

IEEE P802.15
Wireless Personal Area Networks

Project	IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)		
Title	Integrated BB, LMP, and L2CAP SDL Model		
Date Submitted	[10 July, 2000]		
Source	[David Cypher] [NIST] [100 Bureau Drive, STOP 8920 Gaithersburg, MD 20899-8920]	Voice: [+1 301 975 4855] Fax: [+ 1 301 590 0932] E-mail: [david.cypher@nist.gov]	
Re:	Revision of three contributions: IEEE P802.15-00/045r0 (BB), IEEE P802.15-00/046r0, (LMP) and IEEE P802.15-00/001r2 (L2CAP)		
Abstract	This document revises the above three contributions and provides an integrated BB, LMP, and L2CAP SDL model as Letter Ballot #3 comments.		
Purpose	To replace the existing SDLs in Letter Ballot #3 as part of the comments		
Notice	This document has been prepared to assist the IEEE P802.15. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.		
Release	This is a contribution by the NIST and is not subject to copyright within the U.S. The contributor acknowledges and accepts that this contribution becomes the property of IEEE and may be made publicly available by P802.15.		

This contribution is part of the IEEE 802.15.1 Letter Ballot #3 comments and is submitted in this form due to the nature of its contents. The content of this contribution is an SDL model containing the BB, LMP, and L2CAP. This SDL is not complete, but is more complete than what is currently part of the letter ballot #3 draft.


This SDL completes Telelogic's Tau version 3.6 Analyzer without any errors, warnings, or infos. The Behaviour observed under the Simulator Tool appears to behave correctly.


The first part is the Signals defined and used in the model.

The second part includes the main block structure and the LMP and Link Manager. Dummy Blocks are used for SCO and HCI.

The third part is the BaseBand model.

The fourth part is the L2CAP model.

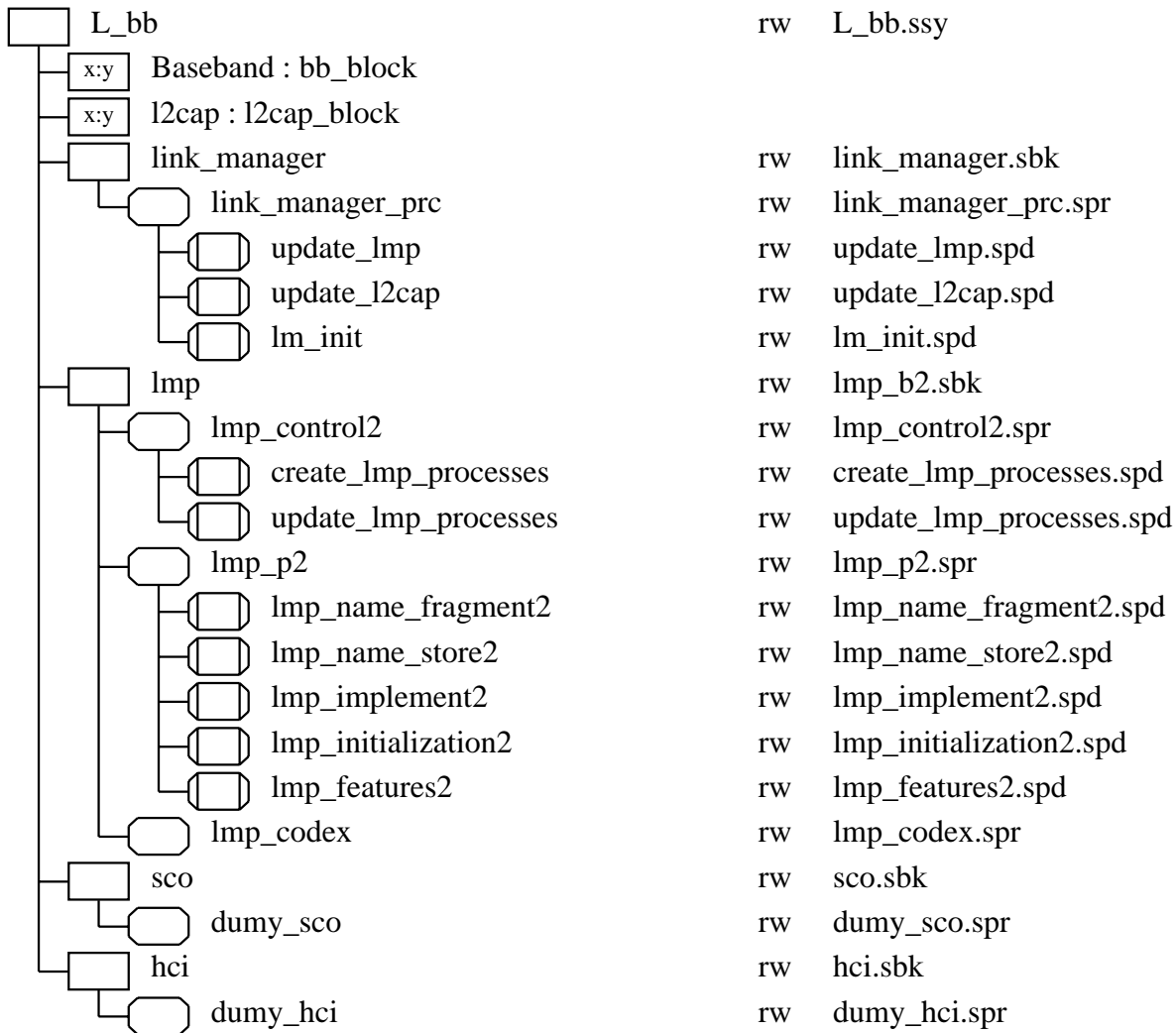
 rw /import/fs1/home/disk1/cypher/BLUETOOTH/new_bb/untitled.sdt

 rw /import/fs1/home/disk1/cypher/BLUETOOTH/new_bb/

— Analysis Model

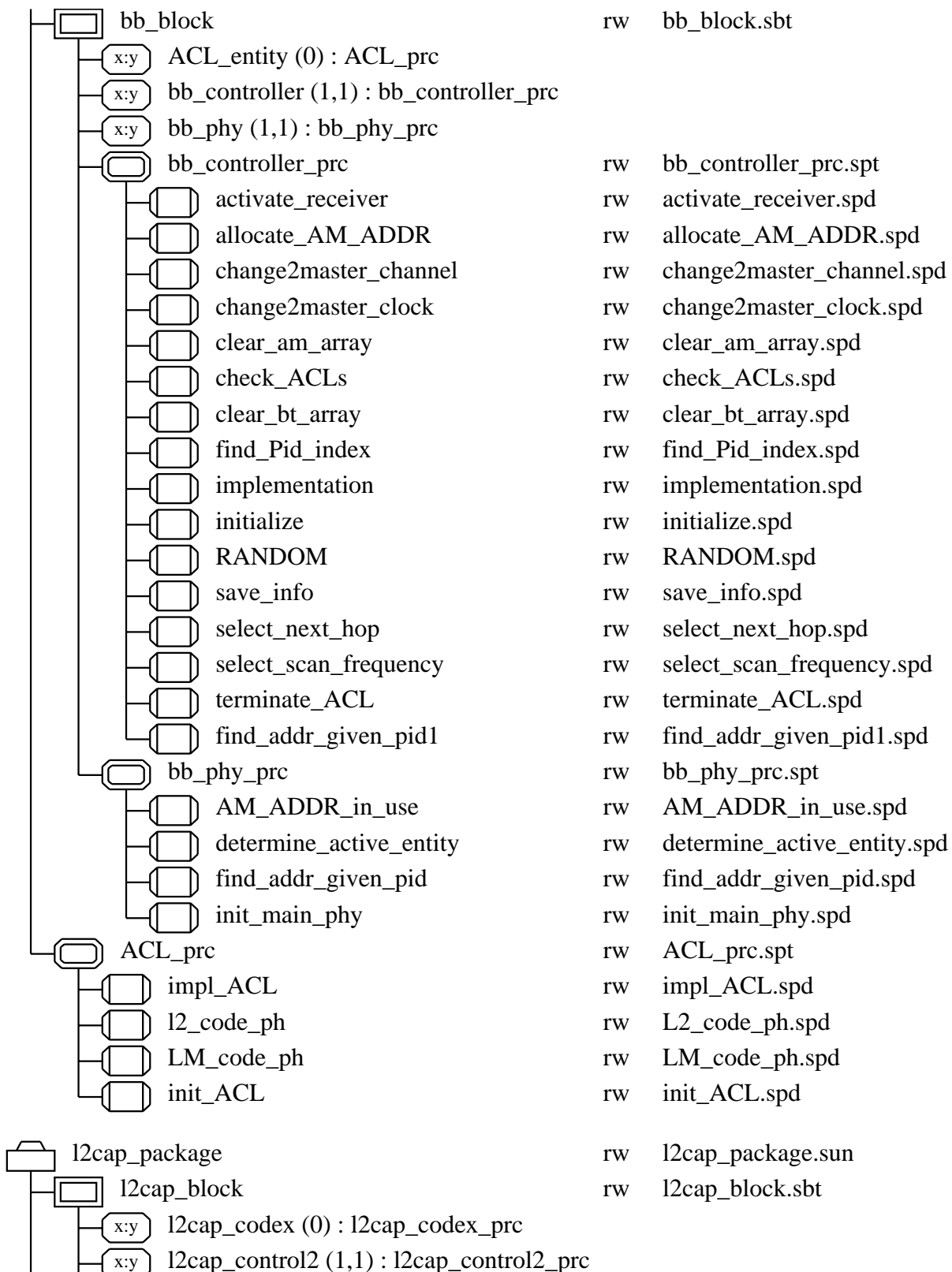
— Used Files

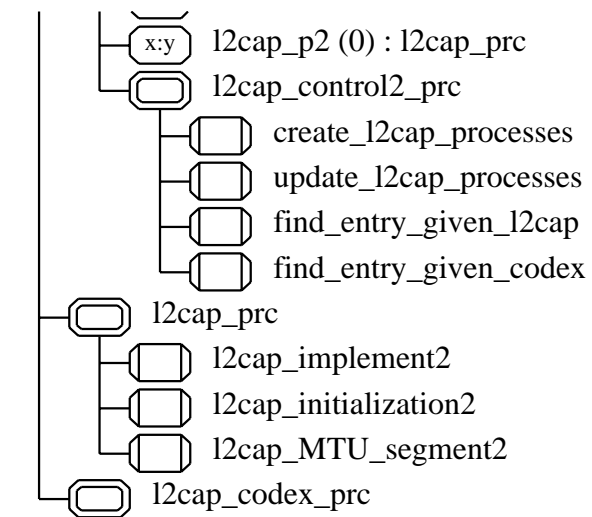
— new_bb



 sig_type_def 4 rw sig_type_def.sun

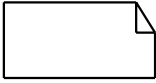
 bb_package rw bb_package.sun





rw l2cap_control2_prc.spt
rw create_l2cap_processes.spd
rw update_l2cap_processes.spd
rw find_entry_given_l2cap.spd
rw find_entry_given_codex.spd
rw l2cap_prc.spt
rw l2cap_implement2.spd
rw l2cap_initialization2.spd
rw l2cap_MTU_segment2.spd
rw l2cap_codex_prc.spt

— Other Documents



package sig_type_def

1(16)



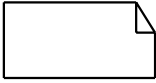
SIGNAL
voice_sample(integer,integer),HV1,HV2,HV3,DV,
com,
LMP_error,
listen4page,
BB_Create_Connection(Address_type),
BB_Inquiry,
BB_Inquiry_Cancel,
BB_Reset,
BB_Detach,
BB_EvntPkt,
l4p,t2p,gi,stop_inq,xReset;

SIGNAL
hold_mode_on,
BB_parkModeOn(integer,integer),
BB_parkModeOff(integer),
sniff_mode_on,
sniff_mode_off;

SIGNAL
real_slot(pckt_new);

SIGNAL
DAC, GIAC, DIAC, NULL, POLL,
FHS(integer), DM1(data_format), DH1,
AUX1, DM3, DH3, DM5, DH5,
bb_addtab(Address_type,integer,Pid),
bb_deltab(Address_type,integer,Pid),
bb_terminate, bb_send_id(pid), bb_send_id2(Pid),
received_something,
must_tx, ACL_timeout, amimaster(booleen);

SIGNALLIST sig_lmp = lmp_Create_ID, lmp_Return_ID;
SIGNALLIST sig_2lmp = bb_Create_ID, bb_Return_ID;
SIGNALLIST sig_sco = voice_sample,HV1,HV2,HV3,DV;
SIGNALLIST sig_2sco = voice_sample,HV1,HV2,HV3,DV;
SIGNALLIST sig_l2cap = l2cap_Return_ID;
SIGNALLIST sig_2l2cap = bb_Return_ID;
SIGNALLIST sig_hci = com,listen4page,
BB_Create_Connection, BB_Inquiry,
BB_Inquiry_Cancel, BB_Reset;
SIGNALLIST sig_2hci = com, BB_EvntPkt;
SIGNALLIST user_hci=l4p, t2p, gi, stop_inq, xReset;



package sig_type_def

2(16)



```
newtype pkt_new choice
  ID_pkt ac_types;
  non_ID_pkt slot;
endnewtype pkt_new;

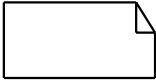
newtype slot
  struct
    access_code ac_types;
    header pk_head;
    payload contents;
  endnewtype slot;

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;
```

package sig_type_def

3(16)



```
newtype fhs_payload
struct
    LAP Integer;
    SR sr_values;
    SP sp_values;
    UAP Integer;
    NAP Integer;
    AM_ADDR Integer;
    CLK27_2 Integer;
    pg_scn_md Integer;
endnewtype fhs_payload;

newtype sr_values
    literals R0, R1, R2, reserved;
endnewtype sr_values;

newtype sp_values
    literals P0, P1, P2, reserved;
endnewtype sp_values;

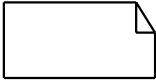
newtype data_format
struct
    L_CH lch_type;
    FLOW Integer;
    Length Integer;
    real_payload Imp_or_I2cap;
endnewtype data_format;

newtype lch_type
    literals NA, cont, storall, LM;
endnewtype lch_type;

newtype Imp_or_I2cap choice
    pyld_Imp LMP_pdu;
    pyld_I2cap I2cap_pdu;
endnewtype Imp_or_I2cap;

newtype voice_contents
    literals 10bytes, 20bytes, 30bytes;
endnewtype voice_contents;

newtype dv_frame_format
struct
    vf voice_contents;
    df data_format;
endnewtype dv_frame_format;
```

package sig_type_def

4(16)

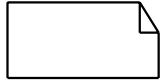


```
newtype role
  literals slave, master
endnewtype role;
```

```
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;
  endnewtype bt_entry;

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

package sig_type_def

5(16)



Information added by LMP

```
newtype Lmp_R_tab
  Array(Address_type,Process_type)
endnewtype;
syntype Address_type=Integer
endsyntype;
newtype Process_type
  struct
    lmp Pid;
    lmp_codex Pid;
    BB Pid;
    handle Handle_type;
  endnewtype;
newtype Handle_type
  literals non_exist, ACL, new_SCO, exist_SCO;
endnewtype;

newtype Packet_type
  literals DM1, DH1, DM3, DH3, DM5, DH5;
endnewtype;
newtype Packet_type1
  literals HV1, HV2, HV3;
endnewtype;

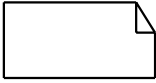
newtype Lmp_reason_type
  literals authentication_failure,_key_missing,max_SCO_connections,
  limited_resources,security_reasons,personal_device,
  host_timeout,user_ended_connection,low_resources,
  about_to_power_off,connection_terminate,repeated_attempts,
  pairing_not_allowed,unknown_PDU,unsupported_feature,
  SCO_offset,SCO_interval,SCO_air_mode,invalid_parameters,
  unspecified_error,unsupported_value,switch_not_allowed,
  transaction_collision,PDU_not_allowed,not_agree_encryption,
  unit_key_used;
endnewtype;

newtype Service_type
  literals no_traffic, best_effort, guaranteed;
endnewtype;
newtype Encryption_mode
  literals disabled, point_to_point, both;
endnewtype;
newtype Mode_type
  literals _active, sniff, hold, park;
endnewtype;
```

```
newtype Decision_type
  literals yes, no;
endnewtype;
newtype Role_type
  literals master, slave;
endnewtype;
newtype Key_flag_type
  literals semi_permanent, temporary;
endnewtype;
newtype Link_type
  literals ACL, SCO;
endnewtype;
newtype Encryption_enable
  literals ON, OFF;
endnewtype;
newtype Key_type
  literals unit_key, comb_key;
endnewtype;
newtype Paging_mode
  literals mandatory, _optional;
endnewtype;

newtype Power_type
  literals min, normal, max;
endnewtype;
newtype Name_type
  Array(Integer,Address_type)
endnewtype;
syntype Key_content=Integer
endsyntype;
syntype Response_content=Integer
endsyntype;

newtype Lmp_status_type
  literals success, failed;
endnewtype;
newtype Lmp_status_type1
  literals no, success, pending, reject;
endnewtype;
newtype Lmp_status_type2
  literals no, request, agree;
endnewtype;
```

package sig_type_def

6(16)



SIGNAL


```
Create_Connection(Address_type,Packet_type,Paging_mode,Decision_type),
Disconnect(Handle_type,Imp_reason_type),
Add_SCO_Connection(Handle_type,Packet_type1),
Accept_Connection_Request(Address_type,Role_type),
Reject_Connection_Request(Address_type,Imp_reason_type),
Authentication_Requested(Handle_type),
Set_Connection_Encryption(Handle_type,Encryption_enable),
Change_Connection_Link_Key(Handle_type),
Master_Link_Key(Key_flag_type),
Remote_Name_Request(Address_type,Paging_mode),
Read_Remote_Supported_Features(Handle_type),
Read_Remote_Version_Information(Handle_type),
Read_Clock_Offset(Handle_type),
Hold_Mode(Handle_type),
Sniff_Mode(Handle_type,Duration,Duration),
Exit_Sniff_Mode(Handle_type),
Park_Mode(Handle_type),
Exit_Park_Mode(Handle_type),
QoS_Setup(Handle_type,Service_type),
Switch_Role(Address_type,Role_type),
Modify_SCO_Connection(Handle_type,Packet_type1),
Modify_Park(Handle_type),
Read_Timing_Accuracy(Handle_type),
Power_Control(Handle_type),
Change_DM_DH(Handle_type),
Change_Data_Rate(Handle_type),
Multislot_Packets_Control(Handle_type),
Paging_Scheme(Address_type),
Link_Supervision(Handle_type),

Connection_Complete(Imp_status_type,Handle_type,Address_type,Link_type,Encryption_mode),
Connection_Request(Address_type,Link_type),
Disconnection_Complete(Imp_status_type,Handle_type,Imp_reason_type),
Authentication_Complete(Imp_status_type,Handle_type),
Remote_Name_Request_Complete(Imp_status_type,Address_type,Name_type),
Encryption_Change(Imp_status_type,Handle_type,Encryption_enable),
Change_Connection_Link_Key_Complete(Imp_status_type,Handle_type),
Master_Link_Key_Complete(Imp_status_type,Handle_type,Key_flag_type),
Read_Remote_Supported_Features_Complete(Imp_status_type,Handle_type,Integer),
Read_Remote_Version_Information_Complete(Imp_status_type,Handle_type,Integer),
QoS_Setup_Complete(Imp_status_type,Handle_type,Service_type),
Role_Change(Imp_status_type,Address_type,Role_type),
Mode_Change(Imp_status_type,Handle_type,Mode_type),
Link_Key_Notification(Address_type,Key_content),
Read_Clock_Offset_Complete(Imp_status_type,Handle_type),
Read_Timing_Accuracy_Complete(Imp_status_type,Handle_type),
Multislot_Packets_Control_Complete(Imp_status_type,Handle_type),
Paging_Scheme_Complete(Imp_status_type,Address_type);
```



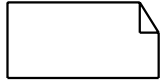
package sig_type_def

7(16)



```
SIGNAL
LMP_name_req(Integer),
LMP_name_res(Integer,Integer,Name_type),
LMP_accepted(Integer),
LMP_not_accepted(Integer,Imp_reason_type),
LMP_clkoffset_req,
LMP_clkoffset_res(Duration),
LMP_detach(Imp_reason_type),
LMP_in_rand(Role_type,Key_content),
LMP_comb_key(Key_content),
LMP_unit_key(Key_content),
LMP_aud_rand(Key_content),
LMP_sres(Response_content),
LMP_temp_rand(Key_content),
LMP_temp_key(Key_content),
LMP_encryption_mode_req(Encryption_mode),
LMP_encryption_key_size_req(Integer),
LMP_start_encryption_req(Key_content),
LMP_stop_encryption_req,
LMP_switch_req,
LMP_hold(Duration),
LMP_hold_req(Role_type,Duration),
LMP_sniff(Duration,Duration,Duration,Duration),
LMP_sniff_req(Role_type,Duration,Duration,Duration,Duration),
LMP_unsniff_req,
LMP_park_req,
LMP_park(Integer,Duration,Integer,Address_type,Address_type),
LMP_set_broadcast_scan_window(Duration),
LMP_modify_beacon(Duration,Integer),
LMP_unpark_BD_ADDR_req(Address_type,Address_type),
LMP_unpark_PM_ADDR_req(Address_type,Address_type),
LMP_incr_power_req,
LMP_decr_power_req,
LMP_max_power,
LMP_min_power,
LMP_auto_rate,
LMP_preferred_rate(Integer),
LMP_version_req(Integer,Integer,Integer),
LMP_version_res(Integer,Integer,Integer),
LMP_features_req(Integer),
LMP_features_res(Integer),
LMP_quality_of_service(Duration,Integer),
LMP_quality_of_service_req(Duration,Integer),
LMP_SCO_link_req(Role_type,Handle_type,Duration,Duration),
LMP_remove_SCO_link_req(Handle_type),
LMP_max_slot(Integer),
LMP_max_slot_req(Integer),
LMP_timing_accuracy_req,
LMP_timing_accuracy_res(Duration,Duration),
LMP_setup_complete(Role_type),
LMP_use_semi_permanent_key,
LMP_host_connection_req,
LMP_slot_offset(Duration,Address_type),
LMP_page_mode_req(Paging_mode),
LMP_page_scan_mode_req(Paging_mode),
LMP_supervision_timeout(Duration);
```





package sig_type_def

8(16)

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,

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,
Switch_Role,
Modify_SCO_Connection,
Modify_Park,
Read_Timing_Accuracy,
Power_Control,
Change_DM_DH,
Change_Data_Rate,
Multislot_Packets_Control,
Paging_Scheme,
Link_Supervision;

signallist

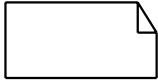
Up_LMP_Cmds1=
Create_Connection,
Remote_Name_Request;

signallist

LMP_Up_Cmds=
Connection_Complete,
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_Change,
Mode_Change,
Link_Key_Notification,
Read_Clock_Offset_Complete,
Read_Timing_Accuracy_Complete,
Multislot_Packets_Control_complete,

LMP_supervision_timeout;

Paging_Scheme_Complete;



package sig_type_def

9(16)

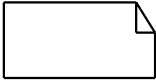


newtype LMP_pdu
choice
pdu1 PDU_type1;
pdu2 PDU_type2;
pdu3 PDU_type3;
pdu4 PDU_type4;
pdu5 Integer;
pdu6 PDU_type6;
pdu7 PDU_type7;
pdu8 PDU_type8;
pdu9 PDU_type9;
pdu10 PDU_type10;
pdu11 PDU_type11;
pdu12 PDU_type12;
pdu13 PDU_type13;
pdu14 PDU_type14;
pdu15 PDU_type15;
pdu16 PDU_type16;
pdu17 PDU_type17;
pdu18 Integer;
pdu19 Integer;
pdu20 PDU_type20;
pdu21 PDU_type21;
pdu22 PDU_type22;
pdu23 PDU_type23;
pdu24 Integer;
pdu25 Integer;
pdu26 PDU_type26;
pdu27 PDU_type27;
pdu28 PDU_type28;
pdu29 PDU_type29;
pdu30 PDU_type30;
pdu31 Integer;
pdu32 Integer;
pdu33 Integer;
pdu34 Integer;
pdu35 Integer;
pdu36 PDU_type36;
pdu37 PDU_type37;
pdu38 PDU_type38;
pdu39 PDU_type39;
pdu40 PDU_type40;
pdu41 PDU_type41;
pdu42 PDU_type42;
pdu43 PDU_type43;
pdu44 PDU_type44;
pdu45 PDU_type45;
pdu46 PDU_type46;
pdu47 Integer;
pdu48 PDU_type48;
pdu49 PDU_type49;
pdu50 Integer;
pdu51 Integer;
pdu52 PDU_type52;
pdu53 PDU_type53;
pdu54 PDU_type54;

SIGNAL
Imp_Create_ID(Address_type,PId),
Imp_Return_ID(PId),
Imp_Send_ID(PId),
Imp_terminate,
Imp_Send_ID1(PId,PId),
Imp_terminate1,
Lmp_Sdu(LMP_pdu);

DEC Added page

```
pdu55 PDU_type55;  
endnewtype;
```



package sig_type_def

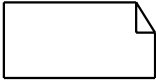
10(16)



```
newtype PDU_type1 struct
  code Integer;
  offset Integer; endnewtype;
newtype PDU_type2 struct
  code Integer;
  offset Integer;
  length Integer;
  fragment Name_type; endnewtype;
newtype PDU_type3 struct
  code Integer;
  op_code Integer; endnewtype;
newtype PDU_type4 struct
  code Integer;
  op_code Integer;
  reason Imp_reason_type; endnewtype;
newtype PDU_type6 struct
  code Integer;
  offset Duration; endnewtype;
newtype PDU_type7 struct
  code Integer;
  reason Imp_reason_type; endnewtype;
newtype PDU_type8 struct
  code Integer;
  trans_ID Role_type;
  rand Key_content; endnewtype;
newtype PDU_type9 struct
  code Integer;
  rand Key_content; endnewtype;
newtype PDU_type10 struct
  code Integer;
  key Key_content; endnewtype;
newtype PDU_type11 struct
  code Integer;
  rand Key_content; endnewtype;
newtype PDU_type12 struct
  code Integer;
  res Response_content; endnewtype;
newtype PDU_type13 struct
  code Integer;
  rand Key_content; endnewtype;
newtype PDU_type14 struct
  code Integer;
  key Key_content; endnewtype;
newtype PDU_type15 struct
  code Integer;
  mode Encryption_mode;
endnewtype;
newtype PDU_type16 struct
  code Integer;
  key_size Integer; endnewtype;
newtype PDU_type17 struct
  code Integer;
```

```
newtype PDU_type20 struct
  code Integer;
  hold_time Duration; endnewtype;
newtype PDU_type21 struct
  code Integer;
  trans_ID Role_type;
  hold_time Duration; endnewtype;
newtype PDU_type22 struct
  code Integer;
  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_type26 struct
  code Integer;
  DB Integer;
  TB Duration;
  NB Integer;
  PM_ADDR Address_type;
  AR_ADDR Address_type;
endnewtype;
newtype PDU_type27 struct
  code Integer;
  scan_window Duration;
endnewtype;
newtype PDU_type28 struct
  code Integer;
  TB Duration;
  NB Integer; endnewtype;
newtype PDU_type29 struct
  code Integer;
  AM_ADDR Address_type;
  BD_ADDR Address_type;
endnewtype;
```

```
rand Key_content; endnewtype;
```



package sig_type_def

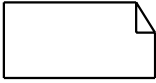
11(16)



```
newtype PDU_type30 struct
  code Integer;
  AM_ADDR Address_type;
  PM_ADDR Address_type;
endnewtype;
newtype PDU_type36 struct
  code Integer;
  data_rate Integer; endnewtype;
newtype PDU_type37 struct
  code Integer;
  VersNr Integer;
  Compld Integer;
  SubVersNr Integer; endnewtype;
newtype PDU_type38 struct
  code Integer;
  VersNr Integer;
  Compld Integer;
  SubVersNr Integer; endnewtype;
newtype PDU_type39 struct
  code Integer;
  features Integer; endnewtype;
```

```
newtype PDU_type40 struct
  code Integer;
  features Integer; endnewtype;
newtype PDU_type41 struct
  code Integer;
  interval Duration;
  NBC Integer; endnewtype;
newtype PDU_type42 struct
  code Integer;
  interval Duration;
  NBC Integer; endnewtype;
newtype PDU_type43 struct
  code Integer;
  trans_ID Role_type;
  handle Handle_type;
  Dsco Duration;
  Tscs Duration; endnewtype;
newtype PDU_type44 struct
  code Integer;
  handle Handle_type; endnewtype;
newtype PDU_type45 struct
  code Integer;
  max_slots Integer; endnewtype;
newtype PDU_type46 struct
  code Integer;
  max_slots Integer; endnewtype;
newtype PDU_type48 struct
  code Integer;
  drift Duration;
  jitter Duration; endnewtype;
newtype PDU_type49 struct
  code Integer;
  trans_ID Role_type; endnewtype;
newtype PDU_type52 struct
  code Integer;
  offset Duration;
  BD_ADDR Address_type;
endnewtype;
newtype PDU_type53 struct
  code Integer;
  paging_scheme Paging_mode;
endnewtype;
newtype PDU_type54 struct
  code Integer;
  paging_scheme Paging_mode;
endnewtype;
newtype PDU_type55 struct
  code Integer;
  timeout Duration; endnewtype;
```





package sig_type_def

12(16)

Information added for L2CAP

SIGNAL
LP_ConnectCfm(Address_type),
LP_ConnectInd(Address_type,PId),
LP_ConnectReq(Address_type),
LP_ConnectRsp(Address_type),
LP_DisconnectInd(Address_type),
LP_QoS_Cfm(Flow_type),
LP_QoSReq(Flow_type),
LP_QoSViolationInd(Flow_type),

L2CAP_Unknown_Command,
L2CAP_Data(Integer),
L2CA_ConnectReq(Address_type,PSM_type),
L2CA_ConnectRsp(Address_type,Integer,Integer,Response_type,l2cap_status_type),
L2CA_ConfigReq(Integer,Integer,Duration,Duration,Flow_type),
L2CA_ConfigRsp(Integer,Integer,Flow_type,Response_type1),
L2CA_DisconnectReq(Integer),
L2CA_DataRead(Integer,Integer,Integer),
L2CA_ConnectCfm(Integer,Result_type1,l2cap_status_type),
L2CA_ConfigCfm(Result_type3,Integer,Duration,Duration,Flow_type),
L2CA_DisconnectCfm,
L2CA_TimeOutInd,
L2CA_ConnectInd(Address_type,Integer,PSM_type,Integer),
L2CA_ConfigInd(Integer,Integer,Duration,Flow_type),
L2CA_DisconnectInd(Integer),
L2CA_QoSViolationInd(Address_type),

l2cap_Return_ID(PId),
BB_Create_ID(Address_type,PId), BB_Return_ID(PId),
l2cap_Send_ID(PId,PId), l2cap_Terminate,
l2cap_Send_ID1(PId,PId), l2cap_terminate1,
l2cap_Sdu(L2CAP_pdu);

signal LP_ConnectCfmNeg inherits LP_ConnectCfm;
signal LP_QoS_CfmNeg inherits LP_QoS_Cfm;
signal LP_ConnectRspNeg inherits LP_ConnectRsp;
signal L2CA_DisconnectRsp inherits L2CA_DisconnectReq;
signal L2CA_DataWrite inherits L2CA_DataRead;

package sig_type_def

13(16)

```
signallist
L2CAP_LP_Cmds=
LP_ConnectReq,
LP_QoSReq,
LP_ConnectRsp,
LP_ConnectRspNeg;
```

```
signallist
Up_L2CAP_Cmds=
L2CA_ConnectReq,
L2CA_ConnectRsp,
L2CA_ConfigReq,
L2CA_ConfigRsp,
L2CA_DisconnectReq,
L2CA_DisconnectRsp,
L2CA_DataRead,
L2CA_DataWrite;
```

```
signallist
L2CAP_Cmds=
L2CAP_CmdReject,
L2CAP_ConnectReq,
L2CAP_ConnectRsp,
L2CAP_ConfigReq,
L2CAP_ConfigRsp,
L2CAP_DisconnectReq,
L2CAP_DisconnectRsp,
L2CAP_Unknown_Command,
L2CAP_Data;
```

```
signallist
LP_L2CAP_Cmds=
LP_ConnectCfm,
LP_ConnectCfmNeg,
LP_DisconnectInd,
LP_QoSReq,
LP_QoSReqNeg,
LP_QoSViolationInd;
```

```
signallist
L2CAP_Up_Cmds=
L2CA_ConnectInd,
L2CA_ConnectCfm,
L2CA_ConfigInd,
L2CA_ConfigCfm,
L2CA_DisconnectInd,
L2CA_DisconnectCfm,
L2CA_TimeOutInd,
L2CA_QoSViolationInd;
```

```
syntype SmallInt=Integer
constants 0:10
endsyntype;
newtype Result_type1
literals success, pending,
refused, timeout;
endnewtype;
newtype Result_type3
literals success, failure, timeout;
endnewtype;
newtype Response_type
literals success,pending,refused;
endnewtype;
newtype Response_type1
literals success,failed;
endnewtype;
```

```
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_codex PId;
LP PId;
CID Integer;
Iden Integer;
endnewtype;
```



```

SIGNAL
L2CAP_CmdReject(Integer,Integer,Reason_type,Data_type),
L2CAP_ConnectReq(Integer,Integer,PSM_type,Integer),
L2CAP_ConnectRsp(Integer,Integer,Integer,Integer,Result_type,l2cap_status_type),
L2CAP_ConfigReq(Integer,Integer,Integer,Flag_type,Integer,Duration,Flow_type),
L2CAP_ConfigRsp(Integer,Integer,Integer,Flag_type,Result_type2,Integer,Duration,Flow_type),
L2CAP_DisconnectReq(Integer,Integer,Integer,Integer),
L2CAP_DisconnectRsp(Integer,Integer,Integer,Integer);

```

```

newtype L2CAP_pdu
choice
l2pdu1 l2pdu_type1;
l2pdu2 l2pdu_type2;
l2pdu3 l2pdu_type3;
l2pdu4 l2pdu_type4;
l2pdu5 l2pdu_type5;
l2pdu6 l2pdu_type6;
l2pdu7 l2pdu_type7;
endnewtype;

```

```

newtype l2pdu_type1 struct
code Integer;
iden Integer;
len Integer;
reason Reason_type;
data Data_type;
endnewtype;

newtype Reason_type
literals not_understood,
MTU_exceed, invalid_CID;
endnewtype;

```

```

newtype Data_type
literals not_applicable,
actual_MTU, requested_CID;
endnewtype;

```

```

newtype l2pdu_type2 struct
code Integer;
iden Integer;
len Integer;
PSM PSM_type;
LCID Integer;
endnewtype;

```

```

newtype l2pdu_type3 struct
code Integer;
iden Integer;
len Integer;
LCID Integer;
RCID Integer;
result Result_type;
status l2cap_status_type;
endnewtype;

```

```

newtype PSM_type
literals SDP, RFCOMM, TCP;
endnewtype;

```

```

newtype Result_type
literals success, pending,
refused;
endnewtype;

```

```

newtype l2cap_status_type
literals no_further, pending;
endnewtype;

```

```

newtype l2pdu_type6 struct
code Integer;
iden Integer;
len Integer;
RCID Integer;
LCID Integer;
endnewtype;

```

```

newtype l2pdu_type7 struct
code Integer;
iden Integer;
len Integer;
LCID Integer;
RCID Integer;
endnewtype;

```

```

newtype l2pdu_type4 struct
code Integer;
iden Integer;
len Integer;
RCID Integer;
C_flag Flag_type;
MTU Integer;
FlushTO Duration;
Flow Flow_type;
endnewtype;

```

```

newtype l2pdu_type5 struct
code Integer;
iden Integer;
len Integer;
RCID Integer;
C_flag Flag_type;
result Result_type2;
MTU Integer;
FlushTO Duration;
Flow Flow_type;
endnewtype;

```

```

newtype Flag_type
literals 0,1;
endnewtype;

```

```

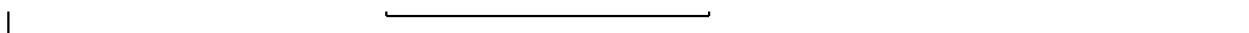
newtype Result_type2
literals success, failure;
endnewtype;

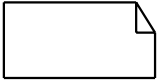
```

```

newtype Flow_type struct
service_type Service_type;
QoS_para Integer;
endnewtype;

```





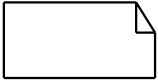
package sig_type_def

15(16)



Link Manager

Signal
db_update(Address_type, Pid, Pid, Pid);




package sig_type_def


16(16)



HCI

```
newtype allen_type  
  literals Giacaddr, Diacaddr;  
endnewtype allen_type;
```


 rw /import/fs1/home/disk1/cypher/BLUETOOTH/new_bb/untitled.sdt

 rw /import/fs1/home/disk1/cypher/BLUETOOTH/new_bb/


— Analysis Model 4


— Used Files 4

— new_bb 4

 L_bb 4 rw L_bb.ssy


 Baseband : bb_block


 l2cap : l2cap_block


 link_manager

 link_manager_prc


 update_lmp


 update_l2cap

 lm_init


 lmp

 lmp_control2


 create_lmp_processes


 update_lmp_processes

 lmp_p2


 lmp_name_fragment2

 lmp_name_store2


 lmp_implement2

 lmp_initialization2


 lmp_features2

 lmp_codex

 sco

 dumy_sco

 hci

 dumy_hci

6 rw link_manager.sbk

8 rw link_manager_prc.spr

24 rw update_lmp.spd

26 rw update_l2cap.spd

28 rw lm_init.spd

30 rw lmp_b2.sbk

32 rw lmp_control2.spr

40 rw create_lmp_processes.spd

42 rw update_lmp_processes.spd

44 rw lmp_p2.spr

130 rw lmp_name_fragment2.spd

131 rw lmp_name_store2.spd

132 rw lmp_implement2.spd

133 rw lmp_initialization2.spd

138 rw lmp_features2.spd

139 rw lmp_codex.spr

185 rw sco.sbk

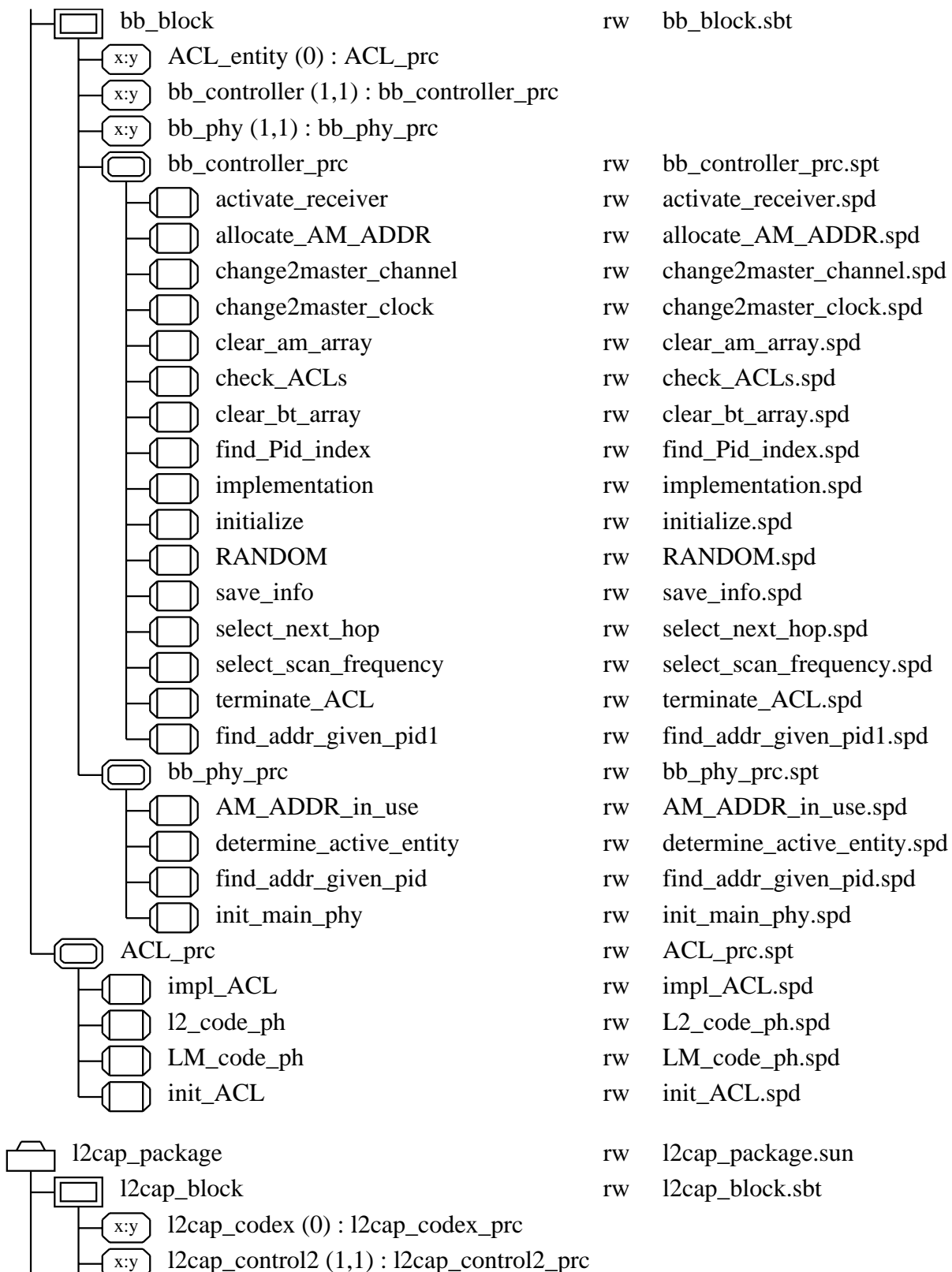
187 rw dumy_sco.spr

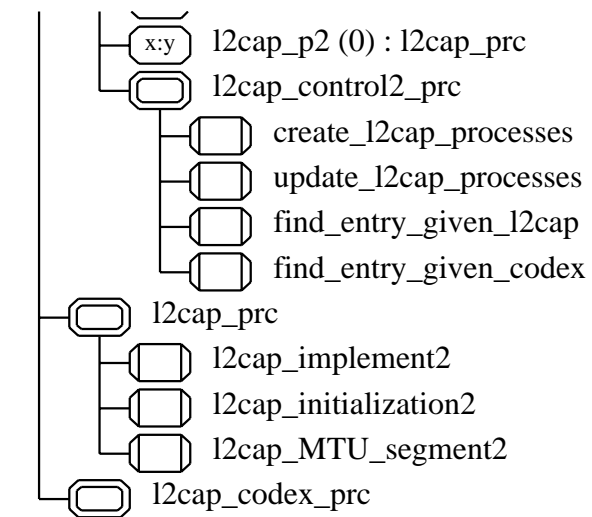
189 rw hci.sbk

191 rw dumy_hci.spr

 sig_type_def rw sig_type_def.sun

 bb_package rw bb_package.sun





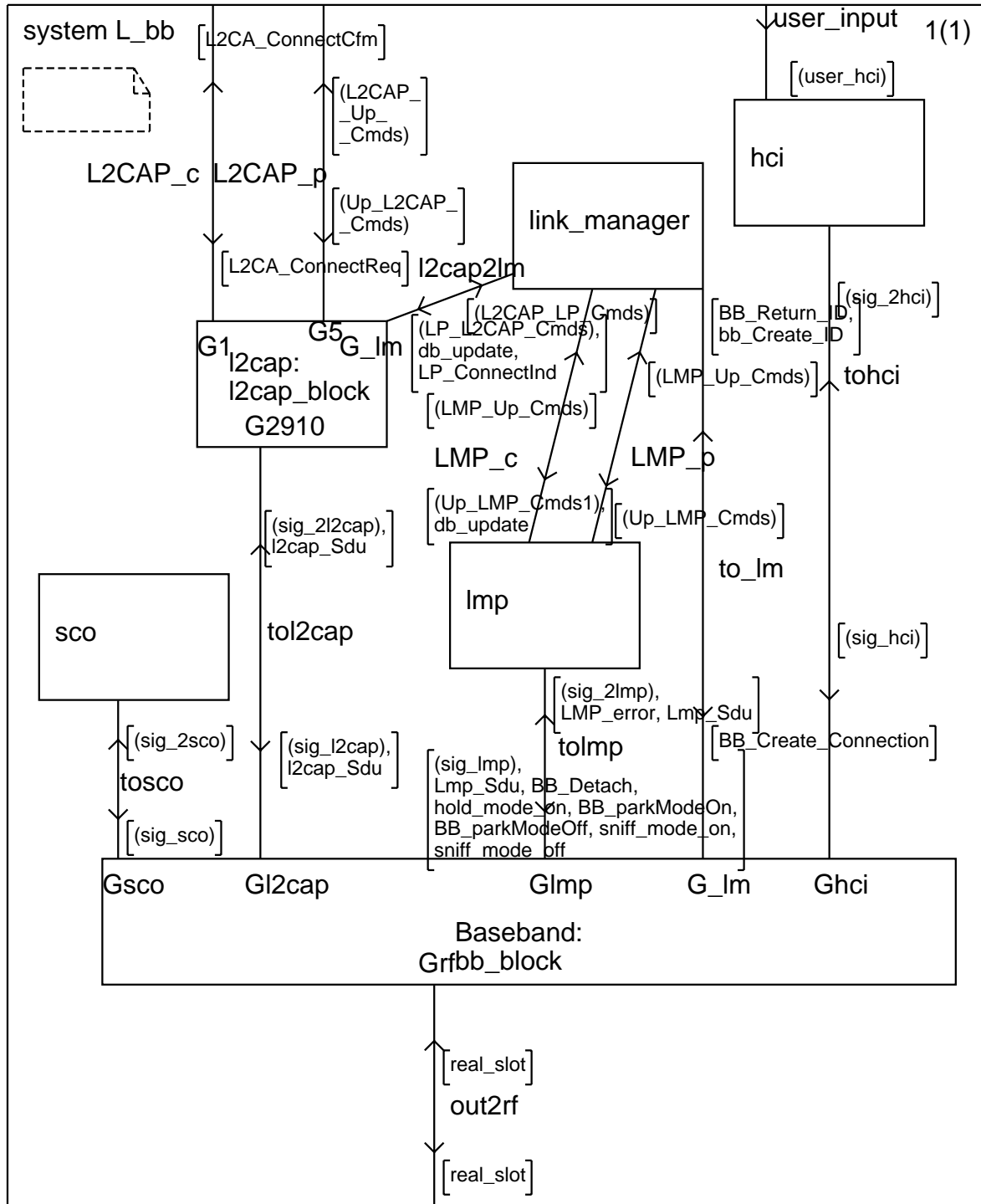
rw	l2cap_control2_prc.spt
rw	create_l2cap_processes.spd
rw	update_l2cap_processes.spd
rw	find_entry_given_l2cap.spd
rw	find_entry_given_codex.spd
rw	l2cap_prc.spt
rw	l2cap_implement2.spd
rw	l2cap_initialization2.spd
rw	l2cap_MTU_segment2.spd
rw	l2cap_codex_prc.spt

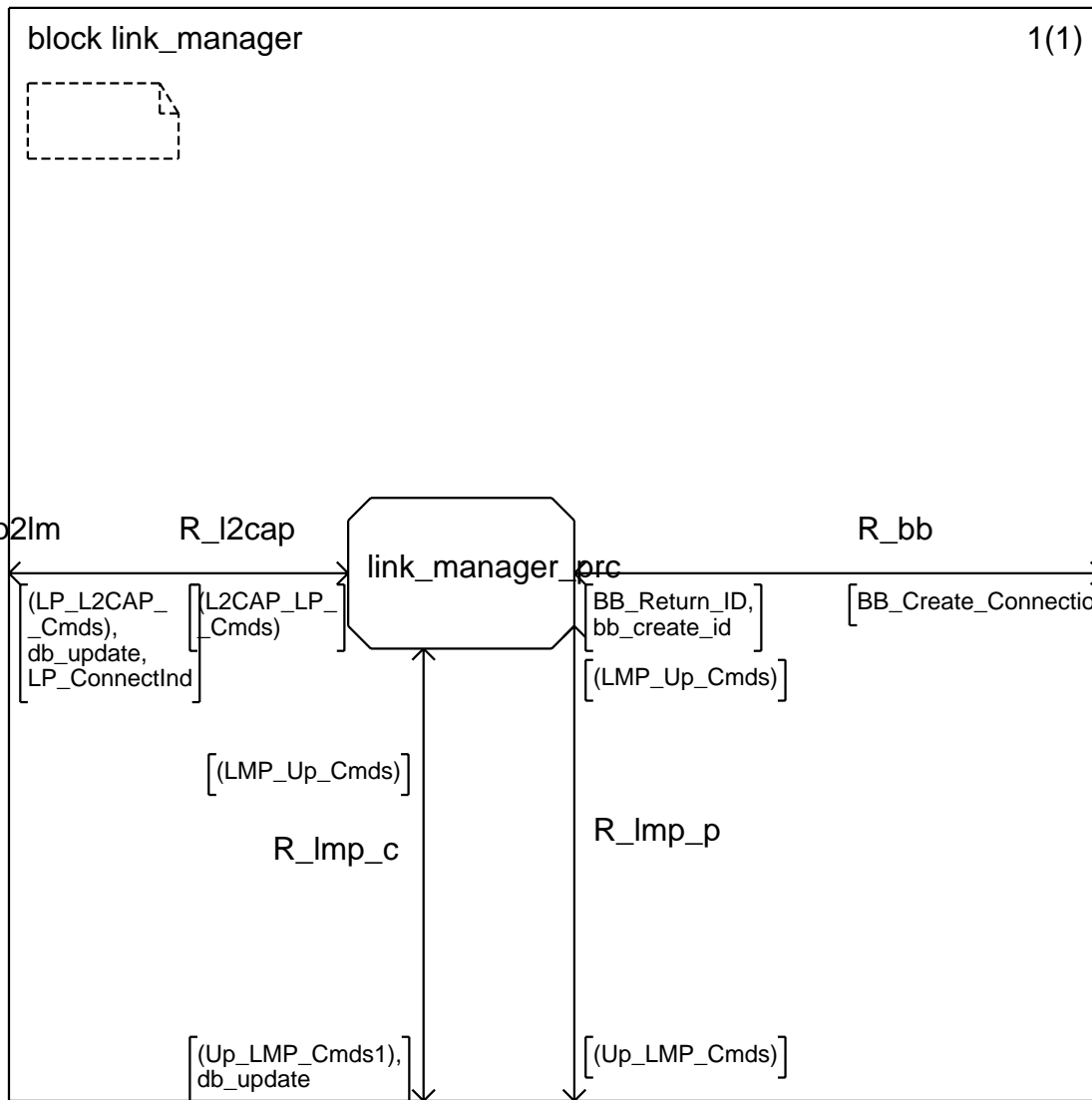
— Other Documents

```

USE sig_type_def;
USE l2cap_package;
USE bb_package;

```





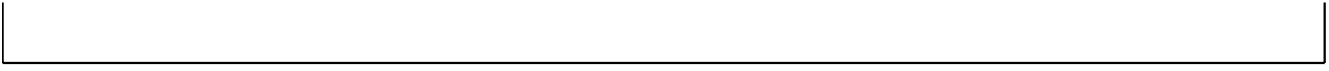
LMP_c LMP_p



```
DCL
Im_db Im_R_tab;
DCL
addr Address_type;
DCL
pt Packet_type;
DCL
pm Paging_mode;
DCL
dt Decision_type;
DCL
Impstst Imp_status_type;
DCL
hndlt Handle_type;
DCL
lnkt link_type;
DCL
encrym encryption_mode;

newtype Im_R_tab
  array(address_type, db_entry)
endnewtype Im_R_tab;

newtype db_entry
struct
  bb_pid PID;
  Imp_pid PID;
  Imp_codex_pid PID;
  l2cap_pid PID;
endnewtype db_entry;
```





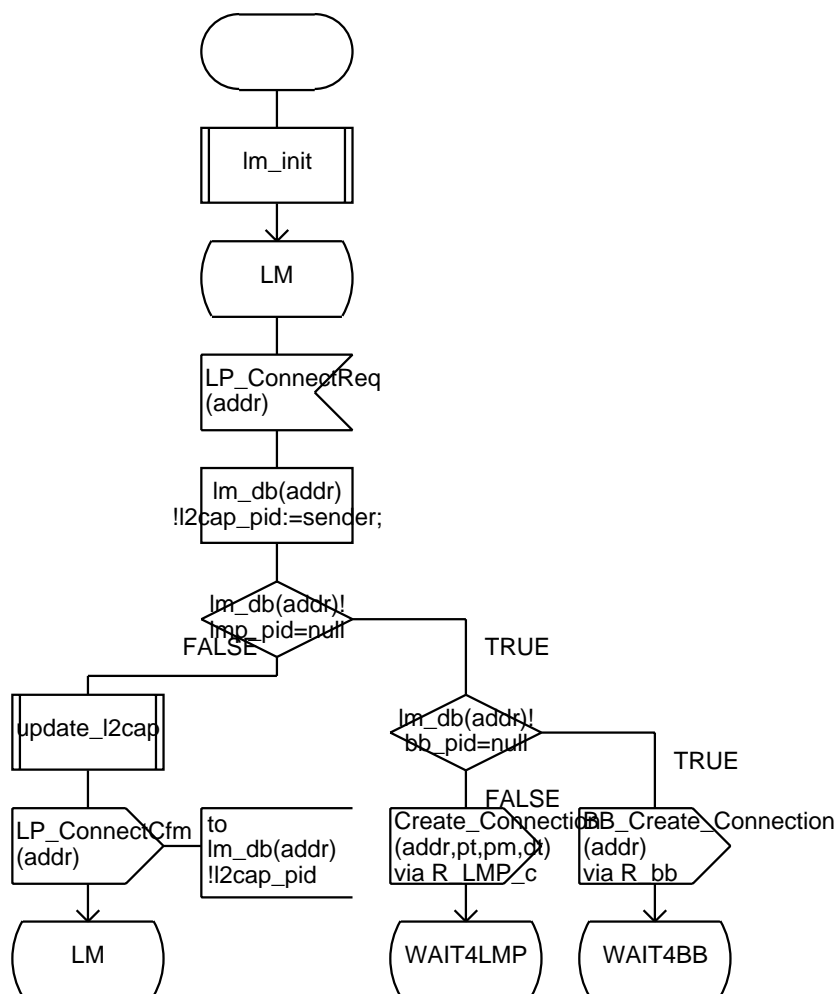
DCL
bb_id, Imp_id, Imp_codex_id PID;

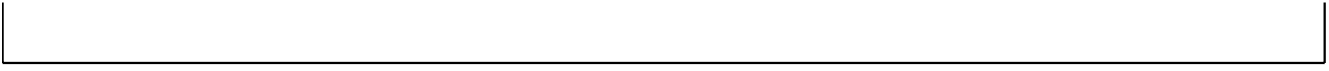
update_imp

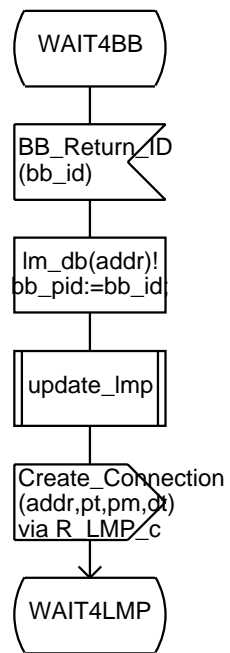
update_2cap

lm_init

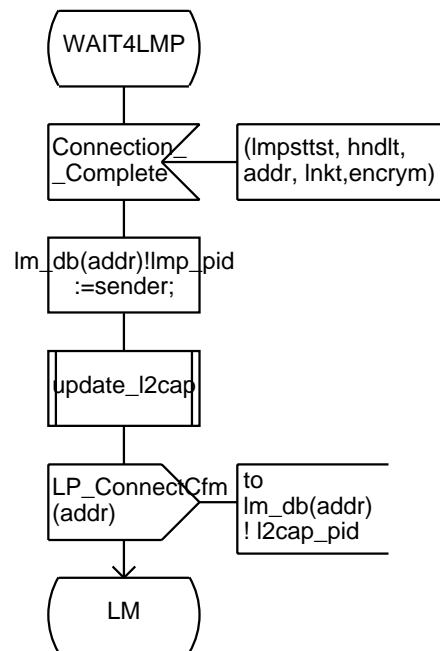


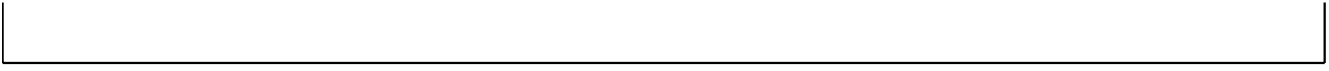


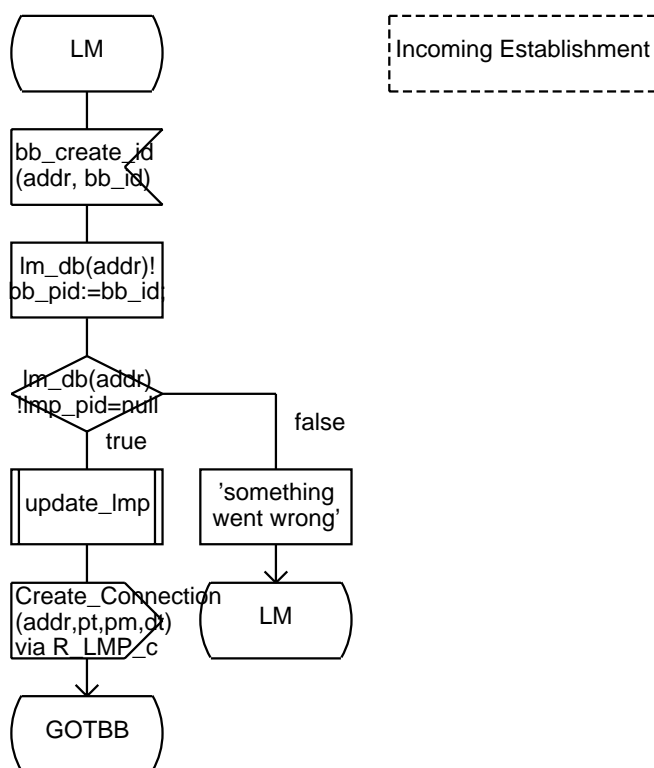


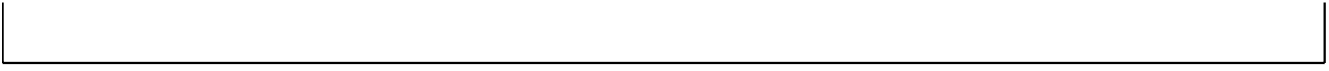


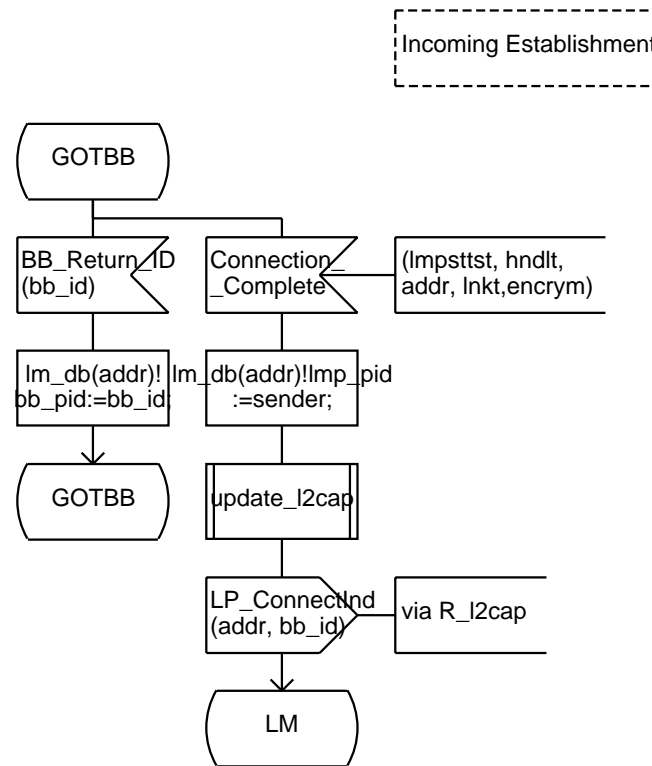


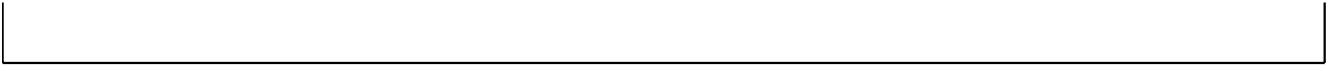


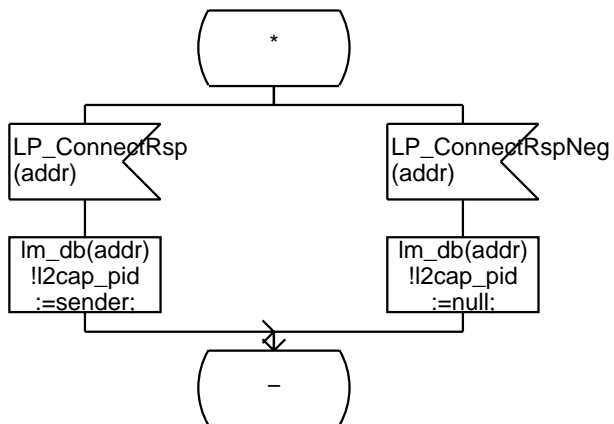








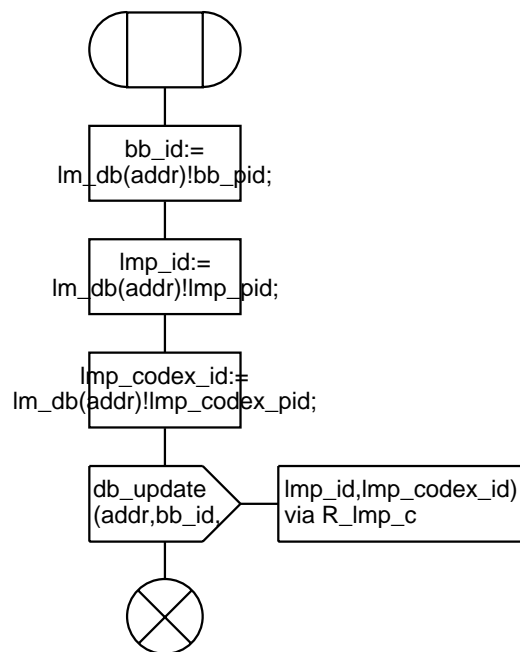






procedure update_Imp

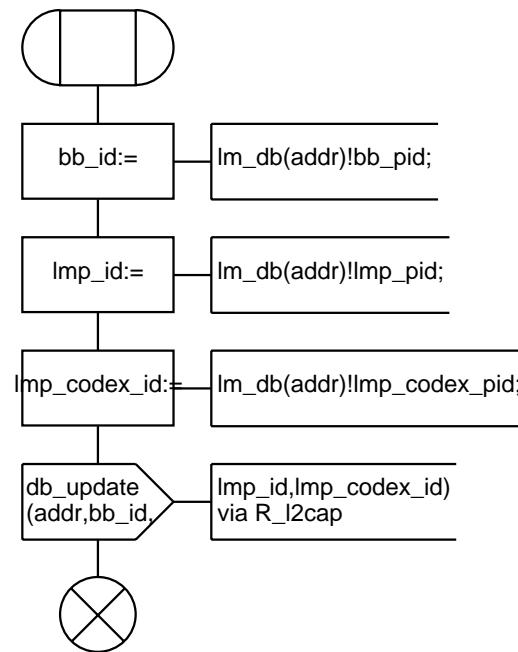
1(1)





procedure update_I2cap

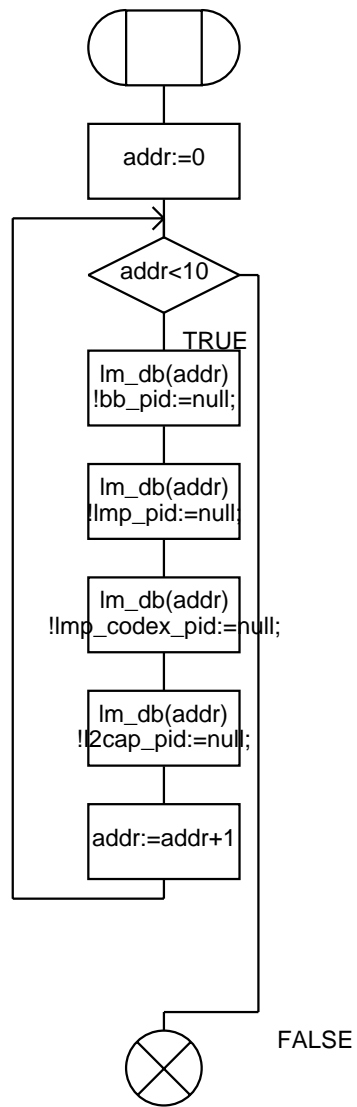
1(1)



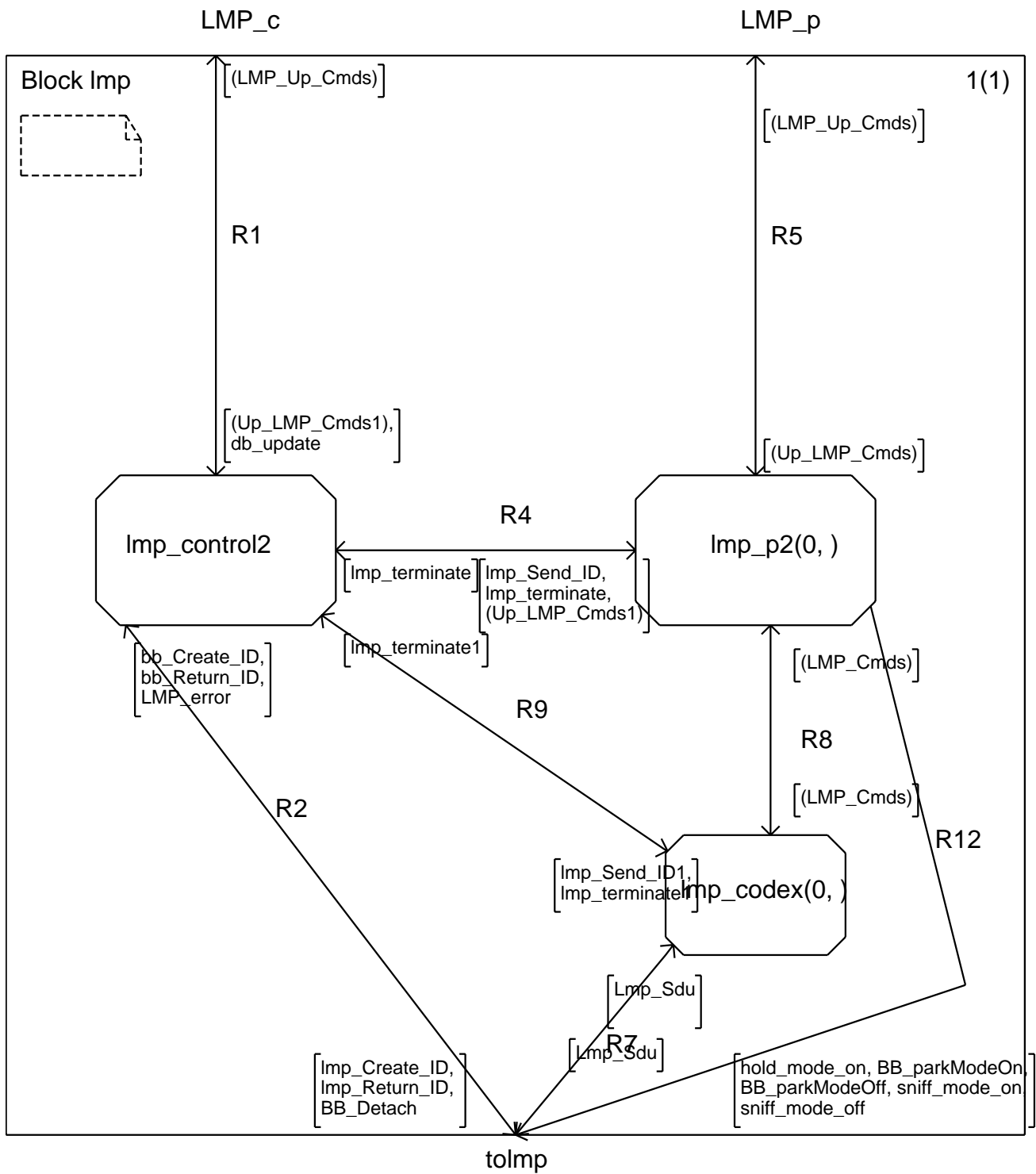


procedure Im_init

1(1)







process Imp_control2

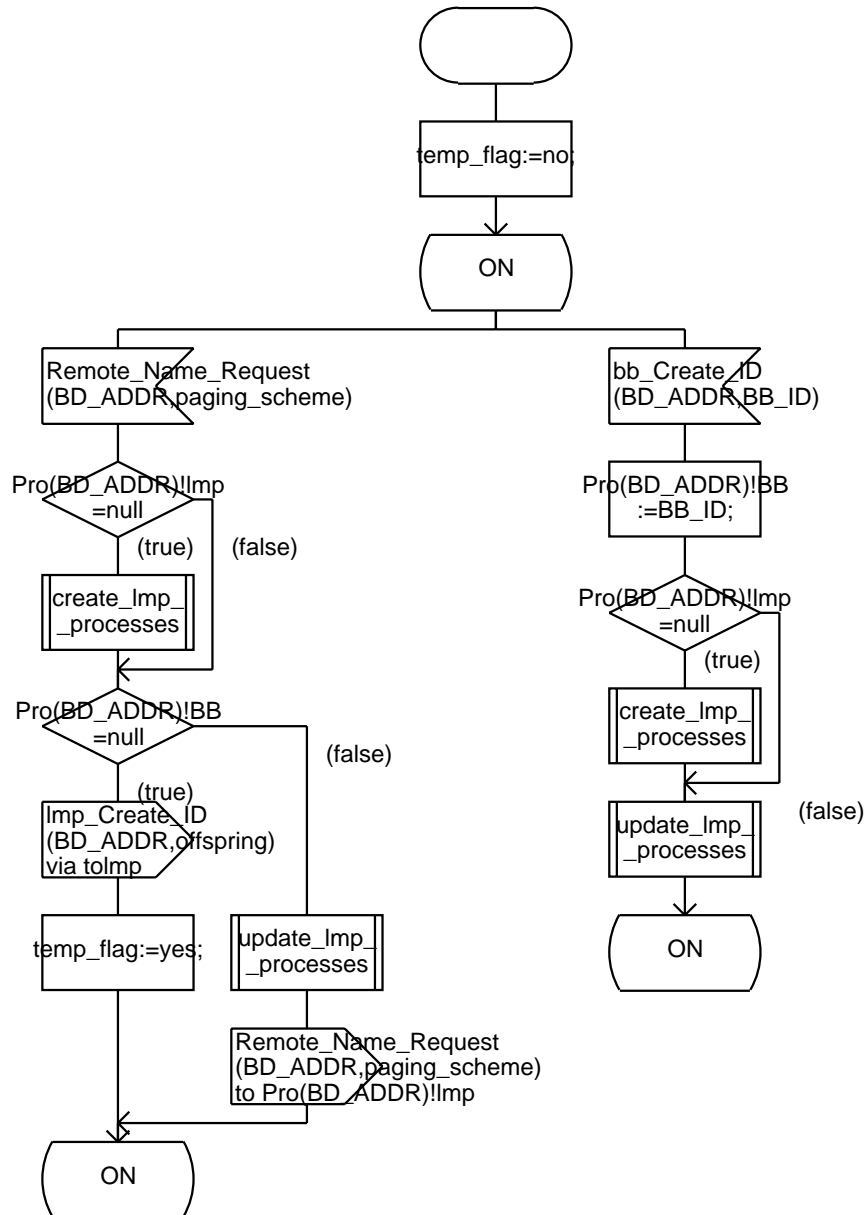
1(4)

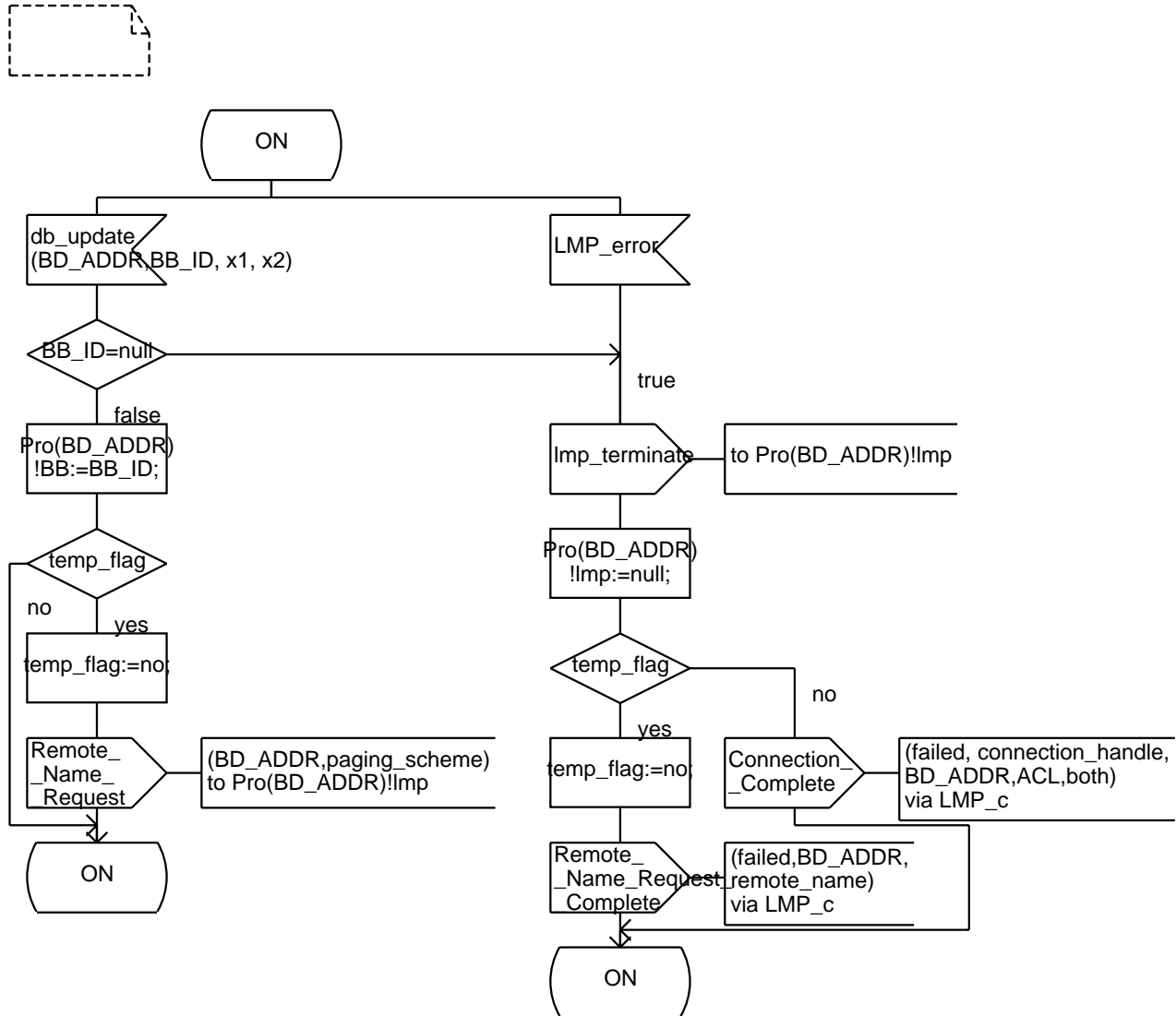


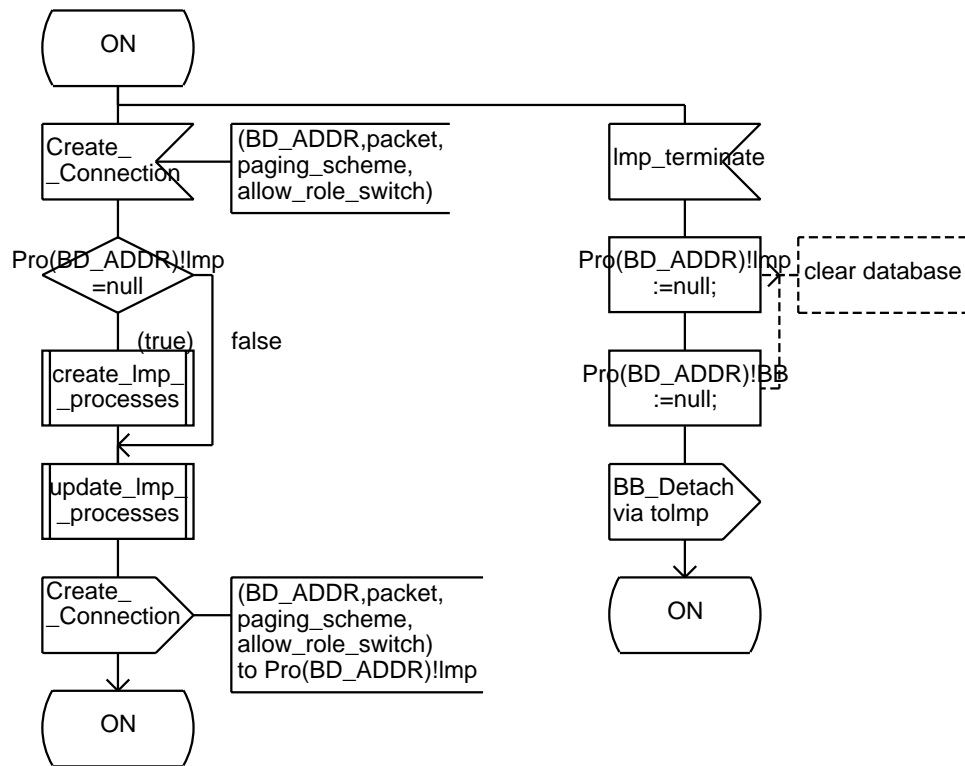
```
dcl Pro Lmp_R_tab;  
dcl BB_ID PId;  
dcl BD_ADDR Address_type;  
dcl packet Packet_type;  
dcl paging_scheme Paging_mode;  
dcl temp_flag,  
    allow_role_switch Decision_type;  
dcl connection_handle Handle_type;  
dcl remote_name Name_type;  
DCL  
x1, x2 PId;
```

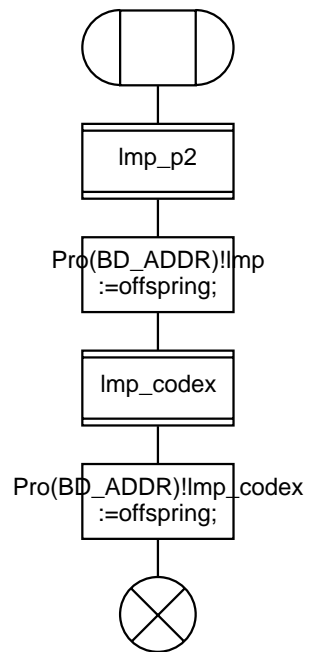
create_imp_
processes

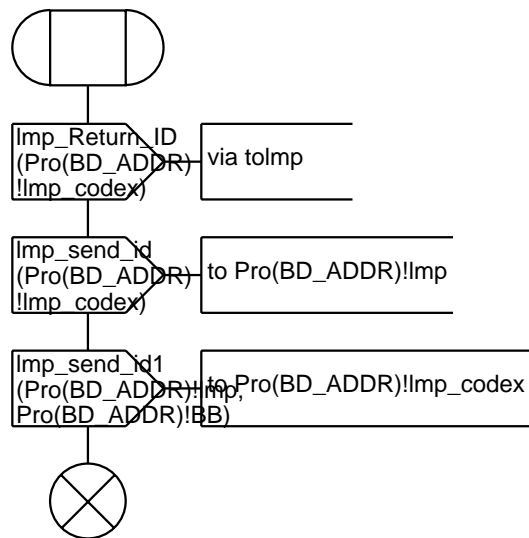
update_imp_
processes













```

timer Auth, Hold, Supervise;
dcl Pro Lmp_R_tab;
dcl BB_ID PId;
dcl BD_ADDR, AM_ADDR, AR_ADDR, PM_ADDR Address_type;
dcl connection_handle Handle_type;
dcl service_level Service_type;
dcl packet Packet_type;
dcl packet1 Packet_type1;
dcl device_type, role, trans_ID Role_type;

dcl Dsniff, Tsniff, sniff_attempt, sniff_timeout, clock_offset, TB,
    broadcast_scan_window, poll_interval, Dsco, Tsco, drift, jitter,
    slot_offset, auth_value, auth_timeout, hold_time, super_timeout Duration;
dcl features, LMP_version, op_code, name_offset, name_length,
    N, key_size, DB, NB, data_rate, VersNr, Compld, SubVersNr,
    NBC, max_slots, power_level, max_power, min_power, auth_time,
    snd_key, max_times, enabling, rcv_hold_req,
    rcv_sniff_req, SCO_number, size_ACL Integer;

dcl encr_enable Encryption_enable;
dcl encr_mode Encryption_mode;
dcl key, rand Key_content;
dcl key_flag, temp_key Key_flag_type;
dcl link_key Key_type;
dcl reason Imp_reason_type;
dcl auth_res Response_content;
dcl auth_done Imp_status_type1;
dcl encry_mode_done Imp_status_type2;
dcl remote_name, name, name_fragment Name_type;
dcl paging_scheme Paging_mode;
dcl power_flag Power_type;

dcl encryption1, slot_offset1, timing_accuracy1, switch1,
    hold_mode1, sniff_mode1, park_mode1, channel_quality1,
    SCO_link1, paging_scheme1, power_control1,
    encryption2, slot_offset2, timing_accuracy2, switch2,
    hold_mode2, sniff_mode2, park_mode2, channel_quality2,
    SCO_link2, paging_scheme2, power_control2 Decision_type;
dcl allow_role_switch, change_link_key, change_current_link_key,
    pair_done, version_information, supported_features,
    name_request, snd_park_req, park_accepted,
    hold_req, command_enable Decision_type;
dcl common_key, fixed_PIN, slot_full,
    auth_res_correct, accept_pairing, req_switch1,
    snd_unit_key, key_missing, agree_encry_mode,
    return_key_size_req, accept_key_size, accept_switch,
    return_hold_req, accept_hold_mode, req_sniff,
    return_sniff_req, accept_sniff_mode, req_park,
    accept_park_mode, unpark_PM_ADDR, set_broadcast,
    req_QoS, accept_QoS, accept_max_slot_req,
    support_timing_accuracy, accept_page_mode,
    accept_page_scan_mode, ls_my_address, accept_DM_DH,
    mutual_auth, page_scan, RSSI_incr Decision_type;

```


process Imp_p2

2(74)



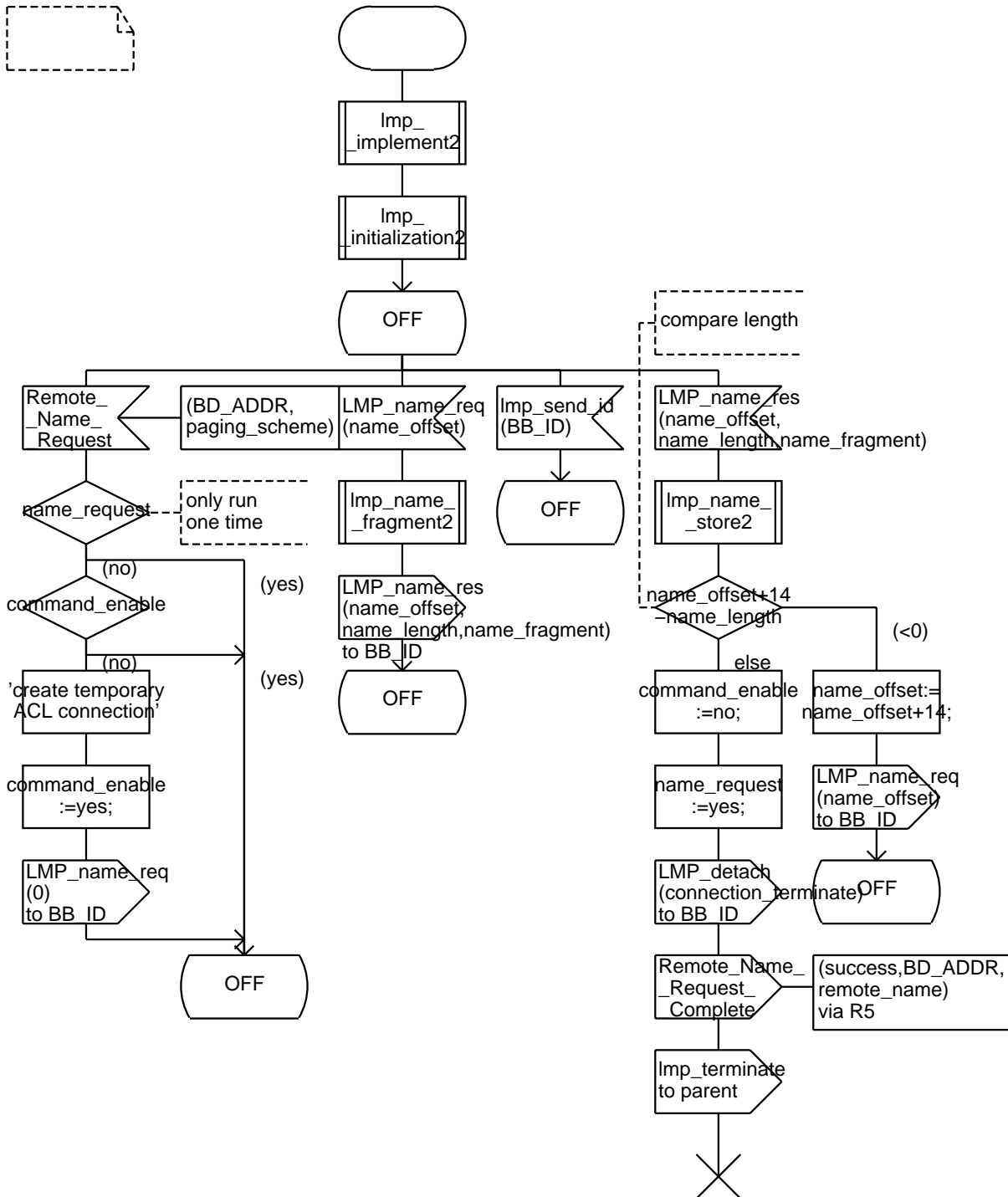
Imp_implement2

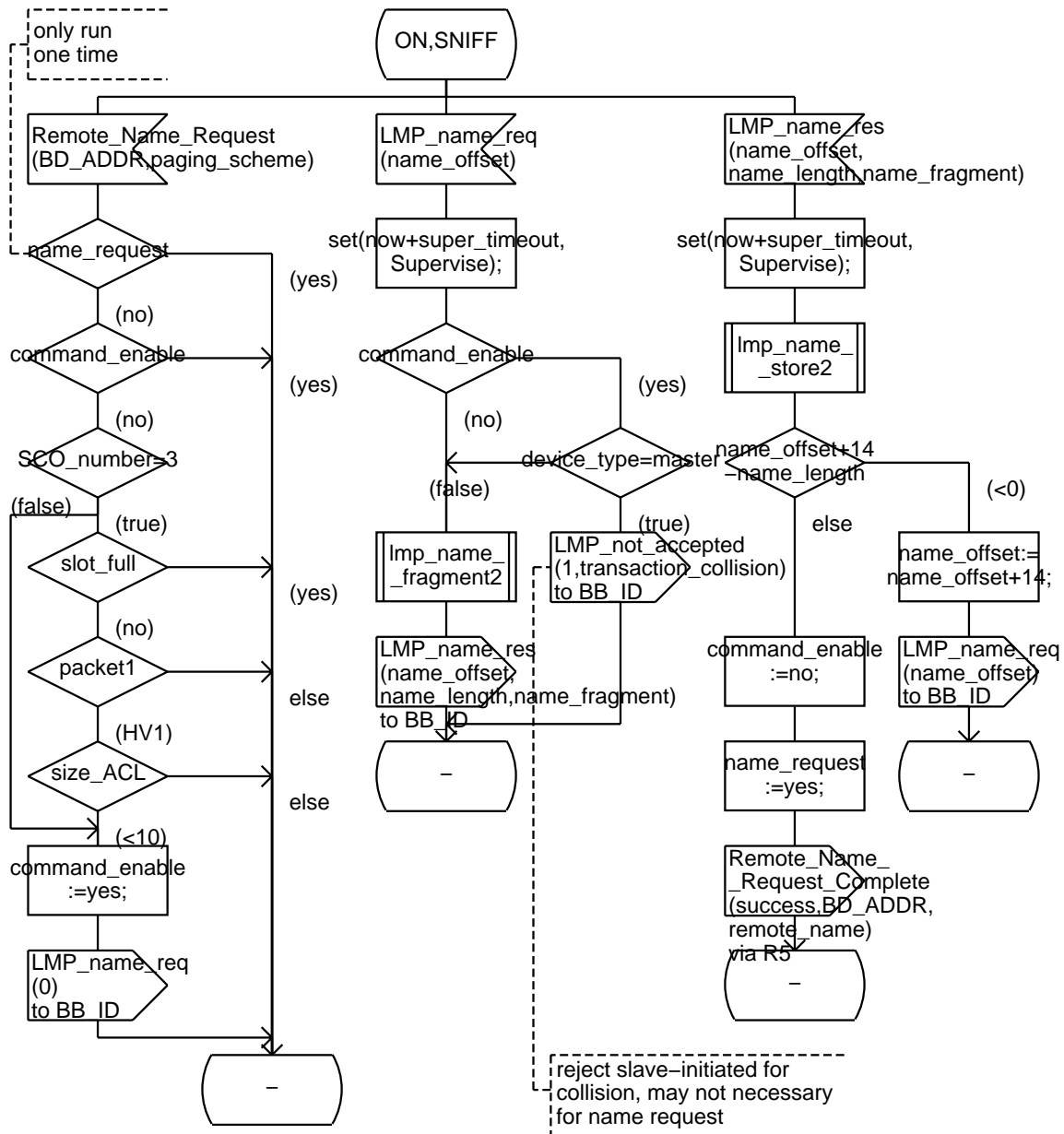
Imp_initialization2

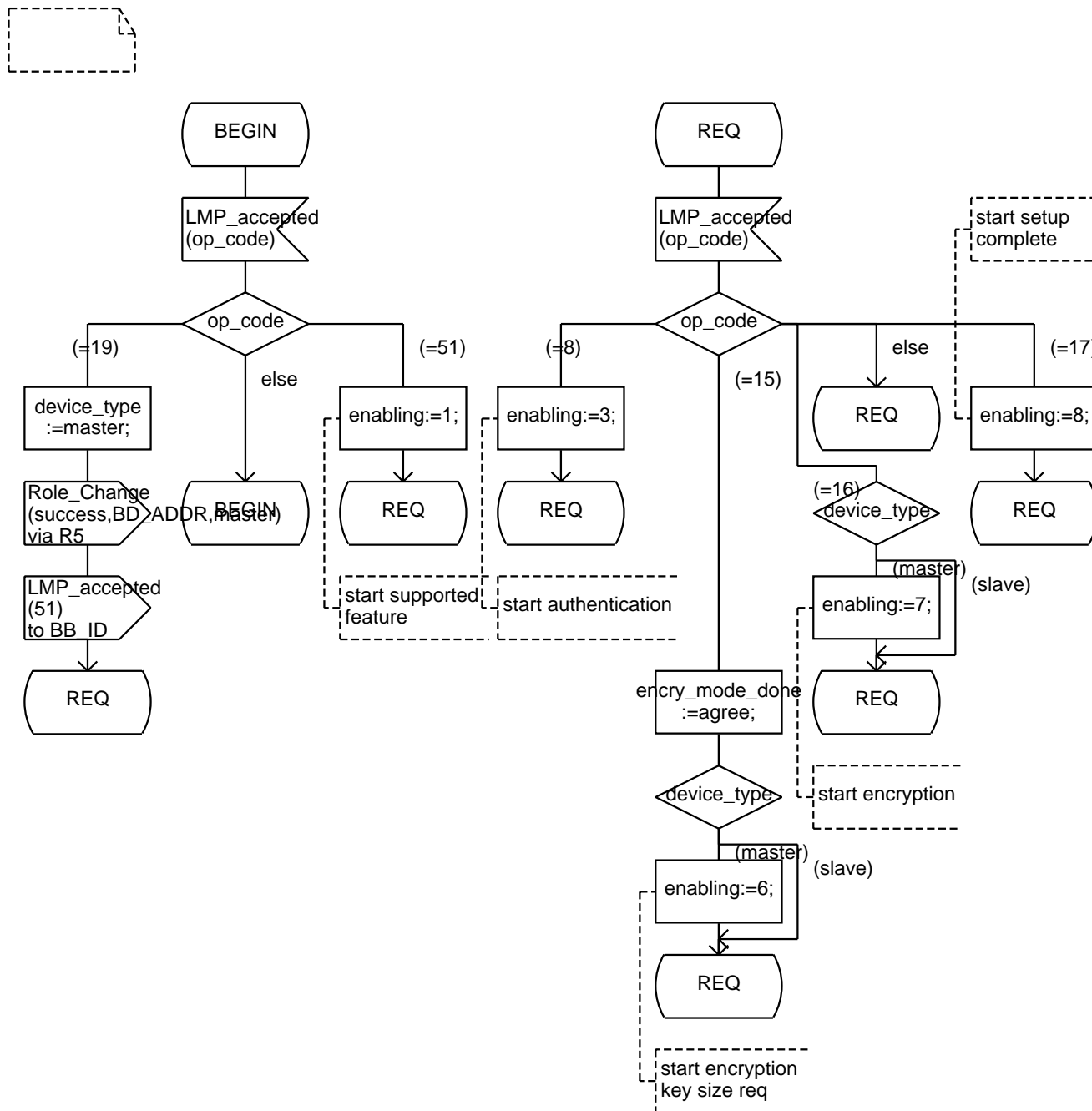
Imp_name_
_fragment2

Imp_name_
_store2

Imp_features2

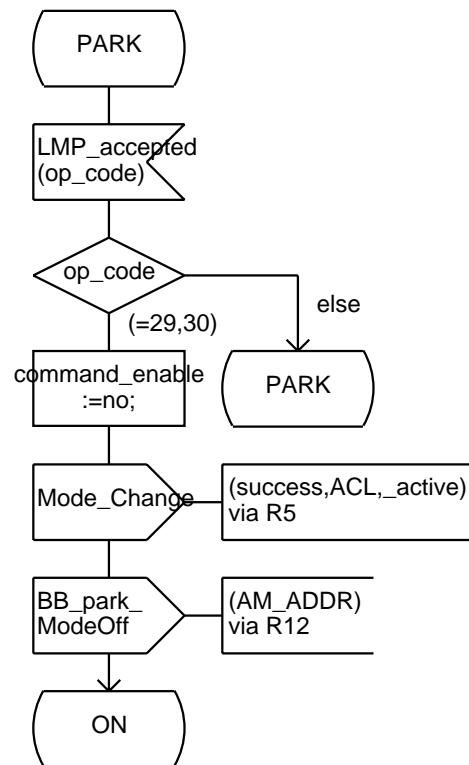


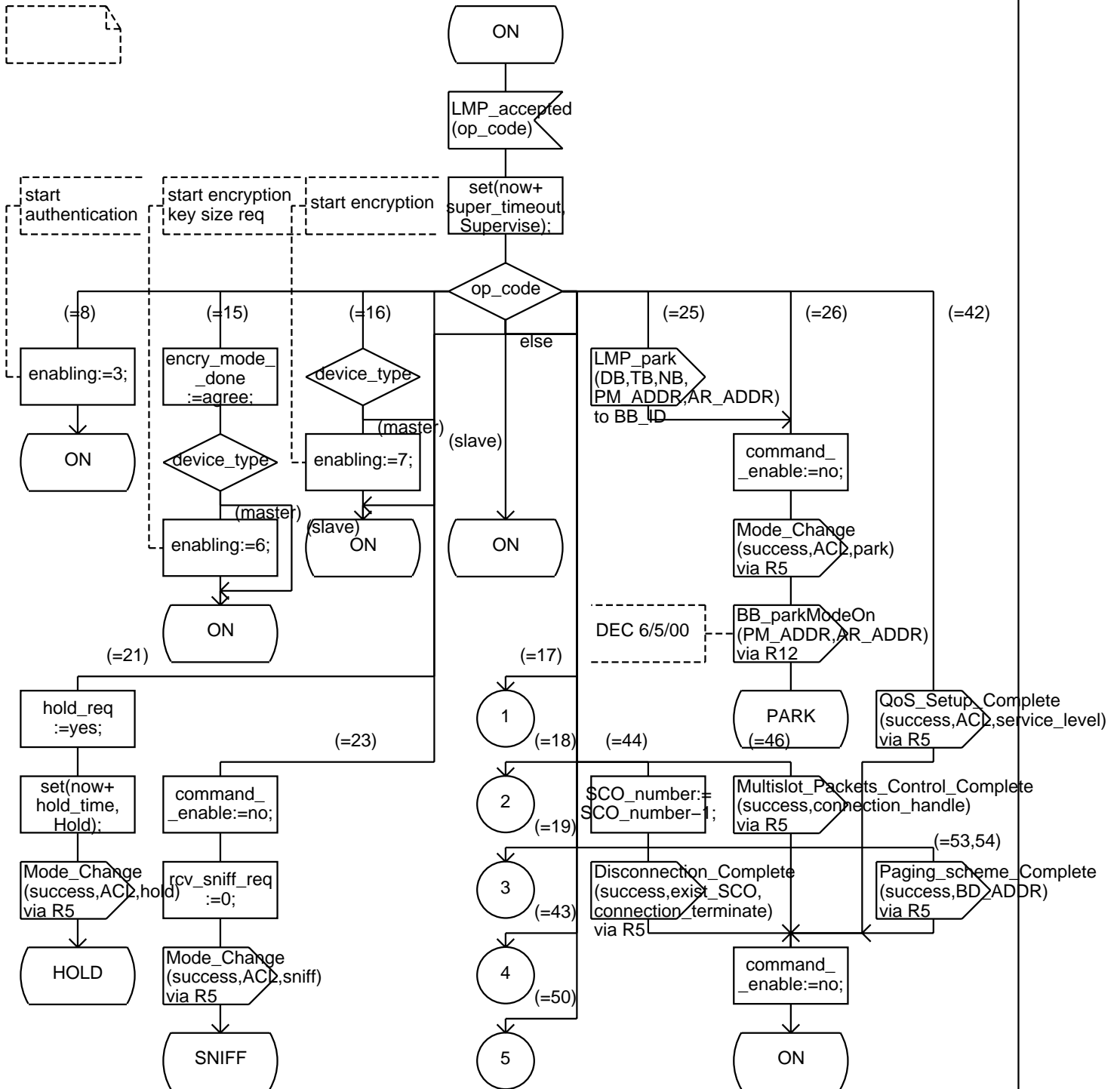


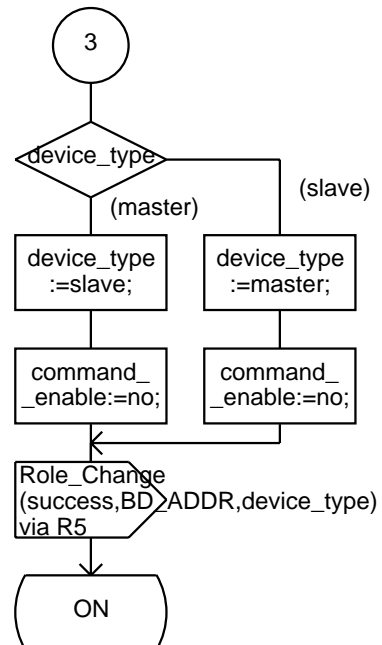
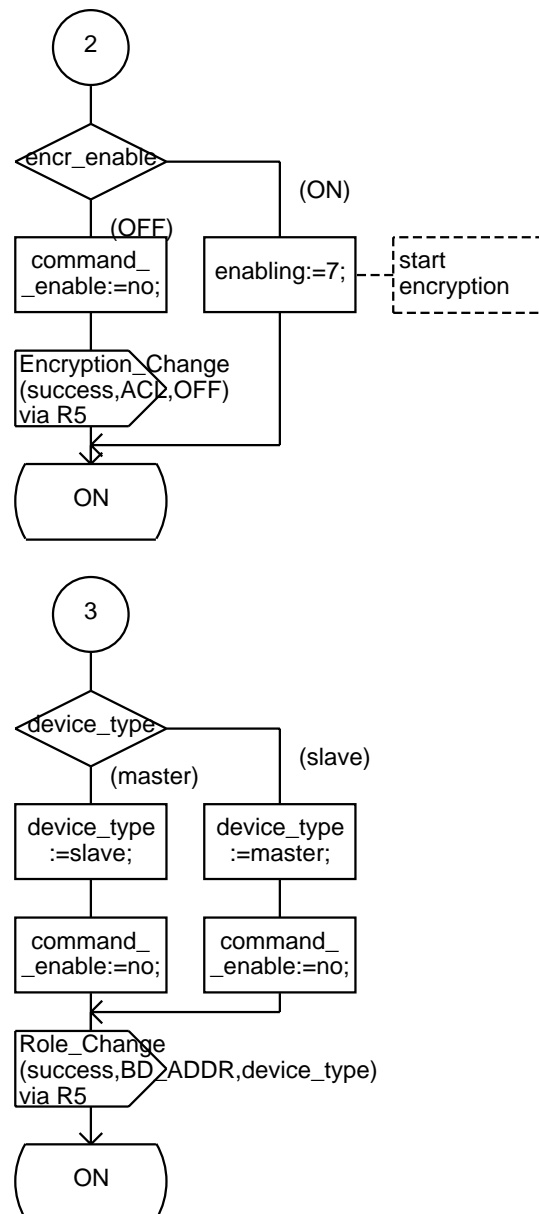
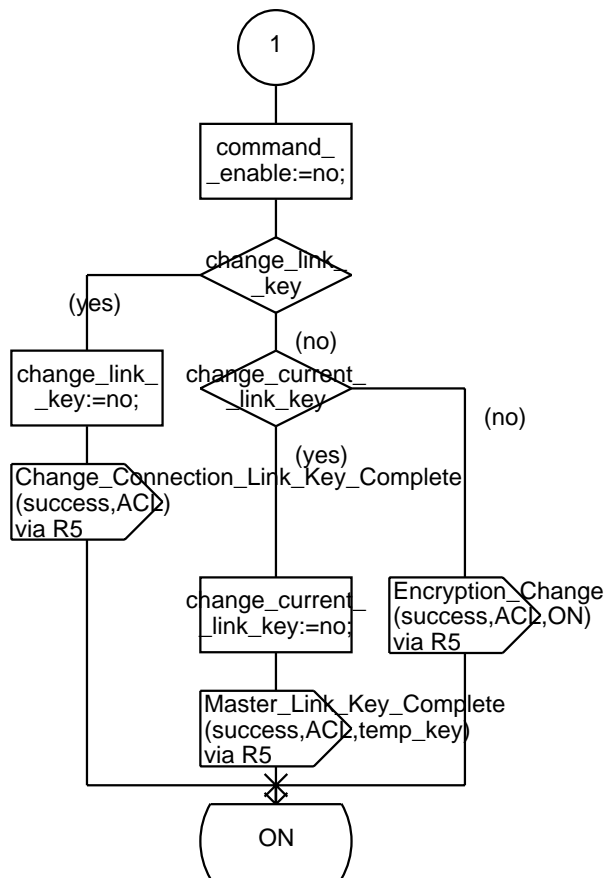


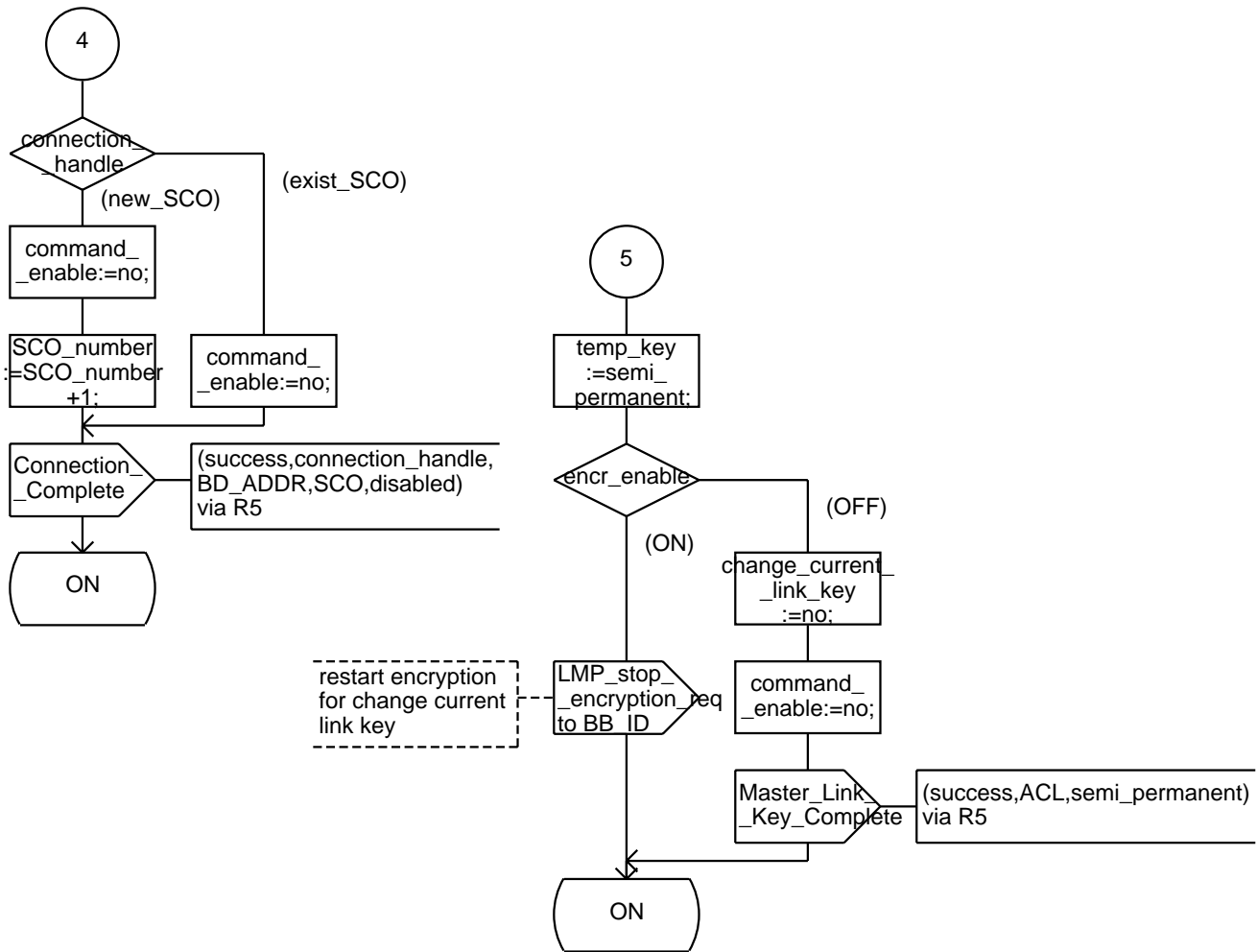
process Imp_p2

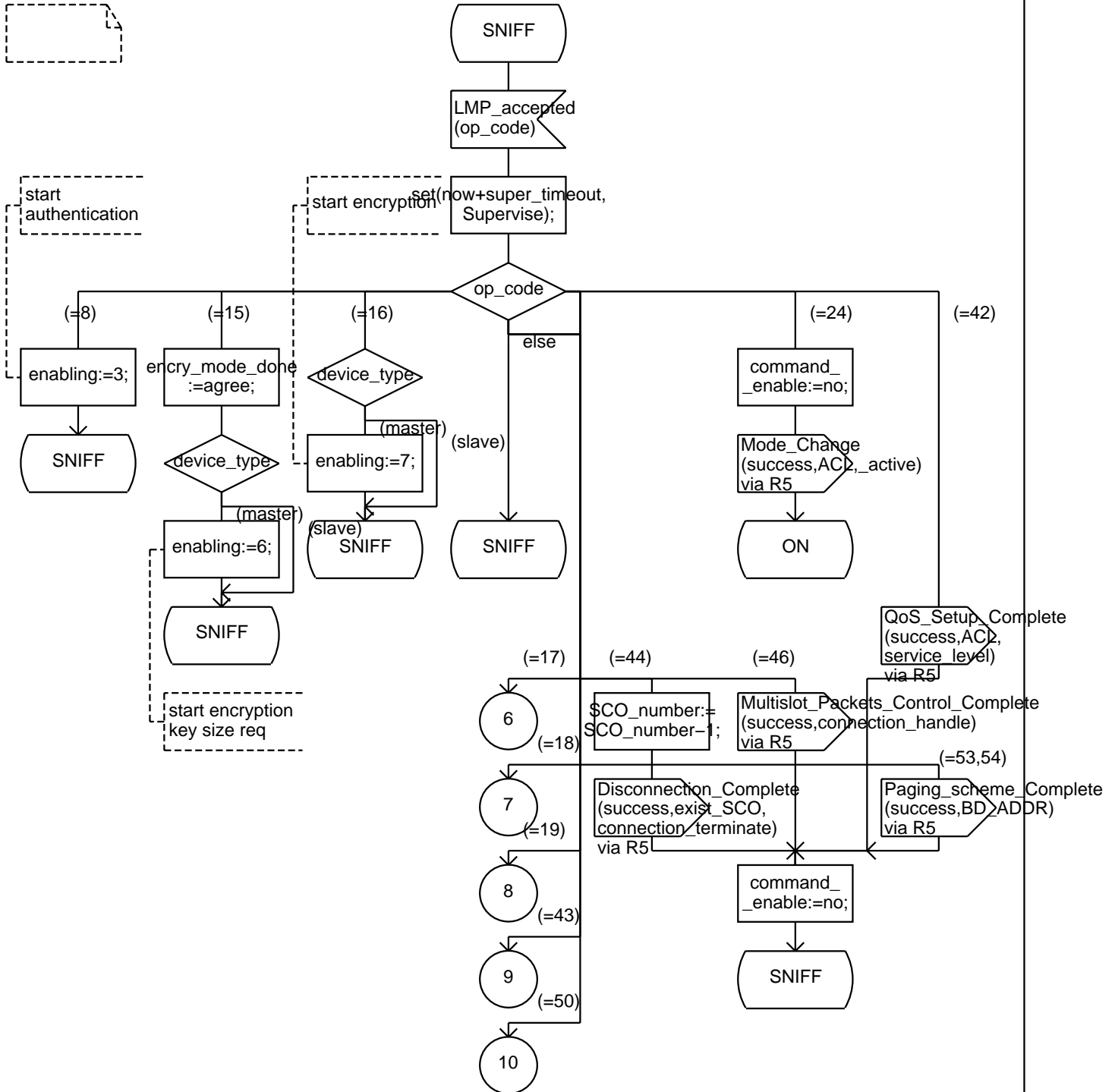
6(74)





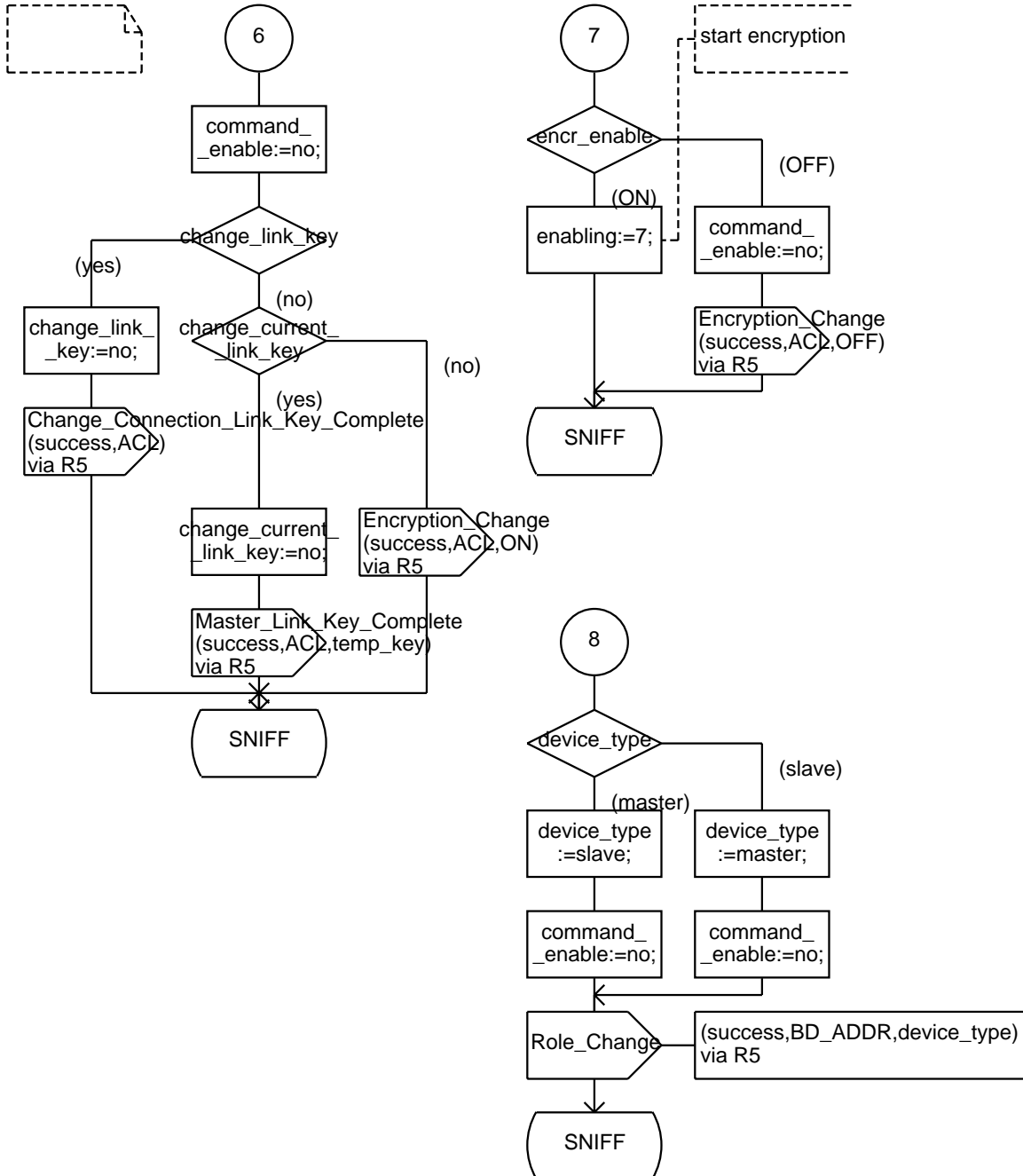


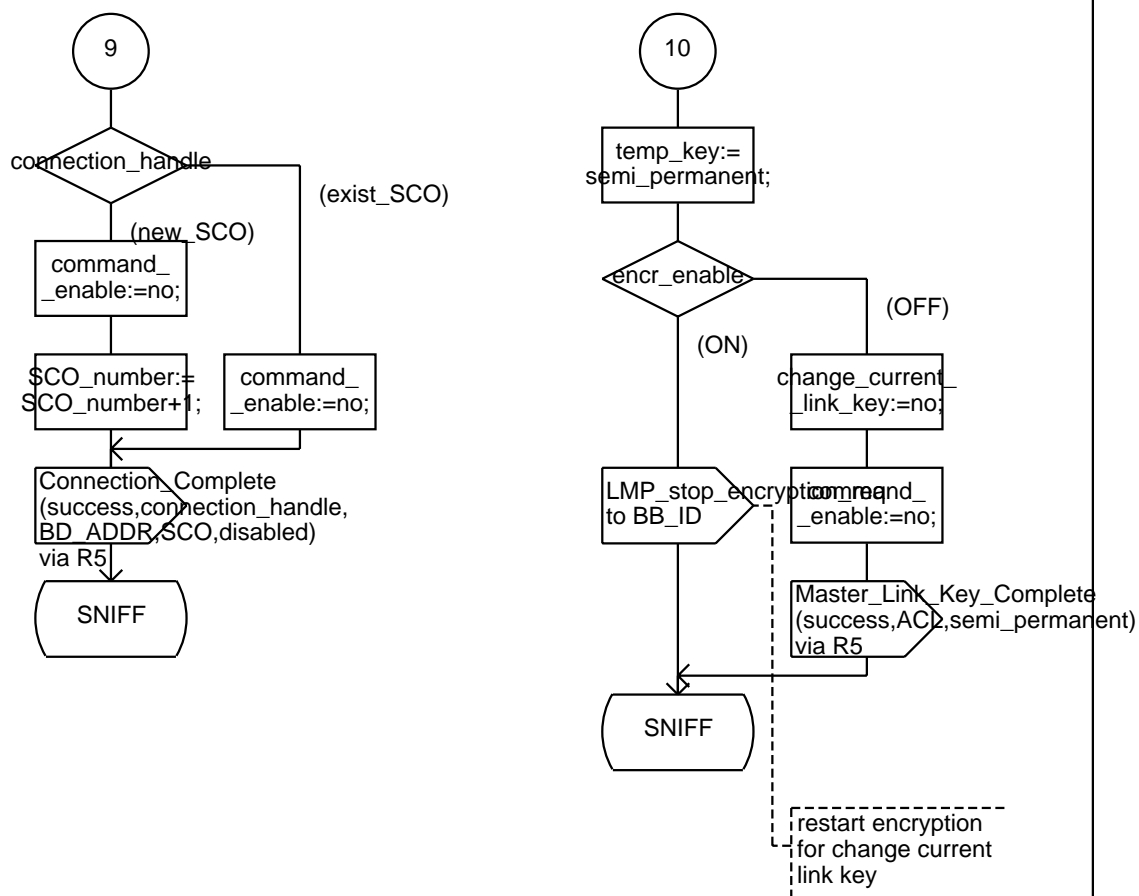




process Imp_p2

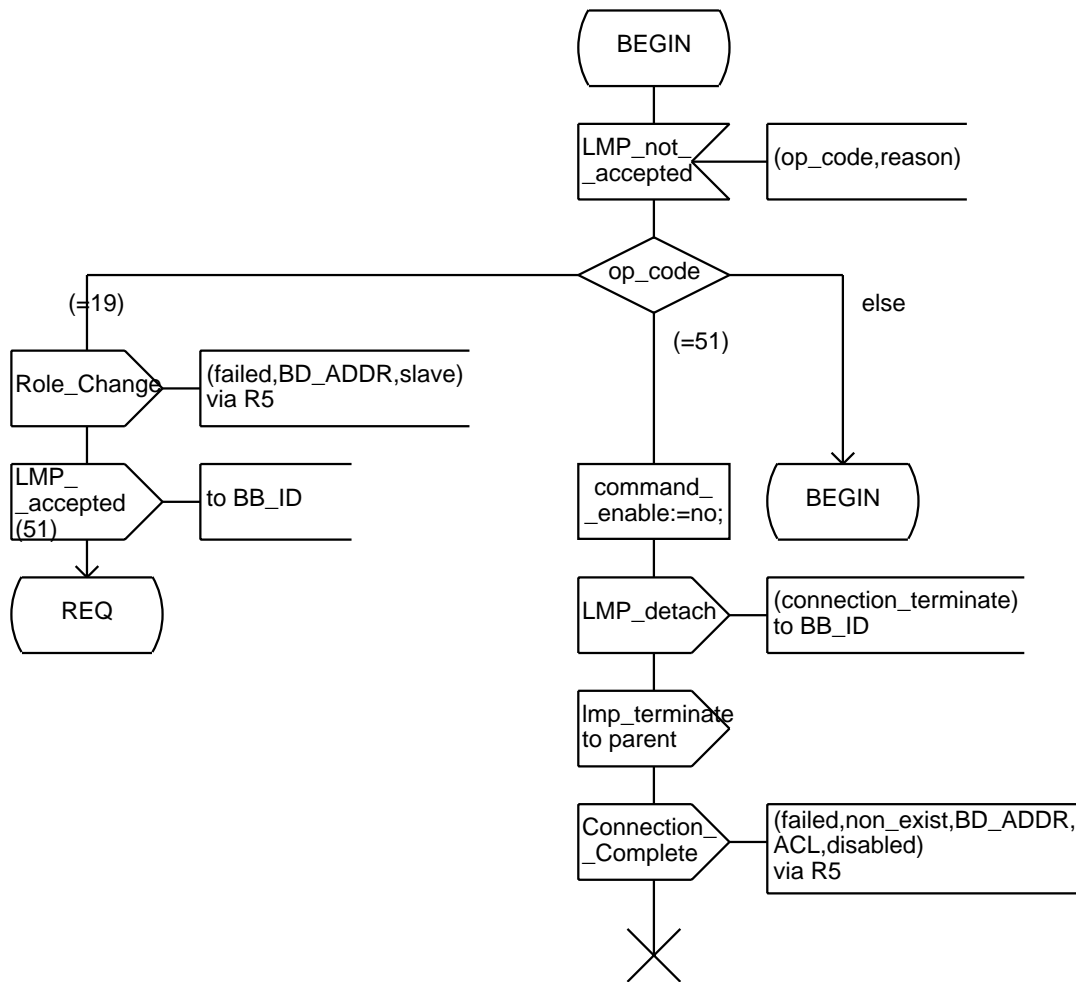
11(74)

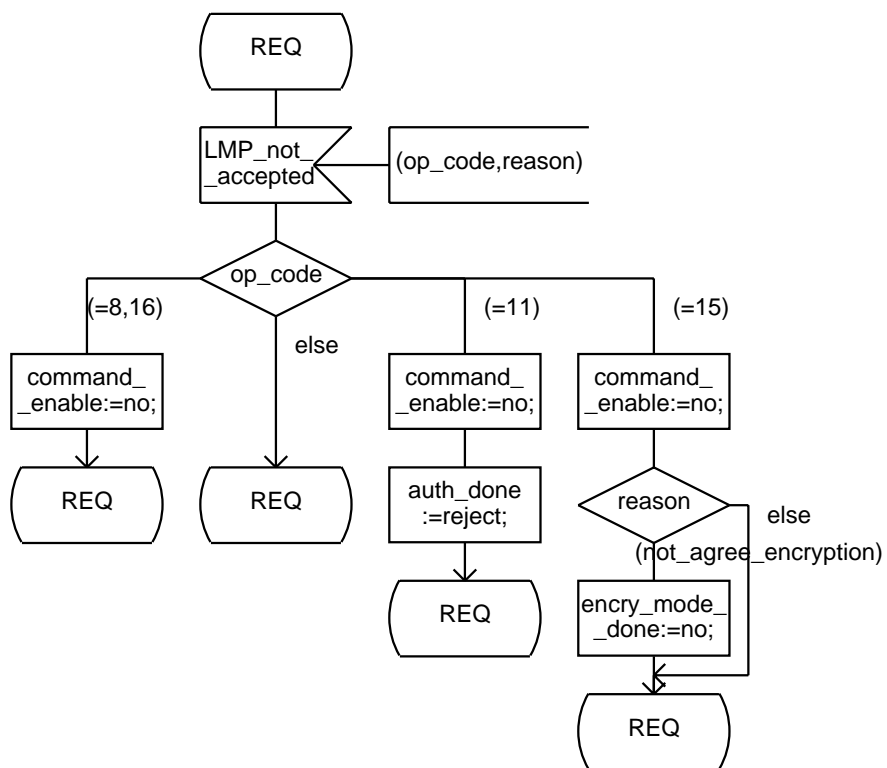




process Imp_p2

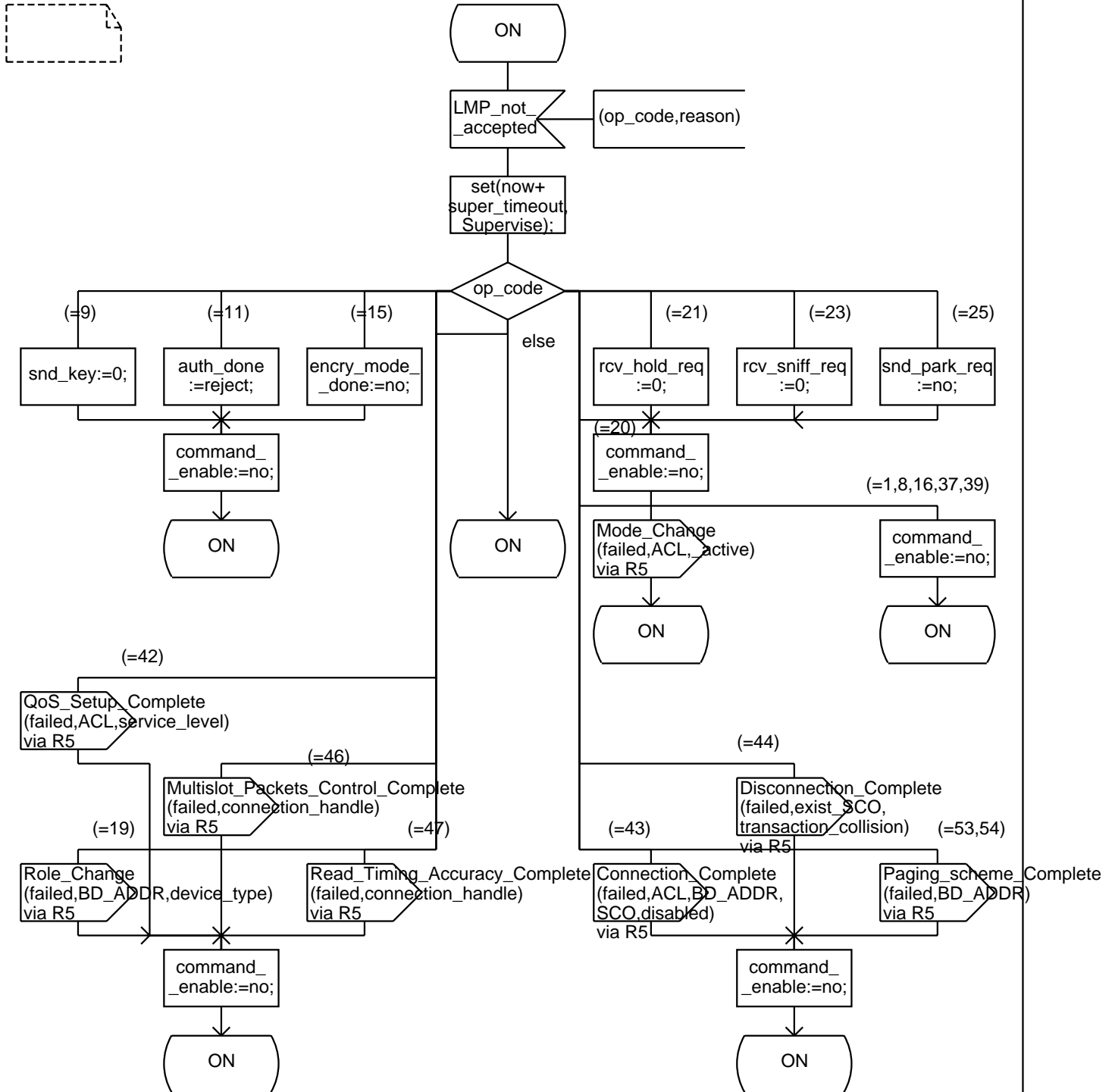
13(74)

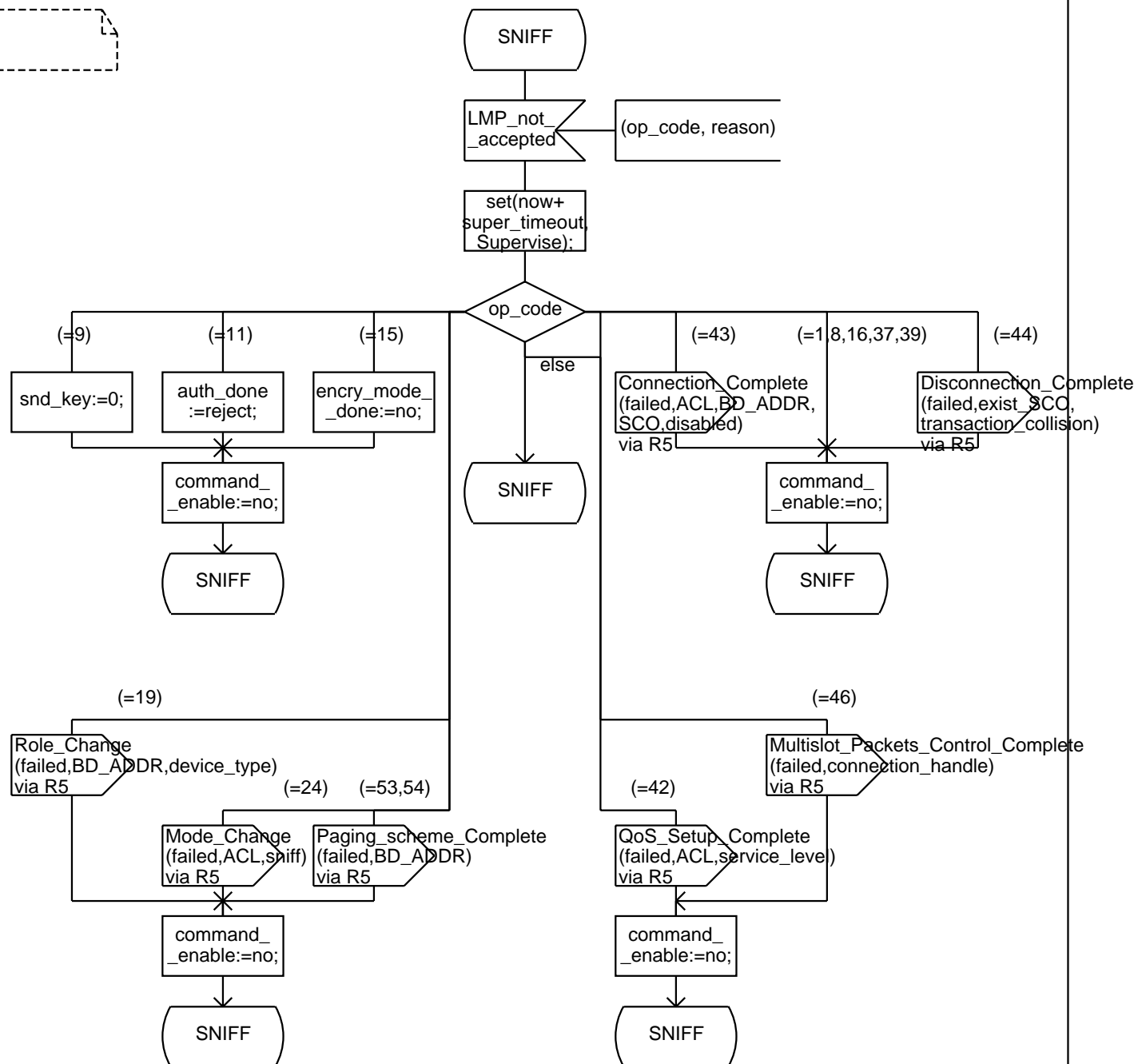


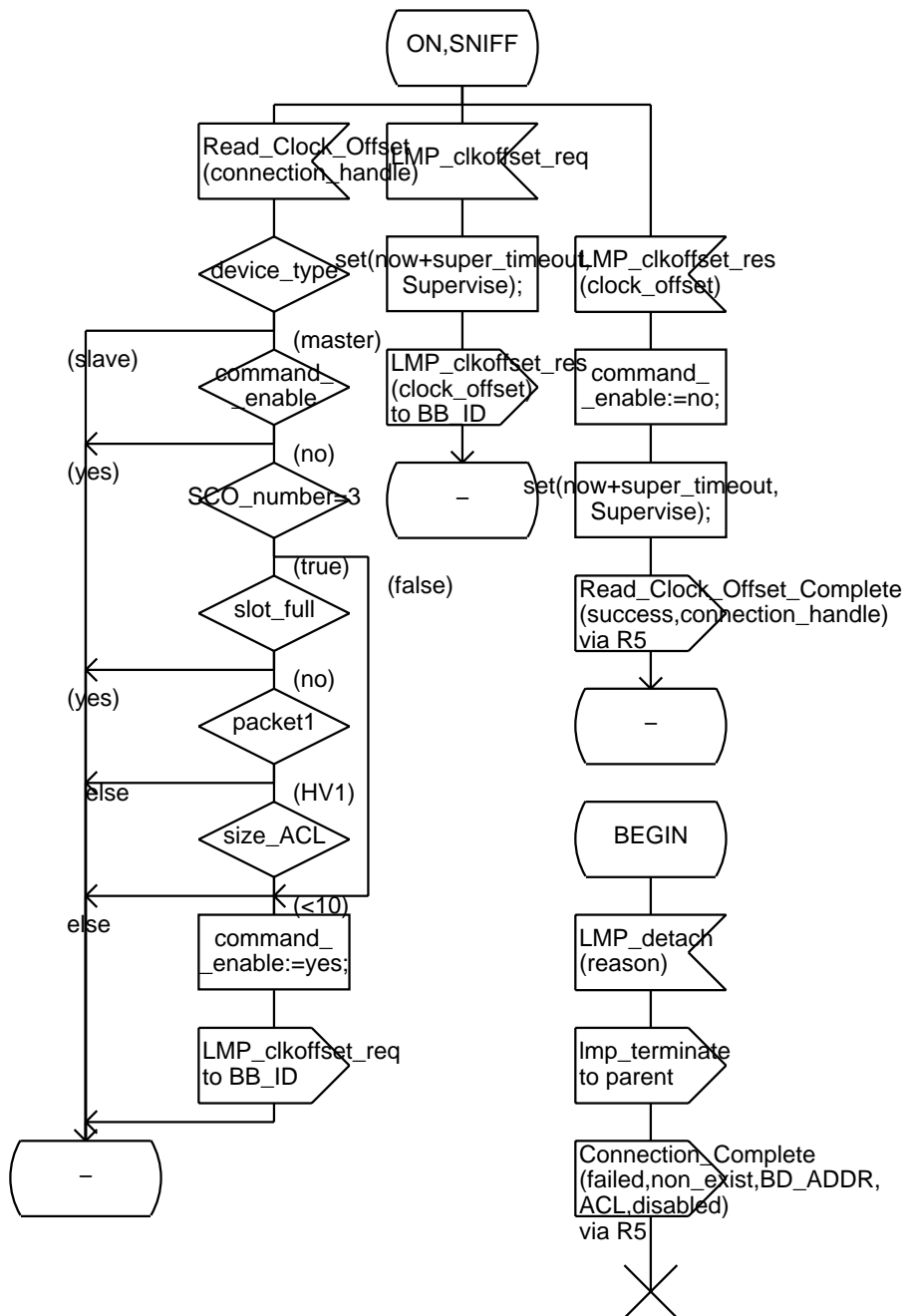


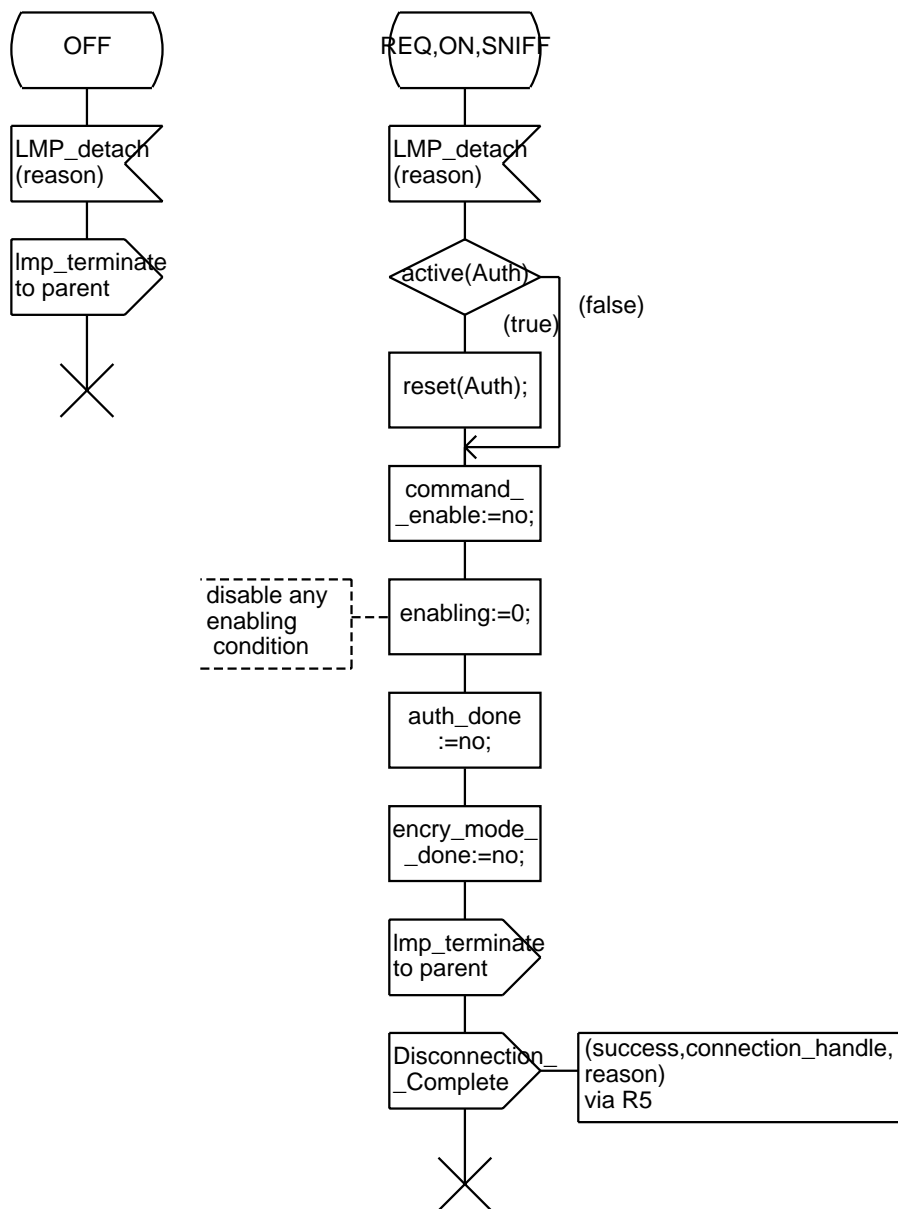
process lmp_p2

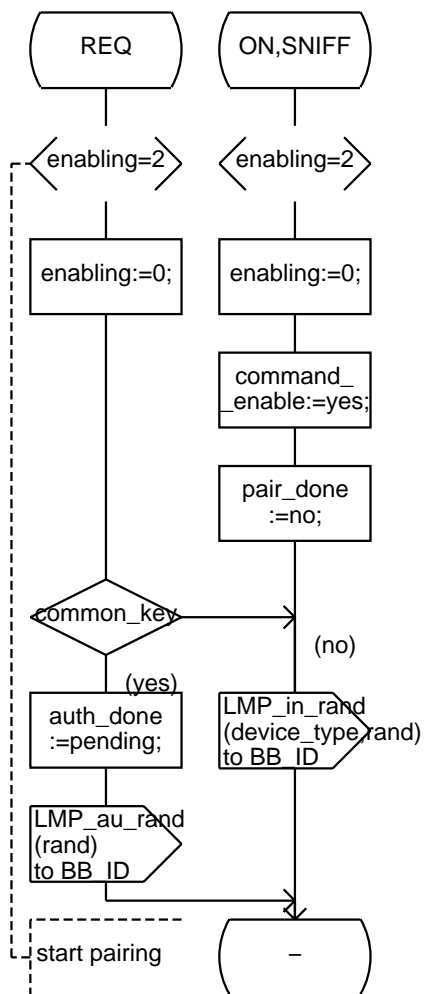
15(74)

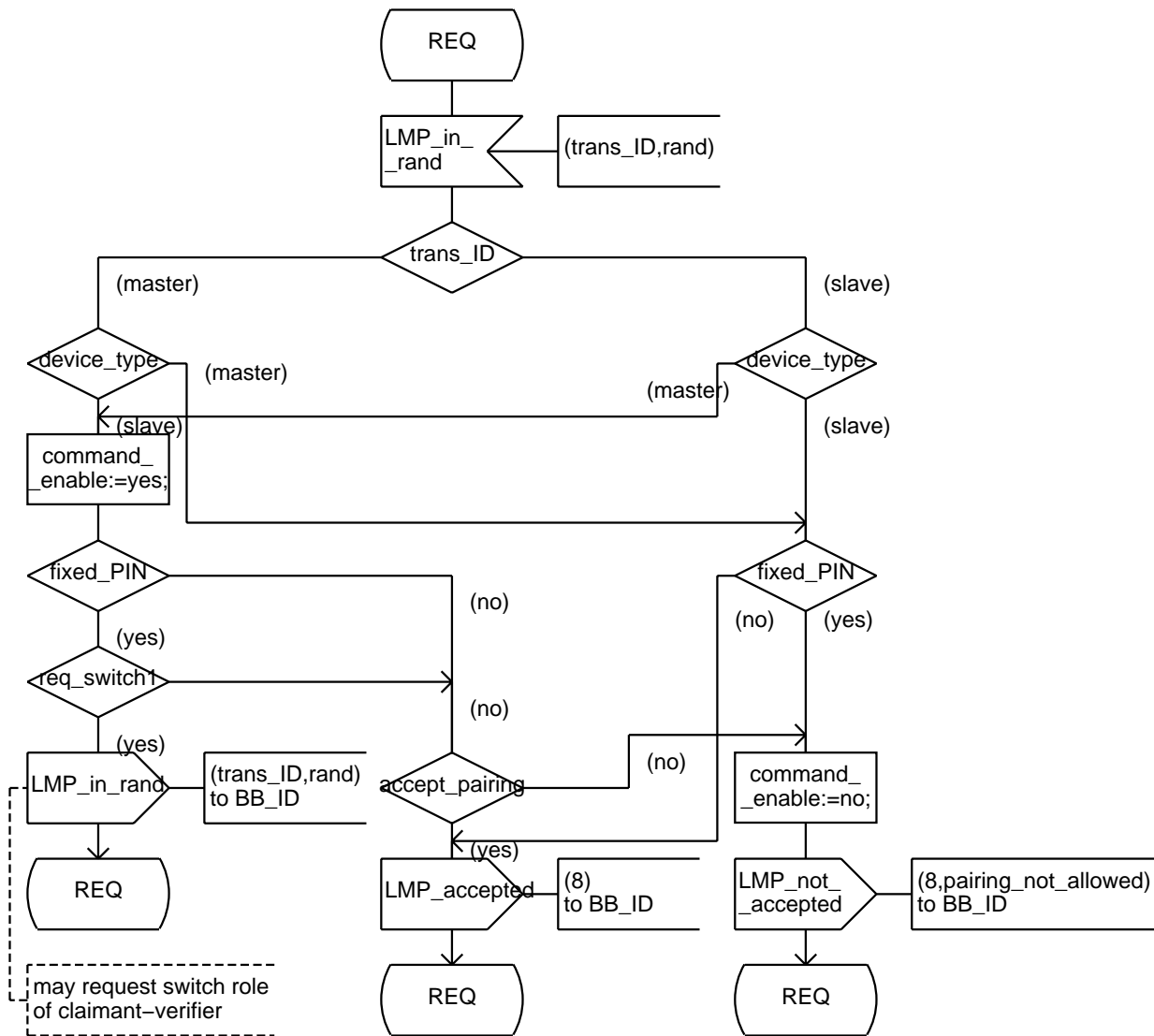


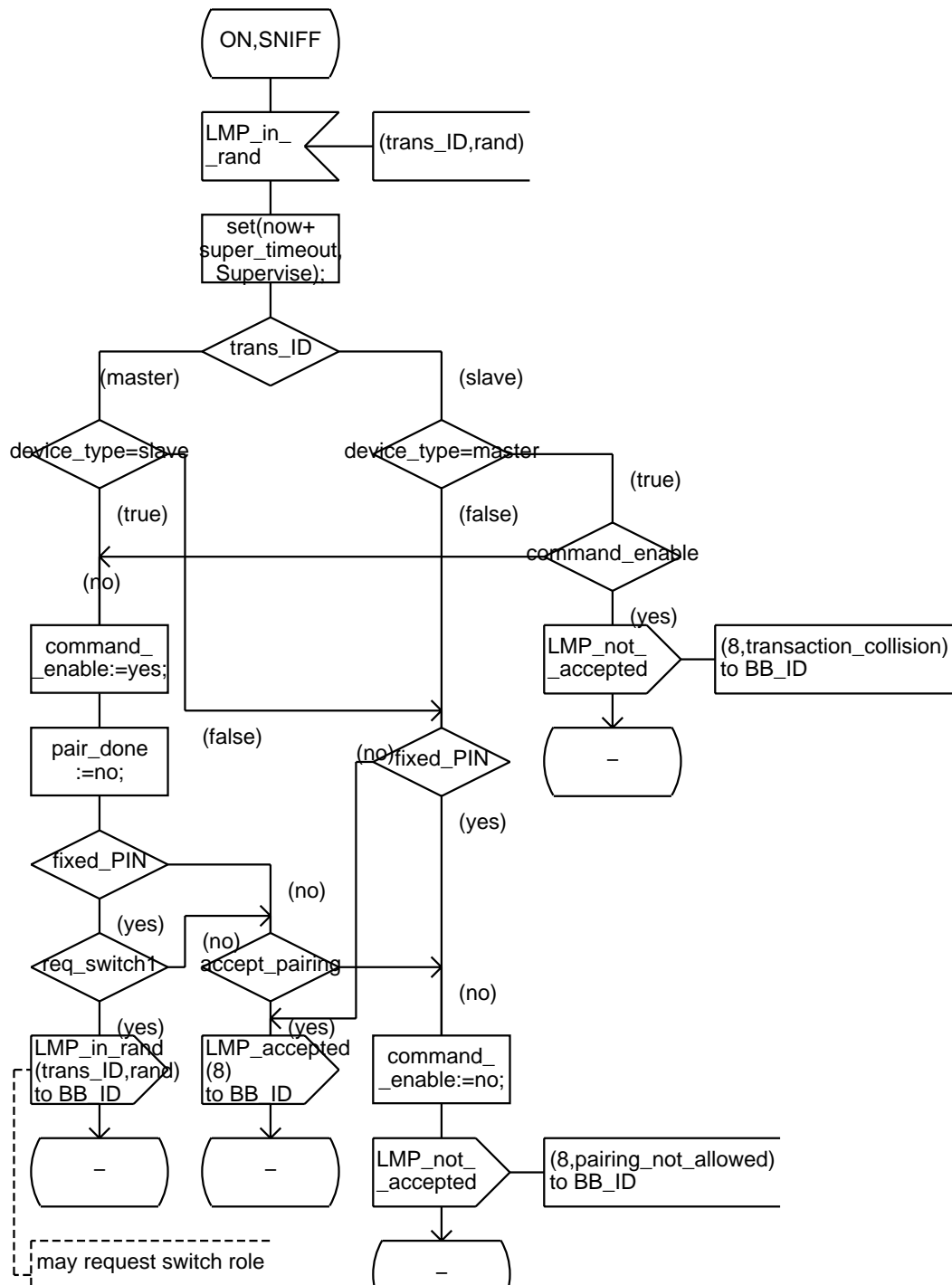






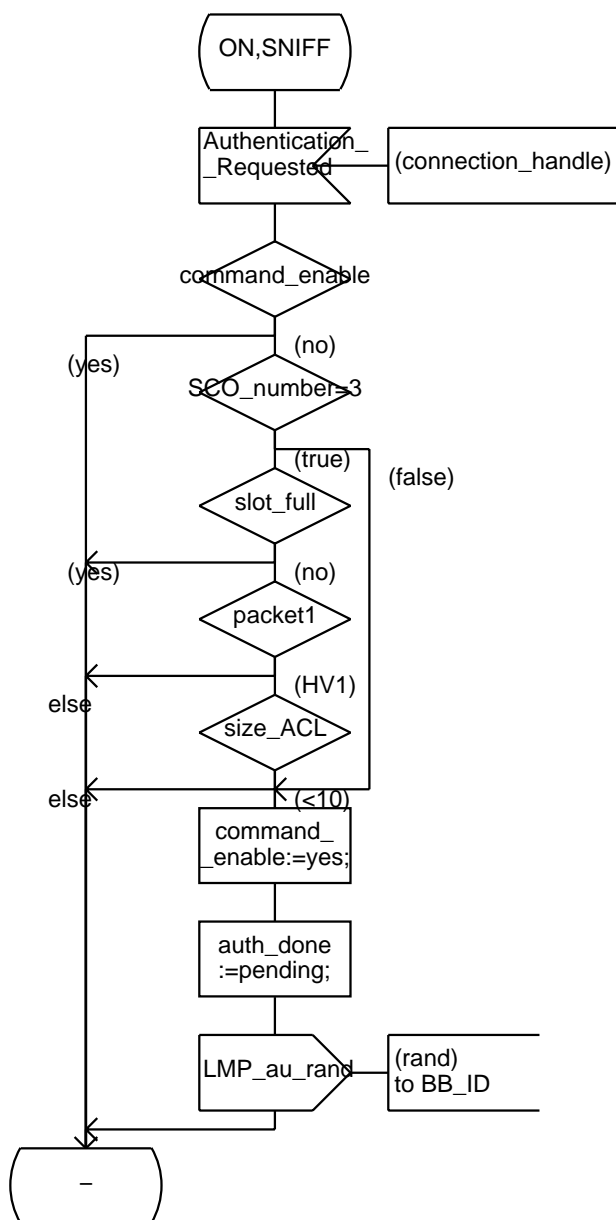


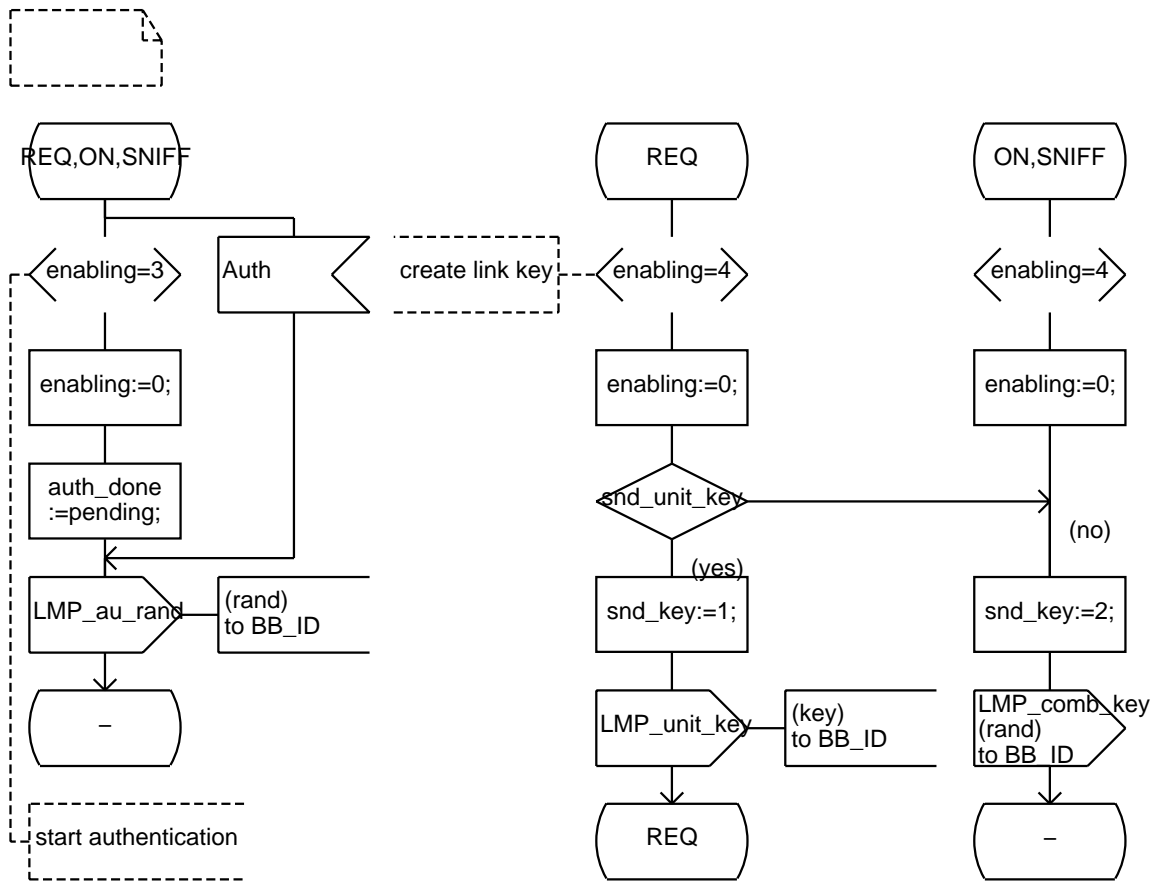


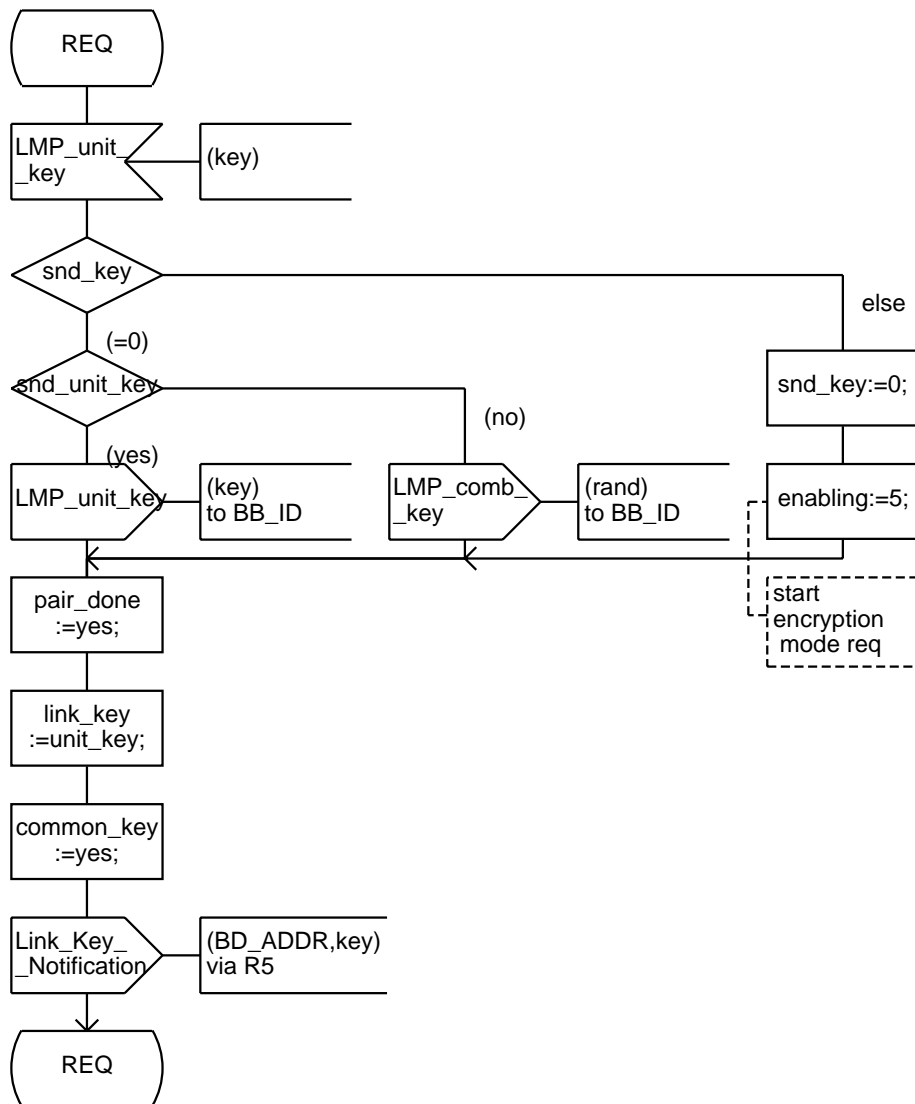


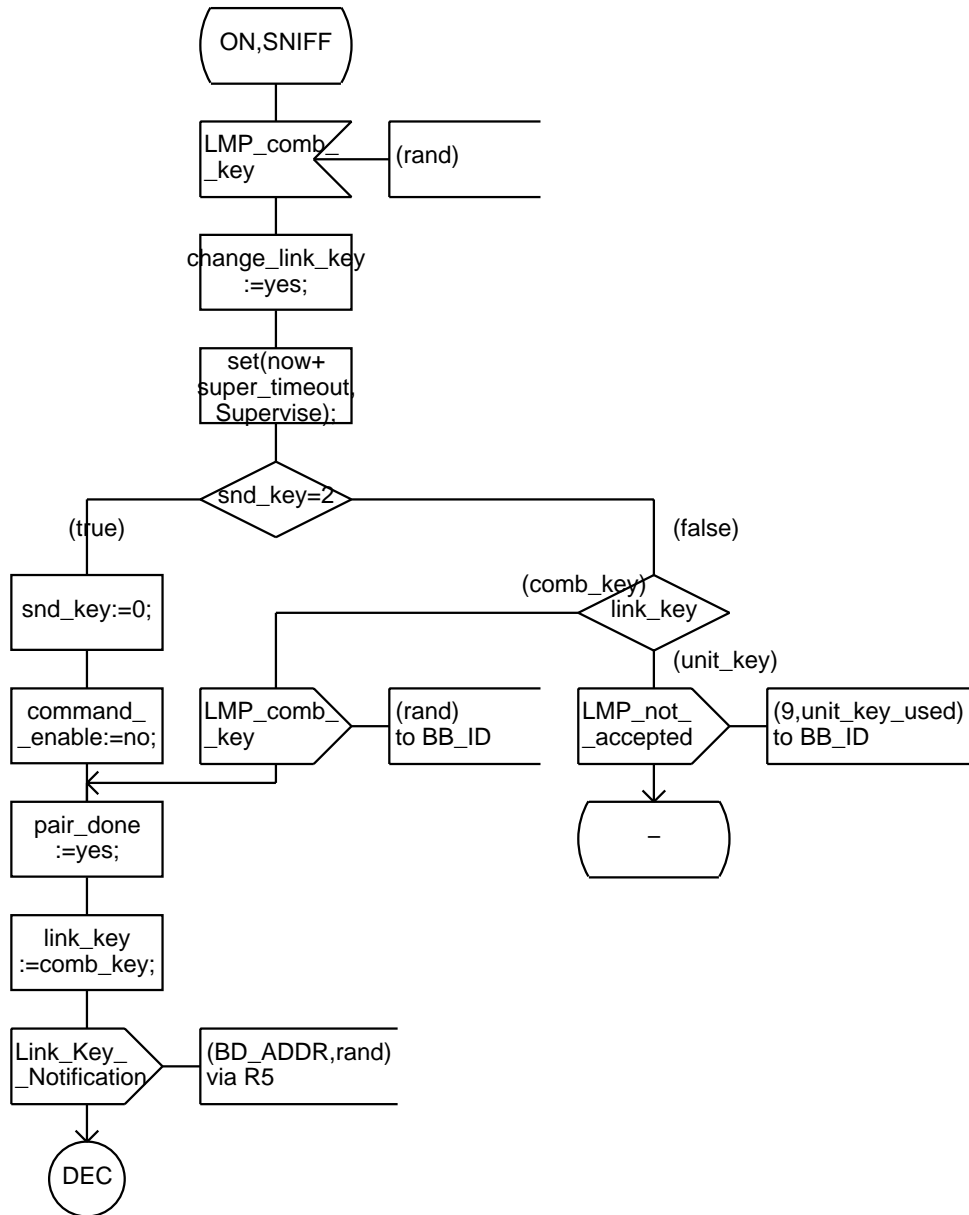
of claimant-verifier

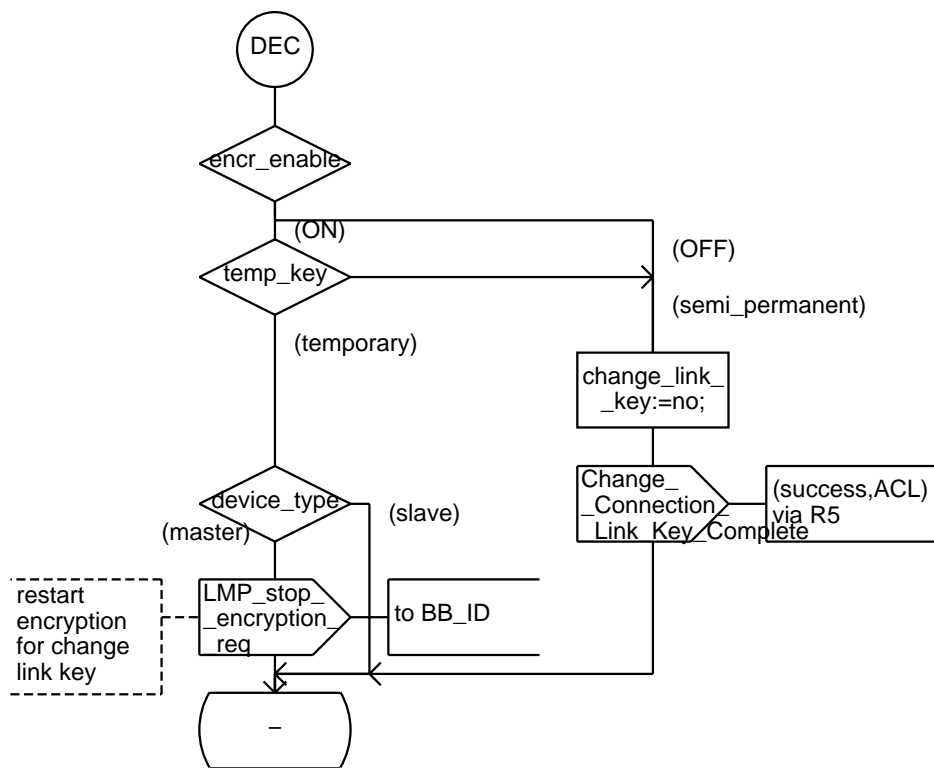




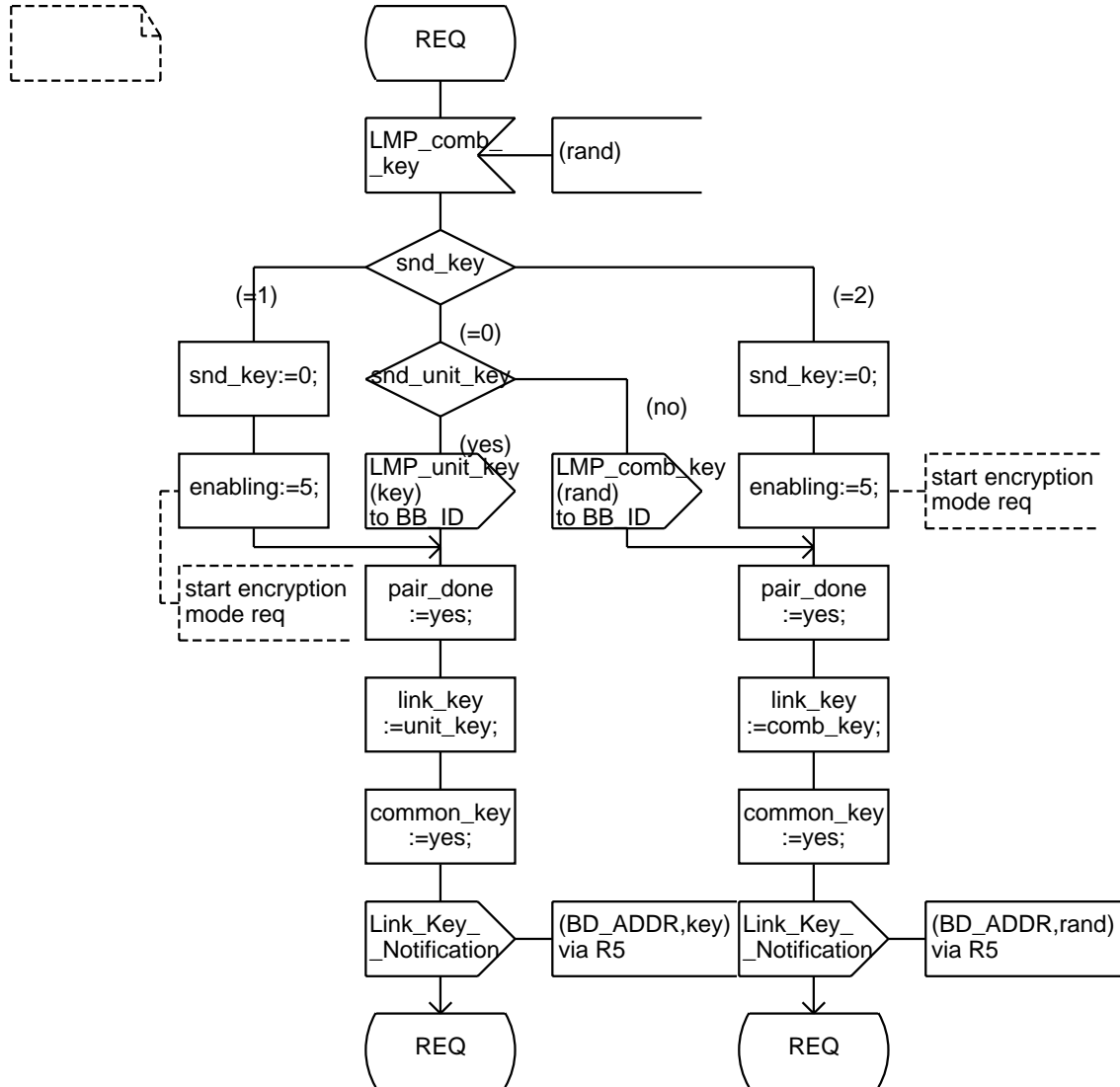


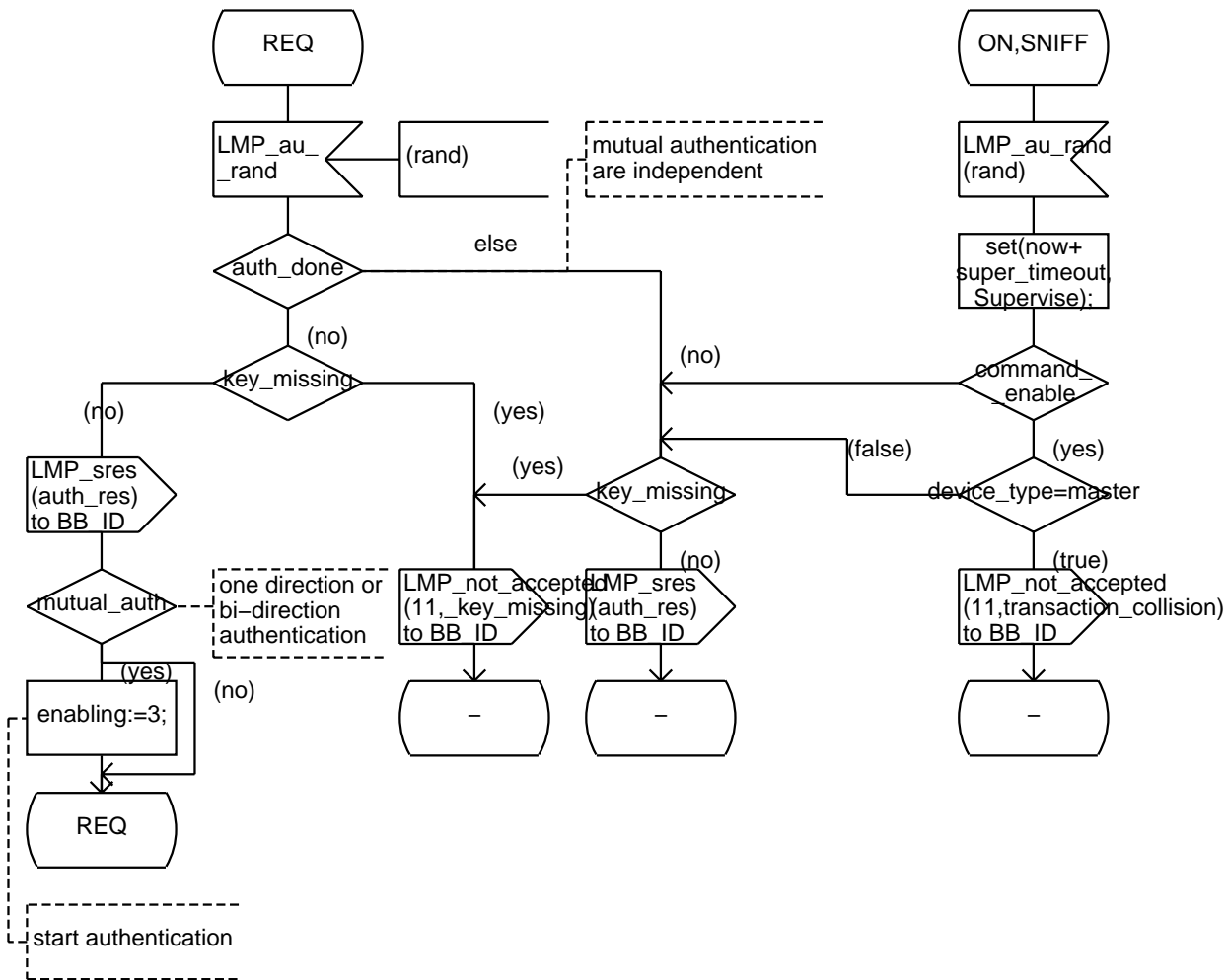


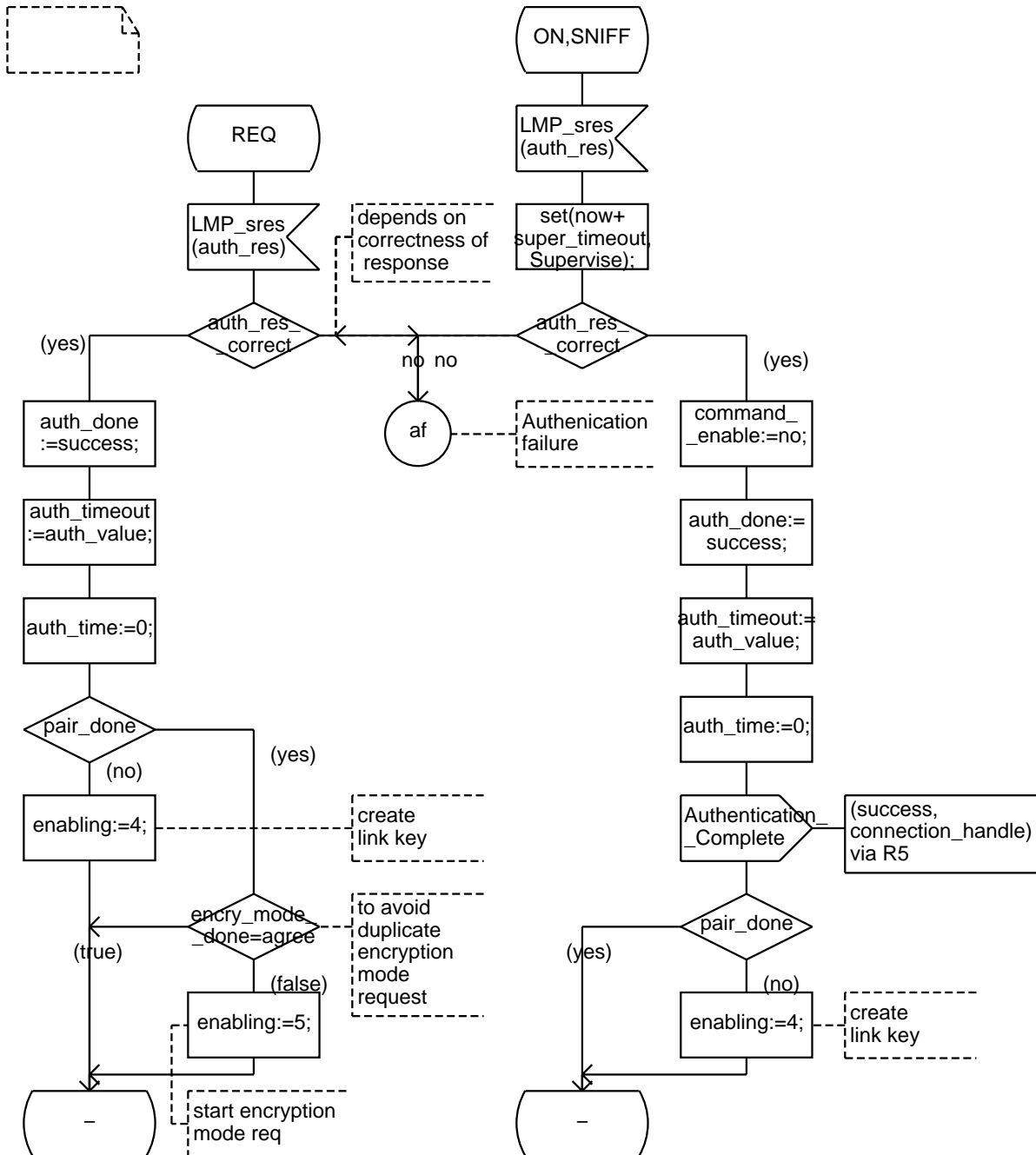


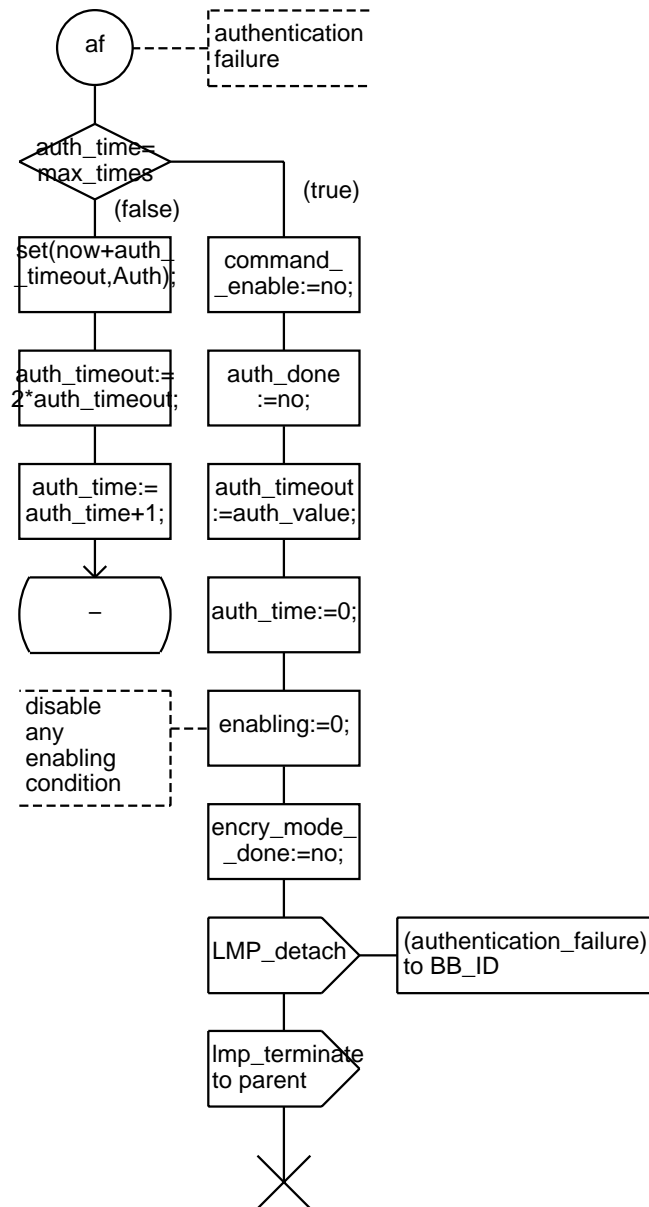


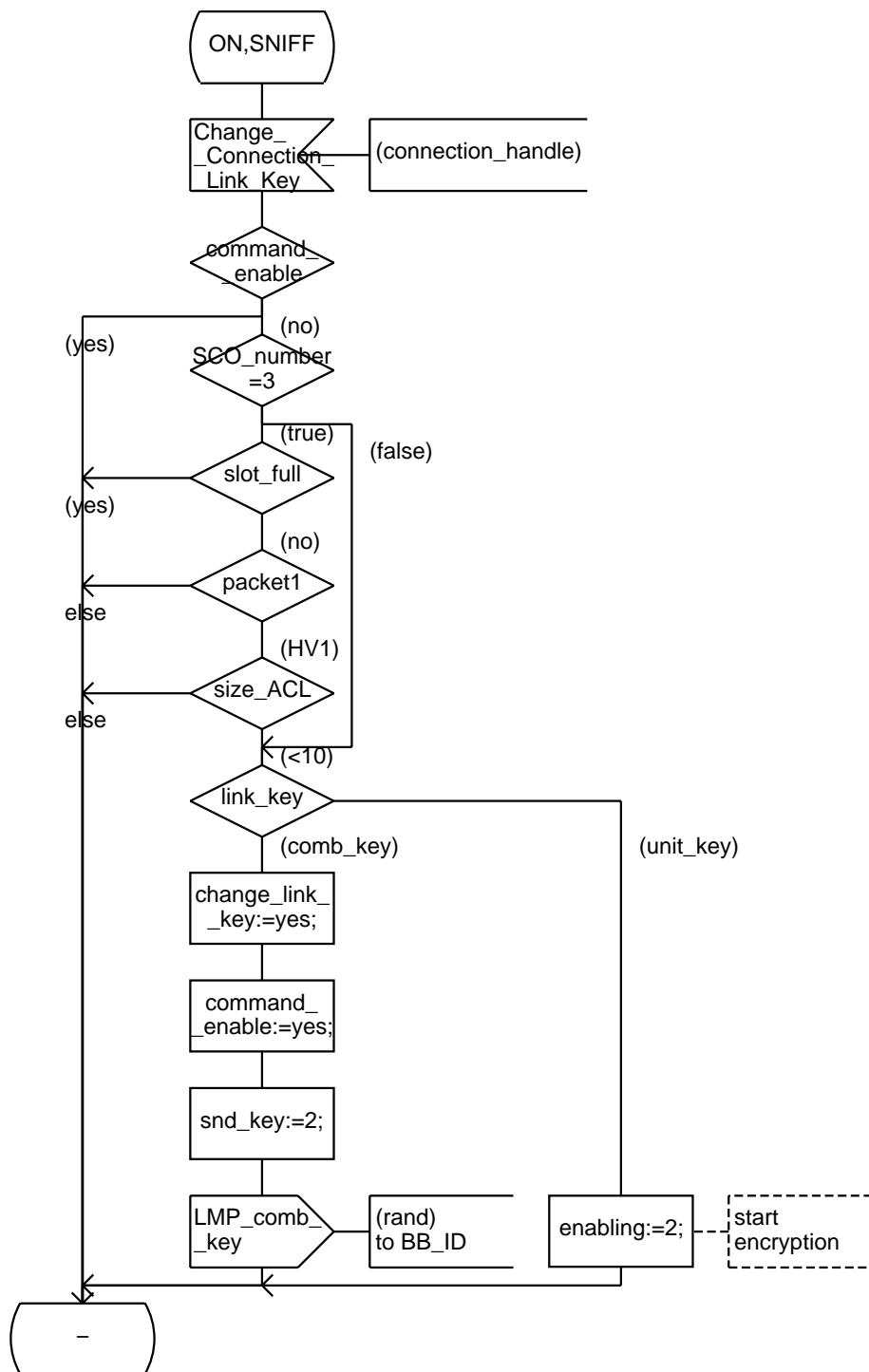


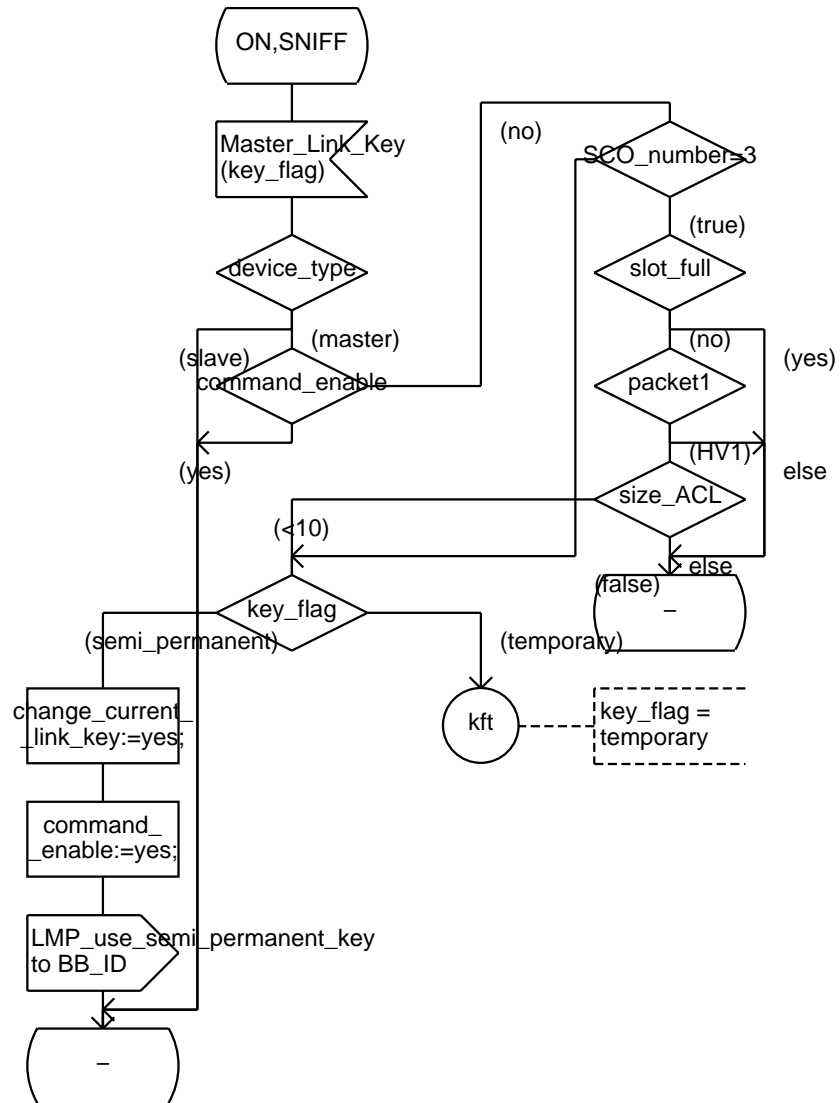


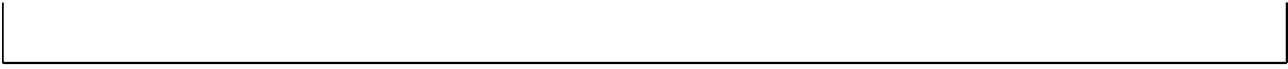


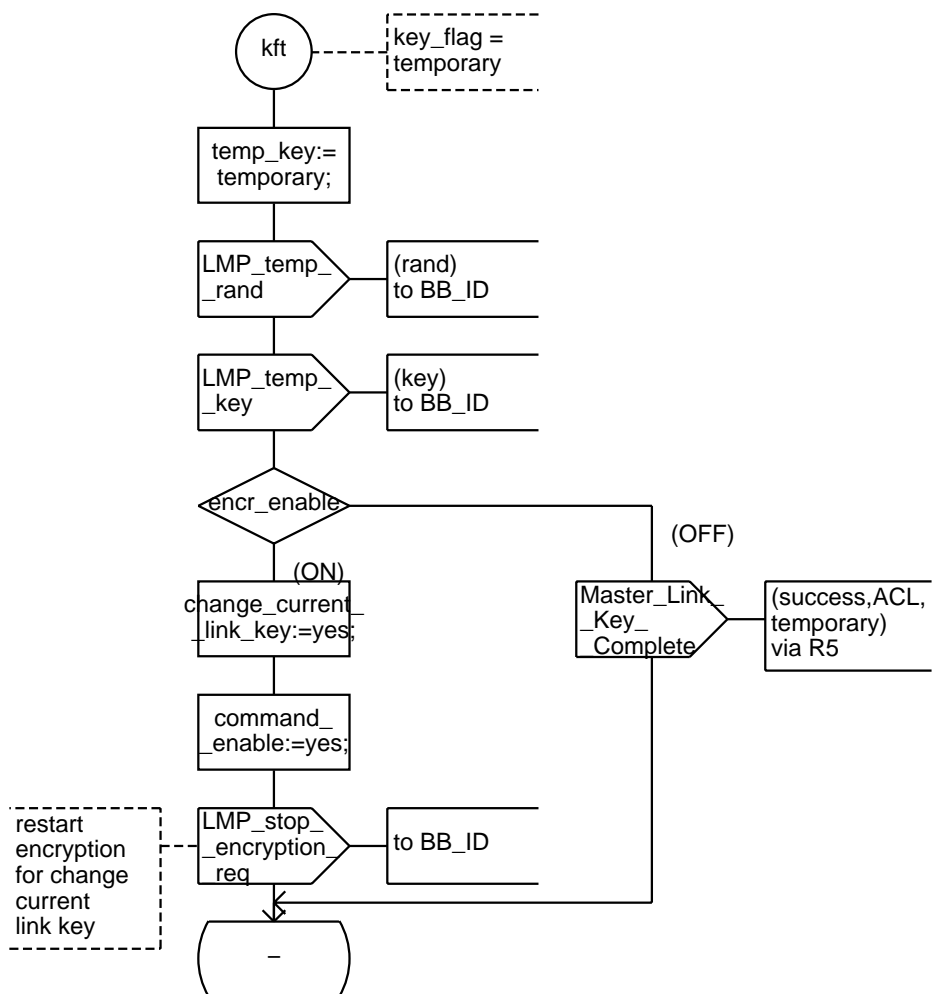


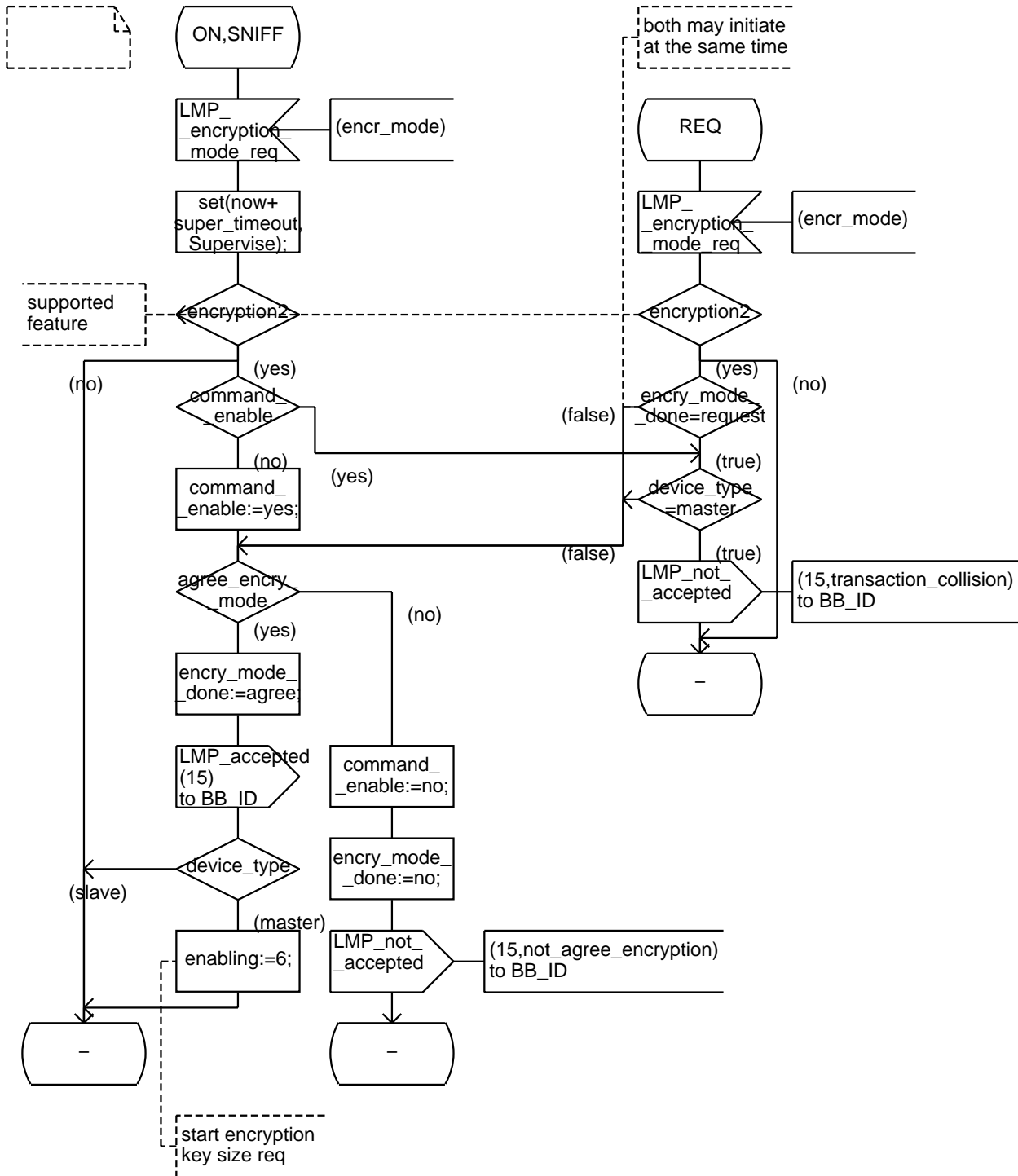


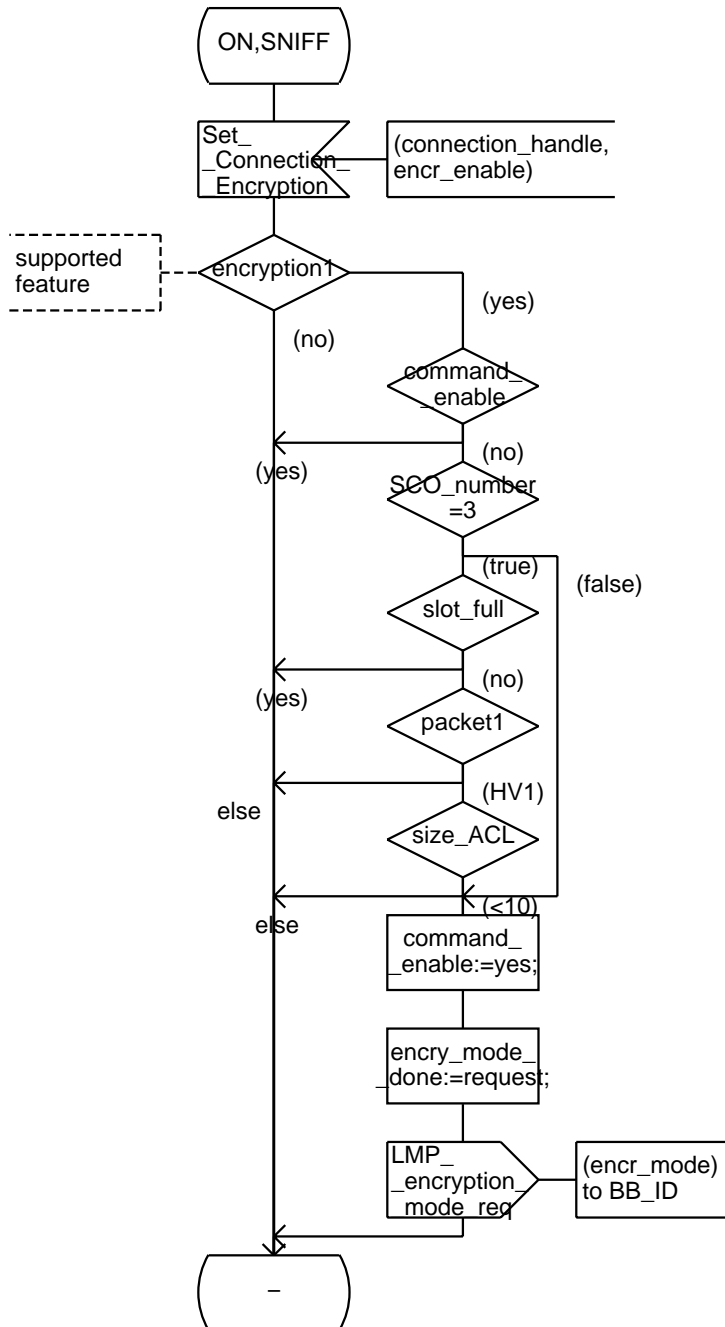


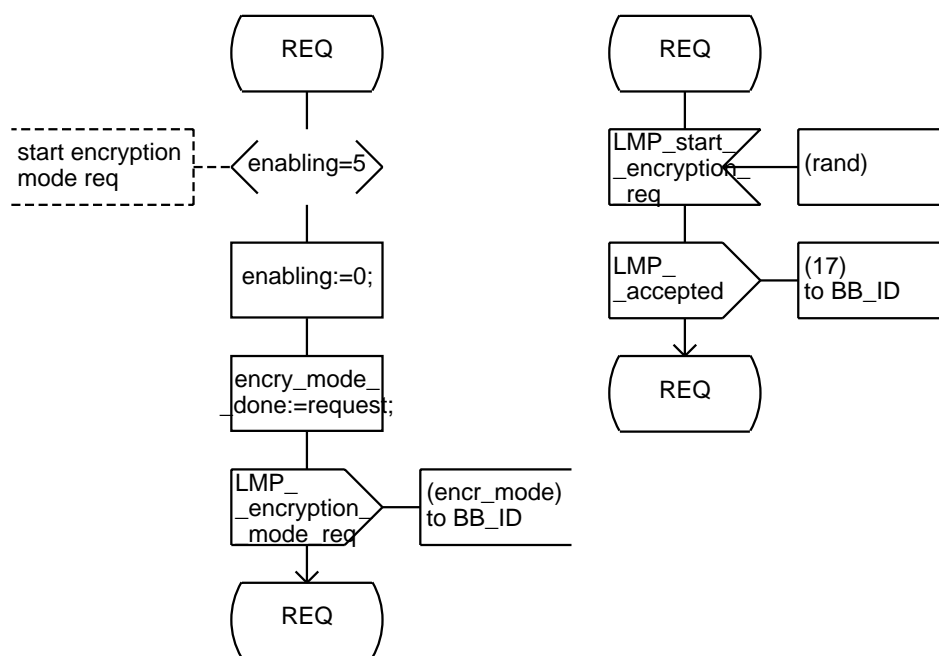


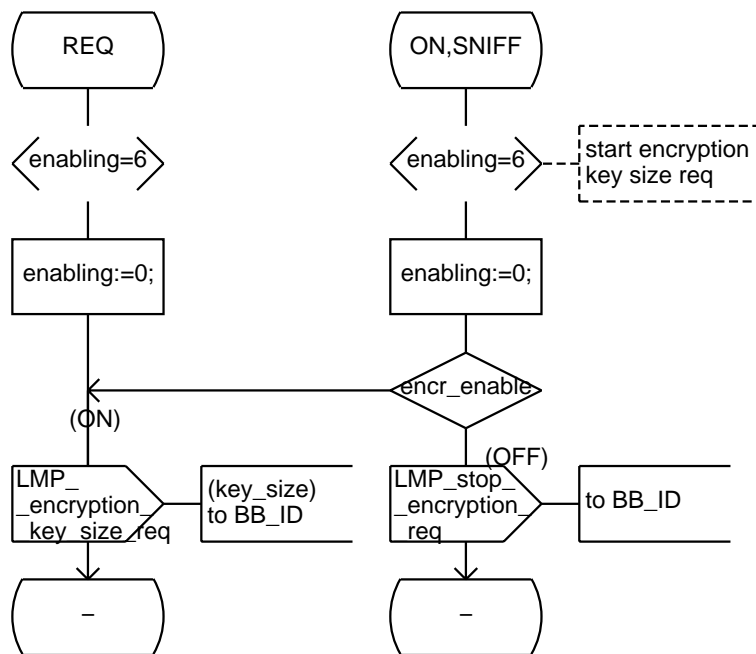


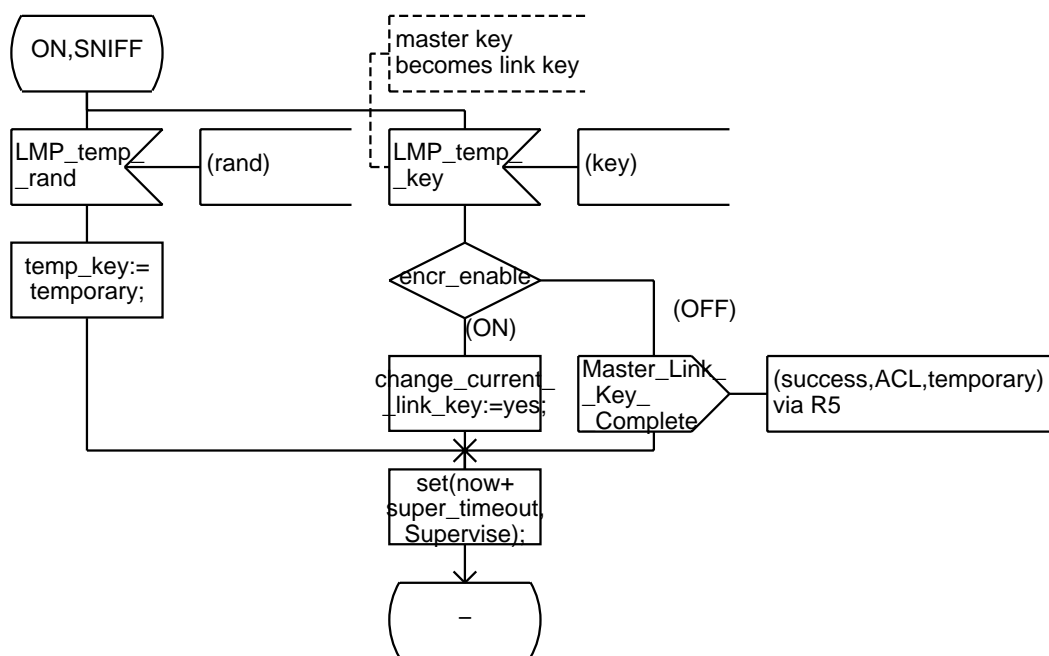


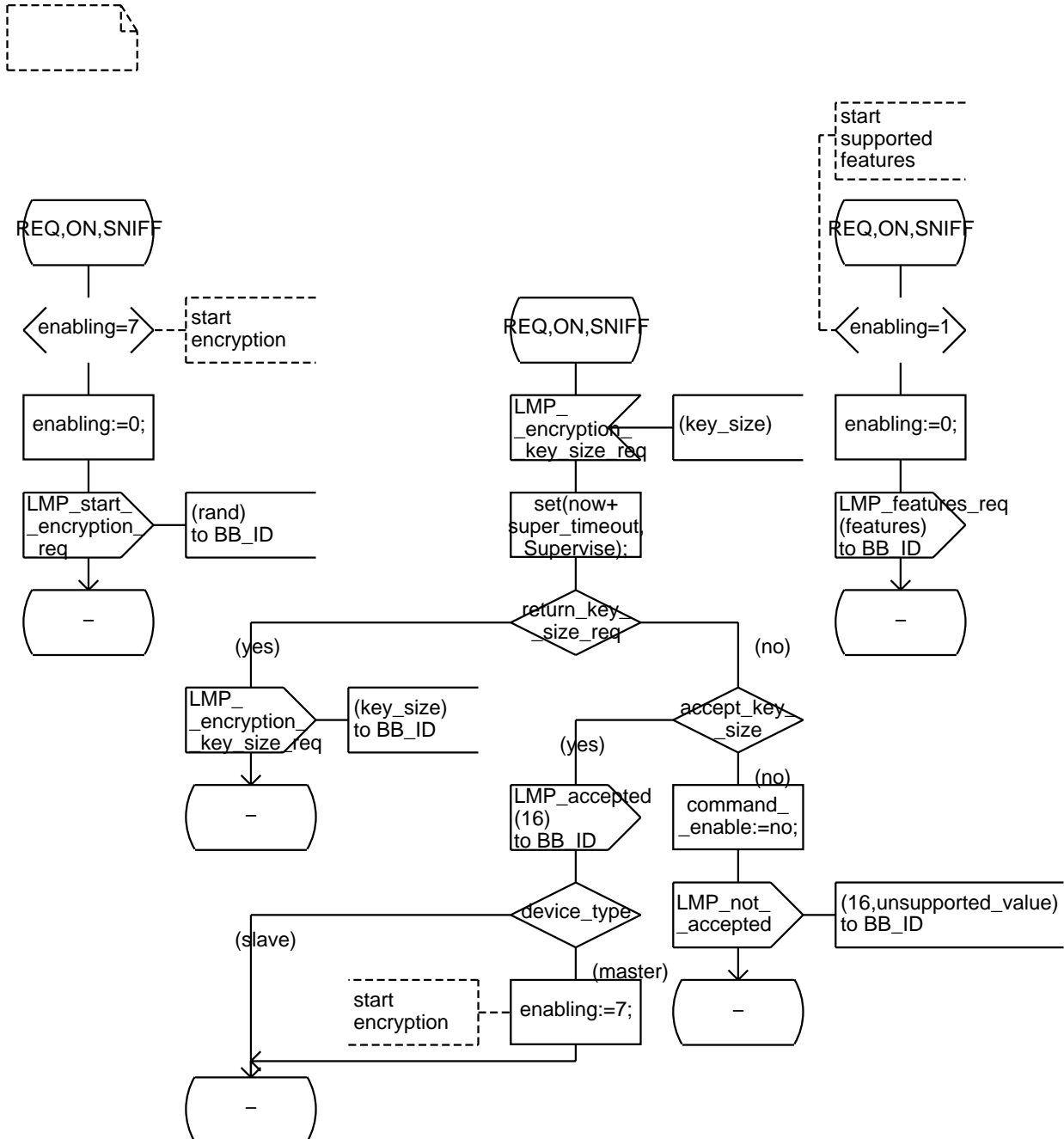


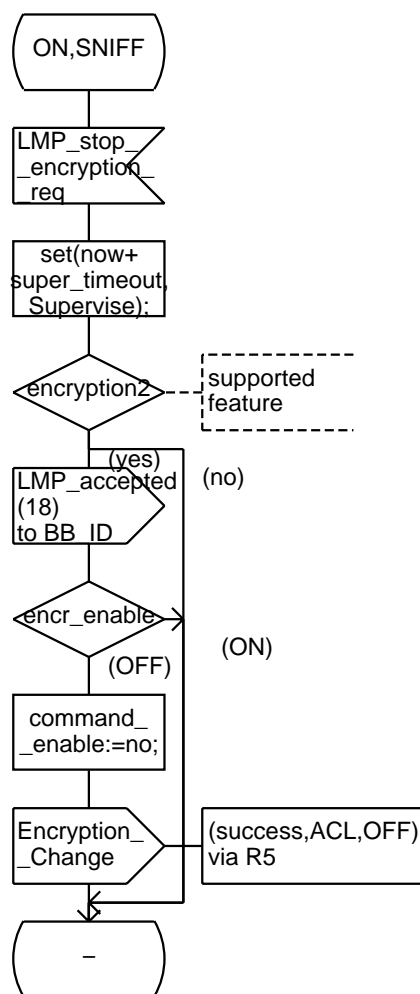
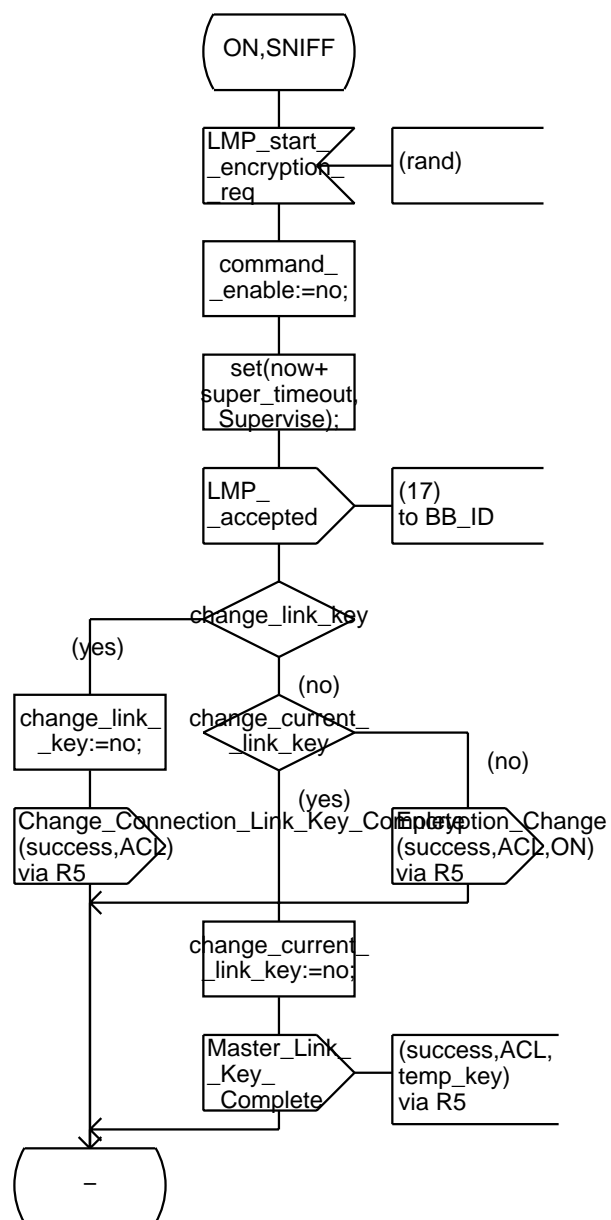


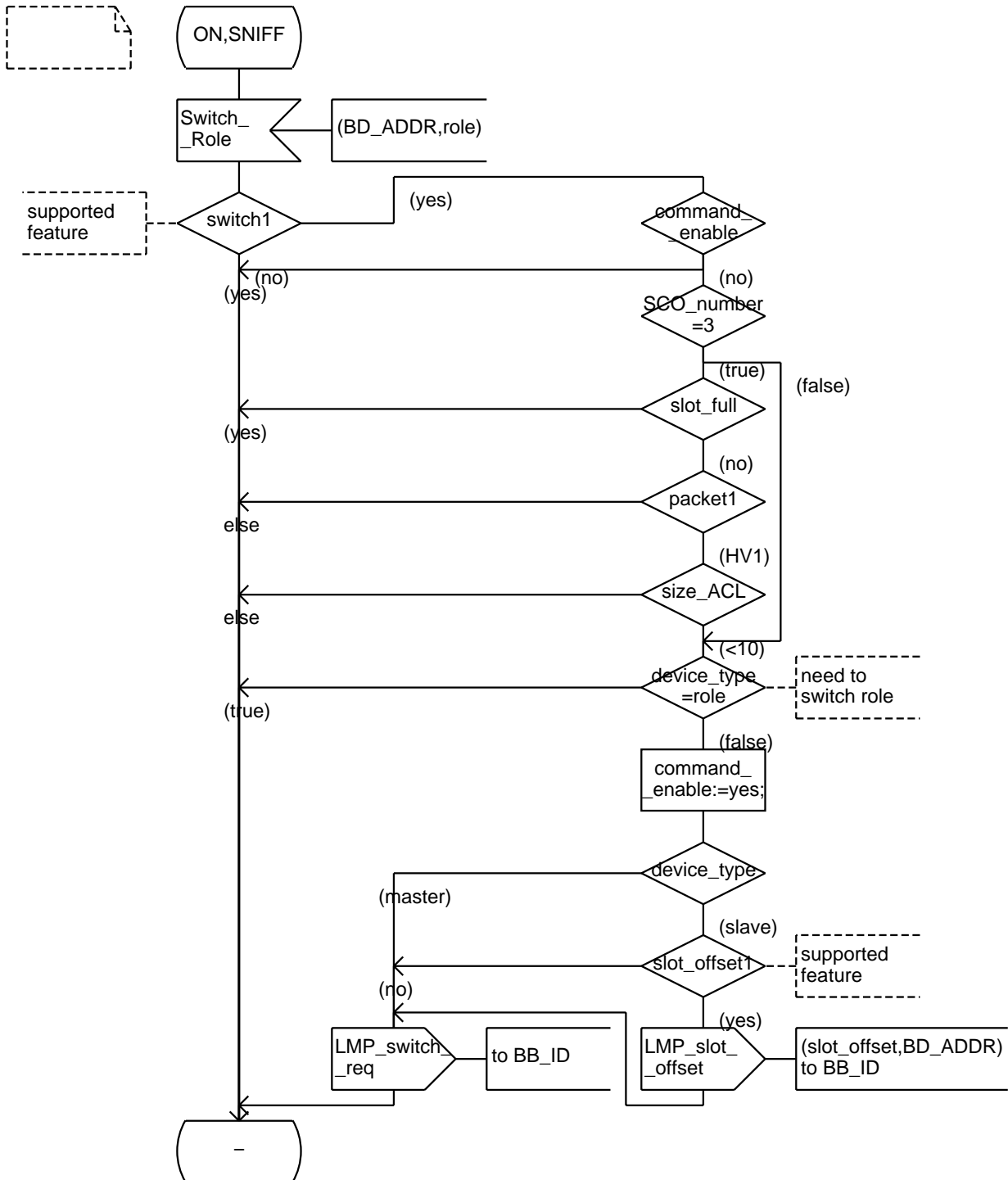


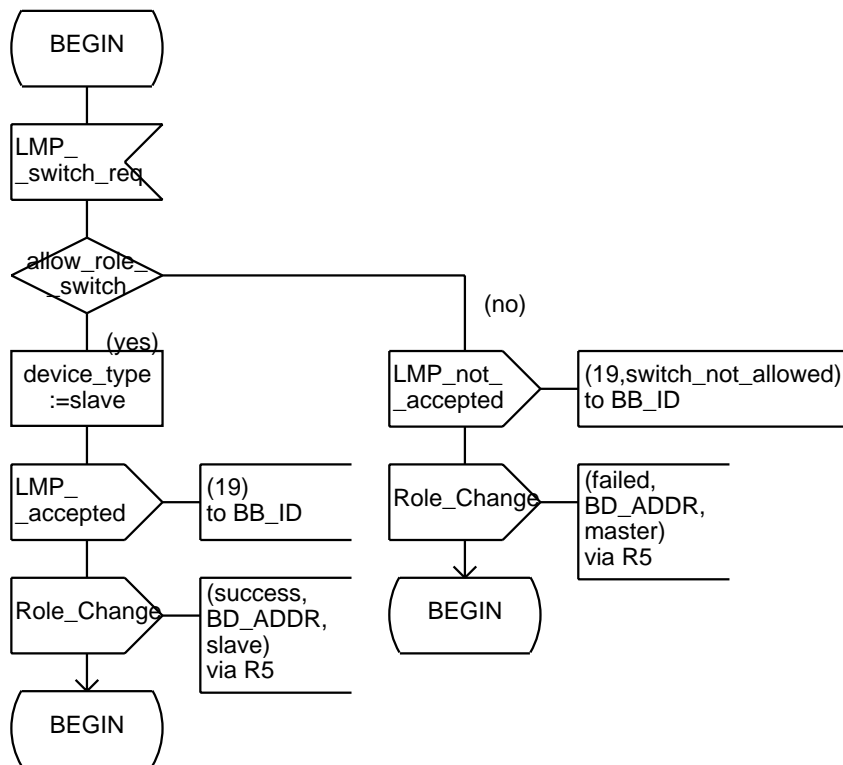


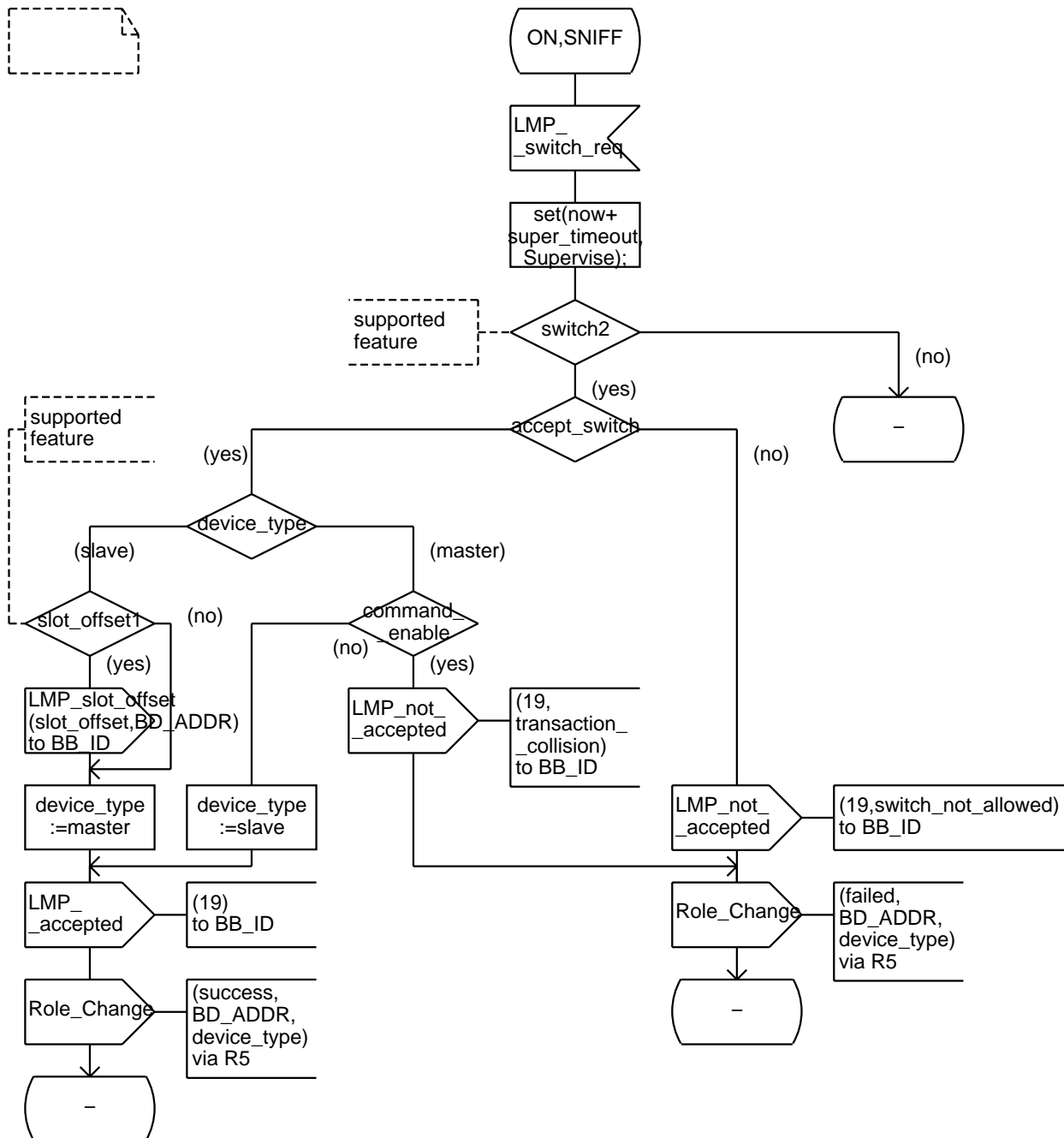


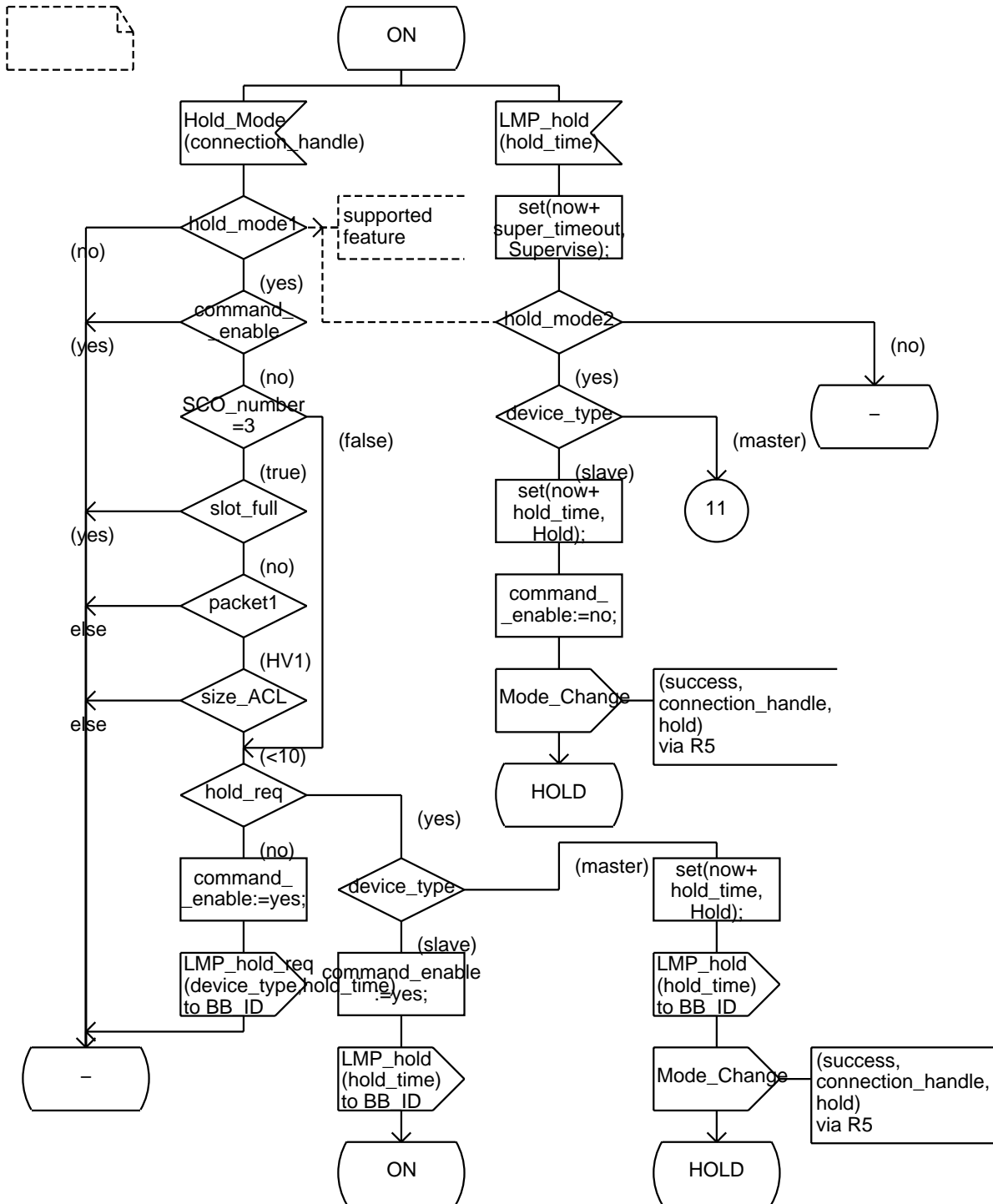


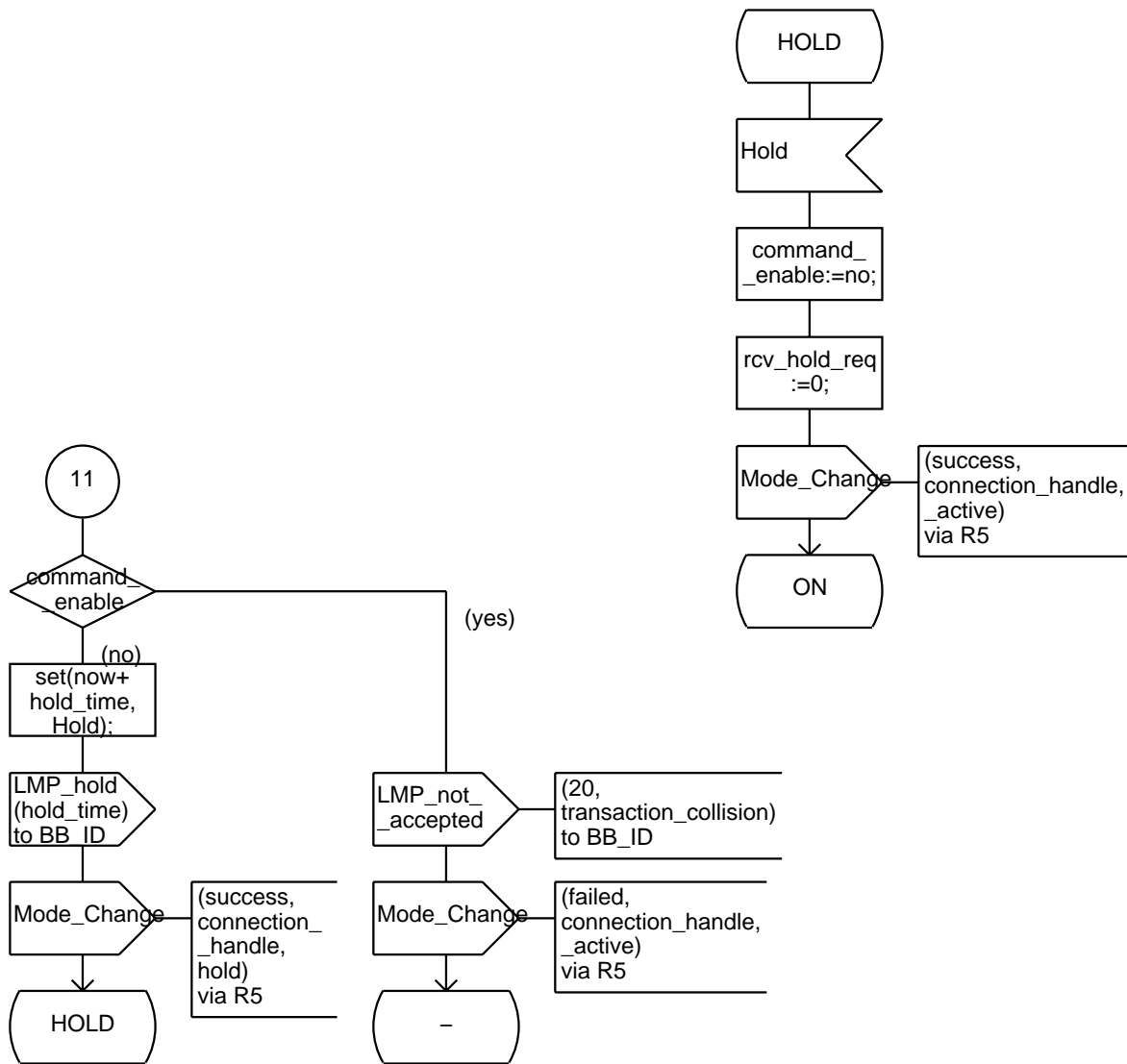


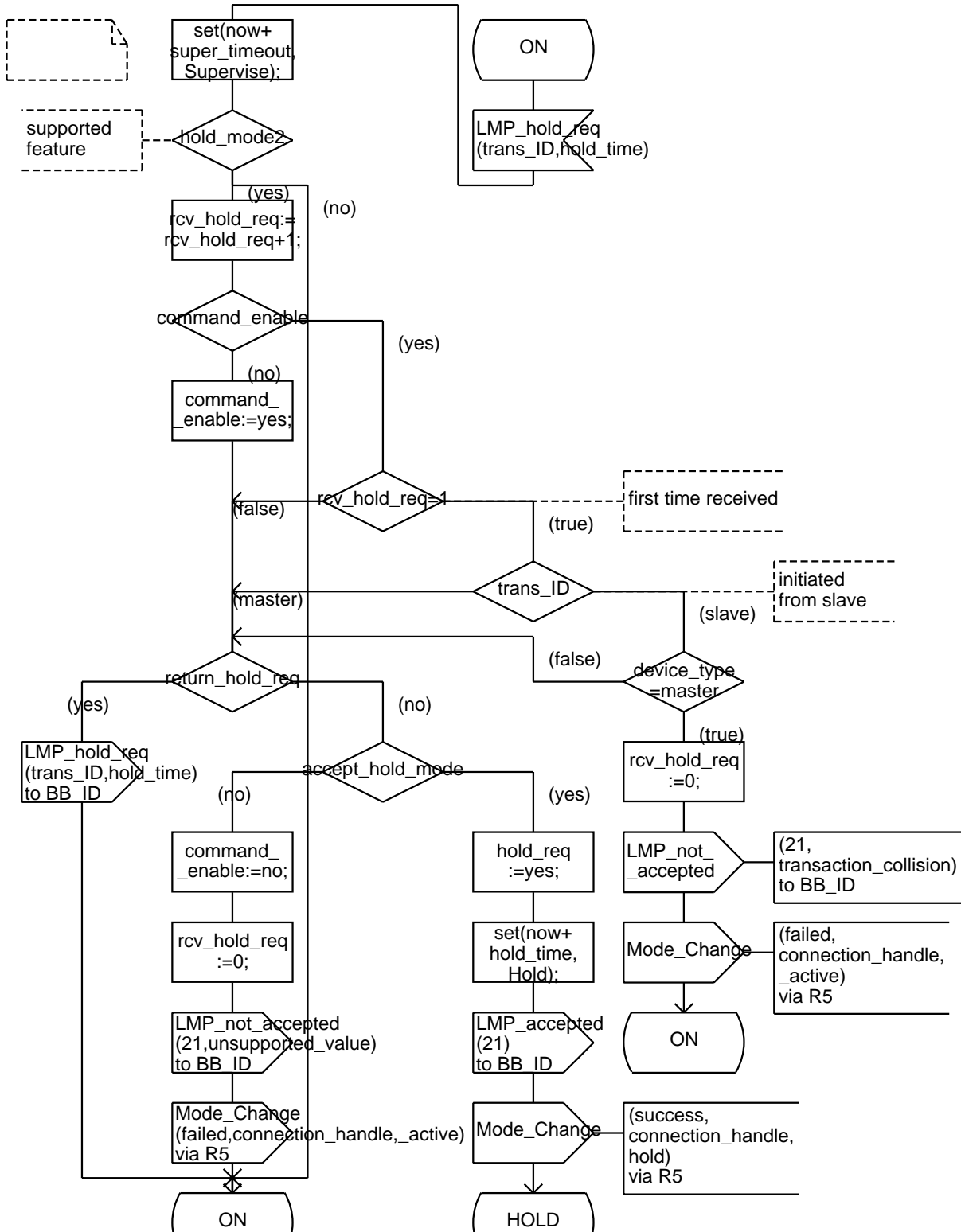


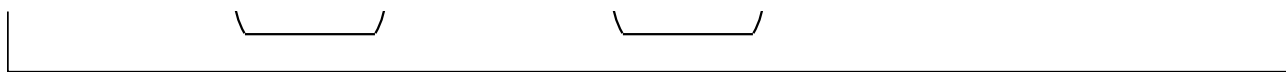


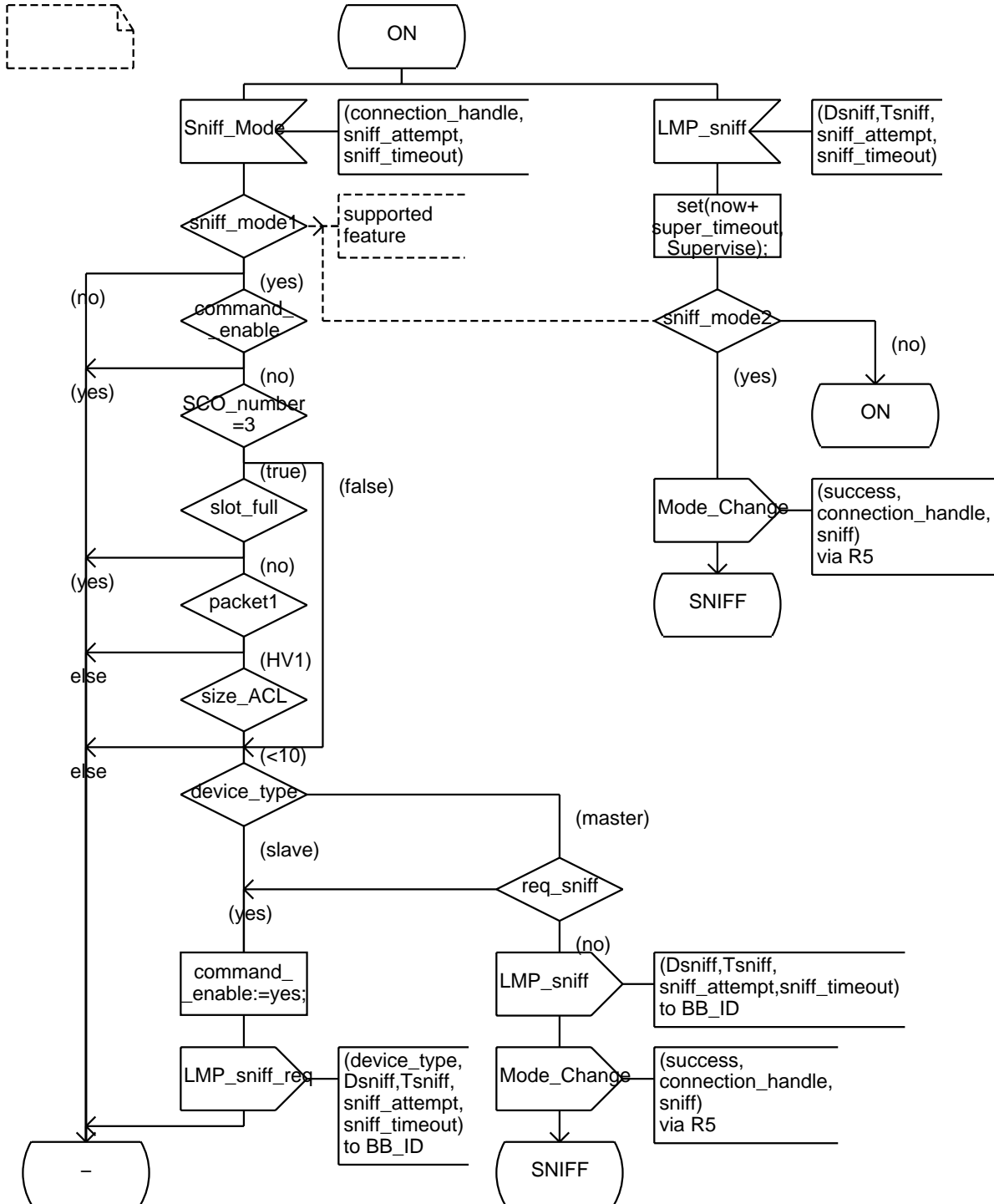


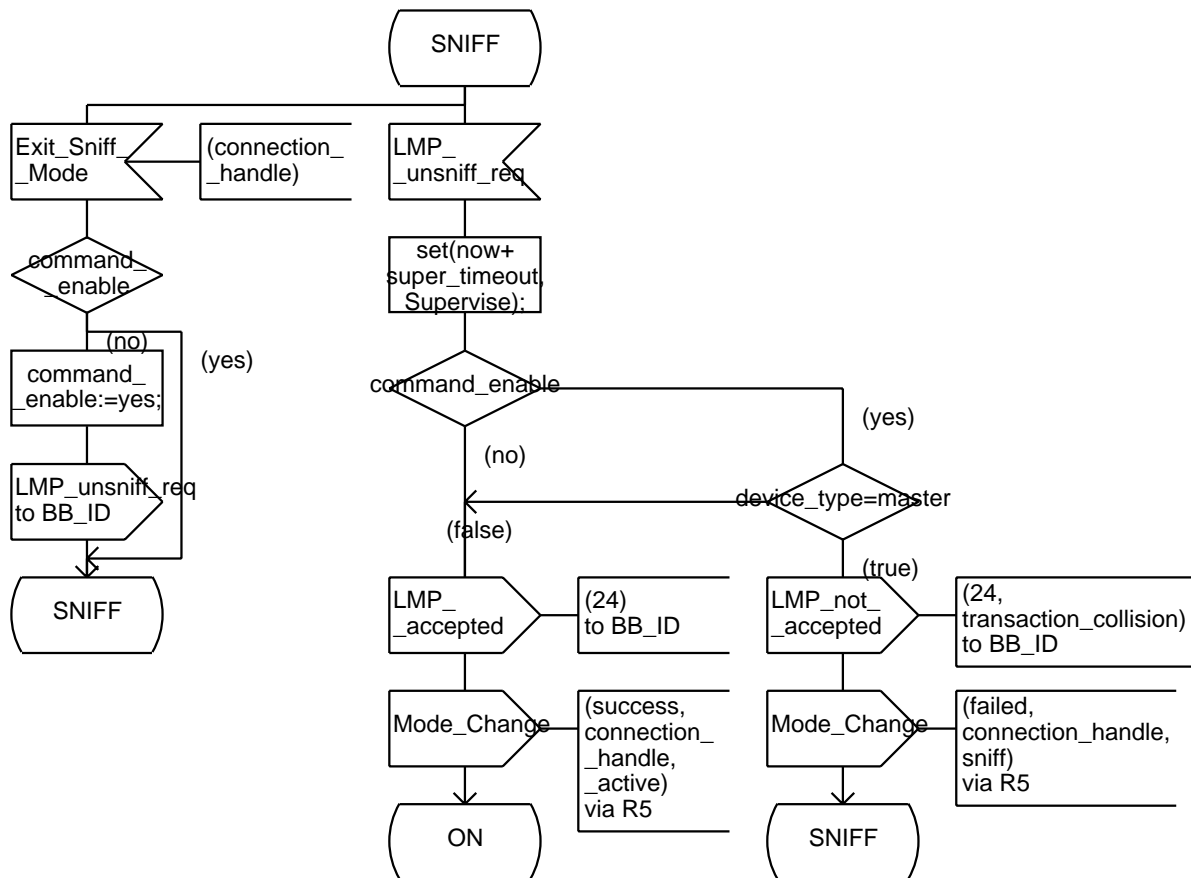


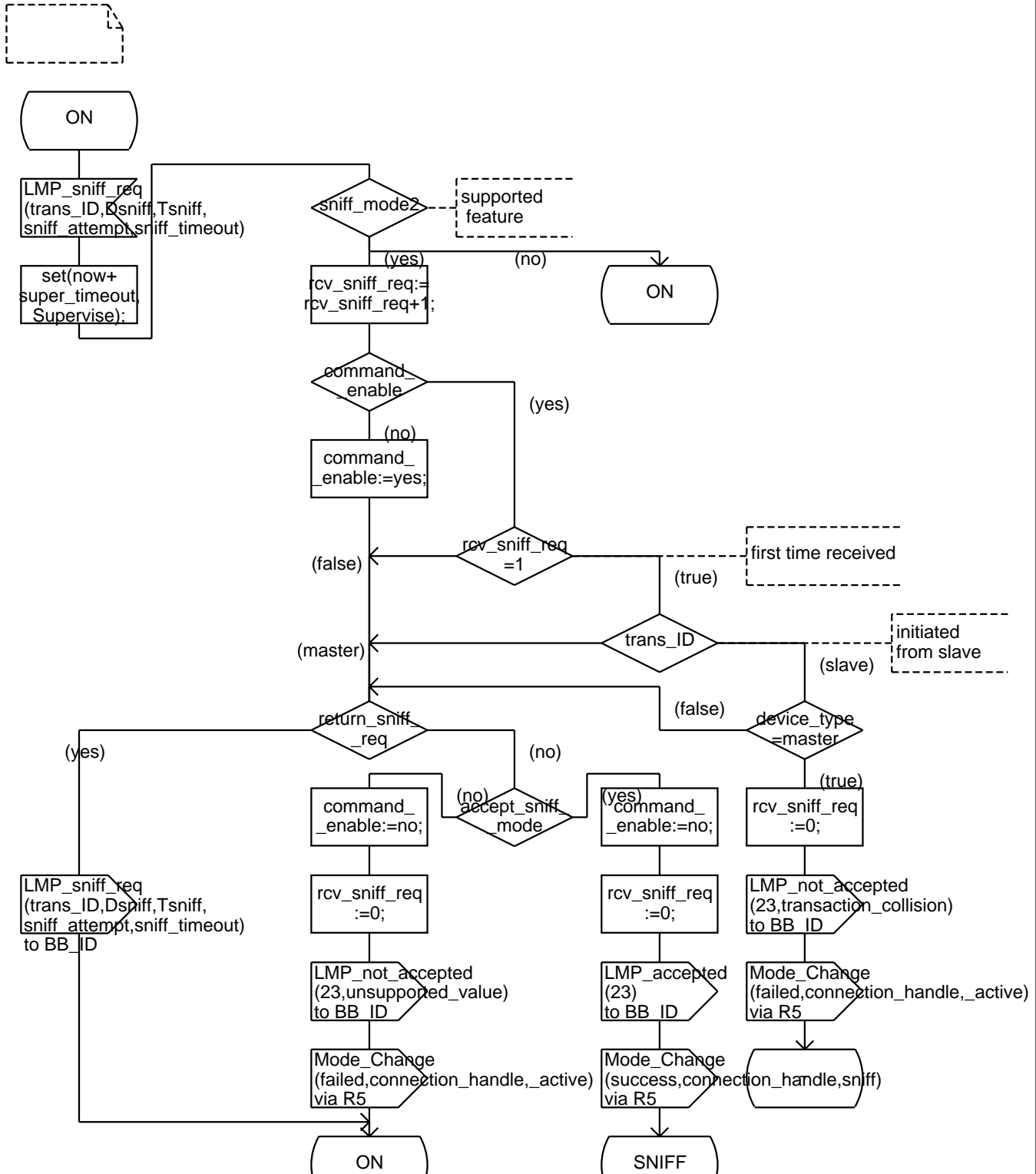




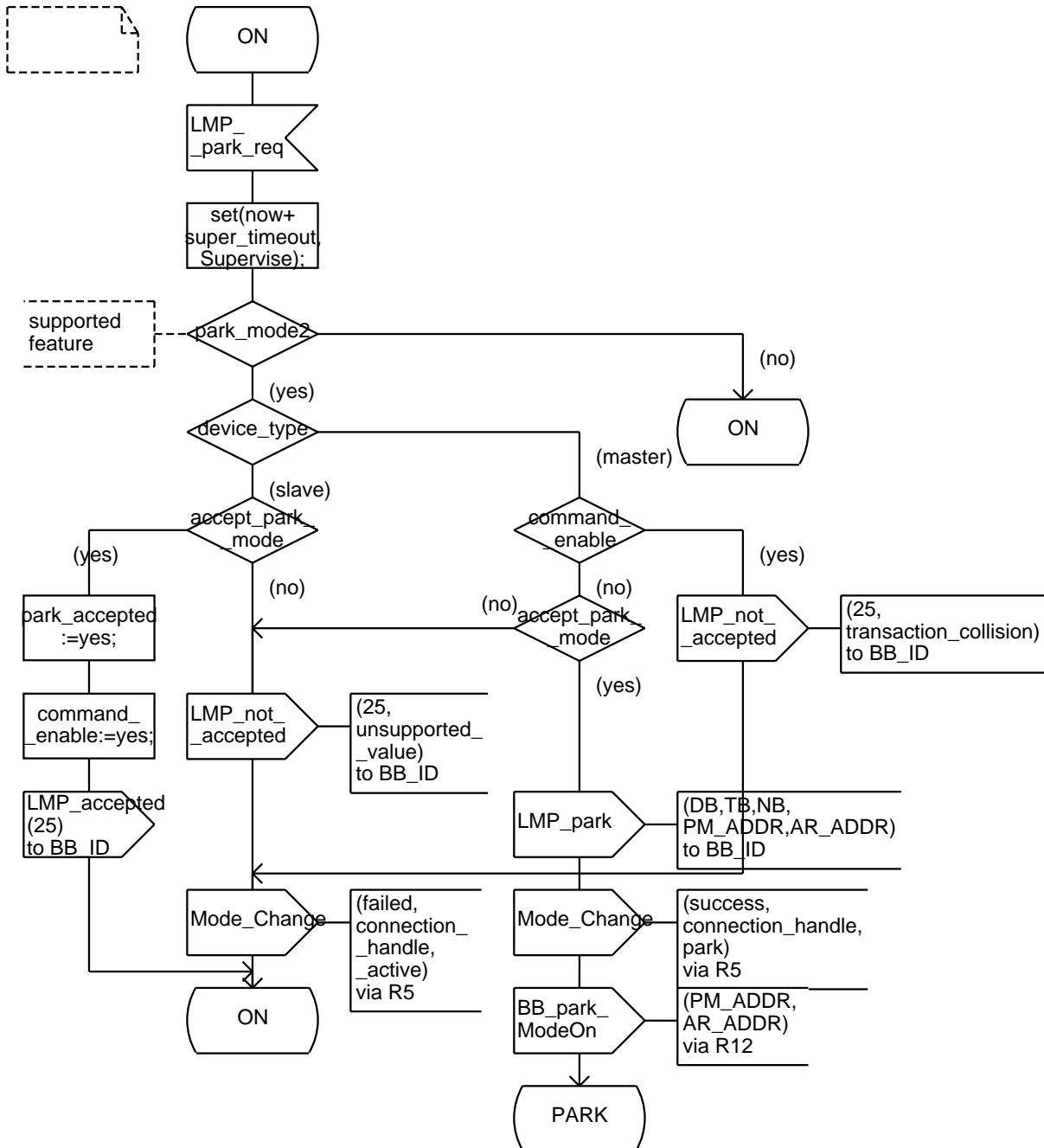


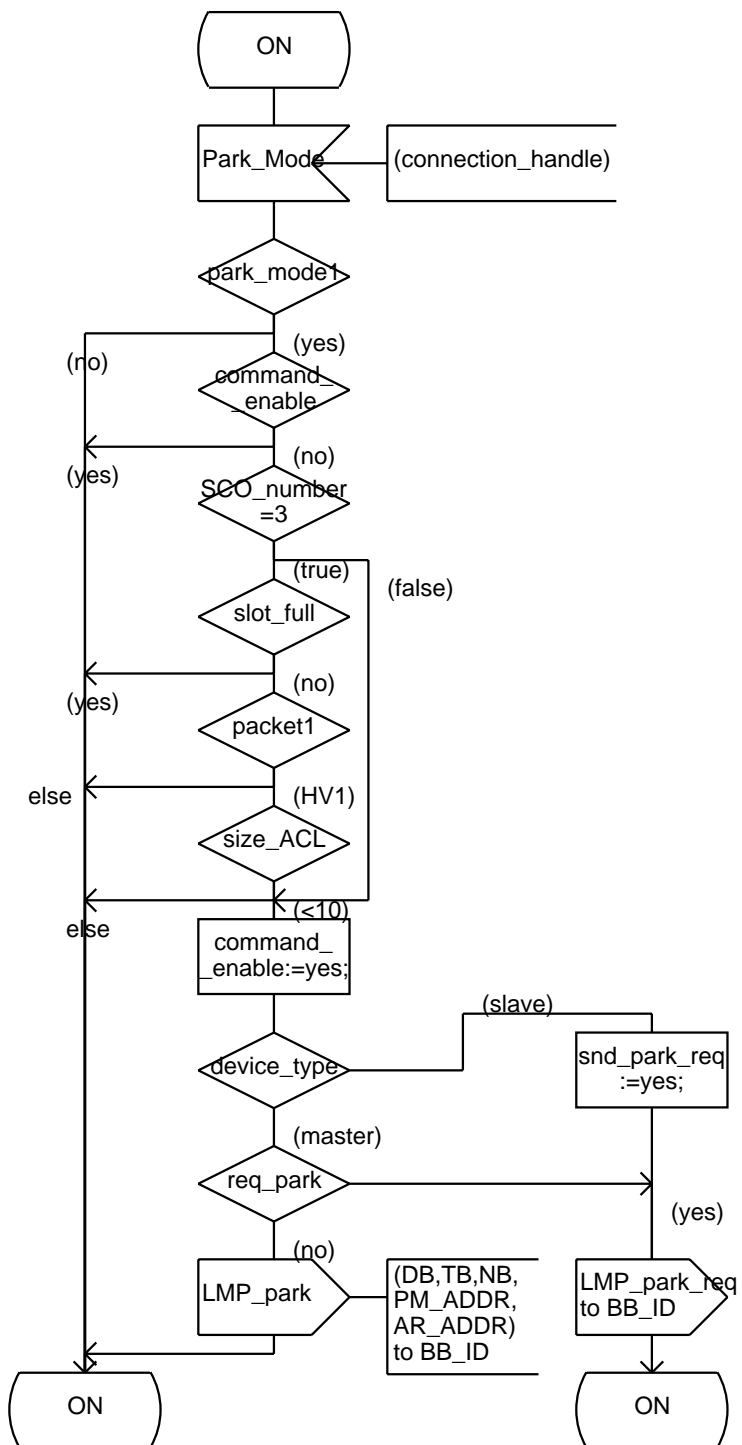






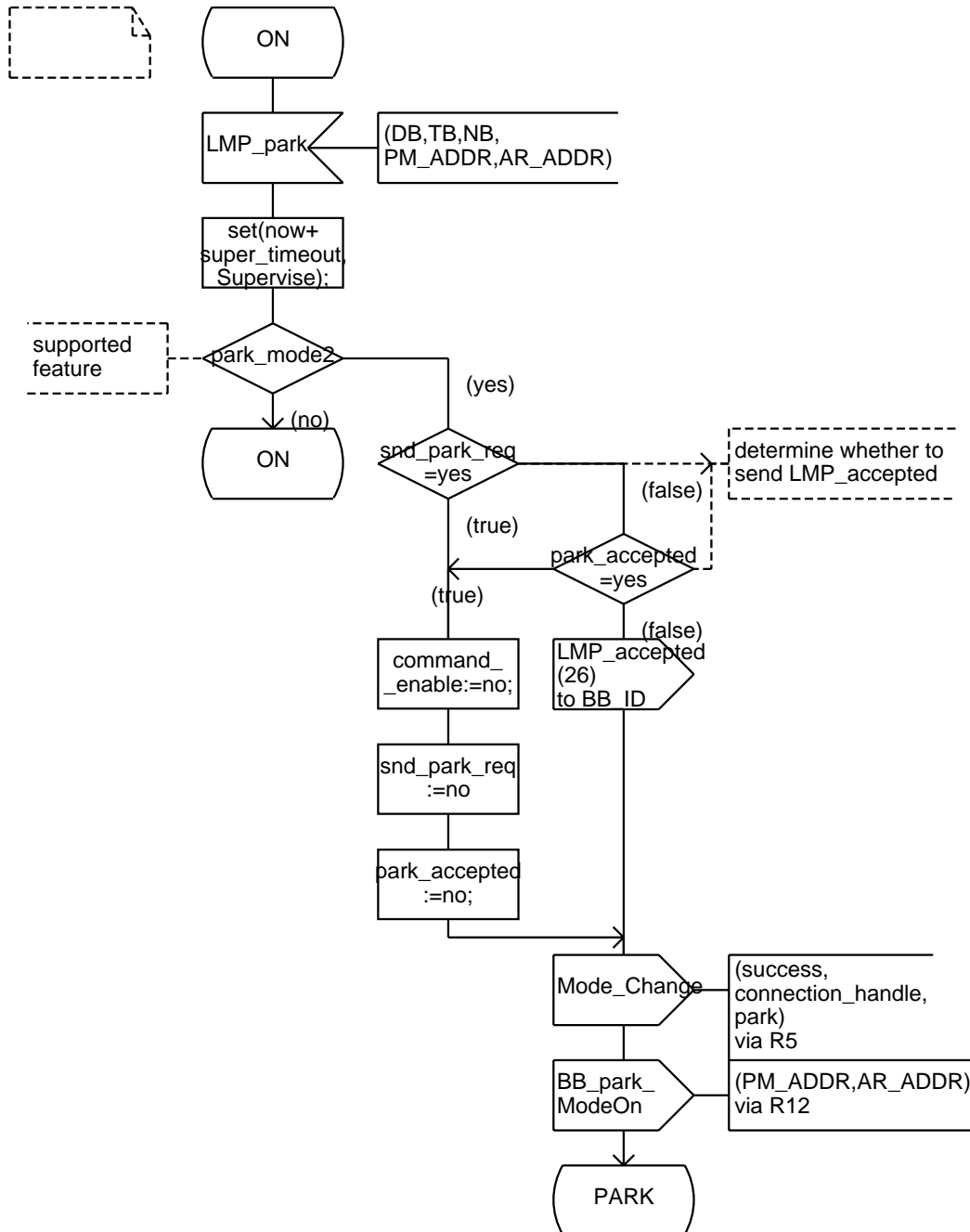


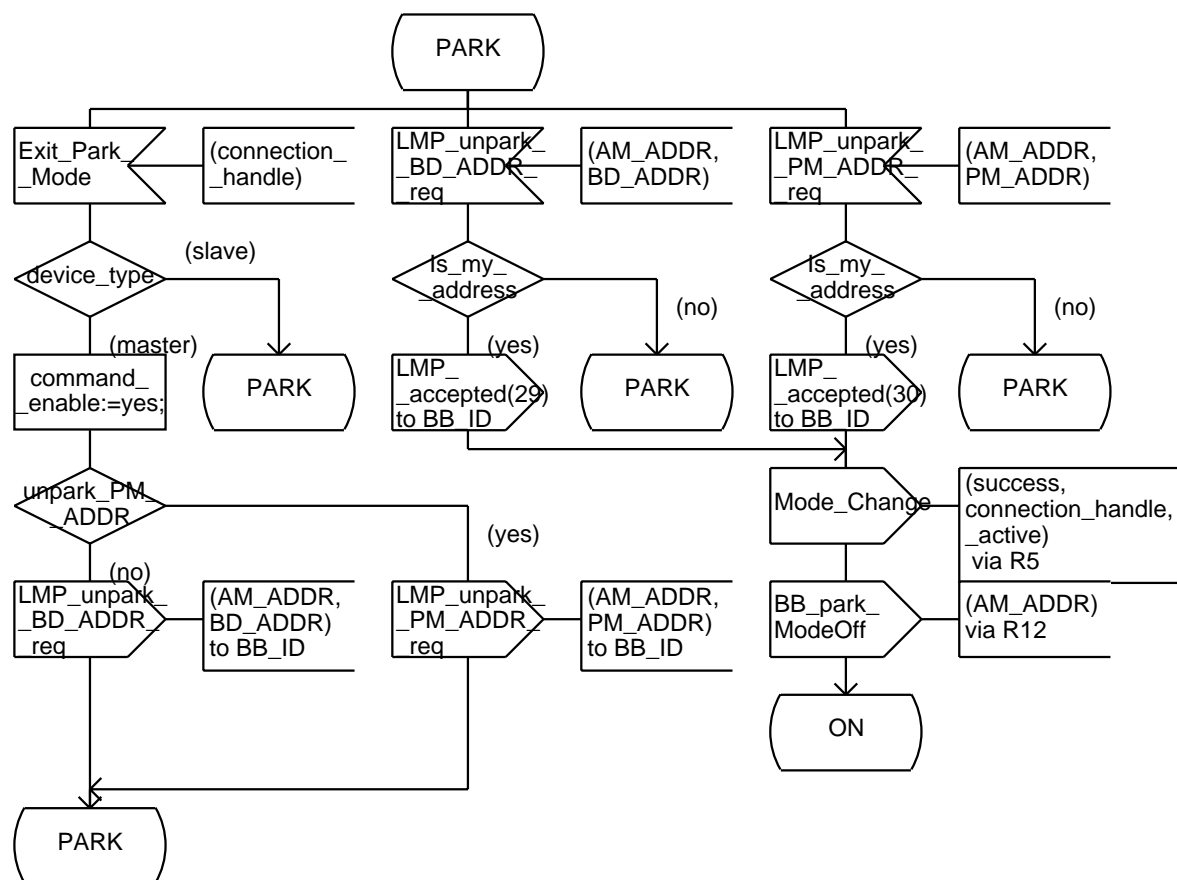


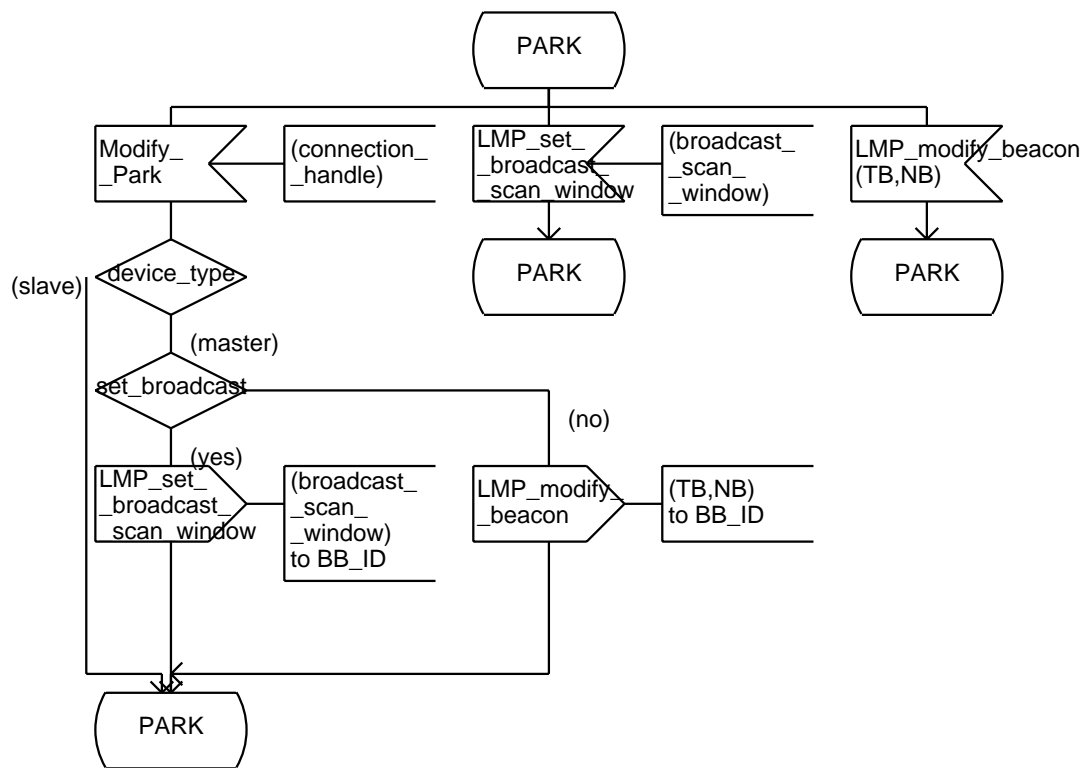


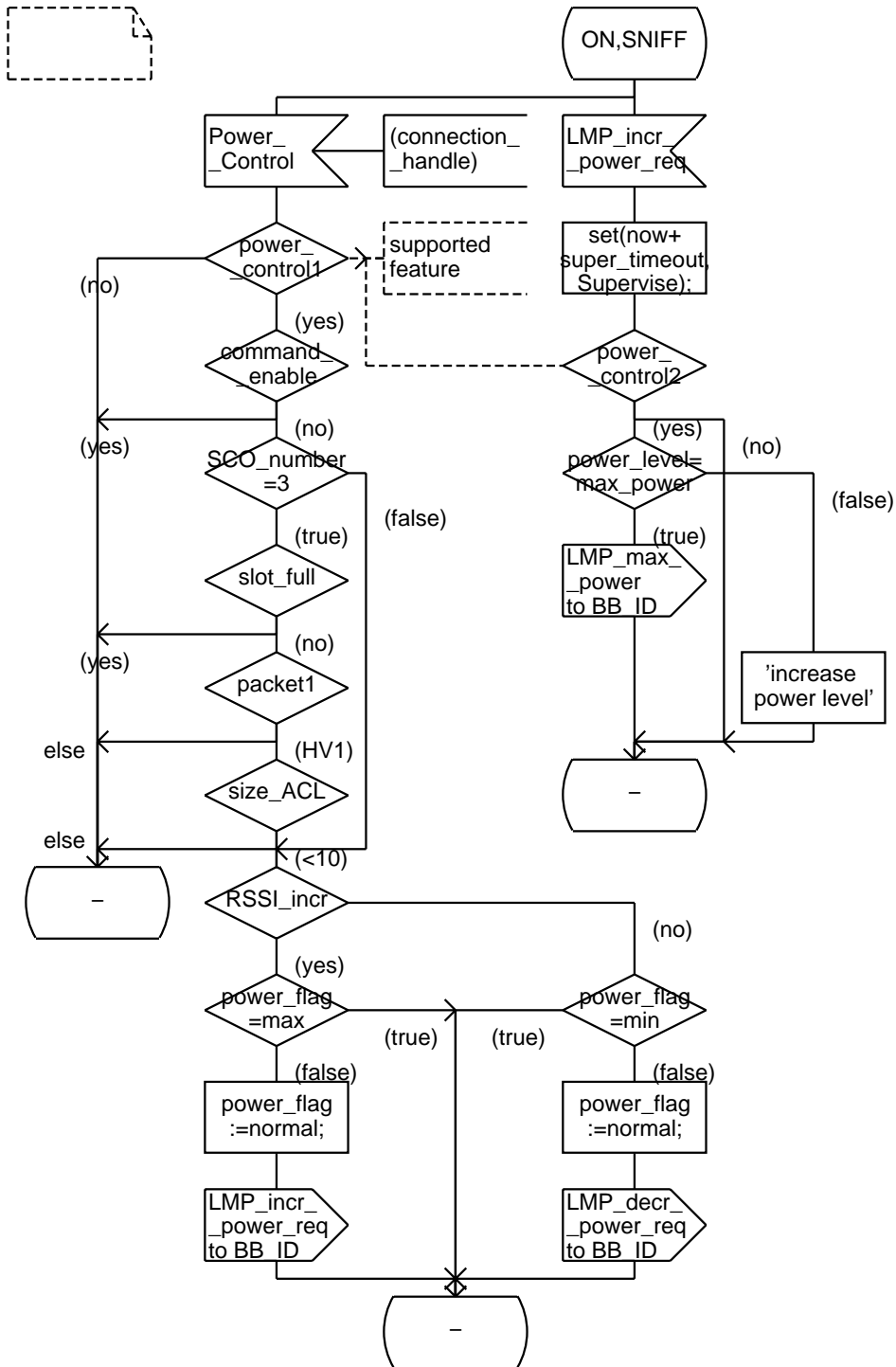
process lmp_p2

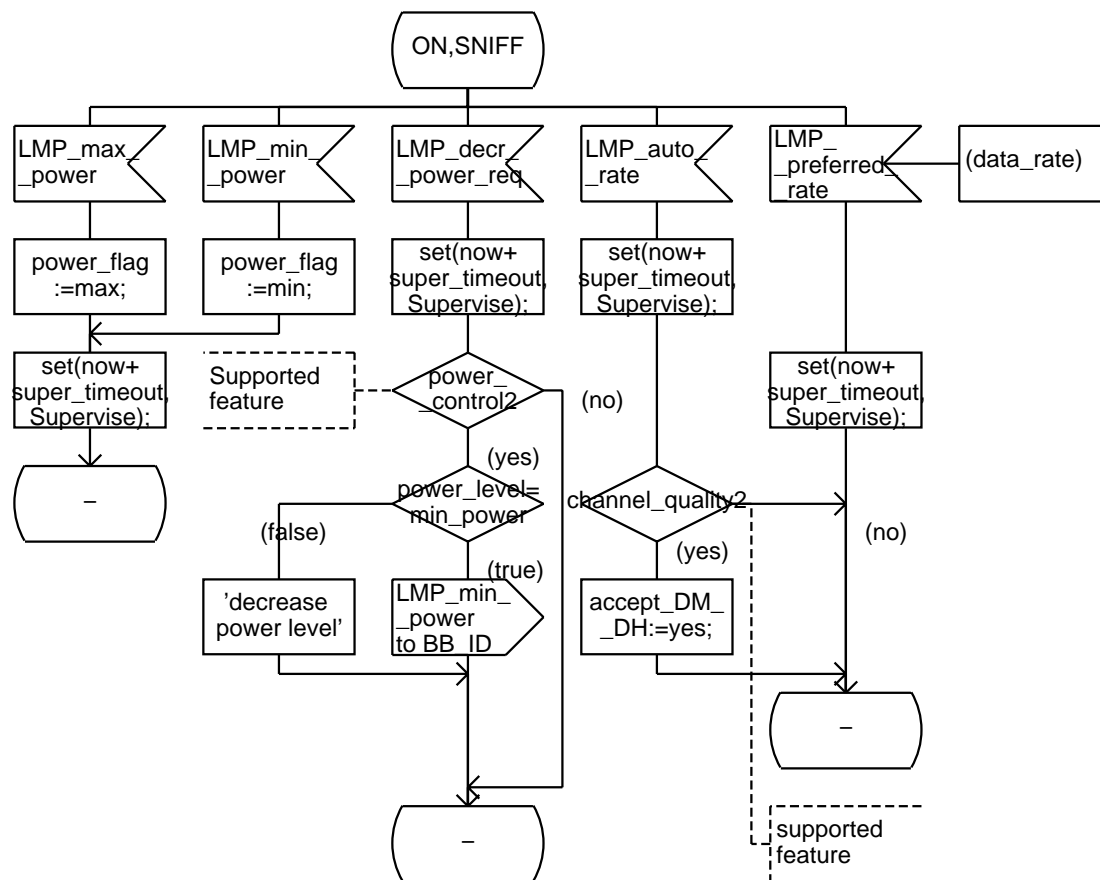
52(74)

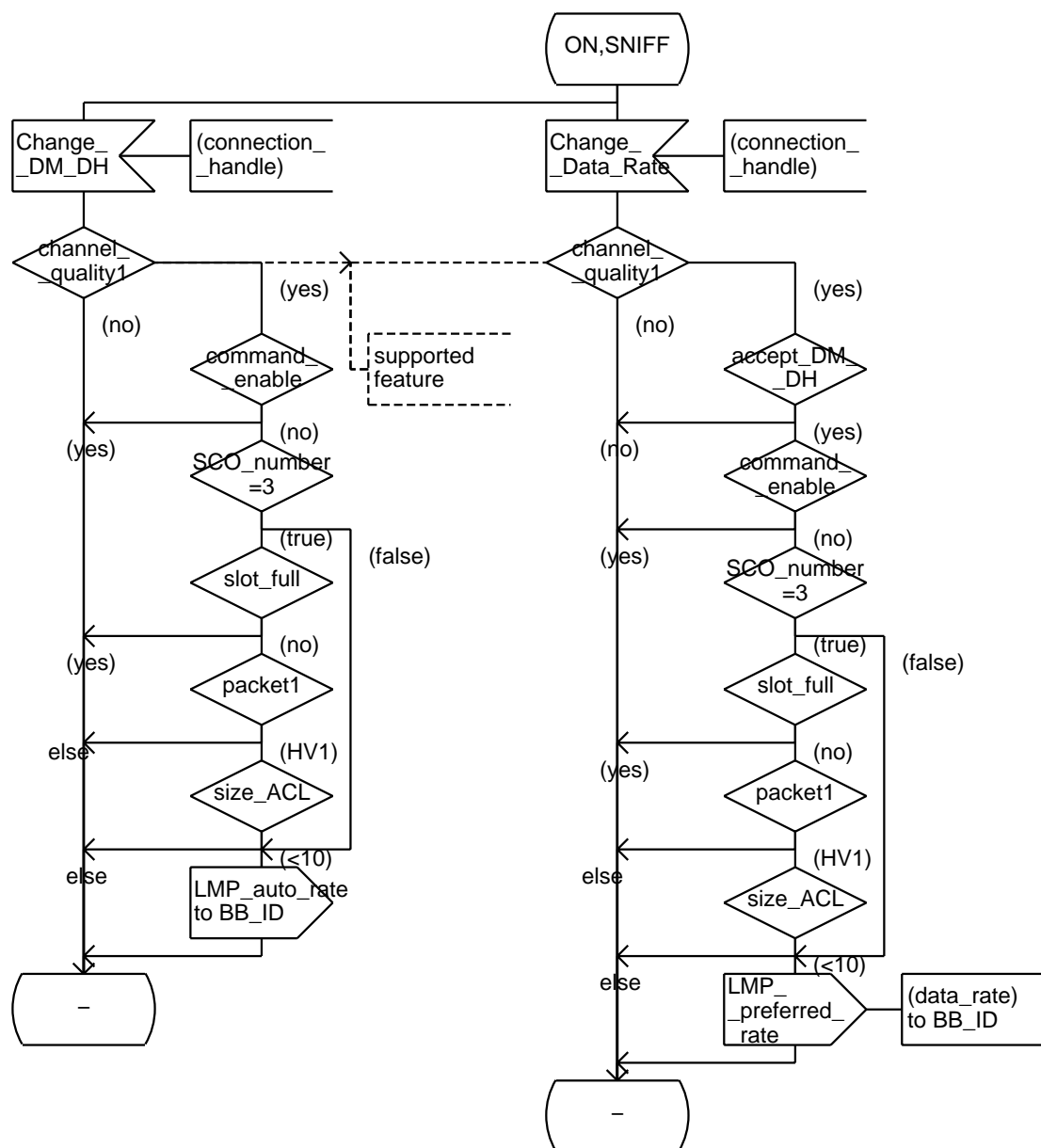


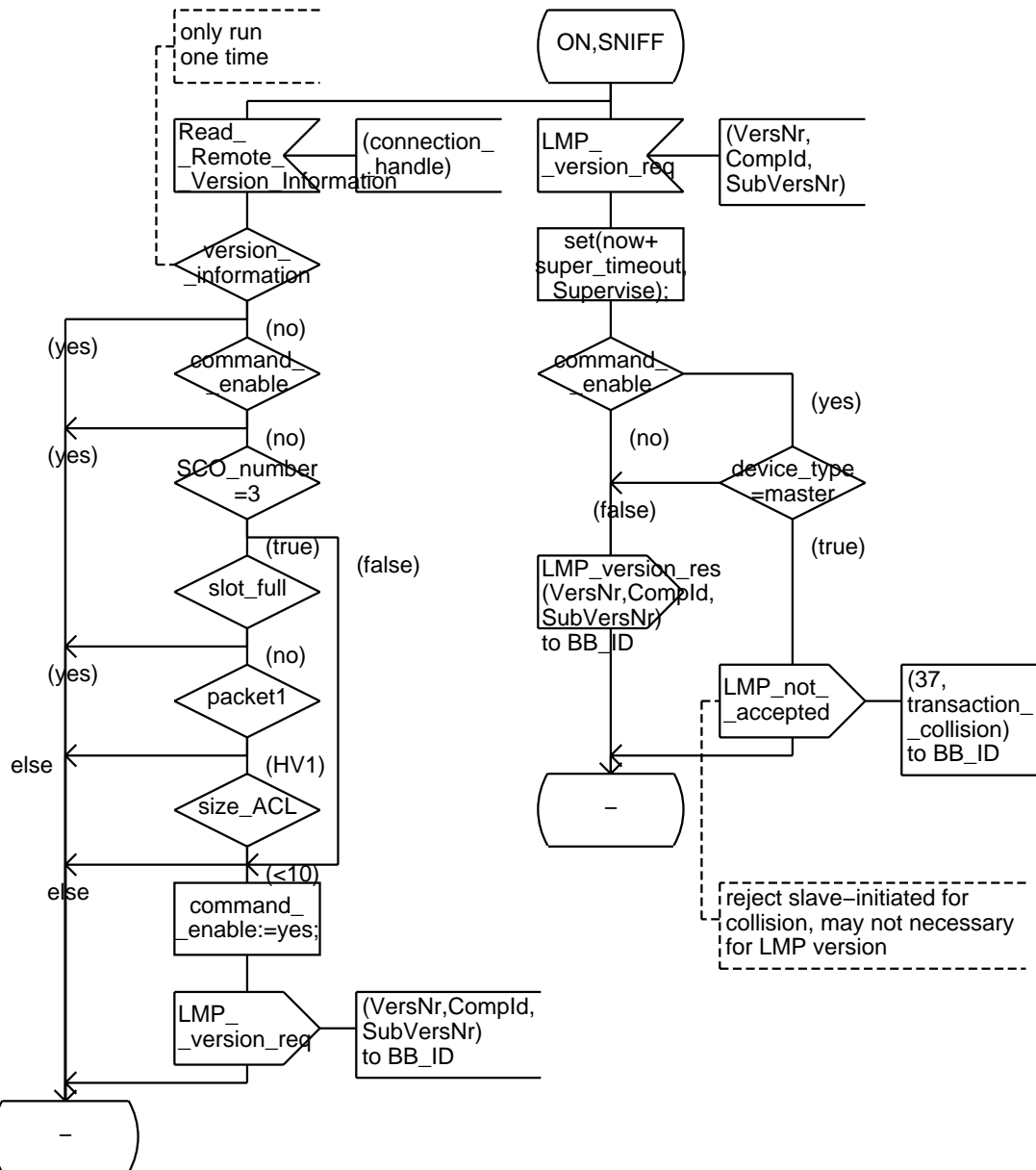


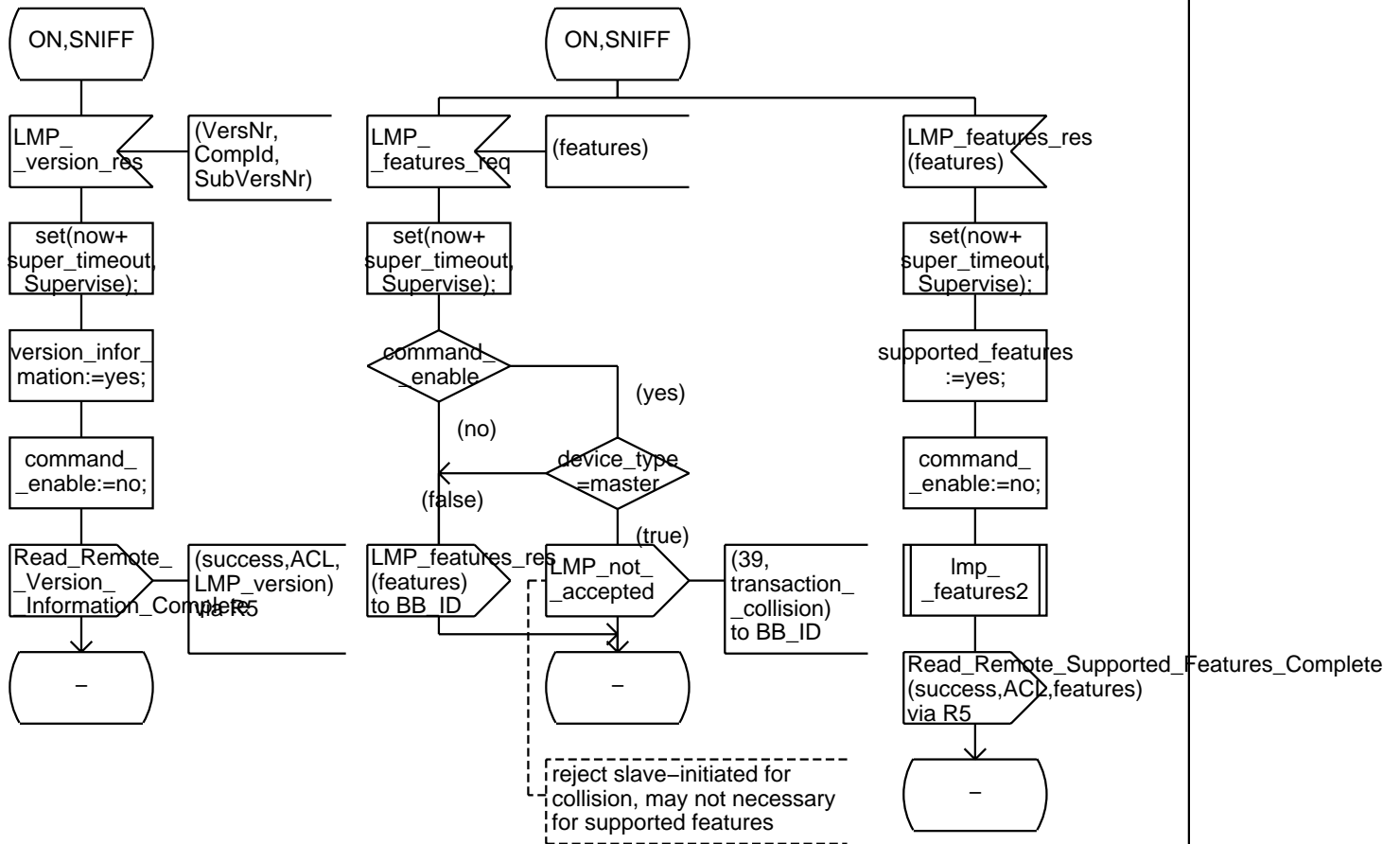


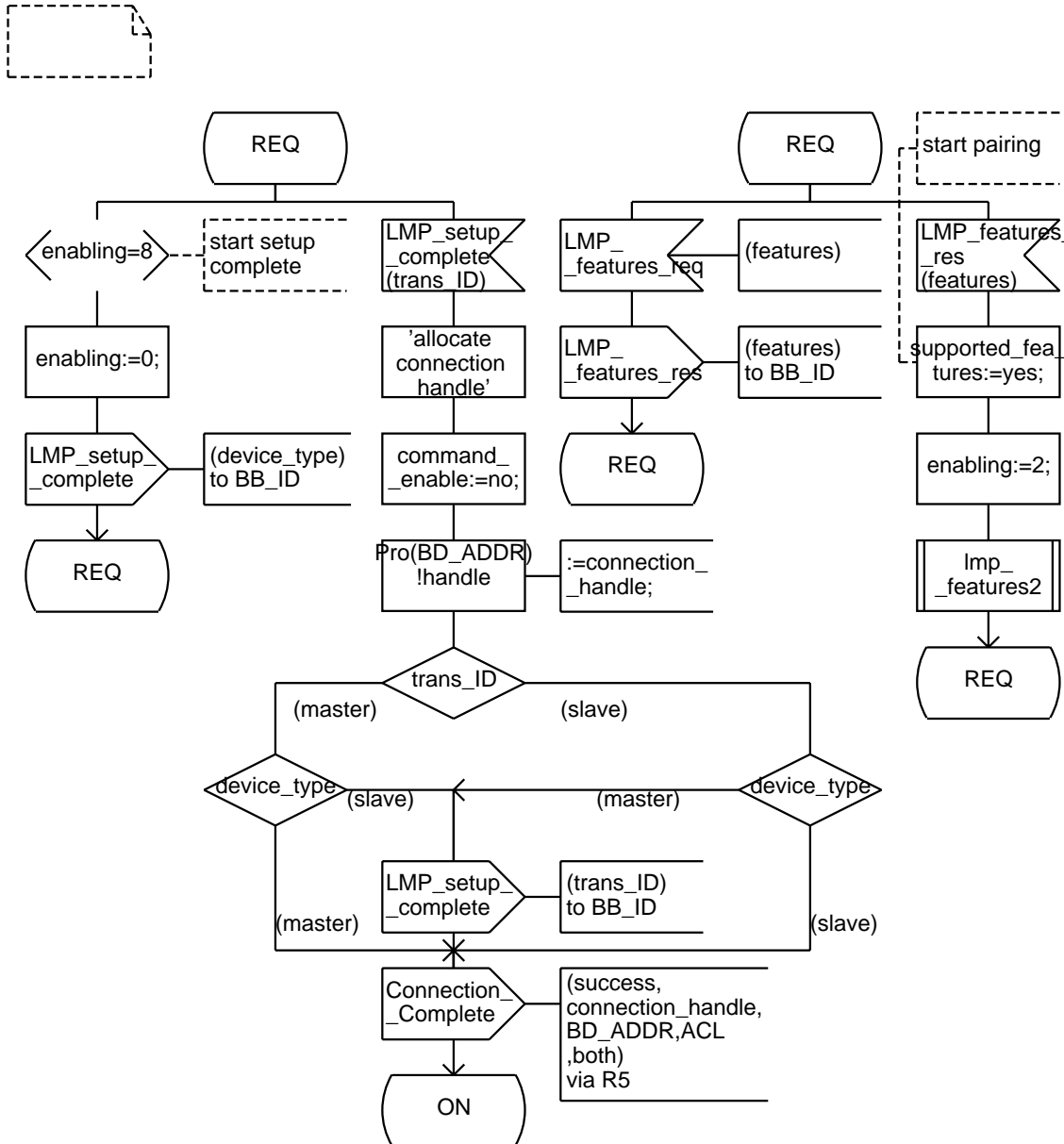


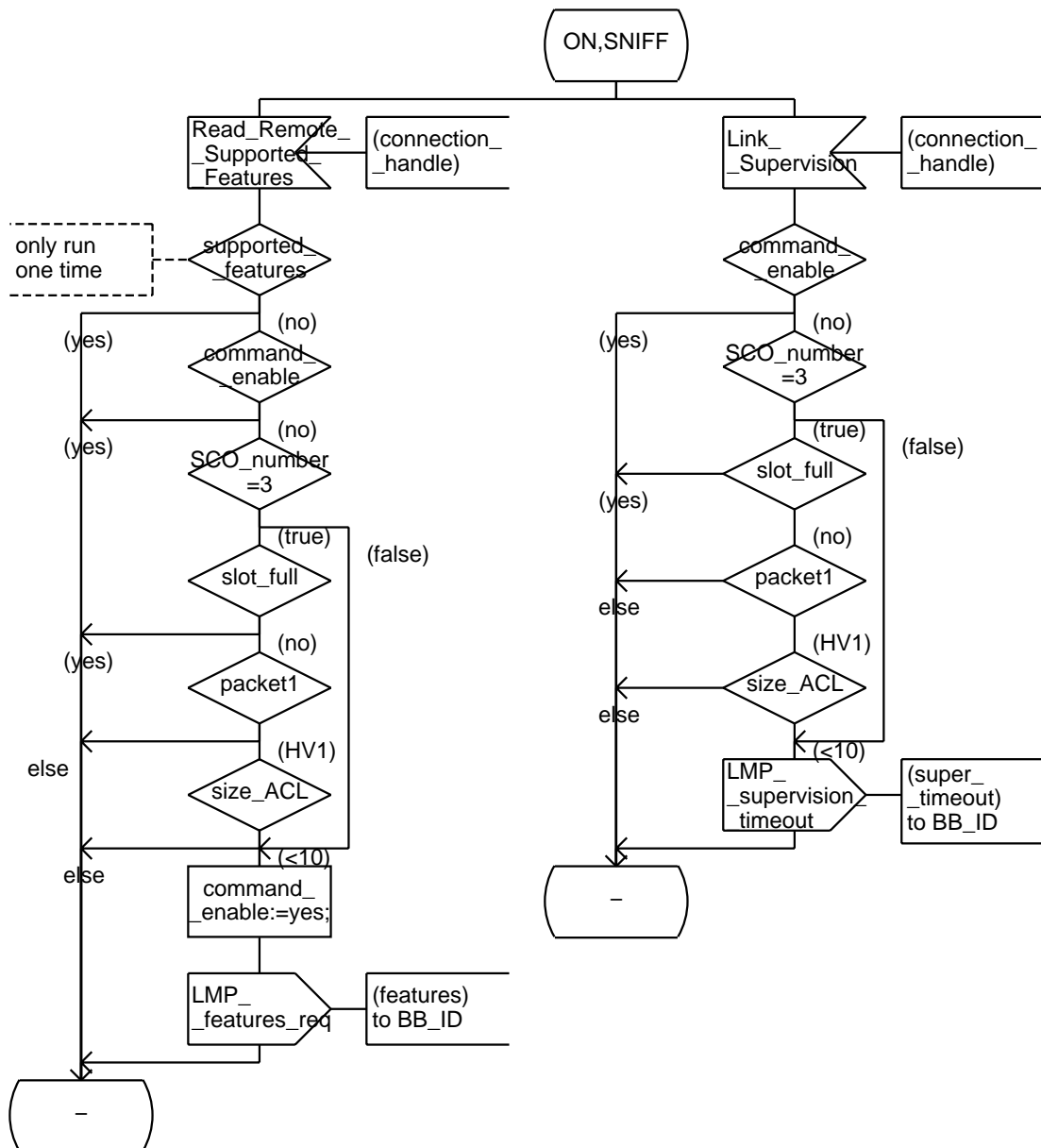


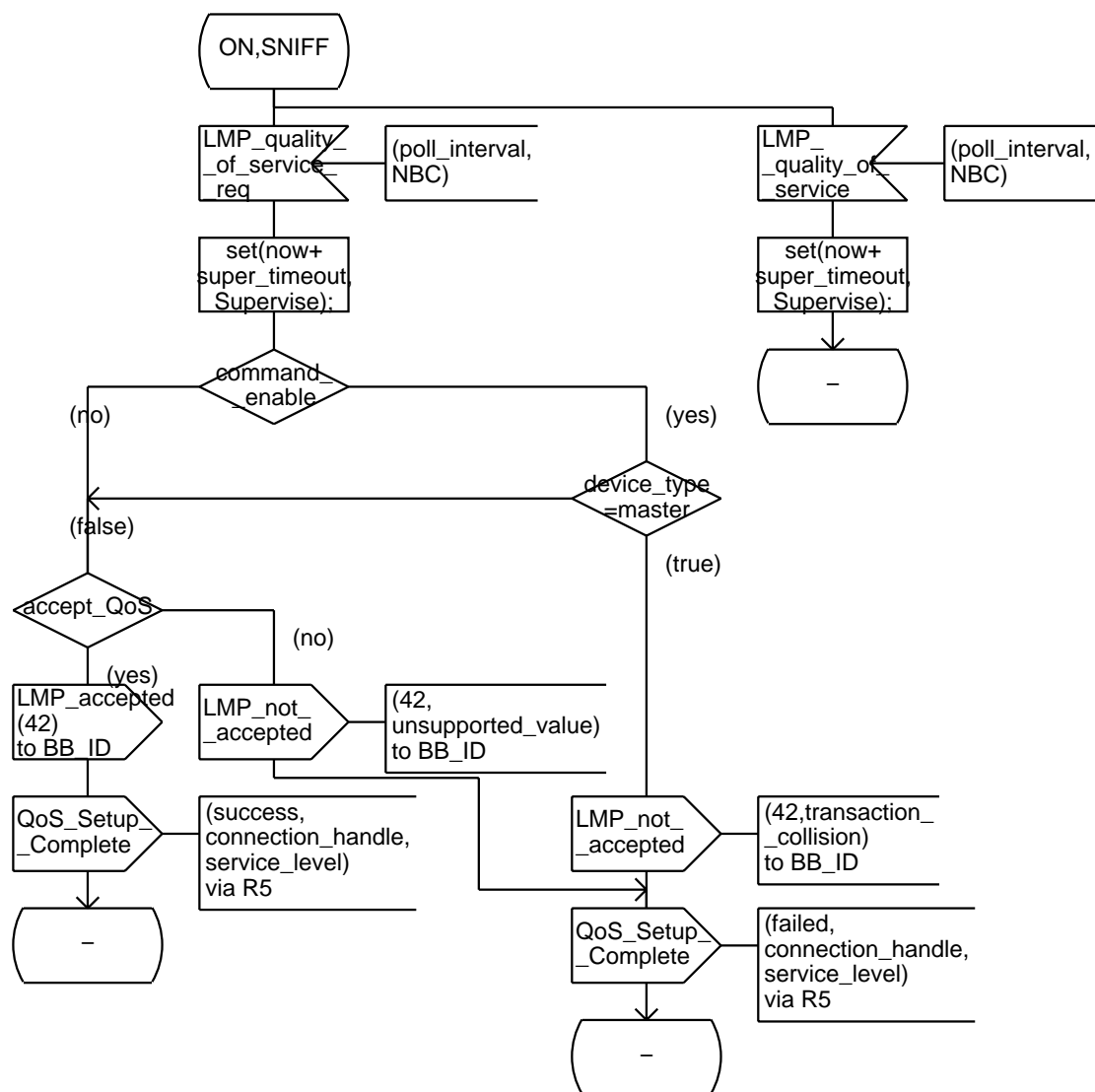


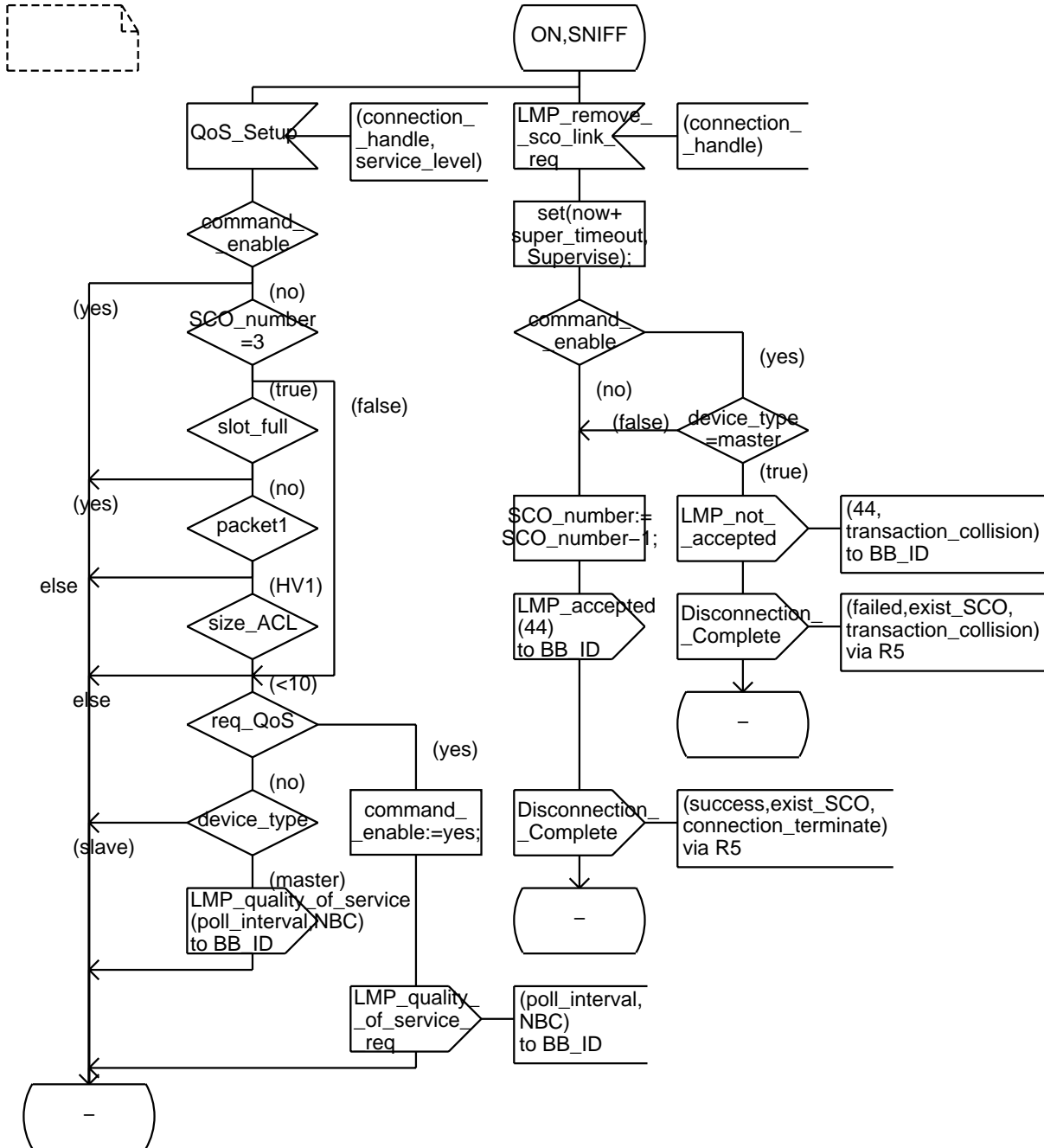


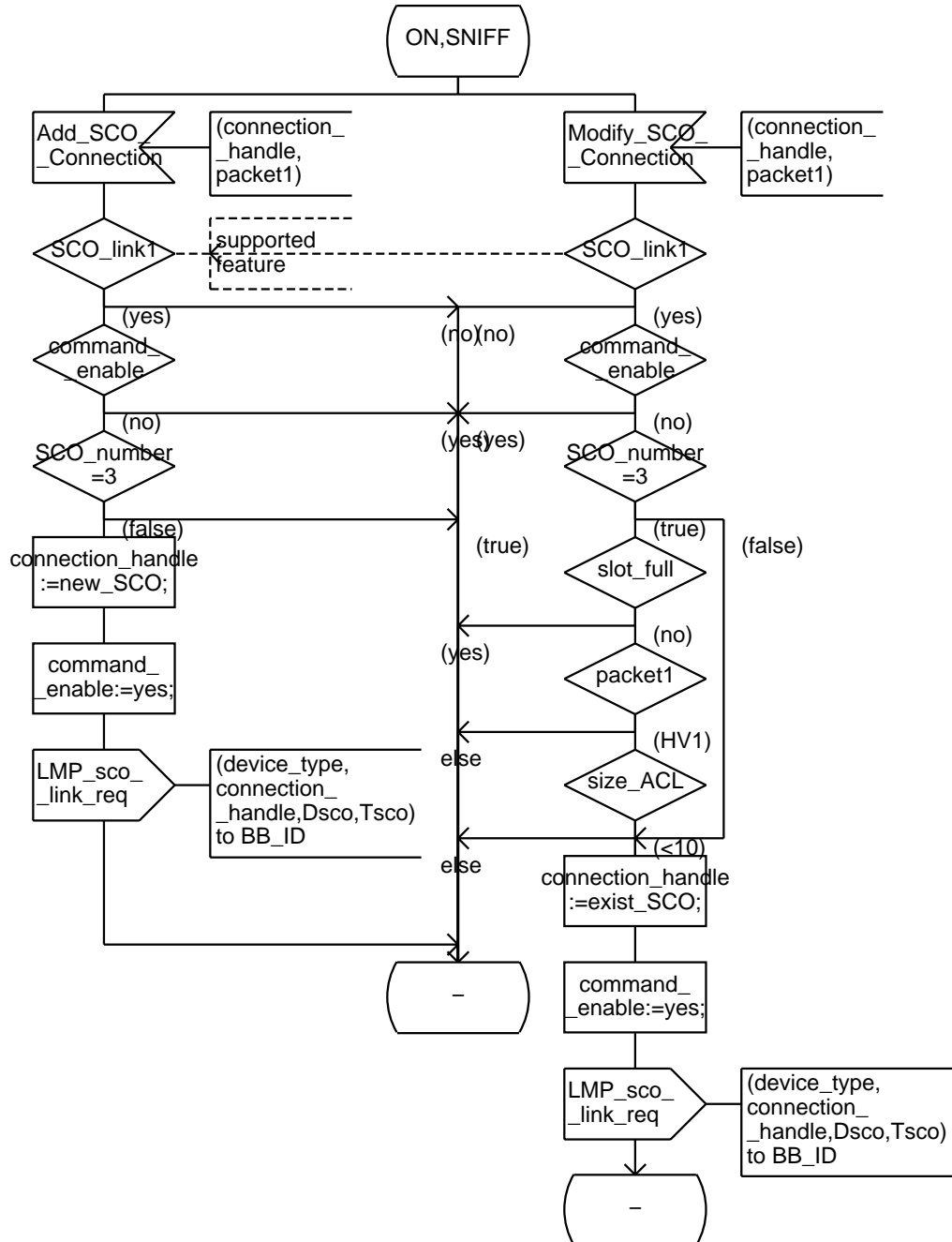


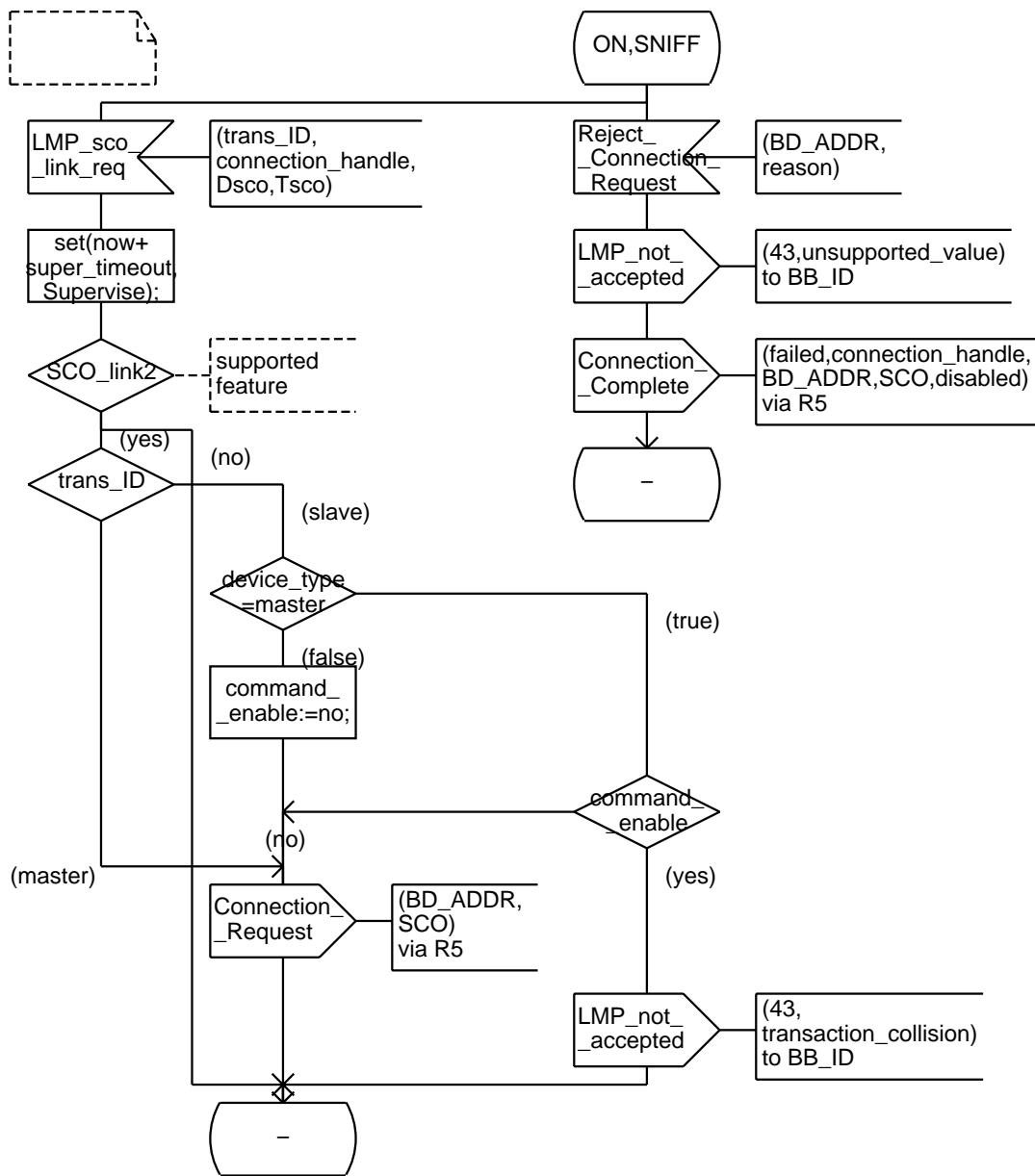


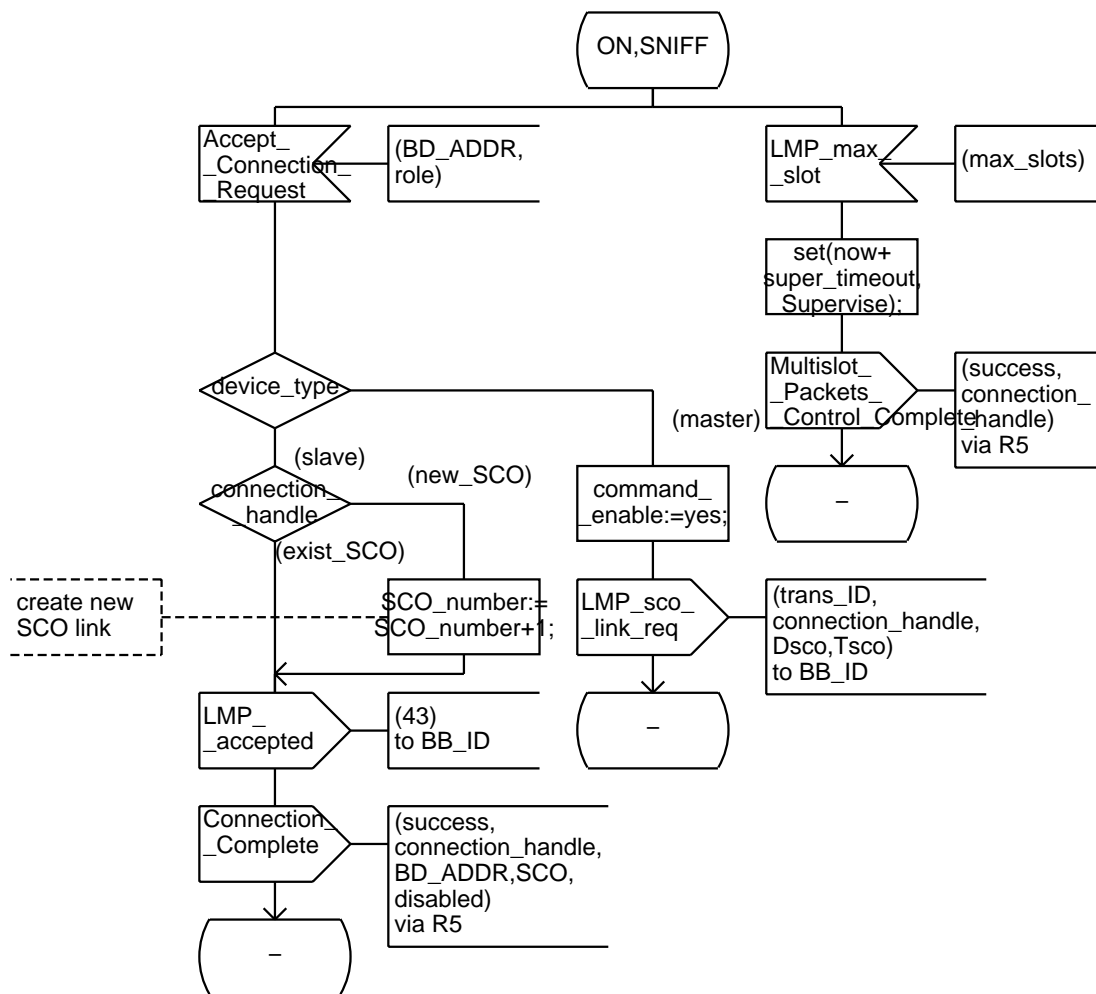


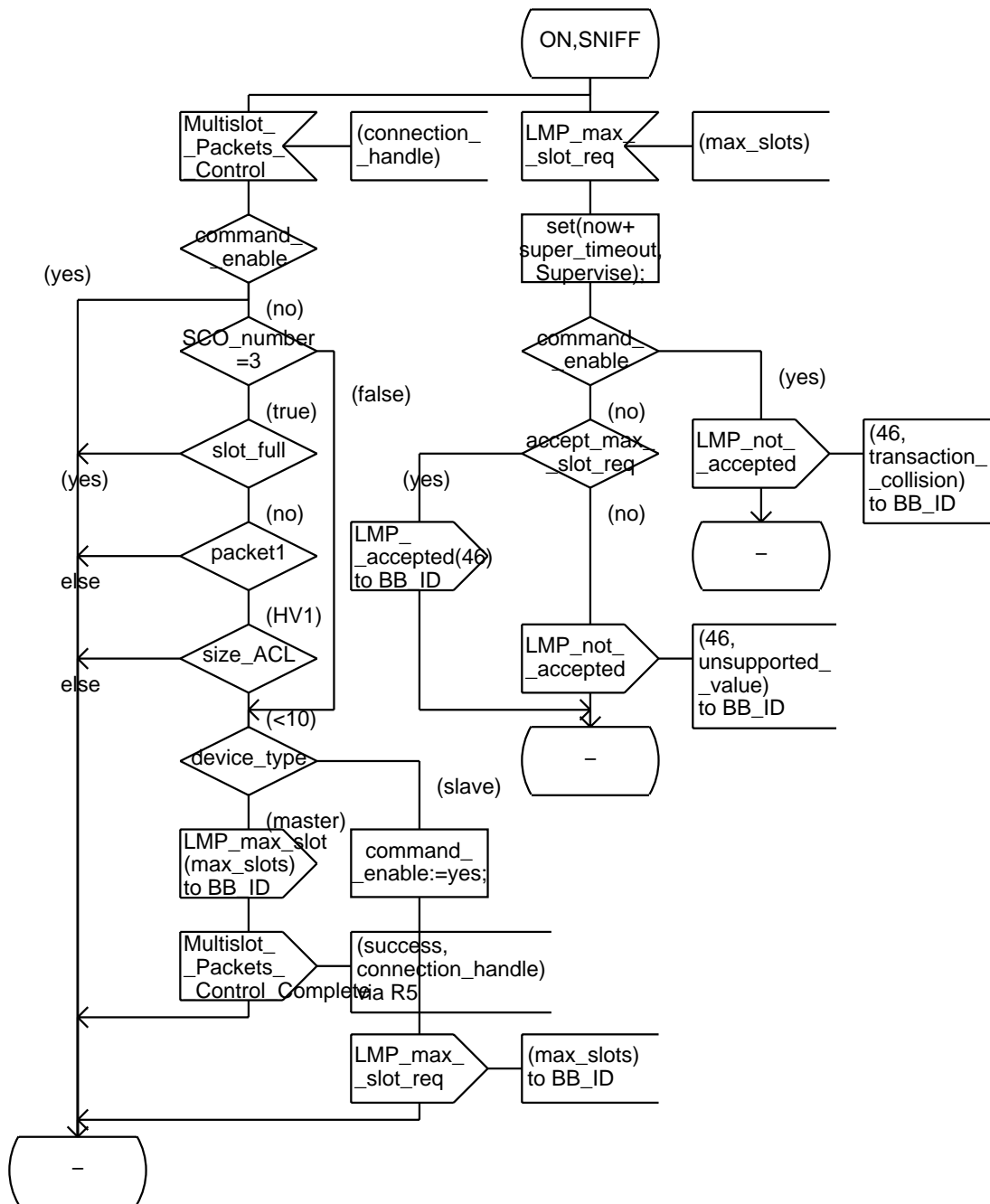


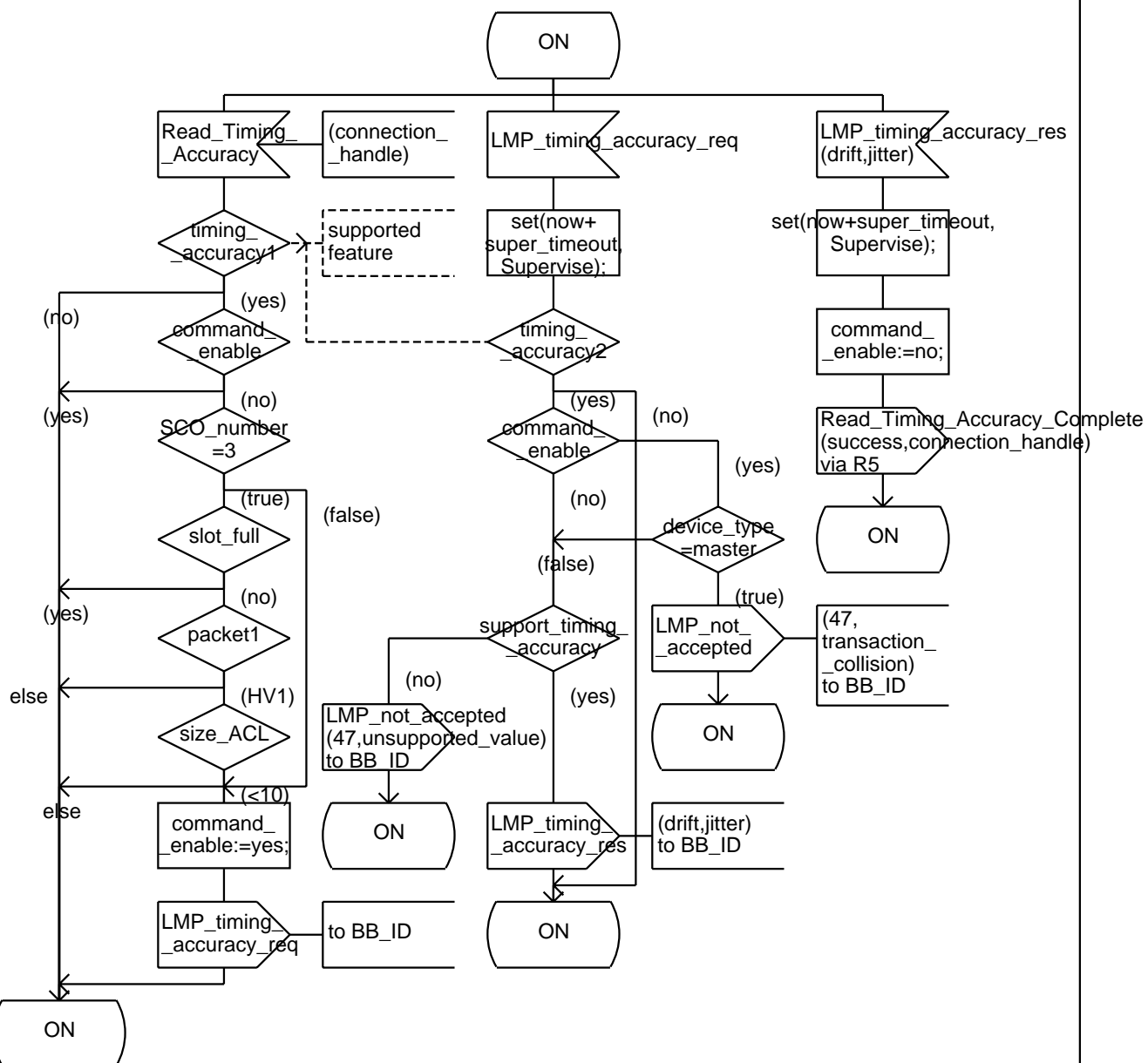


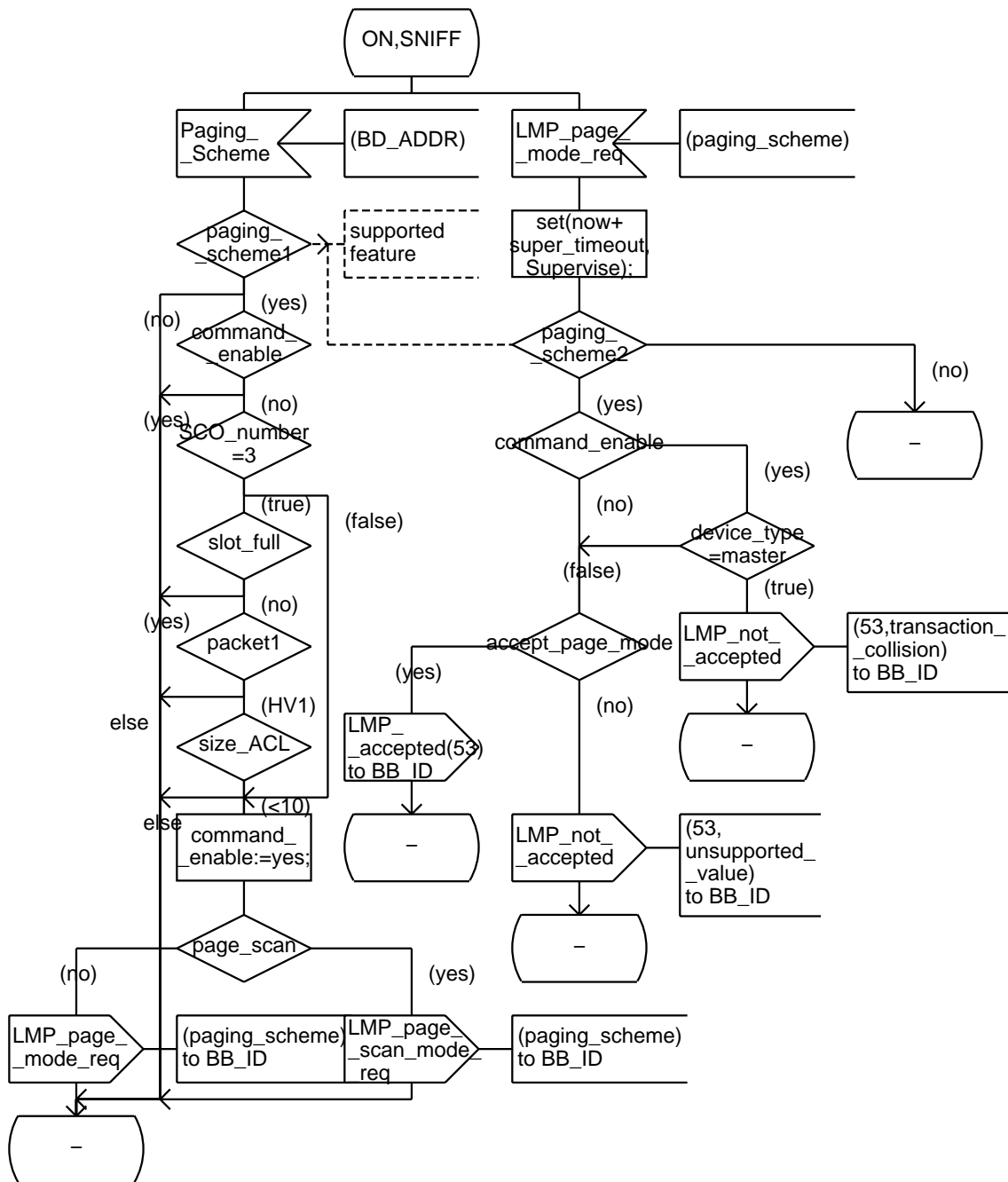


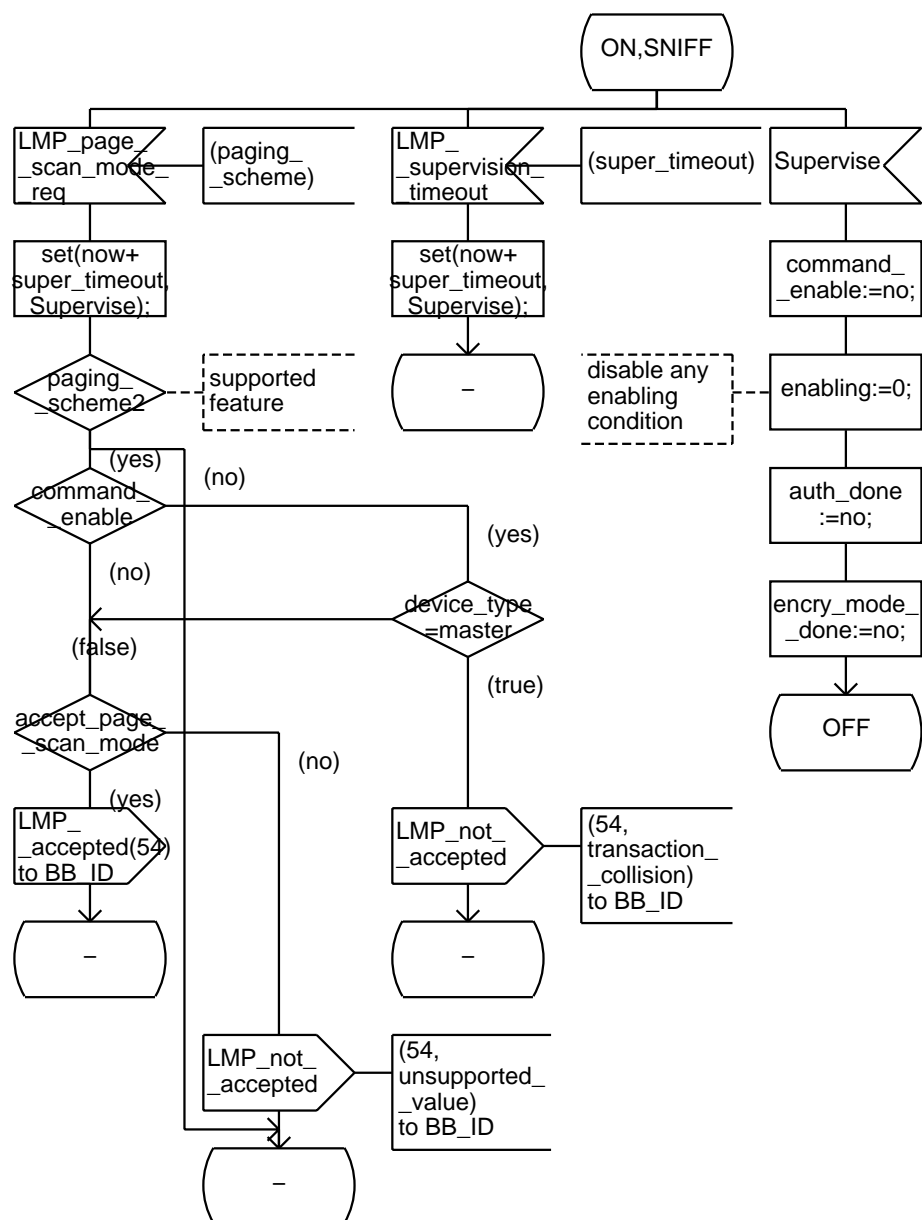


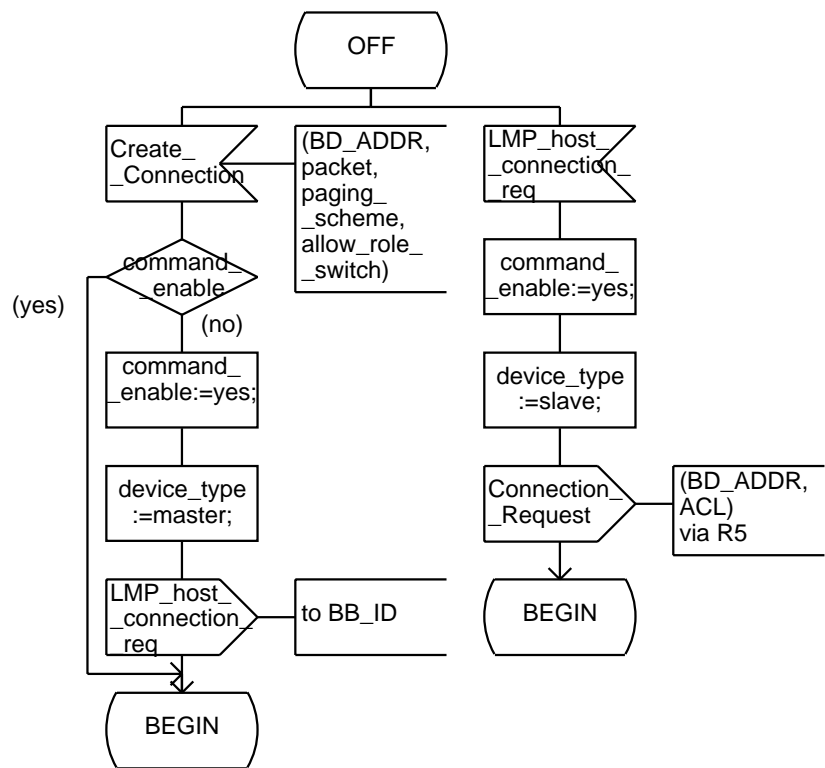






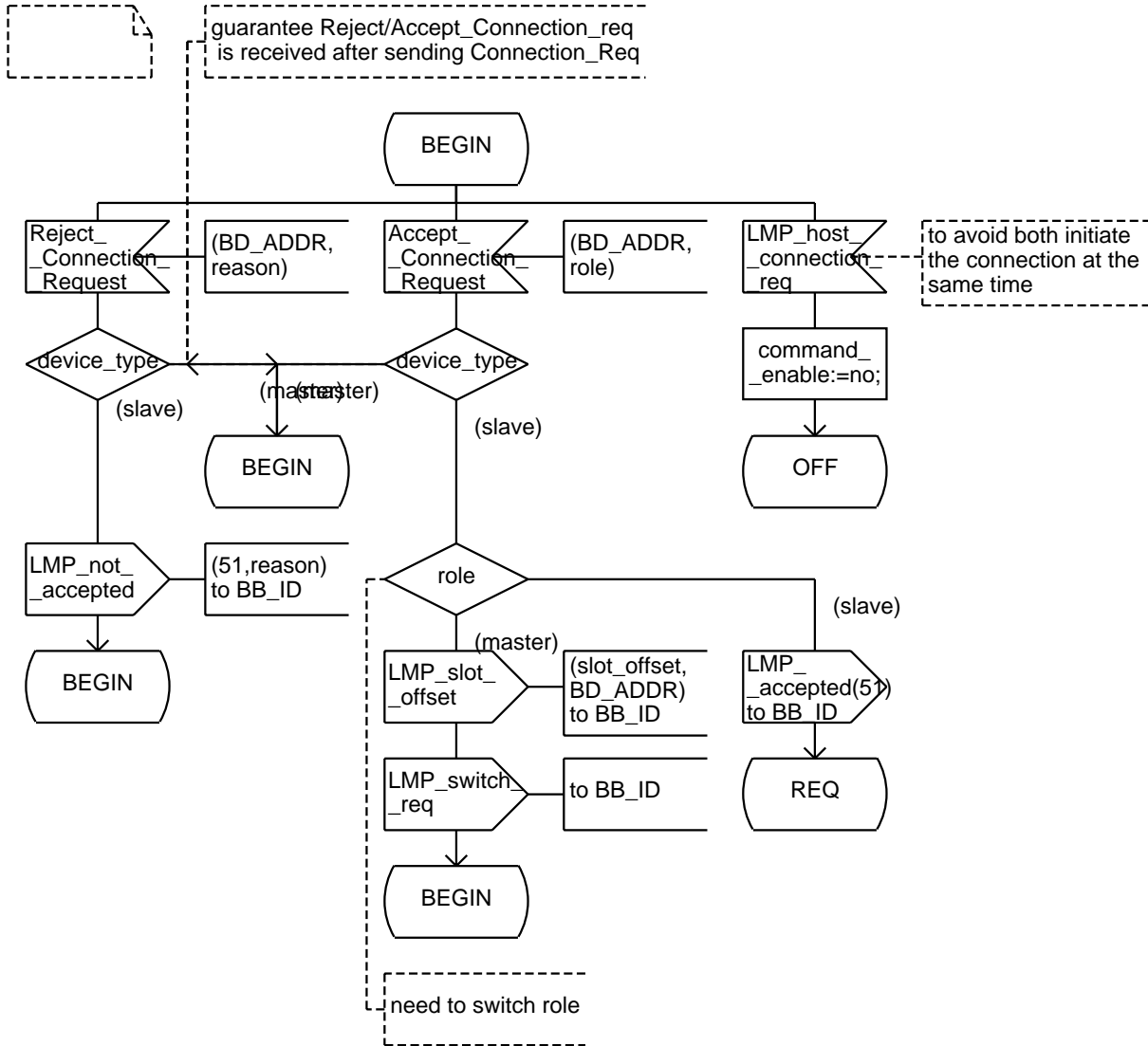


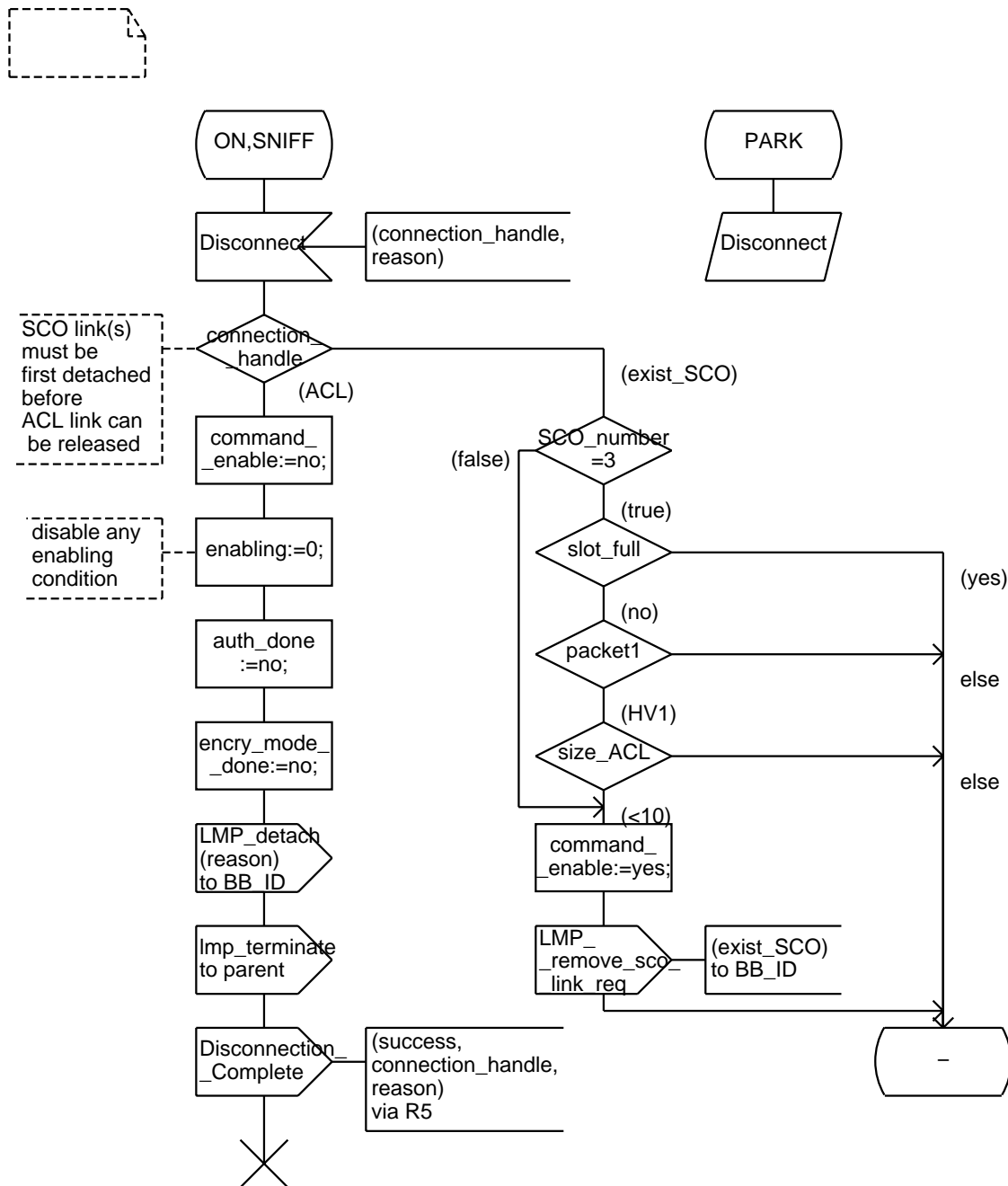


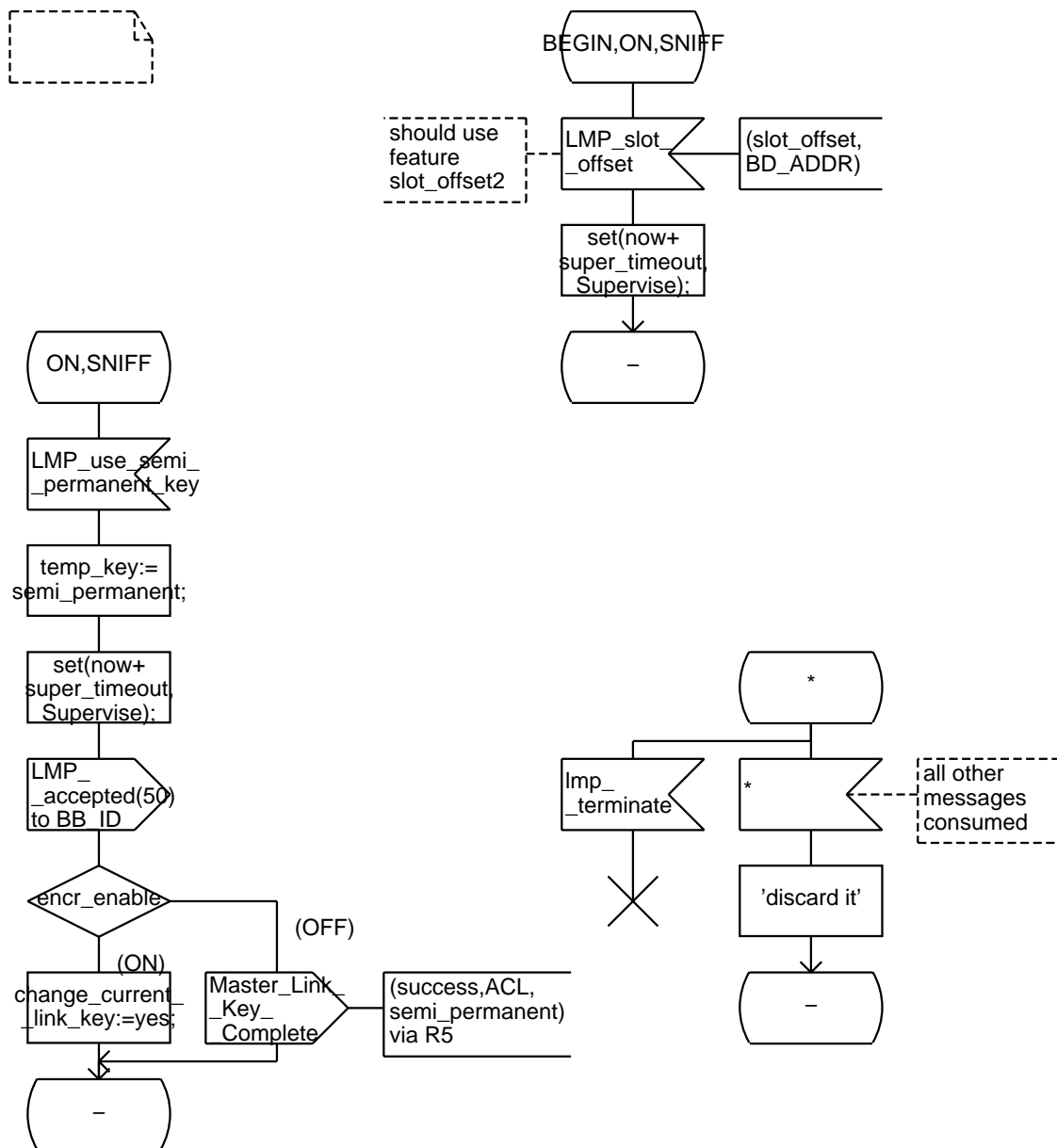


process Imp_p2

72(74)

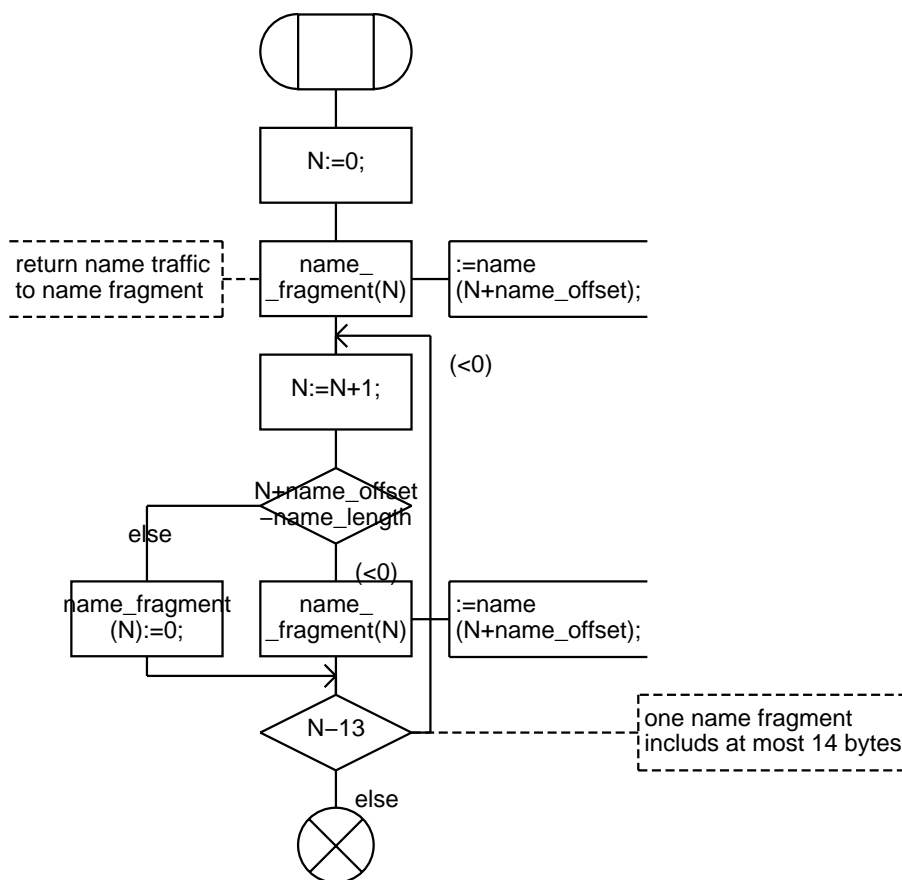


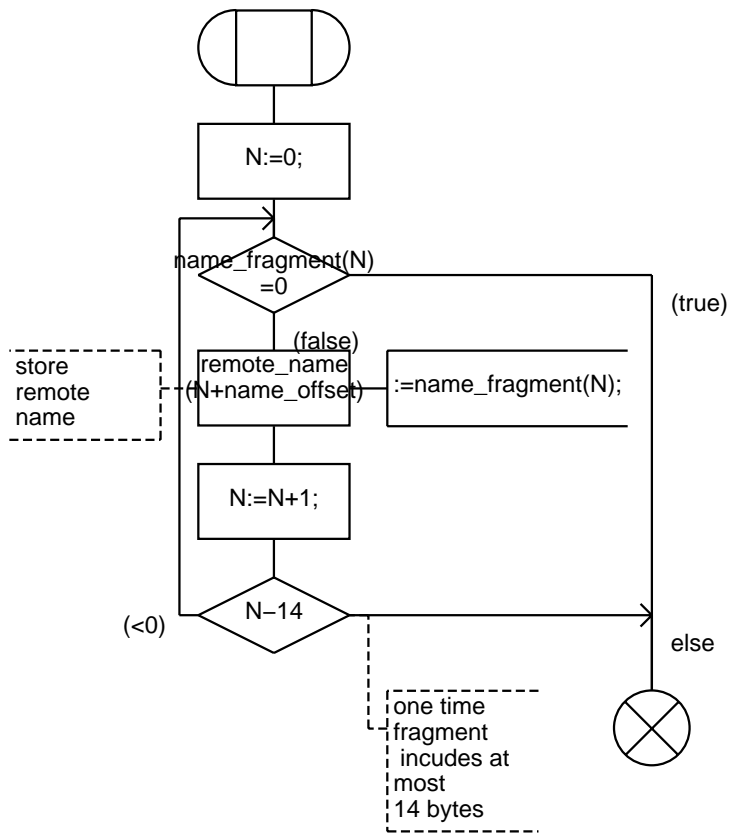




procedure Imp_name_fragment2

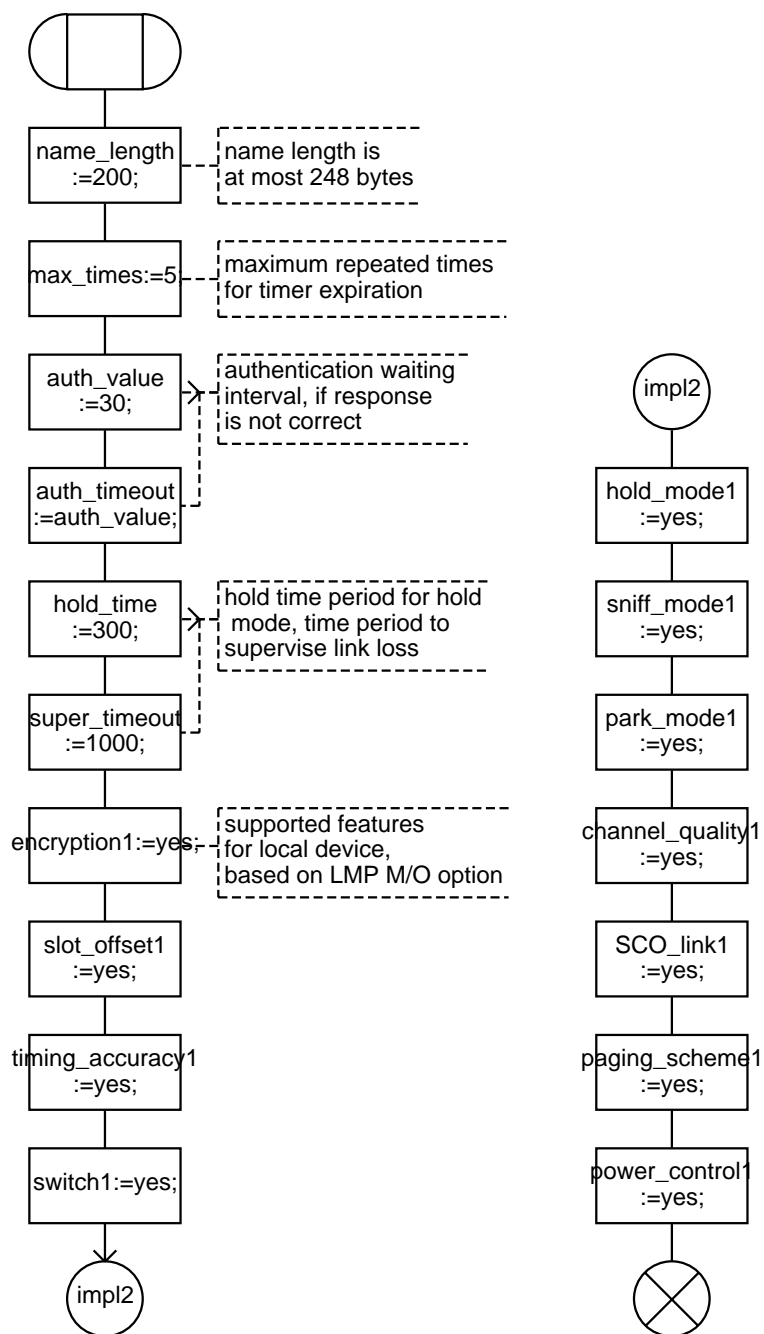
1(1)





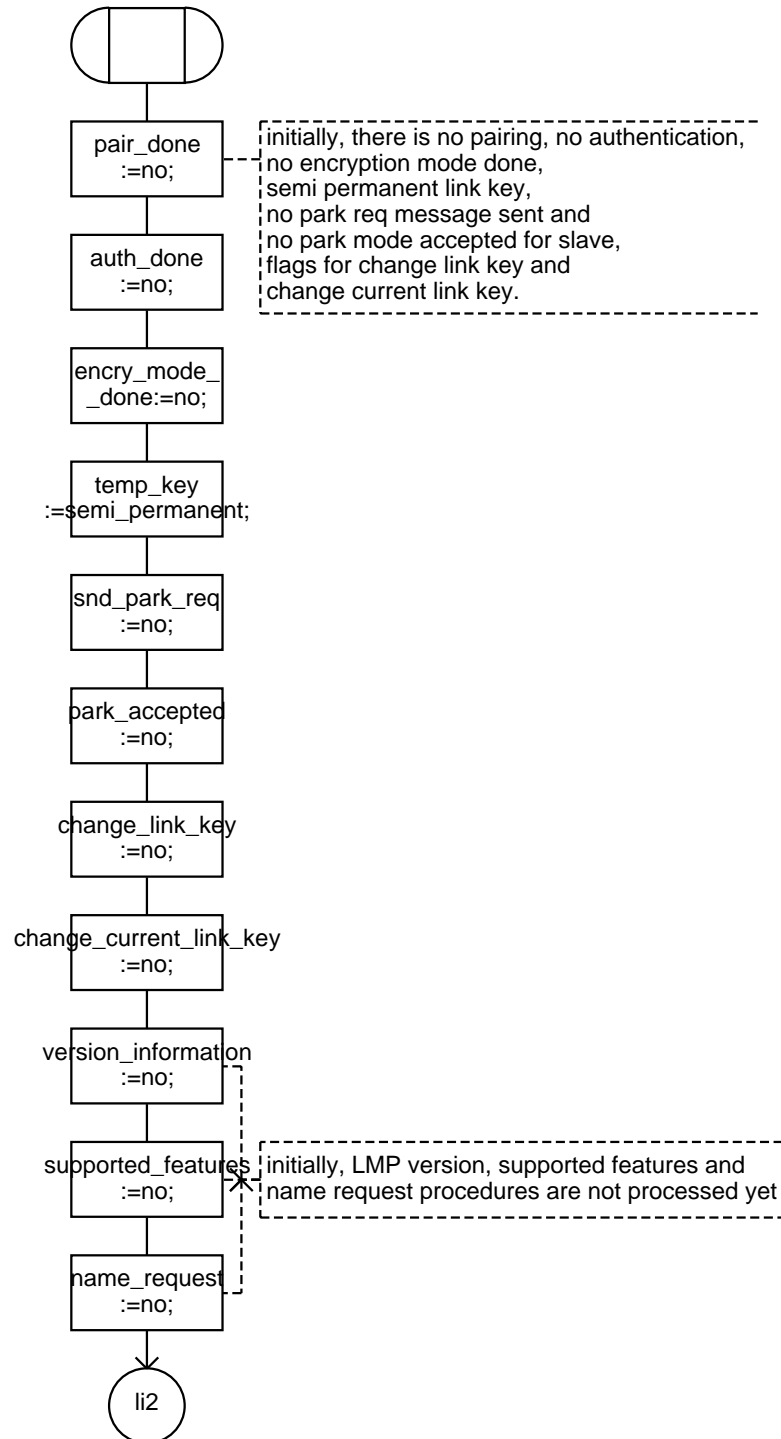
procedure Imp_implement2

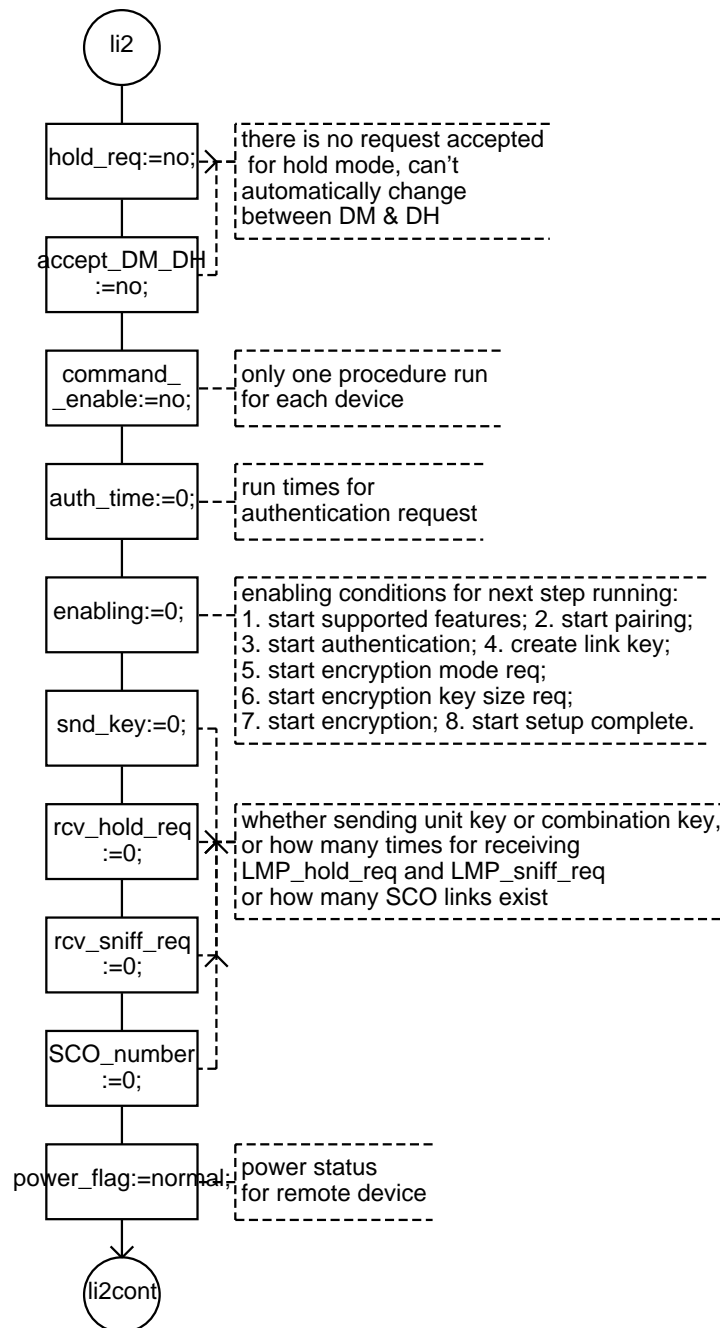
1(1)

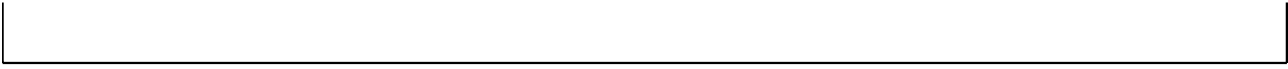


procedure Imp_initialization2

1(3)

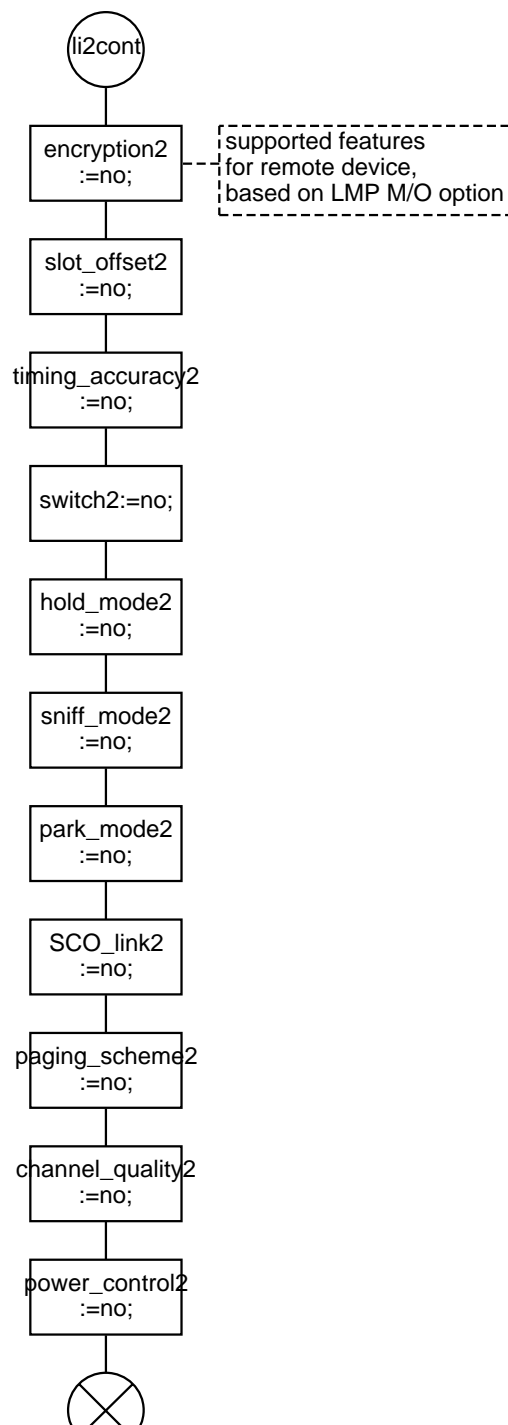




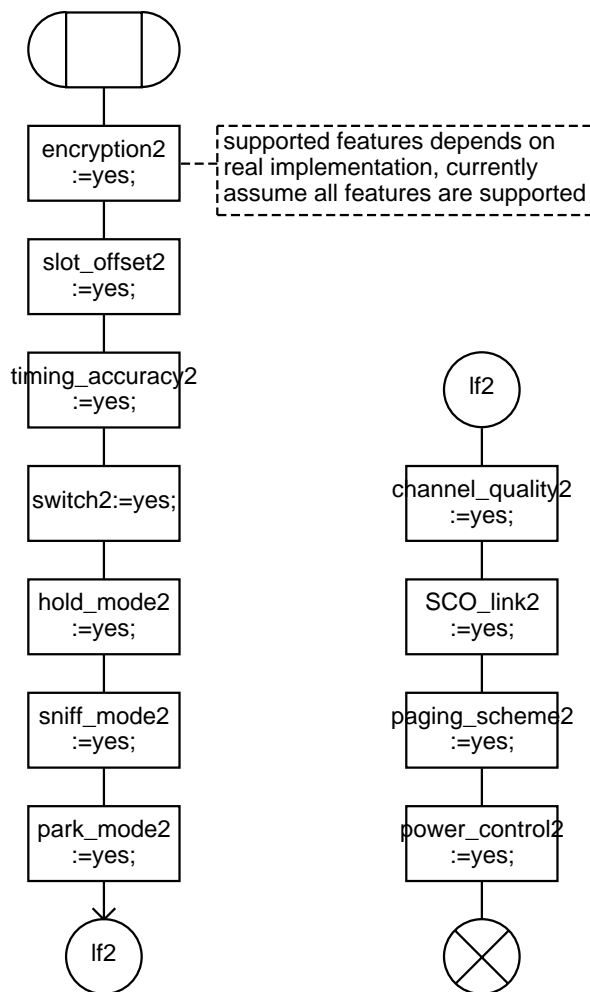


procedure Imp_initialization2

3(3)





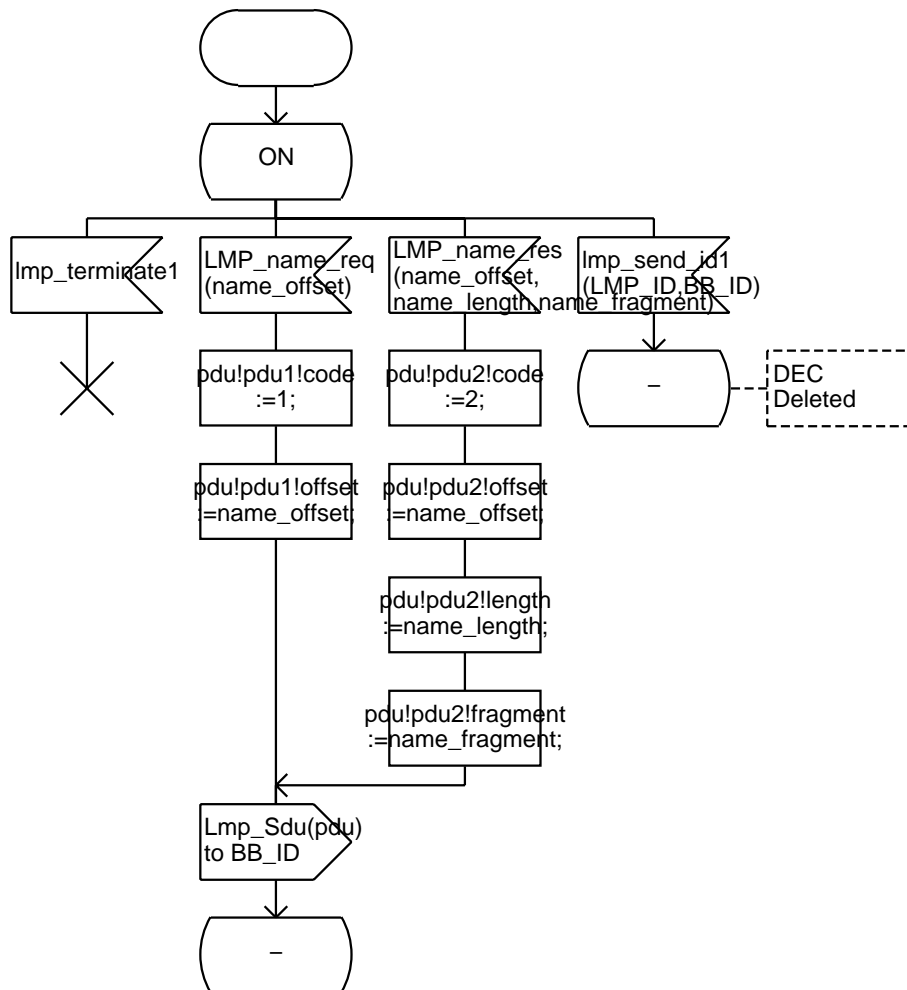


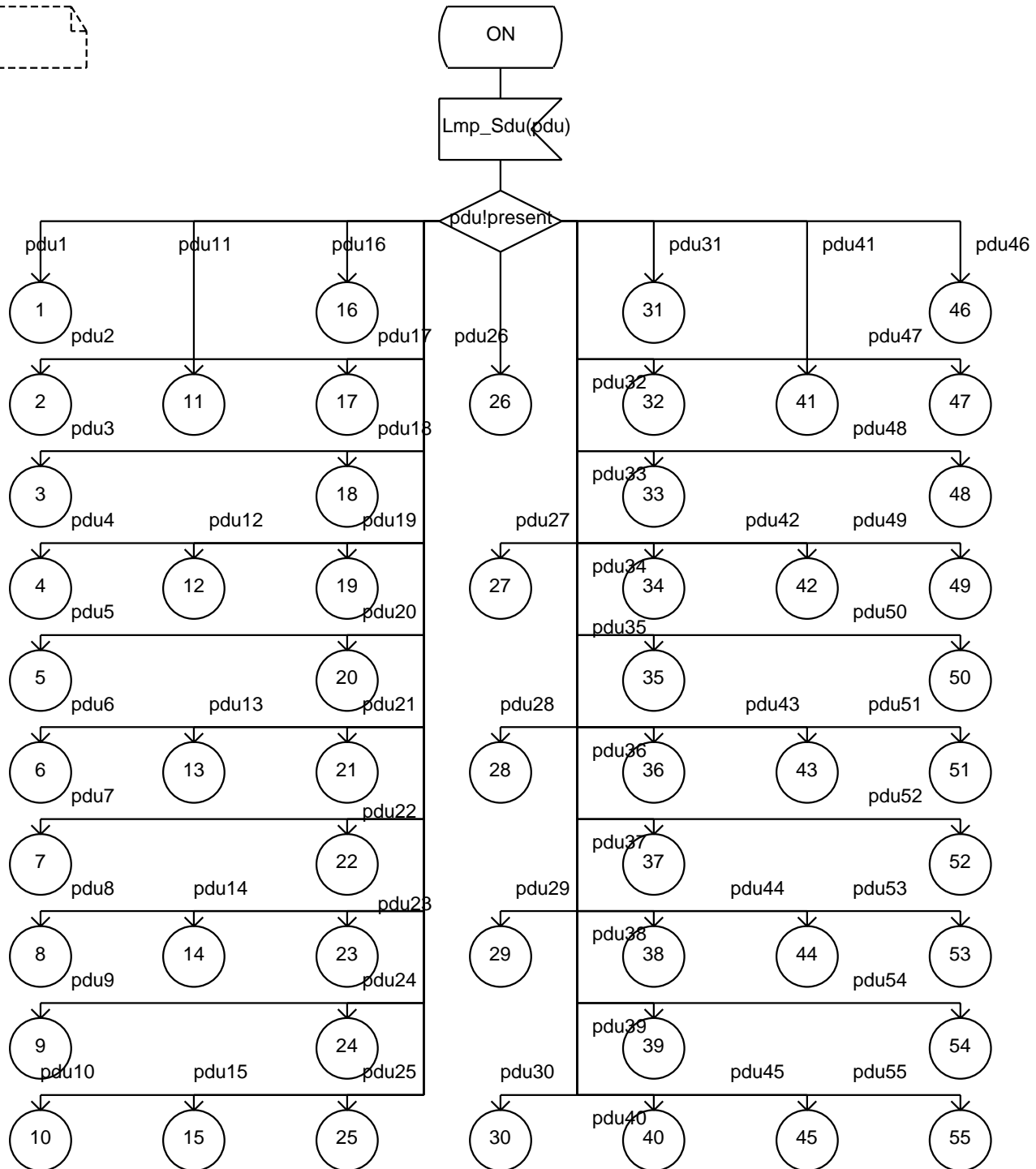


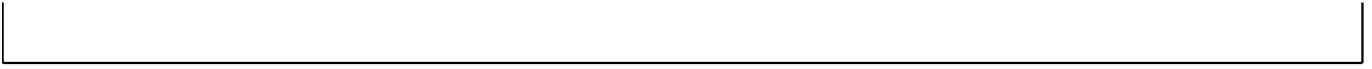
DEC delete Pro_ID & Pro_flag

```
dcl pdu LMP_pdu;  
dcl LMP_ID, BB_ID PId;  
dcl BD_ADDR, AM_ADDR, AR_ADDR,  
    PM_ADDR Address_type;  
dcl connection_handle Handle_type,  
    trans_ID Role_type;  
  
dcl Dsniff, Tsniff, sniff_attempt, sniff_timeout,  
    clock_offset, TB, broadcast_scan_window,  
    poll_interval, Dsco, Tsco, drift, jitter,  
    slot_offset, hold_time, super_timeout Duration;  
dcl features, op_code, name_offset, name_length,  
    key_size, DB, NB, data_rate, VersNr, Compld,  
    SubVersNr, NBC, max_slots Integer;  
  
dcl encr_mode Encryption_mode;  
dcl key, rand Key_content;  
dcl reason Imp_reason_type,  
    auth_res Response_content,  
    name_fragment Name_type,  
    paging_scheme Paging_mode;
```



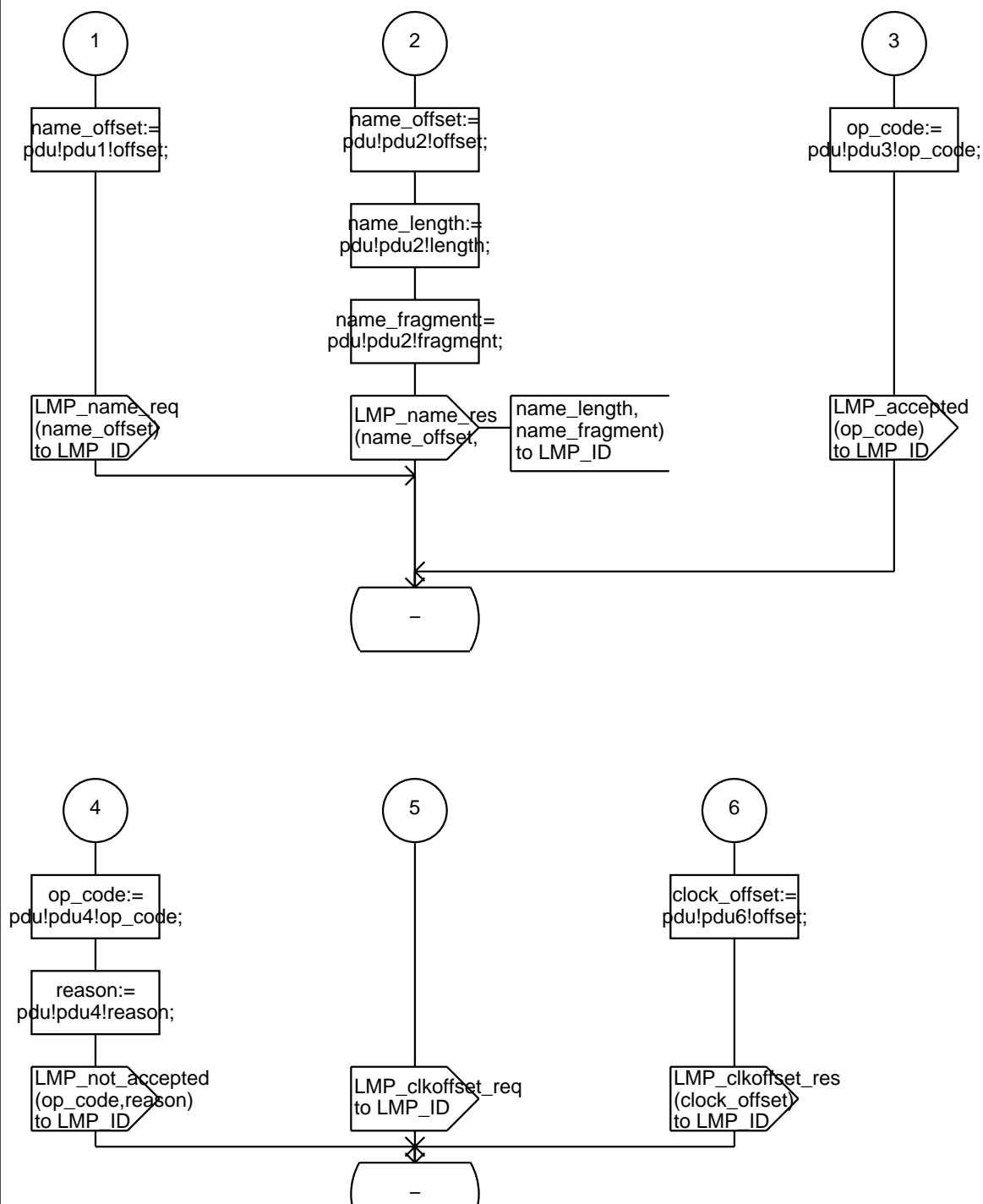




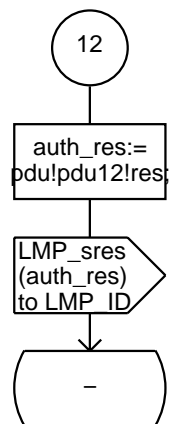
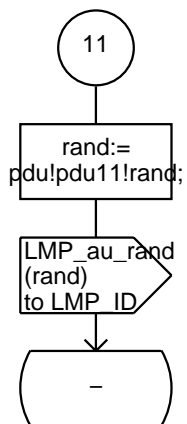
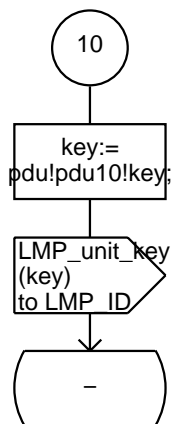
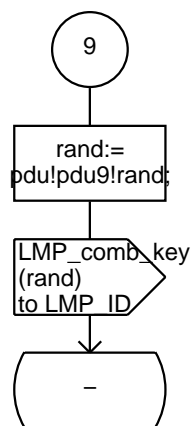
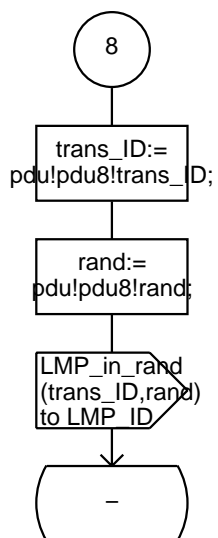
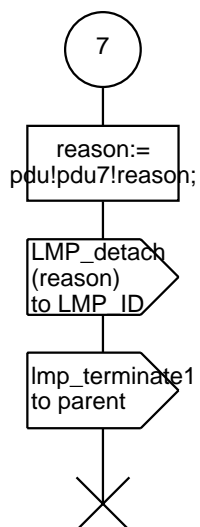


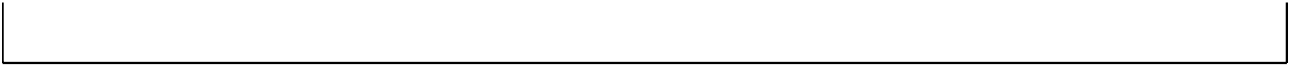
process Imp_codex

4(23)



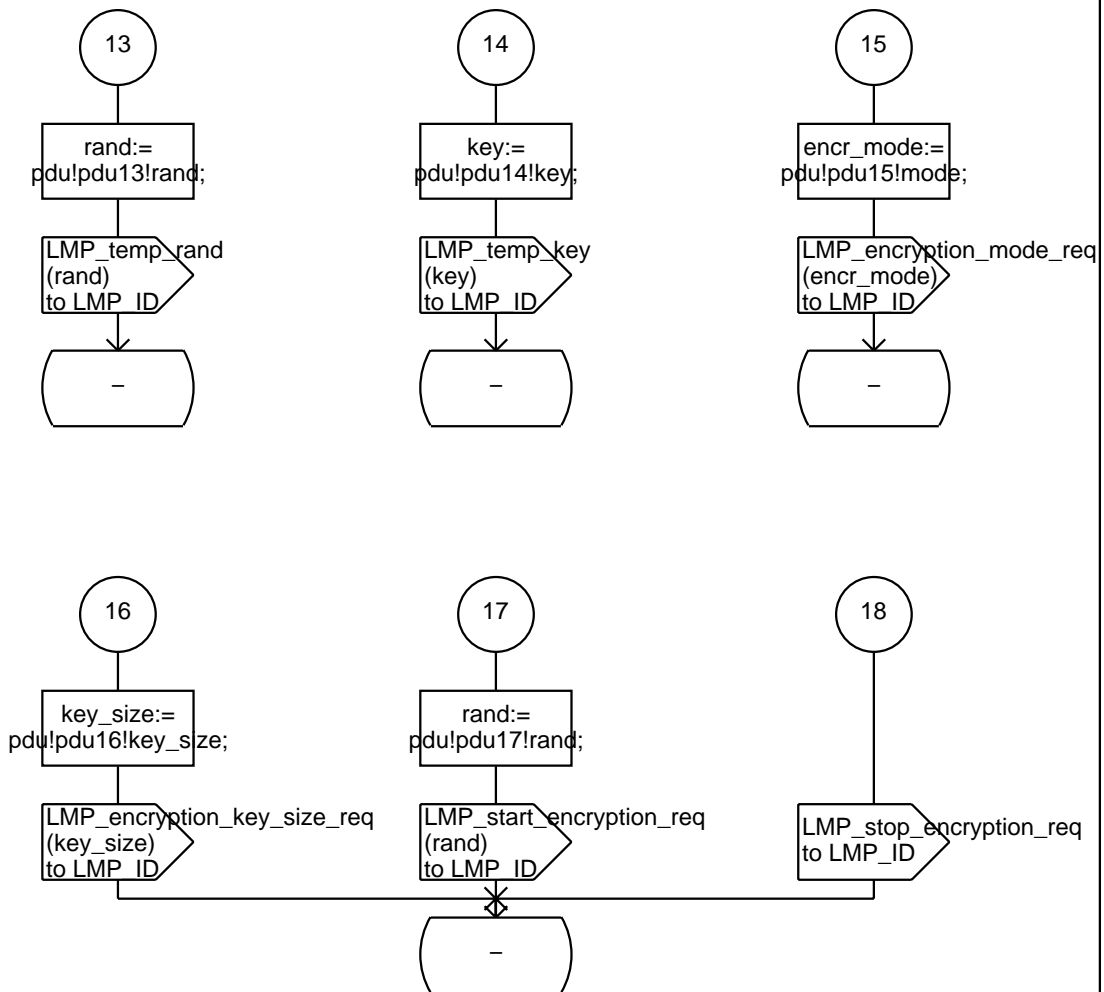




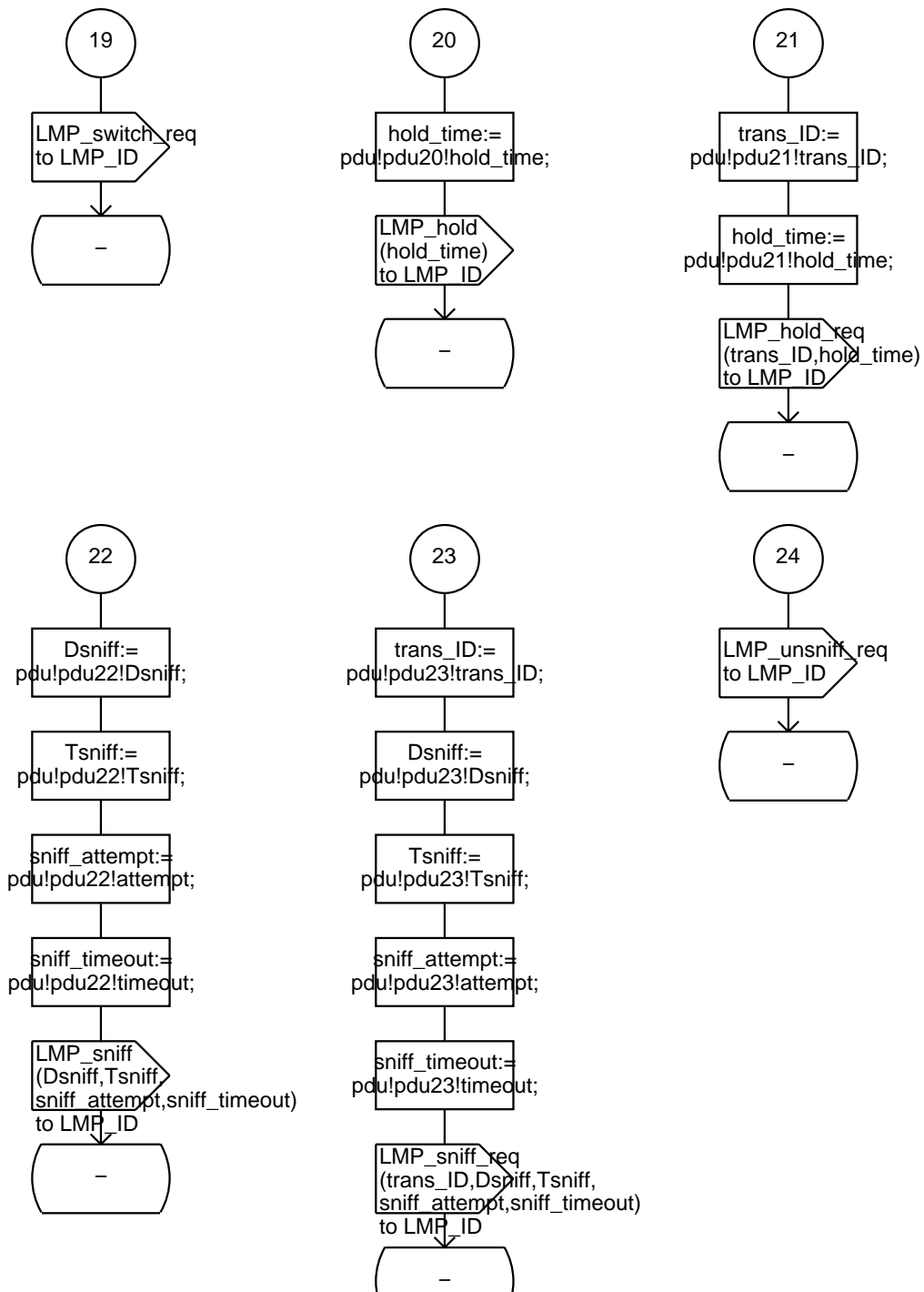


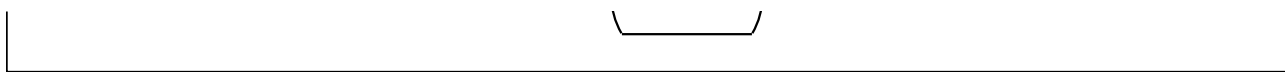
process lmp_codex

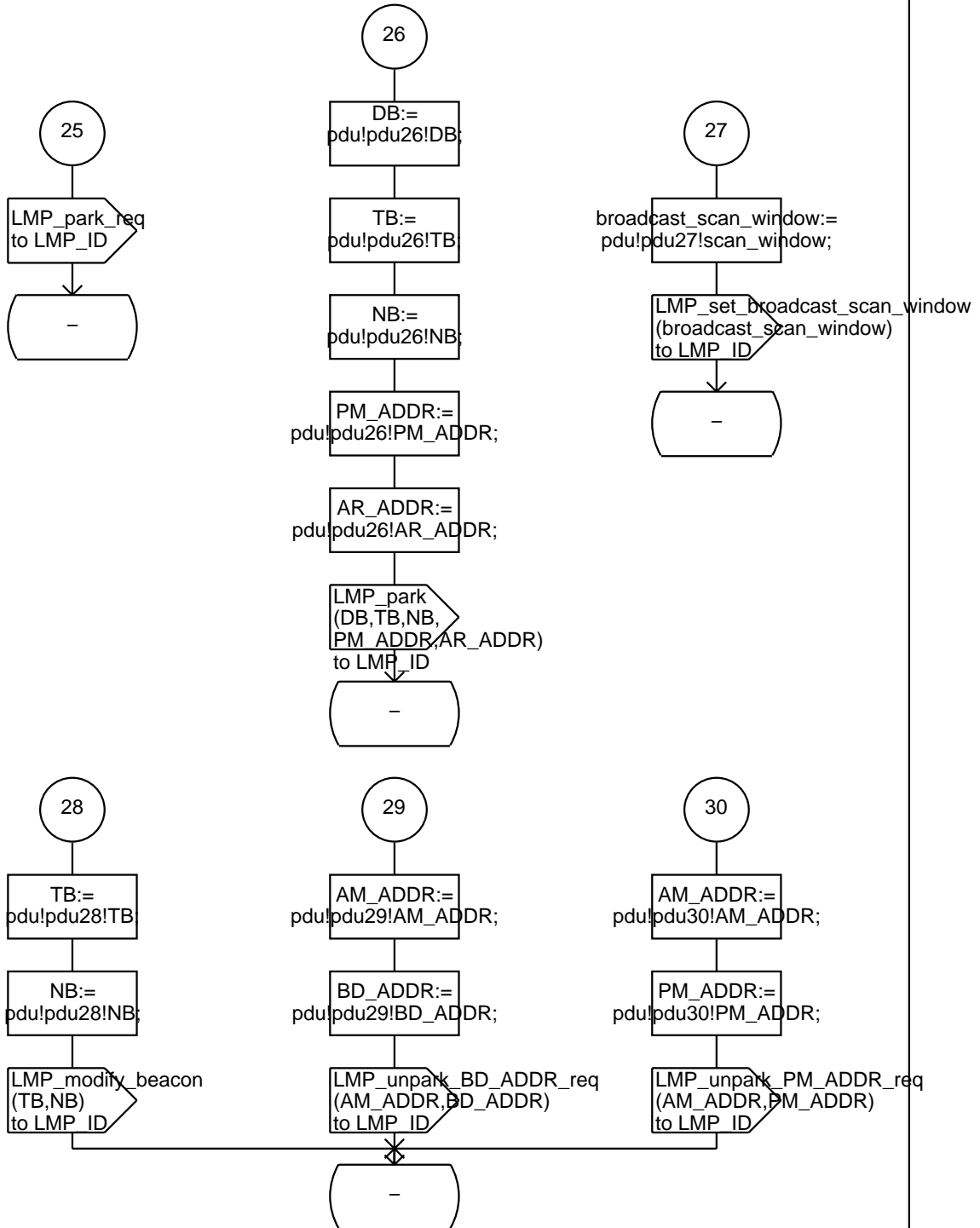
6(23)

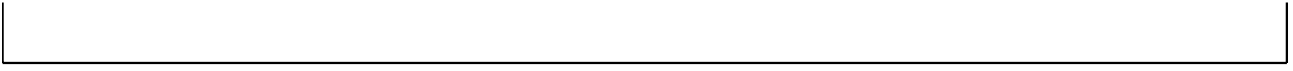






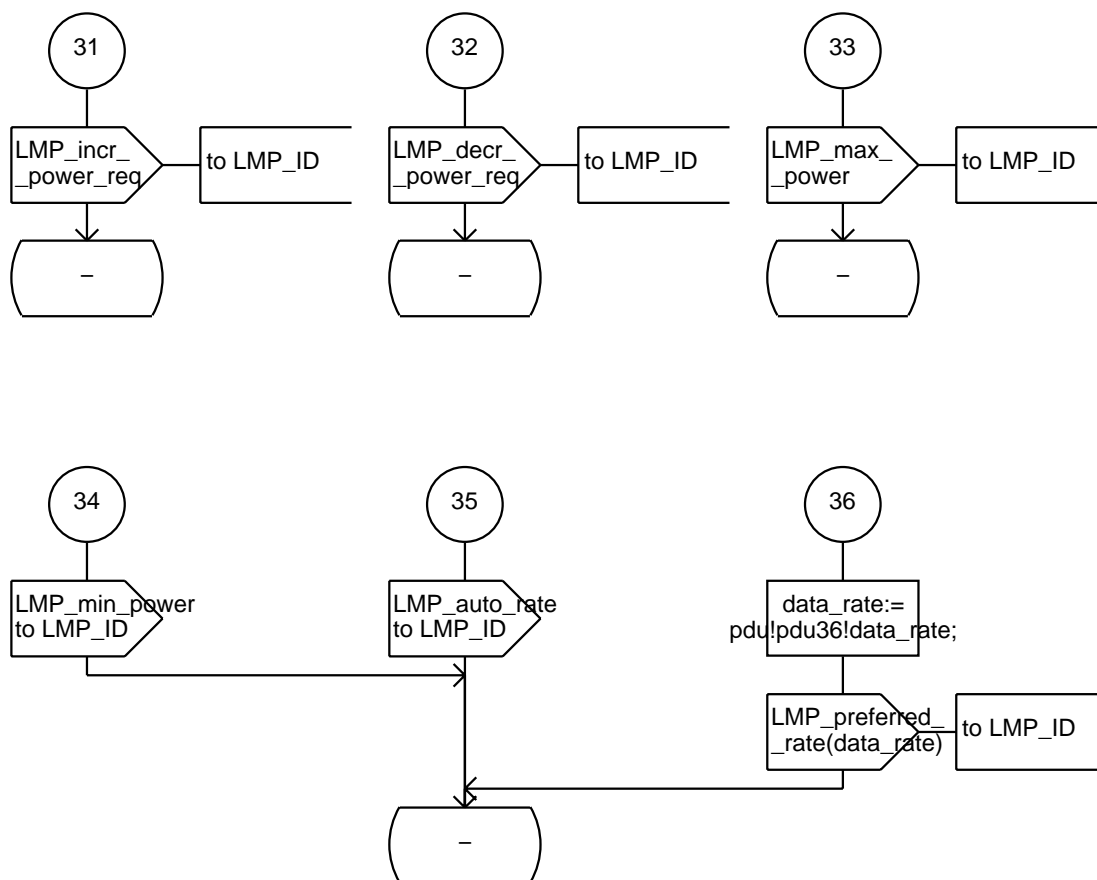


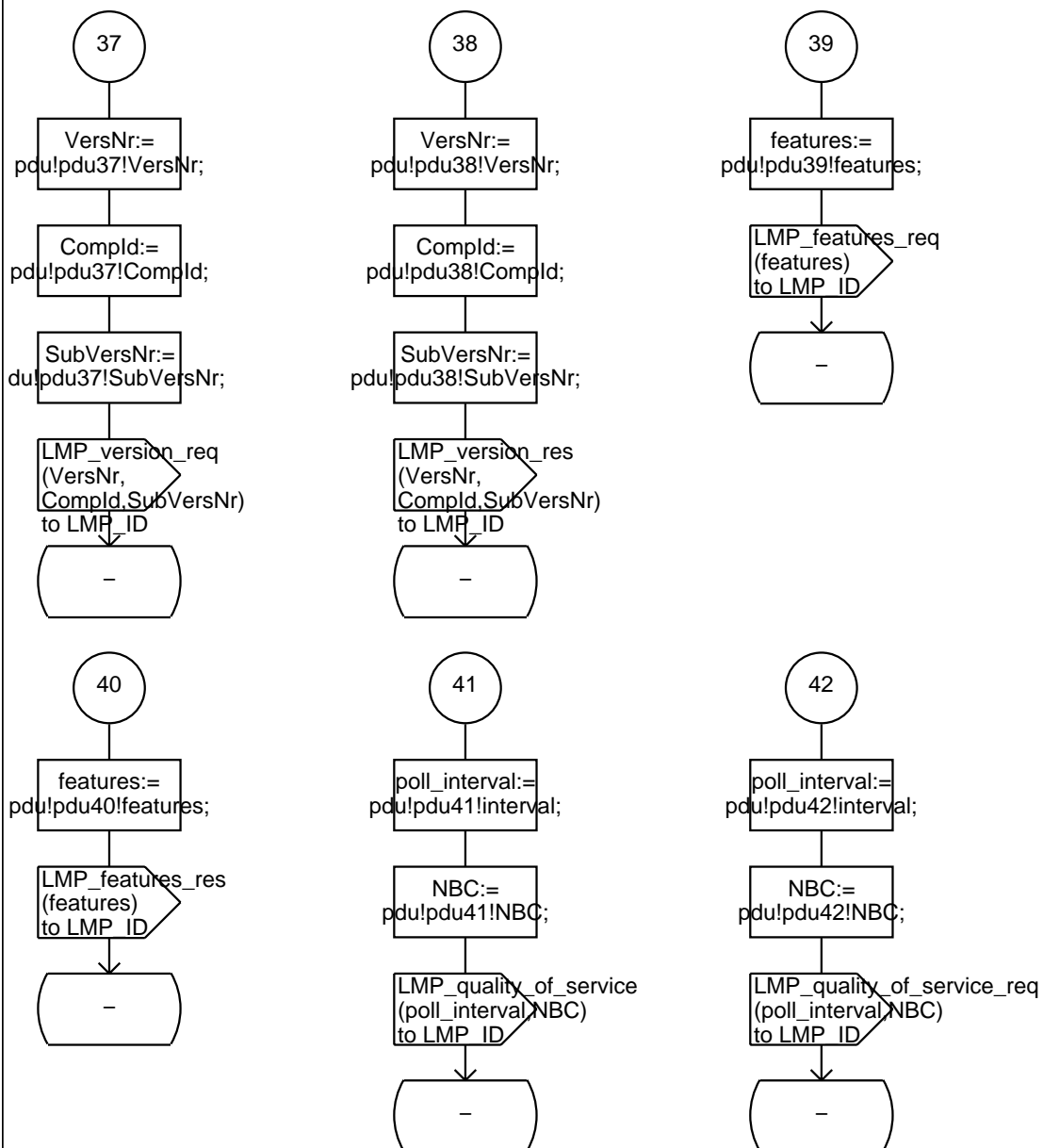




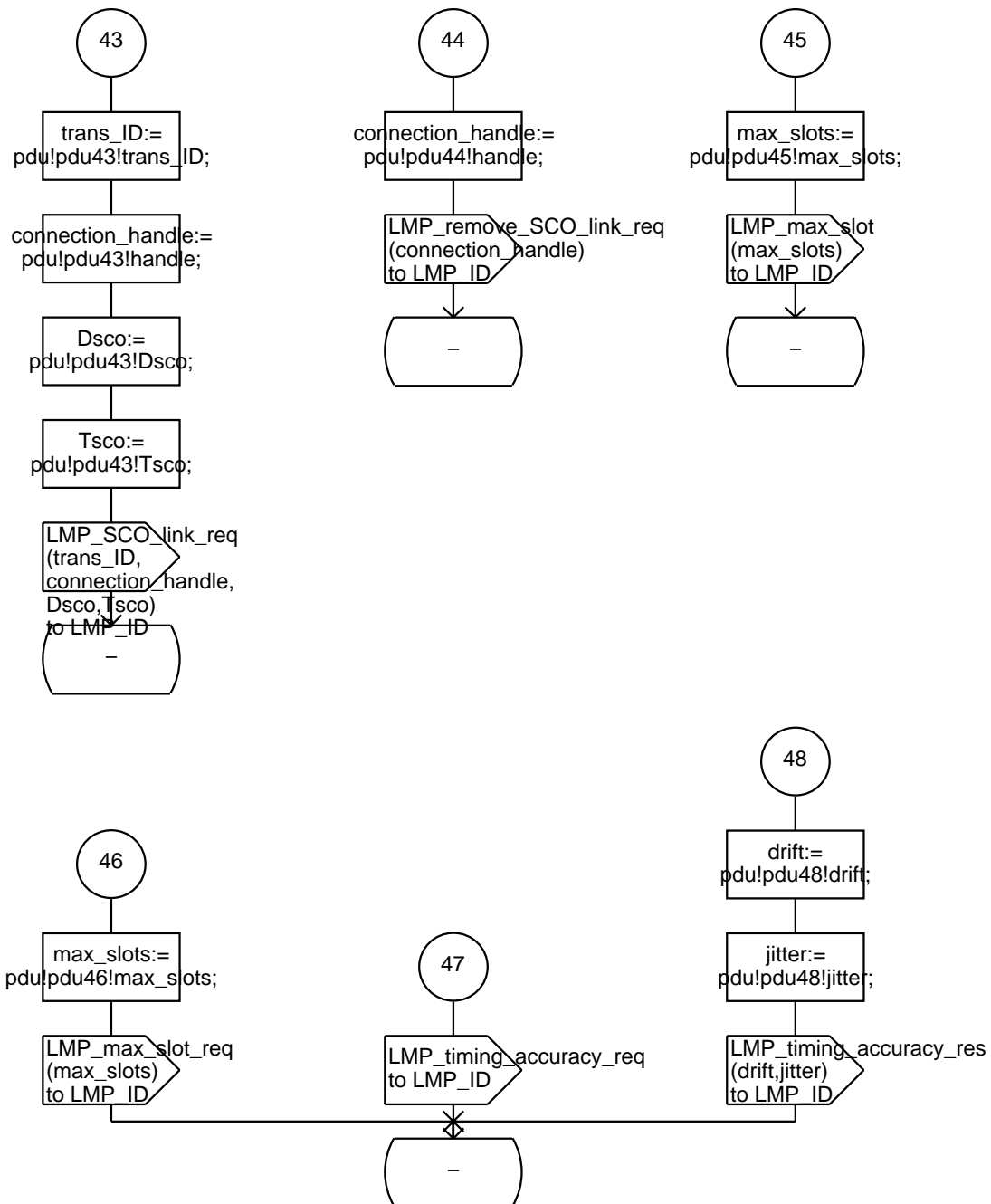
process lmp_codex

9(23)

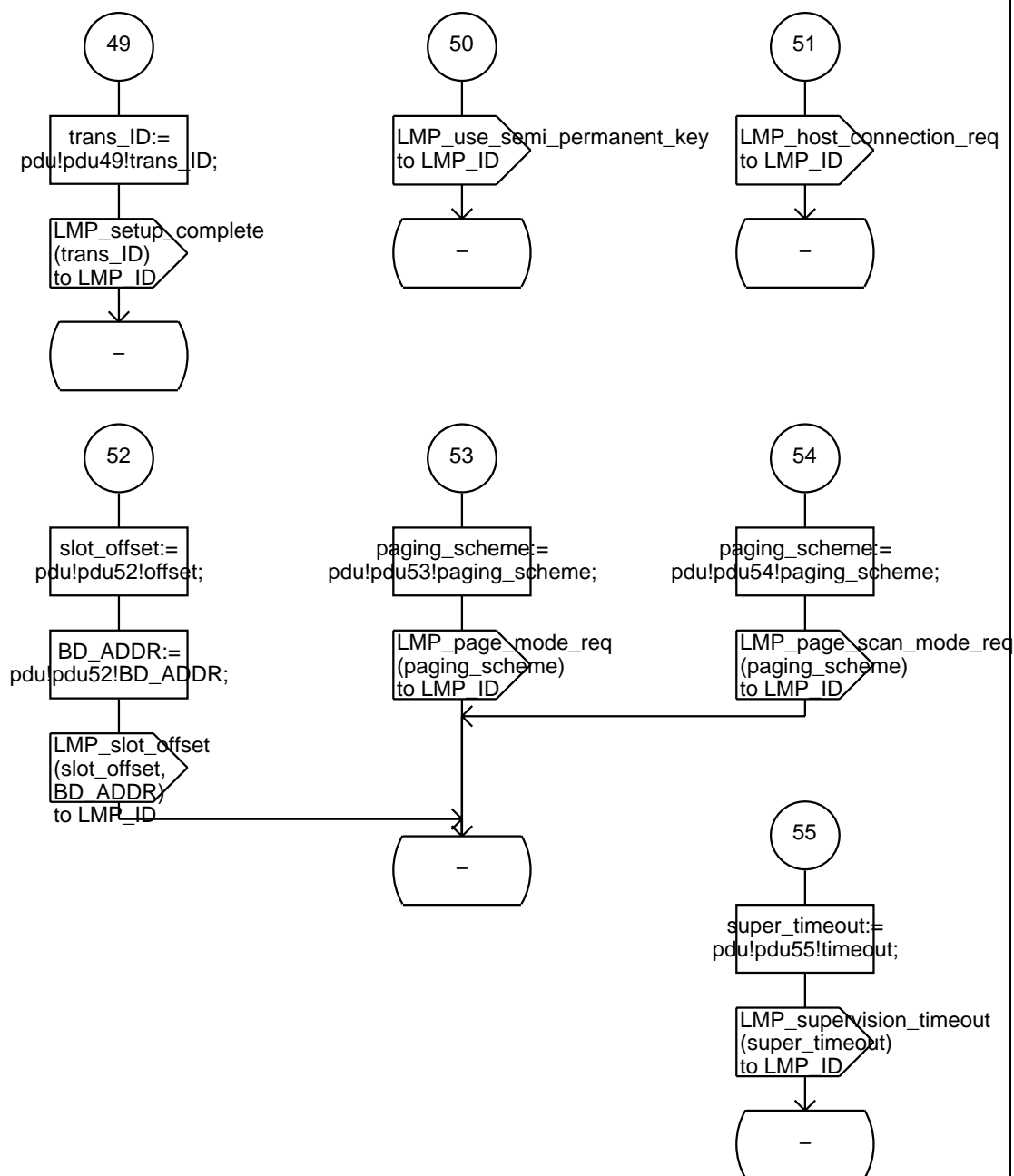




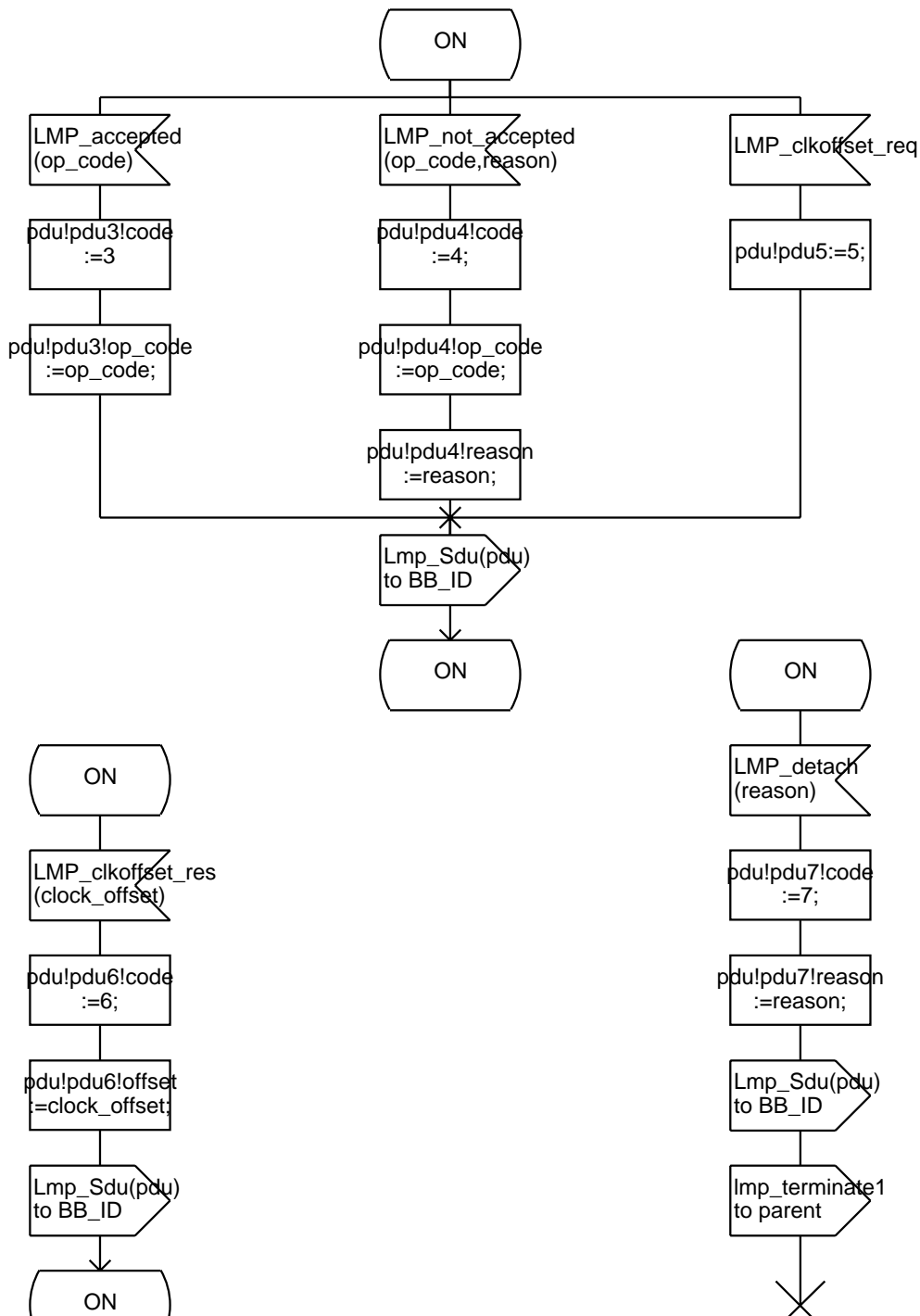




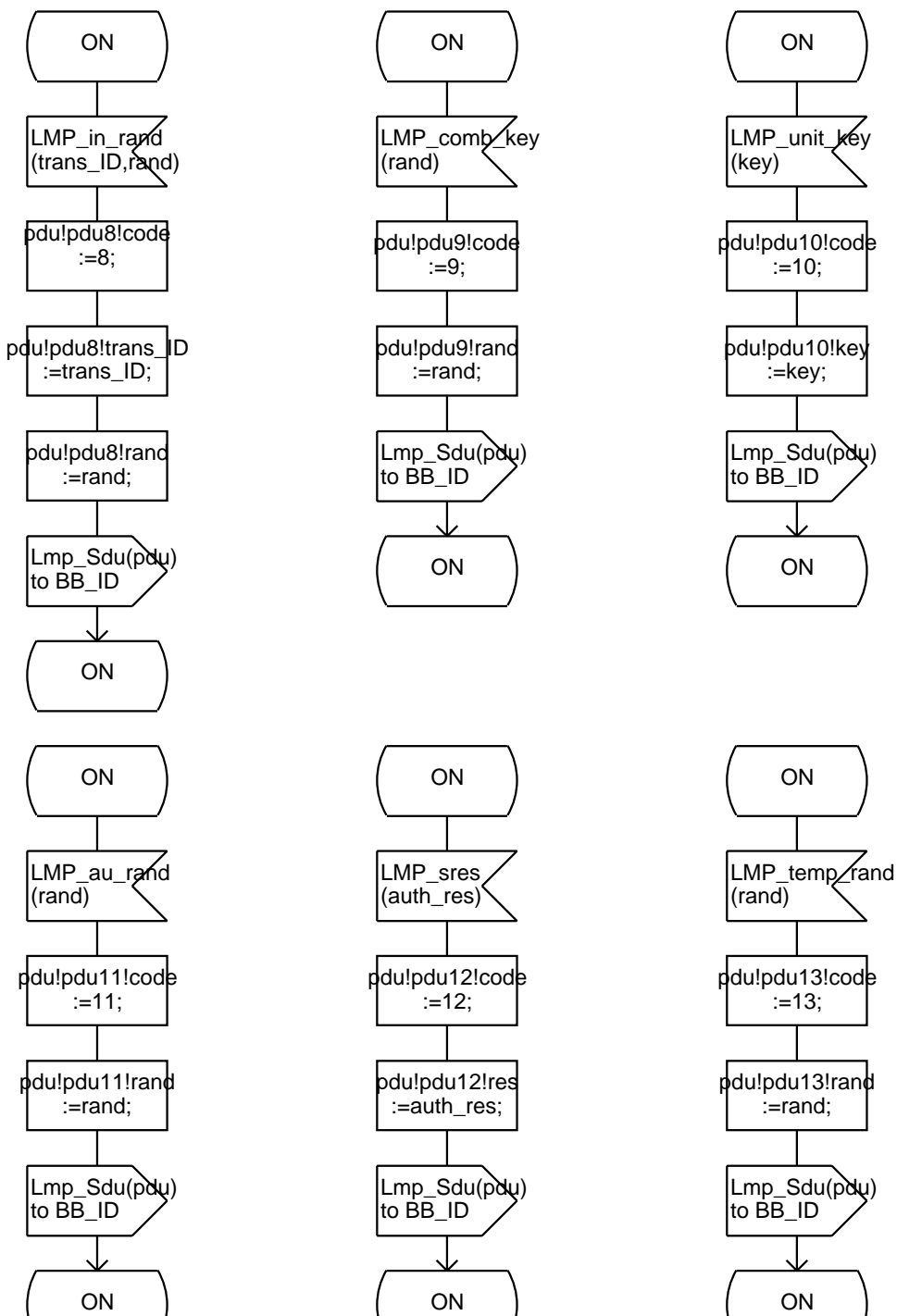




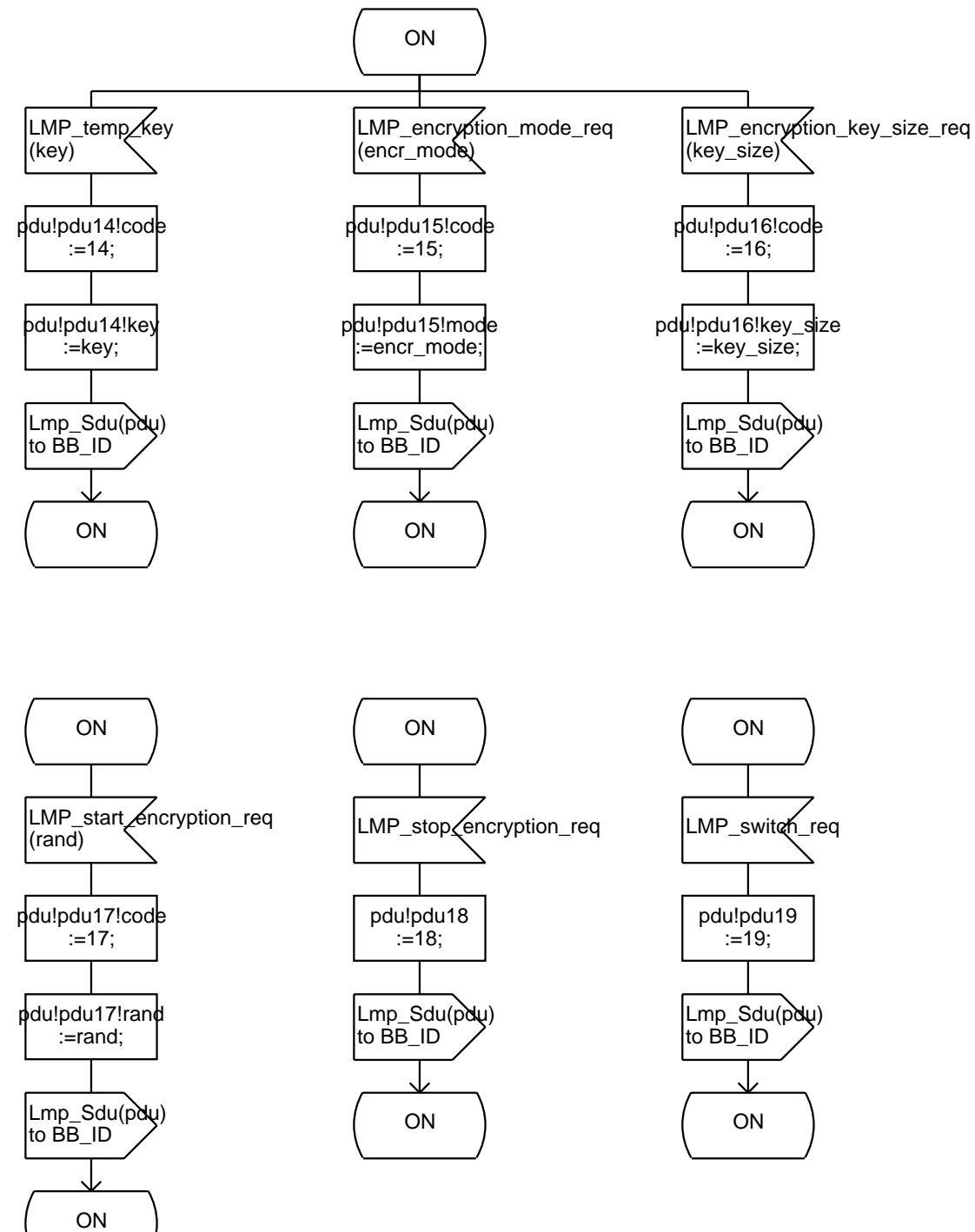


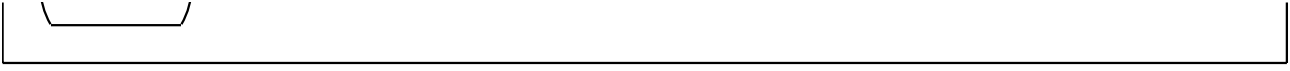


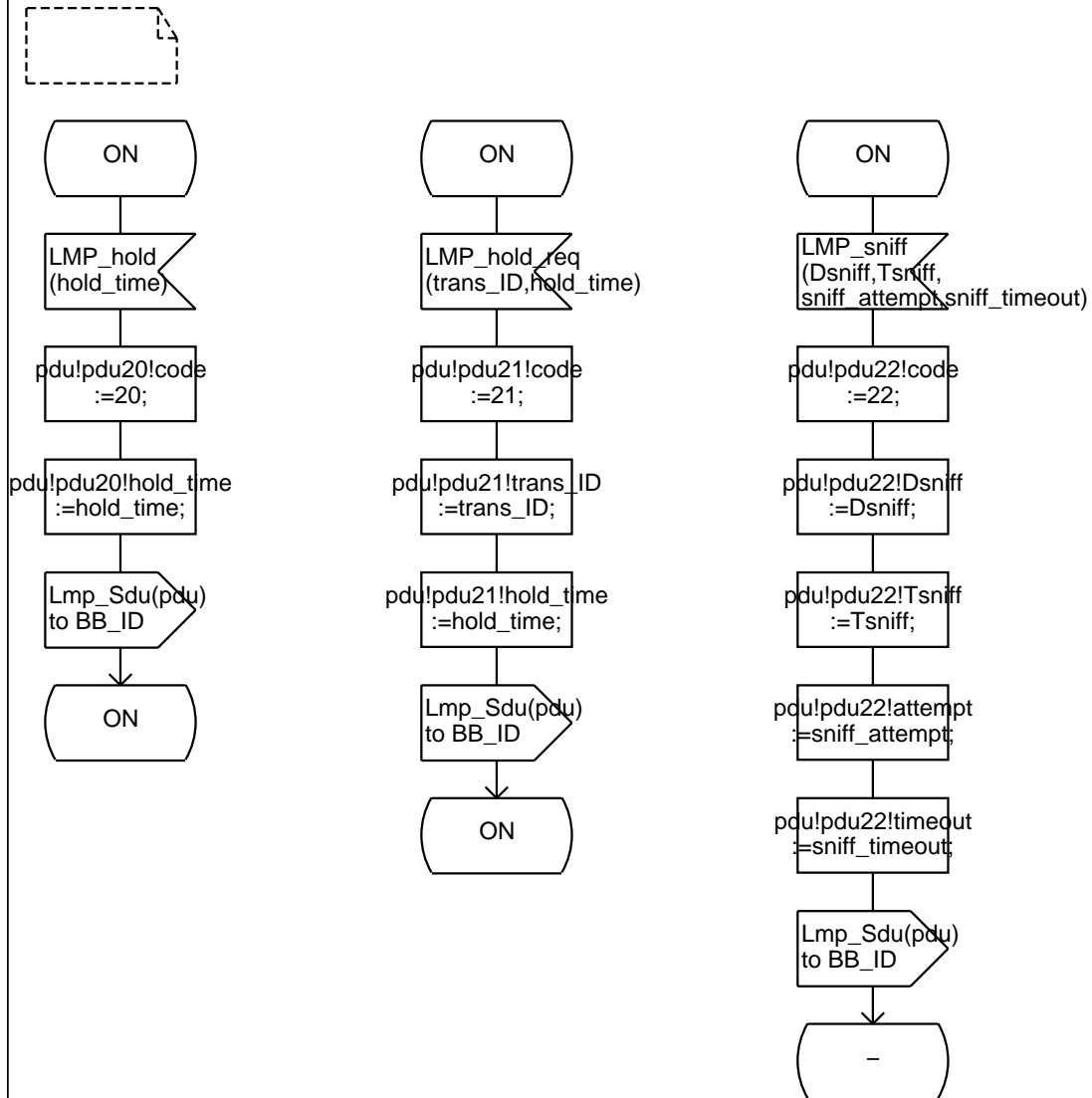


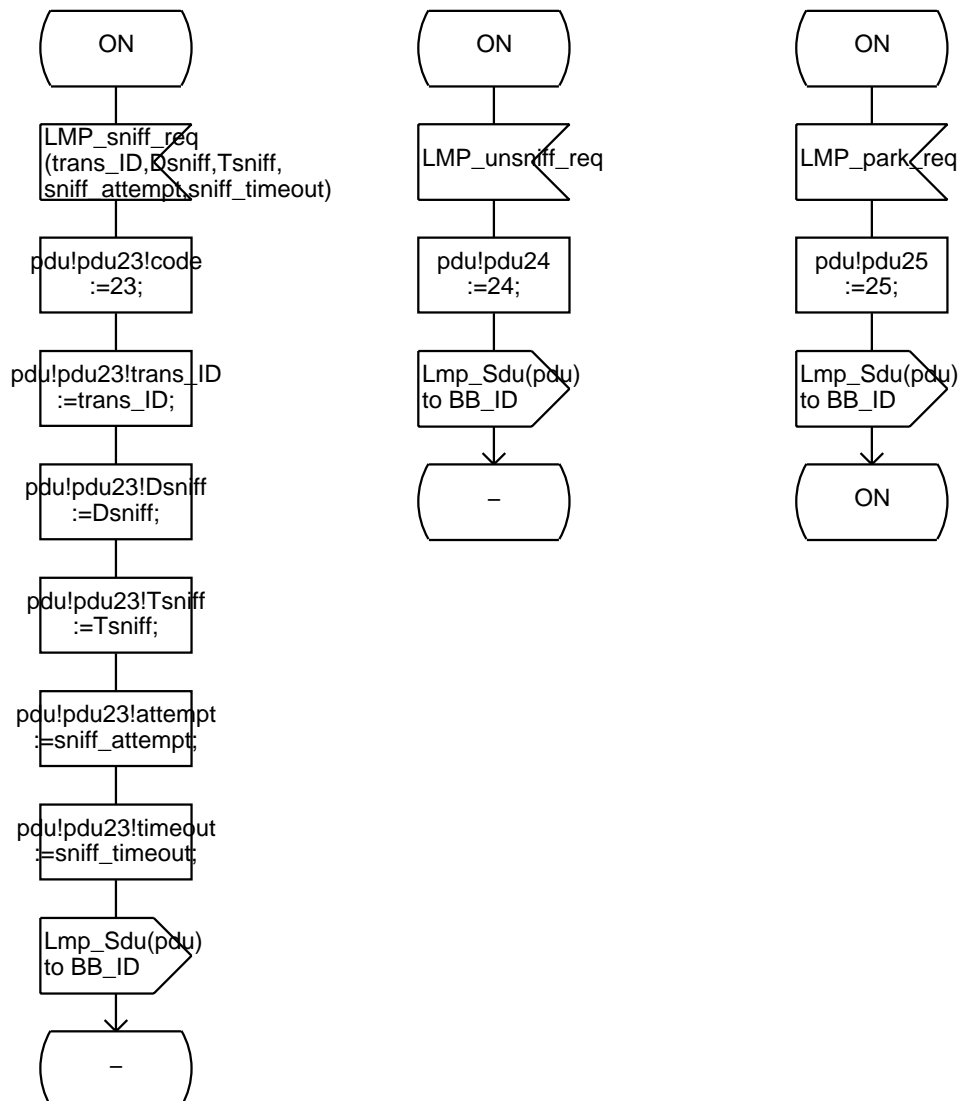


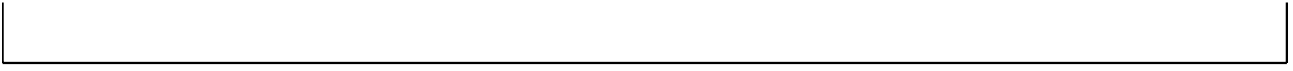


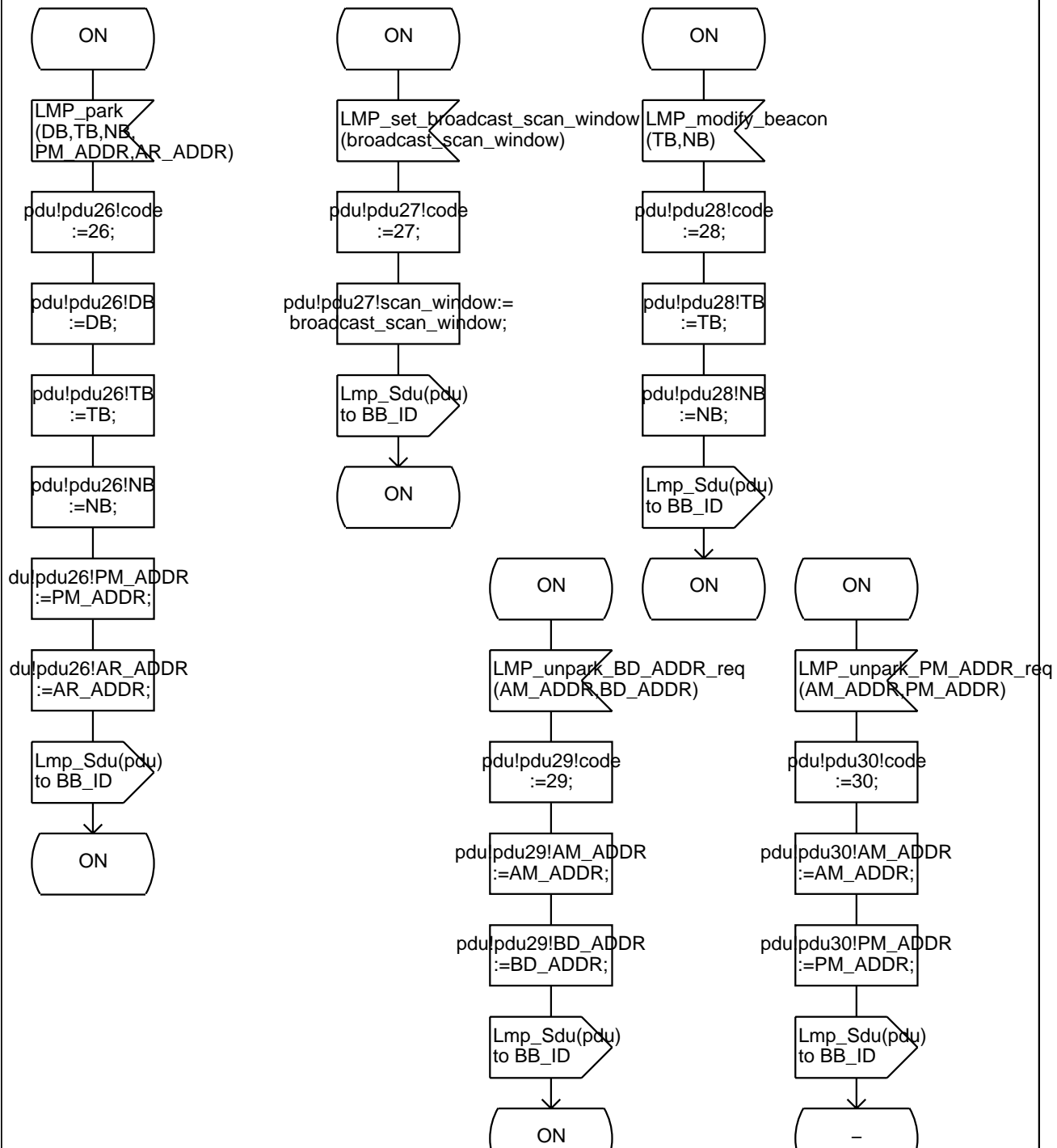


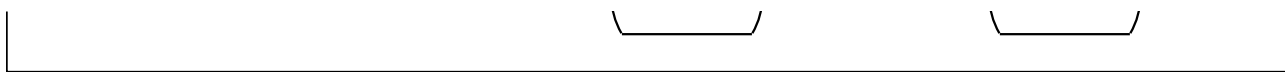


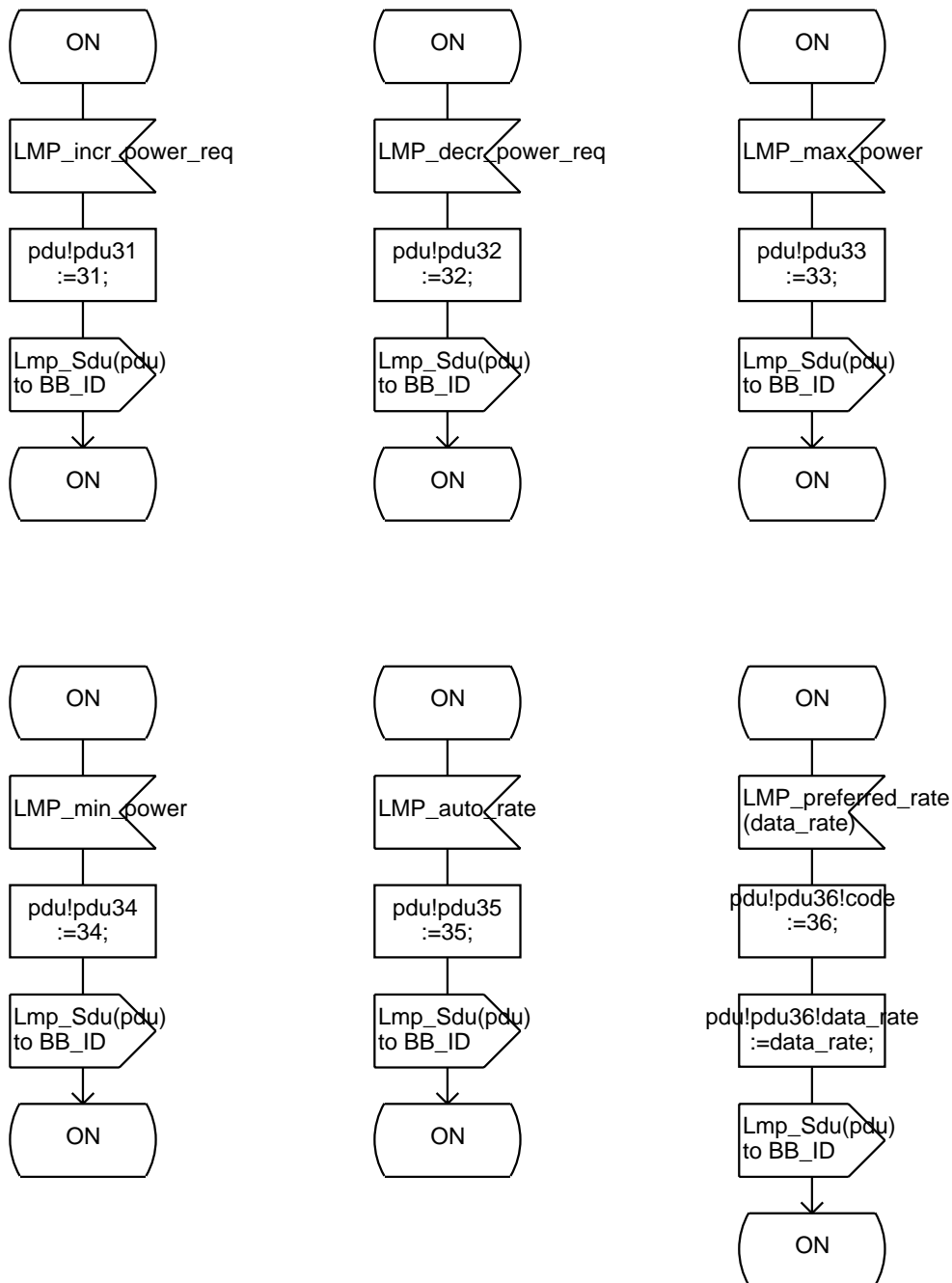


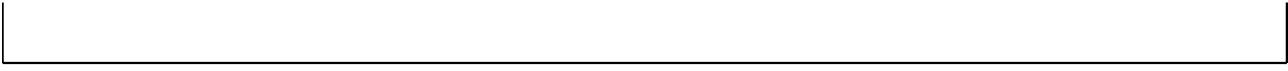


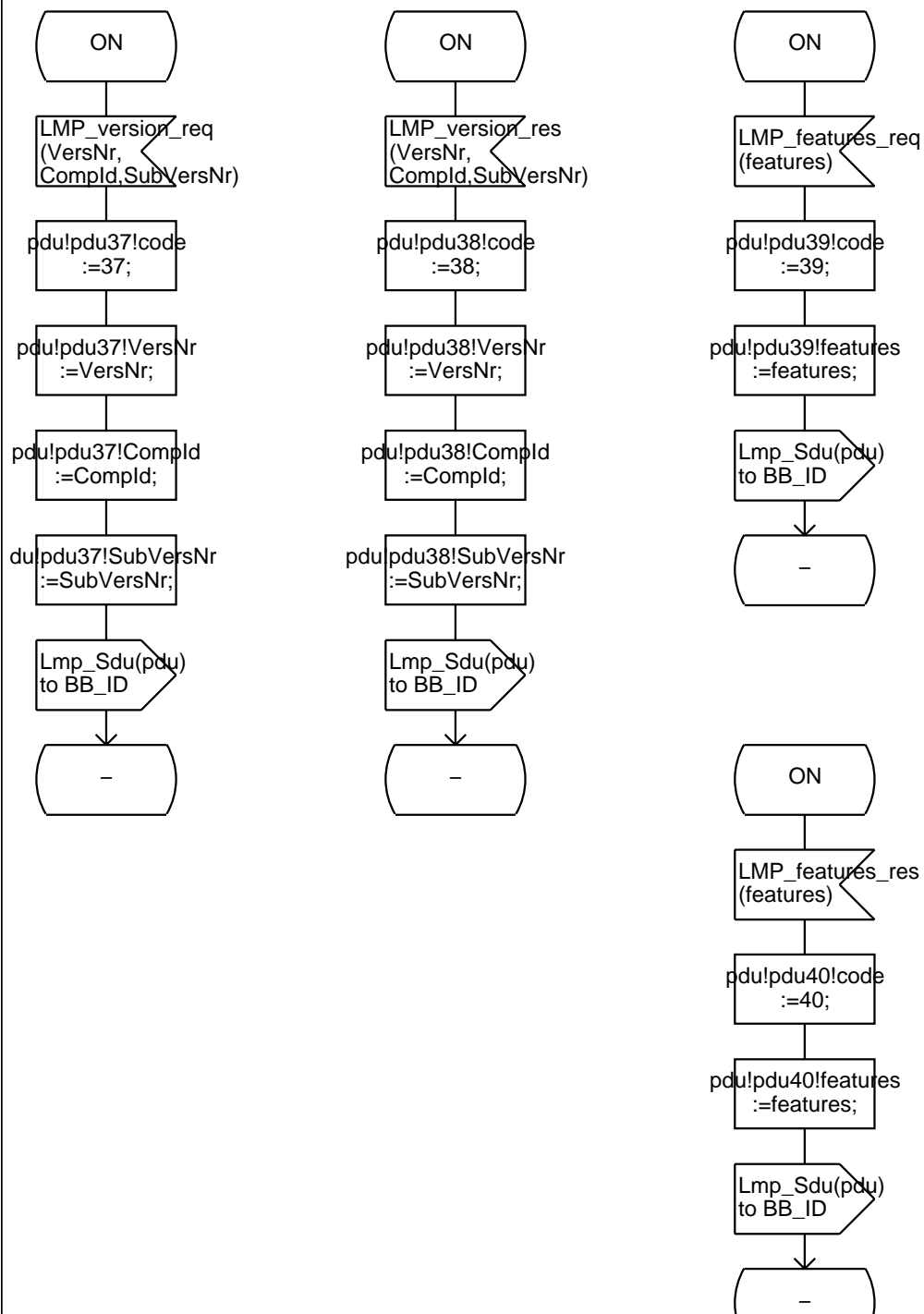




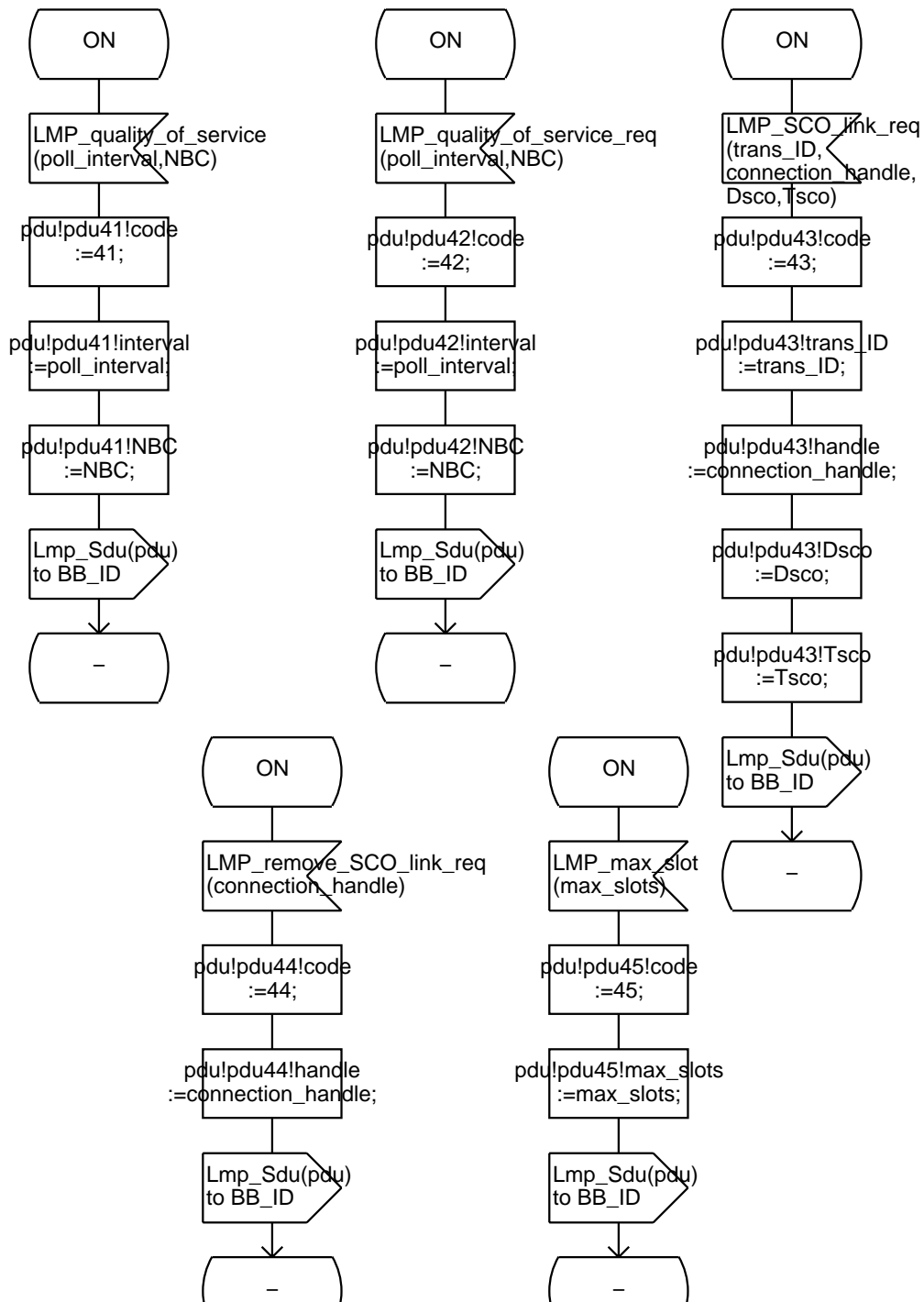




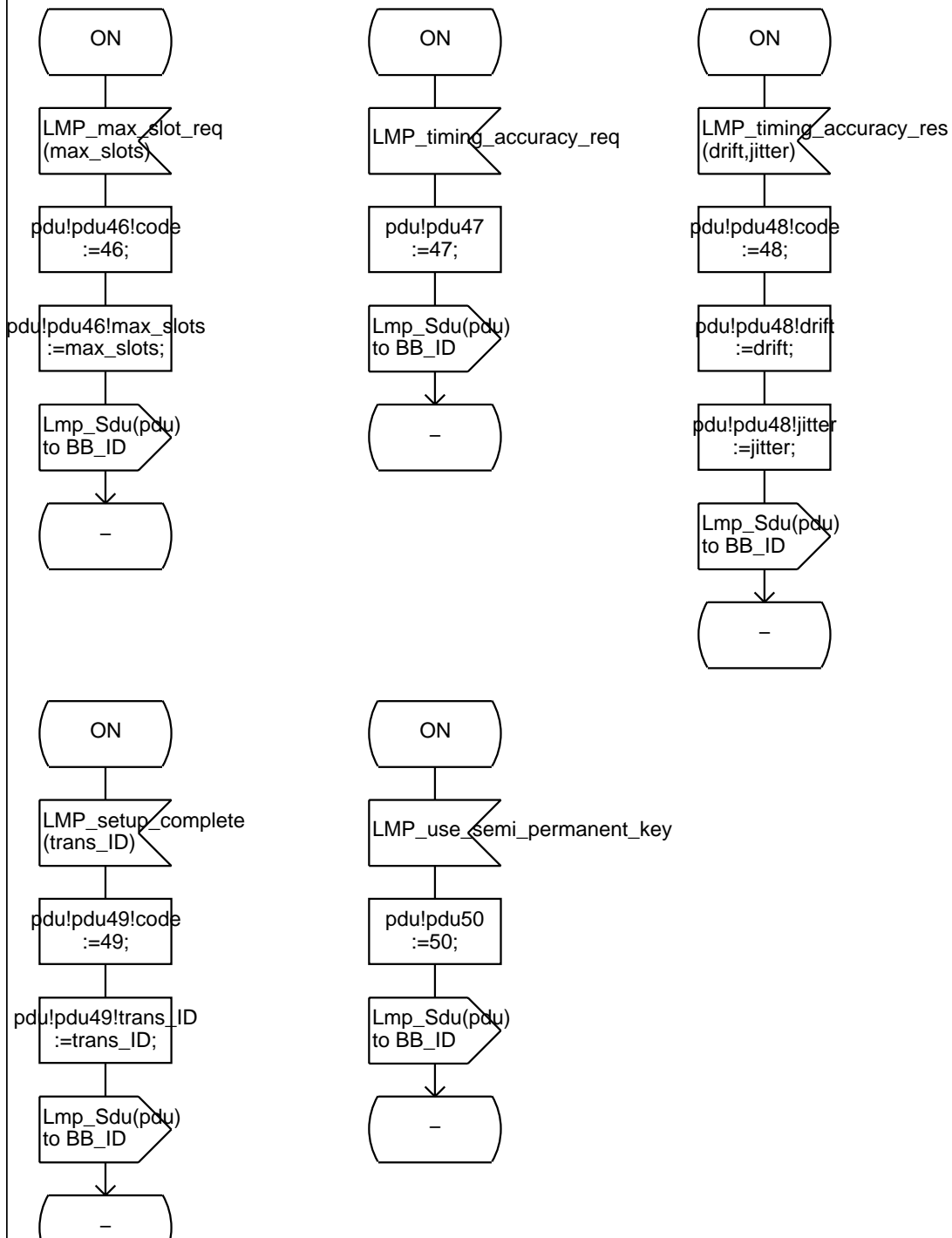


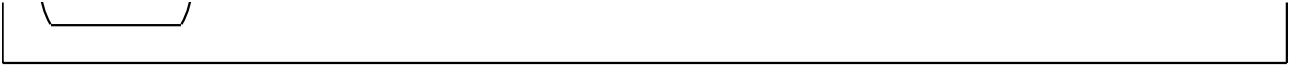














ON

LMP_host_connection_req

pdu!pdu51
:=51;

Lmp_Sdu(pdu)
to BB_ID

ON

ON

LMP_slot_offset
(slot_offset
BD_ADDR)

pdu!pdu52!code
:=52;

pdu!pdu52!offset
:=slot_offset;

pdu!pdu52!BD_ADDR
:=BD_ADDR;

Lmp_Sdu(pdu)
to BB_ID

ON

ON

LMP_page_mode_req
(paging_scheme)

pdu!pdu53!code
:=53;

pdu!pdu53!paging_scheme
:=paging_scheme;

Lmp_Sdu(pdu)
to BB_ID

ON

ON

LMP_page_scan_mode_req
(paging_scheme)

pdu!pdu54!code
:=54;

pdu!pdu54!paging_scheme
:=paging_scheme;

Lmp_Sdu(pdu)
to BB_ID

ON

ON

LMP_supervision_timeout
(super_timeout)

pdu!pdu55!code
:=55;

pdu!pdu55!timeout
:=super_timeout;

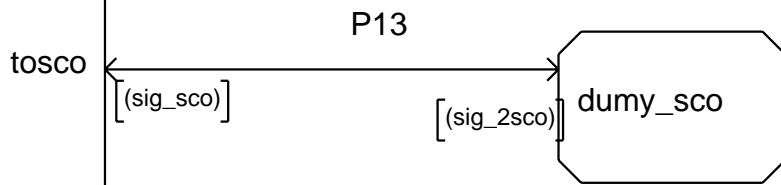
Lmp_Sdu(pdu)
to BB_ID

ON



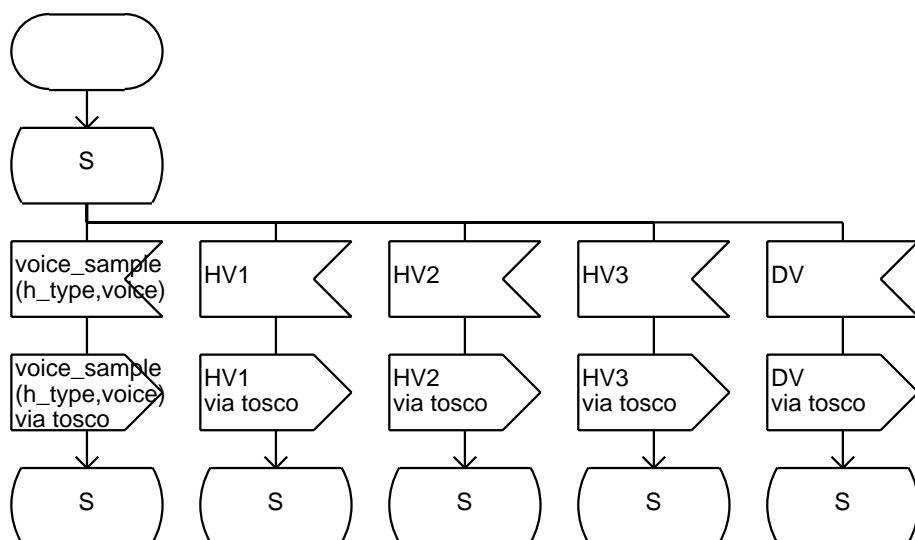
block sco

1(1)

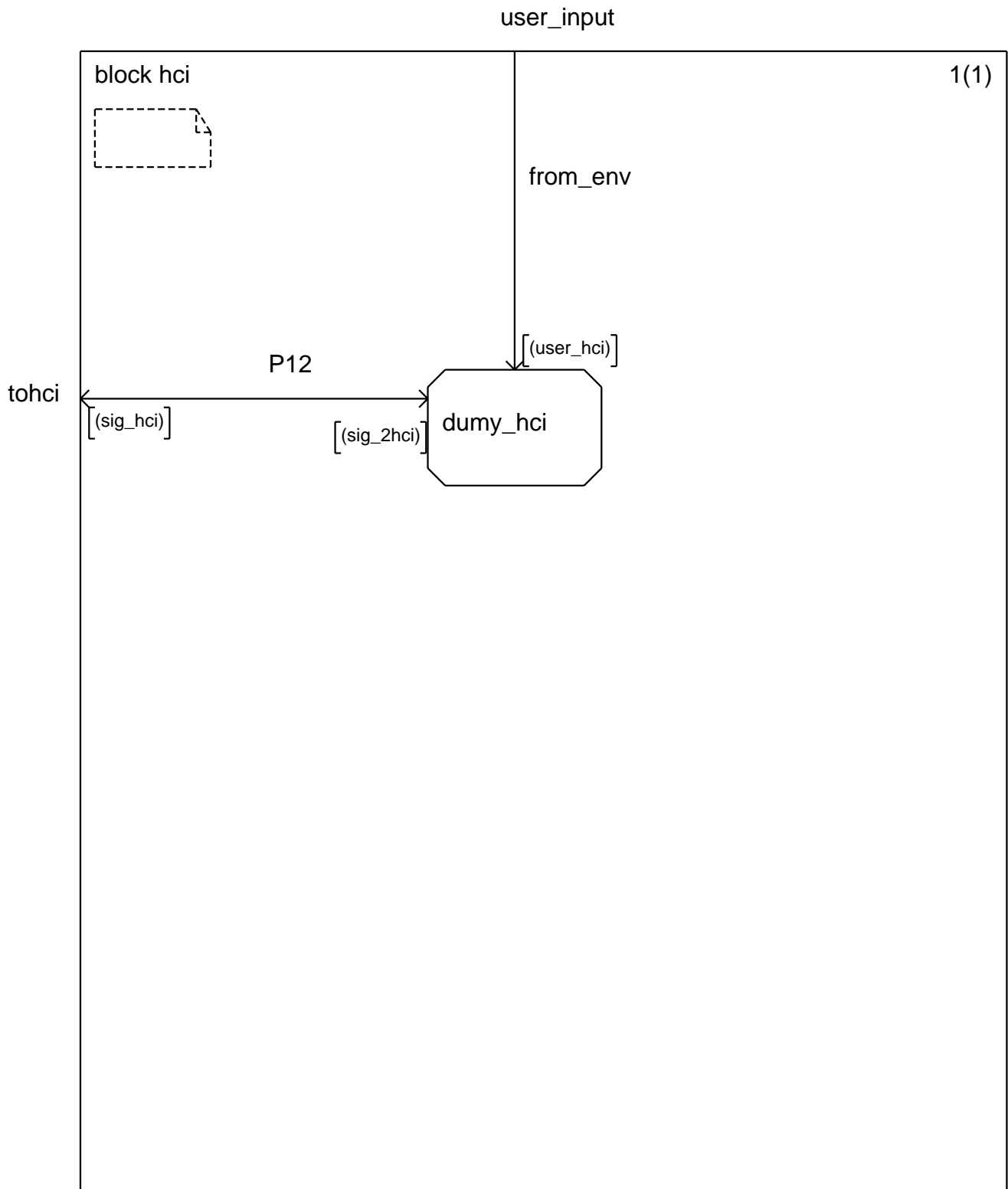




dcl
h_type, voice integer;





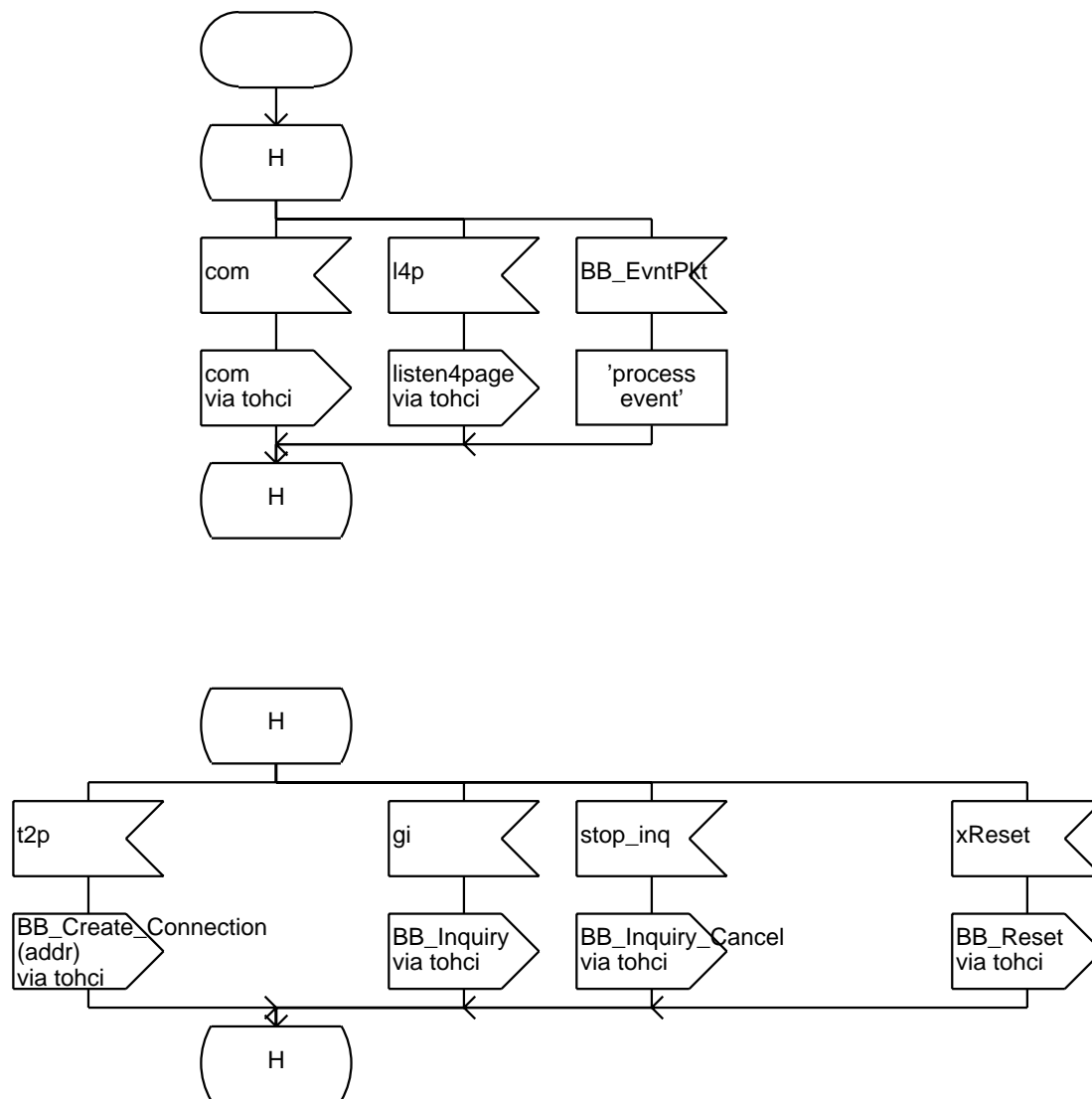


process dummy_hci


1(1)




DCL
addr Address_type;





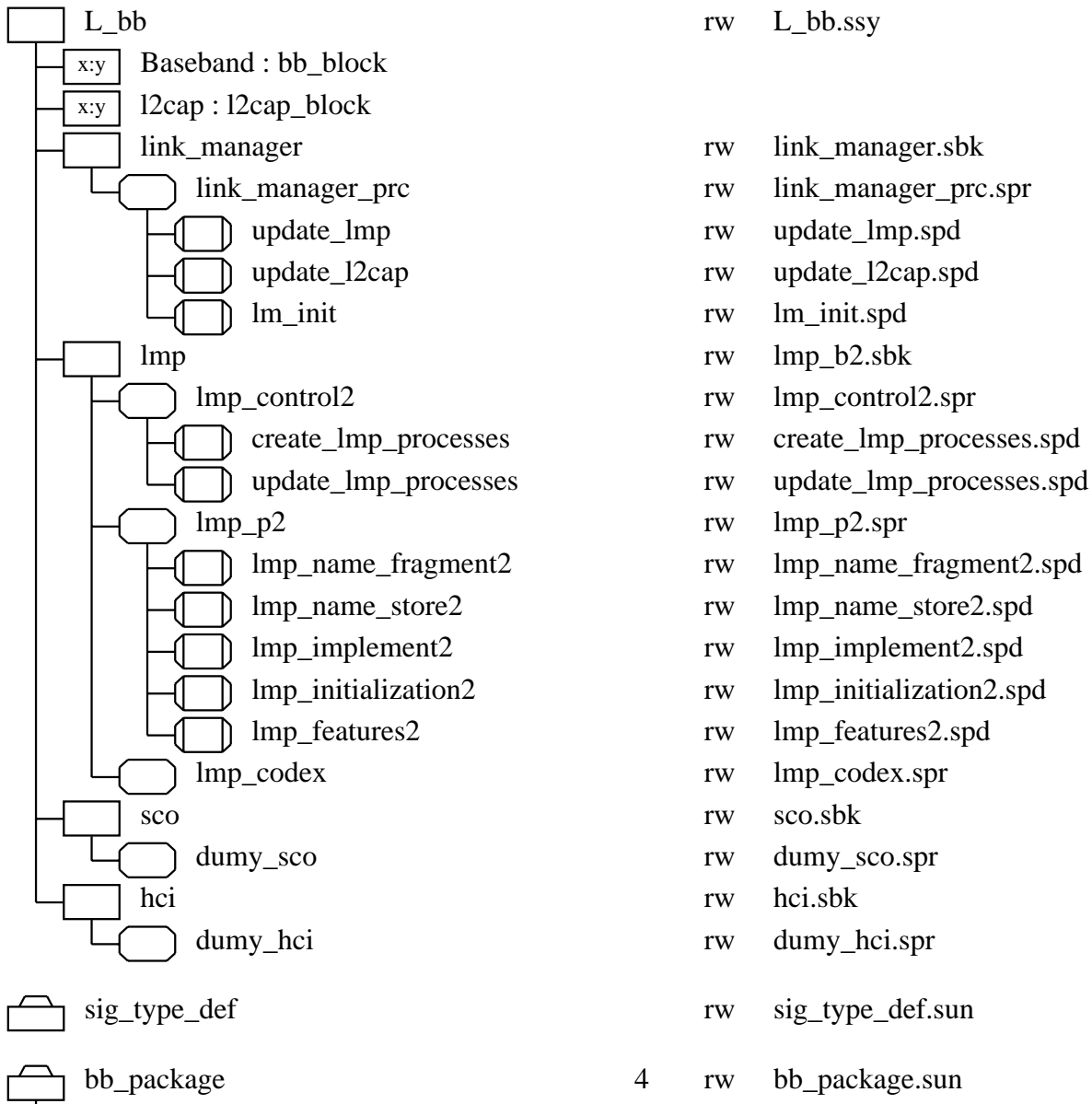
 rw /import/fs1/home/disk1/cypher/BLUETOOTH/new_bb/untitled.sdt




































 rw /import/fs1/home/disk1/cypher/BLUETOOTH/new_bb/

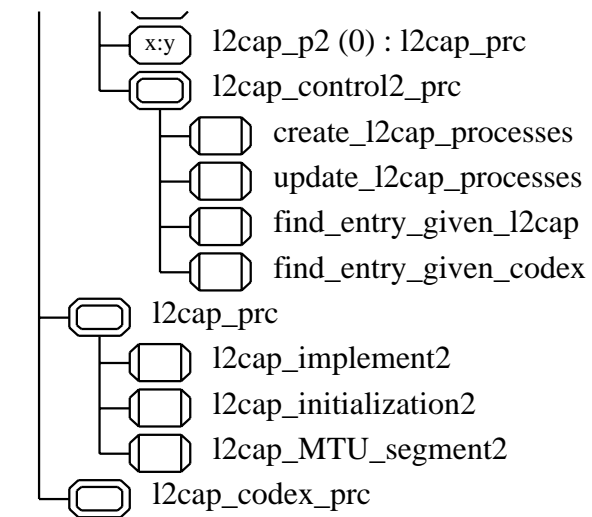
— Analysis Model

— Used Files

— new_bb



	bb_block	6	rw	bb_block.sbt
	ACL_entity (0) : ACL_prc			
	bb_controller (1,1) : bb_controller_prc			
	bb_phy (1,1) : bb_phy_prc			
	bb_controller_prc	8	rw	bb_controller_prc.spt
	activate_receiver	57	rw	activate_receiver.spd
	allocate_AM_ADDR	59	rw	allocate_AM_ADDR.spd
	change2master_channel	61	rw	change2master_channel.spd
	change2master_clock	63	rw	change2master_clock.spd
	clear_am_array	65	rw	clear_am_array.spd
	check_ACLs	67	rw	check_ACLs.spd
	clear_bt_array	69	rw	clear_bt_array.spd
	find_Pid_index	71	rw	find_Pid_index.spd
	implementation	73	rw	implementation.spd
	initialize	75	rw	initialize.spd
	RANDOM	77	rw	RANDOM.spd
	save_info	79	rw	save_info.spd
	select_next_hop	81	rw	select_next_hop.spd
	select_scan_frequency	83	rw	select_scan_frequency.spd
	terminate_ACL	85	rw	terminate_ACL.spd
	find_addr_given_pid1	87	rw	find_addr_given_pid1.spd
	bb_phy_prc	89	rw	bb_phy_prc.spt
	AM_ADDR_in_use	109	rw	AM_ADDR_in_use.spd
	determine_active_entity	111	rw	determine_active_entity.spd
	find_addr_given_pid	113	rw	find_addr_given_pid.spd
	init_main_phy	115	rw	init_main_phy.spd
	ACL_prc	117	rw	ACL_prc.spt
	impl_ACL	139	rw	impl_ACL.spd
	l2_code_ph	141	rw	L2_code_ph.spd
	LM_code_ph	143	rw	LM_code_ph.spd
	init_ACL	145	rw	init_ACL.spd
	l2cap_package		rw	l2cap_package.sun
	l2cap_block		rw	l2cap_block.sbt
	l2cap_codex (0) : l2cap_codex_prc			
	l2cap_control2 (1,1) : l2cap_control2_prc			



rw l2cap_control2_prc.spt
 rw create_l2cap_processes.spd
 rw update_l2cap_processes.spd
 rw find_entry_given_l2cap.spd
 rw find_entry_given_codex.spd
 rw l2cap_prc.spt
 rw l2cap_implement2.spd
 rw l2cap_initialization2.spd
 rw l2cap_MTU_segment2.spd
 rw l2cap_codex_prc.spt

— Other Documents

USE sig_type_def;

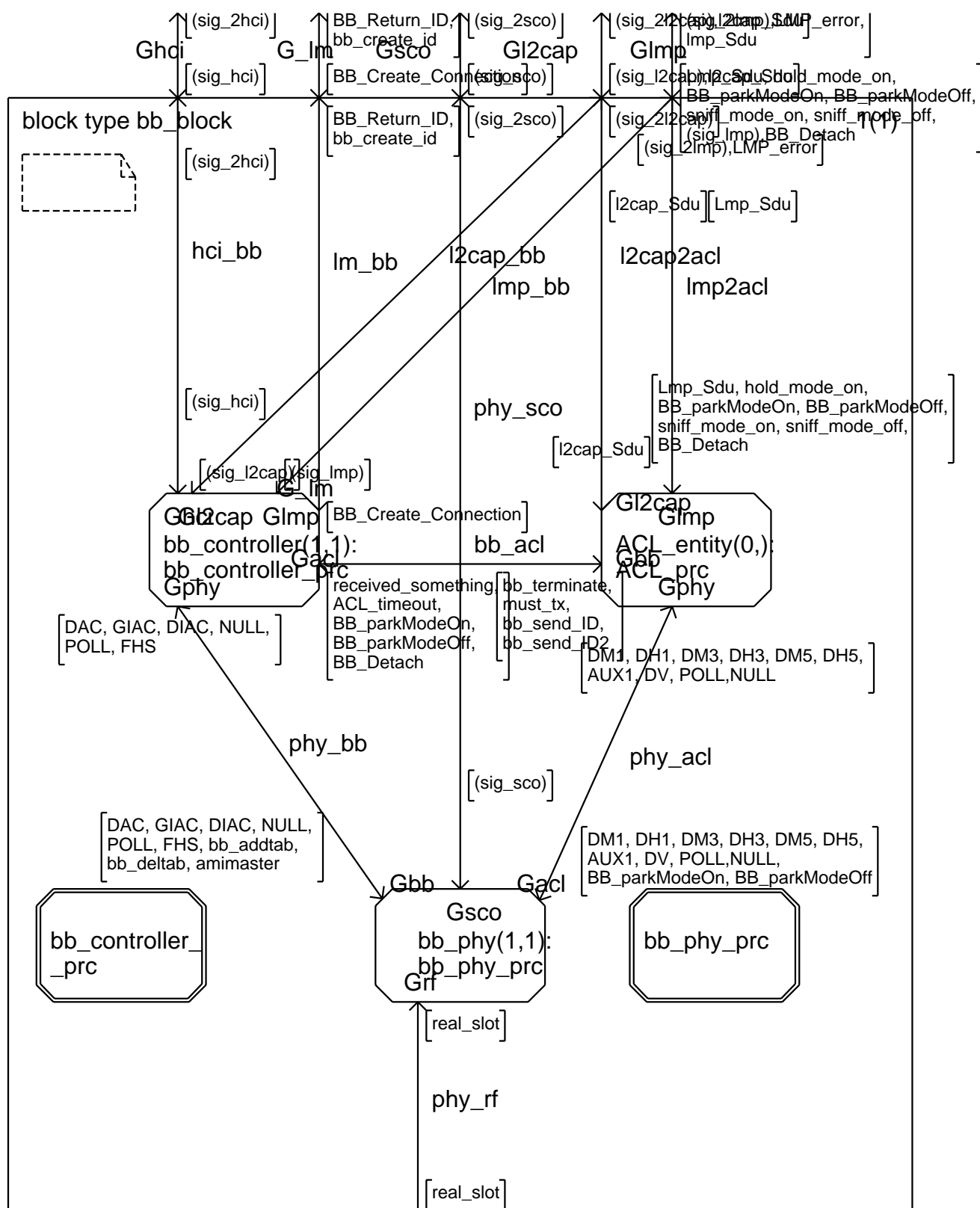
package bb_package

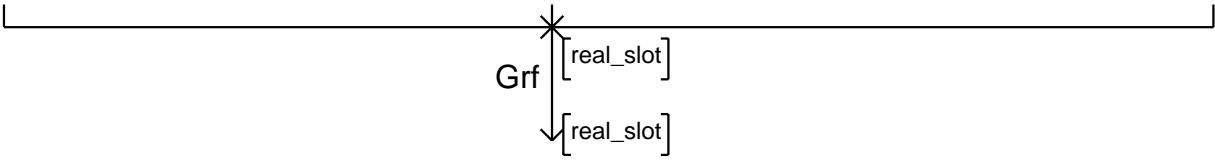
1(1)

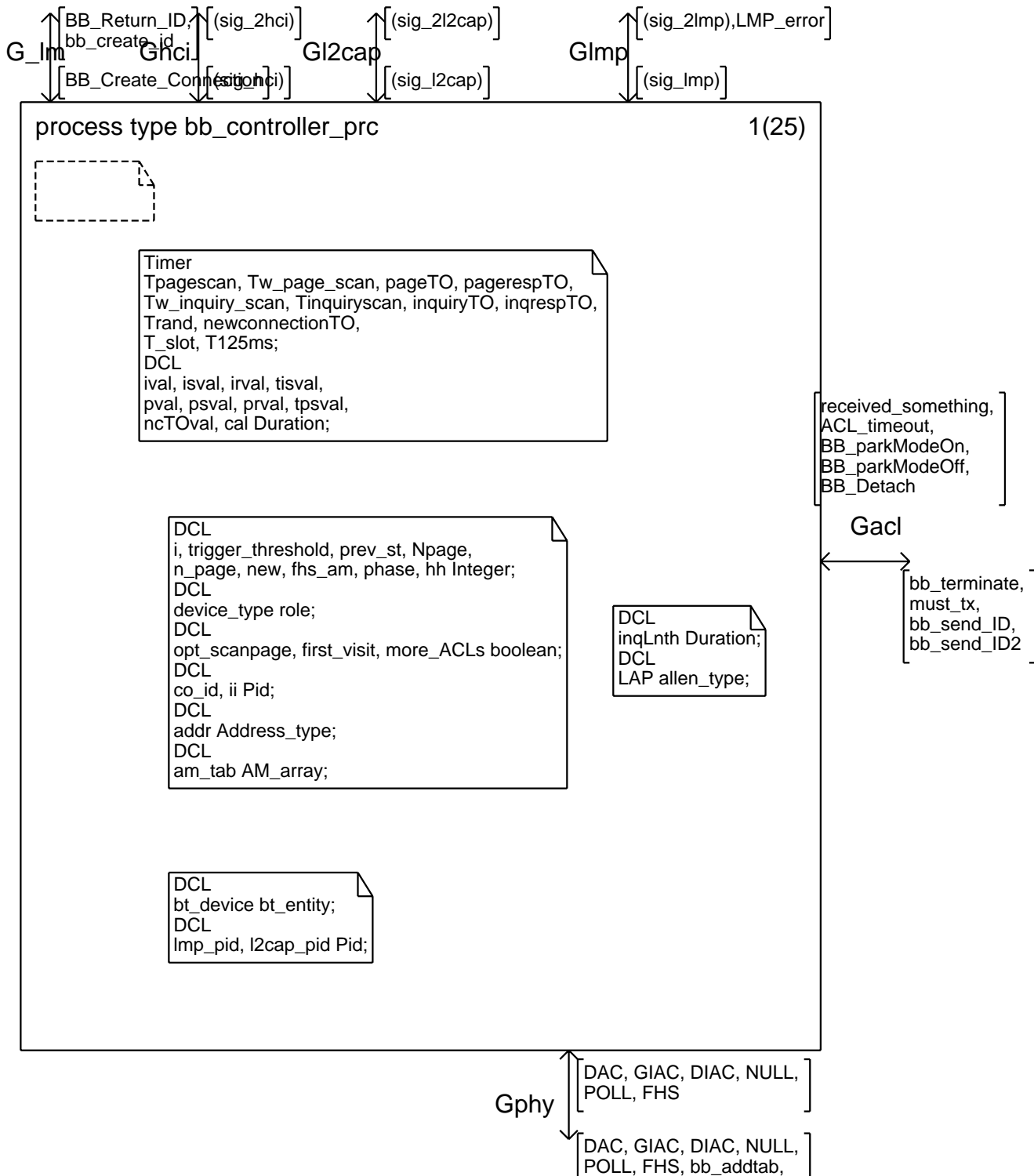


bb_block

ACL_prc







[bb_deltab, animaster]



activate
receiver

allocate
AM_ADDR

change2
master

channel
change2
master

clock
clear_am
array

check_
ACLs

clear_bt
array

find_Pid
index

imple_
mentation

initialize

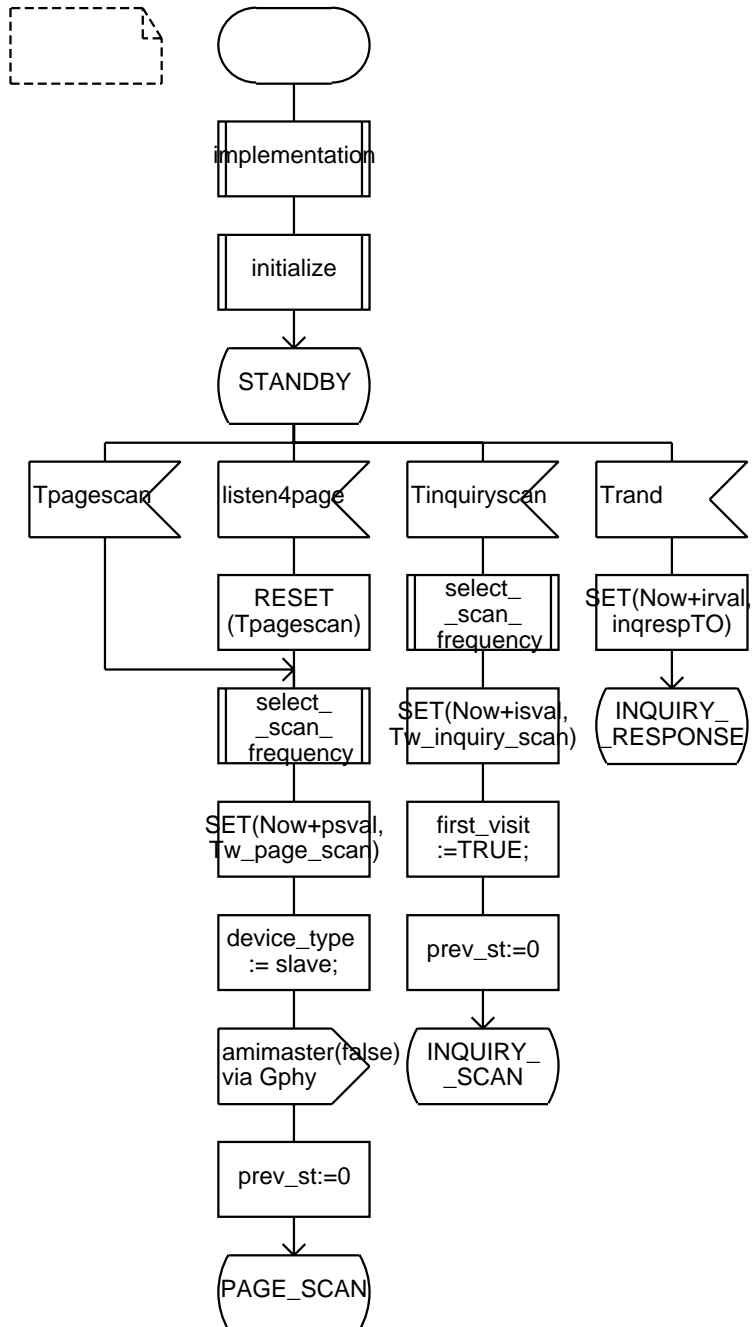
RANDOM

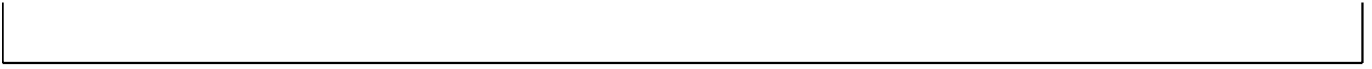
save_info

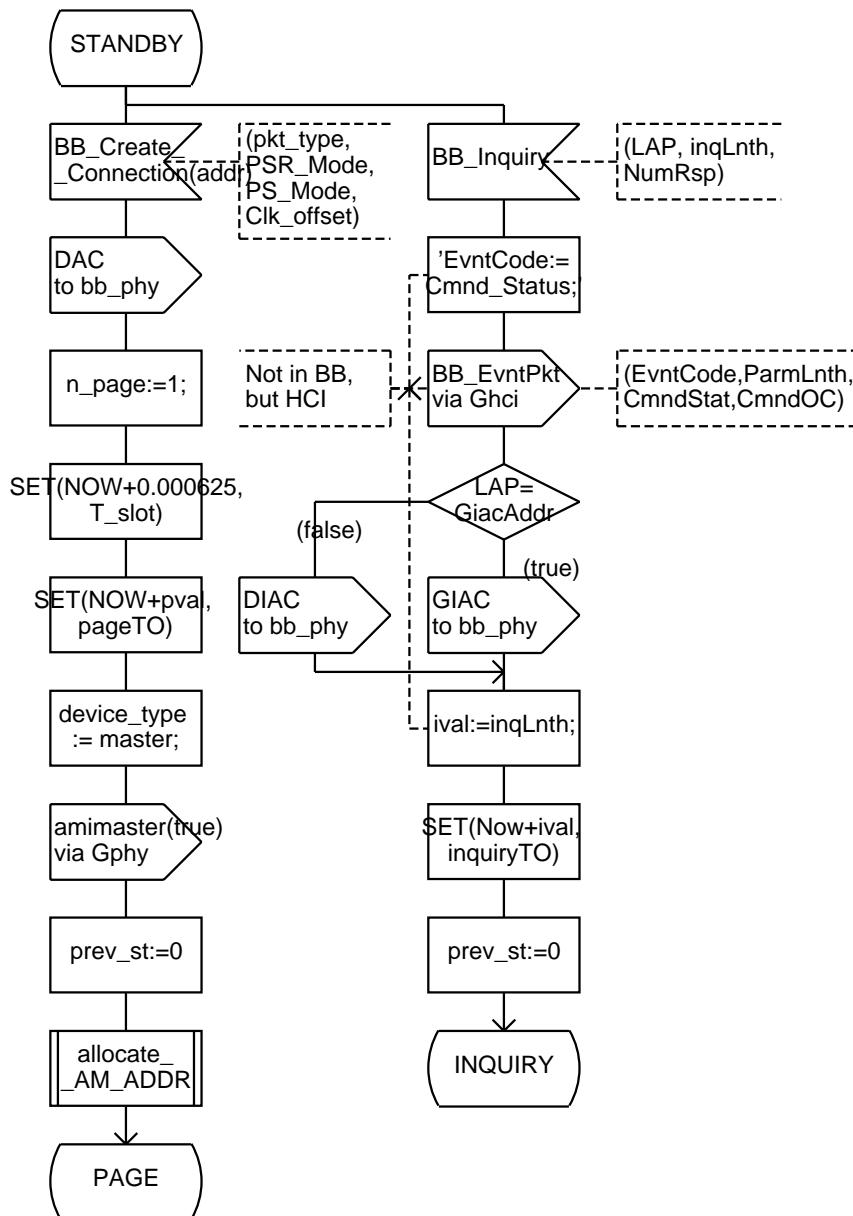
select_
next_hop

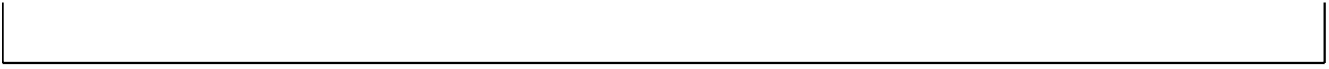
select_
scan_
frequency

terminate_
ACL



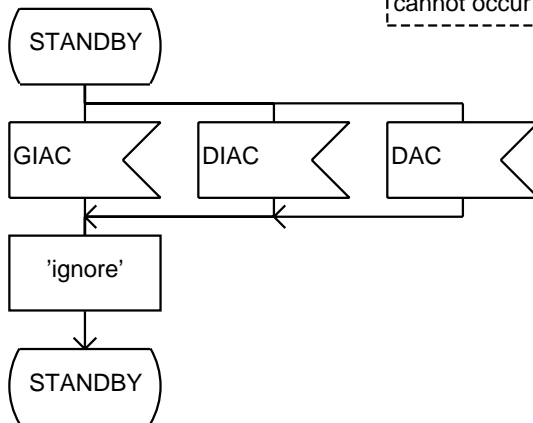


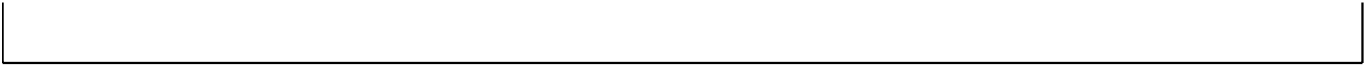


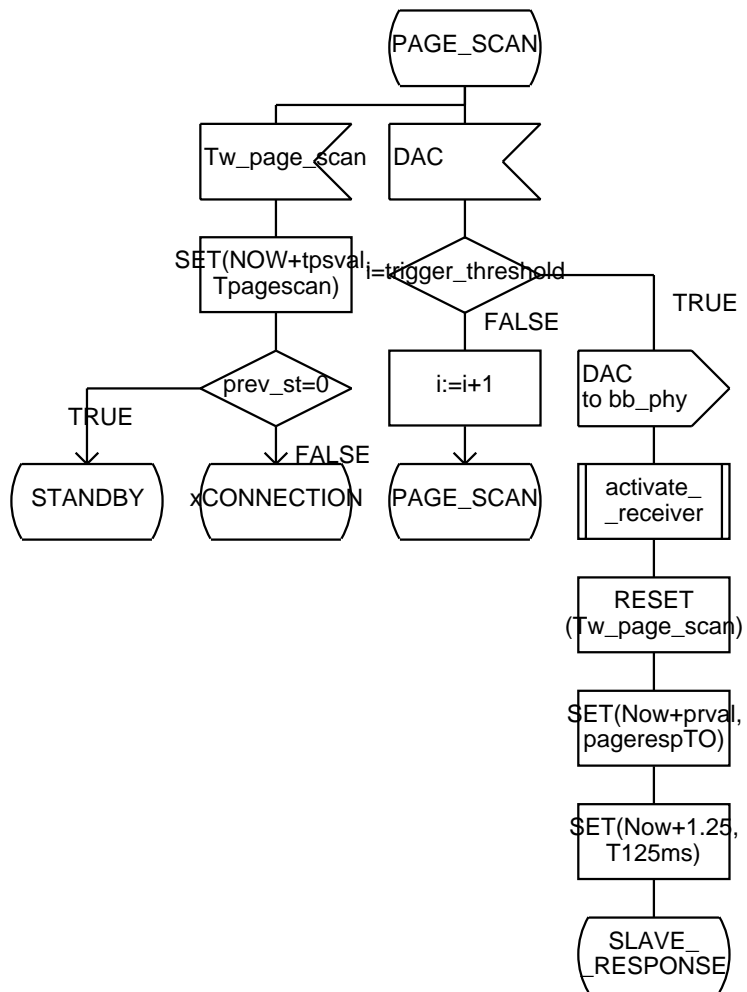


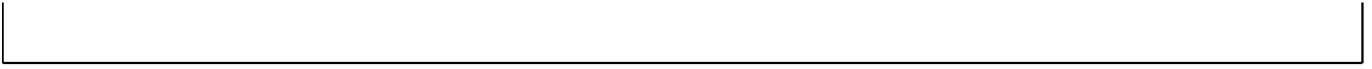


All signals received on this page are considered unexpected and were added to complete the protocol during verification to remove implicit consumption of these signals. A real implementation does not need this because these signals cannot occur while in this state.



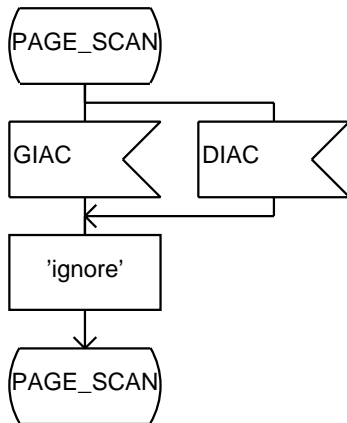


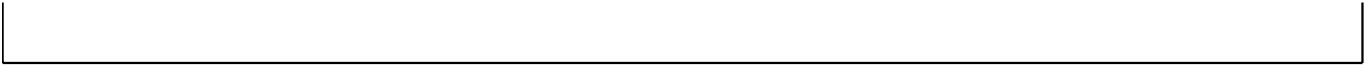


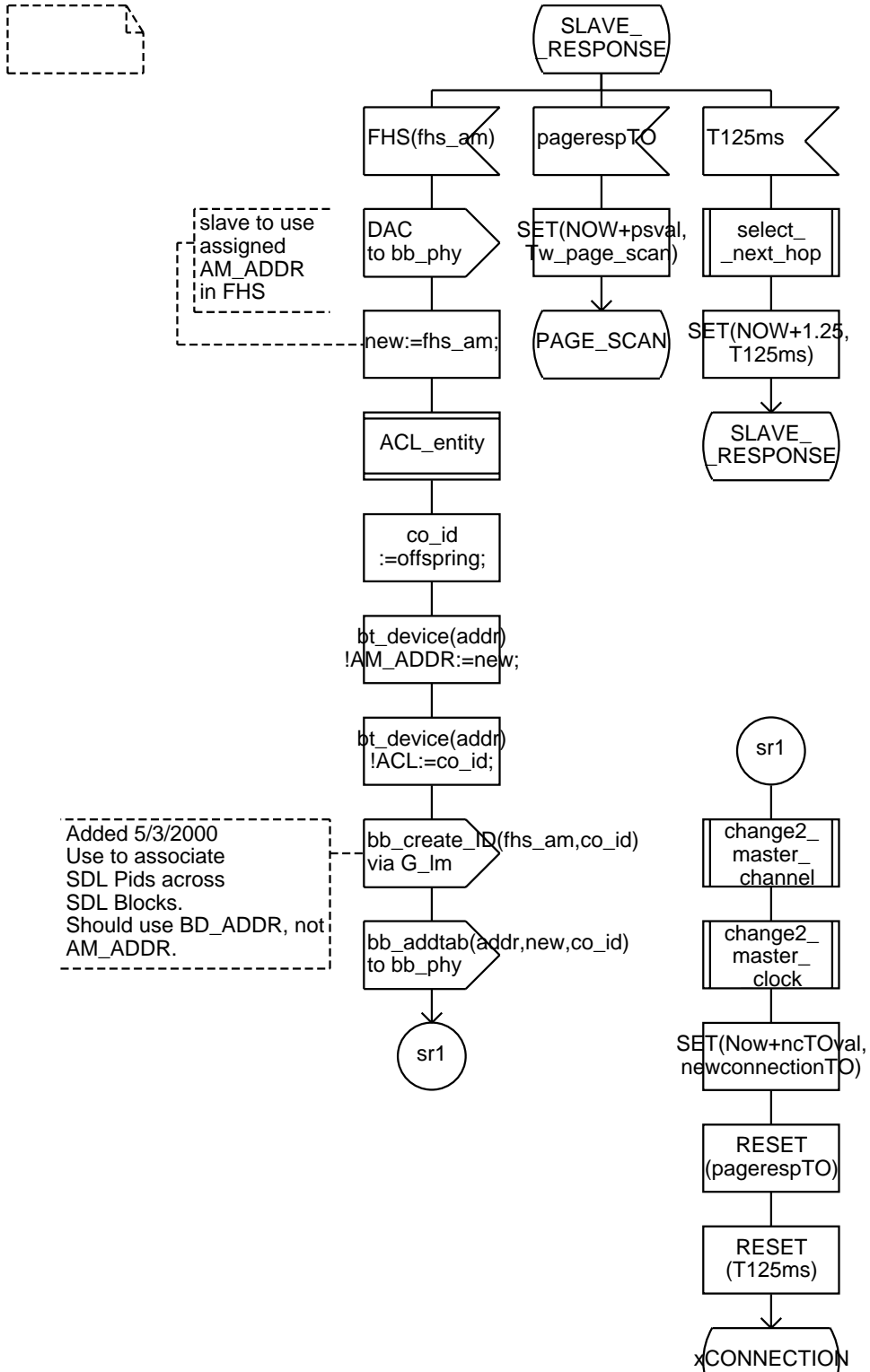




All signals received on this page are considered unexpected and were added to complete the protocol during verification to remove implicit consumption of these signals. A real implementation does not need this because these signals cannot occur while in this state.



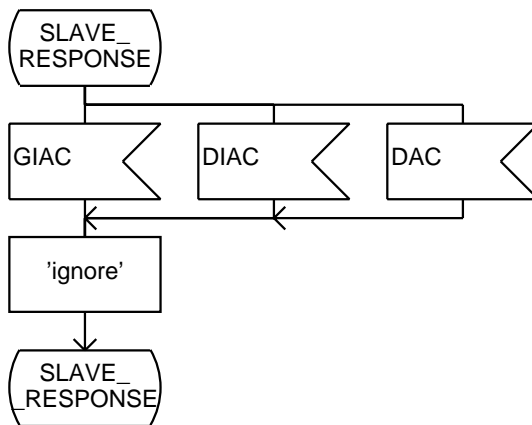


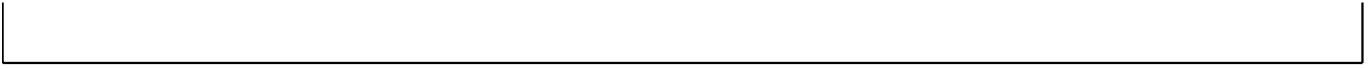


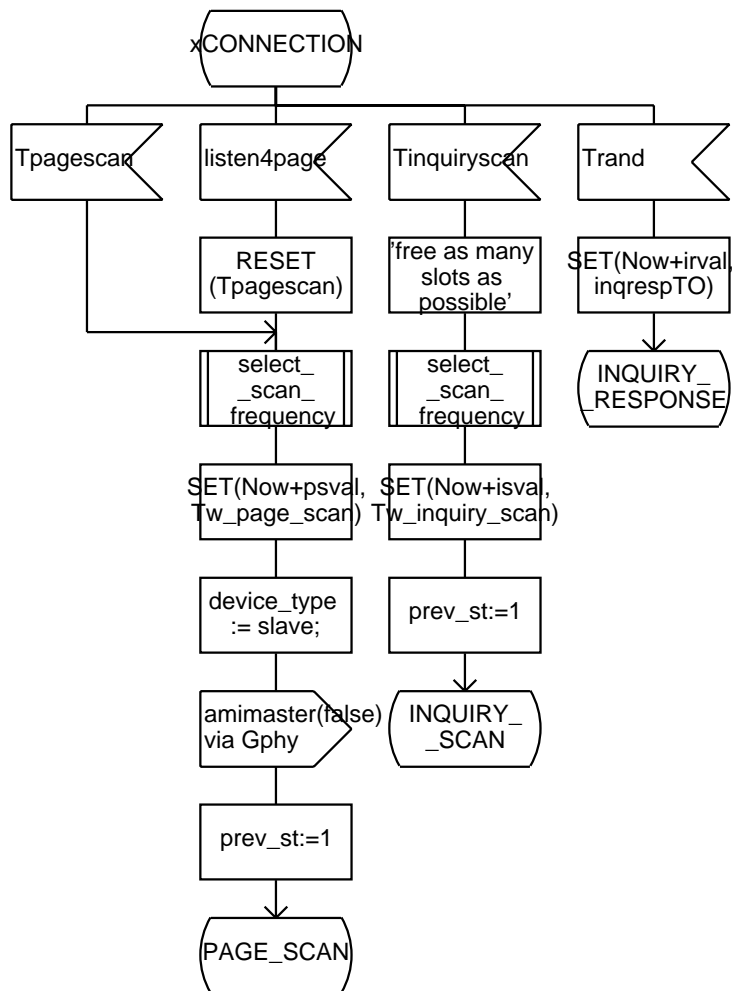


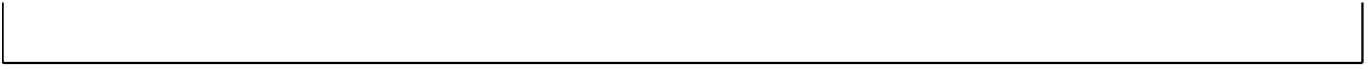


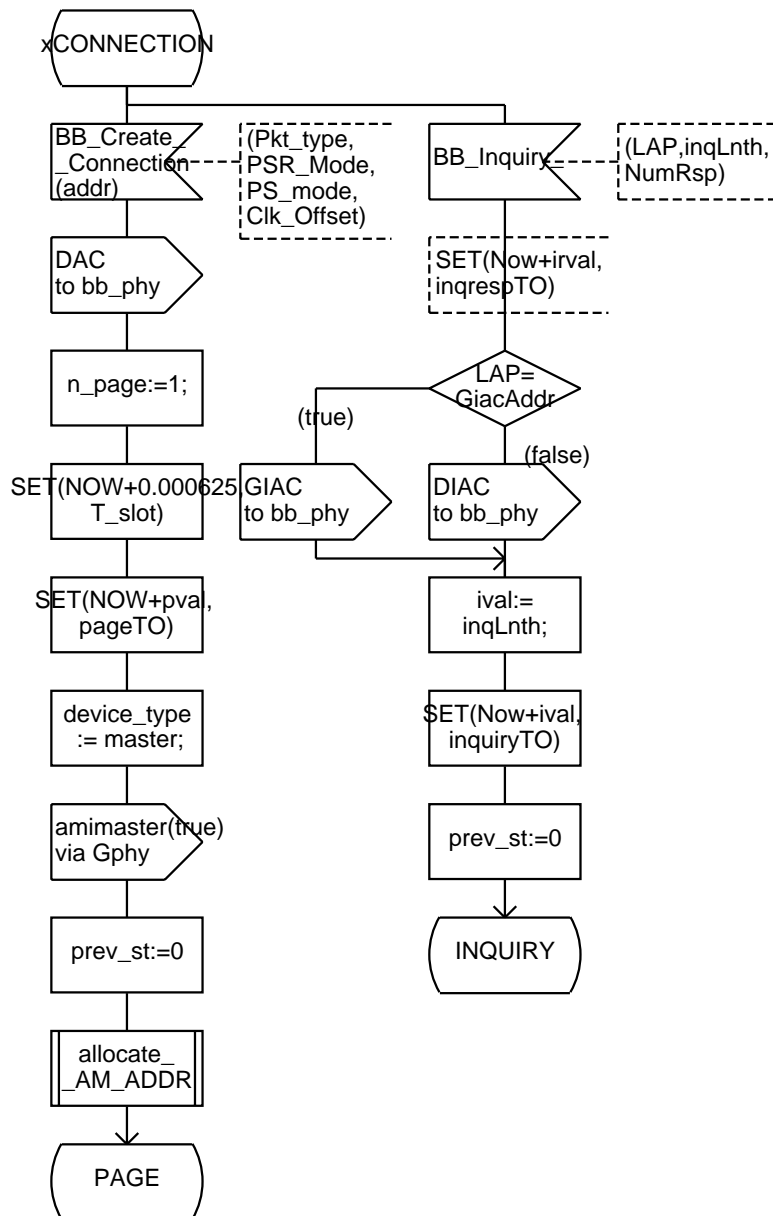
All signals received on this page are considered unexpected and were added to complete the protocol during verification to remove implicit consumption of these signals. A real implementation does not need this because these signals cannot occur while in this state.



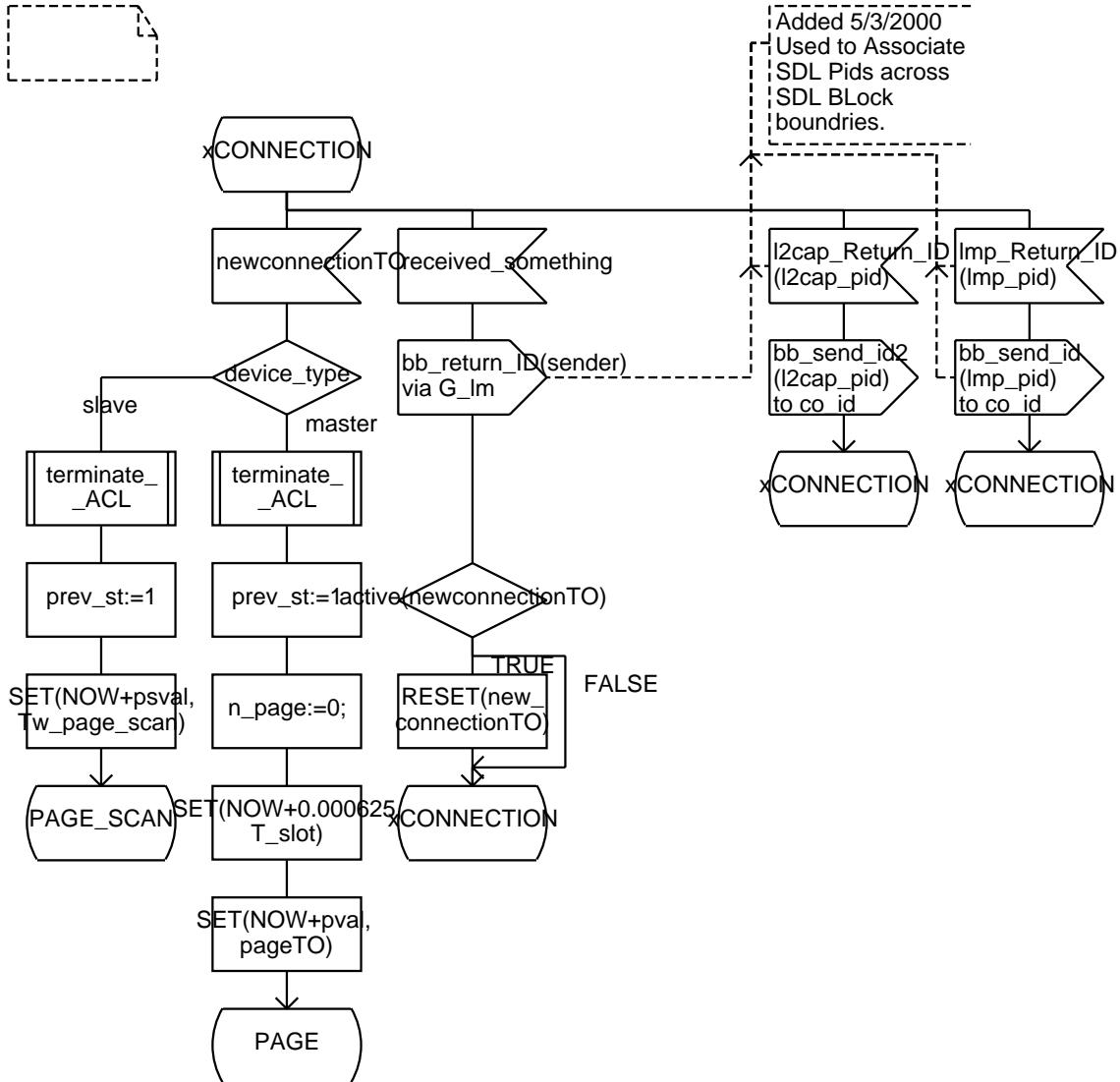


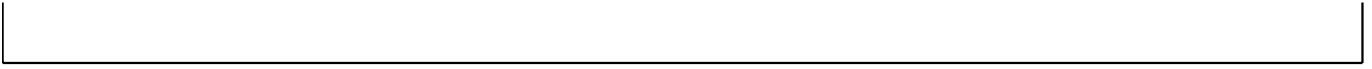


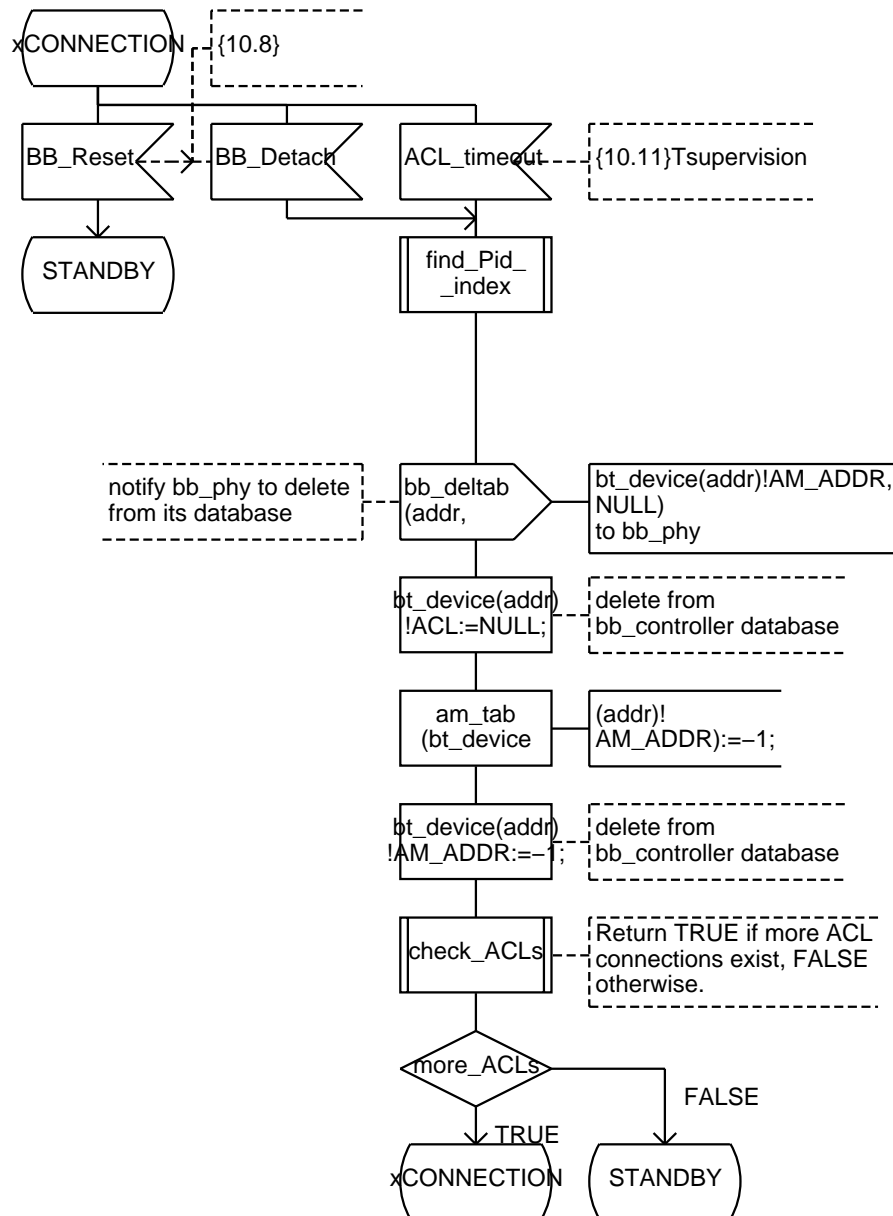


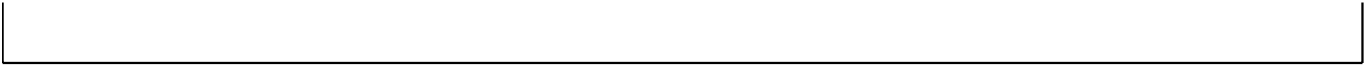








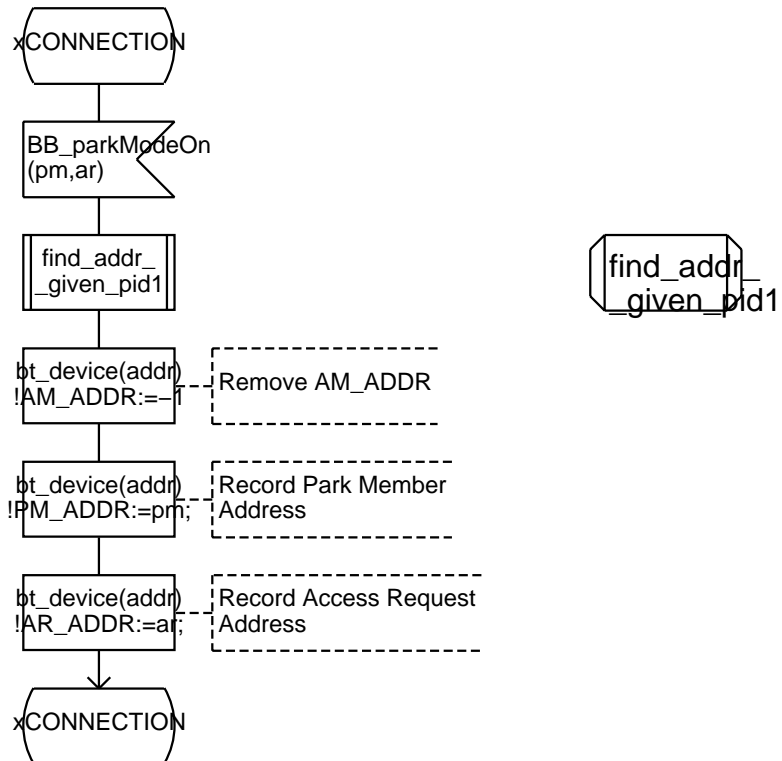


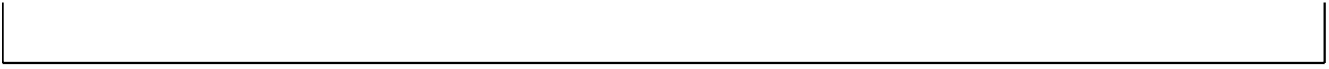




undefined signal to
coordinate parking of
device.

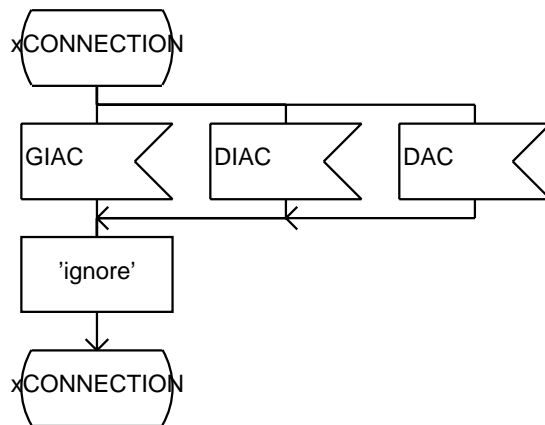
DCL
pm, ar Integer;

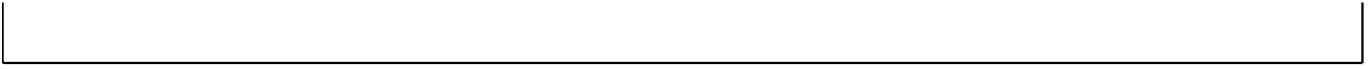


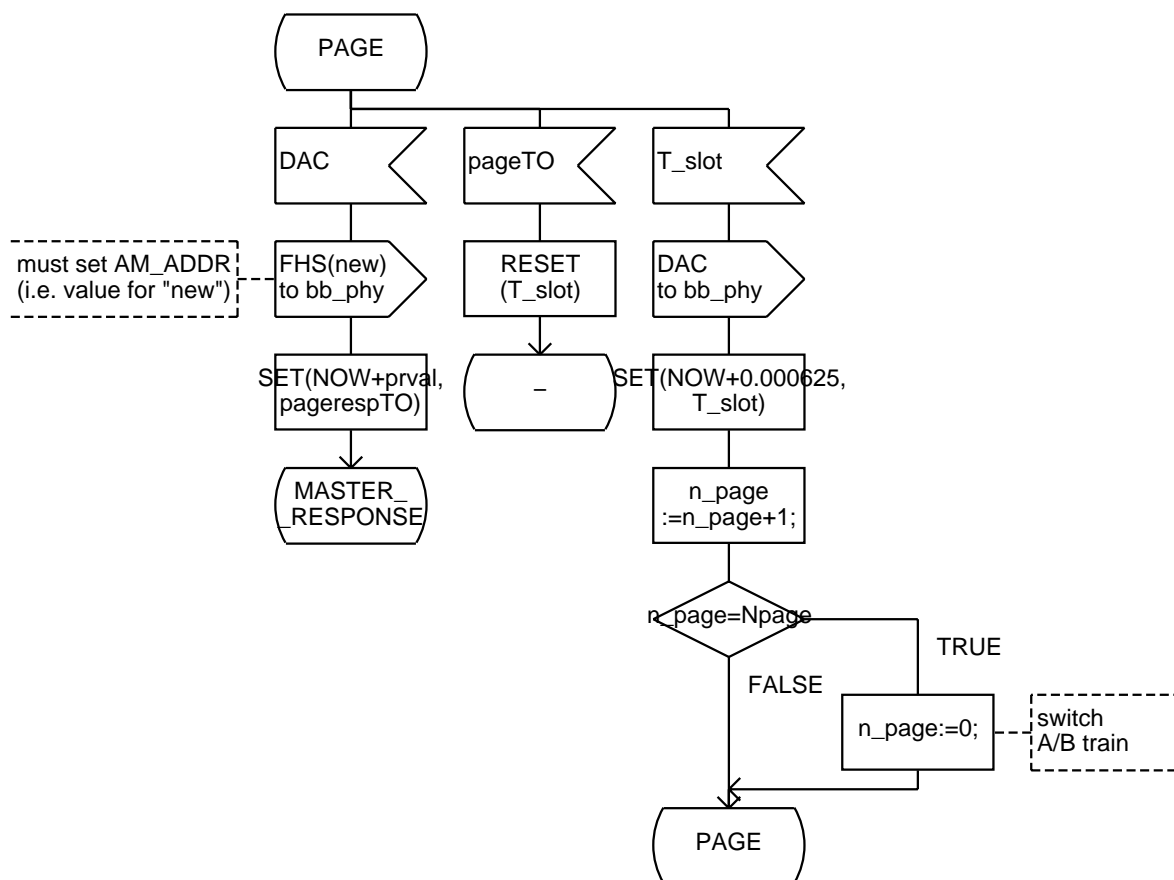


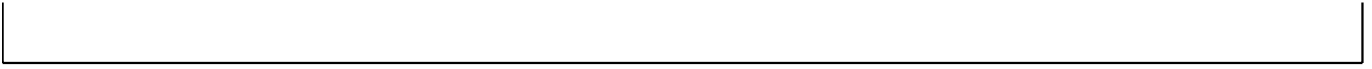


All signals received on this page are considered unexpected and were added to complete the protocol during verification to remove implicit consumption of these signals. A real implementation does not need this because these signals cannot occur while in this state.



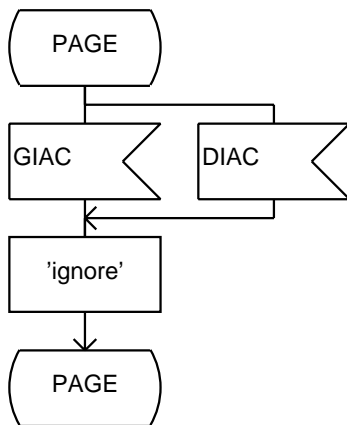


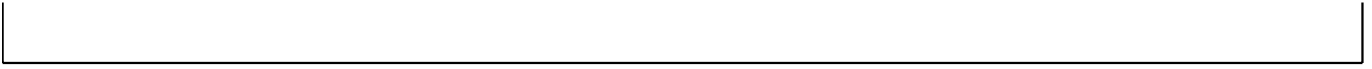


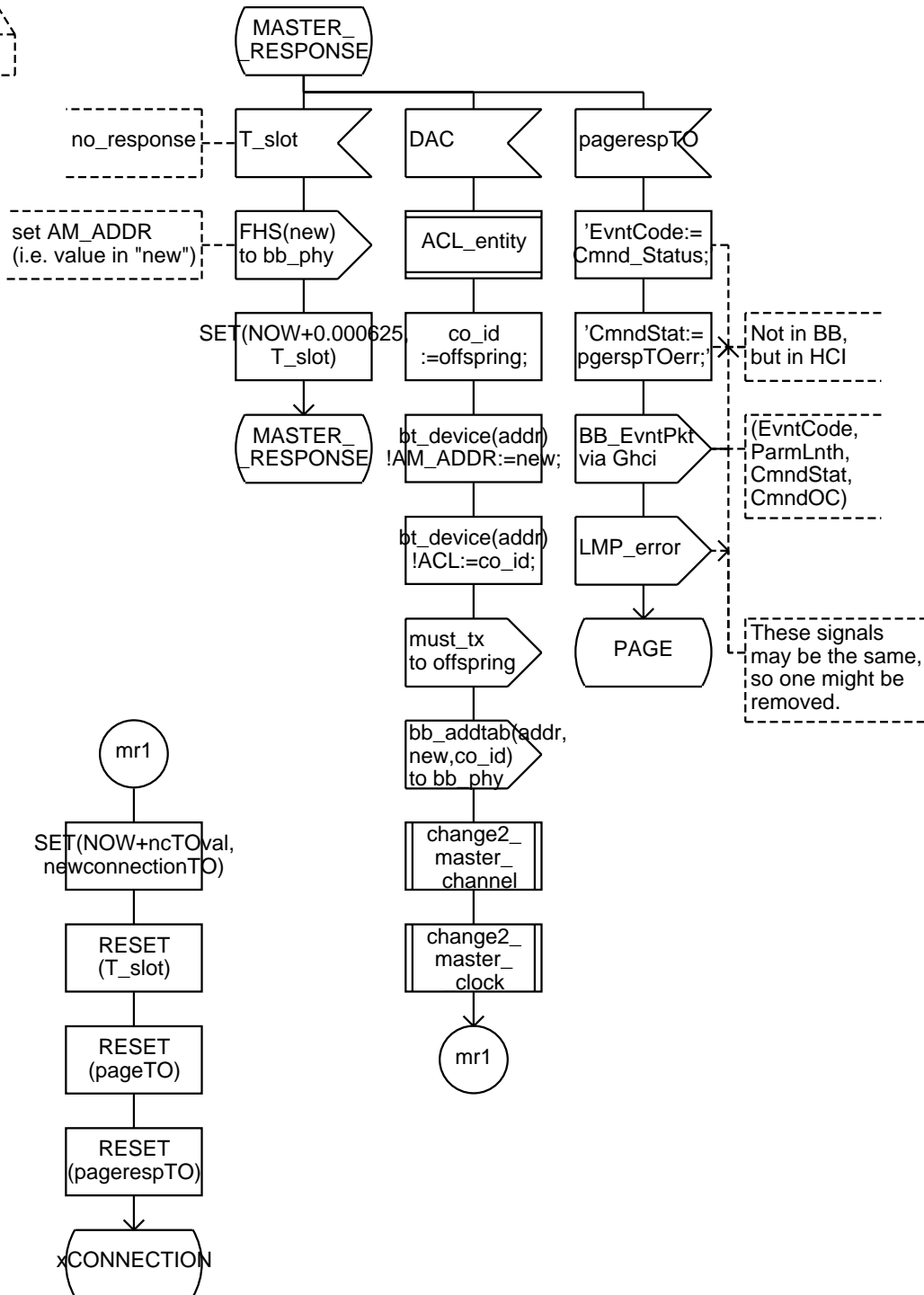


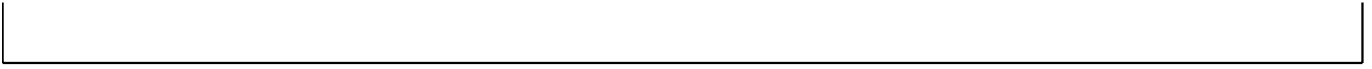


All signals received on this page are considered unexpected and were added to complete the protocol during verification to remove implicit consumption of these signals. A real implementation does not need this because these signals cannot occur while in this state.



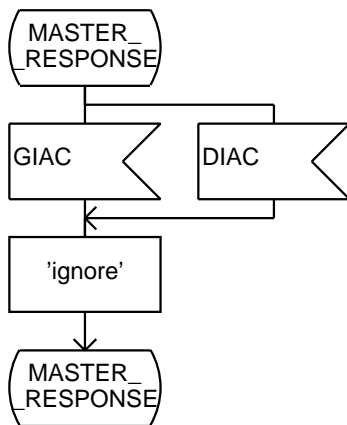


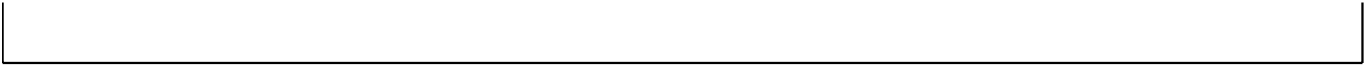


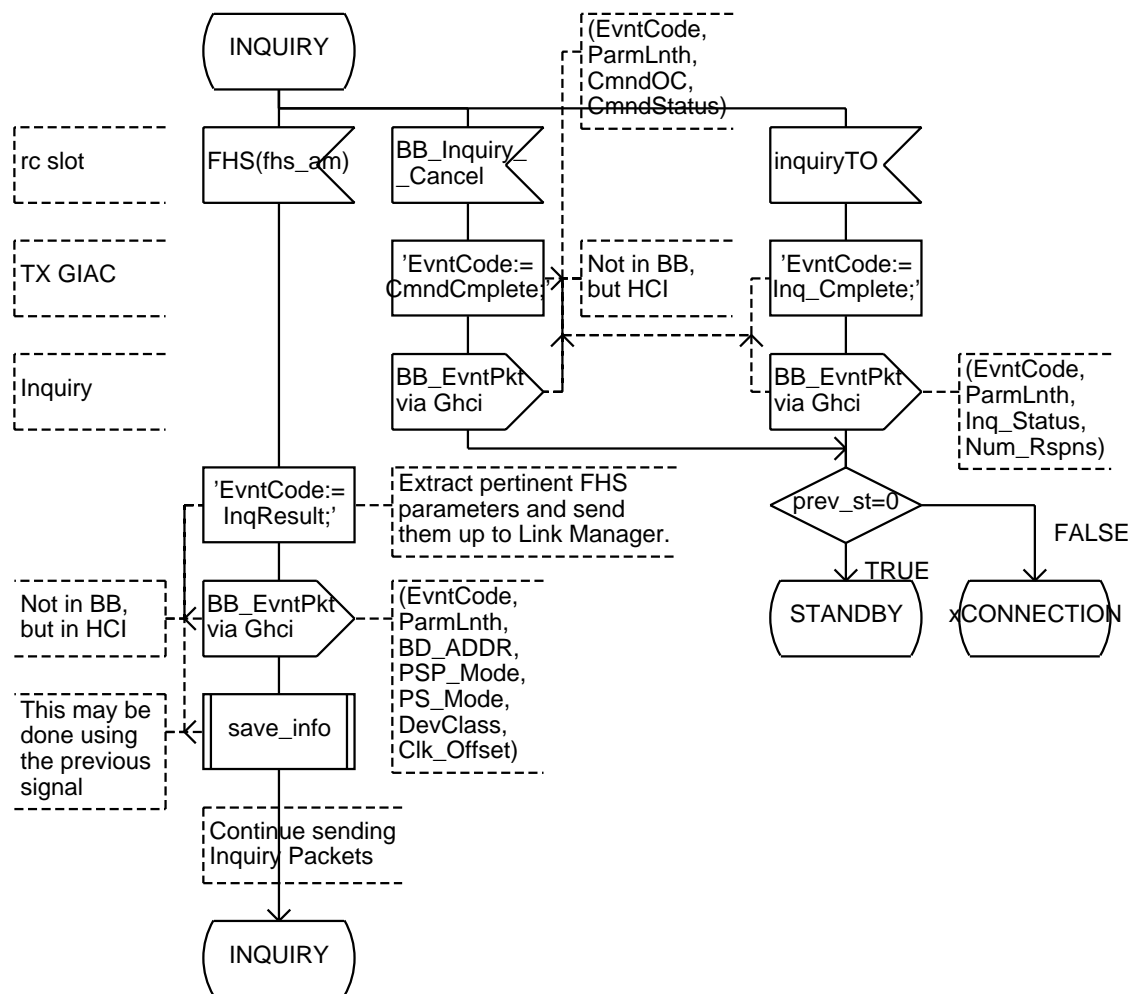


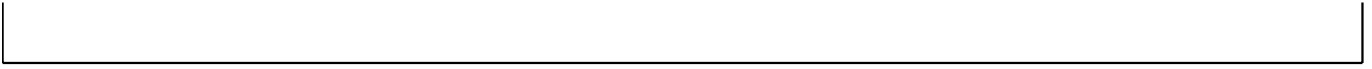


All signals received on this page are considered unexpected and were added to complete the protocol during verification to remove implicit consumption of these signals. A real implementation does not need this because these signals cannot occur while in this state.



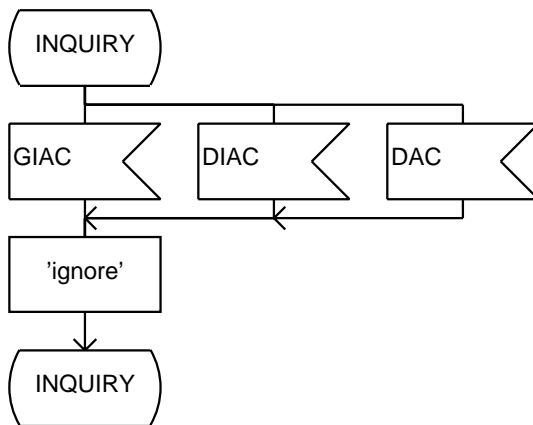


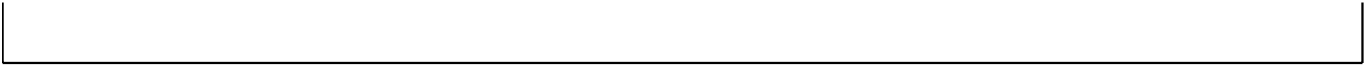


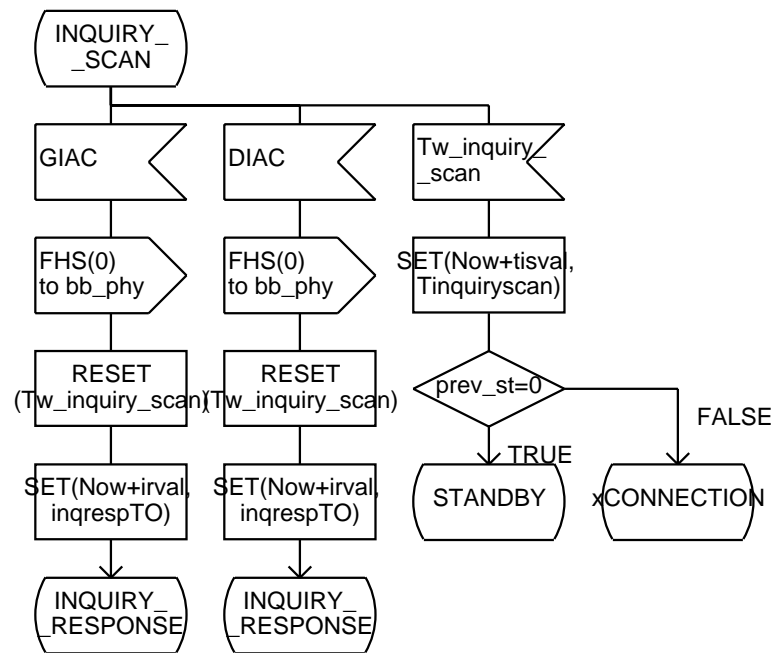


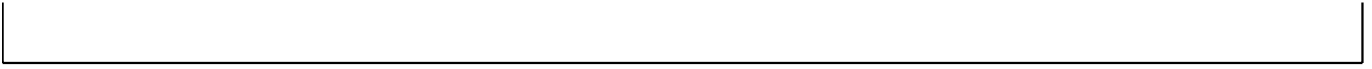


All signals received on this page are considered unexpected and were added to complete the protocol during verification to remove implicit consumption of these signals. A real implementation does not need this because these signals cannot occur while in this state.



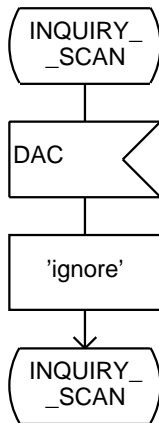


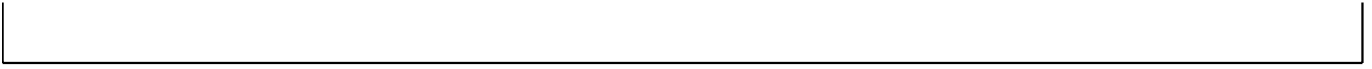


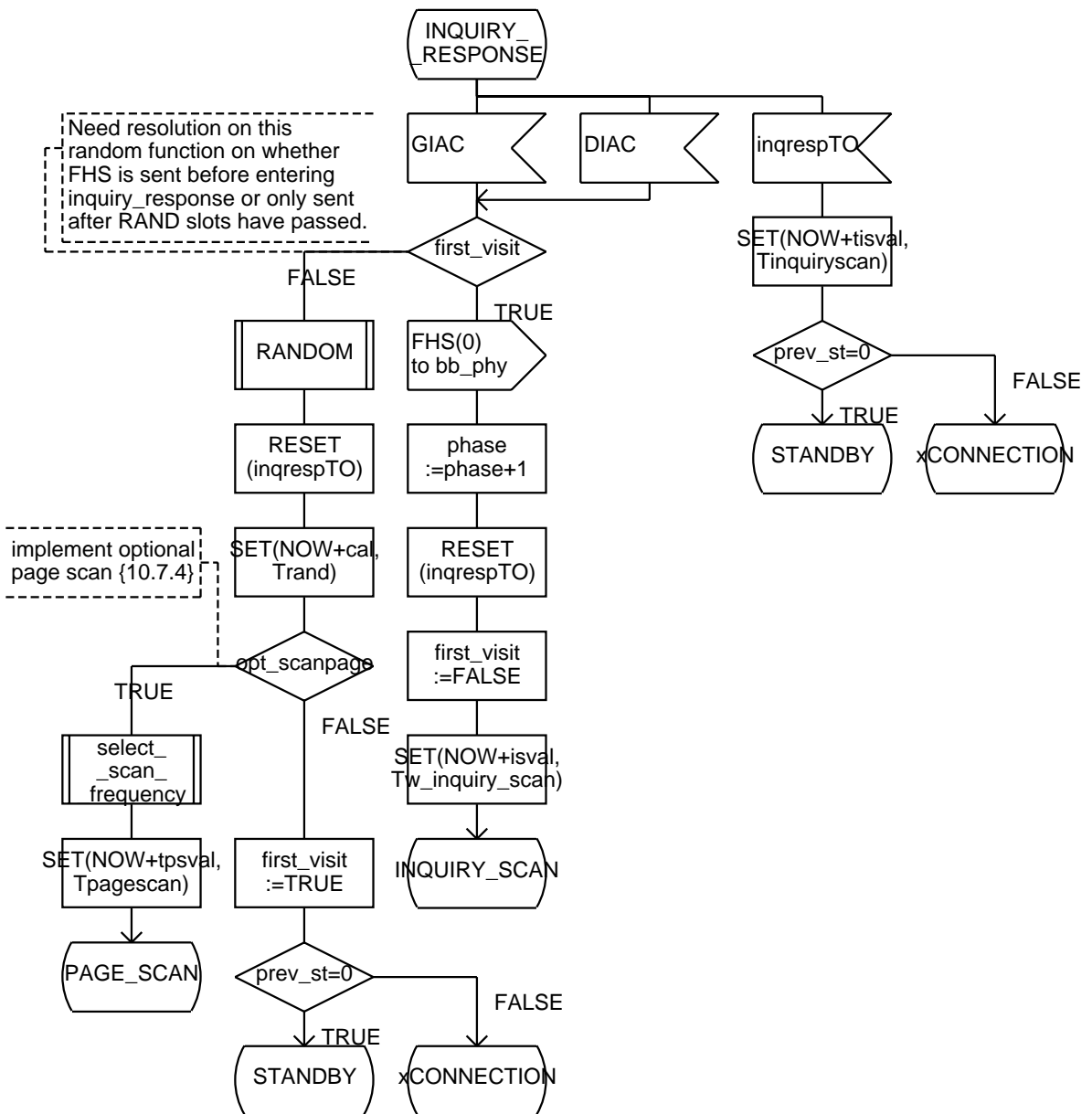


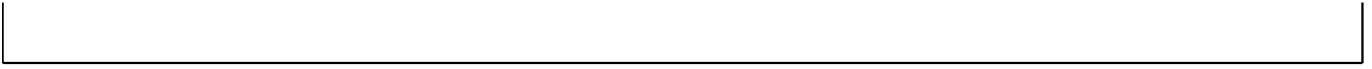


All signals received on this page are considered unexpected and were added to complete the protocol during verification to remove implicit consumption of these signals. A real implementation does not need this because these signals cannot occur while in this state.



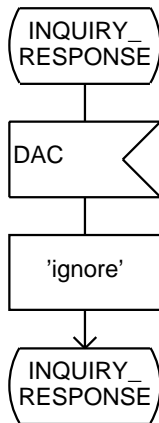


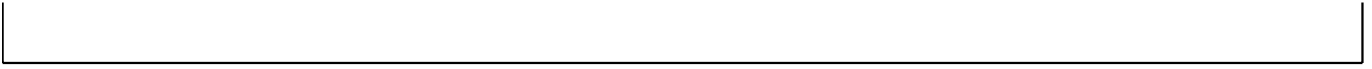






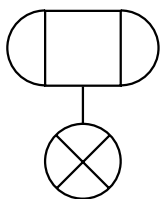
All signals received on this page are considered unexpected and were added to complete the protocol during verification to remove implicit consumption of these signals. A real implementation does not need this because these signals cannot occur while in this state.





procedure activate_receiver

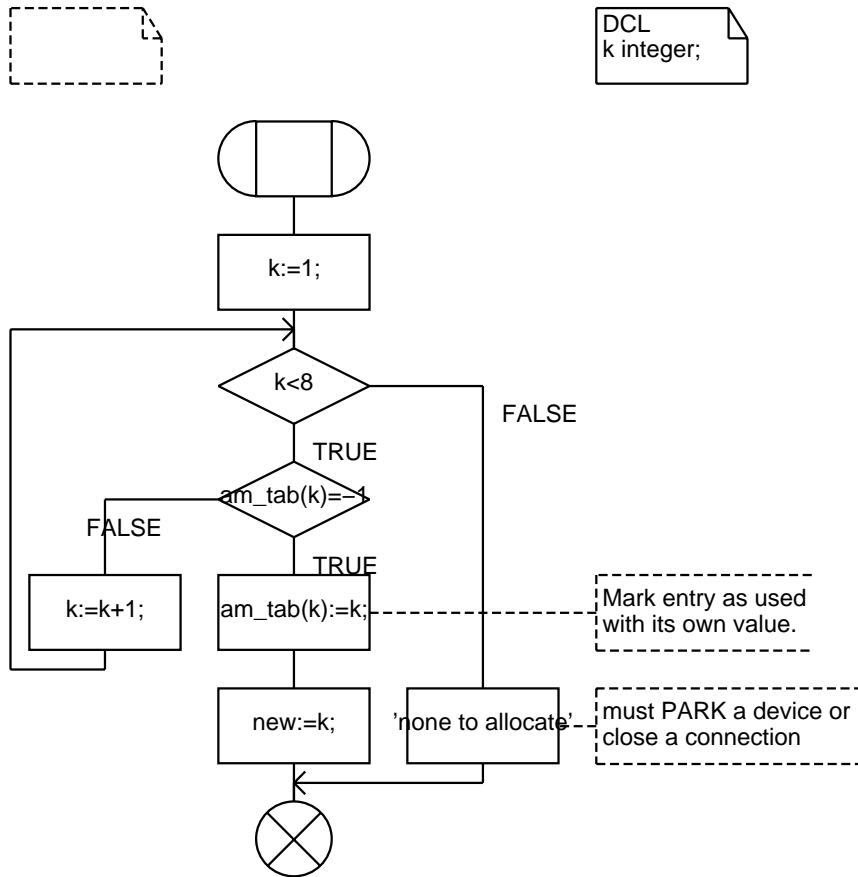
1(1)

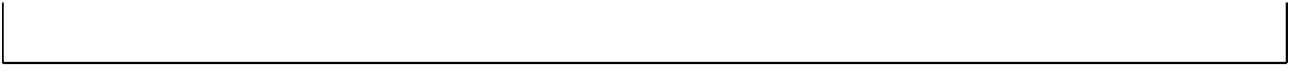


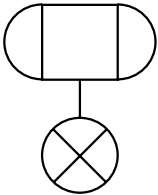


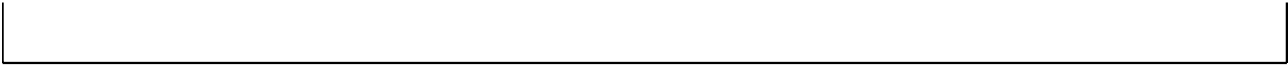
procedure allocate_AM_ADDR

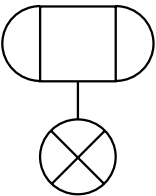
1(1)











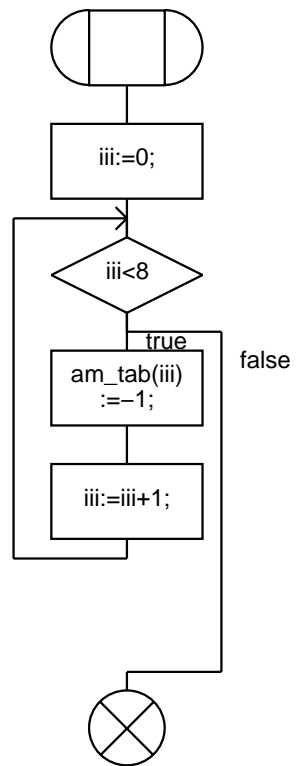


procedure clear_am_array

1(1)



DCL
iii Integer;



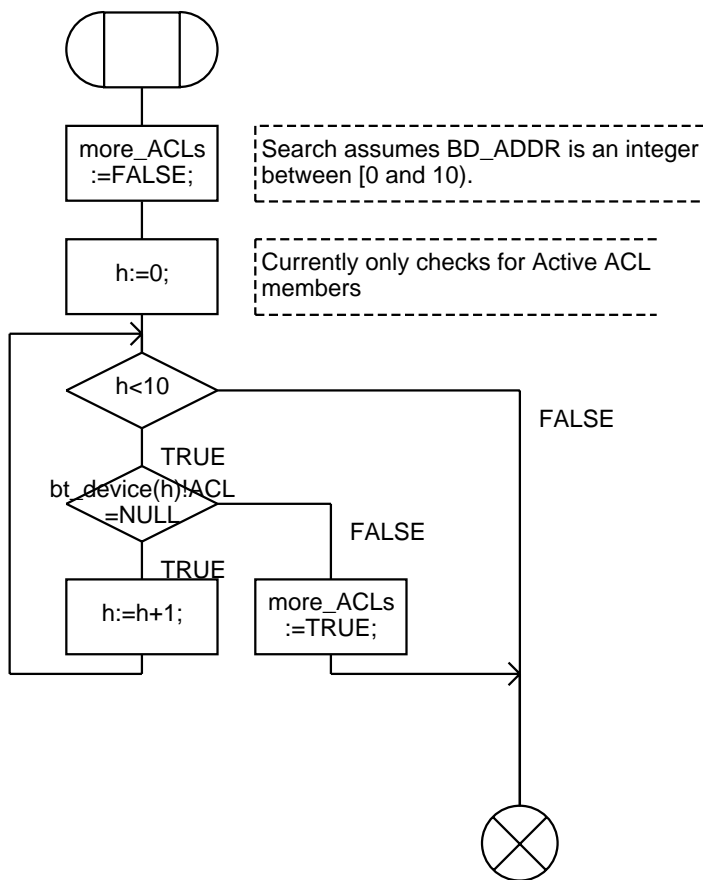


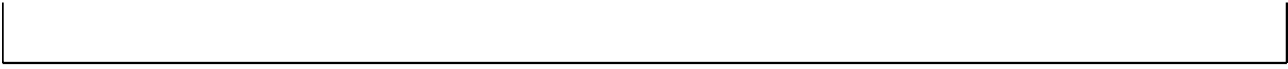
procedure check_ACLs

1(1)



DCL
h integer;



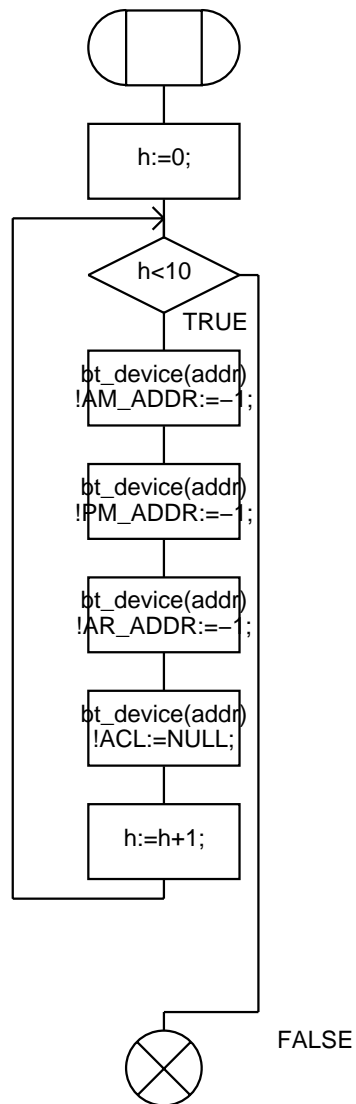


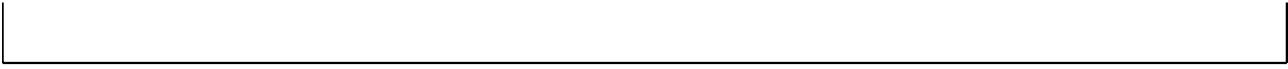
procedure clear_bt_array

1(1)



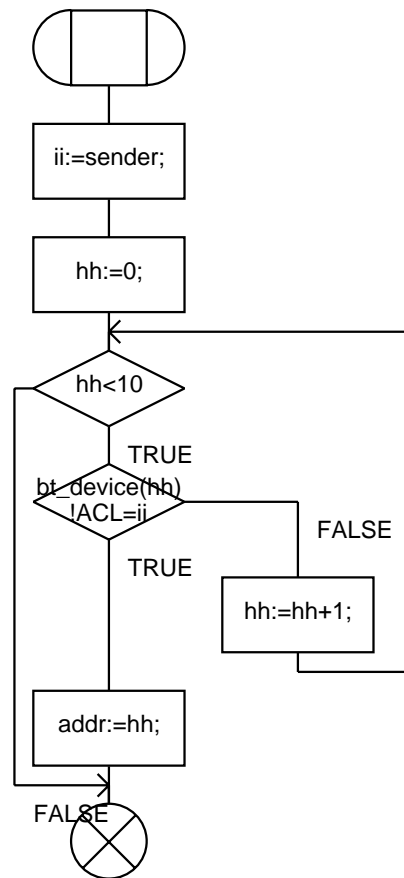
DCL
h integer;



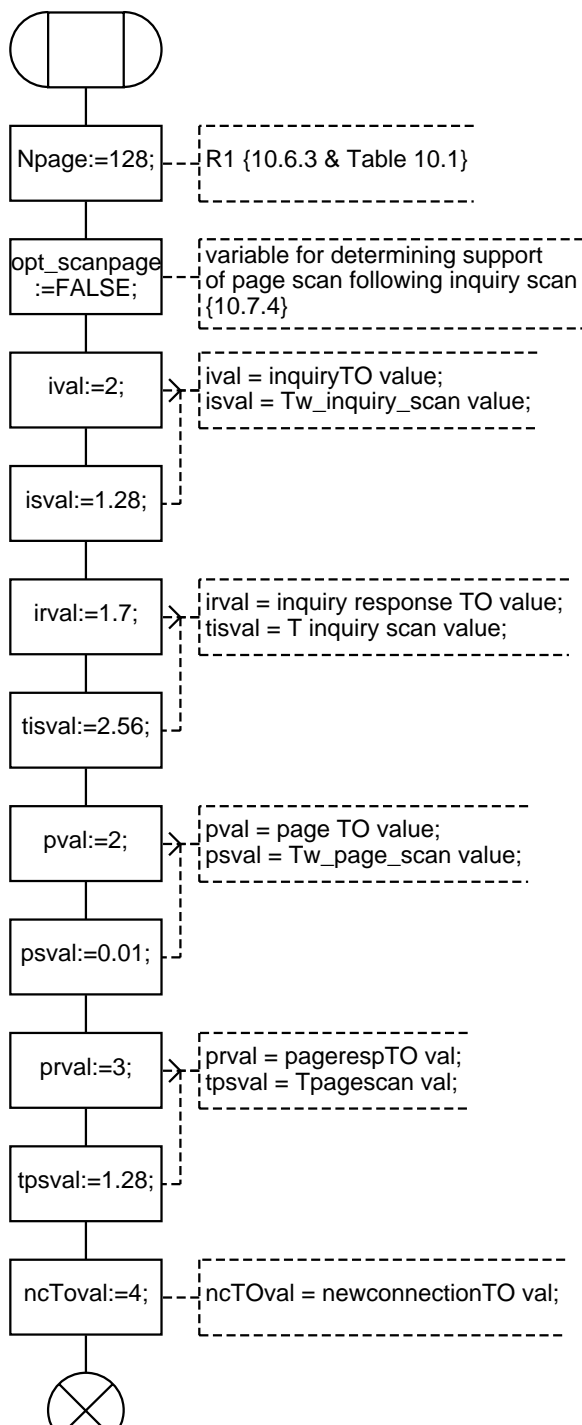


procedure find_Pid_index

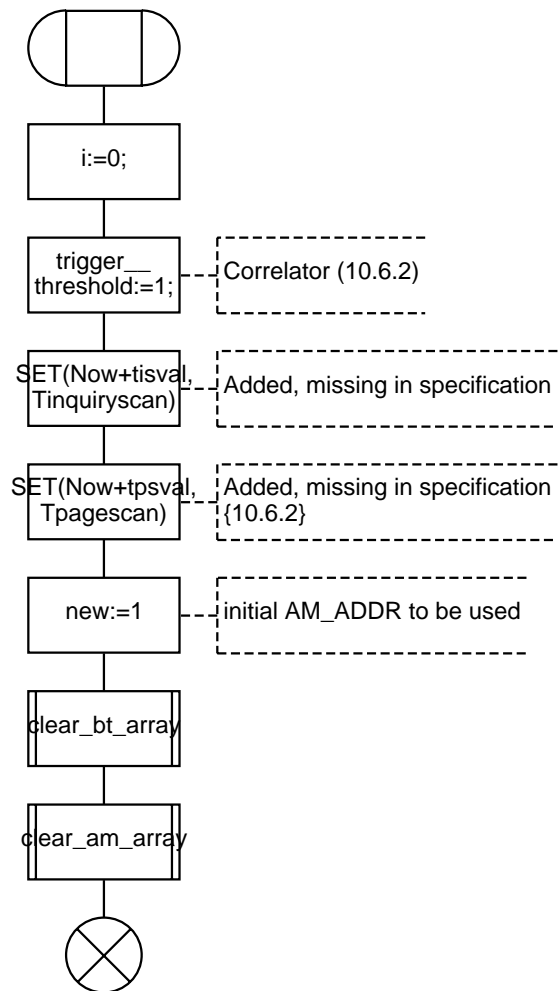
1(1)







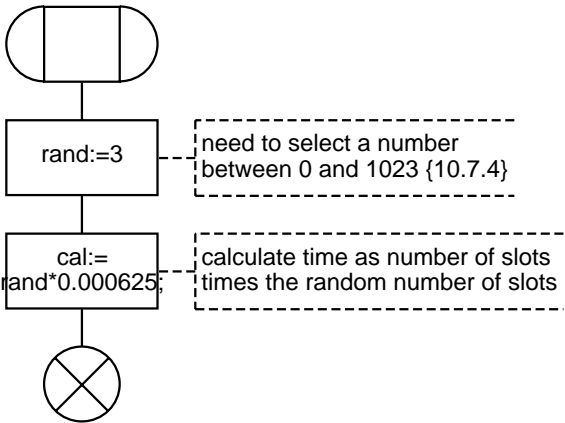


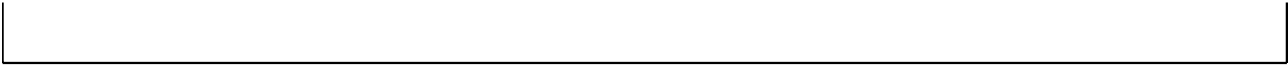


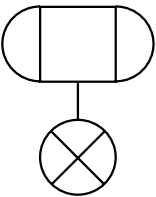


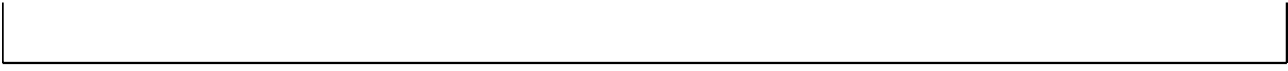


DCL
rand Duration;



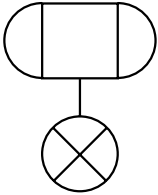


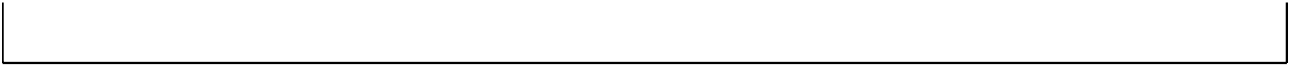


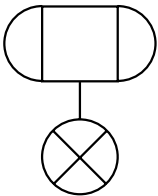


procedure select_next_hop

1(1)



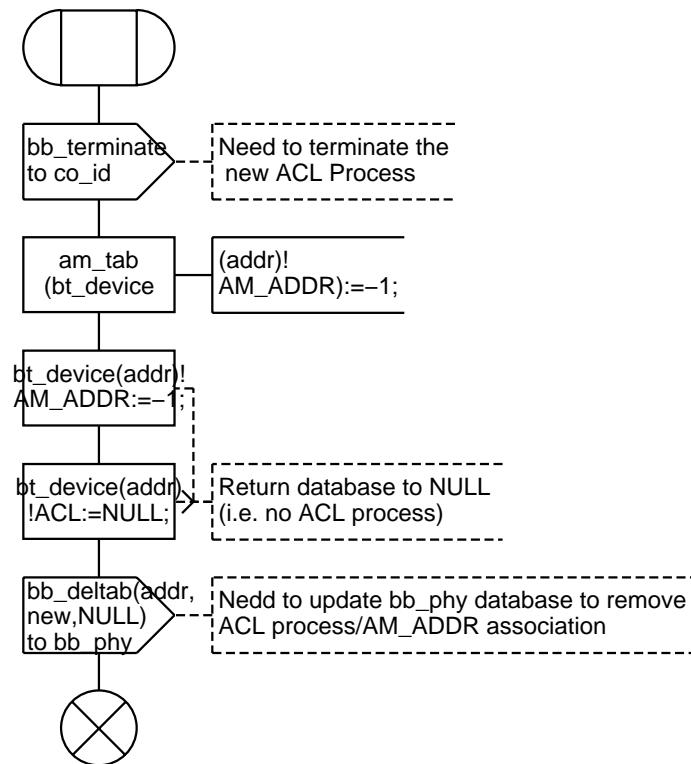






procedure terminate_ACL

1(1)





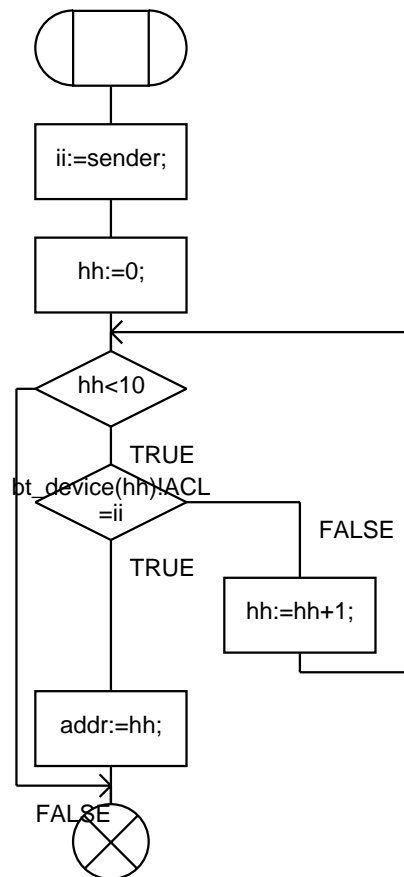
procedure find_addr_given_pid1

1(1)

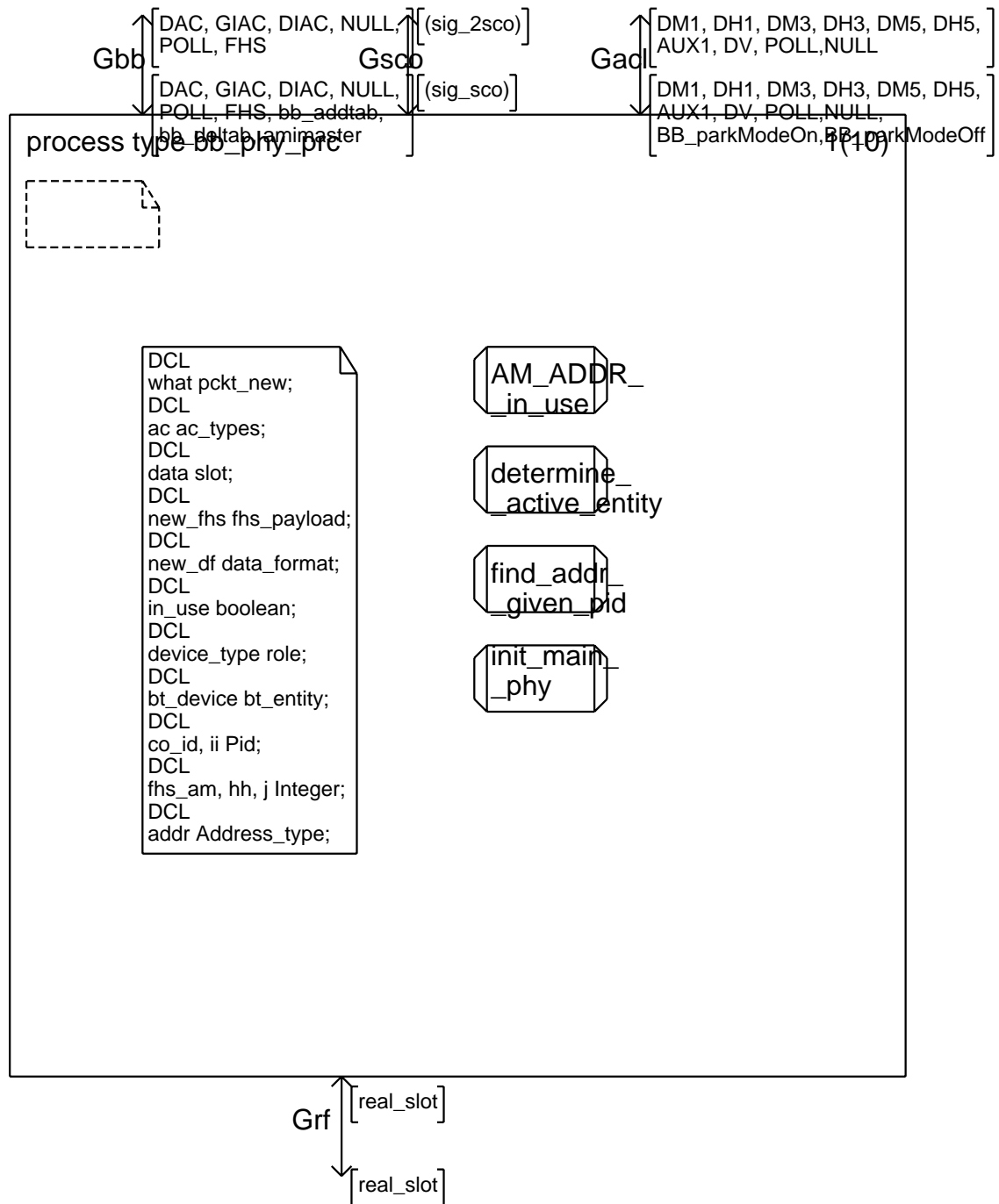


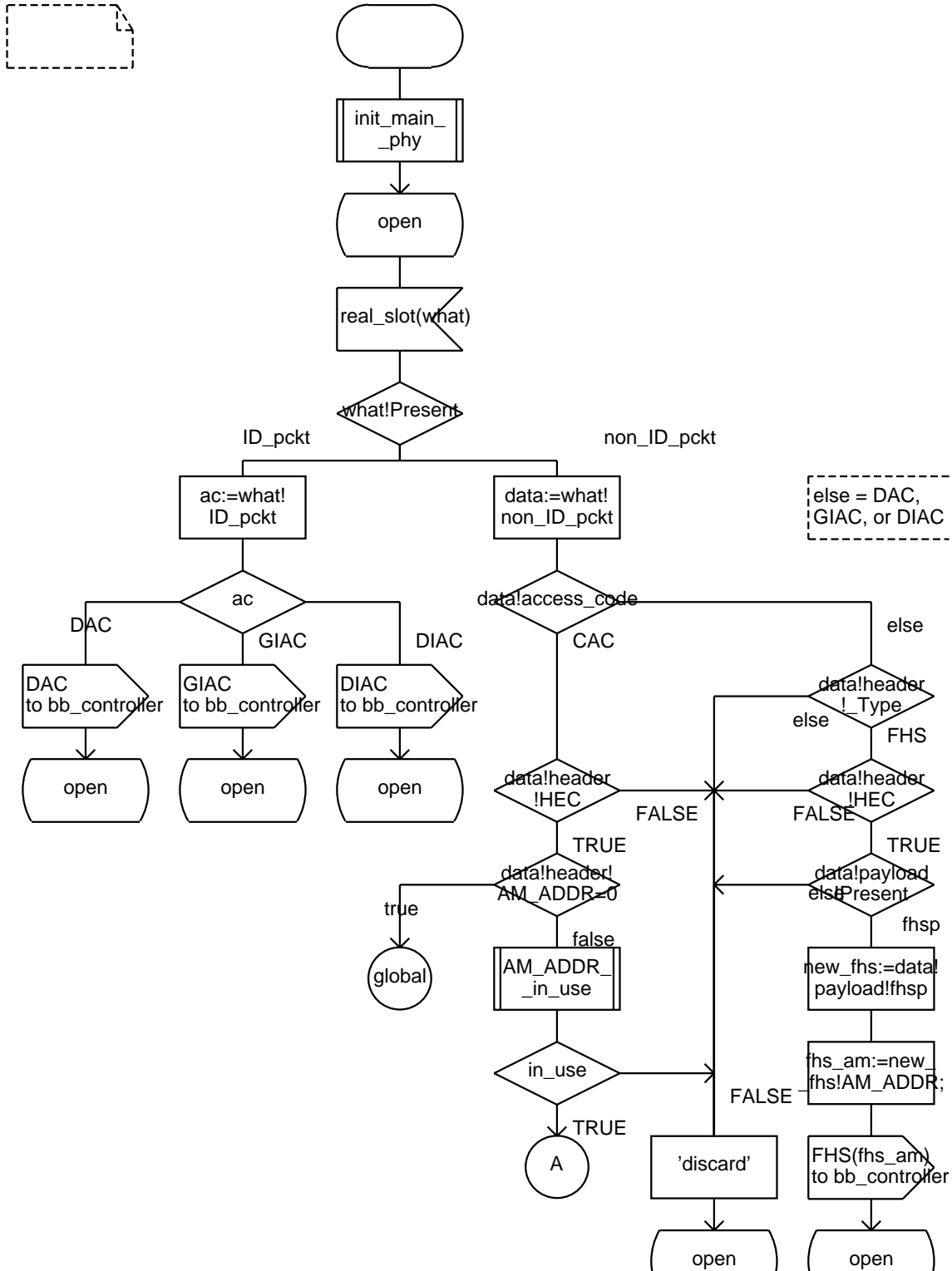
Given a Process ID
Find the BD_ADDR

Assume BD_ADDR are
in the range [0,10)

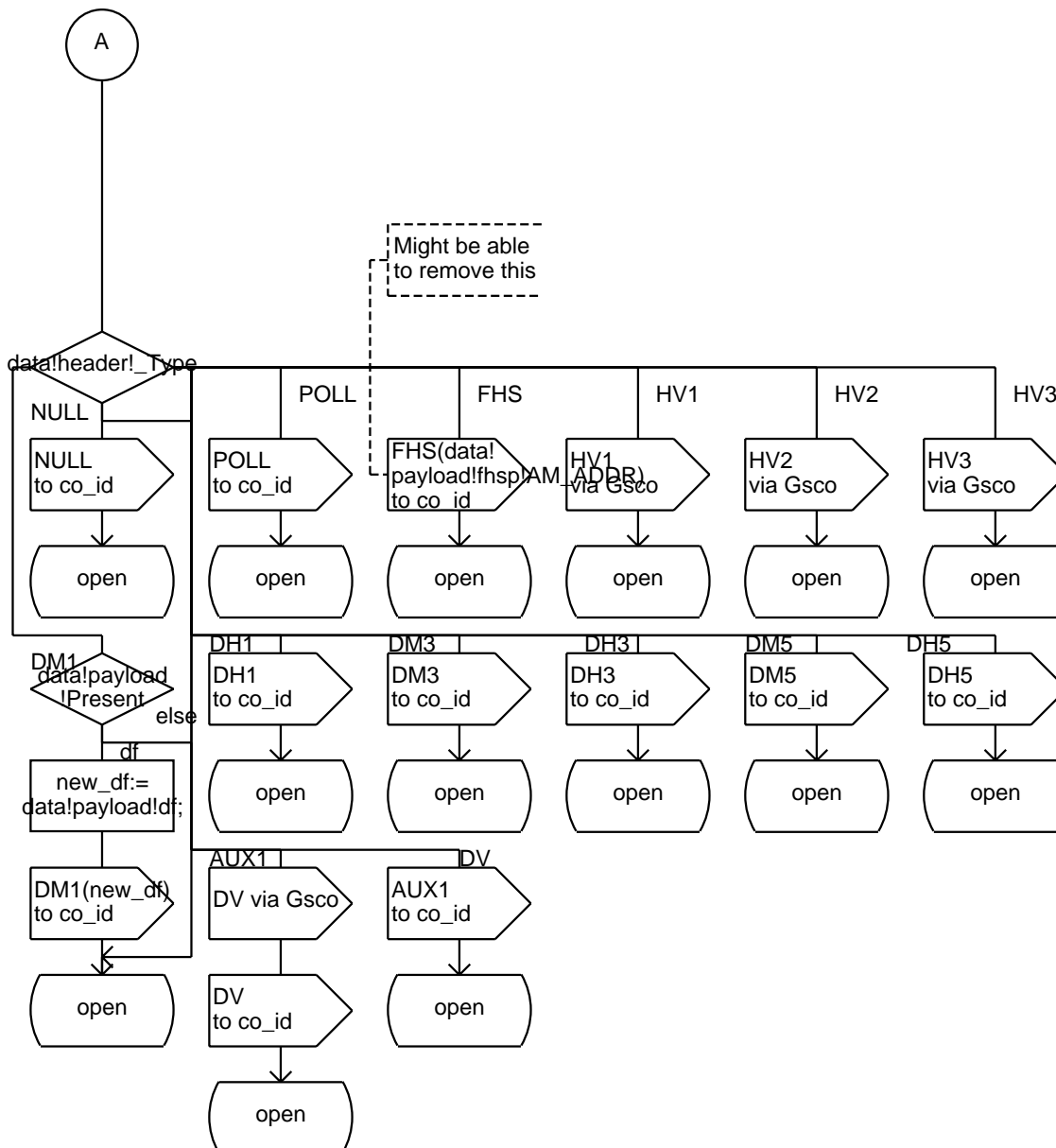




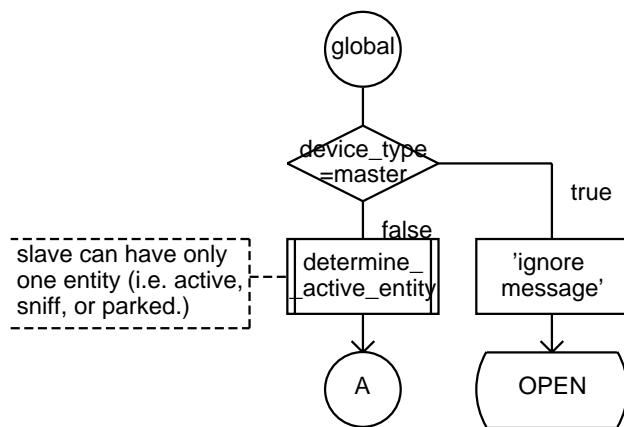




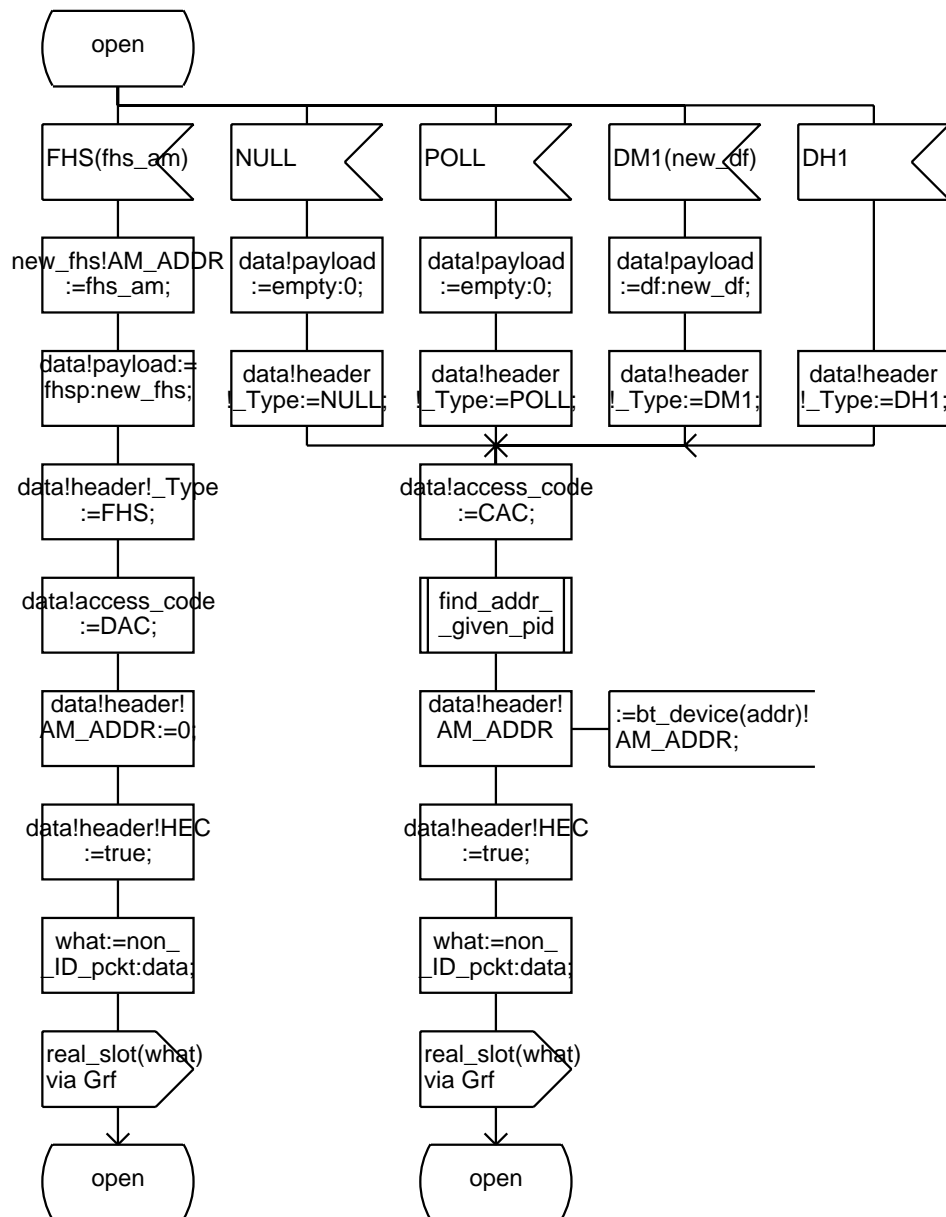


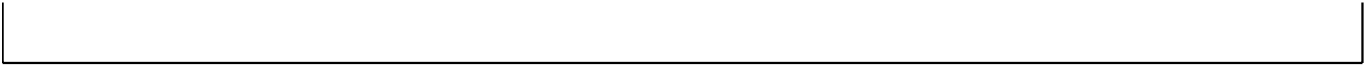


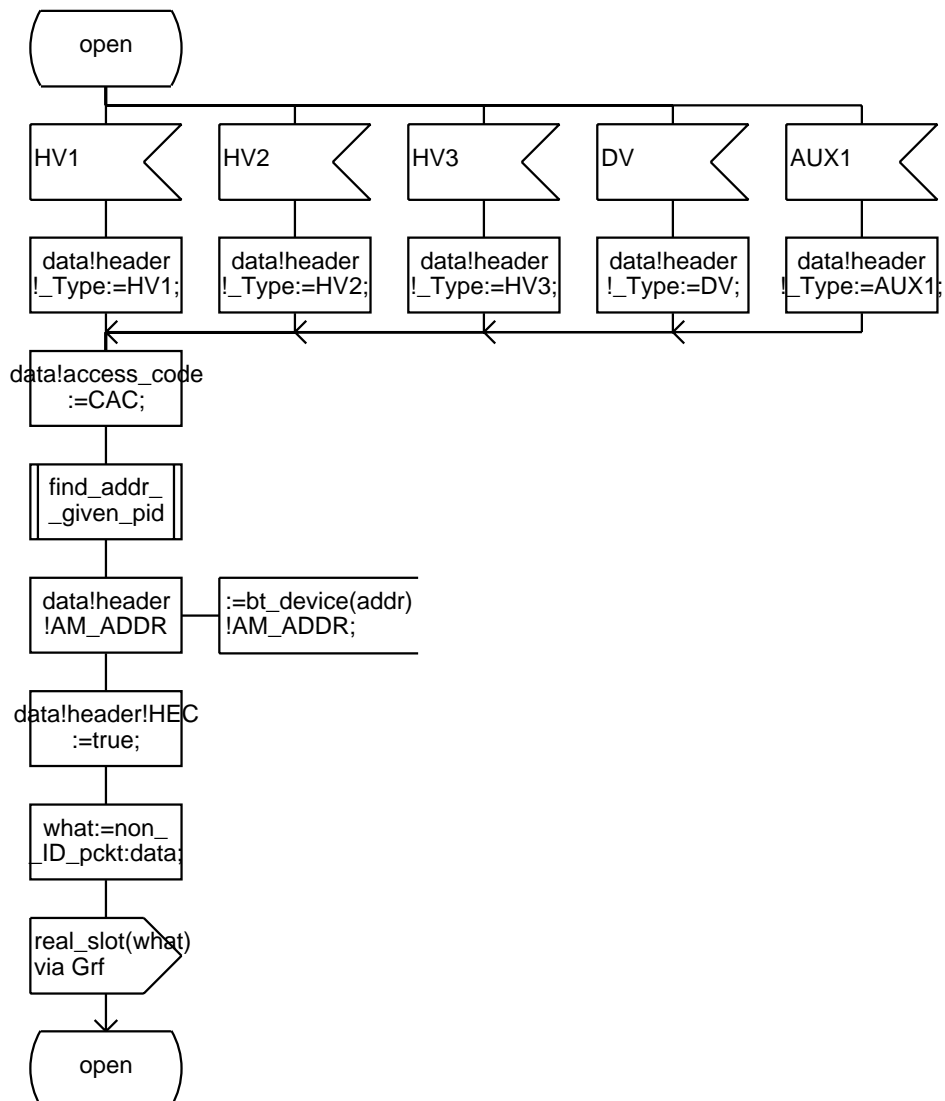


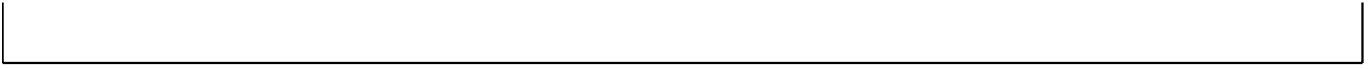


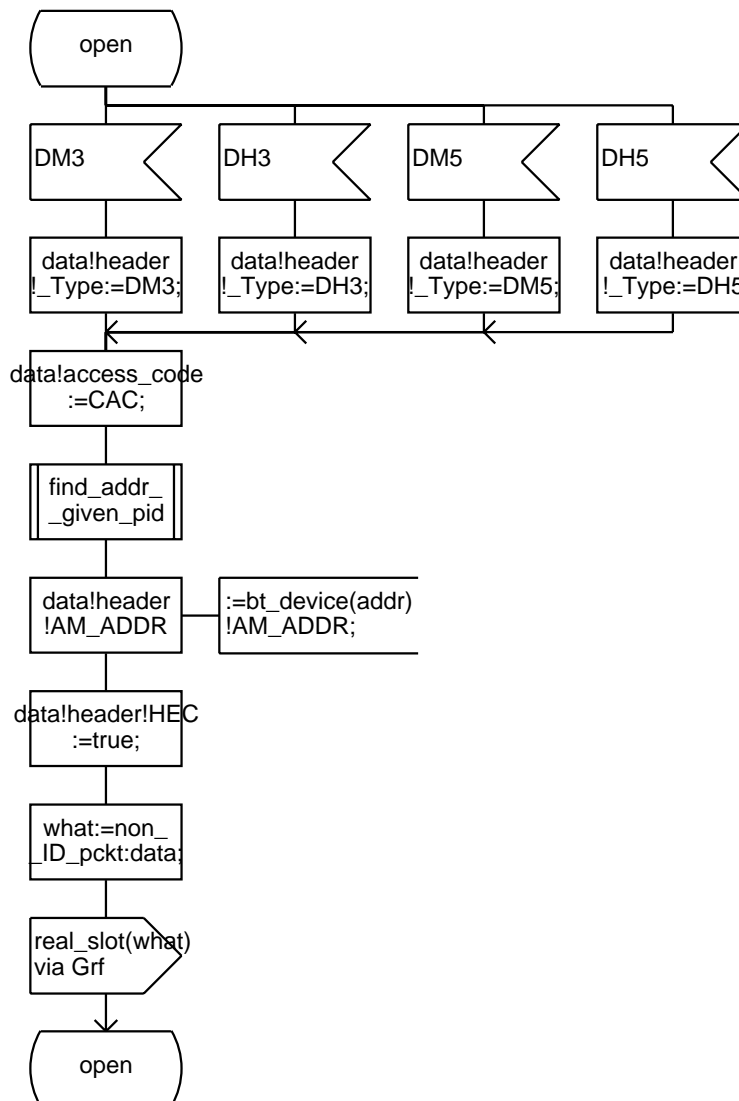


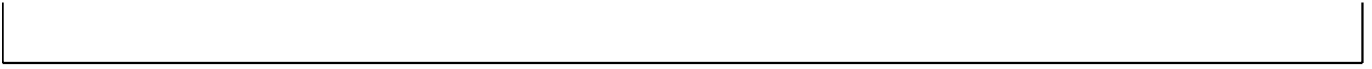


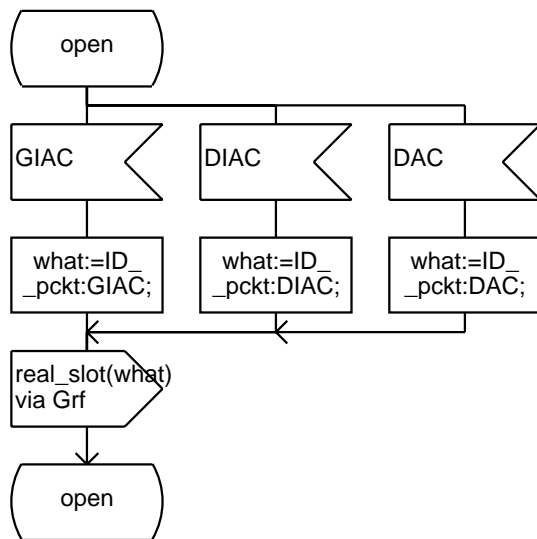


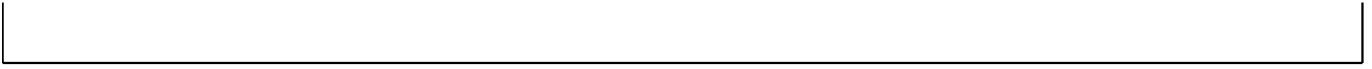








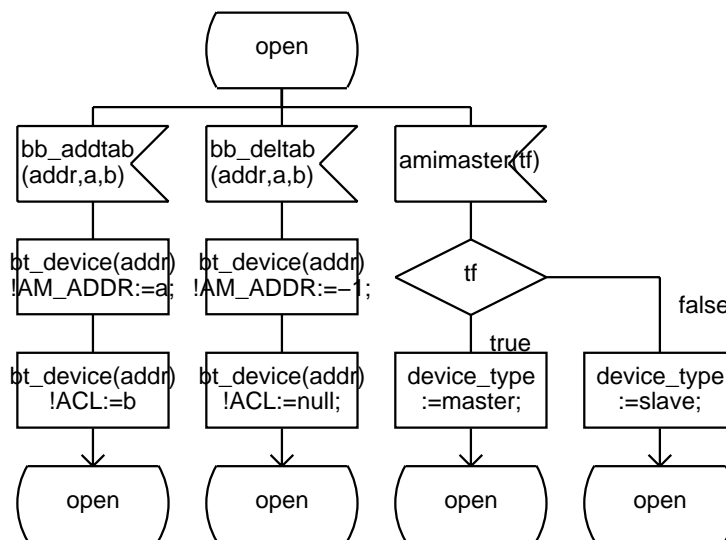


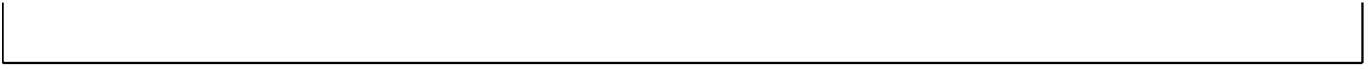




DCL
a Integer;
DCL
b Pid;
DCL
tf boolean;

Signals on this page are not part of the specification, but are implementation signals to exchange information among processes. These may be replaced if another method is found within the SDL.

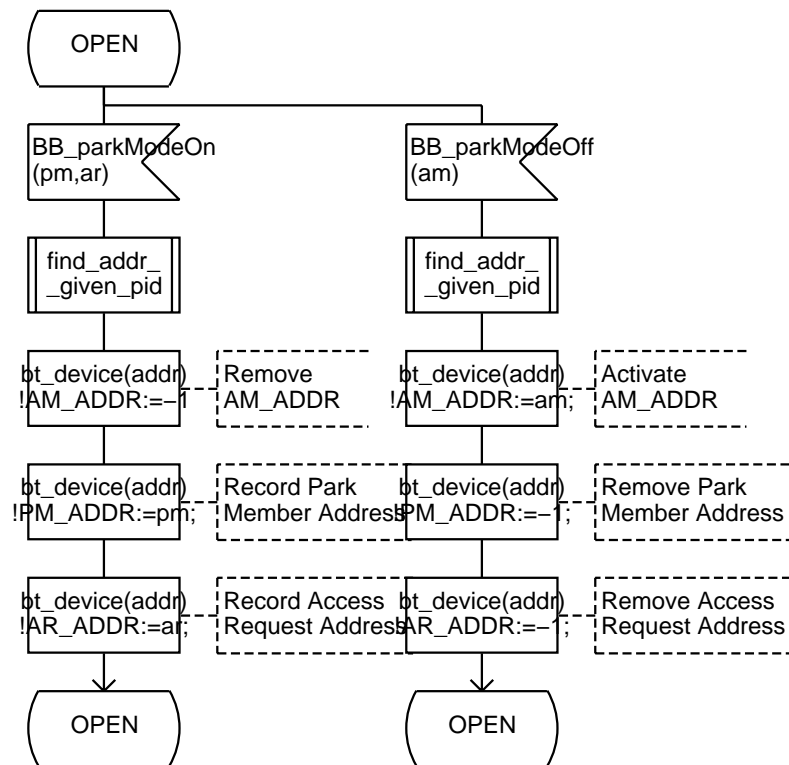


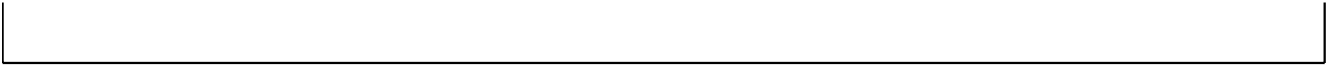




undefined signals to
coordinate parking &
unparking of device.

DCL
pm, ar, am Integer;



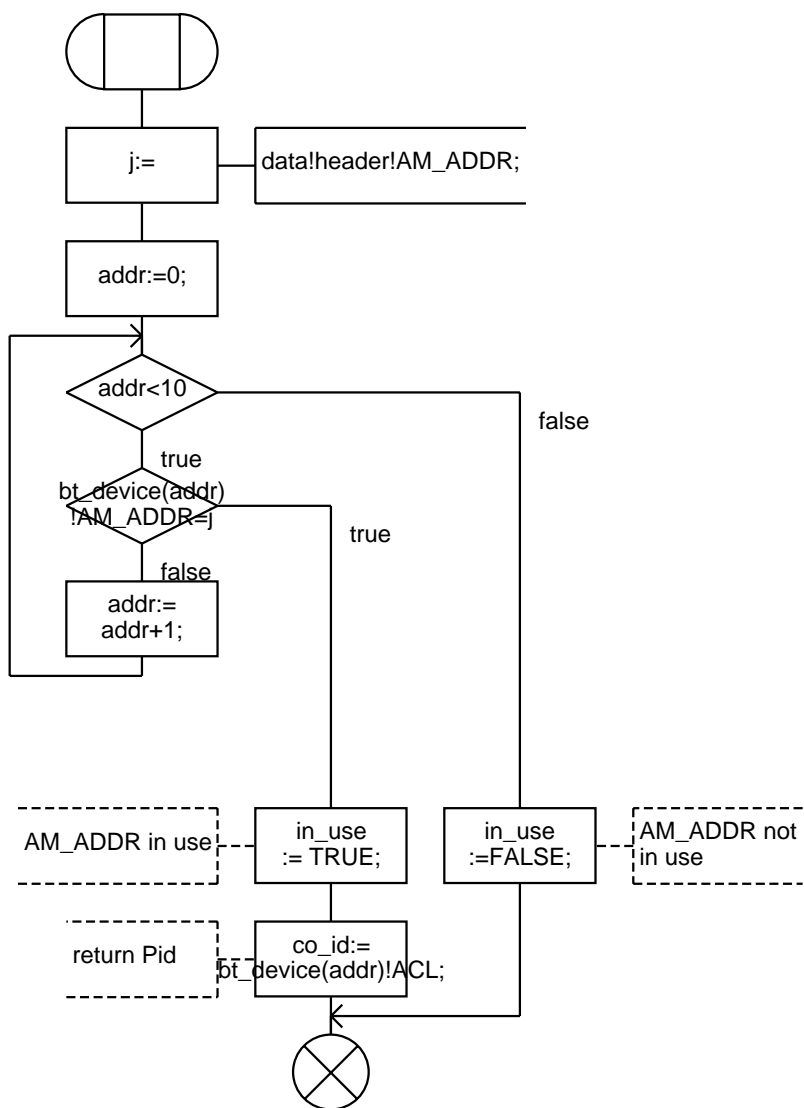


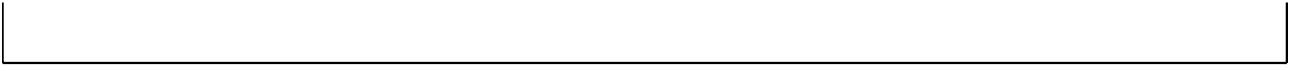
procedure AM_ADDR_in_use

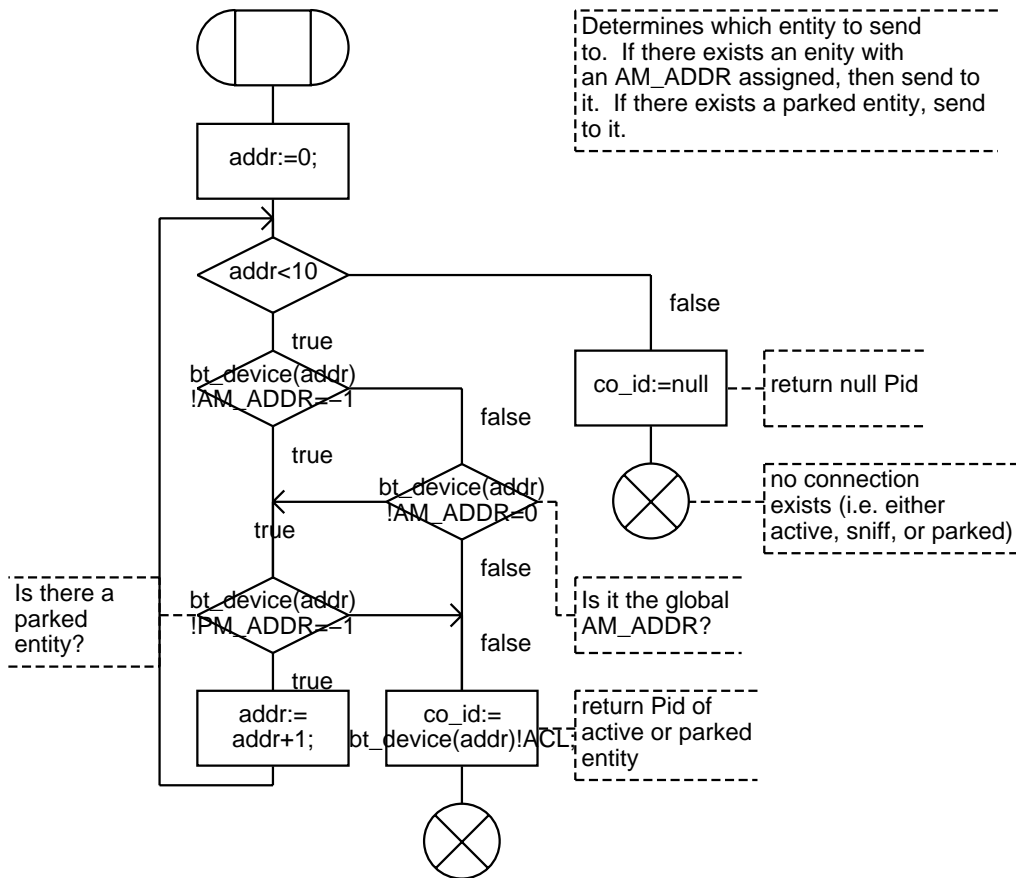
1(1)

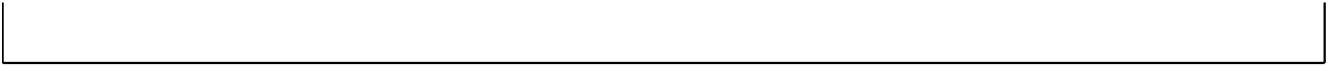


Not the most efficient method,
but then, this is not to be standardized.









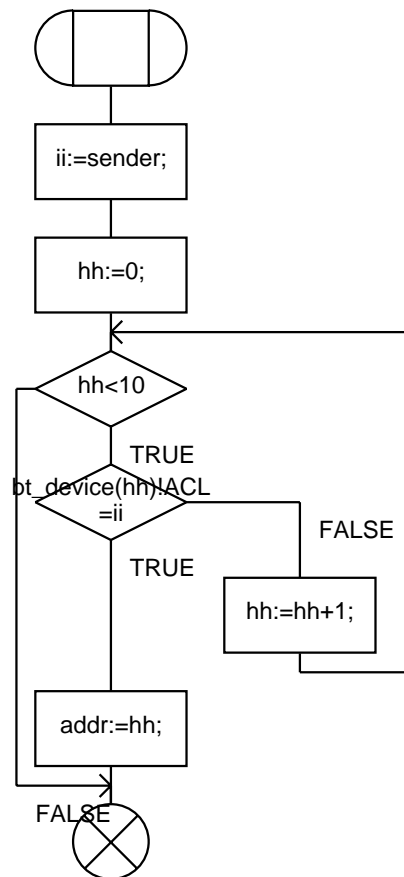
procedure find_addr_given_pid

1(1)



Given a Process ID
Find the BD_ADDR

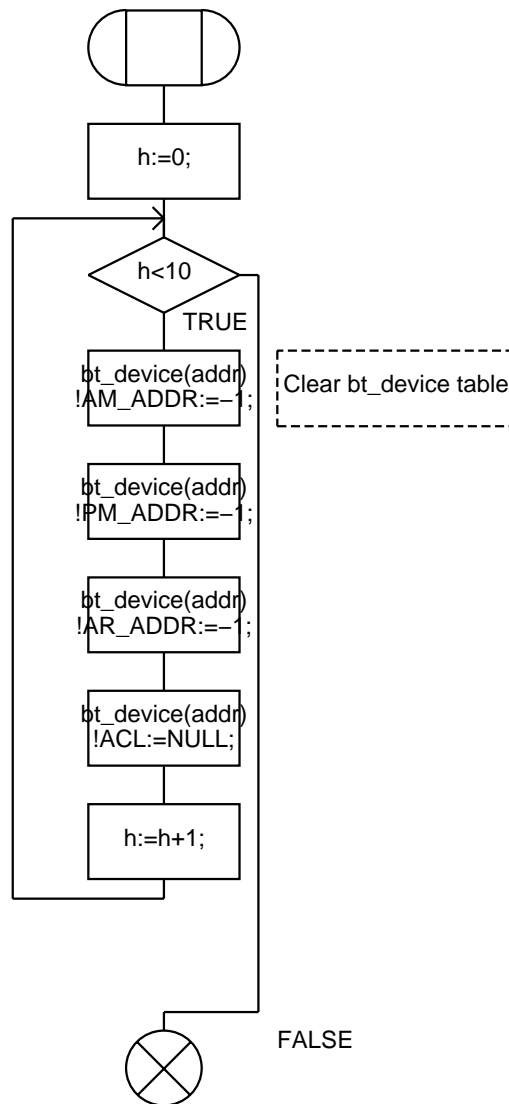
Assume BD_ADDR are
in the range [0,10)



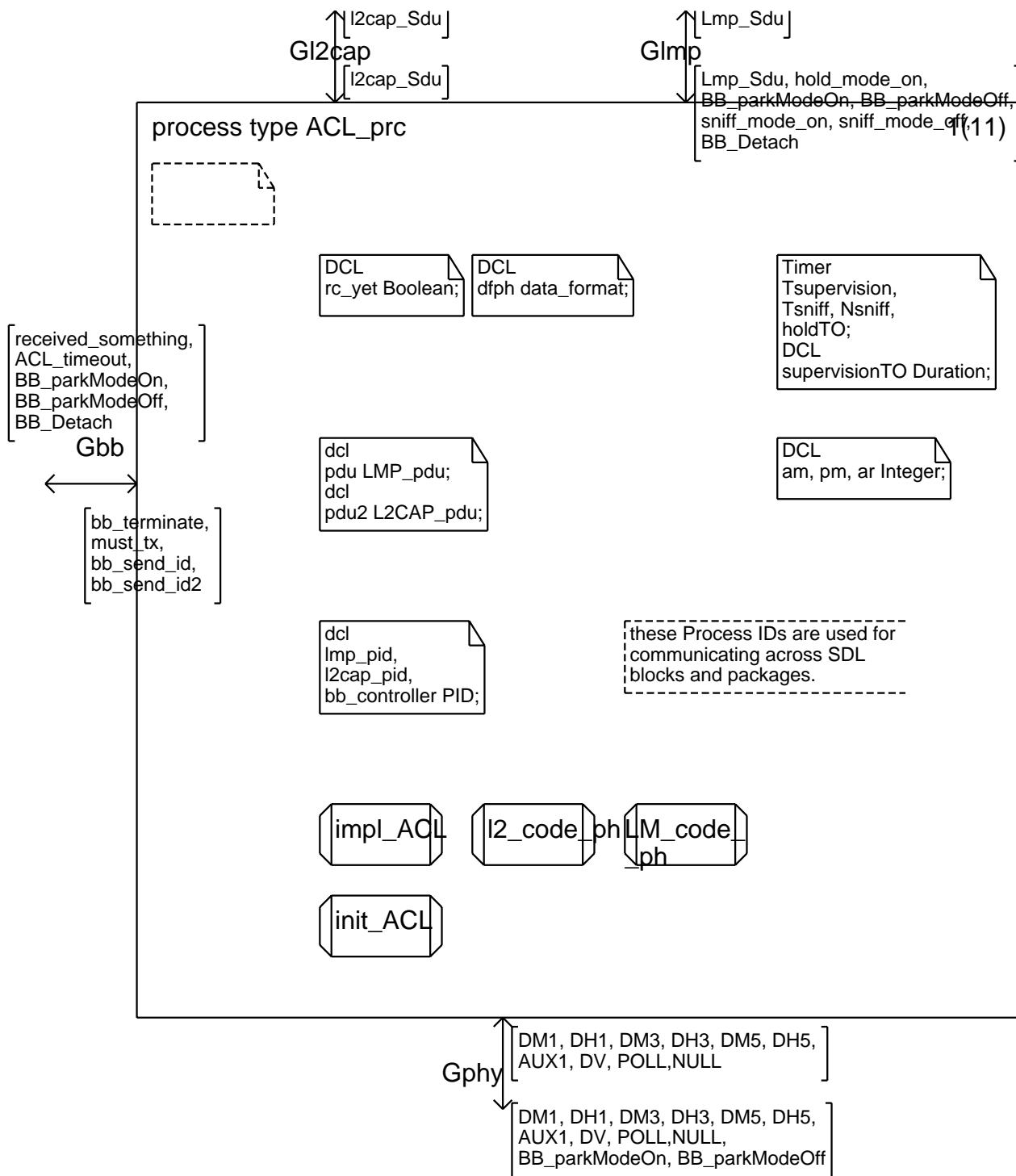


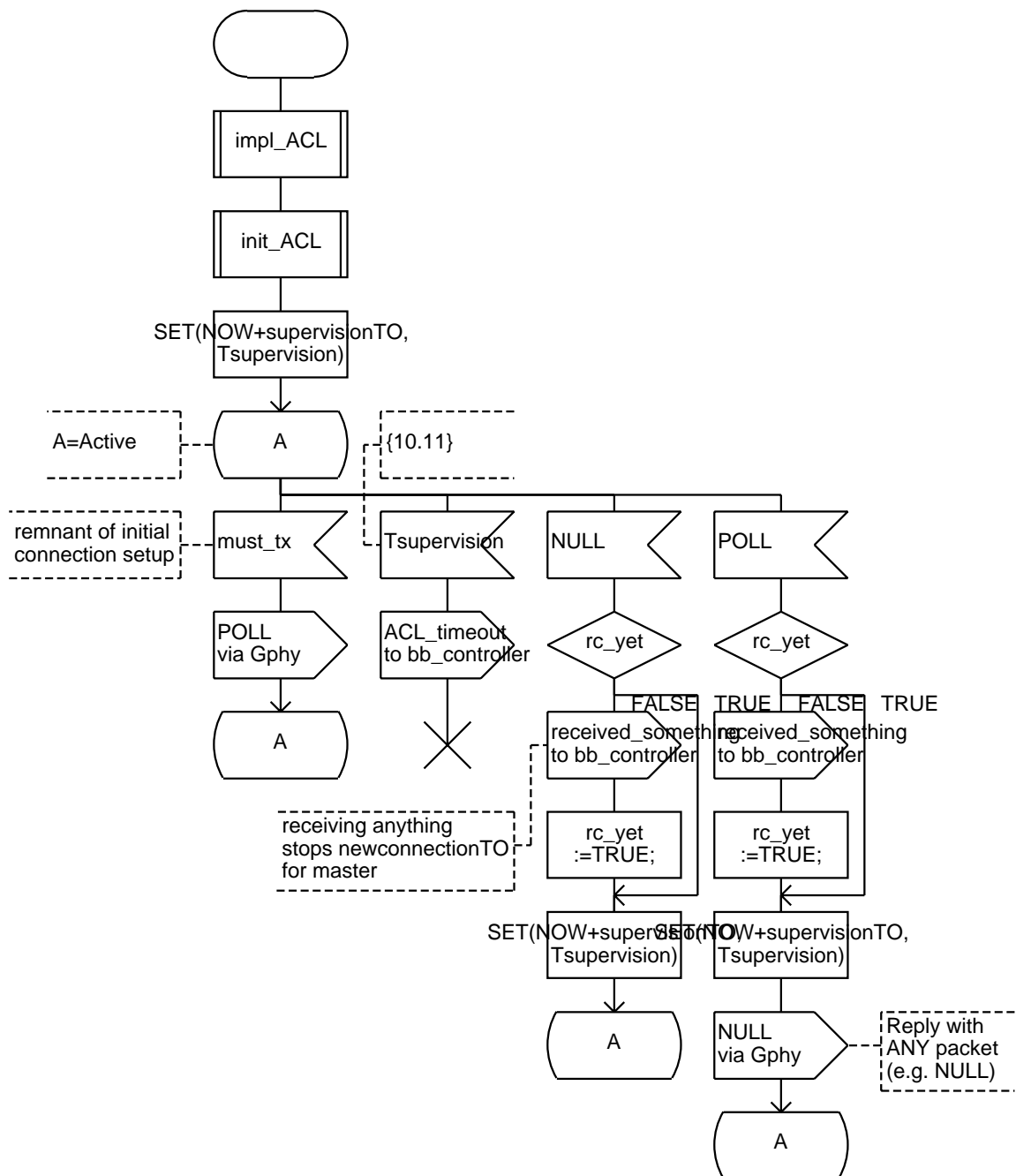


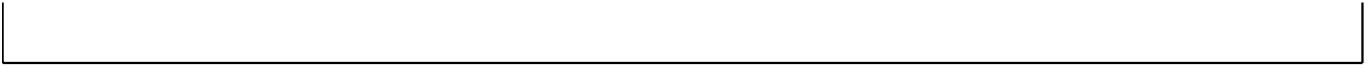
DCL
h integer;

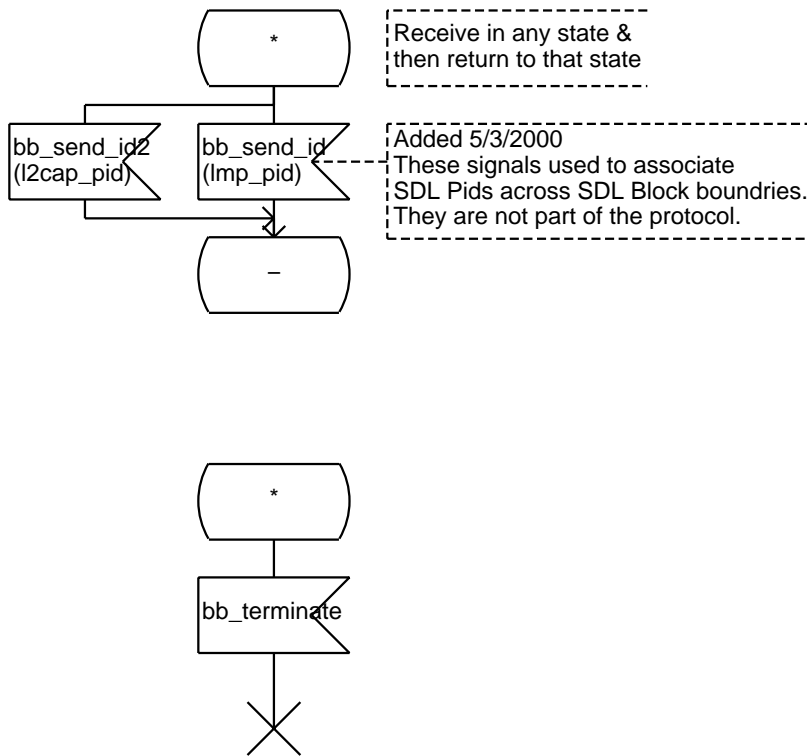


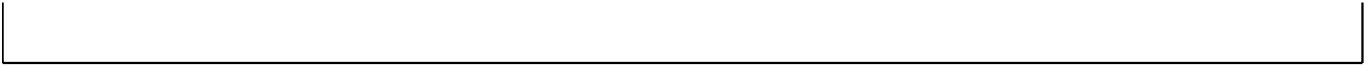


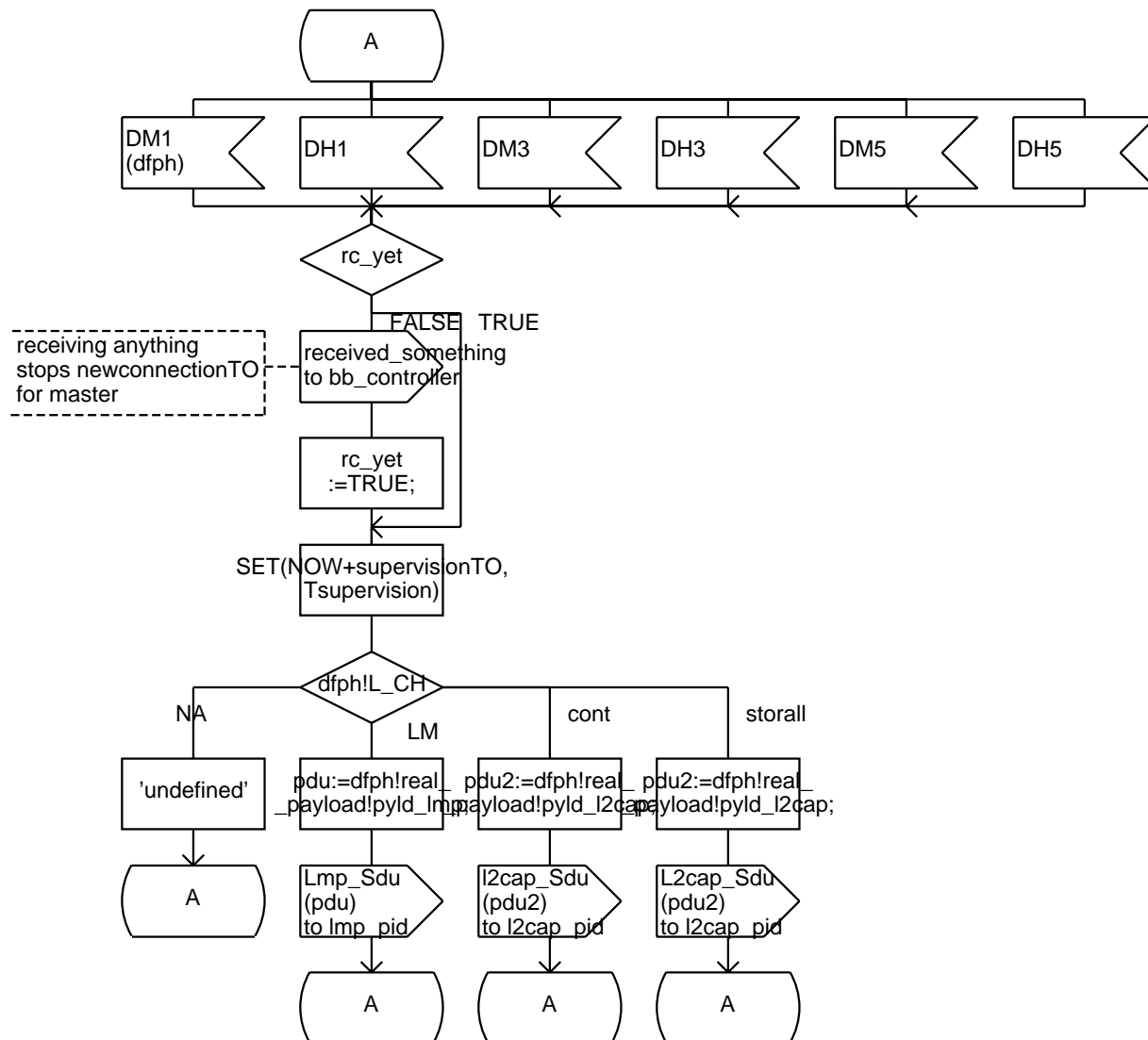


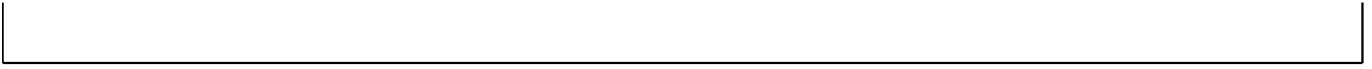


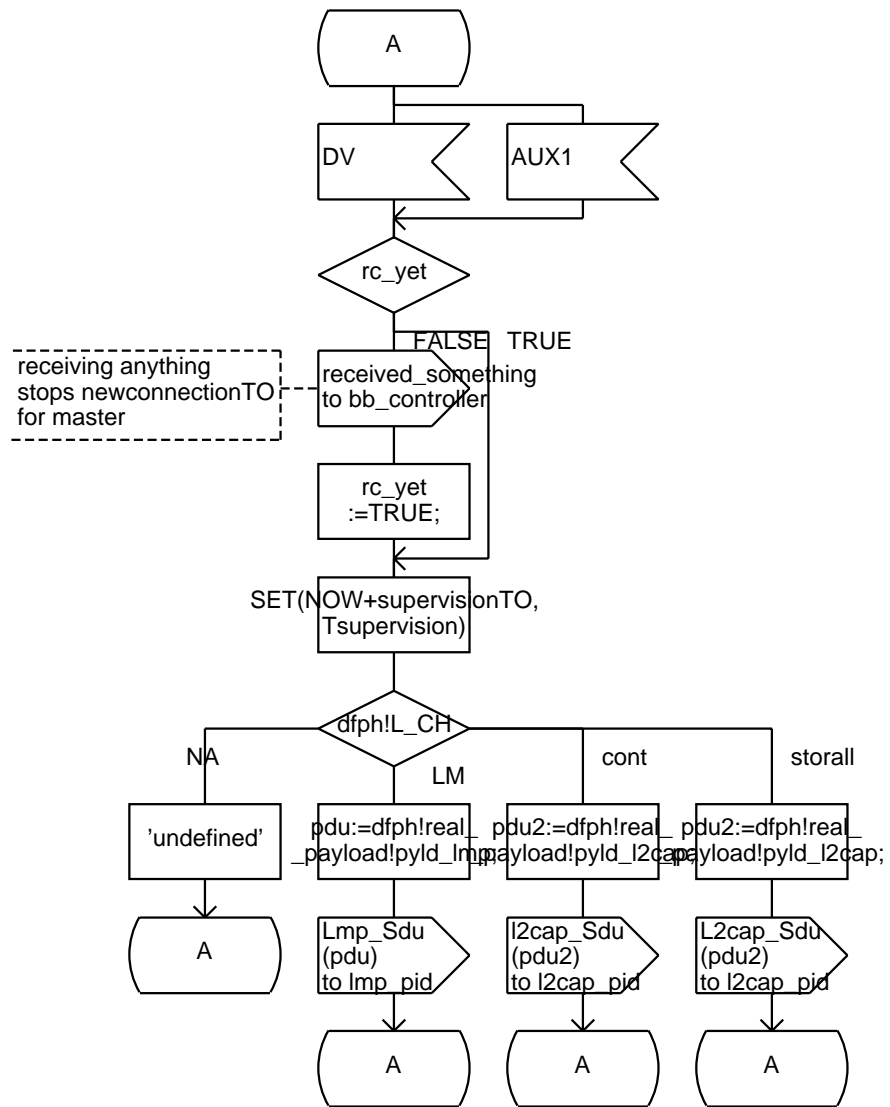


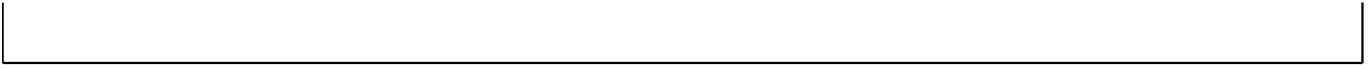


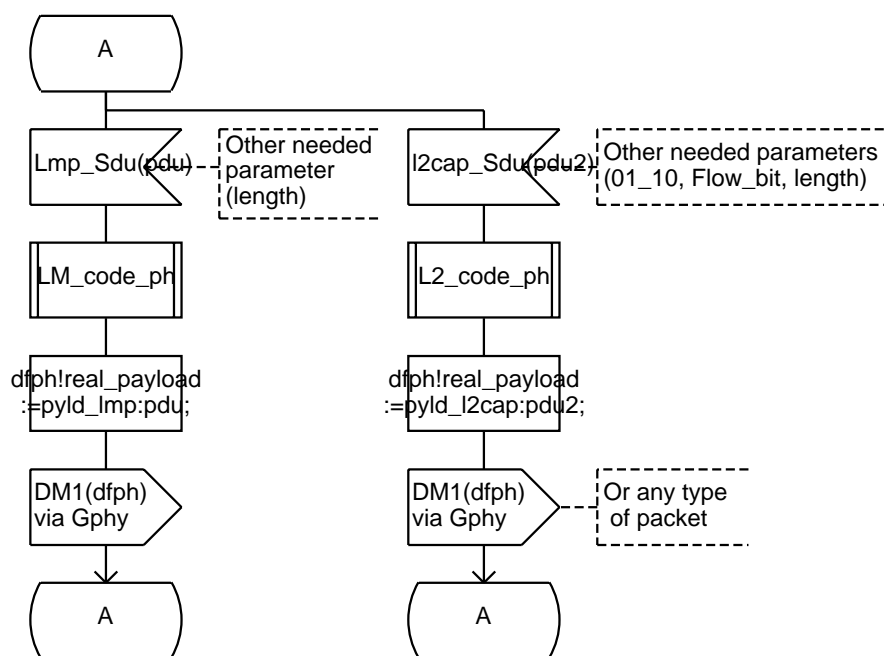


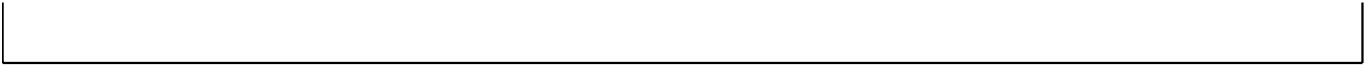


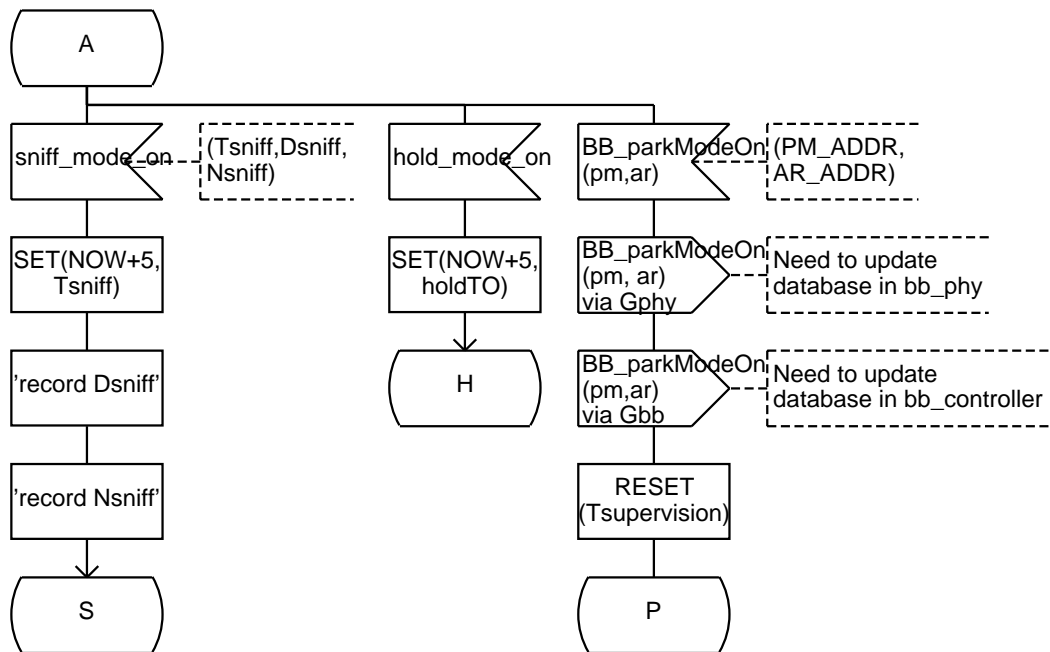


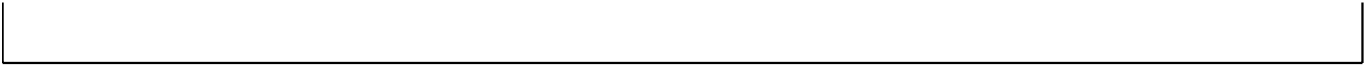


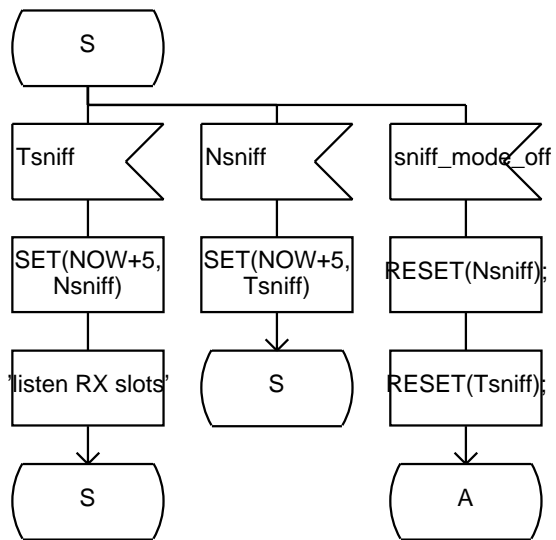


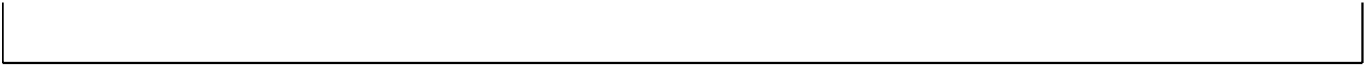


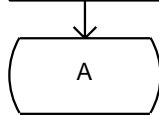
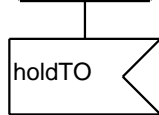
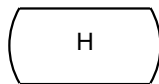




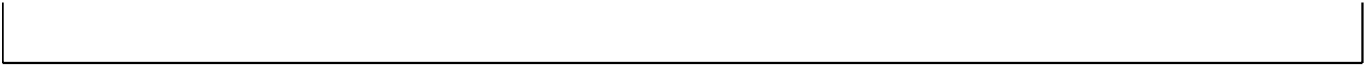


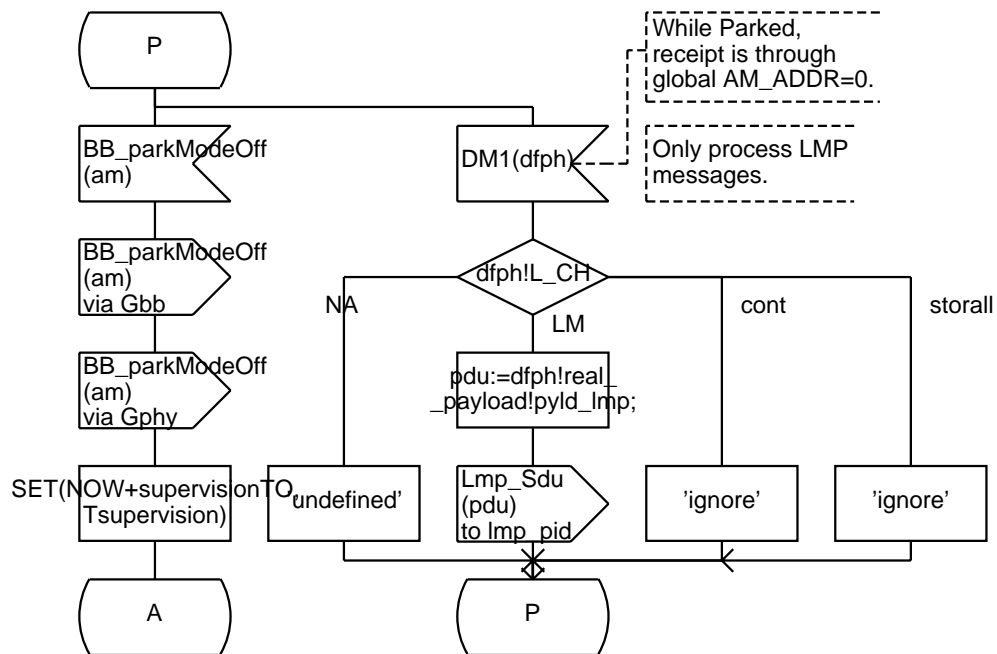


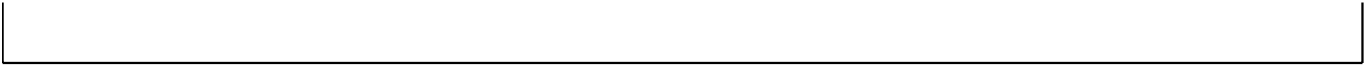


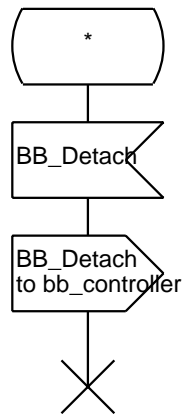


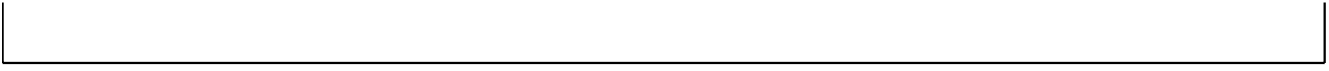
Receipt of any other message
during the hold period is invalid
and shall be ignored.

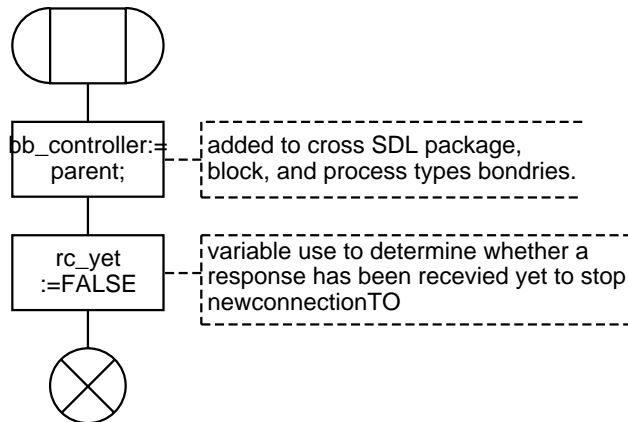




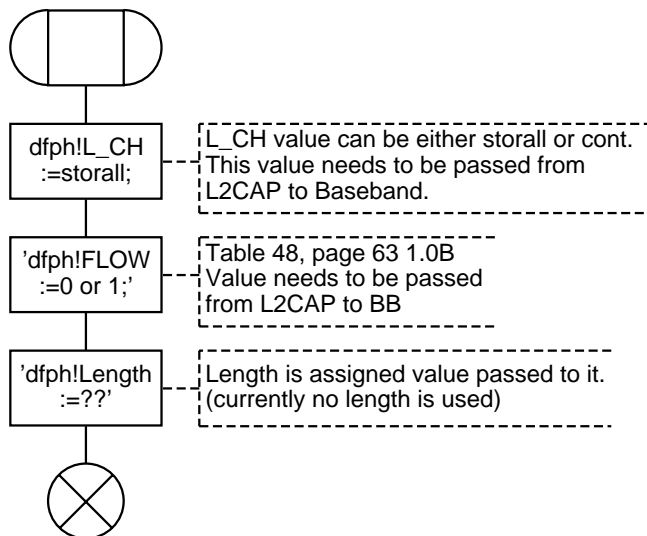




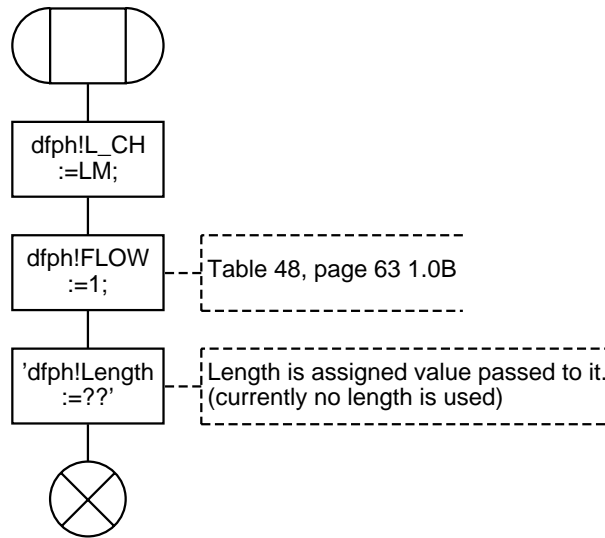




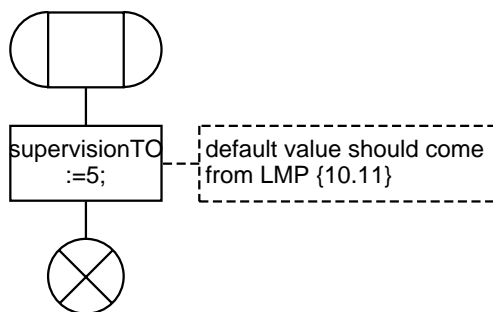
















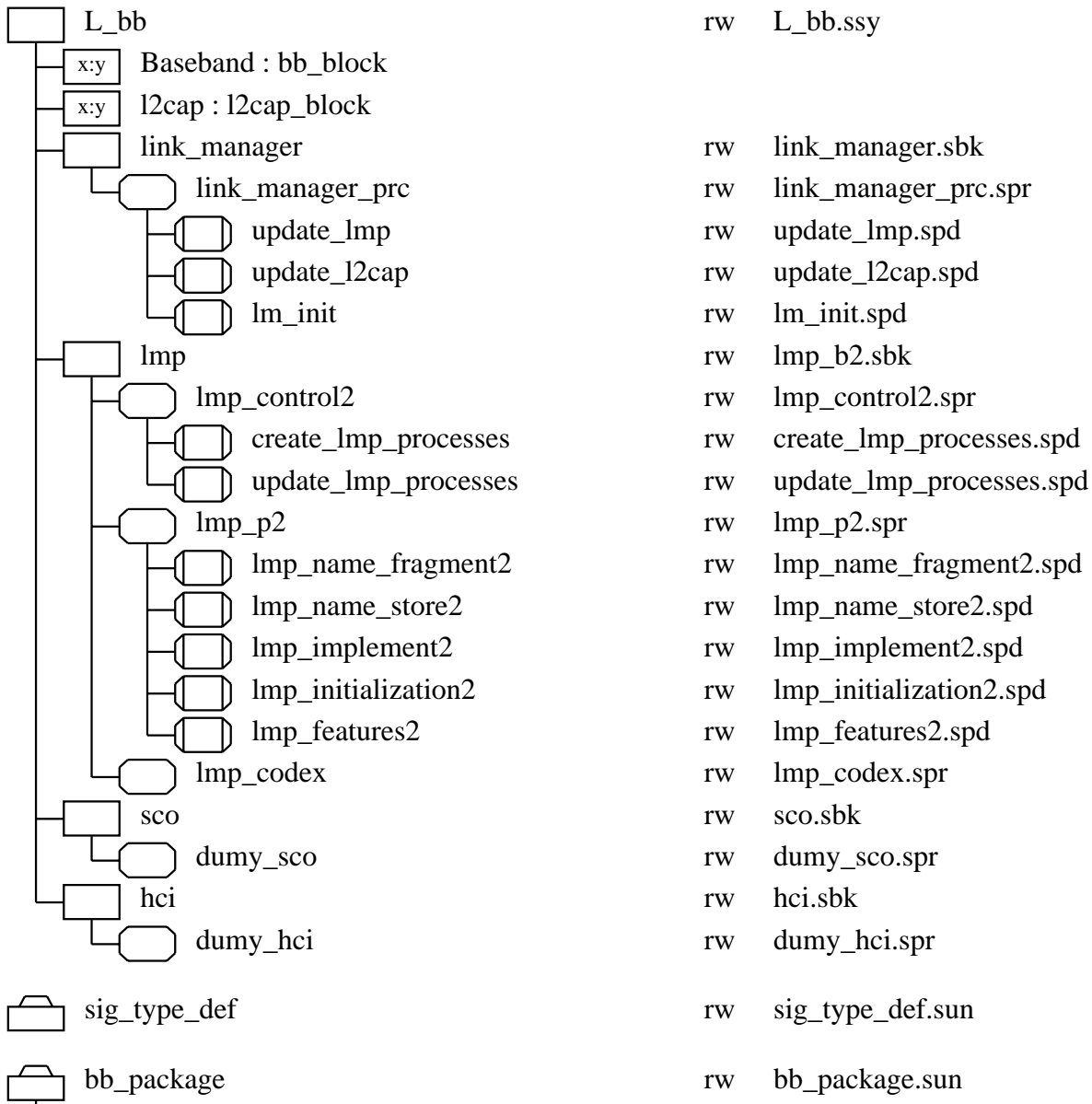
 rw /import/fs1/home/disk1/cypher/BLUETOOTH/new_bb/untitled.sdt

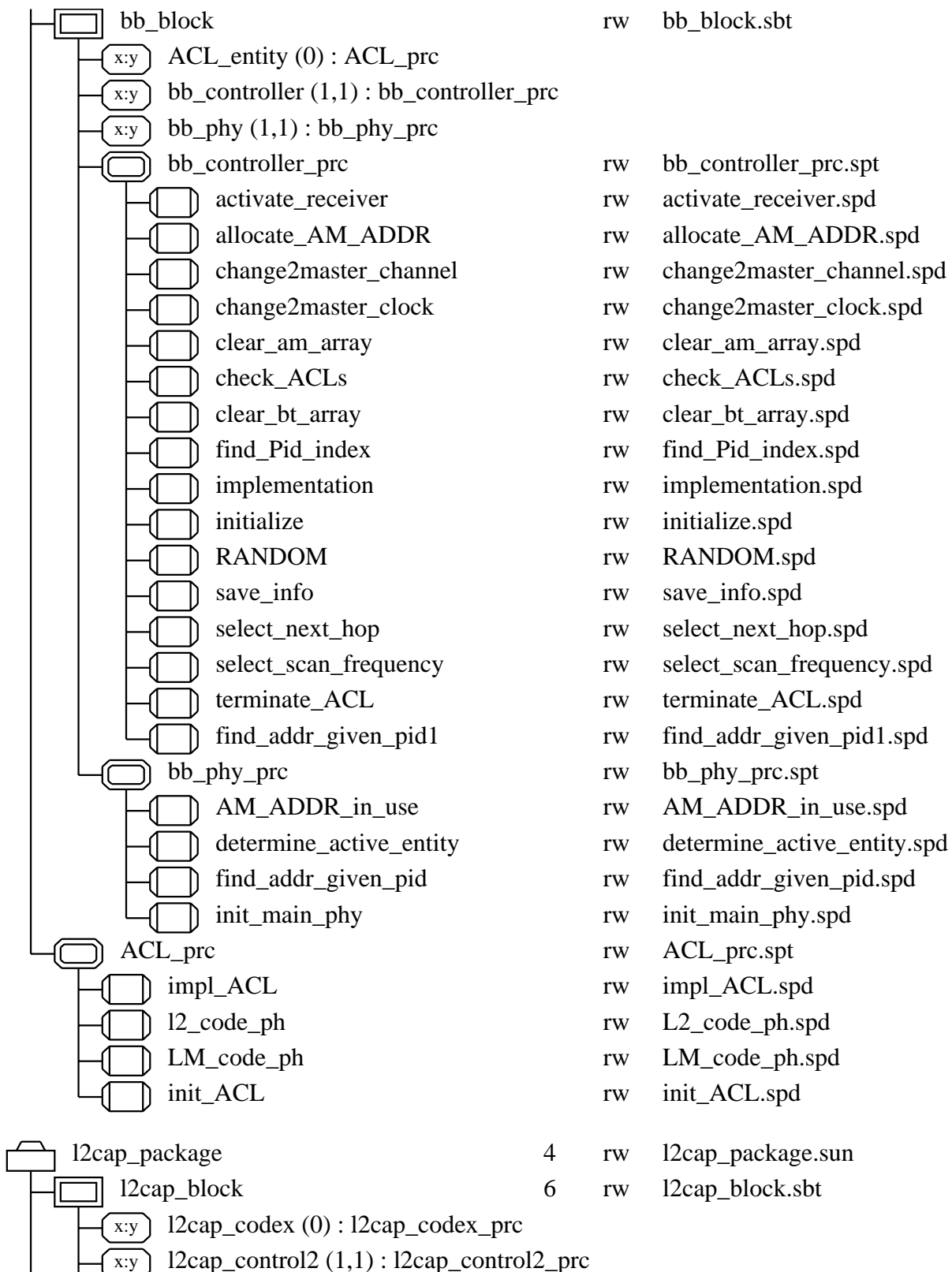
 rw /import/fs1/home/disk1/cypher/BLUETOOTH/new_bb/

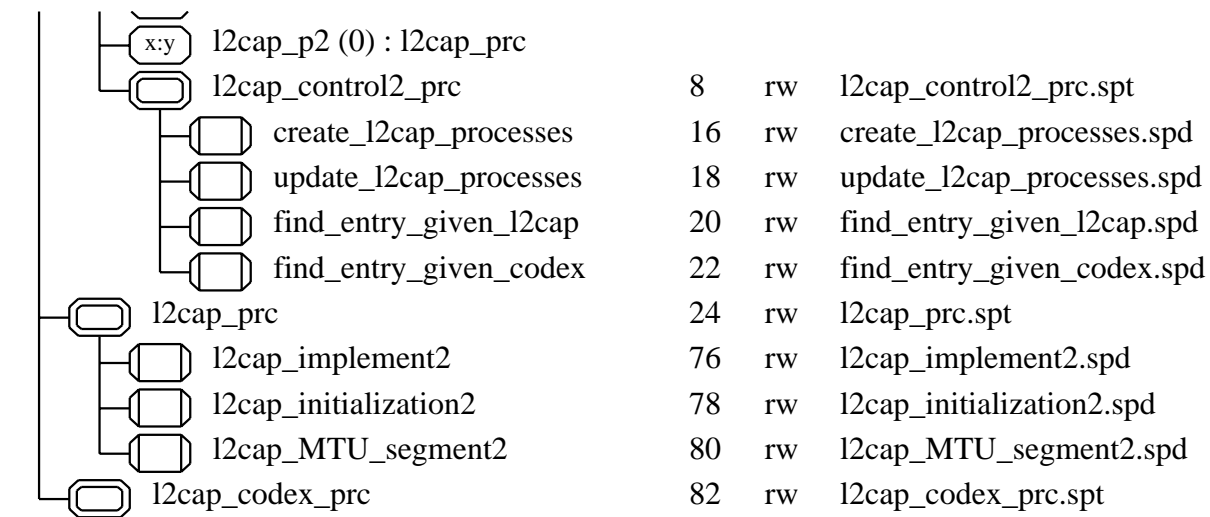
— Analysis Model

— Used Files

— new_bb







Other Documents

USE sig_type_def;

package l2cap_package

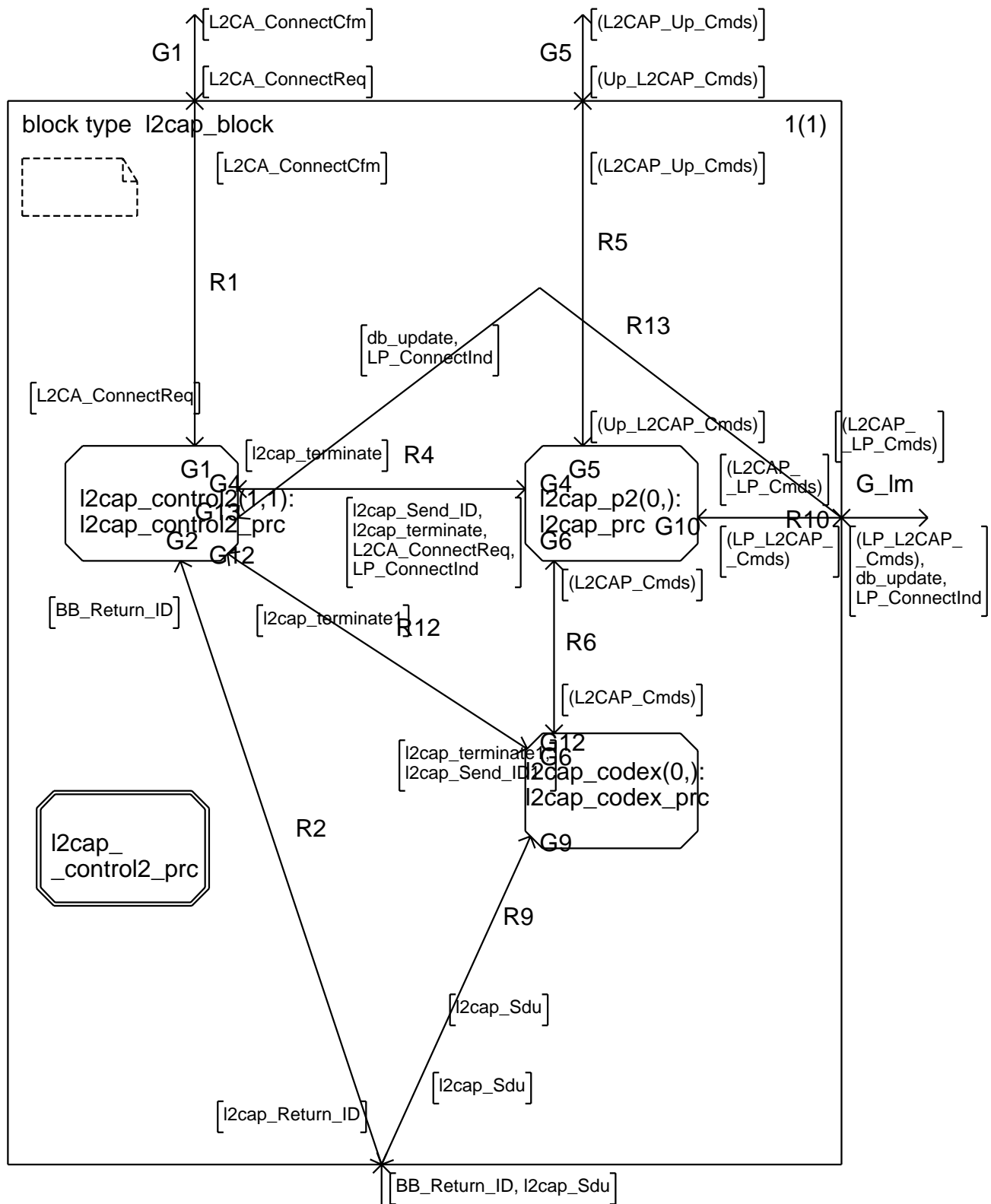
1(1)



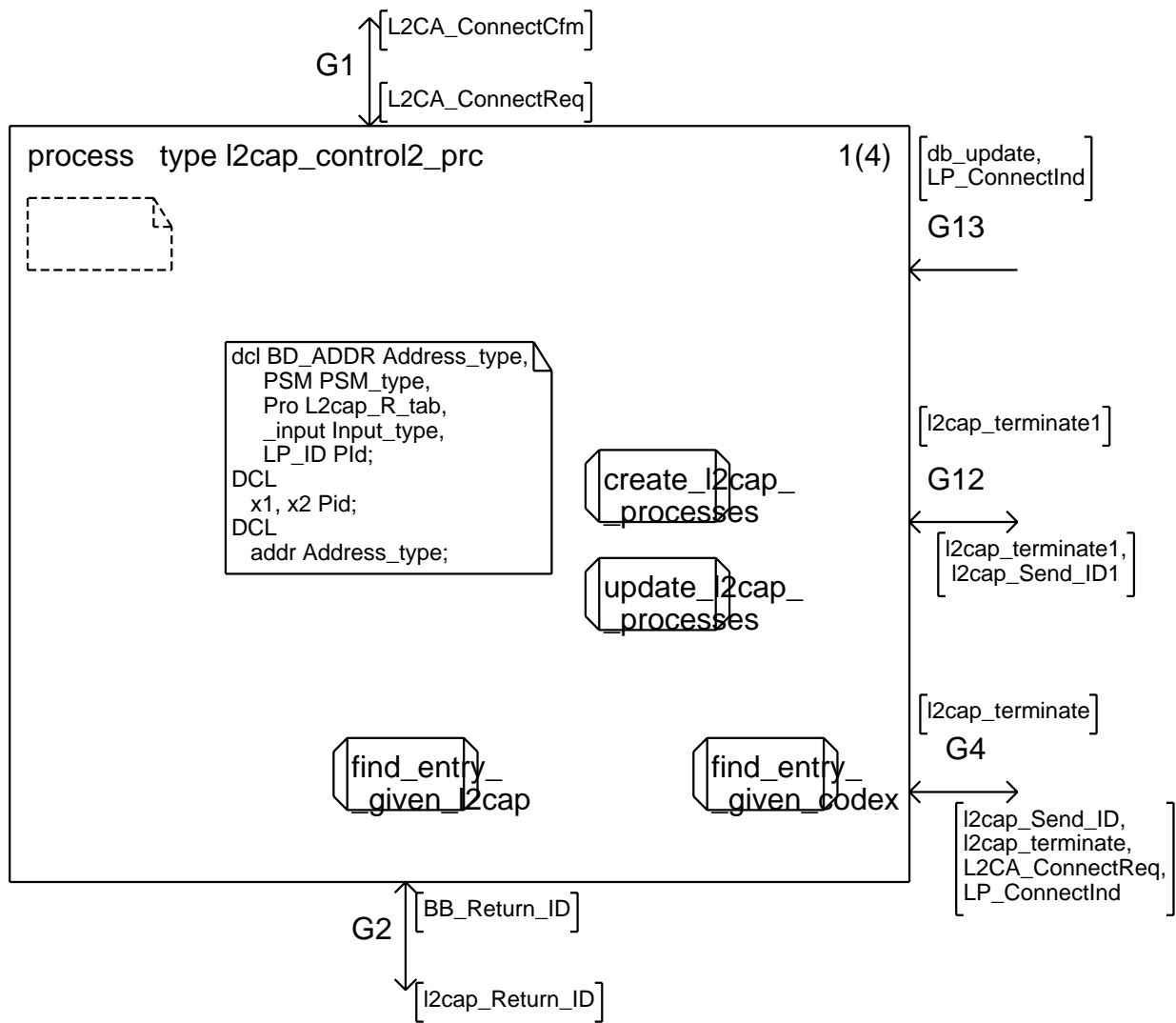
l2cap_block

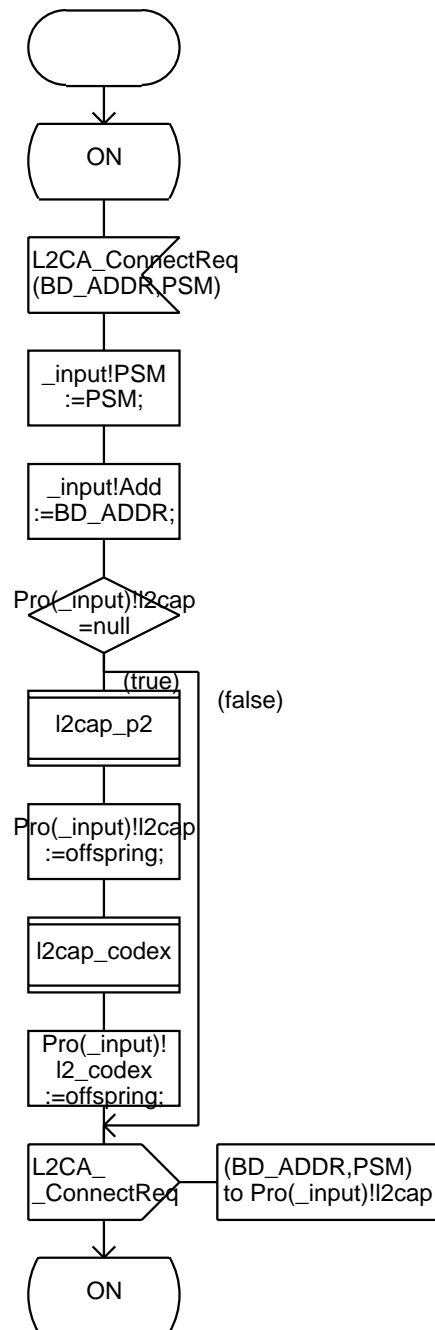
l2cap_prc

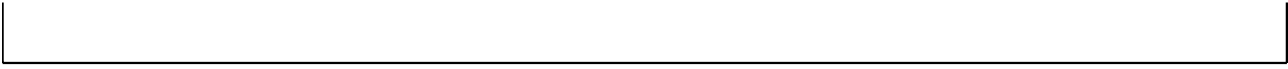
l2cap_codex_
_prc

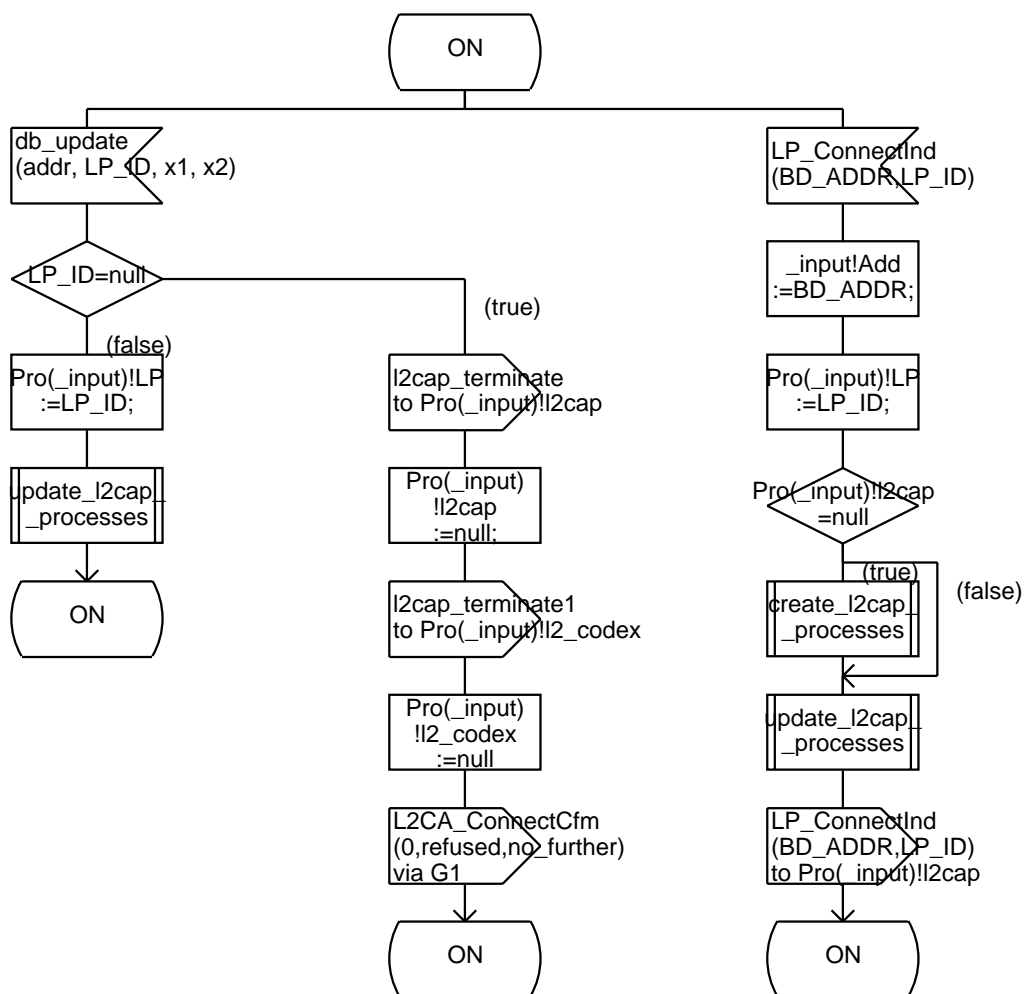


G2910
↓
[l2cap_Return_ID,l2cap_Sdu]

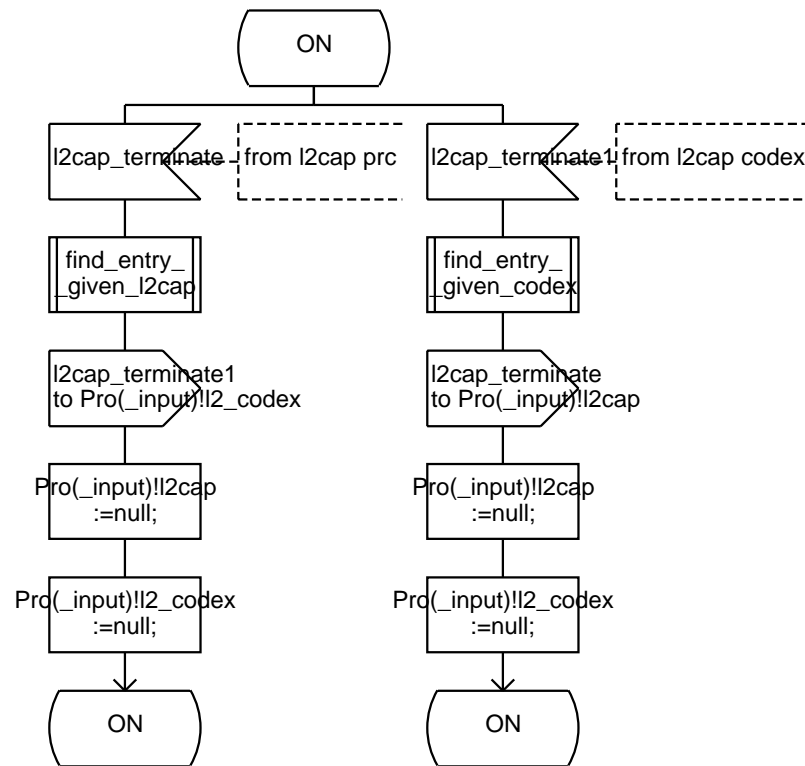








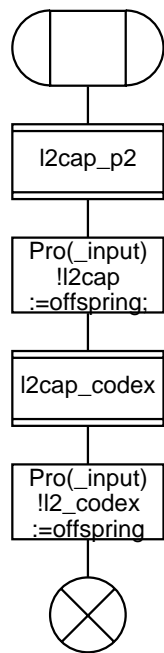




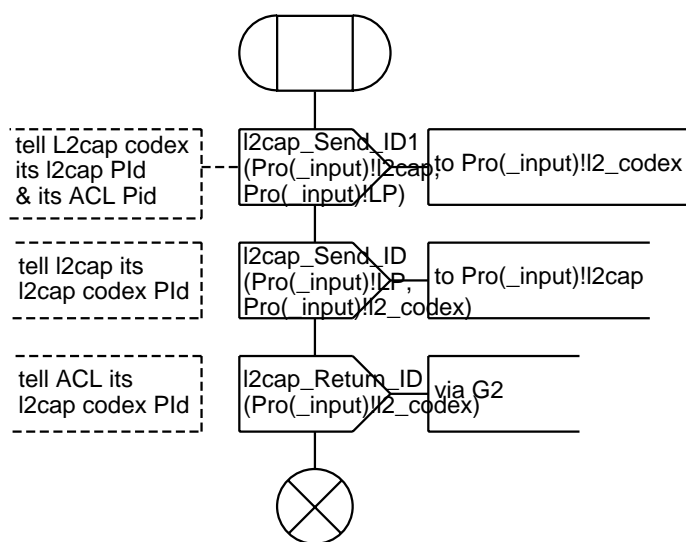


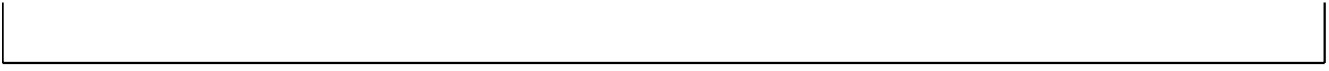
procedure create_l2cap_processes

1(1)









procedure find_entry_given_l2cap

1(1)

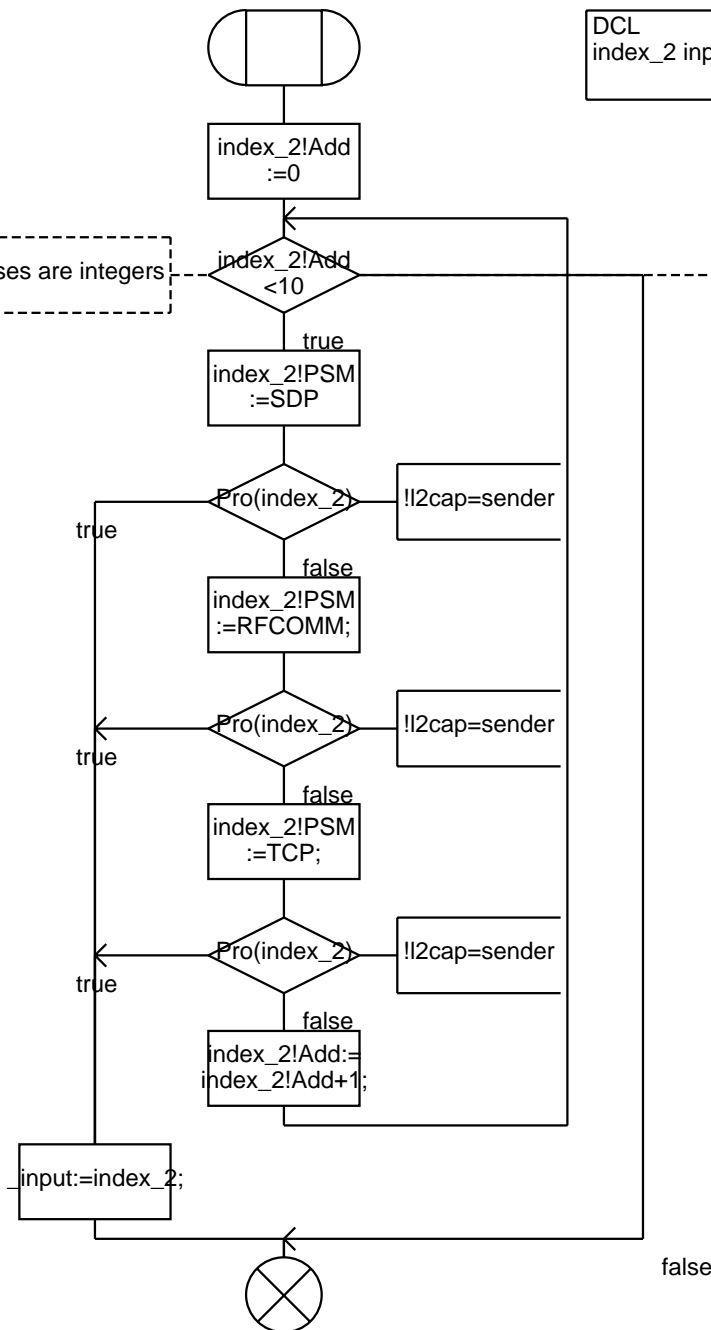


SDL based search
to link/unlink SDL
Processes.

DCL
index_2 input_type;

Assuming Addresses are integers

An ERROR has occurred, if
this path is reached.





procedure find_entry_given_codex

1(1)

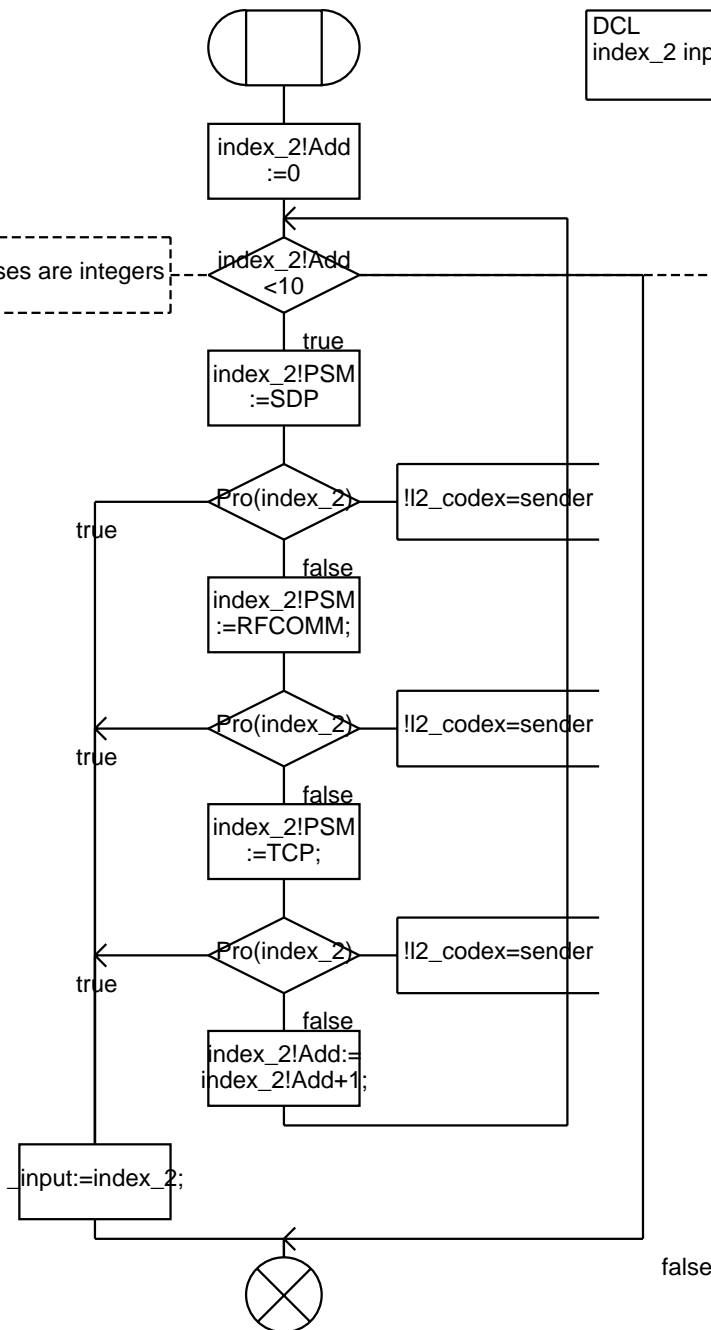


SDL based search
to link/unlink SDL
Processes.

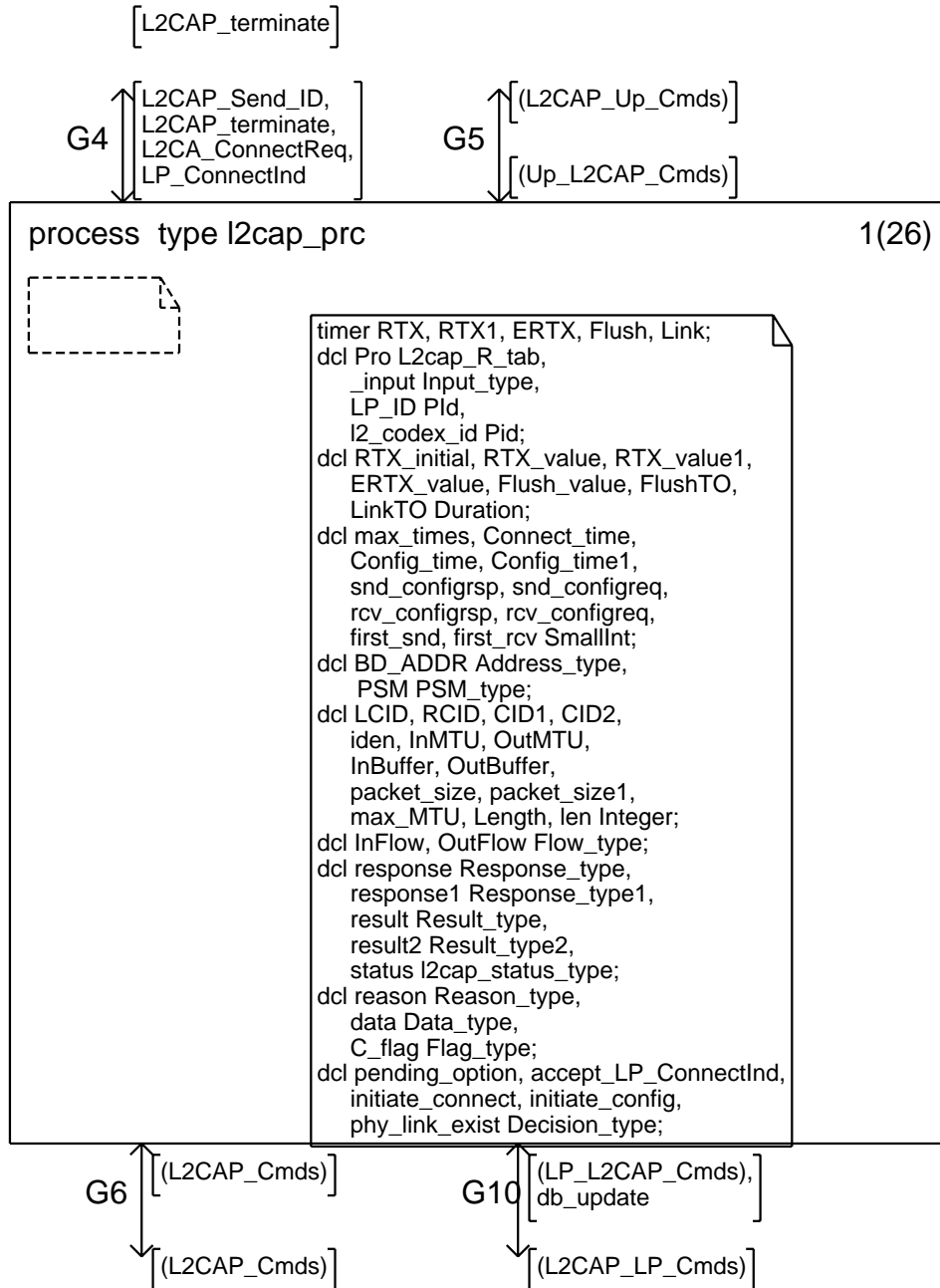
DCL
index_2 input_type;

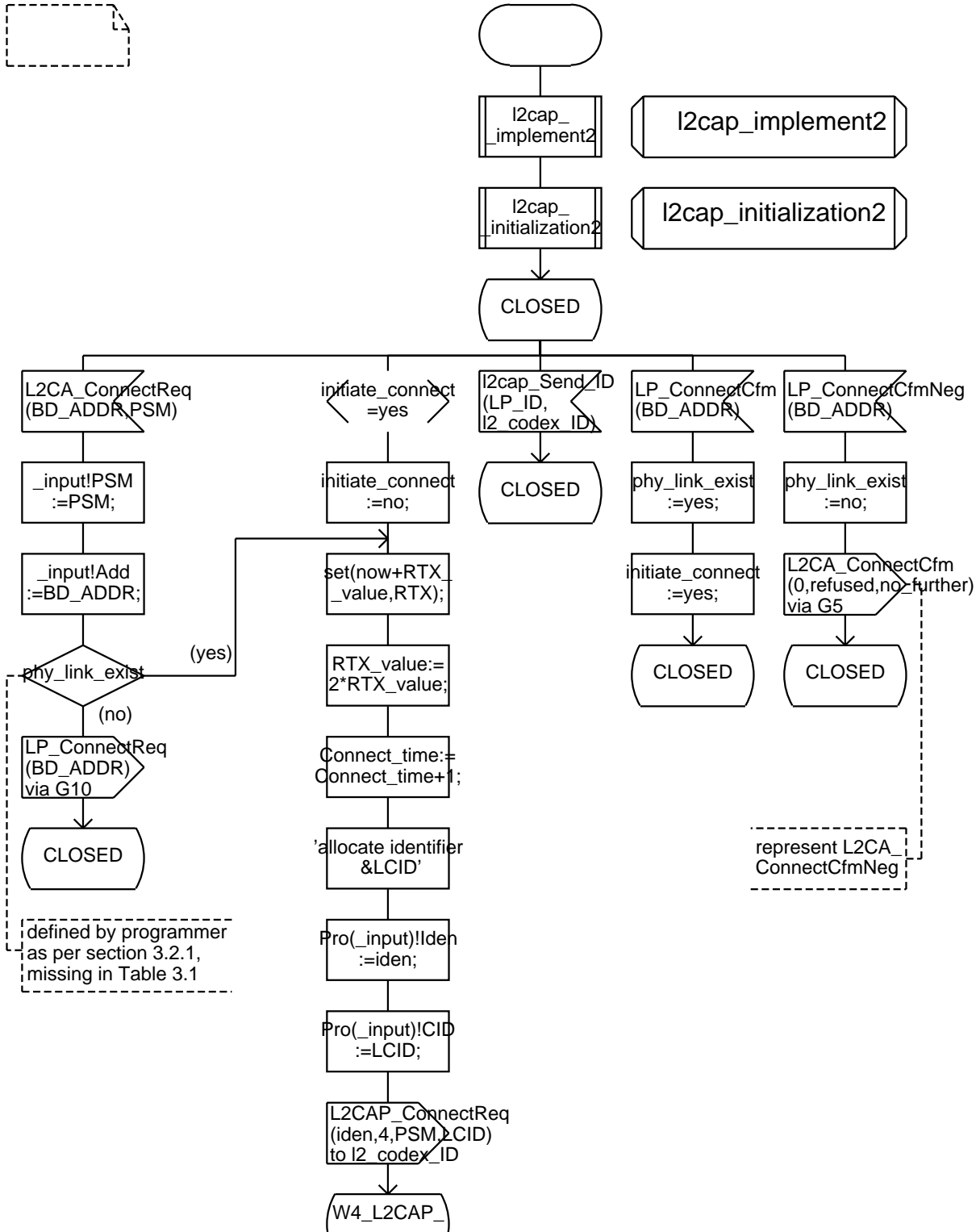
Assuming Addresses are integers

An ERROR has occurred, if
this path is reached.



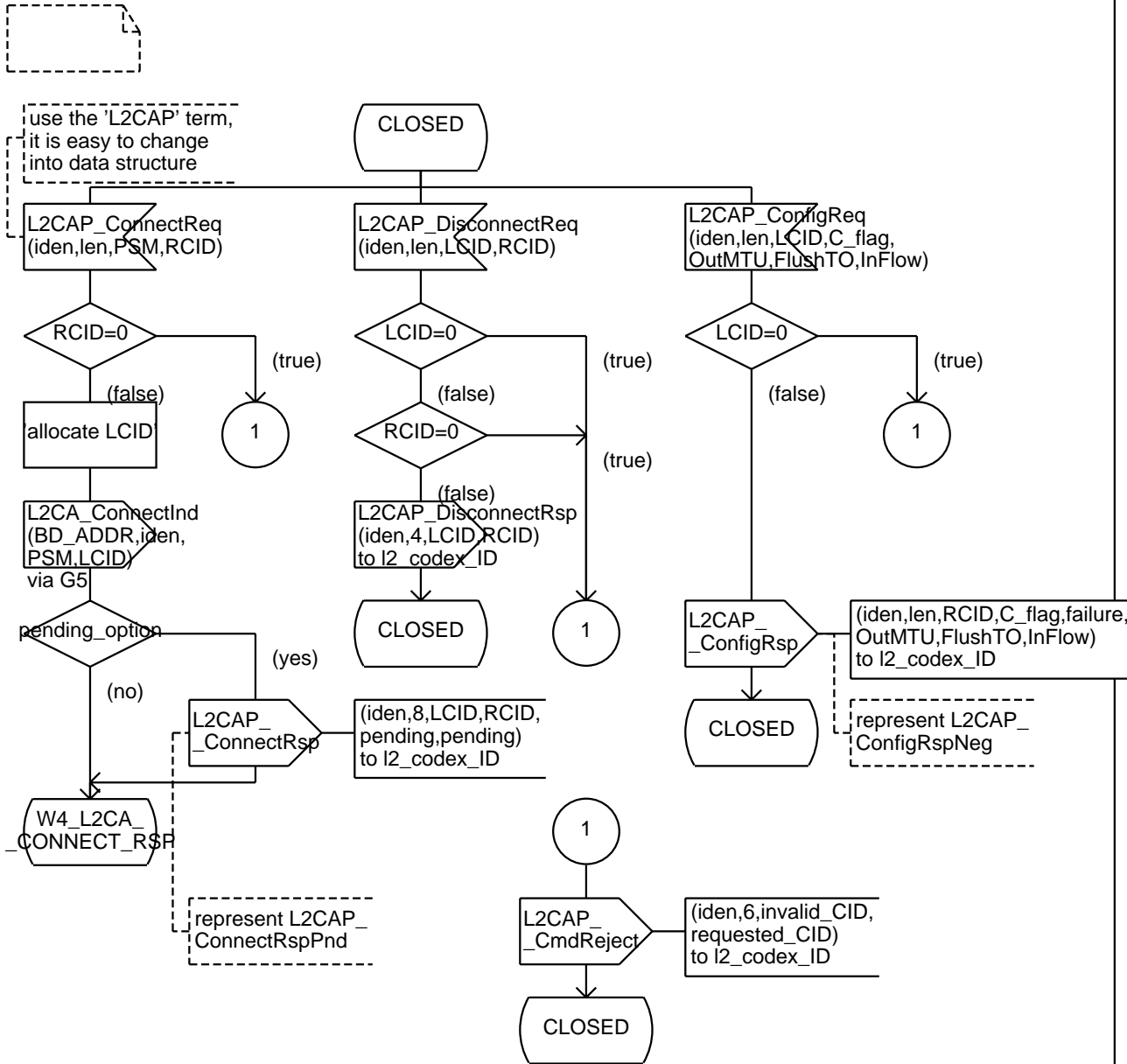


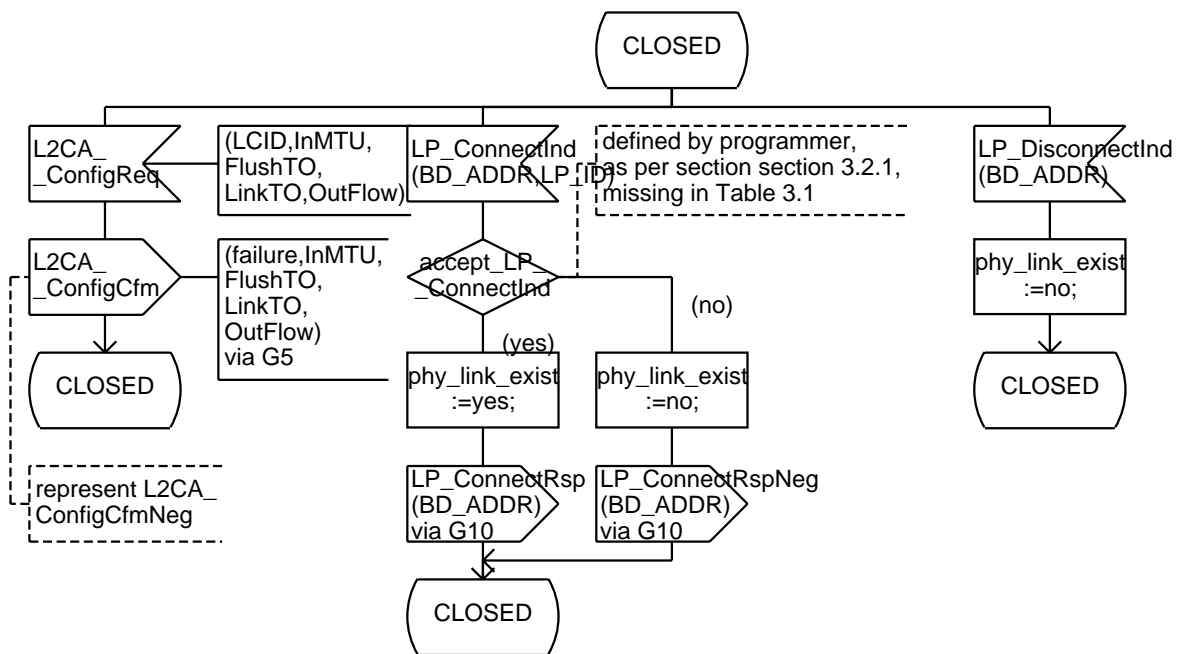


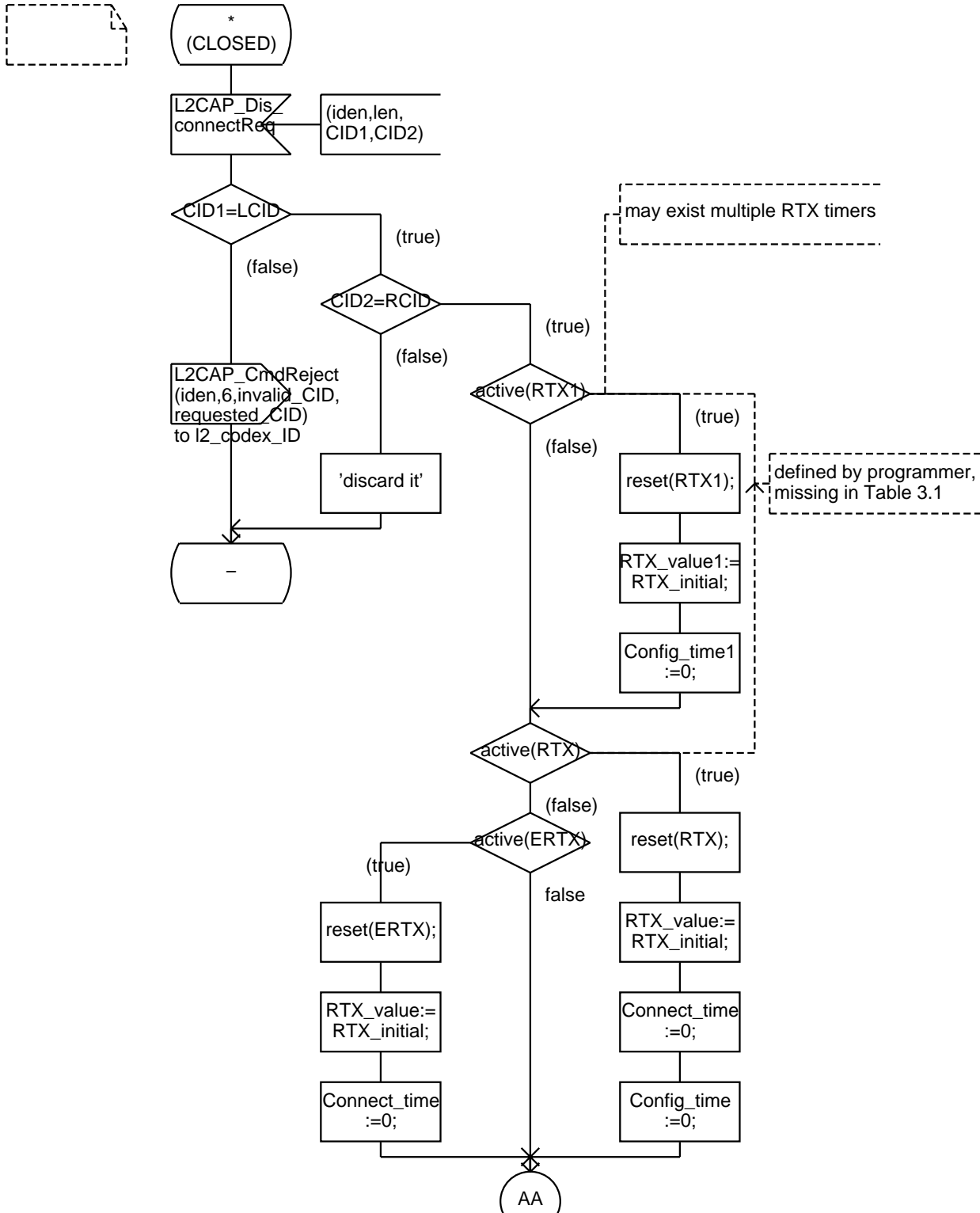


_CONNECT_RSP

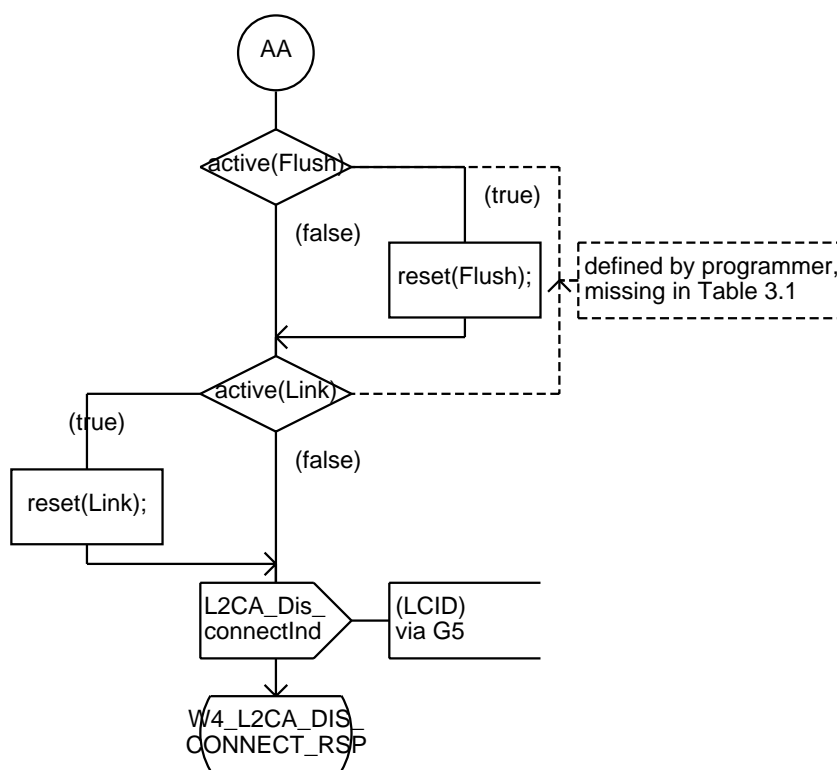
A diagram consisting of a horizontal line. Above the line, on the left side, is a bracket. Above the bracket is the text "_CONNECT_RSP".



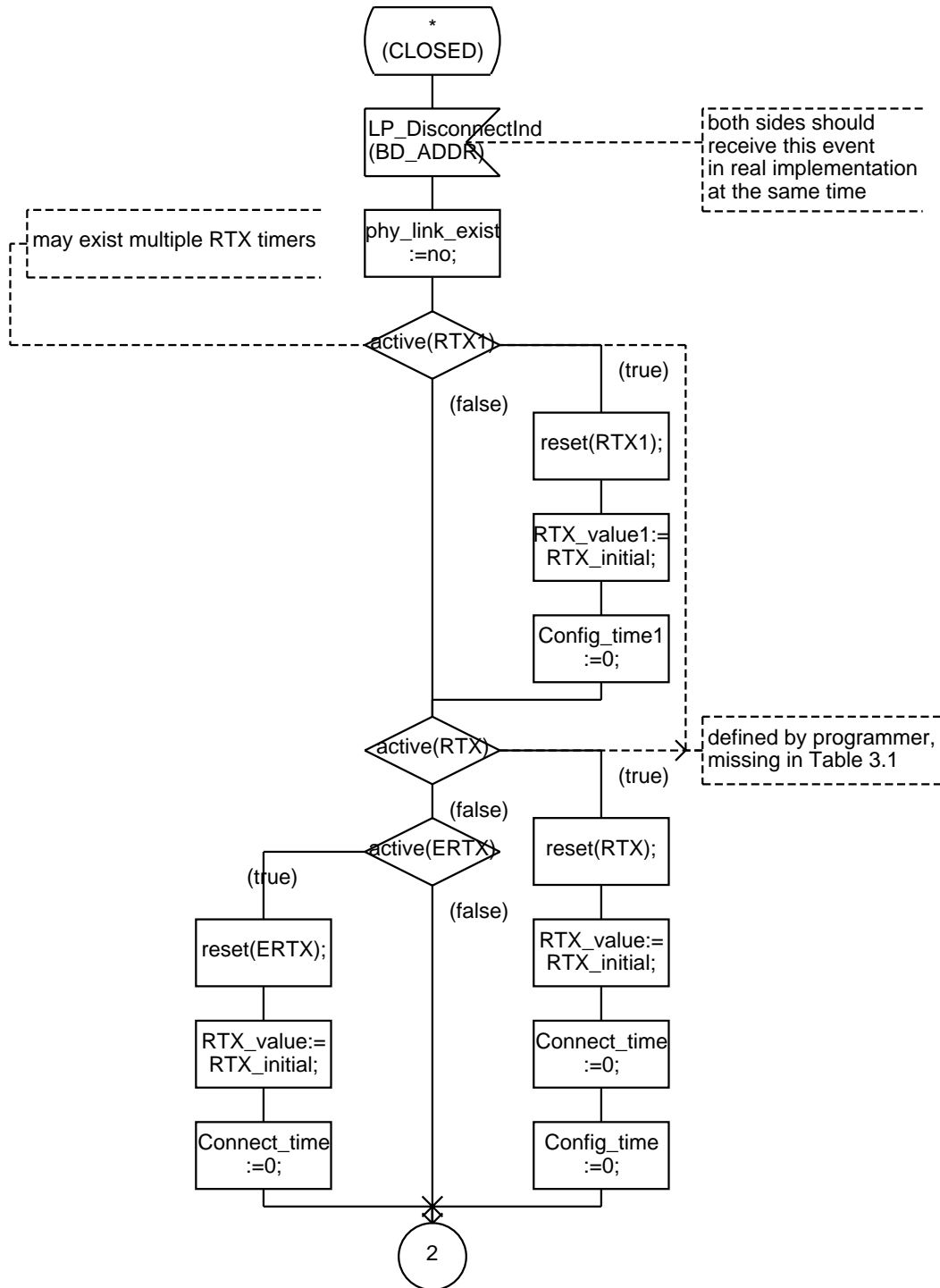


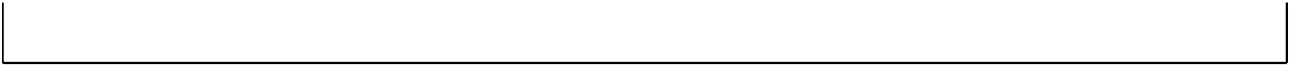


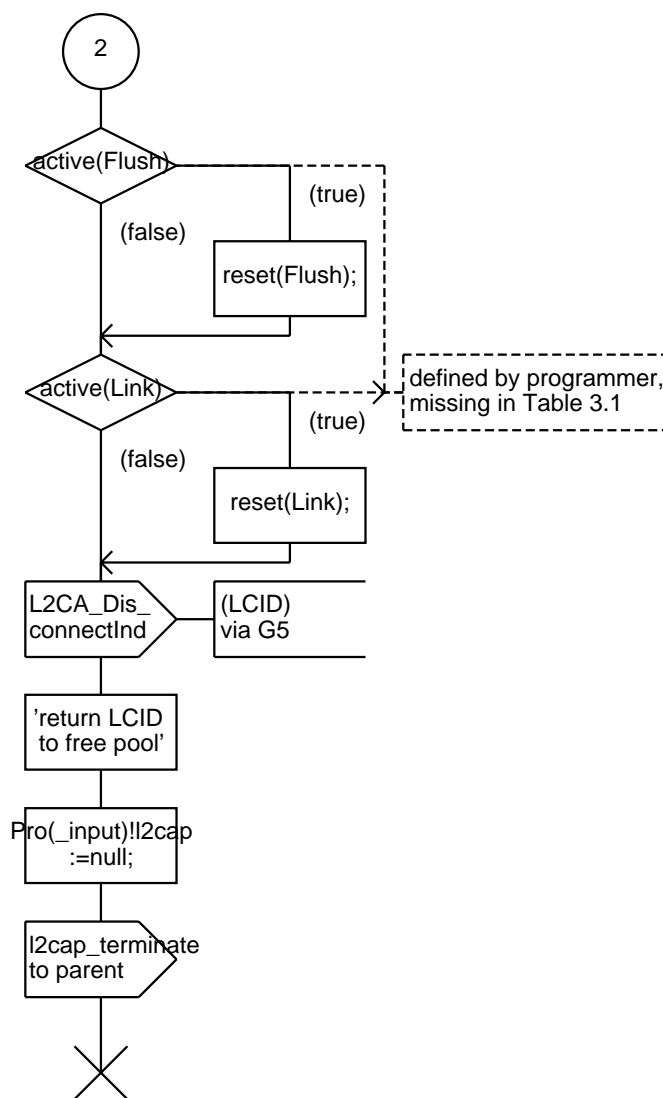


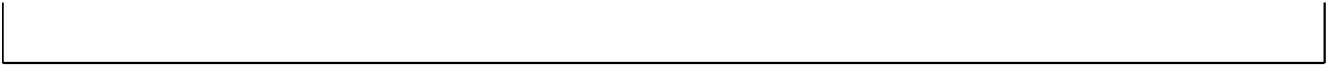


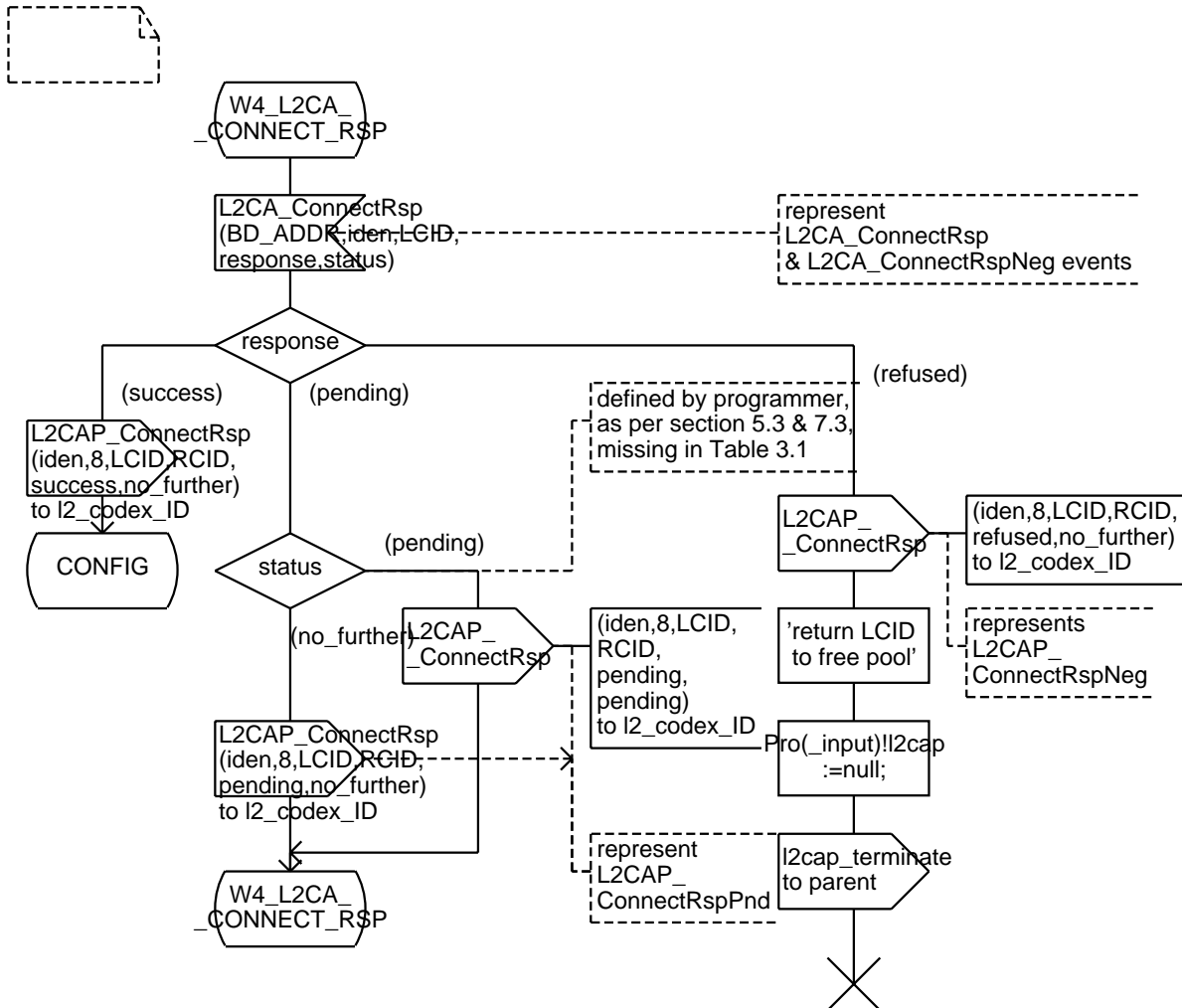


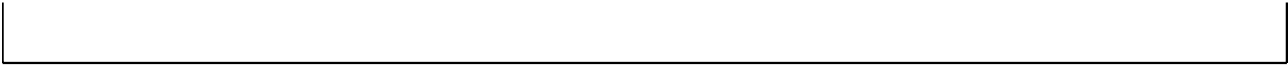


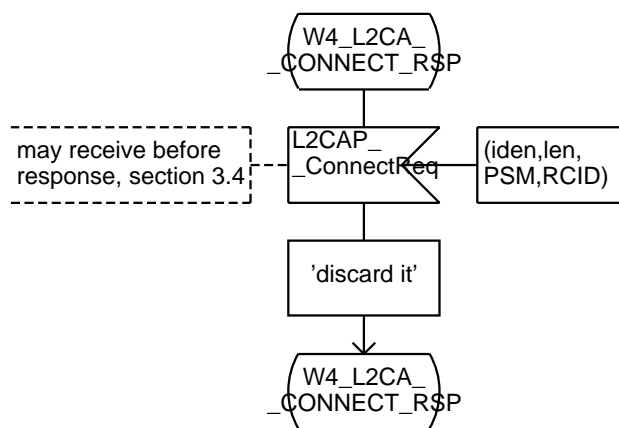




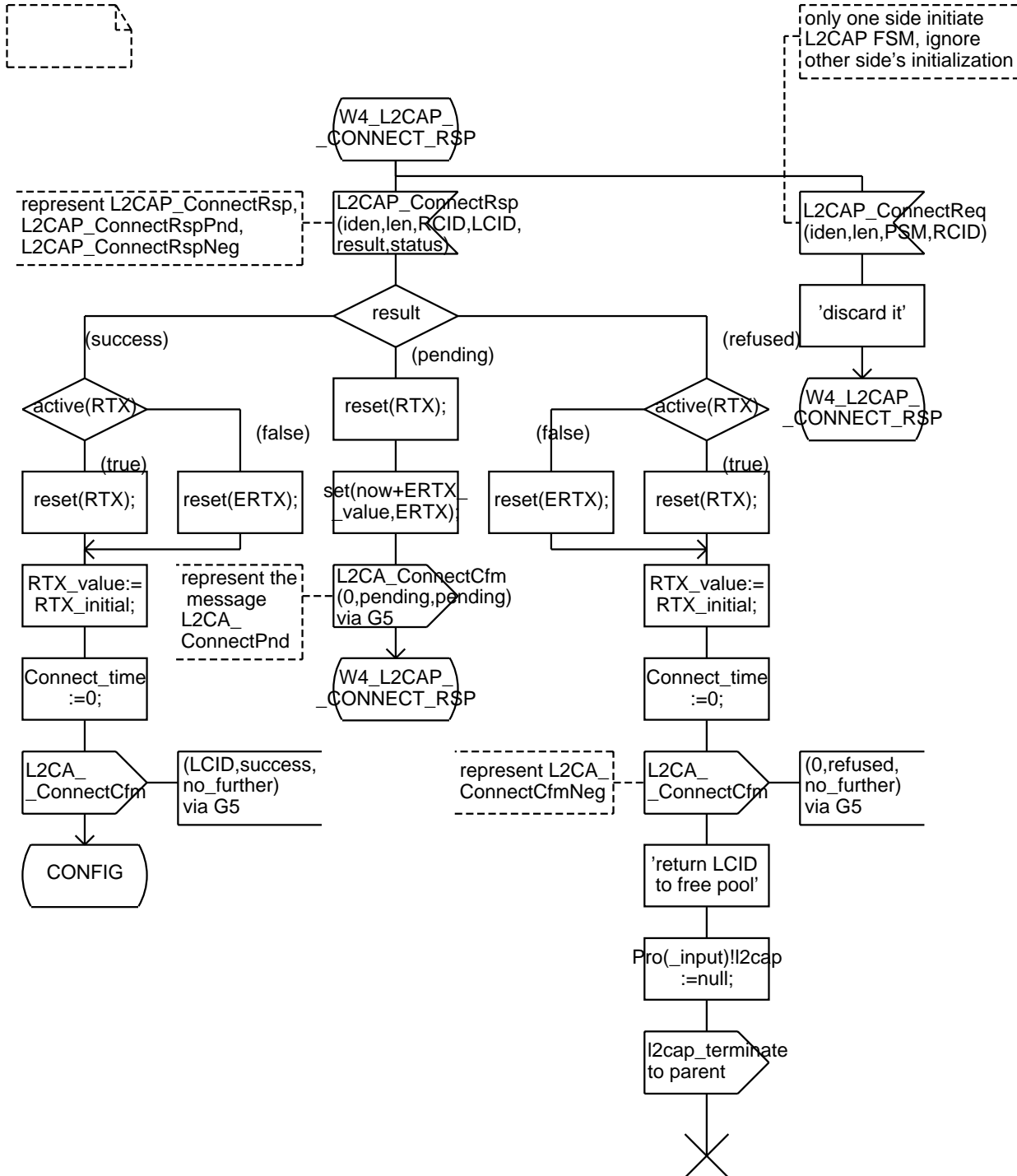


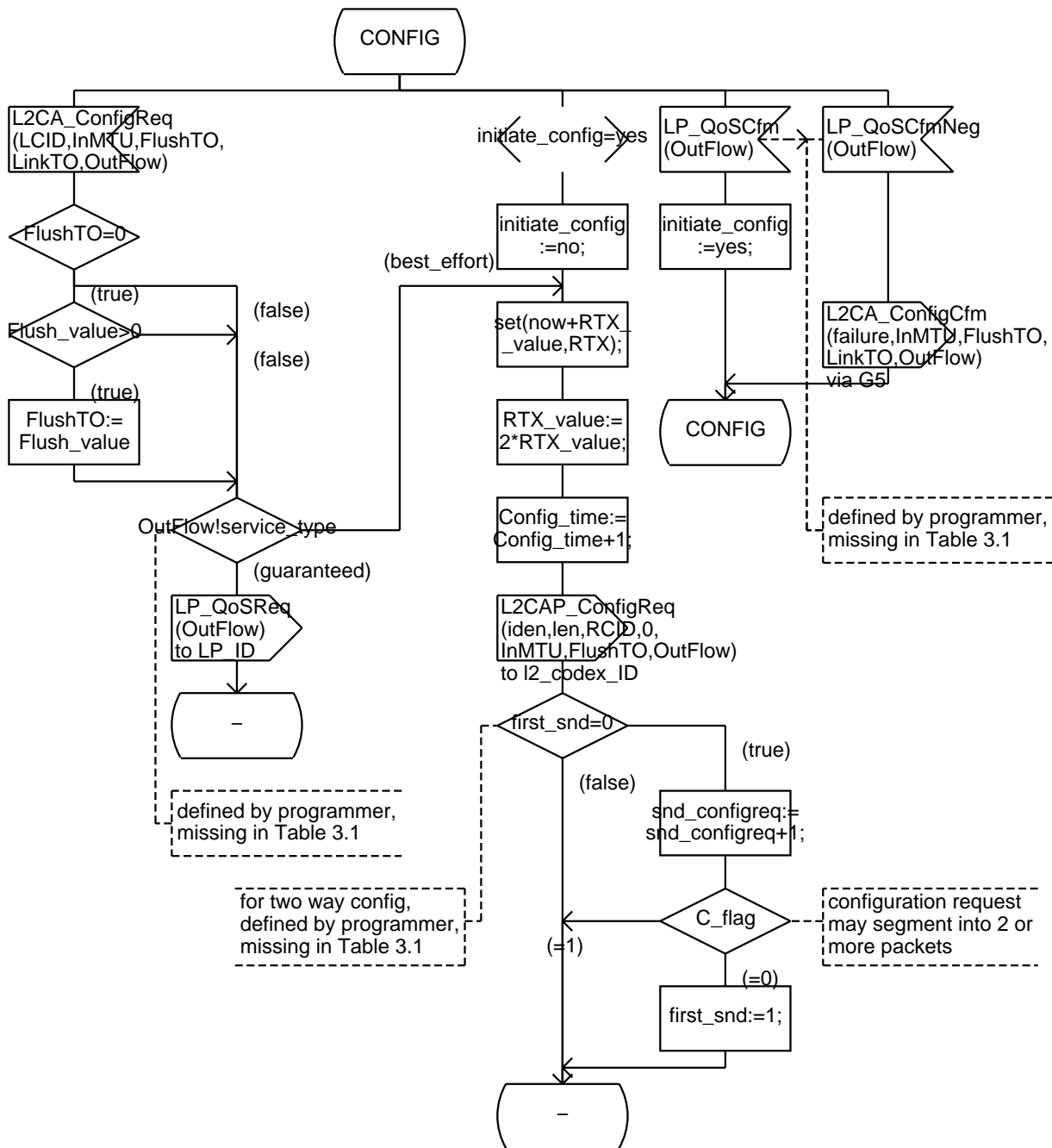


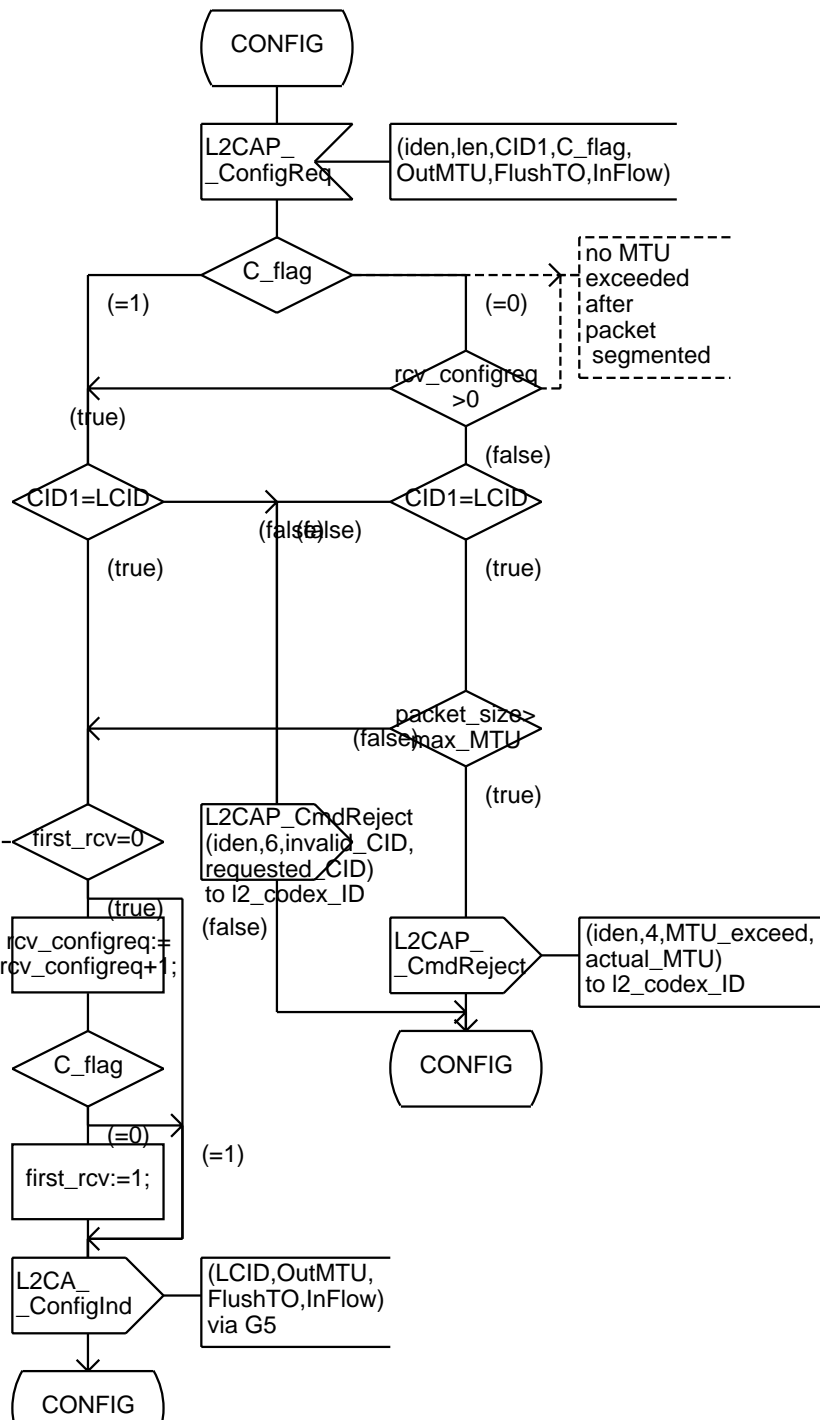








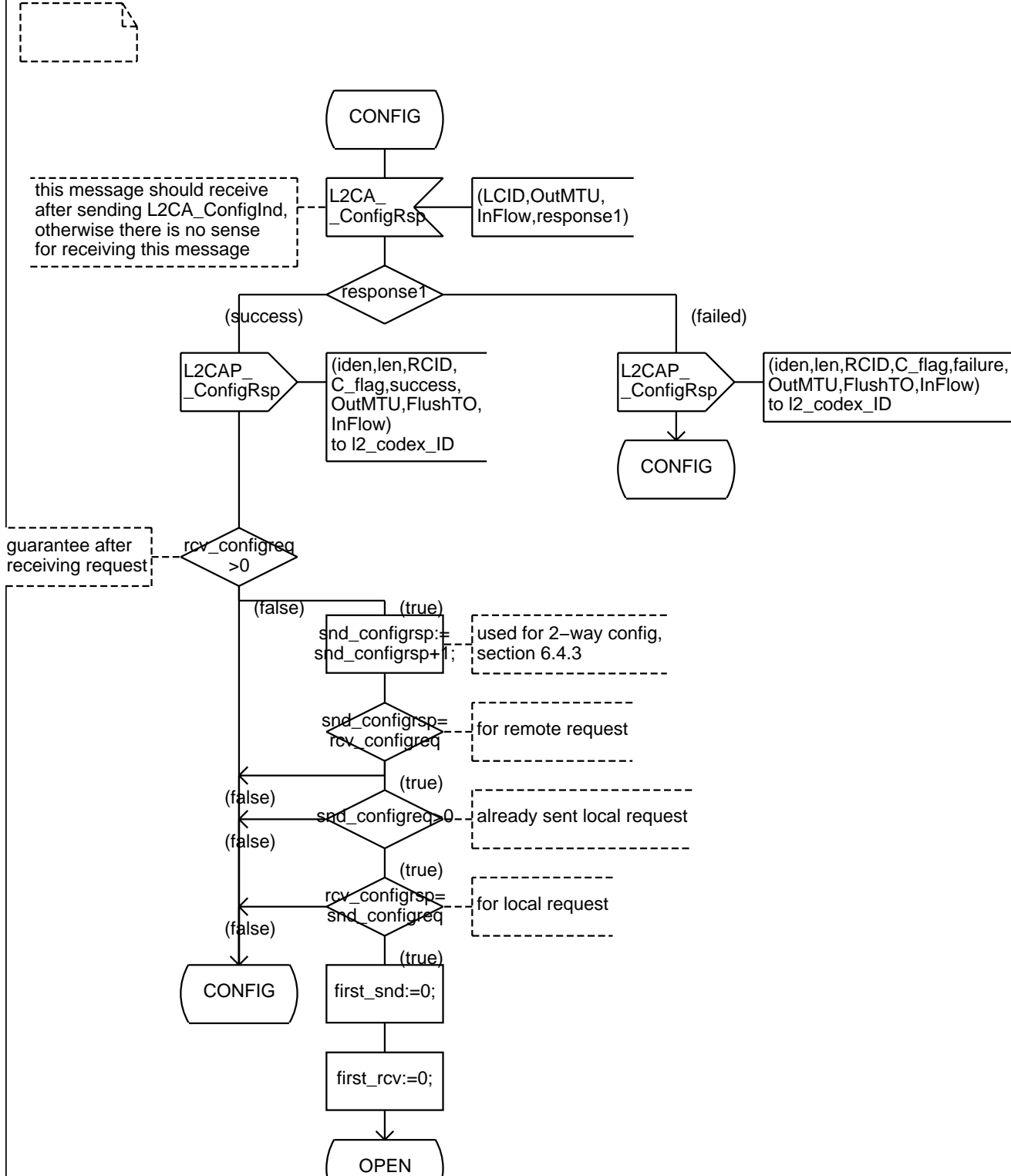




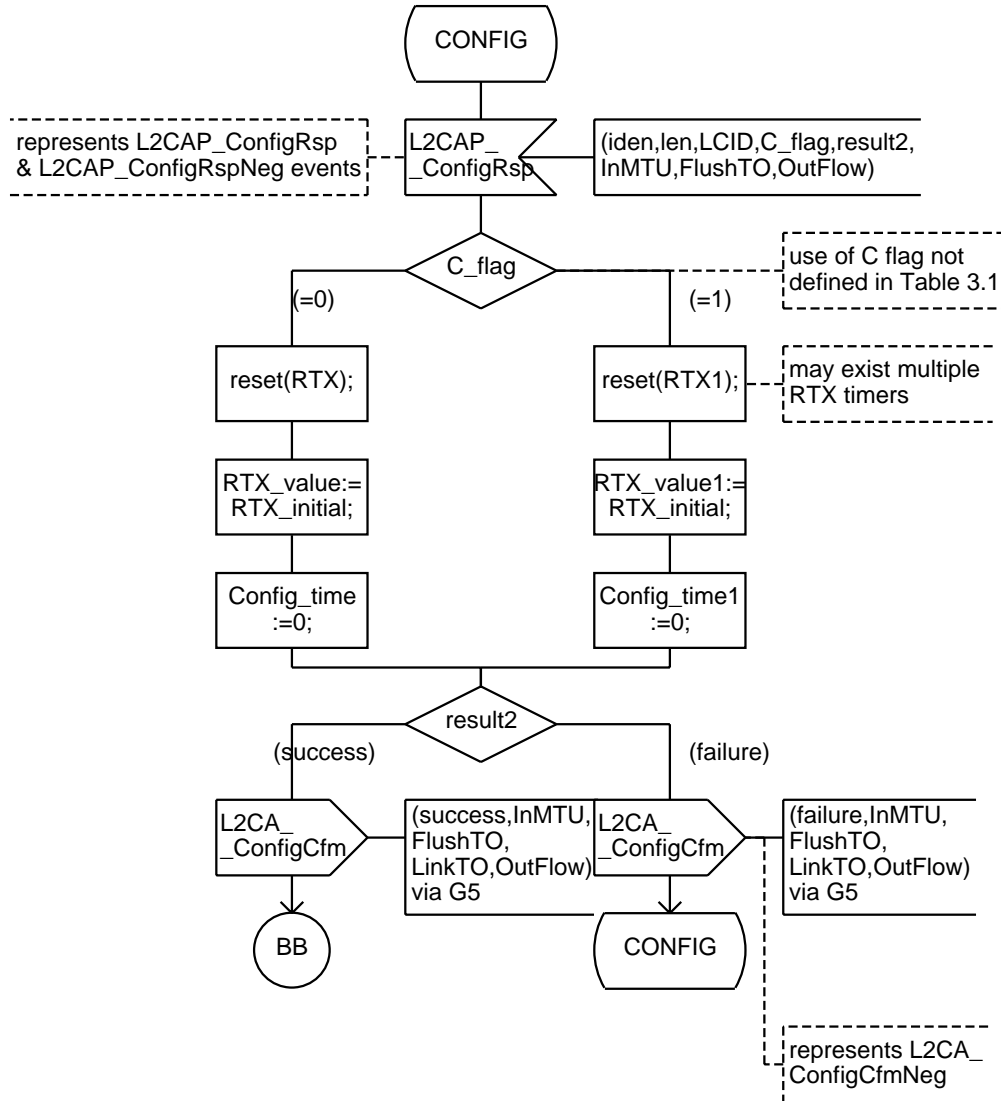
two way
fig, defined
programmer,
sing in Table 3.1

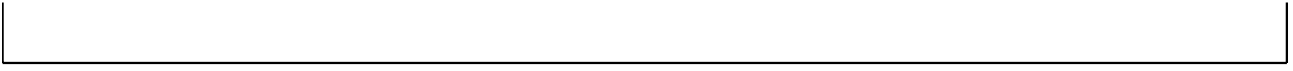
no MTU
exceeded
after
packet
segmented

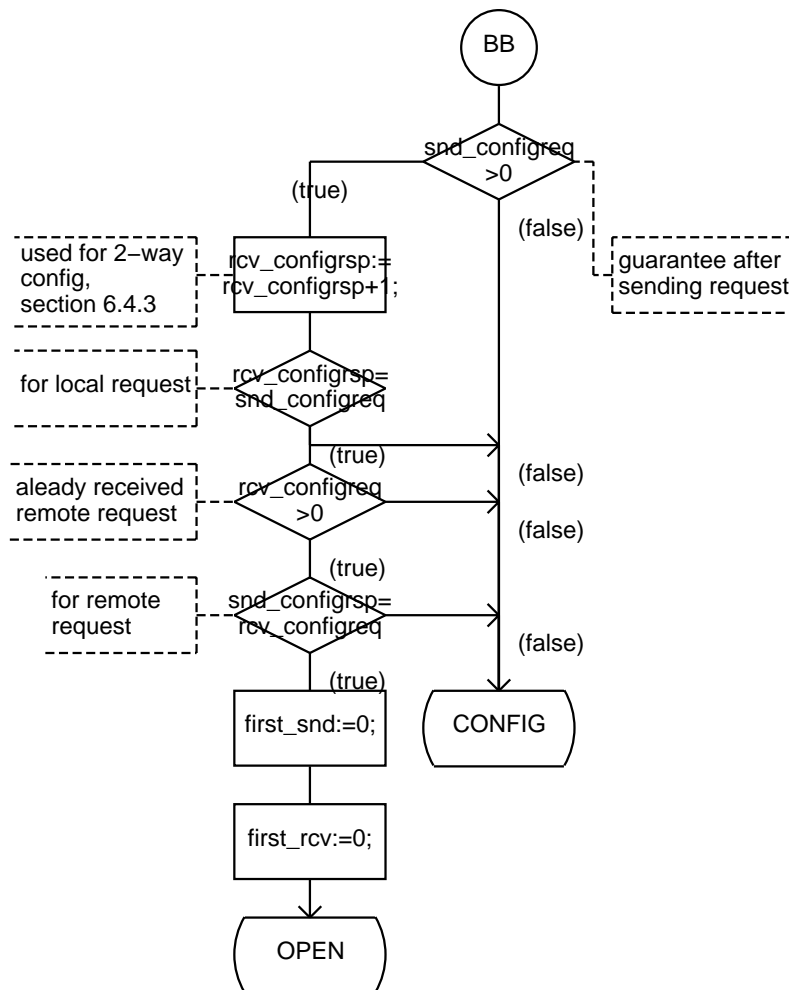


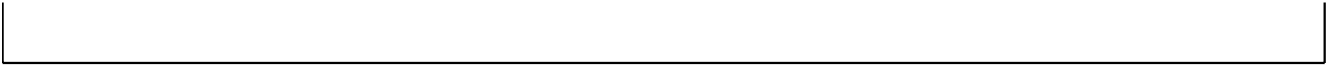


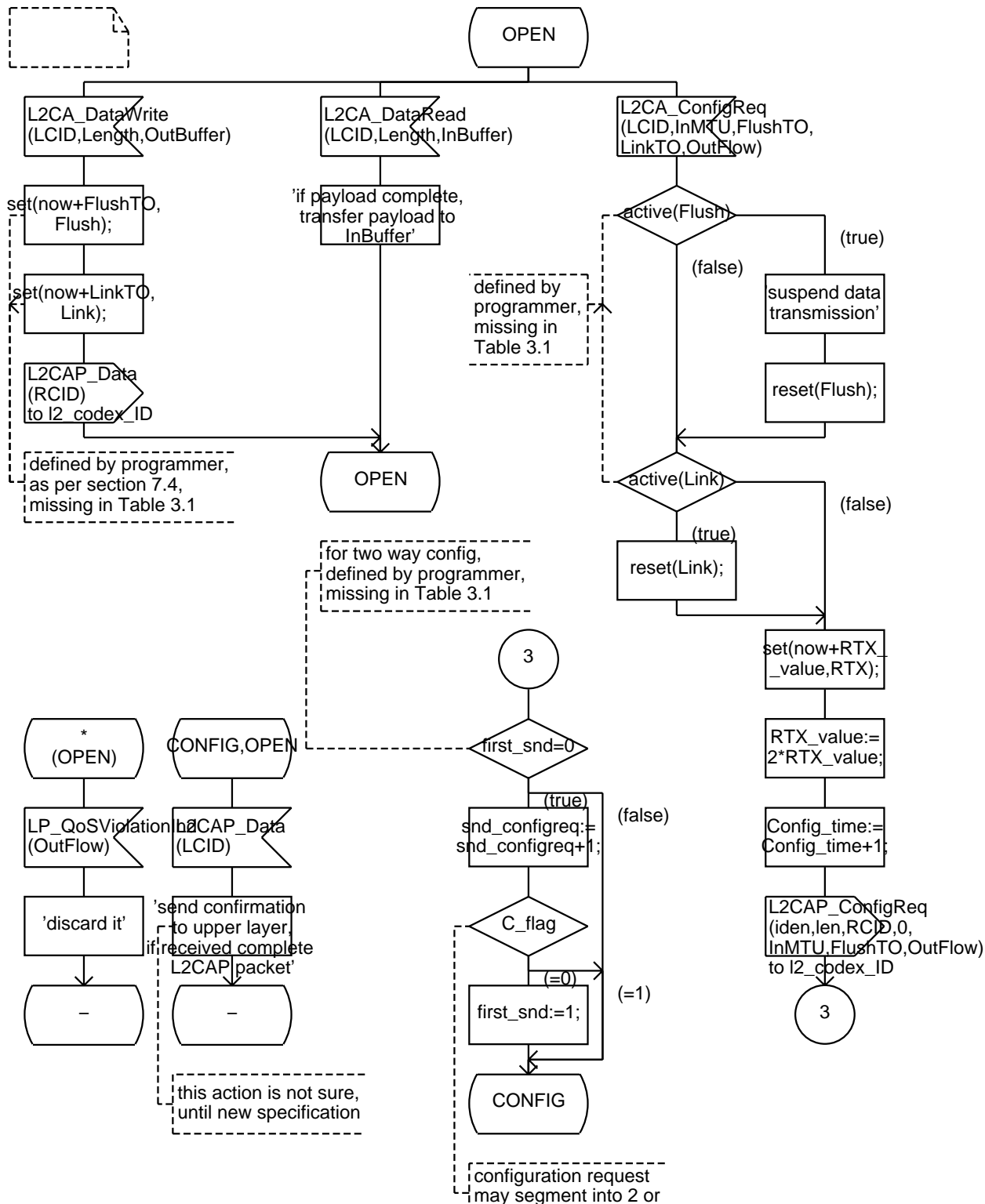




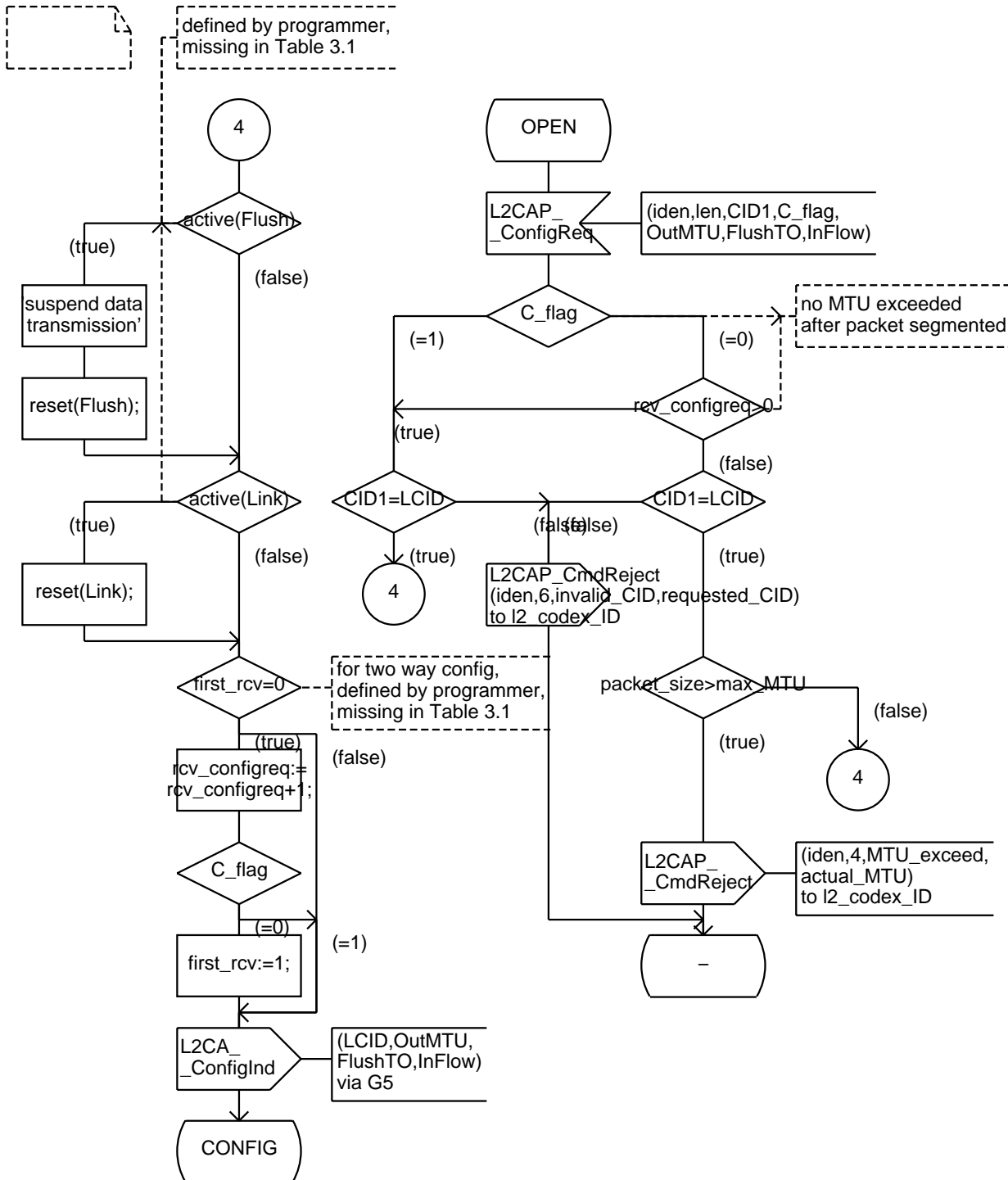


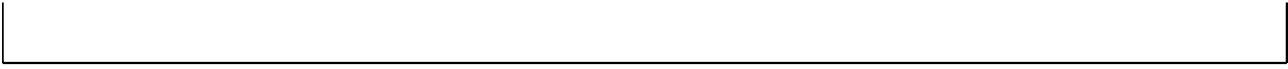


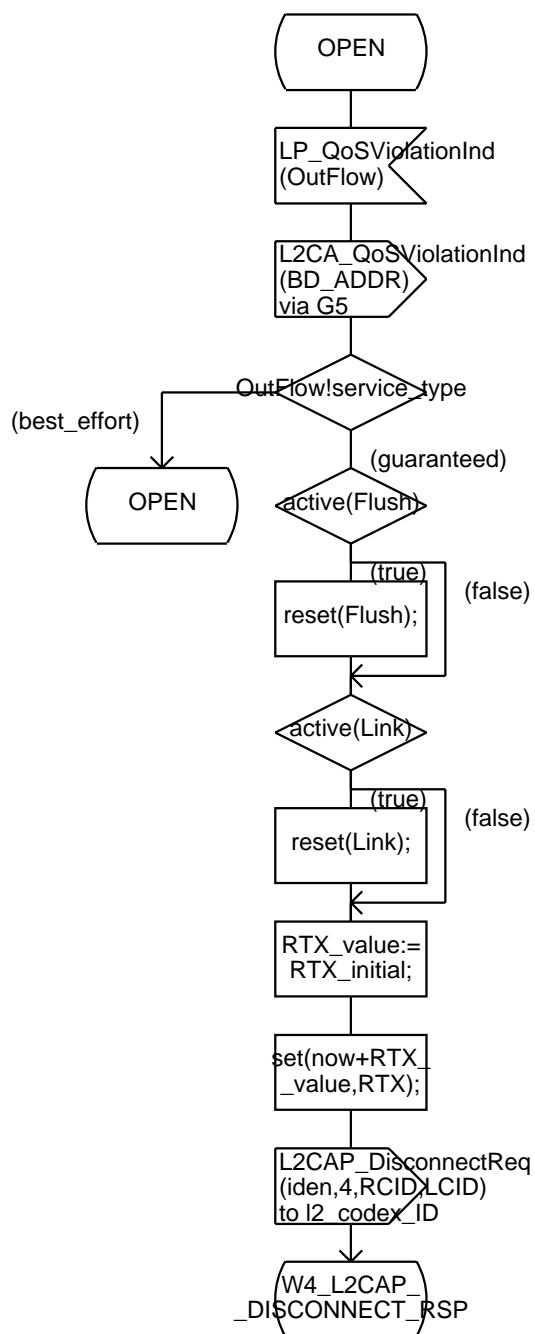


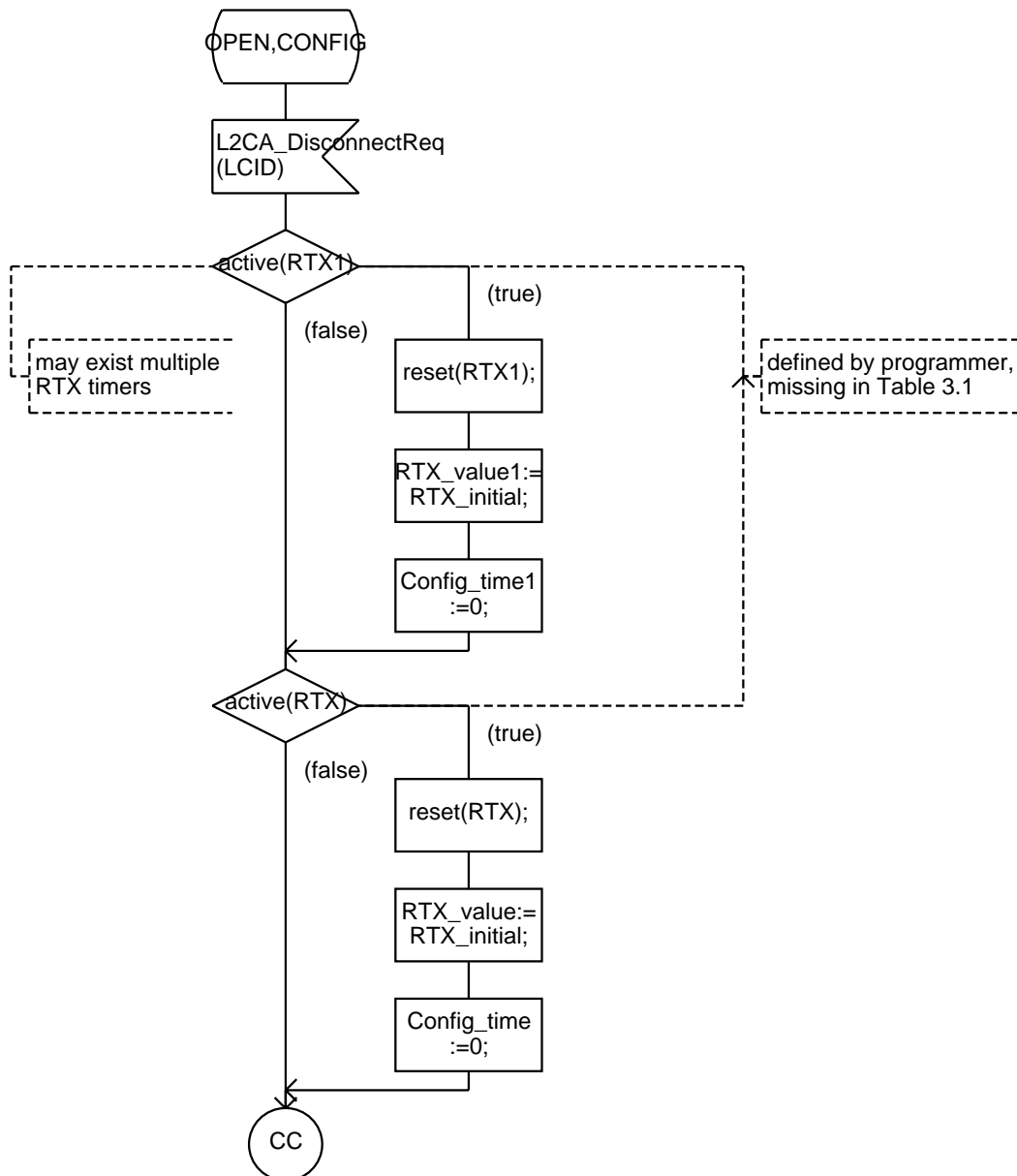


more packets

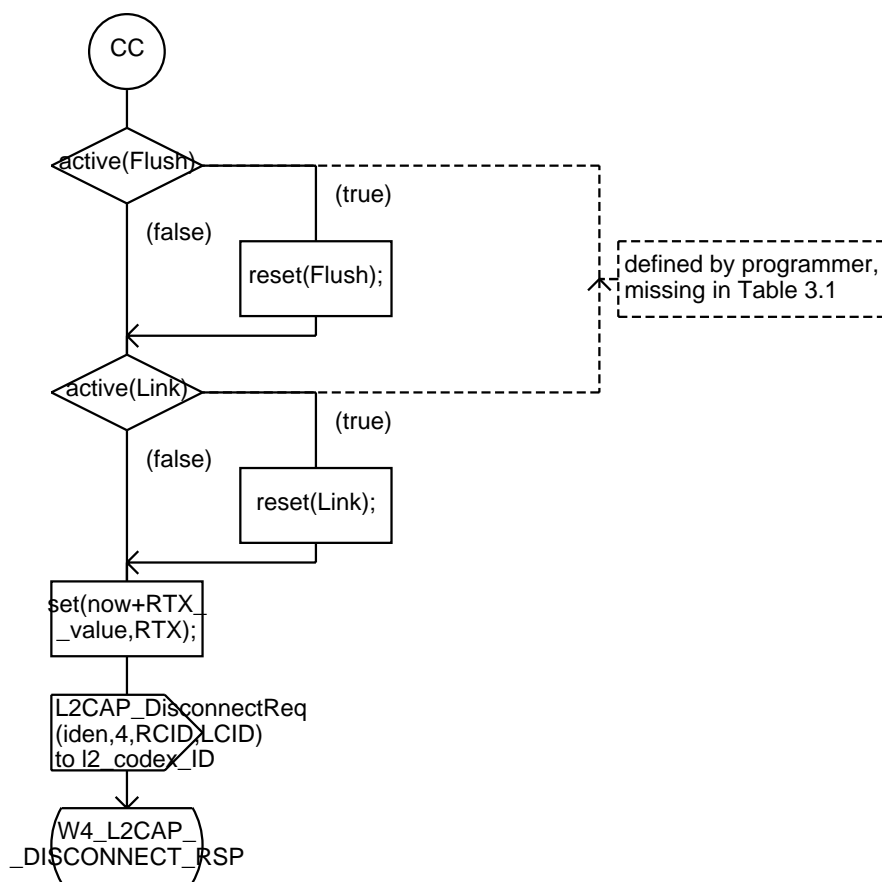




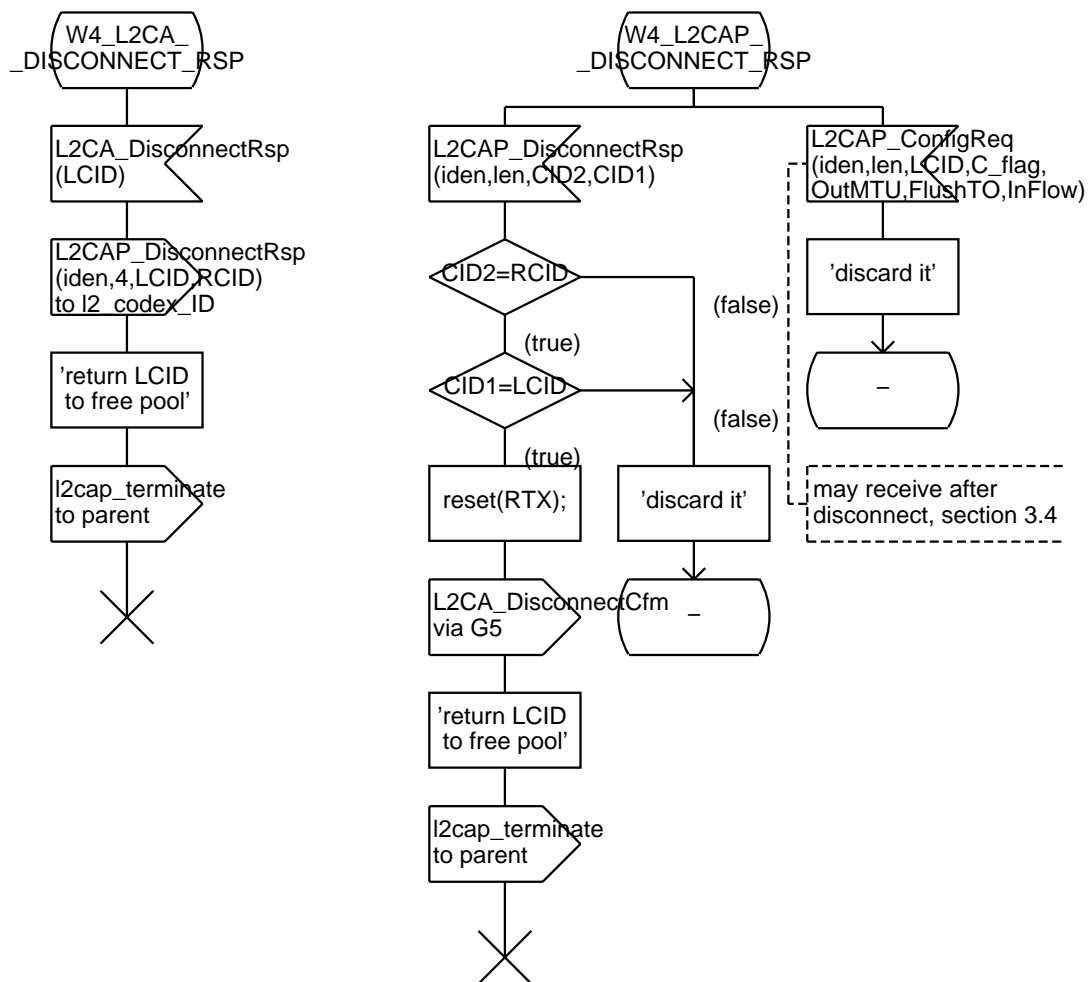


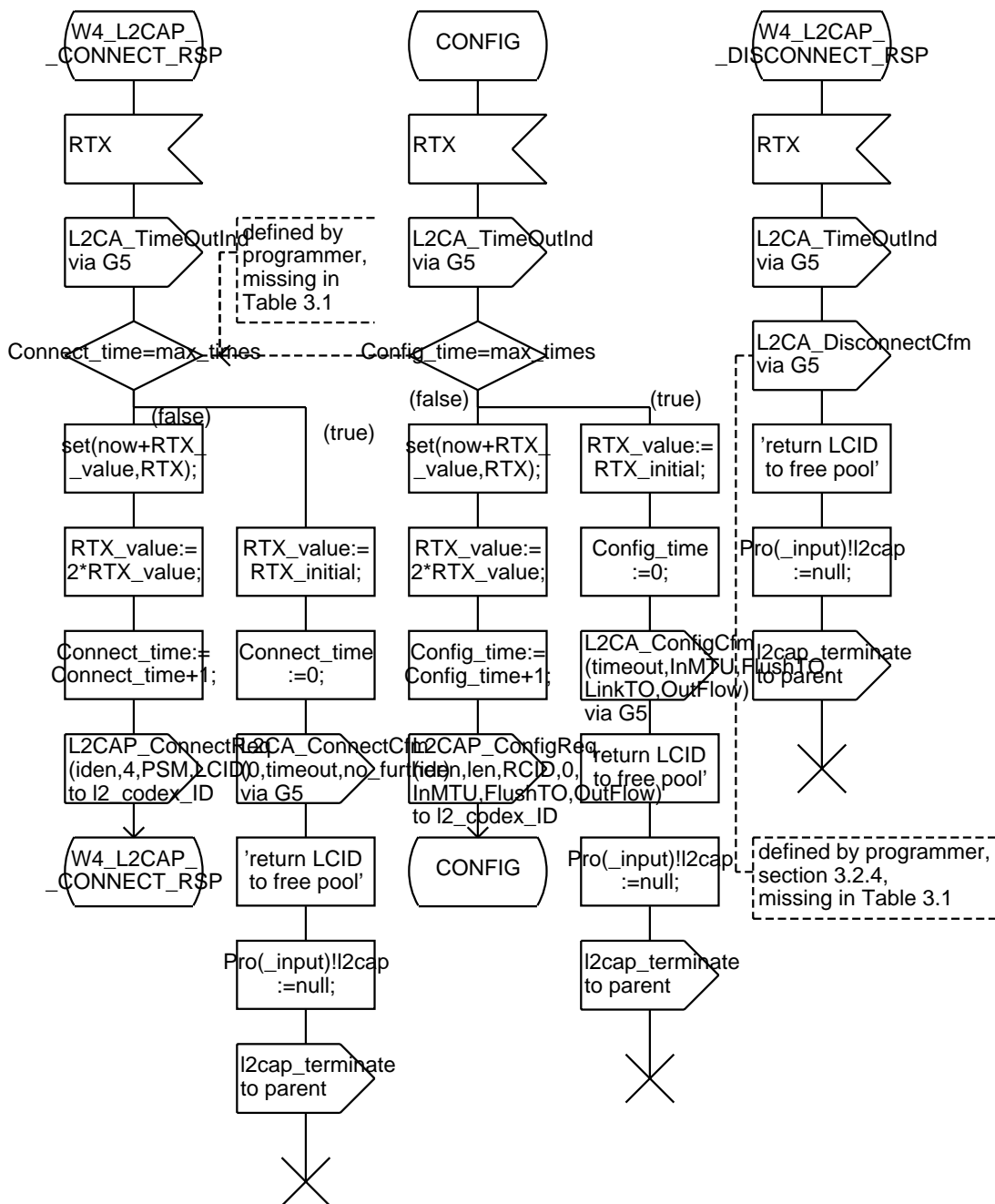




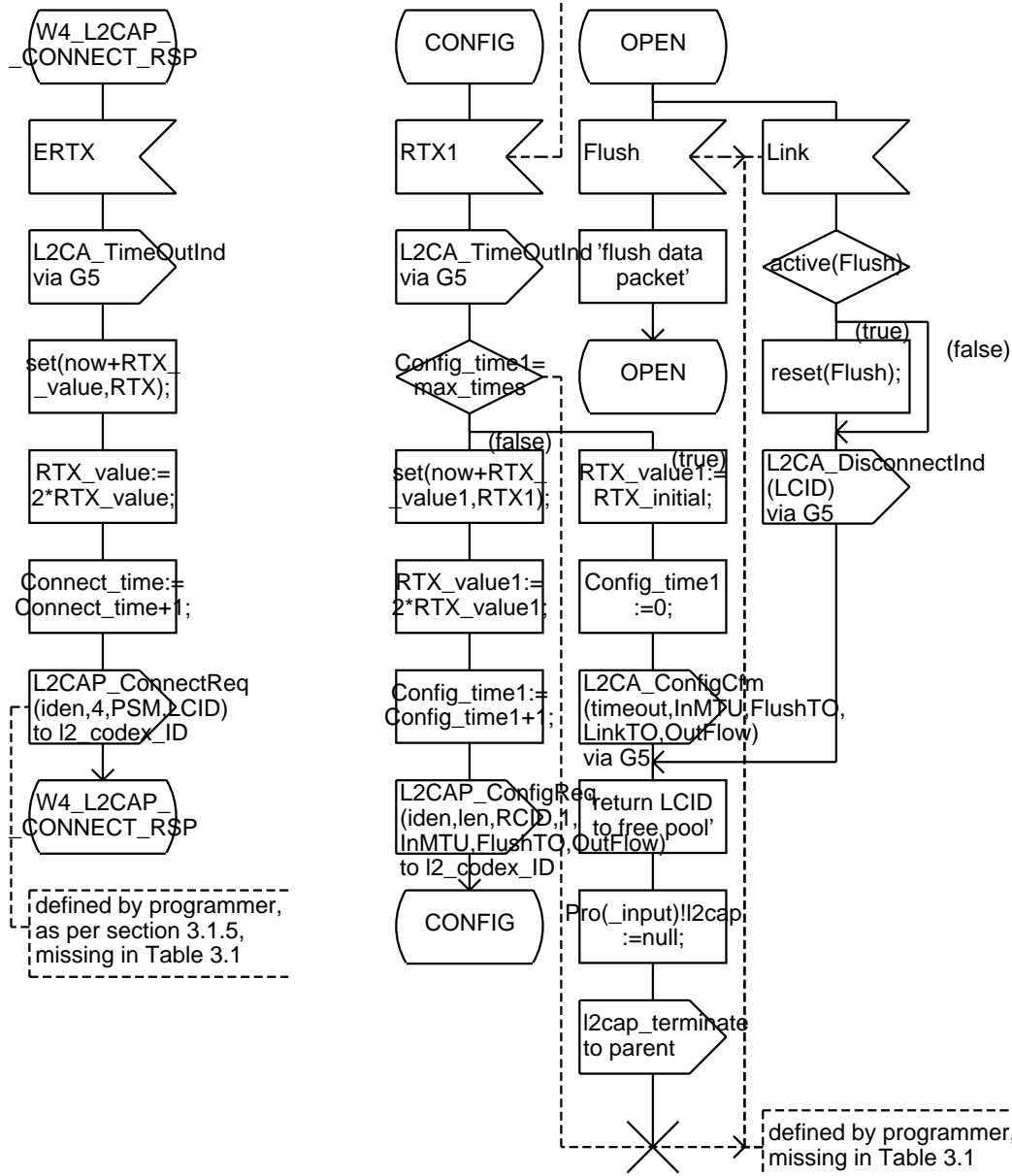


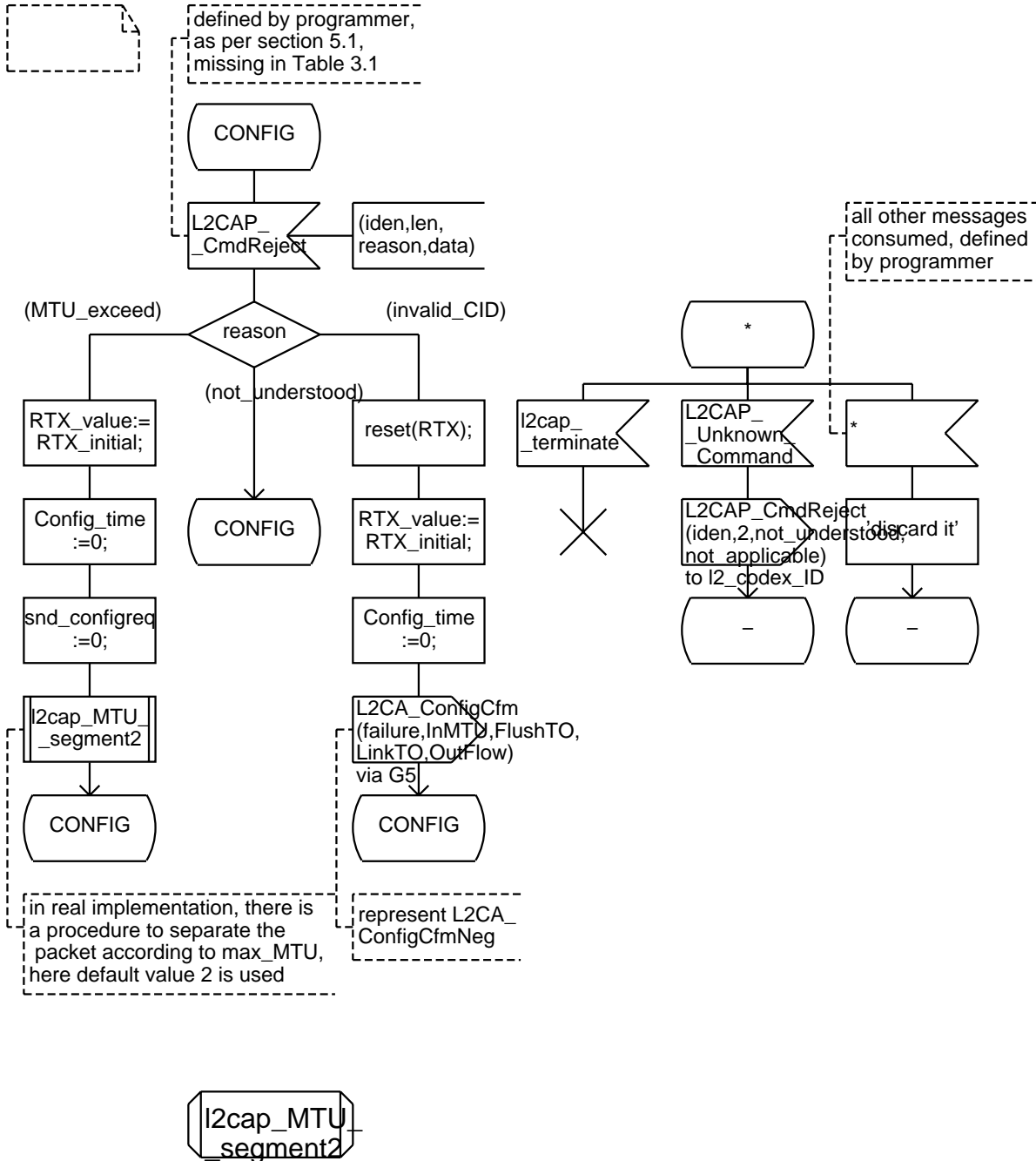


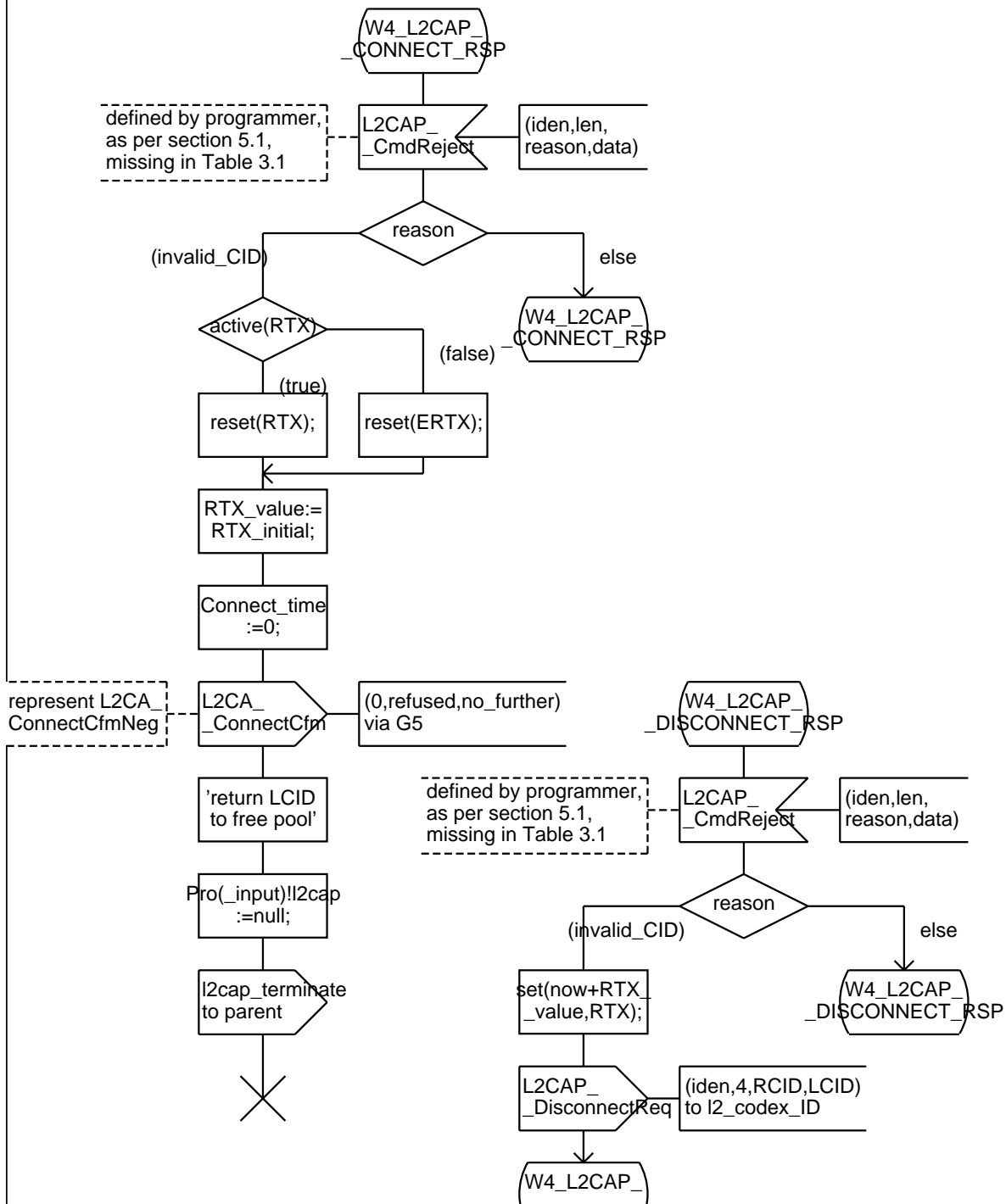




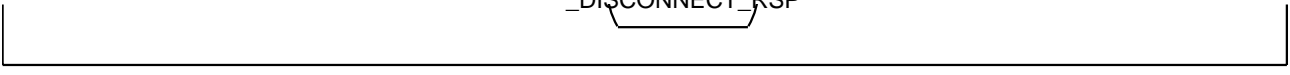


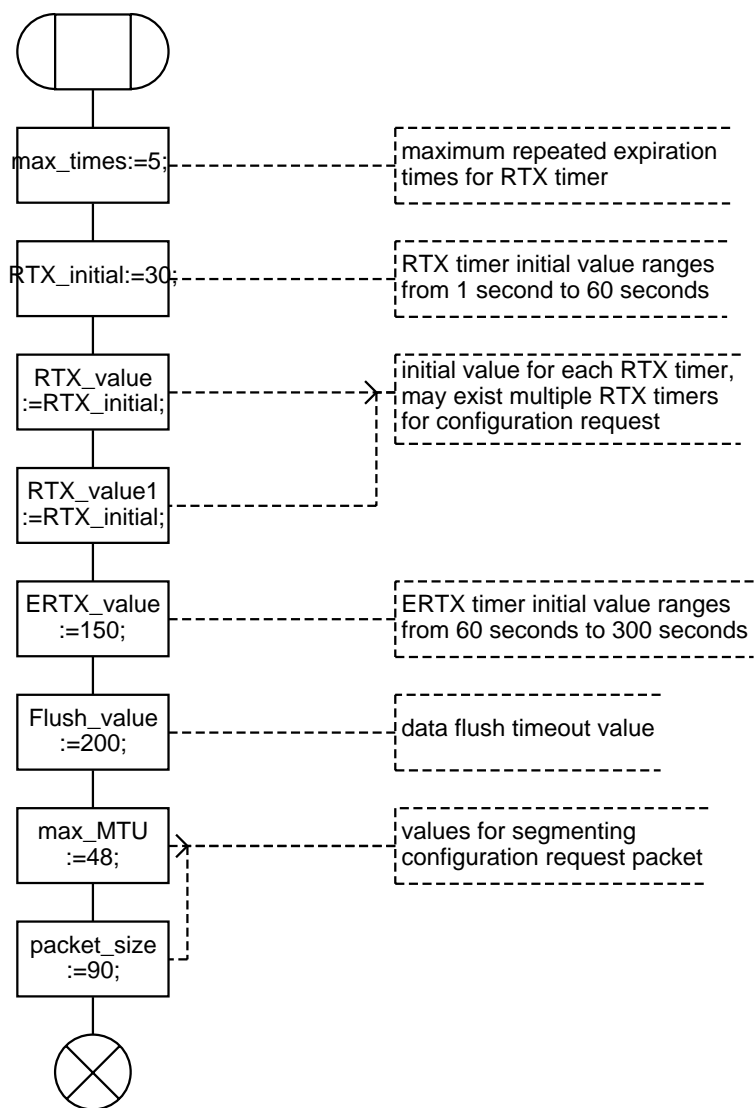






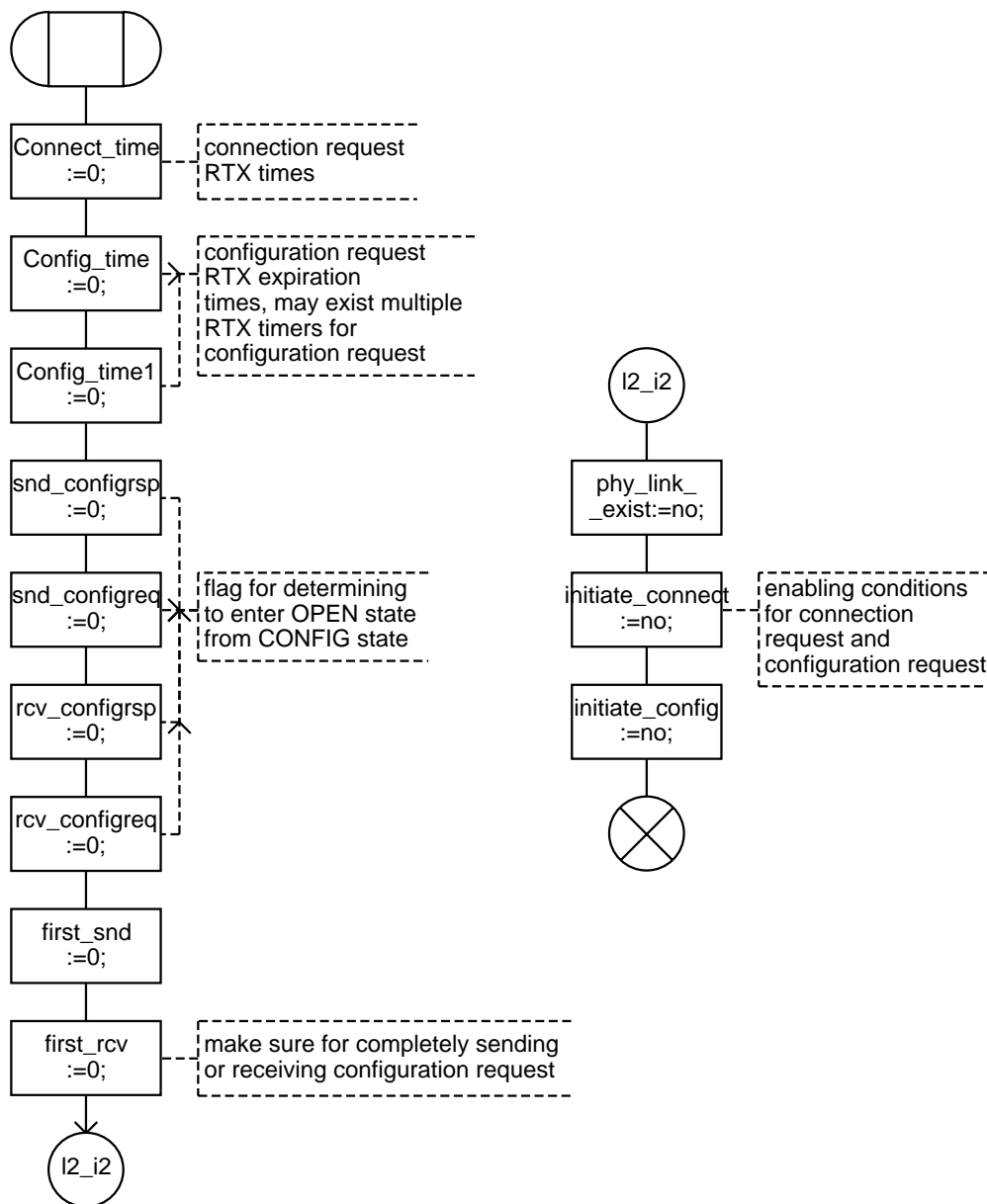
_DISCONNECT_RSP





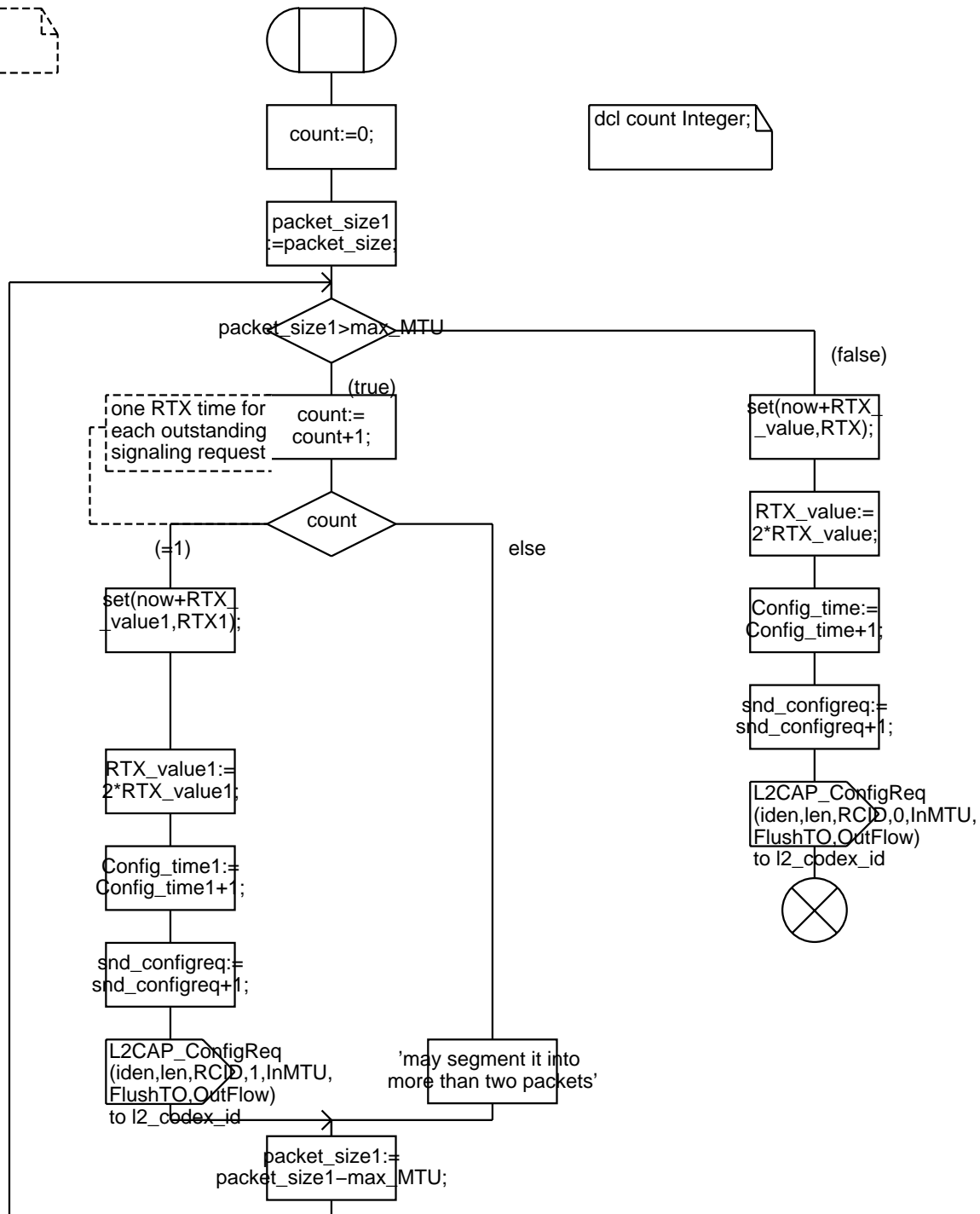
procedure l2cap_initialization2

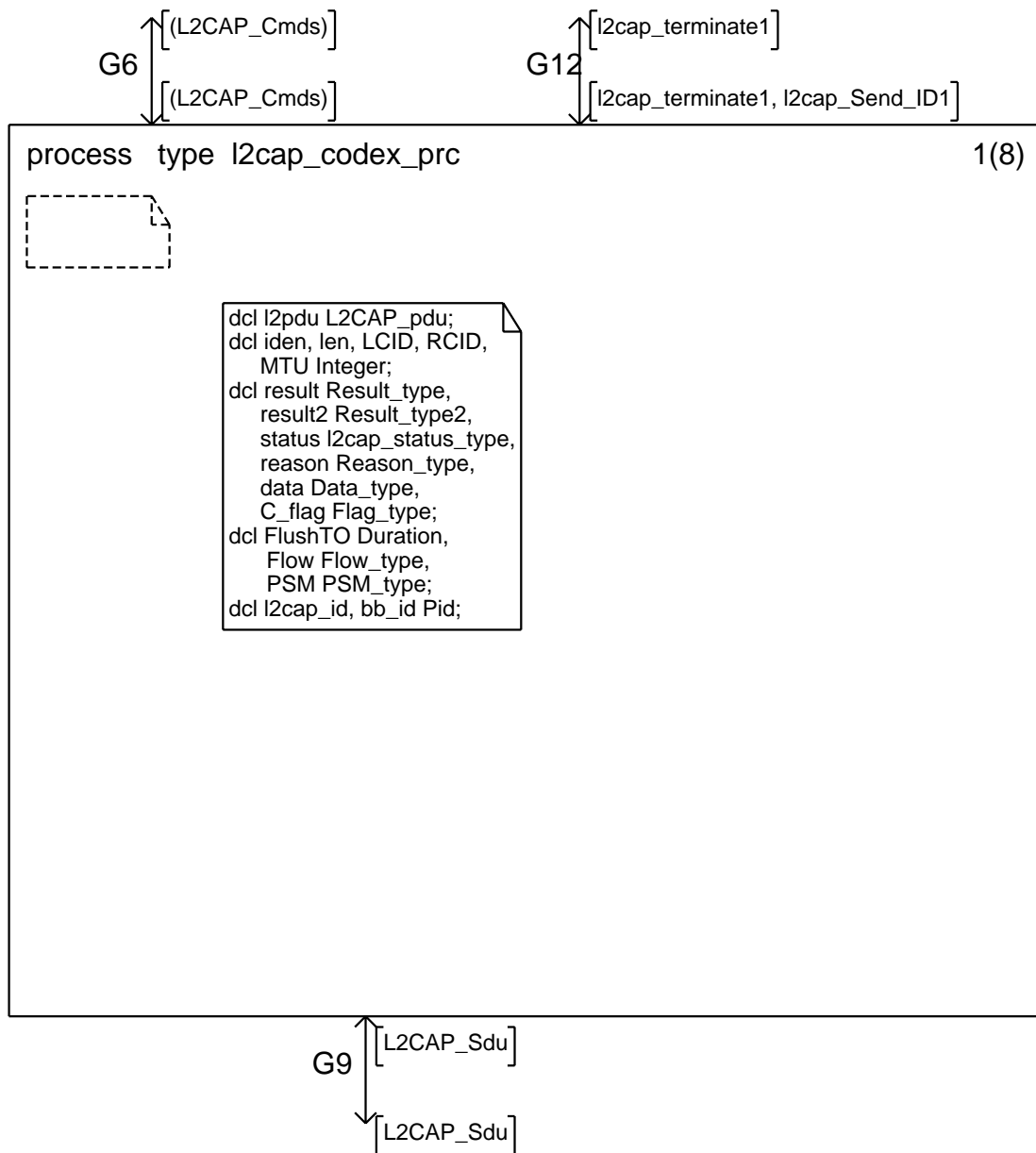
1(1)

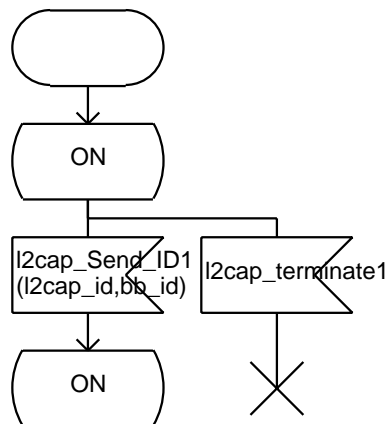


procedure l2cap_MTU_segment2

1(1)







These signals are strictly
for SDL process controls
(i.e. Not part of Protocol)

