SEMTECH

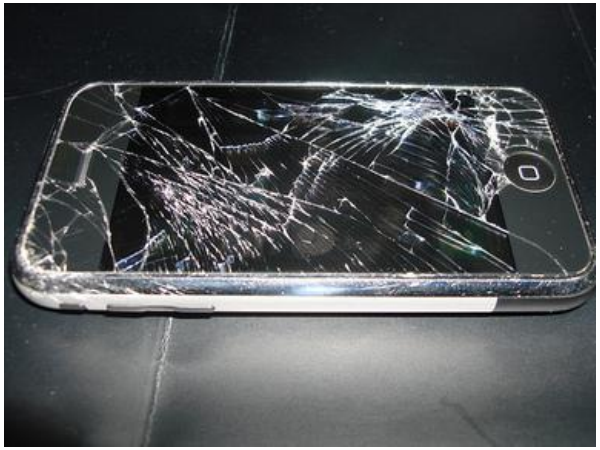
www.semtech.com

Redundancy in PTP Networks

David Spencer – Product Marketing Manager

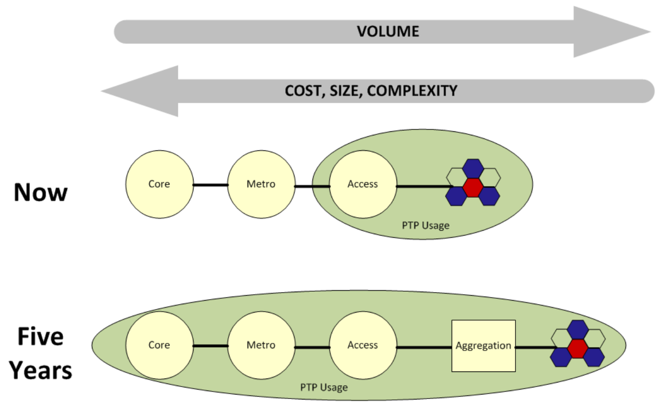
Redundancy for Cellular Backhaul

- Two big changes are coming as LTE gains popularity
  - We need time – frequency isn't enough anymore
    - Both for network sync (LTE-TDD) and for end applications
  - We need time everywhere
    - Small cells are being deployed all over the place (supposedly)
- Have to develop reliable time delivery methods
  - Users just expect it to work
  - People hate dropped calls, frozen video and so on …
- To achieve this implementers have to build in redundancy
  - No one sync method will ever be reliable enough long-term
- So we can avoid this …



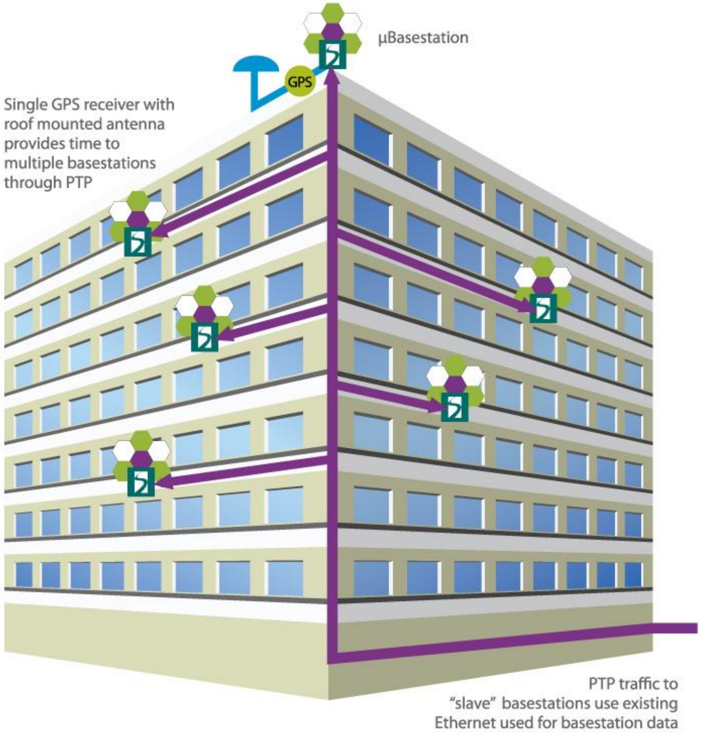
Localized Sync

- **The proliferation of small cell deployment calls for a different sync architecture**
  - We are seeing a move towards local clusters of small cells
  - Each member connects back to an aggregation point
    - Maybe combined with a macro basestation
  - Aggregation node is hub for synchronization
    - Provides sync to its cluster members
  - Sync redundancy implemented in this aggregation node
    - Multiple sync sources to ensure high availability
  - Typically no redundancy in the individual small cells
    - Potentially too expensive to implement
    - Impact of losing one small cell should be fairly low



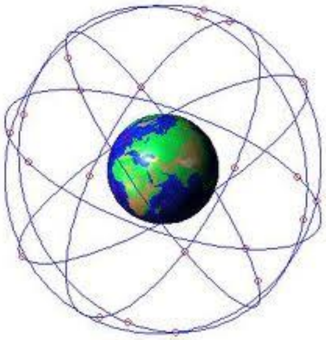
Example Small Cell Deployment

- Combination of macro base station and small cells in large office building
- Sync network uses combination of GPS and PTP
  - GPS to macro base station
  - PTP to small cells



When GPS isn't enough

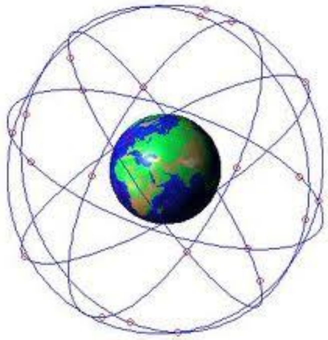
- **Mobile operators adopted the "GPS everywhere" approach**
  - Simple to implement on a traditional macro basestation
  - Relatively low cost
- **But GPS isn't reliable**
  - We all know that!
  - For frequency-only a good OCXO can provide holdover, but more of a challenge for time
    - 1 µs in 24 hours is a frequency error of < 12 ppt (yes – trillion)
    - And that assumes you were perfectly aligned to start with
    - Requires rubidium or advanced quartz technology
  - And what about spoofing?
    - If GPS is the only source of timing then subtle disturbances won't be spotted
    - Leaves the whole network wide-open to attack
- **For these reasons alone, GPS is not enough**



But There's More …

SEMTECH

- **There's a big move towards small cells**
  - Increase bandwidth available for given spectrum by reducing users per cell
  - Small cells will be everywhere
    - Lamp posts
    - Traffic light poles
    - On each floor of high-rise buildings
- **This is a big problem for using GPS**
  - Could be no view of sky if inside a building
    - And cabling to a roof-mounted antenna could be cost-prohibitive
  - Even if there is sky view there may be problems with multi-path reflections
    - From adjacent building
- **Therefore, two real-world reasons why GPS alone won't cut it**



But Isn't IEEE1588 the Answer?

SEMTECH

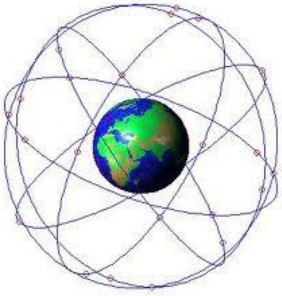
- **Weren't we sold 1588 as the best thing since sliced bread?**
- **PTP (IEEE1588) is good, but it's not that good**
  - The devil is in the detail
- **PTP would work perfectly over a noise-free, symmetric network**
- **But real networks aren't like this**
  - Network traffic adds noise – so-called packet delay variation (PDV)
  - PDV can reduce accuracy significantly
  - PTP-aware networks and clever slave algorithms help mitigate against PDV but there may still be network conditions that render PTP unworkable at times
  - Network asymmetry causes time offset
    - Can be near static – impossible to filter
  - Network outages will break PTP, but maybe still necessary to maintain sync
- **We could build dedicated network for PTP**
  - But only with a big cost penalty

IEEE CONFORMITY ASSESSMENT PROGRAM IEEE 1588 CONFORMITY ALLIANCE

What we Need is Both

SEMTECH

- GPS + PTP used together provide high-availability sync
- Typically, GPS will be the primary timing source …
  - … and PTP will be a backup
- During normal operation downstream sync derived from GPS
  - PTP provides secondary ‘probe’ sync source
  - Can be used to detect anomalies in GPS time
    - For example through spoofing or multi-path effects
- Switch to PTP flow when GPS not available
  - Total outage
  - Or cannot be trusted
  - Switch back once GPS restored
- Could use PTP as primary source of time
  - But unlikely because of PDV effects etc.



+

IEEE CONFORMITY ASSESSMENT PROGRAM IEEE 1588 CONFORMITY ALLIANCE

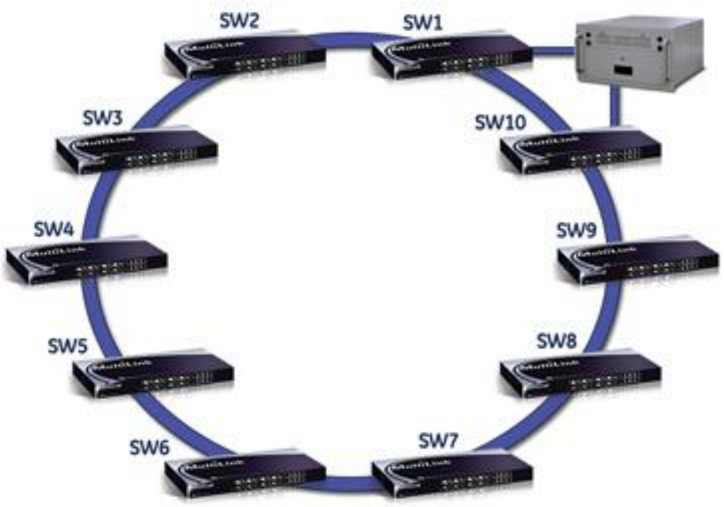
What If We Only Have PTP?

SEMTECH

- In some locations GPS may not be available at all
- In which case, we can use two PTP flows
- Very similar to GPS + PTP
  - Except now, we have to use PTP as our primary flow
  - Can use backup flow to detect disturbance or asymmetry change in main flow
- Need to think carefully about network architecture
  - Don’t want both PTP flows coming from the same grandmaster
    - Single point of failure at the grandmaster
  - Want to try and avoid both PTP flows following the same route
    - Since any asymmetry will be common to both flows and undetectable
- Workable, but more of a challenge than GPS + PTP

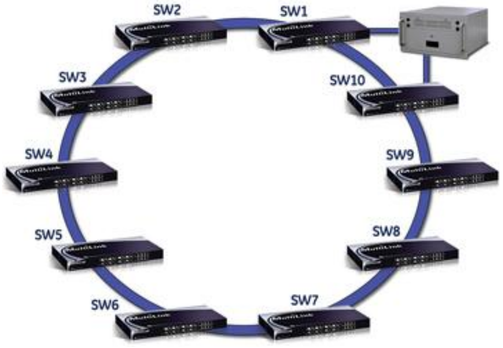
Ring Architectures

- **Ring architectures are already common for data transport**
  - Traffic can pass from any node to any other node through one of two routes
    - Traditionally called East and West (viewed from bottom of ring)
  - Provides alternate route in event of single failure
    - As long as ring is geographically separated
- **For communication between any pair of nodes one path is likely preferred**
  - Normally the shortest one
  - But other path provides a reasonable backup
  - Different for different nodes
    - For example SW9 would talk to SW1 anticlockwise while SW5 would talk to SW1 clockwise



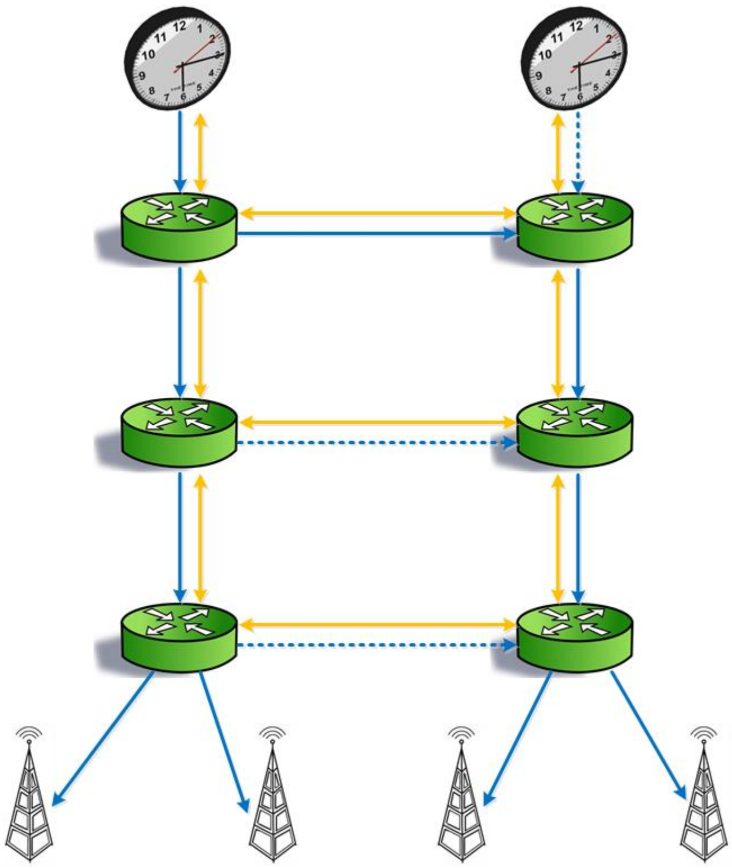
PTP with a Ring Architecture

- How can we use PTP with a ring architecture?
- Each node represents an aggregation point or a grandmaster
  - At least two grandmasters in the system
- Any node can reach any grandmaster through two routes
  - Lock to two masters simultaneously
  - With careful design can avoid shared routes
- Each node can be both boundary clock and slave
  - Boundary clocks need to support east and west flows
  - This could be a problem
    - BCs are single one->many flow
    - We need dual one->one flow
  - WE can solve this problem by having dual-domain BC
    - One PTP domain for “east” flow
    - Another for “west”
  - This will require some tweaking to current standards



Probe Method

- ❑ **Could also use peer-to-peer delay measurements as probes**
- ❑ **Network of multiple masters and slaves**
  - Boundary clocks at each node
- ❑ **Blue lines indicate primary timing flow**
  - Dotted when excluded by BMCA
  - Only one master is active
- ❑ **Orange lines show Pdelay secondary timing flow**
  - Active between each node and its neighbors
  - Allows each node to get a secondary timing source to use as a probe
- ❑ **Remains to be seen how practical this is in a real network**

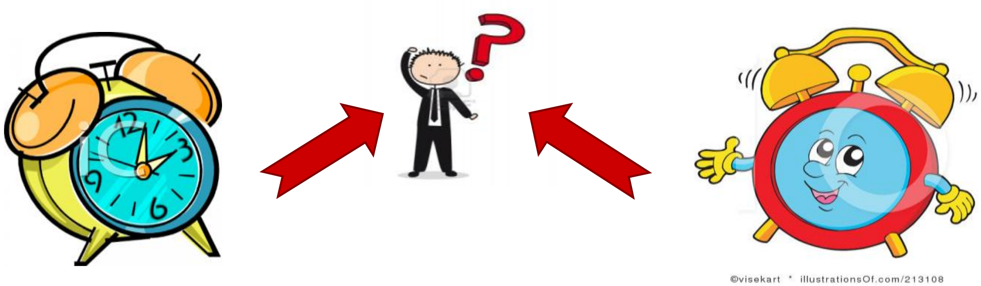


PTP + Synchronous Ethernet

- **Synchronous Ethernet is frequency only**
  - But it's a stable frequency
  - Can be used to maintain a local short term time-holdover clock
- **Maintain two clocks synchronized to PTP**
  - One higher bandwidth locked directly to PTP
  - Second clock lower bandwidth using SyncE frequency as "tick"
    - Effectively a local time holdover clock
- **During normal operation both clocks will remain the same**
  - But an asymmetry change or disturbance will cause the two to differ
  - If the change appears constant enough over the period in which we have confidence in the SyncE clock then the difference can be used as an asymmetry correction
- **This is really just a smart use of a PTP / SyncE hybrid mode**
  - How well it works depends on the system architecture
  - Congruent mode – not so good
  - Coherent mode – possibility of excellent immunity to network changes

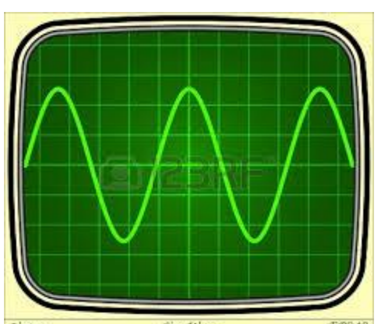
Majority Voting

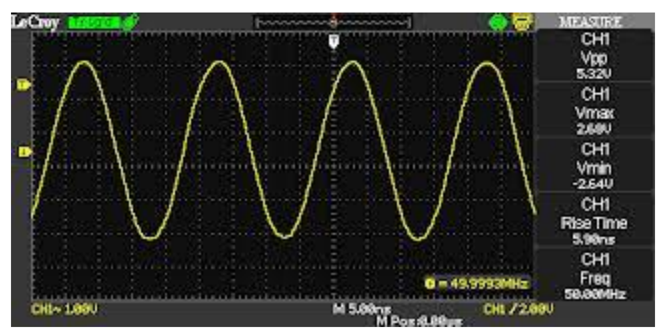
- **Ideally we should have three timing flows**
  - To allow majority voting to determine source of discrepancy
- **Can use:**
  - GPS + PTP + PTP
  - GPS + PTP + Local Oscillator
  - PTP + PTP + Local Oscillator
- **For example, two timing flows A & B and local oscillator L**
  - B and L stay in sync – A jumps with respect to them
    - A has gone bad
  - A and L stay in sync – B jumps with respect to them
    - B has gone bad
  - A and B stay in sync – L moves with respect to them
    - Likely that the local oscillator has drifted



Good Times … Bad Times …

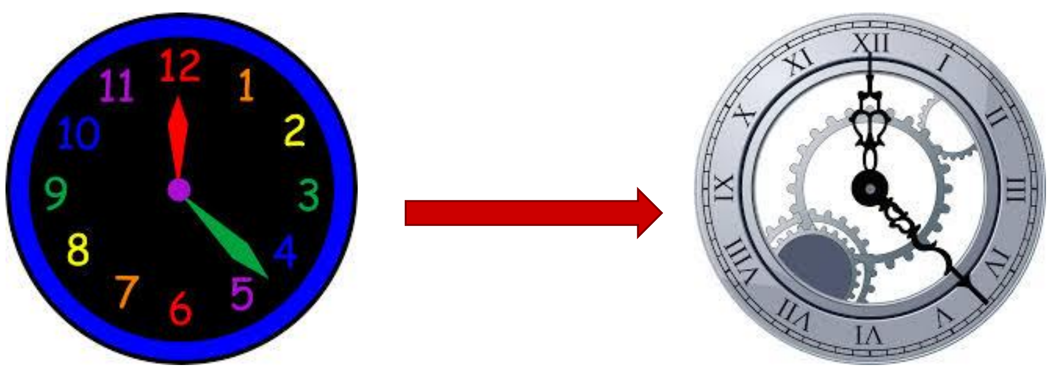
- **So, that's how we get multiple sources of time**
  - But what do we do with them
- **If a failure occurs we need to switch between sources**
- **In the frequency world this is simple:**
  - Phase Build-Out (PBO)
  - Method for switching between two clocks that may have different phase alignments and slightly difference frequencies
  - Works by adjusting output phase in gradual steps until output frequency matches new input
  - Step size can be chosen to control maximum fractional frequency offset
    - Something specified by sync standards
    - Don't want to pull too quickly or too slowly





What About Switching Time?

- **We need something similar for time**
- **There should only be one true time**
  - After all, UTC it UTC
  - However there may be some variation
    - Particularly if switching between PTP sources
  - Also PTP could be used with an arbitrary epoch
    - For frequency-only delivery through PTP
    - Any alignment possible
- **Therefore:**
  - Need some mechanism to maintain or build-out time alignment between two time sources used for time-of-day
  - Some mechanism to provide PBO between two time sources used for frequency

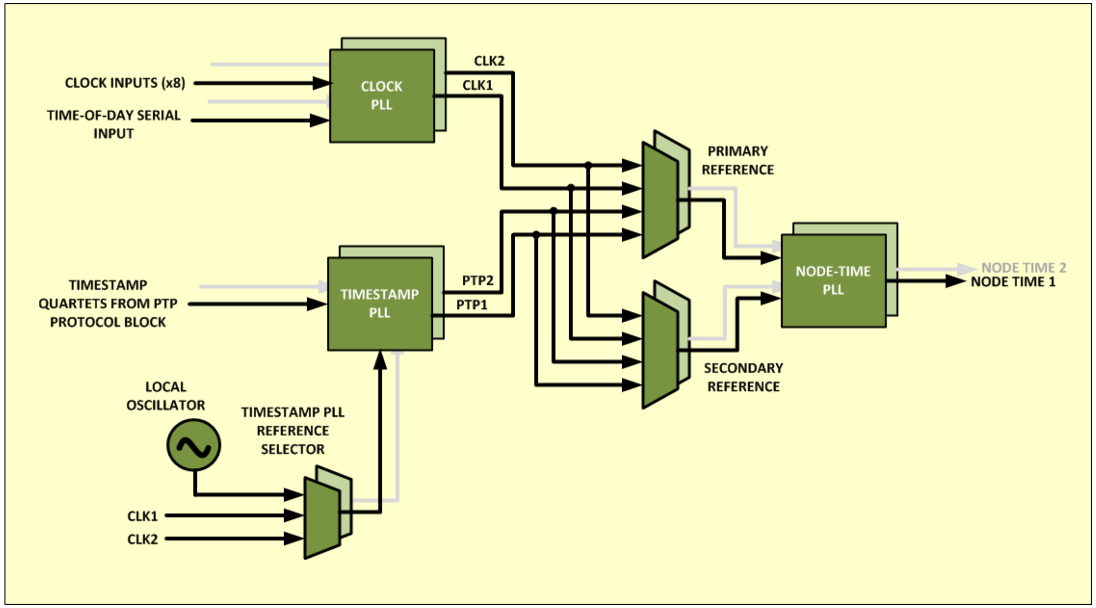


Options for Switching Time

- **Options for merging time depend on nature of source**
- **If both sources are nominally time-aligned**
  - Assume that they are syntonized – same “tick” rate
  - Can either blend to new timebase or build-in offset to maintain output time
  - Blend:
    - Advance or retard time at defined rate until output time matches new timebase
    - Rate either defined as time to complete blend or desired pull rate
    - Only sensible if close alignment between two timebases
    - Probably best approach if switch is permanent (PTP to PTP)
  - Offset:
    - Record offset between old and new time at point of switchover
    - Derive output from new timebase, but offset to avoid any jump in output
    - Good approach if switch is temporary (GPS to PTP)
- **If either source is only frequency aligned**
  - Can use same phase build-out technique as in the frequency world

Slave architecture

- ❑ **How can this be implemented in a practical slave?**
- ❑ **Requires multi-level PLL architecture**
  - One or more traditional PLLs to lock to GPS and maybe SyncE clock
  - One or more packet PLLs to lock to PTP flows
  - “Node-time” PLL to create output from front-end PLLs
    - Handles source selection
    - Disturbance detection and asymmetry correction
    - Phase build-out etc.



Summary

- Sync redundancy in wireless backhaul is vital
  - And getting harder as we need time alignment everywhere
- GPS on its own no longer cuts it for sync
- Neither does a single PTP flow
- GPS + PTP, PTP + PTP or PTP + SyncE provides redundancy
  - By having two timing source options
- Node’s local oscillator acts as third implied source
  - Allowing majority voting to be used to detect issues
- Use of multiple PTP flows requires some careful design
- Slave architectures must be able to make use of multiple flows
  - Particularly in generating a stable output

SEMTECH

www.semtech.com

Thank You

Questions?