Residential Ethernet:

Time-of-day timer synchronization

Maintained by David V James

July 2005 IEEE 802.3 RE Study Group San Francisco 1

This is an RE slide set, with many slides created by DVJ.

RE stands for “Residential Ethernet”, and 802.1 study group.

Alternative names abound, which mean the same thing:
  Residential Bridges, Residential Bridging
  AV Bridges, Audio/Visual Bridges, etc.

Credit is due to many others, whose reviews/comments evolved this concept.

RE-SG basic requirements

- Cheap and simple
  - Delayed snapshot processing
  - Periodic symmetric transmissions
  - Etc., etc.
- Cheap and precise
  - Limited to snapshot capture accuracy
- Cheap and robust
  - Single-phase grand-master selection

The main consideration is to be cheap:
  Easily integrated into consumer bridges.
  Easily integrated into consumer end-points.

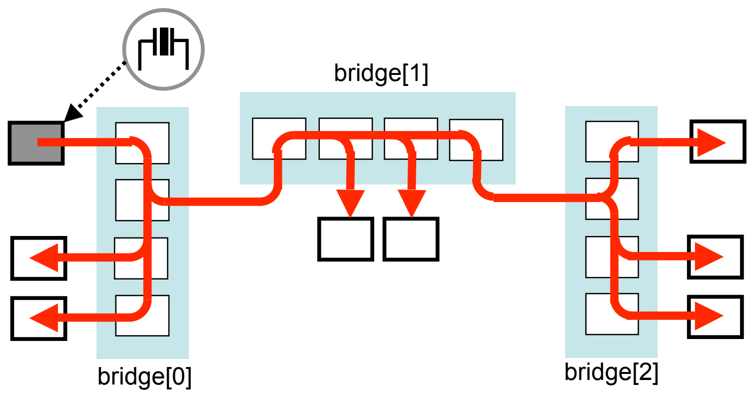
Simplicity is valued, since fewer mistakes are likely to be made.

Precision is valued, since audio-files are very sensitive
  ps jitter desired, although theoretical and filtered is probably OK

Robustness is also critical, with minimal-glitch handovers

Cascaded TOD synchronization

Wall-clock distribution model



The clock-distribution scheme is that of IEEE 1588.

The grand clock master (grand master) is the station whose time-of-day clock is the reference.

From a logical perspective, the clock master broadcasts the current time-of-day to the attached stations.

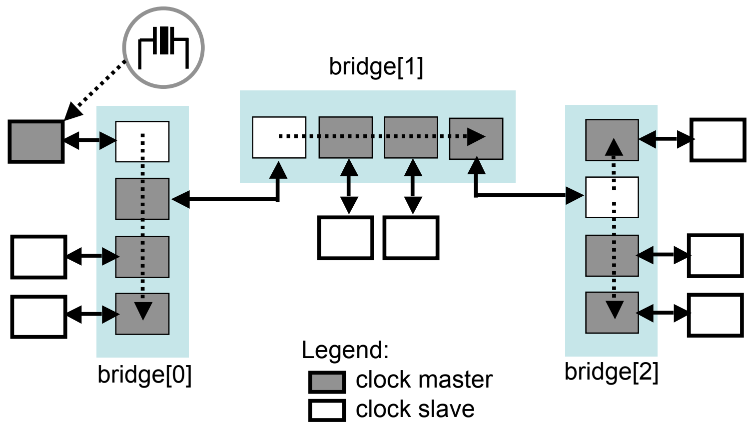
From a physical perspective, such a multicast time distribution would be inaccurate:

1) There may be source transmission delays.
2) There may be bridge forwarding delays.

Therefore, a more-precise synchronization protocols is used.

Cascaded TOD synchronization

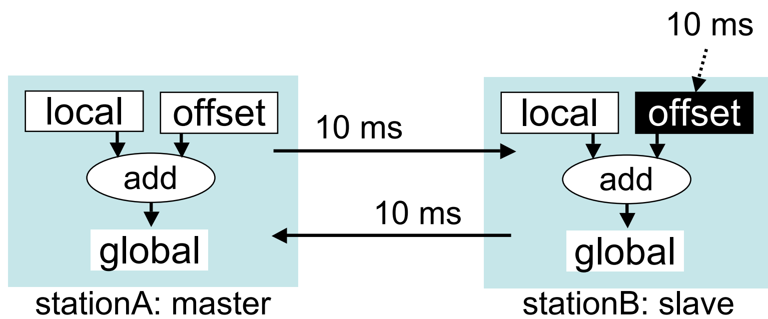
Cascaded adjacent-synchronization hierarchy



To avoid propagation-time inaccuracies:
  Synchronization is done on a point-to-point basis
  Internal bridge distribution (port-to-port) is “magical” and beyond our scope.

Adjacent-station synchronization

Offset value adjustments



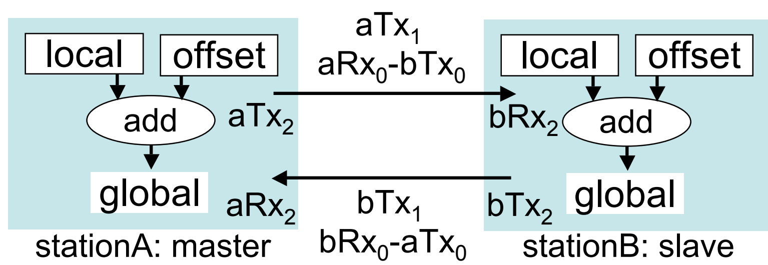
Initial studies indicate a faster update rate is practical.

While 1ms is practical, a slower 10ms allows the cheapest of microprocessors.

The primary purposes of these transfers are for:
Grand-master selection (reduces rogue frame stabilization times)
Offset adjustments, to force current time acceptance
(For precise synchronization, rate adjustments may also be required.)

Adjacent-station synchronization

Snapshot value distribution



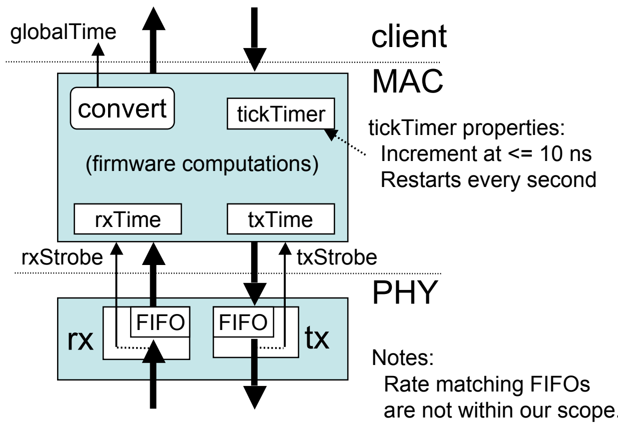
Time snapshots are best sent in the next “cycle”.

- Cheap: easily implemented in hardware (and possibly firmware).
- Precise: observations are more precise than predictions

In the case of stationB, the aTx and aRx values must be sent, since these were measured by stationA and are not known to stationB.

The value of bTx is (in concept) known to stationB and need not be transmitted. However, for simplicity, transmission of this value allows it to be more easily affiliated with the same-cycle indexed aRx value.

Minimalist HW design model



What is the hardware design model?

Simple hardware to snapshot the arrival/departure times

The local time reference (a minimum frequency is sufficient) is OK.

External communications are through normalized time values.

Firmware performs the conversions, frame formatting, etc.

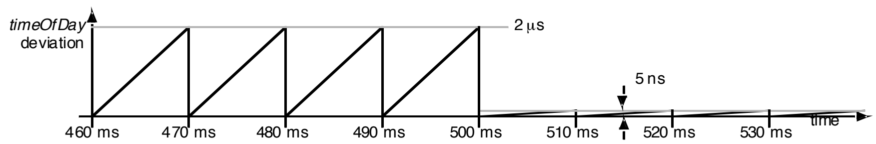
Several strategies for precise snapshots are possible:

FIFOs add ambiguity (existing hardware)

The MAC arranges for FIFOs to be nearly empty, at critical times

The PHY signals the actual clockSync arrival/departure times

Uncompromised precision



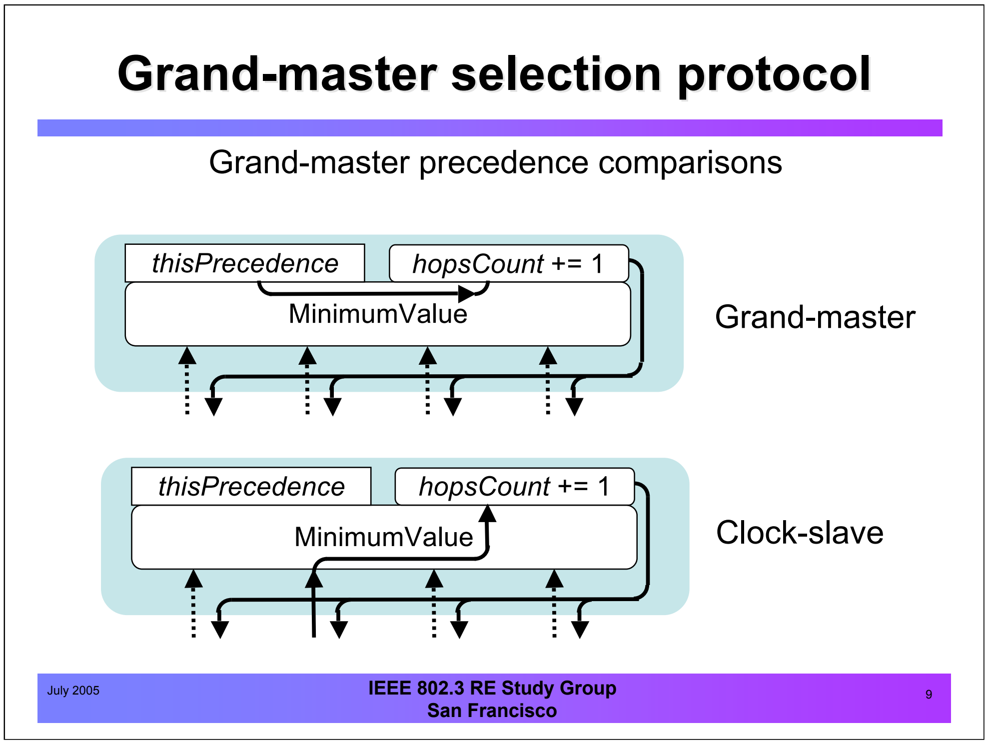
With a 200PPM clock deviation, offset adjustments have limitations.

The problem is the drifts, due to clock-frequency differences.

Thus, the most apparent solution is to have the “slave” match the rate of the master.

Frequency deviations are easily measured, from time snapshots measured over a larger time interval (100ms, perhaps).

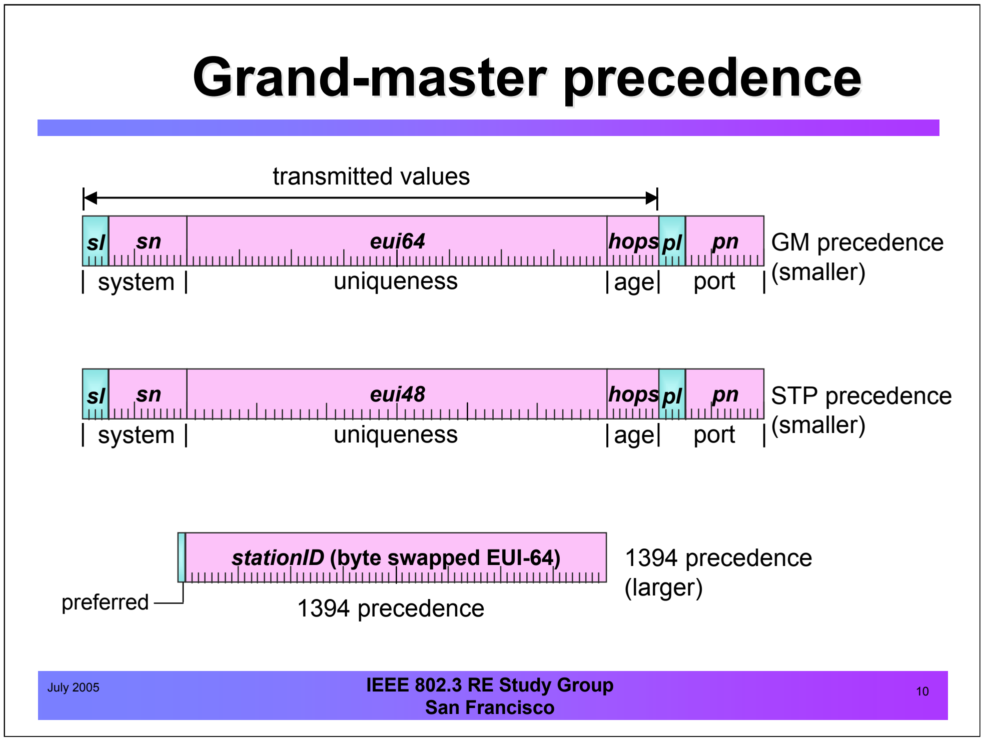
Various frequency compensation schemes are possible; the above waveform illustrates one possible scheme.



For the grand-master selection, spanning-tree protocol is popular in 802.1.

The “minimum” value is distributed throughout the network.

A hopCount value breaks ties, in favor of the shortest-span lengths.



What is the precedence value?
The 802.1 spanning-tree protocol assumes “smart” things set the precedence, “simple” things do the comparisons.

The precedence numbers must be unique, so that only one clock master will be selected. For this purpose, the station address is sufficient.

To communicate preferences, a *sp* (station priority) value is provided. This overrides the MAC address, allowing users to assert their preferences. This weighting can be accessed through the MIB.

For stations with equal *sp/systemID* values, the *stationId* becomes the tie breaker.

For stations with equal *systemTag/systemID* values, the *hops* becomes the tie breaker. This resolves grand-master in favor of smallest hops

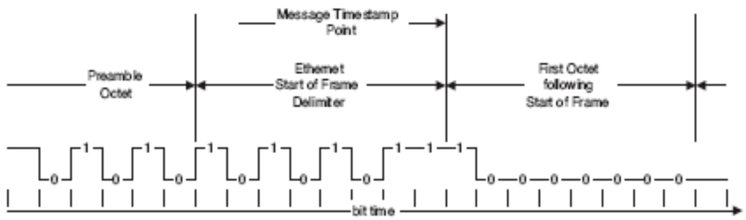
For ports on a bridge, the *portLevel* (pl) and *portNumber* (pn) are similarly used as tie breakers that select between available ports. This resolves same-hops messages on the basis of the arrival ports.

The lowest numerical value has the highest precedence.

Default weighting is the largest numerical value.

The setting of lower-valued weights is a higher level protocols and is beyond the scope of this standard.

Timing specifics…



(from IEEE 1588-2002, subclause D.1.1, page 127)

What (exactly) is the frame arrival/departure time?

Depends on the physical layer details.

Some are already specified, in IEEE 1588.

Parallel-bit transmission schemes may need clarifications.

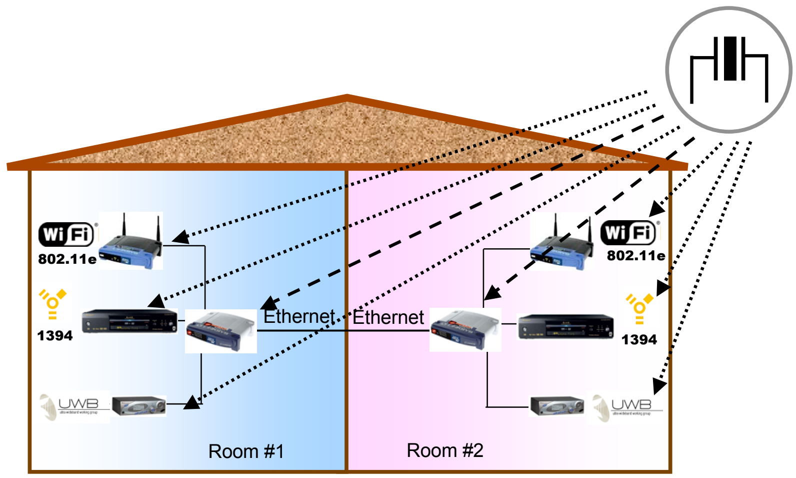
1G CAT-5
10G (in general?)

Backup slides for
Residential Ethernet:
Time-of-day timer synchronization

Maintained by David V James

Backup slides, for pre-review and-or extended question responses.

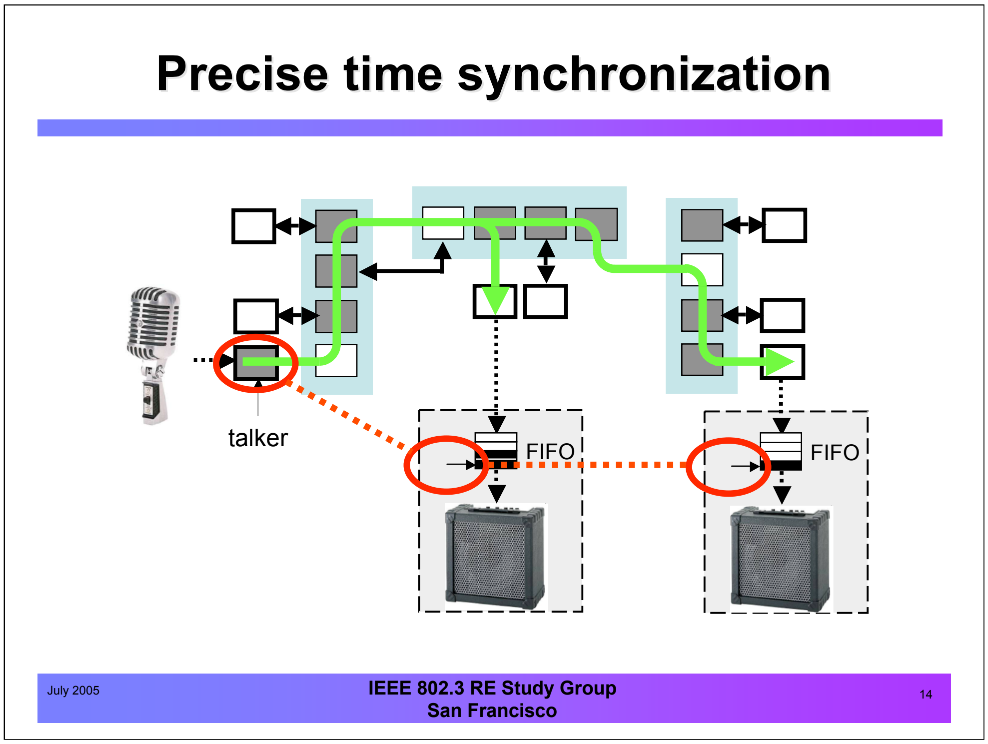
House reference clock



In support of synchronous transfers, all RE devices are assumed to have the same impression of time.

For this presentation, assume an 8kHz cycle time, although a decision on this value has not been finalized.

Requirement: 8kHz cycle frequencies are locked and the “same

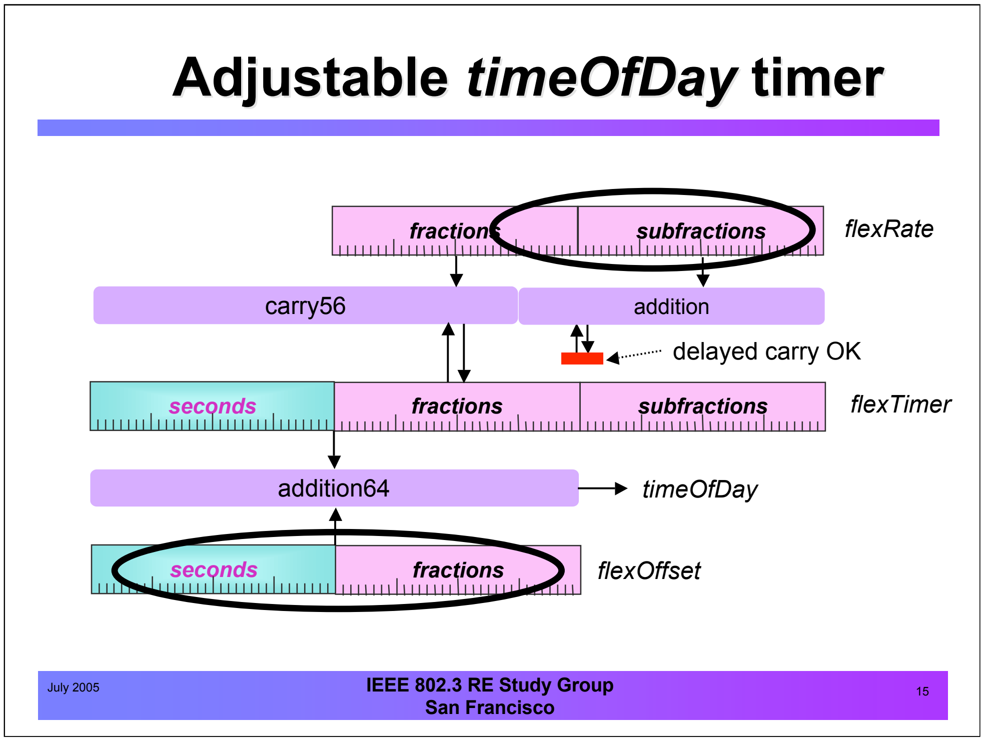


Data arrives early; data is gated at its presentation time.
(Each frame has a presentation time stamp.)

Bridge reclocking has a relatively modest clock-sync accuracy requirement, where microsecond deviations could be acceptable.

Source-data and presentation-data clocking requirements are more severe.

1) Frequency drift is unacceptable, since dropped/replicated values are audible.
2) Presentation time jitter is sub nanosecond, based on slew rates and D/A accuracies.



The basic model assumes rate-adjustable and offset-adjustable clocks.
Since bridge values are rarely needed:
  Any high-enough frequency clock is sufficient
  Value can be converted “on-demand”.

The rate can be adjusted, by updates to flexRate.

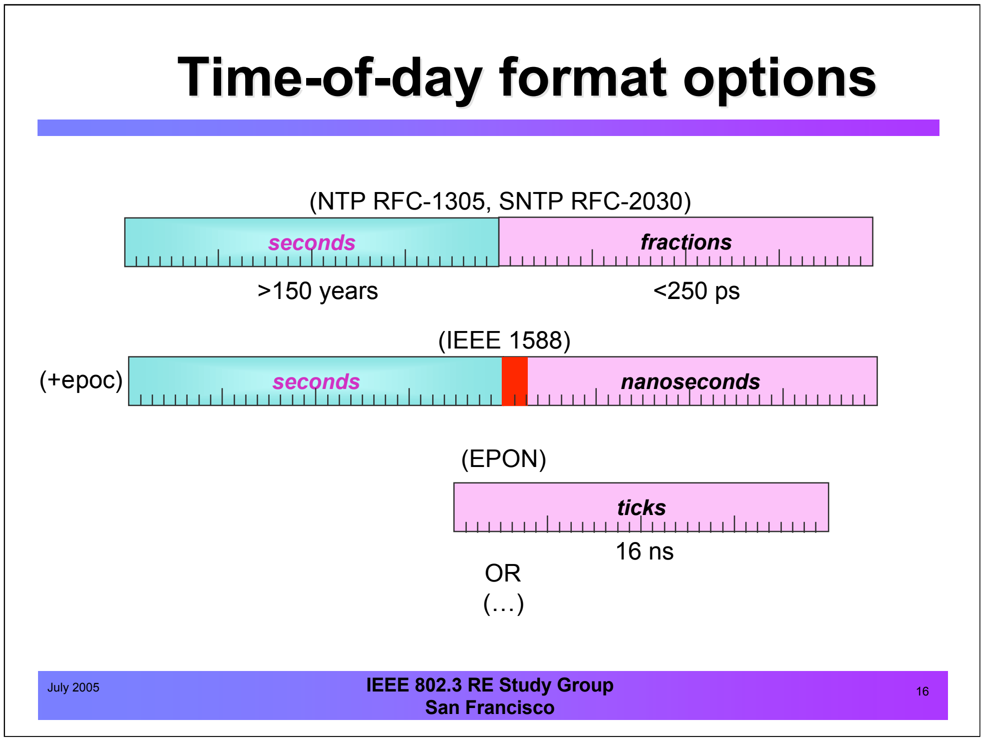
The offset can be adjusted, by changes to flexOffset.

How is a fractions-of-second timer conceptually implemented?
Clocks are typically multiples of something else!

The clock can tick at a natural rate, and the tick-size need-not be “one”.

For example, consider a clock that is updated with a 16ns clock.
  The update value is 68.719476736
  Since the LSB is insignificant, the carry can be delayed

The less significant bits provide precise rate adjustment, for precise tracking



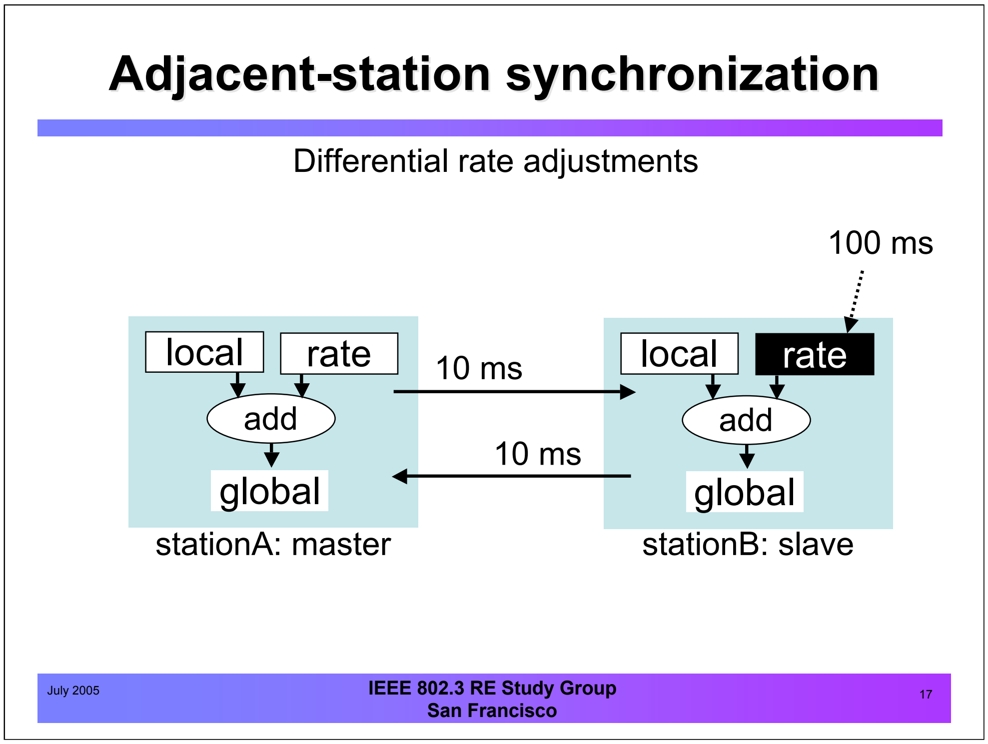
What is synchronized?

Preferably a binary seconds and fractions-of-seconds value.

- Doesn’t overflow within our lifetimes.
- Time resolution induced errors are insignificant.
- Easily added and subtracted.
- Readily converted to other formats…

Other formats are also possible:

- because we have 10 fingers
- a multiple of the cycle frequency
- ….

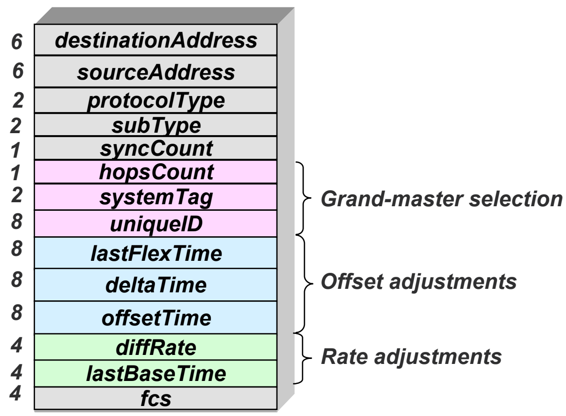


Adjacent-station synchronization is preferrably done less frequently, so that the time-snapshot errors are smaller.

For a 10x longer interval, the effective frequency error is nearly zero.

Yet, the sampling interval is small enough to compensate for modest time-varying temperature drifts.

Frame format



What is contained within each frame?

The standard header.

A syncCount to detect missing frames, thus avoiding last-cycle sampling errors.

Values for the grand-master selection.

Values for the offset adjustments

Values for the rate adjustments

And, this all fits in a minimum-length 64-byte frame!
(Thus, there is no advantage to having “smaller” frames.)

Basic requirements

- KISS (keep it simple, stupid)
  - Delayed snapshot processing
  - Periodic symmetric transmissions
  - Etc., etc.
- NTP (RFC-1305) and SNTP (RFC-2030)
  - Definition of the 64-bit time-of-day value
- For a detailed summary, see:
  - http://dvjames.com/esync
  - dvjTimeSync2005Dec12.pdf (or later revision)