February 2020

METHOD TO IMPROVE LINK BRING-UP TIME FOR HIGH-SPEED ETHERNET BACKPLANE AND COPPER CABLES

Sreenivasa Rao Bandlamudi

Mouli Vytla

Anirban Roy

Richard Michaelraj

Recommended Citation

Bandlamudi, Sreenivasa Rao; Vytla, Mouli; Roy, Anirban; and Michaelraj, Richard, "METHOD TO IMPROVE LINK BRING-UP TIME FOR HIGH-SPEED ETHERNET BACKPLANE AND COPPER CABLES", Technical Disclosure Commons, (February 26, 2020)
https://www.tdcommons.org/dpubs_series/2977

METHOD TO IMPROVE LINK BRING-UP TIME FOR HIGH-SPEED ETHERNET BACKPLANE AND COPPER CABLES

AUTHORS:
Sreenivasa Rao Bandlamudi
Mouli Vytla
Anirban Roy
Richard Michaelraj

ABSTRACT

Modern Massively Scalable Datacenters (MSDCs) delivering high availability software such as Software-as-a-Service (SaaS), Infrastructure-as-a-Service, etc. have stringent metrics to which to adhere. Link availability is one such crucial factor. Techniques presented herein may facilitate decreasing link bring-up time by maintaining a cache for link training (LT) transmit (TX) Finite Impulse Response (FIR) settings. Thus, these techniques may provide for faster link bring-up, faster convergence, and faster fabric module bring-up in an End of Rack (EoR) chassis. Using techniques presented herein, faster link bring-up may become the norm rather than the exception.

DETAILED DESCRIPTION

Auto-negotiation (AN) and Link Training (LT) procedures are mandatory to exchange capabilities and adjust Finite Impulse Response (FIR) settings during the link bring-up sequence for an Ethernet backplane and copper cables. Generally, the LT procedure takes time for a pulse amplitude modulation (PAM) based signal, which results in increased link bring-up time for interface shut/no-shut or micro-flap due to hardware faults and/or physical medium errors. Link UP delay could lead to traffic loss.

Due to the emergence of 5th Generation (5G) technologies, there is a high demand to increase interface speeds to 200Gigabit (G)/400G from 40G/100G. Critical data carried over links used in industries such as healthcare, financial trading, etc. warrants a significant reduction in link bring-up (UP) time when a link flap occurs, which in turn improves the convergence of traffic.

Techniques proposed herein provide a mechanism to reduce link UP time for PAM based 400G/200G speeds. Because copper cables are widely used because of low cost within a Massively Scalable Datacenter (MSDC), techniques herein provide an innovation to bring down link UP time for copper cables and Fabric Interfaces. Faster link UP time benefits routing protocol convergence, a significant reduction in traffic loss, increased bandwidth, and better load balancing. Other areas may benefit from this innovation including practical use-cases such as:

- Micro Flaps between links
- Module Online Insertion and Removal (OIR)
- Peer Line Card Reloads
- Blade servers Reload with Fabric Extenders
- Fast link bring-up between Fabric card and line card modules in EOR/Top of Rack (TOR) chassis

Per Institute of Electrical and Electronics Engineers (IEEE) standards, AN and LT procedures are mandatory for 50G/200G/400G copper direct attached cables (DAC) and backplane connections. Auto-negotiation helps devices to exchange capabilities using the Base page and Next page information. Once both devices exchange Base pages, AN is resolved to highest common supported features such as speed, duplex, and pause. After a successful AN page exchange, the LT process is started to acquire equalization parameters. The AN process may be restarted if a link is not brought up within a predefined link timer expiry. The LT timer for non-return-to-zero (NRZ) modulation based 25G/100G interfaces is ~450 milliseconds (msec) and the LT timer for PAM based 50G/200G/400G interfaces is ~3.25sec. For PAM based interfaces, the LT procedure consists of multiple transmit (TX) Finite Impulse Response (FIR) table walks, which could lead to higher link UP time, as much as 5 to 15 seconds.

This proposal provides techniques to optimize the AN and LT procedures to exchange capabilities and adjust TX FIR settings during link bring-up time. In particular, the techniques involve caching FIR settings learned during LT once a link is up. Since the LT procedure typically takes more time to adjust FIR settings, cached values can be used for subsequent link bring-up scenarios such as a micro flap, copper module OIR, line card (LC) reload, and Fabric card reload. This innovation may be applicable to both copper

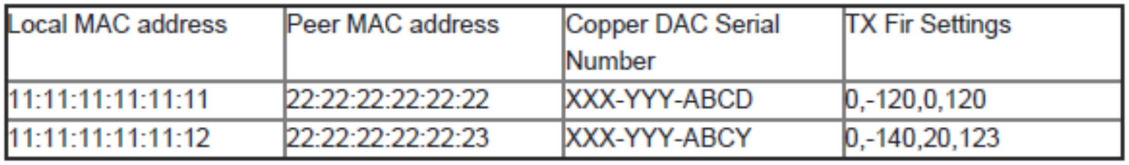
DAC and backplane connections; however, the logic to maintain the cache for TX FIR settings may be different for copper DAC and backplane implementations.

An example workflow for learning FIR settings for copper DAC may include:

1. The first time, AN and LT procedures are performed during link UP and FIR settings are cached.
2. A user may move a copper module from one port to another port on the same device or different device. This scenario can be detected and a determination can be made that a cached value is invalid. Thus, both the AN and LT procedures can be performed again to bring-up the link in which the cache is updated for the subsequent link bring-up scenario.
3. The AN next page exchange capability will be used to exchange local port Media Access Control (MAC) address. Data maintained in a given Copper DAC Cache table may be provided as:

```
Copper DAC Cache table elements:
{
char Module_Serial_Number[50];
unsigned char Peer_MAC_Address[6];
unsigned char Local_MAC_Address[6];
unsigned int LT_FIR_Pre_Main_Post[X]; /* X is hardware specific */
};
```

An example Copper DAC Cache table is shown below in Figure 1.



Local MAC address	Peer MAC address	Copper DAC Serial Number	TX Fir Settings
11:11:11:11:11:11	22:22:22:22:22:22	XXX-YYY-ABCD	0,-120,0,120
11:11:11:11:11:12	22:22:22:22:22:23	XXX-YYY-ABCY	0,-140,20,123

Figure 1

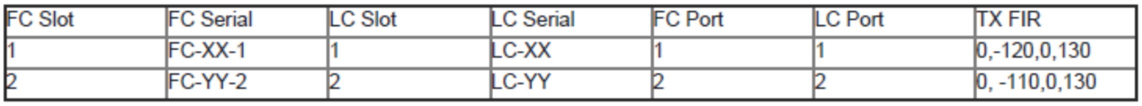
4. To detect copper module movement from one port to another port, the Cache table can be searched for a matching peer MAC address. If a cache hit is determined, the LT procedure is skipped and cached values of TX FIR settings are used to bring-up the link. However, in the case of a cache miss, the LT procedure is performed again.

An example workflow for learning FIR settings for a backplane may include, for the link between a Fabric Card (FC) and Line Card (LC) or a Blade server and Fabric expander, slot numbers can be used to maintain a cache for TX FIR settings. For example, cached values can be used in LC or FC (LC/FC) reload scenarios. A Cache table will be invalidated for the removal of LC/FC. Techniques of this proposal, however, may help in reducing link bring-up time for LC/FC reloads.

Data maintained in a given Backplane Cache table may be provided as:

```
Backplane Cache table elements:
{
char FC_serial_Number[x];
unsigned int FC_slot;
char LC_serial_Number[x];
unsigned int LC_slot;
unsigned int LT_FIR_Pre_Main_Post[X]; /* X is hardware specific */
};
```

An example Backplane Cache table is shown below in Figure 2.



FC Slot	FC Serial	LC Slot	LC Serial	FC Port	LC Port	TX FIR
1	FC-XX-1	1	LC-XX	1	1	0,-120,0,130
2	FC-YY-2	2	LC-YY	2	2	0, -110,0,130

Figure 2

Figures 3 and 4, below, illustrate differences between the sequence/improvements in link bring-up time that may be provided by utilizing an LT cache as proposed herein.

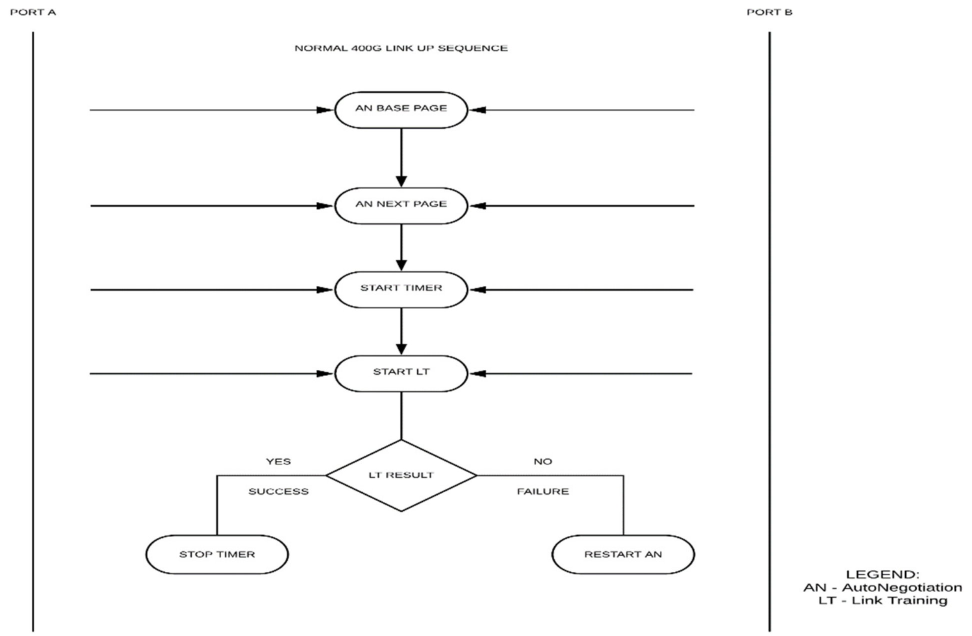


Figure 3: Normal Link UP Sequence with AN and LT procedures

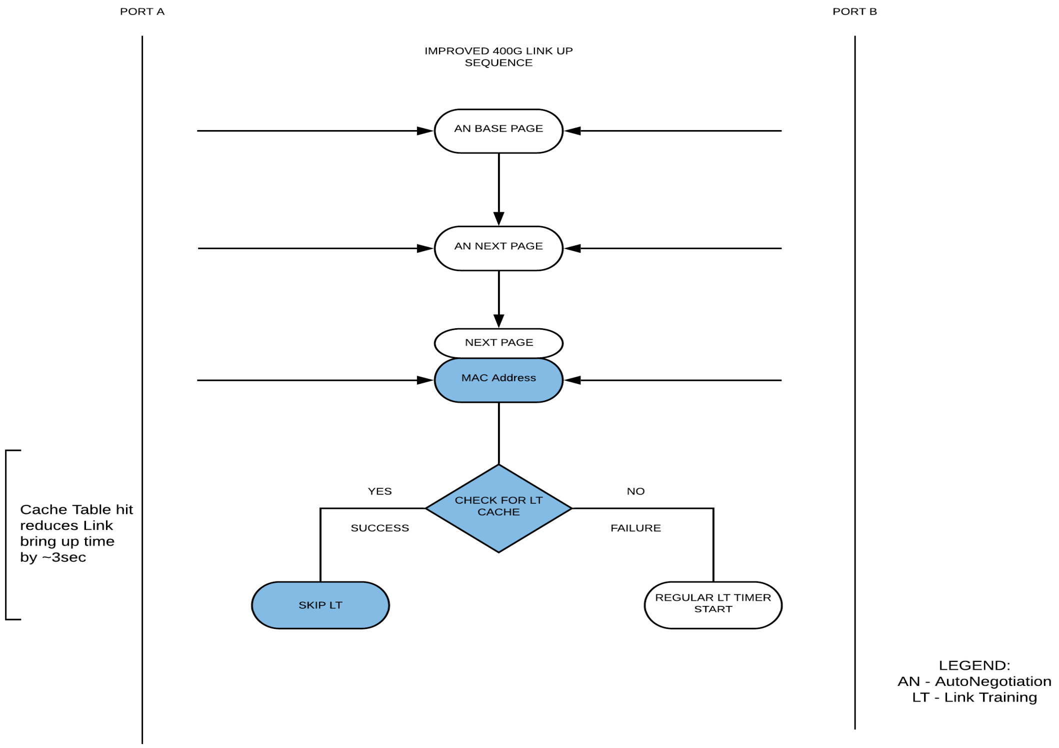


Figure 4: Improved Link UP Sequence With AN and Without LT

To explain the improvements between Figures 3 and 4, for simplicity, imagine that 'T' is the total time taken for the link to come up. During a normal link bring-up, the LT procedure may take several iterations before two ports adjust FIR settings, as shown below in Figure 5 involving a PORT A and a PORT B.

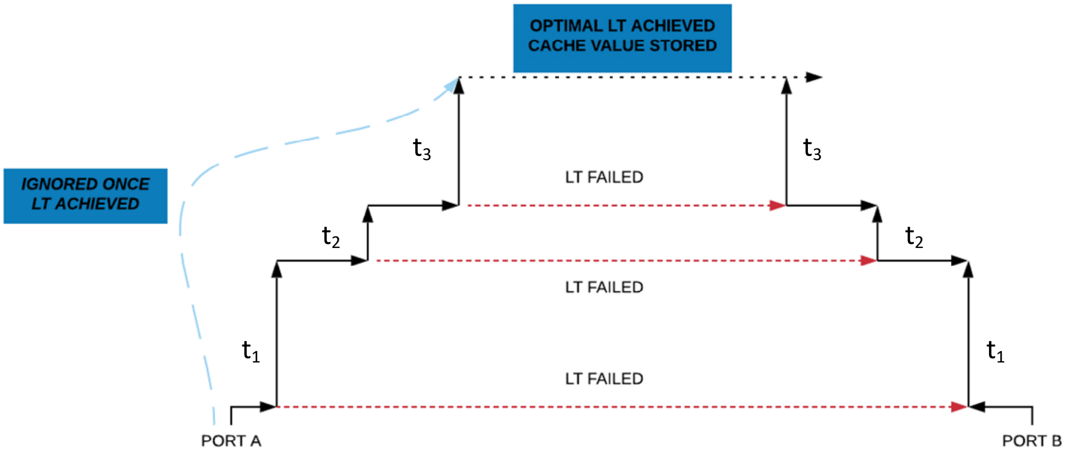


Figure 5

For Figure 5, consider that t_1, t_2, and t_3 are different timestamps in which LT attempts are made between PORT A and PORT B, which on average may range from 5 seconds to 15 seconds (e.g., $t_1 + t_2 + t_3 + . . . + t_n = 10$ to 15 seconds).

Consider that $T_X = t_{AN} + t_{LT}(n)$ in which T_X is the total time for a new link bring-up, t_{AN} is the total time taken for auto-negotiation, t_{LT} is the time taken for link training, and 'n' is the number of LT iterations. For a next link bring-up, techniques of this proposal provide for restoring the Cache value as a constant 'C' that is used to bring-up the link, which is faster due to missing the $t_1 + t_2 + t_3 + . . . + t_n$ attempts for link training; thus, $C=T_X$. Further, consider that T is the total time for the link bring-up with the Cache value in which $T = t_{AN} + C$ in which C is the constant Cache value taken during the first link bring-up. Since C is determined in the first attempt, link training is much faster. Accordingly, techniques herein can facilitate faster link bring-up times by maintaining a cache for LT TX FIR settings.