



Certified Wireless USB Protocol

Dan Froelich
Intel Corporation

*Content also provided by:
John S. Howard, Intel Corporation*

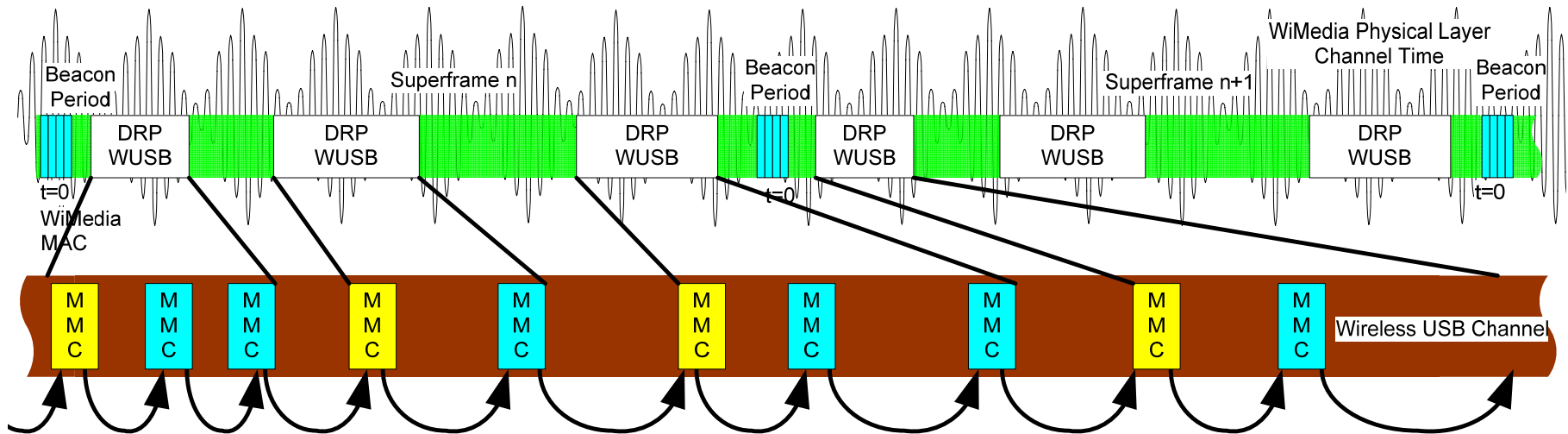


Agenda

- Protocol Components
- Data Transfers (general model)
- Device Notifications
- Flow Control

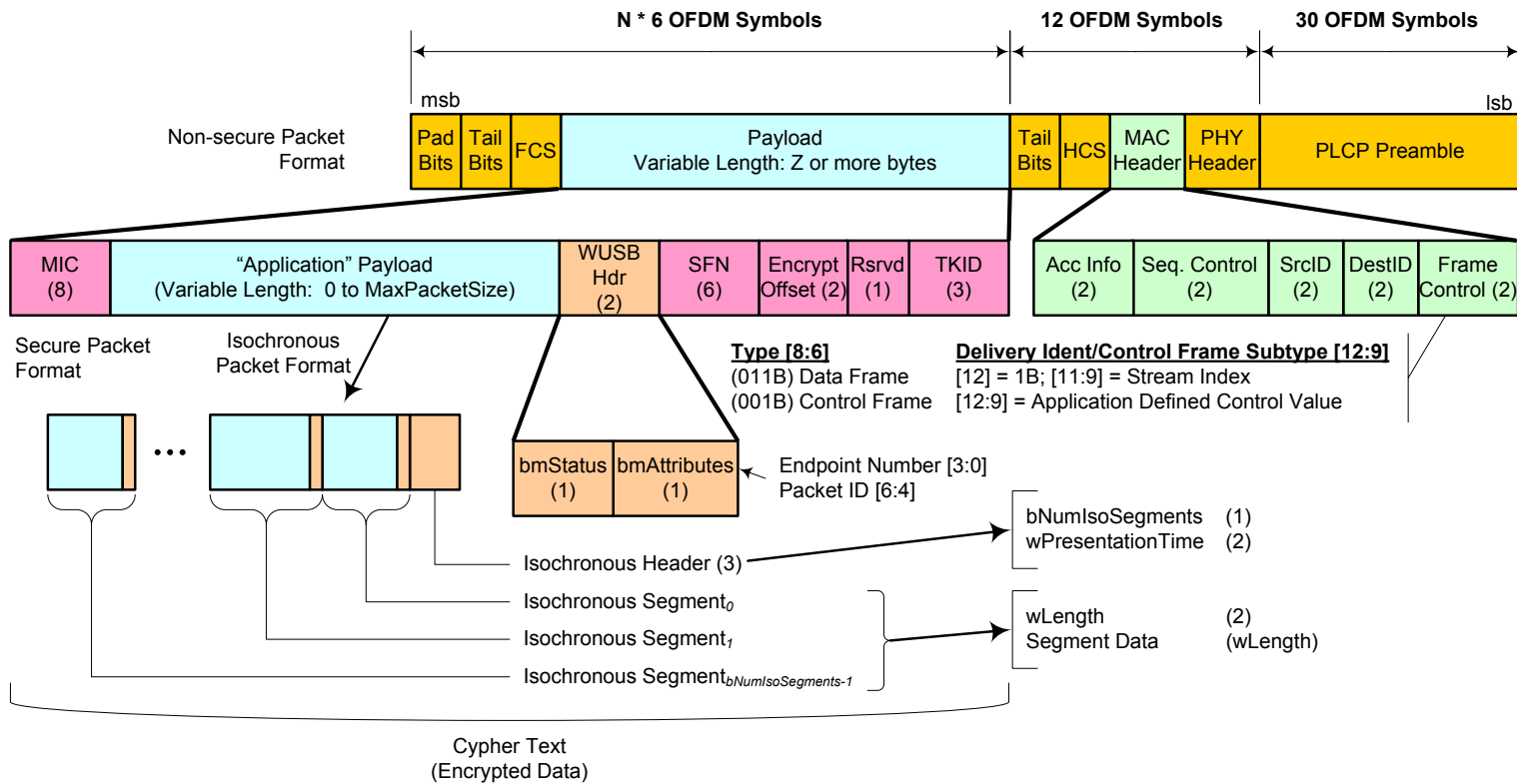
Certified Wireless USB Channel

Review



- The Certified Wireless USB Channel is encapsulated by the WiMedia 'channel'
- The Certified Wireless USB channel is continuous sequence of linked control packets transmitted by the Host during reserved WiMedia channel time
 - CWUSB time is reserved from WiMedia channel time (DRPs)
 - Called MMCs – Micro-schedule Management Commands
- All Certified Wireless USB Data communications are over the CWUSB Channel

Packet Layout

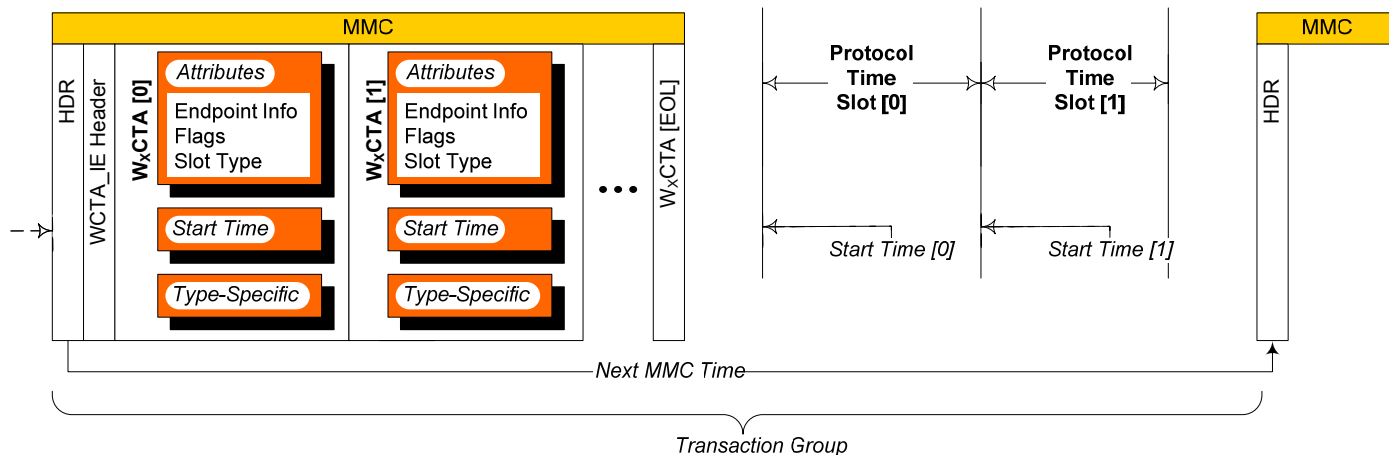


- WiMedia MAC Packet format
- Utilizes Secure 'frame' format

Certified Wireless USB Channel

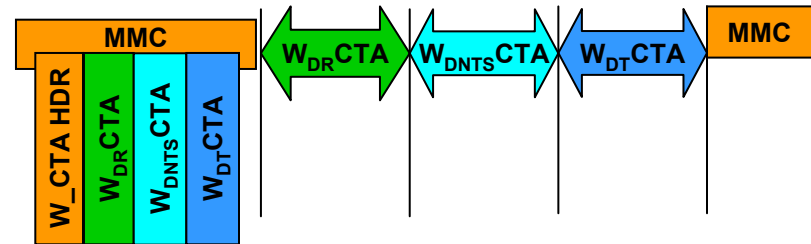


- MMC – Micro-scheduled Management Command
 - Basic structure for maintaining channel and accomplishing data communications
 - CWUSB Channel is MMC plus time to next MMC (Transaction Group)
 - Time slots after MMC used for data, handshake phases
- Slot time duration is difference between adjacent slot Start Times
 - OUTs device know when they can stop listening
 - INs devices transmit what was requested (or less)

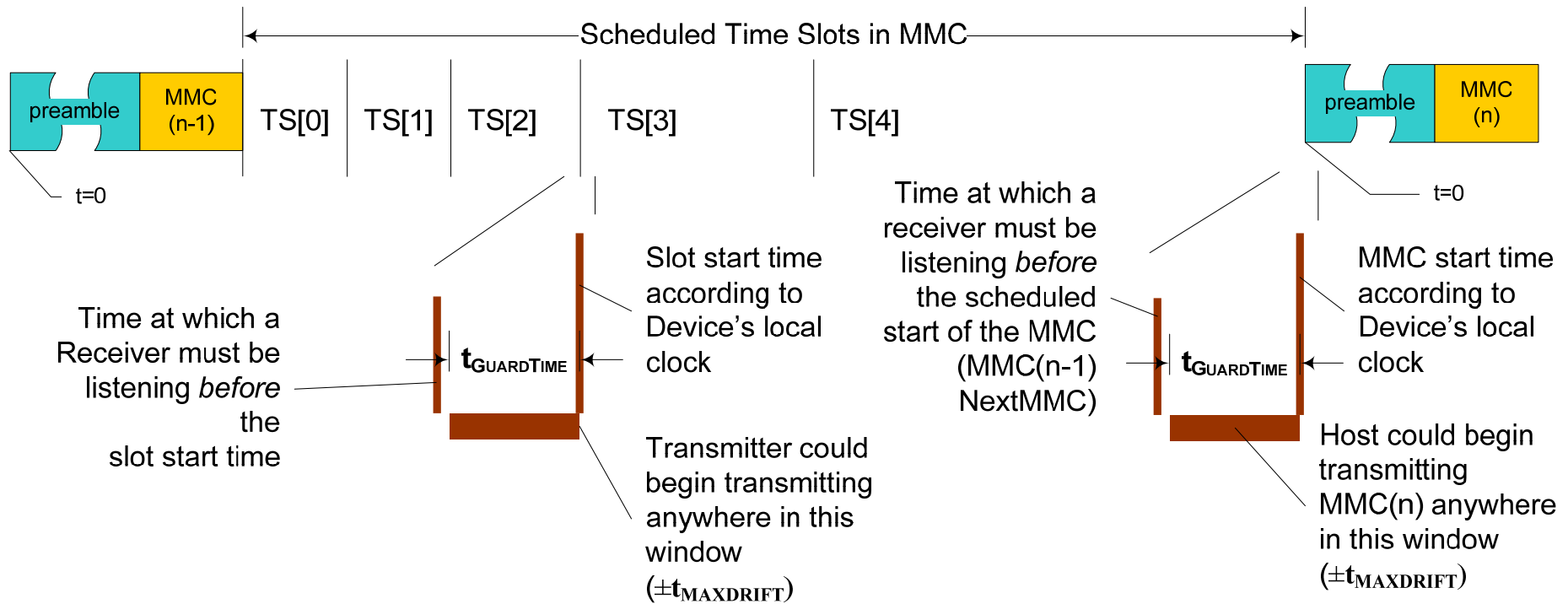


Time Slot Allocations

- Three basic slot types
 - $W_{DR}CTA$ (Device Receive)
 - $W_{DT}CTA$ (Device Transmit)
 - $W_{DNSTS}CTA$ (Device Notification Time Slot)
- $W_{DR}CTA$, $W_{DT}CTA$ s are used for Transaction Protocol
 - Control, Bulk, Interrupt and Isochronous data streams
 - Access is : guaranteed time slot
 - Schedule based on client application data flow needs
- $W_{DNSTS}CTA$ s used to emulate USB signaling events
 - Connect, disconnect, etc.
 - Access is: contention based - Slotted Aloha
 - Host schedules sufficient number of slots to provide adequate service for the cluster

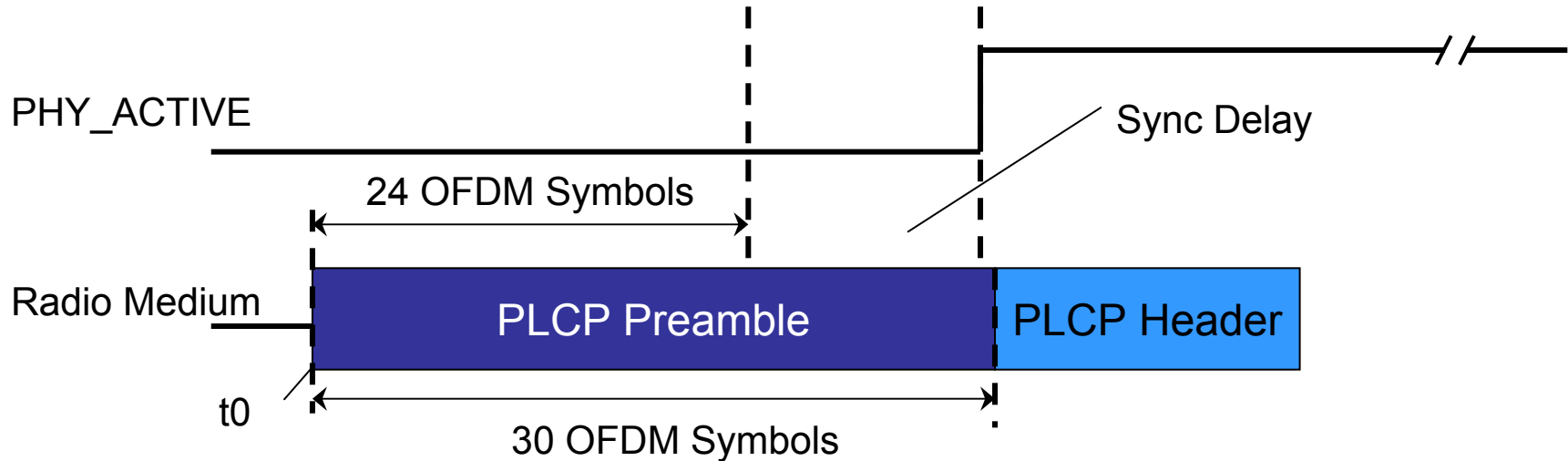


Transaction Group Protocol Synchronization



Transaction Group

Protocol Synchronization (cont.)

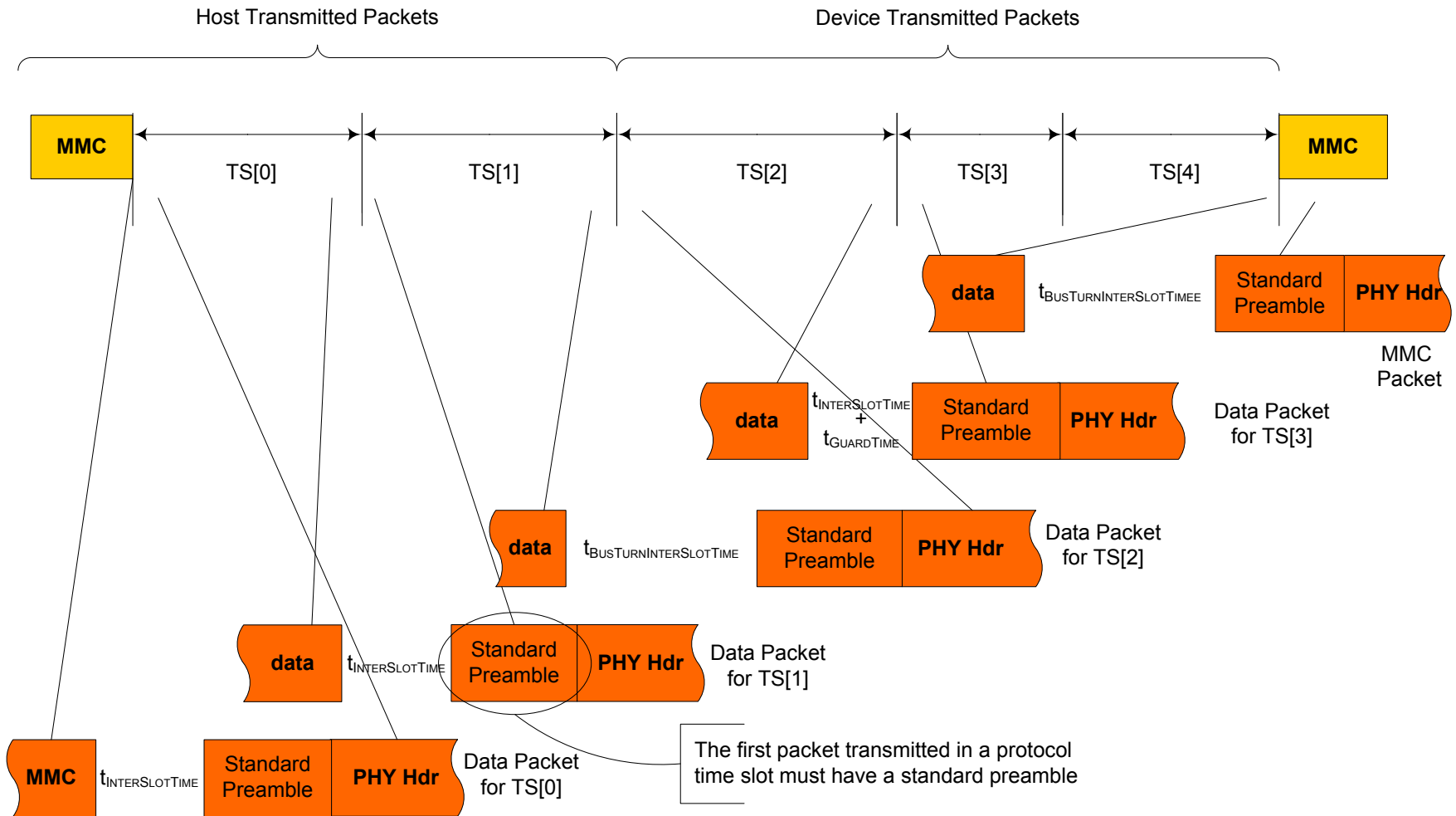


- t_0 is the start time of the first symbol of the MMC's preamble
- Devices can 'calculate' this from the edge of PHY_ACTIVE

PLCP : Physical layer convergence protocol

OFDM : Orthogonal frequency division multiplexing

Time Slot Scheduling Requirements

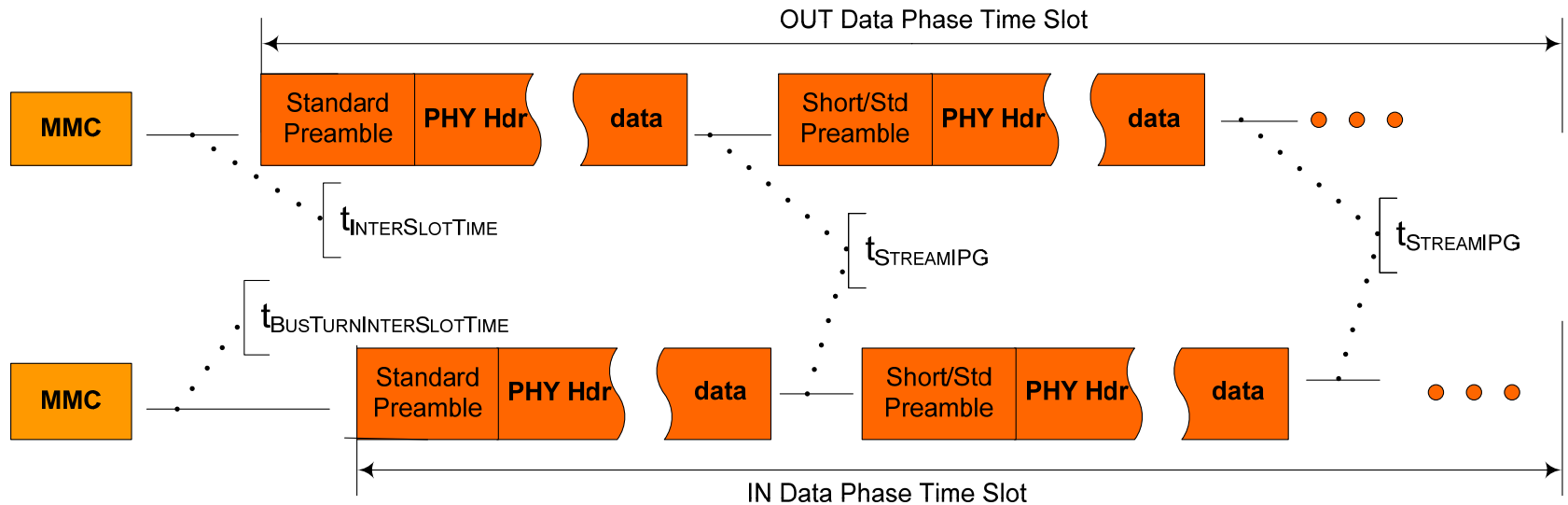


Data Burst Timing



- MMC to first Data OUT: MIFS (1.875 μ s)
- MMC to first Data IN: Bus Turn Time
 - Bus Turn + Guard Time (SIFS + 1 μ s) (11 μ s)
- IPGs between burst packets is fixed (MIFS)
- First packet Preamble must be Standard Length
 - Host determines subsequent preamble to use

MIFS :Minimum Inter-frame Spacing
IPG : Inter-packet Gap
SIFS : Standard Inter-frame Spacing





Agenda

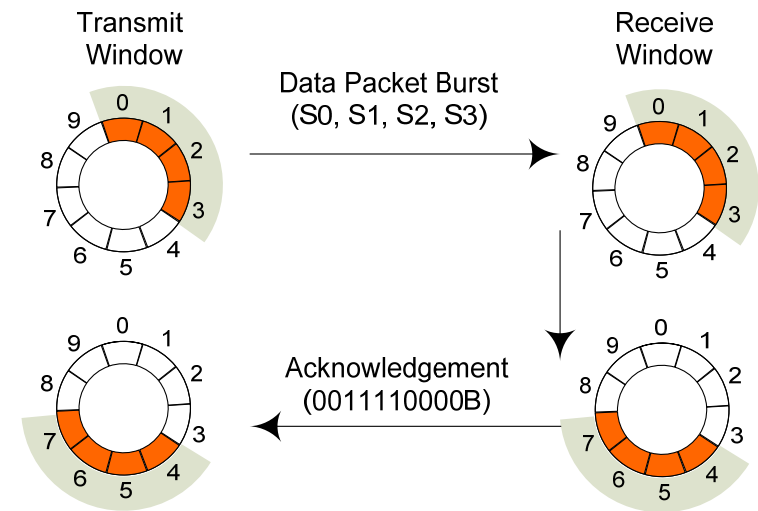
- Protocol Components
- Data Transfers (general model)
- Device Notifications
- Flow Control

Data Stream Synchronization

Basic Model



- Protocol-level support for reliable data delivery
- Identifies:
 - Data ordering requirements
 - Guarantees advancement of data stream only after reliable data delivery
- Terms:
 - Maximum Burst Size
 - Maximum Sequence
 - Maximum Sequence Distance



- Transmitter sends data packets associated with Transmit window Sequence numbers
- Receiver acknowledges with new receive window (what is available now)
- Protocol rules for recovering lost packets, avoiding sequence range wrap, etc.

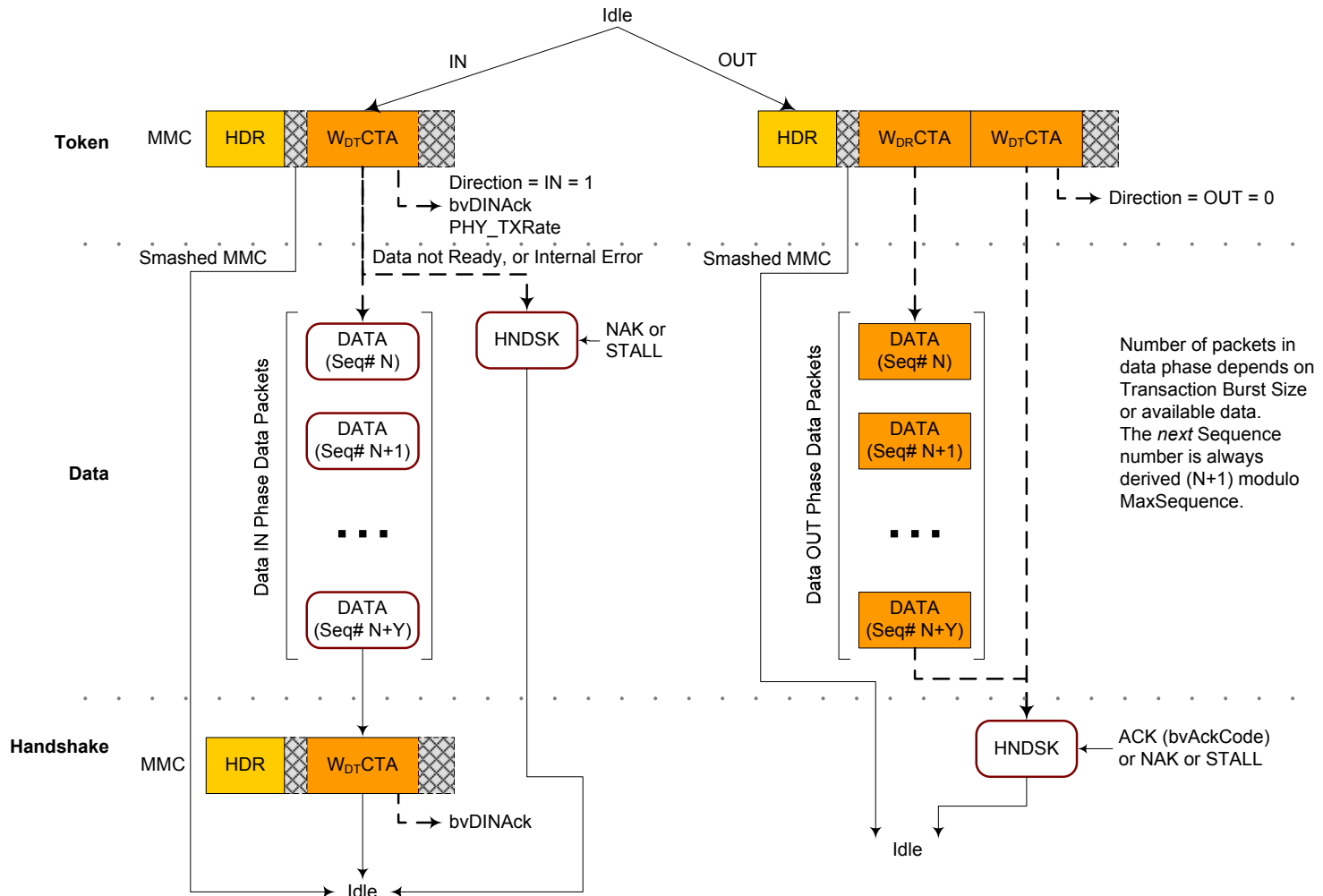
Transaction Transmit Bit Rate



- MMCs are transmitted at ‘base’ signaling rate (53.3 Mb/s)
- Host dictates transmit bit rate for data and handshake phase transmissions
 - Transmit bit rate applies to entire phase
 - OUTs: host uses only TBRs supported by device (assumes Tx/Rx symmetry)
 - INs: $W_{DTCTA.PHY_TXRate}$ directs which TBR device must use during protocol time slot
- Handshakes:
 - Host must use ‘base’ signaling rate for a handshake time slot (OUTs)
 - Host must ensure a data phase time slot is always large enough to accommodate a handshake transmitted at ‘base’ signaling rate

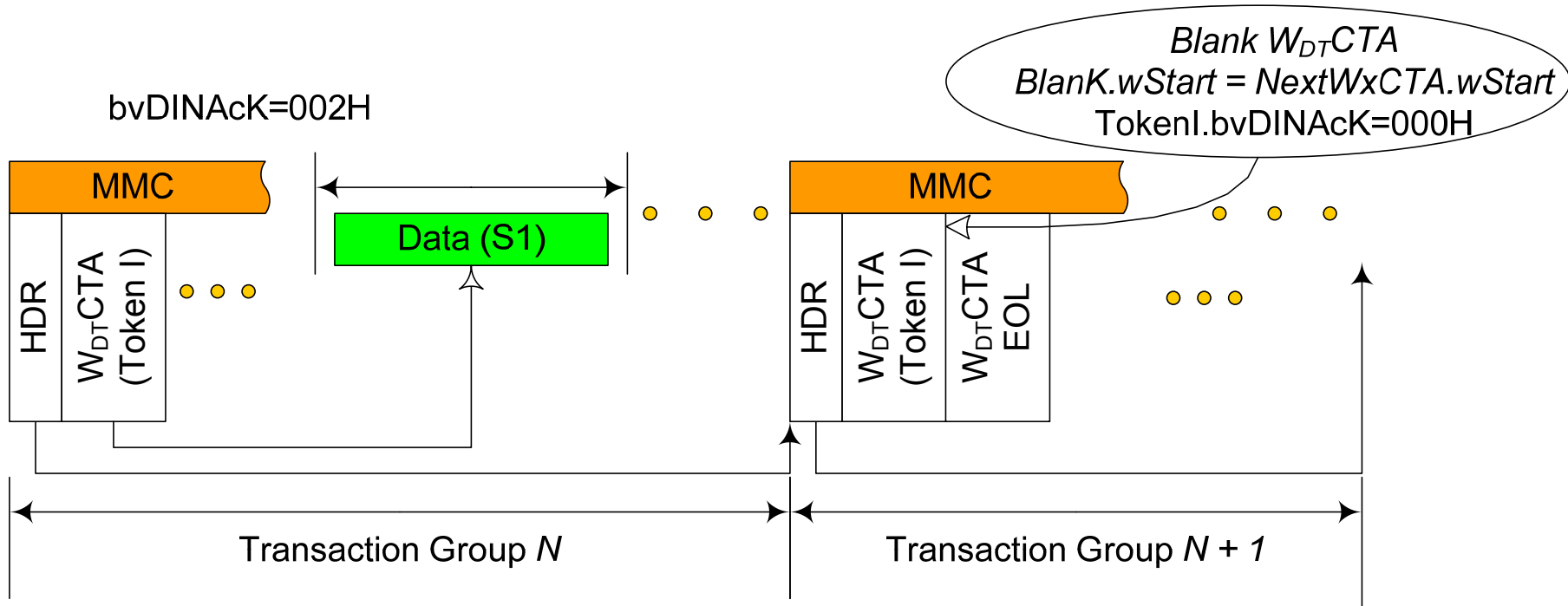
Transaction Format

General Model



Data IN Transfer

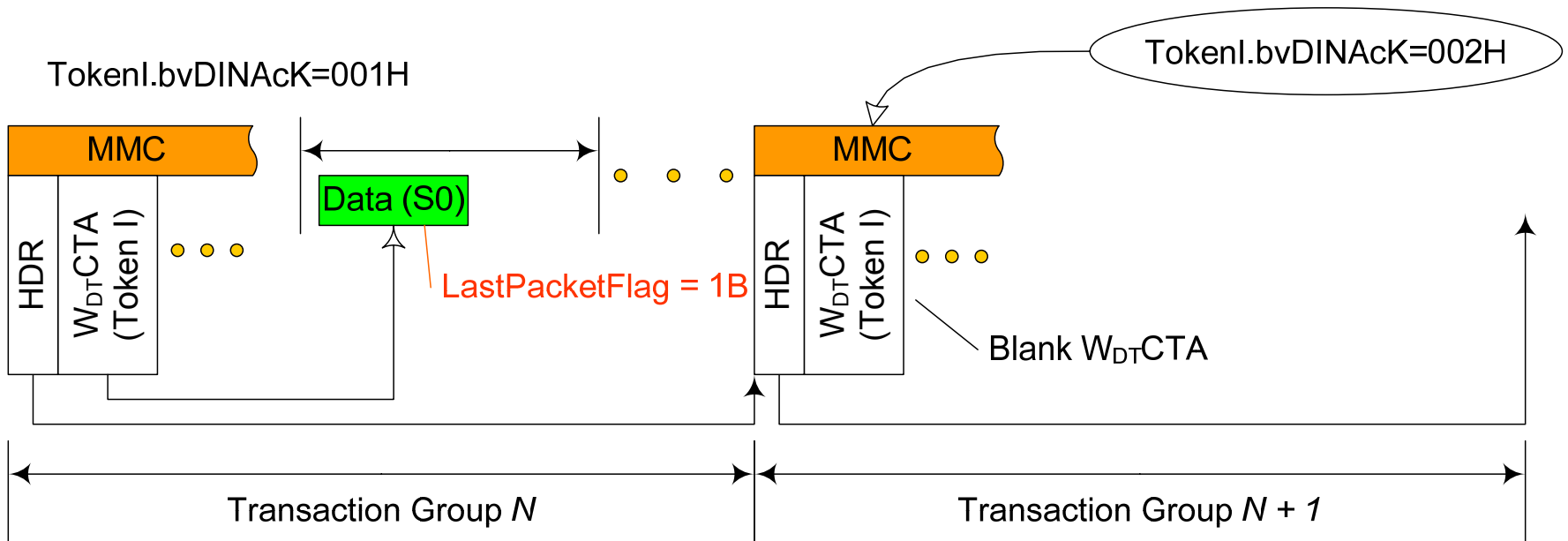
End Of Transfer Example



- At end of Transfer, need to acknowledge last data phase
- Uses 'Blank' IN Token (W_{DTCTA})

Data IN Transfer

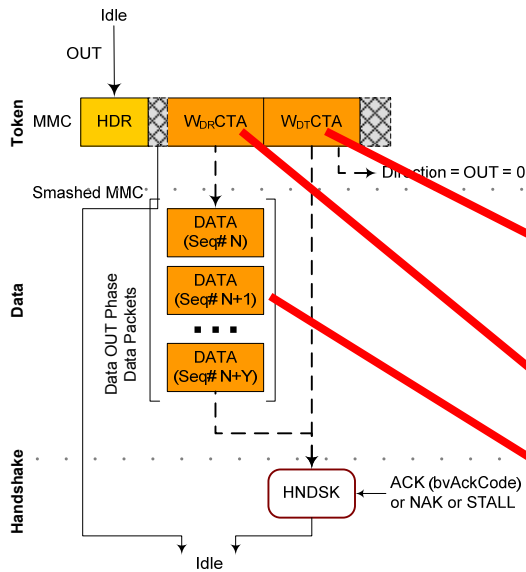
Short Packet Example



Host may commence next transfer immediately (if buffer available)

Data OUT Transfers

Streaming Example



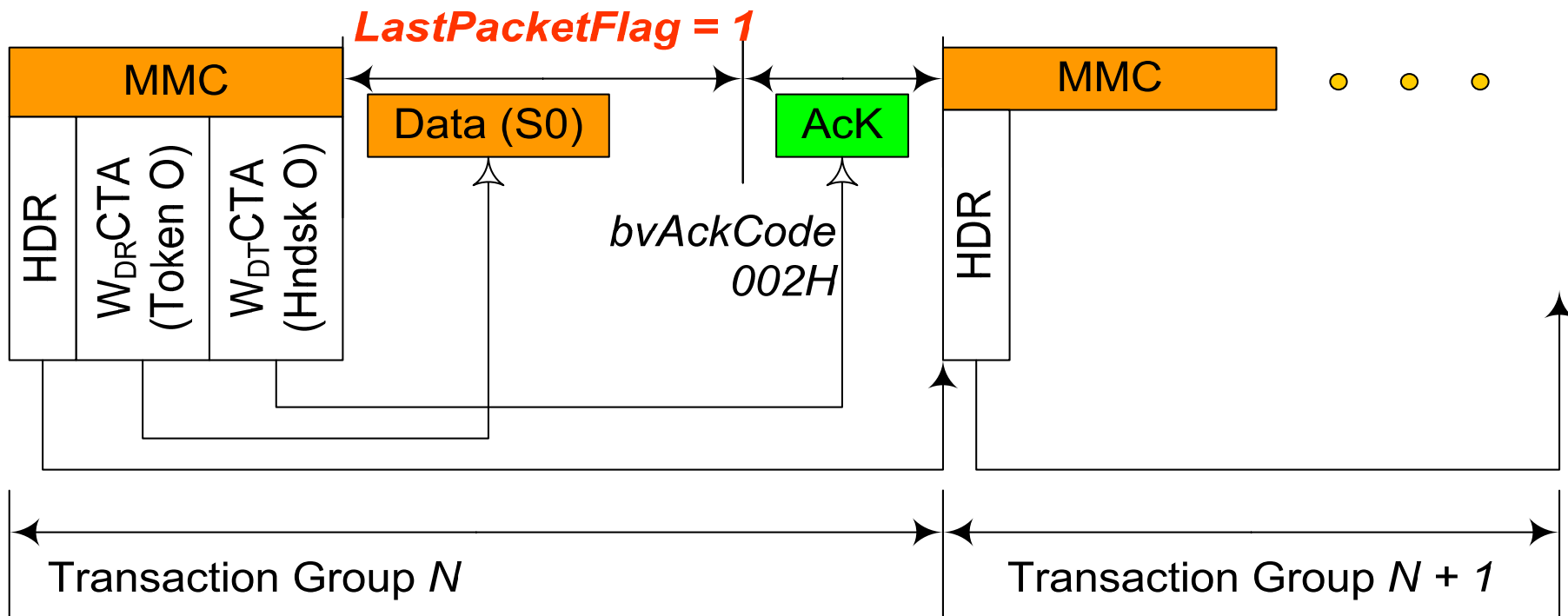
WUSB Packet 398	MMC	Dir	Host Id	Chan TS	IE Id	CTA 0	Addr	Endp	CTA 1	# Notif TS's	CTA 2	Addr	Endp	EDir	bvAckCode
		-->	0xBEEF	0:000	WCTA_IE	DR	1	1	DNTS	16	DT (Hnd)	1	1	OUT	0x00000000
WUSB Packet 399	OUT	Dir	Host Id	DATA	Addr	Endp	EDir	Seq#	Last	Data	Frm Duration	PHY_ACT	Idle	Time Stamp	
		>	0xBEEF	0	1	1	OUT	0	1	31 bytes	15.000 µs	10.714 µs	479.668 µs	7.466400520	
WUSB Packet 400	Hand	Dir	Host Id	ACK	Addr	Endp	EDir	bvAckCode	Frm Duration	PHY_ACT	Idle	Time Stamp			
		--	0xBEEF	1	1	1	OUT	0x0001FFFE	1.875 µs	7.308 µs	2.630 ms	7.466890902			

Handshake Acks the Seq # (0) by advancing (by one) receive window : $0x0000FFFF \ll 1 = 0x0001FFFE$

Disclaimer : early prototype host and devices used for traffic capture

Data OUT Transfers

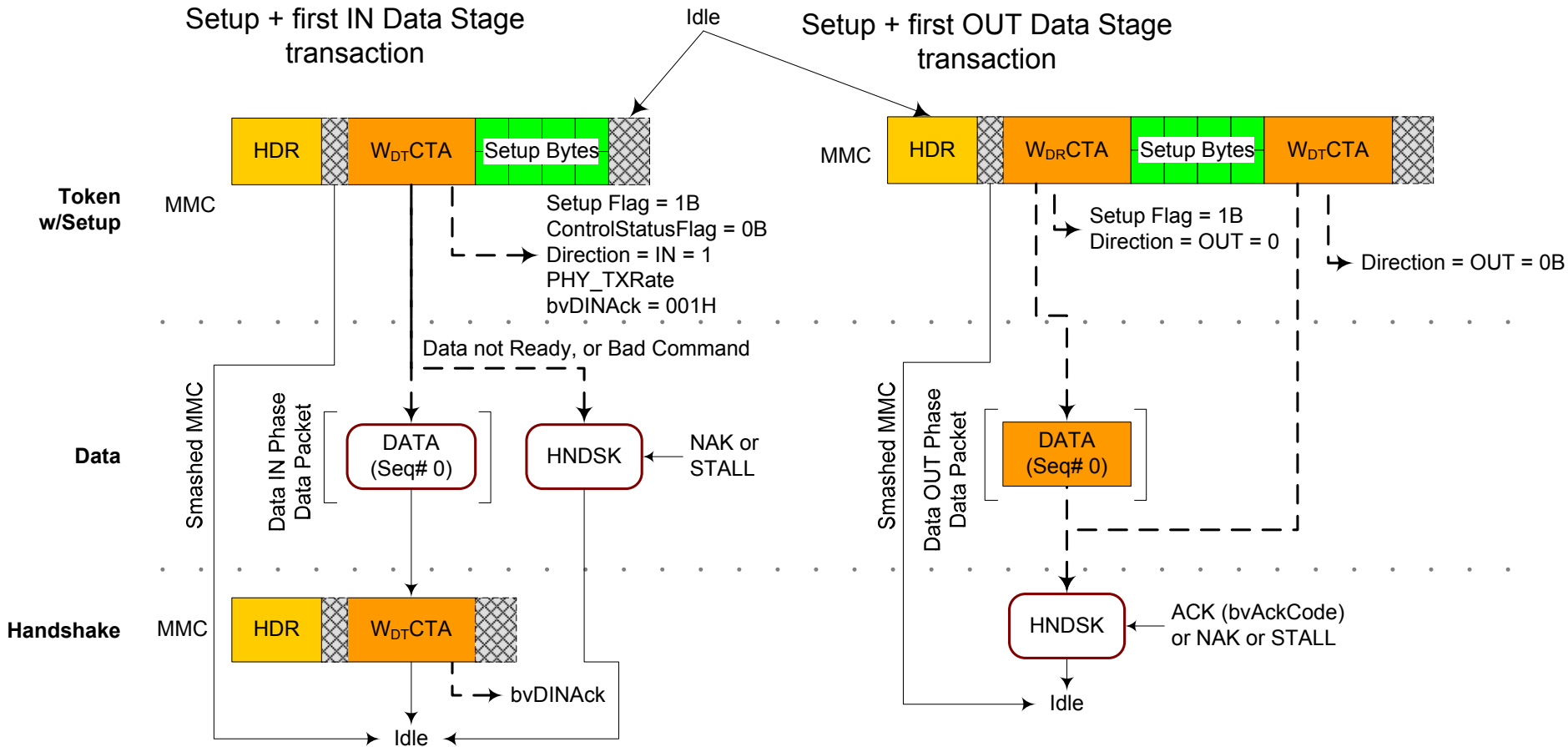
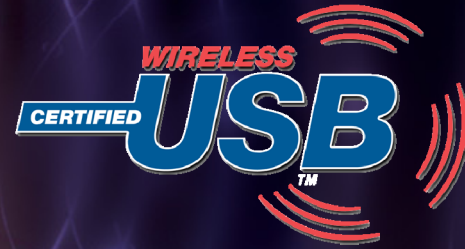
Short Packet Example



Host may commence next transfer immediately (if data available)

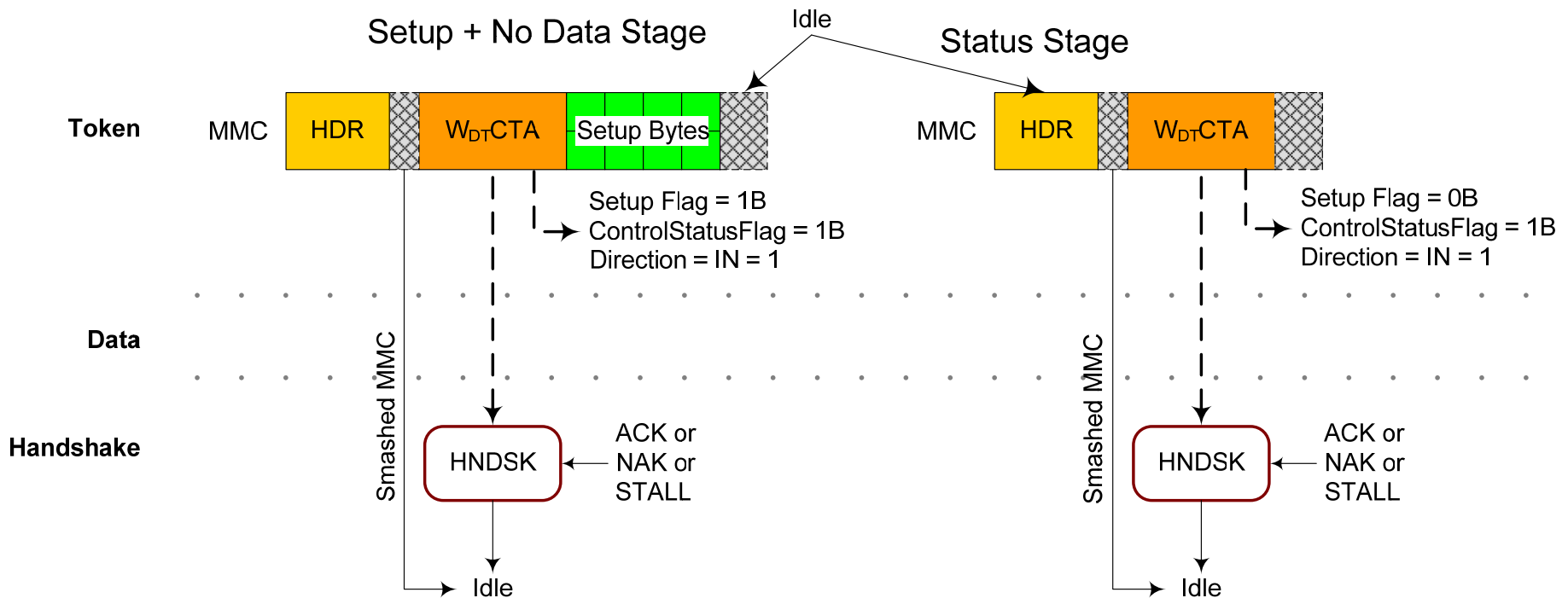
Control Transaction Format

Setup Stage w/Data Stage



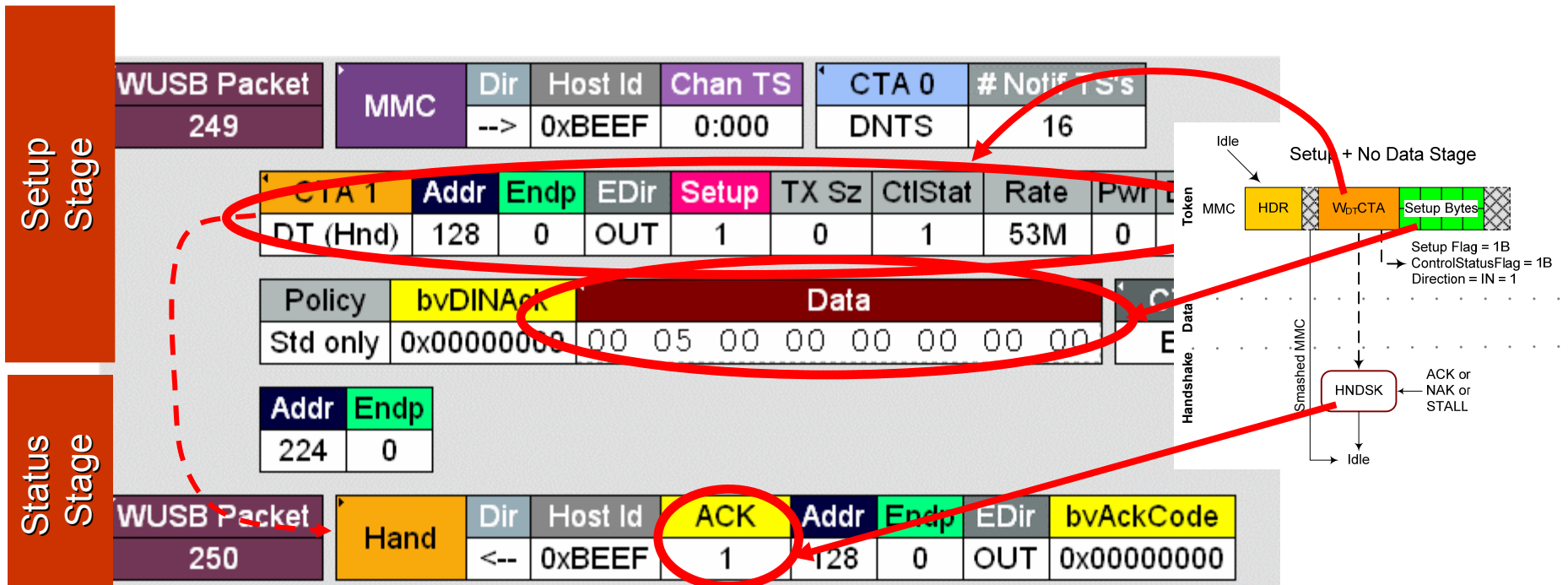
Control Transaction Format

Setup w/No Data Stage; Bare Status Stage



Control Transfer

Example (Set Address - No Data Stage)



Disclaimer : early prototype host and devices used for traffic capture

Control Transfer

Example (Control Write - SetKey)



Setup Stage

Packet	Dir	Host Id	Chan TS	CTA 0	Addr	Endp	Setup	Start	Data
313	-->	0xBEEF	0:000	DR	1	0	1	150µs	25 07 00 22 00 00 1c 00

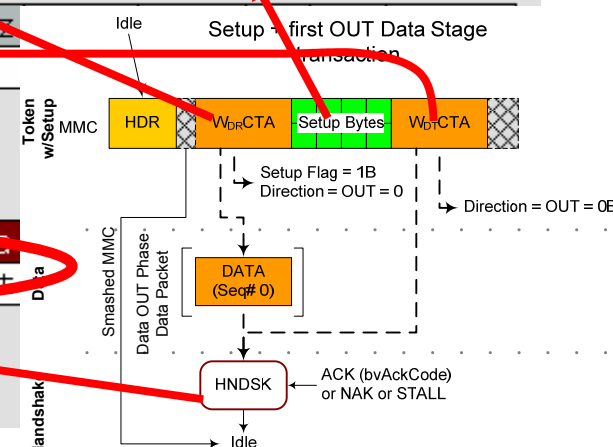
CTA 1	Start	# Notif TS's	CTA 2	Addr	Endp	EDir	Start	TX Sz
DNTS	259µs	16	DT (Hnd)	1	0	OUT	643µs	0

bvDINAck	CTA 3	Addr	Endp	Start
0x00000000	EOL	224	0	716µs

Data Stage

Packet	Dir	Host Id	DATA	Addr	Endp	EDir	Seq #	Last	Data
314	-->	0xBEEF	0	1	0	OUT	0	1	28 byt

Packet	Dir	Host Id	ACK	Addr	Endp	EDir	bvAckCode
315	<--	0xBEEF	1	1	0	OUT	0x00000000



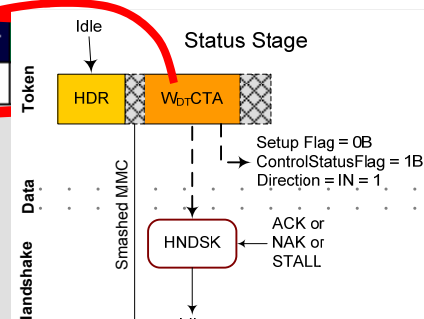
In a subsequent transaction group... the status stage

Status Stage

Packet	Dir	Host Id	Chan TS	CTA 0	Start	# Notif TS's	CTA 1	Addr
318	-->	0xBEEF	0:000	DNTS	212µs	16	DT (Hnd)	1

CtlStat	Rate	Pwr	Brst	Policy	bvDINAck	CTA 2	Addr	Endp	Start
1	53M	0	1	Std only	0x00000000	EOL	224	0	668µs

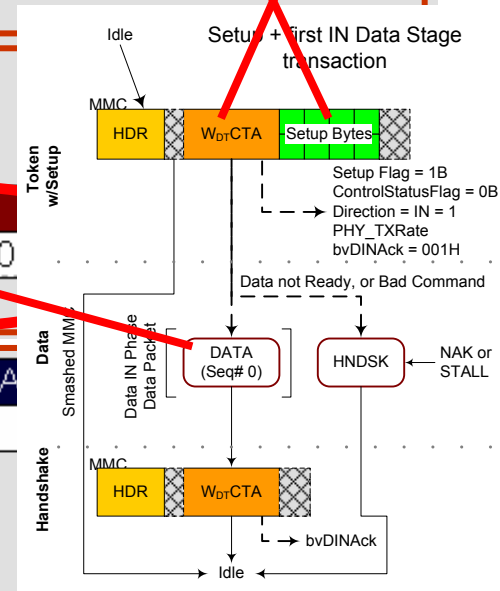
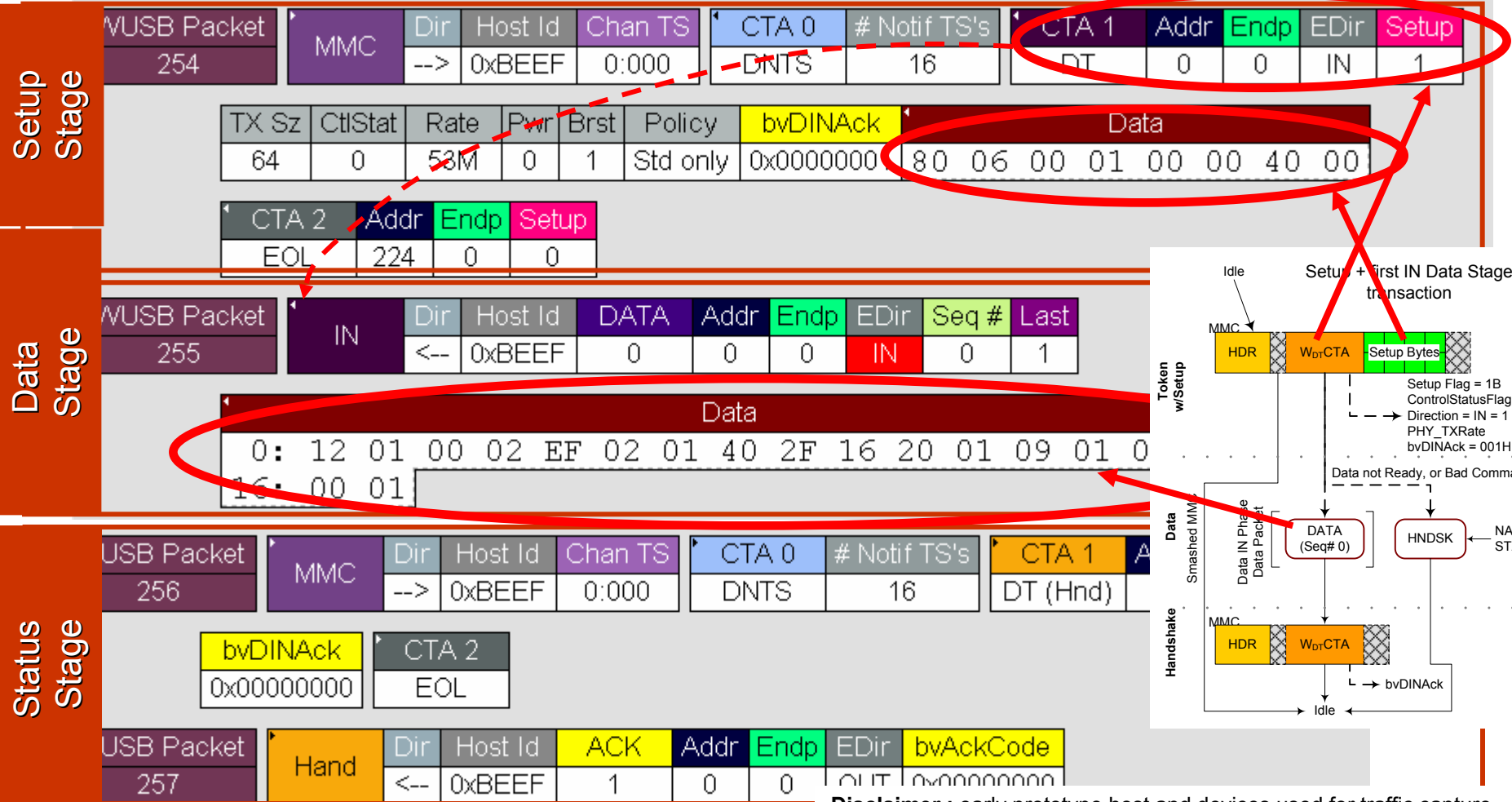
Packet	Dir	Host Id	ACK	Addr	Endp	EDir	bvAckCode
319	<--	0xBEEF	1	1	0	OUT	0x00000000



Disclaimer : early prototype host and devices used for traffic capture

Control Transfer

Example (Control Read – Get Descriptor)



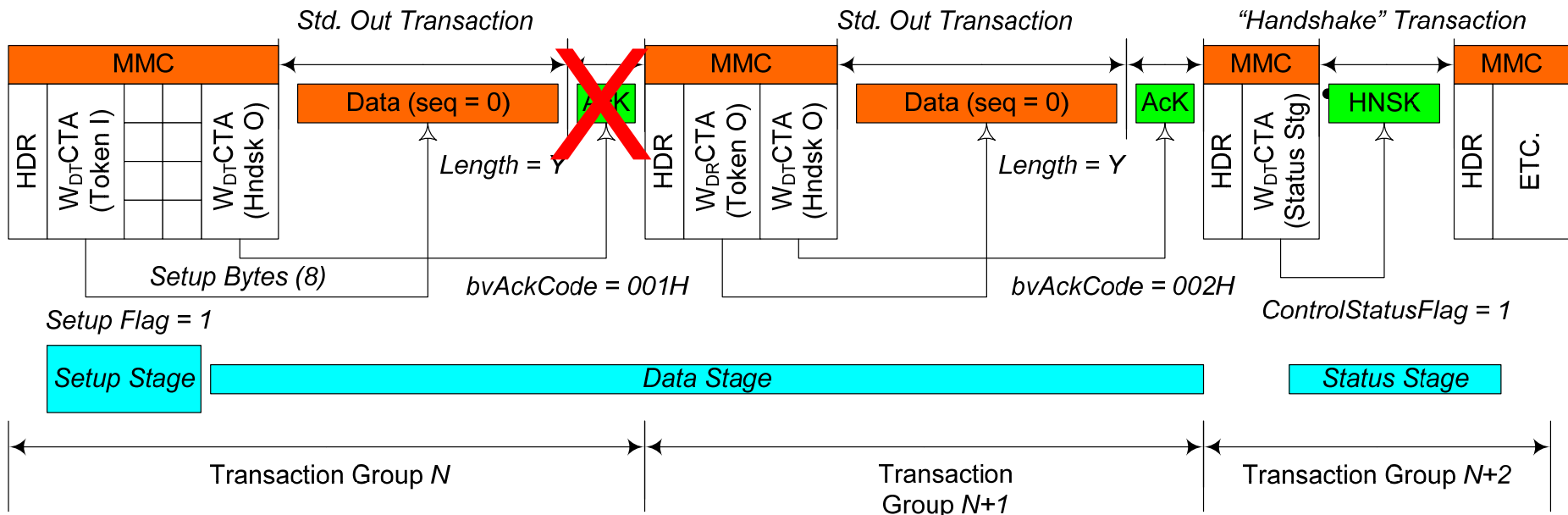
Disclaimer : early prototype host and devices used for traffic capture

Control Transfer

Example (Control Write – Retry Policy)



- No response from device on Setup + OUT
- Retry only first OUT of Data Stage
 - Device response when missed SETUP is a STALL handshake





Agenda

- Packet Layouts
- Transaction Groups
- Timing Constraints
- Data Transfers (general model)
- **Device Notifications**
- Flow Control

Device Notifications Time Slots

DNTS



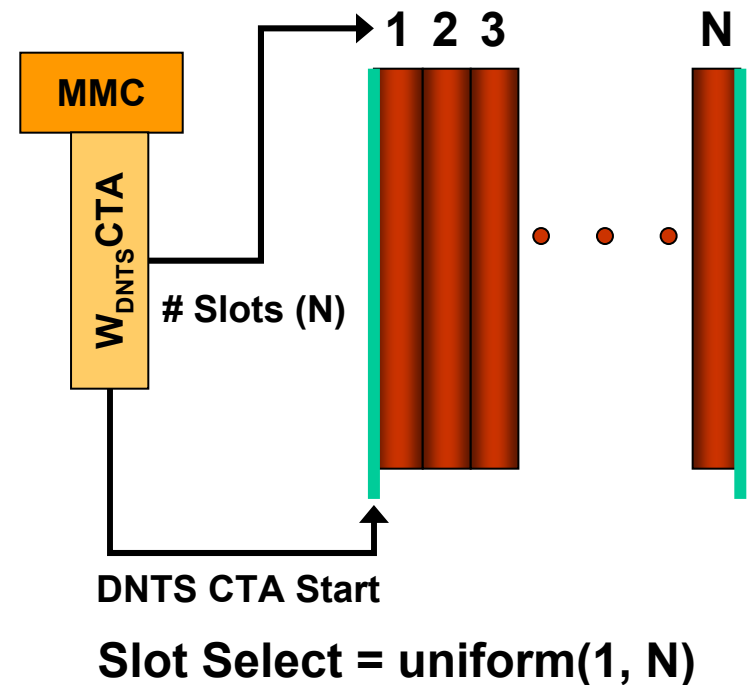
- Devices are only allowed transmitters
- Device notification messages (to the CWUSB Host) are only allowed data communications
- Nature of notification messages are that they are asynchronous and infrequent
- Fixed Maximum Size for notification messages
 - Take advantage of this in the access method
- Host policy to manage 'efficiently'
 - Unused time is just lost to CWUSB data stream

DNTS Access Method

Contention Based: Slotted Aloha



- Fixed (maximum) sized message
 - DNTS is a window of uniform sized message slots
- W_{DNTS} CTA indicates number of message slots
- Device selects a message slot using a uniformly distributed random integer value (in range 1 to N)
- Device transmits message a message slot start time

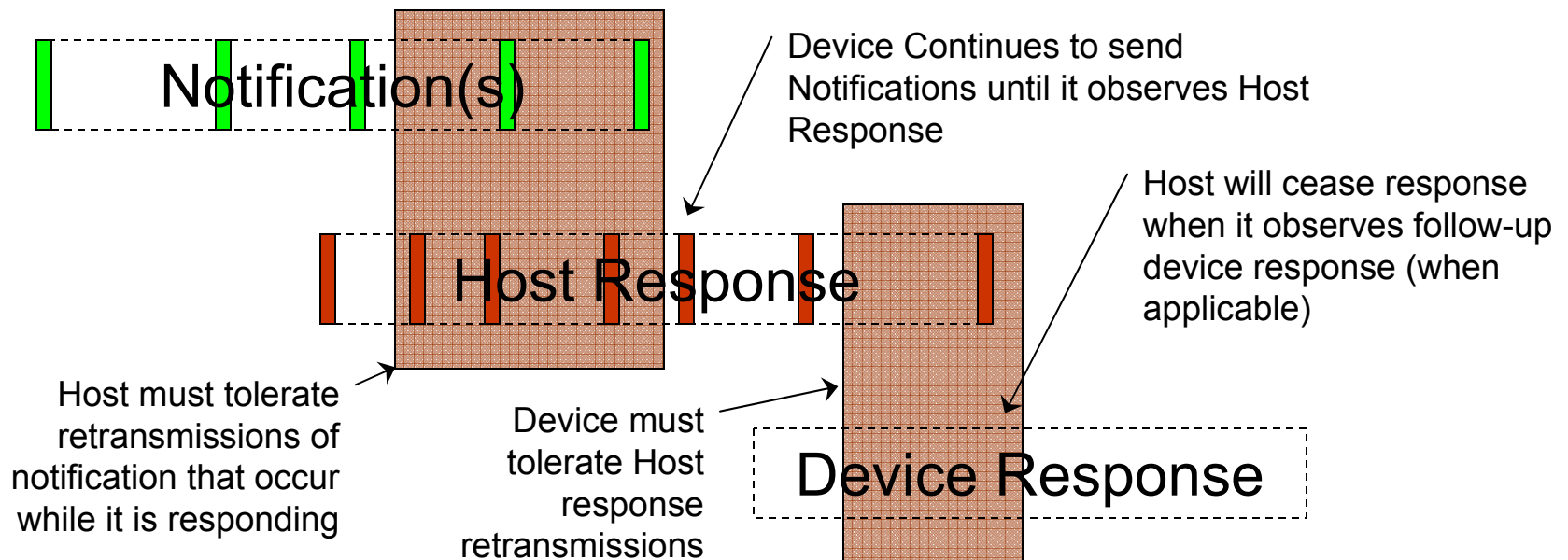


Device Notifications

Reliability



- Protocol for Device Notifications is designed for unreliability of channel





Agenda

- Packet Layouts
- Transaction Groups
- Timing Constraints
- Data Transfers (general model)
- Device Notifications
- Flow Control

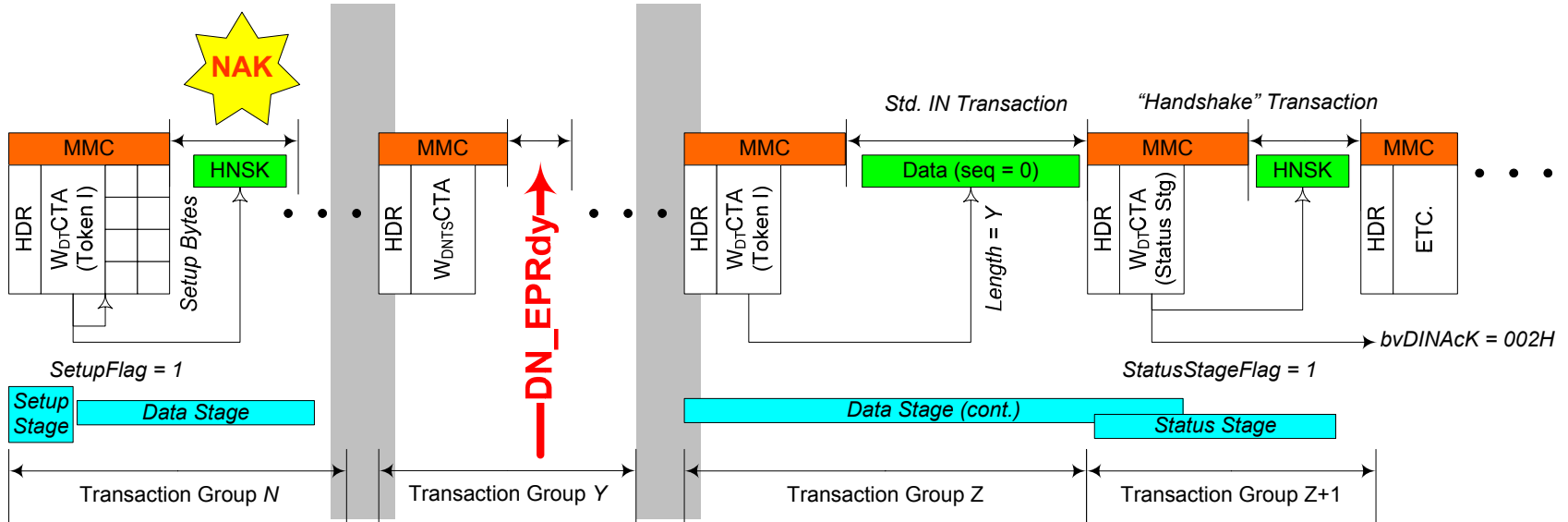
Flow Control



- Used on All transfer types
- Device provides flow-control information in it's response to the host
 - OUTs: the flow control response can be in the ACK handshake
 - ACK the data, but tell host that there is no more room
 - Or endpoint can use NAK handshake
 - Ins: the flow control response is with NAK handshake
- Flow control status from Endpoint results in removal of endpoint from 'Active' schedule
- Resumption of service to a flow-controlled endpoint depends on transfer type
 - Bulk & Control
 - Device delivers a Endpoint Ready (DN_EPRdy) notification to the host during an DNTS
 - Interrupt & Isochronous
 - Next service interval

Control Transfer

Example (Control Read – Flow Control)



- Host stops polling endpoint on NAK
- Device notifies it is ready with DN_EPRdy
- Host resumes polling endpoint
 - Retries data phase transaction in this example



Developers Conference 2006

Taipei, Taiwan