SuperSpeed USB Developers Conference

San Jose, California
November 17-18, 2008

Isochronous Protocol

Dan Froelich
USB 3.0 Isochronous Workgroup Chair, Intel
Peter Teng
NEC Electronics America

Agenda

- USB 2.0 Isochronous Review
- USB 3.0 Isochronous Features
- Synchronization
- Isochronous OUT Overview
- Isochronous IN Overview
- Isochronous Framework
  - Service Interval
  - Maximum Packet Size
- Bus Interval Adjustment Mechanism
- Isochronous and Power Management
- Summary

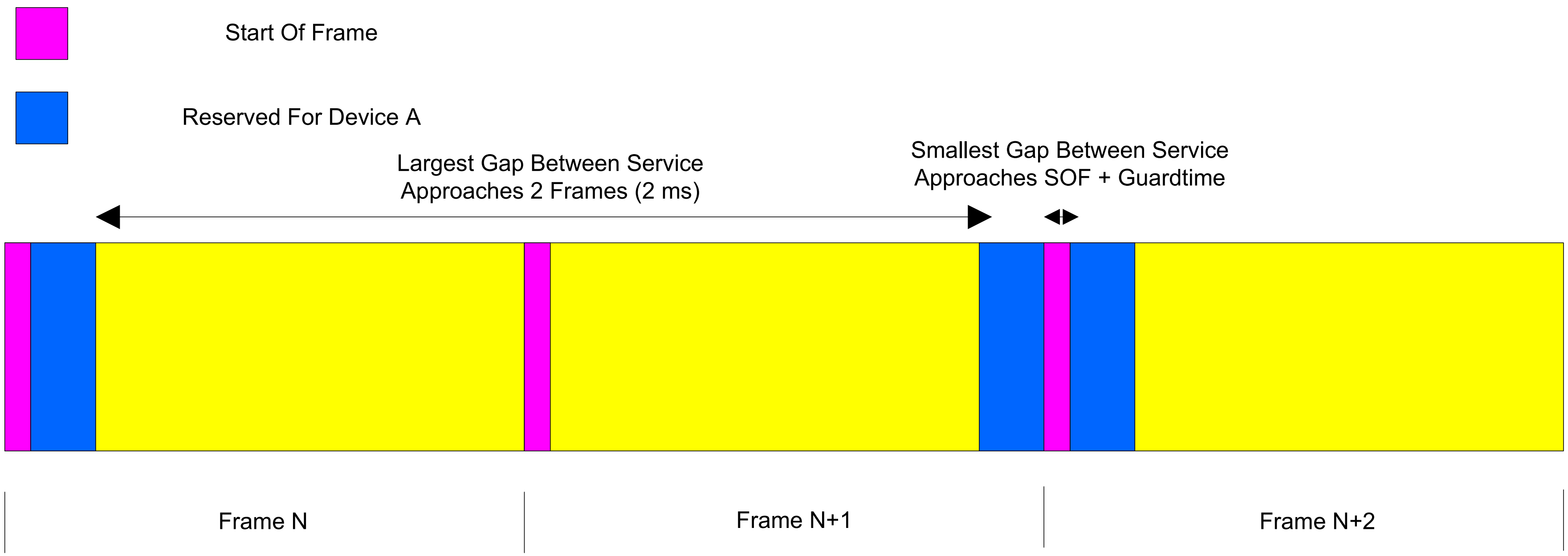
Isochronous Assumptions

- Devices need to be able to have the same notion of time as host to 200 nanosecond accuracy
  - Generally accepted requirement for high end professional audio ~ 100 ns
- Host should be allowed to provide service anywhere each service interval
  - No strict requirement on order or location of ISO packets, etc.
- No Retries
- Requiring additional device buffering or stream latency is bad if it can be reasonably avoided
- Isoch IN
  - Isochronous IN endpoint does not need to know when packets are dropped
  - Host does need to keep track of when packets are dropped
  - Packets must not be sent too early or too late
- Isoch OUT
  - Host does not need to know when packets are dropped for Isoch OUT
  - Endpoint/device needs to know when packets are dropped
  - Packets must not be sent too early or too late

USB 2.0 Isochronous Feature Summary

- Guaranteed Service Attempt Every Service Interval
  - Reserved Bandwidth
  - Service Interval Bound (1 service interval ± 1 service interval)
  - Typical Service Interval Has Little Jitter
- Reliability
  - Bit Error of 10^{-12} Or Better (PER 10^{-9})
  - No Retries Or Handshaking Needed In Protocol
- Minimal Buffering
  - Typical Implementations Require 2 Service Intervals Of Buffering
- Host Timing Information
  - SOFs Occur At Regular 1 ms/125 uS Intervals With Small Jitter (10's of Ns)
  - SOFs Can Be Used As Device Clock.
- Data Delivery Time
  - Data Is Sent In Specified Frames/Microframes
  - Data Is Not Too Early or Too Late

Wired USB 2.0 Isochronous Model



- The host may access an endpoint anytime during the service interval
- Isochronous endpoint must not assume any timing relationship between transactions

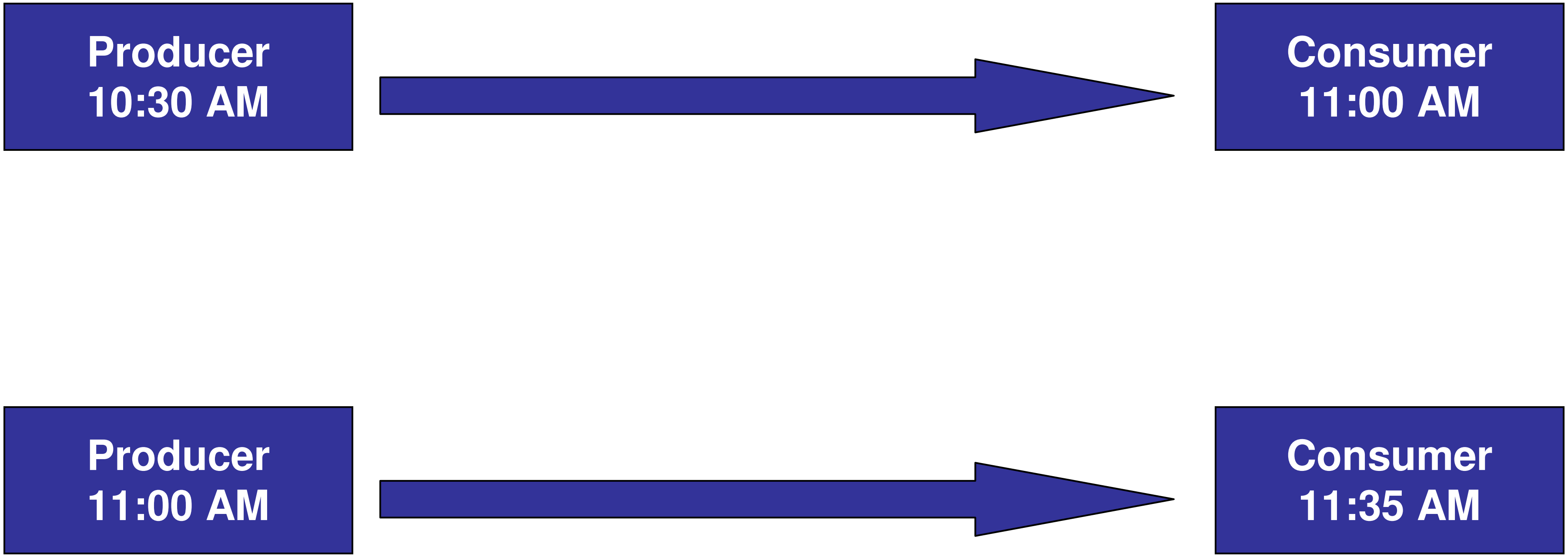
USB 3.0 Isochronous Goals

- Preserve Key Characteristics of USB 2.0 Isochronous Protocol
  - Support Applications That Can Tolerate A Small Percentage of Packet Drops and Have Deadlines For Packets
    - Real Time Video
- Improve Power Efficiency of Isochronous Protocol
- Provide software backwards compatibility with existing isochronous drivers

Retries

- No Retries
  - BER target is low (~ E-12 or better)
  - Isochronous protocols of interest can tolerate packet error rates of E-6 or better
  - Very simple protocol (no acknowledgements)
  - Must have some other mechanism to make sure you don't get too far ahead (Iso out) or behind (Iso In)
- Retries
  - Can increase effective PER (WUSB)
  - Adds protocol complexity
    - How many retries?
    - When do you retry?
    - If you stop retrying without success – how is it handled?
- **USB 3.0 Isochronous – No Retries**

Synchronization – Problem Statement



Producer And Consumer Have To Have A Way To Set Clocks That Drive Production And Consumption At The Same Rate (Time) Or Buffers Overflow/Underflow

Isochronous Timestamp Packet

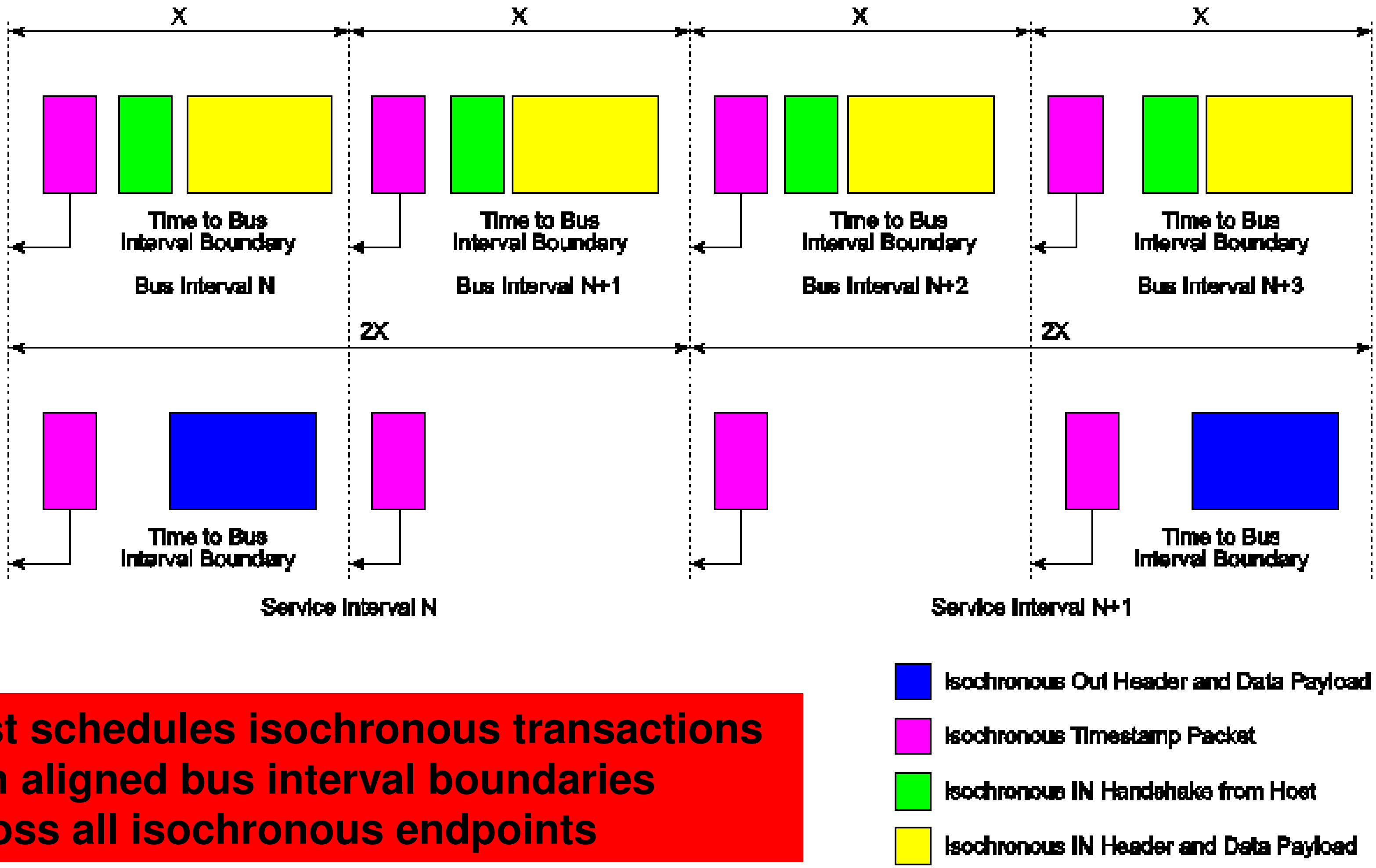
- Isochronous Timestamp Packet (ITP) is used to multicast timestamp information from the host to all active devices. ITPs are used to provide host timing information to devices for synchronization.
- ITPs carry no addressing or routing information and are multicast by hubs to all of their downstream ports with links in the U0 state
  - ITP forwarding in Hub does not return a port link to U0
  - ITPs are not forwarded by hub downstream ports not in U0
- Host transmits an ITP in every bus interval (125µs) within tTimestampWindow (8µs) from a bus interval boundary
  - This allows full bus utilization in host transaction scheduling
  - Host does not transmit an ITP if the host port link is not in U0
- Device stay in U0 around bus interval boundary when ITP is needed
- Device may opt to drop received ITP if delayed flag is set
  - Delayed flag is set upon link retry

Fields in Isochronous Timestamp Packet

Bus interval counter (14 bits**)**. The 1/8 of a millisecond counter.	
Delta (13 bits**)**. The time delta from the start of the current ITP packet to the previous bus interval boundary.	

- Timestamp information are generated from non-spread clock
- Timestamps must be accurate to host time at start of ITP transmission +/- 25 ns
- Timestamp Fields
  - Bus Interval counter - 1/8 millisecond counter increments every 125µs. Counter rollover to 0 when x3FFF is reached
  - Delta – Delta time from the beginning of last bus interval boundary to the beginning of the ITP packet. This is specified in tIsochTimestampGranularity (8 HS bit time, 16.666ns)

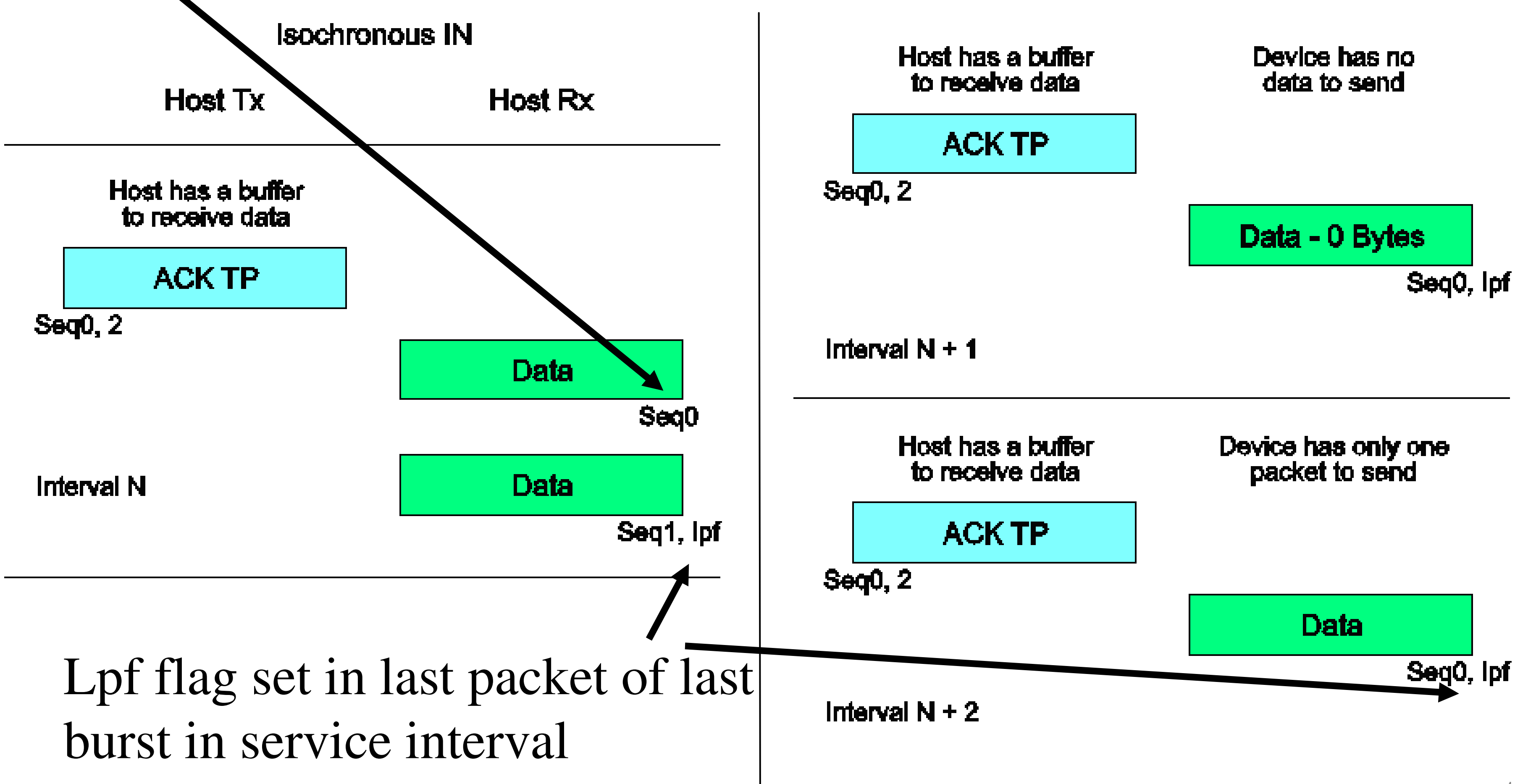
Service Interval Alignment



Host schedules isochronous transactions with aligned bus interval boundaries across all isochronous endpoints

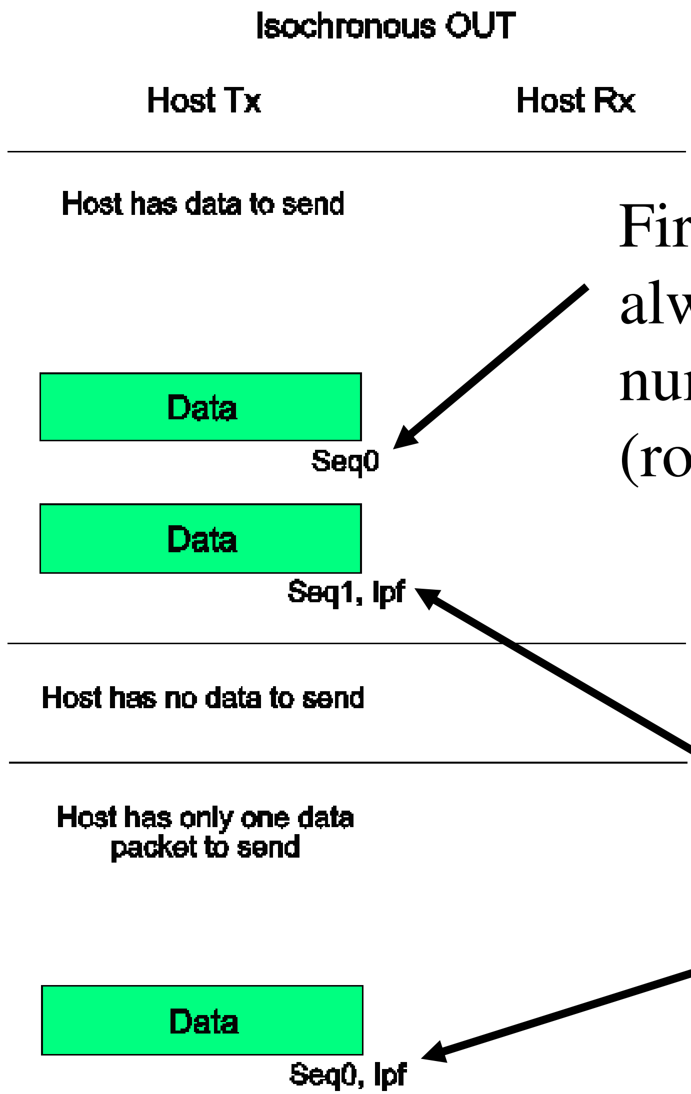
Isochronous IN Sequence

First packet transmitted in service interval always has sequence number 0. Sequence number increases during service interval (rollover after 31)



Lpf flag set in last packet of last burst in service interval

Isochronous OUT Sequence



First packet transmitted in service interval always has sequence number 0. Sequence number increases during service interval (rollover after 31)

Lpf flag set in last packet of last burst in service interval

Isochronous Flow Control

- No flow control needed for Isochronous Out
  - Device must provide peak request buffering anywhere in the service interval, as host may send all data for the interval in a single burst
- No flow control needed for Isochronous IN
  - Host must be able to receive peak requested data once per interval
  - Device must able to send all data for the interval in a single burst

Isochronous Endpoint Bandwidth Request

- Service intervals are the same as USB 2.0.
  - 125µs * $2^{(bInterval-1)}$, bInterval = 1 to 16. Possible intervals are 125µs, 250µs, 512µs, 1024µs, 2048µs and up to 4s
- Maximum of 128MB/s for a single burst per service interval
- There can be up to 3 (see mult in bmAttributes of SuperSpeed Endpoint Companion) bursts per service interval – allowing up to 384 MB/s
- Maximum packet size 1KB. All packets transmitted in each service interval must be of maximum packet size, except the last packet
- wMaxPacketSize must be 1024 unless there is only 1 packet per service interval
- Max bandwidth consumed in a service interval is reported in wBytesPerInterval field in endpoint companion descriptor

Host's Bursting Considerations

- A host may consider the following factors when performing Isochronous Bursting:
  - Availability of data buffer
  - Latency of fetching data from/to system memory
  - Volume of data to be transferred to endpoint
  - Device characteristics, reported thru descriptors
  - Non-related transactions that are already scheduled

Sample Bursting Example

- For example, if host has 11 packets to send to an isochronous endpoint during a service interval, following are possible combinations a host may generate:
  - 1 burst of 11 packets
  - 1 burst of 8 and 1 burst of 3
  - 1 burst of 4, 1 burst of 4 and 1 burst of 3
  - 5 bursts of 2, and 1 burst of 1

Synchronization between Isochronous Devices

- Synchronization is possible with following information:
  - Host receives delay information (wHubDelay) in unit of ns is reported in each hub of the hierarchy. wHubDelay reports the processing delay in hub when packet is forwarded from upstream port to downstream port
  - Host calculates delay in every path and uses Set Isochronous Delay to inform the device on the time delay (0 to 65535ns) starting from a packet transmitted from host to received by device
  - Devices subtracts Isochronous Delay from the timestamp information received in Isochronous Timestamp Packet to get the host's notion of time
  - Device may adjust its processing of Isochronous data, relative to host's time for synchronization with other Isochronous devices. This allows group of devices to perform synchronized activities. For example, HD speakers located in different places of hierarchy

USB 3.0 Isochronous Enhancements

- USB 3.0 allows Isochronous capable devices to enter the lower power USB link state between service intervals
  - A SuperSpeed host must transmit a PING packet to the targeted isochronous device before service interval, to transition the path back to the active link state before initiating the isochronous transfer
- USB 3.0 allows small adjustments to 125µs service interval
  - A device may change the interval depending on the application's requirement
  - USB 3.0 devices may send a Bus Interval Adjustment Message to the host to adjust its 125µs bus interval up to ± 13.333µs

Bus Interval Adjustment Mechanism

- Some devices need to synchronize to an external clock (not the USB host clock)
  - This synchronization may be needed across multiple devices
  - Software only solutions to this problem are difficult
- Mechanism is provided for a device to control timestamp clock rate
  - Supports audio applications needing synchronization between multiple devices

Bus Interval Adjustment Mechanism (2)

- Device Notification to ask for small adjustments to the bus interval
  - Request unit is 4.069ps every bus interval
    - Adjustment is 8 HS bit times (16ns) over 1 second or ~(1/60ppm)
  - Request is for increment/decrement of N units, relative to the current bus interval
- Host must count clock ticks and add or remove 8 HS bit times from a bus interval at the appropriate count
  - Does not require changes to the underlying clock

Bus Interval Adjustment Mechanism (3)

- Device must not send additional bus interval adjustment request until it has waited long enough to accurately observe the effect of the previous bus interval adjustment request on the timestamp value in subsequent ITP
- Device must not make a single request for more than ±4096 units
- Host honors requests from first device. Once request is received, Bus Interval Adjustment Control field in subsequent ITPs is set to the address of the requesting device

Bus Interval Adjustment Mechanism (4)

- One unit bus interval adjustment (BusIntervalAdjustmentGranularity) requires the host to adjust bus interval timer by one 60MHz clock period (8 HS bit times) every 4096 bus intervals
  - The host can make the adjustment in a single bus interval with the rest of 4095 bus interval unadjusted. The bus interval adjustment is averaged over long period of time.
- Host must make adjustments in evenly distributed intervals. Following rules must be obeyed:
  - Difference between the number of eight HS bit time adjustments made in any bus interval shall not be greater than one
  - The distance in bus intervals between consecutive maximum adjustment bus intervals shall not be varied by more than one bus interval

Bus Interval Adjustment Mechanism (5)

- One sample implementation for bus interval adjustment in host
  - 16 bit adder is added with requested adjustment every bus interval
  - Bit[15:12] (in granularity of 60MHz, 16.666ns) of adder is added to Delta Time (bit 26:14) in Isochronous Timestamp Packet (ITP)
  - And once the ITP is transmitted, the upper 3 bits are cleared
  - Implementer may choose to transmit ITP with Bit[15:12] ticks earlier instead, and leave the Delta Time as zero (or solely used for transmit latency)

Bus Interval Adjustment Mechanism (6)

- Example operation 1:
  With the requested adjustment of 10 (all values are presented in decimal),
  - In bus interval 0, the adder is 10. Timestamp value in Timestamp packet is 0, 0. First number is the bus interval counter (bit 13:0), the second number is Delta Time (bit 26:14).
  - In bus interval 1, the adder is 20. Timestamp value in Timestamp packet is 1, 0
  - In bus interval 2, the adder is 30. Timestamp value in Timestamp packet is 2, 0
    ...
  - In bus interval 407, the adder is 4080. Timestamp value in Timestamp packet is 407, 0.
  - In bus interval 408, the adder is 4090. Timestamp value in Timestamp packet is 408, 0.
  - In bus interval 409, the adder is 4100. Bits[15:12] of the adder is 1. Lower bits[11:0] of the adder is 4. Timestamp value in ITP is 409, 1 (added with bits[15:12] of adder). After Timestamp value is updated, upper bits of the adder are cleared.
  - In bus interval 410, the adder is 14. Timestamp value in Timestamp packet is 410, 0

Bus Interval Adjustment Mechanism (7)

- Example operation 2:
  With the requested adjustment is 32000:
  - In bus interval 0, the adder is 32000 ('h7D00). Upper bits[15:12] of the adder is 7. Lower bits[11:0] of the adder is 3328 ('hD00). Timestamp value in Timestamp packet is 0, 7 (added with upper bits[15:12] of adder). After Timestamp value is updated, upper bits of the adder are cleared.
  - In bus interval 1, the adder is 35328 (3328+32000 or 'h8A00). Upper bits[15:12] of the adder is 8. Lower bits[11:0] of the adder is 2560 ('hA00). Timestamp value in ITP packet is 1, 8 (added with upper bits[15:12] of adder). After Timestamp value is updated, upper bits of the adder are cleared.

Isochronous PING scenario

- Software may enable appropriate link states (U1 or U2) based on service interval and throughput.
- Device may go into appropriate lower power link state when idle, ie after all Isochronous transactions are completed in the service interval, and ample time before next service interval accommodating U1/U2 exit time.
- Host sends PING to Isochronous device in advance of first transaction in the service interval. This transitions the path between host and device back to U0. Host assumes worst case exit latency in calculating launch time of PING.
- Intermediate hub (if any) may defer the PING, as the downstream link is in U1/U2. Hub forwards PING once the Link is in U0.
- Device responds with PING_RESPONSE when PING is received, after U0 is returned.
- Host schedules Isochronous transactions and transmits to device. Device is required to stay in U0 after PING_RESPONSE is transmitted, until tPingTimeout time (2*Service Interval).

Host's Considerations in PING Transmission

- Latencies presented in hubs along path
  - U1 Exit Latency (bU1ExitLatency) if U1 is enabled and U2 is disabled
  - U2 Exit Latency (bU2ExitLatency) if U2 is enabled
  - Hub Header decoding delay (bHubHdrDecLat)
  - wHubDelay – Hub forwarding delay
- Latencies presented in targeted device
  - U1 Exit Latency (bU1ExitLatency) if U1 is enabled and U2 is disabled
  - U2 Exit Latency (bU2ExitLatency) if U2 is enabled
  - tPingResponse – from Device reception of a PING to initiating a PING_RESPONSE
- Latencies presented in Host
  - U1 Exit Latency (bU1ExitLatency) if link is in U1
  - U2 Exit Latency (bU2ExitLatency) if link is in U2
  - Isochronous Transaction Scheduling time – from PING_RESPONSE received to the time Isochronous transaction may be transmitted

Summary

- USB 3.0 Isochronous Protocol Preserves Key Characteristics of USB 2.0 Isochronous Protocol
  - No Retries. Simple Protocol
- Timestamps No Longer Broadcast to Improve Power Efficiency
- PING Mechanism To Allow Low Power Link States For Isochronous Devices
- Software Backwards Compatibility
- Maximum number of data transferred in one service interval is increased to 48K bytes.
- Mechanism for Device to Synchronize to External Clock