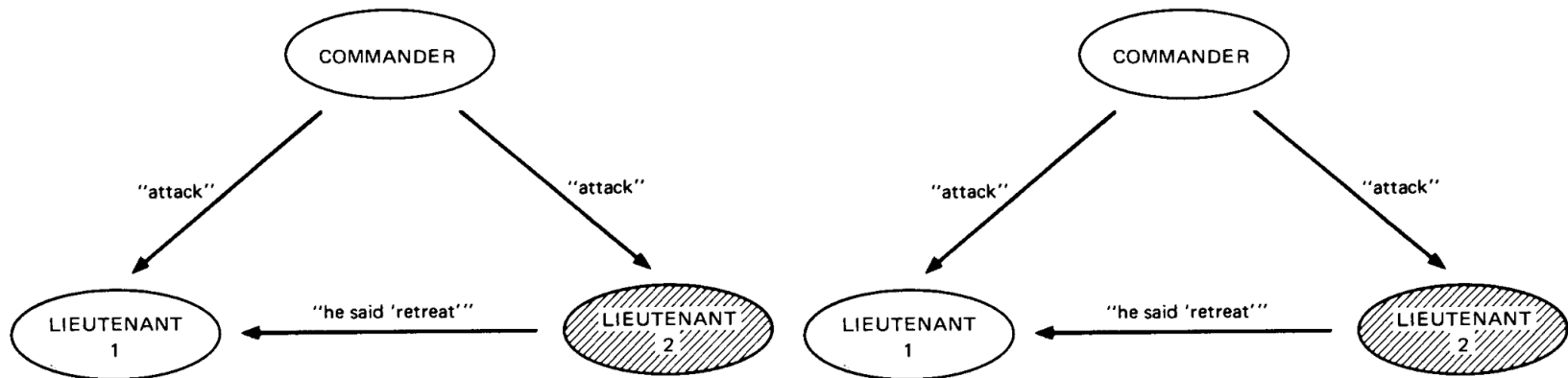


Hashgraph

The Future of Decentralized Technology

Byzantine Generals Problem

- A commanding general must send an order to his $n - 1$ lieutenant generals such that
 - All loyal lieutenants obey the same order
 - If the commanding general is loyal, then every loyal lieutenant obeys the order he sends



Byzantine Generals Problem

- distributed systems need to deal with failure or conflicting behaviours of its components
- fun fact: *Lamport: "I have long felt that, because it was posed as a cute problem about philosophers seated around a table, Dijkstra's dining philosopher's problem received much more attention than it deserves."*

Distributed Consensus Algorithms

- Proof-of-work (Bitcoin, Ethereum)
 - slow (10 transactions per second)
 - waste of energy (\$1M/day ~ Mauritius energy consumption)
- Proof-of-stake (Ethereum Casper)
 - much lower energy consumption
- Proof-of-burn (Slimcoin)
 - mining can be done on low power machine (just a few SHA256 hashes)
- Leader-based (Hyperledger Fabric)
 - Paxos, Raft, PBFT
- Voting-based (no implementation)
 - excellent theoretical properties
 - high bandwidth requirements $O(n^2)$



blockchain



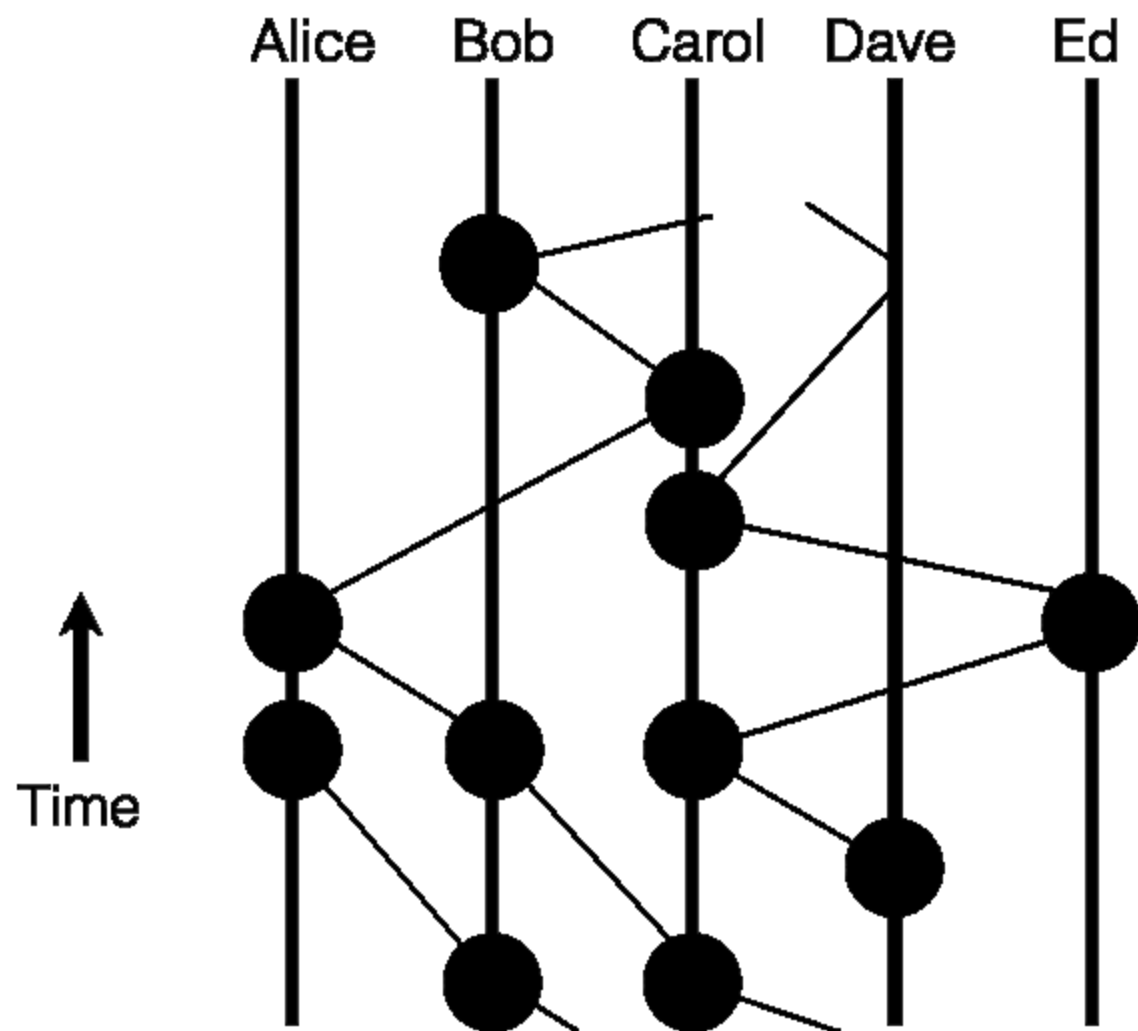
hashgraph

What is hashgraph?

Hashgraph is data structure and consensus algorithm that is:

- Fast: With a very high throughput and low consensus latency
- Secure: Asynchronous Byzantine fault tolerant
- Fair: Fairness of access, ordering, and timestamps

These properties enable new decentralized applications such as a stock market, improved collaborative applications, games, and auctions.




Assumptions to achieve Byzantine Fault Tolerance (BFT)

- more than $2/3$ of participants are honest (theoretical limit for BFT solutions)
 - so how does Bitcoin tolerate up to $< 50\%$ malicious nodes?
- message between honest members eventually get through
 - other than that, attackers are allowed to completely control the network, deleting and delaying messages arbitrarily
 - if messages *never* get through, any distributed system will fail. So it is a failure of the network, not of the consensus algorithm

Interesting characteristics

- absolute confirmation of transactions (unlike proof of work)
- order of transactions preserved (compare with proof of work where transaction order is determined by miners)
- no wasted computation (compare with blockchain forking)
- just gossip and everything will work
- really fast (no additional comms for consensus)

The consensus algorithm

Refer to  Hashgraph detailed example

Intuition of how hashgraph works

- order transactions by the time a majority of the nodes learns about it (thru the gossips)
 - must be $\frac{2}{3}$ or more, 50% doesn't work against $\frac{1}{3}$ attackers
- what are witnesses for? some kind of optimization to save computations
- what is strongly seeing for? to protect against forking

Real world applications

- time-sensitive
 - stock exchange
- high throughput

The Byzantine Generals Problem

<http://research.microsoft.com/users/lamport/pubs/pubs.html#byz>

Deconfusing Decentralization

<https://youtu.be/7S1IqaSLrq8>

Hashgraph introduction at TechCrunch Disrupt

<https://youtu.be/ZrFrXFdRW4k>

Leemon Baird x Harvard Talk

<https://youtu.be/ljQkag6VOo0>

Beginner's Guide to Ethereum Casper Hardfork

<https://blockonomi.com/ethereum-casper/>

Ethereum Proof of Stake FAQ

<https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ>

A (Short) Guide to Blockchain Consensus Protocols

<https://www.coindesk.com/short-guide-blockchain-consensus-protocols/>