Since a slave can synchronize to any packet which is preceded by the proper channel access code, the packets carried on the beacon slots do not have to contain specific broadcast packets for parked slaves to be able to synchronize; any packet may be used. The only requirement placed on the beacon slots is that there is a master-to-slave transmission present on the hopping sequence associated with the park structure. If there is no information to be sent, **NULL** packets may be transmitted by the master. If there is indeed broadcast information to be sent to the parked slaves, the first packet of the broadcast message shall be repeated in every beacon slot of the beacon train. However, synchronous traffic in the synchronous reserved slots may interrupt the beacon transmission if it is on the same hopping sequence as the parked slaves. The master shall configure its park beacon structure so that reserved slots of synchronous logical transports do not cause slaves to miss synchronization on a beacon slot. For example, a master that has active slaves using AHS, and parked slaves using Non-AHS shall ensure that the Park beacons cannot be interrupted by AHS synchronous reserved slots.

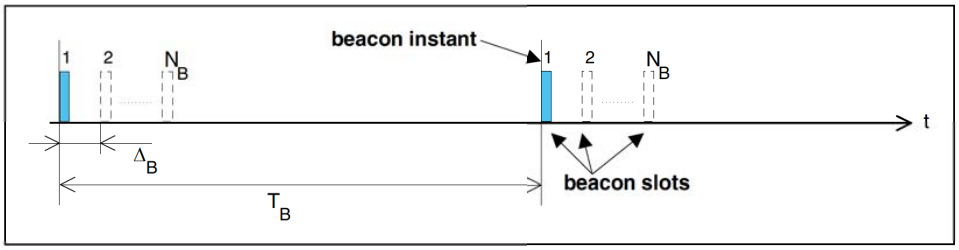


Figure 8.13: General beacon train format

The master can place parked slaves in any of the AFH operating modes, but shall ensure that all parked slaves use the same hop sequence. Masters should use AHS(79) or AHS when all the slaves in the Piconet are AFH capable.

A master that switches a slave between AFH enabled, AFH disabled or AHS(79) operation shall ensure that the AFH_Instant occurs before transmission of the beacon train using this hop sequence.

The master communicates with parked slaves using broadcast messages. Since these messages can be time - critical, an ongoing repetition train of broadcast message may be prematurely aborted by broadcast information destined to parked slaves in beacon slots and in access windows (see Section 8.9.2 on page 189).

8.9.2 Beacon access window

In addition to the beacon slots, an access window is defined where the parked slaves can send requests to be unparked. To increase reliability, the access window may be repeated M_{access} times ($M_{access} \geq 1$), see Figure 8.14 on page 189. The access window starts a fixed delay D_{access} after the beacon instant. The width of the access window is T_{access}.

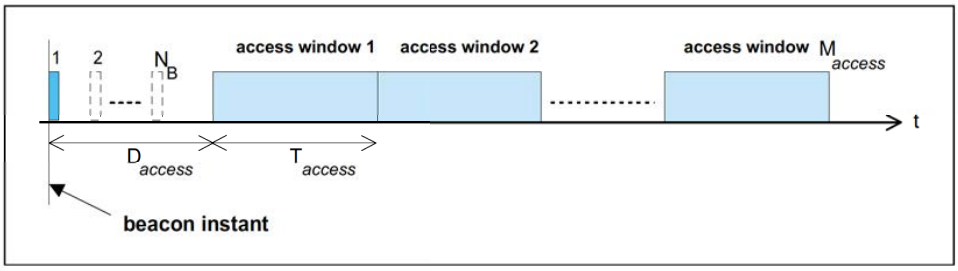


Figure 8.14: Definition of access window

The access window supports a polling slave access technique. The format of the polling technique is shown in Figure 8.15 on page 189. The same TDD structure is used as on the piconet channel, i.e. master-to-slave transmission is alternated with slave-to-master transmission. The slave-to-master slot is divided into two half slots of 312.5 µs each. The half slot a parked slave is allowed to respond in corresponds to its access request address (AR_ADDR), see also section 8.9.6 on page 192. For counting the half slots to determine the access request slot, the start of the access window is used, see Figure 8.15 on page 189. The slave shall only send an access request in the proper slave-to-master half slot if a broadcast packet has been received in the preceding master-to-slave slot. In this way, the master polls the parked slaves.

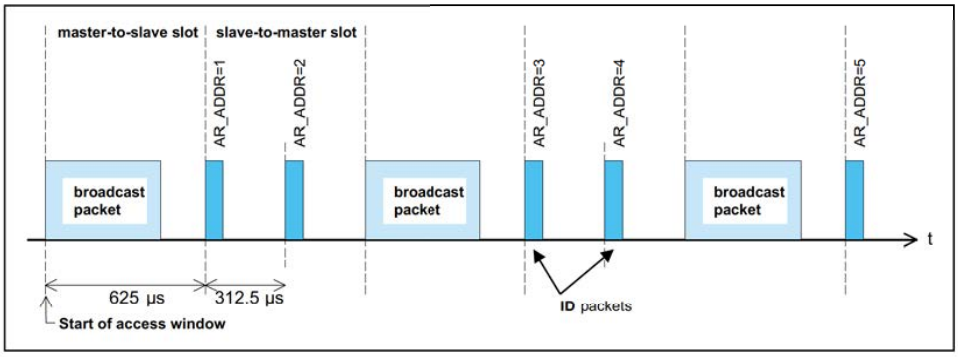


Figure 8.15: Access procedure applying the polling technique.

The slots of the access window may also be used for traffic on the piconet if required. For example, if a synchronous connection has to be supported, the slots reserved for the synchronous link may carry synchronous information

instead of being used for access requests, i.e. if the master-to-slave slot in the access window contains a packet different from a broadcast packet, the following slave-to-master slot shall not be used for slave access requests. If the master transmits a broadcast packet in the access window then it shall use the hop sequence associated with the park structure. Slots in the access window not affected by piconet traffic may still be used according to the defined access structure, (an example is shown in Figure 8.16 on page 190) the access procedure shall be continued as if no interruption had taken place.

When the slave is parked, the master shall indicate what type of access scheme will be used. For the polling scheme, the number of slave-to-master access slots N_{acc_slot} is indicated.

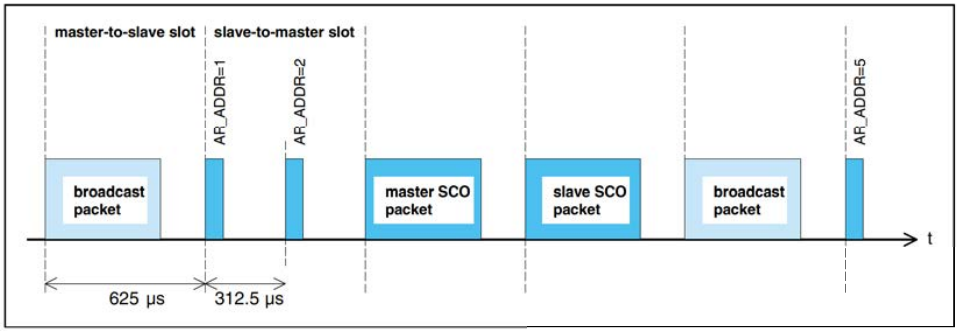


Figure 8.16: Disturbance of access window by SCO traffic

By default, the access window is always present. However, its activation depends on the master sending broadcast messages to the slave at the appropriate slots in the access window. A flag in a broadcast LMP message within the beacon slots may indicate that the access window(s) belonging to this instant will not be activated. This prevents unnecessary scanning of parked slaves that want to request access.

8.9.3 Parked slave synchronization

Parked slaves wake up periodically to re-synchronize to the channel. Any packet exchanged on the channel can be used for synchronization. Since master transmission is mandatory on the beacon slots, parked slaves will use the beacon train to re-synchronize. A parked slave may wake-up at the beacon instant to read the packet sent on the first beacon slot. If this fails, it may retry on the next beacon slot in the beacon train; in total, there are N_B opportunities per beacon instant to re-synchronize. During the search, the slave may increase its search window, see also Section 2.2.5.2 on page 74. The separation between the beacon slots in the beacon train Δ_B shall be chosen such that consecutive search windows will not overlap.

The parked slave may not wake up at every beacon instant. Instead, a sleep interval may be applied which is longer than the beacon interval T_B, see Figure 8.17 on page 191. The slave sleep window shall be a multiple N_{B_sleep} of T_B.

The precise beacon instant the slave may wake up on shall be indicated by the master with D_{B_sleep} which indicates the offset (in multiples of T_B) with respect to the beacon instant ($0 < D_{B\_sleep} < N_{B\_sleep}-1$). To initialize the wake-up period, the applicable equation shall be used:

$CLK_{27-1} \bmod (N_{B\_sleep} \bullet T_B) = D_B + D_{B\_sleep} \bullet T_B$ for initialization 1

$(\overline{CLK_{27}}, CLK_{26-1}) \bmod (N_{B\_sleep} \bullet T_B) = D_B + D_{B\_sleep} \bullet T_B$ for initialization 2

where initialization 1 shall be chosen by the master if the MSB in the current master clock is 0 and initialization 2 shall be chosen by the master if the MSB in the current master clock is 1.

When the master needs to send broadcast messages to the parked slaves, it may use the beacon slots for these broadcast messages. However, if $N_B < N_{BC}$, the slots following the last beacon slot in the beacon train shall be used for the remaining $N_{BC}-N_B$ broadcast packets. If $N_B > N_{BC}$, the broadcast message shall be repeated on all N_B beacon slots.

A parked slave shall read the broadcast messages sent in the beacon slot(s) it wakes up in. If the parked slave wakes up, the minimum wake-up activity shall be to read the channel access code for re-synchronization and the packet header to check for broadcast messages.

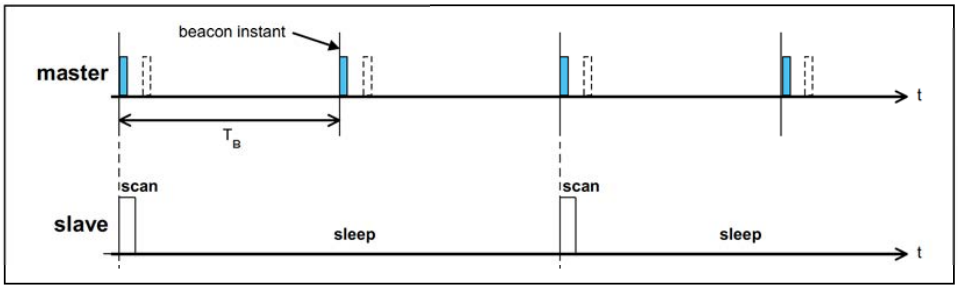


Figure 8.17: Extended sleep interval of parked slaves.

8.9.4 Parking

A master can park an active slave through the exchange of LMP commands. Before being put into park, the slave shall be assigned a PM_ADDR and an AR_ADDR. Every parked slave shall have a unique PM_ADDR or a PM_ADDR of 0. The AR_ADDR is not necessarily unique. The beacon parameters shall be given by the master when the slave is parked. The slave shall then give up its LT_ADDR and shall enter **PARK** state. A master can park only a single slave at a time. The park message is carried with a normal data packet and addresses the slave through its LT_ADDR.

8.9.5 Master-initiated unparking

The master can unpark a parked slave by sending a dedicated LMP unpark command including the parked slave's address. This message shall be sent in a broadcast packet on the beacon slots. The master shall use either the slave's PM_ADDR, or its BD_ADDR. The message also includes the logical transport address LT_ADDR the slave shall use after it has re-entered the piconet. The unpark message may include a number of slave addresses so that multiple slaves may be unparked simultaneously. For each slave, a different LT_ADDR shall be assigned.

After having received the unpark message, the parked slave matching the PM_ADDR or BD_ADDR shall leave the **PARK** state and enter the **CONNECTION** state. It shall keep listening to the master until it is addressed by the master through its LT_ADDR. The first packet sent by the master shall be a POLL packet. The return packet in response to the POLL packet confirms that the slave has been unparked. If no response packets from the slave is received for *newconnectionTO* number of slots after the end of beacon repetition period, the master shall unpark the slave again. The master shall use the same LT_ADDR on each unpark attempt until it has received a link supervision timeout for that slave or the unpark has completed successfully. If the slave does not receive the POLL packet for *newconnectionTO* number of slots after the end of beacon repetition period, it shall return to park, with the same beacon parameters. After confirming that the slave is in the **CONNECTION** state, the master decides in which mode the slave will continue.

When a device is unparked, the SEQN bit for the link shall be reset to 1 on both the master and the slave (see Section 7.6.2.1 on page 148).

8.9.6 Slave-initiated unparking

A slave can request access to the channel through the access window defined in section 8.9.2 on page 189. As shown in Figure 8.15 on page 189, the access window includes several slave-to-master half slots where the slave may send an access request message. The specific half slot the slave is allowed to respond in, corresponds to its access request address (AR_ADDR) which it received when it was parked. The order of the half slots (in Figure 8.15 the AR_ADDR numbers linearly increase from 1 to 5) is not fixed: an LMP command sent in the beacon slots may reconfigure the access window. When a slave desires access to the channel, it shall send an access request message in the proper slave-to-master half slot. The access request message of the slave is the **ID** packet containing the device access code (DAC) of the master (which is the channel access code without the trailer). The parked slave shall only transmit an access request message in the half slot, when in the preceding master-to-slave slot a broadcast packet has been received. This broadcast message may contain any kind of broadcast information not necessarily related to the parked slave(s). If no broadcast information is available, a broadcast **NULL** or broadcast **POLL** packet shall be sent.

After having sent an access request, the parked slave shall listen for an unpark message from the master. As long as no unpark message is received, the slave shall repeat the access requests in the subsequent access windows. After the last access window (there are M_{access} windows in total, see Section 8.9.2 on page 189), the parked slave shall listen for an additional N_{poll} time slots for an unpark message. If no unpark message is received within N_{poll} slots after the end of the last access window, the slave may return to sleep and retry an access attempt after the next beacon instant.

After having received the unpark message, the parked slave matching the PM_ADDR or BD_ADDR shall leave the **PARK** state and enter the **CONNECTION** state. It shall keep listening to the master until it is addressed by the master through its LT_ADDR. The first packet sent by the master shall be a POLL packet. The return packet in response to the POLL packet confirms that the slave has been unparked. After confirming that the slave is in the **CONNECTION** state, the master decides in which mode the slave will continue. If no response packet from the slave is received for *newconnectionTO* number of slots after N_{poll} slots after the end of the last access window, the master shall send the unpark message to the slave again. If the slave does not receive the POLL packet for *newconnectionTO* number of slots after N_{poll} slots after the end of the last access window, it shall return to park, with the same beacon parameters.

When a device is unparked, the SEQN bit for the link shall be reset to 1 on both the master and the slave (see Section 7.6.2.1 on page 148).

8.9.7 Broadcast scan window

In the beacon train, the master can support broadcast messages to the parked slaves. However, it may extend its broadcast capacity by indicating to the parked slaves that more broadcast information is following after the beacon train. This is achieved by an LMP command ordering the parked slaves (as well as the active slaves) to listen to the channel for broadcast messages during a limited time window. This time window starts at the beacon instant and continues for the period indicated in the LMP command sent in the beacon train.

8.9.8 Polling in the park state

In the **PARK** state, parked slaves may send access requests in the access window provided a broadcast packet is received in the preceding master-to-slave slot. Slaves in the **CONNECTION** state shall not send in the slave-to-master slots following the broadcast packet, since they are only allowed to send if addressed specifically.

9 AUDIO

On the air-interface, either a 64 kb/s log PCM (Pulse Code Modulation) format (A-law or μ-law) may be used, or a 64 kb/s CVSD (Continuous Variable Slope Delta Modulation) may be used. The latter format applies an adaptive delta modulation algorithm with syllabic companding.

The voice coding on the line interface is designed to have a quality equal to or better than the quality of 64 kb/s log PCM.

Table 9.1 on page 195 summarizes the voice coding schemes supported on the air interface. The appropriate voice coding scheme is selected after negotiations between the Link Managers.

Voice Codecs	
linear	CVSD
8-bit logarithmic	A-law
	μ-law

Table 9.1: Voice coding schemes supported on the air interface.

9.1 LOG PCM CODEC

Since the synchronous logical transports on the air-interface can support a 64 kb/s information stream, a 64 kb/s log PCM traffic can be used for transmission. Either A-law or μ-law compression may be applied. In the event that the line interface uses A-law and the air interface uses μ-law or vice versa, a conversion from A-law to μ-law shall be performed. The compression method shall follow ITU-T recommendations G. 711.

9.2 CVSD CODEC

A more robust format for voice over the air interface is delta modulation. This modulation scheme follows the waveform where the output bits indicate whether the prediction value is smaller or larger then the input waveform. To reduce slope overload effects, syllabic companding is applied: the step size is adapted according to the average signal slope. The input to the CVSD encoder shall be 64 ksamples/s linear PCM (typically 16 bits, but actual value is implementation specific). Block diagrams of the CVSD encoder and CVSD decoder are shown in Figure 9.1 on page 196, Figure 9.2 on page 196 and Figure 9.3 on page 196. The system shall be clocked at 64 kHz.

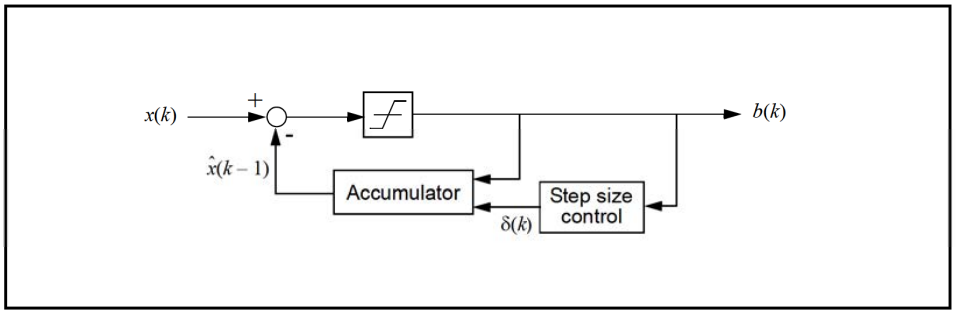


Figure 9.1: Block diagram of CVSD encoder with syllabic companding.

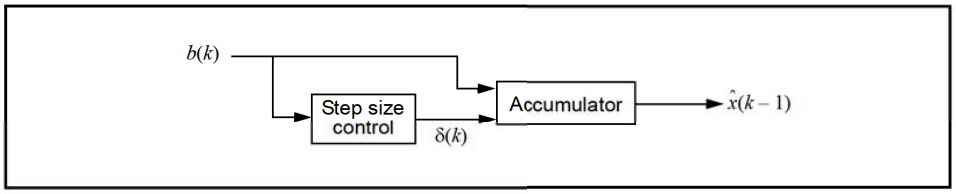


Figure 9.2: Block diagram of CVSD decoder with syllabic companding.

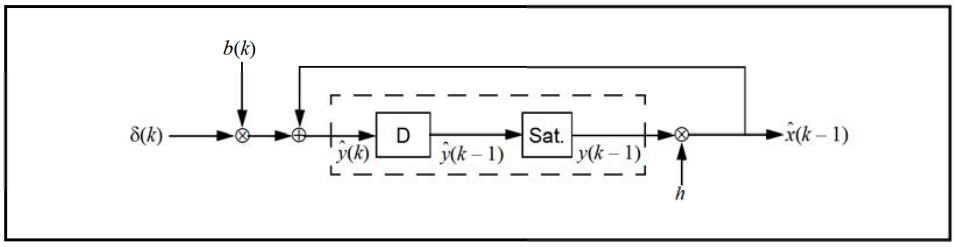


Figure 9.3: Accumulator procedure.

Let $\mathrm{sgn}(x) = 1$ for $x \geq 0$, otherwise $\mathrm{sgn}(x) = -1$. On air these numbers shall be represented by the sign bit; i.e. negative numbers are mapped on “1” and positive numbers are mapped on “0”.

Denote the CVSD encoder output bit $b(k)$, the encoder input $x(k)$, the accumulator contents $y(k)$, and the step size $\delta(k)$. Furthermore, let h be the decay factor for the accumulator, let β denote the decay factor for the step size, and, let α be the syllabic companding parameter. The latter parameter monitors the slope by considering the K most recent output bits

Let

$$\hat{x}(k) = hy(k). \qquad \text{(EQ 13)}$$

Then, the CVSD encoder internal state shall be updated according to the following set of equations:

$$b(k) = \mathrm{sgn}\{x(k) - \hat{x}(k-1)\}, \quad \text{(EQ 14)}$$

$$\alpha = \begin{cases} 1, & \text{if } J \text{ bits in the last } K \text{ output bits are equal,} \\ 0, & \text{otherwise,} \end{cases} \quad \text{(EQ 15)}$$

$$\delta(k) = \begin{cases} \min\{\delta(k-1) + \delta_{min}, \delta_{max}\}, & \alpha = 1, \\ \max\{\beta\delta(k-1), \delta_{min}\}, & \alpha = 0, \end{cases} \quad \text{(EQ 16)}$$

$$y(k) = \begin{cases} \min\{\hat{y}(k), y_{max}\}, & \hat{y}(k) \geq 0. \\ \max\{\hat{y}(k), y_{min}\}, & \hat{y}(k) < 0. \end{cases} \quad \text{(EQ 17)}$$

where

$$\hat{y}(k) = \hat{x}(k-1) + b(k)\delta(k). \quad \text{(EQ 18)}$$

In these equations, δ_{min} and δ_{max} are the minimum and maximum step sizes, and, y_{min} and y_{max} are the accumulator's negative and positive saturation values, respectively. Over air, the bits shall be sent in the same order they are generated by the CVSD encoder.

For a 64 kb/s CVSD, the parameters as shown in Table 9.2 shall be used. The numbers are based on a 16 bit signed number output from the accumulator. These values result in a time constant of 0.5 ms for the accumulator decay, and a time constant of 16 ms for the step size decay

Parameter	Value
h	$1 - \frac{1}{32}$
β	$1 - \frac{1}{1024}$
J	4
K	4
δ_{min}	10
δ_{max}	1280
y_{min}	-2^{15} or $-2^{15} + 1$
y_{max}	$2^{15} - 1$

Table 9.2: CVSD parameter values. The values are based on a 16 bit signed number output from the accumulator.

9.3 ERROR HANDLING

In the DV, HV3, EV3, EV5, 2-EV3, 3-EV3, 2-EV5 and 3-EV5 packets, the voice is not protected by FEC. The quality of the voice in an error-prone environment then depends on the robustness of the voice coding scheme and, in the case of eSCO, the retransmission scheme. CVSD, in particular, is rather insensitive to random bit errors, which are experienced as white background noise. However, when a packet is rejected because either the channel access code, the HEC test was unsuccessful, or the CRC has failed, measures have to be taken to "fill" in the lost speech segment.

The voice payload in the **HV2** and **EV4** packets are protected by a 2/3 rate FEC. For errors that are detected but cannot be corrected, the receiver should try to minimize the audible effects. For instance, from the 15-bit FEC segment with uncorrected errors, the 10-bit information part as found before the FEC decoder should be used. The **HV1** packet is protected by a 3 bit repetition FEC. For this code, the decoding scheme will always assume zero or one-bit errors. Thus, there exist no detectable but uncorrectable error events for **HV1** packets.

9.4 GENERAL AUDIO REQUIREMENTS

9.4.1 Signal levels

For A-law and μ-law log-PCM encoded signals the requirements on signal levels shall follow the ITU-T recommendation G.711.

Full swing at the 16 bit linear PCM interface to the CVSD encoder is defined to be 3 dBm0.

9.4.2 CVSD audio quality

For Bluetooth audio quality the requirements are put on the transmitter side. The 64 ksamples/s linear PCM input signal should have negligible spectral power density above 4 kHz. The power spectral density in the 4-32 kHz band of the decoded signal at the 64 ksample/s linear PCM output, should be more than 20 dB below the maximum in the 0-4 kHz range.

10 LIST OF FIGURES

Figure 1.1: Piconets with a single slave operation (a), a multi-slave operation (b) and a scatternet operation (c). 61
Figure 1.2: Standard Basic Rate packet format. 62
Figure 1.3: Standard Enhanced Data Rate packet format 62
Figure 1.4: Bluetooth clock. 63
Figure 1.5: Format of BD_ADDR. 64
Figure 2.1: Multi-slot packets 69
Figure 2.2: Derivation of CLK in master (a) and in slave (b). 70
Figure 2.3: RX/TX cycle of master transceiver in normal mode for single-slot packets. 71
Figure 2.4: RX/TX cycle of slave transceiver in normal mode for single-slot packets. 72
Figure 2.5: RX timing of slave returning from hold mode. 73
Figure 2.6: Derivation of CLKE. 74
Figure 2.7: RX/TX cycle of transceiver in PAGE mode. 75
Figure 2.8: Timing of page response packets on successful page in first half slot 76
Figure 2.9: Timing of page response packets on successful page in second half slot 77
Figure 2.10: Timing of inquiry response packet on successful inquiry in first half slot 79
Figure 2.11: Timing of inquiry response packet on successful inquiry in second half slot 79
Figure 2.12: General block diagram of hop selection scheme. 81
Figure 2.13: Hop selection scheme in CONNECTION state. 82
Figure 2.14: Single- and multi-slot packets. 83
Figure 2.15: Example of the same channel mechanism. 83
Figure 2.16: Block diagram of the basic hop selection kernel for the hop system. 84
Figure 2.17: XOR operation for the hop system. 85
Figure 2.18: Permutation operation for the hop system. 86
Figure 2.19: Butterfly implementation. 86
Figure 2.20: Block diagram of adaptive hop selection mechanism 88
Figure 4.1: Functional diagram of TX buffering. 97
Figure 4.2: Functional diagram of RX buffering 101
Figure 6.1: General Basic Rate packet format. 107
Figure 6.2: General enhanced data rate packet format 107
Figure 6.3: Access code format 109
Figure 6.4: Preamble 110
Figure 6.5: Construction of the sync word. 111
Figure 6.6: LFSR and the starting state to generate 113

Figure 6.7: Trailer in CAC when MSB of sync word is 0 (a), and when MSB of sync word is 1 (b). 113
Figure 6.8: Header format. 114
Figure 6.9: Format of the FHS payload. 118
Figure 6.10: DV packet format 121
Figure 6.11: Synchronization sequence 127
Figure 6.12: Payload header format for Basic Rate single-slot ACL packets. 128
Figure 6.13: Payload header format for multi-slot ACL packets and all EDR ACL packets. 129
Figure 7.1: Header bit processes. 135
Figure 7.2: Payload bit processes. 135
Figure 7.3: The LFSR circuit generating the HEC. 136
Figure 7.4: Initial state of the HEC generating circuit. 137
Figure 7.5: HEC generation and checking. 137
Figure 7.6: The LFSR circuit generating the CRC. 138
Figure 7.7: Initial state of the CRC generating circuit. 138
Figure 7.8: CRC generation and checking. 138
Figure 7.9: Data whitening LFSR. 139
Figure 7.10: Bit-repetition encoding scheme. 140
Figure 7.11: LFSR generating the (15,10) shortened Hamming code. 141
Figure 7.12: Stage 1 of the receive protocol for determining the ARQN bit. 143
Figure 7.13: Stage 2 (ACL) of the receive protocol for determining the ARQN bit. 144
Figure 7.14: Stage 2 (eSCO) of the receive protocol for determining the ARQN bit. 145
Figure 7.15: Transmit filtering for packets with CRC. 146
Figure 7.16: Broadcast repetition scheme 150
Figure 8.1: State diagram of link controller. 151
Figure 8.2: Conventional page (a), page while one synchronous link present (b), page while two synchronous links present (c). 156
Figure 8.3: Messaging at initial connection when slave responds to first page message. 158
Figure 8.4: Messaging at initial connection when slave responds to second page message. 158
Figure 8.5: Connection state. 166
Figure 8.6: RX/TX timing in multi-slave configuration 167
Figure 8.7: eSCO Window Details for Single-Slot Packets 170
Figure 8.8: eSCO Window Details for Asymmetric Traffic 171
Figure 8.9: Successful hop sequence switching 176
Figure 8.10: Recovery hop sequences 177
Figure 8.11: AFH_Instant changes during multi-slot packets transmitted by the master. 178

Figure 8.12: Sniff anchor points 181
Figure 8.13: General beacon train format 186
Figure 8.14: Definition of access window 187
Figure 8.15: Access procedure applying the polling technique. 187
Figure 8.16: Disturbance of access window by SCO traffic 188
Figure 8.17: Extended sleep interval of parked slaves. 189
Figure 9.1: Block diagram of CVSD encoder with syllabic companding. ...194
Figure 9.2: Block diagram of CVSD decoder with syllabic companding. ...194
Figure 9.3: Accumulator procedure. 194
Figure 12.1: SLR measurement set-up. 203
Figure 12.2: RLR measurement set-up. 203
Figure 12.3: Plot of recommended frequency mask for Bluetooth. The GSM send frequency mask is given for comparison (dotted line) ...204
Figure 12.4: Timing constraint on AFH_Instant with slaves in park, hold and sniff 208

11 LIST OF TABLES

Table 2.1: Control of the butterflies for the hop system 85
Table 2.2: Control for hop system. 89
Table 6.1: Summary of access code types. 109
Table 6.2: Packets defined for synchronous and asynchronous logical transport types. 117
Table 6.3: Description of the FHS payload 119
Table 6.4: Contents of SR field 120
Table 6.5: Contents of page scan mode field....120
Table 6.6: Logical link LLID field contents....130
Table 6.7: Use of payload header flow bit on the logical links. 131
Table 6.8: Link control packets 132
Table 6.9: ACL packets....132
Table 6.10: Synchronous packets....133
Table 8.1: Relationship between scan interval, and paging modes R0, R1 and R2. 153
Table 8.2: Relationship between train repetition, and paging modes R0, R1 and R2 when synchronous links are present. 156
Table 8.3: Initial messaging during start-up. 157
Table 8.4: Increase of train repetition when synchronous links are present. 164
Table 8.5: Messaging during inquiry routines. 165
Table 8.6: Channel classification descriptions 179
Table 9.1: Voice coding schemes supported on the air interface....193
Table 9.2: CVSD parameter values. The values are based on a 16 bit signed number output from the accumulator. 195
Table 12.1: Recommended Frequency Mask for Bluetooth....204

APPENDIX A: GENERAL AUDIO RECOMMENDATIONS

MAXIMUM SOUND PRESSURE

It is the sole responsibility of each manufacturer to design their audio products in a safe way with regards to injury to the human ear. The Bluetooth Specification doesn't specify maximum sound pressure from an audio device.

OTHER TELEPHONY NETWORK REQUIREMENTS

It is the sole responsibility of each manufacturer to design the Bluetooth audio product so that it meets the regulatory requirements of all telephony networks that it may be connected to.

AUDIO LEVELS FOR BLUETOOTH

Audio levels shall be calculated as Send Loudness Rating, SLR, and Receive Loudness Rating, RLR. The calculation methods are specified in ITU-T Recommendation P.79.

The physical test set-up for Handsets and Headsets is described in ITU-T Recommendation P.51 and P.57

The physical test set-up for speakerphones and "Vehicle handsfree systems" is specified in ITU-T Recommendation P.34.

A general equation for computation of loudness rating (LR) for telephone sets is given by ITU-T recommendations P.79 and is given by

$$LR = -\frac{10}{m}\log_{10}\left(\sum_{i=N_1}^{N_2} 10^{m(s_i - w_i)/10}\right), \qquad \text{(EQ 19)}$$

where

m is a constant (~ 0.2).

w_i = weighting coefficient (different for the various LRs).

S_i = the sensitivity at frequency F_i, of the electro-acoustic path

N_1,N_2, consecutive filter bank numbers (Art. Index: 200-4000 Hz)

(EQ 19) on page 204 is used for calculating the (SLR) according to Figure 12.1 on page 205, and (RLR) according to Figure 12.2 on page 205. When calculating LRs one must only include those parts of the frequency band where an actual signal transmission can occur in order to ensure that the additive property of LRs is retained. Therefore ITU-T P.79 uses only the frequency band 200-4000 Hz in LR computations.

MICROPHONE PATH

SLR measurement model

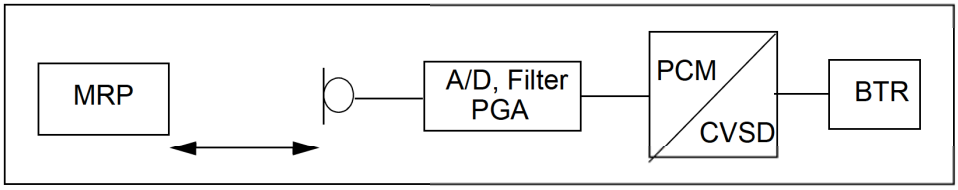


Figure 12.1: SLR measurement set-up.

LOUDSPEAKER PATH

RLR measurement model

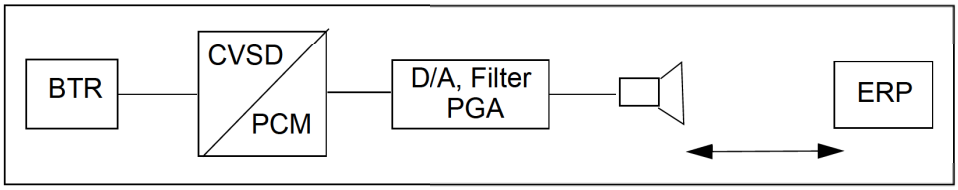


Figure 12.2: RLR measurement set-up.

BLUETOOTH VOICE INTERFACE

The specification for the Bluetooth voice interface should follow in the first place the *ITU-T Recommendations P.79*, which specifies the loudness ratings for telephone sets. These recommendations give general guidelines and specific algorithms used for calculating the loudness ratings of the audio signal with respect to Ear Reference Point (ERP).

For Bluetooth voice interface to the different cellular system terminals, loudness and frequency recommendations based on the cellular standards should be used. For example, GSM 03.50 gives recommendation for both the loudness ratings and frequency mask for a GSM terminal interconnection with Bluetooth.

1- The output of the CVSD decoder are 16-bit linear PCM digital samples, at a sampling frequency of 8 ksample/second. Bluetooth also supports 8-bit log PCM samples of A-law and μ-law type. The sound pressure at the ear reference point for a given 16-bit CVSD sample, should follow the sound pressure level given in the cellular standard specification.

2- A maximum sound pressure which can be represented by a 16-bit linear PCM sample at the output of the CVSD decoder should be specified according 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, μ-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.

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 the Table 12.1 below. The Figure 12.3 below shows a plot of the frequency mask for Bluetooth (solid line). The GSM frequency mask (dotted line) is shown in Figure 12.3 for comparison.

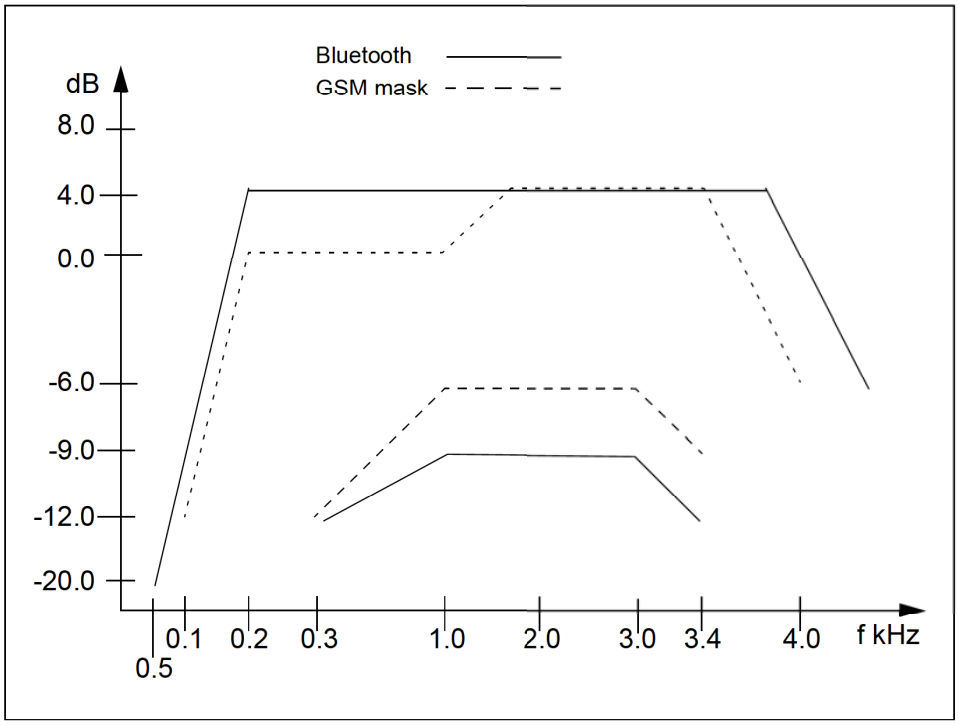


Figure 12.3: Plot of recommended frequency mask for Bluetooth. The GSM send frequency mask is given for comparison (dotted line)

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 12.1: Recommended Frequency Mask for Bluetooth

APPENDIX B: TIMERS

This appendix contains a list of Baseband timers related to inactivity timeout defined in this specification. Definitions and default values of the timers are listed below

All timer values are given in slots.

LIST OF TIMERS

inquiryTO

The *inquiryTO* defines the number of slots the **inquiry** substate will last. The timer value may be changed by the host. HCI provides a a command to change the timer value.

pageTO

The *pageTO* defines the number of slots the **page** substate can last before a response is received. The timer value may be changed by the host. HCI provides a a command to change the timer value.

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*value is 8 slots.

newconnectionTO

Every time a new connection is started through paging, scanning, role switch or unparking, 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 value is 32 slots.

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 LT_ADDR for a period of *supervisionTO*, it will reset the link. The supervision timer keeps running in hold mode, sniff mode and park state.

The *supervisionTO* value may be changed by the host. HCI provides a a command to change the timer value. At the baseband level a default value that is equivalent to 20 seconds will be used.

APPENDIX C: RECOMMENDATIONS FOR AFH OPERATION IN PARK, HOLD AND SNIFF

The three possible AFH operation modes for an AFH capable slave in park, hold and sniff are the same three AFH operation modes used during **CONNECTION** state:

- *Enabled* (using the same AHS as slaves in the **CONNECTION** state)
- *Enabled* (using AHS(79))
- *Disabled*

The master may place an AFH capable slave in any of the three AFH operating modes.

Operation at the Master

A master that has one or more slaves in park, hold or sniff and decides to update them simultaneously shall schedule an *AFH_Instant* for a time that allows it to update all these slaves (as well as its active slaves) with the new adaptation.

A master that has multiple slaves with non-overlapping “wake” times (e.g. slaves in sniff mode with different timing parameters) may keep them *enabled* on the same adaptation provided that its scheduling of the *AFH_Instant* allows enough time to update them all.

This timing is summarized in the figure below. In this example the master decides that a hop sequence adaptation is required. However it cannot schedule an *AFH_Instant* until it has informed its active slave, its slave in hold, its slave in sniff, and had time to un-park its parked slaves.

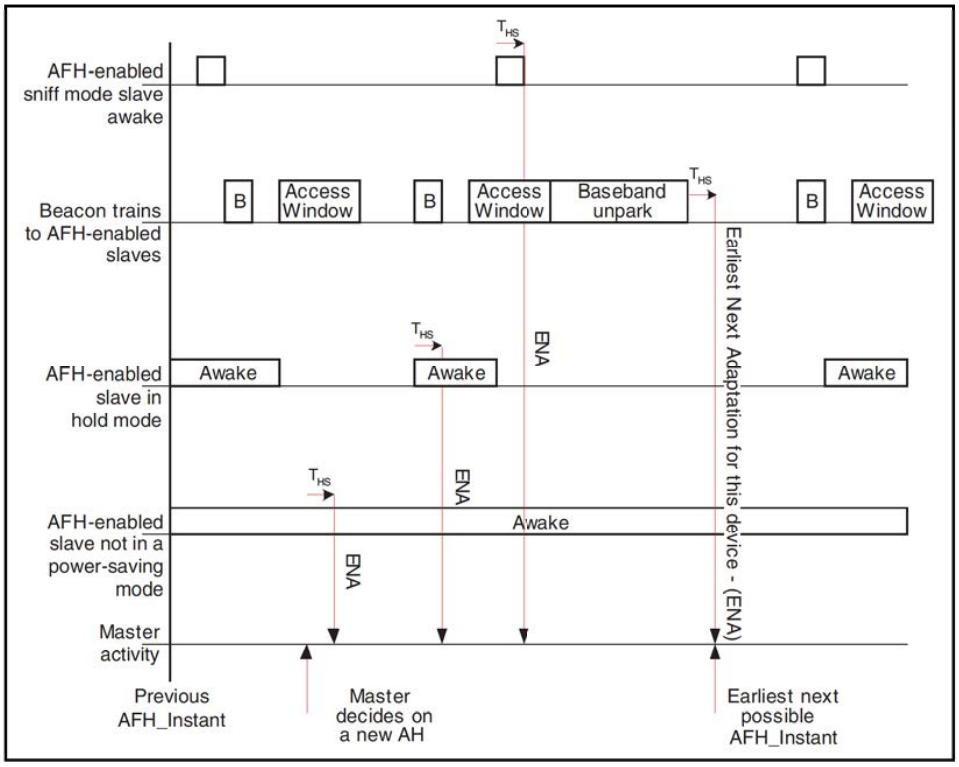


Figure 12.4: Timing constraint on AFH_Instant with slaves in park, hold and sniff

Operation in park

A slave that is in the Park state cannot send or receive any AFH LMP messages. Once the slave has left the Park state the master may subsequently update the slave's adaptation.

AFH Operation in Sniff

Once a slave has been placed in sniff mode, the master may periodically change its AHS without taking the slave out of sniff mode.

AFH Operation in Hold

A slave that is in hold mode cannot send or receive any LMP messages. Once the slave has left hold mode the master may subsequently update the slave's adaptation.



LINK MANAGER PROTOCOL

This specification describes the Link Manager Protocol (LMP) which is used for link set-up and control. The signals are interpreted and filtered out by the Link Manager on the receiving side and are not propagated to higher layers.

CONTENTS

1 Introduction 217

2 General Rules 219

2.1 Message Transport 219
2.2 Synchronization 219
2.3 Packet Format 220
2.4 Transactions 221
2.4.1 LMP Response Timeout 222
2.5 Error Handling 222
2.5.1 Transaction collision resolution 223
2.6 Procedure Rules 223
2.7 General Response Messages 224
2.8 LMP Message Constraints 224

3 Device Features 225

3.1 General Description 225
3.2 Feature Definitions 225
3.3 Feature Mask Definition 230
3.4 Link Manager Interoperability policy 232

4 Procedure Rules 233

4.1 Connection Control 233
4.1.1 Connection establishment 233
4.1.2 Detach 234
4.1.3 Power control 235
4.1.4 Adaptive frequency hopping 237
4.1.4.1 Master enables AFH 238
4.1.4.2 Master disables AFH 238
4.1.4.3 Master updates AFH 239
4.1.4.4 AFH operation in park, hold and sniff modes 239
4.1.5 Channel classification 240
4.1.5.1 Channel classification reporting enabling and disabling 241
4.1.6 Link supervision 242
4.1.7 Channel quality driven data rate change (CQDDR) 243
4.1.8 Quality of service (QoS) 244
4.1.8.1 Master notifies slave of the quality of service 244
4.1.8.2 Device requests new quality of service 245
4.1.9 Paging scheme parameters 246
4.1.9.1 Page mode 246
4.1.9.2 Page scan mode 246

4.1.10 Control of multi-slot packets 247
4.1.11 Enhanced Data Rate 247
4.2 Security 249
4.2.1 Authentication 249
4.2.1.1 Claimant has link key 249
4.2.1.2 Claimant has no link key 250
4.2.1.3 Repeated attempts 250
4.2.2 Pairing 251
4.2.2.1 Responder accepts pairing 251
4.2.2.2 Responder has a fixed PIN 252
4.2.2.3 Responder rejects pairing 252
4.2.2.4 Creation of the link key 253
4.2.2.5 Repeated attempts 253
4.2.3 Change link key 254
4.2.4 Change current link key type 255
4.2.4.1 Change to a temporary link key 255
4.2.4.2 Make the semi-permanent link key the current link key 256
4.2.5 Encryption 257
4.2.5.1 Encryption mode 257
4.2.5.2 Encryption key size 258
4.2.5.3 Start encryption 259
4.2.5.4 Stop encryption 260
4.2.5.5 Change encryption mode, key or random number 260
4.2.6 Request supported encryption key size 261
4.3 Informational Requests 262
4.3.1 Timing accuracy 262
4.3.2 Clock offset 263
4.3.3 LMP version 263
4.3.4 Supported features 264
4.3.5 Name request 266
4.4 Role Switch 267
4.4.1 Slot offset 267
4.4.2 Role switch 268
4.5 Modes of Operation 270
4.5.1 Hold mode 270
4.5.1.1 Master forces hold mode 270
4.5.1.2 Slave forces hold mode 271
4.5.1.3 Master or slave requests hold mode 271
4.5.2 Park state 272
4.5.2.1 Master requests slave to enter park state 274

4.5.2.2 Slave requests to enter park state 275
4.5.2.3 Master sets up broadcast scan window 276
4.5.2.4 Master modifies beacon parameters..... 277
4.5.2.5 Unparking slaves 277
4.5.3 Sniff mode 278
4.5.3.1 Master or slave requests sniff mode 279
4.5.3.2 Moving a slave from sniff mode to active mode 280
4.6 Logical Transports 281
4.6.1 SCO logical transport 281
4.6.1.1 Master initiates an SCO link..... 281
4.6.1.2 Slave initiates an SCO link..... 282
4.6.1.3 Master requests change of SCO parameters 283
4.6.1.4 Slave requests change of SCO parameters . 283
4.6.1.5 Remove an SCO link 283
4.6.2 eSCO logical transport 284
4.6.2.1 Master initiates an eSCO link..... 284
4.6.2.2 Slave initiates an eSCO link..... 285
4.6.2.3 Master or slave requests change of eSCO parameters 286
4.6.2.4 Remove an eSCO link 286
4.6.2.5 Rules for the LMP negotiation and renegotiation 287
4.6.2.6 Negotiation state definitions..... 288
4.7 Test Mode 289
4.7.1 Activation and deactivation of test mode..... 289
4.7.2 Control of test mode 290
4.7.3 Summary of test mode PDUs..... 291
5 Summary 295
5.1 PDU Summary 295
5.2 Parameter Definitions 303
5.3 Default Values 311
6 List of Figures..... 313
7 List of Tables 317

1 INTRODUCTION

The Link Manager Protocol (LMP) is used to control and negotiate all aspects of the operation of the Bluetooth connection between two devices. This includes the set-up and control of logical transports and logical links, and for control of physical links.

The Link Manager Protocol is used to communicate between the Link Managers (LM) on the two devices which are connected by the ACL logical transport. All LMP messages shall apply solely to the physical link and associated logical links and logical transports between the sending and receiving devices.

The protocol is made up of a series of messages which shall be transferred over the ACL-C logical link on the default ACL logical transport between two devices. LMP messages shall be interpreted and acted-upon by the LM and shall not be directly propagated to higher protocol layers.

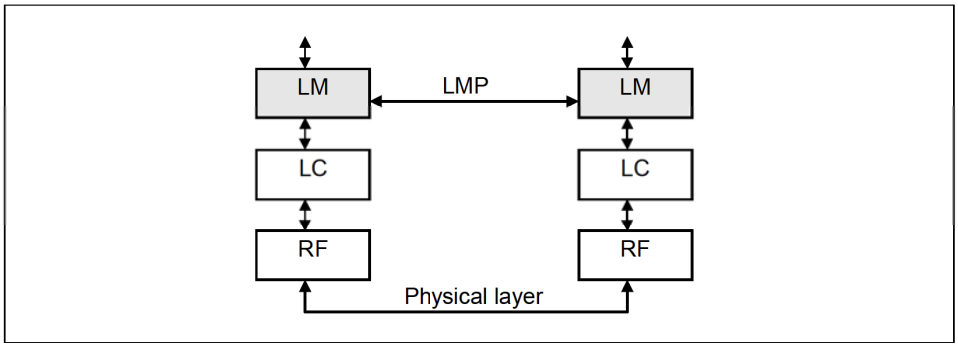


Figure 1.1: Link Manager Protocol signalling layer.

2 GENERAL RULES

2.1 MESSAGE TRANSPORT

LMP messages shall be exchanged over the ACL-C logical link that is carried on the default ACL logical transport (Baseband Specification, Section 4.4, on page 98). The ACL-C logical link is distinguished from the ACL-U (which carries L2CAP and user data) by the Logical Link Identifier (LLID) field carried in the payload header of variable-length packets (Baseband Specification, Section 6.6.2, on page 130).

The ACL-C has a higher priority than other traffic - see Baseband Specification, Section 5.6, on page 108.

The error detection and correction capabilities of the baseband ACL logical transport are generally sufficient for the requirements of the LMP. Therefore LMP messages do not contain any additional error detection information beyond what can be realized by means of sanity checks performed on the contents of LMP messages. Any such checks and protections to overcome undetected errors in LMP messages is an implementation matter.

2.2 SYNCHRONIZATION

This section is informative and explains why many of the LMP message sequences are defined as they are.

LMP messages are carried on the ACL-C logical link, which does not guarantee a time to deliver or acknowledge packets. LMP procedures take account of this when synchronizing state changes in the two devices. For example, criteria are defined that specify when a logical transport address (LT_ADDR) may be re-used after it becomes available due to a device leaving the piconet or entering the park state. Other LMP procedures, such as hold or role switch include the Bluetooth clock as a parameter in order to define a fixed synchronization point. The transitions into and out of sniff mode are protected with a transition mode.

The LC normally attempts to communicate with each slave no less often than every T_{poll} slots (see Section 4.1.8 on page 244) based on the T_{poll} for that slave.

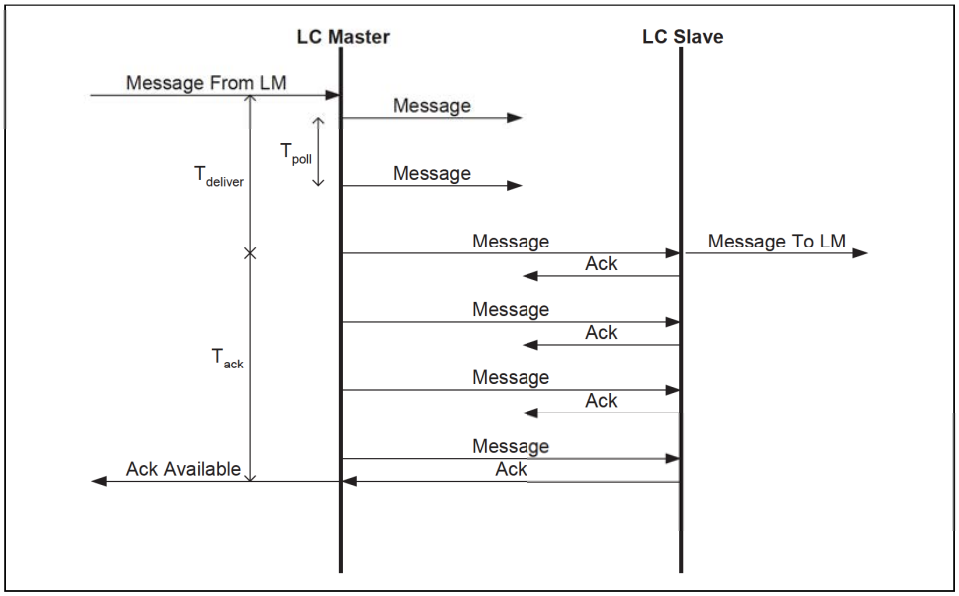


Figure 2.1: Transmission of a message from master to slave[1].

Figure 2.1 on page 220 illustrates the fundamental problem. It shows the transmission of a packet from the master to the slave in conditions of heavy interference for illustrative purposes. It is obvious that neither side can determine the value of either $T_{deliver}$ or T_{ack}. It is therefore not possible to use simple messages to identify uniquely the instant at which a state change occurs in the other device.

2.3 PACKET FORMAT

Each PDU is assigned either a 7 or a 15 bit opcode used to uniquely identify different types of PDUs, see Table 5.1 on page 295. The first 7 bits of the opcode and a transaction ID are located in the first byte of the payload body. If the initial 7 bits of the opcode have one of the special escape values 124-127 then an additional byte of opcode is located in the second byte of the payload and the combination uniquely identifies the PDU.

The FLOW bit in the packet header is always 1 and is ignored on reception.

If the PDU contains one or more parameters these are placed in the payload starting immediately after the opcode, i.e. at byte 2 if the PDU has a 7 bit opcode or byte 3 if the PDU has a 15 bit opcode. The number of bytes used

1. Note the diagram shows the limiting case where the master transmits the message at intervals of T_{poll}. In the case of heavy interference improved performance is gained by transmitting more often.

depends on the length of the parameters. All parameters have a little-endian format, i.e. the least significant byte is transferred first.

LMP messages shall be transmitted using DM1 packets, however if an HV1 SCO link is in use and the length of the payload is less than 9 bytes then DV packets may be used.

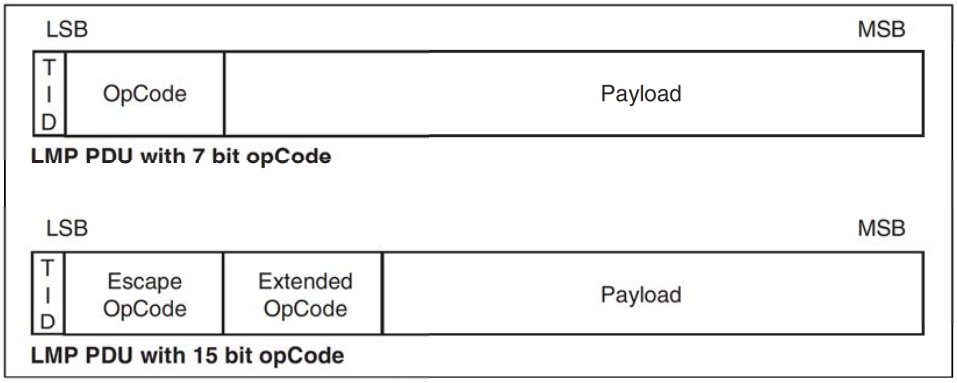


Figure 2.2: Payload body when LMP PDUs are sent.

2.4 TRANSACTIONS

The LMP operates in terms of transactions. A transaction is a connected set of message exchanges which achieve a particular purpose. All PDUs which form part of the same transaction shall have the same value for the transaction ID which is stored as part of the first byte of the opCode (see Section 2.3 on page 220).

The transaction ID is in the least significant bit. It shall be 0 if the PDU forms part of a transaction that was initiated by the master and 1 if the transaction was initiated by the slave.

Each sequence described in Section 4 on page 233 shall be defined as a transaction. For pairing, see Section 4.2.2 on page 251, and encryption, see Section 4.2.5 on page 257, all sequences belonging to each section shall be counted as one transaction and shall use the same transaction ID. For connection establishment, see Section 4.1.1 on page 233, LMP_host_connection_req and the response with LMP_accepted or LMP_not_accepted shall form one transaction and have the transaction ID of 0. LMP_setup_complete is a stand-alone PDU, which forms a transaction by itself.

For error handling, see Section 2.5 on page 222, the PDU to be rejected and LMP_not_accepted or LMP_not_accepted_ext shall form a single transaction.

2.4.1 LMP Response Timeout

The time between receiving a baseband packet carrying an LMP PDU and sending a baseband packet carrying a valid response PDU, according to the procedure rules in Section 4 on page 233, shall be less than the LMP Response Timeout. The value of this timeout is 30 seconds. Note that the LMP Response Timeout is applied not only to sequences described in Section 4 on page 233, but also to the series of the sequences defined as the transactions in Section 4 on page 233. It shall also be applied to the series of LMP transactions that take place during a period when traffic on the ACL-U logical link is disabled for the duration of the series of LMP transactions, for example during the enabling of encryption.

The LMP Response Timeout shall restart each time an LMP PDU which requires a reply is queued for transmission by the baseband.

2.5 ERROR HANDLING

If the LM receives a PDU with unrecognized opcode, it shall respond with LMP_not_accepted or LMP_not_accepted_ext with the error code *unknown LMP PDU.* The opcode parameter that is echoed back is the unrecognized opcode.

If the LM receives a PDU with invalid parameters, it shall respond with LMP_not_accepted or LMP_not_accepted_ext with the error code *invalid LMP parameters*.

If the maximum response time, see Section 2.4 on page 221, is exceeded or if a link loss is detected (see Baseband Specification, Section 3.1, on page 95), the party that waits for the response shall conclude that the procedure has terminated unsuccessfully.

Erroneous LMP messages can be caused by errors on the channel or systematic errors at the transmit side. To detect the latter case, the LM should monitor the number of erroneous messages and disconnect if it exceeds a threshold, which is implementation-dependent.

When the LM receives a PDU that is not allowed, and the PDU normally expects a PDU reply, for example LMP_host_connection_req or LMP_unit_key, the LM shall return PDU LMP_not_accepted or LMP_not_accepted_ext with the error code *PDU not allowed*. If the PDU normally doesn't expect a reply, for example LMP_sres or LMP_temp_key, the PDU will be ignored.

The reception of an optional PDU which is not supported shall be handled in one of two ways: if the LM simply does not know the opcode (e.g. it was added at a later version of the specification) it shall respond with LMP_not_accepted or LMP_not_accepted_ext with the error code *unknown LMP PDU*. If the LM recognizes the PDU as optional but unsupported then it shall reply with LMP_not_accepted or LMP_not_accepted_ext with the error code *unsupported LMP feature* if the PDU would normally generate a reply otherwise no reply is generated.

2.5.1 Transaction collision resolution

Since LMP PDUs are not interpreted in real time, collision situations can occur where both LMs initiate the same procedure and both cannot be completed. In this situation, the master shall reject the slave-initiated procedure by sending LMP_not_accepted or LMP_not_accepted_ext with the error code *LMP error transaction collision*. The master-initiated procedure shall then be completed.

Collision situations can also occur where both LMs initiate different procedures and both cannot be completed. In this situation, the master shall reject the slave-initiated procedure by sending LMP_not_accepted or LMP_not_accepted_ext with the error code *LMP error different transaction collision*. The master- initiated procedure shall then be completed.

2.6 PROCEDURE RULES

Each procedure is described and depicted with a sequence diagram. The following symbols are used in the sequence diagrams:

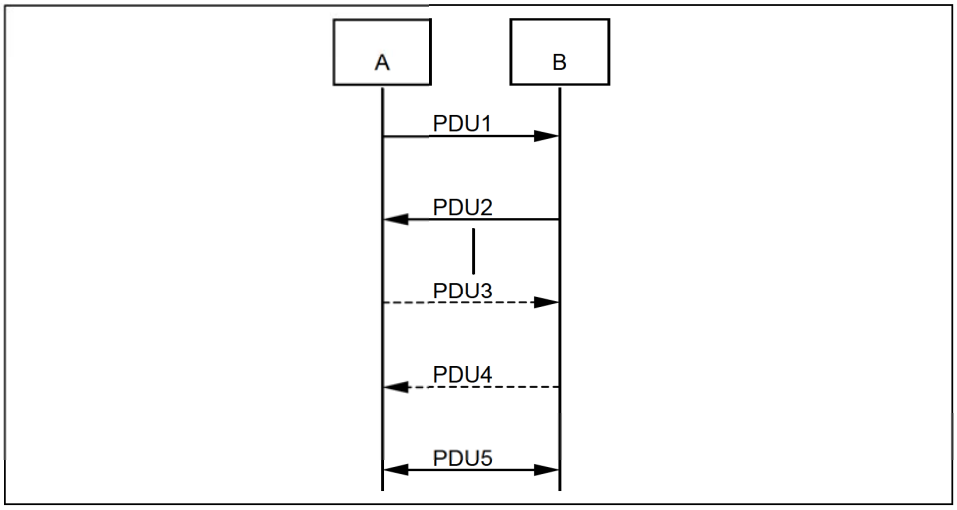


Figure 2.3: Symbols used in sequence diagrams.

PDU1 is a PDU sent from A to B. PDU2 is a PDU sent from B to A. PDU3 is a PDU that is optionally sent from A to B. PDU4 is a PDU that is optionally sent from B to A. PDU5 is a PDU sent from either A or B. A vertical line indicates that more PDUs can optionally be sent.

2.7 GENERAL RESPONSE MESSAGES

The PDUs LMP_accepted, LMP_accepted_ext, LMP_not_accepted and LMP_not_accepted_ext are used as response messages to other PDUs in a number of different procedures. LMP_accepted or LMP_accepted_ext includes the opcode of the message which is accepted. LMP_not_accepted or LMP_not_accepted_ext includes the opcode of the message which is not accepted and the error code why it is not accepted.

LMP_accepted_ext and LMP_not_accepted_ext shall be used when the opcode is 15 bits in length. LMP_accepted and LMP_not_accepted shall be used when the opcode is 7 bits in length.

M/O	PDU	Contents
M	LMP_accepted	op code
M	LMP_not_accepted	op code error code
O	LMP_accepted_ext	escape op code extended op code
O	LMP_not_accepted_ext	escape op code extended op code error code

Table 2.1: General response messages.

2.8 LMP MESSAGE CONSTRAINTS

This section is informative.

- No LMP message shall exceed the maximum payload length of a single DM1 packet i.e. 17 bytes in length (Baseband Specification, Section 6.5.4.1, on page 126).
- All LM messages are of fixed length apart from those sent using broadcast in park state.
- The LMP version number shall not be used to indicate the presence or absence of functionality.

3 DEVICE FEATURES

3.1 GENERAL DESCRIPTION

Each PDU is either Mandatory or Optional as defined by the M/O field in the tables of Section 4 on page 233. An M in this field shall indicate that support for the feature is mandatory. An O in this field shall indicate that support for the PDU is optional and it shall be followed by the number(s) of the feature(s) involved in brackets.

All features added after the 1.1 specification have associated LMP feature bits. Support of these features may be made “mandatory” by the qualification process but the LM still considers them to be optional since it must interoperate with older devices which do not support them.

The LM does not need to be able to transmit a PDU which is optional. Support of optional PDUs is indicated by a device's features mask. The features mask can be read (see Section 4.3.4 on page 264). An LM shall not send or be sent any PDU which is incompatible with the features signalled in its or its peer's features mask, as detailed in Section 3.2 .

3.2 FEATURE DEFINITIONS

Feature	Definition
Extended features	This feature indicates whether the device is able to support the extended features mask using the LMP sequences defined in Section 4.3.4 on page 264.
Timing accuracy	This feature indicates whether the LM supports requests for timing accuracy using the sequence defined in Section 4.3.1 on page 262.
Enhanced inquiry scan	This feature indicates whether the device is capable of supporting the enhanced inquiry scan mechanism as defined in Baseband Specification, Section 8.4.1, on page 164. The presence of this feature has only local meaning and does not imply support for any additional LMP PDUs or sequences.
Interlaced inquiry scan	This feature indicates whether the device is capable of supporting the interlaced inquiry scan mechanism as defined in Baseband Specification, Section 8.4.1, on page 164. The presence of this feature has only local meaning and does not imply support for any additional LMP PDUs or sequences.
Interlaced page scan	This feature indicates whether the device is capable of supporting the interlaced page scan mechanism as defined in Baseband Specification, Section 8.3.1, on page 154. The presence of this feature has only local meaning and does not imply support for any additional LMP PDUs or sequences.

Table 3.1: Feature definitions.

Feature	Definition
RSSI with inquiry results	This feature indicates whether the device is capable of reporting the RSSI with inquiry results as defined in Baseband Specification, Section 8.4.2, on page 165. The presence of this feature has only local meaning and does not imply support for any additional LMP PDUs or sequences.
Paging parameter negotiation	This feature indicates whether the LM is capable of supporting the signaling of changes in the paging scheme as defined in Section 4.1.9 on page 246.
3 slot packets	This feature indicates whether the device supports the transmission and reception of both DM3 and DH3 packets for the transport of traffic on the ACL-U logical link.
5 slot packets	This feature indicates whether the device supports the transmission and reception of both DM5 and DH5 packets for the transport of traffic on the ACL-U logical link.
Flow control lag	This is defined as the total amount of ACL-U data that can be sent following the receipt of a valid payload header with the payload header flow bit set to 0 and is in units of 256 bytes. See further in Baseband Specification, Section 6.6.2, on page 130.
AFH capable slave	This feature indicates whether the device is able to support the Adaptive Frequency Hopping mechanism as a slave as defined in Baseband Specification, Section 2.3, on page 75 using the LMP sequences defined in Section 4.1.4 on page 237.
AFH classification slave	This feature indicates whether the device is able to support the AFH classification mechanism as a slave as defined in Baseband Specification, Section 8.6.8, on page 181 using the LMP sequences defined Section 4.1.5 on page 240.
AFH capable master	This indicates whether the device is able to support the Adaptive Frequency Hopping mechanism as a master as defined in Baseband Specification, Section 2.3, on page 75 using the LMP sequences defined in Section 4.1.4 on page 237.
AFH classification master	This feature indicate whether the device is able to support the AFH classification mechanism as a master as defined in Baseband Specification, Section 8.6.8, on page 181 using the LMP sequences defined Section 4.1.5 on page 240.
Power control	This feature indicates whether the device is capable of adjusting its transmission power. This feature indicates the ability to receive the LMP_incr_power and LMP_decr_power PDUs and transmit the LMP_max_power and LMP_min_power PDUs, using the sequences defined in Section 4.1.3 on page 235. These sequences may be used even if the remote device does not support the power control feature, as long as it supports the Power control requests feature.

Table 3.1: Feature definitions.

Feature	Definition
Power control requests	This feature indicates whether the device is capable of determining if the transmit power level of the other device should be adjusted and will send the LMP_incr_power and LMP_decr_power PDUs to request the adjustments. It indicates the ability to receive the LMP_max_power and LMP_min_power PDUs, using the sequences in Section 4.1.3 on page 235. These sequences may be used even if the remote device does not support the RSSI feature, as long as it supports the power control feature.
Channel Quality Driven Data Rate	This feature indicates whether the LM is capable of recommending a packet type (or types) depending on the channel quality using the LMP sequences defined in section Section 4.1.7 on page 243.
Broadcast encryption	This feature indicates whether the device is capable of supporting broadcast encryption as defined in [Part H] Section 4.2 on page 788 and also the sequences defined in Section 4.2.6 on page 261 and Section 4.2.4 on page 255. Note: Devices compliant to versions of this specification 1.1 and earlier may support broadcast encryption even though this feature bit is not set.
Encryption	This feature indicates whether the device supports the encryption of packet contents using the sequence defined in Section 4.2.5 on page 257.
Slot offset	This feature indicates whether the LM supports the transfer of the slot offset using the sequence defined in Section 4.4.1 on page 267.
Role switch	This feature indicates whether the device supports the change of master and slave roles as defined by baseband Section 8.6.5 on page 175 using the sequence defined in Section 4.4.2 on page 268.
Hold mode	This feature indicates whether the device is able to support Hold mode as defined in baseband Section 8.8 on page 185 using the LMP sequences defined in Section 4.5.1 on page 270.
Sniff mode	This feature indicates whether the device is able to support sniff mode as defined in baseband Section 8.7 on page 183 using the LMP sequences defined in Section 4.5.3 on page 278.
Park state	This feature indicates whether the device is able to support Park state as defined in baseband Section 8.9 on page 185 using the LMP sequences defined in Section 4.5.2 on page 272.
SCO link	This feature shall indicate whether the device is able to support the SCO logical transport as defined in Baseband Specification, Section 4.3, on page 98, the HV1 packet defined in Baseband Specification, Section 6.5.2.1, on page 123 and receiving the DV packet defined in Baseband Specification, Section 6.5.2.4, on page 123 using the LMP sequence in Section 4.6.1 on page 281.
HV2 packets	This feature indicates whether the device is capable of supporting the HV2 packet type as defined in baseband Section 6.5.2.2 on page 123 on the SCO logical transport.

Table 3.1: Feature definitions.

Feature	Definition
HV3 packets	This feature indicates whether the device is capable of supporting the HV3 packet type as defined in baseband Section 6.5.2.3 on page 123 on the SCO logical transport.
μ-law log synchronous data	This feature indicates whether the device is capable of supporting μ-Law Log format data as defined in Baseband Specification, Section 9.1, on page 195 on the SCO and eSCO logical transports.
A-law log synchronous data	This feature indicates whether the device is capable of supporting A-Law Log format data as defined in Baseband Specification, Section 9.1, on page 195 on the SCO and eSCO logical transports.
CVSD synchronous data	This feature indicates whether the device is capable of supporting CVSD format data as defined in Baseband Specification, Section 9.2, on page 195 on the SCO and eSCO logical transports.
Transparent synchronous data	This feature indicates whether the device is capable of supporting transparent synchronous data as defined in Baseband Specification, Section 6.4.3, on page 117 on the SCO and eSCO logical transports.
Extended SCO link	This feature indicates whether the device is able to support the eSCO logical transport as defined Baseband Specification, Section 5.5, on page 108 and the EV3 packet defined in Baseband Specification, Section 6.5.3.1, on page 124 using the LMP sequences defined in Section 4.6.2 on page 284.
EV4 packets	This feature indicates whether the device is capable of supporting the EV4 packet type defined in Baseband Specification, Section 6.5.3.2, on page 124 on the eSCO logical transport.
EV5 packets	This feature indicates whether the device is capable of supporting the EV5 packet type defined in Baseband Specification, Section 6.5.3.3, on page 124 on the eSCO logical transport.
Enhanced Data Rate ACL 2 Mbps mode	This feature indicates whether the device supports the transmission and reception of 2-DH1 packets for the transport of traffic on the ACL-U logical link.
Enhanced Data Rate ACL 3 Mbps mode	This feature indicates whether the device supports the transmission and reception of 3-DH1 packets for the transport of traffic on the ACL-U logical link. This feature bit shall only be set if the “Enhanced Data Rate ACL 2 Mbps mode” feature bit is set.
3-slot Enhanced Data Rate ACL packets	This feature indicates whether the device supports the transmission and reception of three-slot Enhanced Data Rate packets on the ACL-U logical link. This feature bit shall only be set if the “Enhanced Data Rate ACL 2 Mbps mode” feature bit is set.
5-slot Enhanced Data Rate ACL packets	This feature indicates whether the device supports the transmission and reception of 5-slot Enhanced Data Rate packets for the transport of traffic on the ACL-U logical link. This feature bit shall only be set if the “Enhanced Data Rate ACL 2 Mbps mode” feature bit is set.

Table 3.1: Feature definitions.

Feature	Definition
Enhanced Data Rate eSCO 2 Mbps mode	This feature indicates whether the device supports the transmission and reception of 2-EV3 packets for the transport of traffic on the eSCO logical transport.
Enhanced Data Rate eSCO 3 Mbps mode	This feature indicates whether the device supports the transmission and reception of 3-EV3 packets for the transport of traffic on the eSCO logical transport. This feature bit shall only be set if the "Enhanced Data Rate eSCO 2 Mbps mode" feature bit is set.
3-slot Enhanced Data Rate eSCO packets	This feature indicates whether the device supports the transmission and reception of 3-slot Enhanced Data Rate packets for the transport of traffic on the eSCO logical transport. This feature bit shall only be set if the "Enhanced Data Rate eSCO 2 Mbps mode" feature bit is set.

Table 3.1: Feature definitions.

3.3 FEATURE MASK DEFINITION

The features are represented as a bit mask when they are transferred in LMP messages. For each feature a single bit is specified which shall be set to 1 if the feature is supported and set to 0 otherwise. The single exception is the flow control lag which is coded as a 3 bit field with the least significant bit in byte 2 bit 4 and the most significant bit in byte 2 bit 6. All unknown or unassigned feature bits shall be set to 0.

No.	Supported feature	Byte	Bit
0	3 slot packets	0	0
1	5 slot packets	0	1
2	Encryption	0	2
3	Slot offset	0	3
4	Timing accuracy	0	4
5	Role switch	0	5
6	Hold mode	0	6
7	Sniff mode	0	7
8	Park state	1	0
9	Power control requests	1	1
10	Channel quality driven data rate (CQDDR)	1	2
11	SCO link	1	3
12	HV2 packets	1	4
13	HV3 packets	1	5
14	μ-law log synchronous data	1	6
15	A-law log synchronous data	1	7
16	CVSD synchronous data	2	0
17	Paging parameter negotiation	2	1
18	Power control	2	2
19	Transparent synchronous data	2	3
20	Flow control lag (least significant bit)	2	4
21	Flow control lag (middle bit)	2	5
22	Flow control lag (most significant bit)	2	6
23	Broadcast encryption	2	7
24	Reserved	3	0
25	Enhanced Data Rate ACL 2 Mbps mode	3	1

Table 3.2: Feature mask definition

No.	Supported feature	Byte	Bit
26	Enhanced Data Rate ACL 3Mbps mode	3	2
27	Enhanced inquiry scan	3	3
28	Interlaced inquiry scan	3	4
29	Interlaced page scan	3	5
30	RSSI with inquiry results	3	6
31	Extended SCO link (EV3 packets)	3	7
32	EV4 packets	4	0
33	EV5 packets	4	1
34	Reserved	4	2
35	AFH capable slave	4	3
36	AFH classification slave	4	4
37	Reserved	4	5
38	Reserved	4	6
39	3-slot Enhanced Data Rate ACL packets	4	7
40	5-slot Enhanced Data Rate ACL packets	5	6
41	reserved	5	1
42	reserved	5	2
43	AFH capable master	5	3
44	AFH classification master	5	4
45	Enhanced Data Rate eSCO 2 Mbps mode	5	5
46	Enhanced Data Rate eSCO 3 Mbps mode	5	6
47	3-slot Enhanced Data Rate eSCO packets	5	7
48	Reserved	6	0
63	Extended features	7	7

Table 3.2: Feature mask definition

3.4 LINK MANAGER INTEROPERABILITY POLICY

Link managers of any version will interoperate using the lowest common subset of functionality by reading the LMP features mask (defined in Table 3.2 on page 230).

An optional LMP PDU shall only be sent to a device if the corresponding feature bit is set in its feature mask. The exception to this are certain PDUs (see Section 4.1.1 on page 233) which can be sent before the features mask is requested. Note: a later version device with a restricted feature set is indistinguishable from an earlier version device with the same features.

4 PROCEDURE RULES

4.1 CONNECTION CONTROL

4.1.1 Connection establishment

After the paging procedure, LMP procedures with for clock offset request, LMP version, supported features, name request and detach may then be initiated.

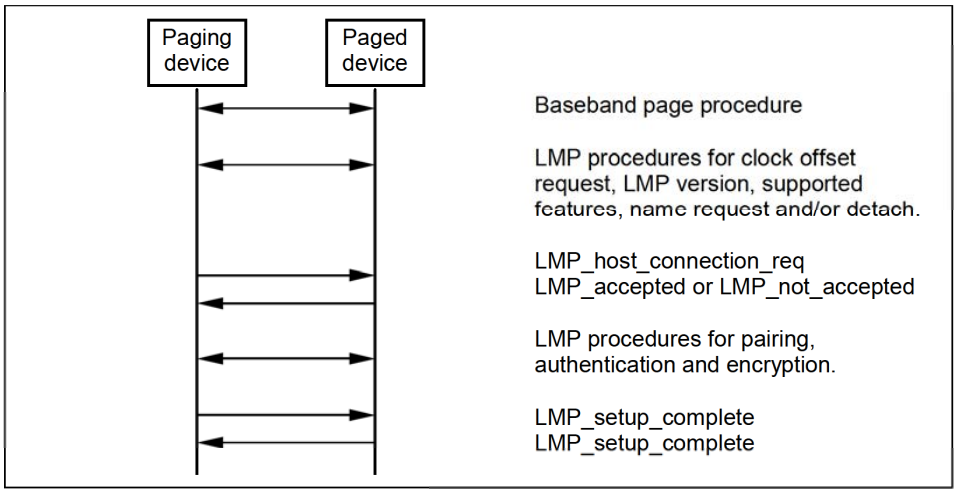


Figure 4.1: Connection establishment.

When the paging device wishes to create a connection involving layers above LM, it sends an LMP_host_connection_req PDU. When the other side receives this message, the host is informed about the incoming connection. The remote device can accept or reject the connection request by sending an LMP_accepted PDU or an LMP_not_accepted PDU. Alternatively, if the slave needs a role switch, see Section 4.4.2 on page 268, it sends an LMP_slot_offset PDU and LMP_switch_req PDU after it has received an LMP_host_connection_req PDU. If the role switch fails the LM shall continue with the creation of the connection unless this cannot be supported due to limited resources in which case the connection shall be terminated with an LMP_detach PDU with error code *other end terminated connection: low resources*. When the role switch has been successfully completed, the old slave will reply with an LMP_accepted PDU or an LMP_not_accepted PDU to the LMP_host_connection_req PDU (with the transaction ID set to 0).

If the paging device receives an LMP_not_accepted PDU in response to LMP_host_connection_req it shall immediately disconnect the link using the mechanism described in Section 4.1.2 on page 234.

If the LMP_host_connection_req PDU is accepted, LMP security procedures (pairing, authentication and encryption) may be invoked. When a device is not going to initiate any more security procedures during connection establishment it sends an LMP_setup_complete PDU. When both devices have sent LMP_setup_complete PDUs the traffic can be transferred on the ACL-U logical transport.

M/O	PDU	Contents
M	LMP_host_connection_req	-
M	LMP_setup_complete	-

Table 4.1: PDUs used for connection establishment.

4.1.2 Detach

The connection between two Bluetooth devices may be detached anytime by the master or the slave. An error code parameter is included in the message to inform the other party of why the connection is detached.

M/O	PDU	Contents
M	LMP_detach	error code

Table 4.2: PDU used for detach.

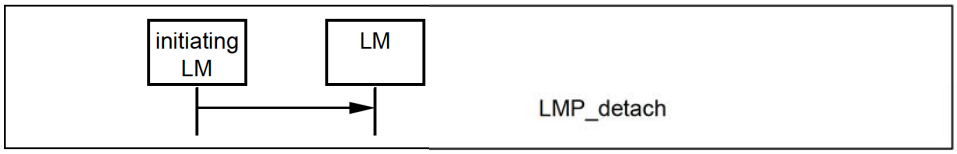
The initiating LM shall pause traffic on the ACL-U logical link (see Baseband Specification, Section 5.3.1, on page 108). The initiating LM then queues the LMP_detach for transmission and it shall start a timer for $6*T_{poll}$ slots where T_{poll} is the poll interval for the connection. If the initiating LM receives the baseband acknowledgement before the timer expires it starts a timer for $3*T_{poll}$ slots. When this timer expires, and if the initiating LM is the master, the LT_ADDR(s) may be re-used immediately. If the initial timer expires then the initiating LM drops the link and starts a timer for $T_{linksupervisiontimeout}$ slots after which the LT_ADDR(s) may be re-used if the initiating LM is the master.

When the receiving LM receives the LMP_detach, it shall start a timer for $6*T_{poll}$ slots if it is the master and $3*T_{poll}$ if it is the slave. On timer expiration, the link shall be detached and, if the receiving LM is the master, the LT_ADDR(s) may be re-used immediately. If the receiver never receives the LMP_detach then a link supervision timeout will occur, the link will be detached, and the LT_ADDR may be re-used immediately.

If at any time during this or any other LMP sequence the Link supervision time-out expires then the link shall be terminated immediately and the LT_ADDR(S) may be re-used immediately.

If the connection is in hold mode, the initiating LM shall wait for hold mode to end before initiating the procedure defined above. If the connection is in sniff mode or park state, the initiating LM shall perform the procedure to exit sniff

mode or park state before initiating the procedure defined above. If the procedure to exit sniff mode or park state does not complete within the LMP response timeout (30 seconds) the procedure defined above shall be initiated anyway.



Sequence 1: Connection closed by sending LMP_detach.

4.1.3 Power control

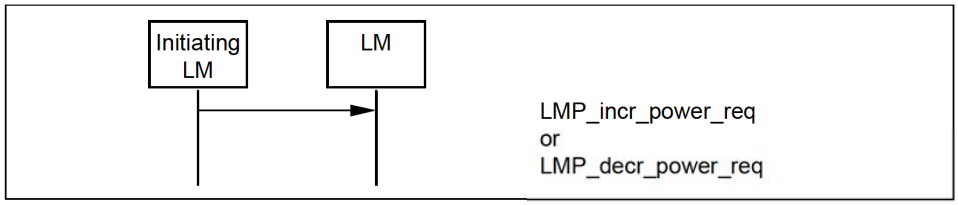
If the received signal characteristics differs too much from the preferred value of a Bluetooth device, it may request an increase or a decrease of the other device's TX power. The power adjustment requests may be made at anytime following a successful baseband paging procedure.

If a device does not support power control requests this is indicated in the supported features list and thus no power control requests shall be sent after the supported features response has been processed. Prior to this time, a power control adjustment might be sent and if the recipient does not support power control it is allowed to send LMP_max_power in response to LMP_incr_power_req and LMP_min_power in response to LMP_decr_power_req. Another possibility is to send LMP_not_accepted with the error code *unsupported LMP feature*.

Upon receipt of an LMP_incr_power_req PDU or LMP_decr_power_req PDU the output power shall be increased or decreased one step. See Radio Specification Section 3, on page 31 for the definition of the step size. The TX power is a property of the physical link, and affects all logical transports carried over the physical link. Power control requests carried over the default ACL-C logical link shall only affect the physical link associated with the default ACL-C logical link: they shall not affect the power level used on the physical links to other slaves.

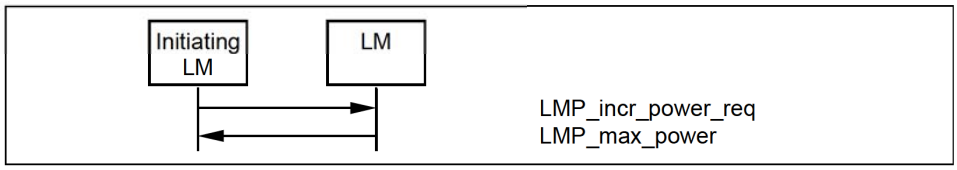
M/O	PDU	Contents
O(9)	LMP_incr_power_req	for future use (1 Byte)
O(9)	LMP_decr_power_req	for future use (1 Byte)
O(18)	LMP_max_power	-
O(18)	LMP_min_power	-

Table 4.3: PDUs used for power control.

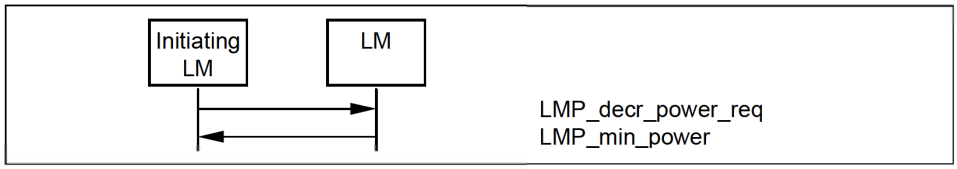


Sequence 2: A device requests a change of the other device's TX power.

If the receiver of LMP_incr_power_req is at maximum power LMP_max_power shall be returned. The device shall only request an increase again after having requested a decrease at least once. If the receiver of LMP_decr_power_req is at minimum power then LMP_min_power shall be returned and the device shall only request a decrease after having requested an increase at least once.



Sequence 3: The TX power cannot be increased.



Sequence 4: The TX power cannot be decreased.

One byte is reserved in LMP_incr/decr_power_req for future use. The parameter value shall be 0x00 and ignored upon receipt.

4.1.4 Adaptive frequency hopping

AFH is used to improve the performance of physical links in the presence of interference as well as reducing the interference caused by physical links on other devices in the ISM band. AFH shall only be used during the connection state.

M/O	PDU	Contents
O(35) Rx O(43) Tx	LMP_set_AFH	AFH_Instant, AFH_Mode, AFH_Channel_Map

Table 4.4: PDUs used for AFH

The LMP_set_AFH PDU contains three parameters: AFH_Instant, AFH_Mode, and AFH_Channel_Map. The parameter, AFH_Instant, specifies the instant at which the hopset switch will become effective. This is specified as a Bluetooth Clock value of the master's clock, that is available to both devices. The AFH instant is chosen by the master and shall be an even value at least $6*T_{poll}$ or 96 slots (whichever is greater) in the future, where T_{poll} is at least the longest poll interval for all AFH enabled physical links. The AFH_instant shall be within 12 hours of the current clock value. The parameter AFH_Mode, specifies whether AFH shall be enabled or disabled. The parameter AFH_Channel_Map, specifies the set of channels that shall be used if AFH is enabled.

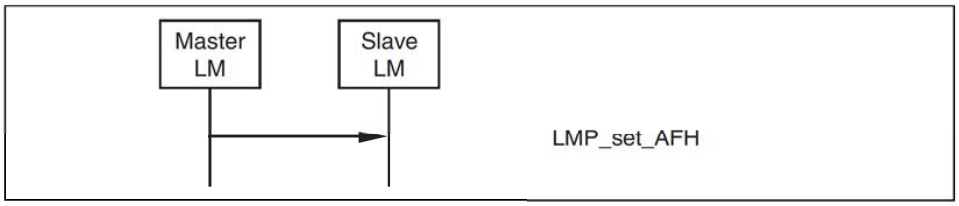
When the LMP_set_AFH PDU is received the AFH instant shall be compared with the current Bluetooth clock value. If it is in the past then the AFH_instant has passed and the slave shall immediately configure the hop selection kernel (see Baseband Specification, Section 2.6.3, on page 89) with the new AFH_mode and AFH_channel_map specified in the LMP_set_AFH PDU. If it is in the future then a timer shall be started to expire at the AFH instant. When this timer expires it shall configure the hop selection kernel with the new AFH_mode and AFH_channel_map.

The master shall not send a new LMP_set_AFH PDU to a slave until it has received the baseband acknowledgement for any previous LMP_set_AFH addressed to that slave and the instant has passed.

Role switch while AFH is enabled shall follow the procedures define by Baseband Specification, Section 8.6.5, on page 175.

4.1.4.1 Master enables AFH

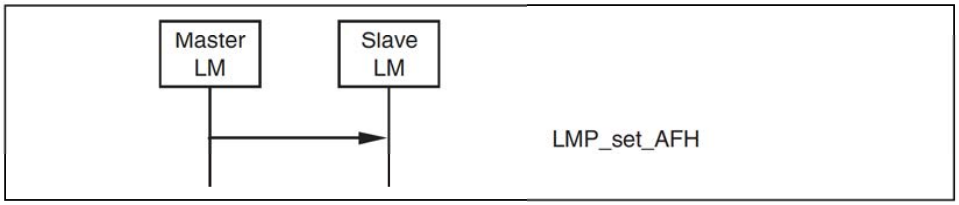
Prior to enabling AFH the master LM shall pause traffic on the ACL-U logical link (see Baseband Specification, Section 5.3.1, on page 108). The master shall then enable AFH on a physical link by sending the LMP_set_AFH PDU with AFH_mode set to AFH_enabled, the AFH_channel_map parameter containing the set of used and unused channels, and an AFH_instant. The LM shall not calculate the AFH instant until after traffic on the ACL-U logical link has been stopped. The master considers the physical link to be AFH_enabled once the baseband acknowledgement has been received and the AFH_instant has passed. Once the baseband acknowledgement has been received the master shall restart transmission on the ACL-U logical link.



Sequence 5: Master Enables AFH.

4.1.4.2 Master disables AFH

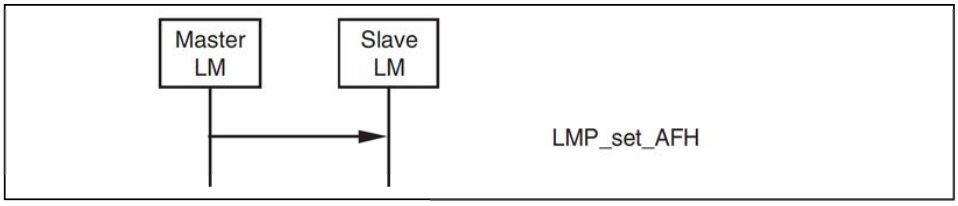
Prior to disabling AFH the master LM shall pause traffic on the ACL-U logical link (Baseband Specification, Section 5.3.1, on page 108). The master shall then disable AFH operation on a physical link by sending the LMP_set_AFH PDU with AFH_mode set to AFH_disabled and an AFH_instant. The AFH_channel_map parameter is not valid when AFH_mode is AFH_disabled. The LM shall not calculate the AFH instant until after traffic on the ACL-U logical link has been stopped. The master considers the physical link to have entered AFH_disabled operation once the baseband acknowledgement has been received and the AFH_instant has passed. Once the baseband acknowledgement has been received the master shall restart transmission on the ACL-U logical link.



Sequence 6: Master disables AFH.

4.1.4.3 Master updates AFH

A master shall update the AFH parameters on a physical link by sending the LMP_set_AFH PDU with AFH_mode set to AFH_enabled, an AFH_instant and a new AFH_channel_map. The master shall consider the slave to have the updated AFH parameters once the baseband acknowledgement has been received and the AFH_instant has passed.



Sequence 7: Master Updates AFH.

4.1.4.4 AFH operation in park, hold and sniff modes

A slave in Park, Hold or Sniff shall retain the AFH_mode and AFH_channel_map prior to entering those modes. A master may change the AFH_mode while a slave is in sniff.

A master that receives a request from an AFH_enabled slave to enter Park, Hold or Sniff and decides to operate the slave using a different hop sequence shall respond with an LMP_set_AFH PDU specifying the new hop sequence.

The master continues with the LMP signalling, for Park, Hold or Sniff initiation, once the baseband acknowledgement for the LMP_set_AFH PDU has been received. Optionally, the master may delay the continuation of this LMP signalling until after the instant. An AFH_capable_slave device shall support both of these cases.

A master that receives a request from an AFH_enabled slave to enter Park, Hold or Sniff and decides not to change the slave's hop sequence shall respond exactly as it would do without AFH. In this case, AFH operation has no effect on the LMP signalling.

4.1.5 Channel classification

A master may request channel classification information from a slave that is AFH_enabled.

A slave that supports the AFH_classification_slave feature shall perform channel classification and reporting according to its AFH_reporting_mode. The master shall control the AFH_reporting_mode using the LMP_channel_classification_req PDU. The slave shall report its channel classification using the LMP_channel_classification PDU.

The slave shall report pairs of channels as *good*, *bad* or *unknown*. See Table 5.2 on page 303 for the detailed format of the AFH_Channel_Classification parameter. When one channel in the n^{th} channel pair is good and the other channel is unknown the n^{th} channel pair shall be reported as good. When one channel in the n^{th} channel pair is bad and the other is unknown the n^{th} channel pair shall be reported as bad. It is implementation dependent what to report when one channel in a channel pair is good and the other is bad.

M/O	PDU	Contents
O(36) Rx O(44) Tx	LMP_channel_classification_req	AFH_Reporting_Mode, AFH_Min_Interval, AFH_Max_Interval
O(36) Tx O(44) Rx	LMP_channel_classification	AFH_Channel_Classification

Table 4.5: PDUs used for Channel Classification Reporting.

The LMP_channel_classification_req PDU contains three parameters: AFH_Reporting_Mode, AFH_Min_Interval, and AFH_Max_Interval. In the AFH_reporting_disabled state, the slave shall not generate any channel classification reports. The parameter AFH_min_interval, defines the minimum amount of time from the last LMP_channel_classification command that was sent before the next LMP_channel_classification PDU may be sent. The parameter AFH_max_interval, defines the maximum amount of time between the change in the radio environment being detected by a slave and its generation of an LMP_channel_classification PDU. The AFH_max_interval shall be equal to or larger than AFH_min_interval.

The AFH_reporting_mode parameter shall determine if the slave is in the AFH_reporting_enabled or AFH_reporting_disabled state. The default state, prior to receipt of any LMP_channel_classification_req PDUs, shall be AFH_reporting_disabled.

AFH_reporting_mode is implicitly set to the AFH_reporting_disabled state when any of the following occur:

- Establishment of a connection at the baseband level
- Master-slave role switch

- Entry to park state operation
- Entry to hold mode operation

AFH_reporting_mode is implicitly restored to its former value when any of the following occur:

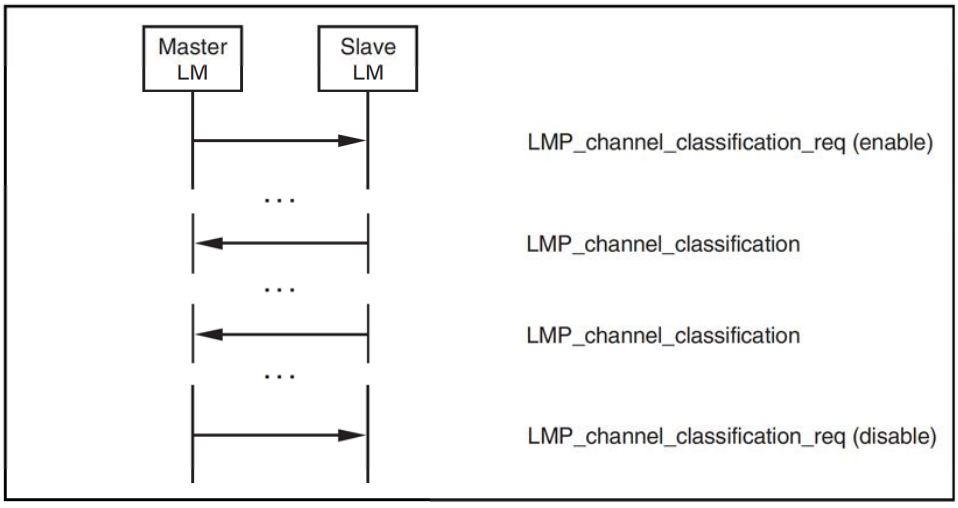
- Exit from park state operation
- Exit from hold mode
- Failure of Master-slave role switch

4.1.5.1 Channel classification reporting enabling and disabling

A master enables slave channel classification reporting by sending the LMP_channel_classification_req PDU with the AFH_reporting_mode parameter set to AFH_reporting_enabled.

When a slave has had classification reporting enabled by the master it shall send the LMP_channel_classification PDU according to the information in the latest LMP_channel_classification_req PDU. The LMP_channel_classification PDU shall not be sent if there has been no change in the slave's channel classification.

A master disables slave channel classification reporting by sending the LMP_channel_classification_req PDU with the AFH_reporting_mode parameter set to AFH_reporting_disabled.



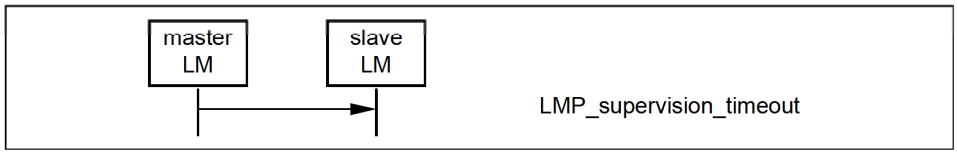
Sequence 8: Channel classification reporting.

4.1.6 Link supervision

Each physical link has a timer that is used for link supervision. This timer is used to detect physical link loss caused by devices moving out of range, or being blocked by interference, a device's power-down, or other similar failure cases. Link supervision is specified in Baseband Specification, Section 3.1, on page 95.

M/O	PDU	Contents
M	LMP_supervision_timeout	supervision timeout

Table 4.6: PDU used to set the supervision timeout.



Sequence 9: Setting the link supervision timeout.

4.1.7 Channel quality driven data rate change (CQDDR)

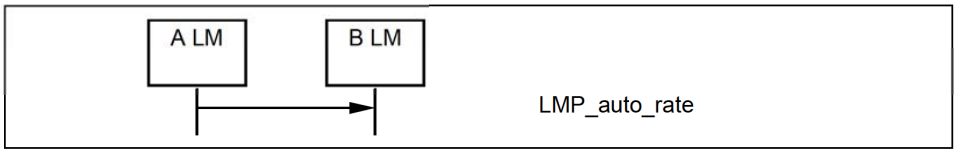
The data throughput for a given packet type depends on the quality of the RF channel. Quality measurements in the receiver of one device can be used to dynamically control the packet type transmitted from the remote device for optimization of the data throughput. Device A sends the LMP_auto_rate PDU once to notify device B to enable this feature. Once enabled, device B may change the packet type(s) that A transmits by sending the LMP_preferred_rate PDU. This PDU has a parameter which determines the preferred coding (with or without 2/3FEC) and optionally the preferred size in slots of the packets. Device A is not required to change to the packet type specified by this parameter. Device A shall not send a packet that is larger than max slots (see Section 4.1.10 on page 247) even if the preferred size is greater than this value.

The data rate parameter includes the preferred rate for Basic Rate and Enhanced Data Rate modes. When operating in Basic Rate mode, the device shall use bits 0-2 to determine the preferred data rate. When operating in Enhanced Data Rate mode, the device shall use bits 3-6 to determine the preferred data rate. For devices that support Enhanced Data Rate, the preferred rates for both Basic Rate and Enhanced Data Rate modes shall be valid at all times.

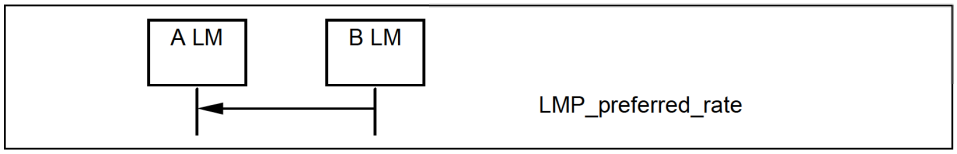
These PDUs may be sent at any time after connection setup is completed.

M/O	PDU	Contents
O(10)	LMP_auto_rate	-
O(10)	LMP_preferred_rate	data rate

Table 4.7: PDUs used for quality driven change of the data rate.



Sequence 10: A notifies B to enable CQDDR



Sequence 11: B sends A a preferred packet type

4.1.8 Quality of service (QoS)

The LM provides QoS capabilities. A poll interval, T_{poll}, that is defined as the maximum time between transmissions from the master to a particular slave on the ACL logical transport, is used to support bandwidth allocation and latency control - see Baseband Specification, Section 8.6.1, on page 169 for details. The poll interval is guaranteed in the active and sniff modes except when there are collisions with page, page scan, inquiry and inquiry scan, during time critical LMP sequences in the current piconet and any other piconets in which the Bluetooth device is a member, and during critical baseband sequences (such as the page response, initial connection state until the first POLL, and master slave switch). These PDUs maybe sent at anytime after connection setup is completed.

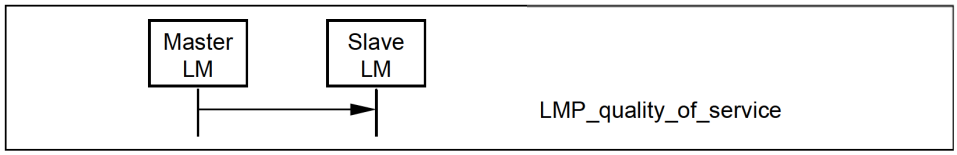
Master and slave negotiate the number of repetitions for broadcast packets (N_{BC}), see Baseband Specification, Section 7.6.5, on page 150.

M/O	PDU	Contents
M	LMP_quality_of_service	poll interval N_{BC}
M	LMP_quality_of_service_req	poll interval N_{BC}

Table 4.8: PDUs used for quality of service.

4.1.8.1 Master notifies slave of the quality of service

The master notifies the slave of the new poll interval and N_{BC} by sending the LMP_quality_of_service PDU. The slave cannot reject the notification.

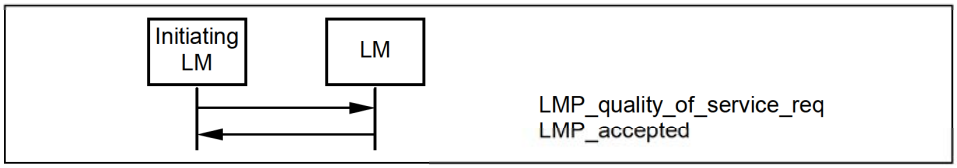


Sequence 12: Master notifies slave of quality of service.

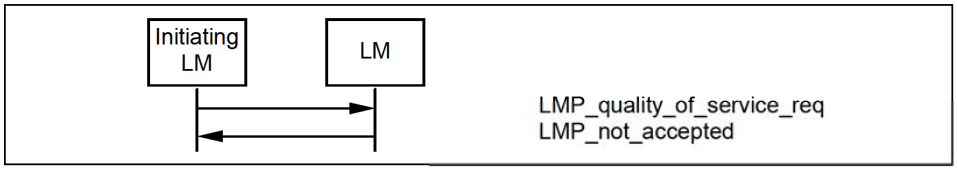
4.1.8.2 Device requests new quality of service

Either the master or the slave may request a new poll interval and N_{BC} by sending an LMP_quality_of_service_req PDU to the slave. The parameter N_{BC} is meaningful only when it is sent by a master to a slave. For transmission of LMP_quality_of_service_req PDUs from a slave, this parameter shall be ignored by the master. The request can be accepted or rejected. This allows the master and slave to dynamically negotiate the quality of service as needed.

The selected poll interval by the slave shall be less than or equal to the specified Access Latency for the outgoing traffic of the ACL link (see L2CAP "Quality of Service (QoS) Option" on page 60[vol. 4]).



Sequence 13: Device accepts new quality of service



Sequence 14: Device rejects new quality of service.

4.1.9 Paging scheme parameters

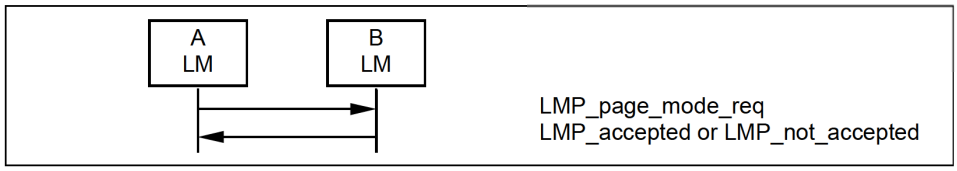
LMP provides a means to negotiate the paging scheme parameters that are used the next time a device is paged.

M/O	PDU	Contents
O(17)	LMP_page_mode_req	paging scheme paging scheme settings
O(17)	LMP_page_scan_mode_req	paging scheme paging scheme settings

Table 4.9: PDUs used to request paging scheme.

4.1.9.1 Page mode

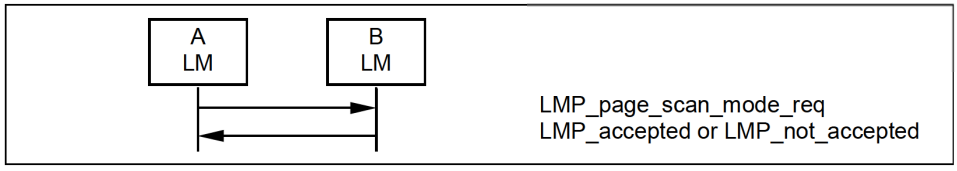
This procedure is initiated from device A and negotiates the paging scheme used when device A pages device B. Device A proposes a paging scheme including the parameters for this scheme and device B can accept or reject. On rejection the old setting will not be changed. A request to switch to a reserved paging scheme shall be rejected.



Sequence 15: Negotiation for page mode.

4.1.9.2 Page scan mode

This procedure is initiated from device A and negotiates the paging scheme and paging scheme settings used when device B pages device A. Device A proposes a paging scheme and paging scheme settings and device B may accept or reject. On reject the old setting is not changed. A request specifying the mandatory scheme shall be accepted. A request specifying a non-mandatory scheme shall be rejected. This procedure should be used when device A changes its paging scheme settings. A slave should also send this message to the master after connection establishment, to inform the master of the slave's current paging scheme and paging scheme settings.



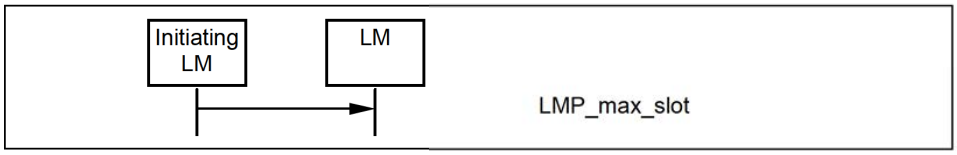
Sequence 16: Negotiation for page scan mode

4.1.10 Control of multi-slot packets

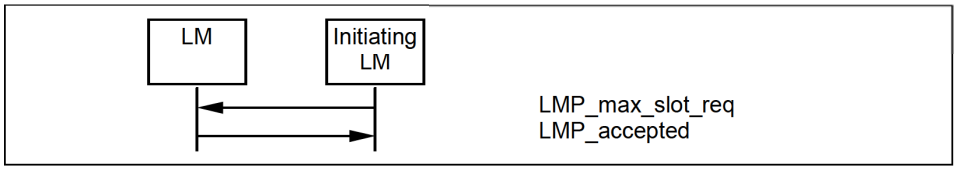
The number of consecutive slots used by a device on an ACL-U logical link can be limited. It does not affect traffic on the eSCO links where the packet sizes are defined as part of link setup. A device allows the remote device to use a maximum number of slots by sending the PDU LMP_max_slot providing max slots as parameter. Each device can request to use a maximal number of slots by sending the PDU LMP_max_slot_req providing max slots as parameter. After a new connection, as a result of page, page scan, role switch or unpark, the default value is 1 slot. These PDUs can be sent at anytime after connection setup is completed.

M/O	PDU	Contents
M	LMP_max_slot	max slots
M	LMP_max_slot_req	max slots

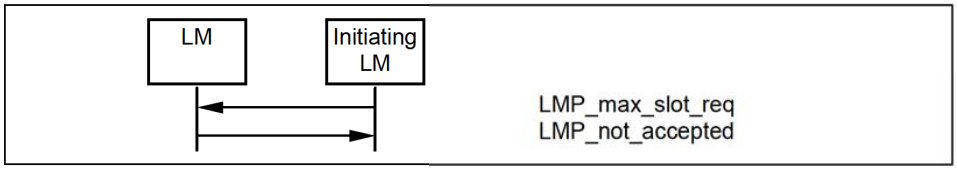
Table 4.10: PDUs used to control the use of multi-slot packets.



Sequence 17: Device allows Remote Device to use a maximum number of slots.



Sequence 18: Device requests a maximum number of slots. Remote Device accepts.



Sequence 19: Device requests a maximum number of slots. Remote Device rejects.

4.1.11 Enhanced Data Rate

A device may change the packet type table, ptt, to select which if any of the optional modulation schemes are to be used on an ACL logical transport.

Either the master or the slave may request a new packet type table and therefore the modulation scheme to be used on this ACL link. After a new baseband connection, as a result of page or page scan, the default value for ptt shall be 0.

The change of the modulation mode for an ACL logical transport shall not affect the packet types used for an associated SCO logical transport on the same LT_ADDR.

Note: Enhanced Data Rate eSCO links are negotiated using the LMP eSCO link_req as described in section 4.6.2.

Before changing the packet type table, the initiator shall finalize the transmission of the current ACL packet with ACL-U information and shall stop ACL-U transmissions. It shall then send the LMP_packet_type_table_req PDU.

If the receiver rejects the change, then it shall respond with an LMP_not_accepted_ext PDU.

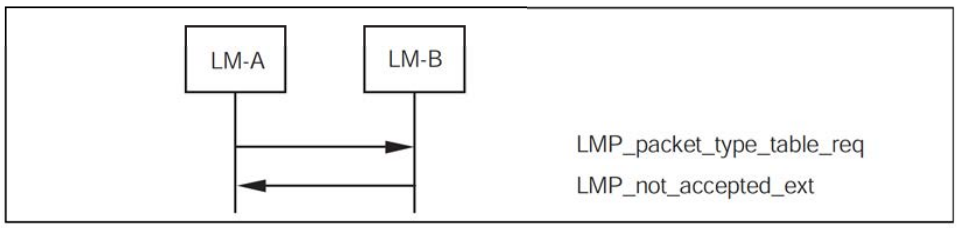
If the receiver accepts the change, then it shall finalize the transmission of the current ACL packet with ACL-U information and shall stop ACL-U transmissions, it shall change to the new packet type table and shall respond with an LMP_accepted_ext PDU. When it receives the baseband level acknowledgement for the LMP_accepted_ext PDU it shall restart ACL-U transmissions.

When the initiator receives an LMP_not_accepted_ext PDU the initiator shall restart ACL-U transmissions.

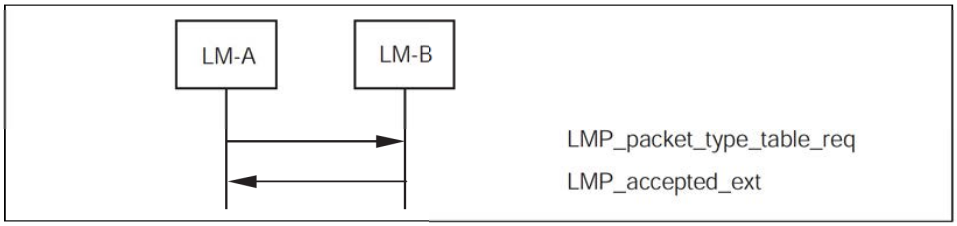
When the initiator receives an LMP_accepted_ext PDU it shall change the packet type table and restart ACL-U transmissions.

M/O	PDU	Contents
O(25)	LMP_packet_type_table_req	packet type table

Table 4.11: PDUs used for Enhanced Data Rate



Sequence 20: Packet type table change is rejected.



Sequence 21: Packet type table change is accepted.

4.2 SECURITY

4.2.1 Authentication

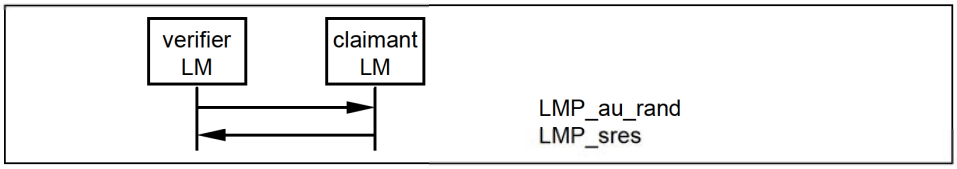
The authentication procedure is based on a challenge-response scheme as described in [Part H] Section 3.2.2 on page 780. The verifier sends an LMP_au_rand PDU that contains a random number (the challenge) to the claimant. The claimant calculates a response, that is a function of this challenge, the claimant's BD_ADDR and a secret key. The response is sent back to the verifier, that checks if the response was correct or not. The response shall be calculated as described in [Part H] Section 6.1 on page 801. A successful calculation of the authentication response requires that two devices share a secret key. This key is created as described in Section 4.2.2 on page 251. Both the master and the slave can be verifiers.

M/O	PDU	Contents
M	LMP_au_rand	random number
M	LMP_sres	authentication response

Table 4.12: PDUs used for authentication.

4.2.1.1 Claimant has link key

If the claimant has a link key associated with the verifier, it shall calculate the response and sends it to the verifier with LMP_sres. The verifier checks the response. If the response is not correct, the verifier can end the connection by sending an LMP_detach PDU with the error code *authentication failure,* see Section 4.1.2 on page 234.

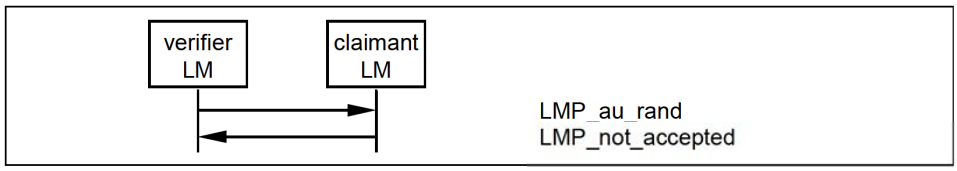


Sequence 22: Authentication. Claimant has link key.

Upon reception of an LMP_au_rand, an LM shall reply with LMP_sres before initiating its own authentication.
Note: there can be concurrent requests caused by the master and slave simultaneously initiating an authentication. The procedures in Section 2.5.1 on page 223 assures that devices will not have different Authenticated Ciphering Offset (ACO, see [Part H] Section 6.1 on page 801) when they calculate the encryption key.

4.2.1.2 Claimant has no link key

If the claimant does not have a link key associated with the verifier it shall send an LMP_not_accepted PDU with the error code *key missing* after receiving an LMP_au_rand PDU.



Sequence 23: Authentication fails. Claimant has no link key.

4.2.1.3 Repeated attempts

The scheme described in [Part H] Section 5.1 on page 799 shall be applied when an authentication fails. This will prevent an intruder from trying a large number of keys in a relatively short time.

4.2.2 Pairing

When two devices do not have a common link key an initialization key (K_{init}) shall be created based on a PIN, and a random number and a BD_ADDR. K_{init} shall be created as specified in [Part H] Section 6.3 on page 805. When both devices have calculated K_{init} the link key shall be created, and a mutual authentication is performed. The pairing procedure starts with a device sending an LMP_in_rand PDU; this device is referred to as the "initiating LM" or "initiator" in Section 4.2.2.1 on page 251 - Section 4.2.2.5 on page 253. The other device is referred to as the "responding LM" or "responder". The PDUs used in the pairing procedure are:

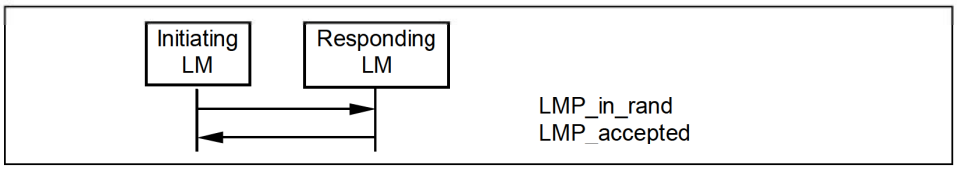
M/O	PDU	Contents
M	LMP_in_rand	random number
M	LMP_au_rand	random number
M	LMP_sres	authentication response
M	LMP_comb_key	random number
M	LMP_unit_key	key

Table 4.13: PDUs used for pairing

All sequences described in Section 4 on page 233, including the mutual authentication after the link key has been created, shall form a single transaction. The transaction ID from the first LMP_in_rand shall be used for all subsequent sequences.

4.2.2.1 Responder accepts pairing

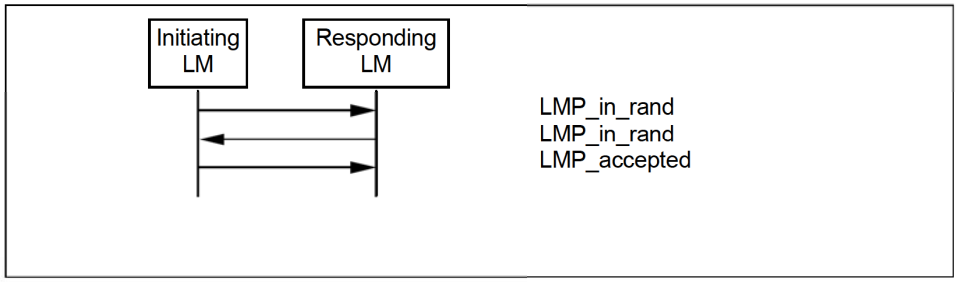
When the initiator sends an LMP_in_rand PDU and the responder shall reply with an LMP_accepted PDU. Both devices shall then calculate K_{init} based on the BD_ADDR of the responder and the procedure continues with creation of the link key; see Section 4.2.2.4 on page 253.



Sequence 24: Pairing accepted. Responder has a variable PIN. Initiator has a variable or fixed PIN.

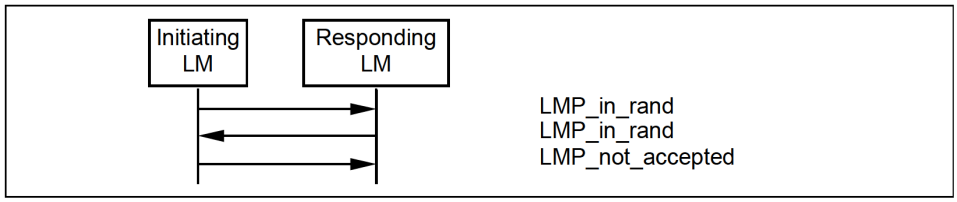
4.2.2.2 Responder has a fixed PIN

If the responder has a fixed PIN it shall generate a new random number and send it back in an LMP_in_rand PDU. If the initiator has a variable PIN it shall accept the LMP_in_rand PDU and shall respond with an LMP_accepted PDU. Both sides shall then calculate K_{init} based on the last IN_RAND and the BD_ADDR of the initiator. The procedure continues with creation of the link key; see Section 4.2.2.4 on page 253.



Sequence 25: Responder has a fixed PIN and initiator has a variable PIN.

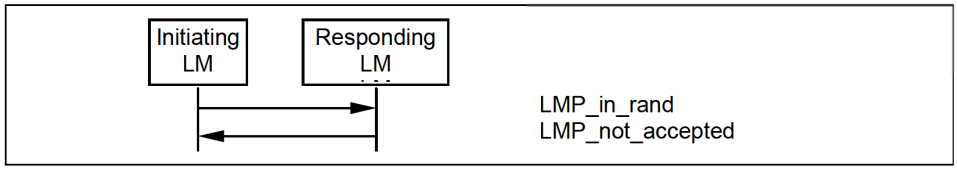
If the responder has a fixed PIN and the initiator also has a fixed PIN, the second LMP_in_rand shall be rejected by the initiator sending an LMP_not_accepted PDU with the error code *pairing not allowed*.



Sequence 26: Both devices have a fixed PIN.

4.2.2.3 Responder rejects pairing

If the responder rejects pairing it shall send an LMP_not_accepted PDU with the error code *pairing not allowed* after receiving an LMP_in_rand PDU.



Sequence 27: Responder rejects pairing.

4.2.2.4 Creation of the link key

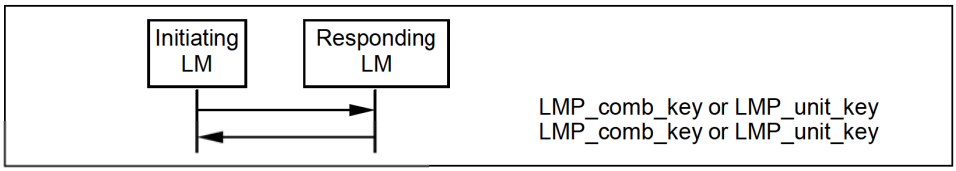
When K_{init} is calculated in both devices the link key shall be created. This link key will be used in the authentication between the two devices for all subsequent connections until it is changed; see Section 4.2.3 on page 254 and Section 4.2.4 on page 255. The link key created in the pairing procedure will either be a combination key or one of the device's unit keys. The following rules shall apply to the selection of the link key:

- if one device sends an LMP_unit_key PDU and the other device sends LMP_comb_key, the unit key will be the link key.
- if both devices send an LMP_unit_key PDU, the master's unit key will be the link key.
- if both devices send an LMP_comb_key PDU, the link key shall be calculated as described in [Part H] Section 3.2 on page 779.

The content of the LMP_unit_key PDU is the unit key bitwise XORed with K_{init}. The content of the LMP_comb_key PDU is LK_RAND bitwise XORed with K_{init}. Any device configured to use a combination key shall store the link key.

The use of unit keys is deprecated since it is implicitly insecure.

When the link key, combination or unit key, has been created mutual authentication shall be performed to confirm that the same link key has been created in both devices. The first authentication in the mutual authentication is performed with the initiator as the verifier. When finalized an authentication in the reverse direction is performed.



Sequence 28: Creation of the link key.

4.2.2.5 Repeated attempts

When the authentication after creation of the link key fails because of an incorrect authentication response, the same scheme as in Section 4.2.1.3 on page 250 shall be used. This prevents an intruder from trying a large number of different PINs in a relatively short time.

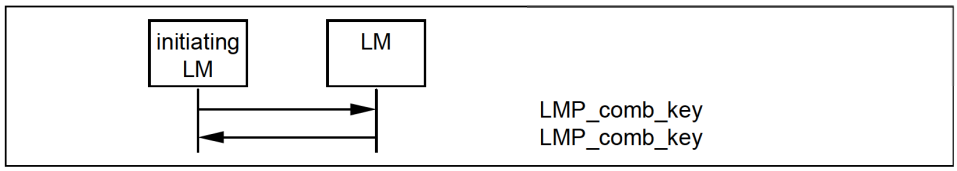
4.2.3 Change link key

If the link key is derived from combination keys and the current link is the semi-permanent link key, the link key can be changed. If the link key is a unit key, the devices shall go through the pairing procedure in order to change the link key. The contents of the LMP_comb_key PDU is protected by a bitwise XOR with the current link key.

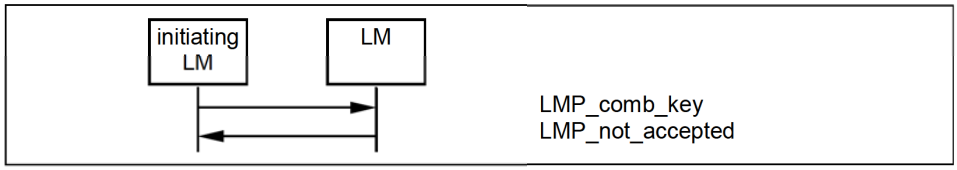
M/O	PDU	Contents
M	LMP_comb_key	random number

Table 4.14: PDUs used for change of link key.

All sequences described in Section 4.2.3 , including the mutual authentication after the link key has been changed, shall form a single transaction. The transaction ID from the first LMP_comb_key PDU shall be used for all subsequent sequences.



Sequence 29: Successful change of the link key.



Sequence 30: Change of the link key not possible since the other device uses a unit key.

If the change of link key is successful the new link key shall be stored and the old link key shall be discarded. The new link key shall be used as link key for all the following connections between the two devices until the link key is changed again. The new link key also becomes the current link key. It will remain the current link key until the link key is changed again, or until a temporary link key is created, see Section 4.2.4 on page 255.

When the new link key has been created mutual authentication shall be performed to confirm that the same link key has been created in both devices. The first authentication in the mutual authentication is performed with the device that initiated change link key as verifier. When finalized an authentication in the reverse direction is performed.

4.2.4 Change current link key type

The current link key can be a semi-permanent link key or a temporary link key. It may be changed temporarily, but the change shall only be valid for the current connection, see [Part H] Section 3.1 on page 777. Changing to a temporary link key is necessary if the piconet is to support encrypted broadcast. The current link key may not be changed before the connection establishment procedure has completed. This feature is only supported if broadcast encryption is supported as indicated by the LMP features mask.

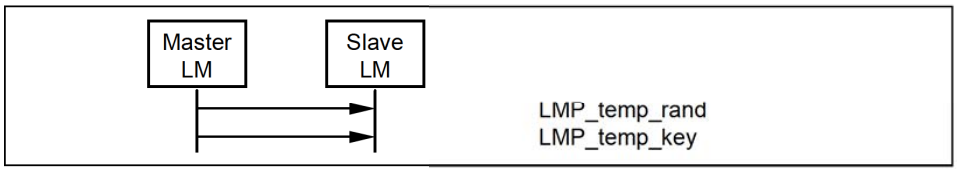
M/O	PDU	Contents
O(23)	LMP_temp_rand	random number
O(23)	LMP_temp_key	key
O(23)	LMP_use_semi_permanent_key	-

Table 4.15: PDUs used to change the current link key.

4.2.4.1 Change to a temporary link key

The master starts by creating the master key K_{master} as specified in Security Specification (EQ 4), on page 784. Then the master shall generate a random number, RAND, and shall send it to the slave in an LMP_temp_rand PDU. Both sides then calculate an overlay denoted OVL as OVL= E_{22}(current link key, RAND, 16). The master shall then send K_{master} protected by a modulo-2 addition with OVL to the slave in an LMP_temp_key PDU. The slave calculates K_{master}, based on OVL, that becomes the current link key. It shall be the current link key until the devices fall back to the semi-permanent link key, see section 4.2.4.2 on page 256.

Note: the terminology in this section is the same as used in [Part H] Section 3.2.8 on page 784.



Sequence 31: Change to a temporary link key.

All sequences described in Section 4.2.4.1 on page 255, including the mutual authentication after K_{master} has been created, shall form a single transaction. The transaction ID shall be set to 0.

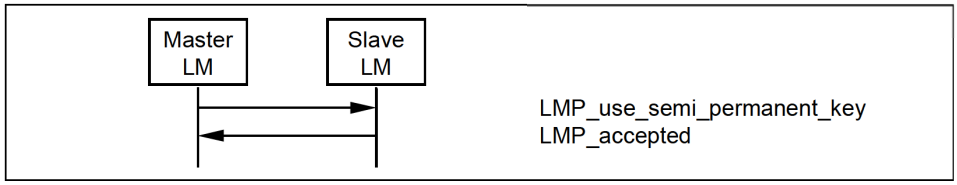
When the devices have changed to the temporary key, a mutual authentication shall be made to confirm that the same link key has been created in both devices. The first authentication in the mutual authentication shall be per-

formed with the master as verifier. When finalized an authentication in the reverse direction is performed.

Should the mutual authentication fail at either side, the LM of the verifier should start the detach procedure. This will allow the procedure to succeed even though one of the devices may be erroneous.

4.2.4.2 Make the semi-permanent link key the current link key

After the current link key has been changed to K_{master}, this change can be undone and the semi-permanent link key becomes the current link key again. If encryption is used on the link, the procedure to go back to the semi-permanent link key shall be immediately followed by the master stopping encryption using the procedure described in Section 4.2.5.4 on page 260. Encryption may be restarted by the master according to the procedures in section 4.2.5.1 on page 257 subsection 3. This is to assure that encryption with encryption parameters known by other devices in the piconet is not used when the semi-permanent link key is the current link key.



Sequence 32: Link key changed to the semi-permanent link key.

4.2.5 Encryption

If at least one authentication has been performed encryption may be used. In order for the master to use the same encryption parameters for all slaves in the piconet it shall issue a temporary key, K_{master}. The master shall make this key the current link key for all slaves in the piconet before encryption is started, see Section 4.2.4 on page 255. This is required if broadcast packets are to be encrypted.

M/O	PDU	Contents
O	LMP_encryption_mode_req	encryption mode
O	LMP_encryption_key_size_req	key size
O	LMP_start_encryption_req	random number
O	LMP_stop_encryption_req	-

Table 4.16: PDUs used for handling encryption.

All sequences described in Section 4.2.5 shall form a single transaction. The transaction ID from the LMP_encryption_mode_req PDU shall be used for all subsequent sequences.

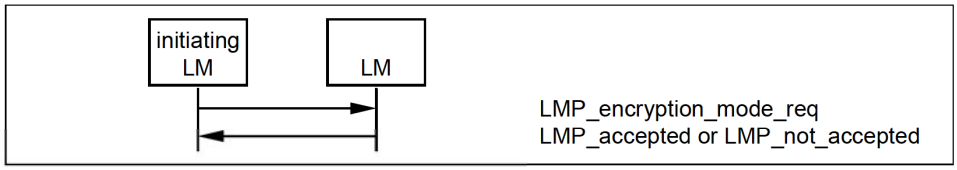
4.2.5.1 Encryption mode

The master and the slave must agree upon whether to use encryption (encryption mode=1 in lmp_encryption_mode_req) or not (encryption mode=0). If the semi-permanent key is used (Key_Flag=0x00) encryption shall only apply to point-to-point packets. If the master link key is used (Key_Flag=0x01) encryption shall apply to both point-to-point packets and broadcast packets. If master and slave agree on the encryption mode, the master continues to give more detailed information about the encryption.

Devices should never send LMP_encryption_mode_req with an encryption mode value of 2 however for backwards compatibility if the LMP_encryption_mode_req is received with an encryption mode value of 2 then it should be treated the same as an encryption mode value of 1.

The initiating LM shall pause traffic on the ACL-U logical link (see Baseband Specification, Section 5.3.1, on page 108). The initiating device shall then send the LMP_encryption_mode_req PDU. If the responding device accepts the change in encryption mode then it shall complete the transmission of the current packet on the ACL logical transport and shall then suspend transmission on the ACL-U logical link. The responding device shall then send the LMP_accepted PDU.

ACL-U logical link traffic shall only be resumed after the attempt to encrypt or decrypt the logical transport is completed i.e. at the end of Sequence 33, 34 or 35.

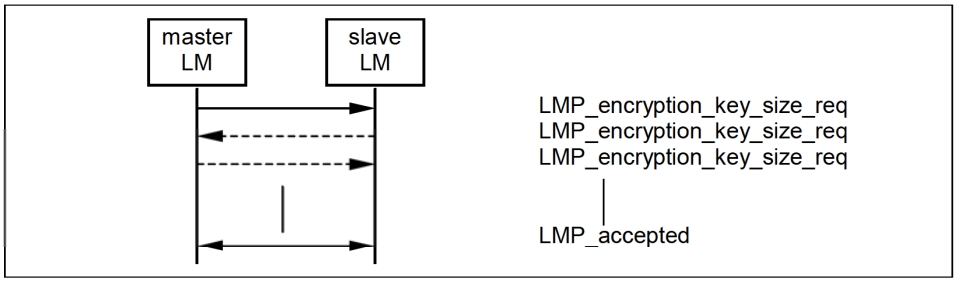


Sequence 33: Negotiation for encryption mode.

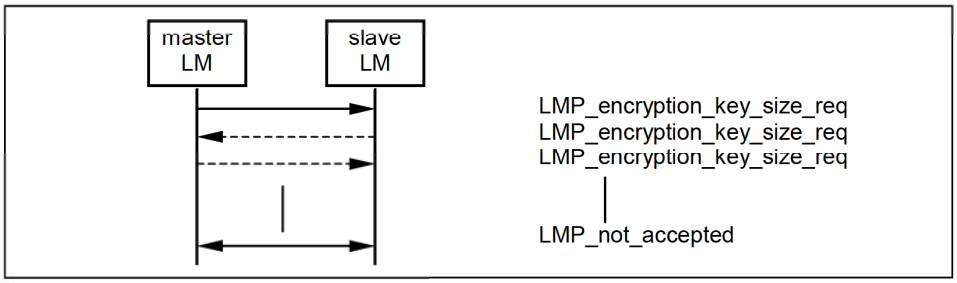
After a device has sent an LMP_encryption_mode_req PDU it shall not send an LMP_au_rand PDU before encryption is started. After a device has received an LMP_encryption_mode_req PDU and sent an LMP_accepted PDU it shall not send an LMP_au_rand PDU before encryption is started. If an LMP_au_rand PDU is sent violating these rules, the claimant shall respond with an LMP_not_accepted PDU with the error code *PDU not allowed*. This assures that devices will not have different ACOs when they calculate the encryption key. If the encryption mode is not accepted or the encryption key size negotiation results in disagreement the devices may send an LMP_au_rand PDU again.

4.2.5.2 Encryption key size

Note: this section uses the same terms as in [Part H] Section 4.1 on page 788. The master sends an LMP_encryption_key_size_req PDU including the suggested key size L_{sug}, m, that is initially equal to $L_{max, m}$. If $L_{min, s} \leq L_{sug, m}$ and the slave supports $L_{sug, m}$ it shall respond with an LMP_accepted PDU and $L_{sug, m}$ shall be used as the key size. If both conditions are not fulfilled the slave sends back an LMP_encryption_key_size_req PDU including the slave's suggested key size $L_{sug, s}$. This value shall be the slave's largest supported key size that is less than $L_{sug, m}$. Then the master performs the corresponding test on the slave's suggestion. This procedure is repeated until a key size agreement is reached or it becomes clear that no such agreement can be reached. If an agreement is reached a device sends an LMP_accepted PDU and the key size in the last LMP_encryption_key_size_req PDU shall be used. After this, encryption is started; see Section 4.2.5.3 on page 259. If an agreement is not reached a device sends an LMP_not_accepted PDU with the error code *unsupported parameter value* and the devices shall not communicate using encryption.



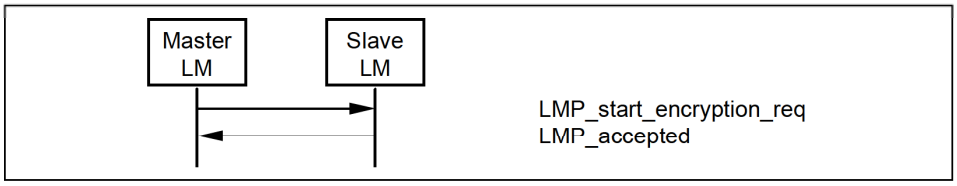
Sequence 34: Encryption key size negotiation successful.



Sequence 35: Encryption key size negotiation failed.

4.2.5.3 Start encryption

To start encryption, the master issues the random number EN_RAND and calculates the encryption key. See [Part H] Section 3.2.5 on page 782. The random number shall be the same for all slaves in the piconet when broadcast encryption is used. The master then sends an LMP_start_encryption_req PDU, that includes EN_RAND. The slave shall calculate the encryption key when this message is received and shall acknowledge with an LMP_accepted PDU.



Sequence 36: Start of encryption.

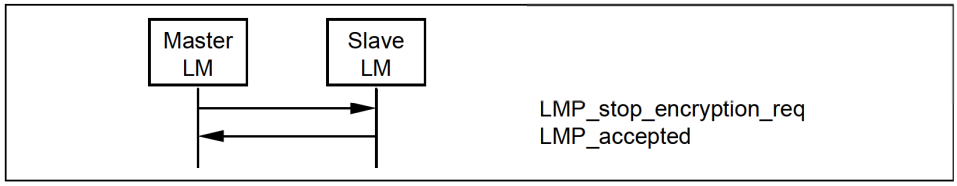
Starting encryption shall be performed in three steps:

1. Master is configured to transmit unencrypted packets and to receive encrypted packets.
2. Slave is configured to transmit and receive encrypted packets.
3. Master is configured to transmit and receive encrypted packets.

Between step 1 and step 2, master-to-slave transmission is possible. This is when an LMP_start_encryption_req PDU is transmitted. Step 2 is triggered when the slave receives this message. Between step 2 and step 3, slave-to-master
transmission is possible. This is when an LMP_accepted PDU is transmitted. Step 3 is triggered when the master receives this message.

4.2.5.4 Stop encryption

To stop encryption a device shall send an LMP_encryption_mode_req PDU with the parameter encryption mode equal to 0 (no encryption). The other device responds with an LMP_accepted PDU or an LMP_not_accepted PDU (the procedure is described in Sequence 33 in section 4.2.5.1 on page 257). If accepted, encryption shall be stopped by the master sending an LMP_stop_encryption_req PDU and the slave shall respond with an LMP_accepted PDU according to Sequence 37.



Sequence 37: Stop of encryption.

Stopping encryption shall be performed in three steps, similar to the procedure for starting encryption.

1. Master is configured to transmit encrypted packets and to receive unencrypted packets.
2. Slave is configured to transmit and receive unencrypted packets.
3. Master is configured to transmit and receive unencrypted packets.

Between step 1 and step 2 master to slave transmission is possible. This is when an LMP_stop_encryption_req PDU is transmitted. Step 2 is triggered when the slave receives this message. Between step 2 and step 3 slave to master transmission is possible. This is when an LMP_accepted PDU is transmitted. Step 3 is triggered when the master receives this message.

4.2.5.5 Change encryption mode, key or random number

If the encryption key or encryption random number need to be changed or if the current link key needs to be changed according to the procedures in Section 4.2.4 on page 255, encryption shall be stopped and re-started after completion, using the procedures in Section 4.2.5 on page 257, subsections 3-4 for the new parameters to take effect.

4.2.6 Request supported encryption key size

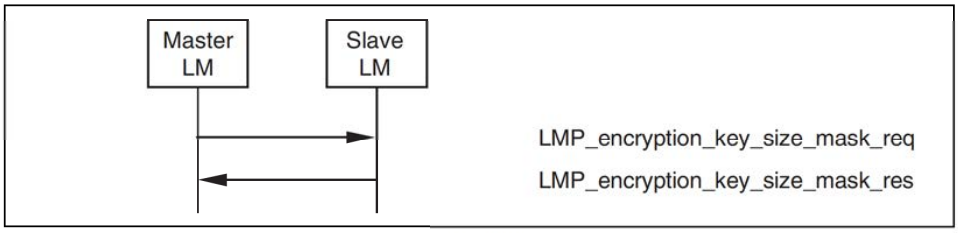
When broadcast encryption is supported via the LMP features mask, it is possible for the master to request a slave's supported encryption key sizes.

M/O	PDU	Contents
O(23)	LMP_encryption_key_size_mask_req	
O(23)	LMP_encryption_key_size_mask_res	key size mask

Table 4.17: PDUs used for encryption key size request

The master shall send an LMP_key_size_req PDU to the slave to obtain the slaves supported encryption key sizes.

The slave shall return a bit mask indicating all broadcast encryption key sizes supported. The least significant bit shall indicate support for a key size of 1, the next most significant bit shall indicate support for a key size of 2 and so on up to a key size of 16. In all cases a bit set to 1 shall indicate support for a key size; a bit set to 0 shall indicate that the key size is not supported.



Sequence 38: Request for supported encryption key sizes.

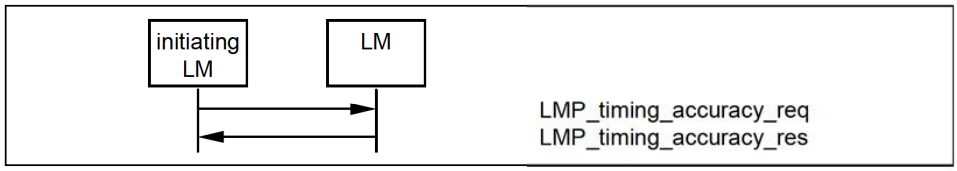
4.3 INFORMATIONAL REQUESTS

4.3.1 Timing accuracy

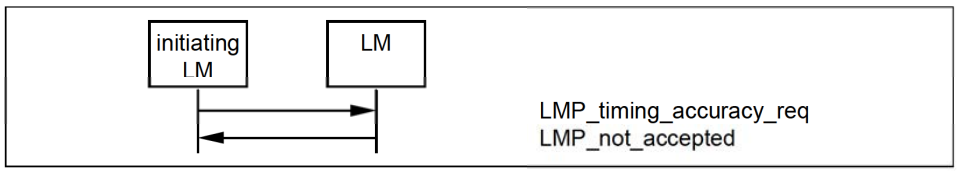
LMP supports requests for the timing accuracy. This information can be used to minimize the scan window during piconet physical channel re-synchronization (see Baseband Specification, Section 2.2.5.2, on page 74). The timing accuracy parameters returned are the long term drift measured in ppm and the long term jitter measured in μs of the worst case clock used. These parameters are fixed for a certain device and shall be identical when requested several times. Otherwise, the requesting device shall assume worst case values (drift=250ppm and jitter=10μs).

M/O	PDU	Contents
O(4)	LMP_timing_accuracy_req	-
O(4)	LMP_timing_accuracy_res	drift jitter

Table 4.18: PDUs used for requesting timing accuracy information.



Sequence 39: The requested device supports timing accuracy information.



Sequence 40: The requested device does not support timing accuracy information.

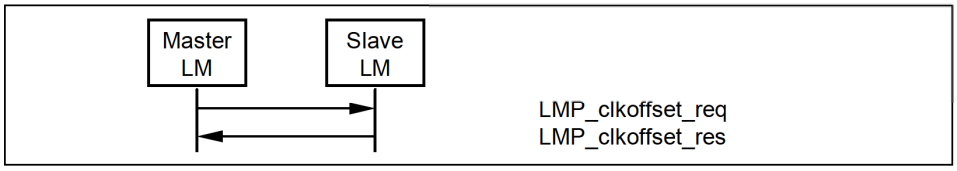
4.3.2 Clock offset

The clock offset can be used to speed up the paging time the next time the same device is paged. The master can request the clock offset at anytime following a successful baseband paging procedure (i.e., before, during or after connection setup). The clock offset shall be defined by the following equation:

$(CLKN_{16\text{-}2\ slave} - CLKN_{16\text{-}2\ master})\ mod\ 2^{**}15.$

M/O	PDU	Contents
M	LMP_clkoffset_req	-
M	LMP_clkoffset_res	clock offset

Table 4.19: PDUs used for clock offset request.



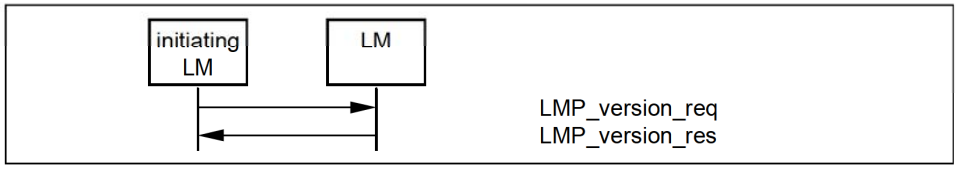
Sequence 41: Clock offset requested.

4.3.3 LMP version

LMP supports requests for the version of the LM protocol. The LMP_version_req and LMP_version_res PDUs contain three parameters: VersNr, CompId and SubVersNr. VersNr specifies the version of the Bluetooth LMP specification that the device supports. CompId is used to track possible problems with the lower Bluetooth layers. All companies that create a unique implementation of the LM shall have their own CompId. The same company is also responsible for the administration and maintenance of the SubVersNr. It is recommended that each company has a unique SubVersNr for each RF/BB/LM implementation. For a given VersNr and CompId, the values of the SubVersNr shall increase each time a new implementation is released. For both CompId and SubVersNr the value 0xFFFF means that no valid number applies. There is no ability to negotiate the version of the LMP. The sequence below is only used to exchange the parameters. LMP version can be requested at anytime following a successful baseband paging procedure.

M/O	PDU	Contents
M	LMP_version_req	VersNr CompId SubVersNr
M	LMP_version_res	VersNr CompId SubVersNr

Table 4.20: PDUs used for LMP version request.



Sequence 42: Request for LMP version.

4.3.4 Supported features

The supported features may be requested at anytime following a successful baseband paging procedure by sending the LMP_features_req PDU. Upon reception of an LMP_features_req PDU, the receiving device shall return an LMP_features_res PDU.

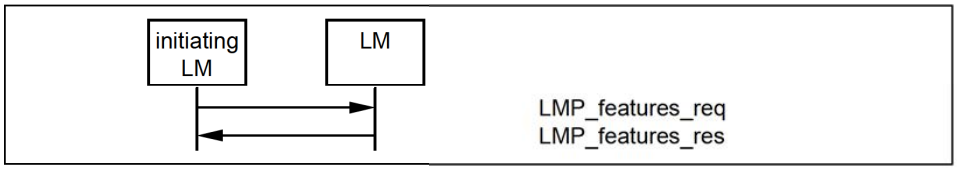
The number of features bits required will in the future exceed the size of a single page of features. An extended features mask is therefore provided to allow support for more than 64 features. Support for the extended features mask is indicated by the presence of the appropriate bit in the LMP features mask. The LMP_features_req_ext and LMP_features_res_ext PDUs operate in precisely the same way as the LMP_features_req and LMP_features_res PDUs except that they allow the various pages of the extended features mask to be requested. The LMP_features_req_ext may be sent at any time following the exchange of the LMP_features_req and LMP_features_rsp PDUs.

The LMP_features_req_ext PDU contains a feature page index that specifies which page is requested and the contents of that page for the requesting device. Pages are numbered from 0-255 with page 0 corresponding to the normal features mask. Each page consists of 64 bits. If a device does not support any page number it shall return a mask with every bit set to 0. It also contains the maximum features page number containing any non-zero bit for this device. The recipient of an LMP_features_req_ext PDU shall respond with an LMP_features_res_ext PDU containing the same page number and the appropriate features page along with its own maximum features page number.

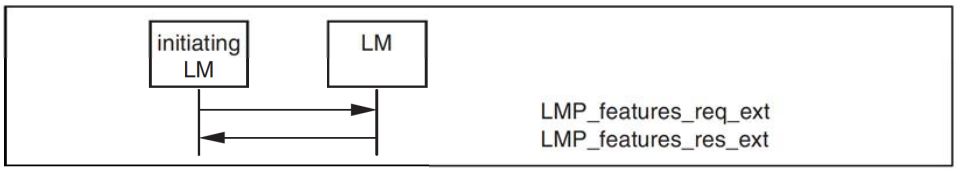
If the extended features request is not supported then all bits in all extended features pages for that device shall be assumed to be zero.

M/O	PDU	Contents
M	LMP_features_req	features
M	LMP_features_res	features
O(63)	LMP_features_req_ext	features page max supported page extended features
O(63)	LMP_features_res_ext	features page max supported page extended features

Table 4.21: PDUs used for features request.



Sequence 43: Request for supported features.



Sequence 44: Request for extended features.

4.3.5 Name request

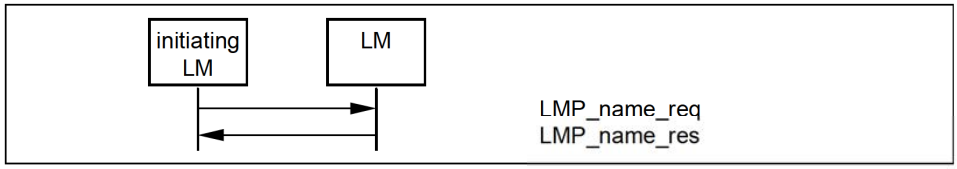
LMP supports name request to another device. The name is a user-friendly name associated with the device and consists of a maximum of 248 bytes coded according to the UTF-8 standard. The name is fragmented over one or more DM1 packets. When an LMP_name_req PDU is sent, a name offset indicates which fragment is expected. The corresponding LMP_name_res PDU carries the same name offset, the name length indicating the total number of bytes in the name of the device and the name fragment, where:

- name fragment(N) = name(N + name offset), if (N + name offset) < name length
- name fragment(N) = 0,otherwise.

Here $0 \leq N \leq 13$. In the first sent LMP_name_req PDU, name offset=0. Sequence 45 is then repeated until the initiator has collected all fragments of the name. The name request may be made at any time following a successful baseband paging procedure.

M/O	PDU	Contents
M	LMP_name_req	name offset
M	LMP_name_res	name offset name length name fragment

Table 4.22: PDUs used for name request.



Sequence 45: Device's name requested and it responses.

4.4 ROLE SWITCH

4.4.1 Slot offset

With LMP_slot_offset the information about the difference between the slot boundaries in different piconets is transmitted. The LMP_slot_offset PDU may be sent anytime after the baseband paging procedure has completed. This PDU carries the parameters slot offset and BD_ADDR. The slot offset shall be the time in microseconds between the start of a master transmission in the current piconet to the start of the next following master transmission in the piconet where the BD_ADDR device (normally the slave) is master at the time that the request is interpreted by the BD_ADDR device.

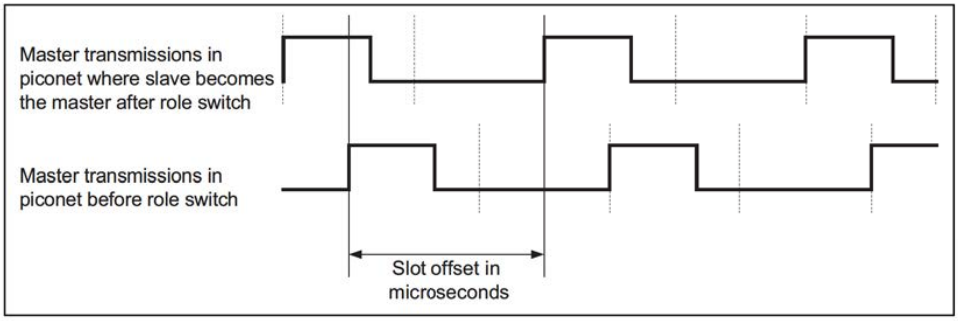
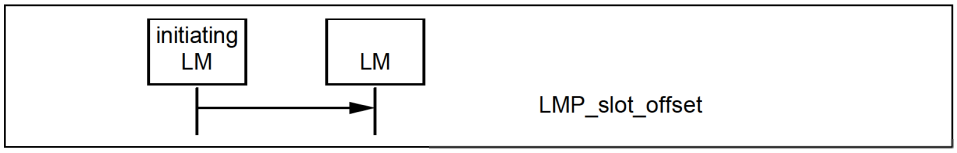


Figure 4.2: Slot offset for role switch.

See Section 4.4 on page 267 for the use of LMP_slot_offset in the context of the role switch. In the case of role switch the BD_ADDR is that of the slave device.

M/O	PDU	Contents
O(3)	LMP_slot_offset	slot offset BD_ADDR

Table 4.23: PDU used for slot offset information.



Sequence 46: Slot offset information is sent.

4.4.2 Role switch

Since the paging device always becomes the master of the piconet, a switch of the master slave role is sometimes needed, see Baseband Specification, Section 8.6.5, on page 175. The LMP_switch_req PDU may be sent anytime after the baseband paging procedure has completed.

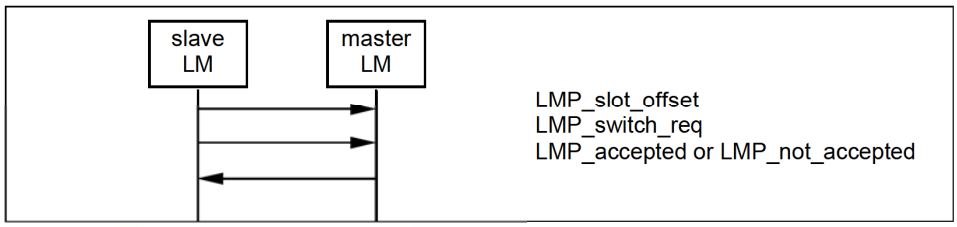
Support for LMP_slot_offset is mandatory if LMP_switch_req is supported.

The LMP_slot_offset shall be sent only if the ACL logical transport is in active mode. The LMP_switch_req shall be sent only if the ACL logical transport is in active mode, when encryption is disabled, and all synchronous logical transports on the same physical link are disabled. Additionally, LMP_slot_offset or LMP_switch_req shall not be initiated or accepted while a synchronous logical transport is being negotiated by LM.

M/O	PDU	Contents
O(5)	LMP_switch_req	switch instant
O(5)	LMP_slot_offset	slot offset BD_ADDR

Table 4.24: PDUs used for role switch.

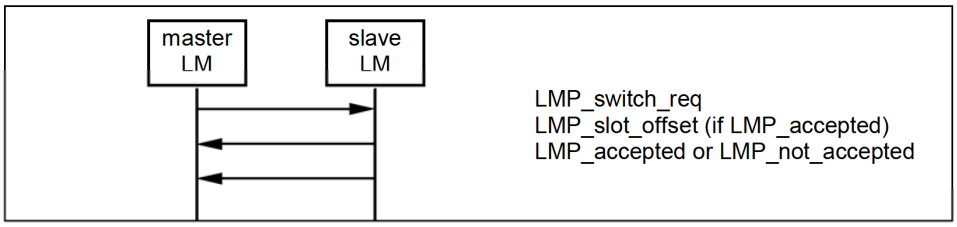
The initiating LM shall pause traffic on the ACL-U logical link (see Baseband Specification, Section 5.3.1, on page 108). It shall then send an LMP_slot_offset PDU immediately followed by an LMP_switch_req PDU. If the master accepts the role switch it shall pause traffic on the ACL-U logical link (seeBaseband Specification, Section 5.3.1, on page 108) and respond with an LMP_accepted PDU. When the role switch has been completed at the baseband level (successfully or not) both devices re-enable transmission on the ACL-U logical link. If the master rejects the role switch it responds with an LMP_not_accepted PDU and the slave re-enables transmission on the ACL-U logical link. The transaction ID for all PDUs in the sequence shall be set to 1.



Sequence 47: Role switch (slave initiated).

If the master initiates the role switch it shall pause traffic on the ACL-U logical link (see Baseband Specification, Section 5.3.1, on page 108) and send an LMP_switch_req PDU. If the slave accepts the role switch it shall pause traffic on the ACL-U logical link (see Baseband Specification, Section 5.3.1, on page 108) and responds with an LMP_slot_offset PDU immediately followed by an

LMP_accepted PDU. When the role switch has been completed at the base-band (successfully or not) both devices re-enable transmission on the ACL-U logical link. If the slave rejects the role switch it responds with an LMP_not_accepted PDU and the master re-enables transmission on the ACL-U logical link. The transaction ID for all PDUs in the sequence shall be set to 0.



Sequence 48: Role switch (master initiated).

The LMP_switch_req PDU contains a parameter, switch instant, which specifies the instant at which the TDD switch is performed. This is specified as a Bluetooth clock value of the master's clock, that is available to both devices. This instant is chosen by the sender of the message and shall be at least $2*T_{poll}$ or 32 (whichever is greater) slots in the future. The switch instant shall be within 12 hours of the current clock value to avoid clock wrap.

The sender of the LMP_switch_req PDU selects the switch instant and queues the LMP_switch_req PDU to LC for transmission and starts a timer to expire at the switch instant. When the timer expires it initiates the mode switch. In the case of a master initiated switch if the LMP_slot_offset PDU has not been received by the switch instant the role switch is carried out without an estimate of the slave's slot offset. If an LMP_not_accepted PDU is received before the timer expires then the timer is stopped and the role switch shall not be initiated.

When the LMP_switch_req is received the switch instant is compared with the current master clock value. If it is in the past then the instant has been passed and an LMP_not_accepted PDU with the error code *instant passed* shall be returned. If it is in the future then an LMP_accepted PDU shall be returned assuming the role switch is allowed and a timer is started to expire at the switch instant. When this timer expires the role switch shall be initiated.

After a successful role switch the supervision timeout and poll interval (T_{poll}) shall be set to their default values. The authentication state and the ACO shall remain unchanged. Adaptive Frequency Hopping shall follow the procedures described in Baseband Specification, Section 8.6.5, on page 175. The default value for max_slots shall be used.

4.5 MODES OF OPERATION

4.5.1 Hold mode

The ACL logical transport of a connection between two Bluetooth devices can be placed in hold mode for a specified hold time. See Baseband Specification, Section 8.8, on page 185 for details.

M/O	PDU	Contents
O(6)	LMP_hold	hold time, hold instant
O(6)	LMP_hold_req	hold time, hold instant

Table 4.25: PDUs used for hold mode.

The LMP_hold and LMP_hold_req PDUs both contain a parameter, hold instant, that specifies the instant at which the hold becomes effective. This is specified as a Bluetooth clock value of the master's clock, that is available to both devices. The hold instant is chosen by the sender of the message and should be at least $6*T_{poll}$ slots in the future. The hold instant shall be within 12 hours of the current clock value to avoid clock wrap.

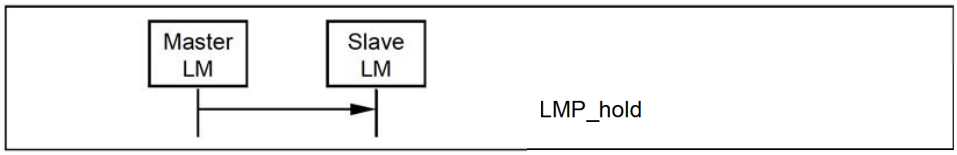
4.5.1.1 Master forces hold mode

The master may force hold mode if there has previously been a request for hold mode that has been accepted. The hold time included in the PDU when the master forces hold mode shall not be longer than any hold time the slave has previously accepted when there was a request for hold mode.

The master LM shall first pause traffic on the ACL-U logical link (see Baseband Specification, Section 5.3.1, on page 108). It shall select the hold instant and queue the LMP_hold PDU to its LC for transmission. It shall then start a timer to wait until the hold instant occurs. When this timer expires then the connection shall enter hold mode. If the baseband acknowledgement for the LMP_hold PDU is not received then the master may enter hold mode, but it shall not use its low accuracy clock during the hold.

When the slave LM receives an LMP_hold PDU it compares the hold instant with the current master clock value. If it is in the future then it starts a timer to expire at this instant and enters hold mode when it expires.

When the master LM exits from Hold mode it re-enables transmission on the ACL-U logical link.



Sequence 49: Master forces slave into hold mode.

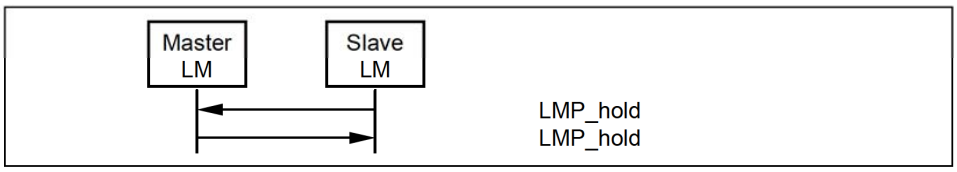
4.5.1.2 Slave forces hold mode

The slave may force hold mode if there has previously been a request for hold mode that has been accepted. The hold time included in the PDU when the slave forces hold mode shall not be longer than any hold time the master has previously accepted when there was a request for hold mode.

The slave LM shall first complete the transmission of the current packet on the ACL logical transport and then shall suspend transmission on the ACL-U logical link. It shall select the hold instant and queue the LMP_hold PDU to its LC for transmission. It shall then wait for an LMP_hold PDU from the master acting according to the procedure described in Section 4.5.1.1 .

When the master LM receives an LMP_hold PDU it shall pause traffic on the ACL-U logical link (see Baseband Specification, Section 5.3.1, on page 108). It shall then inspect the hold instant. If this is less than $6*T_{poll}$ slots in the future it shall modify the instant so that it is at least $6*T_{poll}$ slots in the future. It shall then send an LMP_hold PDU using the mechanism described in Section 4.5.1.1 .

When the master and slave LMs exit from Hold mode they shall re-enable transmission on the ACL-U logical link.



Sequence 50: Slave forces master into hold mode.

4.5.1.3 Master or slave requests hold mode

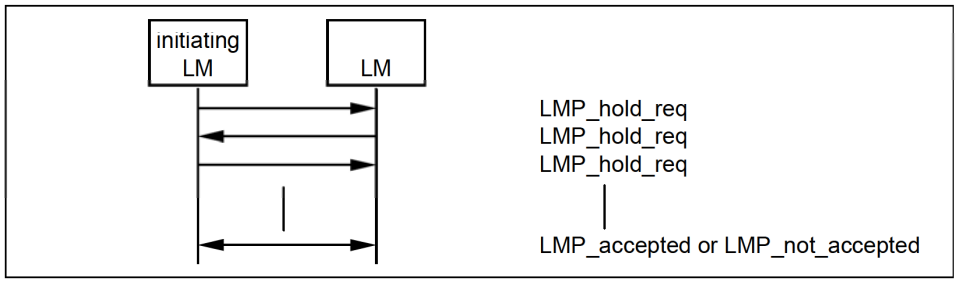
The master or the slave can request to enter hold mode. Upon receipt of the request, the same request with modified parameters can be returned or the negotiation can be terminated. If an agreement is seen an LMP_accepted PDU terminates the negotiation and the ACL link is placed in hold mode. If no agreement is seen, an LMP_not_accepted PDU with the error code *unsupported parameter value* terminates the negotiation and hold mode is not entered.

The initiating LM shall pause traffic on the ACL-U logical link (see Baseband Specification, Section 5.3.1, on page 108). On receiving an LMP_hold_req PDU the receiving LM shall complete the transmission of the current packet on the ACL logical transport and then shall suspend transmission on the ACL-U logical link.

The LM sending the LMP_hold_req PDU selects the hold instant, that shall be at least $9*T_{poll}$ slots in the future. If this is a response to a previous LMP_hold_req PDU and the contained hold instant is at least $9*T_{poll}$ slots in the

future then this shall be used. The LMP_hold_req PDU shall then be queued to its LC for transmission and a timer shall be started to expire at this instant and the connection enters hold mode when it expires unless an LMP_not_accepted or LMP_hold_req PDU is received by its LM before that point. If the LM receiving LMP_hold_req PDU agrees to enter hold mode it shall return an LMP_accepted PDU and shall start a timer to expire at the hold instant. When this timer expires it enters hold mode.

When each LM exits from Hold mode it shall re-enable transmission on the ACL-U logical link.



Sequence 51: Negotiation for hold mode.

4.5.2 Park state

If a slave does not need to participate in the channel, but should still remain synchronized to the master, it may be placed in park state. See Baseband Specification, Section 8.9, on page 185 for details.

Note: to keep a parked slave connected the master shall periodically unpark and repark the slave if the supervision timeout is not set to zero (see Baseband Specification, Section 3.1, on page 95).

All PDUs sent from the master to parked slaves are carried on the PSB-C logical link (LMP link of parked slave broadcast logical transport). These PDUs, LMP_set_broadcast_scan_window, LMP_modify_beacon, LMP_unpark_BD_addr_req and LMP_unpark_PM_addr_req, are the only PDUs that shall be sent to a slave in park state and the only PDUs that shall be broadcast. To increase reliability for broadcast, the packets are as short as possible. Therefore the format for these LMP PDUs are somewhat different. The parameters are not always byte-aligned and the length of the PDUs is variable.

The messages for controlling park state include parameters, defined in Baseband Specification, Section 8.9, on page 185. When a slave is placed in park state it is assigned a unique PM_ADDR, that can be used by the master to unpark that slave. The all-zero PM_ADDR has a special meaning; it is not a valid PM_ADDR. If a device is assigned this PM_ADDR, it shall be identified with its BD_ADDR when it is unparked by the master.

M/O	PDU	Contents
O(8)	LMP_park_req	timing control flags D_B T_B N_B Δ_B PM_ADDR AR_ADDR N_{Bsleep} D_{Bsleep} D_{access} T_{access} $N_{acc\text{-}slots}$ N_{poll} M_{access} access scheme
O(8)	LMP_set_broadcast_scan_window	timing control flags D_B (optional) broadcast scan window
O(8)	LMP_modify_beacon	timing control flags D_B (optional) T_B N_B Δ_B D_{access} T_{access} $N_{acc\text{-}slots}$ N_{poll} M_{access} access scheme

Table 4.26: PDUs used for park state.

M/O	PDU	Contents
O(8)	LMP_unpark_PM_ADDR_req	timing control flags D_B (optional) LT_ADDR 1st unpark LT_ADDR 2nd unpark PM_ADDR 1st unpark PM_ADDR 2nd unpark LT_ADDR 3rd unpark LT_ADDR 4th unpark PM_ADDR 3rd unpark PM_ADDR 4th unpark LT_ADDR 5th unpark LT_ADDR 6th unpark PM_ADDR 5th unpark PM_ADDR 6th unpark LT_ADDR 7th unpark PM_ADDR 7th unpark
O(8)	LMP_unpark_BD_ADDR_req	timing control flags D_B (optional) LT_ADDR LT_ADDR (optional) BD_ADDR BD_ADDR (optional)

Table 4.26: PDUs used for park state.

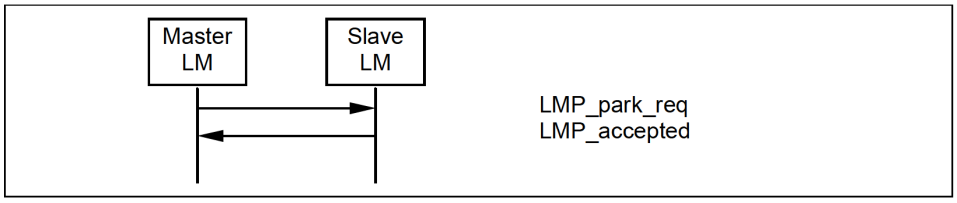
4.5.2.1 Master requests slave to enter park state

The master can request park state. The master LM shall pause traffic on the ACL-U logical link (see Baseband Specification, Section 5.3.1, on page 108) and then send an LMP_park_req PDU. If the slave agrees to enter park state it shall pause traffic on the ACL-U logical link (see Baseband Specification, Section 5.3.1, on page 108). and then respond with an LMP_accepted PDU.

When the slave queues an LMP_accepted PDU it shall start a timer for 6*Tpoll slots. If the baseband acknowledgement is received before this timer expires it shall enter park state immediately otherwise it shall enter park state when the timer expires.

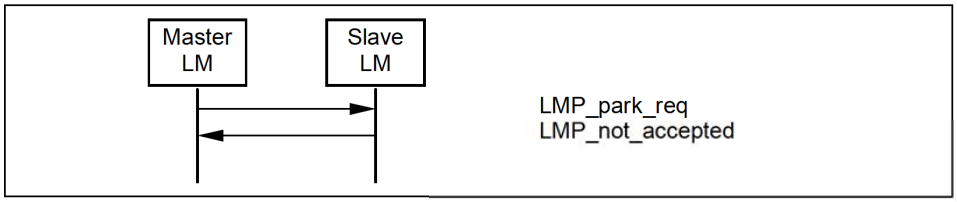
When the master receives an LMP_accepted PDU it shall start a timer for 6*Tpoll slots. When this timer expires the slave is in park state and the LT_ADDR may be re-used.

If the master never receives an LMP_accepted PDU then a link supervision timeout will occur.



Sequence 52: Slave accepts to enter park state.

If the slave rejects the attempt to enter park state it shall respond with an LMP_not_accepted PDU and the master shall re-enable transmission on the ACL-U logical link.



Sequence 53: Slave rejects to enter into park state

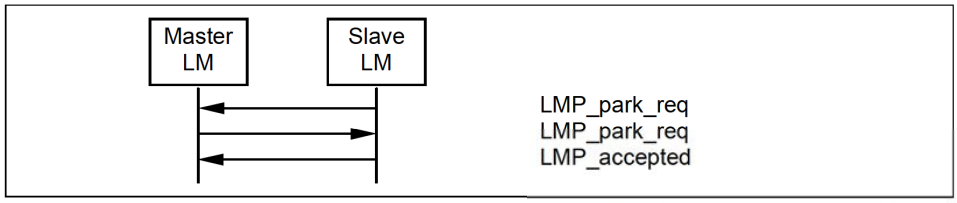
4.5.2.2 Slave requests to enter park state

The slave can request park state. The slave LM shall pause traffic on the ACL-U logical link (see Baseband Specification, Section 5.3.1, on page 108) and then send an LMP_park_req PDU. When sent by the slave, the parameters PM_ADDR and AR_ADDR are not valid and the other parameters represent suggested values. If the master accepts the slave's request to enter park state it shall pause traffic on the ACL-U logical link (see Baseband Specification, Section 5.3.1, on page 108) and then send an LMP_park_req PDU, where the parameter values may be different from the values in the PDU sent from the slave. If the slave can accept these parameter it shall respond with an LMP_accepted PDU.

When the slave queues an LMP_accepted PDU for transmission it shall start a timer for 6*Tpoll slots. If the baseband acknowledgement is received before this timer expires it shall enter park state immediately otherwise it shall enter park state when the timer expires.

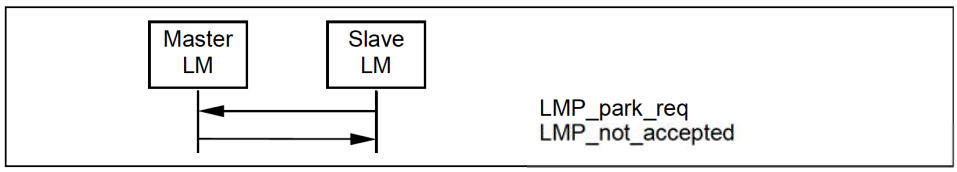
When the master receives an LMP_accepted PDU it shall start a timer for 6*Tpoll slots. When this timer expires the slave is in park state and the LT_ADDR may be re-used.

If the master never receives the LMP_accepted PDU then a link supervision timeout will occur.



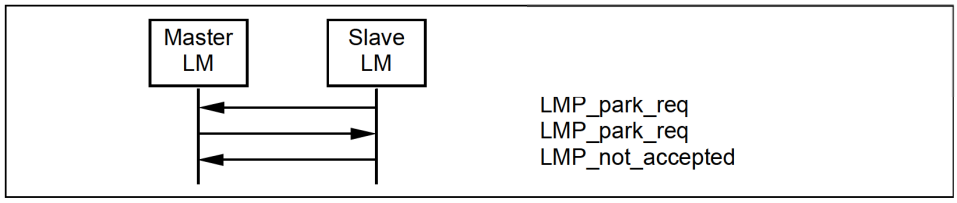
Sequence 54: Slave requests to enter park state and accepts master's beacon parameters.

If the master does not agree that the slave enters park state it shall send an LMP_not_accepted PDU. The slave shall then re-enable transmission on the ACL-U logical link.



Sequence 55: Master rejects slave's request to enter park state

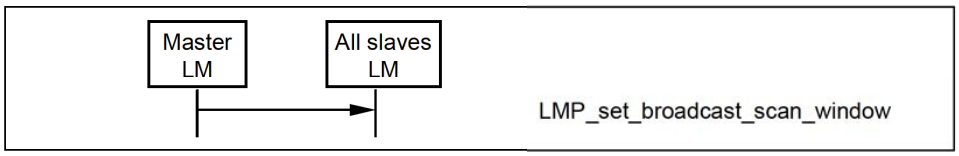
If the slave does not accept the parameters in the LMP_park_req PDU sent from the master it shall respond with an LMP_not_accepted PDU and both devices shall re-enable transmission on the ACL-U logical link.



Sequence 56: Slave requests to enter park state, but rejects master's beacon parameters.

4.5.2.3 Master sets up broadcast scan window

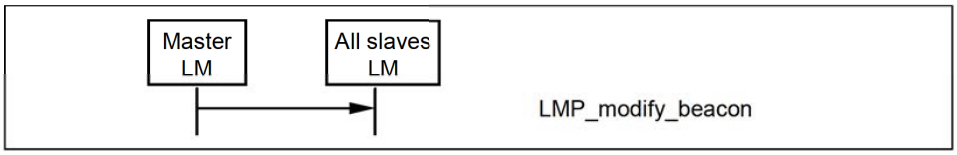
If more broadcast capacity is needed than the beacon train, the master may indicate to the slaves that more broadcast information will follow the beacon train by sending an LMP_set_broadcast_scan_window PDU. This message shall be sent in a broadcast packet at the beacon slot(s). The scan window shall start in the beacon instant and shall only be valid for the current beacon.



Sequence 57: Master notifies all slaves of increase in broadcast capacity.

4.5.2.4 Master modifies beacon parameters

When the beacon parameters change the master notifies the parked slaves of this by sending an LMP_modify_beacon PDU. This PDU shall be sent in a broadcast packet.



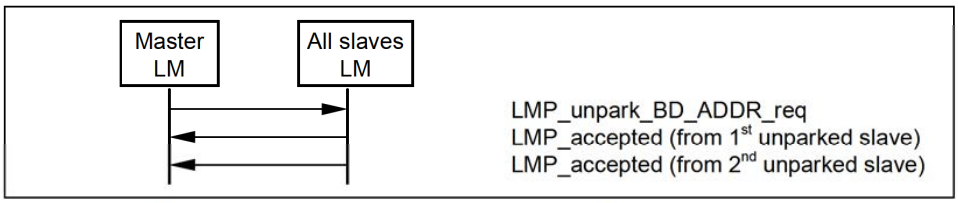
Sequence 58: Master modifies beacon parameters.

4.5.2.5 Unparking slaves

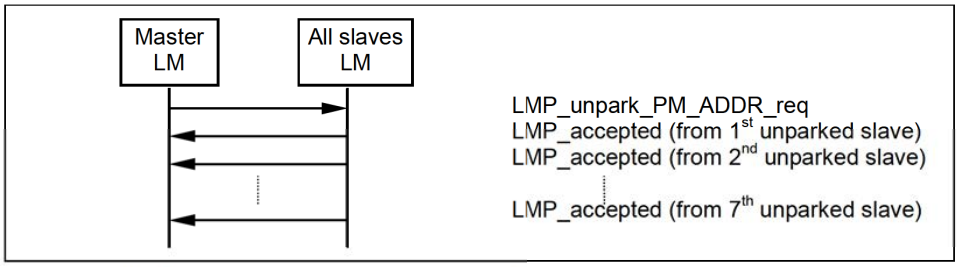
The master can unpark one or many slaves by sending a broadcast LMP message including the PM_ADDR or the BD_ADDR of the device(s) to be unparked. Broadcast LMP messages are carried on the PSB-C logical link. See Baseband Specification, Section 8.9.5, on page 192 for further details. This message also includes the LT_ADDR that the master assigns to the slave(s). After sending this message, the master shall check the success of the unpark by polling each unparked slave by sending POLL packets, so that the slave is granted access to the channel. The unparked slave shall then send a response with an LMP_accepted PDU. If this message is not received from the slave within a certain time after the master sent the unpark message, the unpark failed and the master shall consider the slave as still being in park state.

One PDU is used where the parked device is identified with the PM_ADDR, and another PDU is used where it is identified with the BD_ADDR. Both messages have variable length depending on the number of slaves the master unparks. For each slave the master wishes to unpark an LT_ADDR followed by the PM_ADDR or BD_ADDR of the device that is assigned this LT_ADDR is included in the payload. If the slaves are identified with the PM_ADDR a maximum of 7 slaves can be unparked with the same message. If they are identified with the BD_ADDR a maximum of 2 slaves can be unparked with the same message.

After a successful unparking, both devices re-enable transmission on the ACL-U logical link.



Sequence 59: Master unparks slaves addressed with their BD_ADDR.



Sequence 60: Master unparks slaves addressed with their PM_ADDR.

4.5.3 Sniff mode

To enter sniff mode, master and slave negotiate a sniff interval T_{sniff} and a sniff offset, D_{sniff}, that specifies the timing of the sniff slots. The offset determines the time of the first sniff slot; after that the sniff slots follow periodically with the sniff interval T_{sniff}. To avoid clock wrap-around during the initialization, one of two options is chosen for the calculation of the first sniff slot. A timing control flag in the message from the master indicates this.Only bit1 of the timing control flag is valid.

When the ACL logical transport is in sniff mode the master shall only start a transmission in the sniff slots. Two parameters control the listening activity in the slave: the sniff attempt and the sniff timeout. The sniff attempt parameter determines for how many slots the slave shall listen when the slave is not treating this as a scatternet link, beginning at the sniff slot, even if it does not receive a packet with its own LT_ADDR. The sniff timeout parameter determines for how many additional slots the slave shall listen when the slave is not treating this as a scatternet link if it continues to receive only packets with its own LT_ADDR. It is not possible to modify the sniff parameters while the device is in sniff mode.

M/O	PDU	Contents
O(7)	LMP_sniff_req	timing control flags D_{sniff} T_{sniff} sniff attempt sniff timeout
O(7)	LMP_unsniff_req	-

Table 4.27: PDUs used for sniff mode.

4.5.3.1 Master or slave requests sniff mode

Either the master or the slave may request entry to sniff mode.

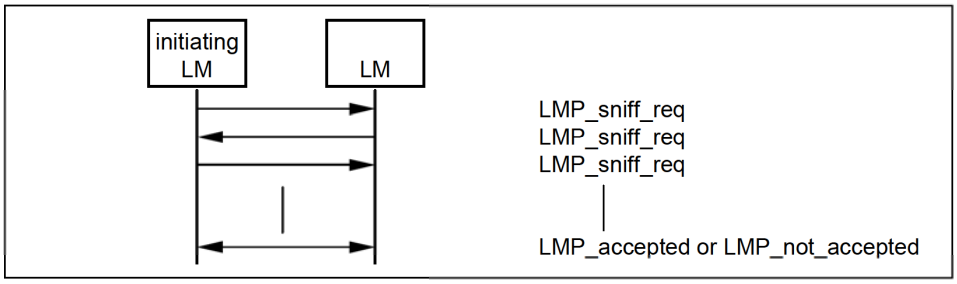
The process is initiated by sending an LMP_sniff_req PDU containing a set of parameters. The receiving LM shall then decide whether to reject the attempt by sending an LMP_not_accepted PDU, to suggest different parameters by replying with an LMP_sniff_req PDU or to accept the request.

Before the first time that the master sends LMP_sniff_req it shall enter sniff transition mode. If the master receives or sends an LMP_not_accepted PDU it shall exit from sniff transition mode. If the master receives an LMP_sniff_req PDU it shall enter sniff transition mode.

If the master decides to accept the request it shall send an LMP_accepted PDU. When the master receives the baseband acknowledgement for this PDU it shall exit sniff transition mode and enter sniff mode.

If the master receives an LMP_accepted PDU the master shall exit from sniff transition mode and enter sniff mode.

If the slave receives an LMP_sniff_req PDU it must decide whether to accept the request. If the slave does not wish to enter sniff mode then it replies with an LMP_not_accepted PDU. If it is happy to enter sniff mode but requires a different set of parameters it shall respond with an LMP_sniff_req PDU containing the new parameters. If the slave decides that the parameters are acceptable then it shall send an LMP_accepted PDU and enter sniff mode. If the slave receives an LMP_not_accepted PDU it shall terminate the attempt to enter sniff mode.



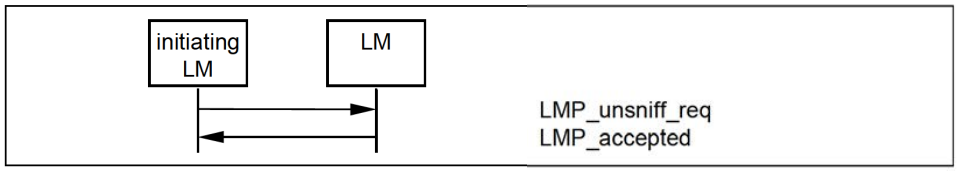
Sequence 61: Negotiation for sniff mode.

4.5.3.2 Moving a slave from sniff mode to active mode

Sniff mode may be exited by either the master or the slave sending an LMP_unsniff_req PDU. The requested device must reply with an LMP_accepted PDU.

If the master requests an exit from sniff mode it shall enter sniff transition mode and then send an LMP_unsniff_req PDU. When the slave receives the LMP_unsniff_req it shall exit from sniff mode and reply with an LMP_accepted PDU. When the master receives the LMP_accepted PDU it shall exit from sniff transition mode and enter active mode.

If the slave requests an exit from sniff mode it shall send an LMP_unsniff_req PDU. When the master receives the LMP_unsniff_req PDU it shall enter sniff transition mode and then send an LMP_accepted PDU. When the slave receives the LMP_accepted PDU it shall exit from sniff mode and enter active mode. When the master receives the baseband acknowledgement for the LMP_accepted PDU it shall leave sniff transition mode and enter active mode.



Sequence 62: Slave moved from sniff mode to active mode.

4.6 LOGICAL TRANSPORTS

When a connection is first established between two devices the connection consists of the default ACL logical links: ACL-C (for LMP messages) and ACL-U (for L2CAP data.) One or more synchronous logical transports (SCO or eSCO) may then be added. A new logical transport shall not be created if it would cause all slots to be allocated to reserved slots on secondary LT_ADDRs.

4.6.1 SCO logical transport

The SCO logical transport reserves slots separated by the SCO interval, T_{sco}. The first slot reserved for the SCO logical transport is defined by T_{sco} and the SCO offset, D_{sco}. See Baseband Specification, Section 8.6.2, on page 169 for details. A device shall initiate a request for HV2 or HV3 packet type only if the other device supports it (bits 12, 13) in its features mask. A device shall initiate CVSD, μ-law or A-law coding or uncoded (transparent) data only if the other device supports the corresponding feature. To avoid problems with a wrap-around of the clock during initialization of the SCO logical transport, the timing control flags parameter is used to indicate how the first SCO slot shall be calculated. Only bit1 of the timing control flags parameter is valid. The SCO link is distinguished from all other SCO links by an SCO handle. The SCO handle zero shall not be used.

M/O	PDU	Contents
O(11)	LMP_SCO_link_req	SCO handle timing control flags D_{sco} T_{sco} SCO packet air mode
O(11)	LMP_remove_SCO_link_req	SCO handle error

Table 4.28: PDUs used for managing the SCO links.

4.6.1.1 Master initiates an SCO link

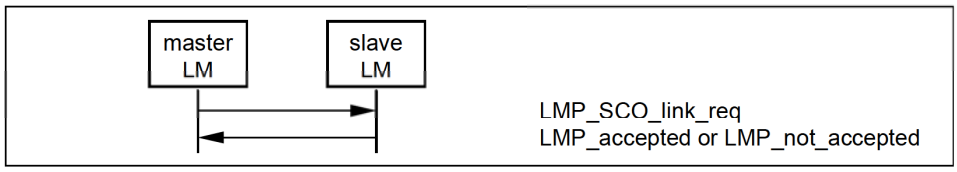
When establishing an SCO link the master sends a request, a LMP_SCO_link_req PDU, with parameters that specify the timing, packet type and coding that will be used on the SCO link. Each of the SCO packet types supports three different voice coding formats on the air-interface: μ-law log PCM, A-law log PCM and CVSD. The air coding by log PCM or CVSD may be deactivated to achieve a transparent synchronous data link at 64 kbits/s.

The slots used for the SCO links are determined by three parameters controlled by the master: T_{sco}, D_{sco} and a flag indicating how the first SCO slot is calculated. After the first slot, the SCO slots follow periodically at an interval of T_{sco}.

If the slave does not accept the SCO link, but is willing to consider another possible set of SCO parameters, it can indicate what it does not accept in the error code field of LMP_not_accepted PDU. The master may then issue a new request with modified parameters.

The SCO handle in the message shall be different from existing SCO link(s).

If the SCO packet type is HV1 the LMP_accepted shall be sent using the DM1 packet.

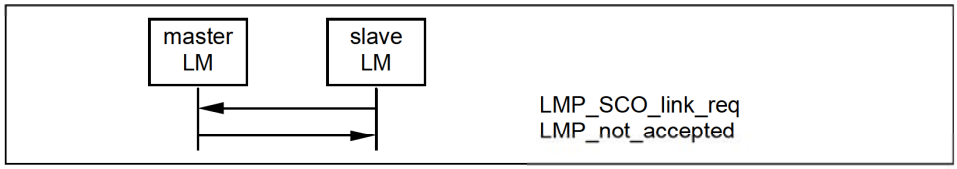


Sequence 63: Master requests an SCO link.

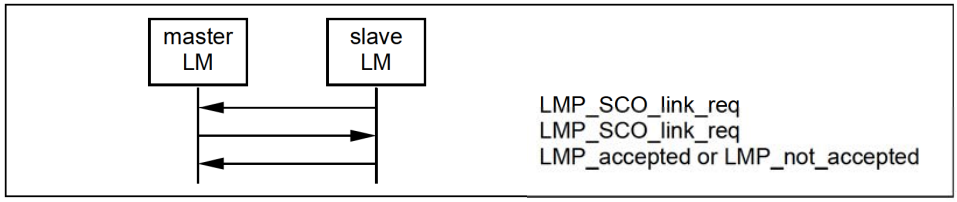
4.6.1.2 Slave initiates an SCO link

The slave may initiate the establishment of an SCO link. The slave sends an LMP_SCO_link_req PDU, but the parameters timing control flags and D_{sco} are invalid as well as the SCO handle, that shall be zero. If the master is not capable of establishing an SCO link, it replies with an LMP_not_accepted PDU. Otherwise it sends back an LMP_SCO_link_req PDU. This message includes the assigned SCO handle, D_{sco} and the timing control flags. The master should try to use the same parameters as in the slave request; if the master cannot meet that request, it is allowed to use other values. The slave shall then reply with LMP_accepted or LMP_not_accepted PDU.

If the SCO packet type is HV1 the LMP_accepted shall be sent using the DM1 packet.



Sequence 64: Master rejects slave's request for an SCO link.



Sequence 65: Master accepts slave's request for an SCO link.

4.6.1.3 Master requests change of SCO parameters

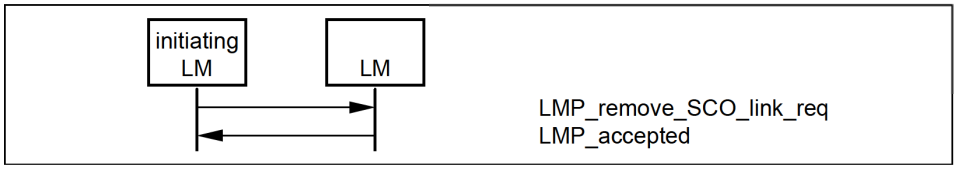
The master sends an LMP_SCO_link_req PDU, where the SCO handle is the handle of the SCO link the master wishes to change parameters for. If the slave accepts the new parameters, it replies with an LMP_accepted PDU and the SCO link will change to the new parameters. If the slave does not accept the new parameters, it shall reply with an LMP_not_accepted PDU and the SCO link is left unchanged. When the slave replies with an LMP_not_accepted PDU it shall indicate in the error code parameter what it does not accept. The master may then try to change the SCO link again with modified parameters. The sequence is the same as in Section 4.6.1.1 on page 281.

4.6.1.4 Slave requests change of SCO parameters

The slave sends an LMP_SCO_link_req PDU, where the SCO handle is the handle of the SCO link to be changed. The parameters timing control flags and D_{sco} are not valid in this PDU. If the master does not accept the new parameters it shall reply with an LMP_not_accpeted PDU and the SCO link is left unchanged. If the master accepts the new parameters it shall reply with an LMP_SCO_link_req PDU containing the same parameters as in the slave request. When receiving this message the slave replies with an LMP_not_accepted PDU if it does not accept the new parameters. The SCO link is then left unchanged. If the slave accepts the new parameters it replies with an LMP_accepted PDU and the SCO link will change to the new parameters. The sequence is the same as in Section 4.6.1.2 on page 282.

4.6.1.5 Remove an SCO link

Master or slave can remove the SCO link by sending a request including the SCO handle of the SCO link to be removed and an error code indicating why the SCO link is removed. The receiving side shall respond with an LMP_accepted PDU.



Sequence 66: SCO link removed.

4.6.2 eSCO logical transport

After an ACL link has been established, one or more extended SCO (eSCO) links can be set up to the remote device. The eSCO links are similar to SCO links using timing control flags, an interval T_{eSCO} and an offset D_{eSCO}. Only bit1 of the timing control flags parameter is valid. As opposed to SCO links, eSCO links have a configurable data rate that may be asymmetric, and can be set up to provide limited retransmissions of lost or damaged packets inside a retransmission window of size W_{eSCO}. The D_{eSCO} shall be based on CLK.

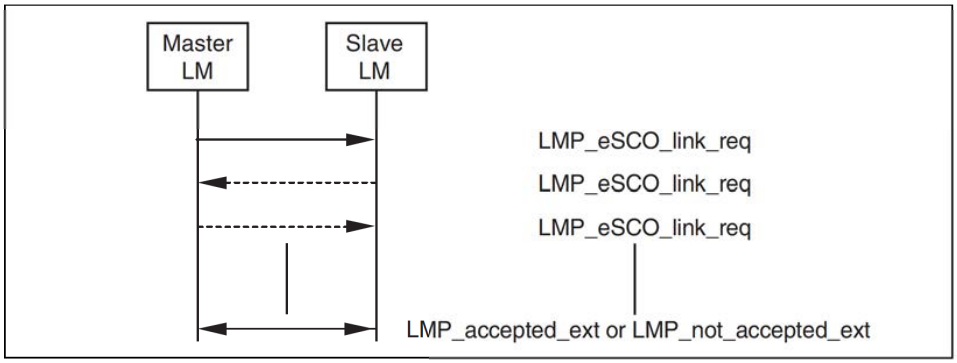
M/O	PDU	Contents
O(31)	LMP_eSCO_link_req	eSCO handle eSCO LT_ADDR timing control flags D_{eSCO} T_{eSCO} W_{eSCO} eSCO packet type M->S eSCO packet type S->M packet length M->S packet length S->M air mode negotiation state
O(31)	LMP_remove_eSCO_link_req	eSCO handle error

Table 4.29: PDUs used for managing the eSCO links

The parameters D_{eSCO}, T_{eSCO}, W_{eSCO}, eSCO packet type M->S, eSCO packet type S->M, packet length M->S, packet length S->M are henceforth referred to as the negotiable parameters.

4.6.2.1 Master initiates an eSCO link

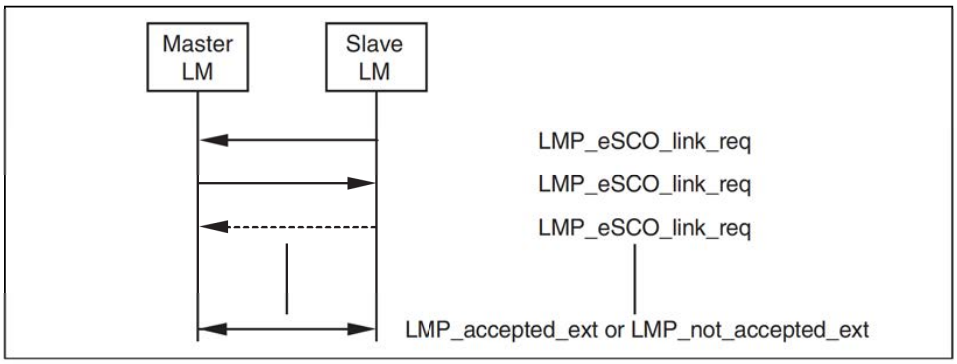
When establishing an eSCO link the master sends an LMP_eSCO_link_req PDU specifying all parameters. The slave may accept this with an LMP_accepted_ext PDU, reject it with an LMP_not_accepted_ext PDU, or respond with its own LMP_eSCO_link_req specifying alternatives for some or all parameters. The slave shall not negotiate the eSCO handle or eSCO LT_ADDR parameters. The negotiation of parameters continues until the master or slave either accepts the latest parameters with an LMP_accepted_ext PDU, or terminates the negotiation with an LMP_not_accepted_ext PDU. The negotiation shall use the procedures defined in Section 4.6.2.5 on page 287.



Sequence 67: Master requests an eSCO link.

4.6.2.2 Slave initiates an eSCO link

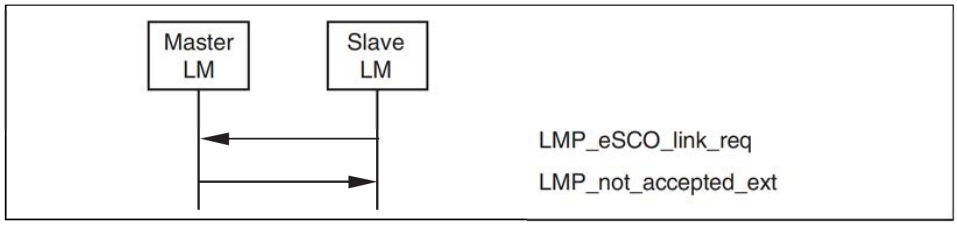
When attempting to establish an eSCO link the slave shall send an LMP_eSCO_link_req PDU specifying all parameters, with the exception of eSCO LT_ADDR and eSCO handle, which are invalid. The latter shall be set to zero. The master may respond to this with an LMP_eSCO_link_req PDU, filling in these missing parameters, and potentially changing the other requested parameters. The slave can accept this with an LMP_accepted_ext PDU, or respond with a further LMP_eSCO_link_req PDU specifying alternatives for some or all of the parameters. The negotiation of parameters continues until the master or slave either accepts the latest parameters with an LMP_accepted_ext PDU, or terminates the negotiation with an LMP_not_accepted_ext PDU.



Sequence 68: Slave requests an eSCO link.

Note that the slave should use the initialization flag appropriate to the master's Bluetooth clock. See Baseband section section 8.6.3.

The master may reject the request immediately with an LMP_not_accepted_ext PDU. The negotiation shall use the procedures defined in Section 4.6.2.5 on page 287.



Sequence 69: Master rejects slave's request for an eSCO link.

4.6.2.3 Master or slave requests change of eSCO parameters

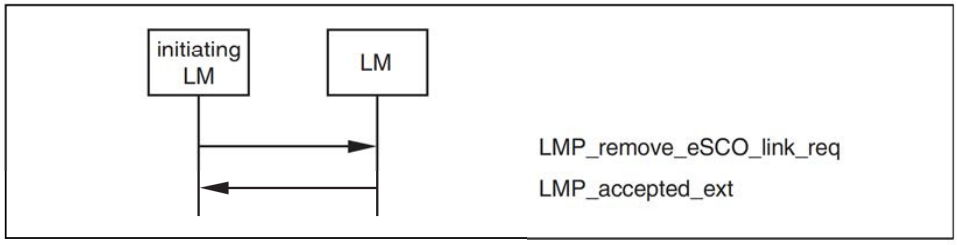
The master or slave may request a renegotiation of the eSCO parameters. The master or slave shall send an LMP_eSCO_link_req PDU with the eSCO handle of the eSCO link the device wishes to renegotiate. The remote device may accept the changed parameters immediately with LMP_accepted_ext PDU, or the negotiation may be continued with further LMP_eSCO_link_req PDUs until the master or slave accepts the latest parameters with an LMP_accepted_ext PDU or terminates the negotiation with an LMP_not_accepted_ext PDU. In the case of termination with an LMP_not_accepted_ext PDU, the eSCO link continues on the previously negotiated parameters.

The sequence is the same as in Section 4.6.2.2 on page 285.

During re-negotiation, the eSCO LT_ADDR and eSCO handle shall not be re-negotiated and shall be set to the originally negotiated values. The negotiation shall use the procedures defined in Section 4.6.2.5 on page 287.

4.6.2.4 Remove an eSCO link

Either the master or slave may remove the eSCO link by sending a request including the eSCO handle of the eSCO link to be removed and a error code indicating why the eSCO link is removed. The receiving side shall respond with an LMP_accepted_ext PDU.



Sequence 70: eSCO link removed

4.6.2.5 Rules for the LMP negotiation and renegotiation

Rule 1: the negotiation_state shall be set to 0 by the initiating LM. After the initial LMP_eSCO_link_req is sent the negotiation_state shall not be set to 0.

Rule 2: if the bandwidth (defined as 1600 times the packet length in bytes divided by T_{eSCO} in slots) for either RX or TX or the air_mode cannot be accepted the device shall send LMP_not_accepted_ext with the appropriate error code.

Rule 3: Bandwidth and air_mode are not negotiable and shall not be changed for the duration of the negotiation. Once one side has rejected the negotiation (with LMP_not_accepted_ext) a new negotiation may be started with different bandwidth and air_mode parameters.

Rule 4: if the parameters will cause a latency violation (T_{eSCO} + W_{eSCO} + reserved synchronous slots > allowed local latency) the device should propose new parameters that shall not cause a reserved slot violation or latency violation for the device that is sending the parameters. In this case the negotiation_state shall be set to 3. Otherwise the device shall send LMP_not_accepted_ext.

Rule 5: once a device has received an LMP_eSCO_link_req with the negotiation_state set to 3 (latency violation), the device shall not propose any combination of packet type, T_{eSCO}, and W_{eSCO} that will give an equal or larger latency than the combination that caused the latency violation for the other device.

Rule 6: if the parameters cause both a reserved slot violation and a latency violation the device shall set the negotiation_state to 3 (latency violation).

Rule 7: if the parameters will cause a reserved slot violation the device should propose new parameters that shall not cause a reserved slot violation. In this case the negotiation_state shall be set to 2. Otherwise the device shall send LMP_not_accepted_ext.

Rule 8: If the requested parameters are not supported the device should propose a setting that is supported, and set the negotiation_state to 4. If it is not possible to find such a parameter set, the device shall send LMP_not_accepted_ext.

Rule 9: when proposing new parameters for reasons other than a latency violation, reserved slot violation, or configuration not supported, the negotiation_state shall be set to 1.

4.6.2.6 Negotiation state definitions

Reserved Slot Violation: a reserved slot violation is when the receiving LM cannot setup the requested eSCO logical transport because the eSCO reserved slots would overlap with other regularly scheduled slots (e.g. other synchronous reserved slots, sniff instants, or park beacons).

Latency Violation: a latency violation is when the receiving LM cannot setup the requested eSCO logical transport because the latency (W_{eSCO}+T_{eSCO} + reserved synchronous slots) is greater than the maximum allowed latency.

Configuration not supported: The combination of parameters requested is not inside the supported range for the device.

4.7 TEST MODE

LMP has PDUs to support different test modes used for certification and compliance testing of the Bluetooth radio and baseband. See "Test Methodology" on page 231[vol. 4] for a detailed description of these test modes.

4.7.1 Activation and deactivation of test mode

The activation may be carried out locally (via a HW or SW interface), or using the air interface.

- For activation over the air interface, entering the test mode shall be locally enabled for security and type approval reasons. The implementation of this local enabling is not subject to standardization.

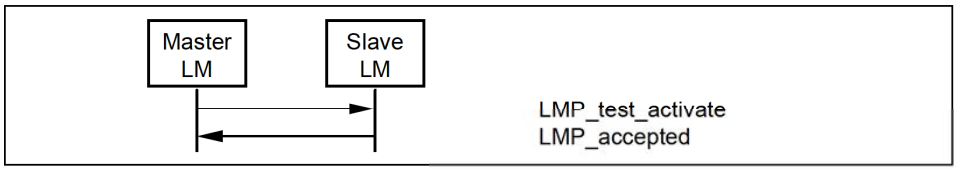
  The tester sends an LMP command that shall force the DUT to enter test mode. The DUT shall terminate all normal operation before entering the test mode.

  The DUT shall return an LMP_Accepted on reception of an activation command. LMP_Not_Accepted shall be returned if the DUT is not locally enabled.

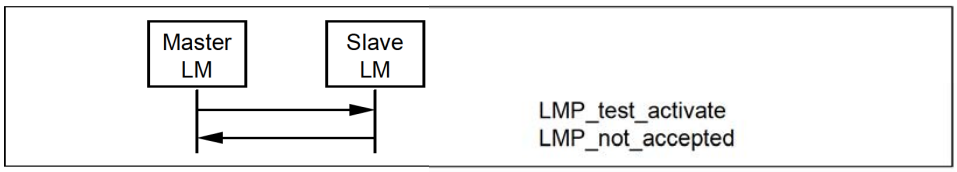
- If the activation is performed locally using a HW or SW interface, the DUT shall terminate all normal operation before entering the test mode.

  Until a connection to the tester exists, the device shall perform page scan and inquiry scan. Extended scan activity is recommended.

The test mode is activated by sending an LMP_test_activate PDU to the device under test (DUT). The DUT is always the slave. The lm shall be able to receive this message anytime. If entering test mode is locally enabled in the DUT it shall respond with an LMP_accepted PDU and test mode is entered. Otherwise the DUT responds with an LMP_not_accepted PDU and the DUT remains in normal operation. The error code in the LMP_not_accepted PDU shall be *PDU not allowed*.



Sequence 71: Activation of test mode successful.



Sequence 72: Activation of test mode fails. Slave is not allowed to enter test mode.

The test mode can be deactivated in two ways. Sending an LMP_test_control PDU with the test scenario set to "exit test mode" exits the test mode and the slave returns to normal operation still connected to the master. Sending an LMP_detach PDU to the DUT ends the test mode and the connection.

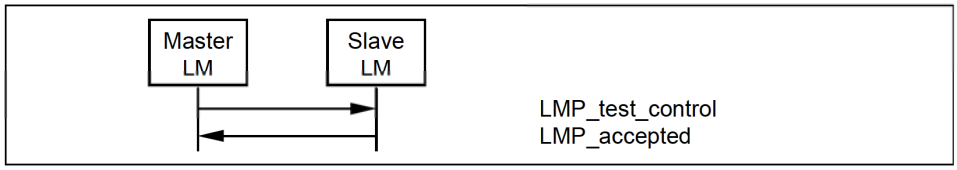
4.7.2 Control of test mode

Control and configuration is performed using special LMP commands (see Section 4.7.3 on page 291). These commands shall be rejected if the Bluetooth device is not in test mode. In this case, an LMP_not_accepted shall be returned. The DUT shall return an LMP_accepted on reception of a control command when in test mode.

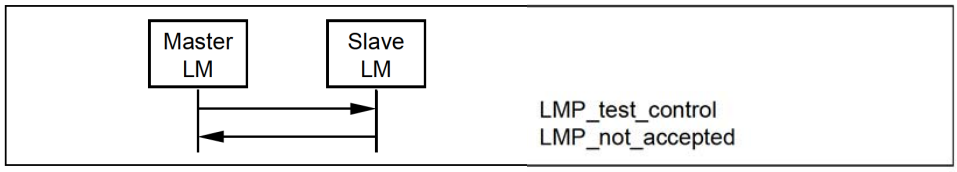
A Bluetooth device in test mode shall ignore all LMP commands not related to control of the test mode. LMP commands dealing with power control and the request for LMP features (LMP_features_req), and adaptive frequency hopping (LMP_set_AFH, LMP_channel_classification_req and LMP_channel_classification) are allowed in test mode; the normal procedures are also used to test the adaptive power control.

The DUT shall leave the test mode when an LMP_Detach command is received or an LMP_test_control command is received with test scenario set to 'exit test mode'.

When the DUT has entered test mode, the PDU LMP_test_control PDU can be sent to the DUT to start a specific test. This PDU is acknowledged with an LMP_accepted PDU. If a device that is not in test mode receives an LMP_test_control PDU it responds with an LMP_not_accepted PDU, where the error code shall be *PDU not allowed*.



Sequence 73: Control of test mode successful.



Sequence 74: Control of test mode rejected since slave is not in test mode.

4.7.3 Summary of test mode PDUs

Table 4.30 lists all LMP messages used for test mode. To ensure that the contents of LMP_test_control PDU are suitably whitened (important when sent in transmitter mode), all parameters listed in Table 4.31 on page 291 are XORed with 0x55 before being sent.

LMP PDU	PDU number	Possible Direction	Contents	Position in Payload
LMP_test_activate	56	m → s		
LMP_test_control	57	m → s	test scenario hopping mode TX frequency RX frequency power control mode poll period packet type length of test data	2 3 4 5 6 7 8 9-10
LMP_detach	7	m → s		
LMP_accepted	3	m ← s		
LMP_not_accepted	4	m ← s		

Table 4.30: LMP messages used for Test Mode

Name	Length (bytes)	Type	Unit	Detailed
Test scenario	1	u_int8		0 Pause Test Mode 1 Transmitter test – 0 pattern 2 Transmitter test – 1 pattern 3 Transmitter test – 1010 pattern 4 Pseudorandom bit sequence 5 Closed Loop Back – ACL packets 6 Closed Loop Back – Synchronous packets 7 ACL Packets without whitening 8 Synchronous Packets without whitening 9 Transmitter test – 1111 0000 pattern 10–254 reserved 255 Exit Test Mode The value is XORed with 0x55.

Table 4.31: Parameters used in LMP_Test_Control PDU

Name	Length (bytes)	Type	Unit	Detailed
Hopping mode	1	u_int8		0 RX/TX on single frequency 1 Normal hopping 2 Reserved 3 Reserved 4 Reserved 5–255 reserved The value is XORed with 0x55.
TX frequency (for DUT)	1	u_int8		f = [2402 + k] MHz The value is XORed with 0x55.
RX frequency (for DUT)	1	u_int8		f = [2402 + k] MHz The value is XORed with 0x55.
Power control mode	1	u_int8		0 fixed TX output power 1 adaptive power control The value is XORed with 0x55.
Poll period	1	u_int8	1.25 ms	The value is XORed with 0x55.
Packet type	1	u_int8		Bits 3-0 numbering as in packet header, see Baseband Specification Bits 7-4 0: ACL/SCO 1: eSCO 2: Enhanced Data Rate ACL 3: Enhanced Data Rate eSCO 4-15: reserved Other values are reserved The value is XORed with 0x55.
length of test sequence (=length of user data in Baseband Specification)	2	u_int16	1 byte	unsigned binary number The value is XORed with 0x5555.

Table 4.31: Parameters used in LMP_Test_Control PDU

The control PDU is used for both transmitter and loop back tests. The following restrictions apply for the parameter settings:

Parameter	Restrictions Transmitter Test	Restrictions Loopback Test
TX frequency	$0 \leq k \leq 93$	$0 \leq k \leq 78$
RX frequency	same as TX frequency	$0 \leq k \leq 78$
Poll period		not applicable (set to 0)
Length of test sequence	depends on packet type: [see table 6.9 and 6.10 in the Baseband specification]	For ACL and SCO packets: not applicable (set to 0) For eSCO packets: [see table 6.10 in the Baseband specification]

Table 4.32: Restrictions for Parameters used in LMP_Test_Control PDU

5 SUMMARY

5.1 PDU SUMMARY

LMP PDU	Length (bytes)	op code	Packet type	Possible direction	Contents	Position in payload
Escape 1	variable	124	DM1	m ↔ s	extended op code	2
					variable	3-?
Escape 2	variable	125	DM1	m ↔ s	extended op code	2
					variable	3-?
Escape 3	variable	126	DM1	m ↔ s	extended op code	2
					variable	3-?
Escape 4	variable	127	DM1	m ↔ s	extended op code	2
					variable	3-?
LMP_accepted	2	3	DM1/ DV	m ↔ s	op code	2
LMP_accepted_ext	4	127/ 01	DM1	m ↔ s	escape op code	3
					extended op code	4
LMP_au_rand	17	11	DM1	m ↔ s	random number	2-17
LMP_auto_rate	1	35	DM1/ DV	m ↔ s	-	
LMP_channel _classification_req	7	127/ 16	DM1	m → s	AFH_reporting_mode	3
					AFH_min_interval	4-5
					AFH_max_interval	6-7
LMP_channel _classification	12	127/ 17	DM1	m ← s	AFH_channel_ classification	3 – 12
LMP_clkoffset_req	1	5	DM1/ DV	m → s	-	
LMP_clkoffset_res	3	6	DM1/ DV	m ← s	clock offset	2–3
LMP_comb_key	17	9	DM1	m ↔ s	random number	2-17
LMP_decr_power_req	2	32	DM1/ DV	m ↔ s	for future use	2
LMP_detach	2	7	DM1/ DV	m ↔ s	error code	2

Table 5.1: Coding of the different LM PDUs.

LMP PDU	Length (bytes)	op code	Packet type	Possible direction	Contents	Position in payload
LMP_encryption_key_size_mask_req	1	58	DM1	m → s		
LMP_encryption_key_size_mask_res	3	59	DM1	m ← s	key size mask	2-3
LMP_encryption_key_size_req	2	16	DM1/ DV	m ↔ s	key size	2
LMP_encryption_mode_req	2	15	DM1/ DV	m ↔ s	encryption mode	2
LMP_eSCO_link_req	16	127/ 12	DM1	m↔s	eSCO handle	3
					eSCO LT_ADDR	4
					timing control flags	5
					D_{eSCO}	6
					T_{eSCO}	7
					W_{eSCO}	8
					eSCO packet type M->S	9
					eSCO packet type S->M	10
					packet length M->S	11-12
					packet length S->M	13-14
					air mode	15
					negotiation state	16
LMP_features_req	9	39	DM1/ DV	m ↔ s	features	2-9
LMP_features_req_ext	12	127/ 03	DM1	m ↔ s	features page	3
					max supported page	4
					extended features	5-12
LMP_features_res	9	40	DM1/ DV	m ↔ s	features	2-9
LMP_features_res_ext	12	127/ 04	DM1	m ↔ s	features page	3
					max supported page	4
					extended features	5-12
LMP_host_connection_req	1	51	DM1/ DV	m ↔ s	-	

Table 5.1: Coding of the different LM PDUs.

LMP PDU	Length (bytes)	op code	Packet type	Possible direction	Contents	Position in payload
LMP_hold	7	20	DM1/ DV	m ↔ s	hold time	2-3
					hold instant	4-7
LMP_hold_req	7	21	DM1/ DV	m ↔ s	hold time	2-3
					hold instant	4-7
LMP_incr_power_req	2	31	DM1/ DV	m ↔ s	for future use	2
LMP_in_rand	17	8	DM1	m ↔ s	random number	2-17
LMP_max_power	1	33	DM1/ DV	m ↔ s	-	
LMP_max_slot	2	45	DM1/ DV	m ↔ s	max slots	2
LMP_max_slot_req	2	46	DM1/ DV	m ↔ s	max slots	2
LMP_min_power	1	34	DM1/ DV	m ↔ s	-	
LMP_modify_beacon	11 or 13	28	DM1	m → s	timing control flags	2
					D_B	3-4
					T_B	5-6
					N_B	7
					Δ_B	8
					D_{access}	9
					T_{access}	10
					$N_{acc\text{-}slots}$	11
					N_{poll}	12
					M_{access}	13:0-3
					access scheme	13:4-7
LMP_name_req	2	1	DM1/ DV	m ↔ s	name offset	2
LMP_name_res	17	2	DM1	m ↔ s	name offset	2
					name length	3
					name fragment	4-17

Table 5.1: Coding of the different LM PDUs.

LMP PDU	Length (bytes)	op code	Packet type	Possible direction	Contents	Position in payload
LMP_not_accepted	3	4	DM1/ DV	m ↔ s	op code	2
					error code	3
LMP_not_accepted_ext	5	127/ 02	DM1	m ↔ s	escape op code	3
					extended op code	4
					error code	5
LMP_packet_type_table_ req	3	127/ 11	DM1	m ↔ s	packet type table	3
LMP_page_mode_req	3	53	DM1/ DV	m ↔ s	paging scheme	2
					paging scheme settings	3
LMP_page_scan_mode _req	3	54	DM1/ DV	m ↔ s	paging scheme	2
					paging scheme settings	3
LMP_park_req	17	25	DM1	m ↔ s	timing control flags	2
					D_B	3-4
					T_B	5-6
					N_B	7
					Δ_B	8
					PM_ADDR	9
					AR_ADDR	10
					N_{Bsleep}	11
					D_{Bsleep}	12
					D_{access}	13
					T_{access}	14
					$N_{acc-slots}$	15
					N_{poll}	16
					M_{access}	17:0-3
					access scheme	17:4-7
LMP_preferred_rate	2	36	DM1/ DV	m ↔ s	data rate	2
LMP_quality_of_service	4	41	DM1/ DV	m → s	poll interval	2-3
					N_{BC}	4

Table 5.1: Coding of the different LM PDUs.

LMP PDU	Length (bytes)	op code	Packet type	Possible direction	Contents	Position in payload
LMP_quality_of_service_req	4	42	DM1/ DV	m ↔ s	poll interval	2-3
					N_{BC}	4
LMP_remove_eSCO_link_req [] see Note4 on page 302	4	127/ 13	DM1	m↔s	eSCO handle	3
					error code	4
LMP_remove_SCO_link_req	3	44	DM1/ DV	m ↔ s	SCO handle	2
					error code	3
LMP_SCO_link_req	7	43	DM1/ DV	m ↔ s	SCO handle	2
					timing control flags	3
					D_{sco}	4
					T_{sco}	5
					SCO packet	6
					air mode	7
LMP_set_AFH	16	60	DM1	m → s	AFH_instant	2-5
					AFH_mode	6
					AFH_channel_map	7-16
LMP_set_broadcast_scan_window	4 or 6	27	DM1	m → s	timing control flags	2
					D_B	3-4
					broadcast scan window	5-6
LMP_setup_complete	1	49	DM1	m ↔ s	-	
LMP_slot_offset	9	52	DM1/ DV	m ↔ s	slot offset	2-3
					BD_ADDR	4-9
LMP_sniff_req	10	23	DM1	m ↔ s	timing control flags	2
					D_{sniff}	3-4
					T_{sniff}	5-6
					sniff attempt	7-8
					sniff timeout	9-10
LMP_sres	5	12	DM1/ DV	m ↔ s	authentication response	2-5
LMP_start_encryption_req	17	17	DM1	m → s	random number	2-17
LMP_stop_encryption_req	1	18	DM1/ DV	m → s	-	

Table 5.1: Coding of the different LM PDUs.

LMP PDU	Length (bytes)	op code	Packet type	Possible direction	Contents	Position in payload
LMP_supervision_timeout	3	55	DM1/ DV	m →s	supervision timeout	2-3
LMP_switch_req	5	19	DM1/ DV	m ↔ s	switch instant	2-5
LMP_temp_rand	17	13	DM1	m → s	random number	2-17
LMP_temp_key	17	14	DM1	m → s	key	2-17
LMP_test_activate	1	56	DM1/ DV	m → s	-	
LMP_test_control	10	57	DM1	m → s	test scenario	2
					hopping mode	3
					TX frequency	4
					RX frequency	5
					power control mode	6
					poll period	7
					packet type	8
					length of test data	9-10
LMP_timing_accuracy_req	1	47	DM1/ DV	m ↔ s	-	
LMP_timing_accuracy_res	3	48	DM1/ DV	m ↔ s	drift	2
					jitter	3
LMP_unit_key	17	10	DM1	m ↔ s	key	2-17
LMP_unpark_BD_ADDR _req	variable	29	DM1	m → s	timing control flags	2
					D_B	3-4
					LT_ADDR 1st unpark	5:0-2
					LT_ADDR 2nd unpark	5:4-6
					BD_ADDR 1st unpark	6-11
					BD_ADDR 2nd unpark	12-17

Table 5.1: Coding of the different LM PDUs.

LMP PDU	Length (bytes)	op code	Packet type	Possible direction	Contents	Position in payload
LMP_unpark_PM_ADDR_req	variable	30	DM1	m → s	timing control flags	2
					D_B	3-4
					LT_ADDR 1st unpark	5:0-3
					LT_ADDR 2nd unpark	5:4-7
					PM_ADDR 1st unpark	6
					PM_ADDR 2nd unpark	7
					LT_ADDR 3rd unpark	8:0-3
					LT_ADDR 4th unpark	8:4-7
					PM_ADDR 3rd unpark	9
					PM_ADDR 4th unpark	10
					LT_ADDR 5th unpark	11:0-3
					LT_ADDR 6th unpark	11:4-7
					PM_ADDR 6th unpark	12
					PM_ADDR 6th unpark	13
					LT_ADDR 7th unpark	14:0-3
					PM_ADDR 7th unpark	15
LMP_unsniff_req	1	24	DM1/ DV	m ↔ s	-	
LMP_use_semi_ permanent_key	1	50	DM1/ DV	m → s	-	
LMP_version_req	6	37	DM1/ DV	m ↔ s	VersNr	2
					CompId	3-4
					SubVersNr	5-6
LMP_version_res	6	38	DM1/ DV	m ↔ s	VersNr	2
					CompId	3-4
					SubVersNr	5-6

Table 5.1: Coding of the different LM PDUs.

Note1: For LMP_set_broadcast_scan_window, LMP_modify_beacon, LMP_unpark_BD_ADDR_req and LMP_unpark_PM_ADDR_req the parameter D_B is optional. This parameter is only present if bit0 of *timing control flags* is 1.

If the parameter is not included, the position in payload for all parameters following D_B are decreased by 2.

Note2: For LMP_unpark_BD_ADDR the LT_ADDR and the BD_ADDR of the 2nd unparked slave are optional. If only one slave is unparked *LT_ADDR 2nd unpark* shall be zero and *BD_ADDR 2nd unpark* is left out.

Note3: For LMP_unpark_PM_ADDR the LT_ADDR and the PM_ADDR of the 2nd – 7th unparked slaves are optional. If N slaves are unparked, the fields up to and including the Nth unparked slave are present. If N is odd, the *LT_ADDR (N+1)th unpark* shall be zero. The length of the message is x + 3N/2 if N is even and x + 3(N+1)/2 -1 if N is odd, where x = 2 or 4 depending on if the D_B is incluDed Or Not (See Note1).

Note4: Parameters coincide with their namesakes in LMP_<remove>_SCO_link_req apart from the following:

1. eSCO_LT_ADDR - the eSCO connection will be active on an additional LT_ADDR that needs to be defined. The master is allowed to re-assign an active eSCO link to a different LT_ADDR.
2. D_{eSCO}, T_{eSCO} - as per LMP_SCO_link_req but with a greater flexibility in values (e.g. no longer fixed with respect to HV1, HV2, and HV3 packet choice).
3. W_{eSCO} - the eSCO retransmission window size (in slots)
4. packet type and packet length may be prescribed differently in Master-to-Slave or Slave-to-Master directions for asynchronous eSCO links
5. packet length (in bytes) - eSCO packet types no longer have fixed length
6. negotiation state – this is used to better enable the negotiation of the negotiable parameters: D_{eSCO}, T_{eSCO}, W_{eSCO}, eSCO packet type M->S, eSCO packet type S->M, packet length M->S, packet length S->M. When responding to an eSCO link request with a new suggestion for these parameters, this flag may be set to 1 to indicate that the last received negotiable parameters are possible, but the new parameters specified in the response eSCO link request would be preferable, to 2 to indicate that the last received negotiable parameters are not possible as they cause a reserved slot violation or to 3 to indicate that the last received negotiable parameters would cause a latency violation. The flag shall be set to zero in the initiating LMP_eSCO_link_req.

5.2 PARAMETER DEFINITIONS

Name	Length (bytes)	Type	Unit	Detailed	Mandatory range
access scheme	1	u_int4		0: polling technique 1-15: Reserved	
AFH_channel_classification	10	multiple bytes	-	This parameter contains 40 2-bit fields. The n^{th} (numbering from 0) such field defines the classification of channels *2n* and *2n+1*, other than the 39^{th} field which just contains the classification of channel 78. Each field interpreted as an integer whose values indicate: 0 = unknown 1 = good 2 = reserved 3 = bad	
AFH_channel_map	10	multiple bytes	-	If *AFH_mode* is *AFH_enabled*, this parameter contains 79 1-bit fields, otherwise the contents are reserved. The n^{th} (numbering from 0) such field (in the range 0 to 78) contains the value for channel *n*. Bit 79 is reserved (set to 0 when transmitted and ignored when received) The 1-bit field is interpreted as follows: 0: channel *n* is unused 1: channel *n* is used	
AFH_instant	4	u_int32	slots	Bits 27:1 of the Bluetooth master clock value at the time of switching hop sequences. Must be even.	
AFH_max_interval	2	u_int16	slots	Range is 0x0640 to 0xBB80 slots (1 to 30s)	
AFH_min_interval	2	u_int16	slots	Range is 0x0640 to 0xBB80 slots (1 to 30s)	

Table 5.2: Parameters in LM PDUs.

Name	Length (bytes)	Type	Unit	Detailed	Mandatory range
AFH_mode	1	u_int8	-	0: AFH_disabled 1: AFH_enabled 2-255: Reserved	
AFH_reporting_mode	1	u_int8	-	0: AFH_reporting_disabled 1: AFH_reporting_enabled 2-255: reserved	
air mode	1	u_int8		0: μ-law log 1: A-law log 2: CVSD 3: transparent data 4-255: Reserved	See Table 5.3 on page 310
AR_ADDR	1	u_int8			
authentication response	4	multiple bytes			
BD_ADDR	6	multiple bytes		BD_ADDR of the sending device	
broadcast scan window	2	u_int16	slots		
clock offset	2	u_int16	1.25ms	($CLKN_{16-2}$ slave - $CLKN_{16-2}$ master) mod 2^{15} MSbit of second byte not used.	
CompId	2	u_int16		see Bluetooth Assigned Numbers (https://www.bluetooth.org/foundry/assignnumb/document/assigned_numbers)	
D_{access}	1	u_int8	slots		

Table 5.2: Parameters in LM PDUs.

Name	Length (bytes)	Type	Unit	Detailed	Mandatory range
data rate	1	u_int8		When in Basic Rate mode: bit0 = 0: use FEC bit0 = 1: do not use FEC bit1-2=0: No packet-size preference available bit1-2=1: use 1-slot packets bit1-2=2: use 3-slot packets bit1-2=3: use 5-slot packets When in Enhanced Data Rate mode: bit3-4=0: use DM1 packets bit3-4=1: use 2 Mbps packets bit3-4=2: use 3 Mbps packets bit3-4=3: reserved bit5-6=0: No packet-size preference available bit5-6=1: use 1-slot packets bit5-6=2: use 3-slot packets bit5-6=3: use 5-slot packets bit7: Reserved - shall be zero	
D_B	2	u_int16	slots		
Δ_B	1	u_int8	slots		
D_{Bsleep}	1	u_int8			
D_{eSCO}	1	u_int8	slots	Valid range is 0 - 254 slots	See Table 5.3 on page 310
drift	1	u_int8	ppm		
D_{sco}	1	u_int8	slots	Only even values are valid[1]	0 to T_{sco} -2
D_{sniff}	2	u_int16	slots	Only even values are valid[1]	0 to (T_{sniff} - 2)

Table 5.2: Parameters in LM PDUs.