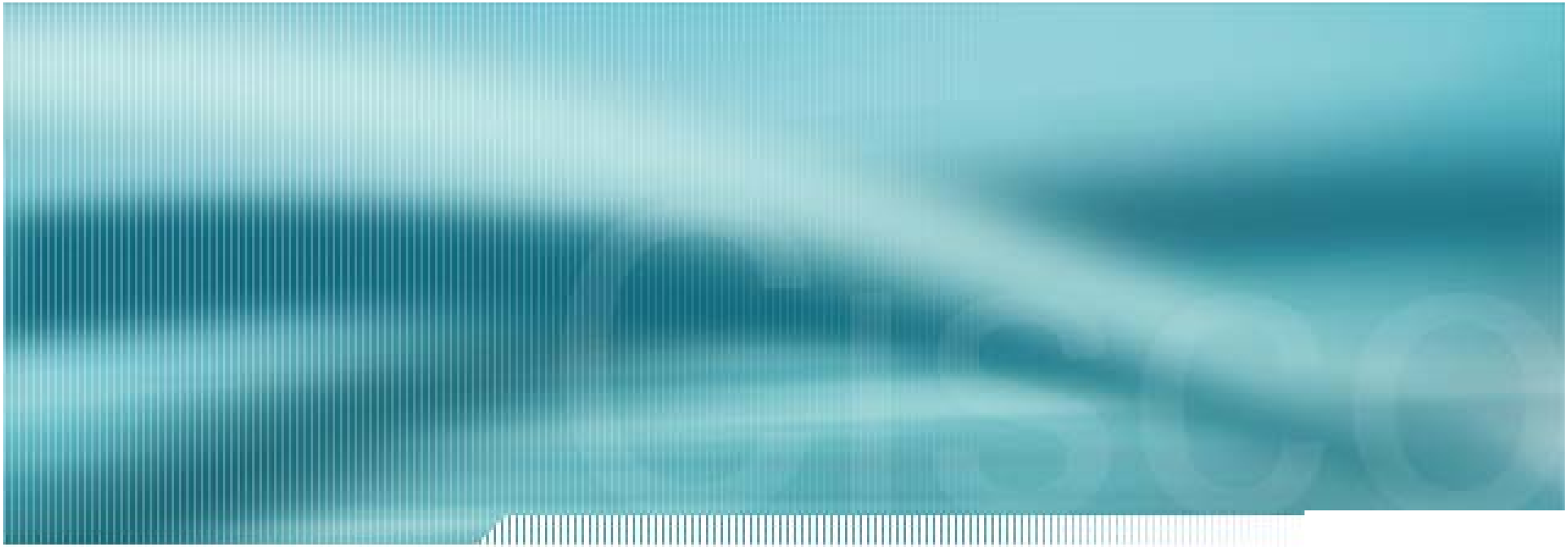

MAC service interface 2 items to consider

Hugh Barrass (Cisco Systems)

Agenda

- **MAC service interface:**
  - **802.3 & 802.1**
- **MAC control, PAUSE**
  - **Update for CM**

Incoming frame – 802.1

M_UNITDATA.indication (

frame_type,

destination_address,

source_address,

mac_service_data_unit,

user_ priority,

frame_check_sequence

)

Outgoing frame – 802.1

M_UNITDATA.request (

frame_type,

destination_address,

source_address,

mac_service_data_unit,

user_priority,

access_priority,

frame_check_sequence

)

802.1, considerations for 802.3

- Incoming:

  frame_type is always user_data_frame

  user_priority is always set to port default

- Outgoing:

  frame_type; user_priority; access_priority are all ignored

- Certain restrictions for aggregation
- Pause function not covered

Incoming frame – 802.3

MA_DATA.indication (

destination_address,

source_address,

mac_service_data_unit,

frame_check_sequence,

reception_status

)

reception_status is ignored by bridge

Outgoing frame – 802.3

MA_DATA.request (

destination_address,

source_address,

mac_service_data_unit,

frame_check_sequence

)

FCS use is optional – for forwarded frames

Note also, minor differences in definitions

What are these service primitives?

- The service primitive is an abstraction:
  - Represents the transfer of information
  - A single event (for a frame)
  - Does not detail local interaction
  - Operates without time (instantaneously?)
- How does a service primitive complete?
  - When the recipient has all of the data
- But – problems for multi-queue client
  - Can the client change its mind?

Proposed modification

- Add an acknowledgement
  - Recipient of service primitive asserts ACK/NACK
  - NACK denotes primitive has not been taken (yet)
  - If request is withdrawn before ACK, then it will not be actioned
- This allows multiple queue and prioritization
  - Frame may be presented and then preempted
- Also need to specify the pipelining (in MAC)

Agenda

- **MAC service interface:**
  - **802.3 & 802.1**
- **MAC control, PAUSE**
  - **Update for CM**

MAC control interface – 802.3

MA_CONTROL.indication (

opcode,

indication_operand_list

)

Indicates a MAC control state – not frame received

Not defined in 802.1 – implicitly ignored by bridge

Outgoing frame – 802.3

MA_CONTROL.request (

destination_address,

opcode,

request_operand_list

)

Used to generate PAUSE frames

Not defined in 802.1

Proposals?

- MA_CONTROL needs to be mentioned in 802.1
  - Receipt of incoming operation and/or status
  - Generation of egress frame
  - Some discussion of actions & responses
  - 802.3 defined opcodes remain in Clause 31/31A etc.
- Multi-priority PAUSE
  - 8 counters – to extend 8 ingress queues
  - See CM presentations for reasoning

New PAUSE

- Same opcode or new?
  - Seamless backward compatibility?
  - Needs negotiation codepoint (somewhere)
  - Could be special MAC control frames to indicate capabilities after autoneg finished
- Indication operand (pause_status) = array of 8
- request_operand has additional bitmap
  - 8 elements, indicate queues for which pause_time is applied

Very contentious proposal!

- Transmission blocking function could be pushed up to 802.1
  - Keep definition & s/m except the part that stops frames being transmitted
  - MA_CONTROL.indicate is sufficient for interface
  - Basically – make frame stopping optional in this layer, if supported in upper layer: like MAC ifs & MAC control for EPON
- Same external behavior – fits better with per-priority
- Architecturally better (as if that matters)
- Could allow 802.1 to specify “non-pausable” queues
  - Is this good or bad?

Items to discuss

- Add ACK/NACK to MAC service i/f
  - Or other method – or keep the same…
- Add MAC control i/f to 802.1
- Define new per-priority PAUSE
  - Same opcode?
  - Negotiation mechanism?
- Move transmit stopper to 802.1