

60802 Time Sync – RR & NRR Drift Tracking and Compensation – Implication

David McCall – Intel Corporation

Version 1

References

1 – David McCall “[60802 Time Sync – Normative Requirements & Testing](#)”, 60802 contribution, 28 April 2023

Contents

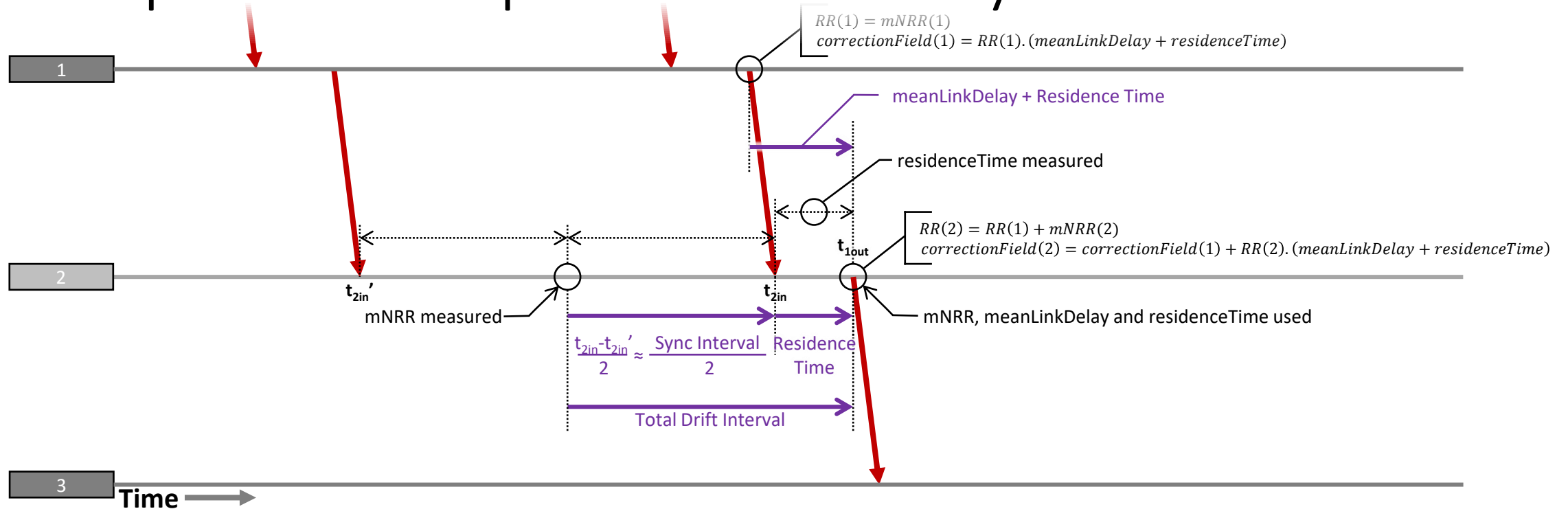
- “Old” error model vs. “New” error model.
 - Implications for NRR, RR, Drift Rate and Correction Field calculations
- Potential for decreasing Sync Interval
 - Simulations
 - Real-World
 - Bad implementaions
 - New Normative Requirements?
 - Back to Simulations
- Summary

“Old” Error Model vs. “New” Error Model

Clock Drift Error – Relevant Intervals

4 Hops – 2nd Hop – NRR from Sync

Old
(From [1])



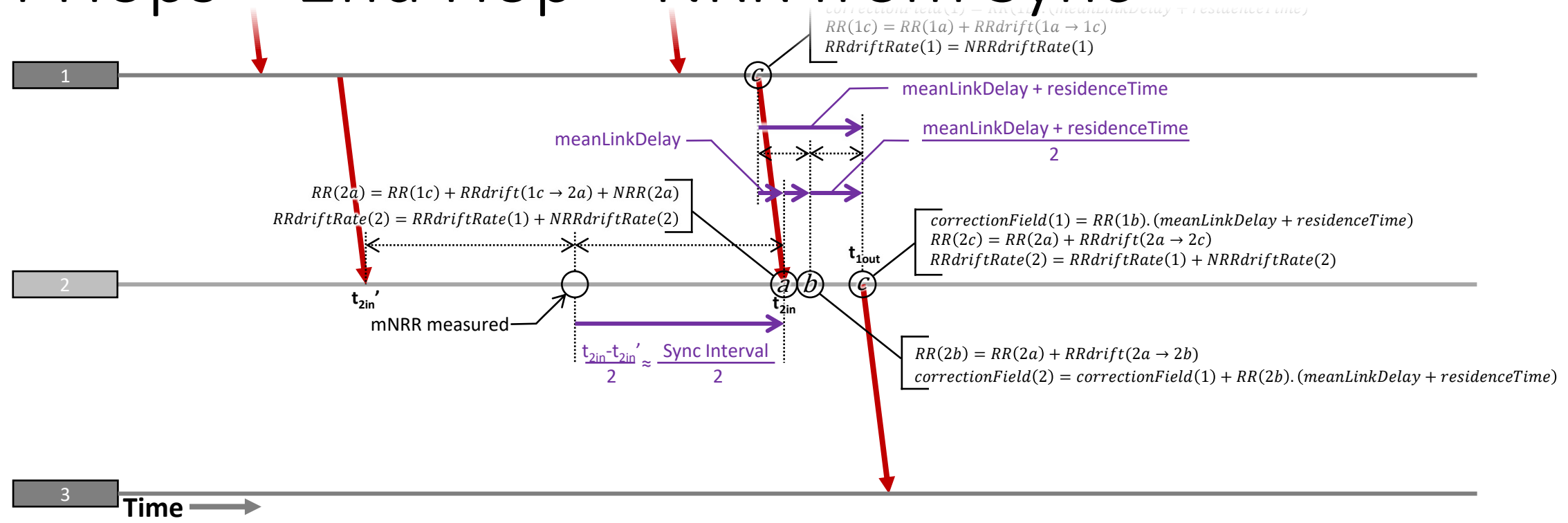
- Same errors in mNRR as 1st Hop.
 - Error due to drift during NRR measurement. (**Node 2 to Node 1**)
 - Error due to drift between measuring and using NRR. (**Node 2 to Node 1**)
 - Error due to drift during Residence Time measurement. (**Node 2 to GM**)
- Additional error from drift between RR(1) calculation, at Node 1, and use in calculating RR(2). (**Node 1 to GM**)
 - In the model the contribution from meanLinkDelay is ignored; only Residence Time is used.



New

Clock Drift Error – Relevant Intervals

4 Hops – 2nd Hop – NRR from Sync



↓ Sync

1. Error due to drift during NRR measurement. (**Node 2 to Node 1**)
2. Error due to drift between measuring and using NRR. (**Node 2 to Node 1**)
3. Error due to drift during Residence Time measurement. (**Node 2 to GM**)
4. Error due to drift between RR(1) calculation, at Node 1, and use in calculating RR(2). (**Node 1 to GM**)
 - In the model the contribution from meanLinkDelay is ignored; only Residence Time is used.

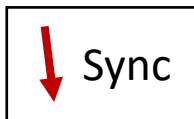
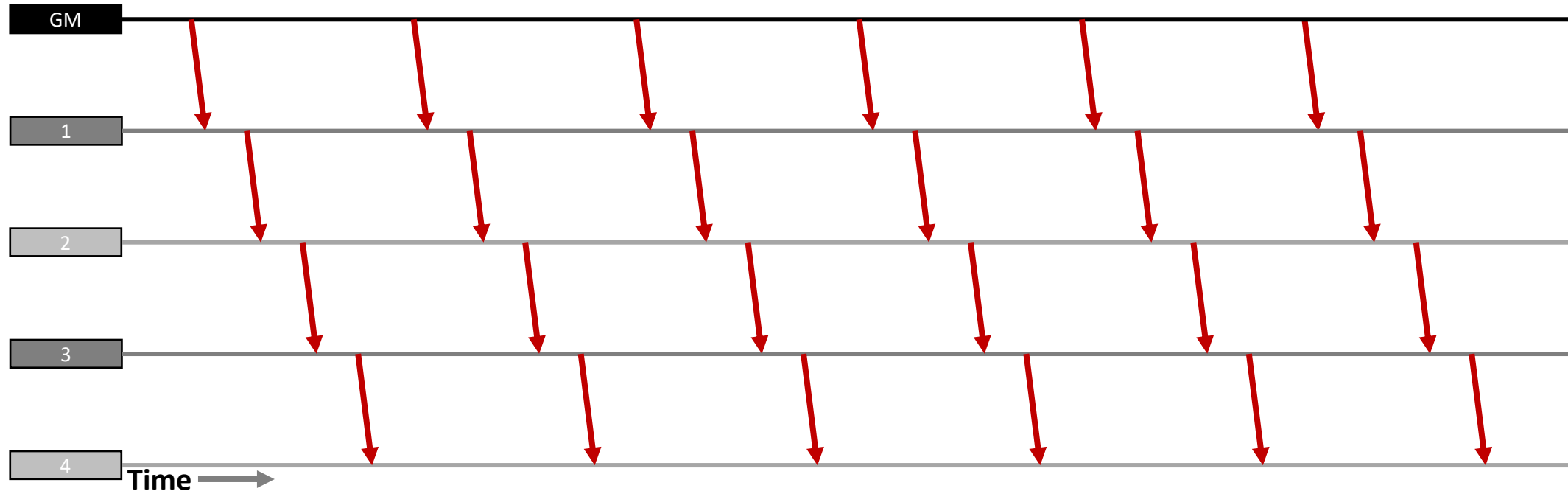
Implications

- Calculate RR and RRdriftRate on arrival of Sync message
 - Account for incoming RRdrift during meanLinkDelay
- Calculate correctionField based on RR half way between Sync TX at previous node and Sync TX at current node
 - Account for RRdrift from Sync RX
- Calculate RR for Sync TX accounting for RRdrift from Sync RX

Potential for Decreased Sync Interval

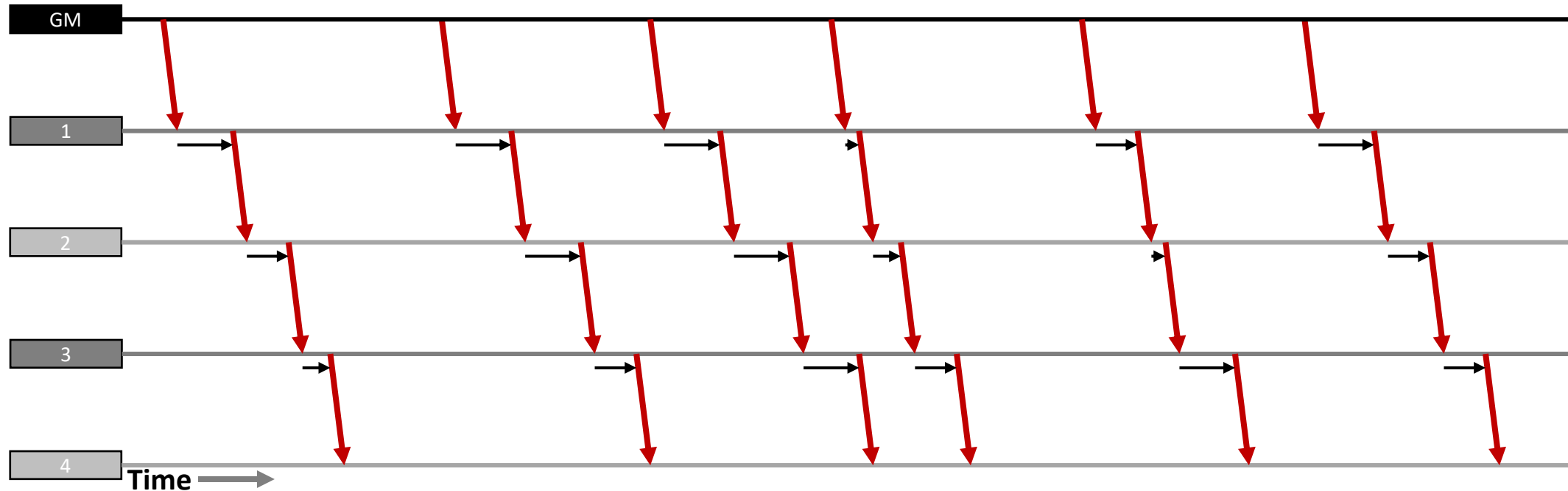
Sync Message Intervals – Idealised*

Consistent Sync Interval at GM; same Residence Time, every time (but...*Not to Scale)



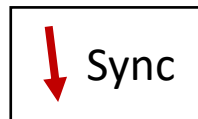
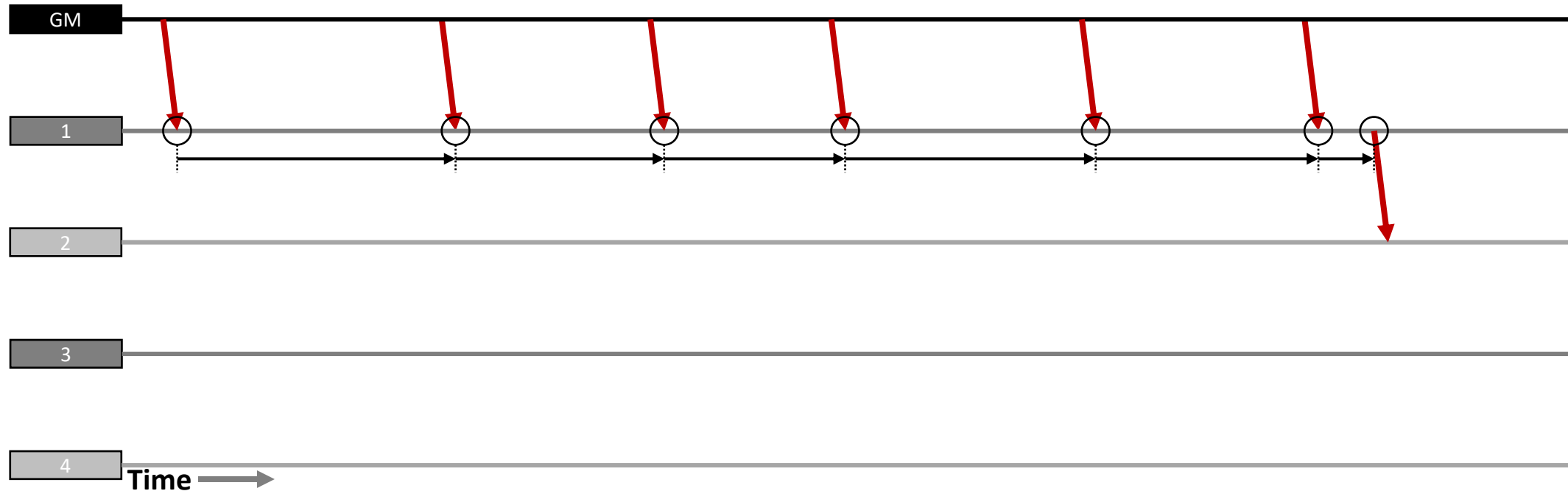
Sync Message Intervals – “Realistic”*

Variable Sync Interval at GM; variable Residence Times (but still... *Not to Scale)



We'll come back to the fact that some of the intervals between Sync messages are getting rather small.

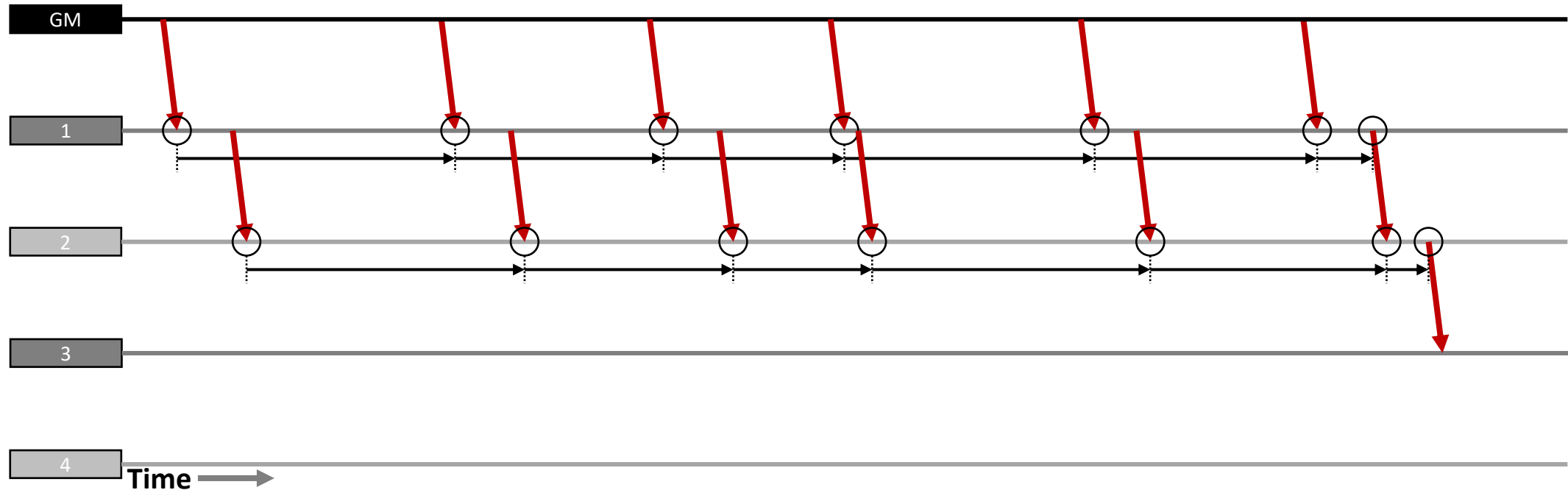
Relevant Intervals – Hop 1



Drift error calculations at Hop 1 are based on

- The Sync Intervals at Hop 1; most recent Residence Time.
- NRR Drift: Node 1 to GM. (Same as RR Drift: Node 1 to GM.)

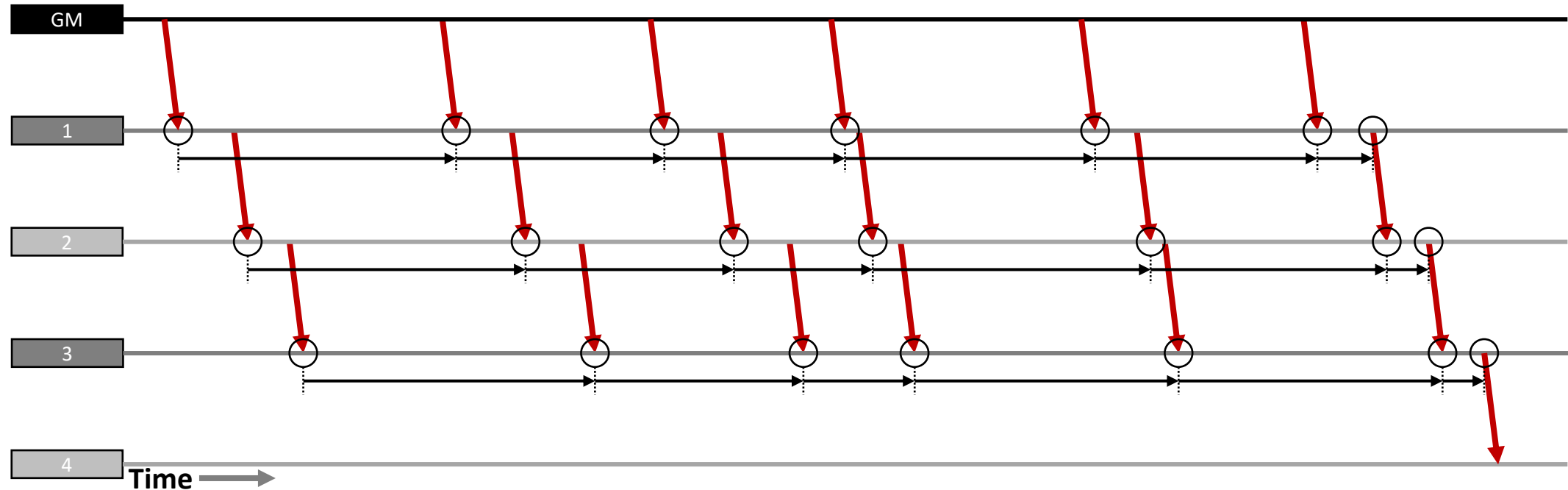
Relevant Intervals – Hop 2



Drift error calculations at Hop 2 are based on

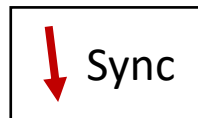
- The Sync Intervals at Hop 2 (not the same as Hop 1; add relevant Residence times); most recent Residence Time.
- NRR Drift: Node 2 to 1.
- RR Drift: Node 2 to GM. (Plus a bit of Node 1 to GM.)

Relevant Intervals – Hop 3

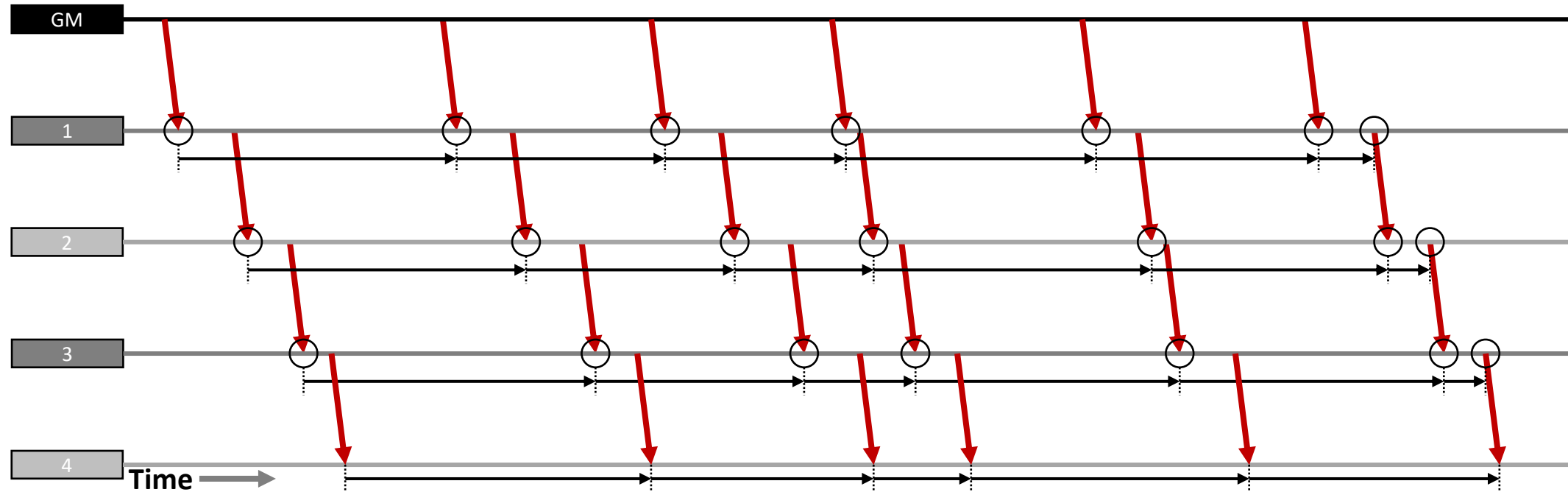


Drift error calculations at Hop 3 are based on

- The Sync Intervals at Hop 3 (not the same as Hop 2; add relevant Residence times); most recent Residence Time.
- NRR Drift: Node 3 to 2.
- RR Drift: Node 3 to GM. (Plus a bit of Node 2 to GM.)

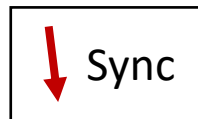


Relevant Intervals – Hop 4



Drift error calculations at Hop 4 are based on

- The Sync Intervals at Hop 4 (not the same as Hop 3; add relevant Residence times).
- NRR Drift: Node 4 to 3.
- RR Drift: Node 4 to GM. (Plus a bit of Node 2 to GM.)



Side Note...

- Note that RRdrift is part of every node calculation, so GM drift and consequent error is a part of every node calculation.
- This is another way of thinking about why GM drift is so pernicious.
 - *pernicious*
having a harmful effect, especially in a gradual or subtle way.

Simulations

- Sync Intervals can decrease and then become negative.
- Which also means that Sync Intervals can be zero...at least in the Monte Carlo simulation!
 - This breaks things.
- Time Series may not suffer from this problem.

- Thanks to Geoff Garner for working through some of the implications on this and following slides.

Real World

- Sync Intervals should never decrease beyond a certain point, especially with 2-step.
- State Machine will reset if a new Sync message is received before previous Sync message is fully processed.
- Even if implementation is bad and the state machine doesn't reset, timestamps will mean the next node won't be affected.
- But...

Real World – Small Sync Intervals

- Small Sync Intervals mean increased Timestamp Error
- Better to have a minimum time between Sync messages
- Should this be a normative requirement?

Simulations – Minimum Sync Interval

- Monte Carlo is currently configured to have a minimum of 1ms between Sync messages.

Thank you!

“Man with binoculars” icon is from [Icon Fonts](#) under CC BY 3 license.