60802 Time Sync – Sync Message Clashes Potential & Implications

David McCall – Intel Corporation

Version 2

March 2024

References

1 – David McCall “60802 Time Sync – RR & NRR Drift Tracking and Compensation – Implications Version 2”, IEC/IEEE 60802 contribution, June 2023

[2] – Geoff Garner, “Further 60802 Error Generation Time Series Simulation Results Version 2”, contribution to IEC/IEEE 60802 13th March 2024

Background

- [1] raised the possibility that Sync messages could "clash", i.e. variable Residence Times means that, over long chains of devices, some Sync intervals could shrink to the point where their processing overlaps.
- It was said at the time that some simulations to investigation into whether this is a real risk or only a theoretical one (for 60802's goal of 100 network hops). Also some analysis of what effect this might have on performance (e.g. calculation of NRR, because 60802 uses Sync messages – with the Drift Tracking TLV – to calculate NRR).
- This was low-priority as a cursory analysis indicated that the risk of any bad effects was low, but it remained an open item.
- This presentation is intended to close the topic.

Contents

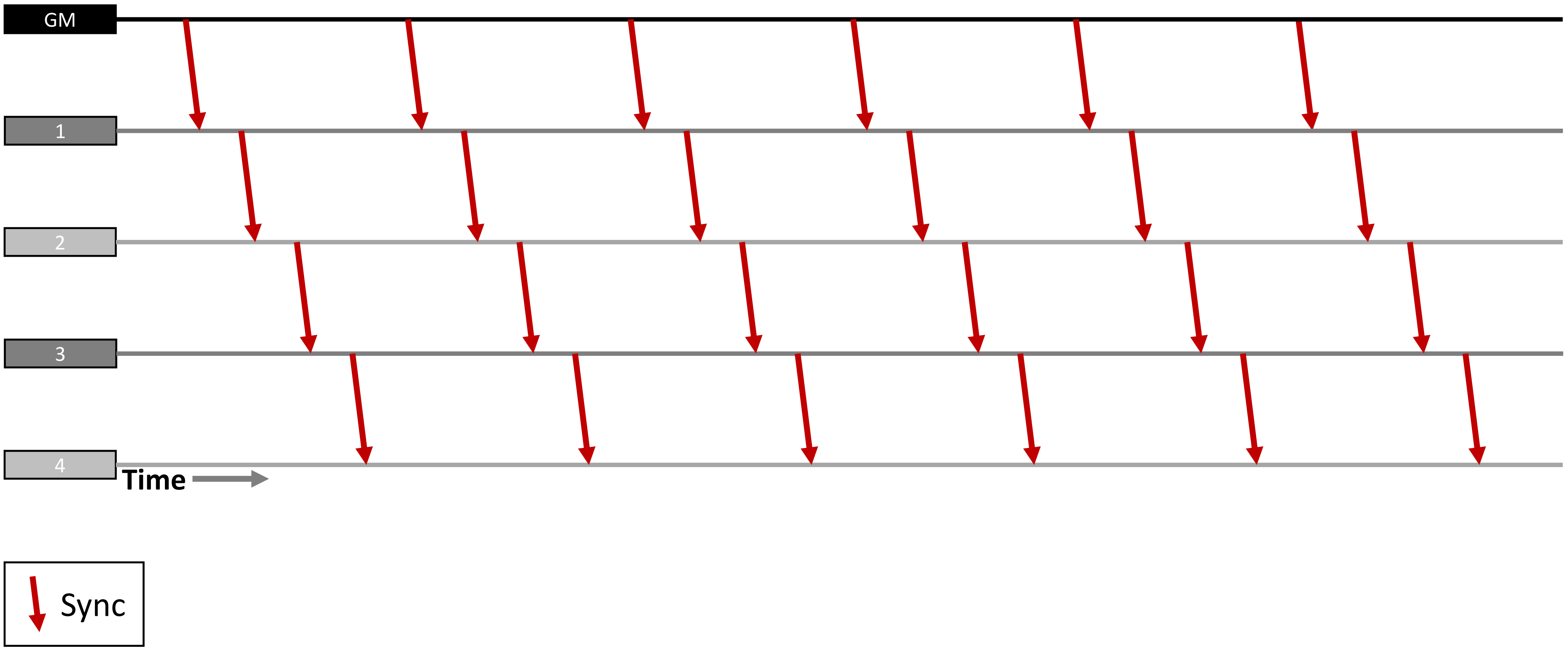
- Potential for decreasing Sync Interval (from [1])
- Sync Message Clashes – What Actually Happens?
- Simulation Description
- Simulation Results
- Conclusion

Potential for Decreased Sync Interval

From [1]

Sync Message Intervals – Idealised*

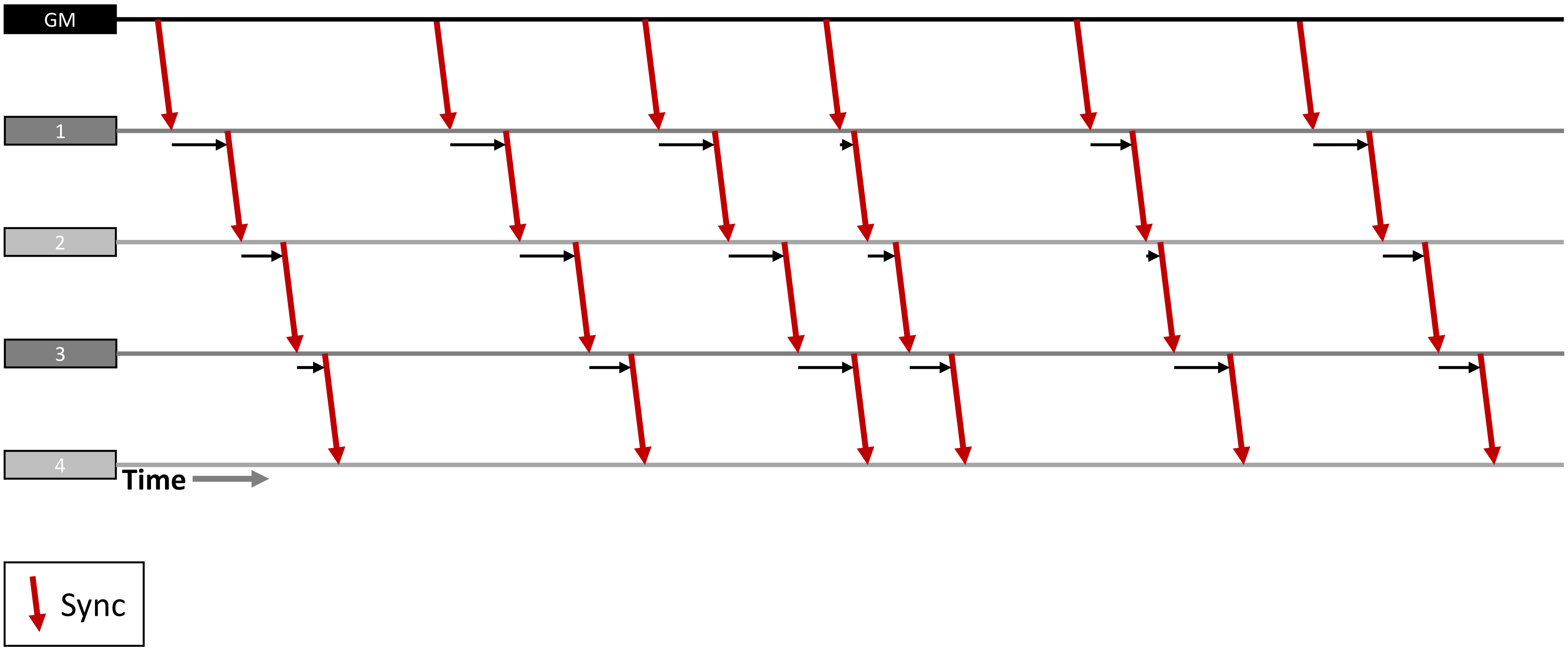
Consistent Sync Interval at GM; same Residence Time, every time (but…*Not to Scale)



From [1]

Sync Message Intervals – "Realistic"*

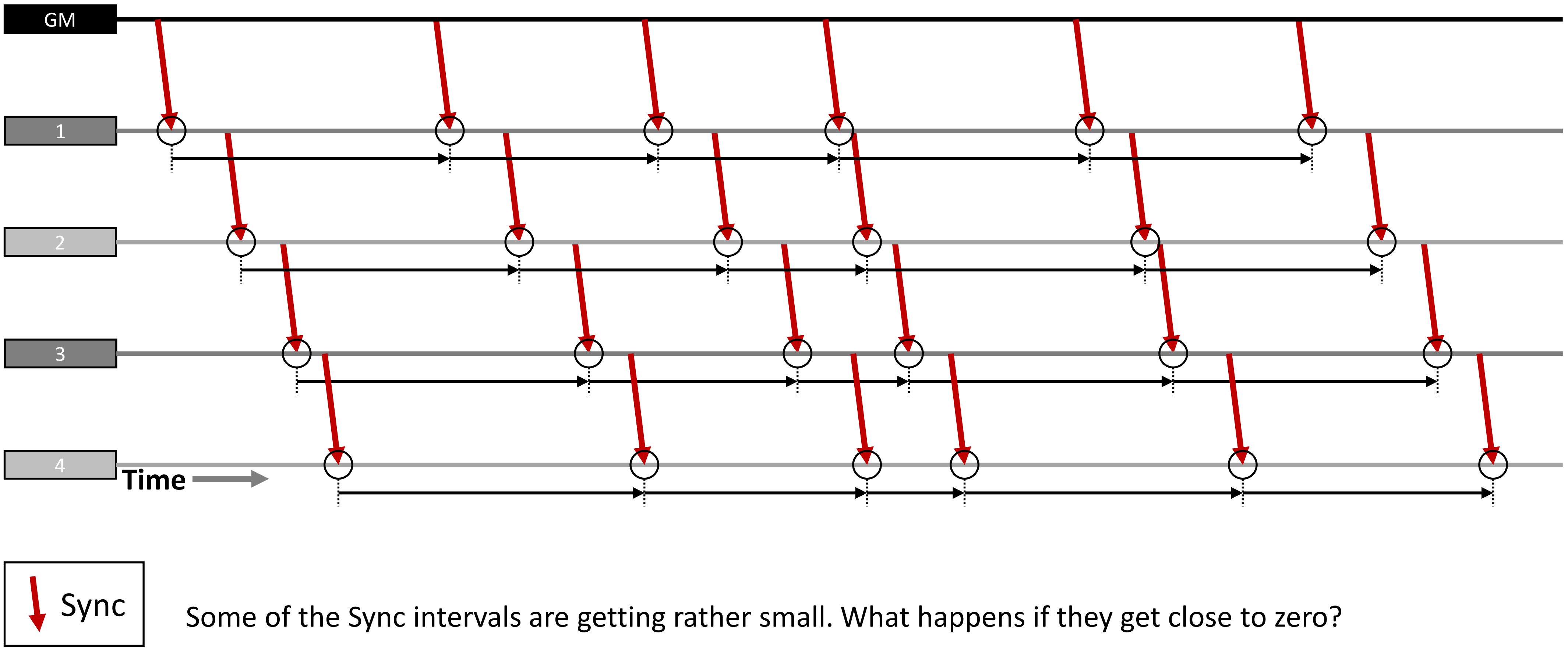
Variable Sync Interval at GM; variable Residence Times (but still…*Not to Scale)



From [1]

Sync Message Intervals – "Realistic"*

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



Some of the Sync intervals are getting rather small. What happens if they get close to zero?

From [1]

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.

From [1]

Real World

- Sync Intervals should never decrease beyond a certain point, especially with 2-step. **Further analysis → probably not true**
- 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…

From [1]

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?

Further analysis → risks are substantially mitigated by NRR drift tracking and smoothing algorithms (see D.5.2 & D.5.3) using data from past 2 x 16 (tracking) or 8 (smoothing) messages.

From [1]

Simulations – Minimum Sync Interval

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

Sync Message Clashes – What Actually Happens?

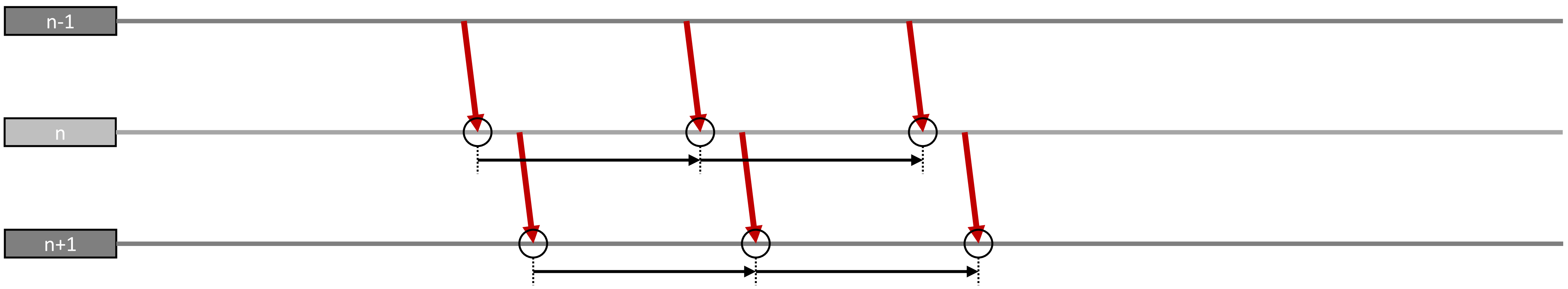
State Machines & Implemetations – 1

- The exact implementation of the state machines than handle Sync message (including Follow Up) may vary. The state machines in IEEE 802.1AS are silent on some aspects, and where they are not alternative implementations may deliver interoperability.
- Regarding Sync message clashes, there are a few different likely behaviours for 1-step. If a second incoming Sync message arrives before the consequent outgoing Sync for the first incoming message is sent the implementation might…
  - Not send the first outgoing Sync message
  - Send both outgoing Sync messages in strict order
  - Send both outgoing Sync messages but the order might vary

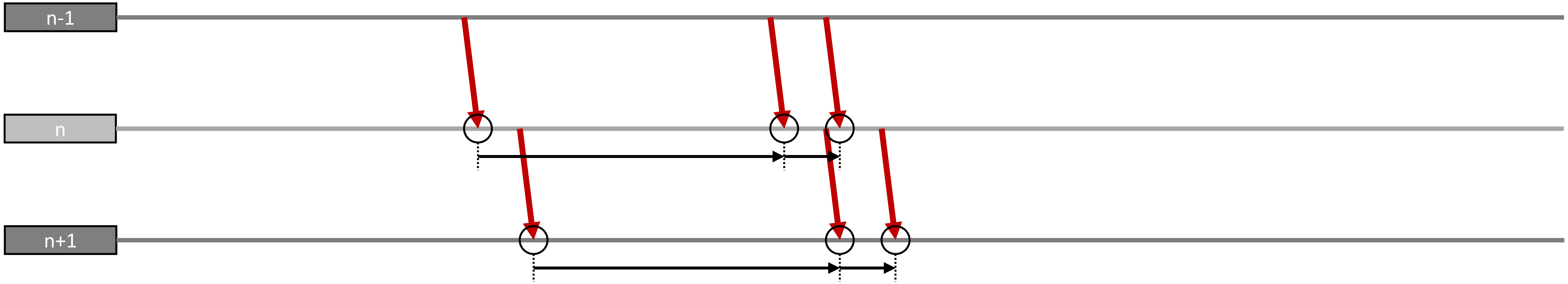
State Machines & Implemetations – 2

- For 2-step, there is the additional question of what happens on arrival of a second incoming Sync message when the first outgoing Sync message has been sent, but the first outgoing Follow Up message has not. The implementation might fail to send the first outgoing Follow Up message, or not.
- The riskiest implementations are those where a Clash means the relevant Sync message or Follow Up message is not sent. It is therefore this implementation that is illustrated in this section and on which the simulation is based.

Sync Message Clash – 1-step

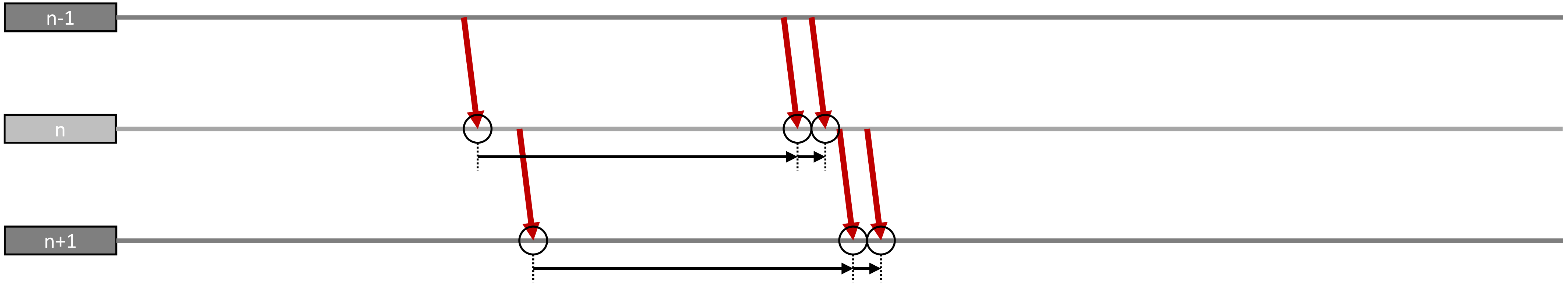


Sync Message Clash – 1-step



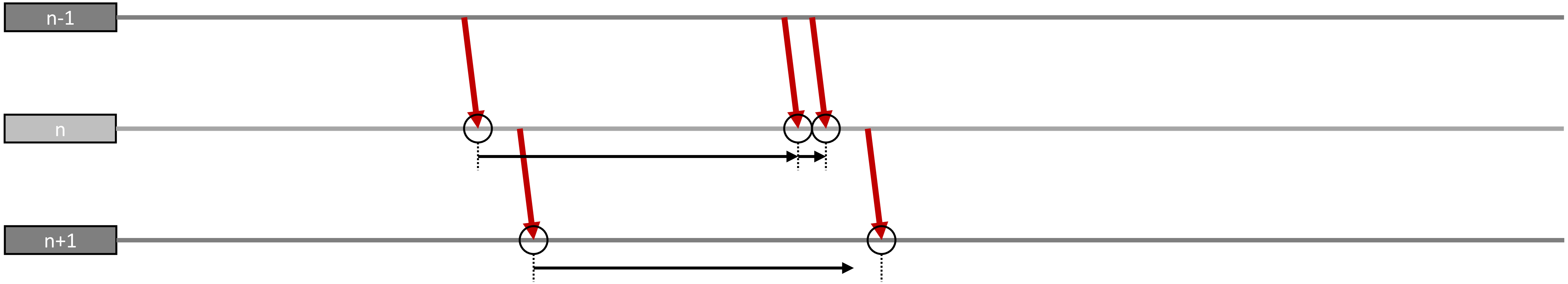
This is OK.

Sync Message Clash – 1-step



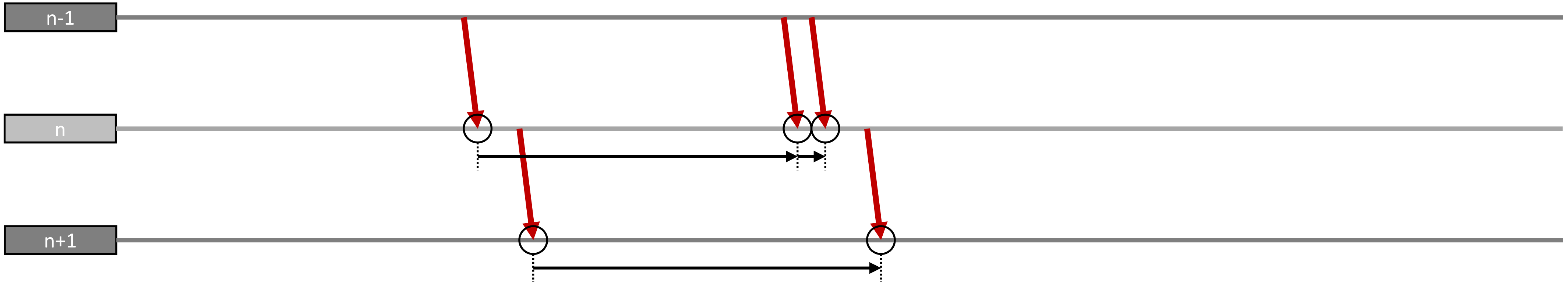
This is not OK. Node n receives a Sync message (C) before it has processed the previous Sync message (B).

Sync Message Clash – 1-step



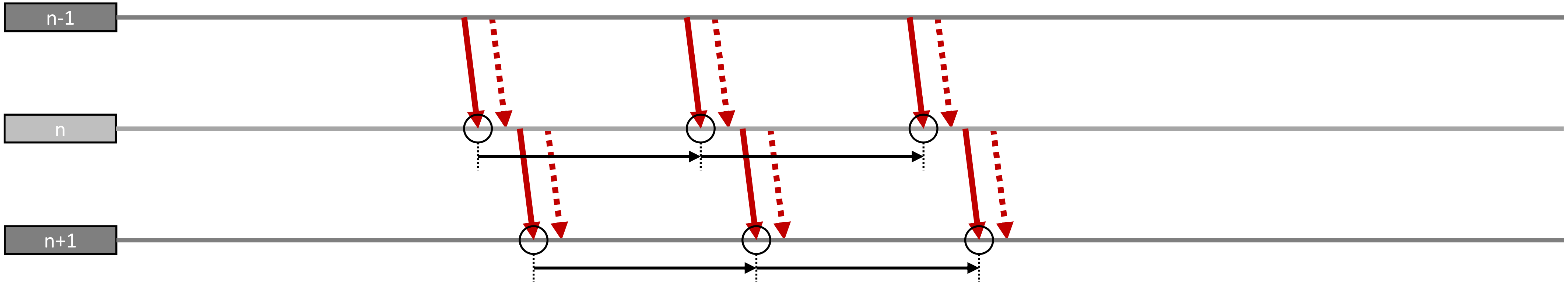
Sync message (B) is never sent, i.e. if Residence Time is greater than relevant Sync Interval.

Sync Message Clash – 1-step



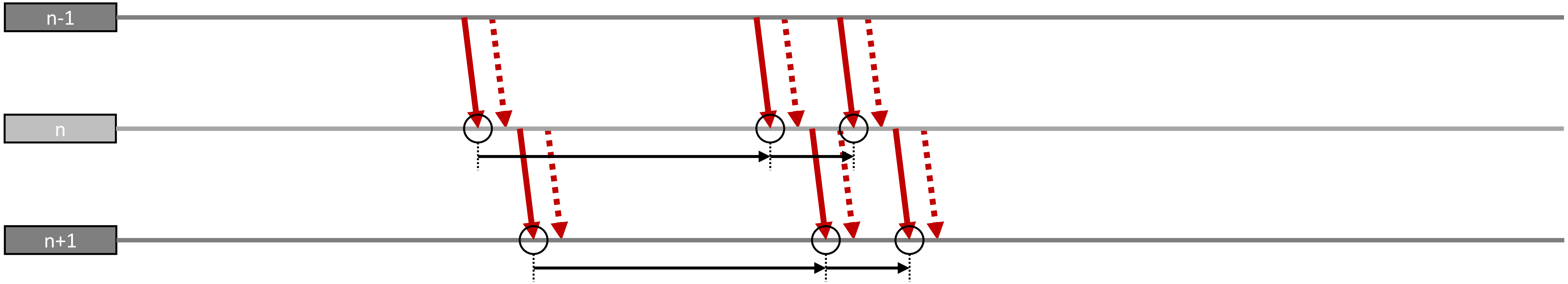
Sync message (B) effectively disappears and the Sync Intervals (A→B) and (B→C) combine into a single interval (A→C).

Sync Message Clash – 2-step



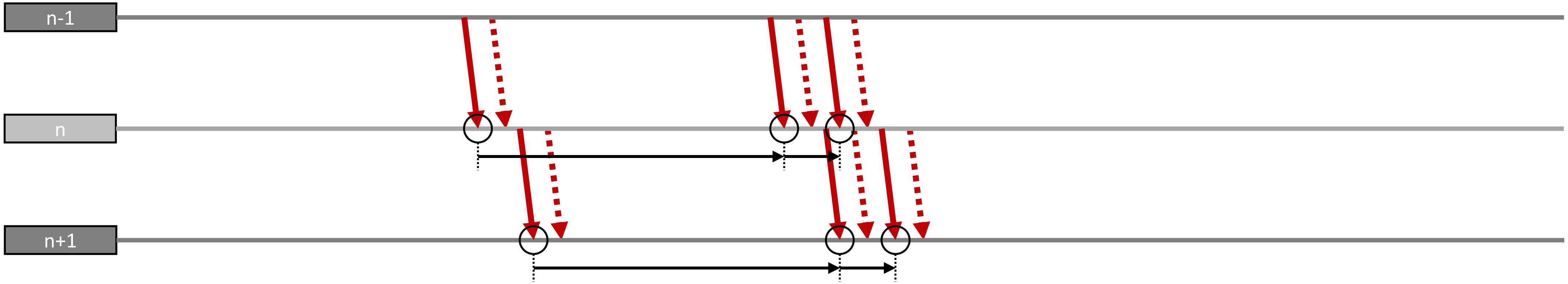
Very similar to 1-step, but Sync message processing at a node isn't complete until Follow_Up message is sent.

Sync Message Clash – 2-step



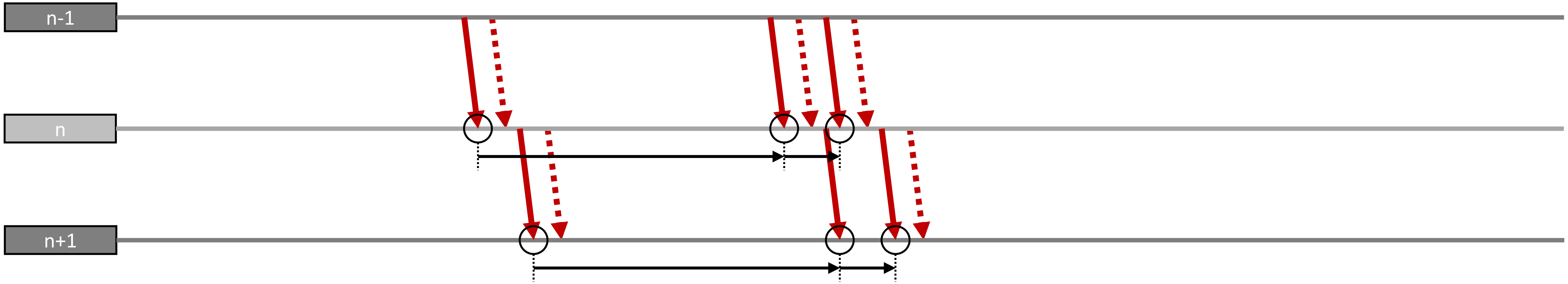
This is OK.

Sync Message Clash – 2-step



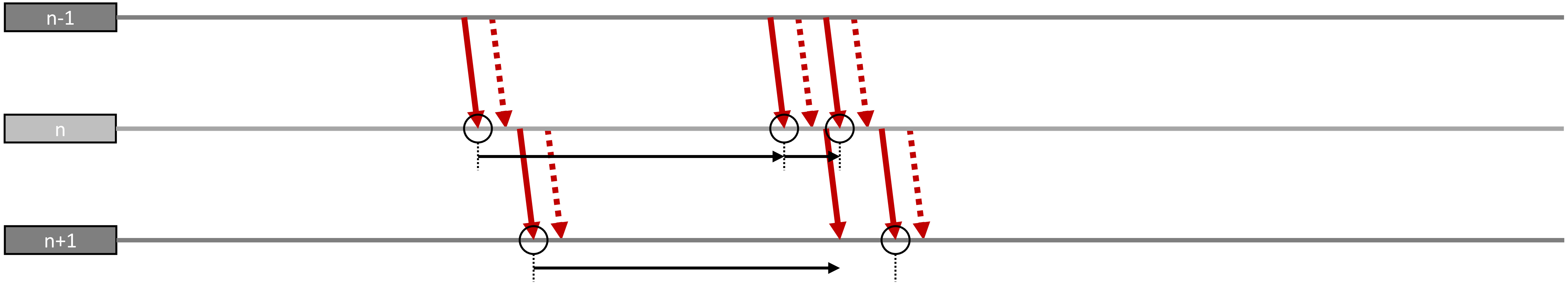
This is not OK. Node n receives a Sync message (C) before it has processed the previous Sync message (B).

Sync Message Clash – 2-step



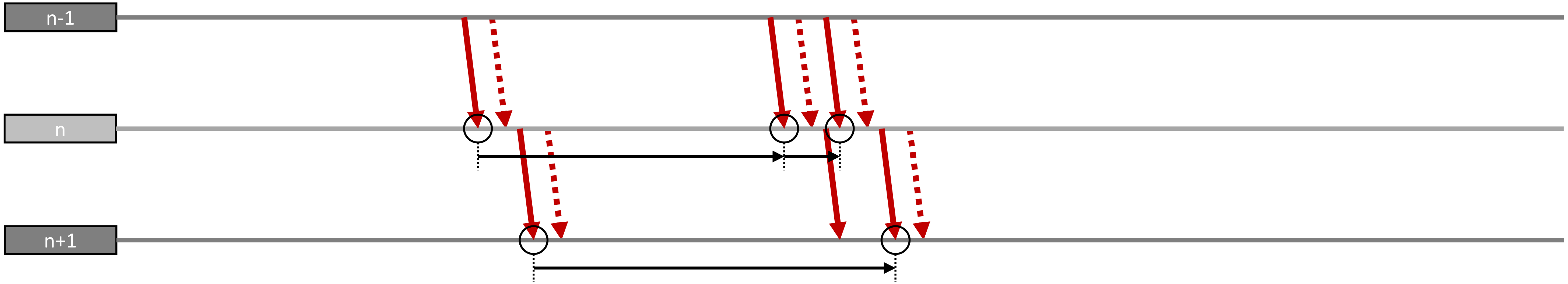
The Follow_Up message for Sync message (B) is not sent by Node n.

Sync Message Clash – 2-step



So, Node n+1 can not complete processing Sync message (B).

Sync Message Clash – 2-step



Again, Sync message (B) effectively disappears and the Sync Intervals (A→B) and (B→C) combine into a single interval (A→C).

Summary

- If there is a Sync message clash, a Sync message effectively disappears
  - 1-step: if Residence Time is greater than Sync Interval
  - 2-step: if Residence Time + Follow_Up Interval is greater than Sync Interval
- When a Sync message disappears the two Sync Intervals either side of it effectively merge.

Simulation Description

Simulation Description – 1

- RStudio Script for R
- Constructed only for this purpose
  - Only simulates a series of Sync messages (Sync Intervals, Residence Times, Follow Up Intervals)
  - Does not simulate time sync or Pdelay messages
  - Does not simulate timestamp errors or clock drift…or anything other than the simulation-time clock (the platonic ideal)
- Constructed to simulate very long timespans, as clashes are assumed to be relatively rare.
  - Simulates blocks of time in series
  - A single Replication contains multiple Runs over a number of Hops
    - Each Run is a single Sync Interval,
    - Typically 1,000,000 runs per replication, approximately 1.4 days of simulated time.
  - 1,000,000 runs over 100 hops takes a few minutes.
  - Multiple replications can deliver a year of simulated time in 2 to 3 hours.

Simulation Description – 2

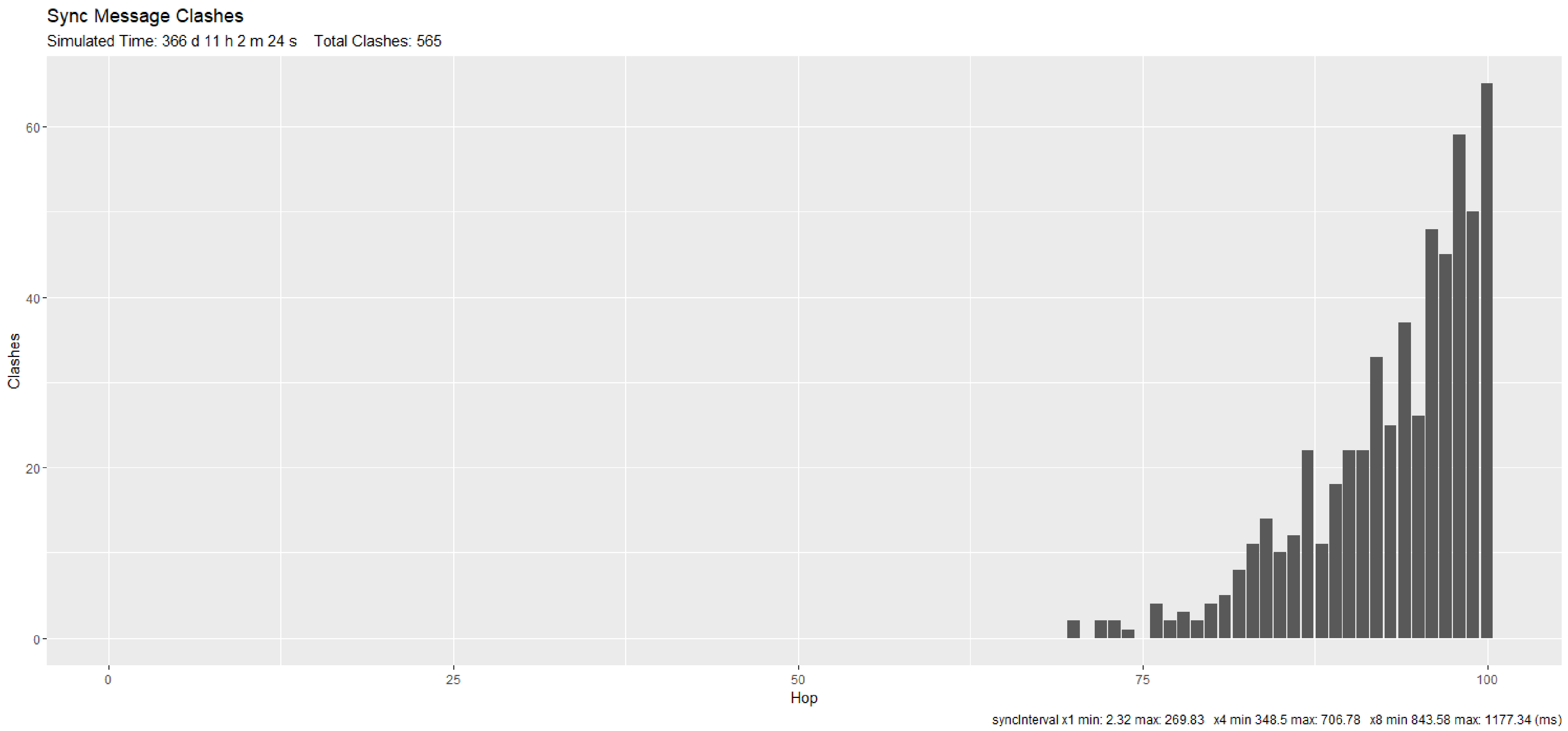
- For each replication…
  - Generates Sync Intervals (Runs)
  - For each Hop…
    - Generates Residence Times (and, for 2-step, Follow Up Intervals)
    - Checks to see if there is a clash
    - If there is a clash, register it (for that hop) and merge the Sync Intervals to either side
    - Adjust the Sync intervals for the next hop (based on Residence Times to either side)
  - At the last hop, find the min and max… (for this and prior replications)
    - Sync Interval
    - 4 x Sync Interval series
    - 8 x Sync Interval series (used for NA smoothing)
- When all replications are complete, plot results

Simulation Results

Simulation Parameters

- Same configuration and parameters as [2]
  - Sync Interval: Uniform distribution, 119 to 131 ms.
  - Residence Time: Normal Distribution, $\mu=5$, $\sigma=1.8$, truncated 1 to 15 ms
  - Follow Up Interval: 2.5 ms (maximum to increase chance of Clash)
- 2-step is more likely to generate Sync message clashes (due to Follow Up message processing time), so only simulated that.
- One year of simulated time.
  - Approx 252 million Sync messages.

Simulation Results



Summary

- One year of simulated time: 565 Sync message Clashes
  - First clash after 70 hops
- Sync Interval min and max results…

Across X Sync Intervals	Nominal (ms)	Min (ms)	Max (ms)
1	125	2.32 (2%)	269.83 (216%)
4	500	348.5 (70%)	706.78 (141%)
8	1,000	843.58 (84%)	1,177.34 (118%)

- Note that NRR smoothing uses 1 second of data (i.e. across 8 Sync intervals)

Conculsion

- We’re fine
- But only because we use NRR smoothing.
  - Would not recommend using an NA value of less than 4
  - Might want to add a note about this to the Informative Annex D

Thank you!