Synchronized time-of-day clocks
(a Residential Ethernet SG presentation)

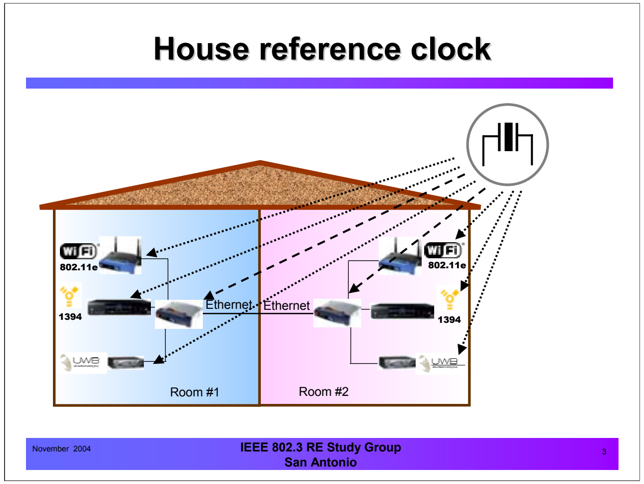
David V James, JGG
Alexei Beliaev, Gibson

Support of real-time streaming traffic requires synchronized clocks.
We address the topics of: what, why, and how.

Synchronized time-of-day clocks

What?

What are synchronized clocks?
Are we legislating crystal frequencies (no).

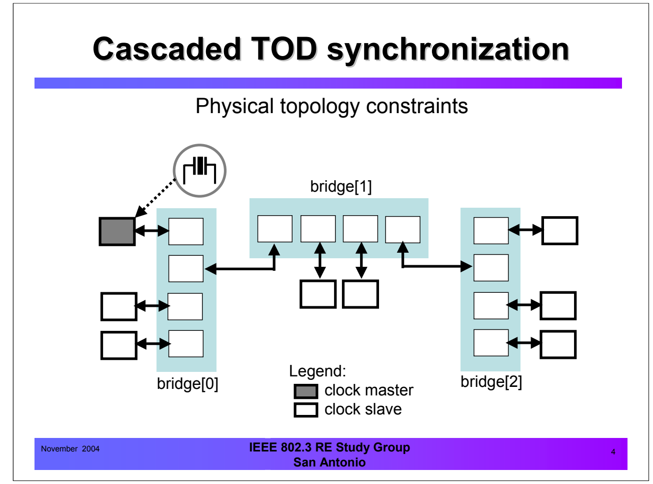


In support of synchronous transfers, all RE devices are assumed to have the same impression of time.

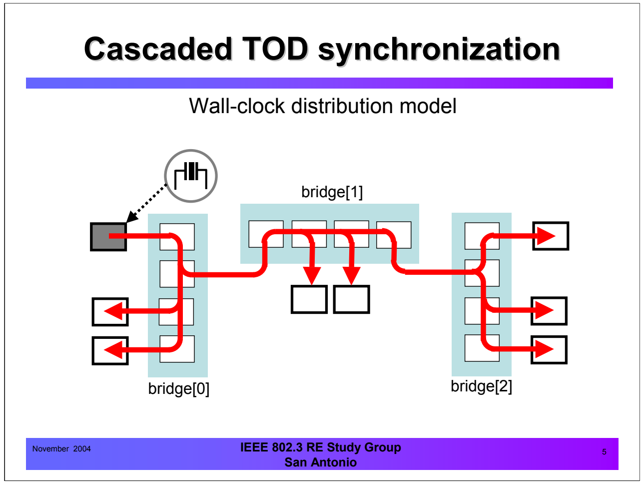
For this presentation, assume an 8kHz cycle time, although a decision on this value has not been finalized.

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

Desired: cycle-count and cycle-offset are also the same.



The clock master is the station whole time-of-day clock is the reference.

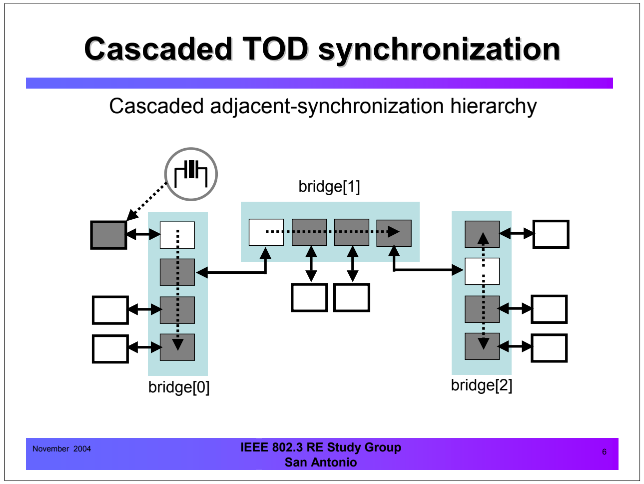


From a logical perspective, the clock master broadcasts the current time-of-day to the attached stations.

From a physical perspective, a multicast time distribution is inaccurate:

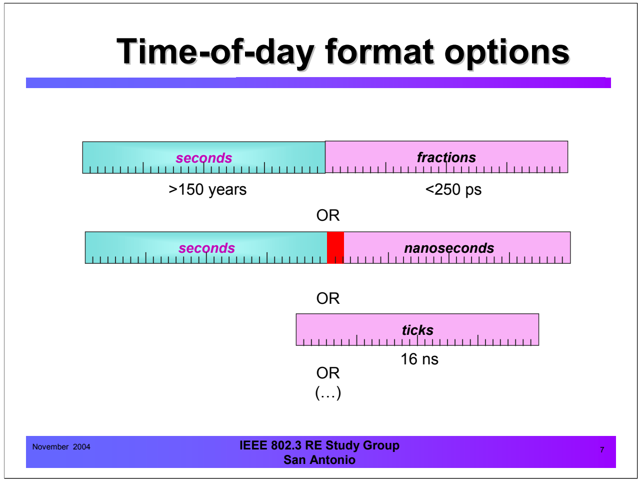
1) There may be source transmission delays.

2) There may be bridge forwarding delays.



To simplify the problem, the link specification defines:

clock-master selection
clock-value synchronization (adjacent "ports" only)



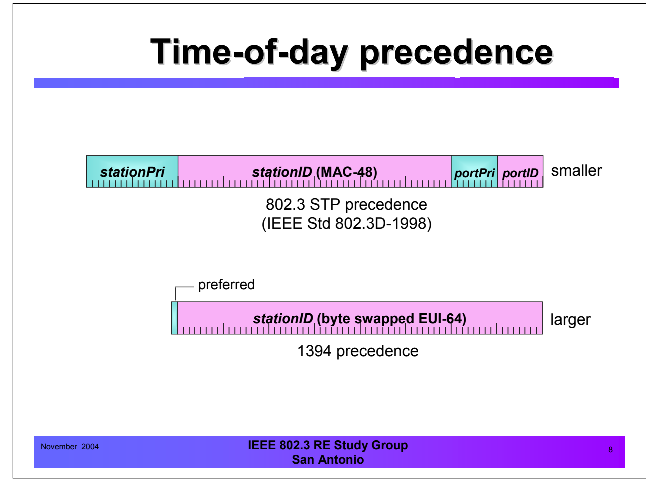
What is synchronized?

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.

Other formats are also possible:

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



The precedence numbers must be unique, so that only one clock master will be selected.

To communicate preferences, a *stationPriority* value is provided;
its default value is all ones, which corresponds to the lowest priority.
This weighting can be accessed through the MIB.

For stations with equal *stationPriority* values, the *stationId* becomes the tie breaker.

For ports on a bridge, the portPri and portID are similarly used as tie breakers that select between available 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.

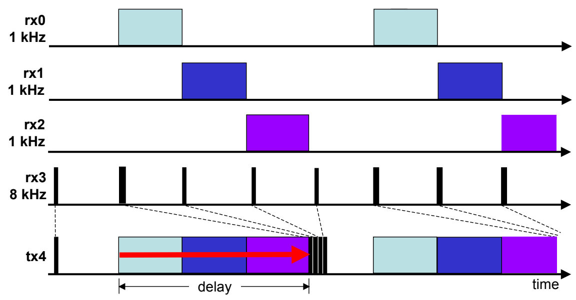
Synchronized time-of-day clocks

Why?

November 2004 IEEE 802.3 RE Study Group San Antonio 9

Now that we know what, the question is “why”.

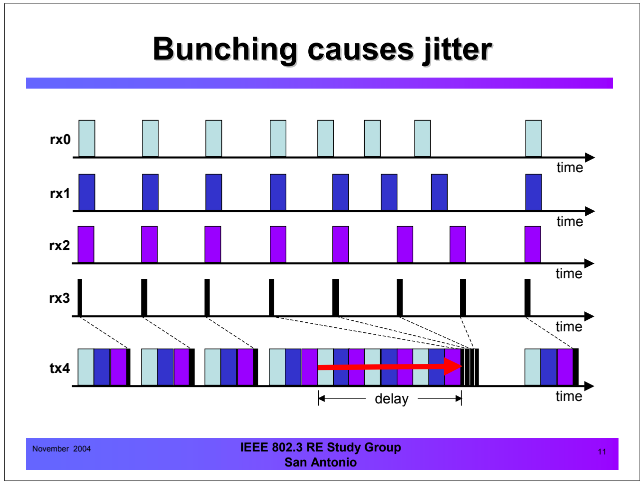
Bursting causes jitter



November 2004 IEEE 802.3 RE Study Group San Antonio 10

We assume transfers will be cycle-clock based, with an 8 kHz clock.
Distinct cycle-clock rates are explicitly disallowed, due to the latency impact of (in this example) 2 kHz transmissions on 8 kHz transmissions.

While the long term bandwidths are less than the link capacity, the short-term bandwidths are not.
And, from the perspective of the 8 kHz transfers, the 2 kHz transfers appear to be transient line-rate transmissions.
Excessive 8 kHz delays occur during the unfortunate transients.



To solve bursting, force sources send at 8 kHz, using only frame lengths (not transmit-frame periodicity) to adjust their transmission rates.

Is this sufficient to bound the latency delays?

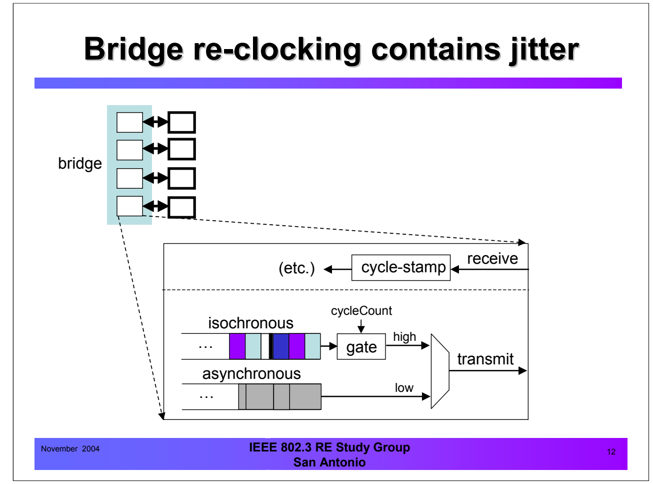
No, there is still the problem of bunching, caused to “early” arrivals.

Consider, for example, a bridge that receives some isochronous data “early”.

The cumulative effect of multiple early transmissions is similar to bursting: a low-rate transfer can be blocked for multiple cycles.

The basic problem is the cumulative effect of bursting and bunching, which depends on the number of bridges, the number of bridge ports, and the interconnection topology.

The cheapest solution is to stop the cumulative effects, so that an end-station and retransmitting bridge port have isochronous timing behaviors.



To ensure the same end-station/bridge behaviors, the same transmission model is assumed.

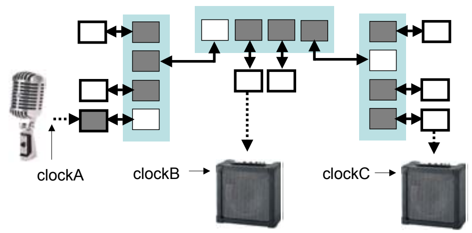
Isochronous data is not immediately transmitted at highest priority.
* The isochronous data is staged in the transmit buffer, before the target transmission cycle.
* When the cycleCount is reached, the isochronous data is transmitted at highest priority.

This _does_not_ ensure the minimal transmission latency, since frames are often delayed “unnecessarily”, to the start of the “next” cycle.

Depending on link bandwidths, max packet sizes, etc., “next” could be more than one cycle (four isochronous cycles is simple and more than sufficient).

This _does_ ensure a bounded min/max transmission latency, and that latency remains unaffected by the number of bridge traversals, port counts, etc..

Synchronized reception/presentation



No long-term drift: clockA, clockB, clockC
Clock jitter: sub nanosecond (after PLL)

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.

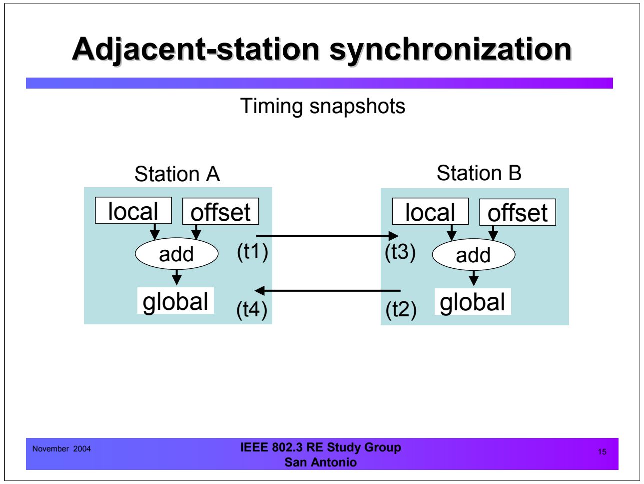
1) Frequency drift is unacceptable, since dropped/replicated values are audible.
2) Presentation time jitter is sub nanosecond, based on slew rates and D/A accuracies.

Synchronized time-of-day clocks

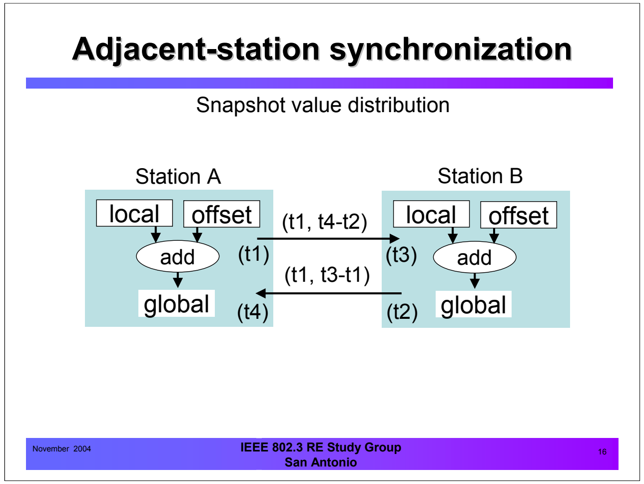
How?

How can this be done?

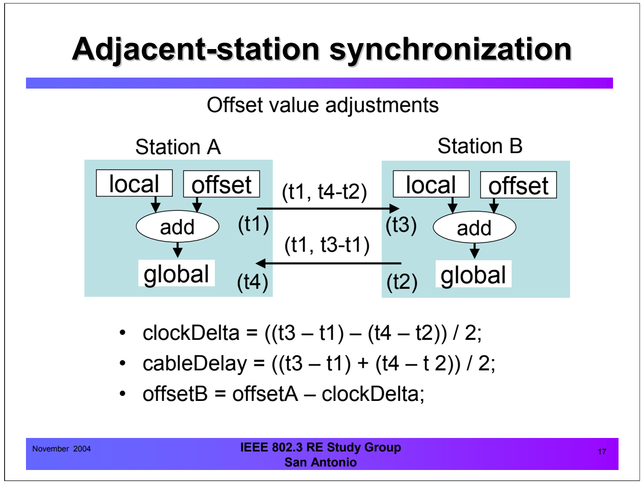
Are synchronization delays affected by cable lengths?
(No, one the asymmetry in transmit/receive paths)



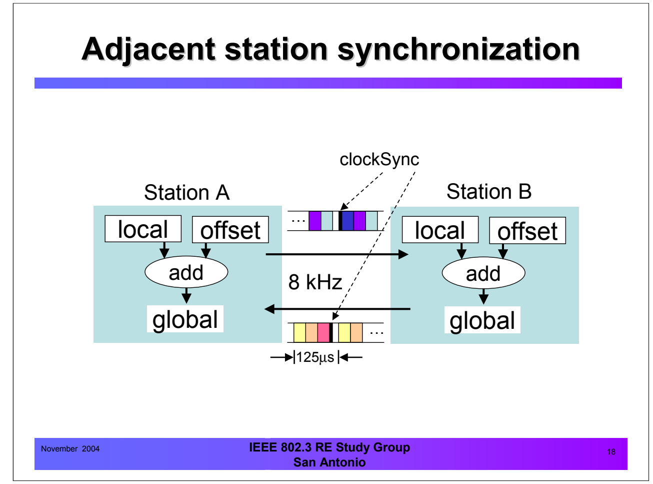
The design model assumes stations snapshot the arrival/departure times of clockSync frames, illustrated as times {t1, t2, t3, t4}.



In the next cycle, these sampled value can be communicated to one's neighbors.



From sampled values, the new offset for the clock slave is readily derived.
The cable length is also readily computed.
The symmetric portions of the cable delays cancel, introducing no errors.
The protocols remain the same, regardless of master/slave selection, except for the final offset-value adjustment.



From sampled values, the new offset for the clock slave is readily derived.
The cable length is also readily computed.
The symmetric portions of the cable delays cancel, introducing no errors.
The protocols remain the same, regardless of master/slave selection, except for the final offset-value adjustment.

In summary

- Time-of-day synchronization (house clock)
  - Global synchronization is required
  - Implemented as cascaded adjacent synchronizations
- Time synchronization formats
  - Binary time is accurate with simple add/subtract
  - Clock-master voting: 48+ or 64+ selection priorities
- Time-of-day applications
  - Synchronous reception and presentation, within applications
  - Synchronous re-clocking within bridges
- Time-of-day distribution
  - Pipelined sampling for highest accuracies
  - Cable delays can be derived, based on the same information

The summary…

Synchronized time-of-day clocks

Questions?

Opportunity for questions.