BRAIN 2021

An Analysis of Multi-hop Iterative Approximate Byzantine Consensus with Local Communication

Matthew Ding

matthewding@berkeley.edu

5 September 2021

The Byzantine Generals Problem

- Lamport, Shostak, and Pease (1982)
- Coined the term "byzantine fault"
- Very strict assumption, but sometimes necessary in real life, e.g. blockchain

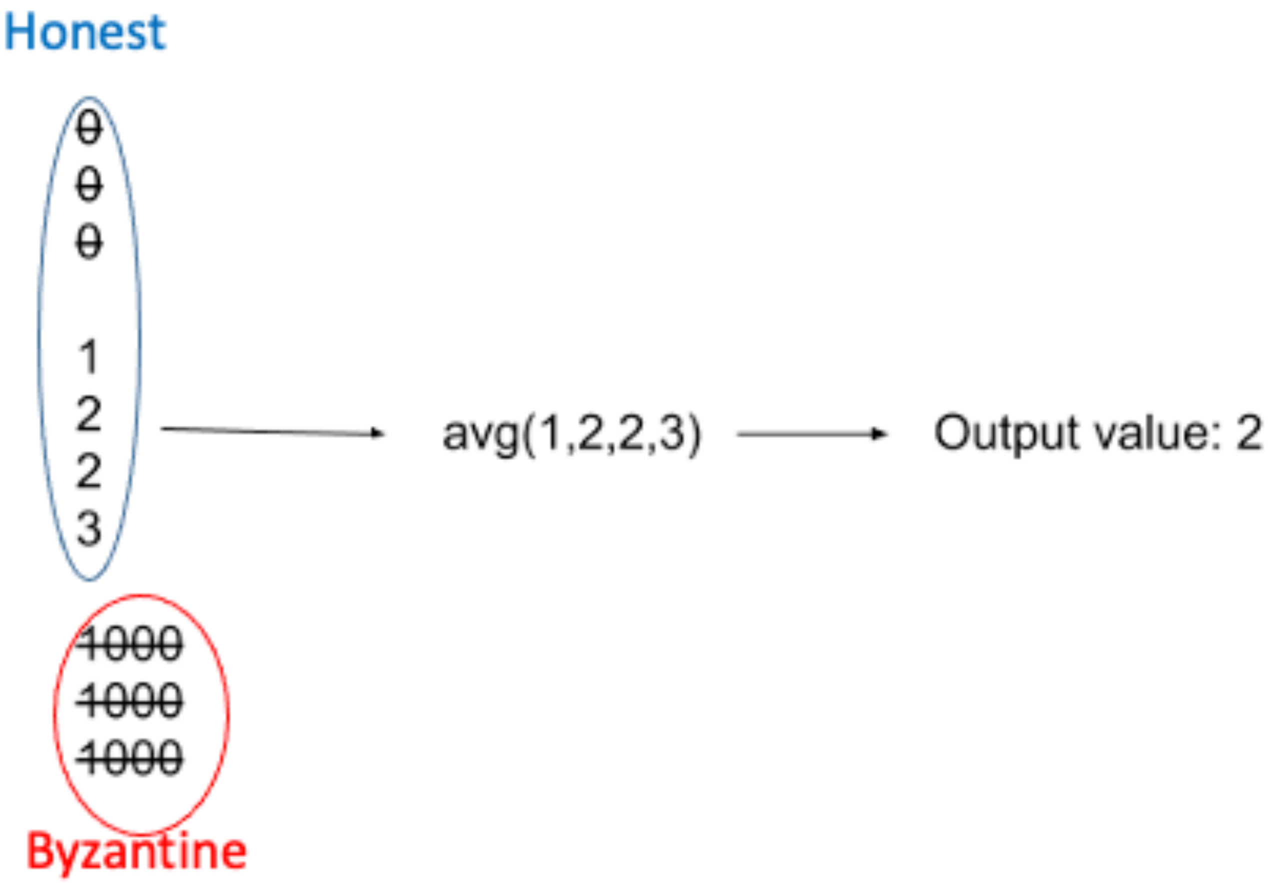
Iterative Approximate Byzantine Consensus (IABC)

- Approximate consensus rather than exact consensus
- Aim to satisfy two conditions:
  1. Convergence
  2. Validity

Trimmed-Mean Step

- Given a list of at least 3f+1 values:
    1. Eliminate the greatest and least f values
    2. Output the arithmetic mean of the remaining values

- This is a robust aggregation step for up to f byzantine nodes



Existing IABC Algorithm

- Vaidya (2012)
- Transmits current state to all neighbors
- Perform a trimmed-mean step to determine new state

Our Contributions

- Signatures
  - Reliable proof of who created a message
- Relays
  - Using signatures, we can now reliably relay messages across a graph

Our Contributions (continued)

- All honest nodes may send and receive messages to every other honest node
- Our algorithm creates a “pseudo-complete” graph in order to increase the efficiency of communication

Relay-IABC Algorithm

3.4 Relay-IABC Algorithm

Algorithm 1: Relay-IABC

Remark. *This algorithm is implemented by a specific machine i. Each machine $i \in H$ will implement this algorithm concurrently.*

Result: Each state $v_i(i)$ remains within the convex hull of the initial states at each Iteration, and each state converges to the same value as Iteration $t \to \infty$.

Initialization:

$v_i(i) \leftarrow$ Intial State of node i (with signature i).

for *Iteration* $t \leftarrow 0$ **to** T **do**

Broadcast v_i to all machines $j \in N_i^O$

Receive v_j from all machines $j \in N_i^I$

Remark. *When receiving v_j, ignore all parameters received that are not properly signed. If no proper message is received from a certain node, set their incoming value to be an arbitrary predefined real value (e.g. 0).*

$G_i \leftarrow N_i^O \cup \{i\}$

for $j \leftarrow 0$ **to** $m - 1$ **do**

Remark. *In the next two lines, we do the following: Out of all parameters $v(j)$ received from the broadcast step, set $v_i(j)$ to a single arbitrary one $v'(j)$*

if $j \neq i$ **then**

$v_i(j) \leftarrow v'(j)$

end

end

if $t \mod D = 0$ **then**

Trimmed-mean update step:

In a new vector, sort the values of v_i in increasing order:

$$v_i^* \leftarrow sort(v_i) \tag{1}$$

Ignore the least and greatest b values, and set the value of $v_i(i)$ to be the average of all remaining values in v_i^*, as defined below:

$$v_i(i) \leftarrow \frac{1}{m - 2b} \sum_{k=b}^{m-b-1} v_i^*(k) \tag{2}$$

Add signature i to $v_i(i)$

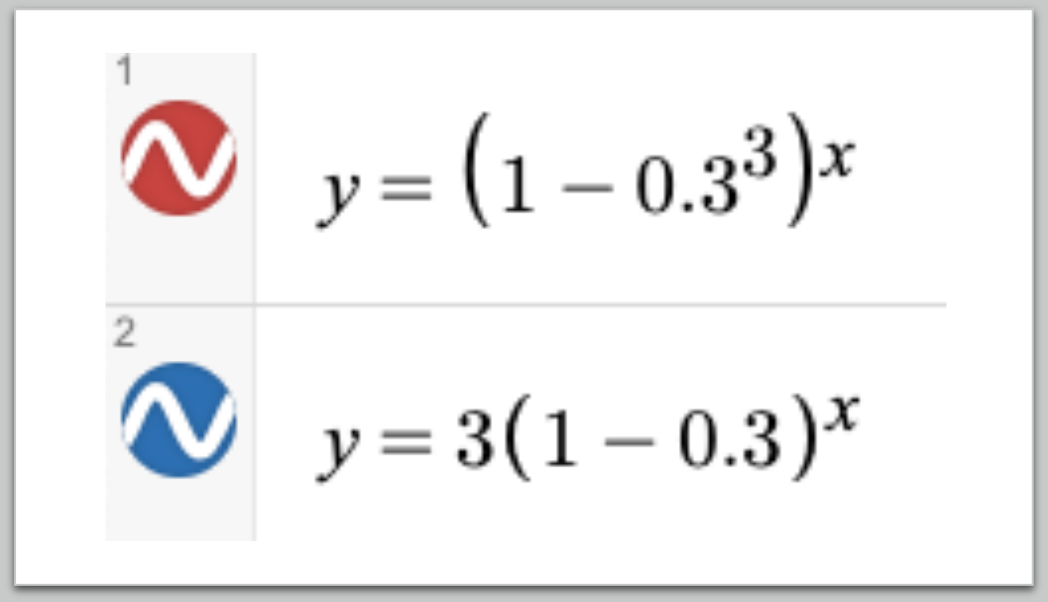
end

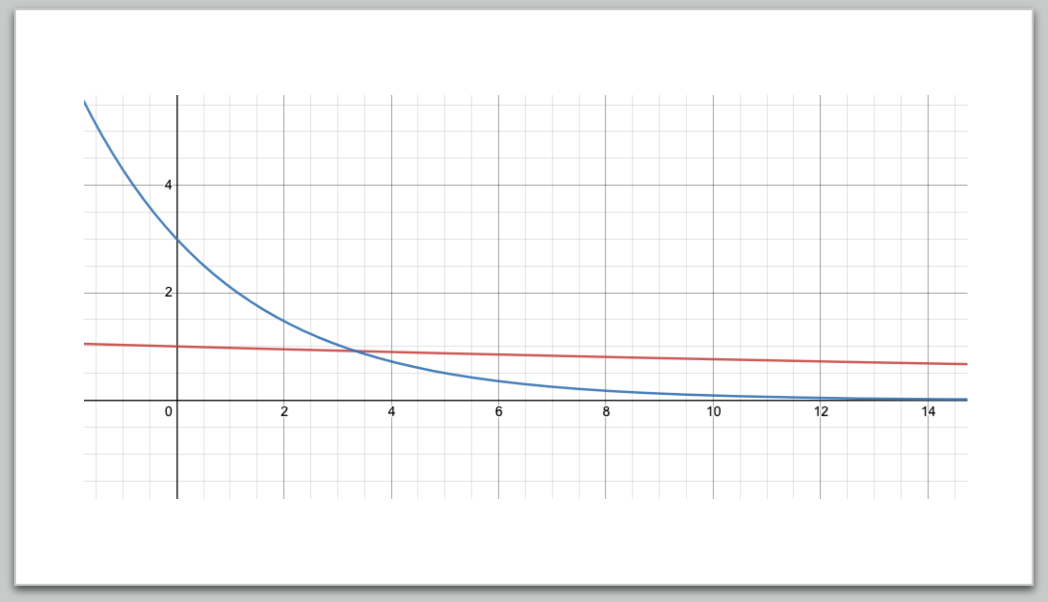
end

- Every honest node stores and relays most recent state values of every other node
- Trimmed-mean is used with the state values of all nodes instead of just neighbors, performed every d iterations

Theoretical Convergence Rate

- Original IABC algorithm
  - Non-zero column in M^{rh}
- Relay-IABC algorithm
  - Non-zero column in M^3
  - d times more iterations per M, but net convergence is faster
- $\left(1-\varepsilon^d\right)^T \gg d(1-\varepsilon)^T$

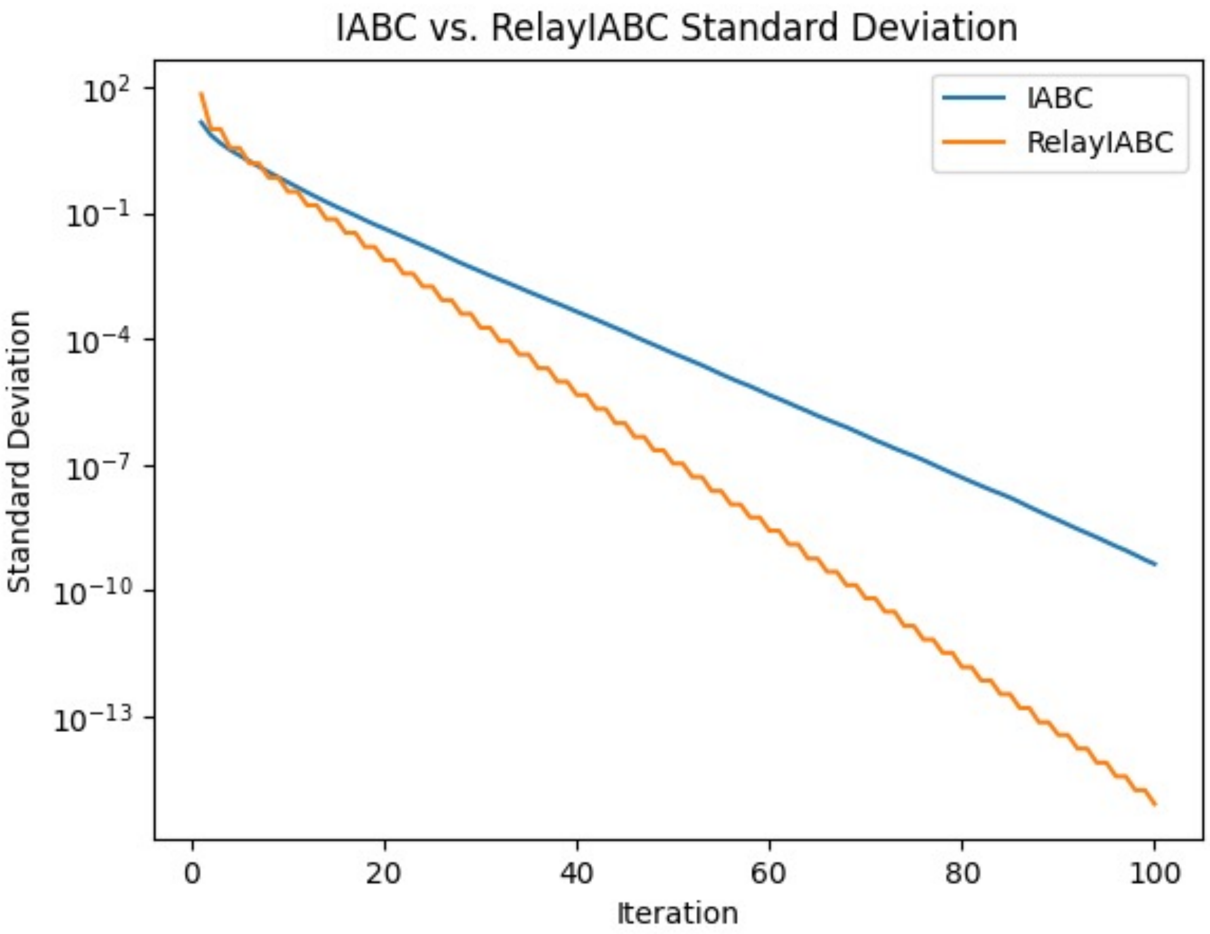




Simulation Results

- Compares IABC and Relay-IABC convergence rates

- Relay-IABC achieves faster convergence



Simulation Graph: Network of 30 honest nodes, 14 byzantine nodes

Blockchain Applications

- Faster Convergence Rate
- Sparse Network Connectivity
- Scalable and Dynamic Protocols

Future Work

- Relationship between update frequency and convergence rate
- Tolerating a higher proportion of Byzantine nodes (signatures)

Acknowledgements

- MIT PRIMES

- Hanshen Xiao

- BRAIN 2021

Thank you!