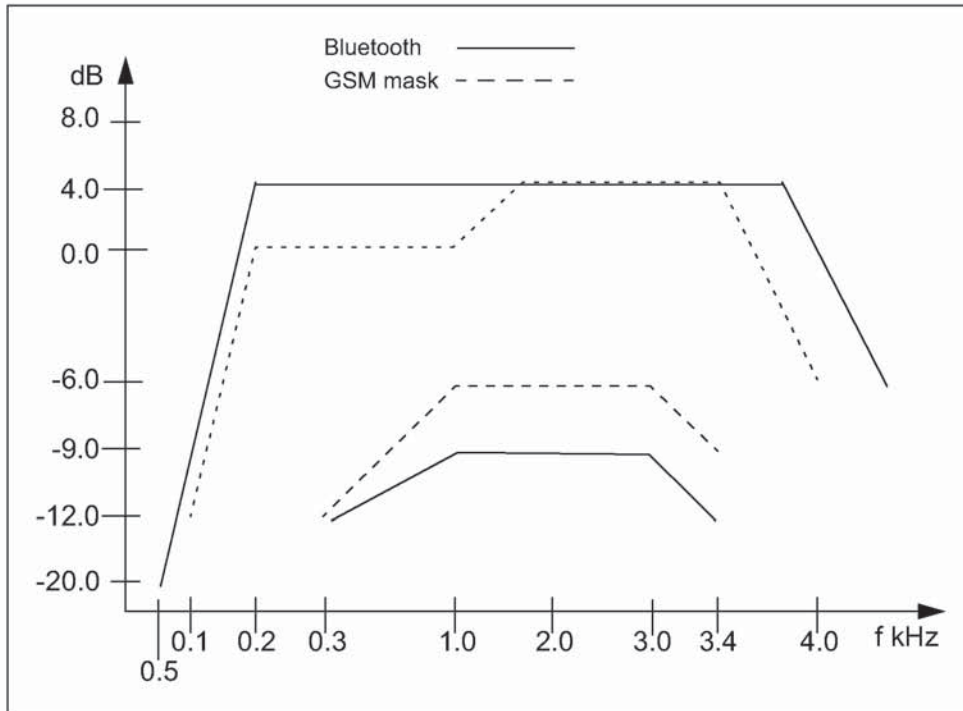


Figure 1.3: Plot of recommended frequency mask for Bluetooth. The GSM send frequency mask is given for comparison (dotted line)

Frequency (Hz)	Upper Limit (dB)	Lower Limit (dB)
50	-20	-
300	4	-12
1000	4	-9
2000	4	-9
3000	4	-9
3400	4	-12
4000	0	-

Table 1.1: Recommended Frequency Mask for Bluetooth

## Appendix VI

# BASEBAND TIMERS

This appendix contains a list of all timers defined the Baseband Specification.



## CONTENTS

---

<b>1</b>	<b>Baseband Timers .....</b>	<b>996</b>
1.1	LIST OF TIMERS .....	996
1.1.1	inquiryTO .....	996
1.1.2	pageTO .....	996
1.1.3	pagerespTO .....	996
1.1.4	inqrespTO .....	996
1.1.5	newconnectionTO .....	996
1.1.6	supervisionTO .....	997

## 1 BASEBAND TIMERS

---

This appendix contains a list of all timers defined in this specification. Definitions and default values of the timers are listed below.

All timer values are given in slots.

### 1.1 LIST OF TIMERS

#### 1.1.1 inquiryTO

The *inquiryTO* defines the number of slots the **inquiry** substate will last. Its value is determined by an HCI command.

#### 1.1.2 pageTO

The *pageTO* defines the number of slots the **page** substate can last before a response is received. Its value is determined by an HCI command.

#### 1.1.3 pagerespTO

In the slave, it defines the number of slots the slave awaits the master's response, FHS packet, after sending the page acknowledgment ID packet. In the master, *pagerespTO* defines the number of slots the master should wait for the FHS packet acknowledgment before returning to **page** substate. Both master and slave units should use the same value for this timeout, to ensure common page/scan intervals after reaching *pagerespTO*.

The *pagerespTO* default value is 8 slots.

#### 1.1.4 inqrespTO

In the inquiry scan substate, when a device triggers on an inquiry, it waits a RAND random number of slots and returns to inquiry scan. The *inqRespTO* defines the number of slots the device will stay in the inquiry scan substate without triggering on an inquiry after the RAND wait period. The timeout value should preferably be in multiples of an inquiry train period. Upon reaching the *inqrespTO*, the device returns to **CONNECTION** or **STANDBY** state.

The *inqrespTO* default value is 128 slots.

#### 1.1.5 newconnectionTO

Every time a new connection is started through paging, scanning, master-slave switch or unpairing, the master sends a POLL packet as the first packet in the new connection. Transmission and acknowledgment of this POLL packet is used to confirm the new connection. If the POLL packet is not received by the

slave or the response packet is not received by the master for *newconnectionTO* number of slots, both the master and the slave will return to the previous substate.

- | *newconnectionTO* default value is 32 slots.

### 1.1.6 supervisionTO

The *supervisionTO* is used by both the master and slave to monitor link loss. If a device does not receive any packets that pass the HEC check and have the proper AM\_ADDR for a period of *supervisionTO*, it will reset the link *supervisionTO* will work through hold and sniff periods.

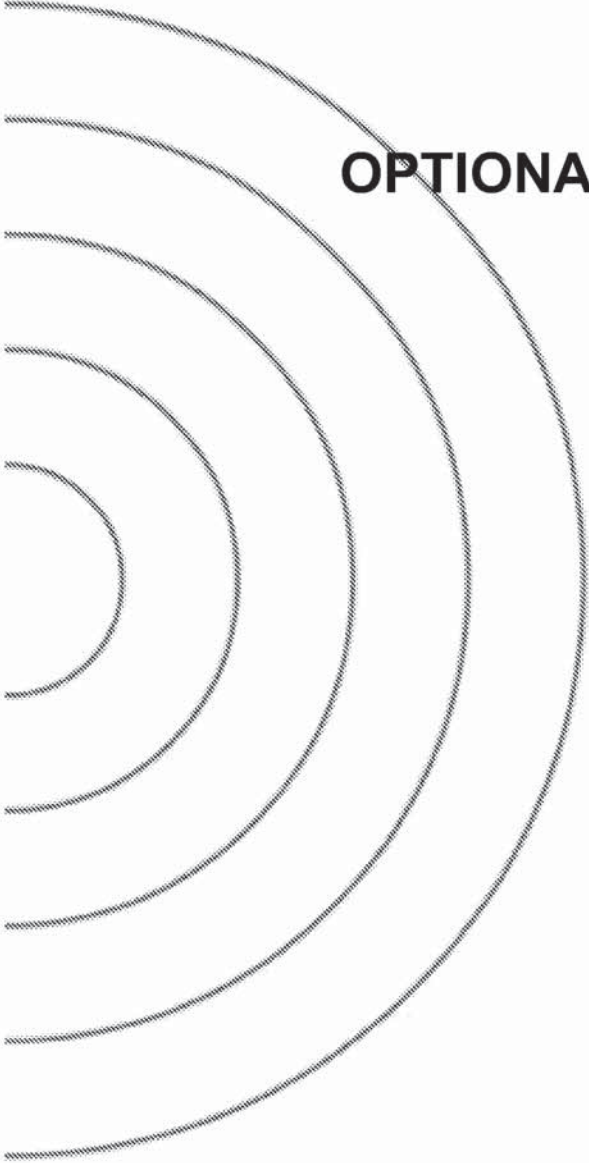
The *supervisionTO* value is determined by an HCI command. At the baseband level a default value that is equivalent to 20 seconds will be used.





**Appendix VII**

**OPTIONAL PAGING SCHEMES**





---

## CONTENTS

---

1	General.....	1003
2	Optional Paging Scheme I.....	1004
2.1	Page.....	1004
2.2	Page Scan .....	1006
2.3	Page Response Procedures .....	1006
2.4	Train Tracing .....	1007



## 1 GENERAL

---

For the access procedure, several paging schemes may be used. There is one mandatory paging scheme which has to be supported by all Bluetooth devices. This scheme has been described in Baseband Specification Section 10.6 on page 99. In addition to the mandatory scheme, a Bluetooth unit may support one or more optional paging schemes. The method used for page scan is indicated in the FHS payload, see Baseband Specification Section 4.4.1.4 on page 56. Three additional optional paging schemes are possible; only optional paging scheme *I* has been defined yet.

## 2 OPTIONAL PAGING SCHEME I

In this section the first optional paging scheme is described which may be used according to the rules specified in Baseband Specification Section 10 on page 95 and LMP Specification Section 3.23 on page 223. The paging code for optional scheme *I* is 1 (0 is used for the mandatory scheme), see also Baseband Specification Section 4.4.1.4 on page 56

The main difference between the first optional paging scheme and the mandatory scheme is the construction of the page train sent by the pager. In addition to transmission in the even master slots, the master is transmitting in the odd master slots as well. This allows the slave unit to reduce the scan window.

### 2.1 PAGE

The same 32 frequencies that are used for transmitting ID-packets in the mandatory paging scheme are used in the optional paging scheme *I* (for the construction of page trains, see Baseband Specification Section 11.3.2 on page 135). The 32 frequencies are also split into an **A-train** and **B train**. In contrast to the mandatory scheme, the same 32 frequencies that are used for transmitting are also used for reception trials, to catch the response from the addressed device.

The construction of the page train in optional page scheme *I* differs from the page train in the mandatory scheme in two ways:

- the page train consists of 10 slots, or 6.25 ms
- the first 8 slots of the train are used to transmit the ID packets, the 9th slot is used to send a marker packet, and the 10th slot is used for the return of a slave response

The marker packets precede the return slot, indicating the position where the slave can respond, and with which frequency. For the marker codes  $M\_ID$ , bit-inverted page access codes are used. If a marker code is received at  $T_m$  with frequency  $f_k$ , a return is expected at nominally  $T_m + 625\mu s$  at frequency  $f_k$ .

**Note:** The bit-inverted code  $M\_ID$  to be used as marker code is beneficial for the implementation of the correlators, because the sign of the correlation peak can be used to identify the mark code during page scanning. Still, the transmitting party is uniquely identified, since inverted ID packets are not identical to the ID packets for the device with bit-wise inverted LAP.

The frequency ordering in the train and the frequencies used for the marker and receive slots change after every train. After 8 trains, all of which have a different appearance, the entire procedure is repeated. It is, therefore, more appropriate to talk about subtrains, each with length 6.25ms. Eight subtrains form a supertrain, which is repeated. An example of a supertrain with the eight subtrains is

illustrated in Figure 2.1. The supertrain length is 50ms. In this example, the **A-train** is assumed with an estimated frequency of  $f_8$ ; as a consequence, the frequencies selected for the train range from  $f_0$  to  $f_{15}$ . The marker codes M\_ID are indicated as **M**; the receive (half) slots are indicated as **R**.

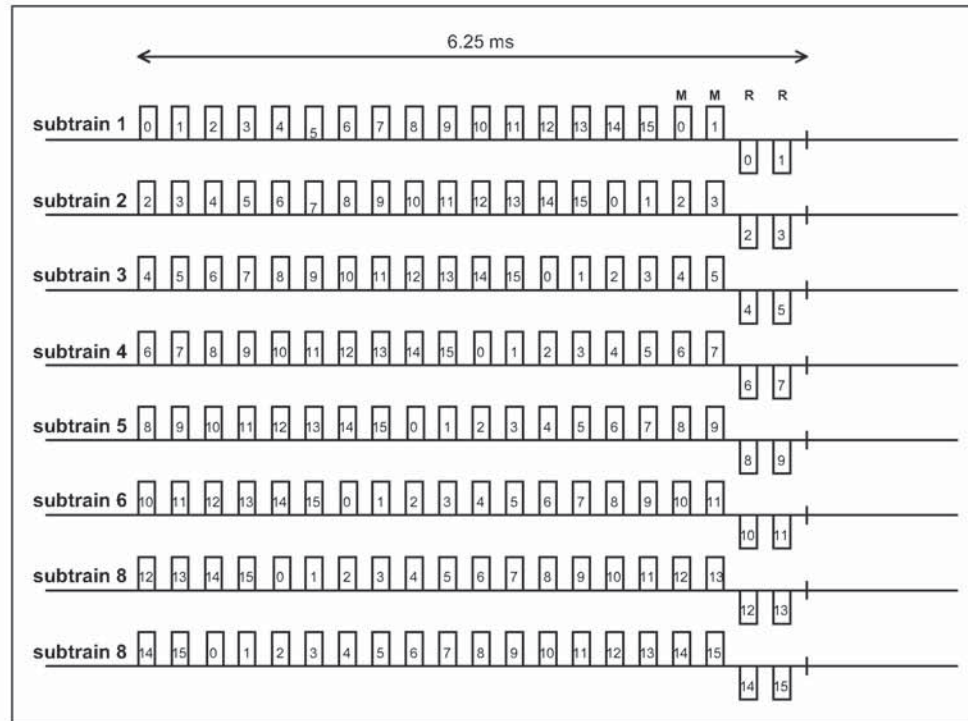


Figure 2.1: Example of train configuration for optional page scheme I.

Corresponding to the paging modes R0, R1 and R2 of the mandatory scheme, the optional scheme supports the same three modes as described for the mandatory scheme in Baseband Specification Section 10.6.2 on page 99

Since the subtrain length is now 10 slots, the 1.28s interval does not cover a multiple of (sub)trains any longer. Therefore, in contrast to the mandatory scheme, the exchange from **A-train** to **B-train** and vice versa is not based on the 1.28s interval, but instead on a multiple number of supertrains. For the R1 and R2 modes, the repetition of a supertrain  $N_{sup}$  is indicated in Table 2.1 below.

mode	No SCO link	One SCO link (HV3)	Two SCO links (HV3)
R1	$N_{sup}=26$	$N_{sup}=52$	$N_{sup}=77$
R2	$N_{sup}=52$	$N_{sup}=103$	$N_{sup}=154$

Table 2.1: Relation between repetition duration of **A-** and **B-**trains and paging modes R1 and R2 when SCO links are present

In accordance with the phase input to the hop selection scheme  $X_p$  in (EQ 4) on page 135 in the Baseband Specification (Section 11.3.2), the phase input  $X_{p\_opt}$  in the optional mode is determined by:

$$X_{p\_opt} = [k_{offset\_opt} + ST(cnt)] \bmod 32 \quad (\text{EQ A1})$$

where  $k_{offset\_opt}$  is determined by the A/B selection and the clock estimation of the recipient:

$$k_{offset\_opt} = \begin{cases} \text{CLKE}_{16-12} + 24 & \text{A-train} \\ \text{CLKE}_{16-12} + 8 & \text{B-train} \end{cases} \quad (\text{EQ A2})$$

and  $ST$  is a function determining the structure of the sub- and supertrain:

$$ST(cnt) = (cnt \bmod 160 - 2 * \text{INT}[(cnt \bmod 160) / 20]) \bmod 16 \quad (\text{EQ A3})$$

$k_{offset\_opt}$  is determined once at the beginning of the repetition period.

The CLKE value as is found at the beginning of the repetition interval is taken (the repetition interval being the interval in which the same supertrain is repeated all the time). As long as no train change takes place,  $k_{offset\_opt}$  is not updated.  $cnt$  is a counter which is reset to zero at the beginning of the repetition interval and is incremented at the half-slot rate (3200 cycles/s)

The first two ID-packets of a train are transmitted in an even numbered slot.

## 2.2 PAGE SCAN

The basic page scanning is identical to the mandatory scheme except that a scan duration of  $9.5 \cdot 0.625 = 5.9375$  ms is sufficient at the slave side.

If a device wants to scan concurrently for the mandatory and optional mode (e.g. after an inquiry response was sent), the device shall try to identify whether the paging party uses the optional scheme after an ID packet was caught. This can be done by train tracing; i.e. the device can determine whether transmission takes place in consecutive slots (optional paging scheme **I**) or in every over slot (mandatory paging scheme), and/or whether mark codes are sent.

## 2.3 PAGE RESPONSE PROCEDURES

The page response procedures at the master and slave sides are almost identical to the procedures described in the mandatory mode (see Baseband Specification Section 10.6.4 on page 104). There are two differences:

- The page response routine starts after the transmission and reception of the marker code  $M\_ID$
- The ID packet sent by recipient is identical to the frequency in which the marker code was received

For the page response timing, see Figure 2.2 and Figure 2.3.



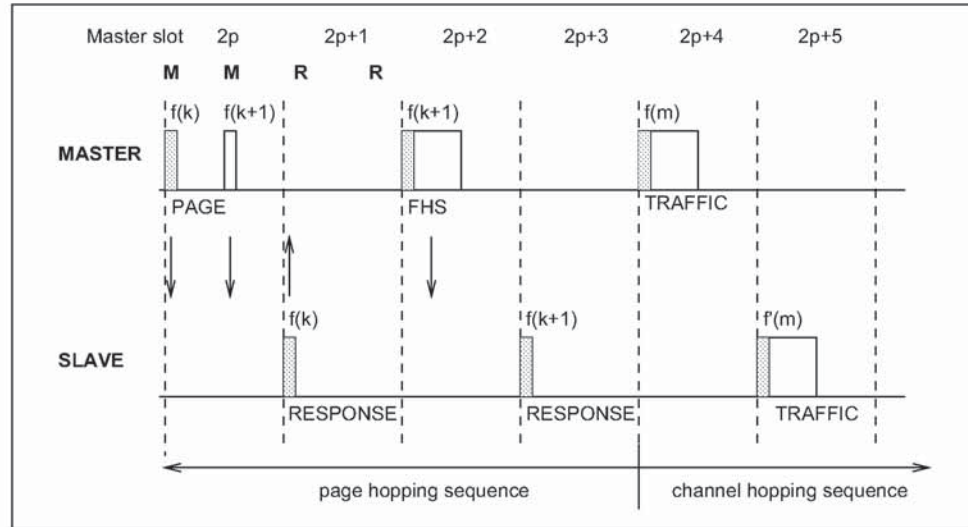


Figure 2.2: Messaging when marker code is received in first half slot of even master slot

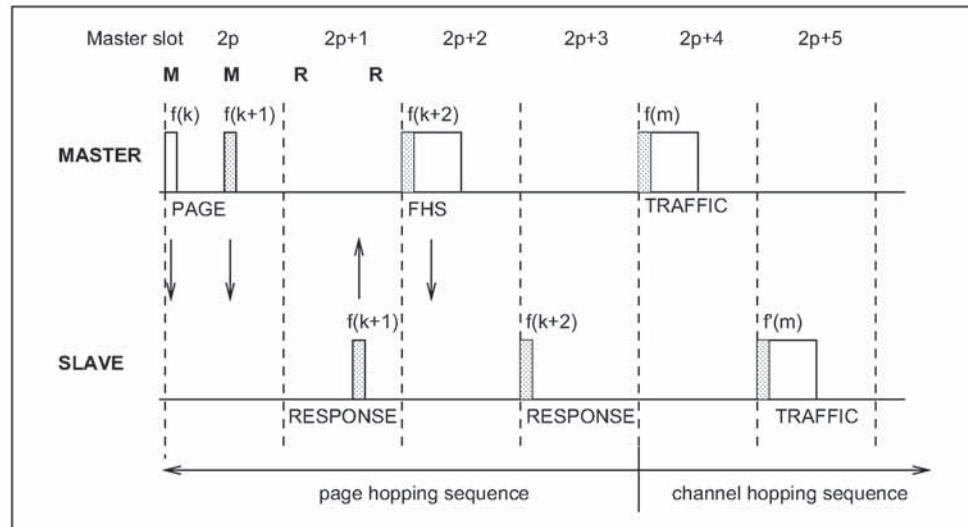


Figure 2.3: Messaging when marker code is received in second half slot of even master slot

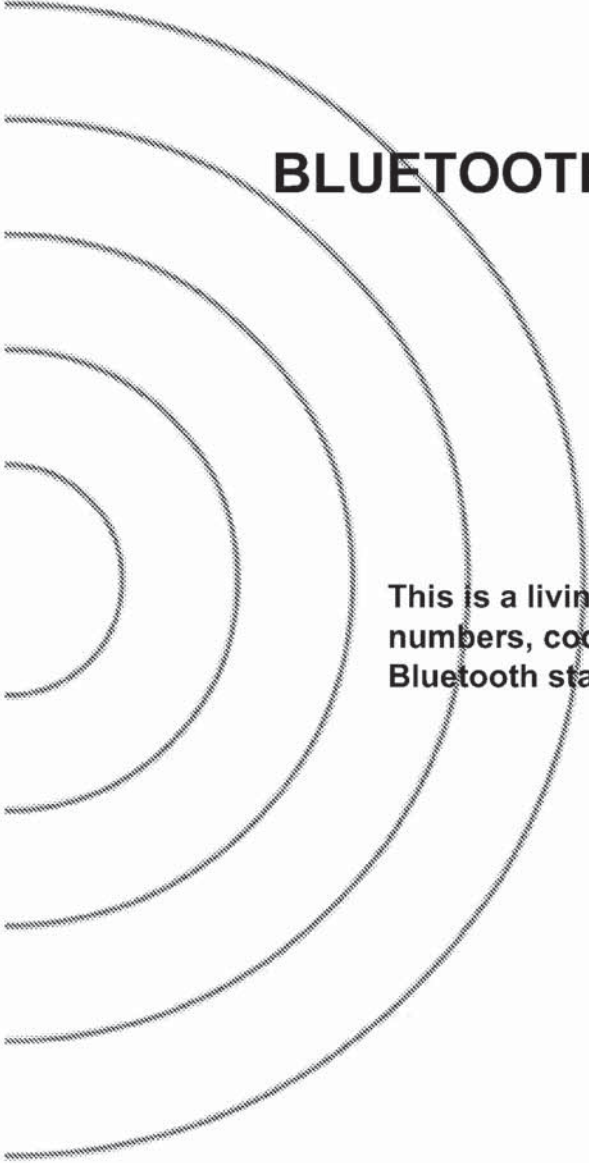
## 2.4 TRAIN TRACING

This section outlines how a slave may search for the mark code although the current partitioning into A- and B-trains at the master side is not known. Train tracing means that the slave tries to receive as many page access codes from the train as possible, to catch a mark code as soon as possible. When searching for the mark codes, or trying to distinguish between the mandatory paging mode and the optional paging mode, a unit shall set up a hopping pattern for train tracing after the reception of the first access code. The hopping pattern

shall ensure that the transmission and reception is performed with a 50% probability on the same frequency regardless of the actual frequency set (16 frequencies) used for paging.

## Appendix VIII

# BLUETOOTH ASSIGNED NUMBERS



This is a living document that lists assigned numbers, codes and identifiers in the Bluetooth standard.



**CONTENTS**

**1 Bluetooth Baseband ..... 1012**

    1.1 The General- and Device-Specific Inquiry Access Codes (DIACs)..... 1012

    1.2 The Class of Device/Service field ..... 1012

        1.2.1 Major Service Classes..... 1013

        1.2.2 Major Device Classes..... 1014

        1.2.3 The Minor Device Class field..... 1014

        1.2.4 Minor Device Class field - Computer Major Class.... 1015

        1.2.5 Minor Device Class field - Phone Major Class ..... 1015

        1.2.6 Minor Device Class field - LAN Access Point Major Class ..... 1016

        1.2.7 Minor Device Class field - Audio Major Class ..... 1017

**2 Link Manager Protocol (LMP)..... 1018**

    2.1 The Link Manger Version parameter..... 1018

    2.2 The LMP\_CompId parameter codes..... 1018

**3 Logical Link Control and Adaptation Protocol (L2CAP)..... 1019**

    3.1 Channel Identifiers ..... 1019

    3.2 Protocol and Service Multiplexor (PSM) ..... 1019

**4 Service Discovery Protocol (SDP)..... 1020**

    4.1 Universally Unique Identifier (UUID) short forms ..... 1020

    4.2 Base Universally Unique Identifier (UUID)..... 1020

    4.3 Protocols ..... 1021

    4.4 Service classes ..... 1022

    4.5 Attribute Identifier codes ..... 1023

    4.6 Protocol Parameters ..... 1024

    4.7 Host Operating Environment Identifiers ..... 1024

        4.7.1 ClientExecutableURL substitution strings ..... 1024

        4.7.2 IconURL substitution strings..... 1027

**5 References ..... 1028**

**6 Terms and Abbreviations ..... 1029**

**7 List of Figures..... 1030**

**8 List of Tables ..... 1031**

# 1 BLUETOOTH BASEBAND

## 1.1 THE GENERAL- AND DEVICE-SPECIFIC INQUIRY ACCESS CODES (DIACS)

The Inquiry Access Code is the first level of filtering when finding Bluetooth devices and services. The main purpose of defining multiple IACs is to limit the number of responses that are received when scanning devices within range.

#	LAP value	Usage
0	0x9E8B33	General/Unlimited Inquiry Access Code (GIAC)
1	0x9E8B00	Limited Dedicated Inquiry Access Code (LIAC)
2-63	0x9E8B01-0x9E8B32, 0x9E8B34-0x9E8B3F	RESERVED FOR FUTURE USE

Table 1.1: The Inquiry Access Codes

The Limited Inquiry Access Code (LIAC) is only intended to be used for limited time periods in scenarios where both sides have been explicitly caused to enter this state, usually by user action. For further explanation of the use of the LIAC, please refer to the Generic Access Profile [7].

In contrast it is allowed to be continuously scanning for the General Inquiry Access Code (GIAC) and respond whenever inquired.

## 1.2 THE CLASS OF DEVICE/SERVICE FIELD

The Class of Device/Service (CoD) field has a variable format. The format is indicated using the 'Format Type field' within the CoD. The length of the Format Type field is variable and ends with two bits different from '11'. The version field starts at the least significant bit of the CoD and may extend upwards.

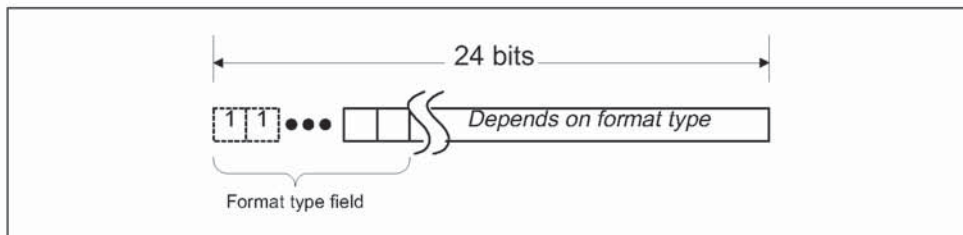


Figure 1.1: General format of Class of Device/Service

In the 'format #1' of the CoD (Format Type field = 00), 11 bits are assigned as a bit-mask (multiple bits can be set) each bit corresponding to a high level generic category of service class. Currently 7 categories are defined. These

are primarily of a 'public service' nature. The remaining 11 bits are used to indicate device type category and other device-specific characteristics.

Any reserved but otherwise unassigned bits, such as in the Major Service Class field, should be set to 0.

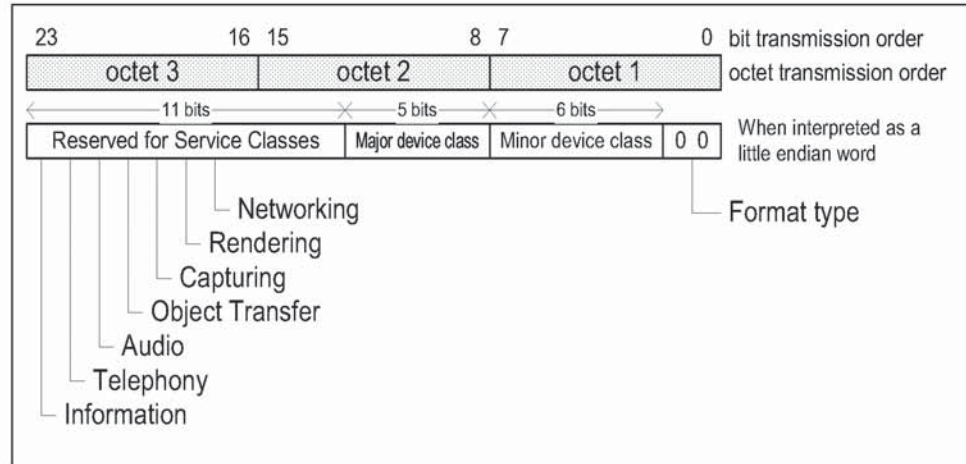


Figure 1.2: The Class of Device/Service field (format type 1). Note the order in which the octets are sent on the air and stored in memory.

**1.2.1 Major Service Classes**

Bit no	Major Service Class
13	Limited Discoverable Mode <sup>1</sup>
14	(reserved)
15	(reserved)
16	(reserved)
17	Networking (LAN, Adhoc, ...)
18	Rendering (Printing, Speaker, ...)
19	Capturing (Scanner, Microphone, ...)
20	Object Transfer (v-Inbox, v-Folder, ...)
21	Audio (Speaker, Microphone, Headset service, ...)
22	Telephony (Cordless telephony, Modem, Headset service, ...)
23	Information (WEB-server, WAP-server, ...)

Table 1.2: Major Service Classes

1. As defined in [7]

### 1.2.2 Major Device Classes

The Major Class segment is the highest level of granularity for defining a Bluetooth Device. The main function of a device is used to determine the major class grouping. There are 32 different possible major classes. The assignment of this Major Class field is defined in Table 1.3.

Code (bits)					Major Device Class
12	11	10	9	8	bit no of CoD
0	0	0	0	0	Miscellaneous <sup>1</sup>
0	0	0	0	1	Computer (desktop, notebook, PDA, organizers, ...)
0	0	0	1	0	Phone (cellular, cordless, payphone, modem, ...)
0	0	0	1	1	LAN Access Point
0	0	1	0	0	Audio (headset, speaker, stereo, ...)
0	0	1	0	1	Peripheral (mouse, joystick, keyboards, ...)
x	x	x	x	x	Range 0x06 to 0x1E reserved
1	1	1	1	1	Unclassified, specific device code not assigned

Table 1.3: Major Device Classes

1. Used where a more specific Major Device Class code is not suited (but only as specified in this document. Devices that do not have a major class code assigned can use the all-1 code until 'classified')

### 1.2.3 The Minor Device Class field

The 'Minor Device Class field' (bits 7 to 1 in the CoD), are to be interpreted only in the context of the Major Device Class (but independent of the Service Class field). Thus the meaning of the bits may change, depending on the value of the 'Major Device Class field'. When the Minor Device Class field indicates a device class, then the primary device class should be reported, e.g. a cellular phone that can also work as a cordless handset should use 'Cellular' in the minor device class field.



**1.2.4 Minor Device Class field - Computer Major Class**

Code (bits)						Minor Device Class
7	6	5	4	3	2	bit no of CoD
0	0	0	0	0	0	Unclassified, code for device not assigned
0	0	0	0	0	1	Desktop workstation
0	0	0	0	1	0	Server-class computer
0	0	0	0	1	1	Laptop
0	0	0	1	0	0	Handheld PC/PDA (clam shell)
0	0	0	1	0	1	Palm sized PC/PDA
x	x	x	x	x	x	Range 0x06-0x7F reserved

Table 1.4: Sub Device Class field for the 'Computer' Major Class

**1.2.5 Minor Device Class field - Phone Major Class**

Code (bits)						Minor Device Class
7	6	5	4	3	2	bit no of CoD
0	0	0	0	0	0	Unclassified, code not assigned
0	0	0	0	0	1	Cellular
0	0	0	0	1	0	Cordless
0	0	0	0	1	1	Smart phone
0	0	0	1	0	0	Wired modem or voice gateway
x	x	x	x	x	x	Range 0x05-0x7F reserved

Table 1.5: Sub Device Classes for the 'Phone' Major Class

**1.2.6 Minor Device Class field - LAN Access Point Major Class**

Code (bits)			Minor Device Class
7	6	5	bit no of CoD
0	0	0	Fully available
0	0	1	1-17% utilized
0	1	0	17 - 33% utilized
0	1	1	33 - 50% utilized
1	0	0	50 - 67% utilized
1	0	1	67 - 83% utilized
1	1	0	83 - 99% utilized
1	1	1	No Service Available <sup>1</sup>

Table 1.6: The LAN Access Point Load Factor field

1. "Device is fully utilized and cannot accept additional connections at this time, please retry later"

The exact loading formula is not standardized. It is up to each LAN Access Point implementation to determine what internal conditions to report as a utilization percentage. The only requirement is that the number reflects an ever-increasing utilization of communication resources within the box. As a recommendation, a client that locates multiple LAN Access Points should attempt to connect to the one reporting the lowest load.

Code (bits)			Minor Device Class
4	3	2	bit no of CoD
0	0	0	Unclassified (use this value if no other apply)
x	x	x	range 0x01-0x0F reserved

Table 1.7: Reserved sub-field for the LAN Access Point



**1.2.7 Minor Device Class field - Audio Major Class**

Code (bits)						Minor Device Class
7	6	5	4	3	2	bit no of CoD
0	0	0	0	0	0	Unclassified, code not assigned
0	0	0	0	0	1	Device conforms to the Headset profile [9]
x	x	x	x	x	x	Range 0x02-0x7F reserved

Table 1.8: Sub Device Classes for the 'Audio' Major Class

## 2 LINK MANAGER PROTOCOL (LMP)

### 2.1 THE LINK MANGER VERSION PARAMETER

Parameter name	Assigned values	
VersNr	0	Bluetooth LMP 1.0, [2]
	1-255	(reserved)

Table 2.1: The LMP Version Parameter Values

### 2.2 THE LMP\_COMPID PARAMETER CODES

This is the parameter used in the LMP Version procedure.

Code	Company
0	Ericsson Mobile Communications
1	Nokia Mobile Phones
2	Intel Corp.
3	IBM Corp.
4	Toshiba Corp.
5 - 65534	(reserved)
65535	Unassigned. For use in internal and interoperability tests before a Company ID has been assigned. May not be used in products.

Table 2.2: The LMP\_CompId parameter codes

### 3 LOGICAL LINK CONTROL AND ADAPTATION PROTOCOL (L2CAP)

Please see Section 4.3 for assigned PSM values.

#### 3.1 CHANNEL IDENTIFIERS

Destination CID	Protocol/usage	Reference
0x0000	Illegal, should not be used	[3]
0x0001	L2CAP signalling channel	[3]
0x0002	L2CA connection less data	[3]
0x0003 - 0x003F	(reserved)	

Table 3.1: Pre-defined L2CAP Channel Identifiers

#### 3.2 PROTOCOL AND SERVICE MULTIPLEXOR (PSM)

Protocol	PSM	Reference
SDP	0x0001	[4]
RFCOMM	0x0003	[5]
TCS-BIN	0x0005	[6]
TCS-BIN-CORDLESS	0x0007	[6]

Table 3.2: Assigned Protocol and Service Multiplexor values (PSM)

## 4 SERVICE DISCOVERY PROTOCOL (SDP)

### 4.1 UNIVERSALLY UNIQUE IDENTIFIER (UUID) SHORT FORMS

The Bluetooth Service Discovery Protocol (SDP) specification defines a way to represent a range of UUIDs (which are nominally 128-bits) in a shorter form. A *reserved* range of  $2^{32}$  values can be represented using 32-bits (denoted uuid32). Of these, a sub-range of  $2^{16}$  values can be represented using only 16-bits (denoted uuid16). Any value in the  $2^{32}$  range that is not assigned in this document is reserved pending future revisions of this document. In other words, no value in this range may be used except as specified in this or future revisions of this document. UUID values outside of this range can be allocated as described in [19] for any purpose the allocator desires.

### 4.2 BASE UNIVERSALLY UNIQUE IDENTIFIER (UUID)

The Base UUID is used for calculating 128-bit UUIDs from 'short UUIDs' (uuid16 and uuid32) as described in the SDP Specification [4].

Mnemonic	UUID
BASE_UUID	00000000-0000-1000-8000-00805F9B34FB

**4.3 PROTOCOLS**

Mnemonic	UUID	Name	Ref.
SDP	uuid16: 0x0001 <sup>1</sup>	sdp.bt	[4]
RFCOMM	uuid16: 0x0003	com.bt	[5]
TCS-BIN	uuid16: 0x0005	tcs.bt	[6]
L2CAP	uuid16: 0x0100		[3]
IP	uuid16: 0x0009		
UDP	uuid16: 0x0002		
TCP	uuid16: 0x0004		
TCS-AT	uuid16: 0x0006	modem	
OBEX	uuid16: 0x0008	obex	
FTP	uuid16: 0x000A	ftp	
HTTP	uuid16: 0x000C	http	
WSP	uuid16: 0x000E	wsp	

Table 4.1: Protocol Universally Unique Identifiers and Names

1. 'Short UUID'

### 4.4 SERVICE CLASSES

Mnemonic	UUID	Profile <sup>1</sup>	AbstractName
ServiceDiscoveryServerServiceClassID	uuid16: 0x1000		
BrowseGroupDescriptorServiceClassID	uuid16: 0x1001		
PublicBrowseGroup	uuid16: 0x1002		
SerialPort	uuid16: 0x1101	[7]	serial.bt
LANAccessUsingPPP	uuid16: 0x1102		
DialupNetworking	uuid16: 0x1103	[13]	
IrMCSync	uuid16: 0x1104	[17]	
OBEXObjectPush	uuid16: 0x1105	[16]	
OBEXFileTransfer	uuid16: 0x1106	[15]	
IrMCSyncCommand	uuid16: 0x1107	[17]	
Headset	uuid16: 0x1108	[7]	headset
CordlessTelephony	uuid16: 0x1109	[10]	
Intercom	uuid16: 0x1110	[11]	
Fax	uuid16: 0x1111	[12]	
HeadsetAudioGateway	uuid16: 0x1112	[7]	
PnPInformation	uuid16: 0x1200		
GenericNetworking	uuid16: 0x1201	n/a	
GenericFileTransfer	uuid16: 0x1202	n/a	
GenericAudio	uuid16: 0x1203	n/a	
GenericTelephony	uuid16: 0x1204	n/a	

Table 4.2: Service Class Identifiers and Names

1. If the specified Service Class directly and exactly implies a certain Profile, the Profile is indicated here (i.e. for concrete Service Classes). Leave empty for abstract Service Classes.

The Profile column in Table 4.2 indicates which Service Class identifiers that also directly corresponds to a Bluetooth Profile. It is not allowed to use the Service Class UUID unless the service complies with the specified Profile. These UUIDs might also appear as Profile Identifiers in the BluetoothProfileDescriptorList attribute.



### 4.5 ATTRIBUTE IDENTIFIER CODES

Mnemonic	Attribute ID	Reference
ServiceRecordHandle	0x0000	[4] <i>Bluetooth Service Discovery Protocol (SDP)</i> , Bluetooth SIG
ServiceClassIDList	0x0001	
ServiceRecordState	0x0002	
ServiceID	0x0003	
ProtocolDescriptorList	0x0004	
BrowseGroupList	0x0005	
LanguageBaseAttributeIDList	0x0006	
ServiceInfoTimeToLive	0x0007	
ServiceAvailability	0x0008	
BluetoothProfileDescriptorList	0x0009	
DocumentationURL	0x000A	
ClientExecutableURL	0x000B	
Icon10	0x000C	
IconURL	0x000D	
Reserved	0x000E-0x01FF	
ServiceName	0x0000 + b <sup>1</sup>	
ServiceDescription	0x0001 + b	
ProviderName	0x0002 + b	
VersionNumberList	0x0200	
ServiceDatabaseState	0x0201	
GroupID	0x0200	
Remote audio volume control	0x0302 <sup>2</sup>	[7]
External network	0x0301	[10]
Service Version	0x0300	
Supported Data Stores List	0x0301	[17]
Supported Formats List	0x0303	[16]

Table 4.3: Attribute Identifiers

Mnemonic	Attribute ID	Reference
Fax Class 1 Support	0x0302	[12]
Fax Class 2.0 Support	0x0303	
Fax Class 2 Support	0x0304	
Audio Feedback Support	0x0305	

Table 4.3: Attribute Identifiers

- 'b' in this table represents a base offset as given by the LanguageBaseAttributeIDList attribute. For the primary language, 'b' must be equal to 0x0100 as described in the SDP specification.
- Items in *italic* are tentative values in this version of the document.

## 4.6 PROTOCOL PARAMETERS

Protocol	Parameter mnemonic	Index
L2CAP	PSM	1
TCP or UDP	Port	1
RFCOMM	Channel	1

Table 4.4: Protocol Parameters

## 4.7 HOST OPERATING ENVIRONMENT IDENTIFIERS

### 4.7.1 ClientExecutableURL substitution strings

The operating environment identifier strings have the following format<sup>1</sup>:

```
<cpu_type>-<manufacturer>-[<kernel>-]<os>[<version>][-<object_format>]
```

The general rule is that is that a new identifier should only be defined as required to differentiate incompatible operating environments concerning an executable file image. That is, for example different <version>-tags should not be used for compatible versions of the same operating system.

1. It is based on a format used by the GNU AutoConfig tools

Currently defined tags:

CPU-Type ID	Description
alpha	Digital Alpha* compatible
arm	ARM* core or compatible
i86	Any Intel* 80x86-family compatible CPU
i960	Intel* i960 compatible
jvm	Java Virtual Machine*
mips	MIPS MIPS* compatible
ppc	IBM/Motorola PowerPC* compatible
sh3	Hitachi SH-3* compatible
sh4	Hitachi SH-4* compatible
sparc	Sun Sparc* compatible
Kernel ID	Description
chorus, linux, javaos, os9, qnx, vxworks	
<os>	An 'OS identifier' as listed below, might appear in the <kernel> field when the requested OS platform is Java based.
OS+Version-Identifiers	
amigaos, beos4.5, ejava, epocc, epoce, epocq, epocs, gnu, jre1.1, jre1.2, macos, macosx, os2, os9, palms, pjava, pjava1.1, photon, plan9, qnx, rtjava, win95, win98, win2000, wince, winnt4	
Object Format Identifiers <sup>1</sup>	
aout, bout, coff, elf, jar	
Manufacturer Identifiers	
amiga*, apple*, be*, ericsson*, ibm*, intel*, lucent*, microsoft*, microware*, motorola*, nokia*, palm*, psion*, qnx*, sun*, symbian*, toshiba*, unknown <sup>2</sup>	

1. Only applicable when the object format is not otherwise uniquely implied by the identifier string.
2. Use when no other applies.

*Bluetooth Assigned Numbers***Bluetooth.**

For Linux, the 'manufacturer' field may be used to indicate Linux distribution if so required (in which case <version> indicates the version of the distribution). Otherwise use 'unknown'.

Linux Distribution Identifiers
caldera, debian, dlx, doslinux, linuxpro, linuxware, mandrake, mklinux, redhat, slackware, stampede, suse, turbolinux, yggdrasil

Example Operating Environment Identifier Strings		
i86-microsoft-win95	ppc-apple-macos	i86-redhat-linux-gnu6
i86-microsoft-win98	m68k-apple-macos	ppc-mklinux-linux-gnu
i86-microsoft-winnt4	ppc-apple-macosx	
alpha-microsoft-winnt4	i86-apple-macosx	
i86-microsoft-win2000	m68k-amiga-amigaos	
alpha-microsoft-win2000	ppc-amiga-amigaos	
i86-be-beos4.5	jvm-sun-jre1.2	
ppc-be-beos4.5	jvm-sun-pjava1.1	
arm-symbian-epoc3	jvm-sun-ejava	
i86-unknown-linux-gnu	m68k-palm-palmos-coff	
sh3-microsoft-wince	ppc-ibm-vxworks-pjava1.2	
arm-microsoft-wince	sparc-sun-javaos-jre1.2	

**4.7.2 IconURL substitution strings**

The IconURL operating environment identifier strings have the following general format:

```
<horizontal_pixels>x<vertical_pixels>x<color_depth>[m].<file_format>
```

The optional tag 'm' indicates monochrome or grayscale. The host is free to try to match/request any graphics file format as indicated by a <file\_format> tag, however at a minimum files conforming to the Portable Network Graphic standard [18] should be made available at the resulting URL (indicated by <file\_format>=png)<sup>2</sup>.

File format tag	Description
png	Portable Network Graphics [18]
gif	Graphics Interchange File format
bmp	Windows bitmap

Currently defined IconURL Icon format identifier strings:

Example Icon format Identifier Strings	
32x32x8.png	256 color 32 by 32 icon (or 255 colors + transparent)
16x16x8.png	
16x16x1m.png	Black and white (or monochrome + transparent)
10x10x2m.png	4 gray-scales

2. The use of PNG, and whether a subset of PNG should be required, is currently pending further investigation.

## 5 REFERENCES

---

- [1] *Bluetooth Baseband Specification*, Bluetooth SIG
- [2] *Bluetooth Link Manager Specification*, Bluetooth SIG
- [3] *Logical Link Control and Adaptation Protocol Specification*, Bluetooth SIG
- [4] *Bluetooth Service Discovery Protocol (SDP)*, Bluetooth SIG
- [5] *RFCOMM with TS 07.10*, Bluetooth SIG
- [6] *Bluetooth Telephony Control Specification / TCS Binary*, Bluetooth SIG
- [7] *Generic Access Profile*, Bluetooth SIG
- [8] *Serial Port Profile*, Bluetooth SIG
- [9] *Headset Profile*, Bluetooth SIG
- [10] *Cordless Telephony Profile*, Bluetooth SIG
- [11] *Intercom Profile*, Bluetooth SIG
- [12] *Fax Profile*, Bluetooth SIG
- [13] *Dial-up Networking Profile*, Bluetooth SIG
- [14] *IrDA Interoperability*, Bluetooth SIG
- [15] *File Transfer Profile*, Bluetooth SIG
- [16] *Object Push Profile*, Bluetooth SIG
- [17] *Synchronization Profile*, Bluetooth SIG
- [18] *Portable Network Graphics (PNG)*, <http://www.w3.org/Graphics/PNG>
- [19] *UUIDs and GUIDs*, P. J. Leach et al, <http://www.ietf.org/internet-drafts/draft-leach-uuids-guids-01.txt>

## 6 TERMS AND ABBREVIATIONS

LMP	Link Management Protocol
L2CA	Logical Link Control and Adaptation, protocol multiplexer layer for Bluetooth
MTU	Maximum Transmission Unit
SAP	Service Access Points
Baseband	Baseband Protocol
Service Discovery	The ability to discover the capability of connecting devices or hosts.
PnP	Plug and Play
SAR	Segmentation and Reassembly
IP	Internet Protocol
IrDA	InfraRed Data Association
PPP	Point-to-Point Protocol
IETF	Internet Engineering Task Force
RFC	Request For Comments

---

## 7 LIST OF FIGURES

---

Figure 1.1: General format of Class of Device/Service .....	1012
Figure 1.2: The Class of Device/Service field (format type 1).....	1013



**8 LIST OF TABLES**

Table 1.1:	The Inquiry Access Codes .....	1012
Table 1.2:	Major Service Classes .....	1013
Table 1.3:	Major Device Classes .....	1014
Table 1.4:	Sub Device Class field for the 'Computer' Major Class .....	1015
Table 1.5:	Sub Device Classes for the 'Phone' Major Class.....	1015
Table 1.6:	The LAN Access Point Load Factor field .....	1016
Table 1.7:	Reserved sub-field for the LAN Access Point .....	1016
Table 1.8:	Sub Device Classes for the 'Audio' Major Class .....	1017
Table 2.1:	The LMP Version Parameter Values .....	1018
Table 2.2:	The LMP_CompId parameter codes .....	1018
Table 3.1:	Pre-defined L2CAP Channel Identifiers .....	1019
Table 3.2:	Assigned Protocol and Service Multiplexor values (PSM).....	1019
Table 4.1:	Protocol Universally Unique Identifiers and Names .....	1021
Table 4.2:	Service Class Identifiers and Names .....	1022
Table 4.3:	Attribute Identifiers .....	1023
Table 4.4:	Protocol Parameters .....	1024



## Appendix IX

# MESSAGE SEQUENCE CHARTS

Between Host and Host Controller/Link Manager

This document shows examples of interworking between HCI Commands and LM Protocol Data Units in form of message sequence charts. It helps to understand and to correctly use the HCI Commands.



**CONTENTS**

<b>1</b>	<b>Introduction .....</b>	<b>1037</b>
<b>2</b>	<b>Services without connection request.....</b>	<b>1038</b>
2.1	Remote Name Request.....	1038
2.2	One-Time Inquiry .....	1039
2.3	Periodic Inquiry .....	1040
<b>3</b>	<b>ACL connection establishment and detachment.....</b>	<b>1042</b>
3.1	ACL Connection Request phase.....	1043
3.2	ACL Connection Setup phase.....	1045
3.2.1	Pairing .....	1045
3.2.2	Authentication.....	1047
3.3	Encryption and Connection Setup Complete .....	1047
3.4	ACL Disconnection.....	1048
<b>4</b>	<b>Optional activities after ACL Connection establishment .....</b>	<b>1050</b>
4.1	Authentication Requested .....	1050
4.2	Set Connection Encryption.....	1051
4.3	Change Connection Link Key.....	1052
4.4	Master Link Key .....	1053
4.5	Read Remote Supported Features .....	1055
4.6	Read Clock Offset.....	1055
4.7	Read Remote Version Information.....	1056
4.8	QoS Setup .....	1057
4.9	Switch Role .....	1057
<b>5</b>	<b>SCO Connection establishment and detachment.....</b>	<b>1059</b>
5.1	SCO Connection setup .....	1059
5.1.1	Master activates the SCO Connection setup .....	1059
5.1.2	Slave activates the SCO Connection setup .....	1060
5.2	SCO Disconnection.....	1060
<b>6</b>	<b>Special modes: sniff, hold, park .....</b>	<b>1062</b>
6.1	Sniff Mode .....	1062
6.2	Hold Mode.....	1063
6.3	Park Mode.....	1065
6.3.1	Enter park mode.....	1065
6.3.2	Exit Park Mode .....	1066
<b>7</b>	<b>Buffer management, flow control .....</b>	<b>1068</b>
<b>8</b>	<b>Loopback Mode.....</b>	<b>1070</b>
8.1	Local Loopback Mode .....	1070
8.2	Remote Loopback Mode .....	1072

*Message Sequence Charts*

**Bluetooth.**

9	List of Acronyms and Abbreviations .....	1073
10	List of Figures .....	1074
11	List of Tables .....	1075
12	References.....	1076

## 1 INTRODUCTION

---

The goal of this document is to show the interworkings of HCI-Commands and LM-PDUs. It focuses on the message sequence charts for the procedures specified in [3] "Bluetooth Host Controller Interface Functional Specification" with regard to LM Procedures from [2] "Link Manager Protocol".

We illustrate here the most useful scenarios, but we do not cover all possible alternatives. Furthermore, the message sequence charts do not consider the transfer error over Air Interface or Host Interface. In all message sequence charts it is assumed that all events are not masked, so the Host Controller will not filter out any events.

Notation used in the message sequence charts:

**Box:**

- Replaces a group of transactions
- Indicates the start of a procedure or a sub-scenario

Note: in a message sequence chart where several sub-scenarios exist, the sub-scenarios can be executed optionally, consequently, exclusively or independently from each other.

**Hexagon:**

- Indicates a condition that is needed to start the transaction below this hexagon

**Arrow:**

- Represents a message, signal or transaction

**Comment:**

- `/* ... */` indicates editor comments

## 2 SERVICES WITHOUT CONNECTION REQUEST

---

### 2.1 REMOTE NAME REQUEST

The service Remote Name Request is used to find out the name of the remote BT Device without an explicit ACL Connection request.

Sending an HCI\_Remote\_Name\_Request (BD\_ADDR, Page\_Scan\_Repetition\_Mode, Page\_Scan\_Mode, Clock\_Offset), the Host expects that its local BT Device will automatically try to connect to the remote BT Device (with the specified BD\_ADDR). Then the local BT Device should try to get the name, to disconnect, and finally to return the name of the remote BT Device back to the Host (see Figure 2.1 Remote Name Request: sub-scenario 1).

Note: if an ACL Connection already exists (see Figure 2.1 Remote Name Request: sub-scenario 2), the Remote Name Request procedure will be executed like an optional service. No Paging and no ACL Detachment need to be done.



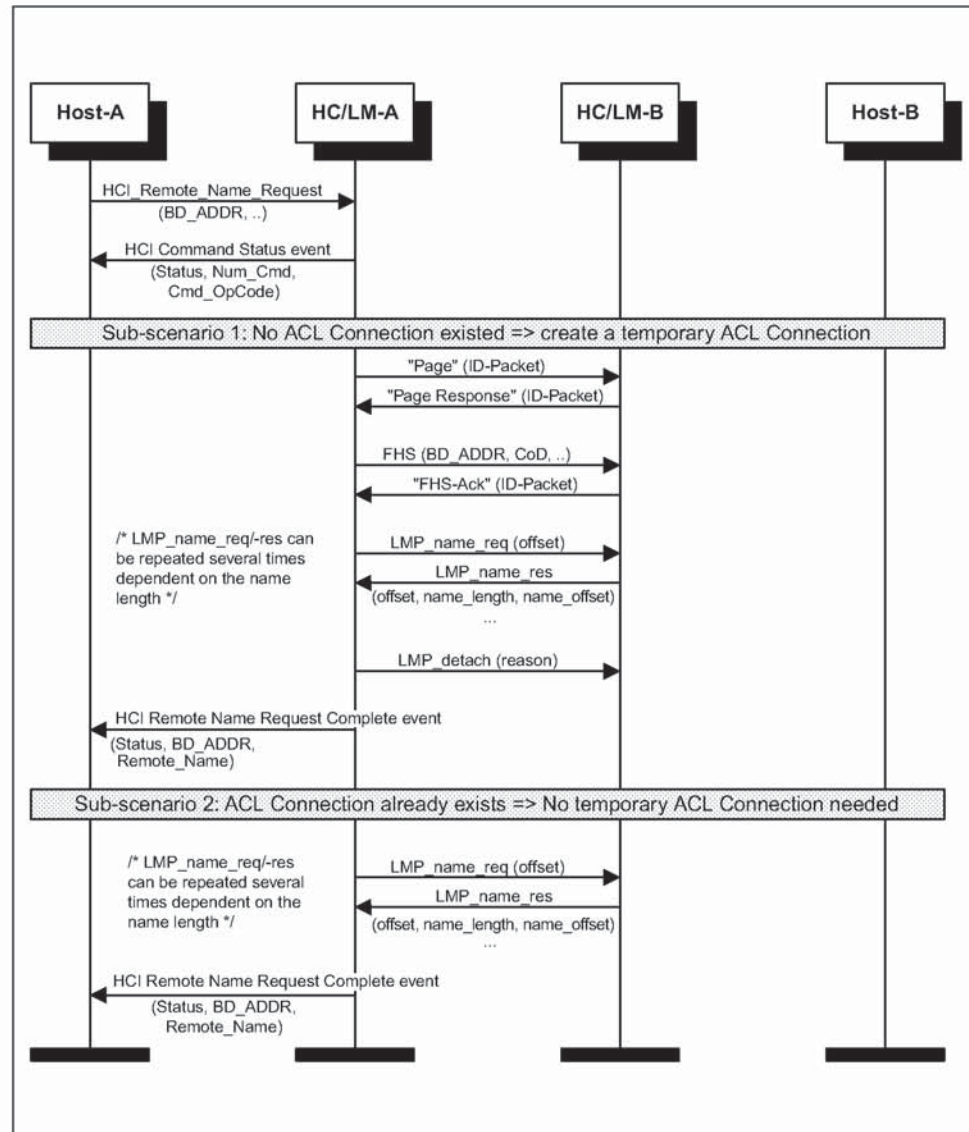


Figure 2.1: Remote Name Request

## 2.2 ONE-TIME INQUIRY

Inquiry is used to detect and collect nearby BT Devices. When receiving the command HCI\_Inquiry (LAP, Inquiry\_Length, Num\_Responses), HC will start the baseband inquiry procedure with an Inquiry Access Code (derived from the specified LAP) and Inquiry Length. When Inquiry Responses are received, HC will filter out and then return the information related to the found BT Devices using one or several Inquiry Result events (Num\_Responses, BD\_ADDR[i], Page\_Scan\_Repetition\_Mode[i], Page\_Scan\_Period\_Mode[i], Page\_Scan\_Mode[i], Class\_Of\_Device[i], Clock\_Offset[i]) to the Host.

The filtering of found BT Devices is specified in HCI\_Set\_Event\_Filter (Filter\_Type, Filter\_Condition\_Type, Condition) with the Filter\_Type = Inquiry Result. When the Inquiry procedure is completed, Inquiry Complete event (Status, Num\_Responses) must be returned to the Host. Otherwise, the command HCI\_Inquiry\_Cancel() will be used to directly stop the inquiry procedure.

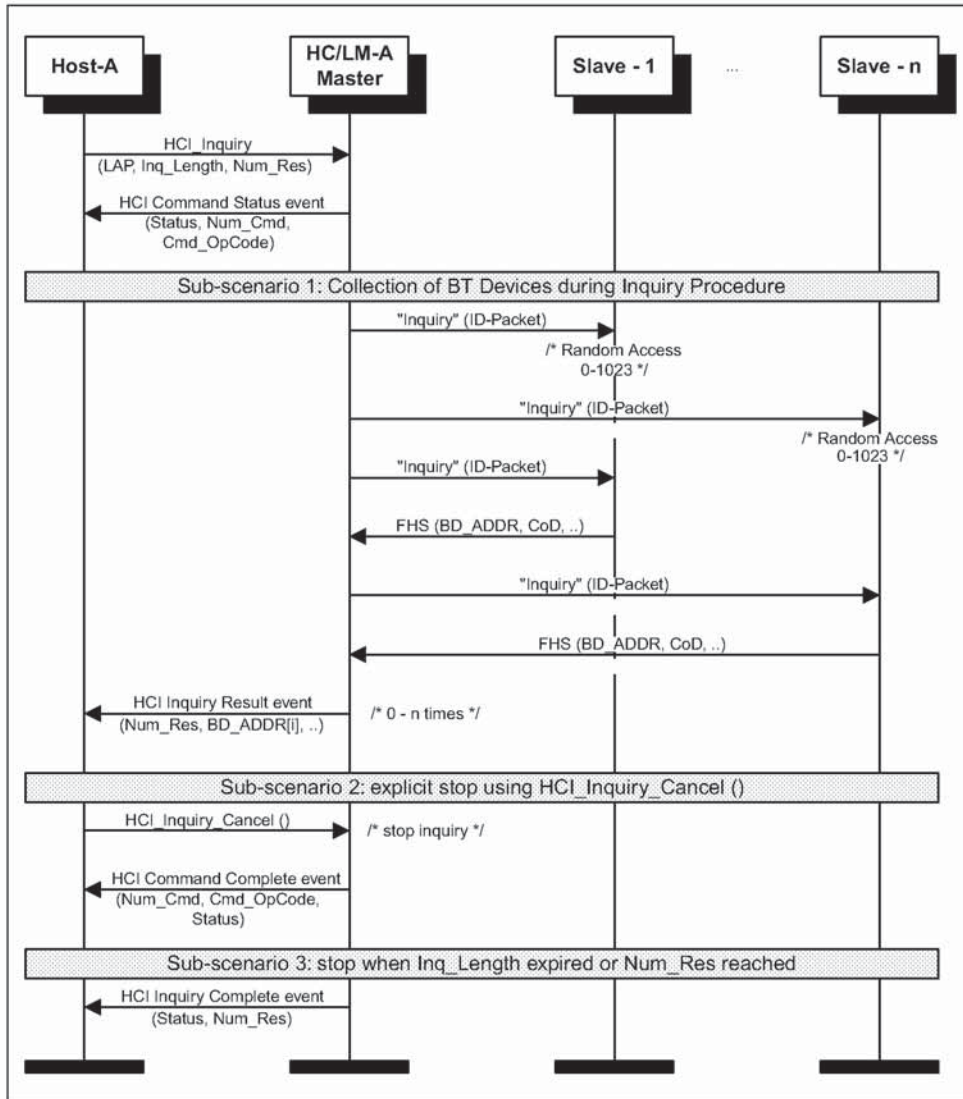


Figure 2.2: One-Time Inquiry

### 2.3 PERIODIC INQUIRY

Periodic inquiry is needed when the inquiry procedure is to be repeated periodically. Receipt of the command HCI\_Periodic\_Inquiry\_Mode (Max\_Period\_Length, Min\_Period\_Length, LAP, Inquiry\_Length, Num\_Responses) HC will start the periodic Inquiry Mode with the specified

parameters Max\_Period\_Length, Min\_Period\_Length, Inquiry\_Access\_code (derived from LAP) and Inquiry\_Length. As in the one-time Inquiry procedure, only BT Devices that are specified in the HCI\_Set\_Event\_Filter (Filter\_Type, Filter\_Condition\_Type, Condition) with the Filter\_Type = Inquiry Result will not be filtered out. Therefore, in the inquiry cycle, one or several Inquiry Result events (Num\_Responses, BD\_ADDR[i], Page\_Scan\_Repetition\_Mode[i], Page\_Scan\_Period\_Mode[i], Page\_Scan\_Mode[i], Class\_Of\_Device[i], Clock\_Offset[i]) and Inquiry Complete event (Status, Num\_Responses) will be returned to the Host with one, or a list of, found BT Devices. The periodic Inquiry can be stopped using HCI\_Exit\_Periodic\_Inquiry\_Mode( ).

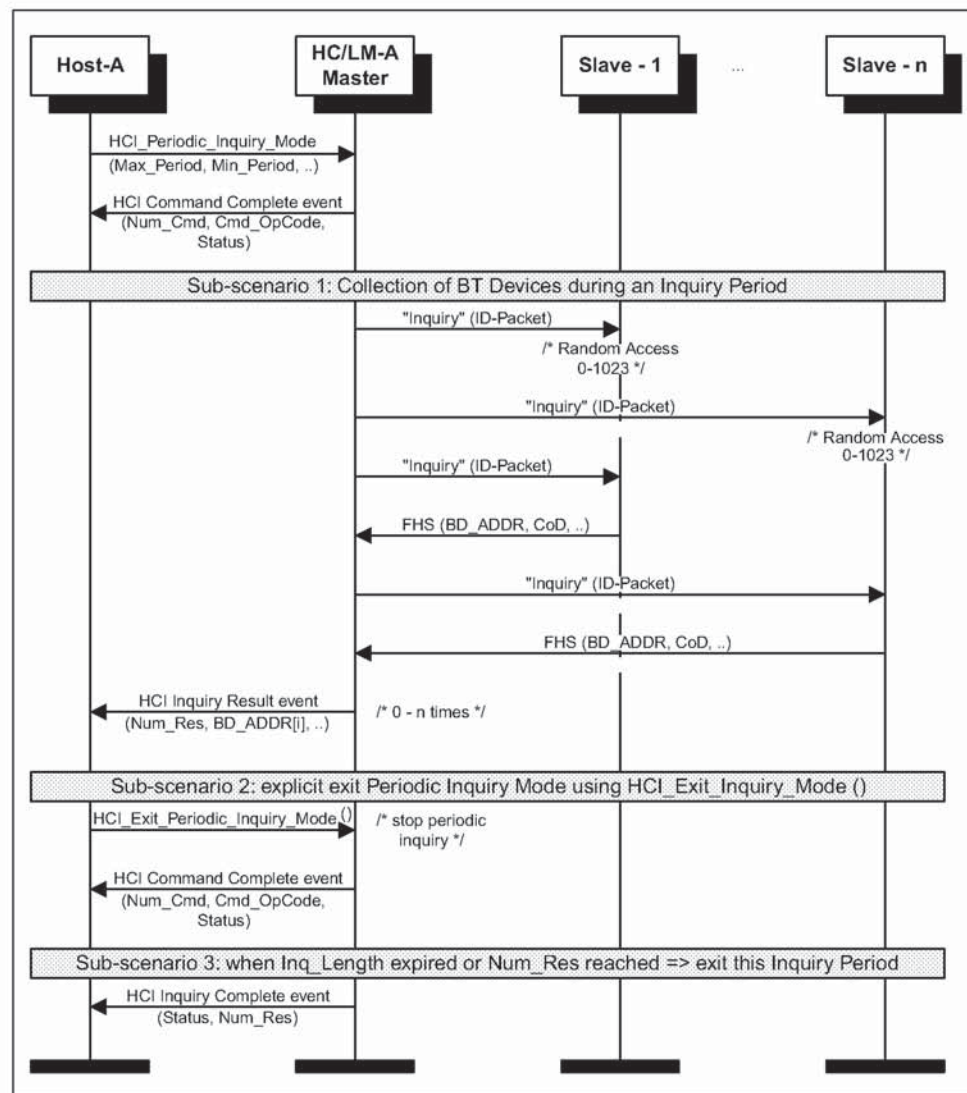


Figure 2.3: Periodic Inquiry

### 3 ACL CONNECTION ESTABLISHMENT AND DETACHMENT

The overview of the ACL Connection establishment and detachment is shown in Figure 3.1 Overview of ACL Connection establishment and detachment.

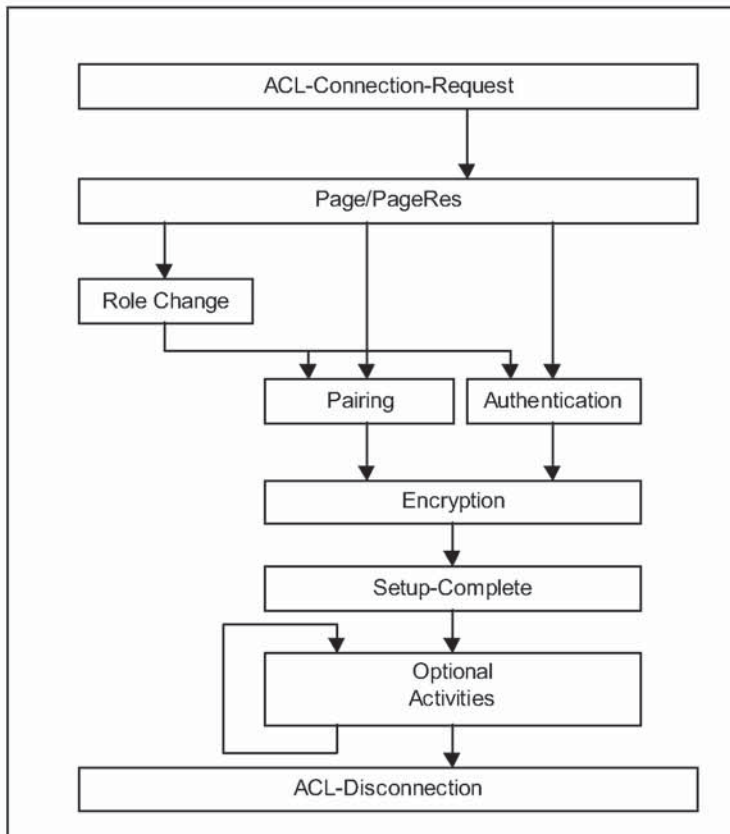


Figure 3.1: Overview of ACL Connection establishment and detachment

### 3.1 ACL CONNECTION REQUEST PHASE

The ACL Connection Request phase is identified between the HCI\_Create\_Connection (BD\_ADDR, Packet\_Type, Page\_Scan\_Repetition\_Mode, Page\_Scan\_Mode, Clock\_Offset, Allow\_Role\_Switch) from the master side and the response from the slave side with rejection or acceptance on the LM level. Three alternative sub-scenarios are shown in Figure 3.2, "ACL Connection Request phase," on page 1044.

#### Sub-scenario 1: Slave rejects ACL Connection Request

If the ACL Connection request is rejected by slave, a Connection Complete event (Status, Connection\_Handle, BD\_ADDR, Link\_Type, Encryption\_Mode) will be then returned to Host, whereby the Status will be copied from the Reason parameter of the command HCI\_Reject\_Connection\_Request (Reason, BD\_ADDR). The parameters Connection\_Handle and Encryption\_Mode will be meaningless.

#### Sub-scenario 2: Slave accepts ACL Connection Request

When the slave responds with LMP\_accepted ( ) correspondent to LMP\_host\_connection\_req ( ), the ACL Connection Request is accepted. The master will continue with the ACL Connection Setup, where pairing, authentication or encryption will be executed.

#### Sub-scenario 3: Slave accepts ACL Connection Request with Role Change

This case is identified when the slave sends an LMP\_switch\_req ( ) to initiate Role Change. If the master accepts, the baseband Master-Slave Switch will be executed. Thereafter, the ACL Connection Setup will continue.

Note: on the slave side, an incoming connection request can be automatically accepted by using HCI\_Set\_Event\_Filter (Filter\_Type, Filter\_Condition\_Type, Condition) with the Filter\_Type = 0x02 /\*Connection\_Setup\*/.

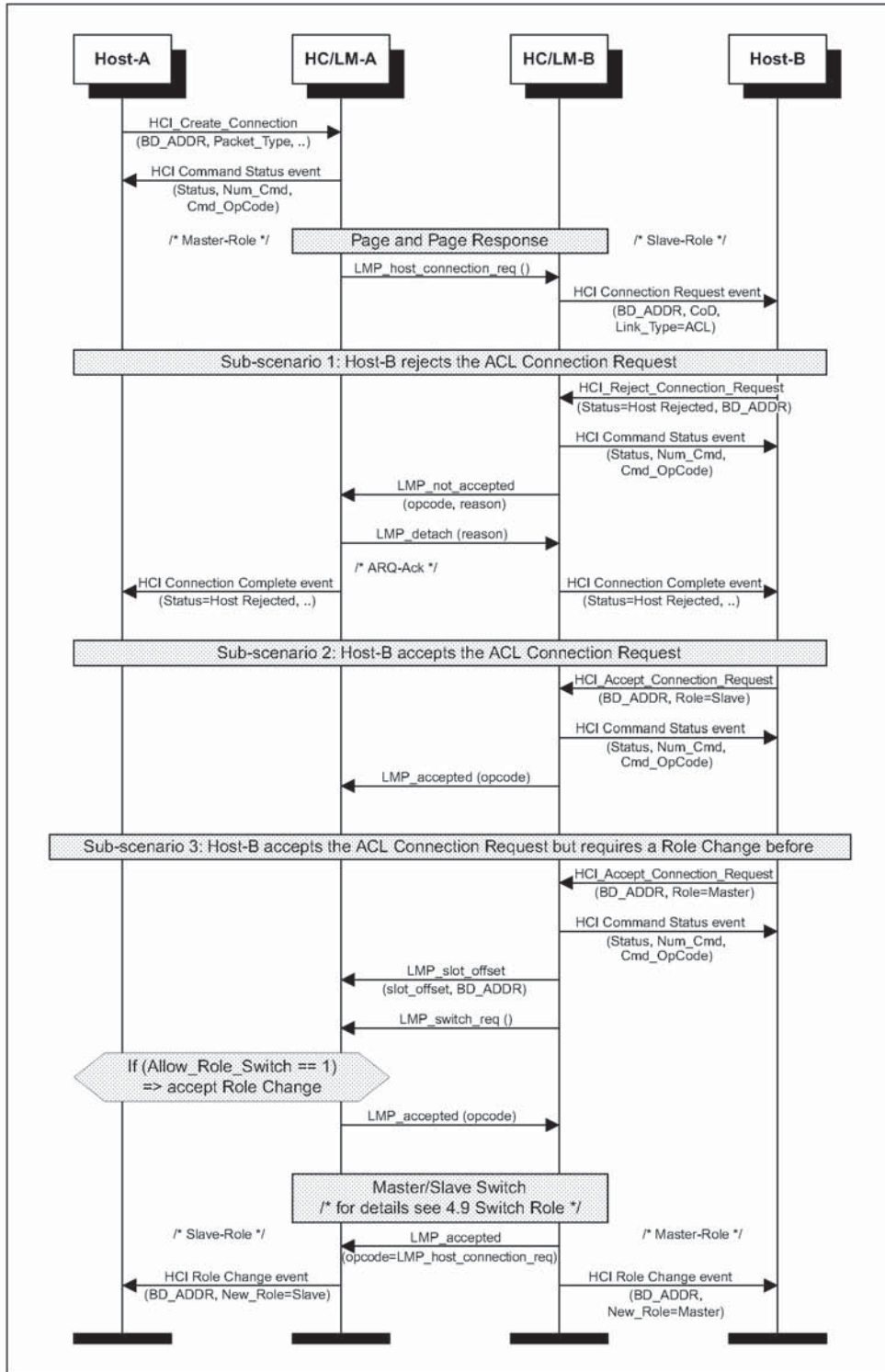


Figure 3.2: ACL Connection Request phase

## 3.2 ACL CONNECTION SETUP PHASE

If the ACL Connection Request phase was successful, the ACL Connection Setup phase will start, with the goal of executing security procedures like pairing, authentication and encryption. The ACL Connection Setup phase is successfully finished when LMP\_setup\_complete ( ) is exchanged and the Connection Complete event (Status=0x00, Connection\_Handle, BD\_ADDR, Link\_Type, Encryption\_Mode) is sent to the Host.

### 3.2.1 Pairing

If authentication is required and the BT Devices to be connected don't have a common link key, the pairing procedure on LM Level will be executed using the PIN Input from Host. During the pairing, the authentication- and link key creation procedures will be done. Note: the created Link Key can be stored either in the BT Device or in the Host.

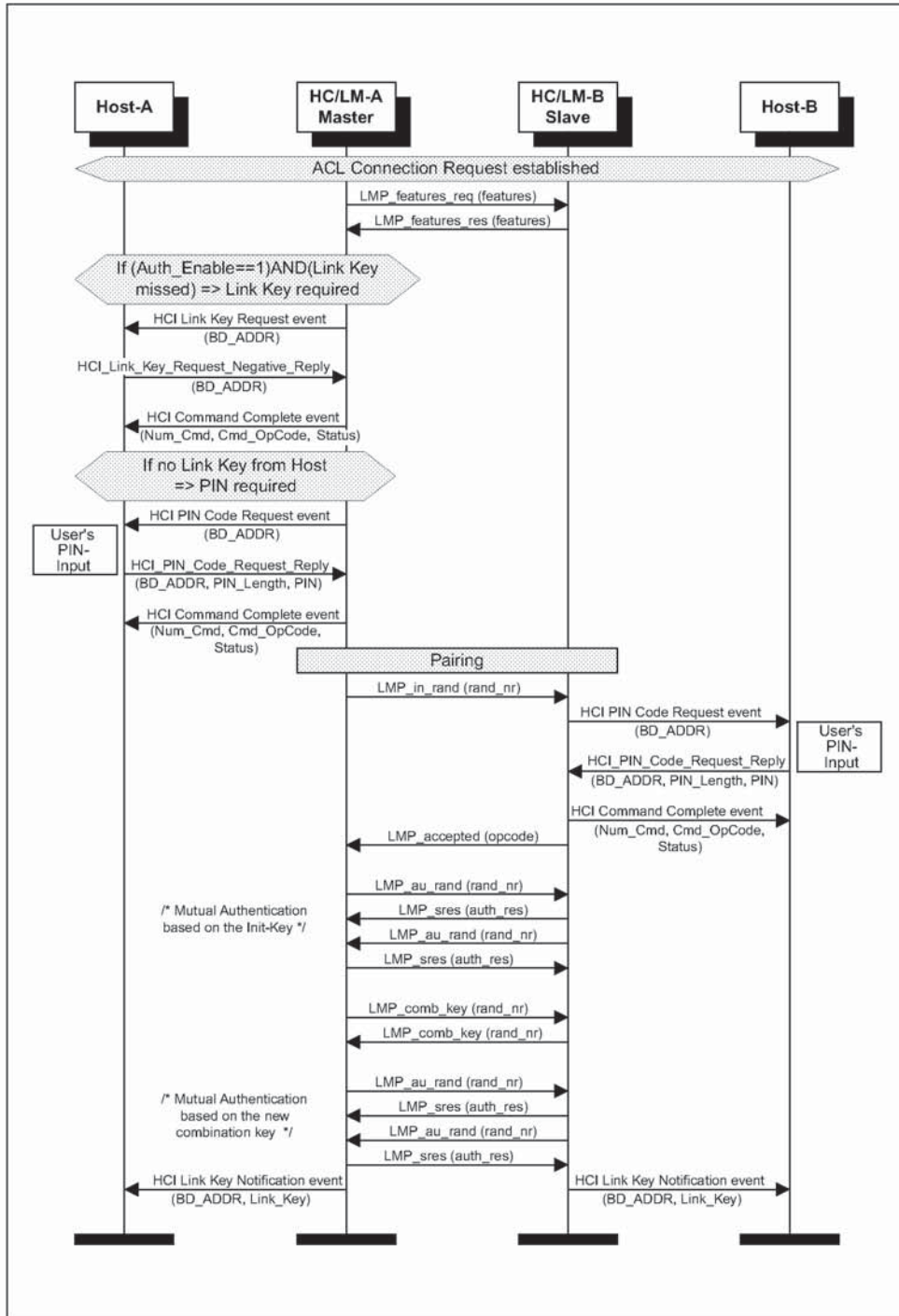


Figure 3.3: ACL Connection setup with pairing



**3.2.2 Authentication**

If a common link key already exists between the BT Devices, pairing is not needed. Note: a Link Key created during pairing can be stored either in the BT Device or in the Host. If the parameter Authentication\_Enable is set, the authentication procedure has to be executed. Here, the MSC only shows the case when Authentication\_Enable is set on both sides.

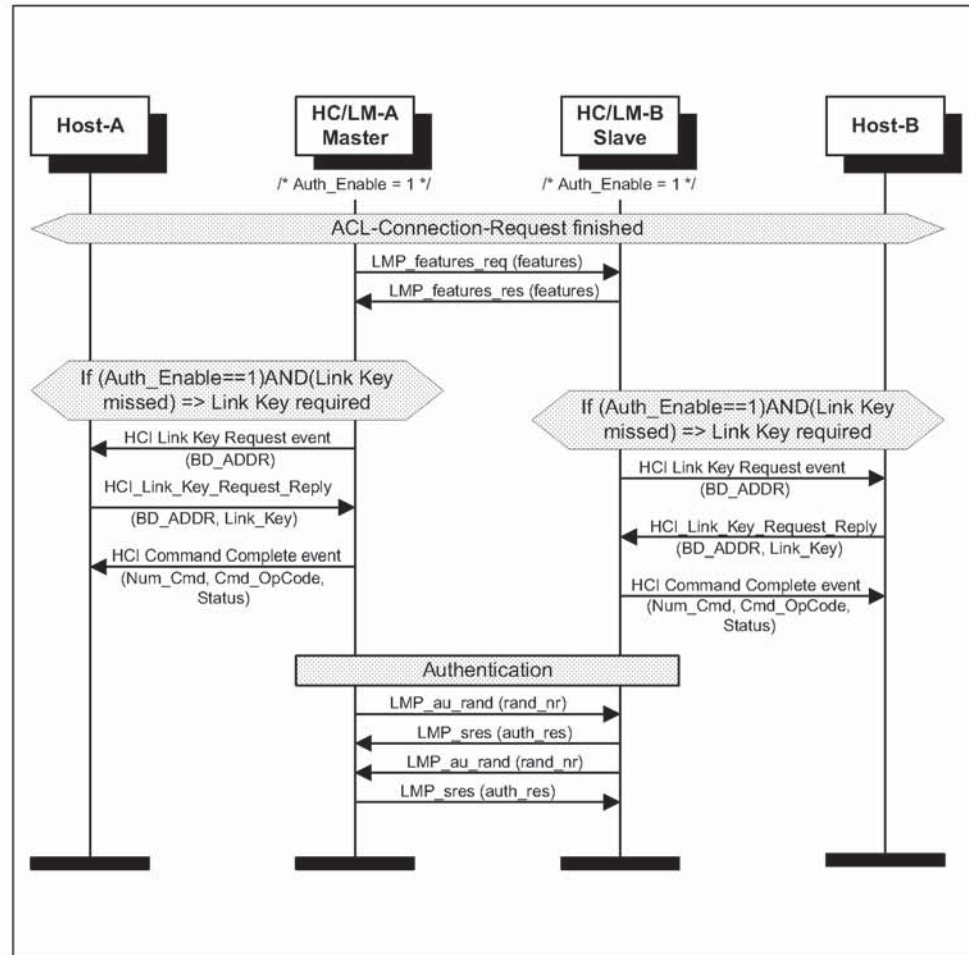


Figure 3.4: ACL Connection setup with authentication

**3.3 ENCRYPTION AND CONNECTION SETUP COMPLETE**

Once the pairing/authentication procedure is successful, the encryption procedure will be started. Here, the MSC only shows how to set up an encrypted point-to-point connection (Encryption\_Mode = 1 /\*point-to-point/). Note: an encrypted connection requires an established common link key.

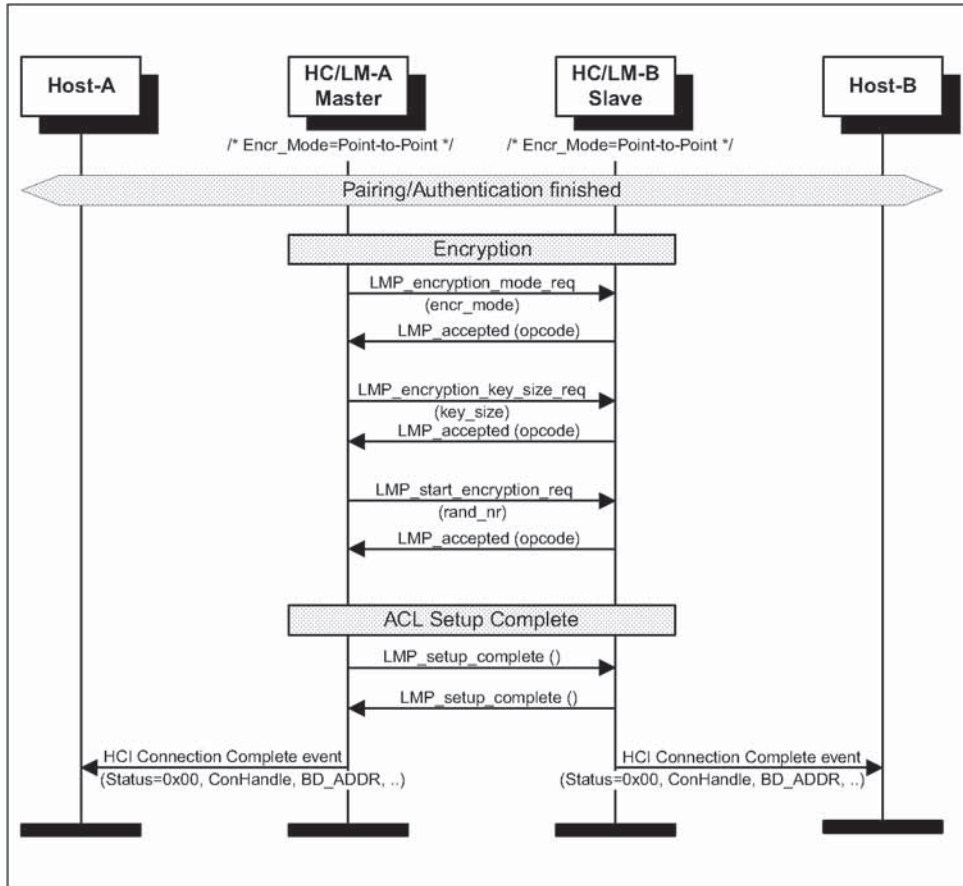


Figure 3.5: Encryption and Setup complete

### 3.4 ACL DISCONNECTION

At any time, an established ACL Connection can be detached by an HCI\_Disconnect (Connection\_Handle, Reason). If one or several SCO Connections exist, they must first be detached before the ACL Connection can be released.

Note: the disconnection procedure is one-sided and doesn't need an explicit acknowledgment from the remote LM. So the ARQ Acknowledgment from the LC is needed, to ensure that the remote LM has received the LMP\_detach (reason).

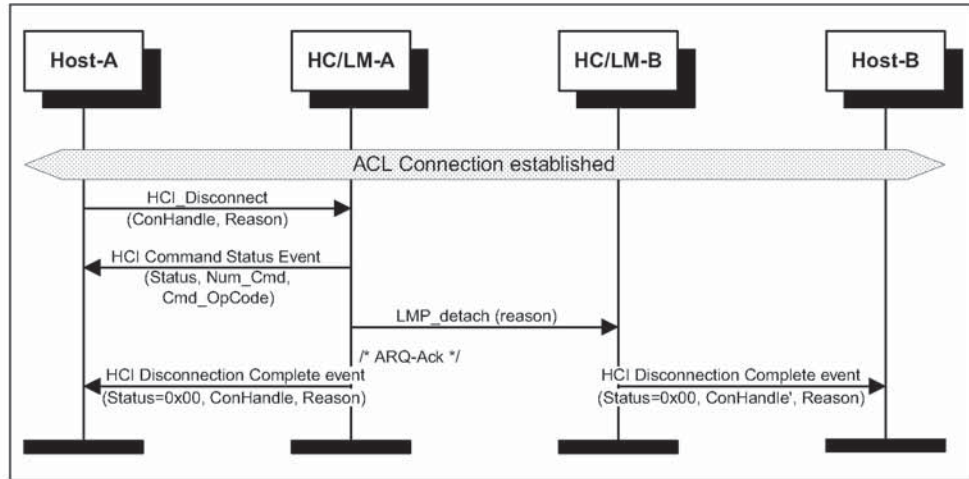


Figure 3.6: ACL Disconnection

## 4 OPTIONAL ACTIVITIES AFTER ACL CONNECTION ESTABLISHMENT

### 4.1 AUTHENTICATION REQUESTED

Authentication can be explicitly executed at any time after an ACL Connection has been established. If the Link Key was missed in HC/LM, the Link Key will be required from the Host, as in the authentication procedure (see 3.2.2).

Note: if the HC/LM and the Host don't have the Link Key a PIN Code Request event will be sent to the Host to request a PIN Code for pairing. A procedure identical to ACL Connection Setup with Pairing (see 3.2.1) will be used.

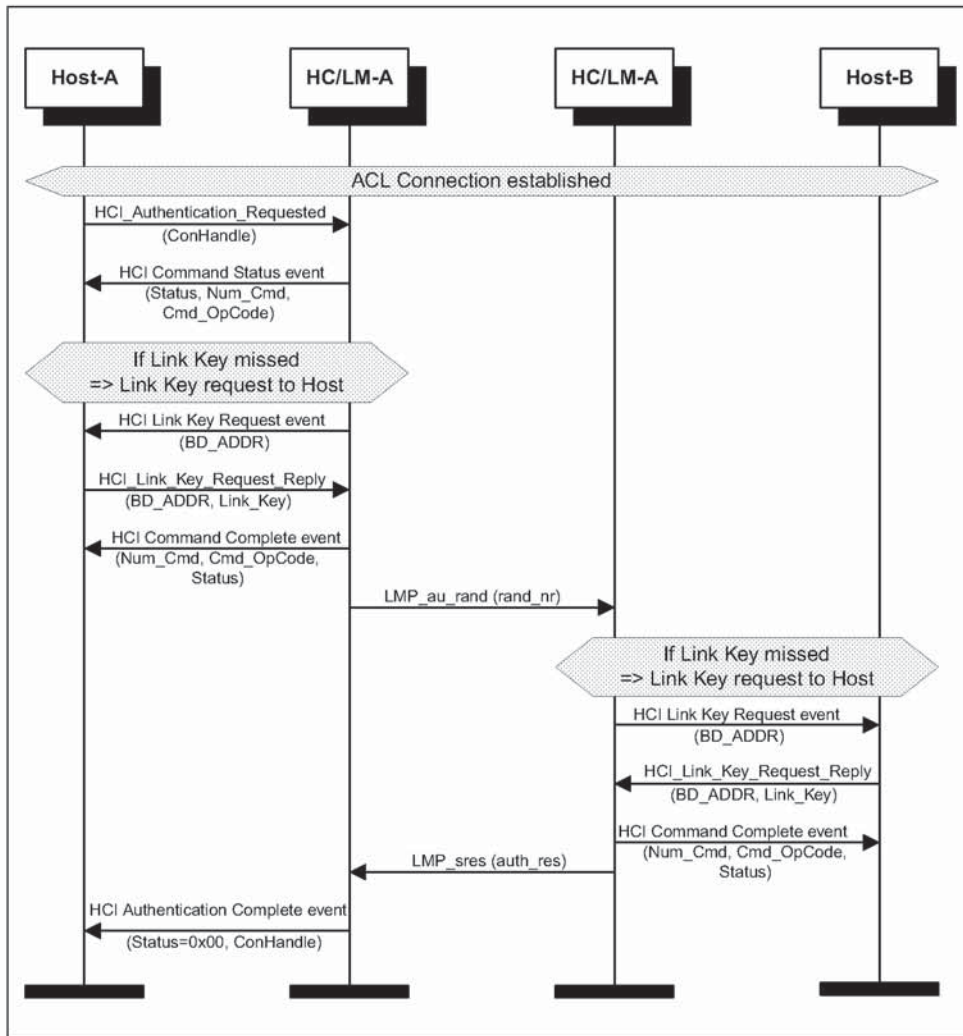


Figure 4.1: Authentication Requested

## 4.2 SET CONNECTION ENCRYPTION

Using the command HCI\_Set\_Connection\_Encryption (Connection\_Handle, Encryption\_Enable), the Host is able to switch the encryption of a connection with the specified Connection\_Handle to ON/OFF. This command can be applied on both the master- and slave sides (only the master side is shown in Figure 4.2 Set Connection Encryption). If this command occurs on the slave side, the only difference is that LMP\_encryption\_mode\_req (encryption\_mode) will be sent from the HC/LM Slave. LMP\_encryption\_key\_size\_req (key\_size) and LMP\_start\_encryption\_req (rand\_nr) will still be requested from the HC/LM master.

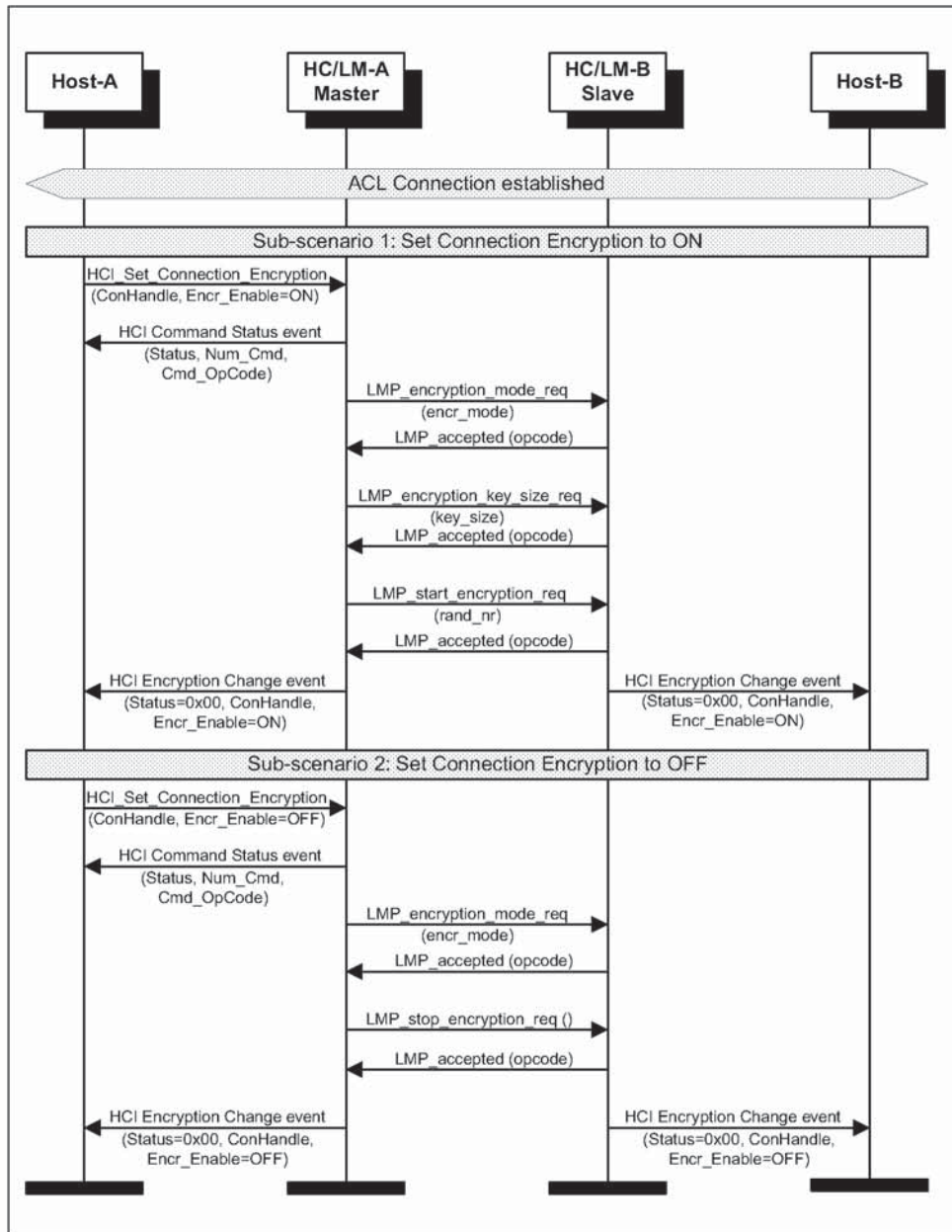


Figure 4.2: Set Connection Encryption

### 4.3 CHANGE CONNECTION LINK KEY

Using the command `HCI_Change_Connection_Link_Key` (Connection\_Handle), the Host can explicitly change the common link key shared between the BT Devices.

Note: if the connection encryption was enabled and the temporary link key was used, it is the task of the BT Master to automatically restart the encryption (first stop and then restart) after the link key is successfully changed.

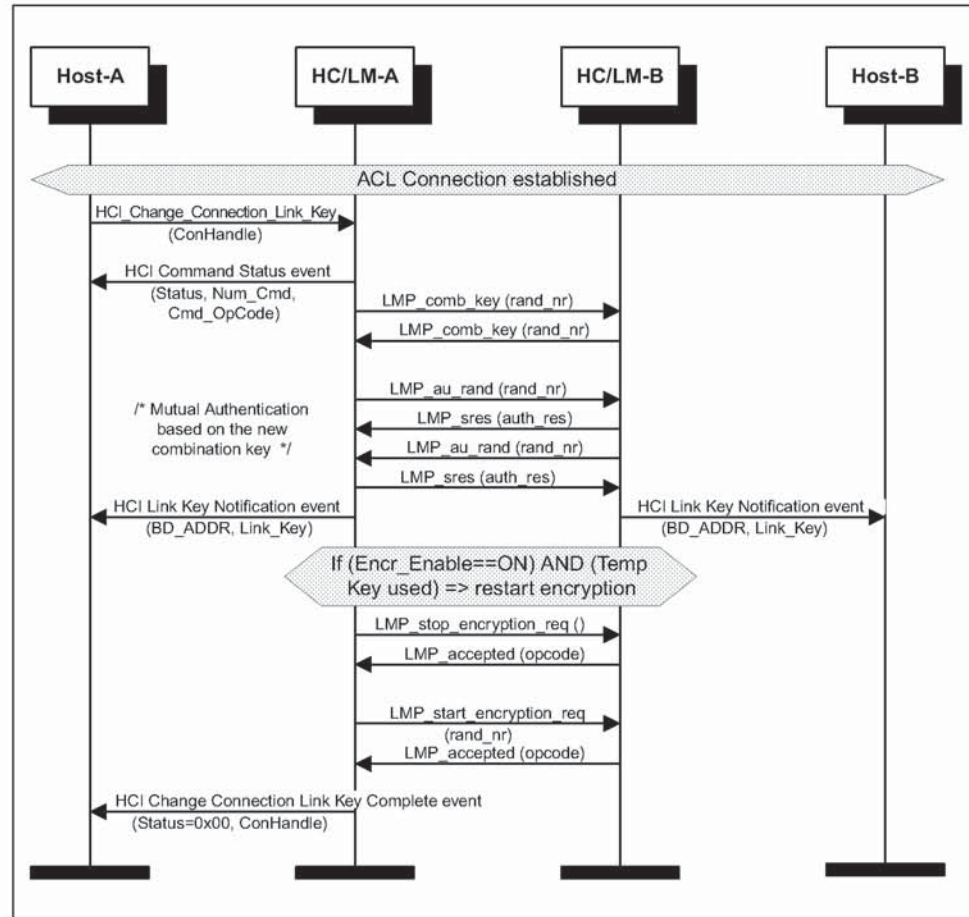


Figure 4.3: Change Connection Link Key

### 4.4 MASTER LINK KEY

The Figure 4.4 Master Link Key shows how the Host can explicitly switch between the temporary Link Key and the semi-permanent Link Key. Since this command can only be used for the BT Master, the Link Key switch will affect all connections.

Note: if encryption was enabled, it is the task of the BT Master to restart the encryption separately for each slave.

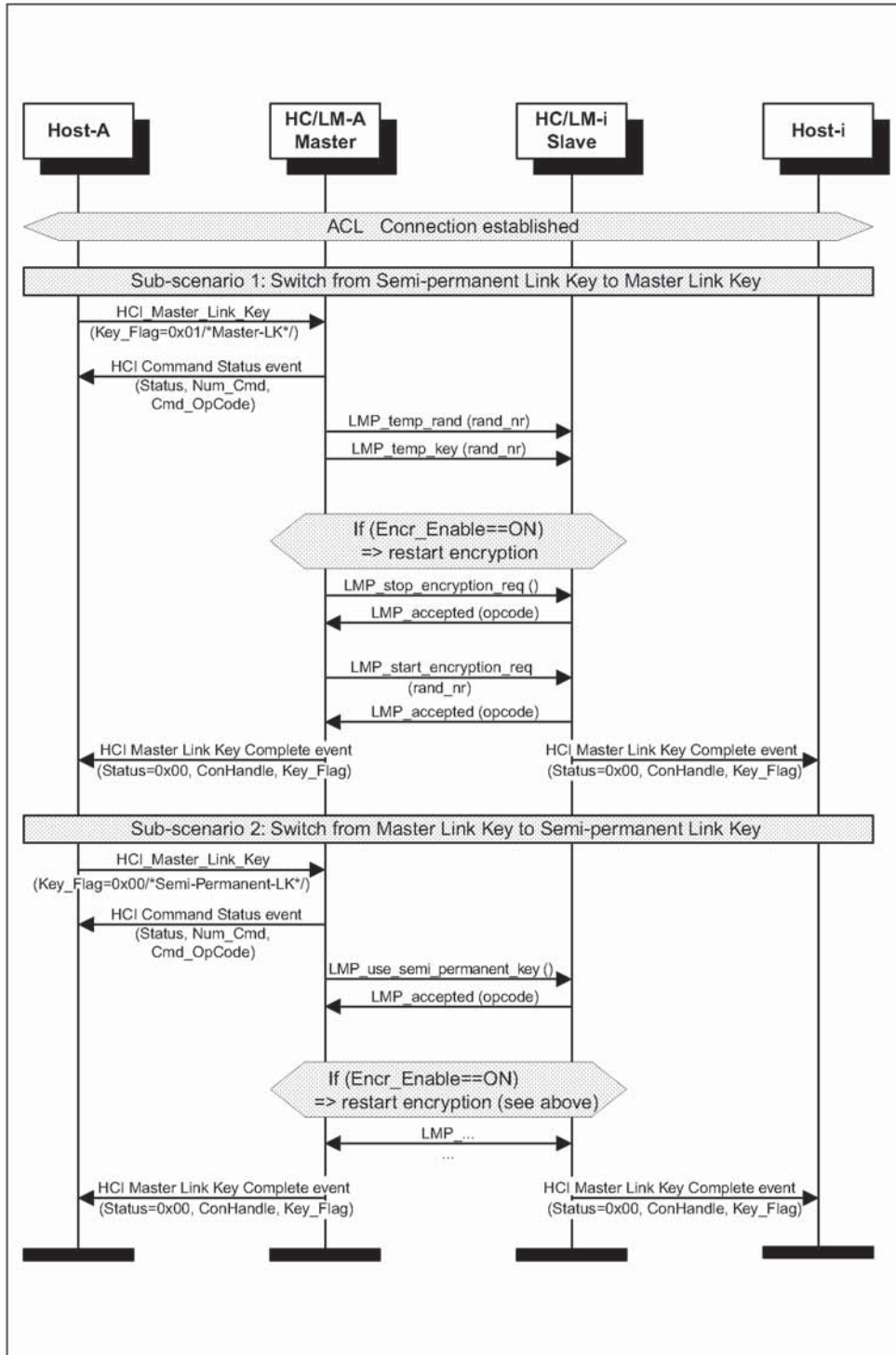


Figure 4.4: Master Link Key



### 4.5 READ REMOTE SUPPORTED FEATURES

Using the command `HCI_Read_Remote_Supported_Features` (`Connection_Handle`) the supported LMP Features of a remote BT Device can be read. These features contain supported packet types, supported modes, supported audio coding modes, etc.

Note: if the LMP Features was exchanged during ACL Connection Setup, the HC/LM A may return the Read Remote Supported Features Complete event (`Status`, `Connection_Handle`, `LMP_Features`) without exchange of LMP PDUs.

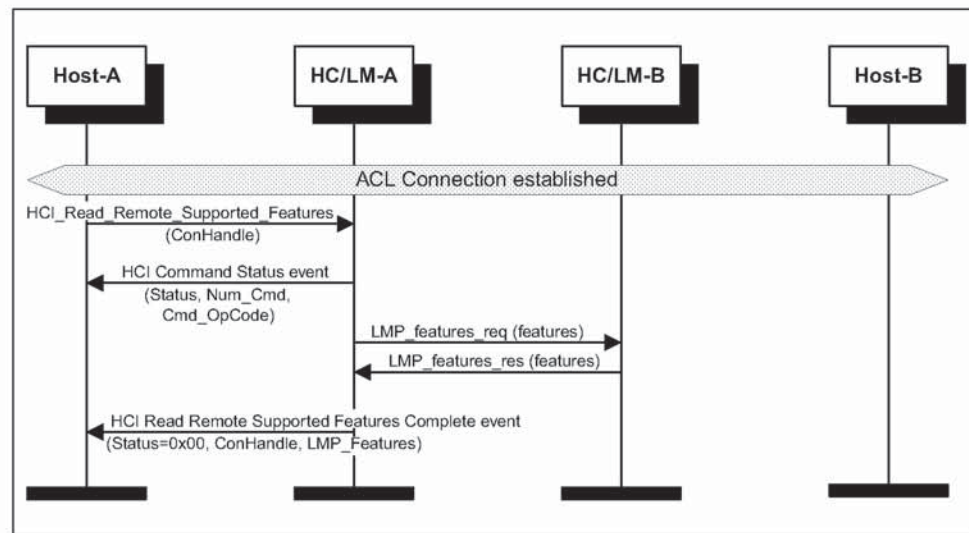


Figure 4.5: Read Remote Supported Features

### 4.6 READ CLOCK OFFSET

Using the command `HCI_Read_Clock_Offset` (`Connection_Handle`) the BT Master can read the Clock Offset of the BT Slaves. The Clock Offset can be used to speed up the paging procedure in a later connection attempt. If the command is requested from the slave device, the HC/LM Slave will directly return a Command Status event and an Read Clock Offset Complete event without exchange of LMP PDUs.

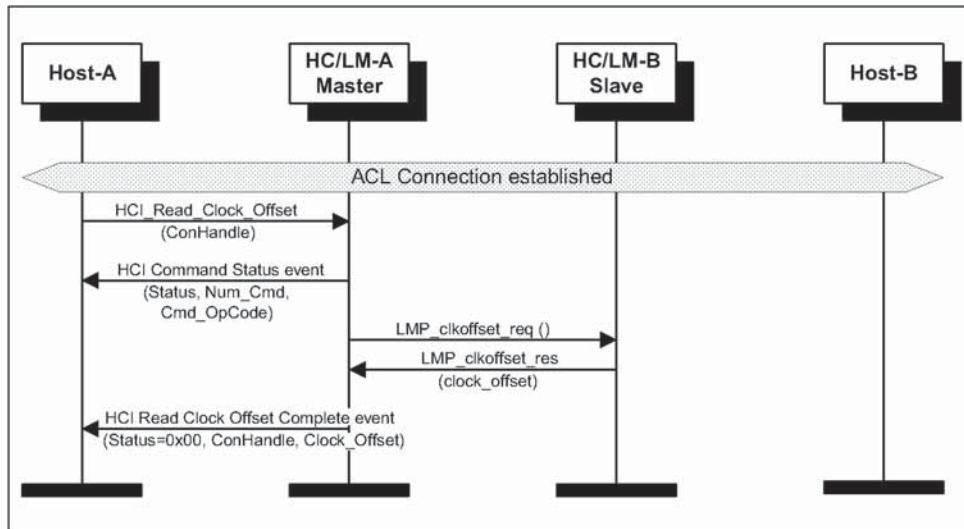


Figure 4.6: Read Clock Offset

### 4.7 READ REMOTE VERSION INFORMATION

Using HCI\_Read\_Remote\_Version\_Information (Connection\_Handle) the version information consisting of LMP\_Version, Manufacturer\_Name and LMP\_Subversion from the remote BT Device can be read.

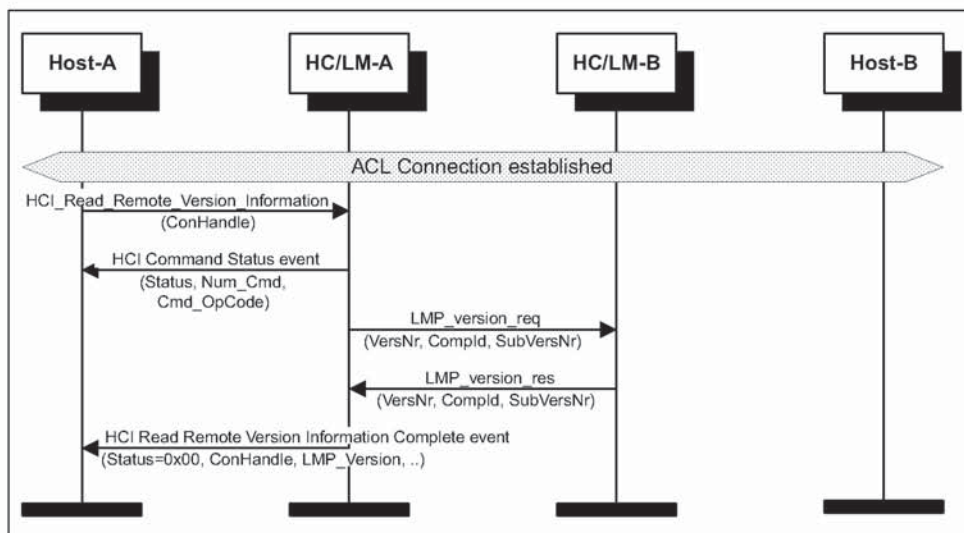


Figure 4.7: Read Remote Version Information

### 4.8 QoS SETUP

To set up the Quality of Service, the command HCI\_QoS\_Setup (Connection\_Handle, Flags, Service\_Type, Token\_Rate, Peak\_Bandwidth, Latency, Delay\_Variation) is used.

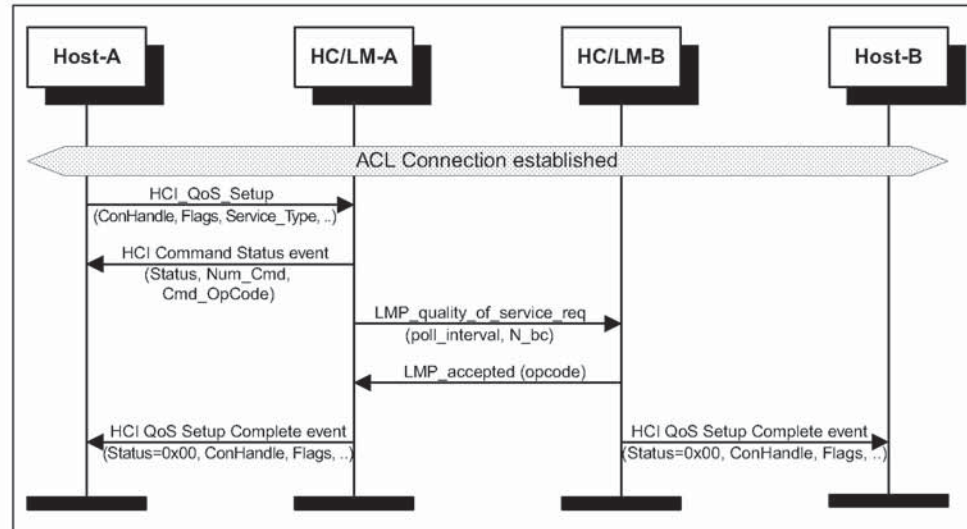


Figure 4.8: QoS Setup

### 4.9 SWITCH ROLE

The command HCI\_Switch\_Role (BD\_ADDR, Role) can be used to explicitly switch the current role of the local BT Device for a particular connection with the specified BT Device (BD\_ADDR). The local HC/LM has to check whether the switch is performed or not.

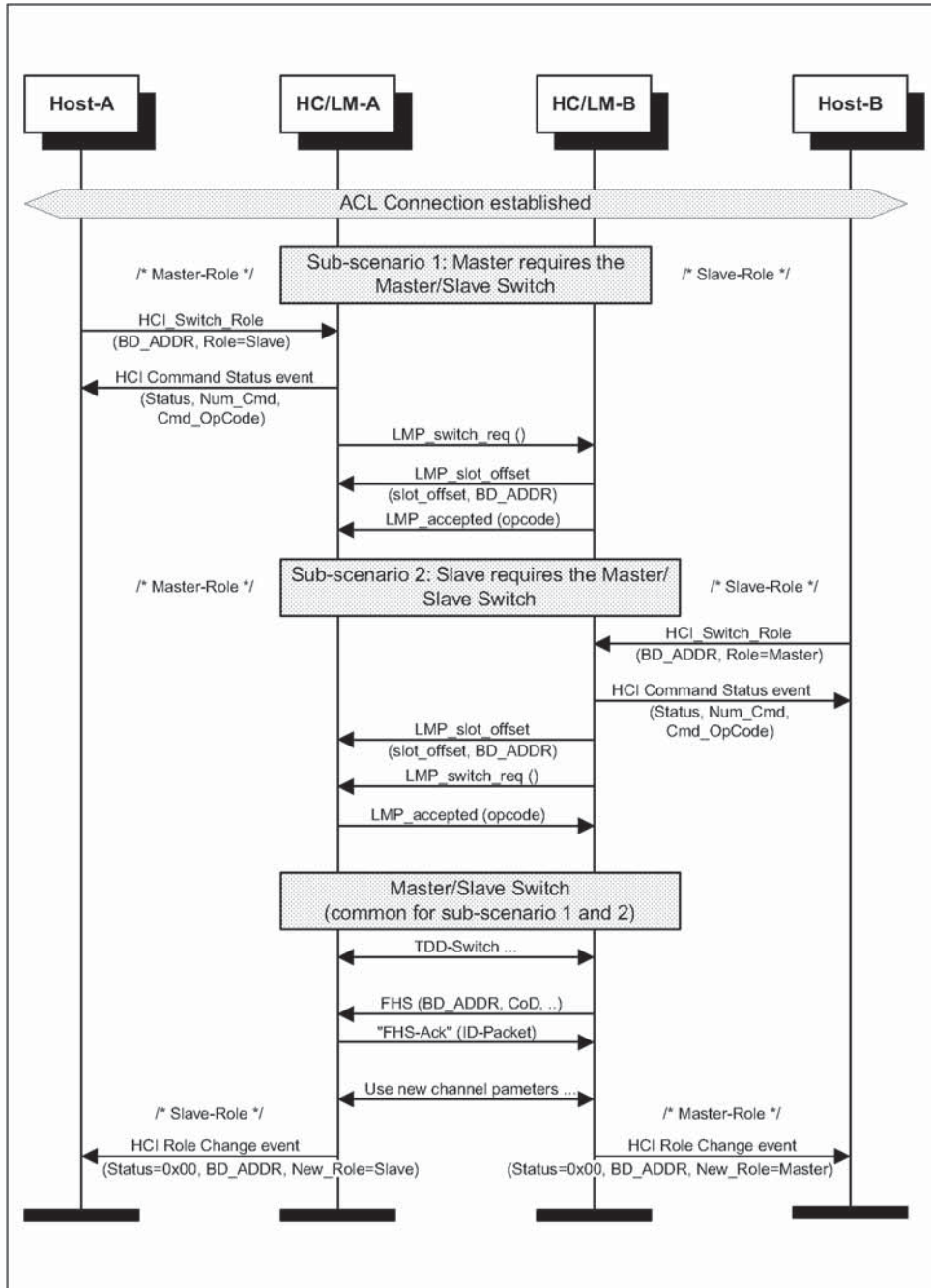


Figure 4.9: Switch Role

## 5 SCO CONNECTION ESTABLISHMENT AND DETACHMENT

### 5.1 SCO CONNECTION SETUP

SCO Connection setup requires an established ACL Connection. It is the task of the Host to create an ACL Connection first and then the SCO Link.

Note: On the slave side, an incoming connection request can be automatically accepted by using HCI\_Set\_Event\_Filter (Filter\_Type, Filter\_Condition\_Type, Condition) with the Filter\_Type = 0x02 /\*Connection\_Setup\*/. Furthermore, for each SCO Link to a BT Device, a separate SCO Connection Handle is needed.

#### 5.1.1 Master activates the SCO Connection setup

To set up an SCO Connection, the HCI\_Add\_SCO\_Connection (Connection\_Handle, Packet\_Type) command is used. The specified Connection\_Handle is related to the ACL Connection that must have been created before the HCI\_Add\_SCO\_Connection is issued.

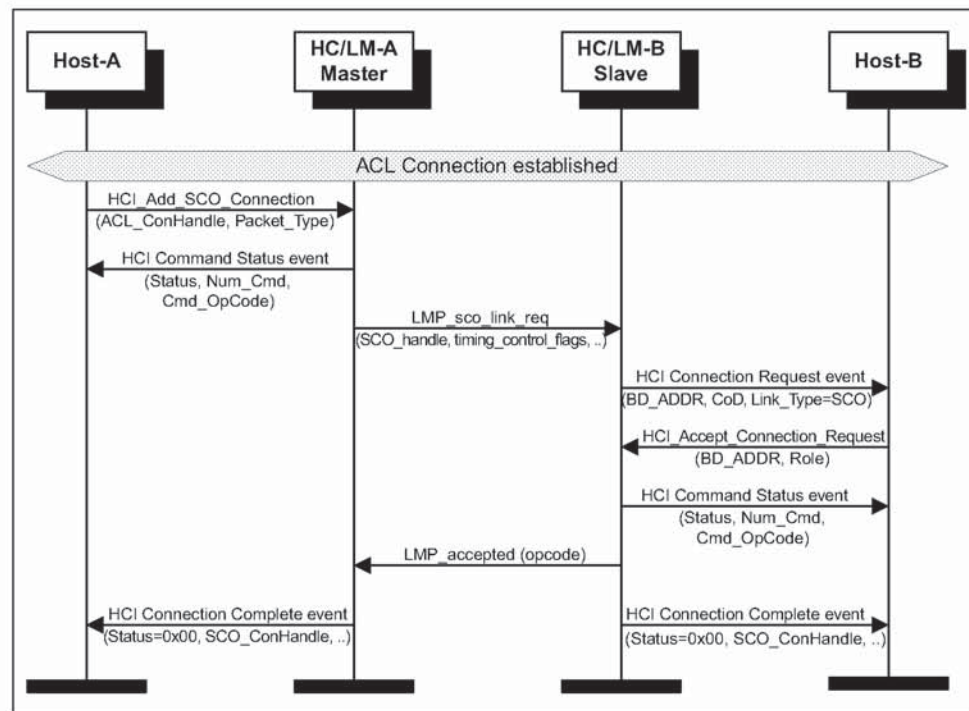


Figure 5.1: SCO Connection setup (activated from master)

**5.1.2 Slave activates the SCO Connection setup**

The same command HCI\_Add\_SCO\_Connection (Connection\_Handle, Packet\_Type) can be used to create an SCO Link when the local BT Device is a BT Slave. Here the specified Connection\_Handle belongs to the established ACL Connection between the BT Devices. Compared to 5.1.1, the only difference is that the HC/LM Slave starts the SCO Setup with LMP\_sco\_link\_req first.

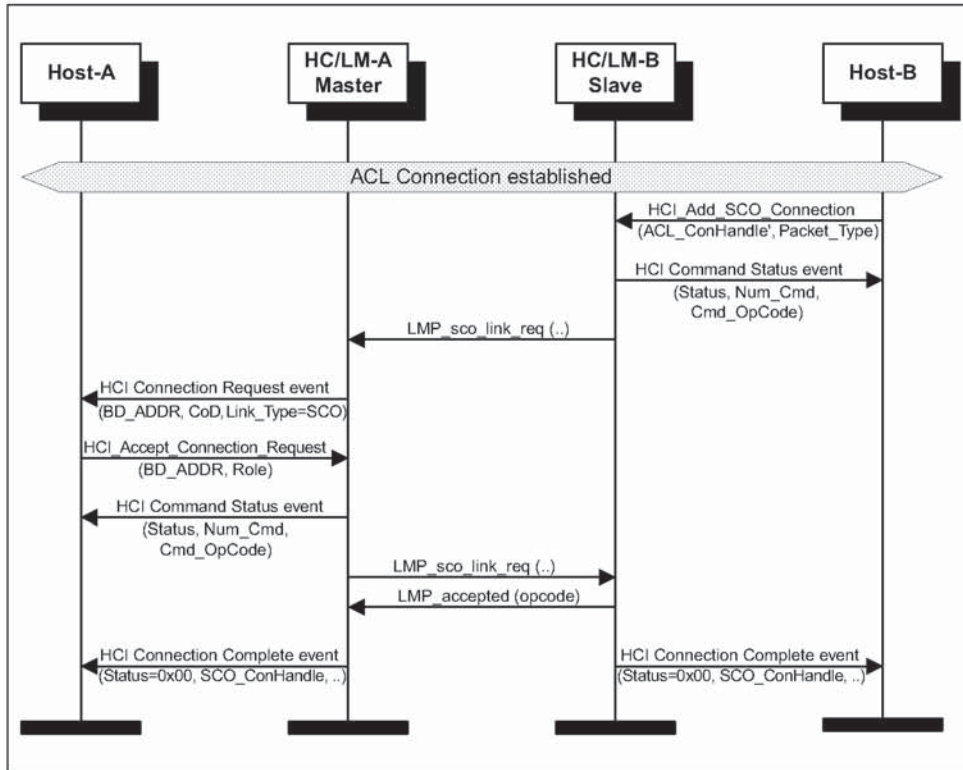


Figure 5.2: SCO Connection setup (activated from slave)

**5.2 SCO DISCONNECTION**

An established SCO Connection can be detached at any time. Since several SCO Connections can exist between a BT Master and a BT Slave, an SCO Disconnection only removes the SCO Link with the specified SCO Connection Handle. The other SCO Connections will still exist.

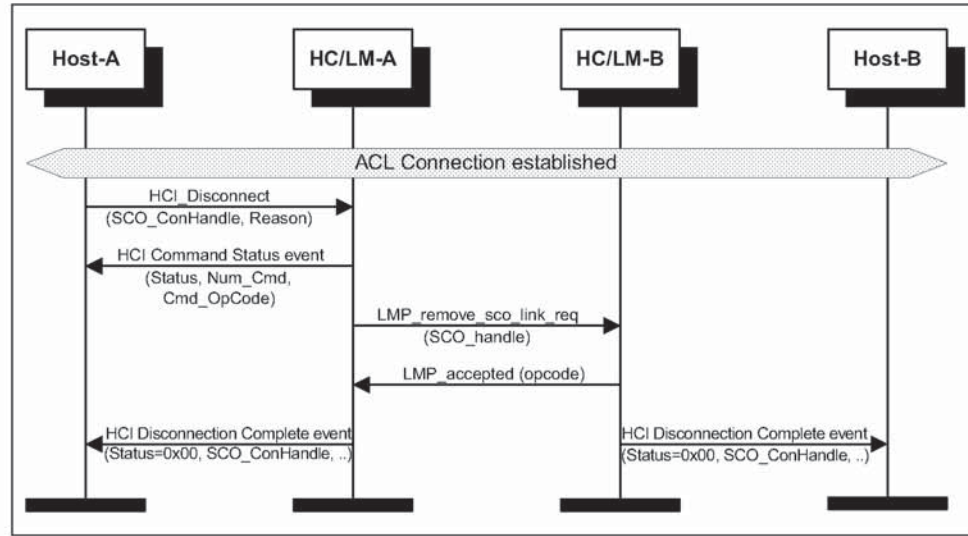


Figure 5.3: SCO Disconnection

## 6 SPECIAL MODES: SNIFF, HOLD, PARK

Entry into sniff, hold or park mode requires an established ACL Connection. The following table summarizes the modes and the BT Role that can request, force, activate or exit the modes.

	Sniff	Hold	Park
<b>Request</b>	Master/Slave	Master/Slave	Master/Slave
<b>Force</b>	Master	Master/Slave	Master
<b>Activation</b>	Master	Master/Slave	Master
<b>Release</b>	Master/Slave	Automatic	Master/Slave

Table 6.1: Summary of modes (Sniff, Hold, Park)

### 6.1 SNIFF MODE

Sniff Mode is used when a slave shall participate in the piconet only in a sniff interval. For the Sniff Mode negotiation, the Host specifies the Sniff\_Max\_Interval and the Sniff\_Min\_Interval so that HC/LM will be able to choose the one sniff interval in this range. The used command is HCI\_Sniff\_Mode (Connection\_Handle, Sniff\_Max\_Interval, Sniff\_Min\_Interval, Sniff\_Attempt, Sniff\_Timeout).

Since Sniff Mode is a periodic mode, the command HCI\_Exit\_Sniff\_Mode (Connection\_Handle) is needed to return to Active Mode.



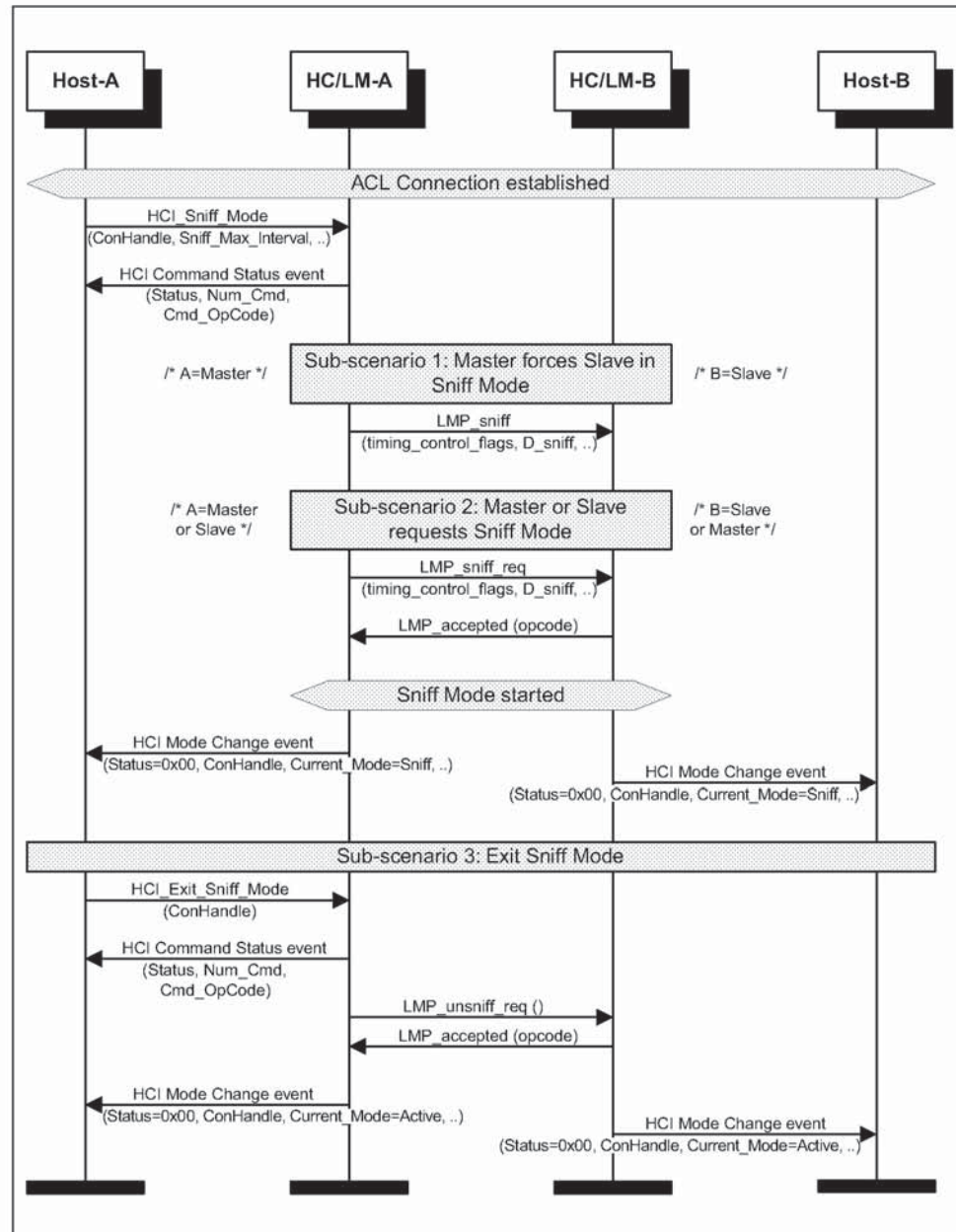


Figure 6.1: Sniff Mode

## 6.2 HOLD MODE

Hold Mode is useful when a BT Device doesn't want to participate in the connection for a Hold Mode Length. Using the command `HCI_Hold_Mode` (Connection\_Handle, Hold\_Max\_Length, Hold\_Min\_Length), the Host specifies the Hold\_Max\_Length and Hold\_Min\_Length. The HC/LM will then be able to negotiate a Hold Mode Length in this range. When the hold mode is started

or complete, Mode Change event (Status, Connection\_Handle, Current\_Mode, Interval) will be used to inform the Host about the actual mode.

Note: the Hold Mode is exited when the Hold Mode Length has expired, so it is no guarantee that the remote BT Device is immediately active.

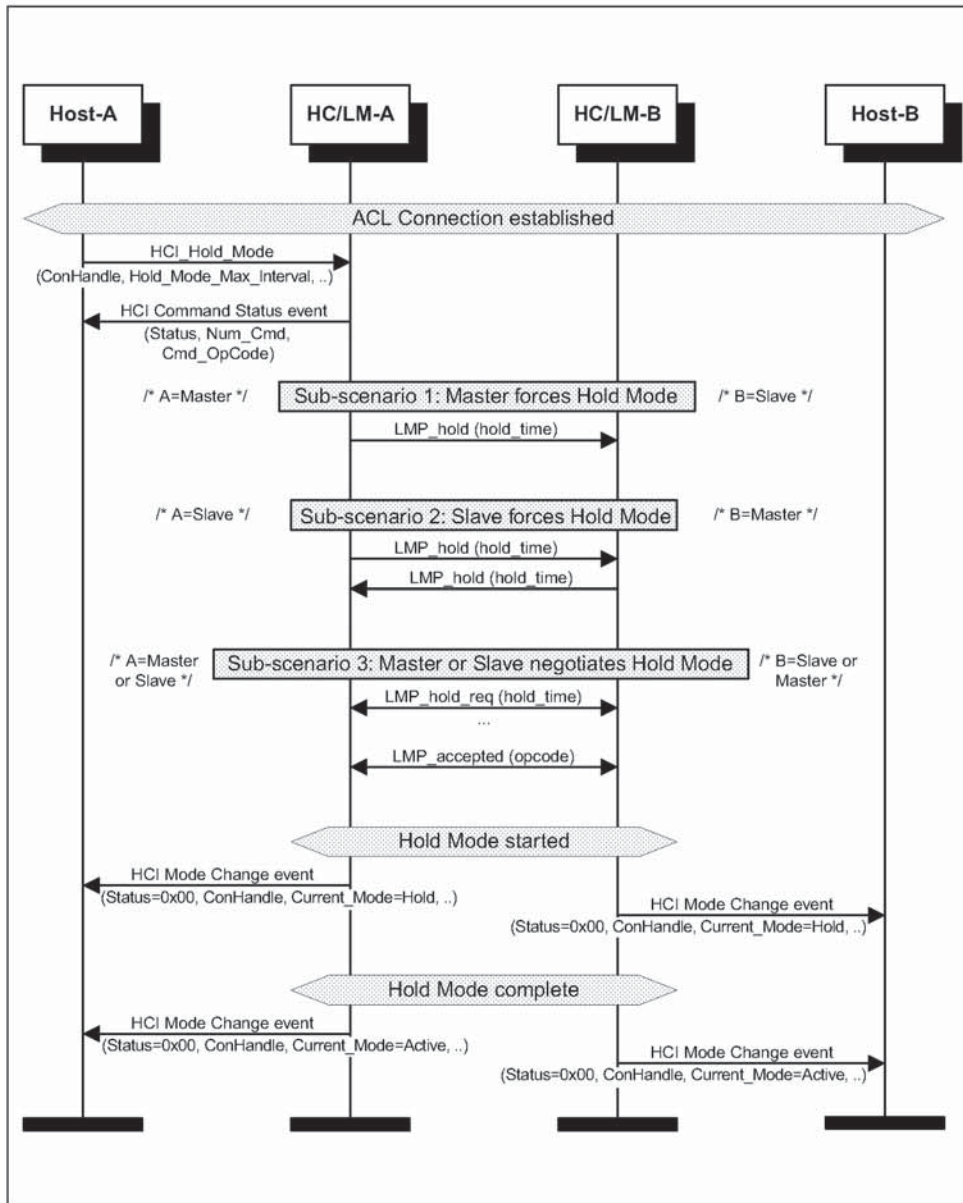


Figure 6.2: Hold Mode

## 6.3 PARK MODE

Park Mode is used to render the slaves inactive but still synchronized to the master using the beacon interval. In park mode, broadcast is performed.

### 6.3.1 Enter park mode

Using the command HCI\_Park\_Mode (Connection\_Handle, Beacon\_Max\_Interval, Beacon\_Min\_Interval) the Host specifies the Beacon\_Max\_Interval and Beacon\_Min\_Interval so that HC/LM can set up a Beacon-Interval in this range for the BT Slaves. In Park Mode, the BT Slave gives up its AM\_ADDR.

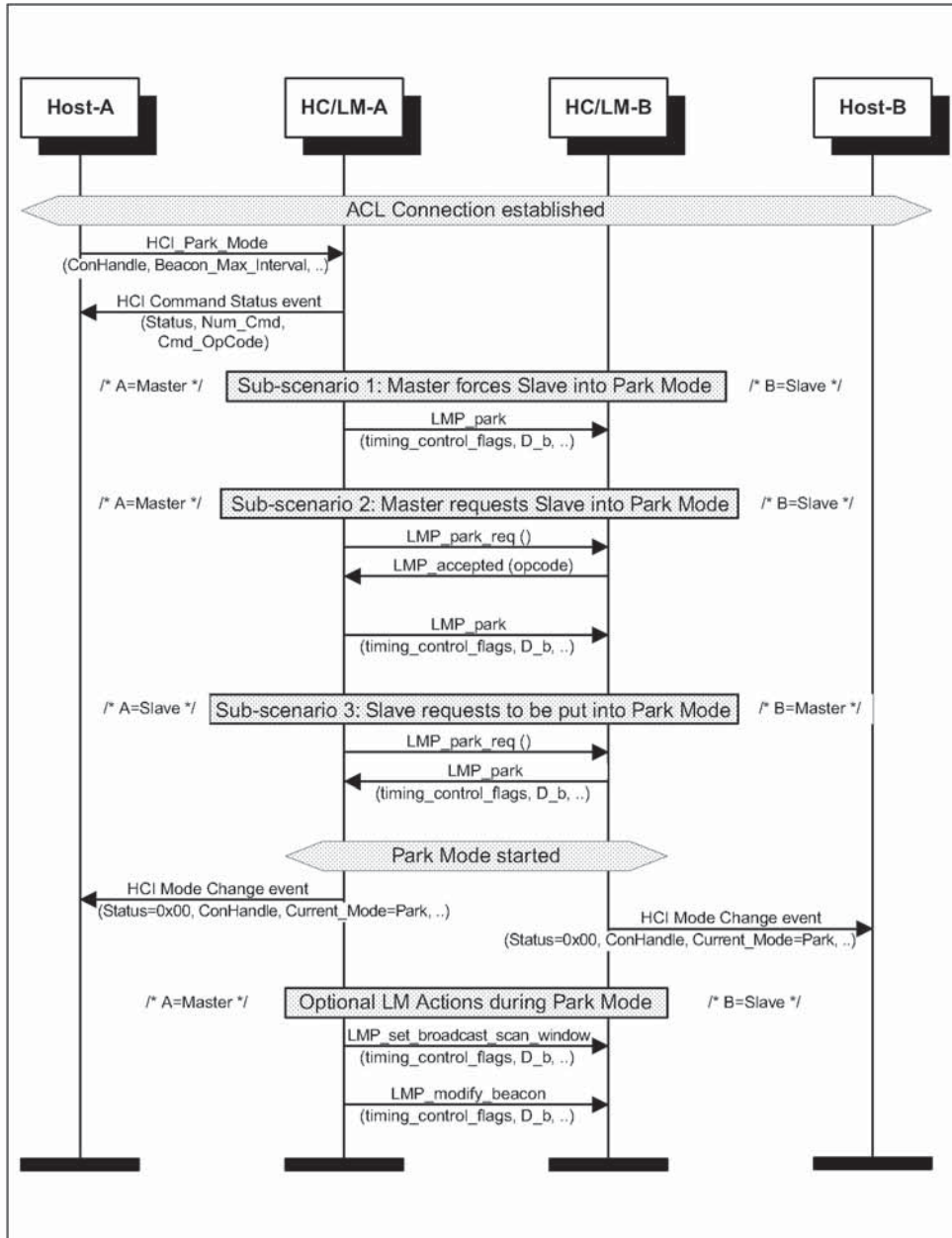


Figure 6.3: Enter Park Mode

### 6.3.2 Exit Park Mode

Since Park Mode is a periodic mode, the command `HCI_Exit_Park_Mode` (Connection\_Handle) will be used to return to Active Mode. A parked BT Slave can send an `Access_Request_Message` to request to leave the Park Mode. It is the task of master HC/LM to use `LMP_unpark_PM_ADDR_req(..)` or `LMP_unpark_BD_ADDR_req(..)` to unpark a BT Slave.

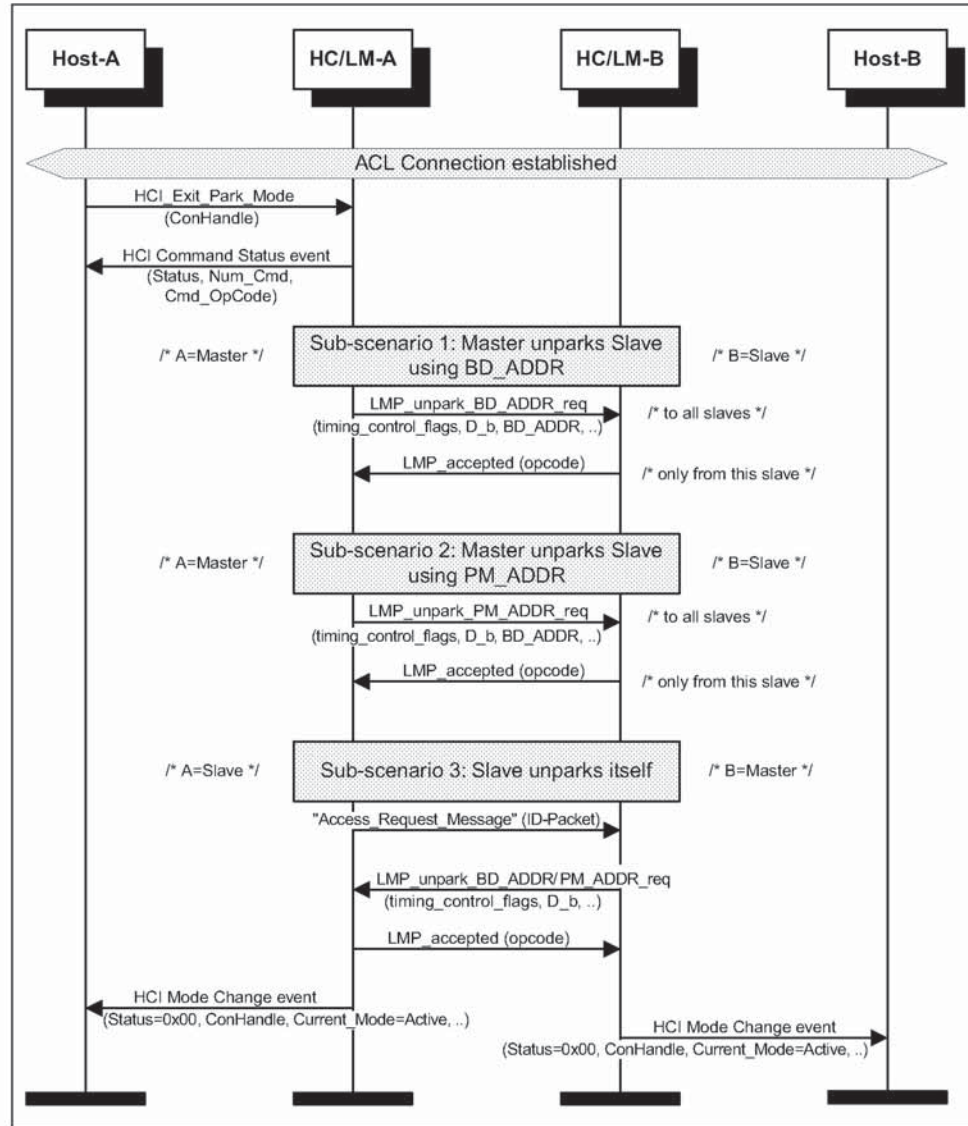


Figure 6.4: Exit Park Mode

## 7 BUFFER MANAGEMENT, FLOW CONTROL

The HC Data buffers are configured by the HC and managed by the Host. On initialization, the Host will issue HCI\_Read\_Buffer\_Size. This specifies the maximum allowed length of HCI data packets sent from the Host to the HC, and the maximum number of ACL and SCO data packets that the HC can store in its buffer. After a connection is created, HC will frequently inform the Host about the number of sent packets using Number Of Completed Packets event (Number\_of\_Handles, Connection\_Handle[i], HC\_Num\_Of\_Completed\_Packets[i]) (see Figure 7.1 Host-to-HC flow control).

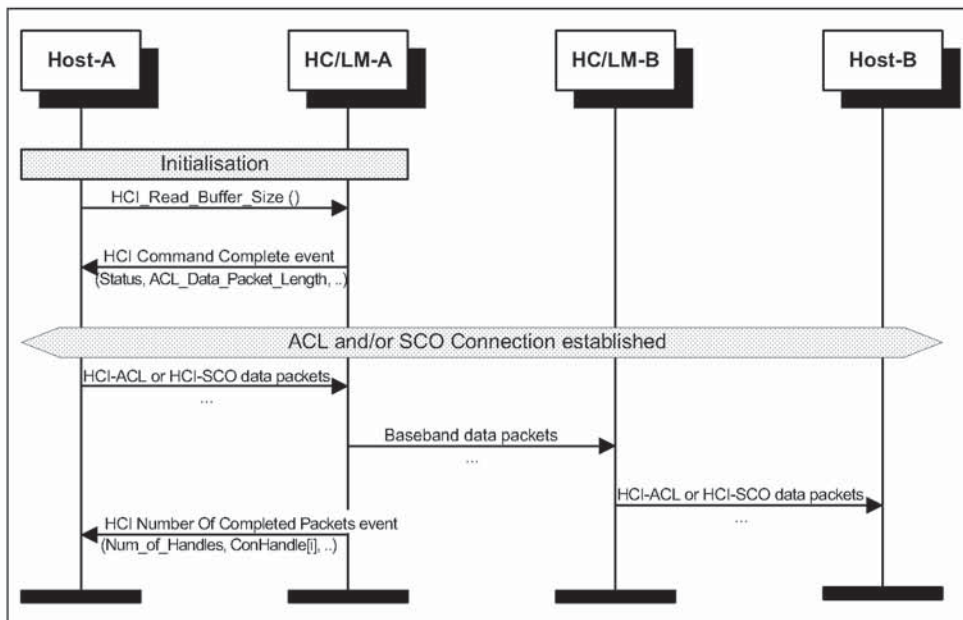


Figure 7.1: Host to HC flow control

Accordingly the HC to Host flow control can be applied in the same way so that during initialization the Host configures the Buffer Size and later the Host Controller will manage the Host Buffers.

Using HCI\_Set\_Host\_Controller\_To\_Host\_Flow\_Control (Flow\_Control\_Enable) the Host can decide to apply the HC to Host flow control or not. For flow control itself HCI\_Host\_Buffer\_Size (Host\_ACL\_Data\_Packet\_Length, Host\_SCO\_Data\_Packet\_Length, Host\_Total\_Num\_ACL\_Data\_Packets, Host\_Total\_Num\_SCO\_Data\_Packets) and HCI\_Host\_Number\_Of\_Completed\_Packets (Number\_of\_Handles, Connection\_Handle[i], Host\_Num\_Of\_Completed\_Packets[i]) will be used (for details see Figure 7.2 HC to Host Flow Control).

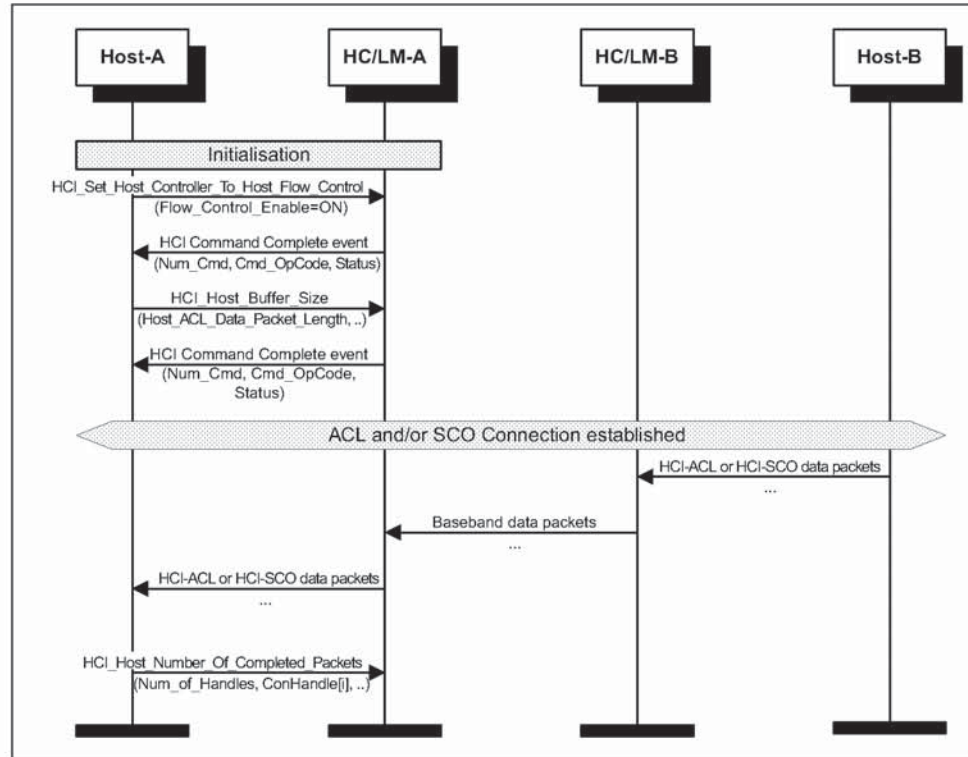


Figure 7.2: HC to Host Flow Control

## 8 LOOPBACK MODE

---

### 8.1 LOCAL LOOPBACK MODE

The local Loopback Mode is used to loopback received HCI Commands, and HCI ACL and HCI SCO packets sent from the Host.

The HC will send four Connection Complete events (one for ACL, three for SCO Connections) so that the Host can use the Connection\_Handles to re-send HCI ACL and HCI SCO Packet to HC. To exit the local Loopback Mode, HCI\_Write\_Loopback\_Mode (Loopback\_Mode=0x00) or HCI\_Reset ( ) will be used.



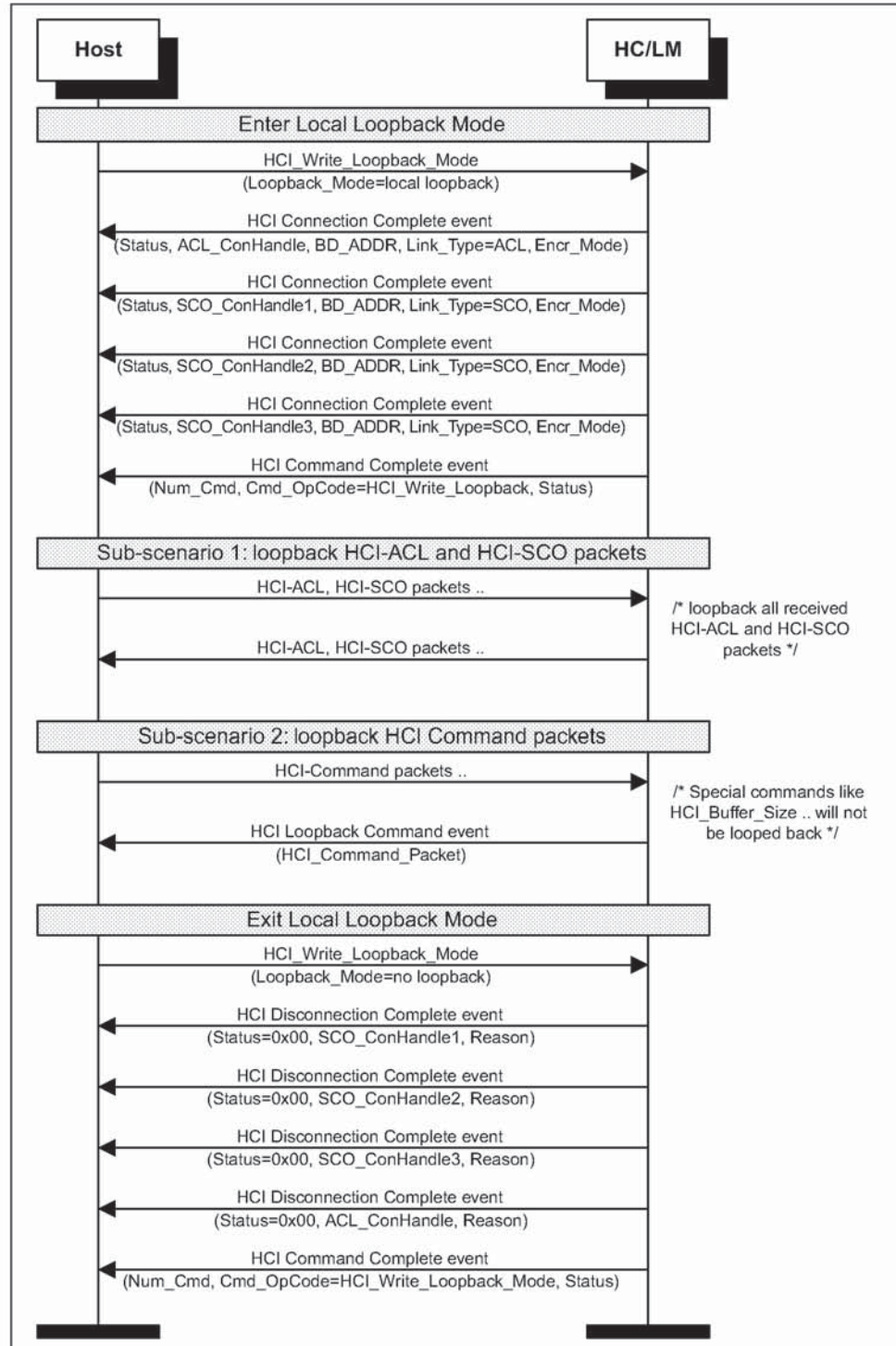


Figure 8.1: Local Loopback Mode

### 8.2 REMOTE LOOPBACK MODE

The remote Loopback Mode is used to loopback all received Baseband ACL and SCO Data received from a remote BT Device. During remote Loopback Mode, ACL and SCO Connection can be created. The remote Loopback Mode can be released with the command HCI\_Write\_Loopback\_Mode (Loopback\_Mode=0x00).

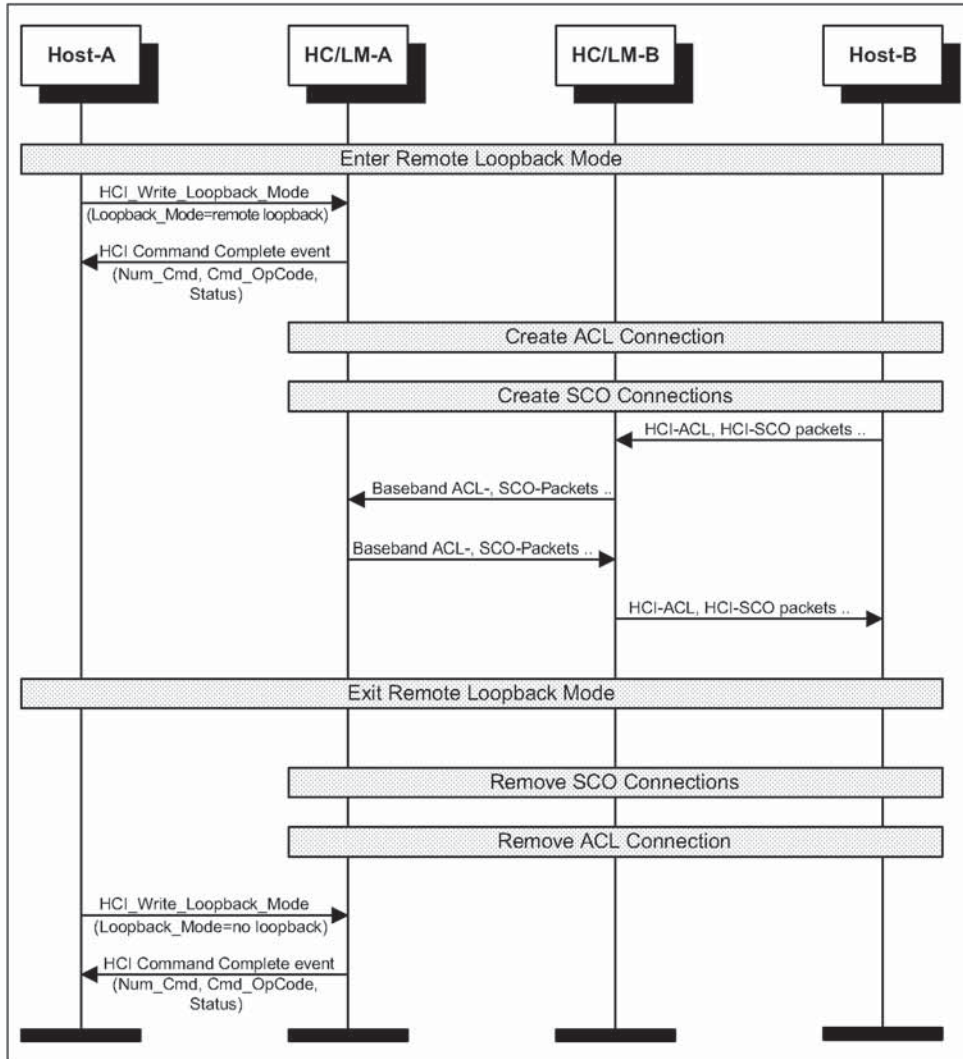


Figure 8.2: Remote Loopback Mode

## 9 LIST OF ACRONYMS AND ABBREVIATIONS

<b>BT</b>	<b>Bluetooth</b>
HC	Host Controller
HCI	Host Controller Interface
LAP	Lower Address Part
LC	Link Controller
LM	Link Manager
LMP	Link Manager Protocol
MSC	Message Sequence Chart
PDU	Protocol Data Unit

**10 LIST OF FIGURES**

Figure 2.1: Remote Name Request ..... 1039

Figure 2.2: One-Time Inquiry ..... 1040

Figure 2.3: Periodic Inquiry ..... 1041

Figure 3.1: Overview of ACL Connection establishment  
and detachment ..... 1042

Figure 3.2: ACL Connection Request phase ..... 1044

Figure 3.3: ACL Connection setup with pairing ..... 1046

Figure 3.4: ACL Connection setup with authentication ..... 1047

Figure 3.5: Encryption and Setup complete ..... 1048

Figure 3.6: ACL Disconnection ..... 1049

Figure 4.1: Authentication Requested ..... 1050

Figure 4.2: Set Connection Encryption ..... 1052

Figure 4.3: Change Connection Link Key ..... 1053

Figure 4.4: Master Link Key ..... 1054

Figure 4.5: Read Remote Supported Features ..... 1055

Figure 4.6: Read Clock Offset ..... 1056

Figure 4.7: Read Remote Version Information ..... 1056

Figure 4.8: QoS Setup ..... 1057

Figure 4.9: Switch Role ..... 1058

Figure 5.1: SCO Connection setup (activated from master) ..... 1059

Figure 5.2: SCO Connection setup (activated from slave) ..... 1060

Figure 5.3: SCO Disconnection ..... 1061

Figure 6.1: Sniff Mode ..... 1063

Figure 6.2: Hold Mode ..... 1064

Figure 6.3: Enter Park Mode ..... 1066

Figure 6.4: Exit Park Mode ..... 1067

Figure 7.1: Host to HC flow control ..... 1068

Figure 7.2: HC to Host Flow Control ..... 1069

Figure 8.1: Local Loopback Mode ..... 1071

Figure 8.2: Remote Loopback Mode ..... 1072

---

## 11 LIST OF TABLES

---

Table 6.1: Summary of modes (Sniff, Hold, Park).....1062

## 12 REFERENCES

---

- [1] "Baseband Specification" on page 33
- [2] "Link Manager Protocol" on page 185
- [3] "Host Controller Interface Functional Specification" on page 517
- [4] "Logical Link Control and Adaptation Protocol Specification" on page 245

## Alphabetical Index

### Numerics

0x7E [H:3] 784

### A

Abort- [F:2] 420  
 ACCESS RIGHTS ACCEPT [F:3] 451  
 ACCESS RIGHTS REJECT [F:3] 451  
 ACCESS RIGHTS REQUEST [F:3] 451  
 Ack Code [H:3] 781  
 Ack code [H:3] 780  
 Acknowledgement Timer (T1) [F:1] 400  
 ALERTING [F:3] 442  
 asynchronous notifications [F:4] 508  
 AT+CMUX [F:1] 399  
 authentication [C] 194

### B

basic option [F:1] 396, [F:1] 399  
 Baud Rate [H:3] 781  
 baud rate [F:1] 391, [H:3] 780  
 beacon [C] 211  
 beginning delimiter [H:3] 785  
 Bluetooth [F:2] 414  
 BOF(0x7E) [H:3] 786  
 Briefcase Trick [F:4] 500  
 business card [F:2] 415  
 byte ordering [F:1] 390  
 byte stream [F:2] 421

### C

Call Control [F:3] 435  
 CALL PROCEEDING [F:3] 441  
 Calling Line Identity [F:3] 456  
 checksum [H:3] 785  
 CL INFO [F:3] 455  
 claimant [C] 194  
 clock offset [C] 202  
 COBS [H:3] 784, [H:3] 785, [H:3] 787  
 COBS code block [H:3] 787  
 COBS code byte [H:3] 787  
 combination key [C] 197  
 commands in TS 07.10 [F:1] 396  
 Configuration distribution [F:3] 449  
 CONNECT [F:3] 442  
 Connect-request [F:2] 418  
 Consistent Overhead Byte Stuffing [H:3] 785

control channel [F:1] 396  
 convergence layer [F:1] 397  
 CRC [H:3] 782  
 CRC-CCITT [H:3] 785, [H:3] 786  
 CTS [H:3] 788  
 current link key [C] 198

### D

Data Link Connection Identifier [F:1] 393  
 data throughput [F:1] 391  
 DCE [F:1] 391  
 default URL [F:4] 509  
 delayed loopback [H:1] 812  
 Delimiter [H:3] 782  
 delimiter 0x7E [H:3] 785  
 delimiter, 0x7E [H:3] 784  
 direction bit [F:1] 400  
 DISC command [F:1] 400, [F:1] 401  
 DISC command frame [F:1] 399  
 DISCONNECT [F:3] 446  
 Disconnect-request [F:2] 419  
 DNS [F:4] 503  
 drift [C] 203  
 DTE [F:1] 391, [F:1] 399  
 DTMF ACKNOWLEDGE [F:3] 457  
 DTMF start & stop [F:3] 456  
 DTR/DSR [F:1] 403

### E

EIATIA-232-E [F:1] 389, [F:1] 391  
 eliminating zeros [H:3] 785  
 emergency call [F:3] 479  
 emulated ports [F:1] 393  
 encryption [C] 199  
 ending delimiter [H:3] 785  
 EOF(0x7E) [H:3] 786  
 Error detection [H:3] 780  
 error message packet [H:3] 785, [H:3] 789  
 Error Message Packet (0x05) [H:3] 779  
 error packet [H:3] 788  
 Error Recovery [H:3] 783  
 error recovery [H:3] 780, [H:3] 784  
 error recovery procedure [H:3] 785, [H:3] 788  
 Error Type [H:3] 786, [H:3] 789  
 ETSI TS 07.10 [F:2] 421  
 external call [F:3] 479

*Confidential Bluetooth***Bluetooth.****F**

Fast inter member access [F:3] 449  
 FCoff [F:1] 403  
 FCon [F:1] 403  
 flow control [F:1] 403  
 Forbidden Message [F:4] 501  
 frame types [F:1] 396

**G**

generator polynomial [H:3] 785  
 Get-request [F:2] 420  
 Group Management [F:3] 435

**H**

HCI RS232 Transport Layer [H:3] 778  
 header ID [F:2] 417  
 hold mode [C] 208  
 Host Controller Interface [F:4] 508  
 HTML [F:4] 505  
 HTTP [F:4] 503, [F:4] 505

**I**

in-band tones/announcements [F:3] 443  
 INFO ACCEPT [F:3] 452  
 INFO SUGGEST [F:3] 452  
 INFORMATION [F:3] 441  
 initialisation key [C] 195  
 intercom call [F:3] 479  
 Internet Engineering Task Force (IETF) [F:4] 504  
 interoperability [F:4] 511  
 interrupt latency [H:3] 780  
 IrCOMM [F:2] 416  
 IrDA [F:2] 414  
 IrMC [F:2] 425  
 IrOBEX [F:2] 414

**J**

JavaScript [F:4] 505  
 jitter [C] 203

**L**

L2CAP channel [F:1] 407  
 latency requirements [F:1] 407  
 link key [C] 194  
 link loss notification [F:1] 399, [F:1] 407  
 link supervision [C] 224  
 LISTEN REJECT [F:3] 454  
 LISTEN REQUEST [F:3] 453  
 LISTEN SUGGEST [F:3] 453

loop back test [F:1] 815  
 low power mode [F:1] 407

**M**

Management Entity [F:4] 508  
 Maximum Frame Size (N1) [F:1] 400  
 Modem Status Command [F:1] 397  
 multiple bearers [F:4] 508  
 multiplexer control channel [F:1] 399  
 Multiplexer Control commands [F:1] 401

**N**

name request [C] 207  
 negotiation packet [H:3] 780, [H:3] 781  
 Negotiation Packet (0x06) [H:3] 779  
 negotiation phase [H:3] 780  
 null modem [F:1] 392  
 null modem emulation [F:1] 391  
 number of data bit [H:3] 780  
 number of stop bit [H:3] 780

**O**

OBEX [F:2] 414  
 OBEX session protocol [F:2] 417  
 Obtain access rights [F:3] 449  
 output power [C] 215

**P**

paging scheme [C] 223  
 pairing [C] 195  
 parity type [H:3] 780  
 park mode [C] 211  
 payload header [C] 192  
 PIN [C] 195  
 PN command [F:1] 402  
 port emulation entity [F:1] 405  
 port proxy entity [F:1] 405  
 Protocol Mode [H:3] 782  
 protocol mode [H:3] 780  
 protocol mode 0x13 [H:3] 785  
 protocol mode 0x14 [H:3] 788  
 Proxy/gateway Addressing [F:4] 509  
 Put-request [F:2] 419

**Q**

Q.931 [F:3] 435  
 Quality of Service [C] 218

**R**

register recall [F:3] 456



**Bluetooth.**

RELEASE [F:3] 446  
 RELEASE COMPLETE [F:3] 446  
 reliability [F:1] 407  
 reliable transmission [F:1] 400  
 Response Timer for Multiplexer Control Channel (T2) [F:1] 400  
 resynchronization [H:3] 784  
 resynchronize [H:3] 788  
 retransmission holding buffer [H:3] 785, [H:3] 788  
 retransmission packets [H:3] 785, [H:3] 788  
 RFCOMM [F:2] 414  
 RFCOMM entity [F:1] 393  
 RFCOMM multiplexer [F:1] 399  
 RFCOMM reference model [F:1] 395  
 RFCOMM Server Channel [F:1] 405  
 RFCOMM server channels [F:1] 393, [F:1] 400  
 RFCOMM session [F:1] 393, [F:1] 399  
 RLS command [F:1] 402  
 RPN command [F:1] 401  
 RS-232 [F:1] 389, [F:1] 391, [F:1] 405  
 RS232 [H:3] 778  
 RS-232 control signals [F:1] 392, [F:1] 397  
 RS232 Transport Packet [H:3] 779  
 RSSI [C] 215  
 RTS [H:3] 788  
 RTS/CTS [F:1] 403, [H:3] 780  
 RTS/CTS Mode [H:3] 782

**S**

SABM command [F:1] 399, [F:1] 400  
 SCO link [C] 219  
 semi-permanent link key [C] 198, [C] 199  
 SEQ No with Error [H:3] 785  
 sequence number [H:3] 779  
 sequence number field [H:3] 785, [H:3] 788  
 Sequence Number with Error field [H:3] 785, [H:3] 788  
 serial port emulation entity [F:1] 395  
 service call [F:3] 479  
 Service Discovery Protocol [F:4] 506, [F:4] 512  
 service records [F:1] 405  
 SetPath- [F:2] 420  
 SETUP [F:3] 439  
 SETUP ACKNOWLEDGE [F:3] 441  
 simple error recovery scheme [H:3] 788  
 Smart Kiosk [F:4] 501  
 sniff mode [C] 209  
 SSL [F:4] 505  
 START DTMF [F:3] 457

START DTMF REJECT [F:3] 457  
 STOP DTMF [F:3] 457  
 STOP DTMF ACKNOWLEDGE [F:3] 457  
 supervision timeout [C] 224  
 synchronization [H:3] 785  
 synchronize [H:3] 788

**T**

TCP [F:4] 505  
 TCP port number [F:2] 423  
 TCP/IP [F:2] 414  
 TCS Binary [F:3] 435  
 Tdetect [H:3] 780  
 Tdetect Time [H:3] 782  
 Tdetect time [H:3] 788  
 temporary link key [C] 198  
 test mode [C] 237, [I:1] 806  
 Tiny TP [F:2] 416  
 transmitter test [I:1] 811  
 TS 07.10 [F:1] 389  
 TS 07.10 multiplexer [F:1] 393, [F:1] 407

**U**

UART [H:3] 780  
 UART Settings [H:3] 781  
 UDP [F:4] 504  
 Uniform Resource Locators [F:4] 509  
 unit key [C] 197  
 URL [F:4] 507  
 User Addressing [F:4] 509

**V**

vCalendar [F:2] 415  
 vCard [F:2] 415  
 verifier [C] 194  
 vMessage [F:2] 415  
 vNotes [F:2] 415

**W**

WAP Client [F:4] 502  
 WAP Proxy/gateway [F:4] 503  
 WAP Server [F:4] 503  
 WDP [F:4] 504  
 Wireless User Group [F:3] 449  
 WSP [F:4] 504  
 WTLS [F:4] 504  
 WTP [F:4] 504  
 WUG [F:3] 449

*Confidential Bluetooth*

**Bluetooth.**

**X**

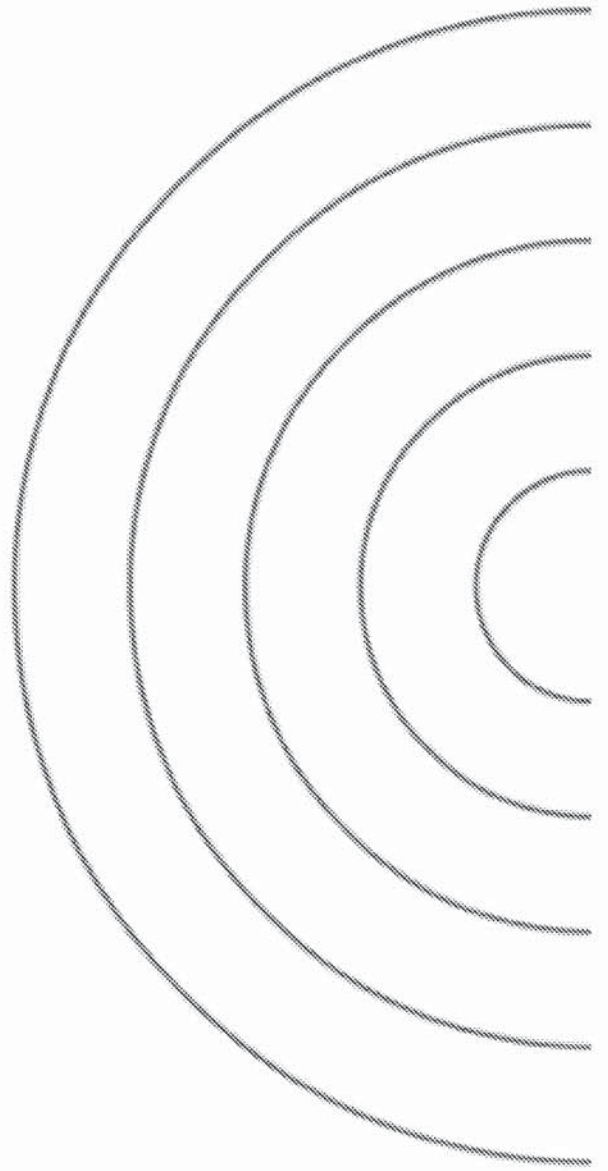
XML [F:4] 505

XON/XOFF [F:1] 403

**Z**

zero elimination [H:3] 787





# Specification of the Bluetooth System

Wireless connections made easy

---

## Profiles

**Bluetooth™**

v1.0 B  
December 1st 1999



<b>BLUETOOTH DOC</b>	Date / Day-Month-Year 01 Dec 99	N.B.	Document No. <b>1.C.47/1.0 B</b>
Responsible	e-mail address		Status

**Bluetooth.**

---

# Profiles of the Bluetooth System

Version 1.0B

## Revision History

The Revision History is shown in Appendix I on page 413

## Contributors

The persons who contributed to this specification are listed in Appendix II on page 421.

## Web Site

This specification can also be found on the Bluetooth web site:  
<http://www.bluetooth.com>

## Disclaimer and copyright notice

THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

Copyright © 1999

Telefonaktiebolaget LM Ericsson,  
International Business Machines Corporation,  
Intel Corporation,  
Nokia Corporation,  
Toshiba Corporation .

\*Third-party brands and names are the property of their respective owners.



**MASTER TABLE OF CONTENTS**

**For the Core Specification, see Volume 1**

**Part K:1**

**GENERIC ACCESS PROFILE**

Contents .....	15
Foreword .....	19
1 Introduction .....	20
2 Profile Overview .....	22
3 User Interface Aspects .....	25
4 Modes .....	29
5 Security Aspects .....	33
6 Idle Mode Procedures .....	37
7 Establishment Procedures .....	45
8 Definitions .....	52
9 Annex A (Normative): Timers and Constants .....	56
10 Annex B (Informative): Information Flows of Related Procedures .....	57
11 References .....	60

**Part K:2**

**SERVICE DISCOVERY APPLICATION PROFILE**

Contents .....	63
Foreword .....	65
1 Introduction .....	66
2 Profile Overview .....	68
3 User Interface Aspects .....	72
4 Application Layer .....	73
5 Service Discovery .....	79
6 L2CAP .....	82
7 Link Manager .....	86
8 Link Control .....	88
9 References .....	91
10 Definitions .....	92
11 Appendix A (Informative): Service Primitives and the Bluetooth PDUs .....	93



**Part K:3**

**CORDLESS TELEPHONY PROFILE**

<b>Contents</b> .....	<b>97</b>
1 Introduction .....	100
2 Profile Overview .....	103
3 Application Layer .....	108
4 TCS-BIN Procedures .....	110
5 Service Discovery Procedures .....	120
6 L2CAP Procedures .....	121
7 LMP Procedures Overview .....	122
8 LC Features .....	124
9 General Access Profile Interoperability Requirements .....	126
10 Annex A (Informative): Signalling Flows .....	128
11 Timers and Counters .....	135
12 References .....	136
13 List of Figures .....	137
14 List of Tables .....	138

**Part K:4**

**INTERCOM PROFILE**

<b>Contents</b> .....	<b>141</b>
1 Introduction .....	143
2 Profile Overview .....	145
3 Application Layer .....	148
4 TCS Binary .....	149
5 SDP Interoperability Requirements .....	153
6 L2CAP Interoperability Requirements .....	154
7 Link Manager (LM) Interoperability Requirements .....	155
8 Link Control (LC) Interoperability Requirements .....	156
9 Generic Access Profile .....	158
10 Annex A (Informative): Signalling flows .....	159
11 Timers and Counters .....	161
12 List of Figures .....	162
13 List of Tables .....	163

---

**Part K:5**

---

**SERIAL PORT PROFILE**

<b>Contents .....</b>	<b>167</b>
Foreword .....	169
1 Introduction .....	170
2 Profile Overview .....	171
3 Application Layer .....	174
4 RFCOMM Interoperability Requirements .....	177
5 L2CAP Interoperability Requirements .....	179
6 SDP Interoperability Requirements .....	181
7 Link Manager (LM) Interoperability Requirements .....	183
8 Link Control (LC) Interoperability Requirements .....	184
9 References .....	186
10 List of Figures .....	187
11 List of Tables .....	188

---

**Part K:6**

---

**HEADSET PROFILE**

<b>Contents .....</b>	<b>191</b>
1 Introduction .....	193
2 Profile Overview .....	196
3 Application Layer .....	200
4 Headset Control Interoperability Requirements .....	201
5 Serial Port Profile .....	210
6 Generic Access Profile .....	214
7 References .....	215
8 List of Figures .....	216
9 List of Tables .....	217




---

**Part K:7**

---

**DIAL-UP NETWORKING PROFILE**

<b>Contents</b> .....	<b>221</b>
1 Introduction .....	223
2 Profile Overview .....	226
3 Application Layer .....	230
4 Dialling and Control Interoperability Requirements .....	231
5 Serial Port Profile Interoperability Requirements .....	235
6 Generic Access Profile Interoperability Requirements .....	238
7 References .....	240
8 List of Figures .....	241
9 List of Tables .....	242

---

**Part K:8**

---

**FAX PROFILE**

<b>Contents</b> .....	<b>245</b>
1 Introduction .....	246
2 Profile Overview .....	249
3 Application Layer .....	253
4 Dialling and Control Interoperability Requirements .....	254
5 Serial Port Profile .....	256
6 Generic Access Profile Interoperability Requirements .....	259
7 References .....	261
8 List of Figures .....	262
9 List of Tables .....	263

**Part K:9****LAN ACCESS PROFILE**

<b>Contents .....</b>	<b>267</b>
1 Introduction .....	269
2 Profile Overview .....	271
3 User Interface Aspects .....	275
4 Application Layer .....	278
5 PPP .....	281
6 RFCOMM .....	284
7 Service Discovery .....	285
8 L2CAP .....	287
9 Link Manager .....	288
10 Link Control .....	290
11 Management Entity Procedures .....	291
12 APPENDIX A (Normative): Timers and counters .....	293
13 APPENDIX B (Normative): Microsoft Windows .....	294
14 APPENDIX C (Informative): Internet Protocol (IP) .....	295
15 List of Figures .....	297
16 List of Tables .....	298
17 References .....	299

**Part K:10****GENERIC OBJECT EXCHANGE PROFILE**

<b>Contents .....</b>	<b>303</b>
Foreword .....	305
1 Introduction .....	306
2 Profile Overview .....	310
3 User Interface Aspects .....	312
4 Application Layer .....	313
5 OBEX Interoperability Requirements .....	314
6 Serial Port Profile Interoperability Requirements .....	324
7 Generic Access Profile Interoperability Requirements .....	326
8 References .....	328

---

**Part K:11**

---

**OBJECT PUSH PROFILE**

<b>Contents .....</b>	<b>331</b>
Foreword .....	333
1 Introduction .....	334
2 Profile Overview.....	338
3 User Interface Aspects.....	340
4 Application Layer .....	344
5 OBEX.....	348
6 Service Discovery .....	351
7 References .....	353

---

**Part K:12**

---

**FILE TRANSFER PROFILE**

<b>Contents .....</b>	<b>357</b>
Foreword .....	359
1 Introduction .....	360
2 Profile Overview.....	364
3 User Interface Aspects.....	367
4 Application Layer .....	370
5 OBEX.....	374
6 Service Discovery .....	383
7 References .....	385

---

**Part K:13**

---

**SYNCHRONIZATION PROFILE**

<b>Contents .....</b>	<b>389</b>
Foreword .....	391
1 Introduction .....	392
2 Profile Overview.....	396
3 User Interface Aspects.....	399
4 Application Layer .....	402
5 IrMC Synchronization Requirements .....	404
6 OBEX.....	406
7 Service Discovery .....	408
8 References .....	411

---

**Bluetooth.**

---

**Appendix I**

---

REVISION HISTORY .....	413
------------------------	-----

---

**Appendix II**

---

CONTRIBUTORS .....	421
--------------------	-----

---

**Appendix III**

---

ACRONYMS AND ABBREVIATIONS .....	429
----------------------------------	-----

---

<b>INDEX</b>	<b>435</b>
--------------	------------

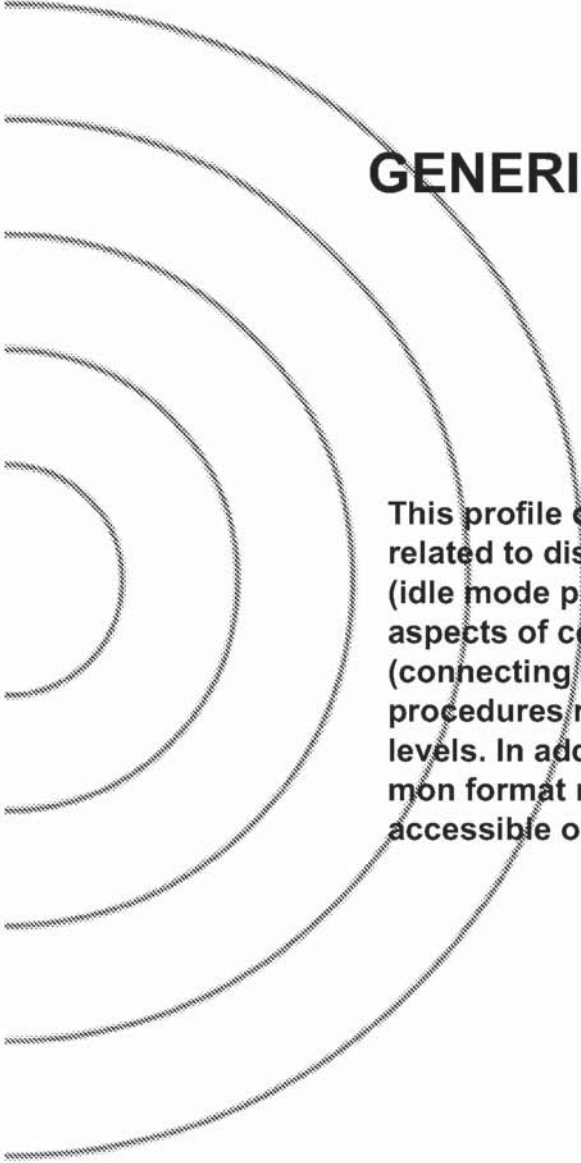
---





## Part K:1

# GENERIC ACCESS PROFILE



This profile defines the generic procedures related to discovery of Bluetooth devices (idle mode procedures) and link management aspects of connecting to Bluetooth devices (connecting mode procedures). It also defines procedures related to use of different security levels. In addition, this profile includes common format requirements for parameters accessible on the user interface level.



**CONTENTS**

<b>1</b>	<b>Introduction .....</b>	<b>20</b>
1.1	Scope .....	20
1.2	Symbols and conventions .....	20
1.2.1	Requirement status symbols .....	20
1.2.2	Signalling diagram conventions.....	21
1.2.3	Notation for timers and counters .....	21
<b>2</b>	<b>Profile overview.....</b>	<b>22</b>
2.1	Profile stack .....	22
2.2	Configurations and roles .....	22
2.3	User requirements and scenarios .....	23
2.4	Profile fundamentals .....	23
2.5	Conformance .....	24
<b>3</b>	<b>User interface aspects.....</b>	<b>25</b>
3.1	The user interface level.....	25
3.2	Representation of Bluetooth parameters .....	25
3.2.1	Bluetooth device address (BD_ADDR) .....	25
3.2.1.1	Definition.....	25
3.2.1.2	Term on user interface level.....	25
3.2.1.3	Representation .....	25
3.2.2	Bluetooth device name (the user-friendly name).....	25
3.2.2.1	Definition.....	25
3.2.2.2	Term on user interface level.....	26
3.2.2.3	Representation .....	26
3.2.3	Bluetooth passkey (Bluetooth PIN) .....	26
3.2.3.1	Definition.....	26
3.2.3.2	Terms at user interface level.....	26
3.2.3.3	Representation .....	26
3.2.4	Class of Device .....	27
3.2.4.1	Definition.....	27
3.2.4.2	Term on user interface level.....	27
3.2.4.3	Representation .....	27
3.3	Pairing.....	28

<b>4</b>	<b>Modes</b> .....	<b>29</b>
4.1	Discoverability modes.....	29
4.1.1	Non-discoverable mode.....	30
4.1.1.1	Definition.....	30
4.1.1.2	Term on UI-level.....	30
4.1.2	Limited discoverable mode.....	30
4.1.2.1	Definition.....	30
4.1.2.2	Conditions.....	31
4.1.2.3	Term on UI-level.....	31
4.1.3	General discoverable mode.....	31
4.1.3.1	Definition.....	31
4.1.3.2	Conditions.....	31
4.1.3.3	Term on UI-level.....	31
4.2	Connectability modes.....	31
4.2.1	Non-connectable mode.....	31
4.2.1.1	Definition.....	31
4.2.1.2	Term on UI-level.....	32
4.2.2	Connectable mode.....	32
4.2.2.1	Definition.....	32
4.2.2.2	Term on UI-level.....	32
4.3	Pairing modes.....	32
4.3.1	Non-pairable mode.....	32
4.3.1.1	Definition.....	32
4.3.1.2	Term on UI-level.....	32
4.3.2	Pairable mode.....	32
4.3.2.1	Definition.....	32
4.3.2.2	Term on UI-level.....	32
<b>5</b>	<b>Security aspects</b> .....	<b>33</b>
5.1	Authentication.....	33
5.1.1	Purpose.....	33
5.1.2	Term on UI level.....	33
5.1.3	Procedure.....	34
5.1.4	Conditions.....	34
5.2	Security modes.....	34
5.2.1	Security mode 1 (non-secure).....	36
5.2.2	Security mode 2 (service level enforced security).....	36
5.2.3	Security modes 3 (link level enforced security).....	36

6	Idle mode procedures .....	37
6.1	General inquiry .....	37
6.1.1	Purpose .....	37
6.1.2	Term on UI level .....	37
6.1.3	Description .....	38
6.1.4	Conditions .....	38
6.2	Limited inquiry .....	38
6.2.1	Purpose .....	38
6.2.2	Term on UI level .....	39
6.2.3	Description .....	39
6.2.4	Conditions .....	39
6.3	Name discovery .....	40
6.3.1	Purpose .....	40
6.3.2	Term on UI level .....	40
6.3.3	Description .....	40
	6.3.3.1 Name request .....	40
	6.3.3.2 Name discovery .....	40
6.3.4	Conditions .....	41
6.4	Device discovery .....	41
6.4.1	Purpose .....	41
6.4.2	Term on UI level .....	41
6.4.3	Description .....	42
6.4.4	Conditions .....	42
6.5	Bonding .....	42
6.5.1	Purpose .....	42
6.5.2	Term on UI level .....	42
6.5.3	Description .....	43
	6.5.3.1 General bonding .....	43
	6.5.3.2 Dedicated bonding .....	44
6.5.4	Conditions .....	44

<b>7</b>	<b>Establishment procedures</b> .....	<b>45</b>
7.1	Link establishment.....	45
7.1.1	Purpose.....	45
7.1.2	Term on UI level.....	45
7.1.3	Description.....	46
	7.1.3.1 B in security mode 1 or 2.....	46
	7.1.3.2 B in security mode 3.....	47
7.1.4	Conditions.....	47
7.2	Channel establishment.....	48
7.2.1	Purpose.....	48
7.2.2	Term on UI level.....	48
7.2.3	Description.....	48
	7.2.3.1 B in security mode 2.....	49
	7.2.3.2 B in security mode 1 or 3.....	49
7.2.4	Conditions.....	49
7.3	Connection establishment.....	50
7.3.1	Purpose.....	50
7.3.2	Term on UI level.....	50
7.3.3	Description.....	50
	7.3.3.1 B in security mode 2.....	50
	7.3.3.2 B in security mode 1 or 3.....	51
7.3.4	Conditions.....	51
7.4	Establishment of additional connection.....	51
<b>8</b>	<b>Definitions</b> .....	<b>52</b>
8.1	General definitions.....	52
8.2	Connection-related definitions.....	52
8.3	Device-related definitions.....	53
8.4	Procedure-related definitions.....	54
8.5	Security-related definitions.....	54
<b>9</b>	<b>Annex A (Normative): Timers and constants</b> .....	<b>56</b>
<b>10</b>	<b>Annex B (Informative): Information flows of related procedures</b> ..	<b>57</b>
10.1	Imp-authentication.....	57
10.2	Imp-pairing.....	58
10.3	Service discovery.....	58
<b>11</b>	<b>References</b> .....	<b>60</b>

## FOREWORD

---

Interoperability between devices from different manufacturers is provided for a specific service and use case, if the devices conform to a Bluetooth SIG-defined profile specification. A profile defines a selection of messages and procedures (generally termed *capabilities*) from the Bluetooth SIG specifications and gives an unambiguous description of the air interface for specified service(s) and use case(s).

All defined features are process-mandatory. This means that, if a feature is used, it is used in a specified manner. Whether the provision of a feature is mandatory or optional is stated separately for both sides of the Bluetooth air interface.

## 1 INTRODUCTION

---

### 1.1 SCOPE

The purpose of the Generic Access Profile is:

To introduce definitions, recommendations and common requirements related to modes and access procedures that are to be used by transport and application profiles.

To describe how devices are to behave in standby and connecting states in order to guarantee that links and channels always can be established between Bluetooth devices, and that multi-profile operation is possible. Special focus is put on discovery, link establishment and security procedures.

To state requirements on user interface aspects, mainly coding schemes and names of procedures and parameters, that are needed to guarantee a satisfactory user experience.

### 1.2 SYMBOLS AND CONVENTIONS

#### 1.2.1 Requirement status symbols

In this document (especially in the profile requirements tables), the following symbols are used:

'M' for mandatory to support (used for capabilities that shall be used in the profile);

'O' for optional to support (used for capabilities that can be used in the profile);

'C' for conditional support (used for capabilities that shall be used in case a certain other capability is supported);

'X' for excluded (used for capabilities that may be supported by the unit but shall never be used in the profile);

'N/A' for not applicable (in the given context it is impossible to use this capability).

Some excluded capabilities are capabilities that, according to the relevant Bluetooth specification, are mandatory. These are features that may degrade operation of devices following this profile. Therefore, these features shall never be activated while a unit is operating as a unit within this profile.

In this specification, the word *shall* is used for mandatory requirements, the word *should* is used to express recommendations and the word *may* is used for options.



**1.2.2 Signalling diagram conventions**

The following arrows are used in diagrams describing procedures :

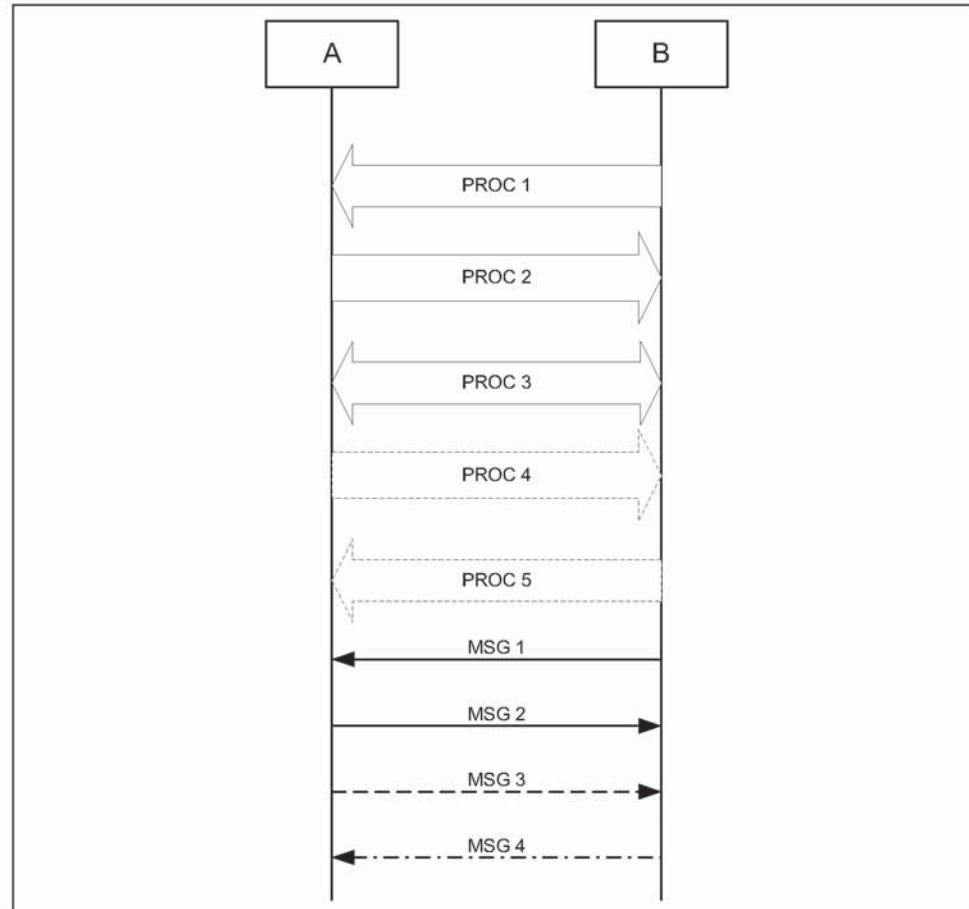


Figure 1.1: Arrows used in signalling diagrams

In the table above, the following cases are shown: PROC1 is a sub-procedure initiated by B. PROC2 is a sub-procedure initiated by A. PROC3 is a sub-procedure where the initiating side is undefined (may be both A or B). Dashed arrows denote optional steps. PROC4 indicates an optional sub-procedure initiated by A, and PROC5 indicates an optional sub-procedure initiated by B.

MSG1 is a message sent from B to A. MSG2 is a message sent from A to B. MSG3 indicates an optional message from A to B, and MSG4 indicates a conditional message from B to A.

**1.2.3 Notation for timers and counters**

Timers are introduced specific to this profile. To distinguish them from timers used in the Bluetooth protocol specifications and other profiles, these timers are named in the following format: 'T<sub>GAP</sub>(*nnn*)'.

## 2 PROFILE OVERVIEW

### 2.1 PROFILE STACK

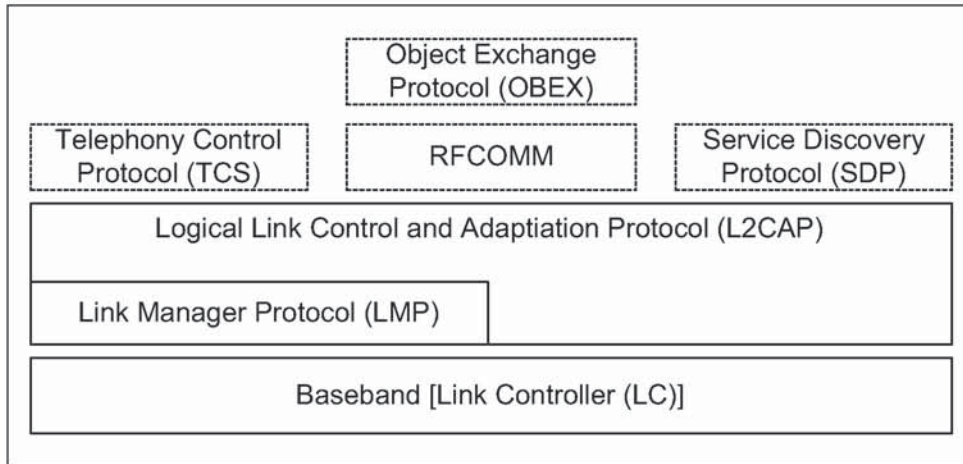


Figure 2.1: Profile stack covered by this profile.

The main purpose of this profile is to describe the use of the lower layers of the Bluetooth protocol stack (LC and LMP). To describe security related alternatives, also higher layers (L2CAP, RFCOMM and OBEX) are included.

### 2.2 CONFIGURATIONS AND ROLES

For the descriptions in this profile of the roles that the two devices involved in a Bluetooth communication can take, the generic notation of the A-party (the *paging device* in case of link establishment, or *initiator* in case of another procedure on an established link) and the B-party (*paged device* or *acceptor*) is used. The A-party is the one that, for a given procedure, initiates the establishment of the physical link or initiates a transaction on an existing link.

This profile handles the procedures between two devices related to discovery and connecting (link and connection establishment) for the case where none of the two devices has any link established as well as the case where (at least) one device has a link established (possibly to a third device) before starting the described procedure.

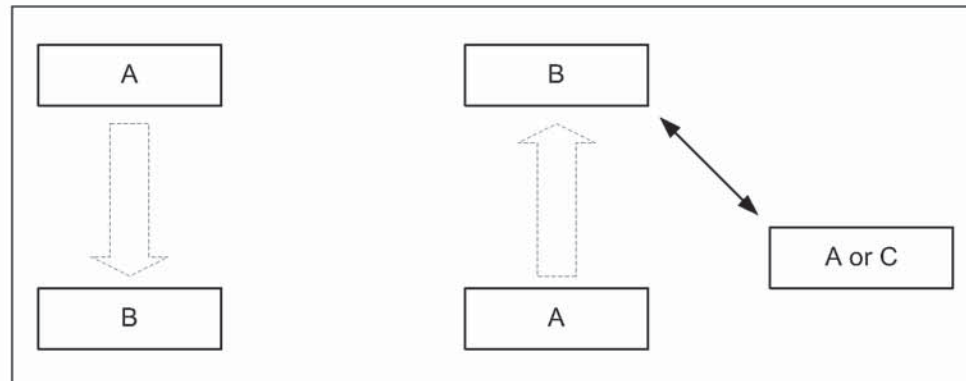


Figure 2.2: This profile covers procedures initiated by one device (A) towards another device (B) that may or may not have an existing Bluetooth link active.

The initiator and the acceptor generally operate the generic procedures according to this profile or another profile referring to this profile. If the acceptor operates according to several profiles simultaneously, this profile describes generic mechanisms for how to handle this.

### 2.3 USER REQUIREMENTS AND SCENARIOS

The Bluetooth user should in principle be able to connect a Bluetooth device to any other Bluetooth device. Even if the two connected devices don't share any common application, it should be possible for the user to find this out using basic Bluetooth capabilities. When the two devices do share the same application but are from different manufacturers, the ability to connect them should not be blocked just because manufacturers choose to call basic Bluetooth capabilities by different names on the user interface level or implement basic procedures to be executed in different orders.

### 2.4 PROFILE FUNDAMENTALS

This profile states the requirements on names, values and coding schemes used for names of parameters and procedures experienced on the user interface level.

This profile defines modes of operation that are not service- or profile-specific, but that are generic and can be used by profiles referring to this profile, and by devices implementing multiple profiles.

This profile defines the general procedures that can be used for discovering identities, names and basic capabilities of other Bluetooth devices that are in a mode where they can be discoverable. Only procedures where no channel or connection establishment is used are specified.

This profile defines the general procedure for how to create bonds (i.e. dedicated exchange of link keys) between Bluetooth devices.

This profile describes the general procedures that can be used for establishing connections to other Bluetooth devices that are in mode that allows them to accept connections and service requests.

## 2.5 CONFORMANCE

Bluetooth devices that do not conform to any other Bluetooth profile shall conform to this profile to ensure basic interoperability and co-existence.

Bluetooth devices that conform to another Bluetooth profile may use adaptations of the generic procedures as specified by that other profile. They shall, however, be compatible with devices compliant to this profile at least on the level of the supported generic procedures.

If conformance to this profile is claimed, all capabilities indicated mandatory for this profile shall be supported in the specified manner (process-mandatory). This also applies for all optional and conditional capabilities for which support is indicated. All mandatory capabilities, and optional and conditional capabilities for which support is indicated, are subject to verification as part of the Bluetooth certification program.

## 3 USER INTERFACE ASPECTS

---

### 3.1 THE USER INTERFACE LEVEL

In the context of this specification, the user interface level refers to places (such as displays, dialog boxes, manuals, packaging, advertising, etc.) where users of Bluetooth devices encounters names, values and numerical representation of Bluetooth terminology and parameters.

This profile specifies the generic terms that should be used on the user interface level. These terms should be translated into languages supported by the Bluetooth device according to tables provided by the Bluetooth SIG.

### 3.2 REPRESENTATION OF BLUETOOTH PARAMETERS

#### 3.2.1 Bluetooth device address (BD\_ADDR)

##### 3.2.1.1 Definition

BD\_ADDR is the unique address of a Bluetooth device as defined in [1]. It is received from a remote device during the device discovery procedure.

##### 3.2.1.2 Term on user interface level

When the Bluetooth address is referred to on UI level, the term 'Bluetooth Device Address' should be used.

##### 3.2.1.3 Representation

On BB level the BD\_ADDR is represented as 48 bits [1].

On the UI level the Bluetooth address shall be represented as 12 hexadecimal characters, possibly divided into sub-parts separated by ':'. (E.g., '000C3E3A4B69' or '00:0C:3E:3A:4B:69'.) At UI level, any number shall have the MSB -> LSB (from left to right) 'natural' ordering (e.g., the number '16' shall be shown as '0x10').

#### 3.2.2 Bluetooth device name (the user-friendly name)

##### 3.2.2.1 Definition

The Bluetooth device name is the user-friendly name that a Bluetooth device presents itself with. It is a character string returned in LMP\_name\_res as response to a LMP\_name\_req.

### 3.2.2.2 Term on user interface level

When the Bluetooth device name is referred to on UI level, the term 'Bluetooth Device Name' should be used.

### 3.2.2.3 Representation

The Bluetooth device name can be up to 248 bytes maximum according to [2]. It shall be coded according to Unicode UTF-8 (i.e. name entered on UI level may be down to 82 characters if UCS-2 is used).

A device can not expect that a general remote device is able to handle more than the first 40 characters of the Bluetooth device name. If a remote device has limited display capabilities, it may use only the first 20 characters.

## **3.2.3 Bluetooth passkey (Bluetooth PIN)**

### 3.2.3.1 Definition

The Bluetooth PIN is used to authenticate two Bluetooth devices (that have not previously exchanged link keys) to each other and create a trusted relationship between them. The PIN is used in the pairing procedure (see Section 10.2) to generate the initial link key that is used for further authentication.

The PIN may be entered on UI level but may also be stored in the device; e.g. in the case of a device without sufficient MMI for entering and displaying digits.

### 3.2.3.2 Terms at user interface level

When the Bluetooth PIN is referred to on UI level, the term 'Bluetooth Passkey' should be used.

### 3.2.3.3 Representation

The Bluetooth PIN has different representations on different level.  $PIN_{BB}$  is used on baseband level, and  $PIN_{UI}$  is used on user interface level.

$PIN_{BB}$  is the PIN used by [1] for calculating the initialization key during the pairing procedure.  $PIN_{UI}$  is the character representation of the PIN that is entered on UI level. The transformation between  $PIN_{BB}$  and  $PIN_{UI}$  shall be according to Unicode UTF-8.

According to [1],  $PIN_{BB}$  can be 128 bits (16 bytes). When PIN is entered on UI level ( $PIN_{UI}$ ), it is to be coded into  $PIN_{BB}$  according to Unicode UTF-8 (i.e. if a

device supports entry of characters outside the Unicode range 0x00 - 0x7F, the maximum number of characters in the PIN<sub>UI</sub> may be less than 16).

**Examples:**

User-entered code	Corresponding PIN <sub>BB</sub> [0..length-1] (value as a sequence of octets in hexadecimal notation)
'0123'	length = 4, value = 0x30 0x31 0x32 0x33
'Ärlich'	length = 7, value = 0xC3 0x84 0x72 0x6C 0x69 0x63 0x68

All Bluetooth devices that support the bonding procedure and support PIN handling on UI level shall support UI level handling of PINs consisting of decimal digits. In addition, devices may support UI level handling of PINs consisting of general characters.

If a device has a fixed PIN (i.e. PIN is stored in the device and cannot be entered on UI level during pairing), the PIN shall be defined using decimal digits. A device that is expected to pair with a remote device that has restricted UI capabilities should ensure that the PIN can be entered on UI level as decimal digits.

**3.2.4 Class of Device**

3.2.4.1 Definition

Class of device is a parameter received during the device discovery procedure, indicating the type of device and which types of service that are supported.

3.2.4.2 Term on user interface level

The information within the Class of Device parameter should be referred to as 'Bluetooth Device Class' (i.e. the major and minor device class fields) and 'Bluetooth Service Type' (i.e. the service class field). The terms for the defined Bluetooth Device Types and Bluetooth Service Types are defined in [11].

When using a mix of information found in the Bluetooth Device Class and the Bluetooth Service Type, the term 'Bluetooth Device Type' should be used.

3.2.4.3 Representation

The Class of device is a bit field and is defined in [11]. The UI-level representation of the information in the Class of device is implementation specific.

### 3.3 PAIRING

Two procedures are defined that make use of the pairing procedure defined on LMP level (LMP-pairing, see Section 10.2). Either the user initiates the bonding procedure and enters the passkey with the explicit purpose of creating a bond (and maybe also a secure relationship) between two Bluetooth devices, or the user is requested to enter the passkey during the establishment procedure since the devices did not share a common link key beforehand. In the first case, the user is said to perform 'bonding (with entering of passkey)' and in the second case the user is said to 'authenticate using the passkey'.



## 4 MODES

	Procedure	Ref.	Support
1	Discoverability modes	4.1	
	Non-discoverable mode		C1
	Limited discoverable mode		C2
	General discoverable mode		C2
2	Connectability modes	4.1.3.3	
	Non-connectable mode		O
	Connectable mode		M
3	Pairing modes	4.2.2.2	
	Non-pairable mode		O
	Pairable mode		C3
C1: If limited discoverable mode is supported, non-discoverable mode is mandatory, otherwise optional.			
C2: A Bluetooth device shall support at least one discoverable mode (limited or/and general).			
C3: If the bonding procedure is supported, support for pairable mode is mandatory, otherwise optional.			

Table 4.1: Conformance requirements related to modes defined in this section

### 4.1 DISCOVERABILITY MODES

With respect to inquiry, a Bluetooth device shall be either in non-discoverable mode or in a discoverable mode. (The device shall be in one, and only one, discoverability mode at a time.) The two discoverable modes defined here are called limited discoverable mode and general discoverable mode. Inquiry is defined in [1].

When a Bluetooth device is in non-discoverable mode it does not respond to inquiry.

A Bluetooth device is said to be made discoverable, or set into a discoverable mode, when it is in limited discoverable mode or in general discoverable mode. Even when a Bluetooth device is made discoverable it may be unable to respond to inquiry due to other baseband activity [1]. A Bluetooth device that does not respond to inquiry for any of these two reasons is called a silent device.

After being made discoverable, the Bluetooth device shall be discoverable for at least  $T_{GAP}(103)$ .

#### **4.1.1 Non-discoverable mode**

##### 4.1.1.1 Definition

When a Bluetooth device is in non-discoverable mode, it shall never enter the INQUIRY\_RESPONSE state.

##### 4.1.1.2 Term on UI-level

Bluetooth device is 'non-discoverable' or in 'non-discoverable mode'.

#### **4.1.2 Limited discoverable mode**

##### 4.1.2.1 Definition

The limited discoverable mode should be used by devices that need to be discoverable only for a limited period of time, during temporary conditions or for a specific event. The purpose is to respond to a device that makes a limited inquiry (inquiry using the LIAC).

A Bluetooth device should not be in limited discoverable mode for more than  $T_{GAP}(104)$ . The scanning for the limited inquiry access code can be done either in parallel or in sequence with the scanning of the general inquiry access code. When in limited discoverable mode, one of the following options shall be used.

##### 4.1.2.1.1 Parallel scanning

When a Bluetooth device is in limited discoverable mode, it shall enter the INQUIRY\_SCAN state at least once in  $T_{GAP}(102)$  and scan for the GIAC and the LIAC for at least  $T_{GAP}(101)$ .

##### 4.1.2.1.2 Sequential scanning

When a Bluetooth device is in limited discoverable mode, it shall enter the INQUIRY\_SCAN state at least once in  $T_{GAP}(102)$  and scan for the GIAC for at least  $T_{GAP}(101)$  and enter the INQUIRY\_SCAN state more often than once in  $T_{GAP}(102)$  and scan for the LIAC for at least  $T_{GAP}(101)$ .

If an inquiry message is received when in limited discoverable mode, the entry into the INQUIRY\_RESPONSE state takes precedence over the next entries into INQUIRY\_SCAN state until the inquiry response is completed.

#### 4.1.2.2 Conditions

When a device is in limited discoverable mode it shall set bit no 13 in the Major Service Class part of the Class of Device/Service field [11].

#### 4.1.2.3 Term on UI-level

Bluetooth device is 'discoverable' or in 'discoverable mode'.

### **4.1.3 General discoverable mode**

#### 4.1.3.1 Definition

The general discoverable mode shall be used by devices that need to be discoverable continuously or for no specific condition. The purpose is to respond to a device that makes a general inquiry (inquiry using the GIAC).

#### 4.1.3.2 Conditions

When a Bluetooth device is in general discoverable mode, it shall enter the INQUIRY\_SCAN state more often than once in  $T_{GAP}(102)$  and scan for the GIAC for at least  $T_{GAP}(101)$ .

A device in general discoverable mode shall not respond to a LIAC inquiry.

#### 4.1.3.3 Term on UI-level

Bluetooth device is 'discoverable' or in 'discoverable mode'.

## **4.2 CONNECTABILITY MODES**

With respect to paging, a Bluetooth device shall be either in non-connectable mode or in connectable mode. Paging is defined in [1].

When a Bluetooth device is in non-connectable mode it does not respond to paging. When a Bluetooth device is in connectable mode it responds to paging.

### **4.2.1 Non-connectable mode**

#### 4.2.1.1 Definition

When a Bluetooth device is in non-connectable mode it shall never enter the PAGE\_SCAN state.

#### 4.2.1.2 Term on UI-level

Bluetooth device is 'non-connectable' or in 'non-connectable mode'.

### **4.2.2 Connectable mode**

#### 4.2.2.1 Definition

When a Bluetooth device is in connectable mode it shall periodically enter the PAGE\_SCAN state.

#### 4.2.2.2 Term on UI-level

Bluetooth device is 'connectable' or in 'connectable mode'.

## **4.3 PAIRING MODES**

With respect to pairing, a Bluetooth device shall be either in non-pairable mode or in pairable mode. In pairable mode the Bluetooth device accepts pairing – i.e. creation of bonds – initiated by the remote device, and in non-pairable mode it does not. Pairing is defined in [1] and [2].

### **4.3.1 Non-pairable mode**

#### 4.3.1.1 Definition

When a Bluetooth device is in non-pairable mode it shall respond to a received LMP\_in\_rand with LMP\_not\_accepted with the reason *pairing not allowed*.

#### 4.3.1.2 Term on UI-level

Bluetooth device is 'non-bondable' or in 'non-bondable mode' or "does not accept bonding".

### **4.3.2 Pairable mode**

#### 4.3.2.1 Definition

When a Bluetooth device is in pairable mode it shall respond to a received LMP\_in\_rand with LMP\_accepted (or with LMP\_in\_rand if it has a fixed PIN).

#### 4.3.2.2 Term on UI-level

Bluetooth device is 'bondable' or in 'bondable mode' or "accepts bonding".

## 5 SECURITY ASPECTS

	Procedure	Ref.	Support
1	Authentication	5.1	C1
2	Security modes	5.2	
	Security mode 1		O
	Security mode 2		C2
	Security mode 3		C2
C1: If security mode 1 is the only security mode that is supported, support for authentication is optional, otherwise mandatory. (Note: support for LMP-authentication and LMP-pairing is mandatory according [2] independent of which security mode that is used.)			
C2: If security mode 1 is not the only security mode that is supported, then support for at least one of security mode 2 or security mode 3 is mandatory.			

Table 5.1: Conformance requirements related to the generic authentication procedure and the security modes defined in this section

### 5.1 AUTHENTICATION

#### 5.1.1 Purpose

The generic authentication procedure describes how the LMP-authentication and LMP-pairing procedures are used when authentication is initiated by one Bluetooth device towards another, depending on if a link key exists or not and if pairing is allowed or not.

#### 5.1.2 Term on UI level

'Bluetooth authentication'.

**5.1.3 Procedure**

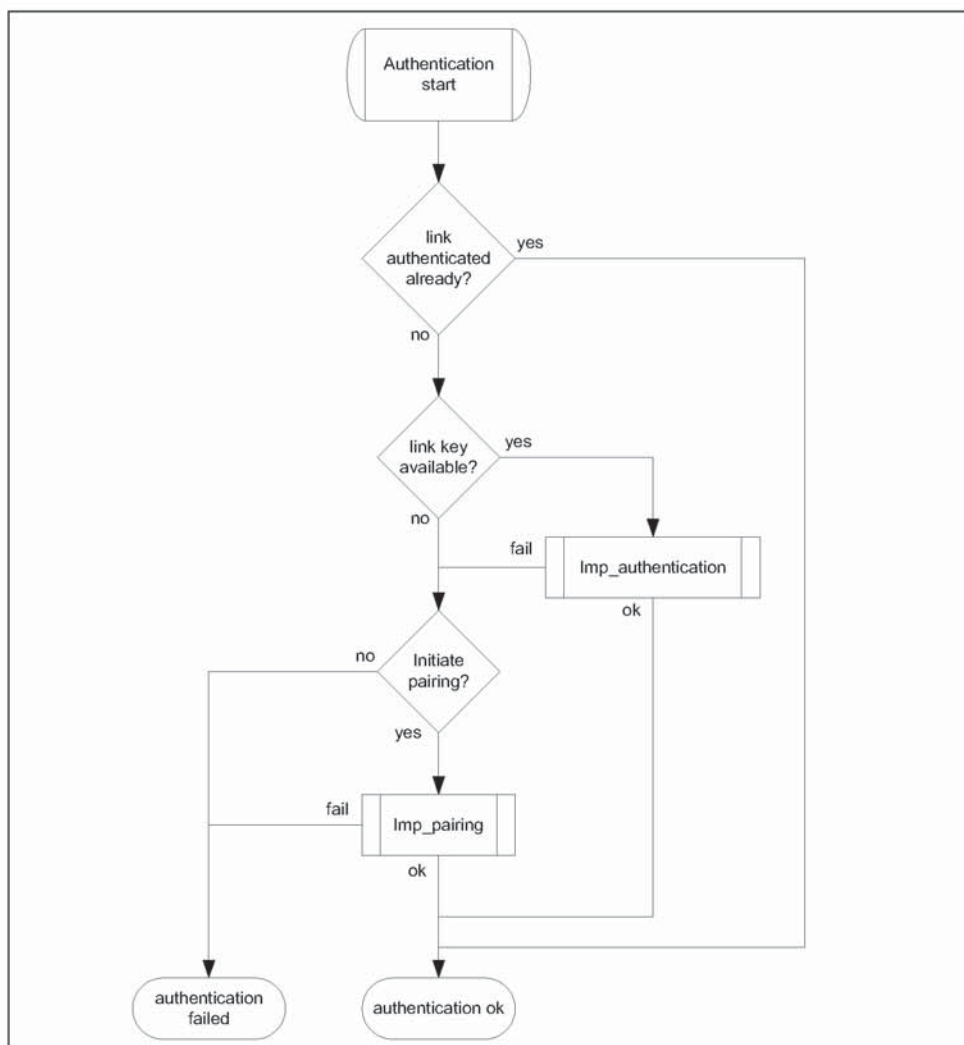


Figure 5.1: Definition of the generic authentication procedure.

**5.1.4 Conditions**

The device that initiates authentication has to be in security mode 2 or in security mode 3.

**5.2 SECURITY MODES**

The following flow chart describes where in the channel establishment procedures initiation of authentication takes place, depending on which security mode the Bluetooth device is in.

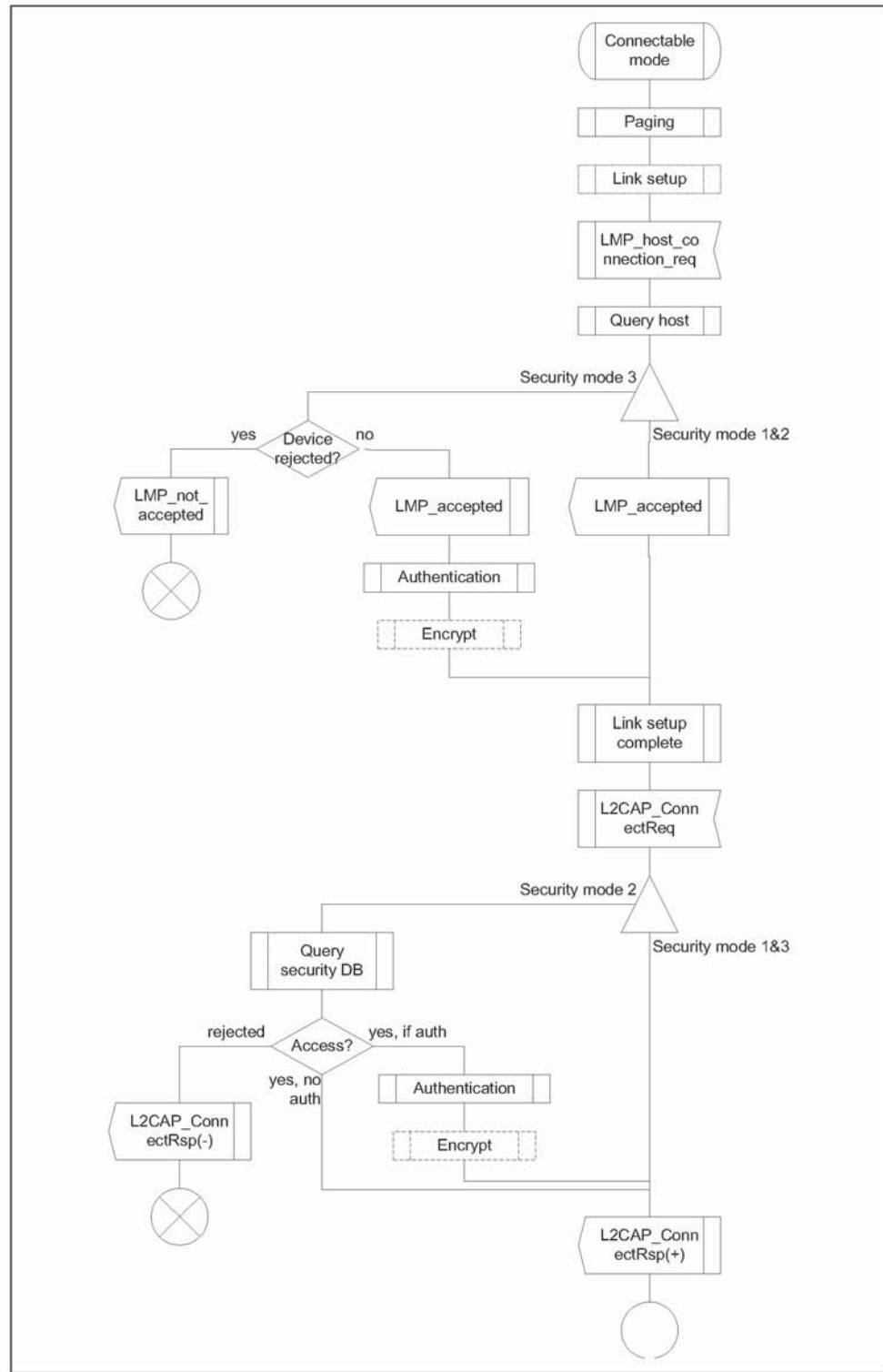


Figure 5.2: Illustration of channel establishment using different security modes.

When authentication is initiated towards a Bluetooth device, it shall act according to [2] and the current pairing mode, independent of which security mode it is in.

### 5.2.1 Security mode 1 (non-secure)

When a Bluetooth device is in security mode 1 it shall never initiate any security procedure (i.e., it shall never send LMP\_au\_rand, LMP\_in\_rand or LMP\_encryption\_mode\_req).

### 5.2.2 Security mode 2 (service level enforced security)

When a Bluetooth device is in security mode 2 it shall not initiate any security procedure before a channel establishment request (L2CAP\_ConnectReq) has been received or a channel establishment procedure has been initiated by itself. (The behavior of a device in security mode 2 is further described in [10].) Whether a security procedure is initiated or not depends on the security requirements of the requested channel or service.

A Bluetooth device in security mode 2 should classify the security requirements of its services using at least the following attributes:

- Authorization required;
- Authentication required;
- Encryption required.

*Note: Security mode 1 can be considered (at least from a remote device point of view) as a special case of security mode 2 where no service has registered any security requirements.*

### 5.2.3 Security modes 3 (link level enforced security)

When a Bluetooth device is in security mode 3 it shall initiate security procedures before it sends LMP\_link\_setup\_complete. (The behavior of a device in security mode 3 is as described in [2].)

A Bluetooth device in security mode 3 may reject the host connection request (respond with LMP\_not\_accepted to the LMP\_host\_connection\_req) based on settings in the host (e.g. only communication with pre-paired devices allowed).



## 6 IDLE MODE PROCEDURES

The inquiry and discovery procedures described here are applicable only to the device that initiates them (A). The requirements on the behavior of B is according to the modes specified in Section 4 and to [2].

	Procedure	Ref.	Support
1	General inquiry	6.1	C1
2	Limited inquiry	6.2	C1
3	Name discovery	6.3	O
4	Device discovery	6.4	O
5	Bonding	6.5	O

C1: If initiation of bonding is supported, support for at least one inquiry procedure is mandatory, otherwise optional.  
(Note: support for LMP-pairing is mandatory [2].)

### 6.1 GENERAL INQUIRY

#### 6.1.1 Purpose

The purpose of the general inquiry procedure is to provide the initiator with the Bluetooth device address, clock, Class of Device and used page scan mode of general discoverable devices (i.e. devices that are in range with regard to the initiator and are set to scan for inquiry messages with the General Inquiry Access Code). Also devices in limited discoverable mode will be discovered using general inquiry.

The general inquiry should be used by devices that need to discover devices that are made discoverable continuously or for no specific condition.

#### 6.1.2 Term on UI level

'Bluetooth Device Inquiry'.

**6.1.3 Description**

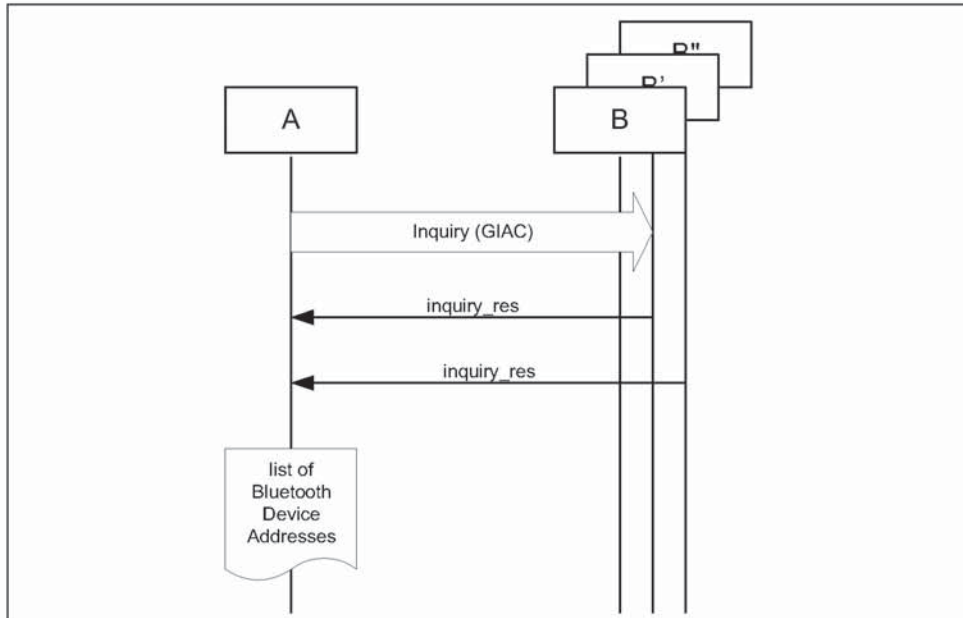


Figure 6.1: General inquiry, where B is a device in non-discoverable mode, B' is a device in limited discoverable mode and B'' is a device in general discoverable mode. (Note that all discoverable devices are discovered using general inquiry, independent of which discoverable mode they are in.)

**6.1.4 Conditions**

When general inquiry is initiated by a Bluetooth device, it shall be in the INQUIRY state for at least  $T_{GAP}(100)$  and perform inquiry using the GIAC.

In order to receive inquiry response, the remote devices in range have to be made discoverable (limited or general).

**6.2 LIMITED INQUIRY**

**6.2.1 Purpose**

The purpose of the limited inquiry procedure is to provide the initiator with the Bluetooth device address, clock, Class of Device and used page scan mode of limited discoverable devices. The latter devices are devices that are in range with regard to the initiator, and may be set to scan for inquiry messages with the Limited Inquiry Access Code, in addition to scanning for inquiry messages with the General Inquiry Access Code.

The limited inquiry should be used by devices that need to discover devices that are made discoverable only for a limited period of time, during temporary conditions or for a specific event. Since it is not guaranteed that the

discoverable device scans for the LIAC, the initiating device may choose any inquiry procedure (general or limited). Even if the remote device that is to be discovered is expected to be made limited discoverable (e.g. when a dedicated bonding is to be performed), the limited inquiry should be done in sequence with a general inquiry in such a way that both inquiries are completed within the time the remote device is limited discoverable, i.e. at least  $T_{GAP}(103)$ .

### 6.2.2 Term on UI level

'Bluetooth Device Inquiry'.

### 6.2.3 Description

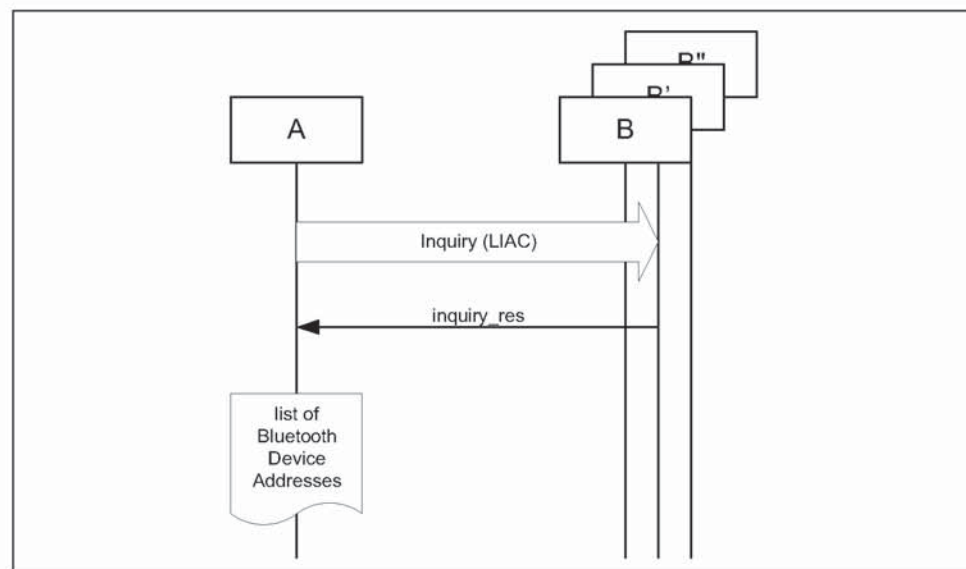


Figure 6.2: Limited inquiry where B is a device in non-discoverable mode, B' is a device in limited discoverable mode and B'' is a device in general discoverable mode. (Note that only limited discoverable devices can be discovered using limited inquiry.)

### 6.2.4 Conditions

When limited inquiry is initiated by a Bluetooth device, it shall be in the INQUIRY state for at least  $T_{GAP}(100)$  and perform inquiry using the LIAC.

In order to receive inquiry response, the remote devices in range has to be made limited discoverable.

## 6.3 NAME DISCOVERY

### 6.3.1 Purpose

The purpose of name discovery is to provide the initiator with the Bluetooth Device Name of connectable devices (i.e. devices in range that will respond to paging).

### 6.3.2 Term on UI level

'Bluetooth Device Name Discovery'.

### 6.3.3 Description

#### 6.3.3.1 Name request

Name request is the procedure for retrieving the Bluetooth Device Name from a connectable Bluetooth device. It is not necessary to perform the full link establishment procedure (see Section 7.1) in order to just to get the name of another device.

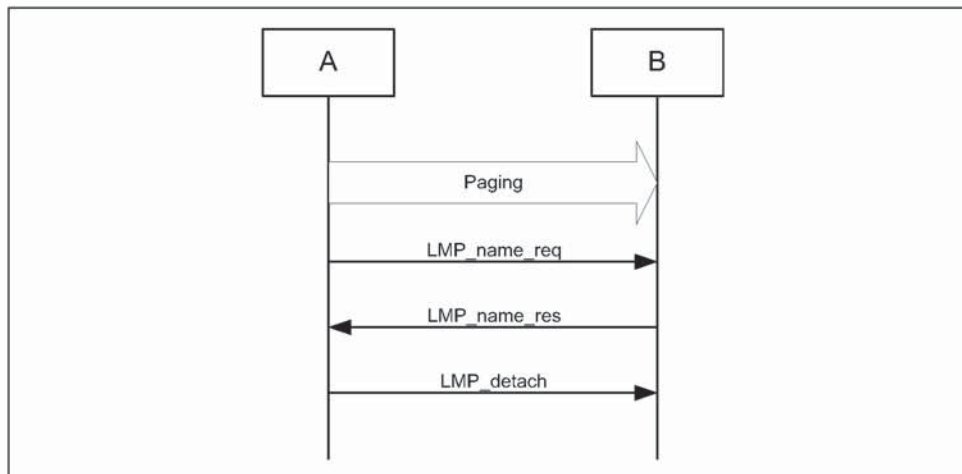


Figure 6.3: Name request procedure.

#### 6.3.3.2 Name discovery

Name discovery is the procedure for retrieving the Bluetooth Device Name from connectable Bluetooth devices by performing name request towards known devices (i.e. Bluetooth devices for which the Bluetooth Device Addresses are available).

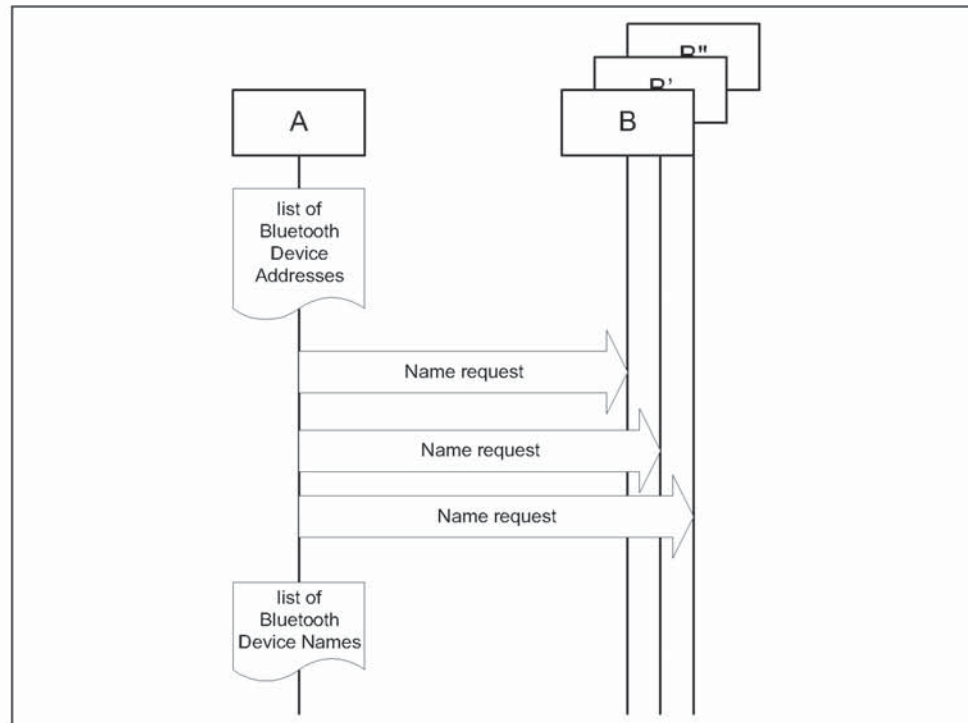


Figure 6.4: Name discovery procedure.

### 6.3.4 Conditions

In the name request procedure, the initiator will use the Device Access Code of the remote device as retrieved immediately beforehand – normally through an inquiry procedure.

## 6.4 DEVICE DISCOVERY

### 6.4.1 Purpose

The purpose of device discovery is to provide the initiator with the Bluetooth Address, clock, Class of Device, used page scan mode and Bluetooth device name of discoverable devices.

### 6.4.2 Term on UI level

'Bluetooth Device Discovery'.

**6.4.3 Description**

During the device discovery procedure, first an inquiry (either general or limited) is performed, and then name discovery is done towards some or all of the devices that responded to the inquiry.

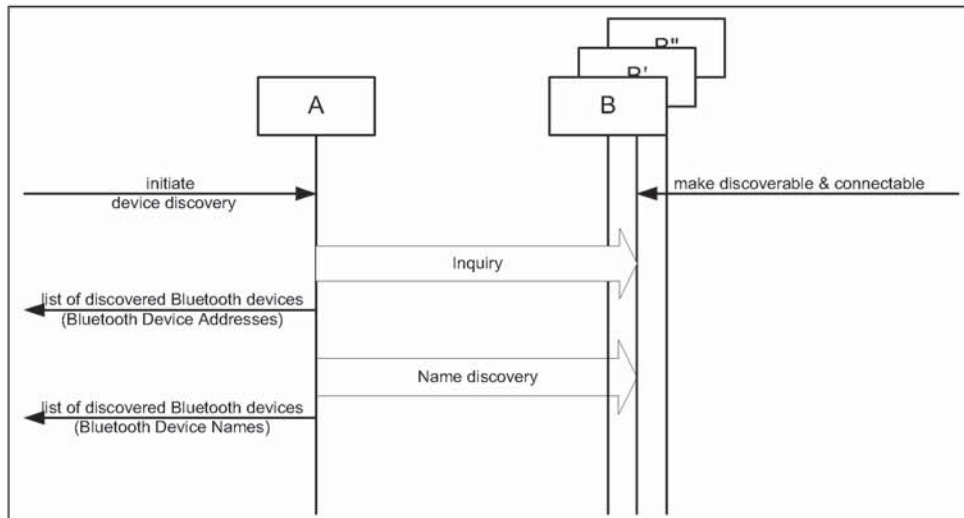


Figure 6.5: Device discovery procedure.

**6.4.4 Conditions**

Conditions for both inquiry (general or limited) and name discovery must be fulfilled (i.e. devices discovered during device discovery must be both discoverable and connectable).

**6.5 BONDING**

**6.5.1 Purpose**

The purpose of bonding is to create a relation between two Bluetooth devices based on a common link key (a bond). The link key is created and exchanged (pairing) during the bonding procedure and is expected to be stored by both Bluetooth devices, to be used for future authentication.

In addition to pairing, the bonding procedure can involve higher layer initialization procedures.

**6.5.2 Term on UI level**

'Bluetooth Bonding'

**6.5.3 Description**

Two aspects of the bonding procedure are described here. Dedicated bonding is what is done when the two devices are explicitly set to perform only a creation and exchange of a common link key.

General bonding is included to indicate that the framework for the dedicated bonding procedure is the same as found in the normal channel and connection establishment procedures. This means that pairing may be performed successfully if A has initiated bonding while B is in its normal connectable and security modes.

The main difference with bonding, as compared to a pairing done during link or channel establishment, is that for bonding it is the paging device (A) that must initiate the authentication.

6.5.3.1 General bonding

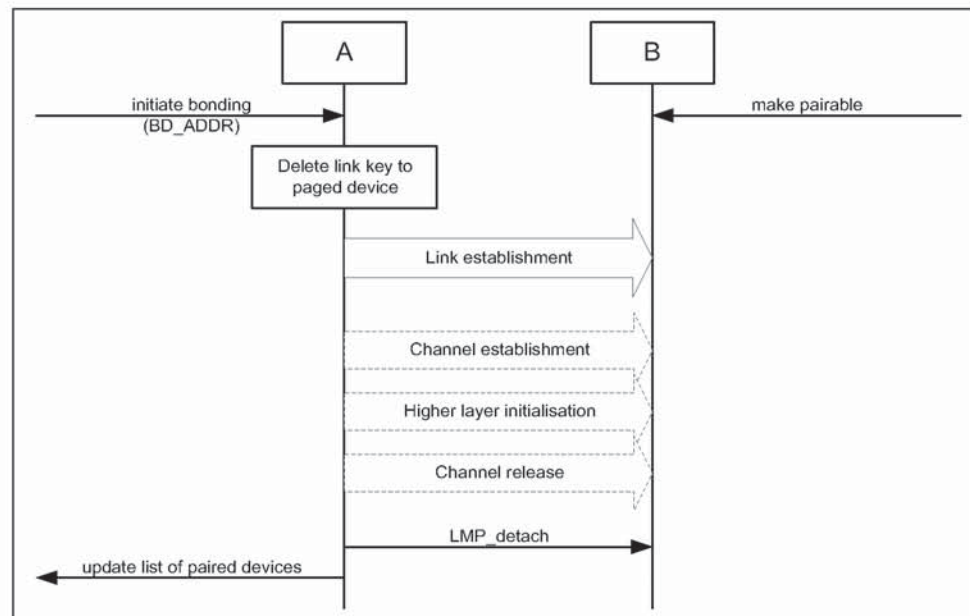


Figure 6.6: General description of bonding as being the link establishment procedure executed under specific conditions on both devices, followed by an optional higher layer initialization process.

6.5.3.2 Dedicated bonding

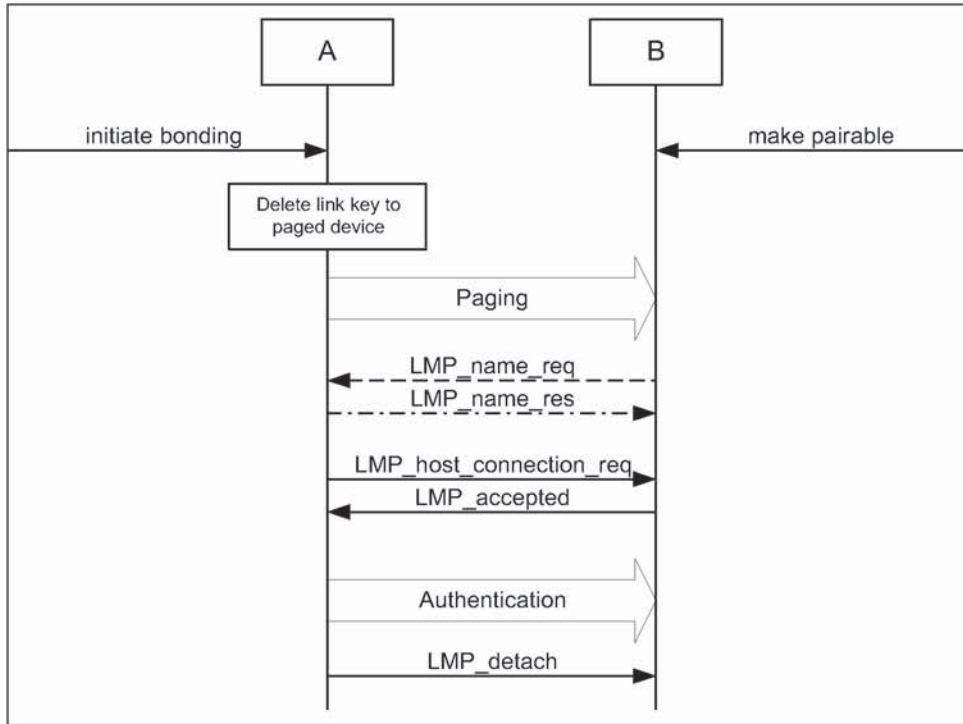


Figure 6.7: Bonding as performed when the purpose of the procedure is only to create and exchange a link key between two Bluetooth devices.

6.5.4 Conditions

Before bonding can be initiated, the initiating device (A) must know the Device Access Code of the device to pair with. This is normally done by first performing device discovery. A Bluetooth Device that can initiate bonding (A) should use limited inquiry, and a Bluetooth Device that accepts bonding (B) should support the limited discoverable mode.

Bonding is in principle the same as link establishment with the conditions:

- The paged device (B) shall be set into pairable mode. The paging device (A) is assumed to allow pairing since it has initiated the bonding procedure.
- The paging device (the initiator of the bonding procedure, A) shall initiate authentication.
- Before initiating the authentication part of the bonding procedure, the paging device should delete any link key corresponding to a previous bonding with the paged device.
- If the paging device does not intend to initiate any higher layer initialization during bonding, it need not send LMP\_host\_request before initiating authentication.



## 7 ESTABLISHMENT PROCEDURES

	Procedure	Ref.	Support in A	Support in B
1	Link establishment	7.1	M	M
2	Channel establishment	7.2	O	M
3	Connection establishment	7.3	O	O

Table 7.1: Establishment procedures

The establishment procedures defined here do not include any discovery part. Before establishment procedures are initiated, the information provided during device discovery (in the FHS packet of the inquiry response or in the response to a name request) has to be available in the initiating device. This information is:

- The Bluetooth Device Address (BD\_ADDR) from which the Device Access Code is generated;
- The system clock of the remote device;
- The page scan mode used by the remote device.

Additional information provided during device discovery that is useful for making the decision to initiate an establishment procedure is:

- The Class of device;
- The Device name.

### 7.1 LINK ESTABLISHMENT

#### 7.1.1 Purpose

The purpose of the link establishment procedure is to establish a physical link (of ACL type) between two Bluetooth devices using procedures from [1] and [2].

#### 7.1.2 Term on UI level

'Bluetooth link establishment'

**7.1.3 Description**

In this sub-section, the paging device (A) is in security mode 3. The paging device cannot during link establishment distinguish if the paged device (B) is in security mode 1 or 2.

7.1.3.1 B in security mode 1 or 2

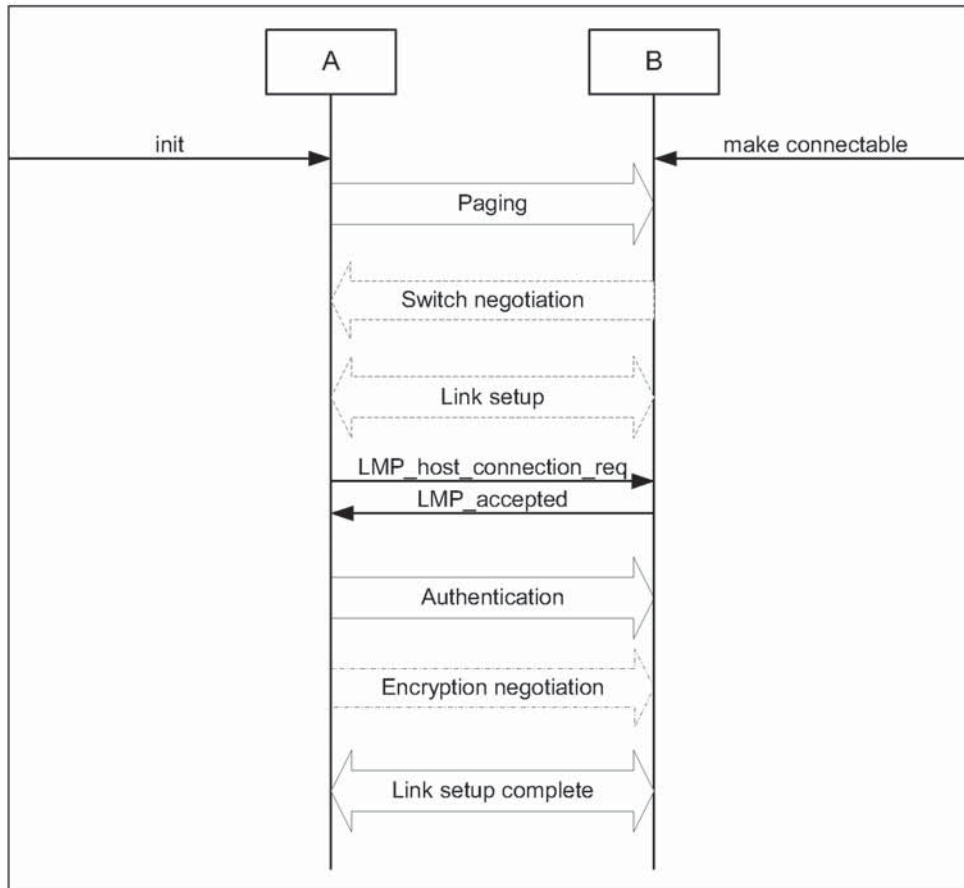


Figure 7.1: Link establishment procedure when the paging device (A) is in security mode 3 and the paged device (B) is in security mode 1 or 2.

7.1.3.2 B in security mode 3

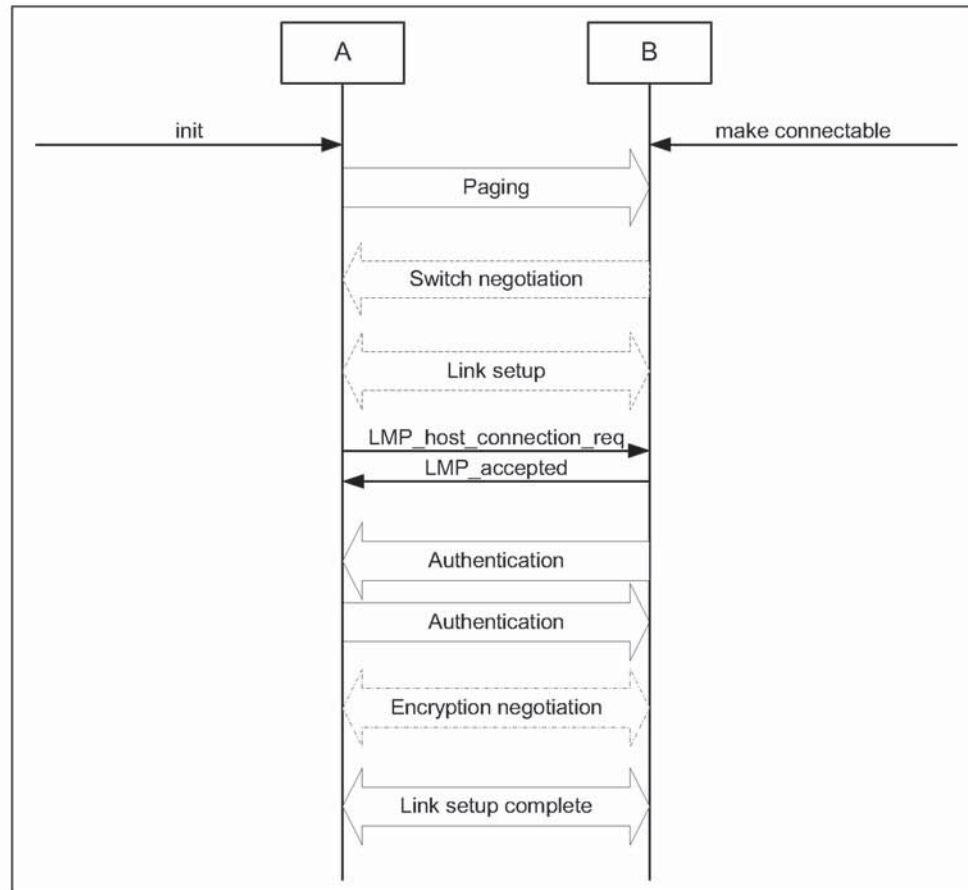


Figure 7.2: Link establishment procedure when both the paging device (A) and the paged device (B) are in security mode 3.

7.1.4 Conditions

The paging procedure shall be according to [1] and the paging device should use the Device access code and page mode received through a previous inquiry. When paging is completed, a physical link between the two Bluetooth devices is established.

If role switching is needed (normally it is the paged device that has an interest in changing the master/slave roles) it should be done as early as possible after the physical link is established. If the paging device does not accept the switch, the paged device has to consider whether to keep the physical link or not.

Both devices may perform link setup (using LMP procedures that require no interaction with the host on the remote side). Optional LMP features can be used after having confirmed (using LMP\_feature\_req) that the other device supports the feature.

When the paging device needs to go beyond the link setup phase, it issues a request to be connected to the host of the remote device. If the paged device is in security mode 3, this is the trigger for initiating authentication.

The paging device shall send LMP\_host\_connection\_req during link establishment (i.e. before channel establishment) and may initiate authentication only after having sent LMP\_host\_connection\_request.

After an authentication has been performed, any of the devices can initiate encryption.

Further link configuration may take place after the LMP\_host\_connection\_req. When both devices are satisfied, they send LMP\_setup\_complete.

Link establishment is completed when both devices have sent LMP\_setup\_complete.

## **7.2 CHANNEL ESTABLISHMENT**

### **7.2.1 Purpose**

The purpose of the channel establishment procedure is to establish a Bluetooth channel (a logical link) between two Bluetooth devices using [3].

### **7.2.2 Term on UI level**

'Bluetooth channel establishment'.

### **7.2.3 Description**

In this sub-section, the initiator (A) is in security mode 3. During channel establishment, the initiator cannot distinguish if the acceptor (B) is in security mode 1 or 3.

7.2.3.1 B in security mode 2

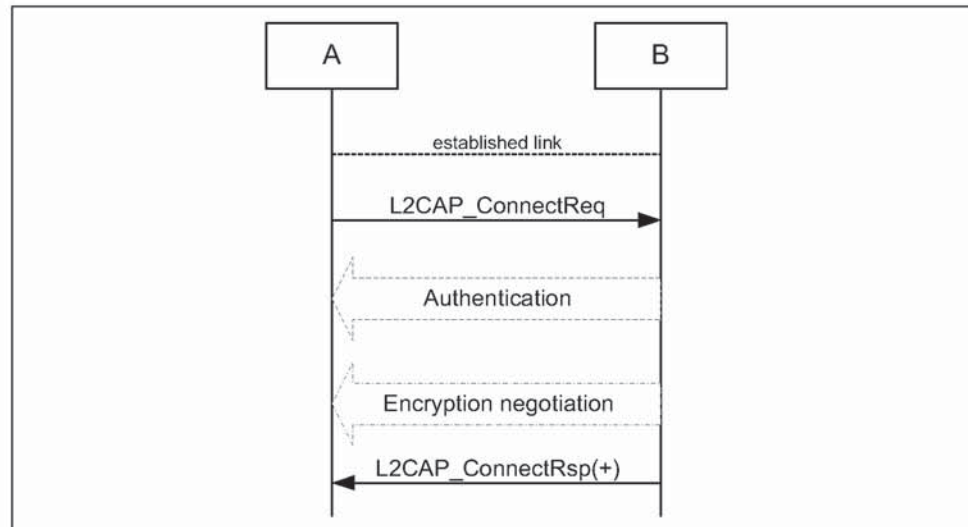


Figure 7.3: Channel establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 2.

7.2.3.2 B in security mode 1 or 3

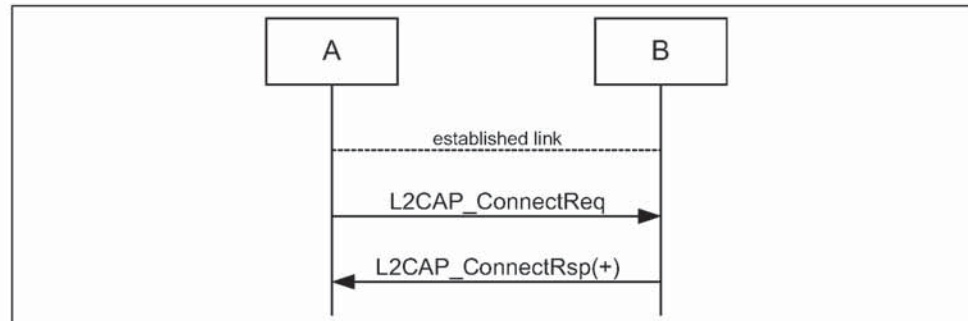


Figure 7.4: Channel establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 1 or 3.

**7.2.4 Conditions**

Channel establishment starts after link establishment is completed when the initiator sends a channel establishment request (L2CAP\_ConnectReq).

Depending on security mode, security procedures may take place after the channel establishment has been initiated.

Channel establishment is completed when the acceptor responds to the channel establishment request (with a positive L2CAP\_ConnectRsp).

### 7.3 CONNECTION ESTABLISHMENT

#### 7.3.1 Purpose

The purpose of the connection establishment procedure is to establish a connection between applications on two Bluetooth devices.

#### 7.3.2 Term on UI level

'Bluetooth connection establishment'

#### 7.3.3 Description

In this sub-section, the initiator (A) is in security mode 3. During connection establishment, the initiator cannot distinguish if the acceptor (B) is in security mode 1 or 3.

##### 7.3.3.1 B in security mode 2

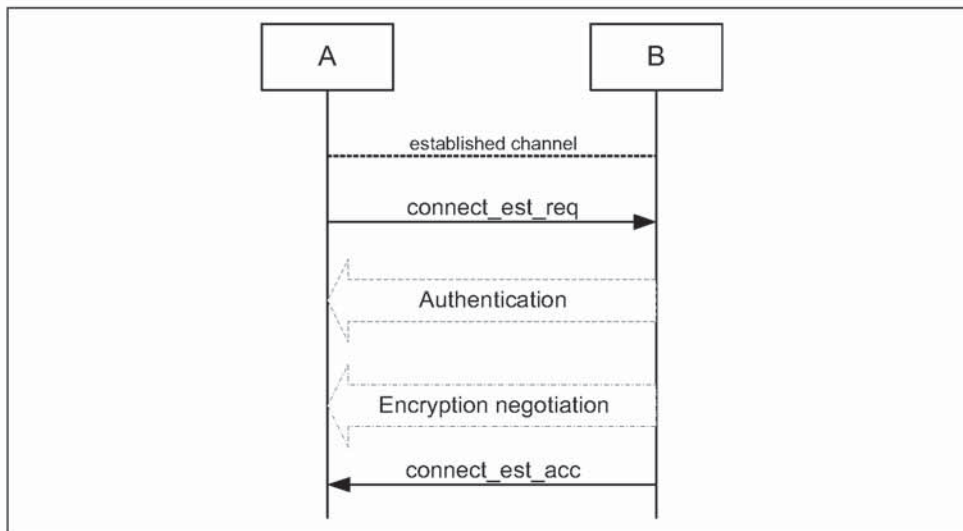


Figure 7.5: Connection establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 2.

### 7.3.3.2 B in security mode 1 or 3

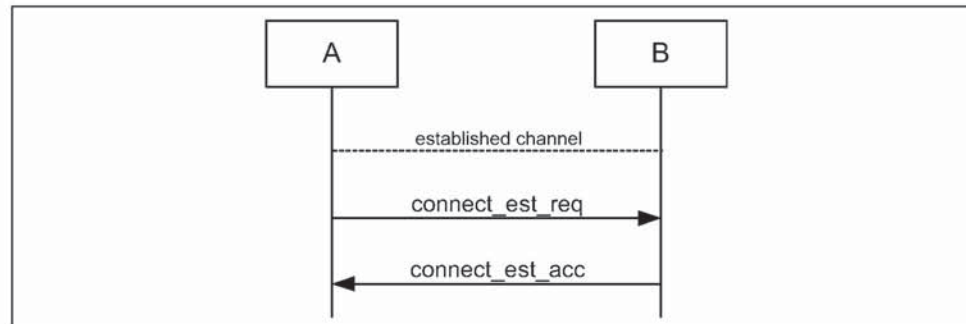


Figure 7.6: Connection establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 1 or 3.

### 7.3.4 Conditions

Connection establishment starts after channel establishment is completed, when the initiator sends a connection establishment request ('connect\_est\_req' is application protocol-dependent). This request may be a TCS SETUP message [5] in the case of a Bluetooth telephony application Cordless Telephony Profile, or initialization of RFCOMM and establishment of DLC [4] in the case of a serial port-based application Serial Port Profile (although neither TCS or RFCOMM use the term 'connection' for this).

Connection establishment is completed when the acceptor accepts the connection establishment request ('connect\_est\_acc' is application protocol dependent).

## 7.4 ESTABLISHMENT OF ADDITIONAL CONNECTION

When a Bluetooth device has established one connection with another Bluetooth device, it may be available for establishment of:

- A second connection on the same channel, and/or
- A second channel on the same link, and/or
- A second physical link.

If the new establishment procedure is to be towards the same device, the security part of the establishment depends on the security modes used. If the new establishment procedure is to be towards a new remote device, the device should behave according to active modes independent of the fact that it already has another physical link established (unless allowed co-incident radio and baseband events have to be handled).

## 8 DEFINITIONS

---

In the following, terms written with capital letters refer to states.

### 8.1 GENERAL DEFINITIONS

**Mode** A set of directives that defines how a device will respond to certain events.

**Idle** As seen from a remote device, a Bluetooth device is idle, or is in idle mode, when there is no link established between them.

**Bond** A relation between two Bluetooth devices defined by creating, exchanging and storing a common link key. The bond is created through the bonding or LMP-pairing procedures.

### 8.2 CONNECTION-RELATED DEFINITIONS

**Physical channel** A synchronized Bluetooth baseband-compliant RF hopping sequence.

**Piconet** A set of Bluetooth devices sharing the same physical channel defined by the master parameters (clock and BD\_ADDR).

**Physical link** A Baseband-level connection<sup>1</sup> between two devices established using paging. A physical link comprises a sequence of transmission slots on a physical channel alternating between master and slave transmission slots.

**ACL link** An asynchronous (packet-switched) connection<sup>1</sup> between two devices created on LMP level. Traffic on an ACL link uses ACL packets to be transmitted.

**SCO link** A synchronous (circuit-switched) connection<sup>1</sup> for reserved bandwidth communications; e.g. voice between two devices, created on the LMP level by reserving slots periodically on a physical channel. Traffic on an SCO link uses SCO packets to be transmitted. SCO links can be established only after an ACL link has first been established.

**Link** Shorthand for an ACL link.

**PAGE** A baseband state where a device transmits page trains, and processes any eventual responses to the page trains.

**PAGE\_SCAN** A baseband state where a device listens for page trains.

---

1. The term 'connection' used here is not identical to the definition below. It is used in the absence of a more concise term.



**Page** The transmission by a device of page trains containing the Device Access Code of the device to which the physical link is requested.

**Page scan** The listening by a device for page trains containing its own Device Access Code.

**Channel** A logical connection on L2CAP level between two devices serving a single application or higher layer protocol.

**Connection** A connection between two peer applications or higher layer protocols mapped onto a channel.

**Connecting** A phase in the communication between devices when a connection between them is being established. (Connecting phase follows after the link establishment phase is completed.)

**Connect (to service)** The establishment of a connection to a service. If not already done, this includes establishment of a physical link, link and channel as well.

### 8.3 DEVICE-RELATED DEFINITIONS

**Discoverable device** A Bluetooth device in range that will respond to an inquiry (normally in addition to responding to page).

**Silent device** A Bluetooth device appears as silent to a remote device if it does not respond to inquiries made by the remote device. A device may be silent due to being non-discoverable or due to baseband congestion while being discoverable.

**Connectable device** A Bluetooth device in range that will respond to a page.

**Trusted device** A paired device that is explicitly marked as trusted.

**Paired device** A Bluetooth device with which a link key has been exchanged (either before connection establishment was requested or during connecting phase).

**Pre-paired device** A Bluetooth device with which a link key was exchanged, and the link key is stored, before link establishment.

**Un-paired device** A Bluetooth device for which there was no exchanged link key available before connection establishment was request.

**Known device** A Bluetooth device for which at least the BD\_ADDR is stored.

**Un-known device** A Bluetooth device for which no information (BD\_ADDR, link key or other) is stored.

**Authenticated device** A Bluetooth device whose identity has been verified during the lifetime of the current link, based on the authentication procedure.

## 8.4 PROCEDURE-RELATED DEFINITIONS

**Paging** A procedure for establishing a physical link of ACL type on baseband level, consisting of a page action of the initiator and a page scan action of the responding device.

**Link establishment** A procedure for establishing a link on LMP level. A link is established when both devices have agreed that LMP setup is completed.

**Channel establishment** A procedure for establishing a channel on L2CAP level.

**Connection establishment** A procedure for creating a connection mapped onto a channel.

**Creation of a trusted relationship** A procedure where the remote device is marked as a trusted device. This includes storing a common link key for future authentication and pairing (if the link key is not available).

**Creation of a secure connection.** A procedure of establishing a connection, including authentication and encryption.

**Device discovery** A procedure for retrieving the Bluetooth device address, clock, class-of-device field and used page scan mode from discoverable devices.

**Name discovery** A procedure for retrieving the user-friendly name (the Bluetooth device name) of a connectable device.

**Service discovery** Procedures for querying and browsing for services offered by or through another Bluetooth device.

## 8.5 SECURITY-RELATED DEFINITIONS

**Authentication** A generic procedure based on LMP-authentication if a link key exists or on LMP-pairing if no link key exists.

**LMP-authentication** An LMP level procedure for verifying the identity of a remote device. The procedure is based on a challenge-response mechanism using a random number, a secret key and the BD\_ADDR of the non-initiating device. The secret key used can be a previously exchanged link key or an initialization key created based on a PIN (as used when pairing).

**Authorization** A procedure where a user of a Bluetooth device grants a specific (remote) Bluetooth device access to a specific service. Authorization

implies that the identity of the remote device can be verified through authentication.

**Authorize** The act of granting a specific Bluetooth device access to a specific service. It may be based upon user confirmation, or given the existence of a trusted relationship.

**LMP-pairing** A procedure that authenticates two devices, based on a PIN, and subsequently creates a common link key that can be used as a basis for a trusted relationship or a (single) secure connection. The procedure consists of the steps: creation of an initialization key (based on a random number and a PIN), LMP-authentication based on the initialization key and creation of a common link key.

**Bonding** A dedicated procedure for performing the first authentication, where a common link key is created and stored for future use.

**Trusting** The marking of a paired device as trusted. Trust marking can be done by the user, or automatically by the device (e.g. when in pairable mode) after a successful pairing.

**9 ANNEX A (NORMATIVE): TIMERS AND CONSTANTS**

The following timers are required by this profile.

Timer name	Recommended value	Description	Comment
$T_{GAP}(100)$	10.24 s	Normal time span that a Bluetooth device performs inquiry.	Used during inquiry and device discovery.
$T_{GAP}(101)$	10.625 ms	Minimum time in INQUIRY_SCAN.	A discoverable Bluetooth device enters INQUIRY_SCAN for at least $T_{GAP}(101)$ every $T_{GAP}(102)$ .
$T_{GAP}(102)$	2.56 s	Maximum time between repeated INQUIRY_SCAN enterings.	Maximum value of the inquiry scan interval, $T_{inquiry\ scan}$ .
$T_{GAP}(103)$	30.72 s	A Bluetooth device shall not be in a discoverable mode less than $T_{GAP}(103)$ .	Minimum time to be discoverable.
$T_{GAP}(104)$	1 min	A Bluetooth device should not be in limited discoverable mode more than $T_{GAP}(104)$ .	Recommended upper limit.

Table 9.1: Defined GAP timers

## 10 ANNEX B (INFORMATIVE): INFORMATION FLOWS OF RELATED PROCEDURES

### 10.1 LMP-AUTHENTICATION

The specification of authentication on link level is found in [2].

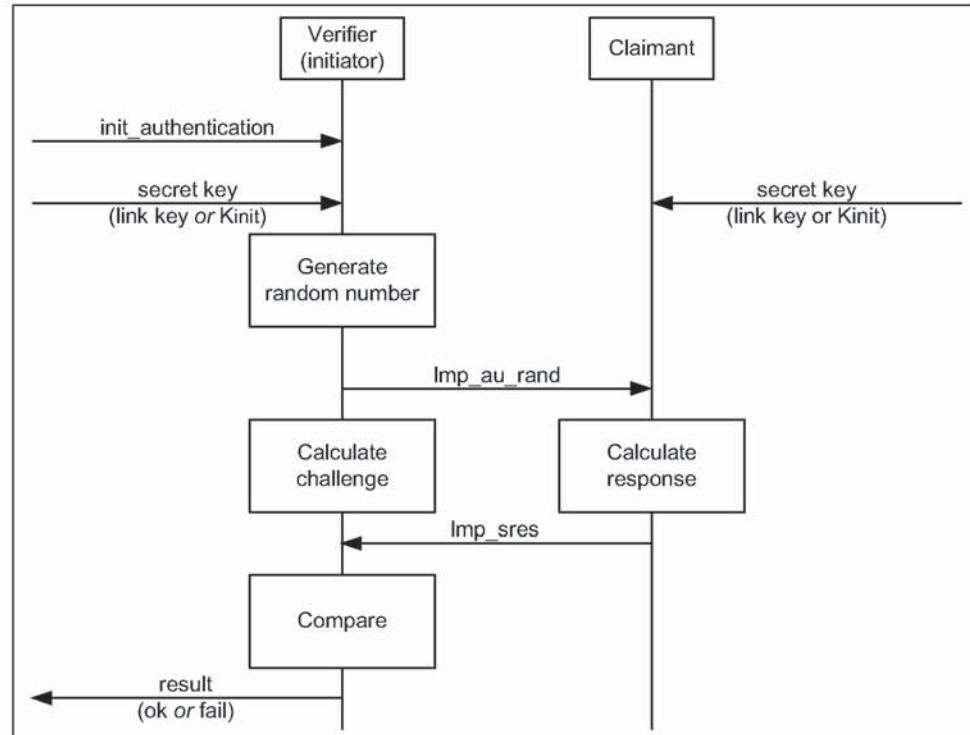


Figure 10.1: LMP-authentication as defined by [2].

The secret key used here may be either an already exchanged link key or an initialization key created in the LMP-pairing procedure.

## 10.2 LMP-PAIRING

The specification of pairing on link level is found in [2].

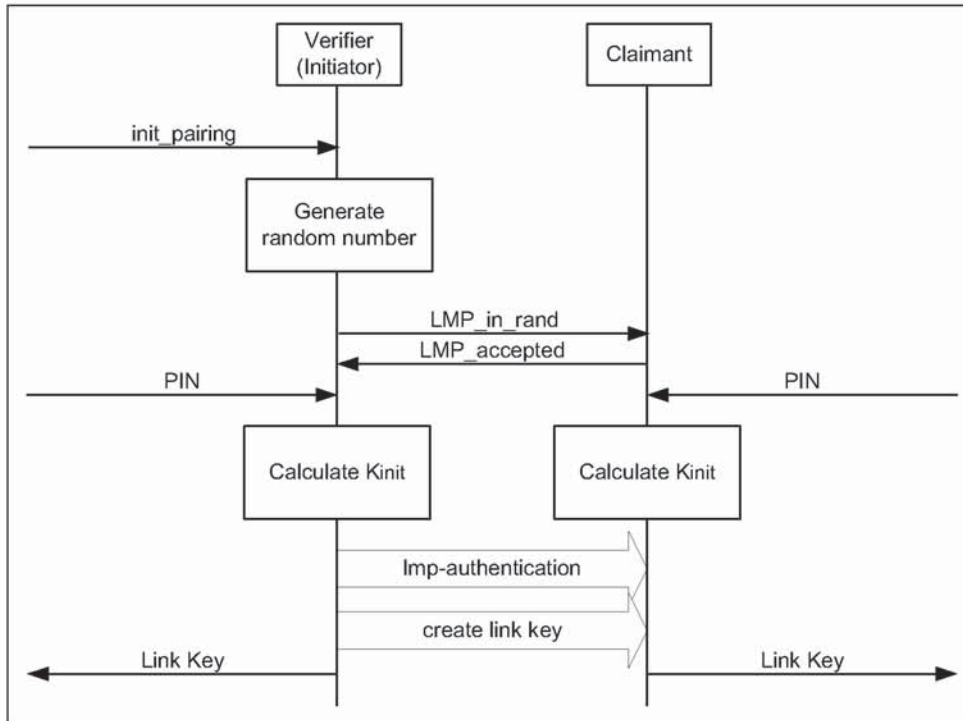


Figure 10.2: LMP-pairing as defined in [2].

The PIN used here is  $PN_{BB}$ .

The create link key procedure is described in section 3.3.4 of [2] and section 14.2.2 of [1]. In case the link key is based on a combination key, a mutual authentication takes place and shall be performed irrespective of current security mode.

## 10.3 SERVICE DISCOVERY

The Service Discovery Protocol [6] specifies what PDUs are used over-the-air to inquire about services and service attributes. The procedures for discovery of supported services and capabilities using the Service Discovery Protocol are described in the Service Discovery Application Profile. This is just an example.

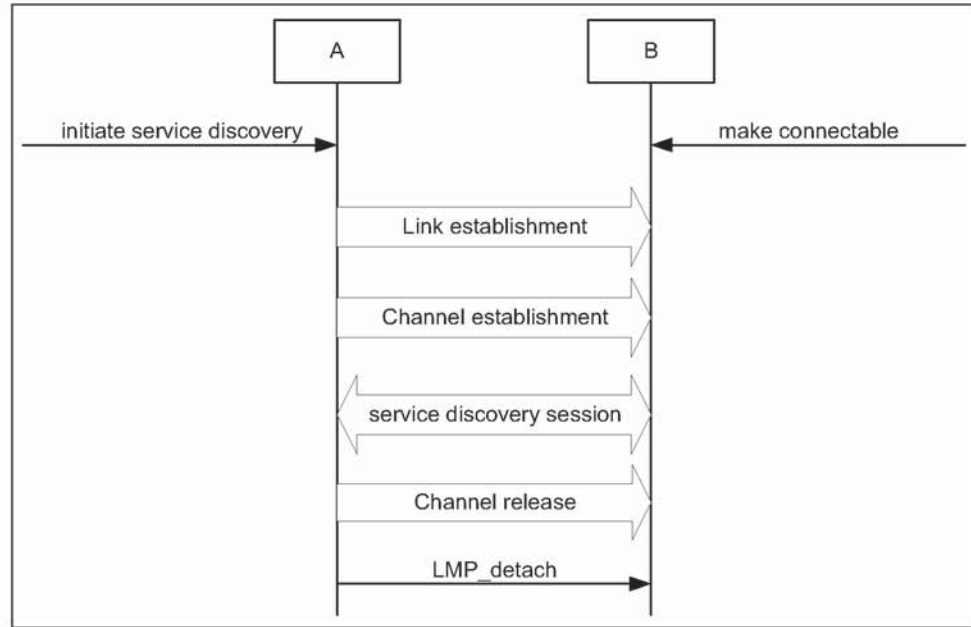


Figure 10.3: Service discovery procedure.

## 11 REFERENCES

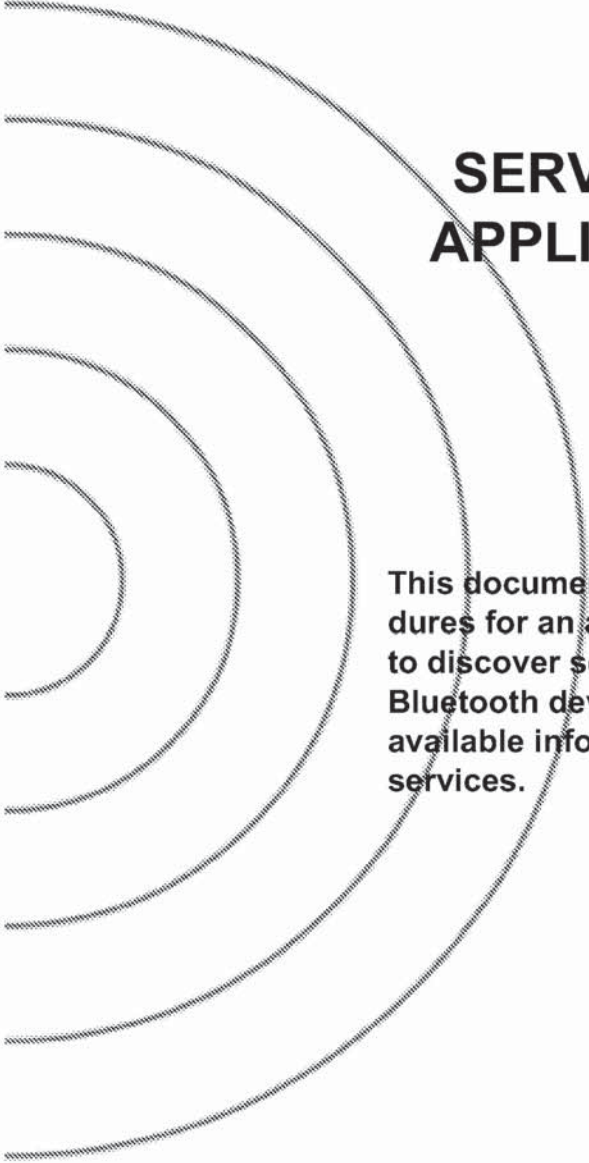
---

- [1] Bluetooth Baseband Specification
- [2] Bluetooth Link Manager Protocol
- [3] Bluetooth Logical Link Control and Adaptation Protocol
- [4] Bluetooth RFCOMM
- [5] Bluetooth Telephony Control Specification
- [6] Bluetooth Service Discovery Protocol
- [7] Bluetooth Service Discovery Application Profile
- [8] Bluetooth Cordless Telephony Profile
- [9] Bluetooth Serial Port Profile
- [10] Bluetooth Security Architecture (white paper)
- [11] Bluetooth Assigned Numbers



## Part K:2

# SERVICE DISCOVERY APPLICATION PROFILE



This document defines the features and procedures for an application in a Bluetooth device to discover services registered in other Bluetooth devices and retrieve any desired available information pertinent to these services.



**CONTENTS**

<b>1</b>	<b>Introduction</b> .....	<b>66</b>
1.1	Scope .....	66
1.2	Symbols and conventions .....	67
<b>2</b>	<b>Profile overview</b> .....	<b>68</b>
2.1	Profile stack .....	68
2.2	Configurations and roles .....	69
2.3	User requirements and scenarios .....	70
2.4	Profile fundamentals .....	71
2.5	Conformance .....	71
<b>3</b>	<b>User interface aspects</b> .....	<b>72</b>
3.1	Pairing .....	72
3.2	Mode selection .....	72
<b>4</b>	<b>Application layer</b> .....	<b>73</b>
4.1	The service discovery application .....	73
4.2	Service primitives abstractions .....	75
4.3	Message sequence charts (MSCs) .....	77
<b>5</b>	<b>Service Discovery</b> .....	<b>79</b>
5.1	An SDP PDU exchange example .....	80
<b>6</b>	<b>L2CAP</b> .....	<b>82</b>
6.1	Channel types .....	83
6.2	Signalling .....	83
6.3	Configuration options .....	83
6.3.1	Maximum Transmission Unit (MTU) .....	83
6.3.2	Flush Time-out .....	83
6.3.3	Quality of Service .....	84
6.4	SDP transactions and L2CAP connection lifetime .....	84
<b>7</b>	<b>Link Manager</b> .....	<b>86</b>
7.1	Capability overview .....	86
7.2	Error behavior .....	87
7.3	Link policy .....	87
<b>8</b>	<b>Link control</b> .....	<b>88</b>
8.1	Capability overview .....	88
8.2	Inquiry .....	89
8.3	Inquiry scan .....	90
8.4	Paging .....	90
8.5	Page scan .....	90
8.6	Error behavior .....	90

9	References.....	91
	9.1 Normative references .....	91
10	Definitions .....	92
11	Appendix A (Informative): Service primitives and the Bluetooth PDUs.....	93

## FOREWORD

---

Interoperability between devices from different manufacturers is provided for a specific service and use case, if the devices conform to a Bluetooth SIG-defined profile specification. A profile defines a selection of messages and procedures (generally termed *capabilities*) from the Bluetooth SIG specifications, and gives an unambiguous description of the air interface for specified service(s) and use case(s).

All defined features are process-mandatory. This means that, if a feature is used, it is used in a specified manner. Whether the provision of a feature is mandatory or optional is stated separately for both sides of the Bluetooth air interface.

## 1 INTRODUCTION

---

### 1.1 SCOPE

It is expected that the number of services that can be provided over Bluetooth links will increase in an undetermined (and possibly uncontrolled) manner. Therefore, procedures need to be established to aid a user of a Bluetooth-enabled device to sort the ever-increasing variety of services that will become available to him/her. While many of the Bluetooth-enabled services that may be encountered are currently unknown, a standardized procedure can still be put into place on how to locate and identify them.

The Bluetooth protocol stack contains a Service Discovery Protocol (SDP) [BT\\_SDP\\_spec:\[7\]](#) that is used to locate services that are available on or via devices in the vicinity of a Bluetooth enabled device. Having located what services are available in a device, a user may then select to use one or more of them. Selecting, accessing, and using a service is outside the scope of this document. Yet, even though SDP is not directly involved in accessing services, information retrieved via SDP facilitates service access by using it to properly condition the local Bluetooth stack to access the desired service.

The service discovery profile defines the protocols and procedures that shall be used by a service discovery application on a device to locate services in other Bluetooth-enabled devices using the Bluetooth Service Discovery Protocol (SDP). With regard to this profile, the service discovery application is a specific user-initiated application. In this aspect, this profile is in contrast to other profiles where service discovery interactions between two SDP entities in two Bluetooth-enabled devices result from the need to enable a particular transport service (e.g. RFCOMM, etc.), or a particular usage scenario (e.g. file transfer, cordless telephony, LAN AP, etc.) over these two devices. Service discovery interactions of the latter kind can be found within the appropriate Bluetooth usage scenario profile documents.

The service discovery in the other profile documents has a very narrow scope; e.g. learning about the protocols and related protocol parameters needed for accessing a particular service. Nevertheless, the fundamentals of the service discovery procedures covered in this profile document, and the use of the Bluetooth protocols in support of these procedures can be replicated in other profile documents as well. The only difference is that for the other profiles these procedures are initiated by application-level actions within the applications described by the corresponding profiles, as opposed to user-level actions for this profile.

SDP provides direct support for the following set of service inquiries:

- Search for services by service class;
- Search for services by service attributes; and
- Service browsing.

The generic service discovery application considered for this profile also covers the above service inquiry scenarios.

The former two cases represent searching for known and specific services. They provide answers to user questions like: "Is service A, or is service A with characteristics B and C, available?" The latter case represents a general service search and provides answers to questions like: "What services are available?" or "What services of type A are available?"

The above service inquiry scenarios can be realized two-fold:

- By performing the service searches on a particular device that a user 'consciously' has already connected to, and/or
- By performing the service searches by 'unconsciously' connecting to devices discovered in a device's vicinity.

Both of the above approaches require that devices need first to be discovered, then linked with, and then inquired about the services they support.

## 1.2 SYMBOLS AND CONVENTIONS

This profile uses the symbols and conventions specified in Section 1.2 of the Generic Access Profile [3].

## 2 PROFILE OVERVIEW

### 2.1 PROFILE STACK

Figure 2.1 shows the Bluetooth protocols and supporting entities involved in this profile.

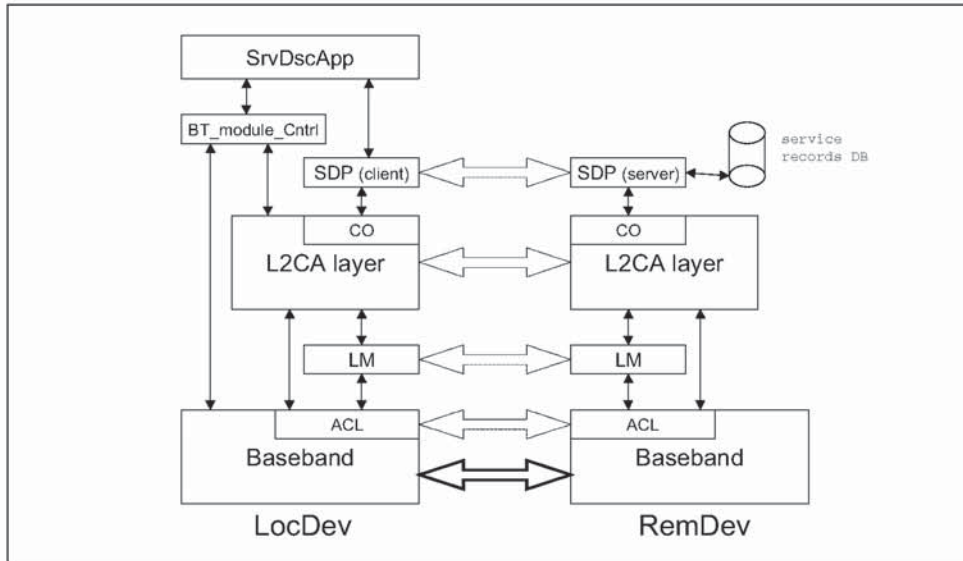


Figure 2.1: The Bluetooth protocol stack for the service discovery profile

The service discovery user application (SrvDscApp) in a local device (LocDev) interfaces with the Bluetooth SDP client to send service inquiries and receive service inquiry responses from the SDP servers of remote devices (RemDevs) BT\_SDP\_spec:[7]. SDP uses the connection-oriented (CO) transport service in L2CAP, which in turn uses the baseband asynchronous connectionless (ACL) links to ultimately carry the SDP PDUs over the air.

Service discovery is tightly related to discovering devices, and discovering devices is tightly related to performing inquiries and pages. Thus, the SrvDscApp interfaces with the baseband via the BT\_module\_Cntrl entity that instructs the Bluetooth module when to enter various search modes of operation.<sup>1</sup>

1. The BT\_module\_Cntrl may be part of a Bluetooth stack implementation (and thus be shared by many Bluetooth-aware applications) or a 'lower part' of the SrvDscApp. Since, no assumptions about any particular stack or SrvDscApp implementations are made, the BT\_module\_Cntrl entity represents a logical entity separate from the SrvDscApp, which may or may not be part of the SrvDscApp itself, a stack component, or any other appropriate piece of code.



The service records database (DB) shown in Figure 2.1 next to an SDP server is a logical entity that serves as a repository of service discovery-related information. The 'physical form' of this database is an implementation issue outside the scope of this profile.

## 2.2 CONFIGURATIONS AND ROLES

The following roles are defined in this profile:

- **Local device (LocDev):** A LocDev is the device that initiates the service discovery procedure. A LocDev must contain at least the *client* portion of the Bluetooth SDP architecture BT\_SDP\_spec:[7]. A LocDev contains the service discovery application (SrvDscApp) used by a user to initiate discoveries and display the results of these discoveries.
- **Remote Device(s) (RemDev(s)):** A RemDev is any device that participates in the service discovery process by responding to the service inquiries generated by a LocDev. A RemDev must contain at least the *server* portion of the Bluetooth SDP architecture BT\_SDP\_spec:[7]. A RemDev contains a service records database, which the server portion of SDP consults to create responses to service discovery requests.

The LocDev or RemDev role assigned to a device is neither permanent nor exclusive. A RemDev may also have a SrvDscApp installed into it as well as an SDP client, and a LocDev may also have an SDP server. In conjunction with which device has an SrvDscApp installed, an SDP-client installed, and an SDP-server installed, the assignment of devices to the above roles is relative to each individual SDP (and related) transaction and which device initiates the transaction. Thus, a device could be a LocDev for a particular SDP transaction, while at the very same time be a RemDev for another SDP transaction.

With respect to this profile, a device without a UI (directly or indirectly available) for entering user input and returning the results of service searches is not considered as a candidate for a LocDev. Nevertheless, even if such a device is not considered as a candidate for a LocDev, the procedures presented in the following sections can still apply if applications running in such a device need to execute a service discovery transaction.

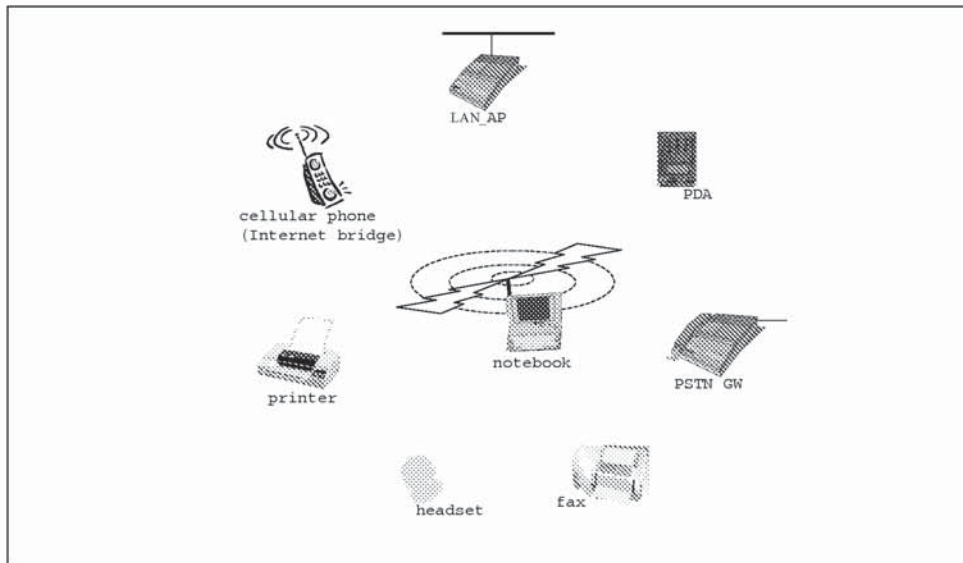


Figure 2.2: A typical service discovery scenario

The figure above shows a local device (the notebook) inquiring for services among a plethora of remote devices.

## 2.3 USER REQUIREMENTS AND SCENARIOS

The scenarios covered by this profile are the following:

- Search for services by service class,
- Search for services by service attributes, and
- Service browsing.

The first two cases represent searching for known and specific services, as part of the user question "Is service A, or is service A with characteristics B and C, available?" The latter case represents a general service search that is a response to the user question "What services are available?"

This profile implies the presence of a Bluetooth-aware, user-level application, the SrvDscApp, in a LocDev that interfaces with the SDP protocol for locating services. In this aspect, this profile is unique as compared to other profiles. It is a profile that describes an application that interfaces to a specific Bluetooth protocol to take full advantage of it for the direct benefit of an end-user.

## 2.4 PROFILE FUNDAMENTALS

Before any two Bluetooth-equipped devices can communicate with each other the following may be needed:

- The devices need to be powered-on and initialized. Initialization may require providing a PIN for the creation of a link key, for device authorization and data encryption.
- A Bluetooth link has to be created, which may require the discovery of the other device's BD\_ADDR via an inquiry process, and the paging of the other device.

While it may seem natural to consider a LocDev serving as a Bluetooth master and the RemDev(s) serving as Bluetooth slave(s), there is no such requirement imposed on the devices participating in this profile. Service discovery as presented in this document can be initiated by either a master or a slave device at any point for which these devices are members of the same piconet. Also, a slave in a piconet may possibly initiate service discovery in a new piconet, provided that it notifies the master of the original piconet that it will be unavailable (possibly entering the hold operational mode) for a given amount of time.<sup>2</sup>

The profile does not require the use of authentication and/or encryption. If any of these procedures are used by any of the devices involved, service discovery will be performed only on the subset of devices that pass the authentication and encryption security 'roadblocks' that may impose to each other. In other words, any security restrictions for SDP transactions are dictated by the security restrictions already in place (if any) on the Bluetooth link.

## 2.5 CONFORMANCE

If conformance to this profile is claimed, all capabilities indicated mandatory for this profile shall be supported in the specified manner (process-mandatory). This also applies to all optional and conditional capabilities for which support is indicated. All mandatory capabilities, and optional and conditional capabilities for which support is indicated, are subject to verification as part of the Bluetooth certification program.

---

2. Recall that a master of a piconet cannot initiate a new piconet. Since a piconet is ultimately identified by the BD\_ADDR and the Bluetooth clock of its master, the latter piconet will be identical to and indistinguishable from the former.

## 3 USER INTERFACE ASPECTS

---

### 3.1 PAIRING

No particular requirements regarding pairing are imposed by this profile. Pairing may or may not be performed. Whenever a LocDev performs service discovery against as yet 'unconnected' RemDev(s), it shall be the responsibility of the SrvDscApp to allow pairing prior to connection, or to by-pass any devices that may require pairing first. This profile is focused on only performing service discovery whenever the LocDev can establish a legitimate and useful baseband link<sup>3</sup> with RemDev(s).

### 3.2 MODE SELECTION

This profile assumes that, under the guidance of the SrvDscApp, the LocDev shall be able to enter the inquiry and/or page states. It is also assumed that a RemDev with services that it wants to make available to other devices (e.g. printer, a LAN DAP, a PSTN gateway, etc.) shall be able to enter the inquiry scan and/or page scan states. For more information about the inquiry and page related states see Section 8.

Since the SrvDscApp may also perform service inquiries against already connected RemDevs, it is not mandatory according to the profile that a LocDev always be the master of a connection with a RemDev. Similarly, a RemDev may not always be the slave of a connection with a LocDev.

---

3. A legitimate and useful baseband link is a Bluetooth baseband link that is properly authenticated and encrypted (if so desired), whenever any of these options are activated by any of the devices participating in this profile.

## 4 APPLICATION LAYER

### 4.1 THE SERVICE DISCOVERY APPLICATION

In this subsection, the operational framework of the SrvDscApp is presented.<sup>4</sup> Figure 4.1 shows alternative possibilities for a SrvDscApp.

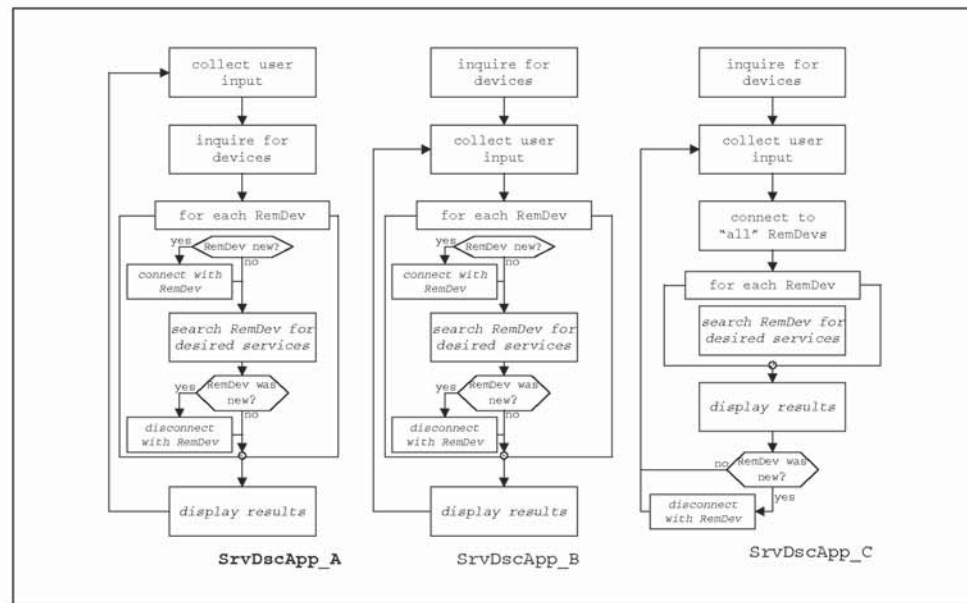


Figure 4.1: Three possible SrvDscApps

The SrvDscApp alternatives shown in Figure 4.1, which are not exhaustive by any means, achieve the same objectives but they follow different paths for achieving them. In the first alternative (SrvDscApp\_A), the SrvDscApp on a LocDev inquires its user to provide information for the desired service search. Following this, the SrvDscApp searches for devices, via the Bluetooth inquiry procedure. For each device found, the LocDev will connect to it, perform any necessary link set-up, see related procedures in Generic Access Profile [3], and then inquire it for the desired services. In the second alternative (SrvDscApp\_B), the inquiry of devices is done prior to collecting user input for the service search.<sup>5</sup>

4. This profile does not dictate any particular implementation for a SevDisApp. It only presents the procedures needed to achieve its objectives.  
 5. Device inquiries may even occur by means outside the scope of a particular SrvDscApp implementation. But, since such other means are not guaranteed to exist, it is recommended that the SrvDscApp activates device inquiries too.