SuperSpeed USB Developers Conference

Tokyo, Japan

May 19-20, 2009

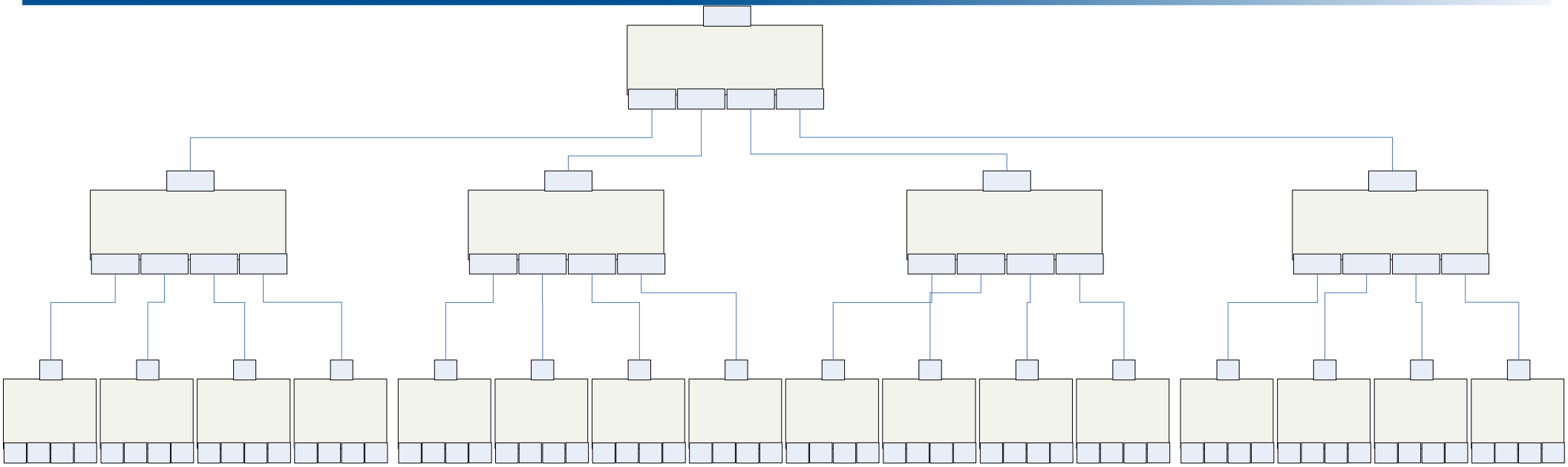
SuperSpeed USB Hubs

Dan Froelich
USB 3.0 Hub Workgroup Chair
Intel Corporation

Agenda

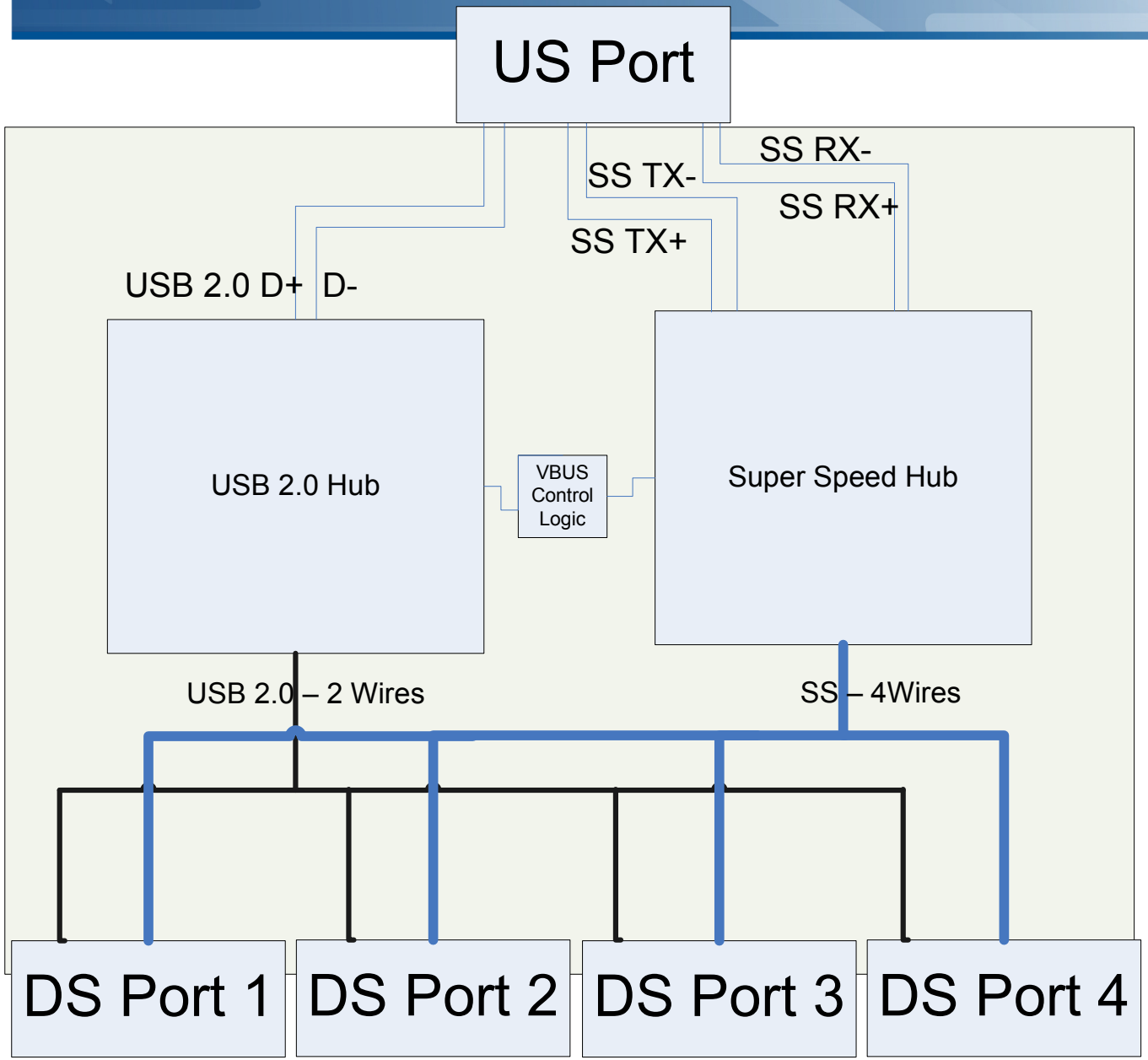
- USB 3.0 Hub Architecture Overview
- USB 3.0 Hub and Peripheral Device Connect State Machines
- Downstream Port State Machine
- Global and Selective Speed Change
- Packet Routing
- Hub Packet Repeater/Forwarder
  - Header Buffer Architecture
  - Data Buffer Architecture
  - Packet Spacing and Insertion Rules
- Hubs and Power Management
- Hubs and Isochronous Transfers
- Resume Signaling
- Conclusions

Hub Bus Expansion



- Hubs have up to 15 downstream ports
  - 4 most common for USB 2.0 hubs
- Up to 5 levels of hubs (three in picture)
- Up to 127 devices

Hub Architecture



Two separate logical hubs:

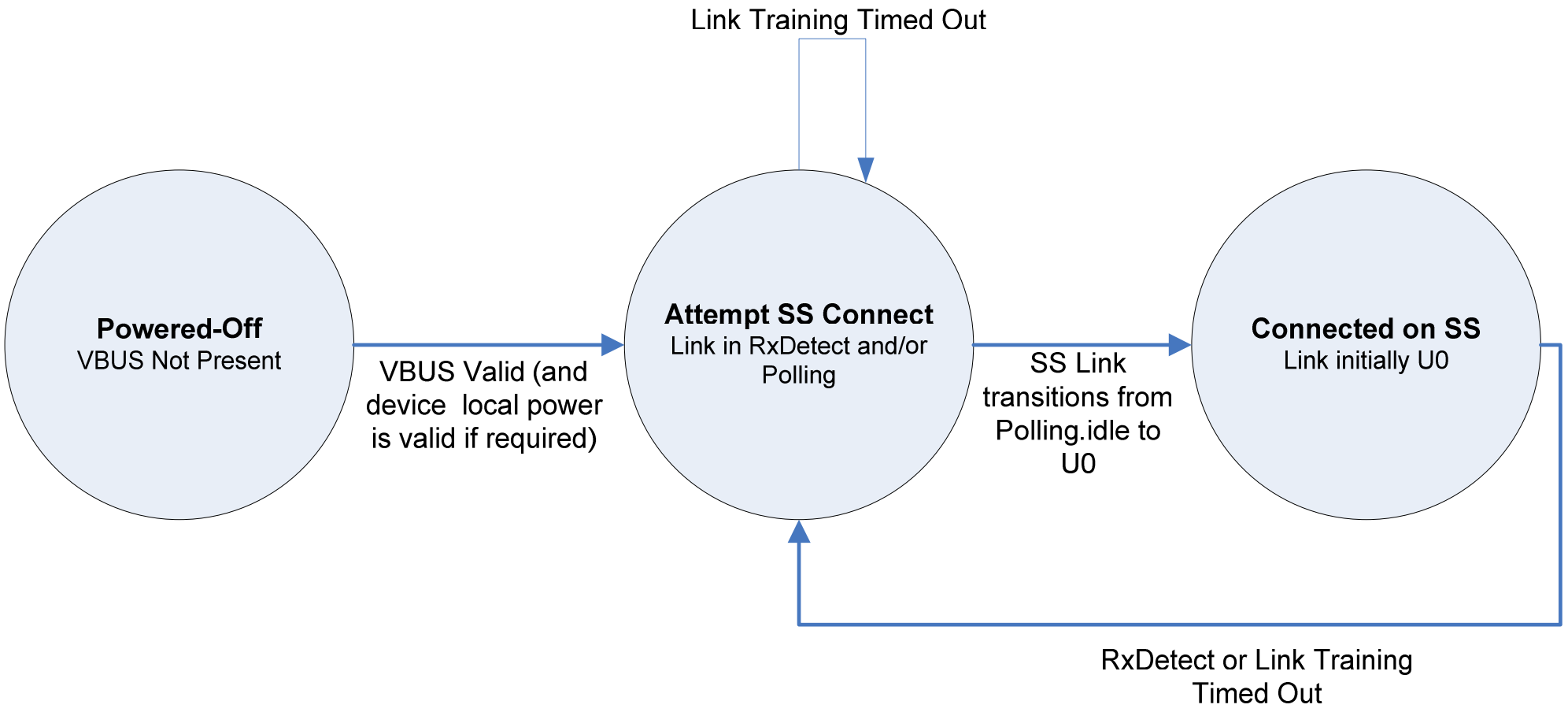
- USB 2.0
- SuperSpeed USB

SuperSpeed hub controller is more than just a repeater. Details on following slides.

Hub connects via upstream port as both SS and USB 2.0 device

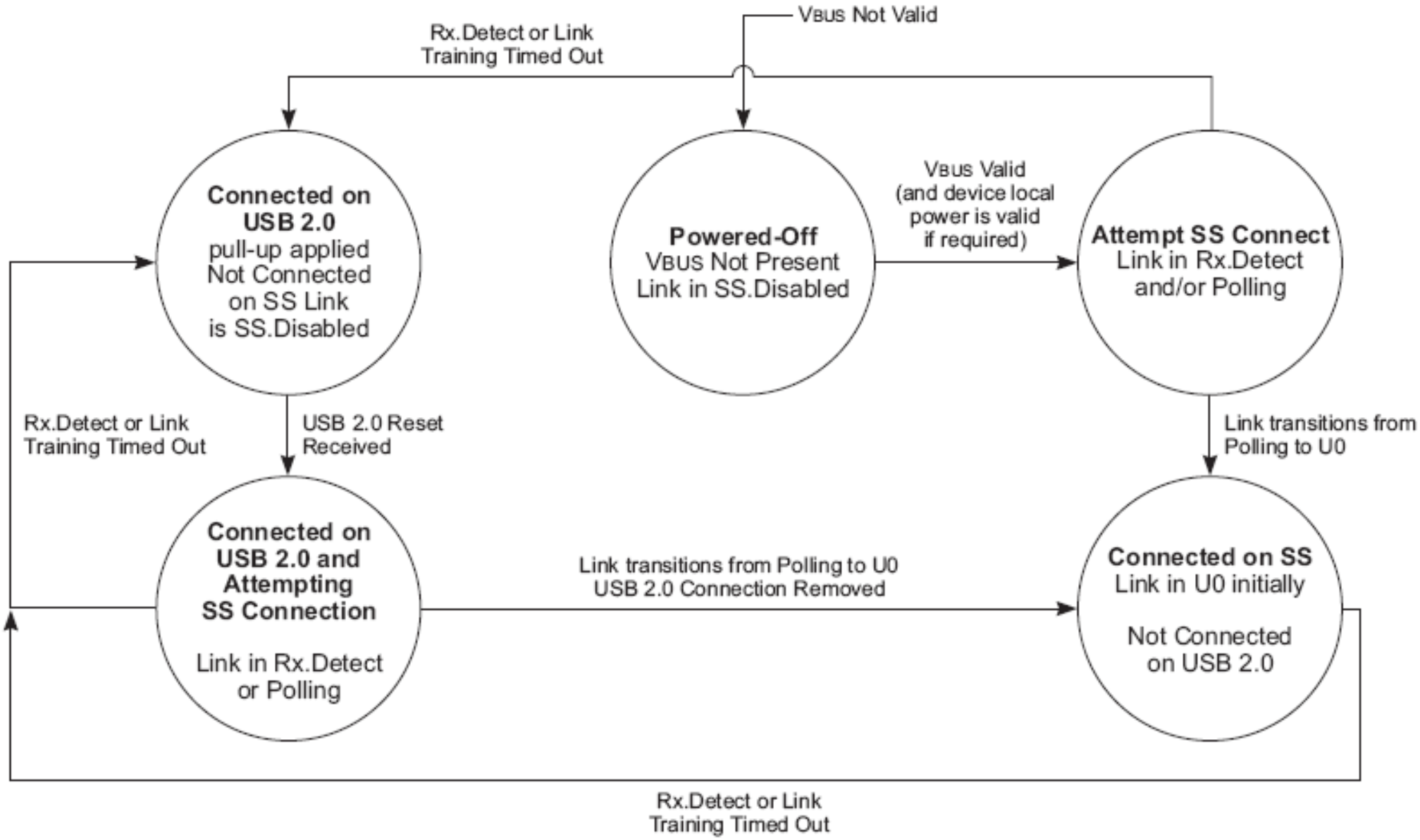
VBUS controlled by both controllers. VBUS on for a DS port if either controller logically requests VBUS

Hub Connect State Machine (SuperSpeed USB)

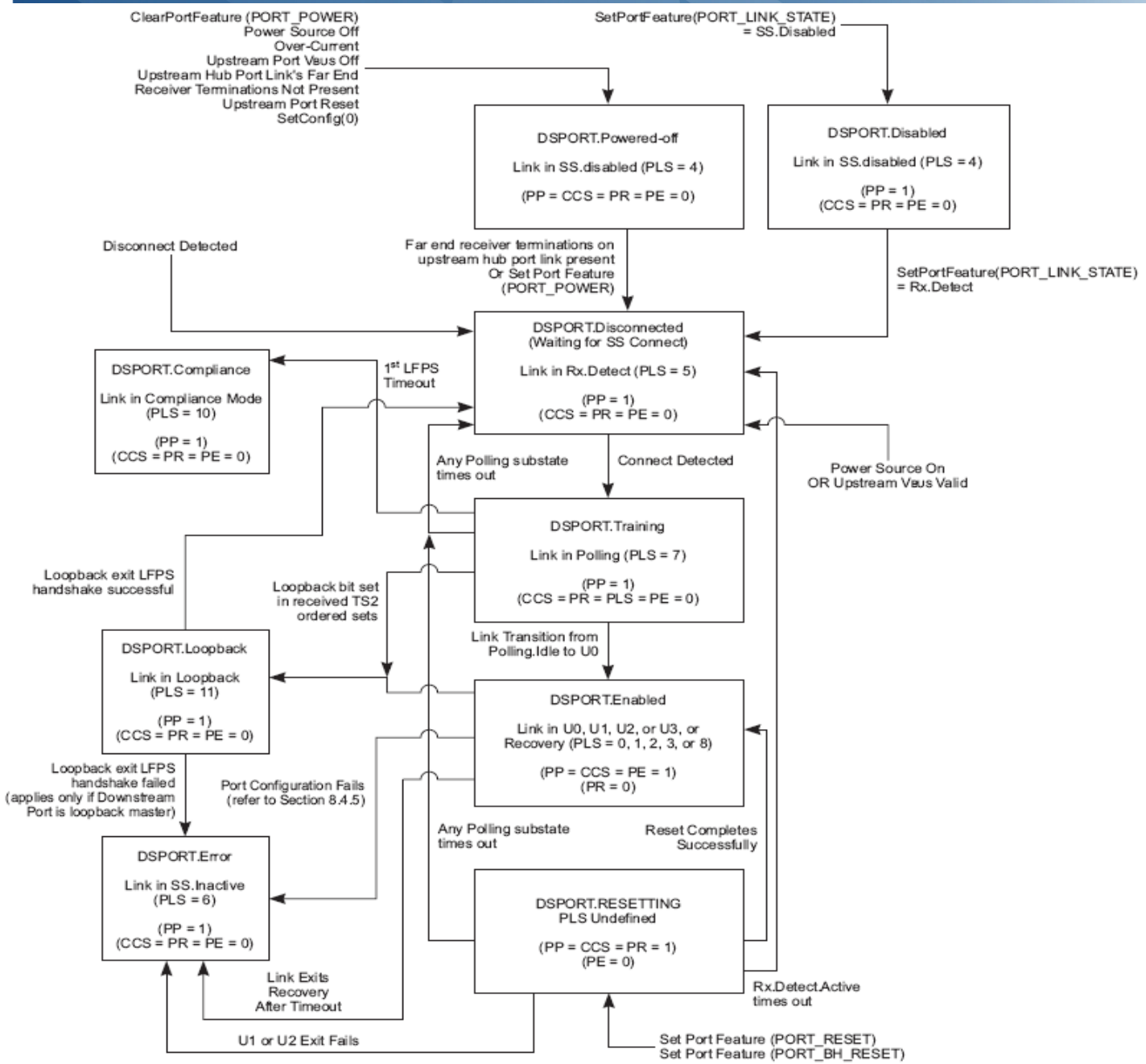


USB 2.0 Connection happens independently of SuperSpeed USB connection per USB 2.0 specification

Peripheral Device Connect State Machine

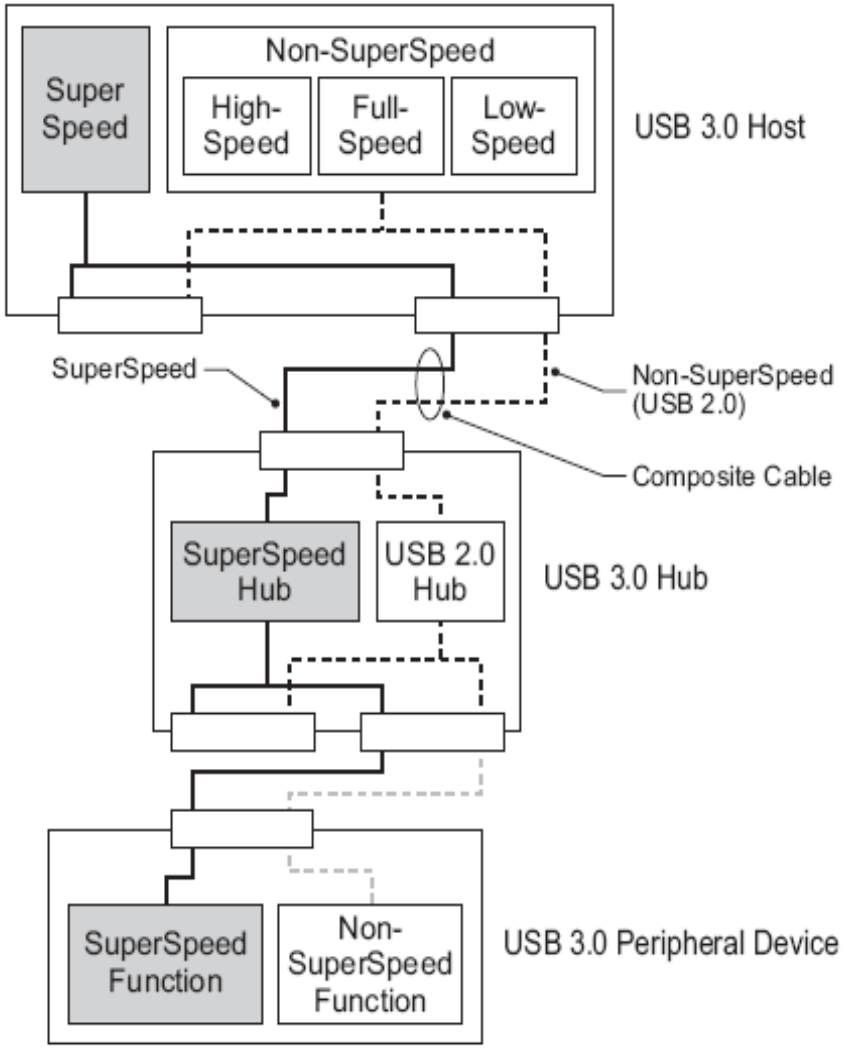


HUB DS Port Connect State Machine



- Significant changes from USB 2.0
- Port States are different than link states
- Transition to Enabled state does not require software
- Hub upstream port state controls SuperSpeed USB Terminations on Downstream Ports

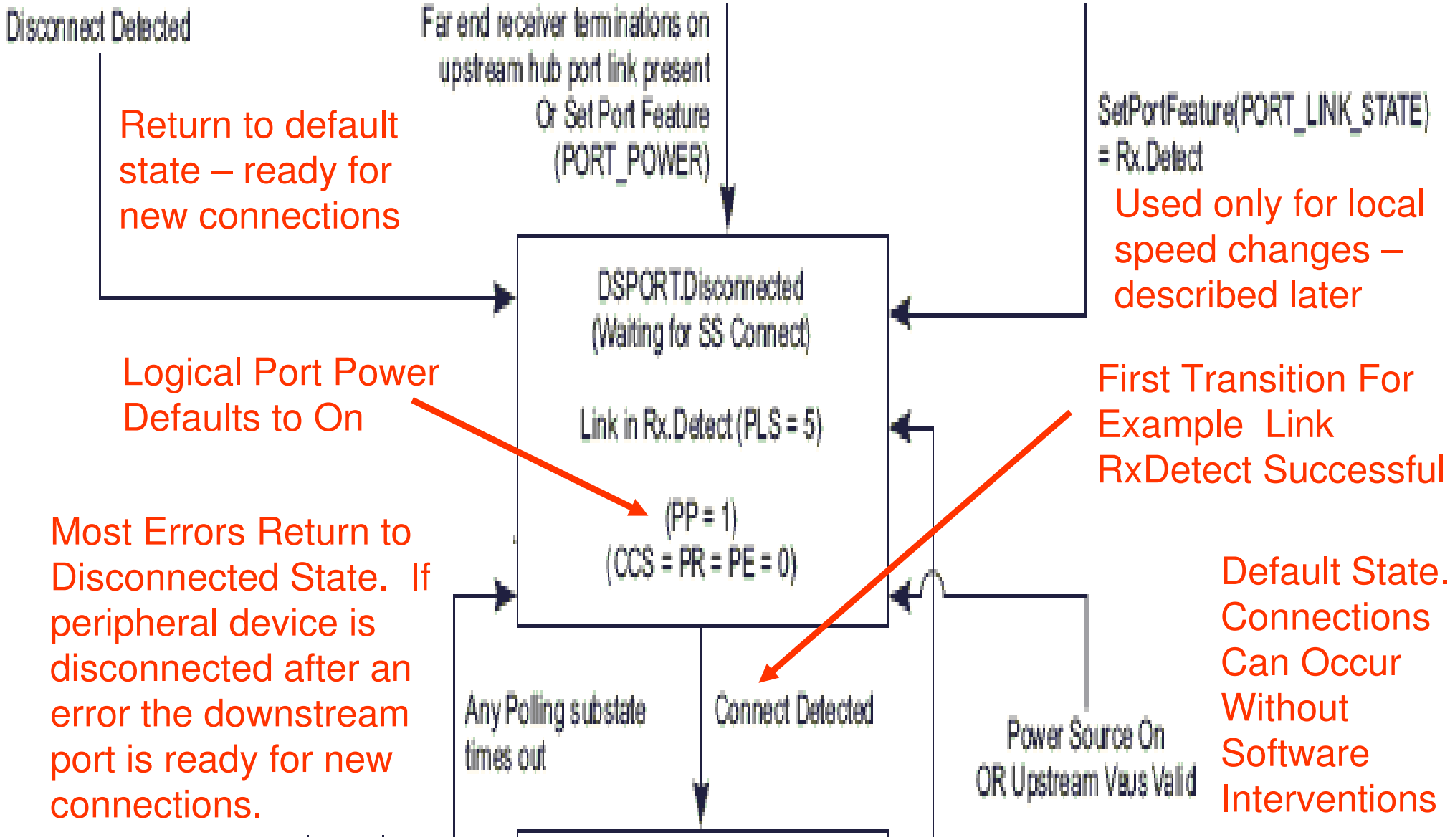
Hub Connection Example



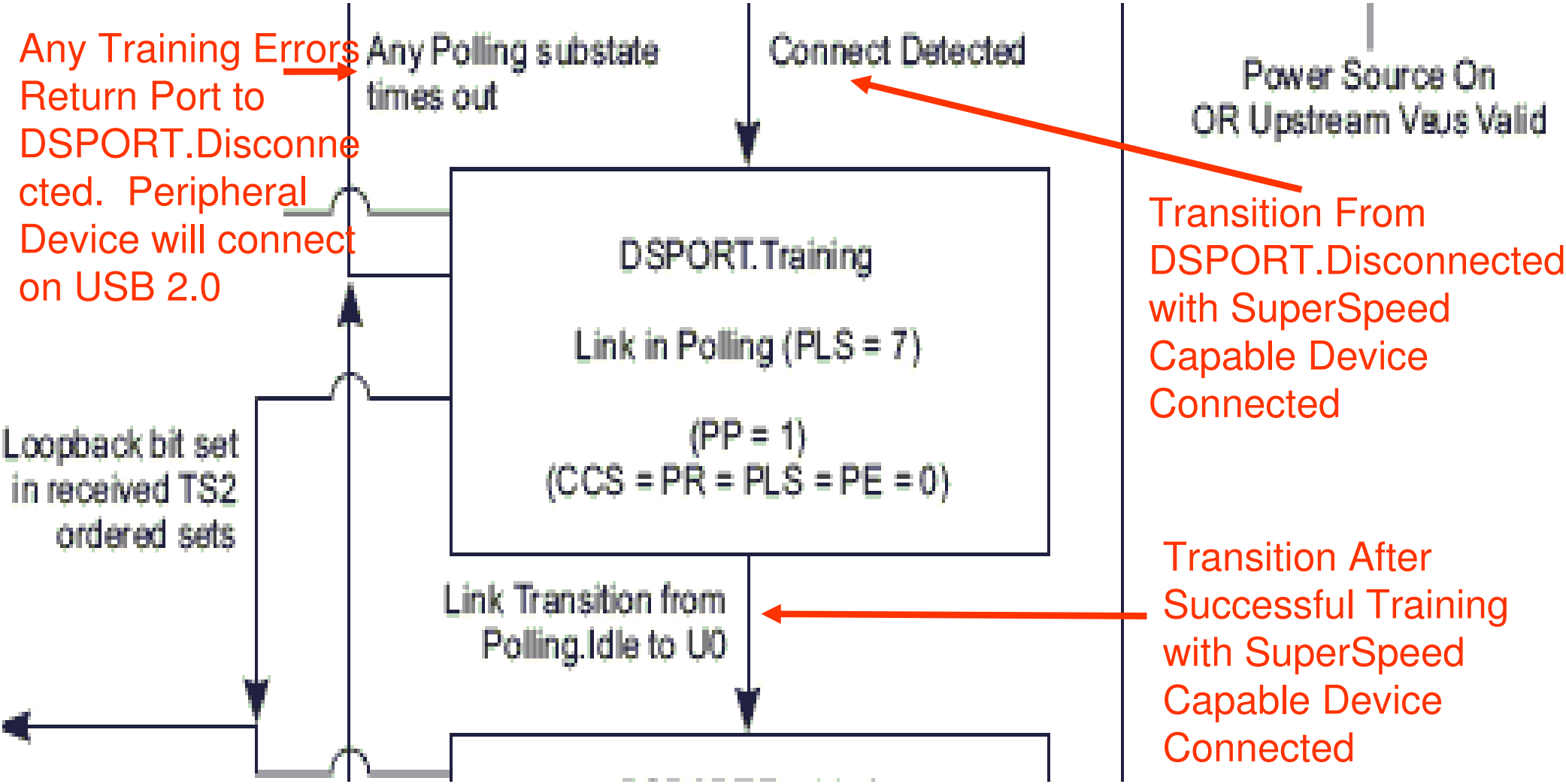
1. Host is powered off
2. Host is powered on with SS support enabled on downstream ports by default
3. Hub detects VBUS and SS support on upstream port and connects as a SS and high-speed device
4. Device detects VBUS and SS support and connects at SS
5. Host system begins hub enumeration at SS and high-speed
6. Host system begins device enumeration at SS

Following Slides Walk Through Downstream Port State Machine Transitions For This Example

Downstream Port Default State DSPORT.Disconnected



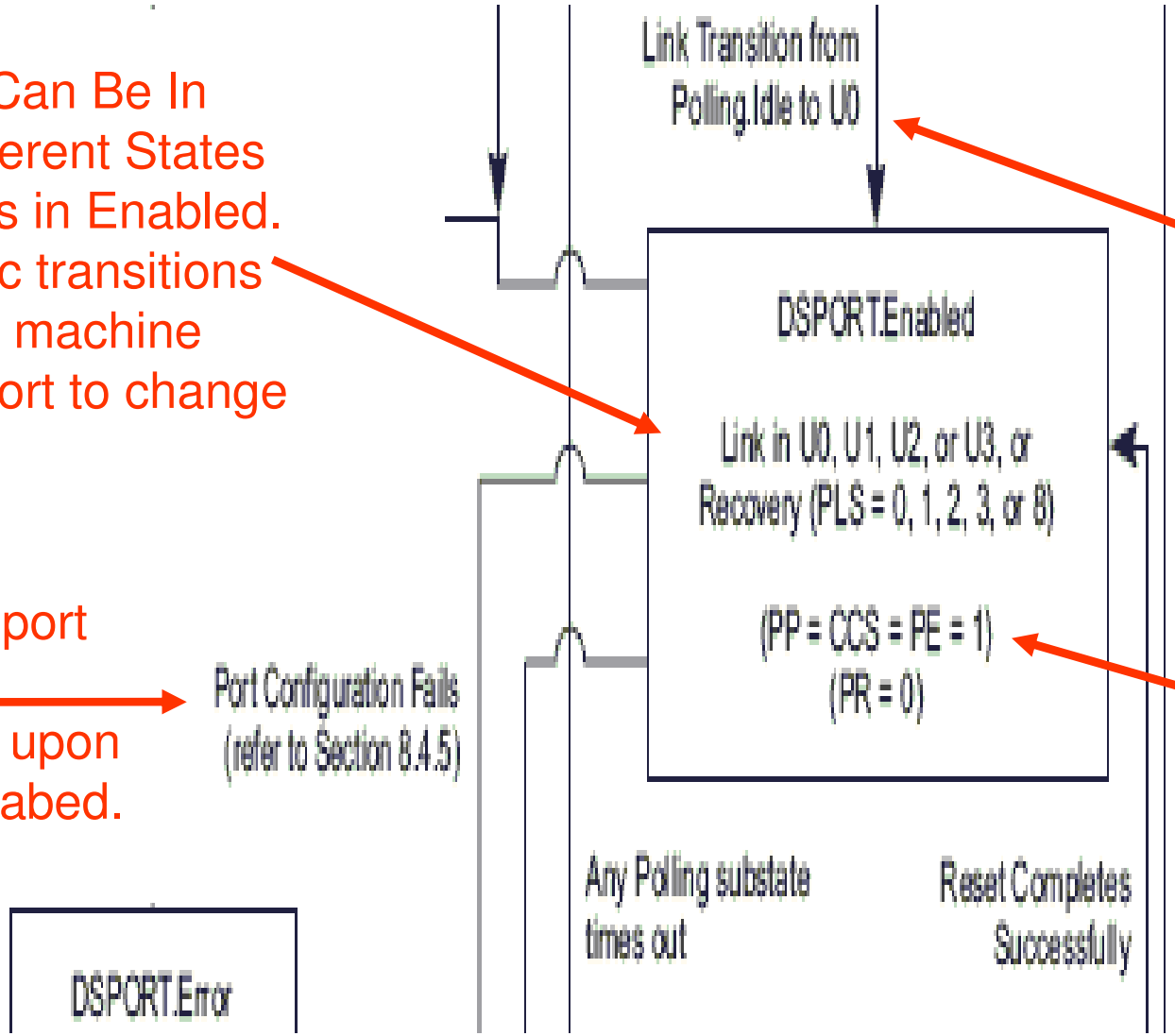
Next State
DSPORT.Training



Next State DSPORT.Enabled

Port's Link Can Be In Several Different States While Port is in Enabled. Only specific transitions in port state machine cause the port to change state.

Port sends port config information upon entering enabed.



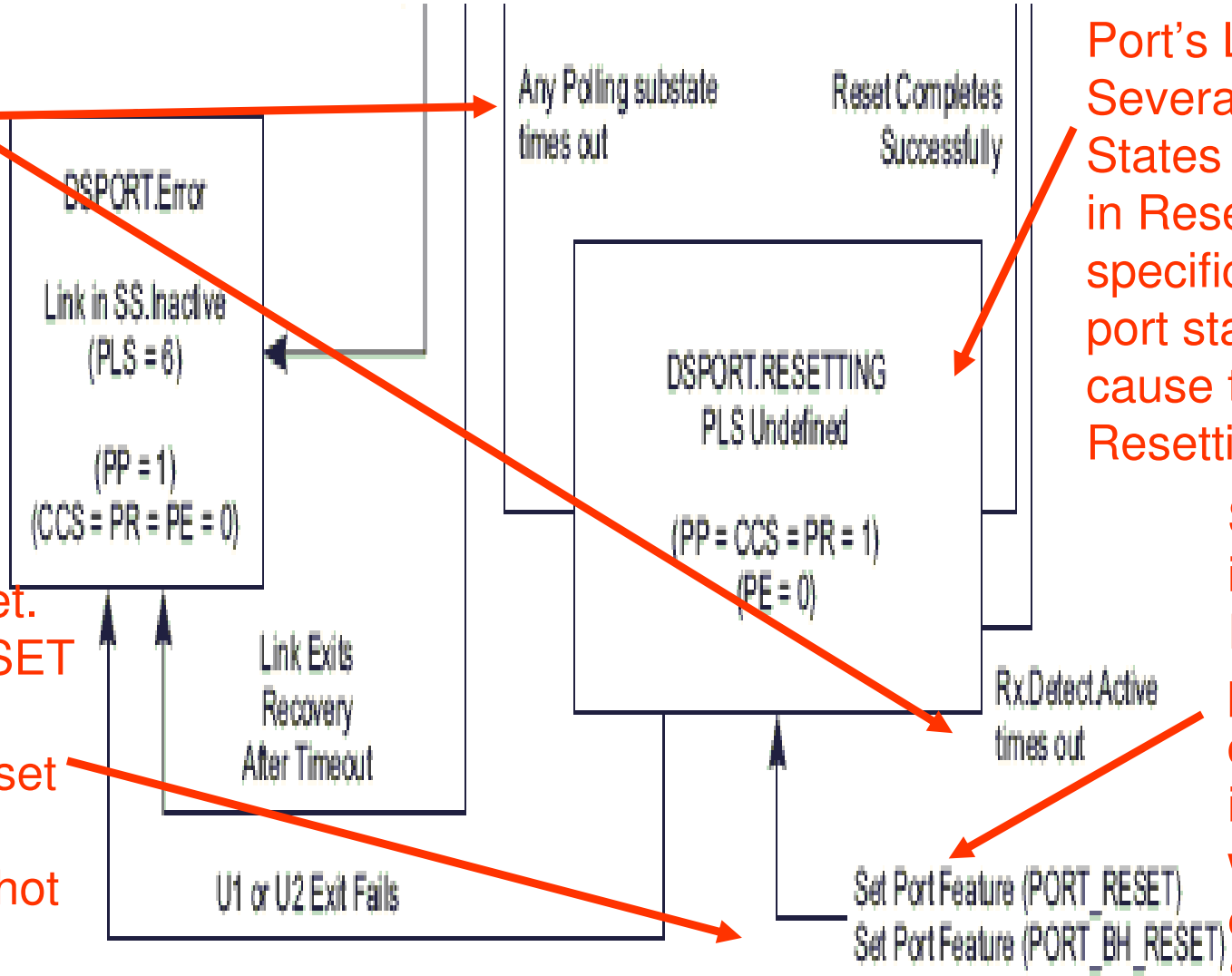
Transition From DSPORT.Training with SuperSpeed Capable Device Connected

Port Connection Status and Port Enable Set

DSPORT.Resetting

Error conditions during warm reset that cause port to exit resetting state

PORT_RESET initiates hot reset. BH_PORT_RESET initiates warm reset. Warm reset is tried automatically if hot reset fails. (port stay in resetting)



Port's Link Can Be In Several Different States While Port is in Resetting. Only specific transitions in port state machine cause the port to exit Resetting.

Software initiates Reset. A peripheral device is implicitly reset when its link enters U0. Software reset is optional.

Global Speed Changes
(SuperSpeed USB Capable Topology)

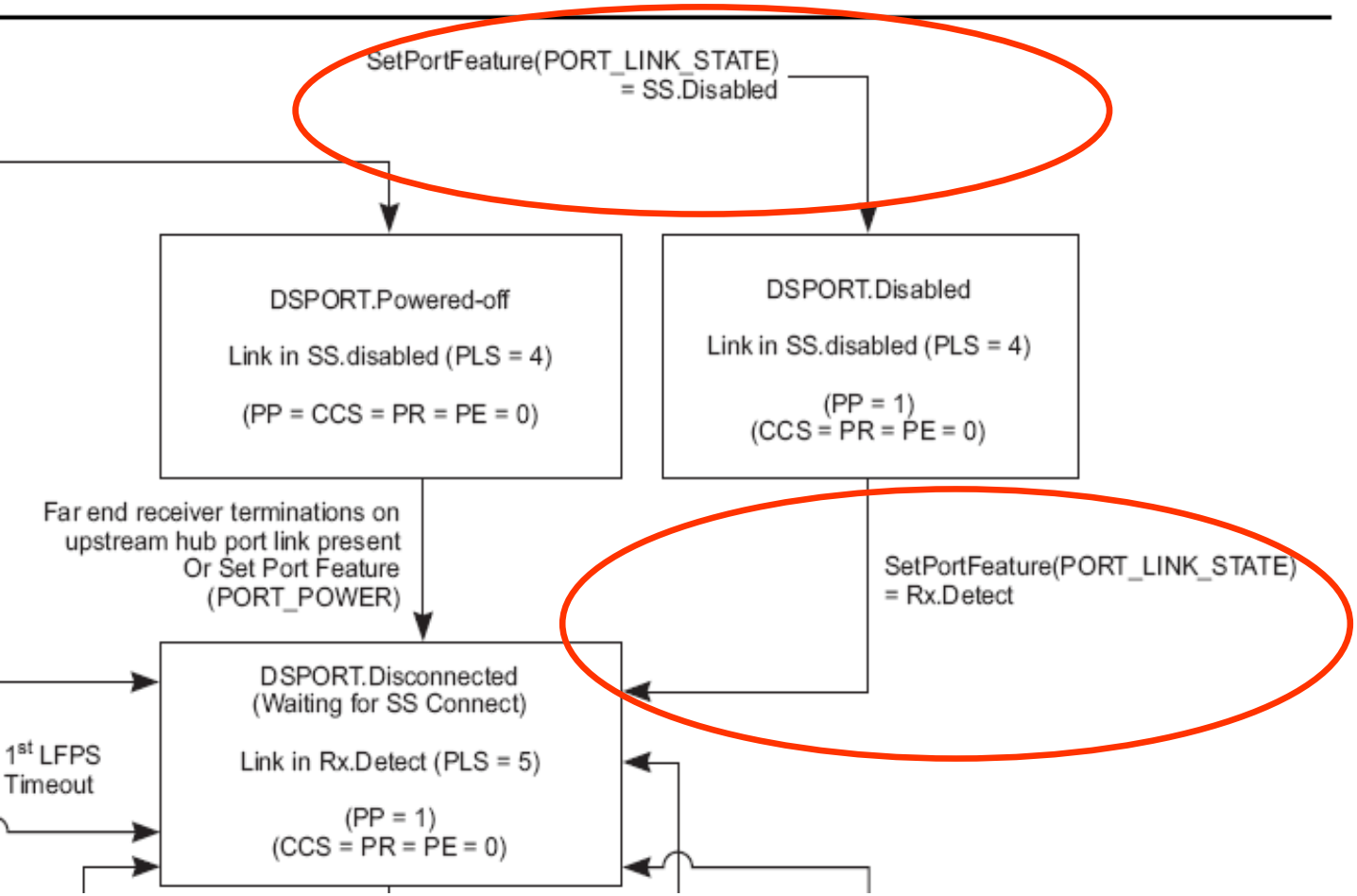
SuperSpeed USB to USB 2.0

- Host Root Port Removes SuperSpeed Terminations
- Hub(s) detect Far End termintions removed on hub upstream port and remove SuperSpeed terminations on all downstream ports
  - Hubs stay connected on USB 2.0
- Peripheral devices detect lack of far end receiver terminations and connect on USB 2.0

USB 2.0 to SuperSpeed USB

- Host Root Port Applies SuperSpeed USB Terminations
- Hub(s) detect Far End terminations on hub upstream port and apply SuperSpeed USB terminations on all downstream ports
  - Hubs stay connected on USB 2.0
- Software drives a USB 2.0 reset throughout topology
- Peripheral devices detect USB 2.0 reset and try SuperSpeed USB connection
- Peripheral devices connect successfully at SuperSpeed USB and drop USB 2.0 connection

Selective (Single Device) Speed Changes



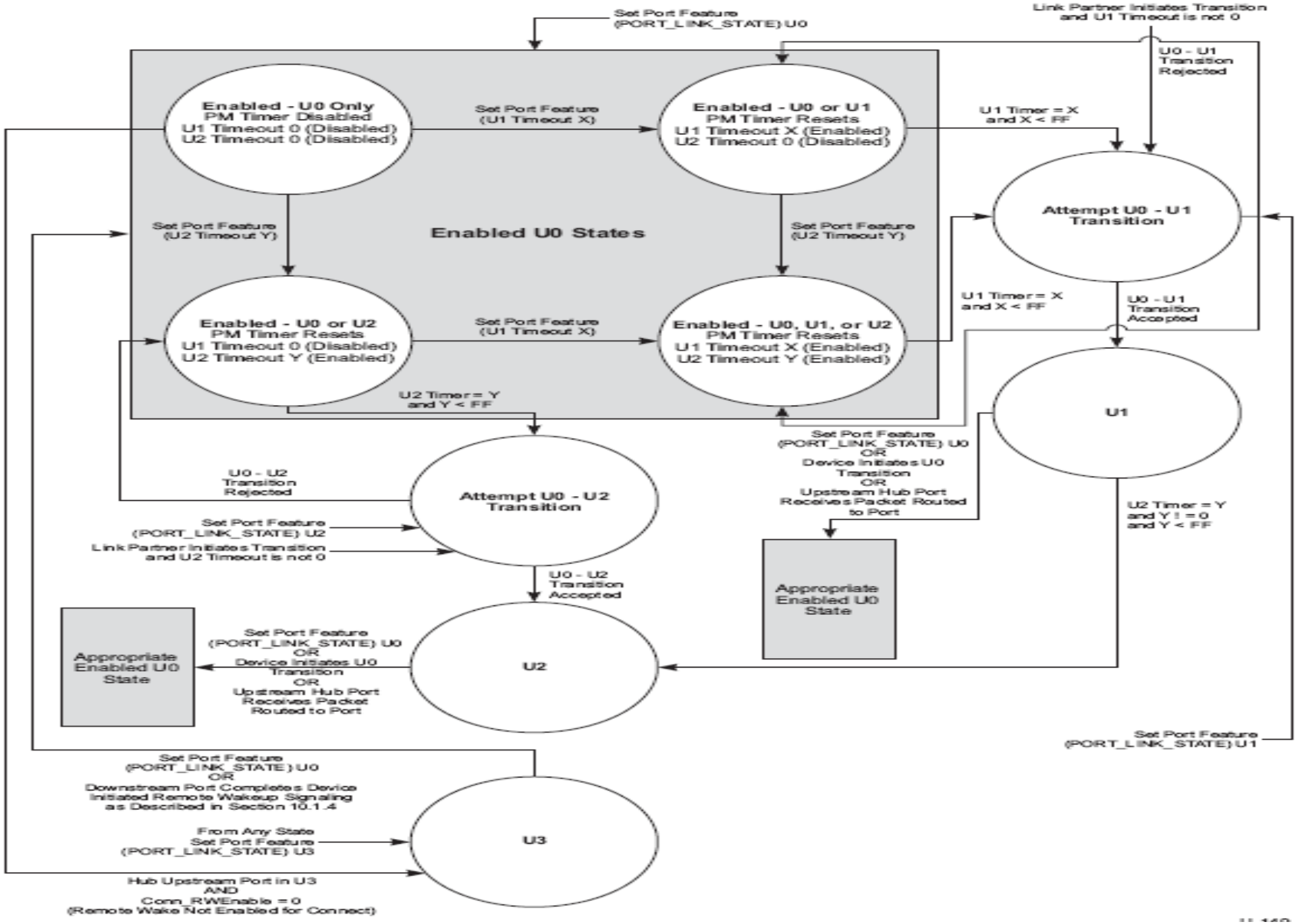
SuperSpeed USB - USB2.0

- Software issues SetPortFeature(PORT_LINK_STATE) – SS.Disabled to parent port for peripheral device
- Port removes terminations
- Peripheral device connects on USB 2.0

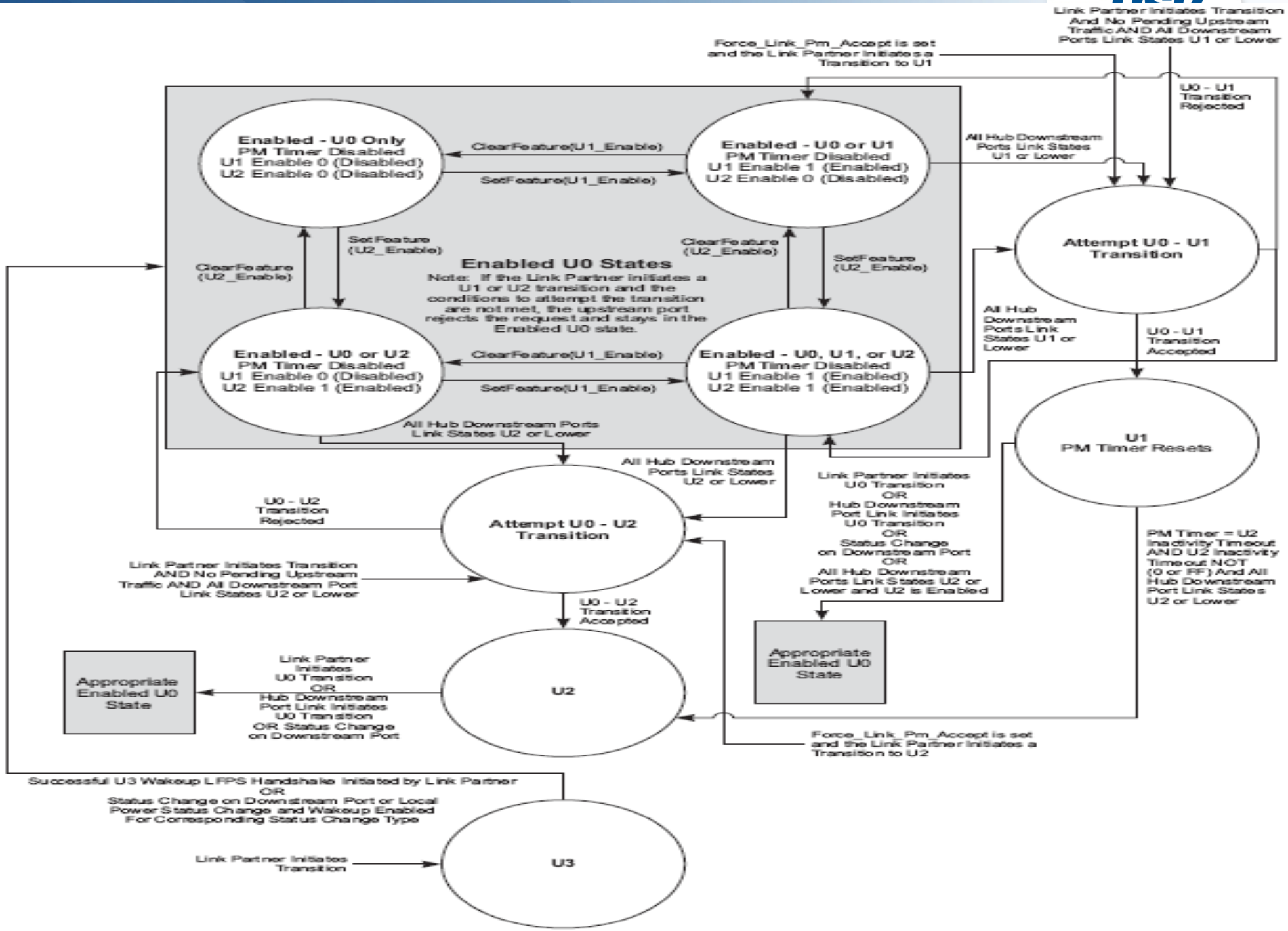
USB 2.0 – SuperSpeed USB

- Software issues SetPortFeature(PORT_LINK_STATE) – Rx.Detect to parent port for peripheral device
- Port applies terminations
- Software issues SetPortFeature(PORT_RESET) to parent port on USB 2.0
- Peripheral device connects at SuperSpeed and disconnects on USB 2.0

Downstream Port Power Management



Upstream Port Power Management



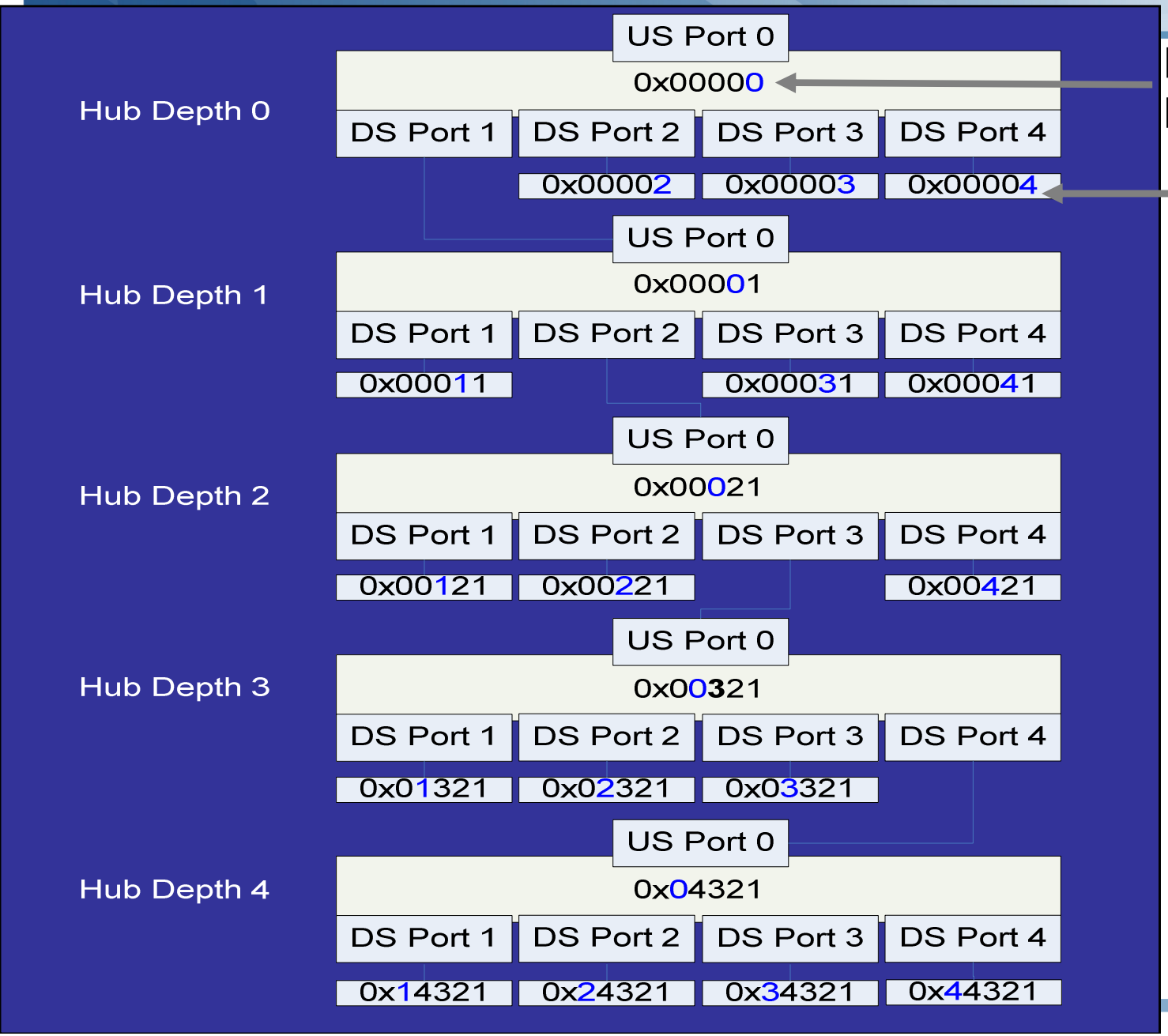
Packet Routing

- Why Route Packets?
  - Broadcasting packets to all downstream ports unnecessarily prevents links from staying in low power states
- Route String Format

19:16	15:12	11:8	7:4	3:0
Target Port at Depth 4	Target Port at Depth 3	Target Port at Depth 2	Target Port at Depth 1	Target Port at Depth 0

- Hub Depth
  - Software issues a request to a hub to program the hub depth
  - A hub ignores route strings before it enters the configured state
  - When configured, the hub uses the target port in the route string for the depth value the hub received:
    - Target Port = 0. Packet is routed to the hub itself
    - Target Port = 1-n. Packet is routed to a downstream port

Packet Routing Example

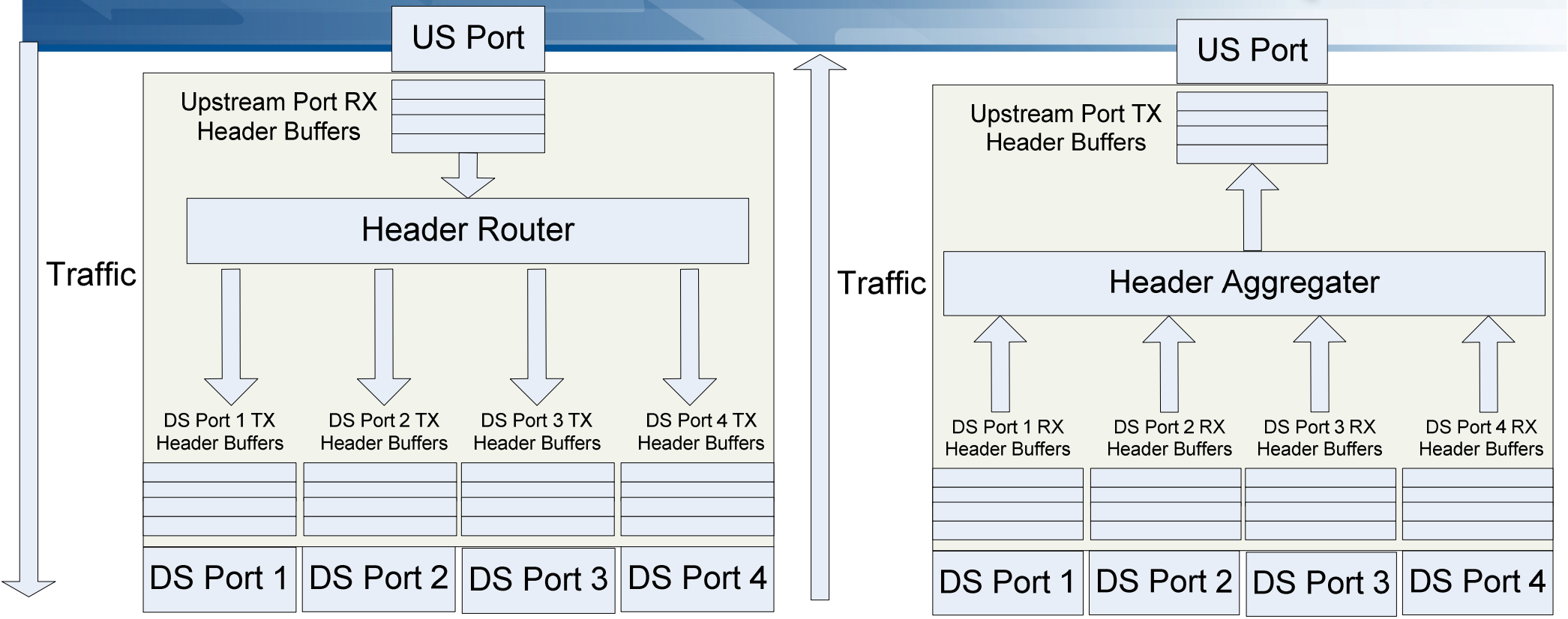


Route String To Route Packets To Hub

Route String To Route Packet To Device Connected to Port 4

Hub depth sent to hub via Set Hub Depth request. Hub routes all packets with route string after it is configured.

Hub Header Buffer Architecture

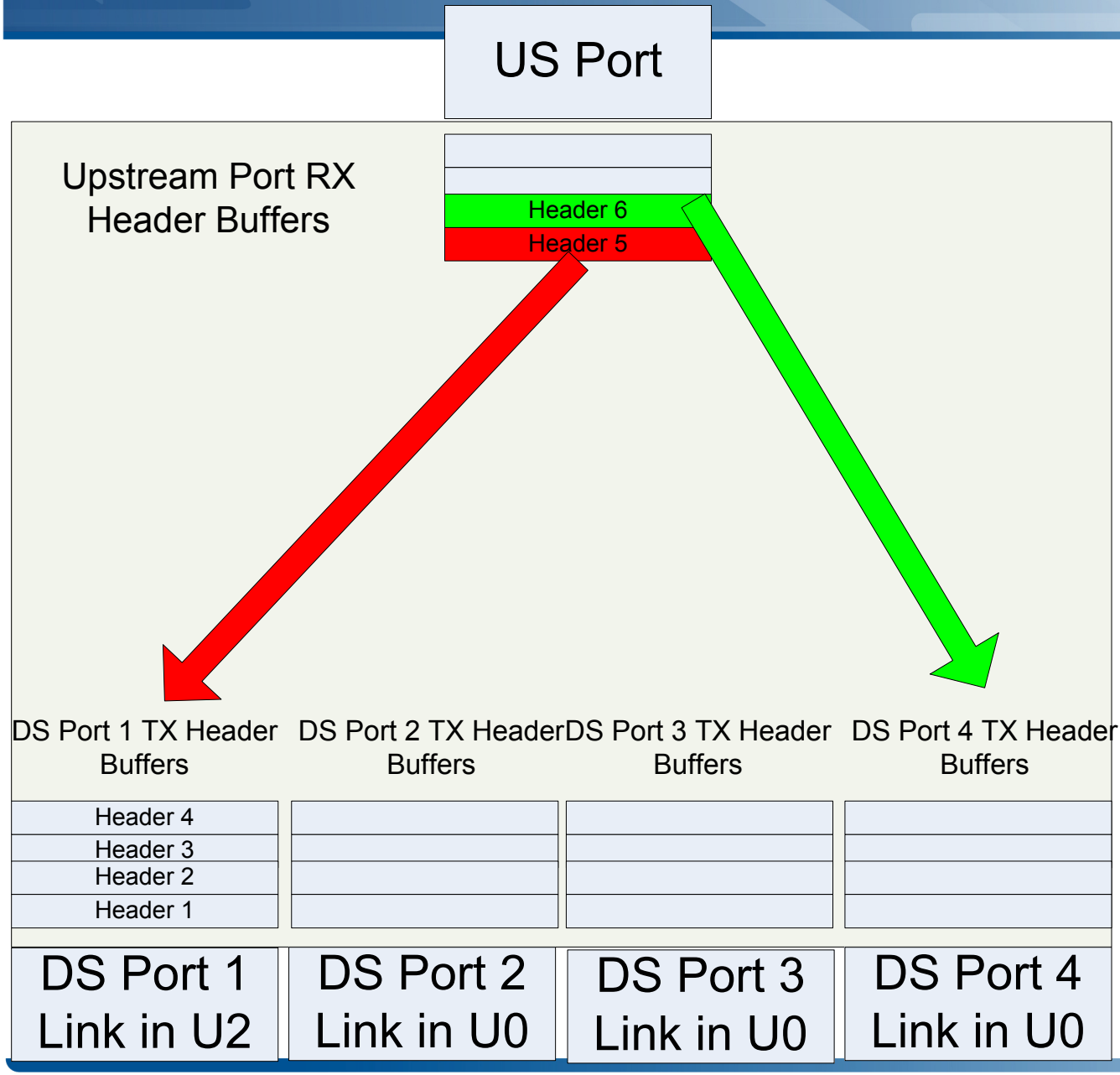


- Hubs have store and forward model for headers
  - Link level retries of headers for high header reliability
  - Allow device driven power management
- Specific architecture shown is optional
  - Spec captures a set of functional requirements

Functional Header Buffer Requirements

- A hub starting with all header packet buffers empty shall be able to receive at least eight header packets directed to the same downstream port that is not in U0 before its upstream port runs out of header packet flow control credits
- A hub that receives a header packet on its upstream port that is routed to a downstream port shall immediately route the header packet to the appropriate downstream port header packet buffer (if space in that buffer is available) regardless of the state of any other downstream port header packet buffers or the state of the upstream port Rx header packet buffer
- A hub starting with all header packet buffers empty shall be able to receive at least eight header packets on the same downstream port directed for upstream transmission when the upstream port is not in U0
- Header packets transmitted by a downstream port shall be transmitted in the order they were received on the upstream port
- Header packets transmitted by an upstream port from the same downstream port shall be transmitted in the order they were received on that downstream port

Downstream Header Example

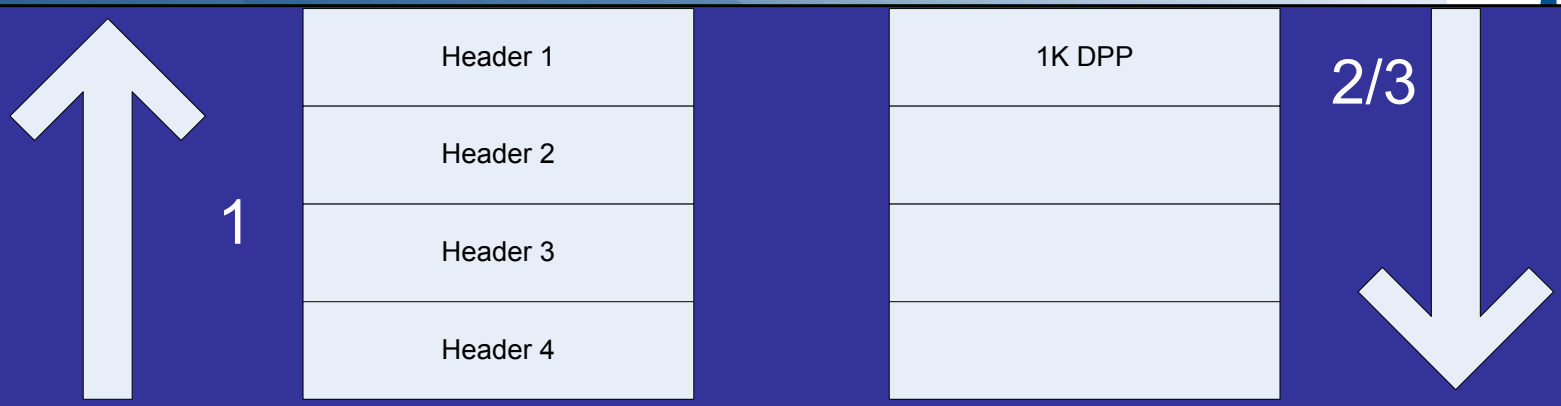


- If the hub receives many packets routed to the same downstream port with a link in a low power link state it will run out of buffers for this port
- As long as the hub still has credits/buffers to receive headers on its upstream port, it must still forward headers to other downstream ports without delay

Received Header Flow – All Links in U0

- Header received on upstream port
- Hub calculates CRC-16 and Link Control Word CRC-5 and checks route string
  - CRC valid, header routed to downstream port
  - LGOOD_n link command queued for transmission on upstream port
- Header queued for transmission on downstream port
- Header removed from upstream port receive header buffer
- LCRD_x link command queued for transmission on upstream port

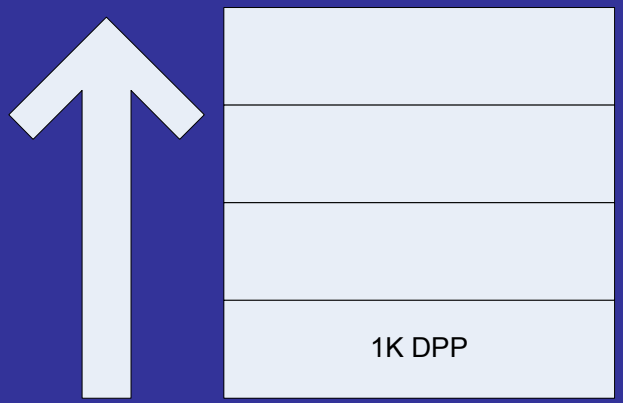
Data Buffering Needed - Example



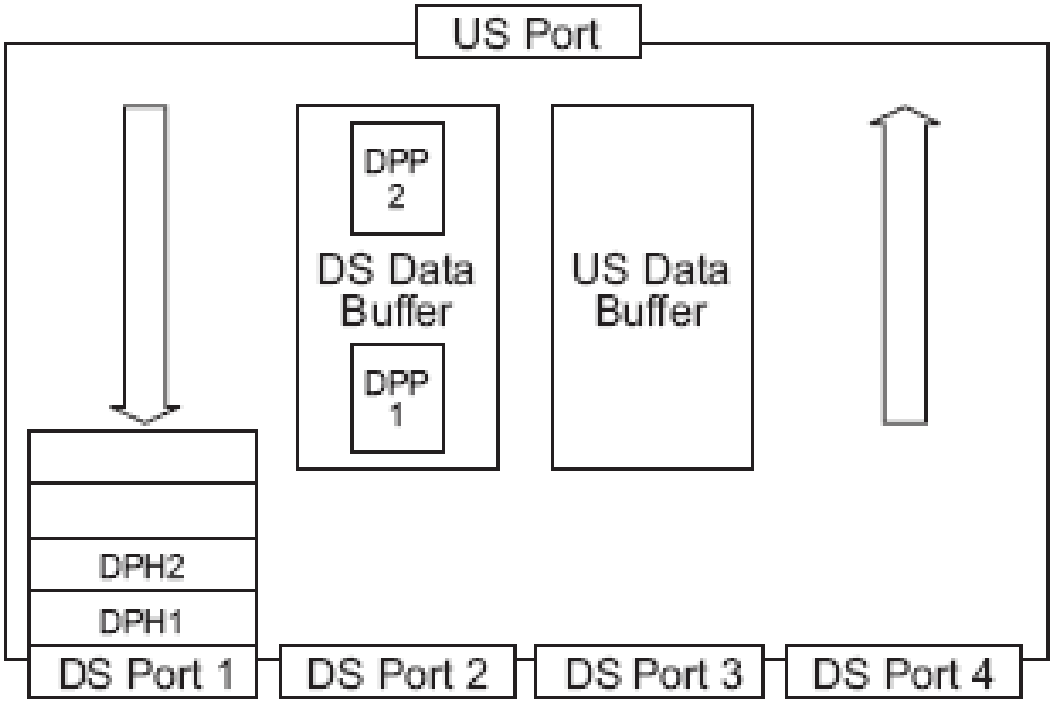
1. 4 Headers are received on hub downstream ports in close succession. Headers could be ACK, PING Response, EP Ready, etc.

2/3. Hub upstream port starts receiving 1k Data packet before first header reaches device connected to the hub upstream port.

2/3. Hub starts receiving 1k Data packet + header on a downstream port. The hub will need to buffer most of this packet. It will not be able to transmit the packet upstream until it receives acknowledges for Headers 1-4 which are stuck behind the 1k packet the hub is receiving on the upstream port.



Hub Data Buffer Model



- Independent Data Buffers US/DS
  - ~1k each
  - Support concurrent IN/OUT
- Data can be dropped in extreme cases – End to end retries recover
  - BER
  - Link in Low Power State
  - Asynchronous Message Storm + Large Packets

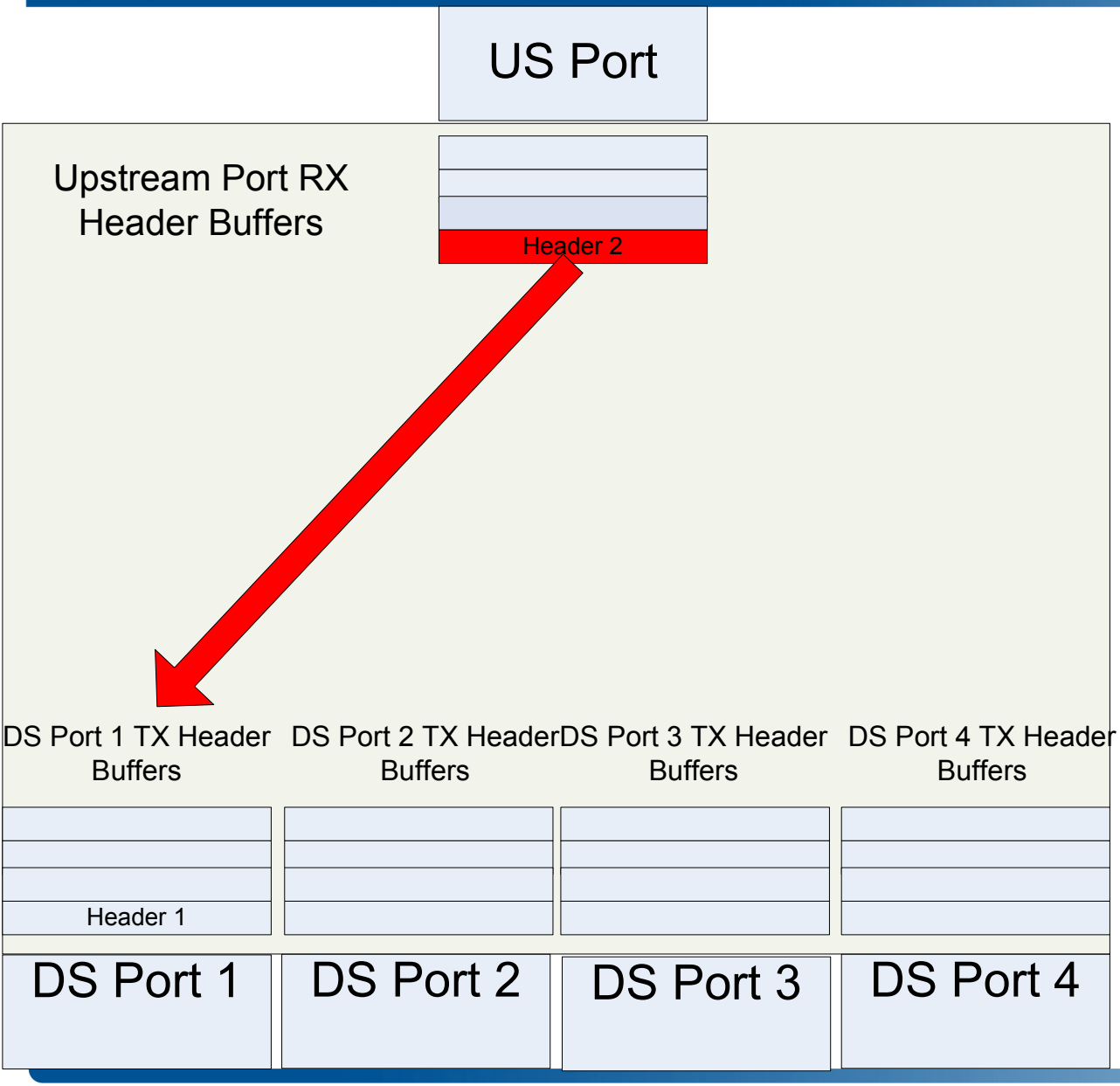
Forwarded Packet Insertion Rules for Hubs

- Downstream Traffic Minimum Spacing
  - DPP – DPH – no space. Hub must preserve
  - Average spacing needed across other packet gaps
    - Prevent underflow/overflow due to SSC
    - Will naturally happen with protocol
- Upstream Traffic Minimum Spacing
  - DPH to DPP – no space. Hub must preserve
  - Average spacing needed across other packet gaps
    - Prevent underflow/overflow due to SSC
    - Will naturally happen with protocol
- Hub Insertion Rules
  - Downstream
    - Packets with isochronous timestamps must have minimum propagation delay variation
      - Minimum variation allowed with no other traffic
      - Host scheduling can ensure minimum impact to timestamps
  - Upstream
    - Insert with following priority
      - Link Commands
      - Headers

Deferred Packet Sequence

- Hub receives a header on upstream port
- Hub computes CRC-16 and Flow Control Word CRC-5 and checks the route string. Header is routed to a downstream port in a low power link state
- Hub performs necessary processing
  - Hub initiates return to U0 on target downstream port
  - Hub modifies header to set deferred flag, set the hub depth, and computes new link control word checksum. Modified header is queued for transmission on downstream port
  - Hub queues deferred header for transmission on upstream port
  - Any data associated with the header is discarded
- After downstream link returns to U0
  - Modified header delivered to device
  - Device sends EP Ready when ready to transfer data
  - Host retries deferred transaction

Delayed Packets



- The hub receives a header (header 2) with a valid CRC and CRC-5 that is routed to a downstream port with current transmit activity.
- If the header would be sent later than in the case where the downstream port was idle – the hub must change the Link Control word and update the checksum to indicate that the header was delayed. This allows isochronous endpoints to ignore timestamps that were delayed.

Maximum Propagation Delay and Variation

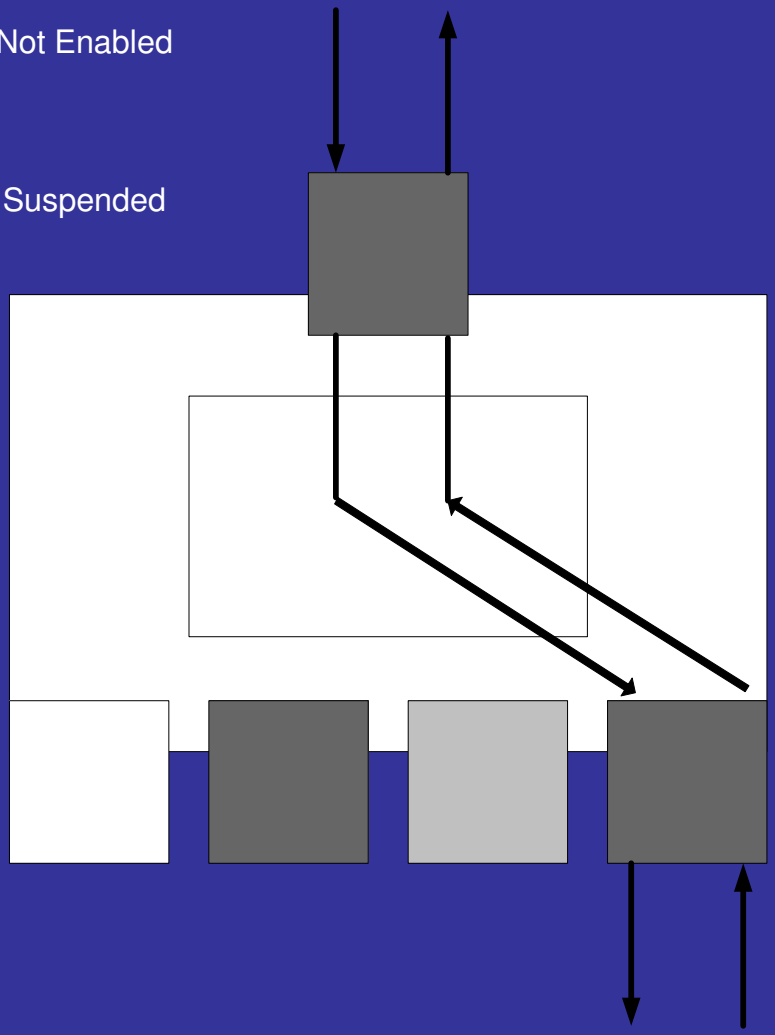
- Average propagation delay sent to device through Set Isochronous Delay Request
  - Will vary with hub depth
  - Allows devices at different hub depths to account for differences in calculated Microframe boundaries
- Delay variation important for rate synchronization through timestamps
  - Hubs not allowed to impact delay variation through link command insertion
  - Headers impacted by other traffic (retries, etc) are marked delayed and can be ignored by devices for synchronization
  - Hubs required to complete header processing and routing within a small time window
  - Overall propagation delay variation not expected to exceed 200 ns

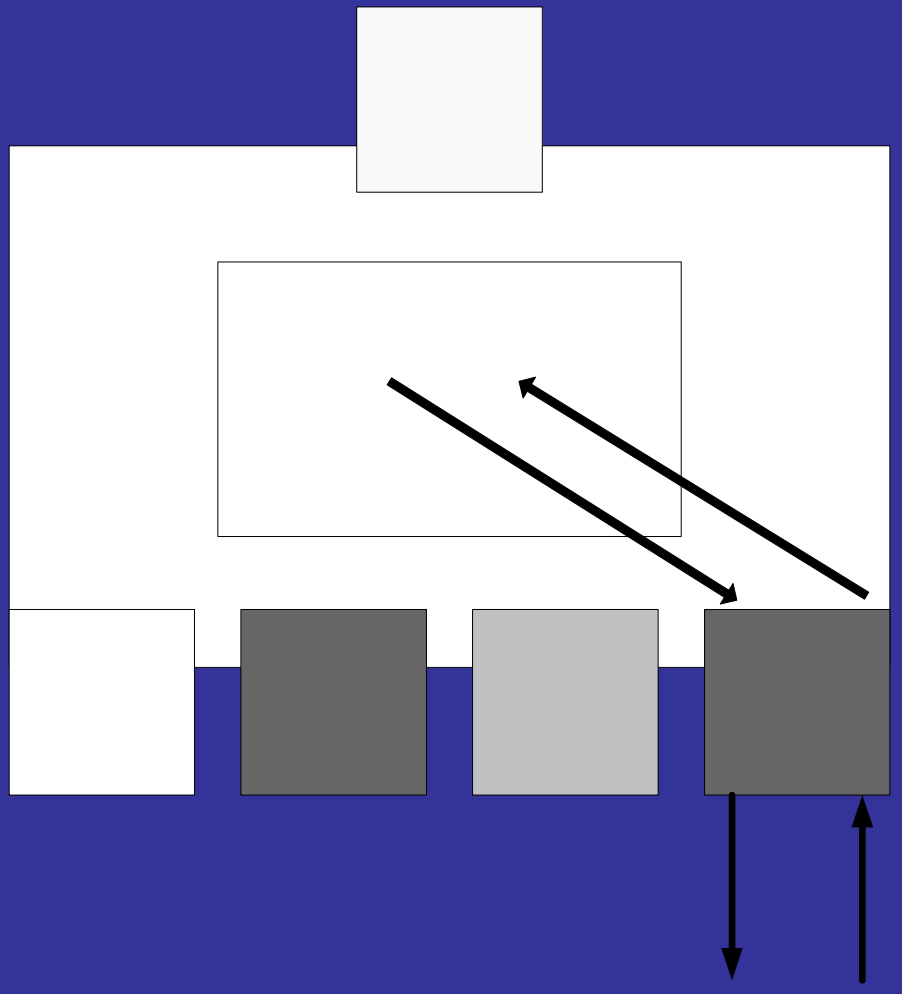
Hubs and Resume Signaling

Enabled

Not Enabled

Suspended





Summary

- USB 3.0 hub provides bus expansion
  - Up to 5 levels of hubs
  - Up to 127 devices
- USB 3.0 hub architecture includes logical USB 2.0 and SuperSpeed USB hubs
- SuperSpeed USB hubs route packets
  - No broadcast packets on SuperSpeed bus
  - Improved Power Efficiency
- SuperSpeed USB hubs include buffering for forwarded header and data traffic
- SuperSpeed USB hubs provide deferred notifications when packets are routed to power managed links
- SuperSpeed USB hubs provide deterministic delivery of isochronous timestamps