

144. Multipoint MAC Control for Nx25G-EPON

144.4 Channel Control Protocol (CCP)

The Channel Control Protocol (CCP) allows the OLT to query and control the state of individual channels (see Table 143-7) through the exchange of CC_REQUEST and CC_RESPONSE CCPDUs, as defined in 144.4.2. Individual channels on the ONU and / or OLT may need to be disabled in a controlled manner for a variety of operational reasons, including power saving, scheduled diagnostic / maintenance activities, optical protection, and others. Any of the channels on the ONU and / or OLT may also fail, requiring a timely notification of the peer station of this fact.

144.4.1 Principles of Channel Control Protocol

The ONU channel lineup discovery is accomplished via an exchange of a CC_REQUEST and CC_RESPONSE CCPDUs. Using the CC_REQUEST CCPDU, the OLT polls the state of all downstream and upstream channels, without altering the state of any channels.

The ONU channel lineup discovery process may be initiated by the OLT MAC Control Client at any time.

Furthermore, the ONU may send an unsolicited CC_RESPONSE CCPDU to notify the OLT about any local changes in the channel lineup, including imminent transceiver element (transmitter and / or receive) failure, local channel disabling, power failure and resulting channel shutdown.

Once the CC_RESPONSE CCPDU from the ONU is received, the OLT saves the channel lineup information associated with the given ONU in the local MAC Control Client.

Once the ONU channel lineup discovery has been completed, the OLT MAC Control Client may change the state of any of the channels on the ONU by requesting the transmission of a CC_REQUEST CCPDU, with specific action encoded for each and every downstream and upstream channel for the given ONU. The OLT may enable a channel, disable a channel, or poll the current channel status.

The OLT may request the ONU to implement changes for the given channel in a persistent or non-persistent manner. Any persistent changes are preserved across ONU reset events. Any non-persistent changes are reverted upon ONU reset and re-registration.

The ONU receives the CC_REQUEST CCPDU, saves the request information in the local MAC Control Client, and attempts to implement the requested changes to the channel lineup. The CC_RESPONSE CCPDU is transmitted back to the OLT with the outcome of the configuration changes for each and every supported channel as well as the current channel state after the change. The ONU updates the *ChState* variable to match the new upstream channel lineup (see 144.3.6.3) and purges any pending transmission envelopes for disabled upstream channels.

The sequence of events on the OLT and ONU, when enabling / disabling one of downstream and upstream channels, is outlined in the following subclauses.

144.4.1.1 Disabling a downstream channel at an ONU

Disabling a downstream channel at an ONU results in the ONU receiver for the given channel being powered down. It is an ONU implementation choice to also shut down the associated receive path through PCS and MCRS sublayers. If the receive path associated with a disabled channel remains active, it shall not result in any accumulation of statistics or increment of error counters on this channel, such as invalid receive blocks or uncorrectable FEC codewords. Disabling a downstream channel at an ONU does not result in the given channel also being disabled at the OLT. The OLT data path and transmitter for the given channel may

remain active if there are other ONUs configured to receive the data transmitted on this downstream channel.

To disable a downstream channel (DC_n) at an ONU, the Channel Control Protocol performs the following sequence of steps:

- 1) MAC Control Client in the OLT stops transmitting any data to the target ONU on the channel being disabled. The said ONU shortly thereafter stops receiving any data on this downstream channel.
- 2) MAC Control Client initiates transmission of a unicast $CC_REQUEST$ CCPDU to the target ONU, requesting the ONU to disable the downstream channel DC_n .
 - i) The OLT starts the *ccp_timer* to determine whether a response from the ONU is missing.
 - ii) If a $CC_RESPONSE$ CCPDU did not arrive before the timer expiration, the CCP at the OLT will retransmit the $CC_REQUEST$ CCPDU at most *CCP_RETRY_LIMIT* times.
- 3) Once the ONU completed the request, the ONU sends a unicast $CC_RESPONSE$ CCPDU to the OLT, indicating the outcome of the requested configuration changes and the new state of all of its downstream and upstream channels.

144.4.1.2 Enabling a downstream channel at an ONU

Enabling a downstream channel at an ONU results in the ONU receiver for the given channel being powered up. Depending on the ONU implementation, the associated receive path through PCS and MCRS sublayers in the ONU may also need to be powered up.

To enable a downstream channel (DC_n) at an ONU, the Channel Control Protocol performs the following sequence of steps:

- 1) MAC Control Client initiates transmission of a unicast $CC_REQUEST$ CCPDU to the target ONU, requesting the ONU to enable the downstream channel DC_n .
 - i) The OLT starts the *ccp_timer* to determine whether a response from the ONU is missing.
 - ii) If a $CC_RESPONSE$ CCPDU did not arrive before the timer expiration, the CCP at the OLT will retransmit the $CC_REQUEST$ CCPDU at most *CCP_RETRY_LIMIT* times.
- 2) Once the ONU completed the request, the ONU sends a unicast $CC_RESPONSE$ CCPDU to the OLT, indicating the outcome of the requested configuration changes and the new state of all of its downstream and upstream channels.
- 3) MAC Control Client in the OLT starts transmitting data to the target ONU on the channel being enabled only when the given downstream channel is confirmed to have been enabled on the ONU. The said ONU shortly thereafter starts receiving data on this downstream channel.

144.4.1.3 Disabling an upstream channel at an ONU

Disabling an upstream channel at an ONU results in the ONU transmitter for the given channel being powered down. It is an ONU implementation choice to also shut down the associated transmit path through PCS and MCRS sublayers. If the transmit path associated with a disabled channel remains active, it shall not result in any accumulation of statistics or increment of error counters on this channel, such as packets or octets transmitted. Disabling an upstream channel at an ONU does not result in the given channel also being disabled at the OLT. The OLT data path and receiver for the given channel may remain active if there are other ONUs configured to transmit data on this upstream channel.

To disable an upstream channel (UC_n) at an ONU, the Channel Control Protocol performs the following sequence of steps:

- 1) MAC Control Client initiates transmission of a unicast $CC_REQUEST$ CCPDU to the target ONU, requesting the ONU to enable the upstream channel UC_n .

- i) The OLT starts the *ccp_timer* to determine whether a response from the ONU is missing. 1
- ii) If a CC_RESPONSE CCPDU did not arrive before the timer expiration, the CCP at the 2
OLT will retransmit the CC_REQUEST CCPDU at most *CCP_RETRY_LIMIT* times. 3
- 2) MAC Control Client in the OLT continues to grant the upstream channel UC_n on the target 4
ONU. 5
- 3) Once the ONU completed the request, the ONU sends a unicast CC_RESPONSE CCPDU to 6
the OLT, indicating the outcome of the requested configuration changes and the new state of all 7
of its downstream and upstream channels. ONU also purges any pending upstream transmis- 8
sion envelopes scheduled for the now disabled upstream channel. 9
- 4) MAC Control Client in the OLT stops granting the upstream channel UC_n on the target ONU 10
only when the given upstream channel is confirmed to have been disabled on the ONU. 11

144.4.1.4 Enabling an upstream channel at an ONU 12 13

Enabling an upstream channel at an ONU results in the ONU transmitter for the given channel being pow- 14
ered up. Depending on the ONU implementation, the associated transmit path through PCS and MCRS sub- 15
layers in the ONU may also need to be powered up. 16
17

To enable an upstream channel (UC_n) at an ONU, the Channel Control Protocol performs the following 18
sequence of steps: 19

- 1) MAC Control Client initiates transmission of a unicast CC_REQUEST CCPDU to the target 20
ONU, requesting the ONU to enable the upstream channel UC_n. 21
22
- i) The OLT starts the *ccp_timer* to determine whether a response from the ONU is missing. 23
24
- ii) If a CC_RESPONSE CCPDU did not arrive before the timer expiration, the CCP at the 25
OLT will retransmit the CC_REQUEST CCPDU at most *CCP_RETRY_LIMIT* times. 26
- 2) Once the ONU completed the request, the ONU sends a unicast CC_RESPONSE CCPDU to 27
the OLT, indicating the outcome of the requested configuration changes and the new state of all 28
of its downstream and upstream channels. 29
- 3) MAC Control Client in the OLT starts granting the upstream channel UC_n on the target ONU 30
only when the given upstream channel is confirmed to have been enabled on the ONU. 31

144.4.1.5 Local channel state changes at an ONU 32 33

MAC Control Client may monitor and react to the changes in the state of the downstream and/or upstream 34
channels, allowing ONU notify the OLT of observed or expected channel state changes. For example, MAC 35
Control Client may have ability to detect failure of one of channel receivers. 36
37

To notify MAC Control Client at the OLT about a local channel state change, the Channel Control Protocol 38
performs the following sequence of steps: 39

- 1) When a channel state change is identified, MAC Control Client at the ONU sends an unsolic- 40
ited unicast CC_RESPONSE CCPDU to the OLT, indicating the new state of all of its down- 41
stream and upstream channels. 42
43
- 2) If a state change for an upstream channel is being reported, MAC Control Client in the OLT 44
may adjust granting for the affected upstream channel. Similarly, if a state change for a down- 45
stream channel is being reported, MAC Control Client in the OLT may adjust downstream 46
transmission on the affected downstream channel. 47
48

144.4.2 CCPDU structure and encoding 49 50

The CCPDU structure is shown in Figure 144–29, and is further defined as follows: 51
52

- a) Destination Address (DA). The DA in CCPDU is the MAC Control Multicast address as specified in the annexes to Clause 31, or the individual MAC address associated with the PLID to which the CCPDU is destined.
- b) Source Address (SA). The SA in CCPDU is the individual MAC address associated with the PLID through which the CCPDU is transmitted. For CCPDUs originating at the OLT end, this can be the address any of the individual MACs. These MACs may all share a single unicast address, as explained in 144.6.2.
- c) Length/Type. The Length/Type in CCPDUs carries the MAC_Control_Type field value as specified in 31.4.1.3.
- d) Opcode. The opcode identifies the specific CCPDU being encapsulated. Values are defined in Table 31A-1.
- e) Data/Reserved/PAD. These 44 octets are used for the payload of the CCPDUs. When not used they are filled with zeros on transmission, and ignored on reception.
- f) FCS. This field is the Frame Check Sequence, typically generated by the MAC.

Fields within a frame are transmitted from top to bottom. Octets within each field are transmitted from least significant to most significant. Bits within each octet are transmitted from LSB to MSB.

| | Octets |
|-----------------------|--------|
| Destination Address | 6 |
| Source Address | 6 |
| Length/Type = 0x88-08 | 2 |
| Opcode | 2 |
| Data/Reserved/Pad | 44 |
| FCS | 4 |

Figure 144-29—Generic CCPDU

144.4.2.1 CC_REQUEST description

The CC_REQUEST message allows the OLT to query the state of ONU channel(s) or change the state of ONU channel(s), depending on the specific action code carried in the action field associated with the given upstream or downstream channel.

The CC_REQUEST CCPDU is an instantiation of the Generic CCPDU and shall be as shown in Figure 144-30 with details defined as follows:

- a) Opcode. The opcode for the CC_REQUEST CCPDU is 0x00-20.
- b) ActionDC0. The action encoding for the downstream channel 0 (DC0), as defined in Table 144-10.
- c) ActionDC1. The action encoding for the downstream channel 1 (DC1), as defined in Table 144-10.
- d) ActionUC0. The action encoding for the upstream channel 0 (UC0), as defined in Table 144-10.

- e) ActionUC1. The action encoding for the upstream channel 1 (UC1), as defined in Table 144–10.

Table 144–10—Channel Action field

| Bit | Field Name | Value | Description |
|-----|--------------------------------|-----------|---|
| 0-3 | ActionCode | 0x0 | No action |
| | | 0x1 | Disable channel |
| | | 0x2 | Enable channel |
| | | 0x3 - 0xF | Reserved, ignored on reception |
| 4-6 | Reserved, ignored on reception | | |
| 7 | PersistenceFlag | 0 | Upon reset, the channel reverts to its previous <u>persistent</u> state |
| | | 1 | The channel preserves its state across the reset event |

| | Octets |
|-----------------------|--------|
| Destination Address | 6 |
| Source Address | 6 |
| Length/Type = 0x88-08 | 2 |
| Opcode = 0x00-20 | 2 |
| ActionDC0 | 1 |
| ActionDC1 | 1 |
| Reserved | 14 |
| ActionUC0 | 1 |
| ActionUC1 | 1 |
| Reserved | 26 |
| FCS | 4 |

Figure 144–30—CC_REQUEST CCPDU

144.4.2.2 CC_RESPONSE description

The CC_RESPONSE message allows the ONU to report current channel(s) state and/or provide the OLT with the action result code, responding to the last requested action for each and every channel on the ONU.

The CC_RESPONSE CCPDU is an instantiation of the Generic CCPDU and shall be as shown in Figure 144–31 with details defined as follows:

- Opcode. The opcode for the CC_RESPONSE CCPDU is 0x00-21.
- StatusDC0. The state and action result encoding for the downstream channel 0 (DC0), as defined in Table 144–11.
- StatusDC1. The state and action result encoding for the downstream channel 1 (DC1), as defined in Table 144–11.
- StatusUC0. The state and action result encoding for the upstream channel 0 (UC0), as defined in Table 144–11.

- e) StatusUC1. The state and action result encoding for the upstream channel 1 (UC1), as defined in Table 144–10.

Table 144–11—Channel Status field

| Bit | Field Name | Value | Description |
|-----|------------------|-----------|---|
| 0-3 | ChannelState | 0x0 | Channel absent |
| | | 0x1 | Channel enabled |
| | | 0x2 | Channel disabled remotely (by the OLT) |
| | | 0x3 | Channel disabled locally (by the ONU) |
| | | 0x4 | Channel failure (PMD failure) |
| | | 0x5 - 0xF | Reserved, ignored on reception |
| 4-7 | ActionResultCode | 0x0 | No action requested |
| | | 0x1 | Action succeeded |
| | | 0x2 | Action failed |
| | | 0x3 | No change required, i.e., the channel is already in the requested state |
| | | 0x4 | Invalid command, i.e., the operation was requested for a non-existing channel |
| | | 0x5 - 0xF | Reserved, ignored on reception |

| | Octets |
|-----------------------|--------|
| Destination Address | 6 |
| Source Address | 6 |
| Length/Type = 0x88-08 | 2 |
| Opcode = 0x00-21 | 2 |
| StatusDC0 | 1 |
| StatusDC1 | 1 |
| Reserved | 14 |
| StatusUC0 | 1 |
| StatusUC1 | 1 |
| Reserved | 26 |
| FCS | 4 |

Figure 144–31—CC_RESPONSE CCPDU

144.4.3 Chanel Control Operation

144.4.3.1 Constants

ACTION_FAILED

This constant represents the value of *ActionResultCode* corresponding to “Action failed”, per Table 144–11.

Value: 0x2

| | |
|--|----|
| ACTION_SUCCEEDED | 1 |
| This constant represents the value of <i>ActionResultCode</i> corresponding to “Action succeeded”, per Table 144–11. | 2 |
| Value: 0x1 | 3 |
| | 4 |
| | 5 |
| ACT_DISABLE | 6 |
| This constant represents the value of <i>ActionCode</i> corresponding to “Disable channel”, per Table 144–10. | 7 |
| Value: 0x1 | 8 |
| | 9 |
| | 10 |
| ACT_ENABLE | 11 |
| This constant represents the value of <i>ActionCode</i> corresponding to “Enable channel”, per Table 144–10. | 12 |
| Value: 0x2 | 13 |
| | 14 |
| | 15 |
| CCP_TIMEOUT | 16 |
| This constant represents the maximum allowed interval of time between the transmission of CC_REQUEST CCPDU and the receipt of CC_RESPONSE from the given ONU, expressed in units of EQT. | 17 |
| Type: 32-bit unsigned integer | 18 |
| Value: 100 ms | 19 |
| | 20 |
| | 21 |
| | 22 |
| CCP_RETRY_LIMIT | 23 |
| This constant represents the maximum number of retransmissions for a CC_REQUEST CCPDU for the given ONU. | 24 |
| Type: 8-bit unsigned integer | 25 |
| VALUE: 3 | 26 |
| | 27 |
| | 28 |
| CH_STATE_ABSENT | 29 |
| This constant represents the value of <i>ChannelState</i> corresponding to “Channel absent”, per Table 144–10. | 30 |
| Value: 0x0 | 31 |
| | 32 |
| | 33 |
| CH_STATE_DISABLED_REMOTE | 34 |
| This constant represents the value of <i>ChannelState</i> corresponding to “Channel disabled remotely (by the OLT)”, per Table 144–10. | 35 |
| Value: 0x2 | 36 |
| | 37 |
| | 38 |
| CH_STATE_ENABLED | 39 |
| This constant represents the value of <i>ChannelState</i> corresponding to “Channel enabled”, per Table 144–10. | 40 |
| Value: 0x1 | 41 |
| | 42 |
| | 43 |
| ACT_NONE | 44 |
| This constant represents the value of <i>ActionCode</i> corresponding to “No action”, per Table 144–10. | 45 |
| Value: 0x0 | 46 |
| | 47 |
| | 48 |
| NO_ACTION_REQUESTED | 49 |
| This constant represents the value of <i>ActionResultCode</i> corresponding to “No action requested”, per Table 144–11. | 50 |
| Value: 0x0 | 51 |
| | 52 |
| | 53 |
| NO_CHANGE_REQUIRED | 54 |
| This constant represents the value of <i>ActionResultCode</i> corresponding to “No change required”, per | |

| | |
|--|--|
| Table 144–11. | 1 |
| Value: 0x3 | 2 |
| | 3 |
| 144.4.3.2 Variables | 4 |
| | 5 |
| ActionCode[] | 6 |
| TYPE: 4-element array of 4-bit unsigned integers | 7 |
| Each element of this array stores a copy of the action request for the given channel, as received in the CC_REQUEST CCPDU, with individual admissible values per Table 144–10, <i>ActionCode</i> field. This array is indexed with the channel designator, i.e., DC0, DC1, UC0, and UC1. | 8 9 10 |
| | 11 |
| CcpRetry | 12 |
| TYPE: 8-bit signed integer | 13 |
| This variable indicates the number of the current CCP_REQUEST CCPDU transmission attempt to the given ONU. | 14 15 |
| | 16 |
| ChState | 17 |
| This variable is defined in 144.3.6.3. {existing <i>ChStatus</i> variable needs to be renamed to <i>ChState</i> } | 18 19 |
| | 20 |
| PrevChState[] | 21 |
| TYPE: 4-element array of 4-bit unsigned integers | 22 |
| Each element of this array indicates the previous (old) status of the given upstream or downstream channel, with individual admissible values per Table 144–11, <i>ChannelState</i> field. This array is indexed with the channel designator, i.e., DC0, DC1, UC0, and UC1. | 23 24 25 |
| | 26 |
| 144.4.3.3 Functions | 27 |
| | 28 |
| GetResponseCode(chIndex) | 29 |
| This function generates an action response code for a channel identified by <i>chIndex</i> . The response code is determined by the channel state before the action was requested (<i>OldState</i>), the type of the requested action (<i>ActionCode</i>), and the channel state after the action was implemented (<i>NewState</i>). | 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 |


```

int4 GetResponseCode( int chIndex )
{
    OldState = PrevChState[ chIndex ];
    NewState = MsgChState[ chIndex ];
    Action =ActionCode[ chIndex ];

    if( Action == ACT_NONE )
        return NO_ACTION_REQUESTED;

    if( NewState == CH_STATE_ABSENT )
        return INVALID_COMMAND;

    if( Action == ACT_ENABLE )
    {
        if( NewState != CH_STATE_ENABLED )
            return ACTION_FAILED;
        else if( NewState == OldState )
            return NO_CHANGE_REQUIRED;
        else
            return ACTION_SUCCEEDED;
    }

    if( Action == ACT_DISABLE )
    {
        if( NewState != CH_STATE_DISABLED_REMOTE )
            return ACTION_FAILED;
        else if( NewState == OldState )
            return NO_CHANGE_REQUIRED;
        else
            return ACTION_SUCCEEDED;
    }

    return INVALID_COMMAND;
}

```

UpdateChState (chIndex, NewState)

This function updates the state record (*ChState*) for the given upstream channel *chIndex*, in the function of the new channel state *NewState*. When an upstream channel is disabled, this function purges the list of envelope descriptors for the given channel, stored in the EnvList array (see 144.3.6.3).

```

int4 UpdateChState( int chIndex, int NewState )
{
    if (NewState == CH_STATE_ENABLED )
        ChState<chIndex> = 1;
    else
    {
        ChState<chIndex> = 0;
        EnvList[chIndex].Clear();
    }
}

```

144.4.3.4 Timers

ccp_timer

This timer measures the time elapsed since the last transmission of CC_REQUEST CCPDU. If this

timer expires before a CC_RESPONSE CCPDU is received, the CC_REQUEST CCPDU is retransmitted (at most *CCP_RETRY* times).

144.4.3.5 Messages

MsgChRequest

A set of parameters (*operand_list*) carried in a CC_REQUEST CCPDU, as defined in 144.4.2.1.

MsgChResponse

A set of parameters (*operand_list*) carried in a CC_RESPONSE CCPDU, as defined in 144.4.2.2.

MsgChState

A set of parameters (*operand_list*) representing the current states of ONU's downstream and upstream channels. The *operand_list* is treated as an array, such that an individual channel state (array element) can be accessed using one of the channel index designators: DC0, DC1, UC0, and UC1 (see Table 143-7). Each channel state is represented by a 4-bit integer with permissible values as defined for the field *ChannelState* per Table 144-11. The *MsgChState* is generated by MAC Control Client and is processed (consumed) by the CCPDU Processing State Diagram in the ONU.

144.4.3.6 State diagrams

The OLT shall implement the CCP Processing State Diagram, as shown in Figure 144-32. The CCP Process in the OLT shall implement multiple instances of the CCP Processing state diagram shown in Figure 144-32 where each instance is associated with a MLID being registered. The ONU shall implement the CCP Processing State Diagram, as shown in Figure 144-33.

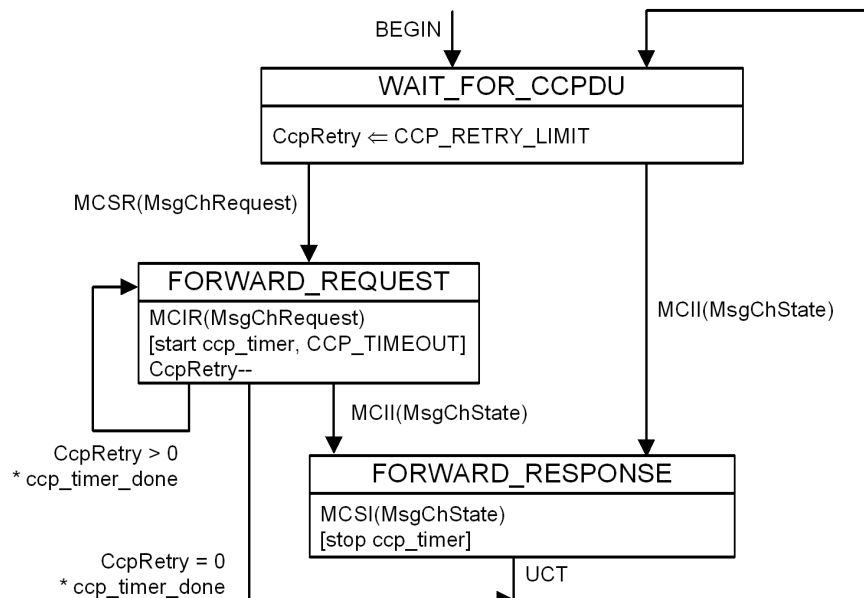


Figure 144-32—CCPDU Processing State Diagram, OLT

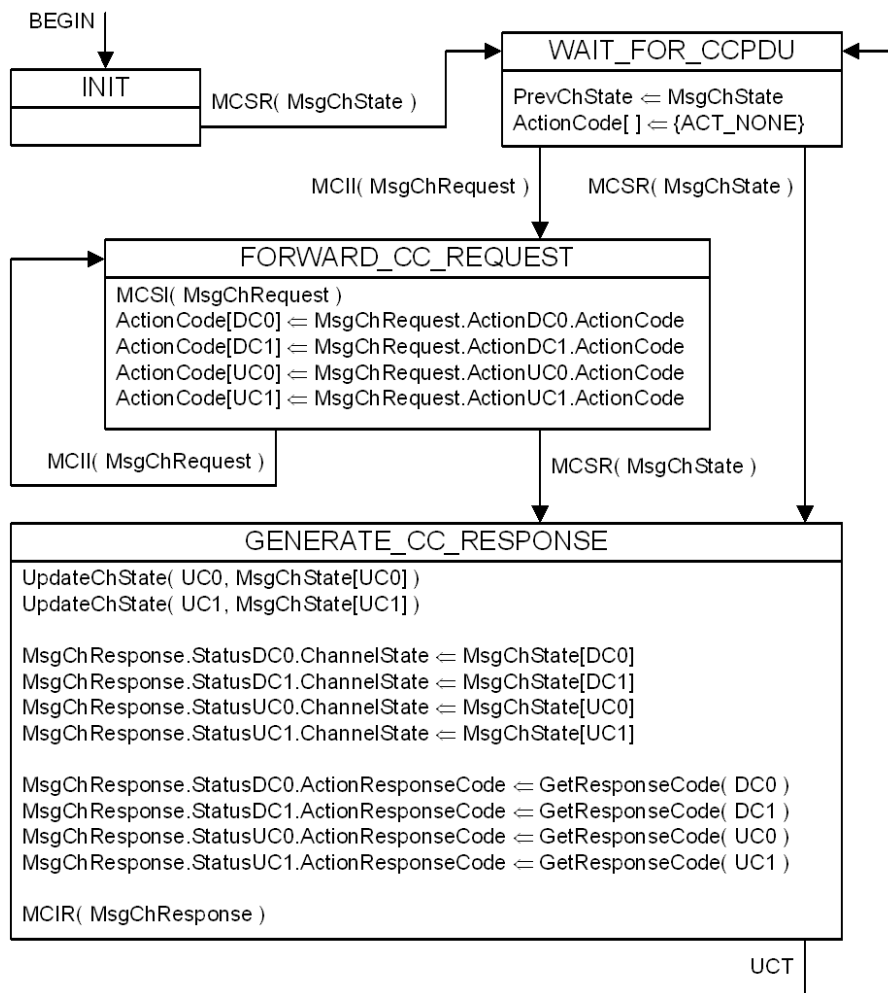


Figure 144–33—CCPDU Processing State Diagram, ONU

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54