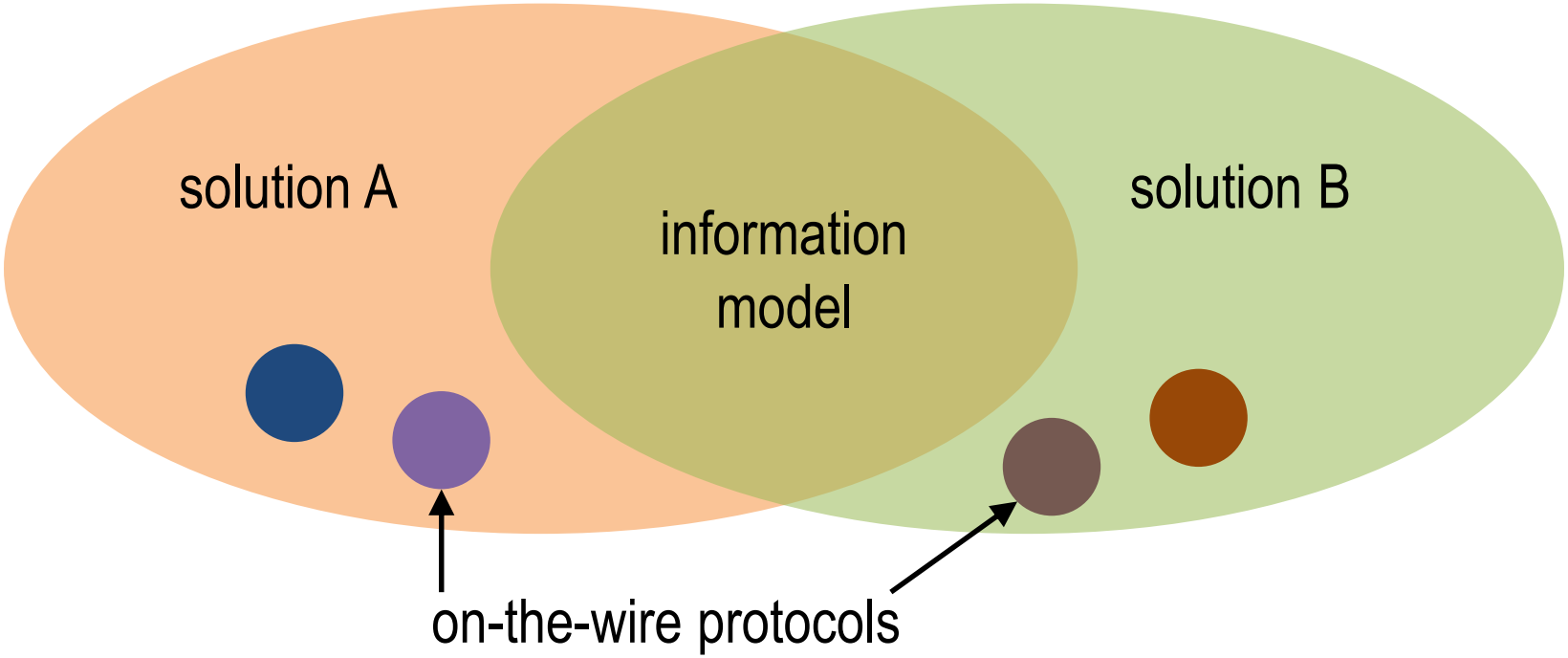
TSN Configuration Roadmap (proposed "what's next?")

Rodney Cummings
National Instruments

Introduction

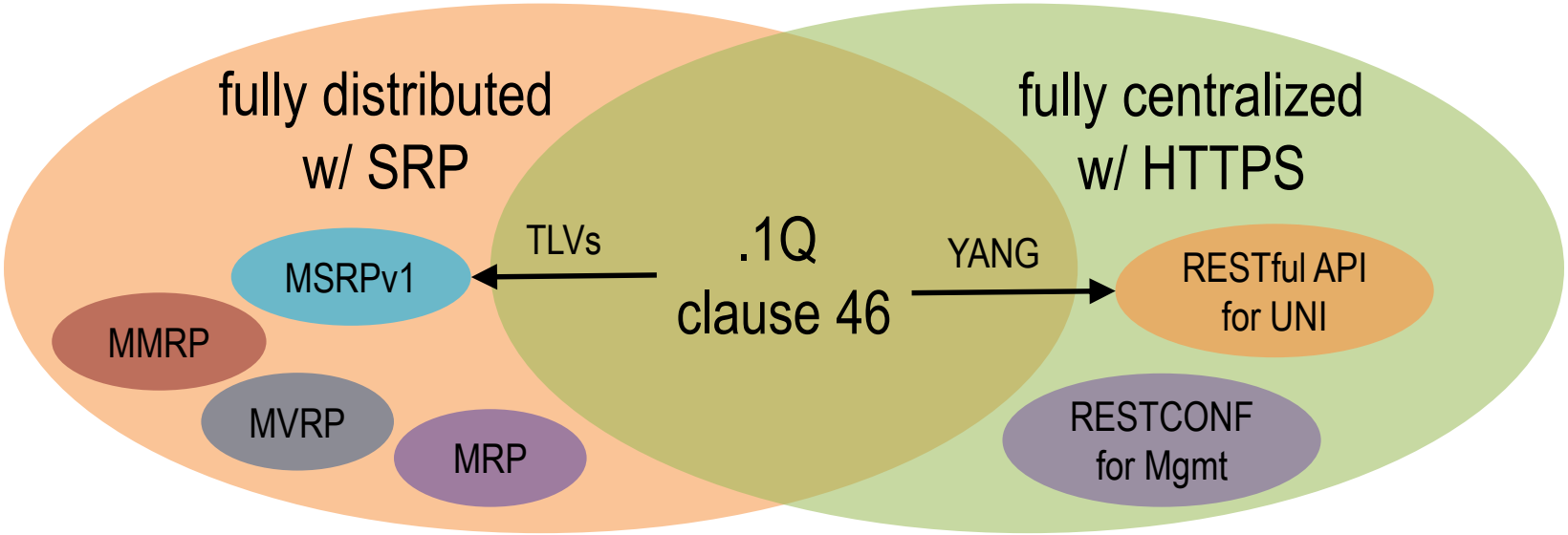
- P802.1Qcc moves TSN config forward in major ways
- We have identified gaps during the .1Qcc project
  - Successful standards evolve, so that is expected
- Since .1Qcc is nearing completion,
  gaps will be addressed in future project(s)
- These slides propose goals for future project(s)
  - Not intended as an 802.1 PAR proposal
  - Presenter believes that 802.1 PAR should be preceded by draft-ready proposals

High-level View of TSN Configuration



- Solution (A, B, etc): Interoperable set of specifications
  1. Select from existing on-the-wire protocols (control-planes)
  2. Create a solution-specific data model by enhancing info model
  3. Map data model onto selected protocols

Two Example Solutions Using .1Qcc



- Technically speaking,
  clause 46 is two data models with an implied information model

Overall Assumptions (.1Qcc & Future)

- Multiple solutions can co-exist in same network
  - Solution represents protocol choice, but protocols co-exist
- All industries, applications, and layers should work toward a common TSN / DetNet architecture
  - Will improve market momentum and usability
- Common 'API' for end stations
  - Easier for non-TSN applications to transition to use TSN, and not worry about 802.1 / IETF details
  - Clause 46 of .1Qcc was designed toward this goal

IETF DetNet Assumptions (1 of 2)

- Let's assume TSN and DetNet share 'hardware stuff'
  - Queuing (shaping/scheduling), policing, data redundancy
- That leaves three major differences
  1. Data plane: Routing of packets inside network (e.g. MPLS)
     - DetNet WG is working on RFC to select among IETF options
     - Once selected, .1CB (or analogous RFC) can integrate
     - No problem
  2. Protocols: IETF has its deployed control planes (e.g. RSVP)
     - As discussed, multiple protocol solutions can co-exist
     - E.g. Slide 7 of http://ieee802.org/1/files/public/docs2017/new-kiessling-RAP-poposal-and-features-0517-v01.pdf
     - No problem

IETF DetNet Assumptions (2 of 2)

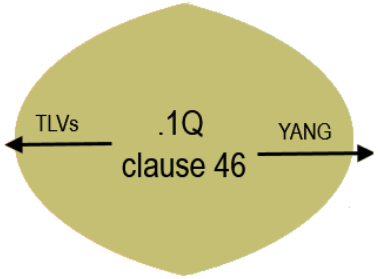
- That leaves three major differences (continued)
  3. Terminology: Different terms for same concept
     - E.g. Node not station, packet not frame, source not talker, flow not stream, deterministic not time-sensitive, ...
     - Differences are essential within each standard group
       - Standards participants insist on their terms... no problem
     - To users of TSN/DetNet (APIs, on-the-wire data, source code), these differences are confusing and unjustifiable... problem
       - E.g. What if we create an API in Linux / Windows / Mac?

       Are there two APIs, exactly the same except for terms?
       Hopefully the answer is No

       Subsequent slide proposes steps to resolve problem

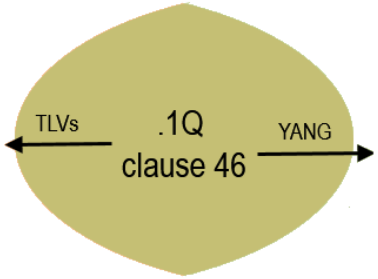
Gaps

Clause 46 (1 of 3)



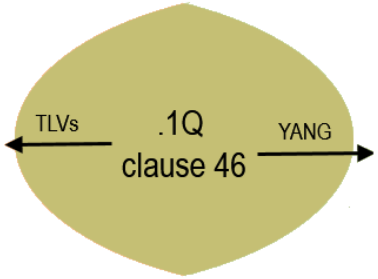
- Gap: Consensus that we want a single common information model
  - One 'API' for UNI
  - Common to TSN and DetNet
    - draft-farkas-detnet-flow-information-model works in this direction
  - Common as subset of data model for .1Q MRP & .1CS LRP
    - "RAP" is really MSRPv2 on LRP
  - Applied to multiple control-plane solutions
    - Info model does not contain control-plane specifics

Clause 46 (2 of 3)



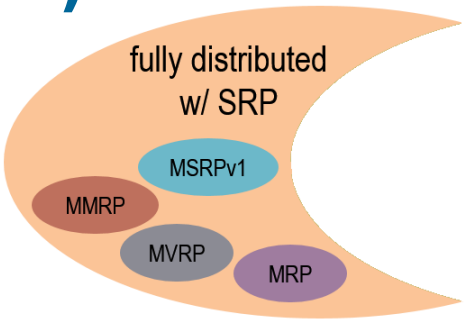
- Gap: Explicit information model
  - Requested in comments to .1Qcc
  - Re-organize clause 46 as info model
    - Specify as groups of elements, each with type and description
    - Text specifications, not TLV or YANG
    - draft-farkas-detnet-flow-information-model is a start
      - Adding DetNet elements as needed
    - Use common TSN/DetNet terms
  - Move TLV specs to clause 35 (SRP)
  - Move YANG to new clause for fully-centralized (more later)
  - Presenter can provide draft-ready proposal

Clause 46 (3 of 3)

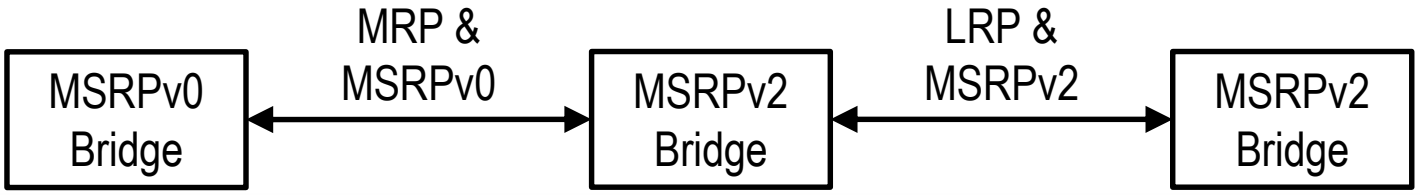


- Gap: Common terminology
  - draft-ietf-detnet-architecture has terms
    - Intended for use by IETF only (not 802.1)
  - Proposed approach:
    1. Raise visibility of shared TSN/DetNet problem in both WGs
    2. Agree on terminology assumptions for info model (at least)
       - Term doesn't need WG prefix (e.g. "node", not "DetNet-node")
       - Each WG has its own terms, but info model uses superset term
         - » E.g. "Relay node can be a bridge, router, firewall, or other..."
    3. Use draft-ietf-detnet-architecture for consensus (DetNet mailing list)
       - Present in 802.1 to solicit comments here
    4. Work out compatible transition for 802.1 TSN
       - E.g. MSRPv0/v1 keeps old terms, but maybe MSRPv2 uses new

Fully-Distributed w/ SRP (1 of 4)

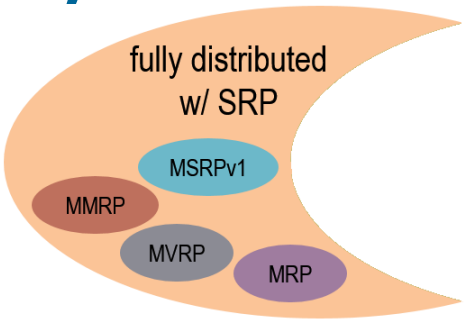


- Gap: Map .1Q applications to LRP (.1CS)
  - Legacy app supports either LRP or MRP
    - MMRP, MVRP, MIRP, MSRP
    - Assumes consistent interface for MRP/LRP
  - MSRPv2 (new features): "Shall use LRP" is okay
- Gap: MSRPv2 is backward compatible
  - Typical 802: Mix of versions no worse than lowest version
  - Bridge with LRP shall also support MRP
  - MSRPv2 translates like .1Qcc did for v0/v1



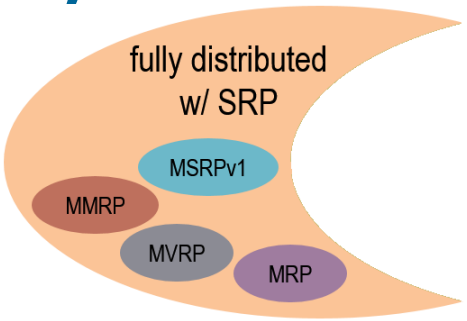
Fully-Distributed w/ SRP (2 of 4)

- (Next four gaps from this and this presentation)



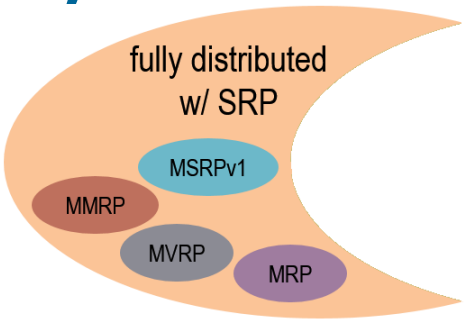
- Gap: MSRPv2: Domain enhancements
  - Negotiate SR class parameters configured by management
  - Question: Why some parameters & not others?

Fully-Distributed w/ SRP (3 of 4)



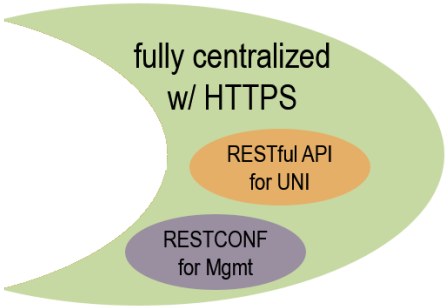
- Gap: MSRPv2: New methodology for latency
  - Management calculates latency-bound for each SR class
    - Using knowledge of topology, each Bridge's shaping, etc
    - Management writes latency-bound to managed objects in Bridge
  - Latency-bound is reported in Domain attribute of SR class
  - New: Talker/Listener requests MaxLatency using UNI, and MSRPv2 uses Domain to find best SR class for that Stream
    - Design paradigm: Domain is ***not*** information model, Domain is a detail of the control-plane
    - Info model maps to each control-plane's data model details

Fully-Distributed w/ SRP (4 of 4)



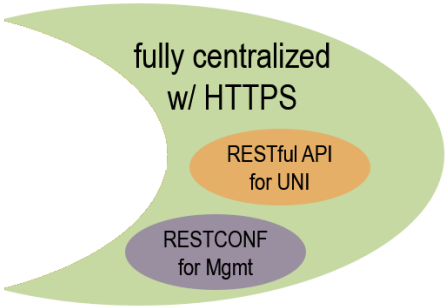
- Gap: MSRPv2: Recent TSN standards
  - E.g. 802.1CB, 802.1Qci, 802.1Qch
  - Question: What is done by management, what is done by SRP?
- Gap: MSRPv2: Other goals
  - "improved distribution", "enhanced diagnostic", "improved reconfiguration on topology change"
- For preceding fully-distributed gaps, volunteer(s) needed for draft-ready proposal

Fully-Centralized w/ HTTPS (1 of 7)

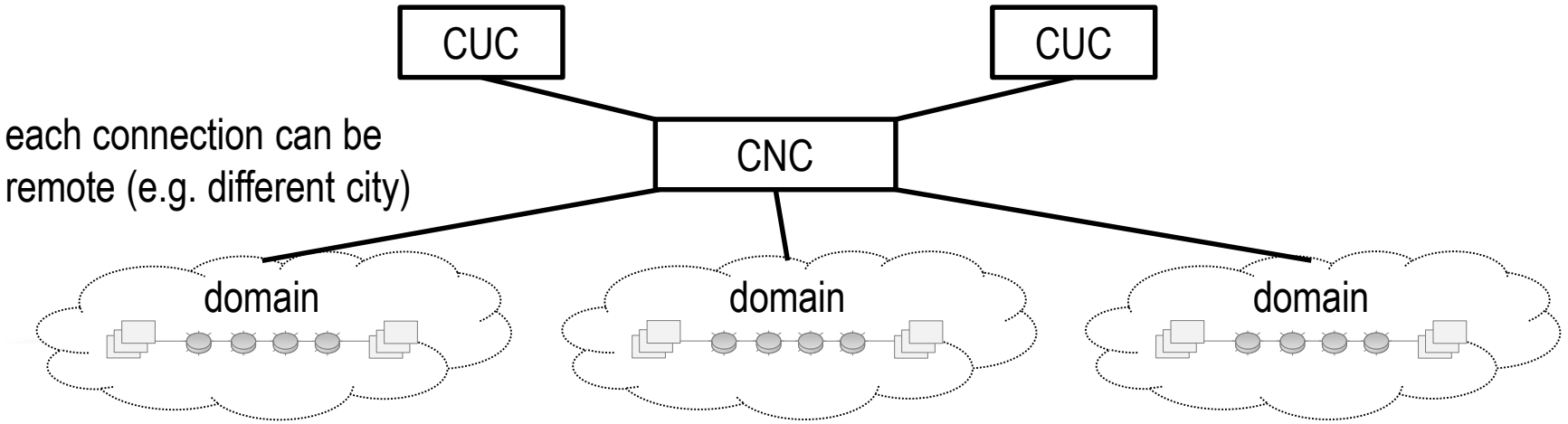


- Gap: Complete YANG module for CNC
  - For the question "Which is best for UNI, RESTful API, or RESTCONF?" answer of "RESTCONF" has become clear
    - YANG provides formal schema; easier to validate than written specs
    - YANG supports actions (RPC functions)
  - Add as a new clause in 802.1Q
    - Completes the picture for interoperability
    - Core YANG text uses .1Qcc's YANG snippets
    - Enhance the YANG module for fully-centralized solution
      - (Gaps on next slides)

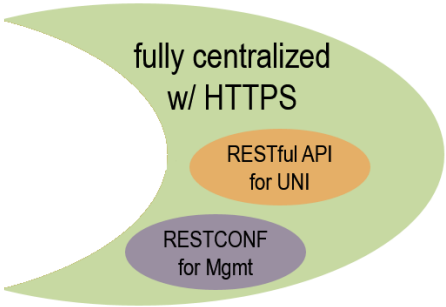
Fully-Centralized w/ HTTPS (2 of 7)



- Gap: CNC YANG: Hierarchy in CNC
  - A CNC supports multiple CUC clients
  - Each CUC can have multiple domains
    - Domain = TSN-capable network with CUC's end stations
  - YANG module in CNC uses this hierarchy
    - RFC6536: Each CUC can control access to Streams in its domain(s)

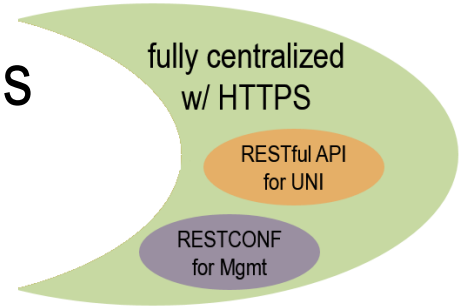


Fully-Centralized w/ HTTPS (3 of 7)



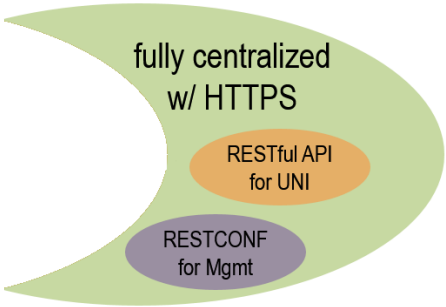
- Gap: CNC YANG: Actions
  - Some CNC tasks can be time consuming
    - Discover physical topology and capabilities
      - LLDP crawl
      - Capabilities = shapers, scheduling, delays, limits, etc
        - » 'Up' from Bridge's management, not 'down' from CUC's Domain
    - Compute reservations
      - Scheduled traffic
      - 'Network calculus': Latency analysis using strict & preemption
    - Configure reservations in Bridges (after compute)
  - Domain in YANG module provides an 'action' to start task
    - Data nodes in domain provide progress of task

Fully-Centralized w/ HTTPS (4 of 7)

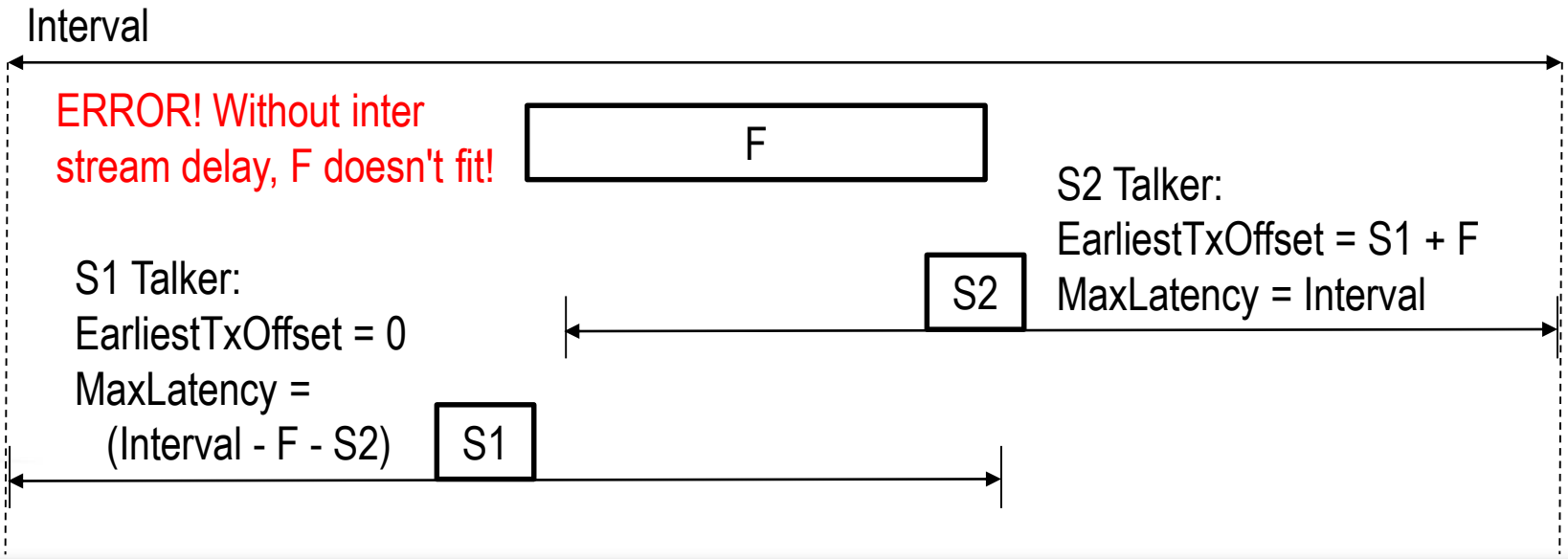


- Gap: CNC YANG: Topology and capabilities
  - Use case: Topology display in CUC
    - Many CUCs do this today, so YANG module provides a way to read domain's physical topology
  - Use case: Offline design
    - Customer designs offline topology in CUC's UI (not physical)
      - User application, Talkers, Listeners, Bridges with capabilities
    - CUC writes offline topology to CNC for 'compute', then reads status
    - When physical network is ready, offline compute re-used for online
  - Augment I2RS topology YANG with LLDP local info, and each Bridge's capabilities
  - CNC module uses TSN topology to enhance for use cases

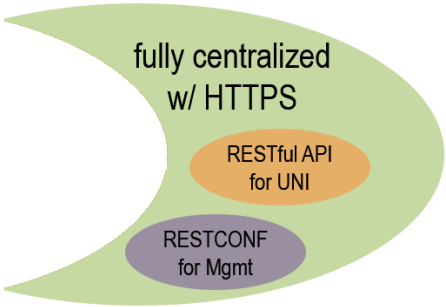
Fully-Centralized w/ HTTPS (5 of 7)



- Gap: CNC YANG: Inter stream delay
  - Time-aware function F & Streams S1 and S2 :
    S1 → F → S2; F has known WCET;
    CUC can locate F anywhere in Interval
  - Inter stream delay: Min time from end-of-S1 to start-of-S2



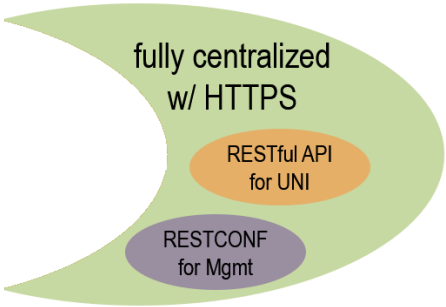
Fully-Centralized w/ HTTPS (6 of 7)



- Gap: CNC YANG: Time sync
  - Time sync is a critical component of TSN
  - Info model integration can be profile-independent
    - TSN/DetNet supports any 1588 profile
  - Use case: Use of 1588-2018 externalPortConfiguration
    - Configure GM and active topology from controller (CNC)
    - Enhance UNI's InterfaceCapabilities / InterfaceConfiguration concept
      - End-station→CNC: Capabilities like GM timePropertiesDS
      - CNC→End-station: Configuration like portState (GM or slave)

Fully-Centralized w/ HTTPS (7 of 7)

- Presenter can provide draft-ready proposals for preceding fully-centralized gaps



- Gap: CNC YANG: Confederation of multiple CNCs
  - Described in last slide of recent presentation
  - Volunteer needed for draft-ready proposal

Thank You