Congestion Management (CM)

Managing the Layer Stack

Jonathan Thatcher

May 2004

Things to Remember

- **Congestion Management (CM) doesn't produce more bandwidth**
  - ***It is about preferential treatment***
  - ***It is about latency vs throughput tradeoffs***
- **CM is not "quality of service (QoS)"**
  - ***In fact, not a "telecom-like service" at all***
  - ***QoS requires knowledge "deeper" than L2***
- **PAUSE < CM < QoS**
  - ***If (CM ≃ PAUSE or CM ≃ QoS) then STOP***

This Presentation Doesn't…

- This presentation doesn't attempt to justify the need for a congestion management project
  - *Others will do that.*
- This presentation doesn't address issues related to queue arbitration mechanisms
  - *This can be left to TF*

Presentation Problem Statement**

- It is possible to place the specification of Congestion Management entirely within 802.1, split between 802.1 and 802.3, or entirely within 802.3
  - *Redundancy between WG should be avoided*
- Each choice has PROs and CONs
- Making this choice subsequent to the establishment of the Task Force is a disservice to IEEE-SA, IEEE-802, IEEE-802.1 and IEEE 802.3

** Note: this is not the “big problem statement”

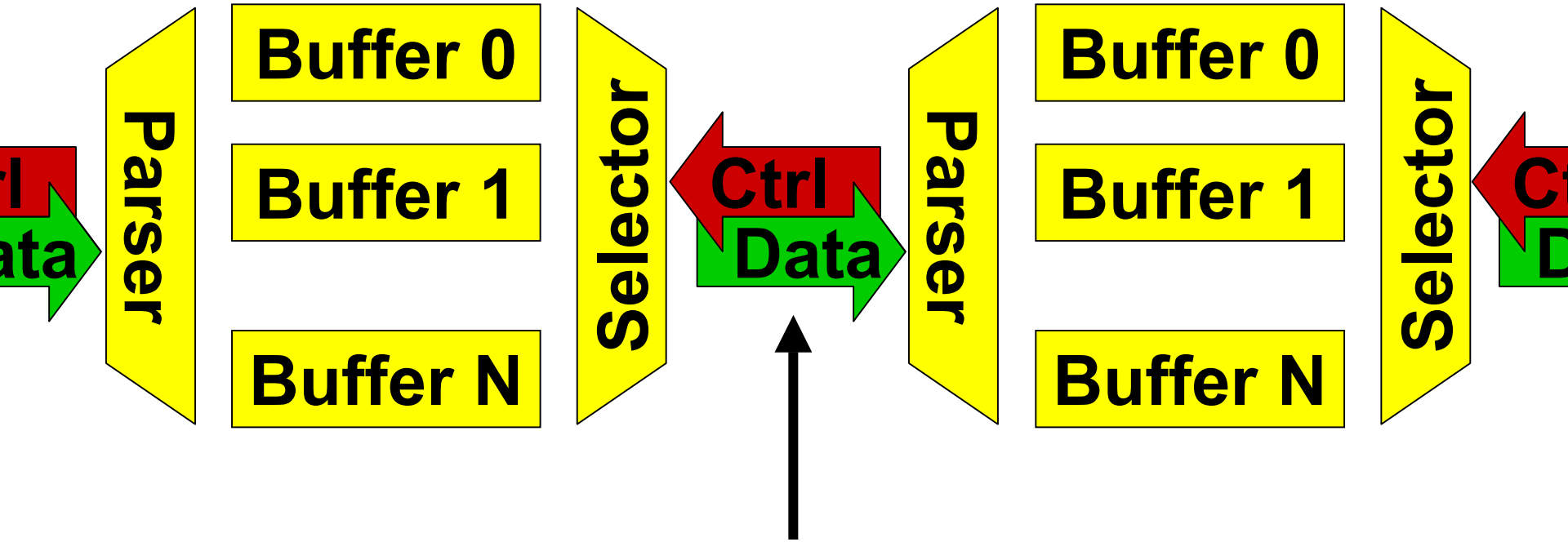
Classification of Flows

- IEEE 802 does not care how ≥ L3 data is classified and mapped to L2 class
  - *The means to classify a flow is beyond 802 scope*
  - *The association between a classified flow and one of its classified packets is beyond 802 scope*
  - *We cannot know future; need flexibility*
- The means to identify the classification of a frame within L2 is within 802's scope
  - *We do not want to look beyond L2 to identify class*
- The rules for treatment of a classification of frames within L2 is within 802's scope

Edge-to-Edge Multi-Vendor Consistency

- Network-wide consistency of operation is a key factor – perhaps the key factor – in deciding placement
- Universal, predictable CM cannot be assured if CM on the link differs from CM within the bridge
- If consistency of CM behavior (e.g. arbitration) is important, then there is a strong affinity to specification in 802.1

Generic Multi-buffer Model



- Could be an 802.3 link or an 802.1 bridge
- Full duplex nature (not shown) implicit

"Selector" is combination of arbiter and multiplexer

CM-Data Frame Identification

- **We can assume that there is a method (e.g. tag) to identify the "class" of the data frames**
  - ***Most obvious is reuse of VLAN priority bits***
    - ***Simple; may not be sufficiently flexible / forward thinking***
  - ***Could use VLAN (including or excluding priority)***
  - ***Could use new Tag (e.g. like LLID in EPON)***
  - ***Could use new EtherType***
  - ***Could use any variety and combination of things***
- **Choice of method is for Task Force, not SG**
  - ***Highly influenced by placement of CM function***
- **The important point here is that CM-DATA packets exist and are readily parsed independent of payload**

CM-Control Frame

- **We can assume that there is a new "Congestion Management Control Frame (CM-CTRL)" that is passed on the link.**
  - ***Most obvious is MAC-Control with new op-code***
    - ***TLV format supporting one or more "class identifiers"***
    - ***Field for one "class identifier" per control packet***
  - ***Could use PAUSE with additional tag***
  - ***Could use new EtherType***
    - ***Still sent as control packet in order to avoid being paused***
    - ***Issues with link-aggregation & OAM which are resolvable, but more complex***
- **The important point here is that CM-CTRL packets exist and are readily parsed**

Service Interface Reconciliation (almost)

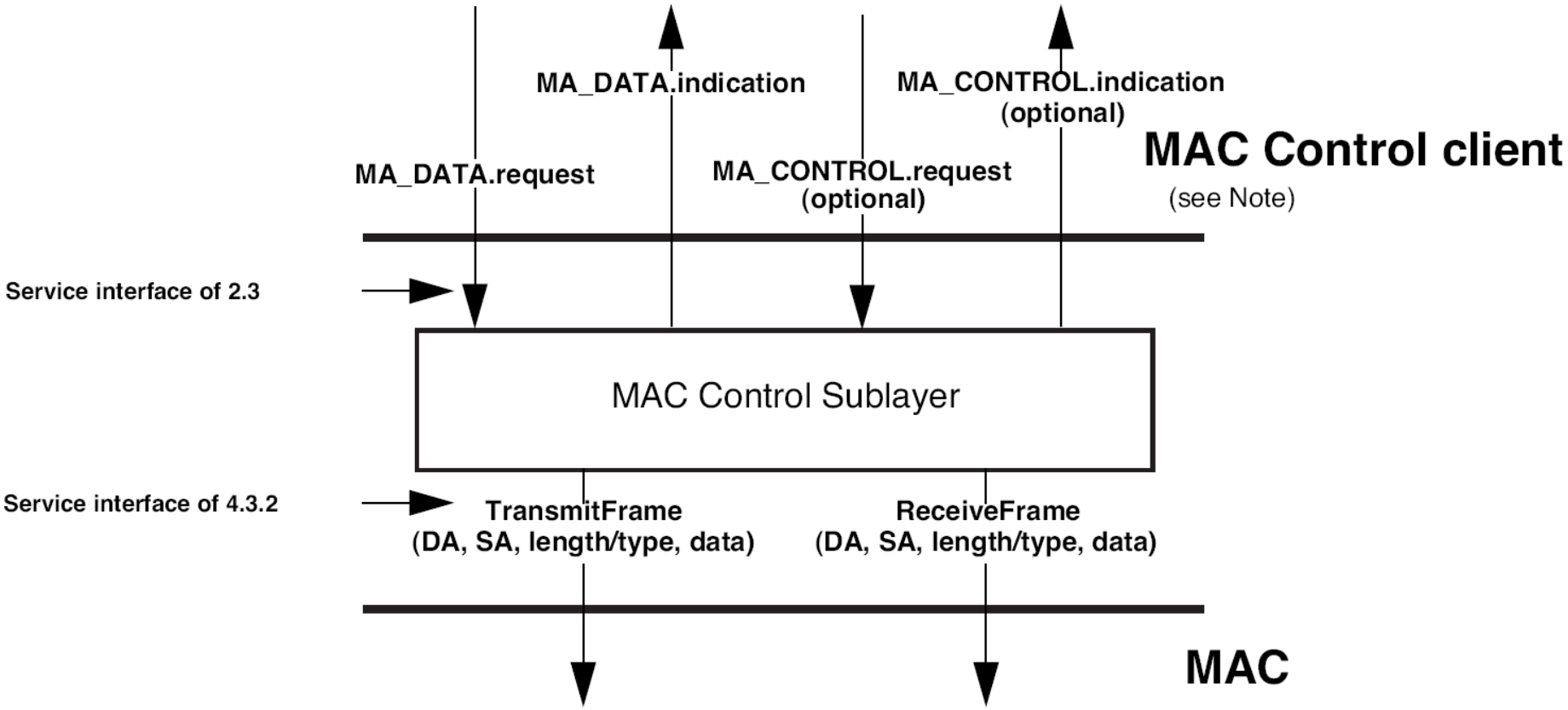
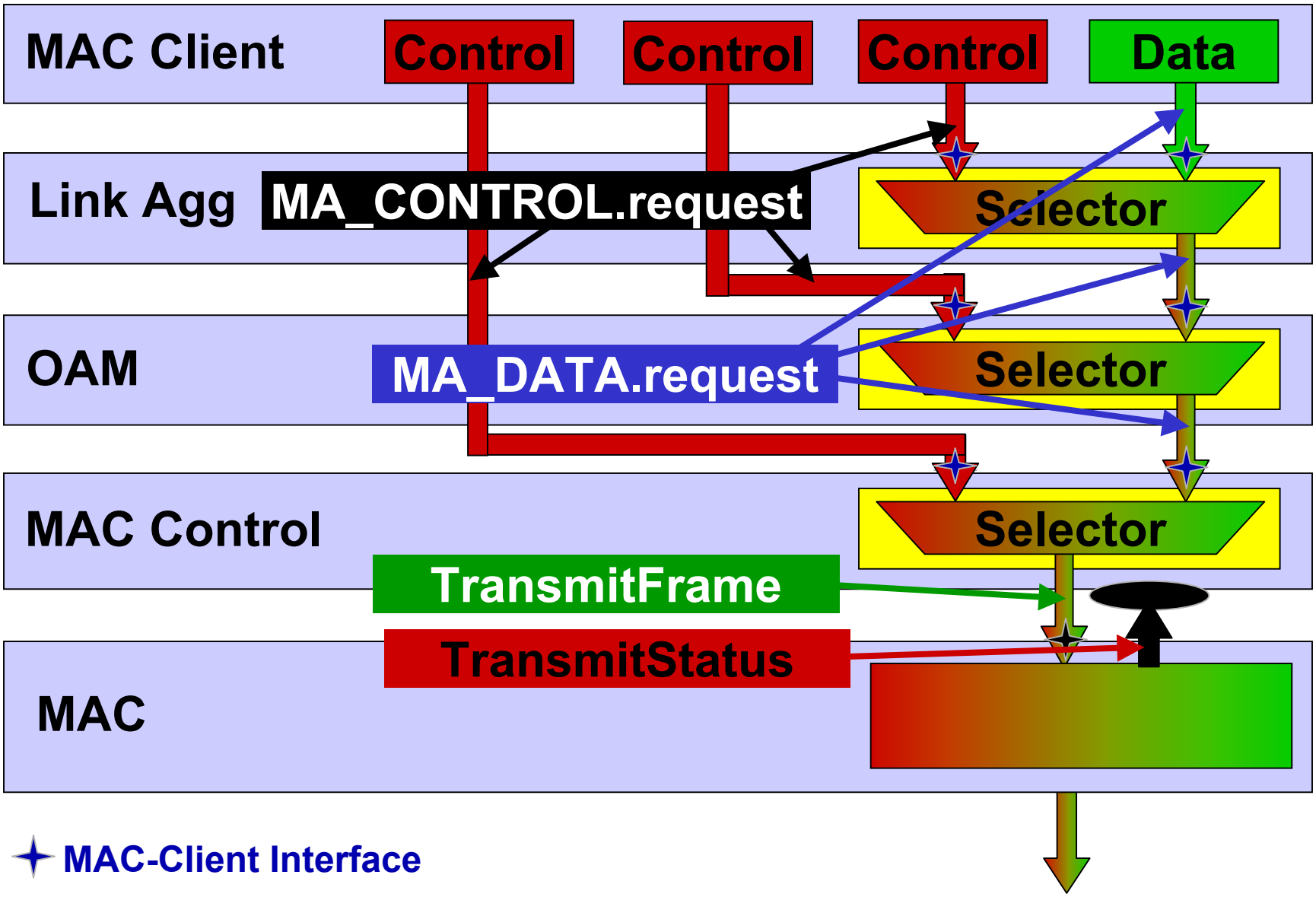


Figure 31–2—MAC Control sublayer support of interlayer service interfaces

802.3-2002 Clause 31.3 NOTE — In the absence of the MAC Control sublayer, Clause 31 makes no attempt to reconcile the long-standing inconsistencies between the interface definitions in subclauses 4.3.2 and 2.3. These existing inconsistencies have not historically hampered the construction of interoperable networking equipments, and are not sufficiently important to merit further attention.

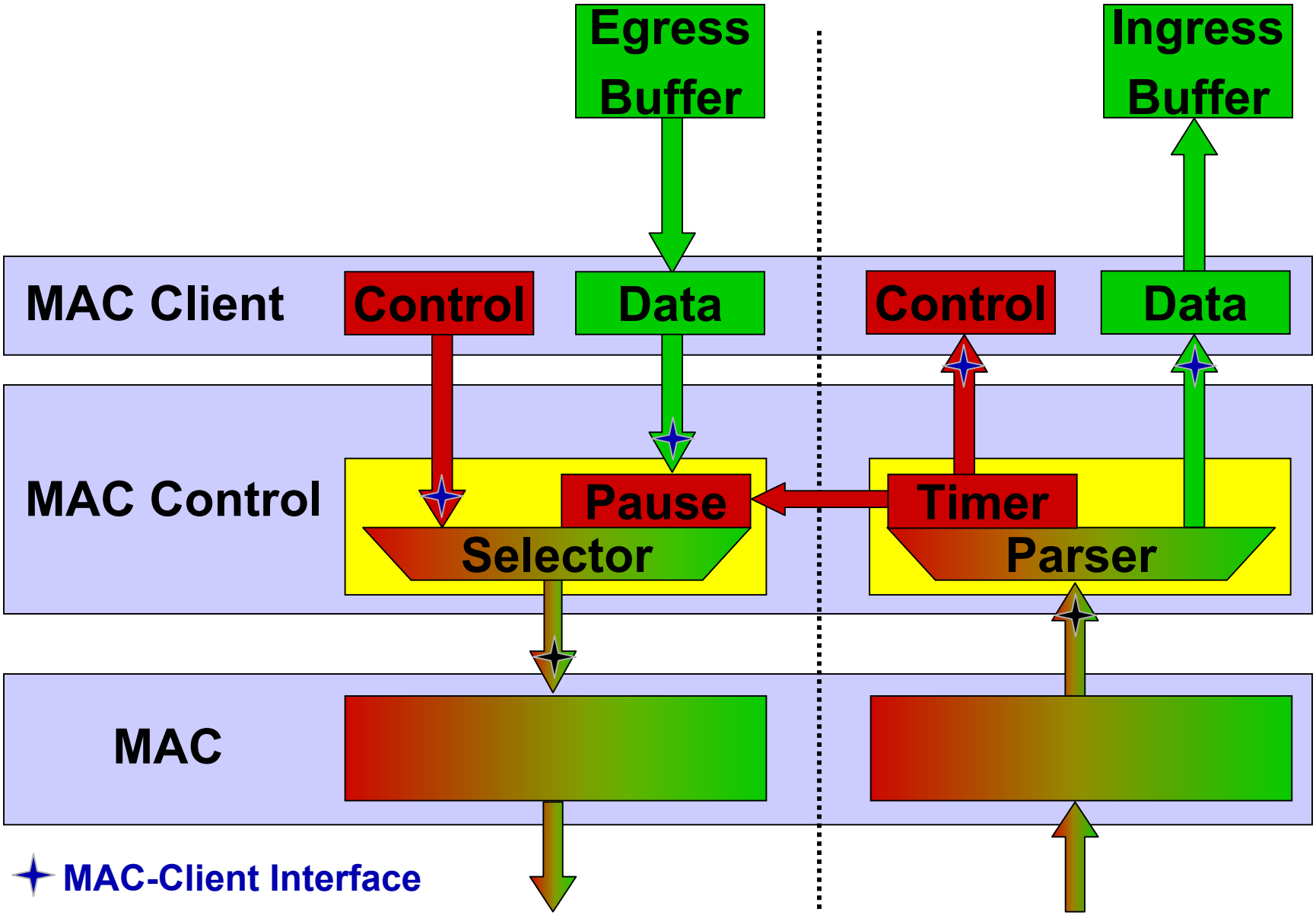
Sublayer Stack – Tx Only (Rx Similar)



No Explicit Pacing Mechanism

- It is implicit within the standard that upper sublayers use the MAC's TransmitStatus I/F to pace (gate) the MA_DATA.request mechanism and thus avoid overflowing the MAC
  - *In fact, TransmitStatus is explicitly used only by layer management (see Clause 5)*
  - *There is no explicit requirement to avoid sending multiple MA_DATA.request primitives*
  - *It is explicit that the MAC will only see the MA_DATA.request active at the time it services the next frame from the upper layer; no other MA_DATA.requests will be seen, implying intermediate requests (frames) are dropped*

Basic MAC Control (Clause 31)



Assumptions

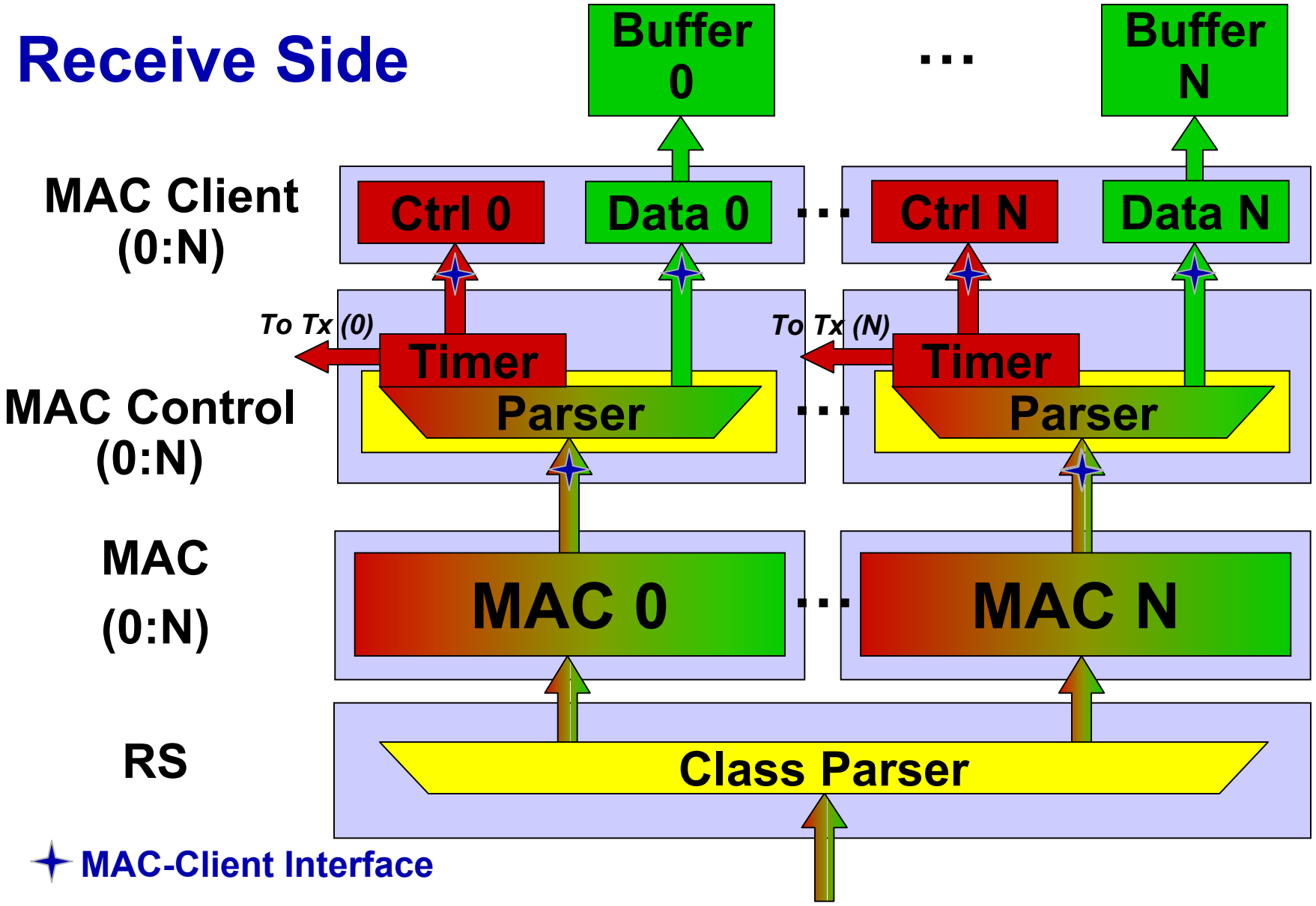
- **No simultaneous usage of current PAUSE & new congestion management assumed here**
  - ***It is possible to have PAUSE default to a class or behave as an "all class" control, but the complications outweigh any advantage***
  - ***Ultimately, this decision is for the Task Force***
  - ***Highly influenced by placement of CM function***
- **The choice of arbitration schemes is to the first order independent of the location of the selector/parser**
  - ***Selection of sublayer affects location of preponderance of work***
  - ***Selection of arbitration scheme is work for the TF***

Placement Options

- It is possible to put the parser and selector in different sublayers. But, there appears to be no advantage to doing so, and there are many disadvantages
  - *Don't want to compromise layer architecture*
  - *Don't want communication between Rx and Tx to cross layer boundaries*
- Therefore, only 4 placements will be considered:
  - *Reconciliation Sublayer (RS)*
  - *MAC Control*
  - *MAC Client (with MAC Control)*
  - *MAC Client (without MAC Control)*

Class Parser in Reconciliation Sublayer

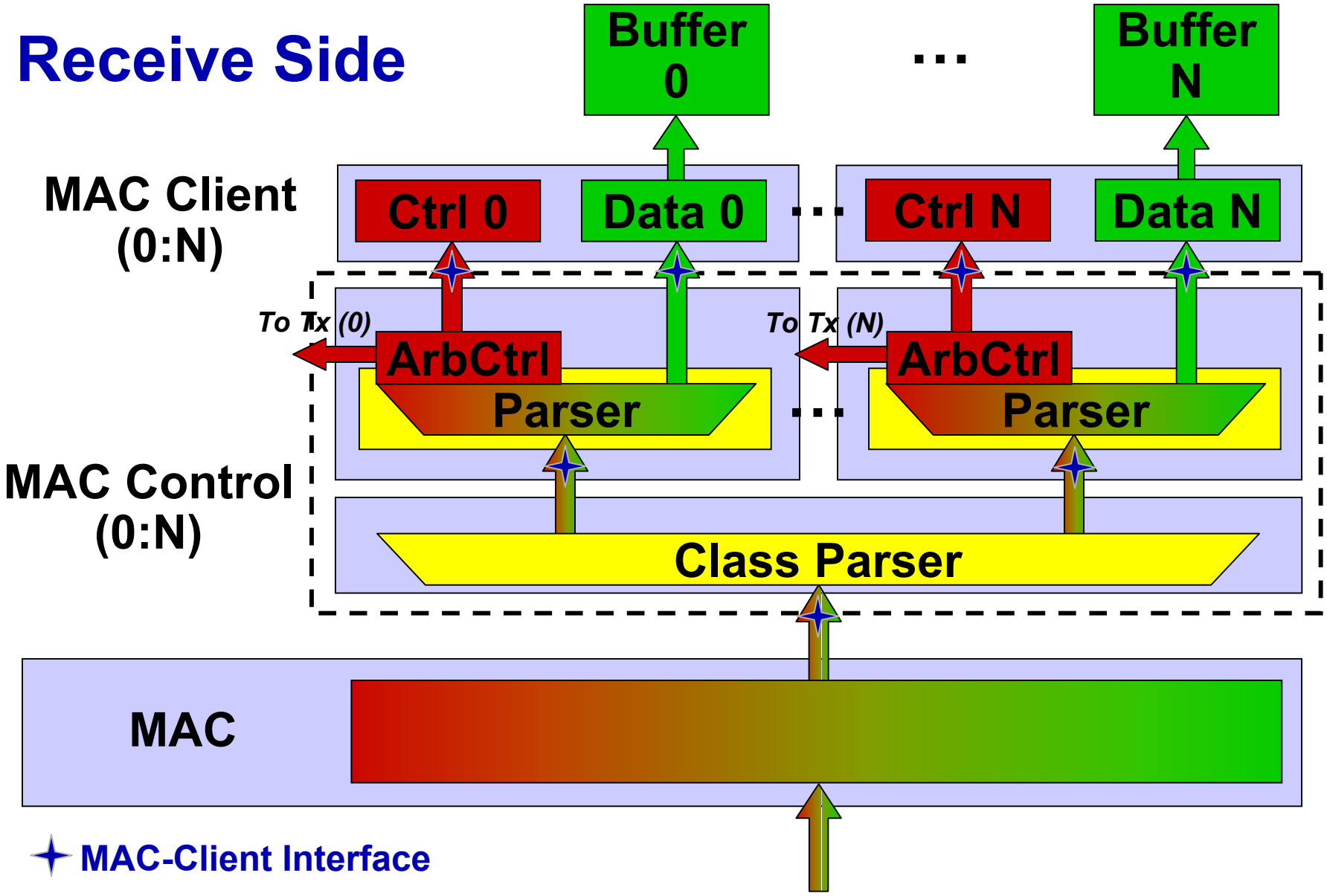
Receive Side



Class Parser in Reconciliation Sublayer

- **Notes:**
  - ***This is not recommended -- just because it can be done, doesn't mean that it should be***
  - ***This is shown for completeness and simplicity to grasp***
  - ***If new tagged MAC-Control frame, then "class parser" simply steers CM-Control & CM-Data packets to correct MAC (0:N)***
    - ***Changes required to MAC-Control for CM tag are minimal***
      - ***Tag might be added/stripped at "class parser sublayer" eliminating changes to the MAC Control layer and above***
  - ***If new CM-Control frame (e.g. new OpCode for MAC-Control), then "class parser" would sink CM-Control frame and source new CM-Control(s) for MAC (O:N)***
    - ***It is not clear that this would make any sense unless the CM-Control frame through the MAC is PAUSE***
      - ***Which implies no changes in the MAC or MAC Control sublayers***
      - ***Hence "Timer" shown in MAC-Control sublayer***

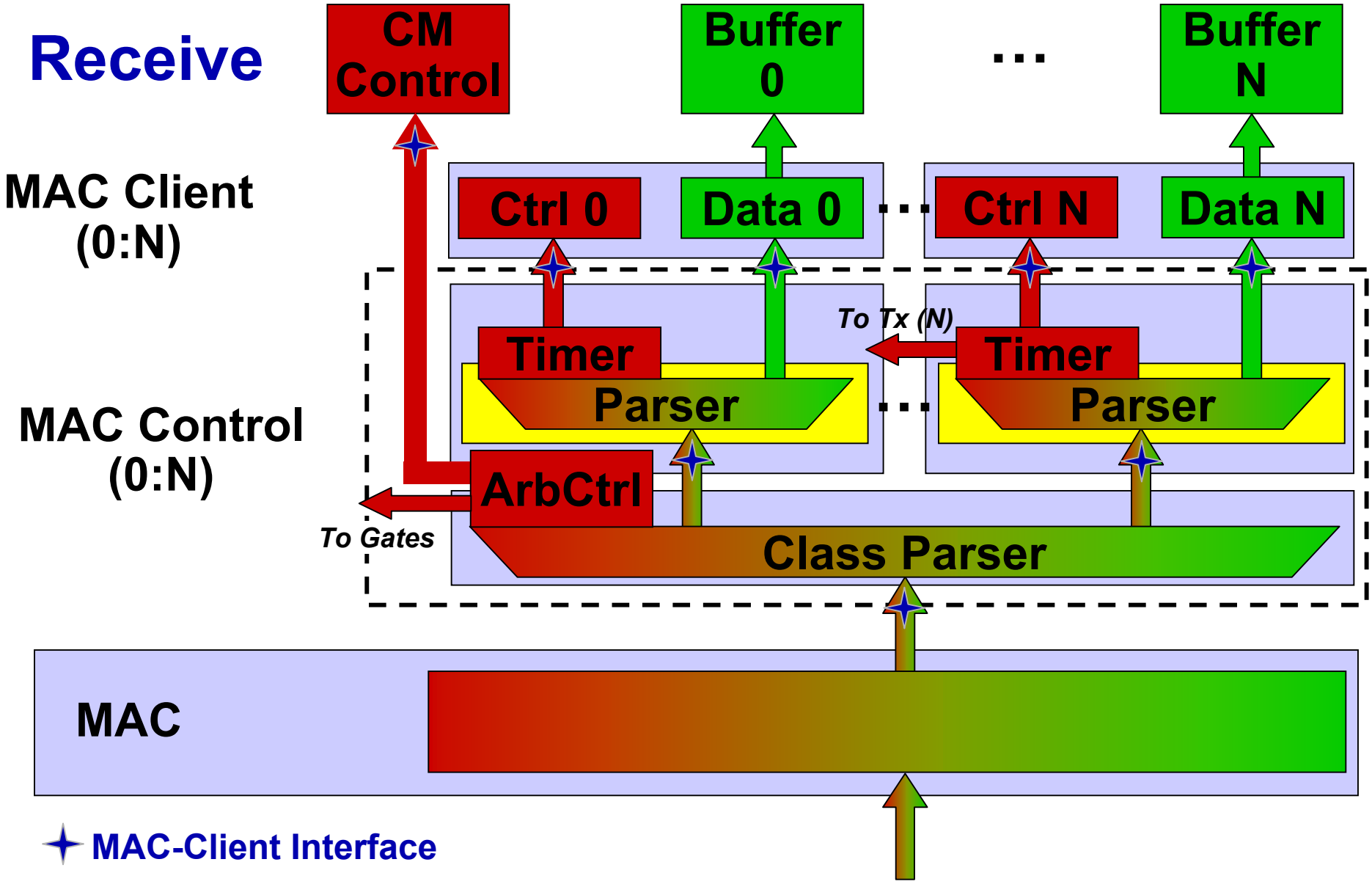
Class Parser in MAC Control Sublayer



Class Parser in MAC Control Sublayer

- **Notes:**
  - ***This figure is functionally equivalent to having a figure with MA_DATA.indicate (packets) tagged with class (0:N) and showing a single MAC-Client interface***
  - ***If new tagged MAC Control frame, then "class parser" simply steers CM-Control & CM-Data packets to correct MAC-Control (0:N)***
    - ***Changes required to MAC-Control for tag are minimal***
      - ***Tag might be added/stripped at "class parser sublayer" eliminating changes to the MAC Control sublayer and above***
  - ***If new CM-Control frame, then parser would sink CM-Control frame and create new Control frames for MAC-Control (O:N)***
    - ***Control frame to MAC-Control could be PAUSE***
      - ***Reduction in work if MAC-Control frame through MAC-Control is PAUSE***

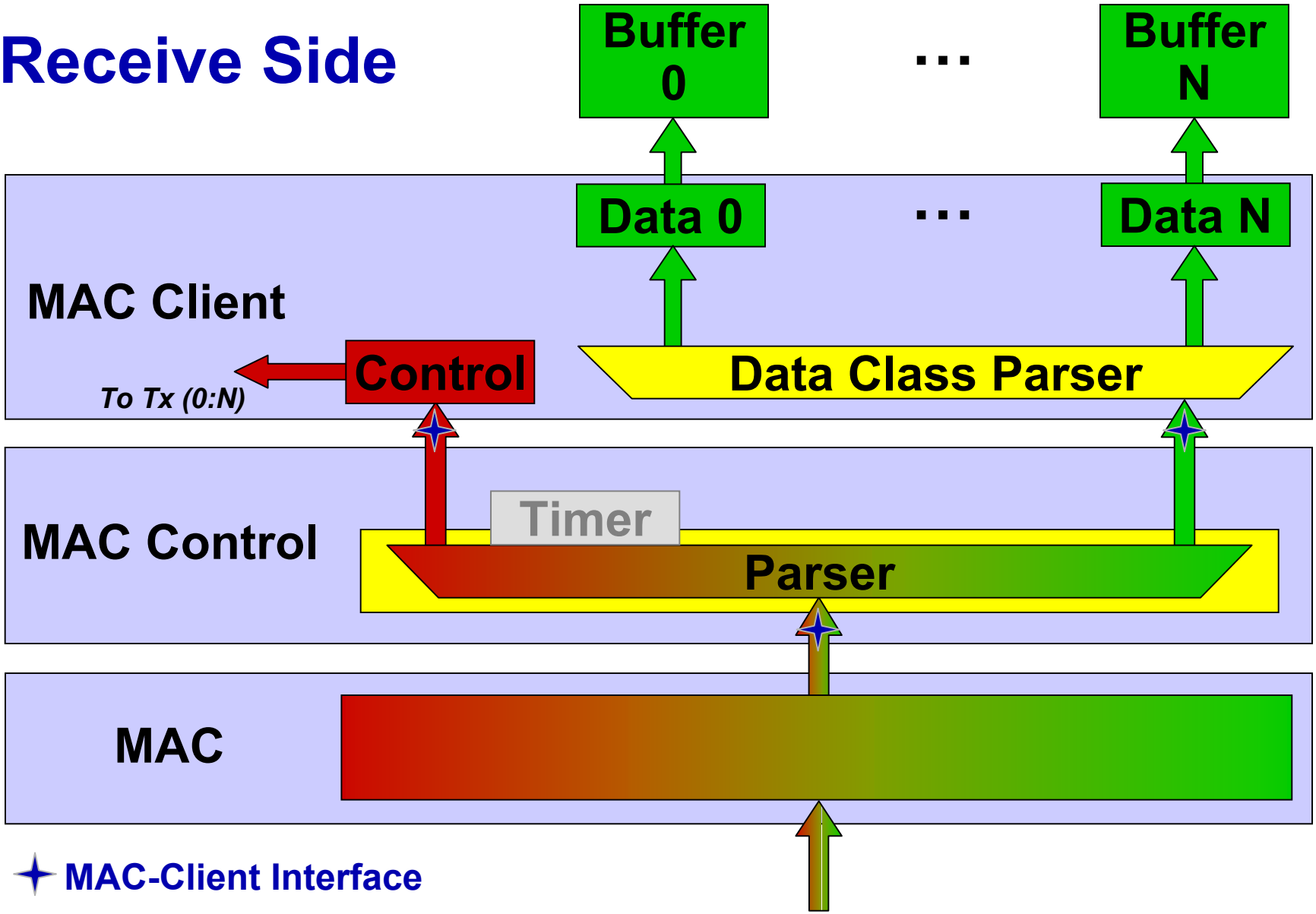
Single CM-Control Method



Single CM-Control Method

- **This is a variation Parser/Selector in MAC-Control Sublayer**
  - ***Key point is leaving existing PAUSE logic alone***
  - ***Multiple sub-variations are possible***
    - ***No reason to show these as separate choices as they do not substantially affect the key decisions at this stage.***

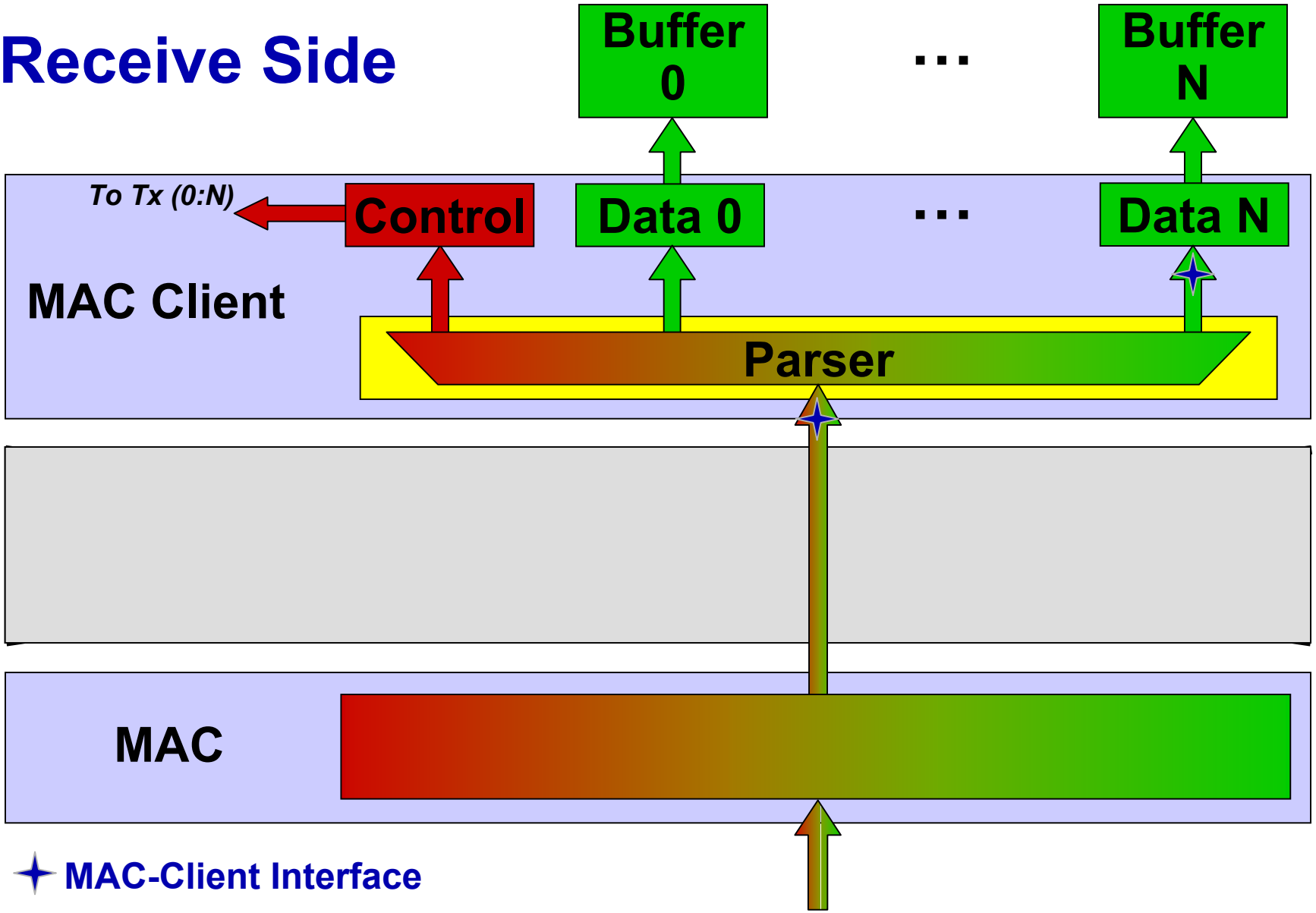
Class Parser in MAC Client Sublayer



Class Parser in MAC Client Sublayer

- **Notes:**
  - ***MAC-Control parser simply steers CM-Control frame to MAC-Control-Client***
    - ***If single Control Client (as shown), then MAC-Control sublayer needs to support a new OpCode***
    - ***It is possible to have one Control Client per class in which case there would be a Control Class Parser in addition to the Data Class Parser within the Client sublayer.***
    - ***Might work within MAC-Client as PAUSE does today***
      - ***This does not limit flexibility for arbitration schemes***

Class Parser in Client; no MAC Control

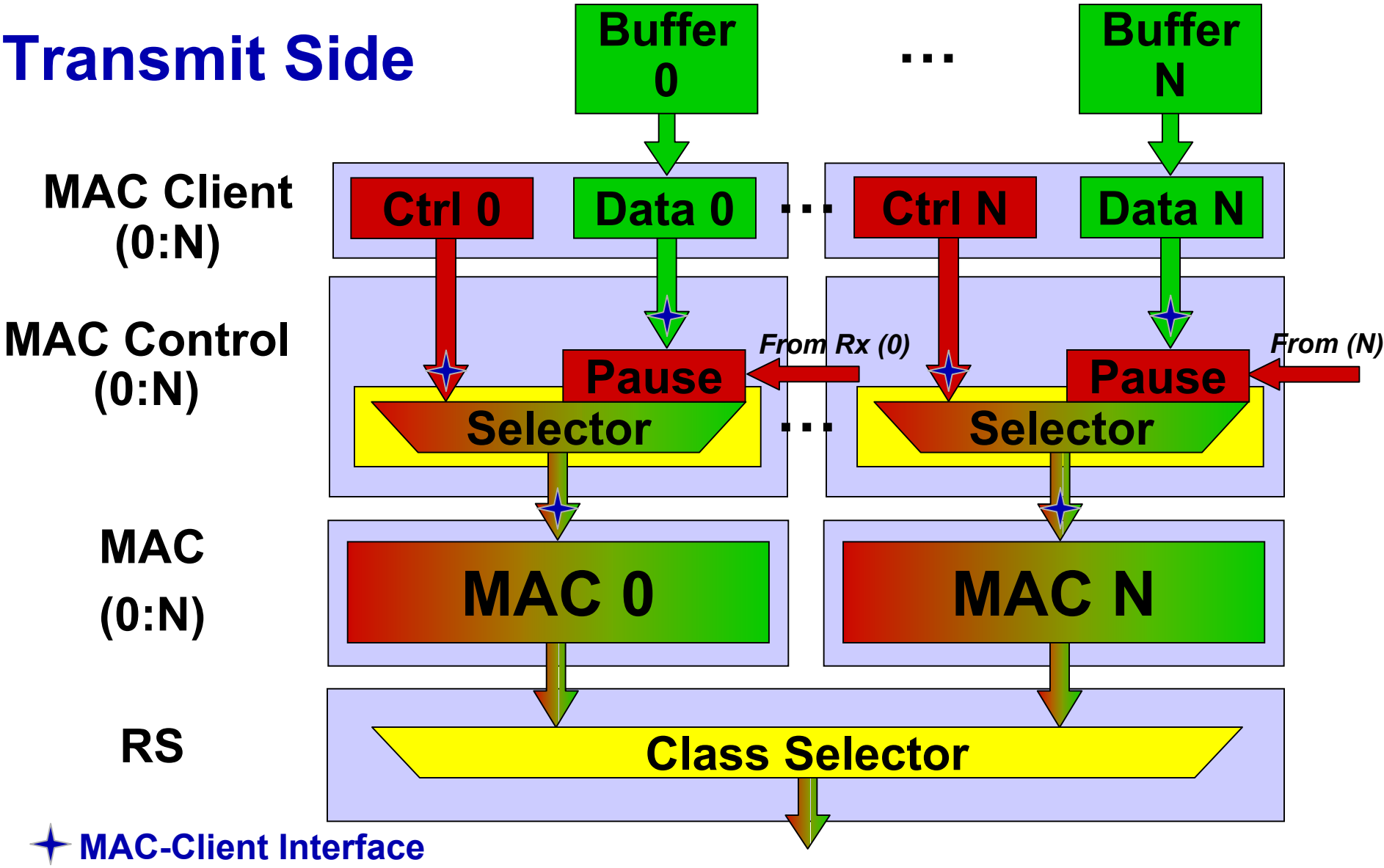


Class Parser in Client; no MAC Control

- **Notes:**
  - ***Similar behavior to Class Parser in Client (with MAC Control Sublayer)***
  - ***But, no MAC-Control should ever be used!***
    - ***Else, potential to block CM-Control packets, which use the MA-Data path***
    - ***No support for PAUSE (no loss)***
    - ***No support for EPON (no loss)***
    - ***No support for future MAC-Control additions (loss?)***
  - ***Implies no work for 802.3***

Class Selector in Reconciliation Sublayer

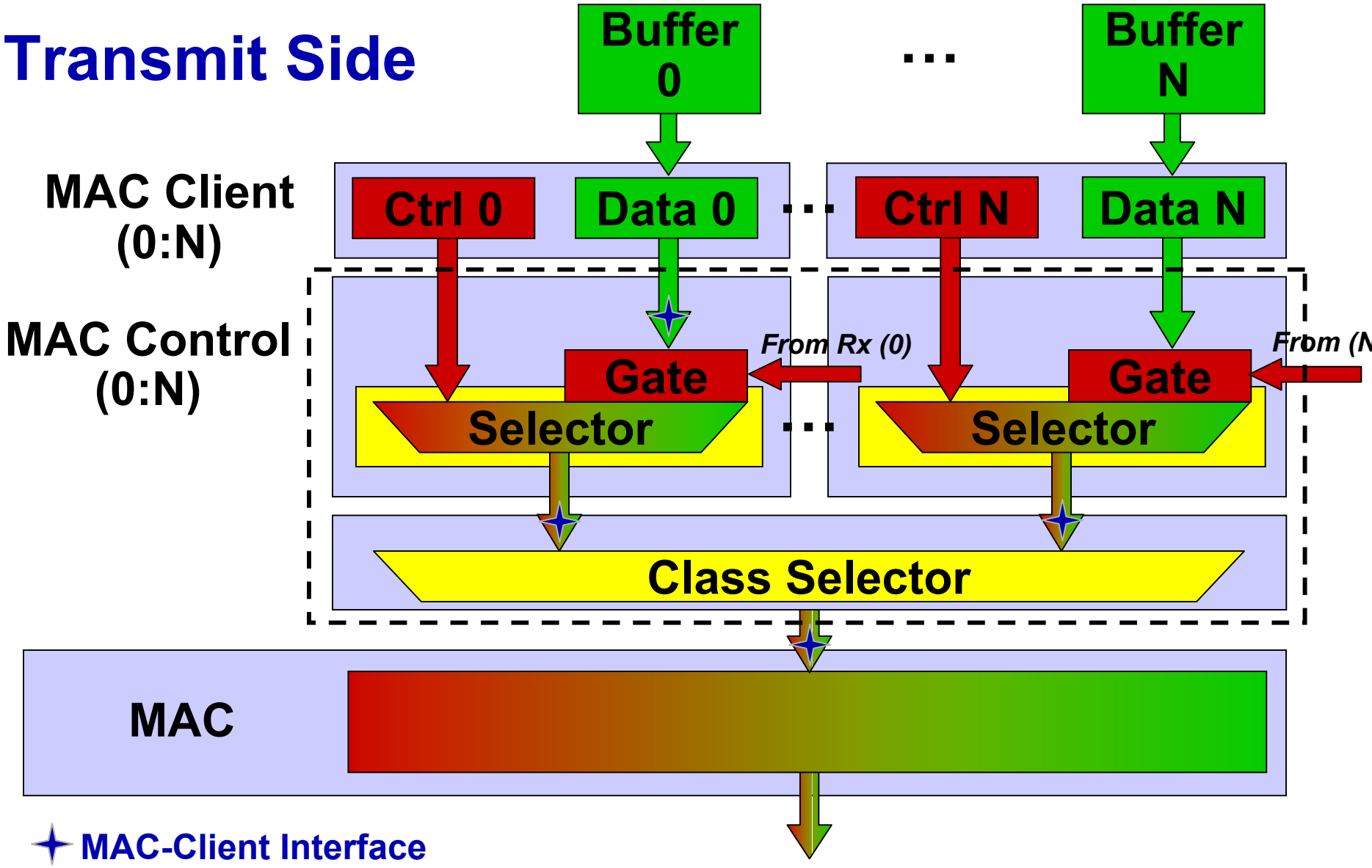
Transmit Side



Class Selector in Reconciliation Sublayer

- **Notes:**
  - ***This is not recommended -- just because it can be done, doesn't mean that it should be***
  - ***This is shown for completeness and simplicity to grasp***
  - ***If new tagged MAC Control frame, then "class selector" leaves CM-Control & CM-Data packets unmodified***
    - ***Changes required to MAC Control for new control code (minimal)***
      - ***Tag might be added/stripped at "class parser sublayer" eliminating changes to the MAC Control layer and above***
  - ***If new CM-Control frame, then "class selector" would sink MAC-Control frames for MAC (0:N) and source new CM-Control(s)***
    - ***Reduction in work if Control frame through MACs is PAUSE***
  - ***It is not clear that this would make any sense unless the control frame through the MAC is PAUSE***
    - ***Which implies no changes in the MAC or MAC Control sublayer***

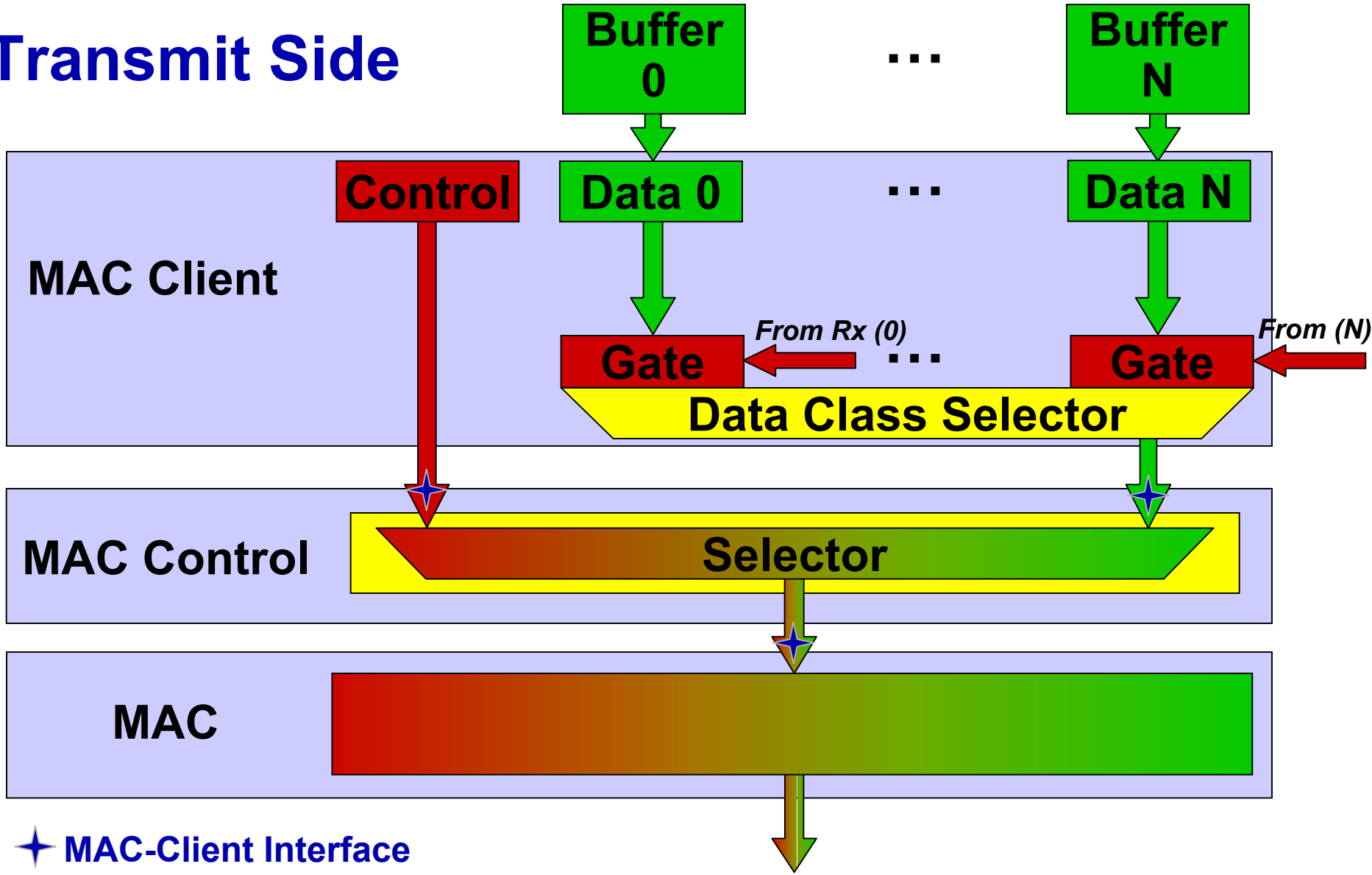
Class Selector in MAC Control Sublayer



Class Selector in MAC Control Sublayer

- **Notes:**
  - ***If new tagged MAC Control frame, then "class selector" leaves CM-Control & CM-Data packets unmodified***
    - ***Changes required to MAC Control for new control code (minimal)***
      - ***Tag might be added at "class selector sublayer" eliminating changes to layers above***
  - ***If new CM-Control frame, then "class selector" might sink MAC-Control frames from MAC-Control (0:N) and source new CM-Control(s)***
    - ***Reduction in work if PAUSE used***

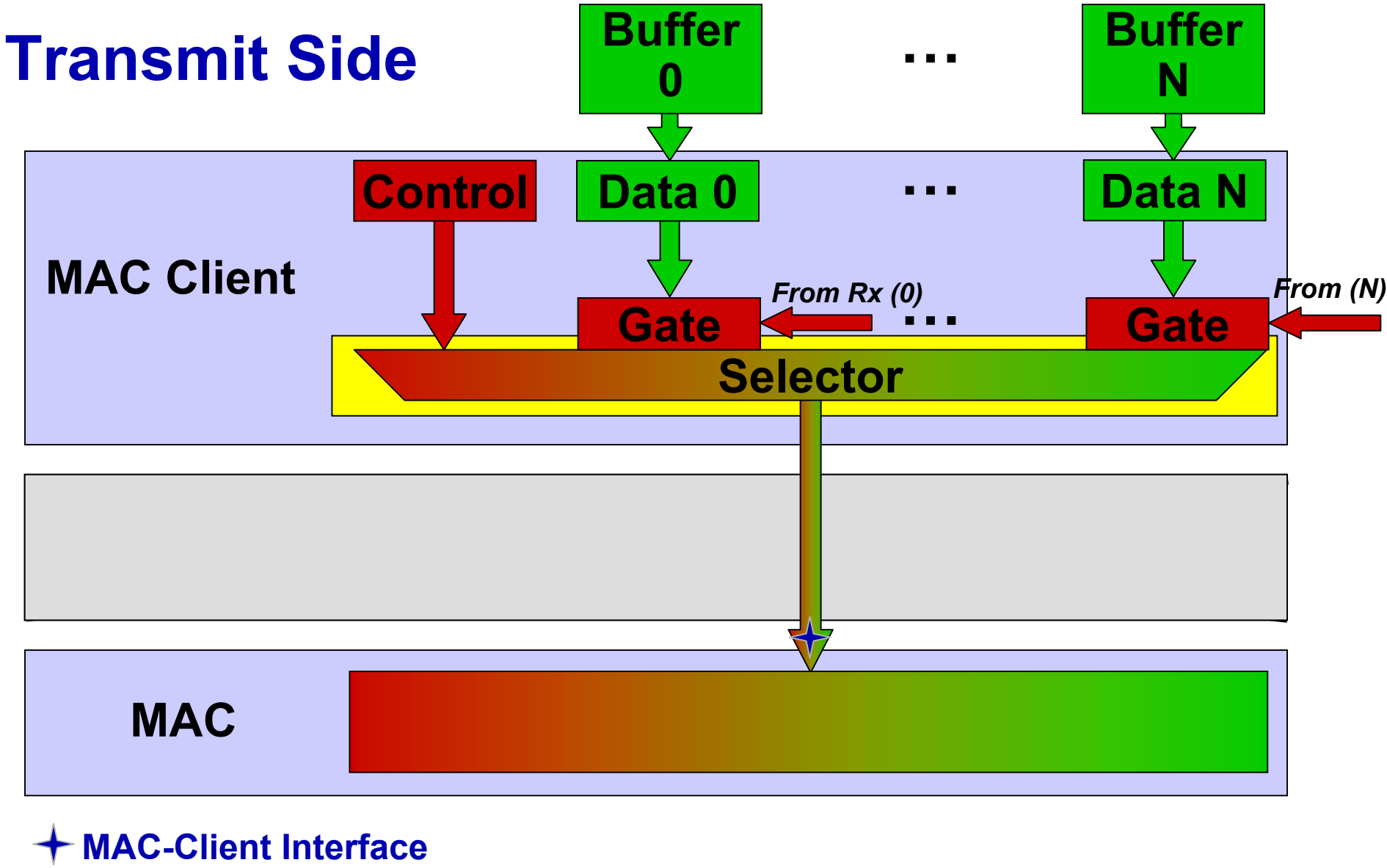
Class Selector in Client Sublayer



Class Selector in Client Sublayer

- **Notes:**
  - ***MAC-Client Data Class Selector "simply" arbitrates between CM-Data classes***
    - ***If single Control Client (as shown), then MAC-Control sublayer needs to support a new CM-Control-Request***
    - ***It is possible to have multiple Control Clients per class (vs single Control shown in Client) in which case there would be a Control Class selector in addition to the Data Class selector within the Client sublayer.***
      - ***If new CM-Control, then selector will sink Control requests and create CM-Control requests***
    - ***Might work within MAC-Client as PAUSE does today***

Class Selector in Client; no MAC Control



Class Selector in Client Sublayer

- **Notes:**
  - ***Similar behavior to Data Class Selector in Client (with MAC Control Sublayer)***
  - ***See notes for Rx side***

Selection Criteria

- Consistency with existing architecture
- Simplicity
  - *Ease of understanding*
  - *Confusion free*
  - *Interoperable*
  - *Open fewest number of clauses for change*
- Flexibility

Analysis of Placements (1/3)

- Parser/Selector in Client – basically what is done today
  - *Sans the congestion management gating*
- If CM control selector below LinkAgg, then selector must either:
  - *choose which link to forward CM management frames (very complex), or*
  - *default to a single link (is delay a problem?)*

Analysis of Placements (2/3)

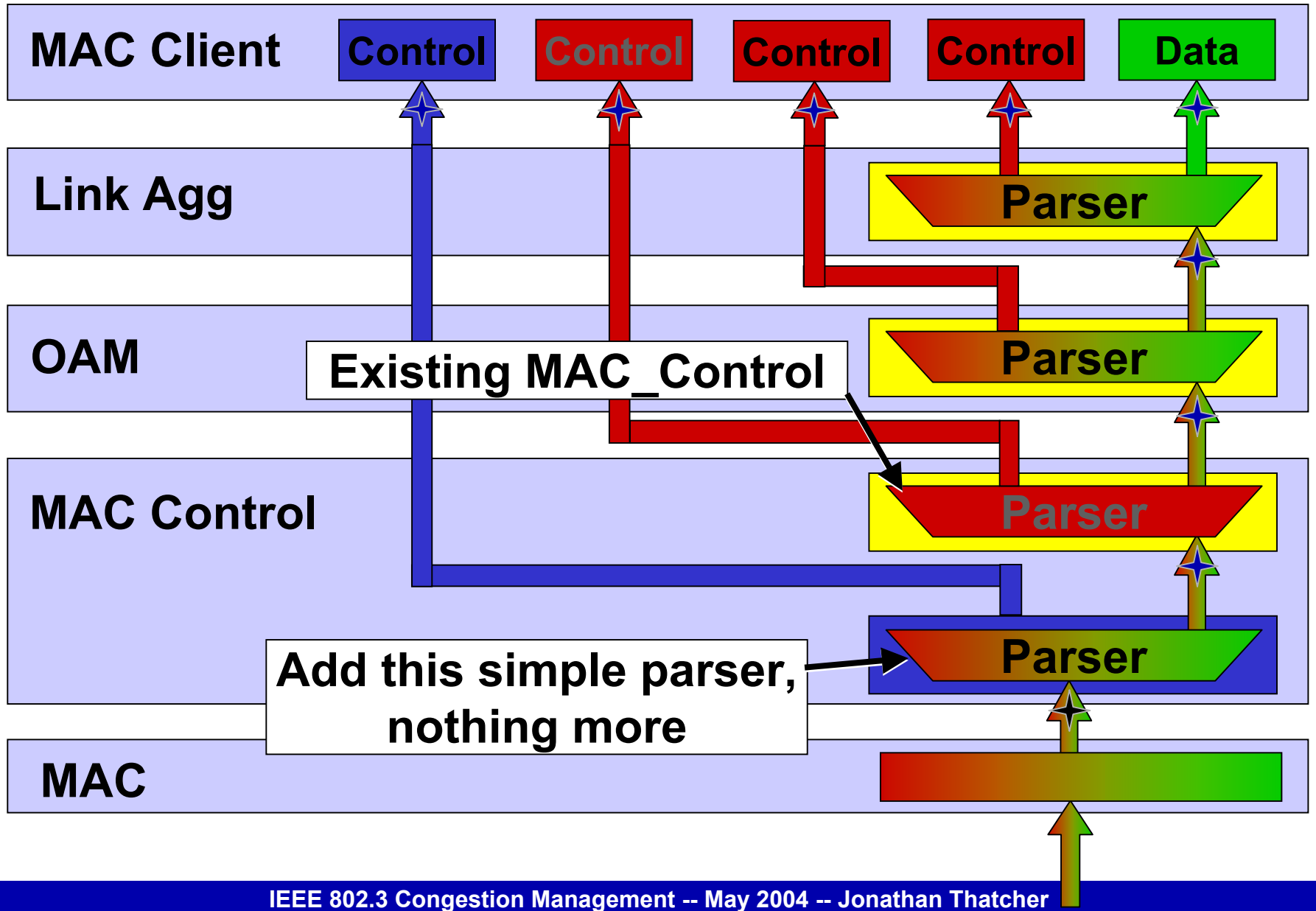
- CM_MA_CONTROL.request priority (relative to LinkAg, OAM…) dependent on placement of selector in sublayer stack
  - *If highest priority control, wants to be near MAC*
  - *If lowest priority control, wants to be near Client*
  - *Ideal is for Client to arbitrate all control priorities – meaning, gate only one MA_CONTROL.request at a time*
- MA_CONTROL.request becomes MA_DATA.request as it moves down through sublayers
  - *If priority different than sublayer stack order, CM selector in MAC_Control would have to parse and differentiate not just class of data, but also type of control in order to correctly "select" next packet*

Analysis of Placements (3/3)

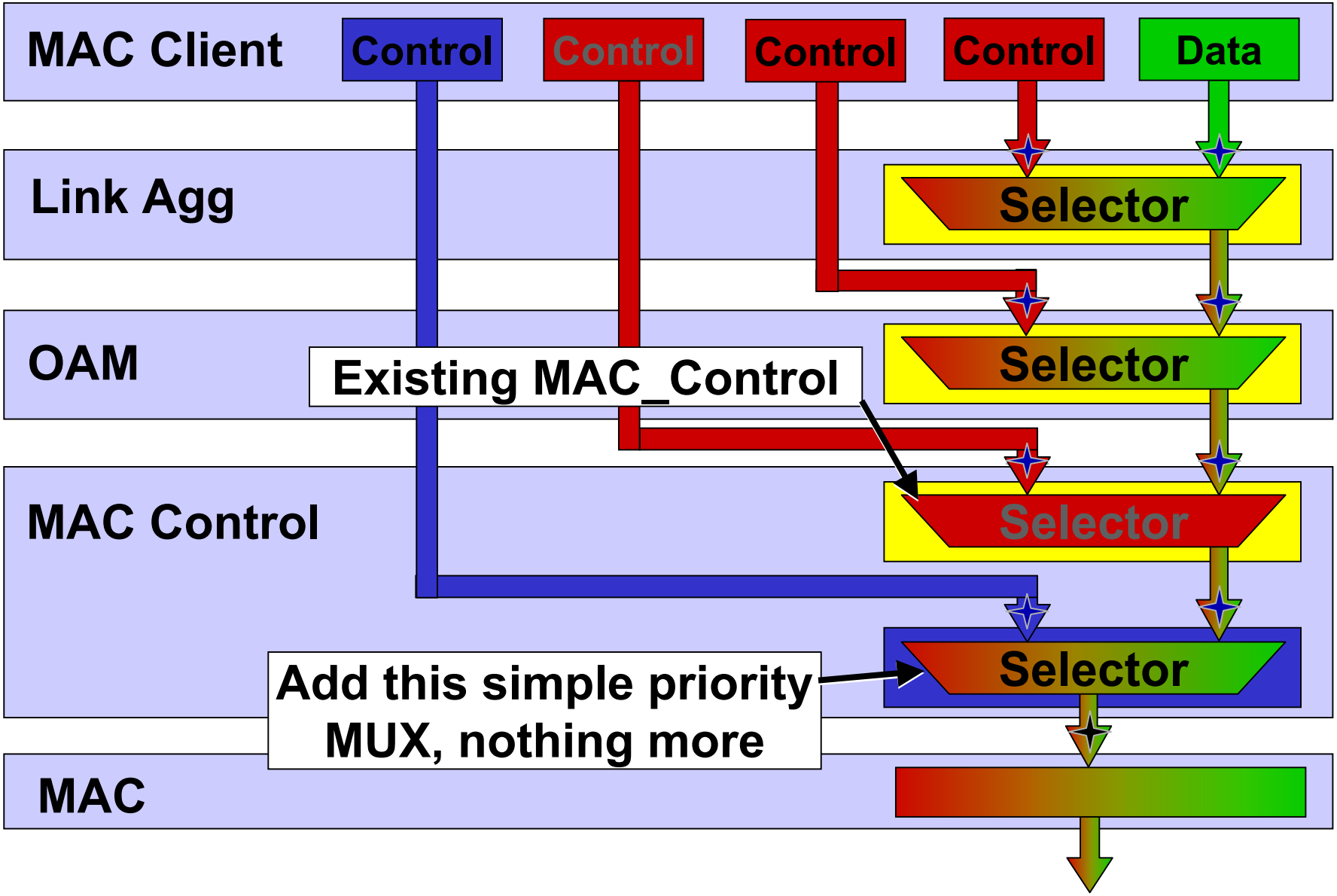
- Selection at lower sublayers implies availability of multiple data classes (MA_DATA.request) and control types (MA_CONTROL.request) simultaneously.
  - *In today's sublayers MA_CONTROL.request blocks MA_DATA.request*
  - *This can work, but it implies that as MA_CONTROL.request(s) is asserted, MA_DATA.request(s) assert and de-assert…. This adds to the existing sublayer interface architecture confusion*

Recommendation

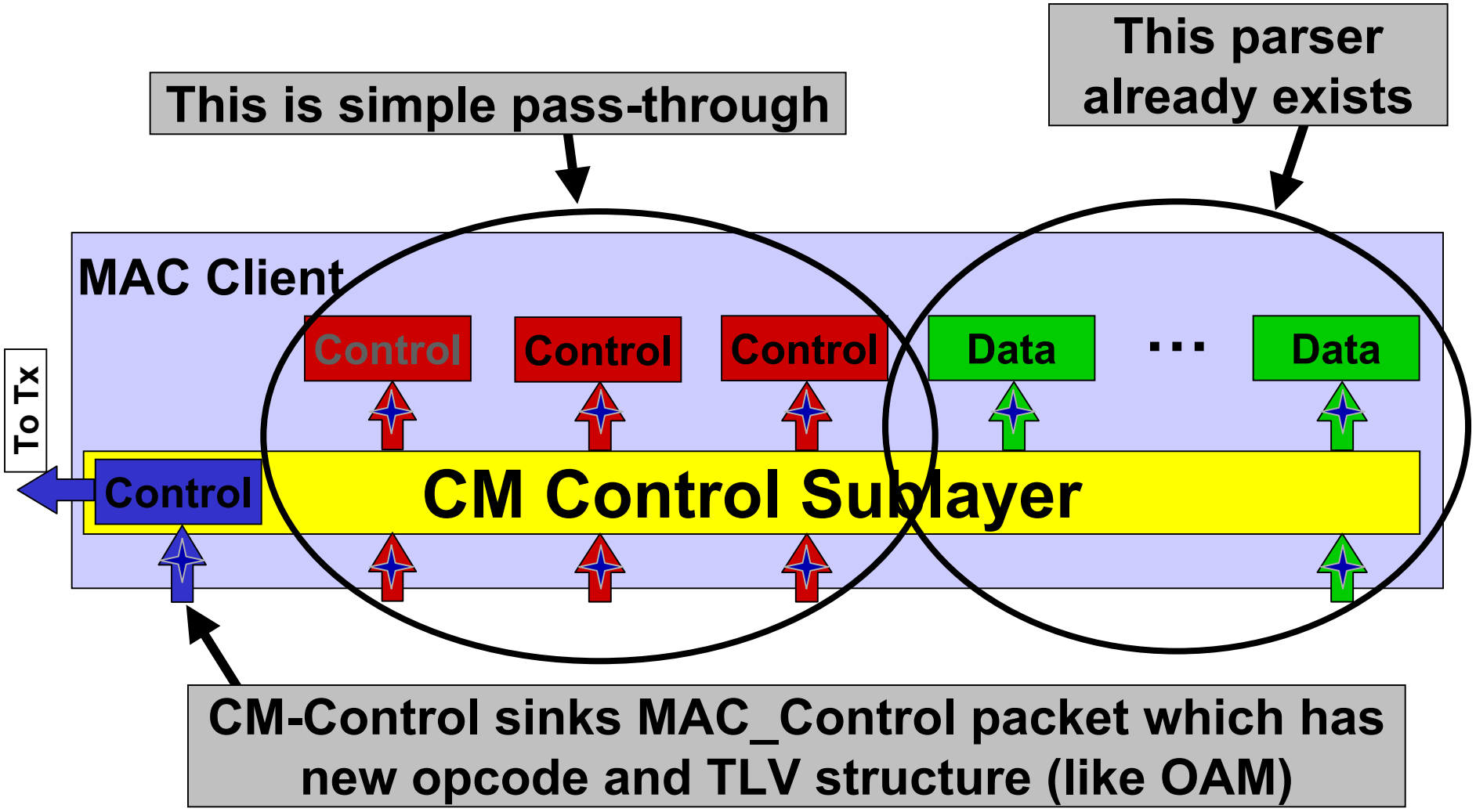
Lower Receive Sublayer Stack



Lower Transmit Sublayer Stack



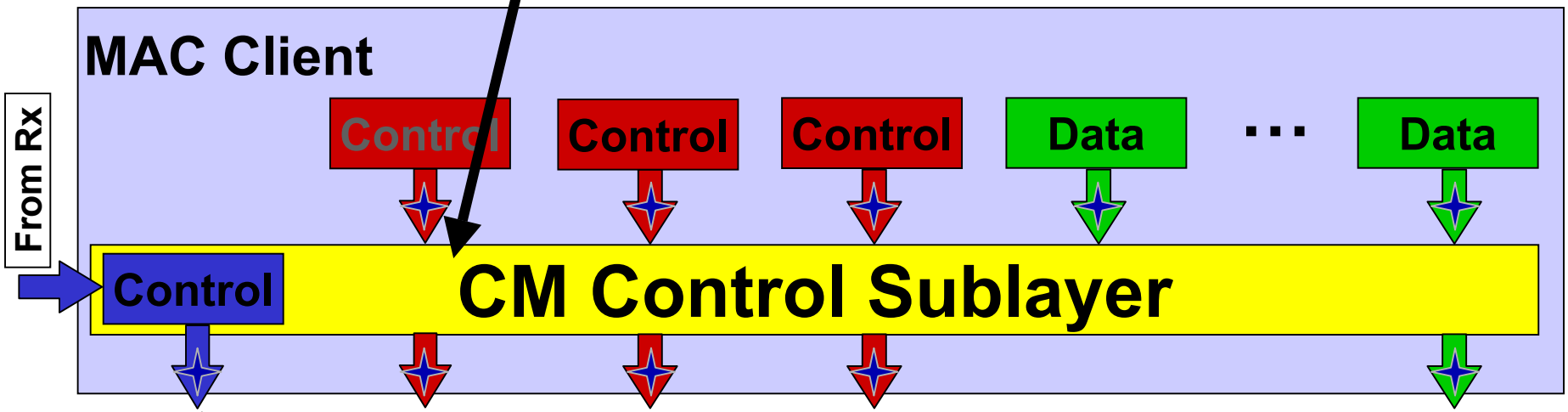
Upper Receive Sublayer Stack



Not Shown: Queue Manager feedback path to CM Control

Upper Transmit Sublayer Stack

CM Control Sublayer gates all MA_DATA.request and MA_CONTROL requests ensuring proper priority and timing of all requests



CM-Control sources MAC_Control packet which has new opcode and TLV structure (like OAM)

This thin sublayer exists between 802.3 and 802.1

Advantages

- **Minimalist modifications to existing clauses**
  - ***Clause 31***
  - ***Management***
  - ***Avoids changes to most confusing and ambiguous aspects of standard***
- **Avoids complexities related to LinkAg (& OAM)**
- **Maximum flexibility**
  - ***To future sublayers between MAC and Client***
  - ***Single point of control -> predictability***
  - ***Arbitration scheme is independent of this structure***
- **Proximity to existing queue management**
  - ***No complexity added to existing 802.3 clauses***

Related Objectives

- No change to:
  - *802.1 / 802.3 layer architecture*
  - *MAC_Client Interface*
  - *Link Aggregation*
  - *OAM*
  - *MAC (Clause 4 or 4A)*
  - *PCS / PMA / PMD*
- No simultaneous support for CM and PAUSE
- No traffic classification
- No reordering within class