



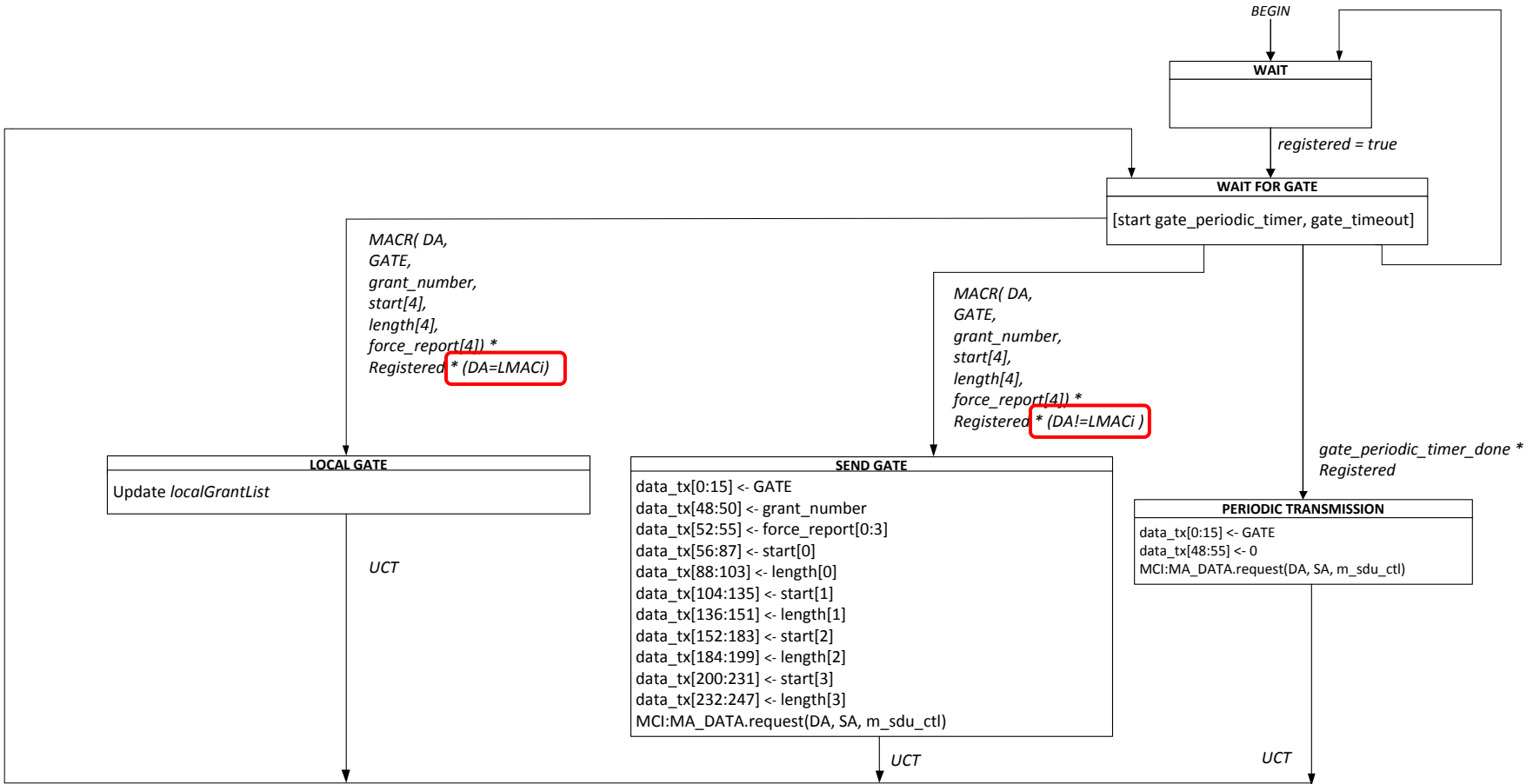
Local Grant Identification for EPOC TDD (baseline proposal amendment)

Andrea Garavaglia and Patrick Stupar
(Qualcomm)

Introduction

- During the last IEEE 802.3bn meeting, a baseline for EPoC TDD was approved by the Task Force (garavaglia_3bn_02a_0313 – [1])
- During the discussion, comments were given about the way the Local Grant introduced by the proposal for the downstream side is identified
 - The current baseline uses the Destination Address field (DA) to discriminate between CLT and CNU – see next slide
- In this presentation we address the comment and revise the proposal

Gate processing in CLT for local (TDD) grant



In the current design, the DA is used to identify a Local Grant from a CNU Grant (see red boxes above).

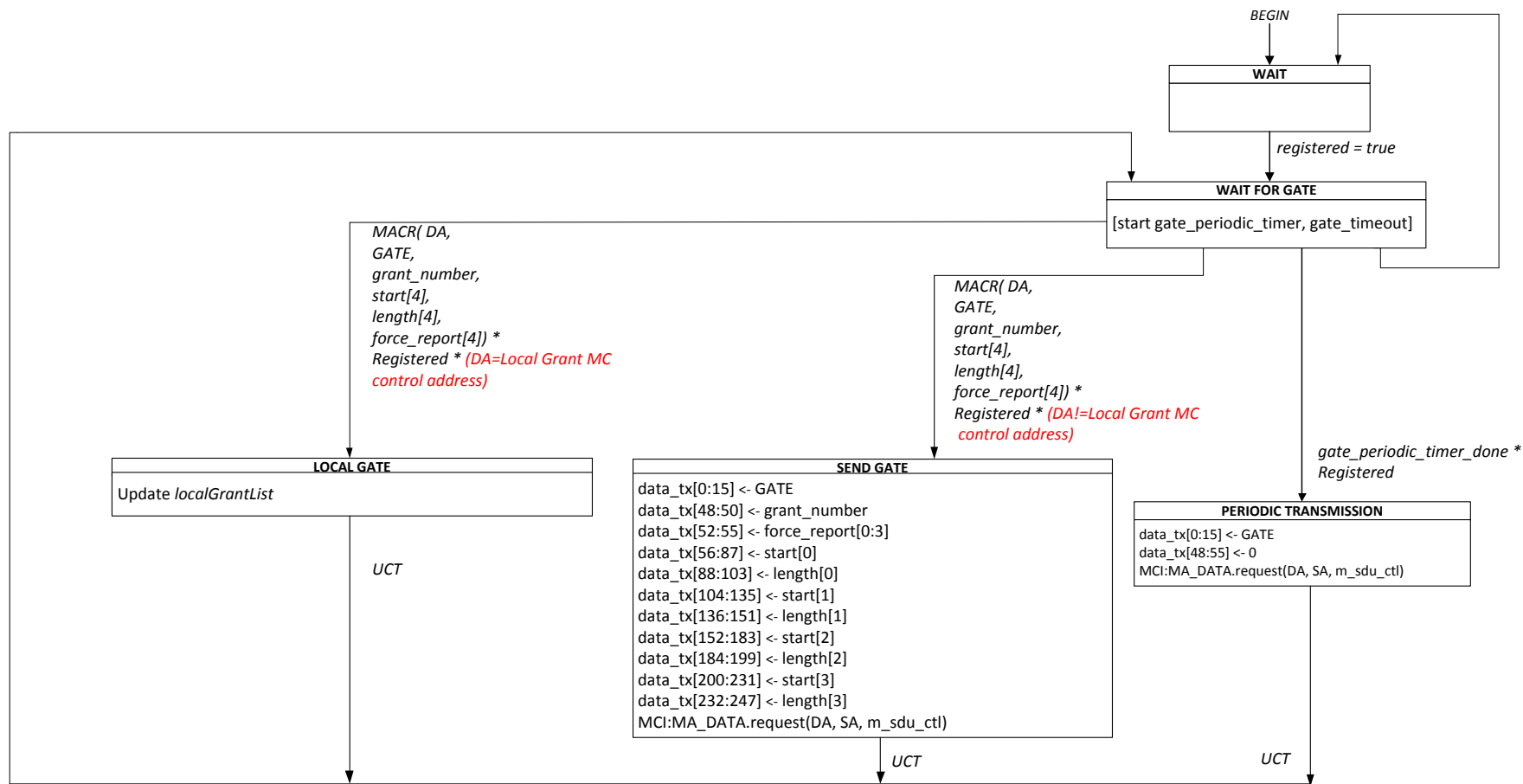
Comment: the Destination Address (DA) for non-discovery GATE in the *MA_CONTROL.request* primitive is defined in IEEE 802.3 Clause 77.3.3.5 as “Multicast MAC Control address as defined in Annex 31B”.

Comment Resolution proposals

Three possible solutions are identified:

1. Definition of a new MAC control address for local grant
 - E.g. similar to the MAC control multicast address 01-80-C2-00-00-01
2. Definition of a local grant (specific) primitive
 - Create a GATE *MA_CONTROL.request* primitive for local grant
3. Extension of an existing primitive to include local grant
 - Extend the existing GATE *MA_CONTROL.request* primitive for local grant, by adding new boolean variable

Solution #1: definition of a new MAC control address



Definition of new MAC control multicast address (e.g. 01-80-C2-00-00-02) and extension of the *MA_CONTROL.request* primitive

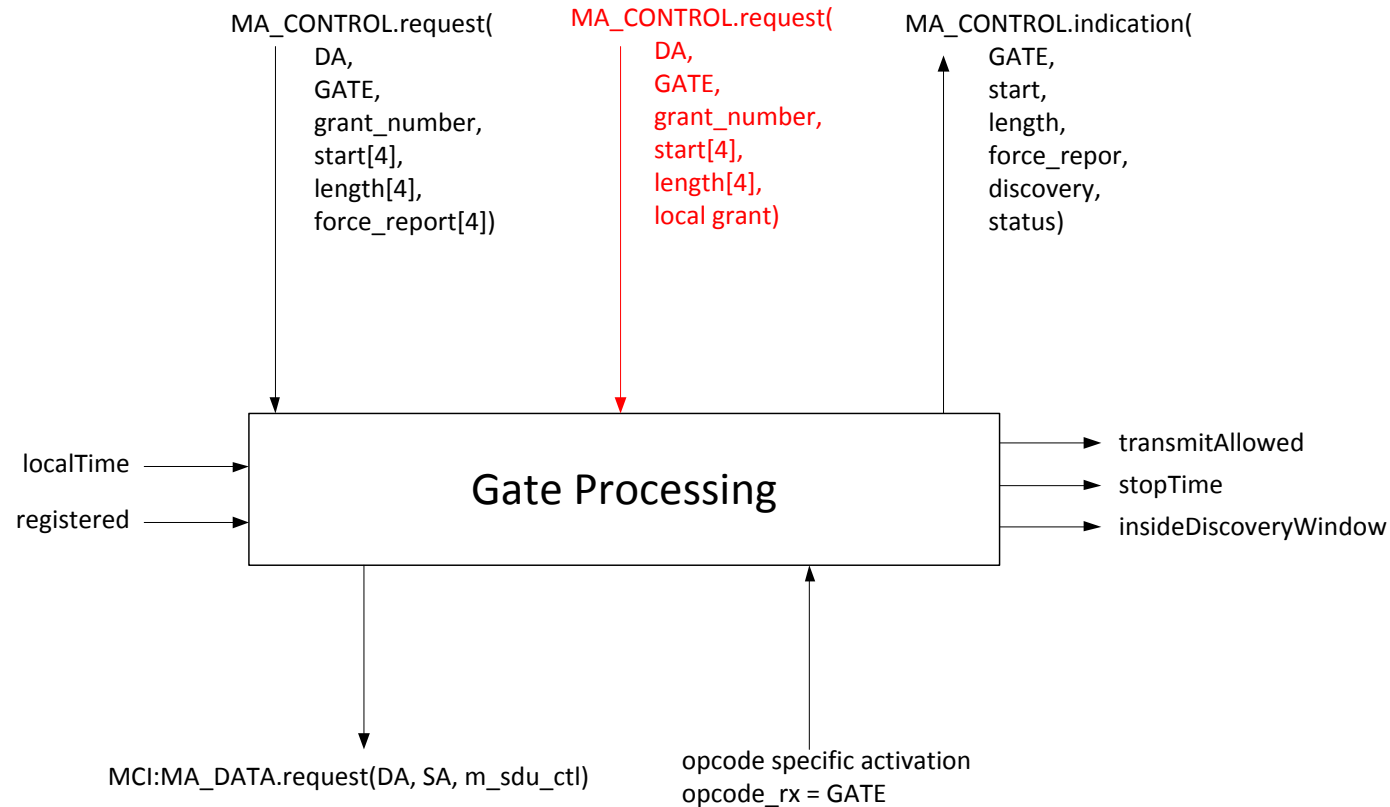
Solution #1: required changes to TDD MPCP clause

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

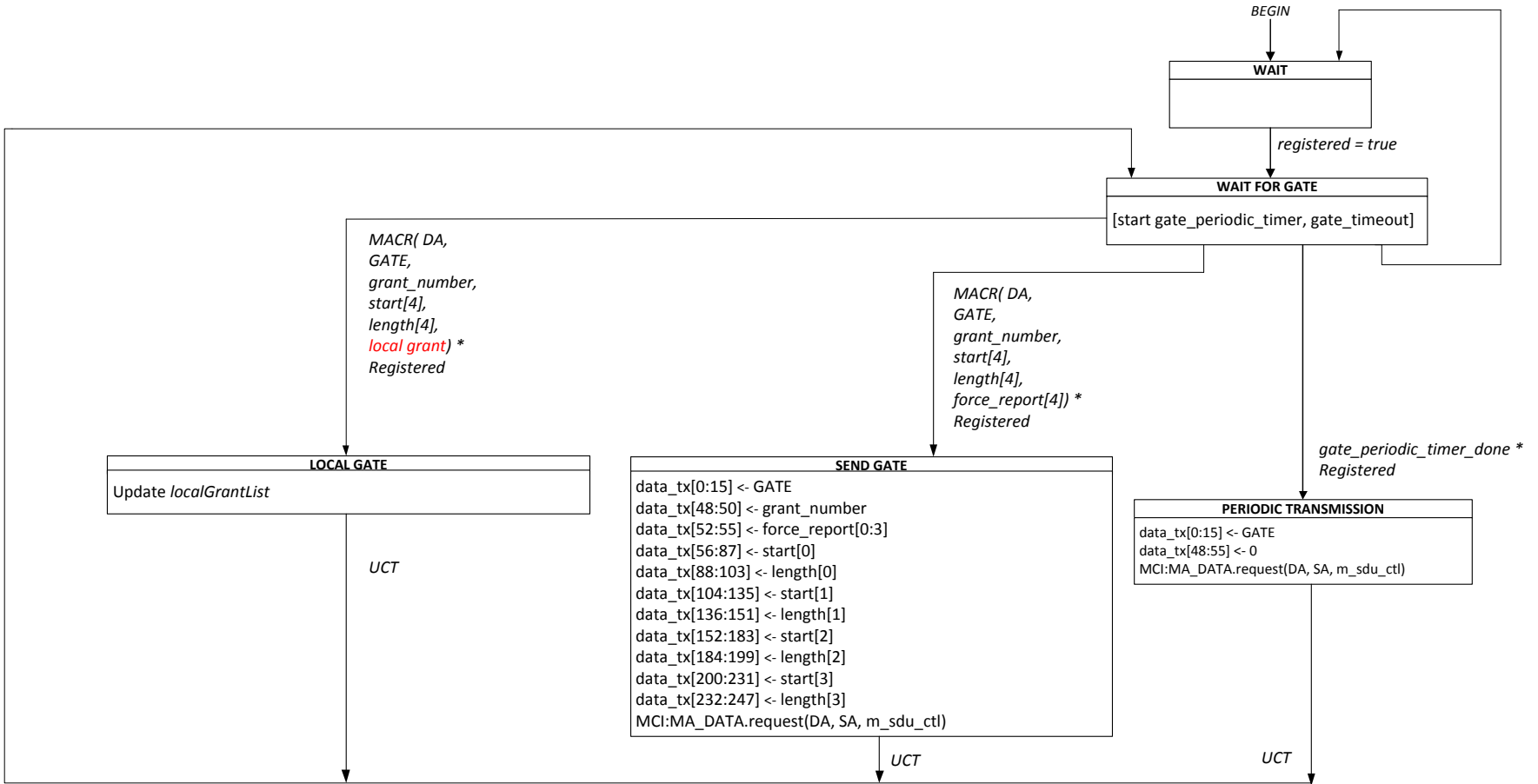
- DA: Multicast MAC Control address as defined in Annex 31B (if the primitive is used to issue the GATE message) or Multicast MAC Control address as defined in XX (if the primitive is used to issue the local grant GATE)
- GATE: Opcode for GATE MPCPDU as defined in Table 31A–1.
- grant_number: Number 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.
- 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.

Solution #2: Definition of a local grant primitive



Two different service primitives entries in Gate processing block (one for local grant GATE and one for GATE messages transmission)

Solution #2: Definition of a local grant primitive (cont.)



GATE primitive for local grant is defined in a similar way as GATE primitive for discovery
Note: the state machine for local grant *MA_CONTROL.request* request primitive may be defined separately

Solution #2: required changes to TDD MPCP clause

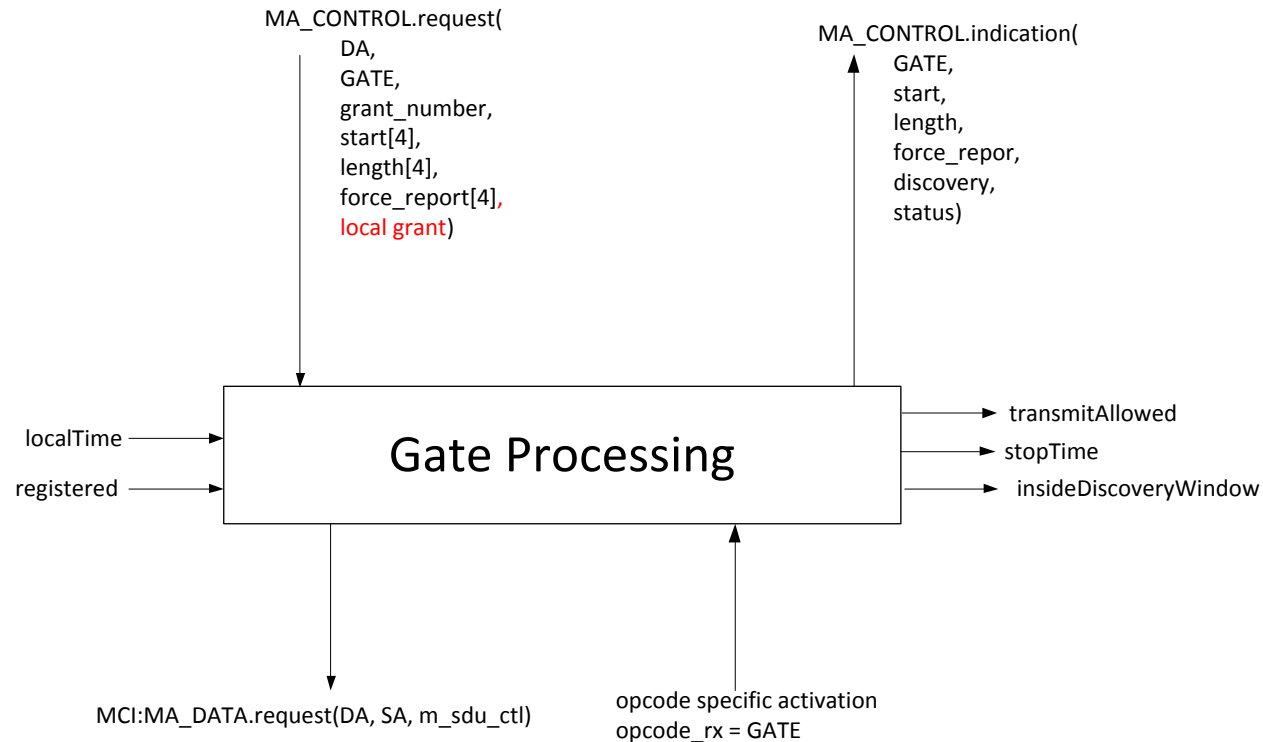
MA_CONTROL.request(DA, GATE, grant_number, start[4], length[4], **local_grant**). This request primitive is used by the MAC Control client at the CLT to issue local grants for downstream transmission in TDD mode.

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–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**

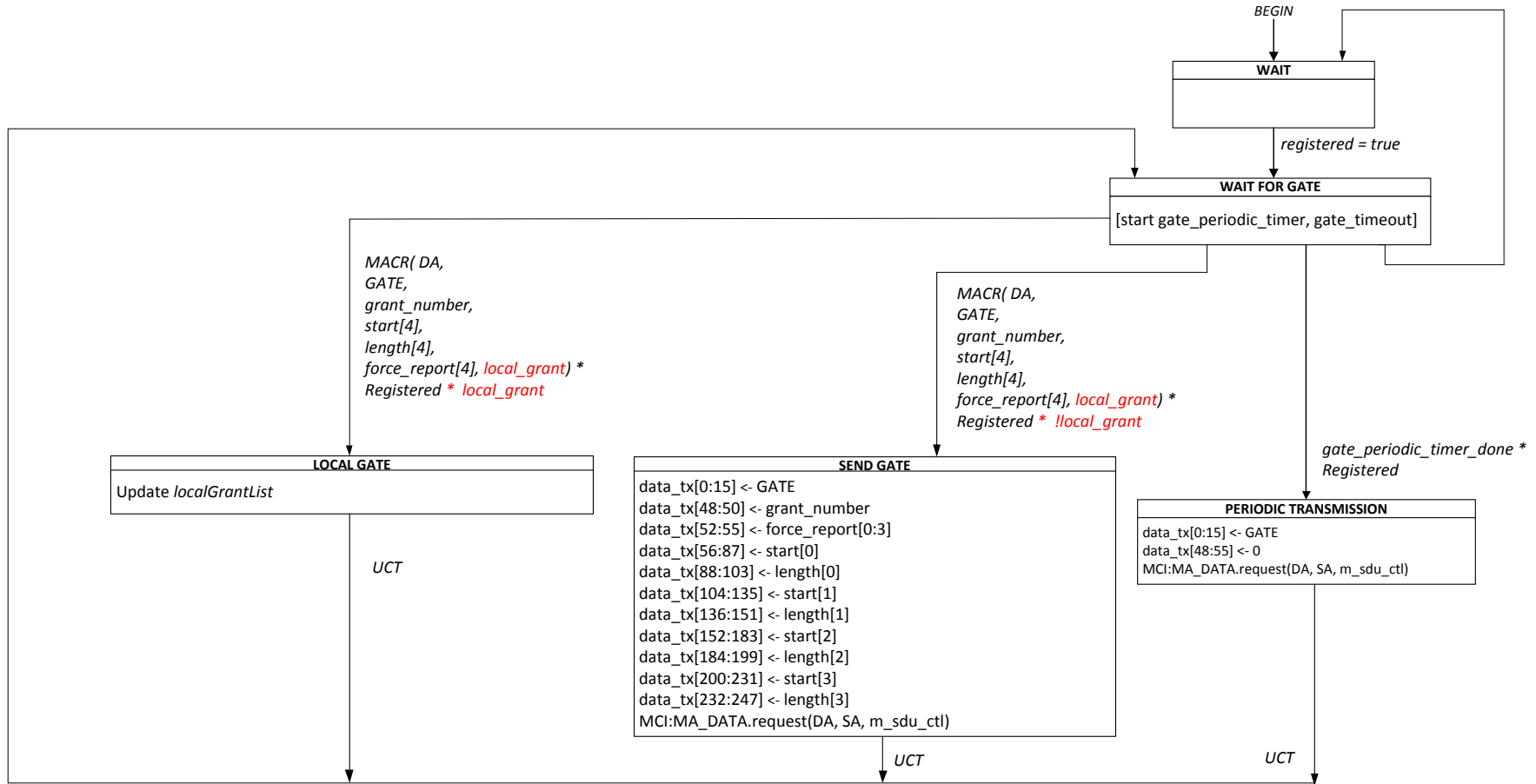
Note: the new local_grant flag is absent for other grants

Solution #3: Extension of the existing primitive



Just one request primitive entry in GATE processing block, applicable to both local and remote grants. The new boolean variable *local_grant* is used to distinguish the local grant from other grants.

Solution #3: Extension of the existing primitive (cont.)



Existing MACR request primitive is extended by addition of a new optional boolean variable named *local_grant*, with default value = FALSE unless explicitly set.

Solution #3: required changes to TDD MPCP clause

MA_CONTROL.request(DA, GATE , grant_number, start[4], length[4], force_report[4], **local_grant**). This request primitive is used by the MAC Control client at the CLT 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–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.
- 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.
- **local_grant**: **Boolean variable specifying whether the given grant is a local grant (when set to TRUE) or not (when set to FALSE). Default value is FALSE.**

Considerations

Solutions	PROS	CONS
Solution 1 – new MC address	None identified	<ul style="list-style-type: none">• Requires registration (allocation) of a new control address
Solution 2* – GATE primitive for local grant	<ul style="list-style-type: none">• Clean approach	<ul style="list-style-type: none">• Requires extension to specification
Solution 3 – extension GATE primitive	<ul style="list-style-type: none">• Minimizes specification extension	<ul style="list-style-type: none">• Less clean than Solution 2

Note: Solution 1 presents a significant disadvantage compared to Solutions 2 and 3, as requiring the allocation of a new control address, thus making Solution 1 less appealing for the amendment.

() A text proposal for solution 2 is available in [garavaglia_3bn_05_0513.pdf](#)*

Considerations (cont.)

Solutions	Specification impact	Backward compatibility
Solution 1 - new MC address	<ul style="list-style-type: none">High – requires allocation of new control address	<ul style="list-style-type: none">Unclear behavior of EPOC MAC control client with EPON MAC control
Solution 2* - GATE primitive for local grant	<ul style="list-style-type: none">Acceptable – requires new primitive description (in TDD section and section 64)	<ul style="list-style-type: none">Supported
Solution 3 – extension GATE primitive	<ul style="list-style-type: none">Minimal - requires extension to existing primitive description (in TDD section and section 64)	<ul style="list-style-type: none">Unclear behavior of EPOC MAC control client with EPON MAC control

Note: Solution 2 introduce a new primitive while guaranteeing backward compatibility, whilst solution 3 limits the specification changes to a primitive extension, but leaving some unclear behavior

(*) A text proposal for solution 2 is available in *garavaglia_3bn_05_0513.pdf*

Conclusion and Proposal

- The slides summarize the issue identified at the last meeting on the GATE processing state diagram for the TDD baseline and propose three options to resolve the issue
- For each option a description is provided and a comparative analysis is included at the end, showing that solution 1 is less desirable
 - For one of the other solutions, an example text proposal is included in document *garavaglia_3bn_05_0513.pdf*
- **Proposal** to the Task Force is to exclude Solution 1 and to focus on Solution 2 and Solution 3, for selecting the final option

References

- [1] **garavaglia_3bn_02a_0313**: “Multipoint MAC Control for EPoC – TDD Mode” – Andrea Garavaglia and Patrick Stupar (Qualcomm)