P802.3ct FEC Frame Alignment

Steve Trowbridge (Nokia)

Clause 153 Editor

Comparison – Ethernet and OTN Frame Alignment

- Clause 82 100GBASE-R PCS:
  - Striped over 20 PCS lanes. Lanes are marked by a 66-bit alignment marker (SH=10, 64 bits with a unique pattern for each of the 20 lanes) every 16K 66B blocks on each lane (1 marker each 209.725µs)
  - When in 66B format without FEC, should be running over a PHY that delivers 10^{-12} BER
  - Ethernet has a SIGNAL_OK parameter that indicates that as far as the PCS knows, data is being properly recovered. Implementations are not required to squelch potentially bad data being passed, however SIGNAL_OK is FALSE when lanes are out of lock or out of alignment
- OTU4/OTL SC-FEC frame:
  - Striped over 20 OTL lanes. Lanes are marked by a frame alignment signal consisting of 5 fixed bytes (common on all lanes) and one byte that counts 0-239. The final byte modulo 20 is the lane number (this is also used to expand the skew detection range). The framing pattern is repeated once per 16320 byte frame on each lane, so one framing pattern per 23.354µs
  - Running over a PHY that delivers 4.62×10^{-3} BER. The lanes need to be identified, deskewed, and re-interleaved before FEC decode.
  - When lanes are out of lock (losing the framing pattern) or out of recovery (losing the lane ID), the last known lane ID and frame start position are “remembered” and data continues to be re-interleaved and passed up the stack without any indication to higher layers of the loss of lock or recovery state. An alarm is raised if those states persist for 3ms (~128 frames)

Goals of the process

- Probability of false lock should be sufficiently low
- Probability of false loss of lock should be sufficiently low

What does the ITU-T defined process do?

- Separate process for recovering fixed framing pattern and for identifying the lane number
- Process for recovering the fixed framing pattern is similar to Figure 153-7. Going from out-of-frame to in-frame occurs when the fixed (5 byte framing pattern) is found a second time the expected position in the next frame period. Going from out of frame to in frame occurs when you don’t find the fixed framing pattern in the expected location for five consecutive occurrences
- Gaining lock on the lane number occurs when the lane ID modulo 20 has the value for 5 consecutive frames. Losing lock on the lane ID occurs when the lane ID modulo 20 does not have the expected value for 5 consecutive frames
- Match on the fixed bytes is specified as a match of 4 of the 5 octets. What is probably intended is that they should be considered a match if any 4 of the 5 octets match (and this presentation will assume that in subsequent calculations), but the wording could be interpreted that this is just implementation flexibility (the implementer can choose which 4 of the 5 octets to check)
- When in an out of frame state, the frame start position is maintained and interleaving continues based on the previously discovered frame alignment position (there is no inband SIGNAL_OK kind of indication as in Ethernet that indicates a confidence that good data is being sent to the layer above)
- An out of frame defect is declared when the out of frame state persists for 3ms.

Probability of False Acquisition of Frame

- ITU and the P802.3ct Draft 2.0 lock on two occurrences of the frame alignment pattern the expected distance apart.
- Assuming we might separate the processes and are only looking at the five fixed octets: Given the OTN scrambler, with the 66B self-synchronous scrambler nested inside of it, it is essentially impossible to deliberately spoof the framing pattern with something you send inside of the data.
- The probability of matching all five, or four out of five octets of the fixed octets of the FAS matching on random data is:

$$2^{-40} + 2^{-32} \times \frac{255}{256} \times 5 = 1.16052 \times 10^{-9}$$

- So that gives a probability of two “four out of five” matches on random data happening exactly 16320 octets apart being 1.3468×10^{-18}, which at 42819 frames per second means false frame alignment will occur once per 549K years. Probably fine if you have to then lose and reacquire lock if it doesn’t happen more often than this.

Probability of false loss of lock

- This occurs when the Rx is properly locked onto a signal, but due to bit-errors, the framing pattern isn't found in the expected location for five consecutive frame periods (for out of frame in both ITU-T and in the P802.3ct draft)
- BER is 4.62×10^{-3}
- Probability of an error in an octet is $1-(1-4.62\times10^{-3})^8=3.6368\times10^{-2}$
- Probability of more than one octet wrong:

$$1 - P(0\ octets\ wrong) - P(1\ octet\ wrong) =$$
$$1-(1-3.368\times10^{-2})^5-5\times3.368\times10^{-2}\times(1-3.368\times10^{-2})^4=$$
$$1.229\times10^{-2}$$

- More than one octet wrong on five consecutive frames will occur about once per 23 hours at the indicated BER
- The ITU-T 3ms timer before declaring an OOF defect would check that the OOF state is maintained for ~128 frames without encountering two in a row "four of five" octet matches.
- This is probably too long and too complex for Ethernet. If we change loss of lock to not getting a "four of five" match for ten occurrences, false loss of lock occurs once per 9.412 million years and only extends the window to 241µs

Lane Identifier

- A single octet lane identifier is allocated as the 6^{th} octet of the frame alignment signal.
- There are 20 lanes. The identifier cycles in a count from 0-239, with the lane identifier being the value modulo 20. The use of the counter provides extra skew detection range.
- If there are errors that affect the lane identifier, it may indicate the wrong lane. Without redundancy, there is higher probability that the lane identifier indicates the wrong lane than that you get a 4 of 5 octet match on the basic framing pattern.
- The following analysis overestimates the probability of false lock on a lane number, as getting two consecutive matches on the <u>same</u> wrong lane number would require multiple bit errors (at least in the second lane number) which is lower probability than single bit errors. But to simplify the math, these calculations assume if a byte is wrong, all 255 wrong values are equally likely.
- Of the 256 possible values for the lane identifier octet, there are 12 values each that represent each of the 20 possible lane numbers, and 16 values that do not represent any lane number
- Two ways to compare subsequent frame lane identifiers:
  - Permissive: Each subsequent frame lane identifier has one of the 12 values that represents the same lane number as the identifier in the original frame. So the current lane identifier modulo 20 equals the original lane identifier modulo 20 (12/256 chance on random data) (ITU uses this method)
  - Restrictive: Each subsequent frame lane identifier has exactly the value expected given lane identifier sequencing: i.e., (the previous frame lane identifier plus 20) modulo 240 (1/256 chance on random data)

ITU Lane Identifier Process

- In recovery/out of recovery states
- Once in recovery, lane identifier and frame start positions are maintained until a different lane identifier is recovered (losing lock doesn’t squelch the signal)
- Carried out after lock acquired on the 5 fixed octets (matching all 5 or 4 of the 5 fixed octets)
- Move to in recovery (IR) state after 5 consecutive “permissive” matches on the lane number (the lane ID modulo 20 has the same value)
- Move to out of recovery (OOR) state after 5 consecutive values modulo 20 have the wrong value

Analysis of Lane ID recovery – ITU-T algorithm

- Not completely accurate math, as more than one bit error in the lane ID has lower probability than a single bit error (although bursts could occur). But for the purpose of these calculations, assume if the byte is received with the wrong value, any of the 255 incorrect values are equally probable.
- 9 of 255 values are the incorrect value for the byte, but represent the correct lane number modulo 20.
- Match/correct probability therefore is:

$$(1 - P_{Be}) + P_{Be}\frac{9}{255} = 0.96491$$

- Probability of 5 consecutive frames indicating the correct Lane ID = 0.86344 (86.3% chance you get it right on the first try). Mean time to recover Lane ID is 5 frame times divided by this probability, or 5.9777 frame times.

Loss of Lane ID – ITU-T algorithm

- Loss of Lane ID occurs after 5 consecutive values modulo 20 do not indicate the locked lane ID
- Assume we are clever enough to ignore errored lane ID values that do not correspond to any valid lane number (values 240-255). So we can assume a wrong value when it is not one of these values, and the value modulo 20 is not the expected value (190 out of 255 possibilities when there is an error in the lane ID)
- Probability of 5 consecutive lane ID wrong:

$$(P_{Be}\frac{190}{255})^5 = 1.46102 \times 10^{-8}$$

- False loss of lock once per 2.2 hours
- Increase to 10 frames with lane ID wrong – 17335 years
- Increase to 12 frames with lane ID wrong – 23.6 million years

General Observations – ITU-T method

- ITU-T loss of lock or loss of recovery occur way too often to be considered a loss of alignment for an 802.3 PHY, given that a consequence would be SIGNAL_OK=FALSE when alignment has been lost
- The only thing that “saves” the ITU-T behavior is that nothing is done when losing lock or alignment unless it persists for 3ms
- Assuming that P802.3ct selects a frame alignment process that has sufficiently low probability of false lock or false loss of lock, this does not create any obstacle to interconnection between a P802.3ct implementation and a “not quite 802.3” implementation that uses the ITU-T frame alignment method and does not have a SIGNAL_OK=FALSE indication when out of lock or out of recovery

What if we lock on the lane ID together with the fixed bytes of the FAS?

- Since the probability of the fixed bytes of the FAS being mismatched in two occurrences is so low, we don’t need to worry about the probability of incorrectly matching the lane ID in the wrong place (i.e., on random data). Also a bit meaningless to look at how often we see the wrong lane ID, rather than a probability we capture an incorrect lane ID when we bring the link up.
- Restrictive match – we consider the first lane ID to be wrong (but accepted) when the lane ID octet contains any of the 239/255 values that is different from the transmitted value but is a valid lane number (even if the lane number is accidentally correct), and the second frame lane ID octet exactly matches the incorrect first frame octet plus 20 modulo 240 (1/255 of the possible incorrect values)
- Permissive match – we consider the first lane ID to be wrong (but accepted) when the lane ID octet contains any of the 221/255 values that represents a valid lane number that is different from the transmitted lane number (so if the octet is errored but by accident translates to the correct lane number, we consider it OK). The lane ID in the second frame has one of the 12/255 values that has the same modulo 20 wrong lane number as the first frame. (ITU-T uses the permissive match method, although numbers are better with the restrictive match)

Probability of capturing the wrong lane ID if done together with the FAS

- Using the restrictive match: 4.8613×10^{-6}
- Using permissive match: 5.3942×10^{-5}
- Consequence of capturing the wrong lane ID is that you lose frame lock, and then re-acquire lock.
- If you lock separately on the lane marker rather than having discrete lane marker “capture” opportunities, you can look independently at false lock or false loss of lock.
- Probably not too bad

False Lock probability together with Lane ID

- Occurs when the following are true:
  - You get two “4 of 5” matches on the fixed octets a frame length apart.
  - The first octet following the framing pattern in the first frame is one of the 240 of 256 values that represents a valid lane ID
  - The first octet following framing pattern in the second frame is (restrictive match) the lane ID recovered from the first frame plus 20 modulo 240 (1 of 256 probability), or 4.93211×10^{-21}
- False lock once per 300 million years

What about loss of lock when done together with the fixed bytes?

- Framing pattern mismatch when either you fail to get a “4 of 5” octet match on the fixed bytes, or the lane ID is wrong (restrictive match, so any error produces a mismatch).

$$1.2290\times10^{-2}+3.6368\times10^{-2} = 4.8658\times10^{-2}$$

Loss of Lock Persistency (Frames)	Average time to False Loss of Lock
5	428 seconds
10	99.5 years
12	50K years
15 (~280µs)	547 million years

Consequence of false loss of lock

- Doesn't need to be considered in the same "bucket" as MTTFPA, since this doesn't deliver bad data to the MAC – it brings the link down
- The link will be down until it re-frames, which could be as short as two frame times (46.7μs), or longer, depending on exactly how the implementation "hunts" for a valid framing pattern (unlikely you actually shift by one bit per frame period until you find it). You could actually take 3.05 seconds to find frame alignment if you shifted by 1 bit per frame period of 130560 bits.
- Time to reframe after false loss of lock is a contributor to the frame loss ratio, which converts to BER. FLR equivalent of 10^{-12} BER is 6.2×10^{-10}
- 3.05 seconds in 50K years (12 frame loss of lock) is 1.92×10^{-12}. 3.05 seconds in 547 million years (15 frame loss of lock) is 1.767×10^{-16}. Therefore, this is not a significant contributor to FLR or the equivalent BER

Proposal

- Frame Lock and Loss of Lock should be done with the fixed bytes of the FAS plus the lane ID together
- Frame lock occurs when a “4 of 5” match on the fixed octets occurs two consecutive times 16320 octets apart, the Lane ID octet following the first FAS is a valid lane ID, and the Lane ID octet following the second FAS equals the lane ID octet following the first FAS plus 20 modulo 240.
- Loss of lock occurs when failing a “4 of 5” match or the wrong value for the lane ID octet (calculated by adding 20 modulo 240 for the expected value in the next frame – restrictive match) occurs for 15 consecutive frames
- SIGNAL_OK is TRUE when all lanes are locked and deskewed, FALSE otherwise
- (optional) Do we need a note explaining that an interface using the frame alignment process of G.798 can interconnect with an interface using the frame alignment process described in P802.3ct?

Next Steps

- If there is general consensus on the approach, I can, as editor, prepare a specific text proposal describing this method and submit a single comment against Draft 2.0 to incorporate this improvement

THANKS!