- wand
  - Constant_declarations
    - PR Declaration
  - general type_declarations
    - PR Declaration
  - type_declarations
    - PR Declaration
  - General PR declaration
    - PR Declaration
    - PR Declaration
  - PR declarations AP
    - PR Declaration
    - PR Declaration
    - PR Declaration
  - PR Declaration MT
    - PR Declaration
    - PR Declaration
  - DefaultPartition
    - PR Declaration
    - PR Declaration
    - PR Declaration
    - PR Declaration
    - MT_Steady_State_Ctrl
      - PR Declaration
      - PR Declaration
      - MSG
        - PR Declaration
      - MBA
        - PR Declaration
      - MTI
        - DefaultPartition
          - PR Declaration
          - PR Declaration
        - wait_AP_answer
      - MTC
        - Initial_state
          - PR Declaration
        - Associated
        - Associated
        - wait_synchro_and_suspend
        - wait_AP_acquisition
        - wait_data

PR Declaration

wait_resume_and_old_AP

system wand

```
/* CONSTANTS */

/*SYNONYM Max_DLX_Buffer = 1000;*/  /* Size of DLX buffer */
/* SYNONYM TT_size = 1000; */

/*SYNONYM Windowsize =16;*/ /* See Martin's reflexion : no more than
                         15 cells can be sent without acknowledge */
SYNONYM MT_MAX_actif NATURAL = 5;       /* 256 (theoretical value) */
SYNONYM BROADCAST_MAC_ADDR NATURAL = 1;
/*SYNONYM CONTROL_MAC_ADDR NATURAL = 0;*/
/*SYNONYM NULL_MAC_ADDR NATURAL = 6;*/



/*SYNONYM CELL_DURATION = 1;*/ /* In real WAND spec : 21,6. 10^-6 s */

/* ----------- channels -------------------- */
SYNONYM VC_max = 10; /* Number max of Virtual Channels associated
                       to each MT,  Theoretical equal to 256. */
/*SYNONYM Control_MVC_Id MVC_id_type = 0;*/
/*SYNONYM M_control MVC_Id_type = 1;*/
/*SYNONYM UNI_Control MVC_Id_type = 2;*/
/*SYNONYM Broadcast_MVC_Id MVC_Id_type = VC_max-1;*/
/*SYNONYM UP Direction=0;*/
/*SYNONYM DOWN Direction = 1;*/
/*SYNONYM the_END Direction = -1;*/
SYNONYM MAX_AP_index = 5; /* Number max of AP infos stored in MTC */
```

system wand

```
/* general type declarations */
/* ------------------------- */

NEWTYPE return_type
  LITERALS Success,/* Connec_Refused, No_resp, Not_associated,
          Already_associated, Timedout,*/ Failed/*, Hunt_success,
        channel_exhausted*/;
ENDNEWTYPE return_type;

NEWTYPE susp_type
   LITERALS later, immediately;
ENDNEWTYPE;

NEWTYPE tip_type
 LITERALS start_tip, end_tip;
ENDNEWTYPE tip_type;

/*NEWTYPE MPDU_sig_type
   LITERALS MPDU_MT_AP_ASSOCIATION, MPDU_MT_AP_MAC_ADDR_RECEIVED,
  MPDU_MT_AP_DEASSOCIATION, MPDU_AP_MT_ASSOCIATION_ACK,
  MPDU_AP_MT_MAC_ADDR_ALLOCATE, MPDU_AP_MT_DEASSOCIATION,
  MPDU_MT_AP_CONN_ACTIVATE, MPDU_AP_MT_CONN_ACTIVE,
  MPDU_AP_MT_CONN_REFUSED, MPDU_AP_MT_CONN_PENDING,
  MPDU_TIP(MT_MAC_AddrType, tip_type), MPDU_TIP_ACK, MPDU_BEACON;
ENDNEWTYPE;*/

/*NEWTYPE TED_type
   LITERALS TX_SOF, TX_FrameHeader, TX_DOWNLINK_MPDU, TX_UPLINK_MPDU,
            RX_SOF, RX_FrameHeader, RX,   RX_OFF;
ENDNEWTYPE;*/

/* MT type declaration */
/* ------------------------------- */

 NEWTYPE HO_type
  LITERALS /*poweron_ho,*/ FORWARD_HO, BACKWARD_HO/*, no_HO*/;
 ENDNEWTYPE;

/* NEWTYPE ATM_ad_type
   LITERALS BROADCAST_ADDR;
 ENDNEWTYPE ATM_ad_type;*/

SYNTYPE MAC_ad_type = NATURAL
  DEFAULT BROADCAST_MAC_ADDR
  CONSTANTS MT_MAX_actif
ENDSYNTYPE MAC_ad_type;

/* NEWTYPE Assoc_type
    LITERALS Non_Associated, Associated;
 ENDNEWTYPE Assoc_type;*/

/* AP type declarations */

syntype AP_MAC_AddrType = integer  constants MAX_AP_index-1
endsyntype AP_MAC_AddrType;

/* Channel type declaration */
syntype Channel_id_type = integer  constants VC_max-1
endsyntype Channel_id_type;

SYNTYPE MT_MAC_AddrType=Integer CONSTANTS 0:MT_MAX_actif
EndSYNTYPE;
```

system wand

```
 /* --------------------- connections -------------------- */
/*SYNTYPE MVC_Id_type =Integer   CONSTANTS 0:VC_max-1
ENDSYNTYPE MVC_Id_type;*/

/*NEWTYPE service_type   LITERALS delay_sensitive, loss_sensitive;
ENDNEWTYPE service_type;*/

/*NEWTYPE ArrayMvcDlrPid ARRAY(MVC_id_type, Pid);
ENDNEWTYPE ArrayMvcDlrPid;*/

/*NEWTYPE MVC_table   ARRAY(MVC_Id_type,boolean);
ENDNEWTYPE MVC_table;*/

/*NEWTYPE MVC_nb STRUCT
  MVC_Id MVC_id_Type;
  nb Integer;
ENDNEWTYPE MVC_nb;*/

/*NEWTYPE MVC_nb_string   STRING(MVC_nb,Void);
ENDNEWTYPE MVC_nb_string;*/

/*NEWTYPE MPDU_type   LITERALS data_type, FH_type, SOF_type;
ENDNEWTYPE MPDU_type;*/

/* ------------ Traffic Table --------------------- */

/*SYNTYPE TT_id_type = Integer CONSTANTS 0:Max_DLX_buffer
ENDSYNTYPE TT_id_type;*/

/*NEWTYPE ATM_cell STRUCT
  Cell_Number Integer;
  SN Integer;
  RN Integer;
  MVC_Id MVC_id_type;
  CRC boolean;
ENDNEWTYPE ATM_cell;*/

/*NEWTYPE ArrayATMcells  ARRAY(TT_id_type,ATM_cell);
ENDNEWTYPE ArrayATMcells;*/

/*NEWTYPE Cell_list  STRING(ATM_cell,Void);
ENDNEWTYPE Cell_list;*/

/*NEWTYPE Traffic_Table  ARRAY(MVC_Id_type,ArrayATMcells);
ENDNEWTYPE Traffic_Table;*/

  /* ------------- Slot Map ----------------- */

/*SYNTYPE Direction = Integer  CONSTANTS -1:1
ENDSYNTYPE Direction;*/

/*NEWTYPE Time_slot_entry STRUCT
 slot TIME;           + MVC_id, MAC_ad, length, offset
 dir Direction;
ENDNEWTYPE Time_slot_entry;*/

/*NEWTYPE TSlot_string             type of the slot map
   STRING(Time_slot_entry, Void);
ENDNEWTYPE TSlot_string;*/
```

system wand

```
/* Upper layer signals */
/* ------------------- */
/* SIGNAL RELEASEAAL0_req,
  OPENAAL5_req, CLOSEAAL5_req, XMITAAL5_req, RELEASEAAL5_req,
  RECVAAL5_ind(signal_type),*/
/*SIGNAL ALLOCATEAAL0_req, ALLOCATEAAL0_cnf,
      XMITAAL0_req(Integer), RECVAAL0_ind(Integer),
      OPENAAL0_req, CLOSEAAL0_req;*/

/* Upper layer signallists */
/* ----------------------- */
/* SIGNALLIST Aatm2MDLX = RECVAAL0_ind, ALLOCATEAAL0_cnf;*/
/* SIGNALLIST MDLX2Aatm = RELEASEAAL0_req, ALLOCATEAAL0_req;*/
/* SIGNALLIST MLCP2Aatm = OPENAAL5_req, CLOSEAAL5_req, XMITAAL5_req,
                     RELEASEAAL5_req; */
/*SIGNALLIST MDLR2Aatm = XMITAAL0_req;*/
/*SIGNALLIST MGDL2Aatm = OPENAAL0_req, CLOSEAAL0_req;*/
```

```
/* PEER TO PEER CONTROL SIGNALS (between AP and MT)  */
/* ------------------------------------------------- */

SIGNAL /* MPDU_MT_AP_ASSOCIATION, MPDU_MT_AP_MAC_ADDR_RECEIVED,
 MPDU_MT_AP_DEASSOCIATION(MAC_ad_type, return_type),*/
/* MPDU_AP_MT_ASSOCIATION_ACK(return_type), */
/* MPDU_AP_MT_MAC_ADDR_ALLOCATE(ATM_ad_type),*/
/* MPDU_AP_MT_DEASSOCIATION(MAC_ad_type, return_type), */
/* MPDU_MT_AP_CONN_ACTIVATE, MPDU_AP_MT_CONN_ACTIVE,
 MPDU_AP_MT_CONN_REFUSED, MPDU_AP_MT_CONN_PENDING, */
 MPDU_TIP(MAC_ad_type,tip_type), MPDU_TIP_ACK(MAC_ad_type),  MPDU_BEACON;

/*SIGNAL RX_FH;*/

/* PEER TO PEER CONTROL SIGNALS (between AP and MT) */
/* ------------------------------------------------ */

/*SIGNALLIST MMAA2MAAA_INITIAL_MPDU = MPDU_MT_AP_ASSOCIATION;*/
/*SIGNALLIST MMAA2MAAA_MPDU = MPDU_MT_AP_MAC_ADDR_RECEIVED,
                           MPDU_MT_AP_DEASSOCIATION;*/
/*SIGNALLIST MAAA2MMAA_MPDU = MPDU_AP_MT_ASSOCIATION_ACK,
                           MPDU_AP_MT_MAC_ADDR_ALLOCATE,
                           MPDU_AP_MT_DEASSOCIATION;*/
/*SIGNALLIST MMMA2MAMA_MPDU = MPDU_MT_AP_CONN_ACTIVATE;*/
/*SIGNALLIST MAMA2MMMA_MPDU = MPDU_AP_MT_CONN_ACTIVE, MPDU_AP_MT_CONN_REFUSED,
                           MPDU_AP_MT_CONN_PENDING;*/
SIGNALLIST MMTI2MATI_MPDU = MPDU_TIP;
SIGNALLIST MATI2MMTI_MPDU = MPDU_TIP_ACK;
SIGNALLIST MABA2MMBA_MPDU = MPDU_BEACON;


/*SIGNAL From_MT_to_AP(MAC_ad_type,MPDU_type,Cell_list, MVC_Id_type, Integer,Integer),
      From_AP_to_MT(MPDU_type,TSlot_string,Cell_list,MVC_Id_type,Integer);*/
/*SIGNAL PID_REGISTERED(MAC_ad_type, MVC_Id_type, Pid),
      PID_DEREGISTERED(MAC_ad_type,MVC_Id_type,Pid);*/
/*SIGNAL SERVICE_PARAMETER(service_type),
      SET_TRAFFIC_PARM(MAC_ad_type,MVC_Id_type, service_type),
      DESTROY(MVC_Id_type);*/
```

system wand

```
/* AP MAC Layer signals */
/* -------------------- */

/* Peer to peer : Mascara Control to Signalling */

/*SIGNAL ALARM_ind, GET_INFO_cnf, RR_STATUS_cnf, ASSOCIATION_ind,
 AP_DEASSOCIATION_ind, AP_VC_RELEASE_cnf(return_type),
 AP_DEASSOCIATION_cnf(return_type),
 AP_ASSOCIATION_LOST_ind, VC_RESERVED_cnf(return_type),
 VC_SWITCHED_cnf(return_type),
 CLOSE_MAC_cnf, OPEN_AP_MAC_cnf(return_type), FRAME_START;*/

/* Peer to peer : Signalling to Mascara Control */

/*SIGNAL AP_VC_RELEASE_req, AP_DEASSOCIATION_req, VC_RESERVED_req,
VC_SWITCHED_req, GET_INFO_req, RR_STATUS_req, CLOSE_MAC_req,
OPEN_AP_MAC_req;*/
```

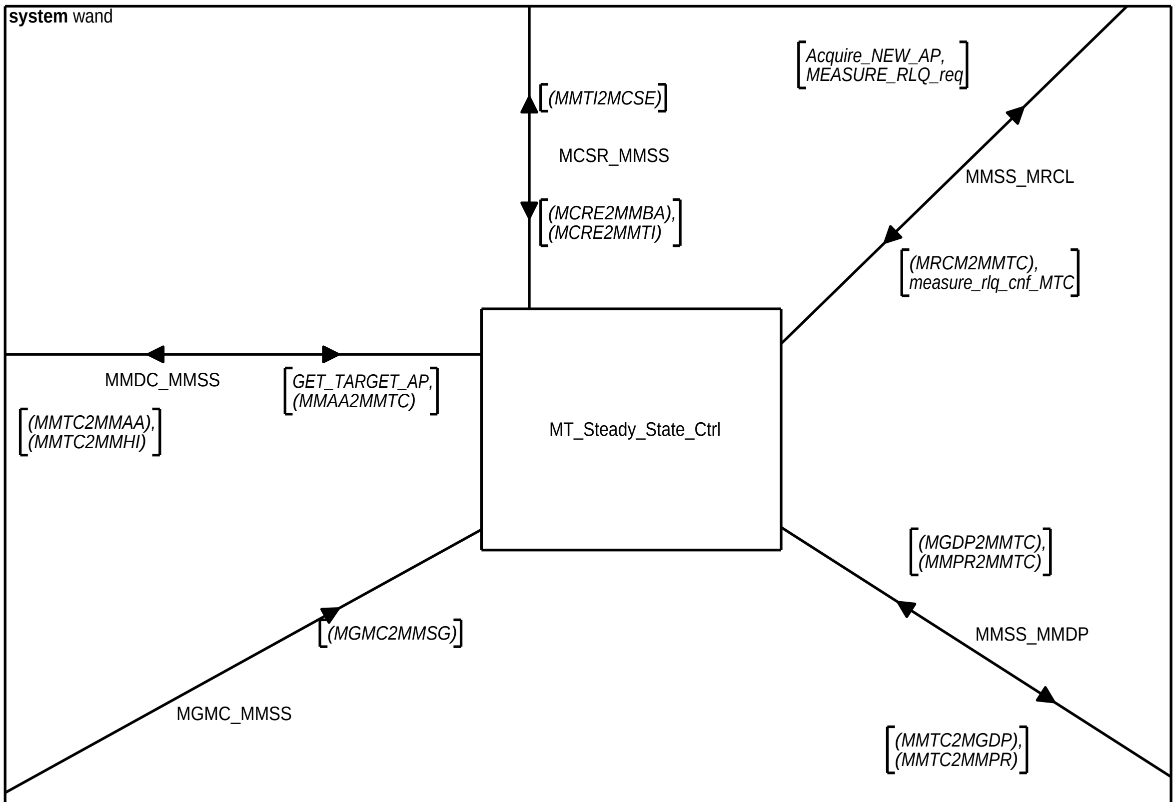
```
/* AP MAC Layer signallists */
/* ------------------------ */

/* AP Peer to peer : Mascara Control to Signalling */
/* ----------------------------------------------- */
/*SIGNALLIST MALS2Ccon = ALARM_ind, GET_INFO_cnf, RR_STATUS_cnf;*/
/*SIGNALLIST MAAA2Ccon = ASSOCIATION_ind, AP_DEASSOCIATION_ind,
                      AP_VC_RELEASE_cnf;*/
/*SIGNALLIST MADG2Ccon = AP_DEASSOCIATION_cnf, AP_ASSOCIATION_LOST_ind,
                      AP_VC_RELEASE_cnf;*/
/*SIGNALLIST MAMA2Ccon = VC_RESERVED_cnf, VC_SWITCHED_cnf;*/
/*SIGNALLIST MGMCA2Ccon = CLOSE_MAC_cnf, OPEN_AP_MAC_cnf;*/

/* AP Peer to peer : Signalling to Mascara Control */
/* ----------------------------------------------- */
/*SIGNALLIST Ccon2MAAA = AP_VC_RELEASE_req;*/
/*SIGNALLIST Ccon2MADG = AP_DEASSOCIATION_req;*/
/*SIGNALLIST Ccon2MAMA = VC_RESERVED_req, VC_SWITCHED_req;*/
/*SIGNALLIST Ccon2MALS = GET_INFO_req, RR_STATUS_req;*/
/*SIGNALLIST Ccon2MGMCA = CLOSE_MAC_req, OPEN_AP_MAC_req;*/
```

/* The 2 signallists Ccon2MGMC have been renamed by Ccon2MGMCA and
Ccon2MGMCM – Idem for MGMC2Ccon */

system wand

```
/* MT MAC Layer signal declaration */
/* ------------------------------- */

/* Peer to peer : Mascara Control to Signalling */
/* -------------------------------------------- */
/*SIGNAL ASSOCIATION_cnf(return_type), MT_DEASSOCIATION_ind,
VC_REOPEN_cnf(return_type), VC_OPEN_cnf(return_type),
MT_VC_RELEASE_cnf(return_type), MT_DEASSOCIATION_cnf(return_type),
HO_ind, MT_ASSOCIATION_LOST_ind, OPEN_MT_MAC_cnf(return_type);*/

/* Peer to peer : Signalling to Mascara Control */
/* -------------------------------------------- */
/*SIGNAL VC_REOPEN_req, ASSOCIATION_req,
              MT_DEASSOCIATION_req(HO_type), OPEN_MT_MAC_req,
              MT_VC_RELEASE_req, VC_OPEN_req;*/  /* CLOSE_MAC_req ? */


/* Output of MT_Deassociation_ind ??? */
```

```
/* MT MAC Layer signallist declaration */
/* ----------------------------------- */

/* Peer to peer : Mascara Control to Signalling */
/* -------------------------------------------- */
/*SIGNALLIST MMAA2Ccon = ASSOCIATION_cnf, MT_DEASSOCIATION_ind,
             VC_REOPEN_cnf, VC_OPEN_cnf, MT_VC_RELEASE_cnf,
             MT_DEASSOCIATION_cnf;*/
/*SIGNALLIST MMHI2Ccon = HO_ind, MT_ASSOCIATION_LOST_ind;*/
/*SIGNALLIST MGMCM2Ccon = CLOSE_MAC_cnf, OPEN_MT_MAC_cnf;*/
/*SIGNALLIST MMMA2Ccon = VC_OPEN_cnf, VC_REOPEN_cnf; */ /* I have added
                 VC_REOPEN_cnf to this list. To be verified ! */

/* Peer to peer : Signalling to Mascara Control */
/* -------------------------------------------- */
/*SIGNALLIST Ccon2MMAA = VC_REOPEN_req, ASSOCIATION_req,
              MT_DEASSOCIATION_req;*/
/*SIGNALLIST Ccon2MGMCM = CLOSE_MAC_req, OPEN_MT_MAC_req;*/
/*SIGNALLIST Ccon2MMMA = MT_VC_RELEASE_req, VC_OPEN_req;*/

/* Qu : the signals VC_OPEN_cnf and VC_REOPEN_cnf are sent
from MAA or MMA to Ccon ? */
```



/* The channel MT_AP replaces the physical layer of both AP and MT, as well as the wireless link between them. */

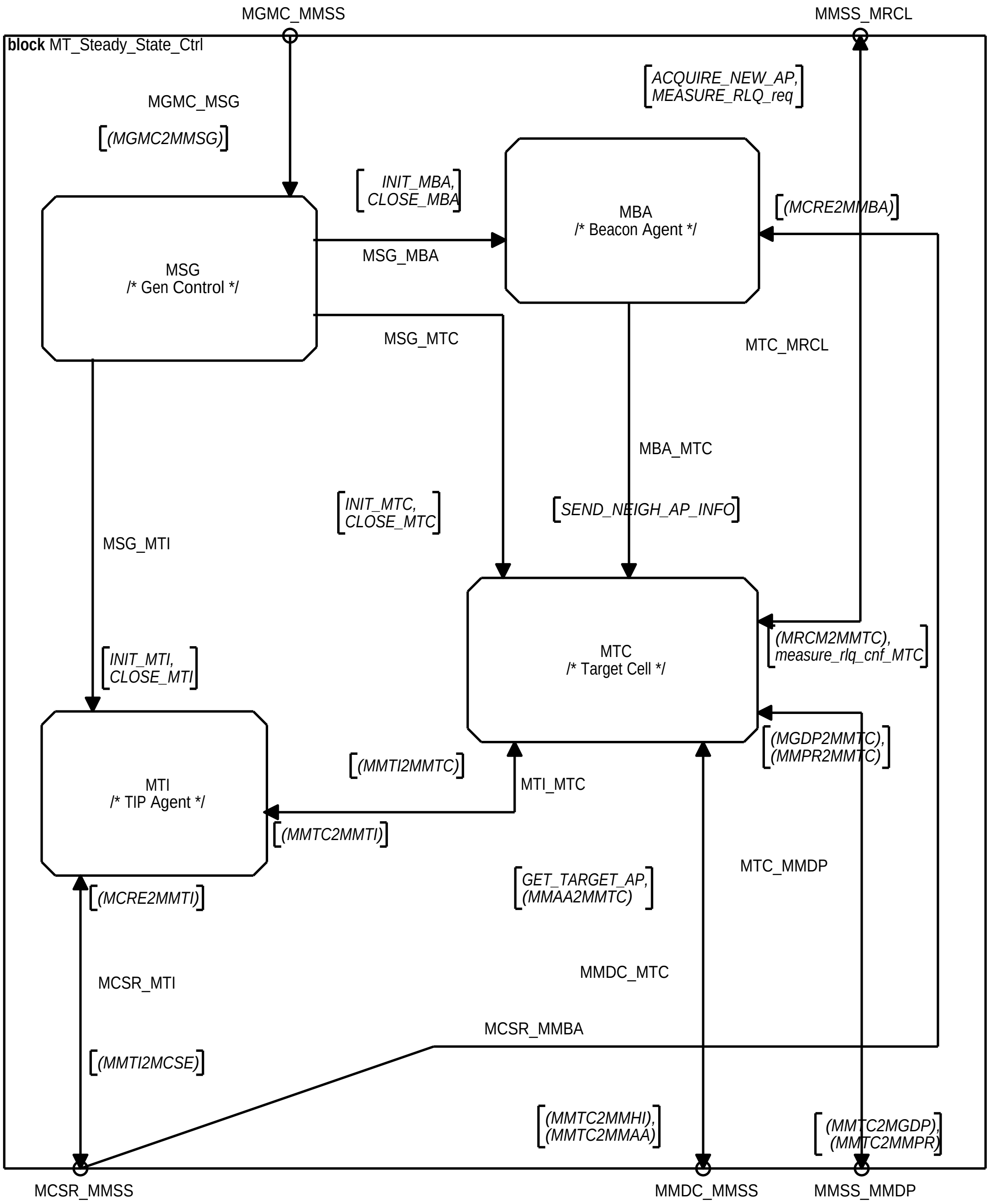
/* The environment is composed of :
- the upper layers of the MT (accessed through channel user)
- the upper layers of AP and the wired ATM network (accessed through channel AP_to_CS_or_ATM) */

/* We will probably need to specify another MT, representing the broadcast MT with MT_MAC_Addr=BROADCAST_MT_MAC_ADDR and MVC_Id = BROADCAST_MVC_Id_ADDR. All broadcast signalling is done in the AP through the CSR/WDLC of that virtual connection*/

```
/* Control - Physical layer */
/*SIGNAL PHY_MEASURE_RSSI, PHY_MEASURED_RSSI, PHY_SELECT_CHANNEL;*/

/*Control - Data Pump */
/*SIGNAL SET_MAC_ADDR(MAC_ad_type), SET_MAC_ADDR_ACK, MPDU_ERRORS_cnf(Integer);*/

/* Control - CSR */

/* CSR - WDLC */
/*SIGNAL XMIT_CTRL_CELL(MPDU_sig_type),
INSTANCIATE_DLC(MAC_ad_type,MVC_Id_type,service_type),
CTRL_CELL_RCVD(MPDU_sig_type);*/

/*WDLC - Mac Data Pump */
/*SIGNAL TRAFFIC_UPDATED(Integer,MVC_Id_type,Cell_list,Integer), GEN_WDLC_FB;*/

/* DataPump - WDLC */
/*SIGNAL MPDU_BODY_RCVD(Cell_List, Integer);*/

SIGNAL /*CELL_ERRORS_req, CELL_ERRORS_cnf(Integer),*/ AP_LOST;
```

```
/* MT MAC Layer */

SIGNAL NEIGH_AP_LOAD_INFO_req, NEIGH_AP_LOAD_INFO_cnf,
RESUME_MSC, RESUME_MSC_OK, SUSPEND_MSC(susp_type), SUSPEND_MSC_OK/*,
 INIT_MDP_DONE, CLOSE_MDP_DONE, MPDU_ERRORS_req,
 TRAFFIC_RELEASED, INIT_MD, CLOSE_MDP, FH_HUNT_req,
FH_HUNT_cnf(return_type)*/;

/* ---------------------------------------------------------- */
/* ICC - Control */
/*  DELETED FROM THIS MODEL ---> abstraction of ICC !!!
SIGNALLIST MMCN2MMAA = (Ccon2MMAA), (Ccon2MMMA);
SIGNALLIST MMCN2MGMC = (Ccon2MGMCM);
SIGNALLIST MMHI2MMCN = (MMHI2Ccon);
SIGNALLIST MMAA2MMCN = (MMAA2Ccon);
SIGNALLIST MMMA2MMCN = (MMMA2Ccon);
SIGNALLIST MGMC2MMCN = (MGMCM2Ccon); */

/* Control - DataPump */
/*SIGNALLIST MGMC2MGDP = INIT_MDP, CLOSE_MDP, SET_TRAFFIC_PARM,
                      TRAFFIC_RELEASED;*/
/*SIGNALLIST MMMA2MGDP = SET_TRAFFIC_PARM, TRAFFIC_RELEASED;*/
/*SIGNALLIST MMAA2MGDP = SET_MAC_ADDR;*/
SIGNALLIST MMTC2MMPR = NEIGH_AP_LOAD_INFO_req;
/*SIGNALLIST MMHI2MMPR = MPDU_ERRORS_req;*/
/*SIGNALLIST MGDP2MGMC = INIT_MDP_DONE, CLOSE_MDP_DONE;*/
SIGNALLIST MMTC2MGDP = SUSPEND_MSC, RESUME_MSC;
/*SIGNALLIST MRCM2MMPX = FH_HUNT_req;*/
SIGNALLIST MGDP2MMTC = RESUME_MSC_OK, SUSPEND_MSC_OK;
/*SIGNALLIST MGDP2MMAA = SET_MAC_ADDR_ACK;*/
SIGNALLIST MMPR2MMTC = NEIGH_AP_LOAD_INFO_cnf;
/*SIGNALLIST MMPX2MRCM = FH_HUNT_cnf;*/
/*SIGNALLIST MMPX2MMHI = AP_LOST;*/

/* Control - WDLC */
/*SIGNALLIST MMMA2MGDL = INSTANCIATE_DLC, DESTROY;*/
/*SIGNALLIST MMHI2MGDL = CELL_ERRORS_req;*/
/*SIGNALLIST MGDL2MMMA = PID_REGISTERED;*/

/* Control - CSR */
/*SIGNALLIST MMAA2MCSE = (MMAA2MAAA_MPDU), (MMAA2MAAA_INITIAL_MPDU);*/
/*SIGNALLIST MMMA2MCSE = (MMMA2MAMA_MPDU);*/
SIGNALLIST MMTI2MCSE = (MMTI2MATI_MPDU);
/*SIGNALLIST MCRE2MMAA = (MAAA2MMAA_MPDU), (MAMA2MMMA_MPDU);*/
SIGNALLIST MCRE2MMTI = (MATI2MMTI_MPDU);
SIGNALLIST MCRE2MMBA = (MABA2MMBA_MPDU);

/* CSR - WDLC */
/*SIGNALLIST MCRE2MGDL = INSTANCIATE_DLC;*/
/*SIGNALLIST MGDL2MCRE = PID_REGISTERED;*/

/* WDLC - Data Pump */
/*SIGNALLIST MGDL2MMPR = PID_REGISTERED, PID_DEREGISTERED;*/
```

```
/* MT Control */

SIGNAL
MEASURE_RLQ_req, measure_rlq_cnf_MTC,
/*INIT_RCL, INIT_RCL_OK, INIT_RCL_KO(return_type),
CLOSE_RCL, CLOSE_RCL_DONE, */
INIT_MSS, CLOSE_MSS, GET_TARGET_AP(HO_type), TARGET_AP_FOUND(AP_MAC_AddrType),
SUSPEND_MHI, RESUME_MHI, SYNCHRO_WITH(AP_MAC_AddrType), ASSOCIATION_COMPLETED,
SYNCHRO_WITH_ACK(return_type),
ACQUIRE_NEW_AP(Channel_id_type),
ACQUIRE_NEW_AP_OK(AP_MAC_AddrType), ACQUIRE_NEW_AP_KO/*,
MT_DEASSOCIATE, INIT_MDC, MT_DEASSOCIATION_DONE*/;

/* MT Control signallists */
/* ---------------------- */

/* Generic  - Radio */
/*SIGNALLIST MGMC2MRCM = INIT_RCL,CLOSE_RCL;*/
/*SIGNALLIST MRCM2MGMC = INIT_RCL_OK, INIT_RCL_KO, CLOSE_RCL_DONE;
*/
/* Generic Control - Dynamic */
/*SIGNALLIST MGMC2MMDG = MT_DEASSOCIATE, INIT_MDC;*/

/* Generic - Steady State */
SIGNALLIST MGMC2MMSG = INIT_MSS, CLOSE_MSS;

/* Steady State - Dynamic */
SIGNALLIST MMTC2MMHI = TARGET_AP_FOUND, AP_LOST, SUSPEND_MHI,
                       RESUME_MHI;
SIGNALLIST MMHI2MMTC = GET_TARGET_AP;
SIGNALLIST MMTC2MMAA = SYNCHRO_WITH_ACK;
SIGNALLIST MMAA2MMTC = SYNCHRO_WITH, ASSOCIATION_COMPLETED;

/* Radio - Steady State */
SIGNALLIST MRCM2MMTC = ACQUIRE_NEW_AP_OK, ACQUIRE_NEW_AP_KO;
```

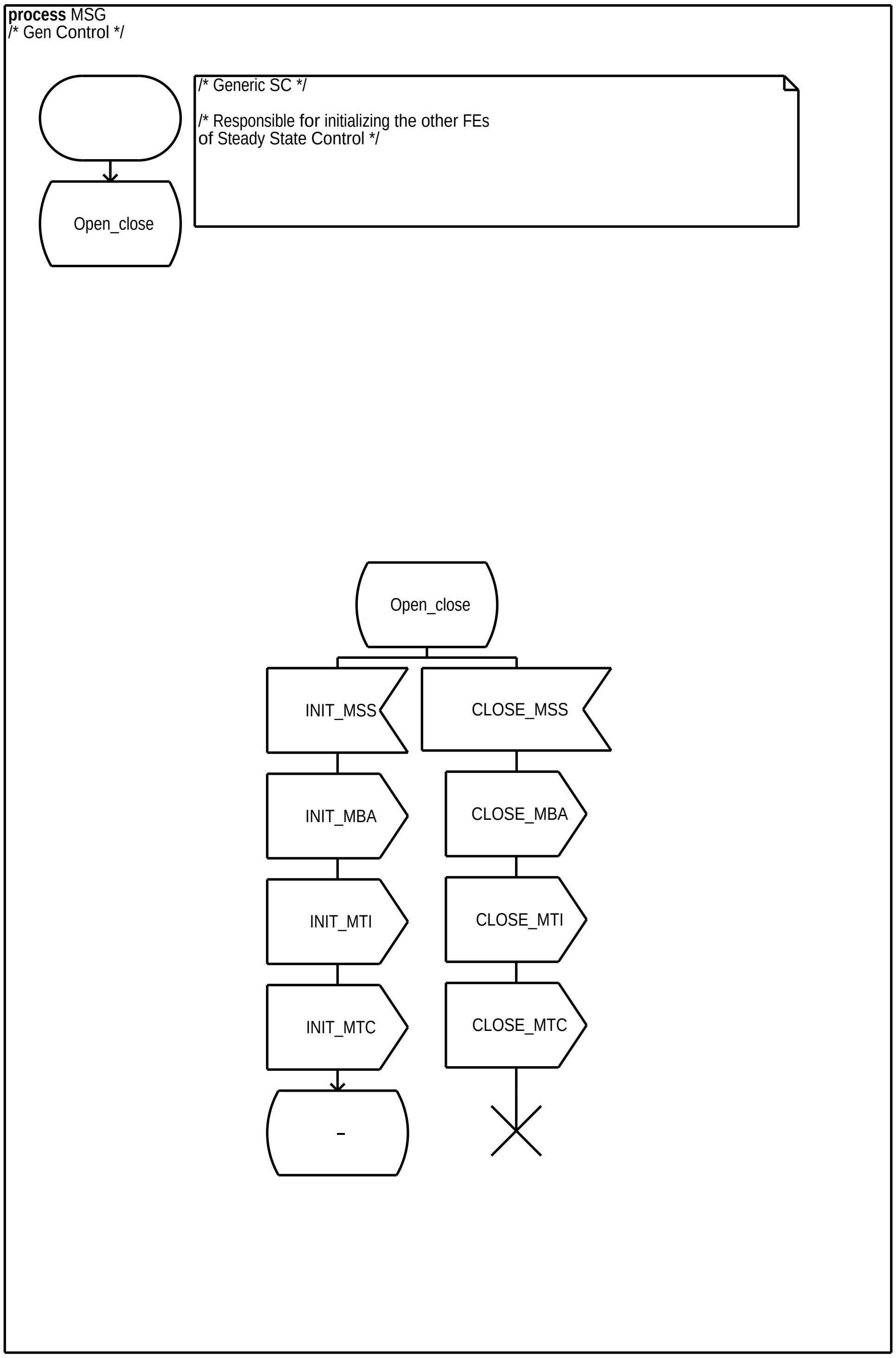


/* Steady State Control*/

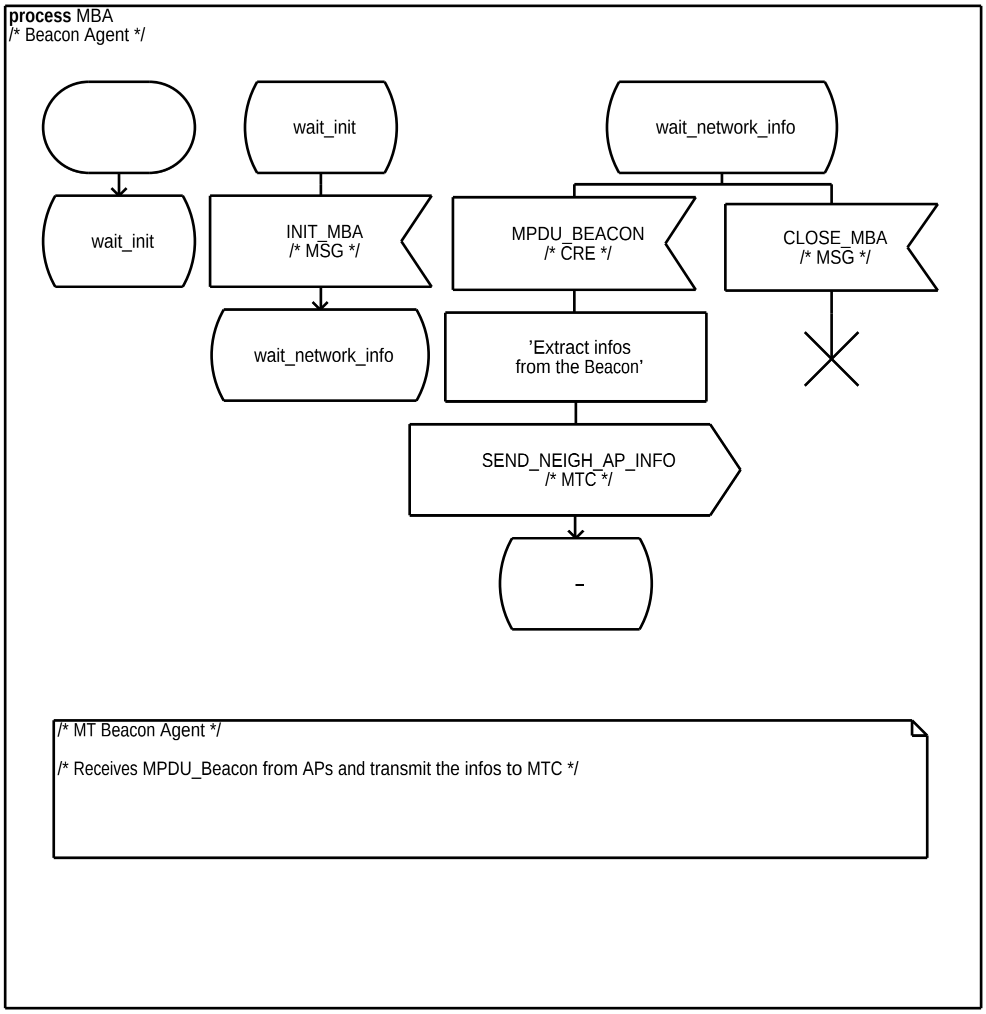
/* These entities corrspond to static control functions.

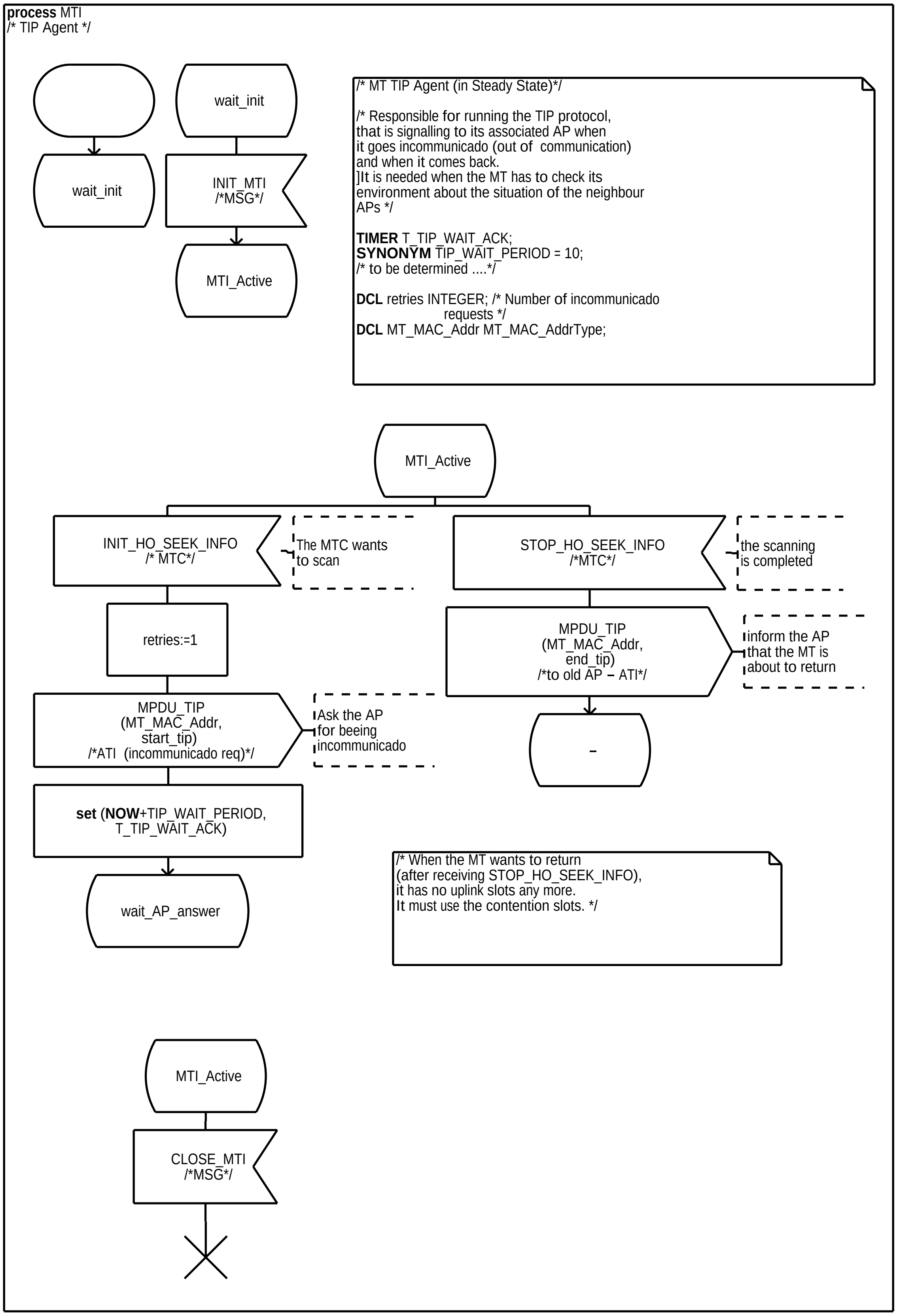
- MBA : Reception of the beacons. Transmit infos to the MTC
- MTI : Runs the TIP protocol */

```
SIGNAL INIT_MBA, CLOSE_MBA, INIT_MTI, CLOSE_MTI, INIT_MTC, CLOSE_MTC,
SEND_NEIGH_AP_INFO ;

SIGNAL INIT_HO_SEEK_INFO, STOP_HO_SEEK_INFO, INIT_HO_SEEK_INFO_OK,
   INIT_HO_SEEK_INFO_KO;

SIGNALLIST MMTC2MMTI = INIT_HO_SEEK_INFO, STOP_HO_SEEK_INFO ;
SIGNALLIST MMTI2MMTC = INIT_HO_SEEK_INFO_OK, INIT_HO_SEEK_INFO_KO;
```

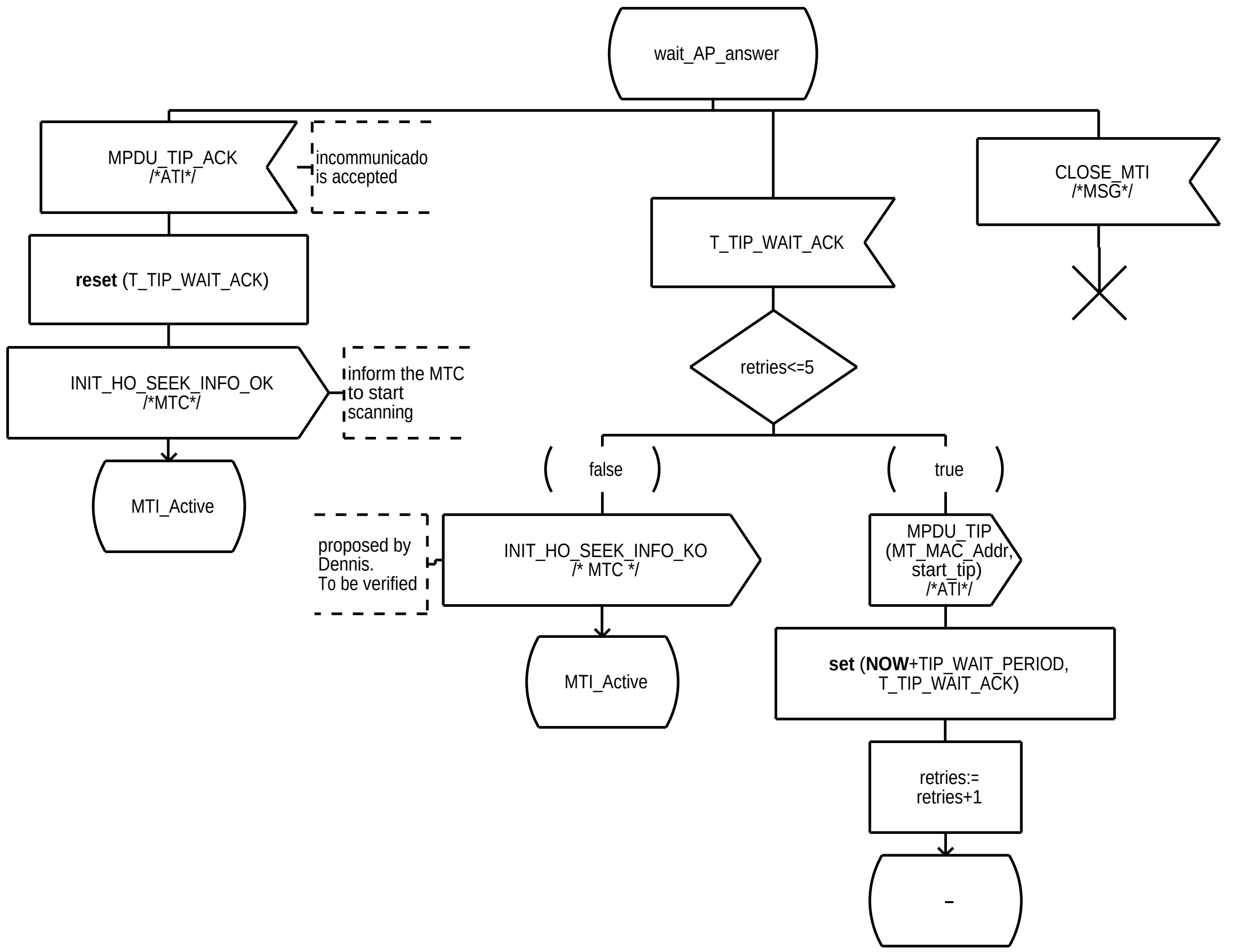


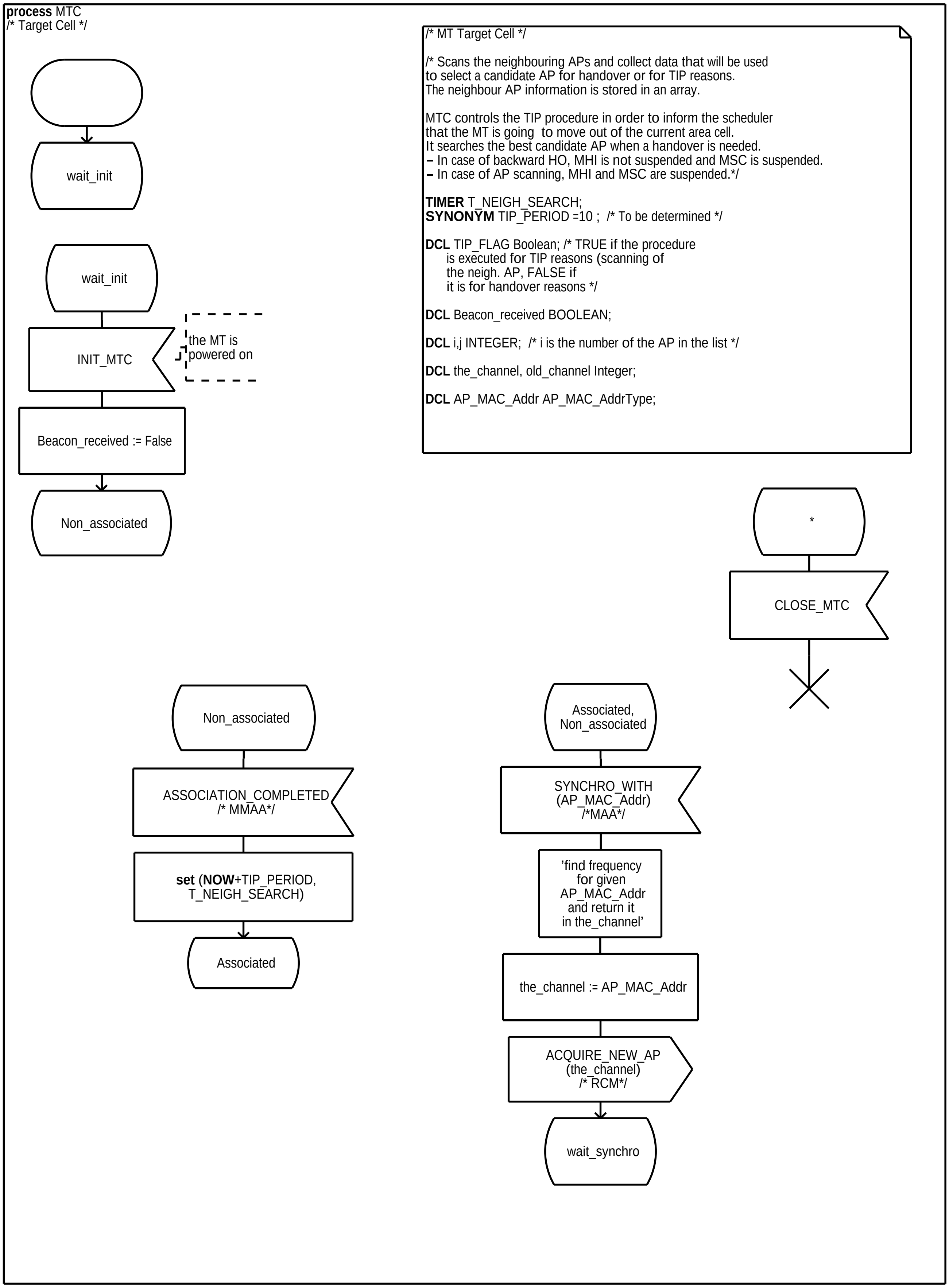
process MSG
/* Gen Control */
/* Generic SC */
/* Responsible for initializing the other FEs
of Steady State Control */
Open_close
Open_close
INIT_MSS
CLOSE_MSS
INIT_MBA
CLOSE_MBA
INIT_MTI
CLOSE_MTI
INIT_MTC
CLOSE_MTC
–


process MBA
/* Beacon Agent */
wait_init
wait_network_info
wait_init
INIT_MBA
/* MSG */
wait_network_info
MPDU_BEACON
/* CRE */
CLOSE_MBA
/* MSG */
'Extract infos
from the Beacon'
SEND_NEIGH_AP_INFO
/* MTC */
-
/* MT Beacon Agent */
/* Receives MPDU_Beacon from APs and transmit the infos to MTC */

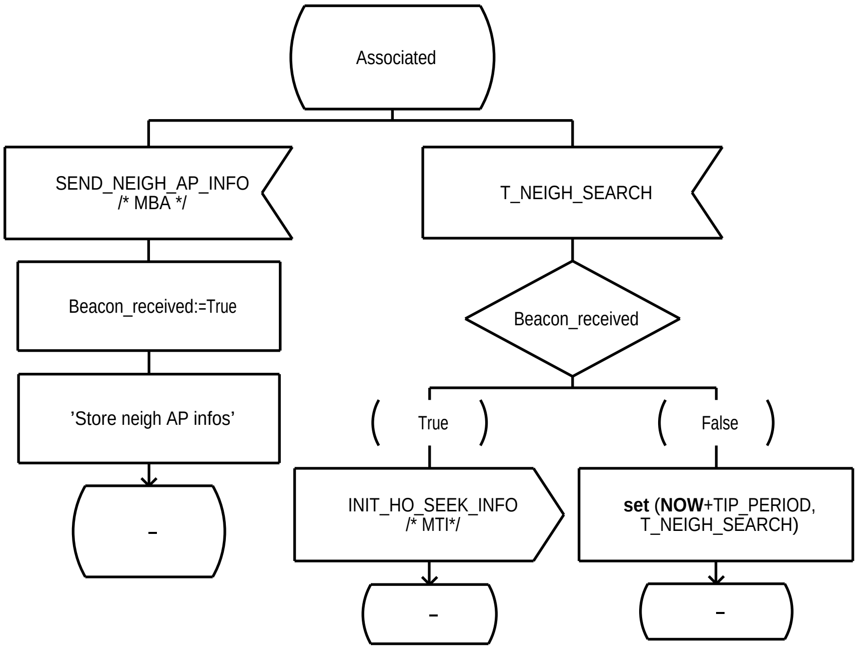

process MTI
/* TIP Agent */
wait_init
INIT_MTI
/*MSG*/
MTI_Active
/* MT TIP Agent (in Steady State)*/
/* Responsible for running the TIP protocol,
that is signalling to its associated AP when
it goes incommunicado (out of communication)
and when it comes back.
]It is needed when the MT has to check its
environment about the situation of the neighbour
APs */
TIMER T_TIP_WAIT_ACK;
SYNONYM TIP_WAIT_PERIOD = 10;
/* to be determined ....*/
DCL retries INTEGER; /* Number of incommunicado
requests */
DCL MT_MAC_Addr MT_MAC_AddrType;
MTI_Active
INIT_HO_SEEK_INFO
/* MTC*/
The MTC wants
to scan
STOP_HO_SEEK_INFO
/*MTC*/
the scanning
is completed
retries:=1
MPDU_TIP
(MT_MAC_Addr,
end_tip)
/*to old AP – ATI*/
inform the AP
that the MT is
about to return
MPDU_TIP
(MT_MAC_Addr,
start_tip)
/*ATI (incommunicado req)*/
Ask the AP
for beeing
incommunicado
–
set (NOW+TIP_WAIT_PERIOD,
T_TIP_WAIT_ACK)
wait_AP_answer
/* When the MT wants to return
(after receiving STOP_HO_SEEK_INFO),
it has no uplink slots any more.
It must use the contention slots. */
MTI_Active
CLOSE_MTI
/*MSG*/

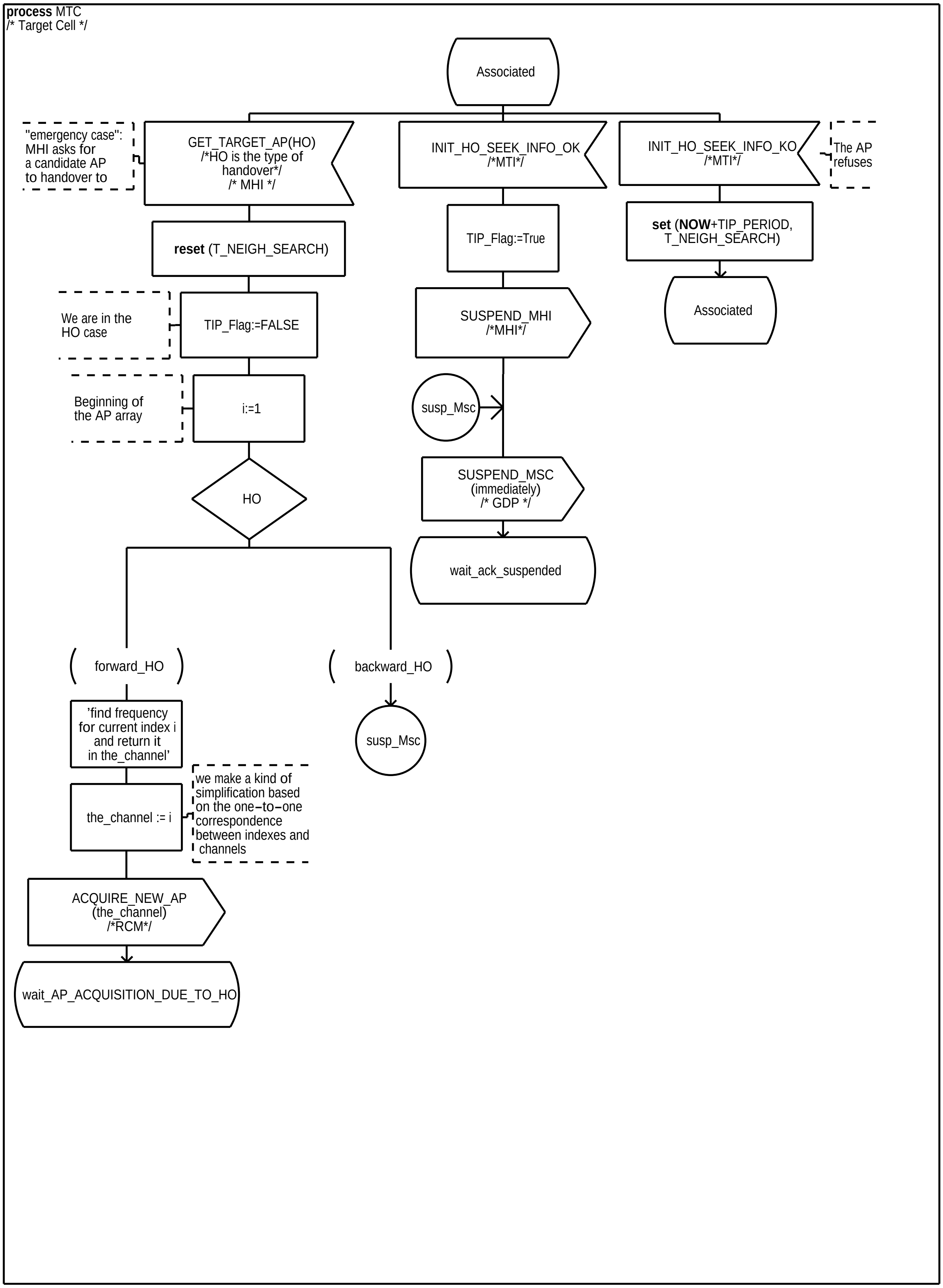
process MTI
/* TIP Agent */



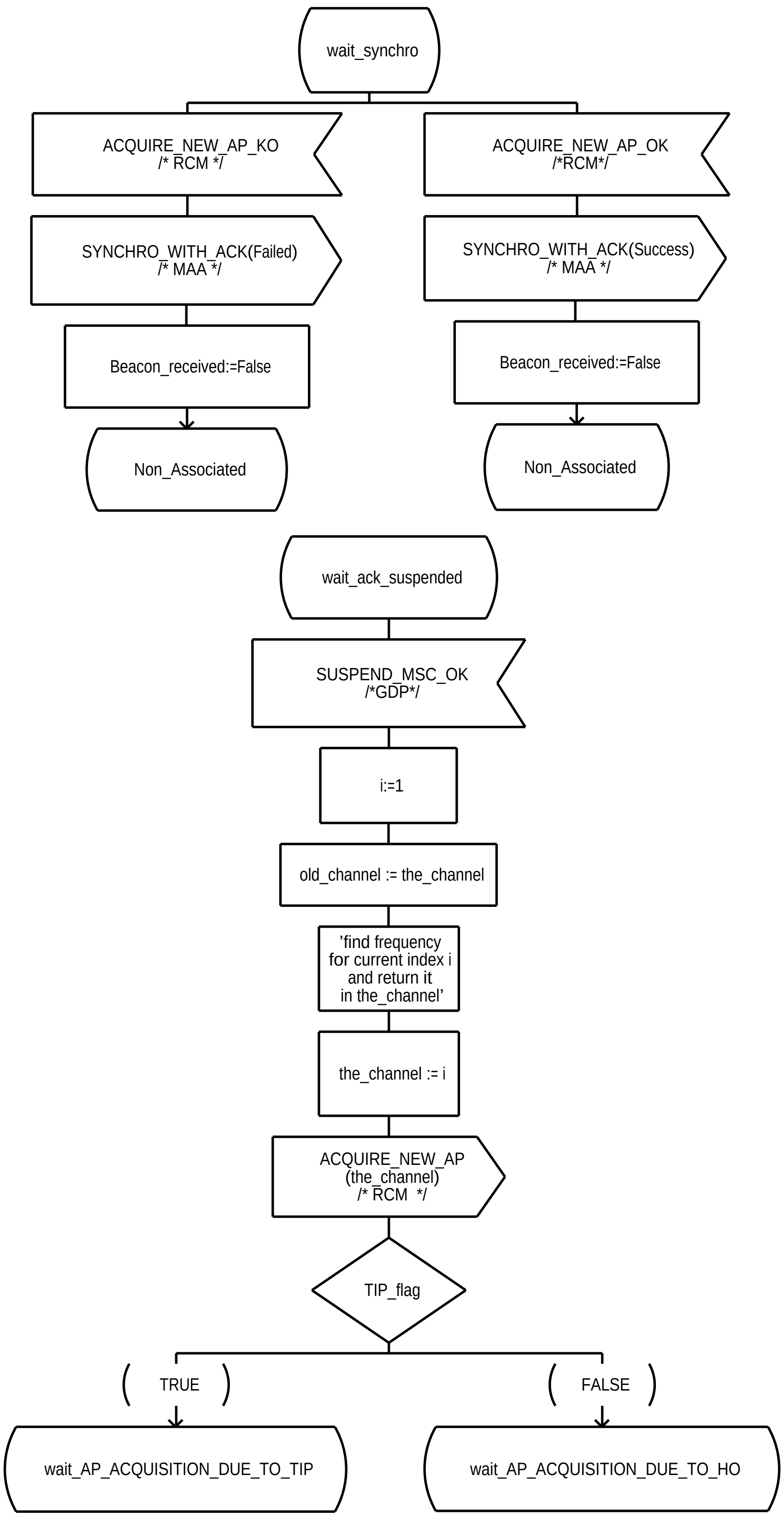

process MTC
/* Target Cell */
/* MT Target Cell */
/* Scans the neighbouring APs and collect data that will be used
to select a candidate AP for handover or for TIP reasons.
The neighbour AP information is stored in an array.
MTC controls the TIP procedure in order to inform the scheduler
that the MT is going to move out of the current area cell.
It searches the best candidate AP when a handover is needed.
– In case of backward HO, MHI is not suspended and MSC is suspended.
– In case of AP scanning, MHI and MSC are suspended.*/
TIMER T_NEIGH_SEARCH;
SYNONYM TIP_PERIOD =10 ; /* To be determined */
DCL TIP_FLAG Boolean; /* TRUE if the procedure
is executed for TIP reasons (scanning of
the neigh. AP, FALSE if
it is for handover reasons */
DCL Beacon_received BOOLEAN;
DCL i,j INTEGER; /* i is the number of the AP in the list */
DCL the_channel, old_channel Integer;
DCL AP_MAC_Addr AP_MAC_AddrType;
wait_init
wait_init
INIT_MTC
the MT is powered on
Beacon_received := False
Non_associated
*
CLOSE_MTC
Non_associated
ASSOCIATION_COMPLETED /* MMAA*/
set (**NOW**+TIP_PERIOD, T_NEIGH_SEARCH)
Associated
Associated, Non_associated
SYNCHRO_WITH (AP_MAC_Addr) /*MAA*/
'find frequency for given AP_MAC_Addr and return it in the_channel'
the_channel := AP_MAC_Addr
ACQUIRE_NEW_AP (the_channel) /* RCM*/
wait_synchro

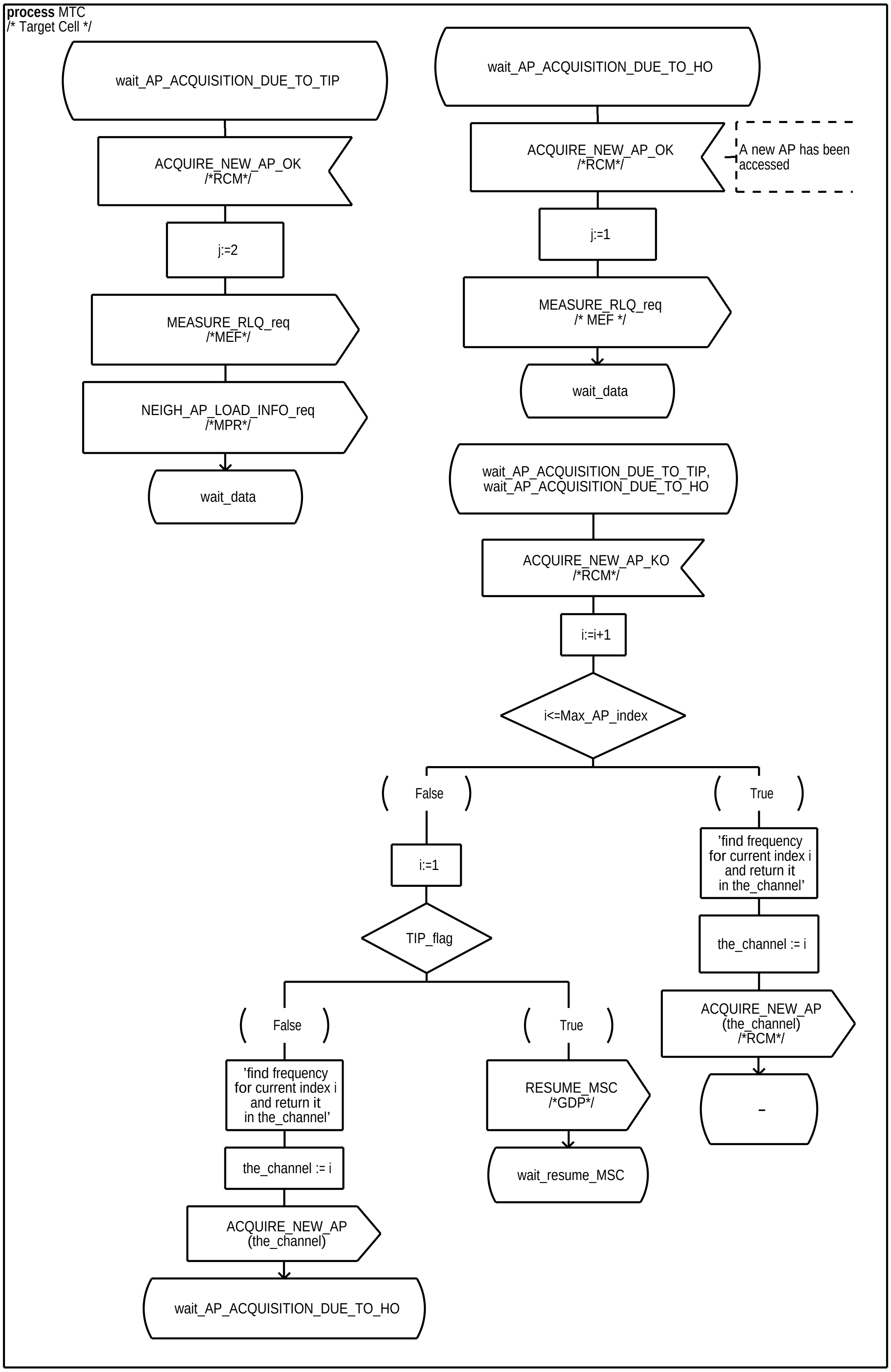
process MTC
/* Target Cell */

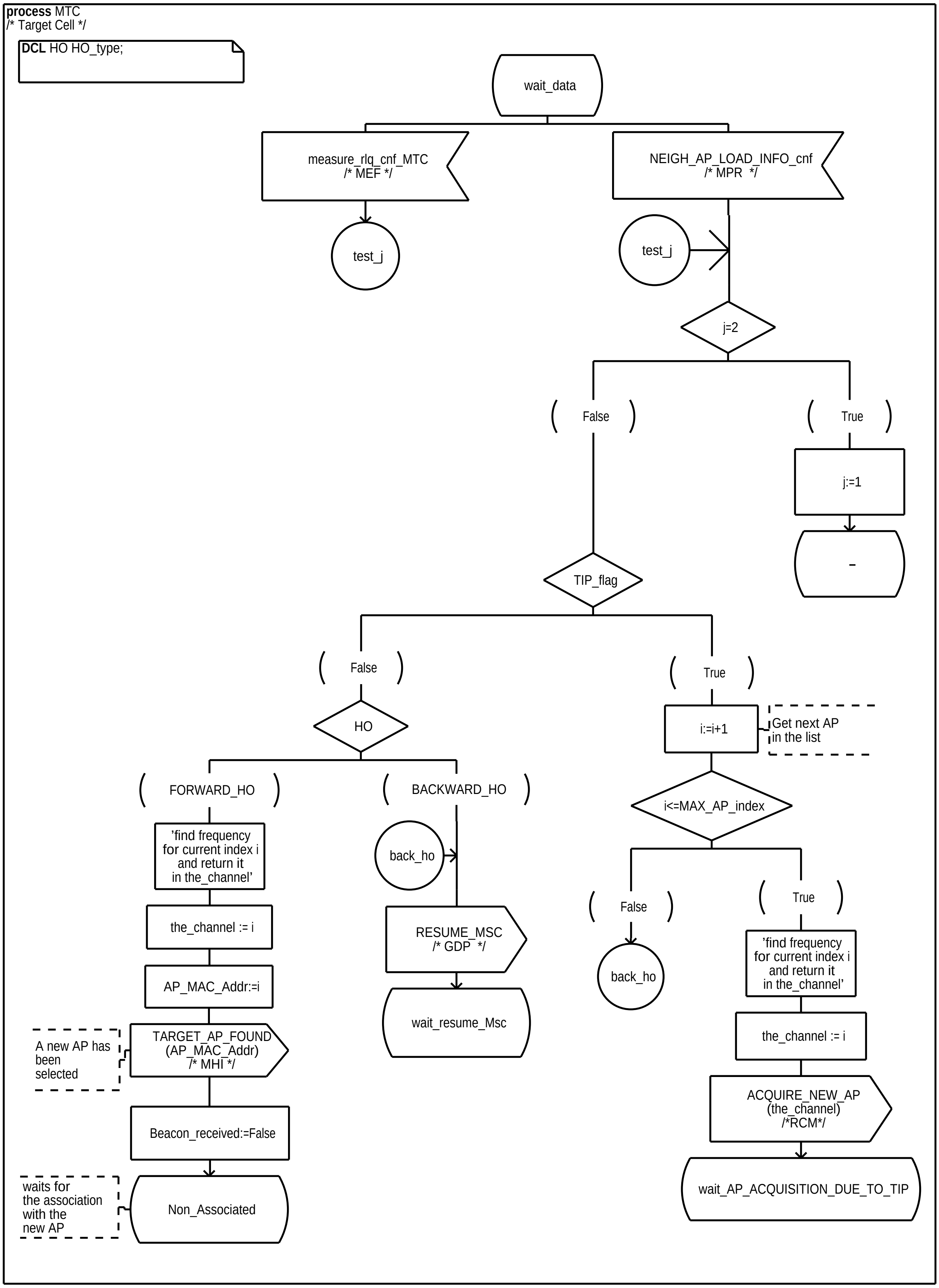


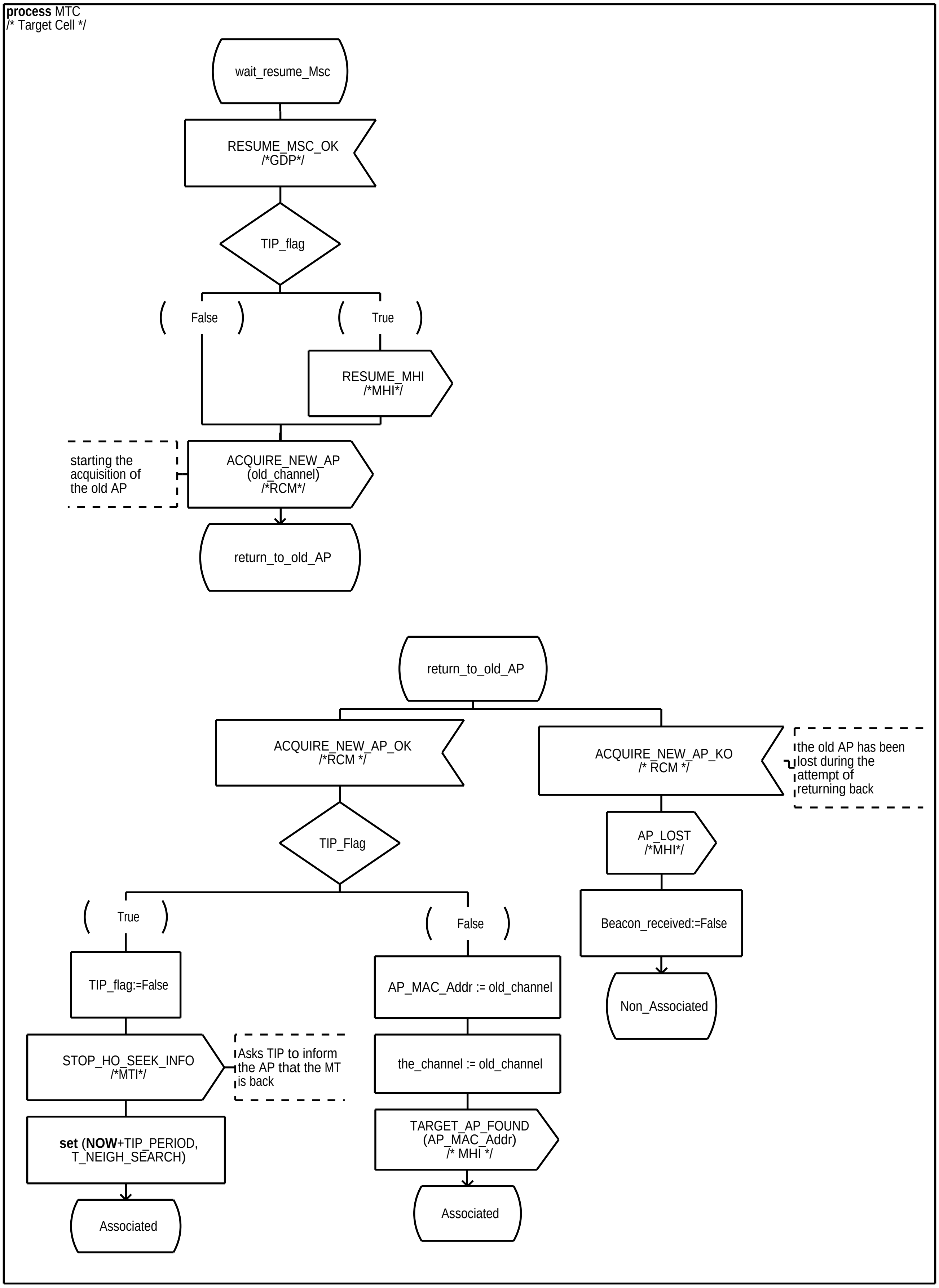

process MTC
/* Target Cell */
Associated
"emergency case":
MHI asks for
a candidate AP
to handover to
GET_TARGET_AP(HO)
/*HO is the type of
handover*/
/* MHI */
INIT_HO_SEEK_INFO_OK
/*MTI*/
INIT_HO_SEEK_INFO_KO
/*MTI*/
The AP
refuses
reset (T_NEIGH_SEARCH)
TIP_Flag:=True
set (NOW+TIP_PERIOD,
T_NEIGH_SEARCH)
Associated
We are in the
HO case
TIP_Flag:=FALSE
SUSPEND_MHI
/*MHI*/
Beginning of
the AP array
i:=1
susp_Msc
HO
SUSPEND_MSC
(immediately)
/* GDP */
wait_ack_suspended
forward_HO
backward_HO
susp_Msc
'find frequency
for current index i
and return it
in the_channel'
the_channel := i
we make a kind of
simplification based
on the one-to-one
correspondence
between indexes and
channels
ACQUIRE_NEW_AP
(the_channel)
/*RCM*/
wait_AP_ACQUISITION_DUE_TO_HO

process MTC
/* Target Cell */




process MTC
/* Target Cell */
wait_AP_ACQUISITION_DUE_TO_TIP
ACQUIRE_NEW_AP_OK /*RCM*/
j:=2
MEASURE_RLQ_req /*MEF*/
NEIGH_AP_LOAD_INFO_req /*MPR*/
wait_data
wait_AP_ACQUISITION_DUE_TO_HO
ACQUIRE_NEW_AP_OK /*RCM*/
A new AP has been accessed
j:=1
MEASURE_RLQ_req /* MEF */
wait_data
wait_AP_ACQUISITION_DUE_TO_TIP, wait_AP_ACQUISITION_DUE_TO_HO
ACQUIRE_NEW_AP_KO /*RCM*/
i:=i+1
i<=Max_AP_index
False
True
i:=1
TIP_flag
False
True
'find frequency for current index i and return it in the_channel'
the_channel := i
ACQUIRE_NEW_AP (the_channel)
wait_AP_ACQUISITION_DUE_TO_HO
RESUME_MSC /*GDP*/
wait_resume_MSC
'find frequency for current index i and return it in the_channel'
the_channel := i
ACQUIRE_NEW_AP (the_channel) /*RCM*/
-


process MTC
/* Target Cell */
DCL HO HO_type;
wait_data
measure_rlq_cnf_MTC
/* MEF */
NEIGH_AP_LOAD_INFO_cnf
/* MPR */
test_j
test_j
j=2
False
True
j:=1
–
TIP_flag
False
True
HO
i:=i+1
Get next AP
in the list
FORWARD_HO
BACKWARD_HO
i<=MAX_AP_index
'find frequency
for current index i
and return it
in the_channel'
back_ho
False
True
the_channel := i
RESUME_MSC
/* GDP */
back_ho
'find frequency
for current index i
and return it
in the_channel'
AP_MAC_Addr:=i
wait_resume_Msc
the_channel := i
A new AP has
been
selected
TARGET_AP_FOUND
(AP_MAC_Addr)
/* MHI */
ACQUIRE_NEW_AP
(the_channel)
/*RCM*/
Beacon_received:=False
wait_AP_ACQUISITION_DUE_TO_TIP
waits for
the association
with the
new AP
Non_Associated


process MTC
/* Target Cell */
wait_resume_Msc
RESUME_MSC_OK
/*GDP*/
TIP_flag
False
True
RESUME_MHI
/*MHI*/
starting the
acquisition of
the old AP
ACQUIRE_NEW_AP
(old_channel)
/*RCM*/
return_to_old_AP
return_to_old_AP
ACQUIRE_NEW_AP_OK
/*RCM */
ACQUIRE_NEW_AP_KO
/* RCM */
the old AP has been
lost during the
attempt of
returning back
AP_LOST
/*MHI*/
Beacon_received:=False
Non_Associated
TIP_Flag
True
False
TIP_flag:=False
AP_MAC_Addr := old_channel
STOP_HO_SEEK_INFO
/*MTI*/
Asks TIP to inform
the AP that the MT
is back
the_channel := old_channel
set (**NOW**+TIP_PERIOD,
T_NEIGH_SEARCH)
TARGET_AP_FOUND
(AP_MAC_Addr)
/* MHI */
Associated
Associated

Object	View	Page
FRAMEWORK	1	1
		2
SYSTEM wand		
PARTITION Constant_declarations	2	3
PARTITION general type_declarations	3	4
PARTITION type_declarations	4	5
PARTITION General PR declaration	5	6
PARTITION PR declarations AP	6	7
PARTITION PR Declaration MT	7	8
PARTITION DefaultPartition	8	9
		10
		11
BLOCK MT_Steady_State_Ctrl		
PARTITION DefaultPartition	9	12
		13
PROCESS MSG /* Gen Control */		
PARTITION DefaultPartition	10	14
PROCESS MBA /* Beacon Agent */		
PARTITION DefaultPartition	11	15
PROCESS MTI /* TIP Agent */		
PARTITION DefaultPartition	12	16
PARTITION wait_AP_answer	13	17
PROCESS MTC /* Target Cell */		
PARTITION Initial_state	14	18
PARTITION Associated	15	19
PARTITION Associated	16	20
PARTITION wait_synchro_and_suspend	17	21
PARTITION wait_AP_acquisition	18	22
PARTITION wait_data	19	23
PARTITION wait_resume_and_old_AP	20	24