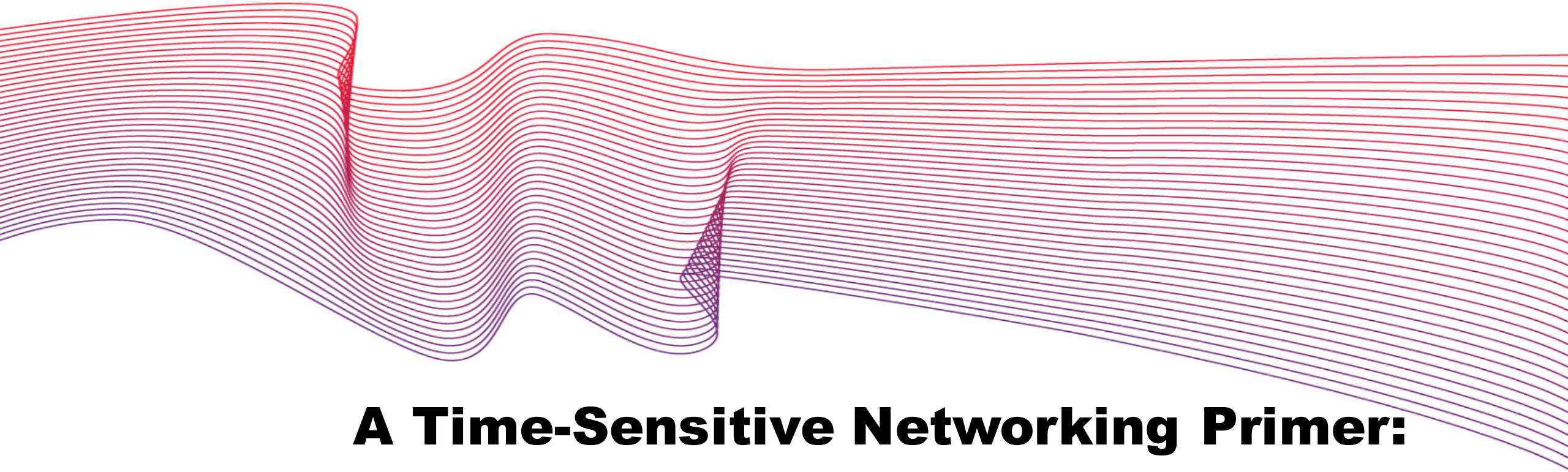


A Time-Sensitive Networking Primer: Putting It All Together

Michael Johas Teener

Senior Technical Director and Plumbing Architect

mikejt@broadcom.com

Agenda

- **802 networks, the internet and "best effort" …**
  - And why it isn't always good enough
- **Objectives for time-sensitive networks …**
  - Live audio and video
  - Control and sensor networks … "Industrial Ethernet" and beyond
- **Time-sensitive networking: now, and improvements coming soon**
  - A new way to meet old goals
  - Time transport
  - Delay (latency) management
  - Resource and path reservation
- **Higher layer IEEE and IETF developments**
- **Where we go from here**

The internet and local area networks

- **The internet and LANs have coevolved**
  - from a bunch of wildly varying link and protocol technologies …
    - remember token ring, ARCNET, LocalTalk, X.25, DecNet, SNA, TP4 …
  - to a common model based on "internet protocols" (IP) running on top of LANs based on the IEEE 802 architecture
- **The driving use case was "business data"**
  - bulk data transfer
  - transactions
- **The important metrics were average delay and average speed**
  - "best effort delivery" was the basic mode of operation
- **Anything more "timely" still used point-to-point connections and circuit switching**
  - or specialized or proprietary specifications
  - e.g. IEEE 1394 ("Firewire"), Profinet, EtherCAT

“Best effort delivery”

- **According to Wikipedia:**

  “does not provide any guarantees that data is delivered or that a user is given a guaranteed quality of service level or a certain priority”

  Hmm … what is “best” about that?

  In practice, it really means:

transfer data as quickly as possible

“best” in this case means “quickest”

“best effort” works!

- **In many, many cases, “best effort” is[1]**
  - in lightly loaded networks
  - where average delay is the primary metric
  - if we can’t, or don’t want to, or it’s too much trouble to differentiate between different classes of traffic
- **But, of course, that’s almost never enough**
  - so we have higher layer services like TCP
  - or we ignore the problem and have audio and video dropouts
- **And “best effort” isn’t[1]**
  - when the worst case delay is the important metric

[1] “best”

Time

- **"Time" means delay**
  - the metric is maximum delay, not average
  - here's where we need something better than "best effort"
- **In addition, "time" means time**
  - wall clock time, synchronization, coordination, phase locking
- **Both bounded delay and a well-known time are required in time-sensitive networks**
  - live audio and video streaming
  - control and sensor networks

Objectives

Control and sensor systems

- **For live audio performances …**
  - the maximum delay between a musician doing "something" and hearing that same "something" is 10ms
  - delay of sound from monitor speakers to the musician, plus DSP delays, plus mixer delays, plus more DSP delays uses up 8ms …
  - so the network gets 2ms for the musician-to-monitor path
- **For speaker arrays …**
  - the maximum synchronization error between speakers must be less than 10us …
  - and, of course, the designers want (and can use) better: down to 1us

Live audio and video networks

- **Both large and small physical extent**
  - Refinery or automotive assembly line: 1 km or more
  - Work cell (robot) up to 5 hops, factory up to 64 hops
  - Homes/offices/autos/aircraft/continents
  - Coexistence of bulk traffic
- **Precise timing**
  - Within the factory ± 100 µs
  - Within the work cell (robot) ± 500 ns
  - "radio over Ethernet" ± 65 ns
- **Deterministic and very small delays**
  - Within the work cell < 5 µs
  - "radio over Ethernet" < 100µs
  - Within the factory < 125 µs (≈ 4 µs per hop)
  - Within the continent < 100ms
- **Safety!**
  - Redundant control / data paths with "instant" switchover
    - Seamless, or at the very least < 1 µs

A new way to implement an old model

- **Back when almost flawless streaming QoS was required …**
  - before the internet and "buffering …"
  - before mobile phones and "can you hear me now?"
  - In other words, when we had land-line circuit-switched telecom networks
- **Network connections were based on "circuits"**
  - nailed-up paths from end to end with deterministic characteristics
- **The internet changed this model to "connections"**
  - highly adaptive, very robust, but timing is very sloppy
  - use the sloppy timing budget as a way to get the data through (retries, adaptive routing, etc)
- **TSN goes back to "circuits"**
  - But we call them "streams"
  - but interoperates with existing models of "the internet"
  - how do we do this?

How to build a time sensitive network

1. **Provide a network-wide precision clock reference**
2. **Limit network delays to a well-known (and hopefully small) value**
3. **Keep non-time-sensitive traffic from messing things up**

- There are many possible ways to do this, but first we need to fix the low-level plumbing, and for networks based on IEEE 802 we use

IEEE 802.1 Time-Sensitive Networking ….

“AVB” and “TSN” … some history

- **AVB is “Audio Video Bridging”**
  - IEEE 802.1 project started in 2005 largely to address the needs of the professional audio market – “IEEE 802.1 Audio Video Bridging Task Group”
    - Originally called “Residential Ethernet”, but that was too limiting and not really in scope for the IEEE 802.3 Ethernet standards
  - But also very, very useful to consumer electronics, professional video, and automotive “infotainment”
    - Associated industry compliance and marketing group called “Avnu”
- **TSN is “Time-Sensitive Networking”**
  - Capabilities of AVB-capable network were also very interesting to other groups
    - Industrial and automotive control and sensing, “IoT” to factories to motorcycles
  - Wider spectrum of requirements, “Audio/Video” was an inappropriate tag

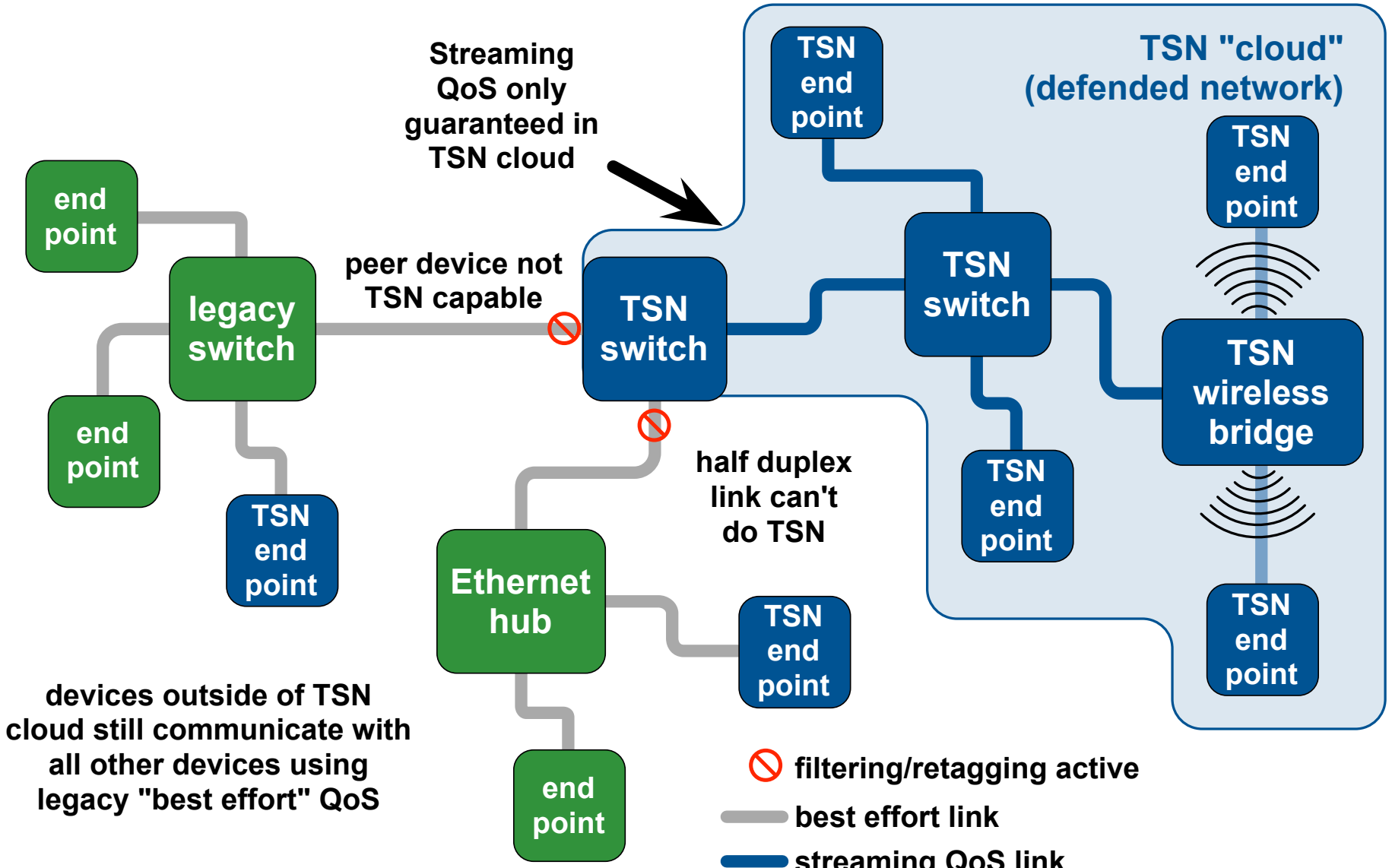
Building the foundation: IEEE 802.1 TSN

- **The IEEE 802.1 Time-sensitive Networking Task Group is "responsible for developing standards that enable time-sensitive applications over IEEE 802 networks"**
  - the IEEE 802.1 Working Group is responsible for bridging (including Ethernet "switches") between LANS
  - interoperability between networks of differing layer 2 technologies
- **The primary projects include:**
  - queuing and forwarding of time-sensitive streams,
    - 802.1Qav credit-based shapers, new P802.1Qbu preemption, P802.1Qbv time-aware queuing, P802.1Qch cyclic queueing, P802.1Qci input gating, and P802.1CB seamless redundancy
  - registration and reservation of time-sensitive streams,
    - 802.1Qat – a distributed "stream reservation protocol", extended in new P802.1Qcc to support preemption, scheduling, centralized control, and interaction with higher layer IETF services
  - time synchronization – IEEE Std 802.1AS (based on IEEE 1588) - and
  - overall system architecture – IEEE Std 802.1BA "audio video systems", P802.1CM fronthaul systems for mobile ("radio over Ethernet") and a new –unnamed- for control systems

Unified layer 2 quality of service

- **Enhance network bridging**
  - Define common QoS services and mapping between different layer 2 technologies
  - IEEE 802.1 is the common technology
- **Common endpoint interface for QoS**
  - "API" for QoS-related services for ALL layer 2 technologies
  - Toolkit for higher layers
    - IEEE 1588 time synch, IEEE 1722 and RTP streaming, IEEE 1722.1, RSVP and SIP session establishment
  - Provide network independence for endpoints without giving up QoS
    - Endpoints don't have to be aware of the particular link technologies (Ethernet, WiFi, EPON, MoCA, powerline, etc.) … common API, support for multiple link types in a path

TSN services only in the “TSN cloud”



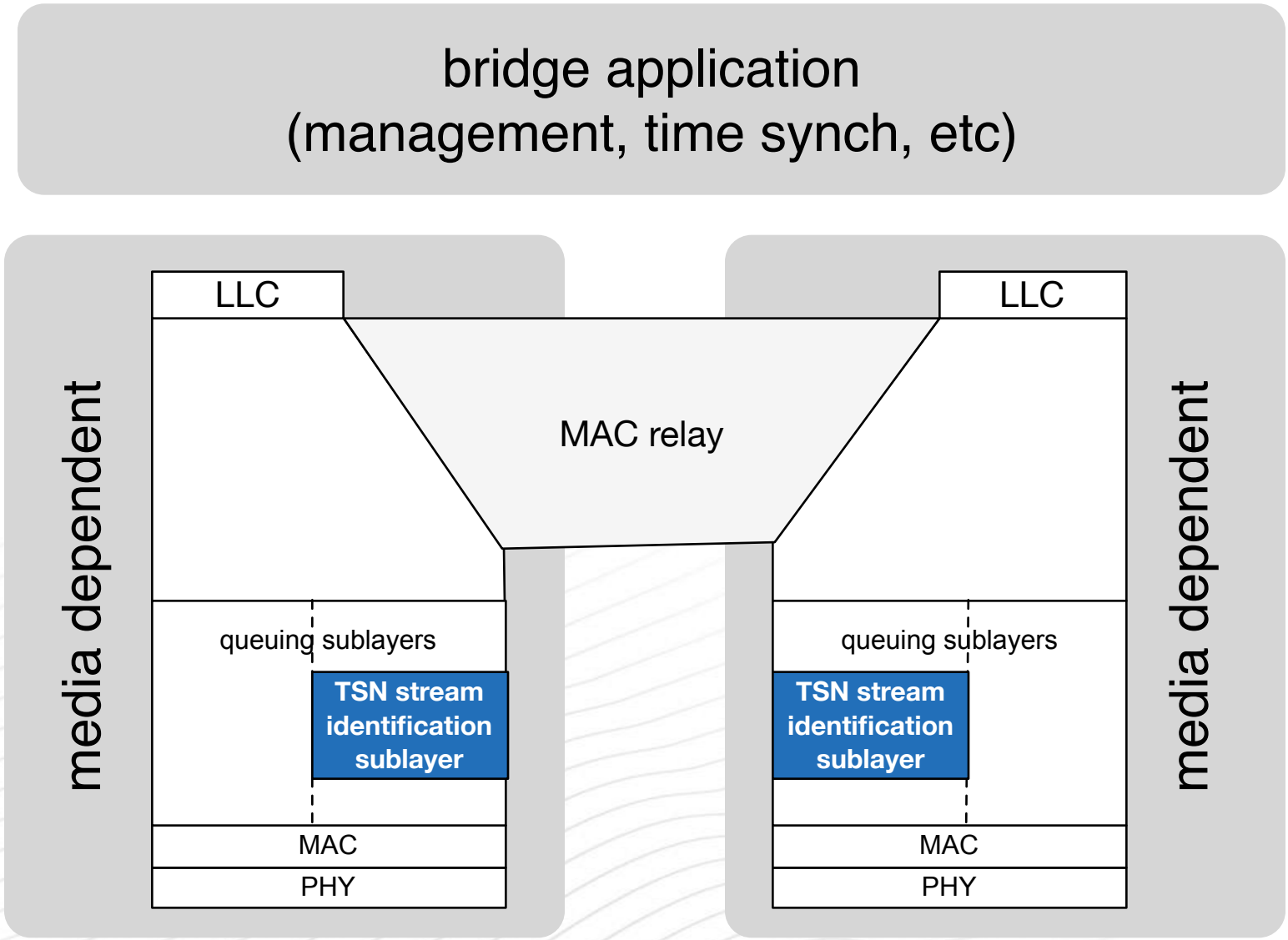
Identifying streams

- **Providing QoS for a stream first requires some way to identify the stream**
- **TSN uses three types of identifying labels:**
  - "stream ID", which is a 48-bit EUI-48 (usually the MAC source address) concatenated with a 16-bit handle to differentiate different streams from the same source
  - "traffic class", which is determined by the 3 priority bits
    - TSN normally only uses one or two classes, by default 2 and 3
  - "stream destination address", which, oddly enough, is the MAC destination address combined with a VLAN ID
    - TSN normally uses locally-managed or group addresses (sometimes called "multicast addresses")

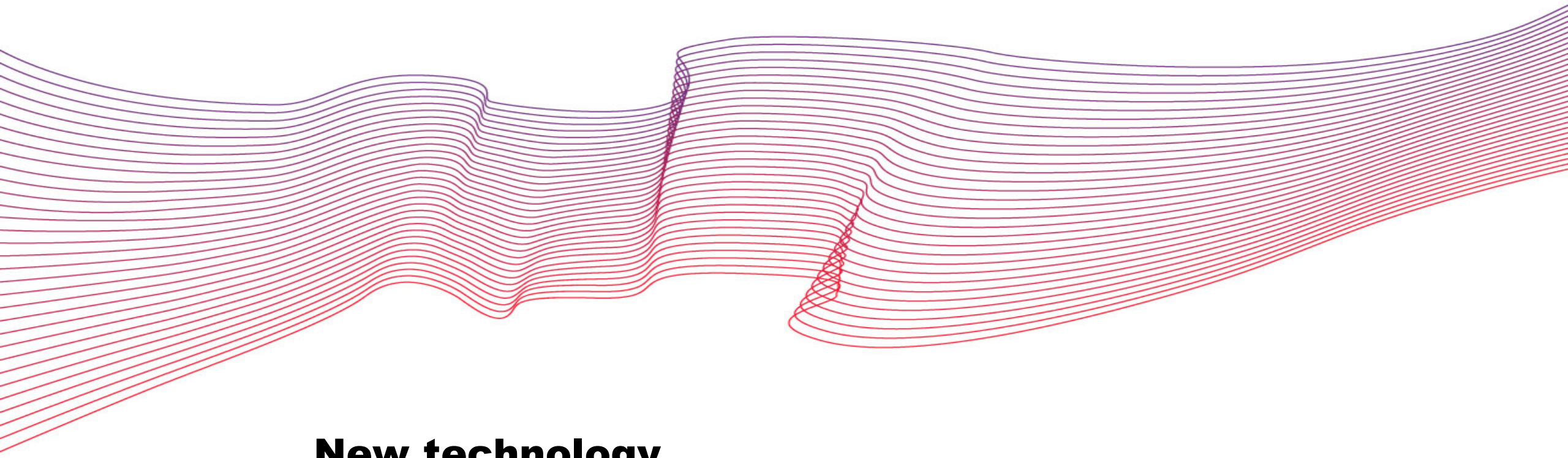
Using stream labels

- **Stream ID is used for management**
  - Unique within a TSN cloud
  - Used by the "control plane" to reserve resources
- **Traffic class and stream destination address identify which "data path resources" are used by a stream**
  - Normally the traffic class and stream destination address stay the same across the network from end to end
    - Not required, however … "circuit" is an abstract concept, and the actual labels used for packets to identify a circuit can change on a hop-by-hop basis
    - A circuit can be labeled by an IP octuple by a transmitter, and relabeled by the originating end station or by a bridge on the network edge.
  - Used by the "data plane" to provide QoS and forwarding (routing) services

Stream identification



- **Stream identification is done in the queuing functions of a bridge or endpoint**
  - Identifies a stream on ingress, and labels it on egress
  - Different ports can use different labels for the same stream
  - Allows "edge" bridge to do QoS proxy services
  - "management" handles this, more later …
- **There are other TSN-related functions**
  - more later …

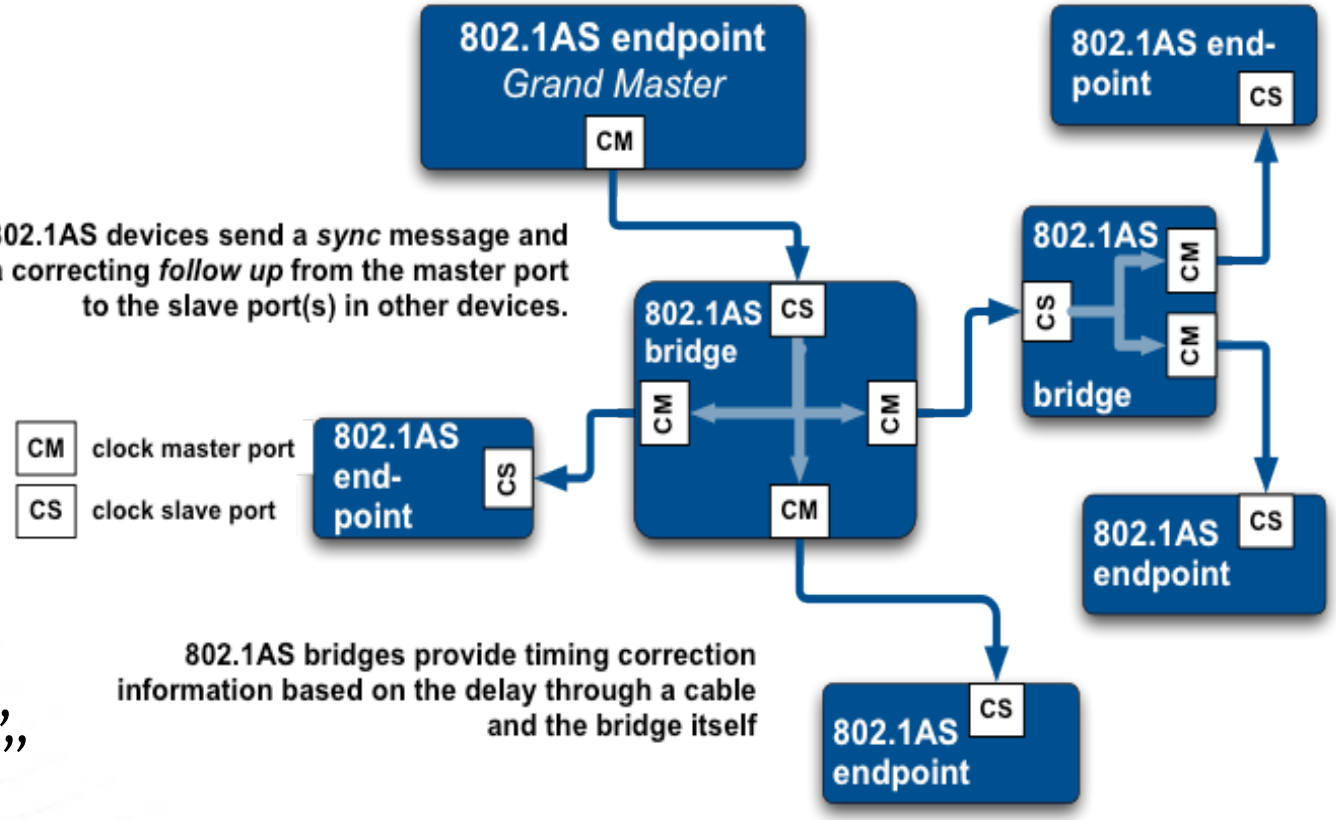


New technology …

Time transport and synchronization

IEEE 802.1AS: Precise synchronization

- **All "time-aware systems" participate in a "native IEEE 802 layer 2 profile" of the IEEE 1588v2 Precision Time Protocol**
  - a very tightly defined subset of standard 1588v2 for Ethernet
    - compatible enhancements for much faster clock locking and easier/lower cost filtering at endpoints
  - superset of 1588v2 to support 802.11 WiFi, EPON and "coordinated shared networks"



- **This precise synchronization has two primary purposes:**
  - allow multiple streams to be synchronized and
  - provide a common time base for sampling data streams at a source device and presenting those streams at the destination device with the same relative timing

802.1AS and 1588 default profile differences

- **All devices participate in BMCA**
  - BMCA almost identical to default 1588-2008
  - Intermediate systems (bridges) has same performance as transparent clock for Ethernet, so no “boundary clock” vs “transparent clock” issue
- **No “partial on path” support**
  - Sync only between participating devices, non-participating device blocks sync path
- **Fewer options**
  - Ethernet is L2 two-step with peer delay
  - Single domain
- **Ports connect via link-technology-dependent methods**
  - WiFi uses 802.11v “timing measurement”
  - EPON and other coordinated shared network links also use link-dependent delay measurement technology
- **All filtering done at ordinary clocks (endpoints)**
  - Intermediate systems (bridges/switches) syntonize with GM almost instantly using neighbor rate ratio calculation and sharing via TLV included in Sync/Follow up.
  - Endpoints lock with GM within a few seconds max

19

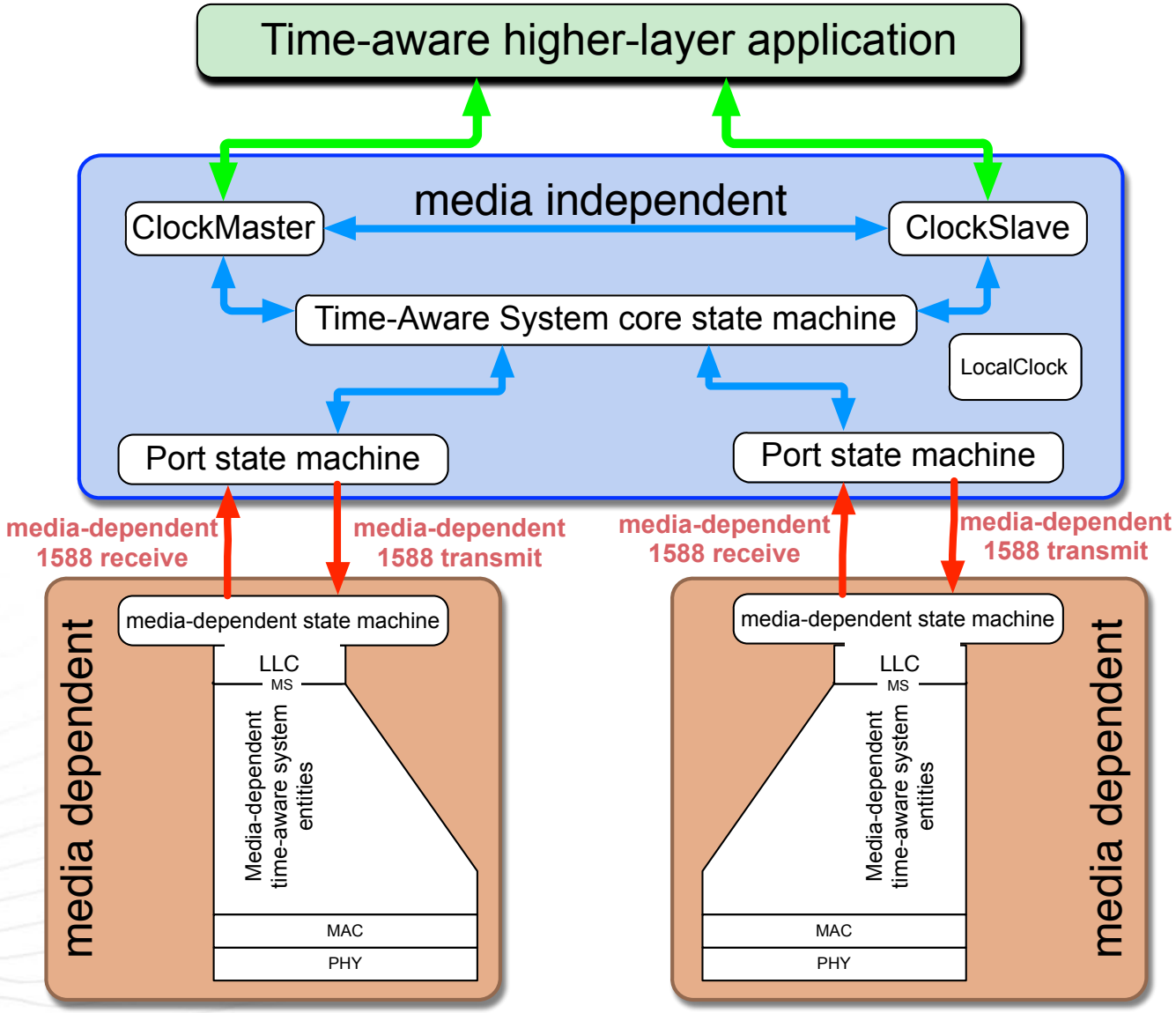
Time synch updates

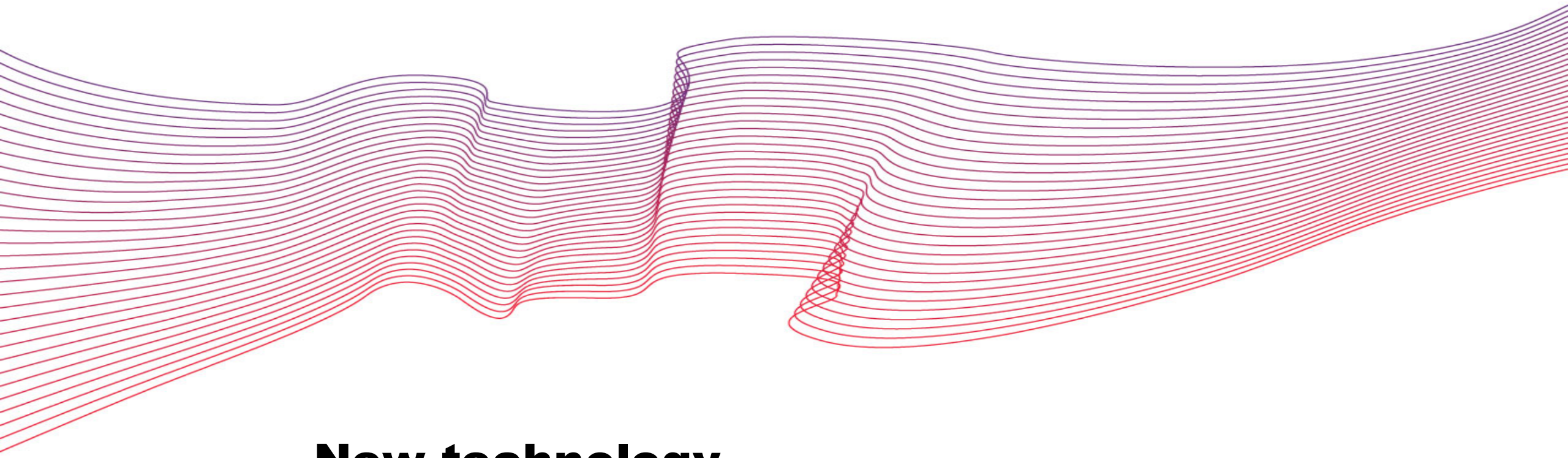
IEEE 802.1AS-revision is an update to 802.1AS for:

- **Enhanced link support**
  - Support for *all* of Ethernet, including link aggregation
    - Working with IEEE 802.3 to improve delay reporting and allow proper standardization of "one-step" time stamping
  - Support for 802.11 "fine timing measurement" for picosecond-level timestamps
- **Improve performance and usability**
  - Support multiple domains
    - Domain zero required, must be locked to TAI, other domains may be "working clock"
  - Responsiveness and reliability, support for "one step" links
    - "one step" is a port attribute, so a system can have ports operating in both one- and two-step modes
    - Master ports can be locked to slave port syncs (like a transparent clock) or not (like a boundary clock)
  - "sourcePortID" is identifier of GM, not just immediate upstream system
  - Timing quality reporting via managed objects and/or signalling
  - Explicit support for centrally-managed systems via port role
- **<u>Start</u> the process towards protocol unification**
  - End the 1588 vs 802.1AS vs NTP confusion
  - 802.1 is coordinating with the 1588 revision project

A unified architecture

- **802.1AS (future)**
  - Still a 1588 profile, but not a superset



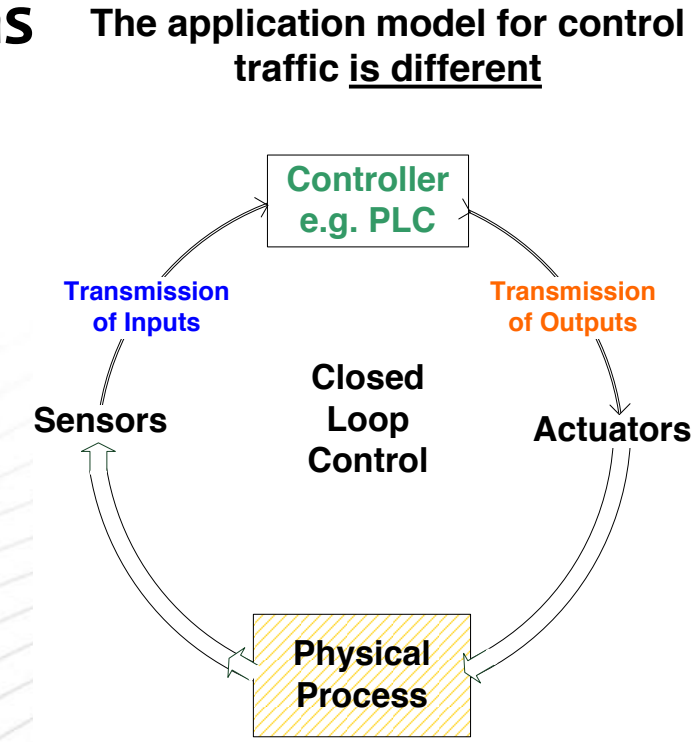


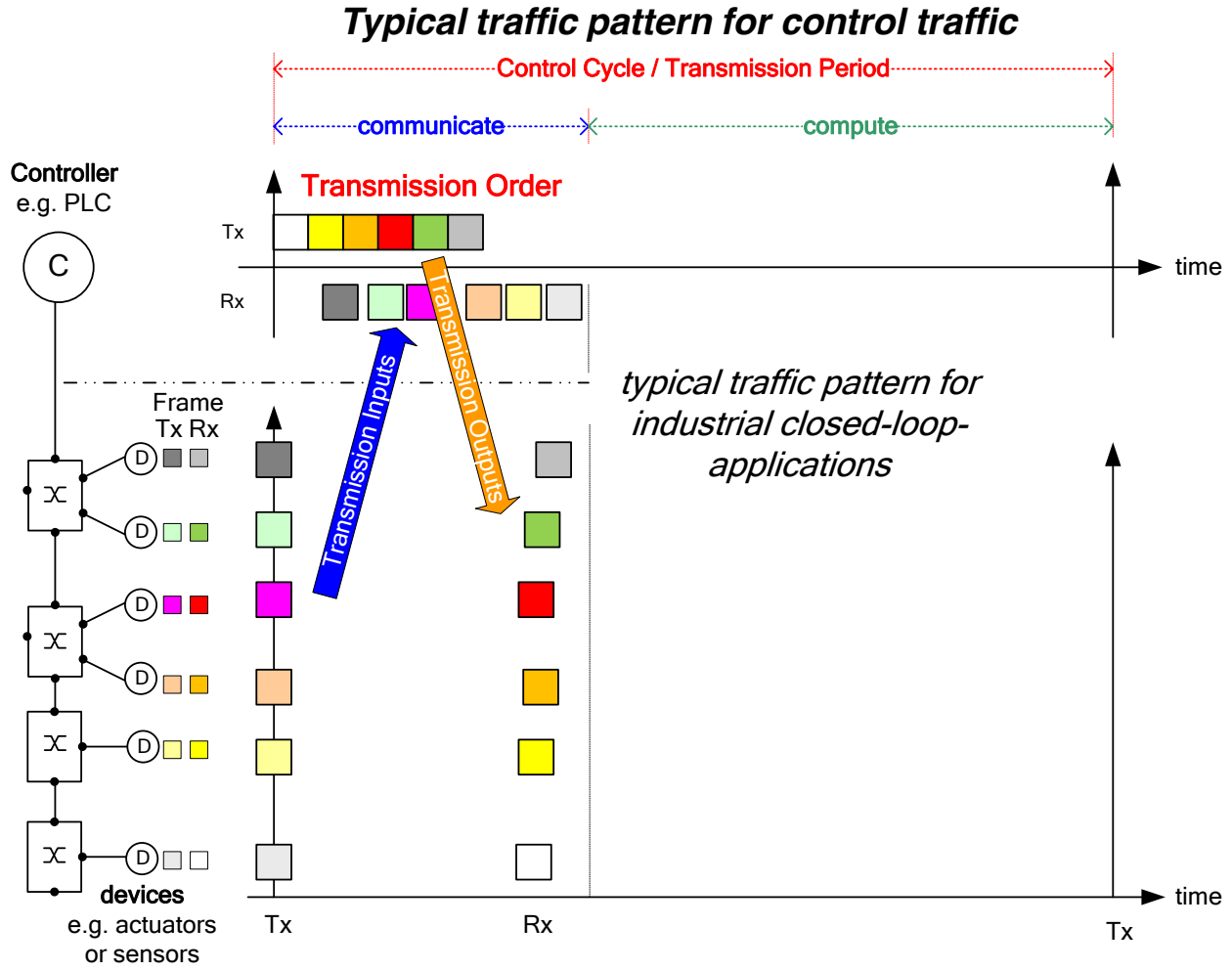
New technology …

Reducing worst-case delays

Reducing delays

- TSN applications do NOT care about average delay, nor fastest delivery
- The important number is “worst case” delay

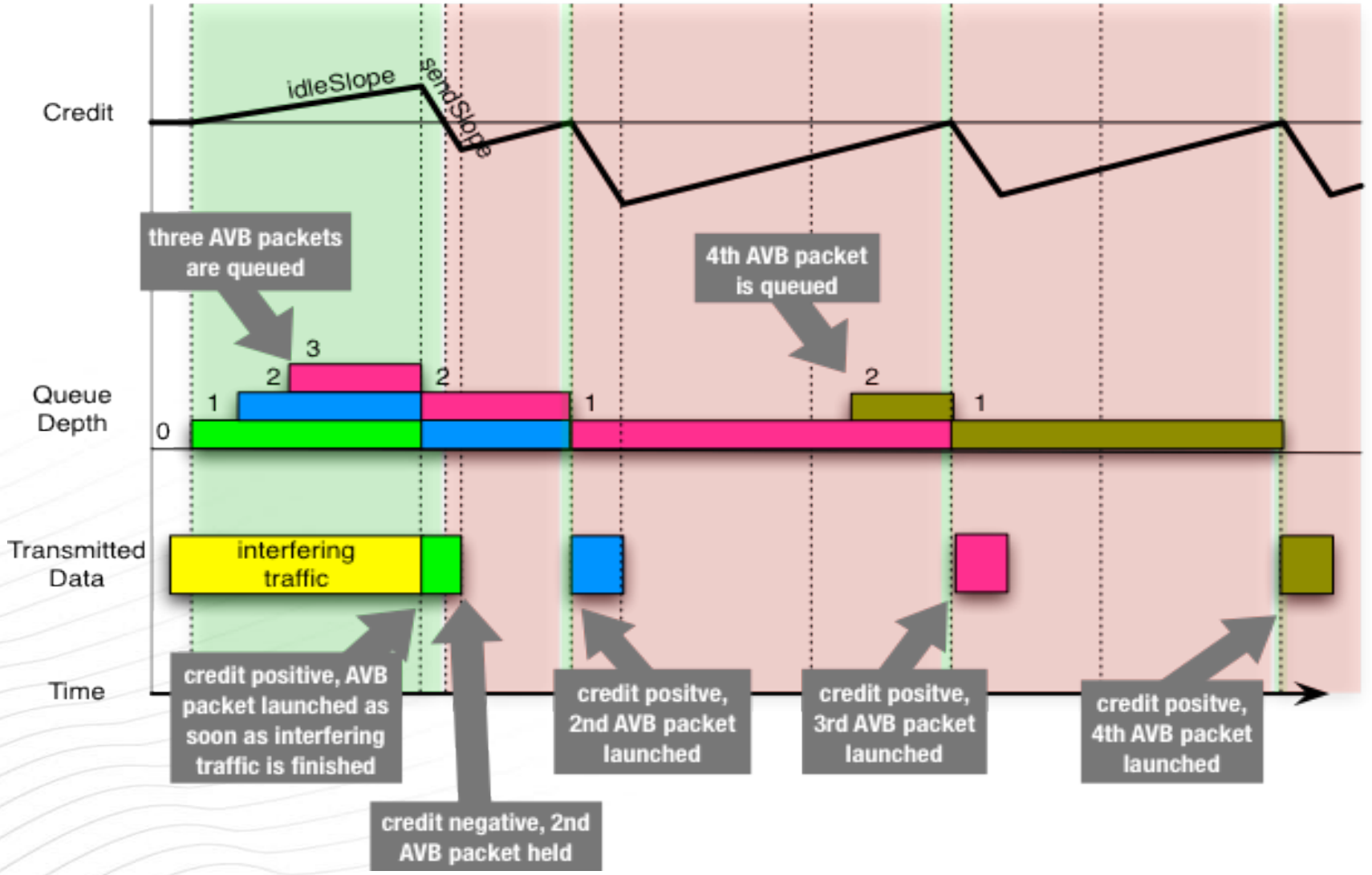




Reducing traffic delays

(IEEE Std 802.1Qav – Forwarding and Queuing for Time-Sensitive Streams – FQTSS)

- Devices in AVB network must "shape traffic"
- Schedule transmission of packets to prevent bunching, which causes overloading of network resources



Improvements for streaming

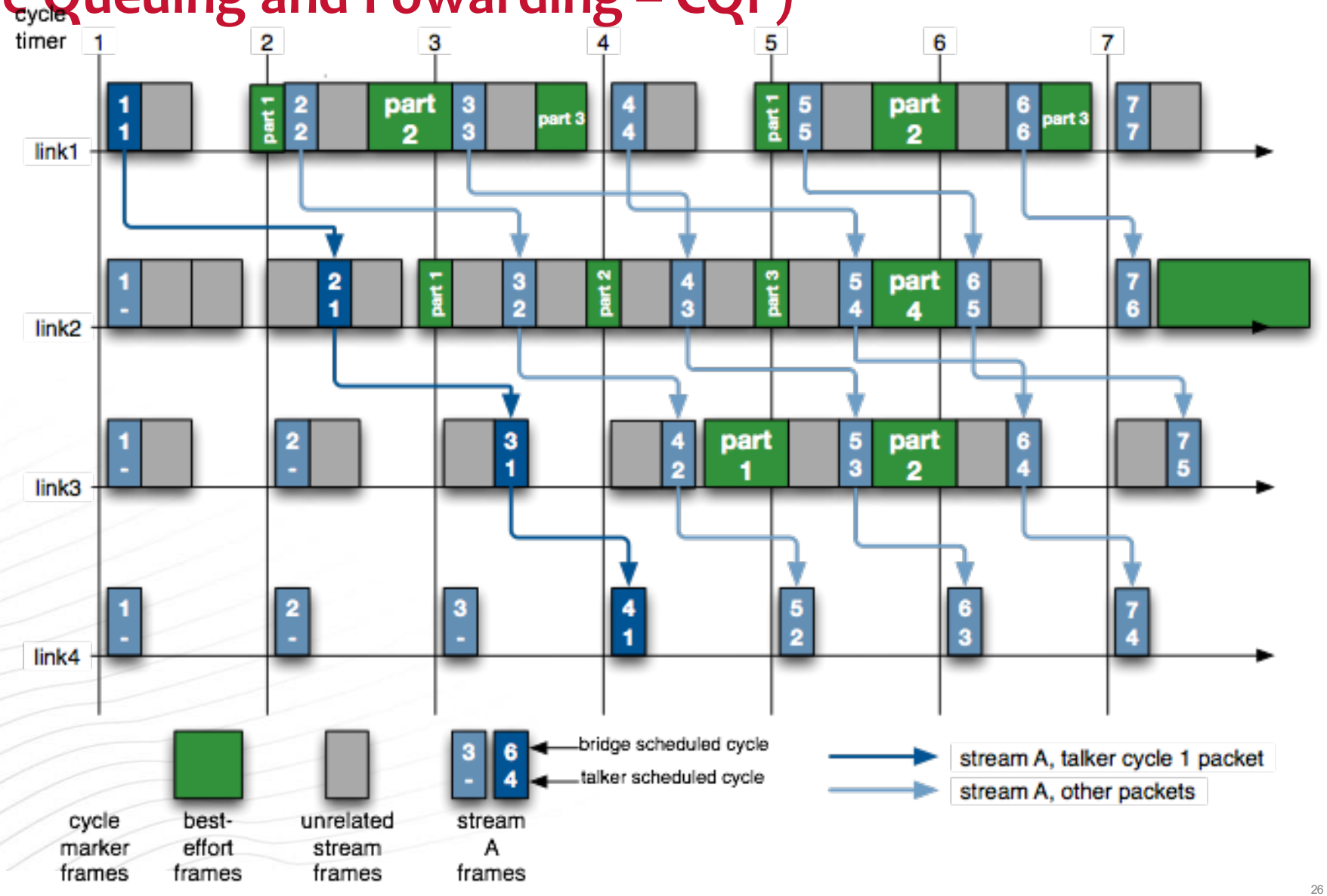
100% reliable delays and fixed delivery jitter

- **Existing FQTSS has some issues**
  - Pathological topologies can result in increased delays
  - Worst case delays are topology dependent, not just a count of hops
    - Buffer requirements in switches are topology dependent
- **New "Cyclic Queuing and Forwarding" (CQF) 802.1Qch**
  - Every switch introduces a fixed delay for each stream in a particular traffic class
    - Buffer requirements are fixed by switch design, topology of network has no effect
    - Delivery jitter is fixed, based only on traffic class
  - Intent is to *replace* the existing FQTSS
    - Compatible with FQTSS, worst-case delay improves with the number of CQF shapers in the path

Completely deterministic delays

(IEEE 802.1Qch – Cyclic Queuing and Fowarding – CQF)

- Expected to be combined with 802.1Qbu/ 802.3br (preemption) to reduce limits on minimum cycle time



Limits on Delay

That's nice, but what can we do that's better?

The fundamental problem is interfering traffic!

If a packet is to be transmitted on a particular egress port, then all traffic, regardless of the priority, must wait until the egress port has completed transmitting that packet.

(head of queue blocking)

What is the best we could do?

Let's turn the problem around …

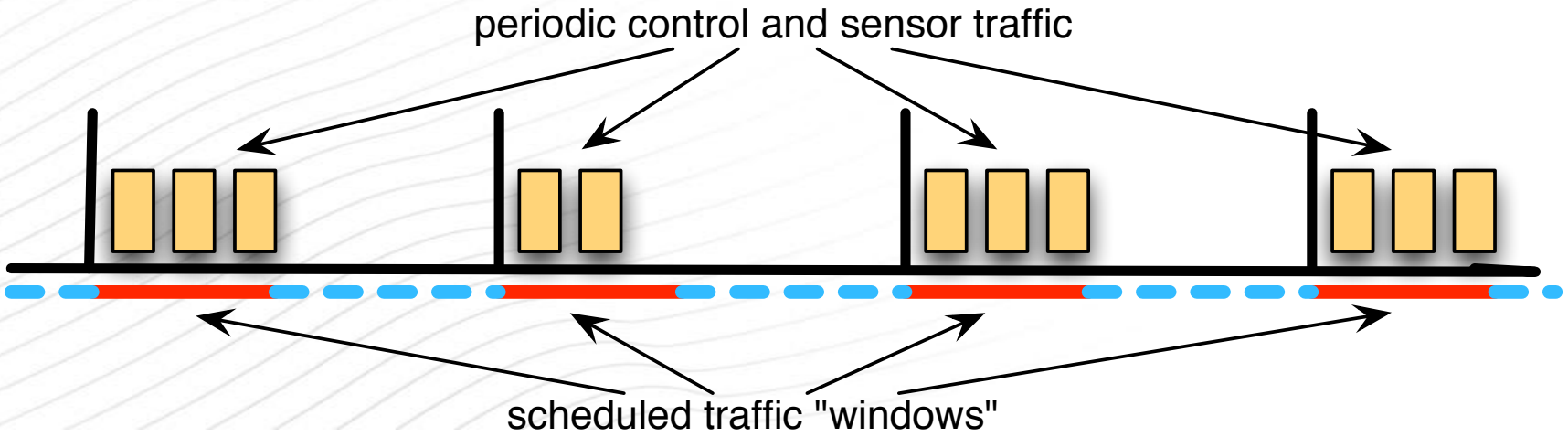
- **Assume no interfering traffic:**
  - Gigabit Ethernet switch could have delays as low as 4.122 µs.
- **Add cut-through switching:**
  - Gigabit Ethernet switch could have delays as low as 2.074 µs!
  - Note that cut-through normally does not help if there is interfering traffic, but in this case, we assume no interference.
- **So, for a 32-hop network, we would have 66.368 µs delays.**

Not bad!

Avoiding Interfering Traffic

802.1Qbv Time-Aware Shaper

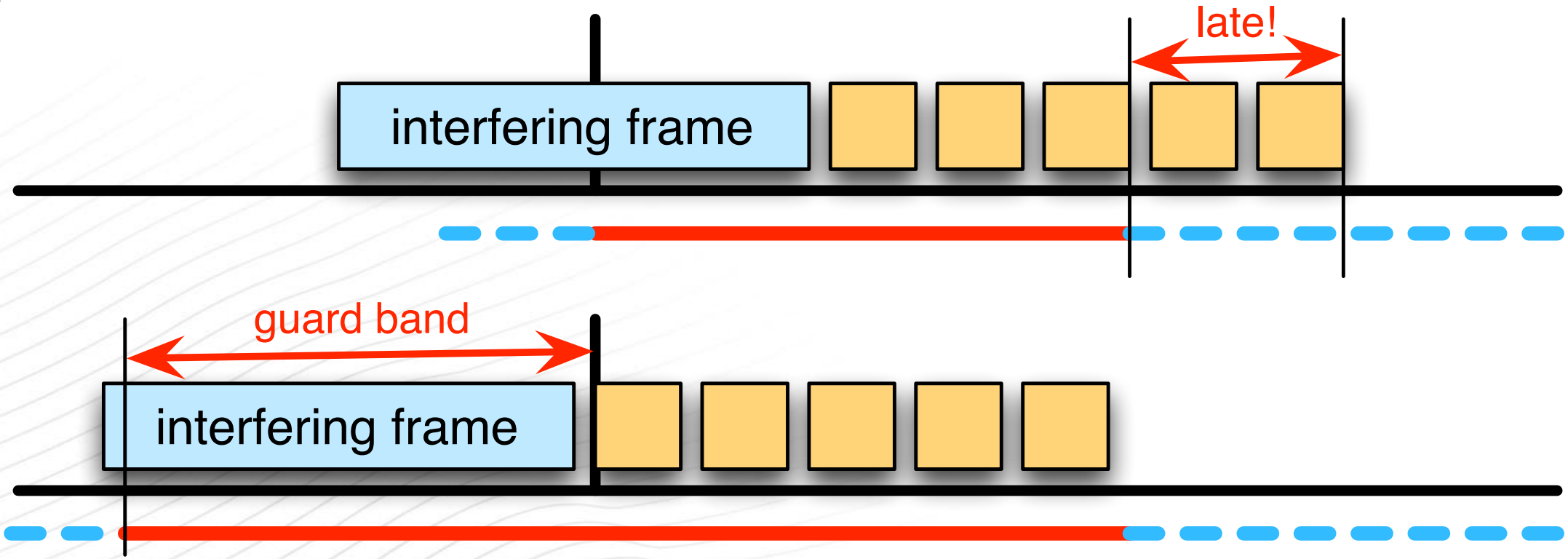
- **Make switches aware of the cycle time for control traffic:**
  - Block non-control traffic during particular windows of time to ensure that the egress port for a control stream is idle when the control traffic is expected.
  - Each egress port would have a separate schedule.
- **Nontrivial calculation in nontrivial networks:**
  - Requires a fully managed network.
  - This is a well-understood but difficult problem currently implemented in proprietary networks such as Siemens' "Profinet."



Time-Aware Shaper Issues

A “guard band” is necessary

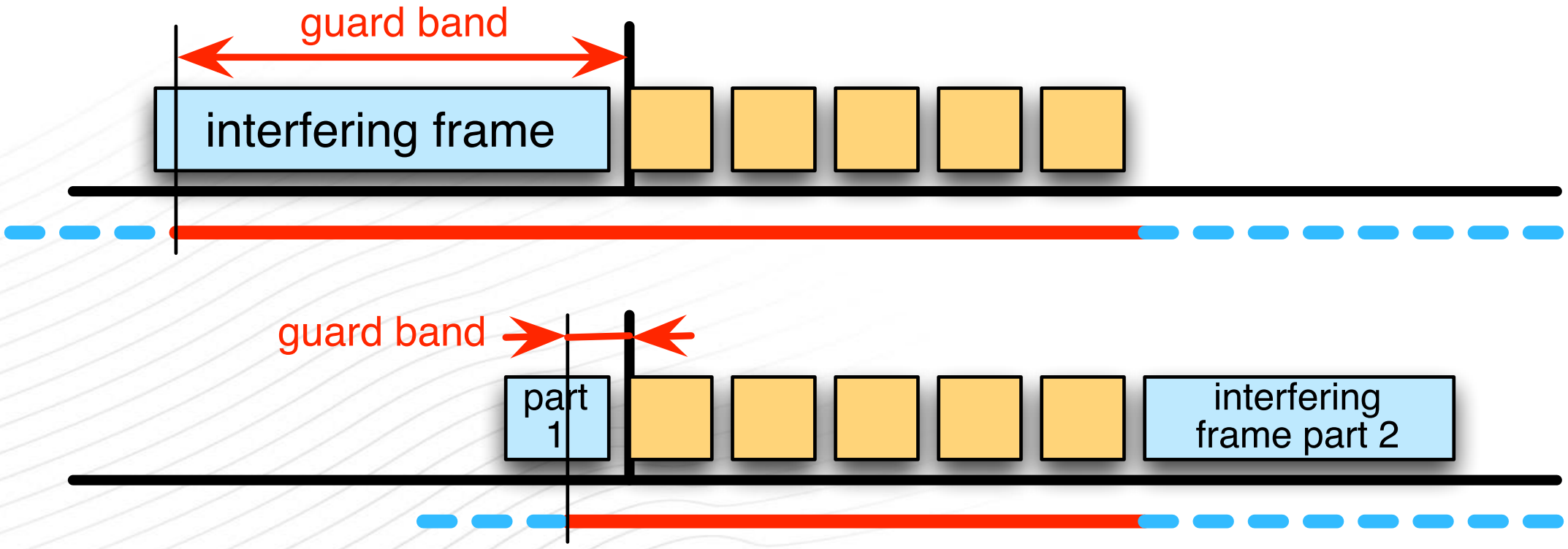
- If an interfering frame begins transmission just before the start of a reserved time period, it can extend critical transmissions outside the window.
- Therefore a guard band equal in size to the largest possible interfering frame is required before the window starts.

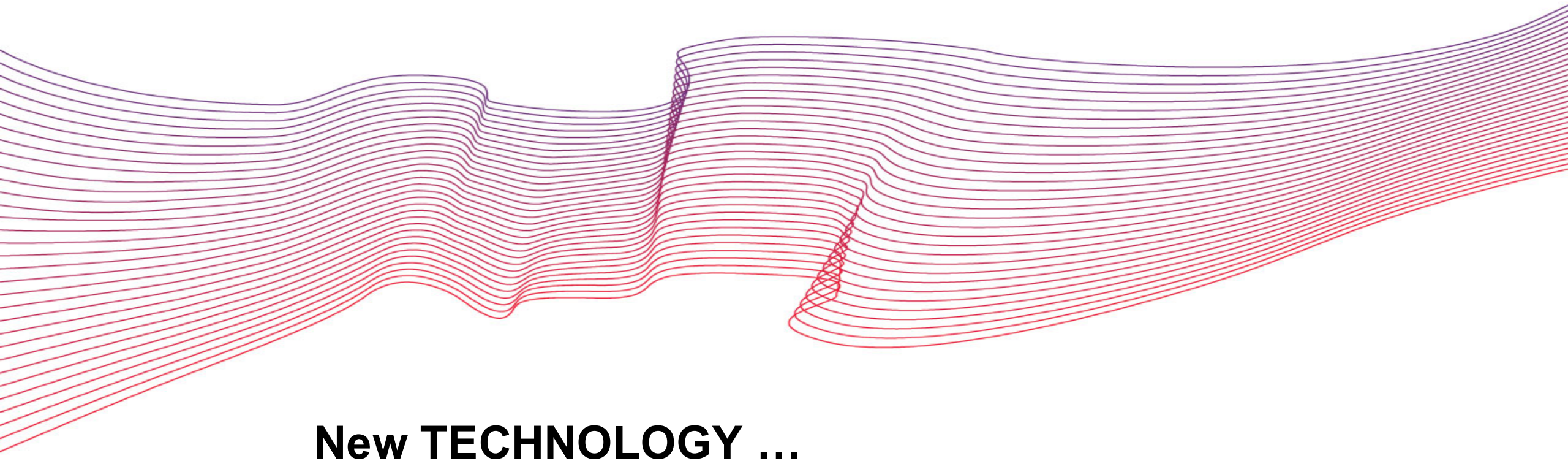


Reducing the Guard Band

Preemption (802.1Qbu/802.3br) is a solution

- If preemption is used, the guard band needs to be only as large as the largest possible interfering fragment instead of the largest possible interfering frame.





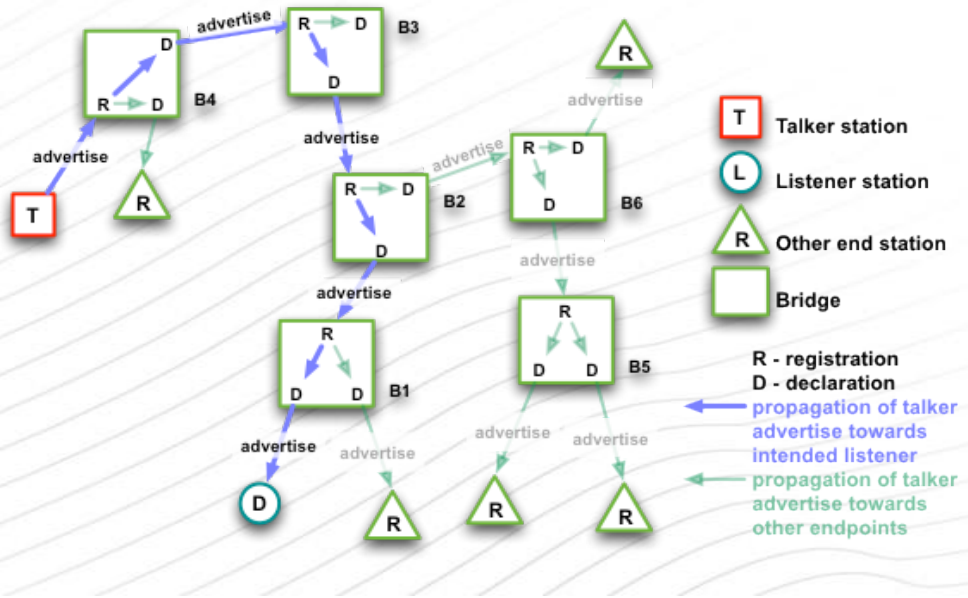
New TECHNOLOGY …

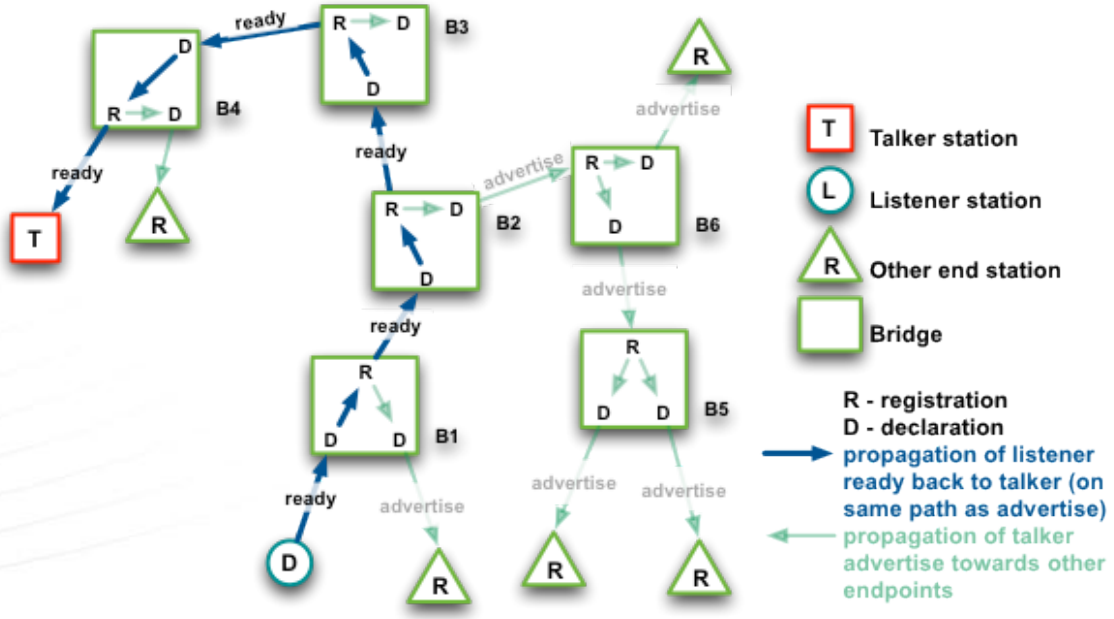
RESOURCE AND PATH RESERVATION (an “API”)

Admission controls

(IEEE Std 802.1Qat)

- **Priorities and shaping work only if the network resources are available along the entire path from the talker to the listener(s)**
  - AVB "talkers" guarantee the path to the listener is available and reserve the resources
- **Done via a "Multiple Registration Protocol" application: SRP ("Stream Reservation Protocol")**
  - Registers streams as a source MAC address combined with a higher level ID (frequently the IP port address)
  - Reserves resources for streams based on bandwidth requirements and latency class





Enhanced Stream Reservation

802.1Qcc and Q??:

- **Explicit interoperation with "God box" centrally-managed systems**
  - Allow centrally-managed and ad hoc systems to coexist
  - Management compatible with IETF YANG/NETCONF
    - Or the simpler IEEE 1722.1 and IETF "constrained application" protocols
- **Explicit interoperation with higher-level reservations**
  - A "UNI" … unified network interface, a common way to request L2 services
  - Higher layers (such as RSVP) can use this as an API the same way IEEE 1722.1 uses the existing SRP
- **Reduce the size and frequency of reservation messages:**
  - *Compatible with existing 802.1Qat Stream Reservation Protocol*
  - Relaxed timers, updates only on link state or reservation changes

TSN management use models

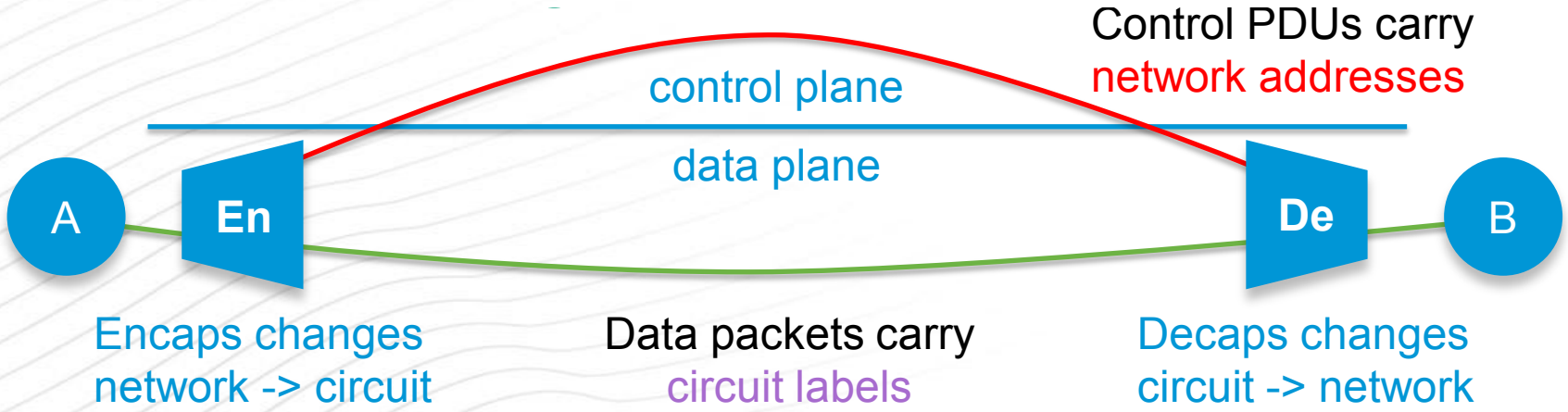
- **Self-managed/distributed management applications**
  - Small office / studios / "field" deployments
  - Many ad-hoc arrangements
- **Centrally managed applications**
  - Very large systems – or -
  - Highest performance / lowest delays / most reliable – or -
  - Commercial constraints for remote management
    - "my movie is breaking up"
  - "best" path or "safest" path or "most efficient" path hard to determine without global knowledge

Use what exists

- **Ad hoc / distributed management uses …**
  - RSTP (simplest) or 802.1Qca (IS-IS-based path control – complex but capable) for path selection
  - New SRP for resource reservation using existing paths
- **Centralized management could …**
  - Use new SRP as the "UNI" … the unified network interface
    - The "API" for network endpoints to request services
  - Endpoints would not need to know what kind of network management is being used
- **This requires …**
  - Updates to SRP to carry information useful for centralized managers to determine appropriate paths
  - Selection of a "default" centralized management system

Example: existing IP-based applications

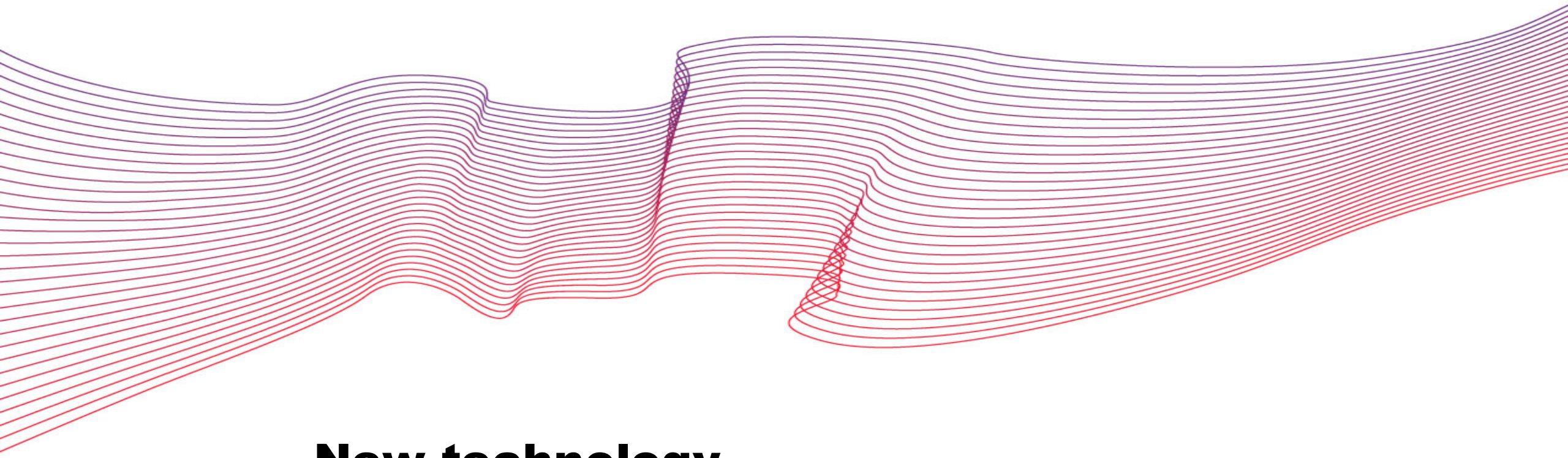
- **TSN streams are identified by both a traffic class tag and a locally-managed group destination MAC address ("circuit label").**
  - IP stacks don't work well with locally-managed addresses
  - Never had to in the past, no mechanisms exist
- **Use enhanced SRP (802.1Qcc) to carry tunneling information (address, traffic class) between endpoints**
  - Enables straight-forward encapsulation/deencapsulation ("encap/decap") using the TSN stream identification function
- **Higher level reservation system (IEEE 1722.1, new RSVP, centralized management) must interface with SRP for this to work.**



Centralized management

Use IETF protocols as the baseline

- **Newest recommendations are YANG/NETCONF**
  - YANG is a much-improved way to describe the data model for configuration and status
    - Replacement for SMI MIB modules, *MUCH* easier for humans to read and parse
  - NETCONF is the protocol used to use the YANG models for actually configuring network devices and getting status
    - Somewhat complex/session based using SSL, uses XML encoding, very verbose
  - Simplified, but compatible subsets to be considered
    - RESTCONF is an HTTP, sessionless version of NETCONF
    - Proposed binary encodings using well-known schema and IEEE 1722.1 or "constrained system" protocols
- **PCEP (path control element protocol)**
  - Existing protocol used for router management in traffic-engineered networks

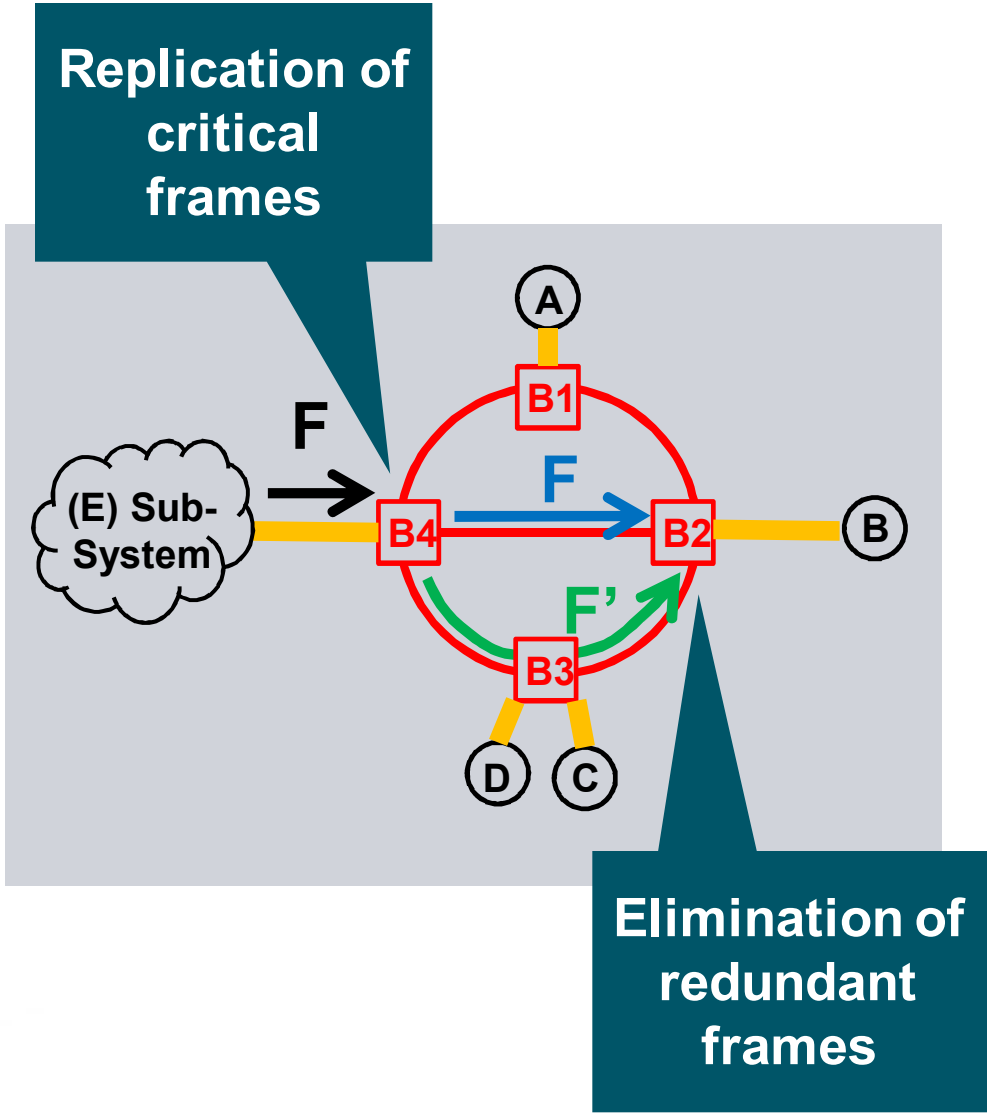


New technology …

Redundancy and reliability

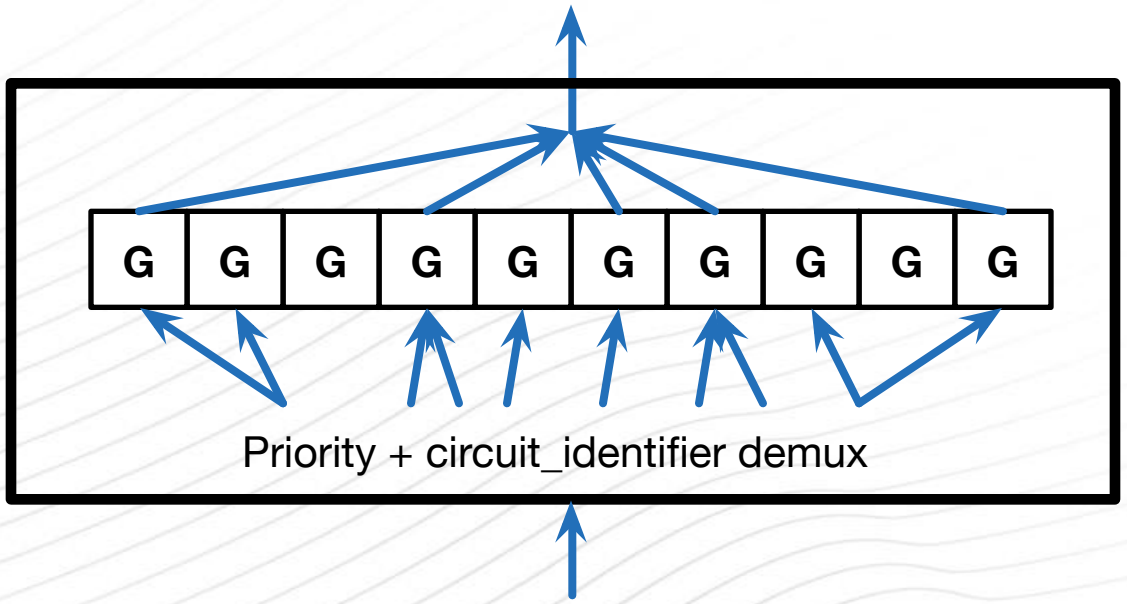
802.1CB: Seamless redundancy

- **Selective packet replication based on address/traffic class and path information**
  - done by TSN stream identification plus "sequence generation" functions
- **Duplicate frame elimination**
  - based on address/traffic class (TSN stream), sequence number and timing
  - timing information needed to limit memory needed for duplicate frame detection
- **Compatible with existing industrial architectures**
  - E.g., HSR, PRP
- **Management NOT TRIVIAL!**
  - Almost certainly requires centralized management



P802.1Qci – Input Gates

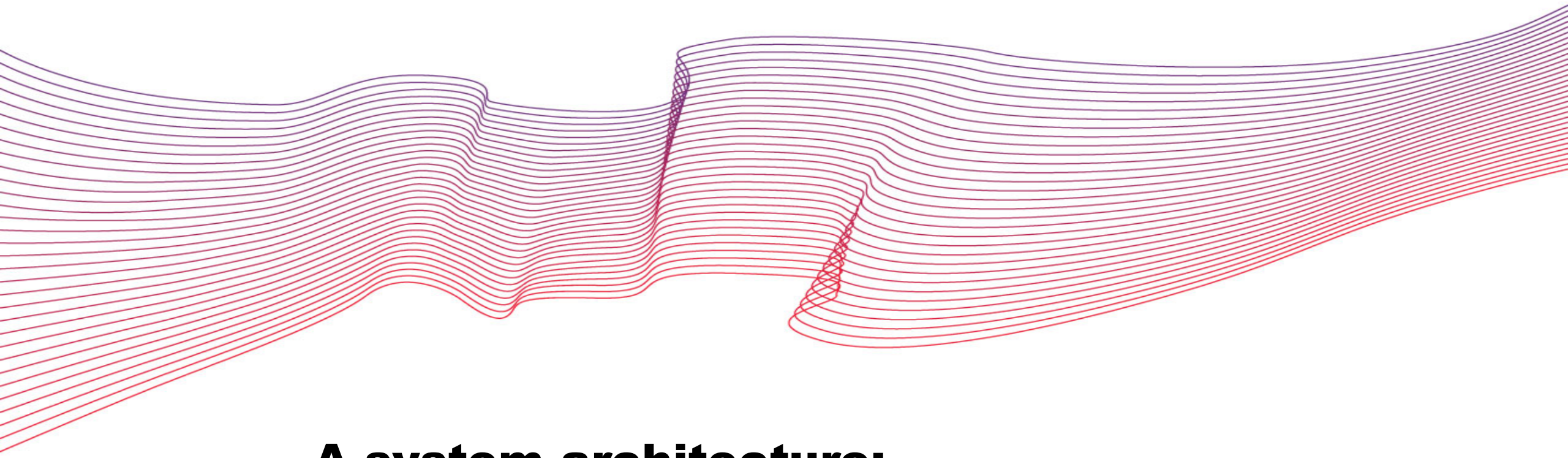
- **Need to provide protection for the QoS and redundancy features**
  - Mainly to protect against software bugs on endpoints, but maybe switches/bridges, maybe hostile devices
- **Make sure streams don't exceed their contracts!**
  - Excess bandwidth, burst sizes, packet sizes, misuse of labels



Each gate has:

1. A pass / don't pass switch
2. Optional policers
3. Counters
4. Service class output specifier

All of these perhaps based on time



A system architecture:

Higher layers

Higher Layer standards: IEEE

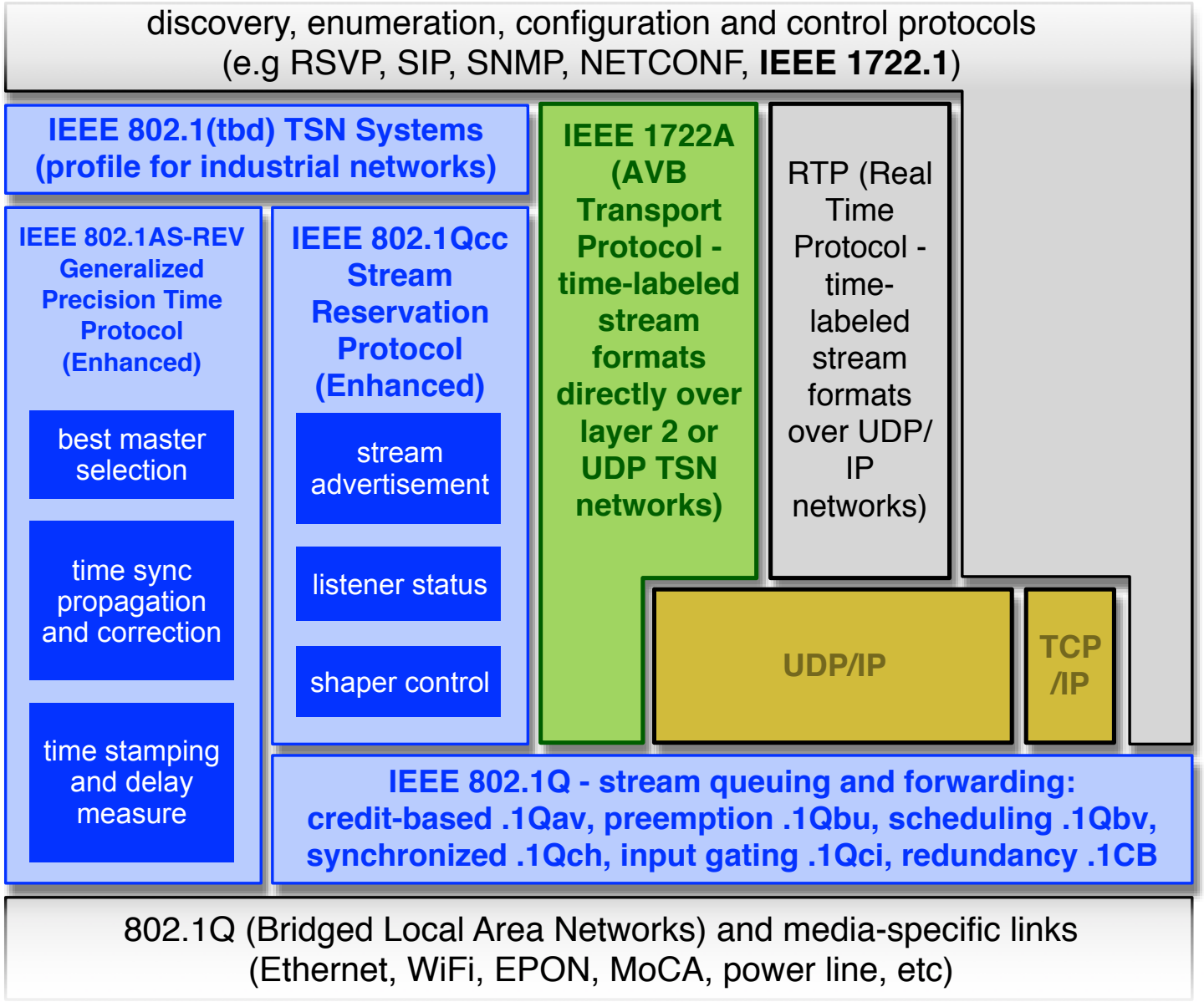
- **IEEE 1722 "AVB transport" (AVBTP)[1]**
  - Stream format for time-sensitive streams
    - Based on IEC 61883 formats used in IEEE 1394/Firewire … common in the pro audio market
    - Originally assumed direct IEEE 802 addressing, no IP encapsulation
    - New 1722-revision adds many new simplified formats, UDP/IP encapsulation, security
  - Method for allocating group MAC addresses used by SRP as stream ID's
- **IEEE 1722.1 "Discovery, enumeration, configuration and control" (DECC)**
  - Fills in the protocols needed to build systems based directly on AVB and 1722
  - Assumes direct 802 addressing, but UDP/IP encapsulation will come with 1722-revision
- **AVnu compliance and interoperability specifications**
  - "AVnu Alliance" (www.avnu.org) set up for compliance and interoperability testing and branding
  - Feeds results of testing and "lessons learned" back to IEEE standards groups

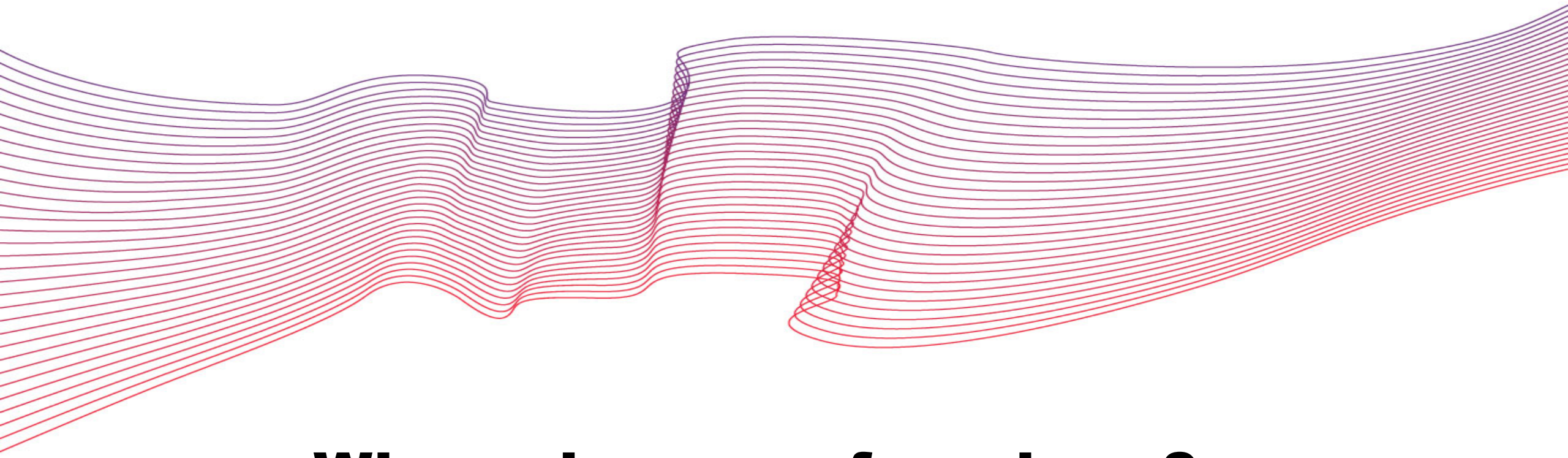
[1] "AVB" name about to change to something more "TSN-ish"

Higher layer standards: IETF and others

- **RTP (real time protocol) for stream encoding**
  - Extremely broad range of applications from VoIP to TV broadcast
    - Many, many options and profiles … no such thing as a universal RTP
    - Works well in heavily managed networks, highly variable quality otherwise
- **RSVP (path and QoS reservations)**
  - Mostly unsuccessful, tied to "INTSERV" … per-stream QoS
    - Not scalable, too many options, not tied to capabilities of lower layers
  - Somewhat more successful as a path reservation system for some MPLS systems
    - RSVP-TE (traffic engineered)
- **AES (Audio Engineering Society)**
  - Incomplete set of standards on how to use RTP and 1588 for transport
- **SMPTE (Society of Motion Picture and Television Engineers)**
  - Set of standards on how to use RTP for uncompressed video
  - Working on methods to use 1588 for studio timing reference

Putting it all together: TSN system layering





Where do we go from here?

With TSN services we have …

"802 everywhere"

- **802.3, 802.11 and 802.15 links are scaling to all sizes, speeds, costs, power**
  - 10/100/1G single pair Ethernet, 25G two pair short range, 100G+, etc, etc
  - multiGigabit wireless, or years-long operation on a coin cell
- **Wide area networks with 802 architecture lower layers can now provide "universal service"**
  - Existing transaction and bulk transfer user models (the traditional internet)
  - Existing streaming services with MUCH better QoS (no more "buffering …" messages from Netflix, shorter delays for voice/video calls … Skype that really works).
  - New time-based data exchange (the "industrial internet")
- **Scaling down to a smaller physical extent …**
  - Mobile front haul … "Radio over Ethernet"
  - Industrial monitoring and control systems
  - Stadium-extent phased array speaker systems
  - Echo-free airport announcing systems
  - Time-synchronized server farms with vastly reduced internal traffic delays
- **Scaling down to the room or desk …**
  - Replacing HDMI, Display Port, FireWire and any other A/V interconnect

Questions?

Broadcom Proprietary and Confidential. © 2014 Broadcom Corporation. All rights reserved.

Why is there any other L2 technology?

How about a really universal interconnect?

- **FibreChannel is already moving towards "FCoE"**
- **Why isn't there "HDMIoE"?**
  - Why the hell is there an optional Ethernet channel muxed into HDMI 1.3
  - And there really should never be a "wireless HDMI" unless you want to jam your WiFi at 60Ghz
  - This only takes a mapping from HDMI onto 1722
- **Why isn't there a USBoE?**
  - And why is there a wireless USB?

Wait a minute …

- **There is "USB Type C Ethernet Alternate Mode"**
  - Or there will be soon (watch this space)
  - Uses the USB 3.1 PHY to run Ethernet at 5 to 20 Gb/s
    - And someday, using Ethernet PHY technology, 50 to 100 Gb/s
  - Next step is, indeed, USB protocols running on top of Ethernet … USBoE
  - Then a USB hub is just an Ethernet switch and …
  - Wireless USB is just 802.11 or 802.15.4