

# **Changes to MAC Service Interface Affected Flow Control**

Glen Kramer, Teknovus, Inc.  
glen.kramer@teknovus.com

# How it Was (802.3-2005)...

- The PAUSE transmit FSM assumed that call to the TransmitFrame() function only returned after the entire frame got transmitted, **not instantaneously**.
- This ensured that transmit\_in\_progress variable remained true for the entire duration of frame transmission

## 31B.3.2.2 Variables

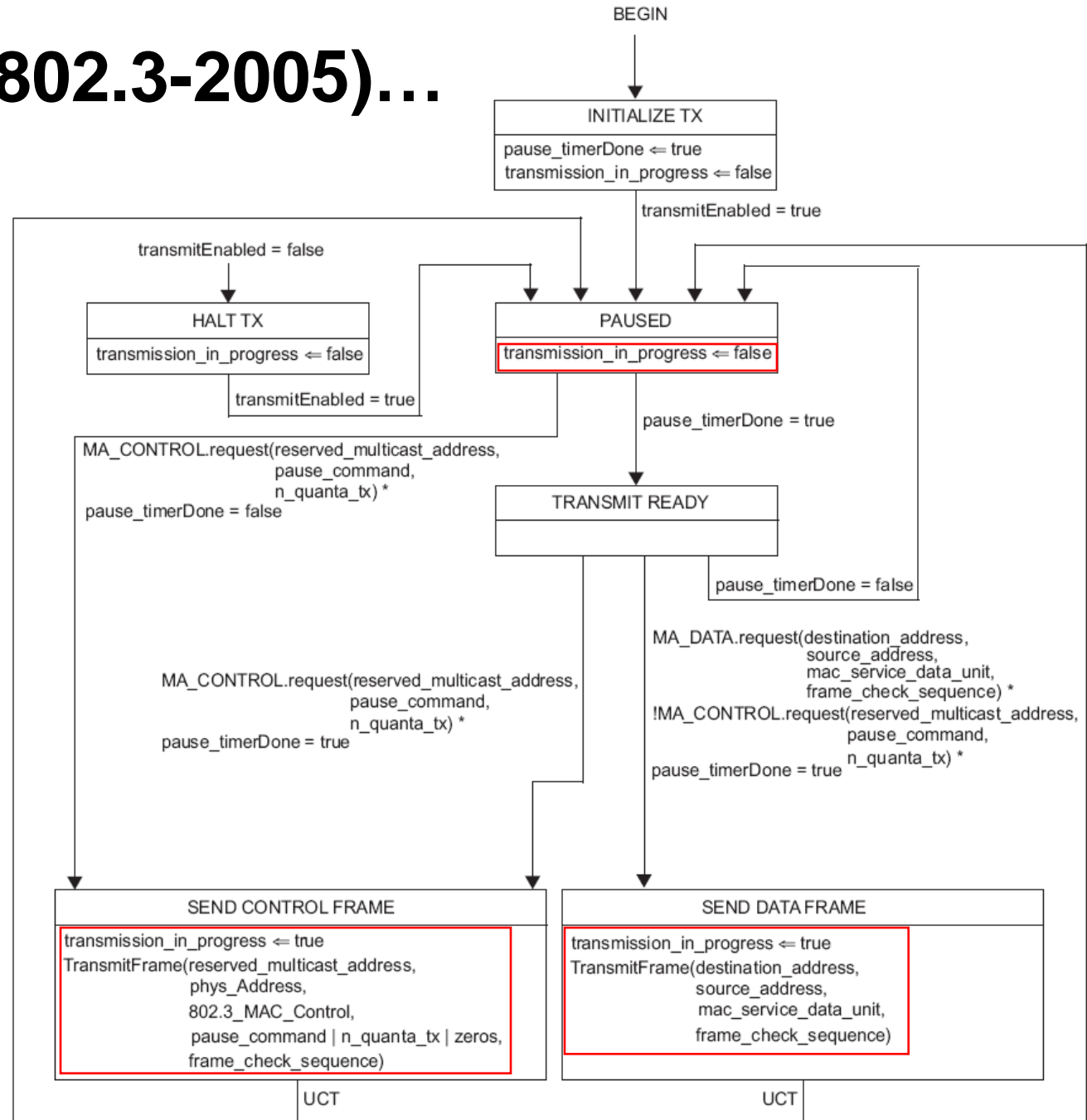
### transmission\_in\_progress

A boolean used to indicate to the Receive state machine that a TransmitFrame function call is pending.

Values:

true; transmission is in progress

false; transmission is not in progress



August 13, 2008

Figure 31B-1—PAUSE Operation Transmit state diagram

# How it Is (after .3as)...

- In states SEND\_CONTROL\_FRAME and SEND\_DATA\_FRAME, the MAC:MA\_DATA.request executes instantaneously and the control passes back to state PAUSED.
- The transmit\_in\_progress is only asserted for an instance and is quickly reset back to false.

• So what?

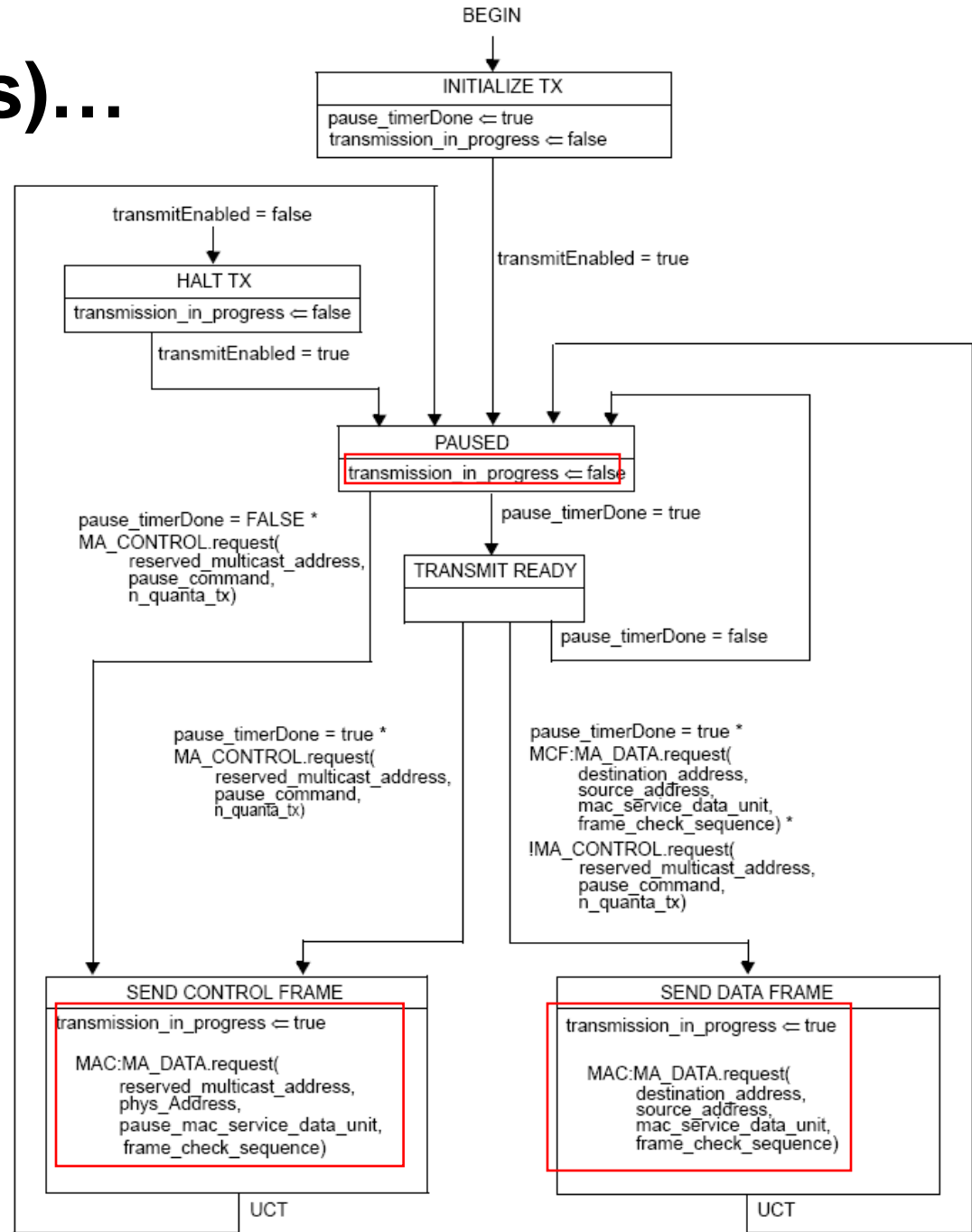


Figure 31B-1—PAUSE Operation Transmit state diagram

# Different Flow Control Operation

- The transmit\_in\_progress variable is used in the receive state machine to decide when to start the Pause timer (it should not start in the middle of transmitting a frame).

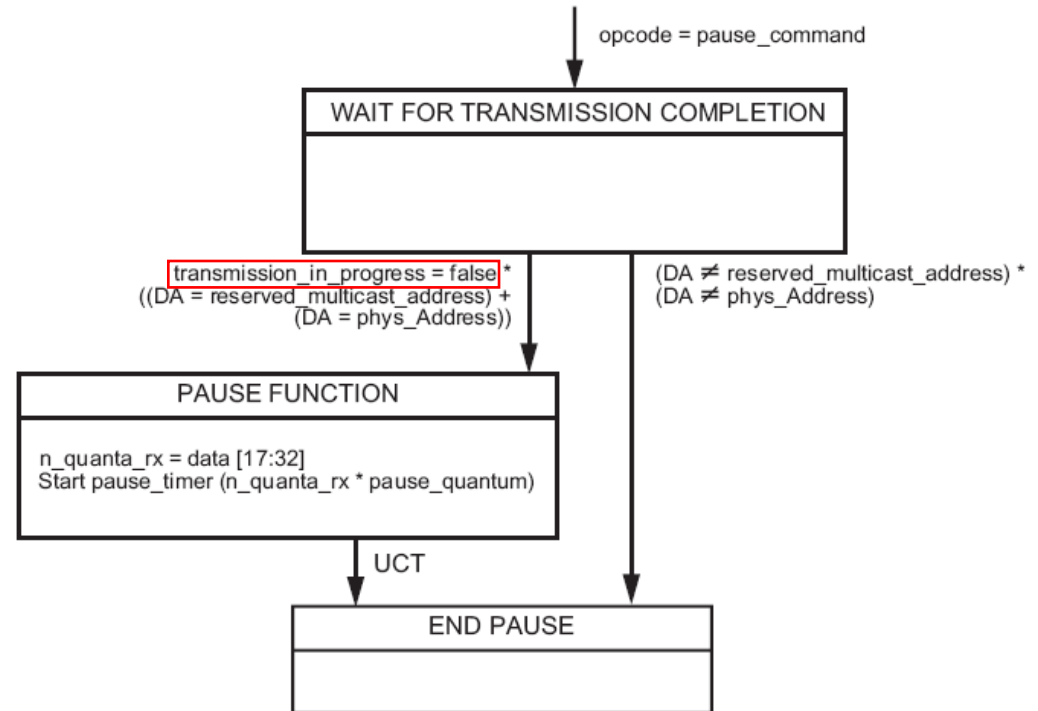
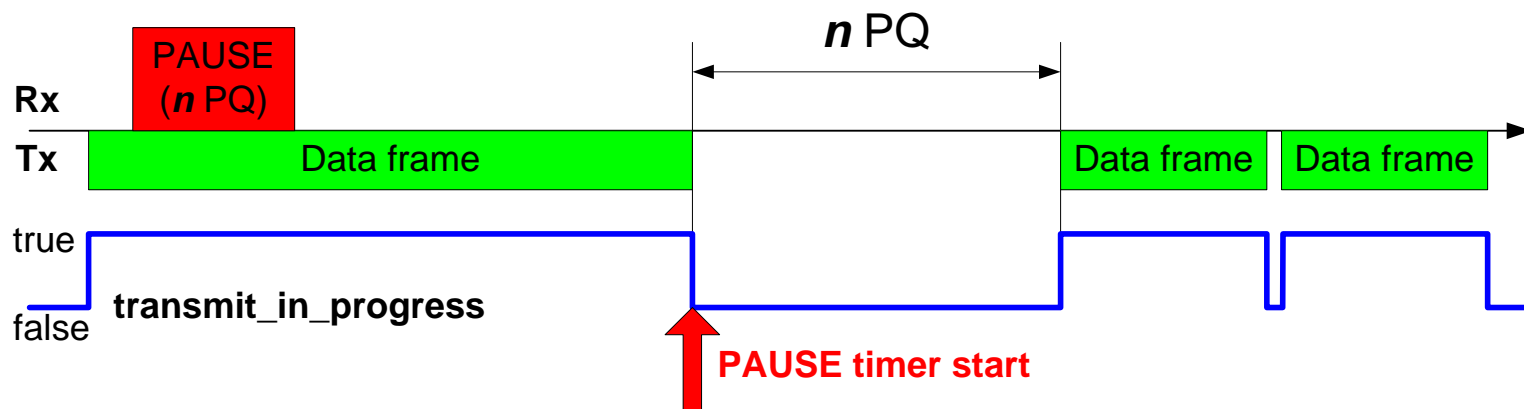


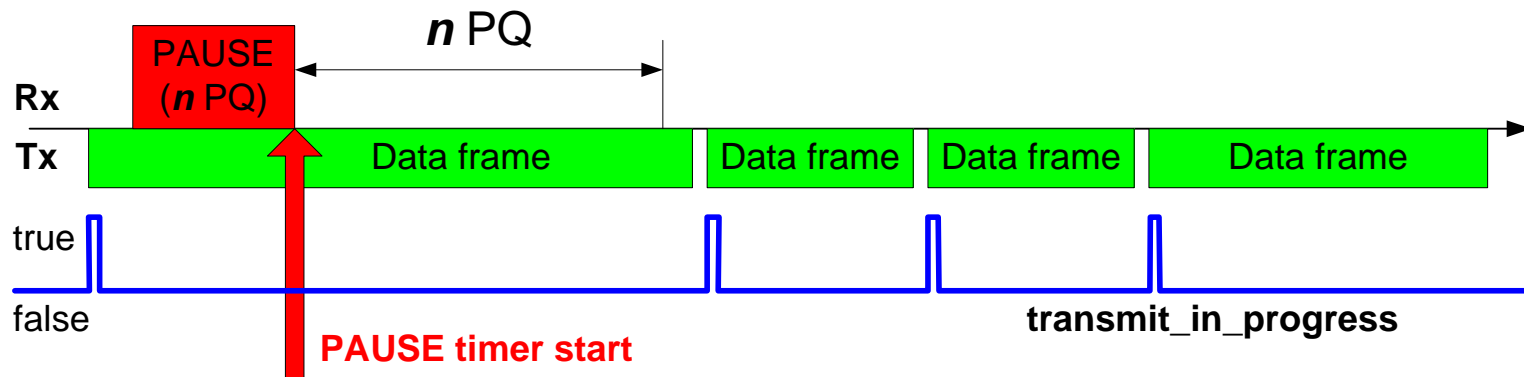
Figure 31B-2—PAUSE Operation Receive state diagram

# Disappearing Flow Control Act

## Original Flow Control Behavior



## Current Flow Control Behavior



- If PAUSE arrives during a frame transmission and pause\_time is shorter than the remaining frame transmission time, the transmitter **will not pause at all!**

# Conclusion

- Today, Flow Control is fundamentally broken
- To fix it, either revert to TransmitFrame and document the function blocking behavior (FSM remains in state until the function completes)
- ...Or...
- Add explicit MAC indication to the MAC Client (flow control in this case) that transmission has ended (as shown →)

August 13, 2008

