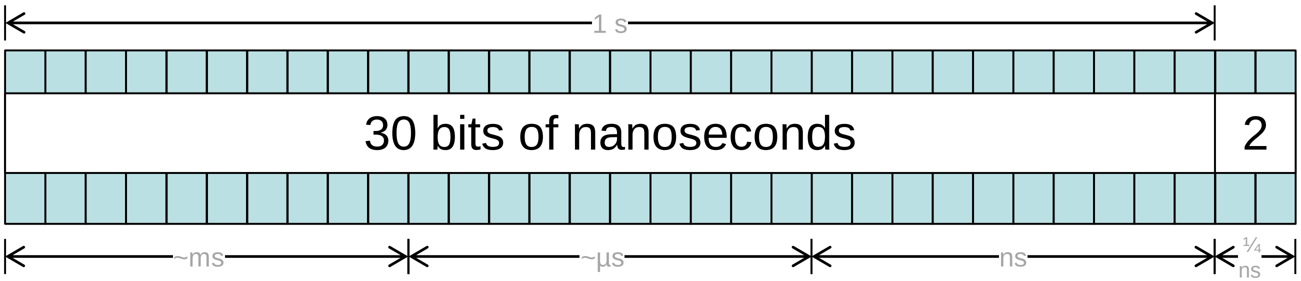
Timestamp Precision

Kevin Bross

26 April 2016

Timestamp Packets

- The *orderInfo* field can be used as a 32-bit timestamp, down to ¼ ns granularity:



- Two main uses of timestamp:
  - Indicating start or end time of flow
  - Indicating presentation time of packets for flows with non-constant data rates

s = second
ms = millisecond
µs = microsecond
ns = nanosecond

Presentation Time

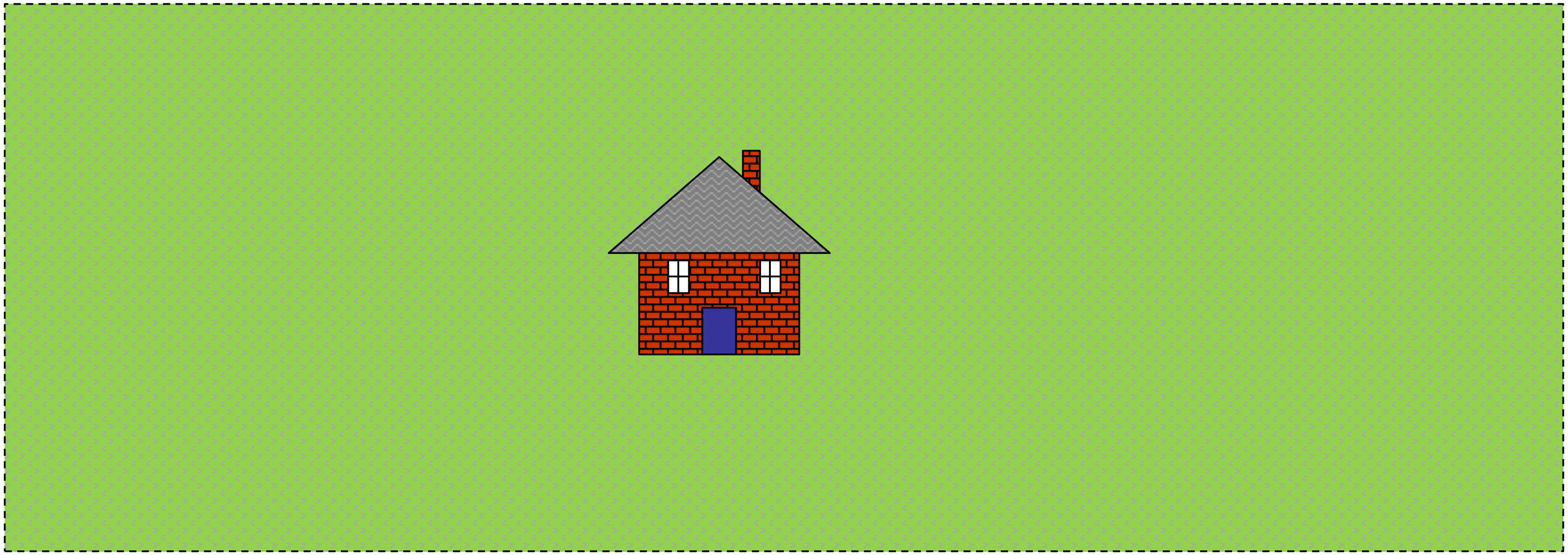
- To reduce bandwidth during idle periods, some protocols will have variable rates
  - Fronthaul may be variable, even if rate to radio unit itself is a constant rate
- Presentation times allows RoE to handle variable data rates
  - Data may experience jitter in network
  - Egress buffer compensates for network jitter
  - Presentation time is when the data is to exit the RoE node
    - Jitter cleaners ensure data comes out cleanly, and on the right bit period

Jitter vs. Synchronization

- Synchronization requirements for LTE are *only* down to ~±65 ns accuracy
  - Each RoE node may be off from TAI by up to 65 ns (or more in some circumstances)
  - Starting and ending a stream may be off by this amount
- …but jitter from packet to packet must be much tighter
  - RoE nodes should be able to output data at precise relative times if timestamp is used for a given packet
  - Relative bit time within a flow is important

Farmhouse Analogy

- Absolute location of farmhouse may be ±several meters from what is envisioned
- Dimensions within the farmhouse need to be accurate to sub-centimeter dimensions
  - Don’t want windows to leak, etc.



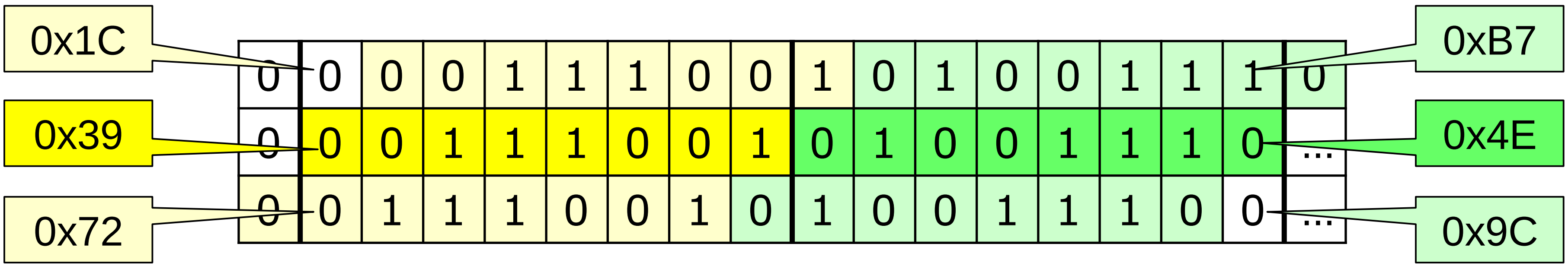
Timestamp Precision

- Is the current ¼ ns granularity tight enough for today's systems, and does it have headroom for the future?
  - Each bit in 9.8 Gbps CPRI is ~1/10 ns
  - Each bit in 24 Gbps CPRI is ~1/24 ns
  - Rates of 100 Gbps or more are likely in the reasonable future

- How do you specify a presentation time with bit times that may be tiny fractions of nanoseconds if the smallest unit is in ¼ nanoseconds?

Hypothetical Example

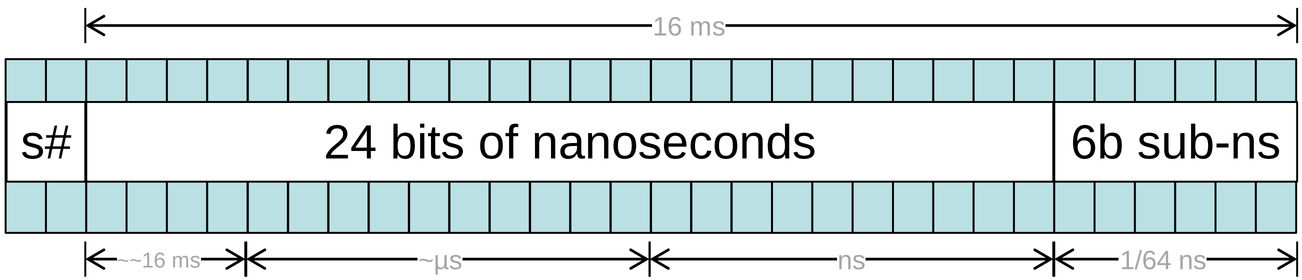
- Assume 100 Gbps raw data rate, with extended idle periods suppressed
- Raw data: …, 0x3F, 0x4E, <807 bytes of 0’s>, 0x39, 0x4E, …



- How does RoE say when that packet is supposed to hit (first byte = 0x39)?
  - One bit position late, first byte = 0x1C
  - One bit position early, first byte = 0x72
- Relative timing of bits is important

Alternate Proposal

- Combine sequence # and timestamp
  - Can cover more than 10 ms LTE radio frame

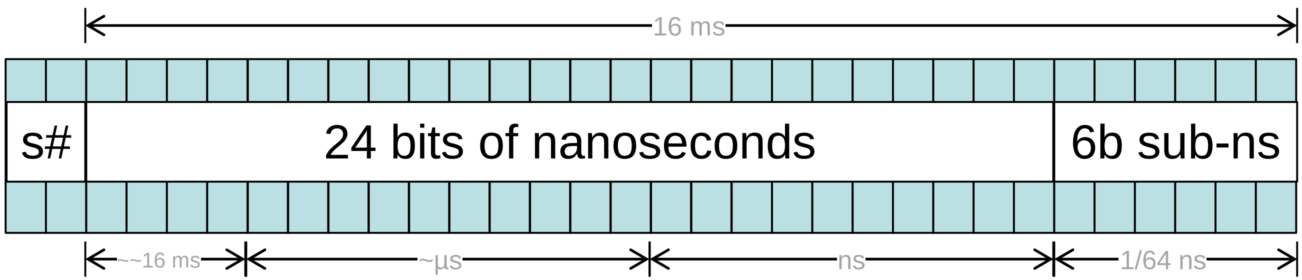


  - This handles presentation times up to 16 ms in the future, while offering precision to ~16 ps
    - Would handle data rates up to ~63 Gbps
  - Timestamp purpose/usage unchanged
- Sequence field could detect up to 3 lost packets

ms = millisecond
µs = microsecond
ns = nanosecond
ps = picosecond

Summary

- Redefine timestamp to provide higher precision, in 1/64 ns increments



- 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

ms = millisecond
µs = microsecond
ns = nanosecond
ps = picosecond