Qbv Small Frame Handling

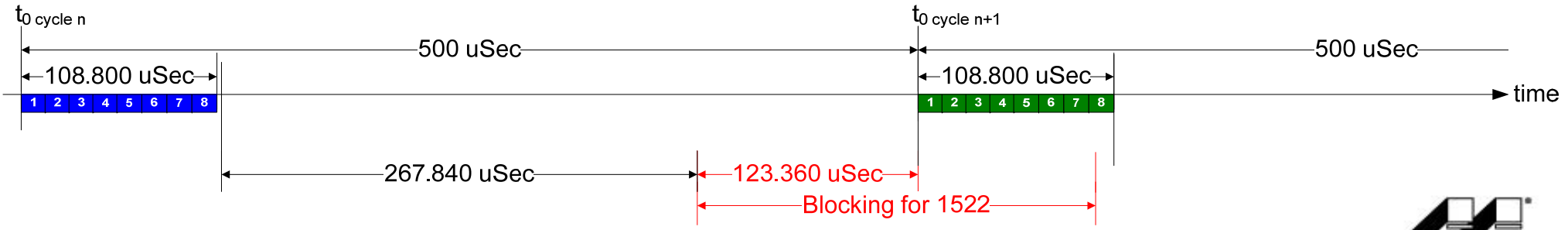
v1 – Sept, 2012

802.1 AVB Face to Face – Santa Cruz, CA

Don Pannell - dpannell@marvell.com

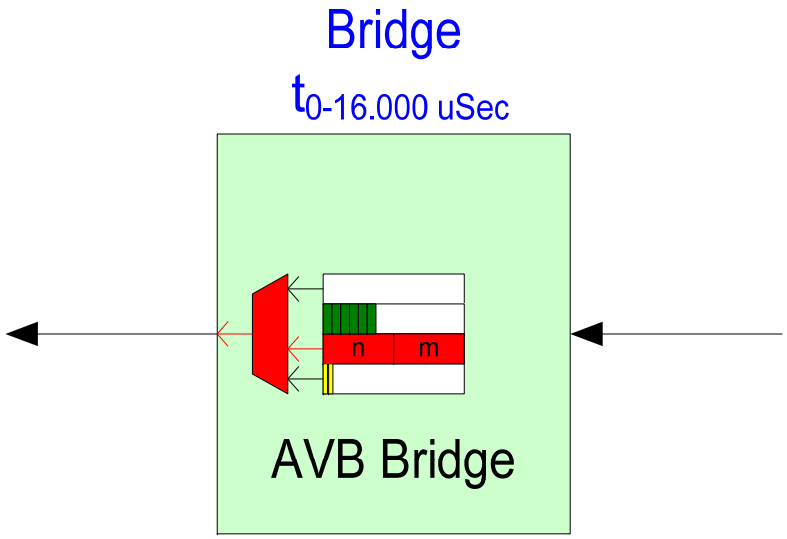
Desire

- Qbv needs a Blocking or ‘Guard band’ before each cycle’s t_0
- In simple implementations this needs to be the size of a Max size frame (1522 bytes or 123.360 uSec at FE)
- If the device is told the start of the ‘Guard band’ (t_0 minus 123.360 uSec at FE), is this enough for smart Qbv Shapers to utilize some of the ‘Guard band’ transmitting less than Max size frames?



Example of the Desire at GE Speeds

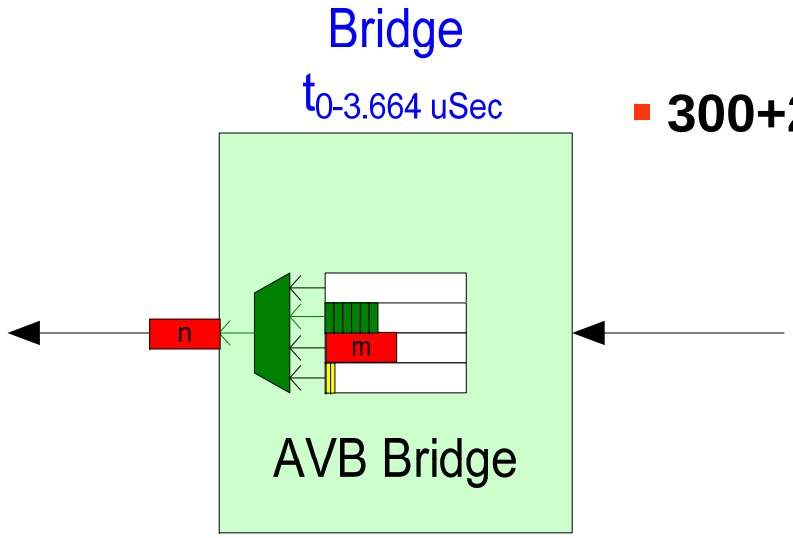
- **Time Progression – Fig 1**
  - **At Bridge $t_{0\text{-}16.000\ uSec\ before\ the\ start\ of\ the\ Cycle}$ the Green Class B frames are being Shaped (gated) by Qav and can't Transmit**
  - **So the Red Max size non-AVB High Priority frame 'n' can start**
  - **At $t_{0\text{-}12.336\ uSec\ before\ the\ start\ of\ the\ Cycle}$ the 'Guard band' starts – this is the 'Don't start any Max size frames' point**
  - **A Max size frame is already transmitting, so finish that frame**



Example of the Desire at GE Speeds

- **Time Progression – Fig 2**
  - **At Bridge $t_{0\text{-}3.664\ uSec\ before\ the\ start\ of\ the\ Cycle}$ the interfering Red Non-AVB frame is done**
  - **Simple Qbv Shapers would stop here**
  - **But smart Qbv Shapers can continue - Now the Green Class B frames are available for transmit - the Credit based Qav Shaper exposed the Green queue with enough credits to burst two frames**

- **16.000-12.336=**

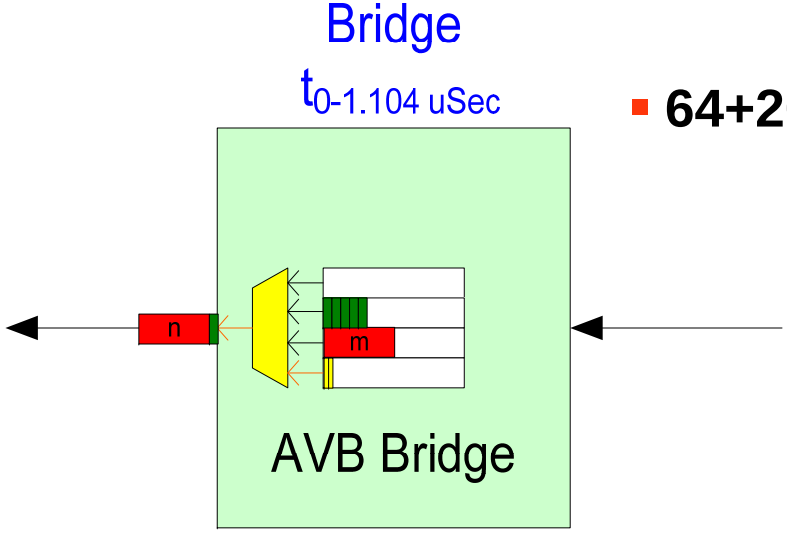


- **300+20 bytes = 2.560**

Example of the Desire at GE Speeds

- **Time Progression – Fig 3**
  - **At Bridge $t_{0\text{-}1.104\ uSec\ before\ the\ start\ of\ the\ Cycle}$ the 1st Green Class B frame is done**
  - **Now the next Green Class B frame has credit to go, but it can't because there is not enough time before t_0 - the start of the Burst Window**
  - **The higher priority Red 'm' frame can't go for the same reason**
  - **But the 64 byte low priority Yellow non-AVB frame can go and does**

- **3.664-2.560=**

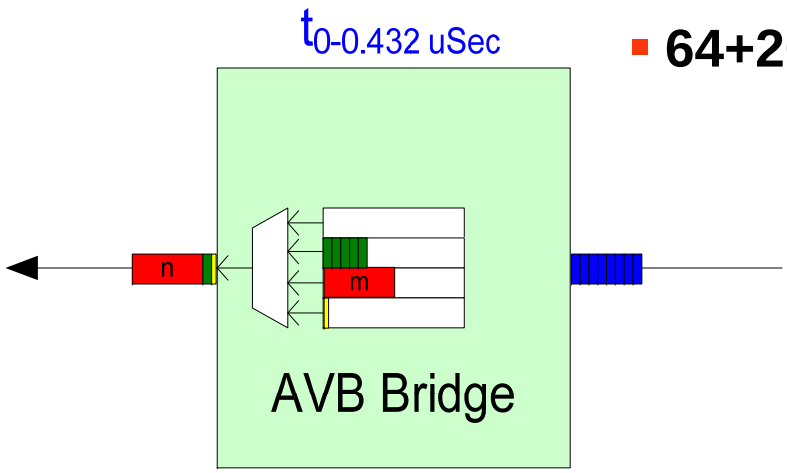


- **64+20 bytes = 672**

Example of the Desire at GE Speeds

- **Time Progression – Fig 4**
  - **At Bridge $t_{0\text{-}0.432\ uSec\ before\ the\ start\ of\ the\ Burst\ Window}$ the 64 byte Yellow frame is done**
  - **The next Green Class B frame has credit to go, but it still can't because there is not enough time before t_0 (its credits are actually increasing)**
  - **Same issue for the high priority non-AVB Red frame 'm'**
  - **The next low priority 64 byte frame can't go either – not enough time**

- **1.104-0.672=**

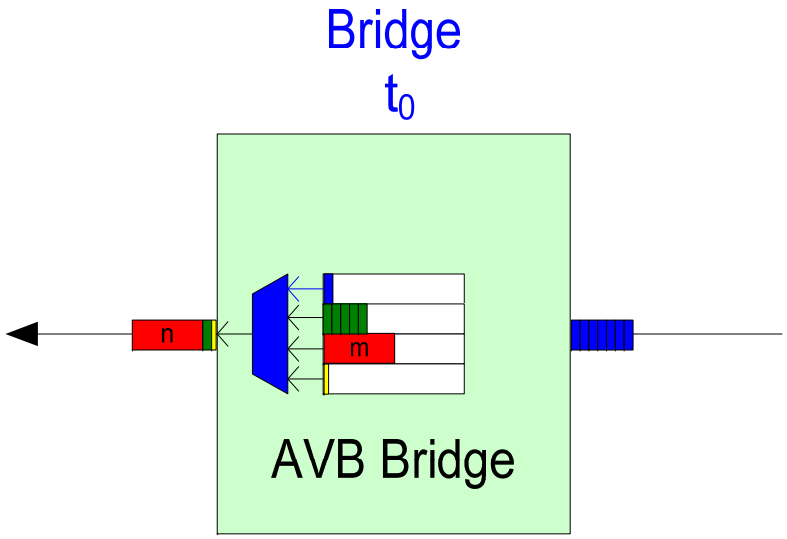


- **64+20 bytes = 672**

Example of the Desire at GE Speeds

Time Progression – Fig 5

- At Bridge $t_{0\ -\ the\ start\ of\ the\ Cycle}$ the port is idle so the newly arrived Blue Class A frames are allowed to egress without any interference!
- A small period after Bridge t_0 the gating on all the non-Class A queues can be released
- The burst of Blue frames will continue since they are the top priority, but as soon as the burst is done the next higher priority frames will go



Problem Question

- If the device is told the start of the 'Guard band' (t_0 minus 123.360 uSec at FE or 12.336 uSec at GE), is this enough for smart Qbv Shapers to utilize some of the 'Guard band' transmitting less than Max size frames?
- Yes
- Even simple Qbv Shapers get to use some of this time – on average 50% as the example showed – since existing frames in transit before the 'Guard band' starts get to use part of the 'Guard band' window
- This bandwidth is not 100% lost

Enhanced Problem Question

- But can smart Qbv Shapers continue to utilize some of the rest of the 'Guard band' by transmitting additional less than Max size frames?
- Yes, but how?

Enhanced Problem Solution

- Program the smart Qbv Shaper with the size of the Max size frame that was used to set the 'Guard band'
- At the start of the 'Guard band' (when simple blocking would start) the Qbv Shaper's 'Size Limit' is loaded with this Max frame size
- At the Tx byte rate of the egress port, the port's 'Size Limit' is decremented by 1 (1 byte for each Tx byte time)
- If an available 'head of line' frame is smaller than the current 'Size Limit' (minus additional Overhead) it is allowed to be transmitted – as it will finish before t_0
- The Overhead needed may be different per design & is needed to insure the port is idle at t_0