

- 1 ONU_timer[SA]
2 This set of timers, dereferenced by a 48 bit index according to the originating MAC
3 address, is used to measure the time since the arrival of the last message in the registration
4 process from the ONU known by its MAC Address == SA.
5 Failure of a message to arrive by the time the timer expires is a fatal error in the discovery
6 process, and causes registration to fail for the specified ONU, who must then retry to
7 register.
8 VALUE:
9
10 random_delay
11 This timer is used to measure a random delay inside the discovery window. The purpose of
12 the delay is to a priori reduce the probability of collision during the registration process, and
13 thus reduce the probability of invocation of the deferral process, thus lowering the
14 expectancy of registration time in the PON.
15 VALUE: A random function of the net discovery window size less the REGISTER_REQ
16 MPCPDU frame size less the idle period and laser turn on and off delays.
17
18 wait_for_register_msg
19 This timer is used to wait for the event signalling the arrival of a REGISTER MPCPDU at
20 the ONU. On expiring, the ONU assumes that it had encountered congestion while
21 attempting to register, and that the REGISTER_REQ MPCPDU was lost due to a collision.
22 As a result, a deferral process is initiated.
23 VALUE: This timer is governed by the constant *max_register_wait* as defined in 56.3.7.1.1.

56.3.7.1.5 Messages

~~Editor's note: To be removed prior to publication. Parameters of primitives should be more elaborately described either in a table or as list following primitive introduction. Possibly unify request/indication primitives to a single primitive per defined message in order to fall in opcode budget~~

MA_CONTROL.request(DA, ~~register_request~~registration, register)

The service primitive used by a client to request the Discovery Process to perform a registration. ~~This primitive may be called multiple times in order to register additional ports.~~

register: Boolean operands describing whether the station is attempting to register (true) or deregister (false).

MA_CONTROL.request(deregister)

The service primitive used by a client to request the Discovery Process to deregister the associated port.

When called with the non-default port for deregistration, the port would need to be subsequently deallocated. When called with the default port, the port reverts to the WAIT state, and will try to re-register at the earliest opportunity, once allowed.

~~Editor's note: To be removed prior to publication. A variation of this request primitive is to be invoked at the OLT following a dying gasp indication in order to deregister the ONU.~~

MA_CONTROL.indication(~~indication~~register_request, SA_list)

The service indication issued by the Discovery Process to notify the client and Layer Management that the registration process is in progress.

A list of source MAC addresses associated with the devices attempting to register are provided in the *SA_list* parameter.

MA_CONTROL.request(~~DA, register~~create_discovery_window, ~~DA~~, start_time, grant_length, length) 1
2

The service primitive used by a peer or client to request the Discovery Process to acknowledge the existence of a discovery window. 3
4

The *DA* parameter is the MAC address of the device requested to register in this window. The device may be recognized by its unicast address, or multiple devices may be requested when using the MAC Control well known multicast address. 5
6
7

The *start_time* parameter, holds the time (relative to the *local_time* counter), at which the discovery window will become active. 8
9

The *grant_length* parameter indicates the length of the discovery window as indicated by the accompanying GATE. 10
11

The *length* parameter indicates the length of the allocated discovery window in *time_quanta*. 12
13

When *Master = true* (i.e. OLT mode), and the function is invoked, a grant is issued with the relevant *start_time* and *length* parameters, addressed to *DA*. For a DTE where *Master = false* (i.e. ONU mode), this function may not be called by the MAC client, and rather it is invoked by the Gate Processing peer entity to signal the arrival of a grant used for discovery. 14
15
16
17
18

MA_CONTROL.indication(register_ack, ~~accepted~~, SA, ID, status, capability, acknowledged_capability, RTT) 19
20

The service indication issued by the Discovery Process to notify the client and Layer Management that the registration process has completed. 21
22

The MAC address of the reciprocating MAC (ONU address at the OLT, and OLT address at the ONU) is passed in the parameter *SA*. The LLID allocated to the ONU is passed in the parameter *ID*. 23
24
25

The parameter *status* holds the possible values: *accepted*, *denied*, *reset*, *deregistered*. 26

The parameter *capability* holds the 64 bit vector published by the far end, as well as the 64 bit vector (*acknowledged_capability*) returned by the far end after the registration completion. 27
28
29

The measured round trip time to/from the ONU is returned in the parameter *RTT*. *RTT* is stated in *time_quanta* units. This parameter holds a valid value only when the invoking Discovery Process is in the OLT (i.e. *Master = true*). 30
31
32
33

~~MA_CONTROL.indication(denied, SA, ID)~~ 34

~~The service indication issued by the Discovery Process to notify the client and Layer Management that the registration process has been denied. The MAC address of the reciprocating MAC (ONU address at the OLT, and OLT address at the ONU) is passed in the parameter *SA*. The LLID allocated to the ONU is passed in the parameter *ID*.~~ 35
36
37
38

~~MA_CONTROL.indication(deregistered, SA)~~ 39

~~The service indication issued by the Discovery Process to notify the client and Layer Management that the registration process has deregistered the port.~~ 40
41

~~The MAC address of the deregistering ONU is passed in the parameter *SA*, when the signal is invoked in the OLT, otherwise the parameter is empty.~~ 42
43
44

~~MA_CONTROL.indication(reset)~~ 45

~~The service indication issued by the Discovery Process to notify the client and Layer Management that the OLT has requested that all ports should be reset.~~ 46
47

OMP.request() 48

The service primitive used by a client to request an OMP functional block function with the specified request_operands. 49
50
51

OMP.indication() 52

The service primitive used by the OMP functional block to signal the client an event with specified parameters. 53
54