

# Synchronising Clocks in an Asynchronous Super-Computer

1 Knowing the time is important to all sorts of networks... [1]

Transport Internet

2 Unfortunately, no two clocks keep exactly the same time.

3 SpiNNaker is an asynchronous real-time neural simulator [2] which relies on its thousands of processors keeping time.

Here is how I do it...

4 Imagine it is the middle ages and every town must synchronise their town clock with the King's Palace Clock.

5 Each town sends a messenger on horse-back to read the palace clock, noting down the time when they left and returned.

6 When the messengers return, the town clocks are updated: [3]

$$\text{Town Clock} = \text{King's Time} + \frac{\text{Journey Time}}{2}$$

7

Day 0 Day 1 Day 2 Day 3

But, invariably, over time the town clocks drift apart.

8 And so the towns have to send out their messengers every day to read the palace time...

...which costs them the use of their messenger and horse!

9 Unfortunately the messengers are not completely reliable...

Ye Newe Pub

...so the clocks are still a little wrong.

10 The towns notice that they're adjusting their clocks by roughly the same amount each time... [4]

$$\Delta f = \frac{\text{average adjustment}}{\text{time between checks}}$$

...meaning that they don't always need to send their messengers.

11 Infrequent checking minimises the errors due to the messengers' unreliability but increases the impact of the clock's unreliability.

12 There's a sweet-spot for how often to send the messenger. We'll call it Allan, because that's its name. [1]

Error

Time Between Updates

13 So, back to the modern day...

Town Clocks = SpiNNaker Chips

Messengers = Network Packets

14

Experimental implementation keep 5,759 SpiNNaker chips' clocks within  $\pm 300$ ns.

15 I'm using these synchronised clocks to do performance tests on SpiNNaker's network.

## References

[1] David W. Allan. Time and frequency (time-domain) characterization, estimation, and prediction of precision clocks and oscillators. In *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 1987, Vol 34.6, p647-654.

[3] Gusella, Riccardo and Zatti, Stefano. The accuracy of the clock synchronization achieved by TEMPO in Berkeley UNIX 4.3 BSD. In *IEEE Transactions on Software Engineering*, 1989, Vol 15.7, p847-853.

[2] SB Furber, Steve Temple, and AD Brown. High-performance computing for systems of spiking neurons. In *AISB06 workshop on GC5: Architecture of Brain and Mind*, volume 2, pages 29-36, 2006.

[4] Mills, David L. Improved algorithms for synchronizing computer network clocks. In *IEEE/ACM Transactions on Networking (TON)*, 1995, Vol 3.3, p245-254.