USB4^{TM} Configuration Layer, USB3 Tunneling, DP Tunneling and PCIe Tunneling

Gal Yedidia *– Architect, Intel Corporation*

USB Developer Days 2019 – Taipei, Taiwan

November 20, 2019

USB

Enabling Connections™

Agenda

- Configuration Layer
- USB3 Tunneling
- DP Tunneling
- PCIe Tunneling

Agenda

- Configuration Layer
- USB3 Tunneling
- DP Tunneling
- PCIe Tunneling

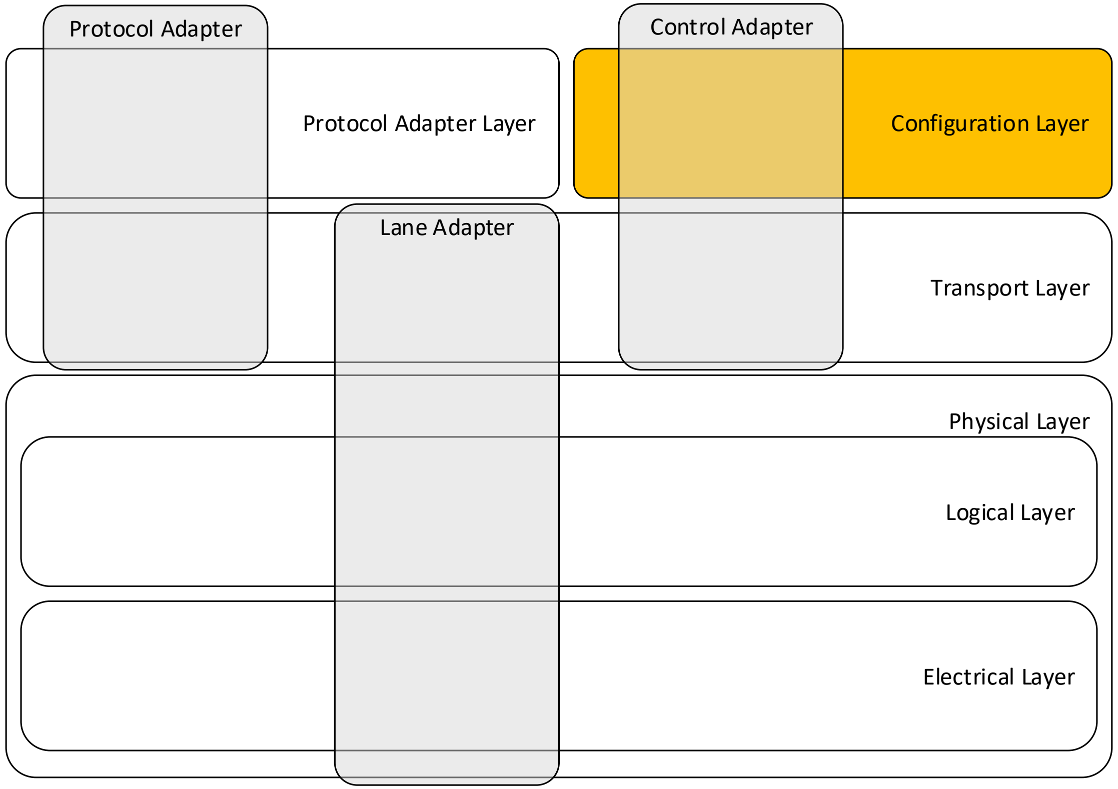
Configuration Layer Agenda

- Roles
- Control Adapter
- Topology ID
- Control Packets
- Configuration Spaces
- Operations

Configuration Layer Roles

- Provides the control plane, used by Connection Manager and Routers
  - Discovery
  - Configuration
  - Notification

- Provides Control channel between two Connection Managers



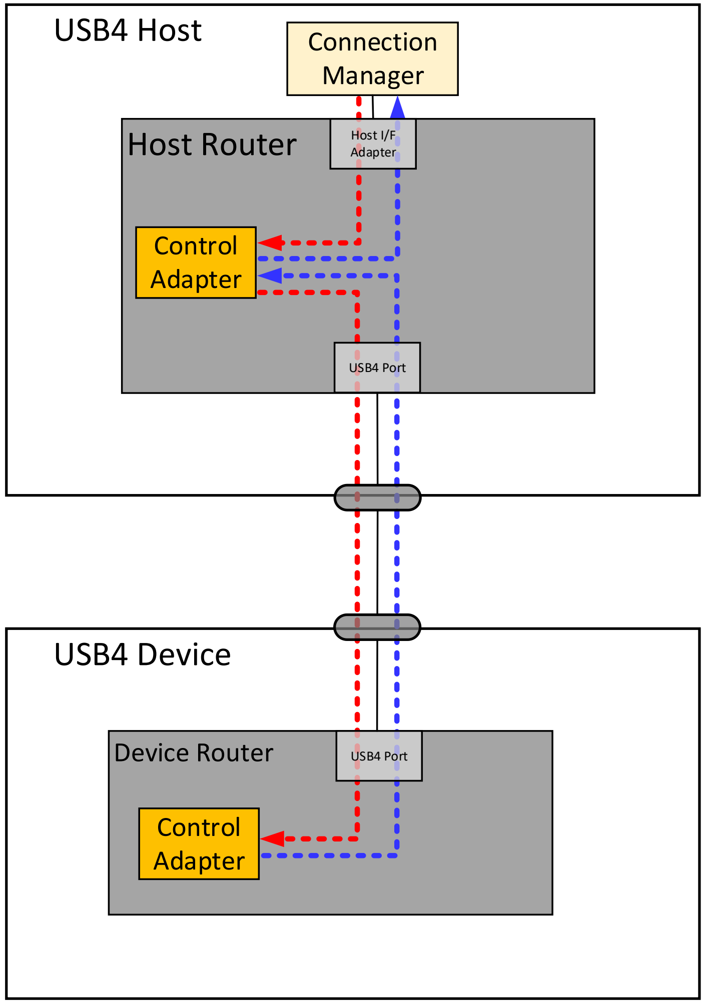
Control Adapter

- Exists as Adapter 0 at every Router

- Attends all control requests received from the Connection Manager

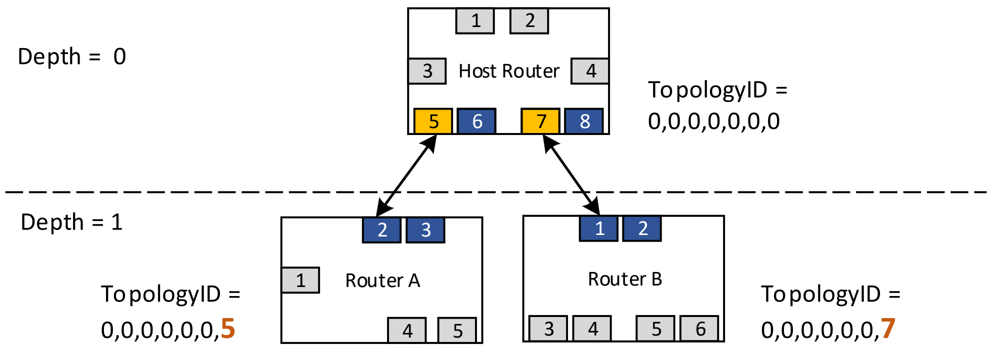
- Generates all needed control responses and notifications to the Connection Manager

- Participates in Control Packets Routing through Path 0



Topology ID

- Uniquely assigned for each Router by the Connection Manager along with the **Depth**



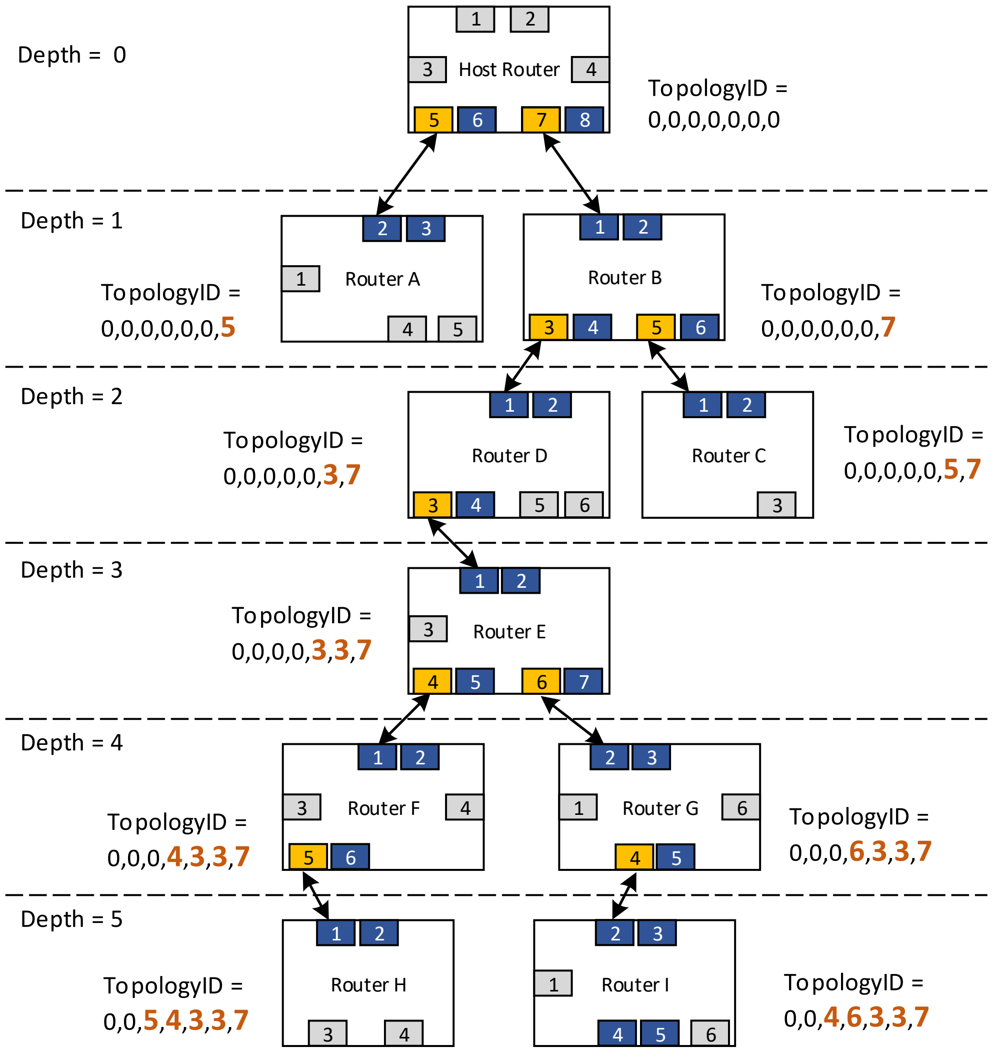
Topology ID

- Uniquely assigned for each Router by the Connection Manager along with the **Depth**
    - Once written, a Router is Enumerated

- Represents the position of the Router within the Domain

- Uses for Control Packets routing

- The TopologyID at depth ‘**X**‘ is denoted as **0,…,0,Px-1,Px-2,…,P0** where **Pn** is the Adapter Number of the Downstream Facing Adapter at Depth ‘n’.



Depth = 0

Host Router

TopologyID = 0,0,0,0,0,0,0

Depth = 1

Router A

TopologyID = 0,0,0,0,0,0,5

Router B

TopologyID = 0,0,0,0,0,0,7

Depth = 2

Router D

TopologyID = 0,0,0,0,0,3,7

Router C

TopologyID = 0,0,0,0,0,5,7

Depth = 3

Router E

TopologyID = 0,0,0,0,3,3,7

Depth = 4

Router F

TopologyID = 0,0,0,4,3,3,7

Router G

TopologyID = 0,0,0,6,3,3,7

Depth = 5

Router H

TopologyID = 0,0,5,4,3,3,7

Router I

TopologyID = 0,0,4,6,3,3,7

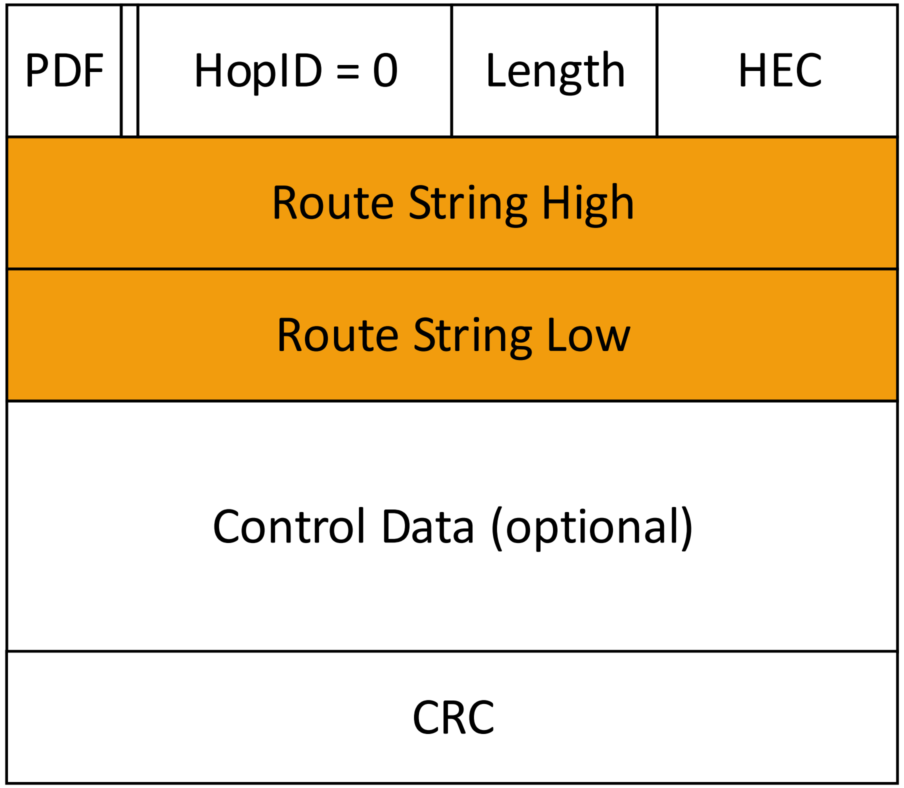
Control Packets

<table>
<tr><td>PDF</td><td></td><td>HopID = 0</td><td>Length</td><td>HEC</td></tr>
<tr><td colspan="5">Route String High</td></tr>
<tr><td colspan="5">Route String Low</td></tr>
<tr><td colspan="5">Control Data (optional)</td></tr>
<tr><td colspan="5">CRC</td></tr>
</table>

- **Header**
  - *HopID* = 0
  - *PDF* sets the Type

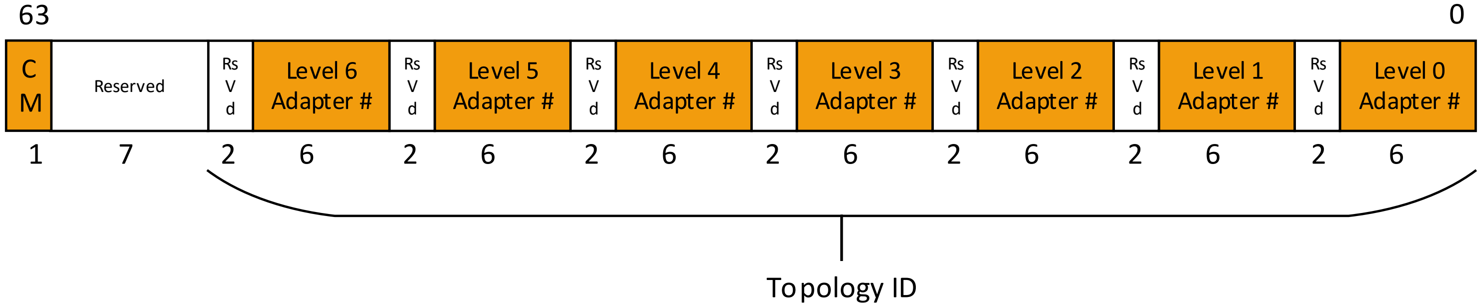
PDF	Type
1	Read Request & Read Response
2	Write Request & Write Response
3	Notification
4	Notification Acknowledgment
5	Hot Plug Event
6	Inter-Domain Request
7	Inter-Domain Response

Control Packets



- **Route String**
  - *TopologyID* + *CM* bit

	Downstream-Bound	Upstream-Bound
Initiated By	Connection Manager	Router (Control Adapter)
CM bit	0b	1b
TopologyID	Target Router	Initiated Router



Control Packets

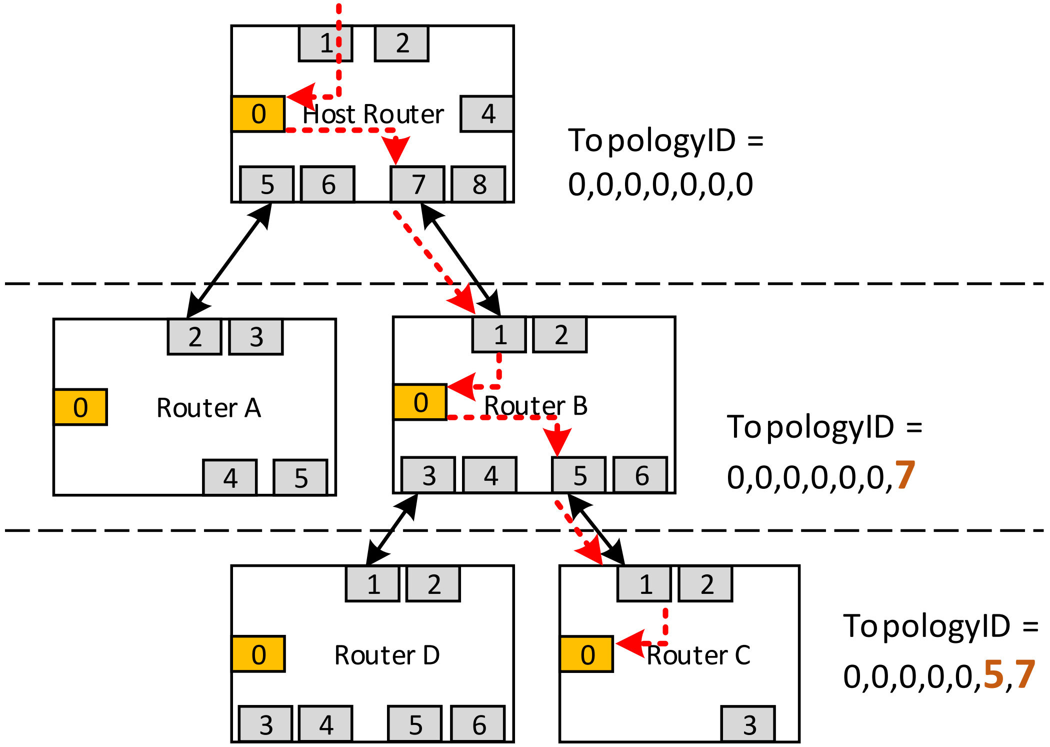
PDF | HopID = 0 | Length | HEC

Route String High

Route String Low

Control Data (optional)

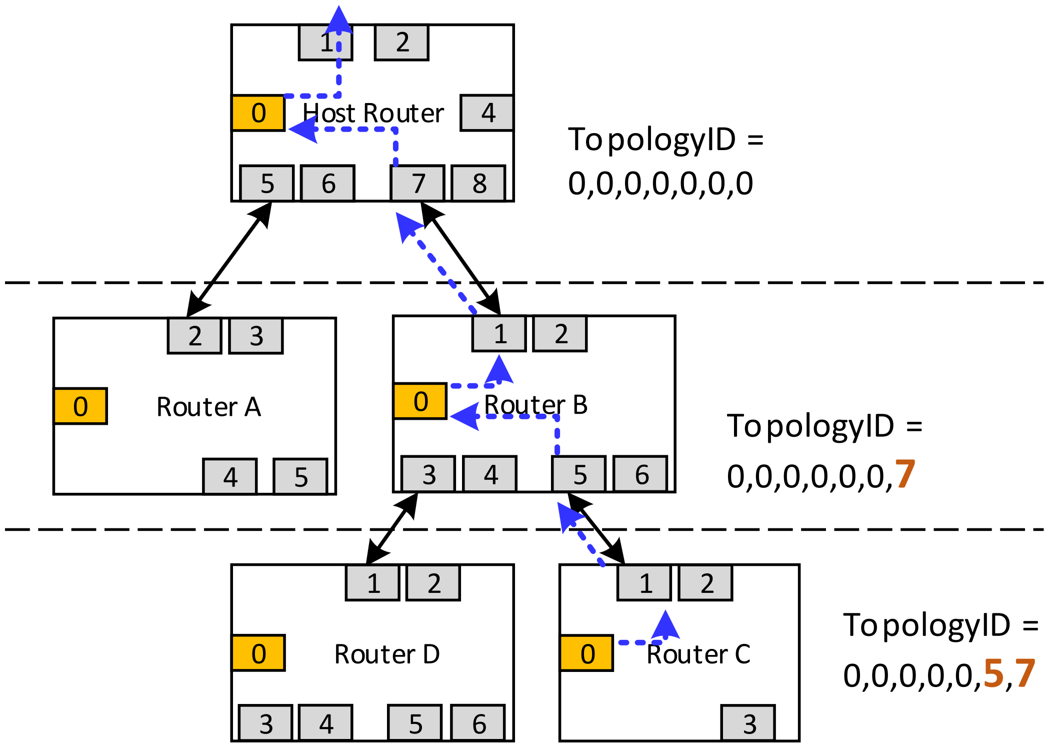
CRC



- **Routing** – **Downstream-Bound** to Router **C**
  - *Route String = 00000057h*
  - Control Adapters of the '*Host Router*' and '*Router-B*' are forwarding the Packet
  - Control Adapter in '*Router-C*' consumes the Packet

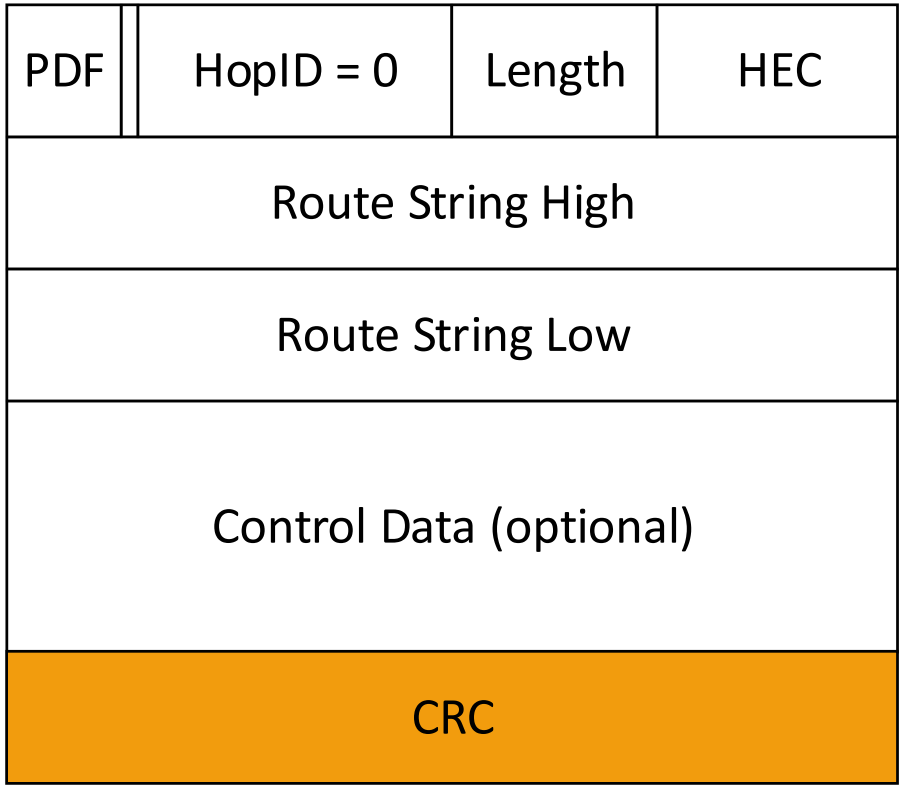
Control Packets

PDF		HopID = 0	Length	HEC
Route String High				
Route String Low				
Control Data (optional)				
CRC				



- **Routing** – Upstream-Bound From Router **C**
  - *Route String = 80000057h (CM = 1b)*
  - Control Adapter in '*Router-C*' initiates the Packet
  - Control Adapters of the '*Host Router*' and '*Router-B*' are forwarding the Packet

Control Packets



PDF | HopID = 0 | Length | HEC

Route String High

Route String Low

Control Data (optional)

CRC

- **Reliability**
  - *CRC* – 32 bits CRC covers the entire Control Packet Payload
  - Response **time** to a Control Packet is bounded
  - Notification Packets are retransmitted by Routers until acknowledgment is received

Configuration Spaces (CS)

- Accessed by a Connection Manager, using **Read Request** & **Write Request** Control Packets.
  - Used for Discovery and Configuration
  - Read Write Registers

Read Request

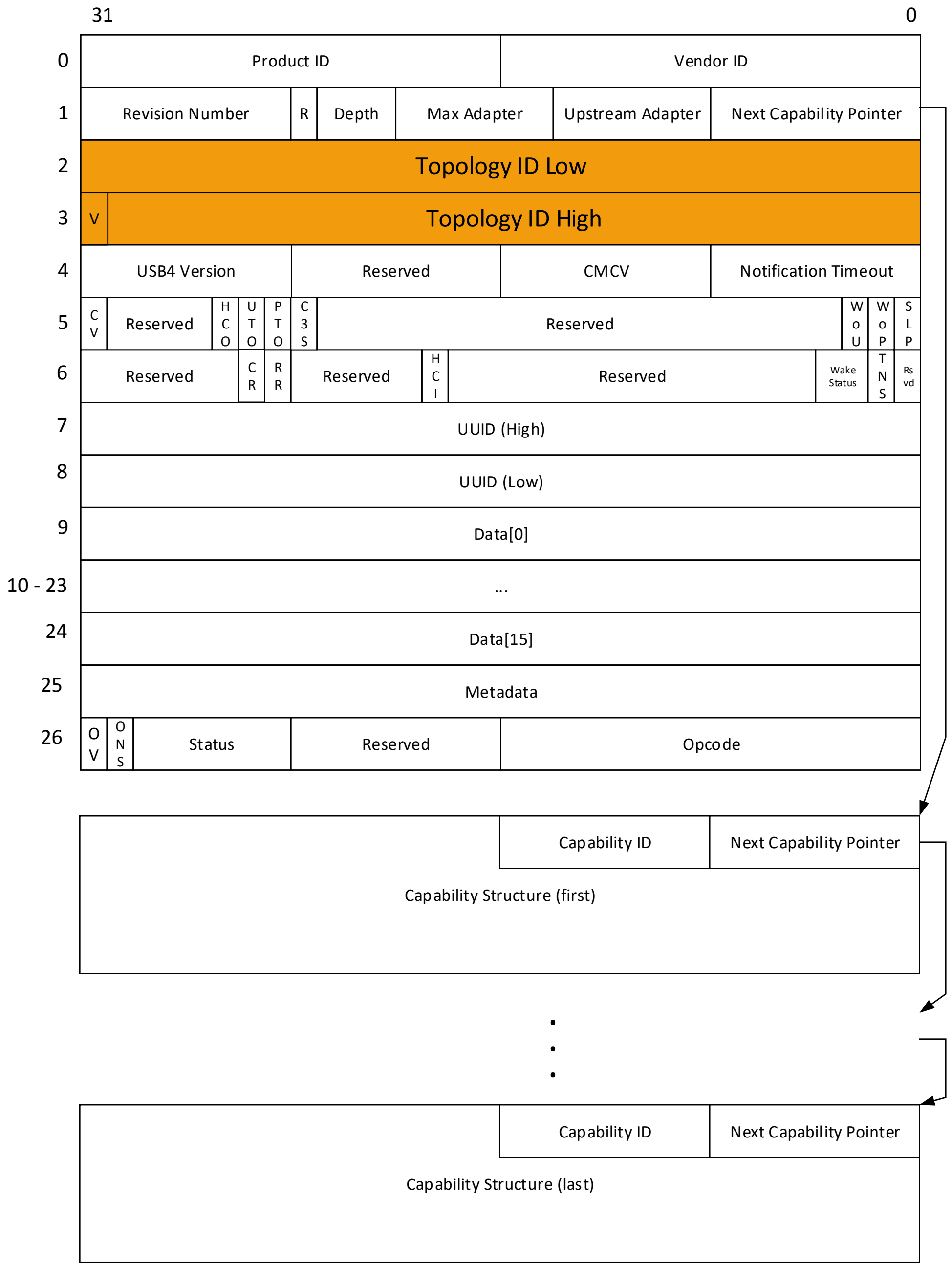
Header					
Route String High					
Route String Low					
Rsvd	SN	**CS**	Adapter Num	Read Size	Address
CRC					

Write Request

Header					
Route String High					
Route String Low					
Rsvd	SN	**CS**	Adapter Num	Write Size	Address
Write Data (1-60 DWs)					
CRC					

Configuration Spaces

- **Router CS**
  - One per Router
  - Information and Control at the Router level
  - Link list: Basic + TMU + Vendor specific
- **Adapter CS**
- **Path CS**
- **Counters CS**



	31									0
0	Product ID	Vendor ID								
1	Revision Number	R	Depth	Max Adapter	Upstream Adapter	Next Capability Pointer				
2	Topology ID Low									
3	V	Topology ID High								
4	USB4 Version	Reserved	CMCV	Notification Timeout						
5	CV	Reserved	HCO	UTO	PTO	C3S	Reserved	WoU	WoP	SLP
6	Reserved	CR	RR	Reserved	HCI	Reserved	Wake Status	TNS	Rsvd	
7	UUID (High)									
8	UUID (Low)									
9	Data[0]									
10 - 23	...									
24	Data[15]									
25	Metadata									
26	OV	ONS	Status	Reserved	Opcode					

Capability ID	Next Capability Pointer

Capability Structure (first)

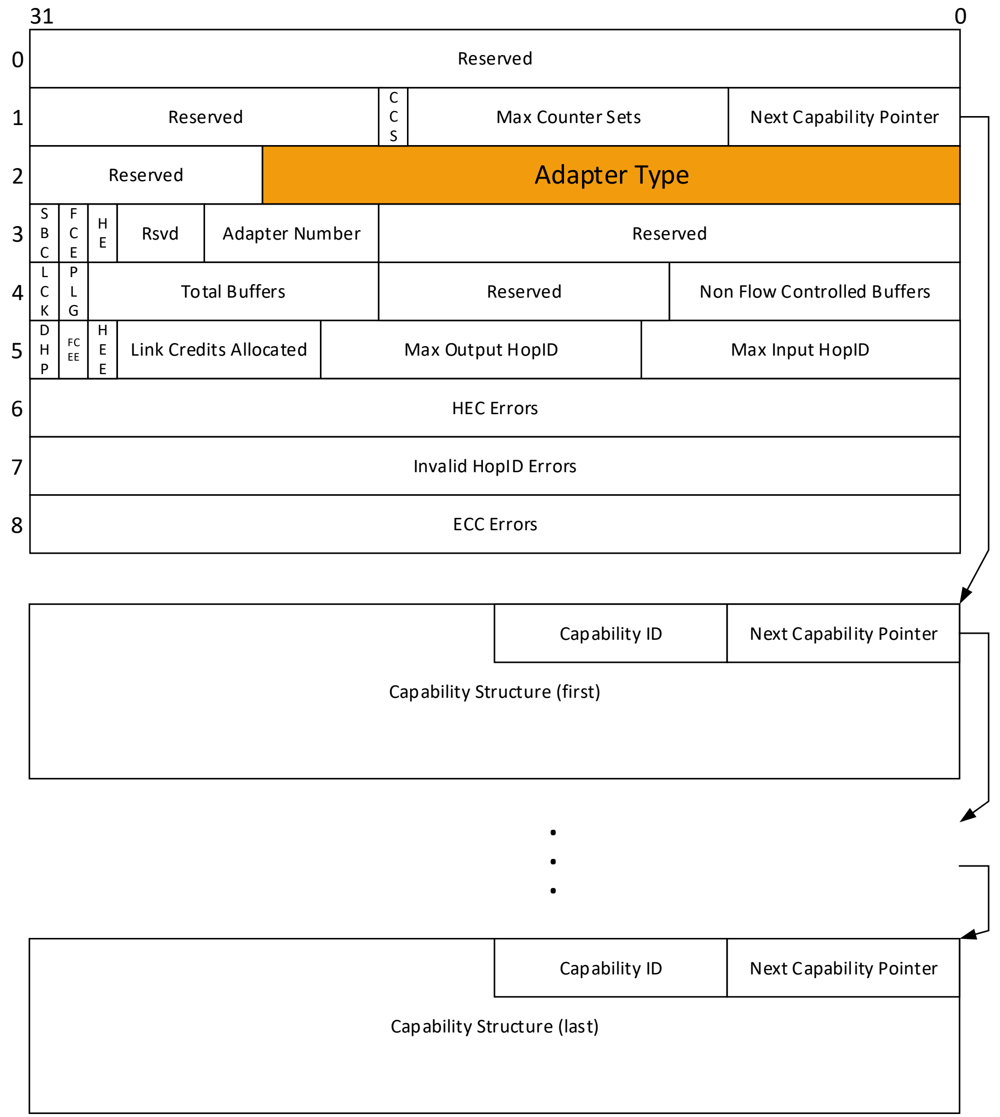
•
•
•

Capability ID	Next Capability Pointer

Capability Structure (last)

Configuration Spaces

- Router CS
- **Adapter CS**
  - One per Adapter
  - Information and Control at Adapter level
  - Link list: Basic + Adapter specific + Vendor specific
- Path CS
- Counters CS



31 0

0 Reserved

1 Reserved | CCS | Max Counter Sets | Next Capability Pointer

2 Reserved | Adapter Type

3 SBC | FCE | HE | Rsvd | Adapter Number | Reserved

4 LCK | PLG | Total Buffers | Reserved | Non Flow Controlled Buffers

5 DHP | FCEE | HEE | Link Credits Allocated | Max Output HopID | Max Input HopID

6 HEC Errors

7 Invalid HopID Errors

8 ECC Errors

Capability ID | Next Capability Pointer

Capability Structure (first)

Capability ID | Next Capability Pointer

Capability Structure (last)

Configuration Spaces

- Router CS
- Adapter CS
- **Path CS**
  - One per Adapter
  - Size is according to the number of supported Paths
  - One entry per Path – 2 DWs
  - Entry attributes are different for
    - Path 0 Vs Non zero
    - Lane Adapter Vs Protocol Adapter
- Counters CS

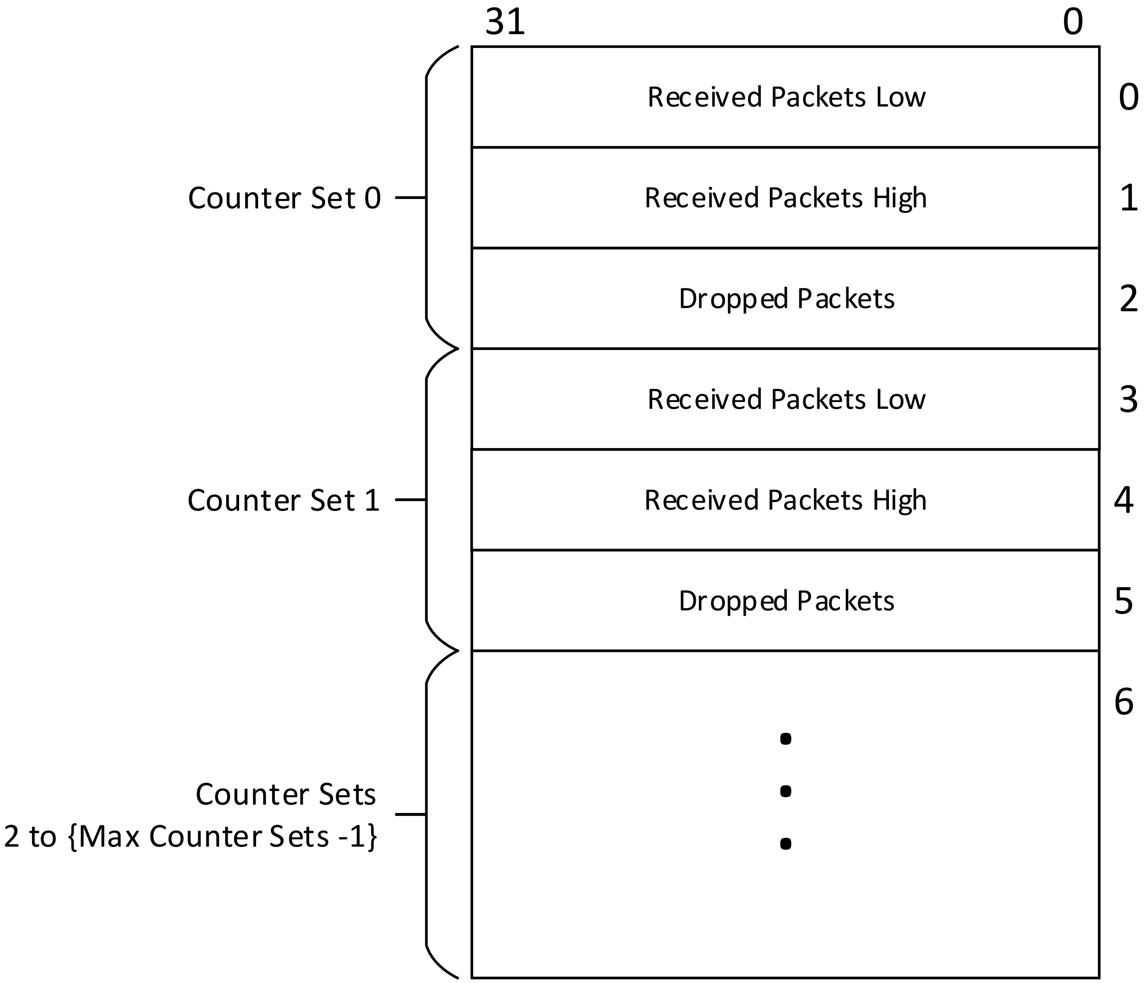
Path 0

<table>
<tr><td></td><td>31</td><td></td><td></td><td></td><td>0</td></tr>
<tr><td>0</td><td>Reserved</td><td colspan="2">Path Credits Allocated</td><td colspan="2">Reserved</td></tr>
<tr><td>1</td><td>Rsvd</td><td>CE</td><td colspan="2">Counter ID</td><td>Reserved</td></tr>
</table>

<table>
<tr><td></td><td>31</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>0</td></tr>
<tr><td>2 x n</td><td>V</td><td colspan="6">Reserved</td><td colspan="2">Path Credits Allocated</td><td colspan="2">Output Adapter</td><td colspan="2">Output HopID</td></tr>
<tr><td>2 x n +1</td><td colspan="2">Reserved</td><td>PP</td><td>ESE</td><td>ISE</td><td>EFC</td><td>IFC</td><td>CE</td><td colspan="2">Counter ID</td><td>Rsvd</td><td>Priority</td><td>Weight</td></tr>
</table>

Configuration Spaces

- Router CS
- Adapter CS
- Path CS
- **Counters CS**
  - One per Adapter
  - Optional
  - Size is according to number of supported Counter Sets
  - Each Counter Set counts Path(s) statistics



31 0

Counter Set 0

0 Received Packets Low

1 Received Packets High

2 Dropped Packets

Counter Set 1

3 Received Packets Low

4 Received Packets High

5 Dropped Packets

Counter Sets
2 to {Max Counter Sets -1}

6 • • •

Operations

- Initiated by a Connection Manager, targeting a Router or a Port
- A request for an elaborated task which might require or return a data structure

- **Router Operations**
  - CM uses the Router CS to initiate a Router Operation
    - DP Operations
    - NVM Operations
    - Discovery Operations
    - Port Control Operations
- **Port Operations**

Router Operations
Query DP Resource Availability
Allocate DP Resource
De-allocate DP Resource
NVM Write
NVM Authenticate Write
NVM Read
NVM Set Offset
DROM Read
Get NVM Sector Size
Get PCIe Downstream Entry Mapping
Get Capabilities
Set Capabilities
Buffer Allocation Request
Get Container-ID
Block Sideband Port Operations
Unblock Sideband Port Operations
Vendor Specific Router Operations

Operations

- Initiated by a Connection Manager, targeting a Router or a Port
- A request for an elaborated task which might requires or returns a data structure

- Router Operations
- **Port Operations**
  - CM uses the SB Register Space to initiate a Port Operation
  - The *Target Port* could be in a local Router, Re-timer or a remote Router
    - Set special modes Operations
    - Execute compliance tests Operations
    - Lane Margining Operations

Port Operations
SET_TX_COMPLIANCE
SET_RX_COMPLIANCE
ROUTER_OFFLINE_MODE
START_BER_TEST
END_BER_TEST
END_BURST_TEST
READ_BURST_TEST
ENTER_EI_TEST
ENUMERATE_RE-TIMERS
READ_LANE_MARGIN_CAP
RUN_HW_LANE_MARGINING
RUN_SW_LANE_MARGINING
READ_SW_MARGIN_ERR

Time for Q&A

Agenda

- Configuration Layer
- USB3 Tunneling
- DP Tunneling
- PCIe Tunneling

USB3 Tunneling Agenda

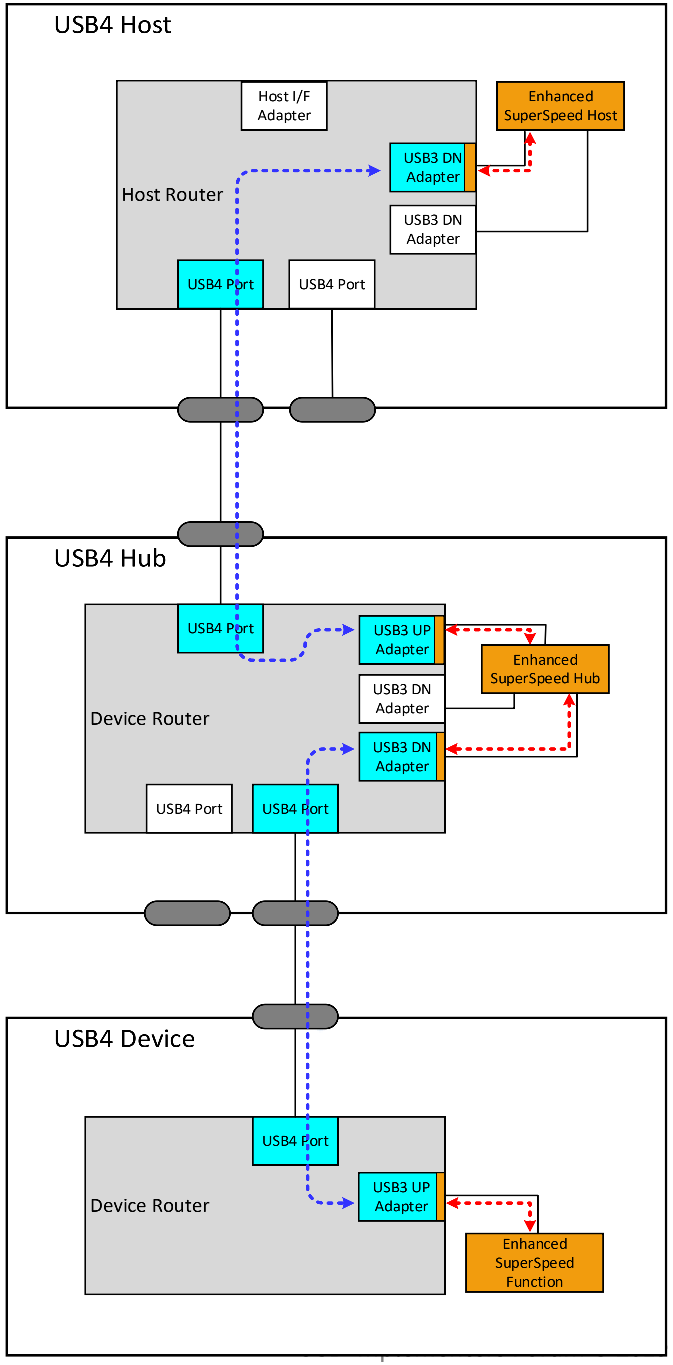
- System View
- Internal USB3 Device
- Protocol Stack
- USB3 Adapter
  - Paths
  - Encapsulation

System View

- Native USB Enhanced SuperSpeed (USB3) is Tunneled over USB4 Fabric

- Originates and consumed as Native Enhanced SuperSpeed protocol
- From USB3 SW perspective, the USB3 topology remains the same

- **USB3 Adapters** are the translators within each Router that allows USB3 protocol to travel back an forth from Native to Tunneled

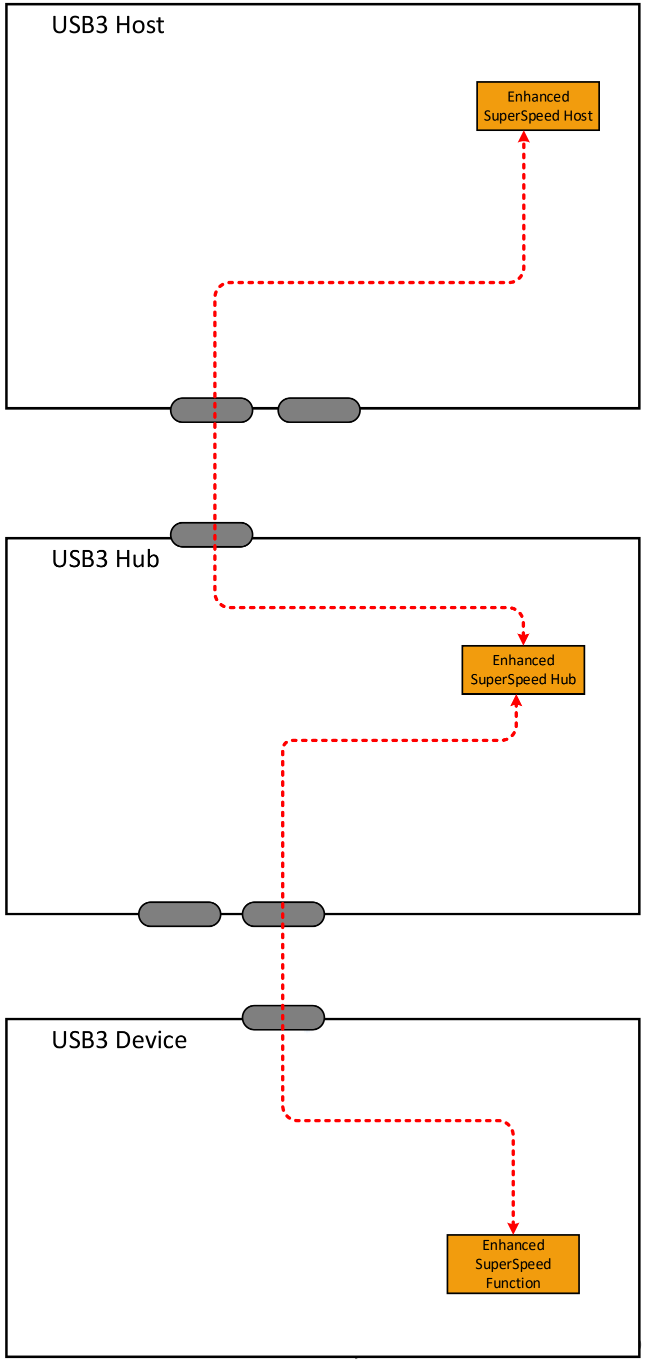


System View

- Native USB Enhanced SuperSpeed (USB3) is Tunneled over USB4 Fabric

- Originates and consumed as Native Enhanced SuperSpeed protocol
- From USB3 SW perspective, the USB3 topology remains the same

- **USB3 Adapters** are the translators within each Router that allows USB3 protocol to travel back an forth from Native to Tunneled

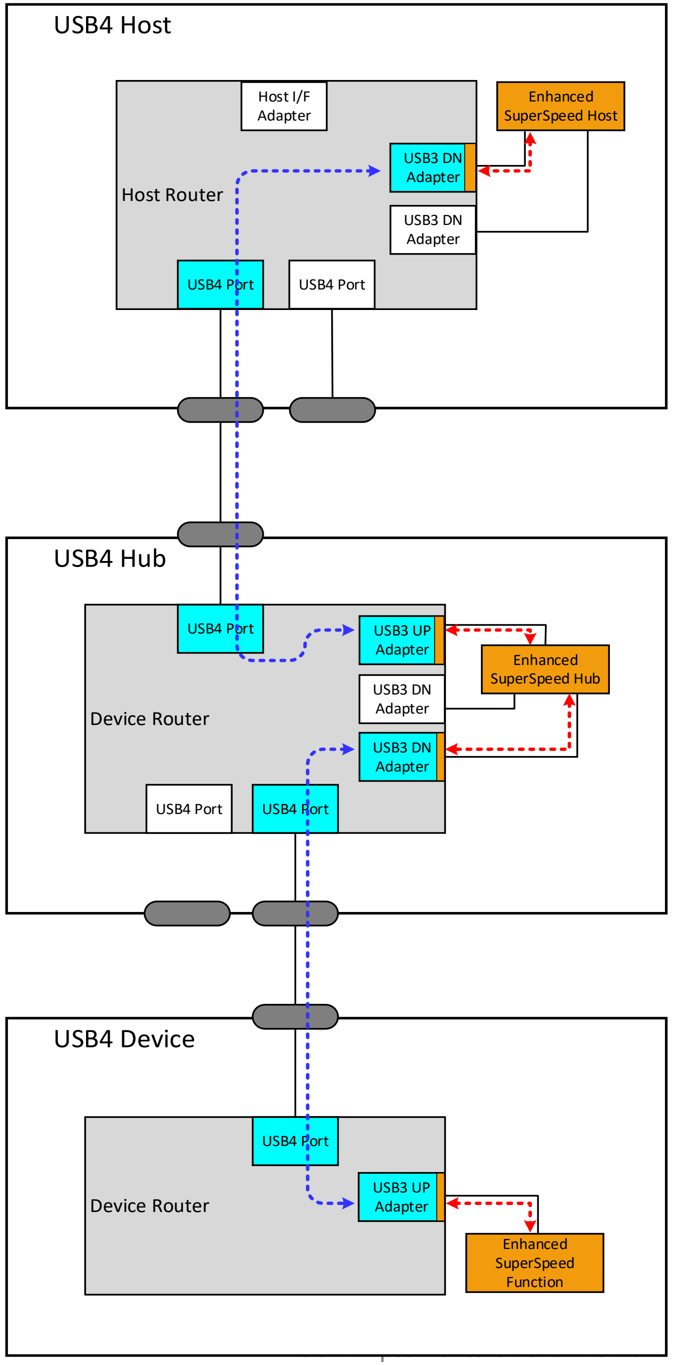


System View

- Native USB Enhanced SuperSpeed (USB3) is Tunneled over USB4 Fabric

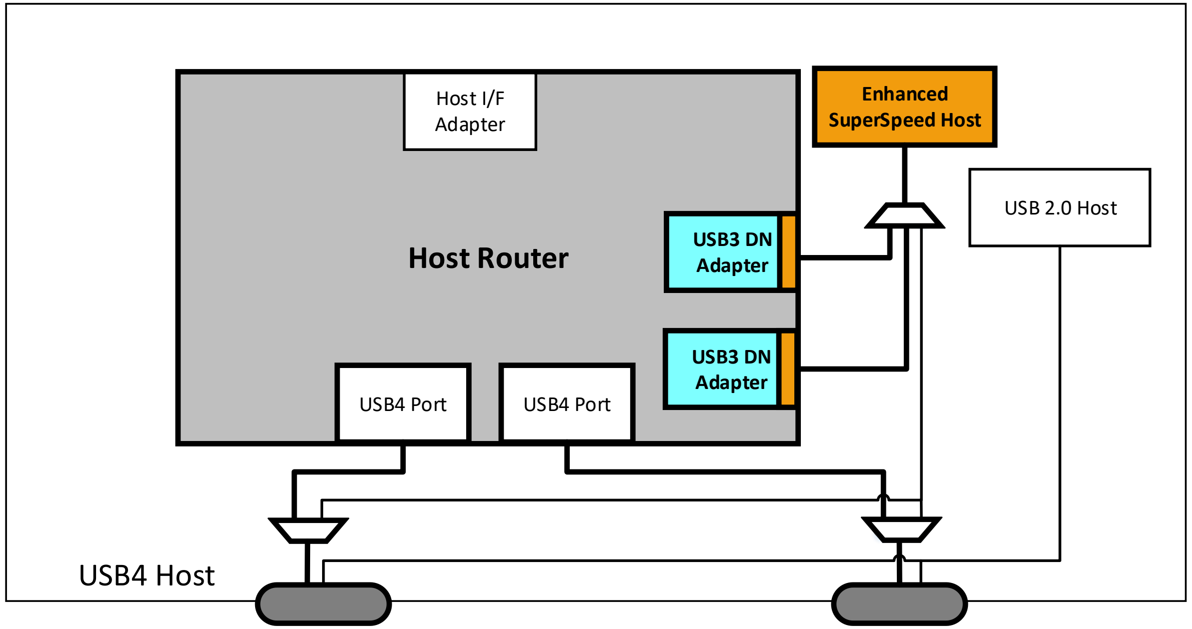
- Originates and consumed as Native Enhanced SuperSpeed protocol
- From USB3 SW perspective, the USB3 topology remains the same

- **USB3 Adapters** are the translators within each Router that allows USB3 protocol to travel back an forth from Native to Tunneled

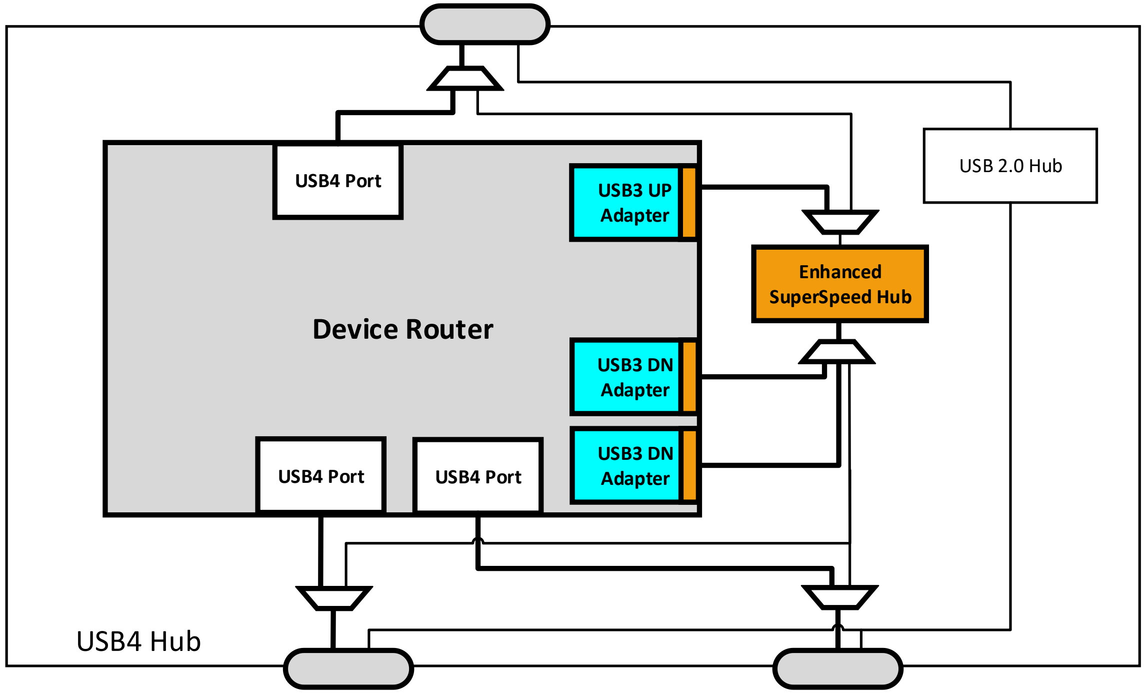


System View – USB4™ Host

- *USB4 Host* must support USB3 Tunneling
- *USB4 Host* implements an Enhanced SuperSpeed Host controller
- *Host Router* has ‘*N*’ USB3 Downstream Adapters
  - ‘*N*’ – Number of Downstream USB Type-C connectors
- *USB4 Host* maintains backward compatibility and is also a USB3 Host



System View – USB4^{TM} Hub

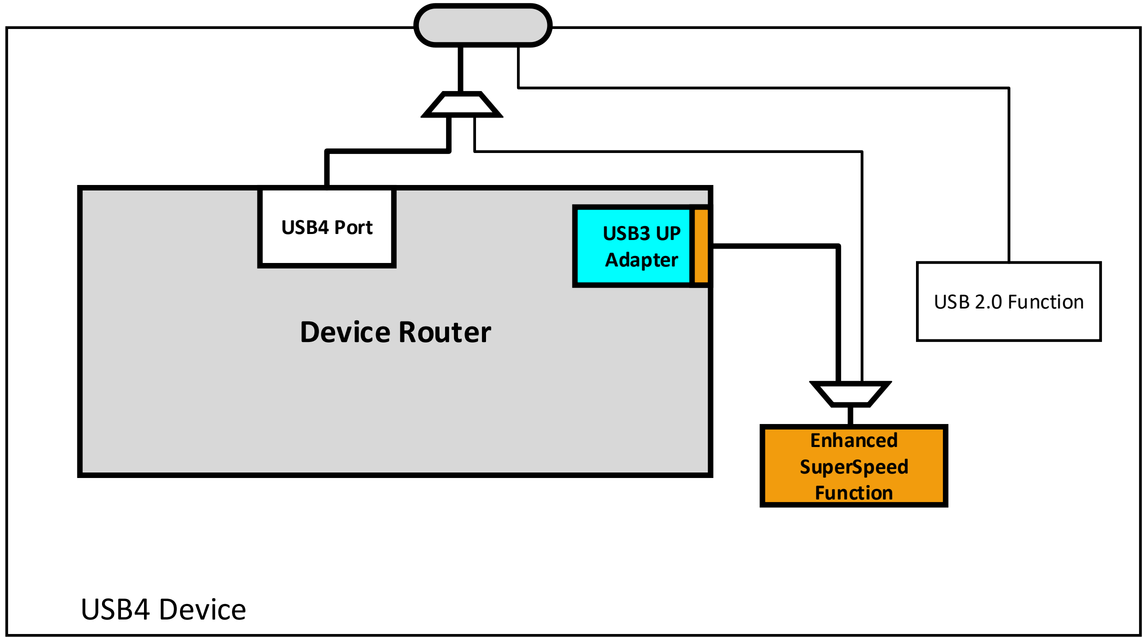


- *USB4 Hub* must support USB3 Tunneling
- *USB4 Hub* implements an Enhanced SuperSpeed Hub
- *Device Router* implements a USB3 Upstream Adapter
- *Device Router* has ‘*N*’ USB3 Downstream Adapters
    - ‘*N*’ – Number of Downstream USB Type-C connectors
- *USB4 Hub* maintains backward compatibility and is also a USB3 Hub

System View – USB4^{TM} Peripheral Device

- *USB4 Peripheral Device* can optionally support USB3 Tunneling

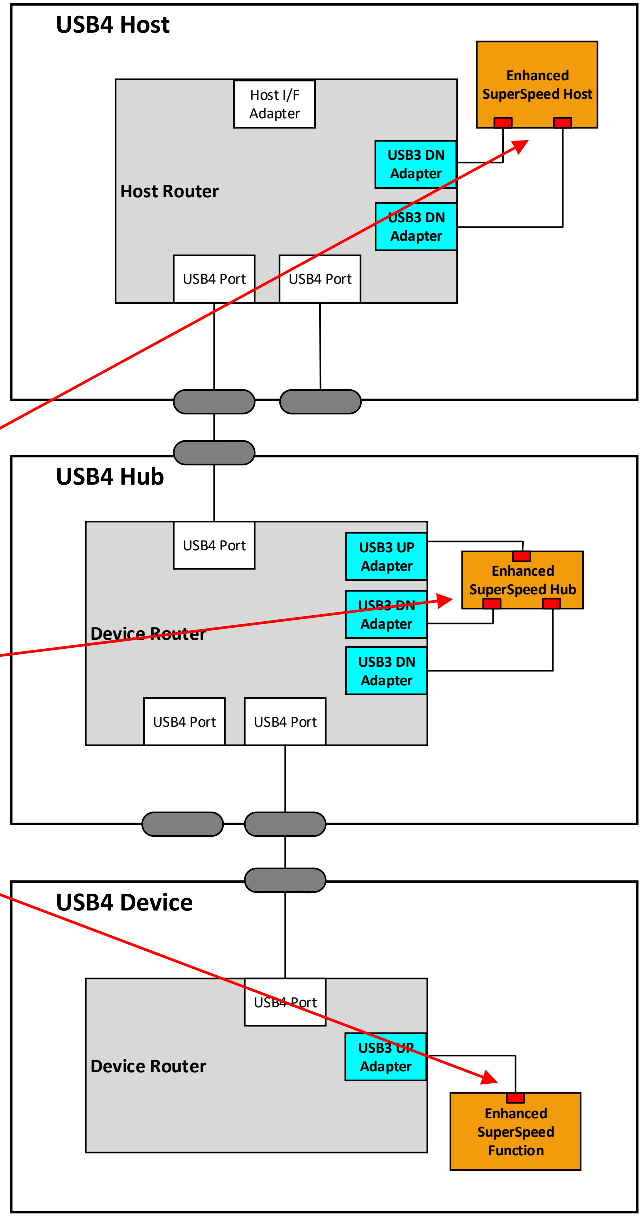
- If the *USB4 Peripheral Device* supports USB3 Tunneling then:
  - *It* implements a SuperSpeed Plus Hub or a USB 3.2 peripheral device
  - *It* maintains backward compatibility and is also a USB3 Hub or a USB 3.2 peripheral device
  - *Device Router* implements a USB3 Upstream Adapter



Internal USB3 Device

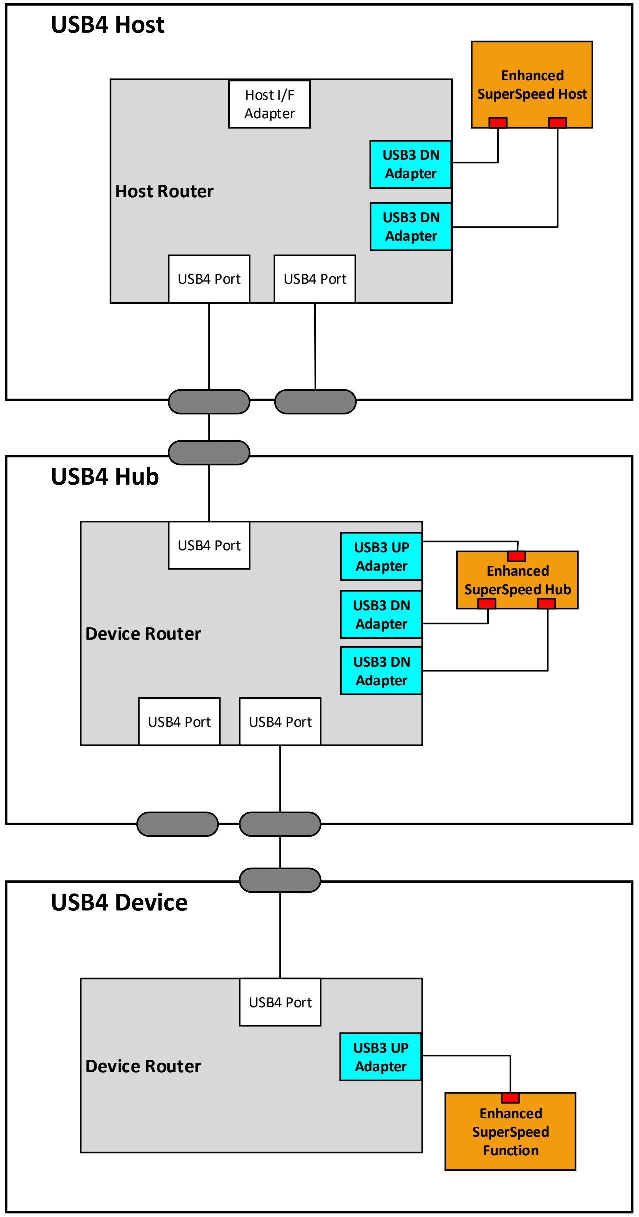
- **Internal USB3 device** refers to either an ***internal USB SuperSpeed Plus hub***, **internal USB peripheral device**, or **internal host controller**

- **Internal USB3 device ports** that interface with a USB3 Adapter differ from the USB 3.2 Spec, mainly at the Physical and Link Layers behavior



Internal USB3 Device

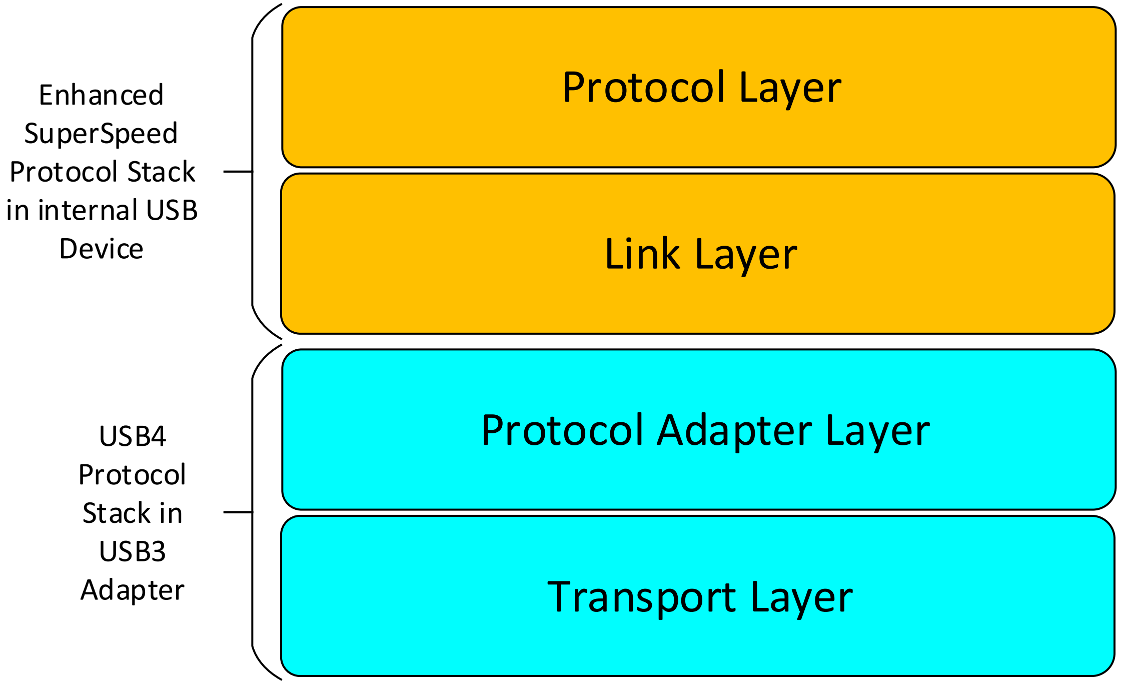
- **Internal USB3 device ports** that interface with a USB3 Adapter differ from the USB 3.2 Spec
  - No ***Physical*** *Layer*
    - No Scrambling
    - No SKIP Ordered-Set
  - ***Link*** *Layer*
    - Must support *Gen 2* Single-Lane (2x1)
    - May support *Gen 2* Dual-Lane (2x2)
    - *Gen 1* not supported
    - U1 not supported
    - tPortConfiguration, PENDING_HP_TIMER and few more timers extended



Protocol Stack

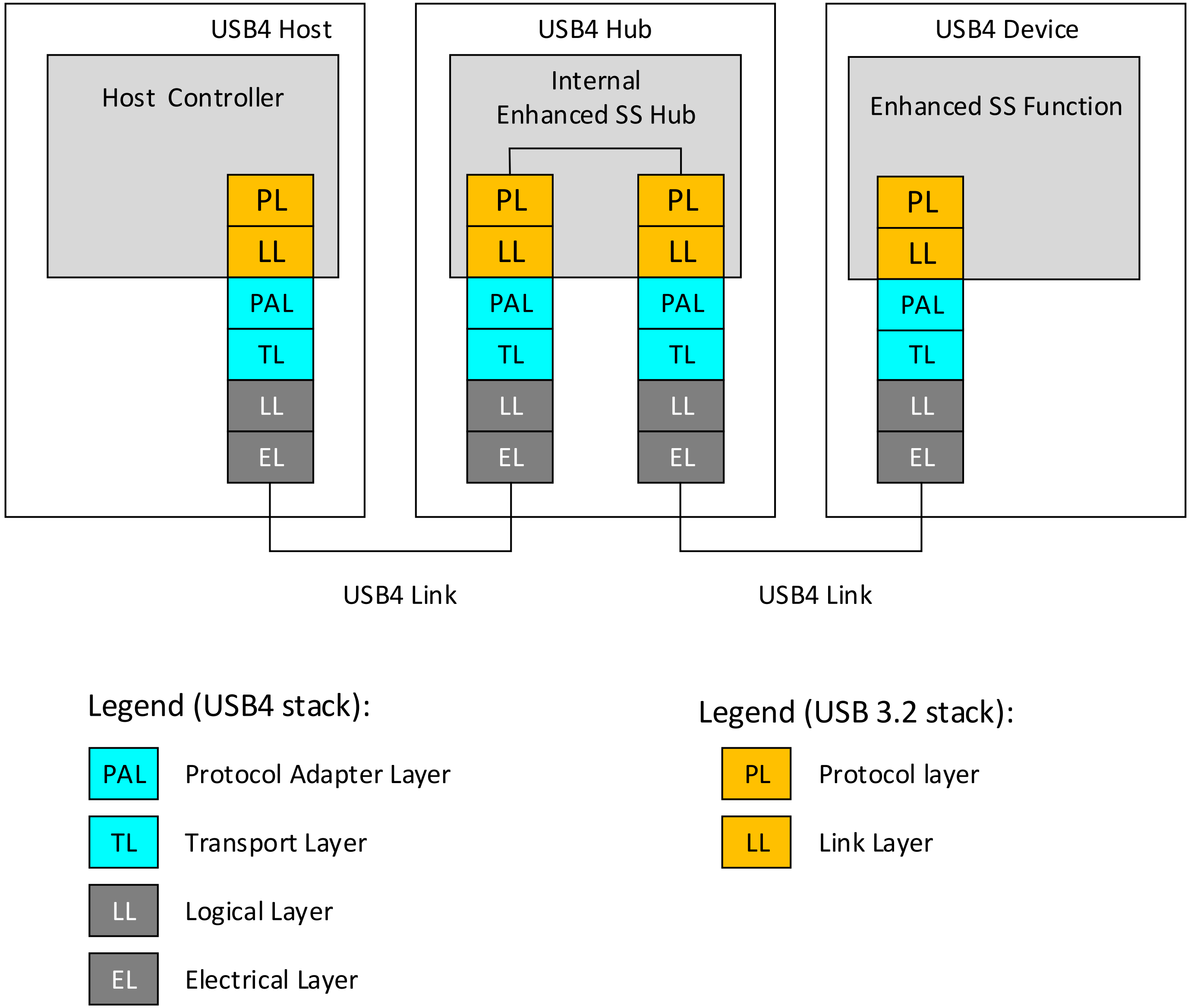
- The Internal USB3 Device interfaces to the USB3 Adapter layer after the Link Layer

- The USB3 Adapter encapsulates the Native USB3 protocol into USB4 Transport Layer Packets


Enhanced SuperSpeed Protocol Stack in internal USB Device
Protocol Layer
Link Layer
USB4 Protocol Stack in USB3 Adapter
Protocol Adapter Layer
Transport Layer

Protocol Stack

- The Internal USB3 Device interfaces to the USB3 Adapter layer after the Link Layer
- The USB3 Adapter encapsulates the Native USB3 protocol into USB4 Transport Layer Packets



Legend (USB4 stack):

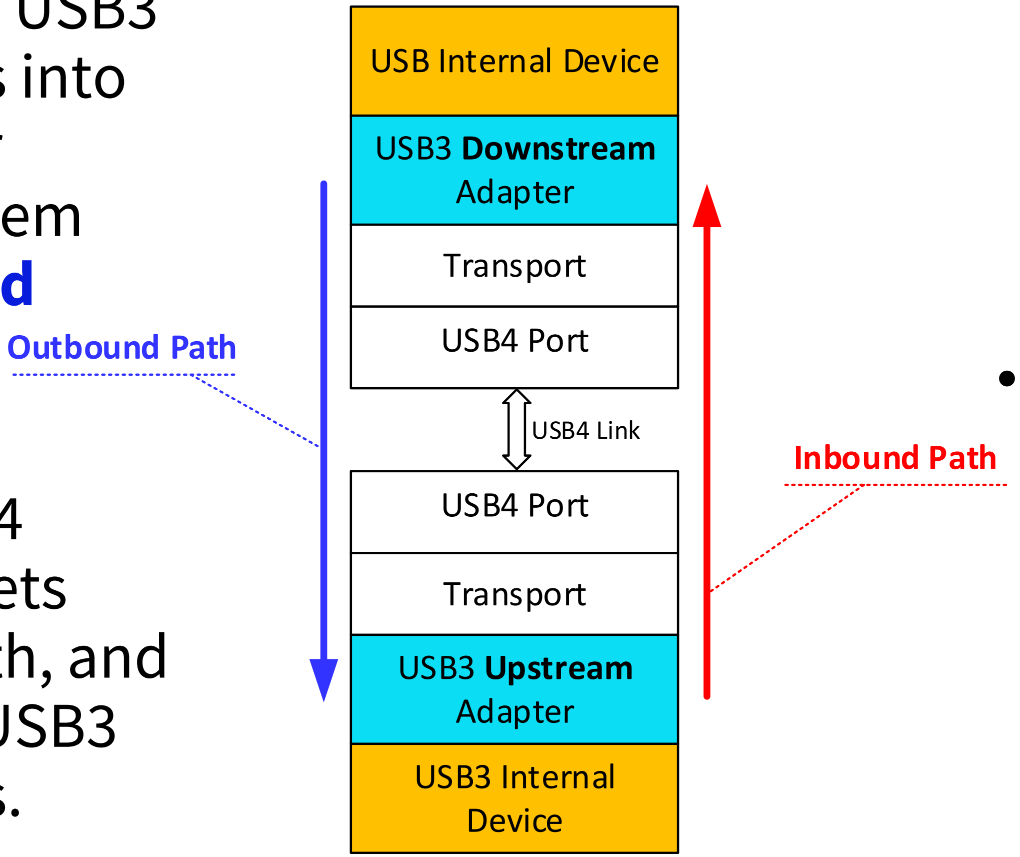
PAL	Protocol Adapter Layer
TL	Transport Layer
LL	Logical Layer
EL	Electrical Layer

Legend (USB 3.2 stack):

PL	Protocol layer
LL	Link Layer

USB3 Adapter - Paths

- A USB3 **Downstream** Adapter encapsulates USB3 events and constructs into USB4 Transport Layer Packets, and sends them through the **Outbound** Path.
- A USB3 **Downstream** Adapter receives USB4 Transport Layer Packets from the **Inbound** Path, and translates them into USB3 events and constructs.



- A USB3 **Upstream** Adapter encapsulates USB3 events and constructs into USB4 Transport Layer Packets, and sends them through the **Inbound** Path.
- A USB3 **Upstream** Adapter receives USB4 Transport Layer Packets from the **Outbound** Path, and translates them into USB3 events and constructs.

USB3 Adapter - Encapsulation

PDF	Type	Payload of Tunneled Packet
0h	LFPS	Indication for an LFPS sequence
1h	Ordered Set	One TS1, TS2, or SDS Ordered Set
2h	Link Command	One USB3 Link Command
3h	LMP / TP / ITP	One LMP, TP, Deferred DPH, or ITP header packet
4h	Start DP Segment	The first segment of a USB3 Data Packet (DPH and DPP)
5h	Middle DP Segment	A segment of a USB3 Data Packet that is not the first segment and is not the last segment
6h	End DP Segment	The last segment of a USB3 Data Packet that is broken into more than one segment

- The PDF defines the construct being tunneled
- Idle Symbols are not tunneled
- The bytes and bits in a Tunneled Packet payload, other than LFPS and Ordered Set, are packed in the same order as the original USB3 construct, including the USB3 framing

USB3 Adapter – Encapsulation – LFPS & OS

- The **LFPS** and **Ordered-Set *(OS)*** count/duration is not maintained over the tunnel
  - Upon new LFPS/OS event from the Internal USB3 device, a USB3 Adapter sends 3 identical LFPS/OS Tunneled Packets
  - A USB3 Adapter receiving LFPS/OS Tunnel Packet will indicate the LFPS/OS reception to the Internal USB3 device until receiving a different Tunneled Packet.

USB3 Adapter – Encapsulation - LFPS

<table>
<tr><td colspan="4">PDF = 0</td><td colspan="1">*</td><td colspan="11">HOP ID</td><td colspan="8">Length = 4</td><td colspan="8">HEC</td></tr>
<tr><td colspan="8">CRC</td><td>R</td><td>LBPM En</td><td>WR</td><td>U3 Wa</td><td>U2 Ex</td><td>SCD2</td><td>SCD1</td><td>RXT En</td><td colspan="8">LBPM</td><td colspan="8">Rsvd</td></tr>
</table>

- **PDF** = 0 ; Length is constant = 4 ; CRC protects the Payload
- **RX Term Enable**
- **SCD1, SCD2, U2 Exit, U3 Wakeup, Warm Reset, LBPM Enable**
- **LBPM**

USB3 Adapter – Encapsulation - LFPS

<table>
<tr><td>PDF = 0</td><td>*</td><td colspan="9">HOP ID</td><td>Length = 4</td><td>HEC</td></tr>
<tr><td colspan="3">CRC</td><td>R</td><td>LBPM En</td><td>WR</td><td>U3 Wa</td><td>U2 Ex</td><td>SCD2</td><td>SCD1</td><td>RXT En</td><td>LBPM</td><td>Rsvd</td></tr>
</table>

- **PDF** = 0 ; Length is constant = 4 ; CRC protects the Payload
- **RX Term Enable**
  - LFPS Tunneled Packet is generated when value changes
  - Represents the current state of the local low-impedance receiver termination in every LFPS Tunneled Packet
- **SCD1**, **SCD2**, **U2 Exit**, **U3 Wakeup**, **Warm Reset**, **LBPM Enable**
- **LBPM**

USB3 Adapter – Encapsulation - LFPS

<table>
<tr><td>PDF = 0</td><td>*</td><td colspan="9">HOP ID</td><td>Length = 4</td><td>HEC</td></tr>
<tr><td colspan="3">CRC</td><td>R</td><td>LBP
M
En</td><td>W
R</td><td>U3
Wa</td><td>U2
Ex</td><td>S
C
D
2</td><td>S
C
D
1</td><td>RX
T
En</td><td>LBPM</td><td>Rsvd</td></tr>
</table>

- **PDF** = 0 ; Length is constant = 4 ; CRC protects the Payload
- **RX Term Enable**
- **SCD1**, **SCD2**, **U2 Exit**, **U3 Wakeup**, **Warm Reset**, **LBPM Enable**
  - LFPS Tunneled Packet is generated when the LFPS indication from the Internal USB3 device changes
  - Only **one** of the bits can be set to 1b
  - When all equal to 0b – It is LFPS stop
- **LBPM**

USB3 Adapter – Encapsulation - LFPS

<table>
<tr><td>PDF = 0</td><td>*</td><td colspan="9">HOP ID</td><td>Length = 4</td><td>HEC</td></tr>
<tr><td colspan="3">CRC</td><td>R</td><td>LBPM En</td><td>WR</td><td>U3 Wa</td><td>U2 Ex</td><td>SCD2</td><td>SCD1</td><td>RXT En</td><td>LBPM</td><td>Rsvd</td></tr>
</table>

- **PDF** = 0 ; Length is constant = 4 ; CRC protects the Payload
- **RX Term Enable**
- **SCD1**, **SCD2**, **U2 Exit**, **U3 Wakeup**, **Warm Reset**, **LBPM Enable**
- **LBPM**
  - LFPS Tunneled Packet is generated when a new LBPM PHY Capability byte is sent the Internal USB3 device
  - **LBPM Enable** is set to 1b

USB3 Adapter – Encapsulation – Ordered Set

- **TS1**, **TS2**, **SDS**
  - Ordered Set Tunneled Packet is generated when the Ordered Set indication from the Internal USB3 device changes
  - One bit and only one is set to 1b
  - **Link Functionality** contains the value received from the Internal USB3 device
- No other Ordered Set is sent over the Tunnel
- Only Ordered Set for Lane 0 are sent

<table>
<tr><td>PDF = 1</td><td>*</td><td colspan="2">HOP ID</td><td>Length = 4</td><td colspan="4">HEC</td></tr>
<tr><td colspan="3">CRC</td><td>Rsvd</td><td>Link Functionality</td><td>Rsvd</td><td>S
D
S</td><td>T
S
2</td><td>T
S
1</td></tr>
</table>

USB3 Adapter – Encapsulation

- **Link Commands (PDF = 2) ; LMP, TP, ITP (PDF = 3)**
  - Each is encapsulated into a single separate Tunneled Packet
  - Packed in the same order as the original USB3 construct, including the USB3 framing symbols.
  - The **first** byte (i.e. the least-significant byte of the encapsulated construct) is mapped to **B0** in the Tunneled Packet payload

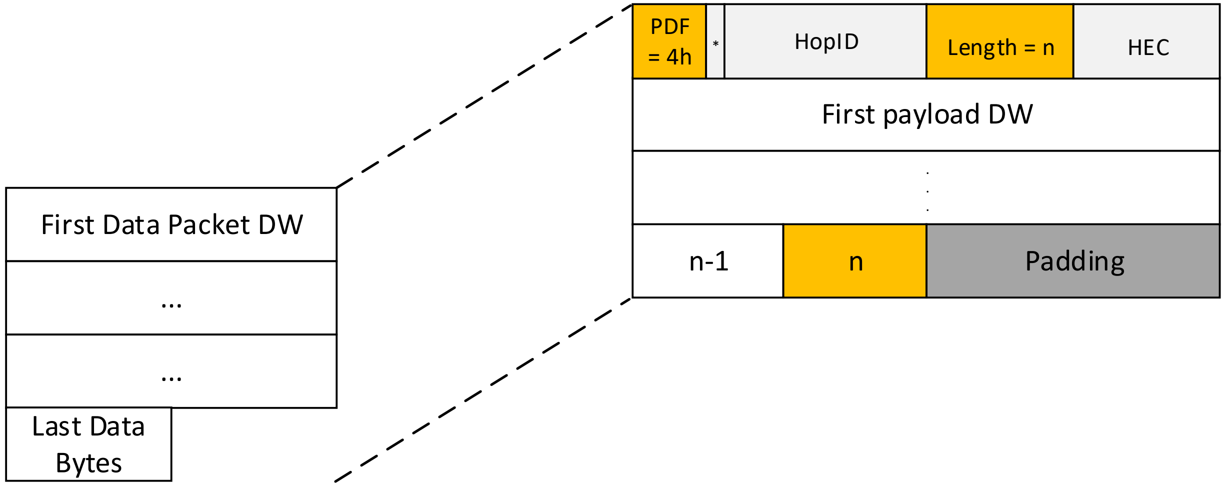
Link Command Example

<table>
<tr><td>PDF = 2</td><td>*</td><td colspan="2">HOP ID</td><td>Length = 8</td><td>HEC</td></tr>
<tr><td colspan="3">SLC</td><td colspan="2">SLC</td><td>SLC</td><td>EPF</td></tr>
<tr><td colspan="3">Link Command Word
Byte 0</td><td colspan="2">Link Command Word
Byte 1</td><td>Link Command Word
Byte 0</td><td>Link Command Word
Byte 1</td></tr>
</table>

USB3 Adapter – Encapsulation – Data Packets

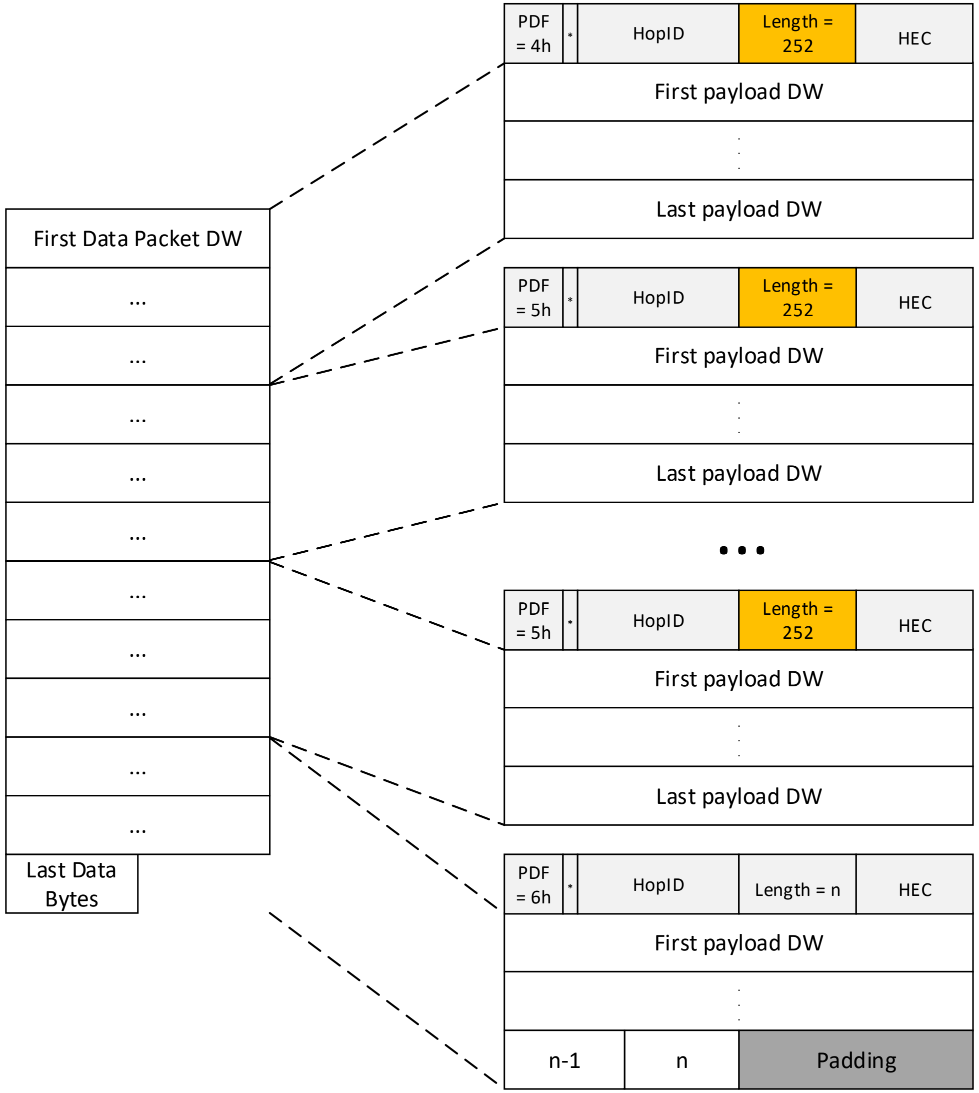
- A USB3 Data Packet and its USB3 framing symbols are segmented into one or more Tunneled Packets

- For a USB3 Data Packet with 252 bytes or less, a single Tunneled Packet of type **Start DP Segment** is sent



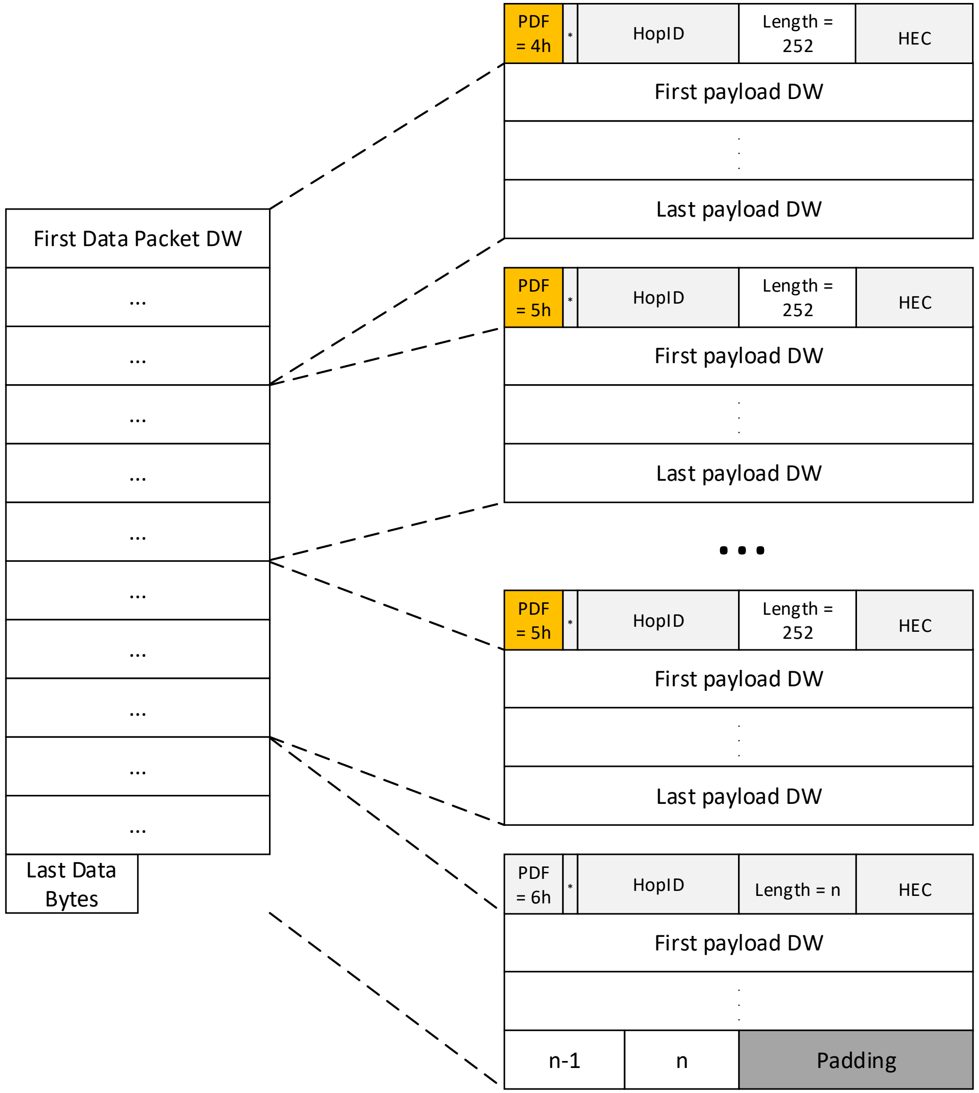
USB3 Adapter – Encapsulation – Data Packets

- A USB3 Data Packet with more than 252 bytes is segmented into multiple Tunneled Packets
- All Tunneled Packets except the last packet includes 252 bytes of payload



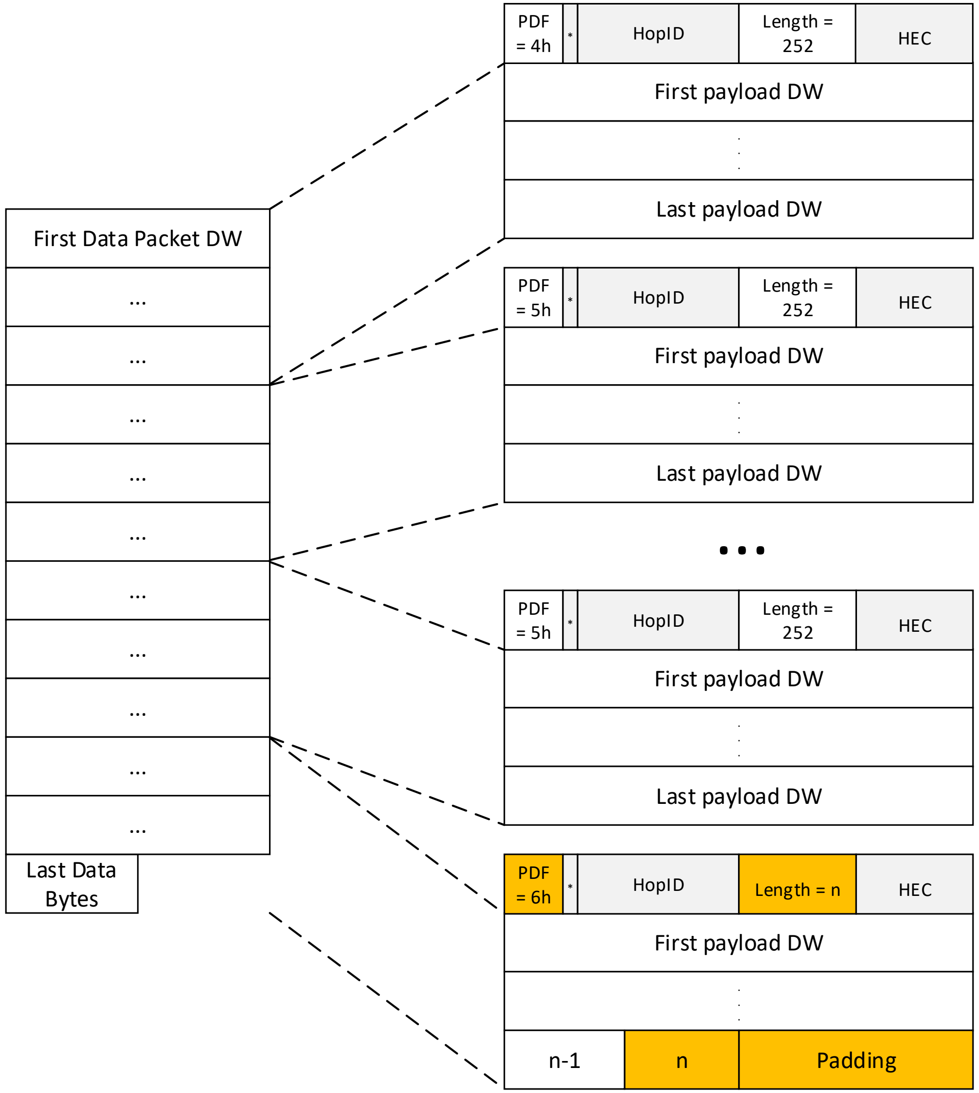
USB3 Adapter – Encapsulation – Data Packets

- The first Tunneled Packet is of type **Start DP Segment** (*PDF* = 4)
- Any following Tunneled Packets other than the last Tunneled Packet is of type **Middle DP Segment** (*PDF* = 5)



USB3 Adapter – Encapsulation – Data Packets

- The last Tunneled Packet is of type **End DP Segment** (*PDF* = 6)
- Padding is added to be DW aligned



Time for Q&A

Agenda

- Configuration Layer
- USB3 Tunneling
- DP Tunneling
- PCIe Tunneling

DP Tunneling Agenda

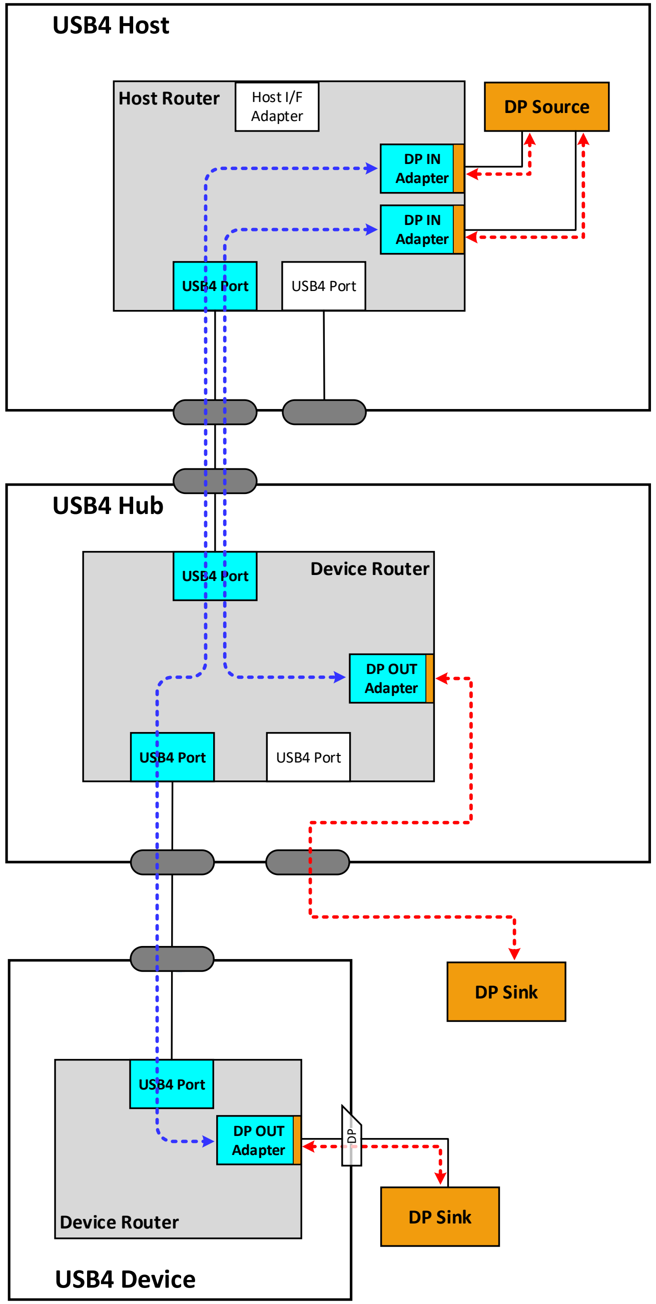
- System View
- Protocol Stack
- Paths and Packet Types
- AUX Handling
- Link Training
- Main-Link Tunneling
  - Basic Concepts
  - SST
  - MST

System View

- Native DisplayPort™ is Tunneled over USB4 Fabric

- Originates and consumed as Native DP protocol
- From DP Source perspective, the USB4 Fabric and the Adapters are either totally transparent or act as an LTTPR

- DP Adapters are the translators within each Router that allow DP protocol to travel back and forth from Native to Tunneled

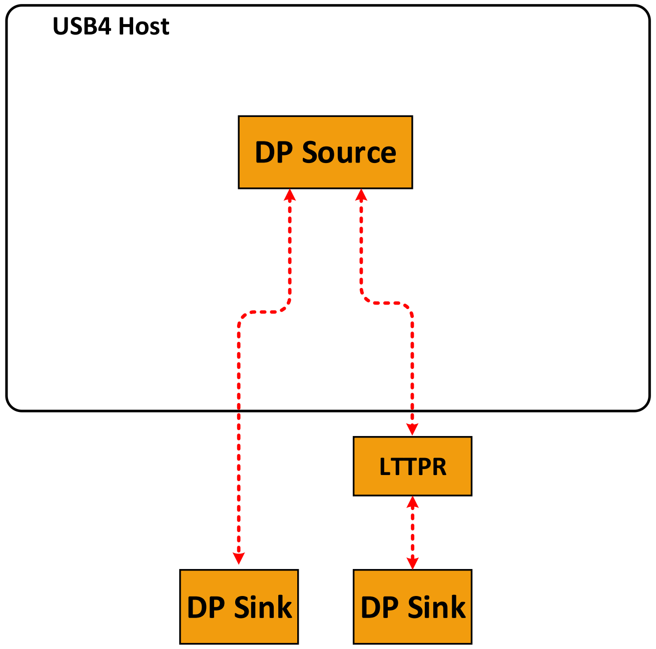


System View

- Native DisplayPort™ is Tunneled over USB4 Fabric

- Originates and consumed as Native DP protocol
- From DP Source perspective, the USB4 Fabric and the Adapters are either totally transparent or act as an LTTPR

- DP Adapters are the translators within each Router that allow DP protocol to travel back and forth from Native to Tunneled

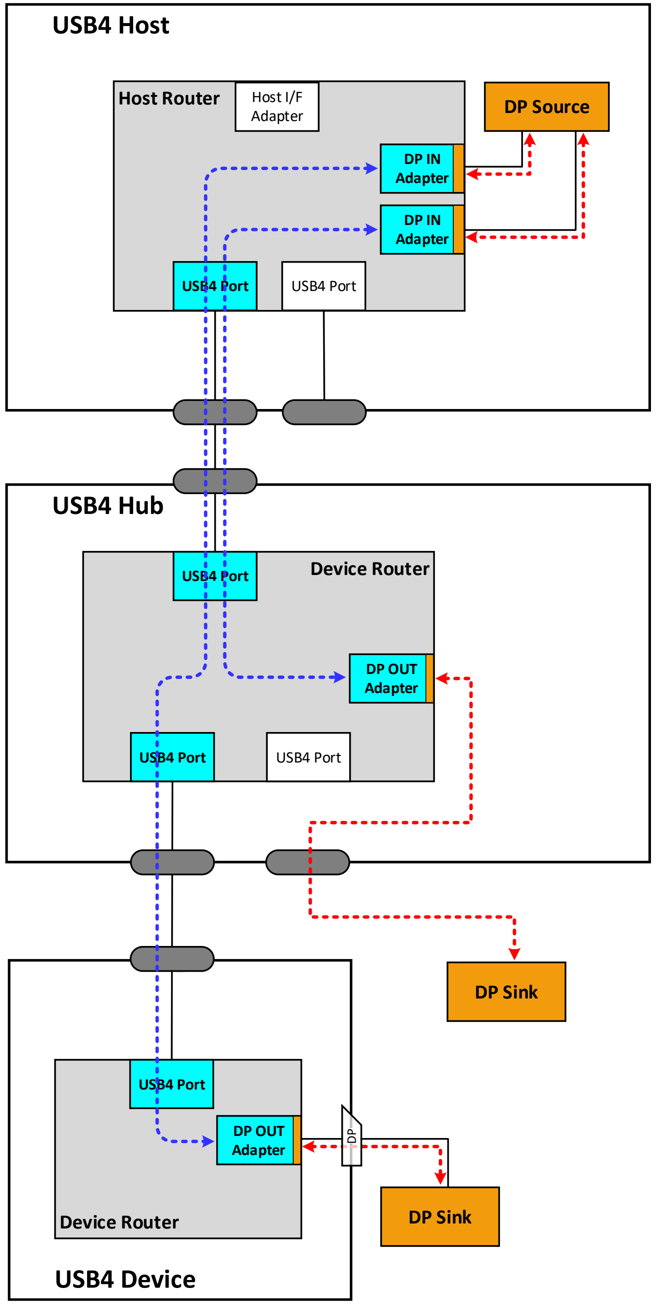


System View

- Native DisplayPort™ is Tunneled over USB4 Fabric

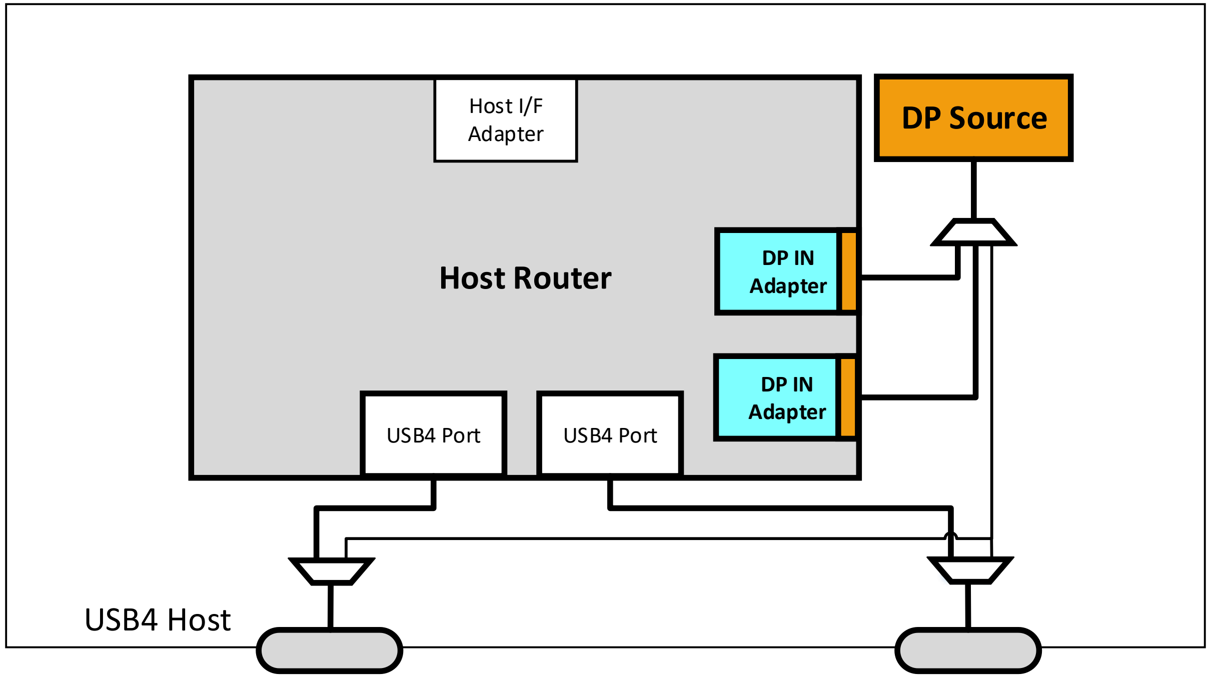
- Originates and consumed as Native DP protocol
- From DP Source perspective, the USB4 Fabric and the Adapters are either totally transparent or act as an LTTPR

- **DP Adapters** are the translators within each Router that allow DP protocol to travel back and forth from Native to Tunneled

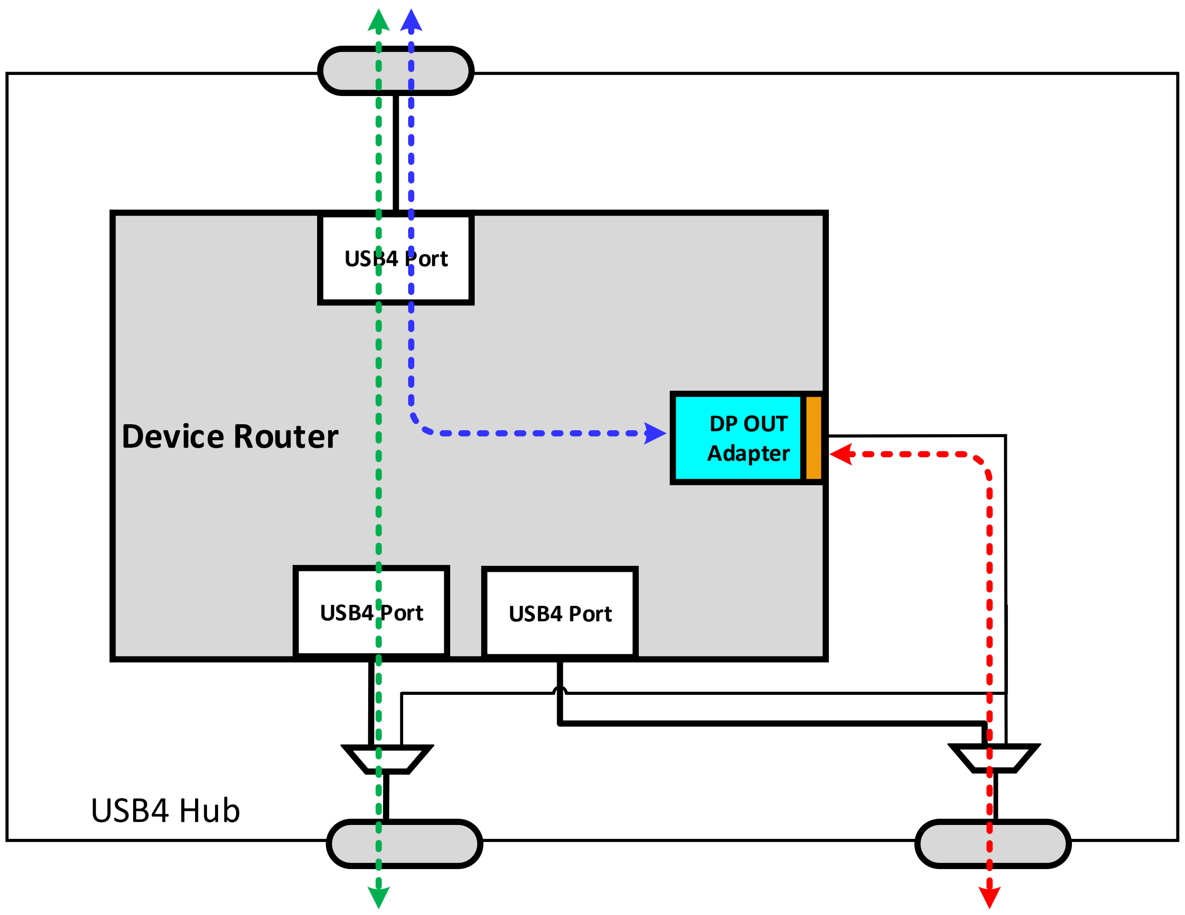


System View – USB4^{TM} Host

- *USB4 Host* must support DP Tunneling
- *Host Router* has at least one DP IN Adapter
  - It can additionally support multiple DP IN and DP OUT Adapters
- *USB4 Host* must support *DP Alternate Mode*



System View – USB4^{TM} Hub

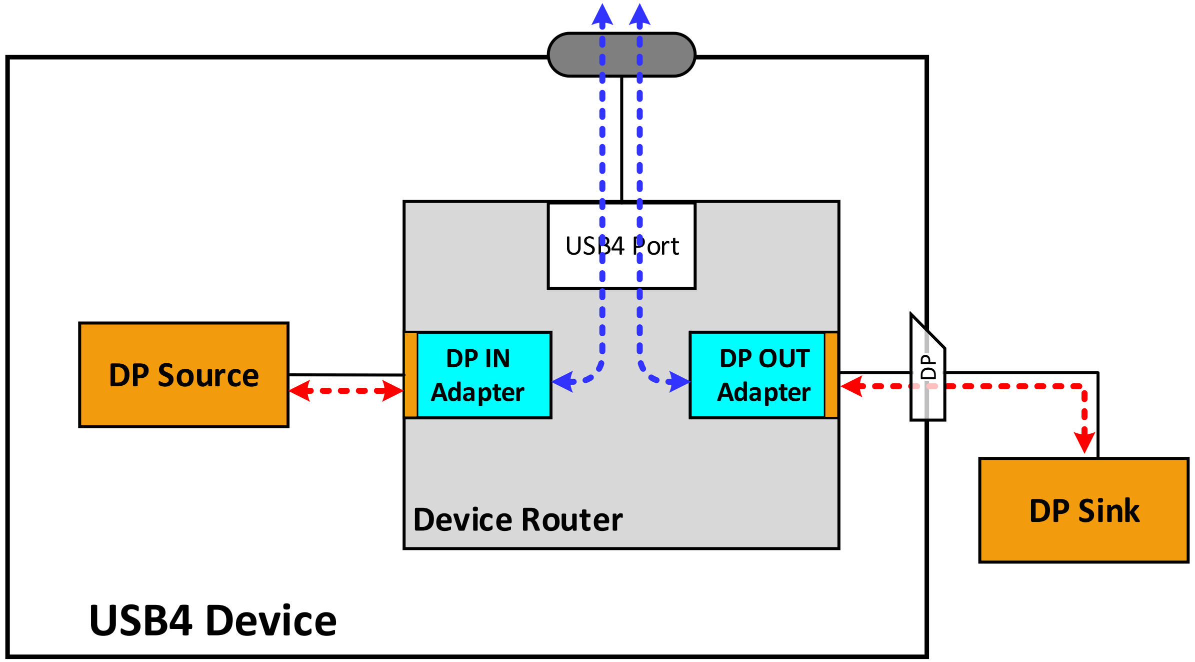


- Two ways that a *USB4 Hub* must support DP Tunneling:
  - Pass Through – DP Traffic routed directly between UFP and DFP
  - DP OUT – *Device Router* has at least one DP OUT Adapter
- *Device Router* can additionally support one or more DP OUT and/or DP IN Adapters
- *USB4 Hub* must support *DP Alternate Mode* on its DFP

System View – USB4^{TM} Peripheral Device

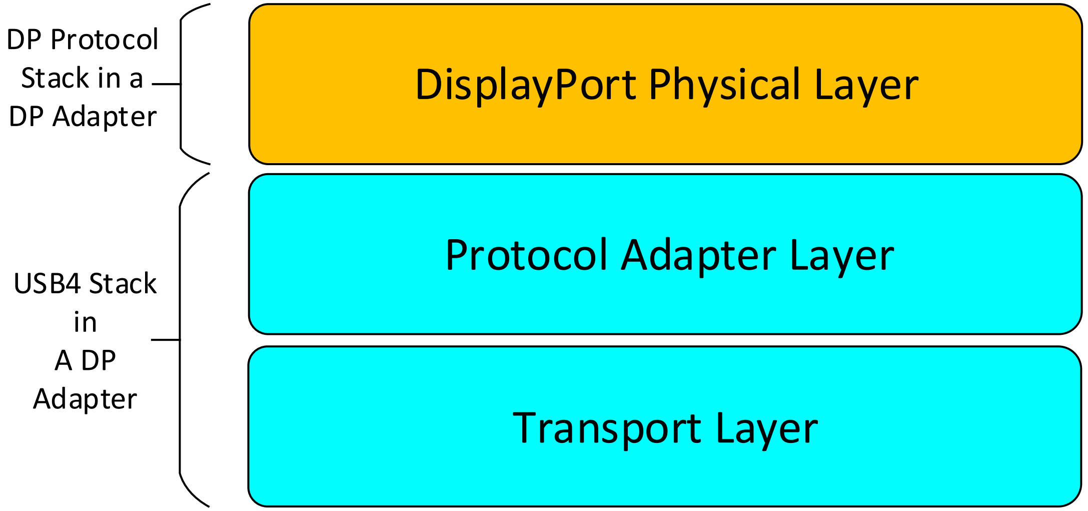
- *USB4 Peripheral Device* can optionally support DP Tunneling

- *Device Router* can support one or more DP IN and/or DP OUT Adapters



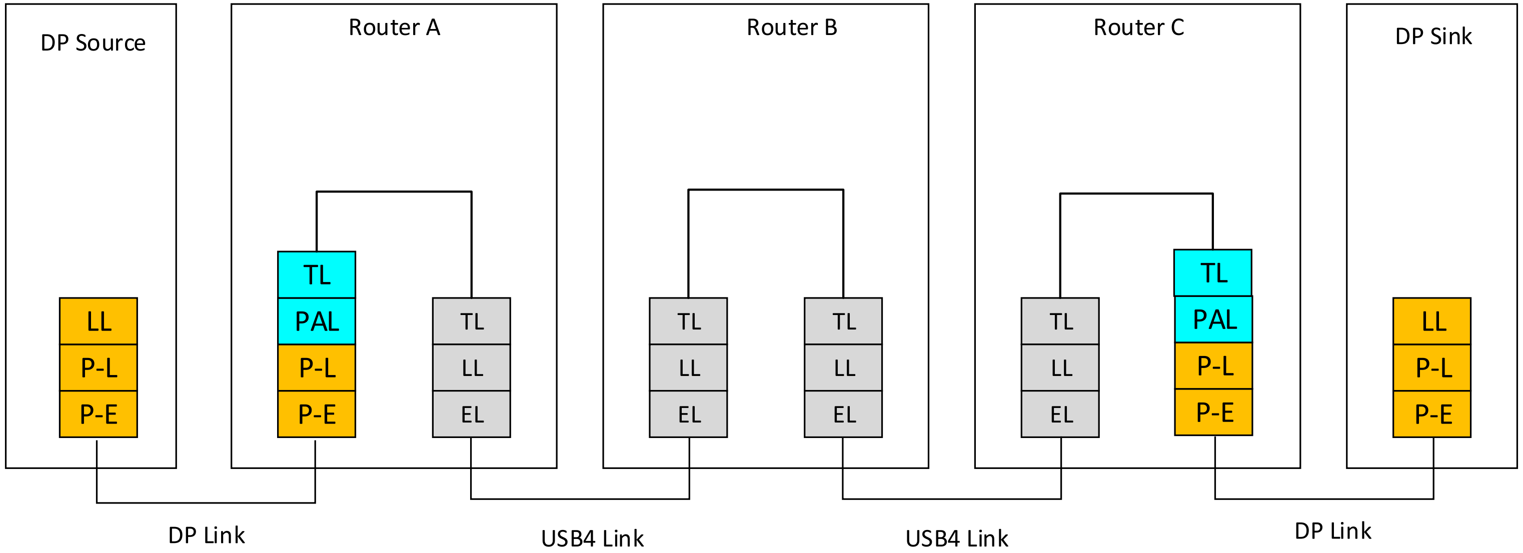
DP Adapter Protocol Stack

- A DP Adapter connects on one side to a Native DP PHY and on the other to the Transport Layer.
- A DP Adapter implements only the Physical Layer of the DP Protocol Stack
- A DP Adapter encapsulates the Native DP protocol into USB4 Tunneled Packets


DP Protocol Stack in a DP Adapter
USB4 Stack in A DP Adapter
DisplayPort Physical Layer
Protocol Adapter Layer
Transport Layer

DP Adapter Protocol Stack

- A DP Adapter connects on one side to a Native DP PHY and on the other to the Transport Layer.
- A DP Adapter implements only the Physical Layer of the DP Protocol Stack
- A DP Adapter encapsulates the Native DP protocol into USB4 Tunneled Packets



Legend (USB4 stack):

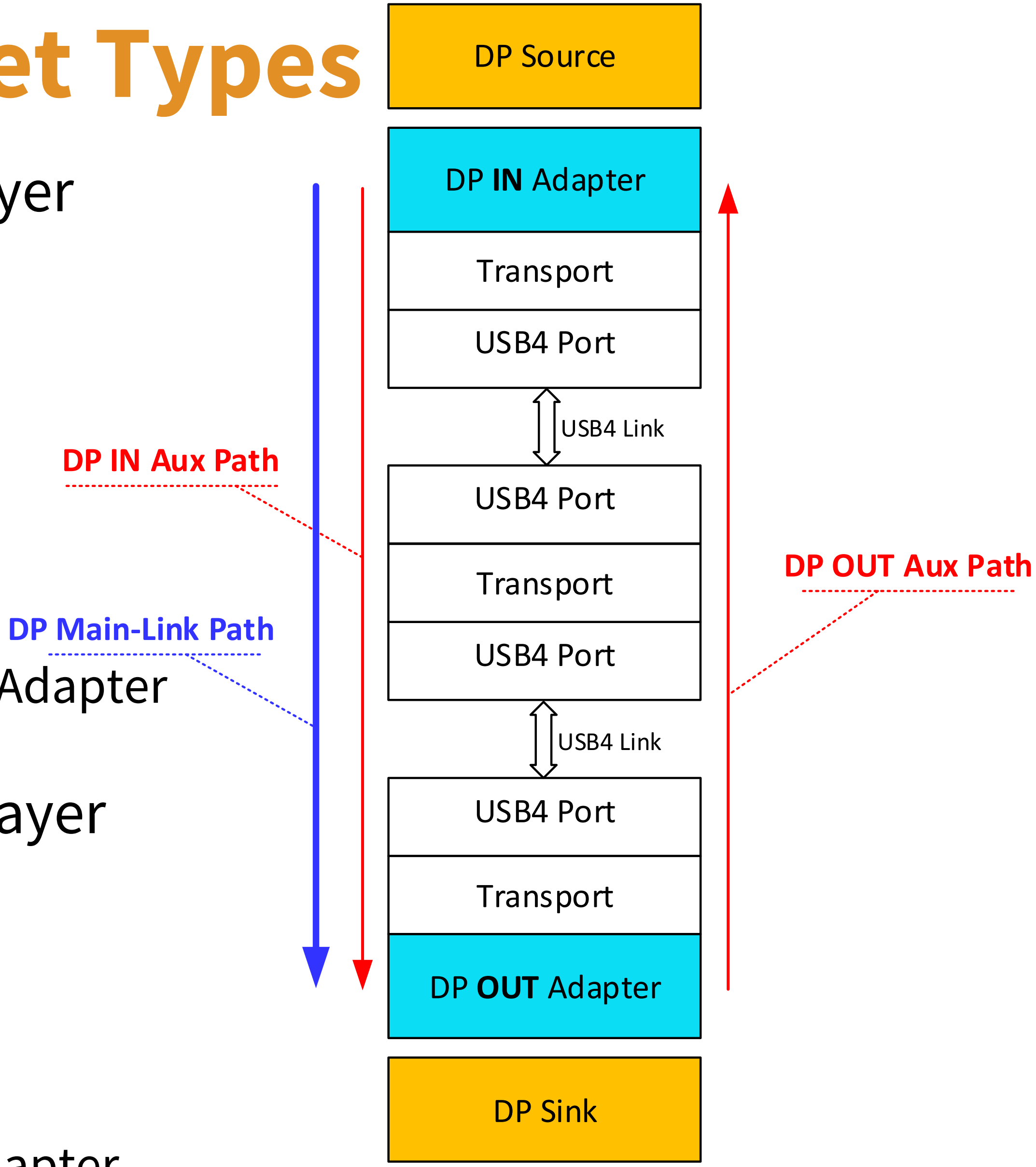
PAL	Protocol Adapter Layer
TL	Transport Layer
LL	Logical Layer
EL	Electrical Layer

Legend (DisplayPort stack):

LL	Link Layer
P-L	PHY Logical Sub-Block
P-E	PHY Electrical Sub-Block

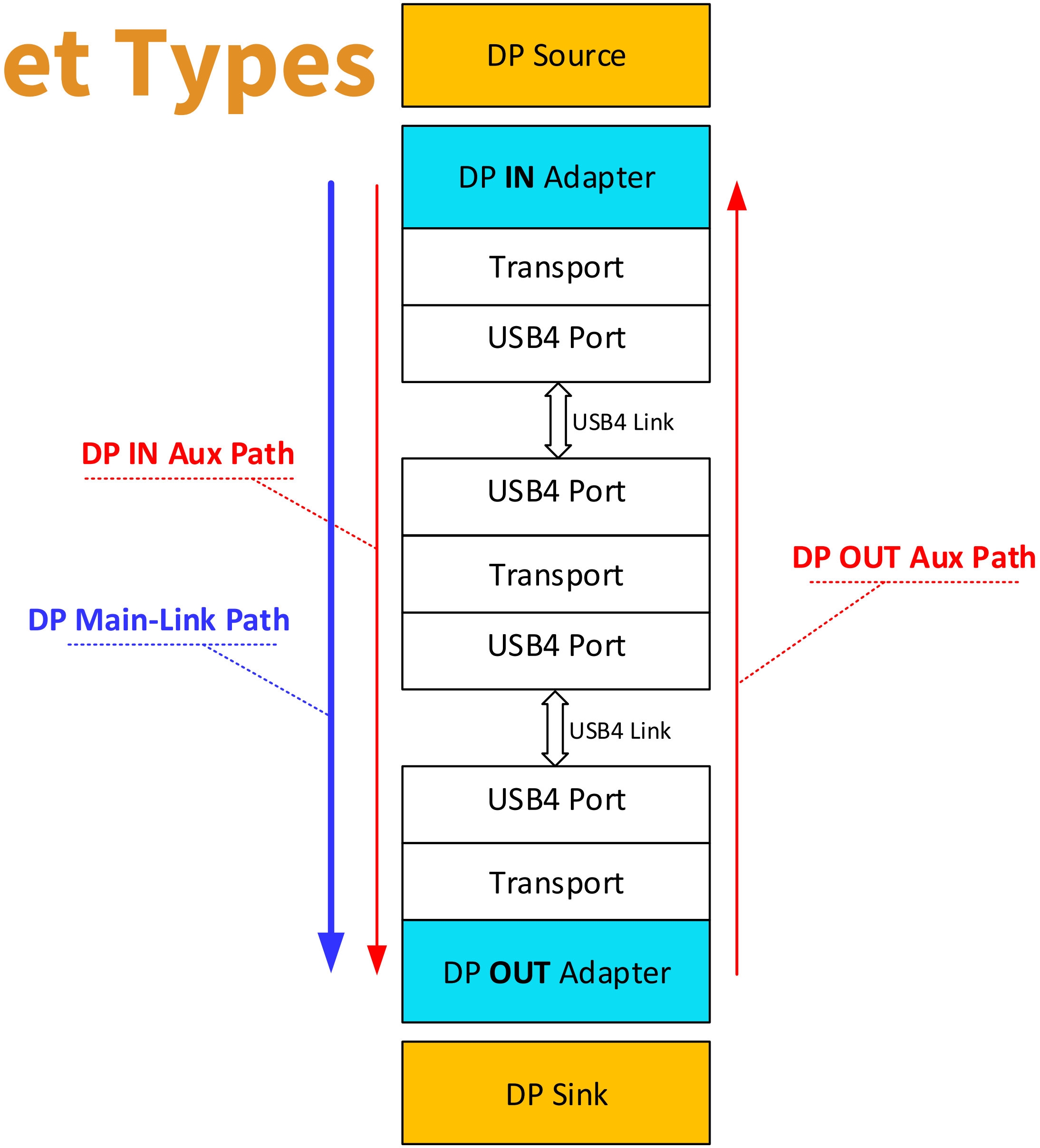
DP Adapter – Paths & Packet Types

- A **DP IN** Adapter uses 2 Paths to send Transport layer Packets to the DP OUT Adapter
  - DP Main-Link Path encapsulates
    - DP High Speed Constructs
    - DP Clock Sync
  - DP IN Aux Path encapsulates
    - AUX Request from DP Source
    - Configuration and Status messages to the **DP OUT** Adapter
- A **DP OUT** Adapter uses 1 Path to send Transport layer Packets to the DP IN Adapter
  - DP OUT Aux Path encapsulates
    - AUX Responses from DP Sink
    - HPD & IRQ from Sink
    - Configuration and Status messages to the **DP IN** Adapter



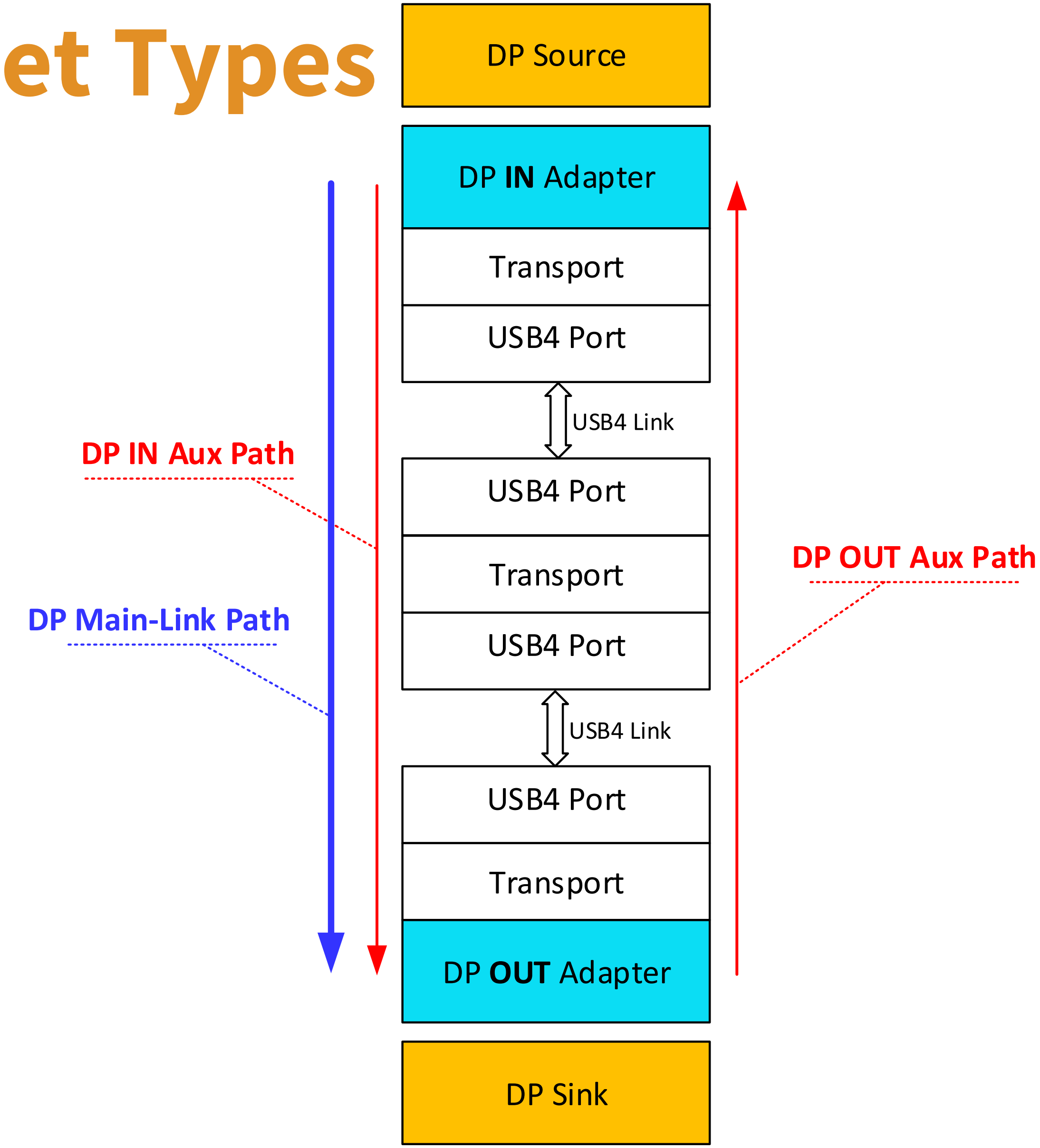
DP Adapter – Paths & Packet Types

PDF	Type	Usage
0	AUX	AUX Request and Response
1	HPD	Send HPD status from DP OUT to DP IN
2	SET_CONFIG	Configuration and Status between the DP Adapters
3	ACK	Acknowledge Packet for HPD and SET_CONFIG reception



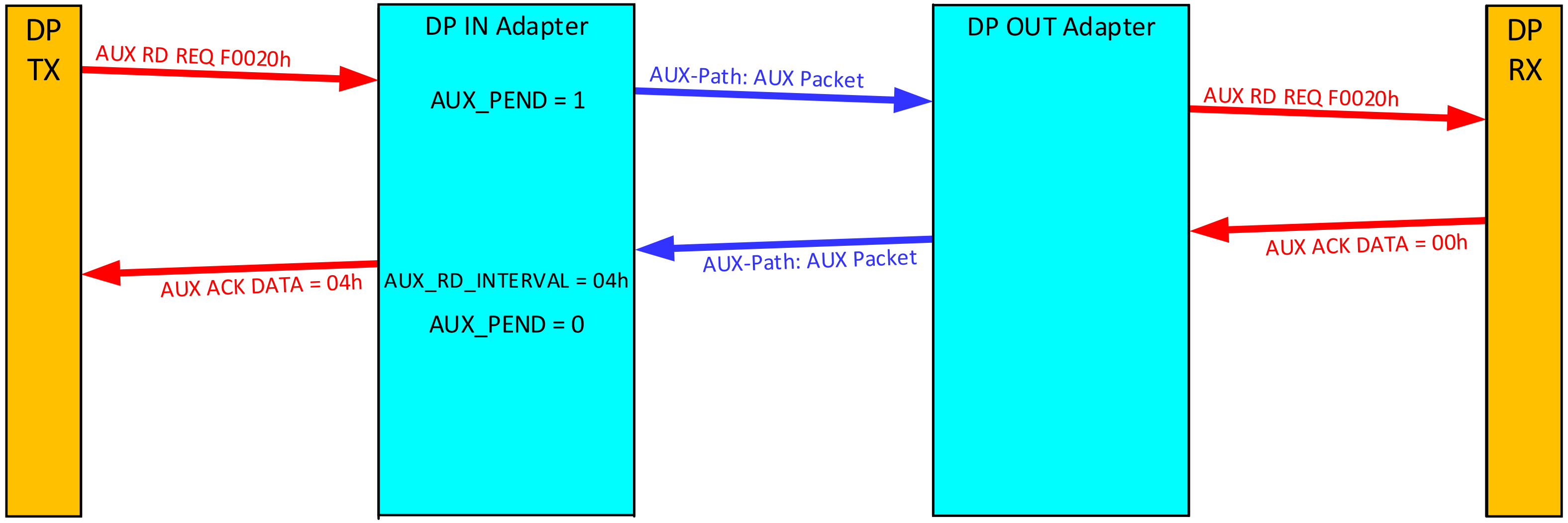
DP Adapter – Paths & Packet Types

PDF	Type	Usage
1	**SST Video**	Active Video data
2	**SST Blank Start**	Blank Start and Scrambler Reset
3	**SST MSA**	Main Stream Attribute
4	**SST Secondary**	Secondary
5	**DP Clock Sync**	DP TMU clock synchronization
6	**MST**	Multi Stream
7	**FEC**	FEC Enable and FEC Disable



AUX Handling – LTTPR Mode

- DP Adapters act as one LTTPR, where the **DP IN Adapter** is the UFP and the **DP OUT Adapter** is DFP
- **DP IN Adapter** maps the AUX Transactions to *Target*, *Snoop*, and *Pass Through*
- *Target* - AUX transactions which target the LTTPR

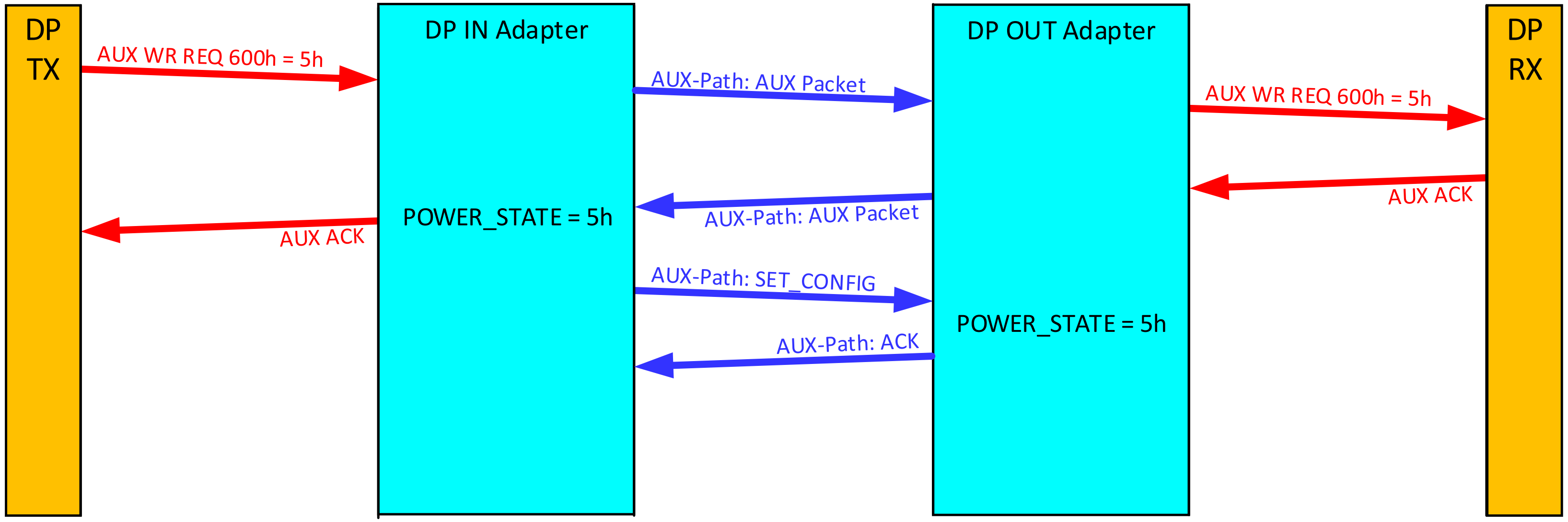


AUX Handling – LTTPR Mode

- DP Adapters act as one LTTPR, where the **DP IN Adapter** is the UFP and the **DP OUT Adapter** is DFP
- **DP IN Adapter** maps the AUX Transactions to *Target*, *Snoop*, and *Pass Through*

- *Snoop* - AUX transactions which hold valuable information for the LTTPR

- **DP IN Adapter** will snoop the data of the Request or the Response and may update the **DP OUT Adapter** with the snooped data

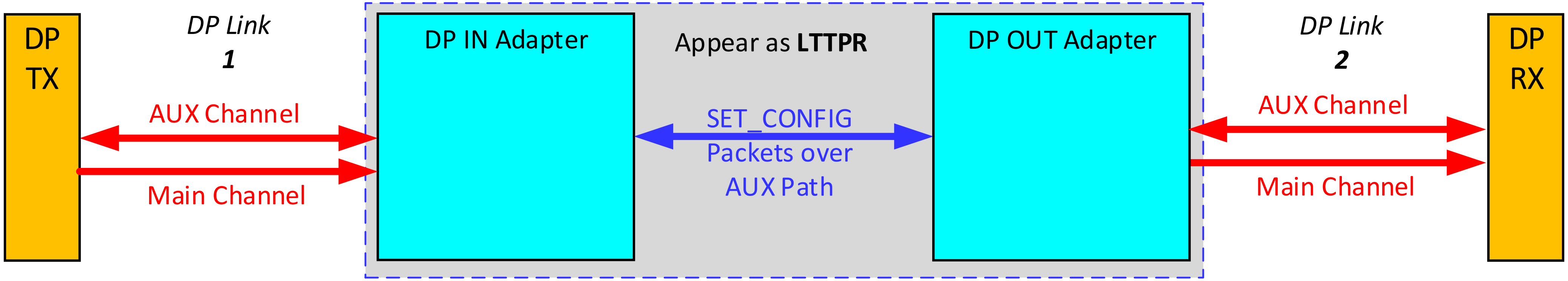


AUX Handling

- **Non LTTPR** is similar to LTTPR Mode, except:
  - AUX Transactions related to Link Training are terminated and responded by the **DP IN Adapter**
  - A **DP IN Adapter** maintains the AUX Timeout towards the DP Source and may issue DEFER Transaction if response does not arrive on time from DP Sink
- A **DP IN Adapter** aggregates the capabilities of the DP Sink and the DP Adapters and reflects the lowest common option to the DP Source
  - DPCD Revision, Link Rate, Lane Count, MST, FEC, DSC, SDP Split

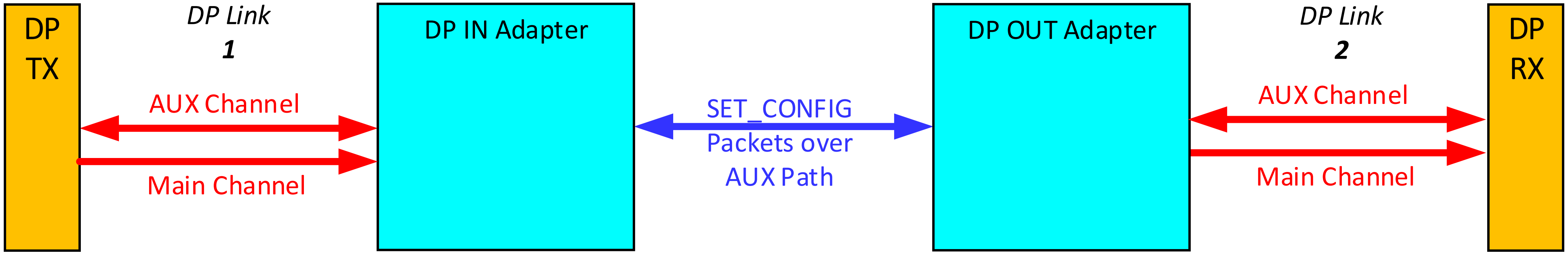
Link Training - LTTPR

- DP Source trains **both** of the DP Links according to DP Spec, where both DP Adapters appear as a single LTTPR
  - The **DP IN Adapter** keeps the **DP OUT Adapter** updated on the Link Training progress using SET_CONFIG Packets.
  - DP Training Patterns are not tunneled over the Main-Link Path. SET_CONFIG Packets sent from **DP IN Adapter** to **DP OUT Adapter**, reflecting the current received Training Pattern



Link Training – Non LTTPR

- DP Source trains a single DP Link (*DP Link **1***), and is unaware of *DP Link **2***
- **DP OUT Adapter** trains *DP Link **2*** in parallel
- **DP IN Adapter** aggregates the link training status of both Adapters
  - DP Source can react to link training failures and activate fallback
  - When DP Source is done with Link Training, both DP Links are trained



Main-Link Tunneling – Base Concepts

- MST and SST share the following base concepts
  - All Main-Link Symbols generated by the **DP OUT Adapter** are identical to the Symbols received by the **DP IN Adapter**
  - The total number of Main-Link Symbol clock cycles, over time, is identical on both DP Links
    - The **DP OUT Adapter** corrects any drift with respect to the **DP IN Adapter** recovered clock
  - Main-Link Symbols are 8b/10 ANSI decoded and de-scrambled by the **DP IN Adapter** and packed into Tunneled Packets as 8-bit data characters

Main-Link Tunneling – Base Concepts

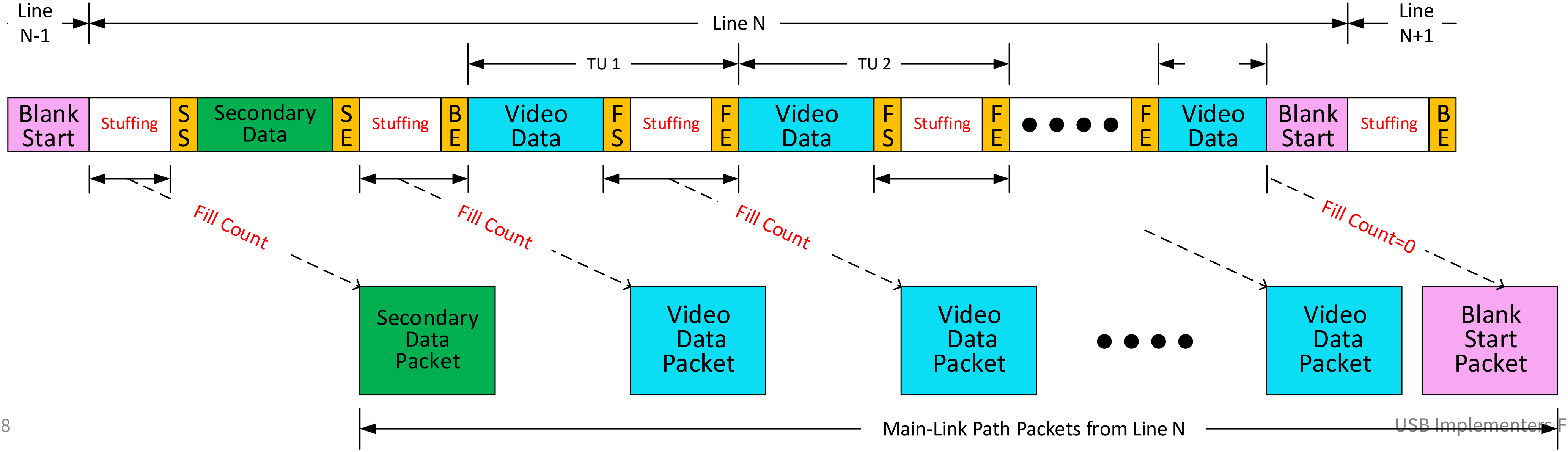
- MST and SST share the following base concepts
  - Main-Link Stuffing Symbols are discarded by the **DP IN Adapter** and reconstructed by the **DP OUT Adapter**
  - FEC RS parity symbols are not packed into Tunneled Packets. A **DP IN Adapter** performs FEC Decoding while a **DP OUT Adapter** performs FEC Encoding
  - HDCP is supported. Encryption and decryption are not performed
  - In order to reconstruct DP Main-Link traffic without any interruptions, a **DP OUT Adapter** implements a buffer that compensates for the jitter in the latency of the received Tunneled Packets.

Main-Link Tunneling – SST

- The continuous Main-Link data stream is encapsulated into Tunneled Packets

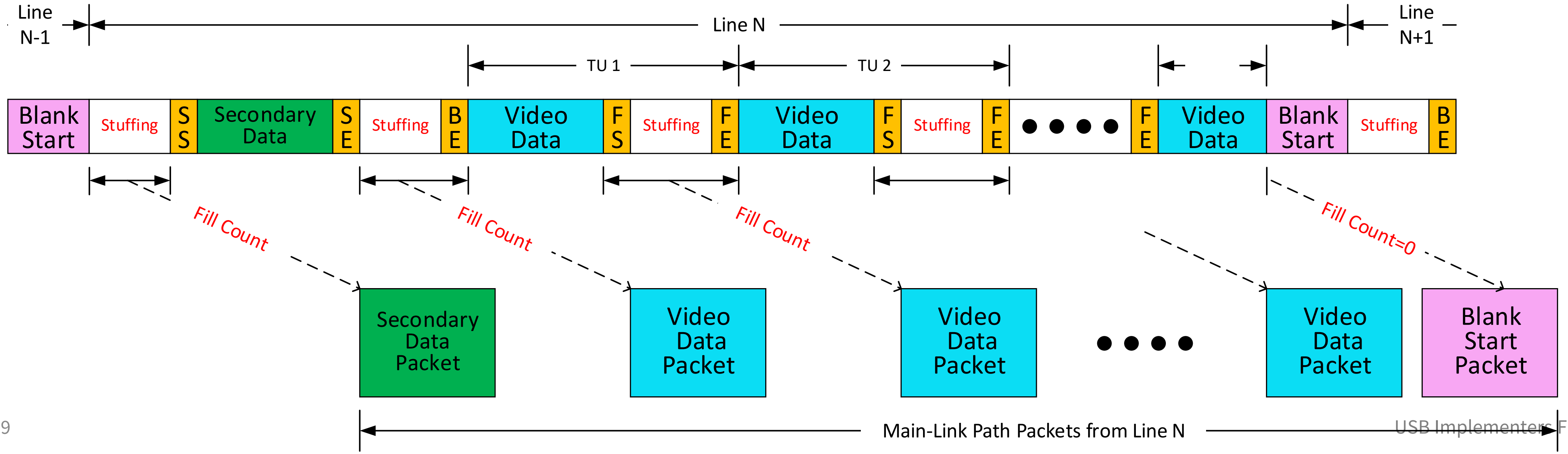
- *Control Link Symbols* are discarded by the **DP IN Adapter** and reconstruct by the **DP OUT Adapter** based on the SST Tunnel Packet type

  - *Example: SS & SE Symbols will be discarded when Secondary Data is encapsulated into a Secondary Tunneled Packet*



Main-Link Tunneling – SST

- *Fill Count* field exists in every SST Tunneled Packet to inform **DP OUT Adapter** on the number of discarded Stuffing Symbols
  - DP IN Adapter discards Dummy Symbols during horizontal and vertical Blanking and Stuffing Symbols during Active Video



Main-Link Tunneling – MST

- The Native DP MST Link is built out of continually transported Multi Stream Transport Packets (*MTP*)

- The *Native DP MTP*s are broken up by the **DP IN Adapter** into *Sub-MTP Transfers Units (TU)* before being encapsulated in MST Tunneled Packets

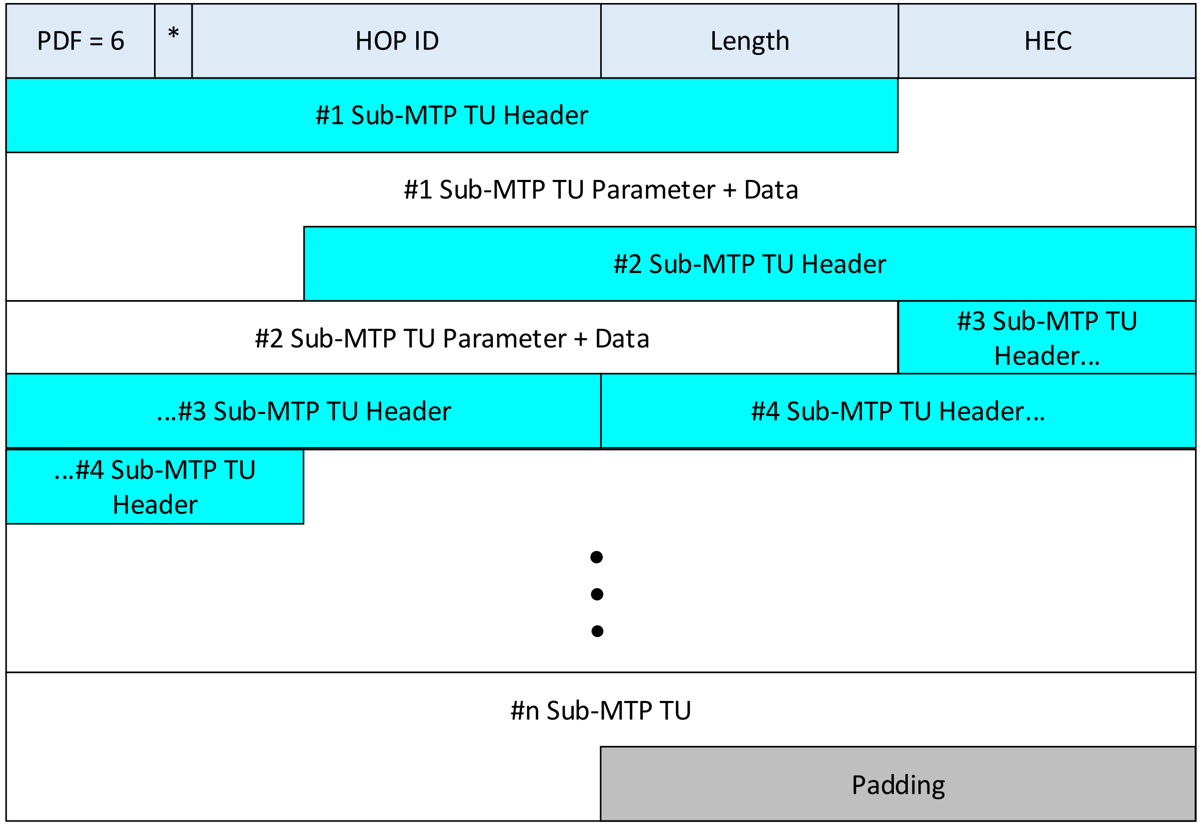
- A **DP OUT Adapter** recreates *Native DP MTP*s out of the *Sub-MTP TUs*

- A **DP IN Adapter** performs encapsulation regardless of Virtual Channels boundaries

Main-Link Tunneling – MST

- An MST Tunneled Packet starts with a *Sub-MTP TU header*

- An MST Tunneled Packet may include up to 17 *Sub-MTP TUs*



PDF = 6 | * | HOP ID | Length | HEC

#1 Sub-MTP TU Header

#1 Sub-MTP TU Parameter + Data

#2 Sub-MTP TU Header

#2 Sub-MTP TU Parameter + Data

#3 Sub-MTP TU Header...

...#3 Sub-MTP TU Header

#4 Sub-MTP TU Header...

...#4 Sub-MTP TU Header

•
•
•

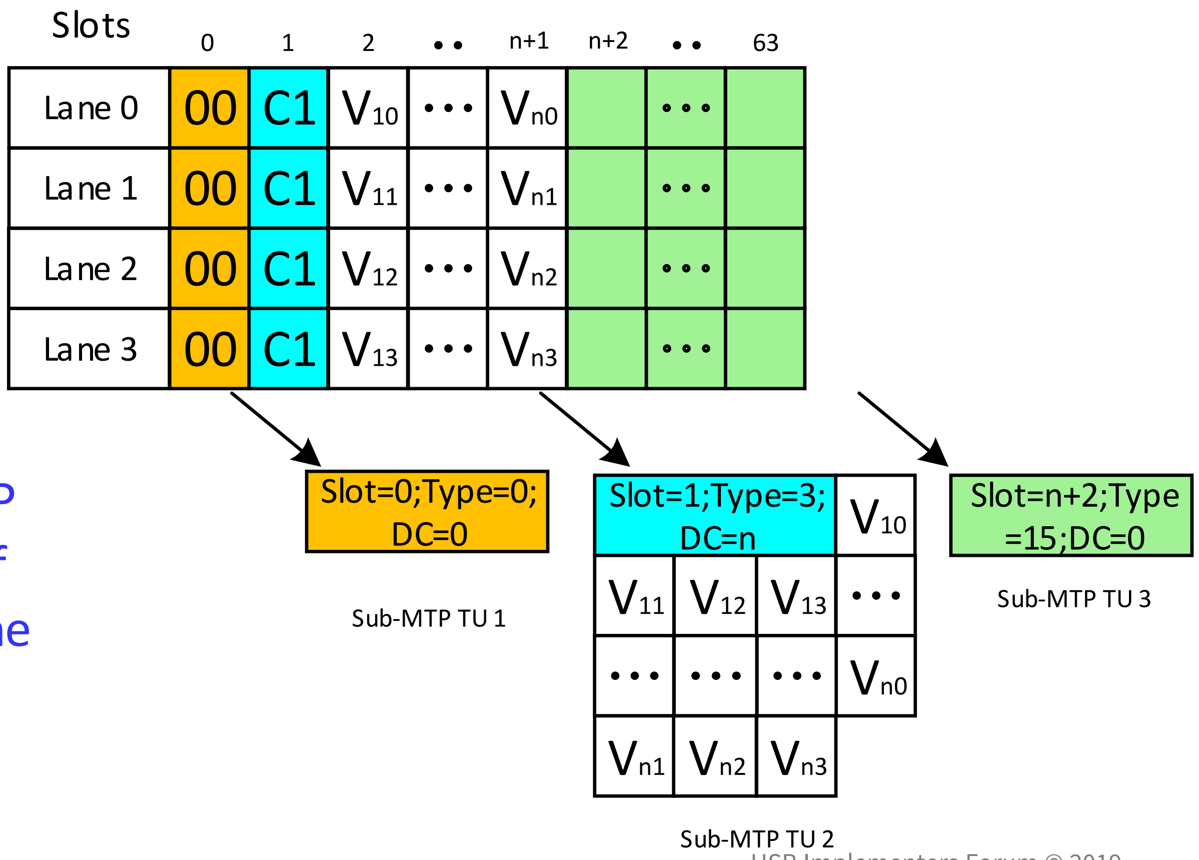
#n Sub-MTP TU

Padding

Main-Link Tunneling – MST

- A *Sub-MTP TU* has a 3 byte header, optional parameters and optional data
  - The header includes the type, the length and the start Slot inside the Native MTP
  - The Parameters may include

    Data or K-Symbol index



Slots	0	1	2	••	n+1	n+2	••	63
Lane 0	00	C1	V_{10}	•••	V_{n0}		•••	
Lane 1	00	C1	V_{11}	•••	V_{n1}		•••	
Lane 2	00	C1	V_{12}	•••	V_{n2}		•••	
Lane 3	00	C1	V_{13}	•••	V_{n3}		•••	

Slot=0;Type=0; DC=0

Sub-MTP TU 1

Slot=1;Type=3; DC=n			V_{10}
V_{11}	V_{12}	V_{13}	•••
•••	•••	•••	V_{n0}
V_{n1}	V_{n2}	V_{n3}	

Sub-MTP TU 2

Slot=n+2;Type =15;DC=0

Sub-MTP TU 3

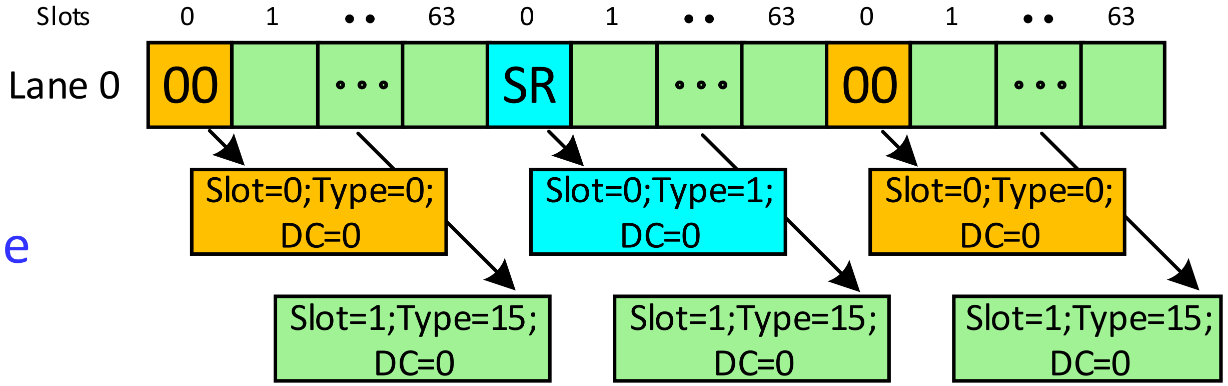
Map a 4 Lanes MTP with BE, ‘n’ slots of active video and the rest is unallocated

Main-Link Tunneling – MST

- A *Sub-MTP TU* has a 3 byte header, optional parameters and optional data
  - The header includes the type, the length and the start Slot inside the Native MTP
  - The Parameters may include

    Data or K-Symbol index

Map a 1 Lane unallocated sequence

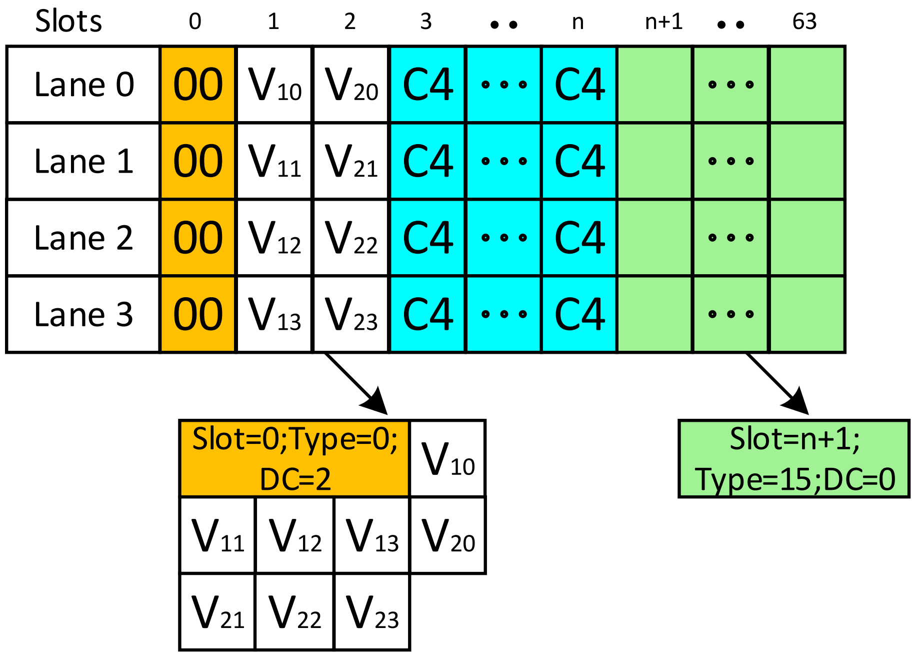


Main-Link Tunneling – MST

- A *Sub-MTP TU* has a 3 byte header, optional parameters and optional data
  - The header includes the type, the length and the start Slot inside the Native MTP
  - The Parameters may include

    Data or K-Symbol index

Map a 4 Lanes MTP with 2 cycles of Active pixel data followed by SF and unallocated slots



Slots	0	1	2	3	••	n	n+1	••	63
Lane 0	00	V_{10}	V_{20}	C4	•••	C4		•••	
Lane 1	00	V_{11}	V_{21}	C4	•••	C4		•••	
Lane 2	00	V_{12}	V_{22}	C4	•••	C4		•••	
Lane 3	00	V_{13}	V_{23}	C4	•••	C4		•••	

Slot=0;Type=0; DC=2			V_{10}
V_{11}	V_{12}	V_{13}	V_{20}
V_{21}	V_{22}	V_{23}	

Slot=n+1; Type=15;DC=0

Time for Q&A

Agenda

- Configuration Layer
- USB3 Tunneling
- DP Tunneling
- PCIe Tunneling

PCIe Tunneling Agenda

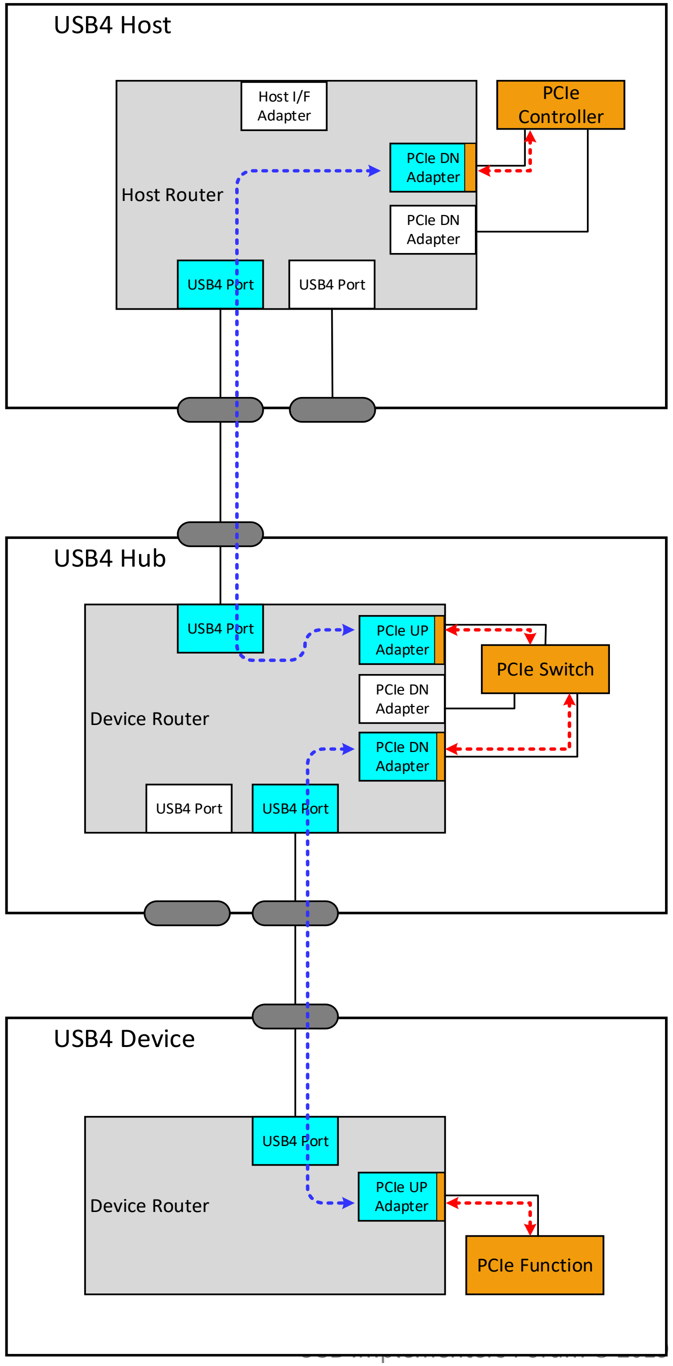
- System View
- Internal PCIe Device
- Protocol Stack
- PCIe Adapter
  - Paths
  - Encapsulation

System View

- Native PCIe is Tunneled over USB4 Fabric

- Originates and consumed as Native PCIe protocol
- From PCIe SW perspective, the PCIe tree remains the same

- **PCIe Adapters** are the translators within each Router that allows PCIe protocol to travel back an forth from Native to Tunneled

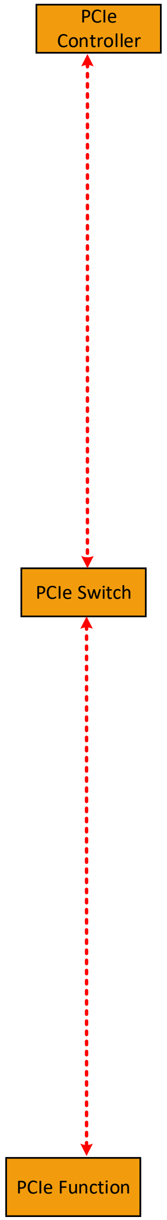


System View

- Native PCIe is Tunneled over USB4 Fabric

- Originates and consumed as Native PCIe protocol
- From PCIe SW perspective, the PCIe tree remains the same

- **PCIe Adapters** are the translators within each Router that allows PCIe protocol to travel back an forth from Native to Tunneled

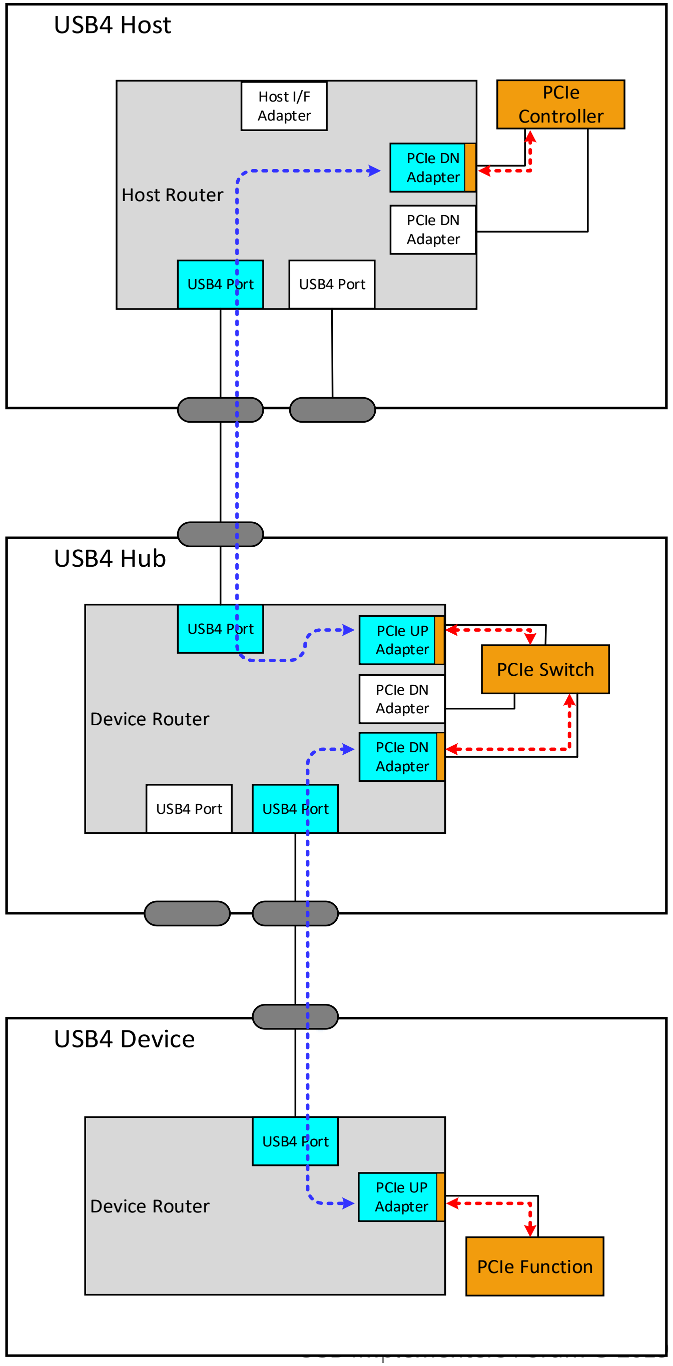


System View

- Native PCIe is Tunneled over USB4 Fabric

- Originates and consumed as Native PCIe protocol
- From PCIe SW perspective, the PCIe tree remains the same

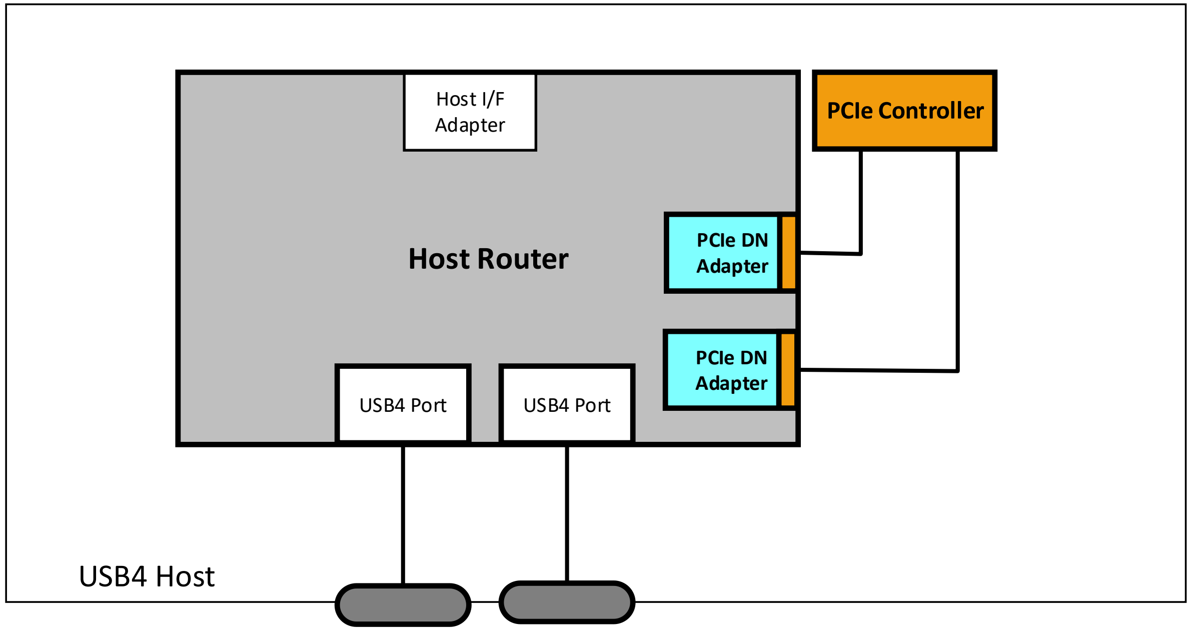
- **PCIe Adapters** are the translators within each Router that allows PCIe protocol to travel back an forth from Native to Tunneled



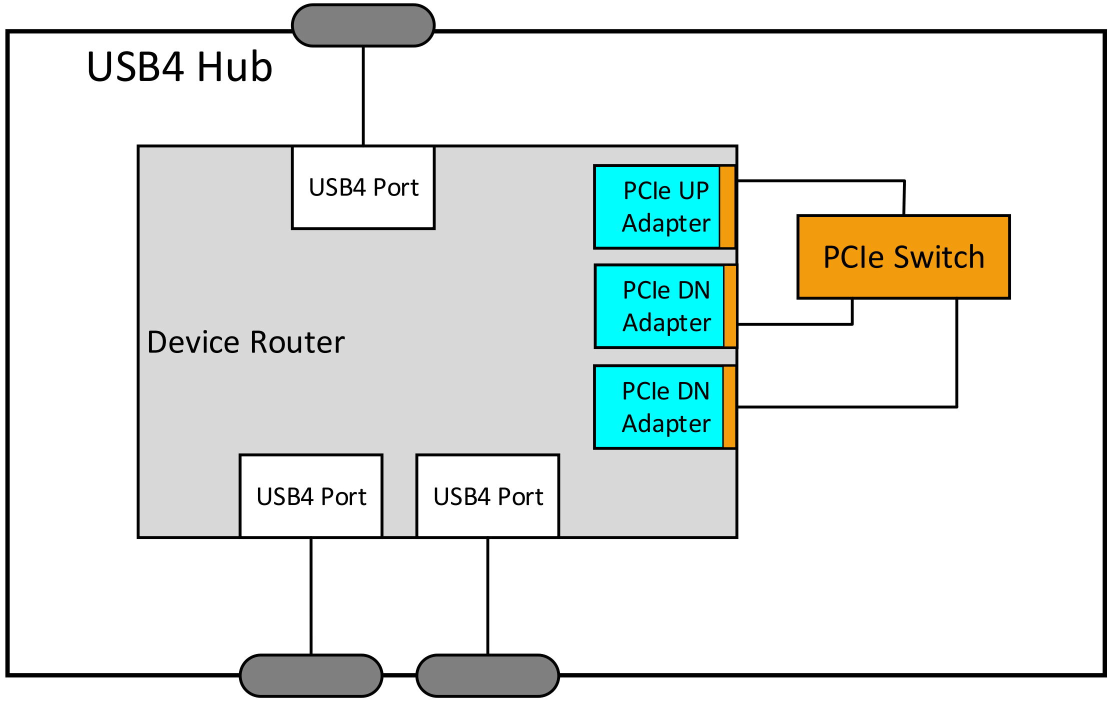
System View – USB4^{TM} Host

- *USB4 Host* can optionally support PCIe Tunneling

- If *USB4 Host* supports PCIe Tunneling then:
  - *It* implements a PCIe controller
  - *Host Router* has ‘*N*’ PCIe Downstream Adapters
    - ‘*N*’ – Number of Downstream USB Type-C connectors



System View – USB4^{TM} Hub

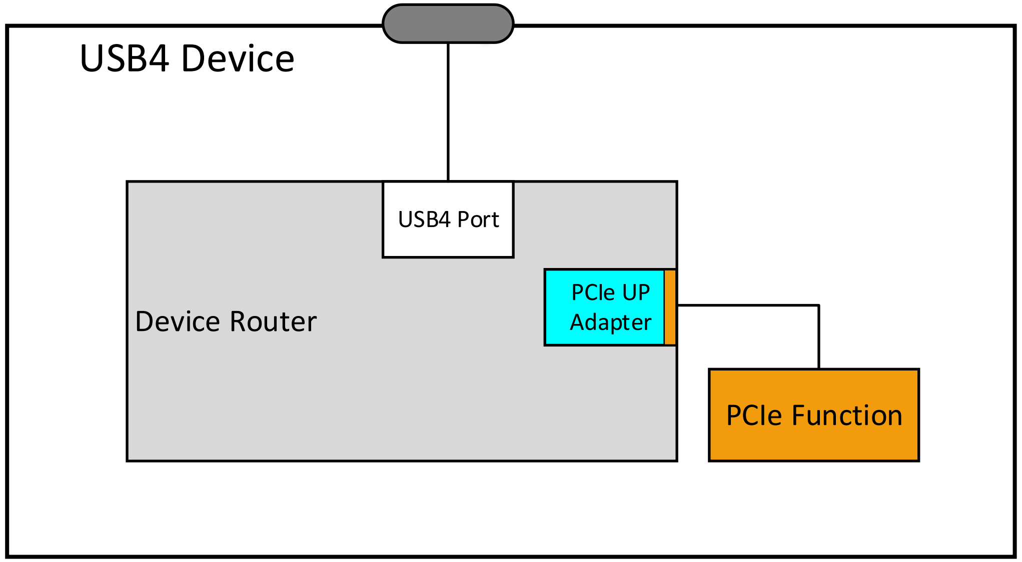


- *USB4 Hub* must support PCIe Tunneling
- *USB4 Hub* implements a PCIe Switch
- *Device Router* implements a PCIe Upstream Adapter
- *Device Router* has ‘*N*’ PCIe Downstream Adapters
  - ‘*N*’ – Number of Downstream USB Type-C connectors

System View – USB4^{TM} Peripheral Device

- *USB4 Peripheral Device* can optionally support PCIe Tunneling

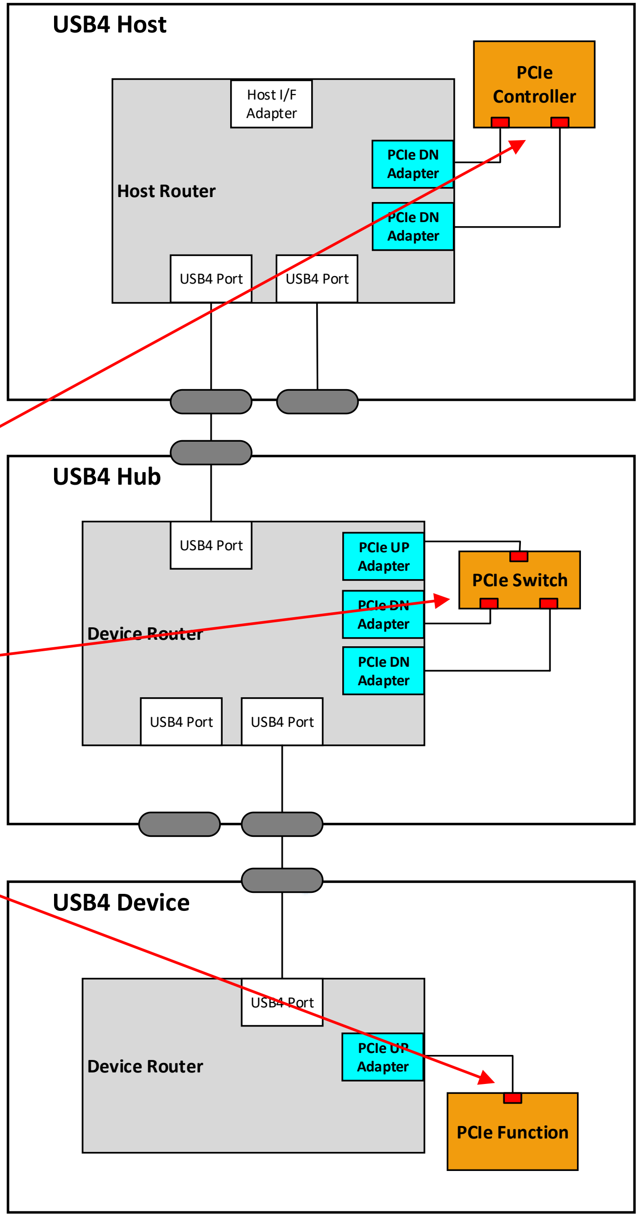
- If the *USB4 Peripheral Device* supports PCIe Tunneling then:
  - *It* implements an internal PCIe Endpoint or Switch
  - *Device Router* implements a USB3 Upstream Adapter



Internal PCIe Port

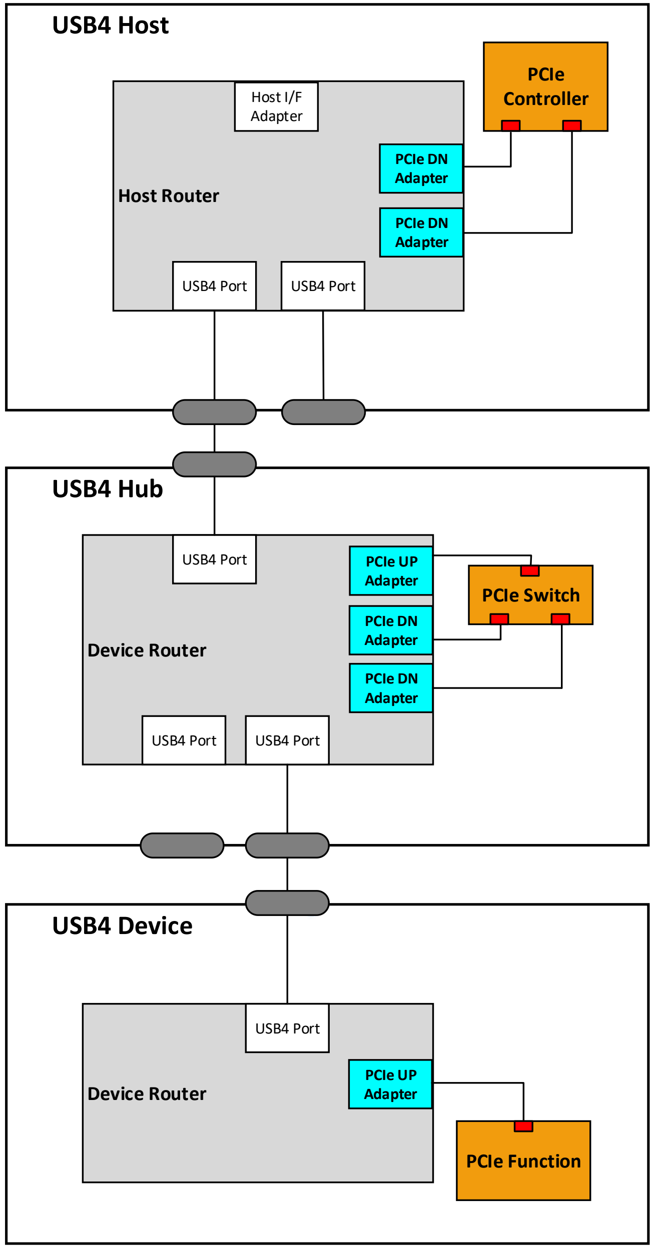
- **Internal PCIe Port** refers to either a **Root Complex PCIe port** , **internal PCIe Switch Port**, or **internal PCIe Endpoint port**

- **Internal PCIe Ports** that interface with a PCIe Adapter differ from the PCIe Spec, mainly at the Physical-Logical and Transaction Layers behavior



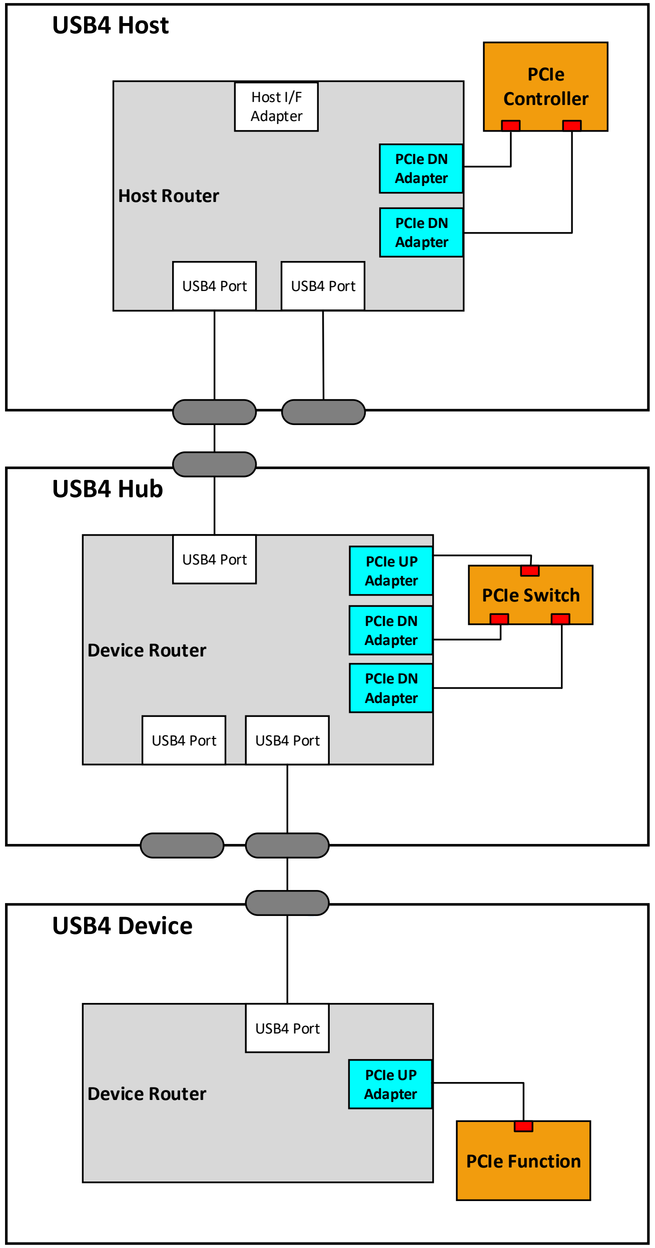
Internal PCIe Port

- **Internal PCIe Ports** that interface with a PCIe Adapter differ from the PCIe Spec
  - No *Physical **Electrical** Sub-Block Layer*
  - *Physical **Logical** Sub-Block Layer*
    - Operates and support *Gen 1* only
      - As this is a virtual link, the actual speed and throughput could be higher than *Gen 1*
    - *Link width* Negotiation is not applicable
      - Only TS for Lane0 are present over the tunnel
    - No Scrambling
    - *L0s* and *L1 PM Sub-states* are not supported



Internal PCIe Port

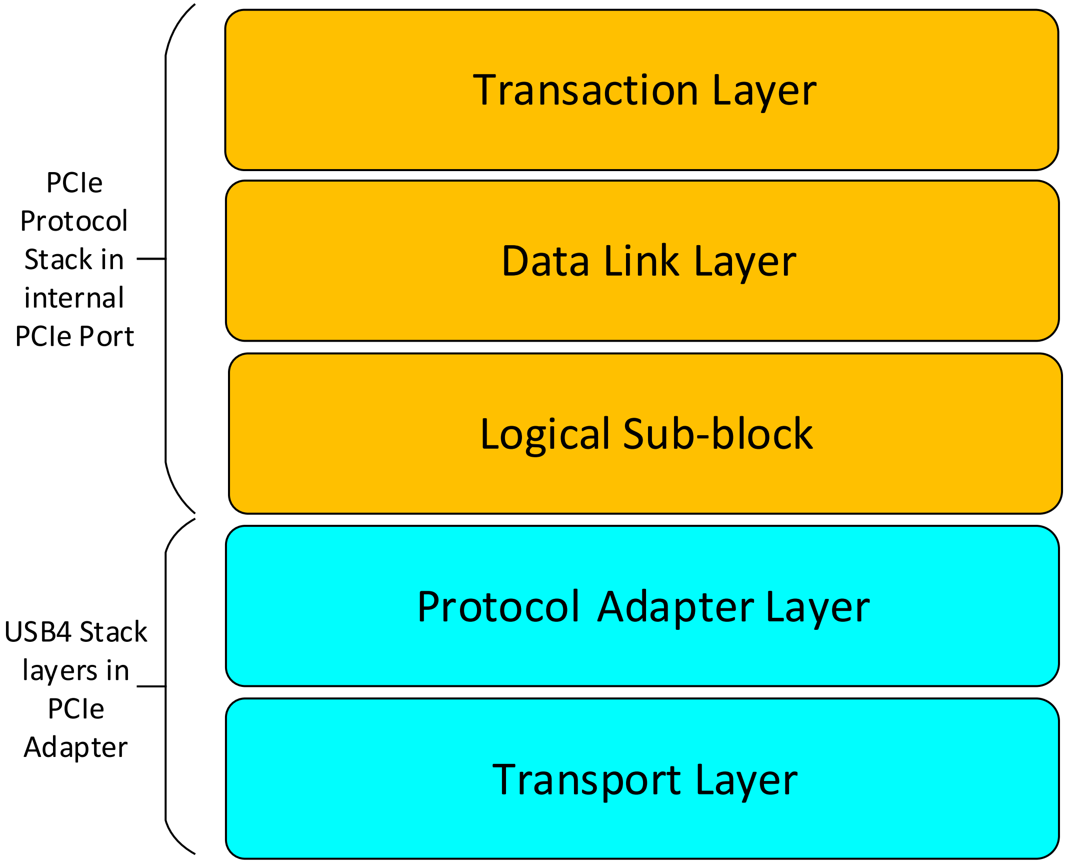
- **Internal PCIe Ports** that interface with a PCIe Adapter differ from the PCIe Spec
  - **Transaction** *Layer*
    - A Max Payload Size of *128* Bytes
    - *LTR* must be supported
    - *Hot-add* and *hot-removal* must be supported
    - A USB4 Hub must support
      - Access Control Services (*ACS*)
      - Flattening Portal Bridge (*FPB*)



Protocol Stack

- The Internal PCIe Port interfaces to the PCIe Adapter layer after the Physical Logical Sub-block

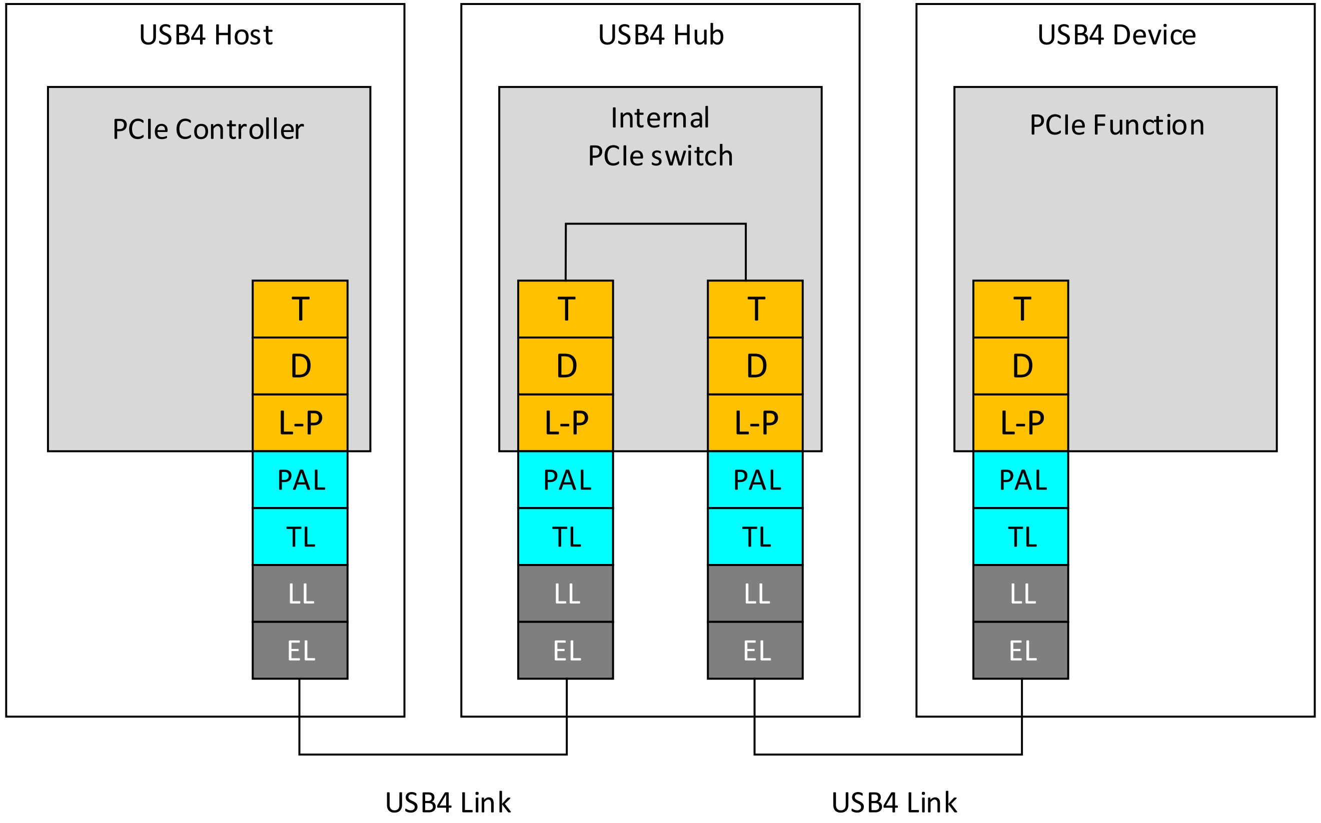
- The PCIe Adapter encapsulates the Native PCIe protocol into USB4 Transport Layer Packets



Protocol Stack

- The Internal PCIe Port interfaces to the PCIe Adapter layer after the Physical Logical Sub-block

- The PCIe Adapter encapsulates the Native PCIe protocol into USB4 Transport Layer Packets



USB4 Host

PCIe Controller

T
D
L-P
PAL
TL
LL
EL

USB4 Hub

Internal
PCIe switch

T
D
L-P
PAL
TL
LL
EL

T
D
L-P
PAL
TL
LL
EL

USB4 Device

PCIe Function

T
D
L-P
PAL
TL
LL
EL

USB4 Link

USB4 Link

Legend (USB4 stack):

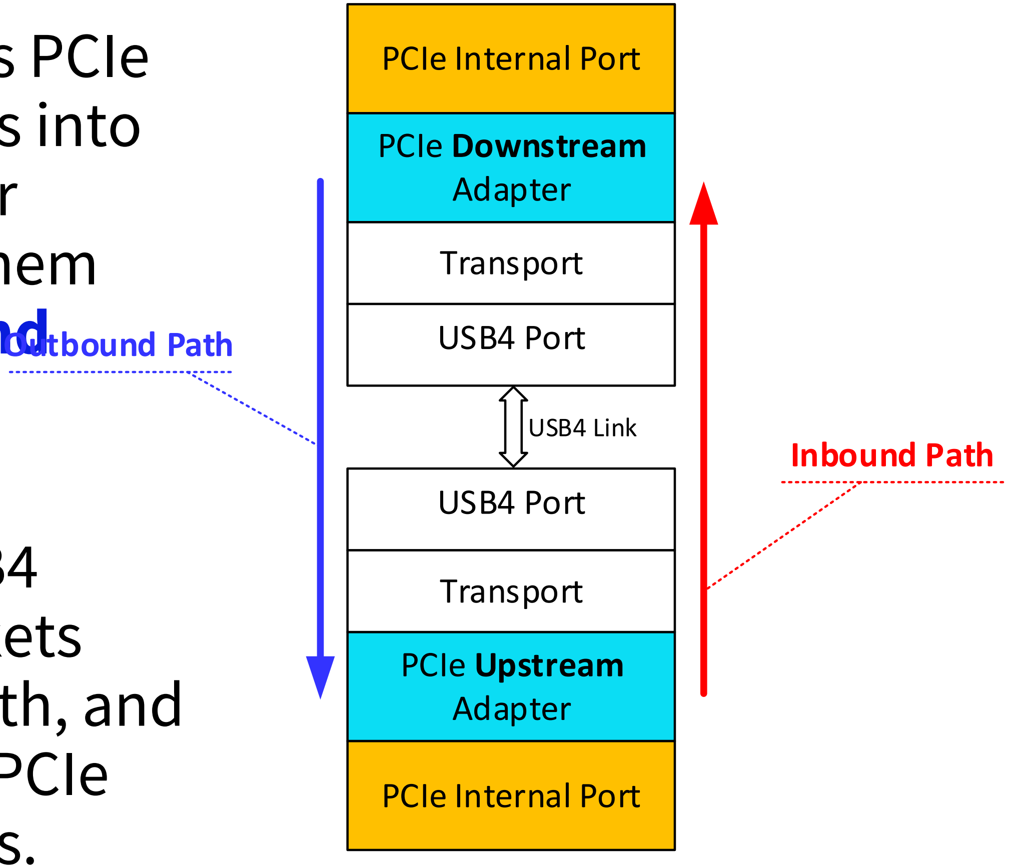
PAL	Protocol Adapter Layer
TL	Transport Layer
LL	Logical Layer
EL	Electrical Layer

Legend (PCIe stack):

T	Transaction Layer
D	Data Link Layer
L-P	Logical sub-block

PCIe Adapter - Paths

- A PCIe **Downstream** Adapter encapsulates PCIe events and constructs into USB4 Transport Layer Packets, and sends them through the **Outbound** Path.
- A PCIe **Downstream** Adapter receives USB4 Transport Layer Packets from the **Inbound** Path, and translates them into PCIe events and constructs.



- A PCIe **Upstream** Adapter encapsulates PCIe events and constructs into USB4 Transport Layer Packets, and sends them through the **Inbound** Path.
- A PCIe **Upstream** Adapter receives USB4 Transport Layer Packets from the **Outbound** Path, and translates them into PCIe events and constructs.

PCIe Adapter - Encapsulation

PDF	Type	Payload of Tunneled Packet
1h	Ordered Set	One PCIe EIOS, TS1, or TS2 Ordered Set.
2h	Electrical Idle State	Indication of an Electrical Idle state on PCIe.
3h	TLP/DLLP	PCIe TLPs and/or DLLPs.
5h	PERST Active	PCIe Reset (PERST) in active state.
6h	PERST Inactive	PCIe Reset (PERST) in inactive state.

- The PDF defines the construct being tunneled
- Logical Idle Symbols are not tunneled
- Each Ordered Set and PCIe out-of-band event is encapsulated into a separate Tunneled Packet

- The order of bytes and bits in the Tunneled Packet are identical to the original PCIe construct. The least-significant byte of the encapsulated construct is mapped to B0 in the Tunneled Packet Payload.

PCIe Adapter – Encapsulation – Ordered Set

- Only **TS1**, **TS2** and **EI** Ordered Set are tunneled
- Only Ordered Set for **Lane 0** are tunneled
- The Ordered Set is tunneled in its **entirety**
- K-Chars are replaced with 8 bit representation. Example of **EIOS**:

<table>
<tr><td>PDF = 1</td><td>*</td><td colspan="2">HOP ID</td><td>Length = 4</td><td>HEC</td></tr>
<tr><td colspan="3">BCh (COM)</td><td>7Ch (IDL)</td><td>7Ch (IDL)</td><td>7Ch (IDL)</td></tr>
</table>

- When a PCIe Adapter receives one or more identical Ordered Sets from the Internal PCIe Port it transmits **N** Ordered-Set Tunneled PCIe Packets
  - **TS** – N is **at least 16**
  - **EI** – N is **2**
- When a PCIe Adapter receives an Ordered Set Tunneled PCIe Packets it sends the Ordered Set content to the Internal PCIe Port

PCIe Adapter – Encapsulation – EI State

- When an Internal PCIe Port indicates that it is in Electrical Idle state, a PCIe Adapter sends at least 3 **Electrical Idle State** Tunneled Packets:

<table>
<tr><td>PDF = 2</td><td>*</td><td colspan="2">HOP ID</td><td>Length = 4</td><td>HEC</td></tr>
<tr><td colspan="3">00h</td><td>00h</td><td>00h</td><td>00h</td></tr>
</table>

- A PCIe Adapter indicates Rx Electrical Idle to the Internal PCIe Port on those events:
  - Receiving **Electrical Idle State** Tunneled Packet
  - Receiving **Ordered-Set Tunneled Packet** with **EI** content
- A PCIe Adapter **stops** indicating Rx Electrical Idle to the Internal PCIe Port if it receives a Tunneled Packet which is not one of the above

PCIe Adapter – Encapsulation – PERST

- Upon detecting an assertion of PERST#, a Host Router sends at least 3 **PERST Active** Tunneled Packets on all Downstream PCIe Adapters:

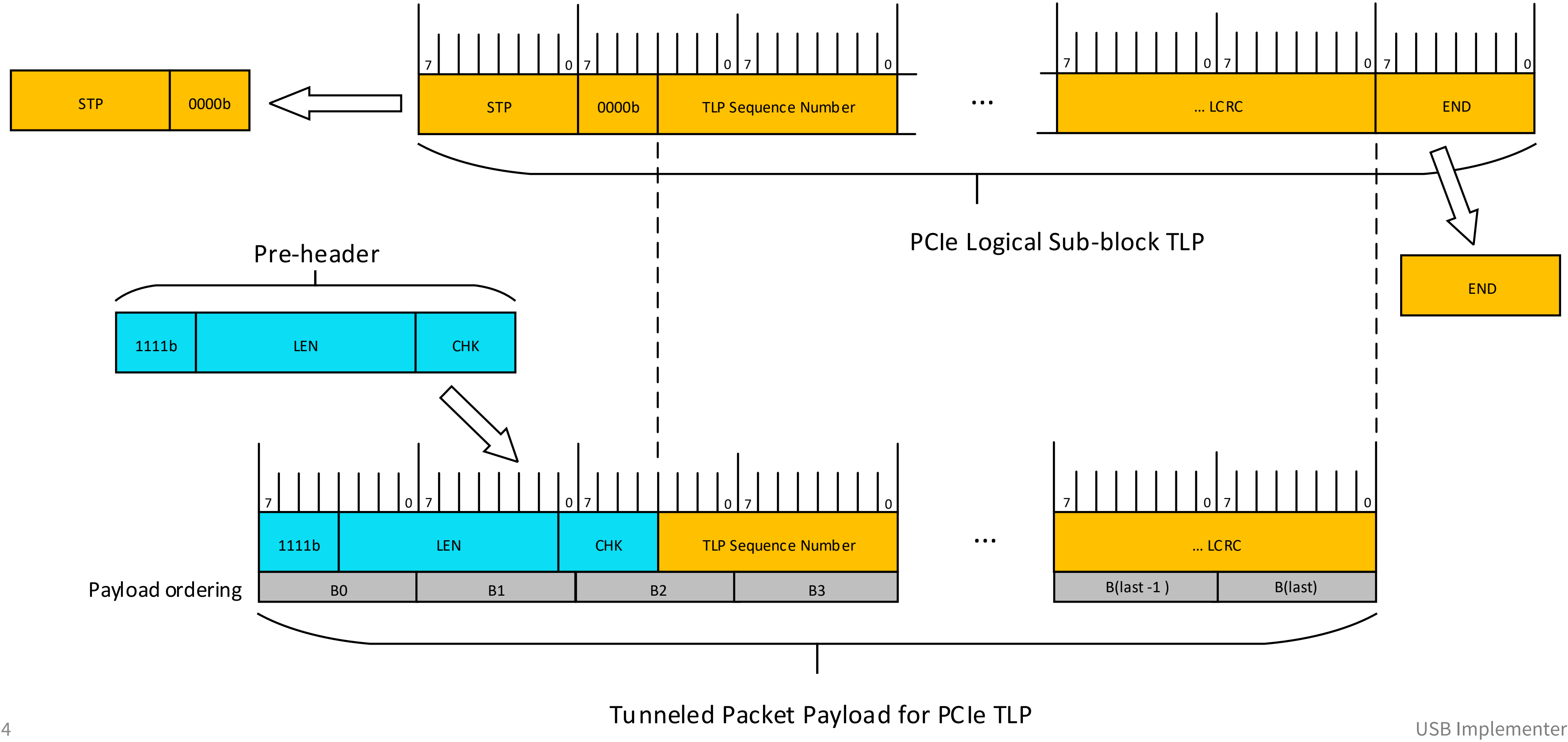
<table>
<tr><td>PDF = 5</td><td>*</td><td colspan="2">HOP ID</td><td>Length = 4</td><td>HEC</td></tr>
<tr><td colspan="3">00h</td><td>00h</td><td>00h</td><td>00h</td></tr>
</table>

- A Device Router which receives **PERST Active** Tunneled Packet on its Upstream PCIe Adapter:
  - Sends at least 3 **PERST Active** Tunneled Packets on all Downstream PCIe Adapters
  - Assert PERST# on all Physical and internal PCIe Ports

- Same flow is executed when detecting de-assertion of PERST# using the **PERST Inactive** Tunneled Packet.

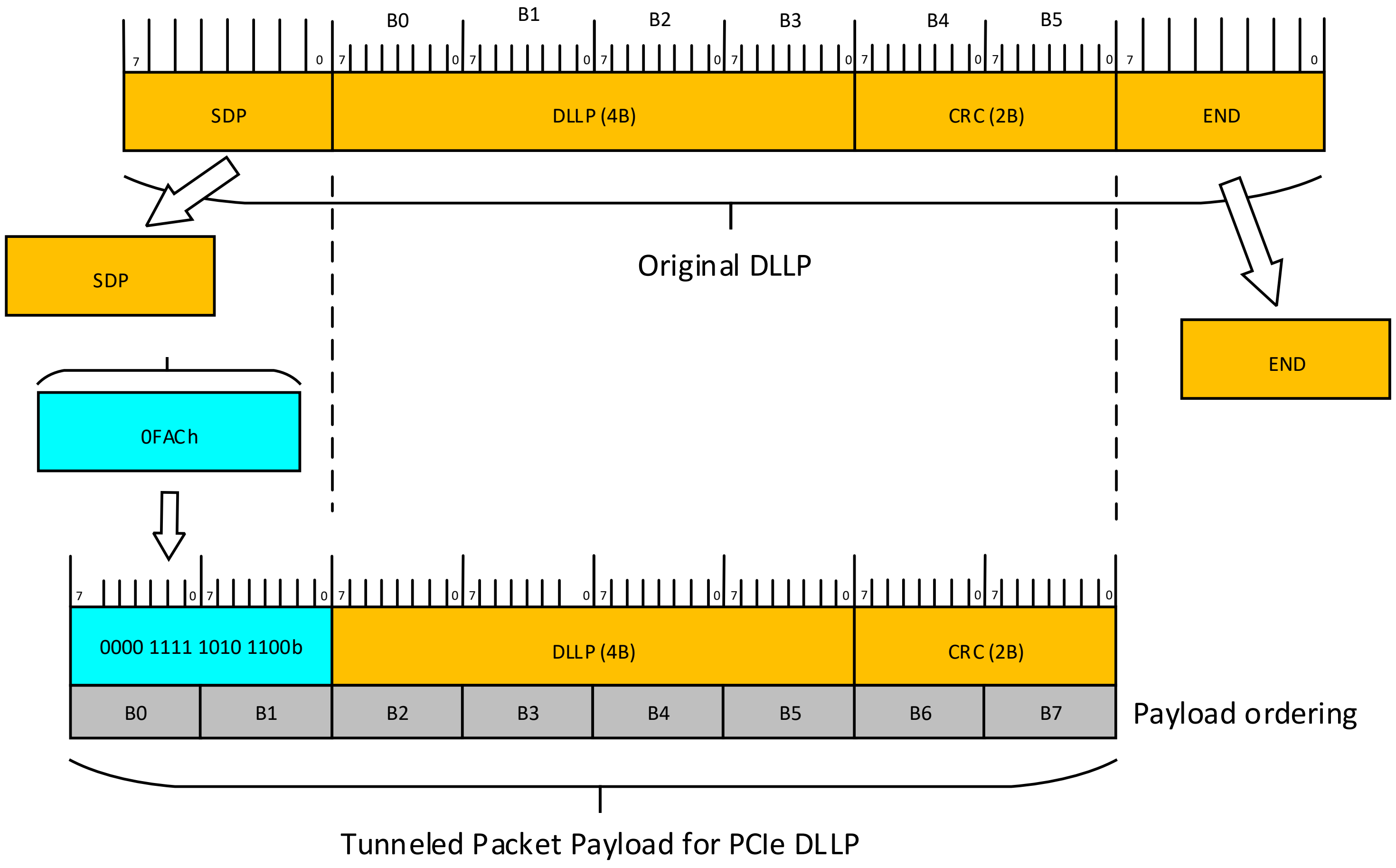
PCIe Adapter – Encapsulation – TLP

- A PCIe Adapter removes **STP** Symbol, 4 leading **reserved** bits and **END** Symbol. It prepends a **pre-header**, containing a marker (**F**h) and length (**LEN**) in DWs



PCIe Adapter – Encapsulation – DLLP

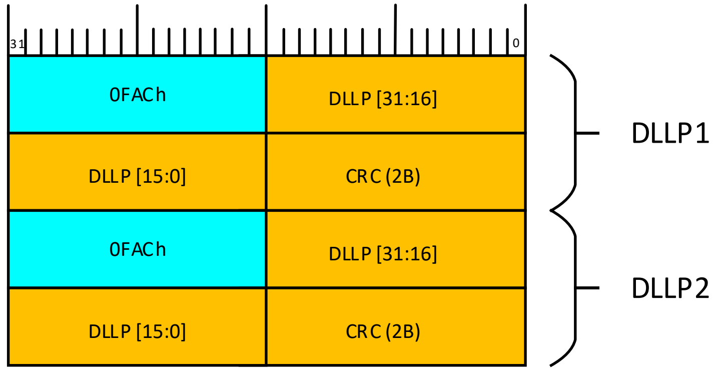
- A PCIe Adapter removes the **STP** Symbol, and the **END** Symbol. It prepends a **pre-header**, containing a marker (**0FACh**)



PCIe Adapter – Encapsulation – Mixed TLP/DLLP

- A PCIe **TLP/DLLP** Tunneled Packet may contain:
  - Single TLP
  - Single DLLP
  - DLLP followed by a DLLP
  - DLLP followed by a TLP

DLLP followed by DLLP



31 0

0FACh | DLLP [31:16]

DLLP [15:0] | CRC (2B)

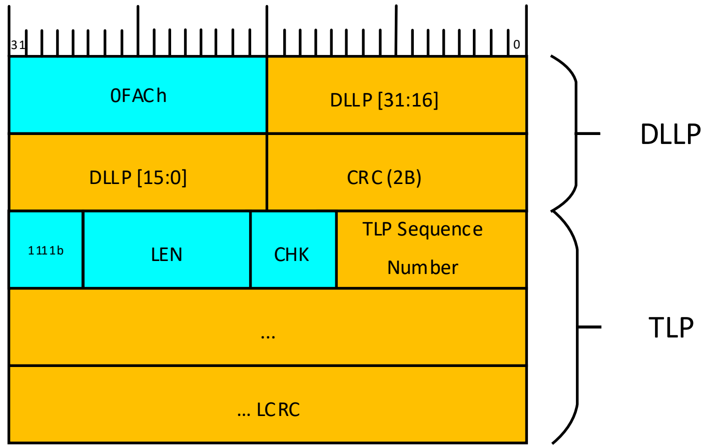
0FACh | DLLP [31:16]

DLLP [15:0] | CRC (2B)

DLLP1

DLLP2

DLLP followed by TLP



31 0

0FACh | DLLP [31:16]

DLLP [15:0] | CRC (2B)

1111b | LEN | CHK | TLP Sequence Number

...

... LCRC

DLLP

TLP

Time for Q&A