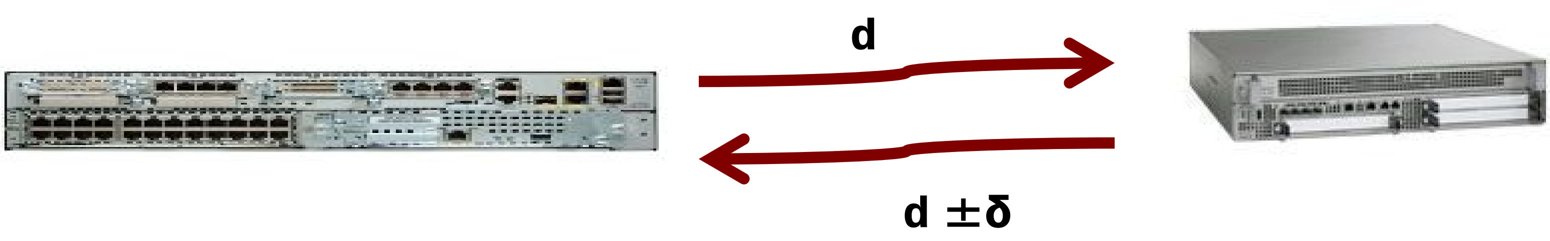
www.semtech.com

Asymmetry in PTP Networks
- What it is and what do we do about it

David Spencer – Product Marketing Manager

What is Asymmetry?

- **Most packet network links are bidirectional**
  - Traffic can flow from A to B …
  - … and from B to A
- **Asymmetry is simply the difference in transit time between the two directions**
- **Since a packet only goes in one direction asymmetry cannot be measured on a per-packet basis**
- **Instead, asymmetry is measured over time**
  - From a few microseconds to a few days or longer
  - Packet times are averaged to determine asymmetry
    - Maybe after some appropriate selection criteria are applied



Why Asymmetry Matters to Sync

❑ It never used to …

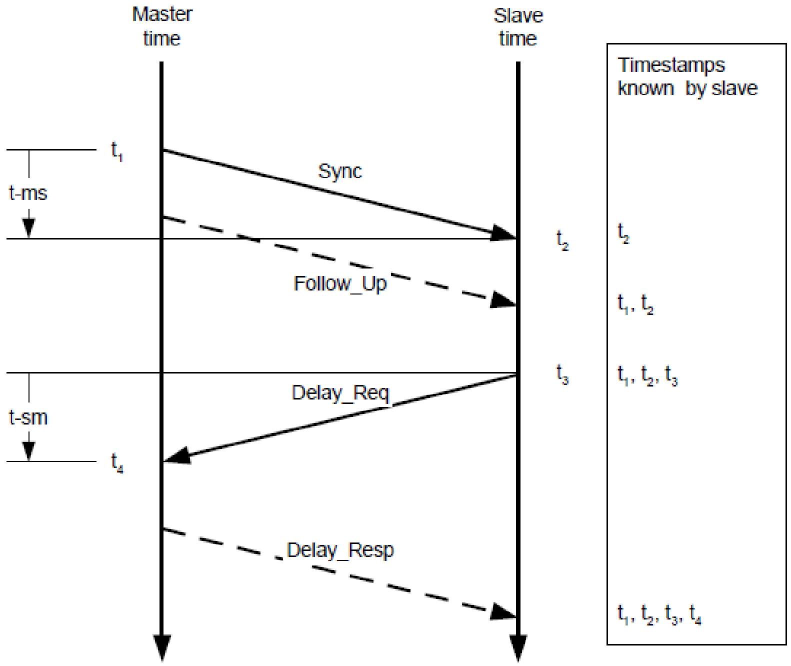
- Back in the days of frequency-only delivery
- Packet-based frequency delivery mechanisms such as adaptive clock recovery can syntonize B to A using just a packet flow form A to B
- Since only one direction is involved asymmetry is irrelevant

❑ However, we now have to deliver time

- Two stage process
  - Syntonize B to A
  - Measure delay from A to B to know how much to adjust B's clock by
- Syntonization is no different to frequency-only case
- Problem is with the delay measurement
  - Chicken and egg situation – to measure delay of packet between A and B requires A and B are synchronized but the whole point of the measurement is to allow synchronization
  - Solution is to measure roundtrip delay of A to B and back to A (or vice-versa)
  - Works because all measurements now at one end
  - However, synchronization requires A-B delay, not A-B-A delay
  - Only solution is to assume A-B delay is half of A-B-A delay
  - Asymmetry breaks this assumption

Synchronization in PTP

- **PTP uses sync messages and delay request messages to measure round-trip delay**
  - And delay response message to get t_4 timestamp back to slave
- **Master to slave delay is calculated as:**
  - Delay = $[(t_4 – t_1) – (t_3 – t_2)] / 2$
  - Subtracts time slave took to respond to sync message from roundtrip delay seem by master
    - Gives roundtrip delay of network
  - Divide-by-two gives master to slave delay
    - **Assuming no asymmetry**
  - **Any asymmetry results in a phase offset of one half the asymmetry**



Defining Asymmetry

- Not all asymmetry is made equal
- We can define three types of asymmetry
  - Based on their duration
  - Or how often they change
- Dynamic asymmetry
  - Changes over a relatively short timeframe
    - As low as few milliseconds
    - To as much as a few minutes
    - Of the same order as the slave's filtering bandwidth
- Static asymmetry
  - Constant – never changes
- Pseudo-static asymmetry
  - The twilight zone between static and dynamic
  - Constant for a lengthy period but not forever

Bandwidth

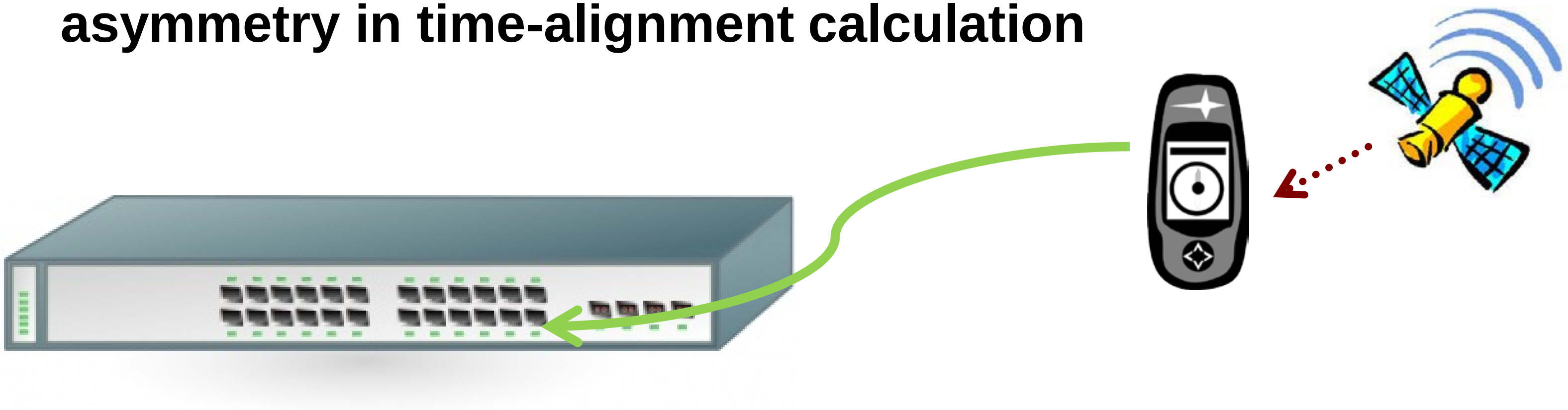
| **Dynamic** | **Pseudo-static → .... Static**

Causes of Static Asymmetry

- Static asymmetry is typically caused by network design
- Cause 1 : Mismatched fiber lengths
  - Separate fibers typically used for each direction
  - Often little attention paid to matching fiber lengths
  - Or lengths deliberately different due to dispersion compensation
  - This will lead to asymmetry
    - Speed of light in fiber is about half of that in a vacuum
    - ~ 6 inches per nanosecond
    - A 200 foot difference in fiber lengths gives 400 ns asymmetry (200 ns phase error)
- Cause 2: Master and slave speed mismatch
  - Occurs when master and slave Ethernet ports are different speed
    - For example 1Gbps master connected to 100Mbps slave
  - Results from store-and-forward nature of most Ethernet switches
    - Packet cannot move through switch until entire packet received
    - Cancels out everywhere except end points
  - 100Mbps / 1Gbps mismatch gives asymmetry of around 8 µs
- Other smaller static asymmetries possible through poor equipment design

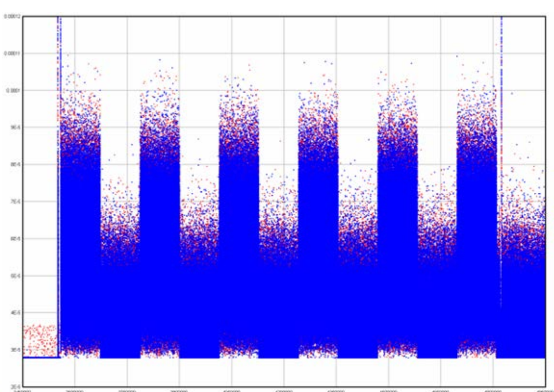
Dealing with Static Asymmetry

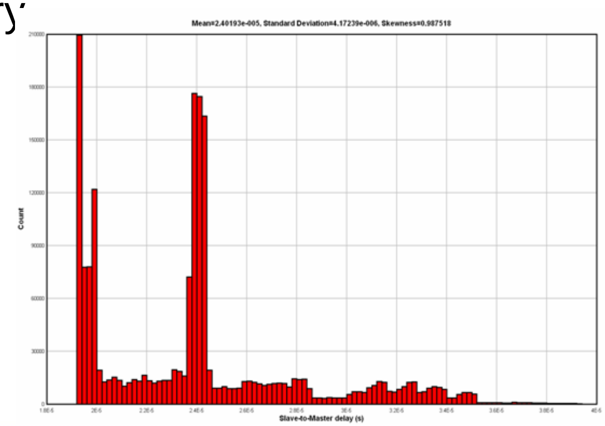
- The good news …
  - Static asymmetry is static
- Therefore it can be eliminated forever if the value is known
- Either calculate the asymmetry
  - For example, resulting from speed mismatch or equipment design
- Or measure it
  - Either independently of the sync solution or with some sort of probe such as a portable GPS receiver
- Once measured, program the sync slave to include the known asymmetry in time-alignment calculation



Causes of Dynamic Asymmetry

- Dynamic asymmetry is almost always caused by congestion
- PTP event messages are competing with other network traffic
- Queue blocking occurs within switches and routers
  - Competition to get both into and out of the egress queue
  - Quality-of-service configuration can partly reduce this by allowing PTP messages to jump ahead of everyone else waiting to egress
  - But QoS does nothing to help getting out of the egress queue
  - Causes variation in packet delay as network traffic changes
- Many applications have very asymmetric traffic flows
  - Think of someone watching YouTube on their iPhone
  - This leads to a traffic-induced asymmetry





Dealing with Dynamic Asymmetry

- **Dynamic asymmetry is by definition short-term**
  - Therefore, the time recovery slave has a fighting chance of being able to filter it out
- **How good a job can be done depends on:**
  - Duration and profile of asymmetry
  - How smart the slave algorithms are
- **Possible algorithmic filtering approaches**
  - Packet selection
    - There may be a clear asymmetry when the average transit time of all packets is considered, but slaves don’t typically look at every single packet
    - Minimum delay may not actually move at all
  - Low-bandwidth filtering
    - If averaging process occurs over much longer period than the asymmetry then the effect will be reduced
    - But puts strain on local oscillator performance
  - There are also a number of more advanced techniques
    - For example, it might be possible to detect when the spread of packet delays in one direction has become excessive and temporarily enter a mini-holdover mode

Causes of Pseudo-static Asymmetry

- **Pseudo-static asymmetry is an asymmetry that could change after an arbitrarily long time**
  - Could change after a network has been operating for weeks or months
  - Or could be a few hours …
  - … or a few years
- **Likely cause is some network architecture change**
  - Instigated by the carrier, or maybe automatic to cover a failure
  - For example, network routing could be changed introducing a new static asymmetry
  - Multiplexing of data onto fiber could be changed
    - ROADM reconfiguration in WDM system
    - Change of wavelength will have effect on packet delay
- **It mat be possible to negotiate a service level agreement that ensures no network changes during normal operation**
  - Could be expensive option though
  - Still doesn’t deal with failure conditions

Dealing with Pseudo-static Asymmetry

- **Can we handle pseudo-static asymmetry in the same way as the others?**
  - Likely too long-term to filter, so the techniques for dynamic asymmetry won't work
  - It's not truly static, so the techniques for static asymmetry won't work
  - So, the answer is "no"

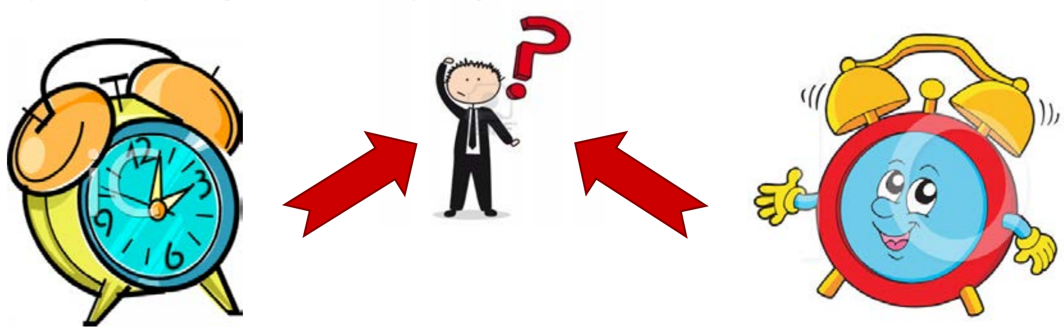
- **The million dollar question:**

"What can we do to handle the effects of pseudo-static asymmetry?"

- **If we can answer this then a major hurdle of PTP deployment has been cleared**
- **If we can't, then PTP will struggle in many applications**
  - And it will only get worse as timing requirements become tighter

Using Two Timing Sources

- **The problem is clear**
  - We have a variable that we cannot predict or measure that impacts the time recovery without the slave knowing what is happening
- **The only possible solution is to have more than one source of timing**
  - We can then use one as a reference to judge the performance of the other
  - This should allow us to detect both asymmetry changes and other disturbances
- **Two possible modes:**
  - Active / active – equal weight is given to both timing sources and the output is derived from some combination of both in normal operation
  - Active / probe – the output is derived from a primary timing source with the secondary one just performing a probe function



Majority Voting

- **There is actually a third timing source**
  - Time derived from the slave’s local oscillator
- **In an ideal system all three timing sources will remain in sync**
  - No network packet delay variation
  - No network asymmetry
  - Perfect stable local oscillator
- **In practice, different events will cause the three to diverge**
- **Consider remote sources A and B and local time L**
  - B and L stay in sync – A jumps with respect to them
    - Likely that network asymmetry to A has changed
  - A and L stay in sync – B jumps with respect to them
    - Likely that network asymmetry to B has changed
  - A and B stay in sync – L moves with respect to them
    - Likely that the local oscillator has drifted
- **Once we can judge what has changed we can correct for it**
  - For example, if time from A jumps with respect to B and L we can use difference to correct a known asymmetry value and bring A back into line

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

Using PTP + GPS

SEMTECH

- ❑ **Everyone knows that GPS is a good source of time …**
- ❑ **But, GPS availability falls far short of that needed for telecoms**
  - Spoofing
  - Deliberate or accidental jamming
  - Multi-path effects
  - Weather
- ❑ **However, GPS is still good enough to be the primary sync source as long as it has a backup**
- ❑ **PTP can be used as that back-up**
  - As a remote alternative to a highly stable local oscillator
- ❑ **The GPS can be used for a continual asymmetry measurement during normal operation**
- ❑ **PTP takes over in the event of a GPS outage**
  - With the likely asymmetry already known
- ❑ **Easily implemented by replacing existing GPS holdover module with PTP-equipped version**

Using Dual PTP Flows

- **Instead of PTP + GPS, how about PTP + PTP?**
- **Need to lock to two separate masters simultaneously**
  - Or could lock twice to the same master
- **Path from slave to each master must be independent**
  - Otherwise asymmetry change on shared portion won't be detected
  - Can relax this for local connection since we're in control
- **This is similar to the standard dual-master upstream redundancy model**
  - Except now, the standby source is used to detect asymmetry changes associated with the primary source and vice-versa
- **Low cost implementation from the slave standpoint**
  - Likely doesn't require any additional hardware
- **However, very hard to architect in practice**
  - Since totally separate routes to masters are required
  - Could require dual links from independent carriers to make it work

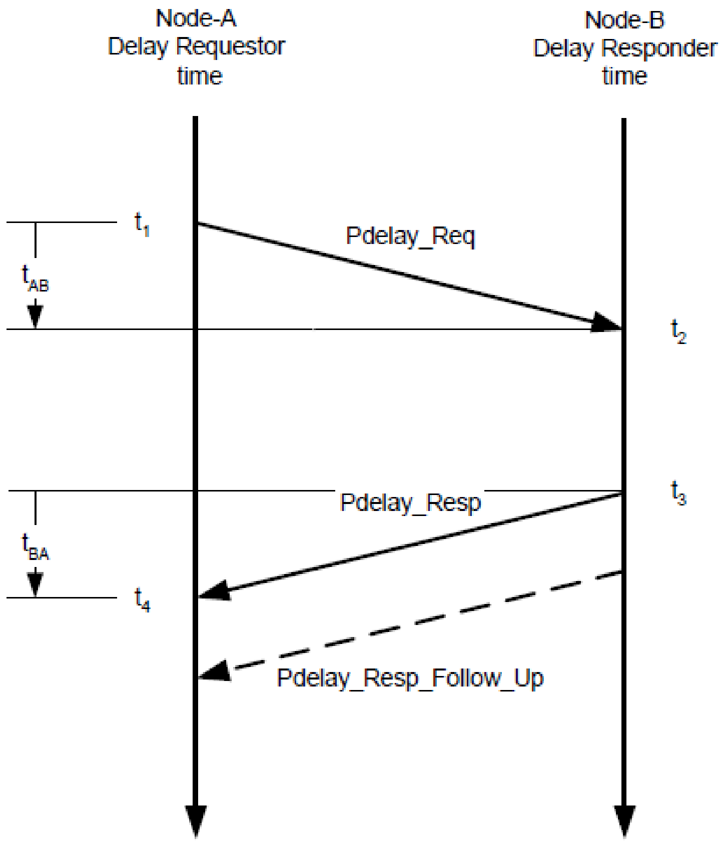
Monitoring an Array of Boundary Clocks

SEMTECH

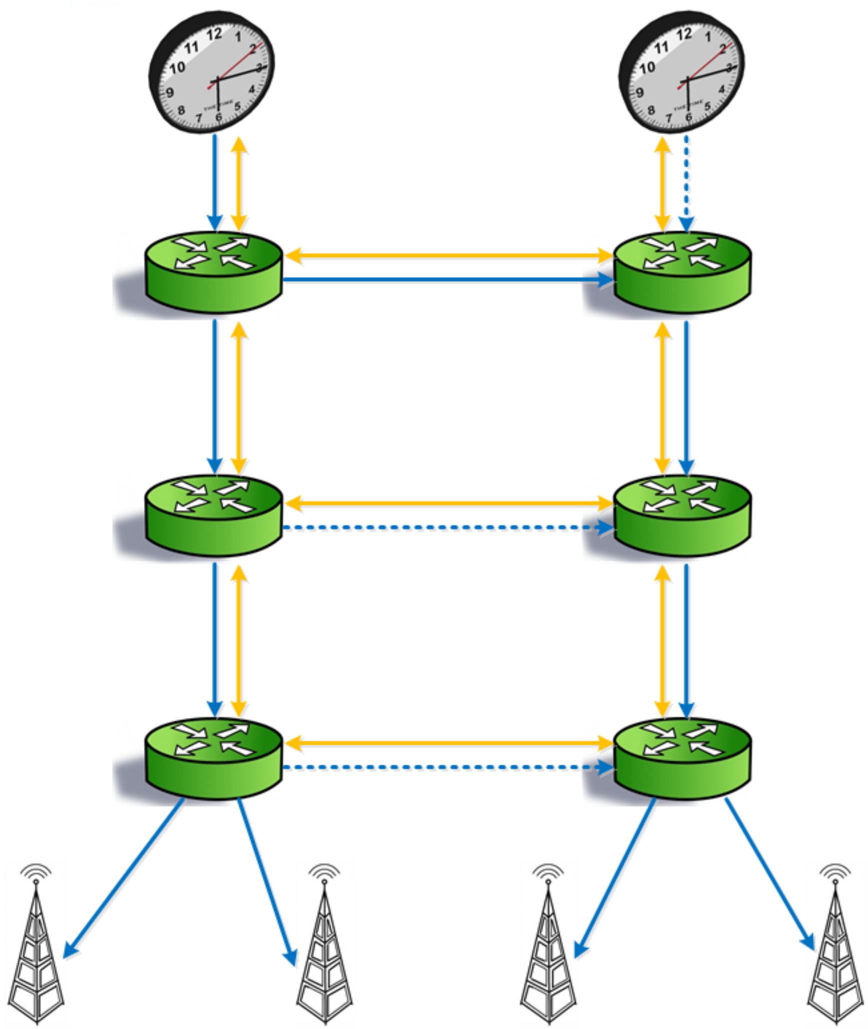
- It has been proposed that multiple PTP flows could be used to monitor the performance of an array of boundary clocks
  - Asymmetry
  - Congestion effects
  - General operation
- Assumes that every node in the network is a boundary clock
- Assumes that multiple masters are available
- Using the standard IEEE 1588-2008 Best Master Clock Algorithm (BMCA) results in:
  - Only one master being selected by all slaves
  - Each boundary clock receiving its timing from a single boundary clock or grandmaster
- All superfluous links to boundary clock forced to passive state
  - Not allowed to take part in standard Sync / Delay Request activity
  - But can still run peer-to-peer delay measurement
- Pdelay mechanism can be used to support secondary timing flow
  - Normally used as part of peer-to-peer transparent clock mechanism

The PTP Pdelay Mechanism

- Runs bidirectionally across all links at all times
  - No master or slave concept
  - Independent of port state
- Works through Pdelay_Req and Pdelay_Resp messages
  - Both timestamped event messages
- IEEE 1588-2008 defines various modes involving timestamps and correction fields
  - Importantly, one mode allows the requestor to get both the responder's time-of-day and the delay
  - Provides a secondary timing source



Example Pdelay Use



- **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 should 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**

Summary

- **Asymmetry impacts packet-based time transport**
  - Will become more of a problem as:
    - PTP deployment becomes more widespread
    - Applications and technologies drive for tighter and tighter sync requirements
- **We need to worry about three types of asymmetry –**
  - Dynamic – short-term
  - Static – never changes
  - Pseudo-static – changes with a long period
- **Different techniques needed to handle each type**
  - Dynamic
    - Clever filtering within the slave
    - Smart network architecture – QoS and avoiding congestion
  - Static
    - Measurement
    - Calculation
  - Pseudo-static
    - Multiple timing flows
    - Majority voting
- **PTP + GPS and PTP + PTP are both possible options**
- **Use of Pdelay mechanism may be an option too**