b - plus

automotive | mobile automation | embedded systems

Cut-Through and Qcc Bridge Delay Model

Christian Boiger
christian.boiger@b-plus.com
IEEE 802 Plenary
November 2016
San Antonio, TX

Current Status of IEEE P802.1Qcc

- The current Qcc draft defines parts of the externally visible behavior of cut-through and managed objects to configure it

- It seems that one major reason to define cut-though in Qcc is to provide bridge delay parameters to create a schedule for Qbv (in case it is used in combination with cut-though bridges)

What's the Problem with Cut-Through in Qcc?

- There are many issues:
  - Cut-through is not supported by the current IEEE 802 architecture
  - Cut-through is not completely defined in Qcc, only a very small part of the externally visible behavior is part of the current draft (this does not guarantee interoperability e.g. behavior of counters, shaper, meter, etc.)
  - The bridge management clause is not the right place to define data plane behavior
  - Cut-through is out of scope of the Qcc PAR
  - Cut-through violates the IEEE 802 architecture and therefore does not comply with the compatibility requirement of the CSD/5Cs
  - Major parts of the work that is needed to support cut-through behavior in the 802 architectural model are not in scope of IEEE 802.1 (e.g. a lot of changes to 802.3 state machines would be necessary)

Why is cut-through not compatible with the IEEE 802 architecture?

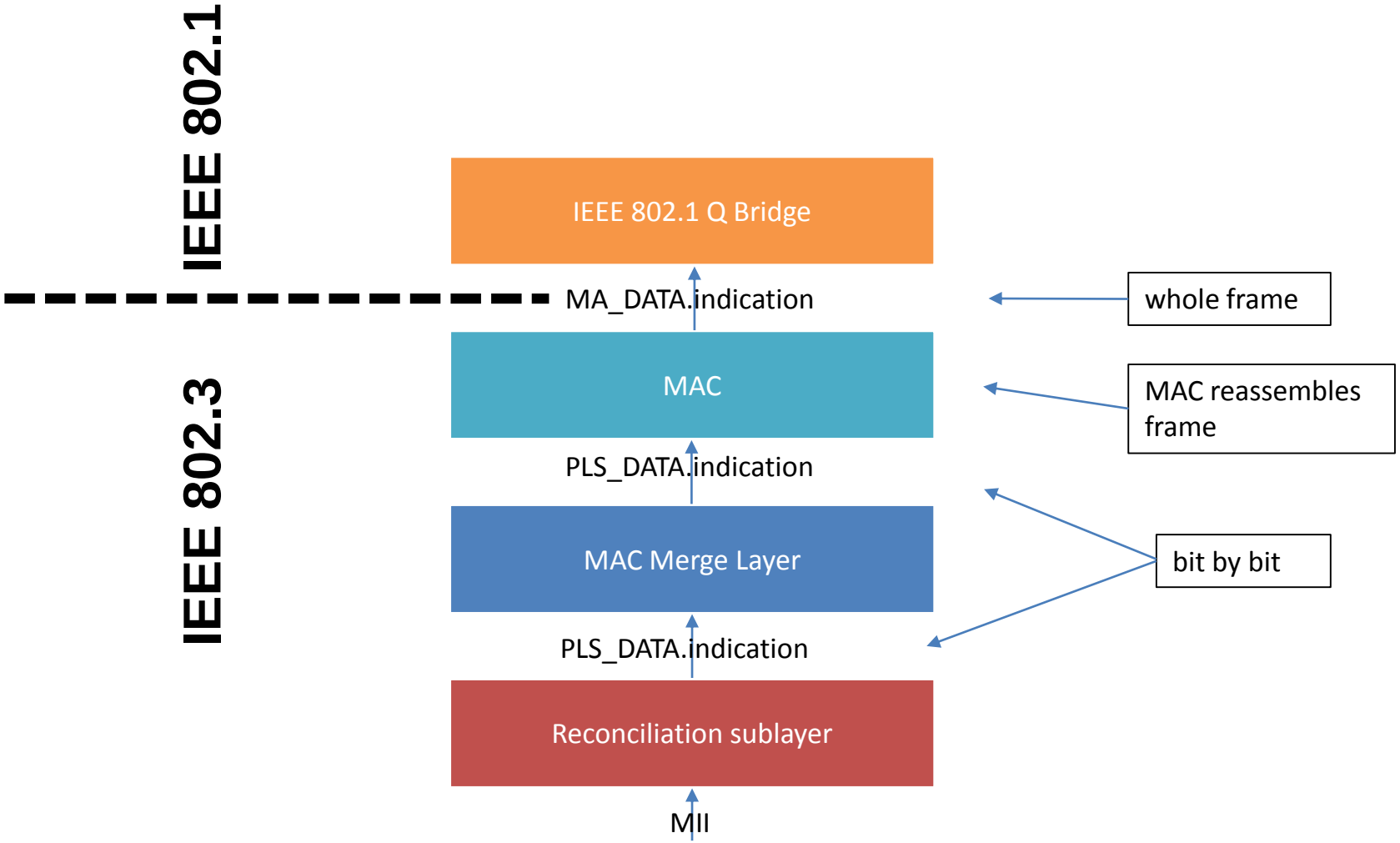
- The interface between 802.1Q and lower layers is described by the MAC service interface, defined in IEEE 802.1AC
- The MAC service interface is defined in a way that a whole frame is presented to the upper or lower layer (not Bits or Bytes)
- The interactions between the two layers are defined as atomic
- Only error free frames issue a M_UNITDATA indication, frames with FCS errors are dropped by the MAC and never pass the MAC relay

  > "All error-free received user data frames give rise to M_UNITDATA indication primitives. A frame that is in error, as defined by the relevant MAC specification, is discarded by the MAC Entity without giving rise to any M_UNITDATA indication"

- IEEE 802.1Q states:

  > "The minimum additional transit delay introduced by a Bridge is the time taken to receive a frame plus that taken to access the media onto which the frame is to be relayed. Note that the frame is completely received before it is relayed as the Frame Check Sequence (FCS) is to be calculated and the frame discarded if in error."

Architectural Model – Receive Path



Conclusions

- Cut-through is not compatible with the current architectural model it requires changes in IEEE 802.1AC, IEEE 802.1Q and IEEE 802.3
- Changing the IEEE 802 architectural model to make it compatible with cut-through operation
  - is complicated
  - would require major changes in IEEE 802.3 and IEEE 802.1
  - would take a very long time until it is standardized (just my guess)
  - is therefore probably not desirable

A possible solution to a part of this problem

Changing the bridge delay model of Qcc

- It might be desirable for people to have standardized managed objects to configure cut-through
- However in my opinion we should not standardize managed objects for non standard features especially not for ones that do not conform to the 802 architectural model
- But it seems to be possible to design the managed objects for bridge delay in a way that also non standard compliant applications can use it (and perhaps also standard compliant with a more complex architecture)
- The current managed objects that report the bridge delay do have very specific implementations in mind (that are probably required in order to support Qbv)
- Why not change the managed objects to support a broader set of implementations, so that it is not necessary to distinguish between cut-through and store and forward?

Bridge Delay in 802.1Q <-> Real Implementations

- IEEE 802.1Q defines an absolute maximum bridge transit delay of 4.0 seconds and a recommended value of 1.0 seconds
- Additionally a minimum transit delay is defined (“The minimum additional transit delay introduced by a Bridge is the time taken to receive a frame plus that taken to access the media onto which the frame is to be relayed.”)
- → this leaves the behavior of implementations quite open
- → compared to this, the bridge delay definitions of Qcc have a more restricted implementation in mind (even if not explicitly stated)

Current Cut-through Definition

- Definition in Qcc: “A Bridge may support the capability of starting egress (transmit) of a frame before the entire frame has completed ingress (receive). This capability is referred to as cut-through.”

- Karl‘s comment on draft 1.0:

  “Cut through definition must include the condition that the forwarding process is independent of frame length.“

- In case the current definition is meant to be relative to the timestamp reference plane it is incorrect

- PHYs can have delays that are higher than a minimum size frame.

- → With the current definition those PHYs would never be able to support cut-through (according to the current definition) which is not true

  e.g. 1000BASE-T1:

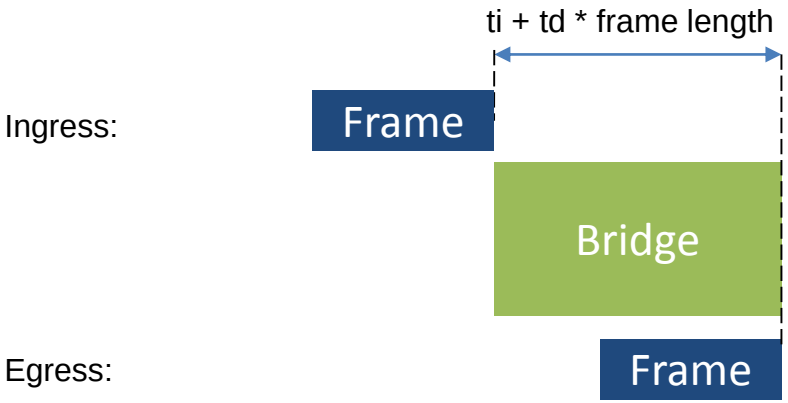
  “The sum of the transmit and receive data delays for an implementation of a 1000BASE-T1 PHY shall not exceed 7168 bit times (14 pause_quanta or 7168 ns)”

Bridge Delay Model

- Only one bridge delay model
- Delay consists of two components, frame length independent and frame length dependent delay component
- Bridge delay is specified as “first bit in to first bit out”
- Four parameters (per port pair)
    - Frame length independent delay (min/max) in ns
    - Frame length dependent delay (min/max) e.g. ps per byte

Example – Store and Forward

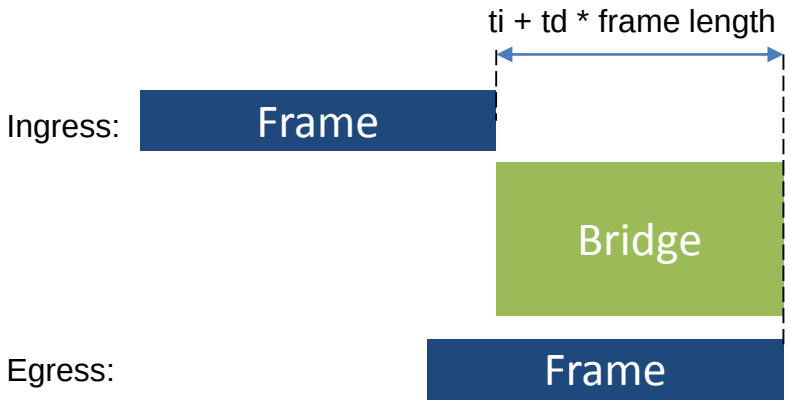
- Store and Forward
- 10 Gigabit links
- storeAndForwardDelayMin/Max (per Qcc D1.1) = 480 ns / 500 ns



- Frame length independent delay component (ti): 480 ns / 500 ns
- Frame length dependent delay component (td): 800 ps/Byte / 800 ps/Byte

Example – Cut-Through

- Cut-through
- 10 Gigabit links
- cutThroughDelayMin/Max (per Qcc D1.1) = 480 ns / 500 ns



- Frame length independent delay component (ti): 480 ns / 500 ns
- Frame length dependent delay component (td): 0 ps/Byte / 0 ps/Byte

Summary

- Cut-through is non standard behavior (so far) and therefore by definition in big parts undefined
- Cut-through breaks parts of IEEE 802 standards and the overall architecture
- People using cut-through need to be aware that this is non standard behavior and they have to think about the consequences and risks (similar if someone uses jumbo frames) → a management clause to configure cut-though inside 802.1Q is not very helpful for this, as it might lead people to the wrong conclusions
- A different delay model might solve part of the cut-through issue without defining cut-through
- A more complex delay model could also help to describe more complex devices
- There might be also better alternative bridge delay models, that achieve the same goal

Thank You