802.1Qcc:
Time Sync at User/Network

Rodney Cummings

National Instruments

Agenda

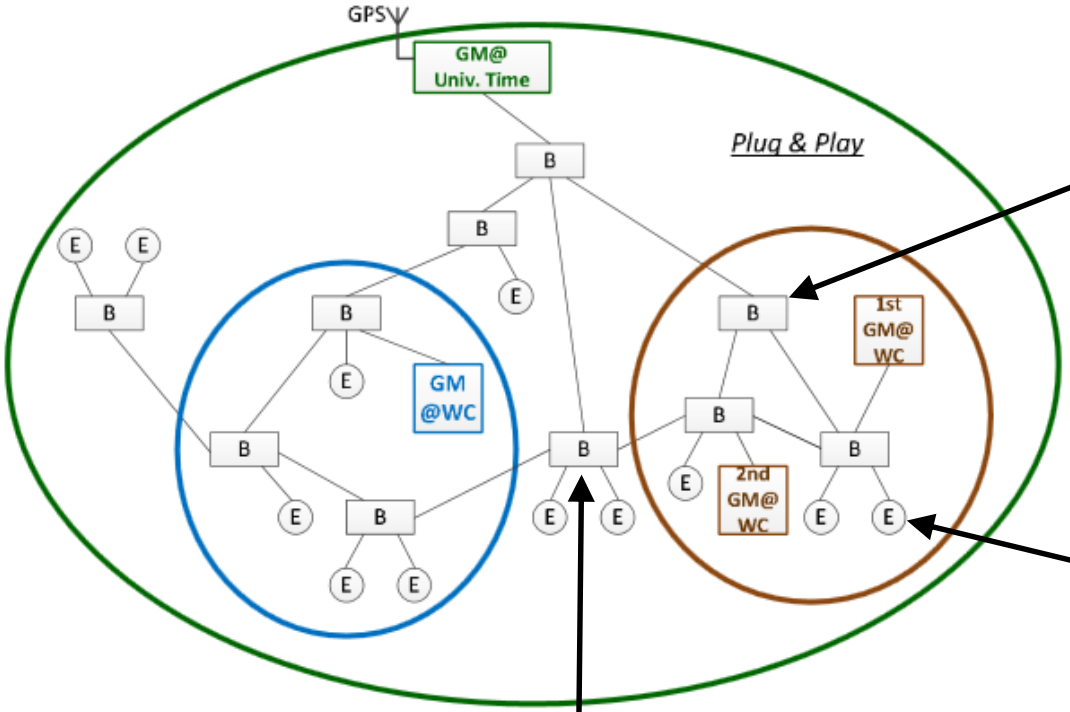
- Problem
- Summary of Qcc UNI concepts
- Discussion to prepare for solution

Problem

802.1AS-Rev / 1588-Rev Features

- Working clock (ARB) and Universal clock (PTP)
- Tools for redundant paths and GMs
  - Each domainNumber is independent
    - Similar to a VLAN ID
  - externalPortConfiguration=TRUE
    - Disables BMCA algorithm for domainNumber
    - Each portState set externally
      - Typically management, but could be another protocol (e.g. IS-IS)

Problems with Working Clocks

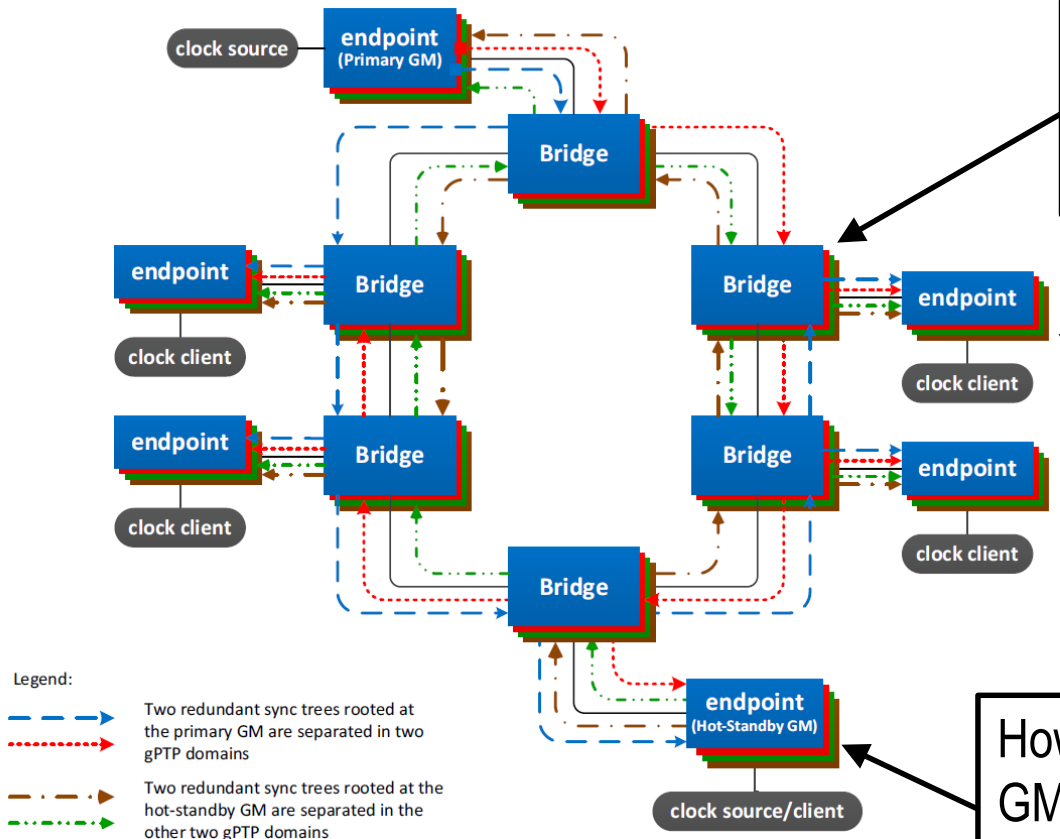


Does Qbv/Qch in this bridge use universal or working clock?

How do end stations know domainNumber?

If this bridge supports 802.1AS, how do we separate the working clock domains?

Problems with Redundancy



How does Qbv/Qch know to use redundant domainNumbers?

How do end stations know the combination algorithm to use?

How do we decide the GMs and their order (e.g. primary)?

Problem Summary

Ideally we can provide a higher-level domain
over which applications using time (users) can agree on:

- Do I want universal clock or working clock?
- Where do I want my application's time to run?
  - Ability to constrain the size of a working clock domain
- What sort of redundancy do I require (paths and GMs)?
  - What combining algorithm(s) do I support?
- To avoid confusing usage of "domain", these slides use

  TimeRealm

  as a temporary term for this concept

Summary of Qcc UNI Benefits

Benefit: User / Network Separation

- Application (user) shouldn't care about network details

- DataFrameSpecification
  - Here is frame I would use without AVB/TSN
- NetworkRequirements
  - Here is the latency and redundancy that I require

- Okay network... go make it happen

Benefit: Protocol Independence

- Design for protocol evolution
  - E.g. MRP to MRP-replacement
- Support multiple configuration models
  - Fully-centralized to fully-distributed
- Avoid forcing unrealistic protocol mandates
  - E.g. "End stations shall run my management protocol"
  - E.g. "Bridge/router shall run my application protocol"

Benefit: Configure Network in User

- The user's box contains network-specific features
  - E.g. Network interface of end station has TSN features
- InterfaceCapabilities
  - User tells network: Here's the stuff my interface supports
    - I don't know how to use this stuff... you tell me
- InterfaceConfiguration
  - Network tells user: Configure your stuff this way
- Retains the clean user/network separation

Benefit: StreamID Concept

- Specified in AVB and continued in Qcc (TSN)
- MAC address plus 16-bit Unique ID
- Uniquely identifies the object (stream) in the network
- Used by application layer users (not only the network)
  - Provides a way to know that we are using the same object
  - E.g. IEEE 1722 Talker and Listeners agree on the stream data format using the StreamID

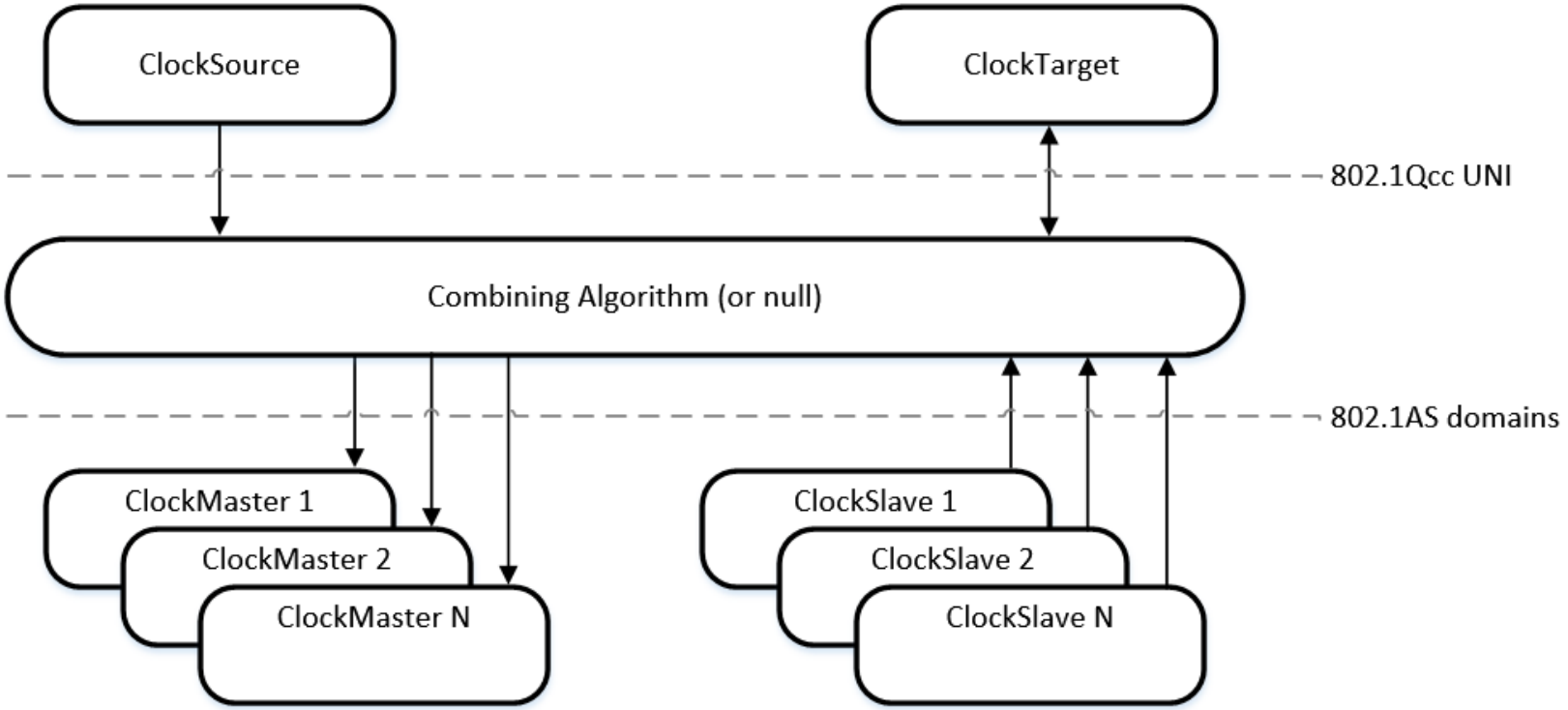
Discussion to Prepare for Solution

Assumptions (1 of 2)

- The benefits of Qcc for data apply to TimeRealm
  - Qcc's UNI can serve as an 'API' for 802.1AS tools
  - UNI applies to use of working clock or redundancy
    - Optional when Universal-only (BMCA-only)
- Qcc enables re-use of protocol roadmaps
  - E.g. Configuration of time redundancy consistent with data redundancy (802.1CB)
    - Don't force end stations to implement management server

Assumptions (2 of 2)

- "Combining algorithm" is network, not user
  - Multiple domainNum are analogous to VIDs and MPLS labels
  - Each 802.1AS system has one (or more) of...



Concept: TimeRealmID

- Analogous to StreamID
  - MAC address and 16-bit unique ID
- Coordinate a single application's TimeRealm
  - Users: Agree on our application requirements
  - Network: Agree on combining algorithm and domainNumbers
- Multiple TimeRealmIDs in network
  - E.g. One ID for IEEE 1722, 2nd ID for OPC-UA, etc
- TimeRealmIDs share time sync resources
  - If requirements are compatible, share domainNumbers
    - Similar to how streams can share a priority and shaper

UNI for TimeRealm

- TimeRealm group: user to network
  - TimeRealmID
  - EndStationInterfaces: Ports in my user that support time sync
  - Requirements: What do I want?
  - Capabilities: What network stuff can I do?
- TimeRealmStatus group: network to user
  - TimeRealmID
  - StatusInfo: ready or fail
  - Configuration: Network configuration for each interface

TimeRealm.Requirements: Assume

- PtpTimescale (boolean)
  - TRUE = PTP (universal clock), FALSE = ARB (working clock)
- NumRedundantTrees (uint8)
  - Num maximally disjoint paths from GM to each slave
  - End-user installed equipment to meet this goal
  - This tells network to go configure it
- NumRedundantGMs (uint8)
  - How many redundant GMs does application require?

TimeRealm.Requirements: Questions

- Who selects the GM, user or network?
  - Fully-distributed: Network
    - All end stations run 802.1AS BMCA and/or PCR4Sync
    - Those protocols pass ClockSource info to network for GM decision
      - E.g. parentDS and timePropertiesDS to carry in Announce
  - Centralized network: User
    - GM passed down as requirement
      - Including whether primary or standby
    - Presumably from the end-user (not an application protocol)
      - Similar to redundant trees/GMs... I know what I installed
      - Many 1588 end-users do this in practice (GM's priority1=0)
  - Recommendation: Support both
    - Group consensus:

TimeRealm.Capabilities: Assume

- MaxDomainNumbers (uint8)
  - How many domainNumbers do I support per interface?
  - All can run as slave; For master see previous slide
- CombinationAlgorithmList
  - List of combination algorithms supported by all interfaces
  - Element is a 24-bit OUI and 8-bit OUI-defined type number
    - Analogous to sequence type of 802.1CB

TimeRealmStatus.Configuration: Assume

- CombinationAlgorithm (uint32)
  - Algorithm to use for all interfaces
- InterfaceList
  - List of configuration for each interface
  - Each element is
    - SlaveDomainList: List of uint8 domainNumber
      - Slave domainNumbers received on this interface
    - MasterDomainList: List of uint8 domainNumber
      - GM domainNumbers transmitted on this interface

TimeRealmStatus.Configuration: Questions

- State machines for GM integration: Not applicable to UNI
  - E.g. When and how do primary/standby GM work?
  - Specified by standard for combo algorithm, not UNI
  - Recommendation: Yes
    - Group consensus:
- Can Master/SlaveDomainList change while running?
  - E.g. Redundant GM(s) added from new TimeRealm
  - Recommendation: Specified by combo algo, not UNI
    - Group consensus:

ClockTarget in Bridges: Question

- Bridges can contain a ClockTarget (time application)
  - Qbv, Qch, etc
  - Not a user in the traditional sense
    - Only requirement is a consistent configuration in all bridges
      - Same domainNumbers, same combo algorithm, etc
  - Multiple TimeRealms use it, so not applicable to UNI
- How do we configure ClockTarget(s)?
  - Recommendation: Specify managed objects
    - For each ClockTarget, domainNumbers and combo algo
  - Group consensus:

Next Steps

- How do we specify the time sync UNI?
- Recommendation: In Qcc, analogous to CB config
  - Draft of preceding slides in Qcc UNI (clause 99)
    - TLV and YANG
  - Phase 1: Map to SRP
    - SRP: Add as optional TLVs to MSRPv1
    - In practice, used with MRP External Control (centralized network)
- Phase 2 (fully distributed) is post-Qcc
  - Map to protocol(s) from new PAR(s)
    - E.g. PCR4Sync
- Group consensus:

Thank You