A Review of Federated Byzantine Agreement Consensus Algorithms

The Case of Ripple

Klitos Christodoulou, *Ph.D.*
Email: christodoulou.kl@unic.ac.cy

Acknowledgements

This research is funded by the Ripple's Impact Fund, an advised fund of Silicon Valley Community Foundation (Grant id: 2018-188546).

Fundamental Challenges

- From Distributed Systems
  - Federated nodes should be able to communicate via *message-passing*
  - Agree with finality on some piece of information or action
  - Tolerate network failures and adversarial actors

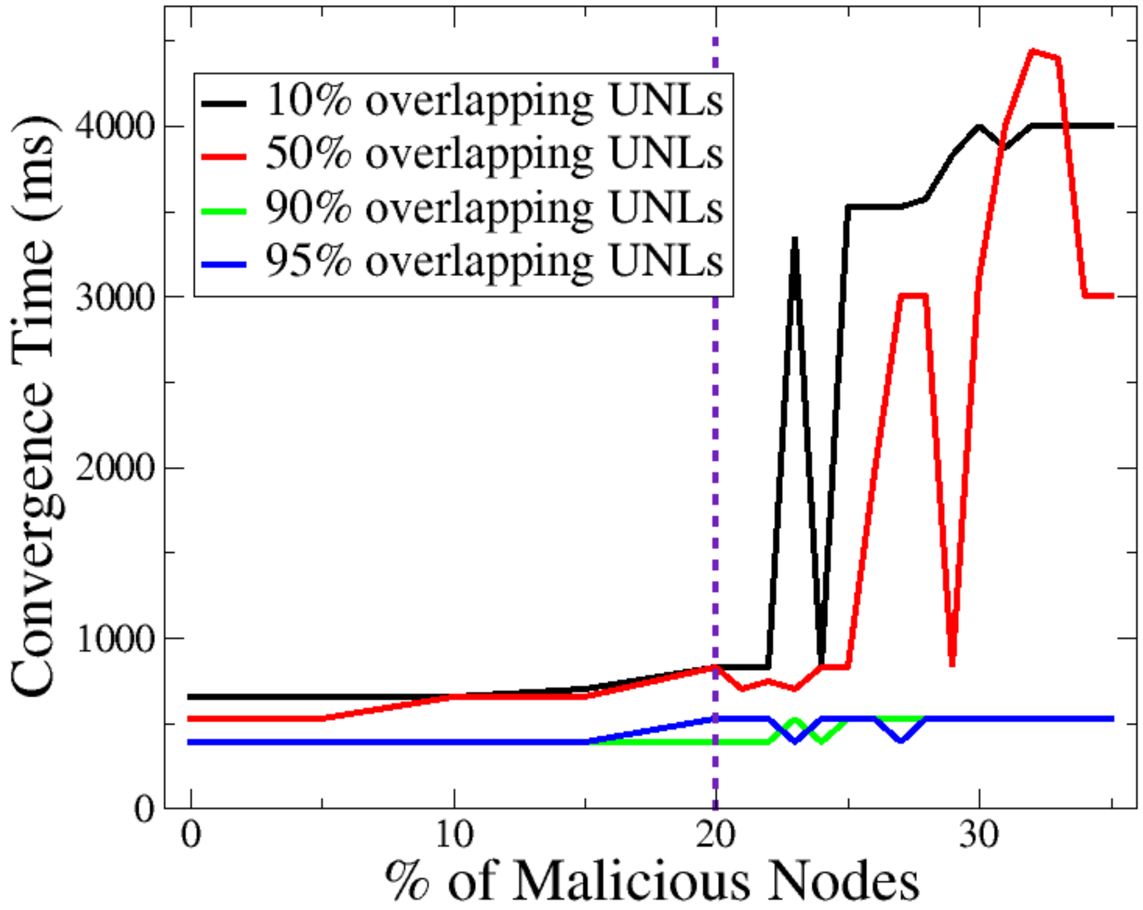
- We suggest that consensus algorithms engineering for distributed-ledgers are expected to consider the following questions:
  - How decentralized a system should be?, and
  - What is the role of the consensus algorithm to maintain an equilibrium between performance, robustness, and decentralization?

Experimental Scenario 1: RPCA

- **Experimental Setup**
  - **Consensus used:** Ripple Protocol Consensus Algorithm (RPCA) as described in [1]
  - Fixed network latency
  - # of nodes for the simulation is 1000
  - Size of UNL: Random size derived from a Uniform Distribution
  - **Overlapping UNLs**: nodes share a varying portion of the UNL list
- **Experimental Purpose**
  - To understand the parameters of RPCA
  - To understand the behavior of RPCA when boundaries are reached in terms of:
    - Unique Node List (UNL) overlapping
    - % of Malicious Nodes
    - Convergence time

https://github.com/UNIC-IFF/ripple-simulator-old/tree/correctness_additions

Experimental Scenario 1: RPCA



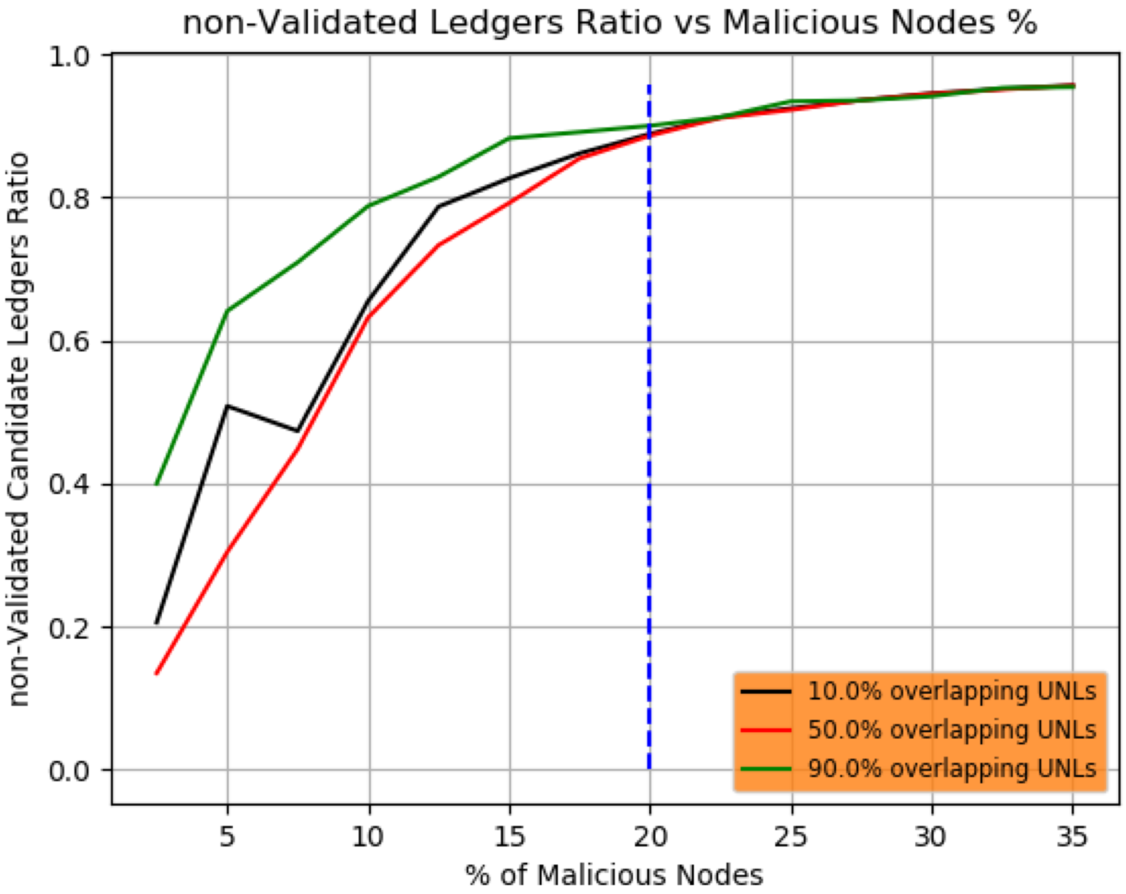
- **Observations**
  - Two broad operational regions
  - R1: Convergence times <=1000ms for any % UNL overlap
  - R2: less robust consensus as UNL overlap decreases. Only 90%, 95% UNL overlap maintains <1000ms
- **Findings**
  - For low % of malicious nodes decentralization degree (from UNL % overlap) can be significantly relaxed
  - Strong UNL % overlap is needed when % of malicious nodes exceeds a significant threshold (weak correctness boundary)

Experimental Scenario 2: Random Attack

- **Experimental Setup**
  - XRP Ledger Consensus Protocol (LCP) as described in [2]
  - **Simulate a Random attack**
    - On time *t,* a randomly selected byzantine node injects an *arbitrary* transaction to its accepted/open ledger
    - Simulation time 2mins
    - Rate of transactions 100tx/s (non-malicious nodes)
    - Rate of malicious injections 100tx/s (byzantine nodes)
    - # of Byzantines nodes [2.5%, 35%] of total. We maintain the % of malicious nodes in the overlapping UNLs.
    - **Overlapping UNLs**: nodes share a varying portion of the UNL list

- **Experimental Purpose**
  - To understand the behavior of XRP LCP when boundaries are reached in terms of:
    - Unique Node List (UNL) overlapping
    - % of Malicious Nodes

Experimental Scenario 2: Random Attack



- We define the ratio:

$$\frac{\#CandidateLedgers - \#FullyValidatedLedgers}{\#CandidateLedgers}$$

- **Observation**
  - A metric to be used as a Network Health Indicator
  - Detection for independent malicious nodes

Key Findings from Experimenting with Ripple's simulator

- Findings that relate with the decentralization degree of the Network
- What are the conditions of decentralization?
- Future Work:
  - Indicators for Malicious nodes and dynamic update of the UNL list
  - Improving malicious defensive actions

Thank you!

Klitos Christodoulou, *Ph.D.*
Email: christodoulou.kl@unic.ac.cy

UNIVERSITY of NICOSIA
ripple

References

[1] Schwartz, David, Noah Youngs, and Arthur Britto. "The ripple protocol consensus algorithm." *Ripple Labs Inc White Paper* 5 (2014): 8.

[2] Chase, Brad, and Ethan MacBrough. "Analysis of the XRP Ledger consensus protocol." *arXiv preprint arXiv:1802.07242* (2018).

[3] MacBrough, Ethan. "Cobalt: BFT governance in open networks." *arXiv preprint arXiv:1802.07240* (2018).