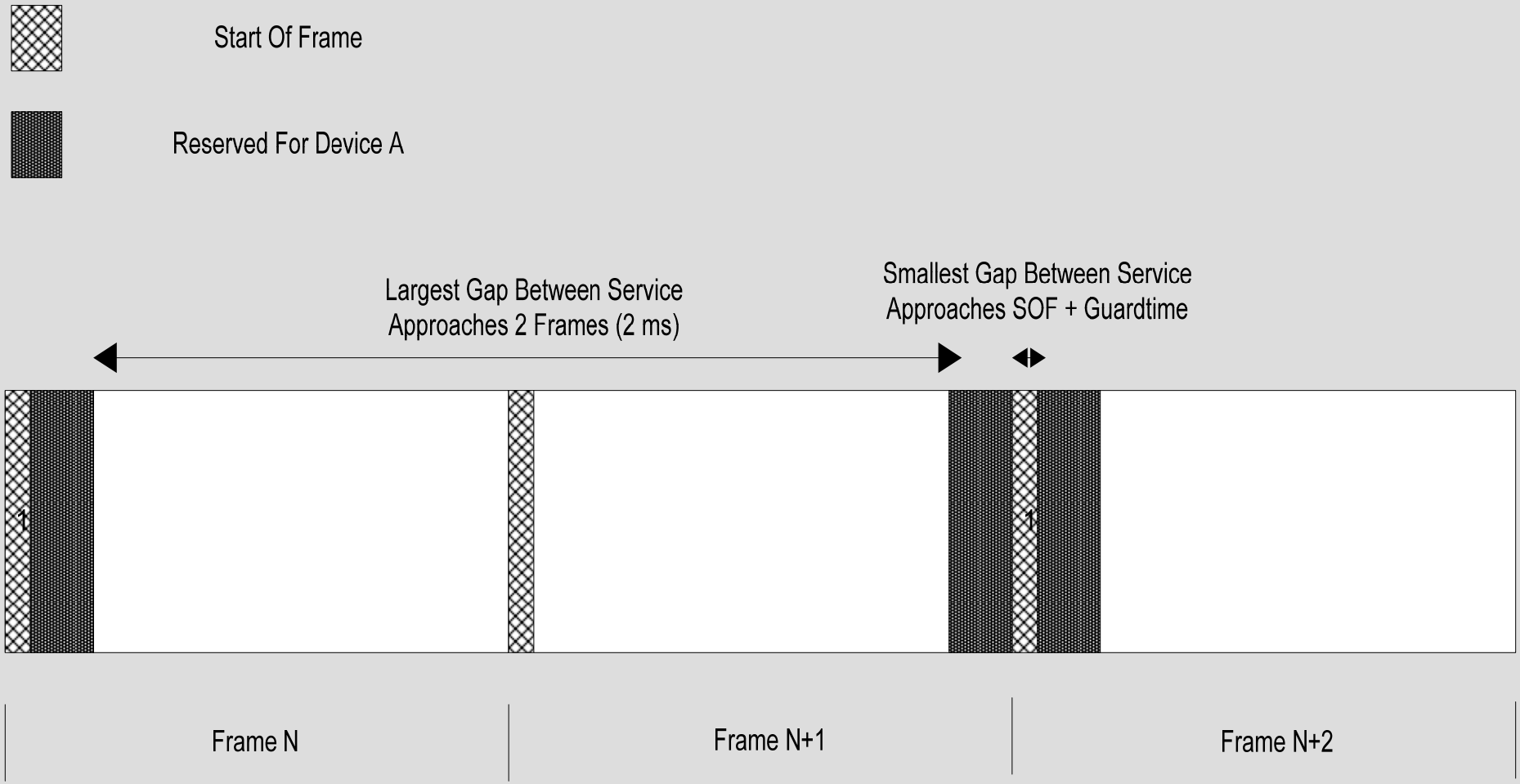

# Certified Wireless USB Periodic Transfer Models

Dan Froelich
Intel Corporation

# Agenda

- Wired Isochronous Model
  - Overview
  - Key Features
- Wireless Media
  - Reliability
  - Coexistence (Shared With Other Hosts And UWB Devices)
- Certified Wireless USB Isochronous Model Overviews
  - Isochronous IN
  - Isochronous OUT
- Isochronous Model Details
  - Synchronization
  - Protocol
  - Data Header Format
  - Buffer Requirements
  - Reservations
  - Flow Control
  - Error Handling Details
- Interrupt Transfers
- Summary and Conclusions

# Wired USB FS Isochronous Model



# Wired USB Isochronous Feature Summary

- Guaranteed Service Attempt Every (micro)Frame
  - Reserved Bandwidth
  - Service Interval Bound 1 ms +- ~ 1 ms
  - Typical Service Interval ~ 1 ms
- Reliability
  - Bit Error Is $10^{-9}$ or Better (PER $10^{-6}$)
  - No Retries or Handshaking Needed in Protocol
- Minimal Buffering
  - Typical Implementations Require 2 Frames (ms) of Buffering
- Synchronization
  - SOFs Occur at Regular 1 ms Intervals
- Data Delivery Time
  - Data Is Sent in Specified (micro)Frames

# Agenda

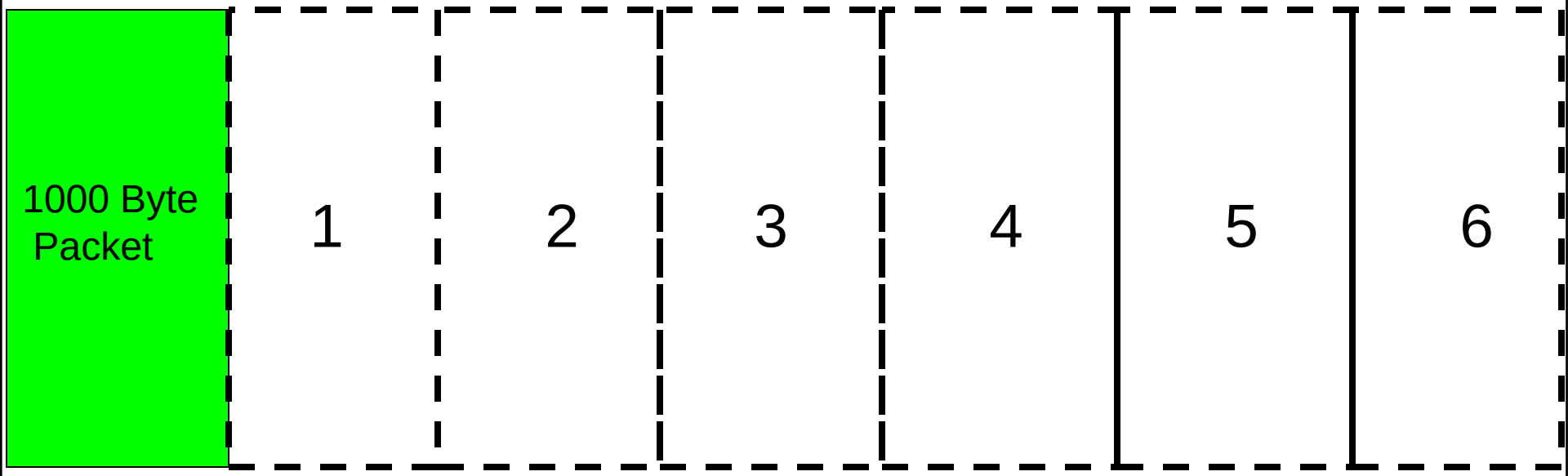
- Wired Isochronous Model
  - Overview
  - Key Features
- Wireless Media
  - Reliability
  - Coexistence (Shared With Other Hosts And UWB Devices)
- Certified Wireless USB Isochronous Model Overviews
  - Isochronous IN
  - Isochronous OUT
- Isochronous Model Details
  - Synchronization
  - Protocol
  - Data Header Format
  - Buffer Requirements
  - Reservations
  - Flow Control
  - Error Handling Details
- Interrupt Transfers
- Summary and Conclusions

# Certified Wireless USB Media Reliability

- Certified Wireless Media Is Unreliable (vs. Wired)
  - Average PER rates will reach 10%+
    - UWB Phy Specification Specifies an 8% PER or better for 90% of paths is “good” for several simulation cases
    - This Makes PER Rates 100,000 X + Wired Rates
  - Events could cause much higher short term error rates
    - People walking between transmitters
    - Adjacent channel interference
    - Merging Device Clusters
- Some streaming formats require Packet Error rates on the order of 10^-5 or better

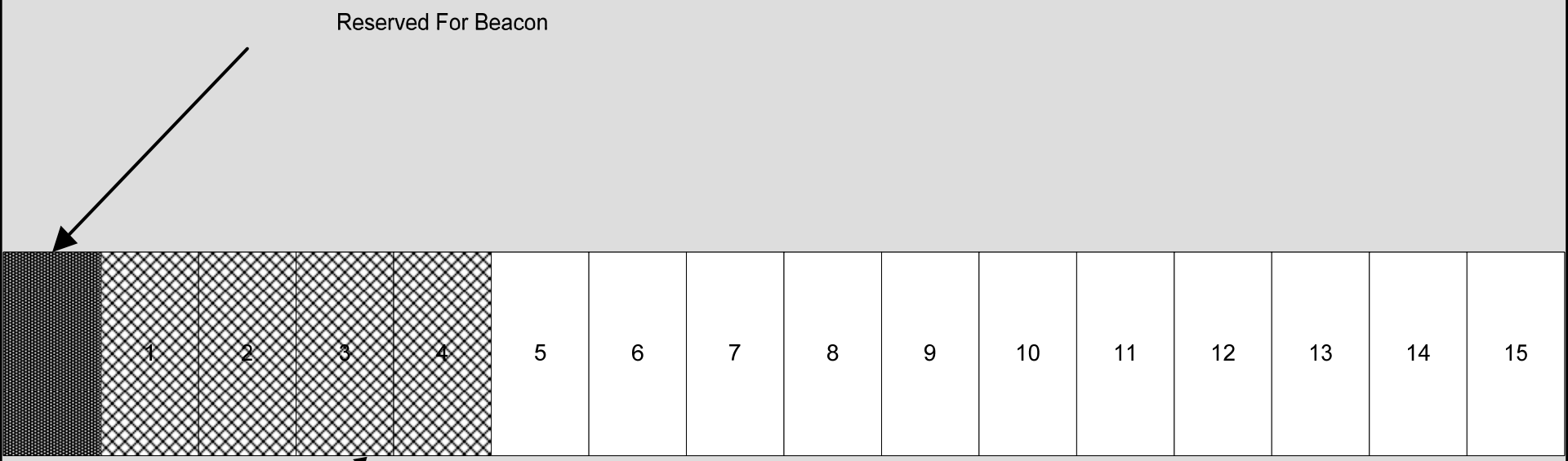
# Certified Wireless USB Media Reliability – Example

## Guaranteeing 10^-06 With 10% PER




To Guarantee 10^-6 Reliability You Need To Be Able to Retry Up to 6 Times in a Slot
Need Handshaking (Allow Reclamation Of Unused Time)
This Still Breaks with Short Term Error Bursts

# UWB Shared Media – Big Continuous Reservations Cause Problems

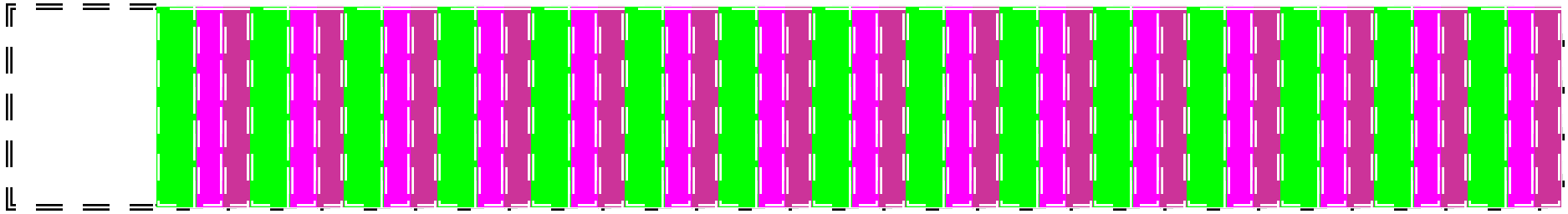



20 ms Service Gap For WUSB Host Joining Channel

Reserved By Other UWB Devices

A Host Can Not Require an Empty Channel for Iso Support
Beacon Reservation and Other UWB Device Reservations Must Be Assumed

# Service Interval



Reserved
For Beacs
~ 4 ms

Dev A

2 ms
Reservation

Intervals

6 ms (Beacon)

2 ms (Non-Beacon)

Dev B

1 ms
Reservation

Intervals

7 ms (Beacon)

3 ms (Non-Beacon)

Dev C

1 ms
Reservation

Intervals

7 ms (Beacon)

3 ms (Non-Beacon)

WUSB Channel Limits Maximum Service Interval to 8 ms (beacon)
or 4 ms (non-beacon).
Possible Service Intervals ~= 4*, 8, 16, 32, . . .

# Service Interval Encoding

| *bIntervalValue* | *Requested Service Interval (ms)* |
|---|---|
| 6 | 4.096* |
| 7 | 8.192 |
| 8 | 16.384 |
| 9 | 32.768 |
| 10 | 65.536 |
| 11 | 131.072 |
| 12 | 262.144 |
| 13 | 524.288 |
| 14 | 1048.576 |
| 15 | 2097.152 |
| 16 | 4194.304 |

Service interval bounds are multiples of 128 microseconds (not 125)

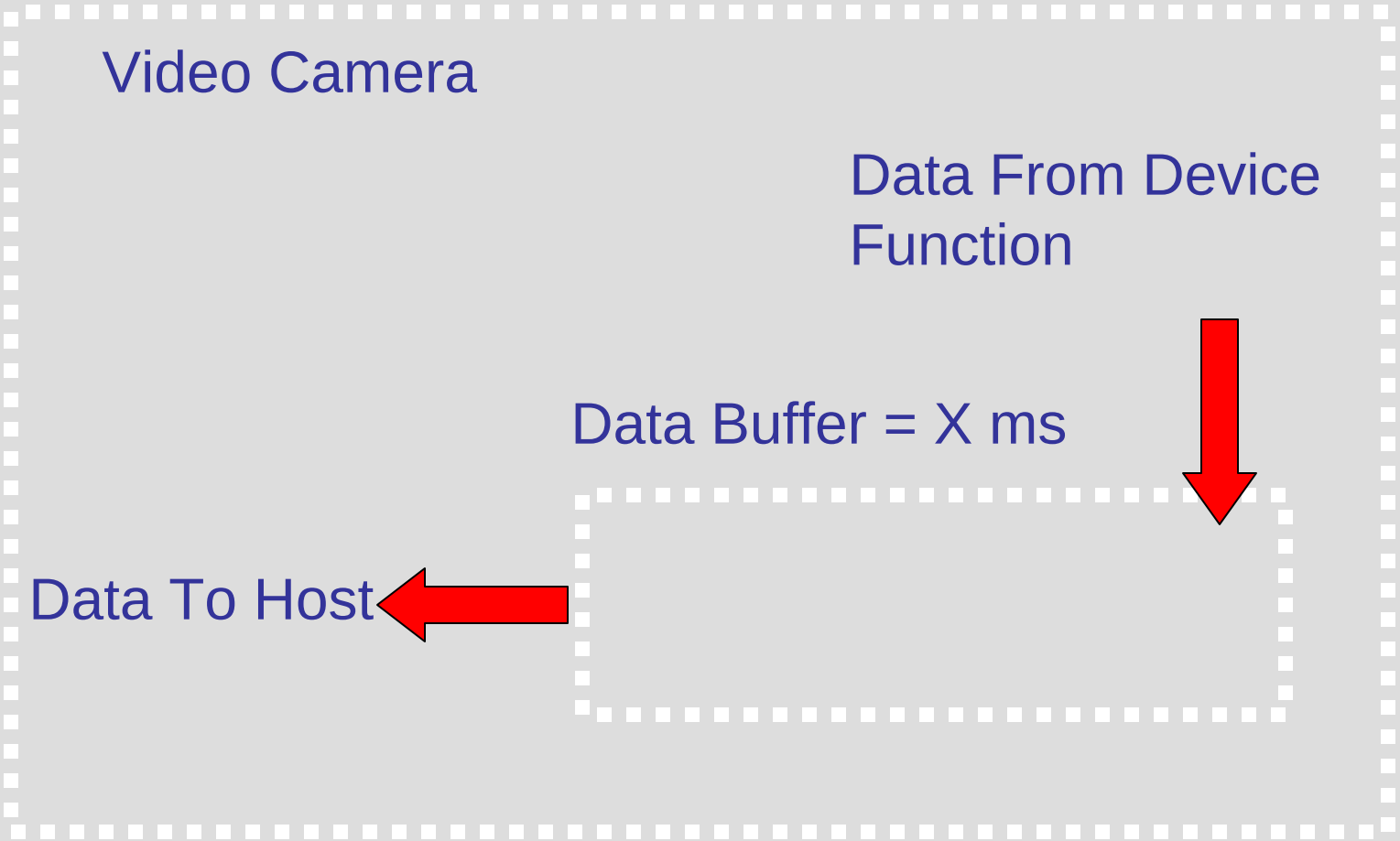
# WUSB Iso Model Cannot Be the Same as Wired

- Media
  - Unreliable.  Error Rates 100000X+ higher than wired
  - Shared.  Larger service interval bounds
- Key Wired Characteristics Can Be Preserved
  - Synchronization (Provided By SOF Or Better)
  - Effective Reliability (10^-5 PER or Better)
    - Same level of reliability can be achieved under most circumstances
    - Definition of when packets need to be delivered changes

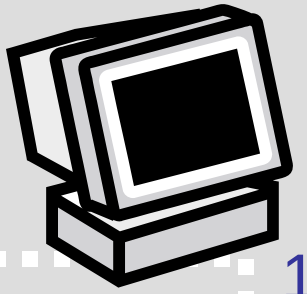
# Agenda

- Wired Isochronous Model
  - Overview
  - Key Features
- Wireless Media
  - Reliability
  - Coexistence (Shared With Other Hosts And UWB Devices)
- Certified Wireless USB Isochronous Model Overviews
  - Isochronous IN
  - Isochronous OUT
- Isochronous Model Details
  - Synchronization
  - Protocol
  - Data Header Format
  - Buffer Requirements
  - Reservations
  - Flow Control
  - Error Handling Details
- Interrupt Transfers
- Summary and Conclusions

# Example – Isochronous In Device



# Example – Isochronous In Stream Start Up

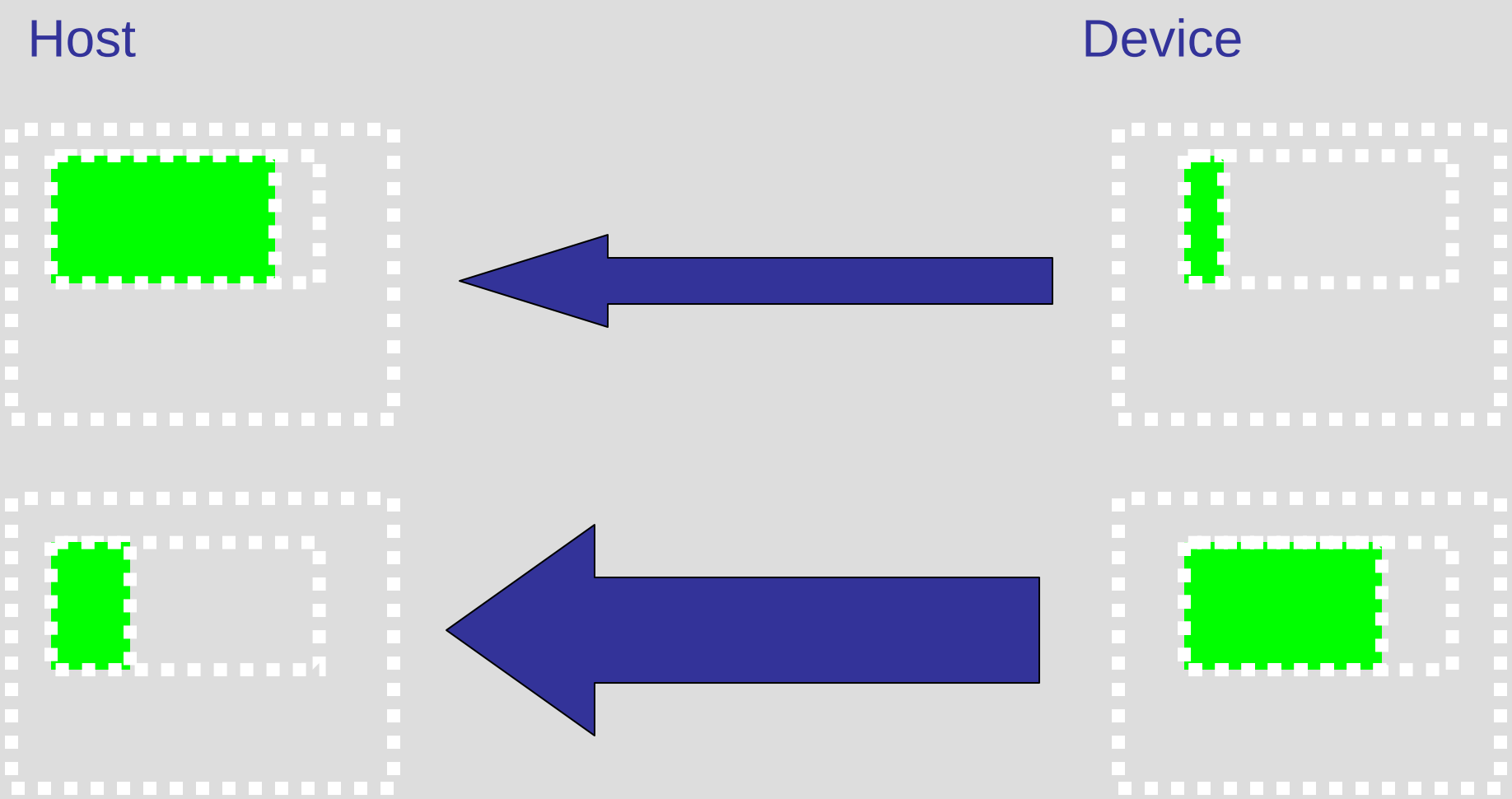


Host

Iso In Device

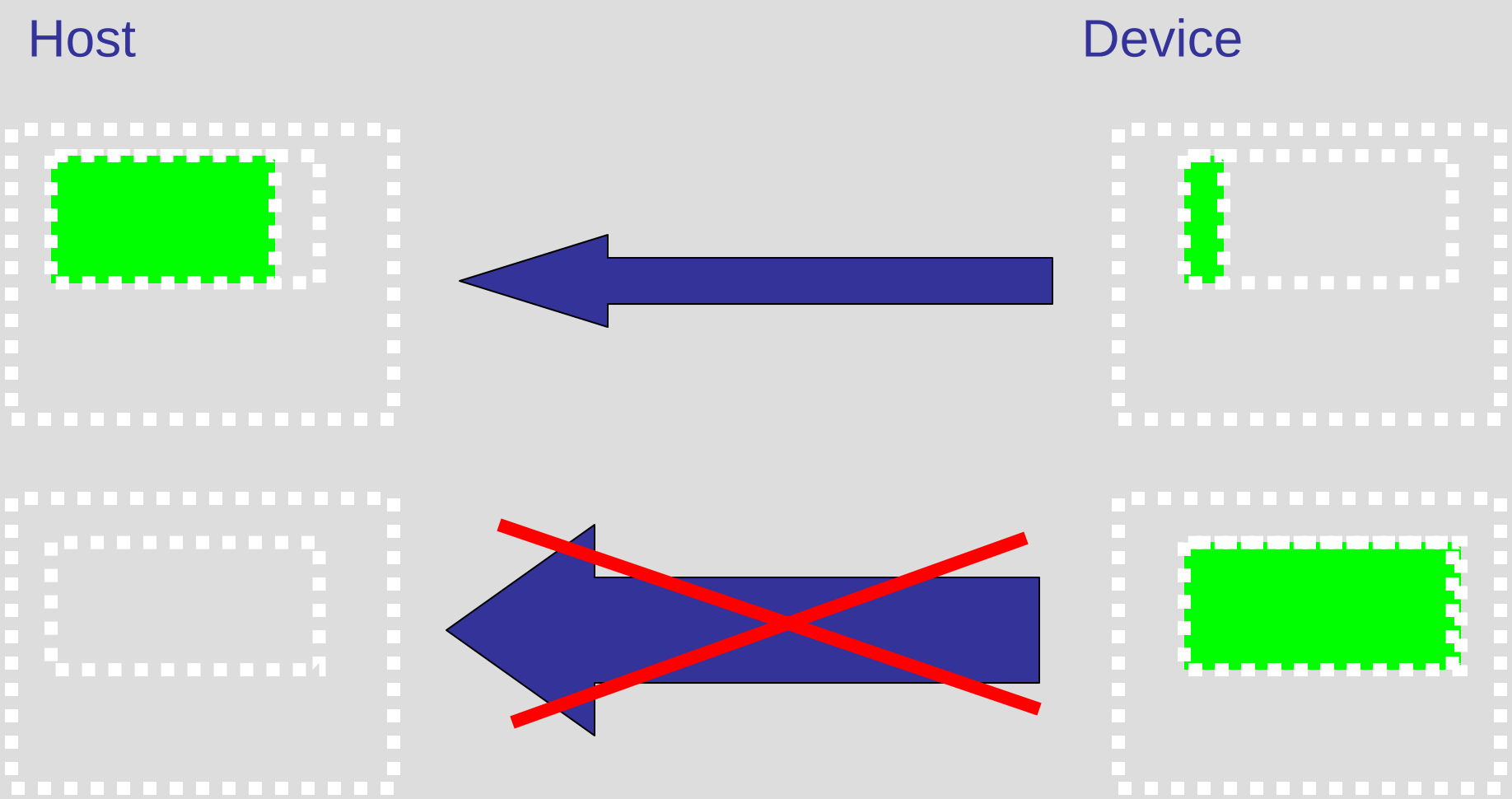
1. Host Obtains Device Buffer Size and Starts Asking For Data
2. Host Buffer Fills
3. Host Starts Using Data

# Example – Isochronous In Normal Operation




Host Can Use Approximate Buffer State to Adjust Scheduling

# Example – Isochronous In Error Case



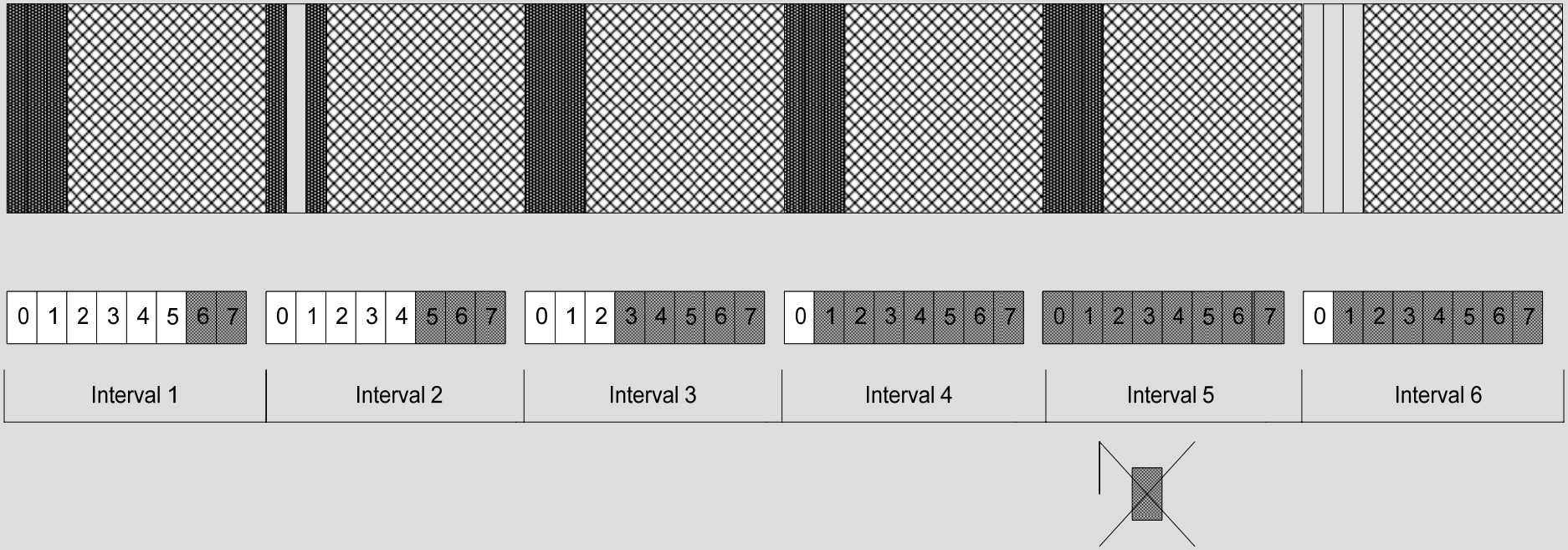

Data Is Only Discarded When Buffer Overflow Happens on Device

# Isochronous In – Data Discard and Recovery

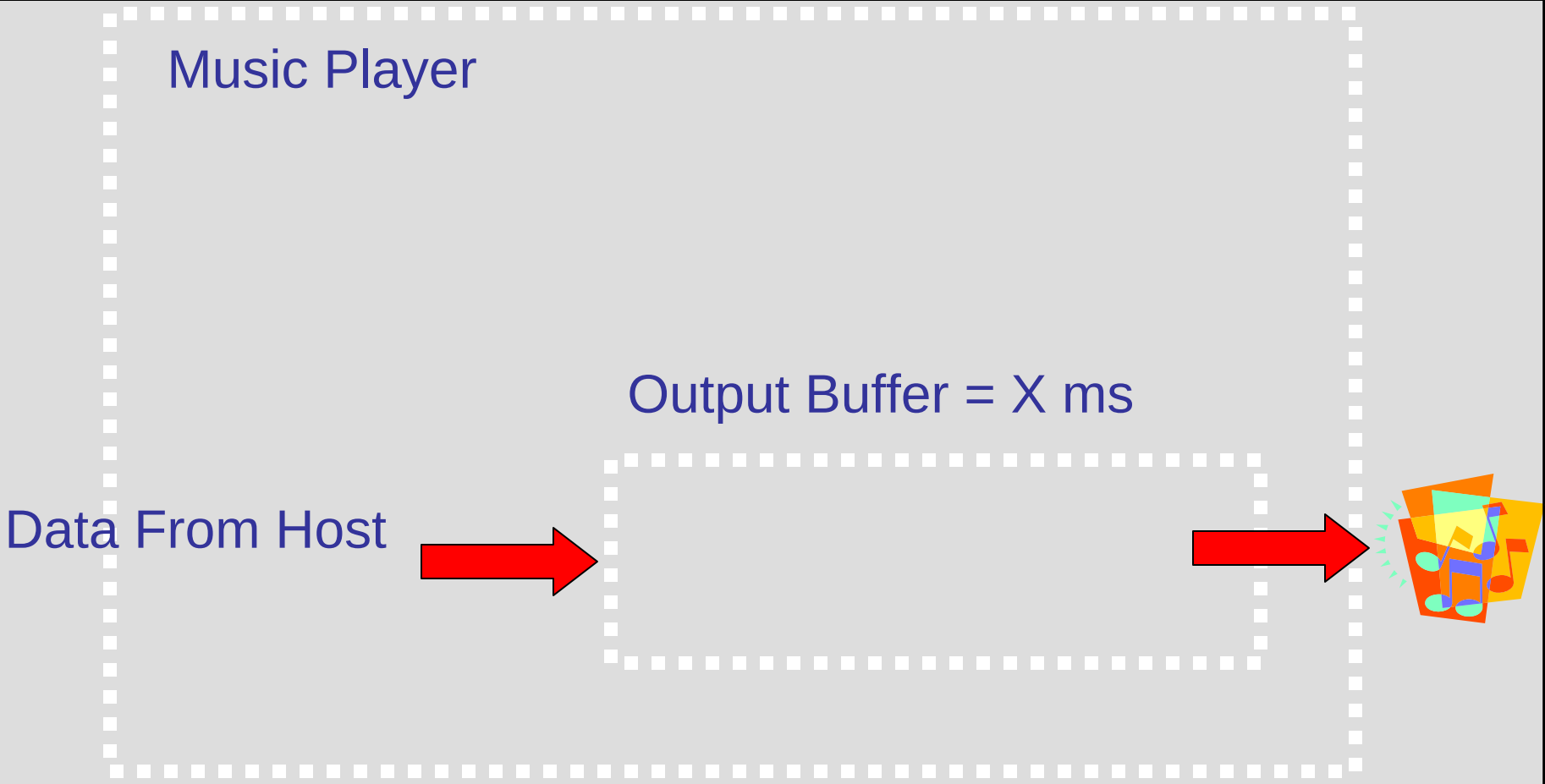
Packet Sent

Packet Smashed

Not Reserved For Iso Stream



# Isochronous Out Device

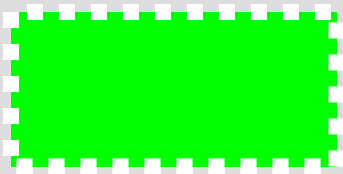


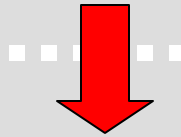
# Isochronous Out Stream Start Up

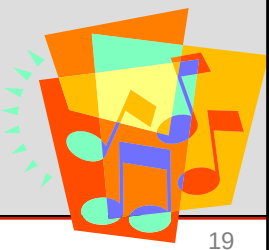
Host

Iso Out Device

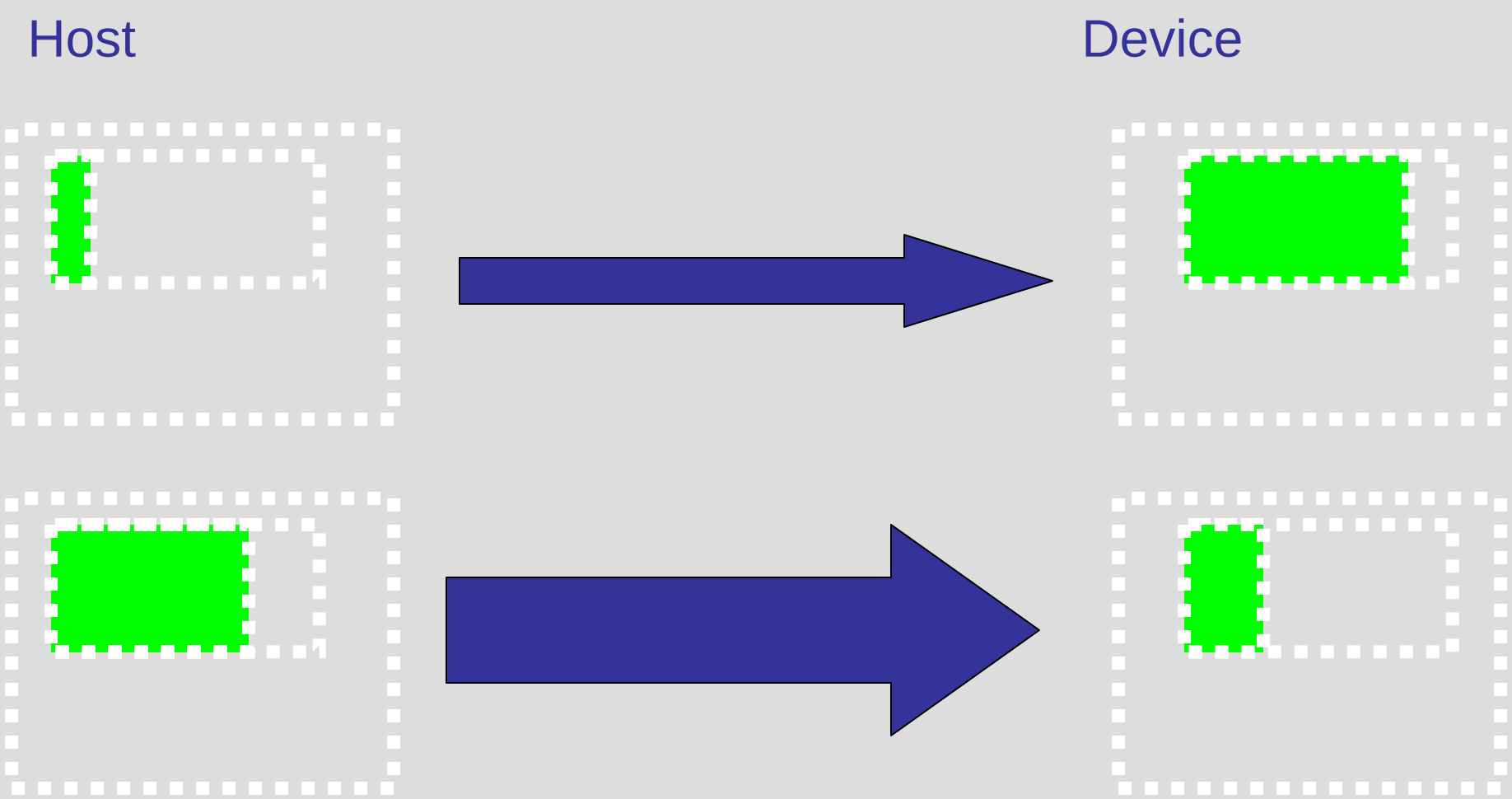
1. Host Obtains Device Buffer Size and Starts Sending Data
2. Device Output Buffer Fills
3. Device Starts Data Output







# Isochronous Out Normal Operation

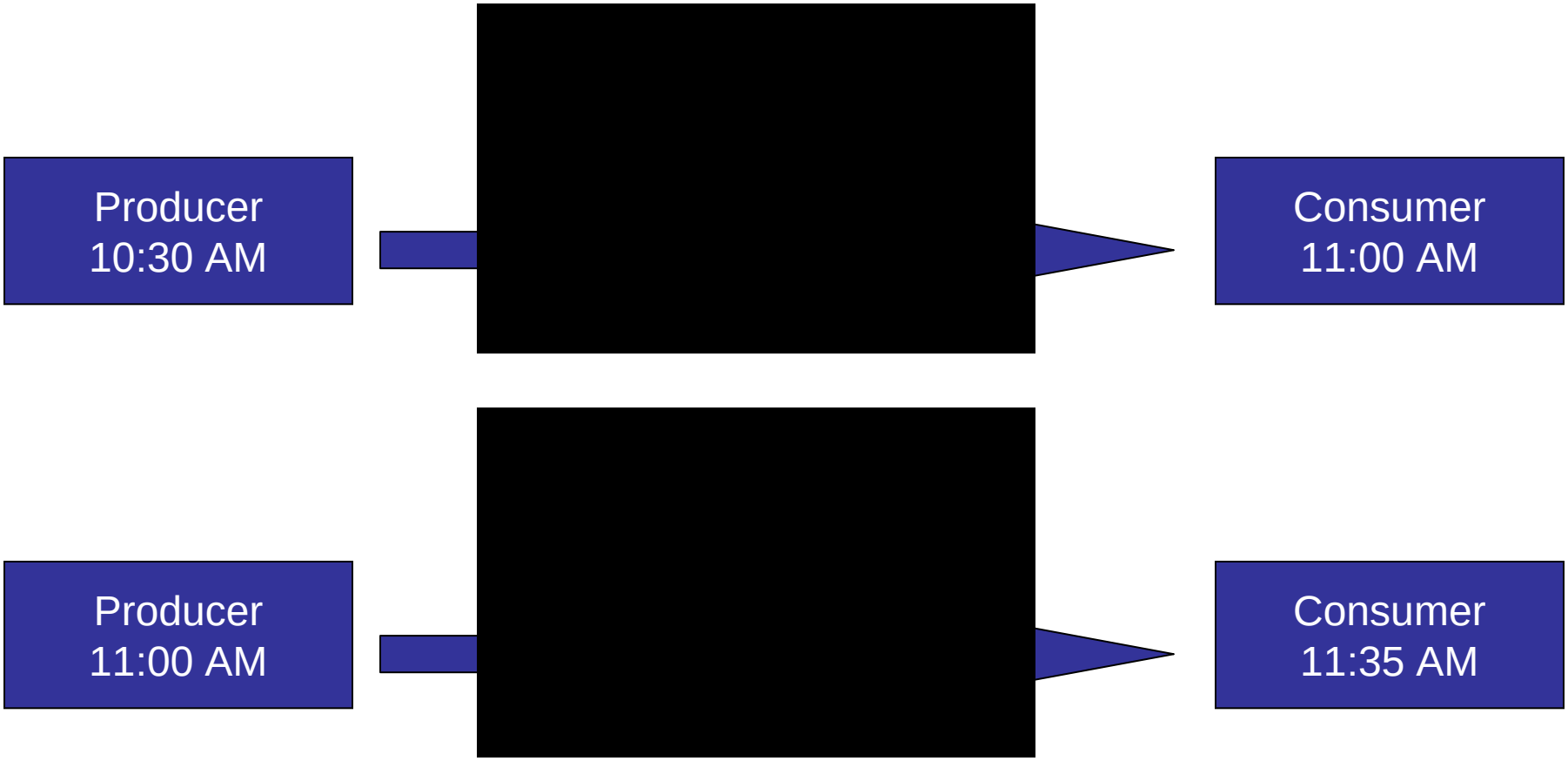



Host Can Use Approximate Buffer State to Adjust Scheduling

# Agenda

- Wired Isochronous Model
  - Overview
  - Key Features
- Wireless Media
  - Reliability
  - Coexistence (Shared With Other Hosts And UWB Devices)
- Certified Wireless USB Isochronous Model Overviews
  - Isochronous IN
  - Isochronous OUT
- Isochronous Model Details
  - Synchronization
  - Protocol
  - Data Header Format
  - Buffer Requirements
  - Reservations
  - Flow Control
  - Error Handling Details
- Interrupt Transfers
- Summary and Conclusions

# Synchronization – Problem Statement



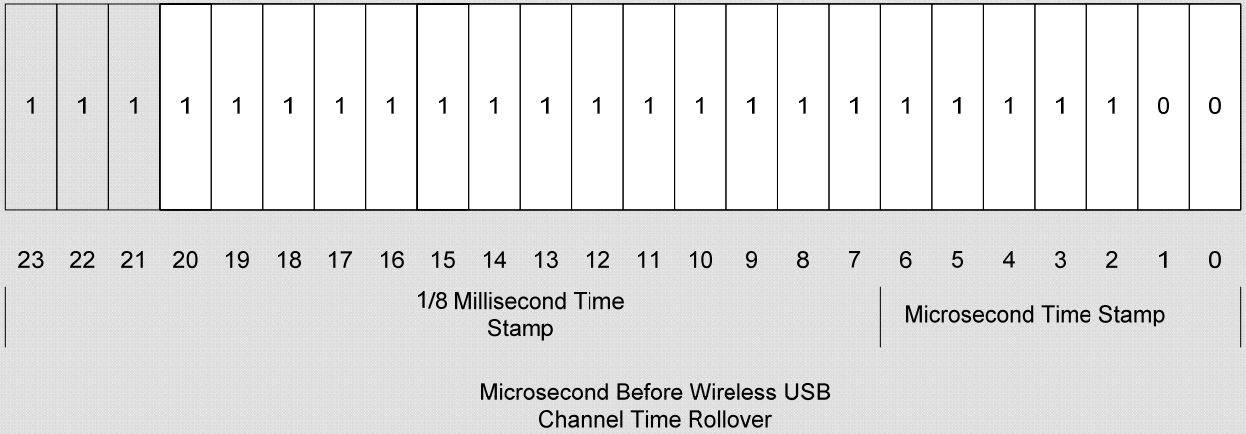

Producer and Consumer Have to Have a Way to Have Clocks That Drive Production and Consumption at the Same Rate/Time or Buffers Overflow/Underflow

# Synchronization – Time Stamp Location

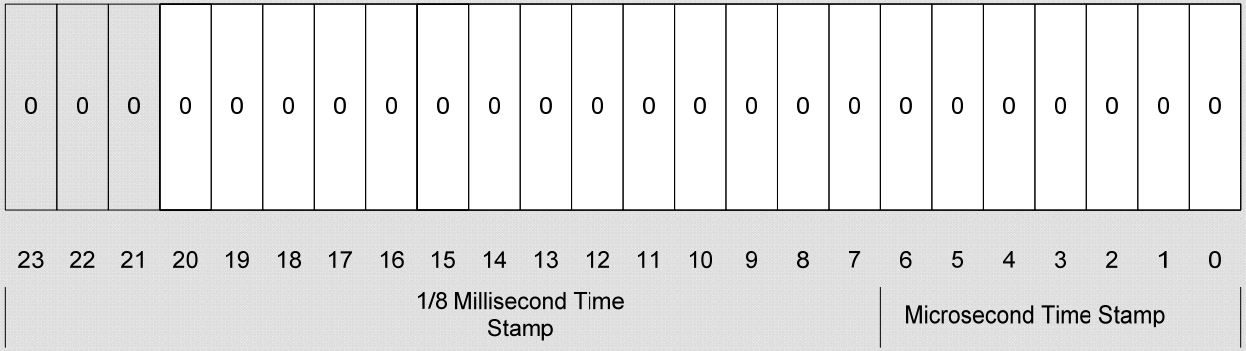
| 2 | 1 | 2 | 2 | 3 | VAR | VAR | … | VAR |
|---|---|---|---|---|---|---|---|---|
| WUSB App Code | MMC Code | Next MMC Time | Reserved | WUSB Channel Time Stamp | IE[0] | IE[1] | | IE[N] |

3 Byte Time Stamp in MMC
Time Stamp Reflects Start Time for MMC Transmission on the Air

# Synchronization – Time Stamp Format




Microsecond Before Wireless USB Channel Time Rollover




Microsecond After Wireless USB Channel Time Rollover

Microsecond Granularity Time Stamp to Easily Map to Wired USB Time

# Synchronization – Time Stamp Accuracy

- PHY TX Delay
  - Variable
  - Not a Problem if Factored in Correctly
- PHY TX Delay Uncertainty
  - Worst Case
    - +/- 2 Pclock
    - +/- 30 Nanoseconds
  - Best Case
    - +/- 1-2 Nanoseconds
- Wireless USB Infrastructure Requires MMC Transmission to Start on Microsecond Boundaries
- MMC Time Stamp Must Be Accurate
  - Time Stamp in MMC Must Be Host Time at Start of MMC Transmission
  - Accuracy requirement is +/- 40 Nanoseconds (Worst Case + Padding)
  - Based on Host Free Running 20 ppm Clock

MMC WUSB Channel Timestamp Provides Wired Equivalent Clock and Accuracy

# Protocol Details

- Retries
  - PER Can Be 10%
  - Short Term Error Bursts Can Occur
  - Retries Required
- Handshaking
  - Wired Reliability Would Require ~6 Broadcasts Per Packet Without Handshaking
    - Short Term Error Burst Tolerance Still Problematic
  - Handshaking Required
- Data Discard
  - Only Required by Physical Limitations
  - Buffer Size

Isochronous Protocol Is Nearly Identical To Wireless Bulk Protocol
Protocol Details In Protocol Presentation

# Isochronous Header Format

| 2 | 1 | 2 | 2 | VAR | 2 | VAR | … | 2 | VAR |
|---|---|---|---|---|---|---|---|---|---|
| Standard WUSB Header Fields | bNumbIso Segments | wPresentation Time | wLength1 | Isoch Data 1 | wLength 2 | Isoch Data 2 | | wLength N | Isoch Data N |
| | | | wPresentationTime | | wPresentationTime + bInterval | | | wPresentationTime + (n-1)*bInterval | |

General Isochronous Data Header
A Data Segment Can Not Be Split Across Packets

# Interval and Maximum Packet Size in Descriptors

- Endpoint Descriptor
  - wMaxPacketSize – Maximum size for an isochronous data segment
  - bInterval – Interval between isochronous data segments
- Endpoint Companion Descriptor
  - wOverTheAirPacketSize – Maximum size of over the air packets (one or more isochronous segments)
  - bOverTheAirInterval – Over the air requested service interval

# General Presentation Time Use

- Isochronous OUT
  - Host System Applies Presentation Times
  - Presentation Times Indicate Wireless USB Channel Time for Intended Data Consumption
    - Times Are Applied to Make Optimal Use of Device Buffering
    - Device Consumes Based on Presentation Time
  - Host Controller Discards Packets if Wireless USB Channel Time Exceeds Presentation Time and Notifies Device
- Isochronous IN
  - Device Applies Presentation Times
  - Presentation Times Indicate Wireless USB Channel Time When Data Was Produced
  - Device Discards Packets Only When Physical Storage Is Exceeded
  - Host System Manages When Applications Start Consumption to Make Optimal Use of Device Buffering

Presentation Time Standardized for Legacy Application, DWA Support

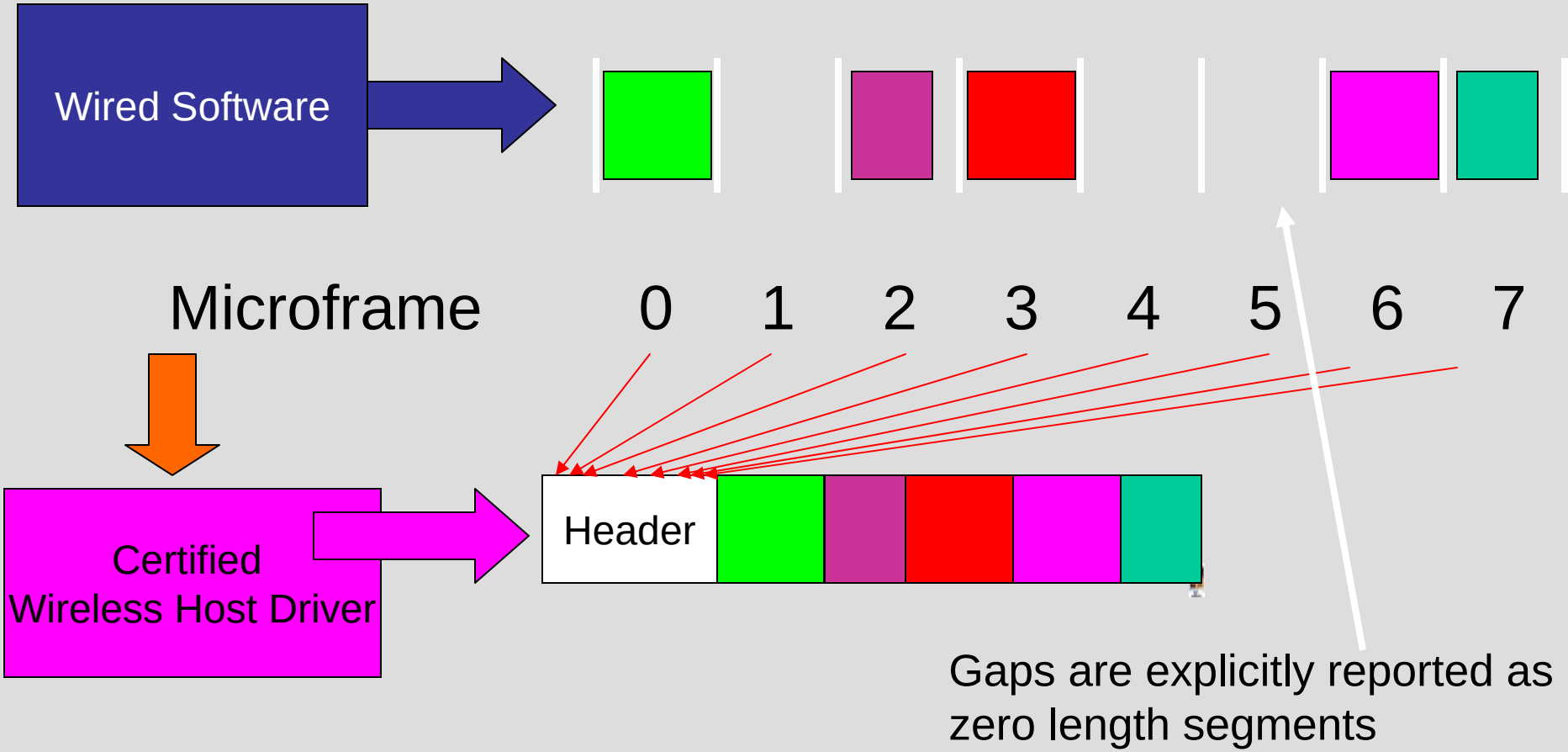
# Isochronous Header and DWA

| 2 | 1 | 2 | 2 | VAR | 2 | VAR | … |
|---|---|---|---|---|---|---|---|
| Standard WUSB Header Fields | bNumberIso Segments | MicroframeX | wLength1 | Data To Send In Microframe X | wLength2 | Data To Send In Microframe X + bInterval | |

Not counted towards maxpacket size

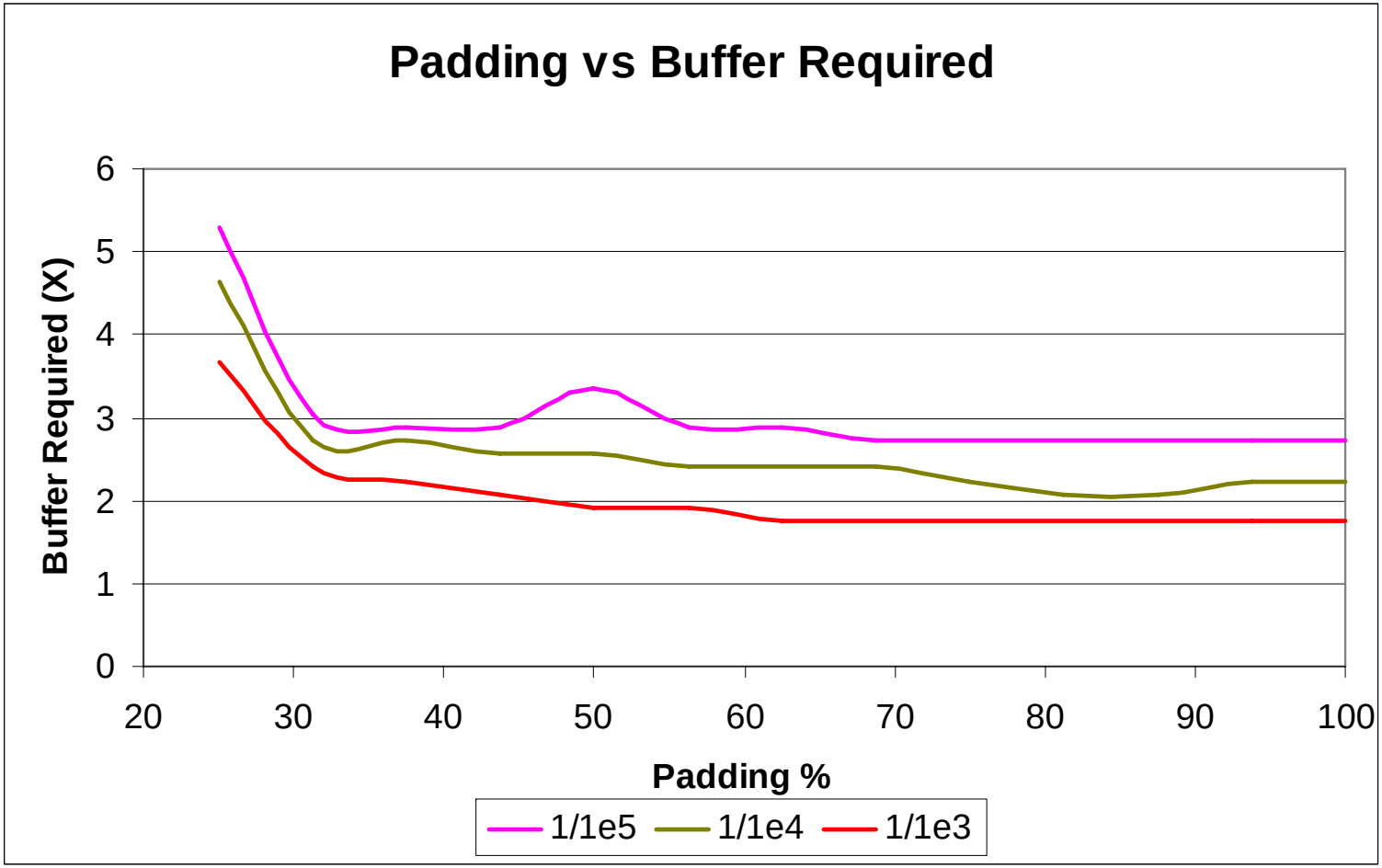
General Isochronous Data Header Can Be Used For DWA Support With Existing Applications and Class Drivers

# Isochronous Header – Wired Applications and Class Drivers




Microframe Level Delivery Information Can Be Conveyed To A Native Wireless Device If Necessary. Possible Support For Legacy Apps/Class Drivers With Native Devices

# Benefit from Padding Reservation Size for Retries




Short Term Error Tolerance Gains From Reservation Padding Diminish Beyond 30-40 %

# Buffer Requirements

- Device Tradeoff
  - Acceptable Stream Delay/Latency
  - Cost
  - Short Term Error Tolerance
- Minimum Buffering
  - 2x Smallest Guaranteed Service Interval
    - 2 x 4* = ~ 8 milliseconds
    - At least 16 milliseconds recommended
- Maximum Buffering
  - Limited by Cost or Acceptable Latency
  - Different Buffer Options Can Be Presented Through Alternate Interfaces

# Reporting Buffer Size

| | | | | |
|---|---|---|---|---|
| 4 | *wMaxStreamDelay* | 2 | Number | For isochronous endpoints this field is a value from 1 to 65535 indicating the maximum amount of delay in 128 microsecond units that can be supported by the stream. The endpoint must provide exactly the amount of buffering to support this delay. An IN endpoint must be able to store this amount of data before having to discard data. An OUT endpoint must be able to fill its buffering (except for storage for less than a maximum size burst) before it is allowed to NAK.<br><br>Refer to the dataflow Section 4.11 for more information.<br><br>For interrupt, bulk, and control endpoints this field is reserved and must be zero. |

- An isochronous OUT endpoint is not allowed to NAK until its buffer is full (less than 1 max burst available)

# Reservations

- Endpoint Asks for Actual (Average) Throughput Need without Padding
  - Service Interval (4* ms, 8 ms , 12 ms, . . .)
  - Maximum Burst Size (16)
  - Maximum Packet Size (3584)
- Host System Provides Reserved Time for Retries
  - 30% Minimum
    - If This Can't Be Met Device Is Not Admitted
    - Reserved Pool Size Does Not Grow Linearly with Multiple Streams
  - Host Policy Can Reserve More in Poor Channel Conditions

# Reservation Limits

| Maximum Burst Size | Maximum Packet Size | Total Payload Per Service Interval |
|---|---|---|
| 1 | 1-3584 | 1-3584 |
| 2 | 257-3584 | 513-7168 |
| 3 | 342-3584 | 1025-10752 |
| 4 | 385-3584 | 1537-14336 |
| 5 | 410-3584 | 2049-17920 |
| 6 | 427-3584 | 2561-21504 |
| 7 | 439-3120 | 3073-21845 |
| 8 | 449-2730 | 3585-21845 |
| … | | |
| 16 | 481-1365 | 7681-21845 |

Average bandwidth limited to 40 mbps to meet regulatory concerns

# Reservations – Admission Decisions

- Set Interface Call Occurs
  - Previous Communication with Device May Only Have Occurred at Base Rate
  - Admission Decisions Depend on Highest Usable Rate
  - Host Needs a Standard Mechanism to Gather Additional Information for Admission Decision
- Required Loopback Commands
  - DATA_LOOBACK_WRITE
  - DATA_LOOPBACK_READ
  - Standard Commands Required on Control Pipe
  - Data Phase Can Occur at Any Supported Rate
  - Will Also Be Used in Testing/Enabling/Compliance

# Flow Control

Can Be Reclaimed

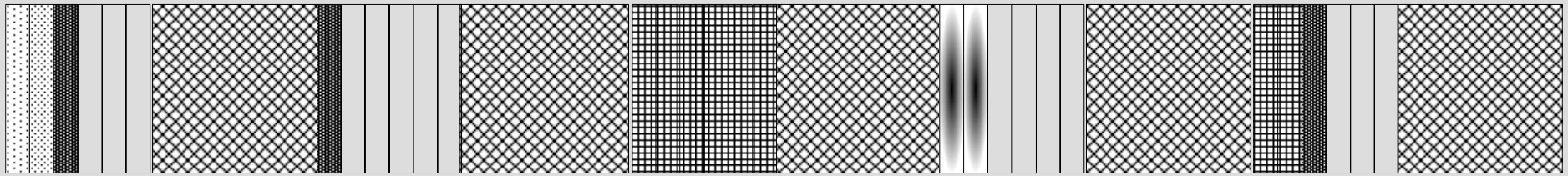
Valid Response

NAK

Smashed Packet

No Device Response

Not Reserved For Periodic Endpoint



| Interval 1 | Interval 2 | Interval 3 | Interval 4 | Interval 5 |
|---|---|---|---|---|

# Isochronous IN Error Conditions - Device Buffer Perspective

*Example*

- 100 Packets Per Interval
- 3 Intervals of Buffering (300 Packets)
- 120 Chances Scheduled Per Interval (20% Padding)

300 Packet Buffer

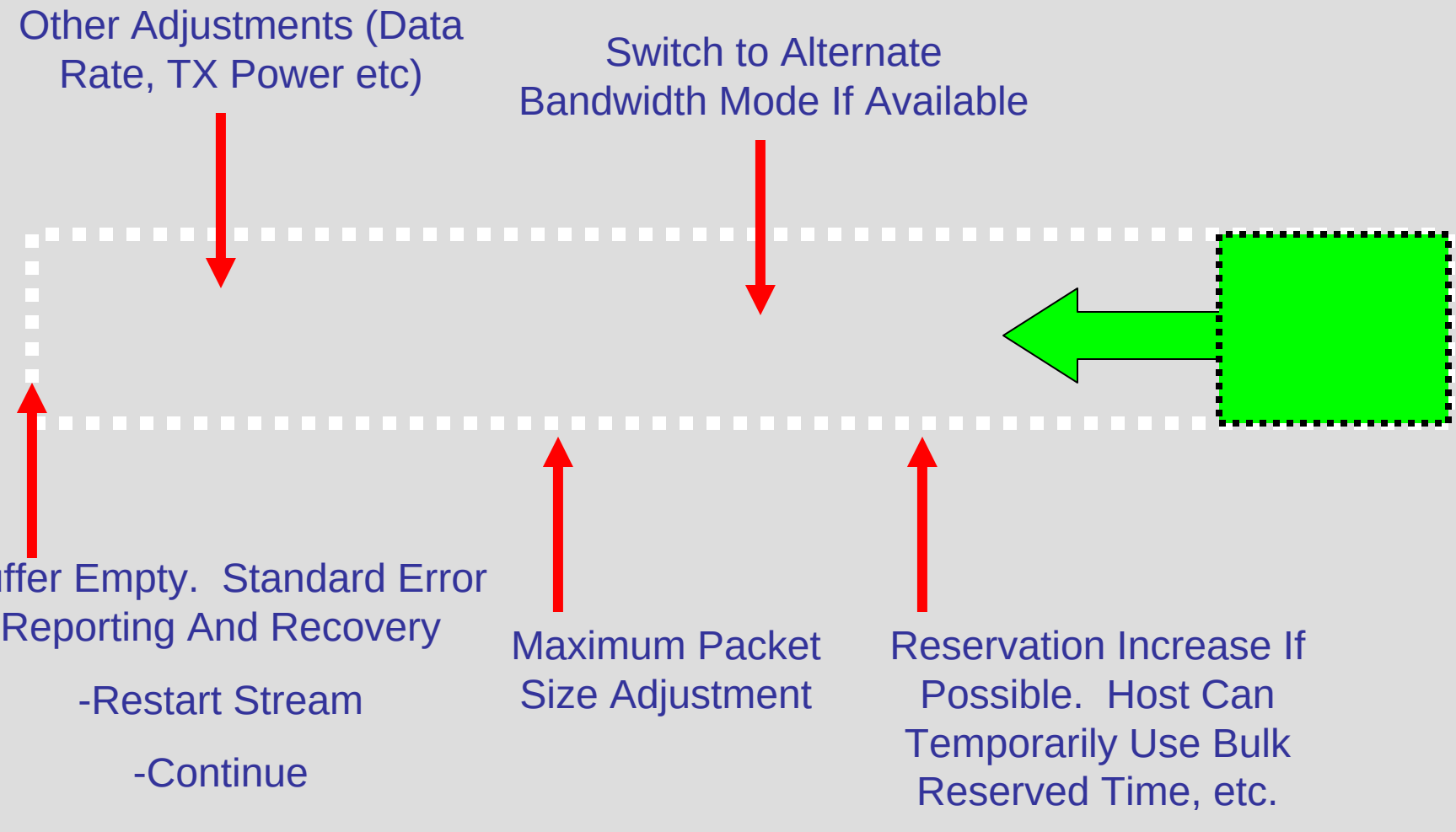
Stays Empty With Good Link

# Error Conditions – Device Buffer Perspective

*Example*

- Error Rate Jumps to 20%
- 96 Packets Transferred Per Interval
- 4 Packets Added to Device Buffer Per Interval
- It Will Take 75 Intervals for Buffer to Overflow (Average)

# General Error Handling (Device Buffer)



# Data Discard Details

- Isoch OUT
  - Isoch Out – Host Discards If Presentation Time Expires
  - Isoch Out – MMC IE Indicates Discard and Burst Sequence Skip
- Isoch IN
  - Isoch In – Device Discards Due to Physical Storage Limits
  - Discard Removes Packet and Replaces It with New Packet Using the Same Burst Sequence Number, etc
  - Requires Host Receivers to “Order” Based on Presentation Time

1.0 Spec defines ISOCH_DISACRD_IE to provide explicit information to devices in Isoch OUT discard cases

# Isochronous Packet Discard IE

| Offset | Field | Size | Value | Description |
|---|---|---|---|---|
| 0 | *bLength* | 1 | Constant | The size of this IE: 14 bytes |
| 1 | *IE_Identifier* | 1 | Constant | WISOCH_DISCARD_IE |
| 2 | *bDiscardID* | 1 | Number | An ID number for the discard IE |
| 3 | *bDeviceAddress* | 1 | Number | Device Address |
| 4 | *bmAttributes* | 1 | Bitmap | This bitmap has the following encoding<br>**Bit** **Value** **Description**<br>3:0 Variable USB Endpoint Number<br>7:4 Zero Reserved |
| 5 | *bFirst ReceiveWindow Position* | 1 | Number | The sequence number of the first position in the *bmDeviceReceiveWindow* field. |
| 6 | *wNumber Discarded Packets* | 2 | Number | The number of discarded packets |
| 8 | *wNumber Discarded Segments* | 2 | Number | The number of discarded isochronous segments. |
| 10 | *bmDevice Receive Window* | 4 | Bitmap | Bitmap indicating the sequence numbers that the host expects to be active in the device receive window.<br>The dataflow chapter explains how the host calculates the expected receive window when discards occur. |

# Agenda

- Wired Isochronous Model
  - Overview
  - Key Features
- Wireless Media
  - Reliability
  - Coexistence (Shared With Other Hosts And UWB Devices)
- Wireless USB Isochronous Model Overviews
  - Isochronous IN
  - Isochronous OUT
- Isochronous Model Details
  - Synchronization
  - Protocol
  - Data Header Format
  - Buffer Requirements
  - Reservations
  - Flow Control
  - Error Handling Details
- Interrupt Transfers
- Summary and Conclusions

# Interrupt Endpoints

- Regular and low power versions
- Retry guarantee
  - Regular 5
  - Low Power 2
- Service Interval
  - 4* milliseconds smallest available due to WUSB Channel characteristics
- Max Burst Size Is 1
- Maximum Packet Sizes
  - Regular – 1024
  - Low Power – 64
- Flow Control Has Same Meaning as Isochronous
  - Regular – Must NAK
  - Low Power – Allowed Not to NAK (but must periodically to avoid trust timeouts)

# Summary and Conclusions

- Wired Iso Model Doesn't Work for Wireless
  - No Retries
  - Small Service Intervals
  - No Chance to Recover if Data Isn't Sent in Specified Interval
- Wireless Media Problems
  - Much Higher Error Rates Than Wired
  - Much Larger Service Interval Bounds
- Key Iso Model Changes
  - Retries and Handshaking
  - Delay Added to Allow Service Chances in Multiple Intervals
    - More Efficient Use of Bandwidth
    - Better Short Term Error Tolerance
    - More Time to React When Link Degradation Persists
  - Data Header with Presentation Time Information
    - Potential Legacy Class/Application Software Support
    - DWA Support


# Developers Conference 2007

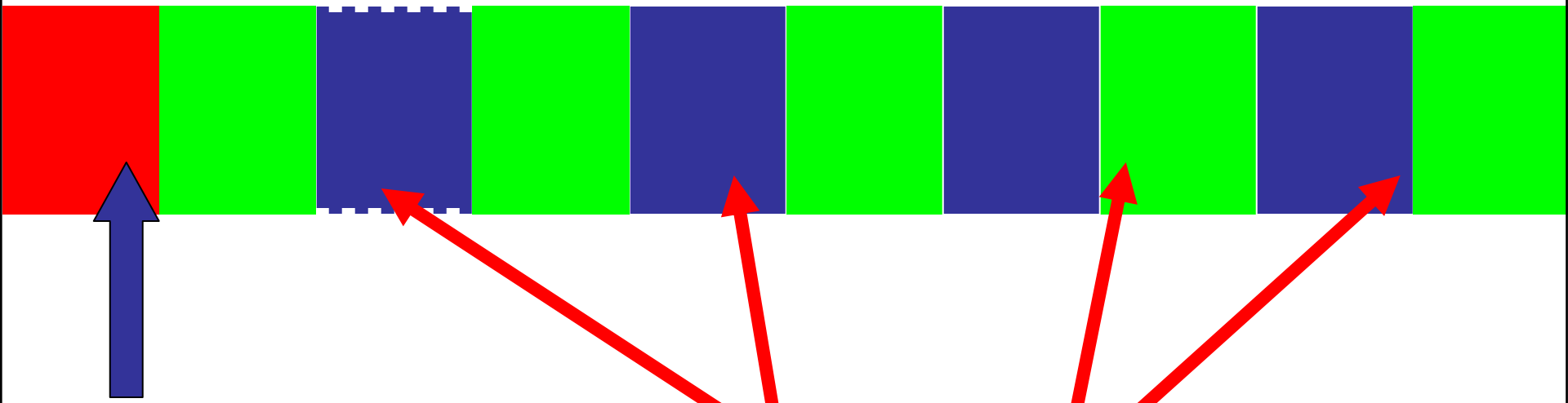
## Amsterdam, The Netherlands


# Backup
# Simulation Model and Results

# Simulation Model

- Parameters
  - Superframe Layout
  - Buffer Size
  - Retry Count
  - PER
  - Packet Size
  - Reservation Padding
- Simulations Results

# Superframe Layout



Worst Case Beacon (4 ms)

WUSB Host Time

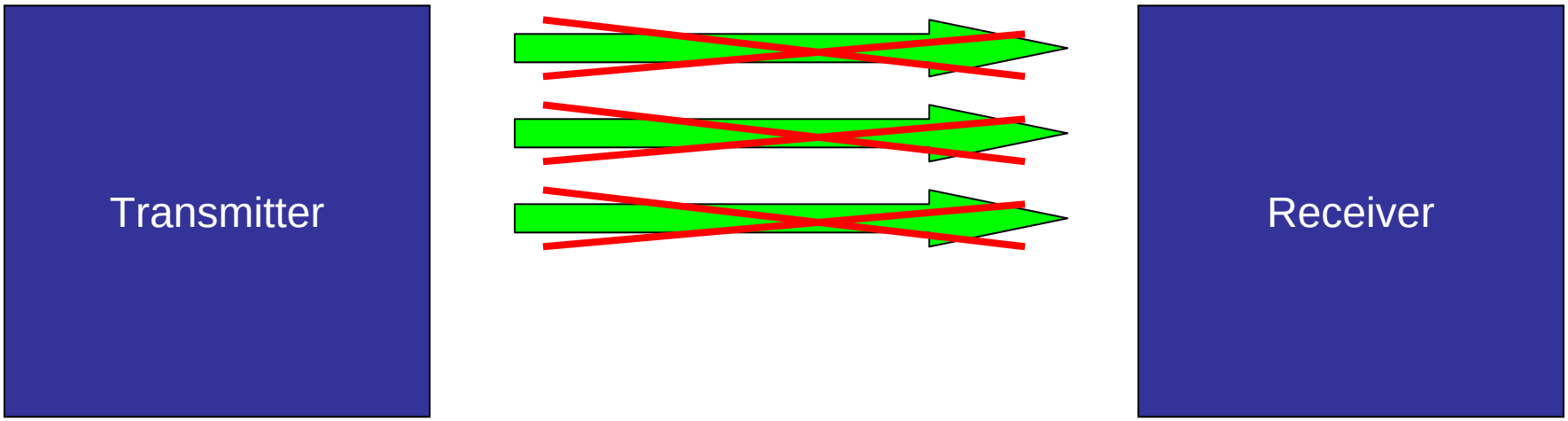
4 ms

CWUSB Host Reservation

4 ms

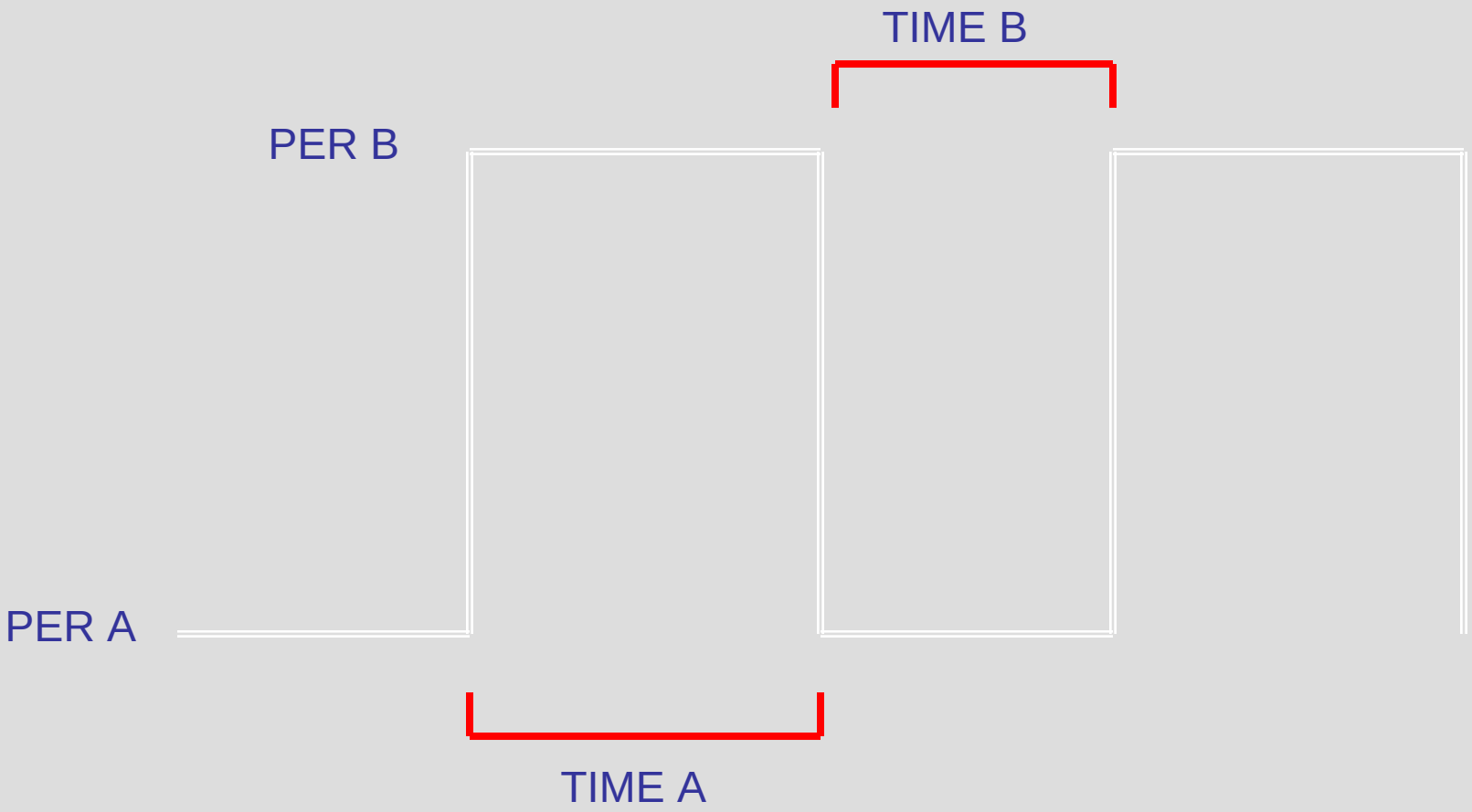
Other Reservation

# Retry Count

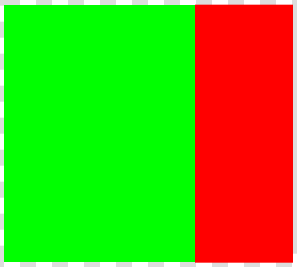



How Many Times Can a Packet Fail Before You Give Up and Move onto the Next Packet

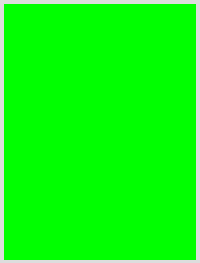
# Packet Error Rate



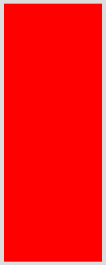
# Reservation Padding



WUSB Host Time Slot



Reservation Size If PER Is Zero



Padding For Retries

# Simulations

- Vary One Parameter at a Time
  - Reservation Padding
    - 10%, 20%, 30%, 40%
  - PER
    - Glitch Height, Width and Spacing
  - Buffer Size
    - 2x, 4x, 6x . . .
  - Retry Count
    - None, 2-7
  - Reservation Spacing
    - One Chunk vs Spread Out In Several Pieces

# 8 ms Buffer

| Error Rate | Glitch Error Rate | Drop (Buffer) | Drop (Retry Count) |
|---|---|---|---|
| 2 | 0 | 71 | 0 |
| 4 | 0 | 128 | 0 |
| 6 | 0 | 404 | 0 |
| 8 | 0 | 582 | 0 |
| 10 | 0 | 951 | 0 |
| 10 | 10 | 858 | 0 |
| 10 | 20 | 1213 | 0 |
| 10 | 30 | 1194 | 0 |
| 10 | 40 | 1402 | 0 |
| 10 | 50 | 1891 | 5 |
| 10 | 60 | 2517 | 19 |
| 10 | 80 | 4396 | 202 |
| 10 | 100 | 7813 | 1103 |

# 16 ms Buffer

| Error Rate | Glitch Error Rate | Drop (Buffer) | Drop (Retry Count) |
|---|---|---|---|
| 2 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 |
| 10 | 0 | 2 | 0 |
| 10 | 10 | 1 | 0 |
| 10 | 20 | 0 | 0 |
| 10 | 30 | 0 | 0 |
| 10 | 40 | 0 | 0 |
| 10 | 50 | 1 | 4 |
| 10 | 60 | 8 | 23 |
| 10 | 80 | 34 | 212 |
| 10 | 100 | 58 | 1148 |

# 24 ms Buffer

| Error Rate | Glitch Error Rate | Drop (Buffer) | Drop (Retry Count) |
|---|---|---|---|
| 2 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 |
| 10 | 10 | 0 | 0 |
| 10 | 20 | 0 | 0 |
| 10 | 30 | 0 | 0 |
| 10 | 40 | 0 | 0 |
| 10 | 50 | 0 | 7 |
| 10 | 60 | 0 | 11 |
| 10 | 80 | 0 | 214 |
| 10 | 100 | 0 | 1148 |

# 24 ms – Various Retry Counts

| Glitch Error Rate | Max Retry Limit | Drop (Retry) | Drop (Buffer) |
|---|---|---|---|
| 40 | 3 | 73 | 0 |
| 40 | 5 | 12 | 0 |
| 40 | 7 | 1 | 0 |
| 60 | 3 | 301 | 0 |
| 60 | 5 | 69 | 0 |
| 60 | 7 | 22 | 0 |
| 80 | 3 | 932 | 0 |
| 80 | 5 | 408 | 0 |
| 80 | 7 | 194 | 0 |
| 100 | 3 | 2057 | 0 |
| 100 | 5 | 1115 | 0 |
| 100 | 7 | 1148 | 0 |