RoE agnostic mapper and Start of Frame

Jouni Korhonen
Broadcom Ltd.
5/30/2016

Background

- The RoE header either carries a sequence number or a timestamp in the orderingInfo field.
  - Once the flow has been setup the orderingInfo is always treated in the same way..
  - Different types do not alternate.

- There are cases where correlating (internal) sequence numbers and timestamps is beneficial – applies to the mode where orderingInfo is a timestamp.

Combined seqnum and timestamp

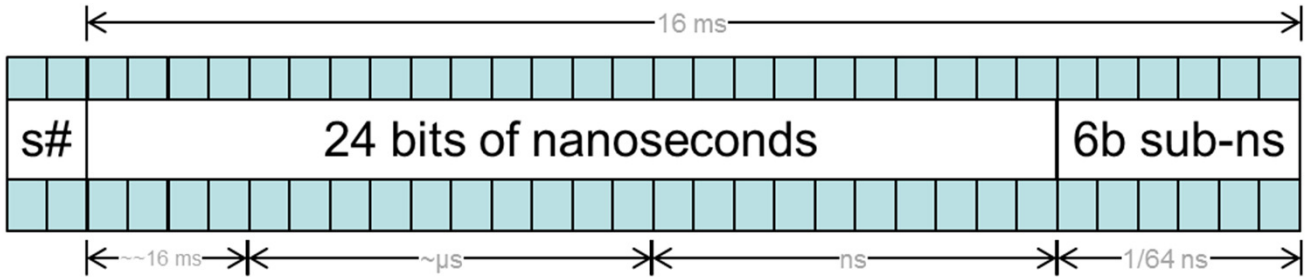
- Referring back to presentation in San Jose April 2016 meeting: tf3_1604_tse_use_cases_1.pdf

- The Case #1 (CPRI trunking case) needs non-trivial clock domain crossing and clock recovery:
  - The structure agnostic CPRI mapping and transport over an RoE transport network is one of the prominent use cases. Timestamps are needed for asynchronous CPRI clock recovery and "the start of frame" event/indication is needed to easily find & align the CPRI data.

- Related to the above and having both timestamp and sequence number present at the same time in a data packet (see tf3_1604_tse_datapath_1.pdf) the consensus was reached as **'2:24:6' timestamp** -> See next slide for more information.

Current baseline

- RoE '2:24:6' timestamp was recently (see tf3_1604_bross_timestamp_4.pdf) approved as:



- Also stating:
  - Benefits of this timestamp
    - 16 ms range covers 1 radio frame
    - Precision down to ~16 ps accuracy (1/64 ns)
  - 2-bit sequence number at top allows detection of up to 3 missed packets **<- "low p-counter bits"**
- This is fine... except.. there are issues left.

2 bit sequence number

- 2 bit sequence number needs to be mapped to an internal sequence number that is long enough to cover at least the RoE jitter buffer - preferably more than the radio frame.
- 2 bit sequence number could be a modulo 4 of the internal sequence number on the sender and modulo 4 of the next expected sequence number on the receiver.
- Works but:
  - Initial synchronization to start of frame is challenging.
  - Detecting and recovering from lost sync requires converting timestamps to sequence numbers (of internal size) and tracking them all the time -> not much use of the 2 bit sequence number.
- Timestamp to sequence number conversion itself is an approximation of a time range around the timestamp, not an exact time.

24:6 timestamp (and CPRI examples)

- Represents a presentation time calculated in the RoE clock domain:
  - Calculated by the sender based on an known (measured) offset and a sync event received e.g., from a CPRI stream. The sender gets a sync event from CPRI of the start of each hyper frame.
  - Until the next hyper frame event arrives the timestamps in between can e.g., be interpolated.
- Works but:
  - Detecting and synchronizing to a proper "start of frame" is not as easy as it looks like – Detection based on timestamp is an approximation.
  - 1/64ns precision is somewhat overkill with the current state of art in Ethernet networks.

What if one uses control packets?

- Current RoE specification has a control packet to combine a sequence number and a timestamp.
- If used together with the '2:24:6' timestamp it would mean:
  - Sending the binding ahead of time and still have the same clock recovery & approximation issues as a plain '2:24:6'.
  - Forces an RoE switch implementation to have a powerful CPU in the solution, which cannot be assumed to be there.
  - communicating between different, possibly distributed, units within the RoE system. A control plane CPU doesn't need to be part of the RoE switch ⇒ gets complicated and increases internal traffic.
  - Sending control packet frequently – like once a hyperframe for each RoE flow ⇒ heavy CPU load and increases non-data RoE traffic.
  - Each port and flow has to be handled independently.

What if 'start of frame' were explicit?

- Requires an explicit marker bit for a "start of frame" in the RoE header.
  - This is a "major synchronization point".
- The start of frame marker would basically tell two things to the receiver:
  - The 2 bit sequence number and the internal sequence number mean the start of frame e.g., a specific pattern to detect/to reset to.
  - This timestamp equals to the "start of frame".
- Requires minimal state and no time based RoE flow tracking at the receiver.

Quick analysis of 'start of frame'

- Benefits?
  - No cumbersome timestamp tracking and adjustment for finding "start of frame".
  - No need to convert timestamps to sequence numbers.
  - Easy recovery after a sequence number sync loss.
  - Less state to maintain – at both ends.
  - Simple implementation.
  - Does not mandate a powerful control plane CPU.
- Cons?
  - None ;)
  - ..well we have no available bits in the RoE header unless we repurpose some existing bits/fields.

Explicit 'start of frame' indication

- Not needed when orderingInfo is in sequence number mode -> specific sequence number patterns do the same!

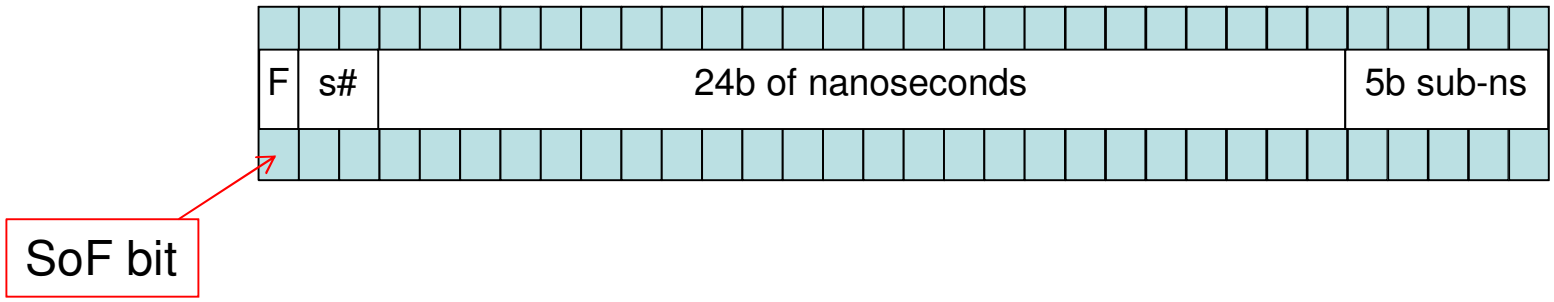
- Applies only to the "2:24:6" timestamp.

- Since 1/64ns is hard to implement and with the foreseen synchronization accuracy at the RoE nodes it does not make real difference ⇒ repurpose one 1/64ns bit for an explicit start of frame indication.

Proposal

- Proposal: a '**1**:2:24:5' timestamp.

- The F (SoF) bit is set for a packet that contains a start of radio frame in it.



- Has 2 bit sequence number
- 24ns of timestamp -> ~16.8ms window
- Still 1/32ns precision -> 31.25ps precision

Clarifications

- What qualifies as a "start of frame"?
  - Depends on the transported/mapped radio framing synchronization.
  - Could be the start of a radio frame. In a case of CPRI could be a start of a hyper frame.
  - The RoE does not care as long as it is periodic.

- Payload alignment with the "start of frame" indication:
  - For the agnostic mapper to work properly with the SoF bit/indication, the first payload octet in the RoE packet has to be the one that triggered the SoF event and also represent the presentation time & 2 bit sequence number.
  - Future specification can define alternate locations.

Discussion..