Using Weighted Voting to Optimise Streamlined Blockchain Consensus Algorithms

TU Delft Delft University of Technology

Author:
Diana Micloiu
d.micloiu@student.tudelft.nl

Responsible Professor:
Jérémie Decouchant

Supervisor:
Rowdy Chotkan

1. Introduction

Consensus - collective agreement of network participants to ensure proper functionality of distributed systems.

Weighted voting - in the consensus mechanism, the voting power of a node depends on a weight metric.

Byzantine Fault Tolerant (BFT) protocol - requires $3f + 1$ nodes in a distributed system to withstand f node failures.

Streamlined consensus algorithms - new leader in each protocol view.

2. Background

Hotstuff [1]:

- **Streamlined algorithm** comprising 5 communication phases.
- $O(n)$ communication complexity.
- Basic Hotstuff - nodes vote on a single block per view.
- Chained Hotstuff - enable a pipelined voting mechanism to simultaneously progress on several blocks per view.

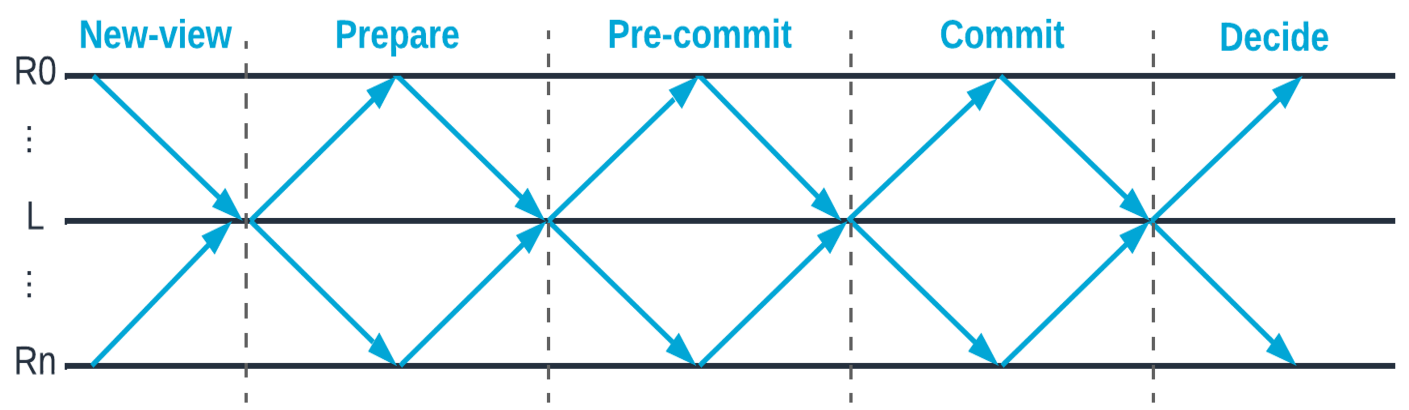

Figure 1: Hotstuff communication phases.

AWARE (Adaptive Wide-Area REplication) [2]:

- Uses Δ additional replicas and develops a deterministic, self-monitoring and self-optimising algorithm for improving the latency of the blockchain.
- Combines **BFT-SMaRt** (enhanced PBFT) [3] as replication protocol and **WHEAT** [4] for the underlying weighting distribution scheme ($Vmax = 1 + \frac{\Delta}{f}$ or $Vmin = 1$ voting power of each replica).
- Self-monitoring - uses a deterministic latency prediction.
- Self-optimisation - employs voting weights tuning and leader relocation mechanisms.

3. Scientific Gap

The impact of **weighted voting** has been applied so far only on **PBFT** in AWARE [2].

This research aims to address the benefits of **weighted voting on streamlined algorithms in terms of** latency reduction by studying the representative Hotstuff [1].

This research also highlights the effectiveness of using a **generalised continuous weighting scheme** (rather than the discrete one) for **optimising the recovery performance of the system.**

5. Results

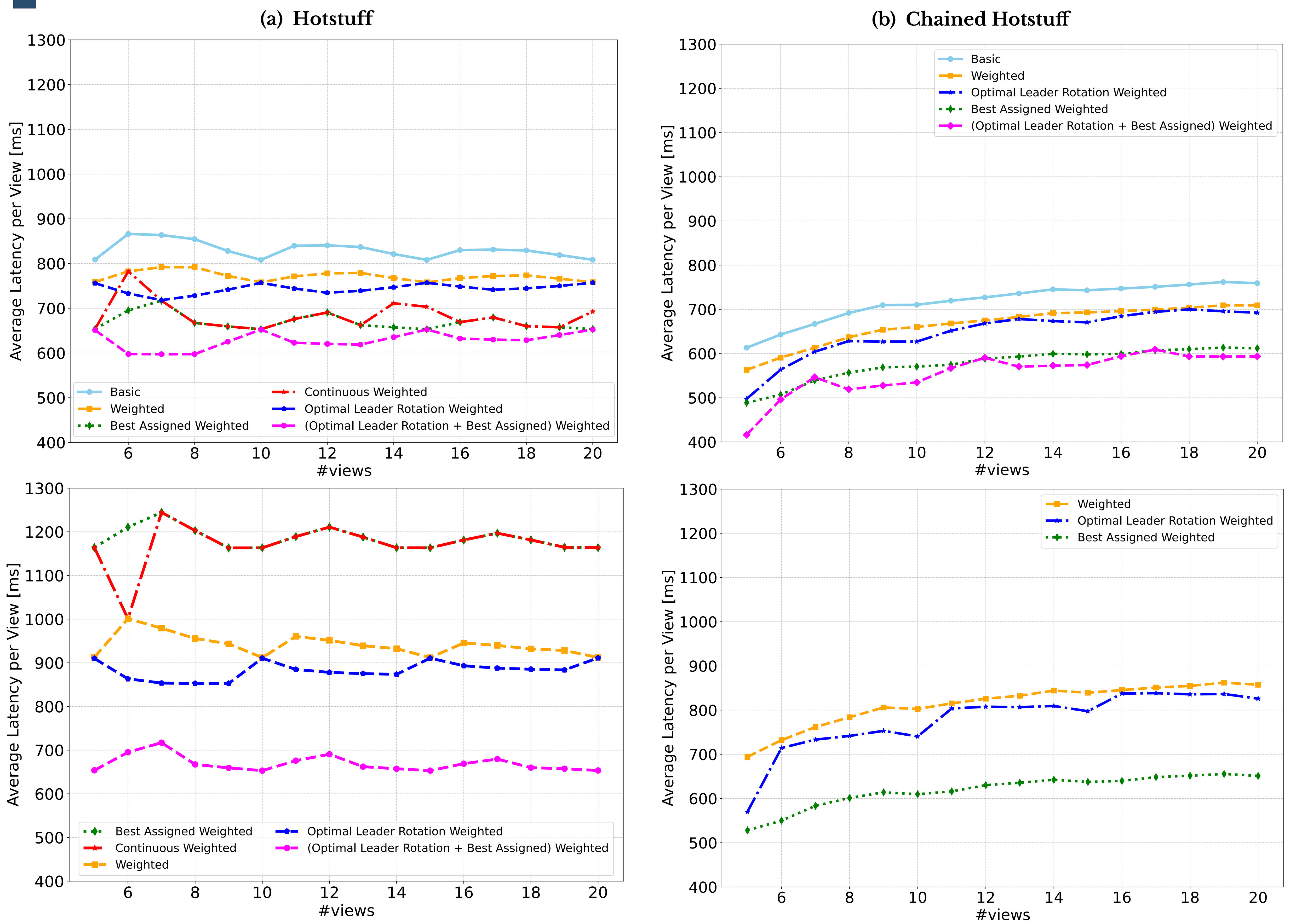

Figure 2: Average latency per view of protocol variants with $f = 1$ and $\Delta = 1$ (**top Figs., non−faulty scenario – all nodes behave normally, and bottom Figs., faulty one – f nodes holding highest weights are considered idle).**

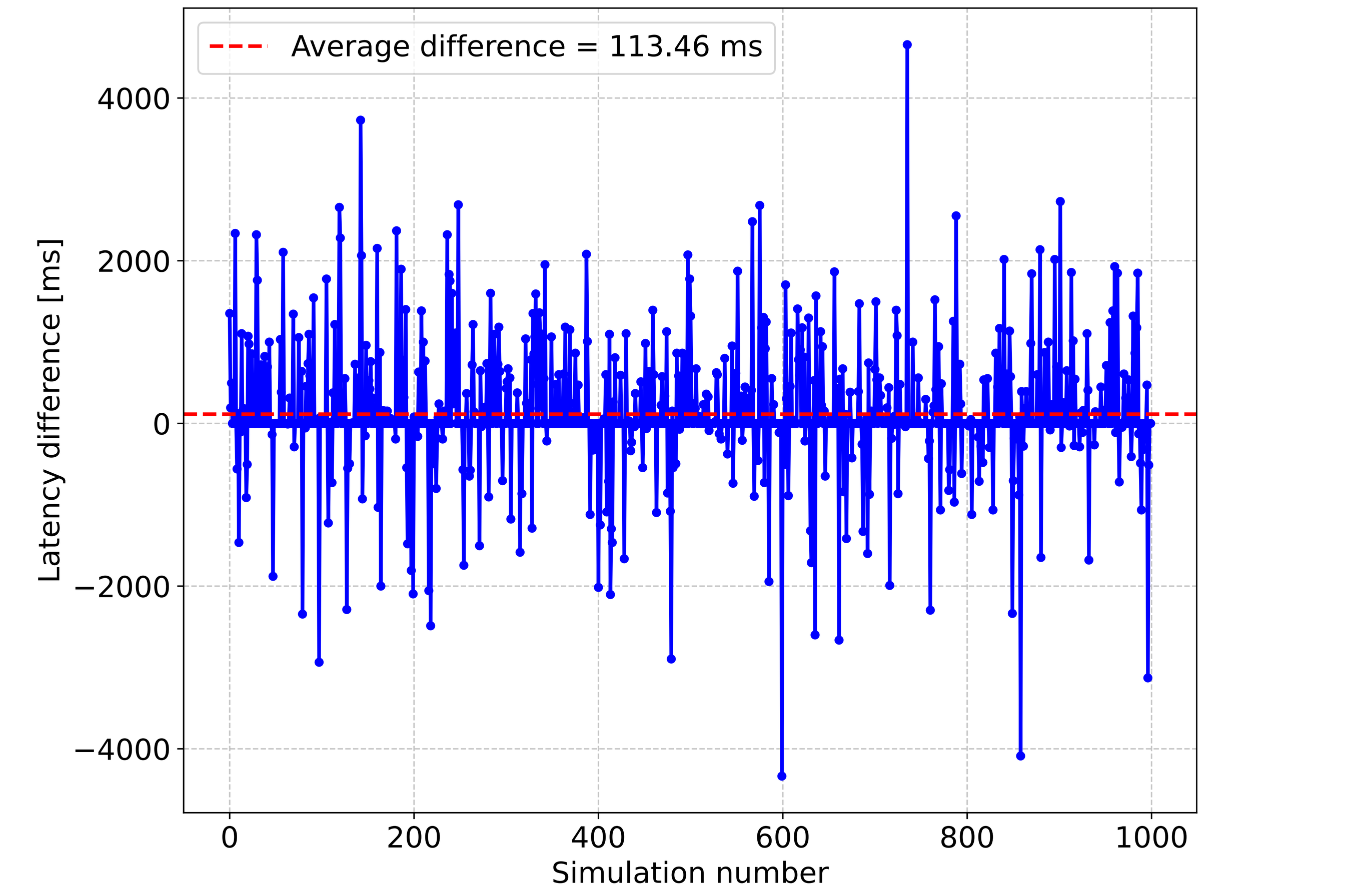

Figure 3: Difference in latency performance between Best Assigned and Continuous Weighted Hotstuff variants for 1000 faulty scenario simulations, $f = 1, \Delta = 1$, 10 views executed.

4. Methodology

1. **Latency prediction models:**
   - Two models - for the **Basic Hotstuff** and **Chained Hotstuff**, respectively.
   - Developed based on the deterministic latency prediction method used in AWARE for self-monitoring.
   - Emulate the streamlined algorithm behaviour combined with weighted voting.
   - Estimate the latency of a protocol run given the set of weights, network scenario and number of views .
2. **Simulated Annealing:**
   We use this metaheuristic method to evaluate the impact of the following possible optimisations on the Weighted blockchain protocols:

   I. **Best Assigned** - assign the highest $Vmax$ weight to the best performing $2f$ replicas.
   II. **Optimal Leader Rotation** - find the best succession of leaders to minimise the overall latency.
   III. **Optimal Leader Rotation + Best Assigned** - combine the two optimisation methods described in I and II.
   IV. Continuous (applied only to Hotstuff) - find a continuous weighting scheme that achieves lowest latency in both faulty and non-faulty scenarios.

Note that the continuous weighting scheme is not limited to the streamlined algorithms but can be applied to any blockchain consensus one.

6. Limitations

- The latency prediction models use same weights and network setting in all views of a protocol run.
- Simulated Annealing algorithms are impractical for $n > 15$.
- Continuous Weighted Hotstuff is infeasible for $f > 4$ due to the high computational complexity of the required quorum safety checks.

7. Conclusion

- Only applying weighted voting to Hotstuff and Chained Hotstuff decreases latency by 7%.
- Optimal Leader Rotation + Best Assigned optimisation reduces latency by almost 25%.
- Continuous Weighted Hotstuff performs equally well or better than Best Assigned one in 85% of simulations.

This research represents a founding base for the study of weighted voting in streamlined algorithms and its shift from the discrete model.

References

[1] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "Hotstuff: Bft consensus with linearity and responsiveness," in Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, 2019, pp. 347–356.
[2] C. Berger, H. P. Reiser, J. Sousa, and A. Bessani, "Aware: Adaptive wide-area replication for fast and resilient byzantine consensus," IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 3, pp. 1605–1620, 2020.
[3] A. Bessani, J. Sousa, and E. E. Alchieri, "State machine replication for the masses with bft-smart," in 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. IEEE, 2014, pp. 355–362.
[4] J. Sousa and A. Bessani, "Separating the wheat from the chaff: An empirical design for geo-replicated state machines," in 2015 IEEE 34th Symposium on Reliable Distributed Systems (SRDS). IEEE, 2015, pp. 146–155