AN and ILT Timing Update – 20 Feb 2025

Kent Lusted, Synopsys

Adee Ran, Cisco

Supporters

- Mike Dudek, Marvell

Preface

- This presentation follows up on lusted_3dj_adhoc_02a_250206
- The objective is to enable ILT, and optionally AN, with sufficient time for configuration and adaptation, while ensuring that management can restart the process on either side within reasonable time.
- Whether this requires specified timeouts or not – is being discussed.
- A complete proposal is planned for WG ballot phase.

Key points
(from lusted_3dj_adhoc_02a_250206)

Establishing the Solution Space

- If an ISL within a multi-ISL link is not configured/trained yet, it should not cause other ISLs that are have finished their training to restart due to timeout
- Allow time for management to configure all components of the link and of the system (local host, retimer, module, etc.)
- Consider software development and debugging
  - Lab debug / development bring up
  - Field deployments / production environment
  - Field debug
- AN restart should be possible without waiting too long

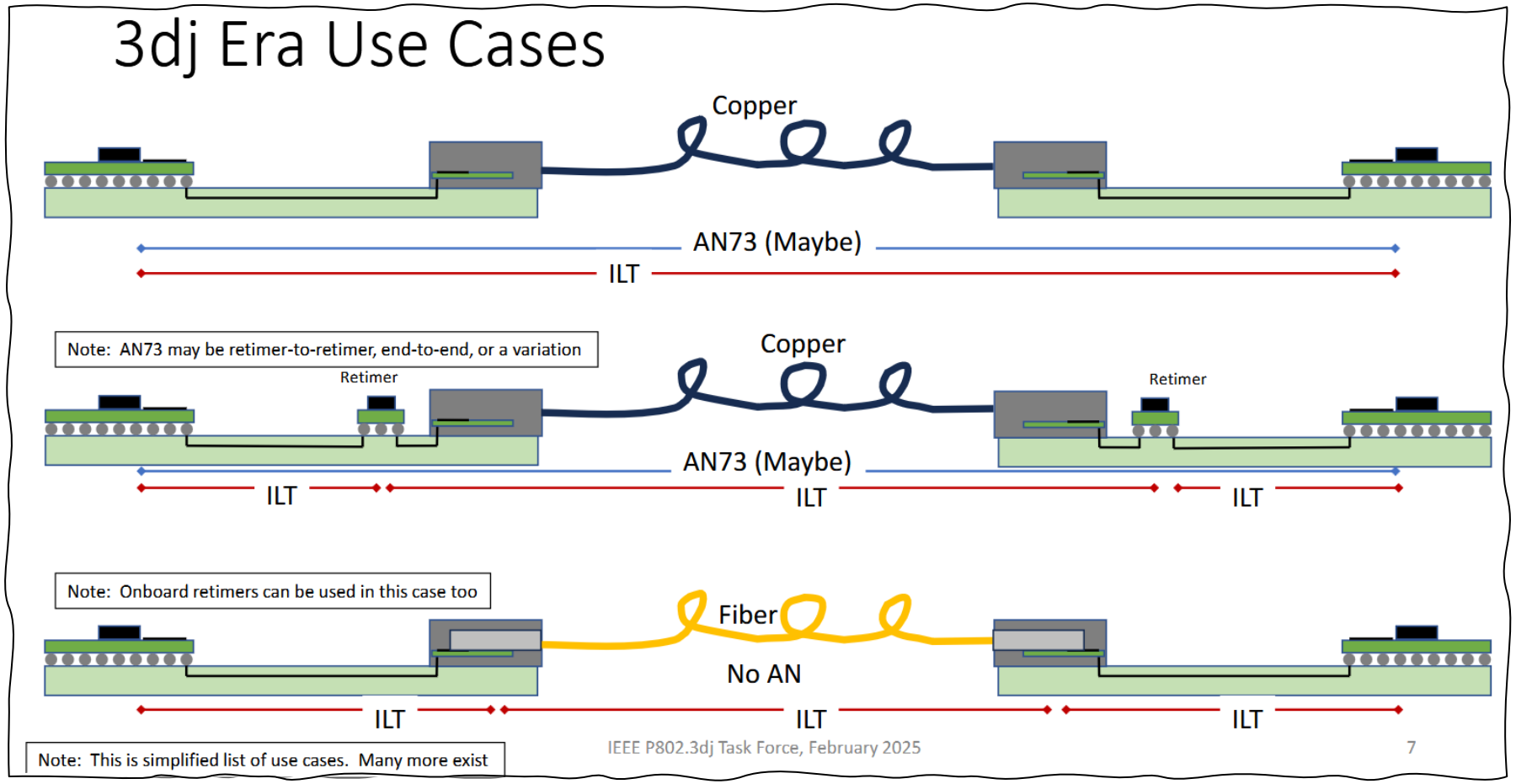
These seem to be in consensus

Summary

- Between now and May interim, working on requirements and use cases towards a complete proposal for consideration during Working Group ballot (D2.0)
  - AN73 timer link_fail_inhibit_timer refinement (slide 4)
  - Should the adaptation time be bound or not? (slide 11)
- Desire to keep AN73-based CR/KR link establishment consistent with user experience at 50G/lane and 100G/lane
  - Retimed copper links need consideration.
  - Non-AN73 copper links… And Optics, too!
- Exploring approaches to ensure interoperability, predictability, debuggability and visibility

Another important point

Some Use Cases



Timing considerations of AN – 3dj Era
(Used in some electrical links)

- When AN page exchange is done, the PCS-to-PCS link is known to be physically connected from end to end
- Management on either side may need to configure the ASIC and possibly a local retimer according to the chosen ability (HCD)
  - This could take a long time depending on both the retimer and the management software
  - Management processor can service many ports in parallel, and have other duties
  - If the HCD is known in advance, it can be much faster
- ILT can run only after the ASICs (and possibly retimers) have been configured
  - It may be long, and may not be performed in all ISLs in parallel
  - But in many cases, it will be fast and parallel
- The time required to bring up the end-to-end path can be much longer than the time consumed by ILT.
  - But in many practical cases ILT will be the dominant period.

Timing in Links without AN – 3dj Era
(all optical links, some electrical links)

- It is assumed that all devices are preconfigured to the data rate
  - May involve some “discovery” and management, e.g., using CMIS
  - The time required for this configuration is beyond the scope of this presentation.
- The time-to-link depends on the time spent in ILT (max across all ISLs).
- In electrical links, adaptation (TRAIN_LOCAL state) is assumed to potentially take a long time…
  - This depends on several factors, see next slide.
- Optical links should be faster (fewer requests exchanged).
- Note that we also don’t limit the time for acquiring training frame lock (transition into TRAIN_LOCAL) and for transitions between other states
  - These should be fast, but implementations might take their time.
  - Time spent in these processes can delay the link-up on both sides.

Dilemma

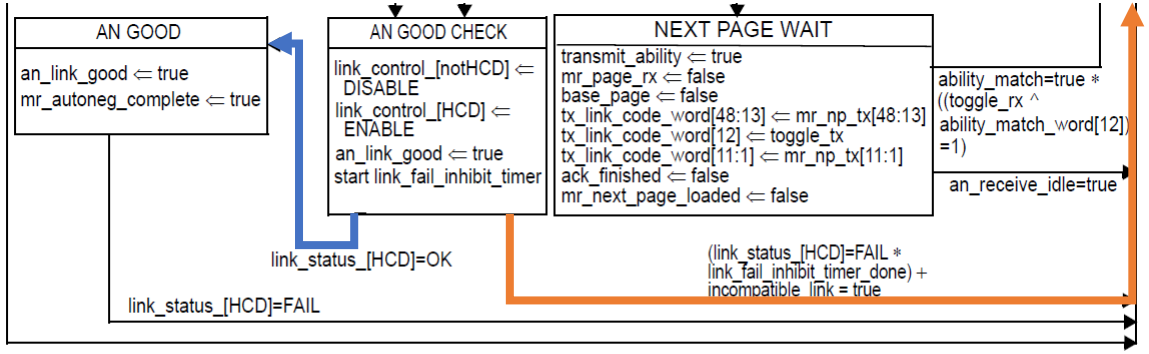


Figure 73–11—Arbitration state diagram

The current definition of link_status allows only OK and FAIL, e.g. in 119.6:

119.6 Auto-Negotiation

The following requirements apply to a PCS used with a 200GBASE-CR4 or 200GBASE-KR4 PMD where support for the Auto-Negotiation process defined in Clause 73 is mandatory. The PCS shall support the AN_LINK.indication(link_status) primitive (see 73.9). The parameter link_status shall take the value FAIL when PCS_status=false and the value OK when PCS_status=true. The primitive shall be generated when the value of link_status changes.

PCS_status is defined in 119.2.6.2.2:

PCS_status

A Boolean variable that is true when align_status is true and is false otherwise.

So link_status is essentially align_status.

- With the existing AN arbitration state diagram, link_fail_inhibit_timer is both "max **time-to-link**" and "min **time-to-retry**"
- Implementations with unknown HCD (and possibly retimers) would benefit from allowing a long "**time-to-link**" ➔ increase link_fail_inhibit_timer
- Implementations with known HCD and no retimers would prefer a short "**time-to-retry**" ➔ decrease link_fail_inhibit_timer

Adaptation time

- This presentation assumes adaptation is implemented using firmware (FW).
- The time spent in TRAIN_LOCAL depends on
  - **Local firmware implementation** – delay between HW generating a request and FW processing and sending it to the partner
  - **Remote firmware implementation** – delay between HW receiving a request and FW handling and responding to it
  - **Rx local adaptation algorithm** – how long it takes to generate the next request
  - **“Search” algorithm** – how many transactions are required
- The processor running the firmware can service multiple lanes in parallel, and have other duties
  - Multi-tasking software can take many forms; not always time-optimized
  - This should not be a “hard real-time” system
- The time spent in TRAIN_REMOTE depends on the link partner…
  - Similar considerations, but the local device has no control.
  - Any recommendation should not include TRAIN_REMOTE.

To timeout, or not to timeout?

“timeout is needed”

- Not having a specified timeout would allow implementations with extremely long adaptation times
  - A device cannot predict how long its partner will require for adaptation – no implementation-specific timeout is “safe”
- Predictable customer experience is important
- Test times should be considered.

“timeout is not needed”

- Imposing a timeout for adaptation by the standard would limit implementation flexibility
  - Deployment of 200G technology is still at early stage – we have partial information
- Having no specified timeout will improve implementation flexibility and interoperability
  - Link-up time can be a differentiating factor
- Debugging deployed links should be considered.

Summary

- Going forward, working on requirements and use cases towards a complete proposal for consideration during Working Group ballot (D2.0)
  - AN73 timer link_fail_inhibit_timer refinement
  - Should the adaptation time be bound or not?
  - Mechanism of restart
- Desire to keep AN73-based CR/KR link establishment consistent with user experience at 50G/lane and 100G/lane
  - Retimed copper links need consideration.
  - Non-AN73 copper links… And Optics, too!
- More study & discussion on link_fail_inhibit_timer is needed
  - It is currently both “max time-to-link” and “min time-to-retry”
- Reach out to us to get involved in offline consensus building meetings

That’s all

Questions?

Backup

Example (for illustration only)

- Assume the following:
  - Delay between HW generating a request and FW processing and sending it to the partner: 25 ms
  - Delay between remote HW receiving a request and FW handling and responding to it: 25 ms
  - How long it takes to generate the next request: 50 ms
  - How many transactions are required: 120
- Total time spent in TRAIN_LOCAL: 120*(25+25+50) ms=**12 seconds** – same as the max_wait_timer value for 802.3ck PMDs
- **The calculation above may not cover all implementations. Hence it should not be mandatory.**

Proposal for addressing adaptation time

178B.11 Equalization control

Equalization control is only available for E1 interfaces.

When the training control state diagram (Figure 178B–8) is in the TRAIN_LOCAL state, the device may request its peer interface to change the transmitter equalization coefficients, either to predefined initial conditions or by individual coefficient control. The criteria for initiating such requests are implementation dependent.

When the training control state diagram (Figure 178B–8) is in either the TRAIN_LOCAL or TRAIN_REMOTE states, the device shall respond to requests received from the peer interface.

When the training control state diagram (Figure 178B–8) is in any state other than TRAIN_LOCAL or TRAIN_REMOTE, the device shall not send any request to the peer interface and shall ignore requests from the peer interface.

A new individual coefficient update request or initial condition update request is not initiated until after the prior request has completed.

Insert the following text at the end of 178B.11:

It is recommended that the time spent in the TRAIN_LOCAL state be no more than X seconds.

X=12 ?

Alternative using a timer – see backup slide