PRIORITY SUPPORT FOR PLCA

WHAT MINIMALLY NEEDS TO BE DONE IN 802.3 SO THAT THE REST CAN BE DONE IN 802.1

DON PANNELL,

DONALD.PANNELL@NXP.COM

SEP 2018 – SPOKANE

SECURE CONNECTIONS FOR A SMARTER WORLD

Agenda

- Overview
- The Problem
- Proposal to Support Priorities in PLCA
- Estimated Latency & Logic Costs
- Worst Case Latency Improvement
- Things Still to Do
- Summary

Supporters (alphabetical by last name):

- Piergiorgio Beruto, 802.3, Canovatech
- Rodney Cummings, 802.1, National Instrument
- Norm Finn, 802.1, Huawei
- Claude Gauthier, 802.3, NXP
- Craig Gunther, 802.1, Craig Gunther Consulting
- Don Pannell, 802.1, NXP
- Jordan Woods, 802.1, Analog Devices
- Wang Xinyan, 802.3, Huawei

PART I

Overview

First, When did Priority become an 802.3 Issue?

- Or, "Why is this the 1st time Priority support in an 802.3 PHY has come up?"
  - Because it wasn't relevant or needed until now:
    - Priorities were added to Ethernet Frames (802.3ac) & Bridges (802.1p) in Sep 1998
    - And the 802.3cg project is the 1st PHY to use the 802.3 CSMA/CD MAC since then!
    - All other past 802.3 PHY projects supported full-duplex point-to-point links (or other MACs) so Priority considerations in a PHY was never an issue (EPON is point-to-multipoint w/TDM)

- Why isn't this an 802.1 problem to solve?
  - It is…
    - But 802.1 can't completely solve the problem by itself in the case where shared media is used
    - "802.1 only" solutions do not lower the worst case latency, they only minimize their occurrences

- What is needed from 802.3?
  - Optional Priority information exchange so that 802.1 can do all the policy decisions

Why is Priority support needed?

- To be consistent with what 802.1 has defined for TSN (Time Sensitive Networking)
  - The TSN architecture is a proven mechanism to support converged data on a single wire
    - Traditional Ethernet traffic (e.g., IP) shares the same link as Reserved &/or Time Critical traffic
    - The Ethernet frame's Priority Code Point bits (the "user priority" bits defined in 802.3ac-1998, now moved to 802.1Q Annex G as referenced by 802.3-2015 1.4.347) are used to distinguish between these various traffic types so that differentiated services can be applied to them
    - The mechanism used in TSN to differentiate the service applied to various traffic types is priority
      - For TSN to work, some frames must be treated better (unfairly) compared to other frames
      - This is needed to support the lowest possible latencies and to guarantee the reserved bandwidth
  - The success of TSN is evident by the increasing number of 802.1 use case TSN Profiles:
    - Audio Video Bridging (AVB) Systems – IEEE 802.1BA-2011
    - Time-Sensitive Networking for Fronthaul – IEEE 802.1CM-2018
    - TSN Profile for Industrial Automation – IEC/IEEE 60802 (PAR approved May 2018)
    - TSN Profile for Automotive – IEEE 802.1?? (PAR to be voted on in Nov 2018)

802.3 & 802.1 have a Long History of Jointly Solving Issues

- VLAN Tag (802.3ac) – where the Priority bits reside
- Frame Expansion (802.3as) – initially for MACSec's extra fields
- Energy-efficient Ethernet (802.3az) – need to know frames need to be transmitted
- MAC Control Frame for Priority-based Flow Control (802.3bd) – for data centers
- Ethernet Support for the IEEE P802.1AS Time Synchronization Protocol (802.3bf)
- Interspersing Express Traffic (802.3br) & Preemption (802.1Qbu)

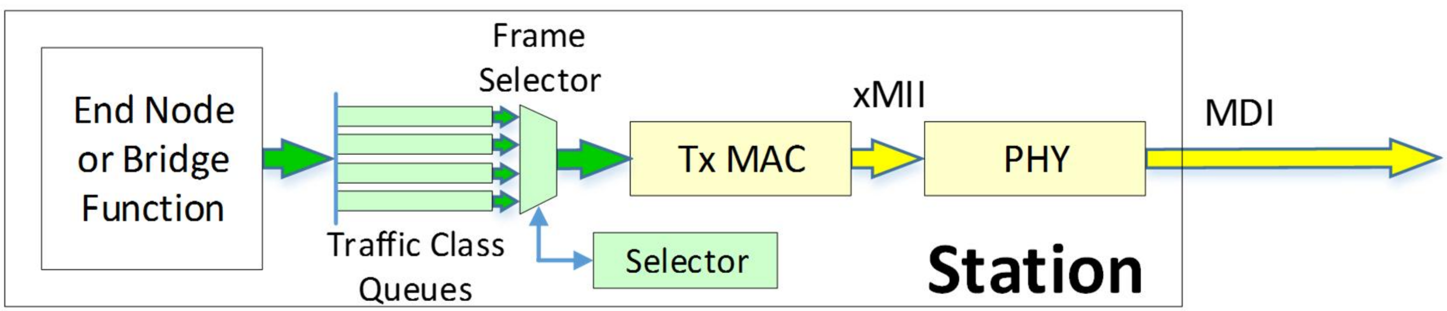
- We need to work together on this issue too!
  - 802.1 can't solve the problem by itself without additional data from 802.3 (http://www.ieee802.org/1/files/public/docs2018/new-TSN-pannell-QoS-for-PLCA-0518-v02.pdf)

- This proposal has 802.3 transferring information only to/from 802.1 and then all the policy decisions are made in 802.1
  - This minimizes the impact to 802.3 & maximizes the flexibility for 802.1

PART 2
The Problem

The TSN Transmit Architecture

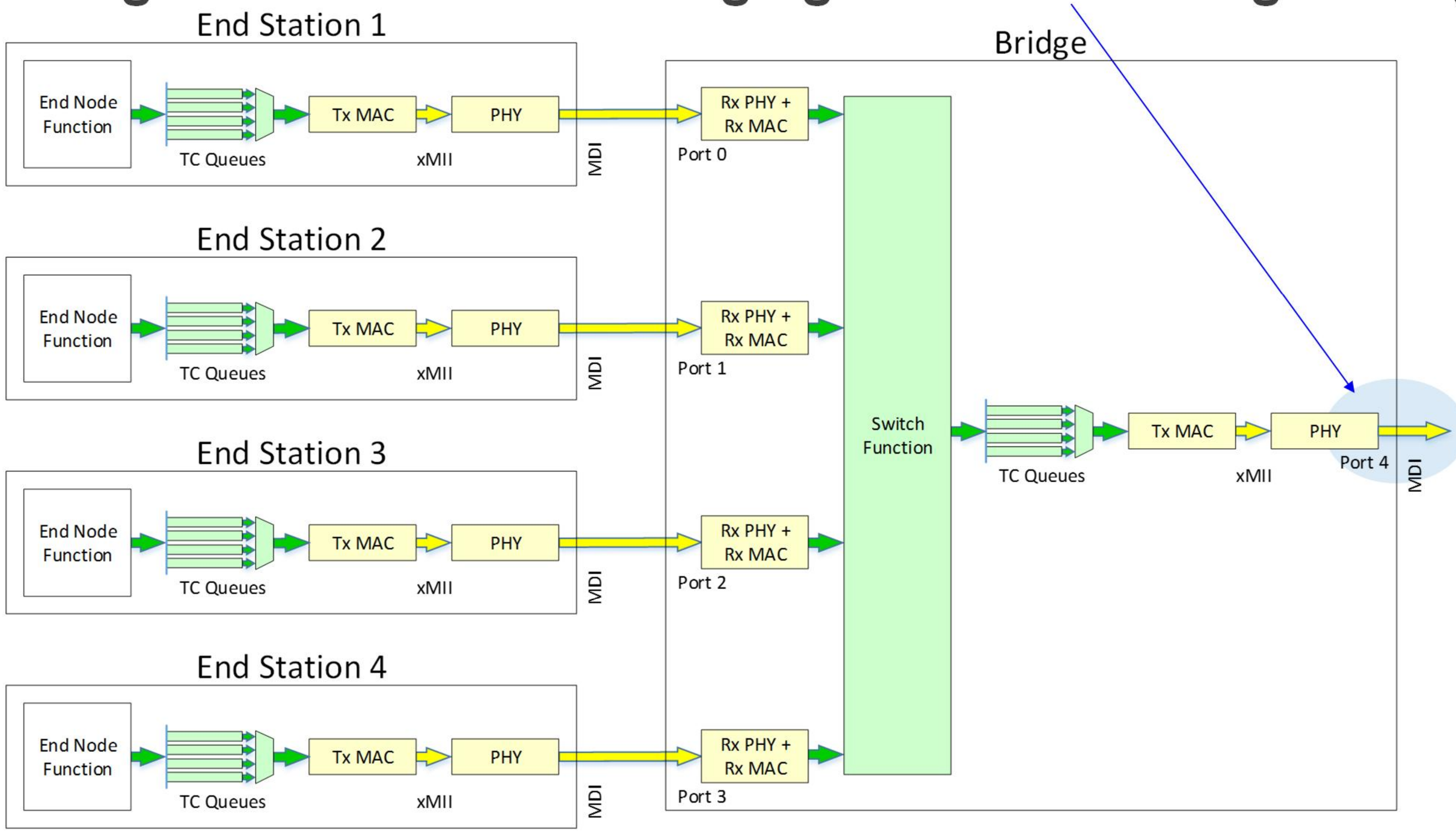
- The 802.1 portion:
  - Stations have 1 or more transmit Traffic Class Queues
  - In TSN, these Queues are used to separate the various traffic types in order to support the differentiated services that each traffic type needs
  - The Selector knows what Queues have frames to transmit & the “Queue’s configured traffic type”, and based on this, selects the next frame to transmit
    - Various Selector algorithms are supported in 802.1 to support the various traffic types
  - Normally this next frame is released to the Tx MAC right away



Green = IEEE 802.1

Yellow = IEEE 802.3

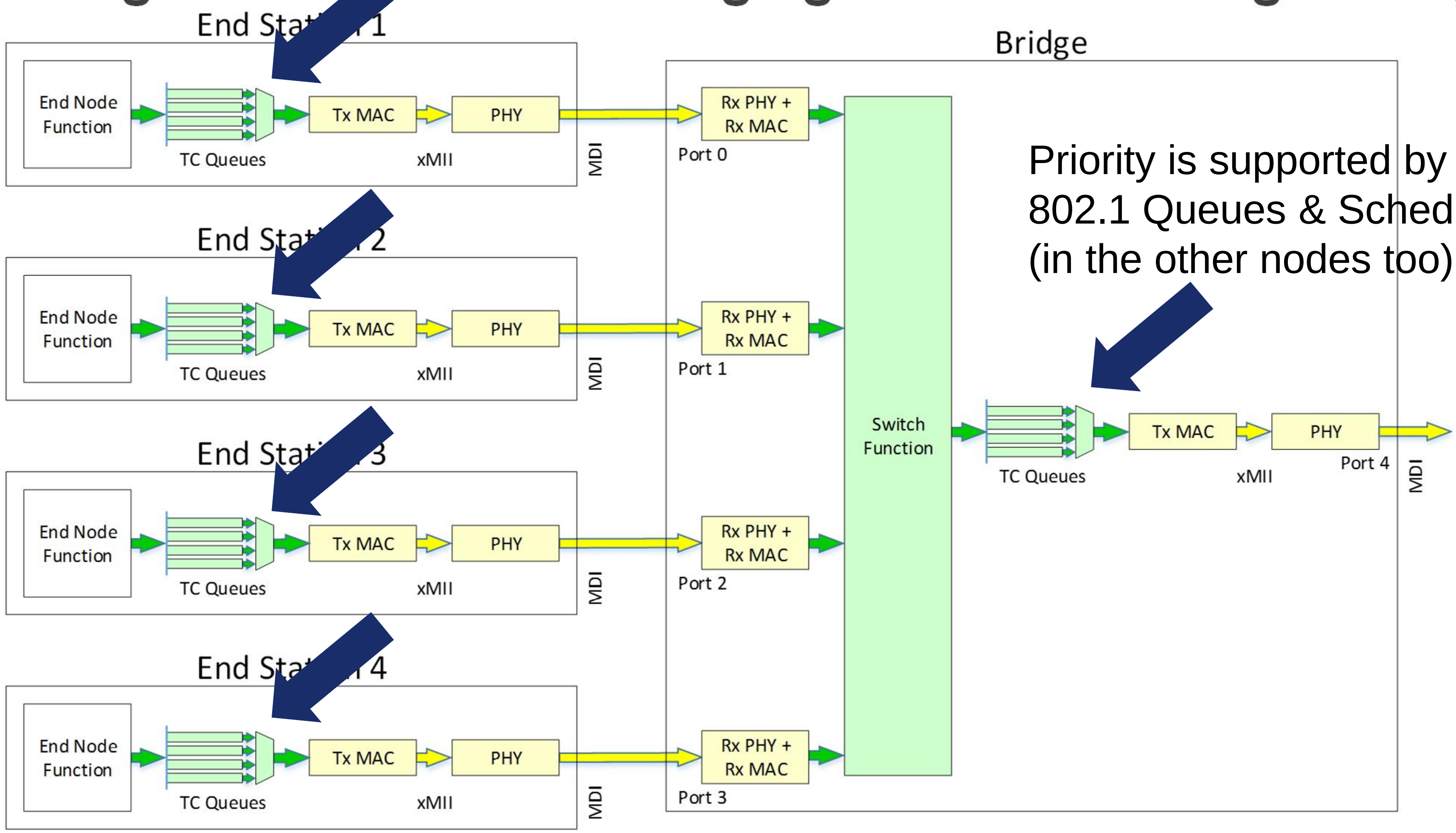
Extending this to 4 Stations merging to 1 Link – Using a Bridge



Green = IEEE 802.1

Yellow = IEEE 802.3

Extending this to 4 Stations merging to 1 Link – Using a Bridge

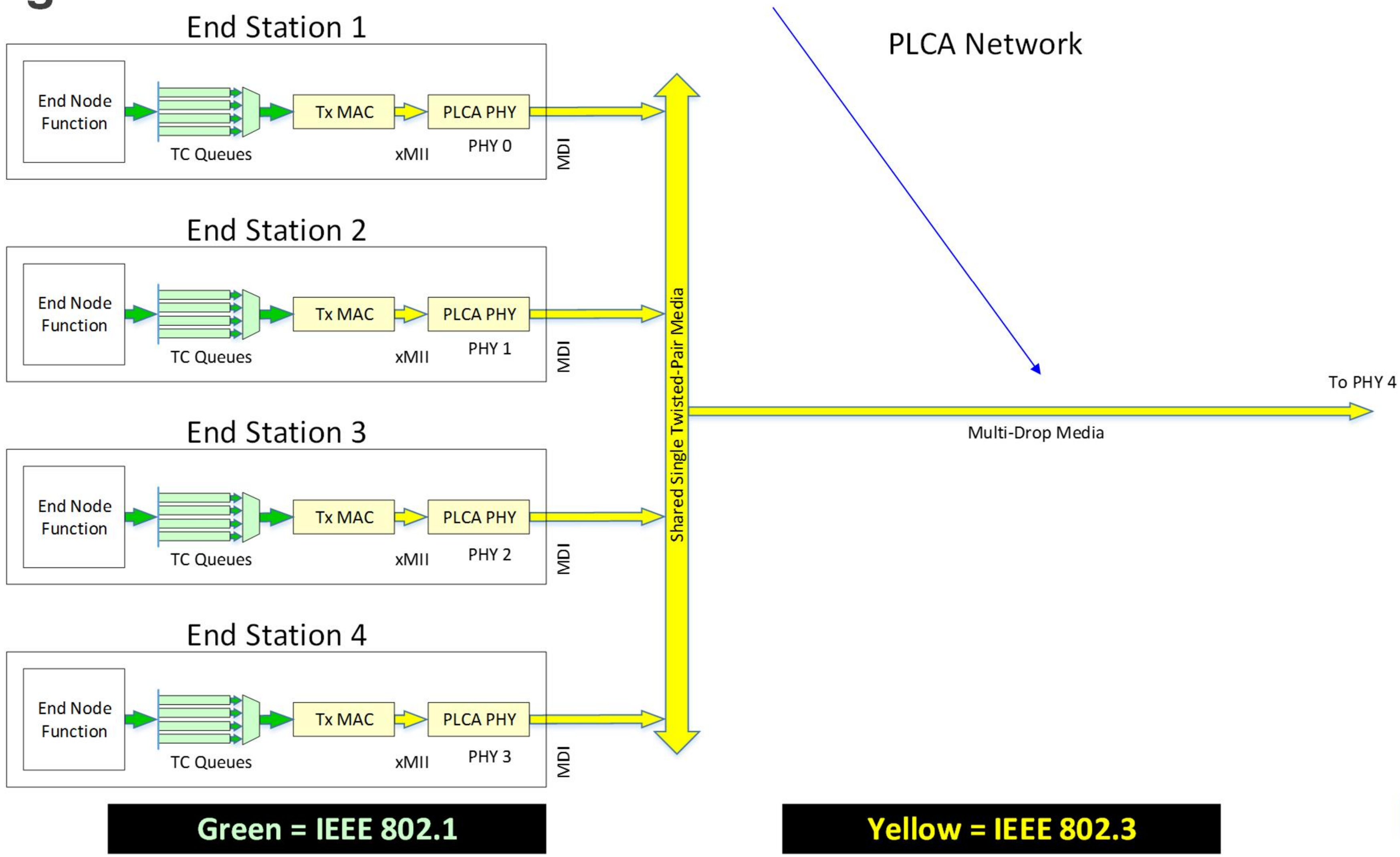


Priority is supported by these 802.1 Queues & Scheduler (in the other nodes too)

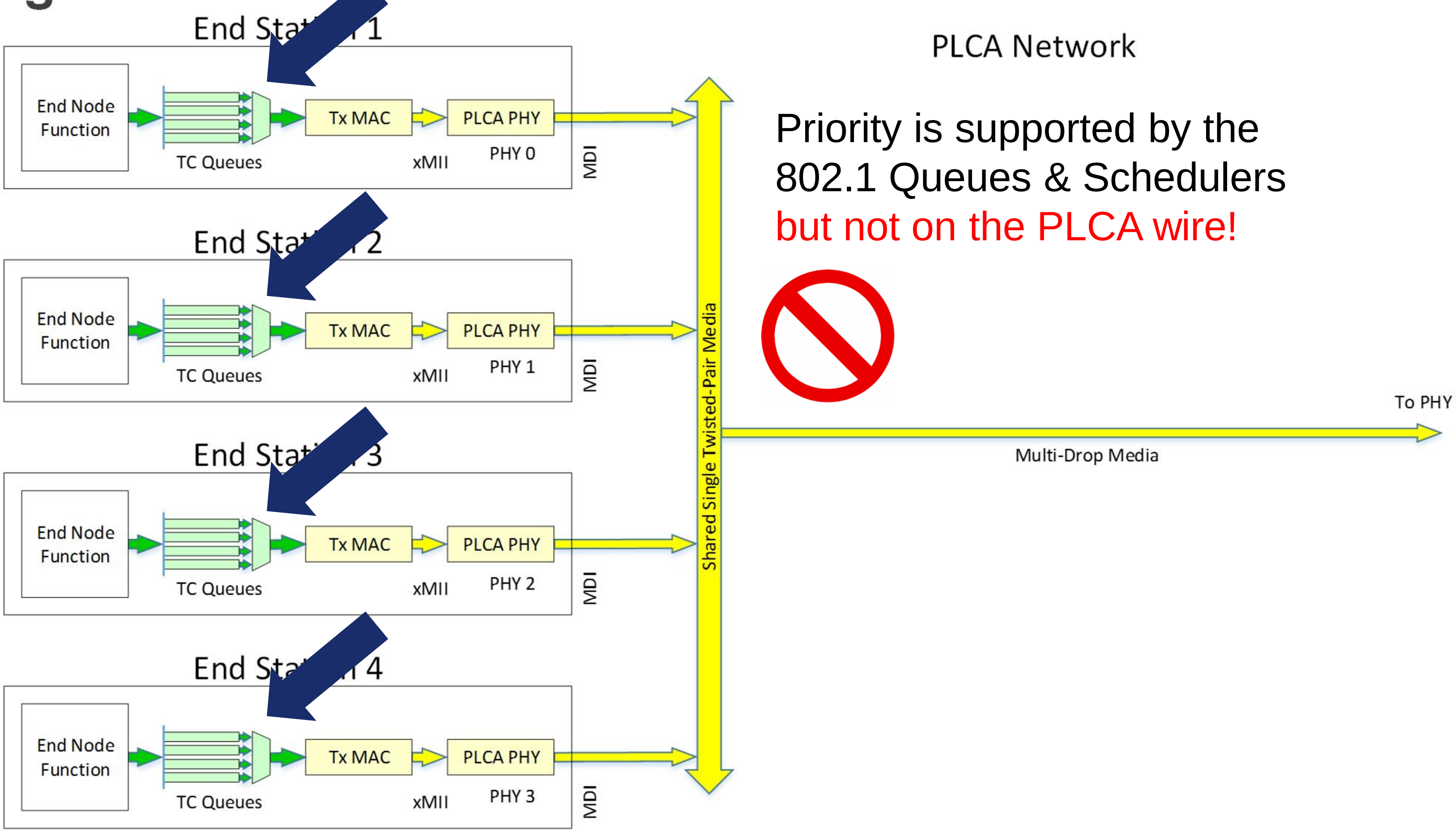
Green = IEEE 802.1

Yellow = IEEE 802.3

Change the 4 End Stations to 1 Link to Now Use PLCA



Change the 4 End Stations to 1 Link to Now Use PLCA

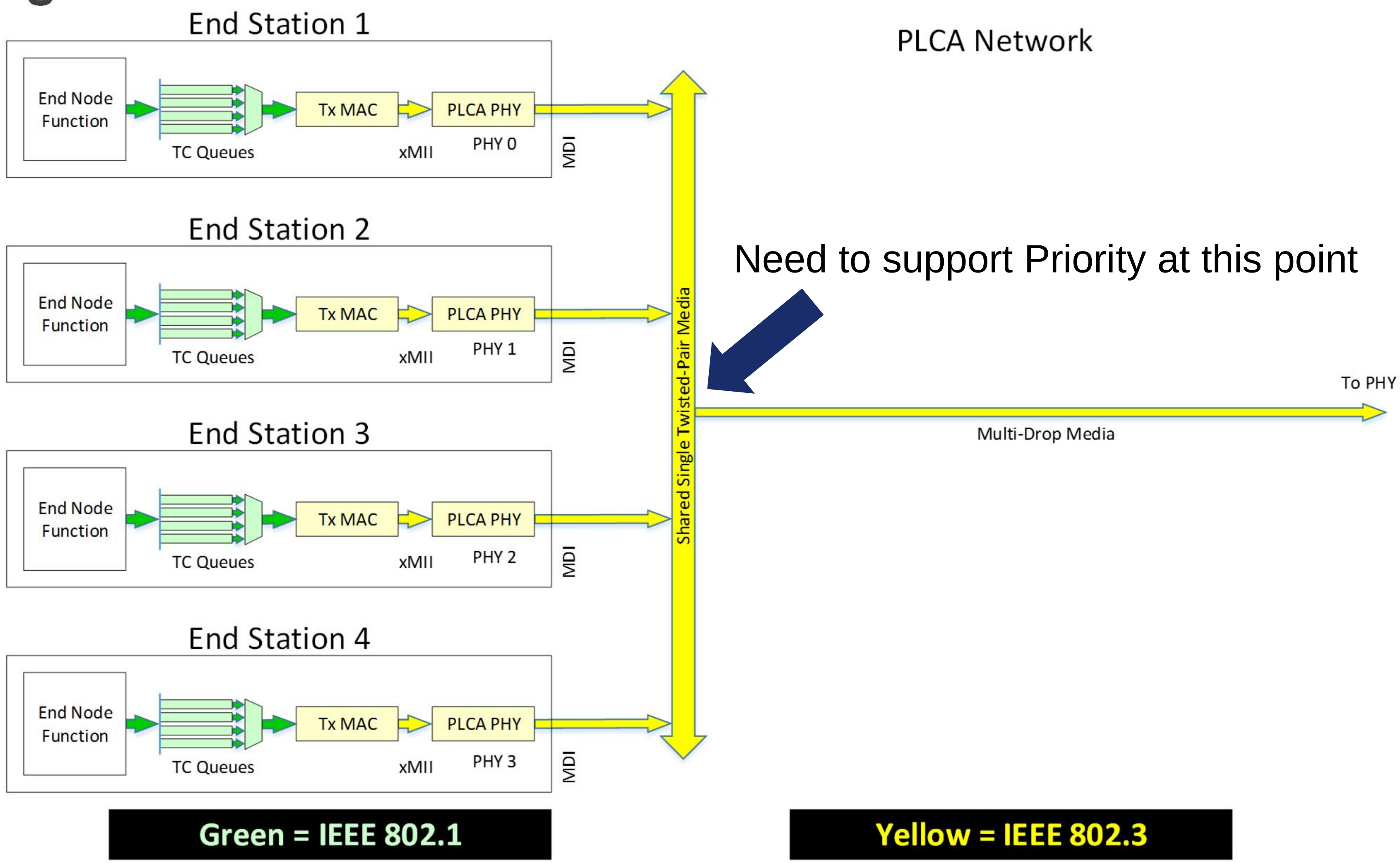


Priority is supported by the 802.1 Queues & Schedulers but not on the PLCA wire!

Green = IEEE 802.1

Yellow = IEEE 802.3

Change the 4 End Stations to 1 Link to Now Use PLCA



Why is Priority support needed?

- PLCA supports equal transmit opportunities between all the nodes on the wire
- It also supports unequal transmit opportunities per node such that you could define:
  - A Node transmit opportunity order of 0, 1, 0, 2, 0, 3, 4, 5 or some other order
  - This example gives Node 0 three times the opportunities compared to the other nodes
- While this approach gives higher opportunities to some <u>nodes</u>, …
- The goal of Priorities is to support different (non-equal) treatment of <u>frames</u>
  - Node 0 may be transmitting only low priority data but it still gets its 3 Tx opportunities!
- The PLCA transmit opportunity schedule is <u>static</u>
  - And this approach must be "engineered" to work right, requiring more 'user' effort
- Priorities support differentiated services <u>dynamically</u> self adjusting as expected!
  - The transmit opportunities of each Node needs to change based on the <u>frame</u> it is Tx'ing
  - Priorities are an integral part of TSN (as long as they follow the 802.1 model)
  - Therefore, there is no extra 'user' effort needed in order to use Priorities

Why is Priority support needed?

- Other ‘solutions’ have been presented that solve the problem of a specific use case by limiting the size of the lower priority frames are controlled (somehow)
- This is an “engineered” solution to a problem that would automatically be solved by using 802.1 TSN methods (i.e., Priorities)
- The TSN target guarantees can only be met if the TSN architecture/methods are used throughout the network
- Making exceptions in certain parts of a network make it really hard, & takes a lot of extra engineering effort to be convinced, that the TSN portion will work as desired

- That being said, any solution must be cost effective, not just in the purchase of components, but also in the cost of using those components
  - More on this later

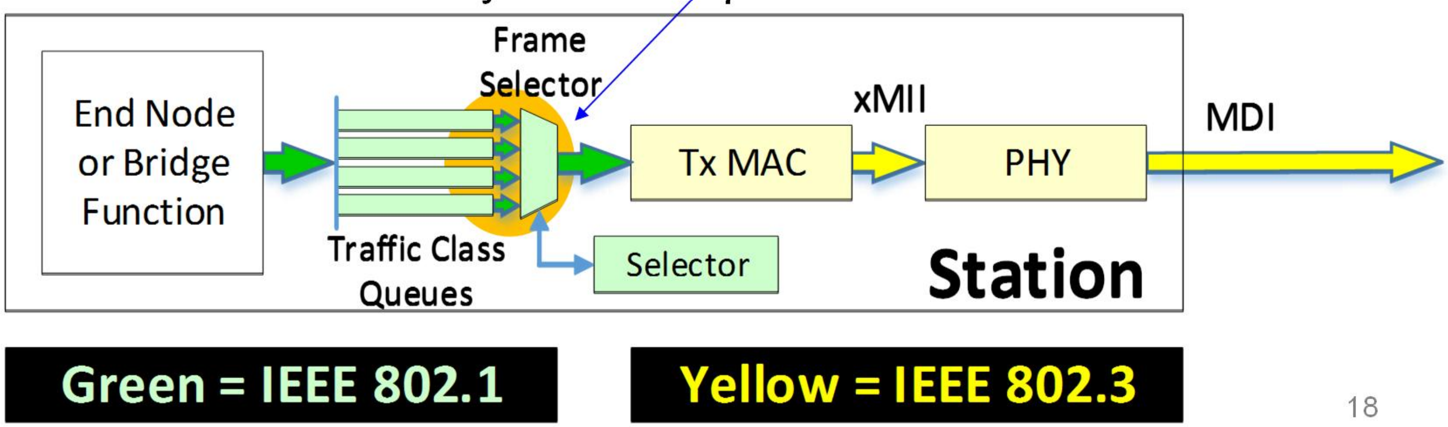
PART 3

Proposal to Support Priorities in PLCA

And just the Priority Information Exchange only!

What is Minimally Needed so 802.1 Can Do the Rest (Policy)?

- Ideally PLCA should mimic as close as possible what is done inside a Bridge
- The Selector in Bridges minimally need to know which Queues (Priorities) have frames ready to be transmitted, and what Queue (Priority) they are in
- That it!
- Based on this information the Selector determines the next frame to transmit
  - This decision (the Policy of which frame goes next) can be quite complex which is why 802.1 has support for many different types of Queue Shapers ahead of the Selector
    - And 802.3 does not need to worry about that part!



Communicating the Priority of the Next Frames to Tx with PLCA

- Before each regular (data) Beacon, an optional Advertise Beacon is issued
  - PHY 0 issues this Advertise Beacon and then indicates the Priority of its ‘frame to Tx’
    - The Priority to advertise of the ‘frame to Tx’ is given to the PLCA function by 802.1
    - The PLCA function sends this Priority down the MII using 1 new code followed by the Priority
    - The PHY then transmits this information similar to how a Beacon is indicated on the wire
    - All the other PHYs see this new code and send the observed Priority up their MII’s
    - The PLCA function sends to 802.1 the received Advertise Beacon and all observed Priorities
  - Each PHY in the normal sequence issues its ‘frame to Tx’ Priority in the same way
  - After all PHYs have advertised their Node’s ‘frame to Tx’ Priority, PHY 0 issues the normal (data) Beacon and the data phase begins
    - At this point in time all the 802.1 Selectors know the current Priority for all the frames that wish to be transmitted on this Beacon cycle
    - Each local Selector then release to the Tx MAC its ‘frame to Tx’ only if it meets the currently configured 802.1 Policy

Communicating the Priority of the Next Frames to Tx over the MII

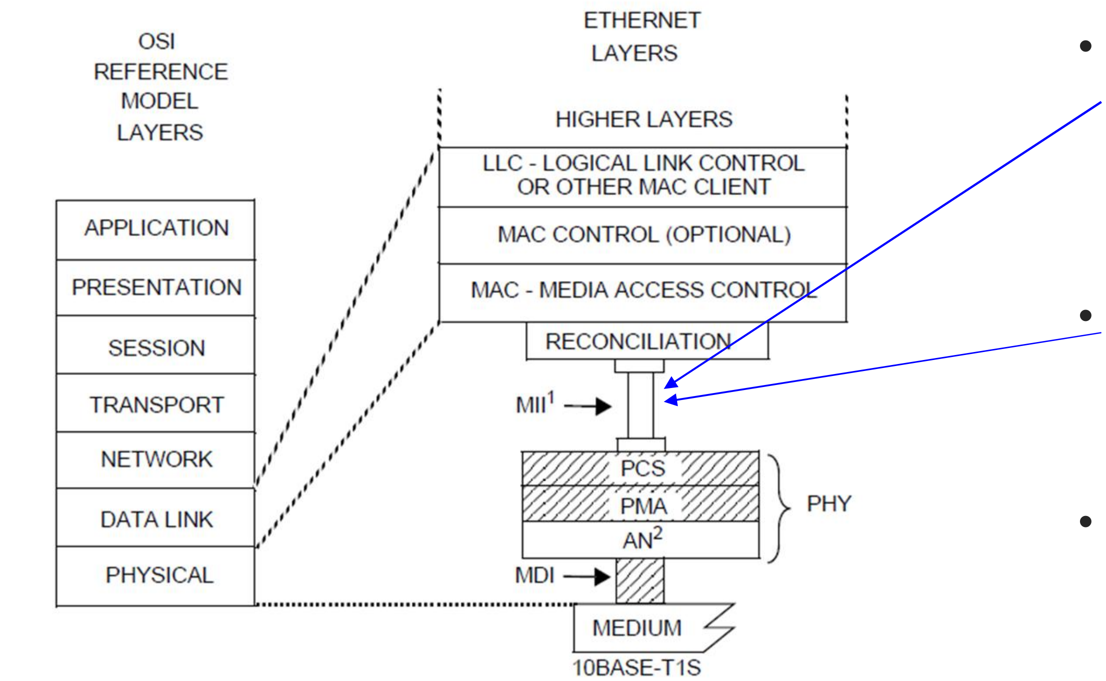
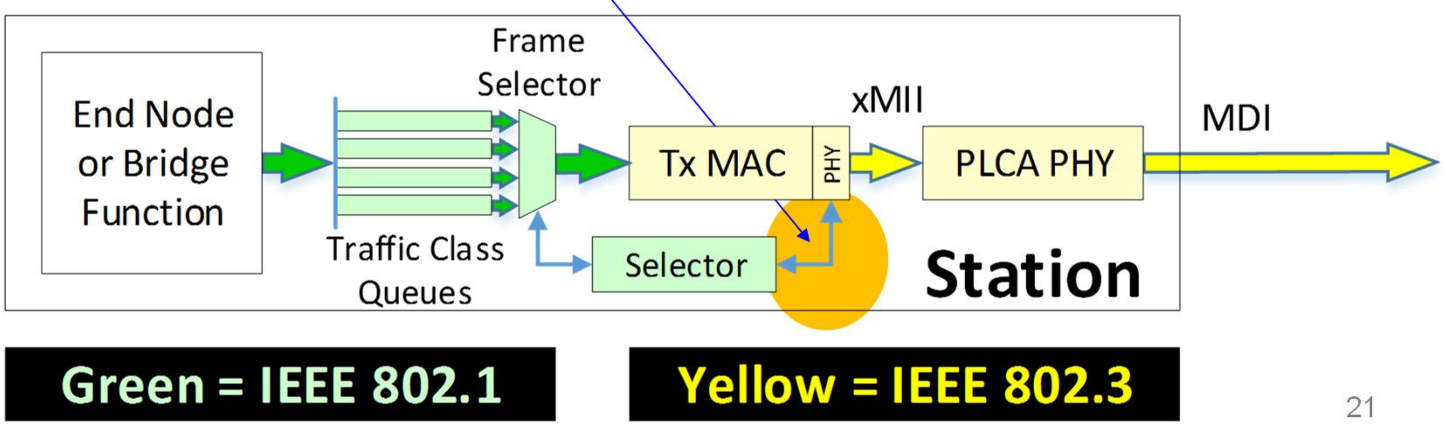


Figure 147–1—Relationship of 10BASE-T1S PHY to the ISO/IEC OSI reference model and the IEEE 802.3 Ethernet Model

- The PLCA function (in the Reconciliation sublayer) issues the locally selected frame’s Priority to advertise down the MII
- And all the neighbor’s received priority they advertised gets sent up the MII
- Then 802.1 can decide if the local frame gets to be transmitted during this Beacon cycle or not
- Creating a new Advertise Beacon allows re-use of existing state machines, saving logic

Communicating the Priority of the Next Frames to Tx to 802.1

- The 802.1 Selector knows what frame it want's to transmit next & gives its Priority to the PLCA function (going around the MAC)
- The 802.1 Selector will 'see' the Advertise Beacon and all the advertised Priorities from itself and all the other Nodes on the multi-drop network segment
- The 802.1 Selector will then send it's selected frame to it's Tx MAC if the frame matches the currently configured 802.1 rules
- A basic rule would be to only release frames that match the highest advertised Priority (this is called Strict Priority)



Communicating the Priority to/from 802.1 Using EEE's Model

78.1.1.1 Reconciliation sublayer service interfaces

Figure 78–1 depicts the LPI Client and the RS interlayer service interfaces.

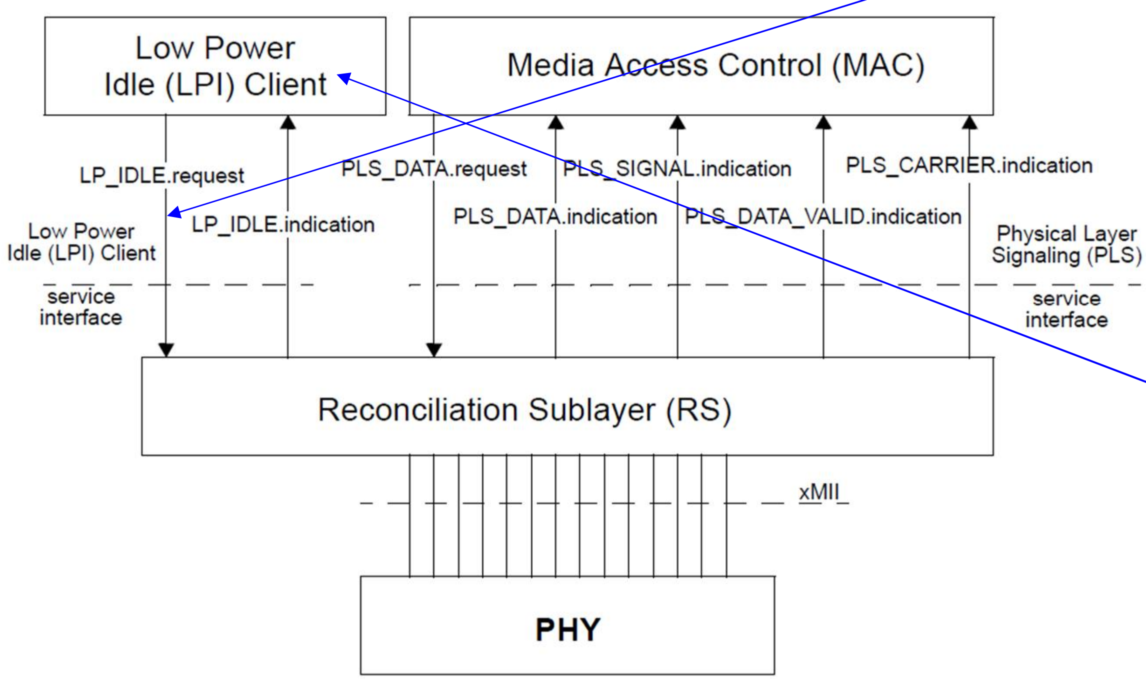
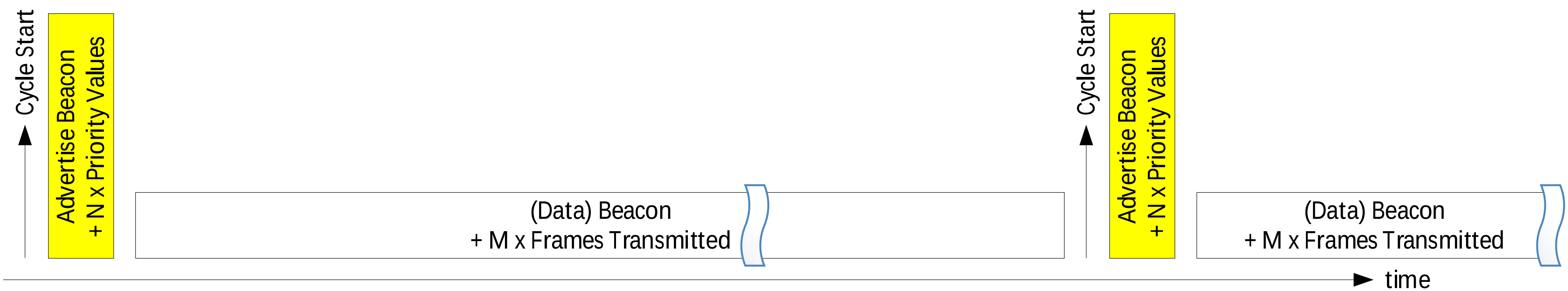


Figure 78–1—LPI Client and RS interlayer service interfaces

- Energy Efficient Ethernet (EEE) needs to know when 802.1's queues are empty so it can power down the PHY
- And it needs to know when 802.1's queues have a frame so it can power up the PHY
- This connection to/from 802.1 is via sideband signals to a Low Power Idle Client so that the MAC is not changed in any way
- The Priorities & Advertise Beacon can use the same method and connect to a Priority Client – going around the MAC!

Timing of the Optional PLCA Priority Advertise Mechanism

- The below figures shows where the Priority Advertise takes place
  - It's at the beginning of each cycle
  - You can think of it as an Advertise Phase before the normal Data Phase of each cycle
- This is <u>NOT drawn to scale</u>! That will be shown next
- The location of the Priority Advertise phase, gets 802.1 the information it needs in time for it to support TSN's differentiated services



Where N = # Nodes, M <= N

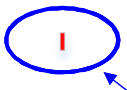
PART 4

Estimated Latency & Logic Costs

Scale Timing of the Optional PLCA Priority Advertise Mechanism

- This figure shows 8 Nodes transmitting maximum size frames + overhead

- This figure shows the added overhead of 8 Nodes of Priority Advertise (to scale)



- The 1st figure is 8 x (~1522+20) = 12,336 bytes
- The 2nd figure is ~32 bytes for Priority Advertise (per discussion w/PLCA Editor)
- The added latency for Priority Advertise with Max size frames is ~ 0.03%

- This figure shows 8 Nodes transmitting minimum size frames + overhead

- This figure is 8 x (64+20) = 672 bytes
- The added latency for Priority Advertise to Min size frames is ~ 4.5%
- In all cases the added latency for Priority Advertise is #Nodes x 4 bytes

Gate Estimate of the Optional PLCA Priority Advertise Mechanism

- The reason to create an Advertise Beacon along with the original Beacon is to re-use as much logic & state machines as possible (per discussion w/ PLCA Editor)

- It is anticipated that this Advertise Beacon logic added to the current PLCA function will add ~ 1000 gates or less (per discussion w/ PLCA Editor)

PART 5

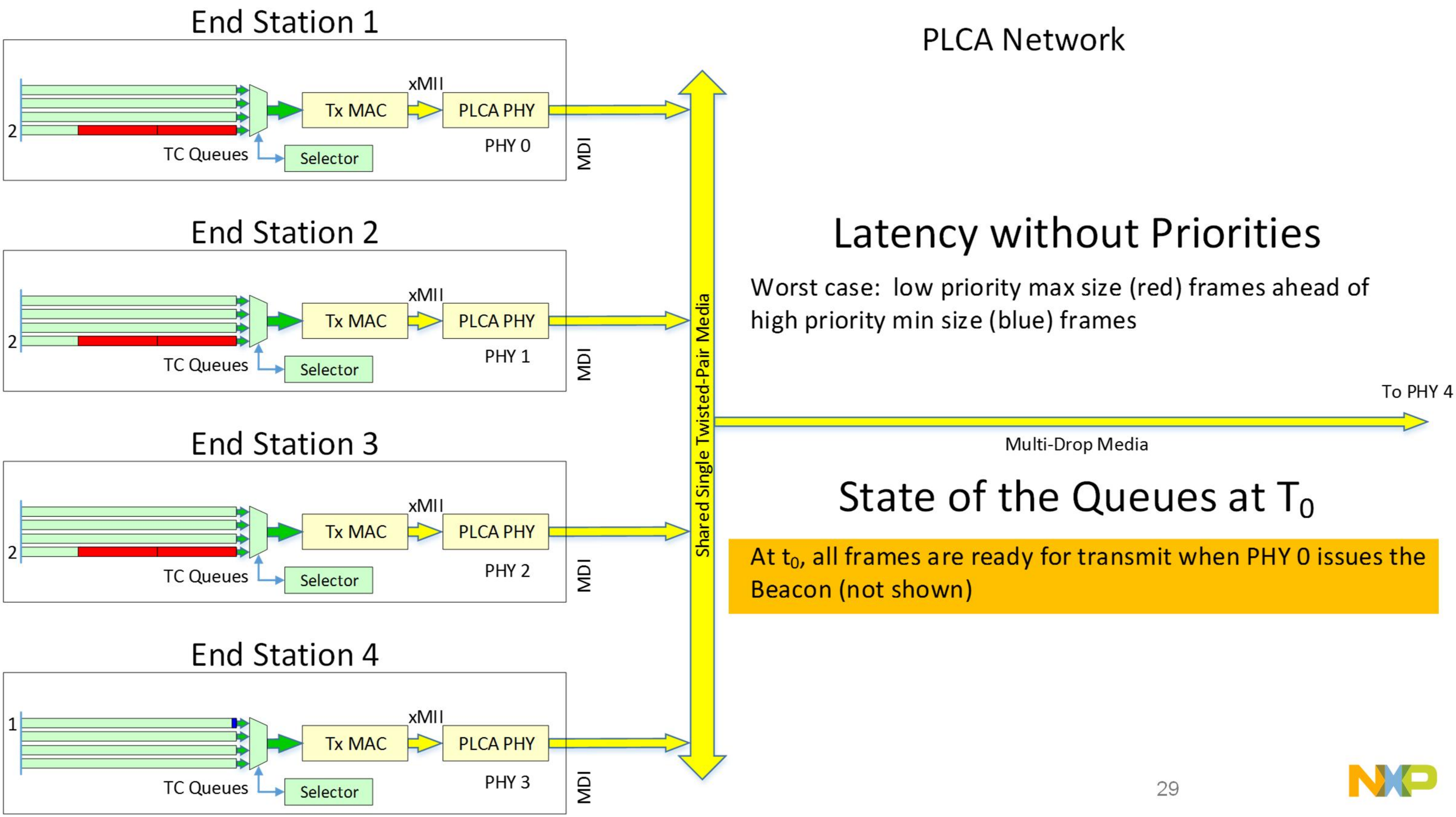
Worst Case Latency Improvement

Worst Case Latency Improvement Example

- 802.1 examines use cases to help determine general needs for a standard improvement
- Then 802.1 likes to look at worst case situations because, as experience has shown, these worst cases ALWAYS show up in networks

- The following example shows 4 End Stations talking to an off-page 5th station
  - The 5th station is not trying to transmit anything so the math is a 4 PHY, 4-to-1 example
    - Makes calculation extrapolations for 8 & 16 node networks easier
  - The 1st three stations have two low priority maximum size frames to transmit
  - And the last station initially has one AVB high priority minimum size frame to transmit

- Again, this is intentionally a worst case setup requiring NO “engineering of the network” as even engineers make mistakes – some which are really hard to find!



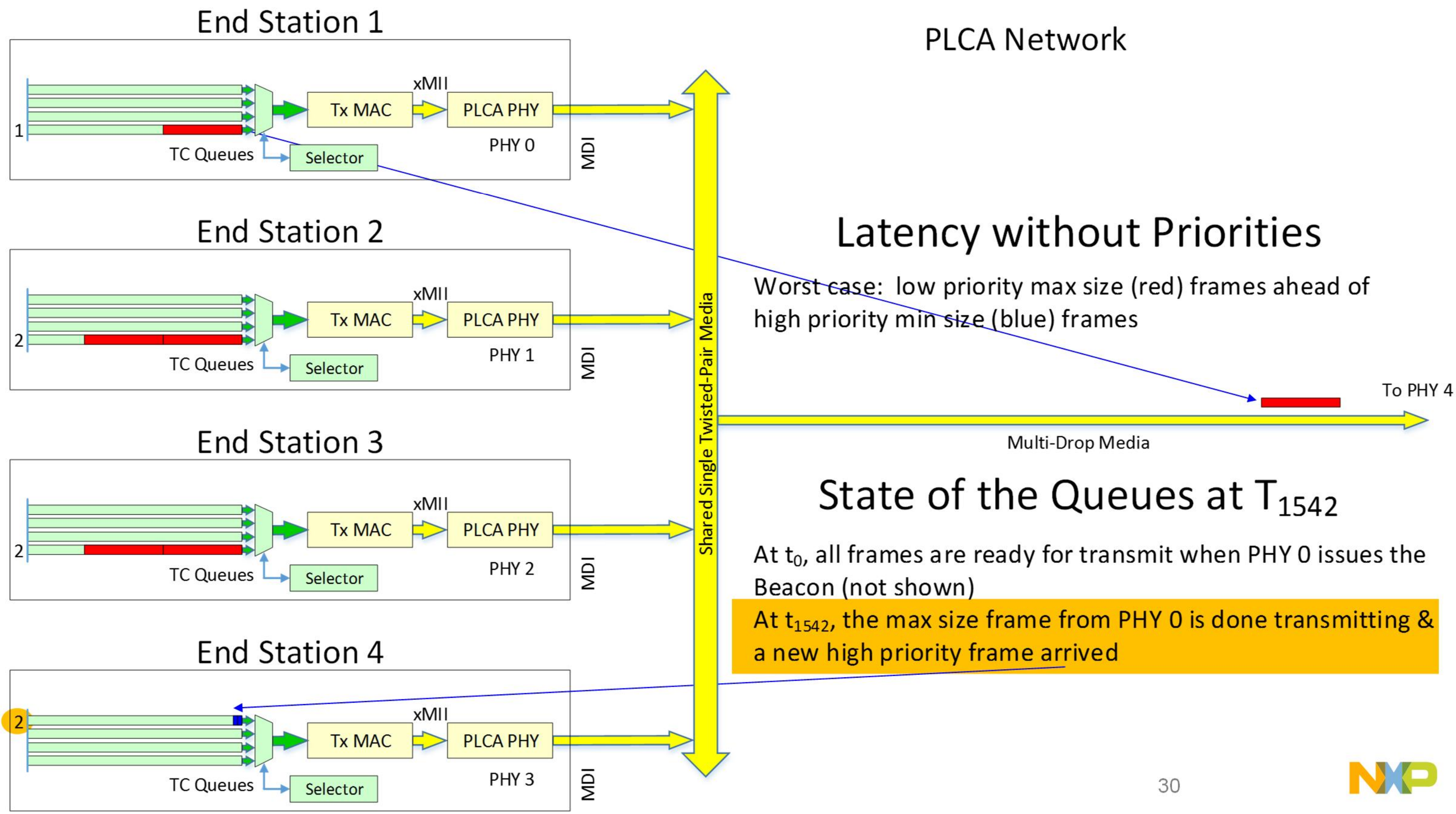
PLCA Network

Latency without Priorities

Worst case: low priority max size (red) frames ahead of high priority min size (blue) frames

State of the Queues at T_0

At t_0, all frames are ready for transmit when PHY 0 issues the Beacon (not shown)



PLCA Network

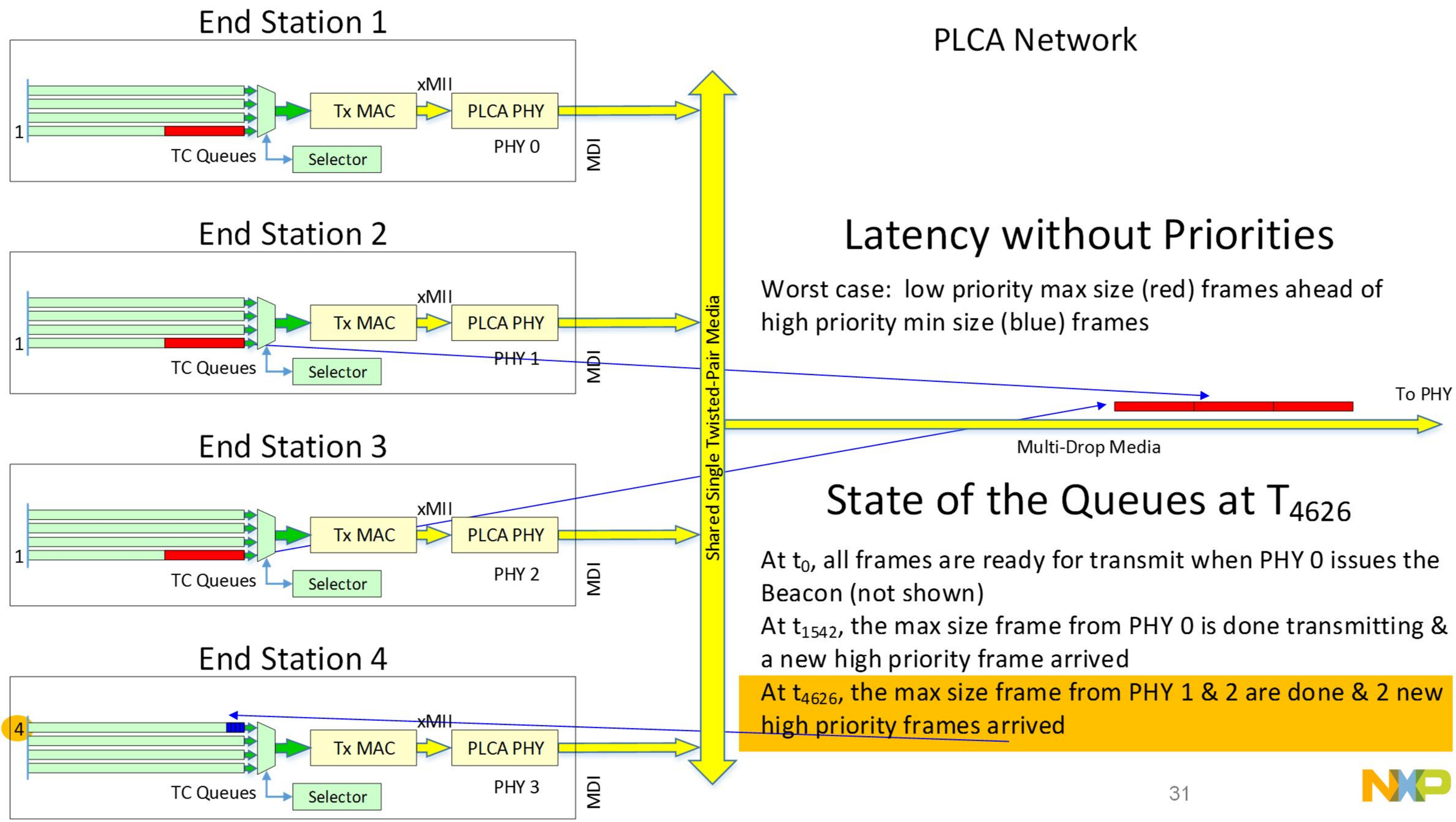
Latency without Priorities

Worst case: low priority max size (red) frames ahead of high priority min size (blue) frames

State of the Queues at T_{1542}

At t_0, all frames are ready for transmit when PHY 0 issues the Beacon (not shown)

At t_{1542}, the max size frame from PHY 0 is done transmitting & a new high priority frame arrived



Latency without Priorities

Worst case: low priority max size (red) frames ahead of high priority min size (blue) frames

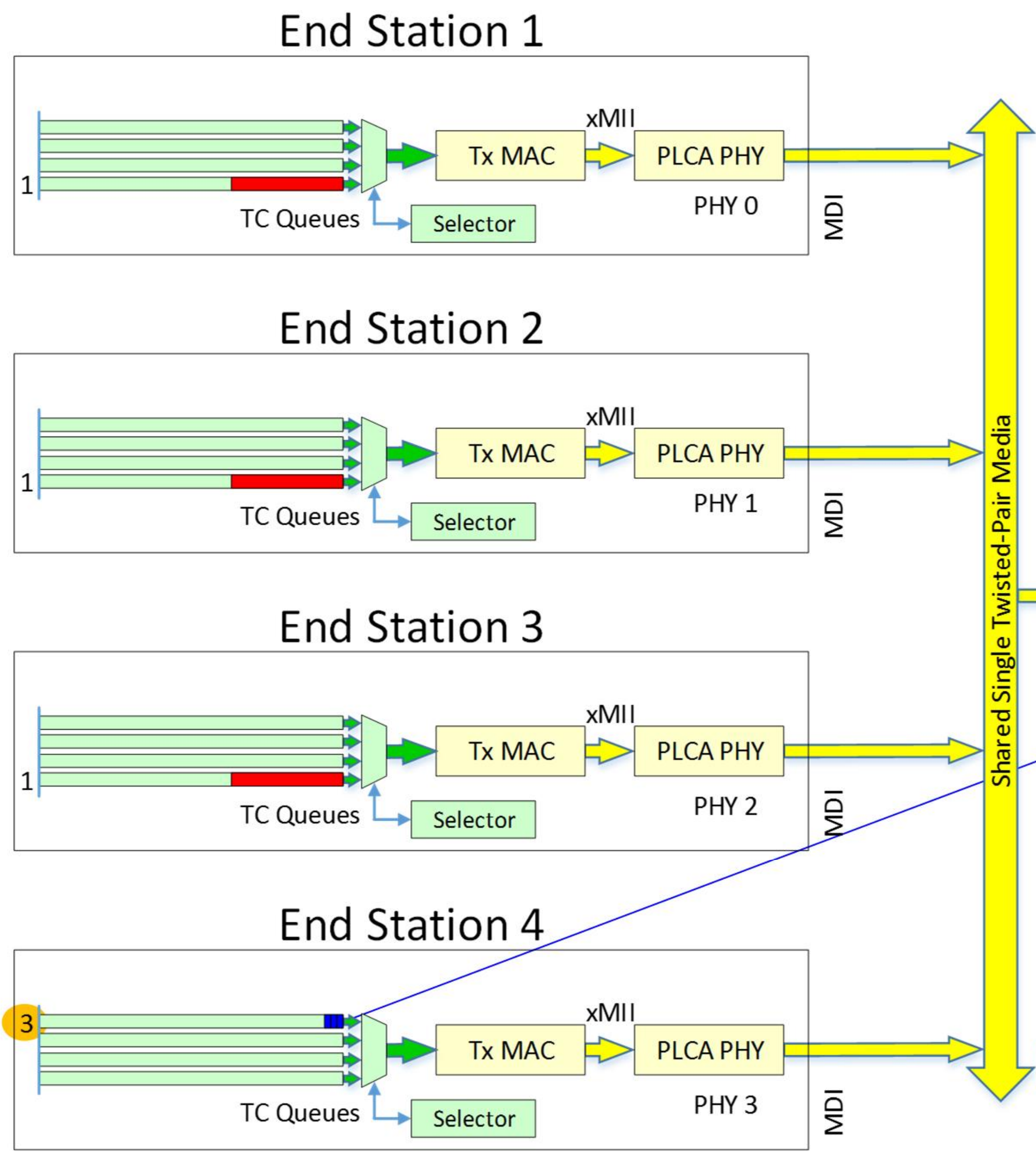
State of the Queues at T_{4626}

At t_0, all frames are ready for transmit when PHY 0 issues the Beacon (not shown)

At t_{1542}, the max size frame from PHY 0 is done transmitting & a new high priority frame arrived

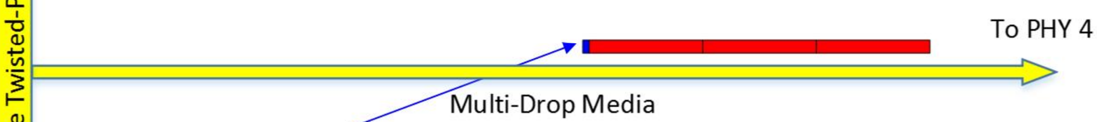
At t_{4626}, the max size frame from PHY 1 & 2 are done & 2 new high priority frames arrived

PLCA Network



Latency without Priorities

Worst case: low priority max size (red) frames ahead of high priority min size (blue) frames



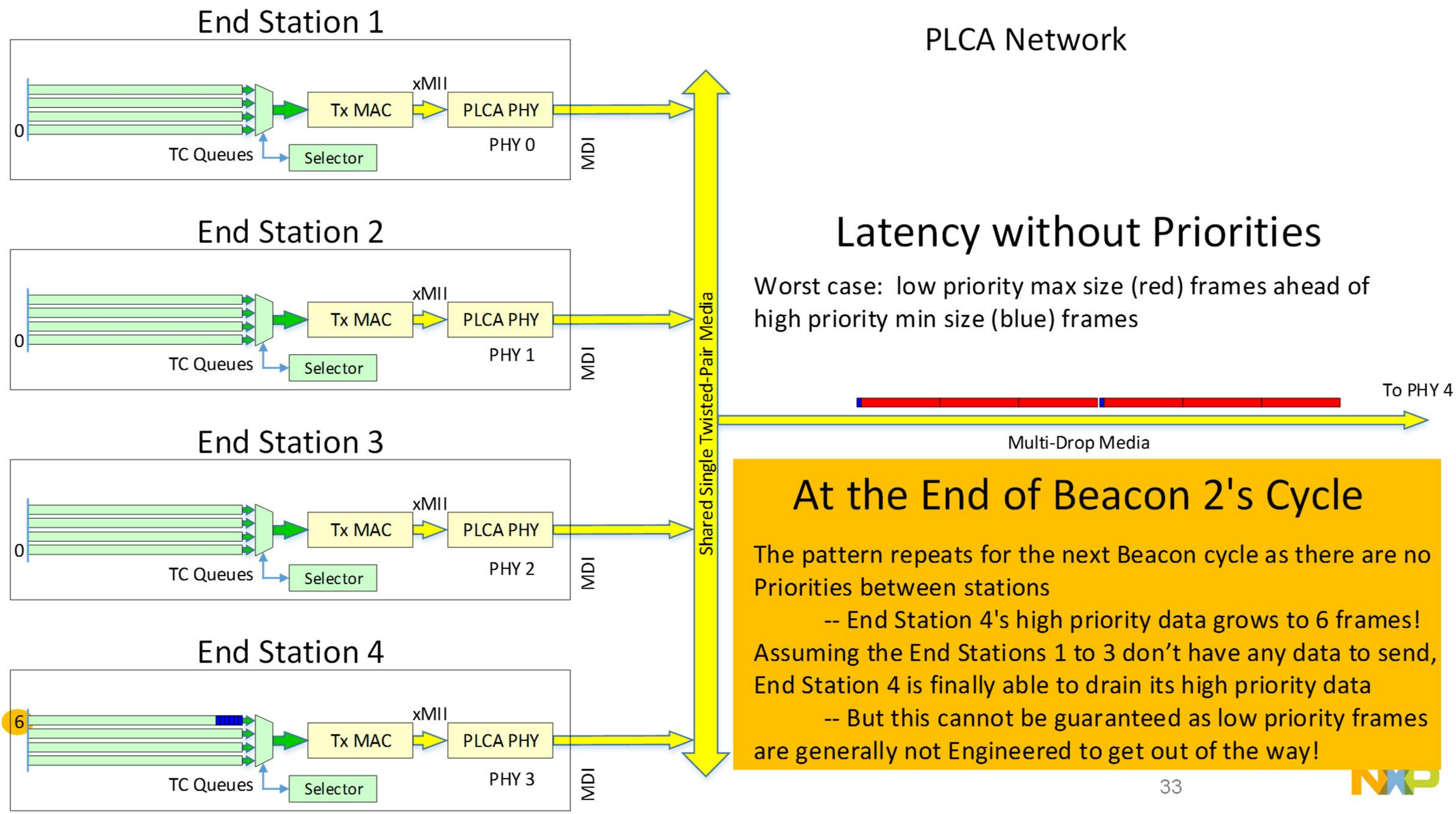
State of the Queues at T_{4690}

At t_0, all frames are ready for transmit when PHY 0 issues the Beacon (not shown)

At t_{1542}, the max size frame from PHY 0 is done transmitting & a new high priority frame arrived

At t_{4626}, the max size frame from PHY 1 & 2 are done & 2 new high priority frames arrived

At t_{4690}, the min size frame from PHY 3 finally gets finished



PLCA Network

Latency without Priorities

Worst case: low priority max size (red) frames ahead of high priority min size (blue) frames

At the End of Beacon 2's Cycle

The pattern repeats for the next Beacon cycle as there are no Priorities between stations

-- End Station 4's high priority data grows to 6 frames!

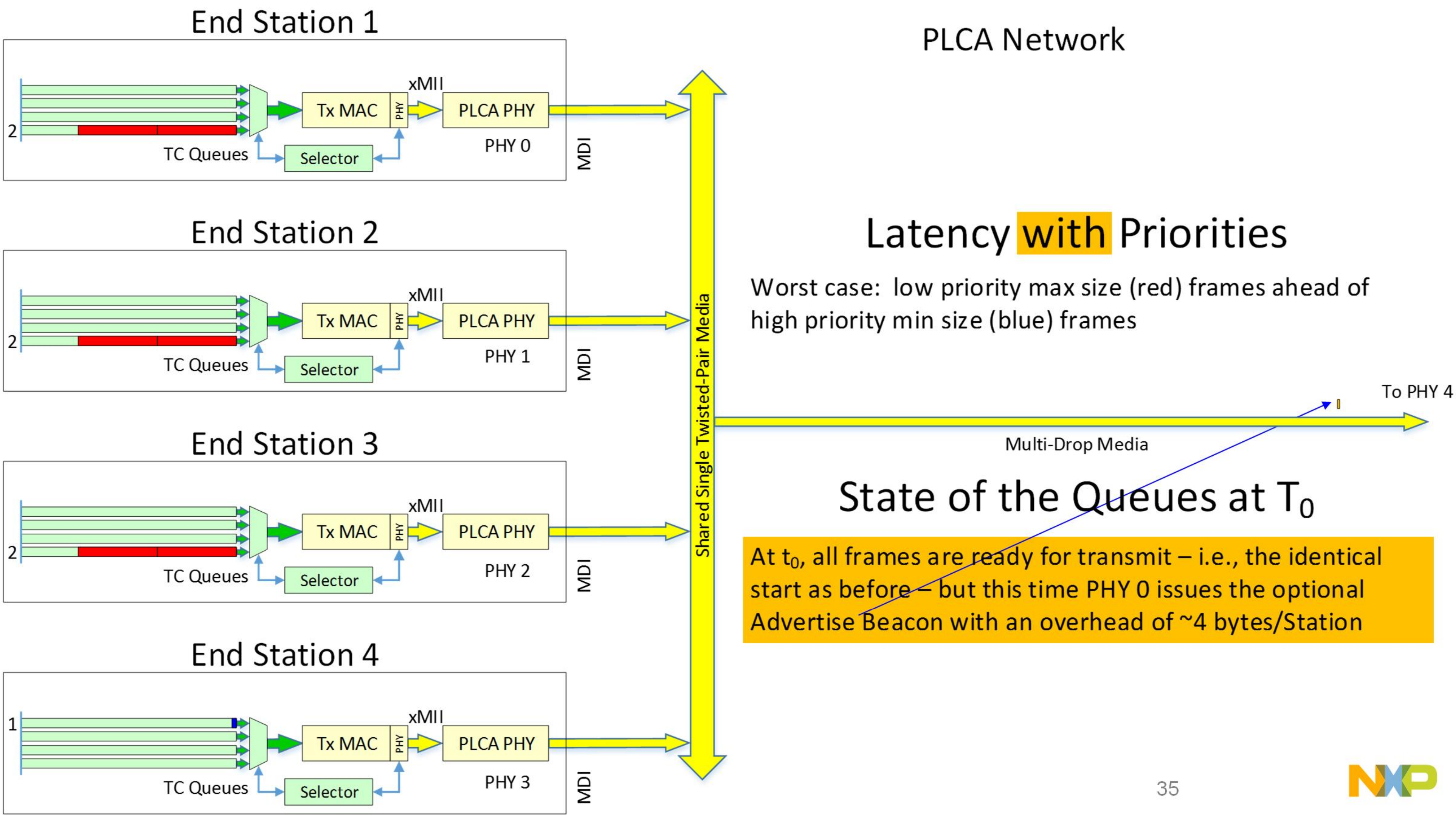
Assuming the End Stations 1 to 3 don't have any data to send, End Station 4 is finally able to drain its high priority data

-- But this cannot be guaranteed as low priority frames are generally not Engineered to get out of the way!

Worst Case Latency Improvement Example – Part 2

- The exact same example is repeated, but this time the optional Priority Advertise is used
- A Strict Priority Selector is assumed to be used in the 802.1 portion in this example
  - Strict Priority was the 1st selector defined by 802.1 in Sep 1998
  - It is the required selector mechanism for the highest priority queues when using AVB (802.1BA-2011)
  - Strict Priority selects only the highest priority frames until there are no more frames of that priority, and then the next lowest priority is chosen, and so on, until there are no more frames in the queues, or a frame appears in a higher priority queue

PLCA Network



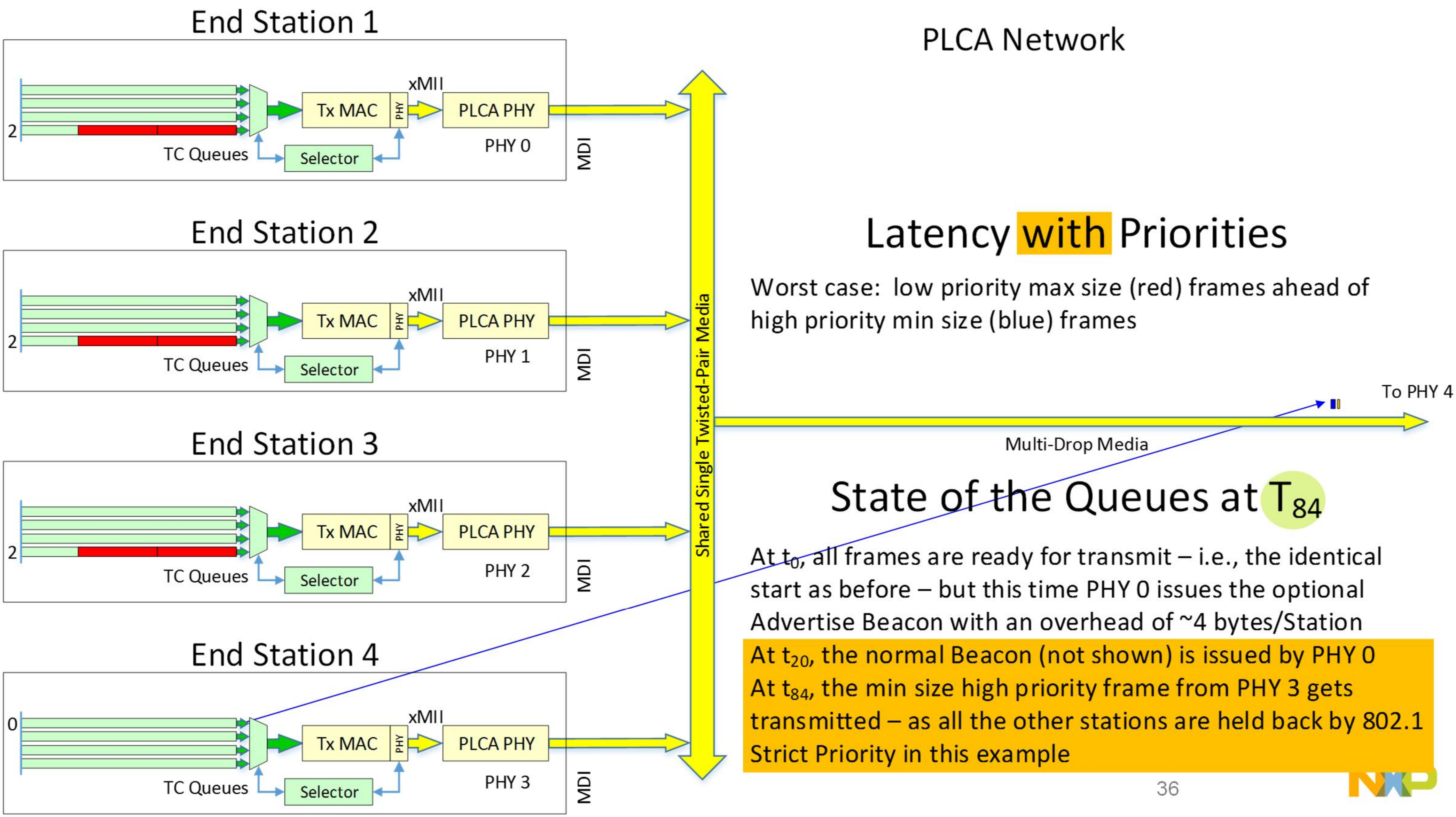
Latency with Priorities

Worst case: low priority max size (red) frames ahead of high priority min size (blue) frames

State of the Queues at T_0

At t_0, all frames are ready for transmit – i.e., the identical start as before – but this time PHY 0 issues the optional Advertise Beacon with an overhead of ~4 bytes/Station

PLCA Network



Latency with Priorities

Worst case: low priority max size (red) frames ahead of high priority min size (blue) frames

State of the Queues at T_{84}

At t_0, all frames are ready for transmit – i.e., the identical start as before – but this time PHY 0 issues the optional Advertise Beacon with an overhead of ~4 bytes/Station

At t_{20}, the normal Beacon (not shown) is issued by PHY 0

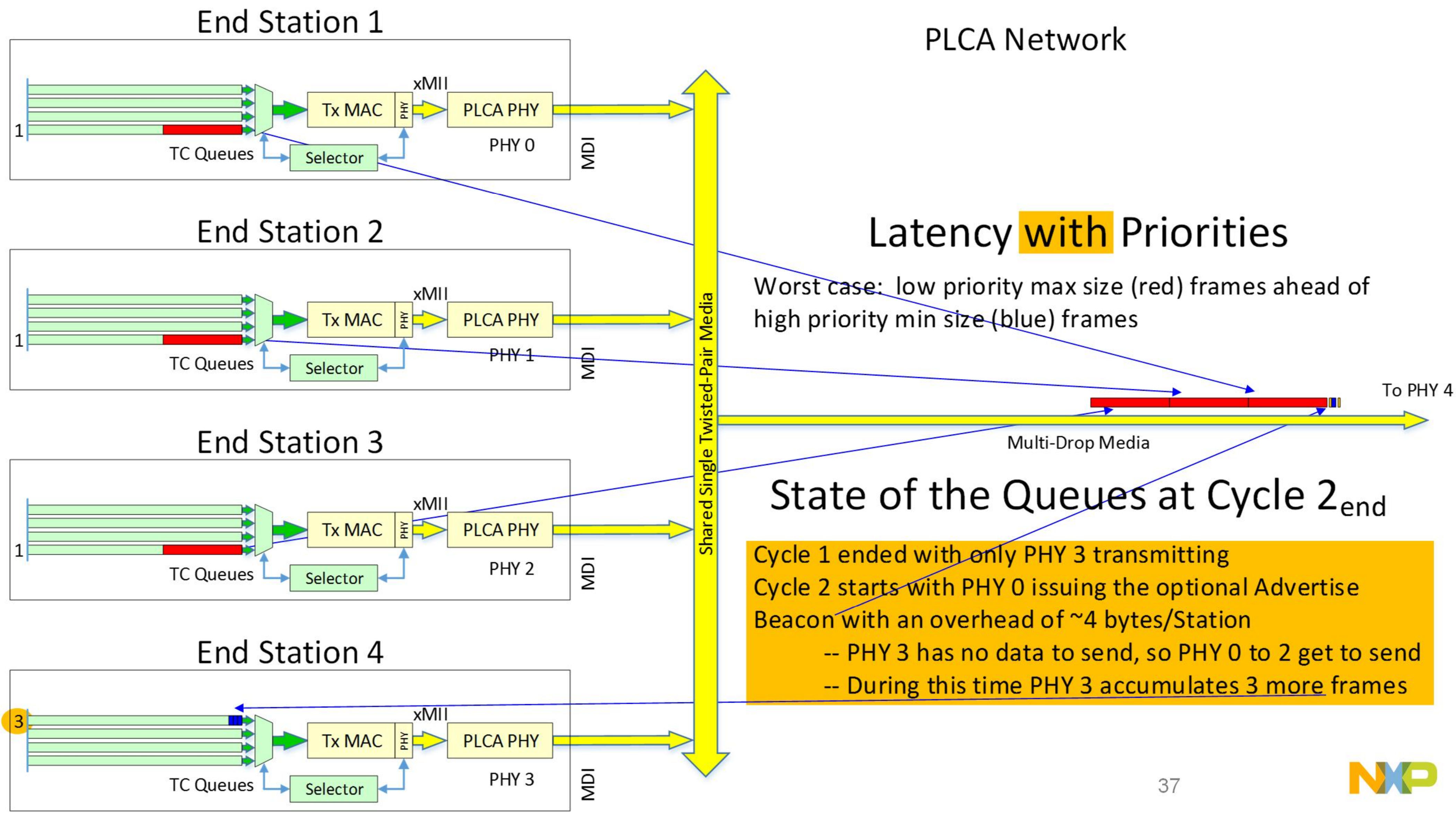
At t_{84}, the min size high priority frame from PHY 3 gets transmitted – as all the other stations are held back by 802.1 Strict Priority in this example

PLCA Network

Latency with Priorities

Worst case: low priority max size (red) frames ahead of high priority min size (blue) frames

State of the Queues at Cycle 2_{end}

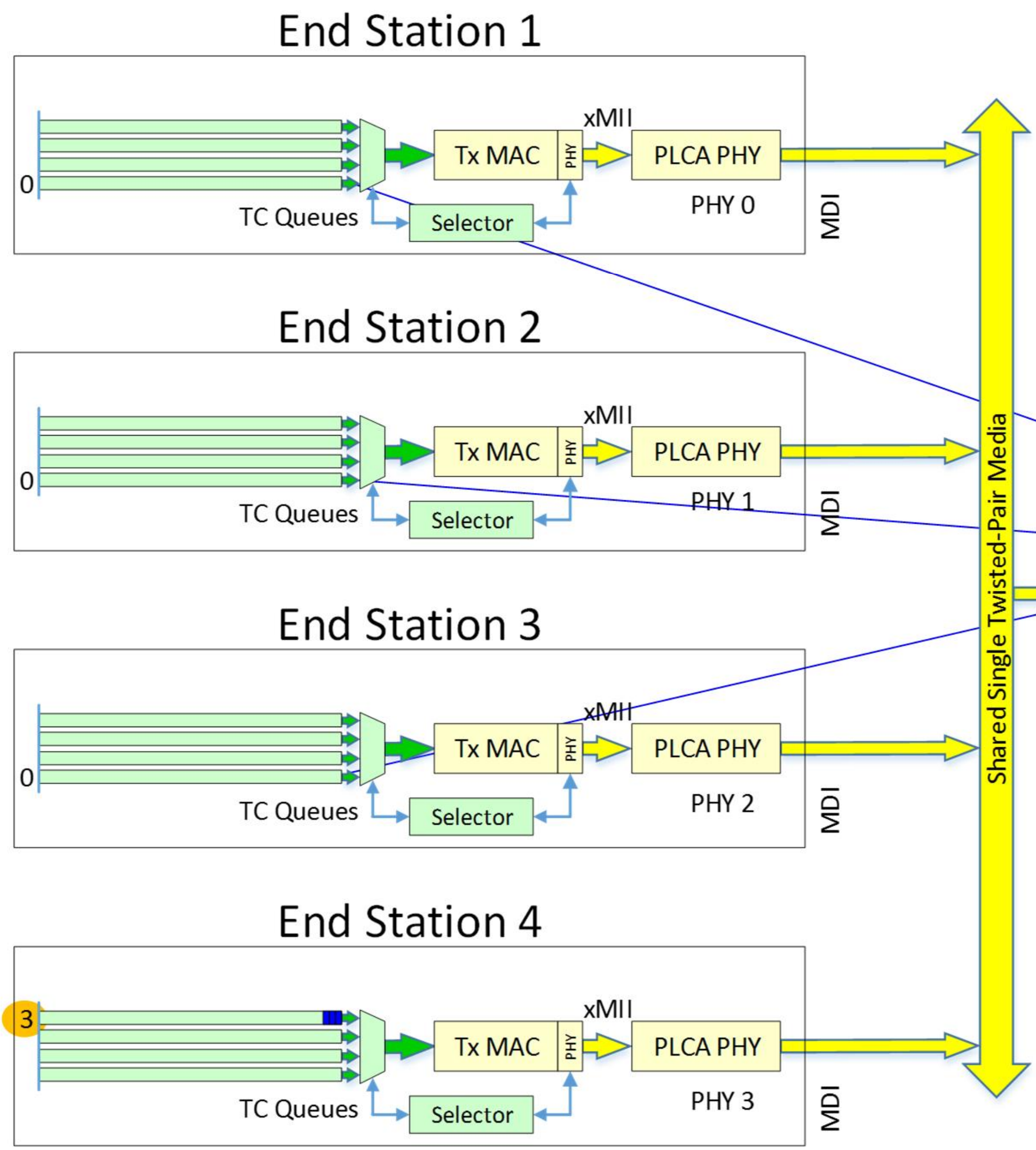


Cycle 1 ended with only PHY 3 transmitting

Cycle 2 starts with PHY 0 issuing the optional Advertise Beacon with an overhead of ~4 bytes/Station

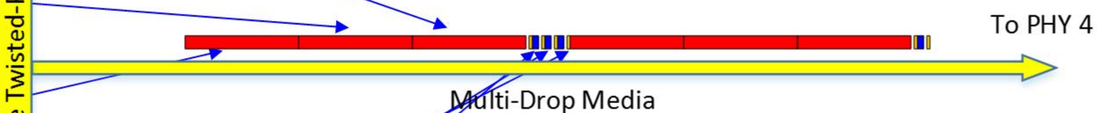
- -- PHY 3 has no data to send, so PHY 0 to 2 get to send
- -- During this time PHY 3 accumulates 3 more frames

PLCA Network



Latency with Priorities

Worst case: low priority max size (red) frames ahead of high priority min size (blue) frames



State of the Queues at Cycle 6_{end}

Cycle 3, 4 & 5 drain the 3 pending high priority frames from PHY 3 – like AVB's credit based shaper bandwidth 'catch-up'

Cycle 6 is a repeat of Cycle 2, allowing the low priority maximum size frames out while PHY 3 accumulates 3 more high priority frames

-- and the pattern could repeat like this

Latency Improvement of the PLCA Priority Advertise Mechanism

- Without the PLCA Priority Advertise (4 PHY, 4-to-1 example):
  - The last byte of the high priority AVB frame was received after: 4690 byte times
  - If the interfering frames continue for subsequent Beacon cycles, this 4690 byte times of delay can repeat for each AVB frame, never allowing, or at least delaying the needed TSN bandwidth ‘catch-up’ supported by the AVB 802.1Qav Credit Based Shaper
  - Bandwidth guarantees, bounded latencies & buffer requirements can’t be determined!

- With the PLCA Priority Advertise (4 PHY, 4-to-1 example):
  - The last byte of the high priority AVB frame was received after: 84 byte times
  - An improvement of 4606 byte times! (subject to the Advertise codes chosen)
  - Just as important this approach supports the very important TSN bandwidth ‘catch-up’ required by the AVB 802.1Qav Credit Based Shaper
  - This approach supports guaranteed bandwidth allocations with defined bounded latencies, two critical aspects of TSN!

PART 6
Things Still to Do

Things Still to Do

- Propose the actual text changes for the next draft which was listed in the comment resolution (#573 http://www.ieee802.org/3/cg/comments/802.3cg_draft2p0_Received_By%20Comment%20ID.pdf):
  - 1) Add a new PRIORITY encoding to Tables 22-1 & 22-2 (the MII interface - p25 & p26). A single encoding is all that is needed as the Priority value indication can follow the PRIORITY code.
  - 2) Add PRIORITY 4B/5B encoding to Table 147-1 (p151) or some other mechanism.
  - 3) Update figure 148-3 (p176) to add connections to a "Priority Client" as was done for Energy Efficient Ethernet's Fig 78-1 (p33 of part 6 of 802.3-2015).
  - 4) Update Fig 148-4 (p181) PLCA Control state diagram and associated text to add in the optional Priority communication phase at the start of each BEACON. The goal here is to reuse as much as possible to minimize gate costs.
  - 5) A register bit will be needed to enable this optional feature, a few PICS added, etc.

PART 7
Summary

Summary

- Adding Priority Advertise support to PLCA is:
  - An optional feature for multi-drop TS1 only – this does not apply to T1L
  - It's expected gate cost (~1000 gates) is not measurable in $'s using today's geometries
  - It's latency overhead is tiny when enabled (~ 4 byte times/Node)
  - It's latency overhead is zero when disabled
- It is needed in order to fully support TSN's differentiated services per traffic type
  - It's as close to the mechanism used in Bridges as it can be
  - And it supports the lowest possible latencies for PLCA's multi-drop mode
- 802.3 only needs to add the mechanism to communicate the Priorities
- 802.1 will do all the policy mechanisms to select which frames are transmitted
  - Initially all that is needed is a Strict Selector which has been defined since 1998
- The MAC and its interfaces are not modified!

Summary – part 2

- Some have stated that this is too complex to add to 802.3cg at that this time
  - It was too complex before, but it is now quite simple
  - All that is needed in 802.3 is the communication of Priority
- Some are concerned that there is no 802.1 project to support Priority Advertise
  - Here we have the famous "chicken and egg – which came first" issue
  - 802.3's EEE did not wait for an 802.1 project to define what connects to the Low Power Idle Client – the work needed to get done and it got done
  - 802.1's gPTP did not wait for an 802.3 project to define how to timestamp frames in a PHY – the 802.1 work completed and then an 802.3 project defined the specifics
  - 802.3cg's addition of Priority Advertise can complete without contingencies on 802.1
  - And the list of Supporters shows that getting a project started in 802.1 will likely happen
- The time to add Priority Advertise is now – as nobody likes to redefine a PHY

Thanks

Questions?