USB4™ Time Synchronization, Host Interface, Connection Manager and Thunderbolt™ 3 Compatibility

Uri Hermoni – *Architect, Intel Corporation*

Gal Yedidia – *Architect, Intel Corporation*

USB Developer Days 2019 – Taipei, Taiwan

November 20, 2019

Agenda

- Time Synchronization
- Host Interface
- Connection Manager
- Thunderbolt™ 3 Compatibility

Agenda

- Time Synchronization
- Host Interface
- Connection Manager
- Thunderbolt™ 3 Compatibility

Time Sync Agenda

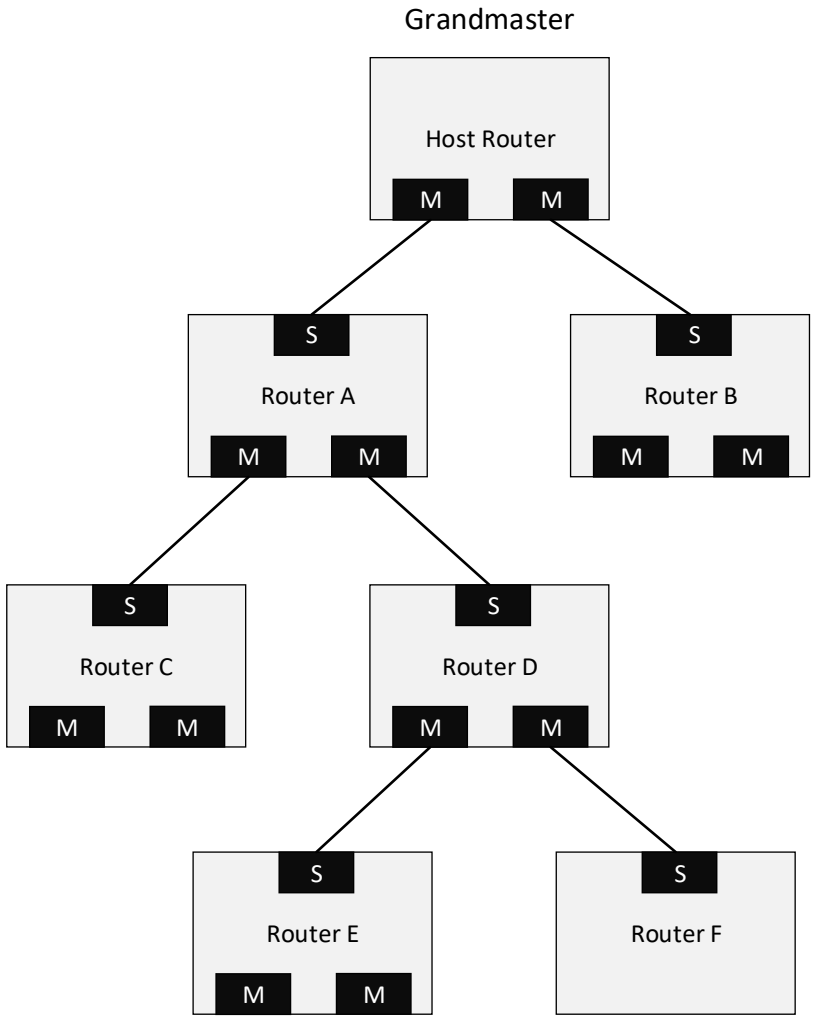
- Why Time Synchronization is needed?
- Time Synchronization Mechanism
  - Time Sync Operation Modes
  - Time Calculation
  - Recursive Propagation
- Use of Time Synchronization in tunneled protocols

Why Time Synchronization is Needed?

- DisplayPort Tunneling
  - Link Clock restoration in the Monitor side based on the GPU side
  - Need a Grandmaster Time to measure the two clocks and adjust
- USB3 Tunneling
  - Isochronous Timestamp Packets correction
  - *Delta* should include the tunnel delay
- PCIe Tunneling
  - Precision Time Measurement messages requires time sense
  - Handshake transforms to use the Grandmaster Time

Time Synchronization Mechanism

- Each Router has its own clock (>125MHz±100ppm)
- Host Router holds the Grandmaster Time
- Each Router holds a Local Time Counter and need to calculate
  - TimeOffsetFromGM
  - FreqOffsetFromGM

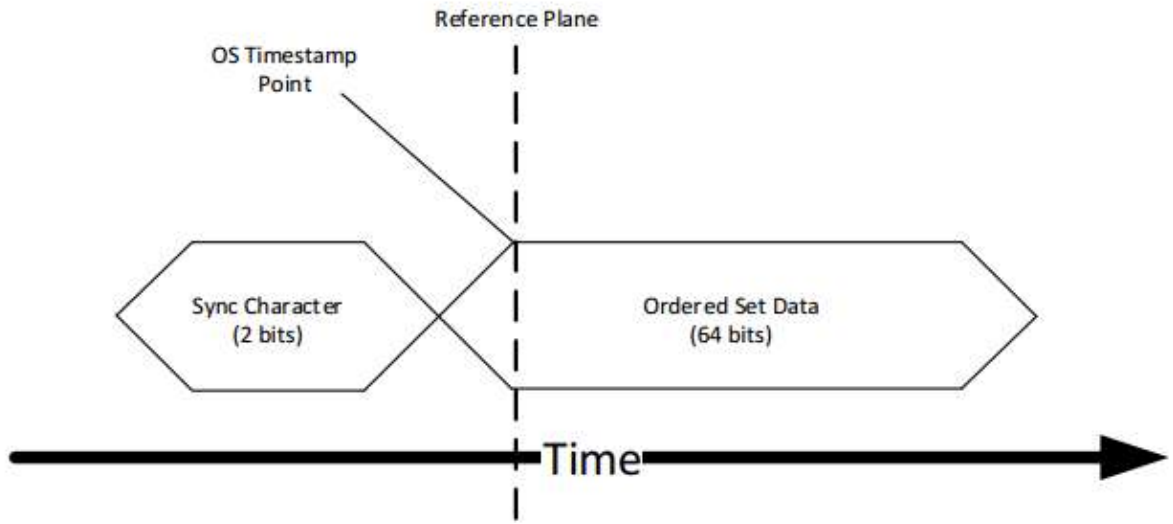


Time Sync Operation Modes

- There are two types
  - Bi-directional
  - Uni-directional
- There are two resolutions
  - HiFi
  - LowRes

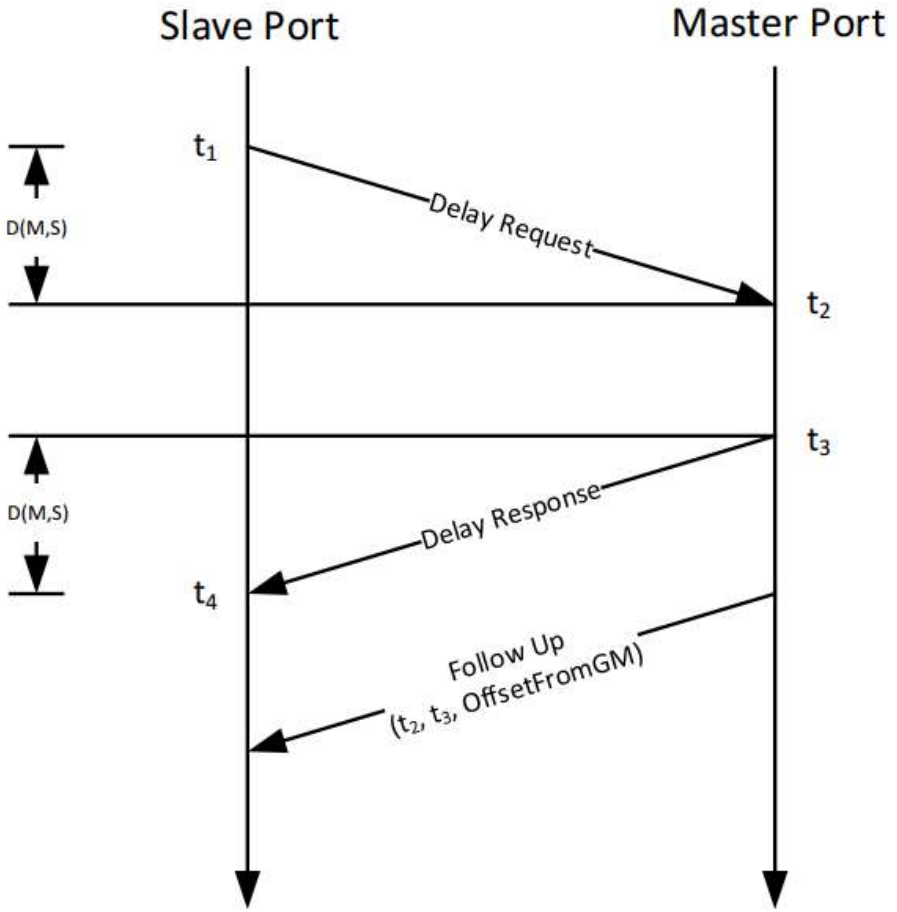
Resolution	Uni-directional	Bi-directional
LowRes	TSNOS every 1ms	N/A
HiFi	TSNOS every 16us	2-way handshake every 16us

- Time Sync Notification Ordered Set (TSNOS) te
  Router when to timestamp
- Information than sent on the Follow Up Packet



Bi-directional Time Sync

- Initiated by Slave
- Periodically send TSNOS every 16us
- Slave timestamp when TX and RX TSNOS
- Master timestamp when RX, immediately responds and timestamp when TX
- Follow Up packet holds information from Master to Slave



Bi-directional Time Sync

- Static latency between Slave and Master

$$D = \frac{(T_4 - T_1) - (T_3 - T_2)}{2}$$

- Time offset between Master and Slave

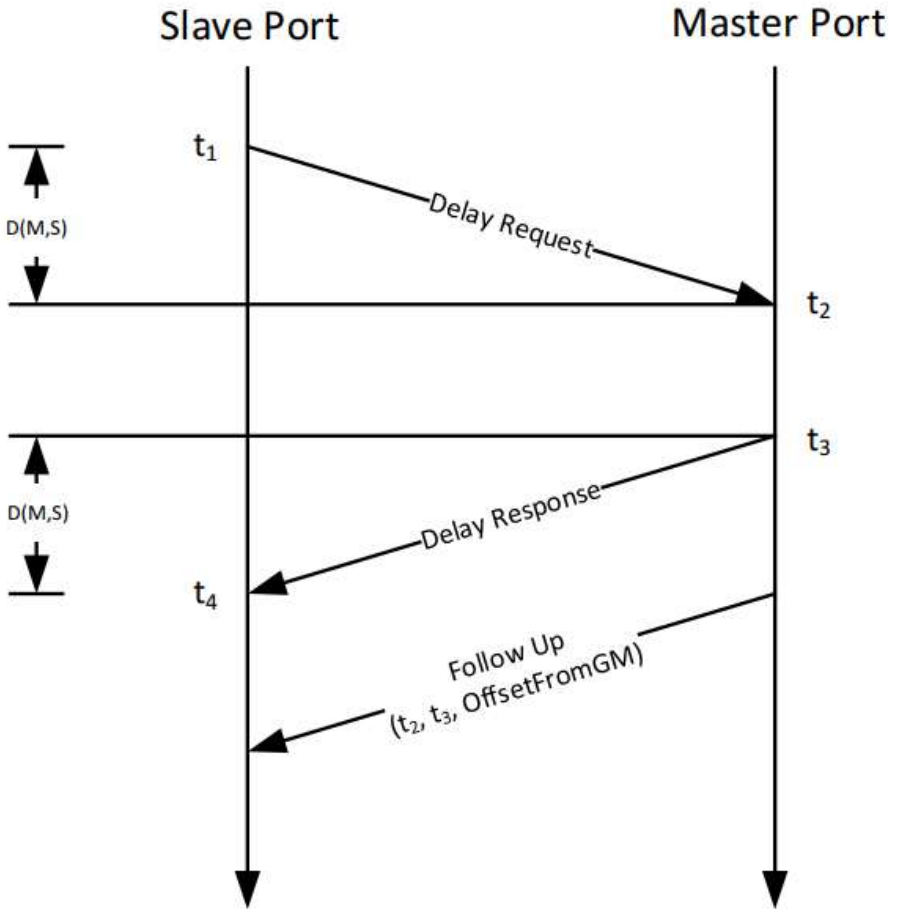
$$TimeOffset = T_3 - T_4 + D$$

- Frequency ratio between Master and Slave

$$FrequencyRatio = \frac{T_4[n] - T_4[m]}{T_3[n] - T_3[m]}$$

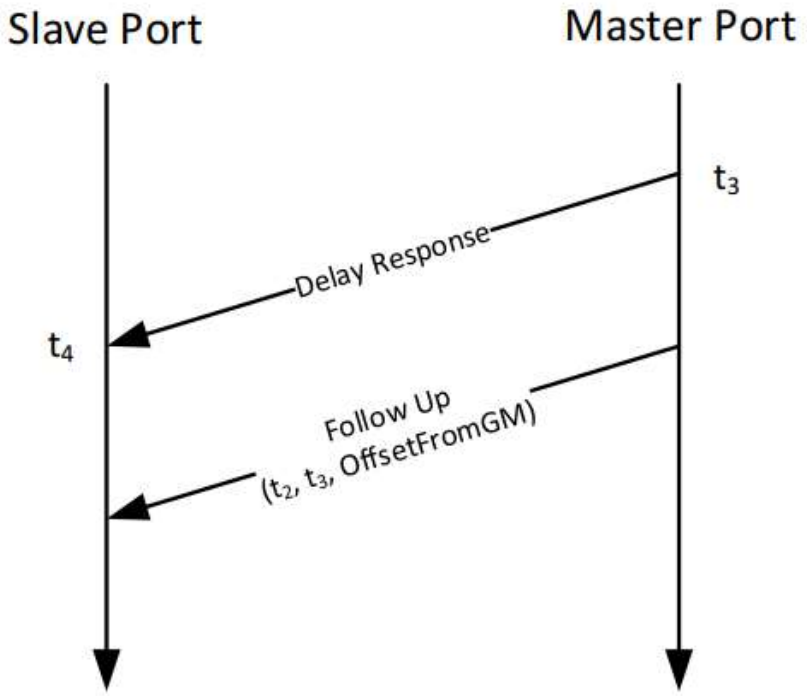
- Frequency offset between Master and Slave

$$FrequencyOffset = (FrequencyRatio - 1) \times 2^{41}$$



Uni-directional Time Sync

- Initiated by Master
- Periodically send TSNOS every 16us/1ms
- Master timestamp when TX TSNOS
- Slave timestamp when RX TSNOS
- Follow Up packet holds information from Master to Slave
- T_2 is empty



Uni-directional Time Sync

- Static latency between Slave and Master can't be measured
- Time offset between Master and Slave

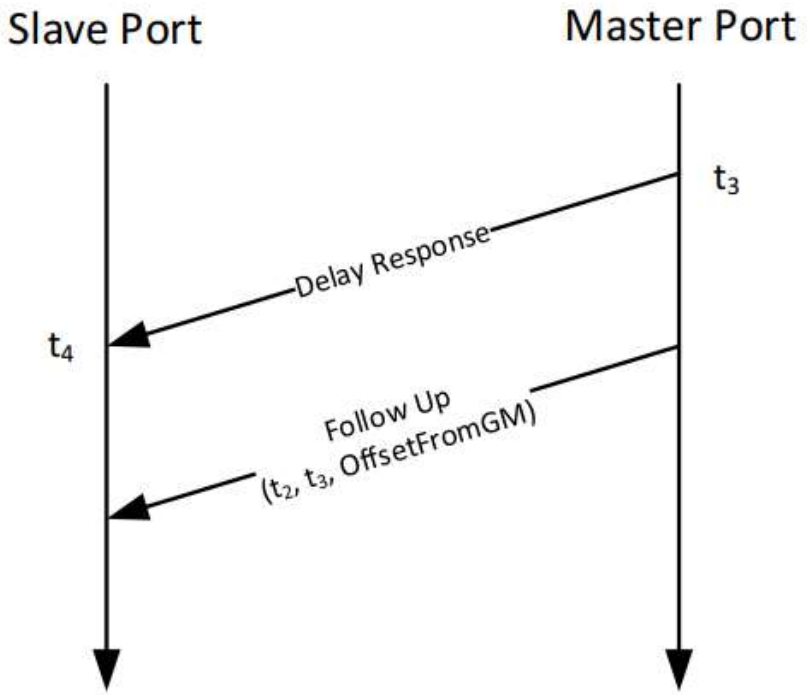
$$TimeOffset = T_3 - T_4 + D$$

- Frequency ratio between Master and Slave

$$FrequencyRatio = \frac{T_4[n] - T_4[m]}{T_3[n] - T_3[m]}$$

- Frequency offset between Master and Slave

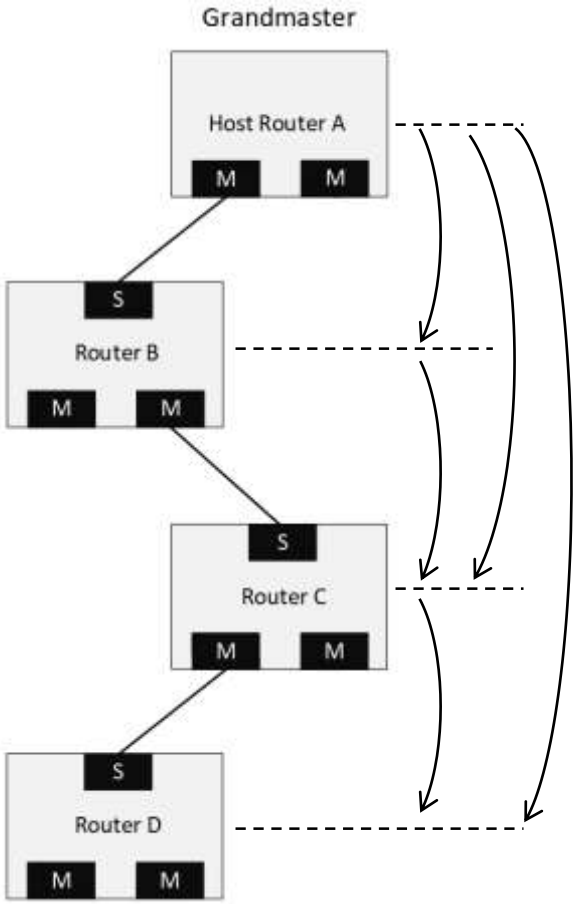
$$FrequencyOffset = (FrequencyRatio - 1) \times 2^{41}$$



Recursive Propagation

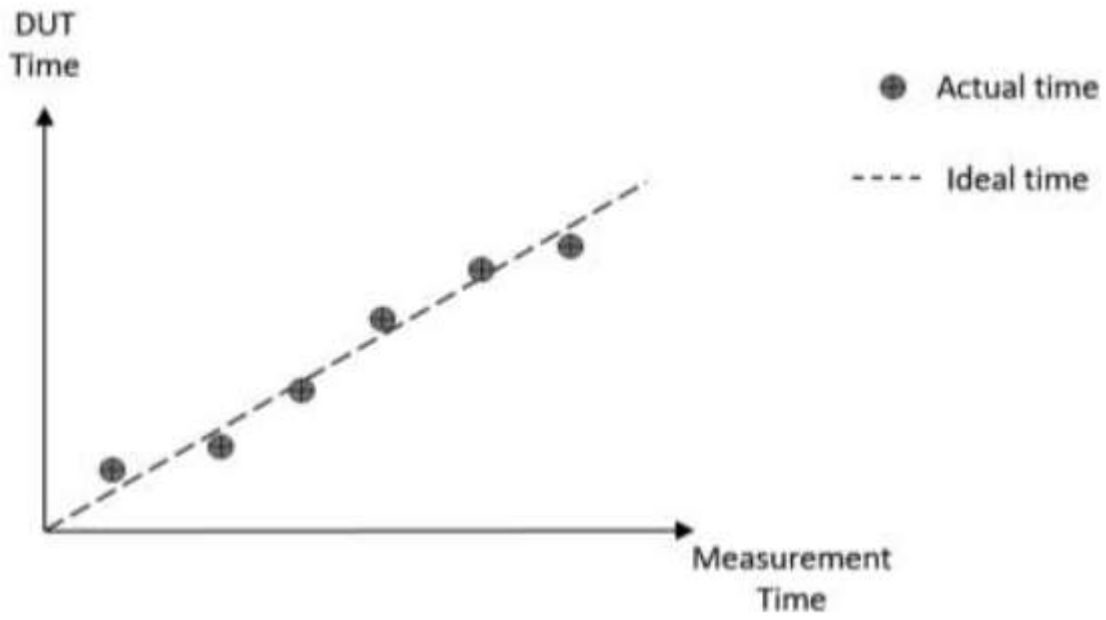
- Each Slave Router calculate offsets from Master Router
- *OffsetFromGM* is sent as part of the handshake
  - TimeOffsetFromGM
  - FreqOffsetFromGM
- Each Router can calculate offset from Grandmaster

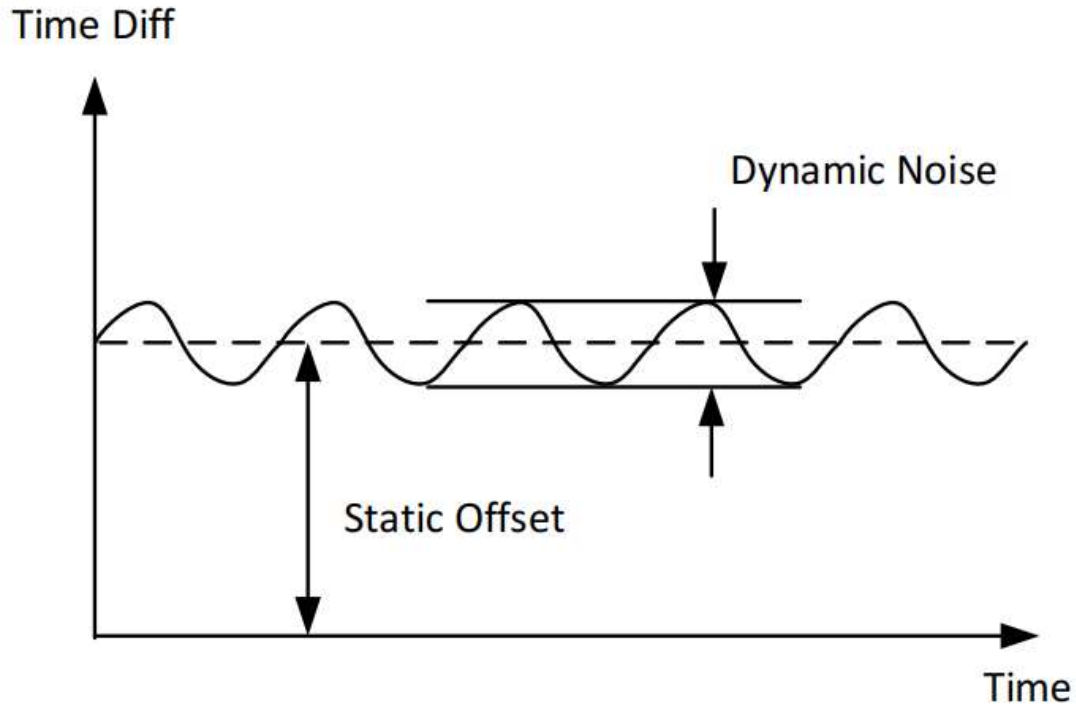
- In Inter-Domain systems there is a special handshake to allow Time Sync between two Hosts



Time Accuracy & Convergence

- Accuracy depends on *TSPacketInterval* (1ns/4ns)
- Convergence should be within *tConvergeTime*
- Results are filtered to improve accuracy



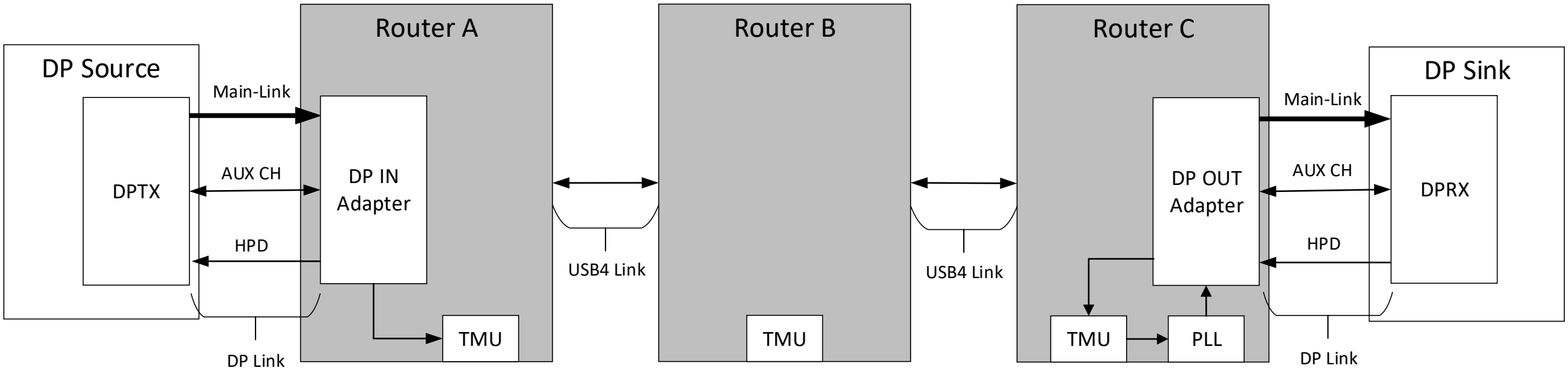


Time Sync Agenda

- Why Time Synchronization is needed?
- Time Synchronization Mechanism
  - Time Sync Operation Modes
  - Time Calculation
  - Recursive Propagation
- Use of Time Synchronization in tunneled protocols

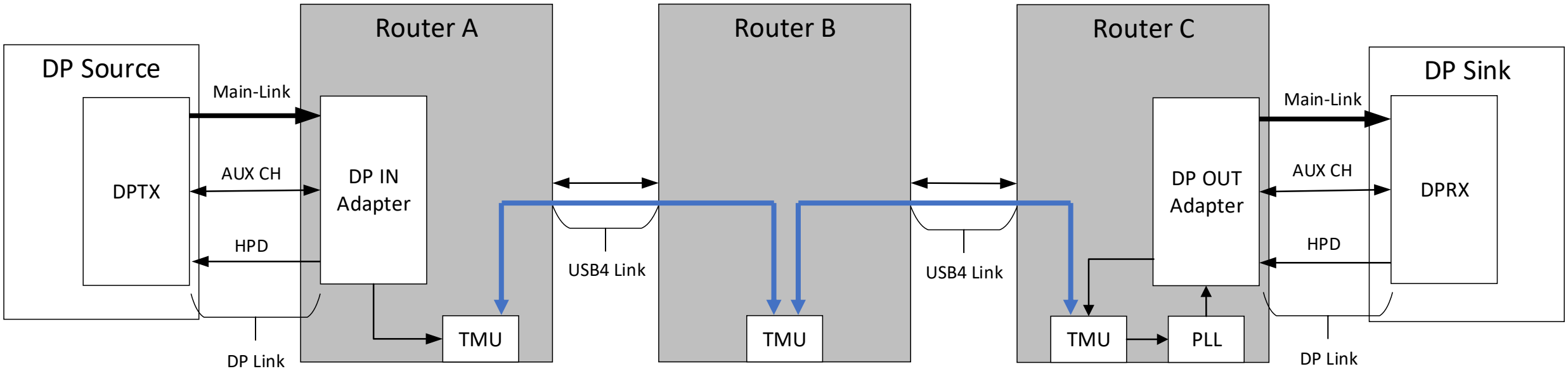
DisplayPort tunneling: Link Clock Restoration

- DisplayPort is tunneled from GPU to Monitor over USB4
- The video stream is restored at the Monitor side as it is sampled from GPU
- In order to recreate the video after tunneling, time sense is required



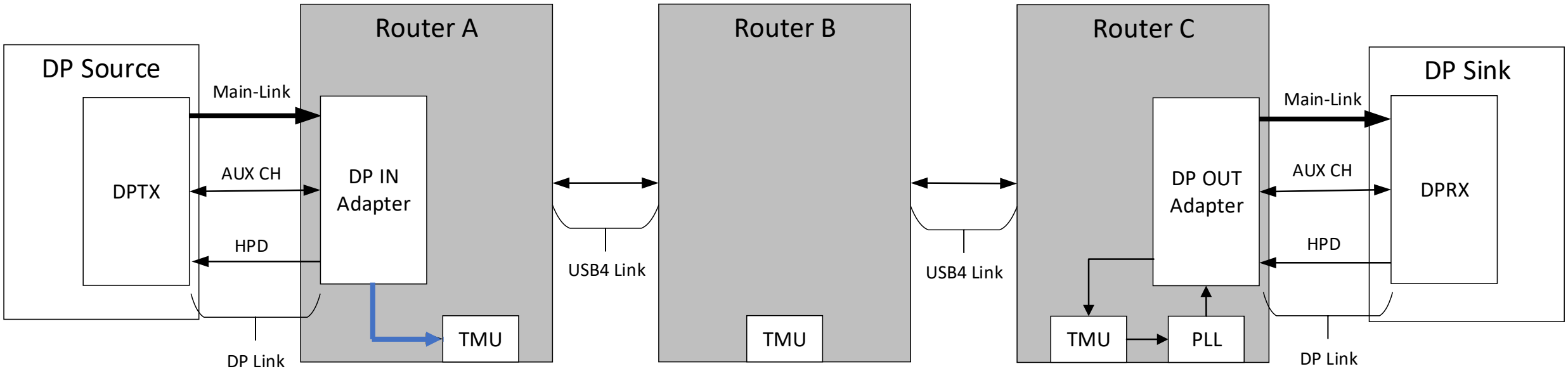
Link Clock Restoration Process

- TMU establish time synchronization over USB4 Link
- Grandmaster Time propagate through the entire USB4 tree



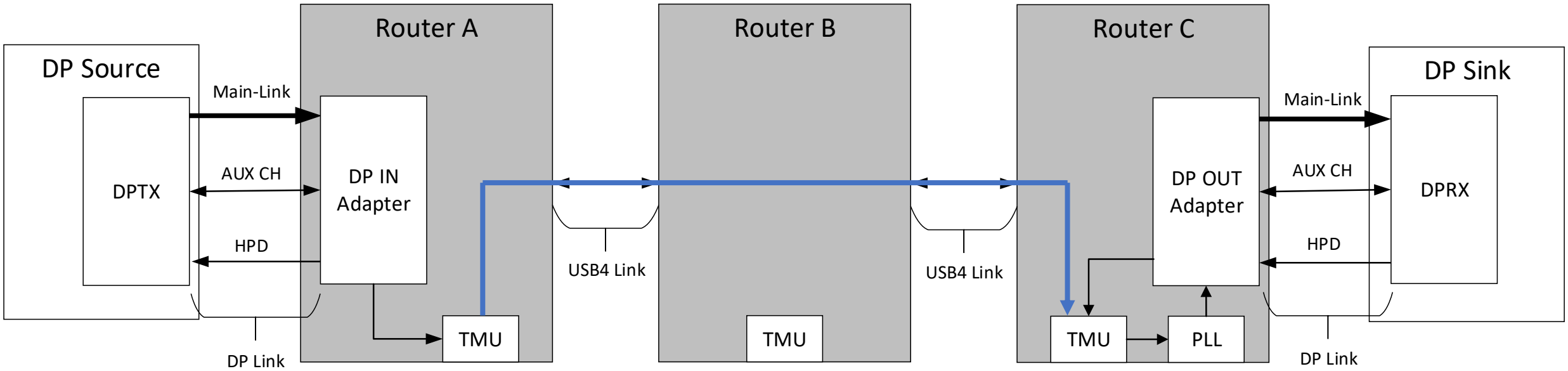
Link Clock Restoration Process

- DisplayPort Link clock is measured at the GPU side based on TMU events



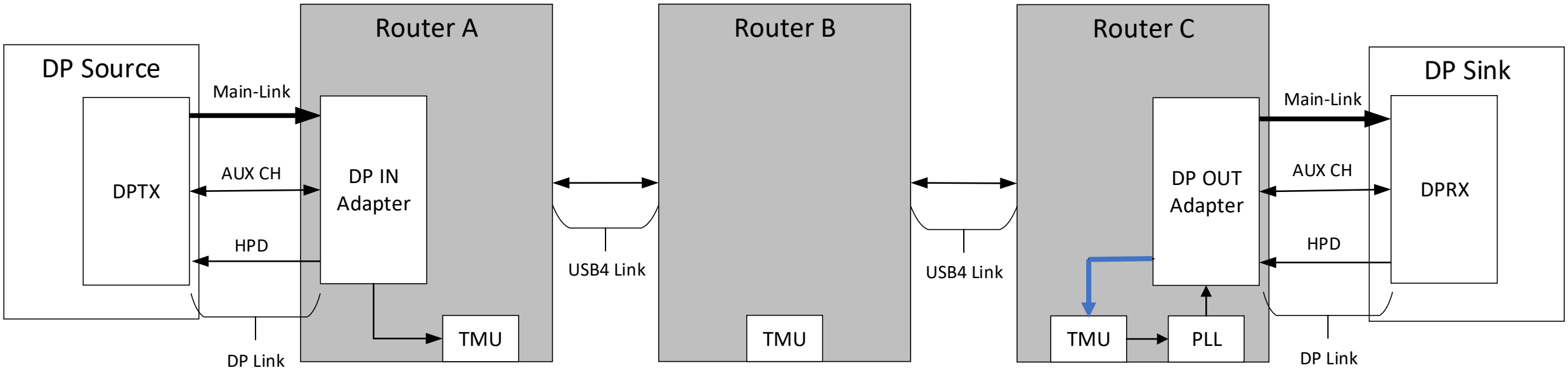
Link Clock Restoration Process

- USB4 packet with the measurement result is sent over USB4 network to the Router at the Monitor side



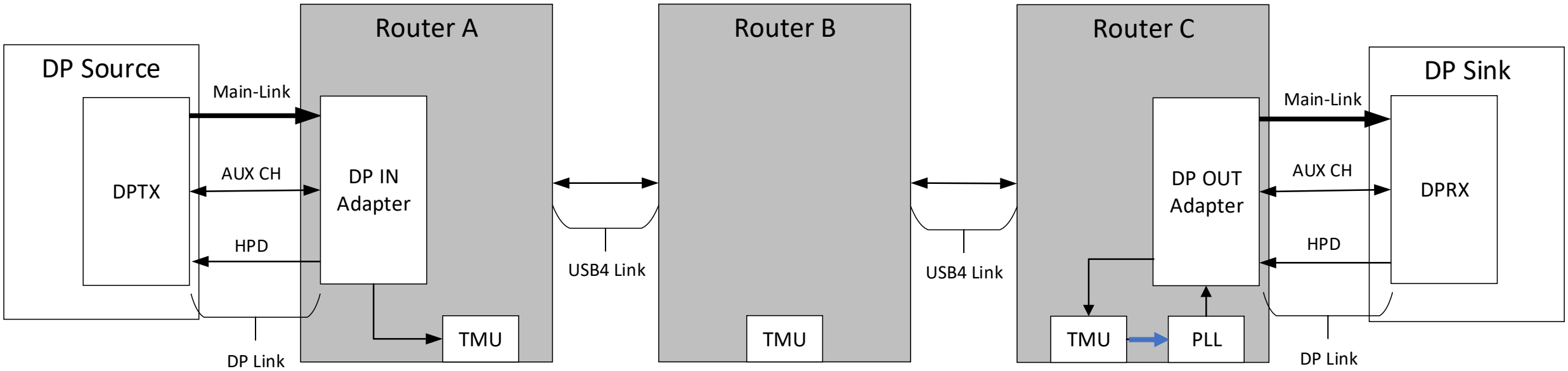
Link Clock Restoration Process

- DisplayPort Link clock is measured at the Monitor side based on TMU events
- The two measurements are compared
- Error correction is computed



Link Clock Restoration Process

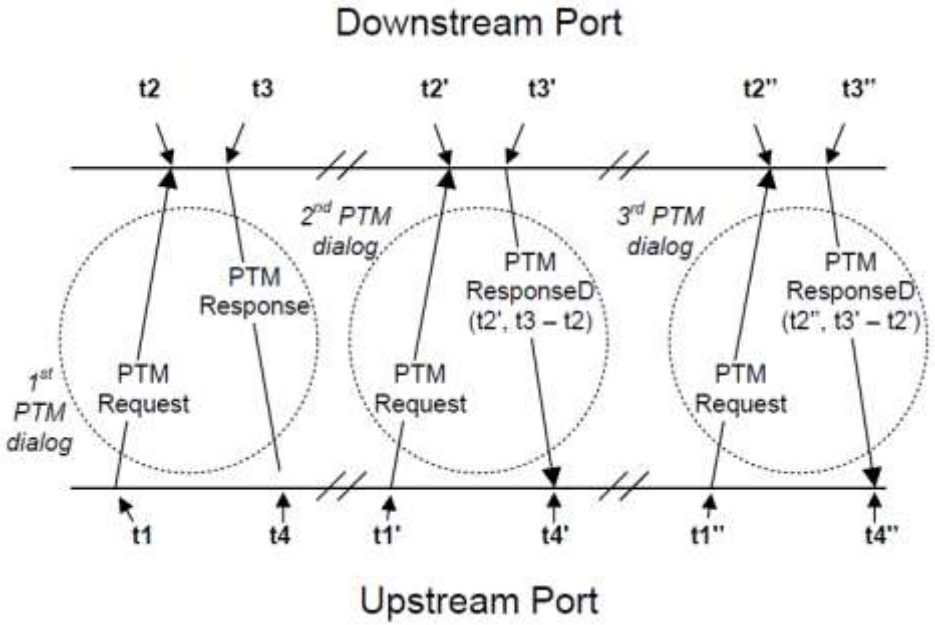
- DisplayPort Link clock frequency in the Monitor side is shifted according to the error correction result
- Entire mechanism is executed periodically



PCIe Precision Time Measurement

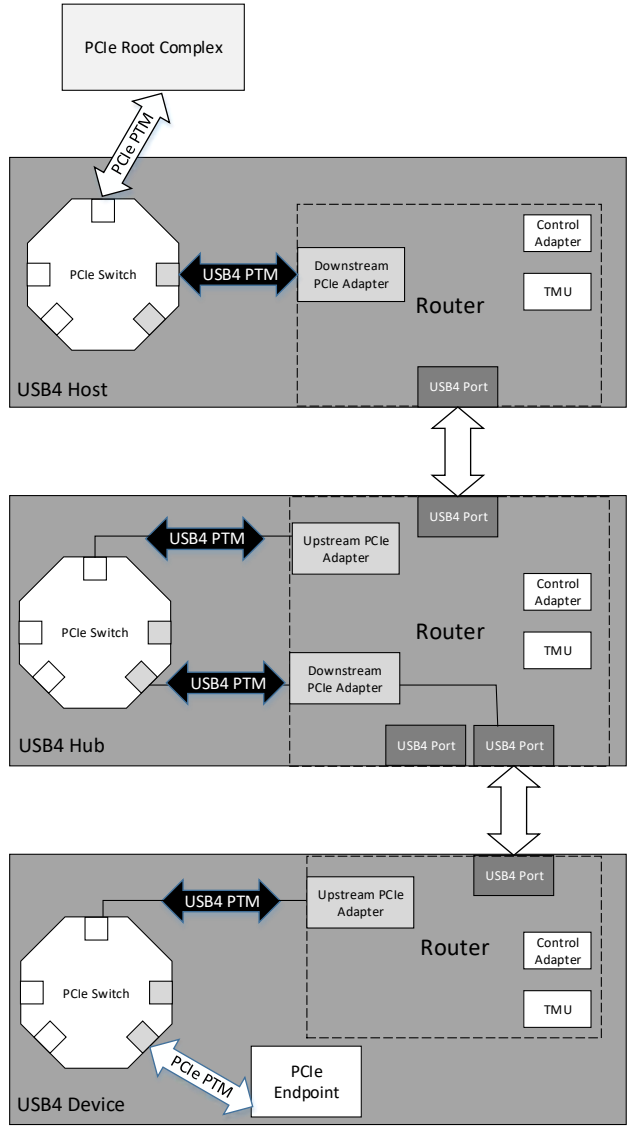
- Supporting components determine the relationship between their local time and a PTM Master Time
- PTM Handshake is used to determines the Link latency
- PTM ResponseD holds the *PTM Master Time* (t2') and the *Propagation Delay* (t3-t2)

- PTM Master Time at $t1' = t2' - \frac{((t4-t1)-(t3-t2))}{2}$



PCIe Tunneling: PTM Time Restoration

- Precision Time Measurement (PTM) message is tunneled over USB4
- PTM messages need USB4 time sense in order to recreate the PTM Master Time at the Endpoint side
- The Tunneled PTM holds two parameters to describe the relation between the PTM Master Time and the USB4 TMU Grandmaster Time
- The parameters **replace** the *PTM Master Time* and *Propagation Delay* fields in the PTM message
- PCIe Endpoint can restore the PTM Master Time from the TMU Grandmaster Time and generated parameters

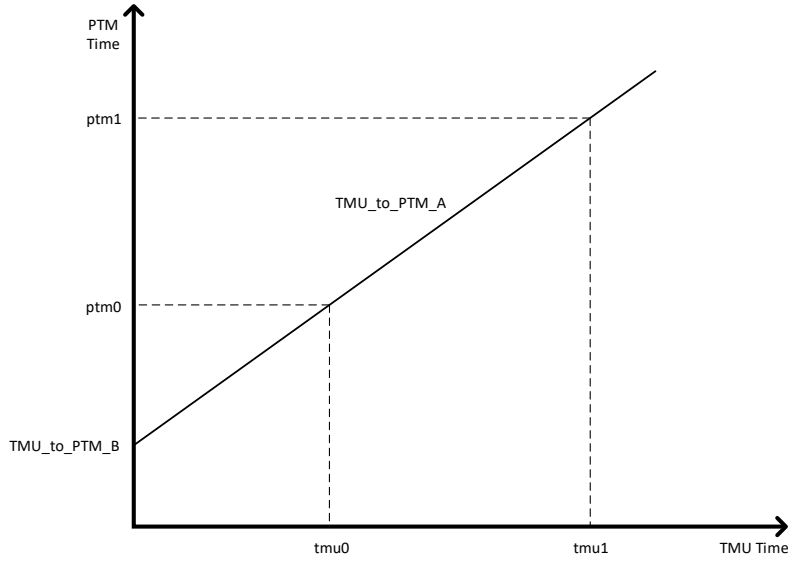


TMU to PTM Parameters

- The parameters represent the linear relationship between the PTM Master Time and TMU Grandmaster Time
- The parameters Generator is the Host Router connected to the PCIe Root Complex
- TMU_to_PTM_A represent the slope of the linear line
- TMU_to_PTM_B represent the intercept of the linear line
- Ideally:

ptm0 = TMU_to_PTM_A * tmu0 + TMU_to_PTM_B

ptm1 = TMU_to_PTM_A * tmu1 + TMU_to_PTM_B

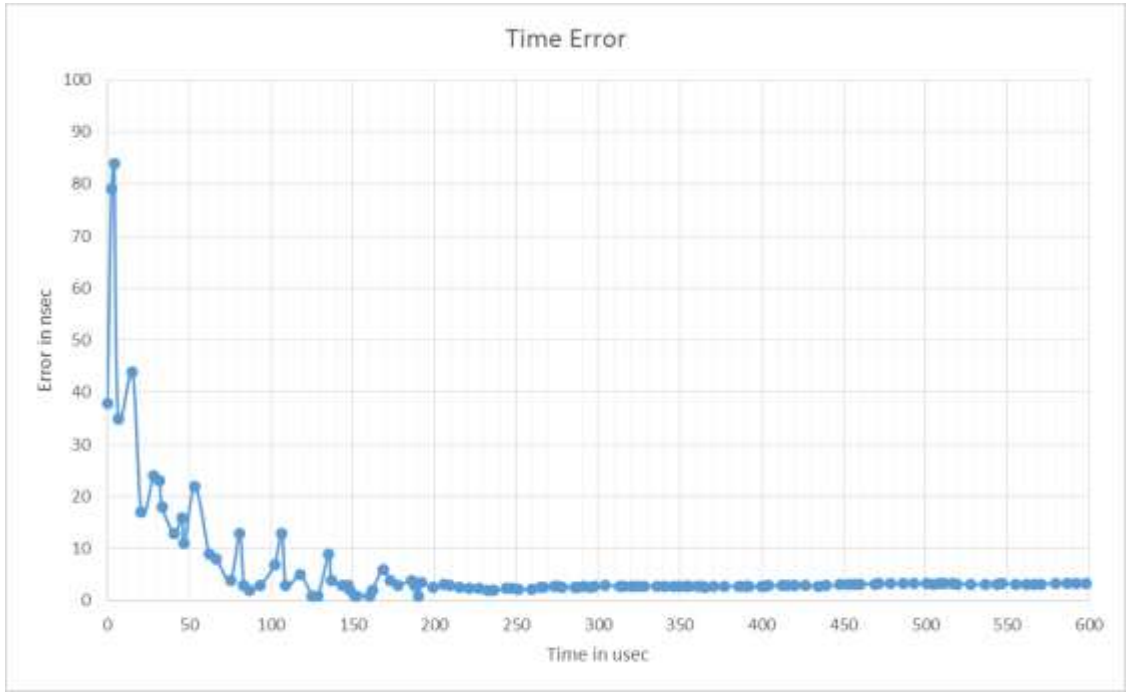


- Due to quantization error a more robust method described in the spec
- It is recommended to calculate the parameters periodically

PTM Master Time Reconstruction

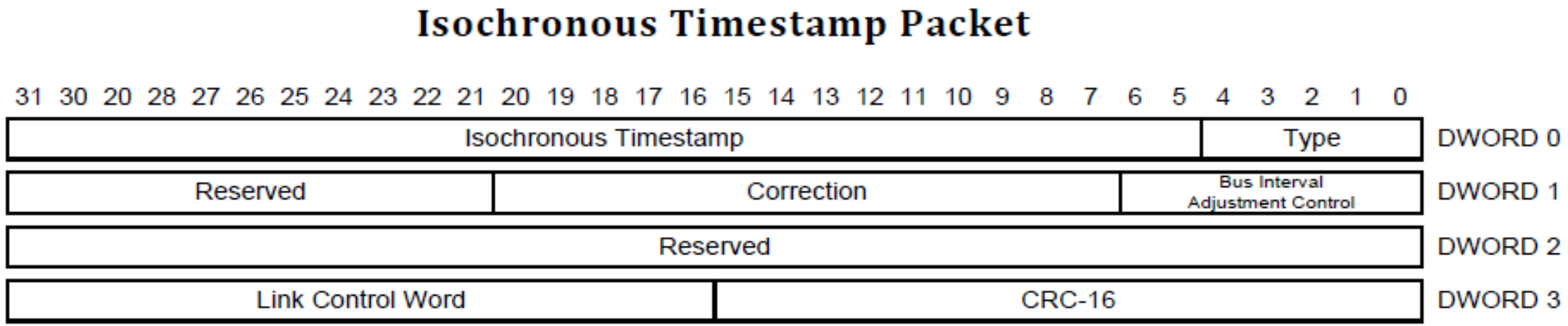
- PCIe Endpoint can use the TMU Grandmaster Time and generated parameters to reconstruct the PTM Master Time

PTM Master Time(t) = TMU_to_PTM_A * TMU Grandmaster Time(t) + TMU_to_PTM_B



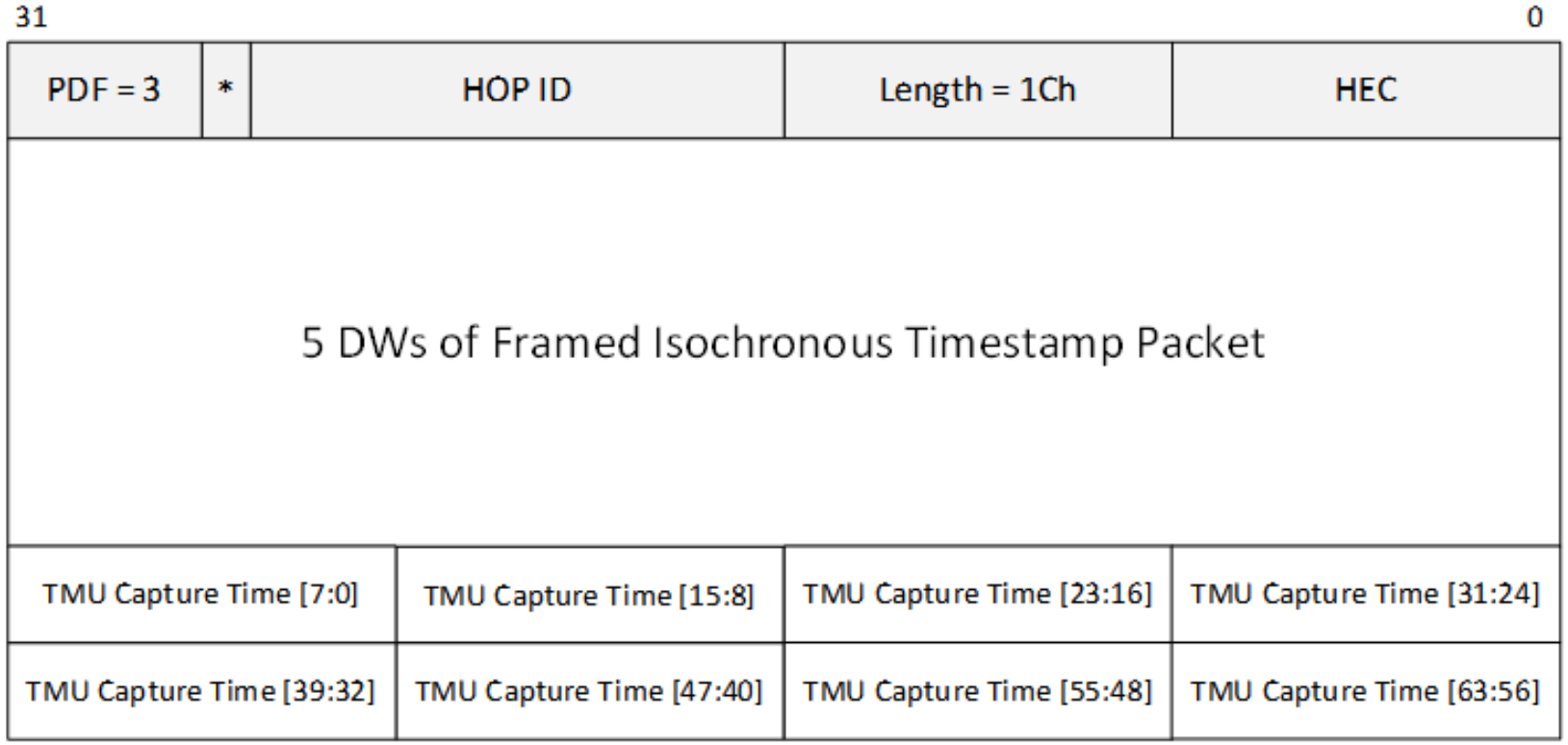
USB Tunneling: ITP Timestamp Correction

- USB3 uses Isochronous Timestamp Packets to deliver timestamps from Host to all active Devices
- The *Isochronous Timestamp* field holds the *Bus Interval counter* and *Delta*



ITP Timestamp Correction

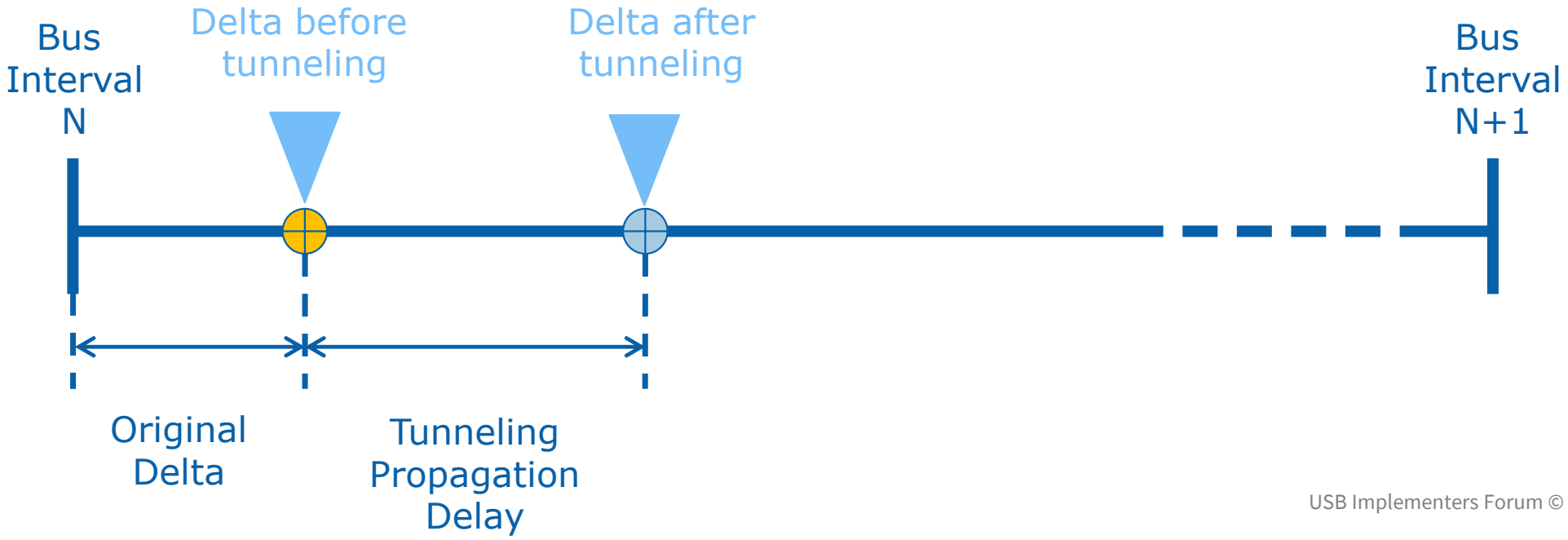
- Timestamp is taken before the ITP enters the tunneling and is added to the tunneled packet



* SuppID

ITP Timestamp Correction

- Another timestamp is taken when ITP exits the tunnel
- Both timestamps are in TMU Grandmaster Time
- An Upstream USB3 Adapter that receives an ITP tunneled packet updates the *Delta* field according to the calculated propagation delay (in tIsochTimeStampGranularity resolution)
- It also update the *CRC-16* field



Time for Q&A

Agenda

- Time Synchronization
- Host Interface
- Connection Manager
- Thunderbolt™ 3 Compatibility

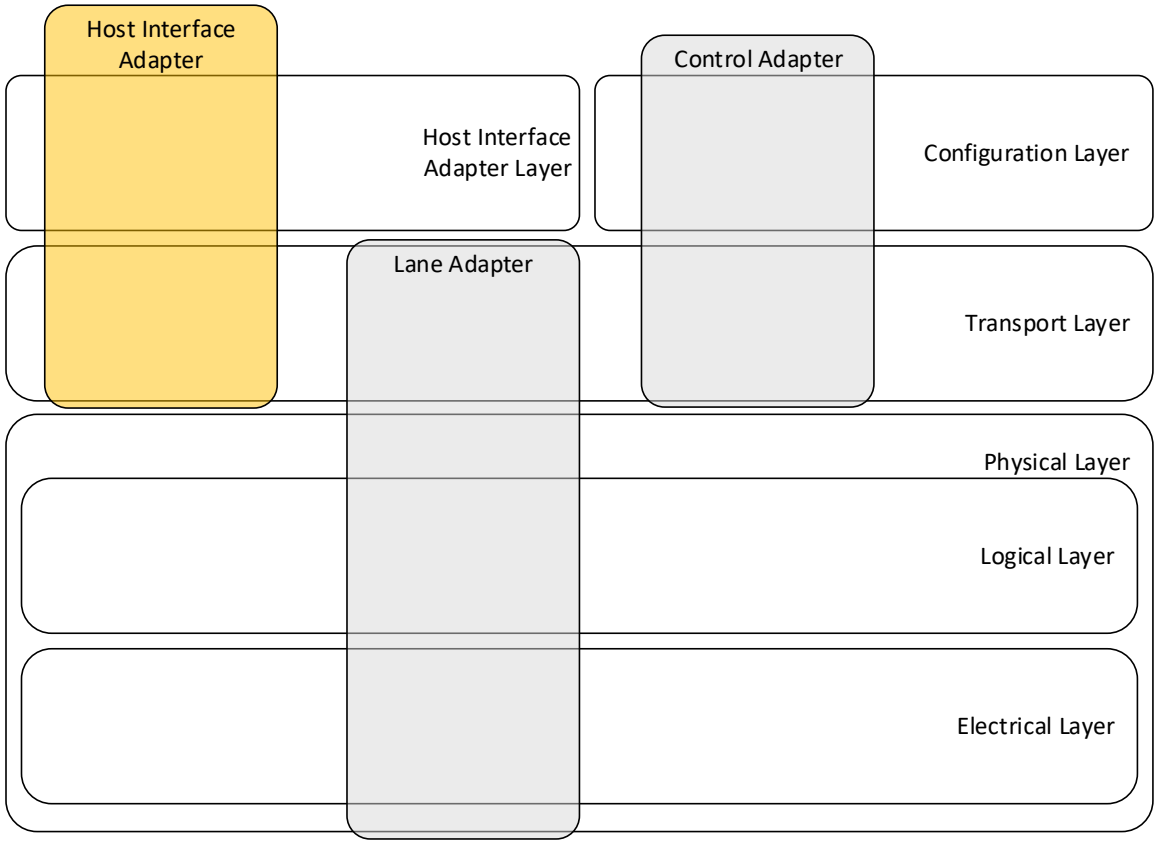
Host Interface Agenda

- Roles
- Descriptor Rings
- Modes of Operation
- E2E Flow Control
- Host-to-Host Tunneling

Host Interface Adapter Roles

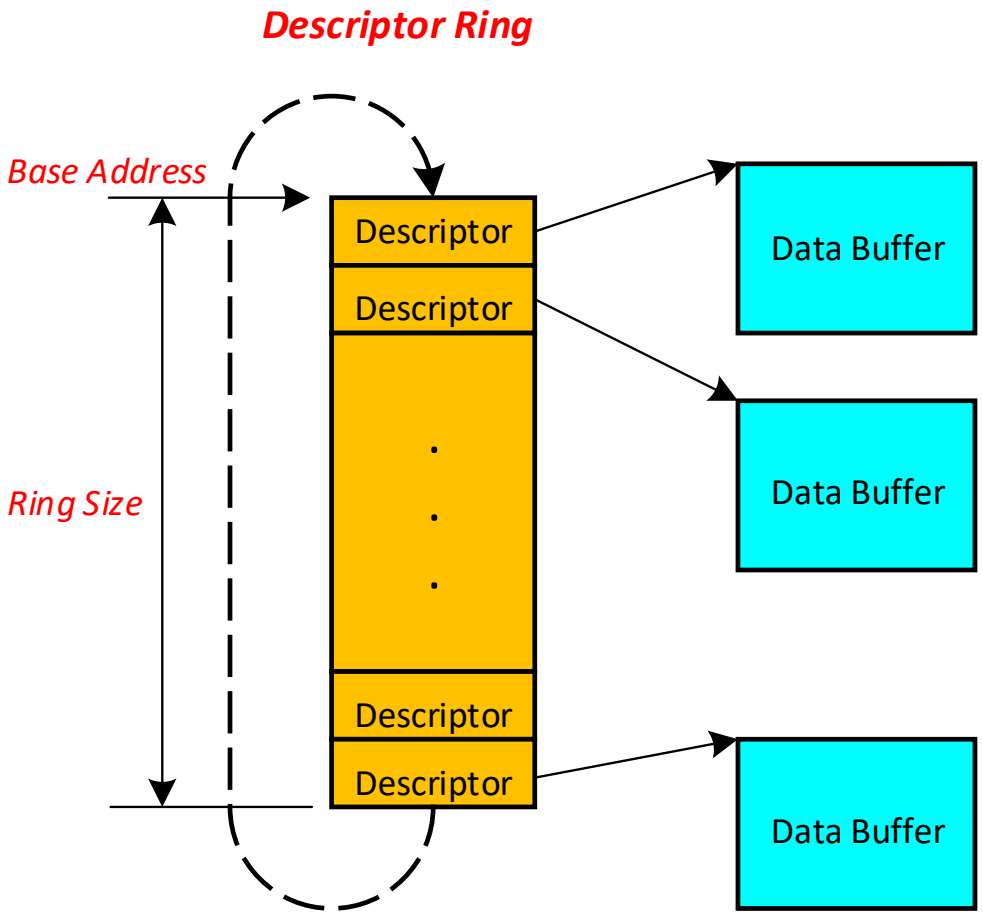
- Interface for Connection Manager to access Transport Layer of Host Router

- Provides parallel communication channels
  - Control Packet routing
  - Host-to-Host Tunneling



Host Interface – Descriptor Rings

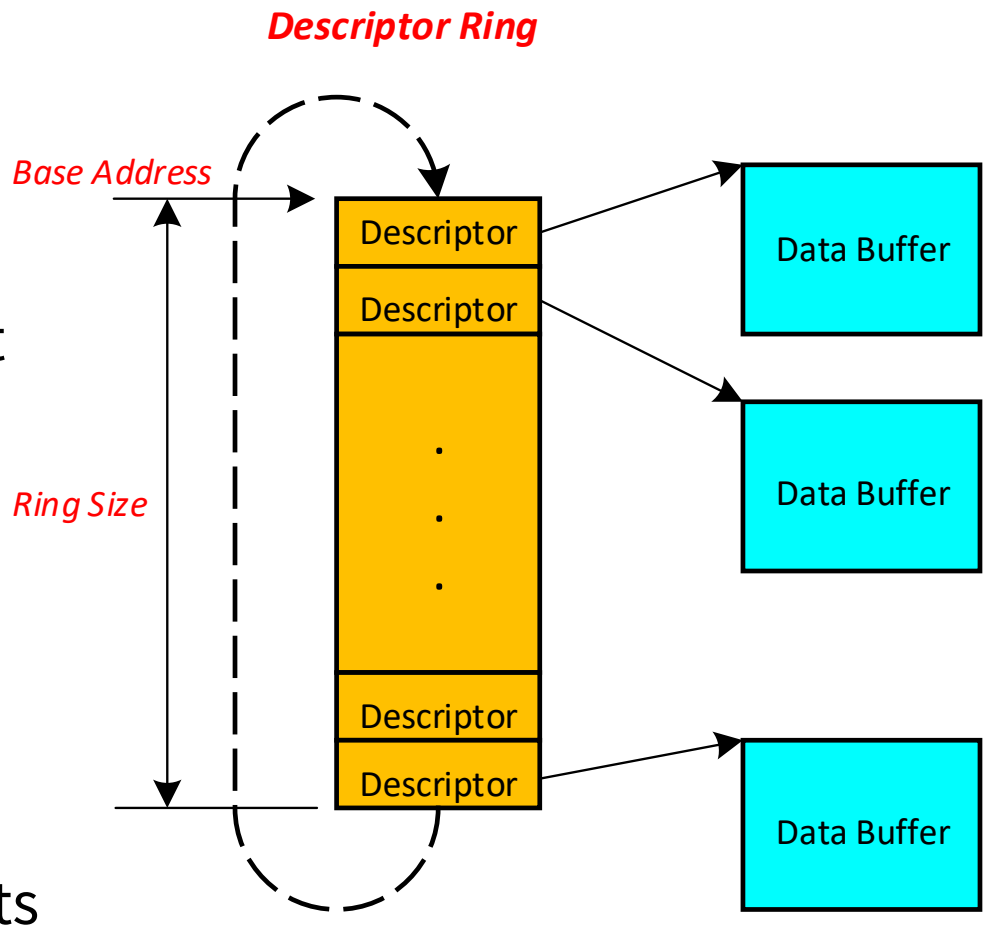
- A Connection Manager and a **Host Interface Adapter** interact via *Descriptor Rings*
- The *Descriptor Rings* reside in Host Memory
- A *Descriptor Ring* is a circular set of *Descriptors*
- Each *Descriptor* contains control fields and a reference to a *Data Buffer* in Host Memory



Host Interface – Descriptor Rings

Transmit Rings

- The Connection Manager
  - Performs Initialization - Sets the Ring Parameters
  - Adds Data Buffer(s) and update the relevant *Descriptor(s)*
  - Notifies the **Host Interface Adapter** when new *Descriptors* are added
- The **Host Interface Adapter**
  - Fetches new *Descriptor(s)*
  - Fetches the Data Buffer(s)
  - Encapsulates the data into Tunneled Packets

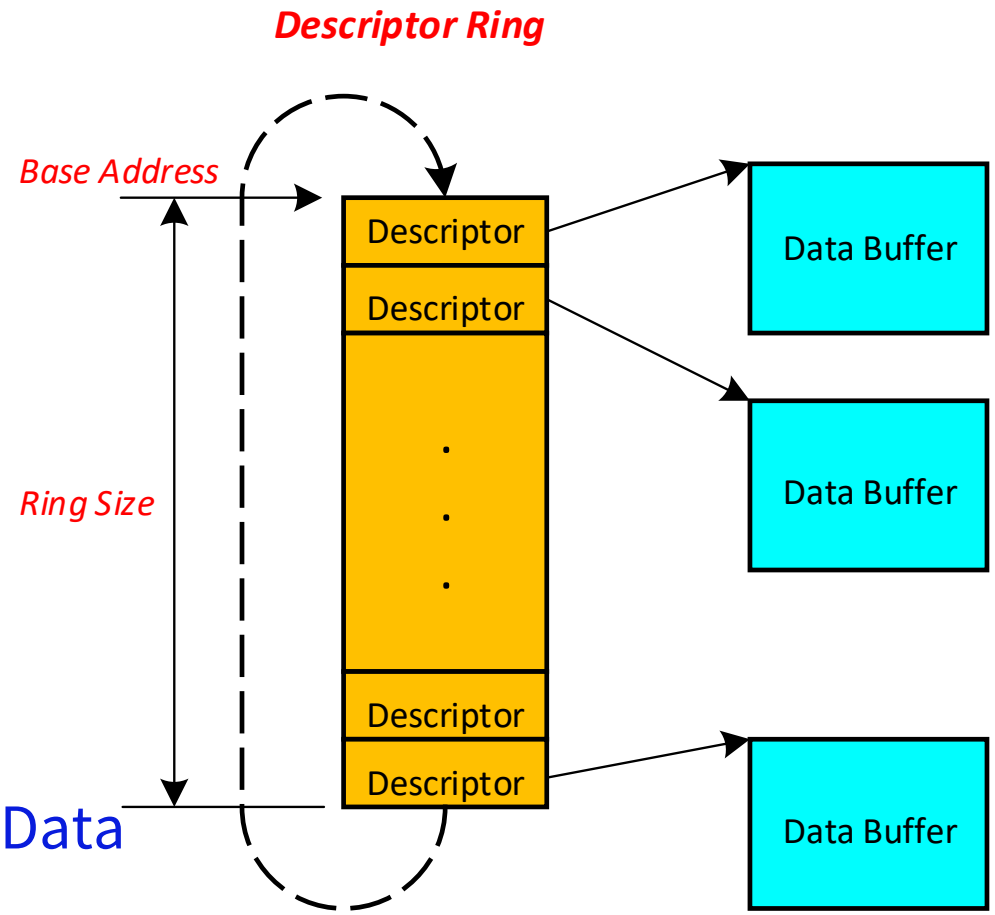


Host Interface – Descriptor Rings

Receive Rings

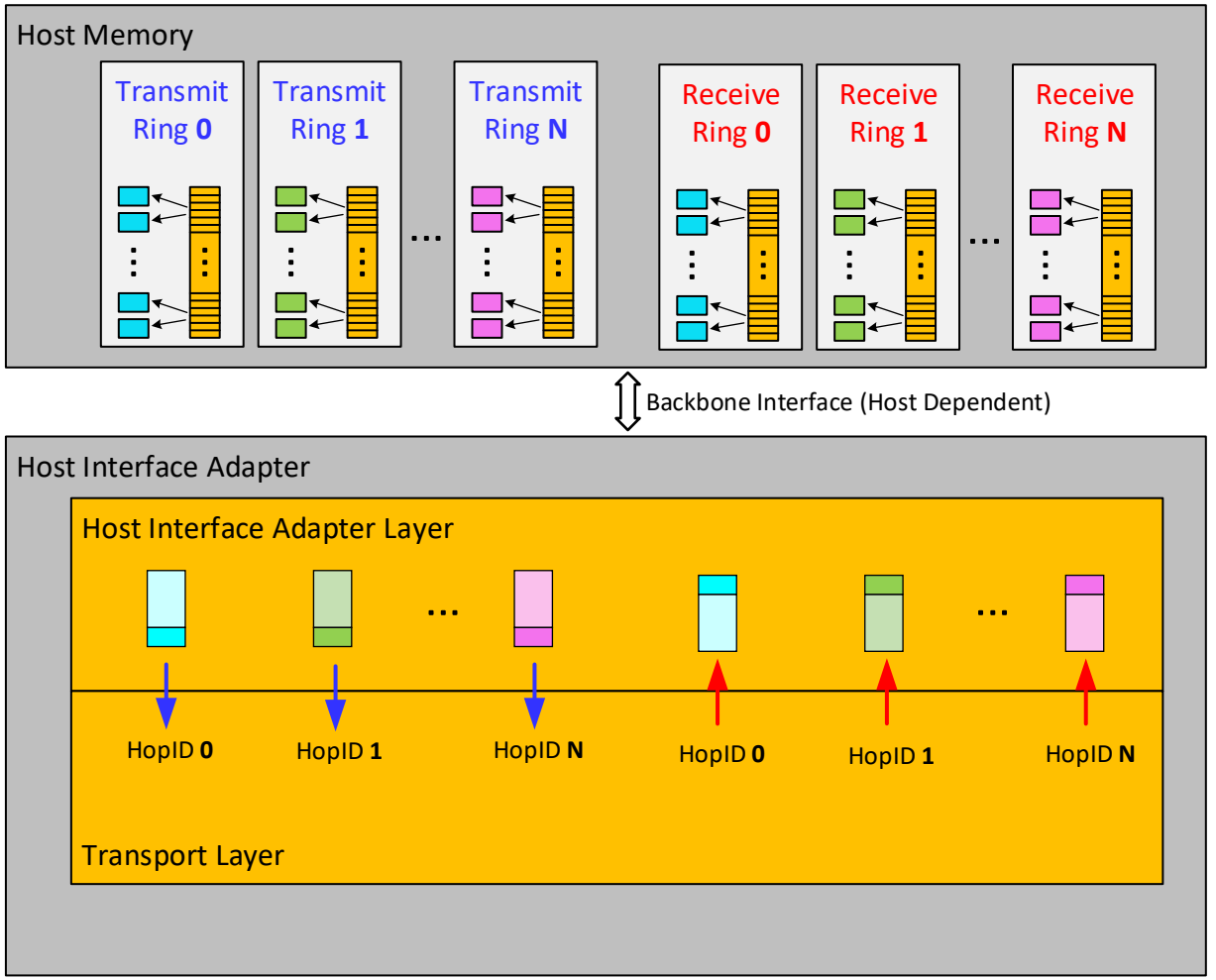
- The Connection Manager
  - Performs Initialization - Sets the Ring Parameters, **empty** Data Buffer(s), and *Descriptor(s)*

- The **Host Interface Adapter**
  - Fetches new *Descriptor(s)*
  - Upon Tunneled Packet reception, Decapsulates the data and stores it at the Data Buffer(s)
  - Updates the Connection Manager when new data is received



Host Interface – Rings & Paths

- **Host Interface Adapter** implements at least 2 Transmit Rings and at least 2 Receive Rings
  - Ring 0 for Control Packets
  - Ring 1 for Host-to-Host Tunneling
- **Host Interface Adapter** implements the same amount of Receive and Transmit Rings, and the same number of Paths
- **HopID** corresponds to Descriptor Ring
  - In Source Adapter, packet with HopID=X carries data from Transmit Ring X
  - In Destination Adapter, packet with HopID=Y carries data to be posted in Receive Ring Y
  - In between Source and Destination Adapters, HopID can be any supported value

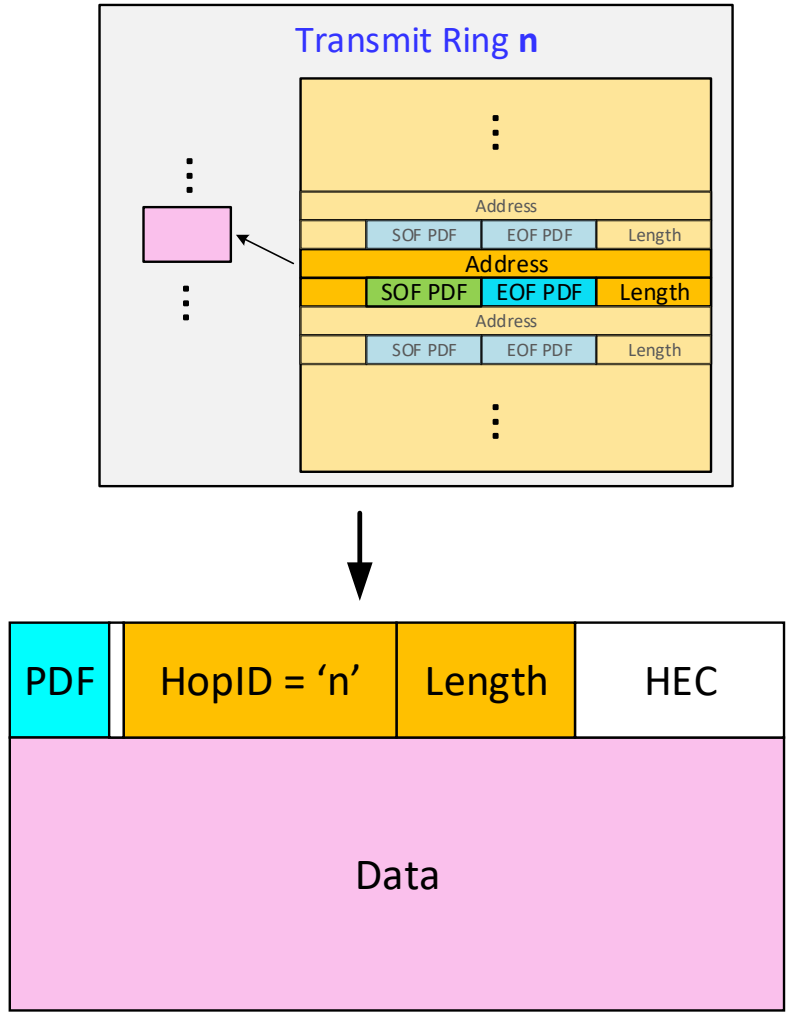


Host Interface – Rings Mode of Operation

- A Ring can be set to operate in one of two modes
  - **Raw**
    - Used mainly for **Control Packets** (Ring 0)
    - Can also be used by Host-to-Host Tunneling
    - Each Descriptor points to the Payload of a single Tunneled Packet
  - **Frame**
    - Used only by **Host-to-Host Tunneling**
    - A data Frame, of up to 4K bytes, is described by one or more Descriptors
    - The data Frame is segmented and encapsulated into one or more Tunneled Packets
    - CRC32 is added at the end of the data Frame

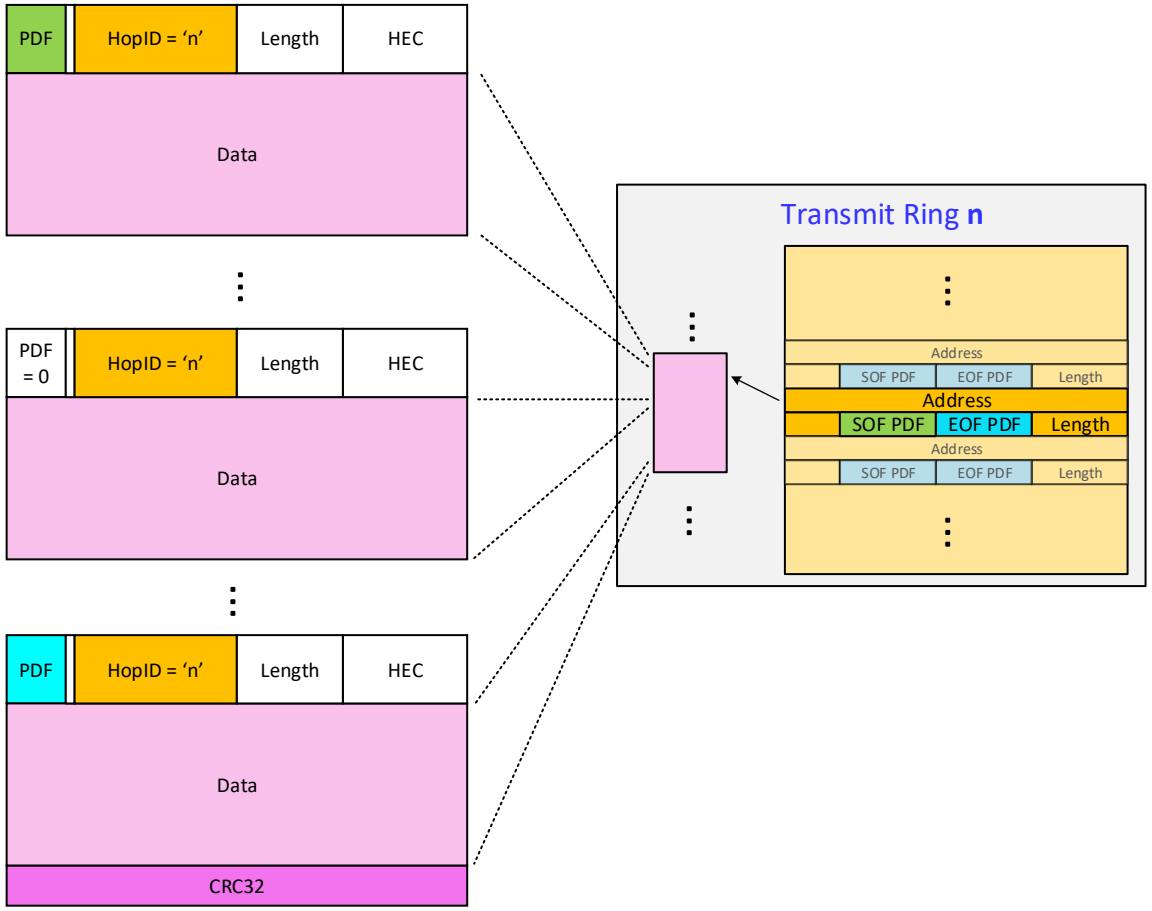
Host Interface – Raw Mode

- Transmit Interface
  - Host Interface Adapter prepends Transport Layer Header to Data Buffer, to form Tunneled Packet
  - **PDF** field is taken from the **EOF PDF** field of the matching Descriptor
  - HopID matches Transport Descriptor Ring
- Receive Interface
  - HopID determines which Receive Descriptor Ring payload is written to
  - Host Interface Adapter removes Transport Layer Packet Header, posts payload to Data Buffer and update Descriptor fields



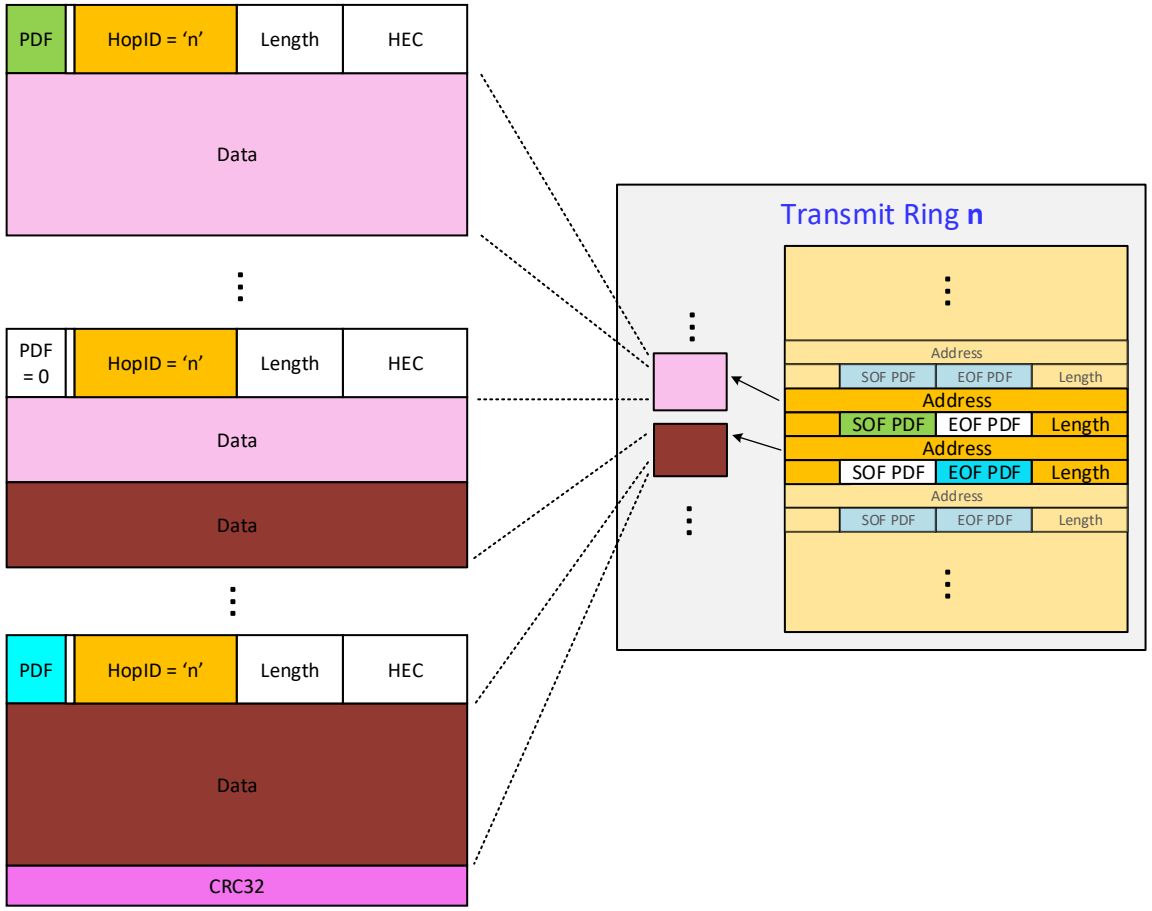
Host Interface – Frame Mode

- Transmit Interface
  - Segments Frame and add CRC into one or more Tunneled Packets
  - **PDF** field identifies if packet contains the start, middle, or end of a Frame
    - Start = **SOF PDF**
    - Middle = 0
    - End (or only) = **EOF PDF**
  - HopID matches Transport Ring
- Receive Interface
  - HopID determines which Receive Ring Frame is written to
  - Host Interface Adapter reassembles Frame, checks CRC, and posts to Receive Ring



Host Interface – Frame Mode

- Data Frame can span across multiple Data Buffers



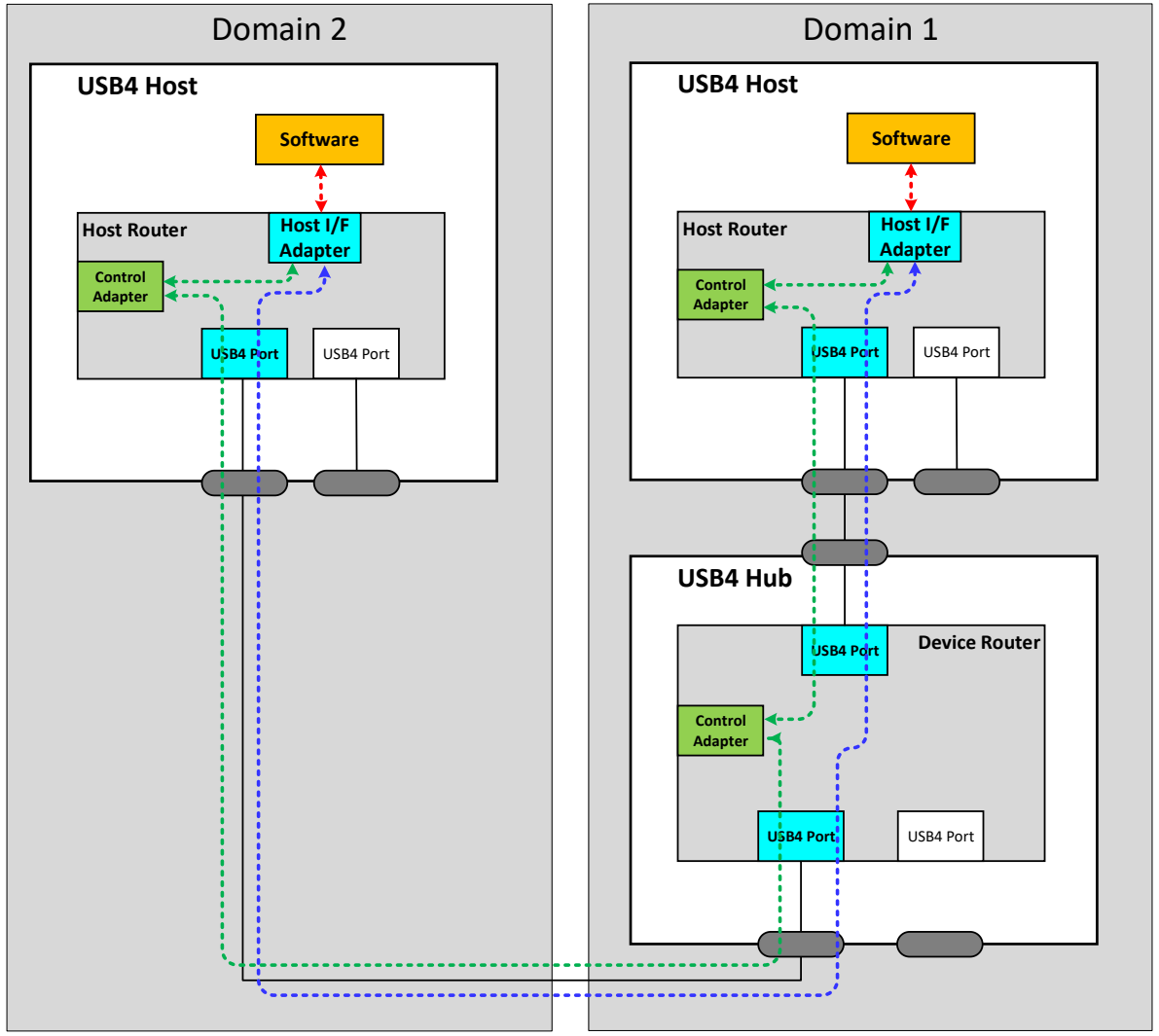
Host Interface – End to End Flow Control

- The E2E Flow Control is a mechanism which prevents overflow of a Receive Descriptor Ring
- The mechanism is used only for Host-to-Host Tunneling
- The Connection Managers on both Domains enable this feature at the Descriptor Ring control
- The E2E protocol
  - *Receive Ring* sends *E2E Credit Grant* with information about the available Descriptors
  - *Transmit Ring* only sends packet when there is room for it. It also sends *E2E Credit Sync* packet with information about the number of sent packets

Host-to-Host Tunneling

- Allows two USB4 hosts to communicate with each other

- The Host-to-Host communication consists of
  - Single **Control Channel** using Control Packets
  - One or more Data Channels using *Descriptor Rings* operating in *Raw* or *Frame* mode



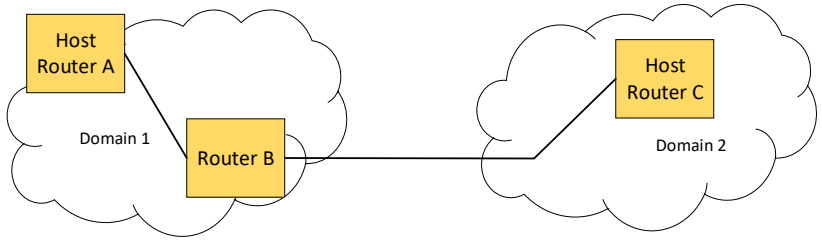
Host-to-Host Tunneling – Control Channel

- The **Control Channel** is used for
  - Discovery of capabilities
  - Inter-Domain Link Management
  - Notifications
  - Setup and Teardown the Data Channel

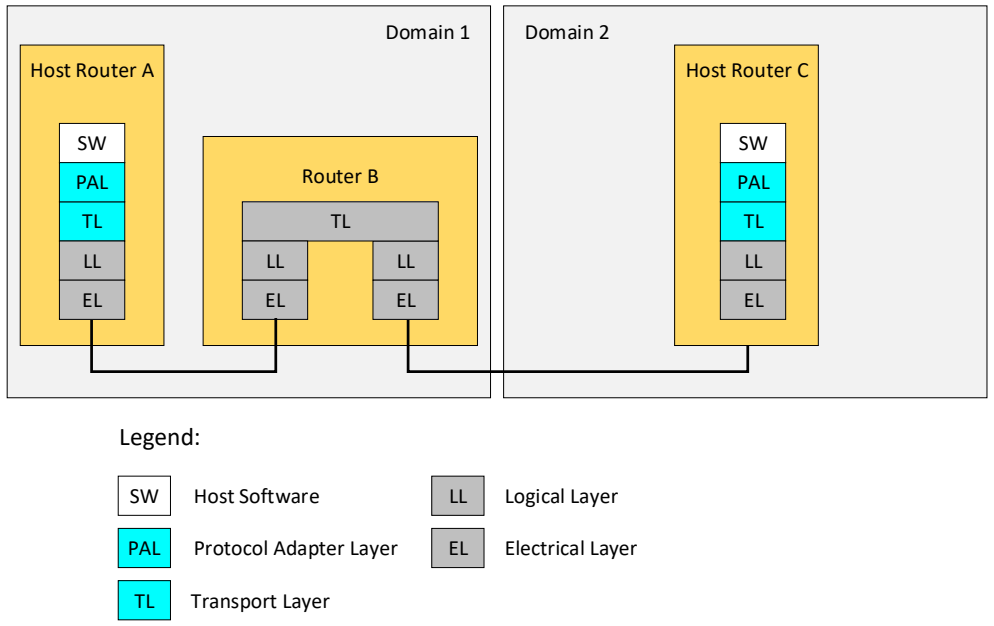
- The **Control Channel** uses ***Inter-Domain Request*** and ***Inter-Domain Response*** Control Packets
  - The protocol of the **Control Channel** is defined in a “USB4 Inter-Domain Service Protocol” software specification

Host-to-Host Tunneling – Data Channel

- The Data Channel is a high throughput channel that can be used for various types of applications



- The Tunneled Packets which travel the Data Channel originate and consumed by the **Host Interface Adapters** in the USB4 Hosts
  - Device Routers along the Path route H2H Tunneled Packets without consuming them



Time for Q&A

Agenda

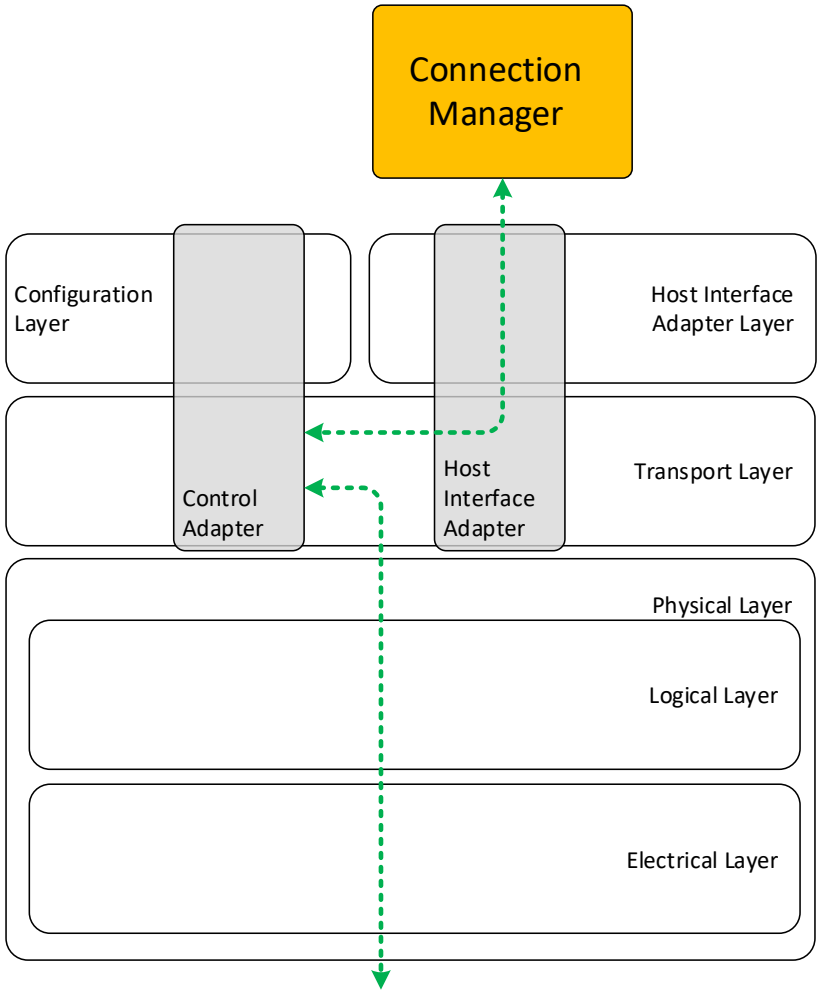
- Time Synchronization
- Host Interface
- Connection Manager
- Thunderbolt™ 3 Compatibility

Connection Manager Agenda

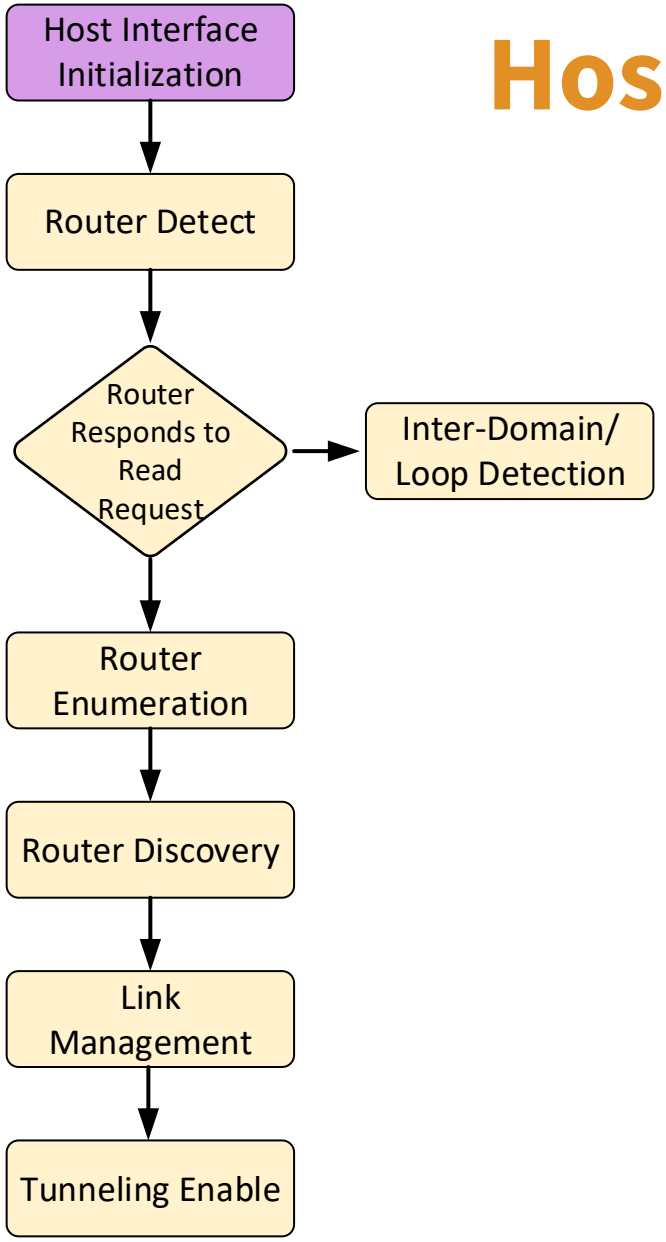
- Roles
- Detect to Tunneling
- Paths
- QoS

Connection Manager Roles

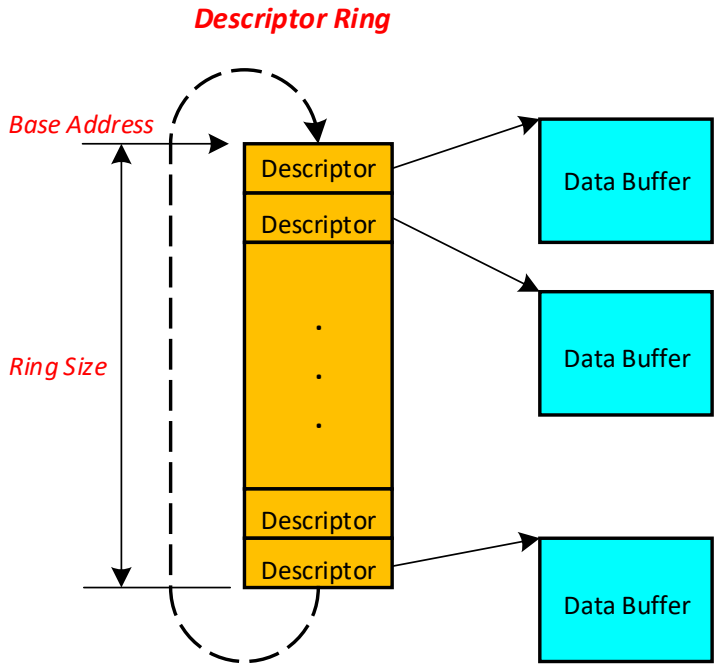
- Manages its Domain through
  - Discovery & Configuration
  - Setting USB4 Link Parameters
  - Configure Paths & QoS

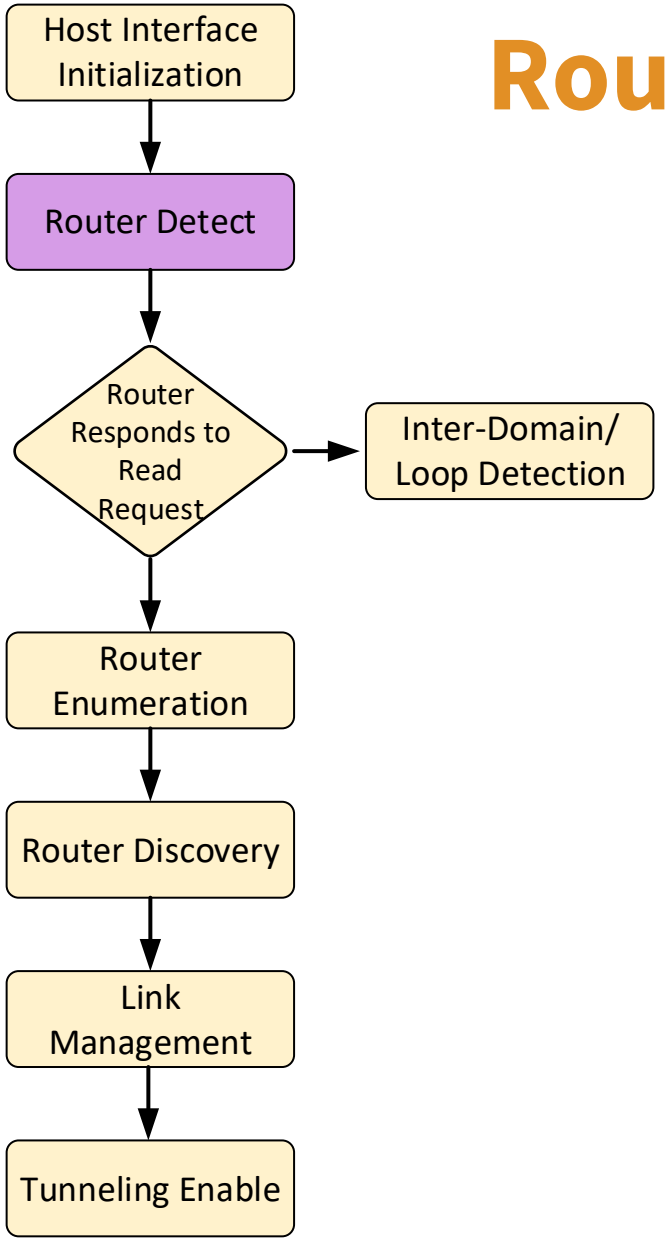


Host Interface Initialization



- A Connection Manager Initialize *Receive Ring **0*** and *Transmit Ring **0*** to allow Control Channel communication
  - Ring Base address
  - Ring Size
  - Ring Interrupts
  - Ring Enable

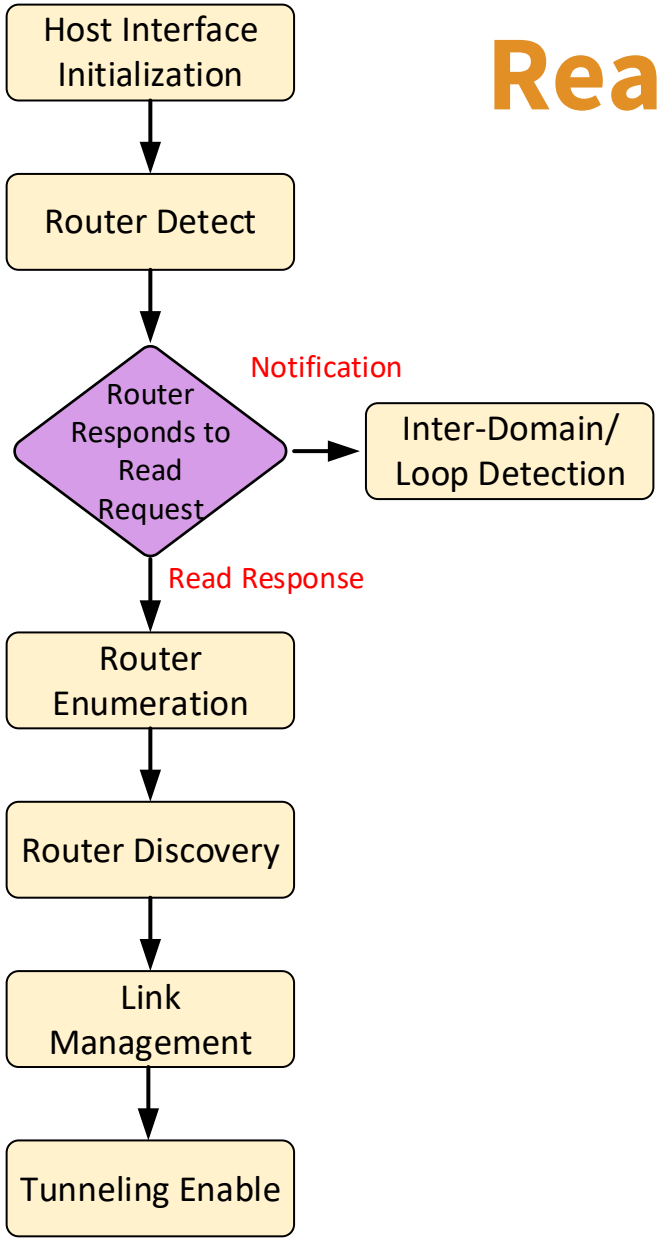




Router Detection

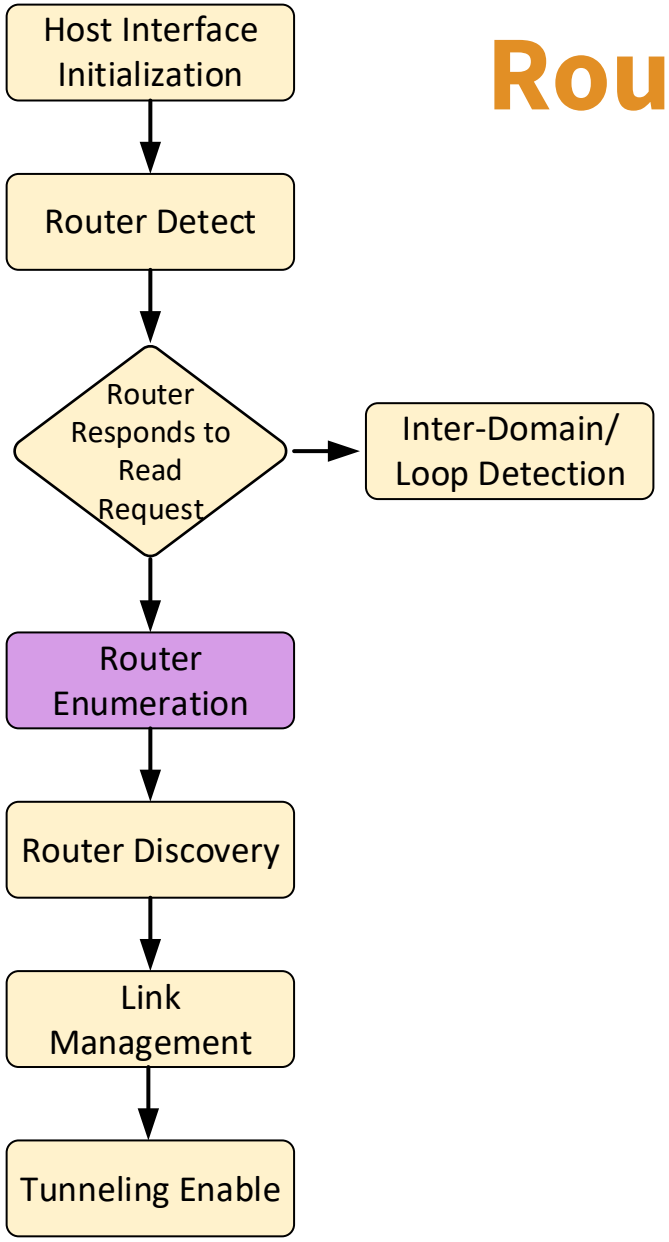
- Detecting a new connected Router
  - Hot Plug Event Packet
  - Polling USB4 Ports status
- Causes for Hot Plug event
  - A Router was hot plugged
  - Domain powers up
  - Exit from sleep
  - Exit from Downstream Port Reset

Read Router CS



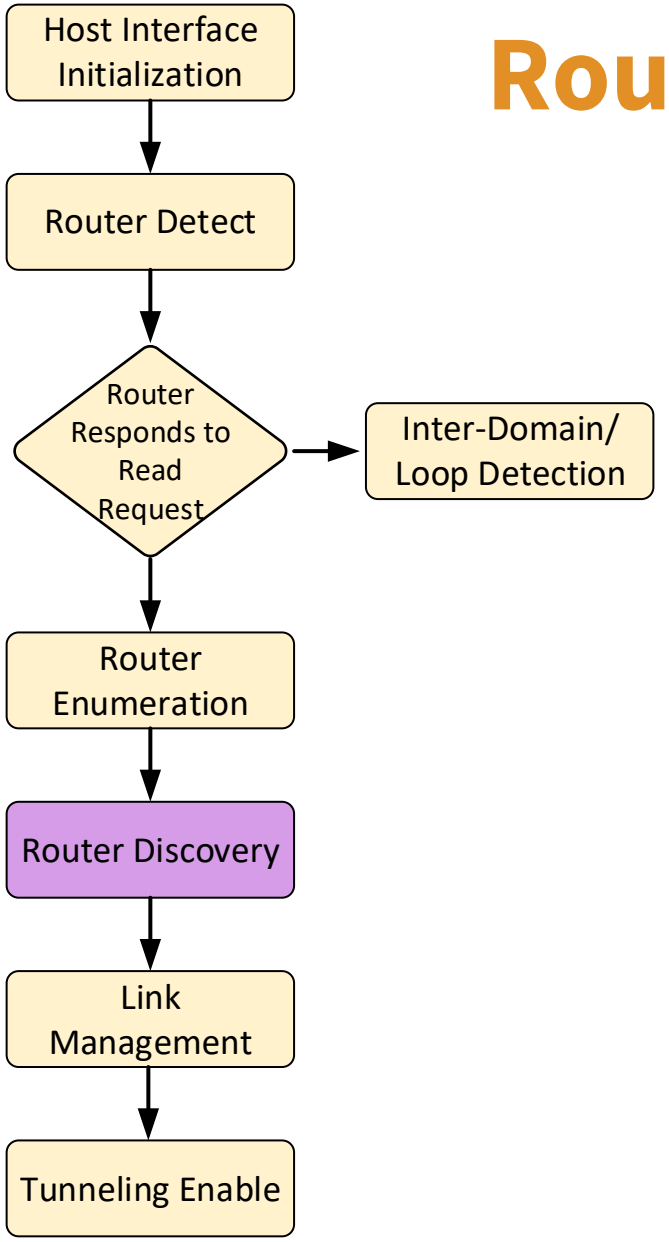
- The Connection Manager **unlocks** the DFP of the upstream Router

- The Connection Manager Attempts to read the newly connected Router's ***Router Configuration Space***
  - If the Router was connected using it's UFP it will respond with a **Read Response Packet**
  - Else it will respond with **Notification Packet** with an Event Code = [ERR_ENUM / ERR_NUA]



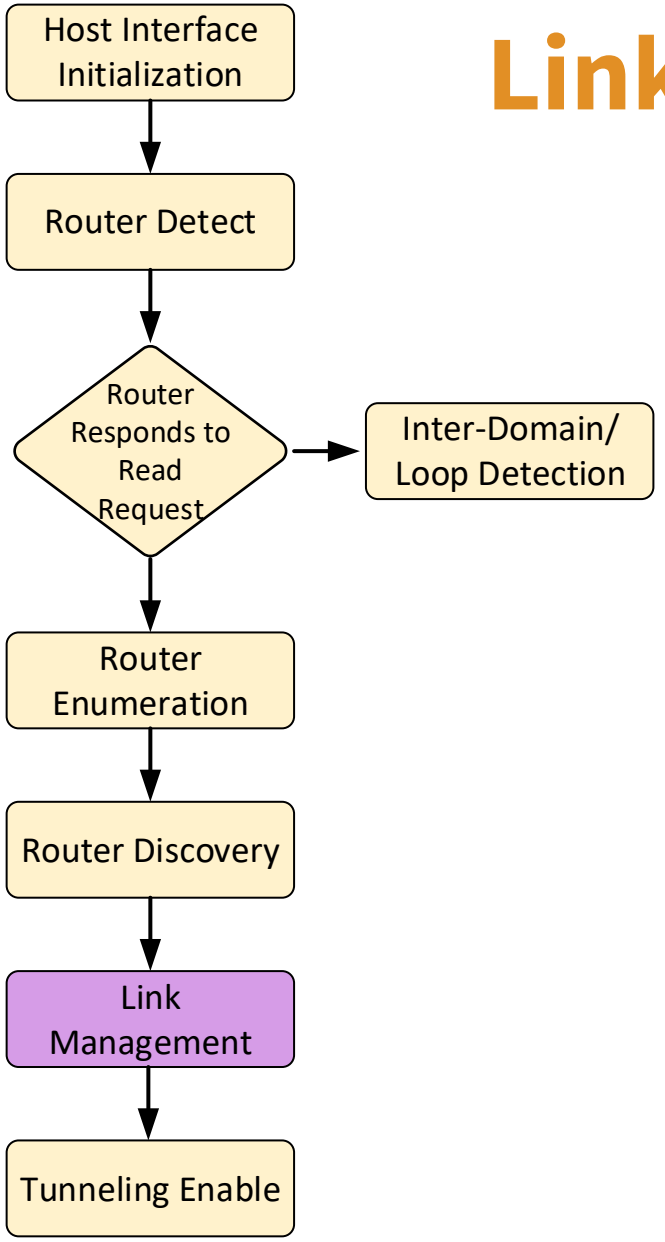
Router Enumeration

- Until **enumerated**, a Router responds only to
  - *Read Request* to its *Router CS*
  - Write Request to DW1-DW4 of its *Router CS*
- **Enumerating** a Router by the Connection Manager means writing *DW1-DW4* of the Routers *Router CS*
  - *TopologyID* + *TopologyID Valid*
  - *Depth*
  - *CM USB4 Version*
- After **enumeration**, the Router is part of the Connection Manager's Domain



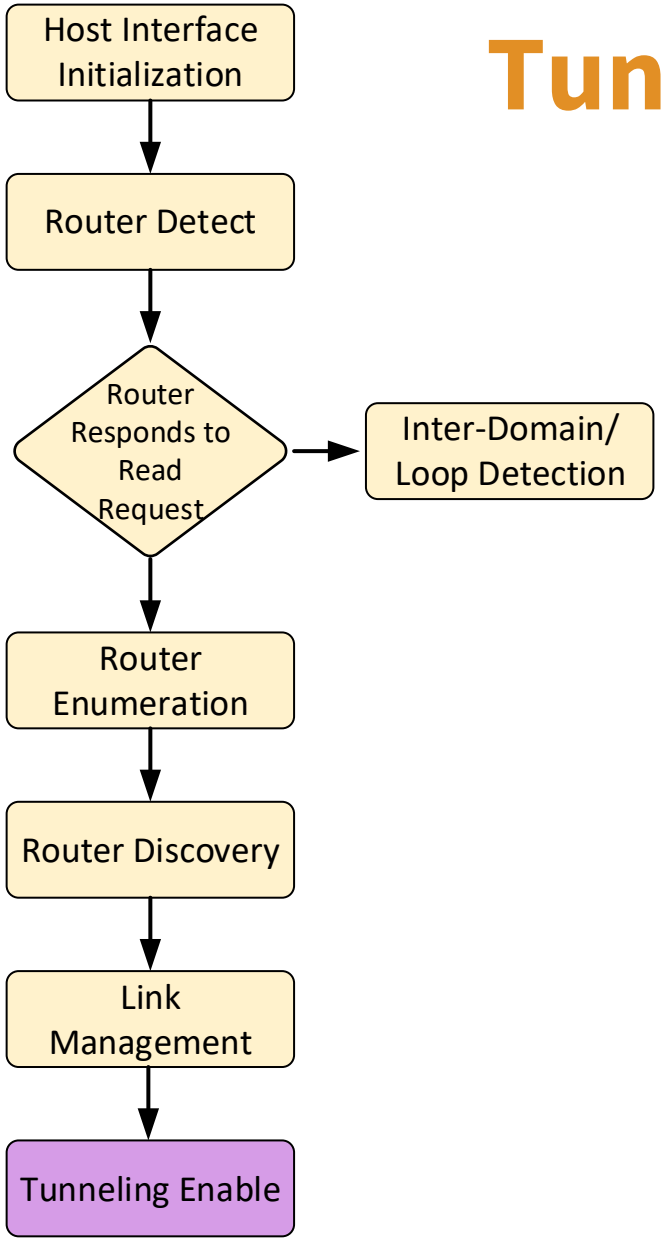
Router Discovery

- A Connection Manager polls the **Router Ready** bit at the *Router CS* . The Bit signify the Router is ready for Discovery
- A Connection Manager Discovers additional information about the Router and the Product
  - Maps all available Adapters, types and counts
  - **DROM Read** Router Operation
    - Product Information
    - Additional Router and Adapters settings
    - Described in “USB4 DROM Specification”



Link Management

- A Connection Manager sets the **Link Mode**
  - *Dual-Lane Link* Vs *Single-Lane Link*
  - *Gen3* Vs *Gen2*
- A Connection manager may enable **CLx**
- A Connection Manager sets the **TMU** operation
  - Enable Vs Disable
  - *LowRes* Vs *Uni-HiFi* Vs Bi-HiFi



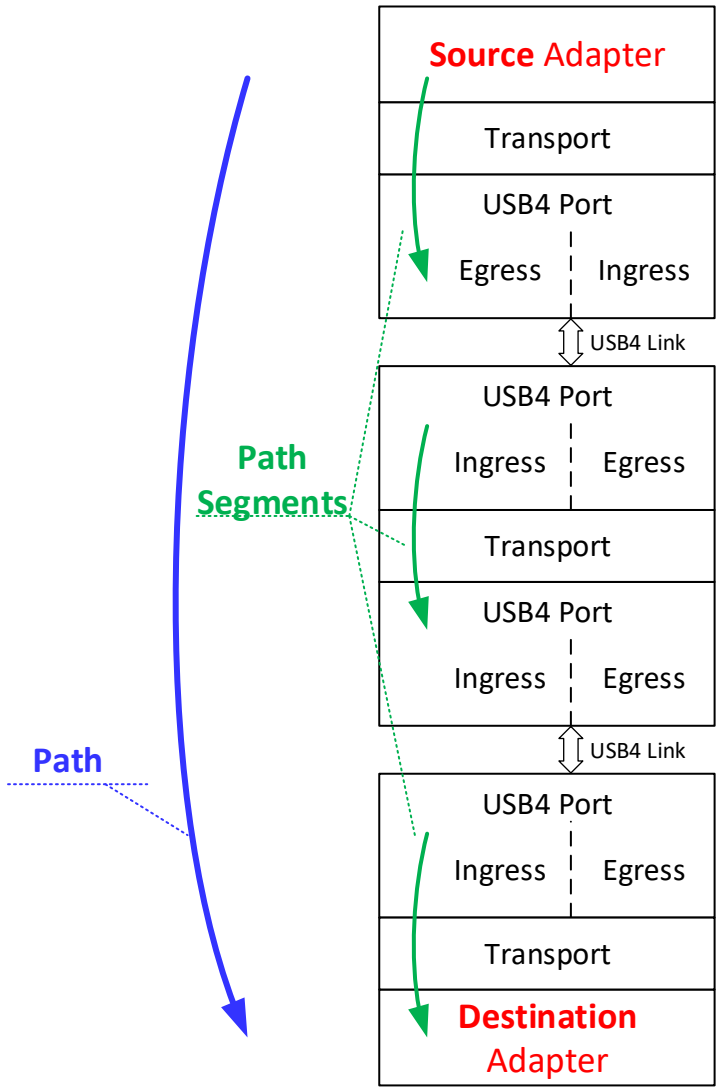
Tunneling Enable

- A Connection Manager sets the Tunneling types it will enable in the *Router CS*
  - *USB3 Tunneling*
  - *PCIe Tunneling*
- *A Connection Manager* polls the *Configuration Ready* bit which signifies that Paths can be setup

- Path setup includes
  - Configure QoS
  - Configure Flow Control Buffer
  - Setup the Path and enable the Adapters

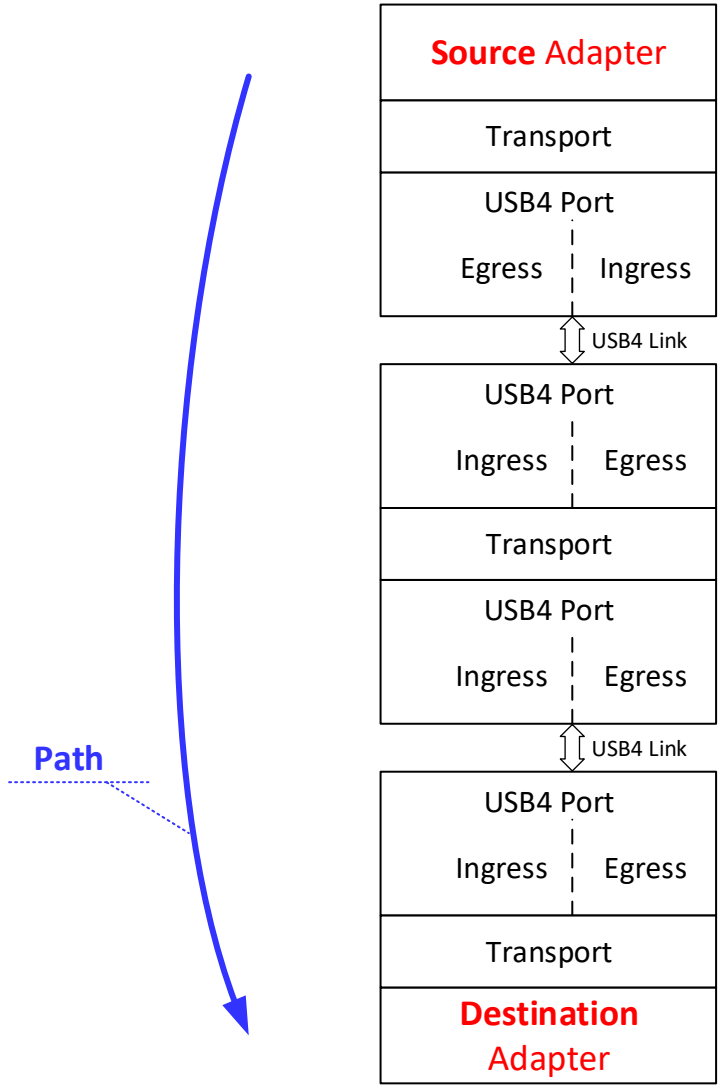
Path Terminology

- Source Adapter
  - The generator of the Tunneled Packets routed through the USB4 fabric
- Destination Adapter
  - The final recipient of Tunneled Packets routed through the USB4 fabric
- Path Segment
  - The portion of the Path within each Router
- Path
  - The one-way tunnel that carries data from a Source Adapter to a Destination Adapter



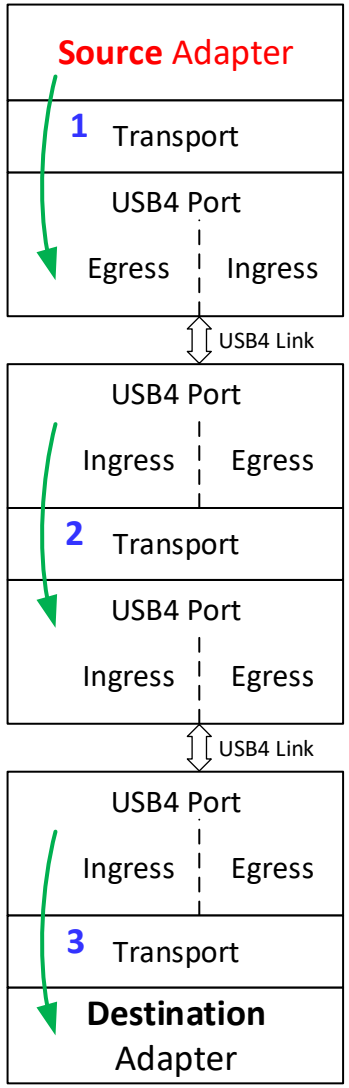
Path Setup

- A Connection Manager setup the Path only when all Routers that the Path traverses are enumerated and have the **Configuration Ready** bit set to 1b



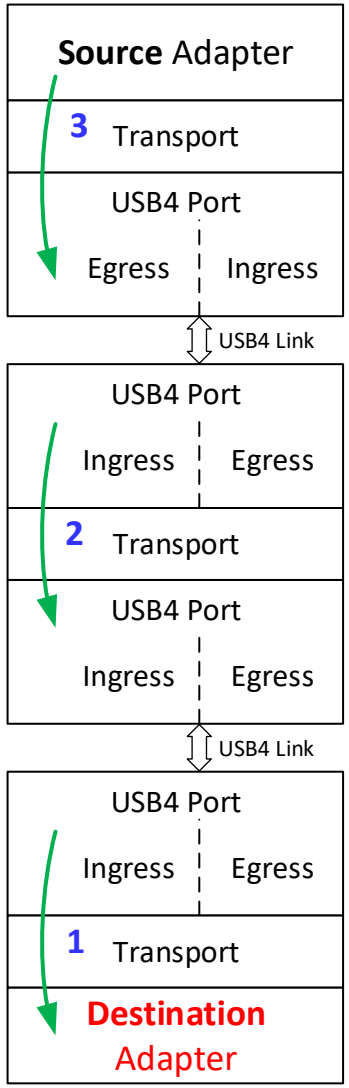
Path Setup

- The steps a Connection Manager takes to setup a Path
  - Configures the Path Segments – From Source to Destination Adapter
  - Writes to the *Path CS* of each Ingress Adapter along the Path configuring
    - HopID, Destination Adapter, QoS
    - Valid to 1b – Allows Transport Packet for that Path
  - Sets the Enable bit to 1b in the Adapter CS of both Adapter Ports
    - Allows Transport Packets for that Adapter



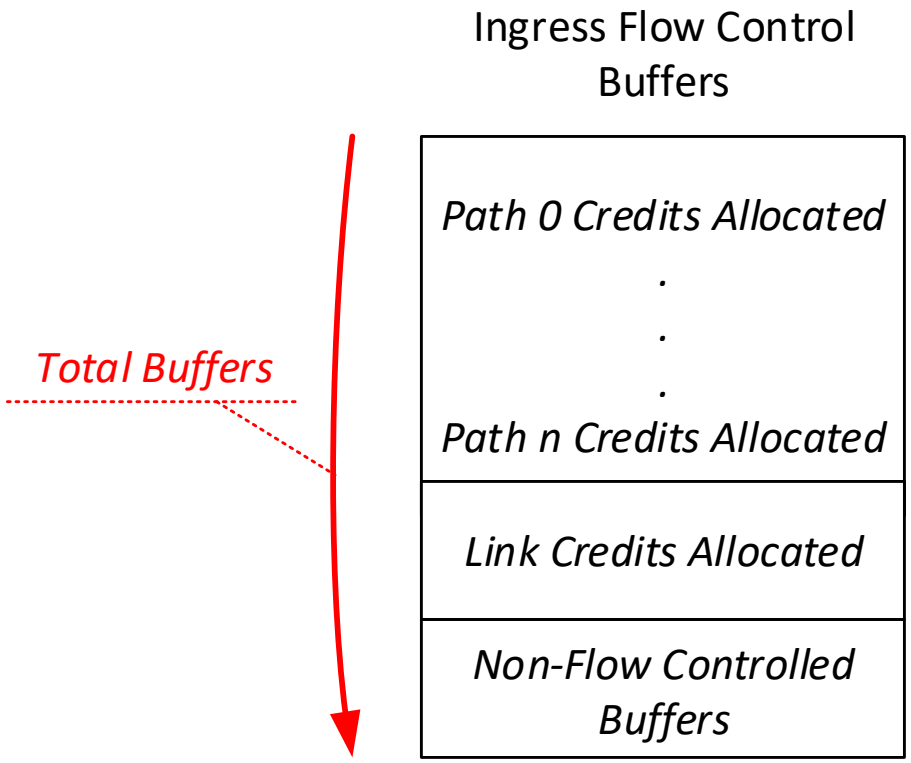
Path Teardown

- A Connection Manager Teardown
  - All Paths which traversed to/from a **disconnected Router**
  - A CM can also optionally perform Path Teardown any time after a Path is configured
- The steps a Connection Manager takes to teardown a Path
  - Sets the Enable bit to 0b in the Adapter CS of both Adapters
  - Teardown the Path Segments from Destination to Source Adapter by writing to the *Path CS* of each Ingress Adapter
    - Sets *Valid* to 0b
    - Poll *Pending Requests* field until it is 0b



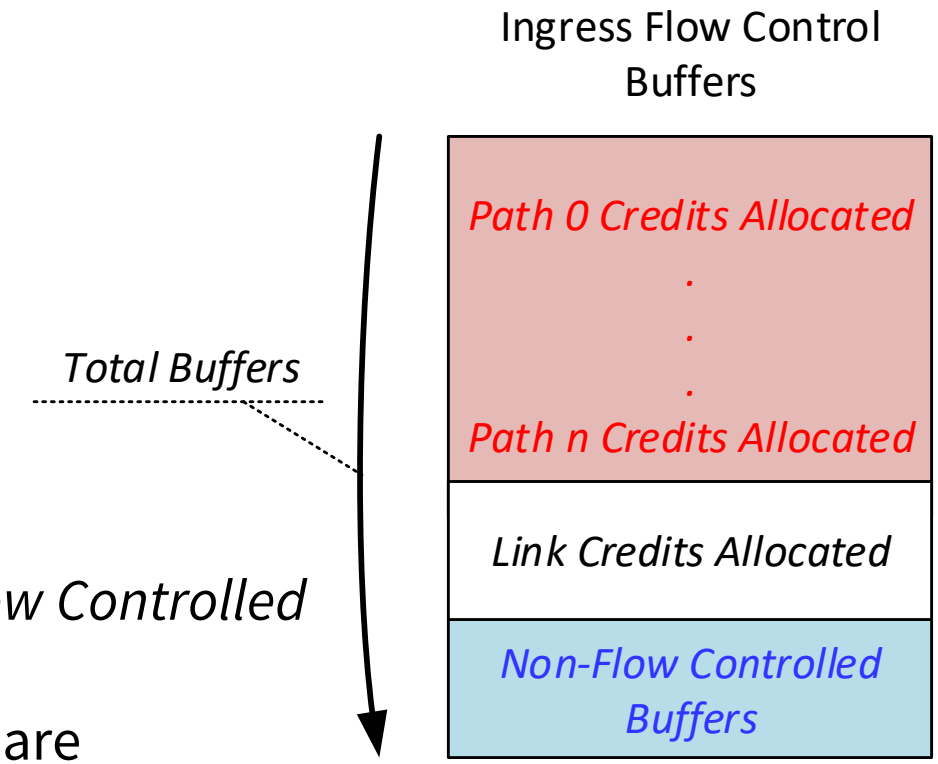
Flow Control

- A Connection Manager sets Flow Control Scheme per Path type
- A Connection Manager allocates Flow Control Buffers per Path, out of the overall *Total Buffers* of the Ingress Adapter
- A Connection Manager ensures a Path is configured to use the same Scheme on both sides of the USB4 Link



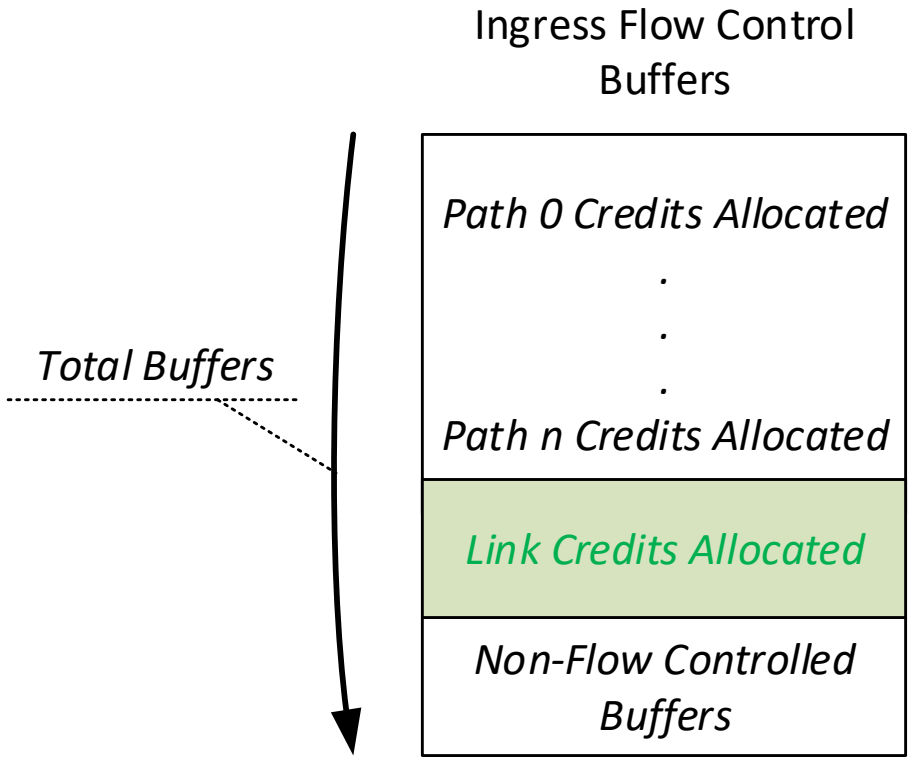
Flow Control

- **Dedicated Flow Control** Scheme
  - Configuration per Path - *Path Credits Allocated*
  - Static configuration, done at Path Setup
  - Used by Paths:
    - Control, DP Aux, PCIe, USB, Host Interface
- **Flow Control Disabled** Scheme
  - Single configuration field for all Paths – *Non-Flow Controlled Buffers*
  - Configuration may change on the fly – As Paths are Setup/Teardown
  - Used by DP Main-Link Path only



Flow Control

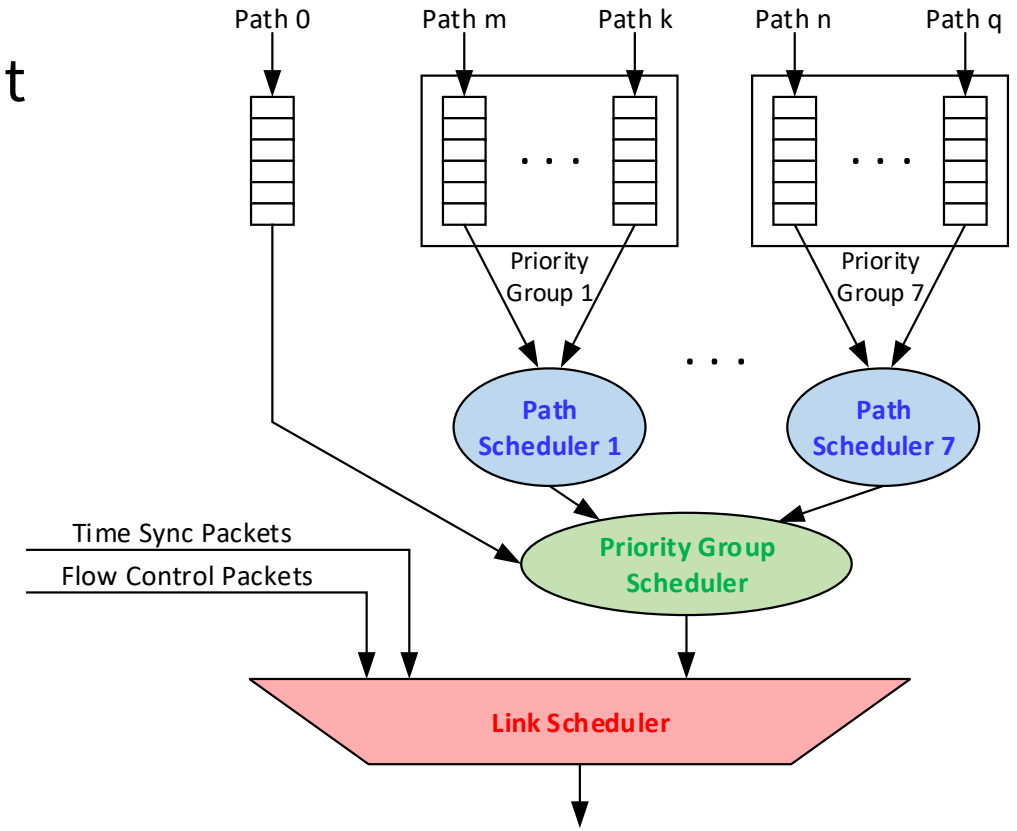
- **Shared Flow Control** Scheme
  - Single configuration field for all Paths – *Link Credits Allocated*
  - Static configuration, done at first shared Path Setup
  - Used by Host Interface Path only



Bandwidth Arbitration

- When a Connection Manager Setup a Path it must assure it has enough BW to operate

- Arbitration is executed in a distributed manner by each Egress Port
  - First Layer – Strict Priority
  - Second Layer – Strict Priority
  - Third Layer - Weighted Round Robin (WRR)



Bandwidth Clients

- *DisplayPort* – Main-Link Path is ***Isoch*** while Aux Paths are ***Non-Isoch***.
  - BW is guaranteed to #Lanes x BitRate x 0.8
- *USB* – **Isoch** and **Non-Isoch**
  - BW is guaranteed through CM-xHCI handshake
- *PCIe* – **Non-Isoch**
  - BW is indirectly guaranteed
- *Host Interface* – **Non-Isoch**
  - BW is not guaranteed

Bandwidth Concepts and Control

- Concepts
  - Guaranteed bandwidth cannot be taken away by a Connection Manager
  - When a DP Tunneling path is being established a Connection Manager tries to allocate the **maximum supported DP BW**
  - USB3 bandwidth is dynamic – **Allocate free bandwidth to USB3** Tunneling

- Controls
  - DisplayPort Tunneling
    - Control Maximum Link Rate and Lane Count
  - USB3 Tunneling
    - CM-xHCI BW handshake
  - Path Settings (Priority and WRR)

Protocol	Priority	WRR
Control	0	NA
DP	1	NA
USB	2	2
PCIe	2	1
Host-to-Host	3	NA

Bandwidth Management Rule

- Before allocating new BW, a CM ensures that, in each USB 4 Port, the calculated Available BW is not negative (per direction):

Available BW = **[0.9 * (USB4 Link BW)]** –
[DisplayPort BW +
USB3 Tunnel Allocated BW * (USB3 WRR + PCIe WRR)/(USB3 WRR)]

Time for Q&A

Agenda

- Time Synchronization
- Host Interface
- Connection Manager
- Thunderbolt™ 3 Compatibility

Agenda

- Terminology
- USB4™ Vs Thunderbolt™ 3
- Thunderbolt™ 3 Requirements
- Mix Topology

Thunderbolt™ 3 Terminology

- *Thundebolt 3* is the existing technology used by the **current TBT3 ecosystem**, containing
  - Hosts
  - Devices
  - Connection Manager
  - Cables

- *Thunderbolt 3 compatibility* – USB4 Specifications defines requirements for a **new USB4 component** to be TBT3-Compatible
  - USB4 Host
  - USB4 Device
  - USB4 Connection Manager
  - USB4 Re-timer

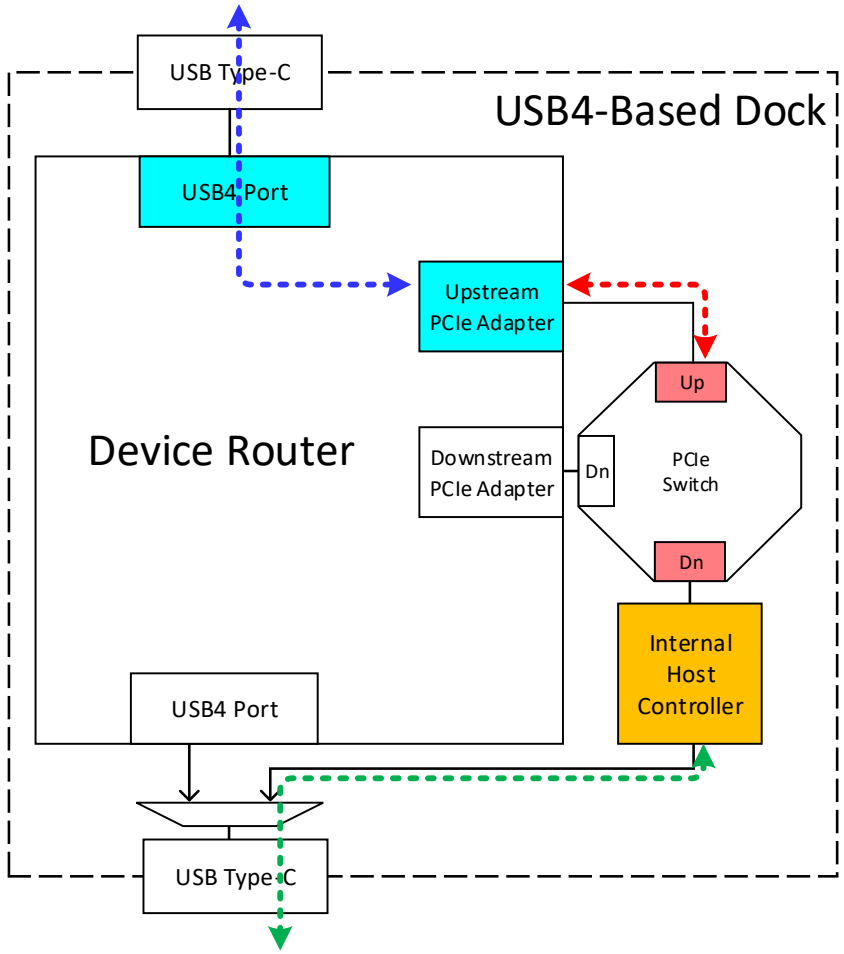
USB4™ Vs Thunderbolt™ 3

- **Chapter 13** in USB4 Specification and **Chapter 6** in Re-timer Specification defines all the implementation requirements needed by a USB4 component to be TBT3-Compatible. The ***Major*** differences are listed below.

	TBT3-Compatible	USB4
Electrical Layer	Gen2 – 10.3125Gbps Gen3 – 20.625Gbps	Gen2 – 10Gbps Gen3 – 20Gbps
Logical Layer & Re-timer	Support Bi-Directional Re-timer Support Two-Single Lane Links Changes in Lane Initialization Change in TxFFE process	
Transport Layer	Lane 1 can act as Upstream Adapter	
Configuration Spaces	Additional CS requirement to allow TBT3 Connection Manager operation	
USB3 Functionality	Support through USB3 Host Controller at USB4 Device	USB3 Tunneling

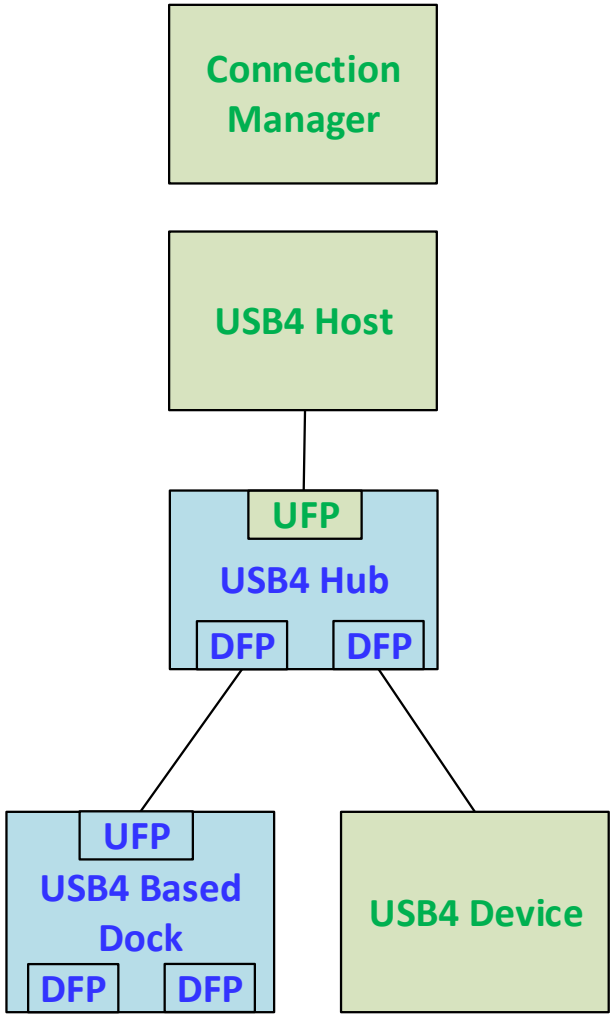
USB4™ Vs Thunderbolt™ 3

- When a USB4-Based Dock establishes a Thunderbolt 3 Link on its UFP, the support of Native USB3 on its DFPs is accomplished by:
  - Incorporating a USB3 Host Controller into the USB4-Based Dock
  - Connecting the Host Controller to the Host using **PCIe Tunneling**



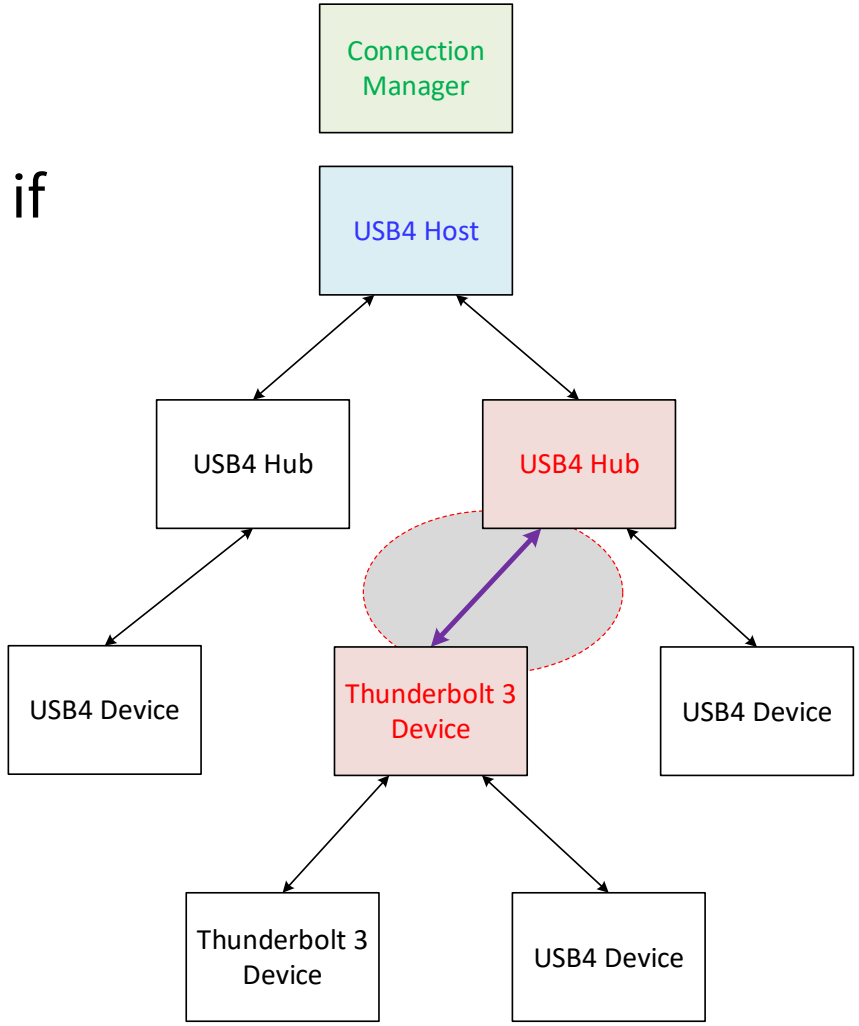
Thunderbolt™ 3 Support Requirement

- A **USB4 host** along with its **Connection Manager** may ***optionally*** support TBT3-Compatibility.
- A **USB4 hub must** support TBT3-Compatibility on all of its **DFP**.
- A **USB4-Based Dock must** support TBT3-Compatibility on its **UFP** in addition to all its **DFP**.
- A **USB4 peripheral device** may ***optionally*** support TBT3-Compatibility.



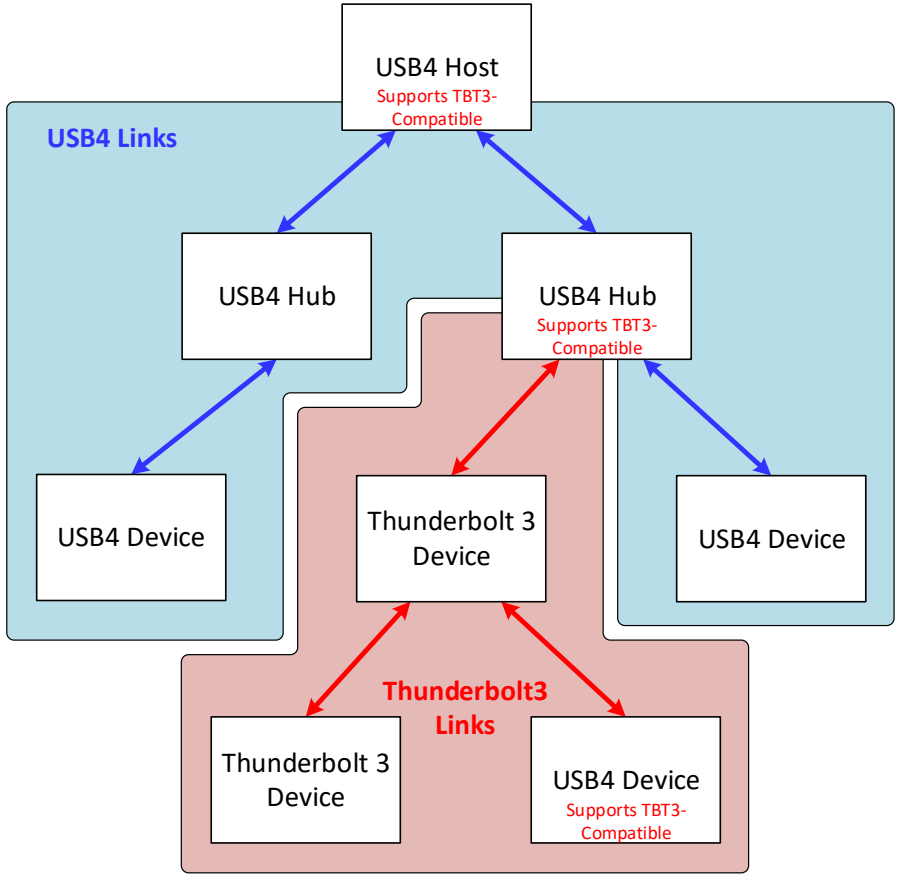
Thunderbolt™ 3 Alt Mode Conditions

- A Thunderbolt 3 Alternate Mode can be established if each of the components below are either a Thunderbolt 3 component or a USB4 component which is TBT3-Compatible
  - The Routers on both sides of the Link
  - The Host Router
  - The Connection Manager
  - The Active cable, if exists



Thunderbolt™ 3 and USB4™ Mix topology

- Once a Thunderbolt 3 Link was established, all the downstream links below that link, are prohibited to be USB4 Links



Time for Q&A