Retry Timeout Adjustment during EDCA Periods:
High-Priority Timeout (HPTO) for P-EDCA

Date: July 2025

Authors:

Name	Affiliations	Address	Phone	email
Behnam Dezfouli	Nokia	520 Almanor, Sunnyvale, CA		behnam.dezfouli@nokia.com
Davis Robertson				
Mikhail Liubogoshchev				
Klaus Doppler				
Salvatore Talarico				
Kerstin Johnsson				

Introduction

- EDCA is the primary access method for STAs to reserve a TXOP, especially to send aperiodic, event-driven traffic
- EDCA struggles when multiple STAs with Low-Latency (LL) traffic (AC_VO) compete, or when LL STAs contend with AC_BE STAs

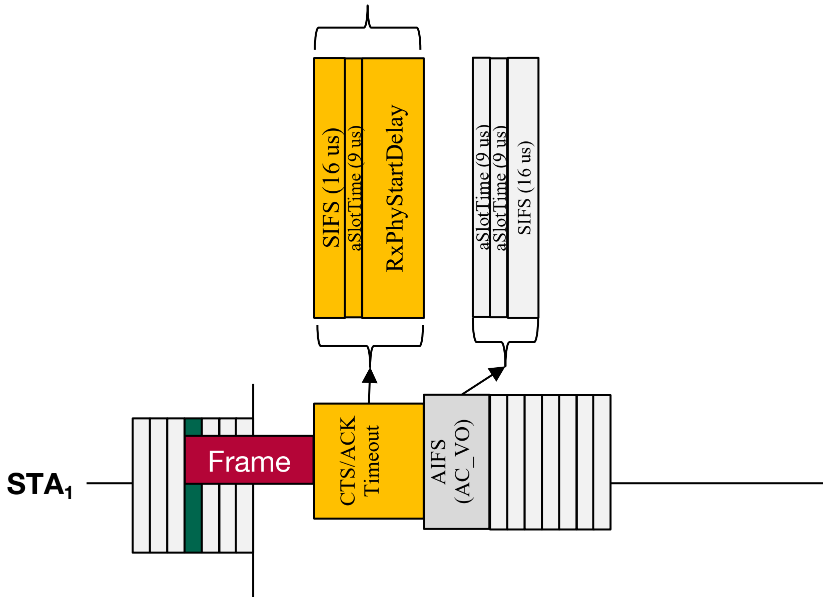
- Lowering the tail-time latency of STAs competing for channel access through EDCA has been addressed in several contributions [11-24/1918][11-24/1144][11-24/0864]
- The P-EDCA mechanism [11-25/1214][11-24/1918][11-24/1144] allows STAs with LL traffic to send Defer Signal (DS) frame after a certain number of retries, and then use RTS/CTS to reserve a TXOP
  - **Shortcoming: Using the CTS/ACK Timeout duration after sending response-soliciting frames (e.g., RTS) leads to wasted channel time before successful channel reservation**

- **In this contribution, we propose a method to reduce the overhead of the P-EDCA procedure by minimizing the time required to detect transmission failure after sending a response-soliciting frame (RTS frame)**
- **A P-EDCA STA is allowed to use a High-Priority Timeout (HPTO) duration instead of CTS/ACK Timeout to retransmit DS frames and announce protected contention periods**

CTS/ACK Timeout in 802.11

- When a STA sends a **response-soliciting frame (e.g., RTS)**:
  - It needs to wait for **CTS/ACK Timeout** + AIFS[AC] before competing for channel access again
  - The **ACK Timeout** duration depends on the PHY protocol

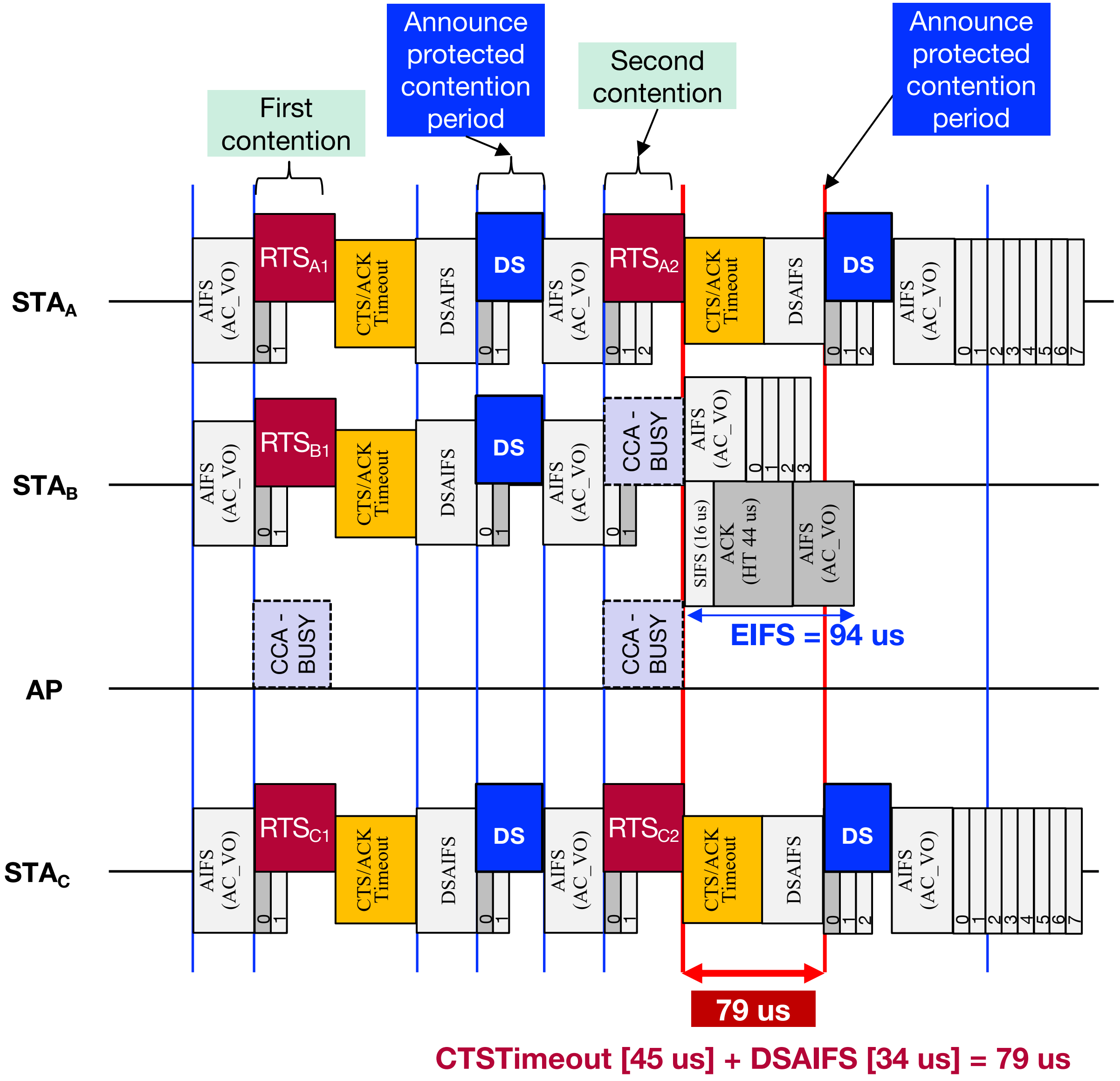
For non-HT OFDM frames:
16 (SIFS) + 9 (aSlotTime) + 20 (RxPhyStartDelay) = **45 us**



	RxPhyStartDelay (us)
OFDM	**20**
HT	28 (HT-mixed), 24 (HT-greenfield)
VHT	$36 + 4 \times N_{VHT\text{-}LTF} + 4$
EHT	$32 + 4 \times N_{EHT\text{-}SIG}$ for EHT MU PPDU 32 µs for EHT TB PPDUs

❑ Sample Scenario

- **First contention round:** STA_A, STA_B, and STA_C compete; all three STAs use CTS/ACK Timeout after sending their frames
- STA_A, STA_B, and STA_C send **DS frames** to announce **protected contention period**
- **Second contention round (protected):** STA_A and STA_B send RTS frames, **and they both fail**
- They are allowed to send DS frames again and compete
- Both STAs wait for CTS/ACK Timeout before sending their next DS frames
  - **The interval between the end of RTS frames and start of DS is 79 us**



A Shorter Retry Timeout: **High-Priority Timeout (HPTO)**

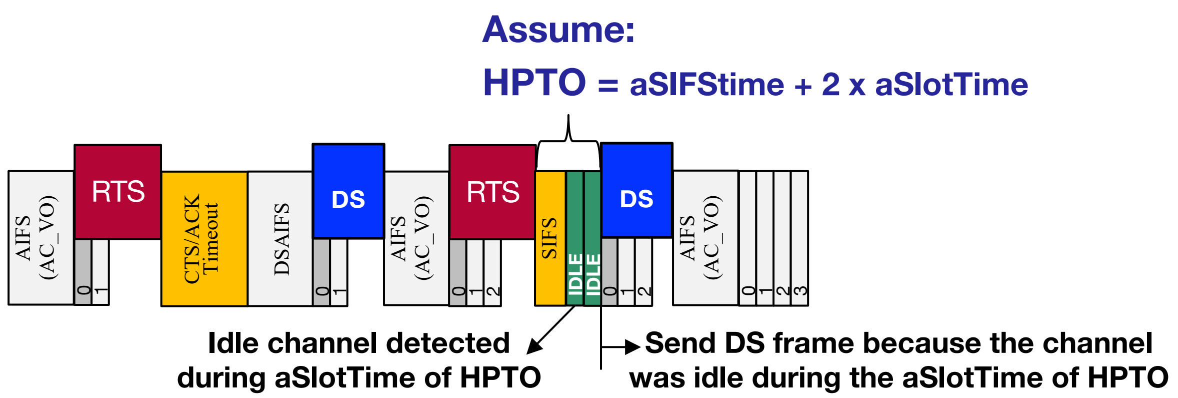
- In this contribution, we allow STAs involved in P-EDCA to **use a High-Priority Timeout (HPTO) duration instead of CTS/ACK Timeout** to reduce the channel time used by P-EDCA
- **HPTO starts when the STA receives the PHY-TXEND.confirm for the transmitted frame (RTS)**

- **$HPTO_{min}$ = aSIFStime + aSlotTime**
  - $HPTO_{min}$ provides
    - Enough time for the **receiver** of the response-soliciting frame to receive, process, and start sending a reply to the sender, and
    - Enough time for the **sender** of the response-soliciting frame to perform carrier sensing (CCA) and switch to TX mode if the channel is sensed as idle
- **When a STA sends a response-soliciting frame (e.g., RTS), it can detect transmission failure if the channel is sensed as idle (i.e., CCA idle) during the aSlotTime of $HPTO_{min}$**

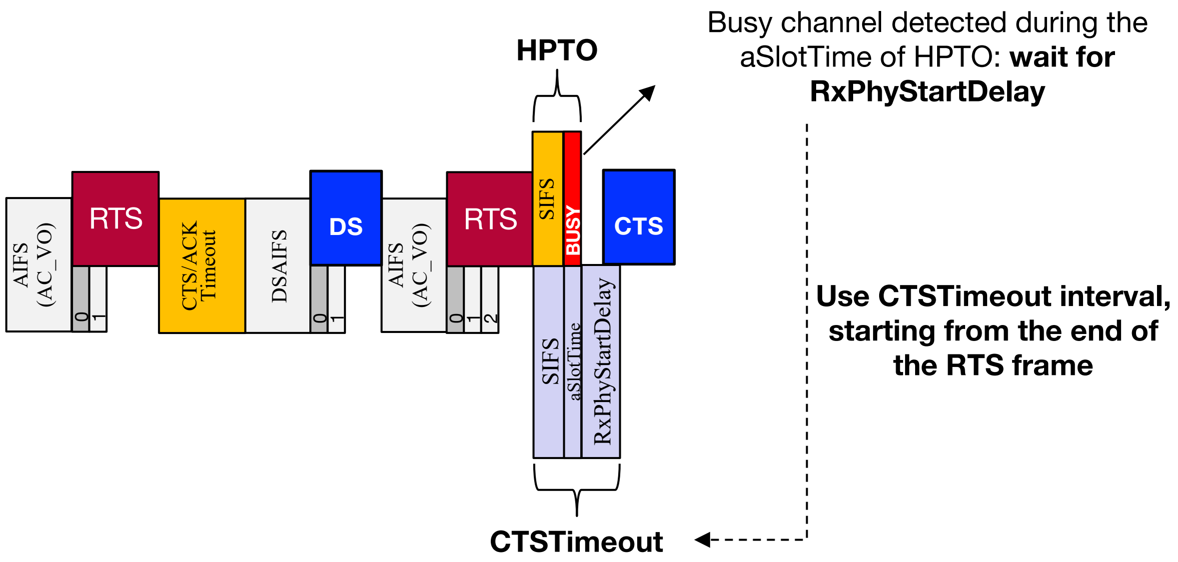
- To align with AIFS values and listen to the channel longer than an aSlotTime, **a longer HPTO may be used; e.g., HPTO = aSIFStime + 2 x aSlotTime**
- **More generically, to align with P-EDCA:**
  - **HPTO = DSAIFS[AC_VO] = aSIFSTime + (AIFSN + DSr) × aSlotTime**

    = aSIFSTime + (2 + DSr) × aSlotTime

Example 1:
Channel is sensed as idle during HPTO



Example 2:
Channel is sensed as busy during HPTO



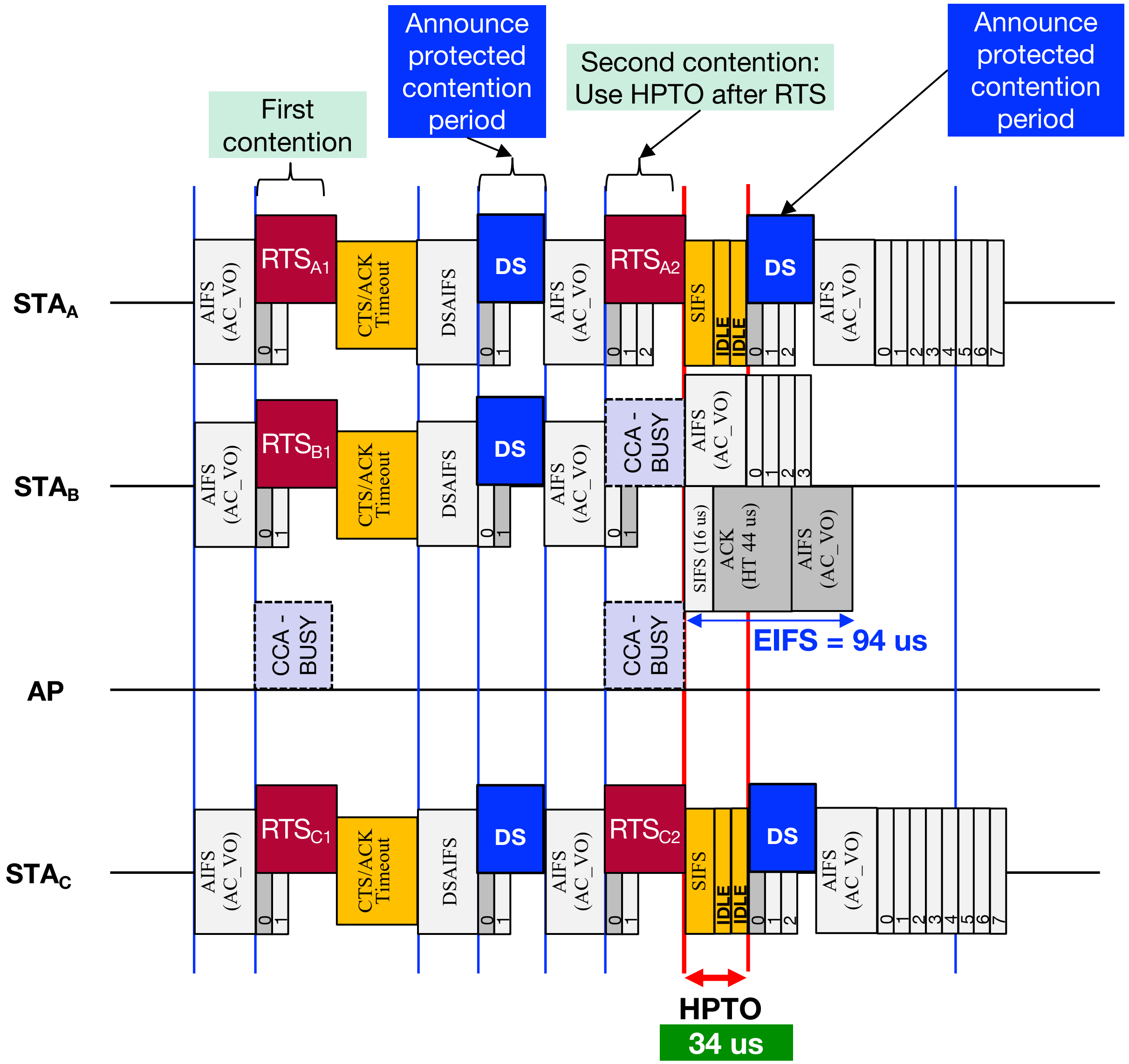
❑ Using HPTO for DS retransmissions

- **We use**
  HPTO = aSIFStime + 2 x aSlotTime

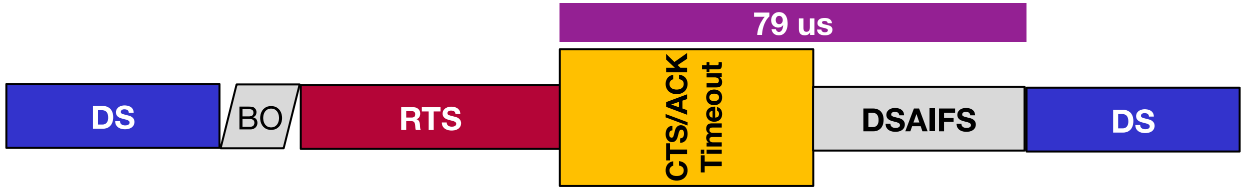
- The earliest time a P-EDCA eligible STA can resend a DS frame is **HPTO after the end of RTS frame**

- **With HPTO: a 34 us interval between the end of an RTS and the start of a DS (this slide)**

- In slide 4, we showed that without HPTO, there is a 79 us interval between the end of an RTS and the start of a DS (when using CTS/ACK Timeout)



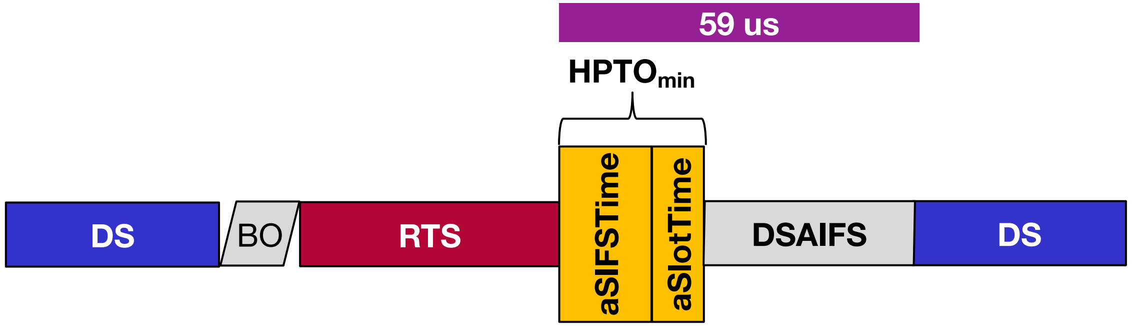
(a) Using CTSTimeout for failure detection, and in the case of failure, wait for DSAIFS before sending DS frame



(b) Using $HPTO_{min}$ for failure detection, and in case of failure, wait for DSAIFS before sending DS frame

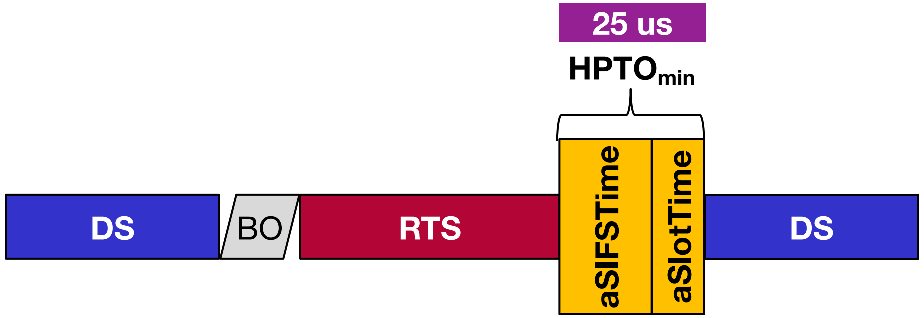
- $HPTO_{min}$ = aSIFStime + aSlotTime

See State Machine S1 (Slide 9)



(c) Using $HPTO_{min}$ for failure detection, and in case of failure, send DS frame immediately

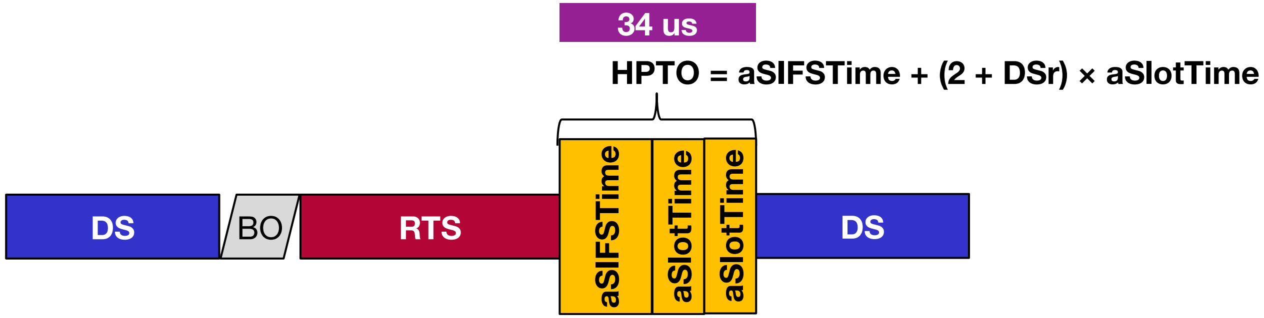
- $HPTO_{min}$ = aSIFStime + aSlotTime



(d) Using HPTO for failure detection, and in case of failure, send DS frame immediately

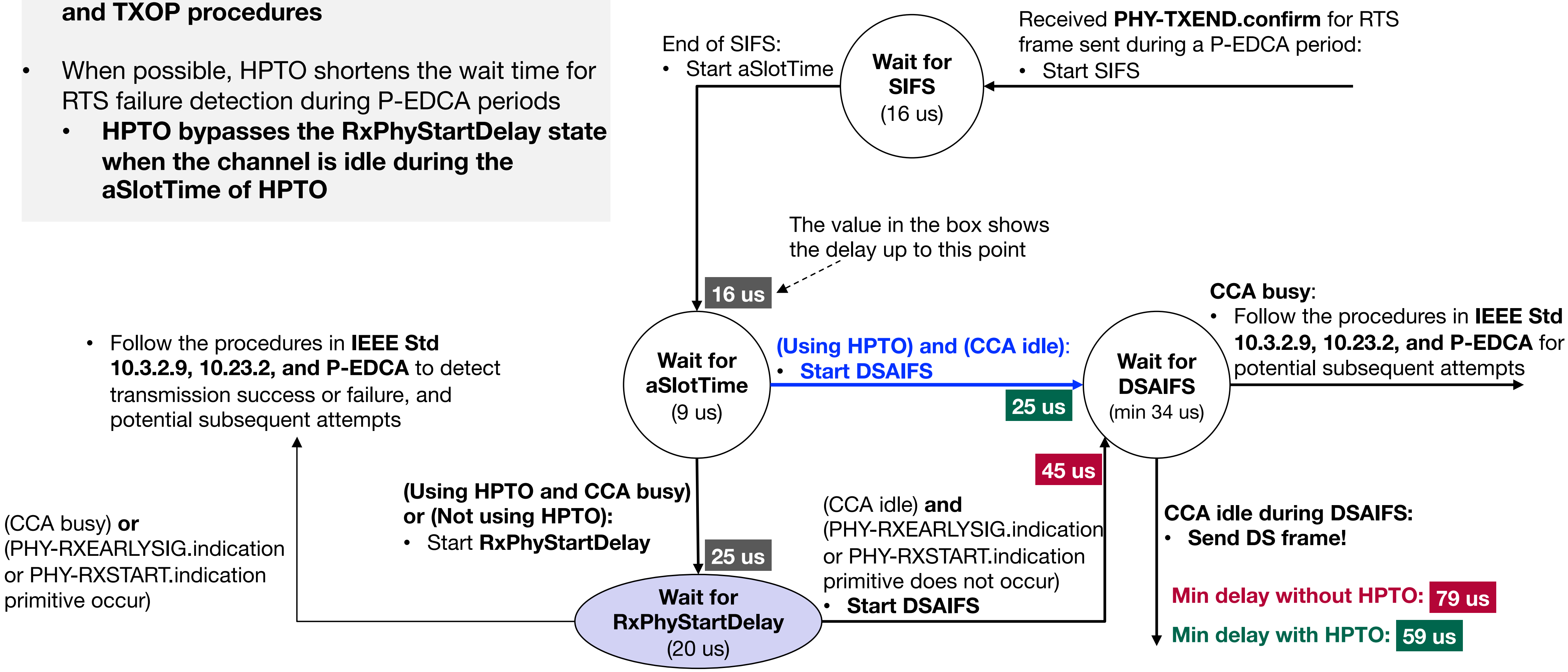
- HPTO = aSIFStime + 2 x aSlotTime
- Compatible with DSAIFS

See State Machine S2 (Slide 10)

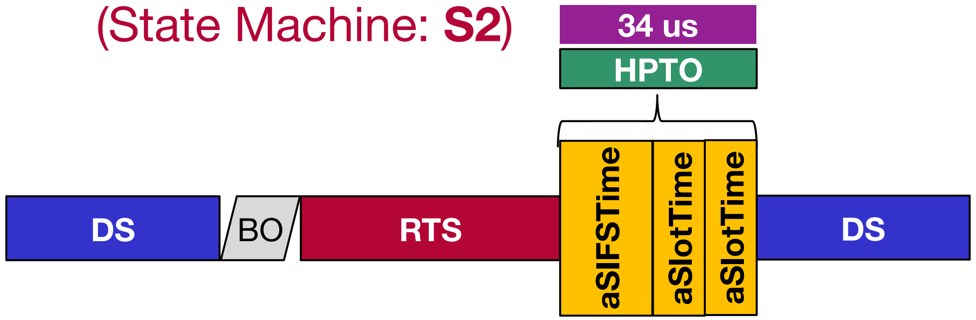


HPTO before DSAIFS (State Machine: S1)

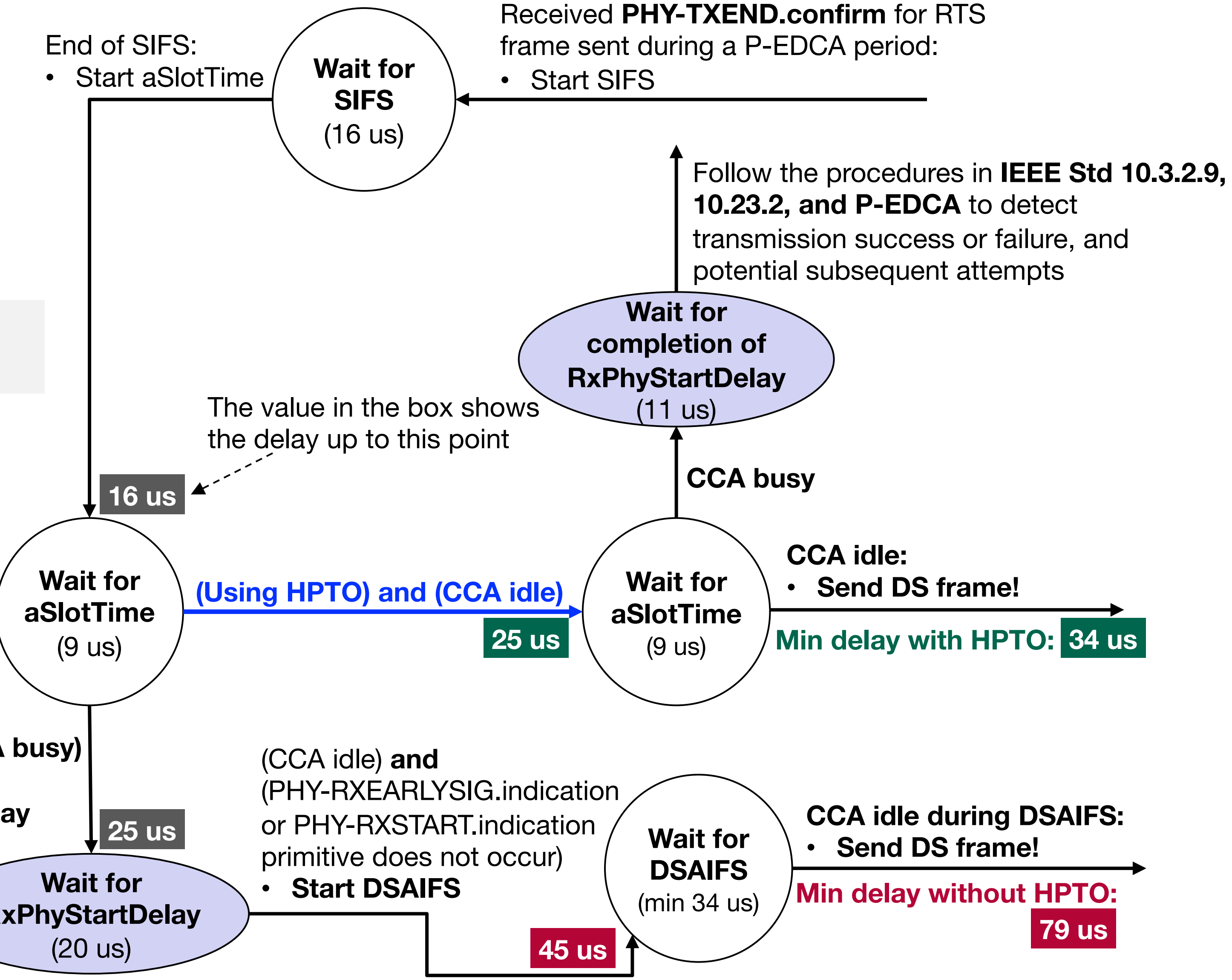
- **HPTO is compatible with the current EDCA and TXOP procedures**
- When possible, HPTO shortens the wait time for RTS failure detection during P-EDCA periods
  - **HPTO bypasses the RxPhyStartDelay state when the channel is idle during the aSlotTime of HPTO**



Integrating HPTO with DSAIFS
(State Machine: S2)

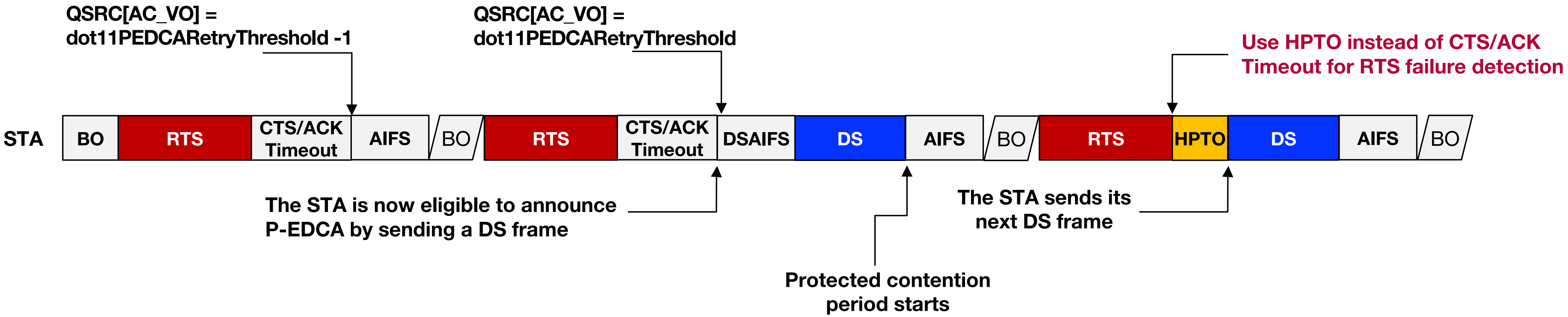


HPTO = DSAIFS[AC_VO]
= aSIFSTime + (AIFSN + DSr) × aSlotTime



❑ Using HPTO within P-EDCA protected contention periods

- Assume a STA starts P-EDCA by sending DS frame when QSRC[AC_VO] = **dot11PEDCARetryThreshold**
- During the P-EDCA's protected contention periods, after sending an RTS frame, the STA uses HPTO to detect RTS transmission failure
- If the channel is sensed as idle during the aSlotTime of HPTO, the STA starts the DSAIFS duration immediately



Summary

- Normally, when a STA sends a response-soliciting frame (e.g., RTS, data), it waits for an CTS/ACK Timeout duration to determine if the transmission has failed

- In this contribution, we proposed that STAs involved in P-EDCA bypass CTS/ACK Timeout and instead use a High-Priority Timeout (HPTO) duration
- **After sending an RTS frame, a STA waits for *HPTO* = *aSIFSTime* + *(AIFSN* + *DSr)* × *aSlotTime* to determine transmission failure**

- The combination of P-EDCA with HPTO:
  - **Allows a higher number of LL STAs use P-EDCA during a shorter time frame**
  - **Reduces the impact of using P-EDCA on legacy STAs**

Straw Poll

SP1. Do you agree that when QSRC[AC_VO] >= dot11PEDCARetryThreshold and PSRC[AC_VO] < dot11PEDCAConsecutiveAttempt, a P-EDCA capable STA may start the High-Priority Timeout (HPTO) once the PHY-TXEND.confirm primitive is received for the transmitted RTS frame, and if the channel is sensed as idle during the HPTO duration, the next DS-CTS frame is sent at the end of the HPTO duration?

- The HPTO duration is defined as HPTO = DSAIFS[AC_VO] = aSIFSTime + (AIFSN + DSr) × aSlotTime

YES/NO/ABSTAIN

OFDM PHY Receiver Specification

IEEE Std 802.11, Section 17.3.10

OFDM PHY Receiver Specification

CCA requirements

- The PHY shall indicate a medium busy condition by issuing a **PHY-CCA.indication** primitive when the carrier sense/clear channel assessment (CS/CCA) mechanism detects a channel busy condition
- For the operating classes requiring CCA-Energy Detect (CCA-ED), the PHY shall also indicate a medium busy condition when CCA-ED detects a channel busy condition
- **The start of an OFDM transmission** at a receive level greater than or equal to the minimum modulation and coding rate sensitivity
    - **–82 dBm for 20 MHz channel spacing**, –85 dBm for 10 MHz channel spacing, and –88 dBm for 5 MHz channel spacing) shall cause CS/CCA to detect a channel busy condition with a **probability > 90% within 4 us** for 20 MHz channel spacing, 8 us for 10 MHz channel spacing, and 16 us for 5 MHz channel spacing

- Additionally, **the CS/CCA mechanism shall detect a medium busy condition within 4 us of any signal with a received energy that is 20 dB above the minimum modulation and coding rate sensitivity** (minimum modulation and coding rate sensitivity + 20 dB resulting in **–62 dBm for 20 MHz channel spacing**, –65 dBm for 10 MHz channel spacing, and –68 dBm for 5 MHz channel spacing)

❑ OFDM PHY Receiver Specification

Receive PHY

- Upon receiving a PHY preamble, the PHY measures the received signal strength level
  - **This indicates activity to the MAC via PHY-CCA.indication primitive**
  - **A PHY-CCA.indication(BUSY) primitive shall be issued for reception of a signal prior to correct reception of the PPDU**
  - The RSSI parameter reported to the MAC in the RXVECTOR

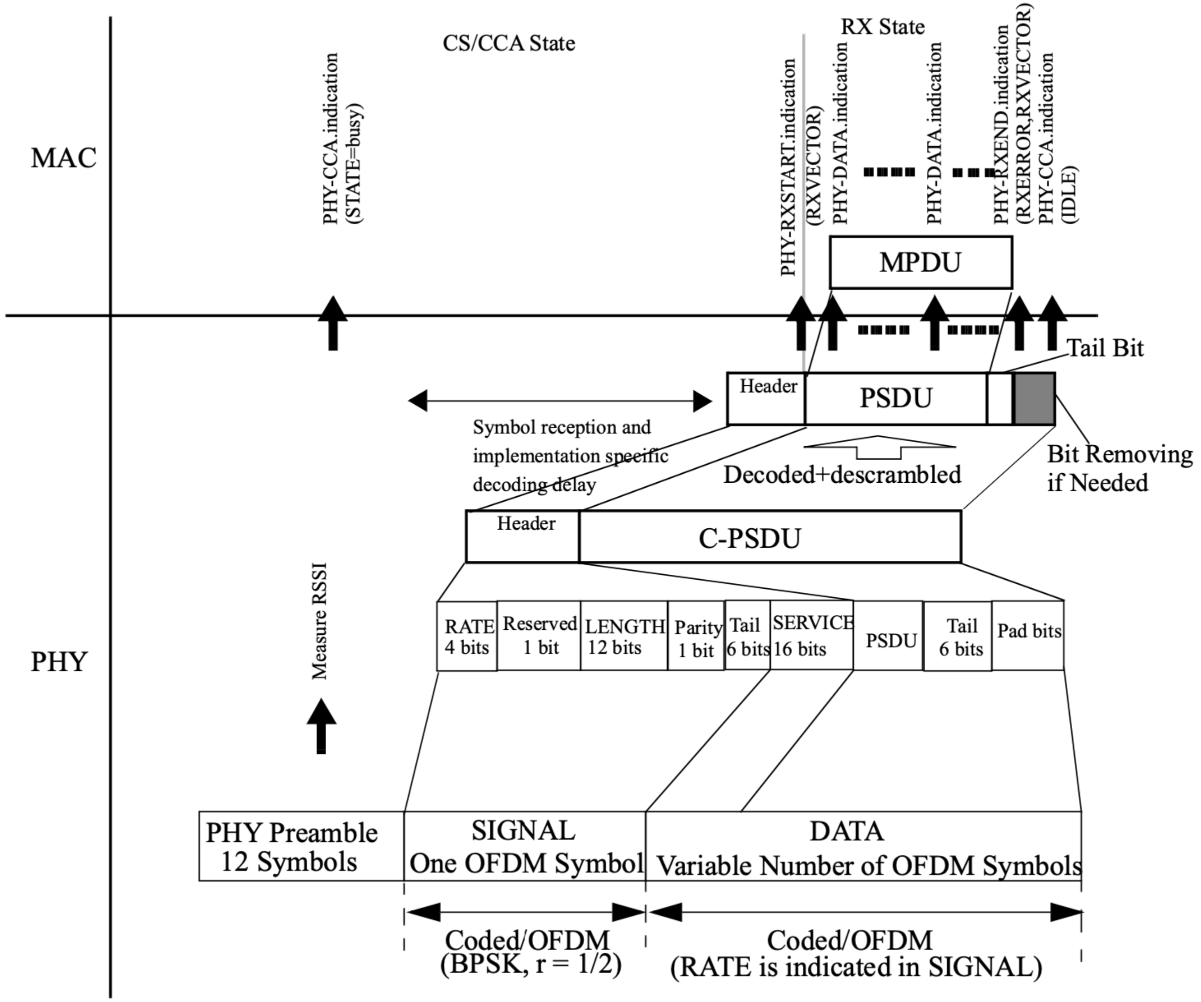


Figure 17-19—Receive PHY

❑ OFDM PHY Receiver Specification

- After a PHY-CCA.indication primitive is issued, the PHY entity shall begin receiving the training symbols and searching for the SIGNAL
  - In order to set the length of the data stream, the demodulation type, and the decoding rate
- **If the PHY header reception is successful (and the SIGNAL field is completely recognizable and supported)**
  - A **PHY-RXSTART.indication(RXVECTOR)** primitive shall be issued

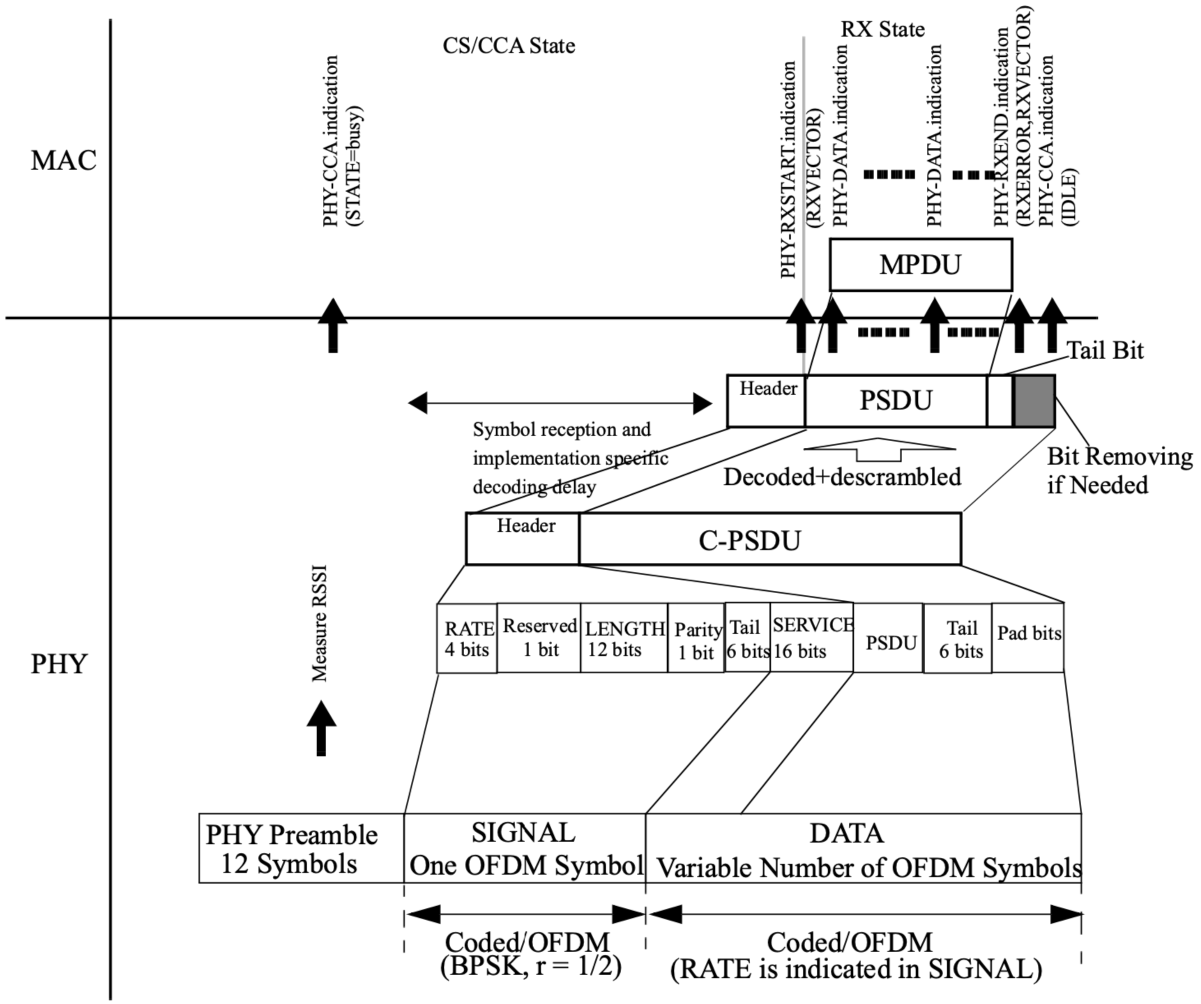


Figure 17-19—Receive PHY

❑ OFDM PHY Receiver Specification

- The received PSDU bits are assembled into octets, decoded, and presented to the MAC using a series of **PHY-DATA.indication(DATA)** primitive exchanges

- The rate change indicated in the SIGNAL field shall be initiated from the SERVICE field data of the PHY header, as described in 17.3.2

- The PHY shall proceed with PSDU reception
- After the reception of the final bit of the last PSDU octet indicated by the LENGTH field of the SIGNAL field, the receiver shall be returned to the RX IDLE state, as shown in Figure 17-19
- A **PHY-RXEND.indication(NoError)** primitive shall be issued

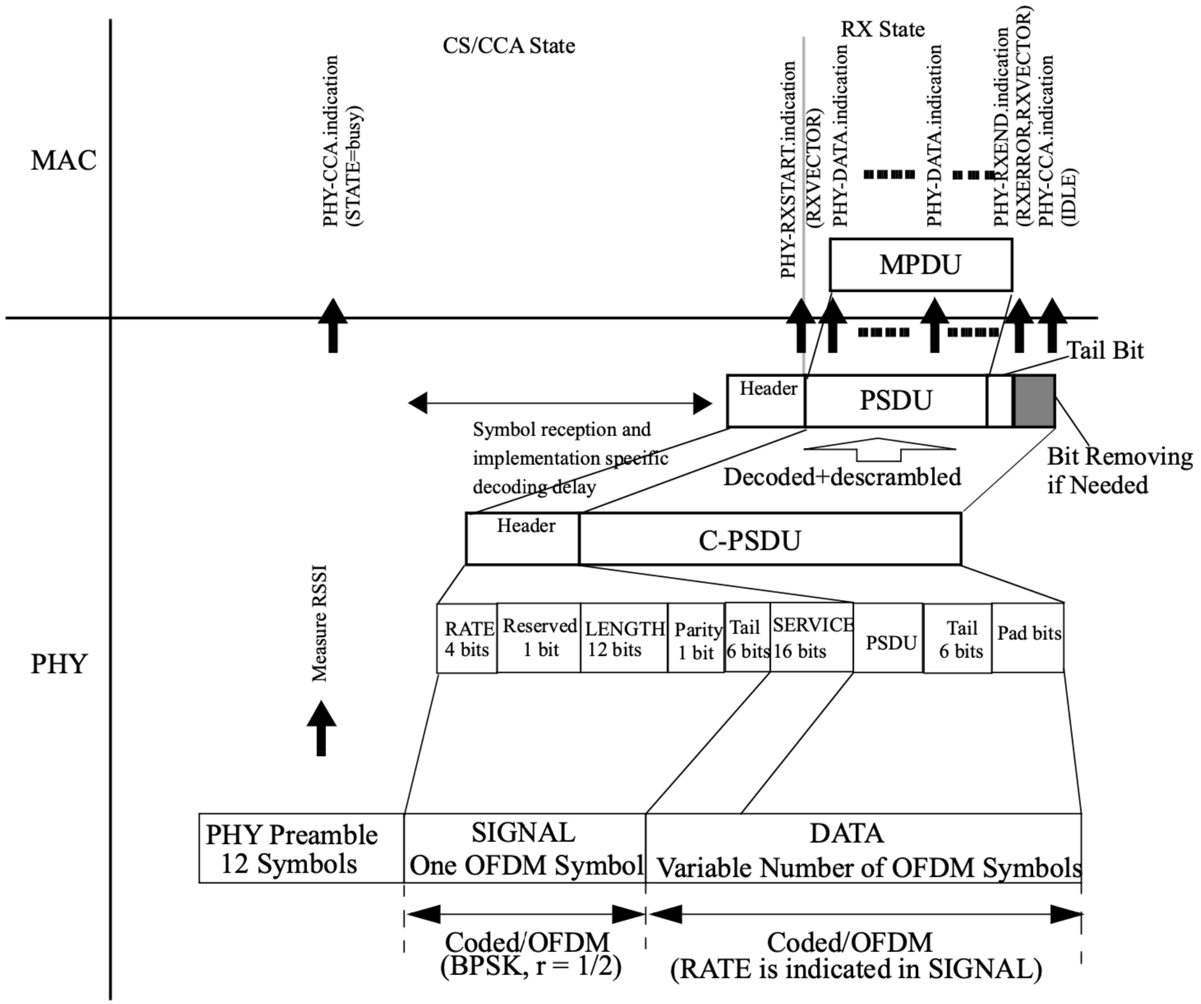


Figure 17-19—Receive PHY

- OFDM PHY Receiver Specification

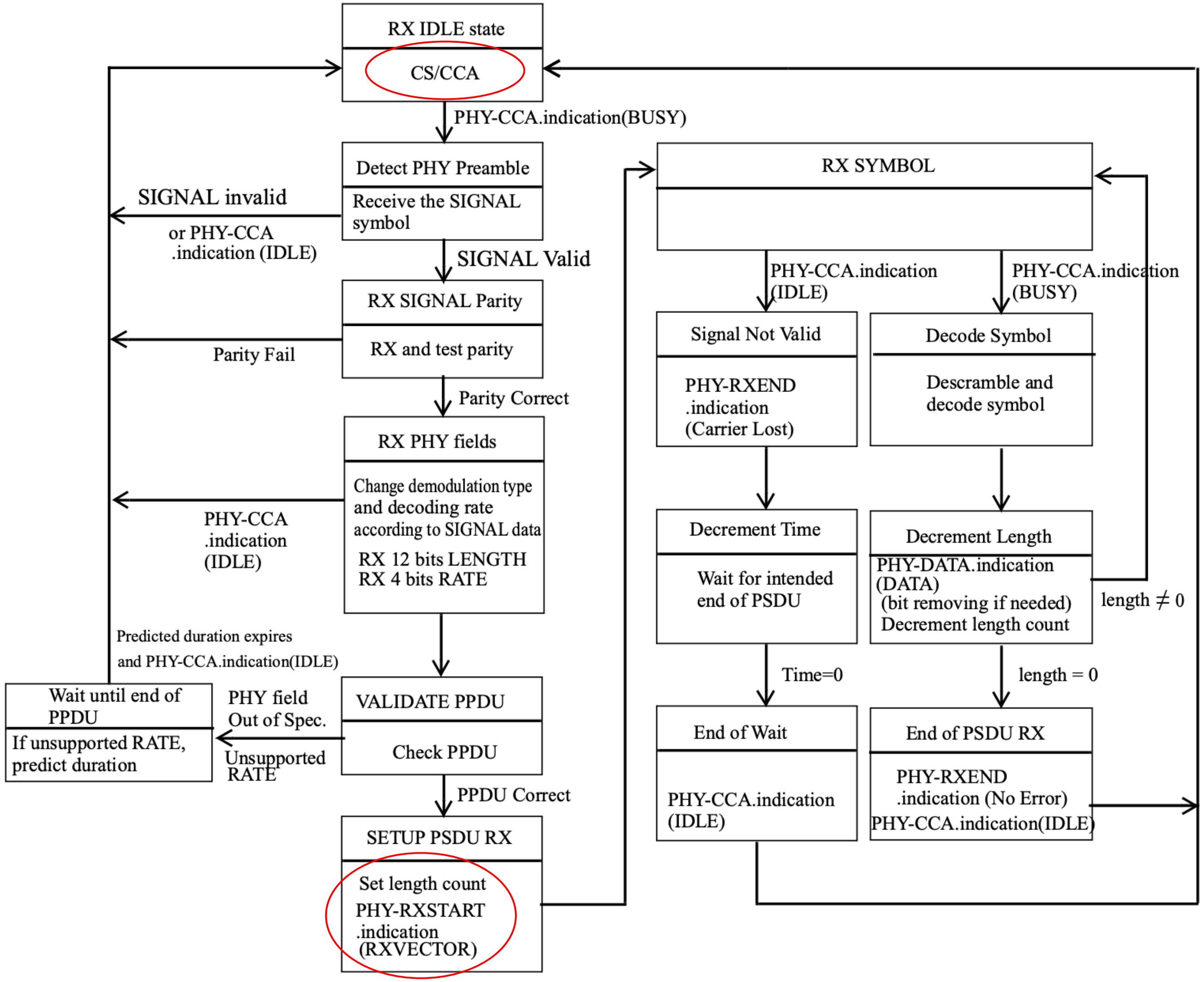


Figure 17-20—PHY receive state machine