AN timeout and fast restart mechanism

Adee Ran, Cisco

Overview

- The issue of AN timeout (**link_fail_inhibit_timer** terminal count) has been addressed by several presentations in P802.3dj.
  - It is currently set to 60 seconds.
- The most recent contribution, ran_3dj_02a_2503:
  - Highlighted the meaning of this timer as both “maximum time to link” and “minimum time to retry”
  - Proposed to break this connection and enable a faster retry (leveraging ILT)
  - Had supporters and did not raise significant issues
- However, due to the state of the project, it was preferred not to make the suggested changes, and the associated comment was rejected.
- This presentation is intended to build consensus towards adopting the proposal during Working Group ballot.

Comments against D1.4

Cl **73** *SC* **73.10.2** *P* **134** *L* **15** # 234

Ran, Adee Cisco

Comment Type **T** *Comment Status* **R** *AN/LT timers*

A value of 60 seconds for link_fail_inhibit_timer does not guarantee a reasonably short time-to-link, and on the downside it creates an unacceptably long time to recover from a failed auto-negotiation attempt if at least one of the link partners adheres to it.

The current value was adopted in order to allow ILT in all ISLs to complete. This should be maintained, but the time to recovery from failure (or enable restart by management) should be shorter,

This can be enabled by adding a third possible value IN_PROGRESS to pcs_status. The rules for generating this value can be derived from existing PCS variables.

With this new value, the period for link_fail_inhibit_timer can be reduced to 12 seconds (as in 802.3ck) or even lower.

SuggestedRemedy

A detailed proposal will be submitted.

Response *Response Status* **C**

REJECT.

The following contribution was reviewed by the CRG:
https://www.ieee802.org/3/dj/public/25_03/ran_3dj_02a_2503.pdf

There was no consensus to implement the proposed changes at this time. Further work and consensus building on this topic are encouraged.

The proposed changes are not required to make this draft technically complete. The commenter is encouraged to pursue this further during Working Group ballot.

Cl **178B** *SC* **178B.14.3.5** *P* **793** *L* **20** # 282

Ran, Adee Cisco

Comment Type **T** *Comment Status* **R** *AN/LT timers*

There may be a desire to limit the time consumed by the adaptation part of ILT. This can be done by adding a timer that would be accessible by management.
Since a local device does not control the timing of the link partner, the timer should be active only during the TRAIN_LOCAL state.
The timer period should be set by the invoking clause, and should be a configurable by management, with perhaps a recommendation in the standard.

SuggestedRemedy

Modify Figure 178B-8, adding a timer, as follows:
In the Train Local state, add "start training_timer".
In the Train Remote state, add "stop training_timer".

Add a new timer definition in 178B.14.3.3:
training_timer
This timer is started when the training control state diagram on a lane enters the TRAIN_LOCAL state (see Figure 178B-8). The terminal count of this timer is controlled by the management variable training_timer_duration. The effect of expiration of this timer is implementation dependent.

Add a new variable definition in 178B.14.3.1:
training_timer_duration
Variable that controls the terminal count of training_timer. The default value of this variable is defined by the PMD or AUI component specification.

Add a statement in each PMD clause (e.g., in 179.8.9) setting the default value of training_timer_duration to 60 seconds (matching the adopted link_fail_inhibit_timer).

Response *Response Status* **C**

REJECT.

Resolve using the response to comment #234.

Summary of the proposal

- Add a third possible value to the **link_status** parameter of the AN_LINK.indication primitive (generated by the PCS):
  - Existing values are **OK** and **FAIL**
  - New value, **IN_PROGRESS**, indicates that ILT has started and is still in progress
  - A value of **IN_PROGRESS** will not cause a restart of AN even if **link_fail_inhibit_timer** expires
  - Failure of ILT on any of the ISLs in the path will propagate to the PCS and cause the value to become **FAIL**, which will restart AN
    - This will happen on both sides through loss of training frame lock
  - **Link_fail_inihibit_timer** will keep the meaning as “minimum time to retry”
    - But “the maximum time to link” can be longer, and there will be no mandatory restart mechanism
  - The new behavior is mandatory only for new PHYs (those using SM-PMA)
    - No new requirements from existing PHYs
- Add a timer to ILT as an indication to management.

Proposed changes in the draft: PCS clauses

In each of the PCS clauses (119, 172, 175):

- Change the definition of **link_status** in the Auto-negotiation subclauses (119.6, 172.6, and 175.7). For example, in 119.6:

  The following requirements apply to a PCS used with a 200GBASE-CR4, 200GBASE-CR2, 200GBASE-KR4, 200GBASE-KR2, 400GBASE-CR4, or 400GBASE-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~~ one of the values FAIL, IN_PROGRESS, or OK, according to Table 119-2a.

  The primitive shall be generated when the value of link_status changes.

Table 119-2a--AN_LINK.indication(link_status) generation

reset + restart_lock	align_status	use_in_progress	link_status
True	Don't care	Don't care	FAIL
False	False	False	FAIL
		True	IN_PROGRESS
	True	Don't care	OK

172.6 and 175.7 can refer to this table.

- Add a definition for a new variable **use_in_progress** (as in the next slide) and an MDIO register mapping (see slide 7).

Definition of use_in_progress

- For example in 119.2.6.2.2:

  use_in_progress

  Boolean variable indicating support of the value IN_PROGRESS for the link_status parameter of AN_LINK.indication (see 119.6). It is true for a PCS in the same package as a SM-PMA. Otherwise, its value is implementation dependent.

In clauses 172 and 175, refer instead to 172.6 and 175.7, respectively.

Positioning of AN

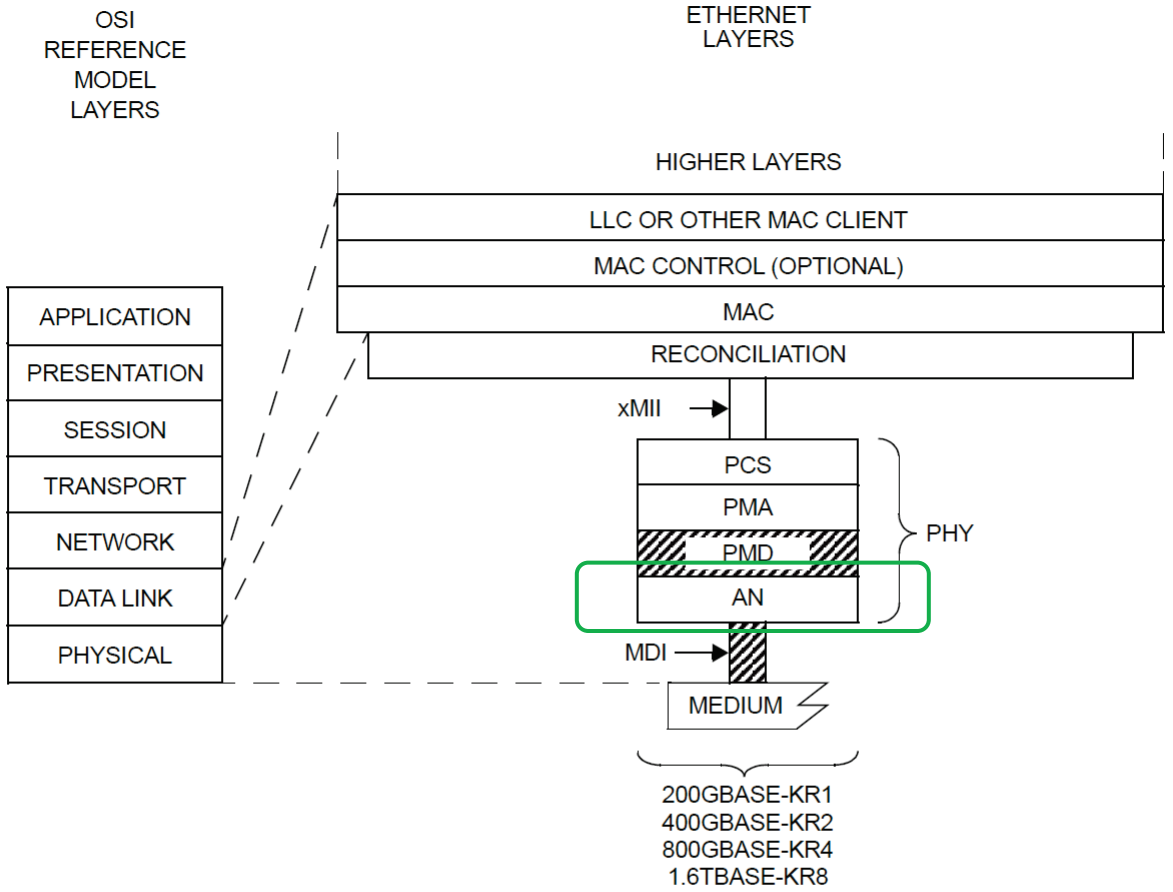

Figure 178–1—200GBASE-KR1, 400GBASE-KR2, 800GBASE-KR4, and 1.6TBASE-KR8 relationship to the ISO/IEC Open Systems Interconnection (OSI) reference model and the IEEE 802.3 Ethernet model

Proposed changes in the draft: AN and MDIO

- In Clause 73 (AN):
  - Change the semantics of AN_LINK.indication (73.9.1.1) as follows:

    The link_status parameter shall assume one of ~~two~~ three values: OK, IN_PROGRESS, or FAIL, indicating ~~whether the underlying receive channel is intact and enabled (OK) or not intact (FAIL)~~ the status of the PCS alignment, as defined in the PCS clause.
  - Change the terminal count of link_fail_inhibit_timer for the new PHYs to be the same as those of 802.3ck (min:12.3, max:12.4 seconds).
- In clause 45 (MDIO), redefine bits 4:3 (currently reserved) in PCS status 1 register (Register 3.1):
  - Bit 4: In-progress ability (1 indicates that **use_in_progress** is true)
  - Bit 3: In-progress indication (1 indicates that **link_status** has the value **IN_PROGRESS**)

Restarting AN after the link is up

- In an active link, AN can be restarted by either side by asserting **reset** in the PMD (e.g., MDIO register 1.0.15).
  - This will cause a restart of the training control state diagram and squelch the transmitter.
  - The link partner's PCS will get uncorrectable codewords (and lose AM lock), which will cause its link_status to become FAIL.
- Unplugging the cable (e.g., case 1U in lusted_3dj_01_2505) will have the same effect.
- Formally, resetting the PCS will also reset any sublayers in the same package, so it will also reset the PMD.
- The process described above is based on the formal service interface and variable definitions, and works regardless of the existence of retimers on either side of the link.
  - Management is formally not required to intervene, but implementations may vary.
- **This is existing behavior that is maintained by this proposal.**

Restarting AN before the link is up

- Before ILT is completed, one of the partner may restart AN (e.g. because it was reset, or due to management action).
- Since the link is not up yet, the link partner's PCS has signal_ok=false, and its link_status is set to IN_PROGRESS.
  - The link partner will eventually lose training frame lock and its training control state diagram will go to FAIL, but it will not automatically cause AN restart, because the PCS is waiting for ILT to complete.
  - The failure of ILT should cause a restart of both ILT and AN. This is formally done by management, but implementations may vary.
- **This is a new feature that is enabled by this proposal.**

ILT timer

178B.14.3.1 Variables

mr_training_timer_duration

Unsigned integer variable that controls the terminal count of training_timer in seconds. A value of 0 corresponds to an infinite time. The default value of this variable is defined by the PMD clause or AUI annex.

178B.14.3.3 Timers

quiet_timer

This timer is started when the training control state diagram on a lane enters the QUIET state (see Figure 178B–8). The terminal count of this timer is between 100 ms and 200 ms.

propagation_timer

This timer is started when the training control state diagram on a lane enters the PATH_READY state (see Figure 178B–8). The terminal count of this timer is between 100 ms and 200 ms.

recovery_timer

This timer is started when the training control state diagram on a lane enters the RECOVERY state (see Figure 178B–8). The terminal count of this timer is between 20 ms and 30 ms.

training_timer

This timer is started when the training control state diagram on a lane enters the TRAIN_LOCAL state (see Figure 178B–8). The terminal count of this timer is controlled by the management variable mr_training_timer_duration. The effect of expiration of this timer is implementation dependent.

Add MDIO mapping for **mr_training_timer_duration** (RW) and **training_timer_done** (RO) in Table 178B–6 and in Clause 45.
Add default values of mr_training_timer_duration: in clauses 178 and 179 – 60, in annexes 176C and 176D – 0.

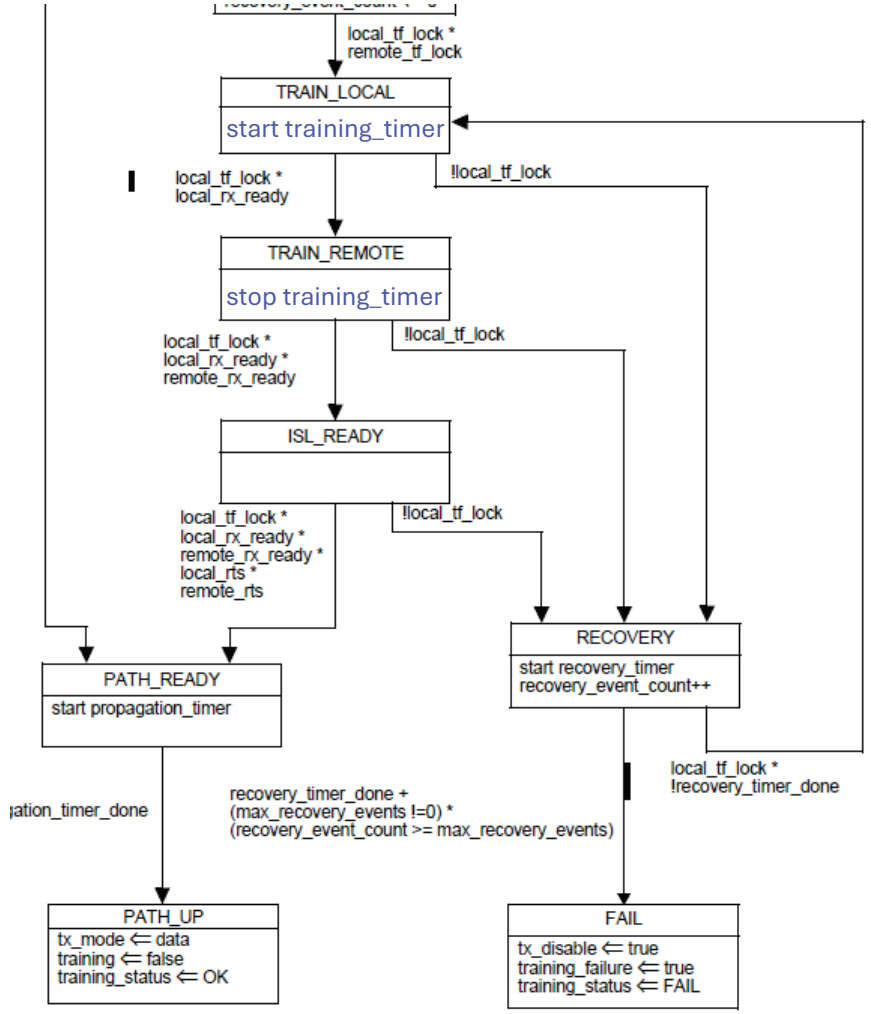

Figure 178B–8—Training control state diagram

Summary

- The proposed changes enable:
  - ILT without a specified timeout
  - AN that does not time out and restart when ILT is in progress
  - Existing behavior: AN is automatically restarted when an active link is broken
  - New feature: AN can be restarted while ILT is running (with management intervention)
  - Indication to management that the training phase of ILT takes longer than expected (configurable).
- Changes to the PCS clauses are mainly in the formal service interface definitions
- Minimal changes to the AN clause (only change is the definition of the link_status parameter)
- In practice, the behavior can be implemented in firmware – no new real-time logic is defined.

That’s all

Questions?