CTR

# HOST CONTROLER INTERFACE

**BLUETOOTH SPEC. Version 1.1**

**Presented by Alan Liu**

**Prepared by Leon Lee**

**(aliu@ee.ccu.edu.tw, leon@ai.ee.ccu.edu.tw)**

**September 12, 2001**

中正大學電信研究中心

Center for Telecommunication Research

# OUTLINE

- Introduction
- HCI flow control
- HCI packets
- HCI commands
- HCI events
- HCI data packets
- Message sequence chart

# OUTLINE

- **Introduction**
- **HCI flow control**
- **HCI packets**
- **HCI commands**
- **HCI events**
- **HCI data packets**
- **Message sequence chart**

# INTRODUCTION

- **What is HCI?**
  - A uniform interface method of accessing the Bluetooth hardware capabilities
  - Two parts of HCI : driver and firmware
- **Role of HCI**
  - Bluetooth protocol architecture view
  - Bluetooth device end-to-end view
  - Bluetooth low layer software view
  - Bluetooth system functional block view
  - Bluetooth hardware architecture view

# What is HCI?

- **Two parts of HCI commands**
  - HCI driver in Bluetooth host
  - HCI firmware in Bluetooth hardware

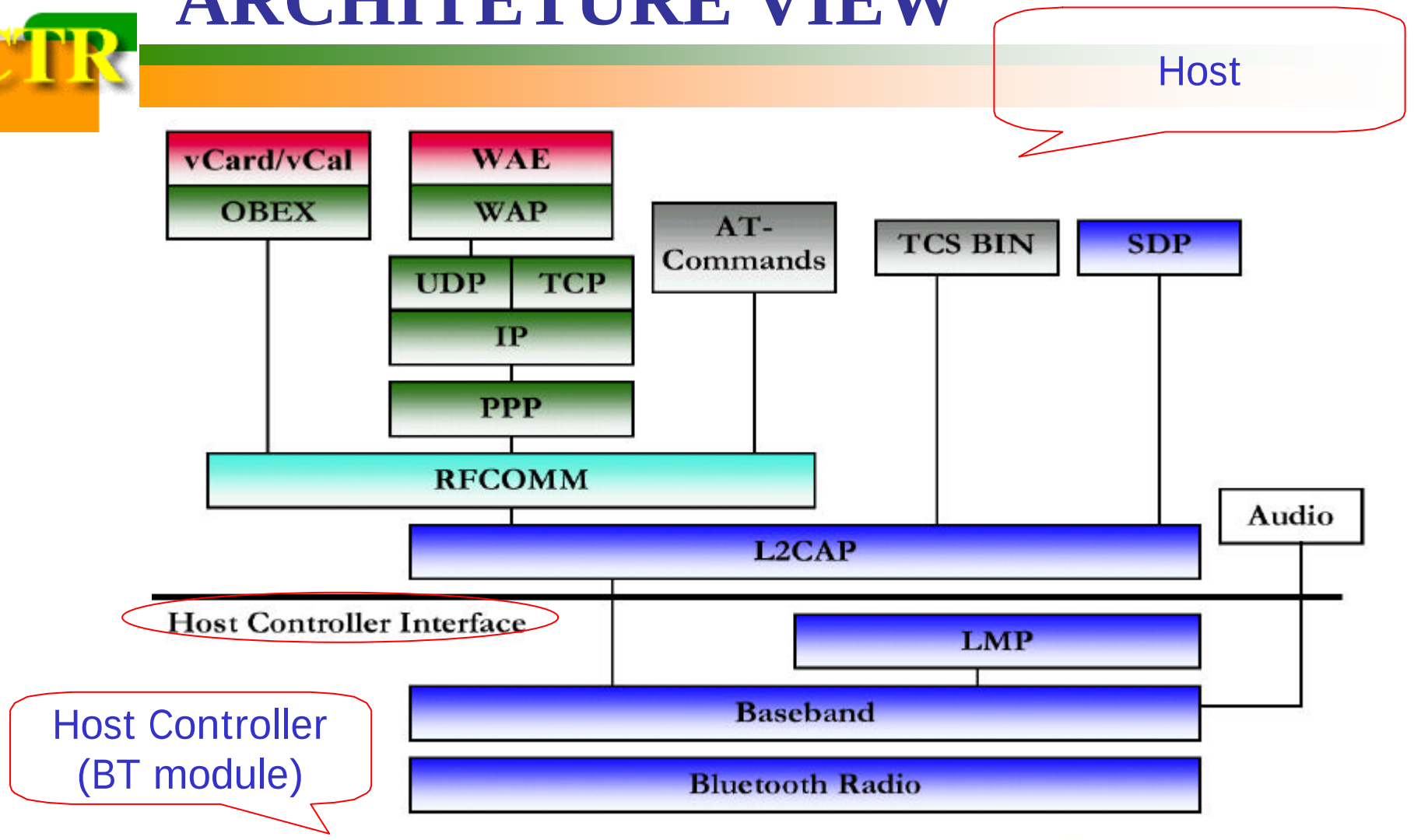
# What is HCI? (cont.)

- **HCI driver on the Bluetooth host**
  - Exchanging data and commands with HCI firmware on the Bluetooth hardware

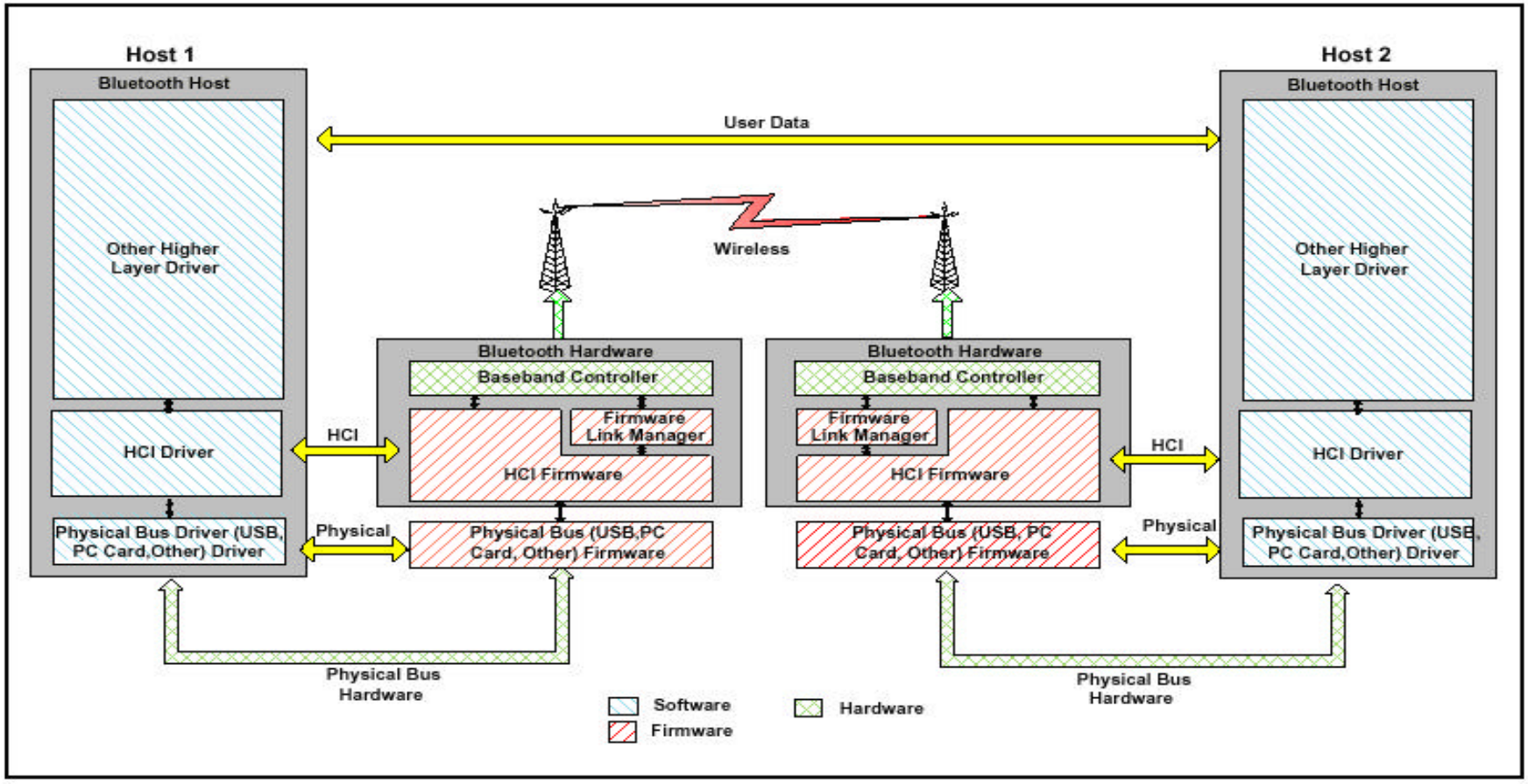
# What is HCI? (cont.)

- **HCI firmware in Bluetooth hardware**
  - Implementing the HCI commands for the Bluetooth hardware by accessing
    - baseband commands,
    - link manager commands,
    - hardware status registers,
    - control registers,
    - and event registers

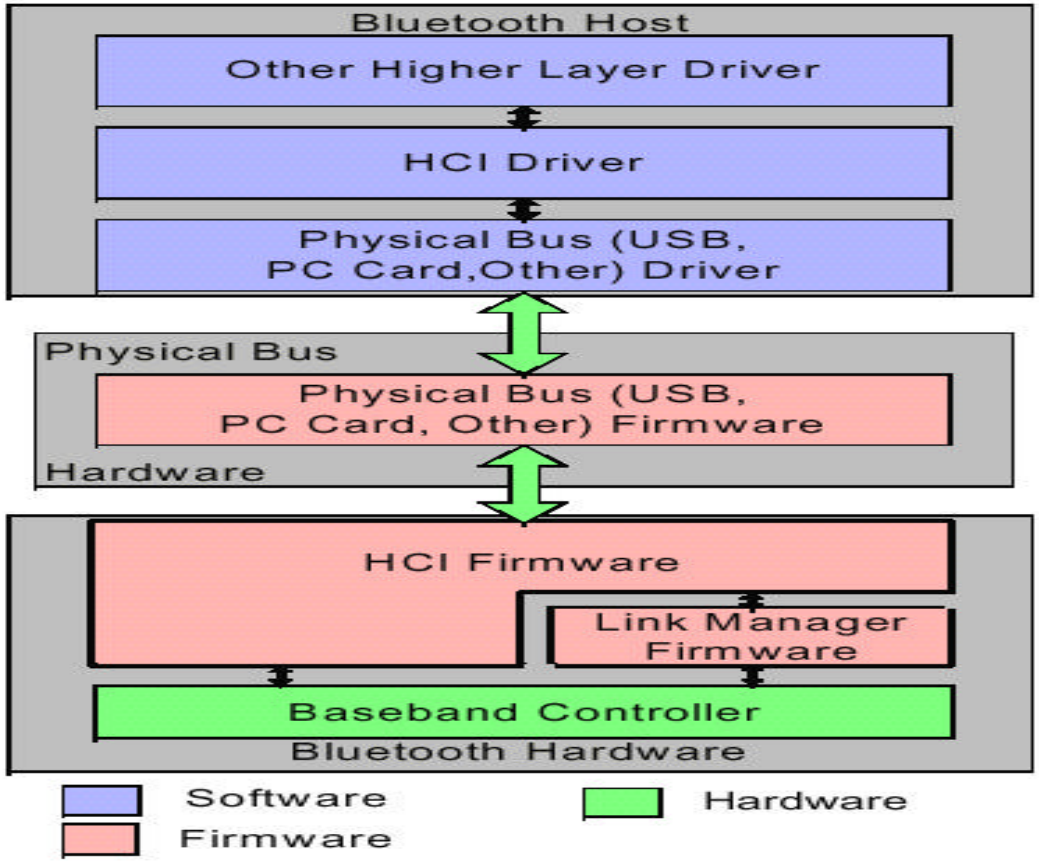
# BLUETOOTH PROTOCOL ARCHITETURE VIEW



# BLUETOOTH DEVICE END-TO-END VIEW



# BLUETOOTH LOW LAYER SOFEWARE VIEW



# BLUETOOTH SYSTEM FUNCTIONAL BLOCK VIEW

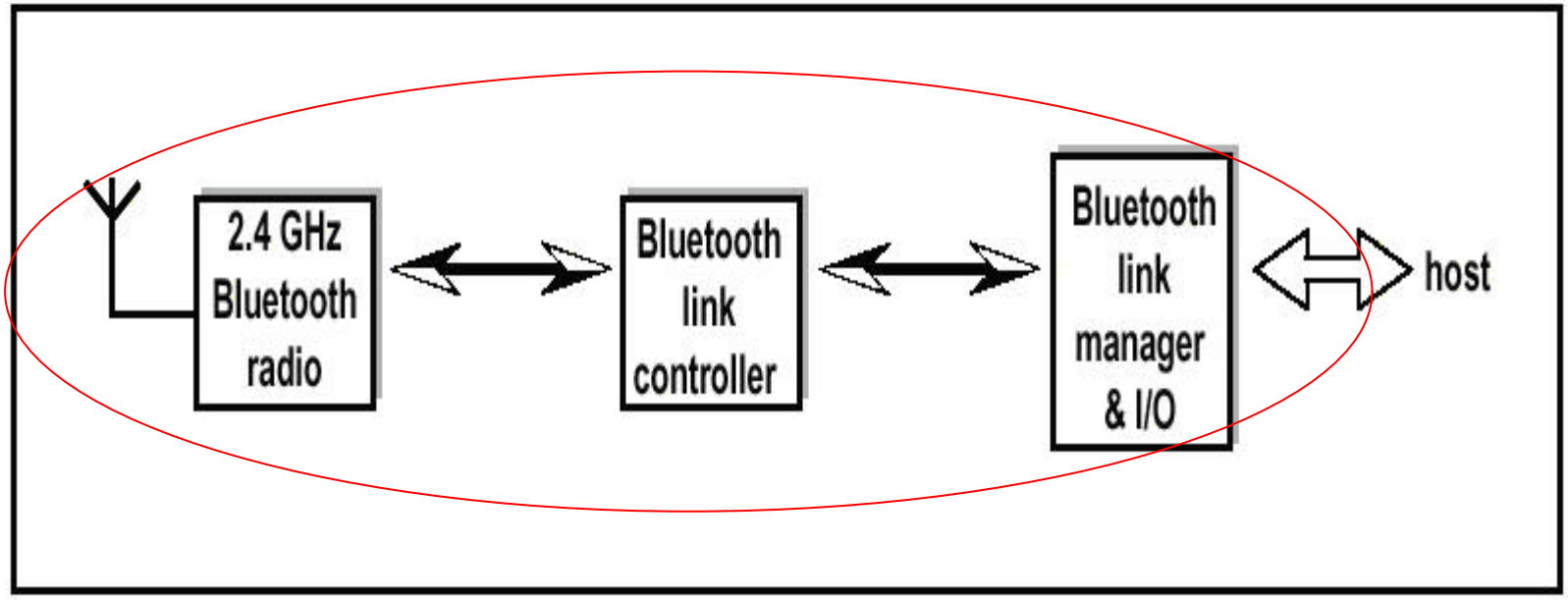



*Figure 1.1: Different functional blocks in the Bluetooth system*

# BLUETOOTH HARDWARE ARCHITECTURE VIEW

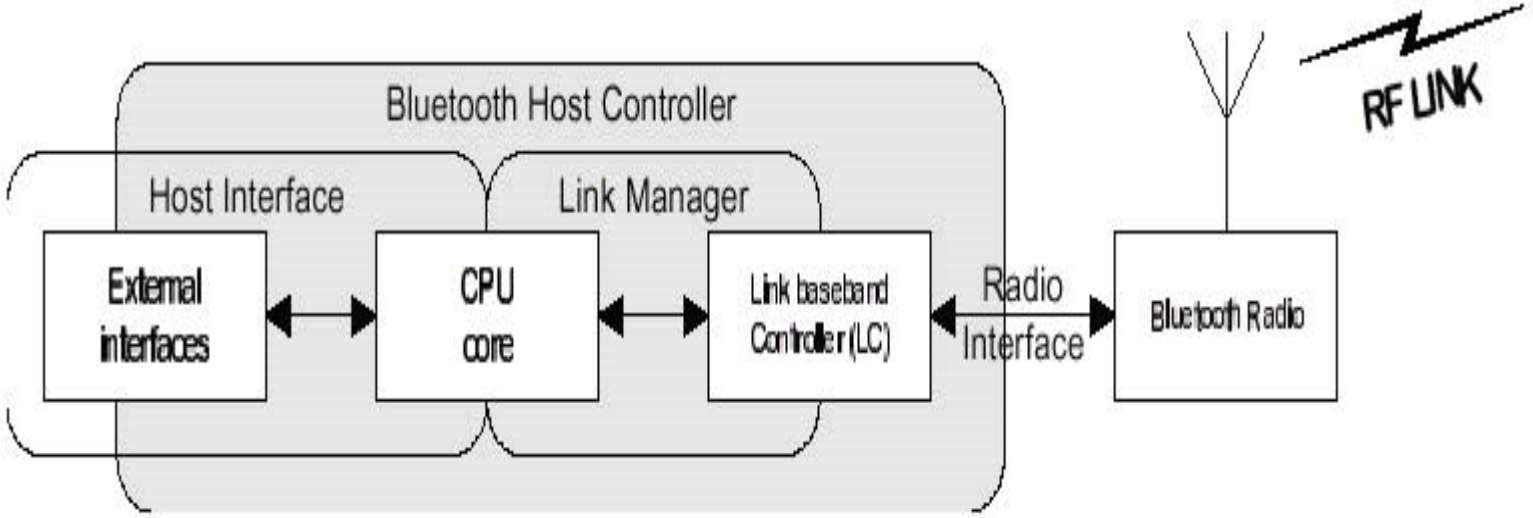



*Figure 1.3: Bluetooth Hardware Architecture Overview - the Host Controller Radio*

# EXAMPLE WITH USB

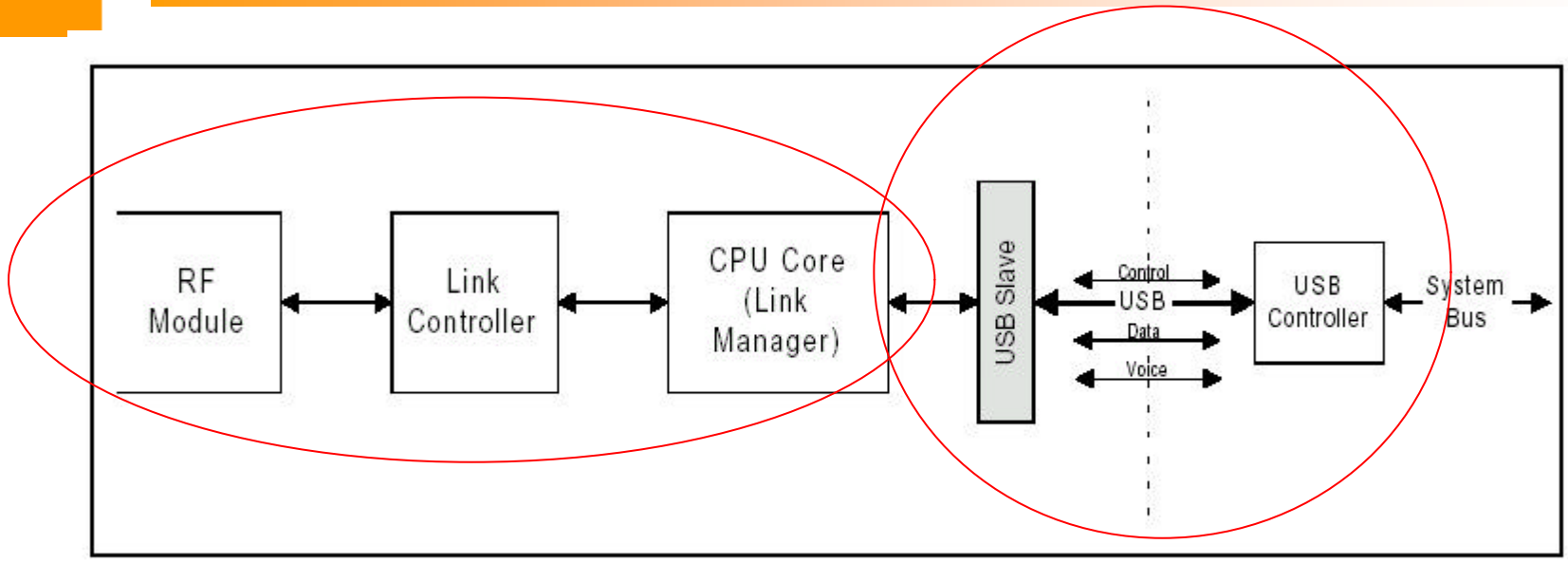



*Figure 1.4: Bluetooth Block Diagram with USB HCI*

# EXAMPLE WITH PC-CARD

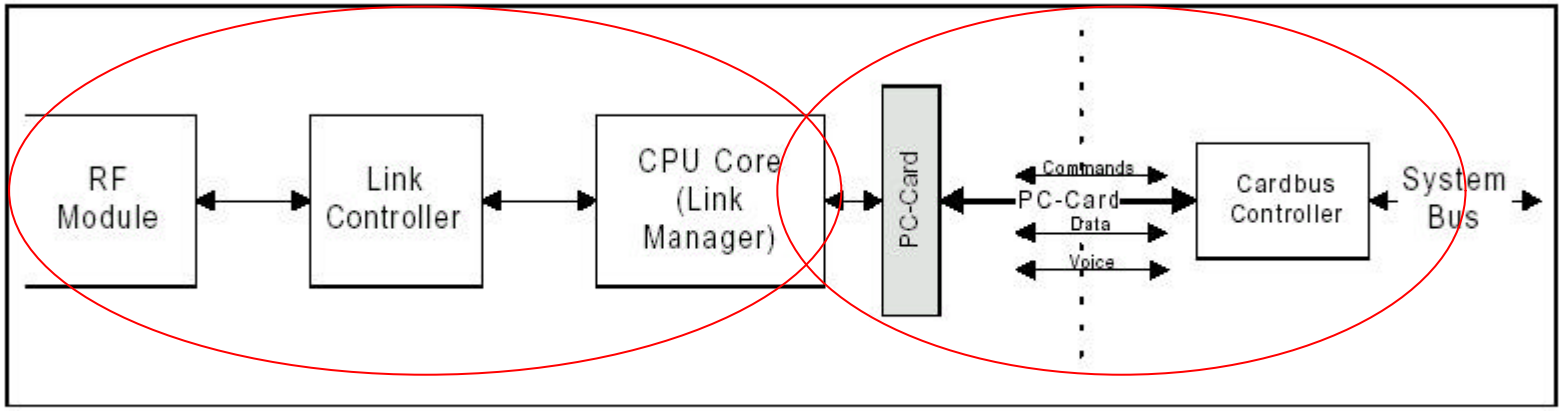



*Figure 1.5: Bluetooth Block Diagram with PC-Card HCI*

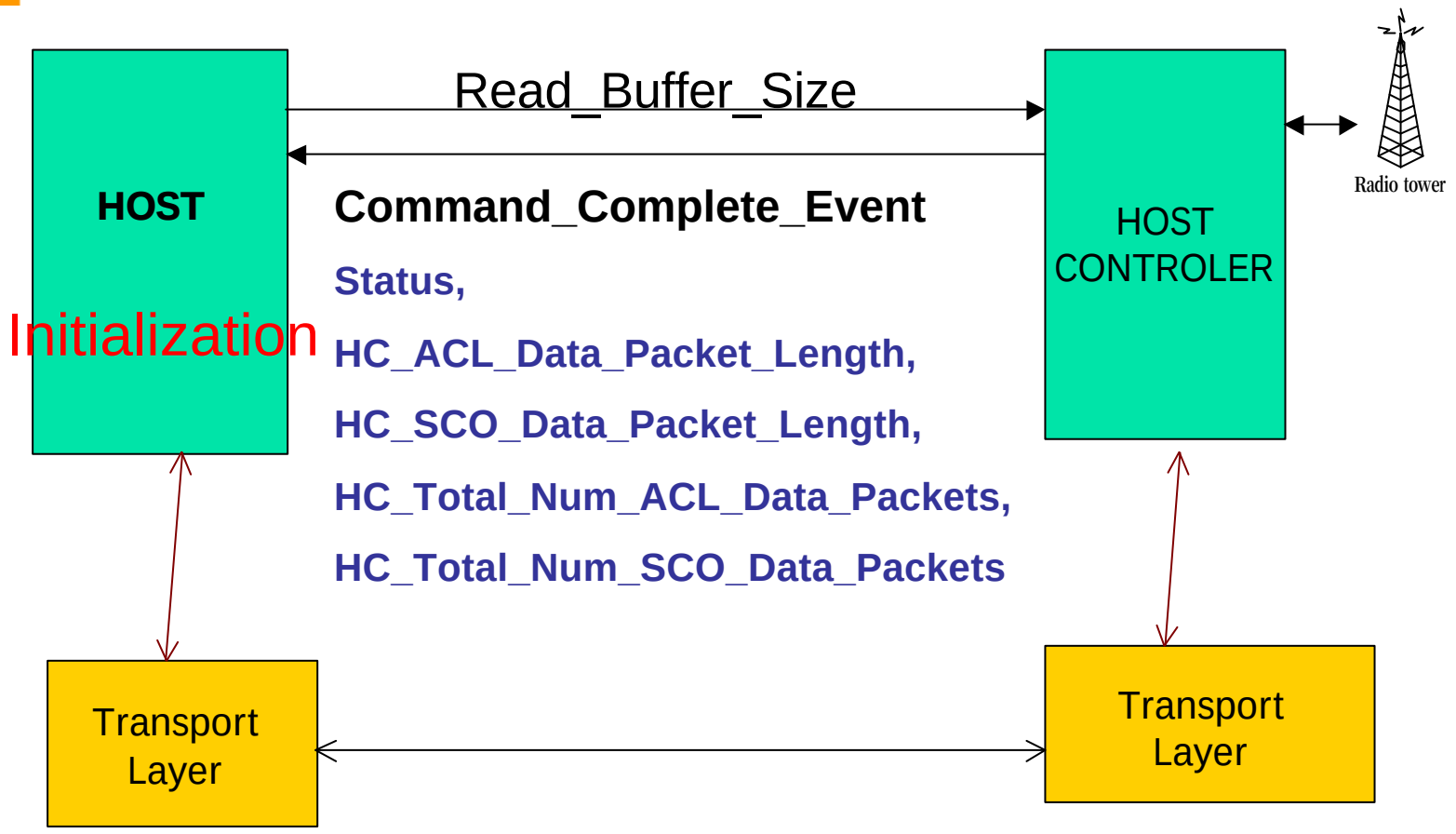
# OUTLINE

- Introduction
- HCI flow control
- HCI packets
- HCI commands
- HCI events
- HCI data packets
- Message sequence chart

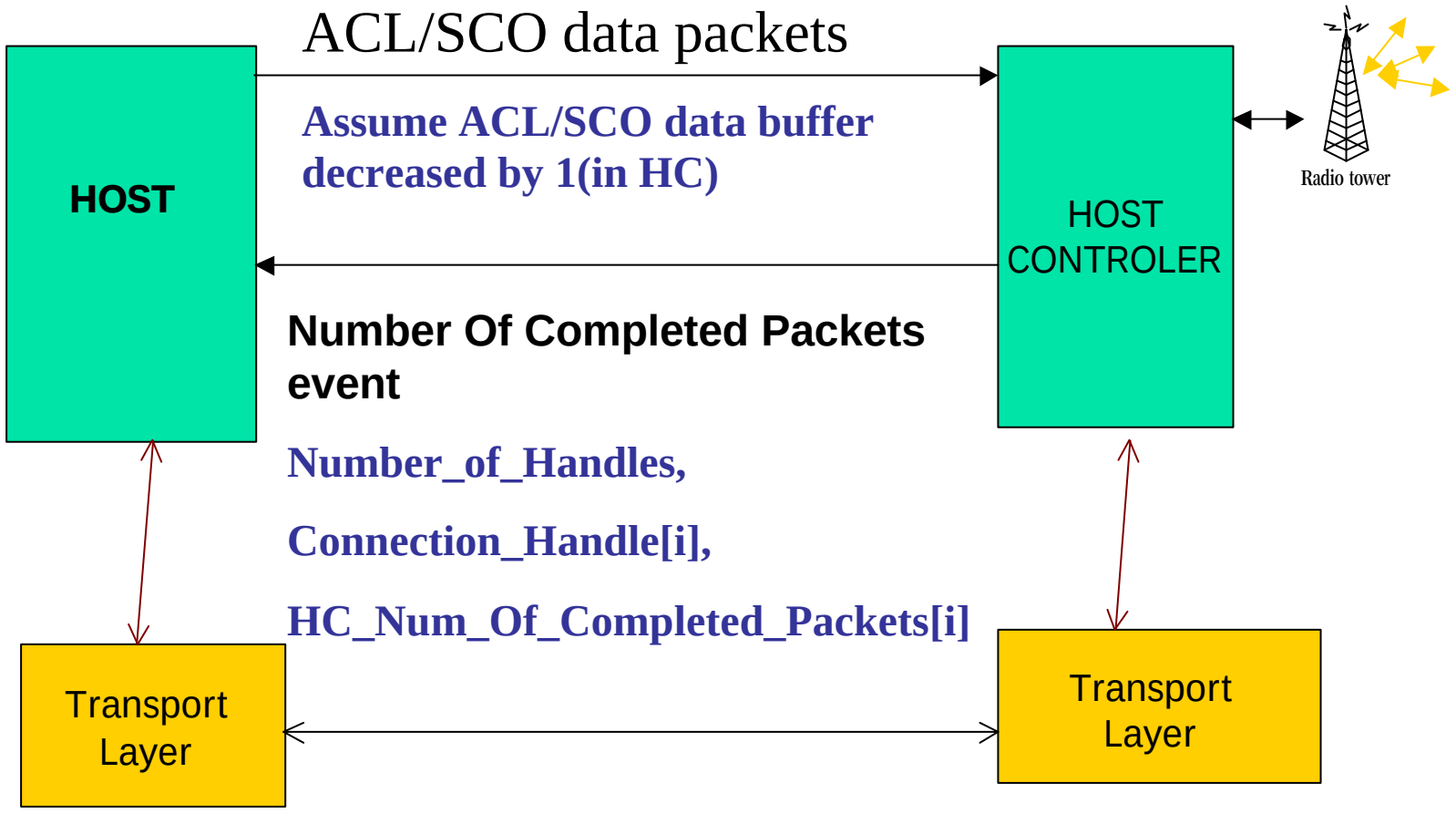
# HCI FLOW CONTROL

- Avoiding filling up the data buffers
- Host manages the data buffers of the Host Controller
- Host Controller manages the data buffers of the Host

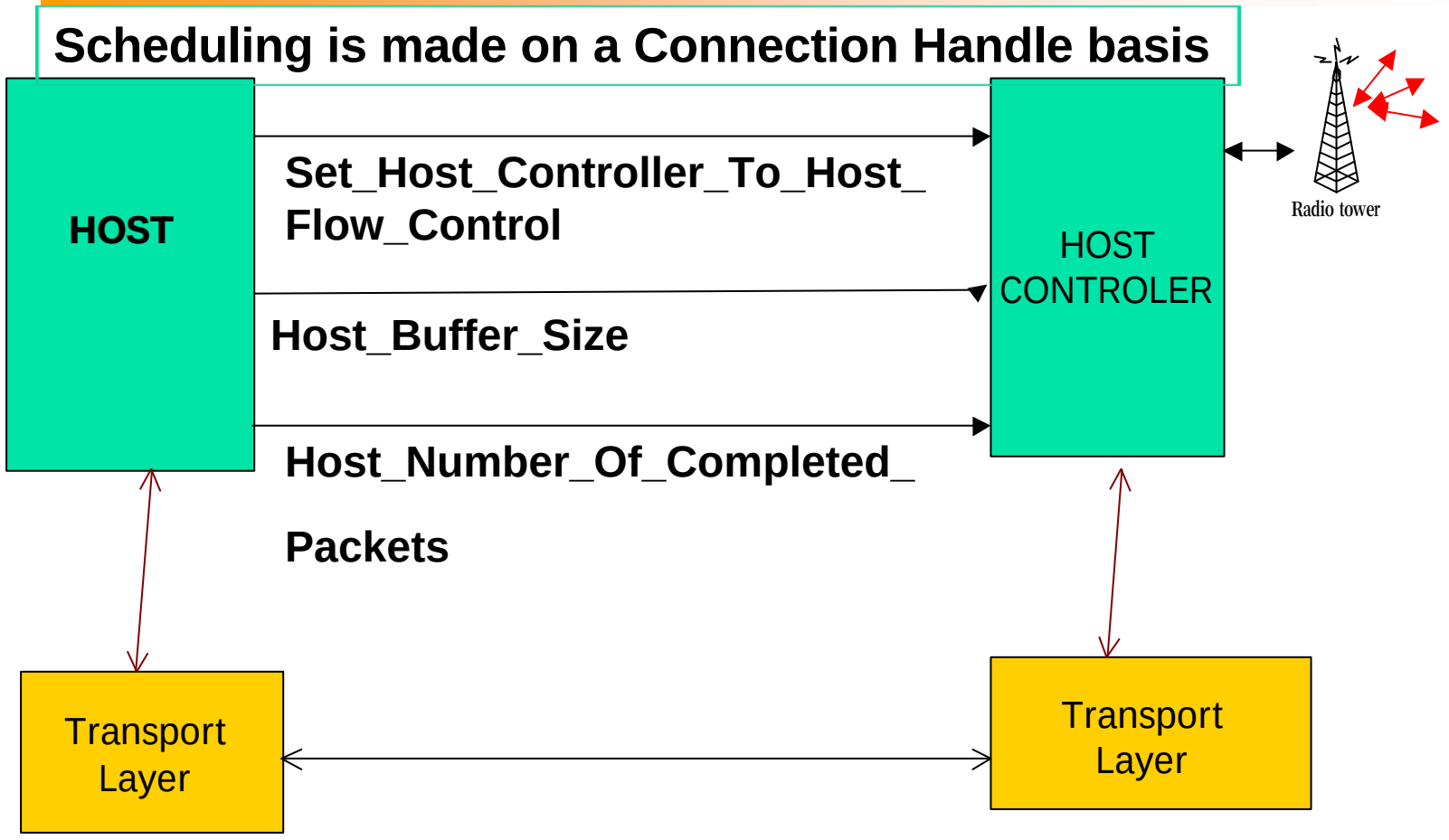
# HCI FLOW CONTROL (cont.) (initialization)



# HCI FLOW CONTROL (cont.) (HC controlling flow of data)



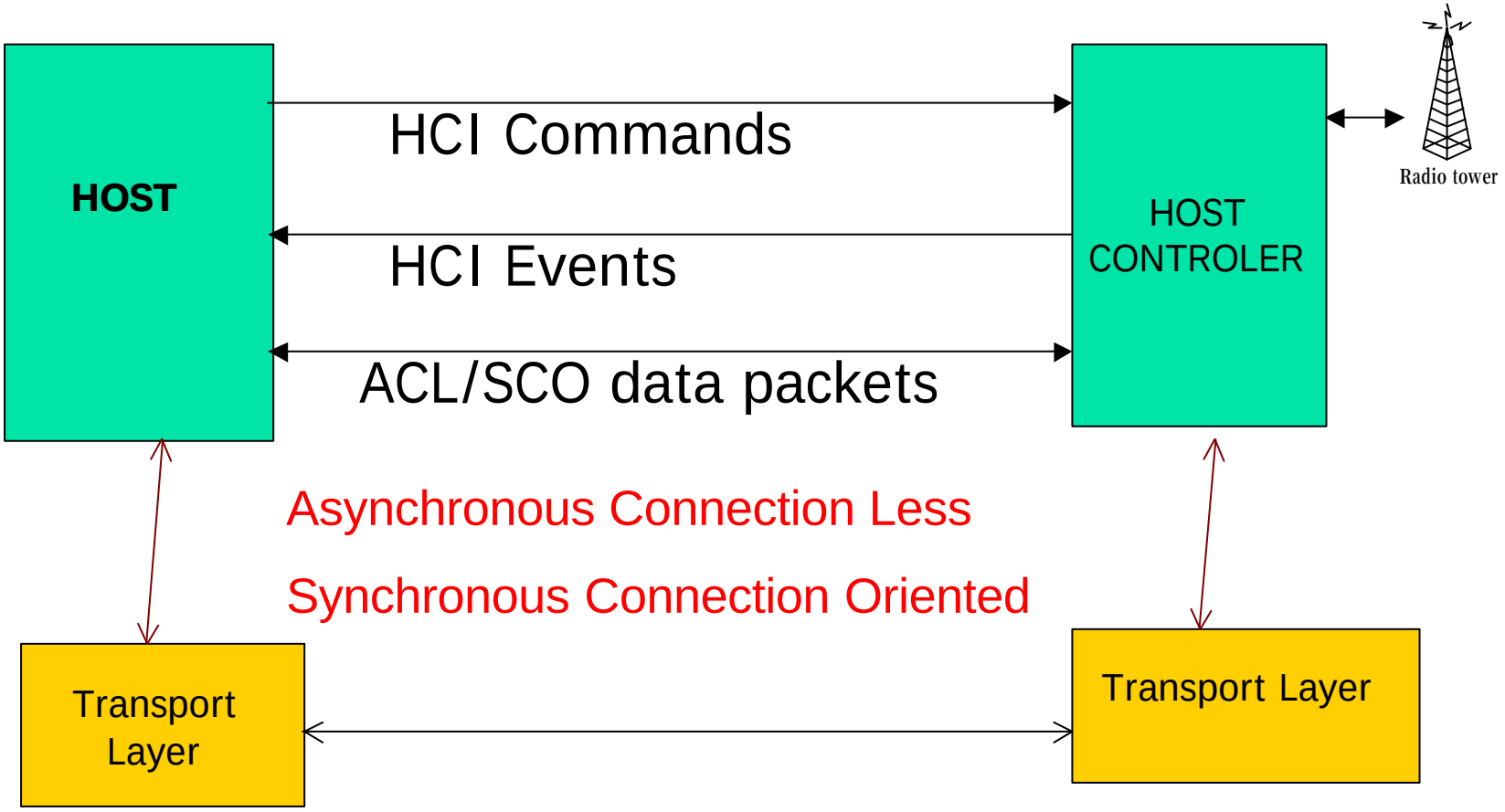
# HCI FLOW CONTROL (cont.) (host controlling the flow)



# OUTLINE

- Introduction
- HCI flow control
- HCI packets
- HCI commands
- HCI events
- HCI data packets
- Message sequence chart

# HCI PACKETS


HOST
HCI Commands
HCI Events
ACL/SCO data packets
HOST CONTROLER
Radio tower
Asynchronous Connection Less
Synchronous Connection Oriented
Transport Layer
Transport Layer

# HCI PACKETS

- **Command packets used by the host to control the module**
- **Event packets used by the module to inform the host**
- **Data packets to pass voice and data between host and module**

# OUTLINE

- Introduction
- HCI flow control
- HCI packets
- HCI commands
- HCI events
- HCI data packets
- Message sequence chart

# HCI COMMANDS

p.553 Fig.4.1

OCF: OpCode Command Field

OGF: OpCode Group Field

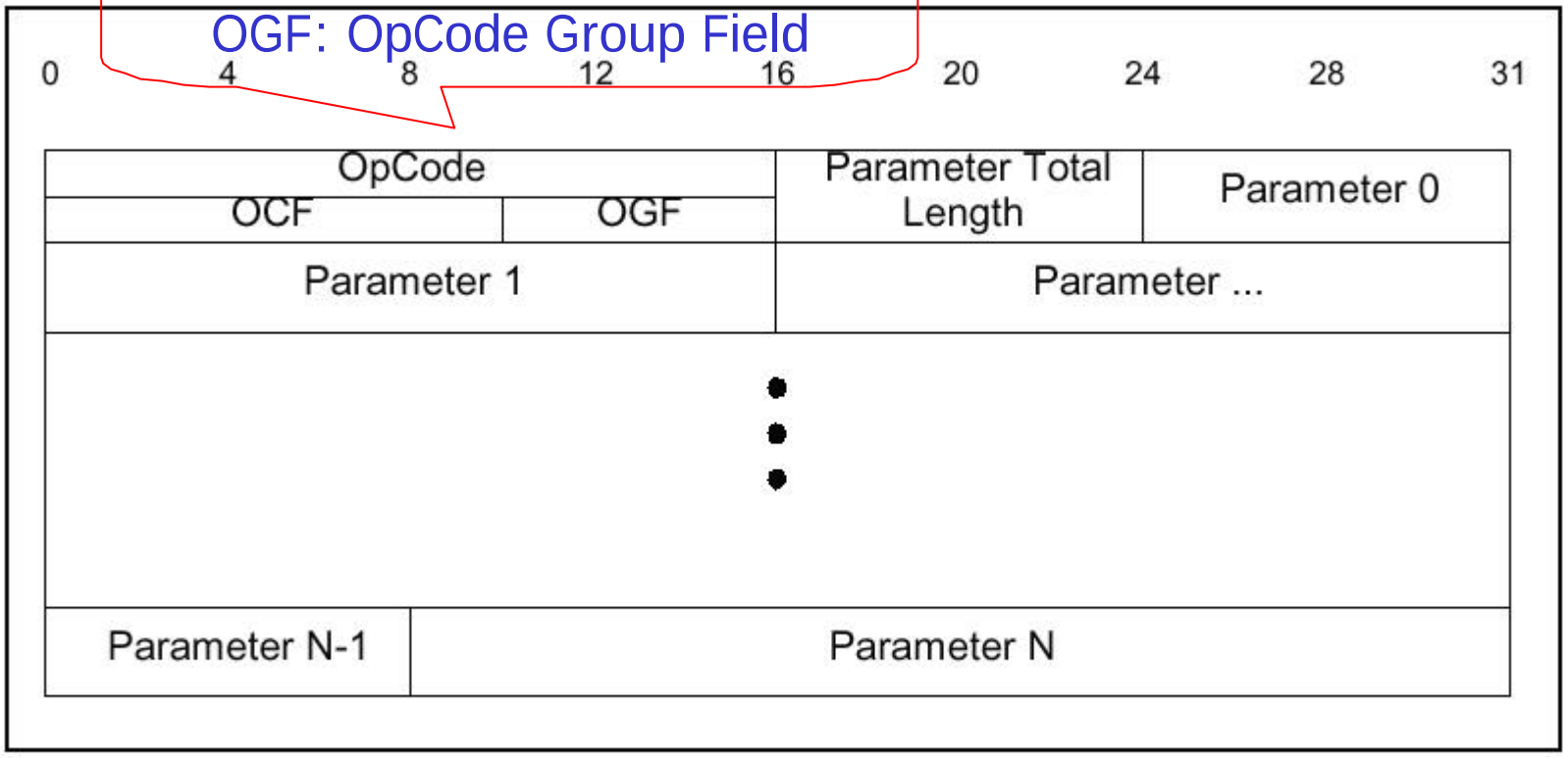



*Figure 4.1: HCI Command Packet*

# HCI COMMANDS (cont.)

*Op Code:* *Size: 2 Bytes*

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

*Parameter Total Length:* *Size: 1 Byte*

| Value | Parameter Description |
|---|---|
| 0xXX | Lengths of all of the parameters contained in this packet measured in bytes. (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 bytes in size. |

# HCI COMMANDS (cont.)

- **HCI command category**
  - Link control commands (22)
  - Link policy commands (10)
  - Host controller & baseband commands (51)
  - Information parameters (5)
  - Status parameters (4)
  - Testing commands (3)

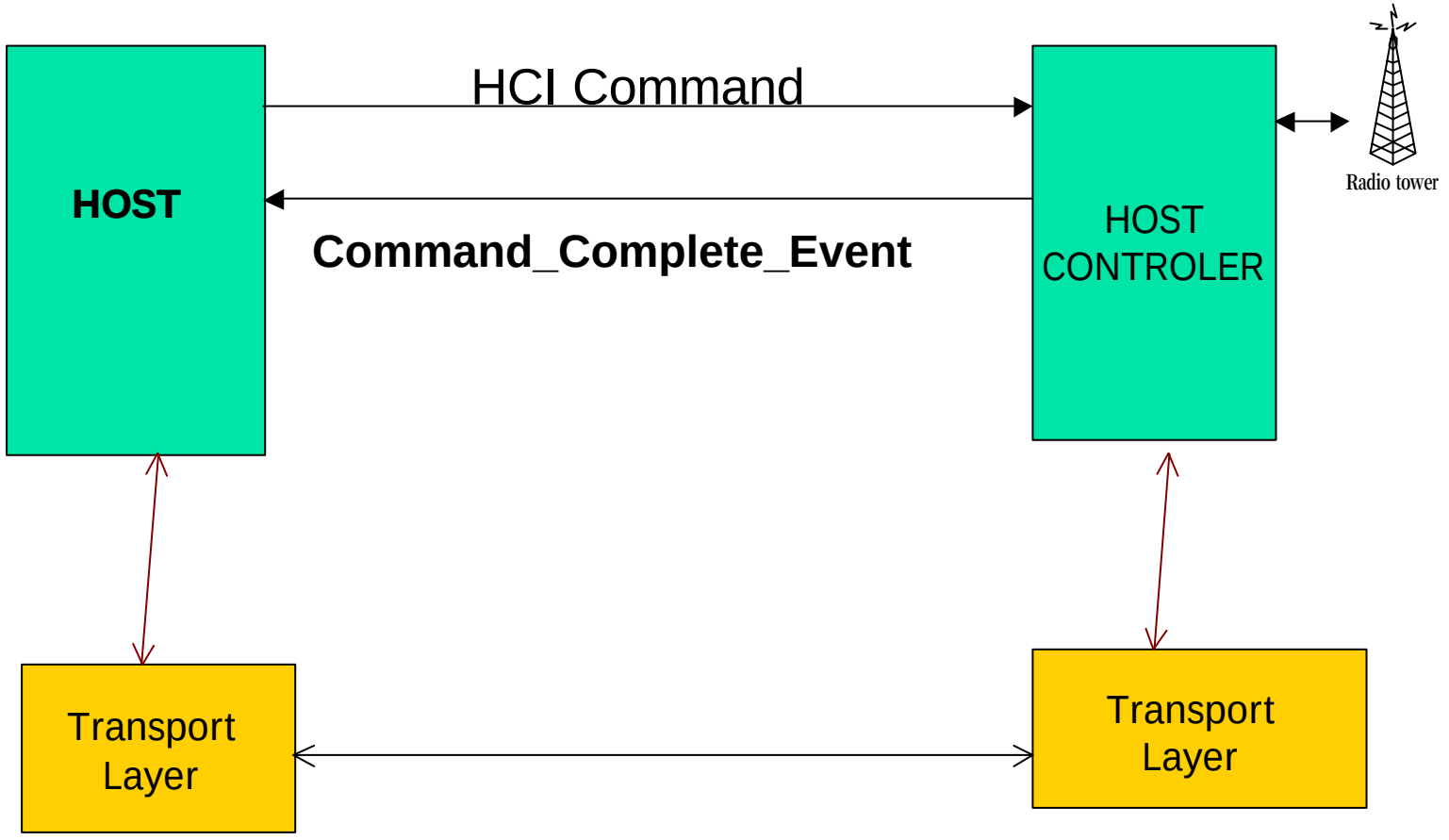
# HCI COMMANDS (cont.)

- **Command sent to the module → the module responds with**
  - **HCI_Command_Complete event**
  - **HCI_command_Status event**

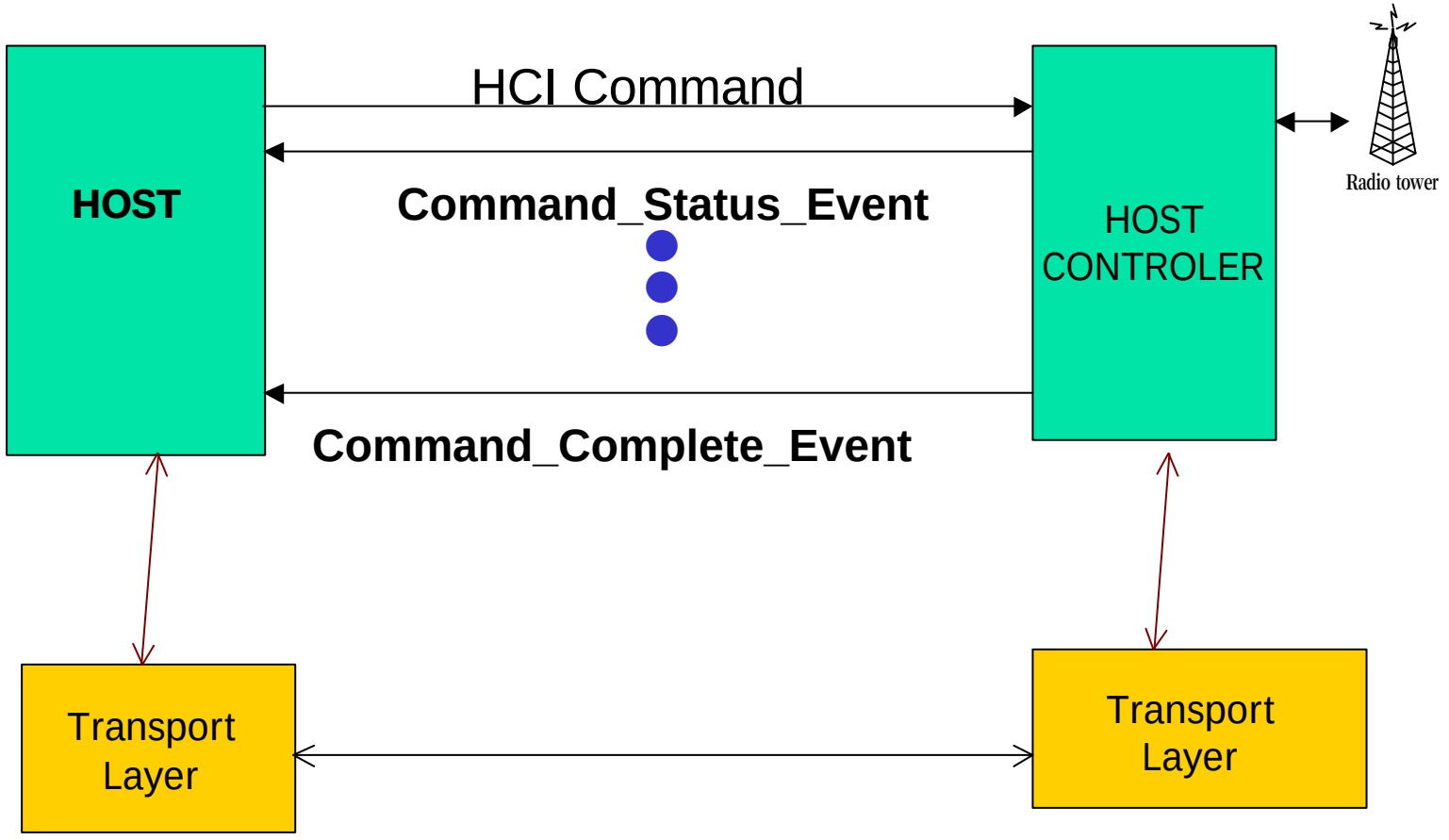
status returned first, then complete event returned when the command has completed

if the command can be executed immediately

# HCI COMMANDS (cont.)



# HCI COMMANDS (cont.)



# HCI COMMAND CATEGORY

- Link control commands (22)
- Link policy commands (10)
- Host controller & baseband commands (51)
- Information parameters (5)
- Status parameters (4)
- Testing commands (3)

# LINK CONTROL COMMANDS

- **Allowing the host controller to control connections to other Bluetooth devices**
- **Total 22 commands**
- **Three types of commands**
  - Performing inquiries of other BD in range (4)
  - Instructing the LM to create and modify link layer connections with remote devices (5)
  - Other LMP commands (13)

OGF: 0x01

# LINK CONTROL COMMANDS (INQUIRY COMMANDS)

Write_Scan_Enable

| | |
|---|---|
| Inquiry | The Inquiry command will cause the Bluetooth radio to enter Inquiry Mode. Inquiry Mode is used to discovery other nearby Bluetooth radios. |
| Inquiry_Cancel | The Inquiry_Cancel command will cause the Bluetooth radio to stop the current Inquiry if the Bluetooth radio is in Inquiry Mode. |
| Periodic_Inquiry_Mode | The Periodic_Inquiry_Mode command is used to configure the Bluetooth radio to perform an automatic Inquiry based on a specified period range. |
| Exit_Periodic_Inquiry_Mode | The Exit_Periodic_Inquiry_Mode command is used to end the Periodic Inquiry mode when the local device is in Periodic Inquiry Mode |

# LINK CONTROL COMMANDS (CONNECTION COMMANDS)

parameters:
BD_ADDR, Packet_Type,
Page_Can_Repetition_Mode,
Page_Scan_mode, Clock_Offset,
Allow_Role_Switch

| | |
|---|---|
| Create_Connection | ll cause the ction to the pecified by the command parameters. |
| Disconnect | The Disconnect command is used to terminate an existing connection |
| Add_SCO_Connection | The Add_SCO_Connection command will cause the link manager to create a SCO connection using the ACL connection specified by the Connection Handle command parameter. |
| Accept_Connection_Request | The Accept_Connection_Request command is used to accept a new incoming connection request. |
| Reject_Connection_Request | The Reject_Connection_Request command is used to decline a new incoming connection request. |

# LINK CONTROL COMMANDS (OTHER COMMANDS)

| | |
|---|---|
| Link_Key_Request_Negative_Reply | The Link_Key_Request_Negative_Reply command is used to reply to a Link Key Request Event from the Host 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_Reply | The PIN_Code_Request_Reply command is used to reply to a PIN Code request Event from the Host Controller and specifies the PIN code to use for a connection |
| PIN_Code_Request_Negative_Reply | The PIN_Code_Request_Negative_Reply command is used to reply to a PIN Code request Event from the Host Controller when the Host cannot specify a PIN code to use for a connection. |
| Change_Connection_Packet_Type | The Change_Connection_Packet_Type command is used to change which packet types can be used for a connection that is currently established. |
| Authentication_Requested | The Authentication_Requested command is used to establish authentication between the two devices associated with the specified Connection Handle. |
| Set_Connection_Encryption | The Set_Connection_Encryption command is used to enable and disable the link level encryption. |
| Change_Connection_Link_Key | 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. |
| Link_Key_Request_Reply | The Link_Key_Request_Reply command is used to reply to a Link Key Request event from the Host 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. |
| Master_Link_Key | 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. |
| Remote_Name_Request | The Remote_Name_Request command is used for obtaining the user-friendly name of another Bluetooth device. |
| Read_Remote_Supported_Features | The Read_Remote_Supported_Features command requests a list of the supported features of a remote device. |
| Read_Remote_Version_Information | The Read_Remote_Version_Information command will read the values for the version information for the remote Bluetooth device. |
| Read_Clock_Offset | The Read_Clock_Offset command allows the host to read clock offset of remote devices |

# HCI COMMAND CATEGORY

- Link control commands (22)
- Link policy commands (10)
- Host controller & baseband commands (51)
- Information parameters (5)
- Status parameters (4)
- Testing commands (3)

to provide methods for the Host to affect how the Link Manager manages the piconet

OGF: 0x02

# LINK POLICY COMMANDS

Power Saving

| | |
|---|---|
| Hold_Mode | The Hold_Mode command is used to alter the behavior of the LM and have the LM place the local or remote device into the hold mode. |
| Sniff_Mode | 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 | The Exit_Sniff_Mode command is used to end the sniff mode for a connection handle, which is current in sniff mode. |
| Park_Mode | The Park_Mode command is used to alter the behavior of the LM and have the LM place the local or remote device into the Park mode. |
| Exit_Park_Mode | The Exit_Park_Mode command is used to switch the Bluetooth device from park mode back to active mode. |

Quality of Service

| | |
|---|---|
| QoS_Setup | The QoS_Setup command is used to specify Quality of Service parameters for a connection handle. |
| Role_Discovery | 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 | The Switch_Role command is used for a Bluetooth device switch the current role the device is performing for a particular connection with the specified Bluetooth device |
| Read_Link_Policy_Settings | The Read_Link_Policy_Settings will read the Link Policy settings 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 | The Write_Link_Policy_Settings will write the Link Policy settings 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. |

# HCI COMMAND CATEGORY

- Link control commands (22)
- Link policy commands (10)
- Host controller & baseband commands (51)
- Information parameters (5)
- Status parameters (4)
- Testing commands (3)

# HOST CONTROLLER AND BASEBAND COMMANDS

- **Providing control of Bluetooth device and the capabilities of the**
  - Host controller
  - Link manager
  - Baseband
- **The host device can use these commands to modify the behavior of the local device**

OGF: 0x03

# HOST CONTROLLER AND BASEBAND COMMANDS

| | |
|---|---|
| Set_Event_Mask | The Set_Event_Mask command is used to control which events are generated by the HCI for the host. |
| Reset | The Reset command will reset the Bluetooth Host Controller, Link Manager, and the radio module. |
| Set_Event_Filter | The Set_Event_Filter command is used by the Host to specify different event filters. The Host may issues this command multiple times to request various conditions for the same type of Event Filter and for different types of Event Filters. |
| Flush | The Flush command is used to discard all data that is currently pending for transmission in the Host Controller for the specified connection handle even if there currently are chunks of data that belong to more than one L2CAP packet in the Host Controller. |

| | |
|---|---|
| Read_PIN_Type | The Read_PIN_Type command is used for the Host to read the value that is specified to indicate if the Host supports variable PIN or if the Host only supports fixed PINs |
| Write_PIN_Type | The Write_PIN_Type command is used for the Host to specific if the Host supports variable PIN or if the Host only supports fixed PINs |
| Create_New_Unit_Key | The Create_New_Unit_Key command is used to create a new unit key. |
| Read_Stored_Link_Key | The Read_Stored_Link_Key command provides the ability to read one or more link keys stored in the Bluetooth Host Controller. |
| Write_Stored_Link_Key | The Write_Stored_Link_Key command provides the ability to write one or more link keys to be stored in the Bluetooth Host Controller. |

# HCI COMMAND CATEGORY

- Link control commands (22)
- Link policy commands (10)
- Host controller & baseband commands (51)
- Information parameters (5)
- Status parameters (4)
- Testing commands (3)

OGF: 0x04

fixed by the manufacturer of the Bluetooth hardware

# INFORMATIONAL PARAMETER COMMANDS

| | |
|---|---|
| Read_Local_Version_Information | The Read_Local_Version_Information command will read the values for the version information for the local Bluetooth device. |
| Read_Local_Supported_Features | The Read_Local_Supported_Features command requests a list of the supported features for the local device. |
| Read_Buffer_Size | The Read_Buffer_Size command returns the size of the HCI buffers. These buffers are used by the Host Controller to buffer data that is to be transmit-ted. |
| Read_Country_Code | The Read_Country_Code command will read the value for the Country Code status parameter. The Country Code defines which range of frequency band of the ISM 2.4 GHz band will be used by the radio. |
| Read_BD_ADDR | The Read_BD_ADDR command will read the value for the BD_ADDR parameter. The BD_ADDR is a 48-bit unique identifier for a Bluetooth radio. |

# HCI COMMAND CATEGORY

- Link control commands (22)
- Link policy commands (10)
- Host controller & baseband commands (51)
- Information parameters (5)
- Status parameters (4)
- Testing commands (3)

provide information about the current state of the Host Controller, Link Manager, and Baseband

OGF: 0x05

# STATUS PARAMETER COMMANDS

| | |
|---|---|
| Read_Failed_Contact_Counter | The Read_Failed_Contact_Counter will read the value for the Failed Contact Counter parameter for a particular connection to another device. The Failed Contact Counter records the number of consecutive incidences 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". |
| Reset_Failed_Contact_Counter | The Reset_Failed_Contact_Counter will reset the value for the Failed Contact Counter parameter for a particular connection to another device. The Failed Contact Counter records the number of consecutive incidences 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". |
| Get_Link_Quality | The Get_Link_Quality command will read the value for the Link Quality for the specified Connection Handle. |
| Read_RSSI | The Read_RSSI command will read the value for the Received Signal Strength Indication (RSSI) for a connection handle to another Bluetooth device. |

# HCI COMMAND CATEGORY

- Link control commands (22)
- Link policy commands (10)
- Host controller & baseband commands (51)
- Information parameters (5)
- Status parameters (4)
- Testing commands (3)

to provide the ability to test various functionalities of the Bluetooth hardware

OGF: 0x06

# TESTING COMMANDS

| | |
|---|---|
| Read_Loopback_Mode | The Read_Loopback_Mode will read the value for the setting of the Host Controllers Loopback Mode. The setting of the Loopback Mode will determine the path of information. |
| Write_Loopback_Mode | The Write_Loopback_Mode will write the value for the setting of the Host Controllers Loopback Mode. The setting of the Loopback Mode will determine the path of information. |
| Enable Device Under Test Mode | The Enable Device Under Test Mode command will cause the local Bluetooth module to enter into test mode. The host would issues this command when the host want local device to be the DUT for the Testing scenarios as described in the Bluetooth Test Mode document. |

# OUTLINE

- Introduction
- HCI flow control
- HCI packets
- HCI commands
- HCI events
- HCI data packets
- Message sequence chart

# HCI EVENT PACKETS

p.557 Fig.4.2

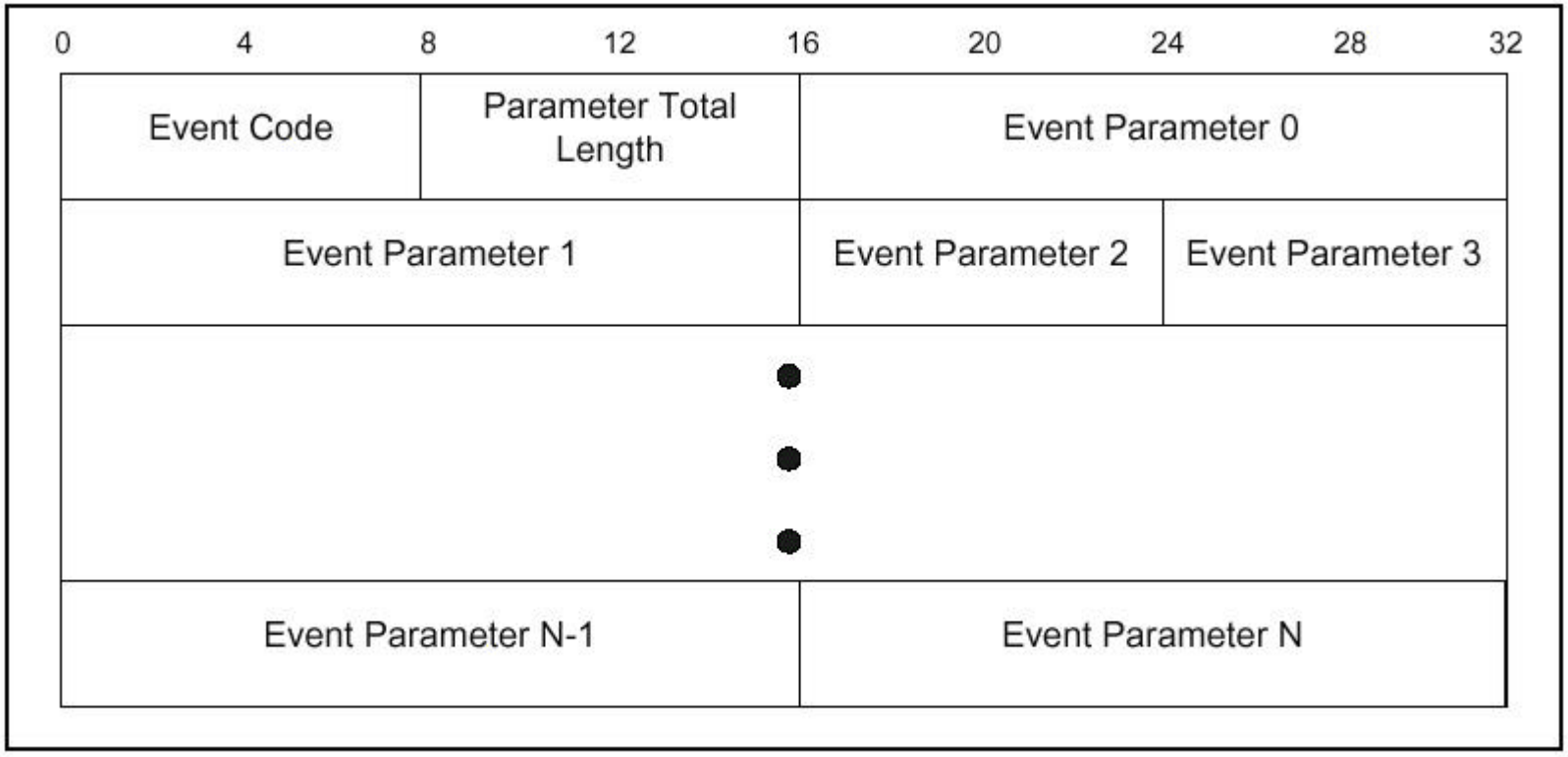



*Figure 4.2: HCI Event Packet*

# HCI EVENT PACKET (cont.)


*Event Code:* *Size: 1 Byte*

| Value | Parameter Description |
|---|---|
| 0xXX | Each Event is assigned a 1 Byte Event Code used to uniquely identify different types of events.<br>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 Byte*

| Value | Parameter Description |
|---|---|
| 0xXX | Length of all of the parameters contained in this packet measured in bytes |

*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 bytes in size. |

# OUTLINE

- Introduction
- HCI flow control
- HCI packets
- HCI commands
- HCI events
- HCI data packets
- Message sequence chart

# HCI DATA PACKETS


- **Exchanging data between the host and host controller**
- **Two data types**
  - **ACL data type** (Asynchronous Connection Less)
  - **SCO data type**(Synchronous Connection Oriented)

# HCI ACL DATA PACKET

p.558 Fig.4.3


*Figure 4.3: HCI ACL Data Packet*

# HCI ACL DATA PACKET (cont.)

*Connection_Handle:* *Size: 12 Bits*

| Value | Parameter Description |
|---|---|
| 0xXXX | Connection Handle to be used for transmitting a data packet or segment.<br>Range: 0x0000-0x0EFF (0x0F00 - 0x0FFF Reserved for future use)<br>Note: If the Broadcast_Flag is set to 01 or 10 for an HCI Data Packet sent from the Host to the Host Controller, the value of the Connection_Handle parameter is ignored by the Host Controller.<br>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 the connection handle for the ACL connection to the master that sent the broadcast. |

From Spec. 1.0a for short

# HCI ACL DATA PACKET (cont.)

*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 a L2CAP packet) |
| 11 | Reserved for Future Use |

*Data_Total_Length:* *Size: 2 Bytes*

| Value | Parameter Description |
|---|---|
| 0xXXXX | Length of data measured in bytes. |

# HCI ACL DATA PACKET (cont.)

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

| Value | Parameter Description |
|---|---|
| 00 | No broadcast. Only point-to-point. |
| 01 | Active 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 or park mode. See note below! |
| 10 | Piconet Broadcast: packet is sent to all slaves and all slaves in park mode (i.e. packet is sent during park beacon slots if there are parked slaves), and it may be received by slaves in sniff mode. See note below! |
| 11 | Reserved for future use. |

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

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

Different with Spec. 1.0a

# HCI SCO DATA PACKET

p.561 Fig.4.4

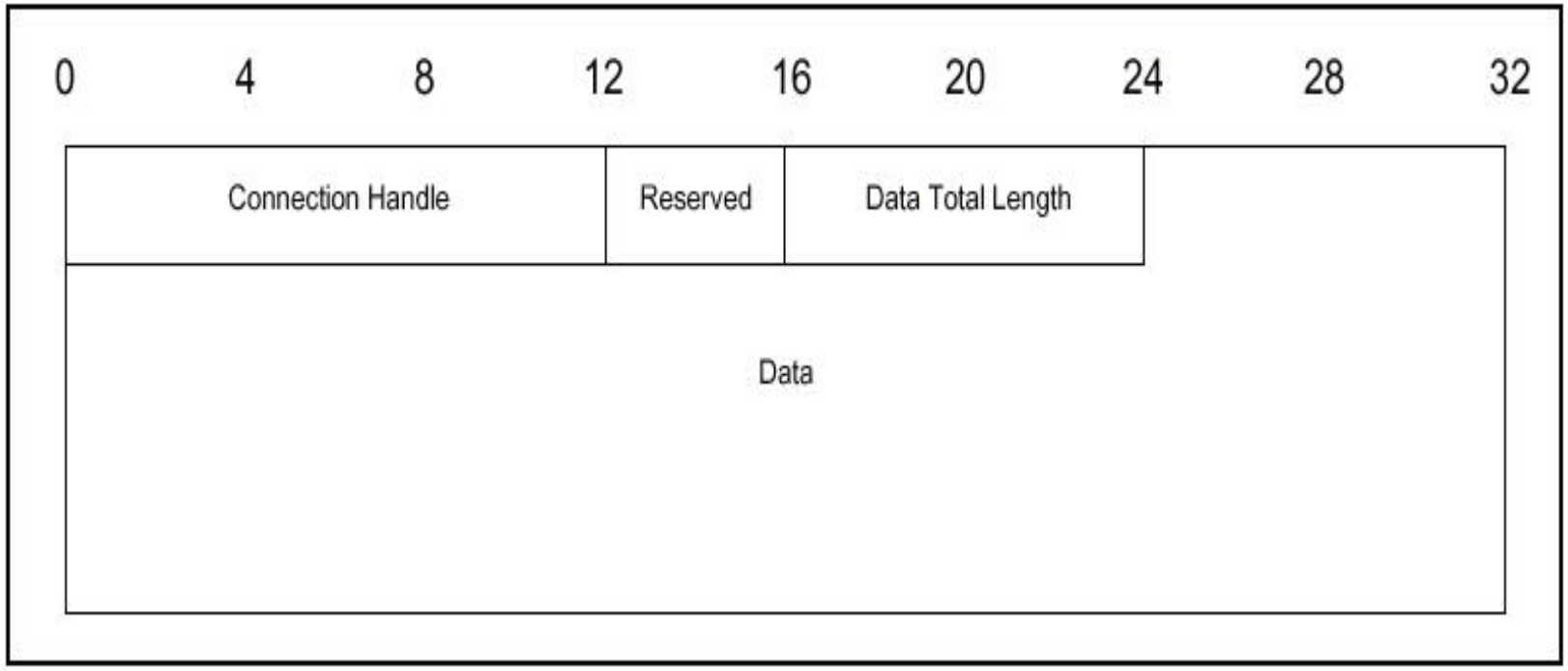



*Figure 4.4: HCI SCO Data Packet*

# HCI SCO DATA PACKET (cont.)

*Connection_Handle:* *Size: 12 Bits*

| Value | Parameter Description |
|---|---|
| 0xXXX | Connection Handle to be used to for transmitting a SCO data packet or segment.<br>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 byte of the HCI SCO Data packet.*

*Reserved:* *Size: 4 Bits*

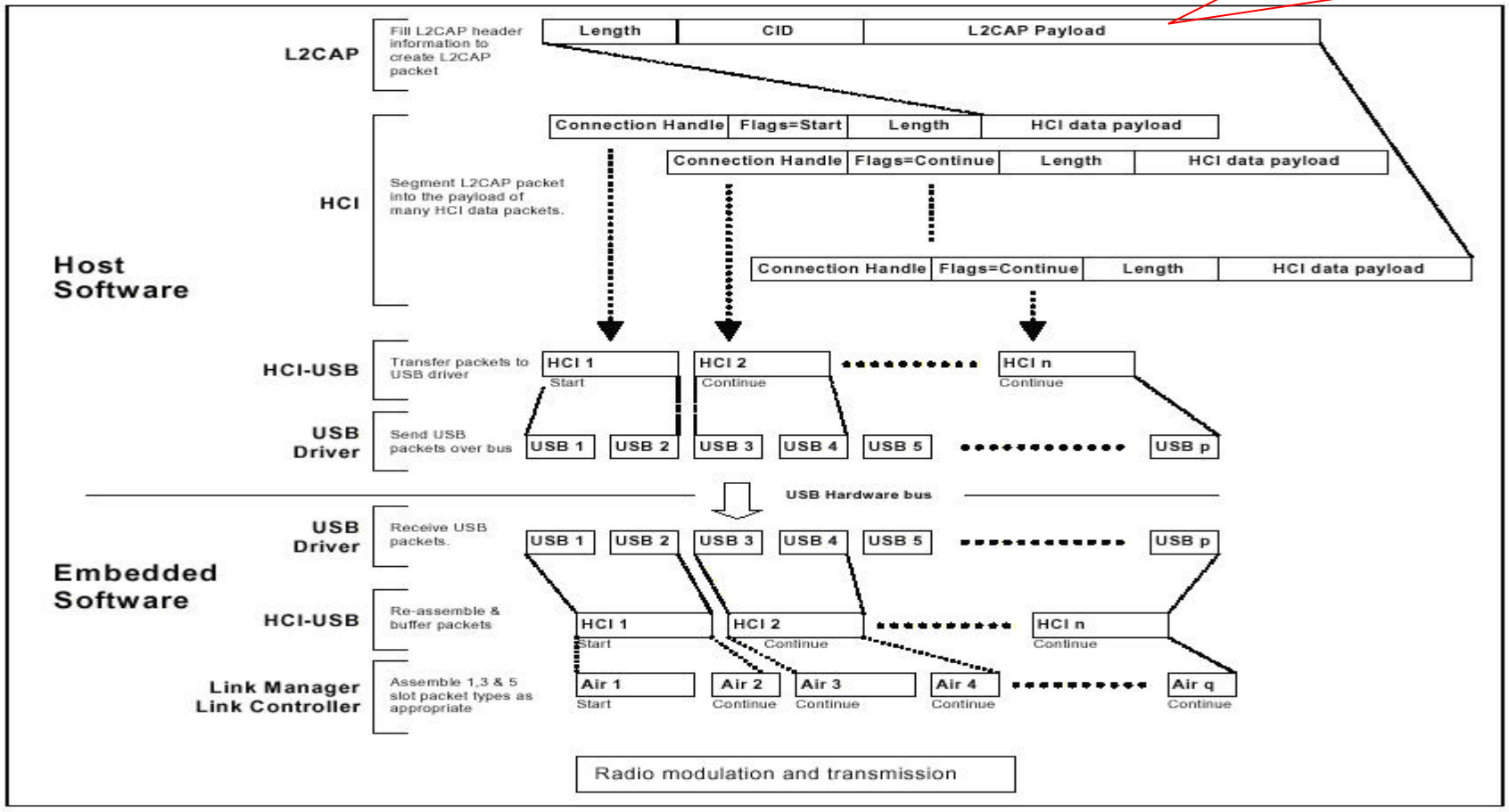
| Value | Parameter Description |
|---|---|
| XXXX | Reserved for future use |

*Data Total Length:* *Size: 1 Byte*

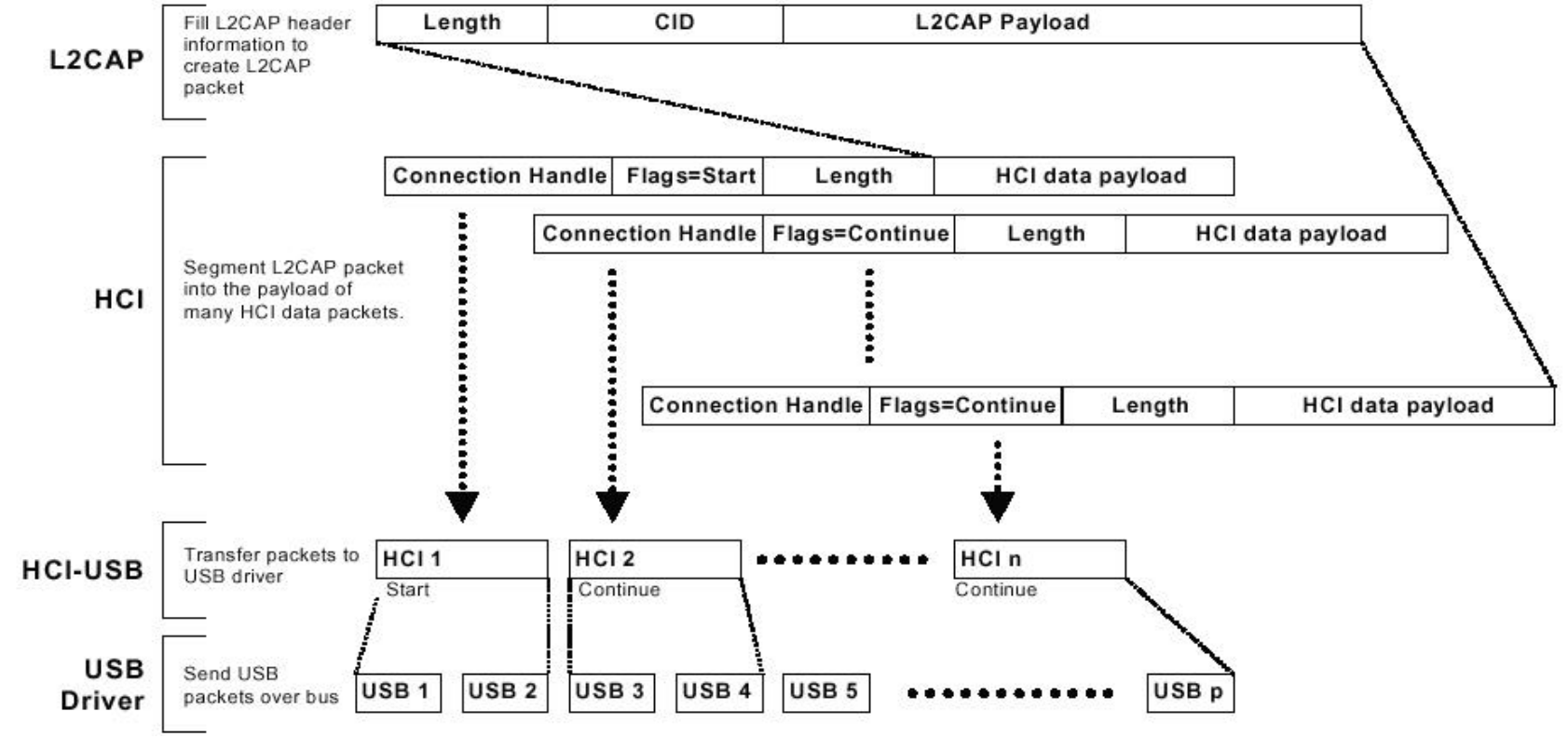
| Value | Parameter Description |
|---|---|
| 0xXX | Length of SCO data measured in bytes |

# SEGMENTATION AND REASSEMBLY OPERATIONS

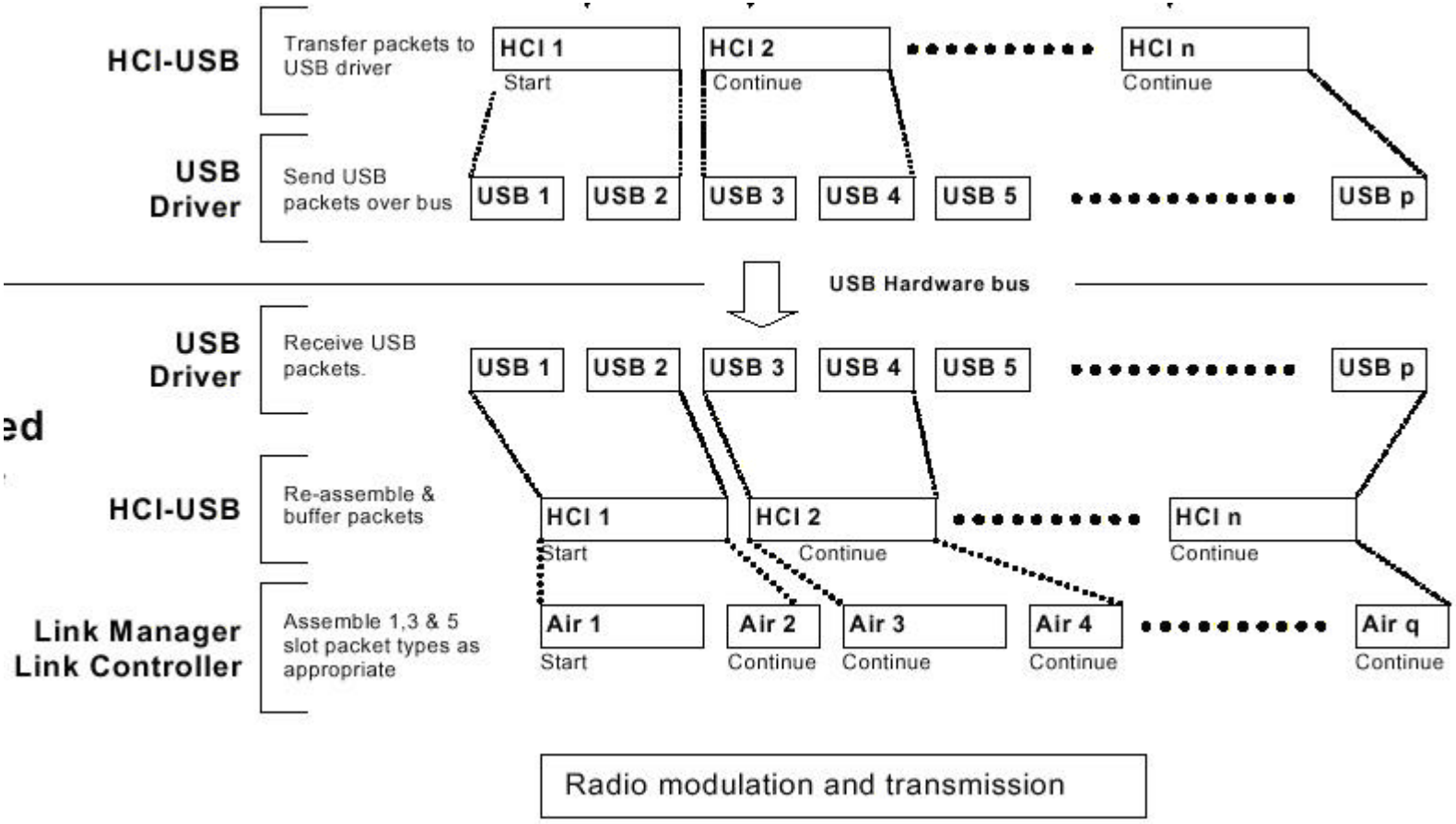
p.265 Fig.2.5



# SEGMENTATION & REASSEM. OPERATIONS (cont.)



# SEGMENTATION & REASSEM. OPERATIONS (cont.)

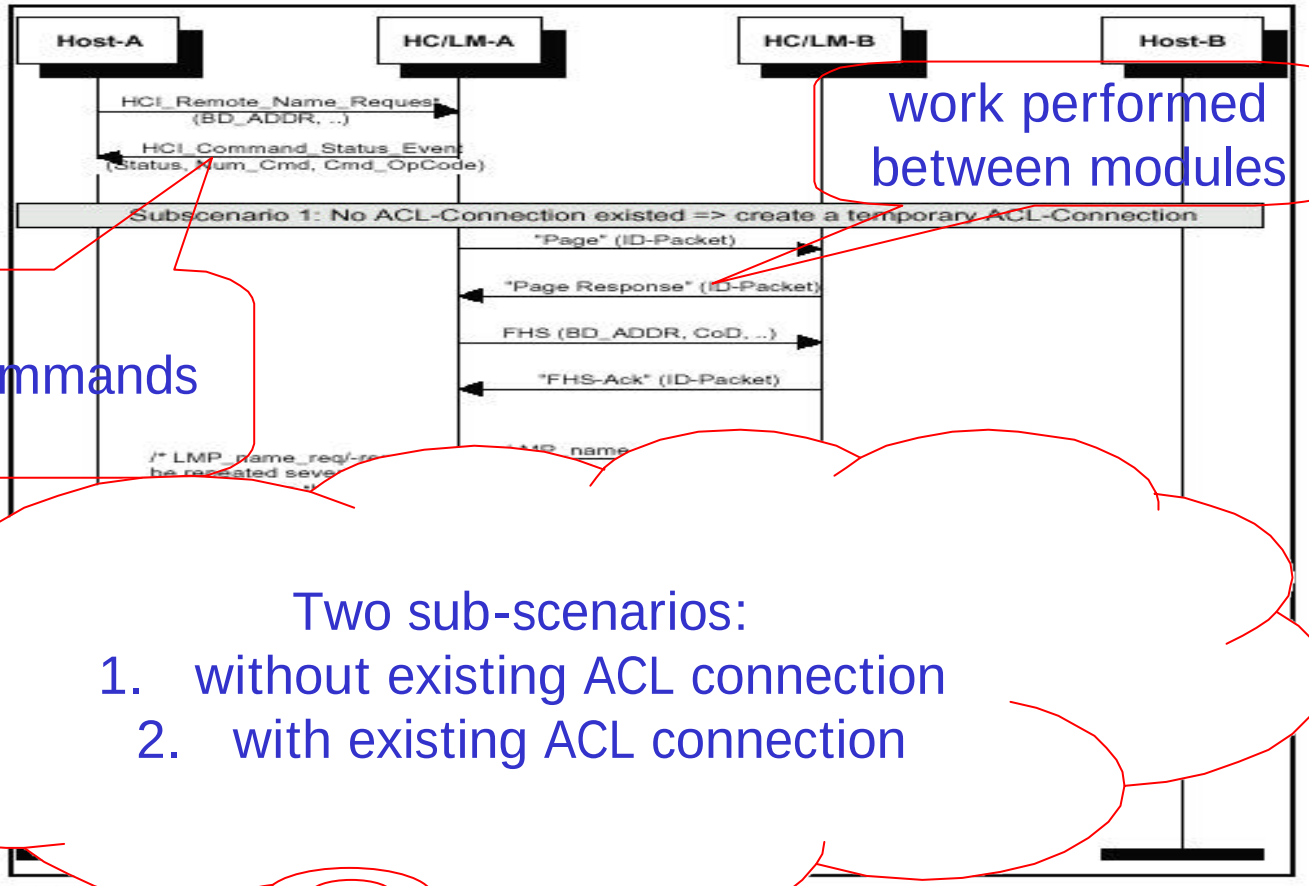


# OUTLINE

- Introduction
- HCI flow control
- HCI packets
- HCI commands
- HCI events
- HCI data packets
- Message sequence chart

Appendix IX p.1035

# MESSAGE SEQUENCE CHART



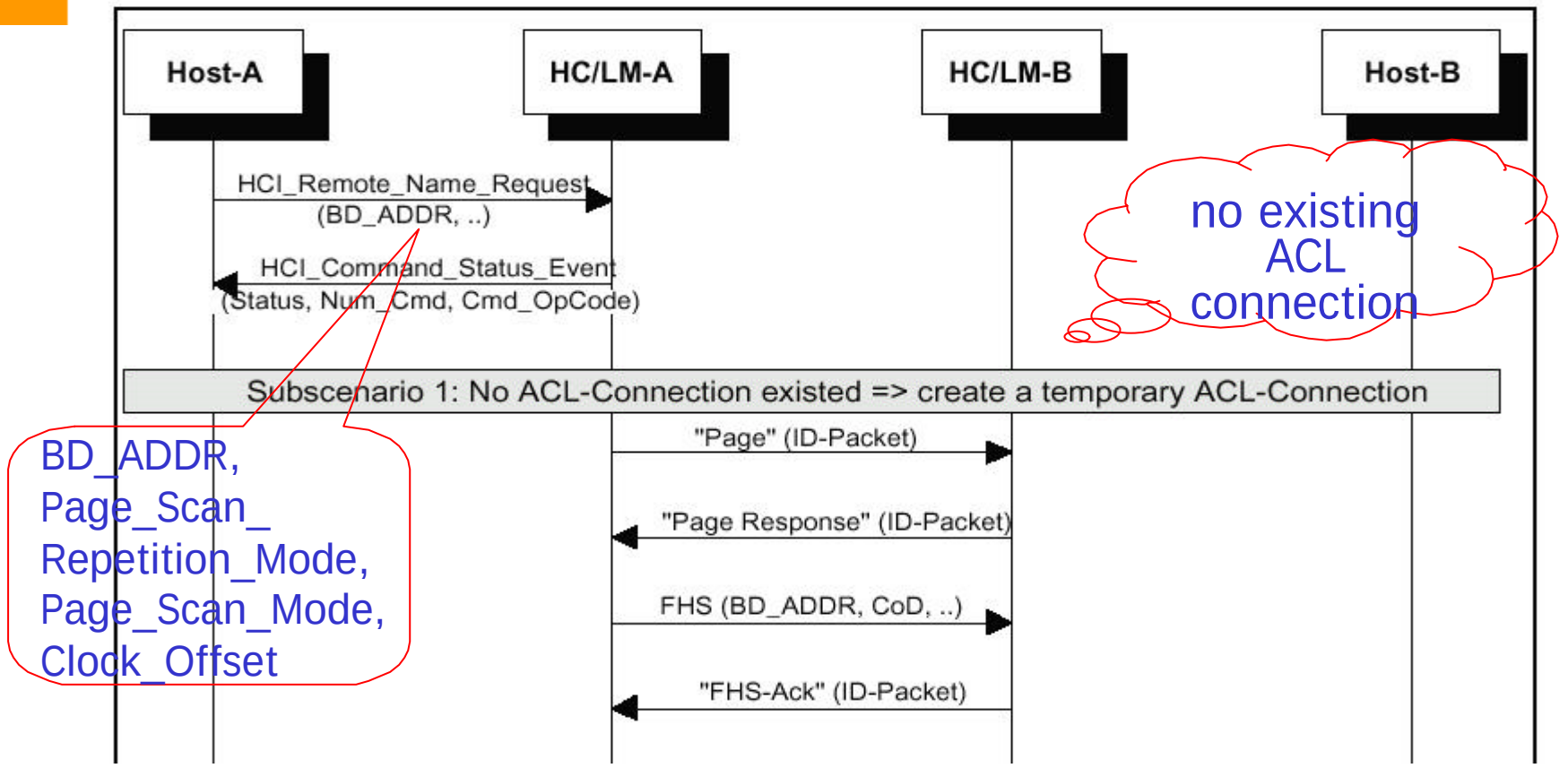

work performed between modules

HCI Commands

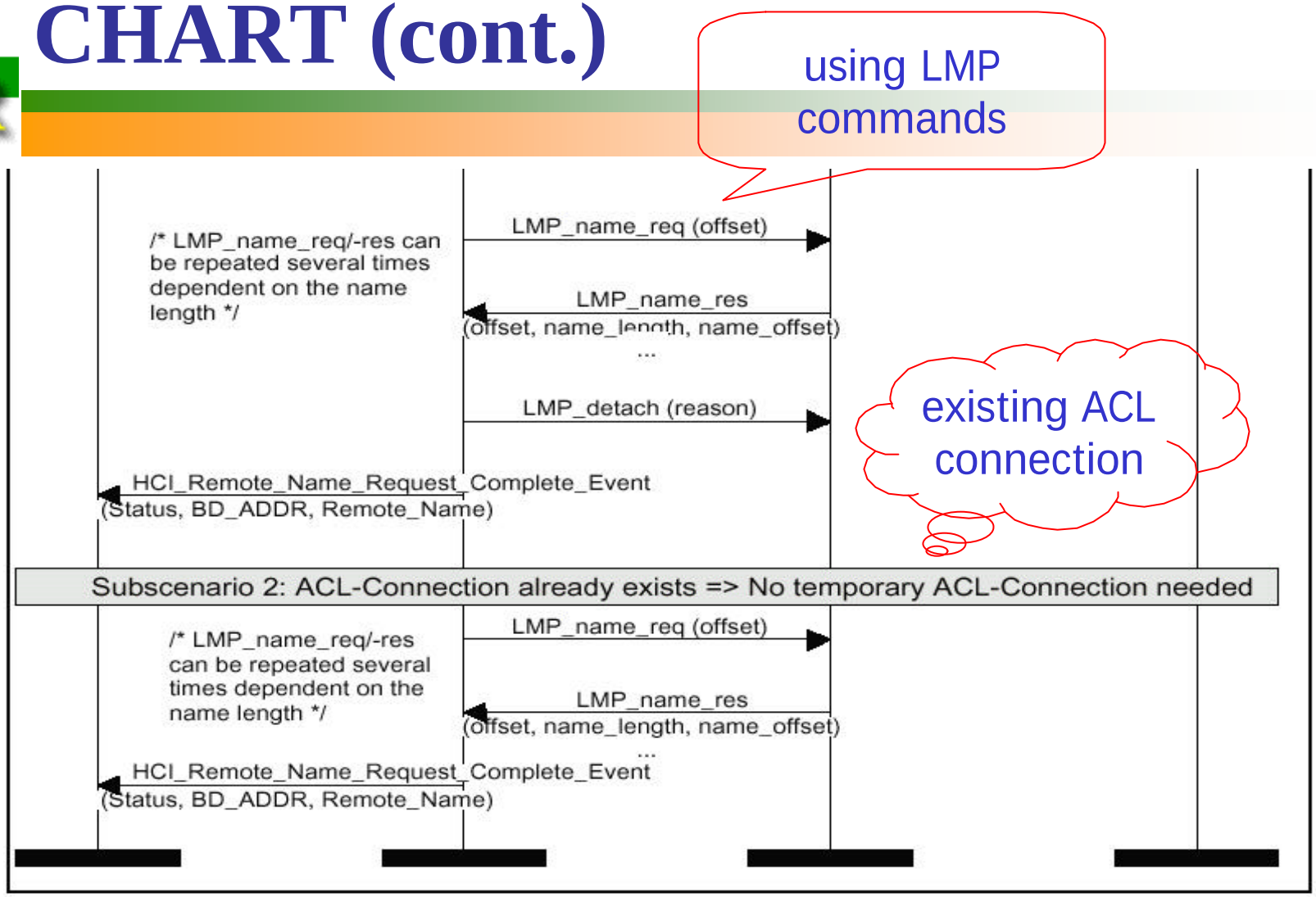
Two sub-scenarios:
1. without existing ACL connection
2. with existing ACL connection

Fig.2.1 Remote Name Request

# MESSAGE SEQUENCE CHART (cont.)



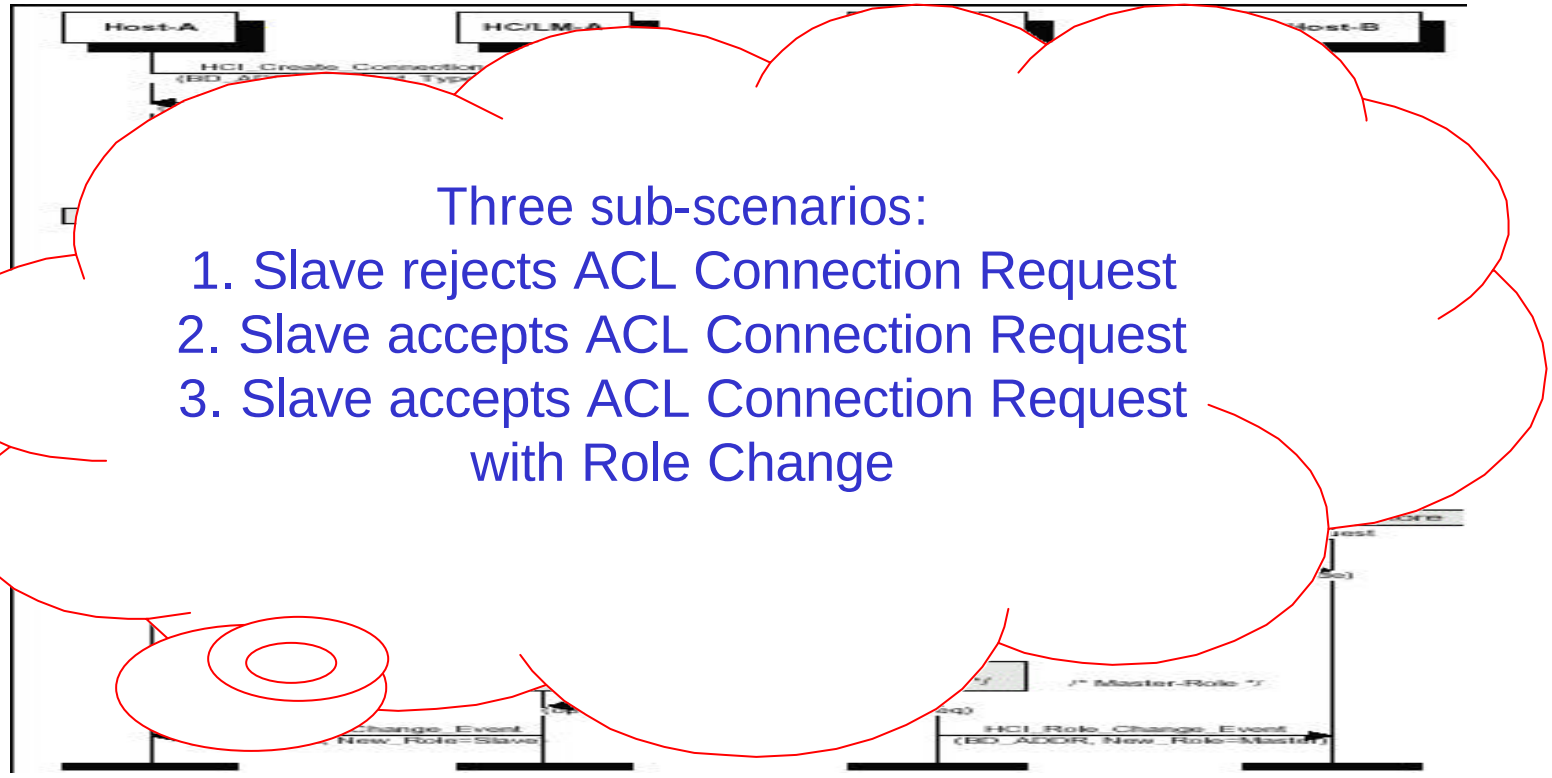
# MESSAGE SEQUENCE CHART (cont.)



# MESSAGE SEQUENCE CHART (cont.)

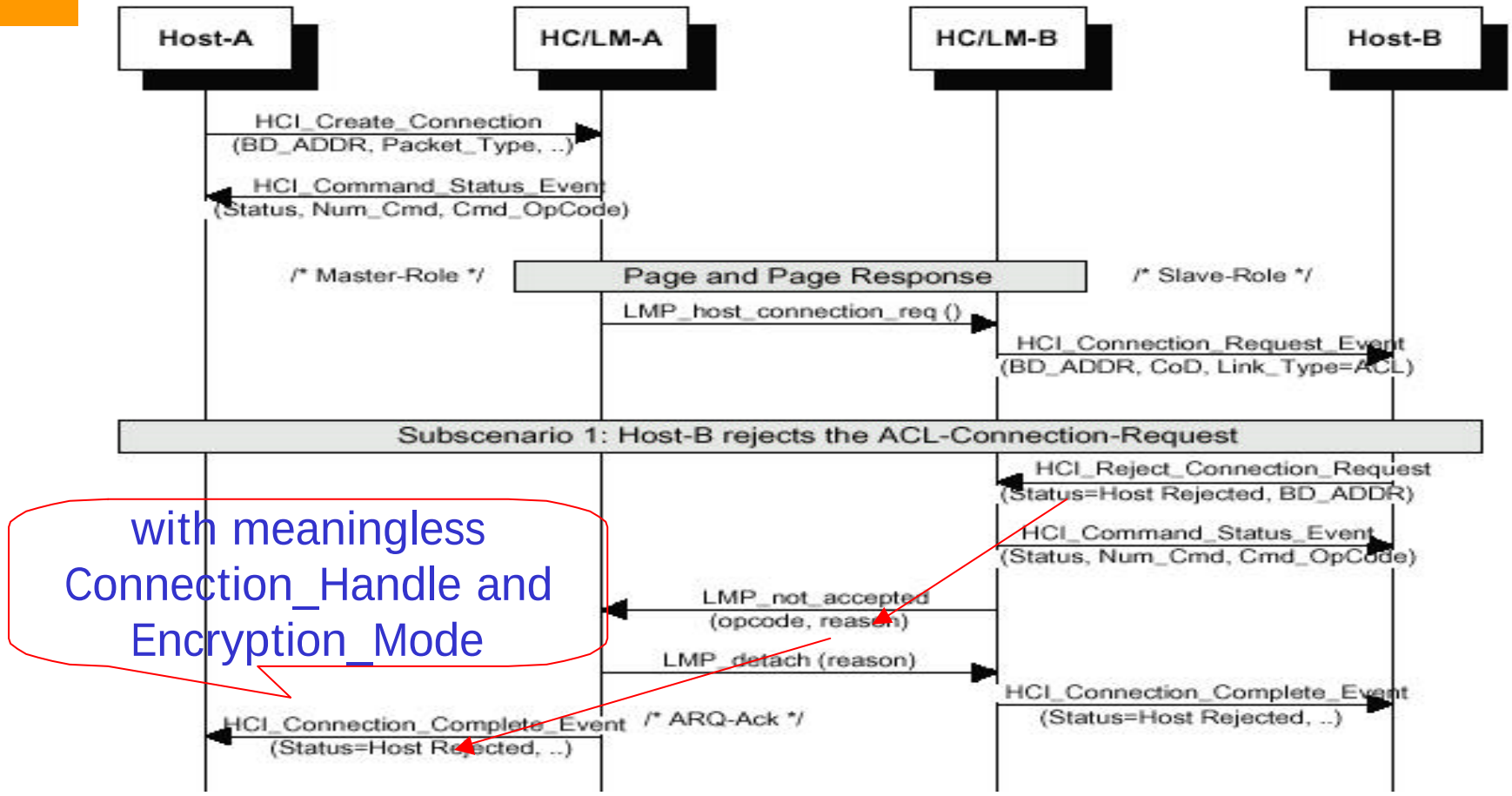
Figure 3.2
ACL Connection Request phase

HCI_Create_Connection



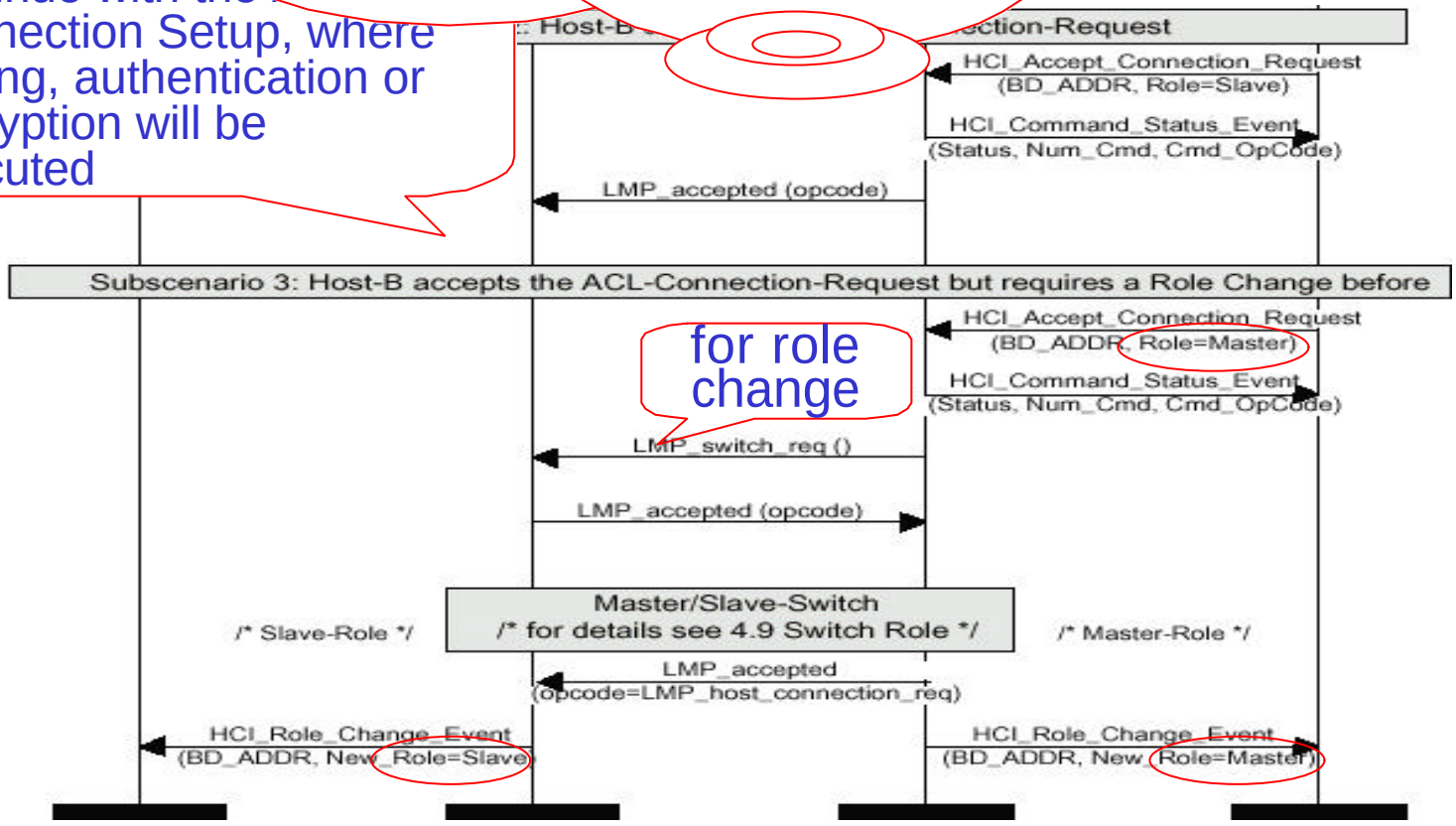

Three sub-scenarios:
1. Slave rejects ACL Connection Request
2. Slave accepts ACL Connection Request
3. Slave accepts ACL Connection Request with Role Change

# MESSAGE SEQUENCE CHART (cont.)



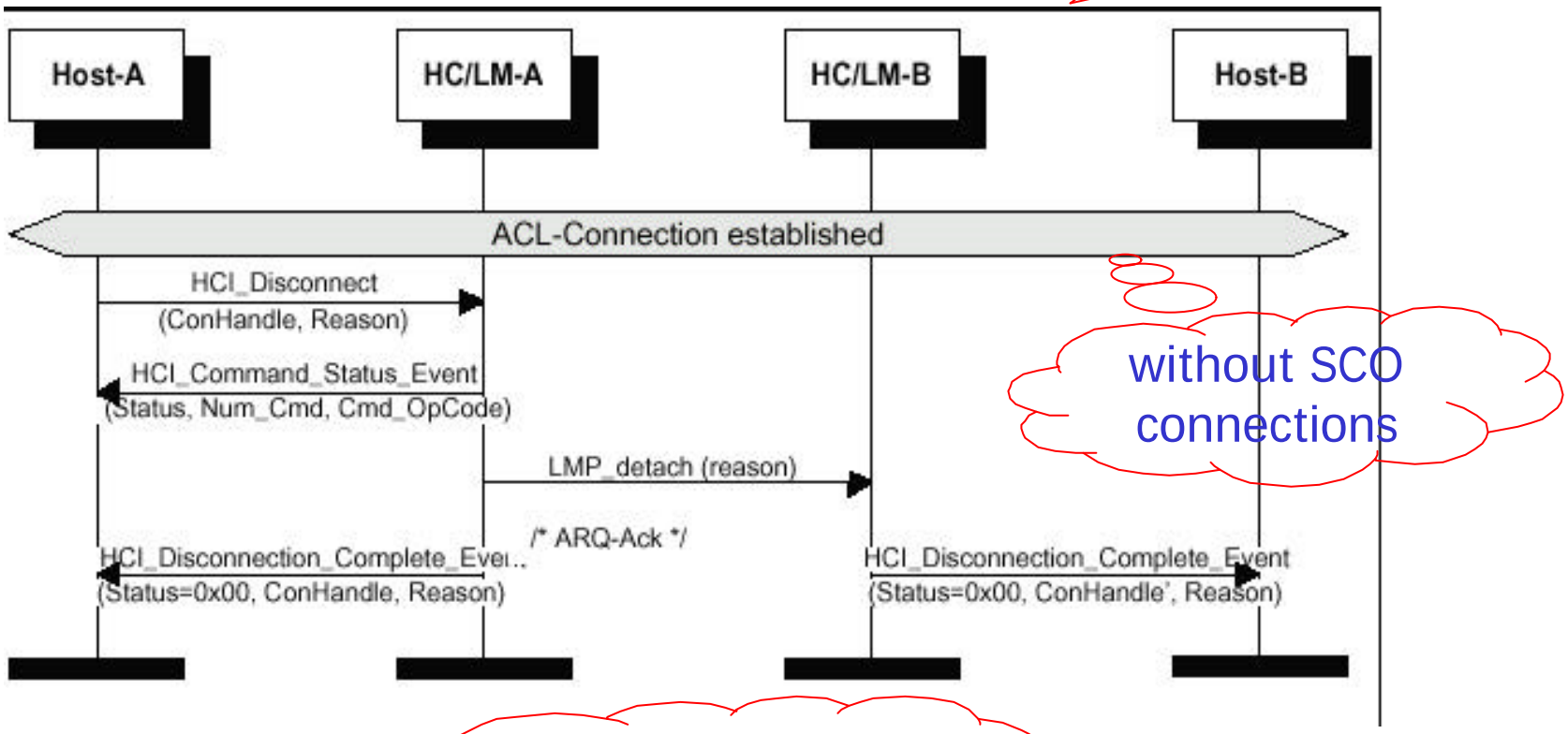
To automatically accept incoming connection request:
HCI_Set_Event_Filter (Filter_Type, Filter_Condition_Type, Condition)
with the Filter_Type = 0x02

continue with the Connection Setup, where pairing, authentication or encryption will be executed



# MESSAGE SEQUENCE CHART (cont.)

*Figure 3.6*
*ACL Disconnection*

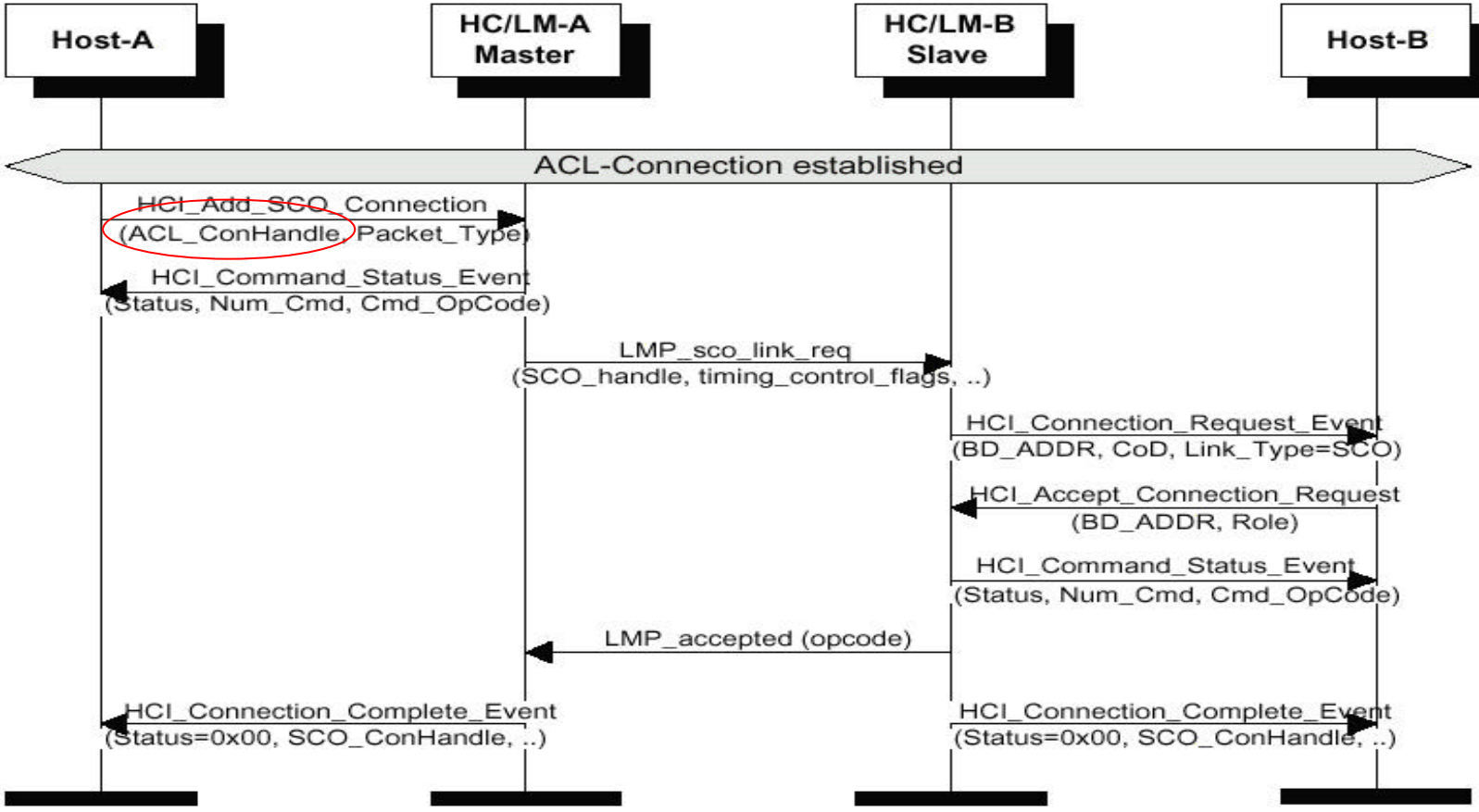



without SCO connections

one-sided procedure

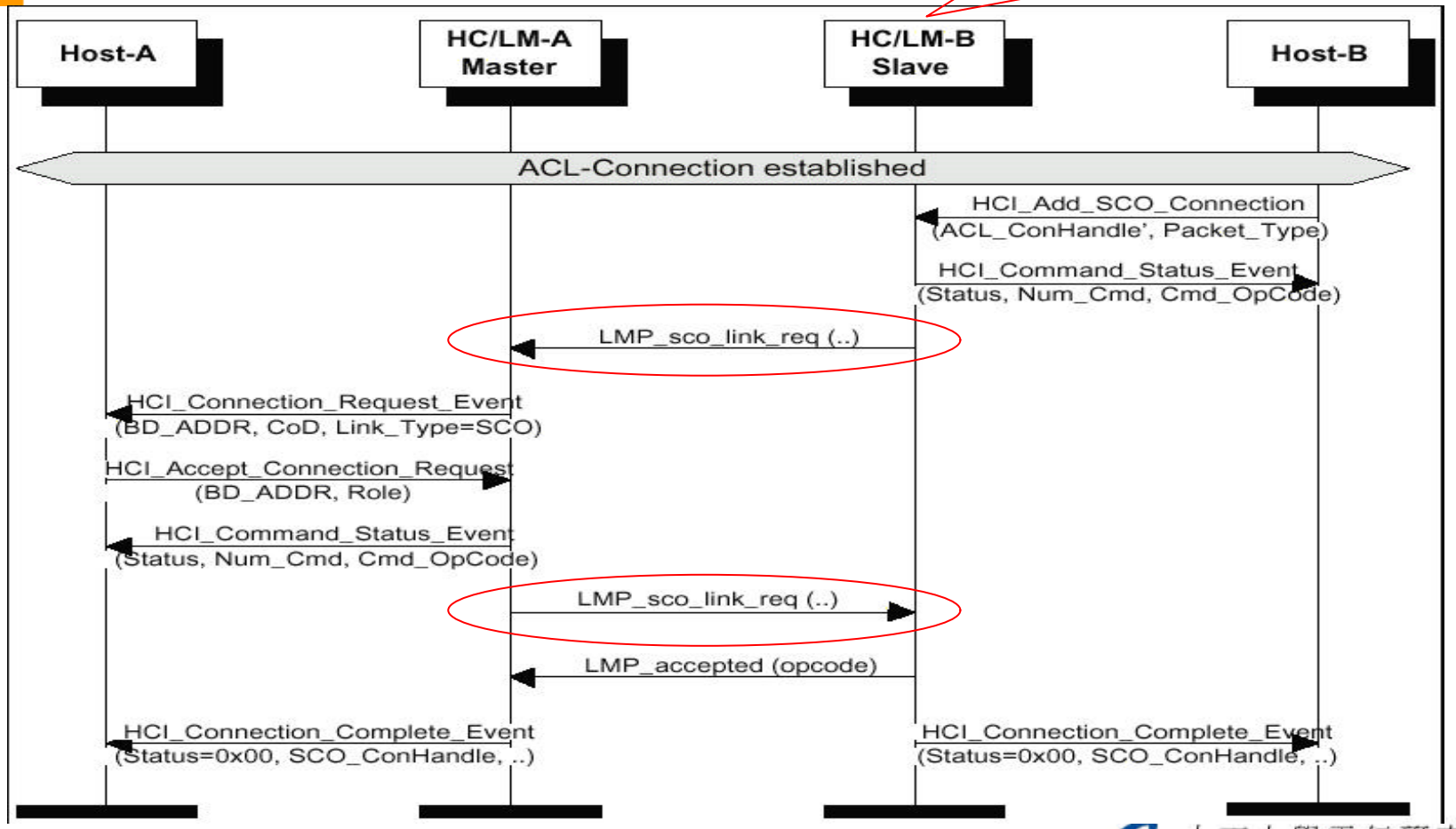
# MESSAGE SEQUENCE CHART (cont.)

*Figure 5.1*
*SCO Connection setup (activated from master)*



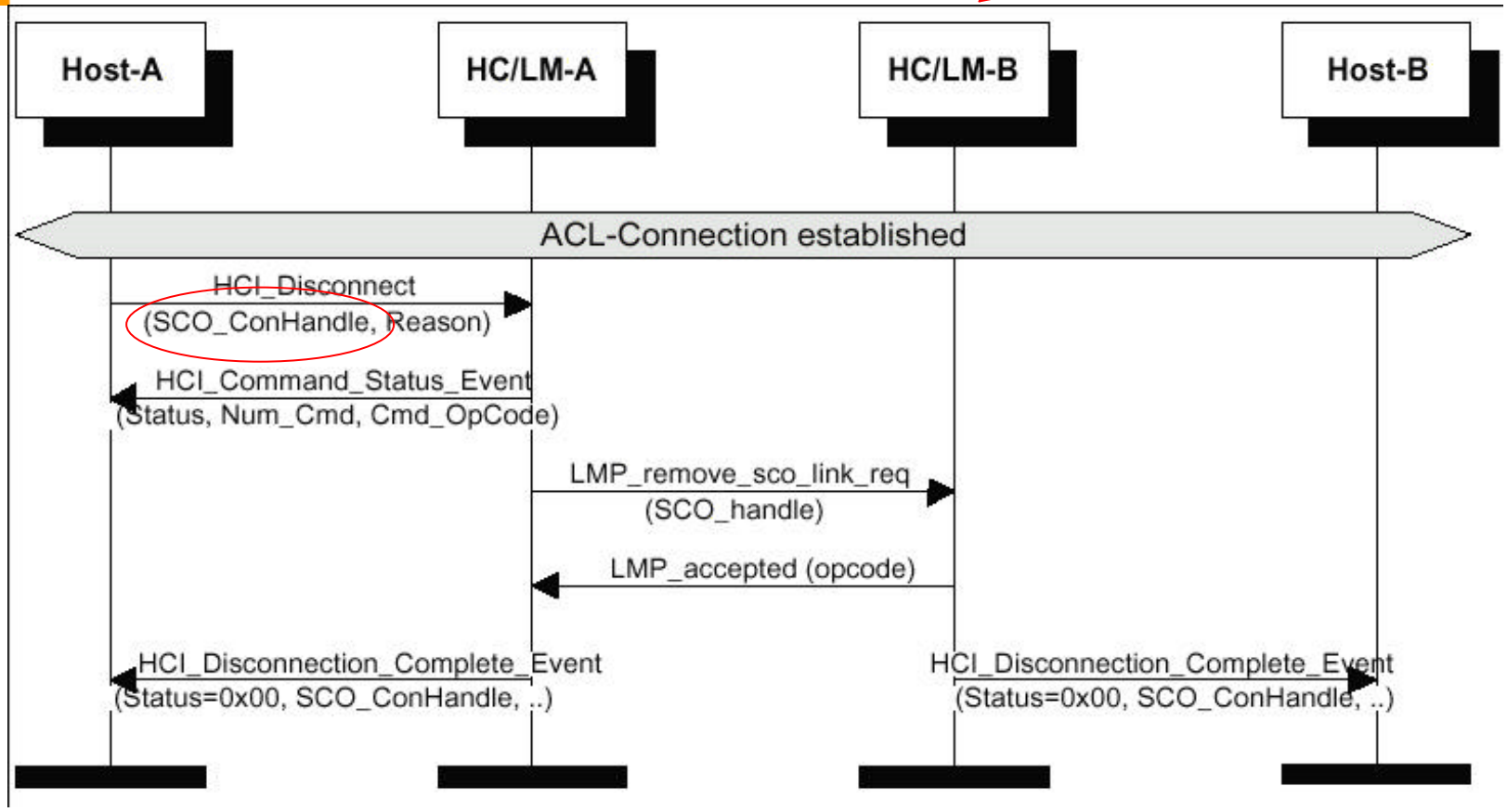
# MESSAGE SEQUENCE CHART (cont.)

*Figure 5.2*
*SCO Connection setup*
*(activated from slave)*



# MESSAGE SEQUENCE CHART (cont.)

*Figure 5.3*
*SCO Disconnection*



# CONCLUSION & DISCUSSION

- **Providing a uniform interface method of accessing the Bluetooth hardware capabilities**
- **Two parts of HCI commands**
  - HCI driver in Bluetooth host
  - HCI firmware in Bluetooth hardware
    - Implementing the HCI commands by accessing :
      - baseband commands,
      - link manager commands,
      - hardware status registers,
      - control registers,
      - event registers