



# Resolving ETS Configuration Inconsistencies

Joe Pelissier

January, 2010

az-pelissier-ETSConfigInconsistencies-0110

# What we have today:

- Three tables are involved in the configuration of ETS:
  - A Transmission Algorithm Selection (TAS) table that specifies the Transmission Selection Algorithm per Traffic Class
  - A Traffic Class Grouping (TCG) table that:
    - Specifies the Transmission Selection Algorithm of being ETS or not ETS
    - If ETS, specifies a Traffic Class to Traffic Class Group Mapping
  - A table that specifies the ETS Bandwidth per TCG
    - Every TCG gets an allocation even if no traffic classes are currently assigned to it
- The problem
  - The first two tables are interdependent. The specification today inadequately specifies how setting a parameter in one table affects the other
  - With DCBX, if one side is willing and observes that the other end is not ETS (i.e. its TG15), there is currently no way to indicate what the local side should do
- Two possible solutions:
  - Specify the effects that programming one has on the other
  - Remove the interdependencies

# Specifying effects

- Here is one possible set of rules:

In the TAS table:

If a TC is changed from Strict to AVB or from AVB to Strict, then no change to the TCG table

If a TC is changed from Strict to ETS, the TCG table is modified to move that TC from TCG15 to TCG0

Implies the admin would in a second step reassign the TC from TCG0 to the desired TC if it was not TCG0

If a TC is changed from ETS to Strict or AVB, the TCG is modified to move that TC from TCG0..7 to TCG15

In the TCG table:

If a TC is changed from TCG0..7 to TCG15, the TAS table is modified to change the TC's algorithm from ETS to Strict

If a TC is changed from TCG15 to TCG0..7, the TAS table is modified to change the TC's algorithm from Strict or AVB to ETS

In DCBX, if the local side is willing, the remote side is not willing, and the local side has some TCs assigned to AVB, those TCs will not change

(In other words, even if you are willing in general, you are never willing to change a priority assigned to a TC that is using AVB)

# Specifying effects

- Some thoughts (my opinions, your mileage may vary ;-)

In general, these interdependencies increase the complexity of configuration

It is desirable to avoid these interdependencies when possible

In DCBX, there is currently no way for one end to indicate to the other that it has AVB enabled on some TC. Therefore if the local device indicates willing and the remote device indicates not willing, it will appear to the remote device that the local device failed to configure itself properly

This is potentially OK from a DCBX point of view since DCBX takes no action to verify this. It would, however, look like a failure to someone examining the MIBs

# Removing the Interdependency

- Leave the TAS table and the ETS bandwidth table as is
- Eliminate TCG15. In the TCG table, TCs may be assigned to TCG0..7
  - If the TC is not currently utilizing ETS, then the assignment to the TCG has no effect
    - Similar to the ETS bandwidth table for a TCG that has no TCs assigned to it
    - Similar to the TCG table for a traffic class that has no priorities assigned to it
- Changing a value in the TAS table does not cause a change in the TCG table, and vice-versa
- In DCBX, create a TLV and corresponding MIB objects to express the TAS table
  - Utilize symmetric attribute passing
  - The TLV would contain a willing bit and eight 4-byte values representing the algorithm for each TC.
- IMHO, this is less confusing and results in less chance of configuration error