Relay Protocol For Approximate Byzantine Consensus

Matthew Ding

PRIMES CS/Bio Fall Conference

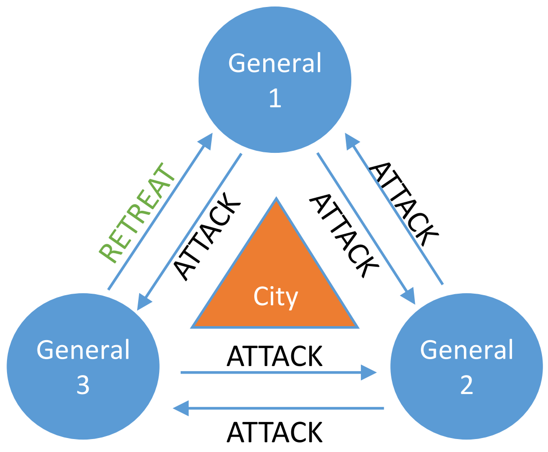
18 October 2020

What is Distributed Computing?

- Information and resources are distributed across a network of different machines
- We want to collectively solve a problem through communication and collaboration

The Byzantine Generals Problem

- Paper published by Lamport, Shostak, and Pease in 1982
- Group of generals camped outside of a city
- The goal is for each general to decide on the same course of action: either "attack" or "retreat"

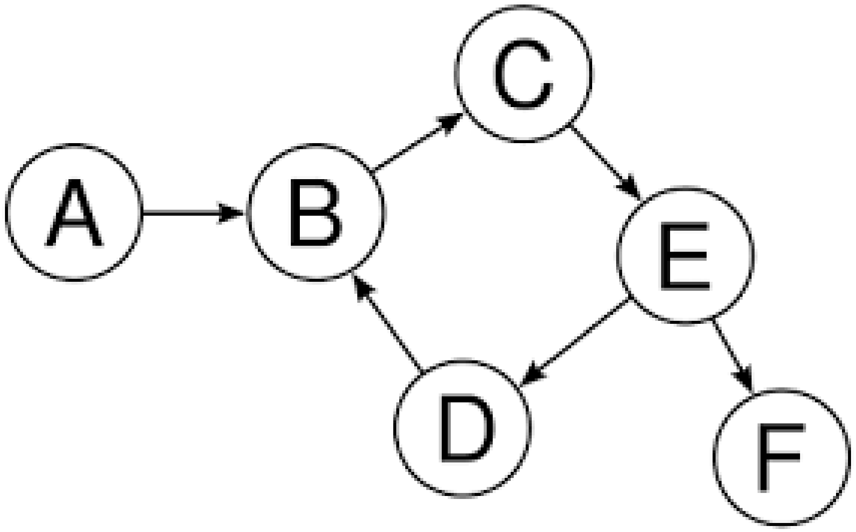


The Byzantine Generals Problem (continued)

- There exist secret “byzantine generals”, who may act arbitrarily and whose goal is to prevent “honest generals” from achieving their goals
- Coined the term “byzantine fault”: a machine that can arbitrarily deviate from an agreed upon protocol in opposition to other users
  - Very strict assumption, but sometimes necessary in real life

A More Mathematical Representation

- Byzantine consensus problems are usually represented as graphs
- In a directed graph:
  - Nodes represent machines (generals)
  - An edge from node i to node j represents a communication link from i to j



Approximate Byzantine Consensus

- Introduced by Dolev et al.
- Each node holds a real number value as their current state
- Nodes achieve approximate consensus on their states with one another rather than exact consensus
- Motivation: Exact consensus is impossible in certain scenarios

What to Solve For

- We may achieve arbitrary exactness by continuing the protocol for an infinite number of rounds

- Aim to satisfy two conditions:
    - Convergence: Every node’s state approaches the same value as the number of iterations approaches infinity
    - Validity: This convergence point is within the range of the initial states

Existing Algorithm

- Developed by Vaidya et al. in 2012
- During each iteration, each node transmits their current state to all neighbors
- Each node performs a trimmed-mean step to determine our new state for the next iteration
- Proven that each node achieves consensus on the same value over time

Trimmed-Mean Step

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

- Vaidya’s algorithm: at least 2f + 1 neighbors, where f is the number of Byzantine nodes
- This is a robust aggregation step for up to f byzantine nodes

Trimmed Mean Step Example

Example of standard mean aggregation step with n=10 and f=3

Honest

0
0
0
1
2
2
3

→ avg(0,0,0,1,2,2,3, 1000,1000,1000) → Output value: 300.8

1000
1000
1000

Byzantine

V.S.

Example trimmed mean step with n=10 and f = 3

Honest

~~0~~
~~0~~
~~0~~
1
2
2
3

→ avg(1,2,2,3) → Output value: 2

~~1000~~
~~1000~~
~~1000~~

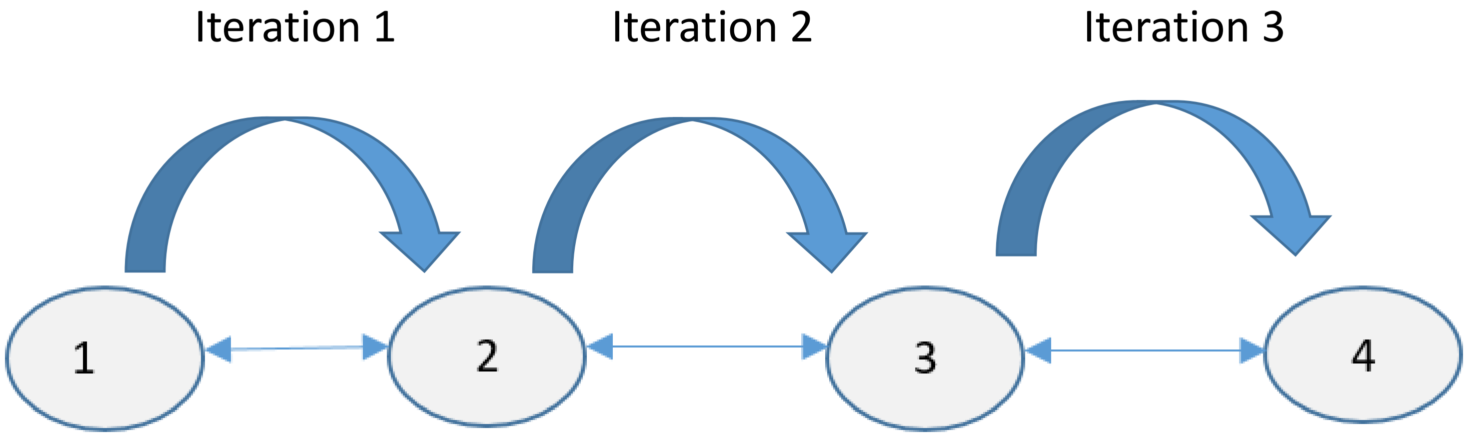
Byzantine

Our Contributions

- Signatures
  - Incredibly important in byzantine consensus, but new to approximate consensus algorithms
  - Reliable proof of who created a message

Our Contributions (Part 2)

- Relays
  - Using signatures, we can now reliably relay messages across a graph
  - Even if a message has been relayed across multiple nodes, we can reliably detect the node of origin



Our Contributions (Part 3)

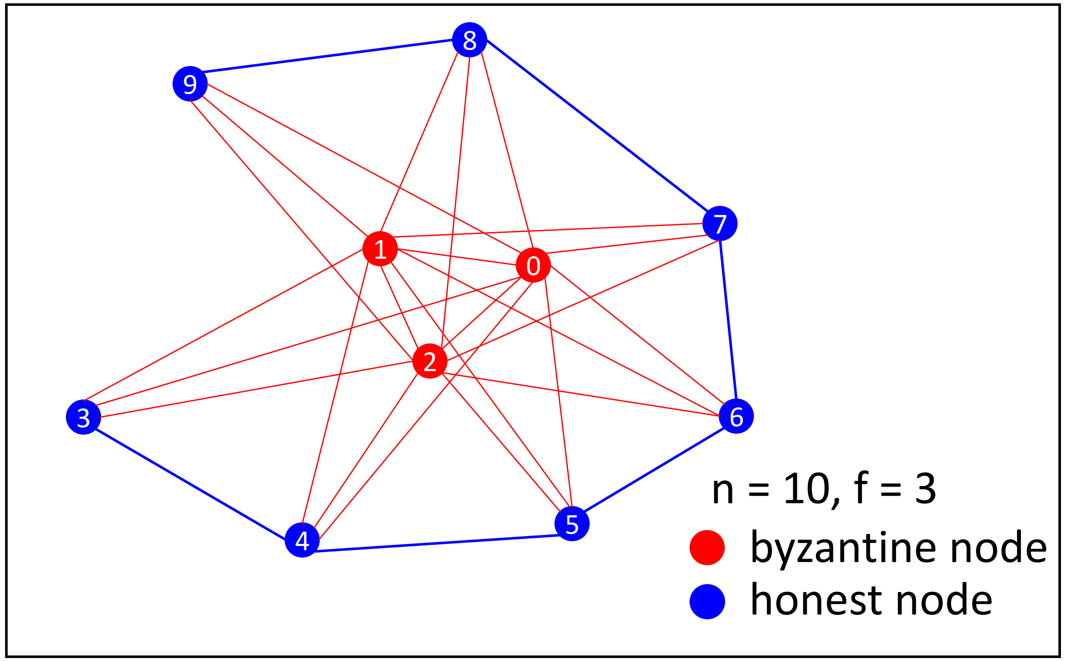
- With relays and signatures, nodes don’t need to be adjacent to communicate with each other
  - All honest nodes in a graph may send and receive messages to every other honest node
- Allows us to assume much less strict network connectivity assumptions
  - Vaidya 2012: Necessary (but insufficient) assumption that each node has 2f+1 neighbors
  - Our protocol: Only assumes bidirectional connectivity

Our Algorithm: Relay-ABC

- Define D to represent the longest distance between any two honest nodes
  - Within D iterations, any message sent from one honest node will have reached every other honest node
- Every node stores most recent state values of every other node
- Every node relays state values of every node to all neighbors
- Each state value in a message is tagged with iteration number and signature
- Trimmed-mean is used with state values of all nodes instead of just neighbors

A Worst-Case Scenario

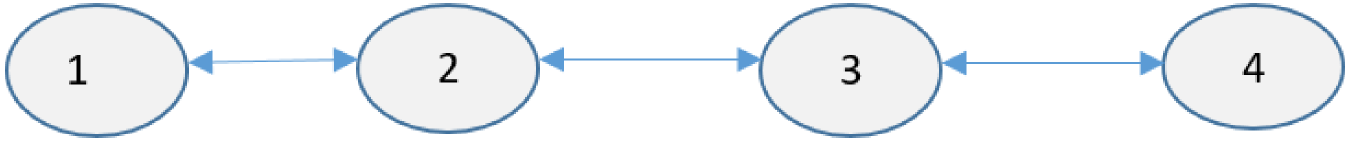
- Honest nodes far outnumber byzantine nodes, but they are not very strongly connected among themselves
  - Every honest node has strictly more byzantine neighbors than honest neighbors
- Only with relays can nodes in this graph achieve consensus



Vaidya's Proof of Convergence

- Vaidya (2012) introduced a proof of convergence using transition matrices
- Given n honest nodes, uses a $n \text{ x } 1$ state vector to represent their states at a given iteration
- Uses $n \text{ x } n$ transition matrices to model the transition between iterations (different every round)

Transition Matrix Example



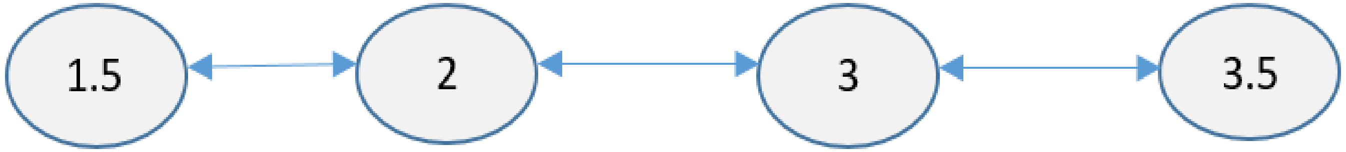
$$\underset{M[0]}{\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}} \underset{v[0]}{\begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}} = \underset{v[1]}{\begin{pmatrix} 1.5 \\ 2 \\ 3 \\ 3.5 \end{pmatrix}}$$

M[0] = Transition Matrix

v[0] = state vector of iteration 0 (initial states)

v[1] = state vector of iteration 1

Transition Matrix Example (Part 2)



M[1] v[1] v[2]

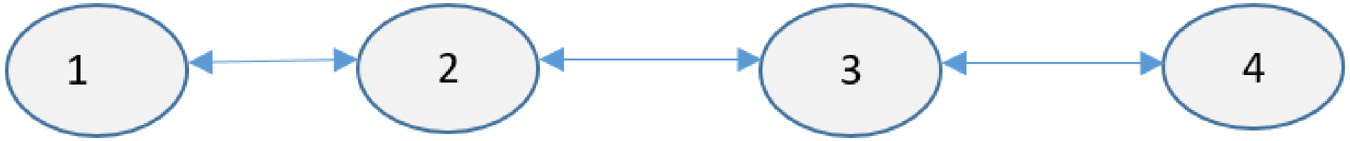
$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 1.5 \\ 2 \\ 3 \\ 3.5 \end{pmatrix} = \begin{pmatrix} 1.75 \\ 2.17 \\ 2.83 \\ 3.25 \end{pmatrix}$$

M[1] = Transition Matrix

v[1] = state vector of iteration 1

v[2] = state vector of iteration 2

Transition Matrix Example (Part 3)



M[0]*M[1]*…*M[7] v[0] v[8]

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}^8 \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} \approx \begin{pmatrix} 2.38845 \\ 2.44897 \\ 2.55102 \\ 2.61154 \end{pmatrix}$$

M[0], M[1]… = Transition Matrices

v[0] = state vector of iteration 0 (initial states)

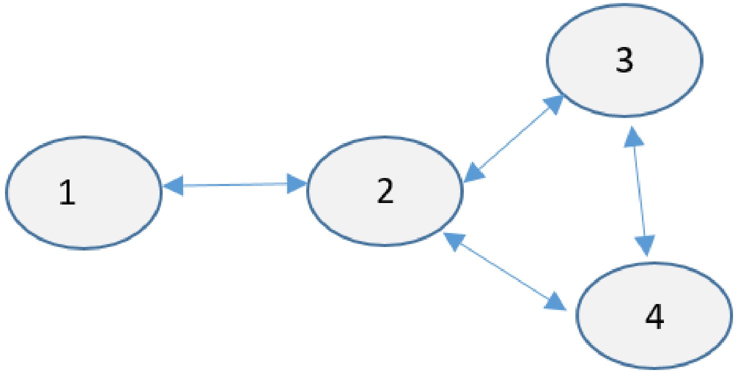
v[8] = state vector of iteration 8

Our Relay Proof of Convergence

- We model our state vector as a nD x 1 matrix, which contains the states of all n honest nodes across D iterations
- The transition matrix is expanded to nD x nD
- We show this expanded version models our algorithm and achieves convergence

Expanded Transition Matrix Example

$$
\begin{array}{ccc}
M[0]*M[1]*M[2] & v[0] & v[3] \\
\begin{pmatrix}
0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
0 & 0 & \frac{1}{16} & \frac{1}{16} & \frac{1}{8} & \frac{1}{8} & \frac{5}{16} & \frac{5}{16} \\
\frac{1}{8} & 0 & \frac{1}{16} & \frac{1}{16} & \frac{1}{8} & \frac{1}{4} & \frac{3}{16} & \frac{3}{16} \\
\frac{1}{8} & 0 & 0 & 0 & \frac{5}{16} & \frac{3}{16} & \frac{3}{16} & \frac{3}{16} \\
\frac{1}{8} & 0 & 0 & 0 & \frac{5}{16} & \frac{3}{16} & \frac{3}{16} & \frac{3}{16}
\end{pmatrix}^{3}
&
\begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}
\approx
&
\begin{pmatrix} 2.34375 \\ 2.44531 \\ 2.51953 \\ 2.51953 \\ 2.45507 \\ 2.45703 \\ 2.45312 \\ 2.45312 \end{pmatrix}
\end{array}
$$



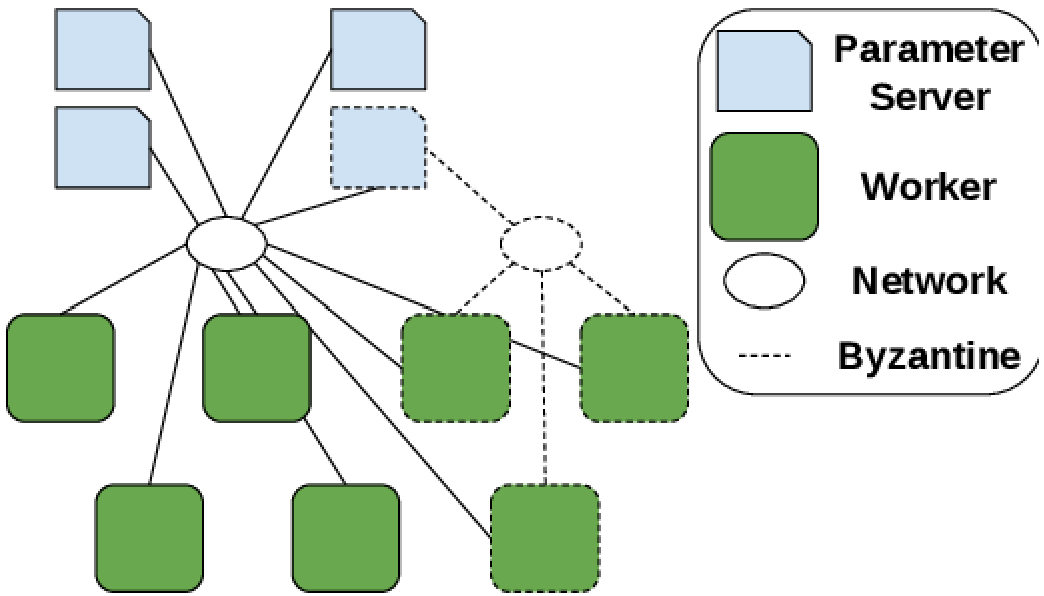
M[0], M[1]… = Transition Matrices

v[0] = state vector of iteration 0 (initial states)

v[3] = state vector of iteration 3

Future Work

- Quantifying convergence rates
- Byzantine machine learning



Acknowledgements

- MIT PRIMES
- My mentor: Hanshen Xiao
- My parents

Thank you!