Data Bursting

Liang Lixin
Philips Semiconductors

Content also provided by:
Bart Vertenten, Philips Semiconductor

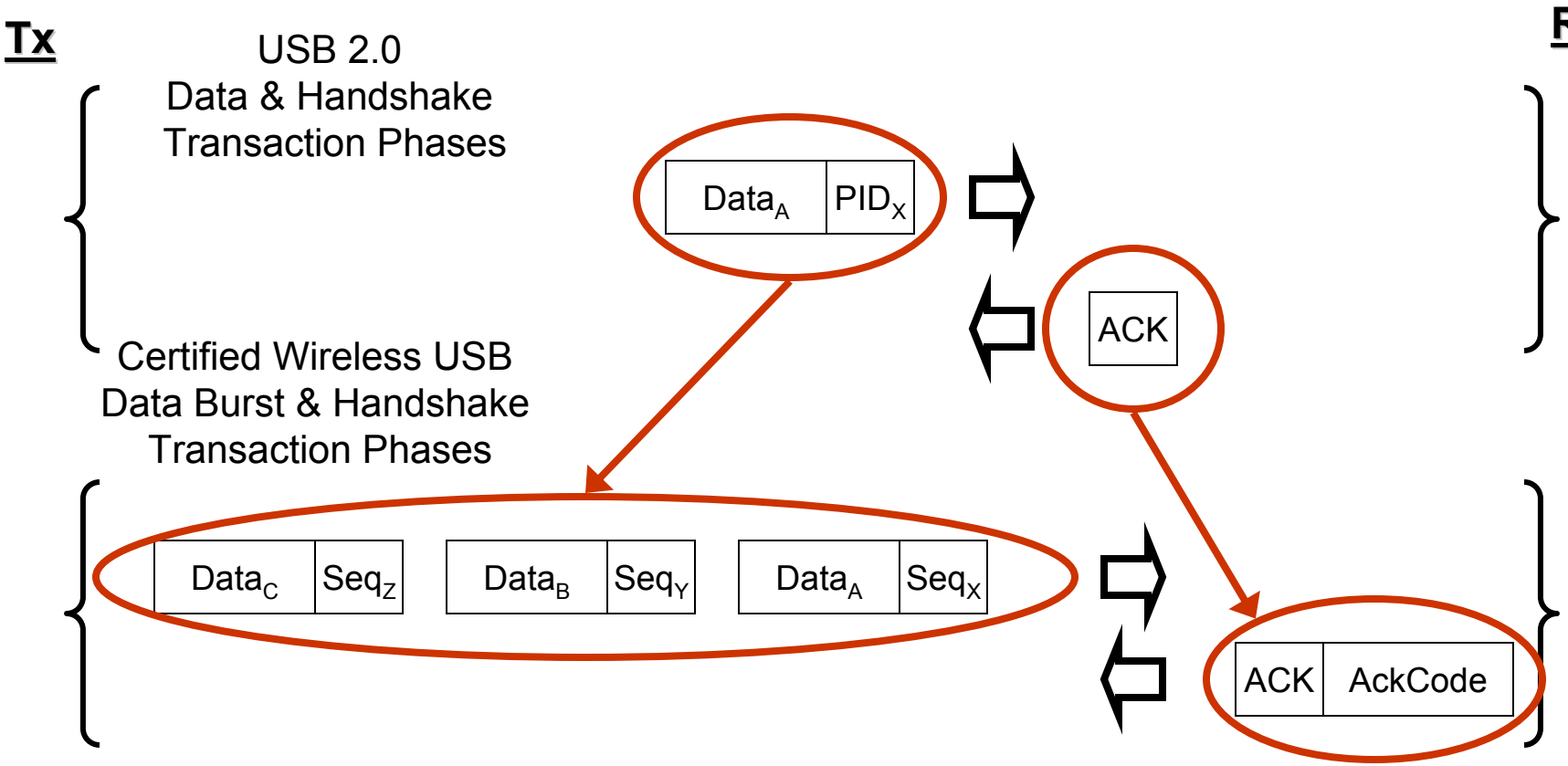
Outline

- Basic model
- Examples
  - OUT endpoint
  - IN endpoint
  - Special cases
- Summary

Outline

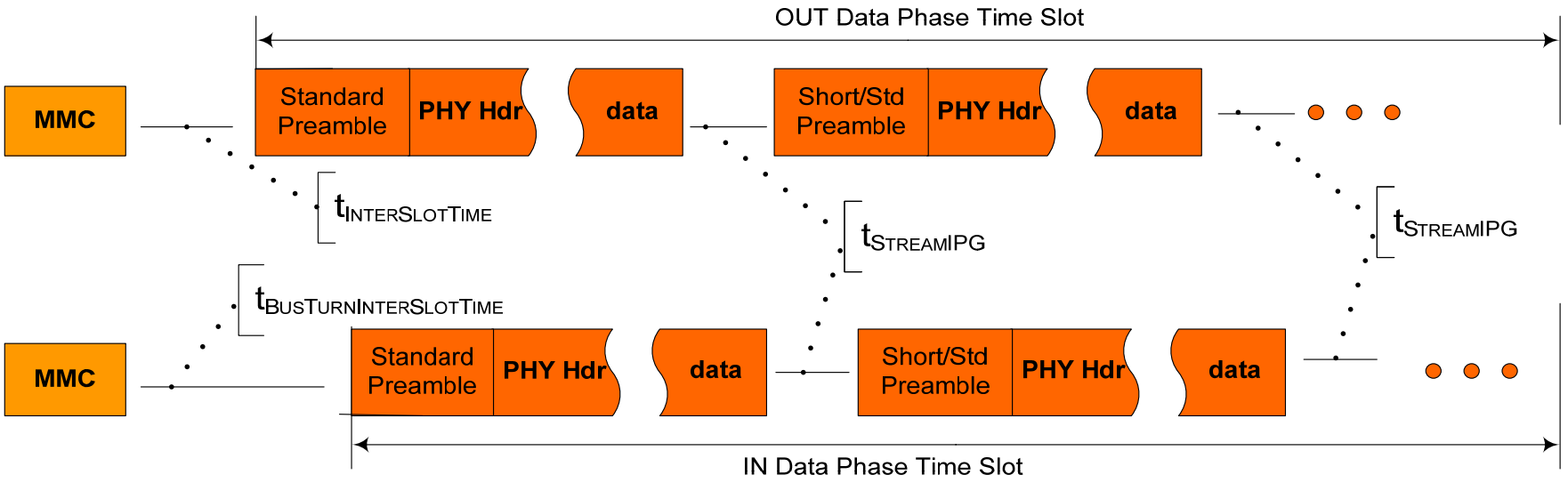
- Basic model
- Examples
  - OUT endpoint
  - IN endpoint
  - Special cases
- Summary

Data Bursting
Basic Idea



- Allow multiple data packets per transaction

Data Bursting
Timing



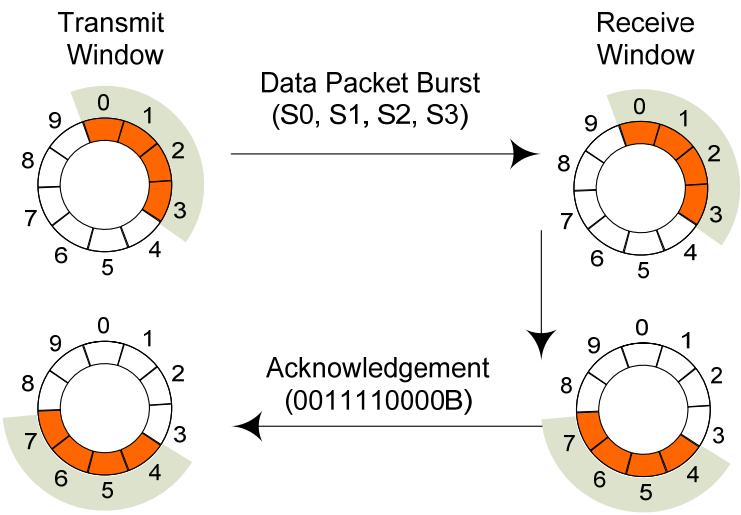
- Time to transmit (including overhead) :
  - 16 x 1 IN data packet of 1024 bytes @ 480Mb/s = 1132 us
  - 1 x 16 IN data packets of 1024 bytes @ 480Mb/s = 525 us

Goals

- Efficient use of media
  - Only dropped packets are retransmitted
  - A burst must be able to include packet retries and new packets
  - Do not send more packets than receiver can accept
- Burst size configurable by device
  - Trade-off between throughput and size
  - Supports bursting N data packets per time slot
    - Burst = N packets of Maximum Packet Size
    - N = {1 to 16}

Data Stream Synchronization
Basic Model

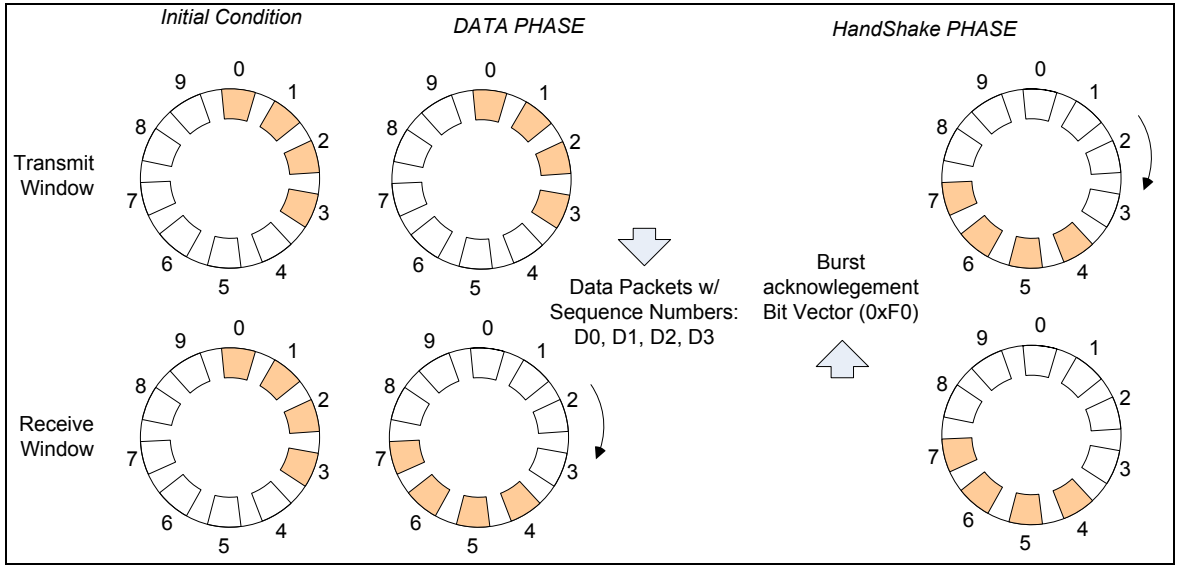
- Protocol-level support for reliable data delivery
- Identifies:
  - Data ordering requirements
  - Guarantees advancement of data stream only after reliable data delivery
- Terms:
  - Maximum Burst Size
  - Maximum Sequence
  - Maximum Sequence Distance



- 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.

AckCode

- ‘1’ indicates that receiver has buffer space
- ‘0’ indicates
  - Packet accepted
  - Buffer not available



Transmitter Rules

- Must initialize to start with sequence number 0
- Must initialize transmit map to MaxBurstSize
- Packets per transaction must be (adjusted) MaxPacketSize
  - Unless last is residual
- Send # packets as indicated in AckCode
  - Unless residual
- Send packets in correct sequence order
  - Mark with sequence values in ascending sequence order
- Retries for missed packets must be sent before new packets (in a burst)

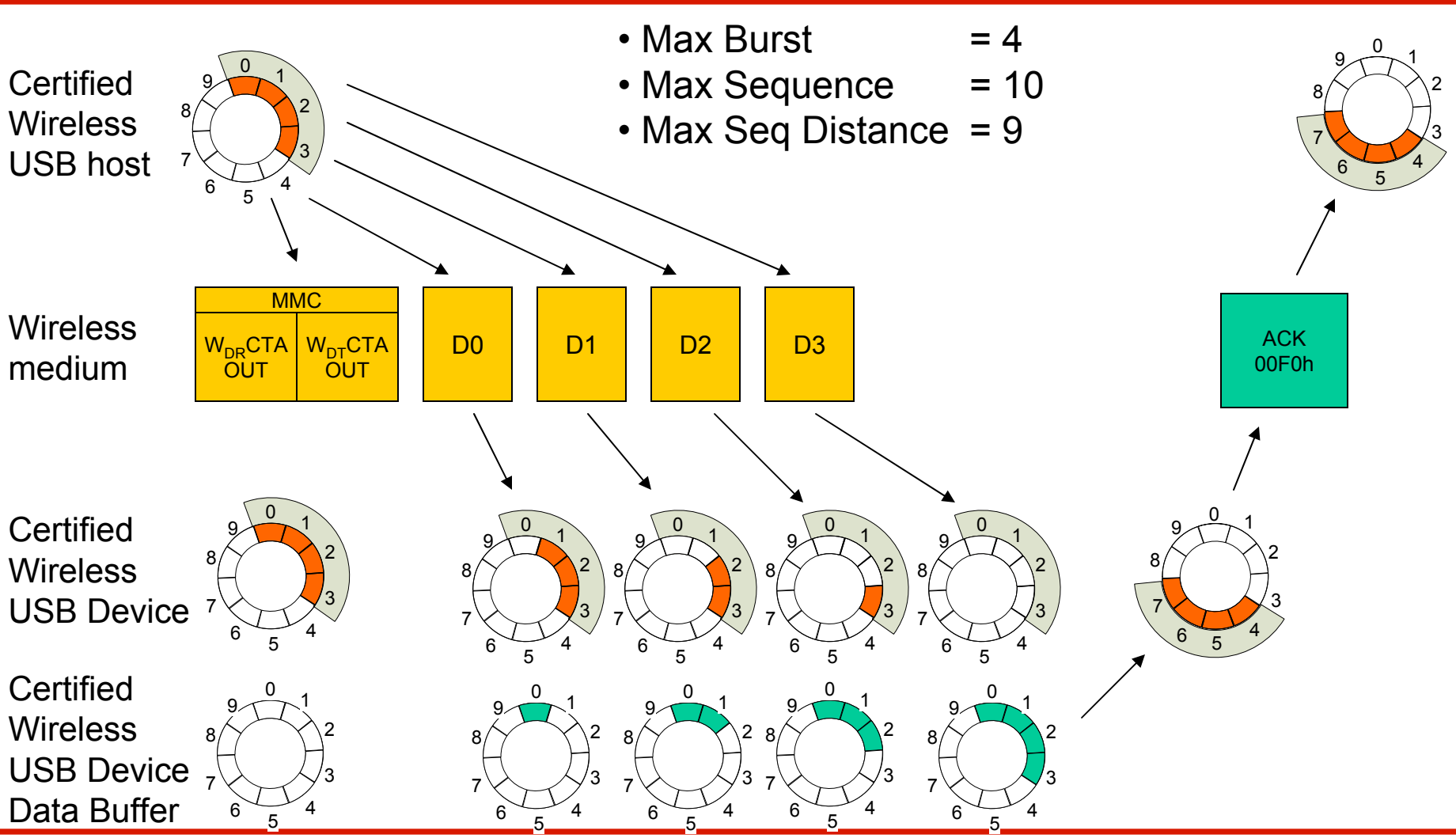
Receiver Rules

- Must initialize to sequence number 0 first
- Rx must always acknowledge data packets it successfully receives
  - Rx may choose to avoid handling out-of-order data by Acking up to the first missing packet
- Rx will use data only in strict sequence order
- Must detect sequence number ordering violations
  - Required to avoid silent data corruption
  - Recovery/reporting method depends receiver

Outline

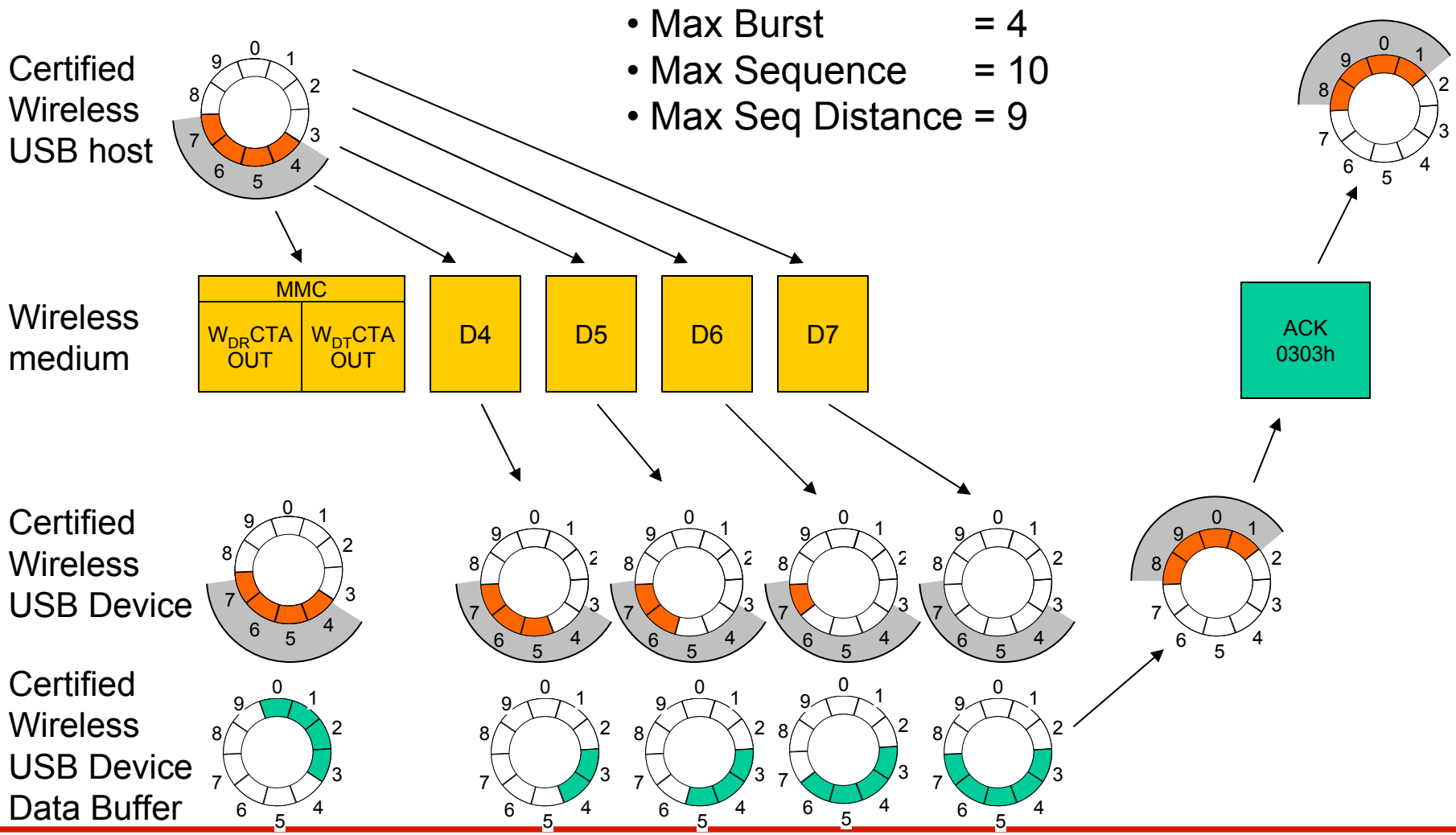
- Basic model
- Examples
  - OUT endpoint
  - IN endpoint
  - Special cases
- Summary

OUT Endpoints
No Smashed Packets



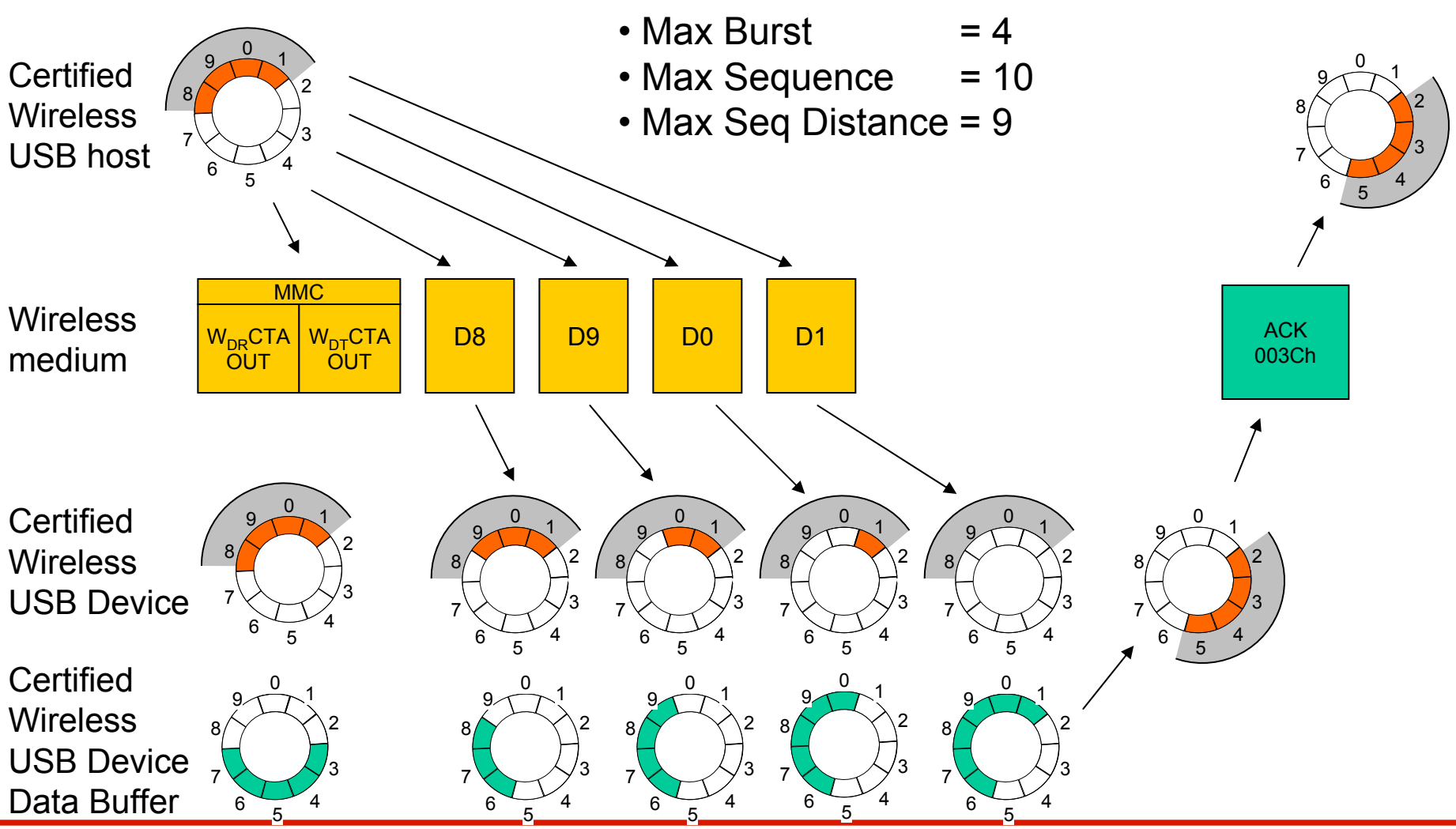
OUT Endpoints

No Smashed Packets (Continued)

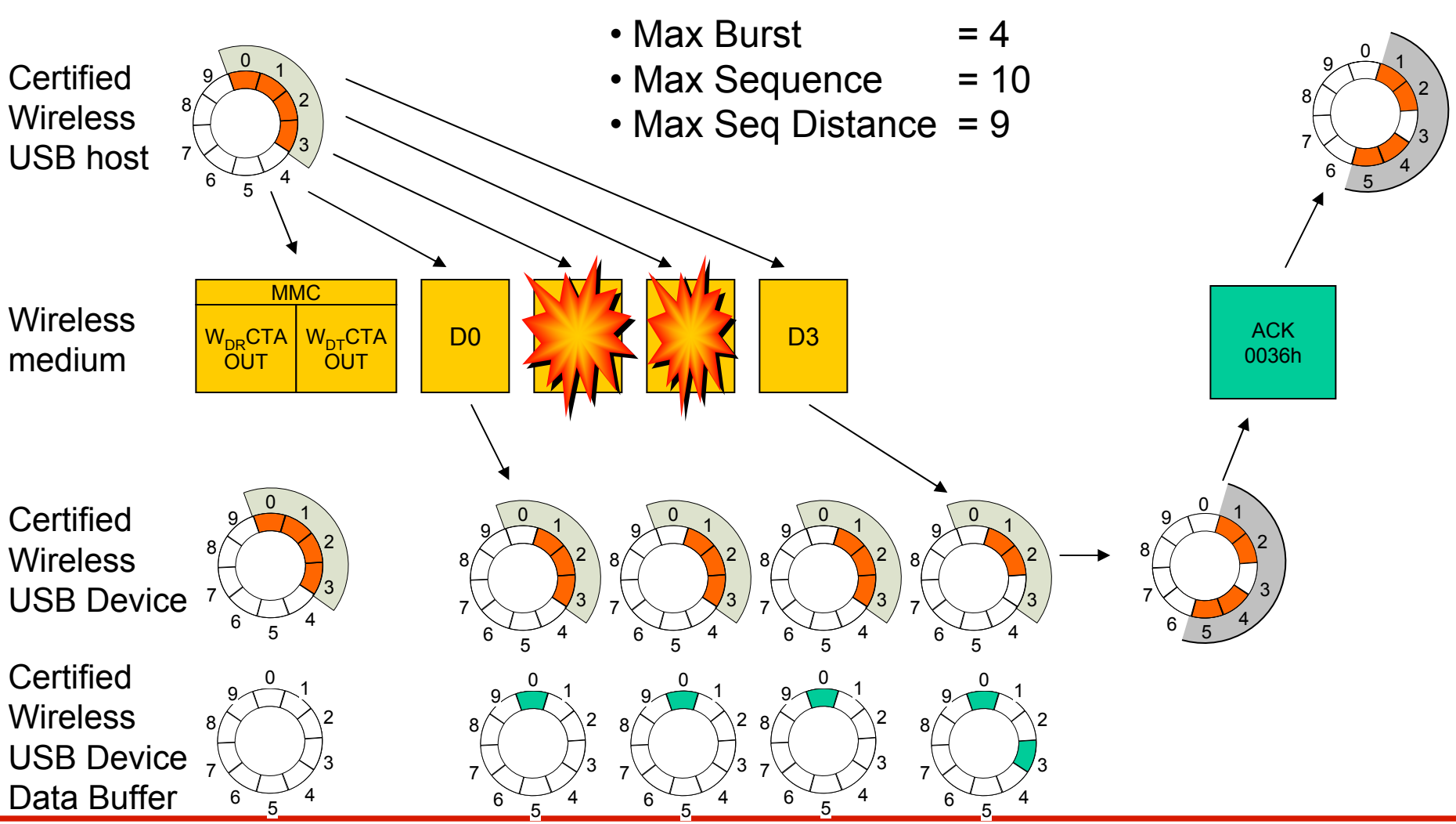


OUT Endpoints

No Smashed Packets (Continued)

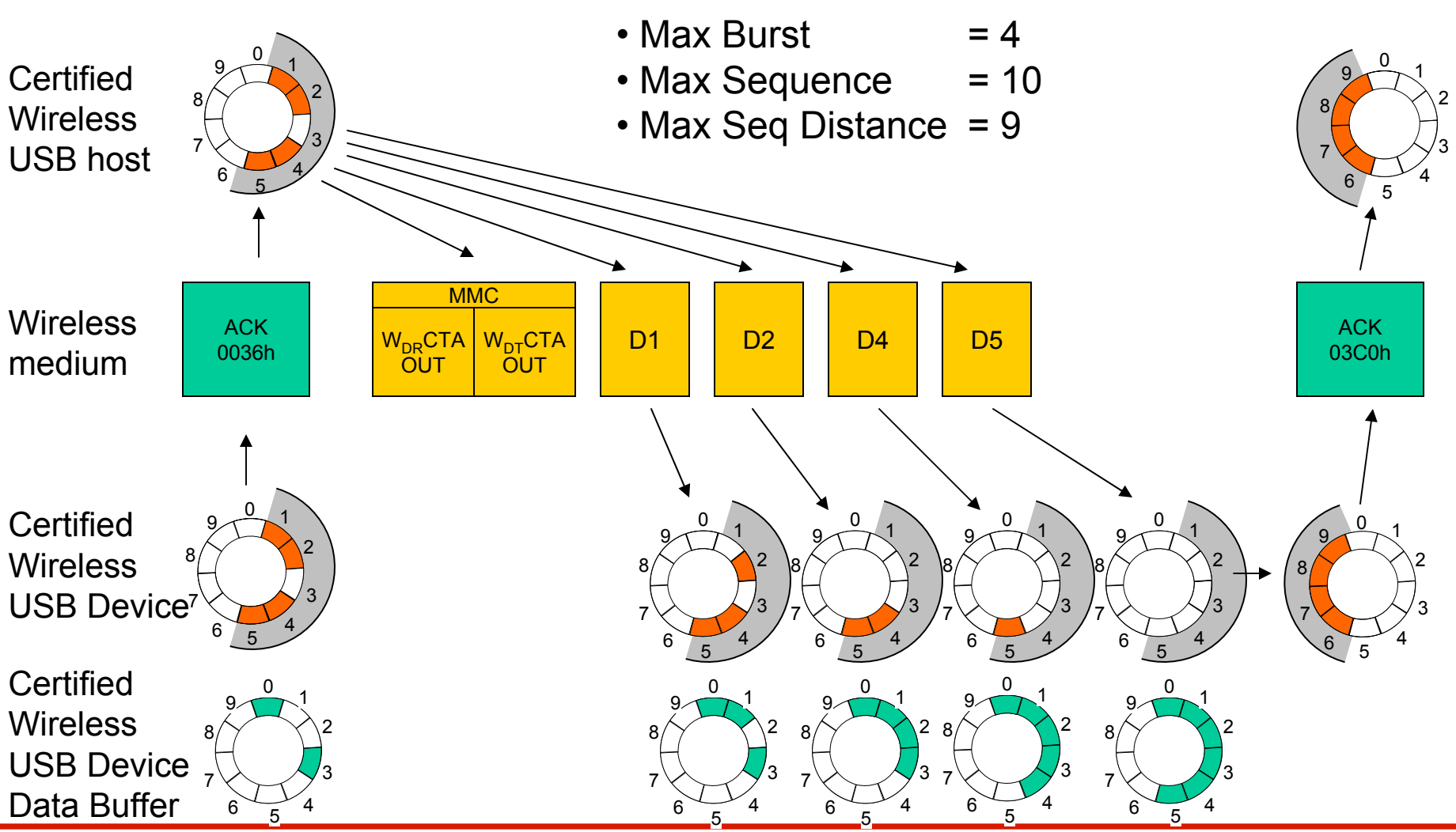


OUT Endpoints
Smashed Data Packets

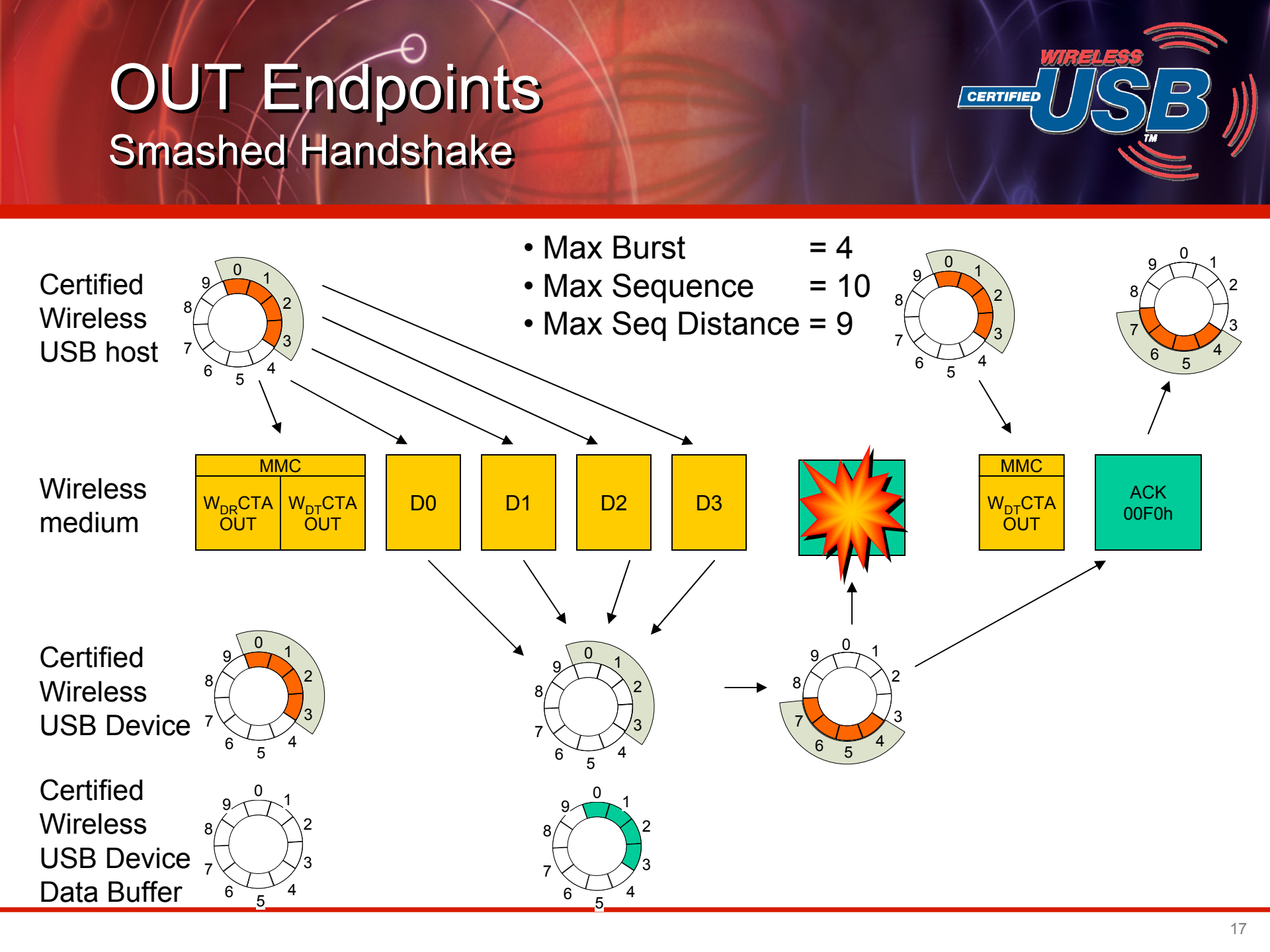


OUT Endpoints

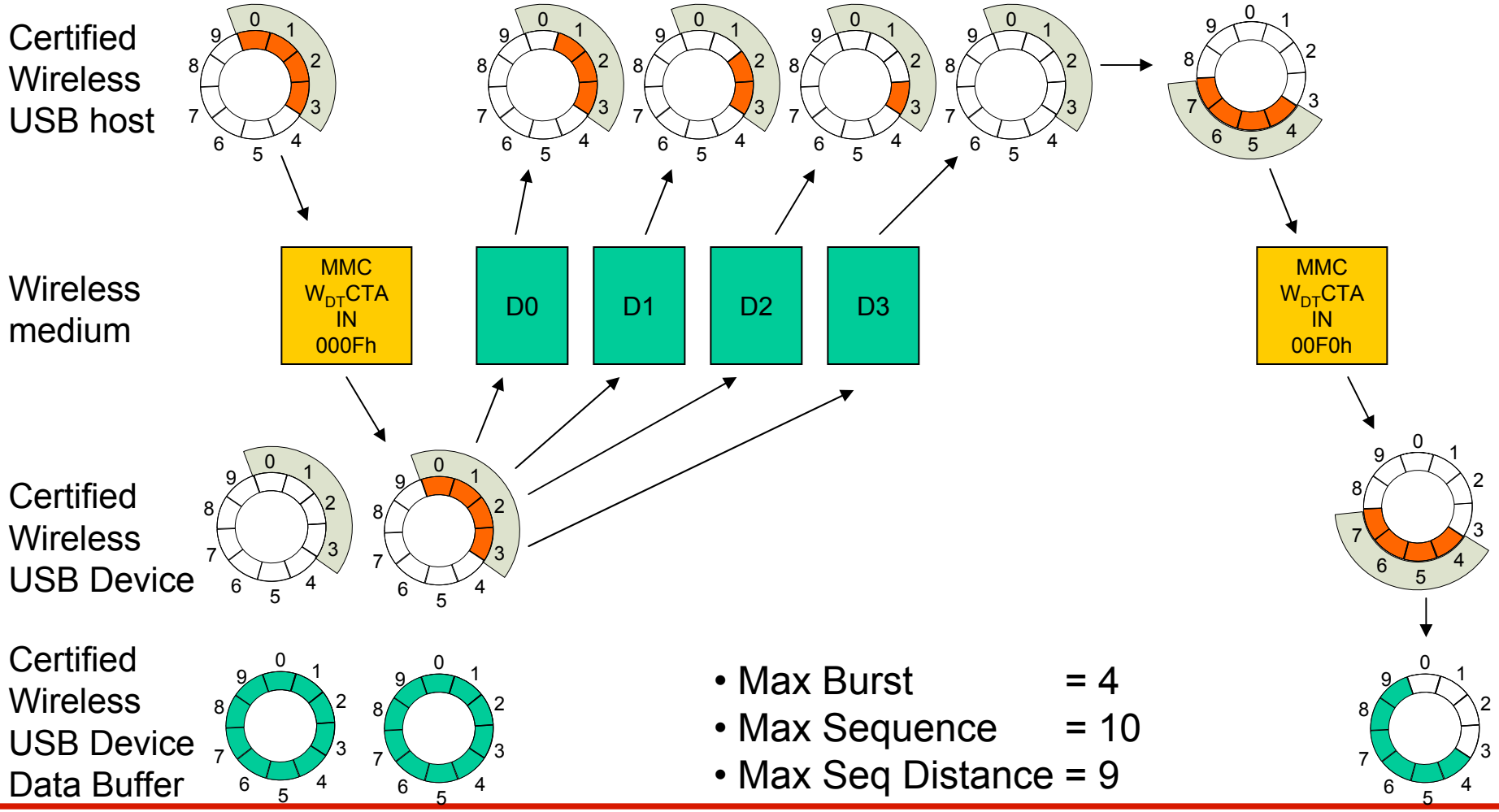
Smashed Data Packets (Continued)



OUT Endpoints
Smashed Handshake

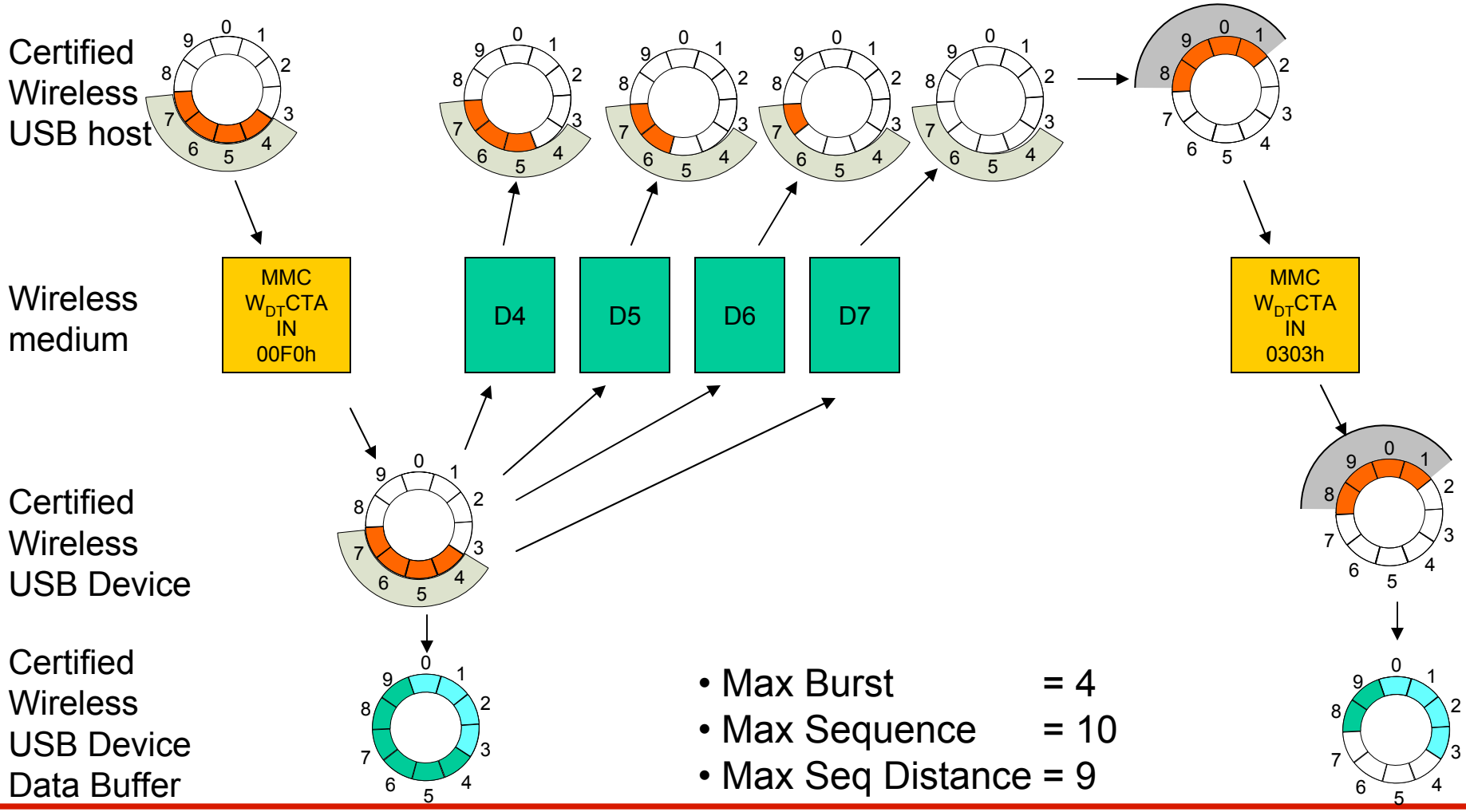


IN Endpoint
No Smashed Packets



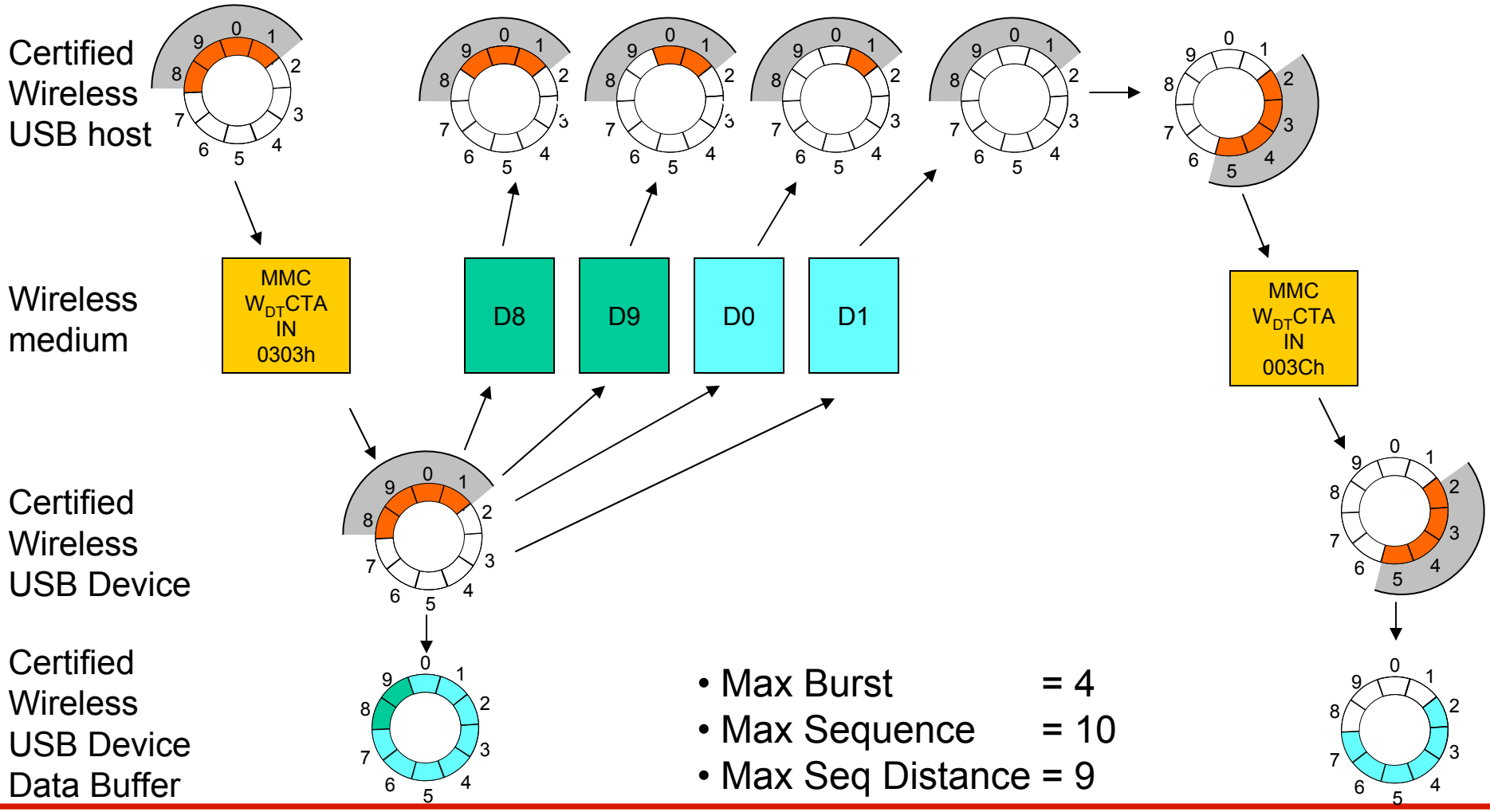
IN Endpoint

No Smashed Packets (Continued)

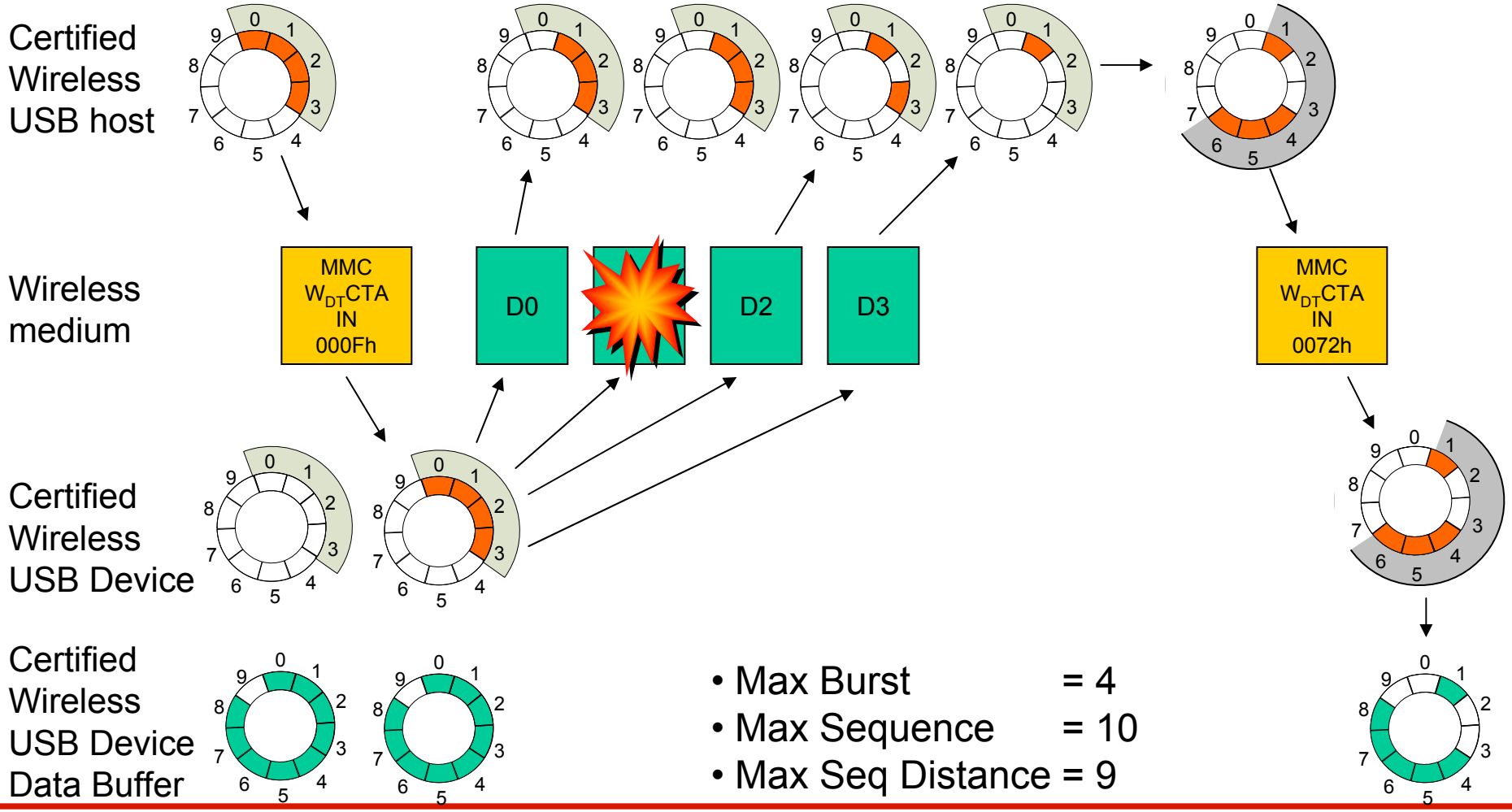


IN Endpoint

No Smashed Packets (Continued)

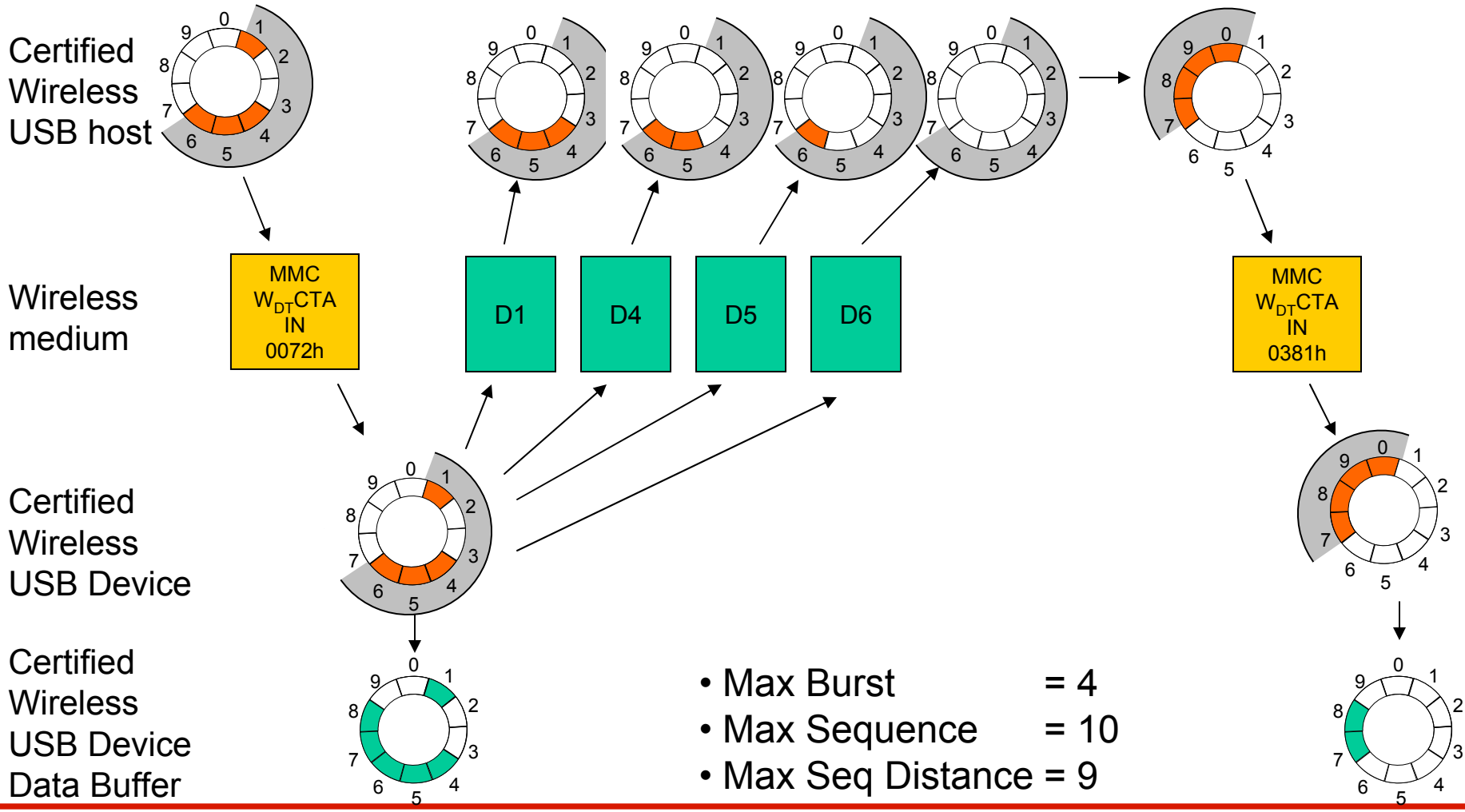


IN Endpoints
Smashed Data Packet

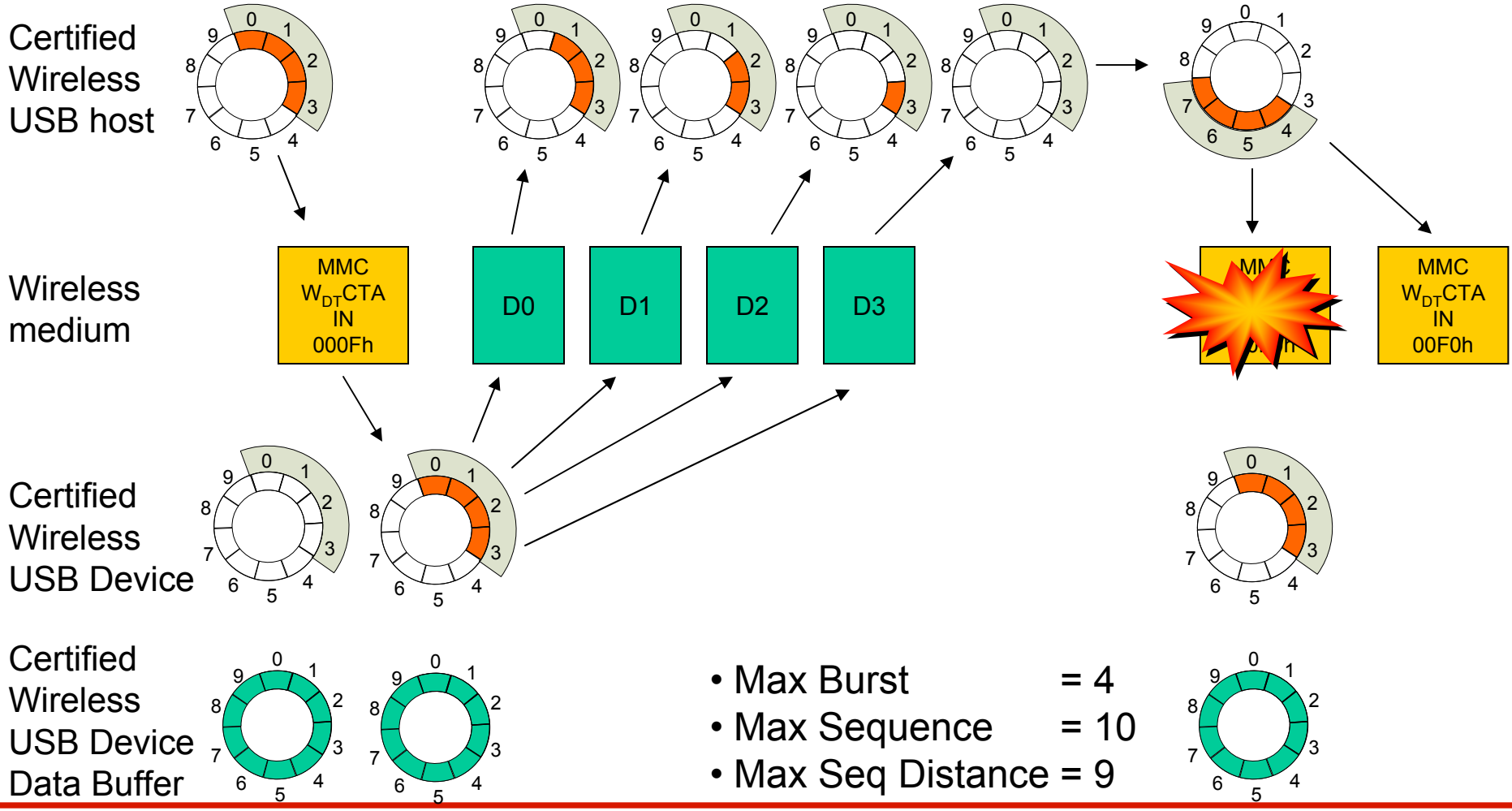


IN Endpoints

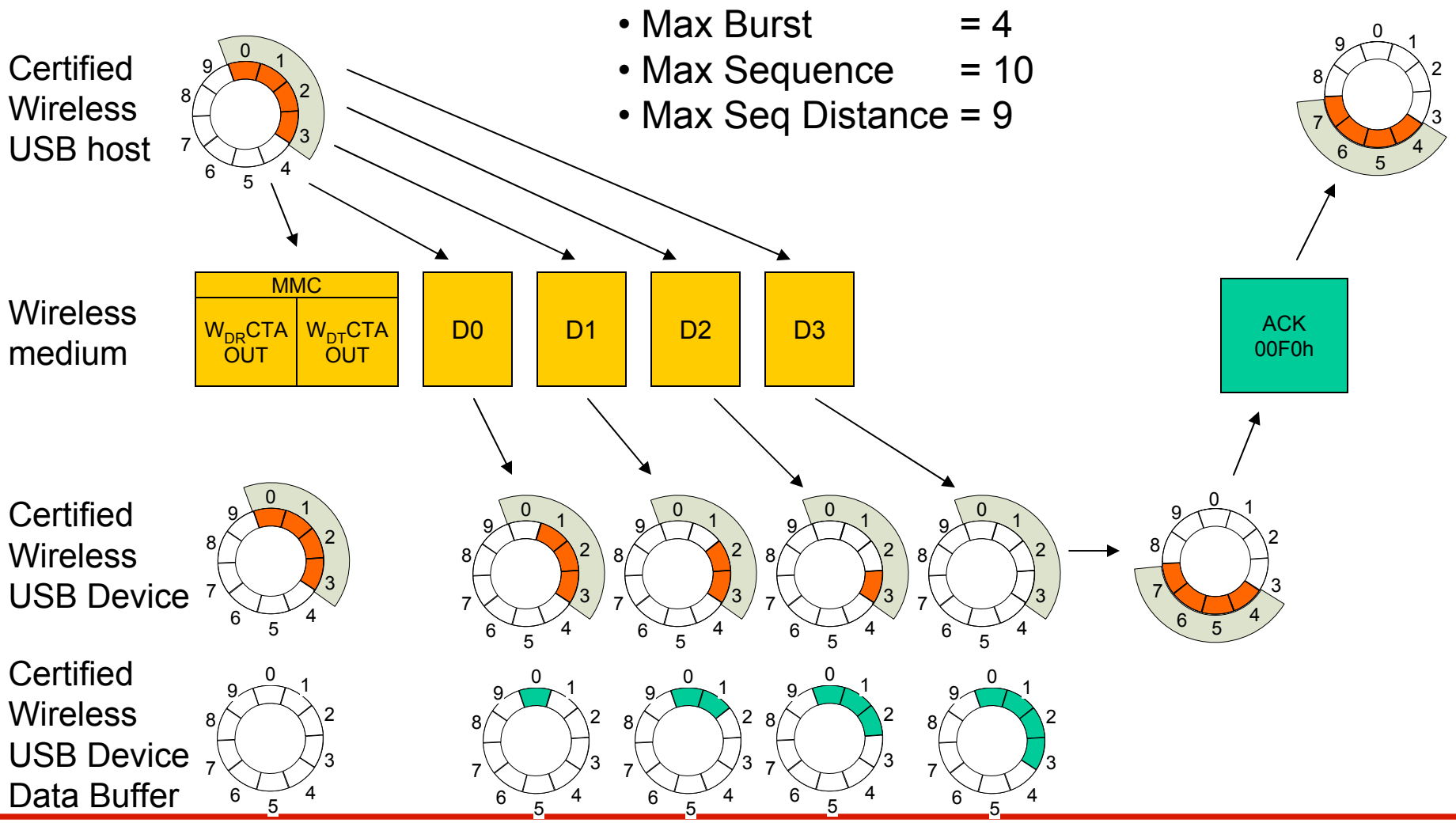
Smashed Data Packet (Continued)



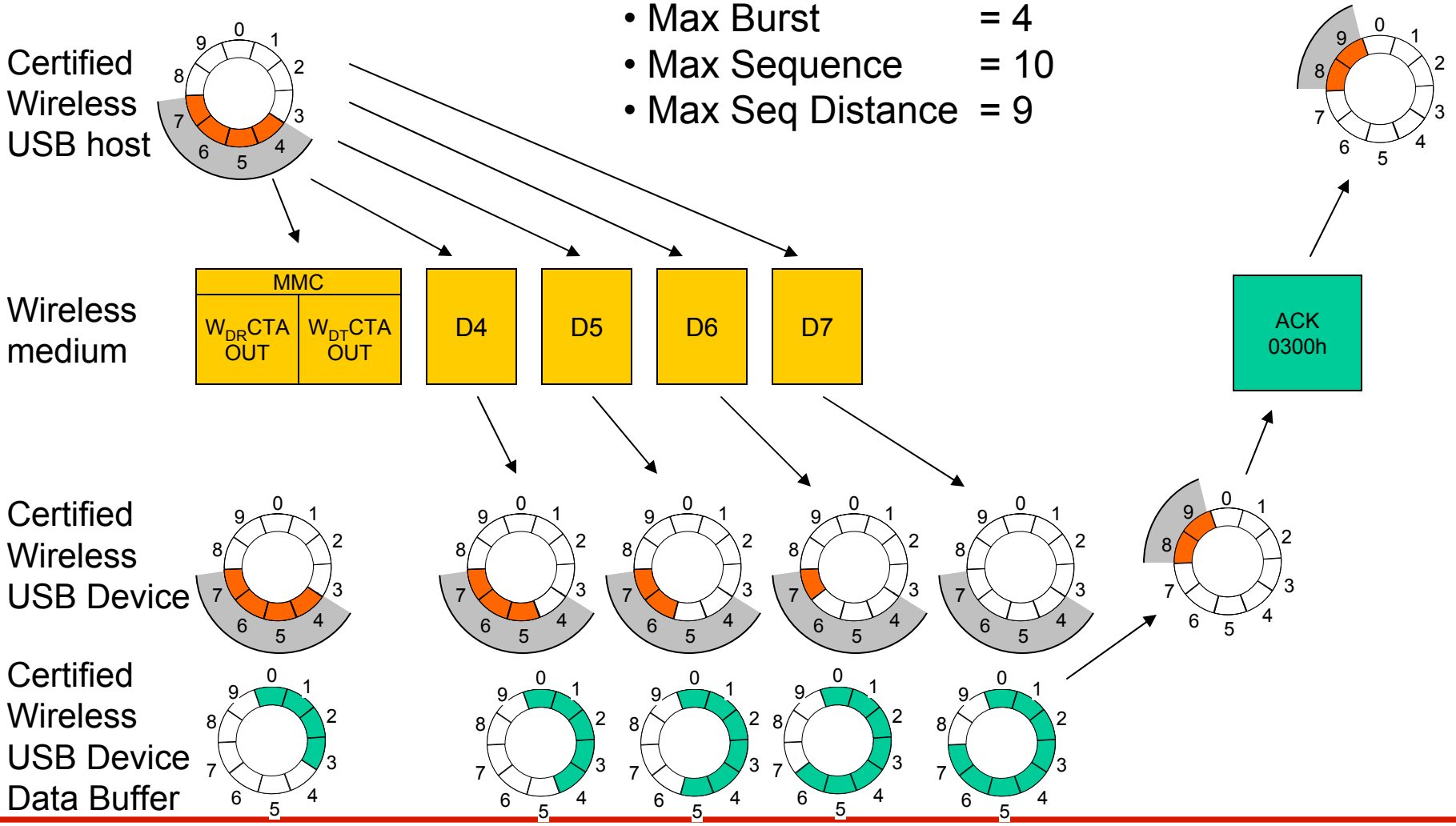
IN Endpoint
Smashed Handshake



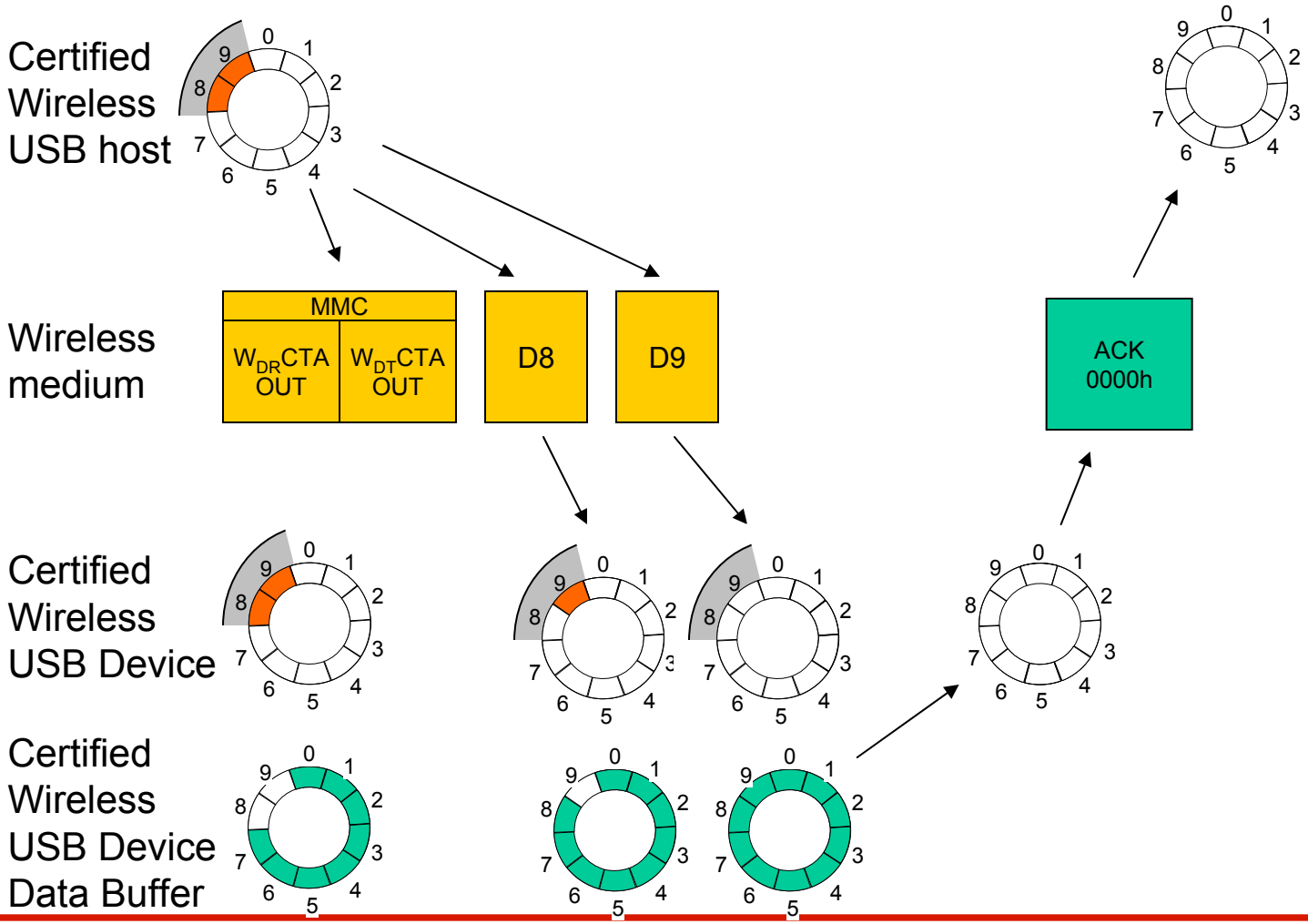
Special Cases
Flow Control



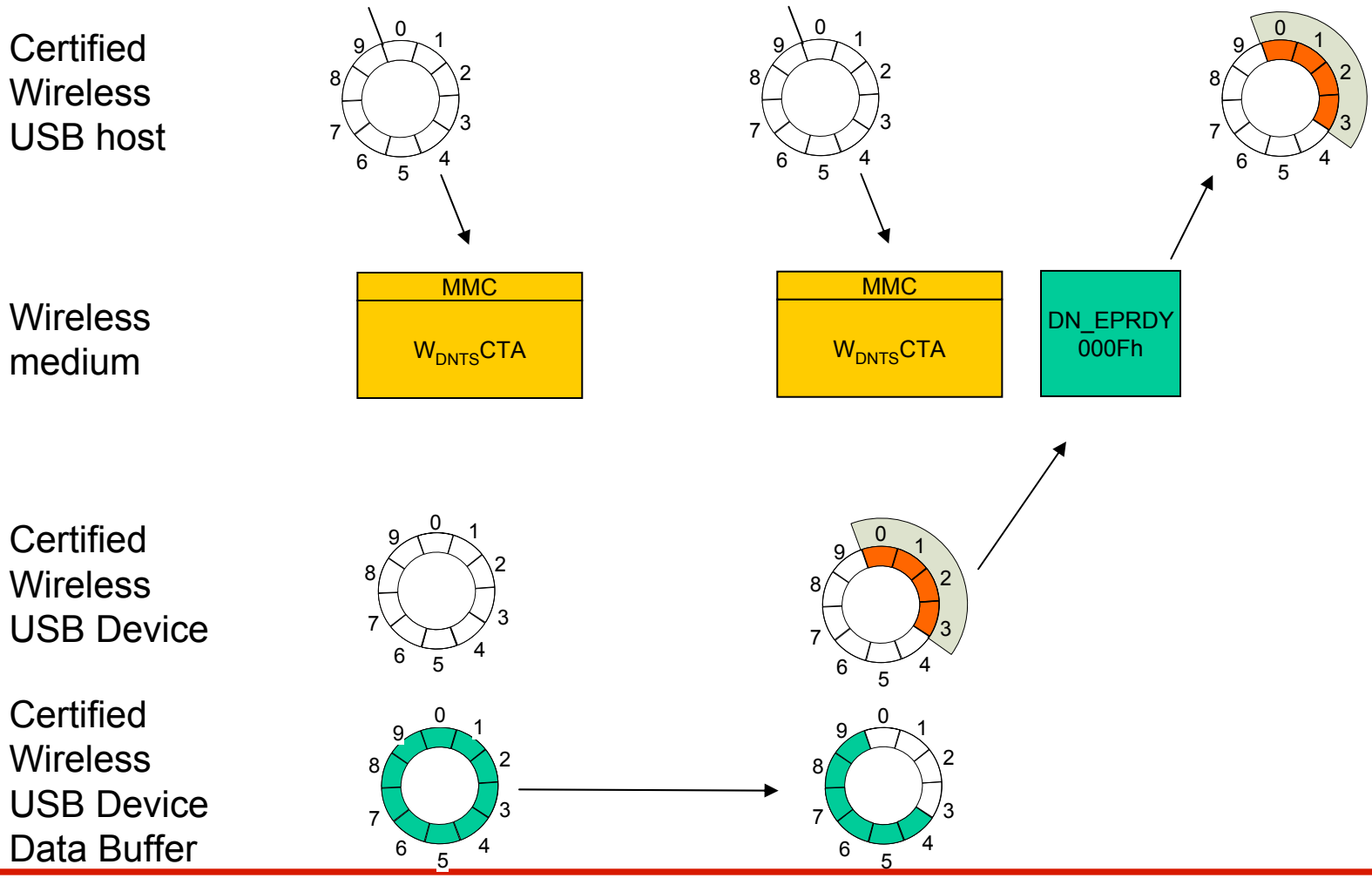
Special Cases
Flow Control (Continued)



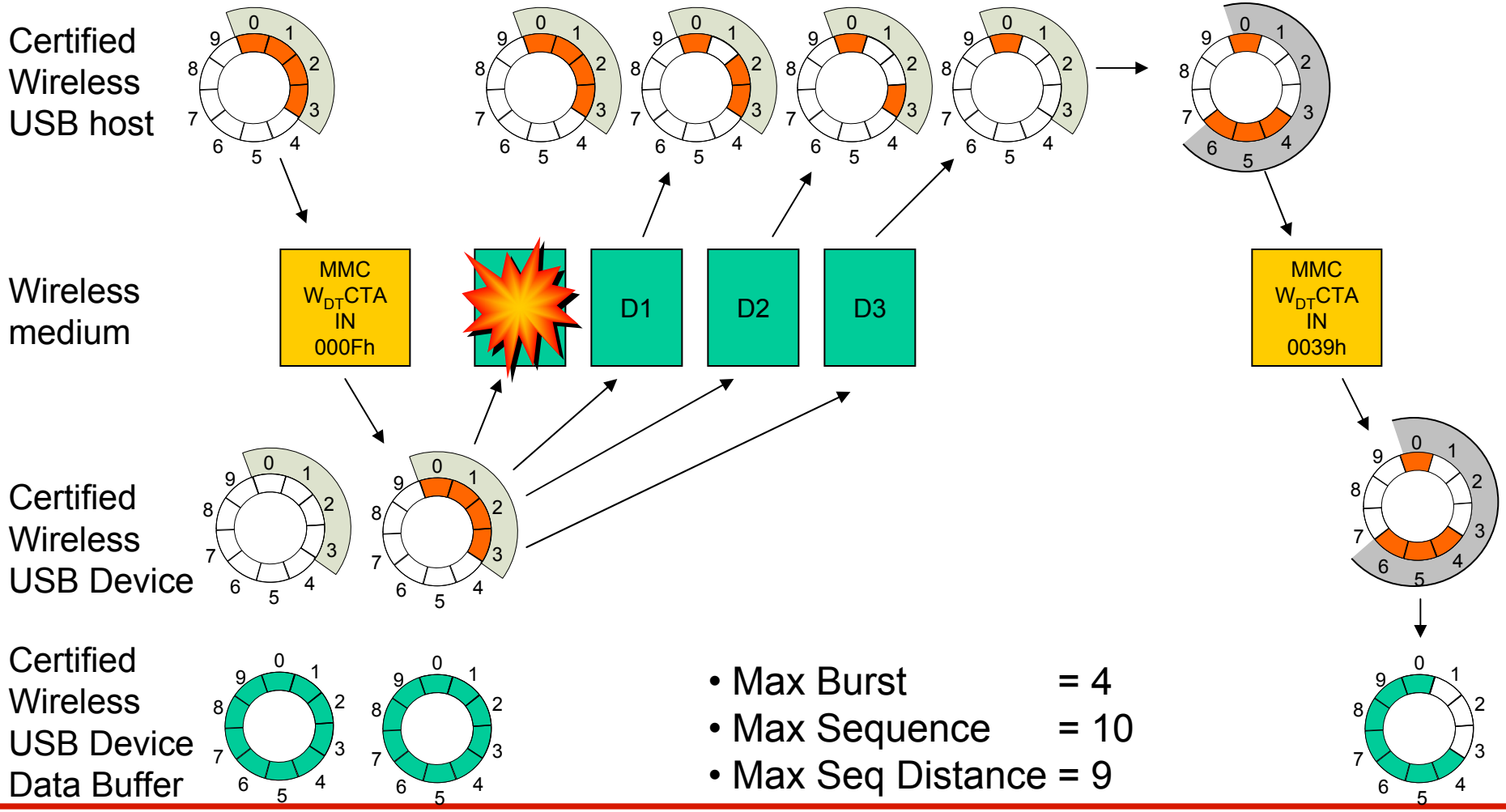
Special Cases
Flow Control (Continued)



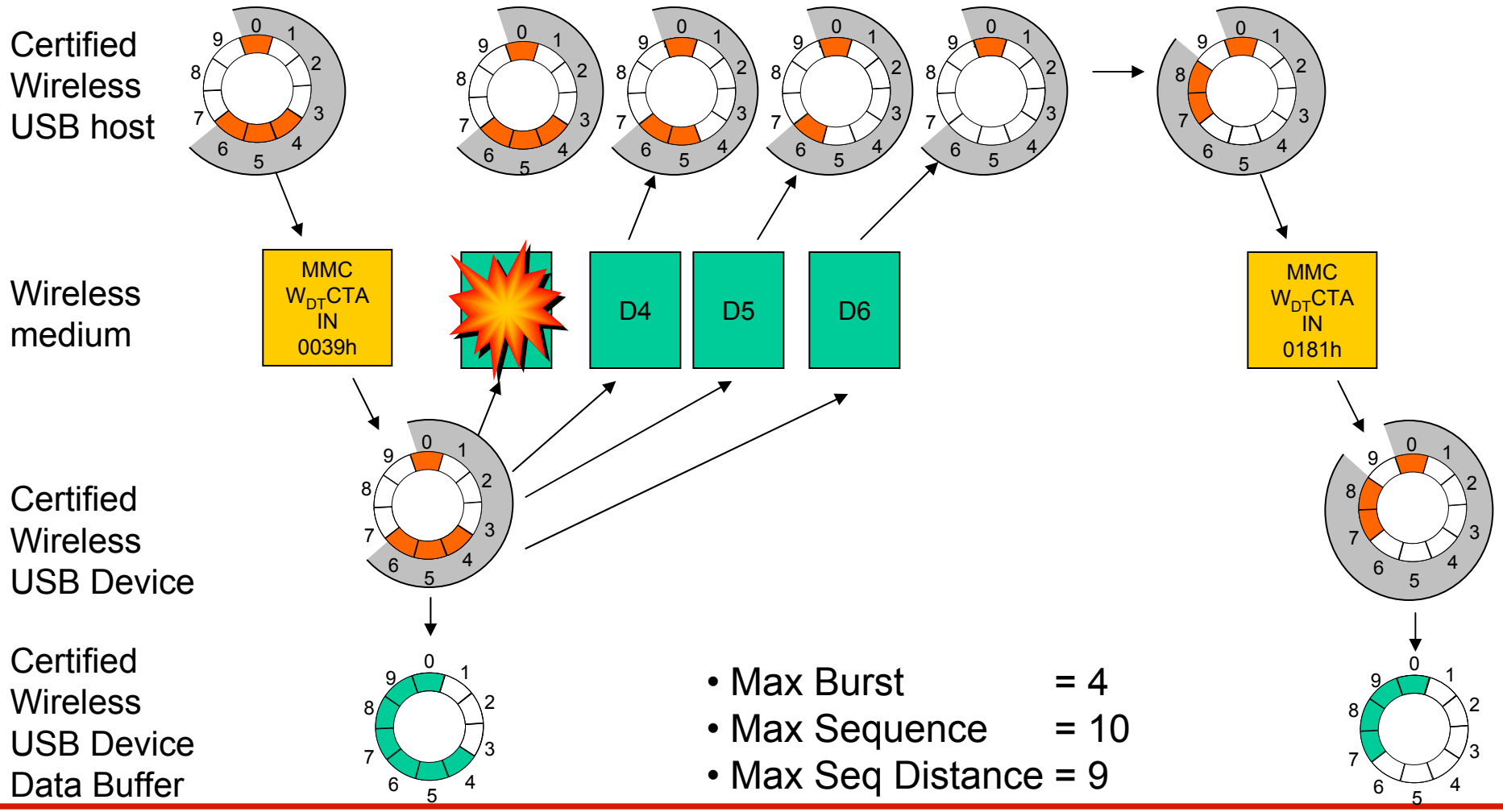
Special Cases
DN_EPReady



Special Cases
Stuck-at Wrap Condition

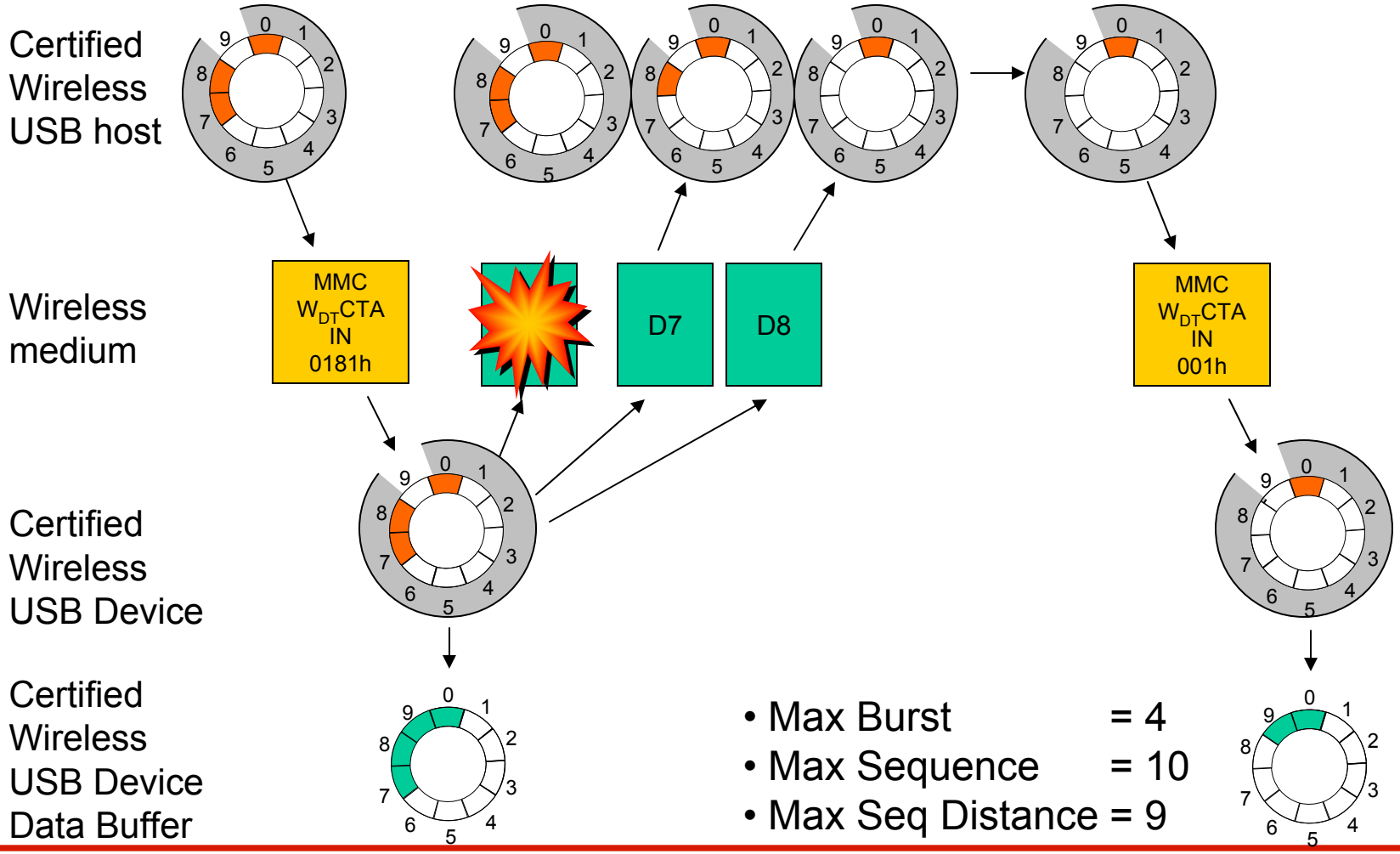


Special Cases
Stuck-at Wrap Condition (Continued)



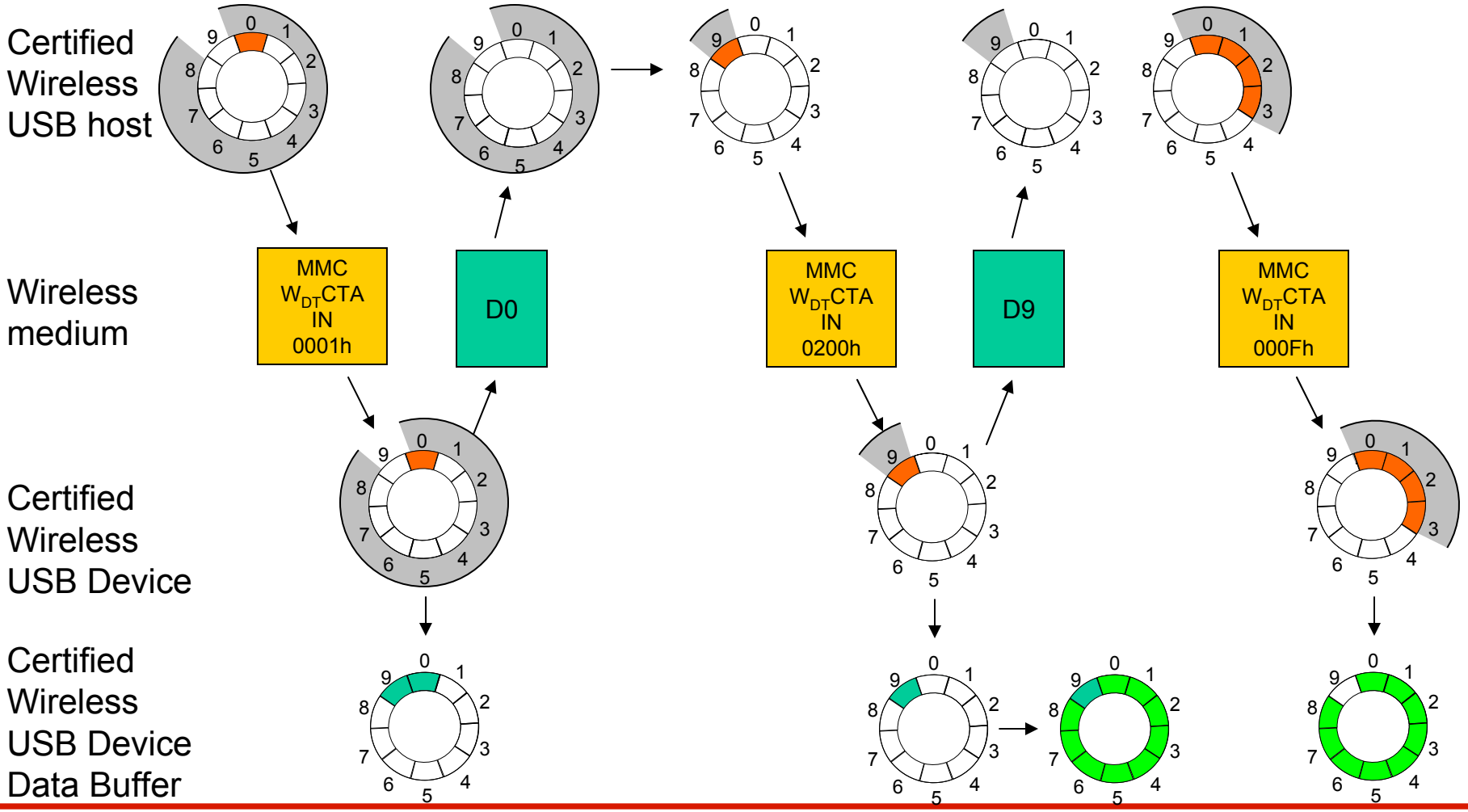
Special Cases

Stuck-at Wrap Condition (Continued)



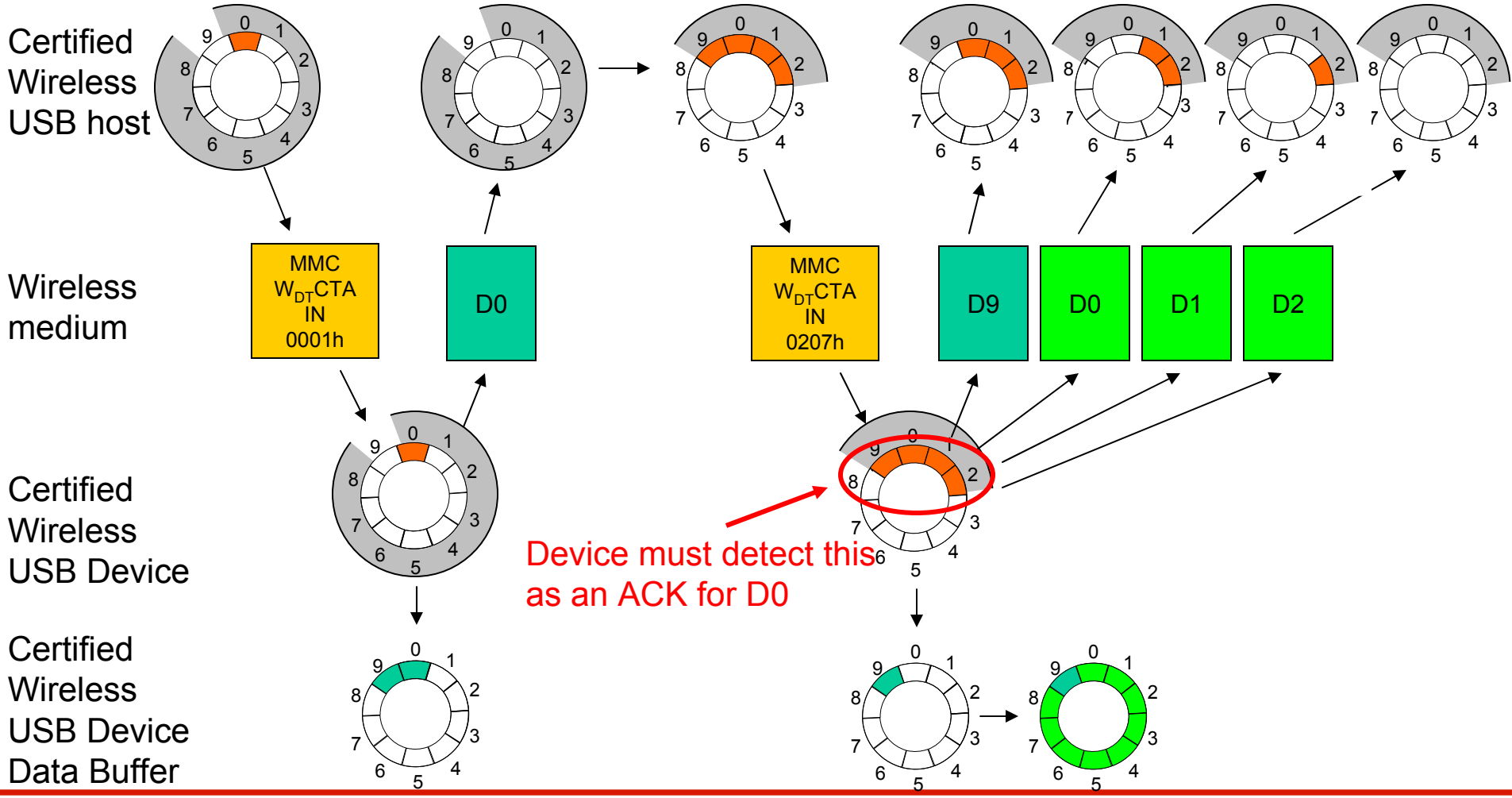
Special Cases

Stuck-at Wrap Condition (Continued)

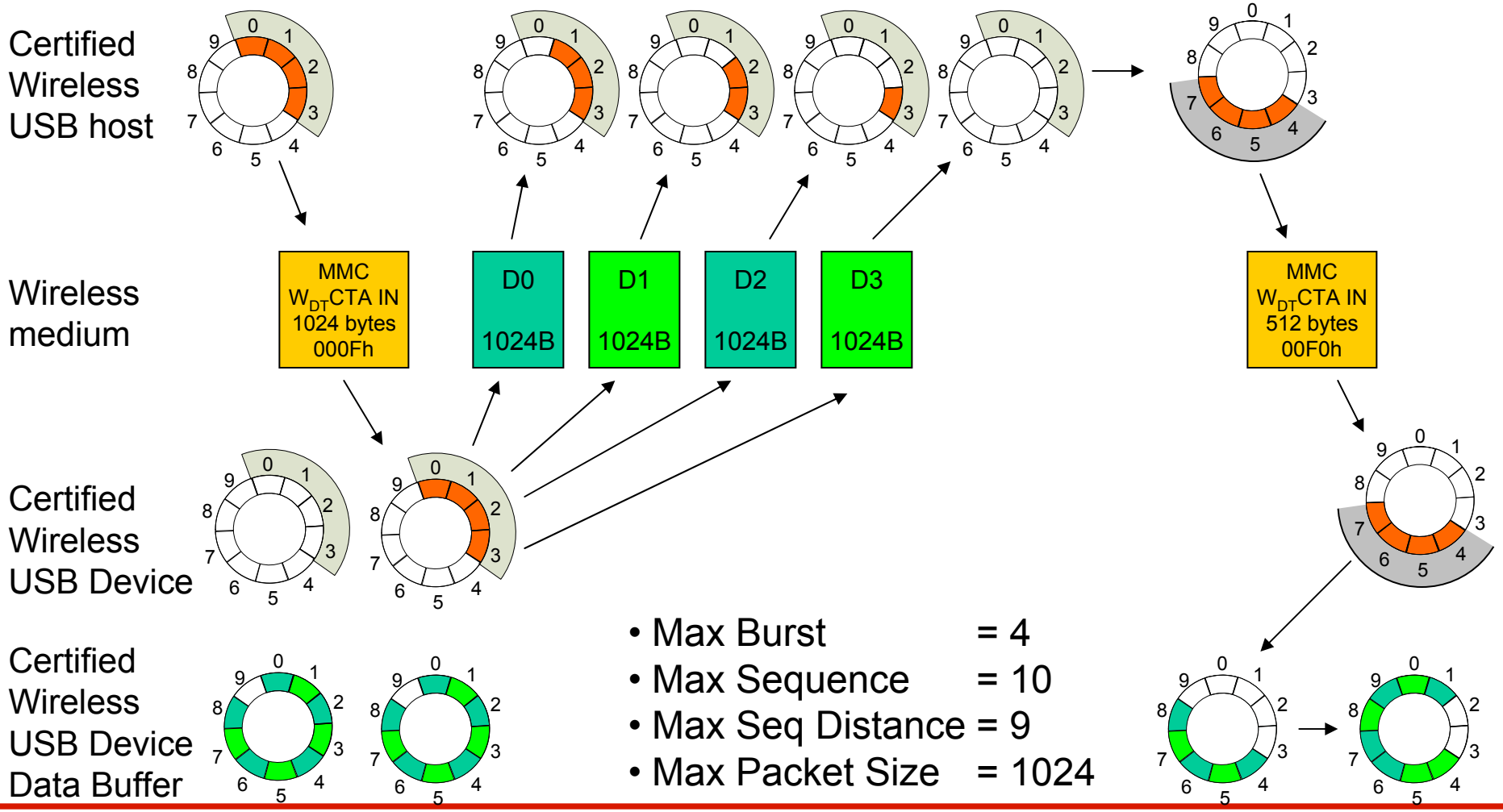


Special Cases

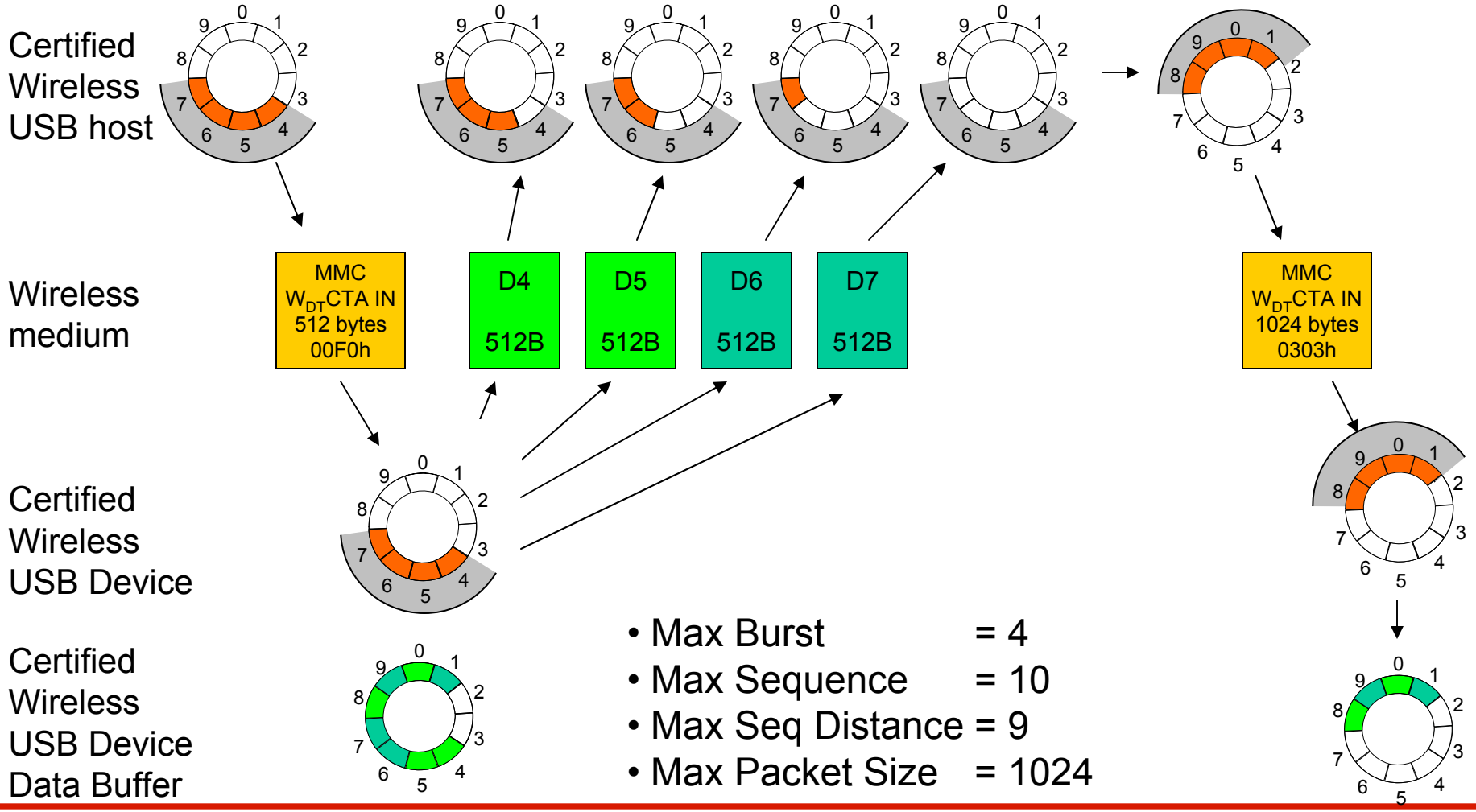
Stuck-at Wrap Condition (Continued)



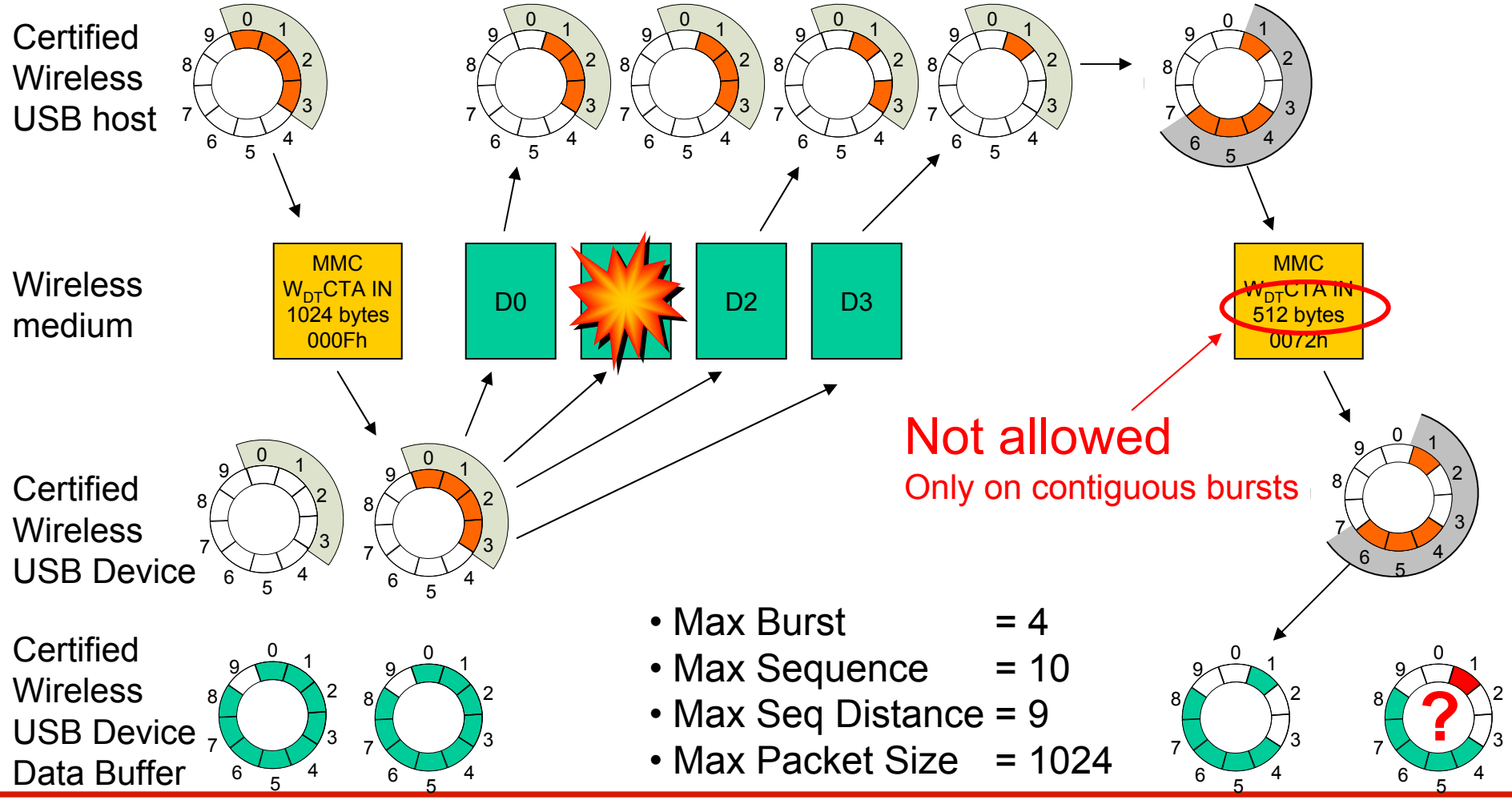
Special Cases
Adjusted Packetsize



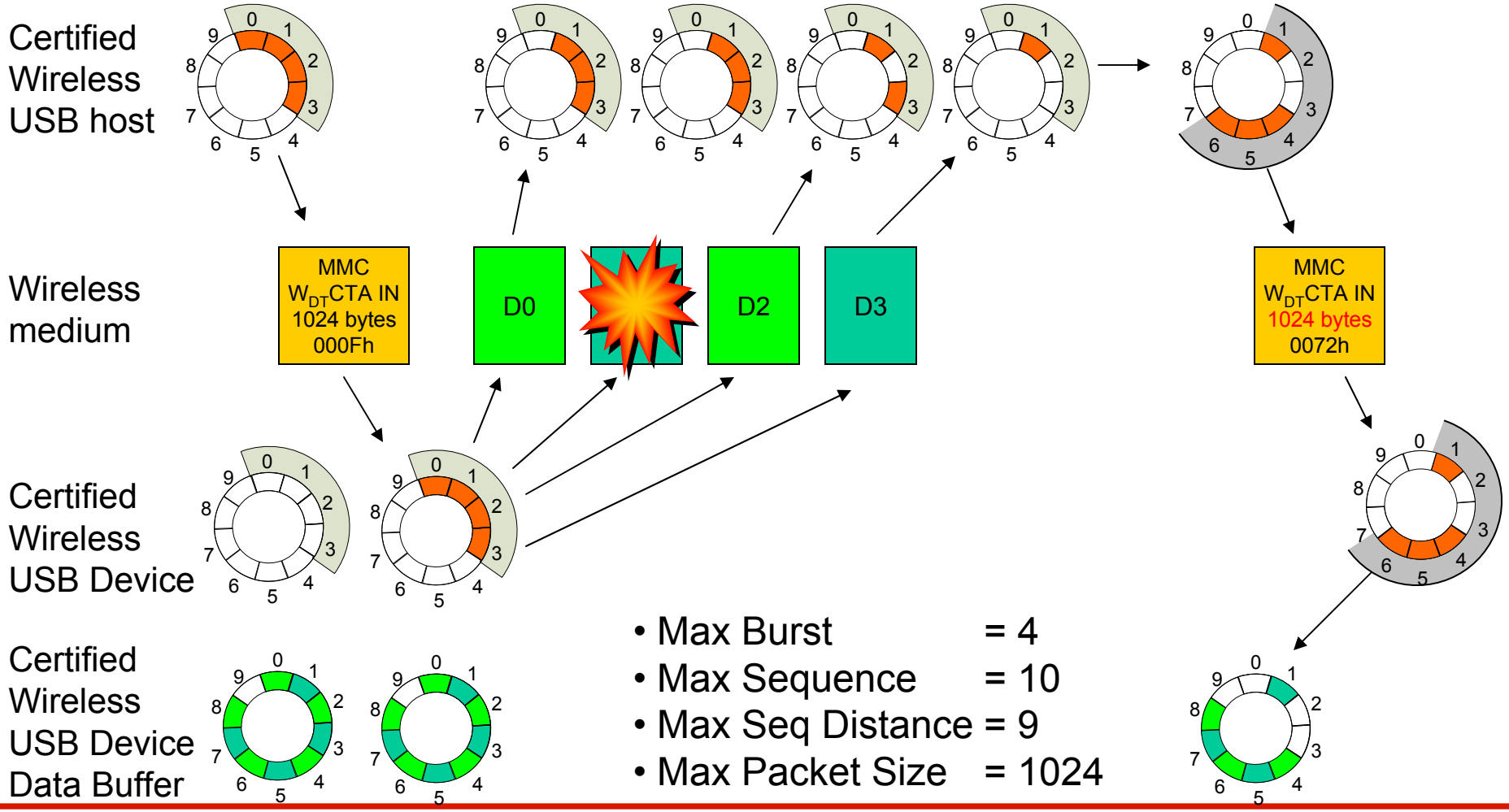
Special Cases
Adjusted Packetsize (Continued)



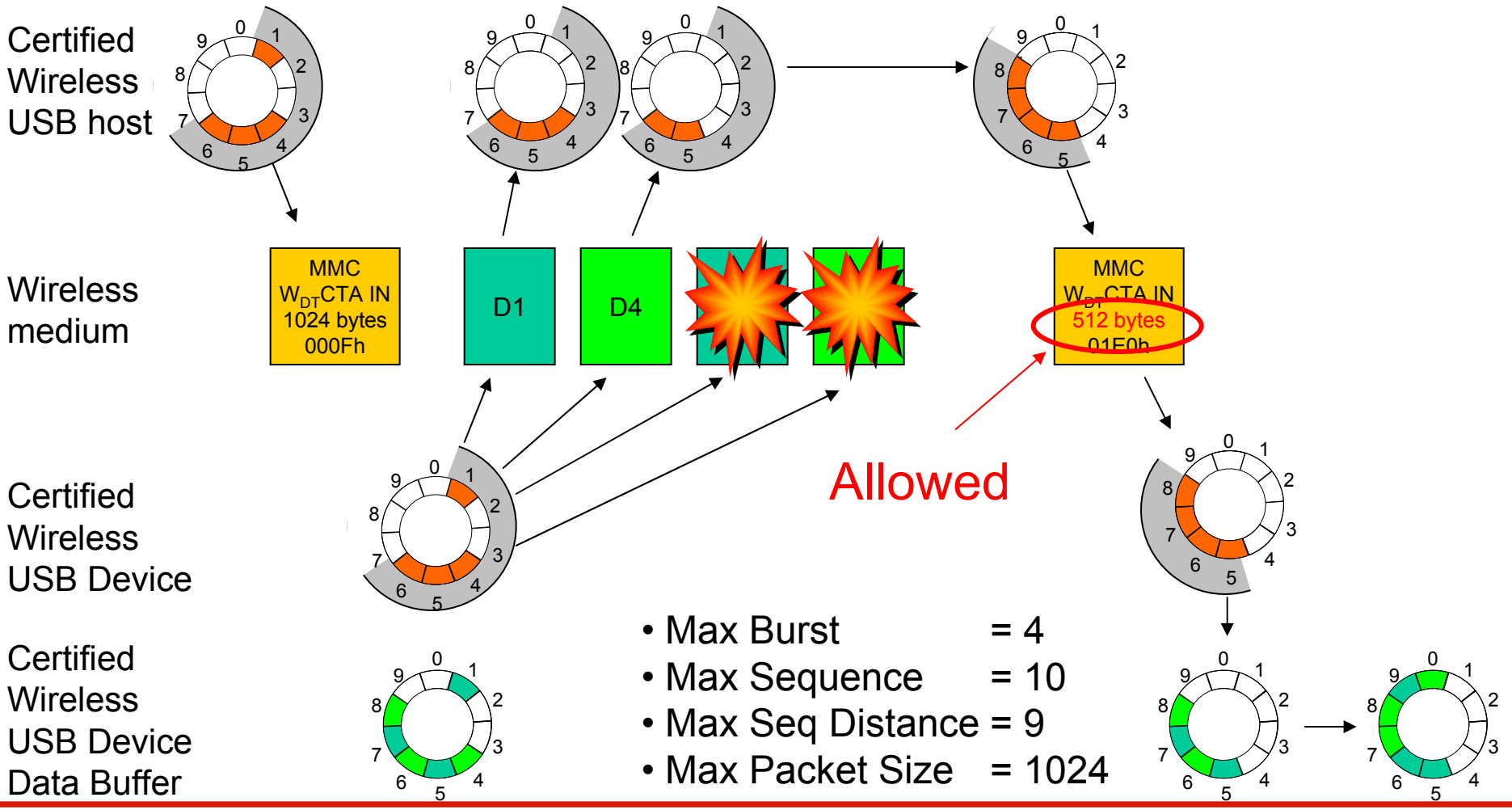
Special Cases
Adjusted Packetsize – Smashed



Special Cases
Adjusted Packetsize – Smashed (Continued)



Special Cases
Adjusted Packetsize – Smashed (Continued)



Outline

- Basic model
- Examples
  - OUT endpoint
  - IN endpoint
  - Special cases
- Summary

Summary

- Handshake contains information on available bufferspace
  - No need to send data if no buffer space available
  - No need to resend data that was already received
- Optimal use of wireless medium
  - Strongly adviced to use bursting as much as possible

Developers Conference 2006

Taipei, Taiwan