Appendix IX

MESSAGE SEQUENCE CHARTS

Between Host and Host Controller/Link Manager

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

CONTENTS

1 INTRODUCTION

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

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

Notation used in the message sequence charts:

Box:

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

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

Hexagon:

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

Arrow:

- Represents a message, signal or transaction

Comment:

- “/* … */” indicates editor comments

2 SERVICES WITHOUT CONNECTION REQUEST

2.1 REMOTE NAME REQUEST

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

Sending an HCI_Remote_Name_Request (BD_ADDR, Page_Scan_ Repetition_Mode, Page_Scan_Mode, Clock_Offset), the Host expects that its local BT Device will automatically try to connect to the remote BT Device (with the specified BD_ADDR). Then the local BT Device should try to get the name, to disconnect, and finally to return the name of the remote BT Device back to the Host (see Figure 2.1 Remote Name Request: sub-scenario 1).

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

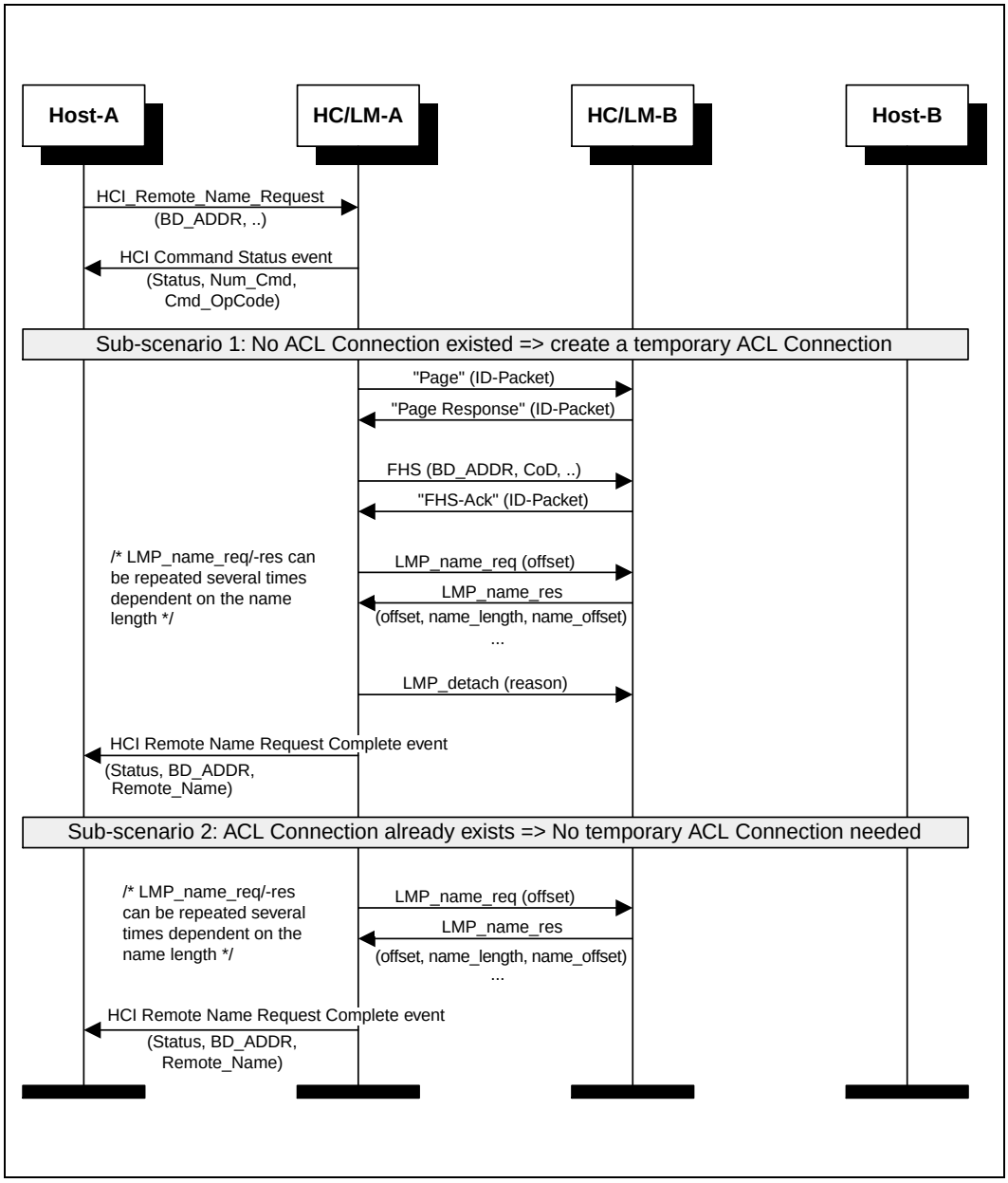

Figure 2.1: Remote Name Request

2.2 ONE-TIME INQUIRY

Inquiry is used to detect and collect nearby BT Devices. When receiving the command HCI_Inquiry (LAP, Inquiry_Length, Num_Responses), HC will start the baseband inquiry procedure with an Inquiry Access Code (derived from the specified LAP) and Inquiry Length. When Inquiry Responses are received, HC will filter out and then return the information related to the found BT Devices using one or several Inquiry Result events (Num_Responses, BD_ADDR[I], Page_Scan_Repetition_Mode[i], Page_Scan_Period_Mode[i], Page_Scan_Mode[i], Class_Of_Device[i], Clock_Offset[i]) to the Host.

The filtering of found BT Devices is specified in HCI_Set_Event_Filter (Filter_Type, Filter_Condition_Type, Condition) with the Filter_Type = Inquiry Result. When the Inquiry procedure is completed, Inquiry Complete event (Status, Num_Responses) must be returned to the Host. Otherwise, the command HCI_Inquiry_Cancel( ) will be used to directly stop the inquiry procedure.

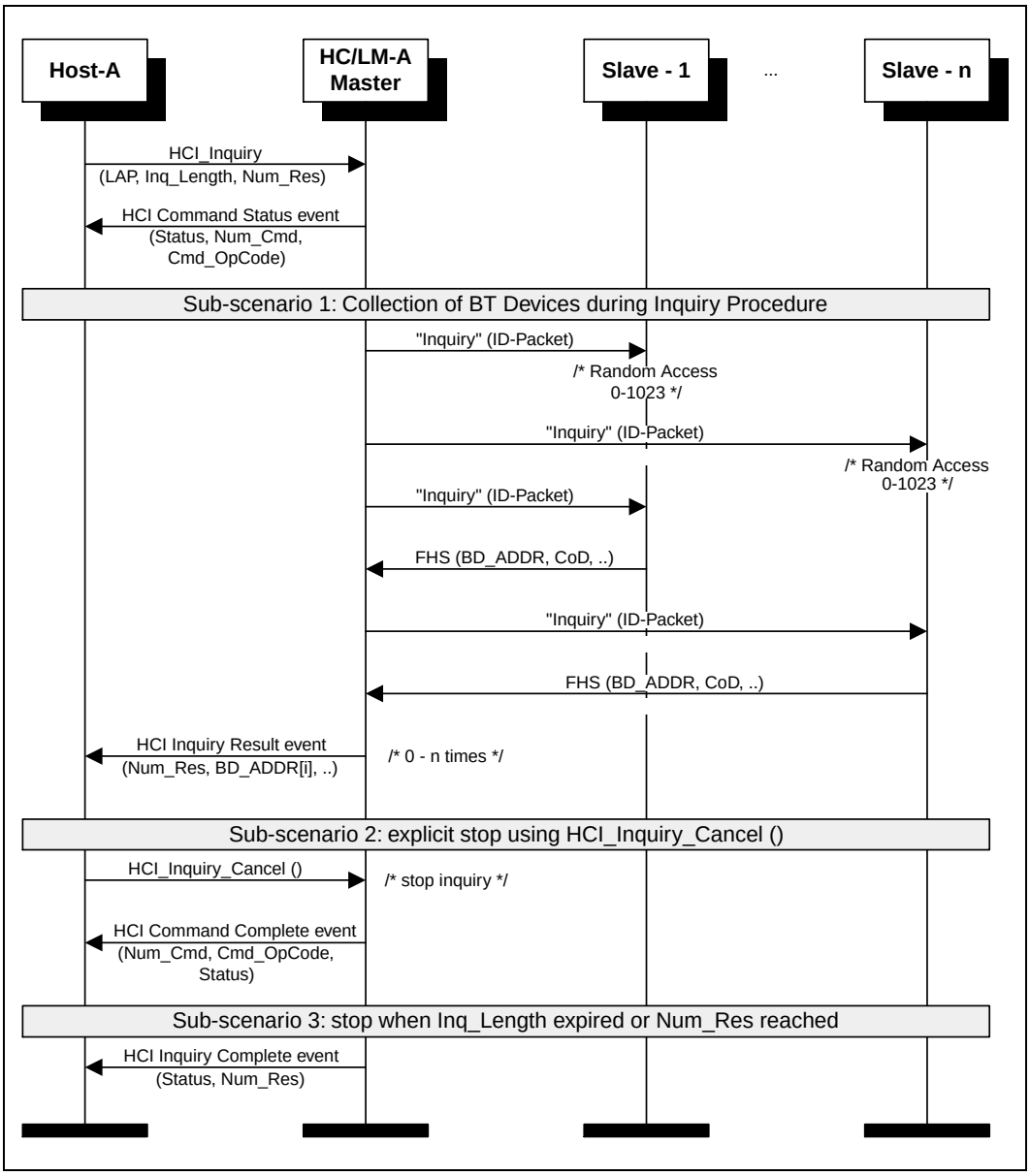

Figure 2.2: One-Time Inquiry

2.3 PERIODIC INQUIRY

Periodic inquiry is needed when the inquiry procedure is to be repeated periodically. Receipt of the command HCI_Periodic_Inquiry_Mode (Max_Period_Length, Min_Period_Length, LAP, Inquiry_Length, Num_Responses) HC will start the periodic Inquiry Mode with the specified

parameters Max_Period_Length, Min_Period_Length, Inquiry_Access_code (derived from LAP) and Inquiry_Length. As in the one-time Inquiry procedure, only BT Devices that are specified in the HCI_Set_Event_Filter (Filter_Type, Filter_Condition_Type, Condition) with the Filter_Type = Inquiry Result will not be filtered out. Therefore, in the inquiry cycle, one or several Inquiry Result events (Num_Responses, BD_ADDR[i], Page_Scan_Repetition_ Mode[i], Page_Scan_Period_Mode[i], Page_Scan_Mode[i], Class_Of_ Device[i], Clock_Offset[i]) and Inquiry Complete event (Status, Num_Responses) will be returned to the Host with one, or a list of, found BT Devices. The periodic Inquiry can be stopped using HCI_Exit_Periodic_ Inquiry_Mode( ).

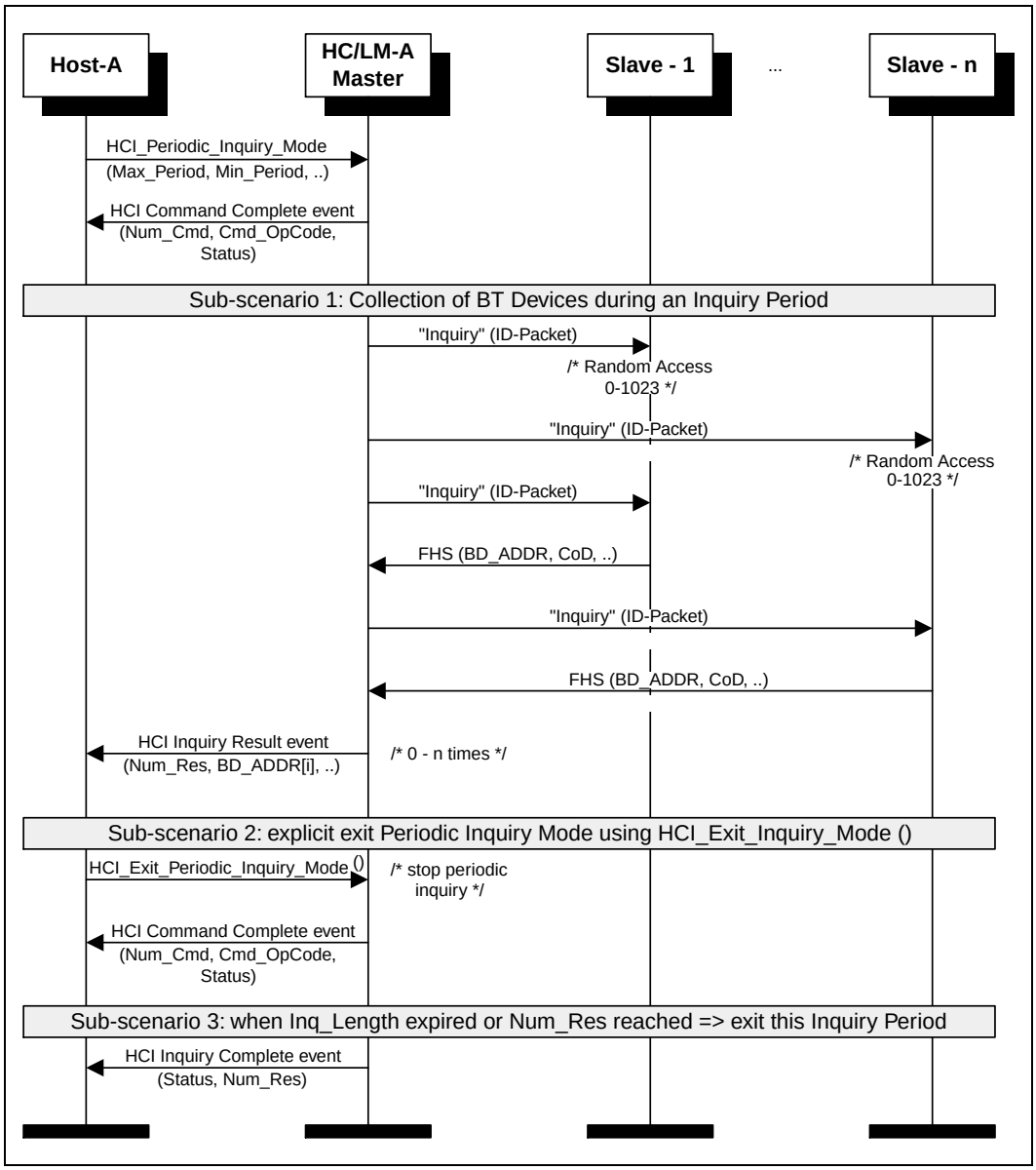

Figure 2.3: Periodic Inquiry

3 ACL CONNECTION ESTABLISHMENT AND DETACHMENT

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

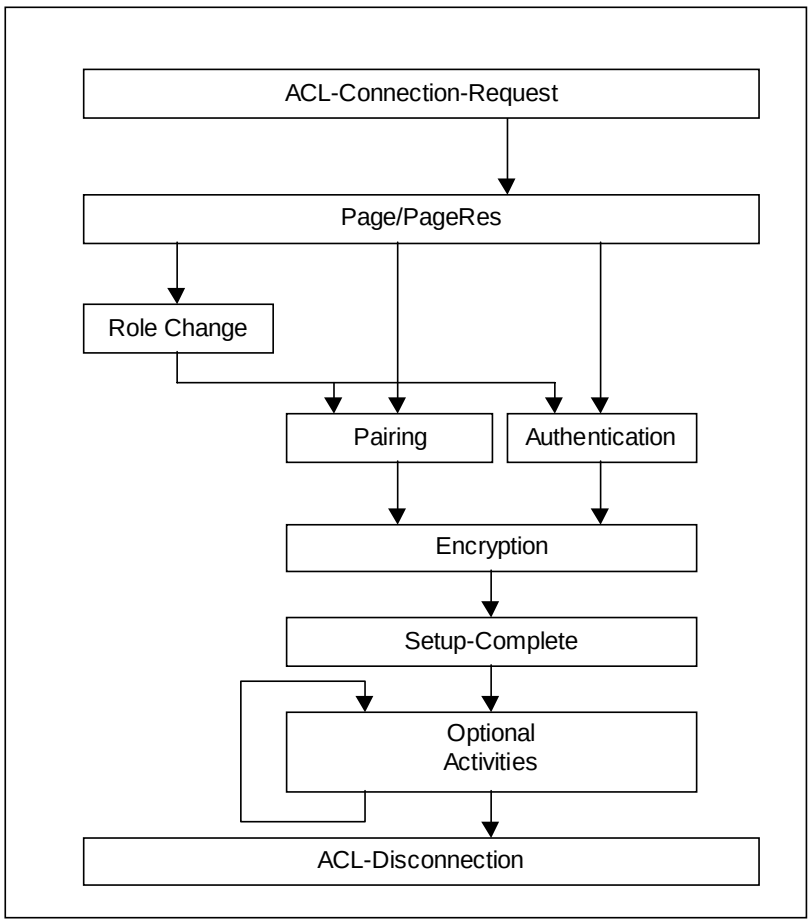

Figure 3.1: Overview of ACL Connection establishment and detachment

3.1 ACL CONNECTION REQUEST PHASE

The ACL Connection Request phase is identified between the HCI_Create_Connection (BD_ADDR, Packet_Type, Page_Scan_Repetition_Mode, Page_Scan_Mode, Clock_Offset, Allow_Role_Switch) from the master side and the response from the slave side with rejection or acceptation on the LM level. Three alternative sub-scenarios are shown in Figure 3.2, “ACL Connection Request phase,” on page 1044.

Sub-scenario 1: Slave rejects ACL Connection Request

If the ACL Connection request is rejected by slave, a Connection Complete event (Status, Connection_Handle, BD_ADDR, Link_Type, Encryption_Mode) will be then returned to Host, whereby the Status will be copied from the Reason parameter of the command HCI_Reject_Connection_Request (Reason, BD_ADDR). The parameters Connection_Handle and Encryption_Mode will be meaningless.

Sub-scenario 2: Slave accepts ACL Connection Request

When the slave responds with LMP_accepted ( ) correspondent to LMP_host_connection_req ( ), the ACL Connection Request is accepted. The master will continue with the ACL Connection Setup, where pairing, authentication or encryption will be executed.

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

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

Note: on the slave side, an incoming connection request can be automatically accepted by using HCI_Set_Event_Filter (Filter_Type, Filter_Condition_Type, Condition) with the Filter_Type = 0x02 /*Connection_Setup*/.

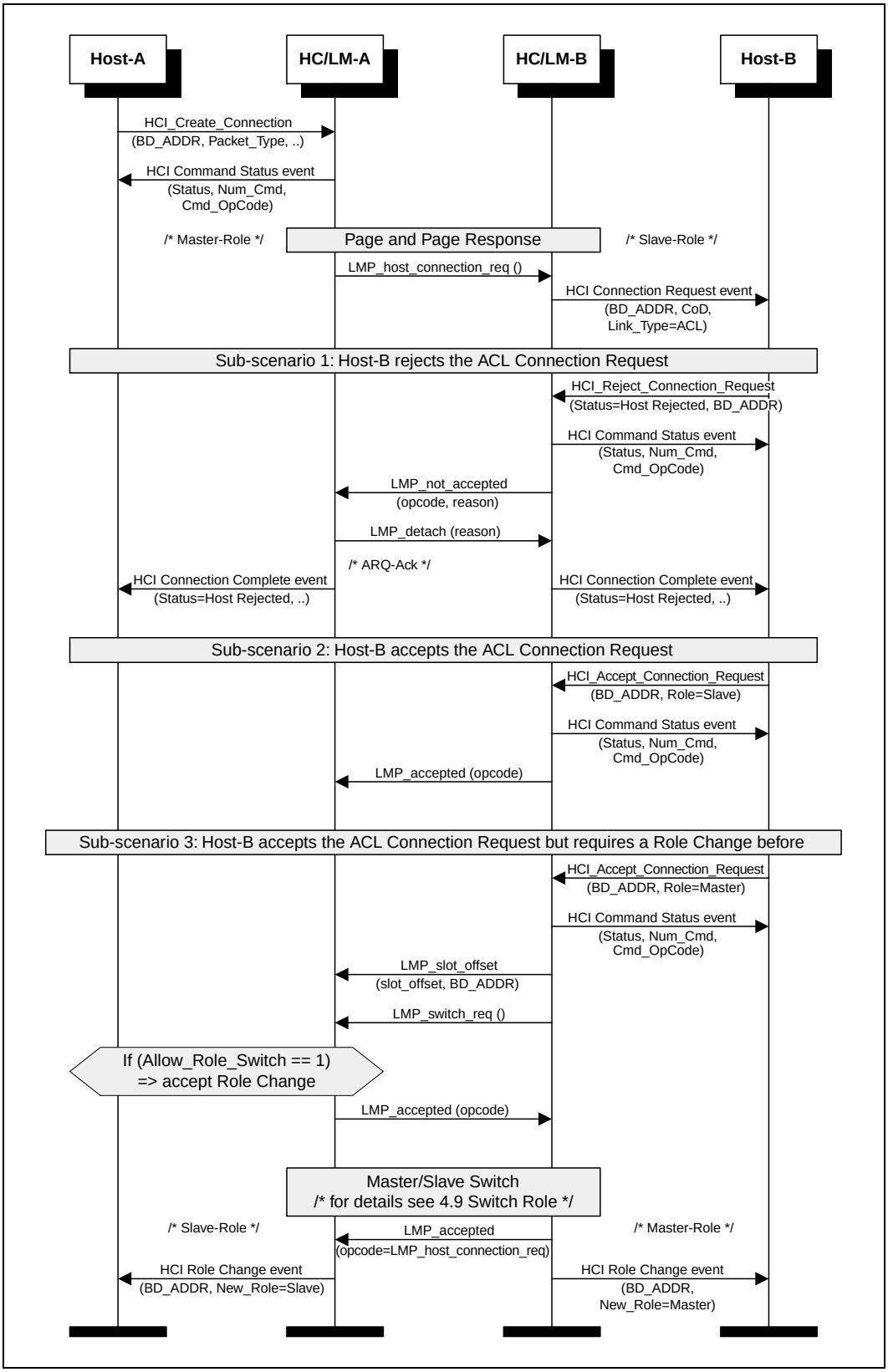

Figure 3.2: ACL Connection Request phase

3.2 ACL CONNECTION SETUP PHASE

If the ACL Connection Request phase was successful, the ACL Connection Setup phase will start, with the goal of executing security procedures like pairing, authentication and encryption. The ACL Connection Setup phase is successfully finished when LMP_setup_complete ( ) is exchanged and the Connection Complete event (Status=0x00, Connection_Handle, BD_ADDR, Link_Type, Encryption_Mode) is sent to the Host.

3.2.1 Pairing

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

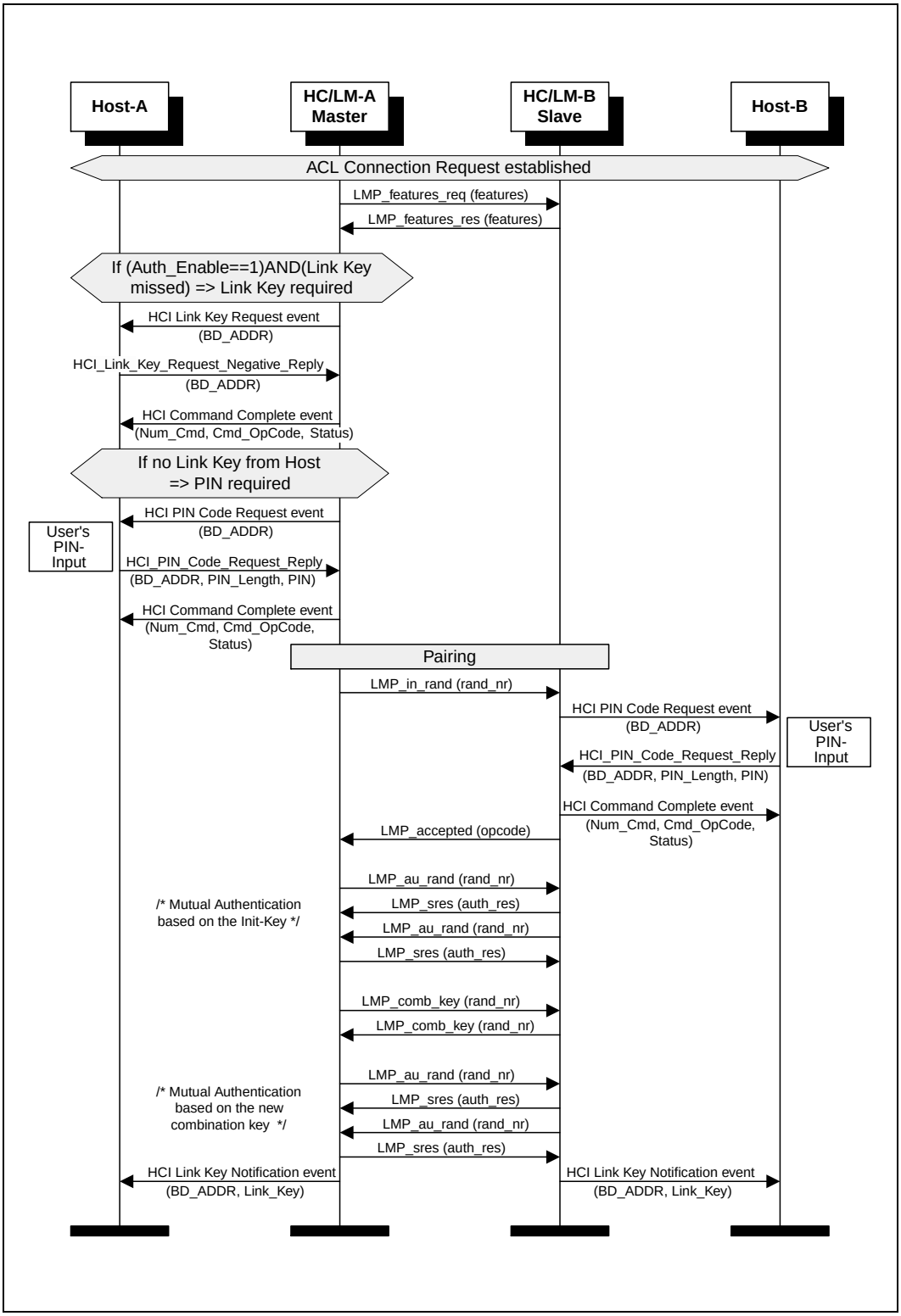

Figure 3.3: ACL Connection setup with pairing

3.2.2 Authentication

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

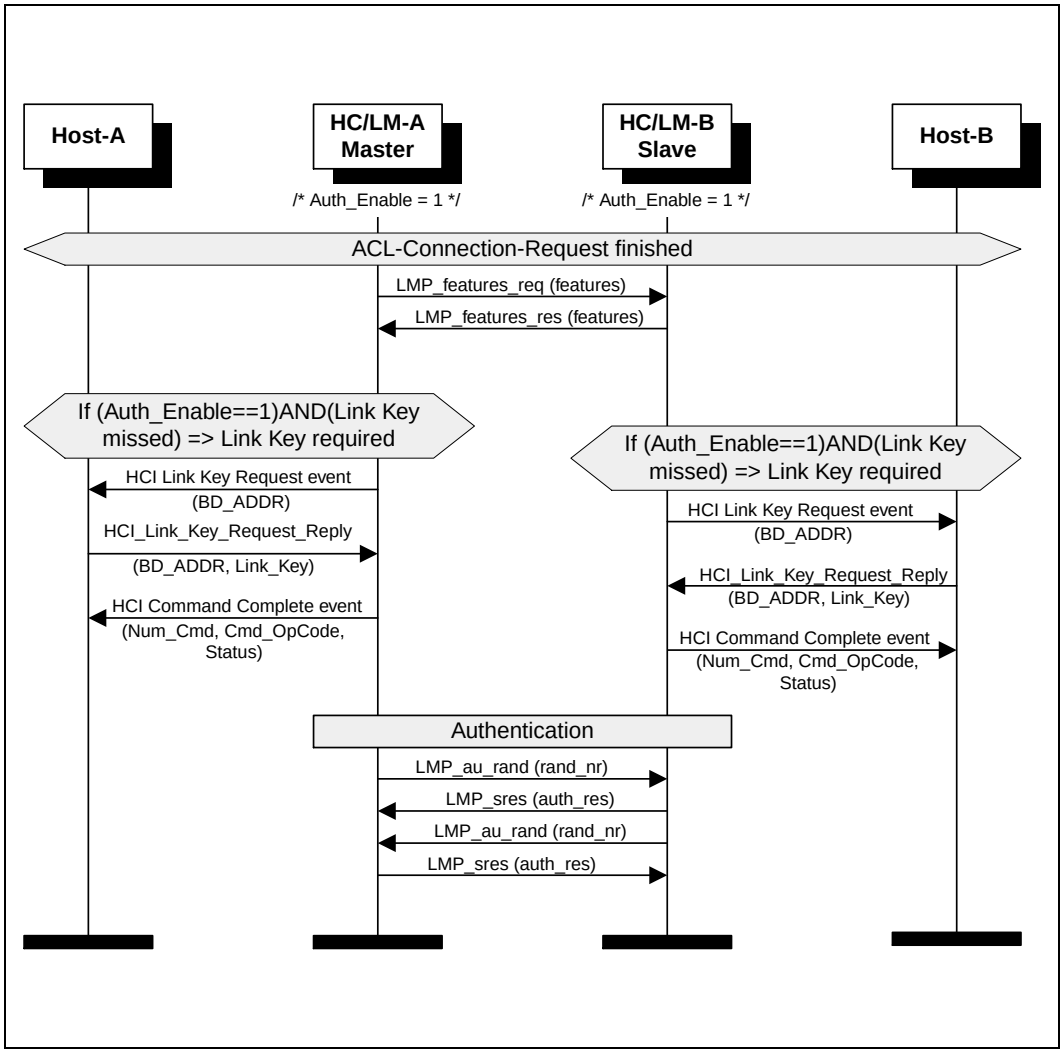

Figure 3.4: ACL Connection setup with authentication

3.3 ENCRYPTION AND CONNECTION SETUP COMPLETE

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

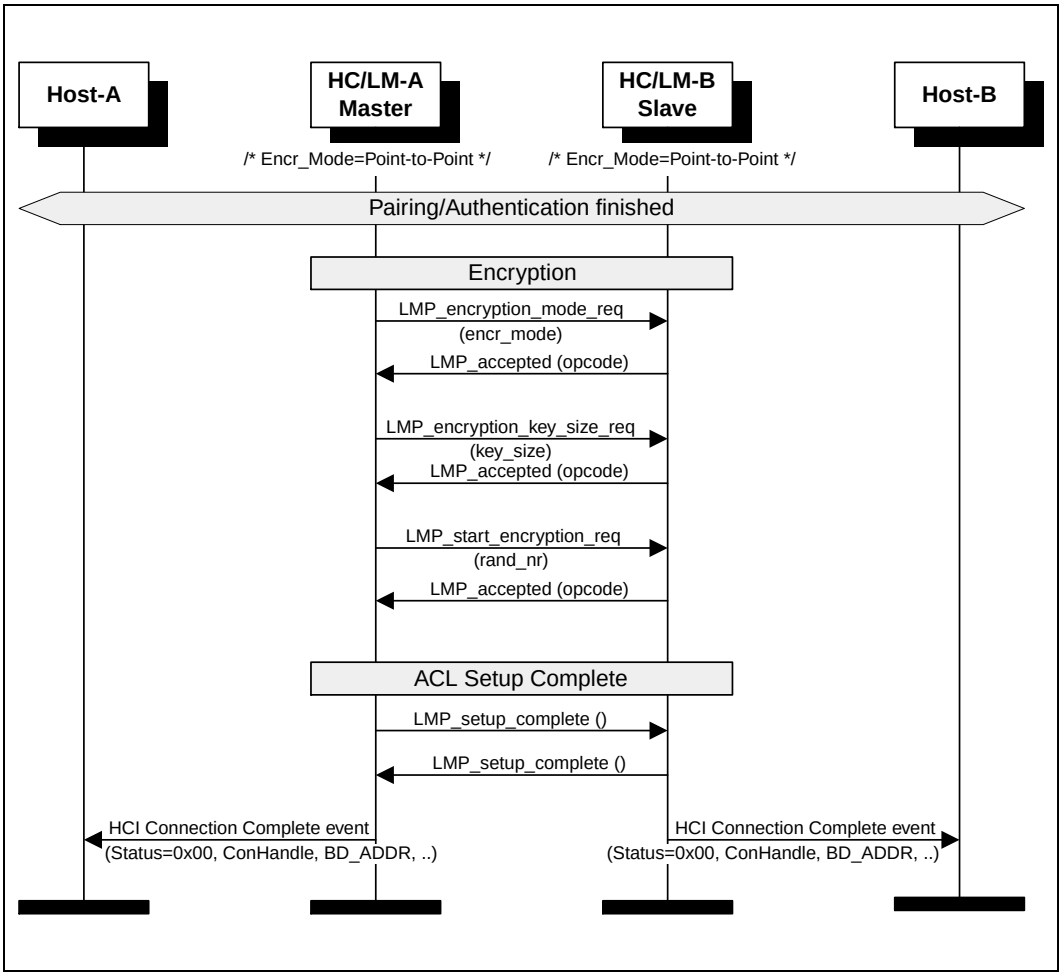

Figure 3.5: Encryption and Setup complete

3.4 ACL DISCONNECTION

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

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

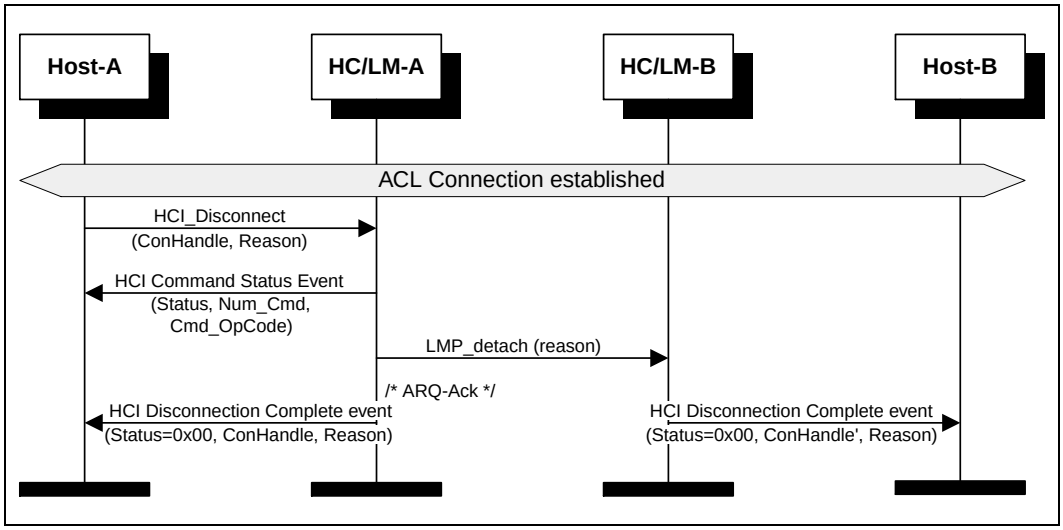

Figure 3.6: ACL Disconnection

4 OPTIONAL ACTIVITIES AFTER ACL CONNECTION ESTABLISHMENT

4.1 AUTHENTICATION REQUESTED

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

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

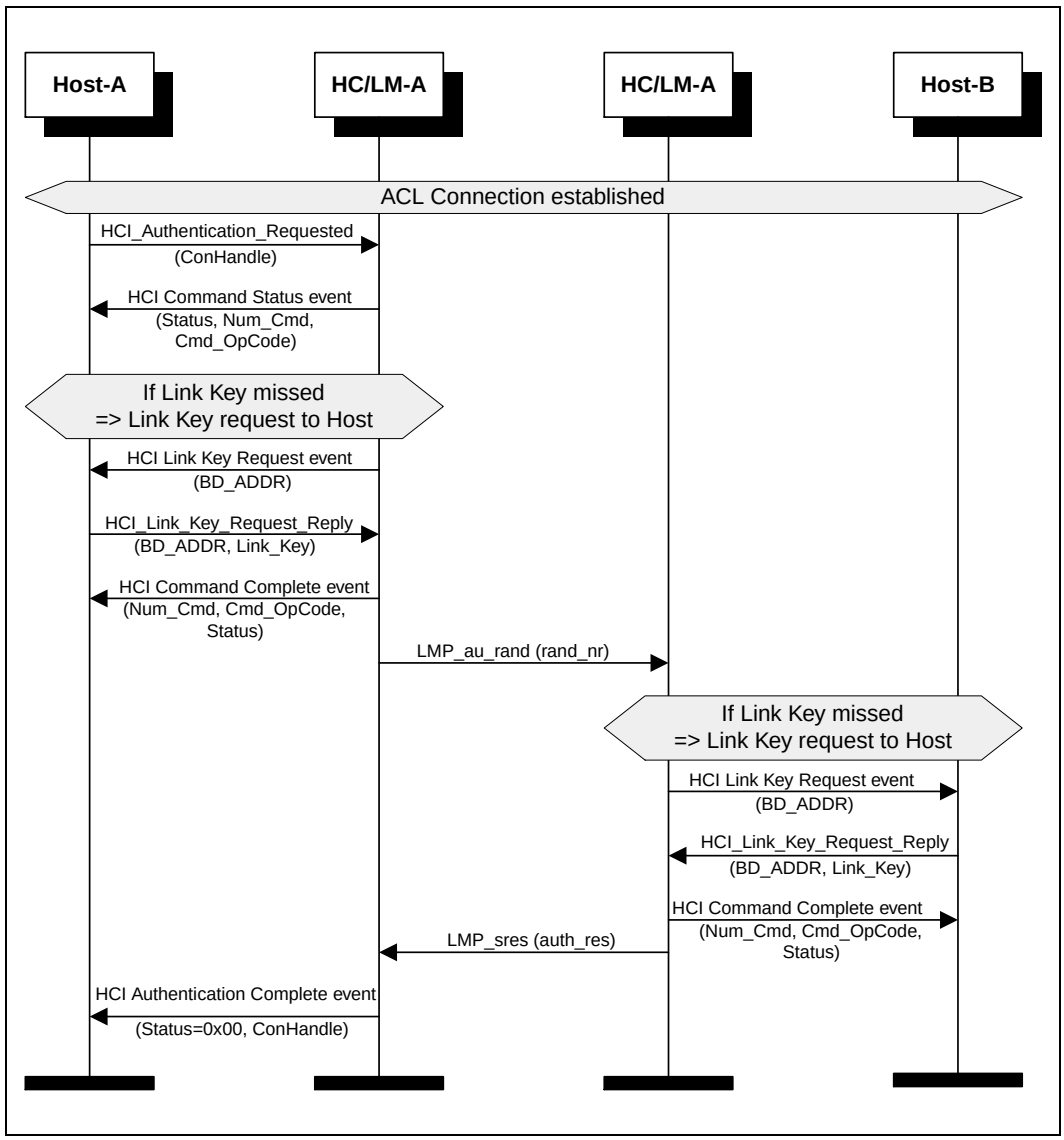

Figure 4.1: Authentication Requested

4.2 SET CONNECTION ENCRYPTION

Using the command HCI_Set_Connection_Encryption (Connection_Handle, Encryption_Enable), the Host is able to switch the encryption of a connection with the specified Connection_Handle to ON/OFF. This command can be applied on both the master- and slave sides (only the master side is shown in Figure 4.2 Set Connection Encryption). If this command occurs on the slave side, the only difference is that LMP_encryption_mode_req (encryption_mode) will be sent from the HC/LM Slave. LMP_encryption_key_size_req (key_size) and LMP_start_encryption_req (rand_nr) will still be requested from the HC/LM master.

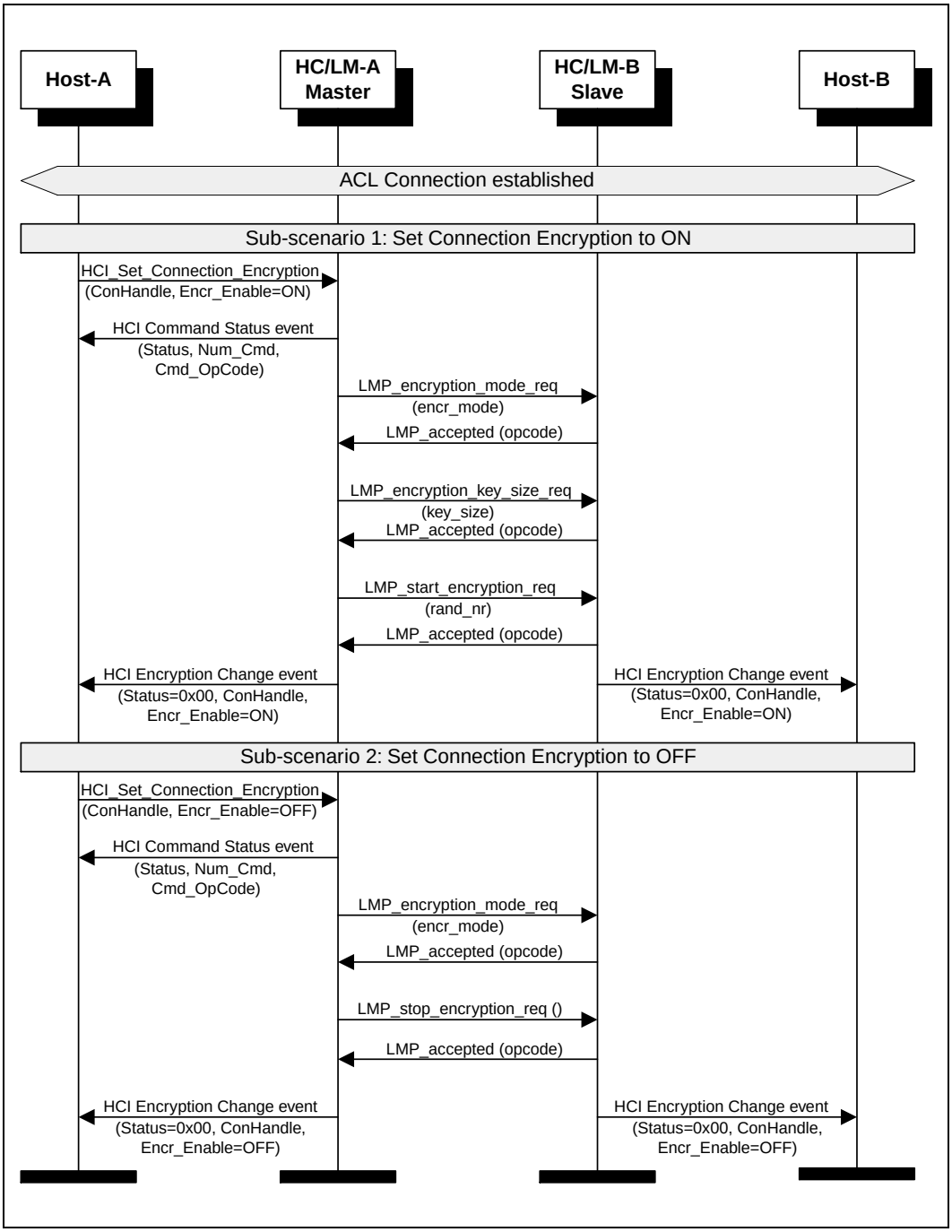

Figure 4.2: Set Connection Encryption

4.3 CHANGE CONNECTION LINK KEY

Using the command HCI_Change_Connection_Link_Key (Connection_Handle), the Host can explicitly change the common link key shared between the BT Devices.

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

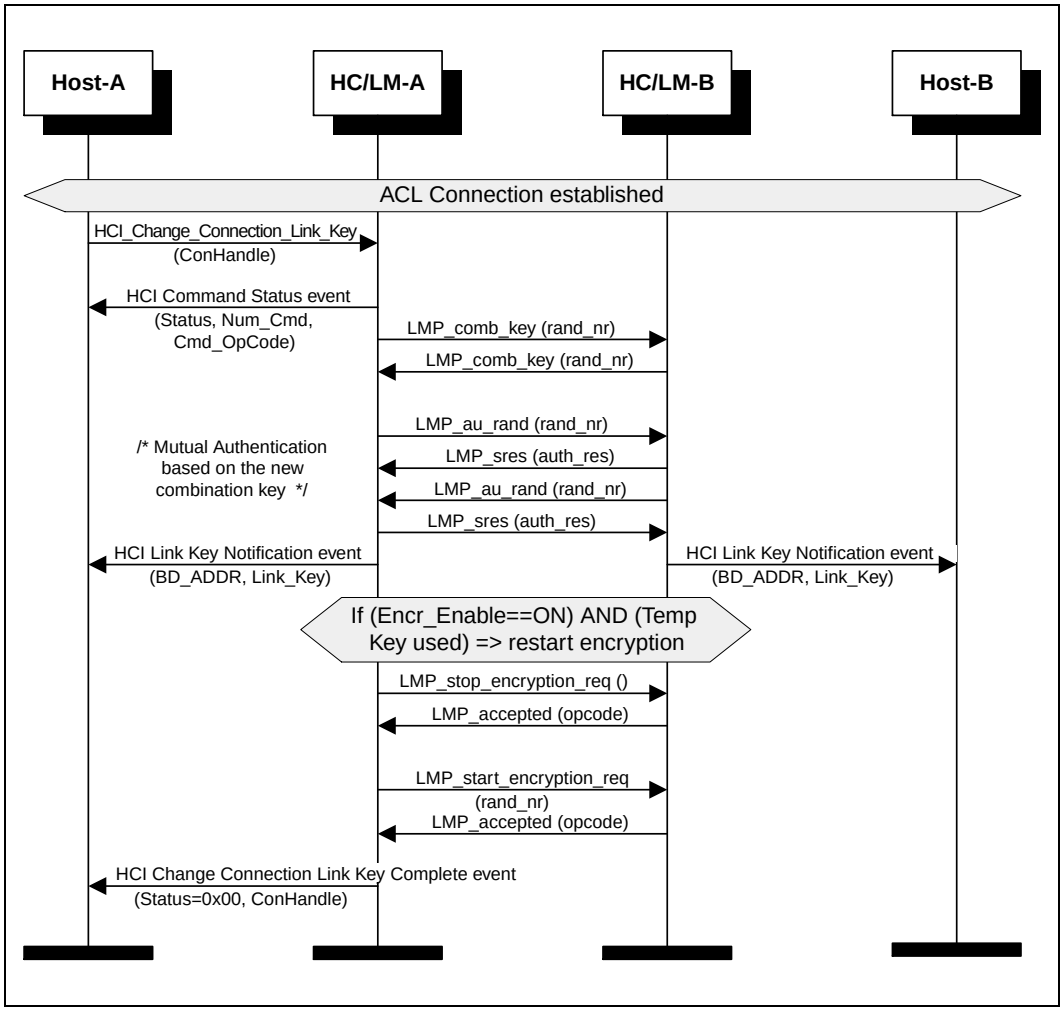

Figure 4.3: Change Connection Link Key

4.4 MASTER LINK KEY

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

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

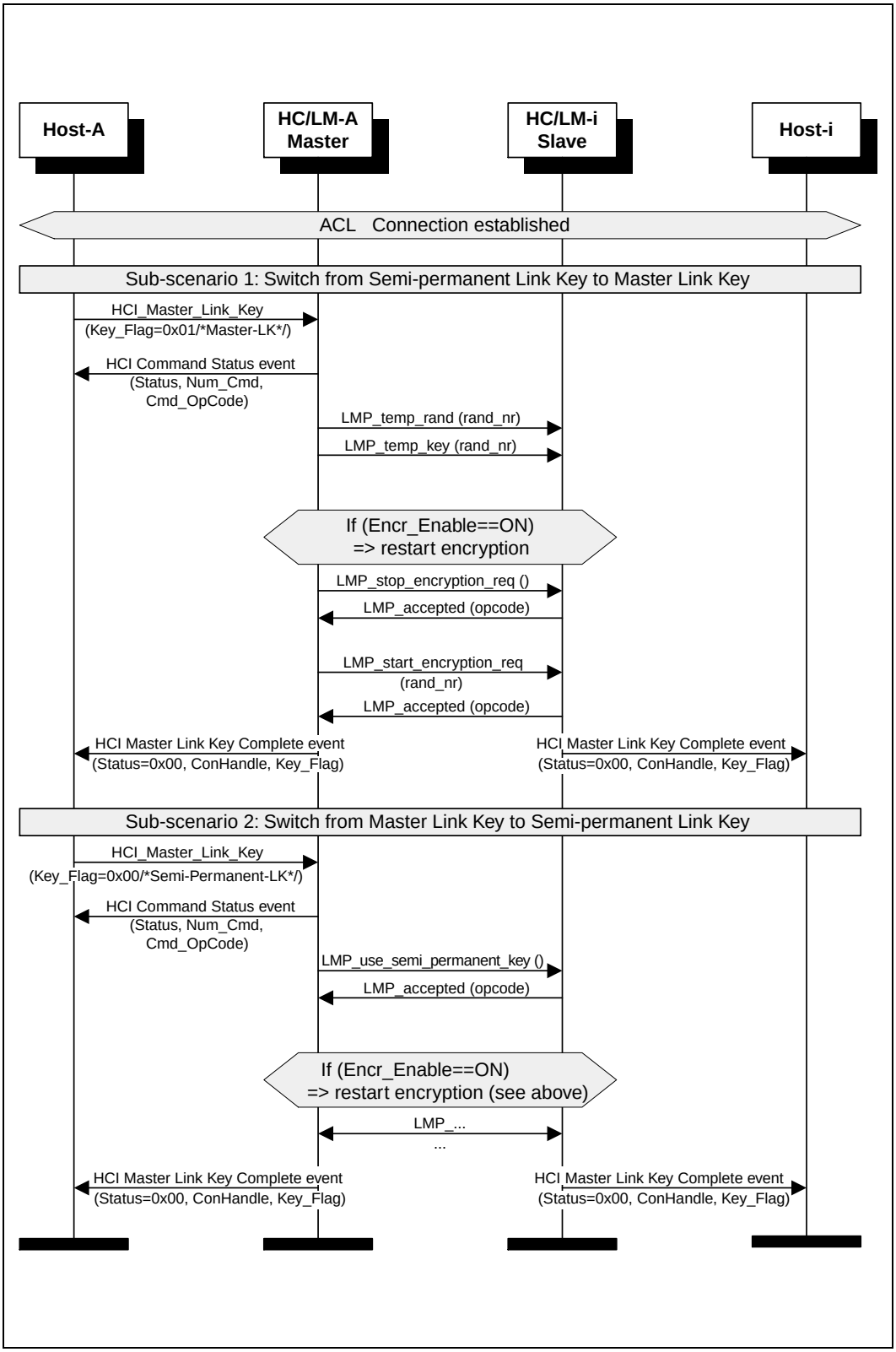

Figure 4.4: Master Link Key

4.5 READ REMOTE SUPPORTED FEATURES

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

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

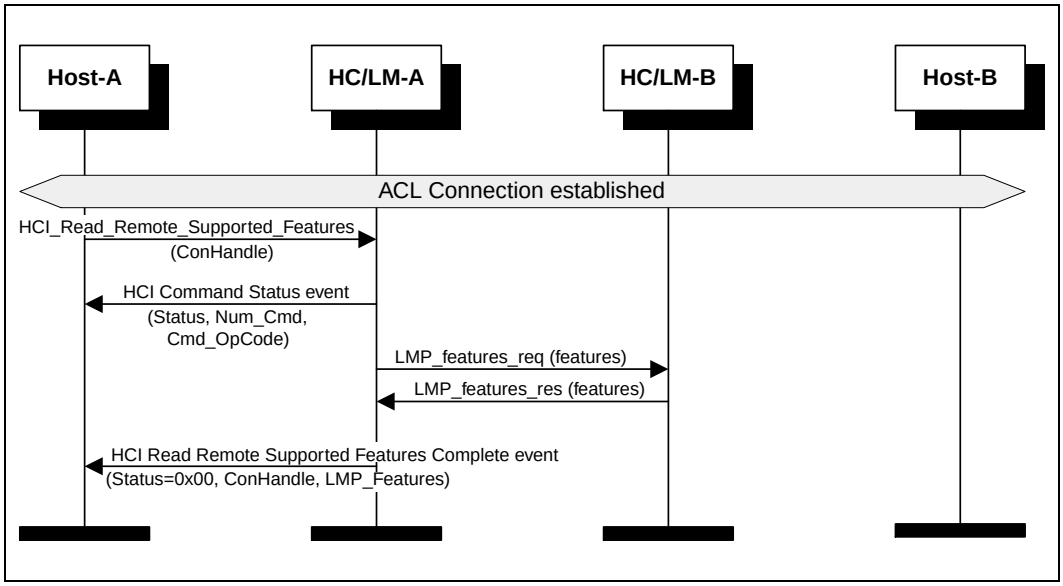

Figure 4.5: Read Remote Supported Features

4.6 READ CLOCK OFFSET

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

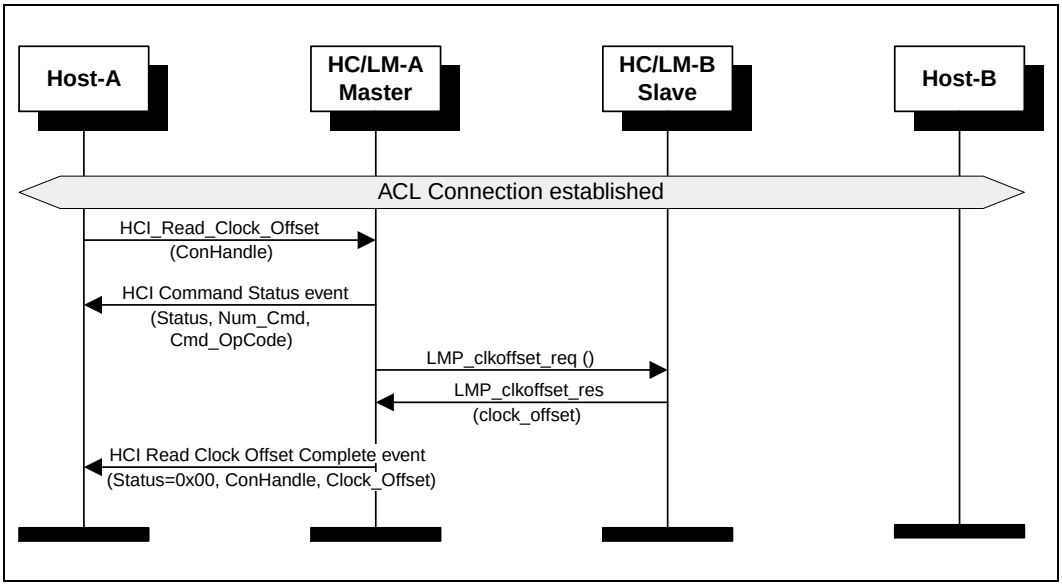

Figure 4.6: Read Clock Offset

4.7 READ REMOTE VERSION INFORMATION

Using HCI_Read_Remote_Version_Information (Connection_Handle) the version information consisting of LMP_Version, Manufacturer_Name and LMP_Subversion from the remote BT Device can be read.

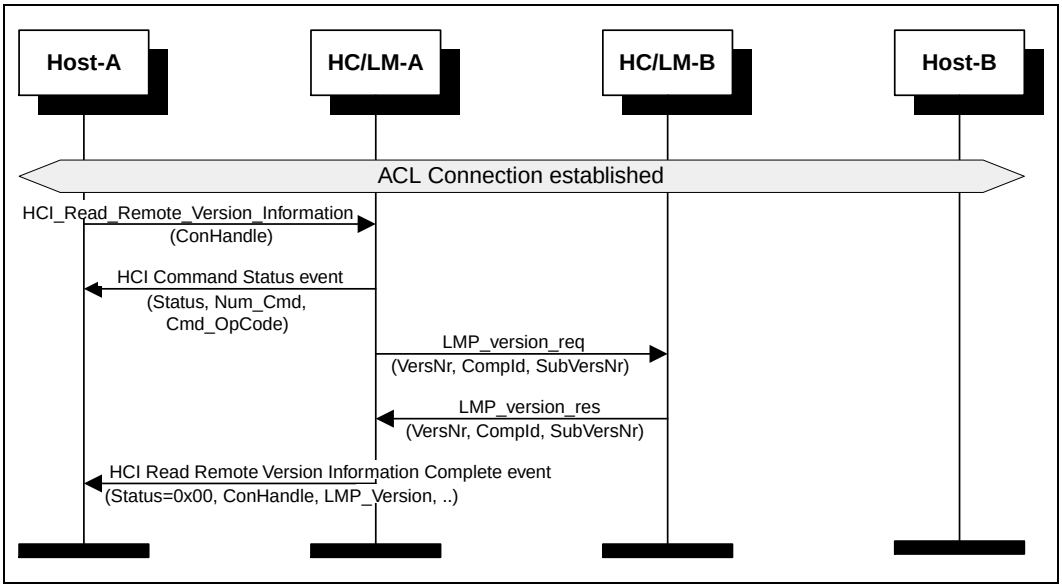

Figure 4.7: Read Remote Version Information

4.8 QOS SETUP

To set up the Quality of Service, the command HCI_QoS_Setup (Connection_Handle, Flags, Service_Type, Token_Rate, Peak_Bandwidth, Latency, Delay_Variation) is used.

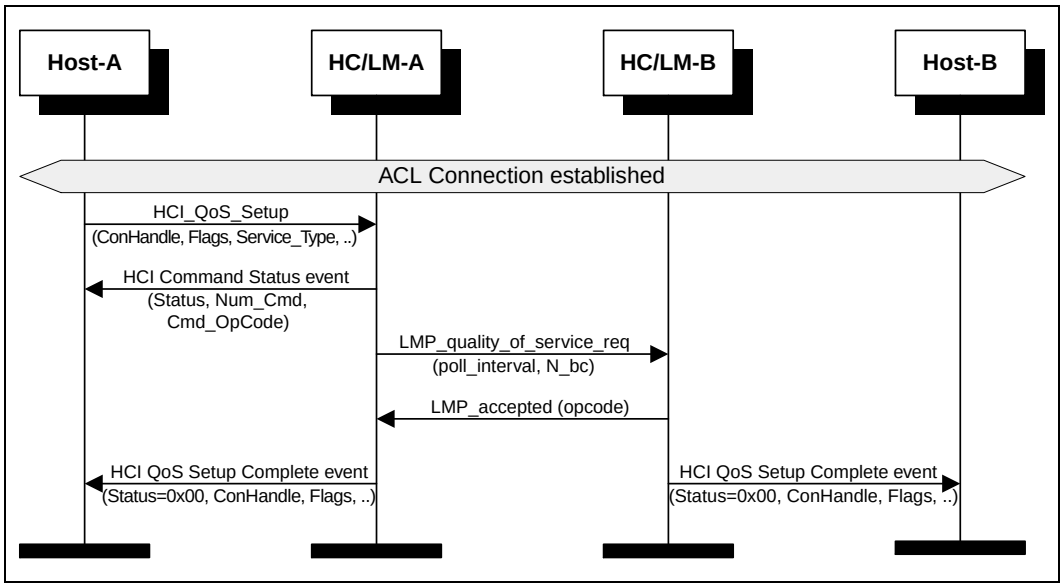

Figure 4.8: QoS Setup

4.9 SWITCH ROLE

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

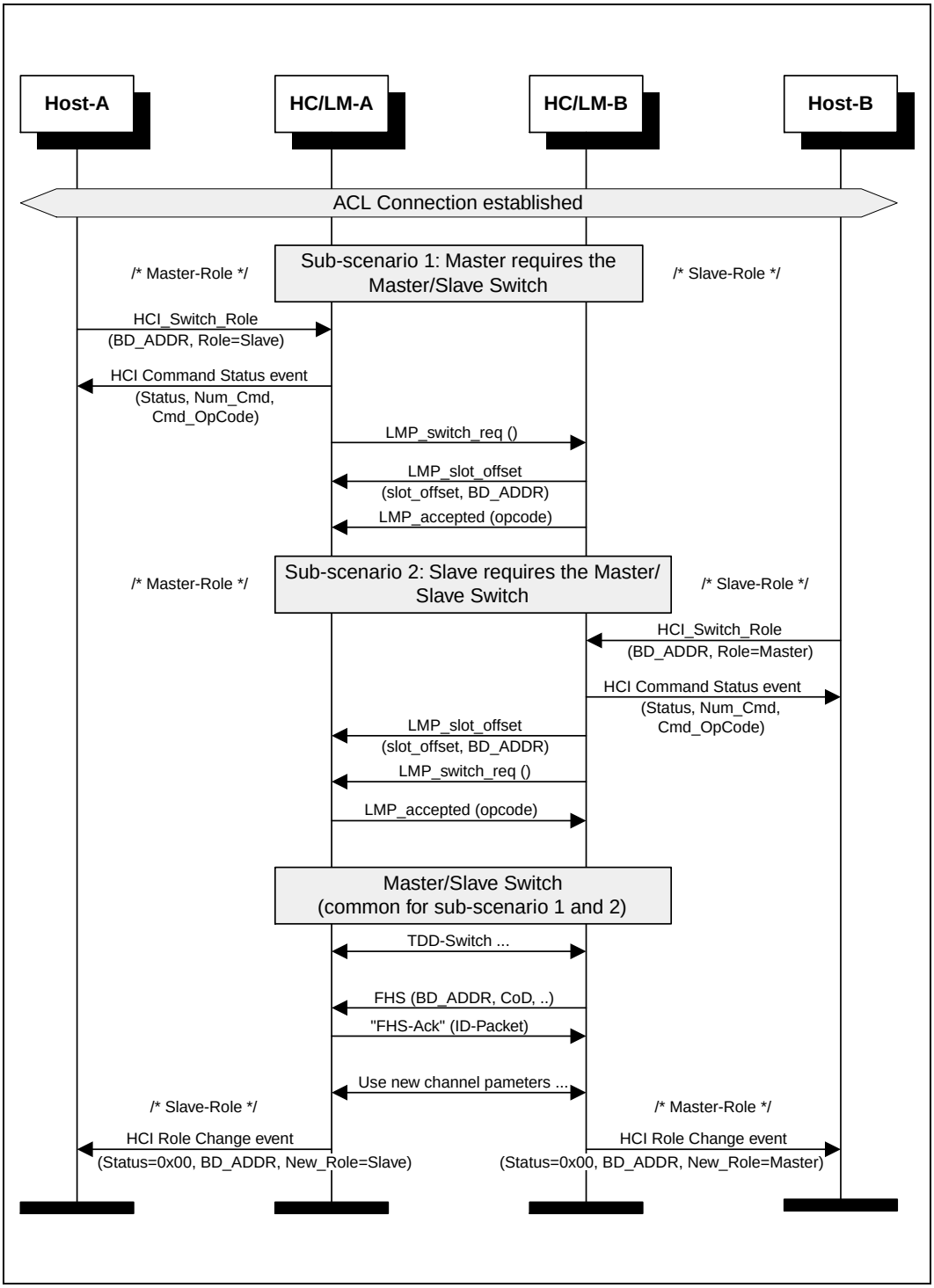

Figure 4.9: Switch Role

5 SCO CONNECTION ESTABLISHMENT AND DETACHMENT

5.1 SCO CONNECTION SETUP

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

Note: On the slave side, an incoming connection request can be automatically accepted by using HCI_Set_Event_Filter (Filter_Type, Filter_Condition_Type, Condition) with the Filter_Type = 0x02 /*Connection_Setup*/. Furthermore, for each SCO Link to a BT Device, a separate SCO Connection Handle is needed.

5.1.1 Master activates the SCO Connection setup

To set up an SCO Connection, the HCI_Add_SCO_Connection (Connection_Handle, Packet_Type) command is used. The specified Connection_Handle is related to the ACL Connection that must have been created before the HCI_Add_SCO_Connection is issued.

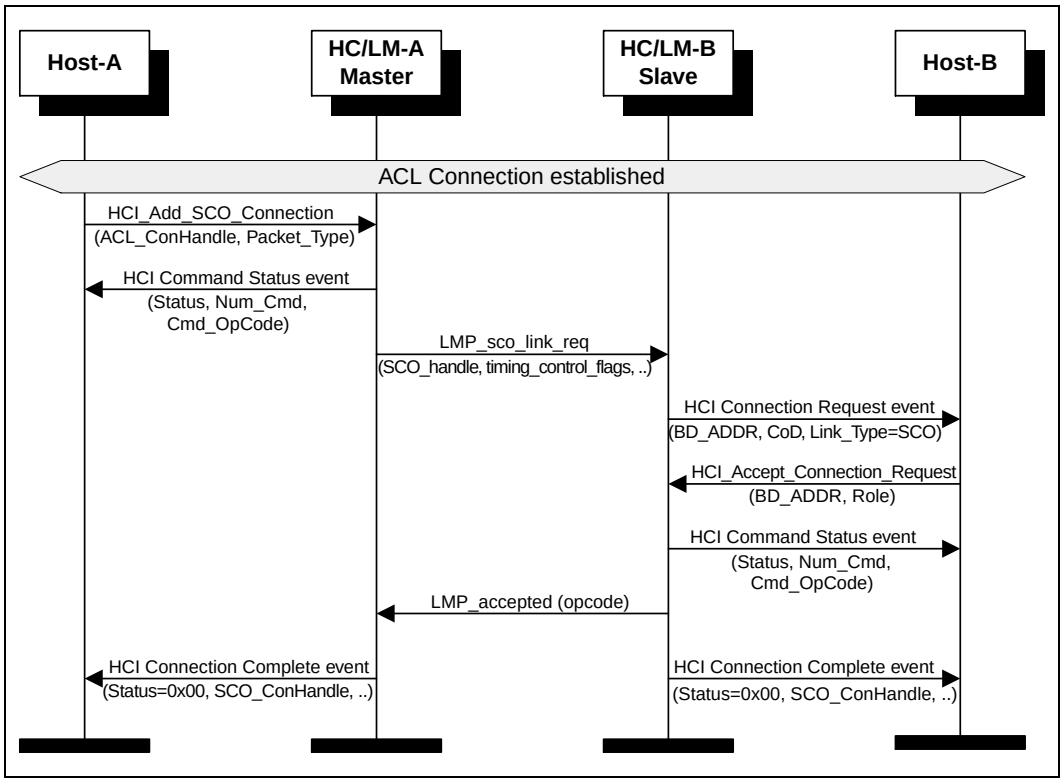

Figure 5.1: SCO Connection setup (activated from master)

5.1.2 Slave activates the SCO Connection setup

The same command HCI_Add_SCO_Connection (Connection_Handle, Packet_Type) can be used to create an SCO Link when the local BT Device is a BT Slave. Here the specified Connection_Handle belongs to the established ACL Connection between the BT Devices. Compared to 5.1.1, the only difference is that the HC/LM Slave starts the SCO Setup with LMP_sco_link_req first.

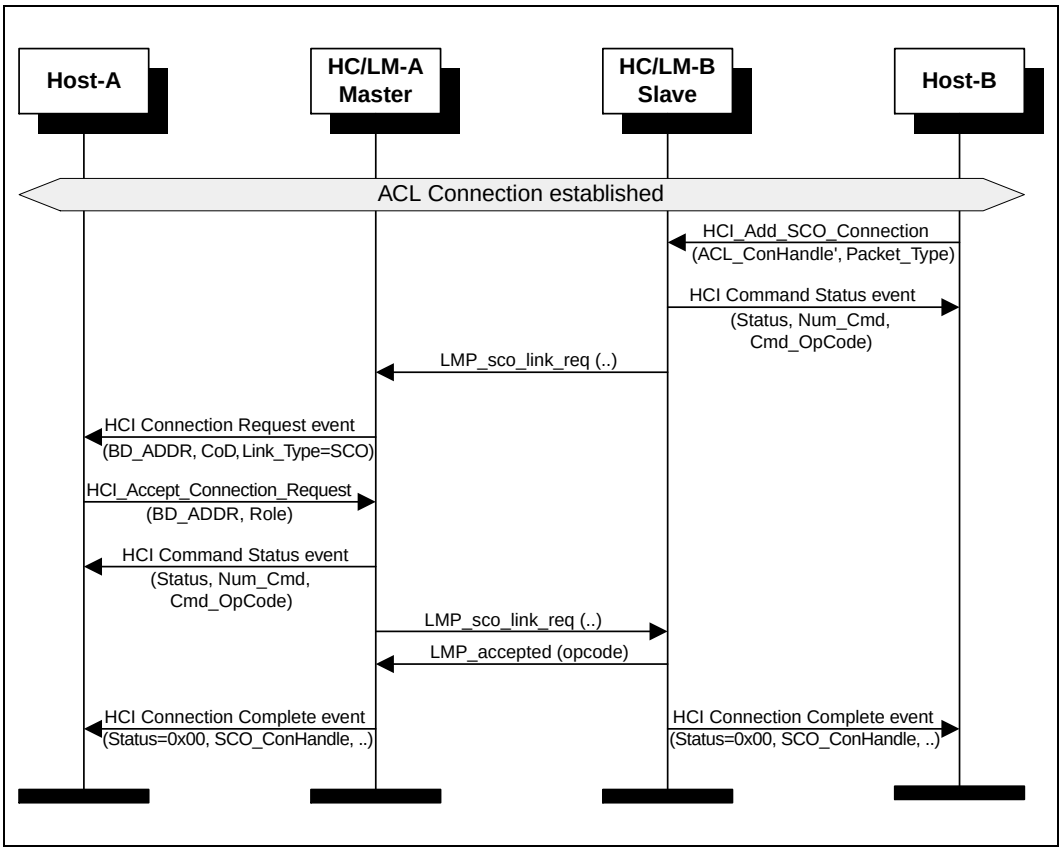

Figure 5.2: SCO Connection setup (activated from slave)

5.2 SCO DISCONNECTION

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

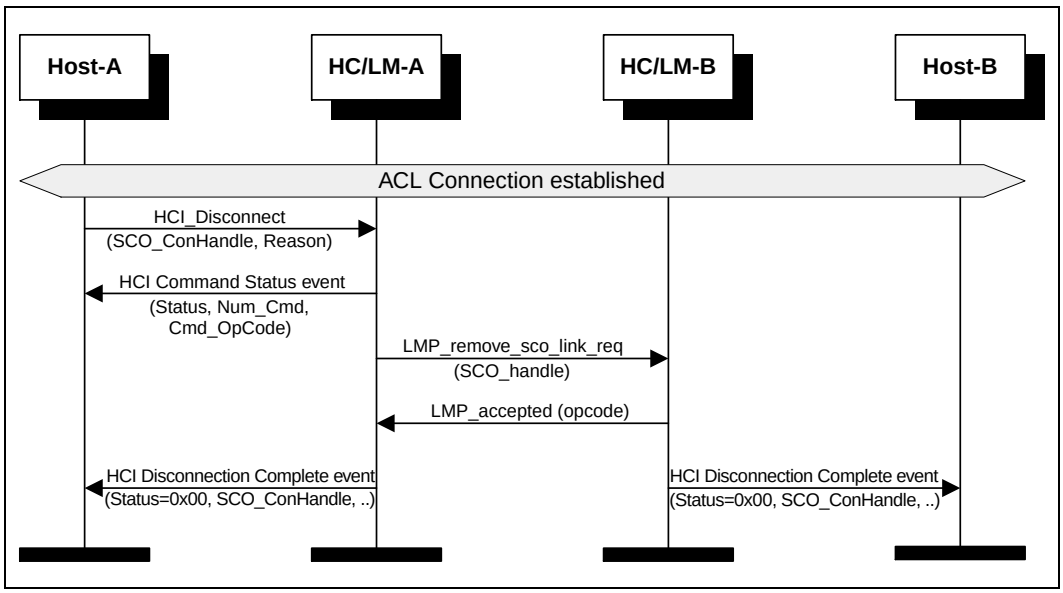

Figure 5.3: SCO Disconnection

6 SPECIAL MODES: SNIFF, HOLD, PARK

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

	Sniff	**Hold**	**Park**
Request	Master/Slave	Master/Slave	Master/Slave
Force	Master	Master/Slave	Master
Activation	Master	Master/Slave	Master
Release	Master/Slave	Automatic	Master/Slave

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

6.1 SNIFF MODE

Sniff Mode is used when a slave shall participate in the piconet only in a sniff interval. For the Sniff Mode negotiation, the Host specifies the Sniff_Max_Interval and the Sniff_Min_Interval so that HC/LM will be able to choose the one sniff interval in this range. The used command is HCI_Sniff_Mode (Connection_Handle, Sniff_Max_Interval, Sniff_Min_Interval, Sniff_Attempt, Sniff_Timeout).
Since Sniff Mode is a periodic mode, the command HCI_Exit_Sniff_Mode (Connection_Handle) is needed to return to Active Mode.

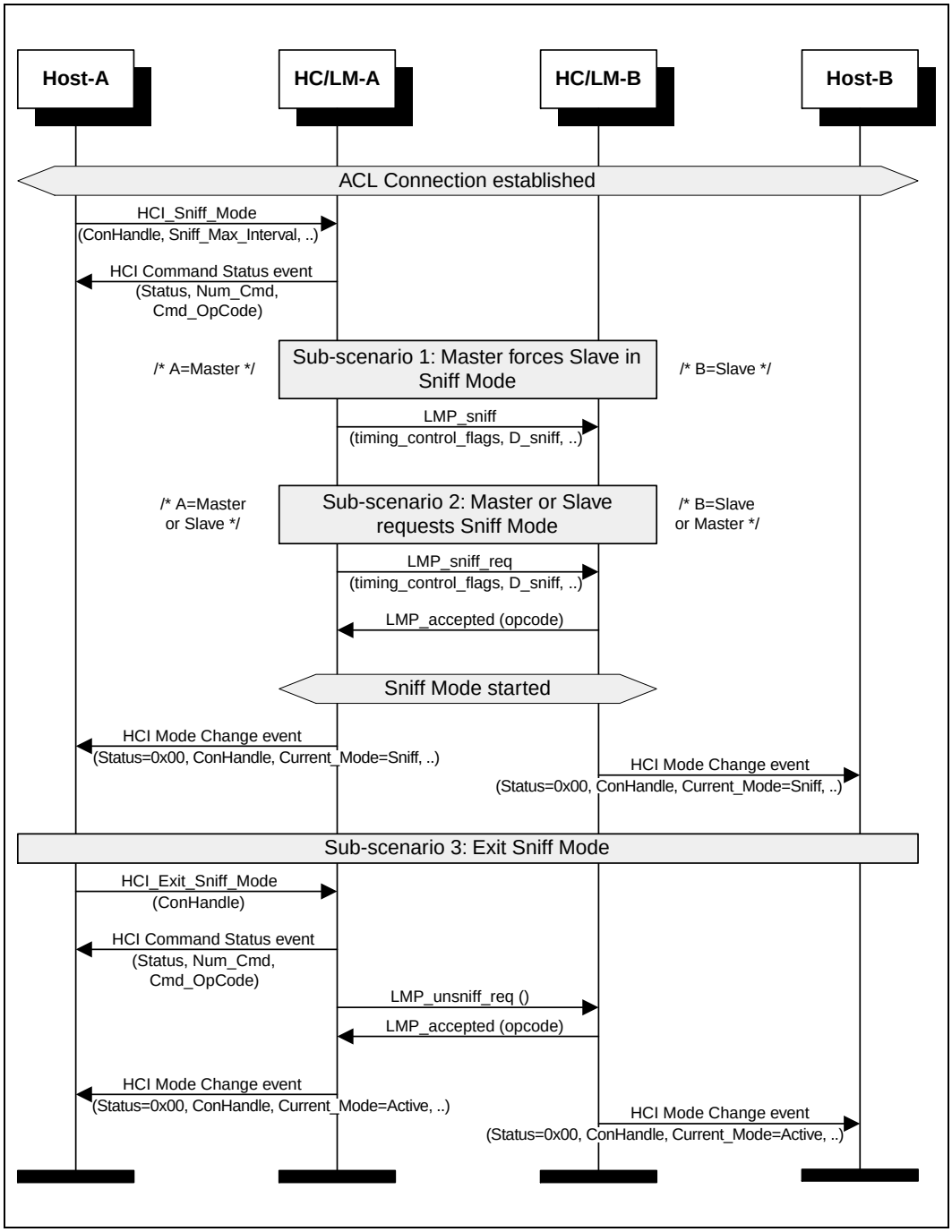

Figure 6.1: Sniff Mode

6.2 HOLD MODE

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

or complete, Mode Change event (Status, Connection_ Handle, Current_Mode, Interval) will be used to inform the Host about the actual mode.

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

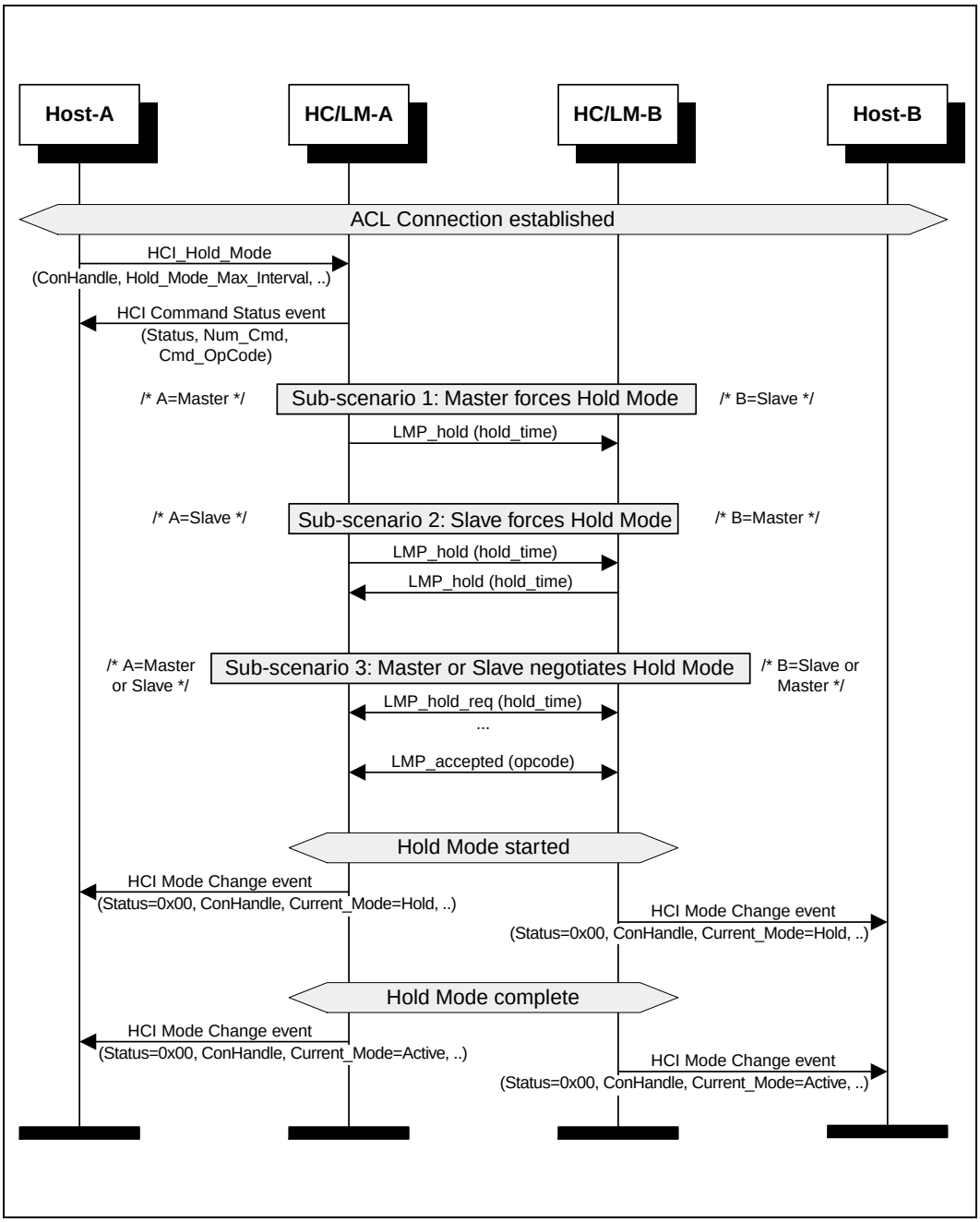

Figure 6.2: Hold Mode

6.3 PARK MODE

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

6.3.1 Enter park mode

Using the command HCI_Park_Mode (Connection_Handle, Beacon_Max_Interval, Beacon_Min_Interval) the Host specifies the Beacon_Max_Interval and Beacon_Min_Interval so that HC/LM can set up a Beacon-Interval in this range for the BT Slaves. In Park Mode, the BT Slave gives up its AM_ADDR.

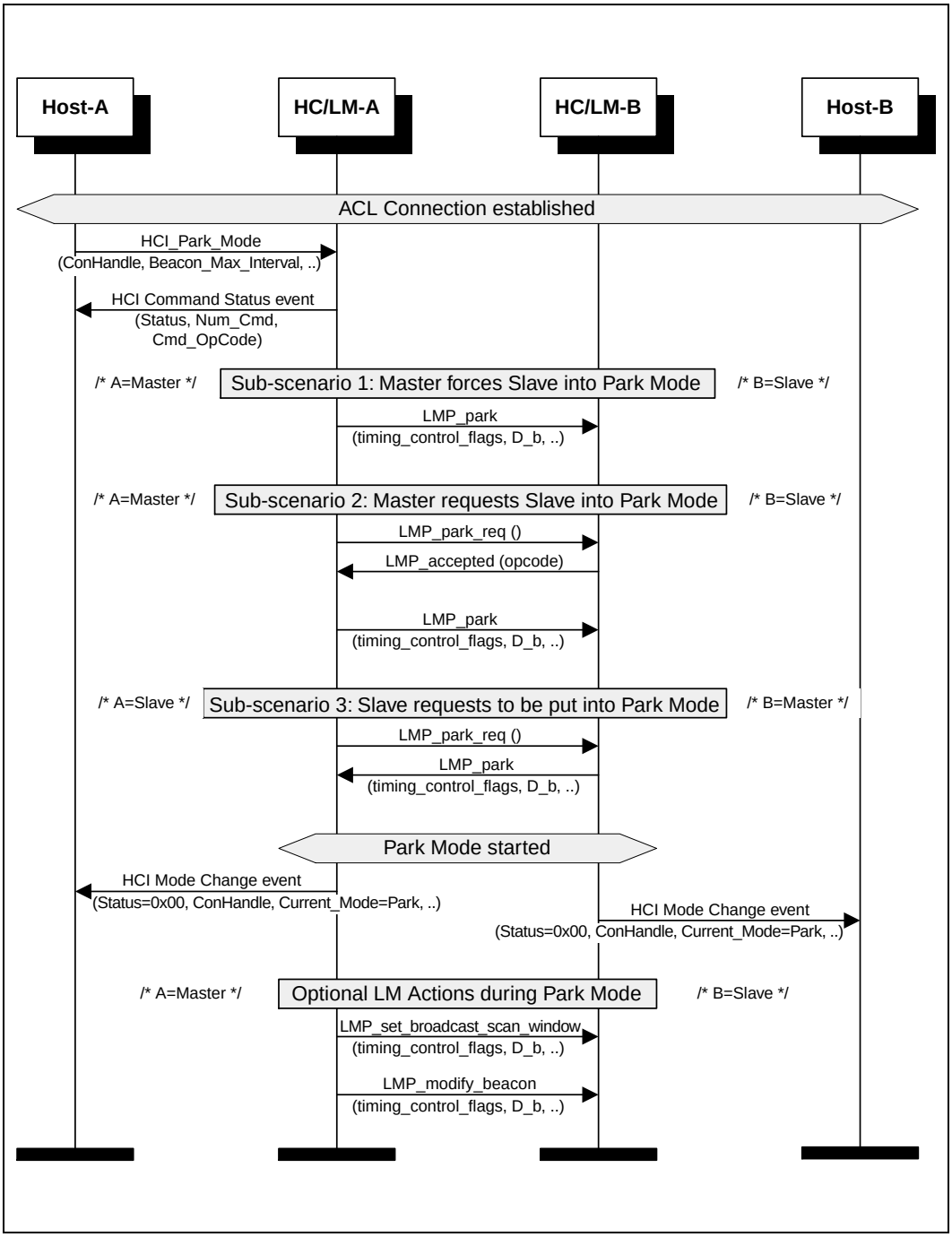

Figure 6.3: Enter Park Mode

6.3.2 Exit Park Mode

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

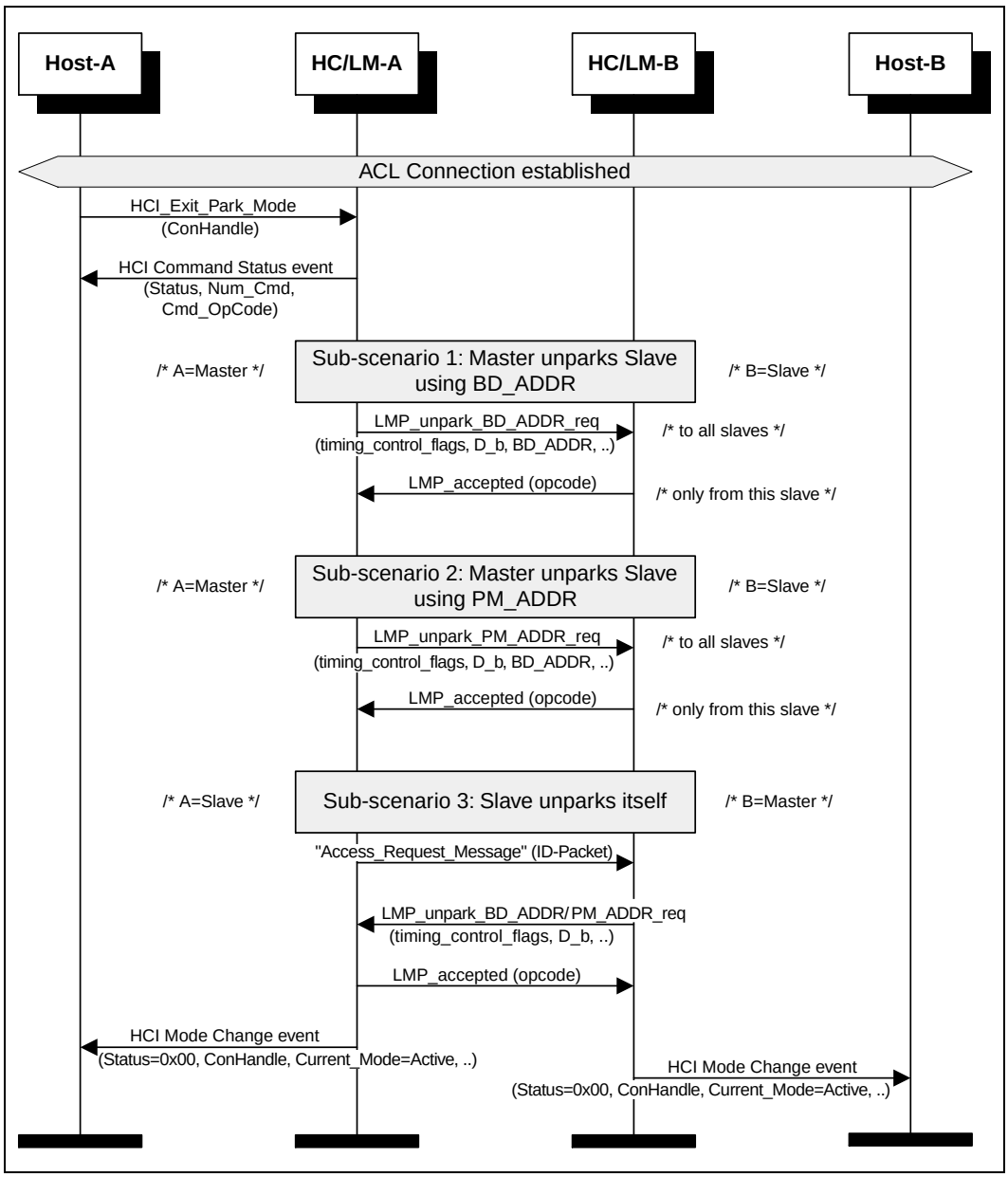

Figure 6.4: Exit Park Mode

7 BUFFER MANAGEMENT, FLOW CONTROL

The HC Data buffers are configured by the HC and managed by the Host. On initialization, the Host will issue HCI_Read_Buffer_Size. This specifies the maximum allowed length of HCI data packets sent from the Host to the HC, and the maximum number of ACL and SCO data packets that the HC can store in its buffer. After a connection is created, HC will frequently inform the Host about the number of sent packets using Number Of Completed Packets event (Number_of_Handles, Connection_Handle[I], HC_Num_Of_ Completed_Packets[i]) (see Figure 7.1 Host-to-HC flow control).

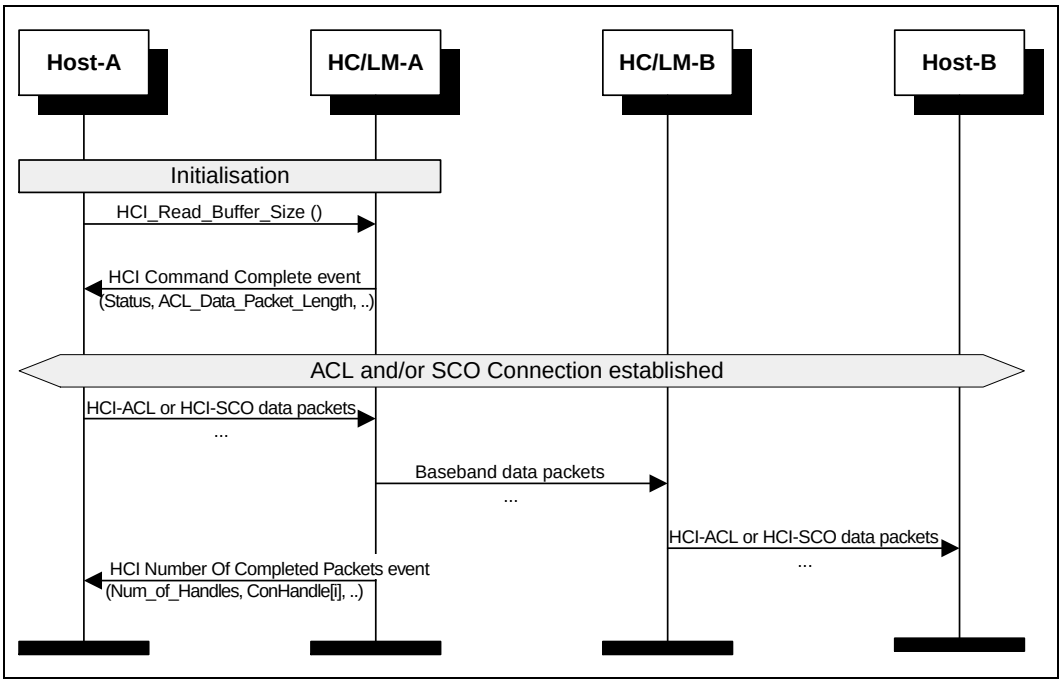

Figure 7.1: Host to HC flow control

Accordingly the HC to Host flow control can be applied in the same way so that during initialization the Host configures the Buffer Size and later the Host Controller will manage the Host Buffers.
Using HCI_Set_Host_Controller_To_Host_Flow_Control (Flow_Control_Enable) the Host can decide to apply the HC to Host flow control or not. For flow control itself HCI_Host_Buffer_Size (Host_ACL_Data_Packet_Length, Host_SCO_Data_Packet_Length, Host_Total_Num_ACL_Data_Packets, Host_Total_Num_SCO_Data_Packets) and HCI_Host_Number_Of_Completed_Packets (Number_of_Handles, Connection_Handle[I], Host_Num_Of_Completed_Packets[I]) will be used (for details see Figure 7.2 HC to Host Flow Control).

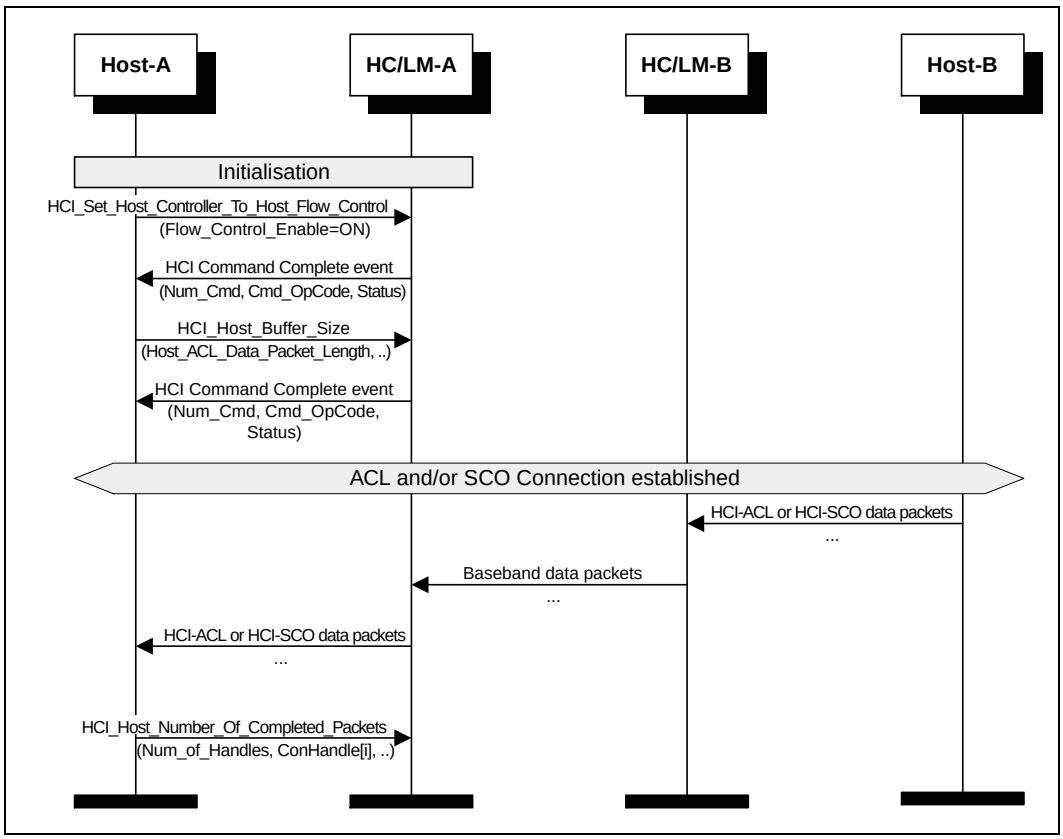

Figure 7.2: HC to Host Flow Control

8 LOOPBACK MODE

8.1 LOCAL LOOPBACK MODE

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

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

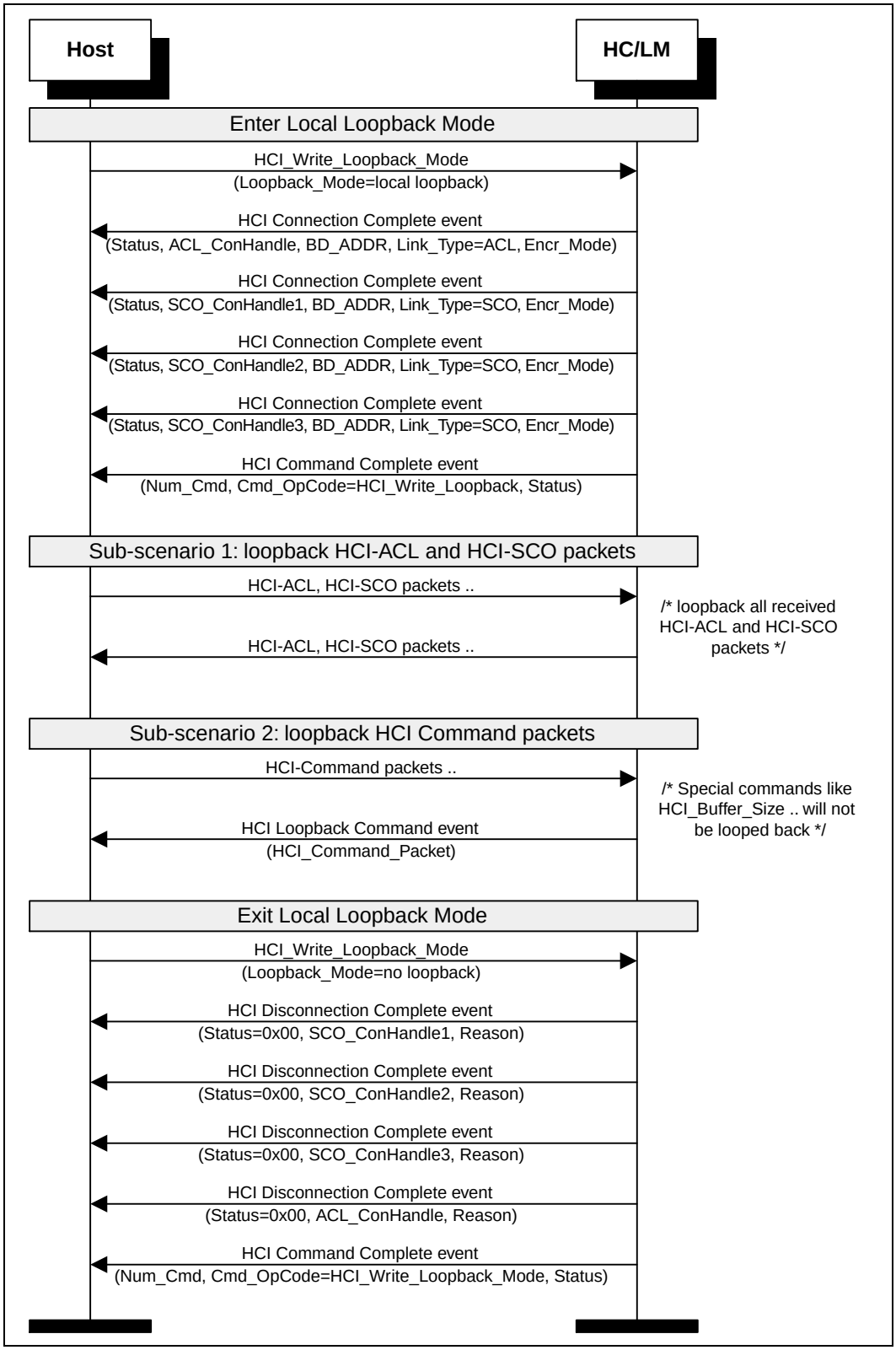

Figure 8.1: Local Loopback Mode

8.2 REMOTE LOOPBACK MODE

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

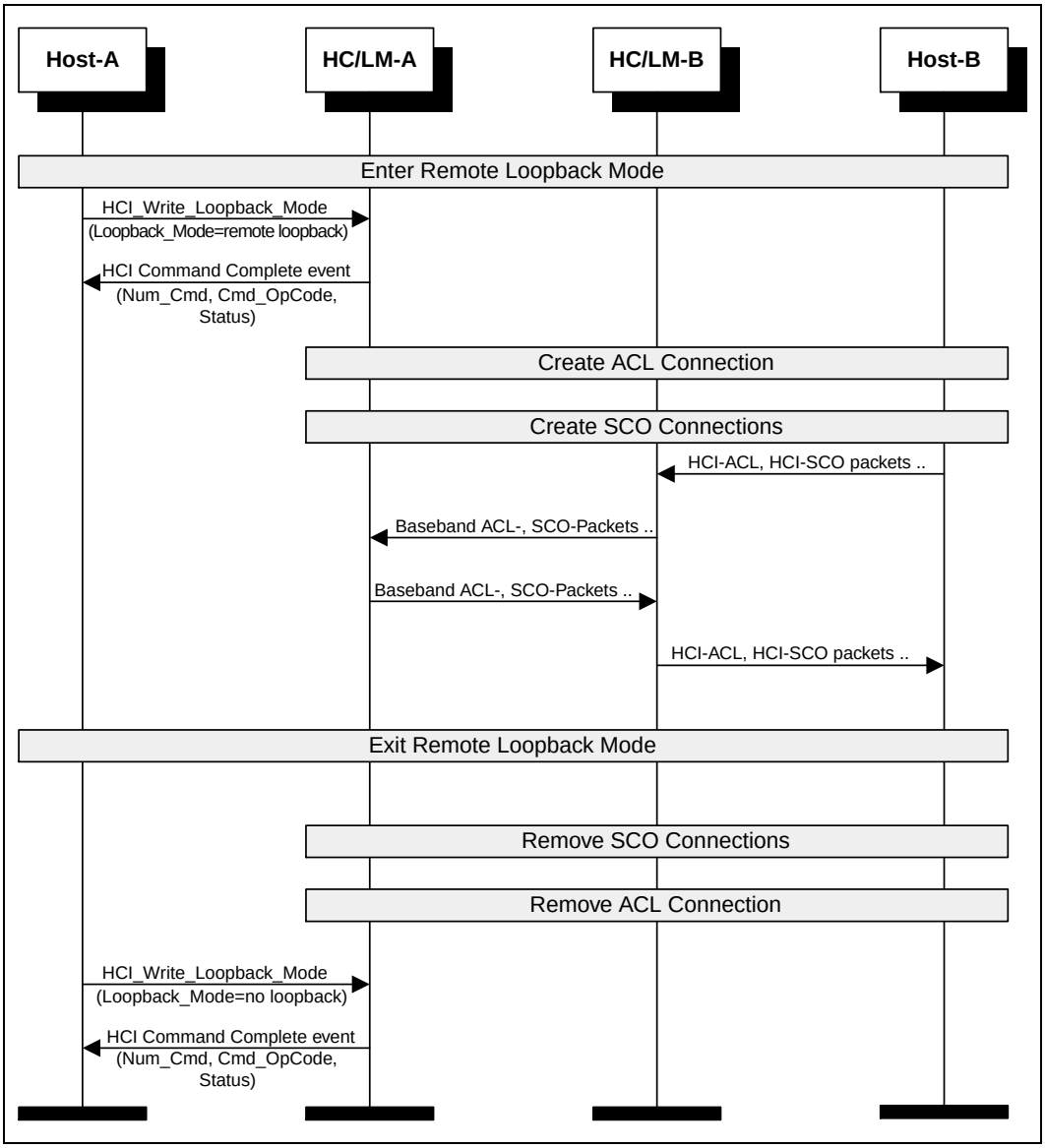

Figure 8.2: Remote Loopback Mode

9 LIST OF ACRONYMS AND ABBREVIATIONS

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

10 LIST OF FIGURES

11 LIST OF TABLES

12 REFERENCES

[1] “Baseband Specification” on page 33

[2] “Link Manager Protocol” on page 185

[3] “Host Controller Interface Functional Specification” on page 517

[4] “Logical Link Control and Adaptation Protocol Specification” on page 245