



Issues with fast ONU re-registration in 802.3ca

Glen Kramer, Broadcom

Marek Hajduczenia, Charter Communications

- ONU Re-registration operation is described in 802.3ca

In 144.3.6.4

The OLT also may send the REGISTER MPCPDU to an already-registered ONU to request it to de-register or re-register. Such REQUEST MPCPDUs are sent in the envelopes with the unicast PLID assigned to the given ONU.

Table 144–5—REGISTER MPCPDU Flag field

Value	Indication	Comment
0	ACK	The ONU's requested registration is successful or a registered ONU is asked to re-register
1	NACK	The registration request is denied or a registered ONU is asked to deregister
2 to 255	Reserved	Ignored on reception

- ❑ The spec says that the OLT can request a registered ONU to de-register or re-register
 - De-registration means that ONU becomes unregistered and will participate in discovery and registration process (the ONU will send REGISTER_REQ MPCPDU)
 - Re-registration means that ONU will update its registration parameters (PLID, MLID, Sp1Length, Sp2Length, or Sp3Length) without becoming unregistered. After updating the parameters, ONU just sends REGISTER_ACK.

- ❑ ONU Registration state diagram 144-22 supports re-registration, but the OLT Registration state diagram 144-21 can never send a REGISTER MPCPDU for re-registration. This is an error in the state diagram.

ONU state diagram supports re-registration

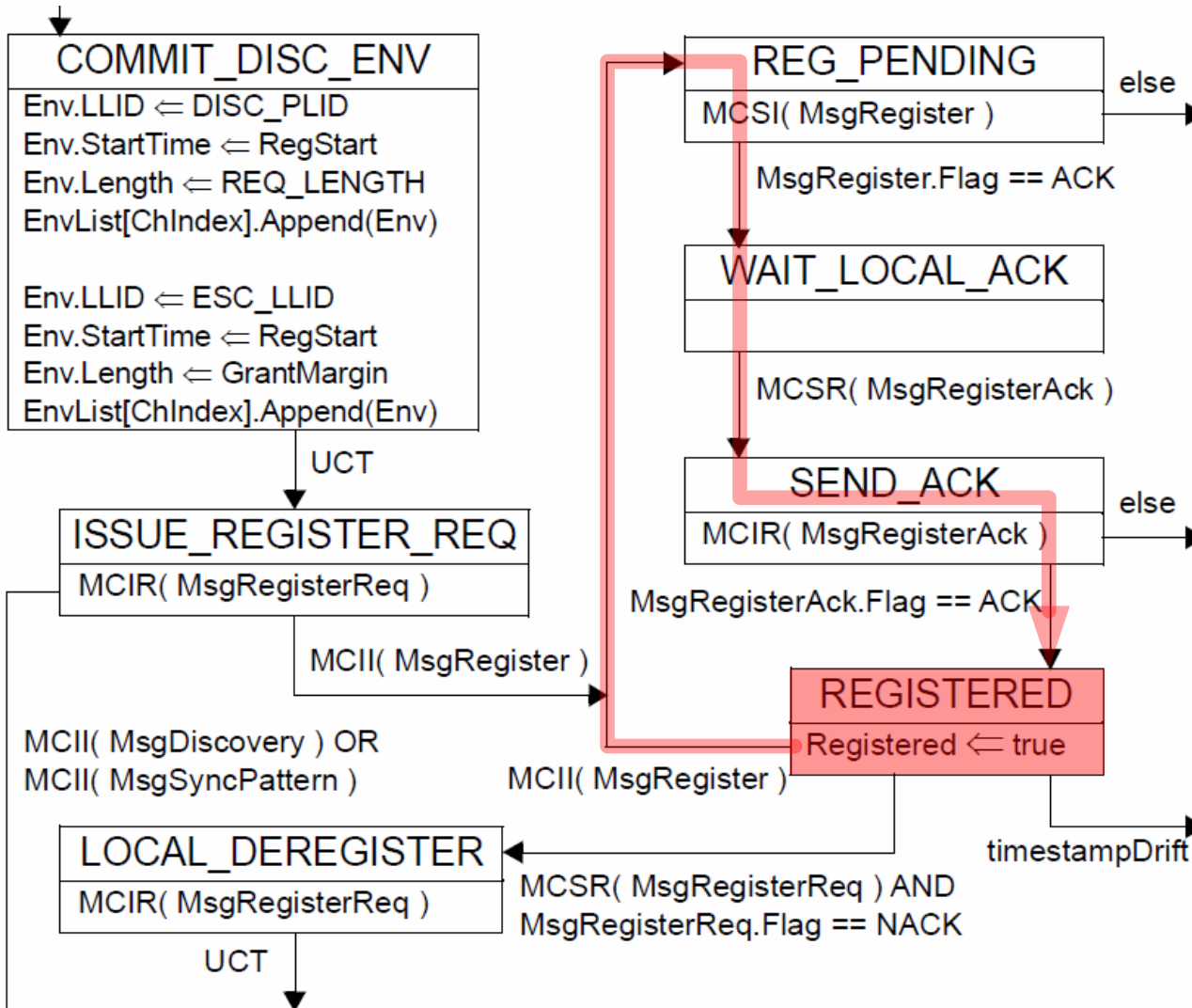
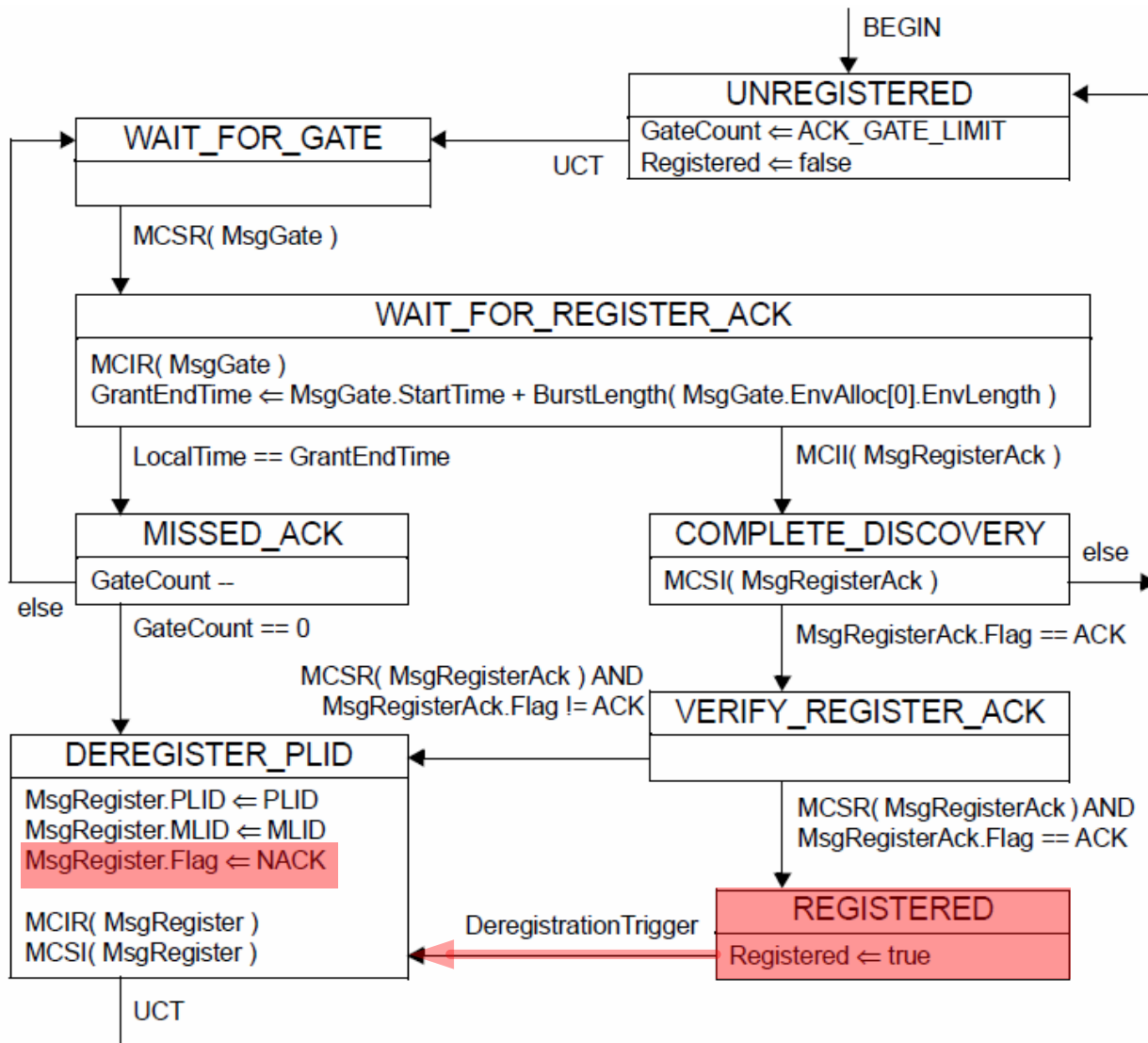


Figure 144-22—ONU Registration state diagram

1. A registered ONU receives REGISTER MPCPDU on its assigned PLID
2. ONU transitions to state REG_PENDING. *Registered* variable remains 'true'. Payload of REGISTER MPCPDU is passed to MPCP Client.
3. If REGISTER Flag was ACK (i.e., this is a re-registration), go to state WAIT_LOCAL_ACK and wait for the MPCP client to confirm setting new parameters.
4. When new parameters are confirmed, generate REGISTER_ACK MPCPDU and queue it for transmission (may be on a new PLID).
5. Return to REGISTERED state.

OLT state diagram



```

DeregistrationTrigger =
    // 1) ONU MPCP is unresponsive
    MissedReportCount == MISSED_REPORT_LIMIT OR

    // 2) Timestamp drift exceeded the safe margin
    TimestampDrift == true OR

    // 3) ONU requested deregistration
    ( MCII( MsgRegisterReq) AND MsgRegisterReq.Flag == NACK ) OR

    // 4) OLT MPMC client initiated ONU deregistration
    ( MCSR( MsgRegisterAck) AND MsgRegisterAck.Flag == NACK )
    
```

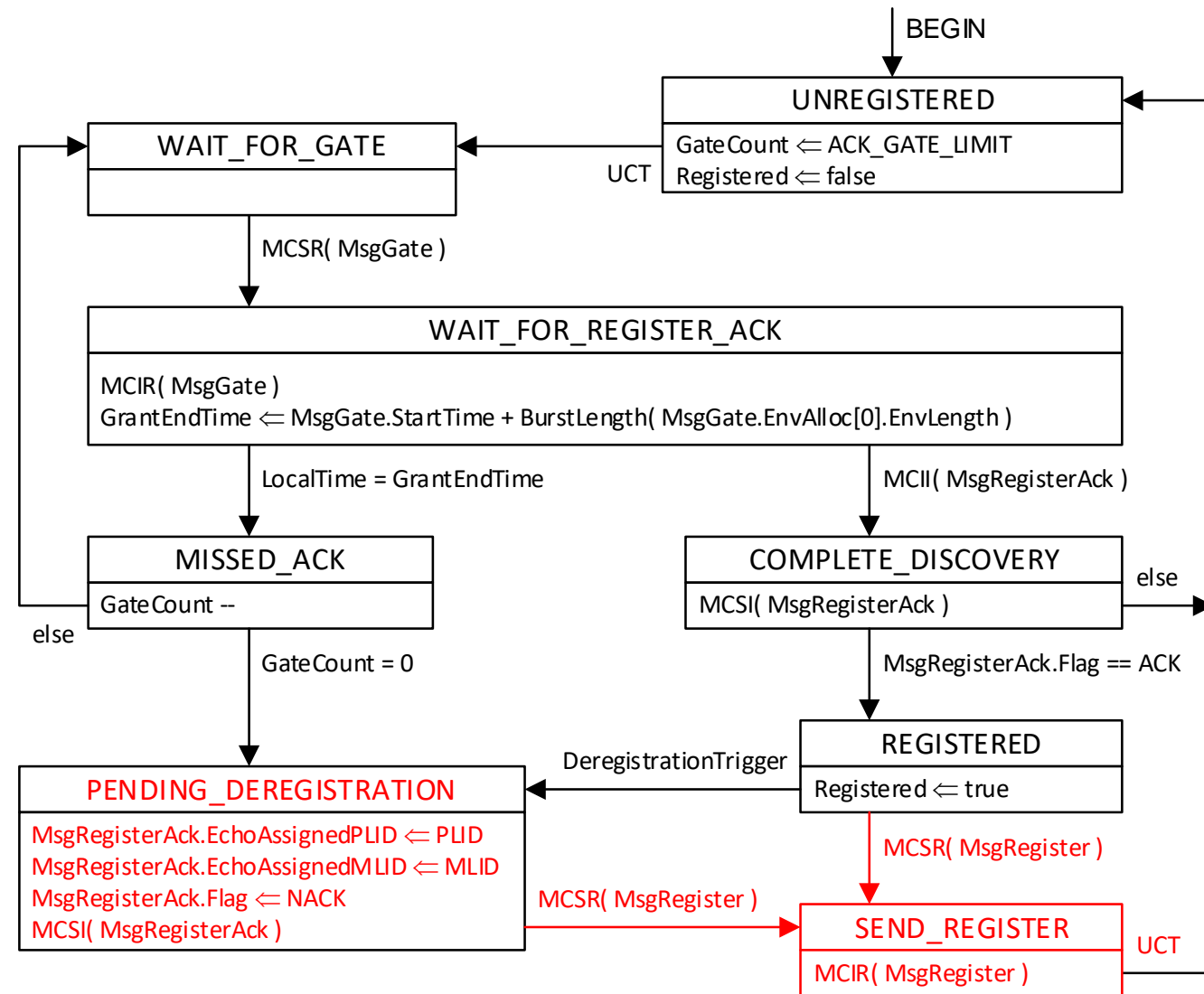
1. *DeregistrationTrigger* only activates when OLT wants to deregister the ONU
2. OLT always sends REGISTER MPCPDU with **Flag = NACK** (in state DEREGISTER_PLID)
3. Another issue: Deregistration REGISTER MPCPDU does not carry *Sp1Length*, *SP2Length*, *SP3Length* and *EchoPending-Envelopes*, in violation to 144.3.6.4.

Figure 144–21—OLT Registration Completion state diagram

Proposed Remedy (1/2)



- ❑ Modify OLT Registration Process state diagram (Fig 144-21) as shown.
 - ❑ In PENDING_DEREGISTRATION state, the OLT Registration Process informs the MPCP Client about the events that client cannot detect:
 1. REGISTER_ACK form ONU is not received after ACK_GATE_LIMIT attempts
 2. DeregistrationTrigger is activated (see next slide)
- (Message MsgRegisterAck is formed in the PENDING_DEREGISTRATION state and is passed to local MPCP Client as an indication, i.e., it is reused for local intra-layer signaling).
- ❑ The decision to de-register or re-register always comes from the MPCP client.



MCSI = MA_CONTROL.indication from MAC Control to MPCP client
MCSR = MA_CONTROL.request from MPCP client to MAC Control

- ❑ Modify the definition of the *DeregistrationTrigger* variable to only activate on external events that MPCP client needs to be informed of:

1. MPCP timeout
2. Timestamp drift
3. Received ONU's request to be deregistered

- ❑ The MPCP client does not need to be informed of a decision that it itself made, such as a decision to de-register or re-register an ONU.

DeregistrationTrigger

Type: Boolean

Description: This variable is set to true when at least one of multiple conditions for ONU deregistration becomes true. Otherwise, the variable is set to false. The *DeregistrationTrigger* is an alias for the following code:

```
DeregistrationTrigger =  
    // 1) ONU MPCP is unresponsive  
    MissedReportCount == MISSED_REPORT_LIMIT OR  
  
    // 2) Timestamp drift exceeded the safe margin  
    TimestampDrift == true OR  
  
    // 3) ONU requested deregistration  
    ( MCII(MsgRegisterReq) AND MsgRegisterReq.Flag == NACK ) OR  
  
    // 4) OLT MPMC client initiated ONU deregistration  
    ( MCSR(MsgRegisterAck) AND MsgRegisterAck.Flag == NACK )
```



Thank you