# 102. Multipoint MAC Control for EPoC

## 102.1 Overview

## 102.2 Multipoint MAC Control operation

## 102.3 Multipoint Control Protocol (MPCP)

### 102.3.1 Principles of Multipoint Control Protocol

### 102.3.2 Compatibility considerations

### 102.3.3 Discovery processing

Discovery is the process whereby newly connected or off-line CNUs are provided access to the EPoC. The process is driven by the CLT, which periodically makes available Discovery Windows during which off-line CNUs are given the opportunity to make themselves known to the CLT. The periodicity of these windows is unspecified and left up to the implementor. The CLT signifies that a discovery period is occurring by broadcasting a discovery GATE MPCPDU, which includes the starting time and length of the discovery window~~, along with the Discovery Information flag field, as defined in 102.3.6.1~~. With the appropriate settings of individual flags contained in this 16 bit wide field, the CLT notifies all the CNUs about its upstream and downstream channel transmission capabilities. Note that the CLT may simultaneously support more than one data rate in the given transmission direction.

Off-line CNUs, upon receiving a Discovery GATE MPCPDU, wait for the period to begin and then transmit a REGISTER_REQ MPCPDU to the CLT. Discovery windows are unique in that they are the only times when multiple CNUs can access the CCDN simultaneously, and transmission overlap can occur. In order to reduce transmission overlaps, a contention algorithm is used by all CNUs. Measures are taken to reduce the probability for overlaps by artificially simulating a random distribution of distances from the CLT. Each CNU waits a random amount of time before transmitting the REGISTER_REQ MPCPDU that is shorter than the length of the discovery window. It should be noted that multiple valid REGISTER_REQ MPCPDUs can be received by the CLT during a single discovery window. Included in the REGISTER_REQ MPCPDU is the CNU's MAC address and number of maximum pending grants. ~~Additionally, a registering CNU notifies the CLT of its transmission capabilities in the upstream and downstream channels by setting appropriately the flags in the Discovery Information field, as specified in 102.3.6.3.~~

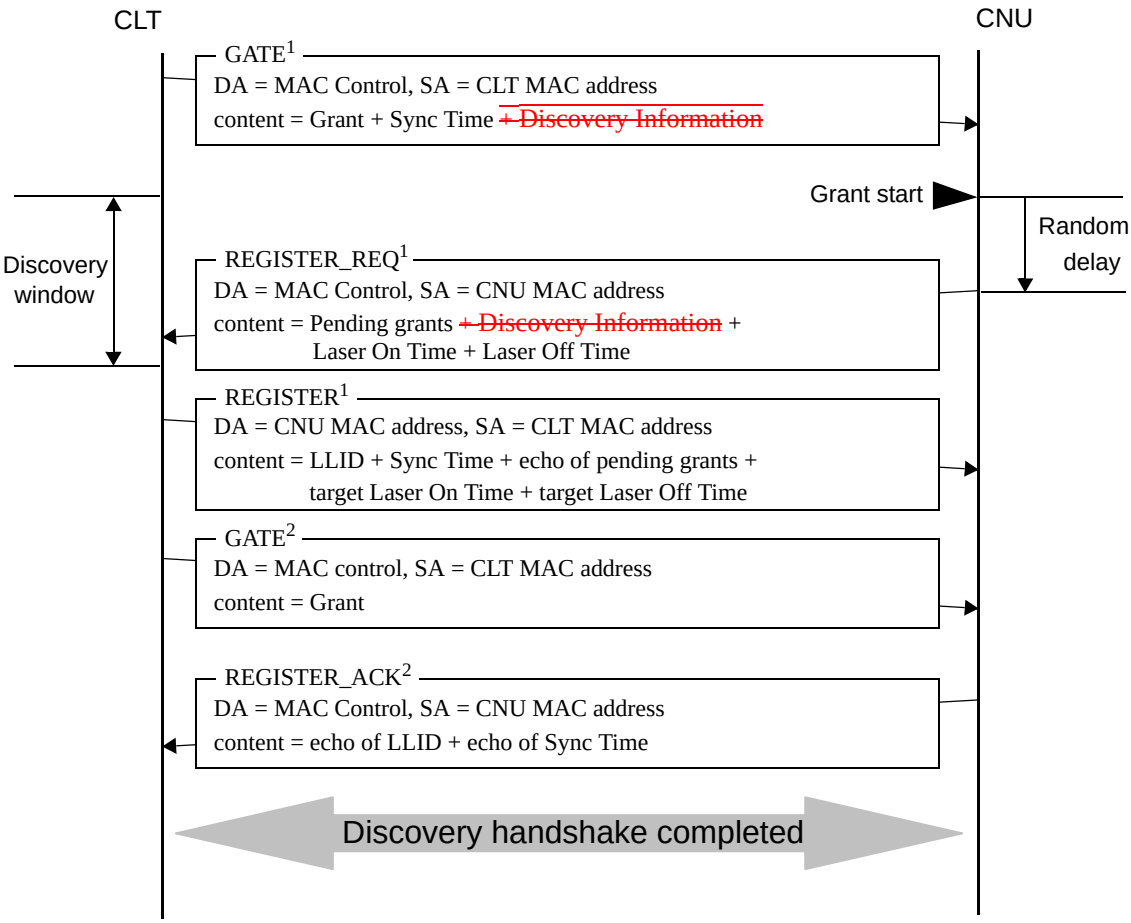
~~Note that even though a compliant CNU is not prohibited from supporting more than one data rate in any transmission channel, it is expected that a single supported data rate for upstream and downstream channel is indicated in the Discovery Information field. Moreover,~~ in order to assure maximum utilization of the upstream channel and to decrease the required size of the guard band between individual data bursts, the registering CNU notifies the CLT of the RF on/off times, by setting appropriate values in the RF On Time and RF Off Time fields, where both values are expressed in the units of time_quanta.

Upon receipt of a valid REGISTER_REQ MPCPDU, the CLT registers the CNU, allocating and assigning a new port identity (LLID), and bonding a corresponding MAC to the LLID.

The next step in the process is for the CLT to transmit a REGISTER MPCPDU to the newly discovered CNU, which contains the CNU's LLID, and the CLT's required synchronization time. Moreover, the CLT echoes the maximum number of pending grants. The CLT also sends the target value of RF on time and RF off time, which may be different than RF on time and RF off time delivered by the CNU in the REGISTER_REQ MPCPDU.

The CLT now has enough information to schedule the CNU for access to the CCDN and transmits a standard GATE message allowing the CNU to transmit a REGISTER_ACK. Upon receipt of the REGISTER_ACK, the discovery process for that CNU is complete, the CNU is registered and normal message traffic can begin. It is the responsibility of Layer Management to perform the MAC bonding, and start transmission from/to the newly registered CNU. The discovery message exchange is illustrated in Figure 102–1.

There may exist situations when the CLT requires that an CNU go through the discovery sequence again and reregister. Similarly, there may be situations where an CNU needs to inform the CLT of its desire to deregister. The CNU can then reregister by going through the discovery sequence. For the CLT, the REGISTER message may indicate a value, Reregister or Deregister, that if either is specified forces the receiving CNU into reregistering. For the CNU, the REGISTER_REQ message contains the Deregister bit that signifies to the CLT that this CNU should be deregistered.




[1] Messages sent on a broadcast channel
[2] Messages sent on unicast channels

**Figure 102–1—Discovery handshake message exchange**

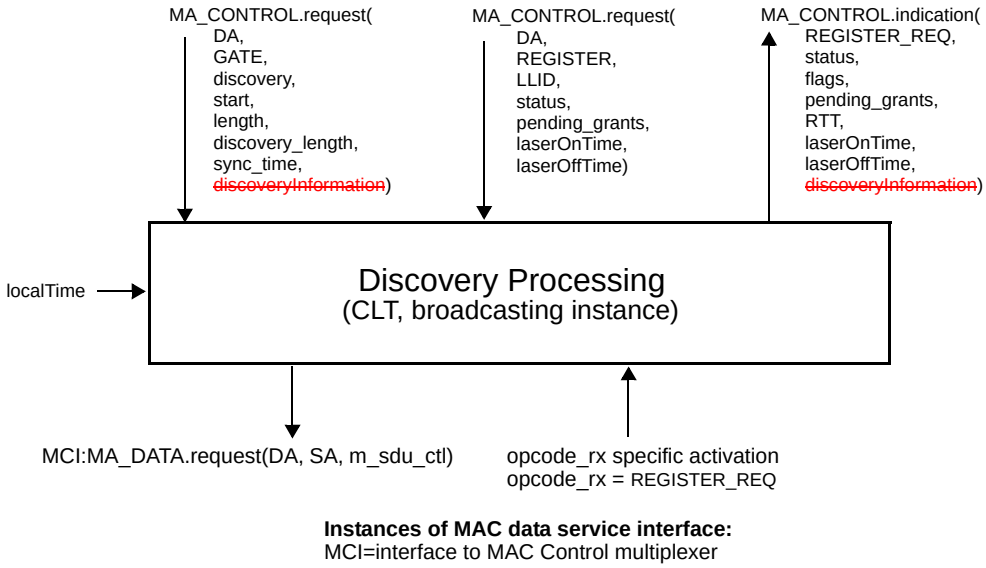



**Figure 102–2—Discovery Processing service interfaces (CLT, unicasting instance)**

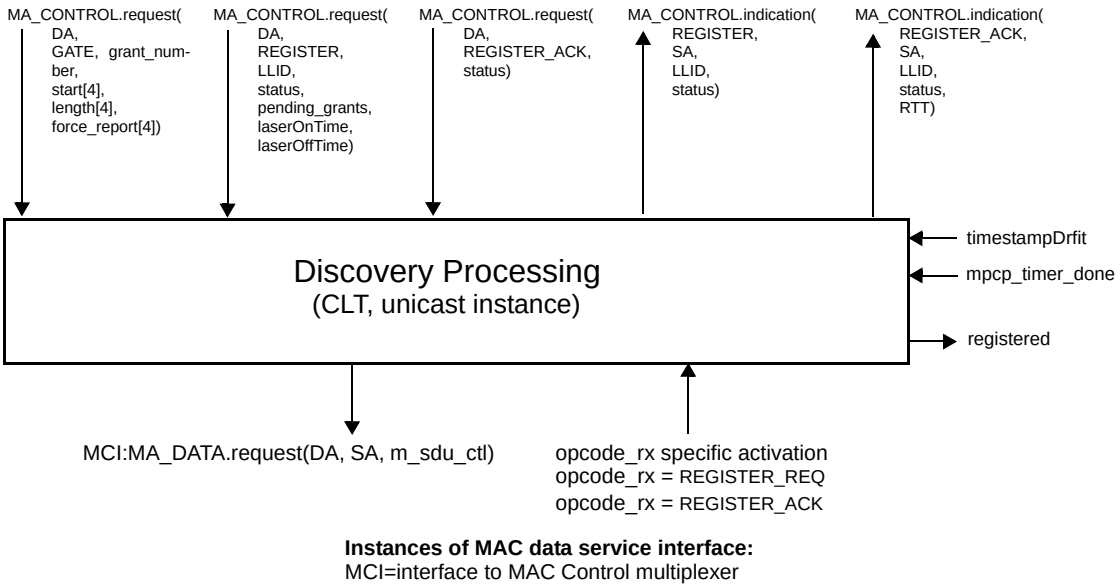



**Figure 102–3—Discovery Processing service interfaces (CNU)**

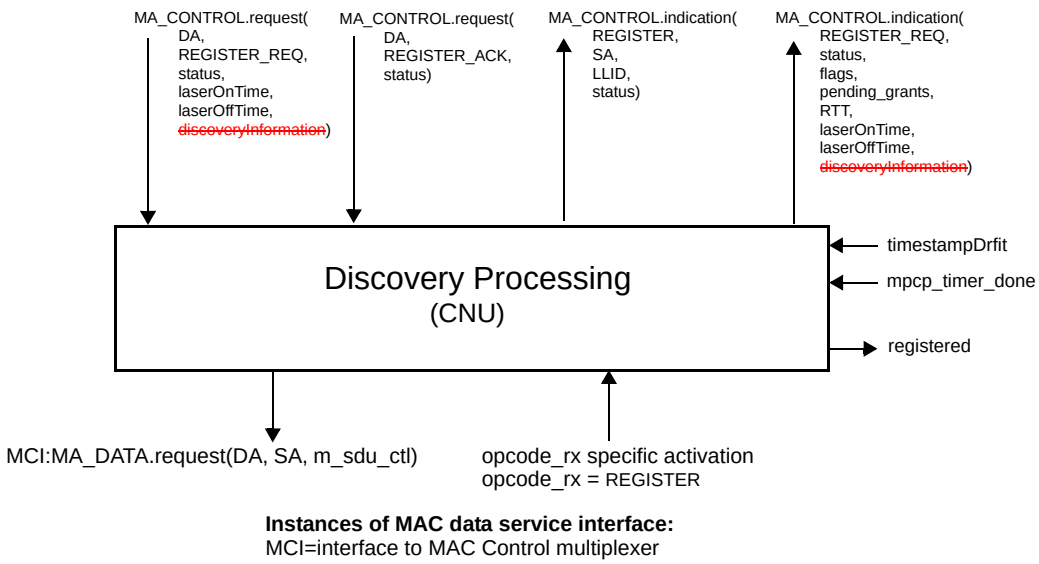



**Figure 102–4—Discovery Processing service interfaces (CNU)**

### 102.3.3.1 Constants

rfOffTimeCapability

TYPE: 8 bit unsigned

This constant represents the time required to terminate the RF, in units of time_quantum. While the default value corresponds to a maximum allowed Toff (as specified in Table 102–8 and Table 102–9), implementations may set it to the actual value time period required for turning off the PMD, as specified in X.7.14.

VALUE: 0x20 (512 ns, default value)

*EDITORS NOTE: the cross reference to X.7.14 cannot be resolved as the paragraph does not exist.*

rfOnTimeCapability

TYPE: 8 bit unsigned

This constant represents the time required to initialize the RF, in units of time_quantum. While the default value corresponds to a maximum allowed Ton (as specified in Table 102–8 and Table 102–9), implementations may set it to the actual value time period required for turning on the PMD, as specified in X.7.14.

VALUE: 0x20 (512 ns, default value)

*EDITORS NOTE: the cross reference to X.7.14 cannot be resolved as the paragraph does not exist.*

### 102.3.3.2 Variables

BEGIN

This variable is defined in 102.2.2.3.

data_rx

This variable is defined in 102.2.2.3.

data_tx

This variable is defined in 102.2.2.3.

grantEndTime
TYPE: 32 bit unsigned
This variable holds the time at which the CLT expects the CNU grant to complete. Failure of a REGISTER_ACK message from an CNU to arrive at the CLT before grantEndTime is a fatal error in the discovery process, and causes registration to fail for the specified CNU, who may then retry to register. The value of grantEndTime is measured in units of time_quantum.

insideDiscoveryWindow
TYPE: Boolean
This variable holds the current status of the discovery window. It is set to true when the discovery window opens, and is set to false when the discovery window closes.

rfOffTime
TYPE: 8 bit unsigned
This variable holds the time required to terminate the RF. It counts in time_quanta units the time period required for turning off the PMD, as specified by the value of Toff in X.7.14.
VALUE: rfOffTimeCapability (default value)

*EDITORS NOTE: the cross reference to X.7.14 cannot be resolved as the paragraph does not exist.*

rfOnTime
TYPE: 8 bit unsigned
This variable holds the time required to initiate the PMD. It counts in time_quanta units the time period required for turning on the PMD, as specified by the value of Ton in X.7.14.
VALUE: rfOnTimeCapability (default value)

*EDITORS NOTE: the cross reference to X.7.14 cannot be resolved as the paragraph does not exist.*

localTime
This variable is defined in 102.2.2.2.

m_sdu_ctl
This variable is defined in 102.2.2.3.

opcode_rx
This variable is defined in 102.2.2.3.

pendingGrants
TYPE: 16 bit unsigned
This variable holds the maximum number of pending grants that an CNU is able to queue.

registered
TYPE: Boolean
This variable holds the current result of the Discovery Process. It is set to true once the discovery process is complete and registration is acknowledged.

syncTime
TYPE: 16 bit unsigned
This variable holds the time required to stabilize the receiver at the CLT. It counts time_quanta units from the point where transmission output is stable to the point where synchronization has been achieved. The value of syncTime includes gain adjustment interval (Treceiver_settling), clock synchronization interval (Tcdr), and code?roup alignment interval (Tcode_group_align), as specified in X.7.14. The CLT conveys the value of syncTime to CNUs in Discovery GATE and REGISTER messages. During the synchronization time a CNU sends synchronization pat-

tern (SP, see Y.3.2.5.2) followed by burst delimiter pattern (BURST_DELIMITER, see Y.3.2.5.2).

*EDITORS NOTE: the cross reference to X.7.14 cannot be resolved as the paragraph does not exist.*

timestampDrift
This variable is defined in 102.2.2.3.

### 102.3.3.3 Functions

None.

### 102.3.3.4 Timers

discovery_window_size_timer
This timer is used to wait for the event signaling the end of the discovery window.
VALUE: The timer value is set dynamically based on the parameters received in a DISCOVERY GATE message.

mpcp_timer
This timer is used to measure the arrival rate of MPCP frames in the link. Failure to receive frames is considered a fatal fault and leads to deregistration.

### 102.3.3.5 Messages

MA_DATA.indication(DA, SA, m_sdu, receiveStatus)
This service primitive is defined in 2.3.2.

MA_DATA.request (DA, SA, m_sdu)
This service primitive is defined in 2.3.2.

MA_CONTROL.request(DA, GATE, discovery, start, length, discovery_length, sync_time~~, discoveryInformation~~)
This service primitive is used by the MAC Control client at the CLT to initiate the Discovery Process. This primitive takes the following parameters:

| | |
|---|---|
| DA: | Multicast or unicast MAC address. |
| GATE: | Opcode for GATE MPCPDU as defined in Table 31A?. |
| discovery: | Flag specifying that the given GATE message is to be used for discovery only. |
| start: | Start time of the discovery window. |
| length: | Length of the grant given for discovery. |
| discovery_length: | Length of the discovery window process. |
| sync_time: | The time interval required to stabilize the receiver at the CLT. |
| ~~discoveryInformation:~~ | ~~This parameter represents the Discovery Information field in GATE MPCPDU as specified in 102.3.6.1, defining the speed(s) the CLT is capable of receiving and speed(s) at which the discovery window is opened for.~~ |

MA_CONTROL.request(DA, GATE, grant_number, start[4], length[4], force_report[4])
This service primitive is used by the MAC Control client at the CLT to issue the GATE message to an CNU and to issue local grants for downstream transmission in TDD mode. This primitive takes the following parameters:

| | |
|---|---|
| DA: | Multicast MAC Control address as defined in Annex 31B. |

GATE: Opcode for GATE MPCPDU as defined in Table 31A?.

grant_number: Number of grants issued with this GATE message. The number of grants ranges from 0 to 4.

start[4]: Start times of the individual grants. Only the first grant_number elements of the array are used.

length[4]: Lengths of the individual grants. Only the first grant_number elements of the array are used.

force_report[4]: Flags indicating whether a REPORT message should be generated in the corresponding grant. Only the first grant_number elements of the array are used.

MA_CONTROL.request(DA,REGISTER_REQ, status, rfOnTime, rfOffTime~~, discoveryInformation~~)

This service primitive is used by a client at the CNU to request the Discovery Process to perform a registration. This primitive takes the following parameters:

DA: Multicast MAC Control address as defined in Annex 31B.

REGISTER_REQ: opcode for REGISTER_REQ MPCPDU as defined in Table 31A?.

status: This parameter takes on the indication supplied by the flags field in the REGISTER_REQ MPCPDU as defined in Table 102–5.

rfOnTime: This parameter holds the rfOnTime value, expressed in units of time_quanta, as reported by MAC client and specified in 102.3.6.3.

rfOffTime: This parameter holds the rfOffTime value, expressed in units of time_quanta, as reported by MAC client and specified in 102.3.6.3.

~~discoveryInformation: This parameter represents the Discovery Information field, as specified in 102.3.6.3, defining the speed(s) the CNU is capable of transmitting and speed(s) at which the registration attempt is made.~~

MA_CONTROL.indication(REGISTER_REQ, status, flags, pending_grants, RTT, rfOnTime, rfOffTime~~, discoveryInformation~~)

The service primitive is issued by the Discovery Process to notify the client and Layer Management that the registration process is in progress. This primitive takes the following parameters:

REGISTER_REQ: Opcode for REGISTER_REQ MPCPDU as defined in Table 31A?.

status: This parameter holds the values incoming or retry. Value incoming is used at the CLT to signal that a REGISTER_REQ message was received successfully. The value retry is used at the CNU to signal to the client that a registration attempt failed and needs to be repeated.

flags: This parameter holds the contents of the flags field in the REGISTER_REQ message. This parameter holds a valid value only when the primitive is generated by the Discovery Process in the CLT.

pending_grants: This parameter holds the contents of the pending_grants field in the REGISTER_REQ message. This parameter holds a valid value only when the primitive is generated by the Discovery Process in the CLT.

RTT: The measured round trip time to/from the CNU is returned in this parameter. RTT is stated in time_quanta units. This parameter holds a valid value only when the primitive is generated by the Discovery Process in the CLT.

rfOnTime: This parameter holds the contents of the rfOnTime field in the REGISTER_REQ message. This parameter holds a valid value only when the primitive is generated by the Discovery Process in the CLT.

rfOffTime: This parameter holds the contents of the rfOffTime field in the REGISTER_REQ message. This parameter holds a valid value only when the primitive is generated by the Discovery Process in the CLT.

~~discoveryInformation:This parameter holds the contents of the Discovery Information field in the REGISTER_REQ MPCPDU. This parameter holds a valid value only when the primitive is generated by the Discovery process in the CLT.~~

MA_CONTROL.request(DA, REGISTER, LLID, status, pending_grants, rfOnTime, rfOffTime)

The service primitive is used by the MAC Control client at the CLT to initiate acceptance of a CNU. This primitive takes the following parameters:

DA: Unicast MAC address or multicast MAC Control address as defined in Annex 31B.

REGISTER: Opcode for REGISTER MPCPDU as defined in Table 31A?.

LLID: This parameter holds the logical link identification number assigned by the MAC Control client.

status: This parameter takes on the indication supplied by the flags field in the REGISTER MPCPDU as defined in Table 102–7.

pending_grants: This parameters echoes back the pending_grants field that was previously received in the REGISTER_REQ message.

rfOnTime: This parameter carries the target value of RF On Time for the given CNU transmitter. This value may be different than the rfOnTime value carried in the REGISTER_REQ MPCPDU received from the corresponding CNU MAC during Discovery stage.

laserOffTime: This parameter carries the target value of RF Off Time for the given CNU transmitter. This value may be different than the rfOffTime value carried in the REGISTER_REQ MPCPDU received from the corresponding CNU MAC during Discovery stage.

MA_CONTROL.indication(REGISTER, SA, LLID, status)

This service primitive is issued by the Discovery Process at the CLT or an CNU to notify the MAC Control client and Layer Management of the result of the change in registration status. This primitive takes the following parameters:

REGISTER: Opcode for REGISTER MPCPDU as defined in Table 31A?.

SA: This parameter represents the MAC address of the CLT.

LLID: This parameter holds the logical link identification number assigned by the MAC Control client.

status: This parameter holds the value of accepted / denied / deregistered / reregistered.

MA_CONTROL.request(DA, REGISTER_ACK, status)

This service primitive is issued by the MAC Control clients at the CNU and the CLT to acknowledge the registration. This primitive takes the following parameters:

DA: Multicast MAC Control address as defined in Annex 31B.

REGISTER_ACK: Opcode for REGISTER_ACK MPCPDU as defined in Table 31A?.

status: This parameter takes on the indication supplied by the flags field in the REGISTER MPCPDU as defined in Table 102–8.

MA_CONTROL.indication(REGISTER_ACK, SA, LLID, status, RTT)

This service primitive is issued by the Discovery Process at the CLT to notify the client and Layer Management that the registration process has completed. This primitive takes the following parameters:

REGISTER_ACK: Opcode for REGISTER_ACK MPCPDU as defined in Table 31A?.

SA: This parameter represents the MAC address of the reciprocating device (CNU address at the CLT, and CLT address at the CNU).

LLID: This parameter holds the logical link identification number assigned by the MAC Control client.

status: This parameter holds the value of accepted/denied/reset/deregistered.

RTT: The measured round trip time to/from the CNU is returned in this parameter. RTT is stated in time_quanta units. This parameter holds a valid value only when the invoking Discovery Process in the CLT.

Opcode-specific function(opcode)

Functions exported from opcode specific blocks that are invoked on the arrival of a MAC Control message of the appropriate opcode.

### 102.3.3.6 State Diagrams

The Discovery Process in the CLT shall implement the discovery window setup state diagram shown in Table 102–5, request processing state diagram as shown in Table 102–6, register processing state diagram as shown in Table 102–7, and final registration state diagram as shown in Table 102–8. The discovery process in the CNU shall implement the registration state diagram as shown in Table 102–9.

Instantiation of state diagrams as described in Table 102–5, Table 102–6, and Table 102–7 is performed only at the Multipoint MAC Control instances attached to the broadcast LLID (0x7FFE). Instantiation of state diagrams as described in Table 102–8 and Table 102–9 is performed for every Multipoint MAC Control instance, except the instance attached to the broadcast channel.

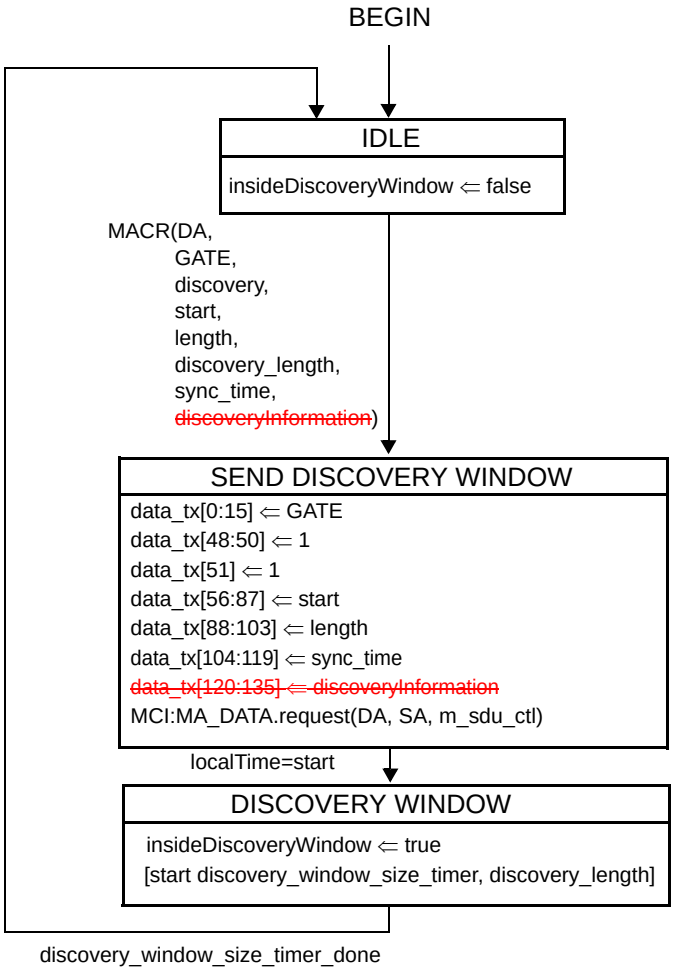



**Figure 102–5—Discovery Processing CLT Window Setup state diagram**

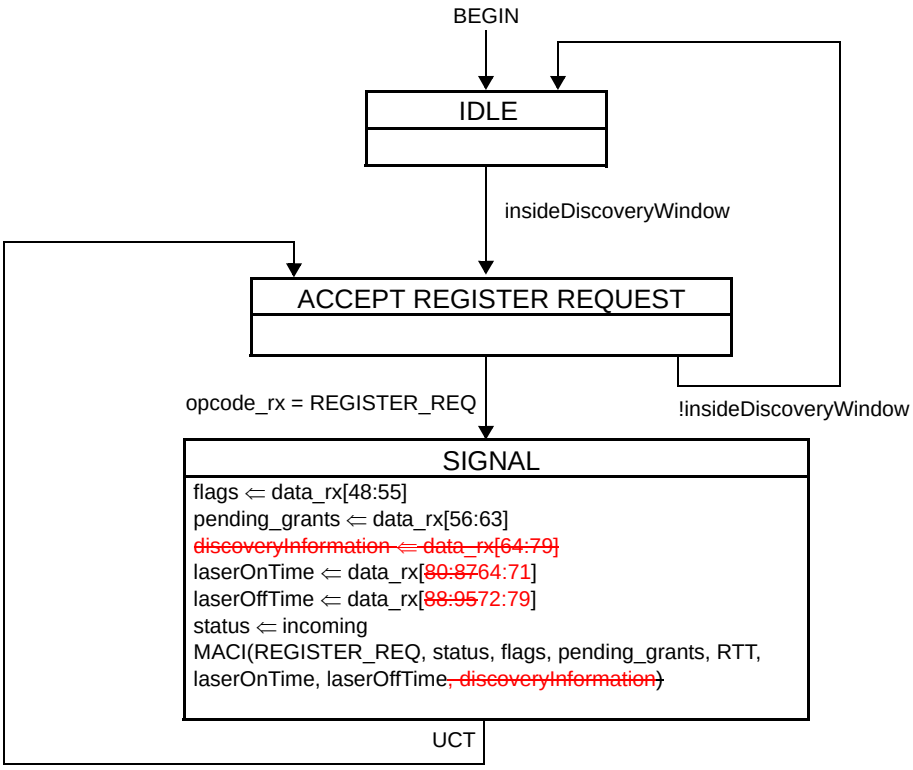



**Figure 102–6—Discovery Processing CLT Process Requests state diagram**

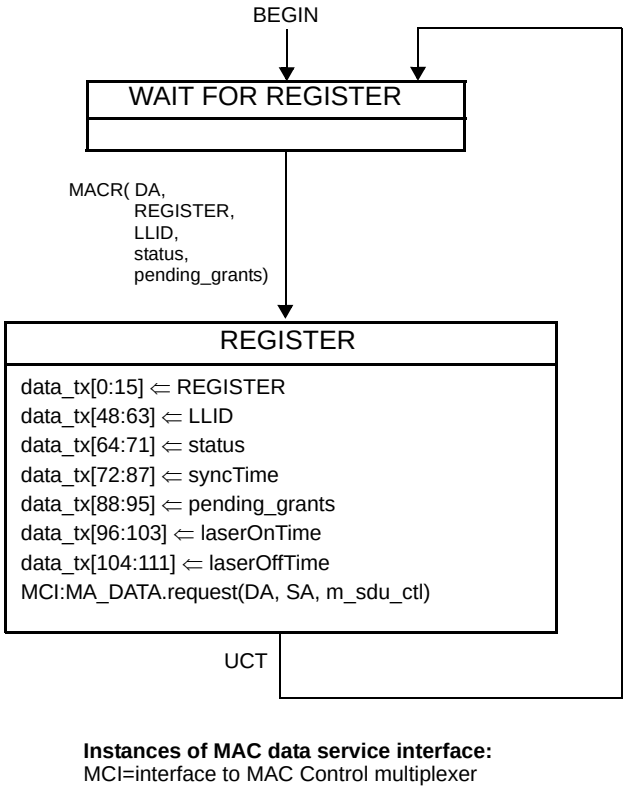
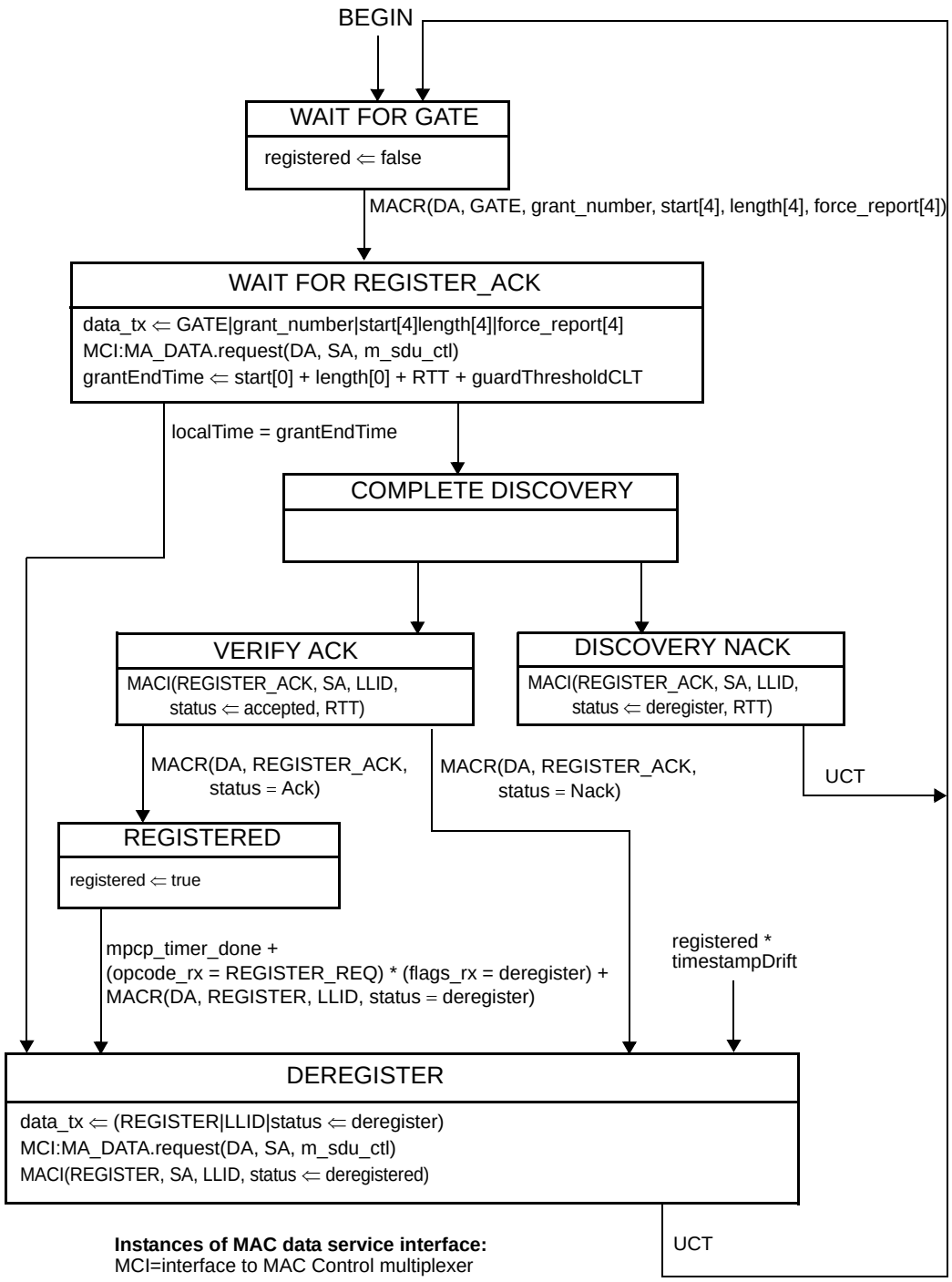



**Figure 102–7—Discovery Processing CLT Register state diagram**




NOTE—The MAC Control Client issues the grant following the REGISTER message, taking the CNU processing delay of REGISTER message into consideration.

**Figure 102–8—Discovery Processing CLT Final Registration state diagram**

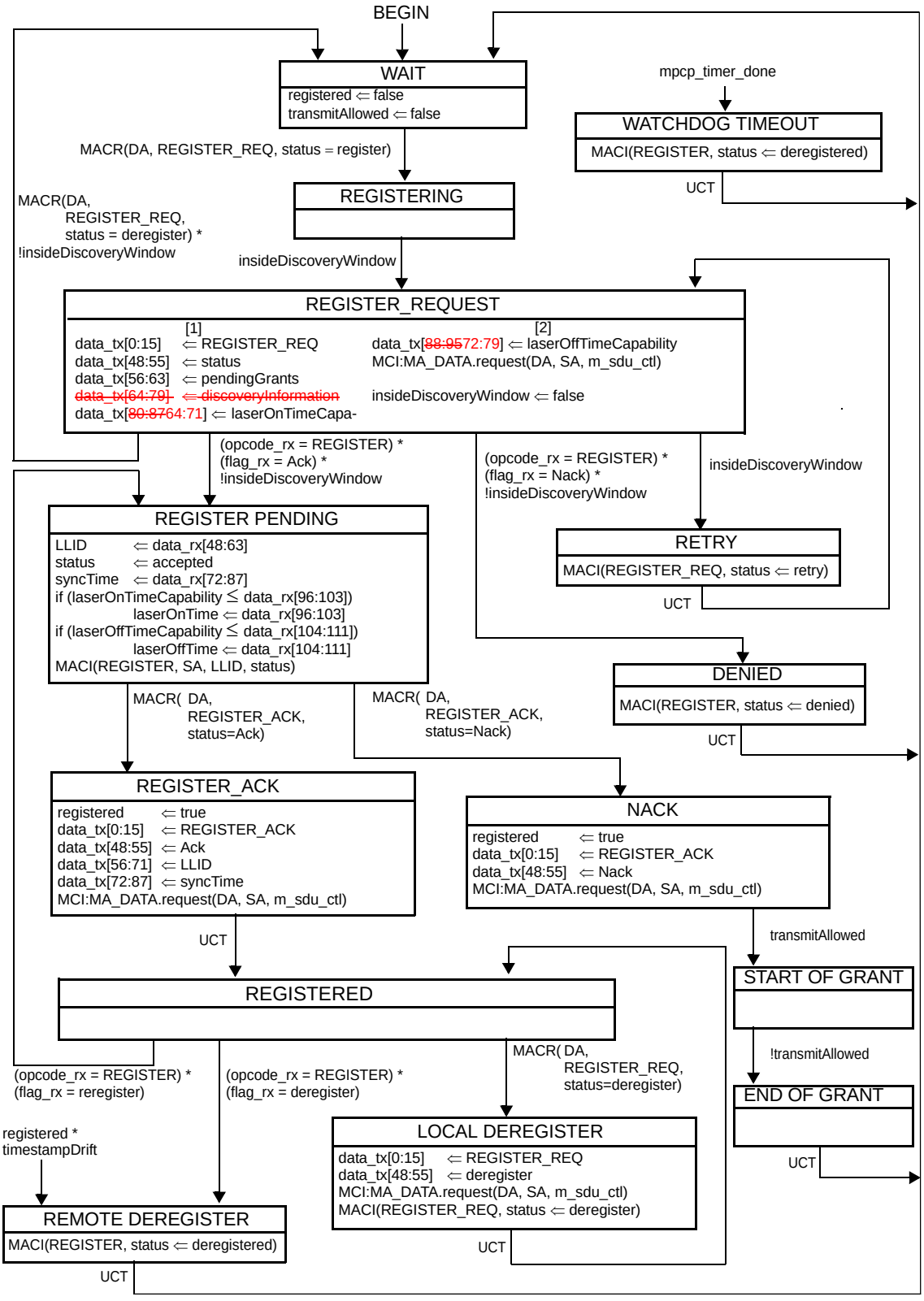



**Figure 102–9—Discovery Processing CNU Registration state diagram**

### 102.3.4 Report Processing

The Report Processing functional block has the responsibility of dealing with queue report generation and termination in the network. Reports are generated by higher layers and passed to the MAC Control sublayer by the MAC Control clients. Status reports are used to signal bandwidth needs as well as for arming the CLT watchdog timer.

Reports shall be generated periodically, even when no request for bandwidth is being made. This keeps a watchdog timer in the CLT from expiring and deregistering the CNU. For proper operation of this mechanism the CLT shall grant the CNU periodically.

The Report Processing functional block, and its MPCP protocol elements are designed for use in conjunction with an IEEE 802.1P capable bridge.

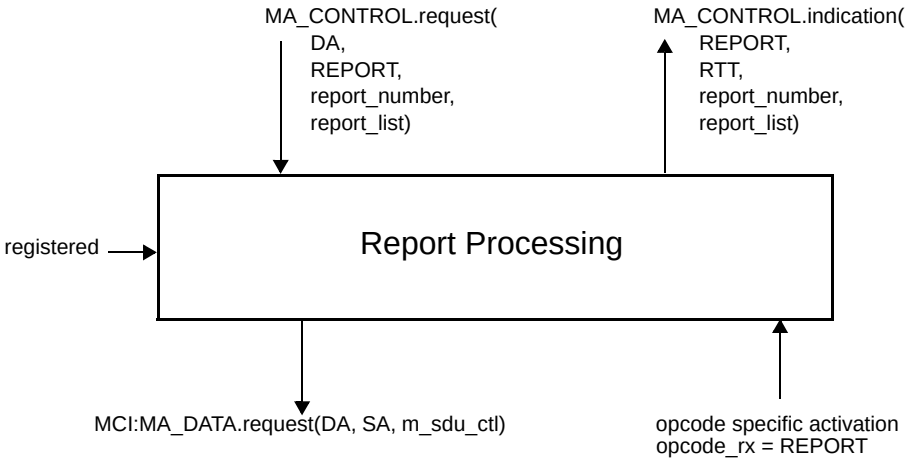



**Figure 102–10—Report Processing service interfaces**

#### 102.3.4.1 Constants

None.

#### 102.3.4.2 Variables

BEGIN
TYPE: Boolean
This variable is used when initiating operation of the functional block state diagram. It is set to true following initialization and every reset.

data_rx
This variable is defined in 102.2.2.3.

data_tx

This variable is defined in 102.2.2.3.

m_sdu_ctl

This variable is defined in 102.2.2.3.

mpcp_timeout

TYPE: 32 bit unsigned

This variable represents the maximum allowed interval of time between two MPCPDU messages. Failure to receive at least one frame within this interval is considered a fatal fault and leads to deregistration. This variable is expressed in units of time_quanta.

VALUE: 0x03B9ACA0 (1 s, default value)

opcode_rx

This variable is defined in 102.2.2.3.

registered

This variable is defined in 102.3.3.2.

report_timeout

TYPE: 32 bit unsigned

This variable represents the maximum allowed interval of time between two REPORT messages generated by the CNU, expressed in units of time_quanta.

VALUE: 0x002FAF08 (50 ms, default value)

#### 102.3.4.3 Functions

None.

#### 102.3.4.4 Timers

report_periodic_timer

CNUs are required to generate REPORT MPCPDUs with a periodicity of less than report_timeout value. This timer counts down time remaining before a forced generation of a REPORT message in an CNU.

mpcp_timer

This timer is defined in 102.3.3.4.

#### 102.3.4.5 Messages

MA_DATA.request (DA, SA, m_sdu)

The service primitive is defined in 2.3.2.

MA_CONTROL.request(DA, REPORT, report_number, report_list)

This service primitive is used by a MAC Control client to request the Report Process at the CNU to transmit a queue status report. This primitive may be called at variable intervals, independently of the granting process, in order to reflect the time varying aspect of the network. This primitive uses the following parameters:

DA: Multicast MAC Control address as defined in Annex 31B.

REPORT: Opcode for REPORT MPCPDU as defined in Table 31A?.

report_number: The number of queue status report sets located in report list. The report_number value ranges from 0 to a maximum of 13.

report_list: The list of queue status reports. A queue status report consists of two fields: valid and status. The parameter valid is a Boolean array of length of 8. The index of an element of this array reflects the numbered priority queue in the IEEE 802.1P nomenclature. An element with the value of '0' or false indicates that the corresponding status field is not present (the length of status field is 0),while '1' or true indicates that the corresponding status field is present (the length of status field is 2 octets). The parameter status is an array of 16 bit unsigned integer values. This array consists only of entries whose corresponding bit in field valid is set to true.

MA_CONTROL.indication(REPORT, RTT, report_number, report_list)

The service primitive is issued by the Report Process at the CLT to notify the MAC Control client and higher layers the queue status of the MPCP link partner. This primitive may be called multiple times, in order to reflect the time?arying aspect of the network. This primitive uses the following parameters:

REPORT: Opcode for REPORT MPCPDU as defined in Table 31A?.

RTT: This parameter holds an updated round trip time value that is recalculated following each REPORT message reception.

report_number: The number of queue status report sets located in report list. The report_number value ranges from 0 to a maximum of 13.

report_list: The list of queue status reports. A queue status report consists of two fields: valid and status. The parameter valid is a Boolean array of length of 8. The index of an element of this array reflects the numbered priority queue in the IEEE 802.1P nomenclature. An element with the value of '0' or false indicates that the corresponding status field is not present (the length of status field is 0),while '1' or true indicates that the corresponding status field is present (the length of status field is 2 octets). The parameter status is an array of 16 bit unsigned integer values. This array consists only of entries whose corresponding bit in field valid is set to true.

Opcode-specific function(opcode)

Functions exported from opcode specific blocks that are invoked on the arrival of a MAC Control message of the appropriate opcode.

### 102.3.4.6 State diagrams

The report process in the CLT shall implement the report processing state diagram as shown in Figure 102–11. The report process in the CNU shall implement the report processing state diagram as shown in Figure 102–12. Instantiation of state diagrams as described is performed for Multipoint MAC Control instances attached to unicast LLIDs only.

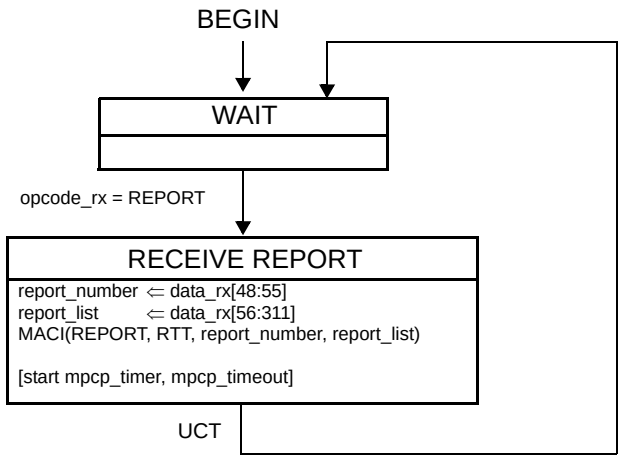
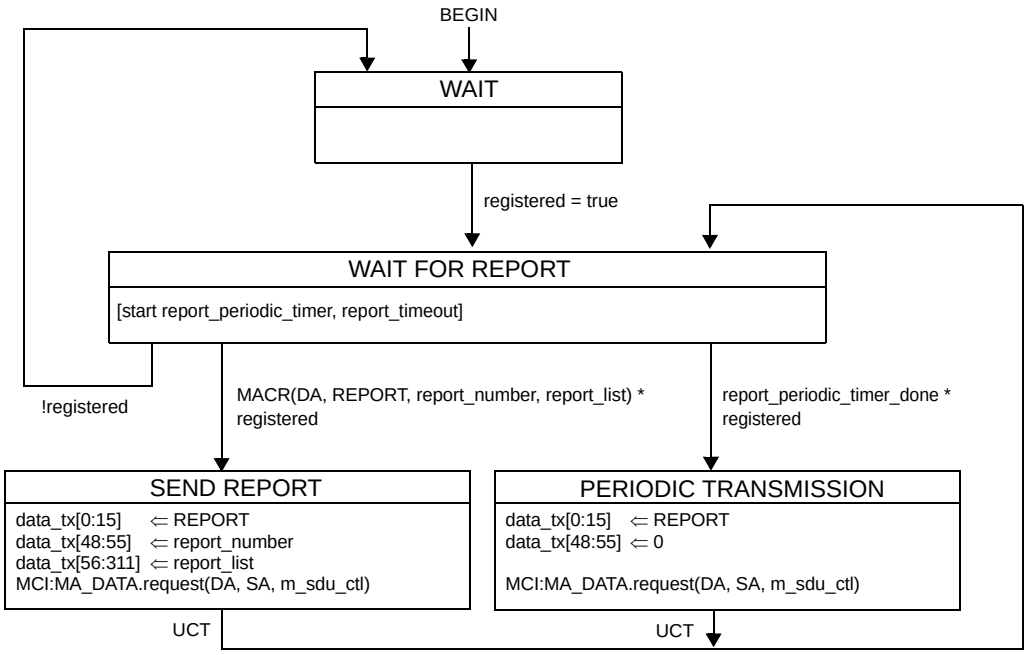



**Figure 102–11—Report Processing state diagram at CLT**




**Instances of MAC data service interface:**
MCI=interface to MAC Control multiplexer

**Figure 102–12—Report Processing state diagram at CNU**

### 102.3.5 Gate Processing

A key concept pervasive in Multipoint MAC Control is the ability to arbitrate a single transmitter out of a plurality of CNUs. The CLT controls an CNU's transmission by the assigning of grants. In addition for TDD mode, the CLT controls the TDD downstream transmission by the assigning of local grants.

The transmitting window of a CNU is indicated in the GATE message where start time and length are specified and the DA field differs from the local address of the CLT. A CNU begins transmission when its localTime counter matches the start_time value indicated in the GATE message. A CNU concludes its transmission with sufficient margin to ensure that the RF transmitter is turned off before the grant length interval has elapsed.

Multiple outstanding grants may be issued to each CNU. The CLT shall not issue more than the maximum supported maximum outstanding grants as advertised by the CNU during registration (see pending grants in 102.3.6.3).

In order to maintain the watchdog timer at the CNU, grants are periodically generated. For this purpose empty GATE messages may be issued periodically.

When registered, the CNU ignores all gate messages where the Discovery flag is set.

The DS transmission window for TDD mode is indicated in the MAC Request for local grant at CLT side where start time and length are specified and the local grant flag is set. In TDD mode, the CLT begins transmission when its localTime counter matches the start_time value indicated in the local grant. The CLT concludes its transmission with sufficient margin to ensure that the RF transmitter is turned off before the local grant length interval has elapsed.

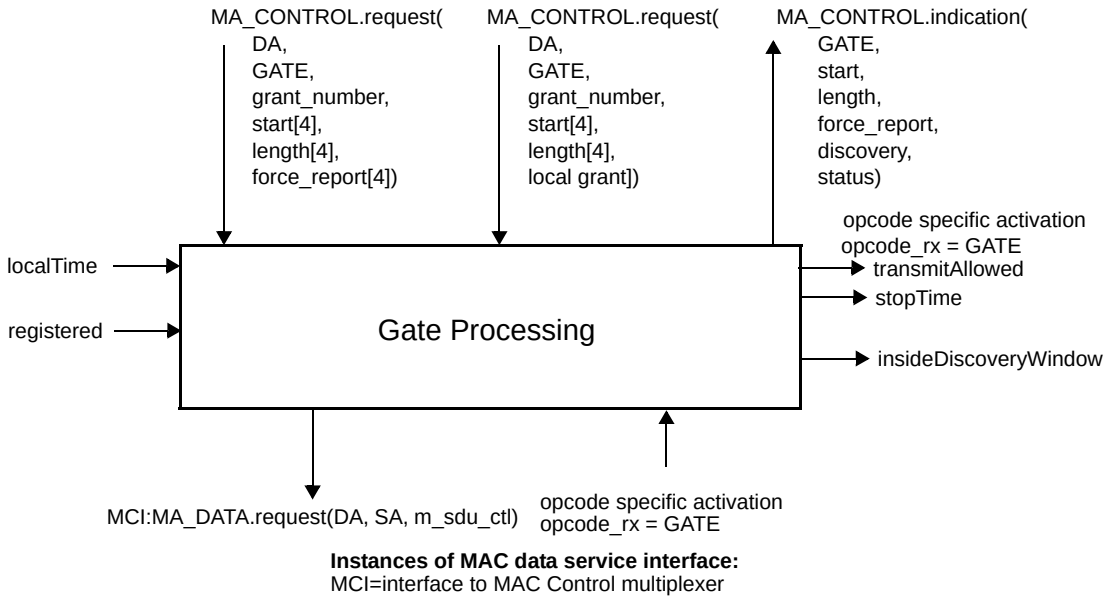



**Figure 102–13—Gate Processing service interface**

### 102.3.5.1 Constants

max_future_grant_time
TYPE: 32 bit unsigned
This constant holds the time limiting the future time horizon for a valid incoming grant.
VALUE: 0x03B9ACA0 (1 s)

min_processing_time
TYPE: 32 bit unsigned
This constant is the time required for the CNU processing time.
VALUE: 0x00000400 (16.384 μs)

minGrantLength
TYPE: 32 bit unsigned
This constant represents the minimum data portion of a grant. The minGrantLength is equal to one FEC codeword (see FEC_CODEWORD_SIZE in 102.2.2.1), less the initial 16 idle octets, expressed in units of time_quanta. The minimum grant length accepted by an CNU is equal to minGrantLength + BurstOverhead (see 102.3.5.2).
VALUE: 12

tqSize
This constant is defined in 102.2.2.1.

### 102.3.5.2 Variables

BEGIN
TYPE: Boolean
This variable is used when initiating operation of the functional block state diagram. It is set to true following initialization and every reset.

BurstOverhead
TYPE: integer
This variable represents the burst overhead and equals the sum of rfOnTime, rfOffTime, syncTime and an additional two time_quanta to account for END_BURST_DELIMITER and two leading IDLE vectors of the payload. This variable is expressed in units of time_quanta.

counter
TYPE: integer
This variable is used as a loop iterator counting the number of incoming grants in a GATE message.

currentGrant
TYPE:
structure
{

| | |
|---|---|
| DA: | 48 bit unsigned, a.k.a MAC address type |
| start | 32 bit unsigned |
| length | 16 bit unsigned |
| force_report | Boolean |
| discovery | Boolean |

}
This variable is used for local storage of a pending grant state during processing. It is dynamically set by the Gate Processing functional block and is not exposed.

The state is a structure field composed of multiple subfields.

data_rx

This variable is defined in 102.2.2.3.

data_tx

This variable is defined in 102.2.2.3.

effectiveLength

TYPE: 32 bit unsigned

This variable is used for temporary storage of a normalized net time value. It holds the net effective length of a grant normalized for elapsed time, and compensated for the periods required to turn the RF on and off, and waiting for receiver lock.

gate_timeout

TYPE: 32 bit unsigned

This variable represents the maximum allowed interval of time between two GATE messages generated by the CLT to the same CNU, expressed in units of time_quanta.

VALUE: 0x002FAF08 (50 ms, default value)

grantList

TYPE: list of elements having the structure define in currentGrant

This variable is used for storage of the list of pending grants. It is dynamically set by the Gate Processing functional block and is not exposed. Each time a grant is received it is added to the list. The list elements are structure fields composed of multiple subfields. The list is indexed by the start subfield in each element for quick searches.

grantStart

This variable is defined in 102.2.2.3.

insideDiscoveryWindow

This variable is defined in 102.3.3.2.

localGrantList

TYPE: list of elements having the structure define in currentGrant for the local grant

This variable is used for storage of the list of pending local grants used in TDD mode. It is dynamically set by the Gate Processing functional block and is not exposed. Each time a local grant is received it is added to the list. The list elements are structure fields composed of multiple subfields. The list is indexed by the start subfield in each element for quick searches.

maxDelay

TYPE: 16 bit unsigned

This variable holds the maximum delay that can be applied by an CNU before sending the REGISTER_REQ MPCPDU. This delay is calculated such that the CNU would have sufficient time to transmit the REGISTER_REQ message and its associated overhead (FEC parity data, end-of-frame sequence, etc.) and terminate the RF before the end of the discovery grant.

m_sdu_ctl

This variable is defined in 102.2.2.3.

nextGrant

TYPE: element having same structure as defined in currentGrant

This variable is used for local storage of a pending grant state during processing. It is dynamically set by the Gate Processing functional block and is not exposed. The content of the variable is the next grant to become active.

nextStopTime
TYPE: 32 bit unsigned
This variable holds the value of the localTime counter corresponding to the end of the next grant.

opcode_rx
This variable is defined in 102.2.2.3.

registered
This variable is defined in 102.3.3.2.

stopTime
This variable is defined in 102.2.2.3.

syncTime
This variable is defined in 102.3.3.2.

transmitAllowed
This variable is defined in 102.2.2.3.

### 102.3.5.3 Functions

empty(list)
This function is use to check whether the list is empty. When there are no elements queued in the list, the function returns true. Otherwise, a value of false is returned.

~~confirmDiscovery(data)~~
~~This function is used to check whether the current Discovery Window is open for the given CNU (TRUE) or not (FALSE). This function returns values as shown in Table 102–1.~~

**Table 102–1—~~Operation of the confirmDiscovery(data) function~~**

| ~~CLT Discovery Information: Discovery Window~~ | | ~~NU Tx capability~~ | | ~~confirmDiscovery(data) returns~~ |
|---|---|---|---|---|
| ~~1G~~ | ~~10G~~ | ~~1G~~ | ~~10G~~ | |
| ~~X~~ | ~~1~~ | ~~0~~ | ~~1~~ | ~~TRUE~~ |
| ~~1~~ | ~~X~~ | ~~1~~ | ~~0~~ | ~~TRUE~~ |
| ~~1~~ | ~~1~~ | ~~1~~ | ~~0~~ | ~~FALSE~~ |
| ~~1~~ | ~~0~~ | ~~0~~ | ~~1~~ | ~~FALSE~~ |
| ~~0~~ | ~~0~~ | ~~X~~ | ~~X~~ | ~~FALSE[a]~~ |

~~[a]These particular values for the Discovery Window fields should not be normally generated by the CLT~~

InsertInOrder(sorted_list, inserted_element)

This function is used to queue an element inside a sorted list. The queuing order is sorted. In the condition that the list is full the element may be discarded. The length of the list is dynamic and it's maximum size equals the value advertised during registration as maximum number of pending grants.

IsBroadcast(grant)

This function is used to check whether its argument represents a broadcast grant, i.e., grant given to multiple CNUs. This is determined by the destination MAC address of the corresponding GATE message. The function returns the value true when MAC address is a global assigned MAC Control address as defined in Annex 31B, and false otherwise.

PeekHead(sorted_list)

This function is used to check the content of a sorted list. It returns the element at the head of the list without dequeuing the element.

Random(r)

This function is used to compute a random integer number uniformly distributed between 0 and r. The randomly generated number is then returned by the function.

RemoveHead(sorted_list)

This function is used to dequeue an element from the head of a sorted list. The return value of the function is the dequeued element.

#### 102.3.5.4 Timers

gntWinTmr

This timer is used to wait for the event signaling the end of a grant window. VALUE: The timer value is dynamically set according to the signaled grant length.

gate_periodic_timer

The CLT is required to generate GATE MPCPDUs with a periodicity of less than gate_timeout value. This timer counts down time remaining before a forced generation of a GATE message in the CLT.

mpcp_timer

This timer is defined in 102.3.3.4.

rndDlyTmr

This timer is used to measure a random delay inside the discovery window. The purpose of the delay is to a priori reduce the probability of transmission overlap during the registration process, and thus lowering the expectancy of registration time in the CCDN.

VALUE: A random value less than the net discovery window size less the REGISTER_REQ MPCPDU frame size less the idle period and RF turn on and off delays less the preamble size less the IFG size. The timer value is set dynamically based on the parameters passed from the client.

#### 102.3.5.5 Messages

MA_DATA.request (DA, SA, m_sdu)

The service primitive is defined in 2.3.2.

MA_CONTROL.request(DA, GATE, grant_number, start[4], length[4], force_report[4])

This service primitive is defined in 102.3.3.5.

MA_CONTROL.request(DA, GATE, grant_number, start[4], length[4], local_grant)

This service primitive is used by the MAC Control client at the CLT to issue the GATE message to an CNU and to issue local grants for downstream transmission in TDD mode. This primitive takes the following parameters:

| | |
|---|---|
| DA: | Multicast MAC Control address as defined in Annex 31B. |
| GATE: | Opcode for GATEMPCPDU as defined in Table 31A–1. |
| grant_number: | Number of grants issued with this GATE message. The number of grants ranges from 0 to 4. |
| start[4]: | Start times of the individual grants. Only the first grant_number elements of the array are used. |
| length[4]: | Lengths of the individual grants. Only the first grant_number elements of the array are used. |
| local_grant: | Flag specifying that the given GATE message is to be used for local grant only |

MA_CONTROL.indication(GATE, start, length, force_report, discovery, status)

This service primitive issued by the Gate Process at the CNU to notify the MAC Control client and higher layers that a grant is pending. This primitive is invoked multiple times when a single GATE message arrives with multiple grants. It is also generated at the start and end of each grant as it becomes active. This primitive uses the following parameters:

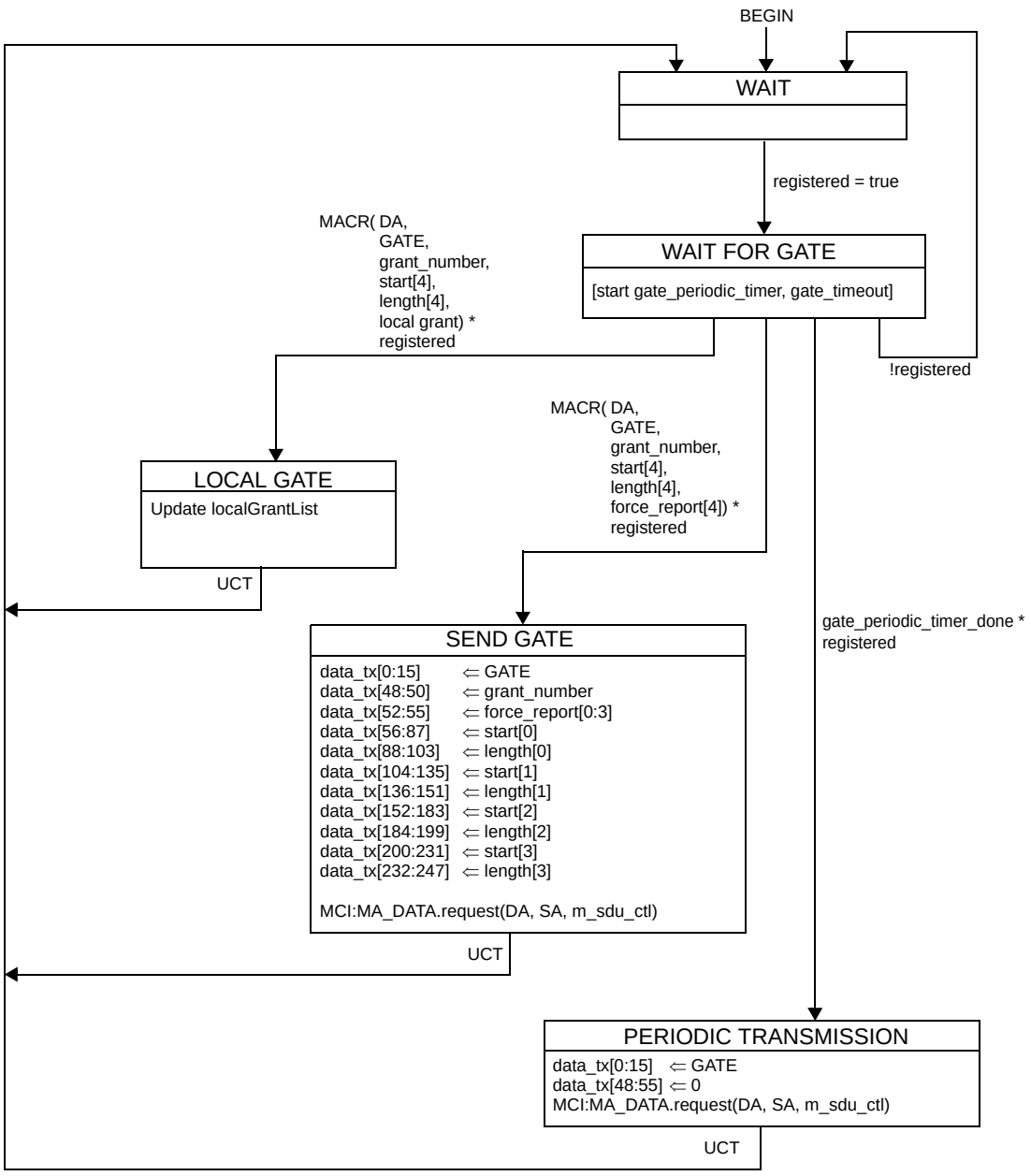
| | |
|---|---|
| GATE: | Opcode for GATE MPCPDU as defined in Table 31A?. |
| start: | start time of the grant. This parameter is not present when the parameter status value is equal to deactive. |
| length: | Length of the grant. This parameter is not present when the parameter status value is equal to deactive. |
| force_report: | Flags indicating whether a REPORT message should be transmitted in this grant. This parameter is not present when the parameter status value is equal to deactive. |
| discovery: | This parameter holds the value true when the grant is to be used for the discovery process, and false otherwise. This parameter is not present when the parameter status value is equal to deactive. |
| status: | This parameter takes the value arrive on grant reception, active when a grant becomes active, and deactive at the end of a grant. |

Opcode-specific function(opcode)

Functions exported from opcode specific blocks that are invoked on the arrival of a MAC Control message of the appropriate opcode.

### 102.3.5.6 State diagrams

The gating process in the CLT shall implement the Gate processing state diagram as shown in Figure 102–14 and, for TDD mode only, in Figure 102–15. The gating process in the CNU shall implement the Gate processing state diagram as shown in Figure 102–16 and Figure 102–17. Instantiation of state diagrams as described is performed for all Multipoint MAC Control instances.




**Instances of MAC data service interface:**
MCI=interface to MAC Control multiplexer

**Figure 102–14—Gate Processing state diagram at CLT**

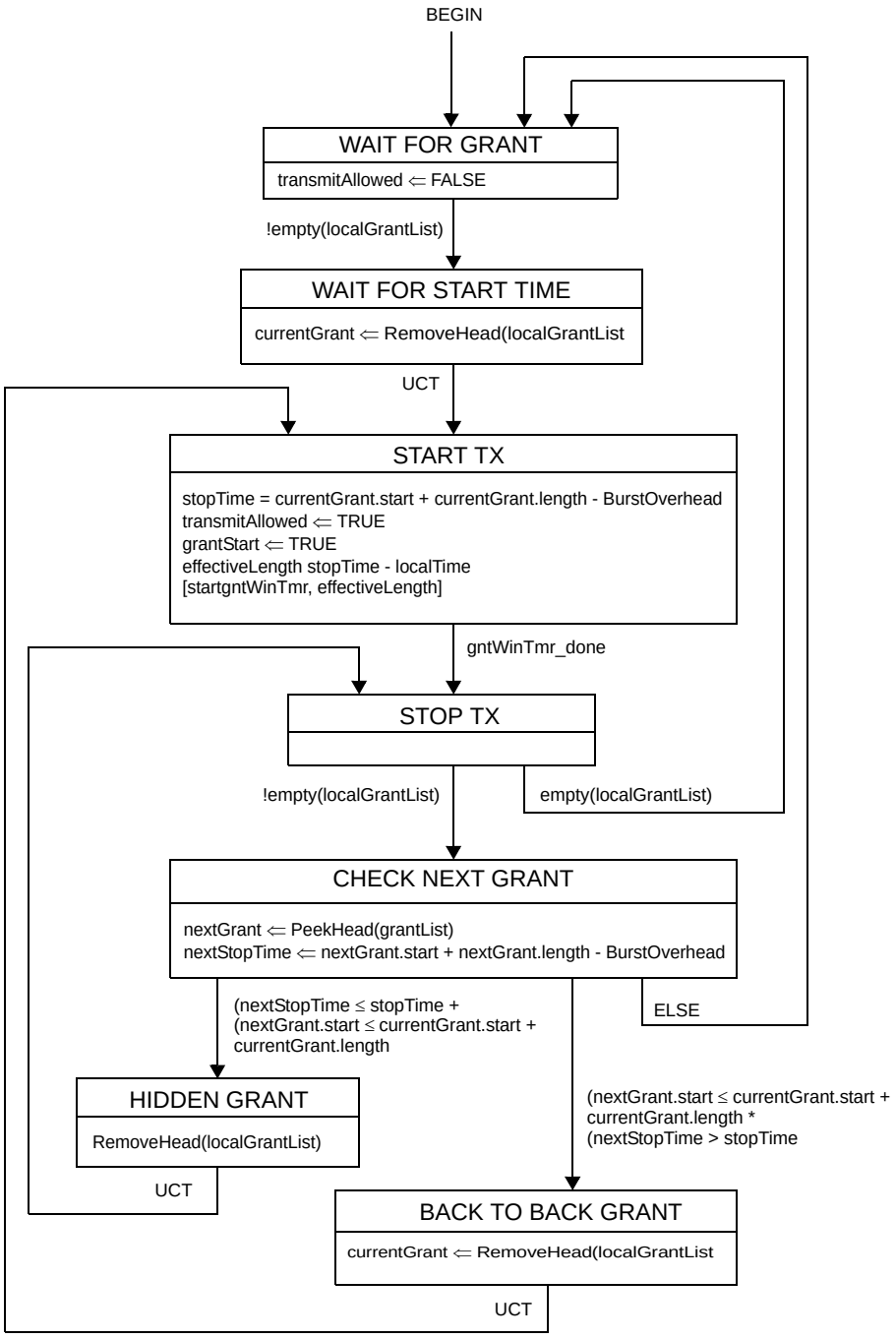



**Figure 102–15—Gate Processing CLT Activation state diagram (TDD mode only)**

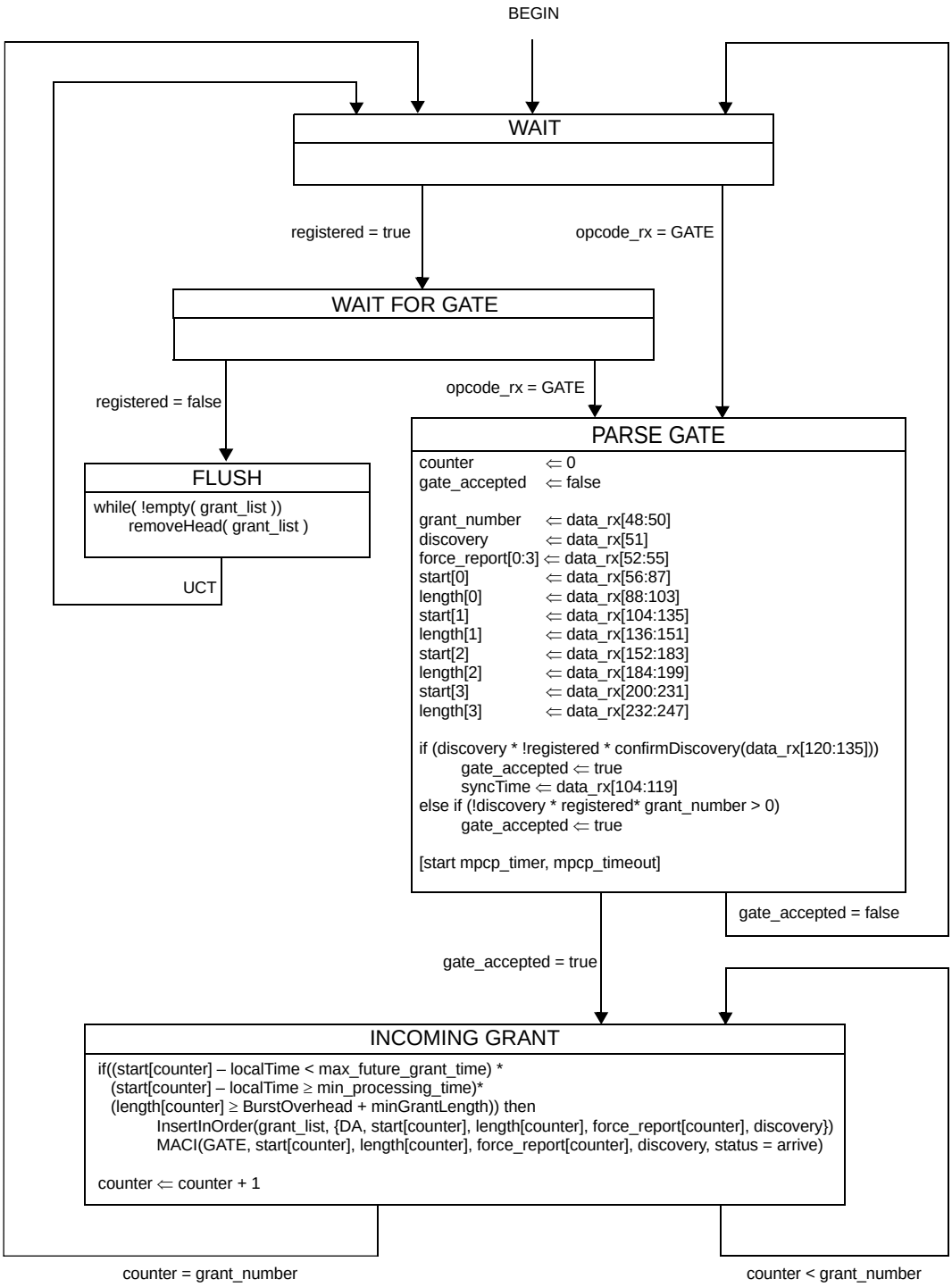



**Figure 102–16—Gate Processing CNU Programing state diagram**

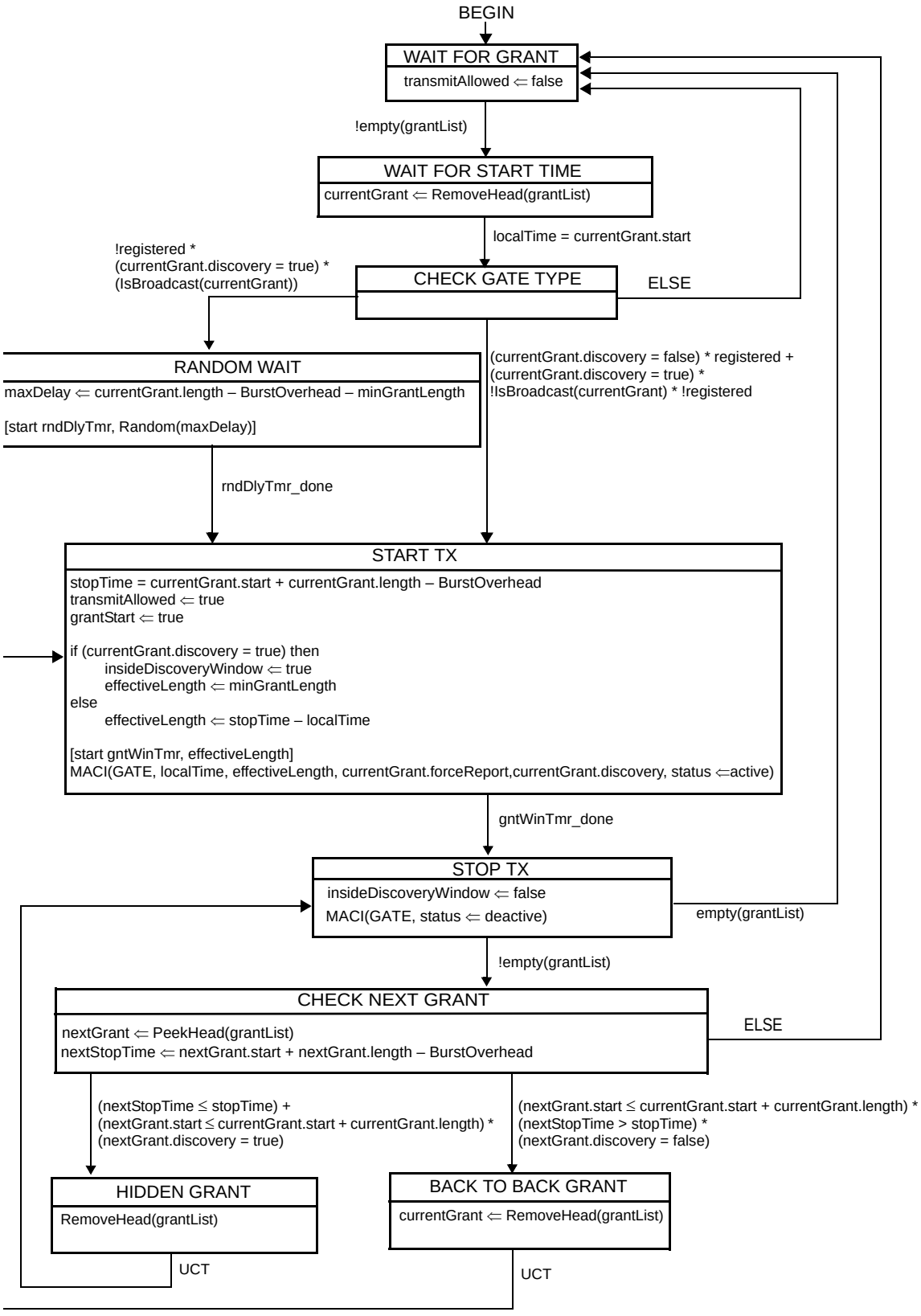



**Figure 102–17—Gate Processing CNU Activation state diagram**

### 102.3.6 MPCPDU structure and encoding

The MPCPDU structure shall be as shown in Figure 102–18, and is further defined in the following definitions:

a) Destination Address (DA). The DA in MPCPDU is the MAC Control Multicast address as specified in the annexes to Clause 31, or the individual MAC address associated with the port to which the MPCPDU is destined.

b) Source Address (SA). The SA in MPCPDU is the individual MAC address associated with the port through which the MPCPDU is transmitted. For MPCPDUs originating at the CLT end, this can be the address any of the individual MACs. These MACs may all share a single unicast address, as explained in 102.1.2.

c) Length/Type. The Length/Type in MPCPDUs carries the MAC_Control_Type field value as specified in 31.4.1.3.

d) Opcode. The opcode identifies the specific MPCPDU being encapsulated. Values are defined in Table 31A?.

e) Timestamp. The timestamp field conveys the content of the localTime register at the time of transmission of the MPCPDUs. This field is 32 bits long and counts time in units of time_quanta.

f) Data/Reserved/PAD. These 40 octets are used for the payload of the MPCPDUs. When not used they would be filled with zeros on transmission, and be ignored on reception.

g) FCS. This field is the Frame Check Sequence, typically generated by the underlying MAC. Based on the MAC instance used to generate the specific MPCPDU, the appropriate LLID shall be generated by the RS.

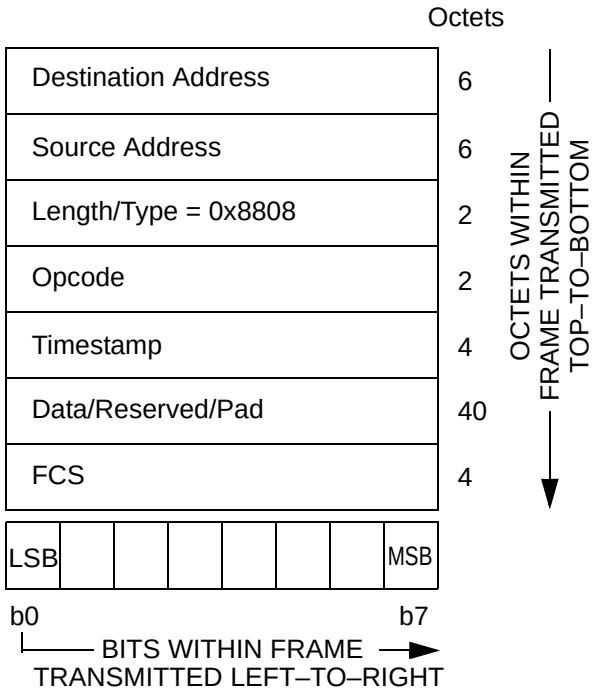



**Figure 102–18—Generic MPCPDU**

### 102.3.6.1 GATE description

The purpose of GATE message is to grant transmission windows to CNUs for both discovery messages and normal transmission. Up to four grants can be included in a single GATE message. The number of grants can also be set to zero for using the GATE message as an MPCP keep alive from CLT to the CNU.

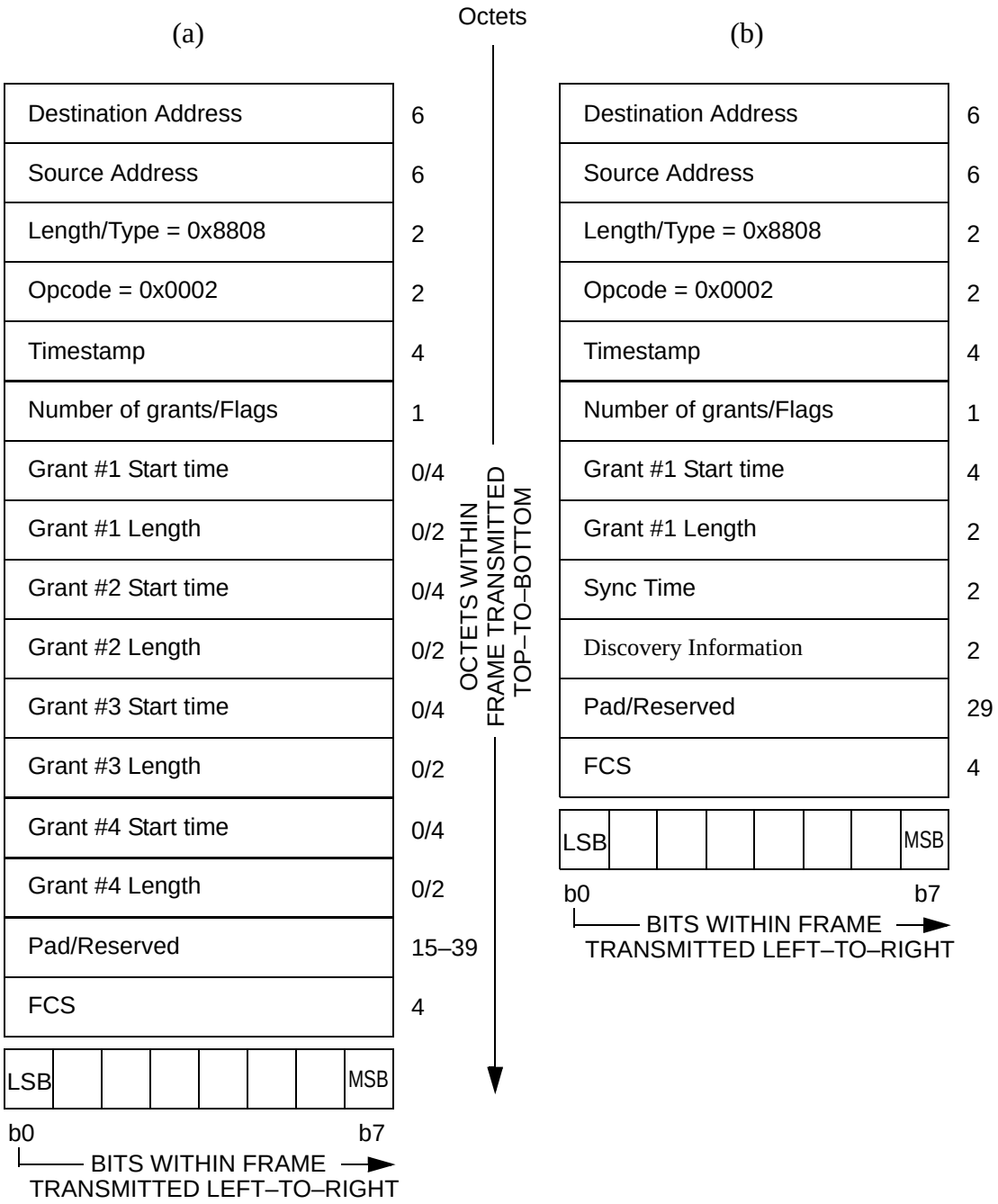



**Figure 102–19—GATE MPCPDU: (a) normal GATE MPCPDU, (b) discovery GATE MPCPDU**

The GATE MPCPDU is an instantiation of the Generic MPCPDU, and is further defined using the following definitions:

a) Opcode. The opcode for the GATE MPCPDU is 0x0002.

b) Flags. This is an 8 bit flag register that holds the following flags: As presented in Table 102–2, the Number of grants field contains the number of grants, composed of valid Length, Start Time pairs in this MPCPDU. This is a number between 0 and 4.

NOTE When Number of grants is set to 0, sole purpose of message is conveying of timestamp to CNU.

The Discovery flag field indicates that the signaled grants would be used for the discovery process, in which case a single grant shall be issued in the GATE message.

The Force Report flag fields ask the CNU to issue a REPORT message related to the corresponding grant number at the corresponding transmission opportunity indicated in this GATE.

**Table 102–2—GATE MPCPDU Number of grants/flags fields**

| Bit | Flag field | Values |
|---|---|---|
| 0 – 2 | Number of grants | 0 – 4 |
| 3 | Discovery | 0 – Normal GATE<br>1 – Discovery GATE |
| 4 | Force Report Grant 1 | 0 – No action required<br>1 – A REPORT frame should be issued at the corresponding transmission opportunity indicated in Grant 1 |
| 5 | Force Report Grant 2 | 0 – No action required<br>1 – A REPORT frame should be issued at the corresponding transmission opportunity indicated in Grant 2 |
| 6 | Force Report Grant 3 | 0 – No action required<br>1 – A REPORT frame should be issued at the corresponding transmission opportunity indicated in Grant 3 |
| 7 | Force Report Grant 4 | 0 – No action required<br>1 – A REPORT frame should be issued at the corresponding transmission opportunity indicated in Grant 4 |

c) Grant #n Start Time. This 32 bit unsigned field represents the start time of the grant. The start time is compared to the local clock, to correlate the start of the grant. Transmitted values shall satisfy the condition Grant #n Start Time < Grant #n+1 Start Time for consecutive grants within the same GATE MPCPDU.

d) Grant #n Length. This 16 bit unsigned field represents the length of the grant. The length is counted in 1 time_quantum increments. There are 4 Grants that are possibly packed into the GATE MPCPDU. The rfOnTime, syncTime, rfOffTime, two initial Idle blocks, FEC parity overhead, and burst terminator sequence (composed of three END_BURST_DELIMITER blocks) are included in and thus consume part of the Grant #n length.

e) Sync Time. This is an unsigned 16 bit value signifying the required synchronization time of the CLT receiver. The CNU calculates the effective grant length by subtracting the syncTime, rfOnTime, rfOffTime and END_BURST_DELIMITER from the grant length it received from the CLT. The

value is counted in 1 time_quantum increments. The advertised value includes synchronization requirement on all receiver elements including PMD, PMA and PCS. This field is present only when the GATE is a discovery GATE, as signaled by the Discovery flag and is not present otherwise.

f) Discovery Information. This ~~is a~~ 16 bit flag register is reserved; all bits are ignored on reception. ~~This field is present only when the GATE is a discovery GATE, as signaled by the Discovery flag and is not present otherwise. Table 102–3 presents the internal structure of the Discovery Information flag field.~~

g) Pad/Reserved. This is an empty field that is transmitted as zeros, and ignored on reception. The size of this field depends on the used Grant #n Length/Start Time entry-pairs as well as the presence of the Sync Time and Discovery Information fields, and varies in length from 15–39 accordingly.

The GATE MPCPDU shall be generated by a MAC Control instance mapped to an active CNU, and as such shall be marked with a unicast type of LLID, except when the MPCPDU is a discovery GATE, as indicated by the Discovery flag being set to true. For the discovery procedure, a MAC Control instance is mapped to all CNUs, and therefore, the discovery GATE MPCPDU is marked with the appropriate broadcast LLID (see 102.3.2.3).

**Table 102–3—~~GATE MPCPDU discovery information fields~~**

| ~~Bit~~ | ~~Flag field~~ | ~~Values~~ |
|---|---|---|
| ~~0~~ | ~~CLT is 1G upstream capable~~ | ~~0 – CLT does not support 1 Gb/s reception~~<br>~~1 – CLT supports 1 Gb/s reception~~ |
| ~~1~~ | ~~CLT is 10G upstream capable~~ | ~~0 – CLT does not support 10 Gb/s reception~~<br>~~1 – CLT supports 10 Gb/s reception~~ |
| ~~2 – 3~~ | ~~Reserved~~ | ~~Ignored on reception~~ |
| ~~4~~ | ~~CLT is opening 1G discovery window~~ | ~~0 – CLT cannot receive 1 Gb/s data in this window~~<br>~~1 – CLT can receive 1 Gb/s data in this window~~ |
| ~~5~~ | ~~CLT is opening 10G discovery window~~ | ~~0 – CLT cannot receive 10 Gb/s data in this window~~<br>~~1 – CLT can receive 10 Gb/s data in this window~~ |
| ~~6 – 15~~ | ~~Reserved~~ | ~~Ignored on reception~~ |

### 102.3.6.2 REPORT description

REPORT messages have several functionalities. Time stamp in each REPORT message is used for round trip (RTT) calculation. In the REPORT messages CNUs indicate the upstream bandwidth needs they request per IEEE 802.1Q priority queue. REPORT messages are also used as keep–alives from CNU to CLT. CNUs issue REPORT messages periodically in order to maintain link health at the CLT as defined in 102.3.4. In addition, the CLT may specifically request a REPORT message.

The REPORT MPCPDU is an instantiation of the Generic MPCPDU, and is further defined using the following definitions:

a) Opcode. The opcode for the REPORT MPCPDU is 0x0003.

b) Number of Queue Sets. This field specifies the number of requests in the REPORT message. A REPORT frame may hold multiple sets of Report bitmap and Queue #n as specified in the Number of Queue Sets field.

c) Report bitmap. This is an 8 bit flag register that indicates which queues are represented in this REPORT MPCPDU, see Table 102–4.

**Table 102–4—REPORT MPCPDU Report bitmap fields**

| Bit | Flag field | Values |
|---|---|---|
| 0 | Queue 0 | 0 – queue 0 report is not present;<br>1 – queue 0 report is present |
| 1 | Queue 1 | 0 – queue 1 report is not present;<br>1 – queue 1 report is present |
| 2 | Queue 2 | 0 – queue 2 report is not present;<br>1 – queue 2report is present |
| 3 | Queue 3 | 0 – queue 3 report is not present;<br>1 – queue 3report is present |
| 4 | Queue 4 | 0 – queue 4 report is not present;<br>1 – queue 4 report is present |
| 5 | Queue 5 | 0 – queue 5 report is not present;<br>1 – queue 5 report is present |
| 6 | Queue 6 | 0 – queue 6 report is not present;<br>1 – queue 6 report is present |
| 7 | Queue 7 | 0 – queue 7 report is not present;<br>1 – queue 7 report is present |

d) Queue #n Report. This value represents the length of queue #n at time of REPORT message generation. The reported length shall be adjusted and rounded up to the nearest time_quantum to account for the necessary inter–frame spacing and preamble. FEC parity overhead is not included in the reported length. The Queue #n Report field is an unsigned 16 bit integer representing the transmission request in units of time_quanta. This field is present only when the corresponding flag in the Report bitmap is set.

e) Pad/Reserved. This is an empty field that is transmitted as zeros, and ignored on reception. The size of this field depends on the used Queue Report entries, and accordingly varies in length from 0 to 39.

The REPORT MPCPDU shall be generated by a MAC Control instance mapped to an active CNU, and as such shall be marked with a unicast type of LLID.

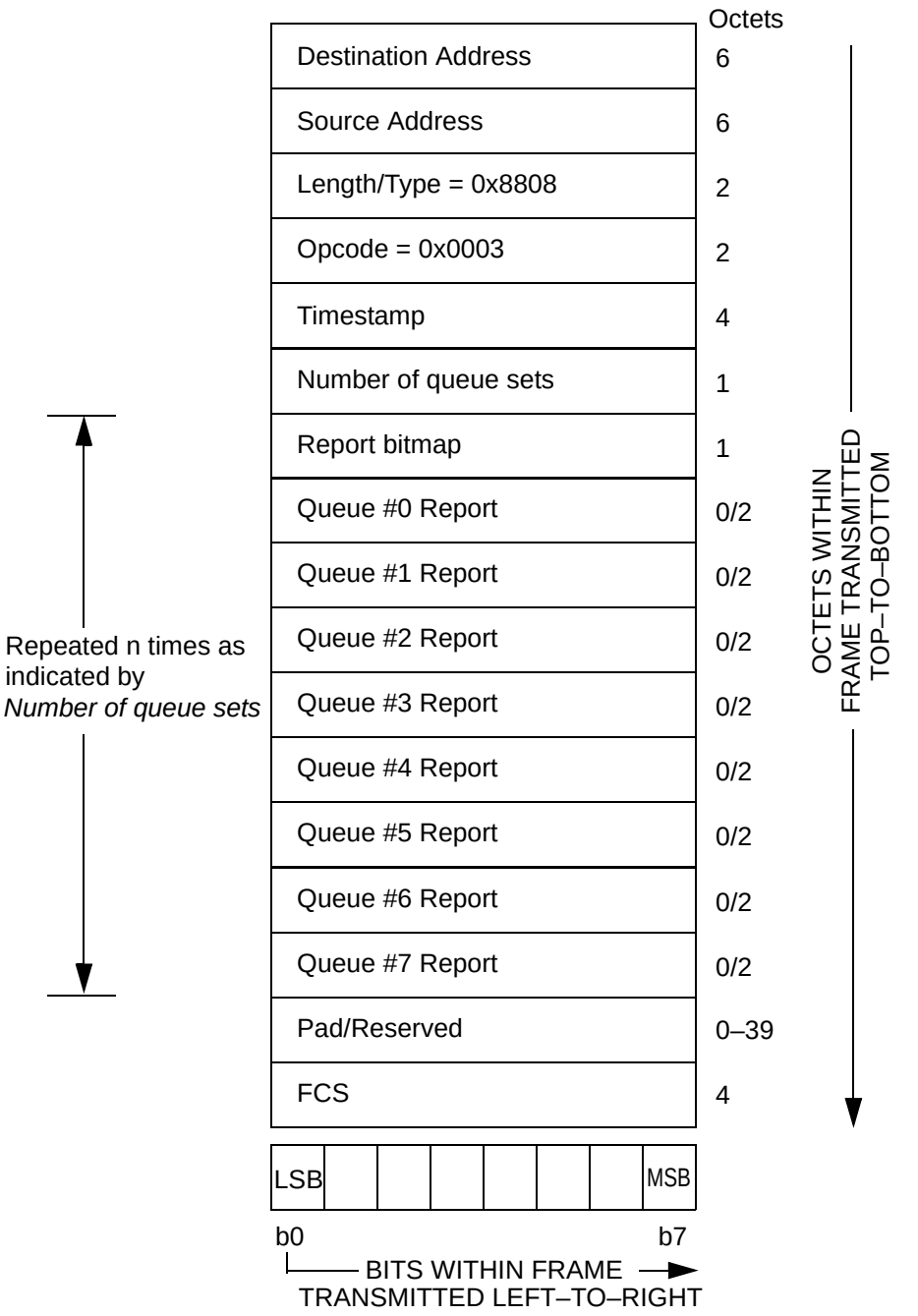



**Figure 102–20—REPORT MPCPDU**

### 102.3.6.3 REGISTER_REQ description

The REGISTER_REQ MPCPDU is an instantiation of the Generic MPCPDU, and is further defined using the following definitions:

a) Opcode. The opcode for the REGISTER_REQ MPCPDU is 0x0004.

b) Flags. This is an 8 bit flag register that indicates special requirements for the registration, as presented in Table 102–5.

**Table 102–5—REGISTER_REQ MPCPDU Flags fields**

| Value | Indication | Comment |
|---|---|---|
| 0 | Reserved | Ignored on reception. |
| 1 | Register | Registration attempt for CNU. |
| 2 | Reserved | Ignored on reception. |
| 3 | Deregister | This is a request to deregister the CNU. Subsequently, the MAC is deallocated and the LLID may be reused. |
| 4 – 255 | Reserved | Ignored on reception. |

c) Pending grants. This is an unsigned 8 bit value signifying the maximum number of future grants the CNU is configured to buffer. The CLT should not grant the CNU more than this maximum number of Pending grants vectors comprised of {start, length, force_report, discovery} into the future.

d) Discovery Information. This ~~is a~~ 16 bit flag register is reserved; all bits are ignored on reception. ~~Table 102–6 presents the structure of the Discovery Information flag.~~

**Table 102–6—~~REGISTER_REQ MPCPDU~~ Discovery Information ~~Fields~~**

| ~~Bit~~ | ~~Flag field~~ | ~~Values~~ |
|---|---|---|
| ~~0~~ | ~~CNU is 1G upstream capable~~ | ~~0 – CNU transmitter is not capable of 1 Gb/s~~<br>~~1 – CNU transmitter is capable of 1 Gb/s~~ |
| ~~1~~ | ~~CNU is 10G upstream capable~~ | ~~0 – CNU transmitter is not capable of 10 Gb/s~~<br>~~1 – CNU transmitter is capable of 10 Gb/s~~ |
| ~~2 – 3~~ | ~~Reserved~~ | ~~Ignored on reception.~~ |
| ~~4~~ | ~~1G registration attempt~~ | ~~0 – 1 Gb/s registration is not attempted~~<br>~~1 – 1 Gb/s registration is attempted~~ |
| ~~5~~ | ~~10G registration attempt~~ | ~~0 – 10 Gb/s registration is not attempted~~<br>~~1 – 10 Gb/s registration is attempted~~ |
| ~~6 – 255~~ | ~~Reserved~~ | ~~Ignored on reception.~~ |

*EDITORS NOTE: should below be rfOnTime and rfOffTime? The text seems to use "RF On Time" and RF Off Time" along with "rfOnTime", "RFOn Time", "rfOffTime", and "RFOff Time". This may be confusing to the reader, recommend using variable simplifying this.*

e) RFOn Time. This field is 1 octet long and carries the RF On Time characteristic for the given CNU transmitter. The value is expressed in the units of time_quanta.

f) RFOff Time. This field is 1 octet long and carries the RF Off Time characteristic for the given CNU transmitter. The value is expressed in the units of time_quanta.

g) Pad/Reserved. This is an empty field that is transmitted as zeros, and ignored on reception.

The REGISTER_REQ MPCPDU shall be generated by a MAC Control instance mapped to an undiscovered CNU, and as such shall be marked with a broadcast type of LLID (102.3.2.3).

*EDITORS NOTE: show Laser On Time and Laser Off Time be replace with RF On time and RF Off Time (or some variant thereof) in the figure below? What about 102-36?*

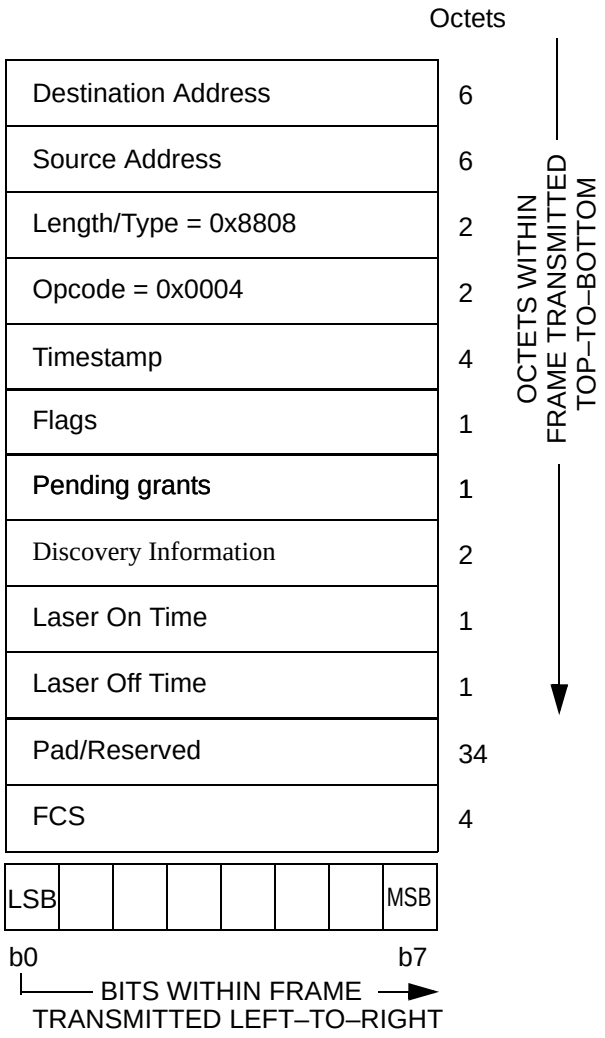



**Figure 102–21—REGISTER_REQ MPCPDU**

### 102.3.6.4 REGISTER description

The REGISTER MPCPDU is an instantiation of the Generic MPCPDU, and is further defined using the following definitions:

a) DA. The destination address used shall be an individual MAC address.

b) Opcode. The opcode for the REGISTER MPCPDU is 0x0005.

c) Assigned Port. This field holds a 16 bit unsigned value reflecting the LLID of the port assigned following registration.

d) Flags. this is an 8 bit flag register that indicates special requirements for the registration, as presented in Table 102–7.

e) Sync Time. This is an unsigned 16 bit value signifying the required synchronization time of the CLT receiver. The CNU calculates the effective grant length by subtracting the syncTime, rfOnTime, rfOffTime, and END_BURST_DELIMITER from the grant length it received from the CLT. The value is counted in 1 time_quantum increments. The advertised value includes synchronization requirement on all receiver elements including PMD, PMA, and PCS.

**Table 102–7—REGISTER MPCPDU Flags field**

| Value | Indication | Comment |
|---|---|---|
| 0 | Reserved | Ignored on reception. |
| 1 | Reregister | The CNU is explicitly asked to re-register. |
| 2 | Deregister | This is a request to deallocate the port and free the LLID. Subsequently, the MAC is deallocated. |
| 3 | Ack | The requested registration is successful. |
| 4 | Nack | he requested registration attempt is denied by the MAC Control Client. |
| 5 – 255 | Reserved | Ignored on reception. |

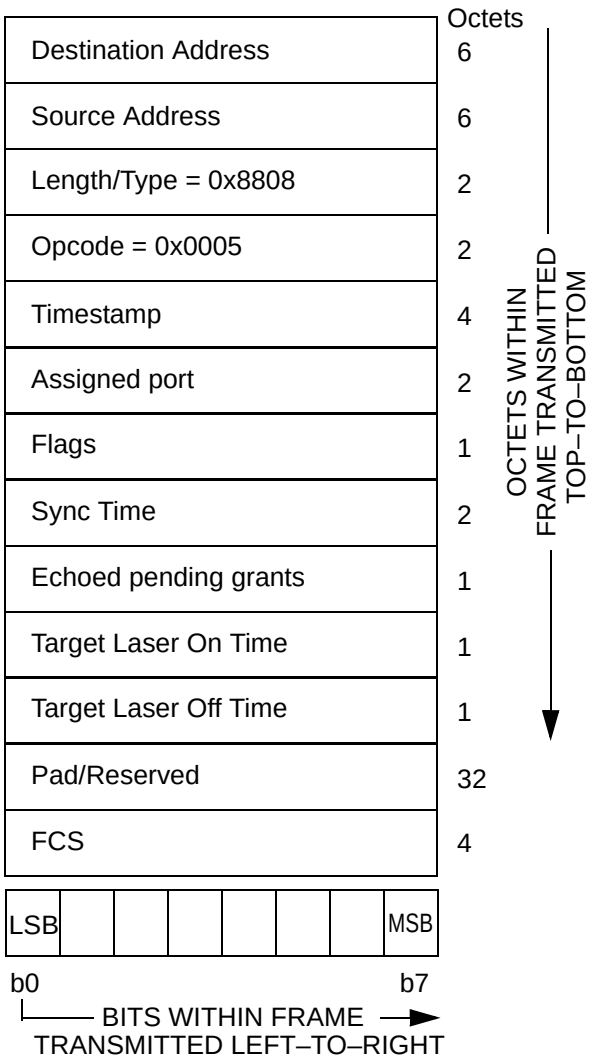



**Figure 102–22—REGISTER MPCPDU**

f) Echoed pending grants. This is an unsigned 8 bit value signifying the number of future grants the CNU may buffer before activating. The CLT should not grant the CNU more than this number of grants into the future.

g) Target RFn Time. This is an unsigned 8 bit value, expressed in the units of time_quanta, signifying the RF On Time for the given CNU transmitter. This value may be different from RF On Time delivered by the CNU in the REGISTER_REQ MPCPDU during the Discovery process. The CNU updates the local rfOnTime variable per state diagram in Figure 102–9.

h) Target RF Off Time. This is an unsigned 8 bit value, expressed in the units of time_quanta, signifying the RF Off Time for the given CNU transmitter. This value may be different from RF Off Time delivered by the CNU in the REGISTER_REQ MPCPDU during the Discovery process. The CNU updates the local rfOffTime variable per state diagram in Figure 102–9.

i) Pad/Reserved. This is an empty field that is transmitted as zeros, and ignored on reception.

The REGISTER MPCPDU shall be generated by a MAC Control instance mapped to all CNUs and such frame is marked by the broadcast LLID (102.3.2.3).

### 102.3.6.5 REGISTER_ACK description

The REGISTER_ACK MPCPDU is an instantiation of the Generic MPCPDU, and is further defined using the following definitions:

a) Opcode. The opcode for the REGISTER_ACK MPCPDU is 0x0006.

b) Flags. This is an 8 bit flag register that indicates special requirements for the registration, as presented in Table 102–8.

c) Echoed assigned port. This field holds a 16 bit unsigned value reflecting the LLID for the port assigned following registration.

d) Echoed Sync Time. This is an unsigned 16 bit value echoing the required synchronization time of the CLT receiver as previously advertised (102.3.6.4).

e) Pad/Reserved. This is an empty field that is transmitted as zeros, and ignored at reception.

**Table 102–8—REGISTER_ACK MPCPDU Flags fields**

| Value | Indication | Comment |
|---|---|---|
| 0 | Nack | The requested registration attempt is denied by the MAC Control Client. |
| 1 | Ack | The registration process is successfully acknowledged. |
| 2 – 255 | Reserved | Ignored on reception. |

The REGISTER_ACK MPCPDU shall be generated by a MAC Control instance mapped to an active CNU, and as such shall be marked with a unicast type of LLID.

## 102.4 Discovery Process in dual-rate systems

*EDITORS NOTE: Sub-clause 102.4 is covered in another comment.*