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 155. The LMP_switch_req PDU may be sent anytime after the baseband paging procedure has completed.

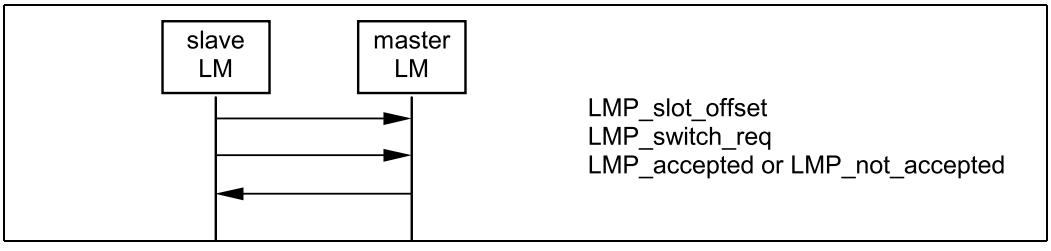
Support for LMP_slot_offset is mandatory if LMP_switch_req is supported.

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 4.23: 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 96). 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 96) 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.

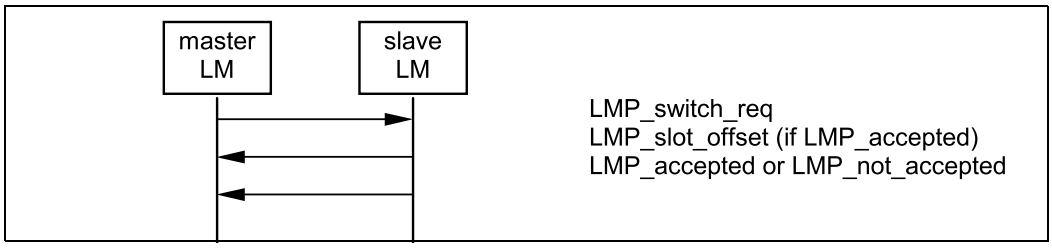


Sequence 45: Role switch (slave initiated).

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 96) 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 96) and responds with an LMP_slot_offset PDU immediately followed by an

Petitioner Kiosoft Exhibit 1011 Part 2
Page 1

LMP_accepted PDU. When the role switch has been completed at the baseband (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 46: 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 155. The default value for max_slots shall be used.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 2

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 165 for details.

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

Table 4.24: 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.

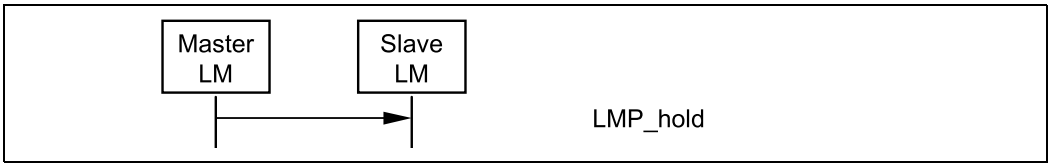
4.5.1.1 Master forces hold mode

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 96). 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 47: Master forces slave into hold mode.

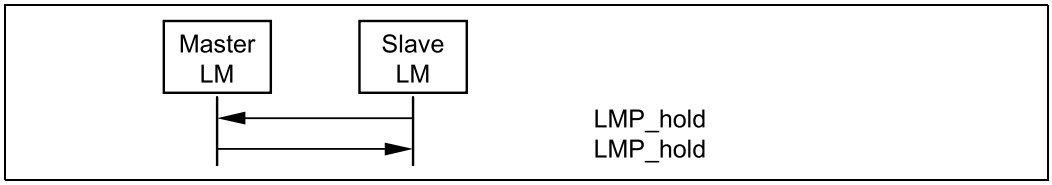
4.5.1.2 Slave forces hold mode

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 96). 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 48: 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.

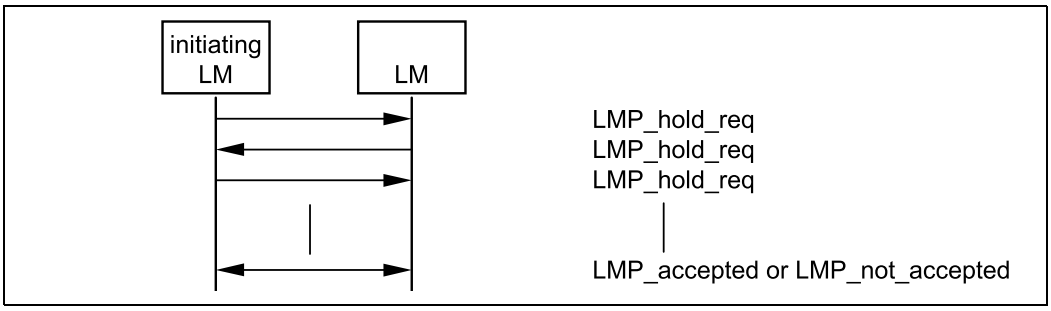
The initiating LM shall pause traffic on the ACL-U logical link (see Baseband Specification, Section 5.3.1, on page 96). On receiving an LMP_hold_req PDU the receiving LM shall complete the transmission of the current packet on the ACL logical transport and then shall suspend transmission on the ACL-U 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

Petitioner Kiosoft Exhibit 1011 Part 2
Page 4

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 49: 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 165 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 83).

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 165. 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 4.25: 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 4.25: PDUs used for park state.

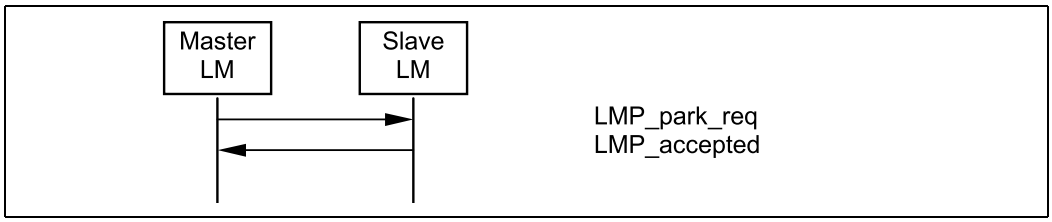
4.5.2.1 Master requests slave to enter park state

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 96) 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 96). 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.

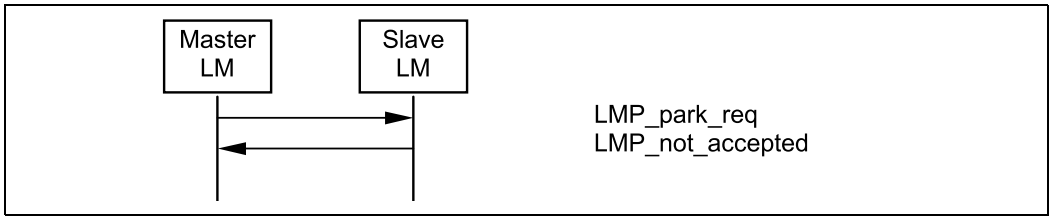
When the master receives an LMP_accepted PDU it shall start a timer for 6*Tpoll slots. When this timer expires the slave is in park state and the LT_ADDR may be re-used.

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



Sequence 50: 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 51: Slave rejects to enter into park state

4.5.2.2 Slave requests to enter park state

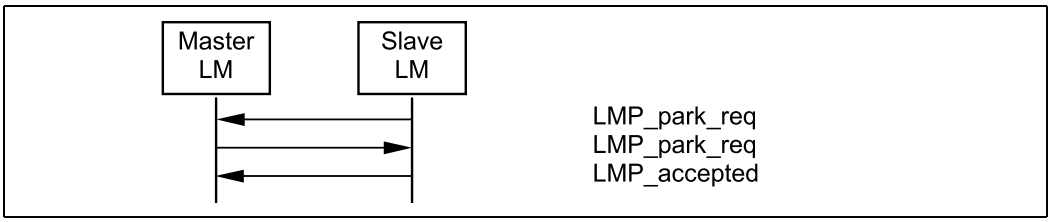
The slave can request park state. The slave LM shall pause traffic on the ACL-U logical link (see Baseband Specification, Section 5.3.1, on page 96) and then send an LMP_park_req PDU. When sent by the slave, the parameters PM_ADDR and AR_ADDR are not valid and the other parameters represent suggested 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 96) 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.

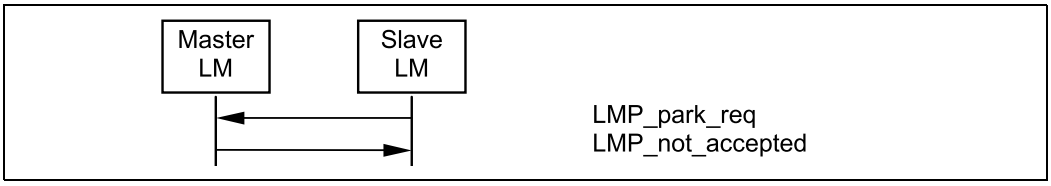
If the master never receives the LMP_accepted PDU then a link supervision timeout will occur.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 8



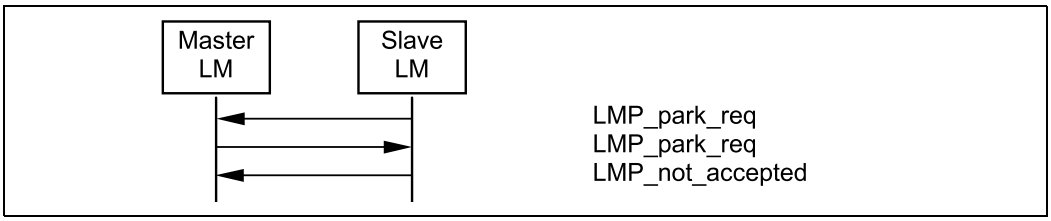
Sequence 52: Slave requests to enter park state and accepts master's beacon parameters.

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 53: Master rejects slave's request to enter park state

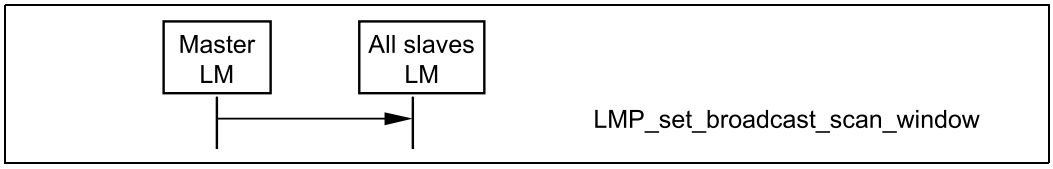
If the slave does not accept the parameters in the LMP_park_req PDU sent from the master it shall respond with an LMP_not_accepted PDU and both devices shall re-enable transmission on the ACL-U logical link.



Sequence 54: Slave requests to enter park state, but rejects master's beacon parameters.

4.5.2.3 Master sets up broadcast scan window

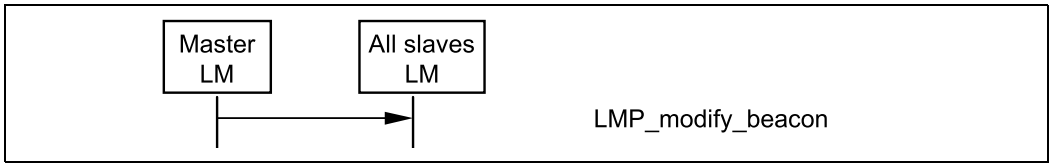
If more broadcast capacity is needed than the beacon train, the master may indicate to the slaves that more broadcast information will follow the beacon train by sending an LMP_set_broadcast_scan_window PDU. This message shall be sent in a broadcast packet at the beacon slot(s). The scan window shall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



Sequence 55: 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.



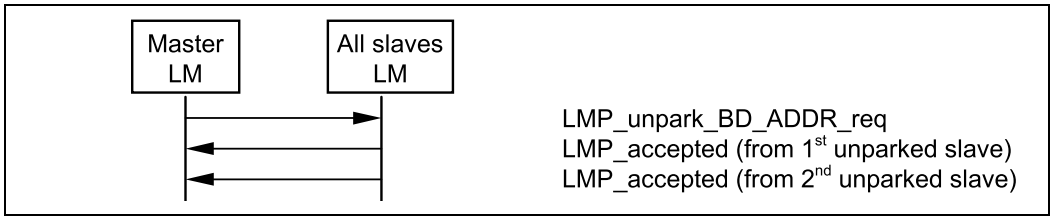
Sequence 56: Master modifies beacon parameters.

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 171 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.

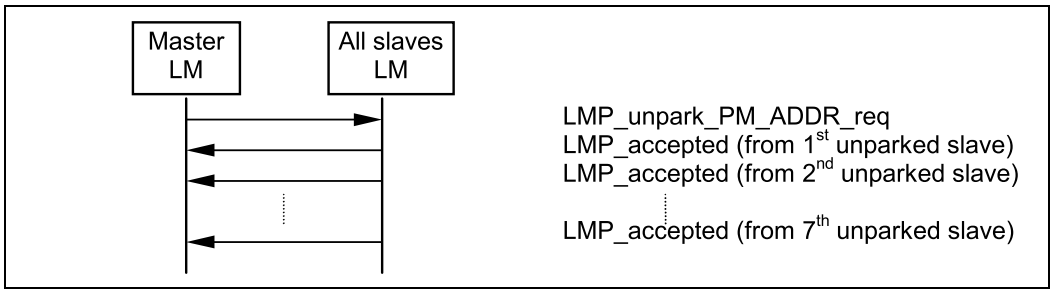
One PDU is used where the parked device is identified with the PM_ADDR, and another PDU is used where it is identified with the BD_ADDR. Both messages have variable length depending on the number of slaves the master unparks. For each slave the master wishes to unpark an LT_ADDR followed by the PM_ADDR 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 57: Master unparks slaves addressed with their BD_ADDR.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 10



Sequence 58: 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 4.26: 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.

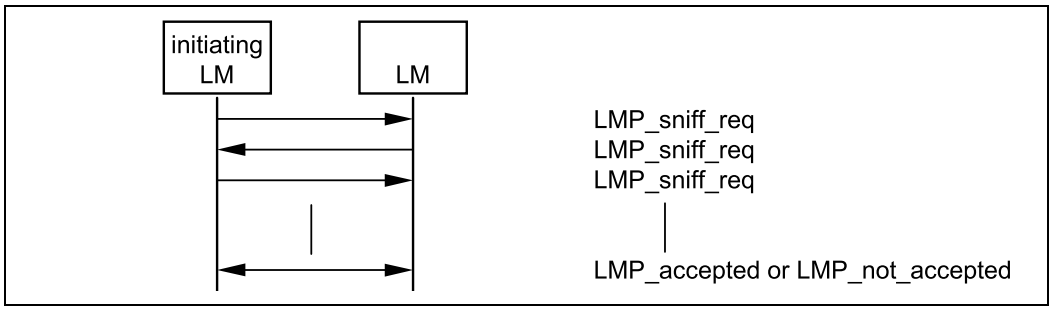
The process is initiated by sending an LMP_sniff_req PDU containing a set of parameters. The receiving LM shall then decide whether to reject the attempt by sending an LMP_not_accepted PDU, to suggest different parameters by replying with an LMP_sniff_req PDU or to accept the request.

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.



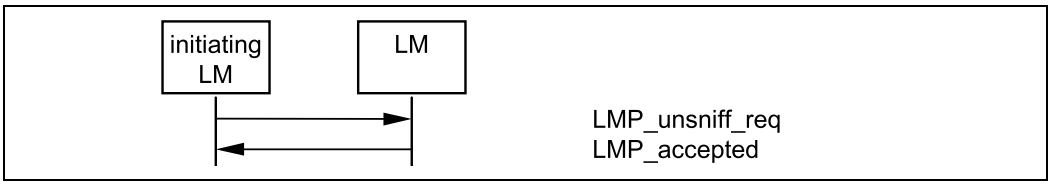
Sequence 59: Negotiation for sniff mode.

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 60: 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 150 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 4.27: PDUs used for managing the SCO links.

4.6.1.1 Master initiates an SCO link

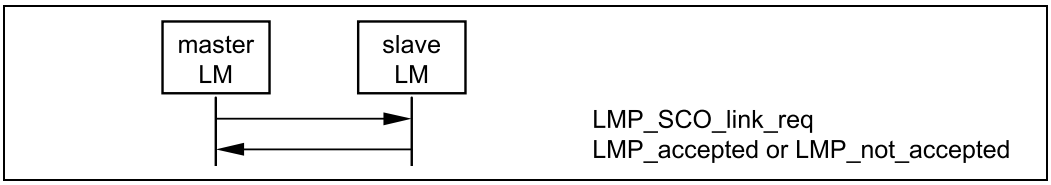
When establishing an SCO link the master sends a request, a LMP_SCO_link_req PDU, with parameters that specify the timing, packet type and coding that will be used on the SCO link. Each of the SCO packet types supports three different voice coding formats on the air-interface: μ-law log PCM, A-law log 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.

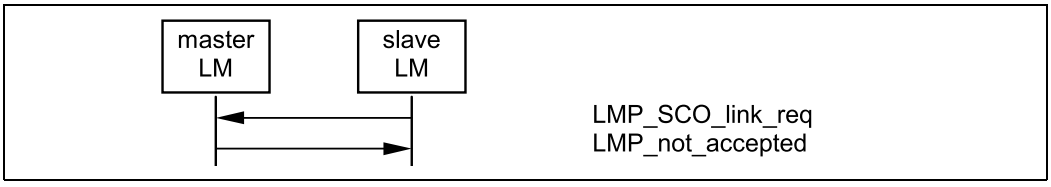


Sequence 61: Master requests an SCO link.

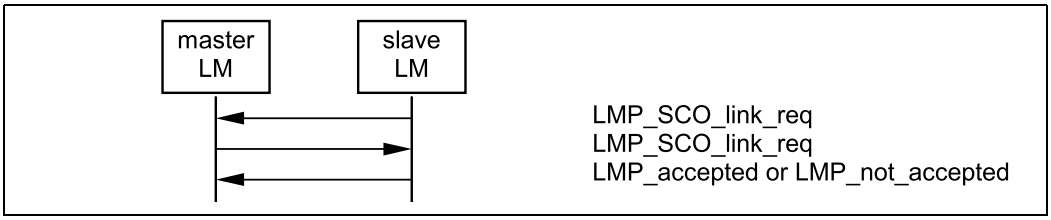
4.6.1.2 Slave initiates an SCO link

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



Sequence 63: Master accepts slave's request for an SCO link.

4.6.1.3 Master requests change of SCO parameters

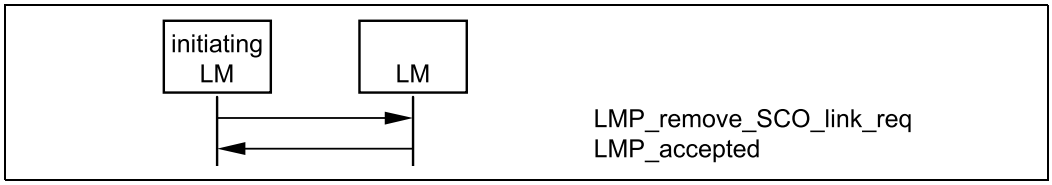
The master sends an LMP_SCO_link_req PDU, where the SCO handle is the handle of the SCO link the master wishes to change parameters for. If the slave accepts the new parameters, it replies with an LMP_accepted PDU and the SCO link will change to the new parameters. If the slave does not accept the new 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 256.

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 257.

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 64: 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.

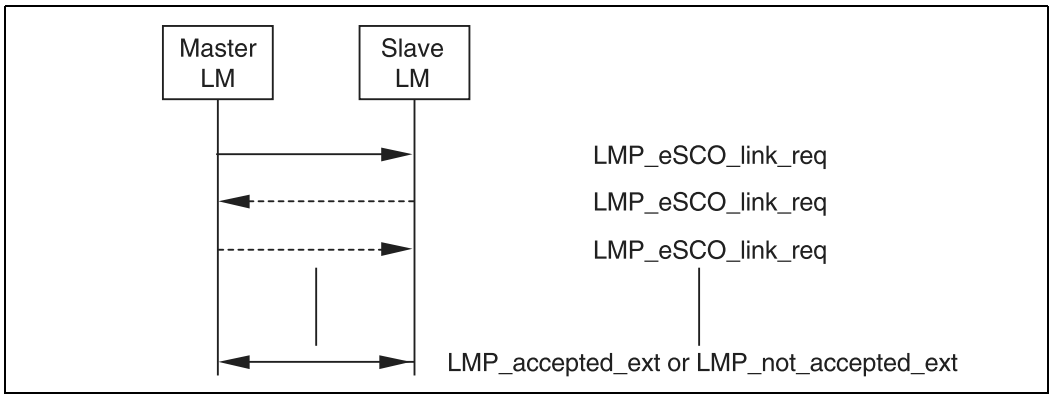
M/O	PDU	Contents
O(31)	LMP_eSCO_link_req	eSCO handle eSCO LT_ADDR timing control flags D_{eSCO} T_{eSCO} W_{eSCO} eSCO packet type M->S eSCO packet type S->M packet length M->S packet length S->M air mode negotiation state
O(31)	LMP_remove_eSCO_link_req	eSCO handle error

Table 4.28: 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

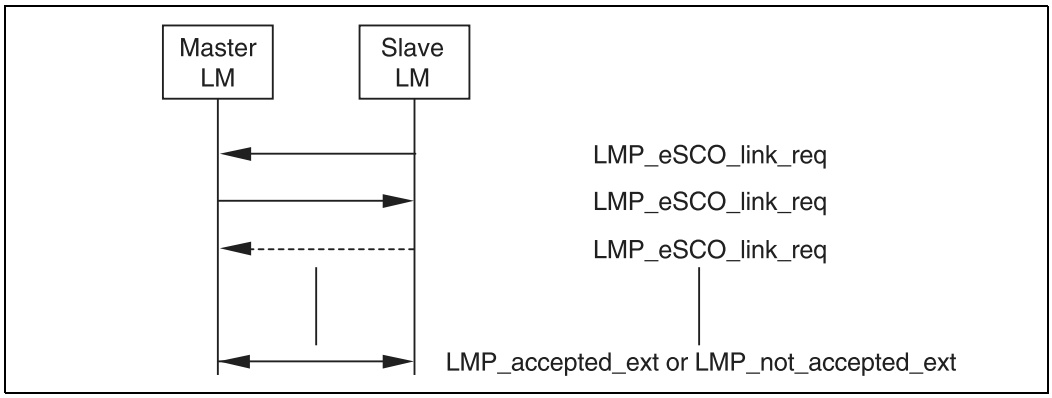
When establishing an eSCO link the master sends an LMP_eSCO_link_req PDU specifying all parameters. The slave may accept this with an LMP_accepted_ext PDU, reject it with an LMP_not_accepted_ext PDU, or respond with its own LMP_eSCO_link_req specifying alternatives for some or all parameters. The slave 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 262.



Sequence 65: Master requests an eSCO link.

4.6.2.2 Slave initiates an eSCO link

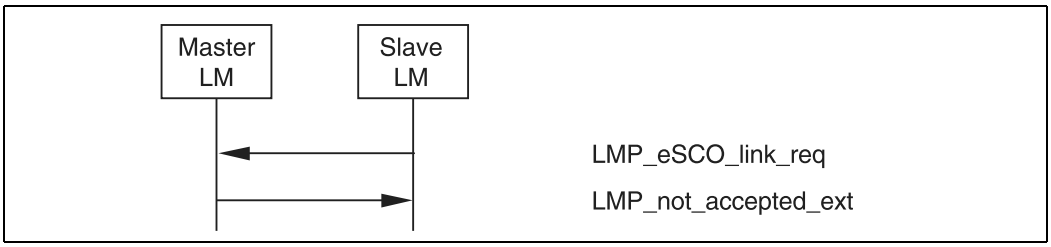
When attempting to establish an eSCO link the slave shall send an LMP_eSCO_link_req PDU specifying all parameters, with the exception of eSCO LT_ADDR and eSCO handle, which are invalid. The latter shall be set to zero. The master may respond to this with an LMP_eSCO_link_req PDU, filling in these missing 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 66: Slave requests an eSCO link.

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 262.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 18



Sequence 67: Master rejects slave's request for an eSCO link.

4.6.2.3 Master or slave requests change of eSCO parameters

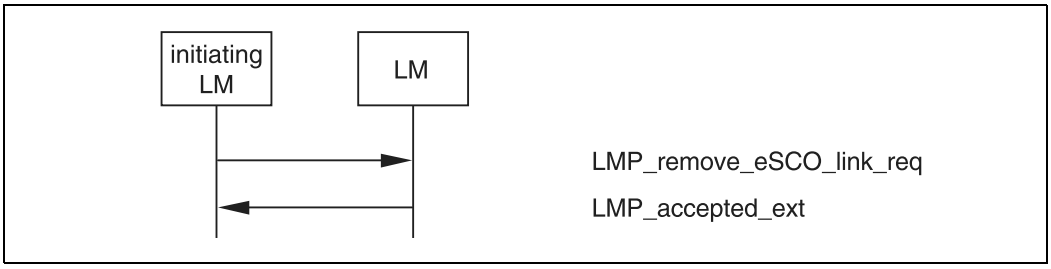
The master or slave may request a renegotiation of the eSCO parameters. The master or slave shall send an LMP_eSCO_link_req PDU with the eSCO handle of the eSCO link the device wishes to renegotiate. The remote device may accept the changed parameters immediately with LMP_accepted_ext PDU, or the negotiation 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 negotation 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 260.

During re-negotiation, the eSCO LT_ADDR and eSCO handle shall not be renegotiated and shall be set to the originally negotiated values. The negotiation shall use the procedures defined in Section 4.6.2.5 on page 262.

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 68: 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.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 20

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 233[vol. 3] 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.

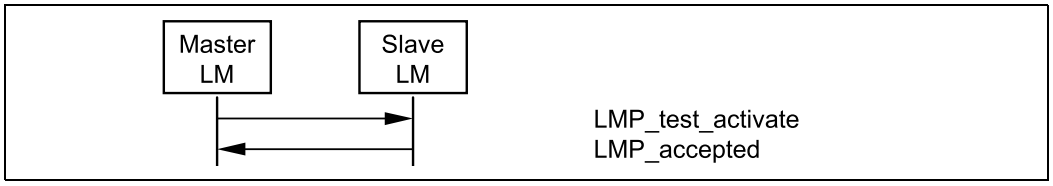
  The tester sends an LMP command that shall force the DUT to enter test mode. The DUT shall terminate all normal operation before entering the test mode.

  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.

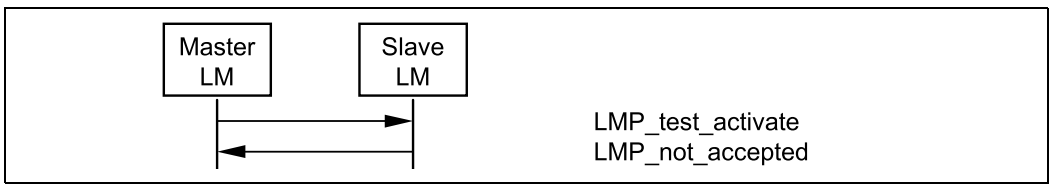
- If the activation is performed locally using a HW or SW interface, the DUT shall terminate all normal operation before entering the test mode.

  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 69: Activation of test mode successful.



Sequence 70: 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.

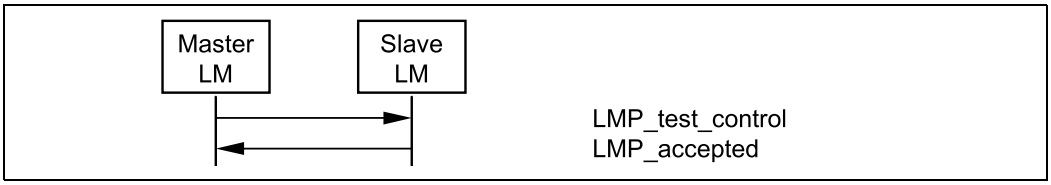
4.7.2 Control of test mode

Control and configuration is performed using special LMP commands (see Section 4.7.3 on page 266). 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.

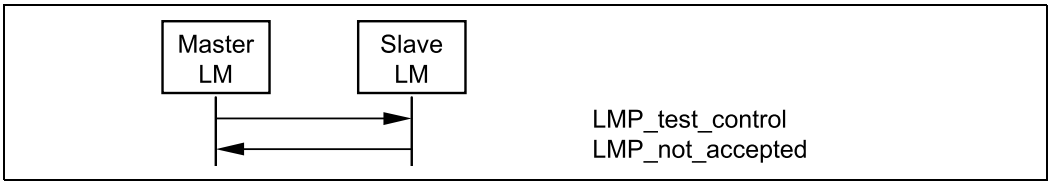
A Bluetooth device in test mode shall ignore all LMP commands not related to control of the test mode. LMP commands dealing with power control and the request for LMP features (LMP_features_req), and adaptive frequency hopping (LMP_set_AFH, LMP_channel_classification_req and LMP_channel_classification) 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 71: Control of test mode successful.



Sequence 72: Control of test mode rejected since slave is not in test mode.

4.7.3 Summary of test mode PDUs

Table 4.29 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 4.30 on page 266 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 4.29: 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 4.30: 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 2-15 : 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 4.30: 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 ≤ k ≤ 93	0 ≤ k ≤ 78
RX frequency	same as TX frequency	0 ≤ k ≤ 78
Poll period		not applicable (set to 0)
Length of test sequence	depends on packet type: DH1: <= 27 bytes DH3: <= 183 bytes DH5: <= 339 bytes AUX1: <= 29 bytes HV3: = 30 bytes EV3: <= 30 bytesEV5: <= 180 bytes	For ACL and SCO packets: not applicable (set to 0) For eSCO packets: EV3: <= 1-30 bytes EV4: <= 1-120 bytes EV5: <= 1-180 bytes

Table 4.31: 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
					SCO packet type M->S	9
					SCO 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_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
					$N_{acc\text{-}slots}$	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
LMP_quality_of_service_ req	4	42	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_remove_eSCO_link_req [] see Note4 on page 276	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	-	
LMP_supervision_timeout	3	55	DM1/ DV	m →s	supervision timeout	2-3

Table 5.1: Coding of the different LM PDUs.

LMP PDU	Length (bytes)	op code	Packet type	Possible direction	Contents	Position in payload
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 2^{nd} unparked slave are optional. If only one slave is unparked *LT_ADDR 2^{nd} unpark* shall be zero and *BD_ADDR 2^{nd} unpark* is left out.

Note3: For LMP_unpark_PM_ADDR the LT_ADDR and the PM_ADDR of the 2^{nd} – 7^{th} unparked slaves are optional. If N slaves are unparked, the fields up to and including the N^{th} 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.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 34

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.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 35

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 283
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		
data rate	1	u_int8		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 bit3-7: Reserved	

Table 5.2: Parameters in LM PDUs.

Name	Length (bytes)	Type	Unit	Detailed	Mandatory range
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 283
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)
encryption mode	1	u_int8		0: no encryption 1: encryption 2: encryption 3 -255: Reserved	
error code	1	u_int8		See Part D on page 291	
escape op code	1	u_int8		Identifies which escape op code is being acknowledged: range 124-127	
eSCO handle	1	u_int8			
eSCO LT_ADDR	1	u_int8		Logical transport address for the eSCO logical transport. The range is extended to 8 bits compared with the normal LT_ADDR field: range 0-7.	0 - 7
eSCO packet type	1	u_int8		0x00: NULL/POLL 0x07: EV3 0x0C: EV4 0x0D: EV5 Other values are reserved	If the value is 0x00 the POLL packet shall be used by the master, the NULL packet shall be used by the slave. See Table 5.3 on page 283
extended features	8	multiple bytes		One page of extended features	
extended op code	1	u_int8		Which extended op code is being acknowledged	
features	8	multiple bytes		See Table 3.2 on page 207	

Table 5.2: Parameters in LM PDUs.

Name	Length (bytes)	Type	Unit	Detailed	Mandatory range
features page	1	u_int8		Identifies which page of extended features is being requested. 0 means standard features 1-255 other feature pages	
hold instant	4	u_int32	slots	Bits 27:1 of the master Bluetooth clock value	
hold time	2	u_int16	slots	Only even values are valid[1]	0x0014 to 0x8000; shall not exceed (supervisionTO * 0.999)
jitter	1	u_int8	µs		
key	16	multiple bytes			
key size	1	u_int8	byte		
key size mask	2	u_int16		Bit mask of supported broadcast encryption key sizes: least significant bit is support for length 1, and so on. The bit shall be one if the key size is supported.	
LT_ADDR	1	u_int4			
M_{access}	1	u_int4		number of access windows	
max slots	1	u_int8	slots		
max supported page	1	u_int8		Highest page of extended features which contains a non-zero bit for the originating device. Range 0-255	
$N_{acc\text{-}slots}$	1	u_int8	slots		
name fragment	14	multiple bytes		UTF-8 characters.	
name length	1	u_int8	bytes		
name offset	1	u_int8	bytes		
N_B	1	u_int8			
N_{BC}	1	u_int8			
N_{Bsleep}	1	u_int8			

Table 5.2: Parameters in LM PDUs.

Name	Length (bytes)	Type	Unit	Detailed	Mandatory range
negotiation state	1	u_int8		0: Initiate negotiation 1: the latest received set of negotiable parameters were possible but these parameters are preferred. 2: the latest received set of negotiable parameters would cause a reserved slot violation. 3: the latest received set of negotiable parameters would cause a latency violation. 4: the latest received set of of negotiable parameters are not supported. Other values are reserved	
N_{poll}	1	u_int8			
op code	1	u_int8			
packet length	2	uint16	bytes	Length of the eSCO payload 0 for POLL/NULL 1-30 for EV3 1-120 for EV4 1-180 for EV5 Other values are invalid	See Table 5.3 on page 283
paging scheme	1	u_int8		0: mandatory scheme 1-255: Reserved	
paging scheme settings	1	u_int8		For mandatory scheme: 0: R0 1: R1 2: R2 3-255: Reserved	
PM_ADDR	1	u_int8			
poll interval	2	u_int16	slots	Only even values are valid[1]	0x0006 to 0x1000
random number	16	multiple bytes			

Table 5.2: Parameters in LM PDUs.

Name	Length (bytes)	Type	Unit	Detailed	Mandatory range
reserved(n)	n	u_int8		Reserved for future use – must be 0 when transmitted, ignore value when received	
SCO handle	1	u_int8			
SCO packet	1	u_int8		0: HV1 1: HV2 2: HV3 3-255: Reserved	
slot offset	2	u_int16	µs	0 ≤ slot offset < 1250	
sniff attempt	2	u_int16	received slots	Number of receive slots	1 to $T_{sniff}/2$
sniff timeout	2	u_int16	received slots	Number of receive slots	0 to 0x0028
SubVersNr	2	u_int16		Defined by each company	
supervision timeout	2	u_int16	slots	0 means an infinite timeout	0 and 0x0190 to 0xFFFF
switch instant	4	u_int32	slots	Bits 27:1 of the master Bluetooth clock value	
T_{access}	1	u_int8	slots		
T_B	2	u_int16	slots		
T_{eSCO}	1	u_int8	slots	Valid range is 4 – 254 slots	See Table 5.3 on page 283
timing control flags	1	u_int8		bit0 = 0: no timing change bit0 = 1: timing change bit1 = 0: use initialization 1 bit1 = 1: use initialization 2 bit2 = 0: access window bit2 = 1: no access window bit3-7: Reserved	
T_{sco}	1	u_int8	slots	Only even values are valid[1]	2 to 6
T_{sniff}	2	u_int16	slots	Only even values are valid[1]	0x0006 to 0x0540; shall not exceed (supervisionTO * 0.999)

Table 5.2: Parameters in LM PDUs.

Name	Length (bytes)	Type	Unit	Detailed	Mandatory range
VersNr	1	u_int8		See Bluetooth Assigned Numbers, (http://www.bluetooth.org/assigned-numbers.htm)	
W_{eSCO}	1	u_int8	slots	Number of slots in the retransmission window Valid range is 0 – 254 slots	See Table 5.3 on page 283

Table 5.2: Parameters in LM PDUs.

1. If a device receives an LMP PDU with an odd value in this parameter field the PDU should be rejected with an error code of *invalid LMP parameters*.

	EV3	EV4	EV5
D_{eSCO}	0-4 (even)	0-14 (even)	0-14 (even)
T_{eSCO}	6	16 (even)	16 (even)
W_{eSCO}	0-4 (even)	0-6 (even)	0-6 (even)
eSCO packet type M->S	EV3	EV3, EV4	EV3, EV5
eSCO packet type S->M	EV3	EV3, EV4	EV3, EV5
packet length M->S	30	1-120	1-180
packet length S->M	30	1-120	1-180
air mode	At least one of A-law, mu-law, CVSD, transparent	transparent	transparent

Table 5.3: Mandatory parameter ranges for eSCO packet types

5.3 DEFAULT VALUES

Devices shall use these values before anything else has been negotiated:

Parameter	Value
AFH_mode	AFH_disabled
AFH_reporting_mode	AFH_reporting_disabled
drift	250
jitter	10
max slots	1
poll interval	40

Table 5.4: Default values.

6 LIST OF FIGURES

Figure 1.1: Link Manager Protocol signalling layer. 195
Figure 2.1: Transmission of a message from master to slave. 198
Figure 2.2: Payload body when LMP PDUs are sent. 199
Figure 2.3: Symbols used in sequence diagrams. 201
Figure 4.1: Connection establishment. 209
Figure 4.2: Slot offset for role switch. 242

Sequence 1: Connection closed by sending LMP_detach. 211
Sequence 2: A device requests a change of the other device's TX power. ..212
Sequence 3: The TX power cannot be increased. 212
Sequence 4: The TX power cannot be decreased. 212
Sequence 5: Master Enables AFH. 214
Sequence 6: Master disables AFH. 214
Sequence 7: Master Updates AFH. 215
Sequence 8: Channel classification reporting. 217
Sequence 9: Setting the link supervision timeout. 218
Sequence 10: A notifies B to enable CQDDR 219

Sequence 11: B sends A a preferred packet type 219
Sequence 12: Master notifies slave of quality of service. 220
Sequence 13: Device accepts new quality of service 221
Sequence 14: Device rejects new quality of service. 221
Sequence 15: Negotiation for page mode. 222
Sequence 16: Negotiation for page scan mode 222
Sequence 17: Device allows Remote Device to use a maximum number of slots. 223
Sequence 18: Device requests a maximum number of slots. Remote Device accepts. 223
Sequence 19: Device requests a maximum number of slots. Remote Device rejects. 223
Sequence 20: Authentication. Claimant has link key. 224

Sequence 21: Authentication fails. Claimant has no link key. 225
Sequence 22: Pairing accepted. Responder has a variable PIN. Initiator has a variable or fixed PIN. 226
Sequence 23: Responder has a fixed PIN and initiator has a variable PIN. 227
Sequence 24: Both devices have a fixed PIN. 227
Sequence 25: Responder rejects pairing. 227
Sequence 26: Creation of the link key. 228
Sequence 27: Successful change of the link key. 229
Sequence 28: Change of the link key not possible since the other device uses

a unit key. 229
Sequence 29: Change to a temporary link key. 230
Sequence 30: Link key changed to the semi-permanent link key. 231

Sequence 31: Negotiation for encryption mode. 233
Sequence 32: Encryption key size negotiation successful. 234
Sequence 33: Encryption key size negotiation failed. 234
Sequence 34: Start of encryption. 234
Sequence 35: Stop of encryption. 235
Sequence 36: Request for supported encryption key sizes. 236
Sequence 37: The requested device supports timing accuracy information. .237
Sequence 38: The requested device does not support timing accuracy information. 237
Sequence 39: Clock offset requested. 238
Sequence 40: Request for LMP version. 239
Sequence 41: Request for supported features. 240
Sequence 42: Request for extended features. 240

Sequence 43: Device’s name requested and it responses. 241
Sequence 44: Slot offset information is sent. 242
Sequence 45: Role switch (slave initiated). 243
Sequence 46: Role switch (master initiated). 244
Sequence 47: Master forces slave into hold mode. 245
Sequence 48: Slave forces master into hold mode. 246
Sequence 49: Negotiation for hold mode. 247
Sequence 50: Slave accepts to enter park state. 250
Sequence 51: Slave rejects to enter into park state 250

Sequence 52: Slave requests to enter park state and accepts master's beacon parameters. 251
Sequence 53: Master rejects slave's request to enter park state 251
Sequence 54: Slave requests to enter park state, but rejects master's beacon parameters. 251
Sequence 55: Master notifies all slaves of increase in broadcast capacity. 251
Sequence 56: Master modifies beacon parameters. 252
Sequence 57: Master unparks slaves addressed with their BD_ADDR. ...252
Sequence 58: Master unparks slaves addressed with their PM_ADDR. ...253
Sequence 59: Negotiation for sniff mode. 254
Sequence 60: Slave moved from sniff mode to active mode. 255
Sequence 61: Master requests an SCO link. 257
Sequence 62: Master rejects slave’s request for an SCO link. 257
Sequence 63: Master accepts slave’s request for an SCO link. 257
Sequence 64: SCO link removed. 258

Sequence 65: Master requests an eSCO link. 260
Sequence 66: Slave requests an eSCO link. 260

Sequence 67: Master rejects slave's request for an eSCO link. 261
Sequence 68: eSCO link removed 261
Sequence 69: Activation of test mode successful. 264
Sequence 70: Activation of test mode fails. Slave is not allowed to enter test mode. 264
Sequence 71: Control of test mode successful. 265
Sequence 72: Control of test mode rejected since slave is not in test mode. .265

Petitioner Kiosoft Exhibit 1011 Part 2
Page 46

7 LIST OF TABLES

Table 2.1: General response messages. 202
Table 3.1: Feature definitions. 203
Table 3.2: Feature mask definition....207
Table 4.1: PDUs used for connection establishment. 210
Table 4.2: PDU used for detach.....210
Table 4.3: PDUs used for power control. 211
Table 4.4: PDUs used for AFH....213
Table 4.5: PDUs used for Channel Classification Reporting....216
Table 4.6: PDU used to set the supervision timeout. 218
Table 4.7: PDUs used for quality driven change of the data rate....219
Table 4.8: PDUs used for quality of service. 220
Table 4.9: PDUs used to request paging scheme....222
Table 4.10: PDUs used to control the use of multi-slot packets....223
Table 4.11: PDUs used for authentication. 224
Table 4.12: PDUs used for pairing 226
Table 4.13: PDUs used for change of link key. 229
Table 4.14: PDUs used to change the current link key. 230
Table 4.15: PDUs used for handling encryption....232
Table 4.16: PDUs used for encryption key size request 236
Table 4.17: PDUs used for requesting timing accuracy information. 237
Table 4.18: PDUs used for clock offset request. 238
Table 4.19: PDUs used for LMP version request. 239
Table 4.20: PDUs used for features request. 240
Table 4.21: PDUs used for name request. 241
Table 4.22: PDU used for slot offset information. 242
Table 4.23: PDUs used for role switch. 243
Table 4.24: PDUs used for hold mode. 245
Table 4.25: PDUs used for park state. 248
Table 4.26: PDUs used for sniff mode. 253
Table 4.27: PDUs used for managing the SCO links. 256
Table 4.28: PDUs used for managing the eSCO links 259
Table 4.29: LMP messages used for Test Mode....266
Table 4.30: Parameters used in LMP_Test_Control PDU....266
Table 4.31: Restrictions for Parameters used in LMP_Test_Control PDU..268
Table 5.1: Coding of the different LM PDUs. 269
Table 5.2: Parameters in LM PDUs. 277
Table 5.3: Mandatory parameter ranges for eSCO packet types....283
Table 5.4: Default values. 284

Petitioner Kiosoft Exhibit 1011 Part 2
Page 48



ERROR CODES

Petitioner Kiosoft Exhibit 1011 Part 2
Page 50

CONTENTS

1 Overview of Error Codes 295

1.1 Usage Descriptions 295
1.2 HCI Command Errors 295
1.3 List of Error Codes 296

2 Error Code Descriptions 299

2.1 Unknown HCI Command (0X01) 299
2.2 Unknown Connection Identifier (0X02) 299
2.3 Hardware Failure (0X03) 299
2.4 Page Timeout (0X04) 299
2.5 Authentication Failure (0X05) 299
2.6 PIN Missing (0X06) 299
2.7 Memory Capacity Exceeded (0X07) 299
2.8 Connection Timeout (0X08) 300
2.9 Connection Limit Exceeded (0X09) 300
2.10 Synchronous Connection Limit to a Device Exceeded (0X0A) 300
2.11 ACL Connection Already Exists (0X0B) 300
2.12 Command Disallowed (0X0C) 300
2.13 Connection Rejected due to Limited Resources (0X0D) 300
2.14 Connection Rejected due to Security Reasons (0X0E) 300
2.15 Connection Rejected due to Unacceptable BD_ADDR (0X0F) 301
2.16 Connection Accept Timeout Exceeded (0X10) 301
2.17 Unsupported Feature or Parameter Value (0X11) 301
2.18 Invalid HCI Command Parameters (0X12) 301
2.19 Remote User Terminated Connection (0X13) 301
2.20 Remote Device Terminated Connection due to Low Resources (0X14) 302
2.21 Remote Device Terminated Connection due to Power Off (0X15) 302
2.22 Connection Terminated by Local Host (0X16) 302
2.23 Repeated Attempts (0X17) 302
2.24 Pairing not Allowed (0X18) 302
2.25 Unknown LMP PDU (0X19) 302
2.26 Unsupported Remote Feature (0X1A) 302
2.27 SCO Offset Rejected (0X1B) 302
2.28 SCO Interval Rejected (0X1C) 303
2.29 SCO Air Mode Rejected (0X1D) 303
2.30 INvalid LMP Parameters (0X1E) 303
2.31 Unspecified Error (0X1F) 303
2.32 Unsupported LMP Parameter Value (0X20) 303
2.33 Role Change Not Allowed (0X21) 303

2.34 LMP Response Timeout (0X22)....303
2.35 LMP Error Transaction Collision (0X23) 304
2.36 LMP PDU Not Allowed (0X24)....304
2.37 Encryption Mode Not Acceptable (0X25)....304
2.38 Link Key Can Not be Changed (0X26) 304
2.39 Requested Qos is Not Supported (0X27) 304
2.40 Instant Passed (0X28) 304
2.41 Pairing with Unit Key Not Supported (0X29)....304
2.42 DIFFERENT TRANSACTION COLLISION (0X2A) 304
2.43 QoS Unacceptable Parameter (0X2C)....304
2.44 QoS Rejected (0X2D)....305
2.45 Channel Classification Not Supported (0X2E)....305
2.46 Insufficient Security (0X2F)....305
2.47 Parameter out of Mandatory Range (0X30)....305
2.48 Role Switch Pending (0X32)....305
2.49 Reserved Slot Violation (0X34)....305
2.50 Role Switch Failed (0X35) 305

1 OVERVIEW OF ERROR CODES

This document lists the various possible error codes. When a command fails, or an LMP message needs to indicate a failure, error codes are used to indicate the reason for the error. Error codes have a size of one octet.

1.1 USAGE DESCRIPTIONS

The purpose of this section is to give descriptions of how the error codes should be used. It is beyond the scope of this document to give detailed descriptions of all situations where error codes can be used, especially as this may be implementation dependent.

1.2 HCI COMMAND ERRORS

If an HCI Command that should generate an HCI_Command_Complete event generates an error then this error shall be reported in the HCI_Command_Complete event.

If an HCI Command that sent an HCI_Command_Status with the error code ‘Success’ to the host before processing may find an error during execution then the error may be reported in the normal completion command for the original command or in an HCI_Command_Status event.

Some HCI Commands may generate errors that need to be reported to be host, but there is insufficient information to determine how the command would normally be processed. In this case, two events can be used to indicate this to the host, the HCI_Command_Complete event and HCI_Command_Status events. Which of the two events is used is implementation-dependent.

1.3 LIST OF ERROR CODES

The error code of 0x00 means Success. The possible range of failure error codes is 0x01-0xFF. Section 2 provides an error code usage description for each failure error code.

Error Code	Name
0x00	Success
0x01	Unknown HCI Command
0x02	Unknown Connection Identifier
0x03	Hardware Failure
0x04	Page Timeout
0x05	Authentication Failure
0x06	PIN Missing
0x07	Memory Capacity Exceeded
0x08	Connection Timeout
0x09	Connection Limit Exceeded
0x0A	Synchronous Connection Limit To A Device Exceeded
0x0B	ACL Connection Already Exists
0x0C	Command Disallowed
0x0D	Connection Rejected due to Limited Resources
0x0E	Connection Rejected Due To Security Reasons
0x0F	Connection Rejected due to Unacceptable BD_ADDR
0x10	Connection Accept Timeout Exceeded
0x11	Unsupported Feature or Parameter Value
0x12	Invalid HCI Command Parameters
0x13	Remote User Terminated Connection
0x14	Remote Device Terminated Connection due to Low Resources
0x15	Remote Device Terminated Connection due to Power Off
0x16	Connection Terminated By Local Host
0x17	Repeated Attempts
0x18	Pairing Not Allowed
0x19	Unknown LMP PDU
0x1A	Unsupported Remote Feature
0x1B	SCO Offset Rejected

Table 1.1: List of Possible Error Codes

Error Code	Name
0x1C	SCO Interval Rejected
0x1D	SCO Air Mode Rejected
0x1E	Invalid LMP Parameters
0x1F	Unspecified Error
0x20	Unsupported LMP Parameter Value
0x21	Role Change Not Allowed
0x22	LMP Response Timeout
0x23	LMP Error Transaction Collision
0x24	LMP PDU Not Allowed
0x25	Encryption Mode Not Acceptable
0x26	Link Key Can Not be Changed
0x27	Requested QoS Not Supported
0x28	Instant Passed
0x29	Pairing With Unit Key Not Supported
0x2A	Different Transaction Collision
0x2B	Reserved
0x2C	QoS Unacceptable Parameter
0x2D	QoS Rejected
0x2E	Channel Classification Not Supported
0x2F	Insufficient Security
0x30	Parameter Out Of Mandatory Range
0x31	Reserved
0x32	Role Switch Pending
0x33	Reserved
0x34	Reserved Slot Violation
0x35	Role Switch Failed

Table 1.1: List of Possible Error Codes

Petitioner Kiosoft Exhibit 1011 Part 2
Page 56

2 ERROR CODE DESCRIPTIONS

2.1 UNKNOWN HCI COMMAND (0X01)

The Unknown HCI Command error code indicates that the controller does not understand the HCI Command Packet OpCode that the host sent. The OpCode given might not correspond to any of the OpCodes specified in this document, or any vendor-specific OpCodes, or the command may not have been implemented.

2.2 UNKNOWN CONNECTION IDENTIFIER (0X02)

The Unknown Connection Identifier error code indicates that a command was sent from the host that should identify a connection, but that connection does not exist.

2.3 HARDWARE FAILURE (0X03)

The Hardware Failure error code indicates to the host that something in the controller has failed in a manner that cannot be described with any other error code. The meaning implied with this error code is implementation dependent.

2.4 PAGE TIMEOUT (0X04)

The Page Timeout error code indicates that a page timed out because of the Page Timeout configuration parameter. This error code may occur only with the HCI_Remote_Name_Request and HCI_Create_Connection commands.

2.5 AUTHENTICATION FAILURE (0X05)

The Authentication Failure error code indicates that pairing or authentication failed due to incorrect results in the pairing or authentication procedure. This could be due to an incorrect PIN or Link Key.

2.6 PIN MISSING (0X06)

The PIN Missing error code is used when pairing failed because of a missing PIN.

2.7 MEMORY CAPACITY EXCEEDED (0X07)

The Memory Capacity Exceeded error code indicates to the host that the controller has run out of memory to store new parameters.

2.8 CONNECTION TIMEOUT (0X08)

The Connection Timeout error code indicates that the link supervision timeout has expired for a given connection.

2.9 CONNECTION LIMIT EXCEEDED (0X09)

The Connection Limit Exceeded error code indicates that an attempt to create another connection failed because the controller is already at its limit of the number of connections it can support. The number of connections a device can support is implementation dependent.

2.10 SYNCHRONOUS CONNECTION LIMIT TO A DEVICE EXCEEDED (0X0A)

The Synchronous Connection Limit to a Device Exceeded error code indicates that the controller has reached the limit to the number of synchronous connections that can be achieved to a device. The number of synchronous connections a device can support is implementation dependent.

2.11 ACL CONNECTION ALREADY EXISTS (0X0B)

The ACL Connection Already Exists error code indicates that an attempt to create a new ACL Connection to a device when there is already a connection to this device.

2.12 COMMAND DISALLOWED (0X0C)

The Command Disallowed error code indicates that the command requested cannot be executed because the controller is in a state where it cannot process this command at this time. This error shall not be used for command OpCodes where the error code Unknown HCI Command is valid.

2.13 CONNECTION REJECTED DUE TO LIMITED RESOURCES (0X0D)

The Connection Rejected Due To Limited Resources error code indicates that an incoming connection was rejected due to limited resources.

2.14 CONNECTION REJECTED DUE TO SECURITY REASONS (0X0E)

The Connection Rejected Due To Security Reasons error code indicates that a connection was rejected due to security requirements not being fulfilled, like authentication or pairing.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 58

2.15 CONNECTION REJECTED DUE TO UNACCEPTABLE BD_ADDR (0X0F)

The Connection Rejected due to Unacceptable BD_ADDR error code indicates that a connection was rejected because this device does not accept the BD_ADDR. This may be because the device will only accept connections from specific BD_ADDRs.

2.16 CONNECTION ACCEPT TIMEOUT EXCEEDED (0X10)

The Connection Accept Timeout Exceeded error code indicates that the Connection Accept Timeout has been exceeded for this connection attempt.

2.17 UNSUPPORTED FEATURE OR PARAMETER VALUE (0X11)

The Unsupported Feature Or Parameter Value error code indicates that a feature or parameter value in an LMP message or HCI Command is not supported.

2.18 INVALID HCI COMMAND PARAMETERS (0X12)

The Invalid HCI Command Parameters error code indicates that at least one of the HCI command parameters is invalid.

This shall be used when :

- the parameter total length is invalid.
- a command parameter is an invalid type.
- a connection identifier does not match the corresponding event.
- a parameter value must be even.
- a parameter is outside of the specified range.
- two or more parameter values have inconsistent values.

Note: An invalid type can be, for example, when an SCO connection handle is used where an ACL connection handle is required.

2.19 REMOTE USER TERMINATED CONNECTION (0X13)

The Remote User Terminated Connection error code indicates that the user on the remote device terminated the connection.

2.20 REMOTE DEVICE TERMINATED CONNECTION DUE TO LOW RESOURCES (0X14)

The Remote Device Terminated Connection due to Low Resources error code indicates that the remote device terminated the connection because of low resources.

2.21 REMOTE DEVICE TERMINATED CONNECTION DUE TO POWER OFF (0X15)

The Remote Device Terminated Connection due to Power Off error code indicates that the remote device terminated the connection because the device is about to power off.

2.22 CONNECTION TERMINATED BY LOCAL HOST (0X16)

The Connection Terminated By Local Host error code indicates that the local device terminated the connection.

2.23 REPEATED ATTEMPTS (0X17)

The Repeated Attempts error code indicates that the controller is disallowing an authentication or pairing procedure because too little time has elapsed since the last authentication or pairing attempt failed.

2.24 PAIRING NOT ALLOWED (0X18)

The Pairing Not Allowed error code indicates that the device does not allow pairing. For example, when a device only allows pairing during a certain time window after some user input allows pairing.

2.25 UNKNOWN LMP PDU (0X19)

The Unknown LMP PDU error code indicates that the controller has received an unknown LMP opcode.

2.26 UNSUPPORTED REMOTE FEATURE (0X1A)

The Unsupported Remote Feature error code indicates that the remote device does not support the feature associated with the issued command or LMP PDU.

2.27 SCO OFFSET REJECTED (0X1B)

The SCO Offset Rejected error code indicates that the offset requested in the LMP_SCO_link_req message has been rejected.

2.28 SCO INTERVAL REJECTED (0X1C)

The SCO Interval Rejected error code indicates that the interval requested in the LMP_SCO_link_req message has been rejected.

2.29 SCO AIR MODE REJECTED (0X1D)

The SCO Air Mode Rejected error code indicates that the air mode requested in the LMP_SCO_link_req message has been rejected.

2.30 INVALID LMP PARAMETERS (0X1E)

The Invalid LMP Parameters error code indicates that some LMP message parameters were invalid. This shall be used when :

- the PDU length is invalid.
- a parameter value must be even.
- a parameter is outside of the specified range.
- two or more parameters have inconsistent values.

2.31 UNSPECIFIED ERROR (0X1F)

The Unspecified Error error code indicates that no other error code specified is appropriate to use.

2.32 UNSUPPORTED LMP PARAMETER VALUE (0X20)

The Unsupported LMP Parameter Value error code indicates that an LMP message contains at least one parameter value that is not supported by the controller at this time. This is normally used after a long negotiation procedure, for example during an LMP_hold_req, LMP_sniff_req and LMP_encryption_key_size_req message exchanges.

2.33 ROLE CHANGE NOT ALLOWED (0X21)

The Role Change Not Allowed error code indicates that a controller will not allow a role change at this time.

2.34 LMP RESPONSE TIMEOUT (0X22)

The LMP Response Timeout error code indicates that an LMP transaction failed to respond within the LMP response timeout.

2.35 LMP ERROR TRANSACTION COLLISION (0X23)

The LMP Error Transaction Collision error code indicates that an LMP transaction has collided with the same transaction that is already in progress.

2.36 LMP PDU NOT ALLOWED (0X24)

The LMP PDU Not Allowed error code indicates that a controller sent an LMP message with an opcode that was not allowed.

2.37 ENCRYPTION MODE NOT ACCEPTABLE (0X25)

The Encryption Mode Not Acceptable error code indicates that the requested encryption mode is not acceptable at this time.

2.38 LINK KEY CAN NOT BE CHANGED (0X26)

The Link Key Can Not be Changed error code indicates that a link key can not be changed because a fixed unit key is being used.

2.39 REQUESTED QoS NOT SUPPORTED (0X27)

The Requested QoS Not Supported error code indicates that the requested Quality of Service is not supported.

2.40 INSTANT PASSED (0X28)

The Instant Passed error code indicates that an LMP PDU that includes an instant can not be performed because the instant when this would have occurred has passed.

2.41 PAIRING WITH UNIT KEY NOT SUPPORTED (0X29)

The Pairing With Unit Key Not Supported error code indicates that it was not possible to pair as a unit key was requested and it is not supported.

2.42 DIFFERENT TRANSACTION COLLISION (0X2A)

The Different Transaction Collision error code indicates that an LMP transaction was started that collides with an ongoing transaction.

2.43 QoS UNACCEPTABLE PARAMETER (0X2C)

The QoS Unacceptable Parameter error code indicates that the specified quality of service parameters could not be accepted at this time, but other parameters may be acceptable.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 62

2.44 QoS REJECTED (0X2D)

The QoS Rejected error code indicates that the specified quality of service parameters can not be accepted and QoS negotiation should be terminated.

2.45 CHANNEL CLASSIFICATION NOT SUPPORTED (0X2E)

The Channel Classification Not Supported error code indicates that the controller can not perform channel classification because it is not supported.

2.46 INSUFFICIENT SECURITY (0X2F)

The Insufficient Security error code indicates that the HCI command or LMP message sent is only possible on an encrypted link.

2.47 PARAMETER OUT OF MANDATORY RANGE (0X30)

The Parameter Out Of Mandatory Range error code indicates that a parameter value requested is outside the mandatory range of parameters for the given HCI command or LMP message.

2.48 ROLE SWITCH PENDING (0X32)

The Role Switch Pending error code indicates that a Role Switch is pending. This can be used when an HCI command or LMP message can not be accepted because of a pending role switch. This can also be used to notify a peer device about a pending role switch.

2.49 RESERVED SLOT VIOLATION (0X34)

The Reserved Slot Violation error code indicates that the current Synchronous negotiation was terminated with the negotiation state set to Reserved Slot Violation.

2.50 ROLE SWITCH FAILED (0X35)

The Role Switch Failed error code indicates that a role switch was attempted but it failed and the original piconet structure is restored. The switch may have failed because the TDD switch or piconet switch failed.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 64

Core System Package [Controller volume]
Part E



HOST CONTROLLER INTERFACE FUNCTIONAL SPECIFICATION

This document describes the functional specification for the Host Controller Interface (HCI).
The HCI provides a command interface to the baseband controller and link manager, and access to configuration parameters. This interface provides a uniform method of accessing the Bluetooth baseband capabilities.

CONTENTS

1 Introduction 317
1.1 Lower Layers of the Bluetooth Software Stack 317
2 Overview of Host Controller Transport Layer....319
3 Overview of Commands and Events 321
3.1 Generic Events....322
3.2 Device Setup....322
3.3 Controller Flow Control 323
3.4 Controller Information....323
3.5 Controller Configuration 324
3.6 Device Discovery 325
3.7 Connection Setup 327
3.8 Remote Information....329
3.9 Synchronous Connections 330
3.10 Connection State....331
3.11 Piconet Structure....332
3.12 Quality of Service....333
3.13 Physical Links 334
3.14 Host Flow Control....335
3.15 Link Information 336
3.16 Authentication and Encryption 337
3.17 Testing....339
3.18 Alphabetical List of Commands and Events 340
4 HCI Flow Control 345
4.1 Host to Controller Data Flow Control 345
4.2 Controller to Host Data Flow Control 346
4.3 Disconnection Behaviour 347
4.4 Command Flow Control 347
4.5 Command Error Handling 348
5 HCI Data Formats 349
5.1 Introduction 349
5.2 Data and Parameter Formats....349
5.3 Connection Handles....350
5.4 Exchange of HCI-Specific Information 351
5.4.1 HCI Command Packet....351
5.4.2 HCI ACL Data Packets....353
5.4.3 HCI Synchronous Data Packets....355
5.4.4 HCI Event Packet....356

6 HCI Configuration Parameters 357
6.1 Scan Enable 357
6.2 Inquiry Scan Interval 357
6.3 Inquiry Scan Window 358
6.4 Inquiry Scan Type 358
6.5 Inquiry Mode 358
6.6 Page Timeout 359
6.7 Connection Accept Timeout 359
6.8 Page Scan Interval 360
6.9 Page Scan Window 360
6.10 Page Scan Period Mode 360
6.11 Page Scan Type 361
6.12 Voice Setting 361
6.13 PIN Type 362
6.14 Link Key 362
6.15 Authentication Enable 362
6.16 Encryption Mode 363
6.17 Failed Contact Counter 364
6.18 Hold Mode Activity 364
6.19 Link Policy Settings 365
6.20 Flush Timeout 366
6.21 Num Broadcast Retransmissions 366
6.22 Link Supervision Timeout 367
6.23 Synchronous Flow Control Enable 367
6.24 Local Name 368
6.25 Class Of Device 368
6.26 Supported Commands 369

7 HCI Commands and Events 373
7.1 Link Control Commands 373
7.1.1 Inquiry Command 373
7.1.2 Inquiry Cancel Command 375
7.1.3 Periodic Inquiry Mode Command 376
7.1.4 Exit Periodic Inquiry Mode Command 379
7.1.5 Create Connection Command 380
7.1.6 Disconnect Command 383
7.1.7 Create Connection Cancel Command 384
7.1.8 Accept Connection Request Command 386
7.1.9 Reject Connection Request Command 388
7.1.10 Link Key Request Reply Command 389
7.1.11 Link Key Request Negative Reply Command 391
7.1.12 PIN Code Request Reply Command 392

7.1.13 PIN Code Request Negative Reply Command 394
7.1.14 Change Connection Packet Type Command 395
7.1.15 Authentication Requested Command.....398
7.1.16 Set Connection Encryption Command 399
7.1.17 Change Connection Link Key Command 400
7.1.18 Master Link Key Command 401
7.1.19 Remote Name Request Command 402
7.1.20 Remote Name Request Cancel Command 404
7.1.21 Read Remote Supported Features Command.....405
7.1.22 Read Remote Extended Features Command 406
7.1.23 Read Remote Version Information Command.....407
7.1.24 Read Clock Offset Command.....408
7.1.25 Read LMP Handle Command 409
7.1.26 Setup Synchronous Connection Command 411
7.1.27 Accept Synchronous Connection Request Command 415
7.1.28 Reject Synchronous Connection Request Command .419
7.2 Link Policy Commands.....420
7.2.1 Hold Mode Command 420
7.2.2 Sniff Mode Command.....422
7.2.3 Exit Sniff Mode Command.....425
7.2.4 Park State Command 426
7.2.5 Exit Park State Command 428
7.2.6 QoS Setup Command 429
7.2.7 Role Discovery Command.....431
7.2.8 Switch Role Command.....432
7.2.9 Read Link Policy Settings Command 433
7.2.10 Write Link Policy Settings Command 434
7.2.11 Read Default Link Policy Settings Command 436
7.2.12 Write Default Link Policy Settings Command 437
7.2.13 Flow Specification Command 438

7.3 Controller & Baseband Commands 440
7.3.1 Set Event Mask Command 440
7.3.2 Reset Command 442
7.3.3 Set Event Filter Command 443
7.3.4 Flush Command 448
7.3.5 Read PIN Type Command 450
7.3.6 Write PIN Type Command 451
7.3.7 Create New Unit Key Command 452
7.3.8 Read Stored Link Key Command 453
7.3.9 Write Stored Link Key Command 454
7.3.10 Delete Stored Link Key Command 456
7.3.11 Write Local Name Command 457
7.3.12 Read Local Name Command 458
7.3.13 Read Connection Accept Timeout Command 459
7.3.14 Write Connection Accept Timeout Command 460
7.3.15 Read Page Timeout Command 461
7.3.16 Write Page Timeout Command 462
7.3.17 Read Scan Enable Command 463
7.3.18 Write Scan Enable Command 464
7.3.19 Read Page Scan Activity Command 465
7.3.20 Write Page Scan Activity Command 467
7.3.21 Read Inquiry Scan Activity Command 468
7.3.22 Write Inquiry Scan Activity Command 469
7.3.23 Read Authentication Enable Command 470
7.3.24 Write Authentication Enable Command 471
7.3.25 Read Encryption Mode Command 472
7.3.26 Write Encryption Mode Command 473
7.3.27 Read Class of Device Command 474
7.3.28 Write Class of Device Command 475
7.3.29 Read Voice Setting Command 476
7.3.30 Write Voice Setting Command 477
7.3.31 Read Automatic Flush Timeout Command 478
7.3.32 Write Automatic Flush Timeout Command 479
7.3.33 Read Num Broadcast Retransmissions Command 480
7.3.34 Write Num Broadcast Retransmissions Command 481
7.3.35 Read Hold Mode Activity Command 482
7.3.36 Write Hold Mode Activity Command 483
7.3.37 Read Transmit Power Level Command 484

7.3.38 Read Synchronous Flow Control Enable Command...486
7.3.39 Write Synchronous Flow Control Enable Command...487
7.3.40 Set Controller To Host Flow Control Command 488
7.3.41 Host Buffer Size Command.....489
7.3.42 Host Number Of Completed Packets Command.....491
7.3.43 Read Link Supervision Timeout Command.....493
7.3.44 Write Link Supervision Timeout Command 494
7.3.45 Read Number Of Supported IAC Command.....496
7.3.46 Read Current IAC LAP Command 497
7.3.47 Write Current IAC LAP Command.....498
7.3.48 Read Page Scan Period Mode Command 500
7.3.49 Write Page Scan Period Mode Command 501
7.3.50 Set AFH Host Channel Classification Command 502
7.3.51 Read Inquiry Scan Type Command 503
7.3.52 Write Inquiry Scan Type Command 504
7.3.53 Read Inquiry Mode Command 505
7.3.54 Write Inquiry Mode Command 506
7.3.55 Read Page Scan Type Command 507
7.3.56 Write Page Scan Type Command 508
7.3.57 Read AFH Channel Assessment Mode Command 509
7.3.58 Write AFH Channel Assessment Mode Command 510
7.4 Informational Parameters.....512
7.4.1 Read Local Version Information Command.....512
7.4.2 Read Local Supported Commands Command.....514
7.4.3 Read Local Supported Features Command.....515
7.4.4 Read Local Extended Features Command 516
7.4.5 Read Buffer Size Command.....518
7.4.6 Read BD_ADDR Command.....520
7.5 Status Parameters.....521
7.5.1 Read Failed Contact Counter Command 521
7.5.2 Reset Failed Contact Counter Command 523
7.5.3 Read Link Quality Command 524
7.5.4 Read RSSI Command.....525
7.5.5 Read AFH Channel Map Command 527
7.5.6 Read Clock Command 529
7.6 Testing Commands 531
7.6.1 Read Loopback Mode Command 531
7.6.2 Write Loopback Mode Command.....532
7.6.3 Enable Device Under Test Mode Command 535

7.7 Events 537
7.7.1 Inquiry Complete Event 537
7.7.2 Inquiry Result Event 538
7.7.3 Connection Complete Event 540
7.7.4 Connection Request Event 541
7.7.5 Disconnection Complete Event 543
7.7.6 Authentication Complete Event 544
7.7.7 Remote Name Request Complete Event 545
7.7.8 Encryption Change Event 546
7.7.9 Change Connection Link Key Complete Event 547
7.7.10 Master Link Key Complete Event 548
7.7.11 Read Remote Supported Features Complete Event 549
7.7.12 Read Remote Version Information Complete Event 550
7.7.13 QoS Setup Complete Event 551
7.7.14 Command Complete Event 553
7.7.15 Command Status Event 554
7.7.16 Hardware Error Event 555
7.7.17 Flush Occurred Event 555
7.7.18 Role Change Event 556
7.7.19 Number Of Completed Packets Event 557
7.7.20 Mode Change Event 558
7.7.21 Return Link Keys Event 560
7.7.22 PIN Code Request Event 561
7.7.23 Link Key Request Event 562
7.7.24 Link Key Notification Event 563
7.7.25 Loopback Command Event 564
7.7.26 Data Buffer Overflow Event 564
7.7.27 Max Slots Change Event 565
7.7.28 Read Clock Offset Complete Event 566
7.7.29 Connection Packet Type Changed Event 567
7.7.30 QoS Violation Event 569
7.7.31 Page Scan Repetition Mode Change Event 570
7.7.32 Flow Specification Complete Event 571
7.7.33 Inquiry Result with RSSI Event 573
7.7.34 Read Remote Extended Features Complete Event 575
7.7.35 Synchronous Connection Complete Event 576
7.7.36 Synchronous Connection Changed event 578

Petitioner Kiosoft Exhibit 1011 Part 2
Page 72

8 List of Figures.....581

9 List of Tables 583

Appendix A:
Deprecated Commands, Events and Configuration Parameters 585

Contents: 585

Page Scan Mode 586

Read Page Scan Mode Command 587

Write Page Scan Mode Command 588

Read Country Code Command 589

Add SCO Connection Command 590

Page Scan Mode Change Event 592

1 INTRODUCTION

This document describes the functional specifications for the Host Controller Interface (HCI). The HCI provides a uniform interface method of accessing the Bluetooth controller capabilities. The next two sections provide a brief overview of the lower layers of the Bluetooth software stack and of the Bluetooth hardware. Section 2, provides an overview of the Lower HCI Device Driver Interface on the host device. Section 4, describes the flow control used between the Host and the Controller. Section 7, describes each of the HCI Commands in details, identifies parameters for each of the commands, and lists events associated with each command.

1.1 LOWER LAYERS OF THE BLUETOOTH SOFTWARE STACK

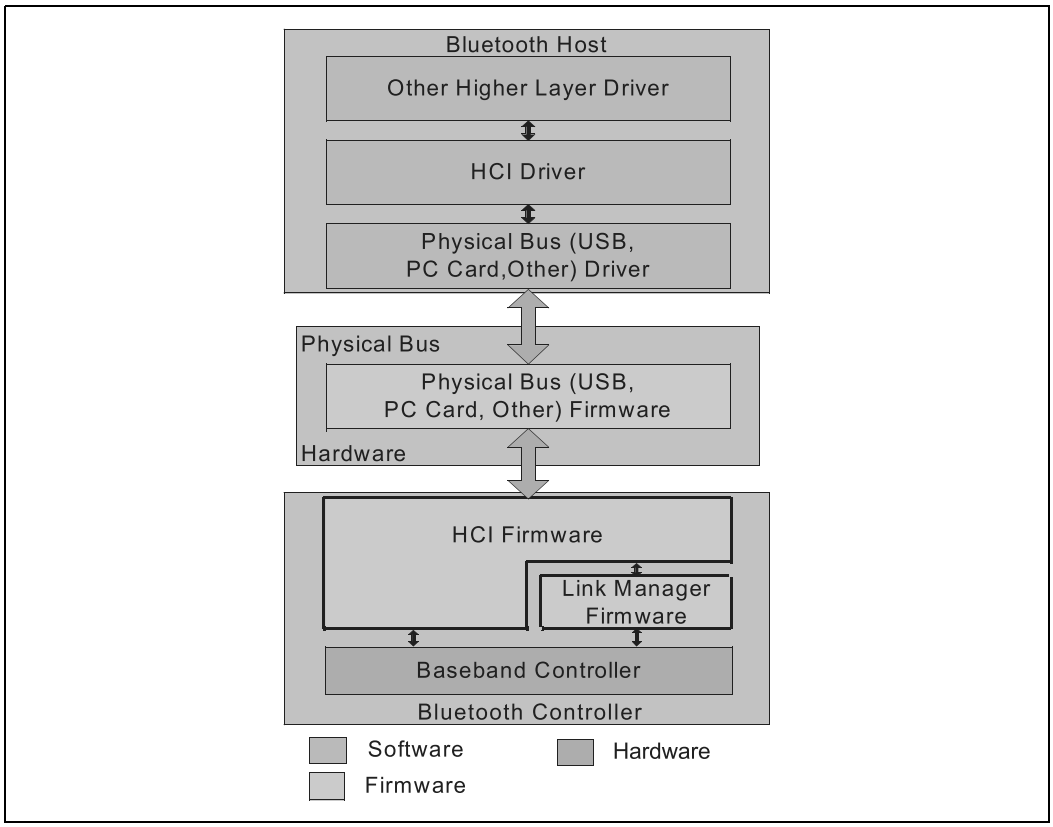


Figure 1.1: Overview of the Lower Software Layers

Figure 1.1, provides an overview of the lower software layers. The HCI firmware implements the HCI Commands for the Bluetooth hardware by accessing baseband commands link manager commands, hardware status registers, control registers, and event registers.

Several layers may exist between the HCI driver on the host system and the HCI firmware in the Bluetooth hardware. These intermediate layers, the Host

Controller Transport Layer, provide the ability to transfer data without intimate knowledge of the data.

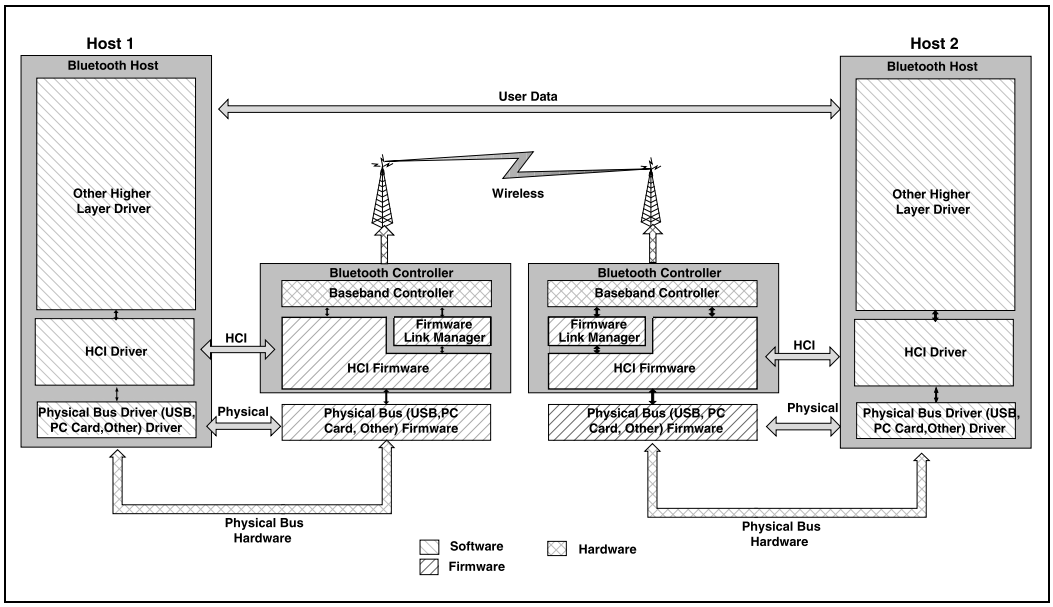


Figure 1.2: End to End Overview of Lower Software Layers to Transfer Data

Figure 1.2, illustrates the path of a data transfer from one device to another. The HCI driver on the Host exchanges data and commands with the HCI firmware on the Bluetooth hardware. The Host Control Transport Layer (i.e. physical bus) driver provides both HCI layers with the ability to exchange information with each other.

The Host will receive asynchronous notifications of HCI events independent of which Host Controller Transport Layer is used. HCI events are used for notifying the Host when something occurs. When the Host discovers that an event has occurred it will then parse the received event packet to determine which event occurred.

2 OVERVIEW OF HOST CONTROLLER TRANSPORT LAYER

The host driver stack has a transport layer between the Host Controller driver and the Host.

The main goal of this transport layer is transparency. The Host Controller driver (which interaces to the Controller) should be independent of the underlying transport technology. Nor should the transport require any visibility into the data that the Host Controller driver passes to the Controller. This allows the interface (HCI) or the Controller to be upgraded without affecting the transport layer.

The specified Host Controller Transport Layers are described in a separate volume. (See specification volume 4)

Petitioner Kiosoft Exhibit 1011 Part 2
Page 78

3 OVERVIEW OF COMMANDS AND EVENTS

The commands and events are sent between the Host and the Controller. These are grouped into logical groups by function.

Generic Events	The generic events can occur due to multiple commands, or events that can occur at any time.
Device Setup	The device setup commands are used to place the Controller into a known state.
Controller Flow Control	The controller flow control commands and events are used to control data flow from the Host to the controller.
Controller Information	The controller information commands allow the Host to discover local information about the device.
Controller Configuration	The controller configuration commands and events allow the global configuration parameters to be configured.
Device Discovery	The device discovery commands and events allow a device to discover other devices in the surrounding area.
Connection Setup	The connection setup commands and events allow a device to make a connection to another device.
Remote Information	The remote information commands and events allow information about a remote device's configuration to be discovered.
Synchronous Connections	The synchronous connection commands and events allow synchronous connections to be created
Connection State	The connection state commands and events allow the configuration of a link, especially for low power operation.
Piconet Structure	The piconet structure commands and events allow the discovery and reconfiguration of piconet.
Quality of Service	The quality of service commands and events allow quality of service parameters to be specified.
Physical Links	The physical link commands and events allow the configuration of a physical link.
Host Flow Control	The Host flow control commands and events allow flow control to be used towards the Host.
Link Information	The link information commands and events allow information about a link to be read.
Authentication and Encryption	The authentication and encryption commands and events allow authentication of a remote device and then encryption of the link.
Testing	The testing commands and events allow a device to be placed into test mode.

Table 3.1: Overview of commands and events

The version information in this section denotes the version number of the specification that this command or event was first specified

3.1 GENERIC EVENTS

The generic events occur due to multiple commands, or events that can occur at any time.

Name	Vers.	Summary description
Command Complete Event	1.1	The Command Complete event is used by the Controller to pass the return status of a command and the other event parameters for each HCI Command.
Command Status Event	1.1	The Command Status event is used to indicate that the command described by the Command_Opcode parameter has been received and the Controller is currently performing the task for this command.
Hardware Error Event	1.1	The Hardware Error event is used to indicate some type of hardware failure for the Controller.

Table 3.2: Generic events

3.2 DEVICE SETUP

The device setup group of commands are used to place the Controller into a known state.

Name	Vers.	Summary description
Reset Command	1.1	The Reset command will reset the Controller, Link Manager, and the Bluetooth radio.

Device setup

3.3 CONTROLLER FLOW CONTROL

The controller flow control group of commands and events are used to control data flow from the Host to the Controller.

Name	Vers.	Summary description
Read Buffer Size Command	1.1	The Read_Buffer_Size command returns the size of the HCI buffers. These buffers are used by the Controller to buffer data that is to be transmitted.
Number Of Completed Packets Event	1.1	The Number Of Completed Packets event is used by the Controller to indicate to the Host how many HCI Data Packets have been completed for each Connection Handle since the previous Number Of Completed Packets event was sent.

Table 3.3: Controller flow control

3.4 CONTROLLER INFORMATION

The controller information group of commands allows the Host to discover local information about the device.

Name	Vers.	Summary description
Read Local Version Information Command	1.1	The Read Local Version Information command will read the version information for the local Bluetooth device.
Read Local Supported Commands Command	1.2	The Read Local Supported Commands command requests a list of the supported HCI commands for the local device.
Read Local Supported Features Command	1.1	The Read Local Supported Features command requests a list of the supported features for the local device.
Read Local Extended Features Command	1.2	The Read Local Extended Features command requests a list of the supported extended features for the local device
Read BD_ADDR Command	1.1	The Read BD_ADDR command will read the value for the BD_ADDR parameter.

Table 3.4: Controller information

3.5 CONTROLLER CONFIGURATION

The controller configuration group of commands and events allows the global configuration parameters to be configured.

Name	Vers.	Summary description
Read Local Name Command	1.1	The Read Local Name command provides the ability to read the stored user-friendly name for the Bluetooth device.
Write Local Name Command	1.1	The Write Local Name command provides the ability to modify the user-friendly name for the Bluetooth device.
Read Class of Device Command	1.1	The Read Class of Device command will read the value for the Class of Device configuration parameter, which is used to indicate its capabilities to other devices.
Write Class of Device Command	1.1	The Write Class of Device command will write the value for the Class_of_Device configuration parameter, which is used to indicate its capabilities to other devices.
Read Number Of Supported IAC Command	1.1	The Read Number of Supported IAC command will read the value for the number of Inquiry Access Codes (IAC) that the local Bluetooth device can simultaneously listen for during an Inquiry Scan.
Read Current IAC LAP Command	1.1	The Read Current IAC LAP command will read the LAP(s) used to create the Inquiry Access Codes (IAC) that the local Bluetooth device is simultaneously scanning for during Inquiry Scans.
Write Current IAC LAP Command	1.1	The Write Current IAC LAP command will write the LAP(s) used to create the Inquiry Access Codes (IAC) that the local Bluetooth device is simultaneously scanning for during Inquiry Scans.
Read Scan Enable Command	1.1	The Read Scan Enable command will read the value for the Scan Enable configuration parameter, which controls whether or not the Bluetooth device will periodically scan for page attempts and/or inquiry requests from other Bluetooth devices.
Write Scan Enable Command	1.1	The Write Scan Enable command will write the value for the Scan Enable configuration parameter, which controls whether or not the Bluetooth device will periodically scan for page attempts and/or inquiry requests from other Bluetooth devices.

Table 3.5: Controller configuration

3.6 DEVICE DISCOVERY

The device discovery group of commands and events allow a device to discovery other devices in the surrounding area.

Name	Vers.	Summary description
Inquiry Command	1.1	The Inquiry command will cause the Bluetooth device to enter Inquiry Mode. Inquiry Mode is used to discovery other nearby Bluetooth devices.
Inquiry Result Event	1.1	The Inquiry Result event indicates that a Bluetooth device or multiple Bluetooth devices have responded so far during the current Inquiry process.
Inquiry Result with RSSI Event	1.2	The Inquiry Result with RSSI event indicates that a Bluetooth device or multiple Bluetooth devices have responded so far during the current Inquiry process.
Inquiry Cancel Command	1.1	The Inquiry Cancel command will cause the Bluetooth device to stop the current Inquiry if the Bluetooth device is in Inquiry Mode.
Inquiry Complete Event	1.1	The Inquiry Complete event indicates that the Inquiry is finished.
Periodic Inquiry Mode Command	1.1	The Periodic Inquiry Mode command is used to configure the Bluetooth device to perform an automatic Inquiry based on a specified period range.
Exit Periodic Inquiry Mode Command	1.1	The Exit Periodic Inquiry Mode command is used to end the Periodic Inquiry mode when the local device is in Periodic Inquiry Mode.
Read Inquiry Scan Activity Command	1.1	The Read Inquiry Scan Activity command will read the value for Inquiry Scan Interval and Inquiry Scan Window configuration parameters. Inquiry Scan Interval defines the amount of time between consecutive inquiry scans. Inquiry Scan Window defines the amount of time for the duration of the inquiry scan.
Write Inquiry Scan Activity Command	1.1	The Write Inquiry Scan Activity command will write the value for Inquiry Scan Interval and Inquiry Scan Window configuration parameters. Inquiry Scan Interval defines the amount of time between consecutive inquiry scans. Inquiry Scan Window defines the amount of time for the duration of the inquiry scan.
Read Inquiry Scan Type Command	1.2	The Read Inquiry Scan Type command is used to read the Inquiry Scan Type configuration parameter of the local Bluetooth device. The Inquiry Scan Type configuration parameter can set the inquiry scan to either normal or interlaced scan.

Table 3.6: Device discovery

Name	Vers.	Summary description
Write Inquiry Scan Type Command	1.2	The Write Inquiry Scan Type command is used to write the Inquiry Scan Type configuration parameter of the local Bluetooth device. The Inquiry Scan Type configuration parameter can set the inquiry scan to either normal or interlaced scan.
Read Inquiry Mode Command	1.2	The Read Inquiry Mode command is used to read the Inquiry Mode configuration parameter of the local Bluetooth device.
Write Inquiry Mode Command	1.2	The Write Inquiry Mode command is used to write the Inquiry Mode configuration parameter of the local Bluetooth device.

Table 3.6: Device discovery

3.7 CONNECTION SETUP

The connection setup group of commands and events are used to allow a device to make a connection to another device.

Name	Vers.	Summary description
Create Connection Command	1.1	The Create Connection command will cause the link manager to create an ACL connection to the Bluetooth device with the BD_ADDR specified by the command parameters.
Connection Request Event	1.1	The Connection Request event is used to indicate that a new incoming connection is trying to be established.
Accept Connection Request Command	1.1	The Accept Connection Request command is used to accept a new incoming connection request.
Reject Connection Request Command	1.1	The Reject Connection Request command is used to decline a new incoming connection request.
Create Connection Cancel Command	1.2	The Create Connection Cancel Command is used to cancel an ongoing Create Connection.
Connection Complete Event	1.1	The Connection Complete event indicates to both of the Hosts forming the connection that a new connection has been established.
Disconnect Command	1.1	The Disconnect command is used to terminate an existing connection.
Disconnection Complete Event	1.1	The Disconnection Complete event occurs when a connection has been terminated.
Read Page Timeout Command	1.1	The Read Page Timeout command will read the value for the Page Reply Timeout configuration parameter, which determines the time the Bluetooth controller will wait for the remote device to respond to a connection request before the local device returns a connection failure.
Write Page Timeout Command	1.1	The Write Page Timeout command will write the value for the Page Reply Timeout configuration parameter, which allows the Bluetooth hardware to define the amount of time a connection request will wait for the remote device to respond before the local device returns a connection failure.
Read Page Scan Activity Command	1.1	The Read Page Scan Activity command will read the values for the Page Scan Interval and Page Scan Window configuration parameters. Page Scan Interval defines the amount of time between consecutive page scans. Page Scan Window defines the duration of the page scan.

Table 3.7: Connection setup

Name	Vers.	Summary description
Write Page Scan Activity Command	1.1	The Write Page Scan Activity command will write the value for Page Scan Interval and Page Scan Window configuration parameters. Page Scan Interval defines the amount of time between consecutive page scans. Page Scan Window defines the duration of the page scan.
Page Scan Repetition Mode Change Event	1.1	The Page Scan Repetition Mode Change event indicates that the connected remote Bluetooth device with the specified Connection_Handle has successfully changed the Page_Scan_Repetition_Mode (SR)."
Read Page Scan Type Command	1.2	The Read Page Scan Type command is used to read the page scan type of the local Bluetooth device. The Page Scan Type configuration parameter can set the page scan to either normal or interlaced scan.
Write Page Scan Type Command	1.2	The Write Page Scan Type command is used to write the page scan type of the local Bluetooth device. The Page Scan Type configuration parameter can set the page scan to either normal or interlaced scan.
Read Connection Accept Timeout Command	1.1	The Read Connection Accept Timeout command will read the value for the Connection Accept Timeout configuration parameter, which allows the Bluetooth hardware to automatically deny a connection request after a specified period has occurred, and to refuse a new connection.
Write Connection Accept Timeout Command	1.1	The Write Connection Accept Timeout command will write the value for the Connection Accept Timeout configuration parameter, which allows the Bluetooth hardware to automatically deny a connection request after a specified period has occurred, and to refuse a new connection.
Read Page Scan Period Mode Command	1.1	The Read Page Scan Period Mode command is used to read the mandatory Page Scan Period Mode of the local Bluetooth device.
Write Page Scan Period Mode Command	1.1	The Write Page Scan Period Mode command is used to write the mandatory Page Scan Period Mode of the local Bluetooth device.

Table 3.7: Connection setup

3.8 REMOTE INFORMATION

The remote information group of commands and events allows information about a remote devices configuration to be discovered.

Name	Vers.	Summary description
Remote Name Request Command	1.1	The Remote Name Request command is used to obtain the user-friendly name of another Bluetooth device.
Remote Name Request Cancel Command	1.2	The Remote Name Request Cancel Command is used to cancel an ongoing Remote Name Request.
Remote Name Request Complete Event	1.1	The Remote Name Request Complete event is used to indicate a remote name request has been completed.
Read Remote Supported Features Command	1.1	The Read Remote Supported Features command requests a list of the supported features of a remote device.
Read Remote Supported Features Complete Event	1.1	The Read Remote Supported Features Complete event is used to indicate the completion of the process of the Link Manager obtaining the supported features of the remote Bluetooth device specified by the Connection Handle event parameter.
Read Remote Extended Features Command	1.2	The Read Remote Extended Features command requests a list of the supported extended features of a remote device
Read Remote Extended Features Complete Event	1.2	The Read Remote Extended Features Complete Event is used to indicate the completion of the process of the Link Manager obtaining the supported Extended features of the remote Bluetooth device specified by the connection handle event parameter.
Read Remote Version Information Command	1.1	The Read Remote Version Information command will read the values for the version information for the remote Bluetooth device.
Read Remote Version Information Complete Event	1.1	The Read Remote Version Information Complete event is used to indicate the completion of the process of the Link Manager obtaining the version information of the remote Bluetooth device specified by the Connection Handle event parameter.

Table 3.8: Remote information

3.9 SYNCHRONOUS CONNECTIONS

The synchronous connections group of commands and events allows synchronous connections to be created.

Name	Vers.	Summary description
Setup Synchronous Connection Command	1.2	The Setup Synchronous Connection command adds a new or modifies an existing synchronous logical transport (SCO or eSCO) on a physical link depending on the Connection Handle parameter specified.
Synchronous Connection Complete Event	1.2	The Synchronous Connection Complete event indicates to both the Hosts that a new Synchronous connection has been established.
Synchronous Connection Changed event	1.2	The Synchronous Connection Changed event indicates to the Host that an existing Synchronous connection has been reconfigured.
Accept Synchronous Connection Request Command	1.2	The Accept_Synchronous_Connection_Request command is used to accept an incoming request for a synchronous connection and to inform the local Link Manager about the acceptable parameter values for the synchronous connection.
Reject Synchronous Connection Request Command	1.2	The Reject_Synchronous_Connection_Request is used to decline an incoming request for a synchronous link.
Read Voice Setting Command	1.1	The Read Voice Setting command will read the values for the Voice Setting configuration parameter, which controls all the various settings for the voice connections.
Write Voice Setting Command	1.1	The Write Voice Setting command will write the values for the Voice Setting configuration parameter, which controls all the various settings for the voice connections.

Table 3.9: Synchronous connections

Petitioner Kiosoft Exhibit 1011 Part 2
Page 88

3.10 CONNECTION STATE

The connection state group of commands and events allows the configuration of a link, especially for low power operation.

Name	Vers.	Summary description
Mode Change Event	1.1	The Mode Change event is used to indicate that the current mode has changed.
Max Slots Change Event	1.1	The Max Slots Change event it used to indicate a change in the max slots by the LM.
Hold Mode Command	1.1	The Hold Mode command is used to initiate Hold Mode.
Sniff Mode Command	1.1	The Sniff Mode command is used to alter the behavior of the LM and have the LM place the local or remote device into the sniff mode.
Exit Sniff Mode Command	1.1	The Exit Sniff Mode command is used to end the sniff mode for a connection handle which is currently in sniff mode.
Park State Command	1.1	The Park State command is used to alter the behavior of the LM and have the LM place the local or remote device into the park state.
Exit Park State Command	1.1	The Exit Park State command is used to switch the Bluetooth device from park state back to active mode.
Read Link Policy Settings Command	1.1	The Read Link Policy Settings command will read the Link Policy configuration parameter for the specified Connection Handle. The Link Policy settings allow the Host to specify which Link Modes the LM can use for the specified Connection Handle.
Write Link Policy Settings Command	1.1	The Write Link Policy Settings command will write the Link Policy configuration parameter for the specified Connection Handle. The Link Policy settings allow the Host to specify which Link Modes the LM can use for the specified Connection Handle.
Read Default Link Policy Settings Command	1.2	The Read Default Link Policy Settings command will read the Default Link Policy configuration parameter for all new connections.
Write Default Link Policy Settings Command	1.2	The Write Default Link Policy Settings command will write the Default Link Policy configuration parameter for all new connections.

Table 3.10: Connection state

3.11 PICONET STRUCTURE

The piconet structure group of commands and events allows the discovery and reconfiguration a piconet.

Name	Vers.	Summary description
Role Discovery Command	1.1	The Role Discovery command is used for a Bluetooth device to determine which role the device is performing for a particular Connection Handle.
Switch Role Command	1.1	The Switch Role command is used to switch master and slave roles of the devices on either side of a connection.
Role Change Event	1.1	The Role Change event is used to indicate that the current Bluetooth role related to the particular connection has been changed.

Table 3.11: Piconet structure

3.12 QUALITY OF SERVICE

The quality of service group of commands and events allows the configuration of links to allow for quality of service parameters to be specified.

Name	Vers.	Summary description
Flow Specification Command	1.2	The Flow Specification command is used to specify the flow parameters for the traffic carried over the ACL connection identified by the Connection Handle.
Flow Specification Complete Event	1.2	The Flow Specification Complete event is used to inform the Host about the Quality of Service for the ACL connection the Controller is able to support.
QoS Setup Command	1.1	The QoS Setup command is used to specify Quality of Service parameters for a connection handle.
QoS Setup Complete Event	1.1	The QoS Setup Complete event is used to indicate that QoS is setup.
QoS Violation Event	1.1	The QoS Violation event is used to indicate the Link Manager is unable to provide the current QoS requirement for the Connection Handle.
Flush Command	1.1	The Flush command is used to discard all data that is currently pending for transmission in the Controller for the specified connection handle.
Flush Occurred Event	1.1	The Flush Occurred event is used to indicate that, for the specified Connection Handle, the data to be transmitted has been discarded.
Read Automatic Flush Timeout Command	1.1	The Read Automatic Flush Timeout will read the value for the Flush Timeout configuration parameter for the specified connection handle. The Flush Timeout parameter is only used for ACL connections.
Write Automatic Flush Timeout Command	1.1	The Write Automatic Flush Timeout will write the value for the Flush Timeout configuration parameter for the specified connection handle. The Flush Timeout parameter is only used for ACL connections.
Read Failed Contact Counter Command	1.1	The Read Failed Contact Counter will read the value for the Failed Contact Counter configuration parameter for a particular connection to another device.
Reset Failed Contact Counter Command	1.1	The Reset Failed Contact Counter will reset the value for the Failed Contact Counter configuration parameter for a particular connection to another device.
Read Num Broadcast Retransmissions Command	1.1	The Read Num Broadcast Retransmissions command will read the parameter value for the Number of Broadcast Retransmissions for the device.
Write Num Broadcast Retransmissions Command	1.1	The Write Num Broadcast Retransmissions command will write the parameter value for the Number of Broadcast Retransmissions for the device.

Table 3.12: Quality of service

3.13 PHYSICAL LINKS

The physical links commands and events allows configuration of the physical link.

Name	Vers.	Summary description
Read Link Supervision Timeout Command	1.1	The Read Link Supervision Timeout command will read the value for the Link Supervision Timeout configuration parameter for the device. This parameter is used by the device to determine link loss.
Write Link Supervision Timeout Command	1.1	The Write Link Supervision Timeout command will write the value for the Link Supervision Timeout configuration parameter for the device. This parameter is used by the device to determine link loss.
Read AFH Channel Assessment Mode Command	1.2	The Read AFH Channel Assessment Mode command will read the value for the AFH Channel Classification Mode parameter. This value is used to enable or disable the Controller's channel assessment scheme.
Write AFH Channel Assessment Mode Command	1.2	The Write AFH Channel Assessment Mode command will write the value for the Channel Classification Mode configuration parameter. This value is used to enable or disable the Controller's channel assessment scheme.
Set AFH Host Channel Classification Command	1.2	The Set AFH Host Channel Classification command allows the Host to specify a channel classification based on its "local information".
Change Connection Packet Type Command	1.1	The Change Connection Packet Type command is used to change which packet types can be used for a connection that is currently established.
Connection Packet Type Changed Event	1.1	The Connection Packet Type Changed event is used to indicate the completion of the process of the Link Manager changing the packet type mask used for the specified Connection Handle.

Table 3.13: Physical links

3.14 HOST FLOW CONTROL

The Host flow control group of commands and events allows flow control to be used towards the Host.

Name	Vers.	Summary description
Host Buffer Size Command	1.1	The Host Buffer Size command is used by the Host to notify the Controller about its buffer sizes for ACL and synchronous data. The Controller will segment the data to be transmitted from the Controller to the Host, so that data contained in HCI Data Packets will not exceed these sizes.
Set Event Mask Command	1.1	The Set Event Mask command is used to control which events are generated by the HCI for the Host.
Set Event Filter Command	1.1	The Set Event Filter command is used by the Host to specify different event filters. The Host may issue this command multiple times to request various conditions for the same type of event filter and for different types of event filters.
Set Controller To Host Flow Control Command	1.1	The Set Controller To Host Flow Control command is used by the Host to turn flow control on or off in the direction from the Controller to the Host.
Host Number Of Completed Packets Command	1.1	The Host Number Of Completed Packets command is used by the Host to indicate to the Controller when the Host is ready to receive more HCI packets for any connection handle.
Data Buffer Overflow Event	1.1	The Data Buffer Overflow event is used to indicate that the Controller's data buffers have overflowed, because the Host has sent more packets than allowed.
Read Synchronous Flow Control Enable Command	1.1	The Read Synchronous Flow Control Enable command provides the ability to read the Synchronous Flow Control Enable setting. By using this setting, the Host can decide if the Controller will send Number Of Completed Packets events for Synchronous Connection Handles.
Write Synchronous Flow Control Enable Command	1.1	The Write Synchronous Flow Control Enable command provides the ability to write the Synchronous Flow Control Enable setting. By using this setting, the Host can decide if the Controller will send Number Of Completed Packets events for Synchronous Connection Handles.

Table 3.14: Controller flow control.

3.15 LINK INFORMATION

The link information group of commands and events allows information about a link to be read.

Name	Vers.	Summary description
Read LMP Handle Command	1.2	The Read LMP Handle command will read the current LMP Handle associated with the Connection Handle.
Read Transmit Power Level Command	1.1	The Read Transmit Power Level command will read the values for the Transmit Power Level parameter for the specified Connection Handle.
Read Link Quality Command	1.1	The Read Link Quality command will read the value for the Link Quality for the specified Connection Handle.
Read RSSI Command	1.1	The Read RSSI command will read the value for the Received Signal Strength Indication (RSSI) for a connection handle to another Bluetooth device.
Read Clock Offset Command	1.1	The Read Clock Offset command allows the Host to read the clock offset of remote devices.
Read Clock Offset Complete Event	1.1	The Read Clock Offset Complete event is used to indicate the completion of the process of the LM obtaining the Clock offset information.
Read Clock Command	1.2	The Read Clock command will read an estimate of a piconet or the local Bluetooth Clock.
Read AFH Channel Map Command	1.2	The Read AFH Channel Map command will read the current state of the channel map for a connection.

Table 3.15: Link information

3.16 AUTHENTICATION AND ENCRYPTION

The authentication and encryption group of commands and events allows authentication of a remote device and then encryption of the link to one or more remote devices.

Name	Vers.	Summary description
Read Authentication Enable Command	1.1	The Read Authentication Enable command will read the value for the Authentication Enable parameter, which controls whether the Bluetooth device will require authentication for each connection with other Bluetooth devices.
Write Authentication Enable Command	1,1	The Write Authentication Enable command will write the value for the Authentication Enable parameter, which controls whether the Bluetooth device will require authentication for each connection with other Bluetooth devices.
Read Encryption Mode Command	1.1	The Read Encryption Mode command will read the value for the Encryption Mode parameter, which controls whether the Bluetooth device will require encryption for each connection with other Bluetooth devices.
Write Encryption Mode Command	1.1	The Write Encryption Mode command will write the value for the Encryption Mode parameter, which controls whether the Bluetooth device will require encryption for each connection with other Bluetooth devices.
Link Key Request Event	1.1	The Link Key Request event is used to indicate that a Link Key is required for the connection with the device specified in BD_ADDR.
Link Key Request Reply Command	1.1	The Link Key Request Reply command is used to reply to a Link Key Request event from the Controller, and specifies the Link Key stored on the Host to be used as the link key for the connection with the other Bluetooth device specified by BD_ADDR.
Link Key Request Negative Reply Command	1.1	The Link Key Request Negative Reply command is used to reply to a Link Key Request event from the Controller if the Host does not have a stored Link Key for the connection with the other Bluetooth Device specified by BD_ADDR.
PIN Code Request Event	1.1	The PIN Code Request event is used to indicate that a PIN code is required to create a new link key for a connection.
PIN Code Request Reply Command	1.1	The PIN Code Request Reply command is used to reply to a PIN Code Request event from the Controller and specifies the PIN code to use for a connection.

Table 3.16: Authentication and encryption

Name	Vers.	Summary description
PIN Code Request Negative Reply Command	1.1	The PIN Code Request Negative Reply command is used to reply to a PIN Code Request event from the Controller when the Host cannot specify a PIN code to use for a connection.
Link Key Notification Event	1.1	The Link Key Notification event is used to indicate to the Host that a new Link Key has been created for the connection with the device specified in BD_ADDR.
Authentication Requested Command	1.1	The Authentication Requested command is used to establish authentication between the two devices associated with the specified Connection Handle.
Authentication Complete Event	1.1	The Authentication Complete event occurs when authentication has been completed for the specified connection.
Set Connection Encryption Command	1.1	The Set Connection Encryption command is used to enable and disable the link level encryption.
Encryption Change Event	1.1	The Encryption Change event is used to indicate that the change in the encryption has been completed for the Connection Handle specified by the Connection Handle event parameter.
Change Connection Link Key Command	1.1	The Change Connection Link Key command is used to force both devices of a connection associated to the connection handle, to generate a new link key.
Change Connection Link Key Complete Event	1.1	The Change Connection Link Key Complete event is used to indicate that the change in the Link Key for the Connection Handle specified by the Connection Handle event parameter had been completed.
Master Link Key Command	1.1	The Master Link Key command is used to force both devices of a connection associated to the connection handle to use the temporary link key of the Master device or the regular link keys.
Master Link Key Complete Event	1.1	The Master Link Key Complete event is used to indicate that the change in the temporary Link Key or in the semi-permanent link keys on the Bluetooth master side has been completed.
Read PIN Type Command	1.1	The Read PIN Type command is used for the Host to read the value that is specified to indicate whether the Host supports variable PIN or only fixed PINs.
Write PIN Type Command	1.1	The Write PIN Type command is used for the Host to specify whether the Host supports variable PIN or only fixed PINs.
Read Stored Link Key Command	1.1	The Read Stored Link Key command provides the ability to read one or more link keys stored in the Controller.

Table 3.16: Authentication and encryption

Name	Vers.	Summary description
Return Link Keys Event	1.1	The Return Link Keys event is used to return stored link keys after a Read Stored Link Key command is used.
Write Stored Link Key Command	1.1	The Write Stored Link Key command provides the ability to write one or more link keys to be stored in the Controller.
Delete Stored Link Key Command	1.1	The Delete Stored Link Key command provides the ability to remove one or more of the link keys stored in the Controller.
Create New Unit Key Command	1.1	The Create New Unit Key command is used to create a new unit key.

Table 3.16: Authentication and encryption

3.17 TESTING

The testing group of commands and events allows a device to be placed into a special testing mode to allow for testing to be performed.

Name	Vers.	Summary description
Read Loopback Mode Command	1.1	The Read Loopback Mode will read the value for the setting of the Controllers Loopback Mode. The setting of the Loopback Mode will determine the path of information.
Write Loopback Mode Command	1.1	The Write Loopback Mode will write the value for the setting of the Controllers Loopback Mode. The setting of the Loopback Mode will determine the path of information.
Loopback Command Event	1.1	The Loopback Command event is used to loop back all commands that the Host sends to the Controller with some exceptions.
Enable Device Under Test Mode Command	1.1	The Enable Device Under Test Mode command will allow the local Bluetooth module to enter test mode via LMP test commands. The Host issues this command when it wants the local device to be the DUT for the Testing scenarios as described in the Bluetooth Test Mode document.

Table 3.17: Testing

3.18 ALPHABETICAL LIST OF COMMANDS AND EVENTS

Commands/Events	Group
Accept Connection Request Command	Connection Setup
Authentication Complete Event	Authentication and Encryption
Authentication Requested Command	Authentication and Encryption
Change Connection Link Key Command	Authentication and Encryption
Change Connection Link Key Complete Event	Authentication and Encryption
Change Connection Packet Type Command	Physical Links
Command Complete Event	Generic Events
Command Status Event	Generic Events
Connection Complete Event	Connection Setup
Connection Packet Type Changed Event	Physical Links
Connection Request Event	Connection Setup
Create Connection Cancel Command	Connection Setup
Create Connection Command	Connection Setup
Create New Unit Key Command	Authentication and Encryption
Data Buffer Overflow Event	Host Flow Control
Delete Stored Link Key Command	Authentication and Encryption
Disconnect Command	Connection Setup
Disconnection Complete Event	Connection Setup
Enable Device Under Test Mode Command	Testing
Encryption Change Event	Authentication and Encryption
Exit Park State Command	Connection State
Exit Periodic Inquiry Mode Command	Device Discovery
Exit Sniff Mode Command	Connection State
Flow Specification Command	Quality of Service
Flow Specification Complete Event	Quality of Service
Flush Command	Quality of Service
Flush Occurred Event	Quality of Service
Hardware Error Eventt	Generic Events
Hold Mode Command	Connection State
Host Buffer Size Command	Host Flow Control

Table 3.18: Alphabetical list of commands and events.

Commands/Events	Group
Host Number Of Completed Packets Command	Host Flow Control
Inquiry Cancel Command	Device Discovery
Inquiry Command	Device Discovery
Inquiry Complete Event	Device Discovery
Inquiry Result Event	Device Discovery
Inquiry Result with RSSI Event	Device Discovery
Link Key Notification Event	Authentication and Encryption
Link Key Request Event	Authentication and Encryption
Link Key Request Negative Reply Command	Authentication and Encryption
Link Key Request Reply Command	Authentication and Encryption
Loopback Command Event	Testing
Master Link Key Command	Authentication and Encryption
Master Link Key Complete Event	Authentication and Encryption
Max Slots Change Event	Connection State
Mode Change Event	Connection State
Number Of Completed Packets Event	Controller Flow Control
Page Scan Repetition Mode Change Event	Connection Setup
Park State Command	Connection State
Periodic Inquiry Mode Command	Device Discovery
PIN Code Request Event	Authentication and Encryption
PIN Code Request Negative Reply Command	Authentication and Encryption
PIN Code Request Reply Command	Authentication and Encryption
QoS Setup Command	Quality of Service
QoS Setup Complete Event	Quality of Service
QoS Violation Event	Quality of Service
Read AFH Channel Assessment Mode Command	Physical Links
Read AFH Channel Map Command	Link Information
Read Authentication Enable Command	Authentication and Encryption
Read Automatic Flush Timeout Command	Quality of Service
Read BD_ADDR Command	Controller Information
Read Buffer Size Command	Controller Flow Control
Read Class of Device Command	Controller Information

Table 3.18: Alphabetical list of commands and events.

Commands/Events	Group
Read Clock Command	Link Information
Read Clock Offset Command	Link Information
Read Clock Offset Complete Event	Link Information
Read Connection Accept Timeout Command	Connection Setup
Read Current IAC LAP Command	Controller Information
Read Default Link Policy Settings Command	Connection State
Read Encryption Mode Command	Authentication and Encryption
Read Failed Contact Counter Command	Quality of Service
Read Hold Mode Activity Command	Connection State
Read Inquiry Mode Command	Device Discovery
Read Inquiry Scan Activity Command	Device Discovery
Read Inquiry Scan Type Command	Device Discovery
Read Link Policy Settings Command	Connection State
Read Link Quality Command	Link Information
Read Link Supervision Timeout Command	Physical Links
Read LMP Handle Command	Link Information
Read Local Extended Features Command	Controller Information
Read Local Name Command	Controller Configuration
Read Local Supported Commands Command	Controller Information
Read Local Supported Features Command	Controller Information
Read Local Version Information Command	Controller Information
Read Loopback Mode Command	Testing
Read Num Broadcast Retransmissions Command	Quality of Service
Read Number Of Supported IAC Command	Controller Information
Read Page Scan Activity Command	Connection Setup
Read Page Scan Period Mode Command	Connection Setup
Read Page Scan Type Command	Connection Setup
Read Page Timeout Command	Connection Setup
Read PIN Type Command	Authentication and Encryption
Read Remote Extended Features Command	Remote Information
Read Remote Extended Features Complete Event	Remote Information
Read Remote Supported Features Command	Remote Information

Table 3.18: Alphabetical list of commands and events.

Commands/Events	Group
Read Remote Supported Features Complete Event	Remote Information
Read Remote Version Information Command	Remote Information
Read Remote Version Information Complete Event	Remote Information
Read RSSI Command	Link Information
Read Scan Enable Command	Controller Information
Read Stored Link Key Command	Authentication and Encryption
Read Synchronous Flow Control Enable Command	Host Flow Control
Read Transmit Power Level Command	Link Information
Read Voice Setting Command	Synchronous Connections
Reject Connection Request Command	Connection Setup
Remote Name Request Cancel Command	Remote Information
Remote Name Request Command	Remote Information
Remote Name Request Complete Event	Remote Information
Reset Command	Device Setup
Reset Failed Contact Counter Command	Quality of Service
Return Link Keys Event	Authentication and Encryption
Role Change Event	Piconet Structure
Role Discovery Command	Piconet Structure
Set AFH Host Channel Classification Command	Physical Links
Set Connection Encryption Command	Authentication and Encryption
Set Controller To Host Flow Control Command	Host Flow Control
Set Event Filter Command	Host Flow Control
Set Event Mask Command	Host Flow Control
Setup Synchronous Connection Command	Synchronous Connections
Sniff Mode Command	Connection State
Switch Role Command	Piconet Structure
Synchronous Connection Changed event	Synchronous Connections
Synchronous Connection Complete Event	Synchronous Connections
Write AFH Channel Assessment Mode Command	Physical Links
Write Authentication Enable Command	Authentication and Encryption
Write Automatic Flush Timeout Command	Quality of Service
Write Class of Device Command	Controller Information

Table 3.18: Alphabetical list of commands and events.

Commands/Events	Group
Write Connection Accept Timeout Command	Connection Setup
Write Current IAC LAP Command	Controller Information
Write Default Link Policy Settings Command	Connection State
Write Encryption Mode Command	Authentication and Encryption
Write Hold Mode Activity Command	Connection State
Write Inquiry Mode Command	Device Discovery
Write Inquiry Scan Activity Command	Device Discovery
Write Inquiry Scan Type Command	Device Discovery
Write Link Policy Settings Command	Connection State
Write Link Supervision Timeout Command	Physical Links
Write Local Name Command	Controller Information
Write Loopback Mode Command	Testing
Write Num Broadcast Retransmissions Command	Quality of Service
Write Page Scan Activity Command	Connection Setup
Write Page Scan Period Mode Command	Connection Setup
Write Page Scan Type Command	Connection Setup
Write Page Timeout Command	Connection Setup
Write PIN Type Command	Authentication and Encryption
Write Scan Enable Command	Controller Information
Write Stored Link Key Command	Authentication and Encryption
Write Synchronous Flow Control Enable Command	Host Flow Control
Write Voice Setting Command	Synchronous Connections

Table 3.18: Alphabetical list of commands and events.

4 HCI FLOW CONTROL

Flow control shall be used in the direction from the Host to the Controller to avoid overflowing the Controller data buffers with ACL data destined for a remote device (using a connection handle) that is not responding. The Host manages the data buffers of the Controller.

4.1 HOST TO CONTROLLER DATA FLOW CONTROL

On initialization, the Host shall issue the Read Buffer Size command. Two of the return parameters of this command determine the maximum size of HCI ACL and synchronous Data Packets (excluding header) sent from the Host to the Controller. There are also two additional return parameters that specify the total number of HCI ACL and synchronous Data Packets that the Controller may have waiting for transmission in its buffers.

When there is at least one connection to another device, or when in local loopback mode, the Controller shall use the Number Of Completed Packets event to control the flow of data from the Host. This event contains a list of connection handles and a corresponding number of HCI Data Packets that have been completed (transmitted, flushed, or looped back to the Host) since the previous time the event was returned (or since the connection was established, if the event has not been returned before for a particular connection handle).

Based on the information returned in this event, and the return parameters of the Read Buffer Size command that specify the total number of HCI ACL and synchronous Data Packets that can be stored in the Controller, the Host decides for which Connection Handles the following HCI Data Packets should be sent.

Every time it has sent an HCI Data Packet, the Host shall assume that the free buffer space for the corresponding link type (ACL, SCO or eSCO) in the Controller has decreased by one HCI Data Packet.

Each Number Of Completed Packets event received by the Host provides information about how many HCI Data Packets have been completed (transmitted or flushed) for each Connection Handle since the previous Number Of Completed Packets event was sent to the Host. It can then calculate the actual current buffer usage.

When the Controller has completed one or more HCI Data Packet(s) it shall send a Number Of Completed Packets event to the Host, until it finally reports that all the pending HCI Data Packets have been completed. The frequency at which this event is sent is manufacturer specific.

Note: The Number Of Completed Packets events will not report on synchronous connection handles if Synchronous Flow Control is disabled. (See Read/

Write Synchronous Flow Control Enable, Section 7.3.38 on page 486 and Section 7.3.39 on page 487)

For each individual Connection Handle, the data must be sent to the Controller in HCI Data Packets in the order in which it was created in the Host. The Controller shall also transmit data on the air that is received from the Host for a given Connection Handle in the same order as it is received from the Host.

Data that is received on the air from another device shall, for the corresponding Connection Handle, be sent in HCI Data Packets to the Host in the same order as it is received. This means the scheduling shall be decided separately for each Connection Handle basis. For each individual Connection Handle, the order of the data shall not be changed from the order in which the data has been created.

4.2 CONTROLLER TO HOST DATA FLOW CONTROL

In some implementations, flow control may also be necessary in the direction from the Controller to the Host. The Set Host Controller To Host Flow Control command can be used to turn flow control on or off in that direction.

On initialization, the Host uses the Host Buffer Size command to notify the Controller about the maximum size of HCI ACL and synchronous Data Packets sent from the Controller to the Host. The command also contains two additional command parameters to notify the Controller about the total number of ACL and synchronous Data Packets that can be stored in the data buffers of the Host.

The Host uses the Host Number Of Completed Packets command in exactly the same way as the Controller uses the Number Of Completed Packets event as was previously described in this section.

The Host Number Of Completed Packets command is a special command for which no command flow control is used, and which can be sent anytime there is a connection or when in local loopback mode. The command also has no event after the command has completed. This makes it possible for the flow control to work in exactly the same way in both directions, and the flow of normal commands will not be disturbed.

4.3 DISCONNECTION BEHAVIOUR

When the Host receives a Disconnection Complete event, the Host shall assume that all unacknowledged HCI Data Packets that have been sent to the Controller for the returned Connection Handle have been flushed, and that the corresponding data buffers have been freed. The Controller does not have to notify the Host about this in a Number Of Completed Packets event.

If flow control is also enabled in the direction from the Controller to the Host, the Controller may, after it has sent a Disconnection Complete event, assume that the Host will flush its data buffers for the sent Connection Handle when it receives the Disconnection Complete event. The Host does not have to notify the Controller about this in a Host Number Of Completed Packets command.

4.4 COMMAND FLOW CONTROL

On initial power-on, and after a reset, the Host shall send a maximum of one outstanding HCI Command Packet until a Command Complete or Command Status event has been received.

The Command Complete and Command Status events contain a parameter called Num HCI Command Packets, which indicates the number of HCI Command Packets the Host is currently allowed to send to the Controller. The Controller may buffer one or more HCI command packets, but the Controller must start performing the commands in the order in which they are received. The Controller can start performing a command before it completes previous commands. Therefore, the commands do not always complete in the order they are started.

To indicate to the Host that the Controller is ready to receive HCI command packets, the Controller may generate a Command Complete event with the Command Opcode 0x0000, and the Num HCI Command Packets event parameter is set to 1 or more. Command Opcode 0x0000 is a NOP (No OPeration), and can be used to change the number of outstanding HCI command packets that the Host can send. The Controller may generate a Command Complete event with the Num HCI Command Packets event parameter set to zero to inform Host it must stop sending commands.

For most commands, a Command Complete event shall be sent to the Host when the Controller completes the command. Some commands are executed in the background and do not return a Command Complete event when they have been completed. Instead, the Controller shall send a Command Status event back to the Host when it has begun to execute the command. When the actions associated with the command have finished, an event that is associated with the command shall be sent by the Controller to the Host.

If the command does not begin to execute (for example, if there was a parameter error or the command is currently not allowed), the Command Status event shall be returned with the appropriate error code in the Status parameter, and the event associated with the sent command shall not be returned.

4.5 COMMAND ERROR HANDLING

If an error occurs for a command for which a Command Complete event is returned, the Return Parameters field may not contain all the return parameters specified for the command. The Status parameter, which explains the error reason and which is the first return parameter, shall always be returned. If there is a Connection Handle parameter or a BD_ADDR parameter right after the Status parameter, this parameter shall also be returned so that the Host can identify to which instance of a command the Command Complete event belongs. In this case, the Connection Handle or BD_ADDR parameter shall have exactly the same value as that in the corresponding command parameter. It is implementation specific whether more parameters will be returned in case of an error.

The above also applies to commands that have associated command specific completion events with a status parameter in their completion event, with two exceptions. The exceptions are the Connection Complete and the Synchronous Connection Complete events. On failure, for these two events only, the second parameter, Connection_Handle, is not valid and the third parameter, BD_ADDR, is valid for identification purposes. The validity of other parameters is likewise implementation specific for failed commands in this group.

Note: The BD_ADDR return parameter of the command Read BD_ADDR is not used to identify to which instance of the Read BD ADDR command the Command Complete event belongs. It is optional for the Controller to return this parameter in case of an error.

5 HCI DATA FORMATS

5.1 INTRODUCTION

The HCI provides a uniform command method of accessing the Bluetooth capabilities. The HCI Link commands provide the Host with the ability to control the link layer connections to other Bluetooth devices. These commands typically involve the Link Manager (LM) to exchange LMP commands with remote Bluetooth devices. For details see “Link Manager Protocol” on page 189 [Part C].

The HCI Policy commands are used to affect the behavior of the local and remote LM. These Policy commands provide the Host with methods of influencing how the LM manages the piconet. The Controller & Baseband, Informational, and Status commands provide the Host access to various registers in the Controller.

HCI commands may take different amounts of time to be completed. Therefore, the results of commands will be reported back to the Host in the form of an event. For example, for most HCI commands the Controller will generate the Command Complete event when a command is completed. This event contains the return parameters for the completed HCI command. For enabling the Host to detect errors on the HCI-Transport Layer, there needs to be a timeout between the transmission of the Host’s command and the reception of the Controller’s response (e.g. a Command Complete or Command Status event). Since the maximum response timeout is strongly dependent on the HCI-Transport Layer used, it is recommended to use a default value of one second for this timer. This amount of time is also dependent on the number of commands unprocessed in the command queue.

5.2 DATA AND PARAMETER FORMATS

- All values are in Binary and Hexadecimal Little Endian formats unless otherwise noted
- In addition, all parameters which can have negative values must use 2’s complement when specifying values
- Arrayed parameters are specified using the following notation: ParameterA[i]. If more than one set of arrayed parameters are specified (e.g. ParameterA[i], ParameterB[i]), then the order of the parameters are as follows: ParameterA[0], ParameterB[0], ParameterA[1], ParameterB[1], ParameterA[2], ParameterB[2], … ParameterA[n], ParameterB[n]
- Unless noted otherwise, all parameter values are sent and received in Little Endian format (i.e. for multi-octet parameters the rightmost (Least Signification Octet) is transmitted first)
- All command and event parameters that are not-arrayed and all elements in an arrayed parameter have fixed sizes (an integer number of octets). The parameters and the size of each not arrayed parameter (or of each element in an arrayed parameter) contained in a command or an event is specified

for each command or event. The number of elements in an arrayed parameter is not fixed.

- Where bit strings are specified, the low order bit is the right hand bit, e.g. 0 is the low order bit in ‘10’.

5.3 CONNECTION HANDLES

Connection Handles are used to identify logical channels between the Host and Controller. Connection Handles are assigned by the Controller when a new logical link is created, using the Connection Complete or Synchronous Connection Complete events. Broadcast Connection Handles are handled differently, and are described below.

The first time the Host sends an HCI Data Packet with Broadcast_Flag set to 01b (active broadcast) or 10b (piconet broadcast) after a power-on or a reset, the value of the Connection Handle parameter must be a value which is not currently assigned by the Host Controller. The Host must use different connection handles for active broadcast and piconet broadcast.

The Host Controller must then continue to use the same connection handles for each type of broadcast until a reset is made. Note: The Host Controller must not send a Connection Complete event containing a new Connection_Handle that it knows is used for broadcast.

Note: In some situations, it may happen that the Host Controller sends a Connection Complete event before having interpreted a Broadcast packet received from the Host, and that the Connection_Handles of both Connection Complete event and HCI Data packet are the same. This conflict has to be avoided as follows:

If a Connection Complete event is received containing one of the connection handles used for broadcast, the Host has to wait before sending any packets for the new connection until it receives a Number Of Completed Packets event indicating that there are no pending broadcast packets belonging to the connection handle. In addition, the Host must change the Connection_Handle used for the corresponding type of broadcast to a Connection_Handle which is currently not assigned by the Host Controller.

This Connection_Handle must then be used for all the following broadcasts of that type until a reset is performed or the same conflict situation happens again. However, this will occur very rarely.

The Host Controller must, in the above conflict case, be able to distinguish between the Broadcast message sent by the Host and the new connection made (this could be even a new synchronous link) even though the connection handles are the same.

For an HCI Data Packet sent from the Host Controller to the Host where the Broadcast_Flag is 01 or 10, the Connection_Handle parameter should contain

Petitioner Kiosoft Exhibit 1011 Part 2
Page 108

the connection handle for the ACL connection to the master that sent the broadcast.

Note: Connection handles used for Broadcast do not identify an ACL point-to-point connection, so they must not be used in any command having a Connection_Handle parameter and they will not be returned in any event having a Connection_Handle parameter except the Number Of Completed Packets event.

5.4 EXCHANGE OF HCI-SPECIFIC INFORMATION

The Host Controller Transport Layer provides transparent exchange of HCI specific information. These transporting mechanisms provide the ability for the Host to send HCI commands, ACL data and synchronous data to the Controller. These transport mechanisms also provide the ability for the Host to receive HCI events, ACL data and synchronous data from the Controller. Since the Host Controller Transport Layer provides transparent exchange of HCI-specific information, the HCI specification specifies the format of the commands, events, and data exchange between the Host and the Controller. The next sections specify the HCI packet formats.

5.4.1 HCI Command Packet

The HCI Command Packet is used to send commands to the Controller from the Host. The format of the HCI Command Packet is shown in Figure 5.1, and the definition of each field is explained below.

The Controller must be able to accept HCI Command Packets with up to 255 bytes of data excluding the HCI Command Packet header.

Each command is assigned a 2 byte Opcode used to uniquely identify different types of commands. The Opcode parameter is divided into two fields, called the OpCode Group Field (OGF) and OpCode Command Field (OCF). The OGF occupies the upper 6 bits of the Opcode, while the OCF occupies the remaining 10 bits. The OGF of 0x3F is reserved for vendor-specific debug commands. The OGF of 0x3E is reserved for Bluetooth Logo Testing. The organization of the Opcodes allows additional information to be inferred without fully decoding the entire Opcode.

Note: the OGF composed of all ‘ones’ has been reserved for vendor-specific debug commands. These commands are vendor-specific and are used during manufacturing, for a possible method for updating firmware, and for debugging.

Note: the OGF composed of all ‘zeros’ and an OCF or all ‘zeros’ is the NOP command. This command Opcode may be used in Command Flow Control. (See Section 4.4 on page 347)

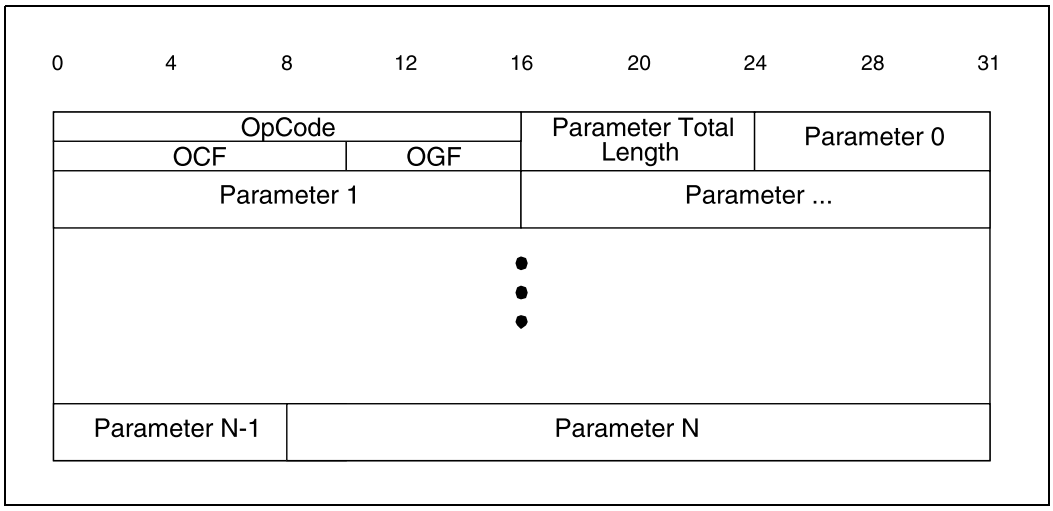


Figure 5.1: HCI Command Packet

Op_Code: *Size: 2 Octets*

Value	Parameter Description
0xXXXX	OGFRange (6 bits): 0x00-0x3F (0x3E reserved for Bluetooth logo testing and 0x3F reserved for vendor-specific debug commands) OCF Range (10 bits): 0x0000-0x03FF

Parameter_Total_Length: *Size: 1 Octet*

Value	Parameter Description
0xXX	Lengths of all of the parameters contained in this packet measured in octets. (N.B.: total length of parameters, not number of parameters)

Parameter 0 - N: *Size: Parameter Total Length*

Value	Parameter Description
0xXX	Each command has a specific number of parameters associated with it. These parameters and the size of each of the parameters are defined for each command. Each parameter is an integer number of octets in size.

5.4.2 HCI ACL Data Packets

HCI ACL Data Packets are used to exchange data between the Host and Controller. The format of the HCI ACL Data Packet is shown in Figure 5.2. The definition for each of the fields in the data packets is explained below.

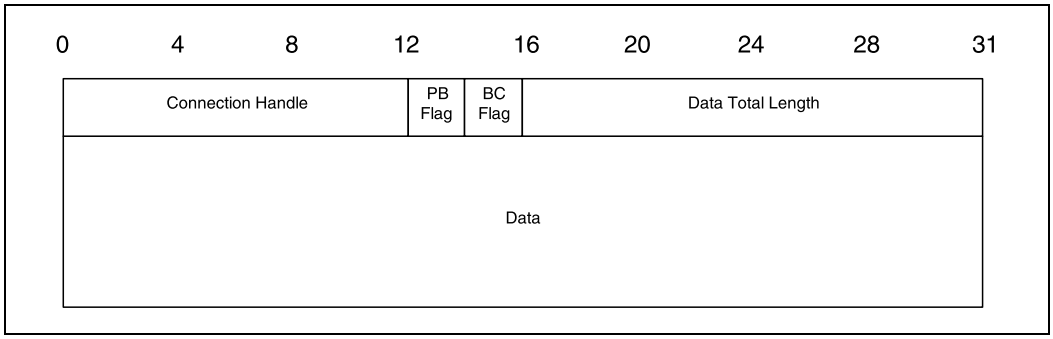


Figure 5.2: HCI ACL Data Packet

Connection_Handle: *Size: 12 Bits*

Value	Parameter Description
0xXXX	Connection Handle to be used for transmitting a data packet or segment. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Flags: *Size: 2 Bits*

The Flag Bits consist of the Packet_Boundary_Flag and Broadcast_Flag. The Packet_Boundary_Flag is located in bit 4 and bit 5, and the Broadcast_Flag is located in bit 6 and 7 in the second octet of the HCI ACL Data packet.

Packet_Boundary_Flag: *Size: 2 Bits*

Value	Parameter Description
00	Reserved for future use
01	Continuing fragment packet of Higher Layer Message
10	First packet of Higher Layer Message (i.e. start of an L2CAP packet)
11	Reserved for future use

Broadcast_Flag (in packet from Host to Controller): *Size: 2 Bits*

Value	Parameter Description
00	No broadcast. Only point-to-point.
01	Active Slave Broadcast: packet is sent to all active slaves (i.e. packet is usually not sent during park beacon slots), and it may be received by slaves in sniff mode or park state.
10	Parked Slave Broadcast: packet is sent to all slaves and all slaves in park state (i.e. packet is sent during park beacon slots if there are parked slaves), and it may be received by slaves in sniff mode.
11	Reserved for future use.

Broadcast_Flag (in packet from Controller to Host): *Size: 2 Bits*

Value	Parameter Description
00	Point-to-point
01	Packet received as a slave not in park state (either Active Slave Broadcast or Parked Slave Broadcast)
10	Packet received as a slave in park state (Parked Slave Broadcast)
11	Reserved for future use.

Note: active slave broadcast packets may be sent in park beacon slots.

Note: slaves in sniff mode may or may not receive a broadcast packet depending on whether they happen to be listening at sniff slots, when the packet is sent.

Data_Total_Length: *Size: 2 Octets*

Value	Parameter Description
0xXXXX	Length of data measured in octets.

5.4.3 HCI Synchronous Data Packets

HCI synchronous (SCO and eSCO) Data Packets are used to exchange synchronous data between the Host and Controller. The format of the synchronous Data Packet is shown in Figure 5.3. The definition for each of the fields in the data packets is explained below.

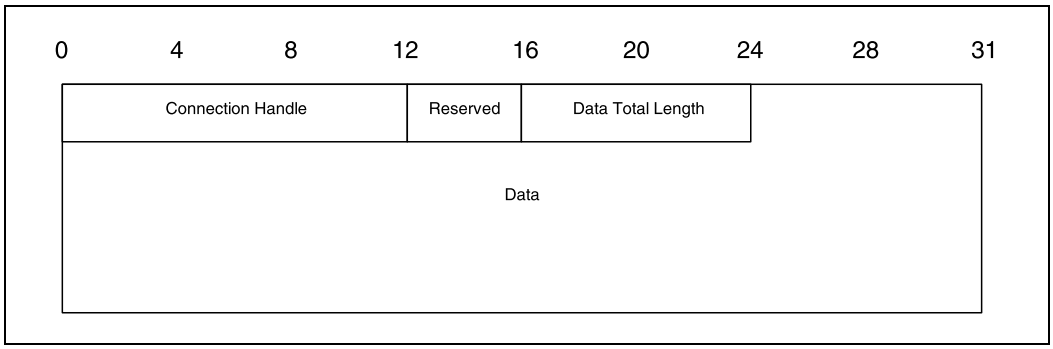


Figure 5.3: HCI Synchronous Data Packet

Connection_Handle: *Size: 12 Bits*

Value	Parameter Description
0xXXX	Connection handle to be used to for transmitting a synchronous data packet or segment. Range: 0x0000-0x0EFF (0x0F00- 0x0FFF Reserved for future use)

The Reserved Bits consist of four bits which are located from bit 4 to bit 7 in the second octet of the HCI Synchronous Data packet.

Reserved: *Size: 4 Bits*

Value	Parameter Description
XXXX	Reserved for future use.

Data_Total_Length: *Size: 1 Octet*

Value	Parameter Description
0xXX	Length of synchronous data measured in octets

5.4.4 HCI Event Packet

The HCI Event Packet is used by the Controller to notify the Host when events occur. The Host must be able to accept HCI Event Packets with up to 255 octets of data excluding the HCI Event Packet header. The format of the HCI Event Packet is shown in Figure 5.4, and the definition of each field is explained below.

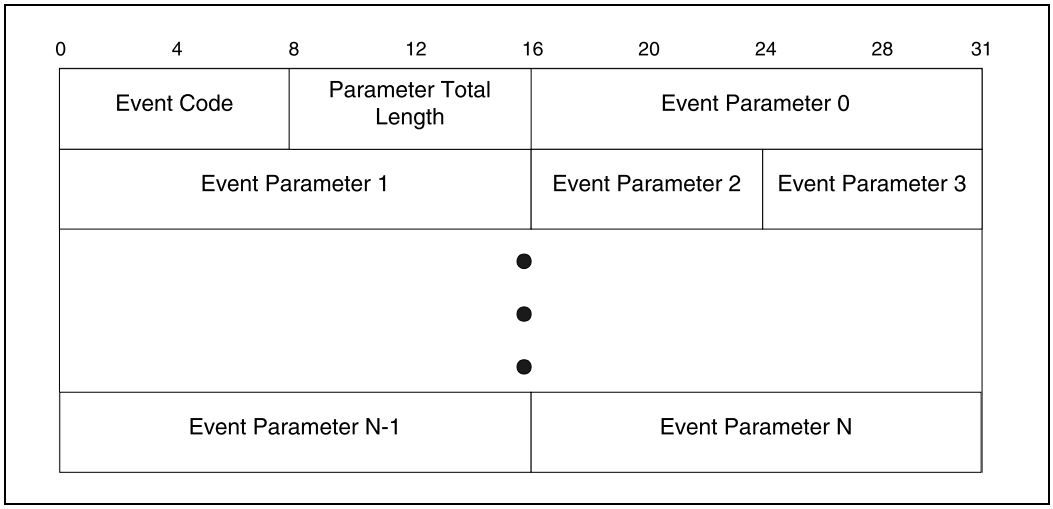


Figure 5.4: HCI Event Packet

Event_Code: *Size: 1 Octet*

Value	Parameter Description
0xXX	Each event is assigned a 1-Octet event code used to uniquely identify different types of events. Range: 0x00-0xFF (The event code 0xFF is reserved for the event code used for vendor-specific debug events. In addition, the event code 0xFE is also reserved for Bluetooth Logo Testing)

Parameter_Total_Length: *Size: 1 Octet*

Value	Parameter Description
0xXX	Length of all of the parameters contained in this packet, measured in octets

Event_Parameter 0 - N: *Size: Parameter Total Length*

Value	Parameter Description
0xXX	Each event has a specific number of parameters associated with it. These parameters and the size of each of the parameters are defined for each event. Each parameter is an integer number of octets in size.

6 HCI CONFIGURATION PARAMETERS

6.1 SCAN ENABLE

The Scan_Enable parameter controls whether or not the Bluetooth device will periodically scan for page attempts and/or inquiry requests from other Bluetooth devices. If Page_Scan is enabled, then the device will enter page scan mode based on the value of the Page_Scan_Interval and Page_Scan_Window parameters. If Inquiry_Scan is enabled, then the device will enter Inquiry Scan mode based on the value of the Inquiry_Scan_Interval and Inquiry_Scan_Window parameters.

Value	Parameter Description
0x00	No Scans enabled.
0x01	Inquiry Scan enabled. Page Scan always disabled.
0x02	Inquiry Scan disabled. Page Scan enabled.
0x03	Inquiry Scan enabled. Page Scan enabled.
0x04-0xFF	Reserved

6.2 INQUIRY SCAN INTERVAL

The Inquiry_Scan_Interval configuration parameter defines the amount of time between consecutive inquiry scans. This is defined as the time interval from when the Controller started its last inquiry scan until it begins the next inquiry scan.

Value	Parameter Description
N = 0xXXXX	Size: 2 Octets Range: 0x0012 to 0x1000; only even values are valid Default: 0x1000 Mandatory Range: 0x0012 to 0x1000 Time = N * 0.625 msec Time Range: 11.25 to 2560 msec Time Default: 2.56 sec

6.3 INQUIRY SCAN WINDOW

The Inquiry_Scan_Window configuration parameter defines the amount of time for the duration of the inquiry scan. The Inquiry_Scan_Window can only be less than or equal to the Inquiry_Scan_Interval.

Value	Parameter Description
N = 0xXXXX	Size: 2 Octets Range: 0x0011 to 0x1000 Default: 0x0012 Mandatory Range: 0x0011 to Inquiry Scan Interval Time = N * 0.625 msec Time Range: 10.625 msec to 2560 msec Time Default: 11.25 msec

6.4 INQUIRY SCAN TYPE

The Inquiry_Scan_Type configuration parameter indicates whether inquiry scanning will be done using non-interlaced scan or interlaced scan. Currently, one mandatory inquiry scan type and one optional inquiry scan type are defined. For details, see the Baseband Specification, “Inquiry scan substate” on page 144 [Part B].

Value	Parameter Description
0x00	Mandatory: Standard Scan (default)
0x01	Optional: Interlaced Scan
0x02-0xFF	Reserved

6.5 INQUIRY MODE

The Inquiry_Mode configuration parameter indicates whether inquiry returns Inquiry Result events in the standard format or with RSSI.

Value	Parameter Description
0x00	Standard Inquiry Result event format
0x01	Inquiry Result format with RSSI
0x02-0xFF	Reserved

6.6 PAGE TIMEOUT

The Page_Timeout configuration parameter defines the maximum time the local Link Manager will wait for a baseband page response from the remote device at a locally initiated connection attempt. If this time expires and the remote device has not responded to the page at baseband level, the connection attempt will be considered to have failed.

Value	Parameter Description
N = 0xXXXX	Size: 2 Octets Range: 0x0001 to 0xFFFF Default: 0x2000 Mandatory Range: 0x0016 to 0xFFFF Time = N * 0.625 msec Time Range: 0.625 msec to 40.9 sec Time Default: 5.12 sec

6.7 CONNECTION ACCEPT TIMEOUT

The Connection_Accept_Timeout configuration parameter allows the Bluetooth hardware to automatically deny a connection request after a specified time period has occurred and the new connection is not accepted. The parameter defines the time duration from when the Controller sends a Connection Request event until the Controller will automatically reject an incoming connection.

Value	Parameter Description
N = 0xXXXX	Size: 2 Octets Range: 0x0001 to 0xB540 Default: 0x1F40 Mandatory Range: 0x00A0 to 0xB540 Time = N * 0.625 msec Time Range: 0.625 msec to 29 sec Time Default: 5 sec

6.8 PAGE SCAN INTERVAL

The Page_Scan_Interval configuration parameter defines the amount of time between consecutive page scans. This time interval is defined from when the Controller started its last page scan until it begins the next page scan.

Value	Parameter Description
N = 0xXXXX	Size: 2 Octets Range: 0x0012 to 0x1000; only even values are valid Default: 0x0800 Mandatory Range: 0x0012 to 0x1000 Time = N * 0.625 msec Time Range: 11.25 msec to 2560 msec Time Default: 1.28 sec

6.9 PAGE SCAN WINDOW

The Page_Scan_Window configuration parameter defines the amount of time for the duration of the page scan. The Page_Scan_Window can only be less than or equal to the Page_Scan_Interval.

Value	Parameter Description
N = 0xXXXX	Size: 2 Octets Range: 0x0011 to 0x1000 Default: 0x0012 Mandatory Range: 0x0011 to Page Scan Interval Time = N * 0.625 msec Time Range: 10.625 msec to Page Scan Interval Time Default: 11.25 msec

6.10 PAGE SCAN PERIOD MODE

Every time an inquiry response message is sent, the Bluetooth device will start a timer (T_mandatory_pscan), the value of which is dependent on the Page_Scan_Period_Mode. As long as this timer has not expired, the Bluetooth device will use the mandatory page scan mode for all following page scans. Note: the timer T_mandatory_pscan will be reset at each new inquiry response. For details see the “Baseband Specification” on page 45 [Part B]. (Keyword: SP-Mode, FHS-Packet, T_mandatory_pscan, Inquiry-Response).

Value	Parameter Description
0x00	P0
0x01	P1
0x02	P2
0x03-0xFF	Reserved.

6.11 PAGE SCAN TYPE

The Page_Scan_Type parameter indicates whether inquiry scanning will be done using non-interlaced scan or interlaced scan. For details, see the Baseband Specification, “Page scan substate” on page 134 [Part B].

Value	Parameter Description
0x00	Mandatory: Standard Scan (default)
0x01	Optional: Interlaced Scan
0x02-0xFF	Reserved

6.12 VOICE SETTING

The Voice_Setting parameter controls all the various settings for voice connections. The input settings apply to all voice connections, and **cannot** be set for individual voice connections. The Voice_Setting parameter controls the configuration for voice connections: Input Coding, Air coding format, input data format, Input sample size, and linear PCM parameter.The air coding format bits in the Voice_Setting command parameter specify which air coding format the local device requests. The air coding format bits do not specify which air coding format(s) the local device accepts when a remote device requests an air coding format. This is determined by the hardware capabilities of the local device.

Value	Parameter Description
00XXXXXXXX	Input Coding: Linear
01XXXXXXXX	Input Coding: μ-law Input Coding
10XXXXXXXX	Input Coding: A-law Input Coding
11XXXXXXXX	Reserved for Future Use
XX00XXXXXX	Input Data Format: 1’s complement
XX01XXXXXX	Input Data Format: 2’s complement
XX10XXXXXX	Input Data Format: Sign-Magnitude
XX11XXXXXX	Input Data Format: Unsigned
XXXX0XXXXX	Input Sample Size: 8-bit (only for linear PCM)
XXXX1XXXXX	Input Sample Size: 16-bit (only for linear PCM)
XXXXXnnnXX	Linear_PCM_Bit_Pos: # bit positions that MSB of sample is away from starting at MSB (only for Liner PCM).
XXXXXXXX00	Air Coding Format: CVSD
XXXXXXXX01	Air Coding Format: μ-law
XXXXXXXX10	Air Coding Format: A-law
XXXXXXXX11	Air Coding Format: Transparent Data

6.13 PIN TYPE

The PIN Type configuration parameter determines whether the Link Manager assumes that the Host supports variable PIN codes or a fixed PIN code. The host controller uses the PIN-type information during pairing.

Value	Parameter Description
0x00	Variable PIN.
0x01	Fixed PIN.

6.14 LINK KEY

The Controller can store a limited number of link keys for other Bluetooth devices. Link keys are shared between two Bluetooth devices, and are used for all security transactions between the two devices. A Host device may have additional storage capabilities, which can be used to save additional link keys to be reloaded to the Bluetooth Controller when needed. A Link Key is associated with a BD_ADDR.

Value	Parameter Description
0xXXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	Link Key for an associated BD_ADDR.

6.15 AUTHENTICATION ENABLE

The Authentication_Enable parameter controls if the local device requires to authenticate the remote device at connection setup (between the Create_Connection command or acceptance of an incoming ACL connection and the corresponding Connection Complete event). At connection setup, only the device(s) with the Authentication_Enable parameter enabled will try to authenticate the other device.

Note: Changing this parameter does not affect existing connections.

Value	Parameter Description
0x00	Authentication not required.
0x01	Authentication required for all connections.
0x02-0xFF	Reserved

6.16 ENCRYPTION MODE

The Encryption_Mode parameter controls if the local device requires encryption to the remote device at connection setup (between the Create_Connection command or acceptance of an incoming ACL connection and the corresponding Connection Complete event). At connection setup, only devices with the Authentication_Enabled configuration parameter set to required and the Encryption_Mode configuration parameter set to required will try to encrypt the physical link to the other device.

Note: Changing this parameter does not affect existing connections.

A temporary link key is used when both broadcast and point-to-point traffic are encrypted.

The Host must not specify the Encryption_Mode parameter with more encryption capability than its local device currently supports, although the parameter is used to request the encryption capability to the remote device. Note that the Host must not request the command with the Encryption_Mode parameter set to 0x01, when the local device does not support encryption.

Note: for encryption to be used, both devices must support and enable encryption.

Value	Parameter Description
0x00	Encryption not required.
0x01	Encryption required for all connections.
0x02-0xFF	Reserved.

Note: in the Connection Complete event the Encryption_Mode parameter will show whether encryption was successfully turned on. The remote device may not support encryption or may have set Encryption_Mode to 0x01 when the local device has not, so the encryption mode returned in the Connection Complete event may not equal the encryption mode set in the HCI_Write_Encryption_Mode Command.

6.17 FAILED CONTACT COUNTER

The Failed_Contact_Counter records the number of consecutive incidents in which either the slave or master didn't respond after the flush timeout had expired, and the L2CAP packet that was currently being transmitted was automatically 'flushed'. When this occurs, the Failed_Contact_Counter is incremented by 1. The Failed_Contact_
Counter for a connection is reset to zero on the following conditions:

1. When a new connection is established
2. When the Failed_Contact_Counter is > zero and an L2CAP packet is acknowledged for that connection
3. When the Reset_Failed_Contact_Counter command has been issued

Value	Parameter Description
0xXXXX	Number of consecutive failed contacts for a connection corresponding to the connection handle.

6.18 HOLD MODE ACTIVITY

The Hold_Mode_Activity value is used to determine what activities should be suspended when the device is in hold mode. After the hold period has expired, the device will return to the previous mode of operation. Multiple hold mode activities may be specified for the Hold_Mode_Activity parameter by performing a bitwise OR operation of the different activity types. If no activities are suspended, then all of the current Periodic Inquiry, Inquiry Scan, and Page Scan settings remain valid during the Hold Mode. If the Hold_Mode_Activity parameter is set to Suspend Page Scan, Suspend Inquiry Scan, and Suspend Periodic Inquiries, then the device can enter a low-power state during the Hold Mode period, and all activities are suspended. Suspending multiple activities can be specified for the Hold_Mode_Activity parameter by performing a bitwise OR operation of the different activity types.The Hold Mode Activity is only valid if all connections are in Hold Mode.

Value	Parameter Description
0x00	Maintain current Power State.
0x01	Suspend Page Scan.
0x02	Suspend Inquiry Scan.
0x04	Suspend Periodic Inquiries.
0x08-0xFF	Reserved for Future Use.

6.19 LINK POLICY SETTINGS

The Link_Policy_Settings parameter determines the behavior of the local Link Manager when it receives a request from a remote device or it determines itself to change the master-slave role or to enter park state, hold, or sniff mode. The local Link Manager will automatically accept or reject such a request from the remote device, and may even autonomously request itself, depending on the value of the Link_Policy_Settings parameter for the corresponding Connection_Handle. When the value of the Link_Policy_Settings parameter is changed for a certain Connection_Handle, the new value will only be used for requests from a remote device or from the local Link Manager itself made after this command has been completed. By enabling each mode individually, the Host can choose any combination needed to support various modes of operation. Multiple LM policies may be specified for the Link_Policy_Settings parameter by performing a bitwise OR operation of the different activity types.

Note: The local device may be forced into hold mode (regardless of whether the local device is master or slave) by the remote device regardless of the value of the Link_Policy_Settings parameter. The forcing of hold mode can however only be done once the connection has already been placed into hold mode through an LMP request (the Link_Policy_Settings determine if requests from a remote device should be accepted or rejected). The forcing of hold mode can after that be done as long as the connection lasts regardless of the setting for hold mode in the Link_Policy_Settings parameter.

Note that the previous description implies that if the implementation in the remote device is a "polite" implementation that does not force another device into hold mode via LMP PDUs, then the Link_Policy_Settings will never be overruled.

Value	Parameter Description
0x0000	Disable All LM Modes Default
0x0001	Enable Role switch.
0x0002	Enable Hold Mode.
0x0004	Enable Sniff Mode.
0x0008	Enable Park State.
0x0010 – 0x8000	Reserved for Future Use.

6.20 FLUSH TIMEOUT

The Flush_Timeout configuration parameter is used for ACL connections only. The Flush Timeout is defined in the Baseband specification Section 7.6.3, “Flushing payloads,” on page 130. This parameter allows ACL packets to be automatically flushed without the Host device issuing a Flush command. This provides support for isochronous data, such as audio. When the L2CAP packet that is currently being transmitted is automatically ‘flushed’, the Failed Contact Counter is incremented by one.

Value	Parameter Description
0	Timeout = ∞; No Automatic Flush
N = 0xXXXX	Size: 2 Octets Range: 0x0001 to 0x07FF Mandatory Range: 0x0002 to 0x07FF Time = N * 0.625 msec Time Range: 0.625 msec to 1279.375 msec

6.21 NUM BROADCAST RETRANSMISSIONS

Broadcast packets are not acknowledged and are unreliable. The Number of Broadcast Retransmissions parameter, N, is used to increase the reliability of a broadcast message by retransmitting the broadcast message multiple times. This parameter defines the number of times the device will retransmit a broadcast data packet. This sets the value N_{BC} in the baseband to one greater than the Num Broadcast Retransmissions value. (See Baseband Specification, Section 7.6.5, on page 130) This parameter should be adjusted as the link quality measurement changes.

Value	Parameter Description
N = 0xXX	N_{BC} = N + 1 Range 0x00-0xFE

6.22 LINK SUPERVISION TIMEOUT

The Link_Supervision_Timeout parameter is used by the master or slave Bluetooth device to monitor link loss. If, for any reason, no Baseband packets are received from that Connection Handle for a duration longer than the Link_Supervision_Timeout, the connection is disconnected. The same timeout value is used for both synchronous and ACL connections for the device specified by the Connection Handle.

Note: Setting the Link_Supervision_Timeout to No Link_Supervision_Timeout (0x0000) will disable the Link_Supervision_Timeout check for the specified Connection Handle. This makes it unnecessary for the master of the piconet to unpark and then park each Bluetooth Device every ~40 seconds. By using the No Link_Supervision_Timeout setting, the scalability of the Park state is not limited.

Value	Parameter Description
0x0000	No Link_Supervision_Timeout.
N = 0xXXXX	Size: 2 Octets Range: 0x0001 to 0xFFFF Default: 0x7D00 Mandatory Range: 0x0190 to 0xFFFF Time = N * 0.625 msec Time Range: 0.625 msec to 40.9 sec Time Default: 20 sec

6.23 SYNCHRONOUS FLOW CONTROL ENABLE

The Synchronous Flow Control Enable confirguration parameter allows the Host to decide if the Controller will send Number Of Completed Packets events for synchronous Connection Handles. This setting allows the Host to enable and disable synchronous flow control.

Value	Parameter Description
0x00	Synchronous Flow Control is disabled. No Number of Completed Packets events will be sent from the Controller for synchronous Connection Handles.
0x01	Synchronous Flow Control is enabled. Number of Completed Packets events will be sent from the Controller for synchronous Connection Handles.

6.24 LOCAL NAME

The user-friendly Local Name provides the user the ability to distinguish one Bluetooth device from another. The Local Name configuration parameter is a UTF-8 encoded string with up to 248 octets in length. The Local Name configuration parameter will be null terminated (0x00) if the UTF-8 encoded string is less than 248 octets.

Note: the Local Name configuration parameter is a string parameter. Endianess does therefore not apply to the Local Name configuration parameter. The first octet of the name is received first.

Value	Parameter Description
	A UTF-8 encoded User Friendly Descriptive Name for the device. If the name contained in the parameter is shorter than 248 octets, the end of the name is indicated by a NULL octet (0x00), and the following octets (to fill up 248 octets, which is the length of the parameter) do not have valid values.

6.25 CLASS OF DEVICE

The Class_of_Device parameter is used to indicate the capabilities of the local device to other devices.

Value	Parameter Description
0xXXXXXX	Class of Device for the device.

6.26 SUPPORTED COMMANDS

The Supported Commands configuration parameter lists which HCI commands the local controller supports. It is implied that if a command is listed as supported, the feature underlying that command is also supported.

The Supported Commands is a 64 octet bit field. If a bit is set to 1, then this command is supported.

Octet	Bit	Command Supported
0	0	Inquiry
	1	Inquiry Cancel
	2	Periodic Inquiry Mode
	3	Exit Periodic Inquiry Mode
	4	Create Connection
	5	Disconnect
	6	Add SCO Connection
	7	Cancel Create Connection
1	0	Accept Connection Request
	1	Reject Connection Request
	2	Link Key Request Reply
	3	Link Key Request Negative Reply
	4	PIN Code Request Reply
	5	PIN Code Request Negative Reply
	6	Change Connection Packet Type
	7	Authentication Request
2	0	Set Connection Encryption
	1	Change Connection Link Key
	2	Master Link Key
	3	Remote Name Request
	4	Cancel Remote Name Request
	5	Read Remote Supported Features
	6	Read Remote Extended Features
	7	Read Remote Version Information
3	0	Read Clock Offset
	1	Read LMP Handle
	2	Reserved
	3	Reserved
	4	Reserved
	5	Reserved
	6	Reserved
	7	Reserved

Octet	Bit	Command Supported
4	0	Reserved
	1	Hold Mode
	2	Sniff Mode
	3	Exit Sniff Mode
	4	Park State
	5	Exit Park State
	6	QoS Setup
	7	Role Discovery
5	0	Switch Role
	1	Read Link Policy Settings
	2	Write Link Policy Settings
	3	Read Default Link Policy Settings
	4	Write Default Link Policy Settings
	5	Flow Specification
	6	Set Event Mark
	7	Reset
6	0	Set Event Filter
	1	Flush
	2	Read PIN Type
	3	Write PIN Type
	4	Create New Unit Key
	5	Read Stored Link Key
	6	Write Stored Link Key
	7	Delete Stored Link Key
7	0	Write Local Name
	1	Read Local Name
	2	Read Connection Accept Timeout
	3	Write Connection Accept Timeout
	4	Read Page Timeout
	5	Write Page Timeout
	6	Read Scan Enable
	7	Write Scan Enable
8	0	Read Page Scan Activity
	1	Write Page Scan Activity
	2	Read Inquiry Scan Activity
	3	Write Inquiry Scan Activity
	4	Read Authentication Enable
	5	Write Authentication Enable
	6	Read Encryption Mode
	7	Write Encryption Mode

Octet	Bit	Command Supported
9	0	Read Class Of Device
	1	Write Class Of Device
	2	Read Voice Setting
	3	Write Voice Setting
	4	Read Automatic Flush Timeout
	5	Write Automatic Flush Timeout
	6	Read Num Broadcast Retransmissions
	7	Write Num Broadcast Retransmissions
10	0	Read Hold Mode Activity
	1	Write Hold Mode Activity
	2	Read Transmit Power Level
	3	Read Synchronous Flow Control Enable
	4	Write Synchronous Flow Control Enable
	5	Set Host Controller To Host Flow Control
	6	Host Buffer Size
	7	Host Number Of Completed Packets
11	0	Read Link Supervision Timeout
	1	Write Link Supervision Timeout
	2	Read Number of Supported IAC
	3	Read Current IAC LAP
	4	Write Current IAC LAP
	5	Read Page Scan Period Mode
	6	Write Page Scan Period Mode
	7	Read Page Scan Mode
12	0	Write Page Scan Mode
	1	Set AFH Channel Classification
	2	reserved
	3	reserved
	4	Read Inquiry Scan Type
	5	Write Inquiry Scan Type
	6	Read Inquiry Mode
	7	Write Inquiry Mode
13	0	Read Page Scan Type
	1	Write Page Scan Type
	2	Read AFH Channel Assessment Mode
	3	Write AFH Channel Assessment Mode
	4	Reserved
	5	Reserved
	6	Reserved
	7	Reserved

Octet	Bit	Command Supported
14	0	Reserved
	1	Reserved
	2	Reserved
	3	Read Local Version Information
	4	Reserved
	5	Read Local Supported Features
	6	Read Local Extended Features
	7	Read Buffer Size
15	0	Read Country Code
	1	Read BD ADDR
	2	Read Failed Contact Count
	3	Reset Failed Contact Count
	4	Get Link Quality
	5	Read RSSI
	6	Read AFH Channel Map
	7	Read BD Clock
16	0	Read Loopback Mode
	1	Write Loopback Mode
	2	Enable Device Under Test Mode
	3	Setup Synchronous Connection
	4	Accept Synchronous Connection
	5	Reject Synchronous Connection[
	6	Reserved
	7	Reserved

7 HCI COMMANDS AND EVENTS

7.1 LINK CONTROL COMMANDS

The Link Control commands allow the Controller to control connections to other Bluetooth devices. When the Link Control commands are used, the Link Manager (LM) controls how the Bluetooth piconets and scatternets are established and maintained. These commands instruct the LM to create and modify link layer connections with Bluetooth remote devices, perform Inquiries of other Bluetooth devices in range, and other LMP commands. For the Link Control commands, the OGF is defined as 0x01.

7.1.1 Inquiry Command

Command	OCF	Command Parameters	Return Parameters
HCI_Inquiry	0x0001	LAP, Inquiry_Length, Num_Responses	

Description:

This command will cause the Bluetooth device to enter Inquiry Mode. Inquiry Mode is used to discover other nearby Bluetooth devices. The LAP input parameter contains the LAP from which the inquiry access code shall be derived when the inquiry procedure is made. The Inquiry_Length parameter specifies the total duration of the Inquiry Mode and, when this time expires, Inquiry will be halted. The Num_Responses parameter specifies the number of responses that can be received before the Inquiry is halted. A Command Status event is sent from the Controller to the Host when the Inquiry command has been started by the Bluetooth device. When the Inquiry process is completed, the Controller will send an Inquiry Complete event to the Host indicating that the Inquiry has finished. The event parameters of Inquiry Complete event will have a summary of the result from the Inquiry process, which reports the number of nearby Bluetooth devices that responded. When a Bluetooth device responds to the Inquiry message, an Inquiry Result event will occur to notify the Host of the discovery.

A device which responds during an inquiry or inquiry period should always be reported to the Host in an Inquiry Result event if the device has not been reported earlier during the current inquiry or inquiry period and the device has not been filtered out using the command Set_Event_Filter. If the device has been reported earlier during the current inquiry or inquiry period, it may or may not be reported depending on the implementation (depending on if earlier results have been saved in the Controller and in that case how many responses that have been saved). It is recommended that the Controller tries to report a particular device only once during an inquiry or inquiry period.

Command Parameters:

LAP: *Size: 3 Octets*

Value	Parameter Description
0x9E8B00– 0X9E8B3F	This is the LAP from which the inquiry access code should be derived when the inquiry procedure is made; see “Bluetooth Assigned Numbers” (https://www.bluetooth.org/foundry/assignnumb/document/assigned_numbers).

Inquiry_Length: *Size: 1 Octet*

Value	Parameter Description
N = 0xXX	Maximum amount of time specified before the Inquiry is halted. Size: 1 octet Range: 0x01 – 0x30 Time = N * 1.28 sec Range: 1.28 – 61.44 Sec

Num_Responses: *Size: 1 Octet*

Value	Parameter Description
0x00	Unlimited number of responses.
0xXX	Maximum number of responses from the Inquiry before the Inquiry is halted. Range: 0x01 – 0xFF

Return Parameters:

None.

Event(s) generated (unless masked away):

A Command Status event is sent from the Controller to the Host when the Controller has started the Inquiry process. An Inquiry Result event will be created for each Bluetooth device which responds to the Inquiry message. In addition, multiple Bluetooth devices which respond to the Inquire message may be combined into the same event. An Inquiry Complete event is generated when the Inquiry process has completed.
Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, the Inquiry Complete event will indicate that this command has been completed. No Inquiry Complete event will be generated for the canceled Inquiry process.

7.1.2 Inquiry Cancel Command

Command	OCF	Command Parameters	Return Parameters
HCI_Inquiry_Cancel	0x0002		Status

Description:

This command will cause the Bluetooth device to stop the current Inquiry if the Bluetooth device is in Inquiry Mode. This command allows the Host to interrupt the Bluetooth device and request the Bluetooth device to perform a different task. The command should only be issued after the Inquiry command has been issued, a Command Status event has been received for the Inquiry command, and before the Inquiry Complete event occurs.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Inquiry_Cancel command succeeded.
0x01-0xFF	Inquiry_Cancel command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Inquiry Cancel command has completed, a Command Complete event will be generated. No Inquiry Complete event will be generated for the canceled Inquiry process.

7.1.3 Periodic Inquiry Mode Command

Command	OCF	Command Parameters	Return Parameters
HCI_Periodic_ Inquiry_Mode	0x0003	Max_Period_Length, Min_Period_Length, LAP, Inquiry_Length, Num_Responses	Status

Description:

The Periodic_Inquiry_Mode command is used to configure the Bluetooth device to enter the Periodic Inquiry Mode that performs an automatic Inquiry. Max_Period_Length and Min_Period_Length define the time range between two consecutive inquiries, from the beginning of an inquiry until the start of the next inquiry. The Controller will use this range to determine a new random time between two consecutive inquiries for each Inquiry. The LAP input parameter contains the LAP from which the inquiry access code shall be derived when the inquiry procedure is made. The Inquiry_Length parameter specifies the total duration of the InquiryMode and, when time expires, Inquiry will be halted. The Num_Responses parameter specifies the number of responses that can be received before the Inquiry is halted. This command is completed when the Inquiry process has been started by the Bluetooth device, and a Command Complete event is sent from the Controller to the Host. When each of the periodic Inquiry processes are completed, the Controller will send an Inquiry Complete event to the Host indicating that the latest periodic Inquiry process has finished. When a Bluetooth device responds to the Inquiry message an Inquiry Result event will occur to notify the Host of the discovery.
Note: Max_Period_Length > Min_Period_Length > Inquiry_Length

A device which responds during an inquiry or inquiry period should always be reported to the Host in an Inquiry Result event if the device has not been reported earlier during the current inquiry or inquiry period and the device has not been filtered out using the command Set_Event_Filter. If the device has been reported earlier during the current inquiry or inquiry period, it may or may not be reported depending on the implementation (depending on if earlier results have been saved in the Controller and in that case how many responses that have been saved). It is recommended that the Controller tries to report a particular device only once during an inquiry or inquiry period.

Command Parameters:

Max_Period_Length: *Size: 2 Octets*

Value	Parameter Description
N = 0xXXXX	Maximum amount of time specified between consecutive inquiries. Size: 2 octets Range: 0x03 – 0xFFFF Time = N * 1.28 sec Range: 3.84 – 83884.8 Sec 0.0 – 23.3 hours

Min_Period_Length: *Size: 2 Octets*

Value	**Parameter Description**
N = 0xXXXX	Minimum amount of time specified between consecutive inquiries. Size: 2 octets Range: 0x02 – 0xFFFE Time = N * 1.28 sec Range: 2.56 – 83883.52 Sec 0.0 – 23.3 hours

LAP: *Size: 3 Octets*

Value	**Parameter Description**
0x9E8B00– 0X9E8B3F	This is the LAP from which the inquiry access code should be derived when the inquiry procedure is made, see “Bluetooth Assigned Numbers” (https://www.bluetooth.org/foundry/assignnumb/document/assigned_numbers).

Inquiry_Length: *Size: 1 Octet*

Value	**Parameter Description**
N = 0xXX	Maximum amount of time specified before the Inquiry is halted. Size: 1 octet Range: 0x01 – 0x30 Time = N * 1.28 sec Range: 1.28 – 61.44 Sec

Num_Responses: *Size: 1 Octet*

Value	**Parameter Description**
0x00	Unlimited number of responses.
0xXX	Maximum number of responses from the Inquiry before the Inquiry is halted. Range: 0x01 – 0xFF

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Periodic Inquiry Mode command succeeded.
0x01-0xFF	Periodic Inquiry Mode command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

The Periodic Inquiry Mode begins when the Controller sends the Command Complete event for this command to the Host. An Inquiry Result event will be created for each Bluetooth device which responds to the Inquiry message. In addition, multiple Bluetooth devices which response to the Inquiry message may be combined into the same event. An Inquiry Complete event is generated when each of the periodic Inquiry processes has completed. No Inquiry Complete event will be generated for the canceled Inquiry process.

7.1.4 Exit Periodic Inquiry Mode Command

Command	OCF	Command Parameters	Return Parameters
HCI_Exit_Periodic_Inquiry_Mode	0x0004		Status

Description:

The Exit Periodic Inquiry Mode command is used to end the Periodic Inquiry mode when the local device is in Periodic Inquiry Mode. If the local device is currently in an Inquiry process, the Inquiry process will be stopped directly and the Controller will no longer perform periodic inquiries until the Periodic Inquiry Mode command is reissued.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Exit Periodic Inquiry Mode command succeeded.
0x01-0xFF	Exit Periodic Inquiry Mode command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

A Command Complete event for this command will occur when the local device is no longer in Periodic Inquiry Mode. No Inquiry Complete event will be generated for the canceled Inquiry process.

7.1.5 Create Connection Command

Command	OCF	Command Parameters	Return Parameters
HCI_Create_Connection	0x0005	BD_ADDR, Packet_Type, Page_Scan_Repetition_Mode, Reserved, Clock_Offset, Allow_Role_Switch	

Description:

This command will cause the Link Manager to create a connection to the Bluetooth device with the BD_ADDR specified by the command parameters. This command causes the local Bluetooth device to begin the Page process to create a link level connection. The Link Manager will determine how the new ACL connection is established. This ACL connection is determined by the current state of the device, its piconet, and the state of the device to be connected. The Packet_Type command parameter specifies which packet types the Link Manager shall use for the ACL connection. When sending HCI ACL Data Packets the Link Manager shall only use the packet type(s) specified by the Packet_Type command parameter or the always-allowed DM1 packet type. Multiple packet types may be specified for the Packet Type parameter by performing a bit-wise OR operation of the different packet types. The Link Manager may choose which packet type to be used from the list of acceptable packet types. The Page_Scan_Repetition_Mode parameter specifies the page scan repetition mode supported by the remote device with the BD_ADDR. This is the information that was acquired during the inquiry process. The Clock_Offset parameter is the difference between its own clock and the clock of the remote device with BD_ADDR. Only bits 2 through 16 of the difference are used, and they are mapped to this parameter as bits 0 through 14 respectively. A Clock_Offset_Valid_Flag, located in bit 15 of the Clock_Offset parameter, is used to indicate if the Clock Offset is valid or not. A Connection handle for this connection is returned in the Connection Complete event (see below). The Allow_Role_Switch parameter specifies if the local device accepts or rejects the request of a master-slave role switch when the remote device requests it at the connection setup (in the Role parameter of the Accept_Connection_Request command) (before the local Controller returns a Connection Complete event). For a definition of the different packet types see the “Baseband Specification” on page 45 [Part B].

Note: The Host should enable as many packet types as possible for the Link Manager to perform efficiently. However, the Host must not enable packet types that the local device does not support.

Command Parameters:

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device to be connected.

Packet_Type: *Size: 2 Octets*

Value	Parameter Description
0x0001	Reserved for future use.
0x0002	Reserved for future use.
0x0004	Reserved for future use.
0x0008[1]	DM1
0x0010	DH1
0x0020	Reserved for future use.
0x0040	Reserved for future use.
0x0080	Reserved for future use.
0x0100	Reserved for future use.
0x0200	Reserved for future use.
0x0400	DM3
0x0800	DH3
0x1000	Reserved for future use.
0x2000	Reserved for future use.
0x4000	DM5
0x8000	DH5

1. this bit will be interpreted as set to 1 by link Controllers based on Bluetooth v1.2 or newer.

Page_Scan_Repetition_Mode: *Size: 1 Octet*

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
0x03 – 0xFF	Reserved.

Reserved: *Size: 1 Octet*

Value	Parameter Description
0x00	Reserved, must be set to 0x00. See “Page Scan Mode” on page 586.

Clock_Offset: *Size: 2 Octets*

Bit format	Parameter Description
Bit 14-0	Bit 16-2 of CLKslave-CLKmaster.
Bit 15	Clock_Offset_Valid_Flag Invalid Clock Offffset = 0 Valid Clock Offset = 1

Allow_Role_Switch: *Size: 1 Octet*

Value	Parameter Description
0x00	The local device will be a master, and will not accept a role switch requested by the remote device at the connection setup.
0x01	The local device may be a master, or may become a slave after accepting a role switch requested by the remote device at the connection setup.
0x02-0xFF	Reserved for future use.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the Create Connection command, the Controller sends the Command Status event to the Host. In addition, when the LM determines the connection is established, the Controller, on both Bluetooth devices that form the connection, will send a Connection Complete event to each Host. The Connection Complete event contains the Connection Handle if this command is successful.
Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, the Connection Complete event will indicate that this command has been completed.

7.1.6 Disconnect Command

Command	OCF	Command Parameters	Return Parameters
HCI_Disconnect	0x0006	Connection_Handle, Reason	

Description:

The Disconnection command is used to terminate an existing connection. The Connection_Handle command parameter indicates which connection is to be disconnected. The Reason command parameter indicates the reason for ending the connection. The remote Bluetooth device will receive the Reason command parameter in the Disconnection Complete event. All synchronous connections on a physical link should be disconnected before the ACL connection on the same physical connection is disconnected.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle for the connection being disconnected. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Reason: *Size: 1 Octet*

Value	Parameter Description
0x05, 0x13-0x15, 0x1A, 0x29	Authentication Failure error code (0x05), Other End Terminated Connection error codes (0x13-0x15), Unsupported Remote Feature error code (0x1A) and Pairing with Unit Key Not Supported error code (0x29), see “Error Codes” on page 291 [Part D] for list of Error Codes.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the Disconnect command, it sends the Command Status event to the Host. The Disconnection Complete event will occur at each Host when the termination of the connection has completed, and indicates that this command has been completed.
Note: No Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, the Disconnection Complete event will indicate that this command has been completed.

7.1.7 Create Connection Cancel Command

Command	OCF	Command Parameters	Return Parameters
HCI_Create_Connection _Cancel	0x0008	BD_ADDR	Status, BD_ADDR

Description:

This command is used to request cancellation of the ongoing connection creation process, which was started by a Create_Connection command of the local device.

Command Parameters:

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Create Connection command request that was issued before and is subject of this cancellation request

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Create Connection Cancel command succeeded
0x01-0xff	Create Connection Cancel command failed. See “Error Codes” on page 291 [Part D] for list of error codes

BD_ADDR: *Size: 6 Octet*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Create Connection command that was issued before and is the subject of this cancellation request.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 142

Event(s) generated (unless masked away):

When the Create Connection Cancel command has completed, a Command Complete event shall be generated.

If the connection is already established by the baseband, but the Controller has not yet sent the Connection Complete event, then the local device shall detach the link and return a Command Complete event with the status “Success”.

If the connection is already established, and the Connection Complete event has been sent, then the Controller shall return a Command Complete event with the error code *ACL Connection already exists* (0x0B).

If the Create Connection Cancel command is sent to the Controller without a preceding Create Connection command to the same device, the Controller shall return a Command Complete event with the error code *Unknown Connection Identifier* (0x02).

The error codes *Unspecified Error* (0x1F) and *Command Disallowed* (0x0C) may be used if the Controller does not support this command.

The Connection Complete event for the corresponding Create Connection Command shall always be sent. The Connection Complete event shall be sent after the Command Complete event for the Create Connection Cancel command. If the cancellation was successful, the Connection Complete event will be generated with the error code *Unknown Connection Identifier* (0x02).

7.1.8 Accept Connection Request Command

Command	OCF	Command Parameters	Return Parameters
HCI_Accept_Connection Request	0x0009	BD_ADDR, Role	

Description:

The Accept_Connection_Request command is used to accept a new incoming connection request. The Accept_Connection_Request command shall only be issued after a Connection Request event has occurred. The Connection Request event will return the BD_ADDR of the device which is requesting the connection. This command will cause the Link Manager to create a connection to the Bluetooth device, with the BD_ADDR specified by the command parameters. The Link Manager will determine how the new connection will be established. This will be determined by the current state of the device, its piconet, and the state of the device to be connected. The Role command parameter allows the Host to specify if the Link Manager shall request a role switch and become the Master for this connection. This is a preference and not a requirement. If the Role Switch fails then the connection will still be accepted, and the Role Discovery Command will reflect the current role.
Note: The Link Manager may terminate the connection if it would be low on resources if the role switch fails. The decision to accept a connection must be completed before the connection accept timeout expires on the local Bluetooth Module.
Note: when accepting synchronous connection request, the Role parameter is not used and will be ignored by the Controller.

Command Parameters:

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device to be connected

Role: *Size: 1 Octet*

Value	Parameter Description
0x00	Become the Master for this connection. The LM will perform the role switch.
0x01	Remain the Slave for this connection. The LM will **NOT** perform the role switch.

Return Parameters:

None.

Event(s) generated (unless masked away):

The Accept_Connection_Request command will cause the Command Status event to be sent from the Controller when the Controller begins setting up the connection. In addition, when the Link Manager determines the connection is established,the local Controller will send a Connection Complete event to its Host, and the remote Controller will send a Connection Complete event or a Synchronous Connection Complete event to the Host. The Connection Complete event contains the Connection Handle if this command is successful. Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, the Connection Complete event will indicate that this command has been completed.

7.1.9 Reject Connection Request Command

Command	OCF	Command Parameters	Return Parameters
HCI_Reject_Connection _Request	0x000A	BD_ADDR, Reason	

Description:

The Reject_Connection_Request command is used to decline a new incoming connection request. The Reject_Connection_Request command shall only be called after a Connection Request event has occurred. The Connection Request event will return the BD_ADDR of the device that is requesting the connection. The Reason command parameter will be returned to the connecting device in the Status parameter of the Connection Complete event returned to the Host of the connection device, to indicate why the connection was declined.

Command Parameters:

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device to reject the connection from.

Reason: *Size: 1 Octet*

Value	Parameter Description
0x0D-0x0F	Host Reject Error Code. See “Error Codes” on page 291 [Part D] for list of Error Codes and descriptions.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the Reject_Connection_Request command, the Controller sends the Command Status event to the Host. Then, the local Controller will send a Connection Complete event to its host, and the remote Controller will send a Connection Complete event or a Synchronous Connection Complete event to the host. The Status parameter of the Connection Complete event, which is sent to the Host of the device attempting to make the connection, will contain the Reason command parameter from this command.
Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, the Connection Complete event will indicate that this command has been completed.

7.1.10 Link Key Request Reply Command

Command	OCF	Command Parameters	Return Parameters
HCI_Link_Key_Request_Reply	0x000B	BD_ADDR, Link_Key	Status, BD_ADDR

Description:

The Link_Key_Request_Reply command is used to reply to a Link Key Request event from the Controller, and specifies the Link Key stored on the Host to be used as the link key for the connection with the other Bluetooth Device specified by BD_ADDR. The Link Key Request event will be generated when the Controller needs a Link Key for a connection.

When the Controller generates a Link Key Request event in order for the local Link Manager to respond to the request from the remote Link Manager (as a result of a Create_Connection or Authentication_Requested command from the remote Host), the local Host must respond with either a Link_Key_Request_Reply or Link_Key_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See “Link Manager Protocol” on page 189 [Part C].)

Command Parameters:

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device of which the Link Key is for.

Link_Key: *Size: 16 Octets*

Value	Parameter Description
0xXXXXXXXXXXXXXXXXXXXXXXXXXXXX	Link Key for the associated BD_ADDR.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Link_Key_Request_Reply command succeeded.
0x01-0xFF	Link_Key_Request_Reply command failed. See “Error Codes” on page 291 [Part D] for error codes and description.

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device of which the Link Key request reply has completed.

Event(s) generated (unless masked away):

The Link_Key_Request_Reply command will cause a Command Complete event to be generated.

7.1.11 Link Key Request Negative Reply Command

Command	OCF	Command Parameters	Return Parameters
HCI_Link_Key_Request_Negative_Reply	0x000C	BD_ADDR	Status, BD_ADDR

Description:

The Link_Key_Request_Negative_Reply command is used to reply to a Link Key Request event from the Controller if the Host does not have a stored Link Key for the connection with the other Bluetooth Device specified by BD_ADDR. The Link Key Request event will be generated when the Controller needs a Link Key for a connection.

When the Controller generates a Link Key Request event in order for the local Link Manager to respond to the request from the remote Link Manager (as a result of a Create_Connection or Authentication_Requested command from the remote Host), the local Host must respond with either a Link_Key_Request_Reply or Link_Key_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See “Link Manager Protocol” on page 189 [Part C].)

Command Parameters:

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device which the Link Key is for.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Link_Key_Request_Negative_Reply command succeeded.
0x01-0xFF	Link_Key_Request_Negative_Reply command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device which the Link Key request negative reply has completed.

Event(s) generated (unless masked away):

The Link_Key_Request_Negative_Reply command will cause a Command Complete event to be generated.

7.1.12 PIN Code Request Reply Command

Command	OCF	Command Parameters	Return Parameters
HCI_PIN_Code_Request_Reply	0x000D	BD_ADDR, PIN_Code_Length, PIN_Code	Status, BD_ADDR

Description:

The PIN_Code_Request_Reply command is used to reply to a PIN Code request event from the Controller, and specifies the PIN code to use for a connection. The PIN Code Request event will be generated when a connection with remote initiating device has requested pairing.

When the Controller generates a PIN Code Request event in order for the local Link Manager to respond to the request from the remote Link Manager (as a result of a Create_Connection or Authentication_Requested command from the remote Host), the local Host must respond with either a PIN_Code_Request_Reply or PIN_Code_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See “Link Manager Protocol” on page 189 [Part C].)

Command Parameters:

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device which the PIN code is for.

PIN_Code_Length: *Size: 1 Octet*

Value	Parameter Description
0xXX	The PIN code length specifics the length, in octets, of the PIN code to be used. Range: 0x01-0x10

PIN_Code: *Size: 16 Octets*

Value	Parameter Description
0xXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	PIN code for the device that is to be connected. The Host should insure that strong PIN Codes are used. PIN Codes can be up to a maximum of 128 bits. Note: the PIN_Code Parameter is a string parameter. Endianess does therefore not apply to the PIN_Code Parameter. The first octet of the PIN code should be transmitted first.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 150

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	PIN_Code_Request_Reply command succeeded.
0x01-0xFF	PIN_Code_Request_Reply command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device which the PIN Code request reply has completed.

Event(s) generated (unless masked away):

The PIN_Code_Request_Reply command will cause a Command Complete event to be generated.

7.1.13 PIN Code Request Negative Reply Command

Command	OCF	Command Parameters	Return Parameters
HCI_PIN_Code_ Request_Negative Reply	0x000E	BD_ADDR	Status, BD_ADDR

Description:

The PIN_Code_Request_Negative_Reply command is used to reply to a PIN Code request event from the Controller when the Host cannot specify a PIN code to use for a connection. This command will cause the pair request with remote device to fail.

When the Controller generates a PIN Code Request event in order for the local Link Manager to respond to the request from the remote Link Manager (as a result of a Create_Connection or Authentication_Requested command from the remote Host), the local Host must respond with either a PIN_Code_Request_Reply or PIN_Code_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See “Link Manager Protocol” on page 189 [Part C].)

Command Parameters:

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device which this command is responding to.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	PIN_Code_Request_Negative_Reply command succeeded.
0x01-0xFF	PIN_Code_Request_Negative_Reply command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device which the PIN Code request negative reply has completed.

Event(s) generated (unless masked away):

The PIN_Code_Request_Negative_Reply command will cause a Command Complete event to be generated.

7.1.14 Change Connection Packet Type Command

Command	OCF	Command Parameters	Return Parameters
HCI_Change_Connection_ Packet_Type	0x000F	Connection_Handle, Packet_Type	

Description:

The Change_Connection_Packet_Type command is used to change which packet types can be used for a connection that is currently established. This allows current connections to be dynamically modified to support different types of user data. The Packet_Type command parameter specifies which packet types the Link Manager can use for the connection. When sending HCI ACL Data Packets the Link Manager shall only use the packet type(s) specified by the Packet_Type command parameter or the always-allowed DM1 packet type. The interpretation of the value for the Packet_Type command parameter will depend on the Link_Type command parameter returned in the Connection Complete event at the connection setup. Multiple packet types may be specified for the Packet_Type command parameter by bitwise OR operation of the different packet types. For a definition of the different packet types see the “Baseband Specification” on page 45 [Part B].

Note: The Host should enable as many packet types as possible for the Link Manager to perform efficiently. However, the Host must not enable packet types that the local device does not support.

Note: to change an eSCO connection, use the Setup Synchronous Connection command.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to for transmitting and receiving voice or data. Returned from creating a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Packet_Type: *Size: 2 Octets*

For ACL Link_Type

Value	Parameter Description
0x0001	Reserved for future use.
0x0002	Reserved for future use.
0x0004	Reserved for future use.
0x0008[1]	DM1
0x0010	DH1
0x0020	Reserved for future use.
0x0040	Reserved for future use.
0x0080	Reserved for future use.
0x0100	Reserved for future use.
0x0200	Reserved for future use.
0x0400	DM3
0x0800	DH3
0x1000	Reserved for future use.
0x2000	Reserved for future use.
0x4000	DM5
0x8000	DH5

1. this bit will be interpreted as set to 1 by link Controllers based on Bluetooth v1.2 or newer.

For SCO Link Type

Value	Parameter Description
0x0001	Reserved for future use.
0x0002	Reserved for future use.
0x0004	Reserved for future use.
0x0008	Reserved for future use.
0x0010	Reserved for future use.
0x0020	HV1
0x0040	HV2
0x0080	HV3
0x0100	Reserved for future use.
0x0200	Reserved for future use.
0x0400	Reserved for future use.
0x0800	Reserved for future use.
0x1000	Reserved for future use.
0x2000	Reserved for future use.
0x4000	Reserved for future use.
0x8000	Reserved for future use.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the Change Connection Packet Type command, the Controller sends the Command Status event to the Host. In addition, when the Link Manager determines the packet type has been changed for the connection, the Controller on the local device will send a Connection Packet Type Changed event to the Host. This will be done at the local side only.
Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, the Connection Packet Type Changed event will indicate that this command has been completed.

7.1.15 Authentication Requested Command

Command	OCF	Command Parameters	Return Parameters
HCI_Authentication_ Requested	0x0011	Connection_Handle	

Description:

The Authentication_Requested command is used to try to authenticate the remote device associated with the specified Connection Handle. On an authentication failure, the Controller or Link Manager shall not automatically detach the link. The Host is responsible for issuing a Disconnect command to terminate the link if the action is appropriate.
Note**:** the Connection_Handle command parameter is used to identify the other Bluetooth device, which forms the connection. The Connection Handle should be a Connection Handle for an ACL connection.

Command Parameters:

Connection_Handle: *Size 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to set up authentication for all Connection Handles with the same Bluetooth device end-point as the specified Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the Authentication_Requested command, it sends the Command Status event to the Host. The Authentication Complete event will occur when the authentication has been completed for the connection and is indication that this command has been completed.
Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, the Authentication Complete event will indicate that this command has been completed.

Note: When the local or remote Controller does not have a link key for the specified Connection_Handle, it will request the link key from its Host, before the local Host finally receives the Authentication Complete event.

7.1.16 Set Connection Encryption Command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Connection_ Encryption	0x0013	Connection_Handle, Encryption_Enable	

Description:

The Set_Connection_Encryption command is used to enable and disable the link level encryption. Note**:** the Connection_Handle command parameter is used to identify the other Bluetooth device which forms the connection. The Connection Handle should be a Connection Handle for an ACL connection. While the encryption is being changed, all ACL traffic must be turned off for all Connection Handles associated with the remote device.

Command Parameters:

Connection_Handle: *Size 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to enable/disable the link layer encryption for all Connection Handles with the same Bluetooth device end-point as the specified Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Encryption_Enable: *Size: 1 Octet*

Value	Parameter Description
0x00	Turn Link Level Encryption OFF.
0x01	Turn Link Level Encryption ON.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the Set_Connection_Encryption command, the Controller sends the Command Status event to the Host. When the Link Manager has completed enabling/disabling encryption for the connection, the Controller on the local Bluetooth device will send an Encryption Change event to the Host, and the Controller on the remote device will also generate an Encryption Change event.
Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, the Encryption Change event will indicate that this command has been completed.

7.1.17 Change Connection Link Key Command

Command	OCF	Command Parameters	Return Parameters
HCI_Change_Connection_ Link_Key	0x0015	Connection_Handle	

Description:

The Change_Connection_Link_Key command is used to force both devices of a connection associated with the connection handle to generate a new link key. The link key is used for authentication and encryption of connections.
Note: the Connection_Handle command parameter is used to identify the other Bluetooth device forming the connection. The Connection Handle should be a Connection Handle for an ACL connection.

Command Parameters:

Connection_Handle: *Size 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the Change_Connection_Link_Key command, the Controller sends the Command Status event to the Host. When the Link Manager has changed the Link Key for the connection, the Controller on the local Bluetooth device will send a Link Key Notification event and a Change Connection Link Key Complete event to the Host, and the Controller on the remote device will also generate a Link Key Notification event. The Link Key Notification event indicates that a new connection link key is valid for the connection.
Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, the Change Connection Link Key Complete event will indicate that this command has been completed.

7.1.18 Master Link Key Command

Command	OCF	Command Parameters	Return Parameters
HCI_Master_Link_Key	0x0017	Key_Flag	

Description:

The Master Link Key command is used to force the device that is master of the piconet to use the temporary link key of the master device, or the semi-permanent link keys. The temporary link key is used for encryption of broadcast messages within a piconet, and the semi-permanent link keys are used for private encrypted point-to-point communication. The Key_Flag command parameter is used to indicate which Link Key (temporary link key of the Master, or the semi-permanent link keys) shall be used.

Command Parameters:

Key_Flag: *Size: 1 Octet*

Value	Parameter Description
0x00	Use semi-permanent Link Keys.
0x01	Use Temporary Link Key.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the Master_Link_Key command, the Controller sends the Command Status event to the Host. When the Link Manager has changed link key, the Controller on both the local and the remote device will send a Master Link Key Complete event to the Host. The Connection Handle on the master side will be a Connection Handle for one of the existing connections to a slave. On the slave side, the Connection Handle will be a Connection Handle to the initiating master.

The Master Link Key Complete event contains the status of this command. Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, the Master Link Key Complete event will indicate that this command has been completed.

7.1.19 Remote Name Request Command

Command	OCF	Command Parameters	Return Parameters
HCI_Remote_Name_Request	0x0019	BD_ADDR, Page_Scan_Repetition_Mode, Reserved, Clock_Offset	

Description:

The Remote_Name_Request command is used to obtain the user-friendly name of another Bluetooth device. The user-friendly name is used to enable the user to distinguish one Bluetooth device from another. The BD_ADDR command parameter is used to identify the device for which the user-friendly name is to be obtained. The Page_Scan_Repetition_Mode parameter specifies the page scan repetition mode supported by the remote device with the BD_ADDR. This is the information that was acquired during the inquiry process. The Clock_Offset parameter is the difference between its own clock and the clock of the remote device with BD_ADDR. Only bits 2 through 16 of the difference are used and they are mapped to this parameter as bits 0 through 14 respectively. A Clock_Offset_Valid_Flag, located in bit 15 of the Clock_Offset command parameter, is used to indicate if the Clock Offset is valid or not. Note: if no connection exists between the local device and the device corresponding to the BD_ADDR, a temporary link layer connection will be established to obtain the name of the remote device.

Command Parameters:

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR for the device whose name is requested.

Page_Scan_Repetition_Mode: *Size: 1 Octet*

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
0x03 – 0xFF	Reserved.

Reserved: *Size: 1 Octet*

Value	Parameter Description
0x00	Reserved, must be set to 0x00. See “Page Scan Mode” on page 586.

Clock_Offset: *Size: 2 Octets*

Bit format	Parameter Description
Bit 14.0	Bit 16.2 of CLKslave-CLKmaster.
Bit 15	Clock_Offset_Valid_Flag Invalid Clock Offffset = 0 Valid Clock Offset = 1

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the Remote_Name_Request command, the Controller sends the Command Status event to the Host. When the Link Manager has completed the LMP messages to obtain the remote name, the Controller on the local Bluetooth device will send a Remote Name Request Complete event to the Host. Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, only the Remote Name Request Complete event will indicate that this command has been completed.

7.1.20 Remote Name Request Cancel Command

Command	OCF	Command Parameters	Return Parameters
HCI_Remote_Name_Request_Cancel	0x001A	BD_ADDR	Status, BD_ADDR

Description:

This command is used to request cancellation of the ongoing remote name request process, which was started by the Remote Name Request command.

Command Parameters:

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Remote Name Request command that was issued before and that is subject of this cancellation request

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Remote Name Request Cancel command succeeded
0x01-0xff	Remote Name Request Cancel command failed. See “Error Codes” on page 291 [Part D] for list of error codes

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Remote Name Request Cancel command that was issued before and that was subject of this cancellation request

Event(s) generated (unless masked away):

When the Remote Name Request Cancel command has completed, a Command Complete event shall be generated.

If the Remote Name Request Cancel command is sent to the Controller without a preceding Remote Name Request command to the same device, the Controller will return a Command Complete event with the error code *Invalid HCI Command Parameters* (0x12).

The error codes *Unspecified Error* (0x1F) and *Command Disallowed* (0x0C) may be used if the Controller does not support this command.

The Remote Name Request Complete event for the corresponding Remote Name Request Command shall always be sent. The Remote Name Request Complete event shall be sent after the Command Complete event for the Remote Name Request Cancel command. If the cancellation was successful, the Remote Name Request Complete event will be generated with the error code *Unknown Connection Identifier (0x02)*.

7.1.21 Read Remote Supported Features Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Remote_Supported_Features	0x001B	Connection_Handle	

Description:

This command requests a list of the supported features for the remote device identified by the Connection_Handle parameter. The Connection_Handle must be a Connection_Handle for an ACL connection. The Read Remote Supported Features Complete event will return a list of the LMP features. For details see “Link Manager Protocol” on page 189 [Part C].

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle’s LMP-supported features list to get. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the Read_Remote_Supported_Features command, the Controller sends the Command Status event to the Host. When the Link Manager has completed the LMP messages to determine the remote features, the Controller on the local Bluetooth device will send a Read Remote Supported Features Complete event to the Host. The Read Remote Supported Features Complete event contains the status of this command, and parameters describing the supported features of the remote device. Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, the Read Remote Supported Features Complete event will indicate that this command has been completed.

7.1.22 Read Remote Extended Features Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Remote_Extended_Features	0x001C	Connection_Handle, Page Number	

Description:

The HCI_Read_Remote_Extended_Features command returns the requested page of the extended LMP features for the remote device identified by the specified connection handle. The connection handle must be the connection handle for an ACL connection. This command is only available if the extended features feature is implemented by the remote device. The Read Remote Extended Features Complete event will return the requested information.
For details see “Link Manager Protocol” on page 189 [Part C].

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The connection handle identifying the remote device for which extended feature information is required. Range: 0x0000-0x0EFF (0x0F00-0x0FFF Reserved for future use)

Page Number: *Size: 1 Octet*

Value	Parameter Description
0x00	Requests the normal LMP features as returned by HCI_Read_Remote_Supported_Features
0x01-0xFF	Return the corresponding page of features

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the HCI_Read_Remote_Extended_Features command the Controller sends the Command Status command to the Host. When the Link Manager has completed the LMP sequence to determine the remote extended features the controller on the local device will generate a Read Remote Extended Features Complete event to the host. The Read Remote Extended Features Complete event contains the page number and the remote features returned by the remote device.

Note: no Command Complete event will ever be sent by the Controller to indicate that this command has been completed. Instead the Read Remote Extended Features Complete event will indicate that this command has been completed.

7.1.23 Read Remote Version Information Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Remote_Version_ Information	0x001D	Connection_Handle	

Description:

This command will obtain the values for the version information for the remote Bluetooth device identified by the Connection_Handle parameter. The Connection_Handle must be a Connection_Handle for an ACL connection.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's version information to get. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the Read_Remote_Version_Information command, the Controller sends the Command Status event to the Host. When the Link Manager has completed the LMP messages to determine the remote version information, the Controller on the local Bluetooth device will send a Read Remote Version Information Complete event to the Host. The Read Remote Version Information Complete event contains the status of this command, and parameters describing the version and subversion of the LMP used by the remote device.
Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, the Read Remote Version Information Complete event will indicate that this command has been completed.

7.1.24 Read Clock Offset Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Clock_Offset	0x001F	Connection_Handle	

Description:

Both the System Clock and the clock offset to a remote device are used to determine what hopping frequency is used by a remote device for page scan. This command allows the Host to read clock offset to remote devices. The clock offset can be used to speed up the paging procedure when the local device tries to establish a connection to a remote device, for example, when the local Host has issued Create_Connection or Remote_Name_Request. The Connection_Handle must be a Connection_Handle for an ACL connection.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Clock Offset parameter is returned. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the Read_Clock_Offset command, the Controller sends the Command Status event to the Host. If this command was requested at the master and the Link Manager has completed the LMP messages to obtain the Clock Offset information, the Controller on the local Bluetooth device will send a Read Clock Offset Complete event to the Host.
Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, only the Read Clock Offset Complete event will indicate that this command has been completed. If the command is requested at the slave, the LM will immediately send a Command Status event and a Read Clock Offset Complete event to the Host, without an exchange of LMP PDU.

7.1.25 Read LMP Handle Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_LMP_Handle	0x0020	Connection_Handle	Status, Connection_Handle, LMP_Handle, Reserved

Description:

This command will read the current LMP Handle associated with the Connection_Handle. The Connection_Handle must be a SCO or eSCO Handle. If the Connection_Handle is a SCO connection handle, then this command shall read the LMP SCO Handle for this connection. If the Connection_Handle is an eSCO connection handle, then this command shall read the LMP eSCO Handle for this connection.

Command Parameters:

Connection_Handle: *Size 2 Octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify which connection to be used for reading the LMP Handle. This must be a synchronous handle. Range: 0x0000-0x0EFF (0x0F00 – 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_LMP_Handle command succeeded.
0x01 – 0xFF	Read_LMP_Handle command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the LMP_Handle has been read. Range: 0x0000-0x0EFF (0x0F00 – 0x0FFF Reserved for future use)

LMP_Handle: *Size: 1 Octet*

Value	Parameter Description
0xXX	The LMP Handle is the LMP Handle that is associated with this connection handle. For a synchronous handle, this would be the LMP Synchronous Handle used when negotiating the synchronous connection in the link manager.

Reserved: *Size: 4 Octets*

Value	Parameter Description
0xXXXXXXXX	This parameter is reserved, must to set to zero.

Events(s) generated (unless masked away):

When the Read_LMP_Handle command has completed, a Command Complete event will be generated.

7.1.26 Setup Synchronous Connection Command

Command	OCF	Command Parameters	Return Parameters
HCI_Setup_Synchronous_ Connection	0x0028	Connection_Handle Transmit_Bandwidth Receive_Bandwidth Max_Latency Voice_Setting Retransmission_Effort Packet Type	

Description:

The HCI Setup Synchronous Connection command adds a new or modifies an existing synchronous logical transport (SCO or eSCO) on a physical link depending on the Connection_Handle parameter specified. If the Connection_Handle refers to an ACL link a new synchronous logical transport will be added. If the Connection_Handle refers to an already existing synchronous logical transport (eSCO only) this link will be modified.The parameters are specified per connection. This synchronous connection can be used to transfer synchronous voice at 64kbps or transparent synchronous data.

When used to setup a new synchronous logical transport, the Connection_Handle parameter shall specify an ACL connection with which the new synchronous connection will be associated. The other parameters relate to the negotiation of the link, and may be reconfigured during the lifetime of the link. The transmit and receive bandwidth specify how much bandwidth shall be available for transmitting and for receiving data. While in many cases the receive and transmit bandwidth parameters may be equal, they may be different. The latency specifies an upper limit to the time in milliseconds between the eSCO (or SCO) instants, plus the size of the retransmission window, plus the length of the reserved synchronous slots for this logical transport. The content format specifies the settings for voice or transparent data on this connection. The retransmission effort specifies the extra resources that are allocated to this connection if a packet may need to be retransmitted. The Retransmission_Effort parameter shall be set to indicate the required behaviour, or to don't care.

When used to modify an existing synchronous logical transport, the Transmit_Bandwidth, Receive_Bandwidth and Voice_Settings shall be set to the same values as were used during the initial setup. The Packet_Type, Retransmission_Effort and Max_Latency parameters may be modified.

The Packet_Type field is a bitmap specifying which packet types the LM shall accept in the negotiation of the link parameters. Multiple packet types are specified by bitwise OR of the packet type codes in the table. At least one packet type must be specified for each negotiation. It is recommended to enable as many packet types as possible. Note that it is allowed to enable packet types that are not supported by the local device.

A connection handle for the new synchronous connection will be returned in the synchronous connection complete event.

Note: The link manager may choose any combination of packet types, timing, and retransmission window sizes that satisfy the parameters given. This may be achieved by using more frequent transmissions of smaller packets. The link manager may choose to set up either a SCO or an eSCO connection, if the parameters allow, using the corresponding LMP sequences.

Note: To modify a SCO connection, use the Change Connection Packet Type command.

Note: If the lower layers cannot acheive the exact transmit and receive bandwidth requested subject to the other parameters, then the link shall be rejected.

A synchronous connection may only be created when an ACL connection already exists and when it is not in park state.

Command Parameters:

Connection_Handle: *2 octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle for the ACL connection being used to create a synchronousConnection or for the existing Connection that shall be modified. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Transmit_Bandwidth: *4 octets*

Value	Parameter Description
0xXXXXXXXX	Transmit bandwidth in octets per second.

Receive_Bandwidth: *4 octets*

Value	Parameter Description
0xXXXXXXXX	Receive bandwidth in octets per second.

Max_Latency: *2 octets*

Value	Parameter Description
0x0000-0x0003	Reserved
0x0004-0xFFFE	This is a value in milliseconds representing the upper limit of the sum of the synchronous interval and the size of the eSCO window.
0xFFFF	Don't care.

Voice_Setting: *2 octets (10 bits meaningful)*

Value	Parameter Description
See Section 6.12 on page 361.	

Retransmission_Effort: *1 octet*

Value	Parameter Description
0x00	No retransmissions
0x01	At least one retransmission, optimize for power consumption.
0x02	At least one retransmission, optimize for link quality
0xFF	Don’t care
0x03 – 0xFE	Reserved

Packet_Type: *2 octets*

Value	Parameter Description
0x0001	HV1
0x0002	HV2
0x0004	HV3
0x0008	EV3
0x0010	EV4
0x0020	EV5
0x0040	Reserved for future use
0x0080	Reserved for future use
0x0100	Reserved for future use
0x0200	Reserved for future use
0x0400	Reserved for future use
0x0800	Reserved for future use
0x1000	Reserved for future use
0x2000	Reserved for future use
0x4000	Reserved for future use
0x8000	Reserved for future use

Return Parameters:

None

Event(s) generated (unless masked away)

When the Controller receives the Setup_Synchronous_Connection command, it sends the Command Status event to the Host. In addition, when the LM determines the connection is established, the local Controller will send a Synchronous Connection Complete event to the local Host, and the remote Controller will send a Synchronous Connection Complete event or a Connection Complete event to the remote Host. The synchronous Connection Complete event contains the Connection Handle if this command is successful.

If this command is used to change the parameters of an existing eSCO link, the Synchronous Connection Changed Event is sent to both hosts. In this case no Connection Setup Complete Event or Connection Request Event will be sent to either host. This command cannot be used to change the parameters of an SCO link.

Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, the synchronous Connection Complete event will indicate that this command has been completed.

7.1.27 Accept Synchronous Connection Request Command

Command	OCF	Command Parameters	Return Parameters
HCI_Accept_ Synchronous_ Connection_Request	0x0029	BD_ADDR Transmit_Bandwidth Receive_Bandwidth Max_Latency Content_Format Retransmission_Effort Packet_Type	

Description:

The Accept_Synchronous_Connection_Request command is used to accept an incoming request for a synchronous connection and to inform the local Link Manager about the acceptable parameter values for the synchronous connection. The Command shall only be issued after a Connection_Request event with link type SCO or eSCO has occurred. Connection_Request event contains the BD_ADDR of the device requesting the connection. The decision to accept a connection must be taken before the connection accept timeout expires on the local device.

The parameter set of the Accept_Synchronous_Connection_Request command is the same as for the Setup_Synchronous_Connection command. The Transmit_Bandwidth and Receive_Bandwidth values are required values for the new link and shall be met. The Max_Latency is an upper bound to the acceptable latency for the Link, as defined in Section 7.1.26 on page 411 Setup_Synchronous_Connection and shall not be exceeded. Content_Format specifies the encoding in the same way as in the Setup_Synchronous_Connection command and shall be met. The Retransmission_Effort parameter shall be set to indicate the required behaviour, or to don't care. The Packet_Type parameter is a bit mask specifying the synchronous packet types that are allowed on the link and shall be met. The reserved bits in the Packet_Type field shall be set to one.

If the Link Type of the incoming request is SCO, then only the Transmit_Bandwidth, Max_Latency, Content_Format, and Packet_Type fields are valid.

If the Connection_Request event is masked away, and the Controller is not set to auto-accept this connection attempt, the Controller will automatically reject it. If the controller is set to automatically accept the connection attempt, the LM should assume default parameters. In that case the Synchronous_Connection_Complete Event shall be generated, unless masked away.

Command Parameters:

BD_ADDR: *6 octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the device requesting the connection

Transmit_Bandwidth: *4 octets*

Value	Parameter Description
0x00000000-0xFFFFFFFE	Maximum possible transmit bandwidth in octets per second.
0xFFFFFFFF	Don’t care

Default: Don’t care

Receive_Bandwidth: *4 octets*

Value	Parameter Description
0x00000000-0xFFFFFFFE	Maximum possible receive bandwidth in octets per second.
0xFFFFFFFF	Don’t care

Default: Don’t care

Max_Latency: *2 octets*

Value	Parameter Description
0x0000-0x0003	Reserved
0x0004-0xFFFE	This is a value in milliseconds representing the upper limit of the sum of the synchronous interval and the size of the eSCO window.
0xFFFF	Don't care.

Default: Don't care

Petitioner Kiosoft Exhibit 1011 Part 2
Page 174

Content_Format: *2 octets (10 bits meaningful)*

Value	Parameter Description
00XXXXXXXX	Input Coding: Linear
01XXXXXXXX	Input Coding: u-law
10XXXXXXXX	Input Coding: A-law
11XXXXXXXX	Reserved for future use.
XX00XXXXXX	Input Data Format: 1's complement
XX01XXXXXX	Input Data Format: 2's complement
XX10XXXXXX	Input Data Format: Sign-Magnitude
XX11XXXXXX	Input Data Format: Unsigned
XXXX0XXXXX	Input Sample Size: 8 bit (only for Linear PCM)
XXXX1XXXXX	Input Sample Size: 16 bit (only for Linear PCM)
XXXXXnnnXX	Linear PCM Bit Position: number of bit positions that MSB of sample is away from starting at MSB (only for Linear PCM)
XXXXXXXX00	Air Coding Format: CVSD
XXXXXXXX01	Air Coding Format: u-law
XXXXXXXX10	Air Coding Format: A-law
XXXXXXXX11	Air Coding Format: Transparent Data

Default: When links are auto-accepted, the values written by the HCI_Write_Voice_Settings are used.

Retransmission_Effort: *1 octet*

Value	Parameter Description
0x00	No retransmissions
0x01	At least one retransmission, optimize for power consumption.
0x02	At least one retransmission, optimize for link quality.
0x03-0xFE	Reserved
0xFF	Don't care

Default: Don't care

Packet_Type: *2 octets*

Value	Parameter Description
0x0001	HV1
0x0002	HV2
0x0004	HV3
0x0008	EV3
0x0010	EV4
0x0020	EV5
0x0040	Reserved for future use
0x0080	Reserved for future use
0x0100	Reserved for future use
0x0200	Reserved for future use
0x0400	Reserved for future use
0x0800	Reserved for future use
0x1000	Reserved for future use
0x2000	Reserved for future use
0x4000	Reserved for future use
0x8000	Reserved for future use

Default: 0xFFFF

Return Parameters:

None

Event(s) generated (unless masked away):

The Accept_Synchronous_Request command will cause the Command Status event to be sent from the Host Controller when the Host Controller starts setting up the connection. When the link setup is complete, the local Controller will send a Synchronous Connection Complete event to its Host, and the remote Controller will send a Connection Complete event or a Synchronous Connection Complete event to the Host. The Synchronous Connection Complete will contain the connection handle and the link parameters if the setup is successful. No Command Complete event will be sent by the host controller as the result of this command.

7.1.28 Reject Synchronous Connection Request Command

Command	OCF	Command Parameters	Return Parameters
HCI_Reject_Synchronous_ Connection_Request	0x002A	BD_ADDR Reason	

Description:

The Reject_Synchronous_Connection_Request is used to decline an incoming request for a synchronous link. It shall only be issued after a Connection Request Event with Link Type equal to SCO or eSCO has occurred. The Connection Request Event contains the BD_ADDR of the device requesting the connection. The Reason parameter will be returned to the initiating host in the Status parameter of the Synchronous connection complete event on the remote side.

Command Parameters:

BD_ADDR: *6 octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the device requesting the connection

Reason: *1 octet*

Value	Parameter Description
0x0D-0x0F	Host Reject Error Code. See “Error Codes” on page 291 [Part D] for error codes and description.

Return Parameters:

None.

Event(s) Generated (unless masked away):

When the Host Controller receives the Reject_Synchronous_Connection_Request, it sends a Command Status Event to the Host. When the setup is terminated, the local Controller will send a Synchronous Connection Complete event to its Host, and the remote Controller will send a Connection Complete event or a Synchronous Connection Complete event to the Host with the Reason code from this command. No Command complete Event will be sent by the Host Controller to indicate that this command has been completed.

7.2 LINK POLICY COMMANDS

The Link Policy Commands provide methods for the Host to affect how the Link Manager manages the piconet. When Link Policy Commands are used, the LM still controls how Bluetooth piconets and scatternets are established and maintained, depending on adjustable policy parameters. These policy commands modify the Link Manager behavior that can result in changes to the link layer connections with Bluetooth remote devices.
Note: only one ACL connection can exist between two Bluetooth Devices, and therefore there can only be one ACL HCI Connection Handle for each physical link layer Connection. The Bluetooth Controller provides policy adjustment mechanisms to provide support for a number of different policies. This capability allows one Bluetooth module to be used to support many different usage models, and the same Bluetooth module can be incorporated in many different types of Bluetooth devices. For the Link Policy Commands, the OGF is defined as 0x02.

7.2.1 Hold Mode Command

Command	OCF	Command Parameters	Return Parameters
HCI_Hold_Mode	0x0001	Connection_Handle, Hold_Mode_Max_Interval, Hold_Mode_Min_Interval	

Description:

The Hold_Mode command is used to alter the behavior of the Link Manager, and have it place the ACL baseband connection associated by the specified Connection Handle into the hold mode. The Hold_Mode_Max_Interval and Hold_Mode_Min_Interval command parameters specify the length of time the Host wants to put the connection into the hold mode. The local and remote devices will negotiate the length in the hold mode. The Hold_Mode_Max_Interval parameter is used to specify the maximum length of the Hold interval for which the Host may actually enter into the hold mode after negotiation with the remote device. The Hold interval defines the amount of time between when the Hold Mode begins and when the Hold Mode is completed. The Hold_Mode_Min_Interval parameter is used to specify the minimum length of the Hold interval for which the Host may actually enter into the hold mode after the negotiation with the remote device. Therefore the Hold_Mode_Min_Interval cannot be greater than the Hold_Mode_Max_Interval. The Controller will return the actual Hold interval in the Interval parameter of the Mode Change event, if the command is successful. This command enables the Host to support a low-power policy for itself or several other Bluetooth devices, and allows the devices to enter Inquiry Scan, Page Scan, and a number of other possible actions.
Note: the connection handle cannot be of the SCO or eSCO link type

If the Host sends data to the Controller with a Connection_Handle corresponding to a connection in hold mode, the Controller will keep the data in its buffers until either the data can be transmitted (the hold mode has ended) or a flush, a flush timeout or a disconnection occurs. This is valid even if the Host has not yet been notified of the hold mode through a Mode Change event when it sends the data.
Note: the above is not valid for an HCI Data Packet sent from the Host to the Controller on the master side where the Connection_Handle is a Connection_Handle used for broadcast and the Broadcast_Flag is set to Active Broadcast or Piconet Broadcast. The broadcast data will then never be received by slaves in hold mode.

The Hold_Mode_Max_Interval shall be less than the Link Supervision Timeout configuration parameter.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Hold_Mode_Max_Interval: *Size: 2 Octets*

Value	Parameter Description
N = 0xXXXX	Maximum acceptable number of Baseband slots to wait in Hold Mode. Time Length of the Hold = N * 0.625 msec (1 Baseband slot) Range for N: 0x0002-0xFFFE; only even values are valid. Time Range: 1.25ms - 40.9 sec Mandatory Range: 0x0014 to 0x8000

Hold_Mode_Min_Interval: *Size: 2 Octets*

Value	Parameter Description
N = 0xXXXX	Minimum acceptable number of Baseband slots to wait in Hold Mode. Time Length of the Hold = N * 0.625 msec (1 Baseband slot) Range for N: 0x0002-0xFF00; only even values are valid Time Range: 1.25 msec - 40.9 sec Mandatory Range: 0x0014 to 0x8000

Return Parameters:

None.

Event(s) generated (unless masked away):

The Controller sends the Command Status event for this command to the Host when it has received the Hold_Mode command. The Mode Change event will occur when the Hold Mode has started and the Mode Change event will occur again when the Hold Mode has completed for the specified connection handle. The Mode Change event signaling the end of the Hold Mode is an estimation of the hold mode ending if the event is for a remote Bluetooth device.
Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, only the Mode Change event will indicate that this command has been completed. If an error occurs after the Command Status event has occurred, then the status in the Mode Change event will indicate the error.

7.2.2 Sniff Mode Command

Command	OCF	Command Parameters	Return Parameters
HCI_Sniff_Mode	0x0003	Connection_Handle, Sniff_Max_Interval, Sniff_Min_Interval, Sniff_Attempt, Sniff_Timeout	

Description:

The Sniff Mode command is used to alter the behavior of the Link Manager and have it place the ACL baseband connection associated with the specified Connection Handle into the sniff mode. The Connection_Handle command parameter is used to identify which ACL link connection is to be placed in sniff mode. The Sniff_Max_Interval and Sniff_Min_Interval command parameters are used to specify the requested acceptable maximum and minimum periods in the Sniff Mode. The Sniff_Min_Interval shall not be greater than the Sniff_Max_Interval. The sniff interval defines the amount of time between each consecutive sniff period. The Controller will return the actual sniff interval in the Interval parameter of the Mode Change event, if the command is successful. For a description of the meaning of the Sniff_Attempt and Sniff_Timeout parameters, see Baseband Specification, Section 8.7, on page 163. Sniff_Attempt is there called $N_{sniff\ attempt}$ and Sniff_Timeout is called $N_{sniff\ timeout}$. This command enables the Host to support a low-power policy for itself or several other Bluetooth devices, and allows the devices to enter Inquiry Scan, Page Scan, and a number of other possible actions.
Note: in addition, the connection handle cannot be one of the synchronous link types. If the Host sends data to the Controller with a Connection_Handle corresponding to a connection in sniff mode, the Controller will keep the data in its buffers until either the data can be transmitted or a flush, a flush timeout or a disconnection occurs. This is valid even if the Host has not yet been notified of the sniff mode through a Mode Change event when it sends the data. Note that it is possible for the master to transmit data to a slave without exiting sniff mode

(see description in Baseband Specification, Section 8.7, on page 163).
Note: the above is not valid for an HCI Data Packet sent from the Host to the Controller on the master side where the Connection_Handle is a Connection_Handle used for broadcast and the Broadcast_Flag is set to Active Broadcast or Piconet Broadcast. In that case, the broadcast data will only be received by a slave in sniff mode if that slave happens to listen to the master when the broadcast is made.

The Sniff_Max_Interval shall be less than the Link Supervision Timeout configuration parameter.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Sniff_Max_Interval: *Size: 2 Octets*

Value	Parameter Description
N = 0xXXXX	Range: 0x0002 to 0xFFFE; only even values are valid Mandatory Range: 0x0006 to 0x0540 Time = N * 0.625 msec Time Range: 1.25 msec to 40.9 sec

Sniff_Min_Interval: *Size: 2 Octets*

Value	Parameter Description
N = 0xXXXX	Range: 0x0002 to 0xFFFE; only even values are valid Mandatory Range: 0x0006 to 0x0540 Time = N * 0.625 msec Time Range: 1.25 msec to 40.9 sec)

Sniff_Attempt: *Size: 2 Octets*

Value	Parameter Description
N = 0xXXXX	Number of Baseband receive slots for sniff attempt. Length = N* 1.25 msec Range for N: 0x0001 – 0x7FFF Time Range: 0.625msec - 40.9 Seconds Mandatory Range for Controller: 1 to $T_{sniff}/2$

Sniff_Timeout: *Size: 2 Octets*

Value	Parameter Description
N = 0xXXXX	Number of Baseband receive slots for sniff timeout. Length = N * 1.25 msec Range for N: 0x0000 – 0x7FFF Time Range: 0 msec - 40.9 Seconds Mandatory Range for Controller: 0 to 0x0028

Return Parameters:

None.

Event(s) generated (unless masked away):

The Controller sends the Command Status event for this command to the Host when it has received the Sniff_Mode command. The Mode Change event will occur when the Sniff Mode has started for the specified connection handle. Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead only the Mode Change event will indicate that this command has been completed. If an error occurs after the Command Status event has occurred, then the status in the Mode Change event will indicate the error.

7.2.3 Exit Sniff Mode Command

Command	OCF	Command Parameters	Return Parameters
HCI_Exit_Sniff_Mode	0x0004	Connection_Handle	

Description:

The Exit_Sniff_Mode command is used to end the sniff mode for a connection handle, which is currently in sniff mode. The Link Manager will determine and issue the appropriate LMP commands to remove the sniff mode for the associated Connection Handle.
Note: in addition, the connection handle cannot be one of the synchronous link types.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

A Command Status event for this command will occur when Controller has received the Exit_Sniff_Mode command. The Mode Change event will occur when the Sniff Mode has ended for the specified connection handle.
Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, only the Mode Change event will indicate that this command has been completed.

7.2.4 Park State Command

Command	OCF	Command Parameters	Return Parameters
HCI_Park_State	0x0005	Connection_Handle, Beacon_Max_Interval, Beacon_Min_Interval	

Description:

The Park State command is used to alter the behavior of the Link Manager, and have the LM place the baseband connection associated by the specified Connection Handle into Park state. The Connection_Handle command parameter is used to identify which connection is to be placed in Park state. The Connection_Handle must be a Connection_Handle for an ACL connection. The Beacon Interval command parameters specify the acceptable length of the interval between beacons. However, the remote device may request shorter interval. The Beacon_Max_Interval parameter specifies the acceptable longest length of the interval between beacons. The Beacon_Min_Interval parameter specifies the acceptable shortest length of the interval between beacons. Therefore, the Beacon Min Interval cannot be greater than the Beacon Max Interval. The Controller will return the actual Beacon interval in the Interval parameter of the Mode Change event, if the command is successful. This command enables the Host to support a low-power policy for itself or several other Bluetooth devices, allows the devices to enter Inquiry Scan, Page Scan, provides support for large number of Bluetooth Devices in a single piconet, and a number of other possible activities.
Note: when the Host issues the Park State command, no Connection Handles for synchronous connections are allowed to exist to the remote device that is identified by the Connection_Handle parameter. If one or more Connection Handles for synchronous connections exist to that device, depending on the implementation, a Command Status event or a Mode Change event (following a Command Status event where Status=0x00) will be returned with the error code 0x0C *Command Disallowed*.
If the Host sends data to the Controller with a Connection_Handle corresponding to a parked connection, the Controller will keep the data in its buffers until either the data can be transmitted (after unpark) or a flush, a flush timeout or a disconnection occurs. This is valid even if the Host has not yet been notified of park state through a Mode Change event when it sends the data.
Note: the above is not valid for an HCI Data Packet sent from the Host to the Controller on the master side where the Connection_Handle is a Connection_Handle used for Piconet Broadcast and the Broadcast_Flag is set to Piconet Broadcast. In that case, slaves in park state will also receive the broadcast data. (If the Broadcast_Flag is set to Active Broadcast, the broadcast data will usually not be received by slaves in park state.)
It is possible for the Controller to do an automatic unpark to transmit data and then park the connection again depending on the value of the Link_Policy_Settings parameter (see Write_Link_Policy_Settings) and depending on whether the imple-

mentation supports this or not (optional feature). The optional feature of automatic unpark/park can also be used for link supervision. Whether Mode Change events are returned or not at automatic unpark/park if this is implemented, is vendor specific. This could be controlled by a vendor specific HCI command.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Beacon_Max_Interval: *Size: 2 Octets*

Value	Parameter Description
N = 0xXXXX	Range: 0x000E to 0xFFFE; only even values are valid Mandatory Range: 0x000E to 0x1000 Time = N * 0.625 msec Time Range: 8.75 msec to 40.9 sec

Beacon_Min_Interval *Size: 2 Octets*

Value	Parameter Description
N = 0xXXXX	Range: 0x000E to 0xFFFE; only even values are valid Mandatory Range: 0x000E to 0x1000 Time = N * 0.625 msec Time Range: 8.75 msec to 40.9 sec

Return Parameters:

None.

Event(s) generated (unless masked away):

The Controller sends the Command Status event for this command to the Host when it has received the Park State command. The Mode Change event will occur when the Park State has started for the specified connection handle.
Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, only the Mode Change event will indicate that this command has been completed. If an error occurs after the Command Status event has occurred, then the status in the Mode Change event will indicate the error.

7.2.5 Exit Park State Command

Command	OCF	Command Parameters	Return Parameters
HCI_Exit_Park_State	0x0006	Connection_Handle	

Description:

The Exit_Park_State command is used to switch the Bluetooth device from park state back to the active mode. This command may only be issued when the device associated with the specified Connection Handle is in park state. The Connection_Handle must be a Connection_Handle for an ACL connection. This function does not complete immediately.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

None.

Event(s) generated (unless masked away):

A Command Status event for this command will occur when the Controller has received the Exit_Park_State command. The Mode Change event will occur when park state has ended for the specified connection handle.
Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, only the Mode Change event will indicate that this command has been completed.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 186

7.2.6 QoS Setup Command

Command	OCF	Command Parameters	Return Parameters
HCI_QoS_Setup	0x0007	Connection_Handle, Flags, Service_Type, Token_Rate, Peak_Bandwidth, Latency, Delay_Variation	

Description:

The QoS_Setup command is used to specify Quality of Service parameters for a connection handle. The Connection_Handle must be a Connection_Handle for an ACL connection. These QoS parameter are the same parameters as L2CAP QoS. For more detail see “Logical Link Control and Adaptation Protocol Specification” on page 15[vol. 3]. This allows the Link Manager to have all of the information about what the Host is requesting for each connection. The LM will determine if the QoS parameters can be met. Bluetooth devices that are both slaves and masters can use this command. When a device is a slave, this command will trigger an LMP request to the master to provide the slave with the specified QoS as determined by the LM. When a device is a master, this command is used to request a slave device to accept the specified QoS as determined by the LM of the master. The Connection_Handle command parameter is used to identify for which connection the QoS request is requested.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify which connection for the QoS Setup. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Flags: *Size: 1 Octet*

Value	Parameter Description
0x00 – 0xFF	Reserved for Future Use.

Service_Type: *Size: 1 Octet*

Value	Parameter Description
0x00	No Traffic.
0x01	Best Effort.
0x02	Guaranteed.
0x03-0xFF	Reserved for Future Use.

Token_Rate: *Size: 4 Octets*

Value	Parameter Description
0xXXXXXXXX	Token Rate in octets per second.

Peak_Bandwidth: *Size: 4 Octets*

Value	Parameter Description
0xXXXXXXXX	Peak Bandwidth in octets per second.

Latency: *Size: 4 Octets*

Value	Parameter Description
0xXXXXXXXX	Latency in microseconds.

Delay_Variation: *Size: 4 Octets*

Value	Parameter Description
0xXXXXXXXX	Delay Variation in microseconds.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the QoS_Setup command, the Controller sends the Command Status event to the Host. When the Link Manager has completed the LMP messages to establish the requested QoS parameters, the Controller on the local Bluetooth device will send a QoS Setup Complete event to the Host, and the event may also be generated on the remote side if there was LMP negotiation. The values of the parameters of the QoS Setup Complete event may, however, be different on the initiating and the remote side. The QoS Setup Complete event returned by the Controller on the local side contains the status of this command, and returned QoS parameters describing the supported QoS for the connection.
Note: No Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, the QoS Setup Complete event will indicate that this command has been completed.

7.2.7 Role Discovery Command

Command	OCF	Command Parameters	Return Parameters
HCI_Role_Discovery	0x0009	Connection_Handle	Status, Connection_Handle, Current_Role

Description:

The Role_Discovery command is used for a Bluetooth device to determine which role the device is performing for a particular Connection Handle. The Connection_Handle must be a Connection_Handle for an ACL connection.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Role_Discovery command succeeded,
0x01-0xFF	Role_Discovery command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify which connection the Role_Discovery command was issued on. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Current_Role: *Size: 1 Octet*

Value	Parameter Description
0x00	Current Role is Master for this Connection Handle.
0x01	Current Role is Slave for this Connection Handle.

Event(s) generated (unless masked away):

When the Role_Discovery command has completed, a Command Complete event will be generated.

7.2.8 Switch Role Command

Command	OCF	Command Parameters	Return Parameters
HCI_Switch_Role	0x000B	BD_ADDR, Role	

Description:

The Switch_Role command is used for a Bluetooth device to switch the current role the device is performing for a particular connection with another specified Bluetooth device. The BD_ADDR command parameter indicates for which connection the role switch is to be performed. The Role indicates the requested new role that the local device performs.
Note: the BD_ADDR command parameter must specify a Bluetooth device for which a connection already exists.

Note: If there is an SCO connection between the local device and the device identified by the BD_ADDR parameter, an attempt to perform a role switch shall be rejected by the local device.

]Note: If the connection between the local device and the device identified by the BD_ADDR parameter is placed in Sniff Mode, an attempt to perform a role switch will be rejected by the local device.

Command Parameters:

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR for the connected device with which a role switch is to be performed.

Role: *Size: 1 Octet*

Value	Parameter Description
0x00	Change own Role to Master for this BD_ADDR.
0x01	Change own Role to Slave for this BD_ADDR.

Return Parameters:

None.

Event(s) generated (unless masked away):

A Command Status event for this command will occur when the Controller has received the Switch_Role command. When the role switch is performed, a Role Change event will occur to indicate that the roles have been changed, and will be communicated to both Hosts.
Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, only the Role Change event will indicate that this command has been completed.

7.2.9 Read Link Policy Settings Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Link_Policy_ Settings	0x000C	Connection_Handle	Status, Connection_Handle Link_Policy_Settings

Description:

This command will read the Link Policy setting for the specified Connection Handle. The Connection_Handle must be a Connection_Handle for an ACL connection. Section 6.19 on page 365.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Link_Policy_Settings command succeeded.
0x01-0xFF	Read_Link_Policy_Settings command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Link_Policy_Settings *Size: 2 Octets*

Value	Parameter Description
0x0000	Disable All LM Modes Default.
0x0001	Enable Role Switch.
0x0002	Enable Hold Mode.
0x0004	Enable Sniff Mode.
0x0008	Enable Park State.
0x0010 – 0x8000	Reserved for Future Use.

Event(s) generated (unless masked away):

When the Read_Link_Policy_Settings command has completed, a Command Complete event will be generated.

7.2.10 Write Link Policy Settings Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Link_Policy_Settings	0x000D	Connection_Handle, Link_Policy_Settings	Status, Connection_Handle

Description:

This command will write the Link Policy setting for the specified Connection Handle. The Connection_Handle must be a Connection_Handle for an ACL connection. See Section 6.19 on page 365.

The default value is the value set by the Write Default Link Policy Settings Command.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Petitioner Kiosoft Exhibit 1011 Part 2
Page 192

Link_Policy_Settings *Size: 2 Octets*

Value	Parameter Description
0x0000	Disable All LM Modes.
0x0001	Enable Role Switch.
0x0002	Enable Hold Mode.
0x0004	Enable Sniff Mode.
0x0008	Enable Park State.
0x0010 – 0x8000	Reserved for Future Use.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Link_Policy_Settings command succeeded.
0x01-0xFF	Write_Link_Policy_Settings command failed. See “Error Codes” on page 291 [Part D] for error codes and description.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Event(s) generated (unless masked away):

When the Write_Link_Policy_Settings command has completed, a Command Complete event will be generated.

7.2.11 Read Default Link Policy Settings Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Default_Link_Policy_Settings	0x000E		Status, Default_Link_Policy_Settings

Description:

This command will read the Default Link Policy setting for all new connections.

Note: Please refer to the Link Policy Settings configuration parameter for more information. See Section 6.19 on page 365.

Command Parameters:

None

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Link_Policy_Settings command succeeded
0x01-0xFF	Read_Link_Policy_Settings command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Default_Link_Policy_Settings *Size: 2 Octets*

Value	Parameter Description
0x0000	Disable All LM Modes Default
0x0001	Enable Role Switch
0x0002	Enable Hold Mode
0x0004	Enable Sniff Mode
0x0008	Enable Park State
0x0010 - 0x8000	Reserved for future use.

Event(s) generated (unless masked away):

When the Read_Default_Link_Policy_Settings command has completed, a Command Complete event will be generated.

7.2.12 Write Default Link Policy Settings Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Default_Link_Policy_Settings	0x000F	Default_Link_Policy_Settings	Status

Description:

This command will write the Default Link Policy configuration value. The Default_Link_Policy_Settings parameter determines the initial value of the Link_Policy_Settings for all new connections.

Note: Please refer to the Link Policy Settings configuration parameter for more information. See Section 6.19 on page 365.

Command Parameters:

Default_Link_Policy_Settings *Size: 2 Octets*

Value	Parameter Description
0x0000	Disable All LM Modes Default
0x0001	Enable Role Switch
0x0002	Enable Hold Mode
0x0004	Enable Sniff Mode
0x0008	Enable Park State
0x0010 - 0x8000	Reserved for future use.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Link_Policy_Settings command succeeded
0x01-0xFF	Write_Link_Policy_Settings command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Default_Link_Policy_Settings command has completed, a Command Complete event will be generated.

7.2.13 Flow Specification Command

Command	OCF	Command Parameters	Return Parameters
HCI_Flow_Specification	0x0010	Connection_Handle, Flags, Flow_direction, Service_Type, Token_Rate, Token_Bucket_Size, Peak_Bandwidth, Access Latency	

Description:

The Flow_Specification command is used to specify the flow parameters for the traffic carried over the ACL connection identified by the Connection_Handle. The Connection_Handle must be a Connection_Handle for an ACL connection. The Connection_Handle command parameter is used to identify for which connection the Flow Specification is requested. The flow parameters refer to the outgoing or incoming traffic of the ACL link, as indicated by the Flow_direction field. The Flow Specification command allows the Link Manager to have the parameters of the outgoing as well as the incoming flow for the ACL connection. The flow parameters are defined in the L2CAP specification “Quality of Service (QoS) Option” on page 60[vol. 3]. The Link Manager will determine if the flow parameters can be supported. Bluetooth devices that are both master and slave can use this command.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle used to identify for which ACL connection the Flow is specified. Range: 0x0000 - 0x0EFF (0x0F00 – 0x0FFF Reserved for future use)

Flags: *Size: 1 Octet*

Value	Parameter Description
0x00 – 0xFF	Reserved for Future Use.

Flow_direction: *Size: 1 Octet*

Value	Parameter Description
0x00	Outgoing Flow i.e. traffic send over the ACL connection
0x01	Incoming Flow i.e. traffic received over the ACL connection
0x02 – 0xFF	Reserved for Future Use.

Service_Type: *Size: 1 Octet*

Value	Parameter Description
0x00	No Traffic
0x01	Best Effort
0x02	Guaranteed
0x03 – 0xFF	Reserved for Future Use

Token Rate: *Size: 4 Octets*

Value	Parameter Description
0xXXXXXXXX	Token Rate in octets per second

Token Bucket Size: *Size: 4 Octets*

Value	Parameter Description
0xXXXXXXXX	Token Bucket Size in octets

Peak_Bandwidth: *Size: 4 Octets*

Value	Parameter Description
0xXXXXXXXX	Peak Bandwidth in octets per second

Access Latency: *Size: 4 Octets*

Value	Parameter Description
0xXXXXXXXX	Latency in microseconds

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the Flow Specification command, the Controller sends the Command Status event to the Host. When the Link Manager has determined if the Flow specification can be supported, the Controller on the local Bluetooth device sends a Flow Specification Complete event to the Host. The Flow Specification Complete event returned by the Controller on the local side contains the status of this command, and returned Flow parameters describing the supported QoS for the ACL connection.

Note: No Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, the Flow Specification Complete event will indicate that this command has been completed.

7.3 CONTROLLER & BASEBAND COMMANDS

The Controller & Baseband Commands provide access and control to various capabilities of the Bluetooth hardware. These parameters provide control of Bluetooth devices and of the capabilities of the Controller, Link Manager, and Baseband. The host device can use these commands to modify the behavior of the local device. For the HCI Control and Baseband Commands, the OGF is defined as 0x03.

7.3.1 Set Event Mask Command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Event_Mask	0x0001	Event_Mask	Status

Description:

The Set_Event_Mask command is used to control which events are generated by the HCI for the Host. If the bit in the Event_Mask is set to a one, then the event associated with that bit will be enabled. The Host has to deal with each event that occurs by the Bluetooth devices. The event mask allows the Host to control how much it is interrupted.

Command Parameters:

Event_Mask: *Size: 8 Octets*

Value	Parameter Description
0x0000000000000000	No events specified
0x0000000000000001	Inquiry Complete Event
0x0000000000000002	Inquiry Result Event
0x0000000000000004	Connection Complete Event
0x0000000000000008	Connection Request Event
0x0000000000000010	Disconnection Complete Event
0x0000000000000020	Authentication Complete Event
0x0000000000000040	Remote Name Request Complete Event
0x0000000000000080	Encryption Change Event
0x0000000000000100	Change Connection Link Key Complete Event
0x0000000000000200	Master Link Key Complete Event
0x0000000000000400	Read Remote Supported Features Complete Event
0x0000000000000800	Read Remote Version Information Complete Event
0x0000000000001000	QoS Setup Complete Event
0x0000000000002000	Reserved
0x0000000000004000	Reserved
0x0000000000008000	Hardware Error Event

Value	Parameter Description
0x0000000000010000	Flush Occurred Event
0x0000000000020000	Role Change Event
0x0000000000040000	Reserved
0x0000000000080000	Mode Change Event
0x0000000000100000	Return Link Keys Event
0x0000000000200000	PIN Code Request Event
0x0000000000400000	Link Key Request Event
0x0000000000800000	Link Key Notification Event
0x0000000001000000	Loopback Command Event
0x0000000002000000	Data Buffer Overflow Event
0x0000000004000000	Max Slots Change Event
0x0000000008000000	Read Clock Offset Complete Event
0x0000000010000000	Connection Packet Type Changed Event
0x0000000020000000	QoS Violation Event
0x0000000040000000	Page Scan Mode Change Event [deprecated]
0x0000000080000000	Page Scan Repetition Mode Change Event
0x0000000100000000	Flow Specification Complete Event
0x0000000200000000	Inquiry Result with RSSI Event
0x0000000400000000	Read Remote Extended Features Complete Event
0x0000000800000000	Reserved
0x0000001000000000	Reserved
0x0000002000000000	Reserved
0x0000004000000000	Reserved
0x0000008000000000	Reserved
0x0000010000000000	Reserved
0x0000020000000000	Reserved
0x0000040000000000	Reserved
0x0000080000000000	Synchronous Connection Complete Event
0x0000100000000000	Synchronous Connection Changed event
0xFFFFE00000000000	Reserved for future use
0x00001FFFFFFFFFFF	**Default** (All events enabled)

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Set_Event_Mask command succeeded.
0x01-0xFF	Set_Event_Mask command failed. See “Error Codes” on page 291 [Part D] for error codes and description.

Event(s) generated (unless masked away):

When the Set_Event_Mask command has completed, a Command Complete event will be generated.

7.3.2 Reset Command

Command	OCF	Command Parameters	Return Parameters
HCI_Reset	0x0003		Status

Description:

The Reset command will reset the Controller and the Link Manager. The reset command shall not affect the used HCI transport layer since the HCI transport layers may have reset mechanisms of their own. After the reset is completed, the current operational state will be lost, the Bluetooth device will enter standby mode and the Controller will automatically revert to the default values for the parameters for which default values are defined in the specification.

Note: the Host is not allowed to send additional HCI commands before the Command Complete event related to the Reset command has been received.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Reset command succeeded, was received and will be executed.
0x01-0xFF	Reset command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the reset has been performed, a Command Complete event will be generated.

7.3.3 Set Event Filter Command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Event_Filter	0x0005	Filter_Type, Filter_Condition_Type, Condition	Status

Description:

The Set_Event_Filter command is used by the Host to specify different event filters. The Host may issue this command multiple times to request various conditions for the same type of event filter and for different types of event filters. The event filters are used by the Host to specify items of interest, which allow the Controller to send only events which interest the Host. Only some of the events have event filters. By default (before this command has been issued after power-on or Reset) no filters are set, and the Auto_Accept_Flag is off (incoming connections are not automatically accepted). An event filter is added each time this command is sent from the Host and the Filter_Condition_Type is not equal to 0x00. (The old event filters will not be overwritten). To clear all event filters, the Filter_Type = 0x00 is used. The Auto_Accept_Flag will then be set to off. To clear event filters for only a certain Filter_Type, the Filter_Condition_Type = 0x00 is used.

The Inquiry Result filter allows the Controller to filter out Inquiry Result events. The Inquiry Result filter allows the Host to specify that the Controller only sends Inquiry Results to the Host if the Inquiry Result event meets one of the specified conditions set by the Host. For the Inquiry Result filter, the Host can specify one or more of the following Filter Condition Types:

1. Return responses from all devices during the Inquiry process
2. A device with a specific Class of Device responded to the Inquiry process
3. A device with a specific BD_ADDR responded to the Inquiry process

The Inquiry Result filter is used in conjunction with the Inquiry and Periodic Inquiry command.

The Connection Setup filter allows the Host to specify that the Controller only sends a Connection Complete or Connection Request event to the Host if the event meets one of the specified conditions set by the Host. For the Connection Setup filter, the Host can specify one or more of the following Filter Condition Types:

1. Allow Connections from all devices
2. Allow Connections from a device with a specific Class of Device
3. Allow Connections from a device with a specific BD_ADDR

For each of these conditions, an Auto_Accept_Flag parameter allows the Host to specify what action should be done when the condition is met. The Auto_Accept_Flag allows the Host to specify if the incoming connection should be auto accepted (in which case the Controller will send the Connection Complete event to the Host when the connection is completed) or if the Host should make the decision (in which case the Controller will send the Connection Request event to the Host, to elicit a decision on the connection).

The Connection Setup filter is used in conjunction with the Read/Write_Scan_Enable commands. If the local device is in the process of a page scan, and is paged by another device which meets one on the conditions set by the Host, and the Auto_Accept_Flag is off for this device, then a Connection Request event will be sent to the Host by the Controller. A Connection Complete event will be sent later on after the Host has responded to the incoming connection attempt. In this same example, if the Auto_Accept_Flag is on, then a Connection Complete event will be sent to the Host by the Controller. (No Connection Request event will be sent in that case.)

The Controller will store these filters in volatile memory until the Host clears the event filters using the Set_Event_Filter command or until the Reset command is issued. The number of event filters the Controller can store is implementation dependent. If the Host tries to set more filters than the Controller can store, the Controller will return the *Memory Full* error code and the filter will not be installed.
Note: the Clear All Filters has no Filter Condition Types or Conditions.
Note: In the condition that a connection is auto accepted, a Link Key Request event and possibly also a PIN Code Request event and a Link Key Notification event could be sent to the Host by the Controller before the Connection Complete event is sent.

If there is a contradiction between event filters, the latest set event filter will override older ones. An example is an incoming connection attempt where more than one Connection Setup filter matches the incoming connection attempt, but the Auto-Accept_Flag has different values in the different filters.

Command Parameters:

Filter_Type: *Size: 1 Octet*

Value	Parameter Description
0x00	Clear All Filters (Note: In this case, the Filter_Condition_type and Condition parameters should not be given, they should have a length of 0 octets. Filter_Type should be the only parameter.)
0x01	Inquiry Result.
0x02	Connection Setup.
0x03-0xFF	Reserved for Future Use.

Filter Condition Types: For each Filter Type one or more Filter Condition types exists.

Inquiry_Result_Filter_Condition_Type: *Size: 1 Octet*

Value	Parameter Description
0x00	Return responses from all devices during the Inquiry process. (Note: A device may be reported to the Host in an Inquiry Result event more than once during an inquiry or inquiry period depending on the implementation, see description in Section 7.1.1 on page 373 and Section 7.1.3 on page 376)
0x01	A device with a specific Class of Device responded to the Inquiry process.
0x02	A device with a specific BD_ADDR responded to the Inquiry process.
0x03-0xFF	Reserved for Future Use

Connection_Setup_Filter_Condition_Type: *Size: 1 Octet*

Value	Parameter Description
0x00	Allow Connections from all devices.
0x01	Allow Connections from a device with a specific Class of Device.
0x02	Allow Connections from a device with a specific BD_ADDR.
0x03-0xFF	Reserved for Future Use.

Condition: For each Filter Condition Type defined for the Inquiry Result Filter and the Connection Setup Filter, zero or more Condition parameters are required – depending on the filter condition type and filter type.

Condition for Inquiry_Result_Filter_Condition_Type = 0x00

Condition: *Size: 0 Octet*

Value	Parameter Description
	The Condition parameter is not used.

Condition for Inquiry_Result_Filter_Condition_Type = 0x01

Condition:

Size: 6 Octets

Class_of_Device: *Size: 3 Octets*

Value	Parameter Description
0x000000	Default, Return All Devices.
0xXXXXXX	*Class of Device* of Interest.

Class_of_Device_Mask: *Size: 3 Octets*

Value	Parameter Description
0xXXXXXX	Bit Mask used to determine which bits of the Class of Device parameter are ‘don’t care’. Zero-value bits in the mask indicate the ‘don’t care’ bits of the Class of Device.

Condition for Inquiry_Result_Filter_Condition_Type = 0x02

Condition:

Size: 6 Octets

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device of Interest

Condition for Connection_Setup_Filter_Condition_Type = 0x00

Condition:

Size: 1 Octet

Auto_Accept_Flag: *Size:1 Octet*

Value	Parameter Description
0x01	Do NOT Auto accept the connection. (Auto accept is off)
0x02	Do Auto accept the connection with role switch disabled. (Auto accept is on).
0x03	Do Auto accept the connection with role switch enabled. (Auto accept is on). Note: When auto accepting an incoming synchronous connection, no role switch will be performed. The value 0x03 of the Auto_Accept_Flag will then get the same effect as if the value had been 0x02.
0x04 – 0xFF	Reserved for future use.

Condition for Connection_Setup_Filter_Condition_Type = 0x01

Condition:

Size: 7 Octets

Class_of_Device: *Size: 3 Octets*

Value	Parameter Description
0x000000	Default, Return All Devices.
0xXXXXXX	*Class of Device* of Interest.

Class_of_Device_Mask: *Size: 3 Octets*

Value	Parameter Description
0xXXXXXX	Bit Mask used to determine which bits of the Class of Device parameter are 'don't care'. Zero-value bits in the mask indicate the 'don't care' bits of the Class of Device. Note: For an incoming SCO connection, if the class of device is unknown then the connection will be accepted.

Auto_Accept_Flag: *Size: 1 Octet*

Value	Parameter Description
0x01	Do NOT Auto accept the connection. (Auto accept is off)
0x02	Do Auto accept the connection with role switch disabled. (Auto accept is on).
0x03	Do Auto accept the connection with role switch enabled. (Auto accept is on). Note: When auto accepting an incoming synchronous connection, no role switch will be performed. The value 0x03 of the Auto_Accept_Flag will then get the same effect as if the value had been 0x02.
0x04 – 0xFF	Reserved for future use.

Condition for Connection_Setup_Filter_Condition_Type = 0x02

Condition:

Size: 7 Octets

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device of Interest.

Auto_Accept_Flag: *Size: 1 Octet*

Value	Parameter Description
0x01	Do NOT Auto accept the connection. (Auto accept is off)
0x02	Do Auto accept the connection with role switch disabled. (Auto accept is on).
0x03	Do Auto accept the connection with role switch enabled. (Auto accept is on). **Note: When auto accepting an incoming synchronous connection, no role switch will be performed. The value 0x03 of the Auto_Accept_Flag will then get the same effect as if the value had been 0x02**.
0x04 – 0xFF	Reserved for future use.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Set_Event_Filter command succeeded.
0x01-0xFF	Set_Event_Filter command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

A Command Complete event for this command will occur when the Controller has enabled the filtering of events. When one of the conditions are met, a specific event will occur.

7.3.4 Flush Command

Command	OCF	Command Parameters	Return Parameters
HCI_Flush	0x0008	Connection_Handle	Status, Connection_Handle

Description:

The Flush command is used to discard all data that is currently pending for transmission in the Controller for the specified connection handle, even if there currently are chunks of data that belong to more than one L2CAP packet in the Controller. After this, all data that is sent to the Controller for the same connection handle will be discarded by the Controller until an HCI Data Packet with the start Packet_Boundary_Flag (0x02) is received. When this happens, a new transmission attempt can be made. This command will allow higher-level software to control how long the baseband should try to retransmit a baseband packet for a connection handle before all data that is currently pending for transmission in the Controller should be flushed. Note that the Flush command

is used for ACL connections ONLY. In addition to the Flush command, the automatic flush timers (see section 7.3.31 on page 478) can be used to automatically flush the L2CAP packet that is currently being transmitted after the specified flush timer has expired.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify which connection to flush. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Flush command succeeded.
0x01-0xFF	Flush command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify which connection the flush command was issued on. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Event(s) generated (unless masked away):

The Flush Occurred event will occur once the flush is completed. A Flush Occurred event could be from an automatic Flush or could be cause by the Host issuing the Flush command. When the Flush command has completed, a Command Complete event will be generated, to indicate that the Host caused the Flush.

7.3.5 Read PIN Type Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_PIN_Type	0x0009		Status, PIN_Type

Description:

The Read PIN Type command is used to read the PIN_Type configuration parameter. See Section 6.13 on page 362.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_PIN_Type command succeeded.
0x01-0xFF	Read_PIN_Type command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

PIN_Type: *Size: 1 Octet*

Value	Parameter Description
0x00	Variable PIN.
0x01	Fixed PIN.

Event(s) generated (unless masked away):

When the Read_PIN_Type command has completed, a Command Complete event will be generated.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 208

7.3.6 Write PIN Type Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_PIN_Type	0x000A	PIN_Type	Status

Description:

The Write_PIN_Type command is used to write the PIN Type configuration paramter. See Section 6.13 on page 362.

Command Parameters:

PIN_Type: *Size: 1 Octet*

Value	Parameter Description
0x00	Variable PIN.
0x01	Fixed PIN.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write PIN Type command succeeded.
0x01-0xFF	Write PIN Type command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_PIN_Type command has completed, a Command Complete event will be generated.

7.3.7 Create New Unit Key Command

Command	OCF	Command Parameters	Return Parameters
HCI_Create_New_Unit_Key	0x000B		Status

Description:

The Create_New_Unit_Key command is used to create a new unit key. The Bluetooth hardware will generate a random seed that will be used to generate the new unit key. All new connection will use the new unit key, but the old unit key will still be used for all current connections.
Note: this command will not have any effect for a device which doesn't use unit keys (i.e. a device which uses only combination keys).

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Create New Unit Key command succeeded.
0x01-0xFF	Create New Unit Key command failed. See "Error Codes" on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Create_New_Unit_Key command has completed, a Command Complete event will be generated.

7.3.8 Read Stored Link Key Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Stored_ Link_Key	0x000D	BD_ADDR, Read_All_Flag	Status, Max_Num_Keys, Num_Keys_Read

Description:

The Read_Stored_Link_Key command provides the ability to read one or more link keys stored in the Bluetooth Controller. The Bluetooth Controller can store a limited number of link keys for other Bluetooth devices. Link keys are shared between two Bluetooth devices, and are used for all security transactions between the two devices. A Host device may have additional storage capabilities, which can be used to save additional link keys to be reloaded to the Bluetooth Controller when needed. The Read_All_Flag parameter is used to indicate if all of the stored Link Keys should be returned. If Read_All_Flag indicates that all Link Keys are to be returned, then the BD_ADDR command parameter must be ignored The BD_ADDR command parameter is used to identify which link key to read. The stored Link Keys are returned by one or more Return Link Keys events. See Section 6.14 on page 362.

Command Parameters:

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR for the stored link key to be read.

Read_All_Flag: *Size: 1 Octet*

Value	Parameter Description
0x00	Return Link Key for specified BD_ADDR.
0x01	Return all stored Link Keys.
0x02-0xFF	Reserved for future use.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Stored_Link_Key command succeeded.
0x01-0xFF	Read_Stored_Link_Key command failed.See “Error Codes” on page 291 [Part D] for error codes and description.

Max_Num_Keys: *Size: 2 Octets*

Value	Parameter Description
0xXXXX	Total Number of Link Keys that the Controller can store. Range: 0x0000 – 0xFFFF

Num_Keys_Read: *Size: 2 Octets*

Value	Parameter Description
0xXXXX	Number of Link Keys Read. Range: 0x0000 – 0xFFFF

Event(s) generated (unless masked away):

Zero or more instances of the Return Link Keys event will occur after the command is issued. When there are no link keys stored, no Return Link Keys events will be returned. When there are link keys stored, the number of link keys returned in each Return Link Keys event is implementation specific. When the Read Stored Link Key command has completed a Command Complete event will be generated.

7.3.9 Write Stored Link Key Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Stored_Link_Key	0x0011	Num_Keys_To_Write, *BD_ADDR*[i], Link_Key[i]	Status, Num_Keys_Written

Description:

The Write_Stored_Link_Key command provides the ability to write one or more link keys to be stored in the Bluetooth Controller. The Bluetooth Controller can store a limited number of link keys for other Bluetooth devices. If no additional space is available in the Bluetooth Controller then no additional link keys will be stored. If space is limited and if all the link keys to be stored will not fit in the limited space, then the order of the list of link keys without any error will determine which link keys are stored. Link keys at the beginning of the list will be stored first. The Num_Keys_Written parameter will return the number of link keys that were successfully stored. If no additional space is available, then the Host must delete one or more stored link keys before any additional link keys are stored. The link key replacement algorithm is implemented by the Host and not the Controller. Link keys are shared between two Bluetooth devices and are used for all security transactions between the two devices. A Host device may have additional storage capabilities, which can be used to save additional link keys to be reloaded to the

Bluetooth Controller when needed. See Section 6.14 on page 362.
Note: Link Keys are only stored by issuing this command.

Command Parameters:

Num_Keys_To_Write: *Size: 1 Octet*

Value	Parameter Description
0xXX	Number of Link Keys to Write. Range: 0x01 - 0x0B

BD_ADDR [i]: *Size: 6 Octets * Num_Keys_To_Write*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR for the associated Link Key.

Link_Key: *Size: 16 Octets*

Value	Parameter Description
0xXXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	Link Key for an associated BD_ADDR.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Stored_Link_Key command succeeded.
0x01-0xFF	Write_Stored_Link_Key command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Num_Keys_Written: *Size: 1 Octets*

Value	Parameter Description
0xXX	Number of Link Keys successfully written. Range: 0x00 – 0x0B

Event(s) generated (unless masked away):

When the Write_Stored_Link_Key command has completed, a Command Complete event will be generated.

7.3.10 Delete Stored Link Key Command

Command	OCF	Command Parameters	Return Parameters
HCI_Delete_Stored_ Link_Key	0x0012	BD_ADDR, Delete_All_Flag	Status, Num_Keys_Deleted

Description:

The Delete_Stored_Link_Key command provides the ability to remove one or more of the link keys stored in the Bluetooth Controller. The Bluetooth Controller can store a limited number of link keys for other Bluetooth devices. Link keys are shared between two Bluetooth devices and are used for all security transactions between the two devices. The Delete_All_Flag parameter is used to indicate if all of the stored Link Keys should be deleted. If the Delete_All_Flag indicates that all Link Keys are to be deleted, then the BD_ADDR command parameter must be ignored This command provides the ability to negate all security agreements between two devices. The BD_ADDR command parameter is used to identify which link key to delete. If a link key is currently in use for a connection, then the link key will be deleted when all of the connections are disconnected. See Section 6.14 on page 362.

Command Parameters:

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR for the link key to be deleted.

Delete_All_Flag: *Size: 1 Octet*

Value	Parameter Description
0x00	Delete only the Link Key for specified BD_ADDR.
0x01	Delete all stored Link Keys.
0x02-0xFF	Reserved for future use.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Delete_Stored_Link_Key command succeeded.
0x01-0xFF	Delete_Stored_Link_Key command failed. See “Error Codes” on page 291 [Part D] for error codes and description.

Num_Keys_Deleted: *Size: 2 Octets*

Value	Parameter Description
0xXXXX	Number of Link Keys Deleted

Event(s) generated (unless masked away):

When the Delete_Stored_Link_Key command has completed, a Command Complete event will be generated.

7.3.11 Write Local Name Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Local_Name	0x0013	Local Name	Status

Description:

The Write_Local_Name command provides the ability to modify the user-friendly name for the Bluetooth device. See Section 6.24 on page 368.

Command Parameters:

Local Name: *Size: 248 Octets*

Value	Parameter Description
	A UTF-8 encoded User-Friendly Descriptive Name for the device. If the name contained in the parameter is shorter than 248 octets, the end of the name is indicated by a NULL octet (0x00), and the following octets (to fill up 248 octets, which is the length of the parameter) do not have valid values.
	Null terminated Zero length String. **Default.**

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Local_Name command succeeded.
0x01-0xFF	Write_Local_Name command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Local_Name command has completed, a Command Complete event will be generated.

7.3.12 Read Local Name Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Local_Name	0x0014		Status, Local Name

Description:

The Read_Local_Name command provides the ability to read the stored user-friendly name for the Bluetooth device. See Section 6.24 on page 368.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Local_Name command succeeded
0x01-0xFF	Read_Local_Name command failed see “Error Codes” on page 291 [Part D] for list of Error Codes

Local Name: *Size: 248 Octets*

Value	Parameter Description
	A UTF-8 encoded User Friendly Descriptive Name for the device. If the name contained in the parameter is shorter than 248 octets, the end of the name is indicated by a NULL octet (0x00), and the following octets (to fill up 248 octets, which is the length of the parameter) do not have valid values.

Event(s) generated (unless masked away):

When the Read_Local_Name command has completed a Command Complete event will be generated.

7.3.13 Read Connection Accept Timeout Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Connection_ Accept_Timeout	0x0015		Status, Conn_Accept_Timeout

Description:

This command will read the value for the Connection_Accept_Timeout configuration parameter. See Section 6.7 on page 359.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Connection_Accept_Timeout command succeeded.
0x01-0xFF	Read_Connection_Accept_Timeout command failed. See "Error Codes" on page 291 [Part D] for list of Error Codes.

Conn_Accept_Timeout: *Size: 2 Octets*

Value	Parameter Description
N = 0xXXXX	Connection Accept Timeout measured in Number of Baseband slots. Interval Length = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xB540 Time Range: 0.625 msec -29 seconds

Event(s) generated (unless masked away):

When the Read_Connection_Timeout command has completed, a Command Complete event will be generated.

7.3.14 Write Connection Accept Timeout Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Connection_ Accept_Timeout	0x0016	Conn_Accept_Timeout	Status

Description:

This command will write the value for the Connection_Accept_Timeout configuration parameter. See Section 6.7 on page 359.

Command Parameters:

Conn_Accept_Timeout: *Size: 2 Octets*

Value	Parameter Description
N = 0xXXXX	Connection Accept Timeout measured in Number of Baseband slots. Interval Length = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xB540 Time Range: 0.625 msec - 29 seconds Default: N = 0x1FA0 Time = 5.06 Sec Mandatory Range for Controller: 0x00A0 to 0xB540

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Connection_Accept_Timeout command succeeded.
0x01-0xFF	Write_Connection_Accept_Timeout command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Connection_Accept_Timeout command has completed, a Command Complete event will be generated.

7.3.15 Read Page Timeout Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Page_Timeout	0x0017		Status, Page_Timeout

Description:

This command will read the value for the Page_Timeout configuration parameter. See Section 6.6 on page 359.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Page_Timeout command succeeded.
0x01-0xFF	Read_Page_Timeout command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Page_Timeout: *Size: 2 Octets*

Value	Parameter Description
N = 0xXXXX	Page Timeout measured in Number of Baseband slots. Interval Length = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625 msec -40.9 Seconds

Event(s) generated (unless masked away):

When the Read_Page_Timeout command has completed, a Command Complete event will be generated.

7.3.16 Write Page Timeout Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Page_Timeout	0x0018	Page_Timeout	Status

Description:

This command will write the value for the Page_Timeout configuration parameter. The Page_Timeout configuration parameter defines the maximum time the local Link Manager will wait for a baseband page response from the remote device at a locally initiated connection attempt. If this time expires and the remote device has not responded to the page at baseband level, the connection attempt will be considered to have failed.

Command Parameters:

Page_Timeout: *Size: 2 Octets*

Value	Parameter Description
0	Illegal Page Timeout. Must be larger than 0.
N = 0xXXXX	Page Timeout measured in Number of Baseband slots. Interval Length = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625 msec -40.9 Seconds **Default: N = 0x2000 Time = 5.12 Sec** Mandatory Range for Controller: 0x0016 to 0xFFFF

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Page_Timeout command succeeded.
0x01-0xFF	Write_Page_Timeout command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Page_Timeout command has completed, a Command Complete event will be generated.

7.3.17 Read Scan Enable Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Scan_Enable	0x0019		Status, Scan_Enable

Description:

This command will read the value for the Scan_Enable parameter configuration parameter. See Section 6.1 on page 357.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Scan_Enable command succeeded.
0x01-0xFF	Read_Scan_Enable command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Scan_Enable: *Size: 1 Octet*

Value	Parameter Description
0x00	No Scans enabled.
0x01	Inquiry Scan enabled. Page Scan disabled.
0x02	Inquiry Scan disabled. Page Scan enabled.
0x03	Inquiry Scan enabled. Page Scan enabled.
0x04-0xFF	Reserved

Event(s) generated (unless masked away):

When the Read_Scan_Enable command has completed, a Command Complete event will be generated.

7.3.18 Write Scan Enable Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Scan_Enable	0x001A	Scan_Enable	Status

Description:

This command will write the value for the Scan_Enable configuration parameter. See Section 6.1 on page 357.

Command Parameters:

Scan_Enable: *Size: 1 Octet*

Value	Parameter Description
0x00	No Scans enabled. **Default.**
0x01	Inquiry Scan enabled. Page Scan disabled.
0x02	Inquiry Scan disabled. Page Scan enabled.
0x03	Inquiry Scan enabled. Page Scan enabled.
0x04-0xFF	Reserved

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Scan_Enable command succeeded.
0x01-0xFF	Write_Scan_Enable command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Scan_Enable command has completed, a Command Complete event will be generated.

7.3.19 Read Page Scan Activity Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Page_Scan_ Activity	0x001B		Status, Page_Scan_Interval, Page_Scan_Window

Description:

This command will read the value for Page_Scan_Interval and Page_Scan_Window configuration parameters. See Section 6.8 on page 360 and Section 6.9 on page 360.
Note: Page Scan is only performed when Page_Scan is enabled (see 6.1, 7.3.17 and 7.3.18).

A changed Page_Scan_Interval could change the local Page_Scan_ Repetition_Mode (see “Baseband Specification” on page 45 [Part B], Keyword: SR-Mode).

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Page_Scan_Activity command succeeded.
0x01-0xFF	Read_Page_Scan_Activity command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Page_Scan_Interval: *Size: 2 Octets*

Value	Parameter Description
N = 0xXXXX	Size: 2 Octets Range: 0x0012 – 0x1000 Time = N * 0.625 msec Range: 11.25 msec - 2560 msec; only even values are valid

Page_Scan_Window: *Size: 2 Octets*

Value	Parameter Description
N = 0xXXXX	Size: 2 Octets Range: 0x0011 - 0x1000 Time = N * 0.625 msec Range: 10.625 msec - 2560 msec

Event(s) generated (unless masked away):

When the Read_Page_Scan_Activity command has completed, a Command Complete event will be generated.

7.3.20 Write Page Scan Activity Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Page_Scan_ Activity	0x001C	Page_Scan_Interval, Page_Scan_Window	Status

Description:

This command will write the values for the Page_Scan_Interval and Page_Scan_Window configuration parameters. The Page_Scan_Window shall be less than or equal to the Page_Scan_Interval. See Section 6.8 on page 360 and Section 6.9 on page 360.
Note: Page Scan is only performed when Page_Scan is enabled (see 6.1, 7.3.17 and 7.3.18). A changed Page_Scan_Interval could change the local Page_Scan_Repetition_Mode (see Baseband Specification, Section 8.3.1, on page 134).

Command Parameters:

Page_Scan_Interval: *Size: 2 Octets*

Value	Parameter Description
See Section 6.8 on page 360	

Page_Scan_Window: *Size: 2 Octets*

Value	Parameter Description
See Section 6.9 on page 360	

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Page_Scan_Activity command succeeded.
0x01-0xFF	Write_Page_Scan_Activity command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Page_Scan_Activity command has completed, a Command Complete event will be generated.

7.3.21 Read Inquiry Scan Activity Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_ Inquiry_Scan_Activity	0x001D		Status, Inquiry_Scan_Interval, Inquiry_Scan_Window

Description:

This command will read the value for Inquiry_Scan_Interval and Inquiry_Scan_Window configuration parameter. See Section 6.2 on page 357 and Section 6.3 on page 358.
Note: Inquiry Scan is only performed when Inquiry_Scan is enabled see 6.1, 7.3.17 and 7.3.18).

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Inquiry_Scan_Activity command succeeded.
0x01-0xFF	Read_Inquiry_Scan_Activity command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Inquiry_Scan_Interval: *Size: 2 Octets*

Value	Parameter Description
N = 0xXXXX	Size: 2 Octets Range: 0x0012 – 0x1000 Time = N * 0.625 msec Range: 11.25 - 2560 msec; only even values are valid

Inquiry_Scan_Window: *Size: 2 Octets*

Value	Parameter Description
N = 0xXXXX	Size: 2 Octets Range: 0x0011 - 0x1000 Time = N * 0.625 msec Range: 10.625 msec - 2560 msec

Event(s) generated (unless masked away):

When the Read_Inquiry_Scan_Activity command has completed, a Command Complete event will be generated.

7.3.22 Write Inquiry Scan Activity Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Inquiry_ Scan_Activity	0x001E	Inquiry_Scan_Interval, Inquiry_Scan_Window	Status

Description:

This command will write the values for the Inquiry_Scan_Interval and Inquiry_Scan_Window configuration parameters. The Inquiry_Scan_Window shall be less than or equal to the Inquiry_Scan_Interval. See Section 6.2 on page 357 and Section 6.3 on page 358.
Note: Inquiry Scan is only performed when Inquiry_Scan is enabled (see 6.1, 7.3.17 and 7.3.18).

Command Parameters:

Inquiry_Scan_Interval: *Size: 2 Octets*

Value	Parameter Description
See Section 6.2 on page 357.	

Inquiry_Scan_Window: *Size: 2 Octets*

Value	Parameter Description
See Section 6.3 on page 358.	

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Inquiry_Scan_Activity command succeeded.
0x01-0xFF	Write_Inquiry_Scan_Activity command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Inquiry_Scan_Activity command has completed, a Command Complete event will be generated.

7.3.23 Read Authentication Enable Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_ Authentication_Enable	0x001F		Status, Authentication_Enable

Description:

This command will read the value for the Authentication_Enable configuration parameter. See Section 6.15 on page 362.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Authentication_Enable command succeeded.
0x01-0xFF	Read_Authentication_Enable command failed.See “Error Codes” on page 291 [Part D] for list of Error Codes.

Authentication_Enable: *Size: 1 Octet*

Value	Parameter Description
0x00	Authentication not required.
0x01	Authentication required for all connections.
0x02-0xFF	Reserved

Event(s) generated (unless masked away):

When the Read_Authentication_Enable command has completed, a Command Complete event will be generated.

7.3.24 Write Authentication Enable Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_ Authentication_Enable	0x0020	Authentication_Enable	Status

Description:

This command will write the value for the Authentication_Enable configuration parameter. See Section 6.15 on page 362.

Command Parameters:

Authentication_Enable: *Size: 1 Octet*

Value	Parameter Description
0x00	Authentication not required. **Default.**
0x01	Authentication required for all connections.
0x02-0xFF	Reserved

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write Authentication_Enable command succeeded.
0x01-0xFF	Write Authentication_Enable command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Authentication_Enable command has completed, a Command Complete event will be generated.

7.3.25 Read Encryption Mode Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Encryption_Mode	0x0021		Status, Encryption_Mode

Description:

This command will read the value for the Encryption_Mode configuration parameter. See Section 6.16 on page 363.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Encryption_Mode command succeeded.
0x01-0xFF	Read_Encryption_Mode command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Encryption_Mode: *Size: 1 Octet*

Value	Parameter Description
See Section 6.16 on page 363	

Event(s) generated (unless masked away):

When the Read_Encryption_Mode command has completed, a Command Complete event will be generated.

7.3.26 Write Encryption Mode Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Encryption_Mode	0x0022	Encryption_Mode	Status

Description:

This command will write the value for the Encryption_Mode configuration parameter. See Section 6.16 on page 363.

Command Parameters:

Encryption_Mode: *Size: 1 Octet*

Value	Parameter Description
See Section 6.16 on page 363.	

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Encryption_Mode command succeeded.
0x01-0xFF	Write_Encryption_Mode command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Encryption_Mode command has completed, a Command Complete event will be generated.

7.3.27 Read Class of Device Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Class_of_Device	0x0023		Status, Class_of_Device

Description:

This command will read the value for the Class_of_Device parameter.
See Section 6.25 on page 368.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Class_of_Device command succeeded.
0x01-0xFF	Read_Class_of_Device command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Class_of_Device: *Size: 3 Octets*

Value	Parameter Description
0xXXXXXX	Class of Device for the device.

Event(s) generated (unless masked away):

When the Read_Class_of_Device command has completed, a Command Complete event will be generated.

7.3.28 Write Class of Device Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Class_of_Device	0x0024	Class_of_Device	Status

Description:

This command will write the value for the Class_of_Device parameter.
See Section 6.25 on page 368.

Command Parameters:

Class_of_Device: *Size: 3 Octets*

Value	Parameter Description
0xXXXXXX	Class of Device for the device.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Class_of_Device command succeeded.
0x01-0xFF	Write_Class_of_Device command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Class_of_Device command has completed, a Command Complete event will be generated.

7.3.29 Read Voice Setting Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Voice_Setting	0x0025		Status, Voice_Setting

Description:

This command will read the values for the Voice_Setting configuration parameter. See Section 6.12 on page 361.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Voice_Setting command succeeded.
0x01-0xFF	Read_Voice_Setting command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Voice_Setting: *Size: 2 Octets (10 Bits meaningful)*

Value	Parameter Description
See Section 6.12 on page 361.	

Event(s) generated (unless masked away):

When the Read_Voice_Setting command has completed, a Command Complete event will be generated.

7.3.30 Write Voice Setting Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Voice_Setting	0x0026	Voice_Setting	Status

Description:

This command will write the values for the Voice_Setting configuration parameter. See Section 6.12 on page 361.

Command Parameters:

Voice_Setting: *Size: 2 Octets (10 Bits meaningful)*

Value	Parameter Description
See Section 6.12 on page 361.	

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Voice_Setting command succeeded.
0x01-0xFF	Write_Voice_Setting command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Voice_Setting command has completed, a Command Complete event will be generated.

7.3.31 Read Automatic Flush Timeout Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Automatic_Flush_Timeout	0x0027	Connection_Handle	Status, Connection_Handle, Flush_Timeout

Description:

This command will read the value for the Flush_Timeout parameter for the specified connection handle. See Section 6.20 on page 366.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Flush Timeout to read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Automatic_Flush_Timeout command succeeded.
0x01-0xFF	Read_Automatic_Flush_Timeout command failed. See "Error Codes" on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Flush Timeout has been read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Flush_Timeout: *Size: 2 Octets*

Value	Parameter Description
0	Timeout = ∞; No Automatic Flush
N = 0xXXXX	Flush Timeout = N * 0.625 msec Size: 11 bits Range: 0x0001 – 0x07FF

Event(s) generated (unless masked away):

When the Read_Automatic_Flush_Timeout command has completed, a Command Complete event will be generated.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 236

7.3.32 Write Automatic Flush Timeout Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Automatic_Flush_Timeout	0x0028	Connection_Handle, Flush_Timeout	Status, Connection_Handle

Description:

This command will write the value for the Flush_Timeout parameter for the specified connection handle. See Section 6.20 on page 366.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Flush Timeout to write to. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Flush_Timeout: *Size: 2 Octets*

Value	Parameter Description
0	Timeout = ∞; No Automatic Flush. **Default.**
N = 0xXXXX	Flush Timeout = N * 0.625 msec Size: 11 bits Range: 0x0001 – 0x07FF Mandatory Range for Controller: 0x0002 to 0x07FF

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Automatic_Flush_Timeout command succeeded.
0x01-0xFF	Write_Automatic_Flush_Timeout command failed. See "Error Codes" on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Flush Timeout has been written. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Event(s) generated (unless masked away):

When the Write_Automatic_Flush_Timeout command has completed, a Command Complete event will be generated.

7.3.33 Read Num Broadcast Retransmissions Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Num_Broadcast_ Retransmissions	0x0029		Status, Num_Broadcast_Retransmissions

Description:

This command will read the device's parameter value for the Number of Broadcast Retransmissions. See Section 6.21 on page 366

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Num_Broadcast_Retransmissions command succeeded.
0x01-0xFF	Read_Num_Broadcast_Retransmissions command failed. See "Error Codes" on page 291 [Part D] for list of Error Codes.

Num_Broadcast_Retransmissions: *Size: 1 Octet*

Value	Parameter Description
See Section 6.21 on page 366.	

Event(s) generated (unless masked away):

When the Read_Num_Broadcast_Retransmission command has completed, a Command Complete event will be generated.

7.3.34 Write Num Broadcast Retransmissions Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Num_Broadcast_ Retransmissions	0x002A	Num_Broadcast_ Retransmissions	Status

Description:

This command will write the device's parameter value for the Number of Broadcast Retransmissions. See Section 6.21 on page 366.

Command Parameters:

Num_Broadcast_Retransmissions: *Size: 1 Octet*

Value	Parameter Description
See Section 6.21 on page 366.	

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Num_Broadcast_Retransmissions command succeeded.
0x01-0xFF	Write_Num_Broadcast_Retransmissions command failed. See "Error Codes" on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Num_Broadcast_Retransmissions command has completed, a Command Complete event will be generated.

7.3.35 Read Hold Mode Activity Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Hold_Mode_Activity	0x002B		Status, Hold_Mode_Activity

Description:

This command will read the value for the Hold_Mode_Activity parameter. See Section 6.18 on page 364.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Hold_Mode_Activity command succeeded.
0x01-0xFF	Read_Hold_Mode_Activity command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Hold_Mode_Activity: *Size: 1 Octet*

Value	Parameter Description
0x00	Maintain current Power State.
0x01	Suspend Page Scan.
0x02	Suspend Inquiry Scan.
0x04	Suspend Periodic Inquiries.
0x08-0xFF	Reserved for Future Use.

Event(s) generated (unless masked away):

When the Read_Hold_Mode_Activity command has completed, a Command Complete event will be generated.

7.3.36 Write Hold Mode Activity Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Hold_Mode_Activity	0x002C	Hold_Mode_Activity	Status

Description:

This command will write the value for the Hold_Mode_Activity parameter. See Section 6.18 on page 364.

Command Parameters:

Hold_Mode_Activity: *Size: 1 Octet*

Value	Parameter Description
0x00	Maintain current Power State. **Default.**
0x01	Suspend Page Scan.
0x02	Suspend Inquiry Scan.
0x04	Suspend Periodic Inquiries.
0x08-0xFF	Reserved for Future Use.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Hold_Mode_Activity command succeeded.
0x01-0xFF	Write_Hold_Mode_Activity command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Hold_Mode_Activity command has completed, a Command Complete event will be generated.

7.3.37 Read Transmit Power Level Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Transmit_ Power_Level	0x002D	Connection_Handle, Type	Status, Connection_Handle, Transmit_Power_Level

Description:

This command will read the values for the Transmit_Power_Level parameter for the specified Connection Handle. The Connection_Handle must be a Connection_Handle for an ACL connection.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Transmit Power Level setting to read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Type: *Size: 1 Octet*

Value	Parameter Description
0x00	Read Current Transmit Power Level.
0x01	Read Maximum Transmit Power Level.
0x02-0xFF	Reserved

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Transmit_Power_Level command succeeded.
0x01-0xFF	Read_Transmit_Power_Level command failed. See "Error Codes" on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Transmit Power Level setting is returned. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Petitioner Kiosoft Exhibit 1011 Part 2
Page 242

Transmit_Power_Level: *Size: 1 Octet*

Value	Parameter Description
N = 0xXX	Size: 1 Octet (signed integer) Range: -30 ≤ N ≤ 20 Units: dBm

Event(s) generated (unless masked away):

When the Read_Transmit_Power_Level command has completed, a Command Complete event will be generated.

7.3.38 Read Synchronous Flow Control Enable Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_ Synchronous_Flow_ Control_Enable	0x002E		Status, Synchronous_Flow_ Control_Enable

Description:

The Read_Synchronous_Flow_Control_Enable command provides the ability to read the Synchronous_Flow_Control_Enable setting. See Section 6.23 on page 367.
Note: the Synchronous_Flow_Control_Enable setting can only be changed if no connections exist.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Synchronous_Flow_Control_Enable command succeeded
0x01-0xFF	Read_Synchronous_Flow_Control_Enable command failed see "Error Codes" on page 291 [Part D] for list of Error Codes

Synchronous_Flow_Control_Enable: *Size: 1 Octet*

Value	Parameter Description
0x00	Synchronous Flow Control is disabled. No Number of Completed Packets events will be sent from the Controller for Synchronous Connection Handles.
0x01	Synchronous Flow Control is enabled. Number of Completed Packets events will be sent from the Controller for Synchronous Connection Handles.

Event(s) generated (unless masked away):

When the Read_Synchronous_Flow_Control_Enable command has completed a Command Complete event will be generated.

7.3.39 Write Synchronous Flow Control Enable Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Synchronous_ Flow_Control_Enable	0x002F	Synchronous_Flow_ Control_Enable	Status

Description:

The Write_Synchronous_Flow_Control_Enable command provides the ability to write the Synchronous_Flow_Control_Enable setting. See Section 6.23 on page 367.
Note: the Synchronous_Flow_Control_Enable setting can only be changed if no connections exist.

Command Parameters:

Synchronous_Flow_Control_Enable: *Size: 1 Octet*

Value	Parameter Description
0x00	Synchronous Flow Control is disabled. No Number of Completed Packets events will be sent from the Controller for synchronous Connection Handles. **Default**
0x01	Synchronous Flow Control is enabled. Number of Completed Packets events will be sent from the Controller for synchronous Connection Handles.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Synchronous_Flow_Control_Enable command succeeded
0x01-0xFF	Write_Synchronous_Flow_Control_Enable command failed see “Error Codes” on page 291 [Part D] for list of Error Codes

Event(s) generated (unless masked away):

When the Write_Synchronous_Flow_Control_Enable command has completed a Command Complete event will be generated.

7.3.40 Set Controller To Host Flow Control Command

Command	OCF	Command Parameters	Return Parameters
HCI_Set_Controller_To_Host_Flow_Control	0x0031	Flow_Control_Enable	Status

Description:

This command is used by the Host to turn flow control on or off for data and/or voice sent in the direction from the Controller to the Host. If flow control is turned off, the Host should not send the Host_Number_Of_Completed_Packets command. That command will be ignored by the Controller if it is sent by the Host and flow control is off. If flow control is turned on for HCI ACL Data Packets and off for HCI synchronous Data Packets, Host_Number_Of_Completed_Packets commands sent by the Host should only contain Connection Handles for ACL connections. If flow control is turned off for HCI ACL Data Packets and on for HCI synchronous Data Packets, Host_Number_Of_Completed_Packets commands sent by the Host should only contain Connection Handles for synchronous connections. If flow control is turned on for HCI ACL Data Packets and HCI synchronous Data Packets, the Host will send Host_Number_Of_Completed_Packets commands both for ACL connections and synchronous connections.

The Flow_Control_Enable setting shall only be changed if no connections exist.

Command Parameters:

Flow_Control_Enable: *Size: 1 Octet*

Value	Parameter Description
0x00	Flow control **off** in direction from Controller to Host. **Default.**
0x01	Flow control **on** for HCI ACL Data Packets and **off** for HCI synchronous Data Packets in direction from Controller to Host.
0x02	Flow control **off** for HCI ACL Data Packets and **on** for HCI synchronous Data Packets in direction from Controller to Host.
0x03	Flow control **on** both for HCI ACL Data Packets and HCI synchronous Data Packets in direction from Controller to Host.
0x04-0xFF	Reserved

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Set_Controller_To_Host_Flow_Control command succeeded.
0x01-0xFF	Set_Controller_To_Host_Flow_Control command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Set_Controller_To_Host_Flow_Control command has completed, a Command Complete event will be generated.

7.3.41 Host Buffer Size Command

Command	OCF	Command Parameters	Return Parameters
HCI_Host_Buffer_Size	0x0033	Host_ACL_Data_Packet_Length, Host_Synchronous_Data_Packet_Length, Host_Total_Num_ACL_Data_Packets, Host_Total_Num_Synchronous_Data_Packets	Status

Description:

The Host_Buffer_Size command is used by the Host to notify the Controller about the maximum size of the data portion of HCI ACL and synchronous Data Packets sent from the Controller to the Host. The Controller will segment the data to be transmitted from the Controller to the Host according to these sizes, so that the HCI Data Packets will contain data with up to these sizes. The Host_Buffer_Size command also notifies the Controller about the total number of HCI ACL and synchronous Data Packets that can be stored in the data buffers of the Host. If flow control from the Controller to the Host is turned off, and the Host_Buffer_Size command has not been issued by the Host, this means that the Controller will send HCI Data Packets to the Host with any lengths the Controller wants to use, and it is assumed that the data buffer sizes of the Host are unlimited. If flow control from the Controller to the Host is turned on, the Host_Buffer_Size command must after a power-on or a reset always be sent by the Host before the first Host_Number_Of_Completed_Packets command is sent.

(The Set Controller To Host Flow Control Command command is used to turn flow control on or off.) The Host_ACL_Data_Packet_Length command parameter will be used to determine the size of the L2CAP segments contained in ACL Data Packets, which are transferred from the Controller to the Host. The Host_Synchronous_Data_Packet_Length command parameter is used to determine the maximum size of HCI synchronous Data Packets. Both the Host and the Controller must support command and event packets, where the data portion (excluding header) contained in the packets is 255 octets in size.

The Host_Total_Num_ACL_Data_Packets command parameter contains the total number of HCI ACL Data Packets that can be stored in the data buffers of the Host. The Controller will determine how the buffers are to be divided between different Connection Handles. The Host_Total_Num_Synchronous_Data_Packets command parameter gives the same information for HCI synchronous Data Packets.
Note: the Host_ACL_Data_Packet_Length and Host_Synchronous_Data_Packet_Length command parameters do not include the length of the HCI Data Packet header.

Command Parameters:

Host_ACL_Data_Packet_Length: *Size: 2 Octets*

Value	Parameter Description
0xXXXX	Maximum length (in octets) of the data portion of each HCI ACL Data Packet that the Host is able to accept.

Host_Synchronous_Data_Packet_Length: *Size: 1 Octet*

Value	Parameter Description
0xXX	Maximum length (in octets) of the data portion of each HCI synchronous Data Packet that the Host is able to accept.

Host_Total_Num_ACL_Data_Packets: *Size: 2 Octets*

Value	Parameter Description
0xXXXX	Total number of HCI ACL Data Packets that can be stored in the data buffers of the Host.

Host_Total_Num_Synchronous_Data_Packets: *Size: 2 Octets*

Value	Parameter Description
0xXXXX	Total number of HCI synchronous Data Packets that can be stored in the data buffers of the Host.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Host_Buffer_Size command succeeded.
0x01-0xFF	Host_Buffer_Size command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Host_Buffer_Size command has completed, a Command Complete event will be generated.

7.3.42 Host Number Of Completed Packets Command

Command	OCF	Command Parameters	Return Parameters
HCI_Host_Number_Of_ Completed_Packets	0x0035	Number_Of_Handles, Connection_ Handle[i], Host_Num_Of_Completed_Packets [i]	

Description:

The Host_Number_Of_Completed_Packets command is used by the Host to indicate to the Controller the number of HCI Data Packets that have been completed for each Connection Handle since the previous Host_Number_Of_ Completed_Packetscommand was sent to the Controller. This means that the corresponding buffer space has been freed in the Host. Based on this information, and the Host_Total_Num_ACL_Data_Packets and Host_Total_Num_ Synchronous_Data_Packets command parameters of the Host_Buffer_Size command, the Controller can determine for which Connection Handles the following HCI Data Packets should be sent to the Host. The command should only be issued by the Host if flow control in the direction from the Controller to the Host is on and there is at least one connection, or if the Controller is in local loopback mode. Otherwise, the command will be ignored by the Controller. When the Host has completed one or more HCI Data Packet(s) it shall send a Host_Number_Of_Completed_Packets command to the Controller, until it finally reports that all pending HCI Data Packets have been completed. The frequency at which this command is sent is manufacturer specific.

(The Set Controller To Host Flow Control Command command is used to turn flow control on or off.) If flow control from the Controller to the Host is turned on, the Host_Buffer_Size command must after a power-on or a reset always be sent by the Host before the first Host_Number_Of_Completed_Packets command is sent.
Note: the Host_Number_Of_Completed_Packets command is a special command in the sense that no event is normally generated after the command has completed. The command may be sent at any time by the Host when there is at least one connection, or if the Controller is in local loopback mode independent of other commands. The normal flow control for commands is not used for the Host_Number_Of_Completed_Packets command.

Command Parameters:

Number_Of_Handles: *Size: 1 Octet*

Value	Parameter Description
0xXX	The number of Connection Handles and Host_Num_Of_Completed_Packets parameters pairs contained in this command. Range: 0-255

*Connection_Handle[i]: Size: Number_Of_Handles*2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

*Host_Num_Of_Completed_Packets [i]: Size: Number_Of_Handles * 2 Octets*

Value	Parameter Description
N = 0xXXXX	The number of HCI Data Packets that have been completed for the associated Connection Handle since the previous time the event was returned. Range for N: 0x0000-0xFFFF

Return Parameters:

None.

Event(s) generated (unless masked away):

Normally, no event is generated after the Host_Number_Of_Completed_ Packets command has completed. However, if the Host_Number_Of_ Completed_Packets command contains one or more invalid parameters, the Controller will return a Command Complete event with a failure status indicating the *Invalid HCI Command Parameters error code*. The Host may send the Host_Number_Of_Completed_Packets command at any time when there is at least one connection, or if the Controller is in local loopback mode. The normal flow control for commands is not used for this command.

7.3.43 Read Link Supervision Timeout Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Link_Supervision_ Timeout	0x0036	Connection_Handle	Status, Connection_Handle, Link_Supervision_ Timeout

Description:

This command will read the value for the Link_Supervision_Timeout parameter for the device.
Note: the Connection_Handle used for this command must be the ACL connection to the appropriate device. See Section 6.22 on page 367.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Link Supervision Timeout value is to be read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Link_Supervision_Timeout command succeeded.
0x01-0xFF	Read_Link_Supervision_Timeout command failed. See "Error Codes" on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Link Supervision Timeout value was read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Link_Supervision_Timeout: *Size: 2 Octets*

Value	Parameter Description
0x0000	No Link_Supervision_Timeout.
N = 0xXXXX	Measured in Number of Baseband slots Link_Supervision_Timeout = N * 0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625ms - 40.9 sec

Event(s) generated (unless masked away):

When the Read_Link_Supervision_Timeout command has completed, a Command Complete event will be generated.

7.3.44 Write Link Supervision Timeout Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Link_Supervision_Timeout	0x0037	Connection_Handle, Link_Supervision_Timeout	Status, Connection_Handle

Description:

This command will write the value for the Link_Supervision_Timeout parameter for the device.
Note: the Connection_Handle used for this command must be the ACL connection to the appropriate device. This command will set the Link_Supervision_Timeout values for other Synchronous Connection_Handles to that device.
See Section 6.22 on page 367.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle's Link Supervision Timeout value is to be written. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Link_Supervision_Timeout: *Size: 2 Octets*

Value	Parameter Description
0x0000	No Link_Supervision_Timeout.
N = 0xXXXX	Measured in Number of Baseband slots Link_Supervision_Timeout = N*0.625 msec (1 Baseband slot) Range for N: 0x0001 – 0xFFFF Time Range: 0.625ms – 40.9 sec **Default**: N = 0x7D00 Link_Supervision_Timeout = 20 sec Mandatory Range for Controller: 0x0190 to 0xFFFF; plus 0 for infinite timeout

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Link_Supervision_Timeout command succeeded.
0x01-0xFF	Write_Link_Supervision_Timeout command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle’s Link Supervision Timeout value was written. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Event(s) generated (unless masked away):

When the Write_Link_Supervision_Timeout command has completed, a Command Complete event will be generated.

7.3.45 Read Number Of Supported IAC Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Number_Of_Supported_IAC	0x0038		Status, Num_Support_IAC

Description:

This command will read the value for the number of Inquiry Access Codes (IAC) that the local Bluetooth device can simultaneous listen for during an Inquiry Scan. All Bluetooth devices are required to support at least one IAC, the General Inquiry Access Code (the GIAC). Some Bluetooth devices support additional IACs.

Command Parameters:

None

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Number_Of_Supported_IAC command succeeded.
0x01-0xFF	Read_Number_Of_Supported_IAC command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Num_Support_IAC *Size: 1 Octet*

Value	Parameter Description
0xXX	Specifies the number of Supported IAC that the local Bluetooth device can simultaneous listen for during an Inquiry Scan. Range: 0x01-0x40

Event(s) generated (unless masked away):

When the Read_Number_Of_Supported_IAC command has completed, a Command Complete event will be generated.

7.3.46 Read Current IAC LAP Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Current_IAC_LAP	0x0039		Status, Num_Current_IAC, IAC_LAP[i]

Description:

This command reads the LAP(s) used to create the Inquiry Access Codes (IAC) that the local Bluetooth device is simultaneously scanning for during Inquiry Scans. All Bluetooth devices are required to support at least one IAC, the General Inquiry Access Code (the GIAC). Some Bluetooth devices support additional IACs.

Command Parameters:

None

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Current_IAC_LAP command succeeded.
0x01-0xFF	Read_Current_IAC_LAP command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Num_Current_IAC *Size: 1 Octet*

Value	Parameter Description
0xXX	Specifies the number of IACs which are currently in use by the local Bluetooth device to simultaneously listen for during an Inquiry Scan. Range: 0x01-0x40

IAC_LAP[i] *Size: 3 Octets * Num_Current_IAC*

Value	Parameter Description
0xXXXXXX	LAPs used to create the IAC which is currently in use by the local Bluetooth device to simultaneously listen for during an Inquiry Scan. Range: 0x9E8B00-0x9E8B3F

Event(s) generated (unless masked away):

When the Read_Current_IAC_LAP command has completed, a Command Complete event will be generated.

7.3.47 Write Current IAC LAP Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Current_IAC_LAP	0x003A	Num_Current_IAC, IAC_LAP[i]	Status

Description:

This command writes the LAP(s) used to create the Inquiry Access Codes (IAC) that the local Bluetooth device is simultaneously scanning for during Inquiry Scans. All Bluetooth devices are required to support at least one IAC, the General Inquiry Access Code (the GIAC). Some Bluetooth devices support additional IACs.

This command shall clear any existing IACs and stores Num_Current_IAC and the IAC_LAPs in to the controller. If Num_Current_IAC is greater than Num_Support_IAC then only the first Num_Support_IACs shall be stored in the controller, and a Command Complete event with error code *Success (0x00)* shall be generated.

Command Parameters:

Num_Current_IAC *Size: 1 Octet*

Value	Parameter Description
0xXX	Specifies the number of IACs which are currently in use by the local Bluetooth device to simultaneously listen for during an Inquiry Scan. Range: 0x01-0x40

IAC_LAP[i] *Size: 3 Octets * Num_Current_IAC*

Value	Parameter Description
0xXXXXXX	LAP(s) used to create IAC which is currently in use by the local Bluetooth device to simultaneously listen for during an Inquiry Scan. Range: 0x9E8B00-0x9E8B3F. The GIAC is the default IAC to be used. If additional IACs are supported, additional default IAC will be determined by the manufacturer.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Current_IAC_LAP command succeeded.
0x01-0xFF	Write_Current_IAC_LAP command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Current_IAC_LAP command has completed, a Command Complete event will be generated.

7.3.48 Read Page Scan Period Mode Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Page_Scan_Period_Mode	0x003B		Status, Page_Scan_Period_Mode

Description:

This command is used to read the mandatory Page_Scan_Period_Mode configuration parameter of the local Bluetooth device. See Section 6.10 on page 360.

Command Parameters:

None

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Page_Scan_Period_Mode command succeeded.
0x01-0xFF	Read_Page_Scan_Period_Mode command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Page_Scan_Period_Mode: *Size: 1 Octet*

Value	Parameter Description
0x00	P0
0x01	P1
0x02	P2
0x03-0xFF	Reserved.

Event(s) generated (unless masked away):

When the Read_Page_Scan_Period_Mode command has completed, a Command Complete event will be generated.

7.3.49 Write Page Scan Period Mode Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Page_Scan_Period_Mode	0x003C	Page_Scan_Period_Mode	Status

Description:

This command is used to write the mandatory Page_Scan_Period_Mode configuration parameter of the local Bluetooth device. See Section 6.10 on page 360.

Command Parameters:

Page_Scan_Period_Mode: *Size: 1 Octet*

Value	Parameter Description
0x00	P0. **Default.**
0x01	P1
0x02	P2
0x03-0xFF	Reserved.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Page_Scan_Period_Mode command succeeded.
0x01-0xFF	Write_Page_Scan_Period_Mode command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Page_Scan_Period_Mode command has completed, a Command Complete event will be generated.

7.3.50 Set AFH Host Channel Classification Command

Command	OCF	Command Parameters	Return Parameters
Set_AFH_Host_Channel_ Classification	0x003F	AFH_Host_Channel_ Classification	Status

Description:

The Set_AFH_Host_Channel_Classification command allows the Bluetooth host to specify a channel classification based on its “local information”. This classification persists until overwritten with a subsequent HCI Set_AFH_Host_Channel_Classification command or until the Controller is reset.

This command shall be supported by a device that declares support for any of the AFH_capable_master, AFH_classification_slave or AFH_classification_master features.

If this command is used, then updates should be sent within 10 seconds, of the host knowing that the channel classification has changed. The interval between two successive commands sent shall be at least 1 second.

Command Parameters:

AFH_Host_Channel_Classification: *Size: 10 Octets (79 Bits meaningful)*

Value	Parameter Description
0xXXXXXXXX XXXXXXXXXX XX	This parameter contains 79 1-bit field. The n^{th} such field (in the range 0 to 78) contains the value for channel *n*: Channel *n* is bad = 0 Channel *n* is unknown = 1 The most significant bit is reserved and shall be set to 0 At least N_{min} channels shall be marked as unknown. (See Baseband Specification, Section 2.3.1, on page 63)

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Set_AFH_Host_Channel_Classification command succeeded.
0x01-0xFF	Set_AFH_Host_Channel_Classification command failed. “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Set_AFH_Host_Channel_Classification command has completed, a Command Complete event will be generated.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 260

7.3.51 Read Inquiry Scan Type Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Inquiry_Scan_Type	0x0042		Status, Inquiry_Scan_Type

Description:

This command is used to read the Inquiry_Scan_Type configuration parameter of the local Bluetooth device. See Section 6.4 on page 358.
For details, see the Baseband Specification, “Inquiry scan substate” on page 144 [Part B].

Command Parameters:

None

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Inquiry_Scan_Type command succeeded
0x01-0xFF	Read_Inquiry_Scan_Type command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Inquiry_Scan_Type: *Size: 1 Octet*

Value	Parameter Description
0x00	Mandatory: Standard Scan (default)
0x01	Optional: Interlaced Scan
0x02-0xFF	Reserved

Event(s) generated (unless masked away):

When the Read_Inquiry_Scan_Type command has completed, a Command Complete event will be generated.

7.3.52 Write Inquiry Scan Type Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Inquiry_Scan_Type	0x0043	Scan_Type	Status

Description:

This command is used to write the Inquiry Scan Type configuration parameter of the local Bluetooth device. See Section 6.4 on page 358.
For details, see the Baseband Specification, “Inquiry scan substate” on page 144 [Part B].

Command Parameters:

Scan_Type: *Size: 1 Octet*

Value	Parameter Description
0x00	Mandatory: Standard Scan (default)
0x01	Optional: Interlaced Scan
0x02-0xFF	Reserved

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Inquiry_Scan_Type command succeeded
0x01-0xFF	Write_Inquiry_Scan_Type command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Inquiry_Scan_Type command has completed, a Command Complete event will be generated.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 262

7.3.53 Read Inquiry Mode Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Inquiry_Mode	0x0044		Status, Inquiry_Mode

Description:

This command is used to read the Inquiry_Mode configuration parameter of the local Bluetooth device. See Section 6.5 on page 358.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Inquiry_Mode command succeeded
0x01-0xFF	Read_Inquiry_Mode command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes

Inquiry_Mode: *Size: 1 Octet*

Value	Parameter Description
0x00	Standard Inquiry Result event format
0x01	Inquiry Result format with RSSI
0x02-0xFF	Reserved

Event(s) generated (unless masked away):

When the Read_Inquiry_Mode command has completed, a Command Complete event will be generated.

7.3.54 Write Inquiry Mode Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Inquiry_Mode	0x0045	Inquiry_Mode	Status

Description:

This command is used to write the Inquiry_Mode configuration parameter of the local Bluetooth device. See Section 6.5 on page 358.

Command Parameters:

Inquiry_Mode: *Size: 1 Octet*

Value	Parameter Description
0x00	Standard Inquiry Result event format (default)
0x01	Inquiry Result format with RSSI
0x02-0xFF	Reserved

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Inquiry_Mode command succeeded
0x01-0xFF	Write_Inquiry_Mode command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Inquiry_Mode command has completed, a Command Complete event will be generated.

7.3.55 Read Page Scan Type Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Page_Scan_Type	0x0046		Status, Page_Scan_Type

Description:

This command is used to read the Page Scan Type configuration parameter of the local Bluetooth device. See Section 6.11 on page 361.
For details, see the Baseband Specification, “Page scan substate” on page 134 [Part B].

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Page_Scan_Type command succeeded
0x01-0xFF	Read_Page_Scan_Type command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Page_Scan_Type: *Size: 1 Octet*

Value	Parameter Description
0x00	Mandatory: Standard Scan (default)
0x01	Optional: Interlaced Scan
0x02-0xFF	Reserved

Event(s) generated (unless masked away):

When the Read_Page_Scan_Type command has completed, a Command Complete event will be generated.

7.3.56 Write Page Scan Type Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Page_Scan_Type	0x0047	Page_Scan_Type	Status

Description:

This command is used to write the Page Scan Type configuration parameter of the local Bluetooth device. See Section 6.11 on page 361.
For details, see the Baseband Specification, “Page scan substate” on page 134 [Part B].

Command Parameters:

Page_Scan_Type: *Size: 1 Octet*

Value	Parameter Description
0x00	Mandatory: Standard Scan (default)
0x01	Optional: Interlaced Scan
0x02-0xFF	Reserved

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Page_Scan_Type command succeeded
0x01-0xFF	Write_Page_Scan_Type command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Page_Scan_Type command has completed, a Command Complete event will be generated.

7.3.57 Read AFH Channel Assessment Mode Command

Command	OCF	Command Parameters	Return Parameters
Read_AFH_Channel_Assessment_Mode	0x0048		Status, AFH_Channel_Assessment_Mode

Description:

The Read_AFH_Channel_Assessment_Mode command reads the value for the AFH_Channel_Assessment_Mode parameter. The AFH_Channel_Assessment_Mode parameter controls whether the controller's channel assessment scheme is enabled or disabled.

This command shall be supported by a device that declares support for any of the AFH_capable_master, AFH_classification_slave or AFH_classification_master features.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_AFH_Channel_Assessment_Mode command succeeded.
0x01-0xFF	Read_AFH_Channel_Assessment_Mode command failed. See "Error Codes" on page 291 [Part D] for list of Error Codes.

AFH_Channel_Assessment_Mode:: *Size: 1 Octet*

Value	Parameter Description
0x00	Controller channel assessment disabled.
0x01	Controller channel assessment enabled.
0x02-0xFF	Reserved for future use.

Event(s) generated (unless masked away):

When the Read_AFH_Channel_Assessment_Mode command has completed, a Command Complete event will be generated.

7.3.58 Write AFH Channel Assessment Mode Command

Command	OCF	Command Parameters	Return Parameters
Write_AFH_Channel _Assessment_Mode	0x0049	AFH_Channel _Assessment_Mode	Status

Description:

The Write_AFH_Channel_Assessment_Mode command writes the value for the AFH_Channel_Assessment_Mode parameter. The AFH_Channel_Assessment_Mode parameter controls whether the Controller's channel assessment scheme is enabled or disabled.

Disabling channel assessment forces all channels to be unknown in the local classification, but does not affect the AFH_reporting_mode or support for the Set_AFH_Host_Channel_Classification command. A slave in the AFH_reporting_enabled state shall continue to send LMP channel classification messages for any changes to the channel classification caused by either this command (altering the AFH_Channel_Assessment_Mode) or HCI Set_AFH_Host_Channel_Classification command (providing a new channel classification from the Host).

This command shall be supported by a device that declares support for any of the AFH_capable_master, AFH_classification_slave or AFH_classification_master features.

If the AFH_Channel_Assessment_Mode parameter is enabled and the Controller does not support a channel assessment scheme, other than via the Set_AFH_Host_Channel_Classification command, then a Status parameter of 'Channel Assessment Not Supported' should be returned. See "Error Codes" on page 291 [Part D] for list of Error Codes.

If the Controller supports a channel assessment scheme then the default AFH_Channel_Assessment_Mode is enabled, otherwise the default is disabled.

Command Parameters:

AFH_Channel_Assessment_Mode: *Size: 1 Octet*

Value	Parameter Description
0x00	Controller channel assessment disabled.
0x01	Controller channel assessment enabled.
0x02-0xFF	Reserved for future use.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_AFH_Channel_Assessment_Mode command succeeded.
0x01-0xFF	Write_AFH_Channel_Assessment_Mode command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_AFH_Channel_Assessment_Mode command has completed, a Command Complete event will be generated.

7.4 INFORMATIONAL PARAMETERS

The Informational Parameters are fixed by the manufacturer of the Bluetooth hardware. These parameters provide information about the Bluetooth device and the capabilities of the Controller, Link Manager, and Baseband. The host device cannot modify any of these parameters. For Informational Parameters Commands, the OGF is defined as 0x04.

7.4.1 Read Local Version Information Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Local_Version_ Information	0x0001		Status, HCI Version, HCI Revision, LMP Version, Manufacturer_Name, LMP Subversion

Description:

This command will read the values for the version information for the local Bluetooth device.

The HCI Version information defines the version information of the HCI layer. The LMP Version information defines the version of the LMP. The Manufacturer_Name information indicates the manufacturer of the local device.

The HCI Revision and LMP Subversion are implementation dependant.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Local_Version_Information command succeeded.
0x01-0xFF	Read_Local_Version_Information command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

HCI_Version: *Size: 1 Octet*

Value	Parameter Description
See https://www.bluetooth.org/foundry/assignnumb/document/assigned_numbers	

HCI_Revision: *Size: 2 Octets*

Value	Parameter Description
0xXXXX	Revision of the Current HCI in the Bluetooth device.

LMP_Version: *Size: 1 Octet*

Value	Parameter Description
0xXX	Version of the Current LMP in the Bluetooth device, See https://www.bluetooth.org/foundry/assignnumb/document/assigned_numbers

Manufacturer_Name: *Size: 2 Octets*

Value	Parameter Description
0xXXXX	Manufacturer Name of the Bluetooth device, See https://www.bluetooth.org/foundry/assignnumb/document/assigned_numbers

LMP_Subversion: *Size: 2 Octets*

Value	Parameter Description
0xXXXX	Subversion of the Current LMP in the Bluetooth device, see Table 5.2 on page 277 in the Link Manager Protocol for assigned values (SubVersNr).

Event(s) generated (unless masked away):

When the Read_Local_Version_Information command has completed, a Command Complete event will be generated.

7.4.2 Read Local Supported Commands Command

Command	OCF	Command Parameters	Return Parameters
HCI_ Read_Local_Supported_Commands	0x0002		Status, Supported Commands

Description:

This command reads the list of HCI commands supported for the local device.

This command will return the Supported Commands configuration parameter. It is implied that if a command is listed as supported, the feature underlying that command is also supported.

See Section 6.26 on page 369 for more information.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0	Read Local Supported Commands command succeeded
0x01-0xff	Read Local Supported Commands command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Supported Commands: *Size: 64 Octets*

Value	Parameter Description
	Bit mask for each HCI Command. If a bit is 1, the radio supports the corresponding command and the features required for the command. Unsupported or undefined commands shall be set to 0. See Section 6.26, “Supported Commands,” on page 369.

Event(s) generated (unless masked away):

When the Read Local Supported Commands command has completed a Command Complete event will be generated.

7.4.3 Read Local Supported Features Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Local_Supported_Features	0x0003		Status, LMP_Features

Description:

This command requests a list of the supported features for the local device. This command will return a list of the LMP features. For details see “Link Manager Protocol” on page 189 [Part C].

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Local_Supported_Features command succeeded.
0x01-0xFF	Read_Local_Supported_Features command failed. See “Error Codes” on page 291 [Part D]

LMP_Features: *Size: 8 Octets*

Value	Parameter Description
0xXXXXXXXX XXXXXXXX	Bit Mask List of LMP features. For details see “Link Manager Protocol” on page 189 [Part C]

Event(s) generated (unless masked away):

When the Read_Local_Supported_Features command has completed, a Command Complete event will be generated.

7.4.4 Read Local Extended Features Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Local_ Extended_Features	0x0004	Page number	Status, Page number, Maximum Page Number, Extended_LMP_Features

Description:

The HCI_Read_Local_Extended_Features command returns the requested page of the extended LMP features.

Command Parameters:

Page Number: *Size: 1 Octet*

Value	Parameter Description
0x00	Requests the normal LMP features as returned by HCI_Read_Local_Supported_Features
0x01-0xFF	Return the corresponding page of features

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	HCI_Read_Local_Extended_Features command succeeded
0x01-0xFF	HCI_Read_Local_Extended_Features command failed. See “Error Codes” on page 291 [Part D] for list of error codes

Page Number: *Size: 1 Octet*

Value	Parameter Description
0x00	The normal LMP features as returned by HCI_Read_Local_Supported_Features
0x01-0xFF	The page number of the features returned

Maximum Page Number: *Size: 1 Octet*

Value	Parameter Description
0x00-0xFF	The highest features page number which contains non-zero bits for the local device

Extended_LMP_Features: *Size: 8 Octets*

Value	Parameter Description
0xFFFFFFFFFFFFFFFF	Bit map of requested page of LMP features. See LMP specification for details

Event(s) generated (unless masked away):

When the Controller receives the HCI_Read_Local_Extended_Features command the Controller sends the Command Complete command to the Host containing the requested information.

7.4.5 Read Buffer Size Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Buffer_Size	0x0005		Status, HC_ACL_Data_Packet_Length, HC_Synchronous_Data_Packet_Length, HC_Total_Num_ACL_Data_Packets, HC_Total_Num_Synchronous_Data_Packets

Description:

The Read_Buffer_Size command is used to read the maximum size of the data portion of HCI ACL and synchronous Data Packets sent from the Host to the Controller. The Host will segment the data to be transmitted from the Host to the Controller according to these sizes, so that the HCI Data Packets will contain data with up to these sizes. The Read_Buffer_Size command also returns the total number of HCI ACL and synchronous Data Packets that can be stored in the data buffers of the Controller. The Read_Buffer_Size command must be issued by the Host before it sends any data to the Controller.

The HC_ACL_Data_Packet_Length return parameter will be used to determine the size of the L2CAP segments contained in ACL Data Packets, which are transferred from the Host to the Controller to be broken up into baseband packets by the Link Manager. The HC_Synchronous_Data_Packet_Length return parameter is used to determine the maximum size of HCI synchronous Data Packets. Both the Host and the Controller must support command and event packets, where the data portion (excluding header) contained in the packets is 255 octets in size. The HC_Total_Num_ACL_Data_Packets return parameter contains the total number of HCI ACL Data Packets that can be stored in the data buffers of the Controller. The Host will determine how the buffers are to be divided between different Connection Handles. The HC_Total_Num_Synchronous_Data_Packets return parameter gives the same information but for HCI synchronous Data Packets.
Note: the HC_ACL_Data_Packet_Length and HC_Synchronous_Data_Packet_Length return parameters do not include the length of the HCI Data Packet header.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Buffer_Size command succeeded.
0x01-0xFF	Read_Buffer_Size command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

HC_ACL_Data_Packet_Length: *Size: 2 Octets*

Value	Parameter Description
0xXXXX	Maximum length (in octets) of the data portion of each HCI ACL Data Packet that the Controller is able to accept.

HC_Synchronous_Data_Packet_Length: *Size: 1 Octet*

Value	Parameter Description
0xXX	Maximum length (in octets) of the data portion of each HCI Synchronous Data Packet that the Controller is able to accept.

HC_Total_Num_ACL_Data_Packets: *Size: 2 Octets*

Value	Parameter Description
0xXXXX	Total number of HCI ACL Data Packets that can be stored in the data buffers of the Controller.

HC_Total_Num_Synchronous_Data_Packets: *Size: 2 Octets*

Value	Parameter Description
0xXXXX	Total number of HCI Synchronous Data Packets that can be stored in the data buffers of the Controller.

Event(s) generated (unless masked away):

When the Read_Buffer_Size command has completed, a Command Complete event will be generated.

7.4.6 Read BD_ADDR Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_BD_ADDR	0x0009		Status, BD_ADDR

Description:

This command shall read the Bluetooth device address (BD_ADDR). See the “Baseband Specification” on page 45 [Part B] for details of how BD_ADDR is used.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_BD_ADDR command succeeded.
0x01-0xFF	Read_BD_ADDR command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device

Event(s) generated (unless masked away):

When the Read_BD_ADDR command has completed, a Command Complete event will be generated.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 278

7.5 STATUS PARAMETERS

The Controller modifies all status parameters. These parameters provide information about the current state of the Controller, Link Manager, and Baseband. The host device cannot modify any of these parameters other than to reset certain specific parameters. For the Status and baseband, the OGF is defined as 0x05.

7.5.1 Read Failed Contact Counter Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Failed_ Contact_Counter	0x0001	Connection_Handle	Status, Connection_Handle, Failed_Contact_Counter

Description:

This command will read the value for the Failed_Contact_Counter parameter for a particular connection to another device. The Connection_Handle must be a Connection_Handle for an ACL connection.
See Section 6.17 on page 364.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the Failed Contact Counter should be read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Failed_Contact_Counter command succeeded.
0x01-0xFF	Read_Failed_Contact_Counter command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the Failed Contact Counter has been read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Failed_Contact_Counter: *Size: 2 Octets*

Value	Parameter Description
0xXXXX	Number of consecutive failed contacts for a connection corresponding to the connection handle.

Event(s) generated (unless masked away):

When the Read_Failed_Contact_Counter command has completed, a Command Complete event will be generated.

7.5.2 Reset Failed Contact Counter Command

Command	OCF	Command Parameters	Return Parameters
HCI_Reset_Failed_ Contact_Counter	0x0002	Connection_Handle	Status, Connection_Handle

Description:

This command will reset the value for the Failed_Contact_Counter parameter for a particular connection to another device. The Connection_Handle must be a Connection_Handle for an ACL connection.
See Section 6.17 on page 364.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the Failed Contact Counter should be reset. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Reset_Failed_Contact_Counter command succeeded.
0x01-0xFF	Reset_Failed_Contact_Counter command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the Failed Contact Counter has been reset. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Event(s) generated (unless masked away):

When the Reset_Failed_Contact_Counter command has completed, a Command
Complete event will be generated.

7.5.3 Read Link Quality Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Link_Quality	0x0003	Connection_Handle	Status, Connection_Handle, Link_Quality

Description:

This command will return the value for the Link_Quality for the specified Connection Handle. The Connection_Handle must be a Connection_Handle for an ACL connection. This command will return a Link_Quality value from 0-255, which represents the quality of the link between two Bluetooth devices. The higher the value, the better the link quality is. Each Bluetooth module vendor will determine how to measure the link quality.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the connection for which link quality parameters are to be read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Link_Quality command succeeded.
0x01-0xFF	Read_Link_Quality command failed.See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the connection for which the link quality parameter has been read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Petitioner Kiosoft Exhibit 1011 Part 2
Page 282

Link_Quality: *Size: 1 Octet*

Value	Parameter Description
0xXX	The current quality of the Link connection between the local device and the remote device specified by the Connection Handle Range: 0x00 – 0xFF The higher the value, the better the link quality is.

Event(s) generated (unless masked away):

When the Read_Link_Quality command has completed, a Command Complete event will be generated.

7.5.4 Read RSSI Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_RSSI	0x0005	Connection_Handle	Status, Connection_Handle, RSSI

Description:

This command will read the value for the difference between the measured Received Signal Strength Indication (RSSI) and the limits of the Golden Receive Power Range (see Radio Specification Section 4.6 on page 39) for a connection handle to another Bluetooth device. The Connection_Handle must be a Connection_Handle for an ACL connection. Any positive RSSI value returned by the Controller indicates how many dB the RSSI is above the upper limit, any negative value indicates how many dB the RSSI is below the lower limit. The value zero indicates that the RSSI is inside the Golden Receive Power Range.
Note: how accurate the dB values will be depends on the Bluetooth hardware. The only requirements for the hardware are that the Bluetooth device is able to tell whether the RSSI is inside, above or below the Golden Device Power Range.

The RSSI measurement compares the received signal power with two threshold levels, which define the Golden Receive Power Range. The lower threshold level corresponds to a received power between -56 dBm and 6 dB above the actual sensitivity of the receiver. The upper threshold level is 20 dB above the lower threshold level to an accuracy of +/- 6 dB.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the RSSI is to be read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_RSSI command succeeded.
0x01-0xFF	Read_RSSI command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the RSSI has been read. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

RSSI: *Size: 1 Octet*

Value	Parameter Description
N = 0xXX	Size: 1 Octet (signed integer) Range: -128 ≤ N ≤ 127 Units: dB

Event(s) generated (unless masked away):

When the Read_RSSI command has completed, a Command Complete event will be generated.

7.5.5 Read AFH Channel Map Command

Command	OCF	Command Parameters	Return Parameters
Read_AFH _Channel_Map	0x0006	Connection_Handle	Status, Connection_Handle, AFH_Mode, AFH_Channel_Map

Description:

This command will return the values for the AFH_Mode and AFH_Channel_Map for the specified Connection Handle. The Connection_Handle must be a Connection_Handle for an ACL connection.

The returned values indicate the state of the hop sequence specified by the most recent LMP_Set_AFH message for the specified Connection_Handle, regardless of whether the master has received the baseband ACK or whether the AFH_Instant has passed.

This command shall be supported by a device that declares support for either the AFH_capable_slave or AFH_capable_master feature.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the Channel Map is to be read. Range: 0x0000-0x0EFF (0x0F00-0x0FFF Reserved for future use)

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_AFH_Channel_Map command succeeded.
0x01-0xFF	Read_AFH_Channel_Map command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the Channel Map is to be read. Range: 0x0000-0x0EFF (0x0F00-0x0FFF Reserved for future use)

AFH_Mode: *Size: 1 Octet*

Value	Parameter Description
0x00	AFH disabled.
0x01	AFH enabled.
0x02-0xFF	Reserved for future use.

AFH_Channel_Map: *Size: 10 Octets (79 Bits meaningful)*

Value	Parameter Description
0xXXXXXXXX XXXXXXXX XXX	If AFH_Mode is AFH enabled then this parameter contains 79 1-bit fields, otherwise the contents are reserved. The n^{th} such field (in the range 0 to 78) contains the value for channel *n*: Channel *n* is unused = 0 Channel *n* is used = 1 Range: 0x00000000000000000000-0x7FFFFFFFFFFFFFFFFFFF (0x80000000000000000000-0xFFFFFFFFFFFFFFFFFFFF Reserved for future use)

Event(s) generated (unless masked away):

When the Read_AFH_Channel_Map command has completed, a Command Complete event will be generated.

7.5.6 Read Clock Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Clock	0x0007	Which_Clock Connection_Handle	Status, Connection_Handle, Clock, Accuracy

Description:

This command will read the estimate of the value of the Bluetooth Clock.

If the Which_Clock value is 0, then the Connection_Handle shall be ignored, and the local Bluetooth Clock value shall be returned, and the accuracy parameter shall be set to 0.

If the Which_Clock value is 1, then the Connection_Handle must be a valid ACL connection handle. If the current role of this ACL connection is Master, then the Bluetooth Clock of this device shall be returned. If the current role is Slave, then an estimate of the Bluetooth Clock of the remote master and the accuracy of this value shall be returned.

The accuracy reflects the clock drift that might have occurred since the slave last received a valid transmission from the master.

Note: The Bluetooth Clock has a minimum accuracy of 250ppm, or about 22 seconds drift in one day.

Note: See Baseband Specification, Section 1.1, on page 54 for more information about the Bluetooth Clock.

Command Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify which connection to be used for reading the masters Bluetooth Clock. Range: 0x0000-0x0EFF (0x0F00 – 0x0FFF Reserved for future use)

Which_Clock: *Size 1 Octet)*

Value	Parameter Description
0xXX	0x00 = Local Clock (Connection Handle does not have to be a valid handle) 0x01 = Piconet Clock (Connection Handle shall be a valid ACL Handle) 0x02 to 0xFF = Reserved

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_BD_CLOCK command succeeded.
0x01 – 0xFF	Read_BD_CLOCK command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The Connection Handle for the Connection for which the master clock has been read. If the Which_Clock was 0, then the Connection_Handle shall be set to 0 and ignored upon receipt Range: 0x0000-0x0EFF (0x0F00 – 0x0FFF Reserved for future use)

Clock: *Size: 4 Octets (28 bits meaningful)*

Value	Parameter Description
0xXXXXXXXX	Bluetooth Clock of the device requested.

Accuracy: *Size: 2 Octets*

Value	Parameter Description
0xXX	+/- maximum Bluetooth Clock error. Value of 0xFFFF means Unknown. Accuracy = +/- N * 0.3125 msec (1 Bluetooth Clock) Range for N: 0x00-0xFFFE, 0xFFFF Time Range for N: 0 - 20479.375 sec

Event(s) generated (unless masked away):

When the Read_Clock command has completed, a Command Complete event will be generated.

7.6 TESTING COMMANDS

The Testing commands are used to provide the ability to test various functionalities of the Bluetooth hardware. These commands provide the ability to arrange various conditions for testing. For the Testing Commands, the OGF is defined as 0x06.

7.6.1 Read Loopback Mode Command

Command	OCF	Command Parameters	Return Parameters
HCI_Read_Loopback_Mode	0x0001		Status, Loopback_Mode

Description:

This command will read the value for the setting of the Controller's Loopback Mode. The setting of the Loopback Mode will determine the path of information. In Non-testing Mode operation, the Loopback Mode is set to Non-testing Mode and the path of the information is as specified by the Bluetooth specifications. In Local Loopback Mode, every Data Packet (ACL, SCO and eSCO) and Command Packet that is sent from the Host to the Controller is sent back with no modifications by the Controller, as shown in Figure 7.1 on page 533.
]For details of loopback modes see Section 7.6.2 on page 532.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Loopback_Mode command succeeded.
0x01-0xFF	Read_Loopback_Mode command failed. See "Error Codes" on page 291 [Part D] for list of Error Codes.

Loopback_Mode: *Size: 1 Octet*

Value	Parameter Description
0x00	No Loopback mode enabled. **Default.**
0x01	Enable Local Loopback.
0x02	Enable Remote Loopback.
0x03-0xFF	Reserved for Future Use.

Event(s) generated (unless masked away):

When the Read_Loopback_Mode command has completed, a Command Complete event will be generated.

7.6.2 Write Loopback Mode Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Loopback_Mode	0x0002	Loopback_Mode	Status

Description:

This command will write the value for the setting of the Controller's Loopback Mode. The setting of the Loopback Mode will determine the path of information. In Non-testing Mode operation, the Loopback Mode is set to Non-testing Mode and the path of the information as specified by the Bluetooth specifications. In Local Loopback Mode, every Data Packet (ACL, SCO and eSCO) and Command Packet that is sent from the Host to the Controller is sent back with no modifications by the Controller, as shown in Figure 7.1 on page 533.

When the Bluetooth Host Controller enters Local Loopback Mode, it shall respond with one to four connection handles, one for an ACL connection and zero to three for synchronous connections. The Host should use these connection handles when sending data in Local Loopback Mode. The number of connection handles returned for synchronous connections (between zero and three) is implementation specific. When in Local Loopback Mode, the Controller loops back commands and data to the Host. The Loopback Command event is used to loop back commands that the Host sends to the Controller.

There are some commands that are not looped back in Local Loopback Mode: Reset, Set_Controller_To_Host_Flow_Control, Host_Buffer_Size, Host_Number_Of_Completed_Packets, Read_Buffer_Size, Read_Loopback_Mode and Write_Loopback_Mode. These commands should be executed in the way they are normally executed. The commands Reset and Write_Loopback_Mode can be used to exit local loopback mode.

If Write_Loopback_Mode is used to exit Local Loopback Mode, Disconnection Complete events corresponding to the Connection Complete events that were sent when entering Local Loopback Mode should be sent to the Host. Furthermore, no connections are allowed in Local Loopback mode. If there is a connection, and there is an attempt to set the device to Local Loopback Mode, the attempt will be refused. When the device is in Local Loopback Mode, the Controller will refuse incoming connection attempts. This allows the Host Controller Transport Layer to be tested without any other variables.

If a device is set to Remote Loopback Mode, it will send back all data (ACL, SCO and eSCO) that comes over the air. It will only allow a maximum of one ACL connection and three synchronous connections, and these shall all be to the same remote device. If there are existing connections to a remote device and there is an attempt to set the local device to Remote Loopback Mode, the attempt shall be refused.

See Figure 7.2 on page 533, where the rightmost device is set to Remote Loopback Mode and the leftmost device is set to Non-testing Mode. This allows the Bluetooth Air link to be tested without any other variables.

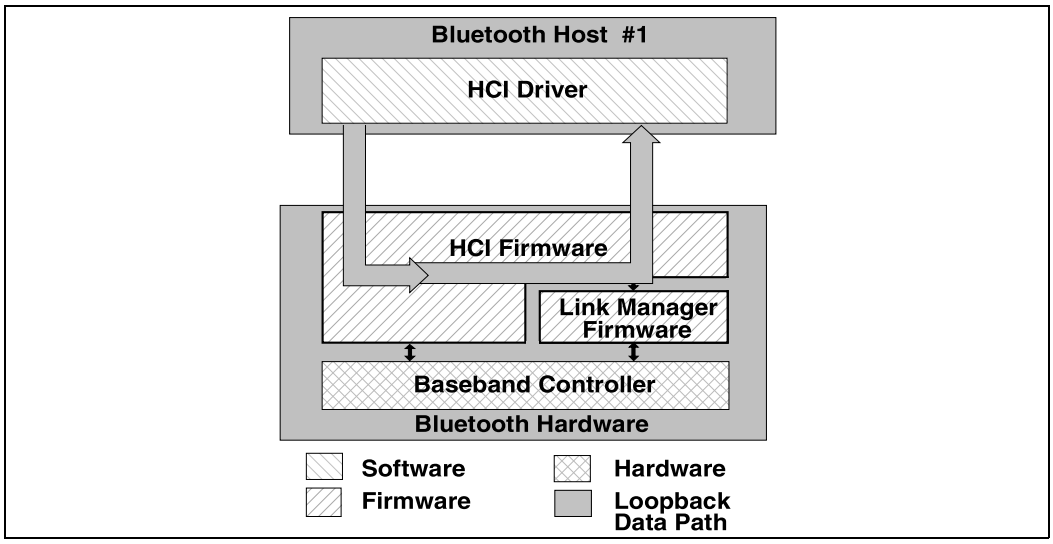


Figure 7.1: Local Loopback Mode

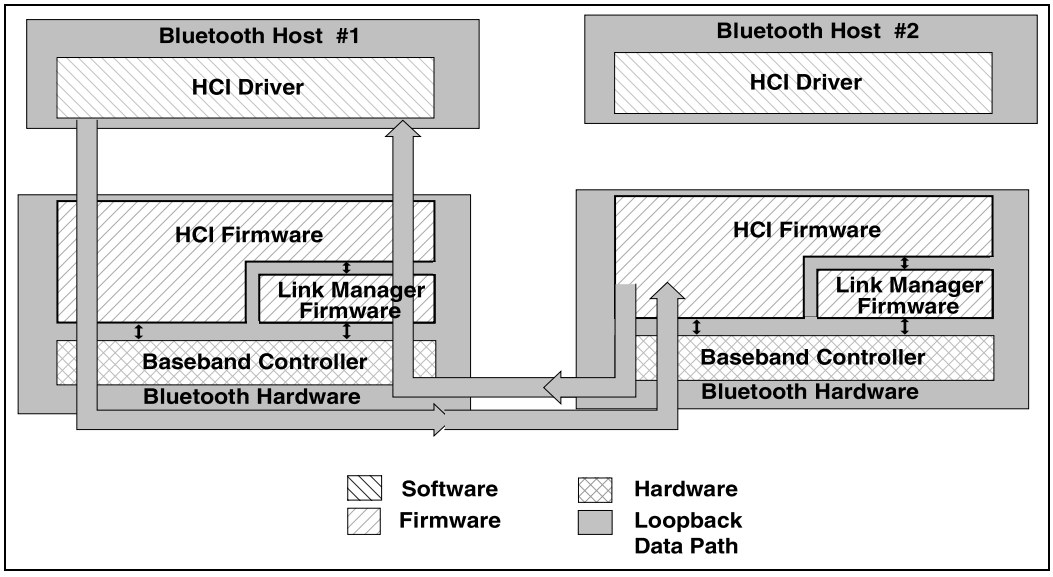


Figure 7.2: Remote Loopback Mode

Command Parameters:

Loopback_Mode: *Size: 1 Octet*

Value	Parameter Description
0x00	No Loopback mode enabled. **Default.**
0x01	Enable Local Loopback.
0x02	Enable Remote Loopback.
0x03-0xFF	Reserved for Future Use.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Loopback_Mode command succeeded.
0x01-0xFF	Write_Loopback_Mode command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Loopback_Mode command has completed, a Command Complete event will be generated.

7.6.3 Enable Device Under Test Mode Command

Command	OCF	Command Parameters	Return Parameters
HCI_Enable_Device_Under_ Test_Mode	0x0003		Status

Description:

The Enable_Device_Under_Test_Mode command will allow the local Bluetooth module to enter test mode via LMP test commands. For details see “Link Manager Protocol” on page 189 [Part C]. The Host issues this command when it wants the local device to be the DUT for the Testing scenarios as described in the “Test Methodology” on page 233[vol. 3]. When the Controller receives this command, it will complete the command with a Command Complete event. The Controller functions as normal until the remote tester issues the LMP test command to place the local device into Device Under Test mode. To disable and exit the Device Under Test Mode, the Host can issue the HCI_Reset command. This command prevents remote Bluetooth devices from causing the local Bluetooth device to enter test mode without first issuing this command.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Enter_Device_Under_Test_Mode command succeeded.
0x01-0xFF	Enter_Device_Under_Test_Mode command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Enter_Device_Under_Test_Mode command has completed, a Command Complete event will be generated.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 294

7.7 EVENTS

7.7.1 Inquiry Complete Event

Event	Event Code	Event Parameters
Inquiry Complete	0x01	Status

Description:

The Inquiry Complete event indicates that the Inquiry is finished. This event contains a status parameter, which is used to indicate if the Inquiry completed successfully or if the Inquiry was not completed.

Event Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Inquiry command completed successfully.
0x01-0xFF	Inquiry command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

7.7.2 Inquiry Result Event

Event	Event Code	Event Parameters
Inquiry Result	0x02	Num_Responses, BD_ADDR[i], Page_Scan_Repetition_Mode[i], Page_Scan_Period_Mode[i], Reserved[i], Class_of_Device[i] Clock_Offset[i]

Description:

The Inquiry Result event indicates that a Bluetooth device or multiple Bluetooth devices have responded so far during the current Inquiry process. This event will be sent from the Controller to the Host as soon as an Inquiry Response from a remote device is received if the remote device supports only mandatory paging scheme. The Controller may queue these Inquiry Responses and send multiple Bluetooth devices information in one Inquiry Result event. The event can be used to return one or more Inquiry responses in one event.

Event Parameters:

Num_Responses: *Size: 1 Octet*

Value	Parameter Description
0xXX	Number of responses from the Inquiry.

BD_ADDR[i]: *Size: 6 Octets * Num_Responses*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR for each device which responded.

Page_Scan_Repetition_Mode[i]: *Size: 1 Octet * Num_Responses*

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
0x03 – 0xFF	Reserved

Page_Scan_Period_Mode[i]: *Size: 1 Octet * Num_Responses*

Value	Parameter Description
0x00	P0
0x01	P1
0x02	P2
0x03 – 0xFF	Reserved

Reserved[i]: *Size: 1 Octet * Num_Responses*

Value	Parameter Description
0xXX	Reserved, must be set to 0x00.

Class_of_Device[i]: *Size: 3 Octets * Num_Responses*

Value	Parameter Description
0xXXXXXX	Class of Device for the device

Clock_Offset[i]: *Size: 2 Octets * Num_Responses*

Bit format	Parameter Description
Bit 14-0	Bit 16-2 of CLKslave-CLKmaster.
Bit 15	Reserved

7.7.3 Connection Complete Event

Event	Event Code	Event Parameters
Connection Complete	0x03	Status, Connection_Handle, BD_ADDR, Link_Type, Encryption_Mode

Description:

The Connection Complete event indicates to both of the Hosts forming the connection that a new connection has been established. This event also indicates to the Host, which issued the Create Connection, or Accept_Connection_Request or Reject_Connection_Request command and then received a Command Status event, if the issued command failed or was successful.

Event Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Connection successfully completed.
0x01-0xFF	Connection failed to Complete. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection between two Bluetooth devices. The Connection Handle is used as an identifier for transmitting and receiving voice or data. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the other connected Device forming the connection.

Link_Type: *Size: 1 Octet*

Value	Parameter Description
0x00	SCO connection.
0x01	ACL connection (Data Channels).
0x02-0xFF	Reserved for future use.

Encryption_Mode: *Size: 1 Octet*

Value	Parameter Description
See Section 6.16 on page 363.	

7.7.4 Connection Request Event

Event	Event Code	Event Parameters
Connection Request	0x04	BD_ADDR, Class_of_Device, Link_Type

Description:

The Connection Request event is used to indicate that a new incoming connection is trying to be established. The connection may either be accepted or rejected. If this event is masked away and there is an incoming connection attempt and the Controller is not set to auto-accept this connection attempt, the Controller will automatically refuse the connection attempt. When the Host receives this event and the link type parameter is ACL, it should respond with either an Accept_Connection_Request or Reject_Connection_Request command before the timer Conn_Accept_Timeout expires. If the link type is SCO or eSCO, the Host should reply with the Accept_Synchronous_Connection_Request or the Reject_Synchronous_Connection_Request Command. If the link type is SCO, the host may respond with Accept_Connection_Request. If the Event is responded to with Accept_Connection_Request, then the default parameter settings of the Accept_Synchronous_Connection_Request (see Section 7.1.27 on page 415) should be used by the local LM when negotiating the SCO or eSCO link parameters. In that case, the Connection_Complete Event and not the Synchronous_Connection_Complete Event, shall be returned on completion of the connection.

Event Parameters:

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the device that requests the connection.

Class_of_Device: *Size: 3 Octets*

Value	Parameter Description
0xXXXXXX	Class of Device for the device, which request the connection.
0x000000	Unknown Class of Device

Link_Type: *Size: 1 Octet*

Value	Parameter Description
0x00	SCO Connection requested
0x01	ACL Connection requested
0x02	eSCO Connection requested
0x03-0xFF	Reserved for Future Use.

7.7.5 Disconnection Complete Event

Event	Event Code	Event Parameters
Disconnection Complete	0x05	Status, Connection_Handle, Reason

Description:

The Disconnection Complete event occurs when a connection is terminated. The status parameter indicates if the disconnection was successful or not. The reason parameter indicates the reason for the disconnection if the disconnection was successful. If the disconnection was not successful, the value of the reason parameter can be ignored by the Host. For example, this can be the case if the Host has issued the Disconnect command and there was a parameter error, or the command was not presently allowed, or a connection handle that didn't correspond to a connection was given.
Note: When a physical link fails, one Disconnection Complete event will be returned for each logical channel on the physical link with the corresponding Connection handle as a parameter.

Event Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Disconnection has occurred.
0x01-0xFF	Disconnection failed to complete. See "Error Codes" on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle which was disconnected. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Reason: *Size: 1 Octet*

Value	Parameter Description
0xXX	Reason for disconnection. See "Error Codes" on page 291 [Part D] for error codes and description.

7.7.6 Authentication Complete Event

Event	Event Code	Event Parameters
Authentication Complete	0x06	Status, Connection_Handle

Description:

The Authentication Complete event occurs when authentication has been completed for the specified connection. The Connection_Handle must be a Connection_Handle for an ACL connection.

Event Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Authentication Request successfully completed.
0x01-0xFF	Authentication Request failed to complete. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle for which Authentication has been performed. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Petitioner Kiosoft Exhibit 1011 Part 2
Page 302

7.7.7 Remote Name Request Complete Event

Event	Event Code	Event Parameters
Remote Name Request Complete	0x07	Status, BD_ADDR, Remote_Name

Description:

The Remote Name Request Complete event is used to indicate that a remote name request has been completed. The Remote_Name event parameter is a UTF-8 encoded string with up to 248 octets in length. The Remote_Name event parameter will be null-terminated (0x00) if the UTF-8 encoded string is less than 248 octets. The BD_ADDR event parameter is used to identify which device the user-friendly name was obtained from.
Note: the Remote_Name Parameter is a string parameter. Endianess does therefore not apply to the Remote_Name Parameter. The first octet of the name is received first.

Event Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Remote_Name_Request command succeeded.
0x01-0xFF	Remote_Name_Request command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR for the device whose name was requested.

Remote_Name: *Size: 248 Octets*

Value	Parameter Description
Name[248]	A UTF-8 encoded user-friendly descriptive name for the remote device. If the name contained in the parameter is shorter than 248 octets, the end of the name is indicated by a NULL octet (0x00), and the following octets (to fill up 248 octets, which is the length of the parameter) do not have valid values.

7.7.8 Encryption Change Event

Event	Event Code	Event Parameters
Encryption Change	0x08	Status, Connection_Handle, Encryption_Enable

Description:

The Encryption Change event is used to indicate that the change in the encryption has been completed for the Connection Handle specified by the Connection_Handle event parameter. The Connection_Handle will be a Connection_Handle for an ACL connection. The Encryption_Enable event parameter specifies the new Encryption Enable for the Connection Handle specified by the Connection_Handle event parameter. This event will occur on both devices to notify the Hosts when Encryption has changed for the specified Connection Handle between two devices.

Event Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Encryption Change has occurred.
0x01-0xFF	Encryption Change failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle for which the link layer encryption has been enabled/disabled for all Connection Handles with the same Bluetooth device endpoint as the specified Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Encryption_Enable: *Size: 1 Octet*

Value	Parameter Description
0x00	Link Level Encryption is OFF.
0x01	Link Level Encryption is ON.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 304

7.7.9 Change Connection Link Key Complete Event

Event	Event Code	Event Parameters
Change Connection Link Key Complete	0x09	Status, Connection_Handle

Description:

The Change Connection Link Key Complete event is used to indicate that the change in the Link Key for the Connection Handle specified by the Connection_Handle event parameter has been completed.
The Connection_Handle will be a Connection_Handle for an ACL connection.
The Change Connection Link Key Complete event is sent only to the Host which issued the Change_Connection_Link_Key command.

Event Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Change_Connection_Link_Key command succeeded.
0x01-0xFF	Change_Connection_Link_Key command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle which the Link Key has been change for all Connection Handles with the same Bluetooth device end point as the specified Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

7.7.10 Master Link Key Complete Event

Event	Event Code	Event Parameters
Master Link Key Complete	0x0A	Status, Connection_Handle, Key_Flag

Description:

The Master Link Key Complete event is used to indicate that the Link Key managed by the master of the piconet has been changed. The Connection_Handle will be a Connection_Handle for an ACL connection. The link key used for the connection will be the temporary link key of the master device or the semi-permanent link key indicated by the Key_Flag. The Key_Flag event parameter is used to indicate which Link Key (temporary link key of the Master, or the semi-permanent link keys) is now being used in the piconet.
Note: for a master, the change from a semi-permanent Link Key to temporary Link Key will affect all Connection Handles related to the piconet. For a slave, this change affects only this particular connection handle. A temporary link key must be used when both broadcast and point-to-point traffic shall be encrypted.

Event Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Master_Link_Key command succeeded.
0x01-0xFF	Master_Link_Key command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle for which the Link Key has been changed for all devices in the same piconet. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Key_Flag: *Size: 1 Octet*

Value	Parameter Description
0x00	Using Semi-permanent Link Key.
0x01	Using Temporary Link Key.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 306

7.7.11 Read Remote Supported Features Complete Event

Event	Event Code	Event Parameters
Read Remote Supported Features Complete	0x0B	Status, Connection_Handle, LMP_Features

Description:

The Read Remote Supported Features Complete event is used to indicate the completion of the process of the Link Manager obtaining the supported features of the remote Bluetooth device specified by the Connection_Handle event parameter. The Connection_Handle will be a Connection_Handle for an ACL connection. The event parameters include a list of LMP features. For details see “Link Manager Protocol” on page 189[vol. 2].

Event Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Remote_Supported_Features command succeeded.
0x01-0xFF	Read_Remote_Supported_Features command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle which is used for the Read_Remote_Supported_Features command. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

LMP_Features: *Size: 8 Octets*

Value	Parameter Description
0xXXXXXXXX XXXXXXXX	Bit Mask List of LMP features. See “Link Manager Protocol” on page 189 [Part C].

7.7.12 Read Remote Version Information Complete Event

Event	Event Code	Event Parameters
Read Remote Version Information Complete	0x0C	Status, Connection_Handle, LMP_Version, Manufacturer_Name, LMP_Subversion

Description:

The Read Remote Version Information Complete event is used to indicate the completion of the process of the Link Manager obtaining the version information of the remote Bluetooth device specified by the Connection_Handle event parameter. The Connection_Handle will be a Connection_Handle for an ACL connection. The LMP_Version event parameter defines the specification version of the Bluetooth device. The Manufacturer_Name event parameter indicates the manufacturer of the remote Bluetooth device. The LMP_Subversion event parameter is controlled by the manufacturer and is implementation dependant. The LMP_Subversion event parameter defines the various revisions that each version of the Bluetooth hardware will go through as design processes change and errors are fixed. This allows the software to determine what Bluetooth hardware is being used and, if necessary, to work around various bugs in the hardware.

Event Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Remote_Version_Information command succeeded.
0x01-0xFF	Read_Remote_Version_Information command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle which is used for the Read_Remote_Version_Information command. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Petitioner Kiosoft Exhibit 1011 Part 2
Page 308

LMP_Version: *Size: 1 Octet*

Value	Parameter Description
0xXX	Version of the Current LMP in the remote Bluetooth device. For LMP_Version information see https://www.bluetooth.org/foundry/assign-numb/document/assigned_numbers

Manufacturer_Name: *Size: 2 Octets*

Value	Parameter Description
0xXXXX	Manufacturer Name of the remote Bluetooth device, see Table 5.2 on page 277 in the Link Manager Protocol for assigned values (CompId).

LMP_Subversion: *Size: 2 Octets*

Value	Parameter Description
0xXXXX	Subversion of the Current LMP in the remote Bluetooth device, see Table 5.2 on page 277 in the Link Manager Protocol for assigned values (SubVersNr).

7.7.13 QoS Setup Complete Event

Event	Event Code	Event Parameters
QoS Setup Complete	0x0D	Status, Connection_Handle, Flags, Service_Type, Token_Rate, Peak_Bandwidth, Latency, Delay_Variation

Description:

The QoS Setup Complete event is used to indicate the completion of the process of the Link Manager setting up QoS with the remote Bluetooth device specified by the Connection_Handle event parameter. The Connection_Handle will be a Connection_Handle for an ACL connection. For more detail see “Logical Link Control and Adaptation Protocol Specification” on page 15[vol. 3].

Event Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	QoS_Setup command succeeded.
0x01-0xFF	QoS_Setup command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle which is used for the QoS_Setup command. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Flags: *Size: 1 Octet*

Value	Parameter Description
0x00 – 0xFF	Reserved for Future Use.

Service_Type: *Size: 1 Octet*

Value	Parameter Description
0x00	No Traffic Available.
0x01	Best Effort Available.
0x02	Guaranteed Available.
0x03-0xFF	Reserved for Future Use.

Token_Rate: *Size: 4 Octets*

Value	Parameter Description
0xXXXXXXXX	Available Token Rate, in octets per second.

Peak_Bandwidth: *Size: 4 Octets*

Value	Parameter Description
0xXXXXXXXX	Available Peak Bandwidth, in octets per second.

Latency: *Size: 4 Octets*

Value	Parameter Description
0xXXXXXXXX	Available Latency, in microseconds.

Delay_Variation: *Size: 4 Octets*

Value	Parameter Description
0xXXXXXXXX	Available Delay Variation, in microseconds.

7.7.14 Command Complete Event

Event	Event Code	Event Parameters
Command Complete	0x0E	Num_HCI_Command_Packets, Command_Opcode, Return_Parameters

Description:

The Command Complete event is used by the Controller for most commands to transmit return status of a command and the other event parameters that are specified for the issued HCI command.

The Num_HCI_Command_Packets event parameter allows the Controller to indicate the number of HCI command packets the Host can send to the Controller. If the Controller requires the Host to stop sending commands, the Num_HCI_Command_Packets event parameter will be set to zero. To indicate to the Host that the Controller is ready to receive HCI command packets, the Controller generates a Command Complete event with the Command_Opcode 0x0000, and the Num_HCI_Command_Packets event parameter is set to 1 or more. Command_Opcode, 0x0000 is a NOP (No OPeration), and can be used to change the number of outstanding HCI command packets that the Host can send before waiting. See each command for the parameters that are returned by this event.

Event Parameters:

Num_HCI_Command_Packets: *Size: 1 Octet*

Value	Parameter Description
N = 0xXX	The Number of HCI command packets which are allowed to be sent to the Controller from the Host. Range for N: 0 – 255

Command_Opcode: *Size: 2 Octets*

Value	Parameter Description
0xXXXX	Opcode of the command which caused this event.

Return_Parameter(s): *Size: Depends on Command*

Value	Parameter Description
0xXX	This is the return parameter(s) for the command specified in the Command_Opcode event parameter. See each command's definition for the list of return parameters associated with that command.

7.7.15 Command Status Event

Event	Event Code	Event Parameters
Command Status	0x0F	Status, Num_HCI_Command_Packets, Command_Opcode

Description:

The Command Status event is used to indicate that the command described by the Command_Opcode parameter has been received, and that the Controller is currently performing the task for this command. This event is needed to provide mechanisms for asynchronous operation, which makes it possible to prevent the Host from waiting for a command to finish. If the command can not begin to execute (a parameter error may have occurred, or the command may currently not be allowed), the Status event parameter will contain the corresponding error code, and no complete event will follow since the command was not started. The Num_HCI_Command_Packets event parameter allows the Controller to indicate the number of HCI command packets the Host can send to the Controller. If the Controller requires the Host to stop sending commands, the Num_HCI_Command_Packets event parameter will be set to zero. To indicate to the Host that the Controller is ready to receive HCI command packets, the Controller generates a Command Status event with Status 0x00 and Command_Opcode 0x0000, and the Num_HCI_Command_Packets event parameter is set to 1 or more. Command_Opcode, 0x0000 is a NOP (No OPeration) and can be used to change the number of outstanding HCI command packets that the Host can send before waiting.

Event Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Command currently in pending.
0x01-0xFF	Command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Num_HCI_Command_Packets: *Size: 1 Octet*

Value	Parameter Description
N = 0xXX	The Number of HCI command packets which are allowed to be sent to the Controller from the Host. Range for N: 0 – 255

Command_Opcode: *Size: 2 Octets*

Value	Parameter Description
0xXXXX	Opcode of the command which caused this event and is pending completion.

7.7.16 Hardware Error Event

Event	Event Code	Event Parameters
Hardware Error	0x10	Hardware_Code

Description:

The Hardware Error event is used to indicate some type of hardware failure for the Bluetooth device. This event is used to notify the Host that a hardware failure has occurred in the Bluetooth device.

Event Parameters:

Hardware_Code: *Size: 1 Octet*

Value	Parameter Description
0x00-0xFF	These Hardware_Codes will be implementation-specific, and can be assigned to indicate various hardware problems.

7.7.17 Flush Occurred Event

Event	Event Code	Event Parameters
Flush Occurred	0x11	Connection_Handle

Description:

The Flush Occurred event is used to indicate that, for the specified Connection Handle, the current user data to be transmitted has been removed. The Connection_Handle will be a Connection_Handle for an ACL connection. This could result from the flush command, or be due to the automatic flush. Multiple blocks of an L2CAP packet could have been pending in the Controller. If one baseband packet part of an L2CAP packet is flushed, then the rest of the HCI data packets for the L2CAP packet must also be flushed.

Event Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle which was flushed. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

7.7.18 Role Change Event

Event	Event Code	Event Parameters
Role Change	0x12	Status, BD_ADDR, New_Role

Description:

The Role Change event is used to indicate that the current Bluetooth role related to the particular connection has changed. This event only occurs when both the remote and local Bluetooth devices have completed their role change for the Bluetooth device associated with the BD_ADDR event parameter. This event allows both affected Hosts to be notified when the Role has been changed.

Event Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Role change has occurred.
0x01-0xFF	Role change failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device for which a role change has completed.

New_Role: *Size: 1 Octet*

Value	Parameter Description
0x00	Currently the Master for specified BD_ADDR.
0x01	Currently the Slave for specified BD_ADDR.

7.7.19 Number Of Completed Packets Event

Event	Event Code	Event Parameters
Number Of Completed Packets	0x13	Number_of_Handles, Connection_Handle[i], HC_Num_Of_Completed_Packets[i]

Description:

The Number Of Completed Packets event is used by the Controller to indicate to the Host how many HCI Data Packets have been completed (transmitted or flushed) for each Connection Handle since the previous Number Of Completed Packets event was sent to the Host. This means that the corresponding buffer space has been freed in the Controller. Based on this information, and the HC_Total_Num_ACL_Data_Packets and HC_Total_ Num_Synchronous_Data_Packets return parameter of the Read_Buffer_Size command, the Host can determine for which Connection Handles the following HCI Data Packets should be sent to the Controller.The Number Of Completed Packets event must not be sent before the corresponding Connection Complete event. While the Controller has HCI data packets in its buffer, it must keep sending the Number Of Completed Packets event to the Host at least periodically, until it finally reports that all the pending ACL Data Packets have been transmitted or flushed. The rate with which this event is sent is manufacturer specific.
Note that Number Of Completed Packets events will not report on synchronous connection handles if synchronous Flow Control is disabled. (See Read/ Write_Synchronous_Flow_Control_Enable on page 486 and page 487.)

Event Parameters:

Number_of_Handles: *Size: 1 Octet*

Value	Parameter Description
0xXX	The number of Connection Handles and Num_HCI_Data_Packets parameters pairs contained in this event. Range: 0-255

*Connection_Handle[i]: Size: Number_of_Handles * 2 Octets(12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

*HC_Num_Of_Completed_Packets [i]: Size: Number_of_Handles * 2 Octets*

Value	Parameter Description
N = 0xXXXX	The number of HCI Data Packets that have been completed (transmitted or flushed) for the associated Connection Handle since the previous time the event was returned. Range for N: 0x0000-0xFFFF

7.7.20 Mode Change Event

Event	Event Code	Event Parameters
Mode Change	0x14	Status, Connection_Handle, Current_Mode, Interval

Description:

The Mode Change event is used to indicate when the device associated with the Connection Handle changes between Active mode, Hold mode, Sniff mode, and Park state. The Connection_Handle will be a Connection_Handle for an ACL connection. The Connection_Handle event parameter is used to indicate which connection the Mode Change event is for. The Current_Mode event parameter is used to indicate which state the connection is currently in. The Interval parameter is used to specify a time amount specific to each state. Each Controller that is associated with the Connection Handle which has changed Modes will send the Mode Change event to its Host.

Event Parameters:

Status: Size: 1 Octet

Value	Parameter Description
0x00	A Mode Change has occurred.
0x01-0xFF	Hold_Mode, Sniff_Mode, Exit_Sniff_Mode, Park_State, or Exit_Park_State command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: Size: 2 Octets(12 Bits meaningful)

Value	Parameter Description
0xXXXX	Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Current_Mode: *Size: 1 Octet*

Value	Parameter Description
0x00	Active Mode.
0x01	Hold Mode.
0x02	Sniff Mode.
0x03	Park State.
0x04-0xFF	Reserved for future use.

Interval: *Size: 2 Octets*

Value	Parameter Description
N = 0xXXXX	Hold: Number of Baseband slots to wait in Hold Mode. Hold Interval = N * 0.625 msec (1 Baseband slot) Range for N: 0x0002-0xFFFE Time Range: 1.25 msec-40.9 sec Sniff: Number of Baseband slots between sniff intervals. Time between sniff intervals = 0.625 msec (1 Baseband slot) Range for N: 0x0002-0xFFFE Time Range: 1.25 msec-40.9 sec Park: Number of Baseband slots between consecutive beacons. Interval Length = N * 0.625 msec (1 Baseband slot) Range for N: 0x0002-0xFFFE Time Range: 1.25 msec-40.9 Seconds

7.7.21 Return Link Keys Event

Event	Event Code	Event Parameters
Return Link Keys	0x15	Num_Keys, BD_ADDR [i], Link_Key[i]

Description:

The Return Link Keys event is used by the Controller to send the Host one or more stored Link Keys. Zero or more instances of this event will occur after the Read_Stored_Link_Key command. When there are no link keys stored, no Return Link Keys events will be returned. When there are link keys stored, the number of link keys returned in each Return Link Keys event is implementation specific.

Event Parameters:

Num_Keys: *Size: 1 Octet*

Value	Parameter Description
0xXX	Number of Link Keys contained in this event. Range: 0x01 – 0x0B

BD_ADDR [i]: *Size: 6 Octets * Num_Keys*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR for the associated Link Key.

Link_Key[i]: *Size:16 Octets * Num_Keys*

Value	Parameter Description
0xXXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	Link Key for the associated BD_ADDR.

7.7.22 PIN Code Request Event

Event	Event Code	Event Parameters
PIN Code Request	0x16	BD_ADDR

Description:

The PIN Code Request event is used to indicate that a PIN code is required to create a new link key. The Host must respond using either the PIN Code Request Reply or the PIN Code Request Negative Reply command, depending on whether the Host can provide the Controller with a PIN code or not.
Note: If the PIN Code Request event is masked away, then the Controller will assume that the Host has no PIN Code.

When the Controller generates a PIN Code Request event in order for the local Link Manager to respond to the request from the remote Link Manager (as a result of a Create_Connection or Authentication_Requested command from the remote Host), the local Host must respond with either a PIN_Code_Request_Reply or PIN_Code_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See “Link Manager Protocol” on page 189 [Part C].)

Event Parameters:

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device which a new link key is being created for.

7.7.23 Link Key Request Event

Event	Event Code	Event Parameters
Link Key Request	0x17	BD_ADDR

Description:

The Link Key Request event is used to indicate that a Link Key is required for the connection with the device specified in BD_ADDR. If the Host has the requested stored Link Key, then the Host will pass the requested Key to the Controller using the Link_Key_Request_Reply Command. If the Host does not have the requested stored Link Key, then the Host will use the Link_Key_Request_Negative_Reply Command to indicate to the Controller that the Host does not have the requested key.
Note: If the Link Key Request event is masked away, then the Controller will assume that the Host has no additional link keys.

When the Controller generates a Link Key Request event in order for the local Link Manager to respond to the request from the remote Link Manager (as a result of a Create_Connection or Authentication_Requested command from the remote Host), the local Host must respond with either a Link_Key_Request_Reply or Link_Key_Request_Negative_Reply command before the remote Link Manager detects LMP response timeout. (See “Link Manager Protocol” on page 189 [Part C].)

Event Parameters:

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device which a stored link key is being requested.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 320

7.7.24 Link Key Notification Event

Event	Event Code	Event Parameters
Link Key Notification	0x18	BD_ADDR, Link_Key,Key_Type

Description:

The Link Key Notification event is used to indicate to the Host that a new Link Key has been created for the connection with the device specified in BD_ADDR. The Host can save this new Link Key in its own storage for future use. Also, the Host can decided to store the Link Key in the Controller's Link Key Storage by using the Write_Stored_Link_Key command. The Key_Type event parameter informs the Host about which key type (combination key, local unit key or remote unit key) that has been used during pairing. If pairing with unit key is not supported, the Host can for instance discard the key or disconnect the link.

Event Parameters:

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the Device for which the new link key has been generated.

Link_Key: *Size: 16 Octets*

Value	Parameter Description
0xXXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	Link Key for the associated BD_ADDR.

Key_Type: *Size: 1 Octets*

Value	Parameter Description
0x00	Combination Key
0x01	Local Unit Key
0x02	Remote Unit Key
0x03-0xFF	Reserved

7.7.25 Loopback Command Event

Event	Event Code	Event Parameters
Loopback Command	0x19	HCI_Command_Packet

Description:

When in Local Loopback mode, the Controller loops back commands and data to the Host. The Loopback Command event is used to loop back all commands that the Host sends to the Controller with some exceptions. See Section 7.6.1, “Read Loopback Mode Command,” on page 531 for a description of which commands that are not looped back. The HCI_Command_Packet event parameter contains the entire HCI Command Packet including the header. Note: the event packet is limited to a maximum of 255 octets in the payload; since an HCI Command Packet has 3 octets of header data, only the first 252 octets of the command parameters will be returned.

Event Parameters:

HCI_Command_Packet: *Size: Depends on Command*

Value	Parameter Description
0xXXXXXX	HCI Command Packet, including header.

7.7.26 Data Buffer Overflow Event

Event	Event Code	Event Parameters
Data Buffer Overflow	0x1A	Link_Type

Description:

This event is used to indicate that the Controller’s data buffers have been overflowed. This can occur if the Host has sent more packets than allowed. The Link_Type parameter is used to indicate that the overflow was caused by ACL or synchronous data.

Event Parameters:

Link_Type: *Size: 1 Octet*

Value	Parameter Description
0x00	Synchronous Buffer Overflow (Voice Channels).
0x01	ACL Buffer Overflow (Data Channels).
0x02-0xFF	Reserved for Future Use.

7.7.27 Max Slots Change Event

Event	Event Code	Event Parameters
Max Slots Change	0x1B	Connection_Handle, LMP_Max_Slots

Description:

This event is used to notify the Host about the LMP_Max_Slots parameter when the value of this parameter changes. It will be sent each time the maximum allowed length, in number of slots, for baseband packets transmitted by the local device, changes. The Connection_Handle will be a Connection_Handle for an ACL connection.

Event Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

LMP_Max_Slots: *Size: 1 octet*

Value	Parameter Description
0x01, 0x03, 0x05	Maximum number of slots allowed to use for baseband packets, see Section 4.1.10 on page 223 and Section 5.2 on page 277 in “Link Manager Protocol” on page 189 [Part C].

7.7.28 Read Clock Offset Complete Event

Event	Event Code	Event Parameters
Read Clock Offset Complete	0x1C	Status, Connection_Handle, Clock_Offset

Description:

The Read Clock Offset Complete event is used to indicate the completion of the process of the Link Manager obtaining the Clock Offset information of the Bluetooth device specified by the Connection_Handle event parameter. The Connection_Handle will be a Connection_Handle for an ACL connection.

Event Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Clock_Offset command succeeded.
0x01-0xFF	Read_Clock_Offset command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 bits meaningful)*

Value	Parameter Description
0xXXXX	Specifies which Connection Handle’s Clock Offset parameter is returned. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Clock_Offset: *Size: 2 Octets*

Bit format	Parameter Description
Bit 14-0	Bit 16-2 of CLKslave-CLKmaster.
Bit 15	Reserved.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 324

7.7.29 Connection Packet Type Changed Event

Event	Event Code	Event Parameters
Connection Packet Type Changed	0x1D	Status, Connection_Handle, Packet_Type

Description:

The Connection Packet Type Changed event is used to indicate that the process has completed of the Link Manager changing which packet types can be used for the connection. This allows current connections to be dynamically modified to support different types of user data. The Packet_Type event parameter specifies which packet types the Link Manager can use for the connection identified by the Connection_Handle event parameter for sending L2CAP data or voice. The Packet_Type event parameter does not decide which packet types the LM is allowed to use for sending LMP PDUs.

Event Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Connection Packet Type changed successfully.
0x01-0xFF	Connection Packet Type Changed failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Packet_Type: *Size: 2 Octets*

For ACL_Link_Type

Value	Parameter Description
0x0001	Reserved for future use.
0x0002	Reserved for future use.
0x0004	Reserved for future use.
0x0008	DM1
0x0010	DH1
0x0020	Reserved for future use.
0x0040	Reserved for future use.
0x0080	Reserved for future use.

Value	Parameter Description
0x0100	Reserved for future use.
0x0200	Reserved for future use.
0x0400	DM3
0x0800	DH3
0x1000	Reserved for future use.
0x2000	Reserved for future use.
0x4000	DM5
0x8000	DH5

For SCO_Link_Type

Value	Parameter Description
0x0001	Reserved for future use.
0x0002	Reserved for future use.
0x0004	Reserved for future use.
0x0008	Reserved for future use.
0x0010	Reserved for future use.
0x0020	HV1
0x0040	HV2
0x0080	HV3
0x0100	Reserved for future use.
0x0200	Reserved for future use.
0x0400	Reserved for future use.
0x0800	Reserved for future use.
0x1000	Reserved for future use.
0x2000	Reserved for future use.
0x4000	Reserved for future use.
0x8000	Reserved for future use.

7.7.30 QoS Violation Event

Event	Event Code	Event Parameters
QoS Violation	0x1E	Connection_Handle

Description:

The QoS Violation event is used to indicate the Link Manager is unable to provide the current QoS requirement for the Connection Handle. This event indicates that the Link Manager is unable to provide one or more of the agreed QoS parameters. The Host chooses what action should be done. The Host can reissue QoS_Setup command to renegotiate the QoS setting for Connection Handle. The Connection_Handle will be a Connection_Handle for an ACL connection.

Event Parameters:

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle that the LM is unable to provide the current QoS requested for. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

7.7.31 Page Scan Repetition Mode Change Event

Event	Event Code	Event Parameters
Page Scan Repetition Mode Change	0x20	BD_ADDR, Page_Scan_Repetition_Mode

Description:

The Page Scan Repetition Mode Change event indicates that the remote Bluetooth device with the specified BD_ADDR has successfully changed the Page_Scan_Repetition_Mode (SR).

Event Parameters:

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the remote device.

Page_Scan_Repetition_Mode: *Size: 1 Octet*

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
0x03 – 0xFF	Reserved.

7.7.32 Flow Specification Complete Event

Event	Event Code	Event Parameters
HCI Flow Specification Complete	0x21	Status, Connection_Handle, Flags, Flow_direction, Service_Type, Token_Rate, Token_Bucket_Size, Peak_Bandwidth, Access Latency

Description:

The Flow Specification Complete event is used to inform the Host about the Quality of Service for the ACL connection the Controller is able to support. The Connection_Handle will be a Connection_Handle for an ACL connection. The flow parameters refer to the outgoing or incoming traffic of the ACL link, as indicated by the Flow_direction field. The flow parameters are defined in the L2CAP specification “Quality of Service (QoS) Option” on page 60[vol. 3]. When the Status parameter indicates a successful completion, the flow parameters specify the agreed values by the Controller. When the Status parameter indicates a failed completion with the Error Code *QoS Unacceptable Parameters* (0x2C), the flow parameters specify the acceptable values of the Controller. This enables the Host to continue the 'QoS negotiation' with a new HCI Flow_Specification command with flow parameter values that are acceptable for the Controller. When the Status parameter indicates a failed completion with the Error Code *QoS Rejected* (0x2D), this indicates a request of the Controller to discontinue the 'QoS negotiation'. When the Status parameter indicates a failed completion, the flow parameter values of the most recently successful completion must be assumed (or the default values when there was no success completion).

Event Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Flow Specification command succeeded
0x01 – 0xFF	Flow Specification command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle used to identify for which ACL connection the Flow is specified. Range: 0x0000 - 0x0EFF (0x0F00 – 0x0FFF Reserved for future use)

Flags: *Size: 1 Octet)*

Value	Parameter Description
0x00 – 0xFF	Reserved for Future Use.

Flow_direction: *Size: 1 Octet*

Value	Parameter Description
0x00	Outgoing Flow i.e. traffic send over the ACL connection
0x01	Incoming Flow i.e. traffic received over the ACL connection
0x02 – 0xFF	Reserved for Future Use.

Service_Type: *Size: 1 Octet*

Value	Parameter Description
0x00	No Traffic
0x01	Best Effort
0x02	Guaranteed
0x03 – 0xFF	Reserved for Future Use

Token Rate: *Size: 4 Octets*

Value	Parameter Description
0xXXXXXXXX	Token Rate in octets per second

Token Bucket Size: *Size: 4 Octets*

Value	Parameter Description
0xXXXXXXXX	Token Bucket Size in octets

Peak_Bandwidth: *Size: 4 Octets*

Value	Parameter Description
0xXXXXXXXX	Peak Bandwidth in octets per second

Access Latency: *Size: 4 Octets*

Value	Parameter Description
0xXXXXXXXX	Access Latency in microseconds

7.7.33 Inquiry Result with RSSI Event

Event	Event Code	Event Parameters
Inquiry Result with RSSI	0x22	Num_responses, BD_ADDR[i], Page_Scan_Repetition_Mode[i], Page_Scan_Period_Mode[i], Class_of_Device[i], Clock_Offset[i], RSSI[i]

Description:

The Inquiry Result with RSSI event indicates that a Bluetooth device or multiple Bluetooth devices have responded so far during the current Inquiry process. This event will be sent from the Controller to the Host as soon as an Inquiry Response from a remote device is received if the remote device supports only mandatory paging scheme. This Controller may queue these Inquiry Responses and send multiple Bluetooth devices information in one Inquiry Result event. The event can be used to return one or more Inquiry responses in one event. The RSSI parameter is measured during the FHS packet returned by each responding slave.

This event shall only be generated if the Inquiry Mode parameter of the last Write_Inquiry_Mode command was set to 0x01 (Inquiry Result format with RSSI).

Note: the only difference between the Inquiry Result with RSSI event and the Inquiry Result event is the additional RSSI parameter.

Event Parameters:

Num_Responses: *Size: 1 Octet*

Value	Parameter Description
0xXX	Number of responses from the Inquiry.

BD_ADDR[i]: *Size: 6 Octets * Num_Responses*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR for each device which responded.

Page_Scan_Repetition_Mode[i]: *Size: 1 Octet* Num_Responses*

Value	Parameter Description
0x00	R0
0x01	R1
0x02	R2
0x03 – 0xFF	Reserved

Page_Scan_Period_Mode[i]: *Size: 1 Octet* Num_Responses*

Value	Parameter Description
0x00	P0
0x01	P1
0x02	P2
0x03 – 0xFF	Reserved

Class_of_Device[i]: *Size: 3 Octets * Num_Responses*

Value	Parameter Description
0xXXXXXX	Class of Device for the device

Clock_Offset[i]: *Size: 2 Octets * Num_Responses*

Bit format	Parameter Description
Bit 14-0	Bit 16-2 of CLKslave-CLKmaster.
Bit 15	Reserved

RSSI[i]: *Size: 1 Octet * Num_Responses*

Value	Parameter Description
0xXX	Range: -127 to +20 Units: dBm

7.7.34 Read Remote Extended Features Complete Event

Event	Event Code	Event Parameters
Read Remote Extended Features Complete	0x23	Status, Connection_Handle, Page_Number, Maximum page number, Extended_LMP_Features

Description:

The Read Remote Extended Features Complete event is used to indicate the completion of the process of the Link Manager obtaining the remote extended LMP features of the remote device specified by the connection handle event parameter. The connection handle will be a connection handle for an ACL connection. The event parameters include a page of the remote devices extended LMP features. For details see “Link Manager Protocol” on page 189 [Part C].

Event Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Request for remote extended features succeeded
0x01-0xFF	Request for remote extended features failed – standard HCI error value

Connection_Handle: *Size: 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	The connection handle identifying the device to which the remote features apply. Range: 0x0000-0x0EFF (0x0F00-0x0FFF Reserved for future use)

Page Number: *Size: 1Octet*

Value	Parameter Description
0x00	The normal LMP features as returned by HCI_Read_Remote_Supported_Features
0x01-0xFF	The page number of the features returned

Maximum Page Number: *Size: 1Octet*

Value	Parameter Description
0x00-0xFF	The highest features page number which contains non-zero bits for the local device

Extended_LMP_Features: *Size: 8 Octets*

Value	Parameter Description
0xFFFFFFFFFFFFFFFF	Bit map of requested page of LMP features. See LMP specification for details

7.7.35 Synchronous Connection Complete Event

Event	Event Code	Event Parameters
Synchronous Connection Complete	0x2C	Status, Connection_Handle, BD_ADDR, Link Type, Transmission_Interval, Retransmission Window, Rx_Packet_Length, Tx_Packet_Length Air Mode

Description:

The Synchronous Connection Complete event indicates to both the Hosts that a new Synchronous connection has been established. This event also indicates to the Host, which issued the Setup_Synchronous_Connection, or Accept_Synchronous_Connection_Request or Reject_Synchronous_Connection_Request command and then received a Command Status event, if the issued command failed or was successful.

Event Parameters:

Status: *1 octet*

Value	Parameter Description
0x00	Connection successfully completed.
0x01 –0xFF	Connection failed to complete.See “Error Codes” on page 291 [Part D] for error codes and description.

Connection_Handle: *2 octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection between two Bluetooth devices. Range: 0x0000-0x0EFF (0x0F00 – 0x0FFF Reserved for future use)

BD_ADDR: *6 octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the other connected device forming the connection.

Link_Type: *Size: 1 Octet*

Value	Parameter Description
0x00	SCO Connection
0x01	Reserved
0x02	eSCO Connection
0x03 – 0xFF	Reserved

Transmission_Interval: *1 octets*

Value	Parameter Description
0xXX	Time between two consecutive eSCO instants measured in slots. Must be zero for SCO links.

Retransmission window: *1 octets*

Value	Parameter Description
0xXX	The size of the retransmission window measured in slots. Must be zero for SCO links.

Rx_Packet_Length: *2 octets*

Value	Parameter Description
0xXXXX	Length in bytes of the eSCO payload in the receive direction. Must be zero for SCO links.

Tx_Packet_Length: *2 octets*

Value	Parameter Description
0xXXXX	Length in bytes of the eSCO payload in the transmit direction. Must be zero for SCO links.

Air Mode: *Size: 1 Octet*

Value	Parameter Description
0x00	μ-law log
0x01	A-law log
0x02	CVSD
0x03	Transparent Data
0x04 – 0xFF	Reserved

7.7.36 Synchronous Connection Changed event

Event	Event Code	Event Parameters
Synchronous Connection Changed	0x2D	Status, Connection_Handle, Transmission_Interval, Retransmission Window, Rx_Packet_Length, Tx_Packet_Length

Description:

The Synchronous Connection Changed event indicates to the Host that an existing Synchronous connection has been reconfigured. This event also indicates to the initiating Host (if the change was host initiated) if the issued command failed or was successful.

Command Parameters:

Status: *1 octet*

Value	Parameter Description
0x00	Connection successfully reconfigured.
0x01 –0xFF	Reconfiguration failed to complete. See “Error Codes” on page 291 [Part D] for error codes and description.

Connection_Handle: *2 octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle to be used to identify a connection between two Bluetooth devices. Range: 0x0000-0x0EFF (0x0F00 – 0x0FFF Reserved for future use)

Transmission_Interval: *1 octet*

Value	Parameter Description
0xXX	Time between two consecutive SCO/eSCO instants measured in slots.

Retransmission window: *1 octet*

Value	Parameter Description
0xXX	The size of the retransmission window measured in slots. Must be zero for SCO links.

Rx_Packet_Length: *2 octets*

Value	Parameter Description
0xXXXX	Length in bytes of the SCO/eSCO payload in the receive direction.

Tx_Packet_Length: *2 octets*

Value	Parameter Description
0xXXXX	Length in bytes of the SCO/eSCO payload in the transmit direction.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 338

8 LIST OF FIGURES

Figure 1.1: Overview of the Lower Software Layers 317
Figure 1.2: End to End Overview of Lower Software Layers to Transfer Data 318
Figure 5.1: HCI Command Packet 352
Figure 5.2: HCI ACL Data Packet 353
Figure 5.3: HCI Synchronous Data Packet 355
Figure 5.4: HCI Event Packet 356
Figure 7.1: Local Loopback Mode 533
Figure 7.2: Remote Loopback Mode 533

Petitioner Kiosoft Exhibit 1011 Part 2
Page 340

9 LIST OF TABLES

Table 3.1: Overview of commands and events 321
Table 3.2: Generic events 322
Table 3.3: Controller flow control 323
Table 3.4: Controller information 323
Table 3.5: Controller configuration 324
Table 3.6: Device discovery 325
Table 3.7: Connection setup 327
Table 3.8: Remote information 329
Table 3.9: Synchronous connections 330
Table 3.10: Connection state 331
Table 3.11: Piconet structure 332
Table 3.12: Quality of service 333
Table 3.13: Physical links 334
Table 3.14: Controller flow control. 335
Table 3.15: Link information 336
Table 3.16: Authentication and encryption 337
Table 3.17: Testing 339
Table 3.18: Alphabetical list of commands and events. 340

APPENDIX A: DEPRECATED COMMANDS, EVENTS AND CONFIGURATION PARAMETERS

Commands, events and configuration parameters in this section were in prior versions of the specification, but have been determined not to be required.

They may be implemented by a controller to allow for backwards compatibility with a host utilizing a prior version of the specification.

A host should not use these commands.

Contents:

Page Scan Mode 658

Read Page Scan Mode Command 659

Write Page Scan Mode Command 660

Read Country Code Command 661

Add SCO Connection Command 662

Page Scan Mode Change Event 664

Page Scan Mode

The Page_Scan_Mode parameter indicates the page scan mode that is used for default page scan. Currently one mandatory page scan mode and three optional page scan modes are defined. Following an inquiry response, if the Baseband timer T_mandatory_pscan has not expired, the mandatory page scan mode must be applied. For details see the “Baseband Specification” on page 45 [Part B] (Keyword: Page-Scan-Mode, FHS-Packet, T_mandatory_pscan)

Value	Parameter Description
0x00	Mandatory Page Scan Mode
0x01	Optional Page Scan Mode I
0x02	Optional Page Scan Mode II
0x03	Optional Page Scan Mode III
0x04-0xFF	Reserved

Read Page Scan Mode Command

Command	OGF	OCF	Command Parameters	Return Parameters
HCI_Read_Page_Scan_Mode	0x03	0x003D		Status, Page_Scan_Mode

Description:

This command is used to read the default Page Scan Mode configuration parameter of the local Bluetooth device. See “Page Scan Mode” on page 586..

Command Parameters:

None

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Page_Scan_Mode command succeeded.
0x01-0xFF	Read_Page_Scan_Mode command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Page_Scan_Mode: *Size: 1 Octet*

Value	Parameter Description
See Appendix A, “Page Scan Mode” on page 586.	

Event(s) generated (unless masked away):

When the Read_Page_Scan_Mode command has completed, a Command Complete event will be generated.

Write Page Scan Mode Command

OGF: 0x03 (Controller and baseband commands)

Command	OGF	OCF	Command Parameters	Return Parameters
HCI_Write_Page_Scan_Mode	0x03	0x003E	Page_Scan_Mode	Status

Description:

This command is used to write the default Page Scan Mode configuration parameter of the local Bluetooth device. See “Page Scan Mode” on page 586.

Command Parameters:

Page_Scan_Mode: *Size: 1 Octet*

Value	Parameter Description
See Appendix A, “Page Scan Mode” on page 586.	

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Write_Page_Scan_Mode command succeeded.
0x01-0xFF	Write_Page_Scan_Mode command failed. See “Error Codes” on page 291 [Part D] for list of Error Codes.

Event(s) generated (unless masked away):

When the Write_Page_Scan_Mode command has completed, a Command Complete event will be generated.

Read Country Code Command

Command	OGF	OCF	Command Parameters	Return Parameters
HCI_Read_Country_Code	0x05	0x0007		Status, Country_Code

Description:

This command will read the value for the Country_Code return parameter. The Country_Code defines which range of frequency band of the ISM 2.4 GHz band will be used by the device. Each country has local regulatory bodies regulating which ISM 2.4 GHz frequency ranges can be used.

Command Parameters:

None.

Return Parameters:

Status: *Size: 1 Octet*

Value	Parameter Description
0x00	Read_Country_Code command succeeded.
0x01-0xFF	Read_Country_Code command failed. See “Error Codes” on page 291 [Part D] for error codes and description.

Country_Code: *Size: 1 Octet*

Value	Parameter Description
0x00	North America & Europe* and Japan
0x01	France
0x04-FF	Reserved for Future Use.

*. Except France

Event(s) generated (unless masked away):

When the Read_Country_Code command has completed, a Command Complete event will be generated.

Add SCO Connection Command

Command	OGF	OCF	Command Parameters	Return Parameters
HCI_Add_SCO_Connection	0x01	0x0007	Connection_Handle, Packet_Type	

Description:

This command will cause the link manager to create a SCO connection using the ACL connection specified by the Connection_Handle command parameter. This command causes the local Bluetooth device to create a SCO connection. The Link Manager will determine how the new connection is established. This connection is determined by the current state of the device, its piconet, and the state of the device to be connected. The Packet_Type command parameter specifies which packet types the Link Manager should use for the connection. The Link Manager must only use the packet type(s) specified by the Packet_Type command parameter for sending HCI SCO Data Packets. Multiple packet types may be specified for the Packet_Type command parameter by performing a bitwise OR operation of the different packet types. The Link Manager may choose which packet type is to be used from the list of acceptable packet types. A Connection Handle for this connection is returned in the Connection Complete event (see below).
Note: An SCO connection can only be created when an ACL connection already exists and when it is not put in park. For a definition of the different packet types, see the “Baseband Specification” on page 45 [Part B].

Note: At least one packet type must be specified. The Host should enable as many packet types as possible for the Link Manager to perform efficiently. However, the Host must not enable packet types that the local device does not support.

Command Parameters:

Connection_Handle *Size 2 Octets (12 Bits meaningful)*

Value	Parameter Description
0xXXXX	Connection Handle for the ACL connection being used to create an SCO connection. Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)

Packet_Type: *Size: 2 Octets*

Value	Parameter Description
0x0001	Reserved for future use.
0x0002	Reserved for future use.
0x0004	Reserved for future use.
0x0008	Reserved for future use.
0x0010	Reserved for future use.
0x0020	HV1
0x0040	HV2
0x0080	HV3
0x0100	Reserved for future use.
0x0200	Reserved for future use.
0x0400	Reserved for future use.
0x0800	Reserved for future use.
0x1000	Reserved for future use.
0x2000	Reserved for future use.
0x4000	Reserved for future use.
0x8000	Reserved for future use.

Return Parameters:

None.

Event(s) generated (unless masked away):

When the Controller receives the Add_SCO_Connection command, it sends the Command Status event to the Host. In addition, when the LM determines the connection is established, the local Controller will send a Connection Complete event to its Host, and the remote Controller will send a Connection Complete event or a Synchronous Connection Complete event to the Host. The Connection Complete event contains the Connection Handle if this command is successful.
Note: no Command Complete event will be sent by the Controller to indicate that this command has been completed. Instead, the Connection Complete event will indicate that this command has been completed.

Page Scan Mode Change Event

Event	Event Code	Event Parameters
Page Scan Mode Change	0x1F	BD_ADDR, Page_Scan_Mode

Description:

The Page Scan Mode Change event indicates that the connected remote Bluetooth device with the specified BD_ADDR has successfully changed the Page_Scan_Mode.

Event Parameters:

BD_ADDR: *Size: 6 Octets*

Value	Parameter Description
0xXXXXXXXXXXXX	BD_ADDR of the remote device.

Page_Scan_Mode: *Size: 1 Octet*

Value	Parameter Description
0x00	Mandatory Page Scan Mode.
0x01	Optional Page Scan Mode I.
0x02	Optional Page Scan Mode II.
0x03	Optional Page Scan Mode III.
0x04 – 0xFF	Reserved.



MESSAGE SEQUENCE CHARTS

Between Host and Host Controller/Link Manager

Examples of interactions between Host Controller Interface Commands and Events and Link Manager Protocol Data Units are represented in the form of message sequence charts. These charts show typical interactions and do not indicate all possible protocol behavior.

CONTENTS

1 1Introduction 599
1.1 Notation 599
1.2 Flow of Control 600
1.3 Example MSC 600
2 Services Without Connection Request 601
2.1 Remote Name Request 601
Step 1: 601
Step 2a: 601
Step 2b: 602
2.2 One-time Inquiry 602
Step 1: 602
Step 2: 603
Step 3a: 603
Step 3b: 604
2.3 Periodic Inquiry 604
Step 1: 604
Step 2: 605
Step 3: 605
Step 4: 605
3 ACL Connection Establishment and Detachment 607
3.1 Connection Setup 608
Step 1: 608
Step 2: 608
Step 3: 608
Step 4a: 609
Step 4b: 609
Step 4c: 610
Step 5: 610
Step 6: 611
Step 7a: 612
Step 7b: 613
Step 8: 613
Step 9: 614
Step 10: 614

4 Optional Activities After ACL Connection Establishment.....615
4.1 Authentication Requested.....615
Step 1: 615
4.2 Set Connection Encryption 616
Step 1: 616
Step 2: 616
4.3 Change Connection Link Key 617
Step 1: 617
4.4 Master Link Key 618
Step 1: 618
Step 2: 619
4.5 Read Remote Supported Features.....620
Step 1: 620
4.6 Read Remote Extended Features 620
Step 1: 620
4.7 Read Clock Offset.....621
Step 1: 621
4.8 Read Remote Version Information.....621
Step 1: 621
4.9 QOS Setup 622
Step 1: 622
4.10 Switch Role.....622
Step 1a: 622
Step 1b: 623
Step 2: 623
5 Synchronous Connection Establishment and Detachment.....625
5.1 Synchronous Connection Setup 625
Step 1a: 625
Step 1b: 626
Step 1c: 626
Step 1d: 627
Step 1e: 627
Step 2a: 628
Step 2b: 628
Step 3a: 629
Step 3b: 629

6 Sniff, Hold and Park 631
6.1 Sniff Mode 631
Step 1: 631
Step 2: 631
6.2 Hold Mode 632
Step 1a: 632
Step 1b: 632
Step 1c: 633
Step 2: 633
6.3 Park State 634
Step 1a: 634
Step 1b: 634
Step 2: 635
Step 3a: 635
Step 3b: 636
7 Buffer Management, Flow Control 637
Step 1: 637
Step 2: 638
8 Loopback Mode 639
8.1 Local Loopback Mode 639
Step 1: 639
Step 2a: 639
Step 2b: 640
Step 3: 640
8.2 Remote Loopback Mode 641
Step 1: 641
Step 2: 641
Step 3: 642

9 List of Figures 643

1 INTRODUCTION

This section shows typical interactions between Host Controller Interface (HCI) Commands and Events and Link Manager (LM) Protocol Data Units (PDU). It focuses on the message sequence charts (MSCs) for the procedures specified in [3] “Bluetooth Host Controller Interface Functional Specification” with regard to LM Procedures from [2] “Link Manager Protocol”.

This section illustrates only the most useful scenarios, it does not cover all possible alternatives. Furthermore, the message sequence charts do not consider errors over the air interface or host interface. In all message sequence charts it is assumed that all events are not masked, so the Host Controller will not filter out any events.

The sequence of messages in these message sequence charts is for illustrative purposes. The messages may be sent in a different order where allowed by the Link Manager or HCI sections. If any of these charts differ with text in the Baseband, Link Manager, or HCI sections, the text in those sections shall be considered normative. This section is informative.

1.1 NOTATION

The notation used in the message sequence charts (MSCs) consists of ovals, elongated hexagons, boxes, lines, and arrows. The vertical lines terminated on the top by a shadow box and at the bottom by solid oval indicate a protocol entity that resides in a device. MSCs describe interactions between these entities and states those entities may be in.

The following symbols represent interactions and states:

Oval	Defines the context for the message sequence chart.
Hexagon	Indicates a condition needed to start the transactions below this hexagon. The location and width of the Hexagon indicates which entity or entities make this decision.
Box	Replaces a group of transactions. May indicate a user action, or a procedure in the baseband.
Dashed Box	Optional group of transactions.
Solid Arrow	Represents a message, signal or transaction. Can be used to show LMP and HCI traffic. Some baseband packet traffic is also shown. These are prefixed by BB followed by either the type of packet, or an indication that there is an ACK signal in a packet.
Dashed Arrow	Represents a optional message, signal or transaction. Can be used to show LMP and HCI traffic.

1.2 FLOW OF CONTROL

Some message sequences are split into several charts. These charts are marked in sequence with different step numbers with multiple paths through with optional letters after the step numbers. Numbers indicate normal or required ordering. The letters represent alternative paths. For example, Step 4 is after Step 3, and Step 5a could be executed instead of Step 5b.

1.3 EXAMPLE MSC

The protocol entities represented in the example shown in Figure 1.1 on page 600 illustrate the interactions of two devices named A and B. Note that each device includes a Host and a LM entity in this example. Other MSCs in this section may show the interactions of more than two devices.

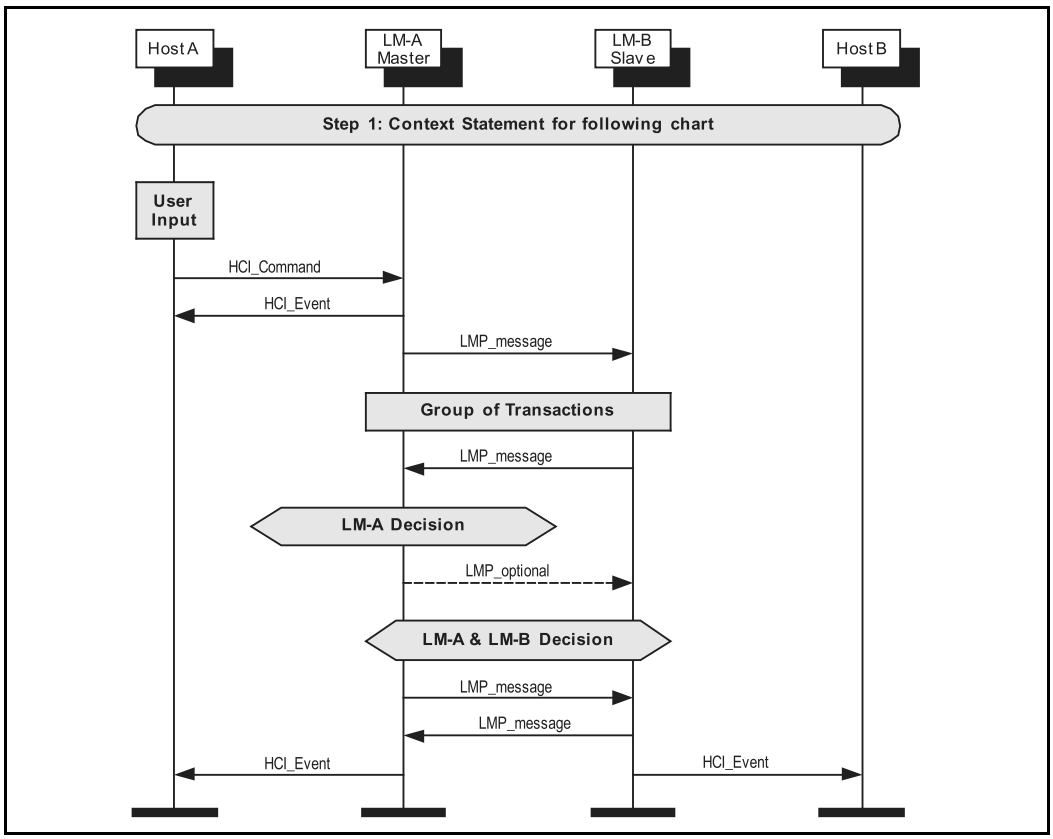


Figure 1.1: Example MSC.

2 SERVICES WITHOUT CONNECTION REQUEST

2.1 REMOTE NAME REQUEST

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

Step 1: The host sends an HCI_Remote_Name_Request command expecting that its local device will automatically try to connect to the remote device. (See Figure 2.1 on page 601)

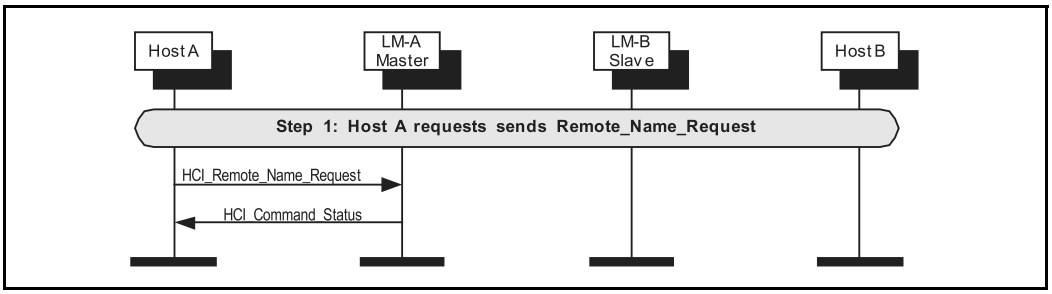


Figure 2.1: Remote name request.

Step 2a: If an ACL Connection does not exist device A pages device B. After the Baseband paging procedure, the local device attempts to get the name, disconnect, and return the name of the remote device to the Host. (See Figure 2.2 on page 601)

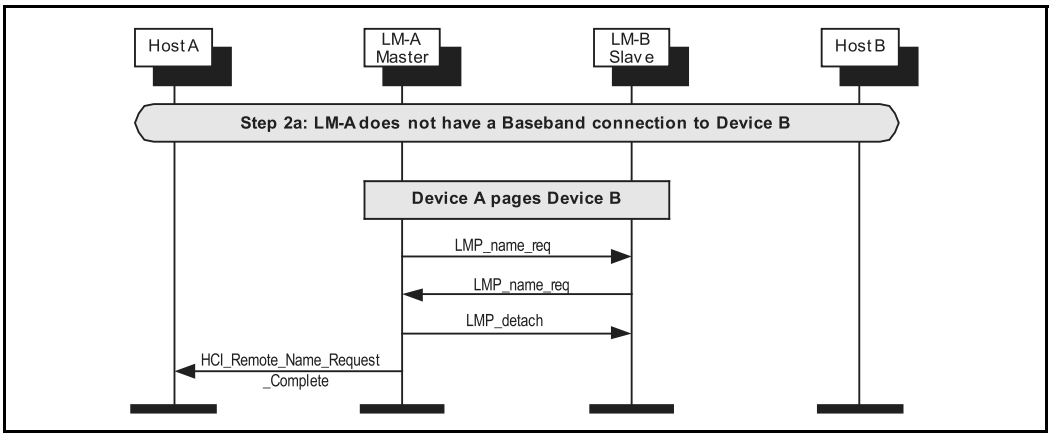


Figure 2.2: Remote name request if no current baseband connection.

Step 2b: If an ACL Connection exists when the request is made, then the Remote Name Request procedure will be executed like an optional service. No Paging and no ACL disconnect is done. (See Figure 2.3 on page 602)

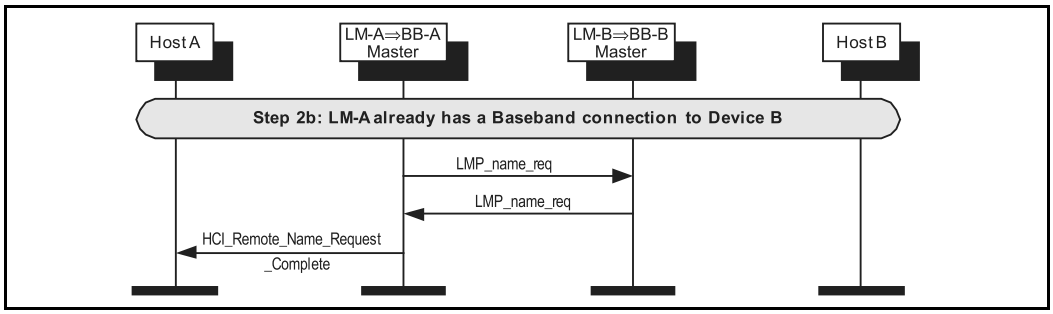


Figure 2.3: Remote name request with baseband connection.

2.2 ONE-TIME INQUIRY

Inquiry is used to detect and collect nearby devices.

Step 1: The host sends an HCI_Inquiry command. (See Figure 2.4 on page 602)

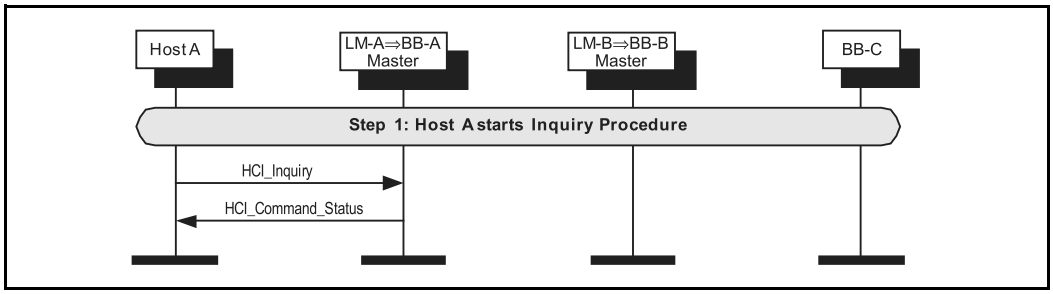


Figure 2.4: Host A starts inquiry procedure.

Step 2: The Controller will start the Baseband inquiry procedure with the specified Inquiry Access Code and Inquiry Length. When Inquiry Responses are received, the Controller extracts the required information and returns the information related to the found devices using one or more Inquiry Result events to the Host. (See Figure 2.5 on page 603)

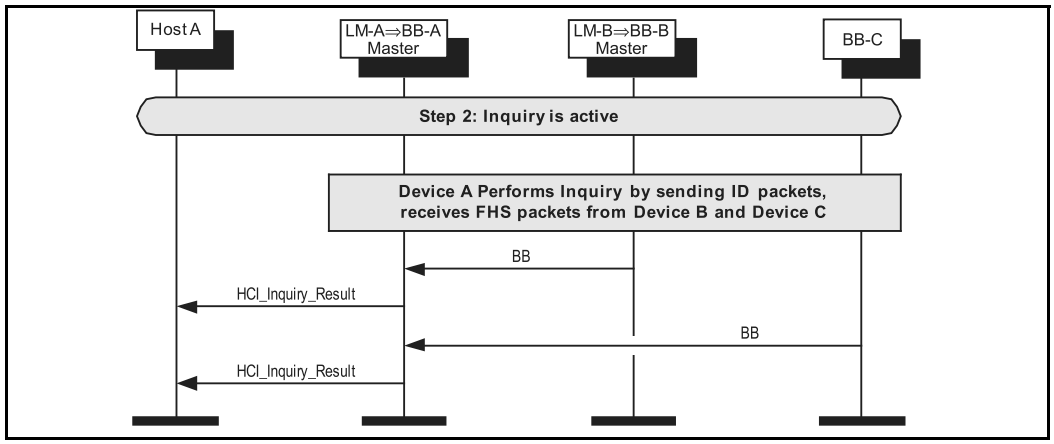


Figure 2.5: LM-A performs inquiry and reports result..

Step 3a: If the host wishes to terminate an Inquiry, the HCI_Inquiry_Cancel command is used to immediately stop the inquiry procedure.
(See Figure 2.6 on page 603)

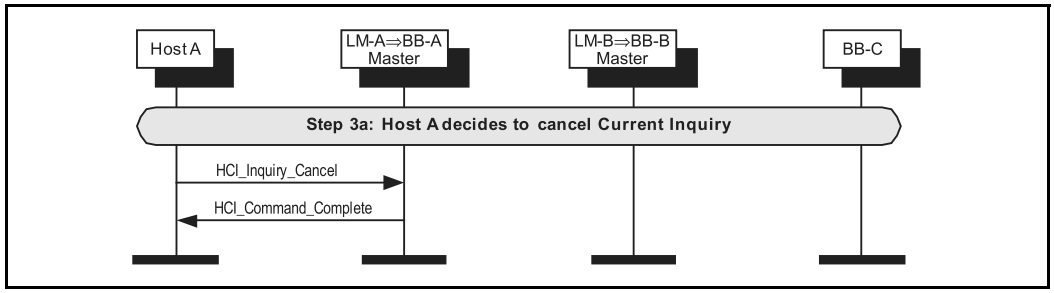


Figure 2.6: Host A cancels inquiry.

Step 3b: If the Inquiry procedure is completed due to the number of results obtained, or the Inquiry Length has expired, an Inquiry Complete event is returned to the Host. (See Figure 2.7 on page 604)

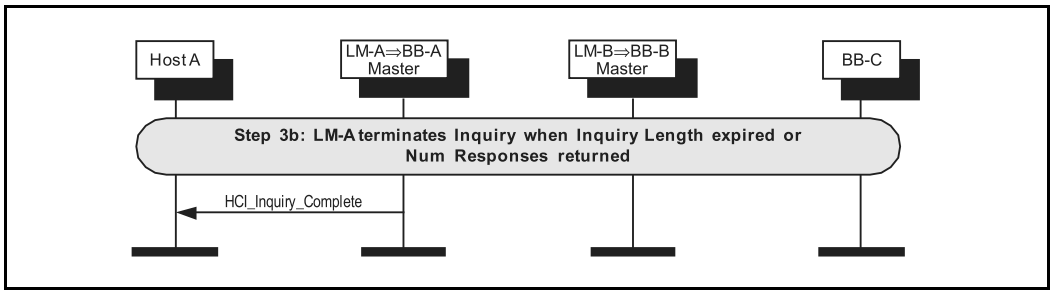


Figure 2.7: LM-A terminates current inquiry.

2.3 PERIODIC INQUIRY

Periodic inquiry is used when the inquiry procedure is to be repeated periodically.

Step 1: The hosts sends an HCI_Periodic_Inquiry_Mode command. (See Figure 2.8 on page 604)

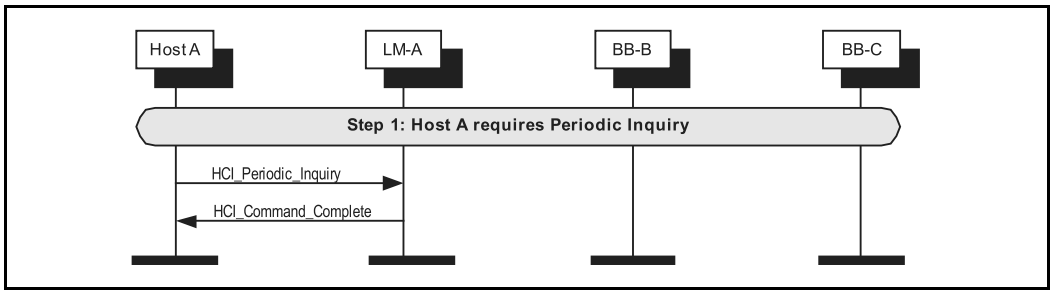


Figure 2.8: Host A starts periodic inquiry.

Step 2: The Controller will start a periodic Inquiry. In the inquiry cycle, one or several Inquiry Result events will be returned. (See Figure 2.9 on page 605)

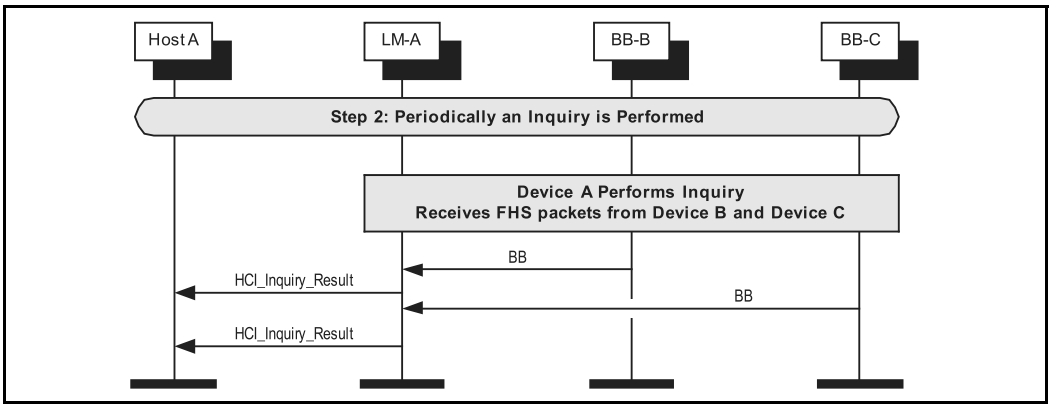


Figure 2.9: LM-A periodically performs an inquiry and reports result.

Step 3: An Inquiry Complete event will be returned to the Host when the current periodic inquiry has finished. (See Figure 2.10 on page 605)

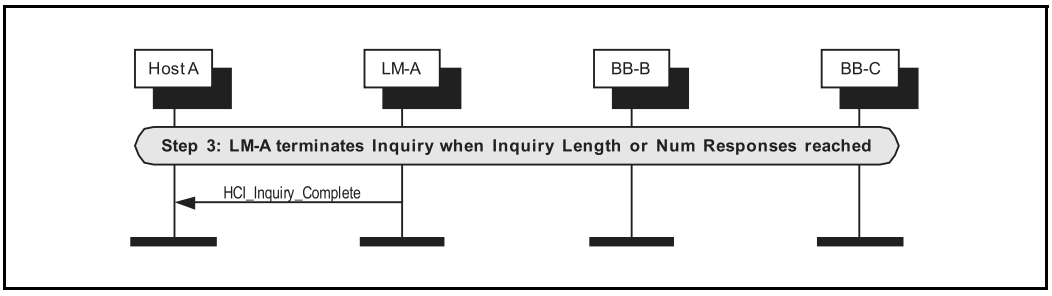


Figure 2.10: LM-A terminates current inquiry.

Step 4: The periodic Inquiry can be stopped using the HCI_Exit_Periodic_Inquiry_Mode command. (See Figure 2.11 on page 605)

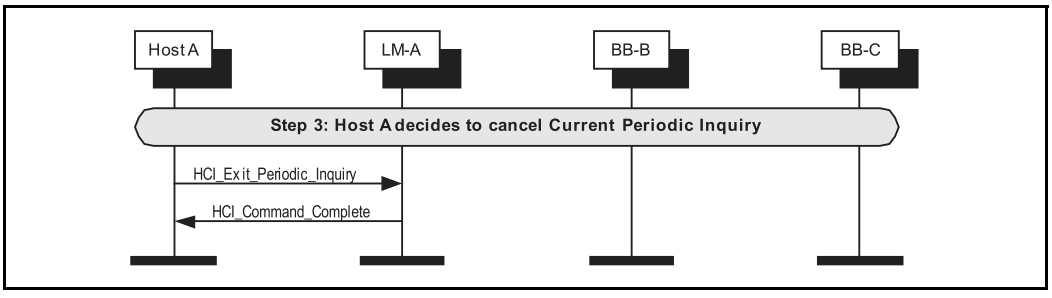


Figure 2.11: Host A decides to exit periodic inquiry.

3 ACL CONNECTION ESTABLISHMENT AND DETACHMENT

A flow diagram of the establishment and detachment of a connection between two devices is shown in Figure 3.1 on page 607. The process is illustrated in 9 distinct steps. A number of these steps may be optionally performed, such as authentication and encryption. Some steps are required, such as the Connection Request and Setup Complete steps. The steps in the overview diagram directly relate to the steps in the following message sequence charts.

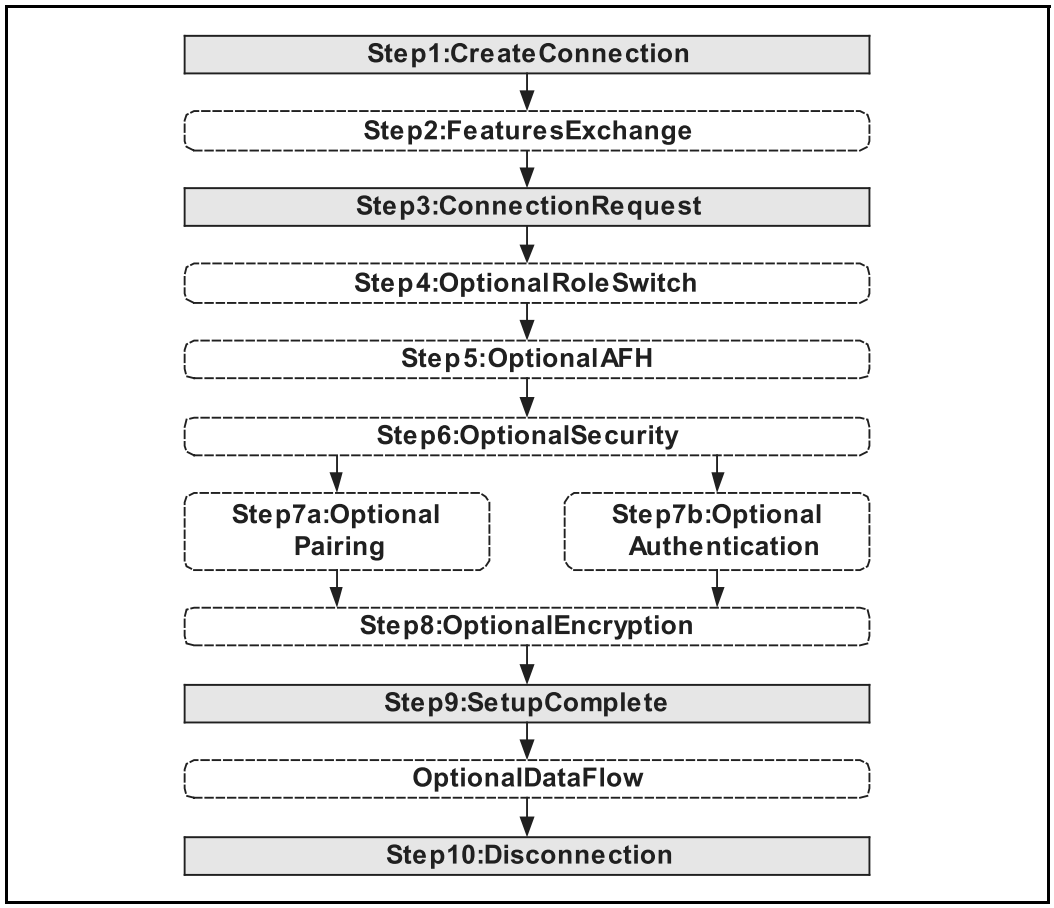


Figure 3.1: Overview diagram for connection setup.

3.1 CONNECTION SETUP

Step 1: The host sends an HCI_Create_Connection command to the Controller. The Controller then performs a Baseband paging procedure with the specified BD_ADDR. (See Figure 3.2 on page 608)

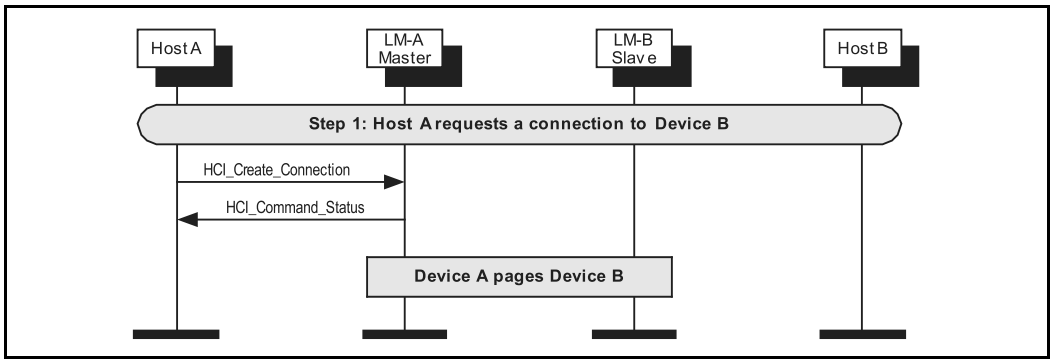


Figure 3.2: Host A requests connection with device B.

Step 2: Optionally, the LM may decide to exchange features. (See Figure 3.3 on page 608)

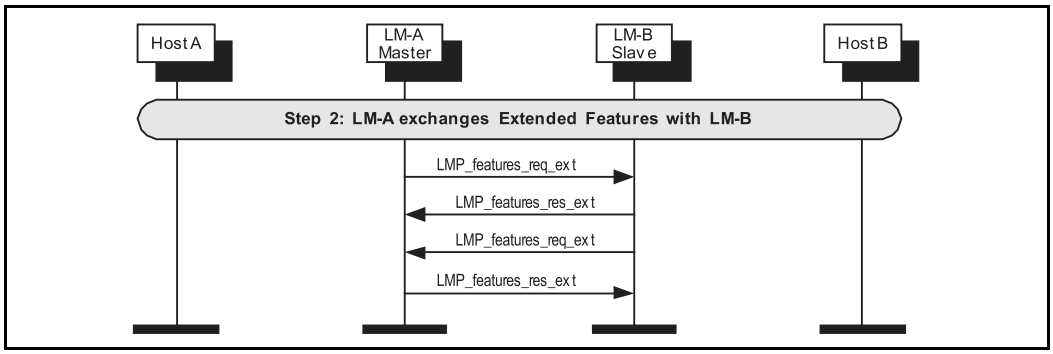


Figure 3.3: LM-A and LM-B exchange features.

Step 3: The LM on the master will request an LMP_host_connection_req PDU. The LM on the slave will then confirm that a connection is OK, and if so, what role is preferred. (See Figure 3.4 on page 608)

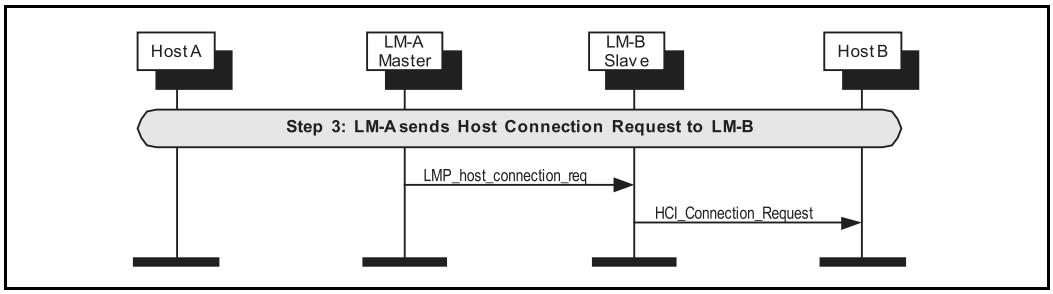


Figure 3.4: LM-A requests host connection.

Step 4a: The remote host rejects this connection, and the link is terminated. (See Figure 3.5 on page 609)

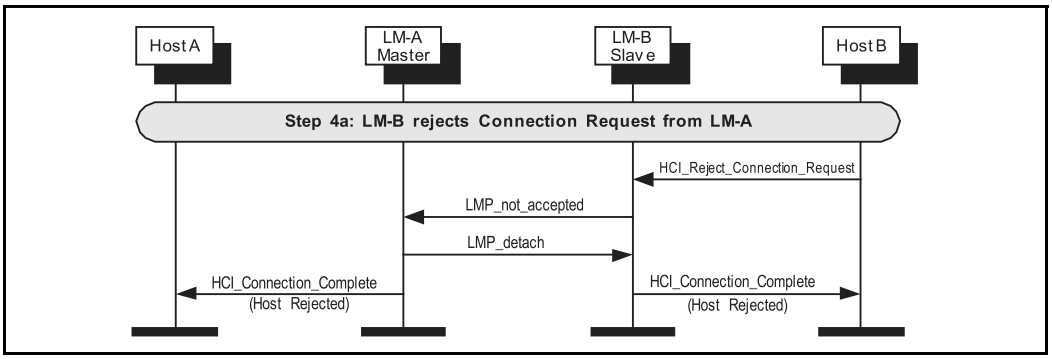


Figure 3.5: Device B rejects connection request.

Step 4b: The remote host accepts this connection. (See Figure 3.6 on page 609)

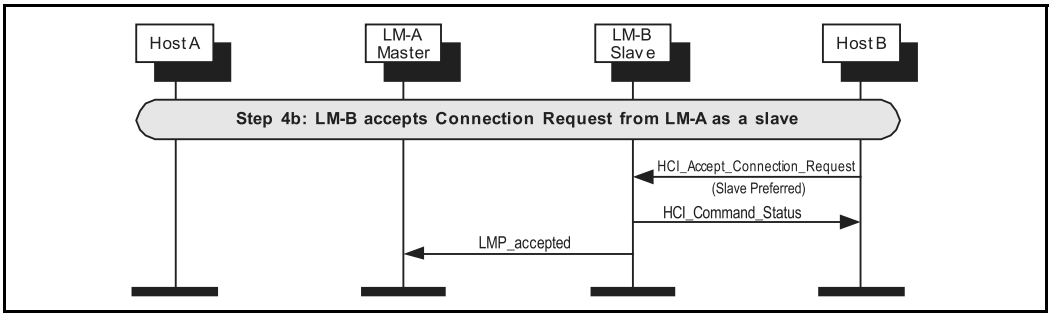


Figure 3.6: Device B accepts connection request.

Step 4c: The remote host accepts this connection but with the preference of being a master. This will cause a role switch to occur before the LMP_accepted for the LMP_host_connection_req PDU is sent. (See Figure 3.7 on page 610)

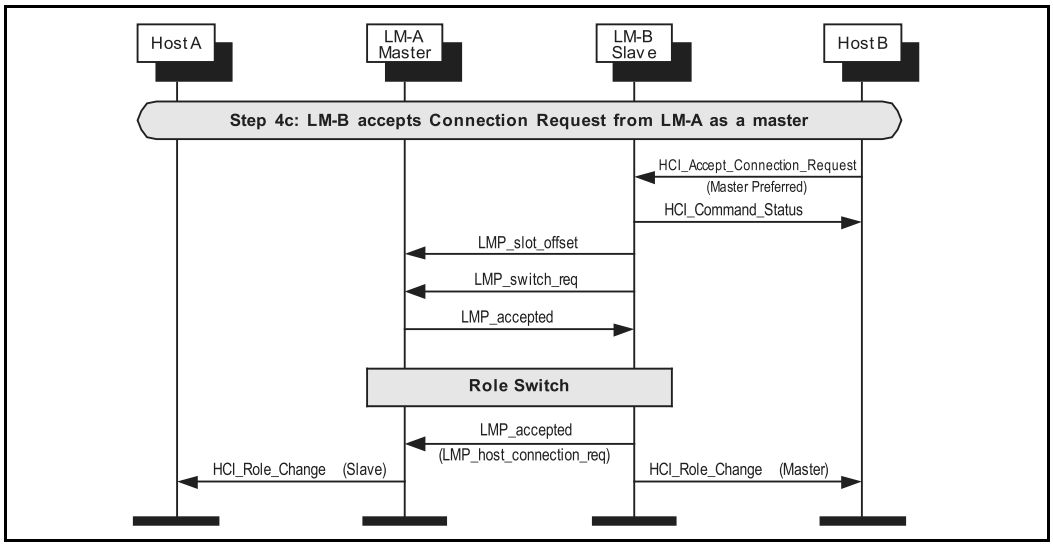


Figure 3.7: Device B accepts connection requests as master.

Step 5: After the features have been exchanged and AFH support is determined to be available, the master may at any time send an LMP_set_AFH and LMP_channel_classification_req PDU. (See Figure 3.8 on page 610)

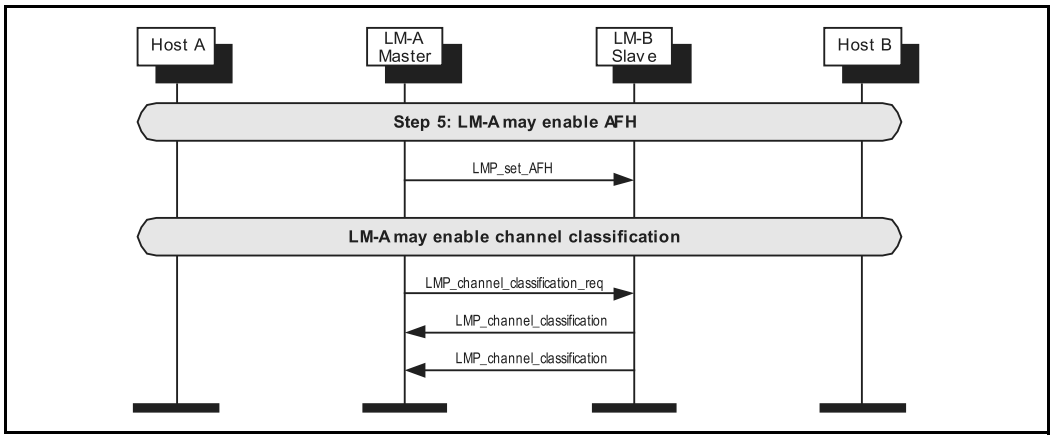


Figure 3.8: LM-A starts adaptive frequency hopping.

Step 6: The LM will request if authentication is required. It does this by requesting the Link Key for this connection from the Host. (See Figure 3.9 on page 611)

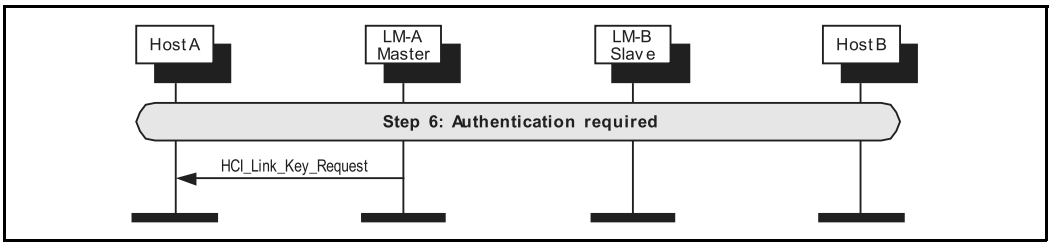


Figure 3.9: Authentication initiated.

Step 7a: If authentication is required by the higher layers and the devices to be connected do not have a common link key, a pairing procedure will be used. The LM will have requested a link key from the host for this connection. If there is a negative reply, then a PIN code will be requested. This PIN code will be requested on both sides of the connection, and authentication performed based on this PIN code. The last step is for the new link key for this connection is passed to the host so that it may store it for future connections.
(See Figure 3.10 on page 612)

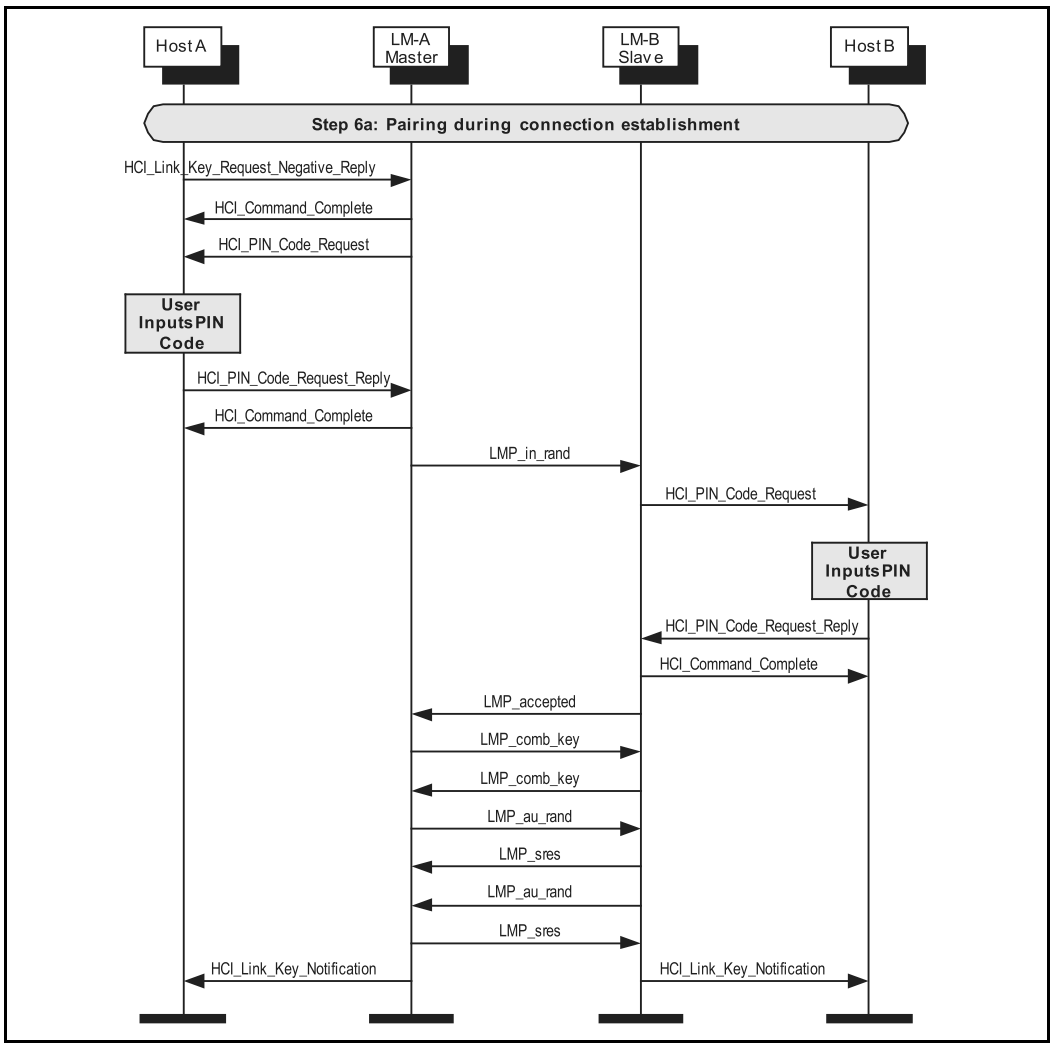


Figure 3.10: Pairing during connection setup.

Step 7b: If a common link key exists between the devices, then pairing is not needed. The LM will have asked for a link key from the host for this connection. If this is a positive reply, then the link key is used for authentication. If the configuration parameter Authentication_Enable is set, then the authentication procedure must be executed. This MSC only shows the case when Authentication_Enable is set on both sides. (See Figure 3.11 on page 613)

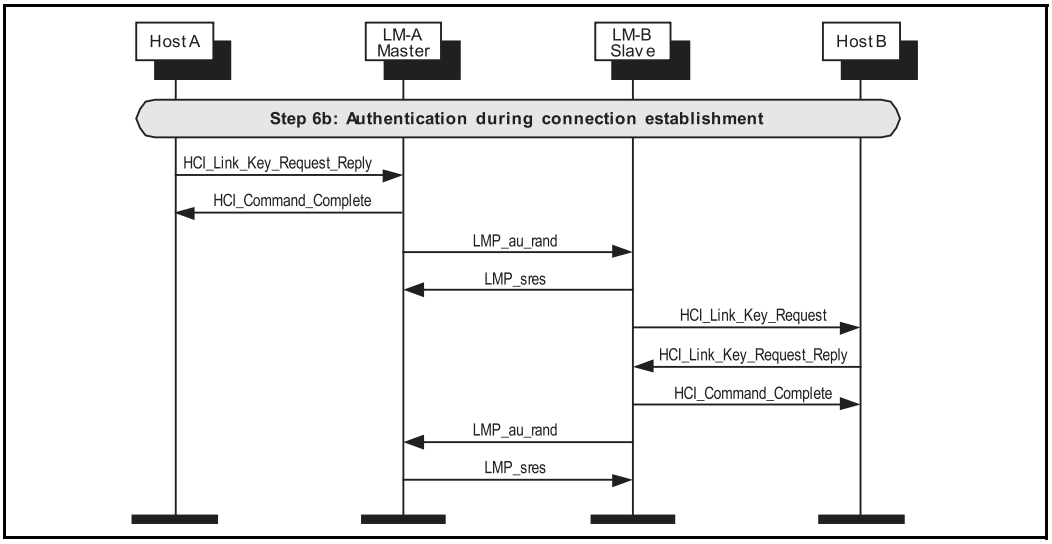


Figure 3.11: Authentication during connection setup.

Step 8: Once the pairing or authentication procedure is successful, the encryption procedure may be started. This MSC only shows the set up of an encrypted point-to-point connection. (See Figure 3.12 on page 613)

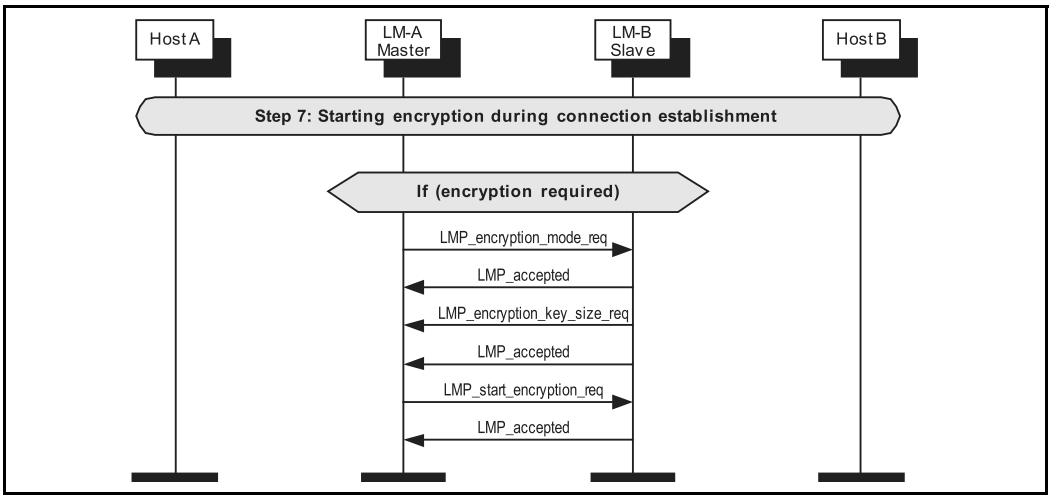


Figure 3.12: Starting encryption during connection setup.

Step 9: The LM's indicate that the connection is setup by sending LMP_setup_complete PDU. This will cause the Host to be notified of the new connection handle, and this connection may be used to send higher layer data such as L2CAP information. (See Figure 3.13 on page 614)

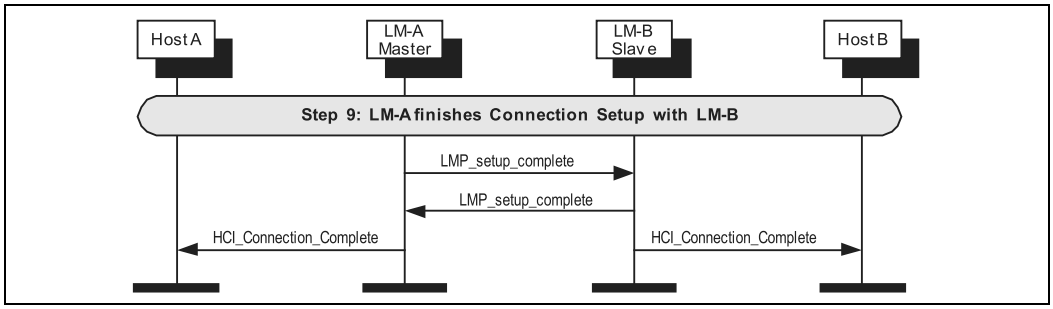


Figure 3.13: LM-A and LM-B finishes connection setup.

Step 10: Once the connection is no longer needed, either device may terminate the connection using the HCI_Disconnect command and LMP_detach message PDU. The disconnection procedure is one-sided and does not need an explicit acknowledgment from the remote LM. The use of ARQ Acknowledgment from the Baseband is needed to ensure that the remote LM has received the LMP_detach PDU. (See figure Figure 3.13 on page 614)

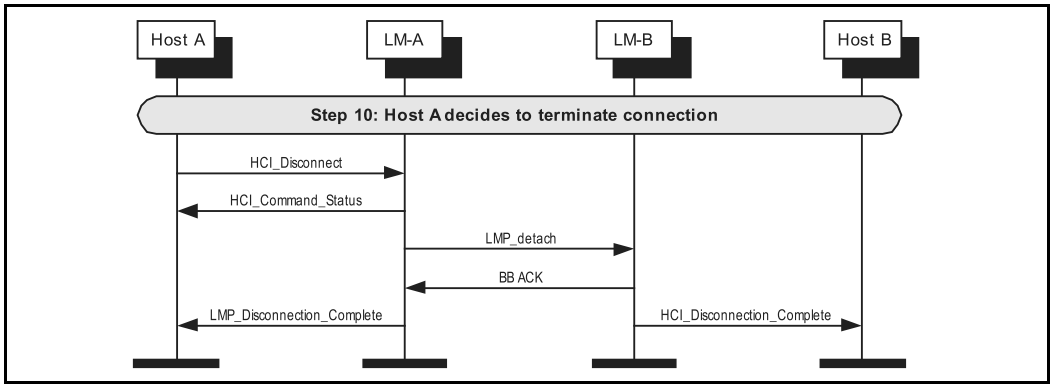


Figure 3.14: Host A decides to disconnect.

4 OPTIONAL ACTIVITIES AFTER ACL CONNECTION ESTABLISHMENT

4.1 AUTHENTICATION REQUESTED

Step 1: Authentication can be explicitly executed at any time after a connection has been established. If no Link Key is available then the Link Key is required from the Host. (See Figure 4.1 on page 615)

Note: If the Controller or LM and the Host do not have the Link Key a PIN Code Request event will be sent to the Host to request a PIN Code for pairing. A procedure identical to that used during Connection Setup (Section 3.1, Step 7a:) will be used. (See Figure 3.9 on page 611)

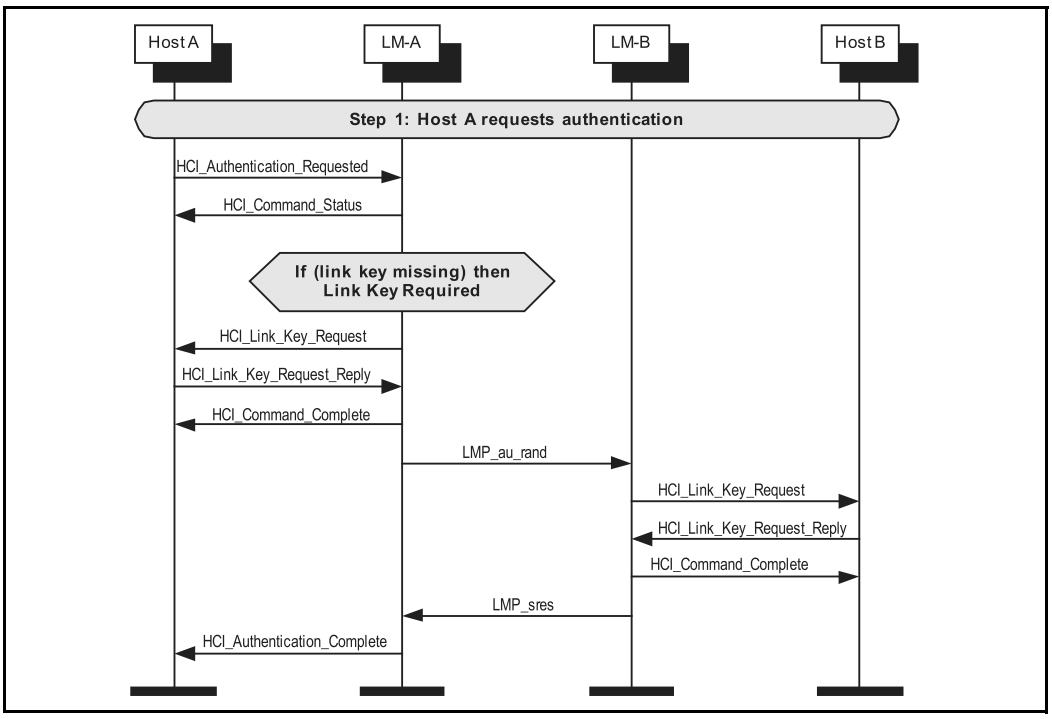


Figure 4.1: Authentication requested.

4.2 SET CONNECTION ENCRYPTION

Step 1: The host may at any time turn on encryption using the HCI_Set_Connection_Encryption command. This command can be originated from either the master or slave sides. Only the master side is shown in figure 4.2. If this command is sent from a slave, the only difference is that the LMP_encryption_mode_req PDU will be sent from the slave. The LMP_encryption_key_size_req and LMP_start_encryption_req PDUs will always be requested from the master. (See Figure 4.2 on page 616)

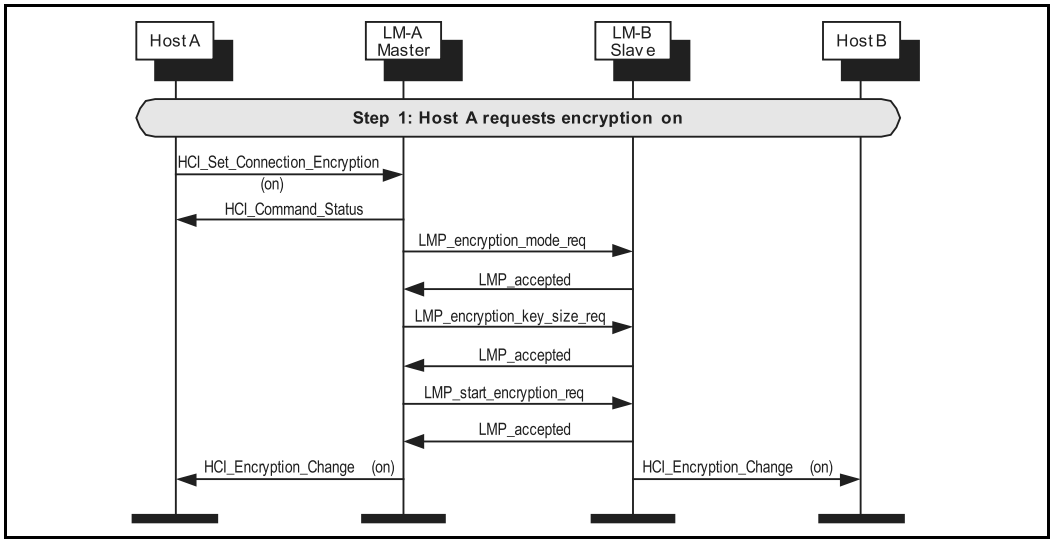


Figure 4.2: Encryption requested.

Step 2: To terminate the use of encryption, The HCI_Set_Connection_Encryption command is used. (See Figure 4.3 on page 616)

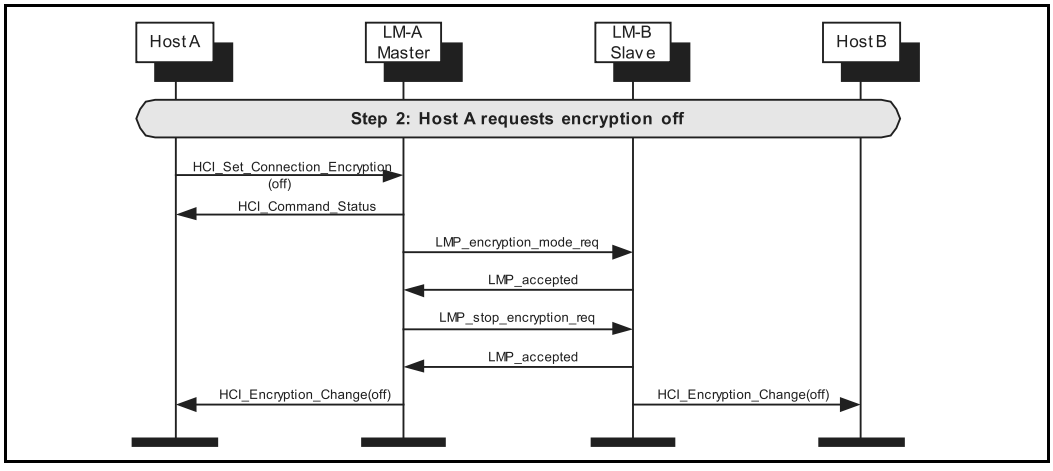


Figure 4.3: Encryption off requested.

4.3 CHANGE CONNECTION LINK KEY

Step 1: The master host (Host A) may change the connection link key using the HCI_Change_Connection_Link_Key command. A new link key will be generated and the hosts will be notified of this new link key. (See Figure 4.4 on page 617).

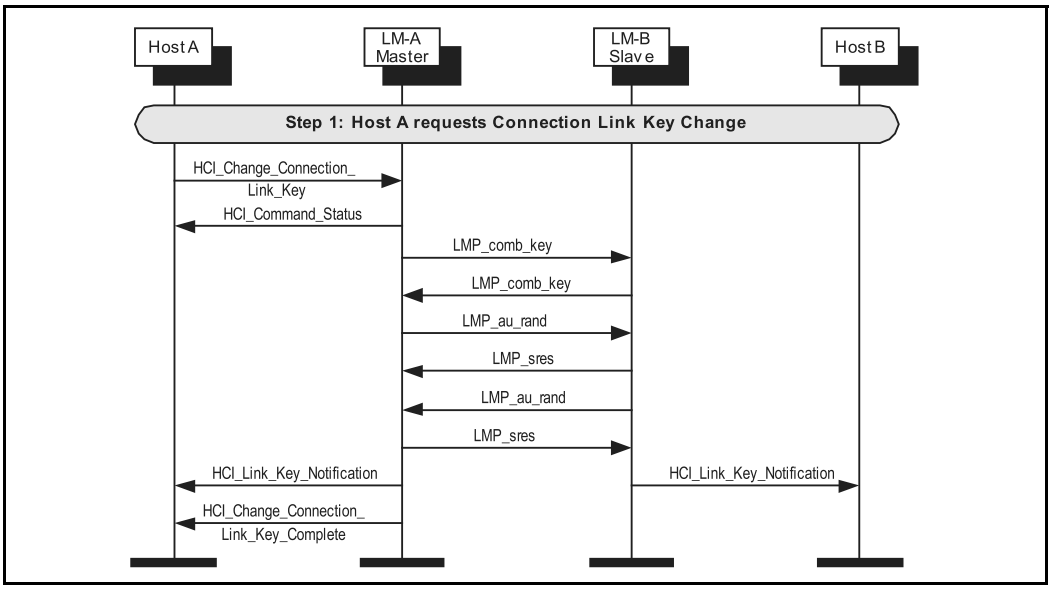


Figure 4.4: Change connection link key.

4.4 MASTER LINK KEY

Step 1: The host changes to a Master Link Key from a Semi-permanent Link Key using the HCI_Master_Link_Key command. (See Figure 4.5 on page 618)

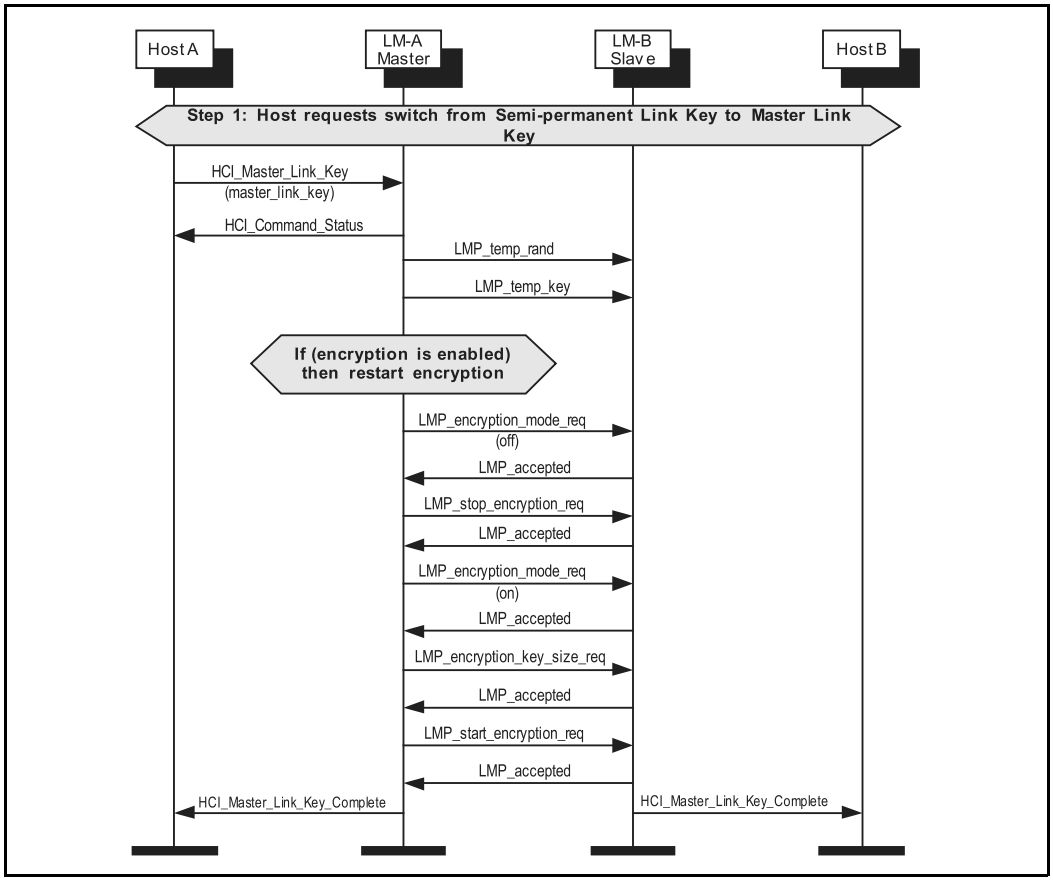


Figure 4.5: Change to master link key.

Step 2: The host changes to a Semi-permanent Link Key from a Master Link Key using the HCI_Master_Link_Key command. (See Figure 4.6 on page 619)

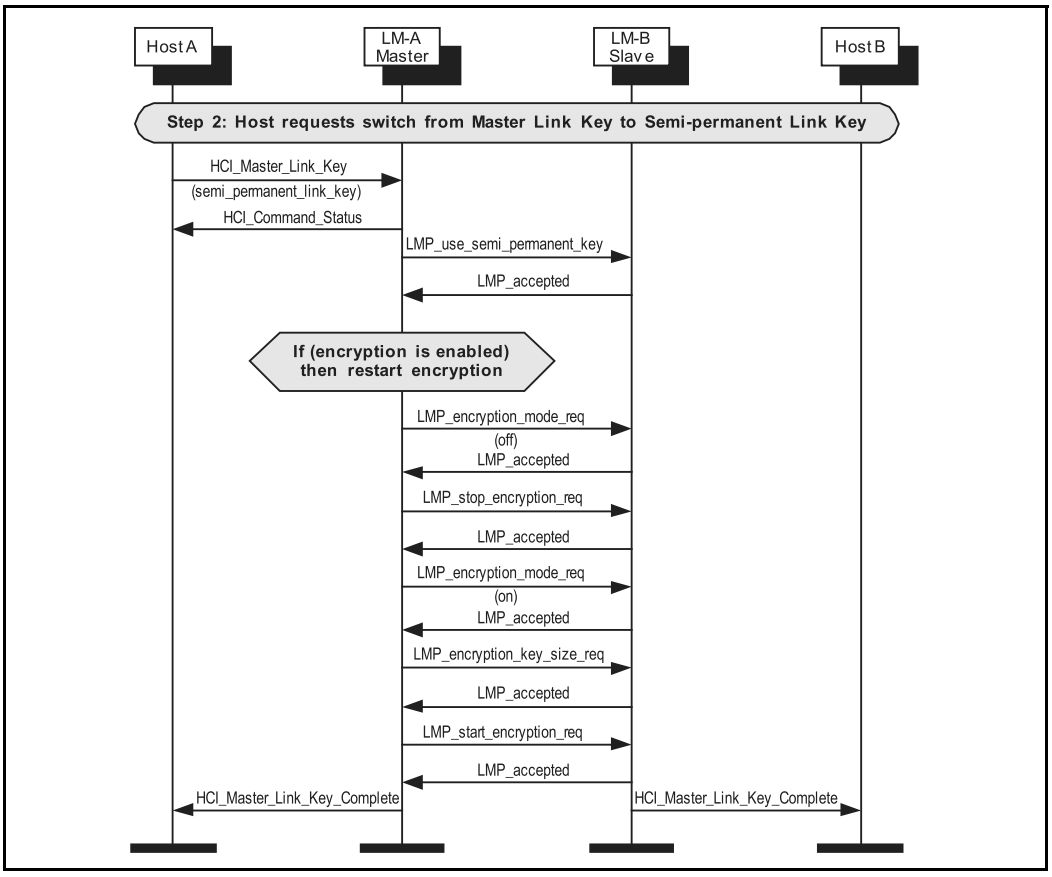


Figure 4.6: Change to semi permanent link key.

4.5 READ REMOTE SUPPORTED FEATURES

Using the HCI_Read_Remote_Supported_Features command the supported LMP Features of a remote device can be read. (See Figure 4.7 on page 620)

If the remote supported features have been obtained previously then the Controller may return them without sending any LMP PDUs.

Step 1: The host requests the supported features of a remote device.

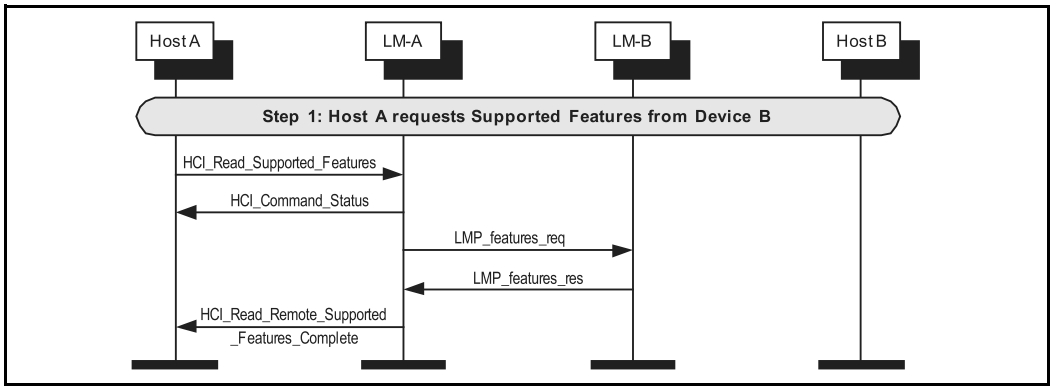


Figure 4.7: Read remote supported features.

4.6 READ REMOTE EXTENDED FEATURES

Using the HCI_Read_Remote_Extended_Features command the extended LMP features of a remote device can be read. (See Figure 4.8 on page 620)

If the remote extended features have been obtained previously then the Controller may return them without sending any LMP PDUs.

Step 1: The host requests the extended features of a remote device.

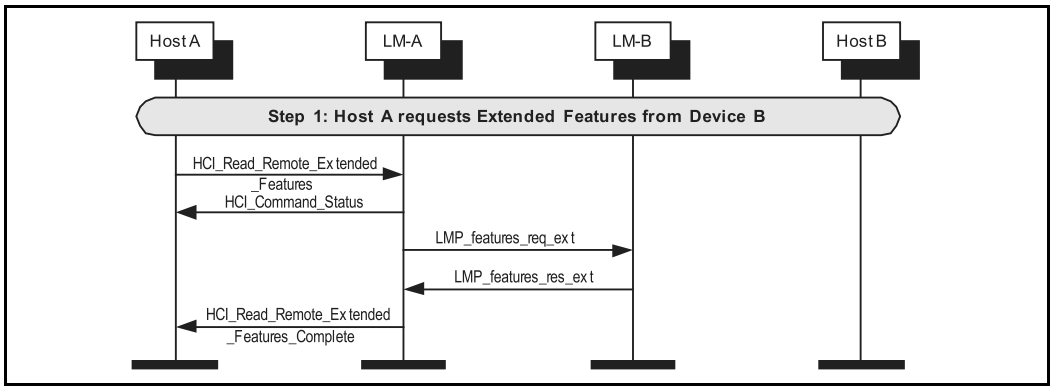


Figure 4.8: Read remote extended features.

4.7 READ CLOCK OFFSET

Using the HCI_Read_Clock_Offset command the device acting as the master can read the Clock Offset of a slave. The Clock Offset can be used to speed up the paging procedure in a later connection attempt. If the command is requested from the slave device, the Controller will directly return a Command Status event and a Read Clock Offset Complete event without sending any LMP PDUs. (See Figure 4.9 on page 621)

Step 1: The host requests the clock offset of a remote device.

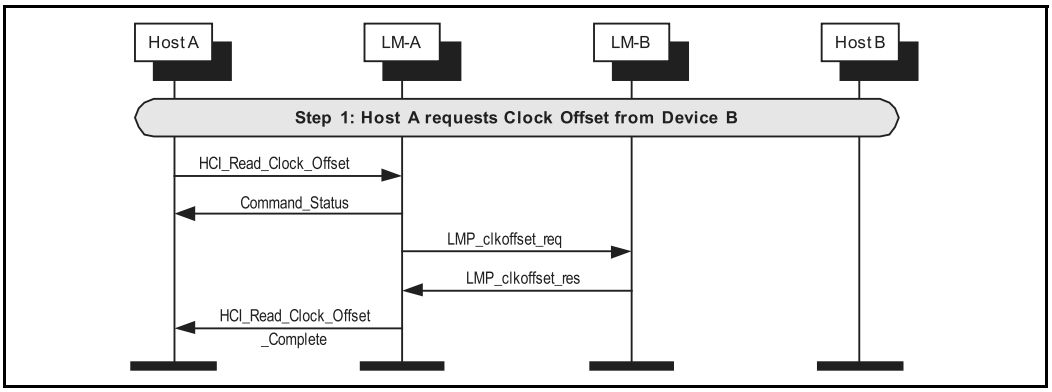


Figure 4.9: Read clock offset.

4.8 READ REMOTE VERSION INFORMATION

Using the HCI_Read_Remote_Version_Information command the version information of a remote device can be read. (See Figure 4.10 on page 621)

If the remote version information has been obtained previously then the Controller may return them without sending any LMP PDUs.

Step 1: The host requests the version information of a remote device.

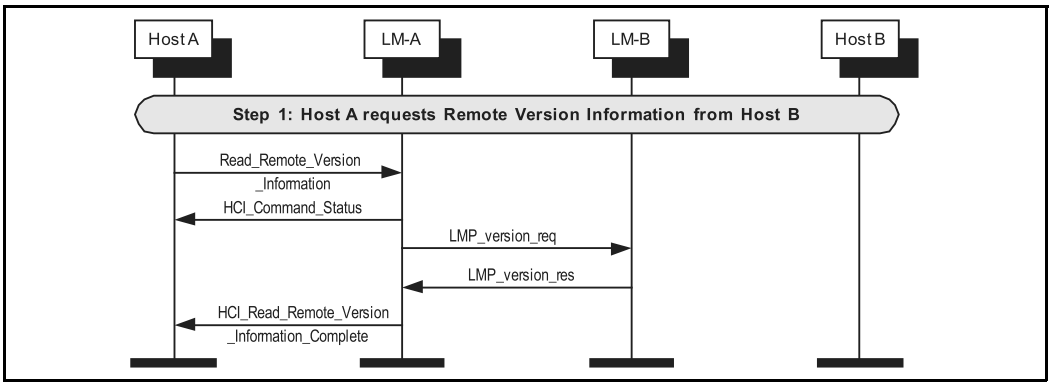


Figure 4.10: Read remote version information.

4.9 QOS SETUP

Using the HCI_Flow_Specification command the Quality of Service (QoS) and Flow Specification requirements of a connection can be notified to a Controller. The Controller may then change the quality of service parameters with a remote device. (See Figure 4.11 on page 622)

Step 1: The host sends QoS parameters to a remote device.

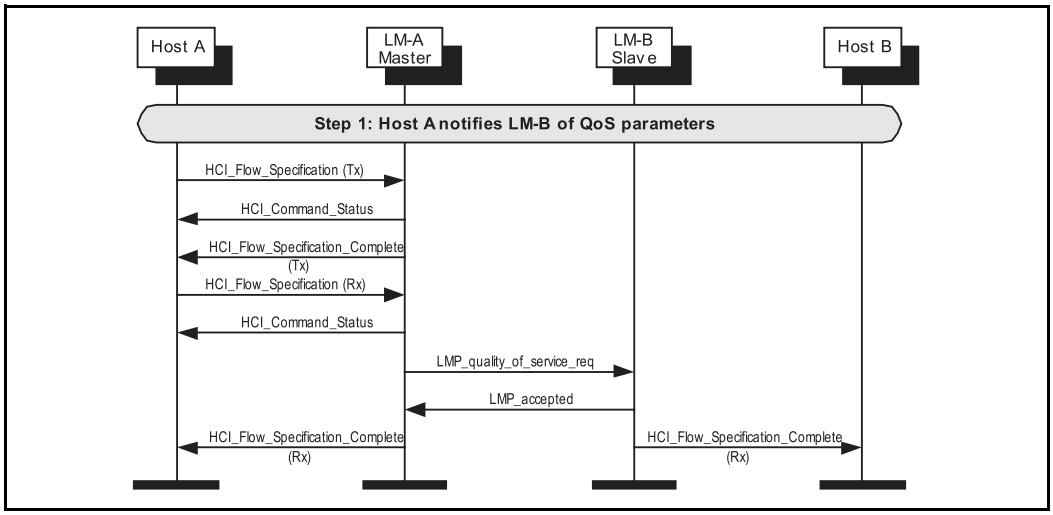


Figure 4.11: QoS flow specification.

4.10 SWITCH ROLE

The HCI_Switch_Role command can be used to explicitly switch the current master / slave role of the local device with the specified device.

Step 1a: The master host (A) requests a role switch with a slave. This will send the switch request, and the slave will respond with the slot offset and accepted. (See Figure 4.12 on page 622)

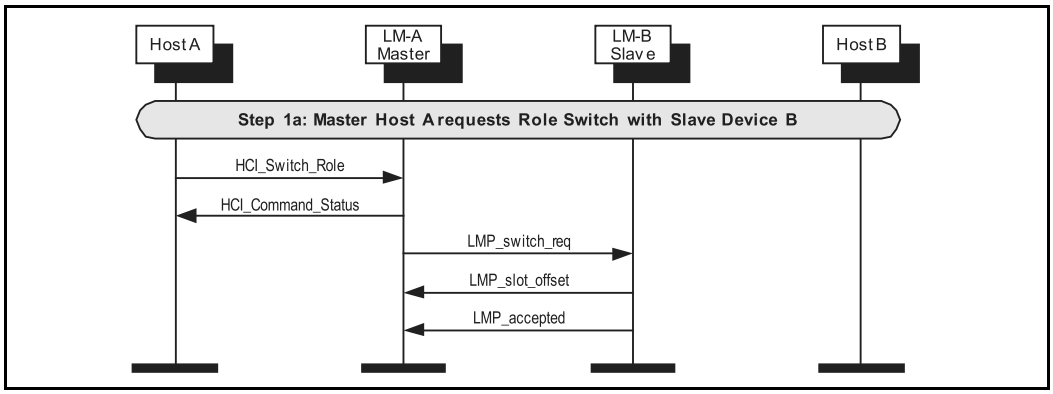


Figure 4.12: Master requests role switch.

Step 1b: The slave host (B) requests a role switch with a master. This will send the slot offset and the switch request, and the master will respond with a LMP_accepted PDU. (See Figure 4.13 on page 623)

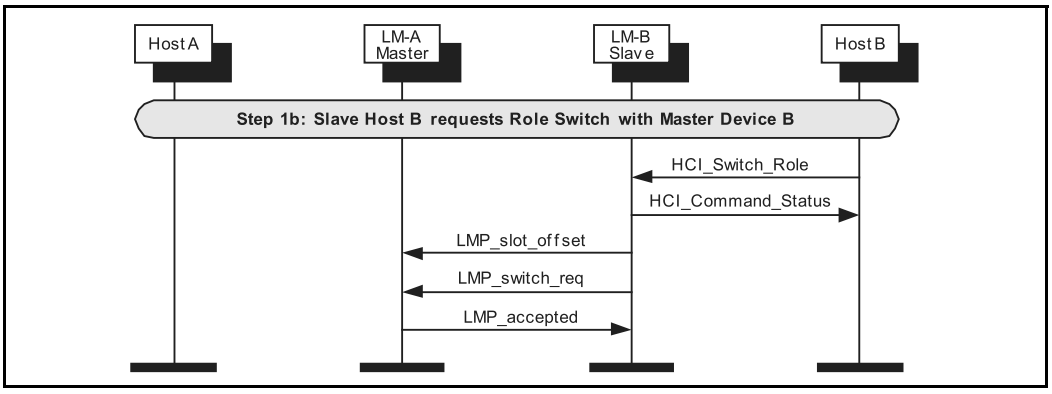


Figure 4.13: Slave requests role switch.

Step 2: The role switch is performed by doing the TDD switch and piconet switch. Finally an HCI_Role_Change event is sent on both sides. (See Figure 4.14 on page 623)

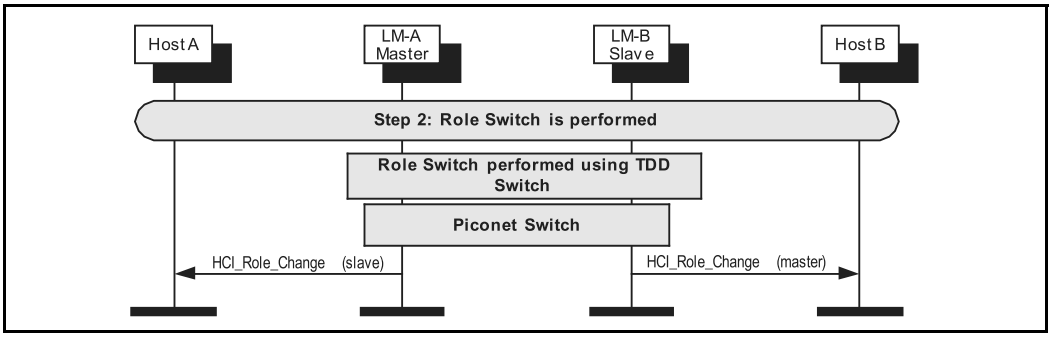


Figure 4.14: Role switch is performed.

Petitioner Kiosoft Exhibit 1011 Part 2
Page 382

5 SYNCHRONOUS CONNECTION ESTABLISHMENT AND DETACHMENT

5.1 SYNCHRONOUS CONNECTION SETUP

Using the HCI_Setup_Synchronous_Connection command, a host can add a synchronous logical channel to the link. A synchronous logical link can be provided by creating a SCO or an eSCO logical transport.

Note: An ACL Connection must be established before a synchonous connection can be created.

Step 1a: Master device requests a synchronous connection with a device. (See Figure 5.1 on page 625)

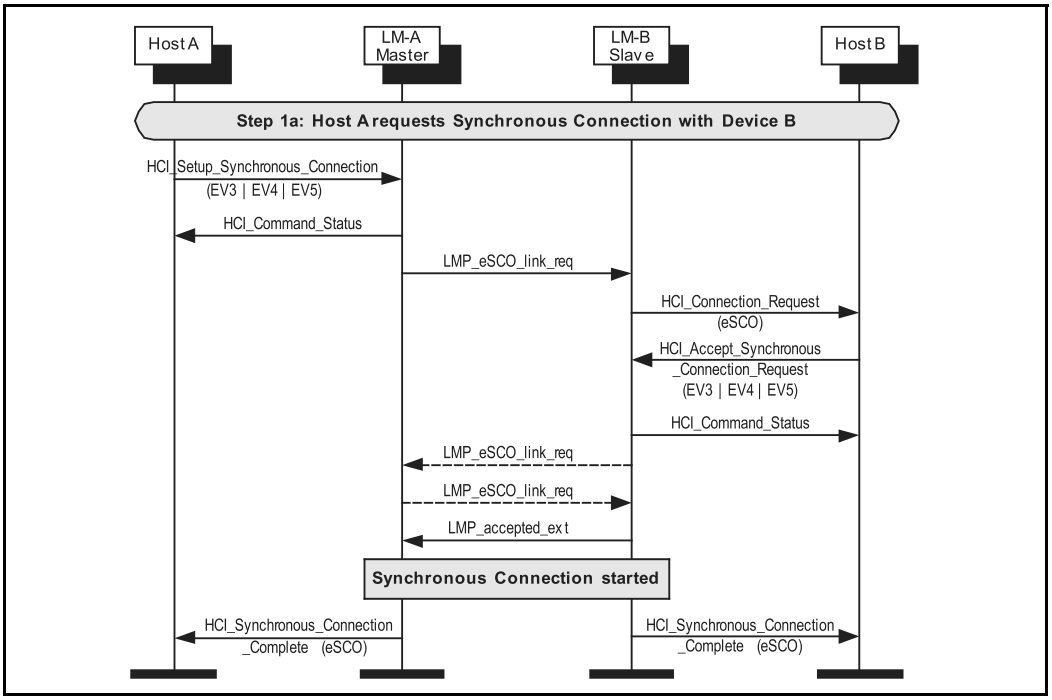


Figure 5.1: Master requests synchronous EV3, EV4 OR EV5 connection.

Step 1b: Slave device requests a synchronous connection with a device. (See Figure 5.2 on page 626)

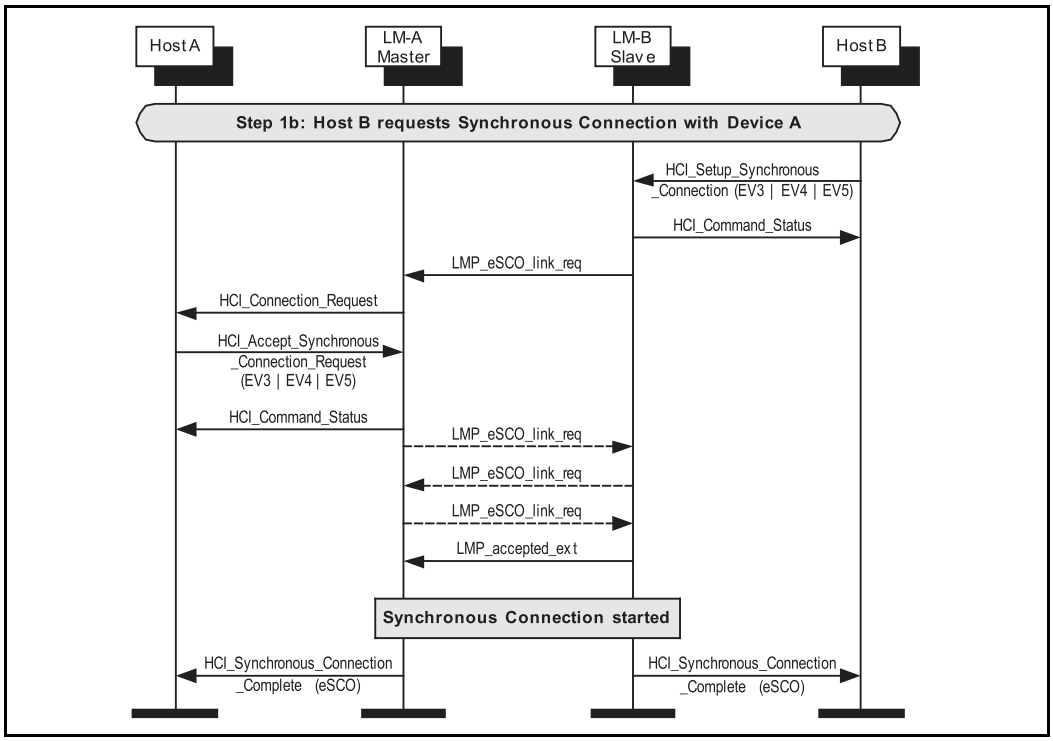


Figure 5.2: Slave requests synchronous EV3, EV4 OR EV5 connection.

Step 1c: Master device requests a SCO connection with a device. (See Figure 5.3 on page 626)

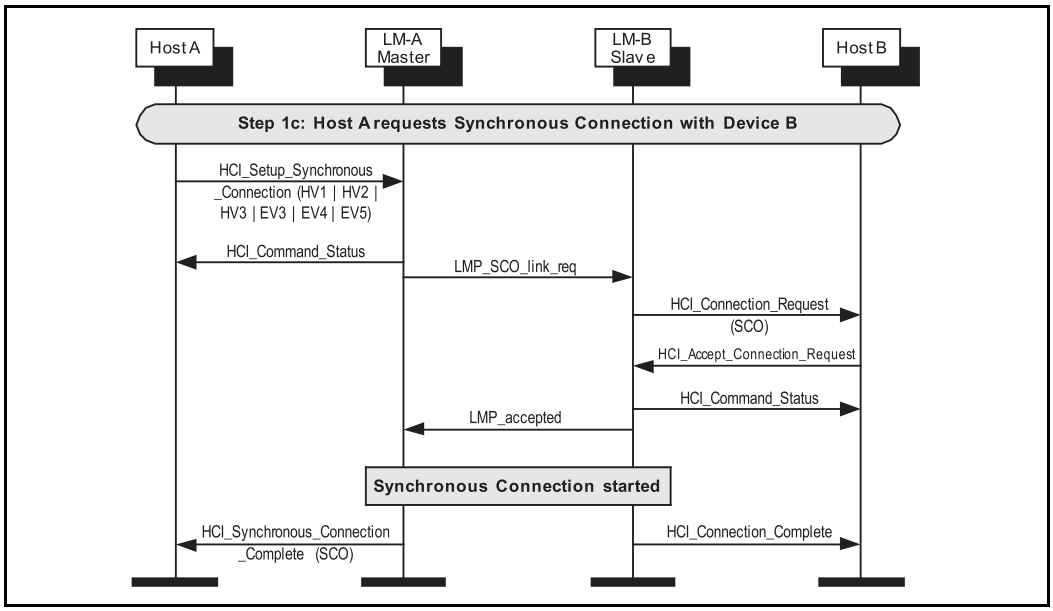


Figure 5.3: Master requests synchronous connection using SCO.

Step 1d: Master device requests a SCO connection with a device.
(See Figure 5.4 on page 627)

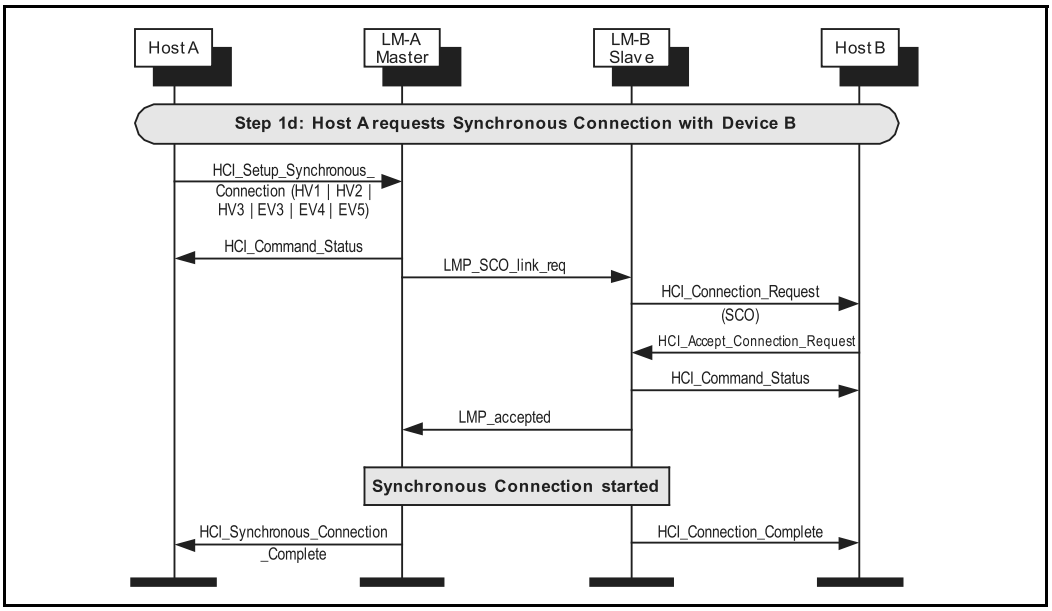


Figure 5.4: Master requests synchronous connection with legacy slave.

Step 1e: Host device requests a SCO connection with a device.
(See Figure 5.5 on page 627)

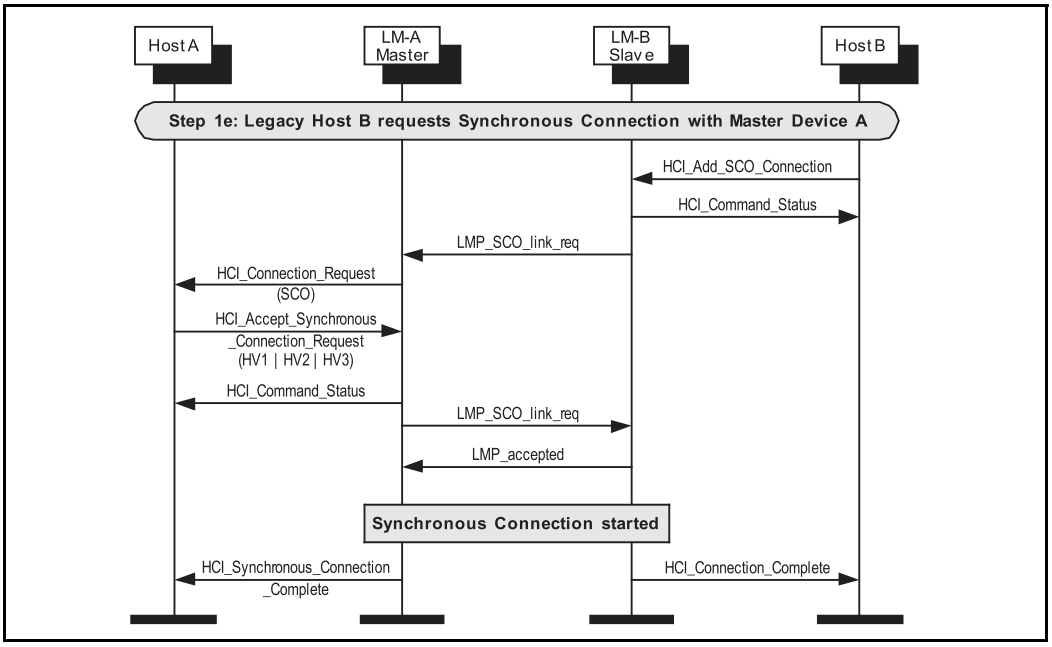


Figure 5.5: Any device that supports only SCO connections requests a synchronous connection with a device.

Step 2a: Master renegotiates eSCO connection (See Figure 5.6 on page 628).

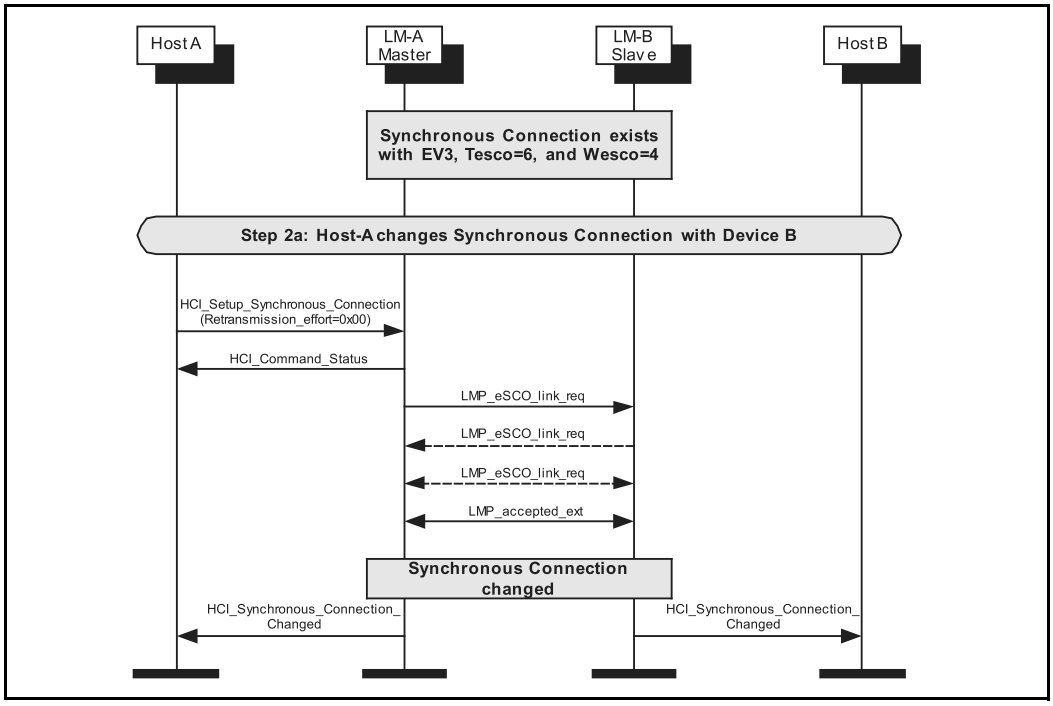


Figure 5.6: Master renegotiates eSCO connection.

Step 2b: Slave renegotiates eSCO connection (See Figure 5.7 on page 628).

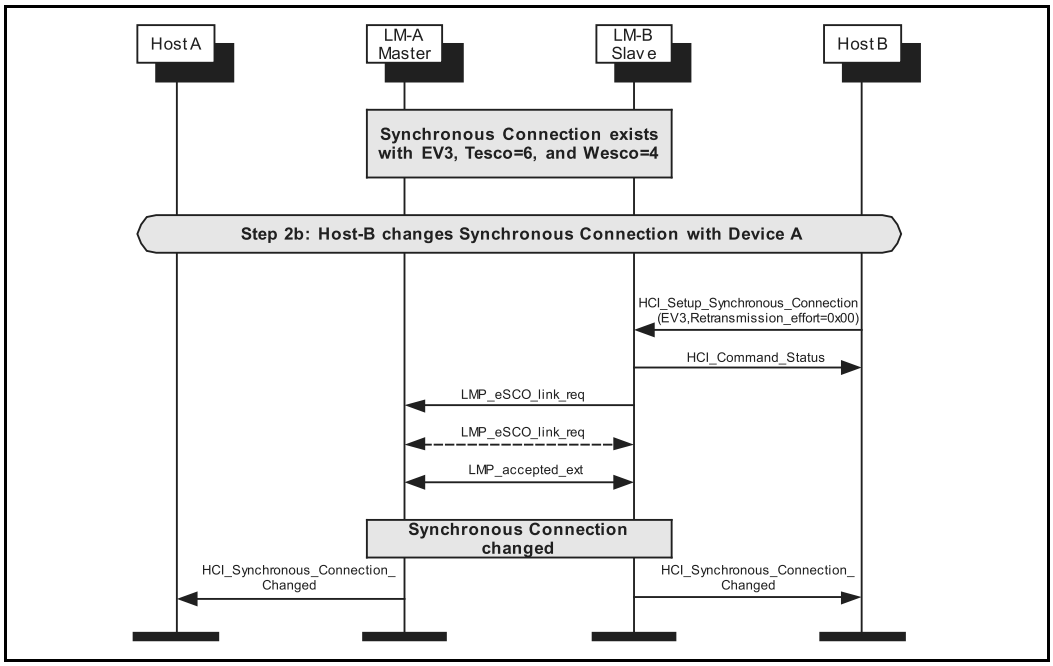


Figure 5.7: Slave renegotiates eSCO connection.

Step 3a: eSCO Disconnection. (See Figure 5.8 on page 629)

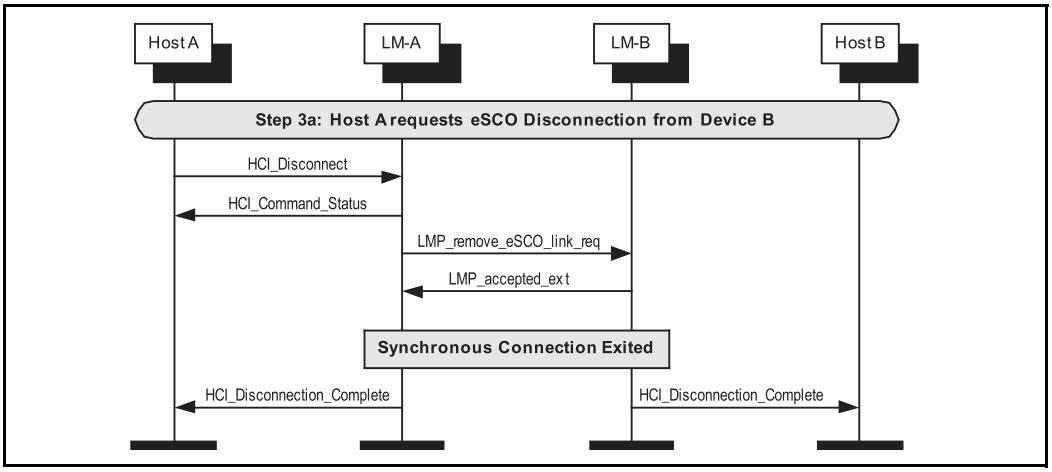


Figure 5.8: Synchronous disconnection of eSCO connection.

Step 3b: SCO Disconnection. (See Figure 5.9 on page 629)

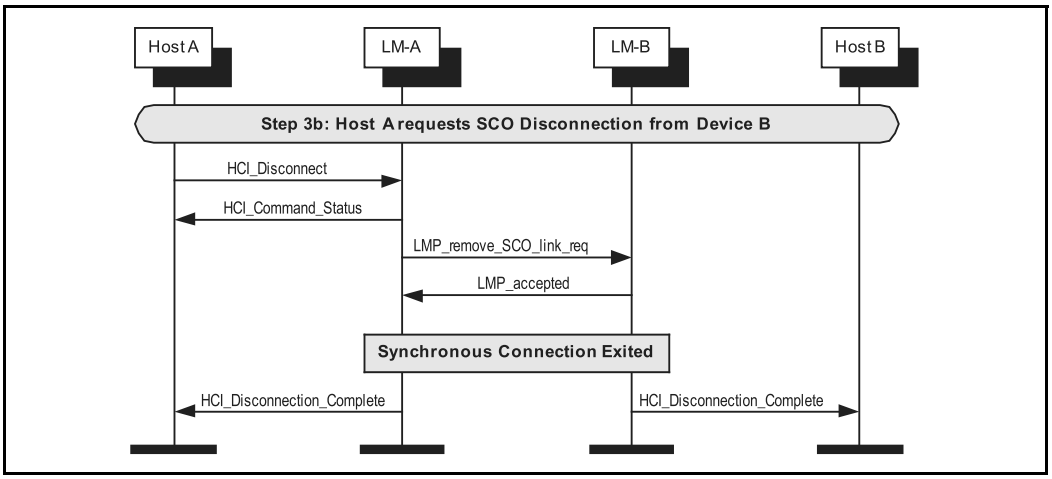


Figure 5.9: Synchronous disconnection of SCO connection.

6 SNIFF, HOLD AND PARK

Entry into sniff mode, hold mode or park state requires an established ACL Connection.

6.1 SNIFF MODE

The HCI_Sniff_Mode command is used to enter sniff mode. The HCI_Exit_Sniff_Mode command is used to exit sniff mode.

Step 1: Host requests to enter sniff mode. Multiple LMP_sniff_req PDUs may be sent as the parameters for sniff mode are negotiated.
(See Figure 6.1 on page 631)

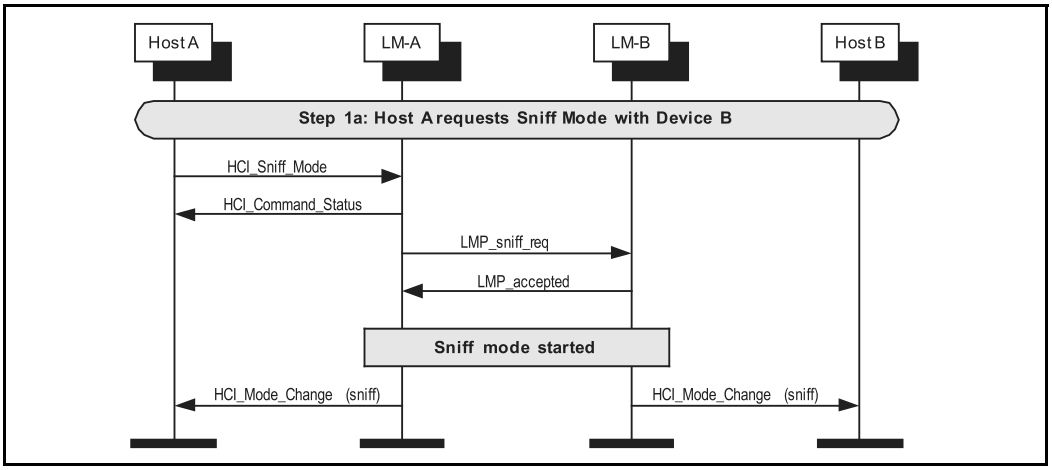


Figure 6.1: Sniff mode request.

Step 2: Host requests to exit sniff mode. (See Figure 6.2 on page 631)

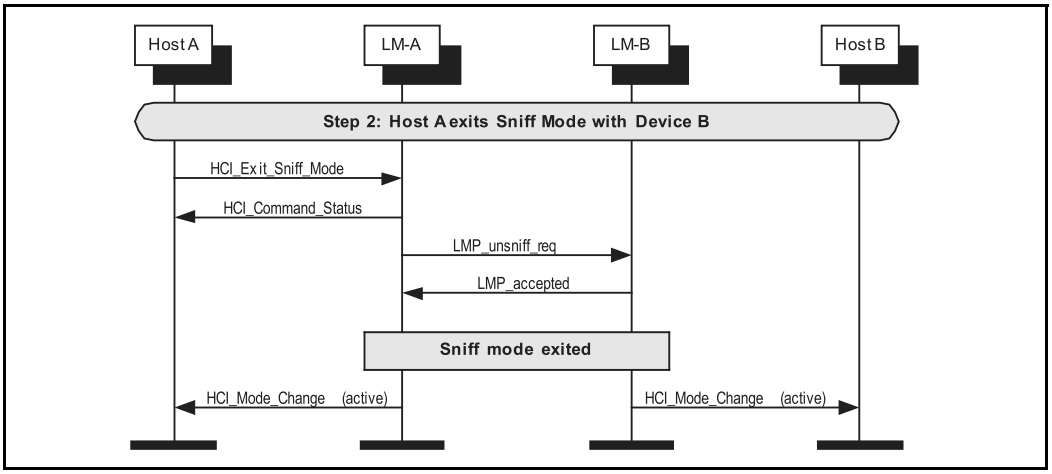


Figure 6.2: Exit sniff mode request.

6.2 HOLD MODE

The HCI_Hold_Mode command can be used to place a device into hold mode. The Controller may do this by either negotiating the hold mode parameters or forcing hold mode. Hold mode will automatically end after the negotiated length of time.

Step 1a: A host requests hold mode. (See Figure 6.3 on page 632)

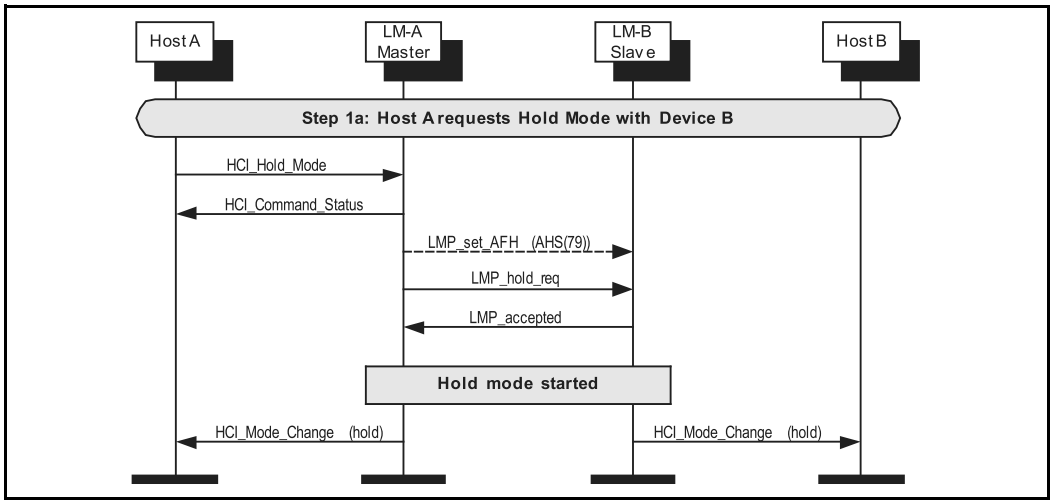


Figure 6.3: Hold request.

Step 1b: A host may force hold mode. (See Figure 6.4 on page 632)

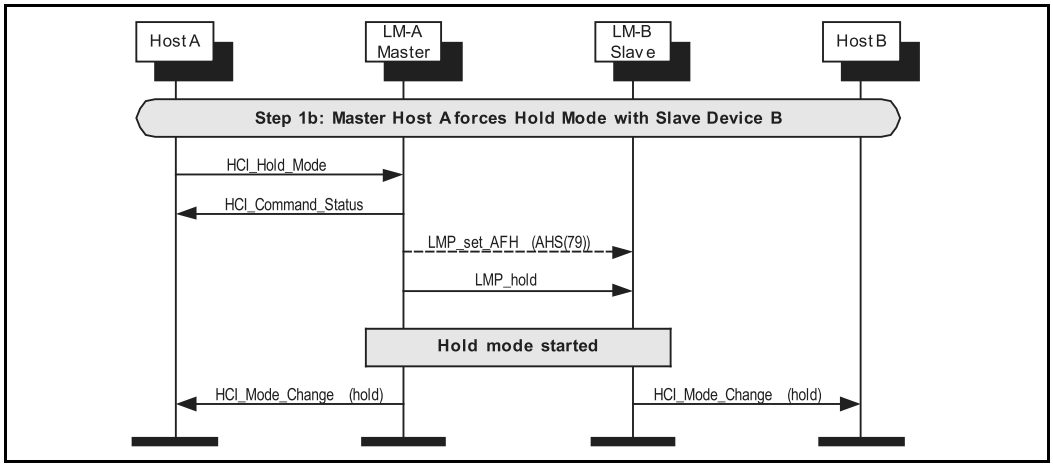


Figure 6.4: Master forces hold mode.

Step 1c: A slave device requests hold mode. (See Figure 6.5 on page 633)

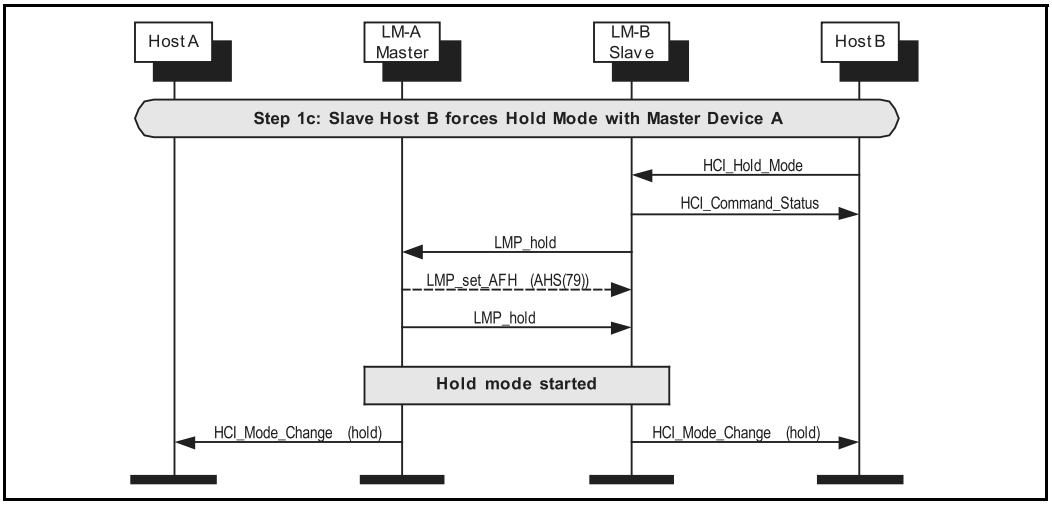


Figure 6.5: Slave forces hold mode.

Step 2: When hold mode completes the hosts are notified using the HCI_Mode_Change event. (See Figure 6.6 on page 633)

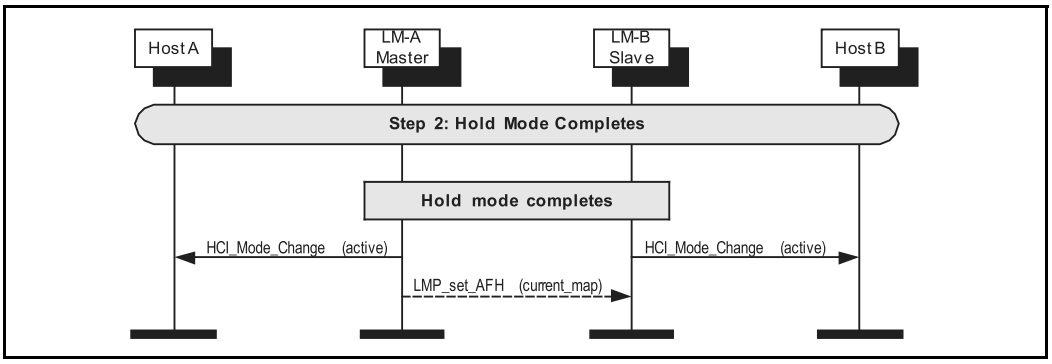


Figure 6.6: Hold mode completes.

6.3 PARK STATE

Park state can be entered by using the HCI_Park_State command.

Step 1a: The master requests to place the slave in the park state. Before sending the LMP_park_req PDU, the master may disable AFH by setting the connection into AHS(79). (See Figure 6.7 on page 634)

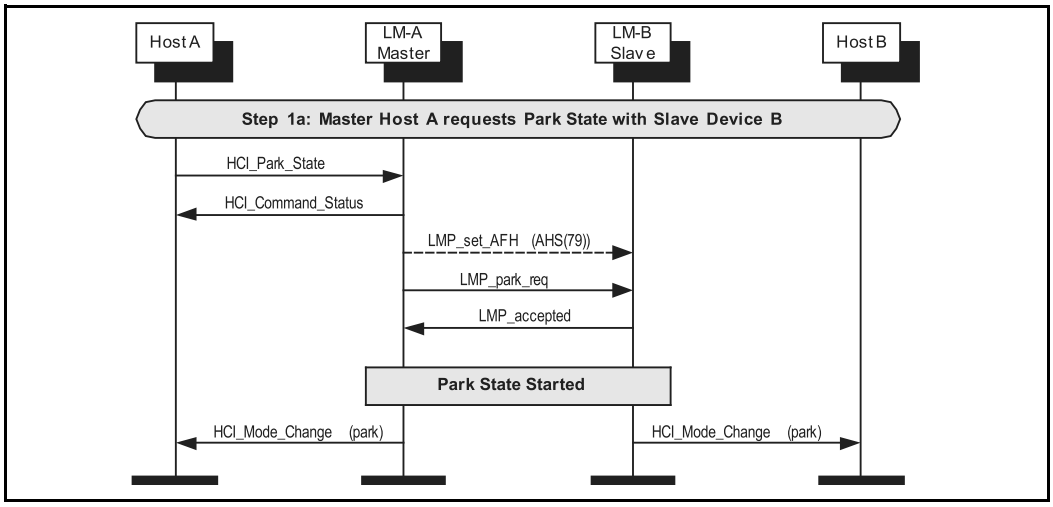


Figure 6.7: Park state request from master.

Step 1b: The slave requests to be placed in the park state. Before sending the LMP_park_req PDU back to the slave, the master may disable AFH by setting the connection into AHS(79). (See Figure 6.8 on page 634)

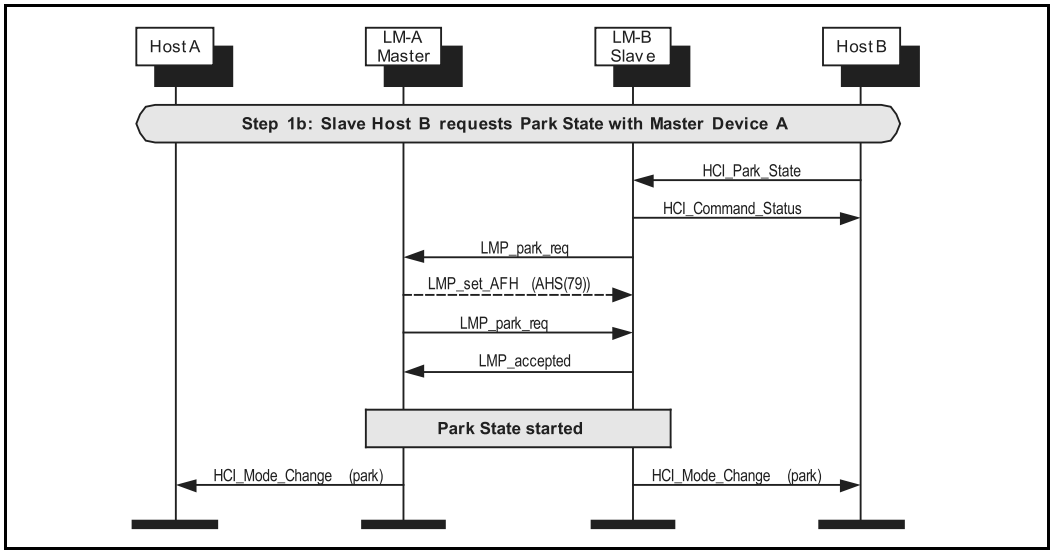


Figure 6.8: Park state request from slave.

Step 2: When in the park state, a slave still needs to unparked for link supervision purposes. The master sends an LMP_unpark_PM_ADDR_req PDU or an LMP_unpark_BD_ADDR_req PDU to the slave during the beacon. Only the PM_ADDR version is illustrated in the figure. (See Figure 6.9 on page 635)

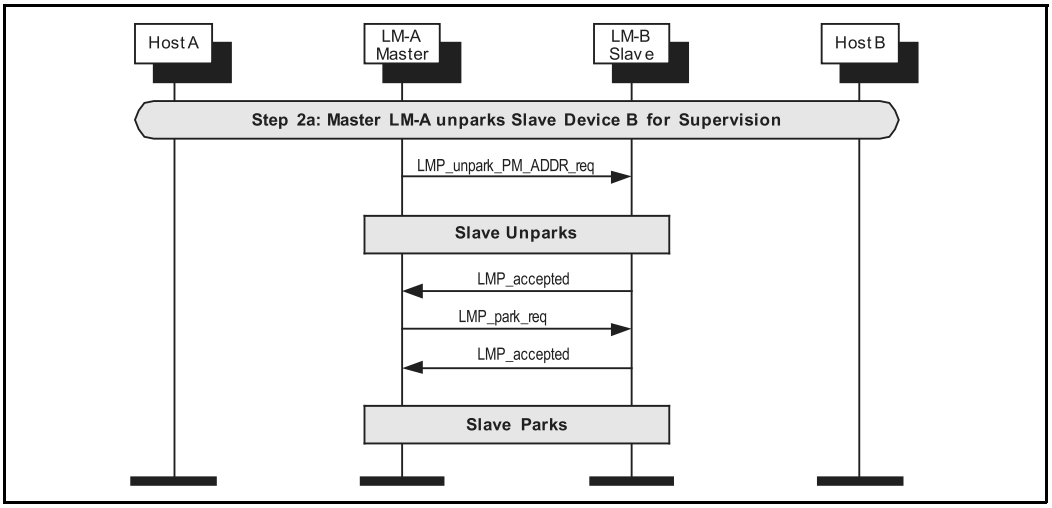


Figure 6.9: Master unparks slave for supervision.

Step 3a: A master may unpark a slave to exit park state. The master should re-enable AFH by setting the current AFH channel map to the unparked slave. (See Figure 6.10 on page 635)

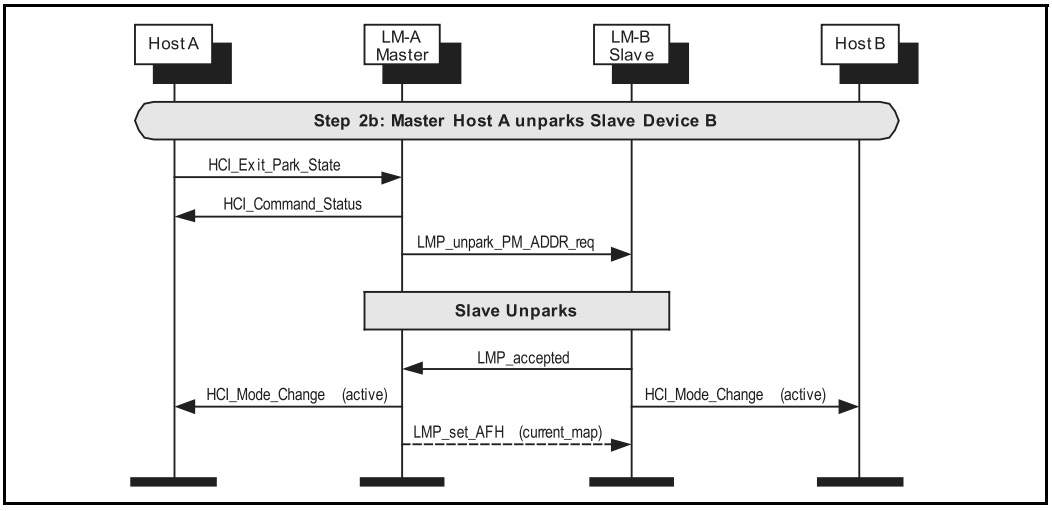


Figure 6.10: Master exits park state with slave.

Step 3b: A slave may unpark itself by sending a message in an access window. It will then receive instructions from the master to unpark. The master should re-enable AFH by setting the current AFH channel map to the unparked slave. (See Figure 6.11 on page 636)

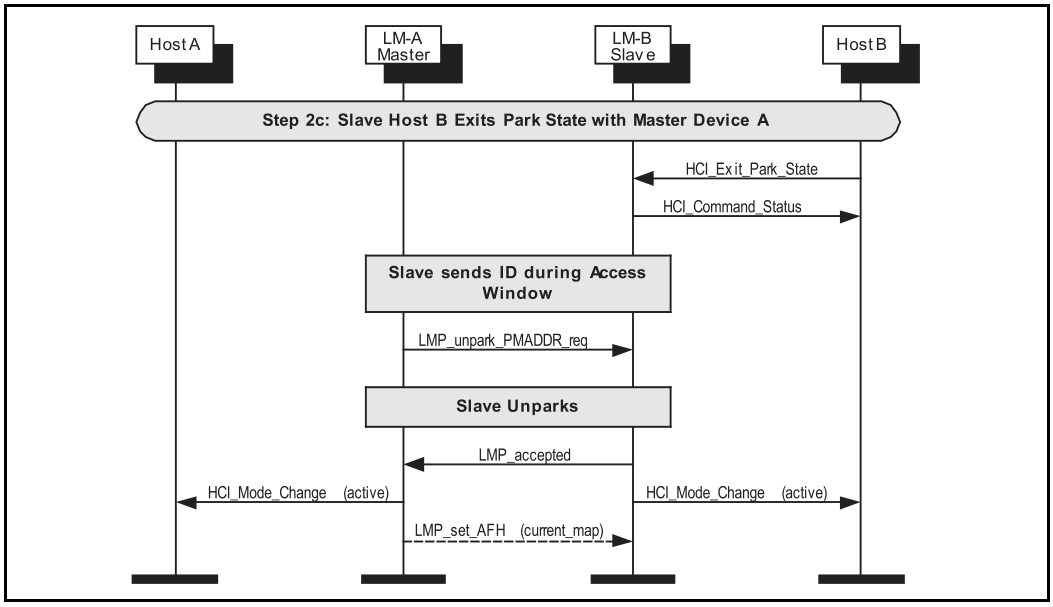


Figure 6.11: Slave exits park state with master.

7 BUFFER MANAGEMENT, FLOW CONTROL

Buffer management is very important for resource limited devices. This can be achieved on the Host Controller Interface using the HCI_Read_Buffer_Size command, and the HCI_Number_Of_Completed_Packets event, and the HCI_Set_Host_Controller_To_Host_Flow_Control, HCI_Host_Buffer_Size and HCI_Host_Number_Of_Completed_Packets commands.

Step 1: During initialization, the host reads the buffer sizes available in the Controller. When an HCI Data Packet has been transferred to the remote device, and a Baseband acknowledgement has been received for this data, then an HCI_Number_Of_Completed_Packets event will be generated. (See Figure 7.1 on page 637)

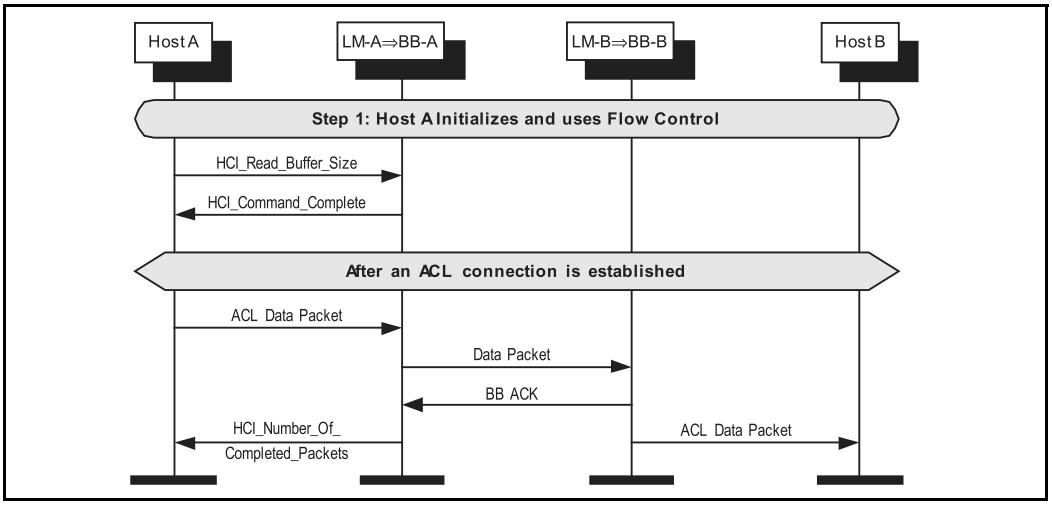


Figure 7.1: Host to Controller flow control.

Step 2: During initialization, the host notifies the Controller that host flow control shall be used, and then the host buffer sizes available. When a data packet has been received from a remote device, an HCI Data Packet is sent to the host from the Controller, and the host shall acknowledge its receipt by sending HCI_Host_Number_Of_Completed_Packets. (See Figure 7.2 on page 638)

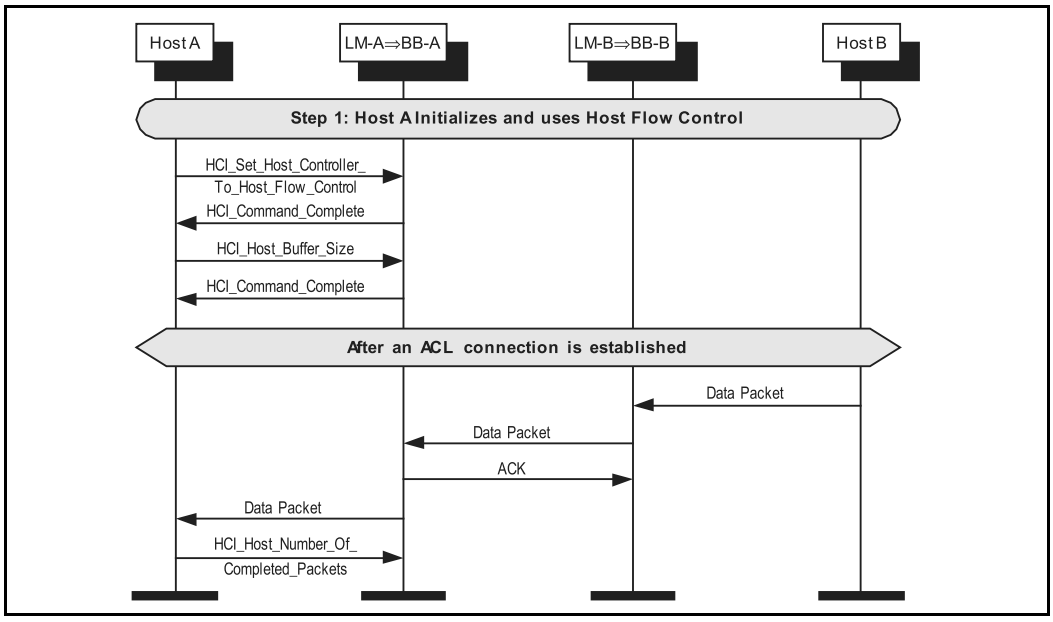


Figure 7.2: Controller to Host flow control.

8 LOOPBACK MODE

The loopback modes are used for testing of a device only.

8.1 LOCAL LOOPBACK MODE

The local loopback mode is used to loopback received HCI Commands, and HCI ACL and HCI Synchronous packets sent from the Host to the Controller.

Step 1: The host enters local loopback mode. Four connection complete events are generated and then a command complete event.
(See Figure 8.1 on page 639)

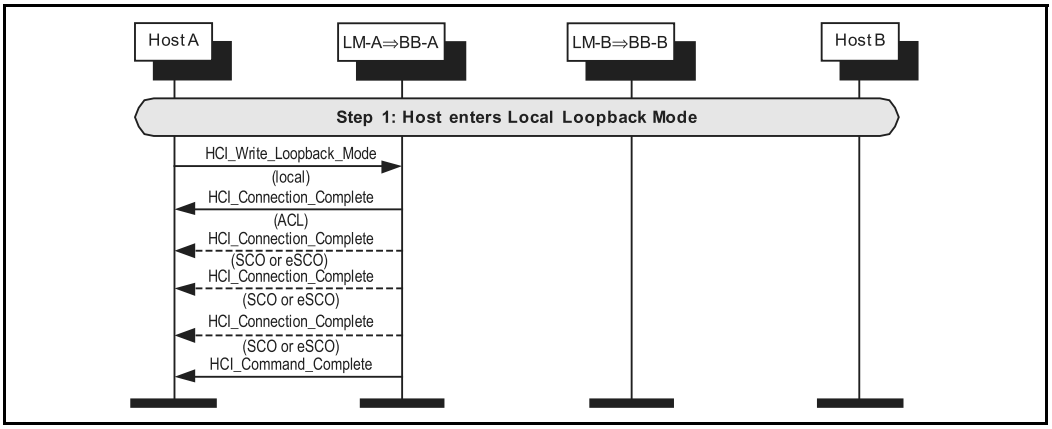


Figure 8.1: Entering local loopback mode.

Step 2a: The host sending HCI Data Packet will receive the exact same data back in HCI Data Packets from the Controller. (See Figure 8.2 on page 639)

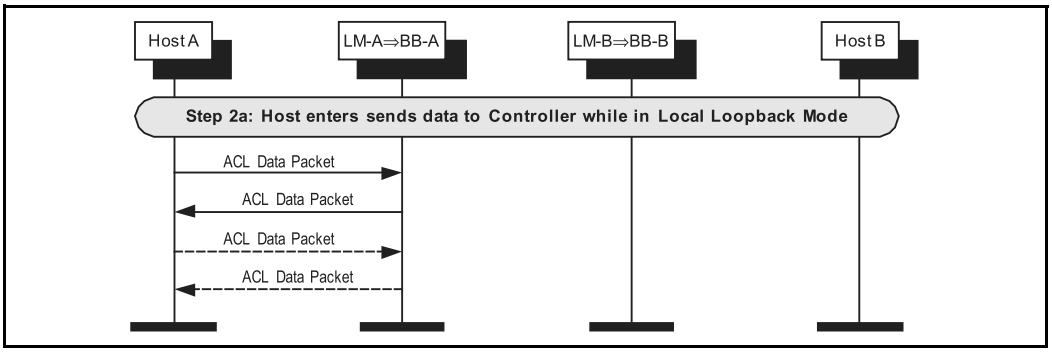


Figure 8.2: Looping back data in local loopback mode.

Step 2b: The host sending most HCI Command Packets to the Controller will receive an HCI_Loopback_Command event with the contents of the HCI Command Packet in the payload. (See Figure 8.3 on page 640)

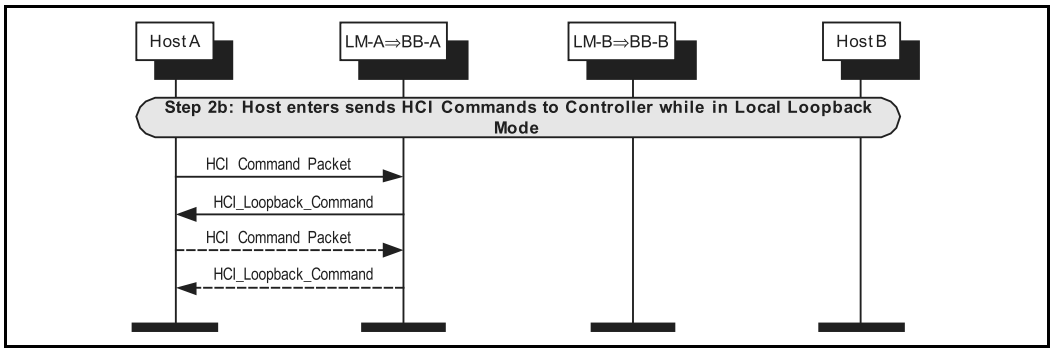


Figure 8.3: Looping back commands in local loopback mode.

Step 3: The host exits local loopback mode. Multiple disconnection complete events are generated before the command complete event.
(See Figure 8.4 on page 640)

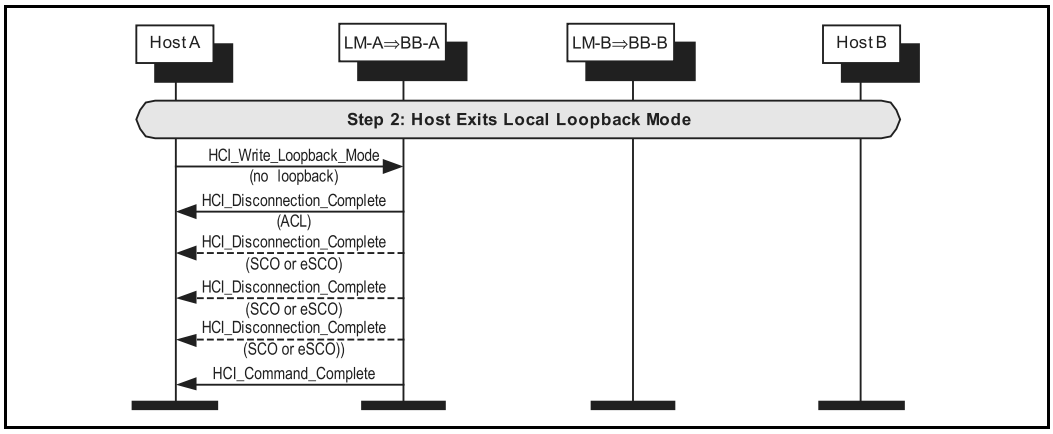


Figure 8.4: Exiting local loopback mode.

8.2 REMOTE LOOPBACK MODE

The remote loopback mode is used to lookback data to a remote device over the air.

Step 1: The remote host first sets up an connection to the local device. The local device then enables remote loopback. (See Figure 8.5 on page 641)

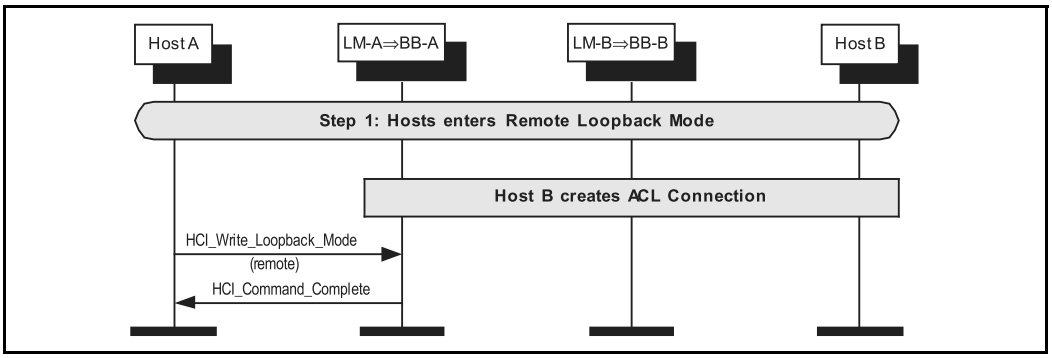


Figure 8.5: Entering remote loopback mode.

Step 2: Any data received from the remote host will be loopbacked in the Controller of the local device. (See Figure 8.6 on page 641)

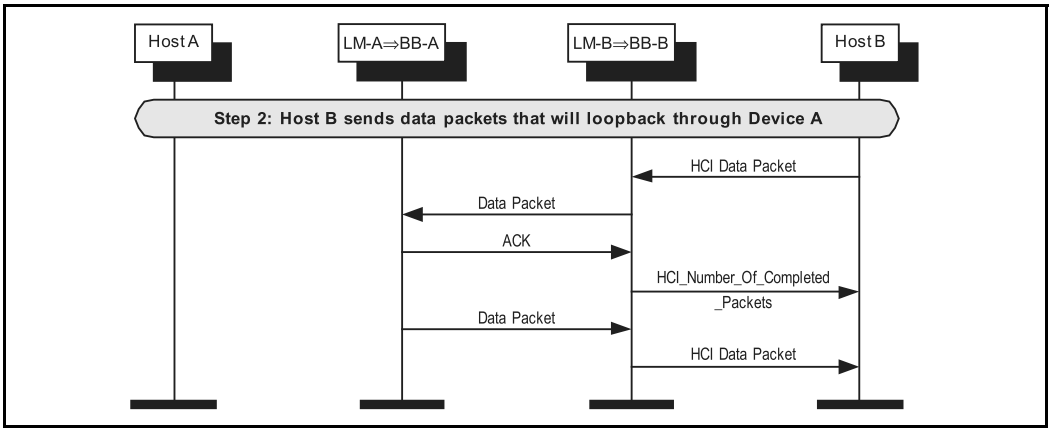


Figure 8.6: Looping back data in Remote Loopback Mode.

Step 3: The local host exits remote loopback mode. Any connections can then be disconnected by the remote device. (See Figure 8.7 on page 642)

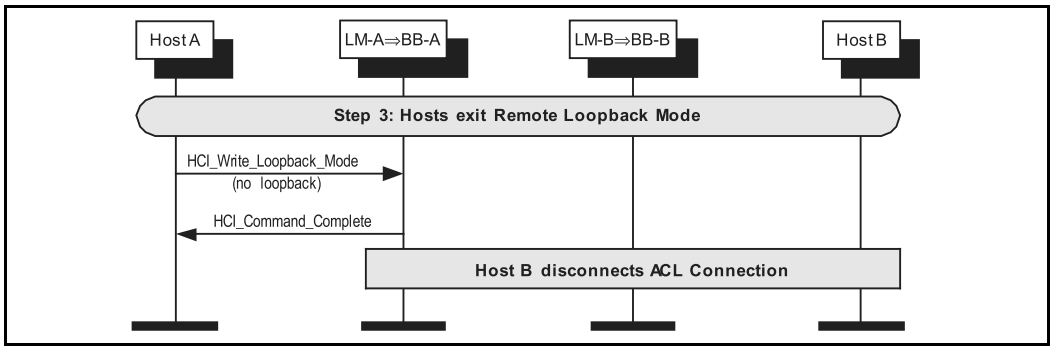


Figure 8.7: Exiting remote loopback mode.