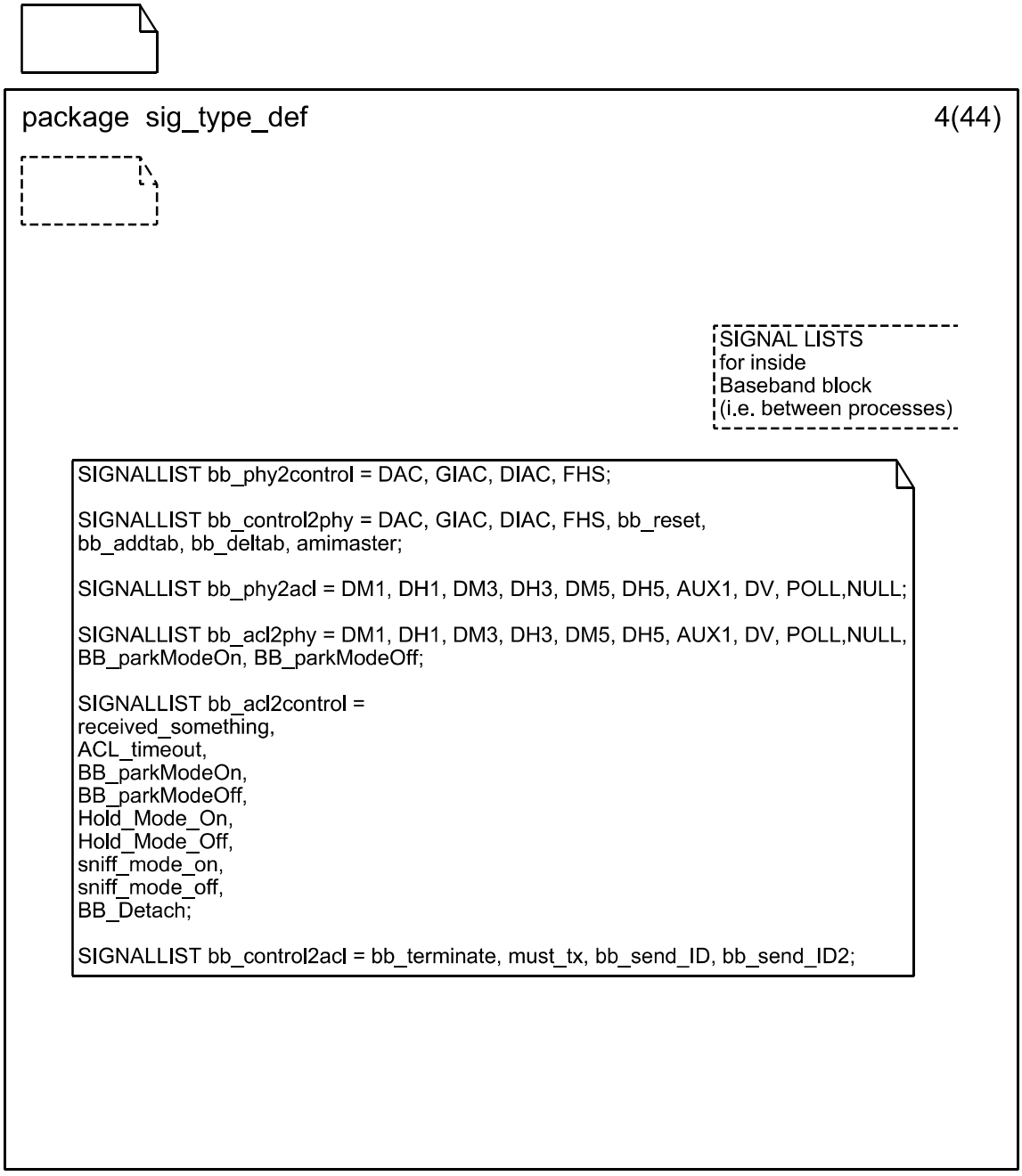
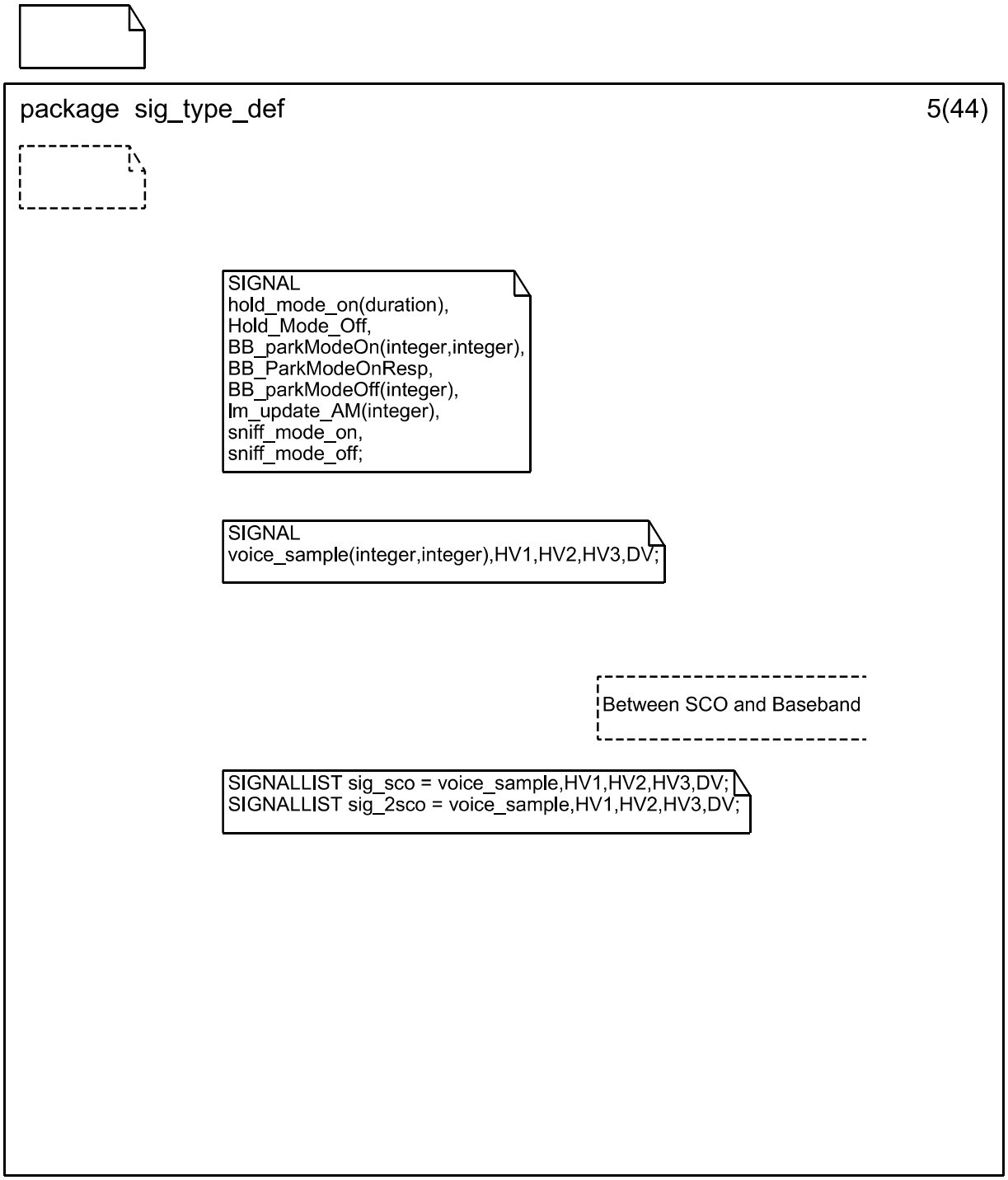
B.7.4 sig_type_def-4



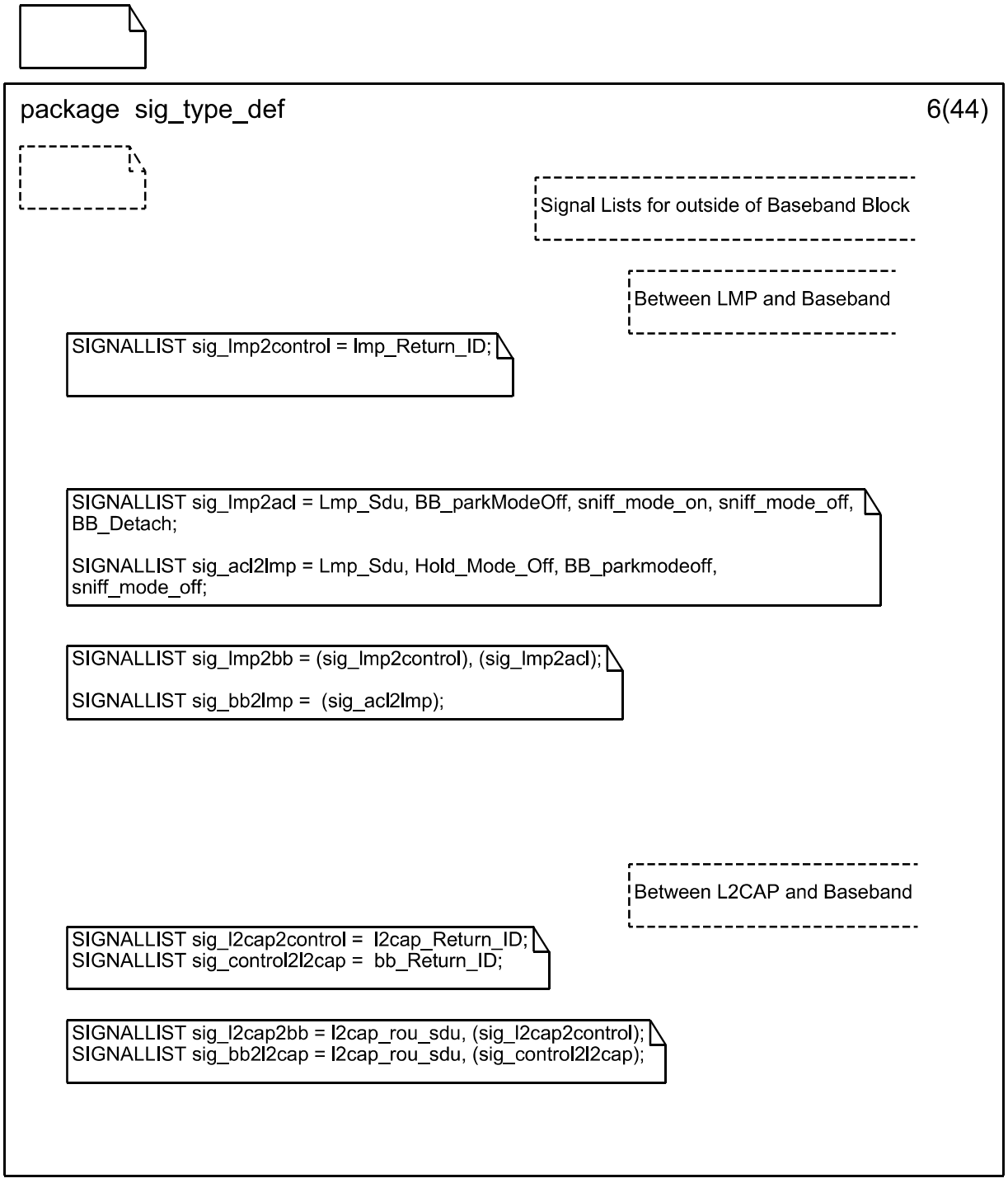
Copyright © 2002 IEEE. All rights reserved.

B.7.5 sig_type_def-5



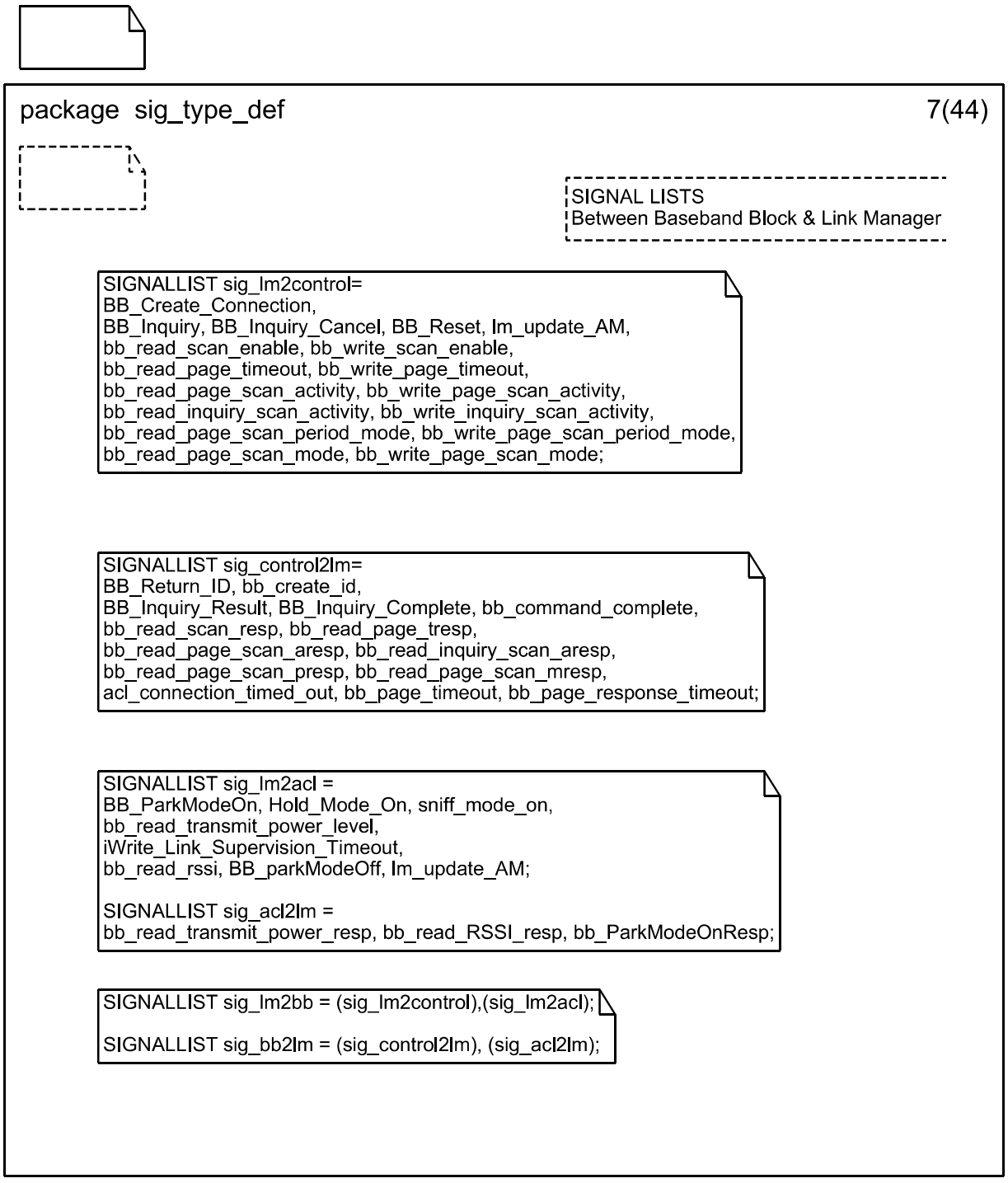
Copyright © 2002 IEEE. All rights reserved.

B.7.6 sig_type_def-6



Copyright © 2002 IEEE. All rights reserved.

B.7.7 sig_type_def-7



Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1040 of 1169

B.7.8 sig_type_def-8



package sig_type_def 8(44)



```
newtype pckt_new choice
      ID_pckt ac_types;
      non_ID_pckt slot;
endnewtype pckt_new;

newtype slot
   struct
       access_code ac_types;
       header pk_head;
       payload  contents;
endnewtype slot;

newtype ac_types
  literals CAC, DAC, GIAC, DIAC
endnewtype ac_types;

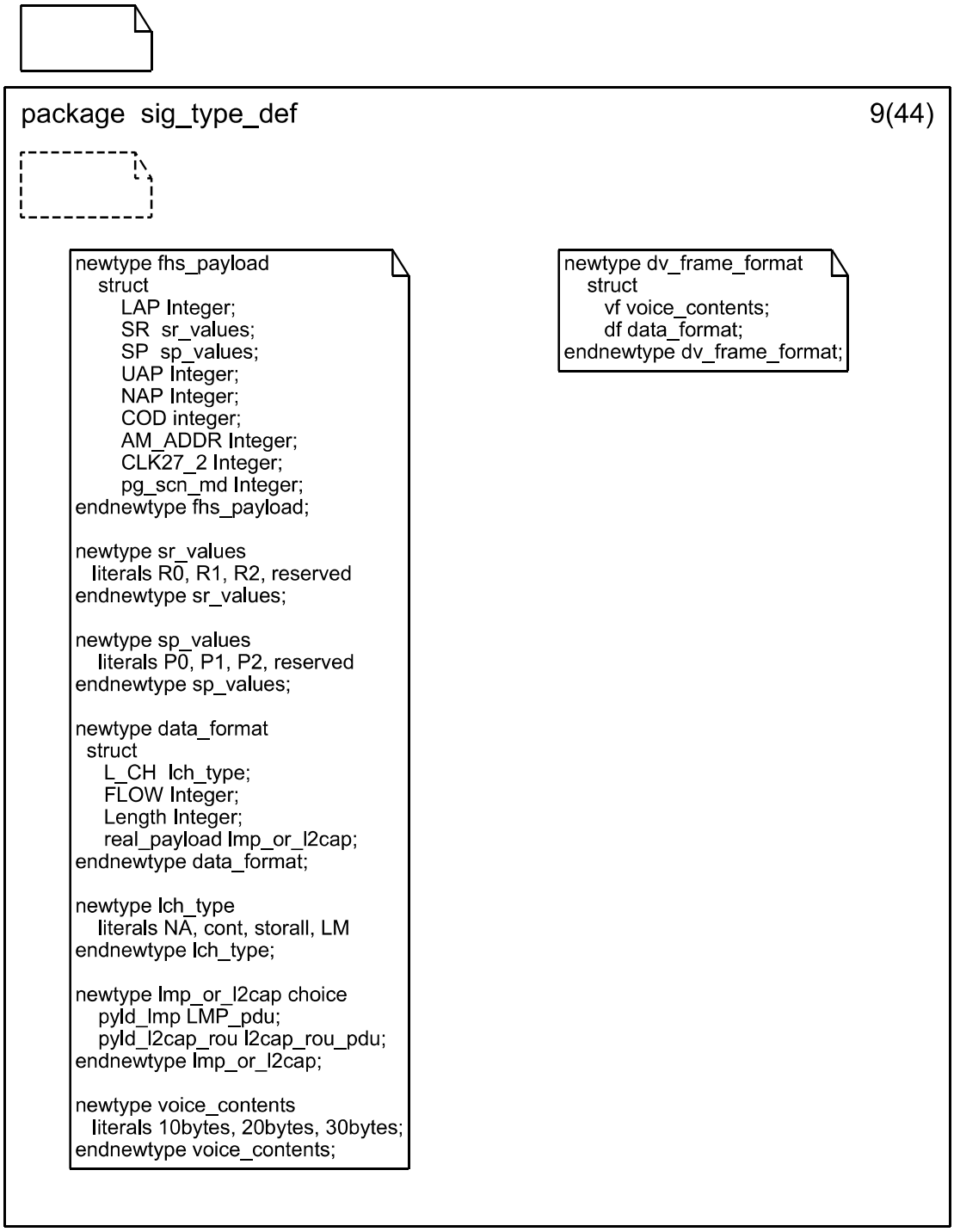
newtype pk_head
   struct
       AM_ADDR  Integer;
       _Type  p_type;
       flow  Integer;
       arqn  Integer;
       seqn  Integer;
       HEC boolean;
endnewtype pk_head;

newtype  p_type
  literals   NULL, POLL, FHS, DM1, DH1, HV1, HV2, HV3,
             DV, AUX1, DM3, DH3, DM5, DH5;
endnewtype p_type;

newtype contents choice
   df  data_format;
   vc  voice_contents;
   fhsp fhs_payload;
   dvp  dv_frame_format;
   empty integer;
endnewtype contents;
```


Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1041 of 1169

B.7.9 sig_type_def-9



Copyright © 2002 IEEE. All rights reserved.

B.7.10 sig_type_def-10



package sig_type_def 10(44)



```
newtype bt_entity
   array(Address_type, bt_entry)
endnewtype bt_entity;

newtype bt_entry
struct
    AM_ADDR integer;
    PM_ADDR integer;
    AR_ADDR integer;
    ACL Pid;
    Mode Mode_type;
endnewtype bt_entry;

newtype AM_array
   array(integer,Address_type)
endnewtype AM_array;
```

```
newtype RSSI_Type
  literals below, inside, above
endnewtype RSSI_Type;
```

```
newtype l2cap_rou_pdu
  struct
    CID Integer;
    Length Integer;
    pyld_l2cap l2cap_pdu;
endnewtype l2cap_rou_pdu;

newtype l2cap_pdu
  choice
    sign sign_type;
    co co_type;
    cl cl_type;
endnewtype l2cap_pdu;

newtype co_type
  struct
    pyld Integer;
endnewtype co_type;

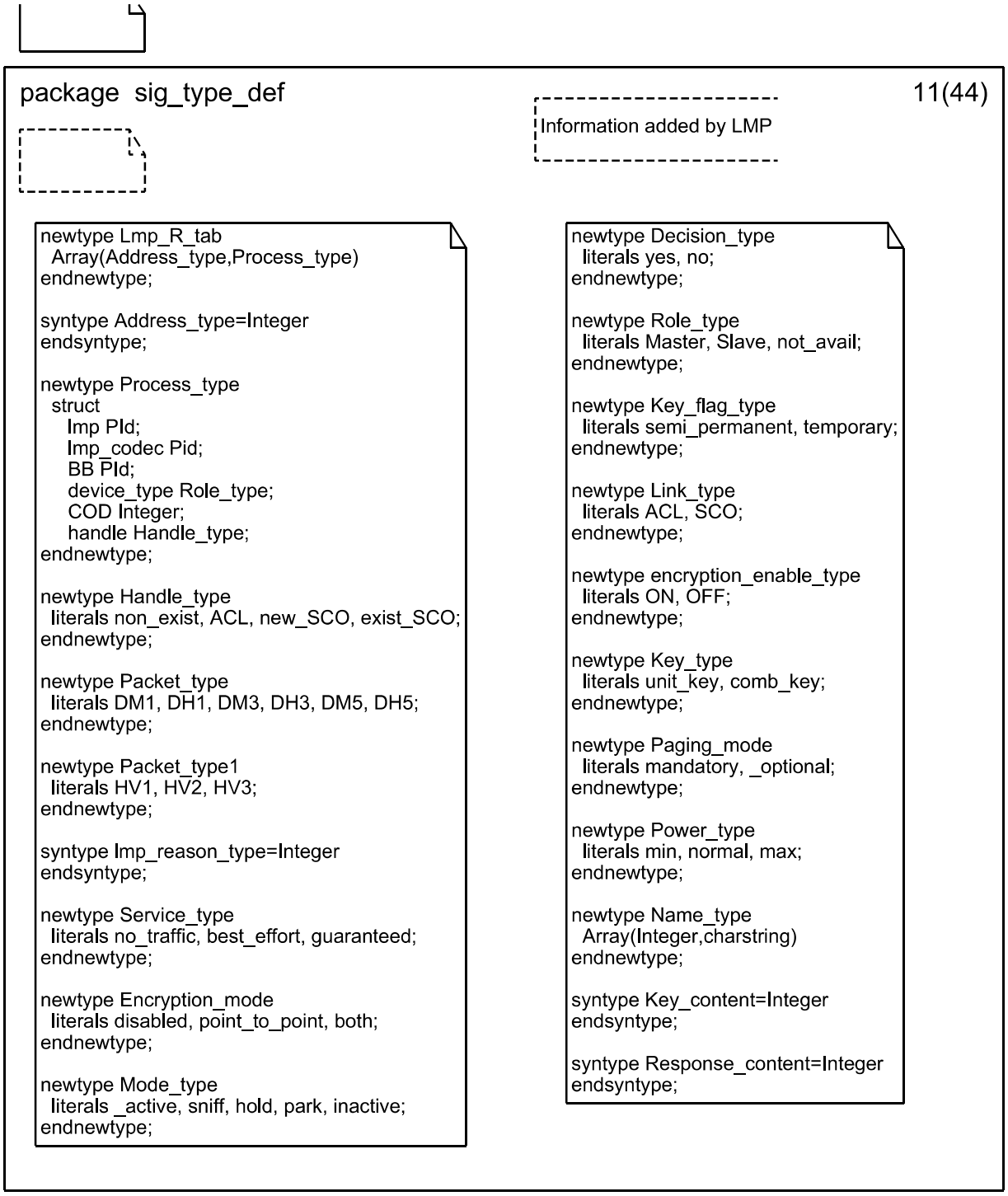
newtype cl_type
  struct
    PSM PSM_type;
    pyld Integer;
endnewtype cl_type;
```

```
newtype rou_db_type
  array (Integer, rou_info_type)
endnewtype rou_db_type;

newtype rou_info_type
  struct
    RCID Integer;
    CO_pid Pid;
endnewtype rou_info_type;
```


Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1043 of 1169

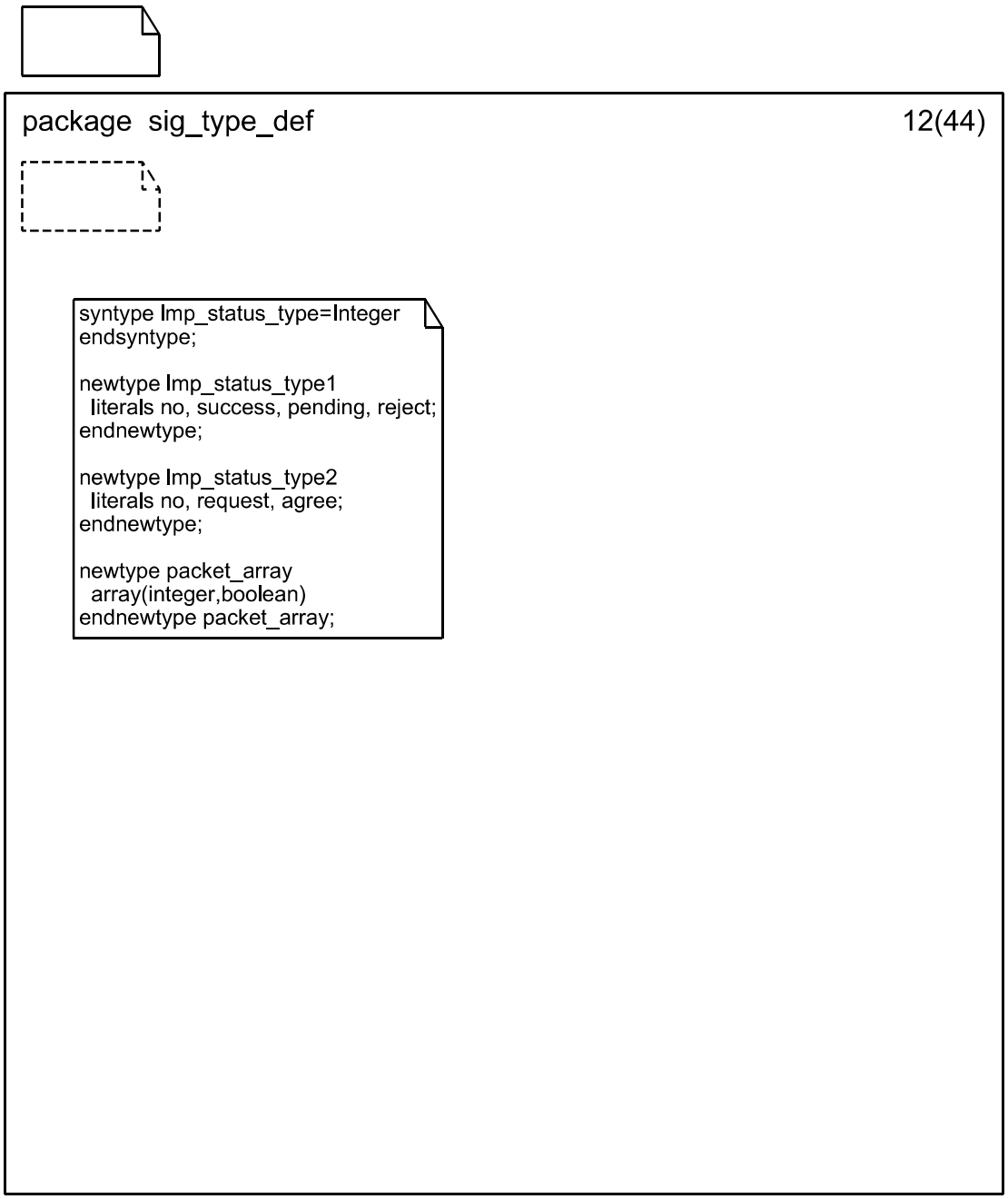
B.7.11 sig_type_def-11



Copyright © 2002 IEEE. All rights reserved.

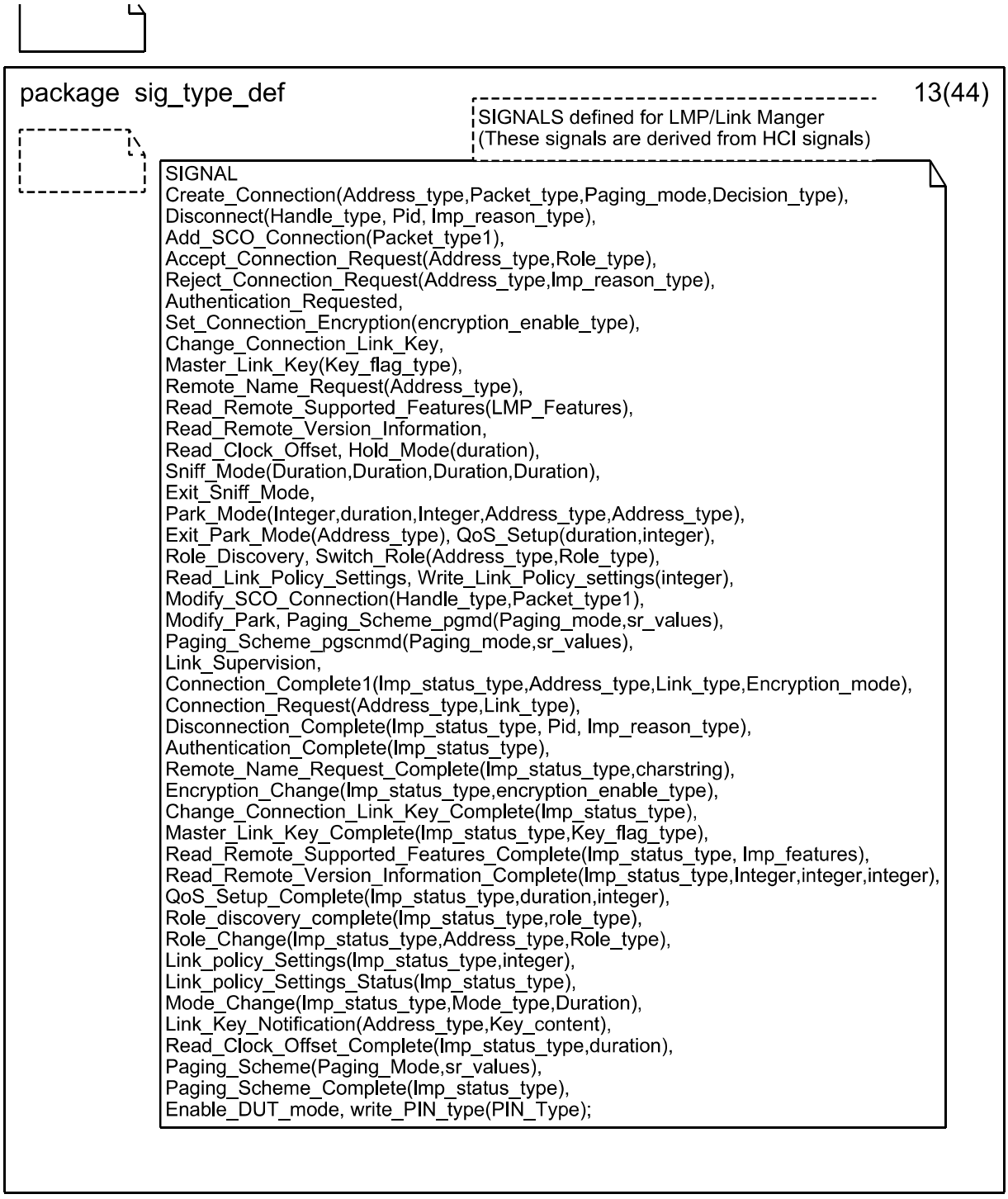
LINKPLAY EXHIBIT 1008
Page 1044 of 1169

B.7.12 sig_type_def-12



Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1045 of 1169

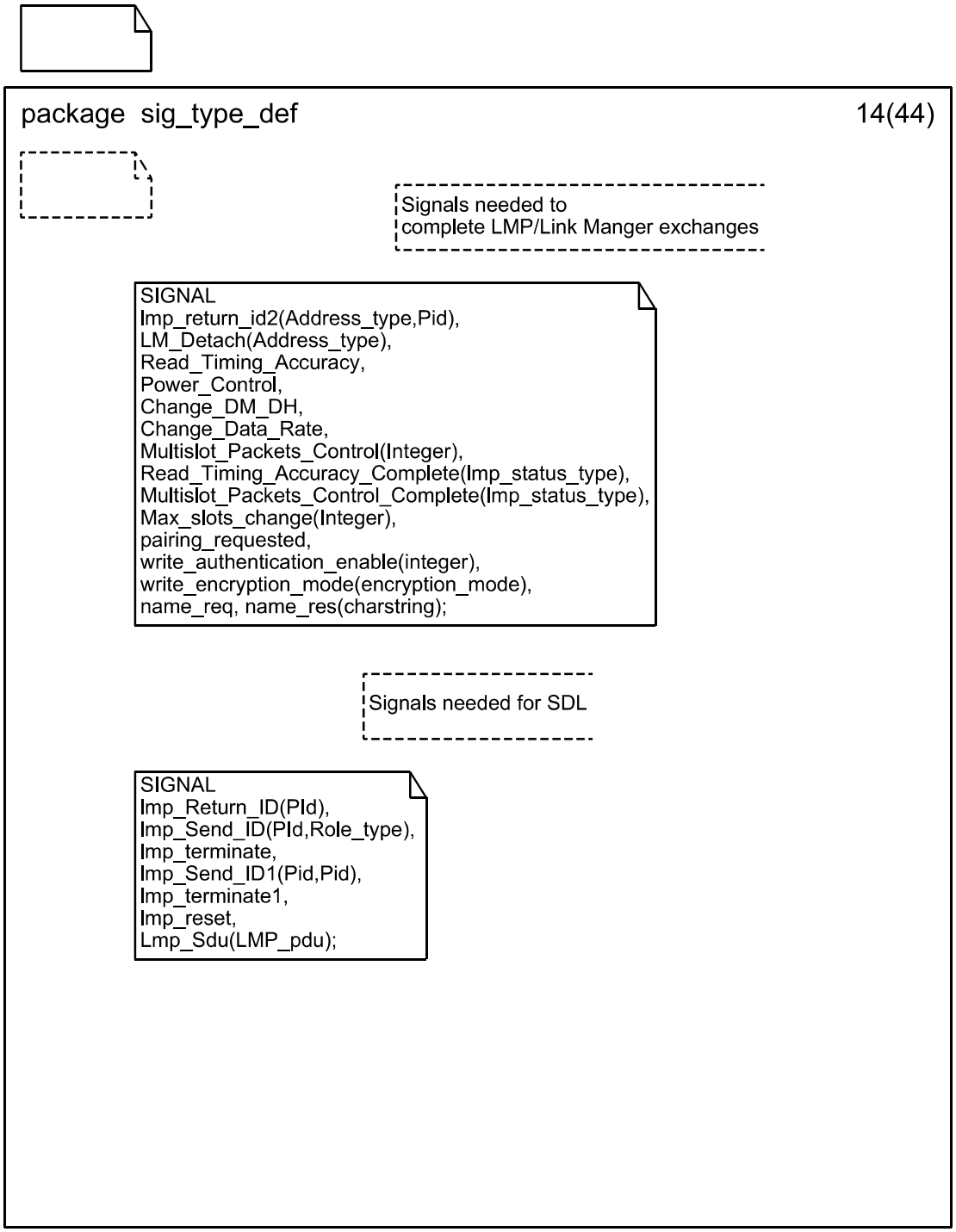
B.7.13 sig_type_def-13



Copyright © 2002 IEEE. All rights reserved.

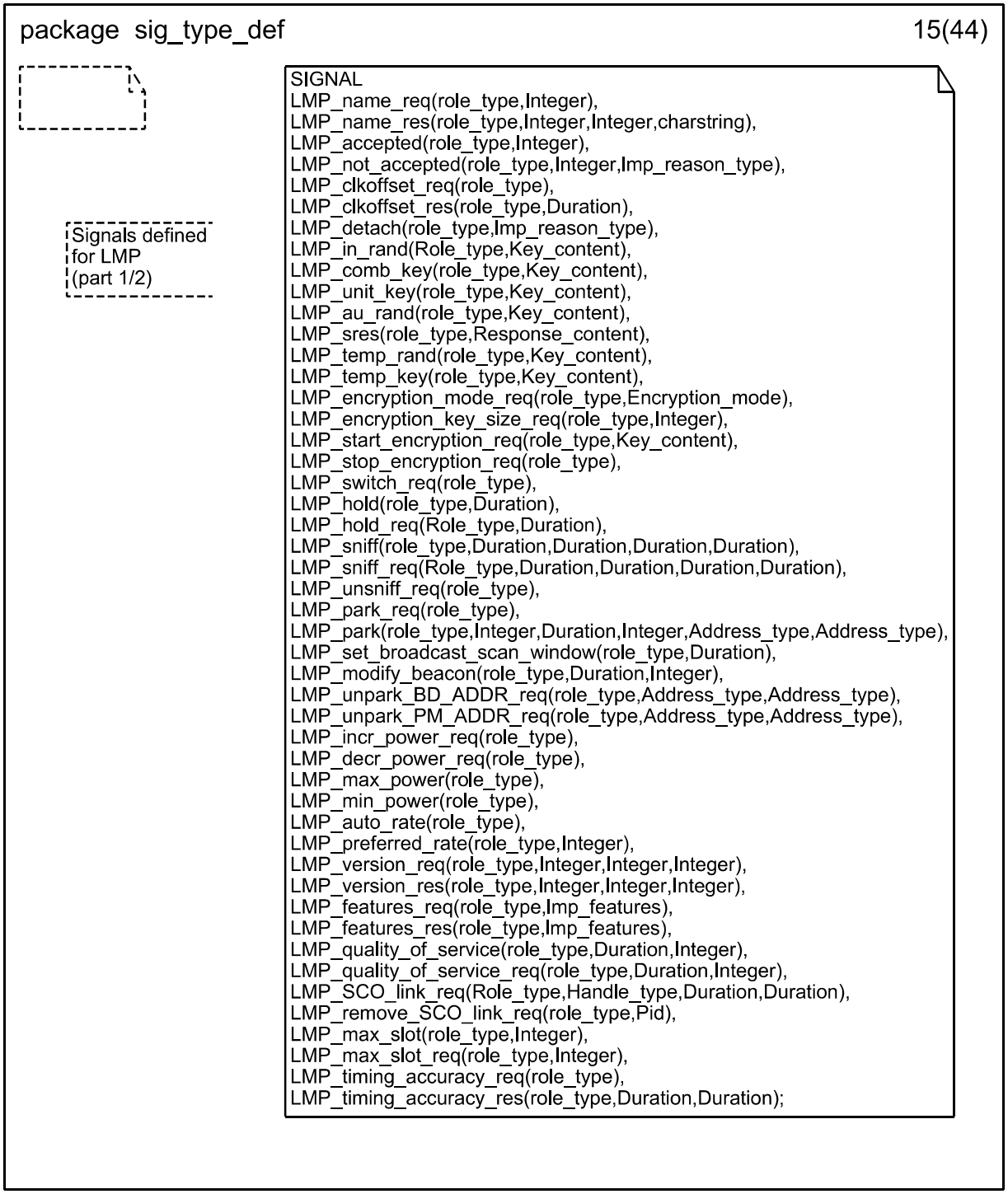
LINKPLAY EXHIBIT 1008
Page 1046 of 1169

B.7.14 sig_type_def-14



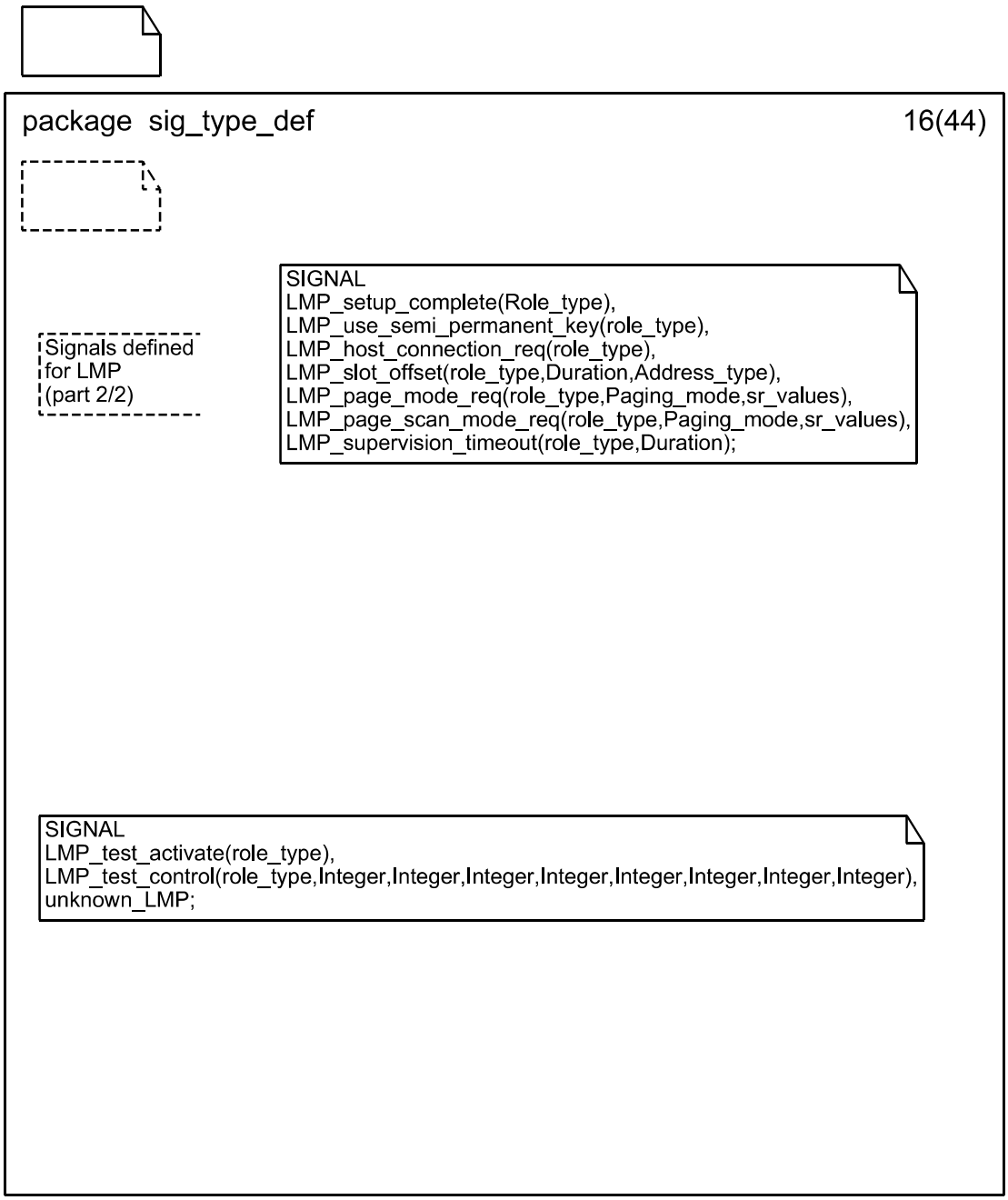
Copyright © 2002 IEEE. All rights reserved.

B.7.15 sig_type_def-15



Copyright © 2002 IEEE. All rights reserved.

B.7.16 sig_type_def-16



Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1049 of 1169

B.7.17 sig_type_def-17



package sig_type_def 17(44)



```
signallist LMP_Cmds=
 LMP_name_req,                    LMP_name_res,
 LMP_accepted,                    LMP_not_accepted,
 LMP_clkoffset_req,               LMP_clkoffset_res,
 LMP_detach,
 LMP_in_rand,
 LMP_comb_key,                    LMP_unit_key,
 LMP_au_rand,                     LMP_sres,
 LMP_temp_rand,                   LMP_temp_key,
 LMP_encryption_mode_req,         LMP_encryption_key_size_req,
 LMP_start_encryption_req,        LMP_stop_encryption_req,
 LMP_switch_req,
 LMP_hold,                        LMP_hold_req,
 LMP_sniff,                       LMP_sniff_req,
 LMP_unsniff_req,
 LMP_park_req,                    LMP_park,
 LMP_set_broadcast_scan_window,
 LMP_modify_beacon,
 LMP_unpark_BD_ADDR_req,
 LMP_unpark_PM_ADDR_req,
 LMP_incr_power_req,              LMP_decr_power_req,
 LMP_max_power,                   LMP_min_power,
 LMP_auto_rate,                   LMP_preferred_rate,
 LMP_version_req,                 LMP_version_res,
 LMP_features_req,                LMP_features_res,
 LMP_quality_of_service,          LMP_quality_of_service_req,
 LMP_SCO_link_req,
 LMP_remove_SCO_link_req,
 LMP_max_slot,                    LMP_max_slot_req,
 LMP_timing_accuracy_req,         LMP_timing_accuracy_res,
 LMP_setup_complete,
 LMP_use_semi_permanent_key,
 LMP_host_connection_req,
 LMP_slot_offset,
 LMP_page_mode_req,
 LMP_page_scan_mode_req,
 LMP_supervision_timeout,
 LMP_test_activate,               LMP_test_control,
 unknown_LMP;
```

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1050 of 1169

B.7.18 sig_type_def-18



package sig_type_def 18(44)



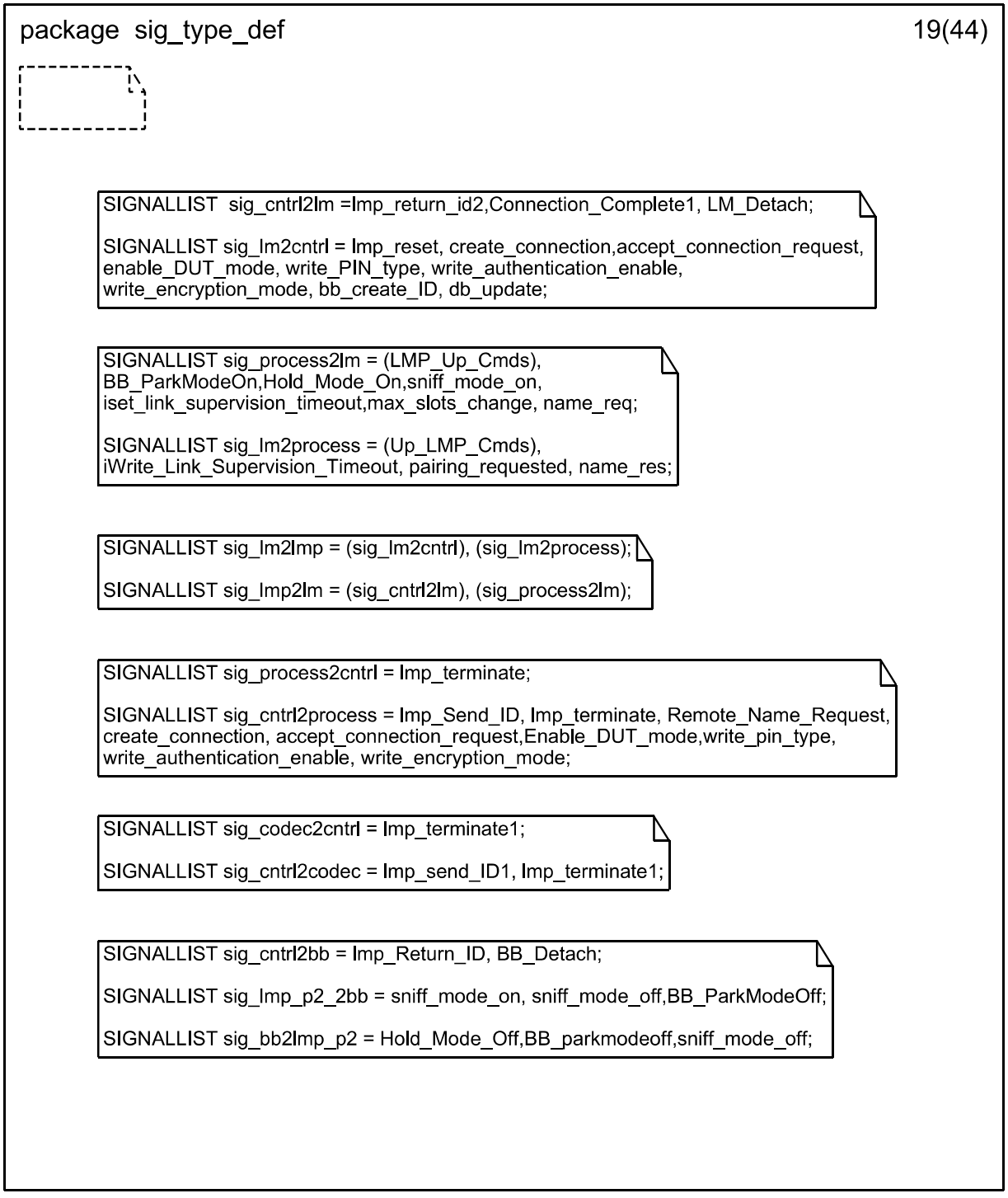
```
signallist Up_LMP_Cmds=
 Disconnect,
 Add_SCO_Connection,
 Accept_Connection_Request,
 Reject_Connection_Request,
 Authentication_Requested,
 Set_Connection_Encryption,
 Change_Connection_Link_Key,
 Master_Link_Key,
 Remote_Name_Request,
 Read_Remote_Supported_Features,
 Read_Remote_Version_Information,
 Read_Clock_Offset,
 Hold_Mode,
 Sniff_Mode,
 Exit_Sniff_Mode,
 Park_Mode,
 Exit_Park_Mode,
 QoS_Setup,
 Role_discovery,
 Switch_Role,
 Read_Link_Policy_Settings,
 Write_Link_Policy_settings,

 Modify_SCO_Connection,
 Modify_Park,
 Read_Timing_Accuracy,
 Power_Control,
 Change_DM_DH,
 Change_Data_Rate,
 Multislot_Packets_Control,
 Paging_Scheme_pgmd,
 Paging_Scheme_pgscnmd,
 Link_Supervision;
```

```
SIGNALLIST LMP_Up_Cmds=
 Connection_Complete1,
 Connection_Request,
 Disconnection_Complete,
 Authentication_Complete,
 Remote_Name_Request_Complete,
 Encryption_Change,
 Change_Connection_Link_Key_Complete,
 Master_Link_Key_Complete,
 Read_Remote_Supported_Features_Complete,
 Read_Remote_Version_Information_Complete,
 QoS_Setup_Complete,
 Role_discovery_complete,
 Role_Change,
 Link_policy_Settings,
 Link_policy_Settings_Status,
 Mode_Change,
 Link_Key_Notification,
 Read_Clock_Offset_Complete,
 Read_Timing_Accuracy_Complete,
 Multislot_Packets_Control_complete,
 Paging_Scheme,
 Paging_Scheme_Complete,
 lm_update_AM,
lm_detach;
```


Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1051 of 1169

B.7.19 sig_type_def-19

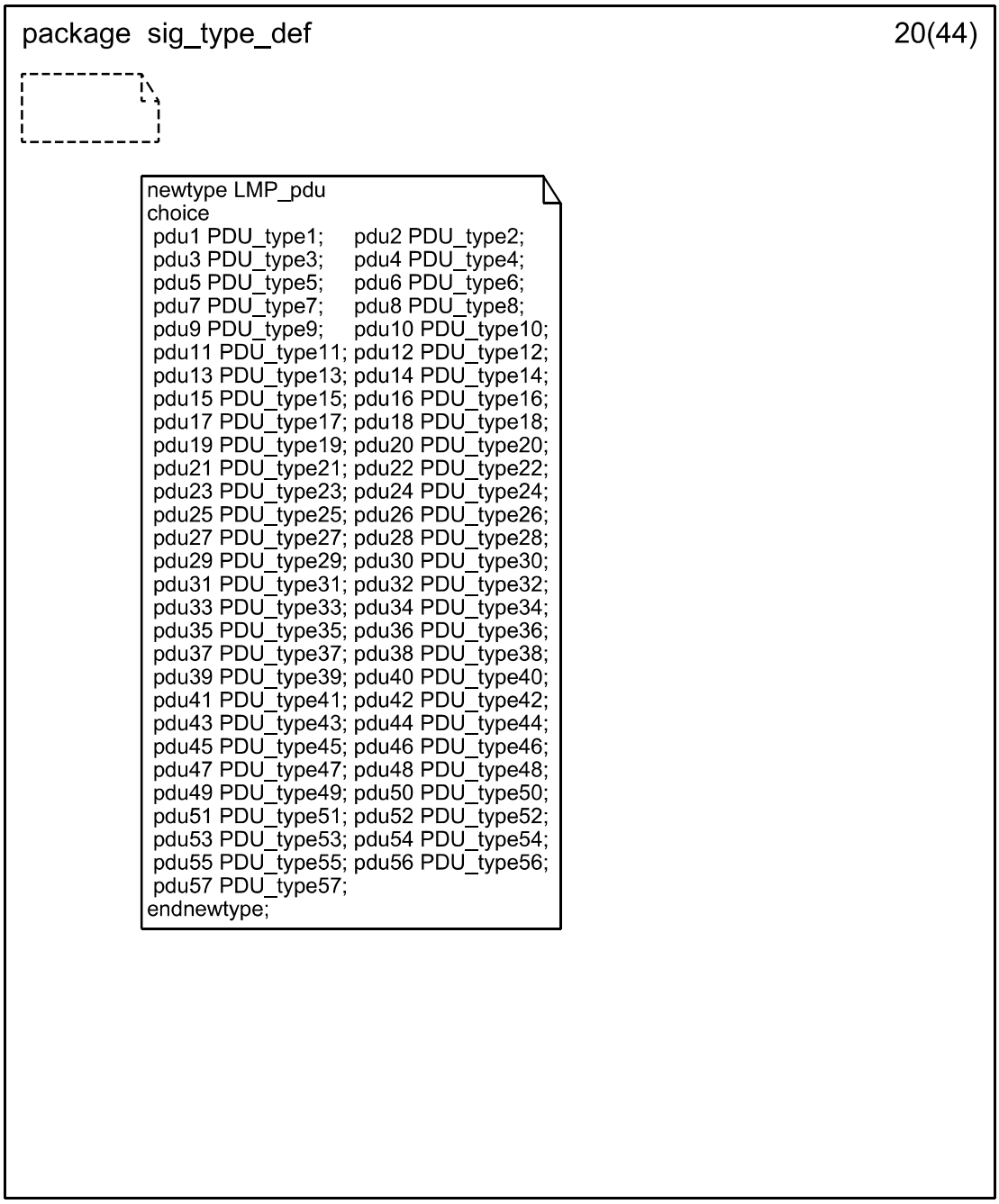


Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1052 of 1169

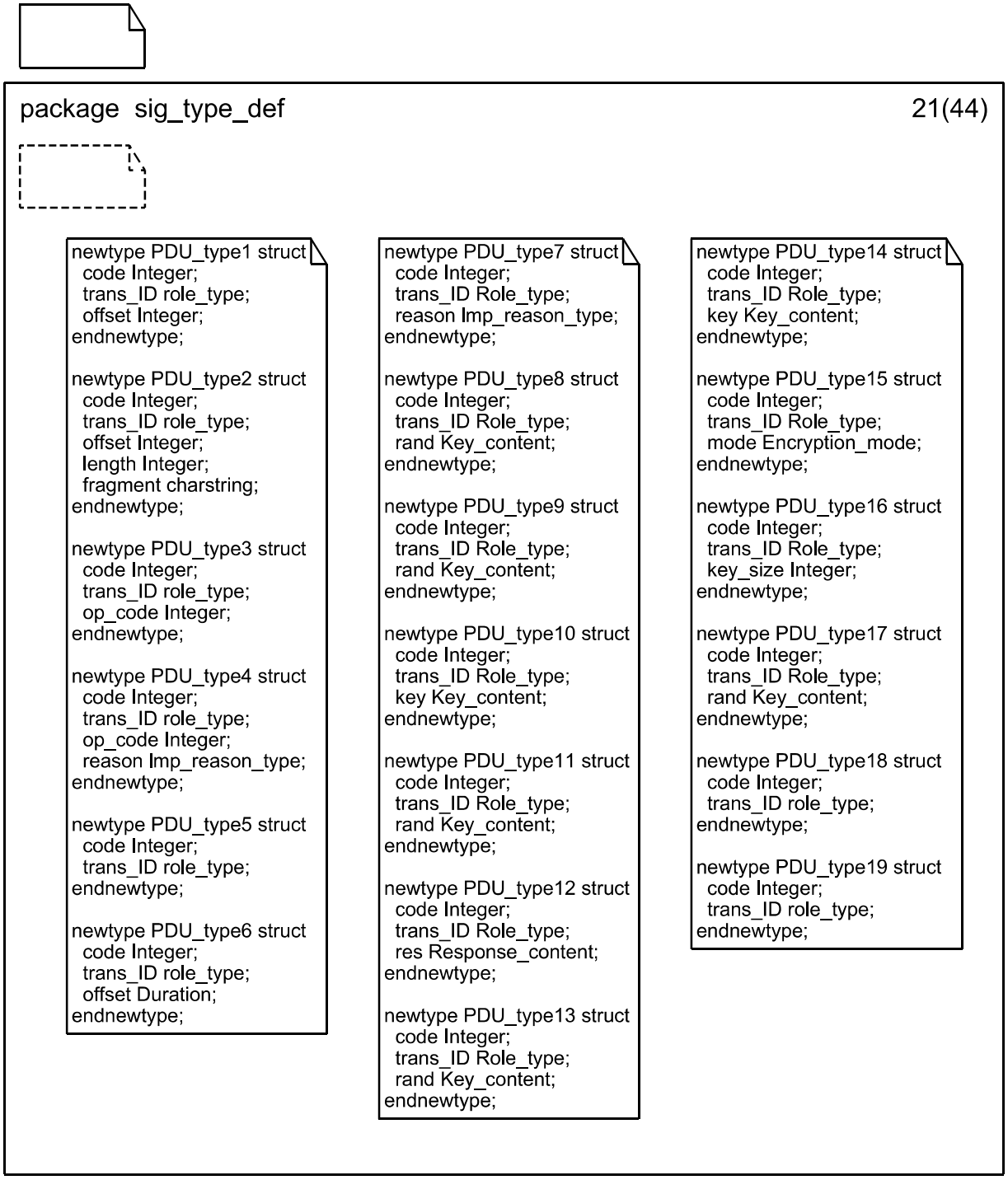
B.7.20 sig_type_def-20





Copyright © 2002 IEEE. All rights reserved.

B.7.21 sig_type_def-21



Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1054 of 1169

B.7.22 sig_type_def-22



package sig_type_def 22(44)



```
newtype PDU_type20 struct
  code Integer;
  trans_ID Role_type;
  hold_time Duration;
endnewtype;

newtype PDU_type21 struct
  code Integer;
  trans_ID Role_type;
  hold_time Duration;
endnewtype;

newtype PDU_type22 struct
  code Integer;
  trans_ID role_type;
  Dsniff Duration;
  Tsniff Duration;
  attempt Duration;
  timeout Duration;
endnewtype;

newtype PDU_type23 struct
  code Integer;
  trans_ID Role_type;
  Dsniff Duration;
  Tsniff Duration;
  attempt Duration;
  timeout Duration;
endnewtype;

newtype PDU_type24 struct
  code Integer;
  trans_ID role_type;
endnewtype;

newtype PDU_type25 struct
  code Integer;
  trans_ID role_type;
endnewtype;
```

```
newtype PDU_type26 struct
  code Integer;
  trans_ID role_type;
  DB Integer;
  TB Duration;
  NB Integer;
  PM_ADDR Address_type;
  AR_ADDR Address_type;
endnewtype;

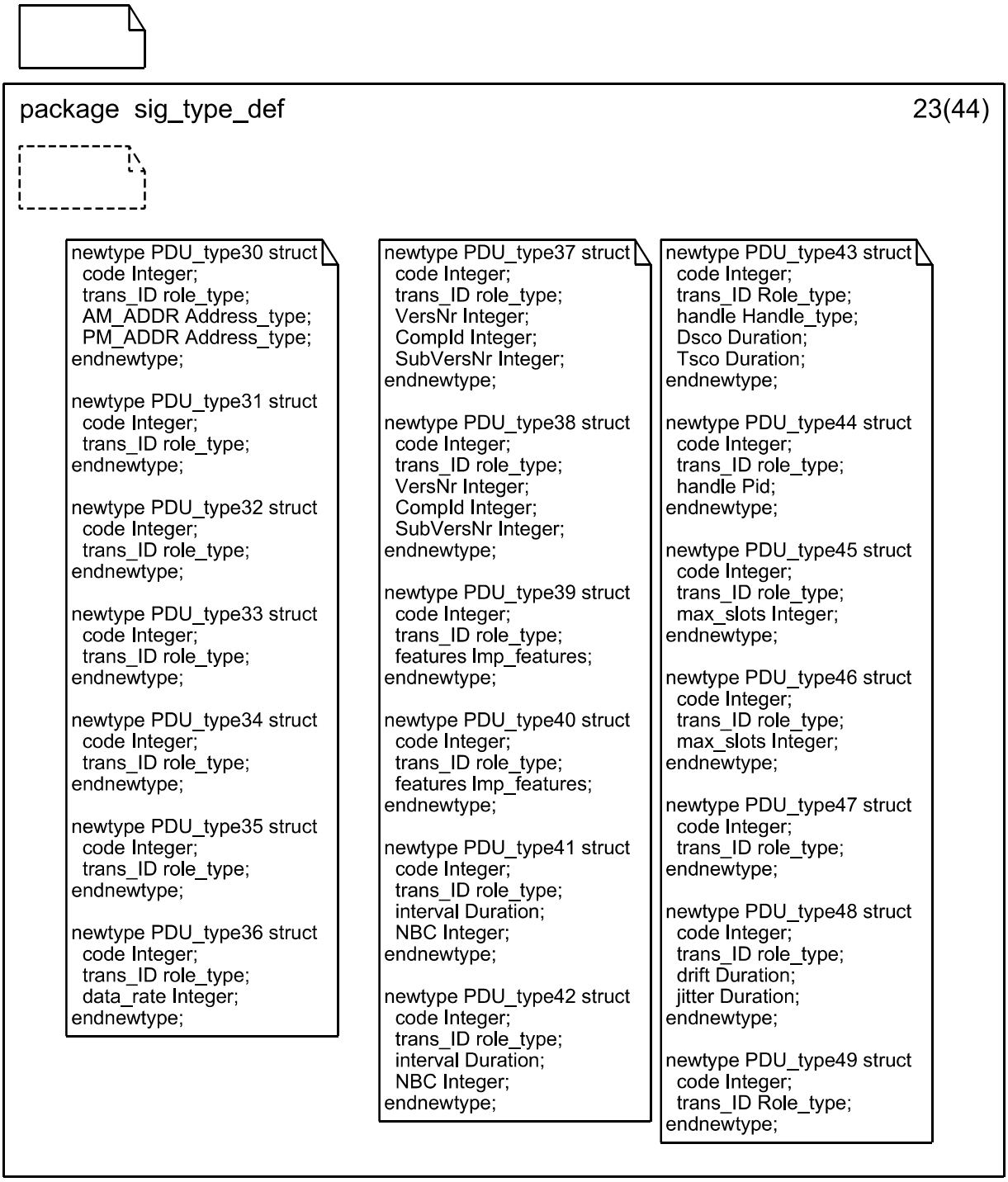
newtype PDU_type27 struct
  code Integer;
  trans_ID role_type;
  scan_window Duration;
endnewtype;

newtype PDU_type28 struct
  code Integer;
  trans_ID role_type;
  TB Duration;
  NB Integer;
endnewtype;

newtype PDU_type29 struct
  code Integer;
  trans_ID role_type;
  AM_ADDR Address_type;
  BD_ADDR Address_type;
endnewtype;
```


Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1055 of 1169

B.7.23 sig_type_def-23



Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1056 of 1169

B.7.24 sig_type_def-24



package sig_type_def 24(44)



```
newtype PDU_type50 struct
  code Integer;
  trans_ID role_type;
endnewtype;

newtype PDU_type51 struct
  code Integer;
  trans_ID role_type;
endnewtype;
```

```
newtype PDU_type52 struct
  code Integer;
  trans_ID role_type;
  offset Duration;
  BD_ADDR Address_type;
endnewtype;

newtype PDU_type53 struct
  code Integer;
  trans_ID role_type;
  paging_scheme Paging_mode;
  paging_scheme_settings sr_values;
endnewtype;

newtype PDU_type54 struct
  code Integer;
  trans_ID role_type;
  paging_scheme Paging_mode;
  paging_scheme_settings sr_values;
endnewtype;

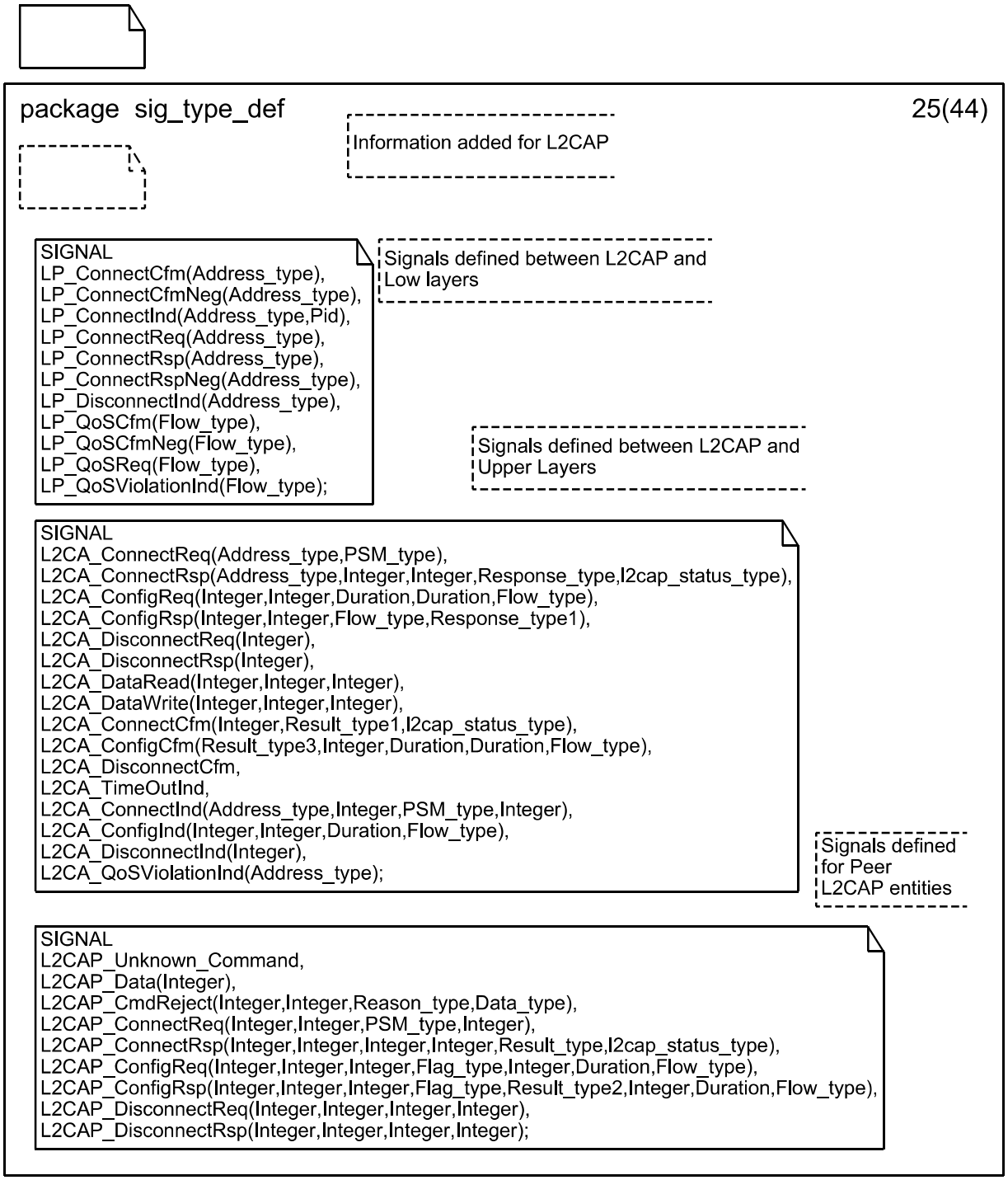
newtype PDU_type55 struct
  code Integer;
  trans_ID role_type;
  timeout Duration;
endnewtype;

newtype PDU_type56 struct
  code Integer;
  trans_ID role_type;
endnewtype;

newtype PDU_type57 struct
   code Integer;
   trans_ID role_type;
   test_scenario,
   hopping_mode,
   TX_freq, RX_freq,
   power_control_mode,
   poll_period,
   packet_type,
   length_test_data Integer;
endnewtype;
```


Copyright © 2002 IEEE. All rights reserved.

B.7.25 sig_type_def-25



Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1058 of 1169

B.7.26 sig_type_def-26



package sig_type_def 26(44)

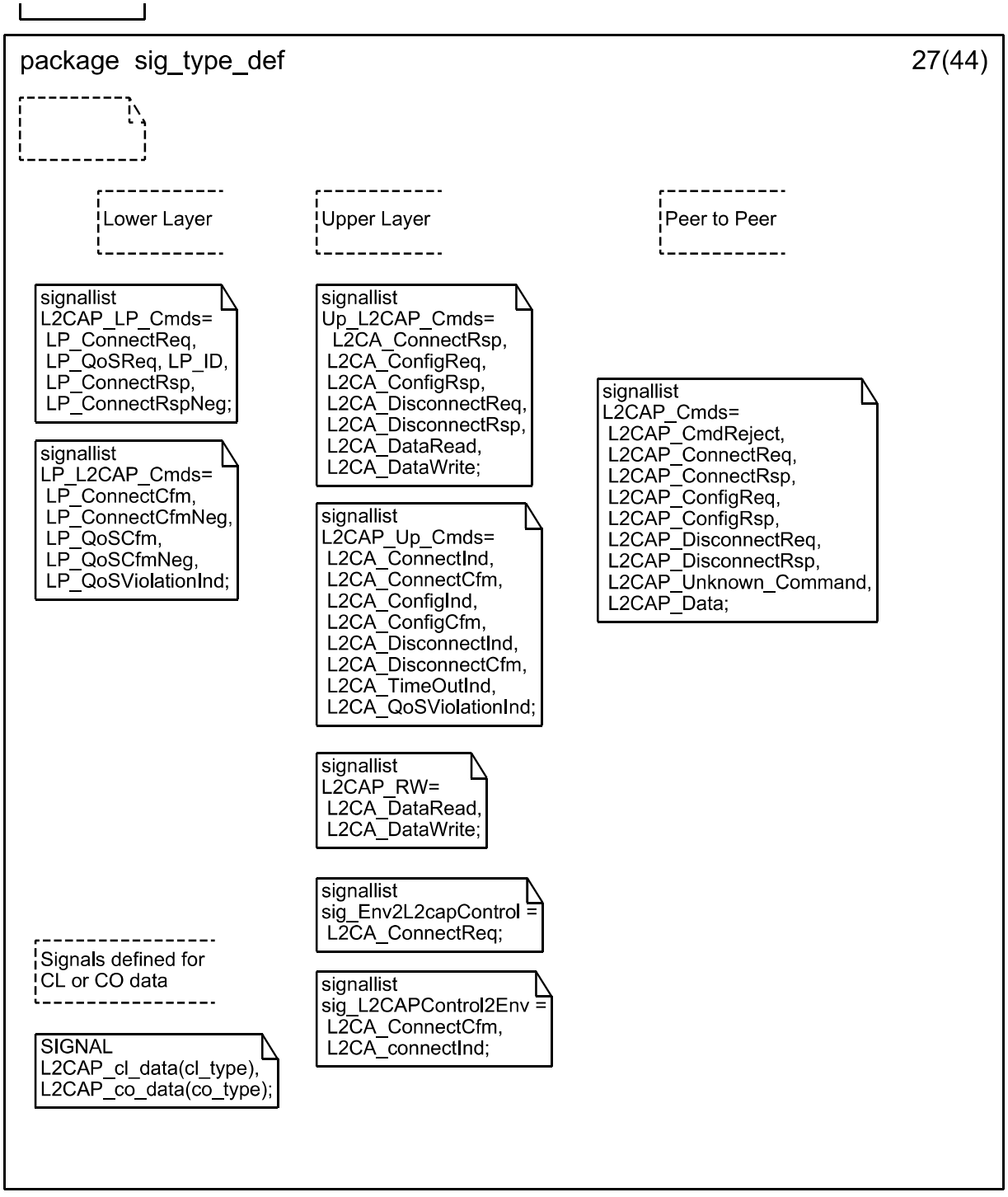


Signals needed for SDL

```
SIGNAL
I2cap_Return_ID(PId),
BB_Create_ID(Address_type,PId,integer,Role_Type), BB_Return_ID(PId,role_type),
I2cap_Send_ID(PId,PId), I2cap_Terminate,
I2cap_Send_ID1(Pid,Pid), I2cap_terminate1,
I2cap_Send_ID2(Pid,Pid), I2cap_terminate3,
I2cap_send_info(address_type,PSM_type,Integer,Integer),
I2cap_send_info1(Integer),
I2cap_Sdu(sign_type), I2cap_rou_sdu(I2cap_rou_pdu),
I2cap_send_CO_ID(Integer, Integer,Pid), I2cap_del_CO_ID(Integer,Pid),
I2cap_send_ROU_ID(Pid), I2cap_terminate_CO, LP_ID(Pid),
L2cap_close_CO, I2cap_open_CO,I2cap_reset;
```

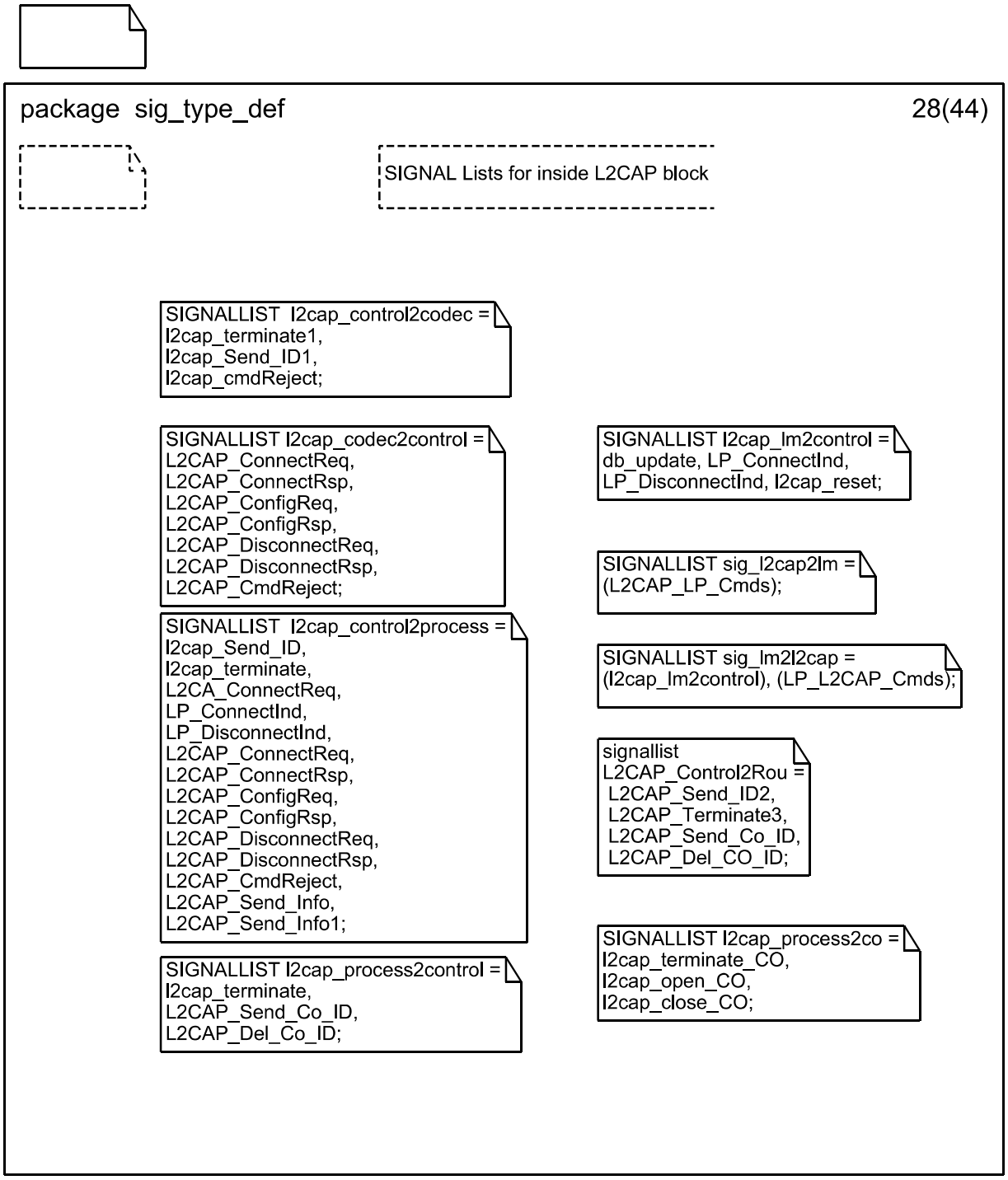

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1059 of 1169

B.7.27 sig_type_def-27



Copyright © 2002 IEEE. All rights reserved.

B.7.28 sig_type_def-28



Copyright © 2002 IEEE. All rights reserved.

B.7.29 sig_type_def-29



package sig_type_def 29(44)



```
newtype sign_type
 choice
  l2pdu1 l2pdu_type1;
  l2pdu2 l2pdu_type2;
  l2pdu3 l2pdu_type3;
  l2pdu4 l2pdu_type4;
  l2pdu5 l2pdu_type5;
  l2pdu6 l2pdu_type6;
  l2pdu7 l2pdu_type7;
endnewtype sign_type;

newtype L2cap_R_tab
  Array(Input_type,Channel_type)
endnewtype;

newtype Input_type
 struct
  PSM PSM_type;
  Add Address_type;
endnewtype;

newtype Channel_type
 struct
  l2cap PId;
  l2_codec Pid;
  l2cap_rou Pid;
  LP PId;
  CID Integer;
  Iden Integer;
endnewtype;

newtype Flow_type struct
    code integer;
    length integer;
    flags integer;
    service_type Service_type;
    t_rate real;
    t_bucket integer;
    P_bwidth integer;
    Lat integer;
    Del_var integer;
endnewtype;
```

```
newtype Reason_type
  literals not_understood, MTU_exceed, invalid_CID;
endnewtype;

newtype Data_type
  literals not_applicable, actual_MTU, requested_CID;
endnewtype;

newtype PSM_type
  literals SDP, RFCOMM, TCP, unknown;
endnewtype;

newtype Result_type
  literals success, pending, refused;
endnewtype;

newtype l2cap_status_type
  literals no_further, pending;
endnewtype;

newtype Flag_type
  literals 0, 1;
endnewtype;

newtype Result_type1
  literals success, pending, refused, timeout;
endnewtype;

newtype Result_type2
   literals success, failure;
endnewtype;

newtype Result_type3
  literals success, failure, timeout;
endnewtype;

newtype Response_type
  literals success, pending, refused;
endnewtype;

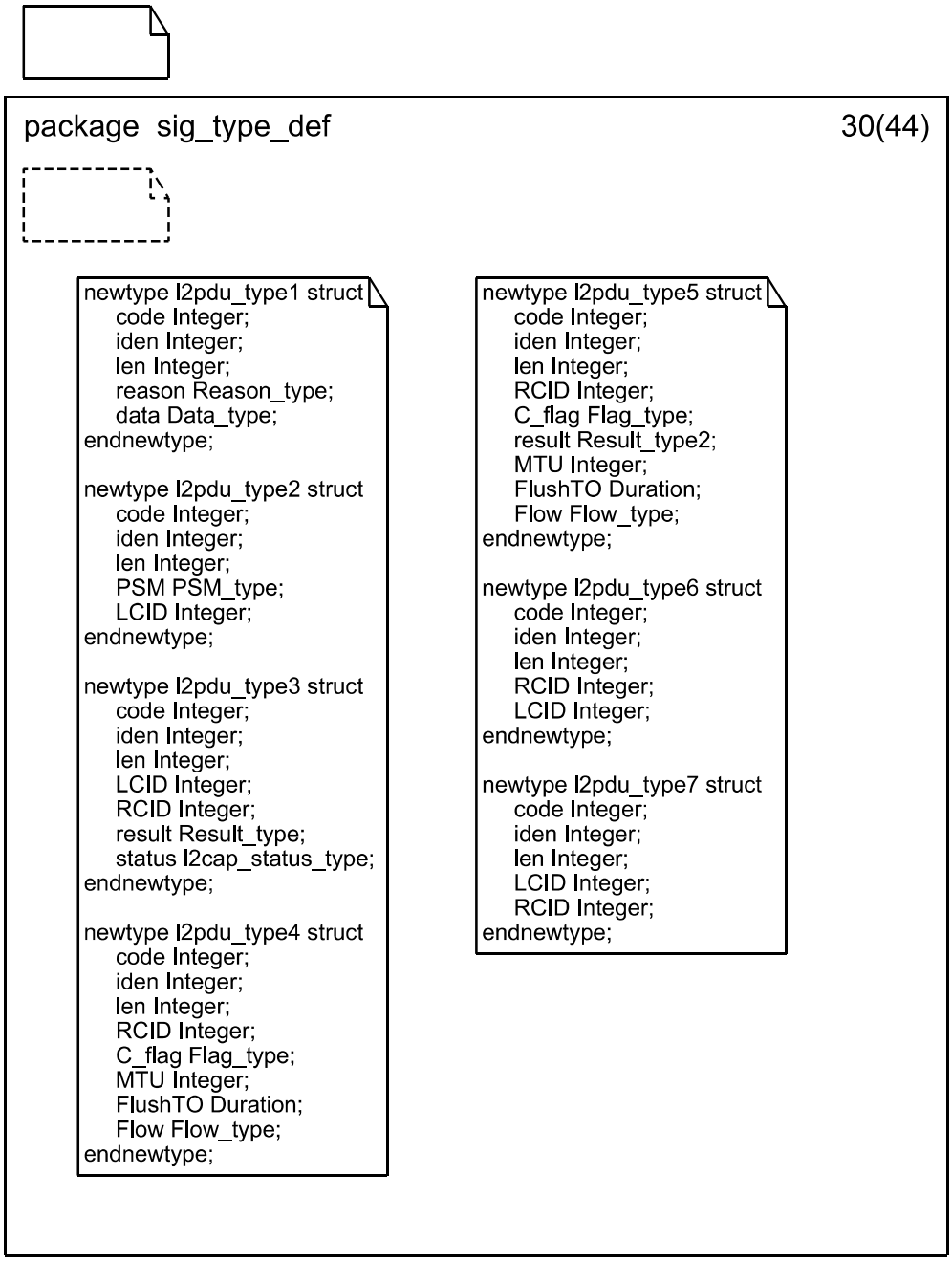
newtype Response_type1
  literals success,failed;
endnewtype;

syntype SmallInt=Integer
   constants 0:10
endsyntype;
```

Copyright © 2002 IEEE. All rights reserved.

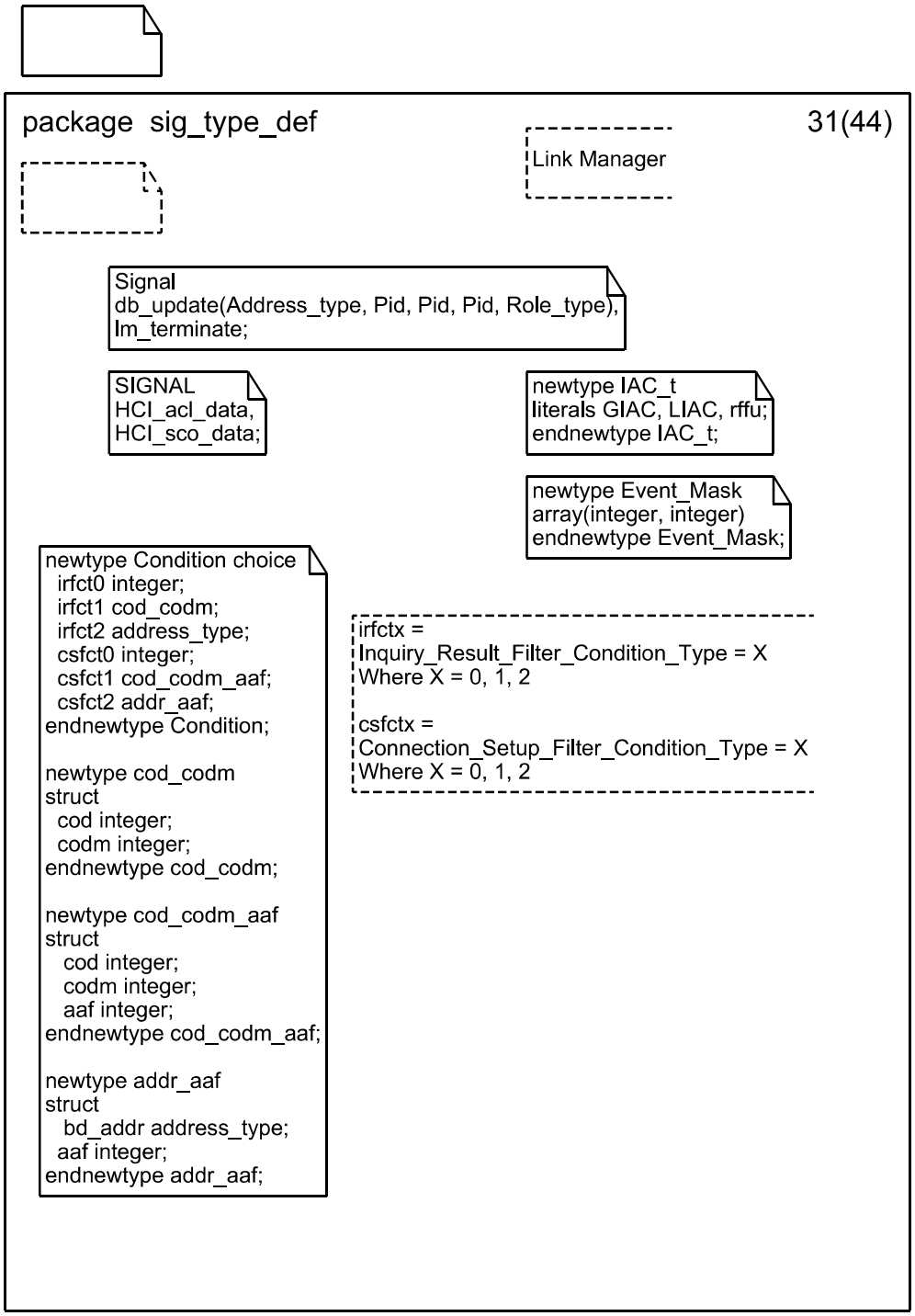
LINKPLAY EXHIBIT 1008
Page 1062 of 1169

B.7.30 sig_type_def-30



Copyright © 2002 IEEE. All rights reserved.

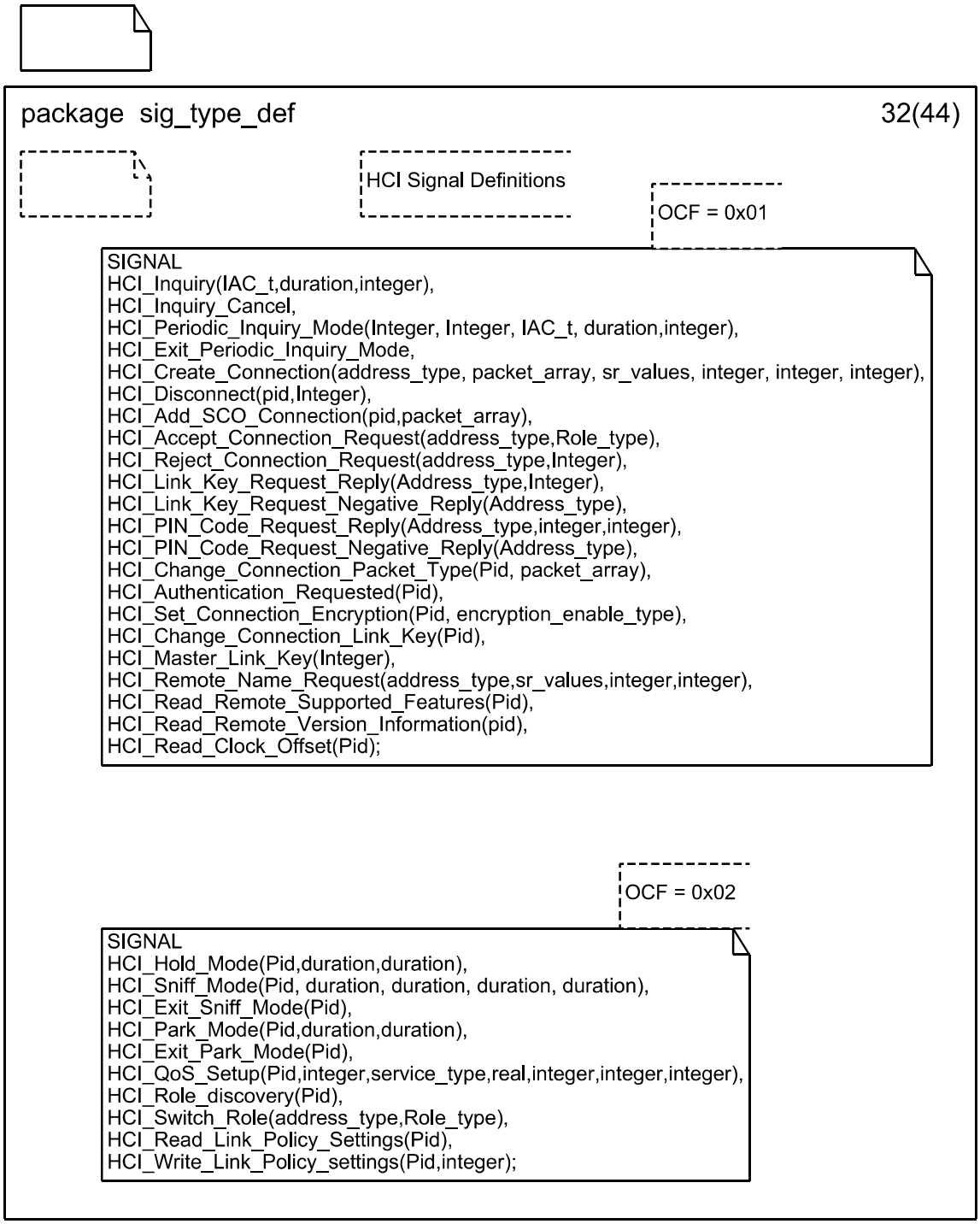
B.7.31 sig_type_def-31



Copyright © 2002 IEEE. All rights reserved.

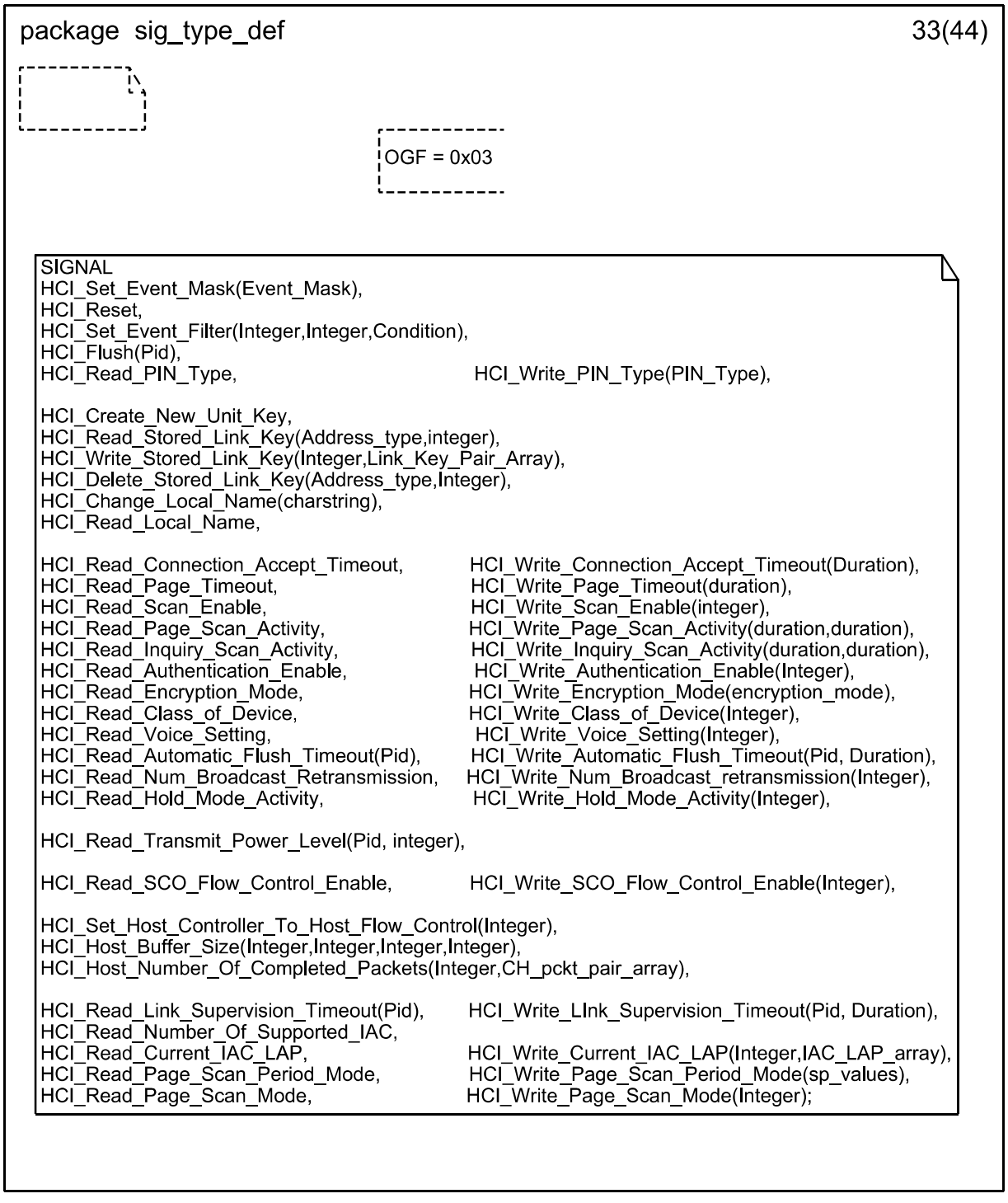
LINKPLAY EXHIBIT 1008
Page 1064 of 1169

B.7.32 sig_type_def-32



Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1065 of 1169

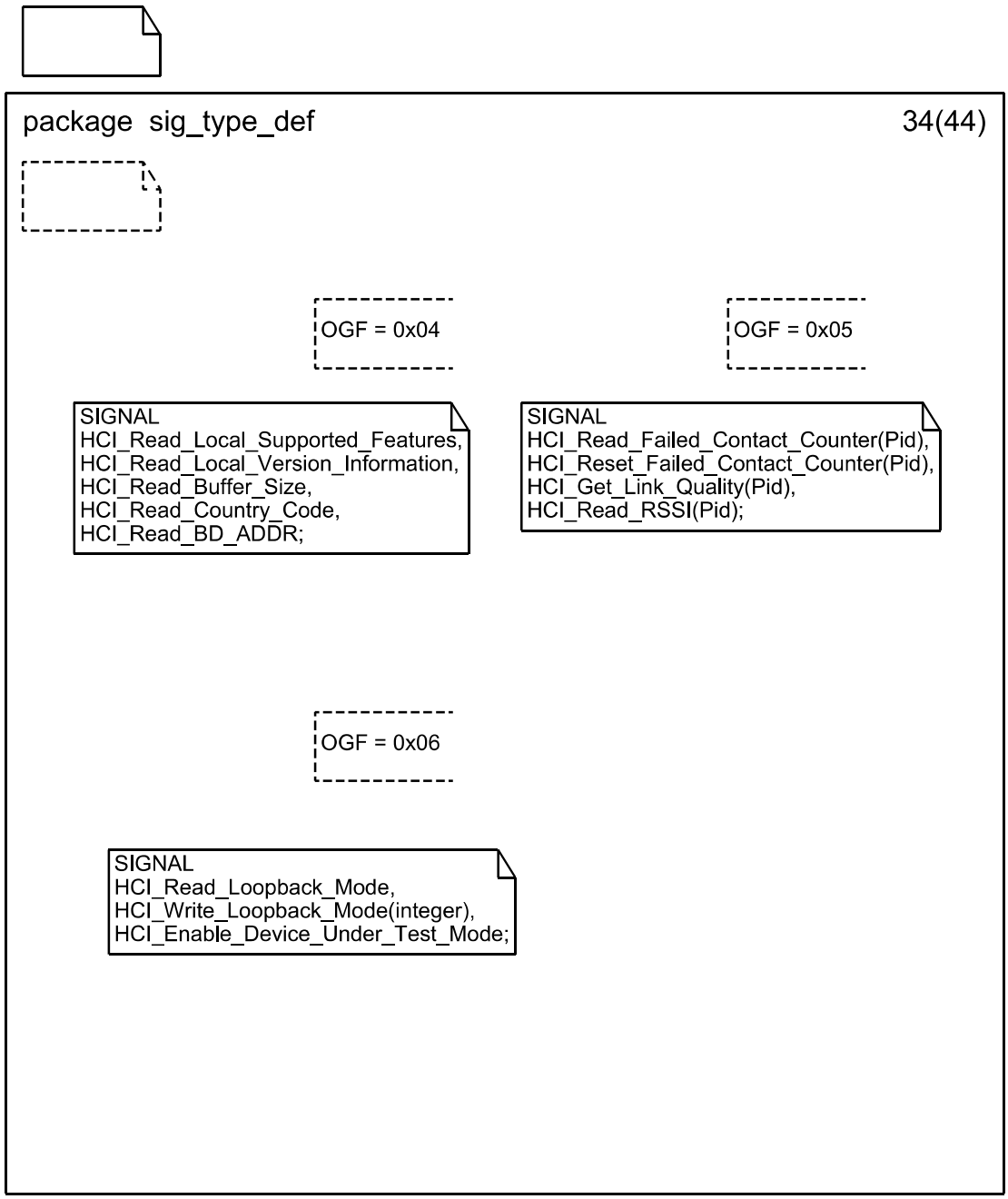
B.7.33 sig_type_def-33



Copyright © 2002 IEEE. All rights reserved.

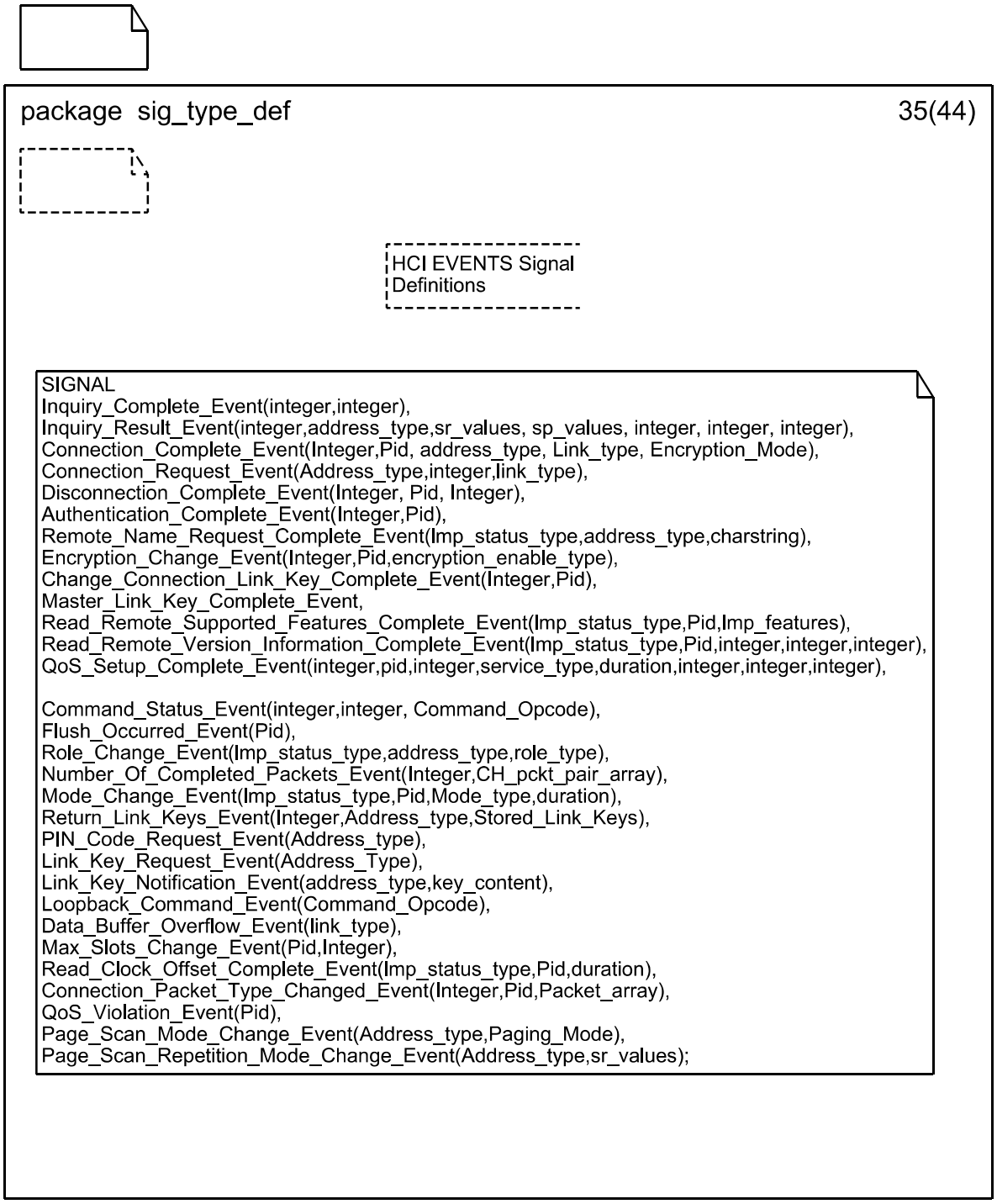
LINKPLAY EXHIBIT 1008
Page 1066 of 1169

B.7.34 sig_type_def-34



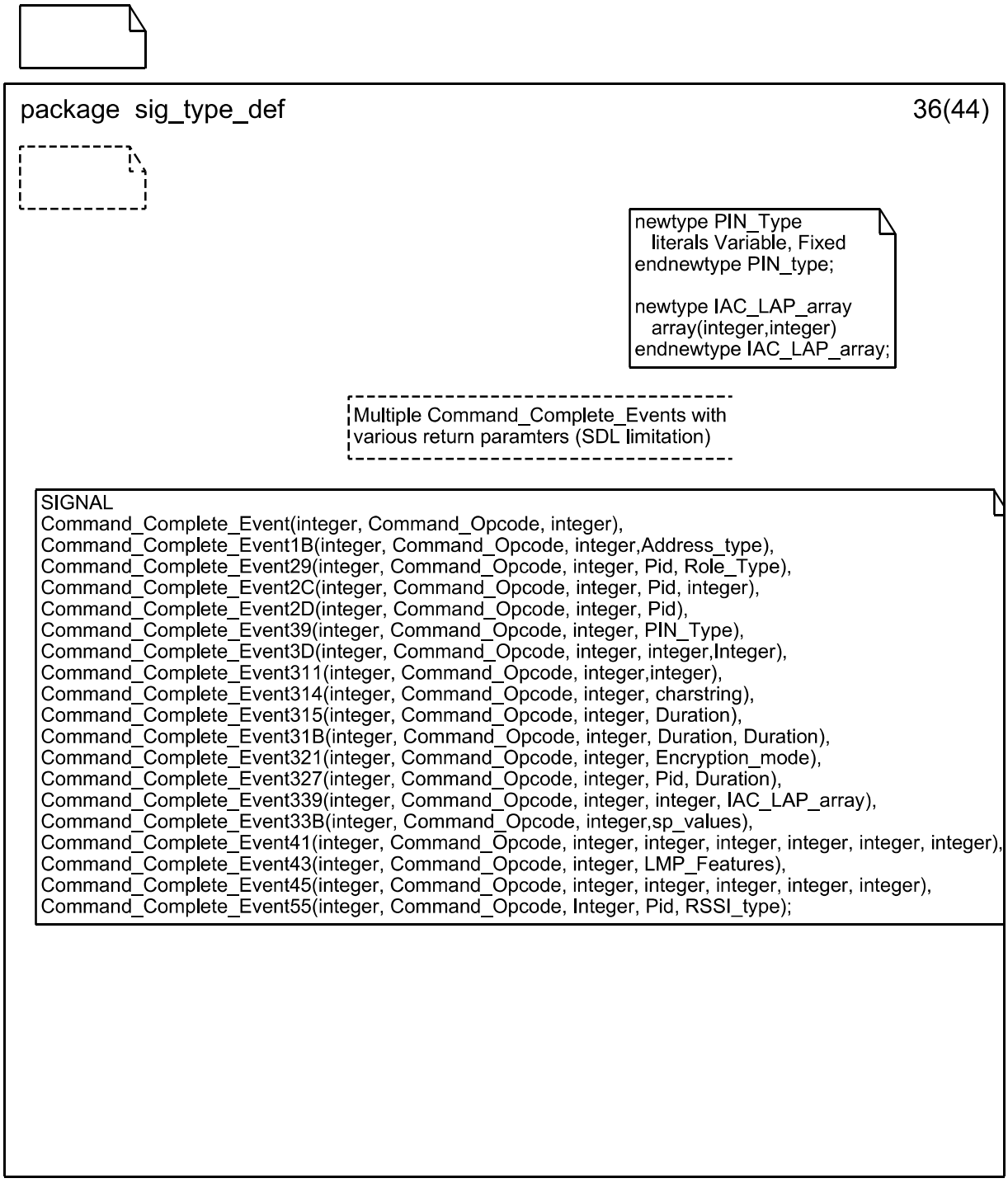
Copyright © 2002 IEEE. All rights reserved.

B.7.35 sig_type_def-35



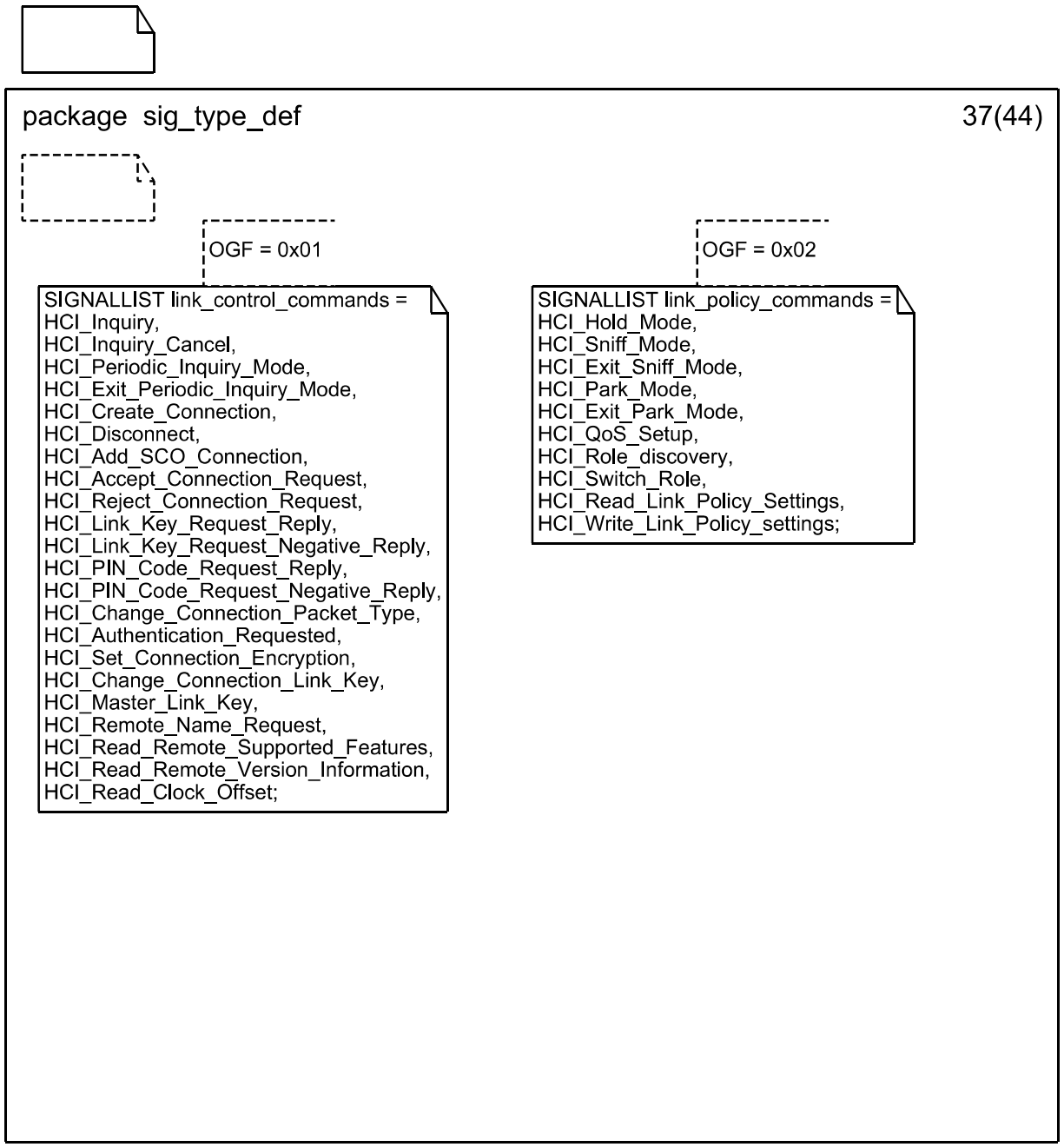
Copyright © 2002 IEEE. All rights reserved.

B.7.36 sig_type_def-36



Copyright © 2002 IEEE. All rights reserved.

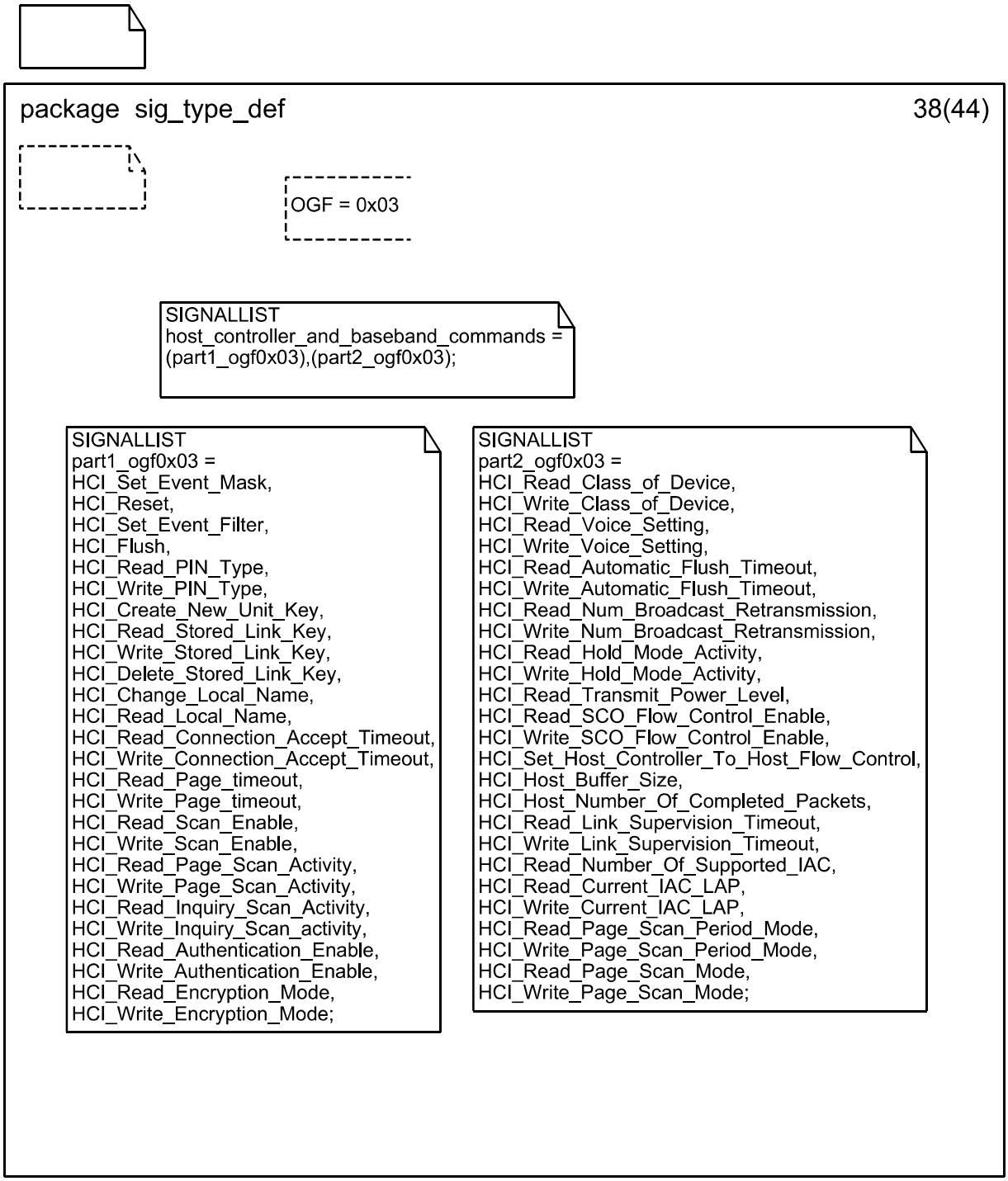
B.7.37 sig_type_def-37



Copyright © 2002 IEEE. All rights reserved.

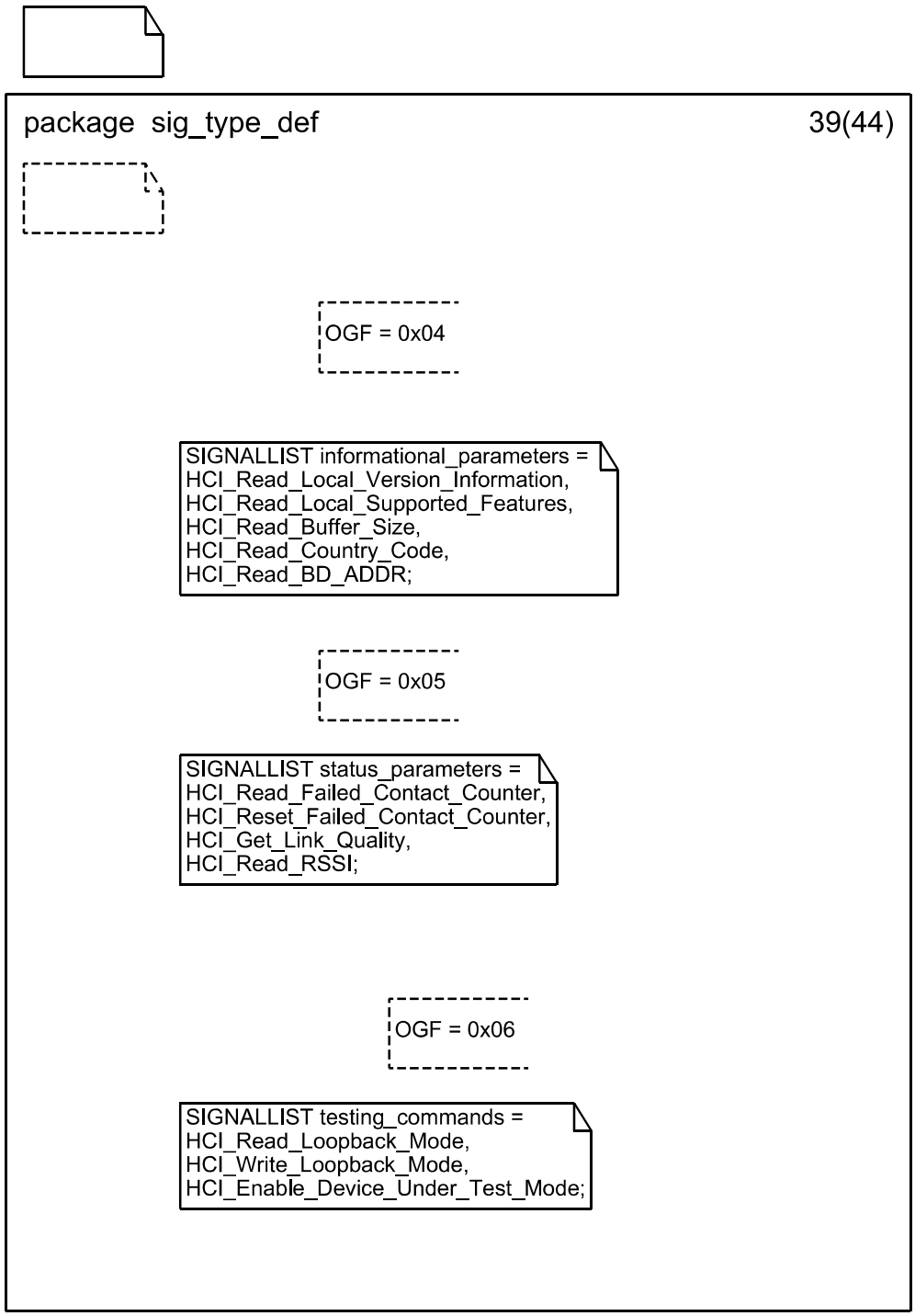
LINKPLAY EXHIBIT 1008
Page 1070 of 1169

B.7.38 sig_type_def-38



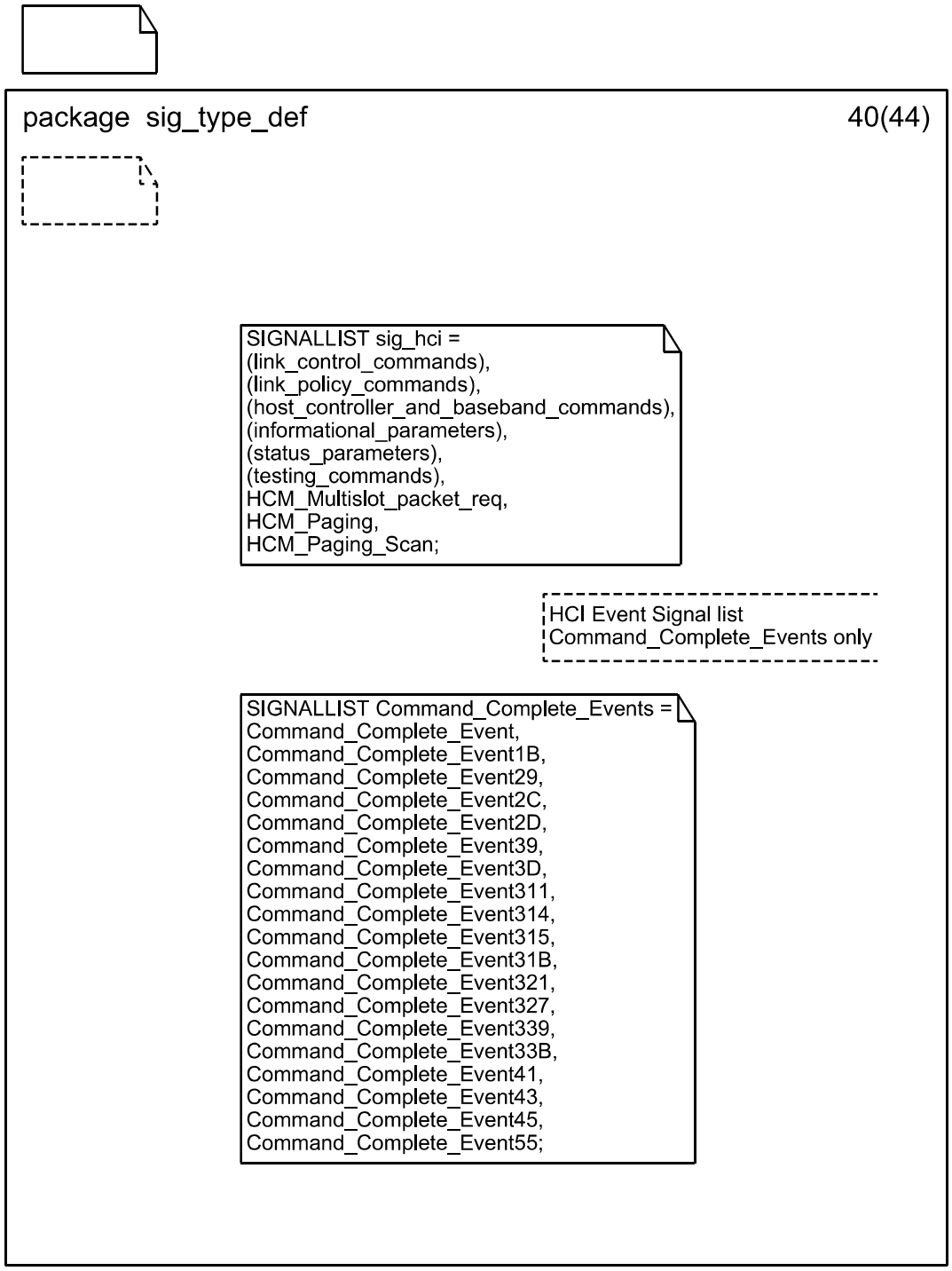
Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1071 of 1169

B.7.39 sig_type_def-39



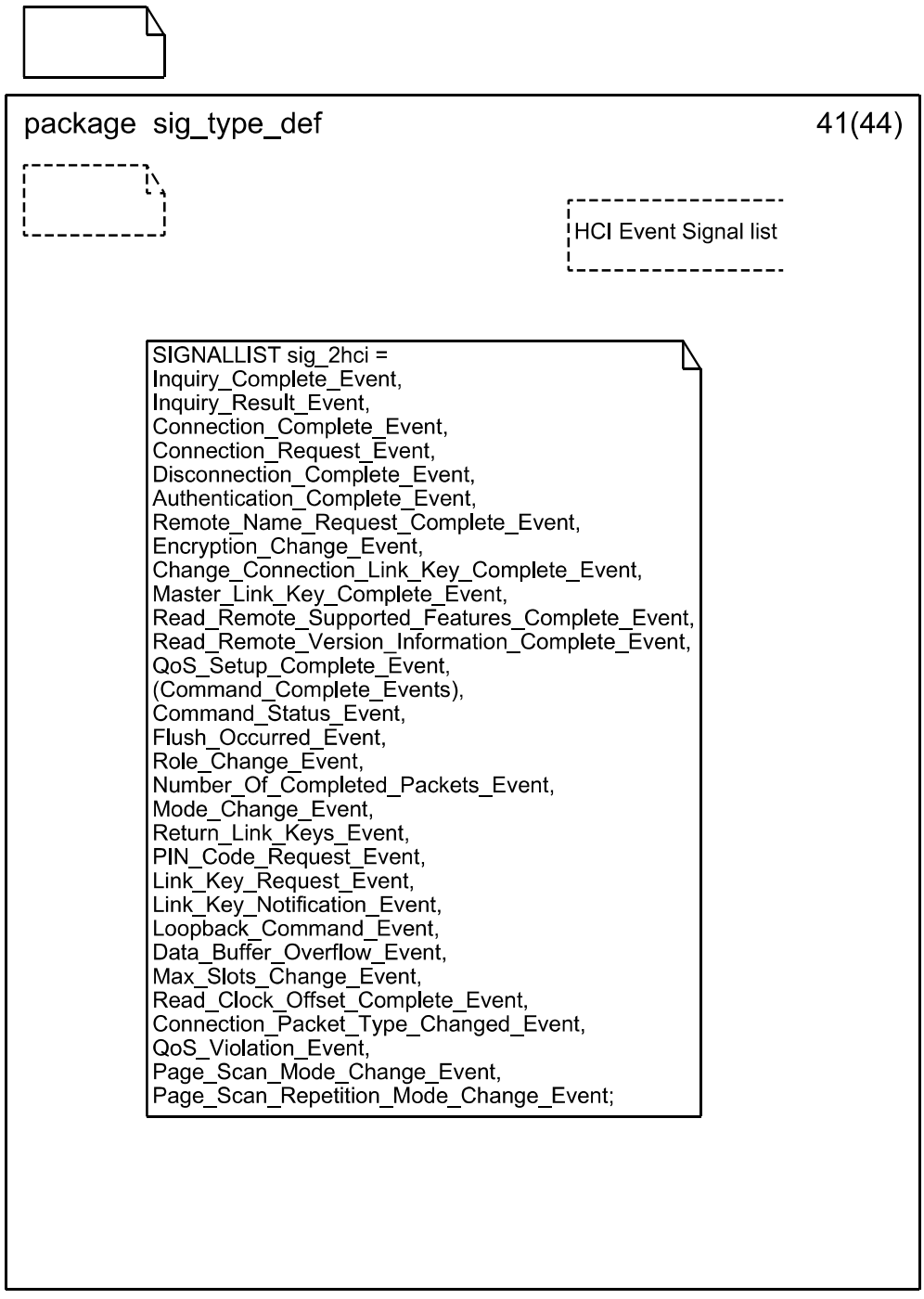
Copyright © 2002 IEEE. All rights reserved.

B.7.40 sig_type_def-40



Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1073 of 1169

B.7.41 sig_type_def-41



Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1074 of 1169

B.7.42 sig_type_def-42



package sig_type_def 42(44)



HCI Error Codes

```
synonym success Integer = 0 ,
 unknown_HCI_cmd integer = 1 ,
 no_connection integer = 2 ,

 page_timeout integer = 4 ,
 authentication_failure integer = 5 ,
 key_missing integer = 6 ,
 memory_full integer = 7 ,
 connection_timeout integer = 8 ,
 max_connections integer = 9 ,
 max_SCO_connections integer = 10 ,
 ACL_already_exists integer = 11 ,

 host_timeout integer = 16 ,
 unsupported_fe_pa_value integer = 17 ,
 invalid_HCI_pa integer = 18 ,

 connection_terminate integer = 22 ,
 repeated_attempts integer = 23 ,
 pairing_not_allowed integer = 24 ,
 unknown_LMP_PDU integer = 25 ,
 unsupported_feature integer = 26 ,


 unspecified_error integer = 31 ,
 unsupported_value integer = 32 ,
 switch_not_allowed integer = 33 ,

 transaction_collision integer = 35 ,
 PDU_not_allowed integer = 36 ,
 not_agree_encryption integer = 80 ,
 unit_key_used integer = 81;
```

```
These error codes are not
used.
================
 hardware_failure integer = 3 ,

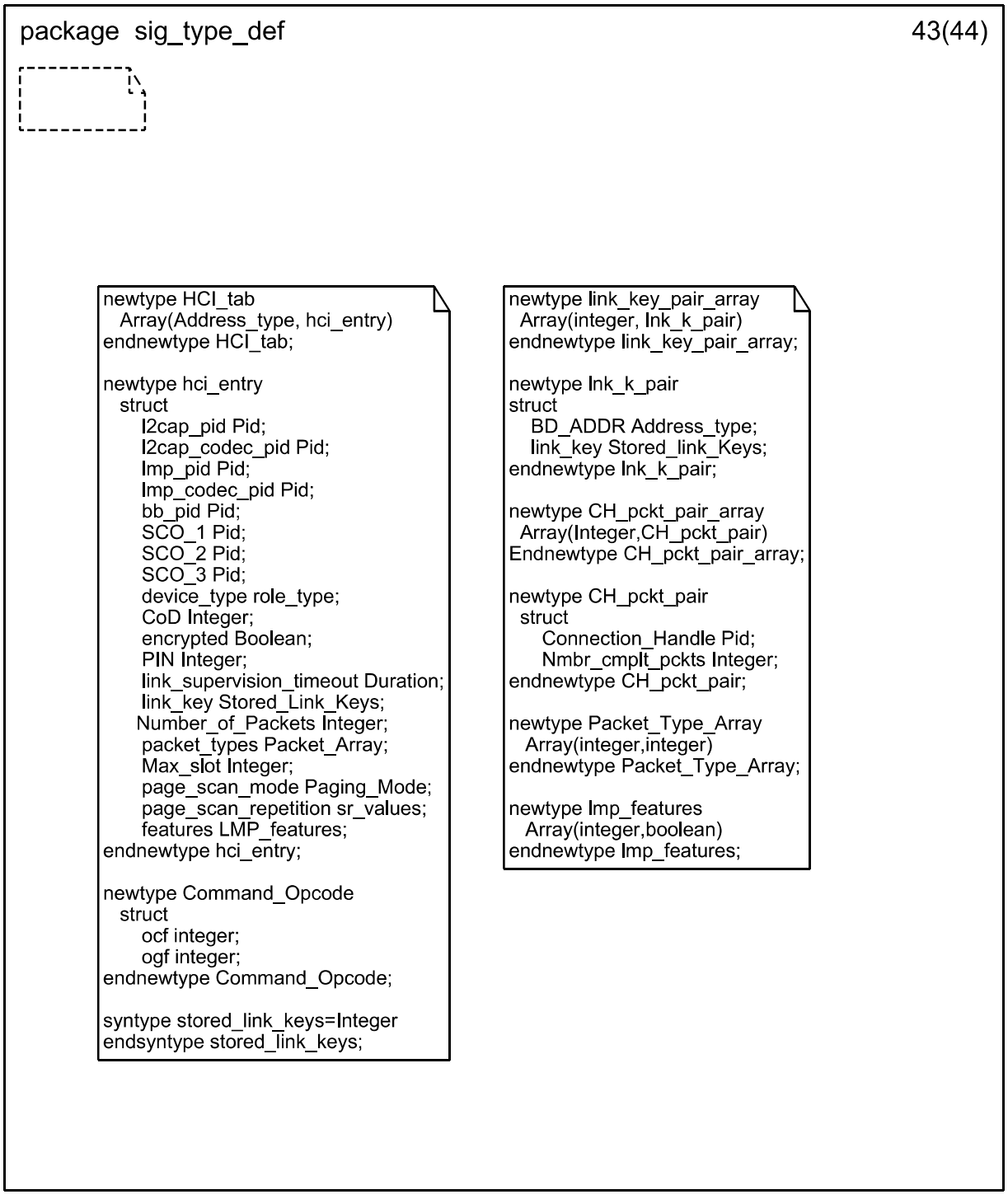
 Cmd_disallowed integer = 12 ,
 limited_resources integer = 13 ,
 security_reasons integer = 14 ,
 personal_device integer = 15 ,

 user_ended_connection integer = 19 ,
 low_resources integer = 20 ,
 about_to_power_off integer = 21 ,

 SCO_offset integer = 27 ,
 SCO_interval integer = 28 ,
 SCO_air_mode integer = 29 ,
 invalid_parameters integer = 30 ,
 LMP_response_timeout integer = 34 ,
```

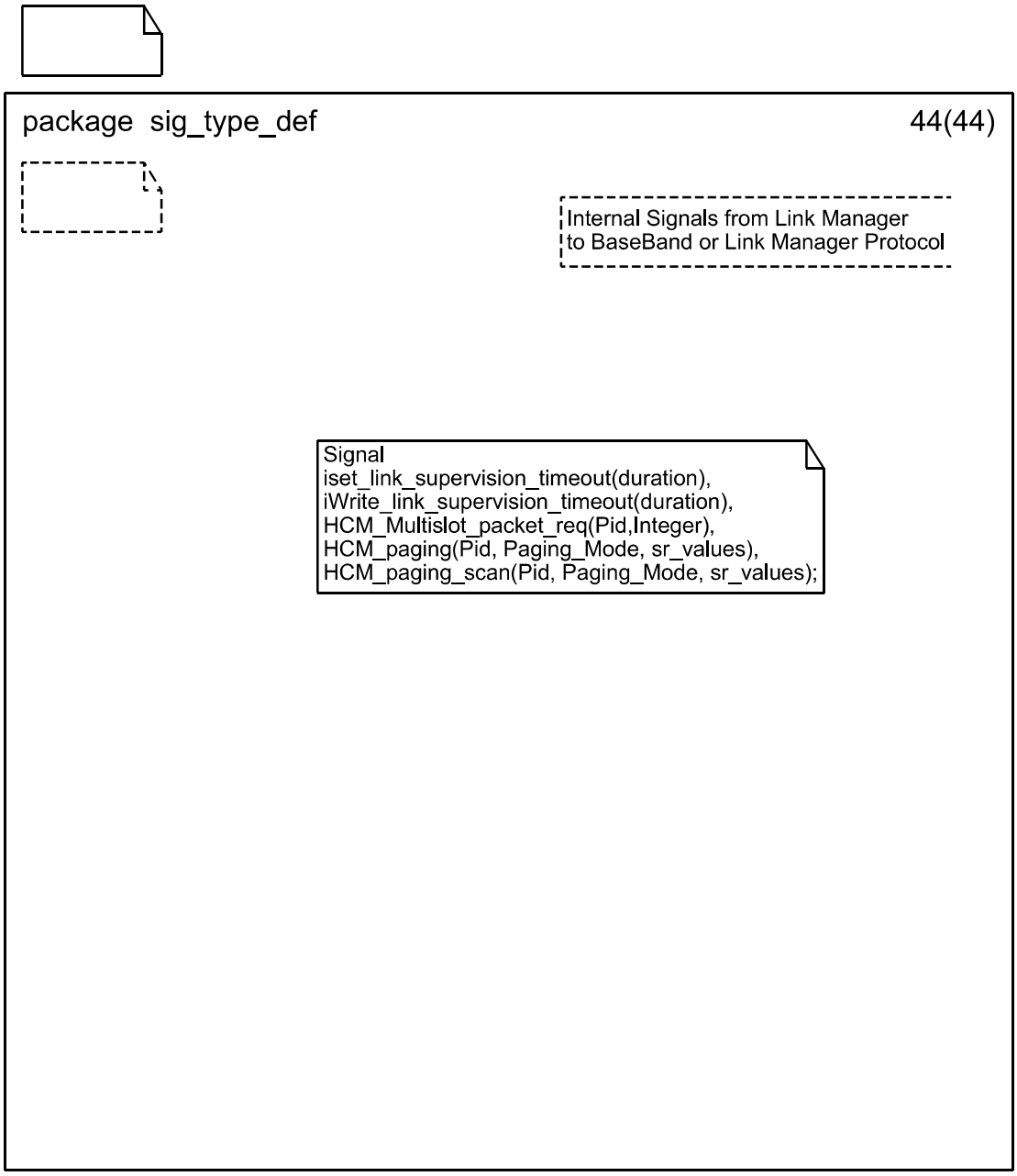

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1075 of 1169

B.7.43 sig_type_def-43



Copyright © 2002 IEEE. All rights reserved.

B.7.44 sig_type_def-44



Copyright © 2002 IEEE. All rights reserved.

Annex C

(normative)

Generic access profile (GAP)

C.1 Introduction

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

Interoperability between devices from different manufacturers is provided for a specific service and use case, if the devices conform to a Bluetooth-SIG-defined profile specification. A profile defines a selection of messages and procedures (generally termed *capabilities*) from the Bluetooth SIG specifications (see 2.4) and gives an unambiguous description of the air interface for specified services and use cases.

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

C.1.1 Scope

The purpose of the GAP is to

— Introduce definitions, recommendations, and common requirements related to modes and access procedures that are to be used by transport and application profiles.
— Describe how devices are to behave in standby and connecting states in order to guarantee that links and channels always can be established between Bluetooth devices and that multiprofile operation is possible. Special focus is put on discovery, link establishment, and security procedures.
— State requirements on user interface aspects, mainly coding schemes and names of procedures and parameters, that are needed to guarantee a satisfactory user experience.

C.1.2 Symbols and conventions

C.1.2.1 Requirement status symbols

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

M	Mandatory to support (used for capabilities that shall be used in the profile)
O	Optional to support (used for capabilities that can be used in the profile)
C	Conditional support (used for capabilities that shall be used in case a certain other capability is supported)
X	Excluded (used for capabilities that may be supported by the unit, but shall never be used in the profile)
N/A	Not applicable (in the given context it is impossible to use this capability)

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1078 of 1169

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

C.1.2.2 Signalling diagram conventions

Figure C.1 shows how arrows are used in diagrams describing procedures.

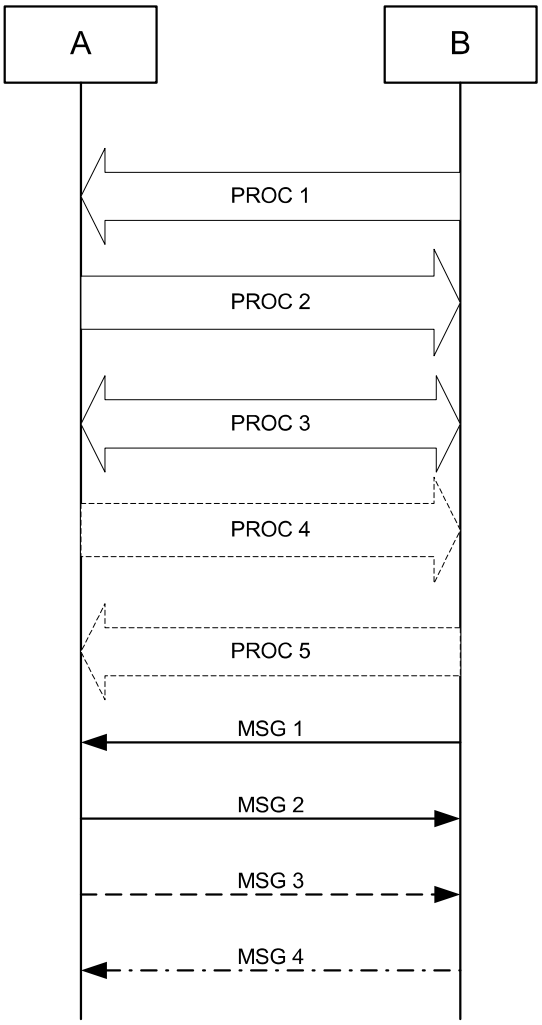


Figure C.1—Arrows used in signalling diagrams

In Figure C.1, the following cases are shown: PROC1 is a subprocedure initiated by device B. PROC2 is a subprocedure initiated by device A. PROC3 is a subprocedure where the initiating side is undefined (may be both device A or device B). Dashed arrows denote optional steps. PROC4 indicates an optional subprocedure initiated by device A, and PROC5 indicates an optional subprocedure initiated by device B.

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

Copyright © 2002 IEEE. All rights reserved.

C.1.2.3 Notation for timers and counters

Timers are introduced specific to this profile. To distinguish them from timers used in the Bluetooth protocol specifications and other profiles, these timers are named in the following format: $T_{GAP}(nnn)$.

C.2 Profile overview

C.2.1 Profile stack

The main purpose of this profile is to describe the use of the lower layers of the Bluetooth protocol stack (i.e., LC and LMP). To describe security-related alternatives, higher layers (L2CAP, RFCOMM, and OBEX) are also included. See Figure C.2.

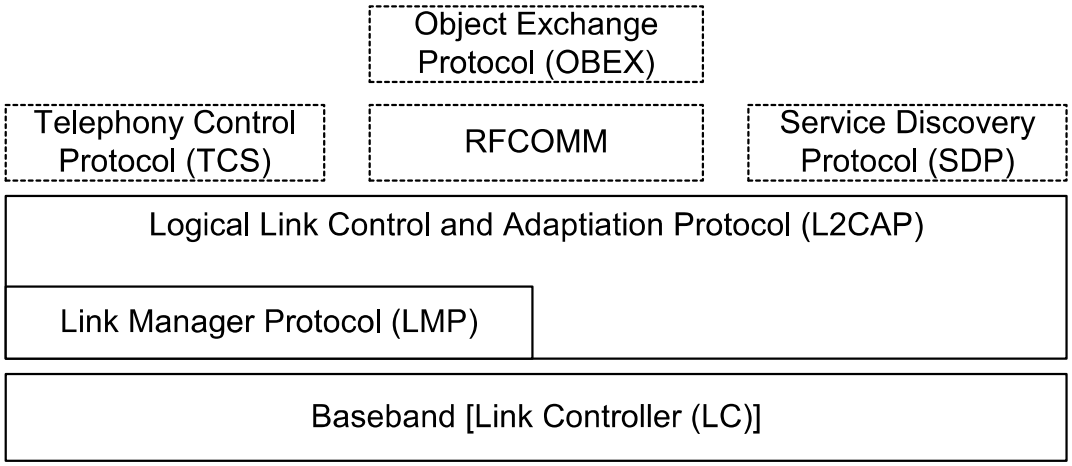


Figure C.2—Profile stack covered by this profile

C.2.2 Configurations and roles

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

This profile handles the procedures between two devices related to discovery and connecting (link and connection establishment) for the case where none of the two devices has any link established as well as for the case where (at least) one device has a link established (possibly to a third device) before starting the described procedure. Figure C.3 illustrates this profile, which covers procedures initiated by one device (A) towards another device (B), which may or may not have an existing Bluetooth link active.

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

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1080 of 1169

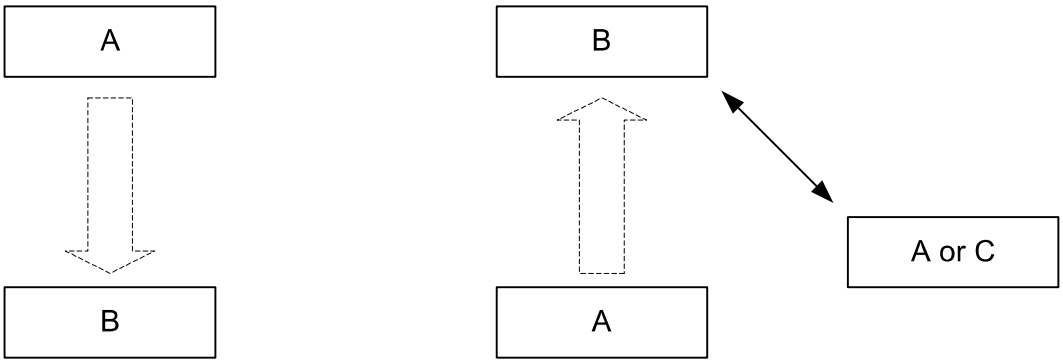


Figure C.3—This profile covers procedures initiated by one device (A) toward another device (B), which may or may not have an existing Bluetooth link active

C.2.3 User requirements and scenarios

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

C.2.4 Profile fundamentals

The GAP provides the following fundamental functions:

- States the requirements on names, values, and coding schemes used for names of parameters and procedures experienced on the user interface level.
- Defines modes of operation that are not service- or profile-specific, but that are generic and can be used by profiles referring to this profile and by devices implementing multiple profiles.
- Defines the general procedures that can be used for discovering identities, names, and basic capabilities of other Bluetooth devices that are in a mode where they can be discoverable. Only procedures where no channel or connection establishment is used are specified.
- Defines the general procedure for how to create bonds (i.e., dedicated exchange of link keys) between Bluetooth devices.
- Describes the general procedures that can be used for establishing connections to other Bluetooth devices that are in a mode that allows them to accept connections and service requests.

C.2.5 Conformance

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

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

Copyright © 2002 IEEE. All rights reserved.

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

C.3 User interface aspects

C.3.1 The user interface level

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

This profile specifies the generic terms that should be used on the user interface level.

C.3.2 Representation of Bluetooth parameters

C.3.2.1 BD_ADDR

C.3.2.1.1 Definition

BD_ADDR is the unique address of a Bluetooth device as defined in Clause 8. It is received from a remote device during the device discovery procedure.

C.3.2.1.2 Term on the user interface level

When the Bluetooth address is referred to on the UI level, the term *Bluetooth device address* should be used.

C.3.2.1.3 Representation

On the BB level, the BD_ADDR is represented as 48 bits (see Clause 8).

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

C.3.2.2 Bluetooth device name (the user-friendly name)

C.3.2.2.1 Definition

The Bluetooth device name is the user-friendly name with which a Bluetooth device presents itself. It is a character string returned in the LMP_name_res as a response to a LMP_name_req.

C.3.2.2.2 Term on the user interface level

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

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1082 of 1169

C.3.2.2.3 Representation

The Bluetooth device name can be up to 248 bytes maximum according to Clause 9. It shall be encoded according to UTF-8 (i.e., a name entered on the UI level may be down to 82 characters outside the Unicode range 0x00–0x7F are used).

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

C.3.2.3 Bluetooth passkey (Bluetooth PIN)

C.3.2.3.1 Definition

The Bluetooth PIN is used to authenticate two Bluetooth devices (that have not previously exchanged link keys) to each other and to create a trusted relationship between them. The PIN is used in the pairing procedure (see Annex C.9.2) to generate the initial link key that is used for further authentication.

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

C.3.2.3.2 Terms at the user interface level

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

C.3.2.3.3 Representation

The Bluetooth PIN has different representations on different levels. PIN_{BB} is used on the baseband level, and PIN_{UI} is used on the UI level.

PIN_{BB} is the PIN used for calculating the initialization key during the pairing procedure (see Clause 8). PIN_{UI} is the character representation of the PIN that is entered on the UI level. The transformation from PIN_{UI} to PIN_{BB} shall be according to UTF-8, and decimal digits shall be within the Unicode range 0x00–0x7F.

According to Clause 8, PIN_{BB} can be 128 bits (16 bytes). In other words, if a device supports entry of characters outside the Unicode range 0x00–0x7F, the maximum number of characters in the PIN_{UI} may be less than 16. For examples, see Table C.1.

Table C.1—Representation examples

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

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

Copyright © 2002 IEEE. All rights reserved.

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

C.3.2.4 Class of device

C.3.2.4.1 Definition

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

C.3.2.4.2 Term on the user interface level

The information within the Class of Device parameter should be referred to as *Bluetooth device class* (i.e., the major and minor device class fields) and *Bluetooth service type* (i.e., the service class field). The terms for the defined Bluetooth device types and Bluetooth service types are defined in the Bluetooth assigned numbers specification (see 2.4.3).

When using a mix of information found in the Bluetooth device class and the Bluetooth service type, the term *Bluetooth device type* should be used.

C.3.2.4.3 Representation

The class of device is a bit field and is defined in the Bluetooth assigned numbers specification (see 2.4.3). The UI-level representation of the information in the class of device is implementation-specific.

C.3.3 Pairing

Two procedures are defined that make use of the pairing procedure defined on the LMP level (i.e., LMP-pairing; see Annex C.9.2). Either the user initiates the bonding procedure and enters the passkey with the explicit purpose of creating a bond (and maybe also a secure relationship) between two Bluetooth devices, or the user is requested to enter the passkey during the establishment procedure since the devices did not share a common link key beforehand. In the first case, the user is said to perform “bonding (with entering of passkey)” and in the second case the user is said to “authenticate using the passkey.”

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1084 of 1169

C.4 Modes

The conformance requirements related to the modes defined in C.4 are summarized in Table C.2.

Table C.2—Conformance requirements related to modes defined in C.4

Conformance requirements	Procedure	Reference	Support[a]
1	Discoverability modes	C.4.1	
	Nondiscoverable mode		C1
	Limited discoverable mode		O
	General discoverable mode		O
2	Connectability modes	C.4.1.3.3	
	Nonconnectable mode		O
	Connectable mode		M
3	Pairing modes	C.4.2.2.2	
	Nonpairable mode		O
	Pairable mode		C2

[a]See C.1.2.1 for the definitions of the symbols used in this column.

C1: If limited discoverable mode is supported, nondiscoverable mode is mandatory, otherwise, optional.
C2: If the bonding procedure is supported, support for pairable mode is mandatory, otherwise, optional.

C.4.1 Discoverability modes

With respect to inquiry, a Bluetooth device shall be either in nondiscoverable mode or in a discoverable mode. (The device shall be in one, and only one, discoverability mode at a time.) The two discoverable modes defined in this subclause are called limited discoverable mode and general discoverable mode. Inquiry is defined in Clause 8.

When a Bluetooth device is in nondiscoverable mode, it does not respond to inquiry.

A Bluetooth device is said to be made discoverable, or set into a discoverable mode, when it is in limited discoverable mode or in general discoverable mode. Even when a Bluetooth device is made discoverable, it may be unable to respond to inquiry due to other baseband activity (see Clause 8). A Bluetooth device that does not respond to inquiry for any of these two reasons is called a silent device.

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

C.4.1.1 Nondiscoverable mode

C.4.1.1.1 Definition

When a Bluetooth device is in nondiscoverable mode, it shall never enter the INQUIRY_RESPONSE state.

Copyright © 2002 IEEE. All rights reserved.

C.4.1.1.2 Term on the UI level

Bluetooth device is “nondiscoverable” or in “nondiscoverable mode”.

C.4.1.2 Limited discoverable mode

C.4.1.2.1 Definition

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

A Bluetooth device should not be in limited discoverable mode for more than $T_{GAP}(104)$. The scanning for the LIAC can be done either in parallel or in sequence with the scanning of the GIAC. When in limited discoverable mode, one of the options described in C.4.1.2.1.1 or C.4.1.2.1.2 shall be used.

C.4.1.2.1.1 Parallel scanning

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

C.4.1.2.1.2 Sequential scanning

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

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

C.4.1.2.2 Conditions

When a device is in limited discoverable mode it shall set bit number 13 in the Major Service Class part of the Class of Device/Service field in 2.4.3.

C.4.1.2.3 Term on UI level

Bluetooth device is “discoverable” or in “discoverable mode”.

C.4.1.3 General discoverable mode

C.4.1.3.1 Definition

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

C.4.1.3.2 Conditions

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

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

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1086 of 1169

C.4.1.3.3 Term on UI level

Bluetooth device is “discoverable” or in “discoverable mode”.

C.4.2 Connectability modes

With respect to paging, a Bluetooth device shall be either in nonconnectable mode or in connectable mode. Paging is defined in Clause 8.

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

C.4.2.1 Nonconnectable mode

C.4.2.1.1 Definition

When a Bluetooth device is in nonconnectable mode, it shall never enter the PAGE_SCAN state.

C.4.2.1.2 Term on UI level

Bluetooth device is “nonconnectable” or in “nonconnectable mode”.

C.4.2.2 Connectable mode

C.4.2.2.1 Definition

When a Bluetooth device is in connectable mode, it shall periodically enter the PAGE_SCAN state.

C.4.2.2.2 Term on UI level

Bluetooth device is “connectable” or in “connectable mode”.

C.4.3 Pairing modes

With respect to pairing, a Bluetooth device shall be either in nonpairable mode or in pairable mode. In pairable mode the Bluetooth device accepts pairing, i.e., creation of bonds, initiated by the remote device, and in nonpairable mode it does not. Pairing is defined in Clauses 8 and 9.

C.4.3.1 Nonpairable mode

C.4.3.1.1 Definition

When a Bluetooth device is in nonpairable mode it shall respond to a received LMP_in_rand with LMP_not_accepted with the reason *pairing not allowed*.

C.4.3.1.2 Term on UI level

Bluetooth device is “nonbondable” or in “nonbondable mode” or “does not accept bonding”.

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1087 of 1169

C.4.3.2 Pairable mode

C.4.3.2.1 Definition

When a Bluetooth device is in pairable mode, it shall respond to a received LMP_in_rand with LMP_accepted (or with LMP_in_rand if it has a fixed PIN).

C.4.3.2.2 Term on UI level

Bluetooth device is “bondable” or in “bondable mode” or “accepts bonding”.

C.5 Security aspects

Table C.3 describes the conformance requirements related to the generic authentication procedure and the security modes defined in C.5.

Table C.3—Conformance requirements related to the generic authentication procedure and the security modes defined in C.5

Conformance requirements	Procedure	Reference	Support
1	Authentication	C.5.1	C1
2	Security modes	C.5.2	
	Security mode 1		O
	Security mode 2		C2
	Security mode 3		C2
C1: If security mode 1 is the only security mode that is supported, support for authentication is optional, otherwise mandatory. (NOTE—Support for LMP-authentication and LMP-pairing is mandatory according to Clause 9, independent of which security mode that is used.) C2: If security mode 1 is not the only security mode that is supported, then support for at least one of security mode 2 or security mode 3 is mandatory.			

C.5.1 Authentication

C.5.1.1 Purpose

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

C.5.1.2 Term on UI level

Bluetooth authentication.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1088 of 1169

C.5.1.3 Procedure

Figure C.4 provides a flow chart that defines the generic authentication procedure.

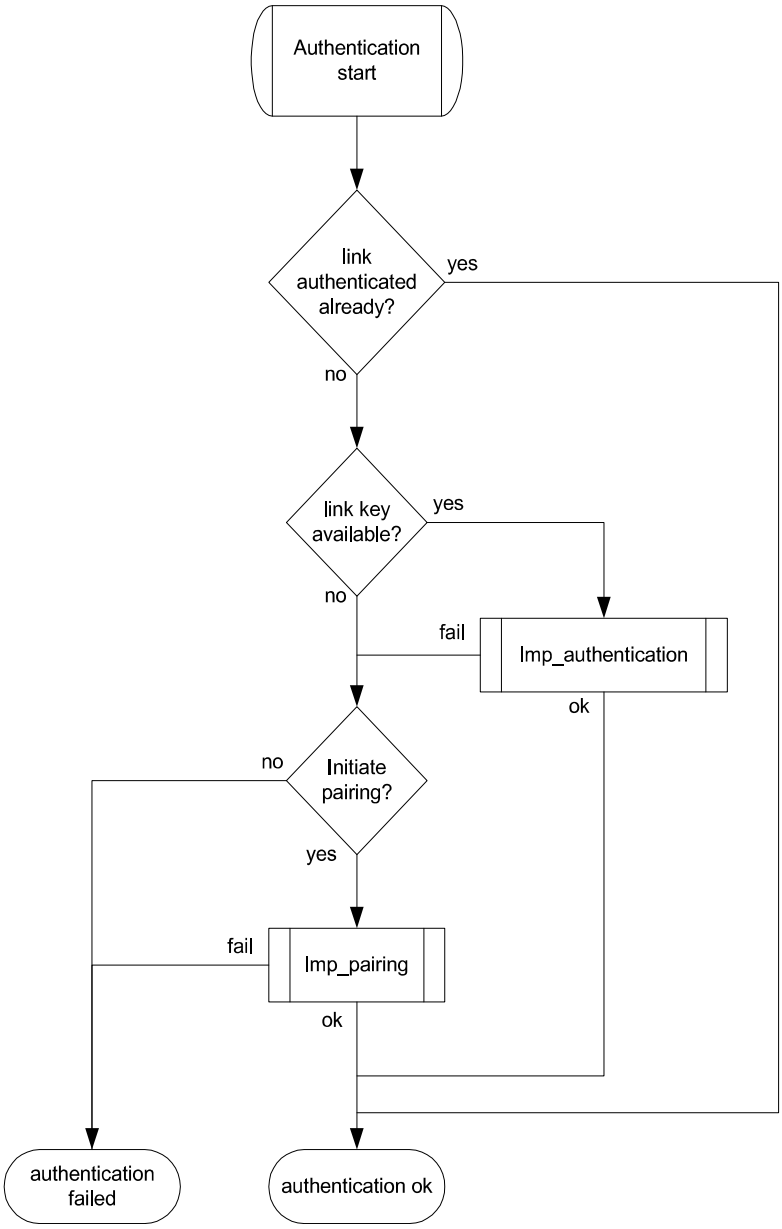


Figure C.4—Definition of generic authentication procedure

C.5.1.4 Conditions

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

C.5.2 Security modes

The flowchart in Figure C.5 describes where in the channel establishment procedures initiation of authentication takes place, depending on which security mode the Bluetooth device is in.

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1089 of 1169

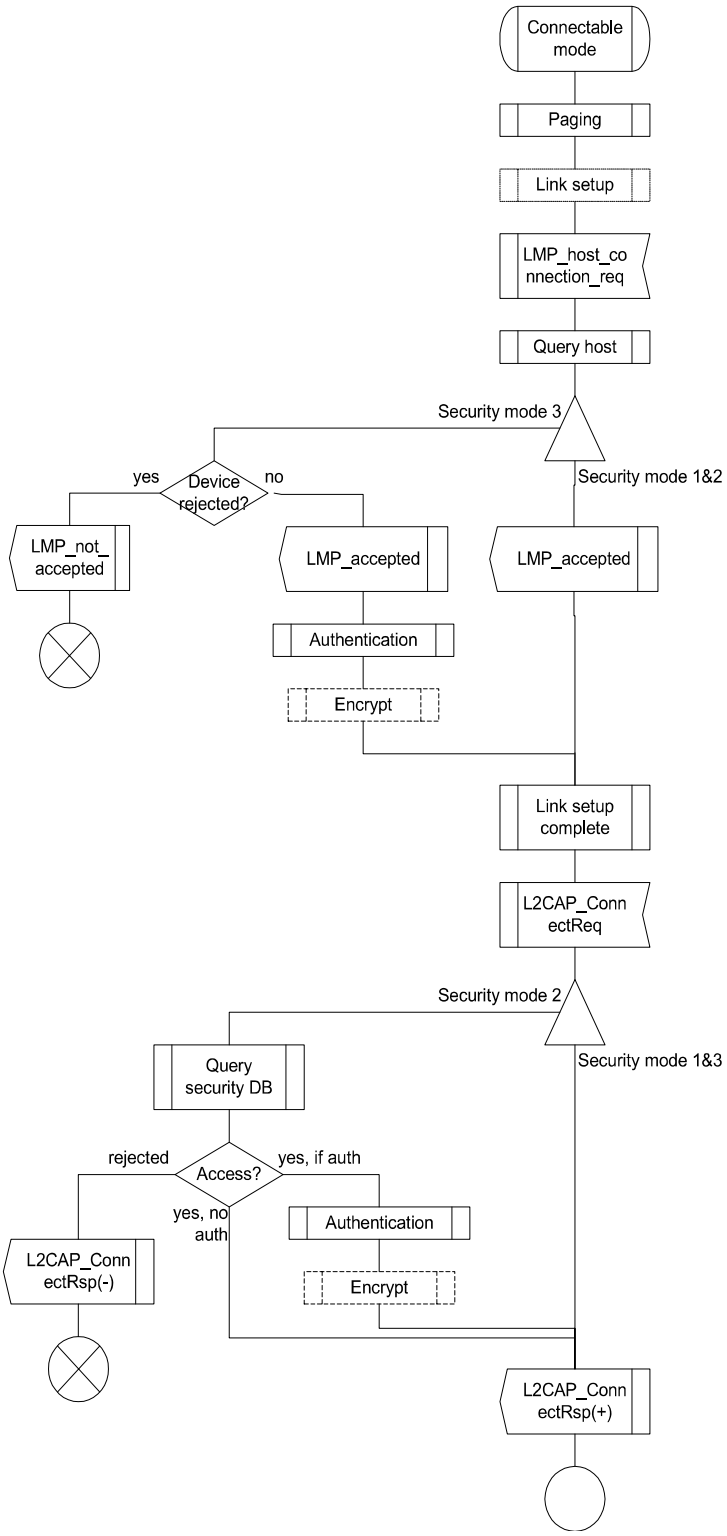


Figure C.5—Illustration of channel establishment using different security modes

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1090 of 1169

When authentication is initiated toward a Bluetooth device, it shall act according to Clause 9 and the current pairing mode, independent of which security mode it is in.

C.5.2.1 Security mode 1 (nonsecure)

When a Bluetooth device is in security mode 1, it shall never initiate any security procedure (i.e., it shall never send LMP_au_rand, LMP_in_rand, or LMP_encryption_mode_req).

C.5.2.2 Security mode 2 (service level enforced security)

When a Bluetooth device is in security mode 2, it shall not initiate any security procedure before a channel establishment request (L2CAP_ConnectReq) has been received or a channel establishment procedure has been initiated by itself. Whether a security procedure is initiated or not depends on the security requirements of the requested channel or service.

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

— Authorization required

— Authentication required

— Encryption required

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

C.5.2.3 Security mode 3 (link level enforced security)

When a Bluetooth device is in security mode 3, it shall initiate security procedures before it sends LMP_link_setup_complete. (The behavior of a device in security mode 3 is as described in Clause 9, Link Manager Protocol.)

A Bluetooth device in security mode 3 may reject the host connection request (respond with LMP_not_accepted to the LMP_host_connection_req) based on settings in the host (e.g., only communication with pre-paired devices allowed).

Copyright © 2002 IEEE. All rights reserved.

C.6 Idle mode procedures

The inquiry and discovery procedures described here are applicable only to the device that initiates them (A). The requirements on the behavior of device B are according to the modes specified in Annex C.4 and in Clause 9. See Table C.4.

Table C.4—Idle mode procedures

Item	Procedure	Reference	Support
1	General inquiry	C.6.1	C1
2	Limited inquiry	C.6.2	C1
3	Name discovery	C.6.3	O
4	Device discovery	C.6.4	O
5	Bonding	C.6.5	O
C1: If initiation of bonding is supported, support for at least one inquiry procedure is mandatory, otherwise optional. (NOTE—Support for LMP-pairing is mandatory in Clause 9.)			

C.6.1 General inquiry

C.6.1.1 Purpose

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

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

C.6.1.2 Term on UI level

Bluetooth Device Inquiry.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1092 of 1169

C.6.1.3 Description

Figure C.6 describes a general inquiry, where B is a device in nondiscoverable mode, B' is a device in limited discoverable mode, and B" is a device in general discoverable mode. Note that all discoverable devices are discovered using general inquiry, independent of which discoverable mode they are in.

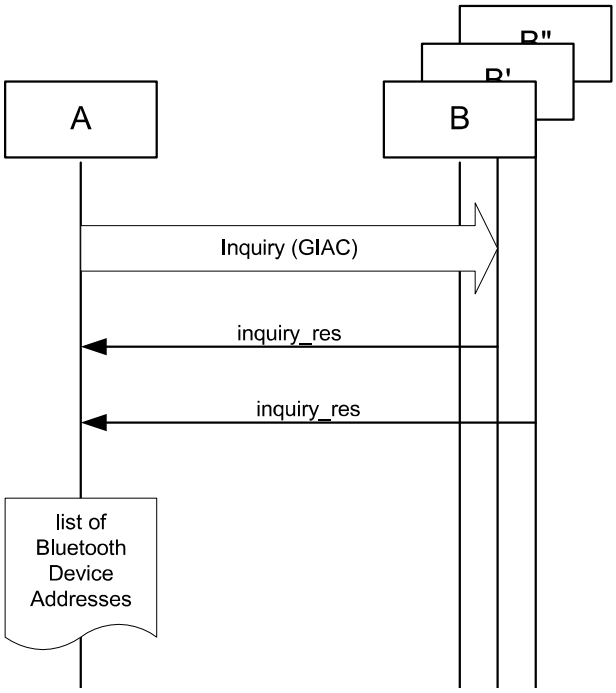


Figure C.6—General inquiry, where B is a device in nondiscoverable mode, B´ is a device in limited discoverable mode, and B" is a device in general discoverable mode

C.6.1.4 Conditions

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

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

C.6.2 Limited inquiry

C.6.2.1 Purpose

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

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

Copyright © 2002 IEEE. All rights reserved.

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

C.6.2.2 Term on UI level

Bluetooth Device Inquiry.

C.6.2.3 Description

Figure C.7 describes a limited inquiry where B is a device in nondiscoverable mode, B' is a device in limited discoverable mode, and B" is a device in general discoverable mode. Note that only limited discoverable devices can be discovered using limited inquiry.

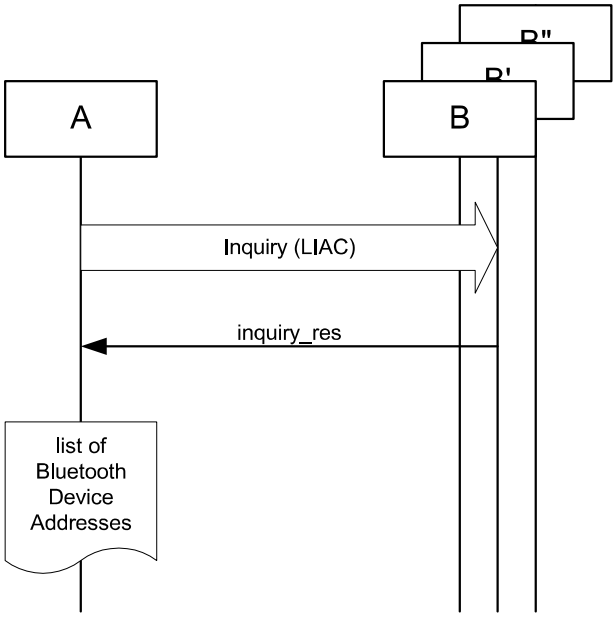


Figure C.7—Limited inquiry, where B is a device in nondiscoverable mode, B' is a device in limited discoverable mode, and B" is a device in general discoverable mode

C.6.2.4 Conditions

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

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

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1094 of 1169

C.6.3 Name discovery

C.6.3.1 Purpose

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

C.6.3.2 Term on UI level

Bluetooth Device Name Discovery.

C.6.3.3 Description

C.6.3.3.1 Name request

Name request is the procedure for retrieving the Bluetooth Device Name from a connectable Bluetooth device. It is not necessary to perform the full link establishment procedure (see Annex C.7.1) in order to just to get the name of another device. See Figure C.8.

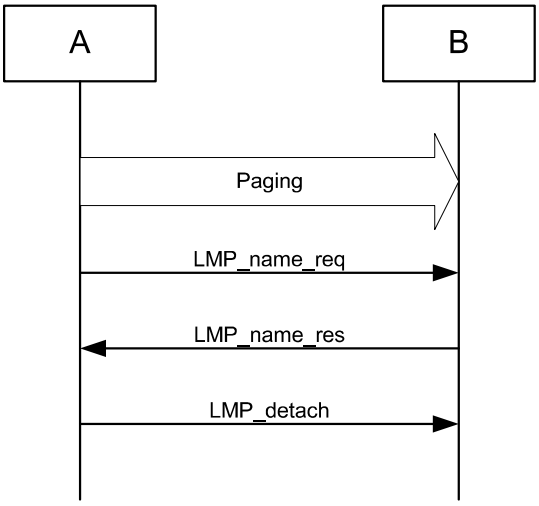


Figure C.8—Name request procedure

C.6.3.3.2 Name discovery

Name discovery is the procedure for retrieving the Bluetooth Device Name from connectable Bluetooth devices by performing name request toward known devices (i.e., Bluetooth devices for which the Bluetooth Device Addresses are available). See Figure C.9.

C.6.3.4 Conditions

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

Copyright © 2002 IEEE. All rights reserved.

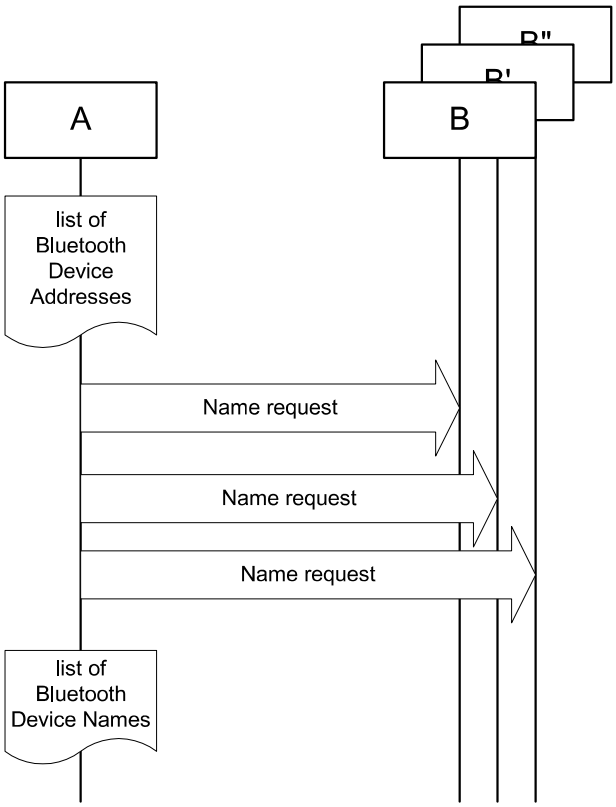


Figure C.9—Name discovery procedure

C.6.4 Device discovery

C.6.4.1 Purpose

The purpose of device discovery is to provide the initiator with the Bluetooth Address, clock, Class of Device, used page scan mode, and Bluetooth device name of discoverable devices. See Figure C.10.

C.6.4.2 Term on UI level

Bluetooth Device Discovery.

C.6.4.3 Description

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

C.6.4.4 Conditions

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

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1096 of 1169

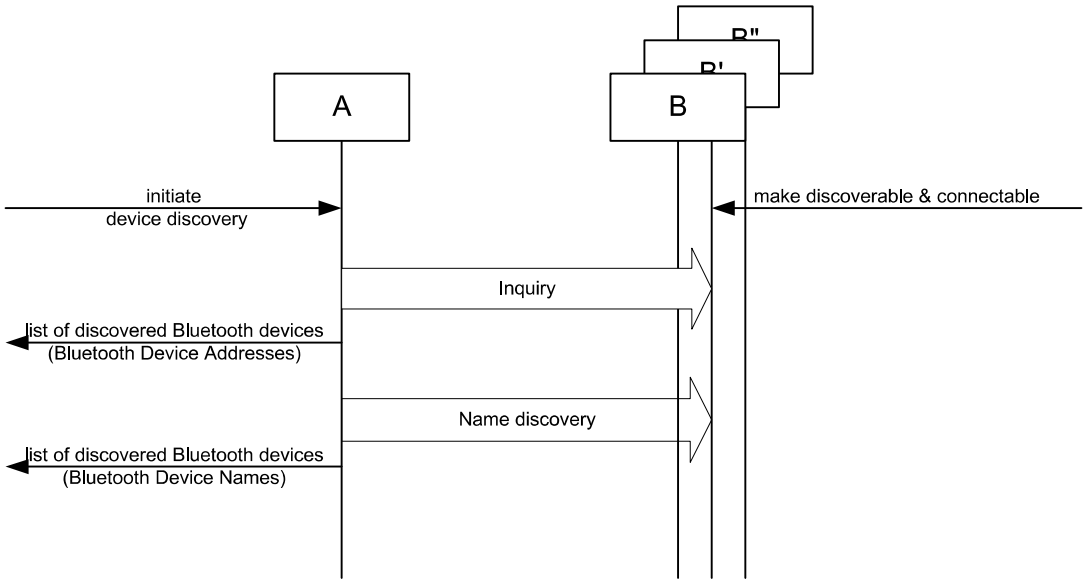


Figure C.10—Device discovery procedure

C.6.5 Bonding

C.6.5.1 Purpose

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

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

C.6.5.2 Term on UI level

Bluetooth Bonding.

C.6.5.3 Description

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

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

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

Copyright © 2002 IEEE. All rights reserved.

C.6.5.3.1 General bonding

Figure C.11 provides the general description of bonding as being the link establishment procedure executed under specific conditions on both devices, followed by an optional higher layer initialization process.

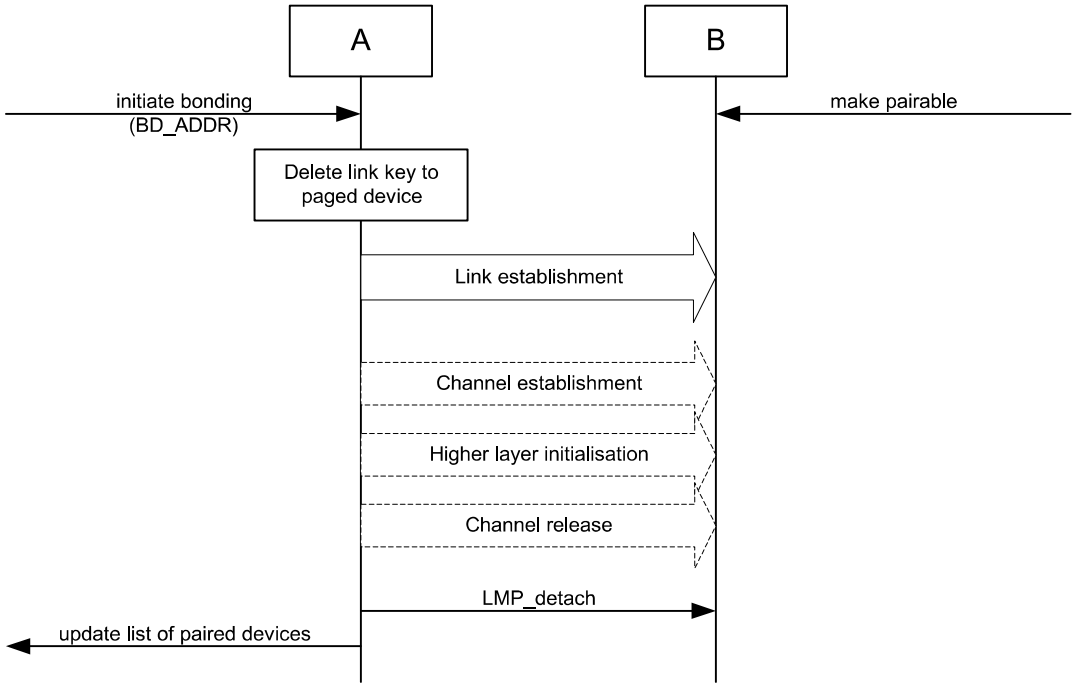


Figure C.11—General description of bonding as being the link establishment procedure executed under specific conditions on both devices, followed by an optional higher layer initialization process

Copyright © 2002 IEEE. All rights reserved.

C.6.5.3.2 Dedicated bonding

Figure C.12 describes the bonding as performed when the purpose of the procedure is only to create and exchange a link key between two Bluetooth devices.

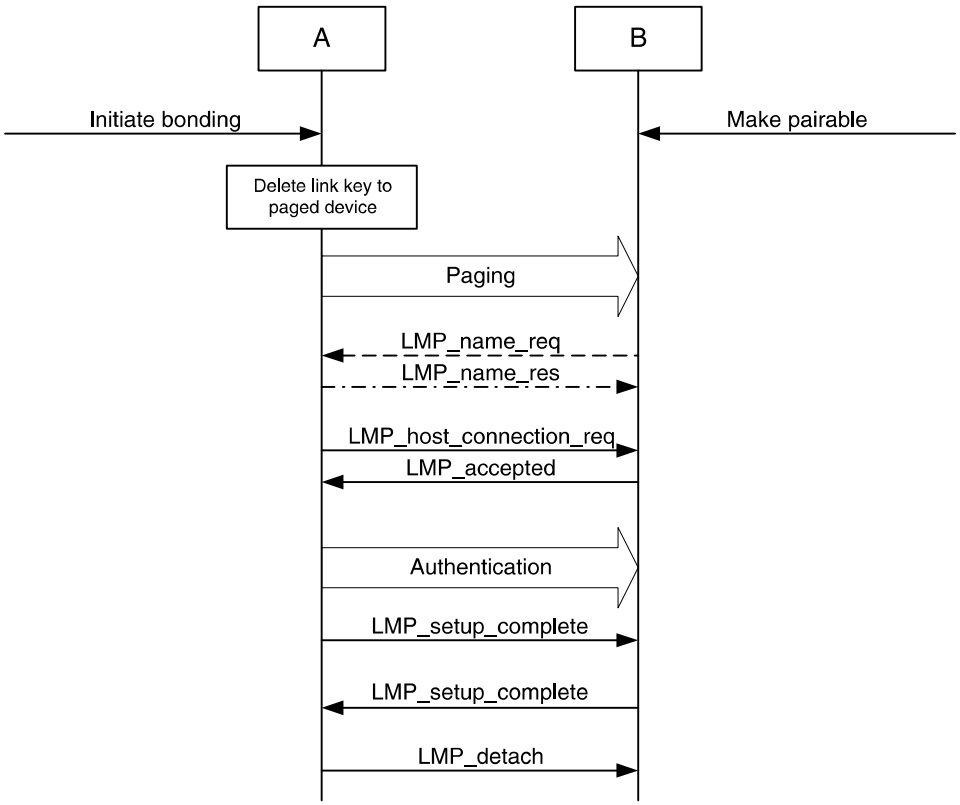


Figure C.12—Bonding as performed when the purpose of the procedure is only to create and exchange a link key between two Bluetooth devices

C.6.5.4 Conditions

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

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

— Paged device (B) shall be set into pairable mode. Paging device (A) is assumed to allow pairing since it has initiated the bonding procedure.

— Paging device A (the initiator of the bonding procedure) shall initiate authentication.

— Before initiating the authentication part of the bonding procedure, the paging device (A) should delete any link key corresponding to a previous bonding with the paged device.

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1099 of 1169

C.7 Establishment procedures

The establishment procedures defined here do not include any discovery part (see Table C.5). Before establishment procedures are initiated, the information provided during device discovery (in the FHS packet of the inquiry response or in the response to a name request) has to be available in the initiating device. This information is as follows:

— Bluetooth Device Address (BD_ADDR) from which the Device Access Code is generated

— System clock of the remote device

— Page scan mode used by the remote device

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

— Class of device

— Device name

Table C.5—Establishment procedures

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

C.7.1 Link establishment

C.7.1.1 Purpose

The purpose of the link establishment procedure is to establish a physical link (of ACL type) between two Bluetooth devices using procedures from Clause 8 and Clause 9.

C.7.1.2 Term on UI level

Bluetooth link establishment.

C.7.1.3 Description

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

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1100 of 1169

C.7.1.3.1 B in security mode 1 or 2

Figure C.13 describes the link establishment procedure when the paging device (A) is in security mode 3 and the paged device (B) is in security mode 1 or 2.

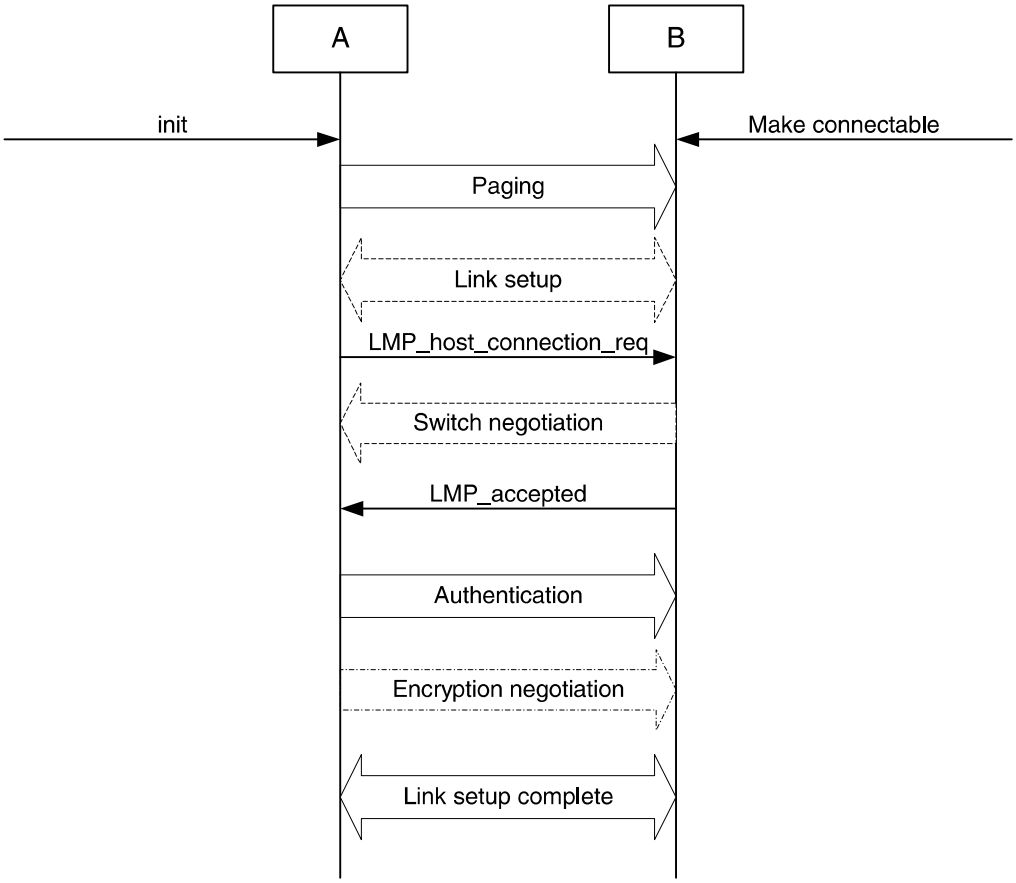


Figure C.13—Link establishment procedure when the paging device (A) is in security mode 3 and the paged device (B) is in security mode 1 or 2

Copyright © 2002 IEEE. All rights reserved.

C.7.1.3.2 B in security mode 3

Figure C.14 describes the link establishment procedure when both the paging device (A) and the paged device (B) are in security mode 3.

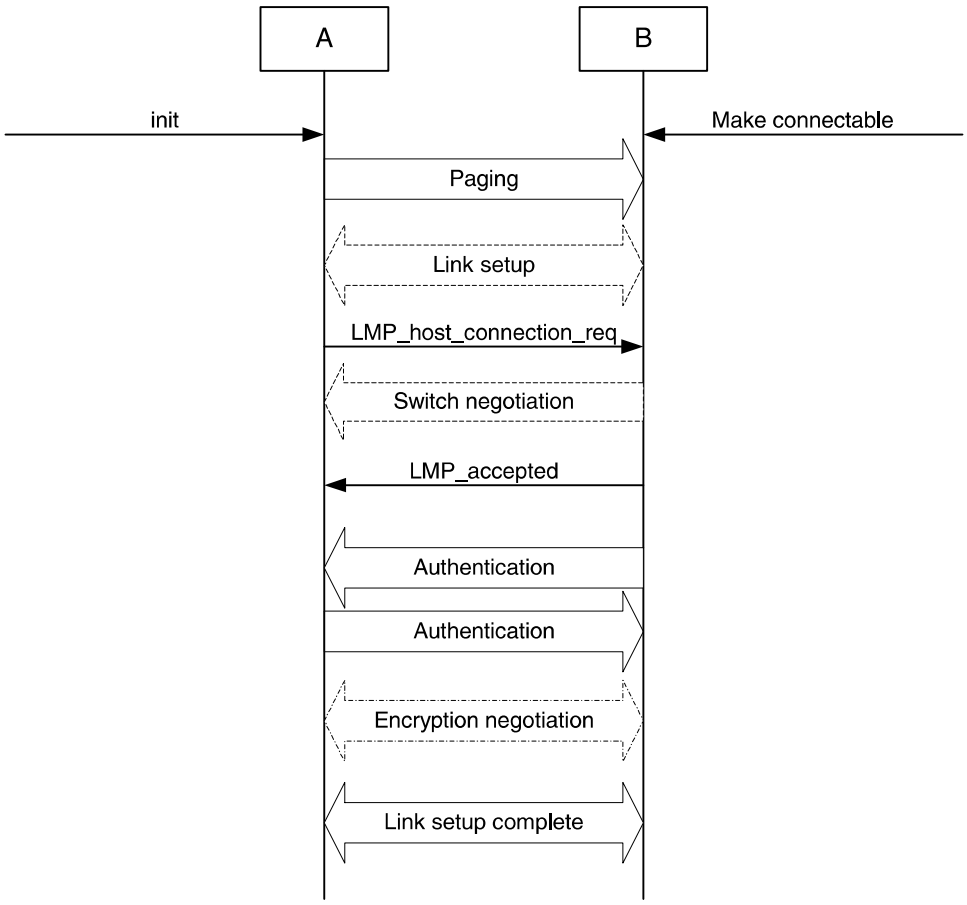


Figure C.14—Link establishment procedure when both the paging device (A) and the paged device (B) are in security mode 3

C.7.1.4 Conditions

The paging procedure shall be according to Clause 8 and the paging device should use the Device access code and page mode received through a previous inquiry. When paging is completed, a physical link between the two Bluetooth devices is established.

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

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

Copyright © 2002 IEEE. All rights reserved.

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

The paging device shall send LMP_host_connection_req during link establishment (i.e., before channel establishment) and may initiate authentication only after having sent LMP_host_connection_request.

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

Further link configuration may take place after the LMP_host_connection_req. When both devices are satisfied, they send LMP_setup_complete.

Link establishment is completed when both devices have sent LMP_setup_complete.

C.7.2 Channel establishment

C.7.2.1 Purpose

The purpose of the channel establishment procedure is to establish a Bluetooth channel (a logical link) between two Bluetooth devices using Clause 10.

C.7.2.2 Term on UI level

Bluetooth channel establishment.

C.7.2.3 Description

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

C.7.2.3.1 B in security mode 2

Figure C.15 describes the channel establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 2.

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1103 of 1169

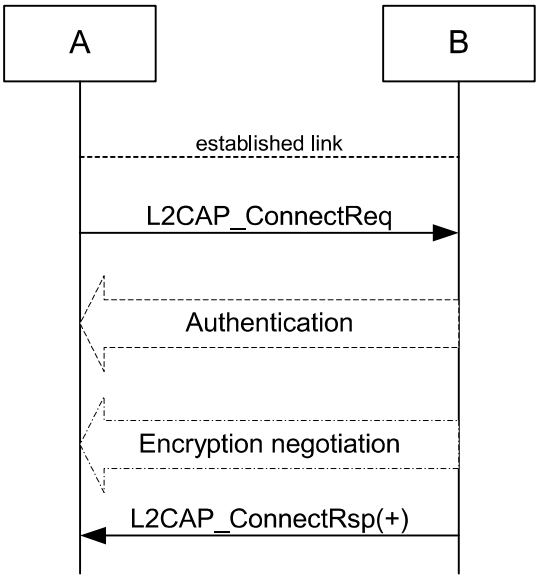


Figure C.15—Channel establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 2

C.7.2.3.2 B in security mode 1 or 3

Figure C.16 describes the channel establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 1 or 3.

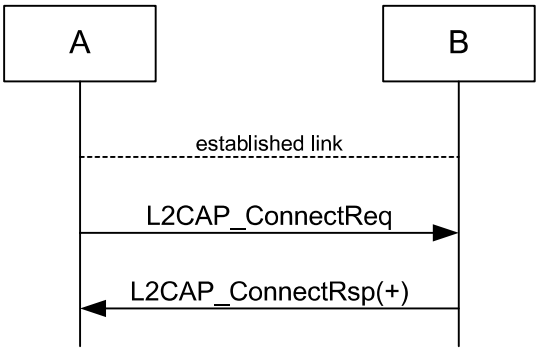


Figure C.16—Channel establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 1 or 3

C.7.2.4 Conditions

Channel establishment starts after link establishment is completed when the initiator sends a channel establishment request (L2CAP_ConnectReq).

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

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1104 of 1169

Channel establishment is completed when the acceptor responds to the channel establishment request (with a positive L2CAP_ConnectRsp).

C.7.3 Connection establishment

C.7.3.1 Purpose

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

C.7.3.2 Term on UI level

Bluetooth connection establishment.

C.7.3.3 Description

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

C.7.3.3.1 B in security mode 2

Figure C.17 describes the connection establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 2.

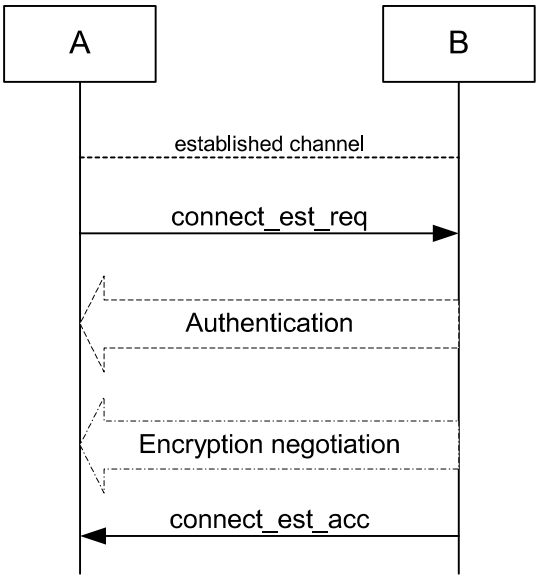


Figure C.17—Connection establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 2

Copyright © 2002 IEEE. All rights reserved.

C.7.3.3.2 B in security mode 1 or 3

Figure C.18 describes the connection establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 1 or 3.

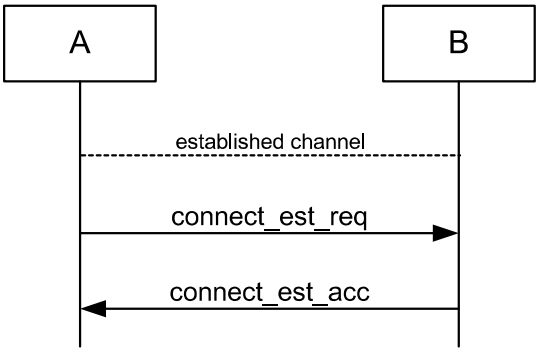


Figure C.18—Connection establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 1 or 3

C.7.3.4 Conditions

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

Connection establishment is completed when the acceptor accepts the connection establishment request ("connect_est_acc" is application protocol dependent).

C.7.4 Establishment of additional connection

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

— A second connection on the same channel, and/or

— A second channel on the same link, and/or

— A second physical link.

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

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1106 of 1169

C.8 Timers and constants

The timers in Table C.6 are required by this profile.

Table C.6—Defined GAP timers

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

Copyright © 2002 IEEE. All rights reserved.

C.9 Information flows of related procedures (Informative)

This subclause (C.9) is informative and is included for information purposes only. It is not an official part of the standard itself.

C.9.1 lmp-authentication

The specification of authentication on link level is found in Clause 9. See Figure C.19.

The secret key used here is an already exchanged link key .

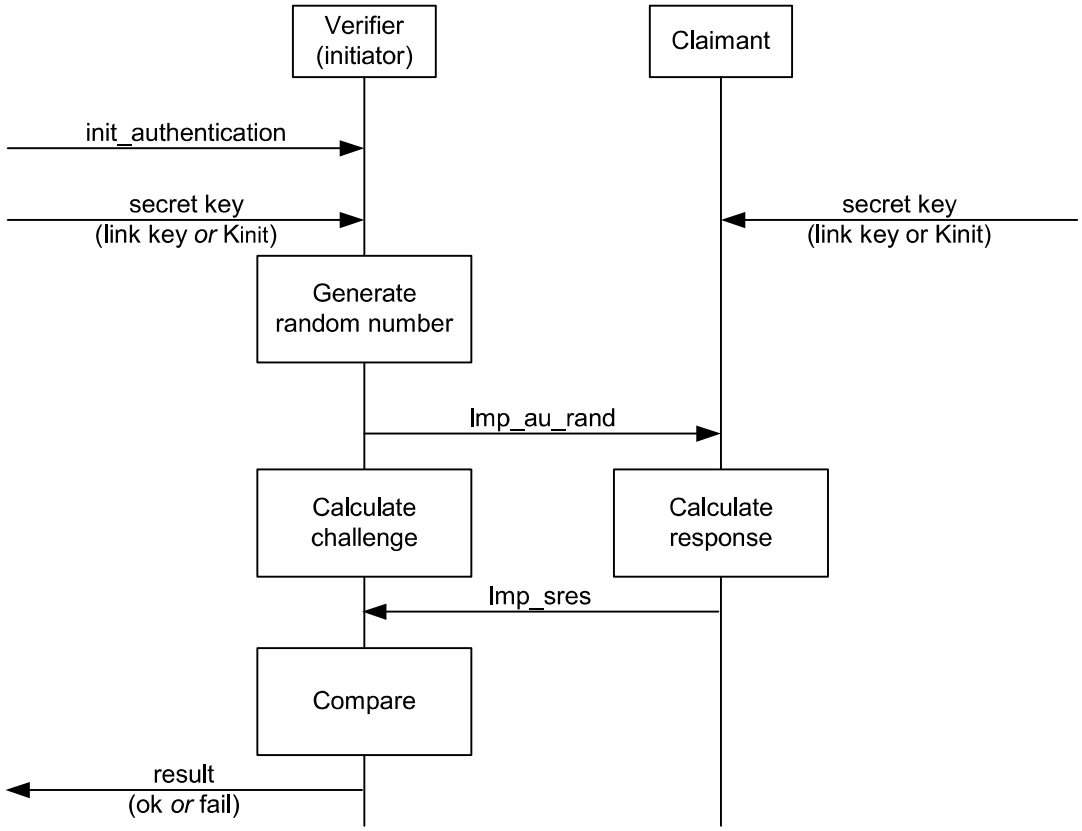


Figure C.19—LMP-authentication as defined by Clause 9

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1108 of 1169

C.9.2 lmp-pairing

The specification of pairing on link level is found in Clause 9. See Figure C.20.

The PIN used here is PN_{BB}.

The create link key procedure is described in 8.14.2.2 and 9.3.3.4. In case the link key is based on a combination key, a mutual authentication takes place and shall be performed irrespective of current security mode.

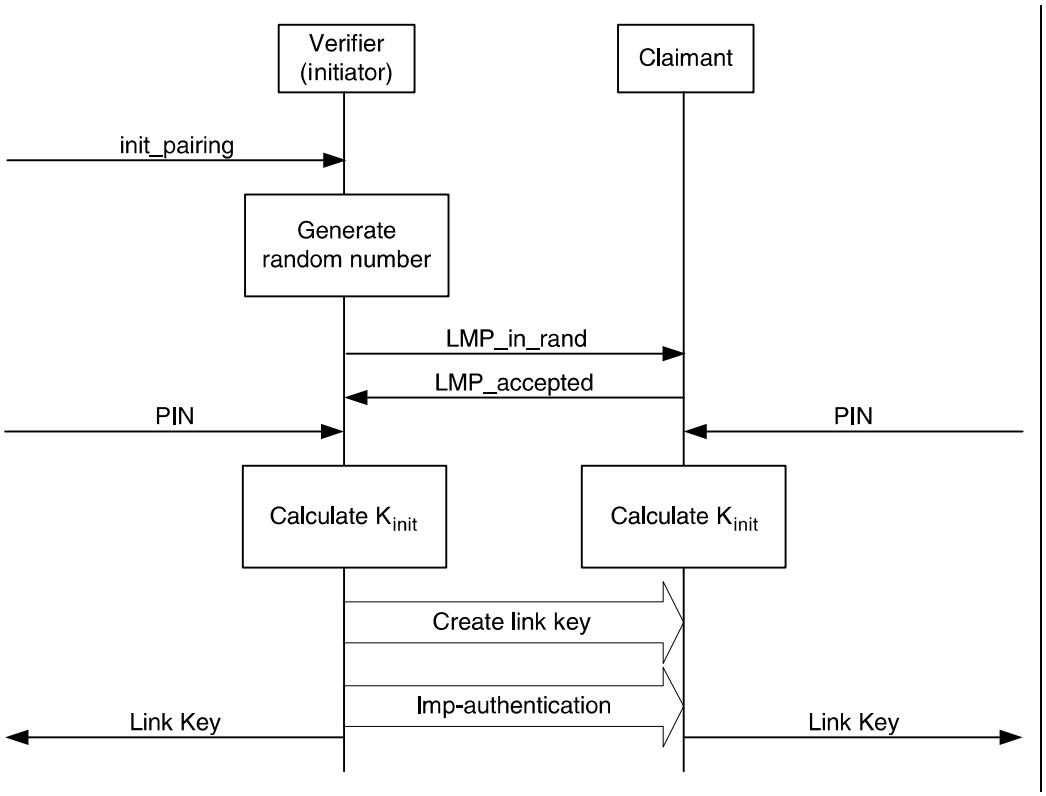


Figure C.20—LMP-pairing as defined in Clause 9

C.9.3 Service discovery

The Service Discovery Protocol specifies what PDUs are used over-the-air to inquire about services and service attributes. The procedures for discovery of supported services and capabilities using the Service Discovery Protocol are described in the Bluetooth Profiles Specification, Part K:2 (see 2.3). Figure C.21 is just an example.

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1109 of 1169

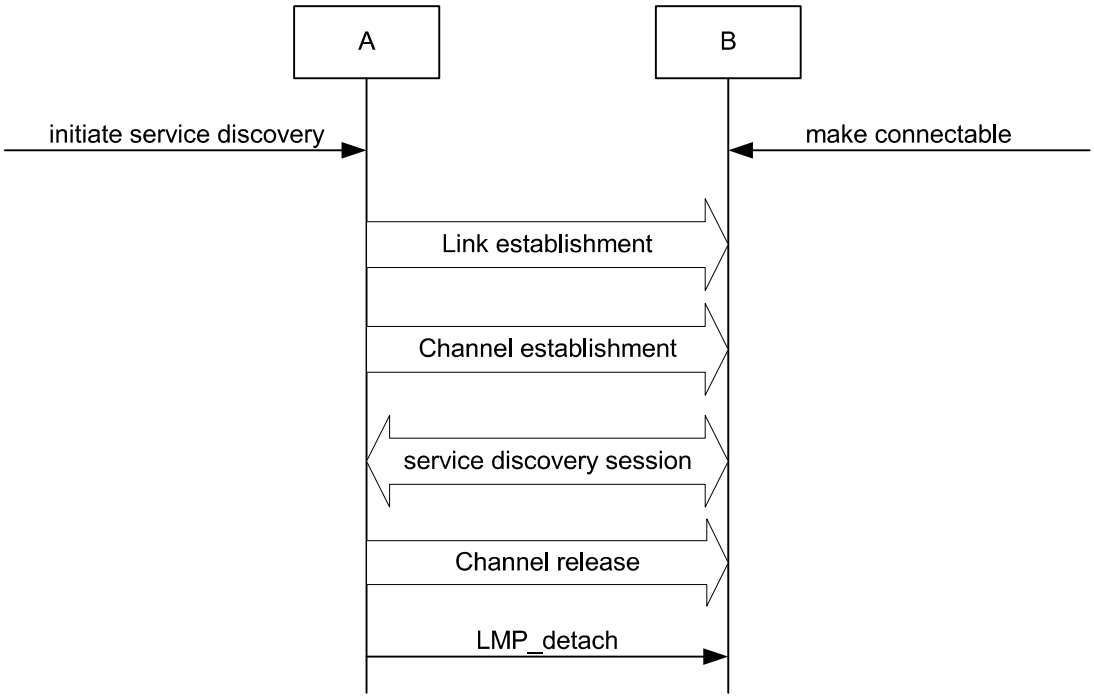


Figure C.21—Service discovery procedure

Copyright © 2002 IEEE. All rights reserved.

Copyright © 2002 IEEE. All rights reserved.

Annex D

(normative)

Optional paging schemes

D.1 General

For the access procedure, several paging schemes may be used. There is one mandatory paging scheme that shall be supported by all Bluetooth devices. This scheme has been described in 8.10.6. In addition to the mandatory scheme, a Bluetooth unit may support one or more optional paging schemes. The method used for page scan is indicated in the FHS payload (see 8.4.4.1.4). Three additional optional paging schemes are possible; however only optional paging scheme I has been defined yet.

D.2 Optional paging scheme I

In this subclause, the first optional paging scheme is described which may be used according to the rules specified in 8.10 and 9.3.23. The paging code for optional scheme I is 1 (0 is used for the mandatory scheme), see also 8.4.4.1.4.

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

D.2.1 Page

The same 32 frequencies that are used for transmitting ID npackets in the mandatory paging scheme are used in the optional paging scheme I (for the construction of page trains, see 8.11.3.2. The 32 frequencies are also split into an A train and B train. In contrast to the mandatory scheme, the same 32 frequencies that are used for transmitting are also used for reception trials to catch the response from the addressed device.

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

— The page train consists of 10 slots, or 6.25 ms.

— The first 8 slots of the train are used to transmit the ID packets, the 9th slot is used to send a marker packet, and the 10th slot is used for the return of a slave response.

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

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

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

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1112 of 1169

supertrain, which is repeated. An example of a supertrain with the eight subtrains is illustrated in Figure D.1. The supertrain length is 50 ms. In this example, the A train is assumed with an estimated frequency of f_8; as a consequence, the frequencies selected for the train range from f_0 to f_{15}. The marker codes M_ID are indicated as **M**; the receive (half) slots are indicated as **R**.

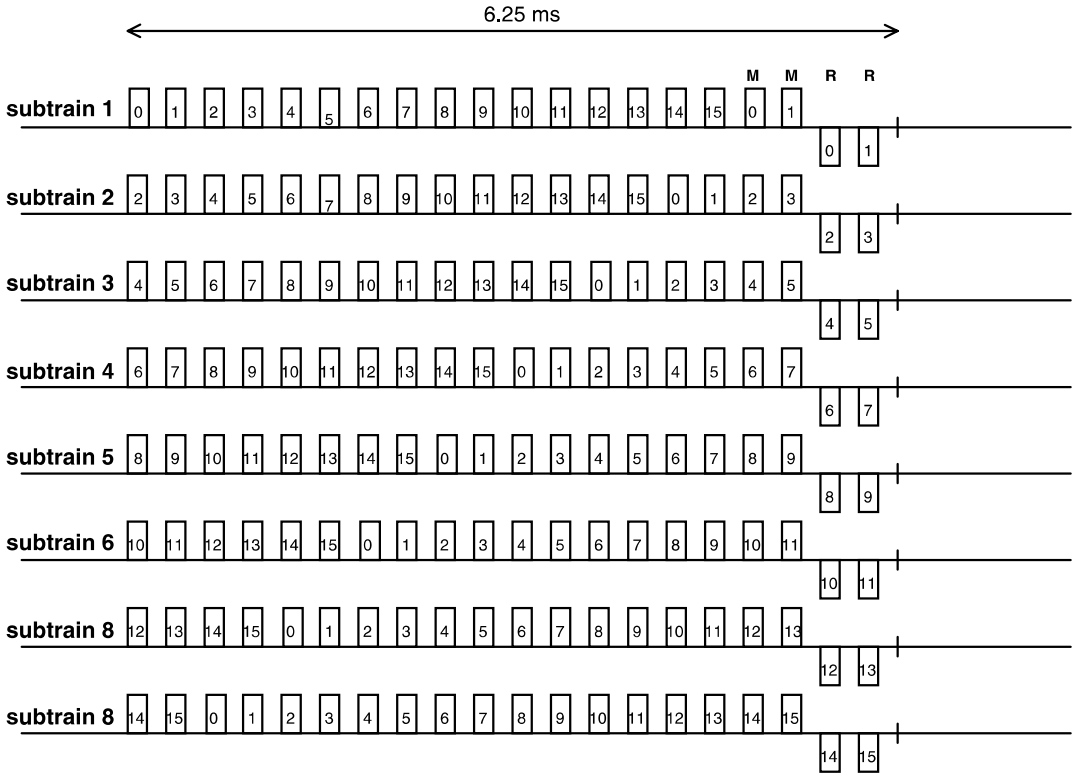


Figure D.1—Example of train configuration for optional page scheme I

Corresponding to the paging modes R0, R1 and R2 of the mandatory scheme, the optional scheme supports the same three modes as described for the mandatory scheme in 8.10.6.2.

Since the subtrain length is now 10 slots, the 1.28 s interval does not cover a multiple of (sub)trains any longer. Therefore, in contrast to the mandatory scheme, the exchange from A train to B train and vice versa is not based on the 1.28 s interval, but instead on a multiple number of supertrains. For the R1 and R2 modes, the repetition of a supertrain N_{sup} is indicated in Table D.1 below.

Table D.1—Relation between repetition duration of A and B trains and paging modes R1 and R2 when SCO links are present

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

Copyright © 2002 IEEE. All rights reserved.

In accordance with the phase input to the hop selection scheme *Xp* in Equation (4) in 8.11.3.2, the phase input *Xp_opt* in the optional mode is determined by:

$$Xp\_opt=[k_{offset\_opt} + ST(cnt)] \bmod 32 \quad \text{(D.1)}$$

where k_{offset_opt} is determined by the A/B selection and the clock estimation of the recipient:

$$k_{offset\_opt} = \begin{matrix} CLKE_{16\text{-}12} + 24 & \text{A-train} \\ CLKE_{16\text{-}12} + 8 & \text{B-train} \end{matrix} \quad \text{(D.2)}$$

and ST is a function determining the structure of the subtrain and supertrain:

$$ST(cnt) = (cnt \bmod 160 - 2*INT[((cnt \bmod 160) + 18)/20]) \bmod 16 \quad \text{(D.3)}$$

k_{offset_opt} is determined once at the beginning of the repetition period.

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

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

D.2.2 Page Scan

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

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

D.2.3 Page Response Procedures

The page response procedures at the master and slave sides are almost identical to the procedures described in the mandatory mode (see 8.10.6.4). There are two differences as follows:

— The page response routine starts after the transmission and reception of the marker code M_ID.

— The ID packet sent by recipient is identical to the frequency in which the marker code was received.

For the page response timing, see Figure D.2 and Figure D.3.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1114 of 1169

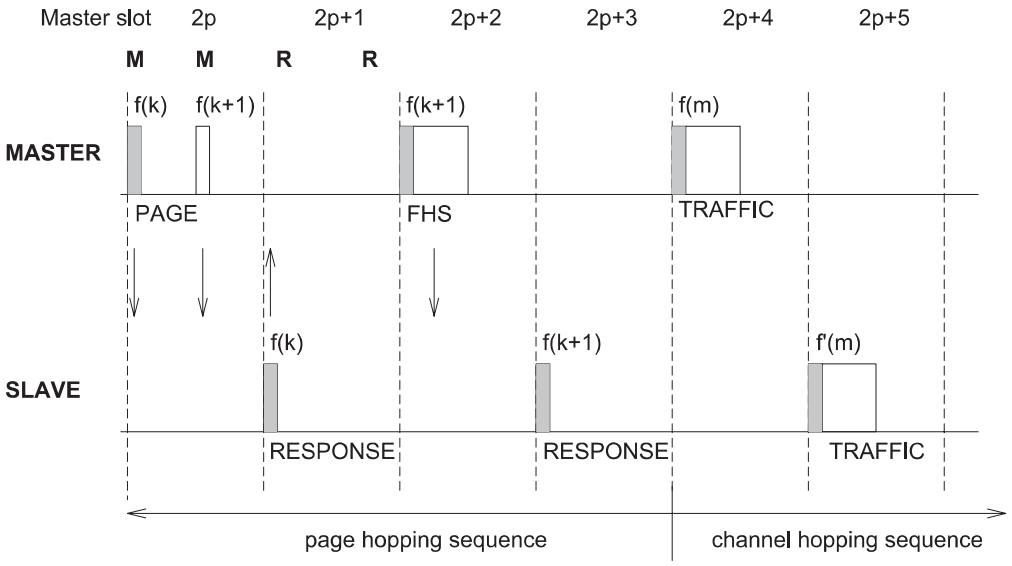


Figure D.2—Messaging when marker code is received in first half slot of even master slot

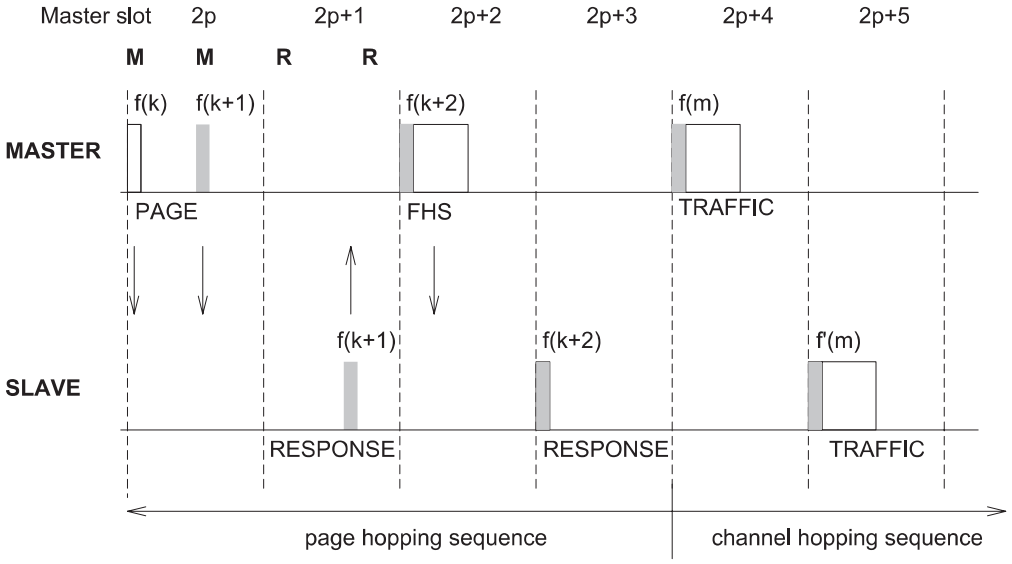


Figure D.3—Messaging when marker code is received in second half slot of even master slot

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1115 of 1169

D.2.4 Train Tracing

This subclause outlines how a slave may search for the mark code although the current partitioning into A trains and B trains at the master side is not known. Train tracing means that the slave tries to receive as many page access codes from the train as possible, to catch a mark code as soon as possible. When searching for the mark codes, or trying to distinguish between the mandatory paging mode and the optional paging mode, a unit shall set up a hopping pattern for train tracing after the reception of the first access code. The hopping pattern shall ensure that the transmission and reception is performed with a 50% probability on the same frequency regardless of the actual frequency set (16 frequencies) used for paging.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1116 of 1169

Copyright © 2002 IEEE. All rights reserved.

Annex E

(normative)

Bluetooth test mode

E.1 General description

This annex describes the test mode for hardware and low-level functionality tests of Bluetooth devices. The test mode includes transmitter tests (i.e., for packets with constant bit patterns) and loopback tests.

The test mode supports testing of the Bluetooth transmitter and receiver. It is intended mainly for certification and compliance testing of the radio and baseband layer and may also be used for regulatory approval or in-production and after-sales testing.

A device in test mode must not support normal operation. For security reasons the test mode is designed so that it offers no benefit to the user. Therefore, no data output or acceptance on a hardware or software interface is allowed.

E.1.1 Test setup

The setup consists of a DUT and a tester. Optionally, additional measurement equipment may be used; see Figure E.1.

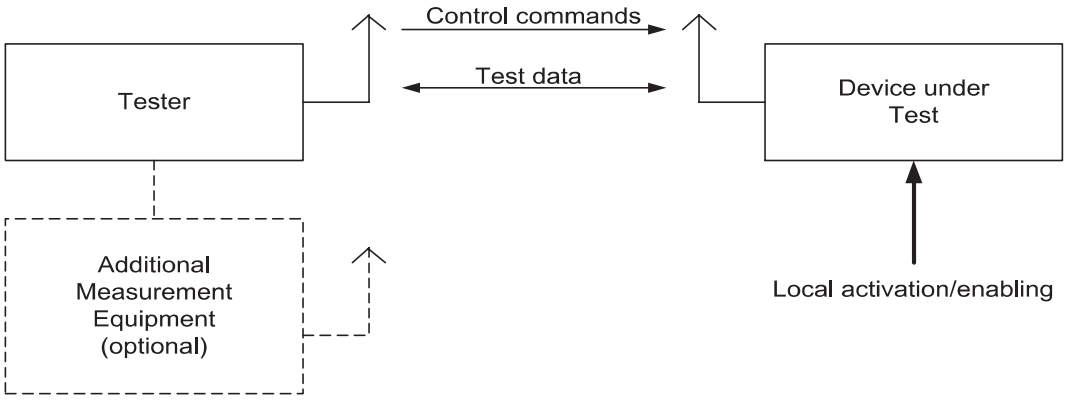


Figure E.1—Setup for test mode

The tester and DUT form a piconet where the tester acts as master and has full control over the test procedure. The DUT acts as slave.

The control is done via the air interface using LMP commands (see E.3 and Clause 9). Hardware interfaces to the DUT may exist, but are not subject to standardization.

The test mode is a special state of the Bluetooth model. For security and type approval reasons, a device in test mode may not support normal operation. When the DUT leaves the test mode, it enters the STANDBY state. After power-off, the Bluetooth device must return to STANDBY state.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1118 of 1169

E.1.2 Activation

The activation may be carried out locally (i.e., via a hardware or software interface) or using the air interface.

— For activation over the air interface, entering the test mode must 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 forces the DUT to enter test mode. The DUT terminates 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 hardware or software interface, the DUT terminates 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.

E.1.3 Control

Control and configuration are performed using special LMP commands (see E.3). These commands must be rejected if the Bluetooth device is not in test mode. In such cases, an LMP_not_accepted is returned. The DUT shall return an LMP_accepted on reception of a control command when in test mode.

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

The DUT can be commanded to leave the test mode by an LMP_Detach command or by sending an LMP_test_control command with test scenario set to “exit test mode.”

E.2 Test scenarios

E.2.1 Transmitter test

The Bluetooth device transmits a constant bit pattern. This pattern is transmitted periodically with packets aligned to the slave TX timing of the piconet formed by tester and DUT. The same test packet is repeated for each transmission.

The transmitter test is started when the master sends the first POLL packet. In nonhopping mode, agreed frequency is used for this POLL packet.

The tester transmits at its TX slots (e.g., control commands or POLL packets). The slave starts burst transmission in the following slave TX slot. The master’s polling interval is fixed and defined beforehand. The DUT may transmit its burst according to the normal timing even if no packet from the tester was received. In such cases, it is highly recommended that the ARQN bit is set to NAK.

The burst length may exceed the length of a one slot packet. In such cases, the tester may take the next free master TX slot for polling. The timing is illustrated in Figure E.2.

Copyright © 2002 IEEE. All rights reserved.

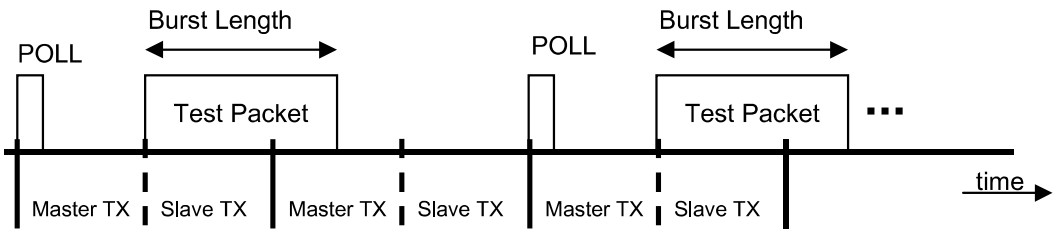


Figure E.2—Timing for transmitter test

E.2.1.1 Packet format

The test packet is a normal Bluetooth packet (see Figure E.3). For the payload itself, see below.

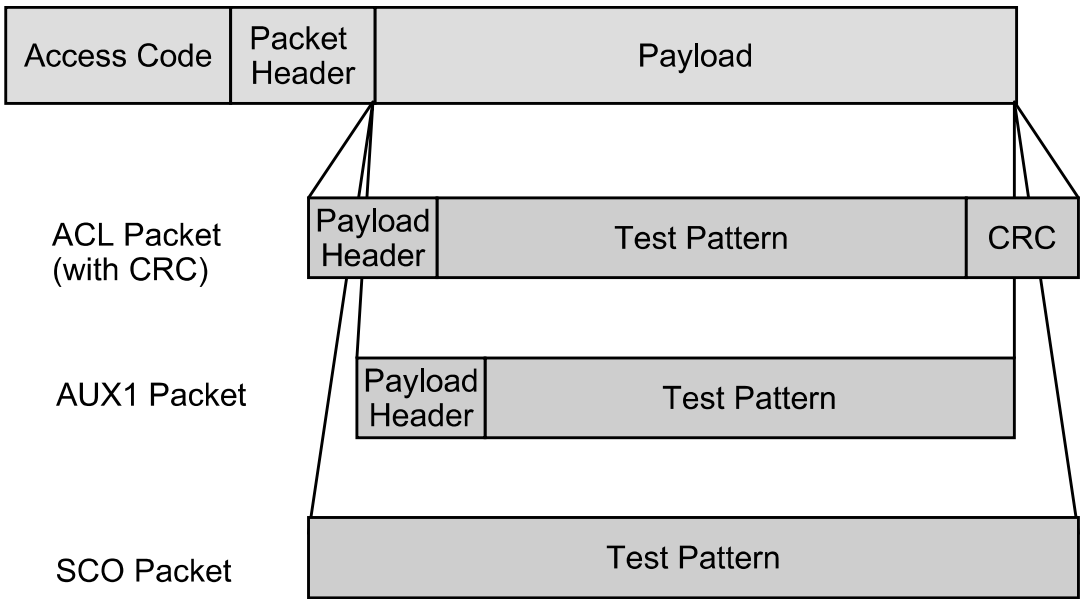


Figure E.3—General format of TX packet

During configuration the tester defines

— The packet type to be used (i.e., ACL packet, AUX1 packet, SCO packet)

— Payload length

For the payload length, the restrictions from the baseband specification apply (see Clause 8). For ACL packets, the payload structure defined in the baseband specification is preserved as well (see Figure E.3).

For the transmitter test mode, only packets without FEC should be used, i.e., HV3, DH1, DH3, DH5, and AUX1 packets. Support of packet type is mandatory only up to the longest implemented packet type.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1120 of 1169

In transmitter test mode, the packets exchanged between tester and DUT are not scrambled with the whitening sequence. Whitening is turned off when the DUT has accepted to enter the transmitter test mode and is turned on when the DUT has accepted to exit the transmitter test mode (see Figure E.4).

NOTE—Implementations must ensure that retransmissions of the LMP_Accepted messages use the same whitening status.

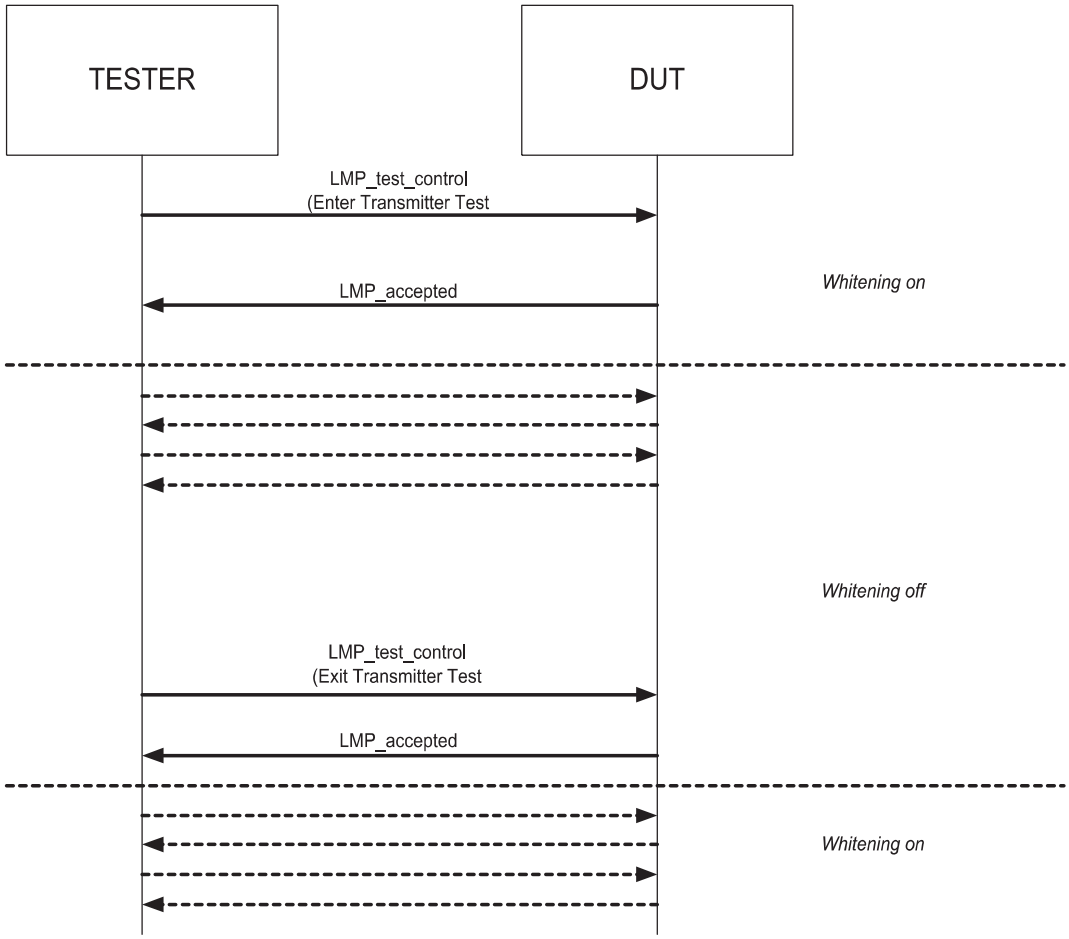


Figure E.4—Use of whitening in transmitter mode

E.2.1.2 Pseudo-random sequence

For pseudo-random bit sequences, the same sequence of bits is used for each transmission (i.e., the packet is repeated, see above). A PRBS-9 Sequence[20] is used. See 2.3.

The properties of this sequence are as follows (see 2.3). The sequence may be generated in a nine-stage shift register whose fifth and ninth stage outputs are added in a modulo-2 addition stage (see Figure E.5), and the result is fed back to the input of the first stage. The sequence begins with the first 1 of nine consecutive 1's, i.e., the shift register is initialized with nine 1's.

[20]Some uncertainties about Japanese regulatory requirements have been reported. If necessary for regulatory type approval in Japan, some features might be added, e.g., a longer pseudo-random noise sequence.

Copyright © 2002 IEEE. All rights reserved.

— Number of shift register stages: 9

— Length of pseudo-random sequence: $2^9 - 1 = 511$ bits

— Longest sequence of 0's: 8 (noninverted signal)

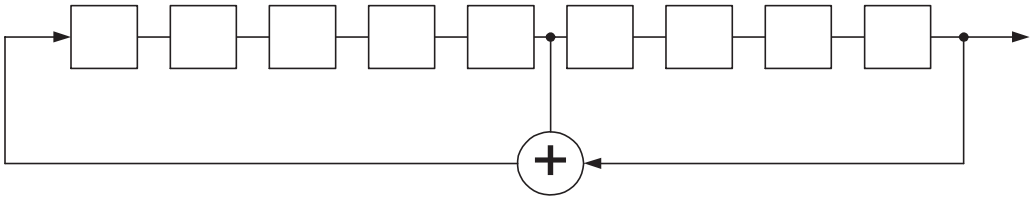


Figure E.5—LFSR for generation of the PRBS

E.2.1.3 Reduced hopping sequence

To support quick testing of the radio over the complete frequency range, a reduced hopping mode is defined. Implementation of this mode is optional for Bluetooth devices and modules.

Reduced hopping uses only five frequencies, on which a sequential hopping is done (channels 0, 23, 46, 69, and 93 are used).[21] See Figure E.6.

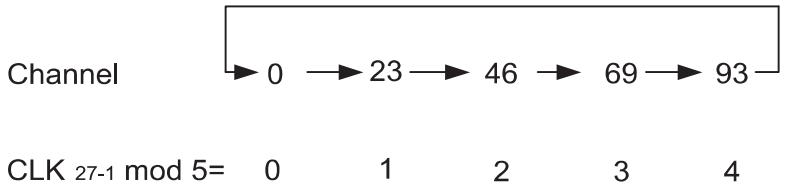


Figure E.6—Reduced hopping scheme

The timing is based on the Bluetooth clock of the tester. The value of CLK_{27-1} (i.e., not using CLK_0, representing half-slots) modulo-5 is used to determine the transmit frequency.

E.2.1.4 Control of transmit parameters

The following parameters can be set to configure the transmitter test:

— Bit pattern
  — Constant zero
  — Constant one
  — Alternating 1010…[22]
  — Alternating 1111 0000 1111 0000…[22]
  — Pseudo-random bit pattern
  — Transmission off

[21]The range is chosen to test the whole frequency range, which covers the normal 79 channels, as well as the French hopping scheme. The frequency assignment rule is the same as for the fixed TX frequency: $f = (2402 + k)$ MHz.

[22]It is recommended that the sequence start with 1; but, as this guideline is irrelevant for measurements, the sequence may also start with 0.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1122 of 1169

- Frequency selection
  - Single frequency
  - Hopping Europe (except France) and USA
  - Hopping France
  - Reduced hopping (implementation in Bluetooth devices and modules is optional)
- TX frequency
  - $k \Rightarrow f := (2402 + k)$ MHz
- Default poll period in TDD frames (n * 1.25 ms)
- Packet type
- Length of test sequence (i.e., user data of packet definition in Clause 8)

E.2.1.5 Power control

If adaptive power control is tested, the normal LMP commands are used. The DUT starts to transmit at the maximum power and reduces or increases its power by one step on every command received.

E.2.1.6 Switch between different frequency settings

A change in the frequency selection becomes effective when the LMP procedure is completed.

When the tester receives the LMP_accepted, it then transmits POLL packets containing the ACK for at least eight slots (i.e., four transmissions). When these transmissions have been completed, the tester moves to the new frequency hop and whitening settings.

After sending a LMP_accepted, the DUT waits for the LC-level ACK for the LMP_accepted. When this PDU is received, it moves to the new frequency hop and whitening settings.

An implementation-defined delay occurs after sending the LMP_accepted before the TX or loopback test starts. Testers must be able to cope with this delay.

Note that the LMP_Accepted packet eventually leads to a loss of frequency synchronization that cannot be recovered. Similar problems occur in normal operation, when the hopping pattern changes.

E.2.2 Loopback test

For loopback testing, the DUT receives normal baseband packets. The received packets are decoded in the DUT, and the payload is sent back using the same packet type. The return packet is sent back in either the TX slot directly following the transmission of the tester, or it is delayed and sent back in the slot after the next transmission of the tester (see Figure E.8 through Figure E.10).

Alternatively, it is possible to implement a delayed loopback instead. Then the return packet is delayed to the following TX slot. No signalling occurs to determine or control the mode. The device behavior must be fixed or adjusted by other means, but must not change randomly.

The tester can select whether whitening is on or off. This setting holds for both uplink and downlink. For switching the whitening status, the same rules as in E.2.1 (see Figure E.4) apply.

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1123 of 1169

For loopback tests, the following rules apply (for illustration, see Figure E.7):

— If the sync word was not detected, no reply is sent.

— If the HEC fails, the DUT replies with a NULL packet with the ARQN bit set to NAK. It is not mandatory to return a NULL packet in this case; the DUT may send nothing.

— If the packet contains an LMP message relating to the control of the test mode, this command is executed and the packet is not returned, although ACK or NAK is still returned as usual procedure. Other LMP commands are ignored, and no packets are returned.

— The payload FEC is decoded, and the payload is coded again for transmission. This procedure allows testing of the FEC handling. If the pure BER shall be determined, the tester chooses a packet type without FEC.

— The CRC is evaluated. In case of a failure, the payload is returned with ARQN = NAK. The CRC for the return packet is calculated for the returned payload.

— If the CRC fails, the number of bytes as indicated in the (possibly erroneous) payload header shall be looped back.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1124 of 1169

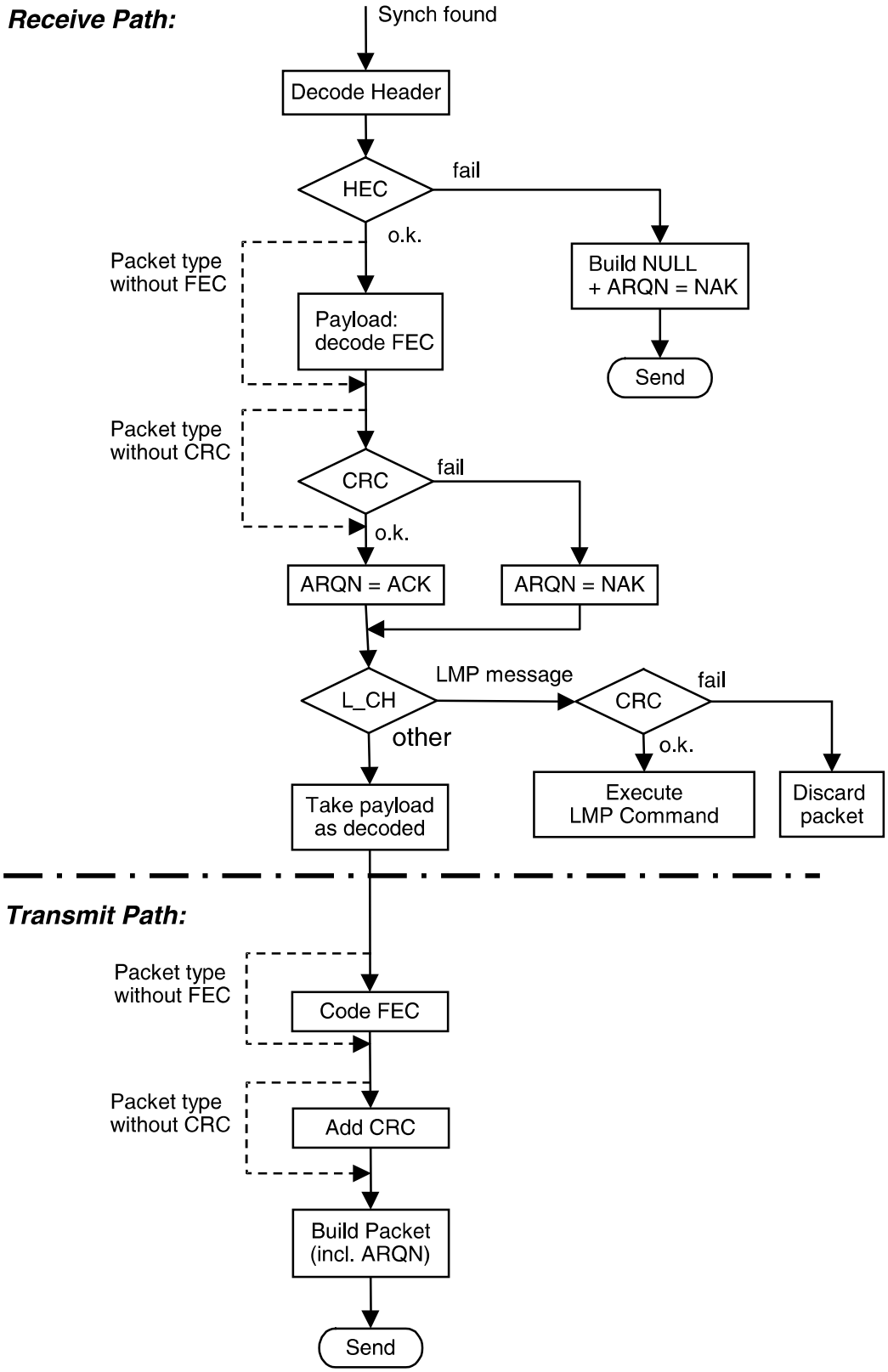


Figure E.7—DUT packet handling in loopback test

Copyright © 2002 IEEE. All rights reserved.

The timing for normal and delayed loopback is illustrated in Figure E.8 through Figure E.10.

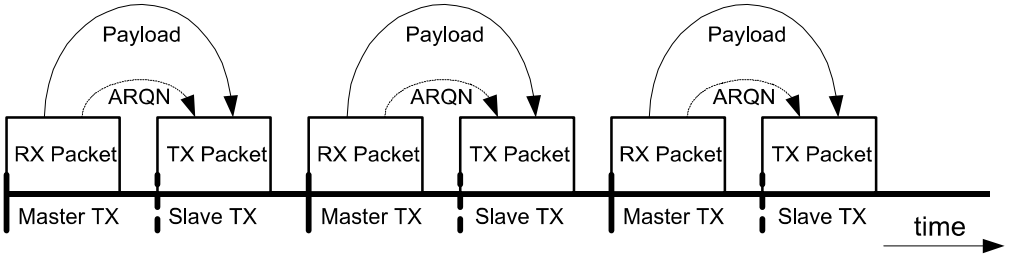


Figure E.8—Payload and ARQN handling in normal loopback

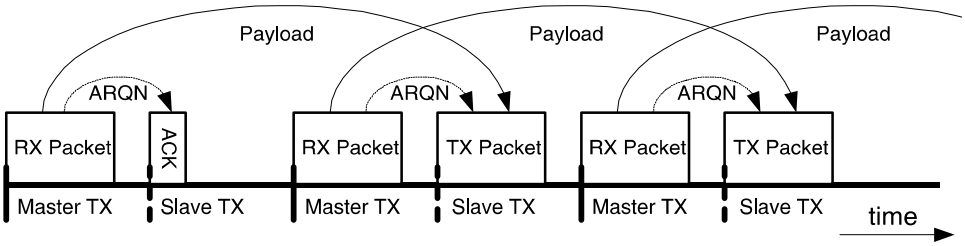


Figure E.9—Payload and ARQN handling in delayed loopback: the start

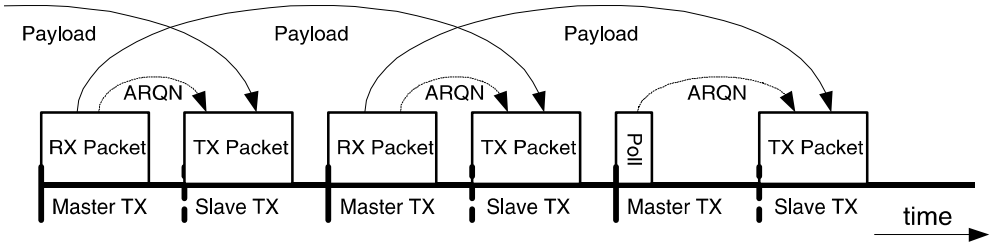


Figure E.10—Payload and ARQN handling in delayed loopback: the end

Copyright © 2002 IEEE. All rights reserved.

The whitening is performed in the same way as it is used in normal active mode.

The following parameters can be set to configure the loopback test:

- Packet class[23]
  - ACL packets
  - SCO packets
  - ACL packets without whitening
  - SCO packets without whitening
- Frequency selection
  - Single frequency (independent for RX and TX)
  - Hopping Europe (except France) and USA
  - Hopping France
  - Hopping reduced uses only five frequencies on which a sequential hopping is done on (channels 0, 23, 46, 69, and 93 are used)
- Power level: (To be used according radio specification requirements)
  - Power control or fixed TX power

The switch of the frequency setting is done exactly as for the transmitter test (see E.2.1.6).

E.2.3 Pause test

Pause test is used by testers when they want the DUT to stop the current loopback or transmitter test and enter a STANDBY mode.

When an LMP_test_control PDU that specifies pause mode is received the DUT stops its current test. In the case of a transmitter test, no more packets will be transmitted. While in pause test mode, the DUT responds normally to POLL packets, i.e., responds with a NULL packet. It also responds normally to all the LMP packets that are allowed in test mode.

When the test scenario is set to “Pause Test,” all the other fields in the LMP_test_control are ignored. There is no change in the hopping scheme or whitening as a result of a request to pause test.

[23]This parameter is included because in the future the packet type numbering may not remain unambiguous.

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1127 of 1169

E.3 Outline of proposed LMP messages

Table E.1 lists all LMP messages used for test mode (see 9.6). To ensure that the contents of LMP_test_control PDU are suitably whitened (important when sent in transmitter mode), all parameters listed in Table E.2 are XORed with 0x55 before being sent.

Table E.1—LMP messages used for test mode

LMP PDU	Opcode	Possible direction[a]	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		

[a]m = master; s = slave.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1128 of 1169

Table E.2—Parameters used in LMP_test_control PDU

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 Pseudo-random bit sequence 5 Closed loopback – ACL packets 6 Closed loopback – SCO packets 7 ACL packets without whitening 8 SCO packets without whitening 9 Transmitter test – 1111 0000 pattern 10–254 reserved 255 Exit test mode Each value is XORed with 0x55.
Hopping mode	1	u_int8		0 RX and TX on single frequency 1 Hopping Europe (except France) and USA 2 Reserved 3 Hopping France 4 Reserved 5 Reduced hopping (optional) 6–255 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		Numbering as in packet header (see Clause 8). The value is XORed with 0x55.
Length of test sequence (= length of user data in Clause 8)	2	u_int16	1 byte	Unsigned binary number The value is XORed with 0x5555.

Copyright © 2002 IEEE. All rights reserved.

The control PDU is used for both transmitter and loopback tests. The restrictions in Table E.3 apply for the parameter settings.

Table E.3—Restrictions for parameters used in LMP_test_control PDU

Parameter	Restrictions	
	Transmitter test	Loopback test
TX frequency	$0 \le k \le 93$	$0 \le k \le 93$
RX frequency	Same as TX frequency	$0 \le k \le 93$
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	Not applicable (set to 0)

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1130 of 1169

Copyright © 2002 IEEE. All rights reserved.

Annex F

(normative)

Baseband timers

F.1 General description

This annex contains a list of all timers defined in Clause 8. Definitions and default values of the timers are also given in this annex.

All timer values are given in slots.

F.1.1 List of timers

F.1.1.1 inquiryTO

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

F.1.1.2 pageTO

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

F.1.1.3 pagerespTO

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

The *pagerespTO* value is eight slots.

F.1.1.4 inqrespTO

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

The *inqrespTO* value is 128 slots.

F.1.1.5 newconnectionTO

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

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1132 of 1169

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

The *newconnectionTO* value is 32 slots.

F.1.1.6 supervisionTO

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

The *supervisionTO* value is determined by an HCI command. At the baseband level, a default value that is equal to 32000 slots (i.e., 20 s) is used.

Copyright © 2002 IEEE. All rights reserved.

Annex G

(normative)

Message sequence charts

G.1 Introduction

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

The goal of this annex is to show the interworkings of HCI commands and LM PDUs. It focuses on the message sequence charts for the procedures specified in Clause 11 with regard to LM procedures from Clause 9.

The most useful scenarios are illustrated in this annex, but all possible alternatives are not covered. 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 include the following:

Box:

— Replaces a group of transactions

— Indicates the start of a procedure or a subscenario

Note that in a message sequence chart where several subscenarios exist, the subscenarios 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

G.2 Services without connection request

G.2.1 Remote name request

The service remote name request is used to find out the name of the remote Bluetooth device without an explicit ACL connection request.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1134 of 1169

Sending an HCI_Remote_Name_Request (BD_ADDR, Page_Scan_Repetition_Mode, Page_Scan_Mode, Clock_Offset), the Host expects that its local Bluetooth device will automatically try to connect to the remote Bluetooth device (with the specified BD_ADDR). Then the local Bluetooth device should try to get the name, to disconnect, and finally to return the name of the remote Bluetooth device back to the Host (see Figure G.1 Remote Name Request: subscenario 1).

Note that if an ACL Connection already exists (see Figure G.1 Remote Name Request: subscenario 2), the Remote Name Request procedure will be executed like an optional service. No Paging and no ACL detachment need to be done.

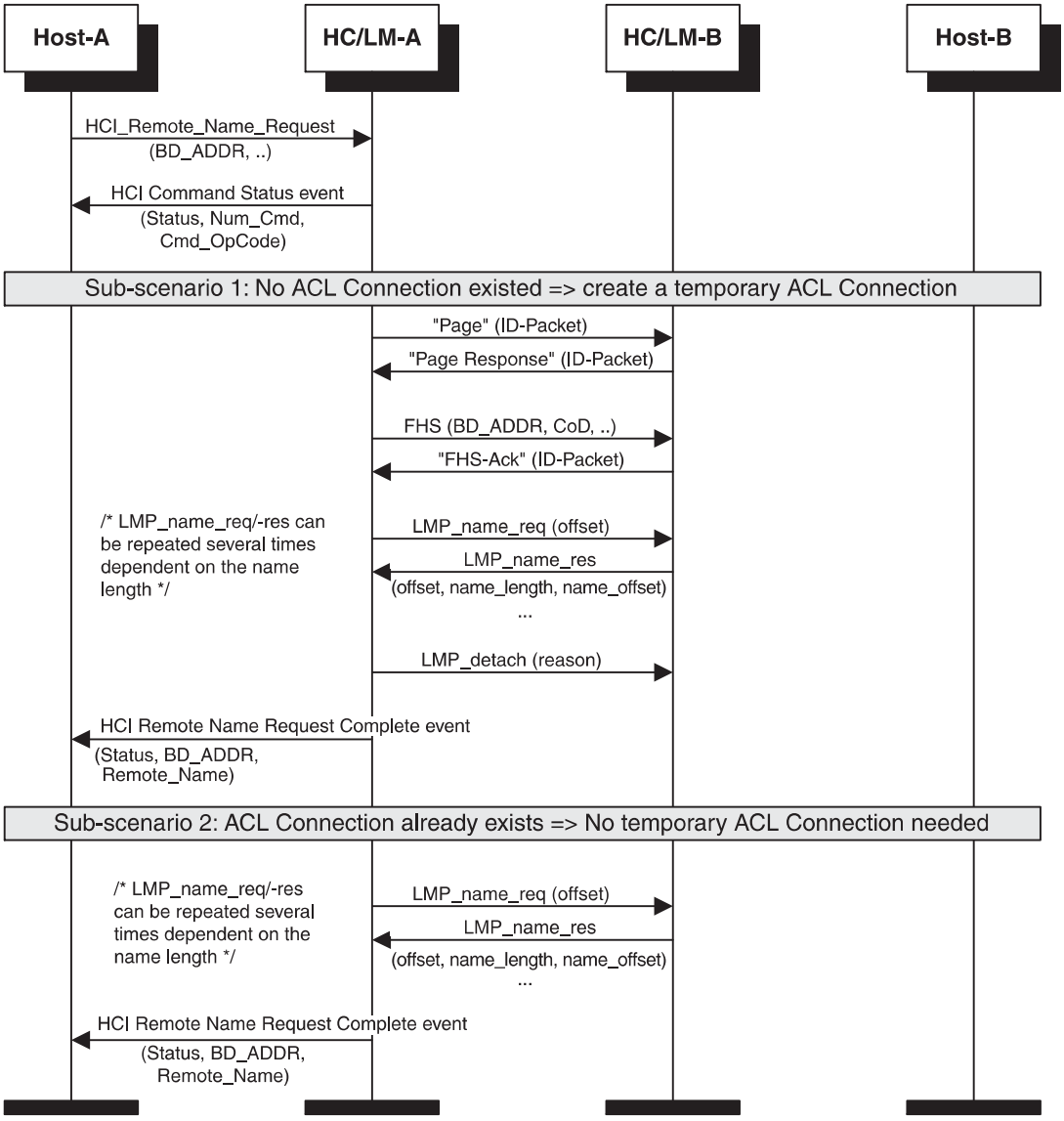


Figure G.1—Remote name request

Copyright © 2002 IEEE. All rights reserved.

G.2.2 One-time inquiry

Inquiry is used to detect and collect nearby Bluetooth devices. See Figure G.2. When receiving the command HCI_Inquiry (LAP, Inquiry_Length, Num_Responses), the 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, the HC will filter out and then return the information related to the found Bluetooth 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 Bluetooth 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) shall be returned to the Host. Otherwise, the command HCI_Inquiry_Cancel( ) will be used to directly stop the inquiry procedure.

G.2.3 Periodic inquiry

Periodic inquiry is needed when the inquiry procedure is to be repeated periodically. See Figure G.3. 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 Bluetooth 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) will be returned to the Host with one, or a list of, found Bluetooth devices. The periodic Inquiry can be stopped using HCI_Exit_Periodic_Inquiry_Mode( ).

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1136 of 1169

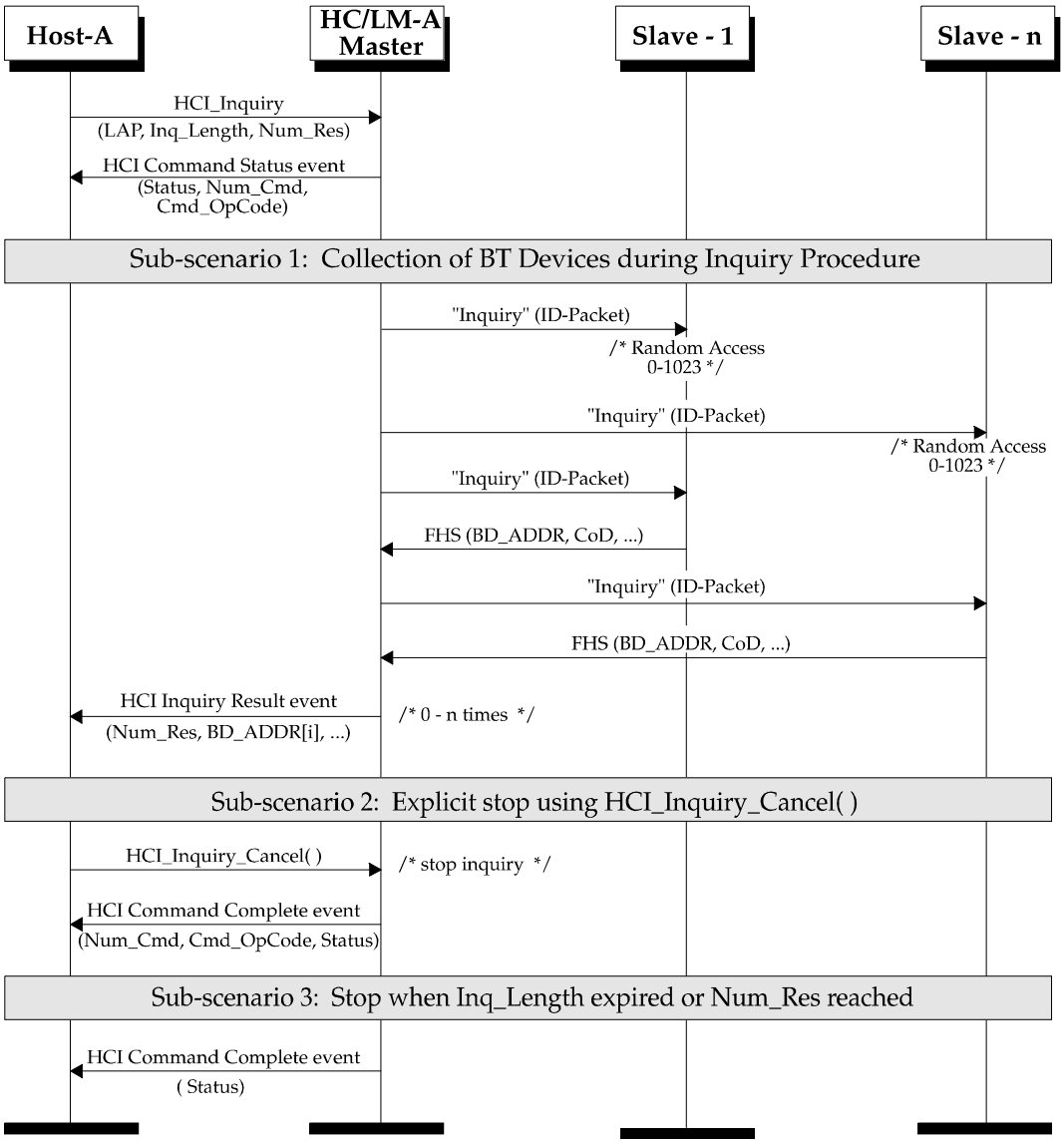


Figure G.2—One-time inquiry

Copyright © 2002 IEEE. All rights reserved.

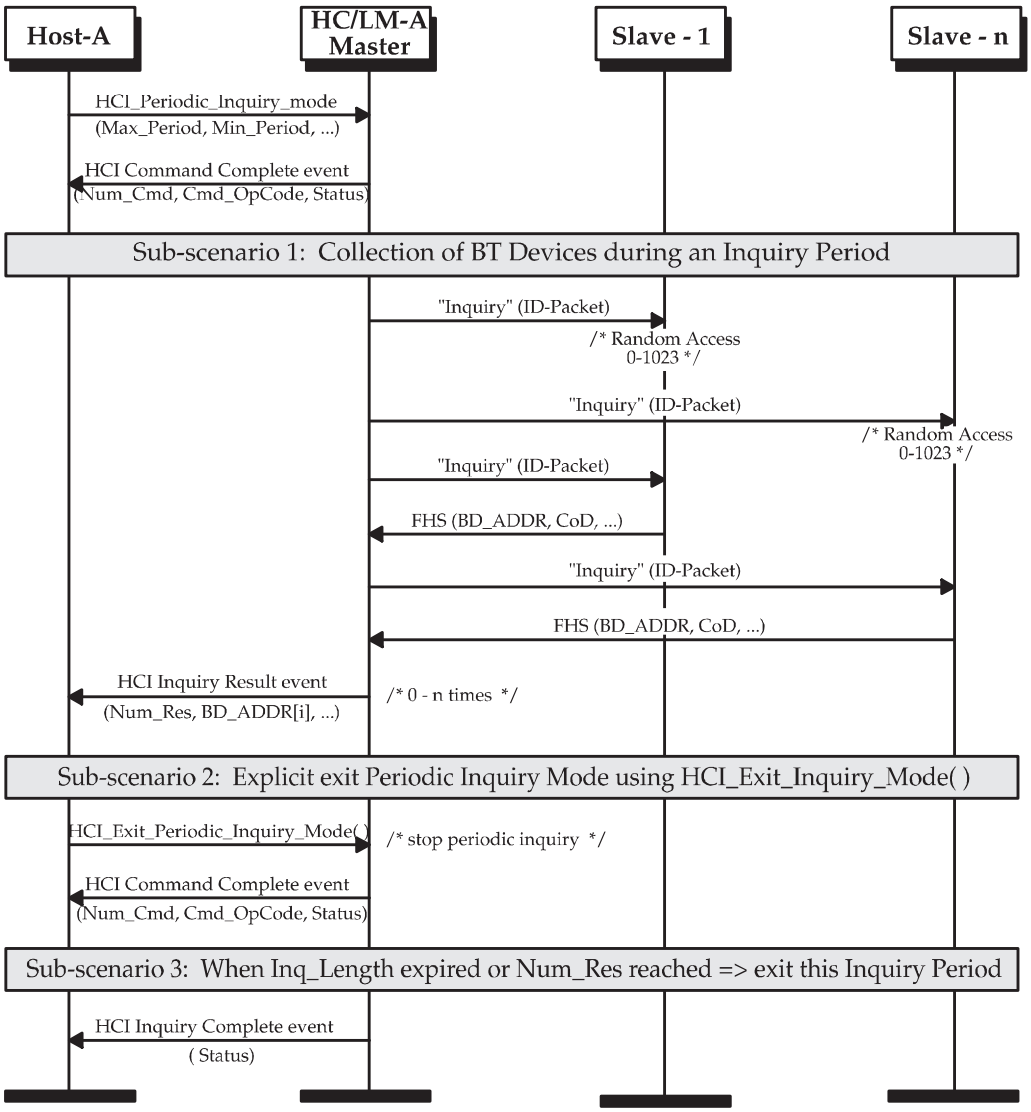


Figure G.3—Periodic inquiry

Copyright © 2002 IEEE. All rights reserved.

G.3 ACL connection establishment and detachment

The overview of the ACL connection establishment and detachment is shown in Figure G.4.

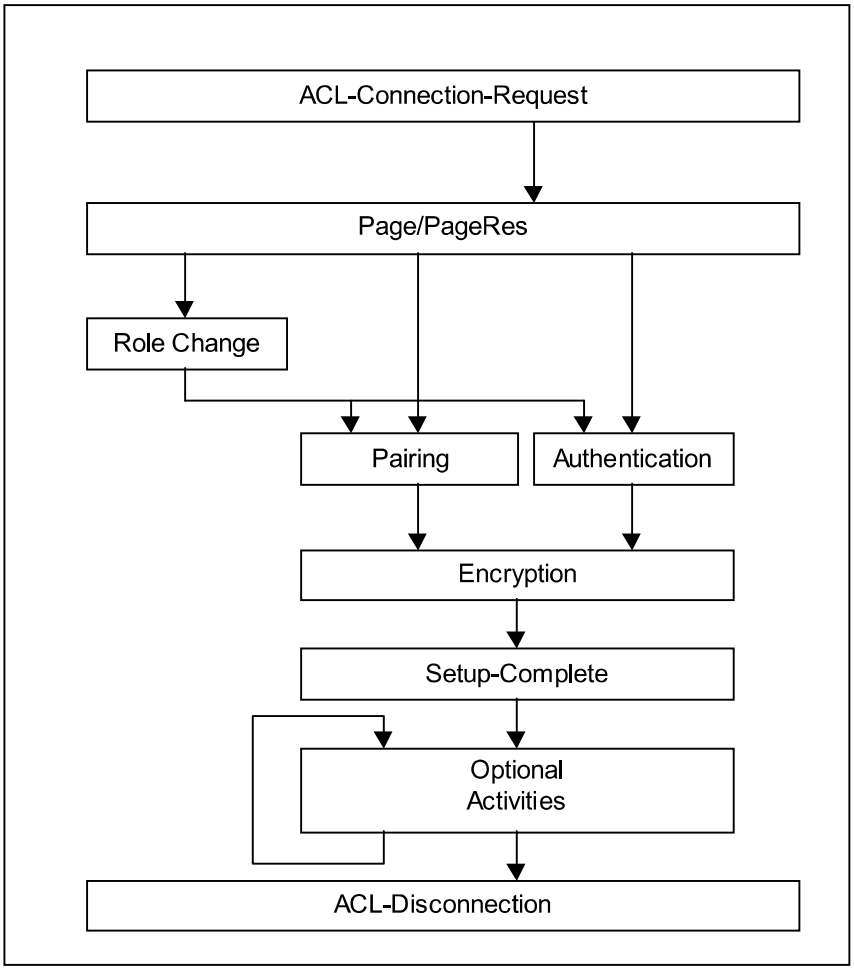


Figure G.4—Overview of ACL connection establishment and detachment

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1139 of 1169

G.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 subscenarios are shown in Figure G.5.

— **Subscenario 1:** Slave rejects the 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.

— **Subscenario 2:** Slave accepts the 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.

— **Subscenario 3:** Slave accepts the 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 that 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*/.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1140 of 1169

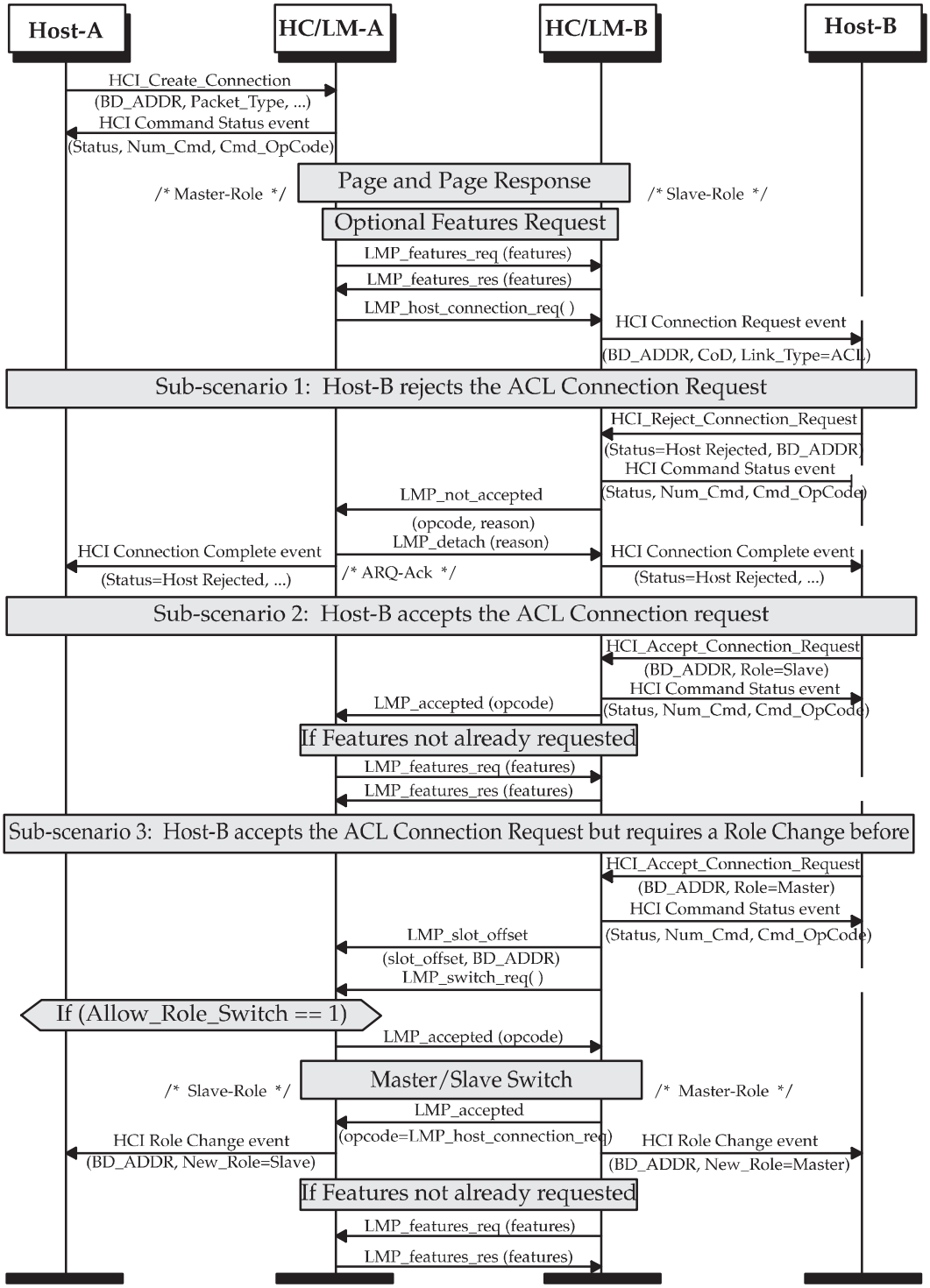


Figure G.5—ACL connection request phase

Copyright © 2002 IEEE. All rights reserved.

G.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.

G.3.2.1 Pairing

If authentication is required and the Bluetooth devices to be connected do not have a common link key, the pairing procedure on the LM level will be executed using the PIN input from Host. See Figure G.6. During the pairing, the link key creation and mutual authentication procedures will be done. Note that the created Link Key can be stored either in the Bluetooth device or in the Host.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1142 of 1169

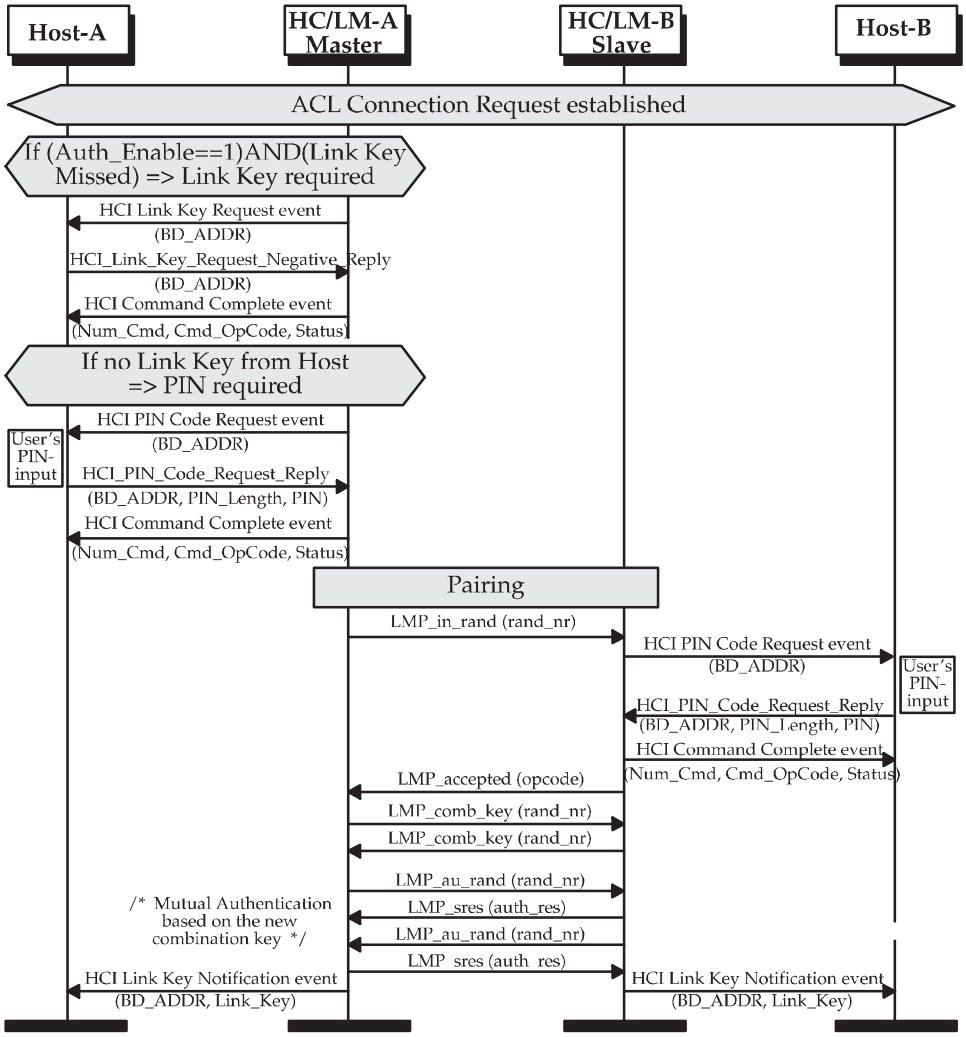


Figure G.6—ACL connection setup with pairing

G.3.2.2 Authentication

If a common link key already exists between the Bluetooth devices, pairing is not needed. Note that a Link Key created during pairing can be stored either in the Bluetooth device or in the Host. If the parameter Authentication_Enable is set, the authentication procedure has to be executed. In Figure G.7, the MSC only shows the case when Authentication_Enable is set on both sides.

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 1143 of 1169

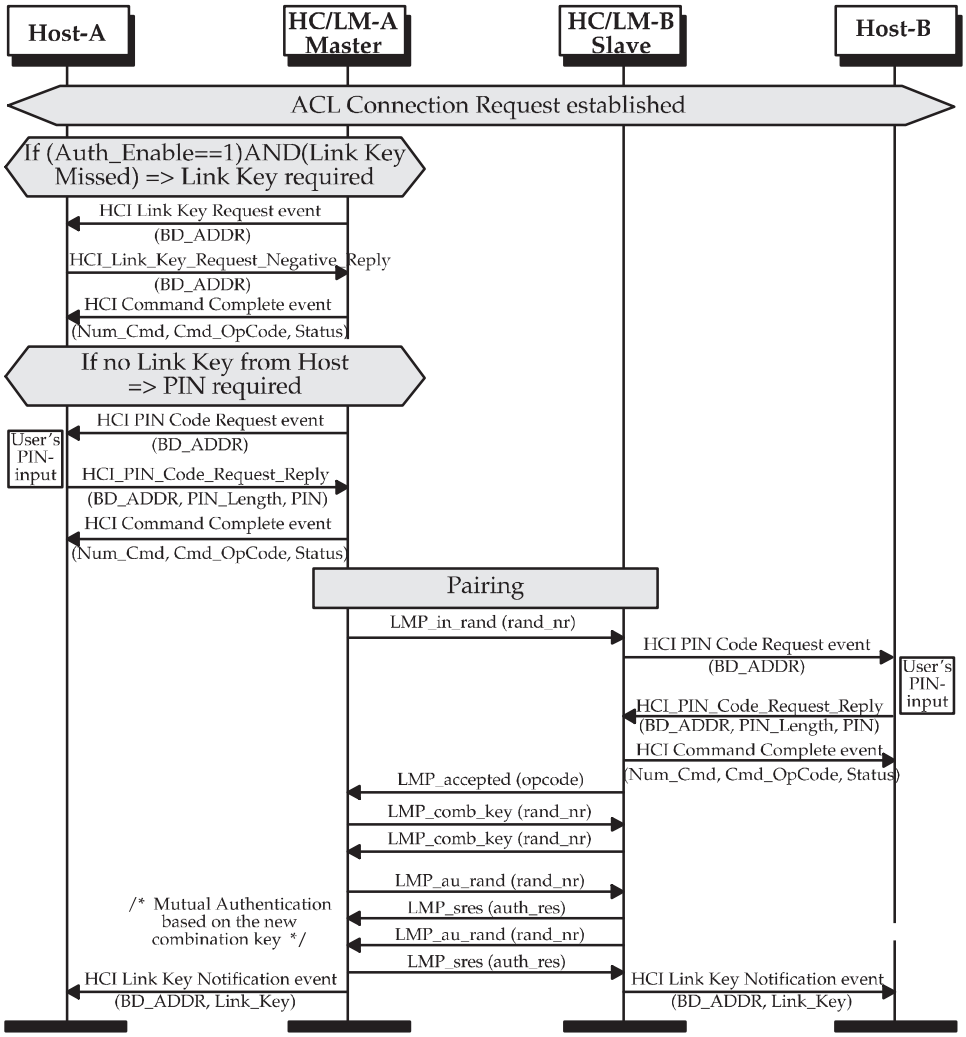


Figure G.7—ACL connection setup with authentication

G.3.3 Encryption and connection setup complete

Once the pairing/authentication procedure is successful, the encryption procedure will be started. In Figure G.8, the MSC shows only how to set up an encrypted point-to-point connection (Encryption_Mode = 1). Note that an encrypted connection requires an established common link key.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1144 of 1169

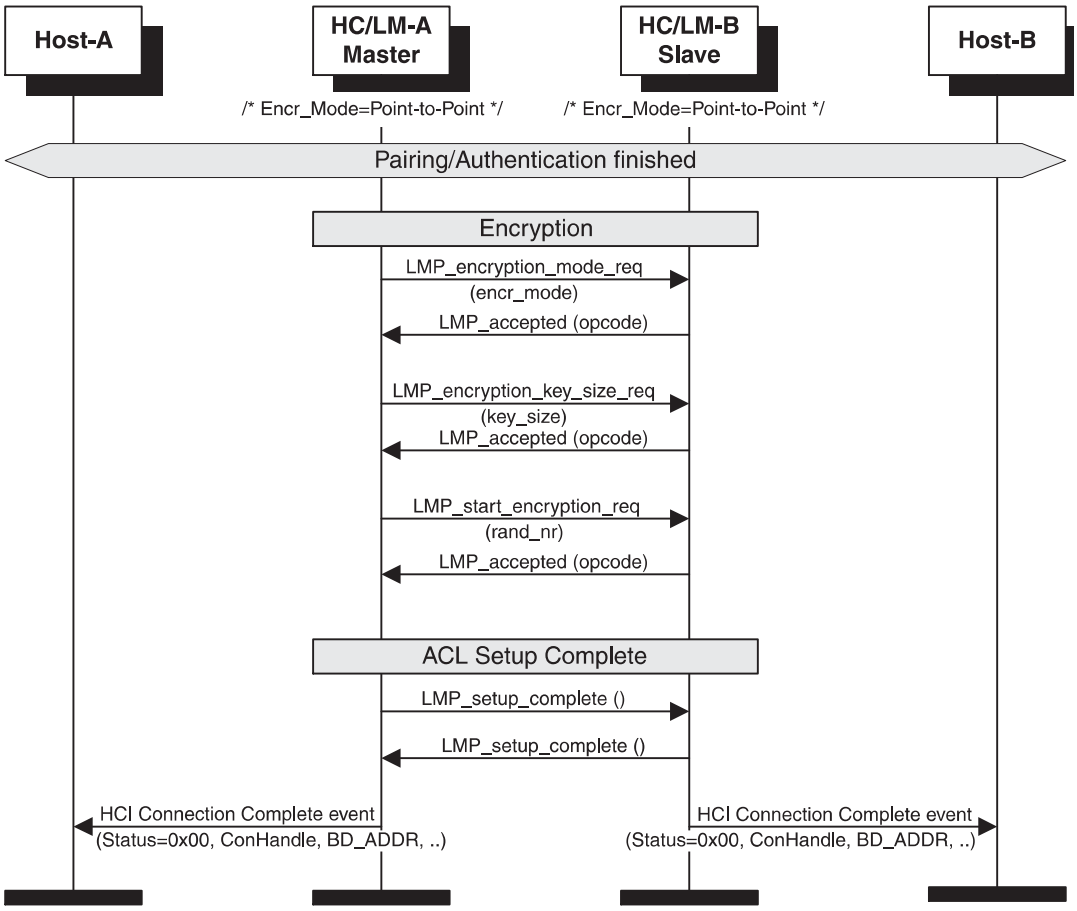


Figure G.8—Encryption and setup complete

G.3.4 ACL disconnection

At any time, an established ACL Connection can be detached by an HCI_Disconnect (Connection_Handle, Reason). See Figure G.9. If one or several SCO Connections exist, they shall first be detached before the ACL Connection can be released.

Note that the disconnection procedure is one-sided and does not 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).

Copyright © 2002 IEEE. All rights reserved.

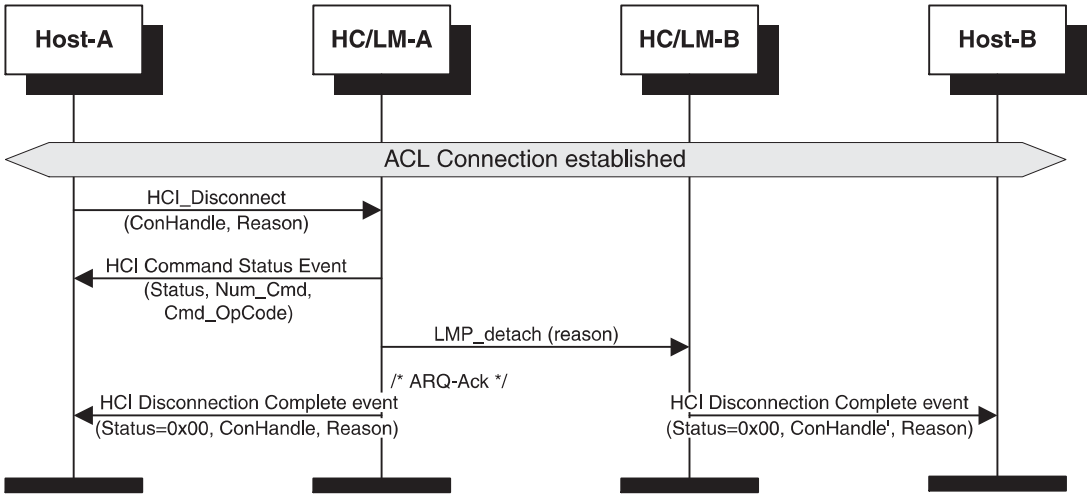


Figure G.9—ACL disconnection

G.4 Optional activities after ACL connection establishment

G.4.1 Authentication requested

Authentication can be explicitly executed at any time after an ACL Connection has been established. See Figure G.10. If the Link Key was missed in HC/LM, the Link Key will be required from the Host, as in the authentication procedure (see G.3.2.2).

Note that if the HC/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 ACL Connection Setup with Pairing (see G.3.2.1) will be used.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1146 of 1169

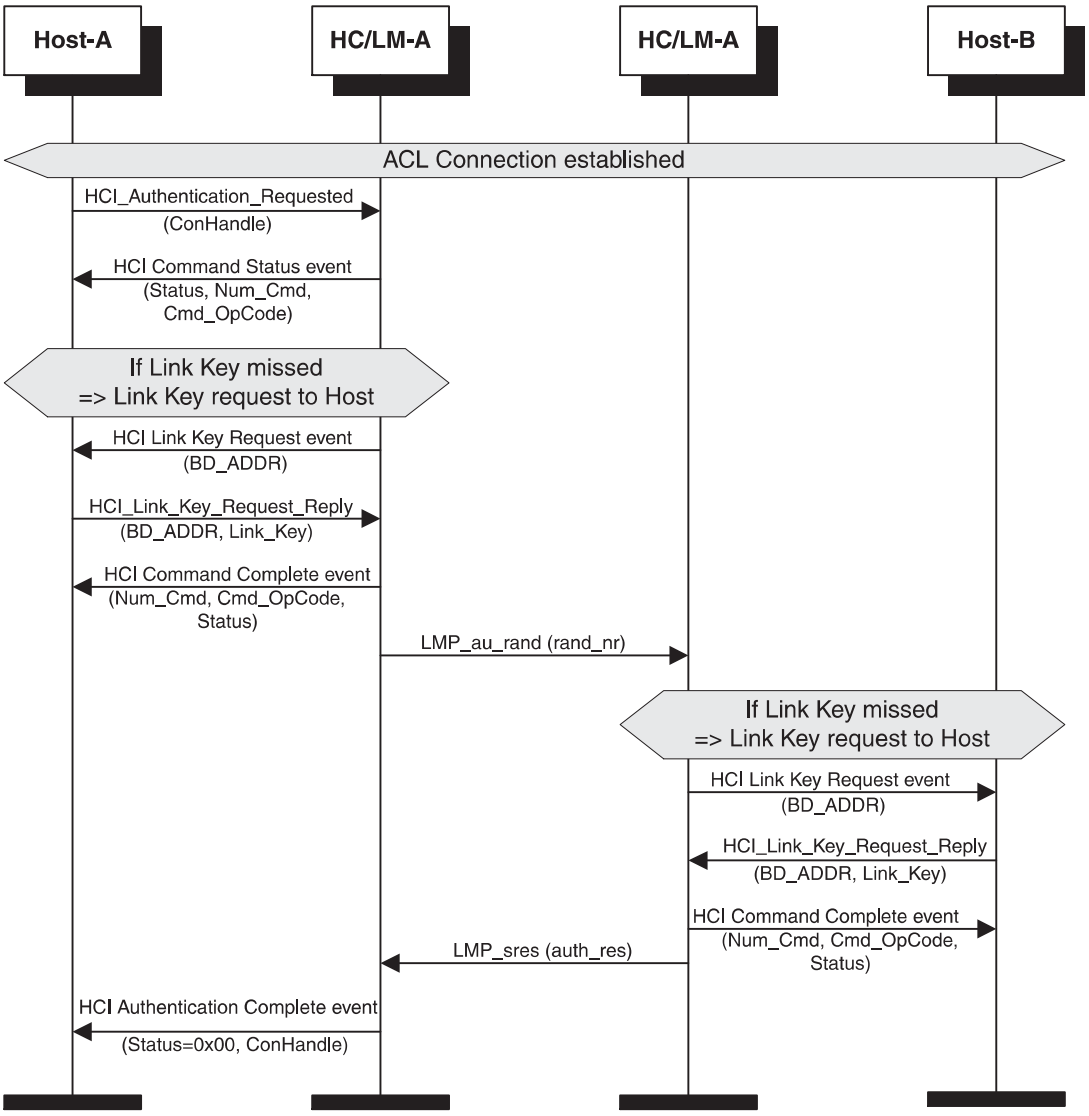


Figure G.10—Authentication requested

G.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 G.11 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.

Copyright © 2002 IEEE. All rights reserved.

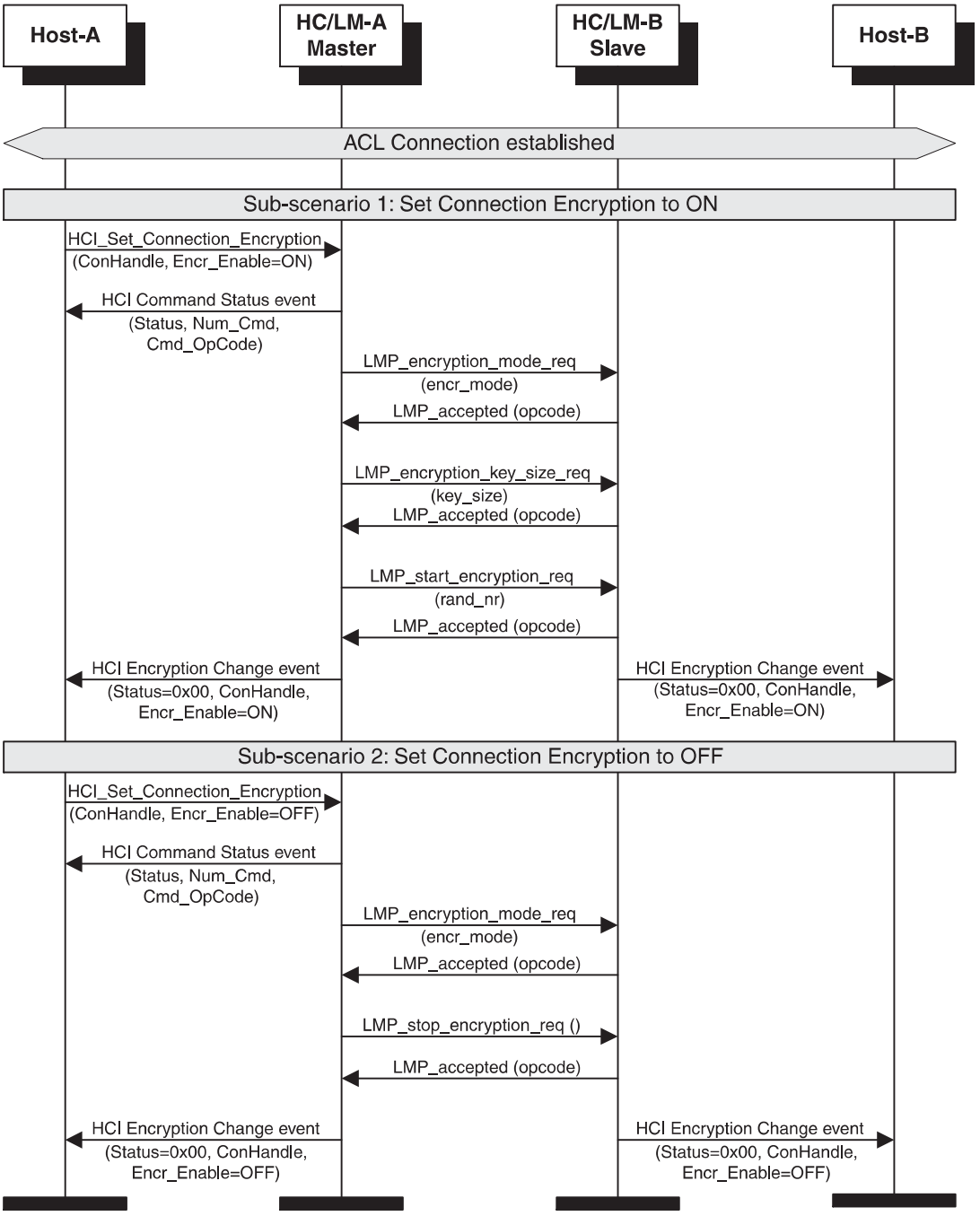


Figure G.11—Set connection encryption

Copyright © 2002 IEEE. All rights reserved.

G.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 Bluetooth devices. See Figure G.12.

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

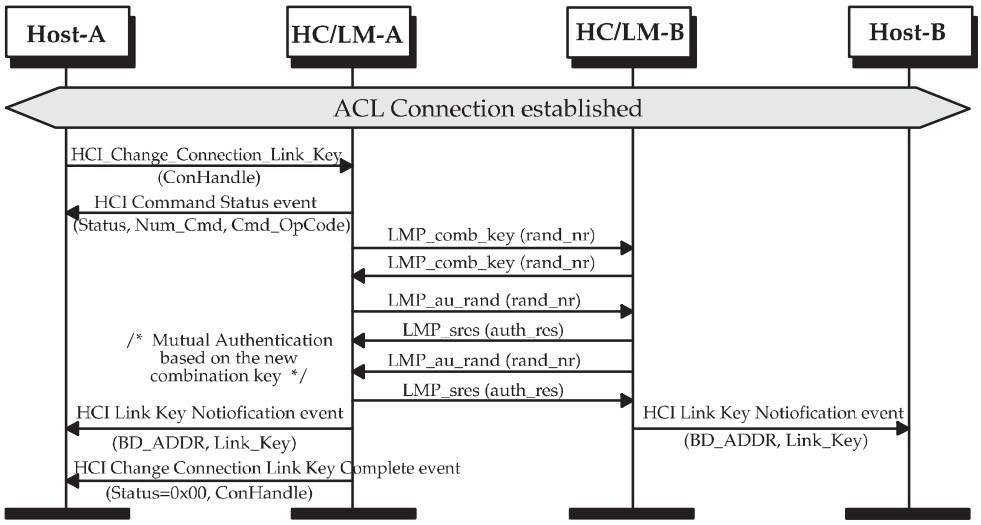


Figure G.12—Change connection link key

G.4.4 Master link key

Figure G.13 shows how the Host can explicitly switch between the temporary Link Key and the semipermanent Link Key. Since this command can only be used for the master, the Link Key switch will affect all connections.

NOTE—If encryption was enabled, it is the task of the master to restart the encryption separately for each slave.

Copyright © 2002 IEEE. All rights reserved.

.

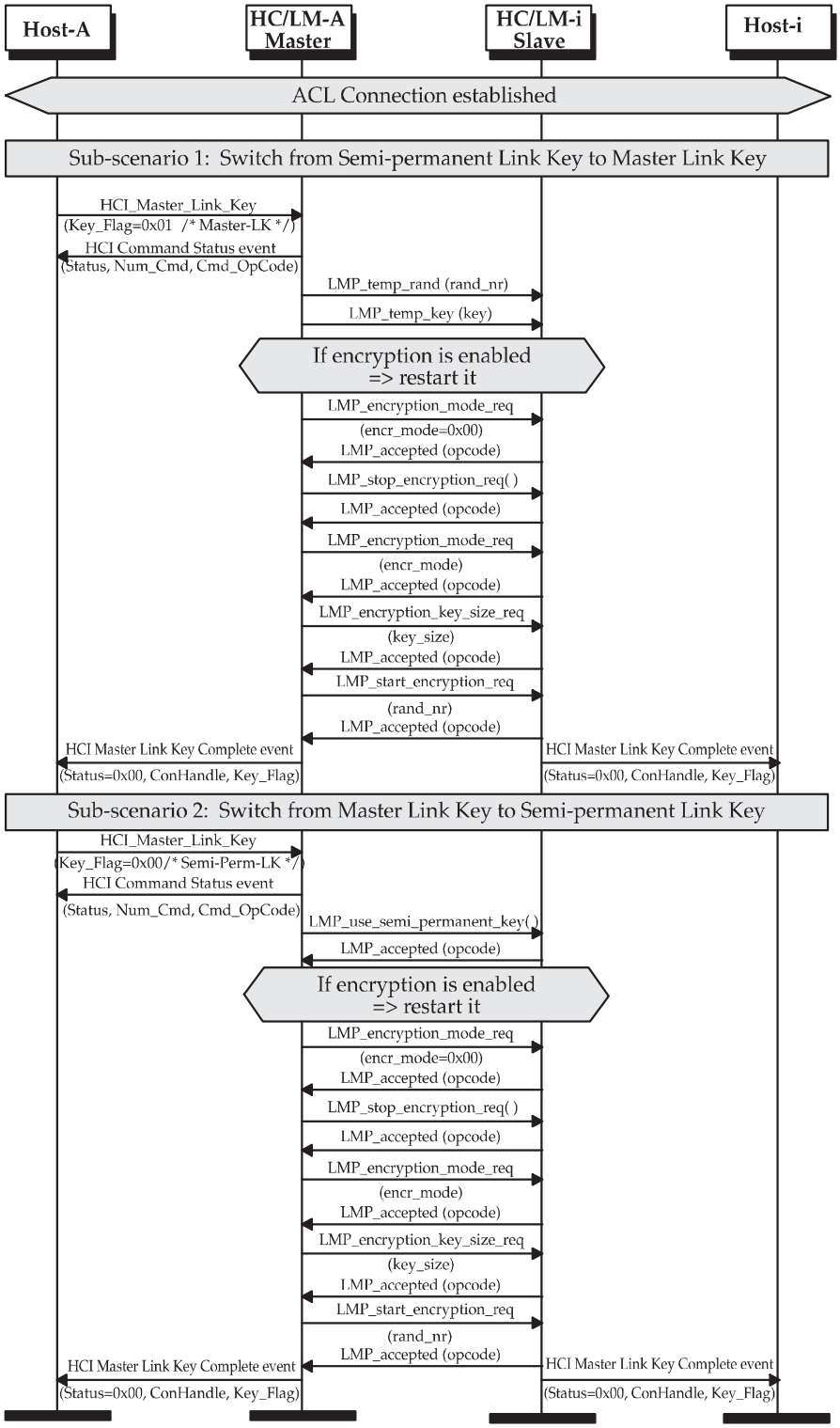


Figure G.13—Master link key

Copyright © 2002 IEEE. All rights reserved.

G.4.5 Read remote supported features

Using the command HCI_Read_Remote_Supported_Features (Connection_Handle), the supported LMP Features of a remote Bluetooth device can be read. These features contain supported packet types, supported modes, supported audio coding modes, etc. See Figure G.14.

Note that if the LMP Features were 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.

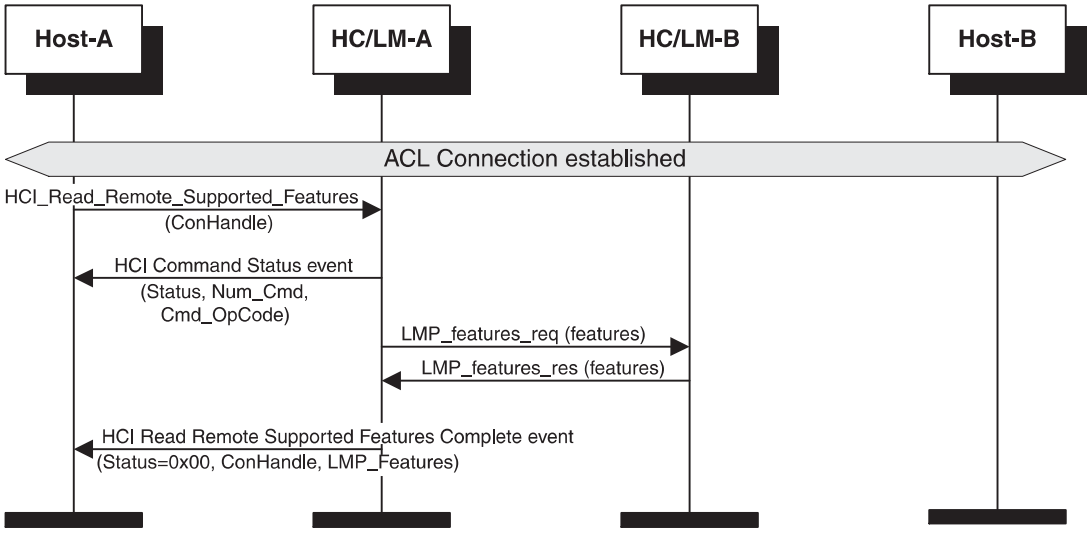


Figure G.14—Read remote supported features

G.4.6 Read clock offset

Using the command HCI_Read_Clock_Offset (Connection_Handle) the Bluetooth master can read the Clock Offset of the Bluetooth slaves. See Figure G.15. 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.

Copyright © 2002 IEEE. All rights reserved.

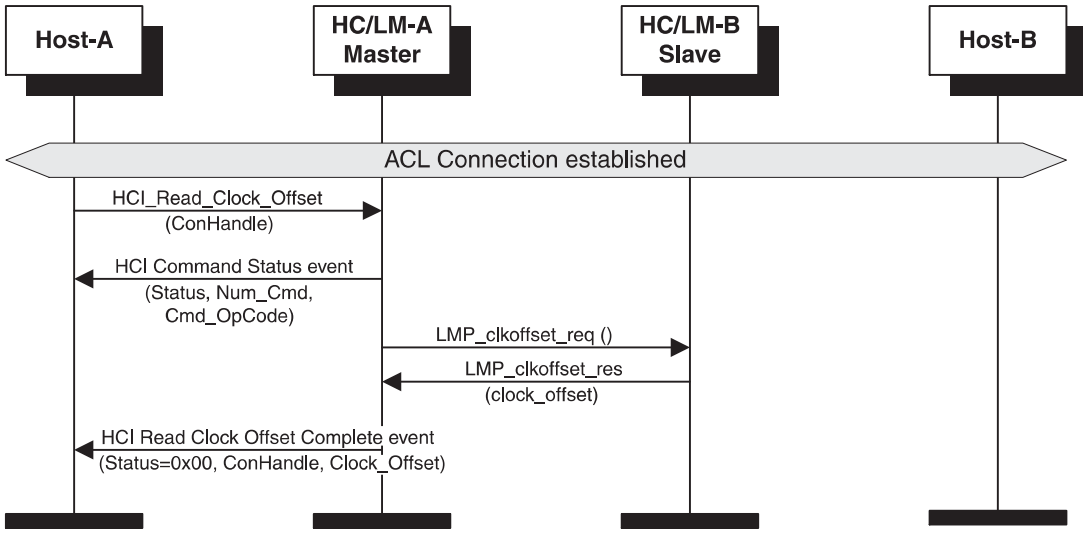


Figure G.15—Read clock offset

G.4.7 Read remote version information

Using the HCI_Read_Remote_Version_Information (Connection_Handle) the version information consisting of LMP_Version, Manufacturer_Name and LMP_Subversion from the remote Bluetooth device can be read. See Figure G.16.

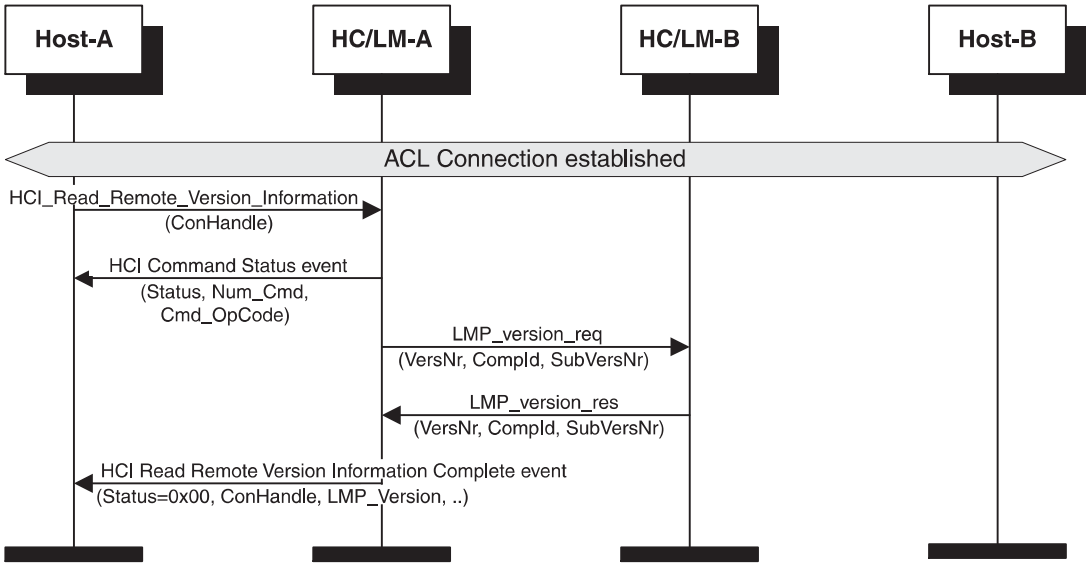


Figure G.16—Read remote version information

Copyright © 2002 IEEE. All rights reserved.

G.4.8 QoS setup

To set up the QoS, the command HCI_QoS_Setup (Connection_Handle, Flags, Service_Type, Token_Rate, Peak_Bandwidth, Latency, Delay_Variation) is used. See Figure G.17.

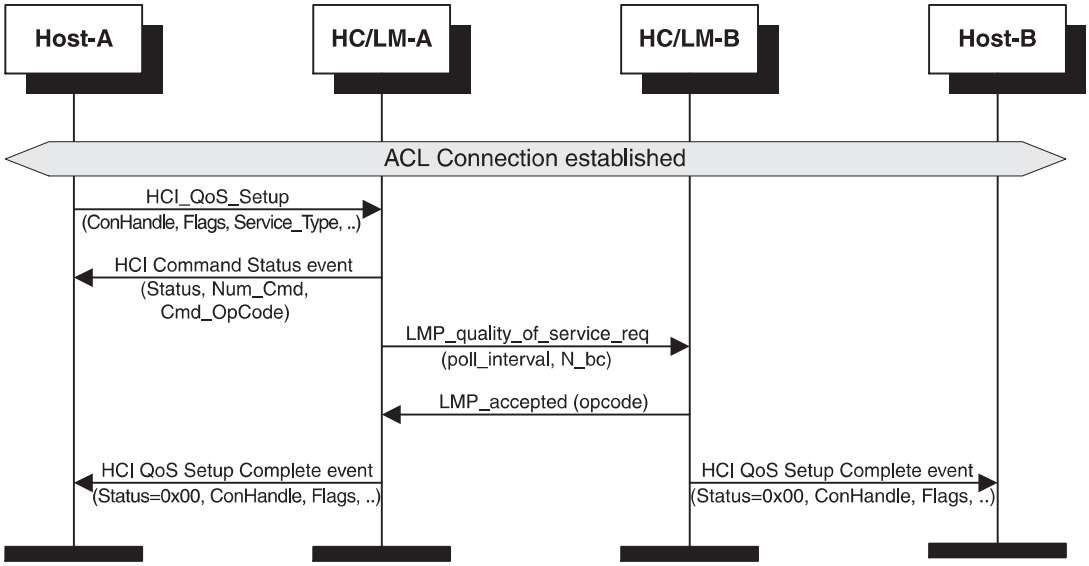


Figure G.17—QoS setup

G.4.9 Switch Role

The command HCI_Switch_Role (BD_ADDR, Role) can be used to explicitly switch the current role of the local Bluetooth device for a particular connection with the specified Bluetooth device (BD_ADDR). The local HC/LM has to check whether the switch is performed or not. See Figure G.18.

Copyright © 2002 IEEE. All rights reserved.

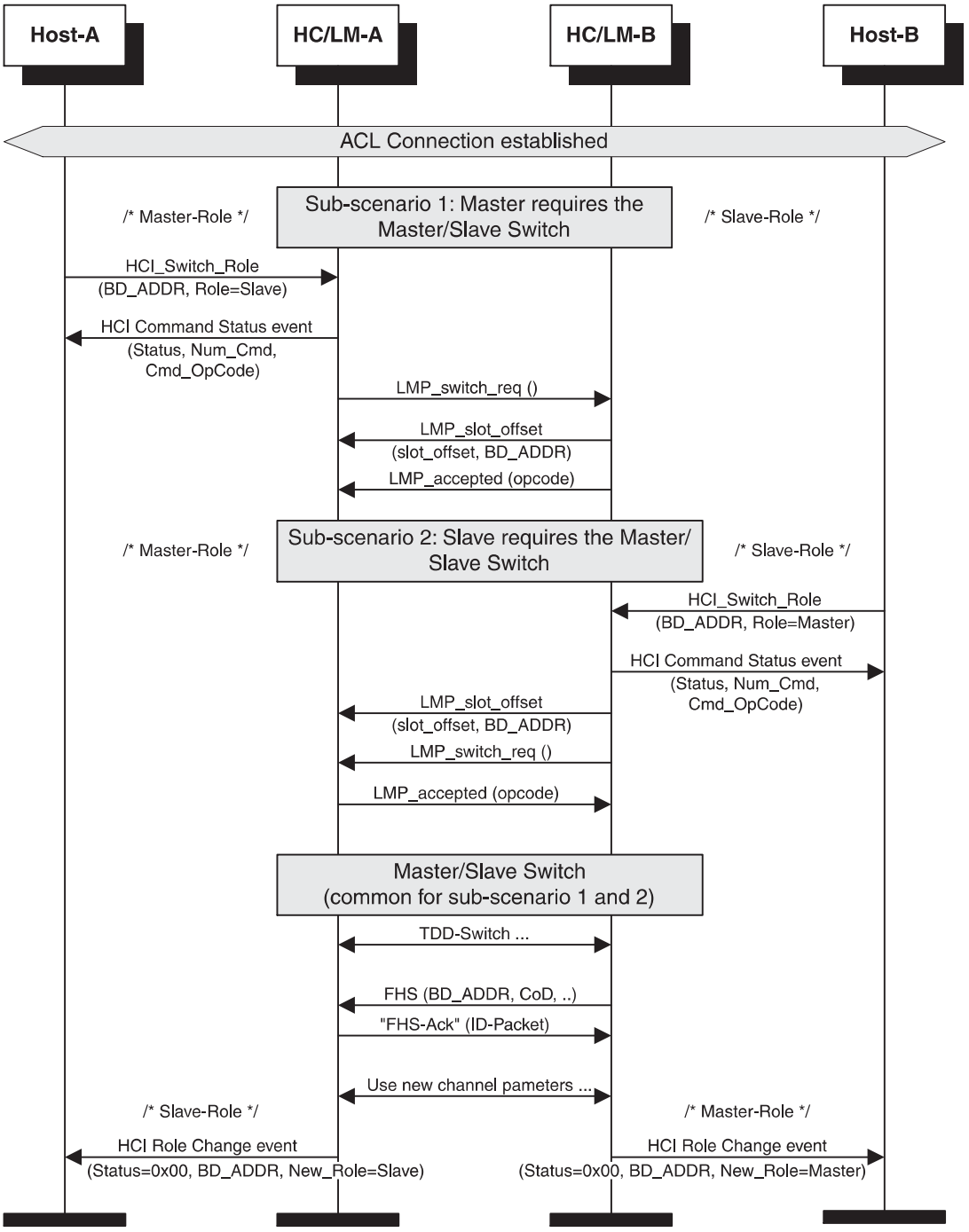


Figure G.18—Switch role

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1154 of 1169

G.5 SCO connection establishment and detachment

G.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 that 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 Bluetooth device, a separate SCO Connection Handle is needed.

G.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. See Figure G.19. The specified Connection_Handle is related to the ACL Connection that must have been created before the HCI_Add_SCO_Connection is issued.

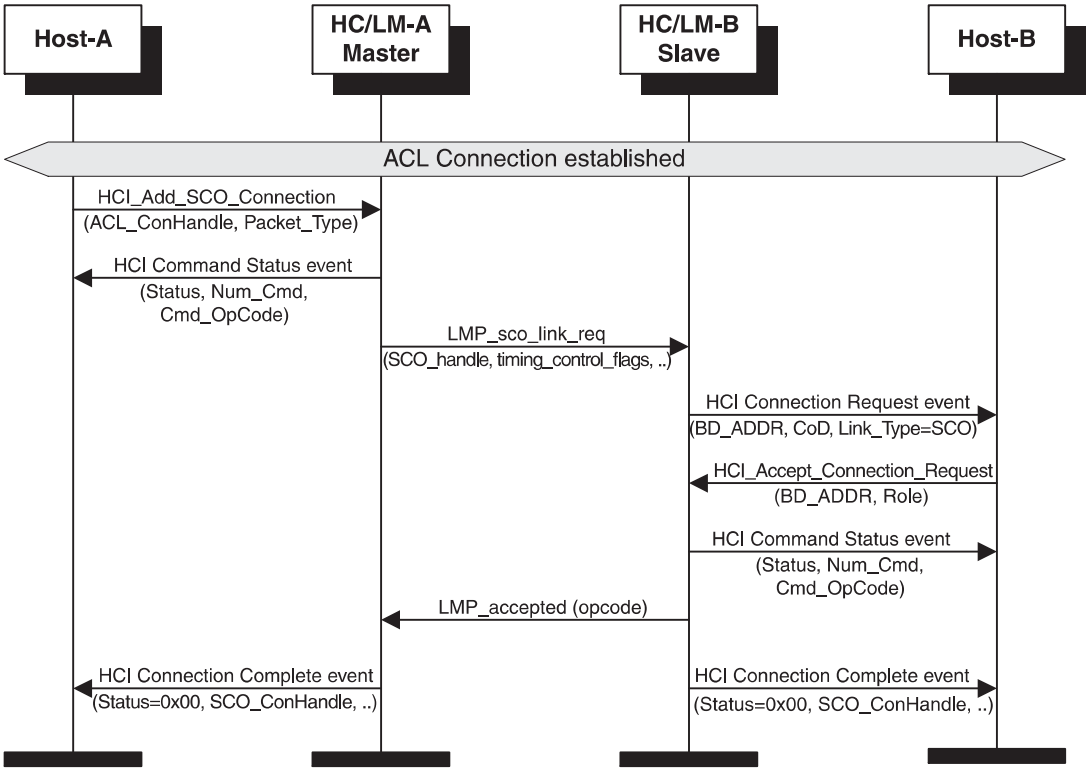


Figure G.19—SCO connection setup (activated from master)

Copyright © 2002 IEEE. All rights reserved.

G.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 Bluetooth device is a Bluetooth Slave. In Figure G.20, the specified Connection_Handle belongs to the established ACL Connection between the Bluetooth devices. Compared to G.5.1.1, the only difference is that the HC/LM Slave starts the SCO Setup with LMP_sco_link_req first.

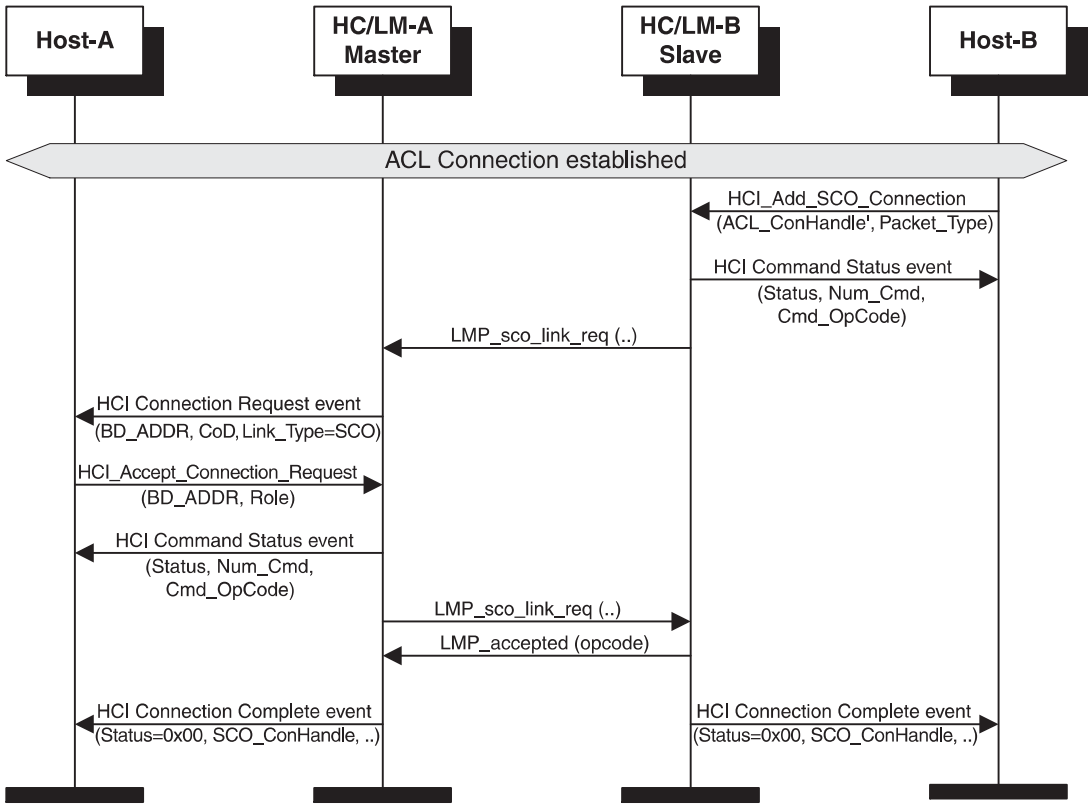


Figure G.20—SCO connection setup (activated from slave)

G.5.2 SCO disconnection

An established SCO Connection can be detached at any time. Since several SCO Connections can exist between a Bluetooth Master and a Bluetooth Slave, an SCO Disconnection only removes the SCO Link with the specified SCO Connection Handle. The other SCO Connections will still exist. See Figure G.21.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1156 of 1169

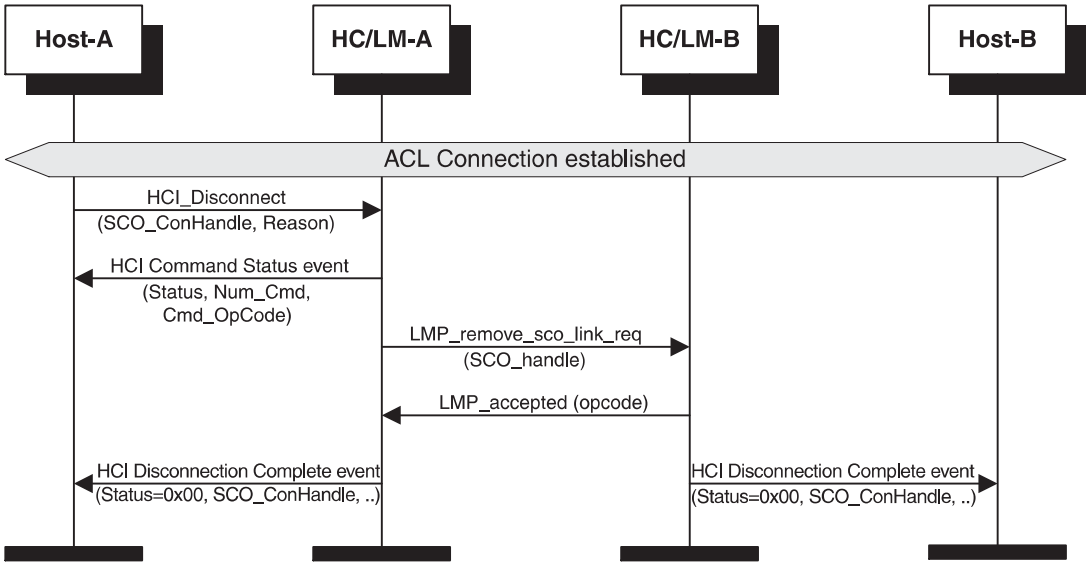


Figure G.21—SCO disconnection

G.6 Special modes: sniff, hold, park

Entry into sniff, hold, or park mode requires an established ACL Connection. Table G.1 summarizes the modes and the Bluetooth role that can request, force, activate, or exit the modes.

Table G.1—Summary of modes (Sniff, Hold, Park)

Commands	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

G.6.1 Sniff mode

Sniff mode is used when a slave shall participate in the piconet only in a sniff interval. See Figure G.22. 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.

Copyright © 2002 IEEE. All rights reserved.

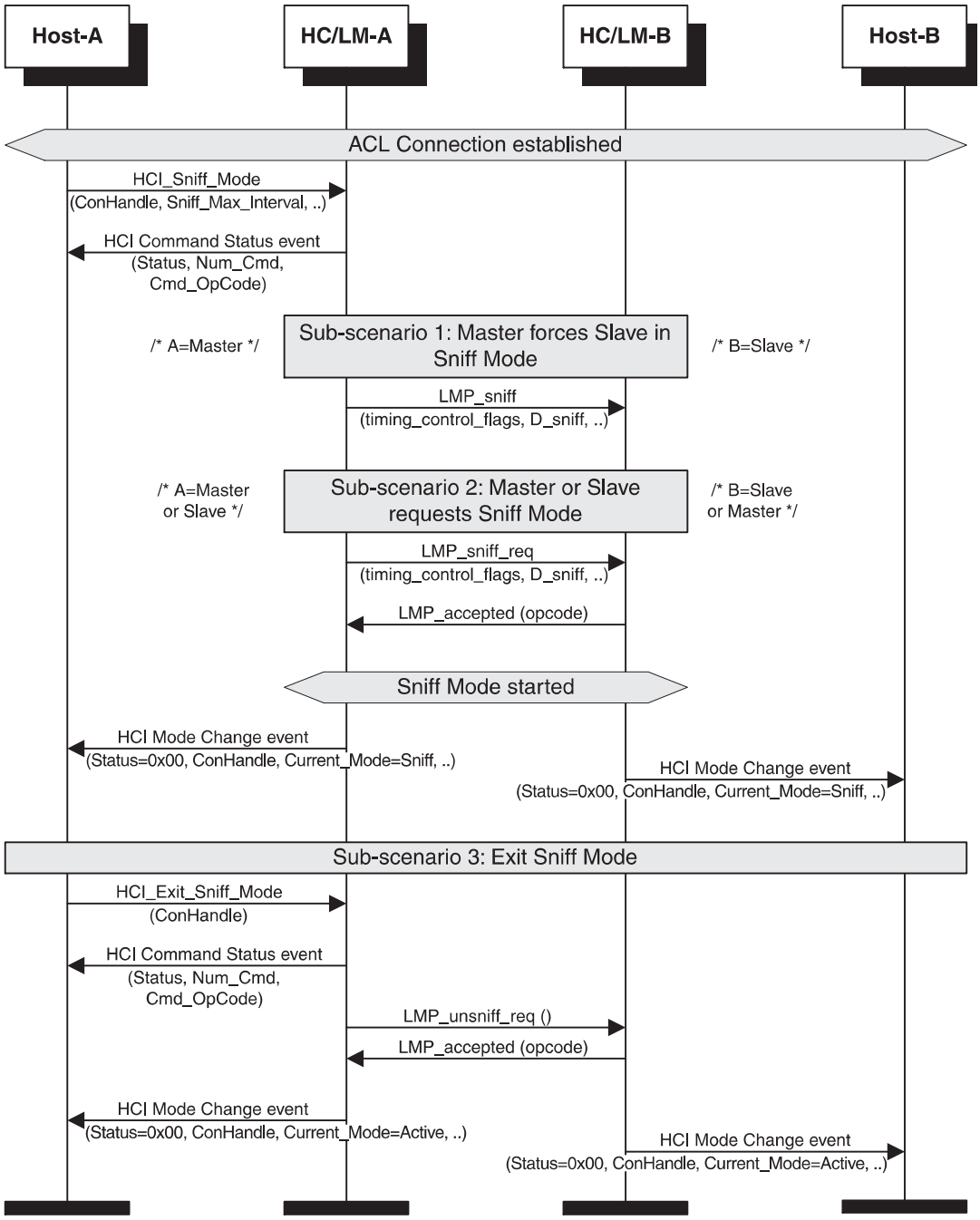


Figure G.22—Sniff mode

Copyright © 2002 IEEE. All rights reserved.

G.6.2 Hold mode

Hold mode is useful when a Bluetooth device does not want to participate in the connection for a Hold Mode Length. See Figure G.23. 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 that the Hold mode is exited when the Hold Mode Length has expired, so it is no guarantee that the remote Bluetooth device is immediately active.

G.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.

G.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 Bluetooth Slaves. In Park Mode, the Bluetooth Slave gives up its AM_ADDR. See Figure G.24.

Copyright © 2002 IEEE. All rights reserved.

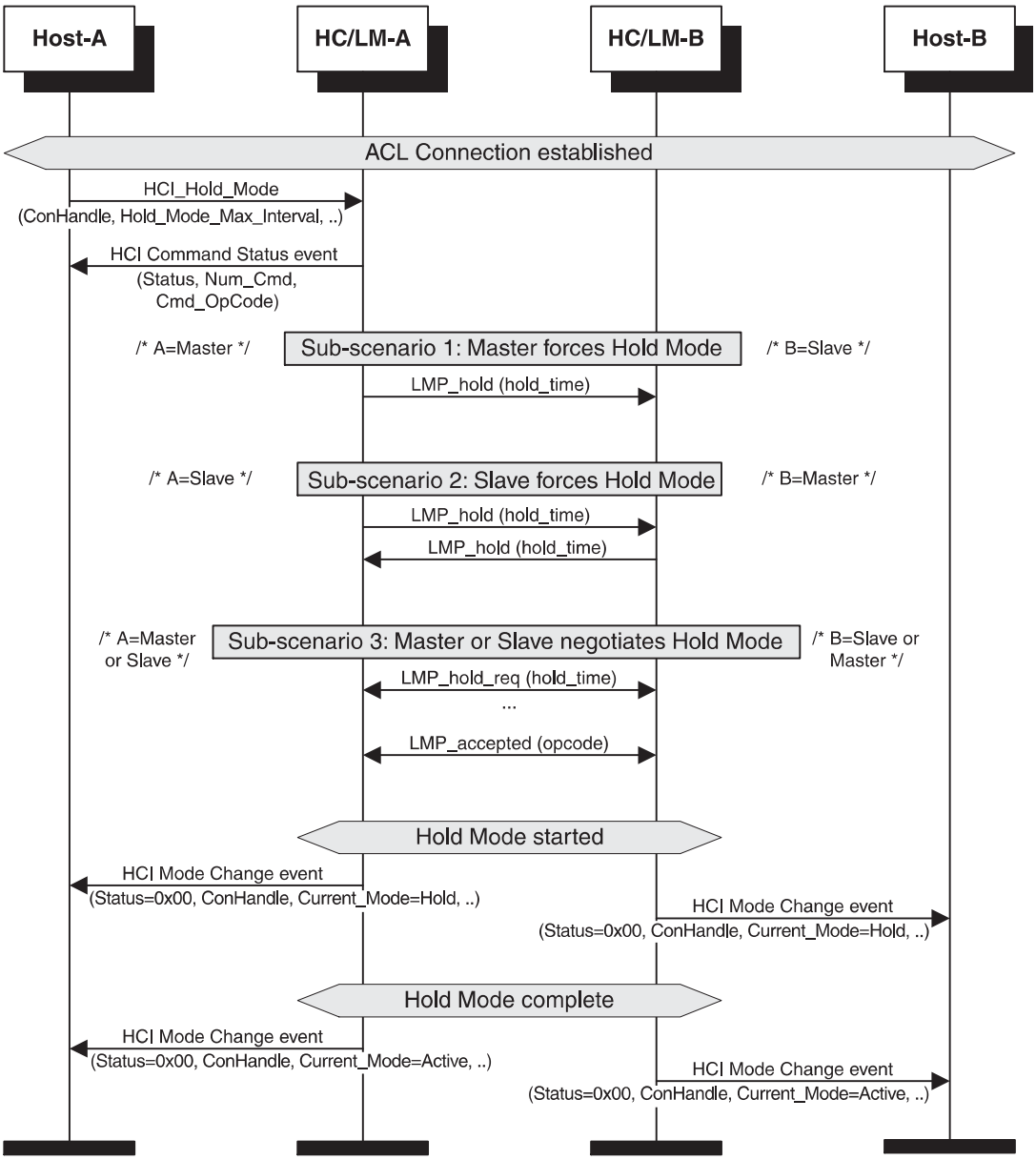


Figure G.23—Hold mode

Copyright © 2002 IEEE. All rights reserved.

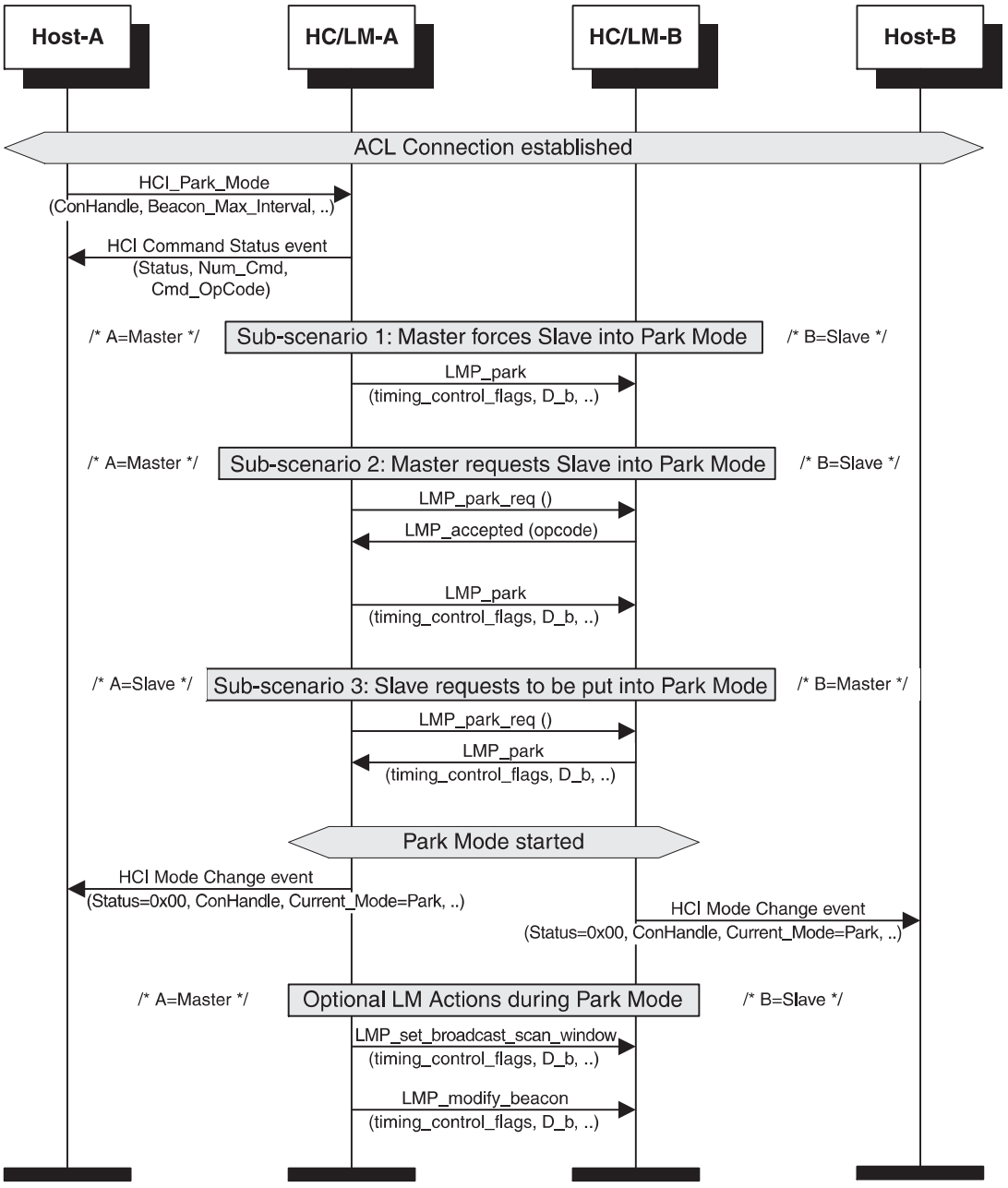


Figure G.24—Enter park mode

G.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. See Figure G.25. A parked Bluetooth 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 Bluetooth slave.

Copyright © 2002 IEEE. All rights reserved.

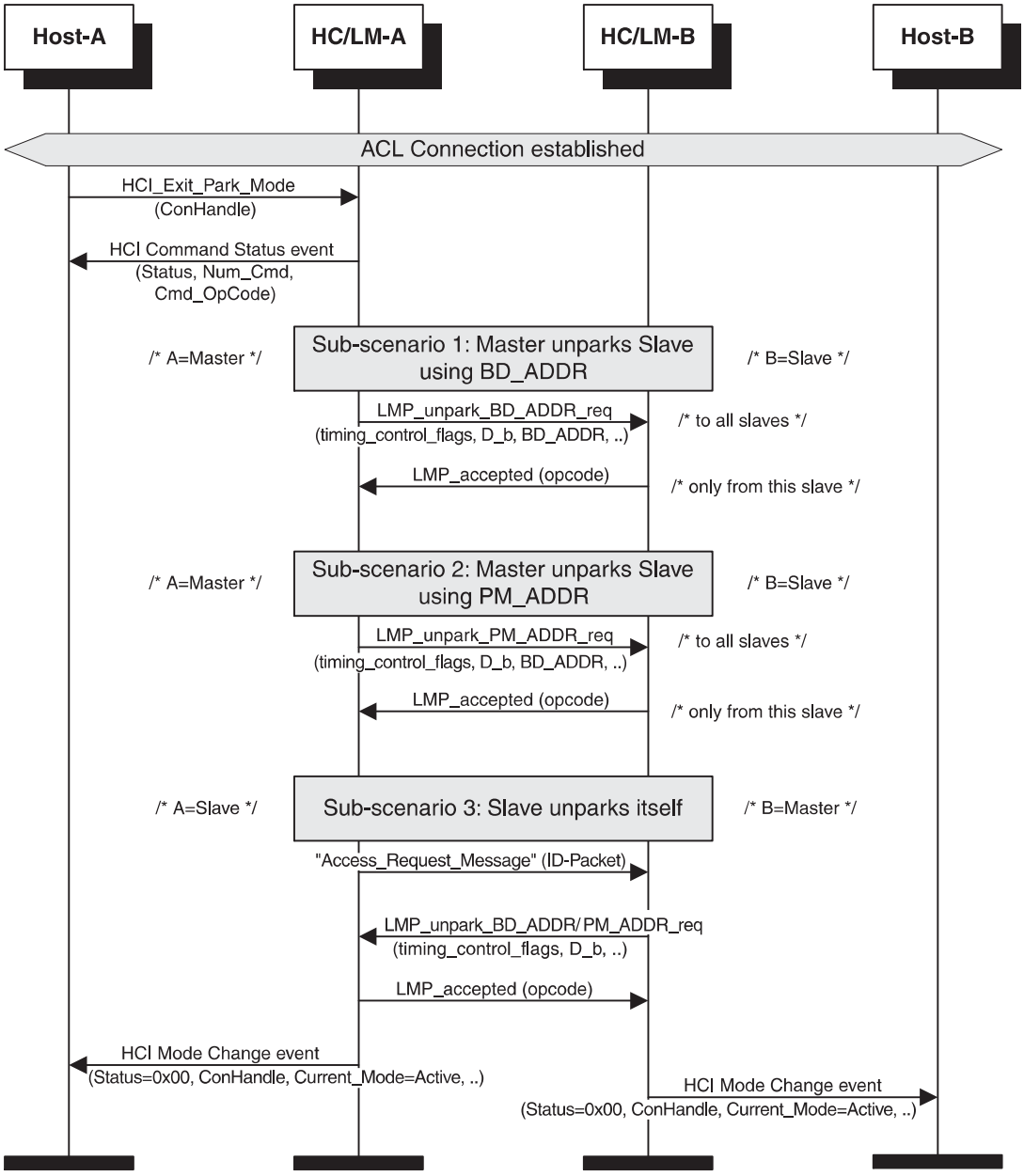


Figure G.25—Exit park mode

Copyright © 2002 IEEE. All rights reserved.

G.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 G.26).

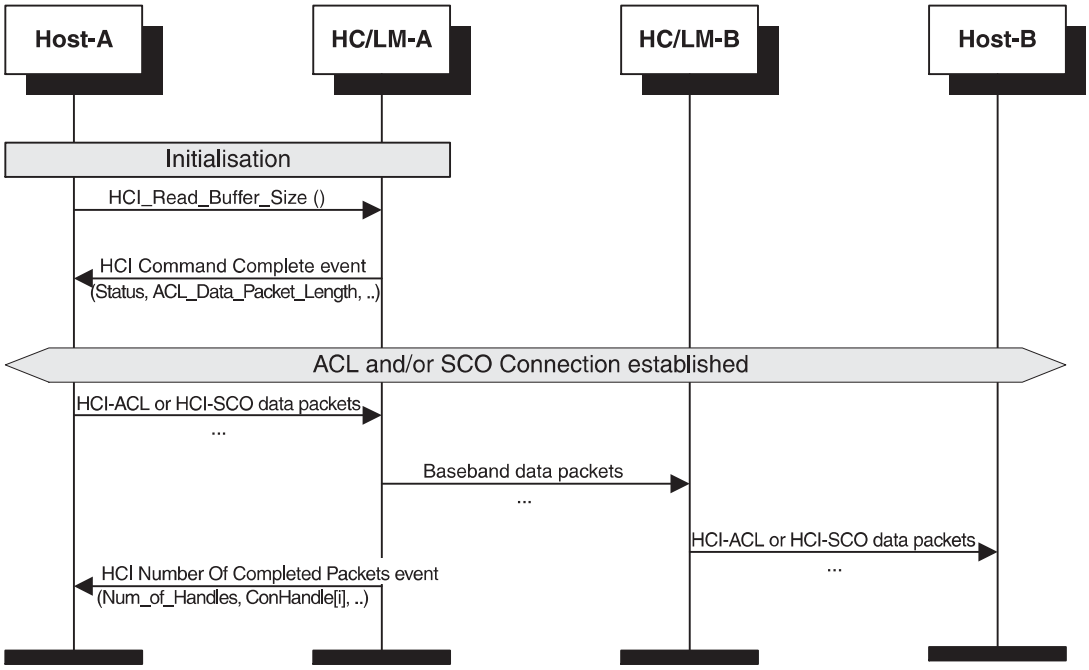


Figure G.26—Host to HC flow control

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 whether to apply the HC to Host flow control or not. For flow control itself, the following commands are used:

— 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)

— HCI_Host_Number_Of_Completed_Packets (Number_of_Handles, Connection_Handle[I], Host_Num_Of_Completed_Packets[I])

For details see Figure G.27.

Copyright © 2002 IEEE. All rights reserved.

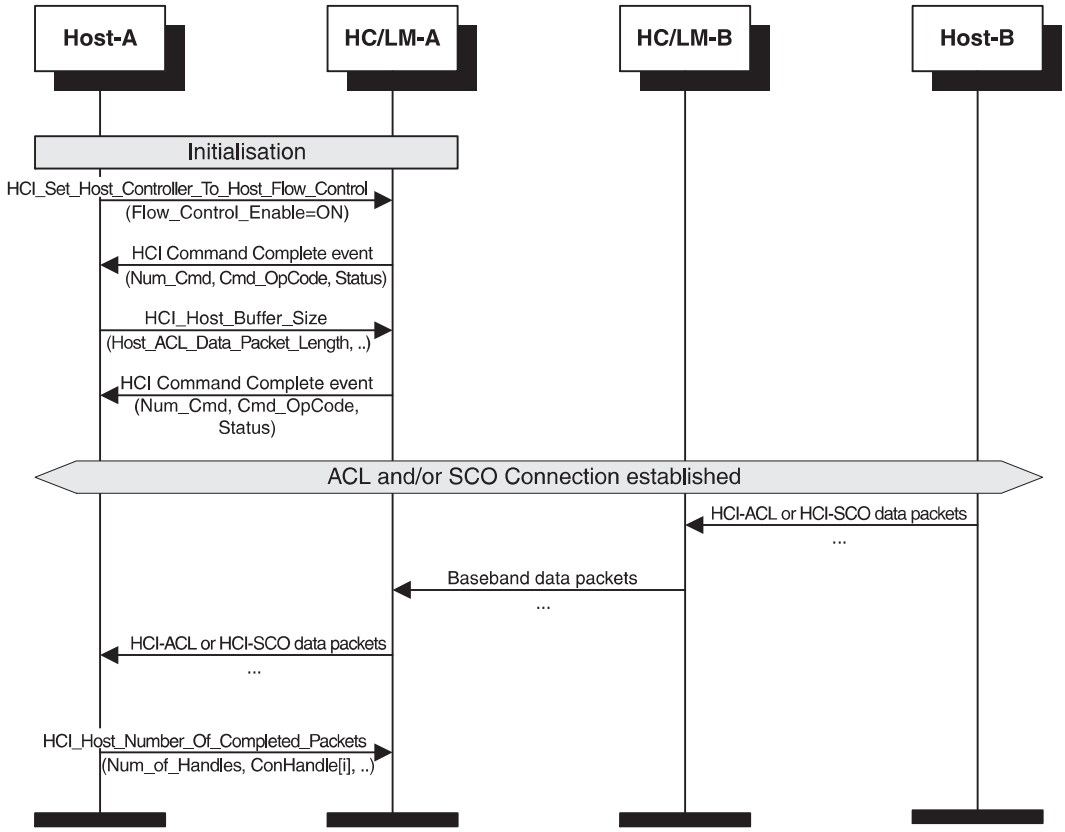


Figure G.27—HC to host flow control

G.8 Loopback mode

G.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. See Figure G.28.

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 resend HCI ACL and HCI SCO packets to the HC. To exit the local Loopback mode, HCI_Write_Loopback_Mode (Loopback_Mode=0x00) or HCI_Reset ( ) will be used.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1164 of 1169

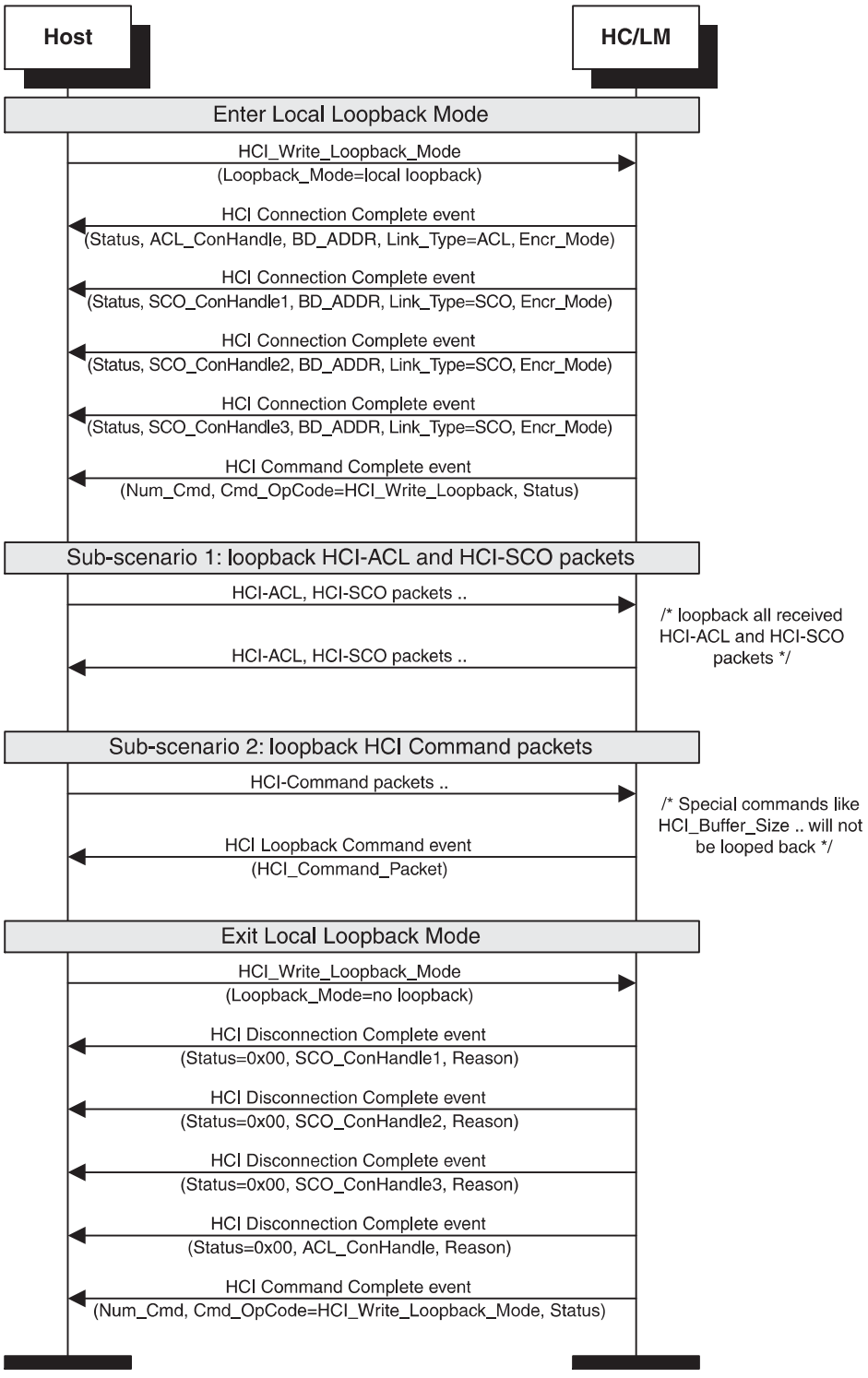


Figure G.28—Local loopback mode

Copyright © 2002 IEEE. All rights reserved.

G.8.2 Remote loopback mode

The remote Loopback mode is used to loopback all received Baseband ACL and SCO Data received from a remote Bluetooth 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). See Figure G.29.

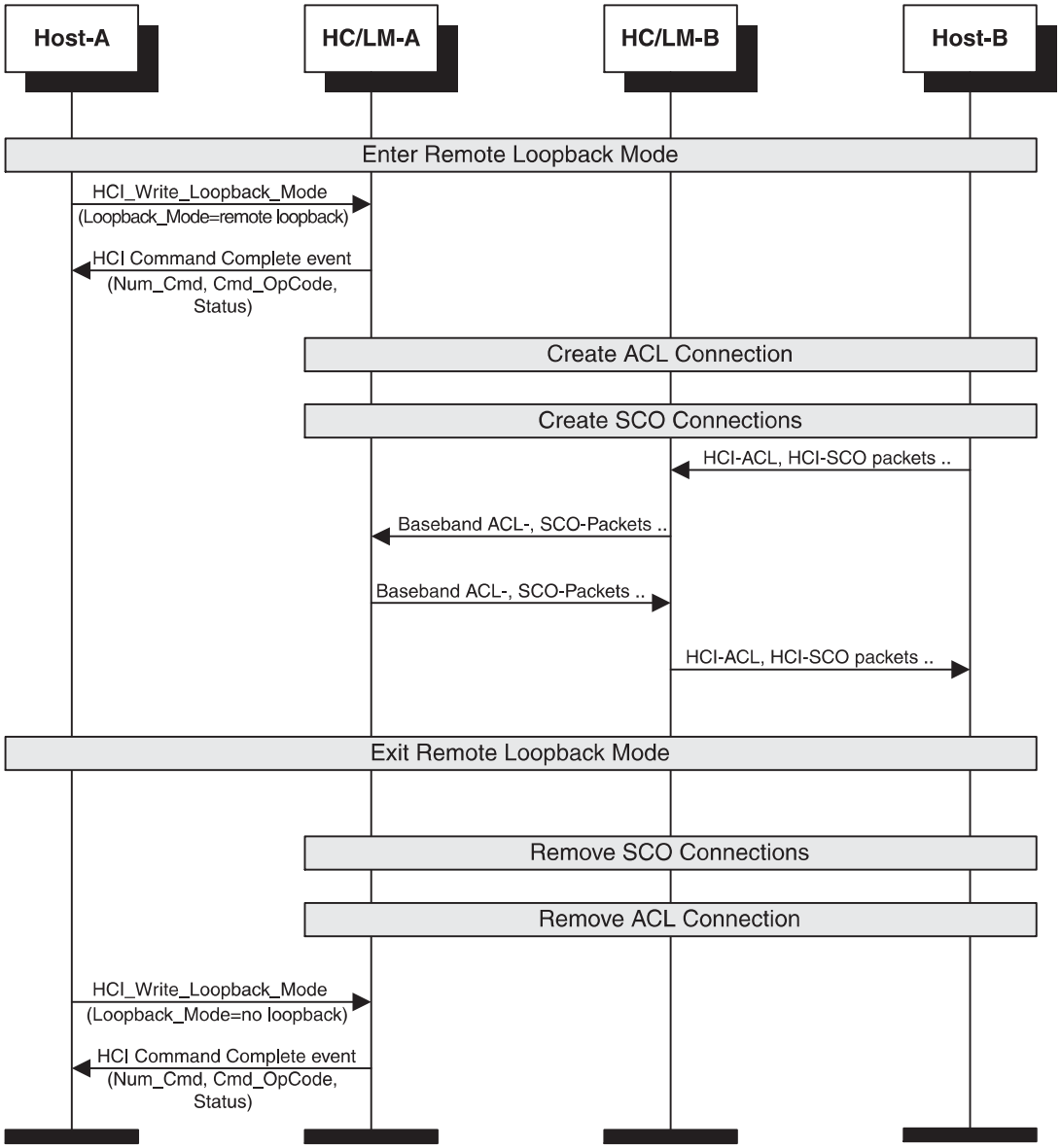


Figure G.29—Remote loopback mode

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1166 of 1169

Copyright © 2002 IEEE. All rights reserved.

Annex H

(informative)

Bibliography

[B1] Association of Radio Industries and Businesses (ARIB), Documents: RCR STD-33A, ARIB STD-T66.

[B2] European Telecommunications Standards Institute (ETSI), Documents: ETS 300-328, ETS 300-826.

[B3] Federal Communications Commission (FCC), United States, Documents: CFR 47, Part 15, Sections 15.205, 15.209, 15.247, 15.249.

[B4] IEEE 100, The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition.

[B5] Industry Canada, Canada, Document: GL36.

[B6] ITU-T Recommendation Z.105 (10/01), SDL combined with ASN.1 modules (SDL/ASN.1).

[B7] ITU-T Recommendation Z.107 (11/99), SDL with embedded ASN.1.

[B8] ITU-T Recommendation Z.109 (11/99), SDL combined with UML.

[B9] ITU-T Recommendation Z.120 (11/99), Message sequence chart (MSC).

[B10] ITU-T Recommendation X.208 (11/88), Specification of Abstract Syntax Notation One (ASN.1).

NOTES:

1—Planned for withdrawal as it has been superseded by ITU-T X.680–X.683 (1997).

2—ASN.1 was defined by ITU-T as recommendation X.208 in 1988, and is also standardized as ISO 8824.

[B11] La Reglementation en France pour les Equipements fonctionnant dans la bande de frequences 2.4 GHz “RLAN-Radio Local Area Network,” Documents: SP/DGPT/ATAS/23, ETS 300-328, ETS 300-826.

[B12] Massey and Rueppel, “Linear ciphers and random sequence generators with multiple clocks,” Advances in Cryptology-Preceedings of EUROCRYPT 84 (LNCS 209), pages 74-87, 1985.

[B13] Siep, Tom, *An IEEE Guide: How to Find What You Need in the Bluetooth Spec.*, IEEE Standards Information Network, IEEE Press, 2001, ISBN 0-7381-2635-7.

[B14] Supplemento Del Numero 164 Del Boletin Oficial Del Estado (Published 10 July 1991, Revised 25 June 1993), Documents: ETS 300-328, ETS 300-826.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 1168 of 1169