Back Channel for Extended Training (BCET) - Applications

Leon Bruckman, Nvidia

Supporters

- Vasudevan Parthasarathy (Broadcom)
- Adee Ran (Cisco)
- Sridhar Ramesh (Maxlinear)
- Jose Vicente Galan Conejos (Maxlinear)
- Vishnu Balan (Nvidia)
- Johan Jacob Mohr (Nvidia)

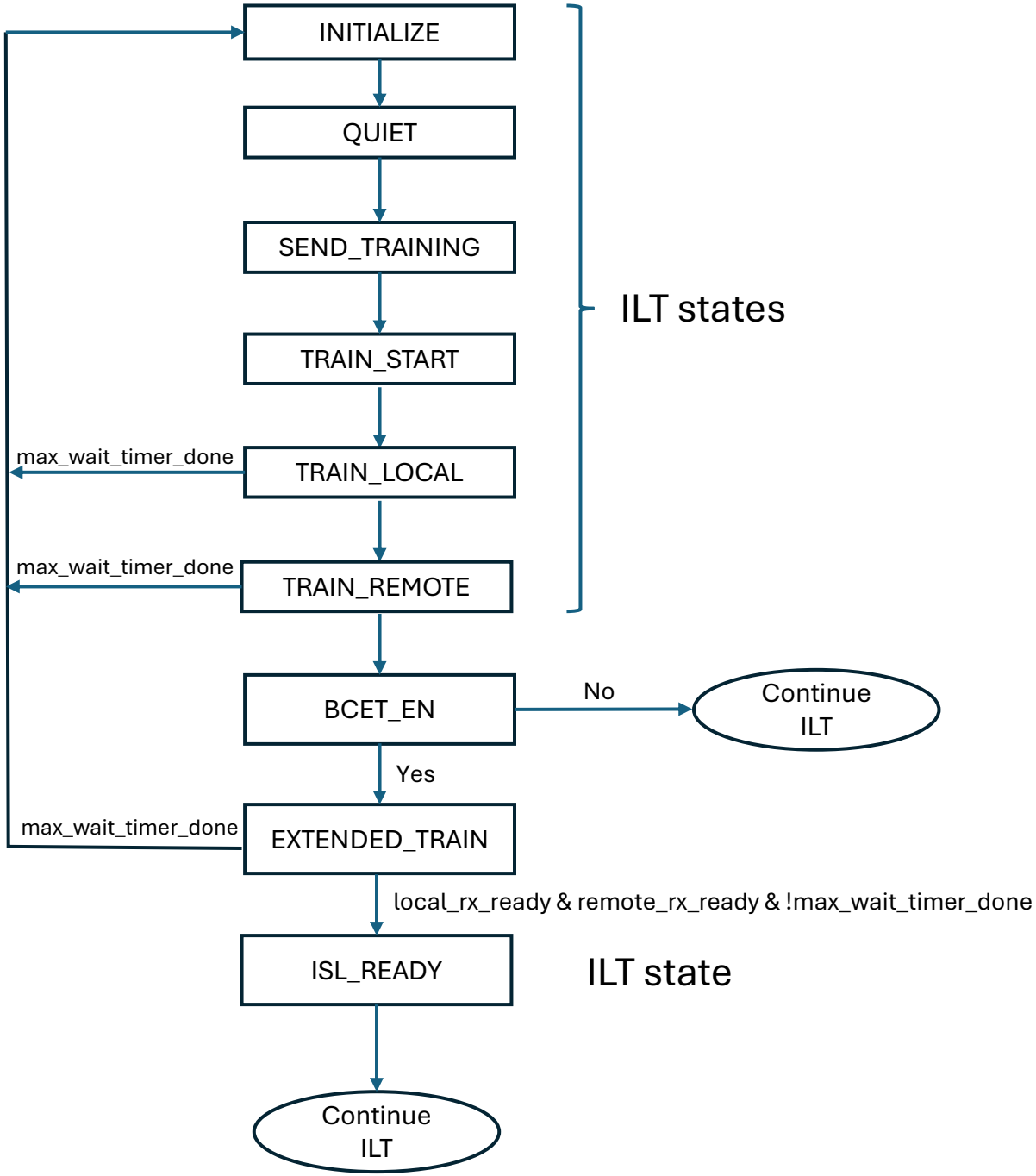
Problem statement

- At 400 Gb/s/lane, the optical receiver may need to tell the far-end optical transmitter much more than “change these few linear equalizer taps,” but the existing ILT signaling in APSU is too limited and inefficient for that exchange.
- BCET is intended to provide a message-based mechanism between the two ends during training, supporting both read and write transactions.

Extended Link Training for high-rate optical links - recap

- Link training as defined in IEEE 802.3 and other standards is based on Link Training (LT) frames that carry a control/status overhead and a PRBS payload.
  - Using the training procedures as defined, the receiver can either request a special Tx equalizer preset or control a limited number of tap coefficients for electrical links.
  - No peer Tx equalizer training capabilities are defined for optical links.
- For optical links operating at high rates, it is beneficial to provide extended training beyond the capabilities provided by conventional Link Training (LT).
  - Conventional LT has low effective bandwidth since each new request must be acknowledged by the receiver.
  - Back Channel for Extended Training (BCET) intent is to improve efficiency of the link training protocol.
- Optional BCET provides a message-based mechanism for peer transmitter control and status exchange, and supports both write and read operations.
  - See: https://www.ieee802.org/3/400GPL/public/2609/bruckman_400GPL_01_2609.pdf
- This presentation shows examples of possible uses of BCET

BCET and ILT


INITIALIZE
QUIET
SEND_TRAINING
TRAIN_START
TRAIN_LOCAL
TRAIN_REMOTE
BCET_EN
EXTENDED_TRAIN
ISL_READY
Continue ILT
Continue ILT
max_wait_timer_done
max_wait_timer_done
max_wait_timer_done
No
Yes
local_rx_ready & remote_rx_ready & !max_wait_timer_done
ILT states
ILT state

BCET use cases:

- Detect peer capabilities
- Configure non-linear look-up table (LUT)
- Vendor specific presets select
- CMIS read
- Presets select

Detect peer capabilities

- Some optional capabilities (see following presentation) may be supported by the peer transmitter
- BCET allows reading info from the peer
- The peer can provide a bit-map with supported optional capabilities, and the receiver can take advantage of them if it decides to
- It also can support future negotiated features
- The concept is similar to the Link Layer Discovery Protocol (LLDP), but at the physical layer
  - Note that the link layer is not active until link training is done
- Other peer characteristics that can be accessed to enhance performance:
  - Module vendor
  - Module type

Configure non-linear look-up table (LUT)

- Some implementations of optical transmitters can benefit from a non-linear LUT
- BCET allows the receiver to read the user capabilities and if support of the optional non-linear LUT capability is advertised, it can take advantage of this capability
- Transport of a 256 bytes LUT using BCET messages and assuming average firmware response time of 2 msec will take approximately ~65 msec.
- The receiver can try several LUTs within the max_wait_timer_duration defined in 802.3dj (60 sec for optical interfaces)
  - May consider a larger max_wait_timer_duration if required for 802.3dv
- Use of this capability by the receiver and implementation by the transmitter should be optional
- In case of a catastrophic LUT configuration that brings the link down (not even able to lock on the LT frames) – Revert to the initialization preset (preset 0)

Vendor specific presets select

- The transmitter must always support an initialization preset – preset 0
- In addition, it may support N vendor specific presets.
  - The receiver can read the value of N supported by the peer transmitter using the BCET read capability, and request to apply the presets, or a subset of them and test the performance
  - The receiver will then select the best performing preset
- Specify basic requirement for the vendor specific presets:
  - Reference receiver shall be able to lock onto the LT frames for any preset within TDM msecs
- In case of a catastrophic preset that brings the link down (not even able to lock on the LT frames) – Revert to the initialization preset (preset 0)
  - The receiver can then remove the offending preset from the list of requested presets

CMIS read

- Access the peer CMIS values
- Values could be used by the receiver to improve remote transmitter control
- Flexible message to allow reading future CMIS additions

Preset select

- Once preset support is indicated by the peer, the receiver can ask for presets from the list
- Number of presets not limited by small field

BCET messages

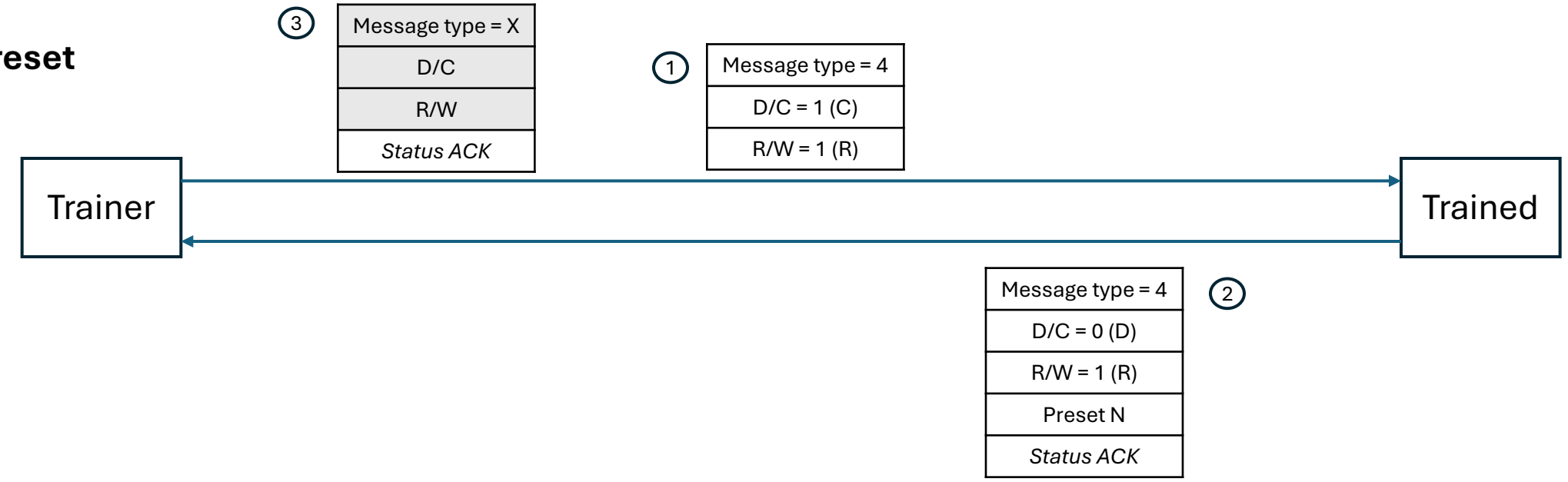
Bit 11	Bit 10	Bit 9	Bit 8	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
D11	*D10*	*D9*	*D8*	*D4*	*D3*	*D2*	*D1*	*D0*
Message type						D/C	R/W	CRC

Type	Message type	Payload length [pages]	Values
BCET support (1)	255	1	Yes/No
Module type	0	3 + N	OUI (3), type (N)
Optional capabilities	1	4	LUT, Presets, …
LUT	2	256	LUT values
CMIS read	3	7 + N	Opcode (1), Page (1), Bank (1), Offset (2), Length (2), Data (0 – 23)
Preset select	4	1	Preset number

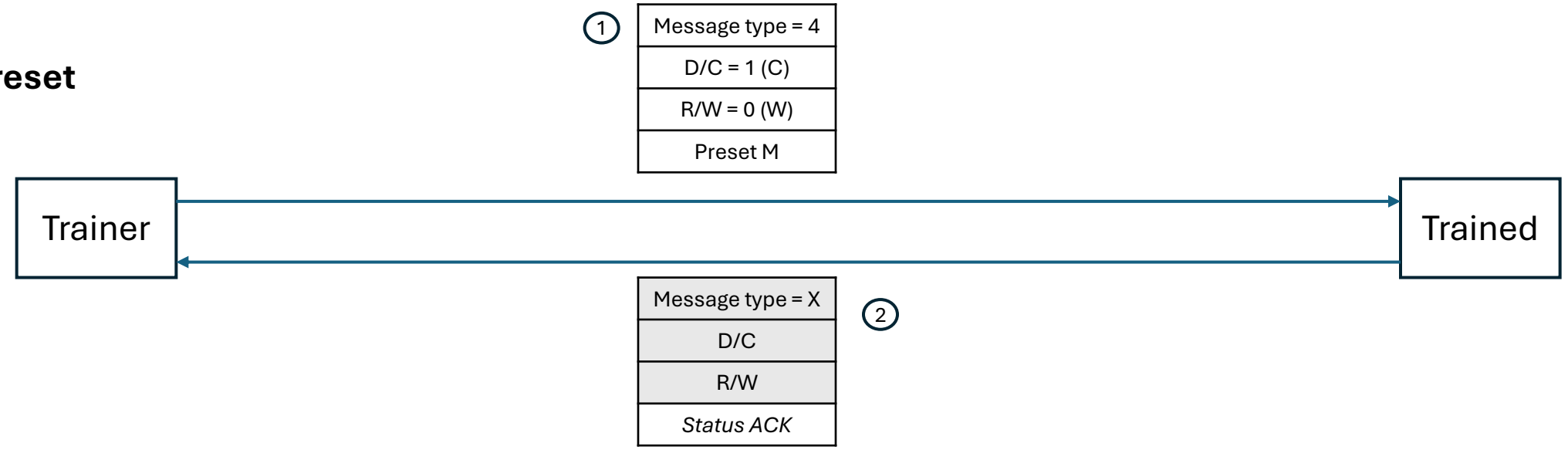
(1) If no ACK received within TBM msec, use standard ILT

Message exchange example – Preset read and write

Read preset



Write preset



Thanks