Proof of *

A Survey of Consensus Protocols, Historical and Today

Monica Quaintance
Head of Public Development

Introduction

This is a survey of base layer consensus mechanisms

Not a discussion of:

- Second layer scaling or liquidity solutions, e.g.:
  - Lightning
  - Cosmos
  - Thunderella
  - Solana
- Directed Acyclic Graphs (DAGs) / Threshold Relay, e.g.:
  - NEM
  - IOTA
  - Hashgraph
  - DFinity

Introduction

About Monica

- Head of Public Development, Kadena
- Co-author, Chainweb protocol paper released at Stanford BPASE 2018
- Former systems engineer, SEC quant, investment banker, and opera singer

Introduction

I. What is Consensus and Why Do We Care

II. Historical Work in Consensus

III. Current Era: Proof of Work Blockchain

IV. New Work in Protocols

What is Consensus

And Why Do We Care

The Problem of Consensus

- Once upon a time, there was one computer
- When there were two computers, we tried to get them to talk to each other
- Once we had multiple computers, we tried to get them to agree with each other regarding shared information about the world

Fault tolerance - how to get a group of computers, or nodes, in a network to agree when there's conflicting information

The Problem of Consensus

- The first consensus problem: aircraft microprocessors
- Outside interference interacts with chip networks
- Need to reconcile conflicting data to provide controlling system with consensus even with false signals

Two critical features of a consensus mechanism:

- **Safety:** Everyone will agree on consistent log of history and there is no direct disagreement
- **Liveness**: Network will eventually come to agreement and system will proceed

Historical Work

Fault Tolerance Problems and Solutions

A Byzantine Problem

- In 1978, Leslie Lamport released his seminal paper on distributed computing
- In 1982, Lamport et al published the Byzantine Generals problem
    - Several generals from Byzantium try to attack Rome
    - Any of the generals can be corrupted
    - Generals can only communicate by messenger
    - BFT (Byzantine Fault Tolerance) can only be achieved by a majority of loyal (non-faulty) generals

The question: how do you have a “leader” if no one can be trusted?

Byzantine Solutions

DLS

- Introduces concept of partial synchrony, some nodes live at the same time
- DLS can allow partial reconciliation for upper bound of liveness
- (Consensus in the presence of Partial Synchrony; Dwork, Lynch, and Stockmeyers)

Paxos

- Lamport created Paxos, solution for crash fault tolerant consensus in async networks in 1990
- Paxos sacrifices liveness for safety -- wait until good behavior reported from nodes
- Used in Unix, Zookeeper
- Paxos is very confusing (Try “Paxos Made Simple”)

Byzantine Solutions

PBFT

- “Practical Byzantine Fault Tolerance” released in 1999 by Miguel Castro and Barbara Liskov
- Higher-performance Byzantine state machine replications
- Processes higher volume of transactions, still has issues with performance at scale

Raft

- In 2013 Diego Ongaro published the Raft paper, a reworking of Paxos
- Equivalent in fault-tolerance and performance to Paxos
- Decomposed into smaller independent sub-problems
- Currently implemented in many languages and powers everything from stock exchanges to “The Cloud”

Current Era

PoW Blockchain: BTC and ETH

Blockchain and Proof of Work

Nakamoto Consensus

- In 2008, the Nakamoto Bitcoin paper introduced a permissionless, economic BFT solution
- System for generating an ordered transaction log that assumes all messages delivered instantly (synchronous)
- Merkle proof passed forward contains record of previous blocks
- The first announced solution is considered valid, and miners compete to generate the next block solution
- Nakamoto consensus allows for leader to be different with every round: metastable leader election function
- As long as 51% of miners are honest, network is stable

Nakamoto Consensus

Forking

- Forks (multiple simultaneous solutions) resolved by using longest-chain rule
- If multiple solutions are offered, network chooses randomly until one is longer
- Probabilistically one chain will pull ahead (in synchronous network)
- Abandoned chain miners get nothing, providing incentive to switch, not compete

Finality

- As block depth increases, likelihood of abandonment decreases quickly / adoption increases
- With reasonable block depth (~6 in BTC), block is probabilistically confirmed

BTC: Stable but slow

Bitcoin's properties make it stable

- Fixed block size
- Fixed block time
- Longest chain fork choice rule + probabilistic confirmation

Features make it a strong and reliable but slow and low-volume for transaction processing

ETH and GHOST

Ethereum introduced two important new concepts to blockchain:

- Blockchain as distributed computation engine (EVM)
- GHOST Protocol Uncle / orphan blocks (not invented but popularized):
  - Rather than using longest-chain fork choice, use heaviest subtree rule
  - Using tree rather than line provides more references between blocks (Merkle tree)
  - More references decreases time to probabilistic finality
  - Allows for faster block confirmation

ETH still suffers from throughput issues (e.g. cryptokitties bottleneck)

New Work

Current and in-progress public protocol developments

New Public Consensus Work

Red Herrings: Non-BFT-economic

- Proof of Reputation (Orbs)
- Ouroboros Praxos (IOHK / Cardano)
- NEM

Deterministic

- Proof of Stake
- Casper TFG (Friendly GHOST)
- Tendermint
- DPOS & Federated POS

Probabilistic

- Proof of Space and Time (Chia)
- Chainweb (Parallel Nakamoto)

Red Herrings: Non-economic-BFT, etc

Methods of fork choice / consensus that are not BFT

- Ouroboros Praxos (IOHK / Cardano)
- Proof of Reputation systems (e.g. Orbs)
- NEM

Validation projects that are not consensus mechanisms:

- Proof of location (foam.space)
- Proof of age / citizenship, supply chain, etc

Proof of Stake

- Proof of Stake first introduced in 2012 by Sunny King and Scott Nadal
- Intended to reduce mining energy consumption in PoW by moving back towards RAFT-like protocol
- Participants stake currency in the network in order to vote for blockchain history
- Current issues with PoS:
    - **Nothing at stake:** block generators have nothing to lose by voting for multiple histories
    - **Long-range attack:** after attacker withdraws stake can double spend
    - **Tragedy of the commons:** cheaper bribery to induce voters to become bad actors
- Different PoS projects attempt to solve issues with delegates, federating, slashing, and other game theory

Proof of Stake Variants

- Casper TFG (Friendly GHOST)
  - Full implementation of GHOST
  - Validators assemble blocks, only staked validators can participate
  - Uses currently active security deposits to solve “Long Range”
  - Bad validation forfeits stake (slashing)
  - Rotating set of validator nodes based on who’s currently active
- Tendermint
  - Hybrid PBFT + Proof of Stake
  - Like BFT: rotating leader election
  - Like blockchain: Merkel proofs hash link blocks
  - Slashing to solve “Nothing at Stake”
  - Favors availability over consistency (eventually consistent)

Proof of Stake Variants

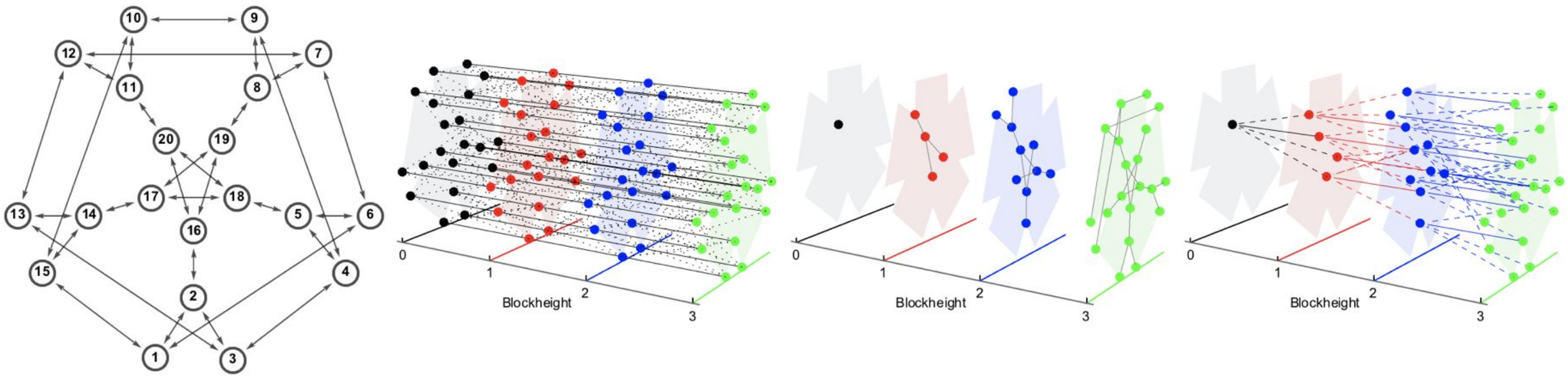
- Delegated / Federated Systems
    - **Neo** - Delegated BFT (DBFT); elect N bookkeepers, 2/3 of all bookkeepers agree on what goes in every block
        - Node holders are also held liable legally, sacrifices Liveness for Safety
    - **BitShares, EOS** - Delegated PoS (DPoS); elect N witnesses, each witness produced 1 block on their own
        - Issues with low voter participation, tragedy-of-the-commons lack of incentive for correctness, users and voters not aligned on interests (e.g. users vs. speculators)
    - **Ripple, Stellar** - Federated Byzantine Agreement (FBA); series of overlapping independent BFTs
        - Not actually BFT, sacrifices Liveness for Safety and in event of failure does not progress

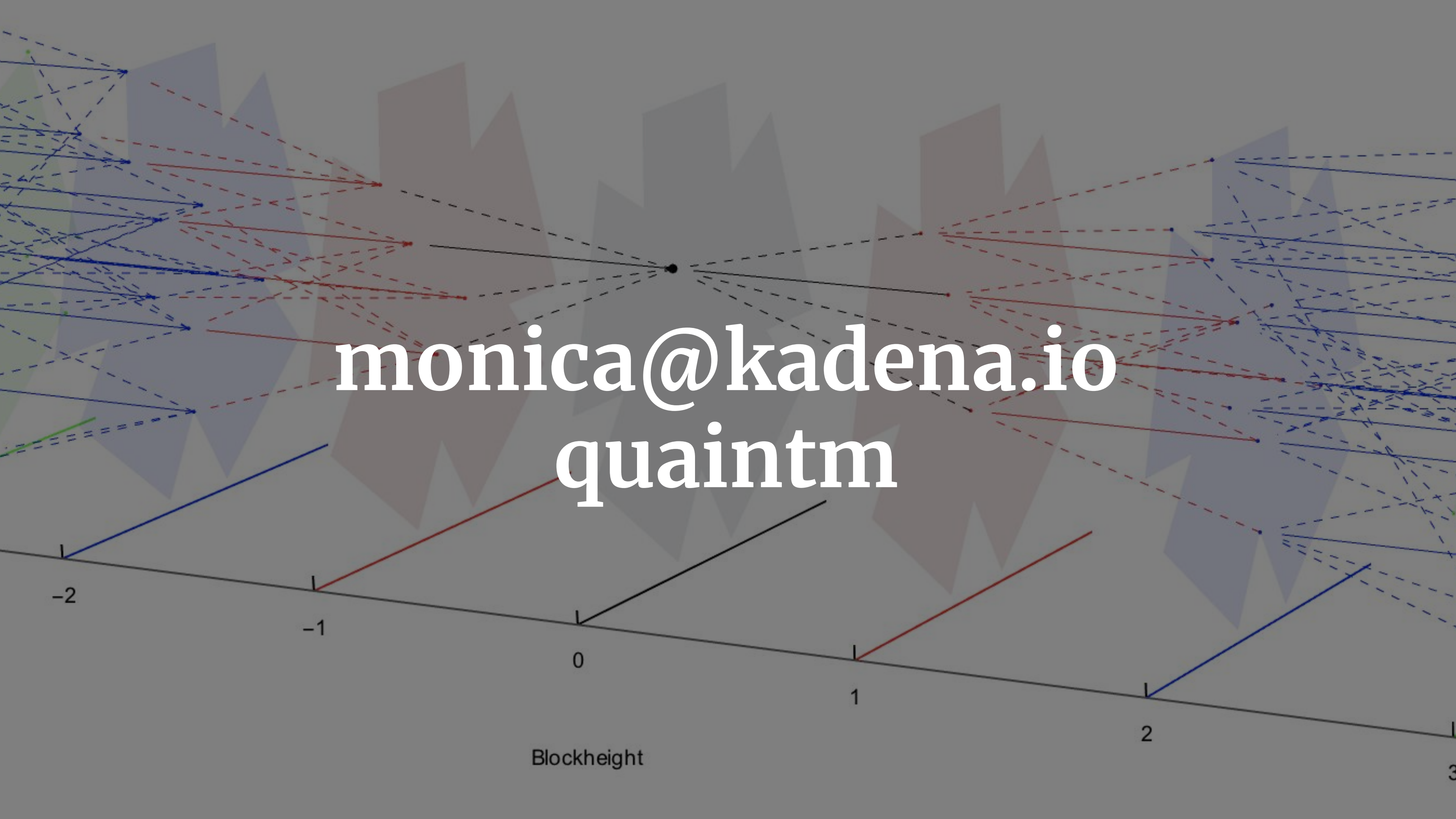
Non-deterministic: Proof of Space and Time

- Pre-computed values take up some agreed-upon disc space
- Computing those values takes some arbitrarily large amount of time
- Network participants pre-compute and store values on dedicated volumes
- Upon request, participants must return a random value from the store in a window of time smaller than the compute time
- Under development by Chia (Bram Cohen + Ryan Singer)

Chainweb

- Under development by Kadena
- Braided Nakamoto consensus that hash links many PoW chains together
- Braiding creates Scalability, Security, Speed
- Platform supports application layer with simple development language





monica@kadena.io
quaintm