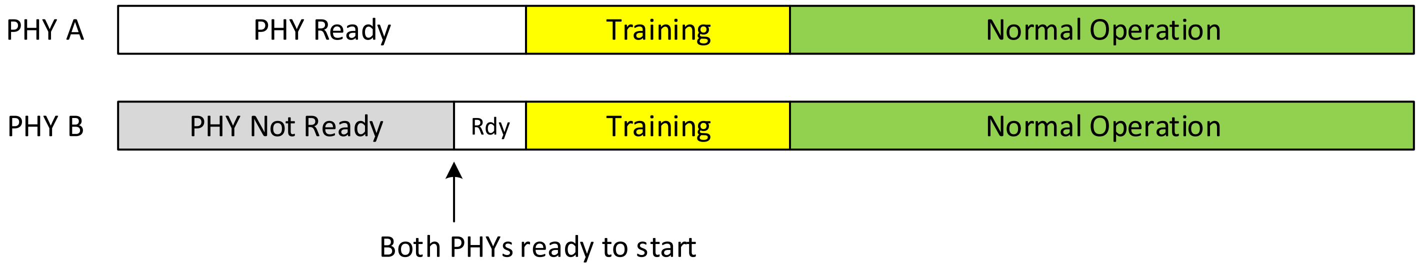
# Use of Auto-Negotiation to synchronize start of training for asymmetric PHYs

William Lo

May 16, 2024

# Why Synchronization Required At Start

- One device may be ready before other
- Makes sure both devices are ready for orderly start of training
- Two methods used in BASE-T1
  - Link Sync
  - Auto-Negotiation (Optional but defined in standard)



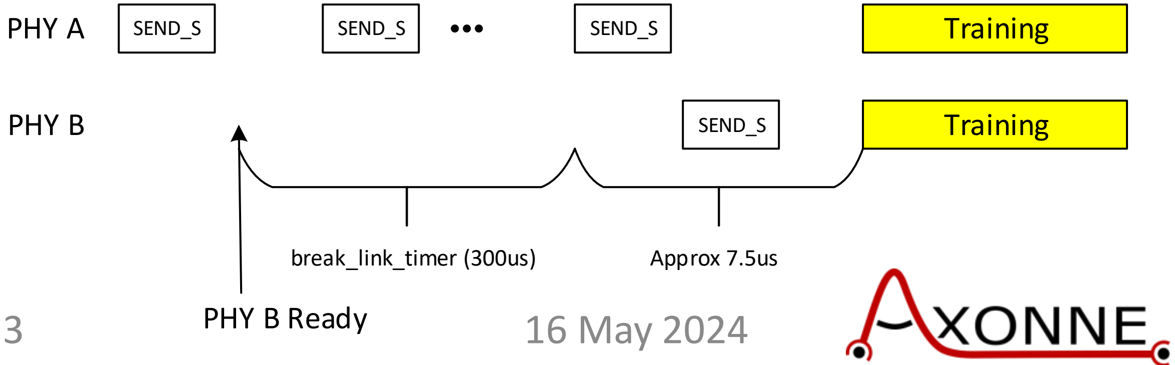
PHY A: PHY Ready | Training | Normal Operation

PHY B: PHY Not Ready | Rdy | Training | Normal Operation

Both PHYs ready to start

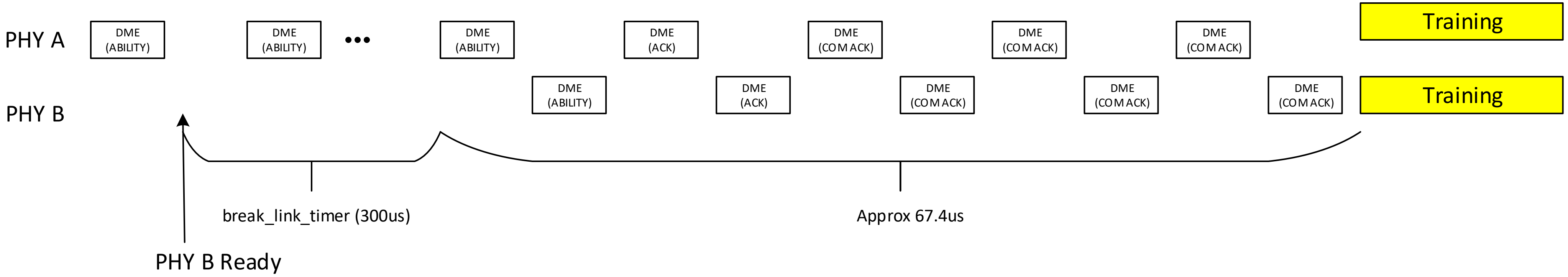
# Link Sync

- Used for engineered system where configuration is known
- Send short 1.25 us burst of PRBS sequence (SEND_S)
  - Very high probability of being detected
- Master keeps sending until it sees SEND_S from slave
- Slave does not transmit SEND_S until it sees SEND_S from master
- Slave only sends one SEND_S burst
- 5.0 us wait after the final SEND_S
- break_link_timer is needed for clean restart
- Fast - from ready to training in approx 307.5 us



# Auto-Negotiation

- Can discover and negotiate capabilities prior to starting
- For engineered system simply advertise only the desired configuration
- DME pages of 4.68 us duration and nominally 2.06 us separating DME pages
- Very robust with acknowledgement from both sides
- Takes 5 DME pages for basic startup
- Can exchange additional info with next pages
- Ready to training in approx 368 ns



# Auto-Negotiation Specification Work

- Use existing Clause 98 single pair Auto-Negotiation
- Add 6 new capabilities:
  - 10GBASE-T1 Up Stream
  - 10GBASE-T1 Down Stream
  - 5GBASE-T1 Up Stream
  - 5GBASE-T1 Down Stream
  - 2.5GBASE-T1 Up Stream
  - 2.5GBASE-T1 Down Stream
- Add new capabilities and priority resolution to lists in Annex 98B
- Add new capabilities to lists in variable definition in 98.5.1

# Question to the task force

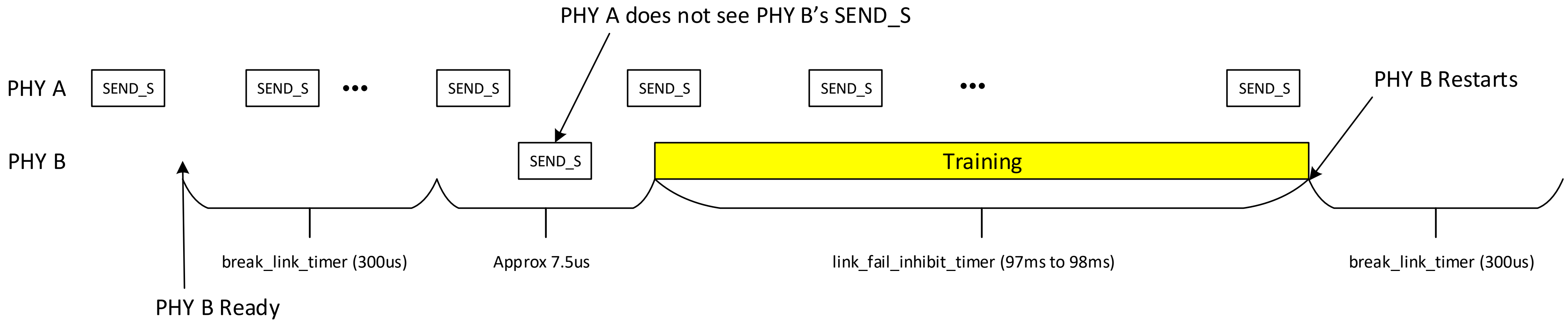
- Do we want to make Auto-Negotiation required for Asymmetrical PHY?

- Eliminates need to specify an asymmetrical link sync mechanism
- Auto-Negotiation is independent of the mission mode modulation
  - Works with EEE, FDD, TDD, CM methods.
- Acknowledge mechanism more robust than link sync (next slide)
- Only takes 60 us longer than link sync

# Link Sync Corner Case

- Slave PHY (PHY B) only send one SEND_S
- If Master does not detect this SEND_S for any reason it does not start training
- Slave starts training since it saw SEND_S
- Only way out is on retry after 97ms link_fail_inhibit_timer times out

- The SEND_S can be robustly detected in functioning systems, but corner case is an architectural oversight



# THANK YOU