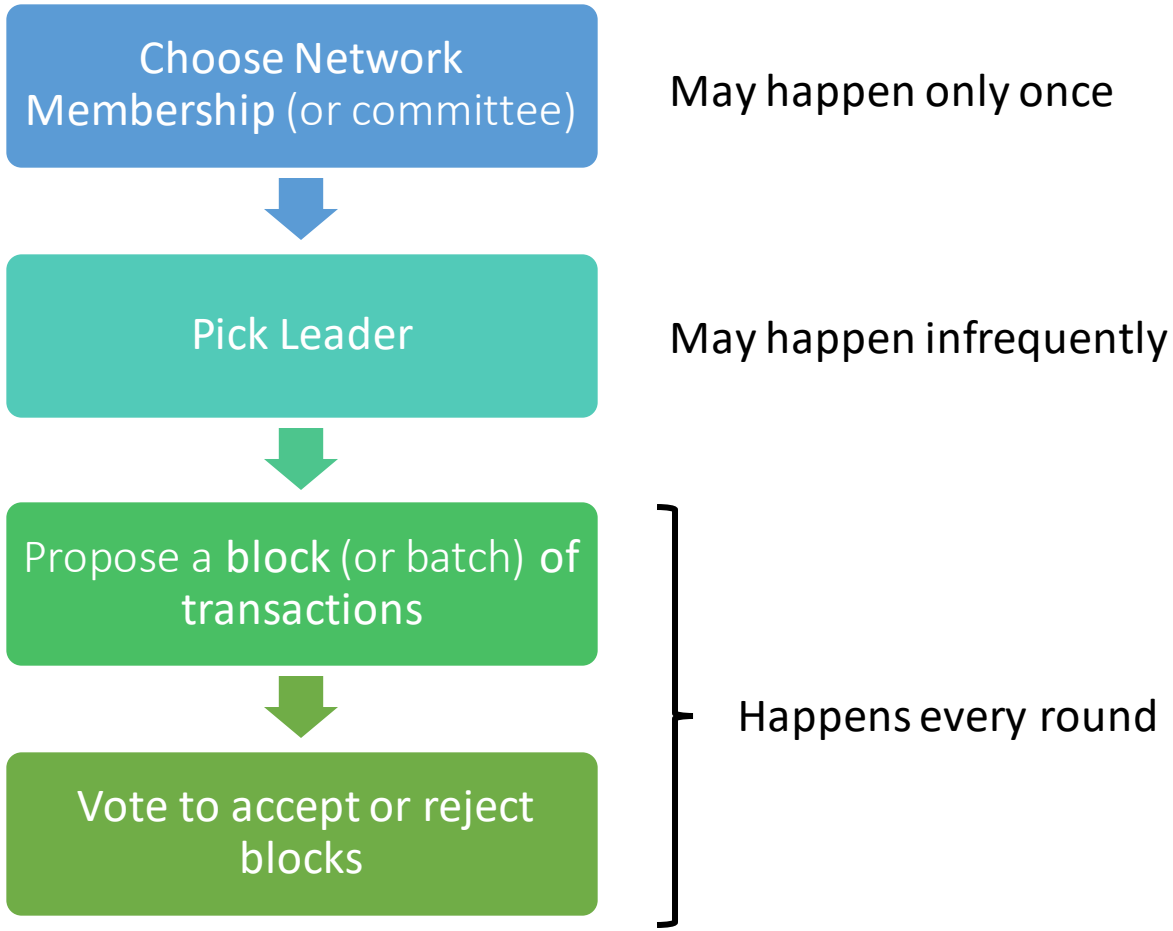
Ethereum 2.0

CS839 -- Kai Mast

Breaking Down Consensus Protocols



1. Picking Membership

Members: Nodes with voting power

	Static (Never changes or changes very infrequently)	**Dynamic (Can change at any point in time)**
Explicit (Everyone knows* all other members)	Pre-Defined Committees (e.g., PBFT and derivatives)	Proof-of-Stake, Delegation Mechanisms
Implicit (Set of members may never be fully known)	N/A	Proof-of-Work, Proof-of-Space, etc.

* Members might still be hidden behind pseudonyms

2. Picking Leaders/Proposers

- Voting: e.g., PBFT, HotStuff
- No Leader: e.g., Avalanche, HoneyBadgerBFT
- Random Function: e.g., Ouroboros, OmniLedger
- Proof-of-Works and similar mechanisms: e.g., Bitcoin, Ethereum 1.0

Why Leaders?

- Allows pre-ordering/batching transactions
- Can coordinate voting process
- Easier to maintain liveness
  - Prevents conflicting proposals
  - Leaders can be replaced relatively easy

Why No Leaders?

- Prevents denial of service attack
- Removes potential bottleneck
- Leader election might be hard under certain network conditions

3. Proposing (Batches/Blocks of) Transactions

- In Leaderless protocols, each client might propose their own set of transactions
- In Leader-based protocols batching can be used for higher throughput

Ordering the Blockchain:

- Blocks might be totally ordered, e.g., Bitcoin, Ethereum
- Transactions/Blocks might form an Acyclic Graph, e.g., Avalanche

Ordering within Blocks:

- Most systems require blocks to be totally ordered
- Content of blocks can also be partially ordered
  - Requires transactions to be independent of each other
  - Concurrency within blocks can cause issues with regards to deterministic execution

4. Voting

Once proposed, the network must accept or reject a particular (block of) transaction(s)

- Makes sure that blocks have been seen by most of the network
  - Ensures availability and partition tolerance
- Prevents a faulty or malicious leader from proposing invalid transactions

Voting Mechanisms

Explicit Voting

- Requires at least two rounds
- Can either be all-to-all communication (PBFT) or star communication (HotStuff)
  - Star communication can be pipelined

Picking the longest chain

- Probabilistic: We never known for sure that this is the longest chain
- Voting happens as part of proposing the next block and by network participants forwarding or rejecting new blocks

Audit Mechanisms

- "Voting" by absence of objections

Endorsement/Attestation (Ethereum 2.0)

- Nodes attach their stake to a proposed block
- Can be viewed as probabilistic voting

Ethereum 2.0

Goals

- Move away from Proof-of-Work
  - Proof-of-Stake is more energy efficient and potentially more decentralized
- Allow for higher throughput
  - Ethereum 2.0 will be sharded
  - The EVM will be replaced by a new virtual machine based on WebAssembly

Proposed in 2014, but still in development

- Development/Research is backed by the Ethereum Foundation
- There is no formal specification for the entire protocol yet
- Upgrade will happen in multiples stages

Ethereum 2.0 Roadmap

Phase 0: The Beacon chain

- Initial test for the sharding protocol
- Exists alongside the Ethereum 1.0 chain
- Nodes become stakers by storing money in a contract on the 1.0 chain
- Beacon chain only keeps track of meta-data, does not hold actual accounts or contracts

Phase 1: The Merge

- The entire Ethereum network will move over to Proof-of-Stake
- There will be one beacon chain and one main chain

Phase 2: Sharding

- The Ethereum chain will be split into multiple shards
- There will be one beacon chain and many shard chains

Gasper: The Eth2.0 Consensus Protocol

Gasper = Casper + GHOST

Casper: The "friendly finality gadget" (Casper FFG)

- Provides 100% certainty that a block is accepted by a probabilistic networks
- Allows for faster confirmation times
- Finality allows pruning state (we know state at some point immutable, so competing forks and transaction history is not needed anymore)

GHOST: Greediest Heaviest Observed SubTree

- Blocks reference "uncles" as well as their parents
- Reduces the number of forks when using high block intervals
- GHOST is used to pick the longest chain in Gasper

Recap: Stake and Slashing

- Nodes lock up stake
  - Determines their voting power
  - Serves as a deposit

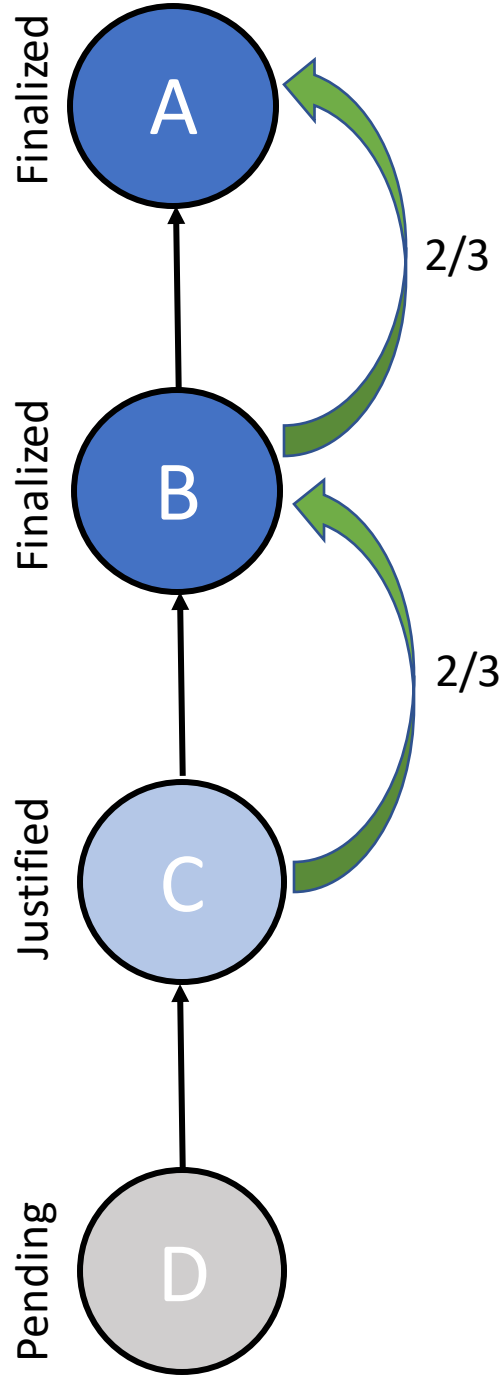
- Nodes that do not follow the protocol get punished by their stake being taken ("slashed")
  - Only works in protocols where the stake is locked up, so not in Ouroboros or Algorand

- For correct Gasper/Ethereum2.0 chain, less than 1/3 of the stake is "slashable"
  - At most 1/3 of the stake is controlled by an attacker
  - Somewhat stronger guarantee than just saying f<=1/3 because even rational nodes have an incentive to always behave correctly because of slashing

Casper FFG

- Block in Casper are proposed similar to Ouroboros
  - Each slot has a pre-determined leader which may propose a block
  - Slots are grouped into epoch
  - But no assumption of a synchronous network. Blocks may arrive late.

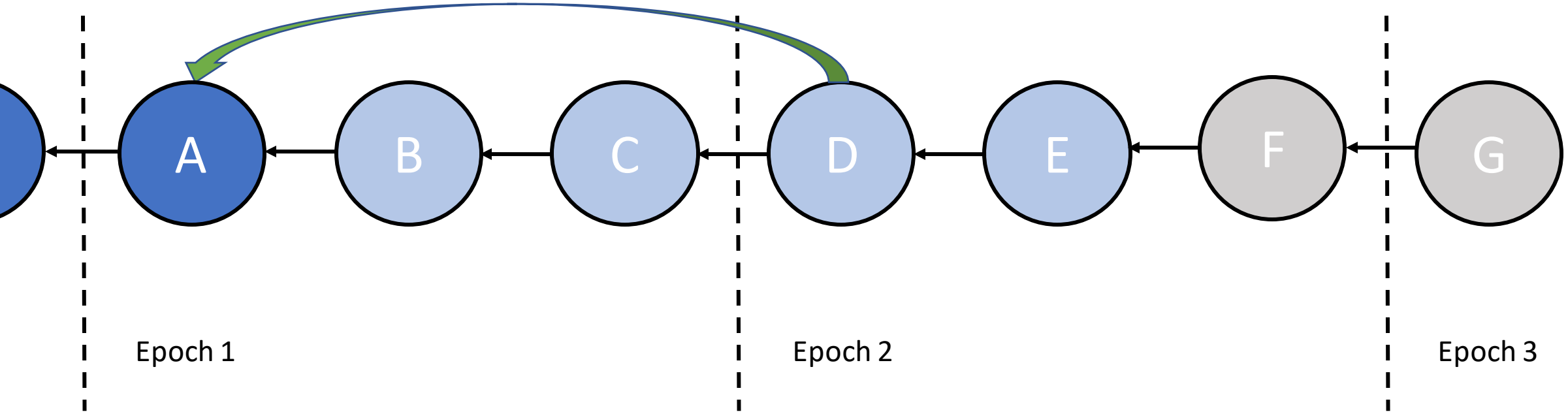
- Blocks are not finalized until approved by the majority of the validators (or stake)
  - Validators voted on edges between epochs to approve or reject blocks
  - A block is *justified* once the edge between it and its predecessor has been approved
  - A block is *finalized* once the edge between it and its successor has been approved



Epochs in Casper

Idea: Do not vote on every block, but only on the first block in an epoch

- If a block is finalized / justified, all their ancestors are also finalized / justified
- A block that is finalized is always justified
- We call the first block in an epoch an *"epoch boundary block" (EBB)*

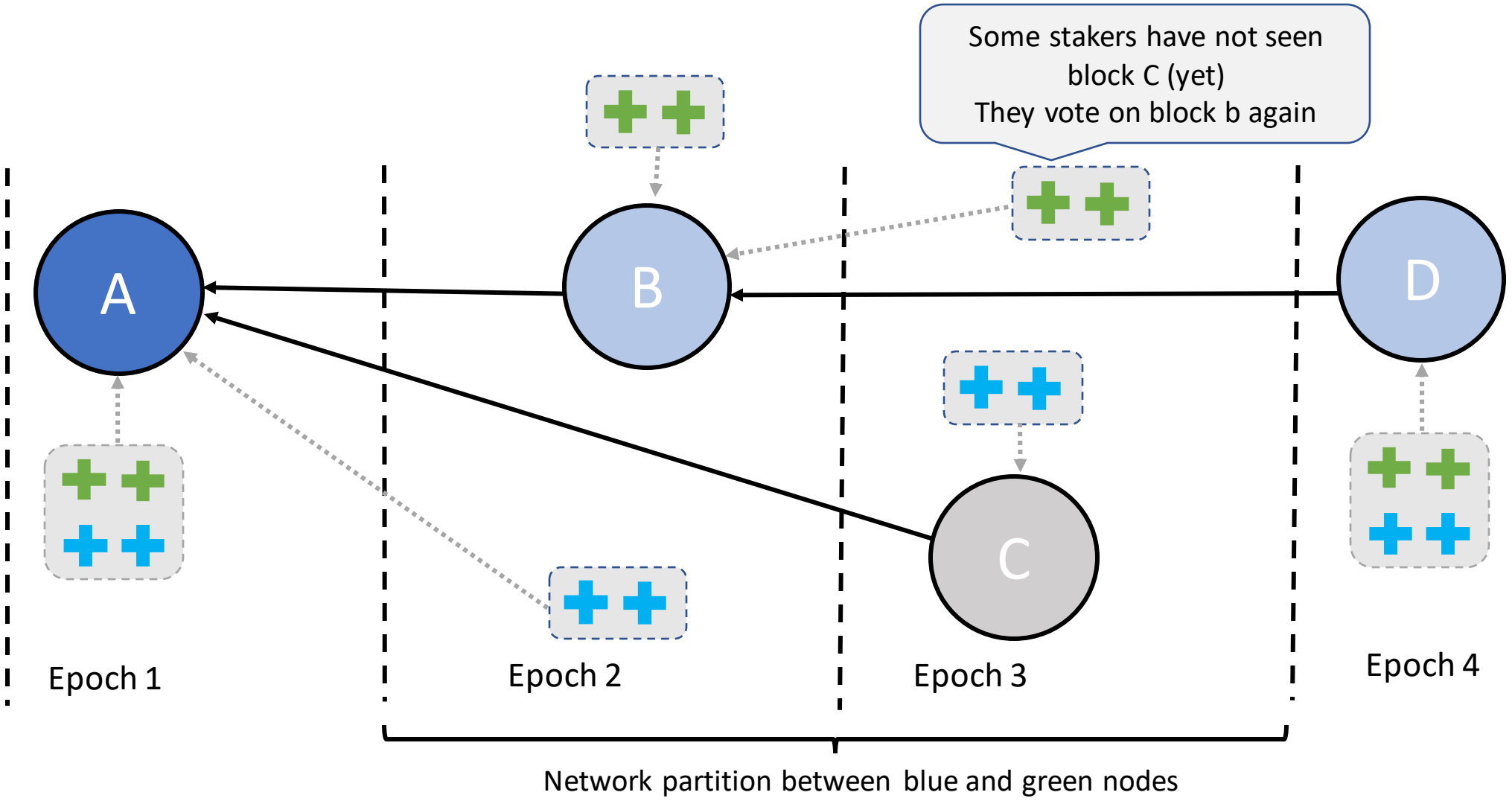


Attestations in Casper

- Staker's vote on a block by issuing an attestation
  - w.l.o.g., we assume each staker has the same fraction of the stake
  - In the real world each staker might have different stakes and a different amount of voting power per epoch
- Restrictions
  - Stakers can only issue one attestation per epoch
  - Stakers can only attest to descendants of blocks that they voted to finalize*
- Attestations are broadcast and eventually included In a block
- Incentives:
  - Stakers issuing invalid attestations are subject to being slashed
  - Stakers issuing correct attestations receive a reward

* not explicitly stated in the writeups but the protocol would not be sound otherwise

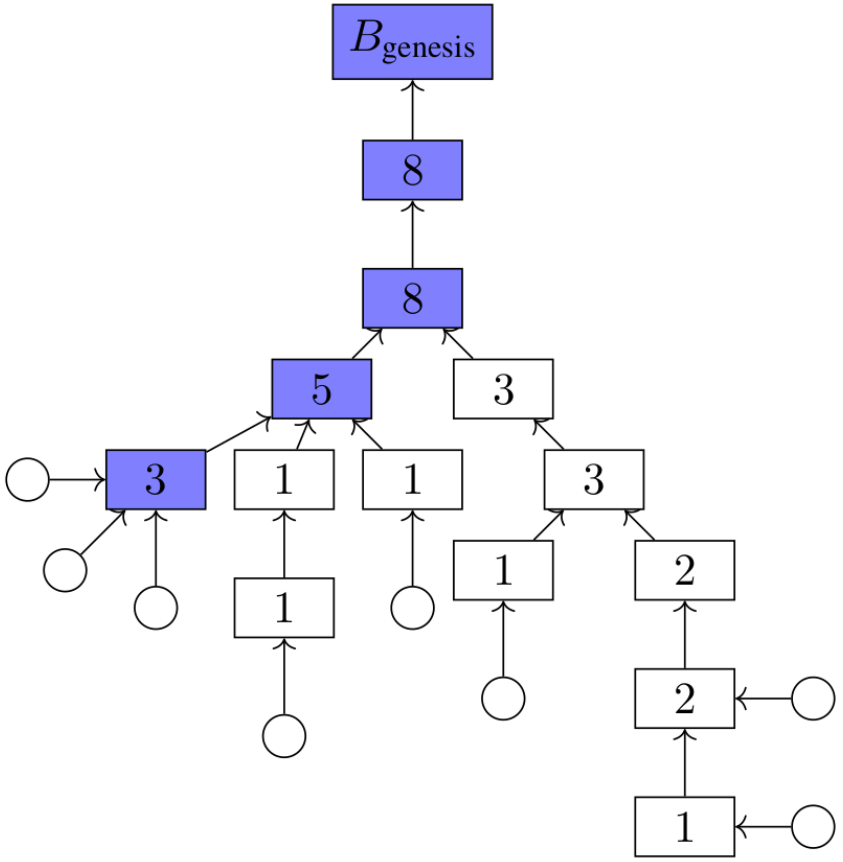
Attestations in Casper: Example



Resolving Forks in Gasper

Gasper relies on Last Message Driven GHOST (LMD GHOST) to choose between forks

- GHOST: pick the heaviest branch of tree
- Last Message Driven
  - A blocks weight is the sum of stake of all its attestations and attestations to its descendants
  - Only the most recently seen attestations count
- If there is a tie, all nodes deterministically break th tie based on the block hashes



- Numbers are weights
- Circles are last-seen attestations

Sharding in Gasper/Ethereum 2.0

- No concrete specification exist yet, but intended to be similar to OmniLedger
- Every epoch (or k epochs) randomly assign stakers to shards
- Stakers store their attestations for shard blocks on the beacon chain
- Attestations serve, both, as
  - Availability proof for the shard chain
  - Attestation to a beacon block (each shard block is anchored to a beacon block)

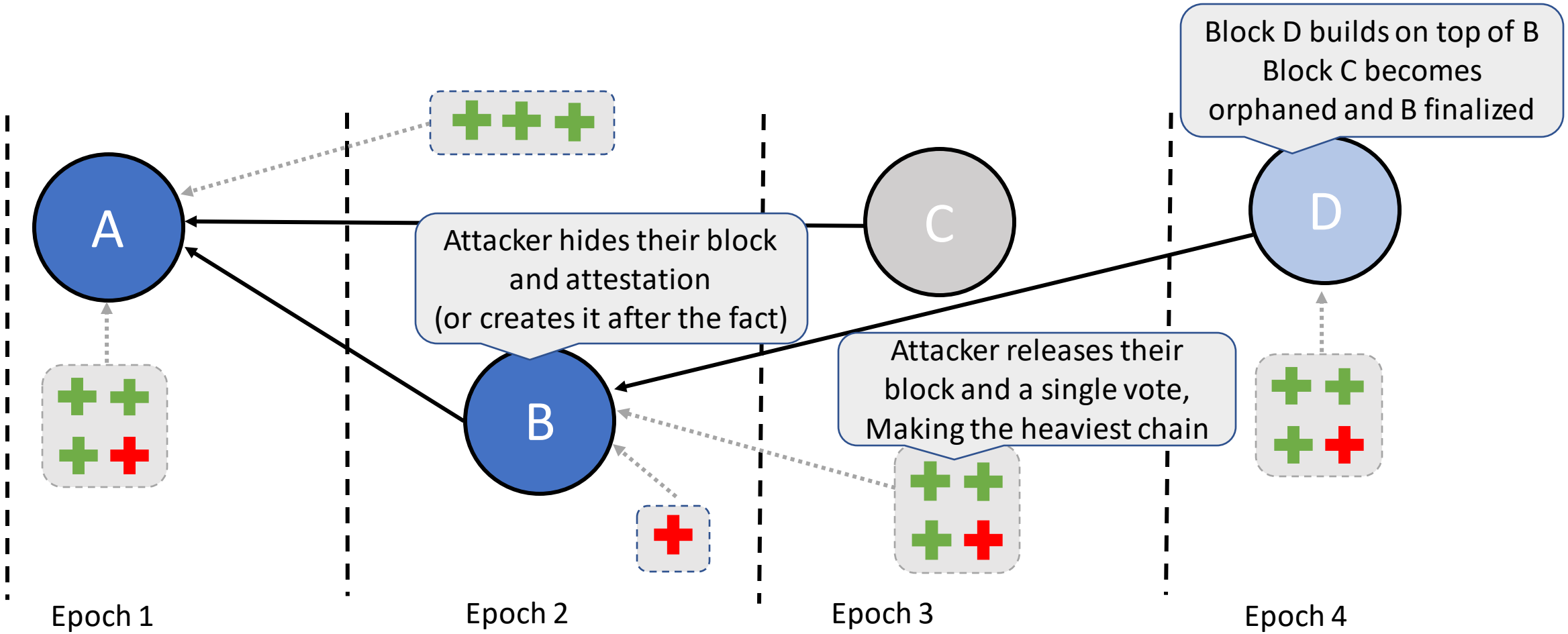
(Known) Attacks on Gasper

Two known types of attacks (can be combined as well)

- Withholding blocks:
  - A malicious leader/block proposer may hide their block to trick the rest of the network
  - Similar to selfish mining in Nakamoto Consensus / Ouroboros
  - Remember, in proof-of-stake blocks can be created after the fact
- Network delays:
  - Blocks and attestations could be delayed causing forks
  - An attacker might control (parts of) a network to do this intentionally
  - There can also be temporary network partitions

Basic Reorg Attack

Reorg = Reorganizing the chain to undo/reorder transactions



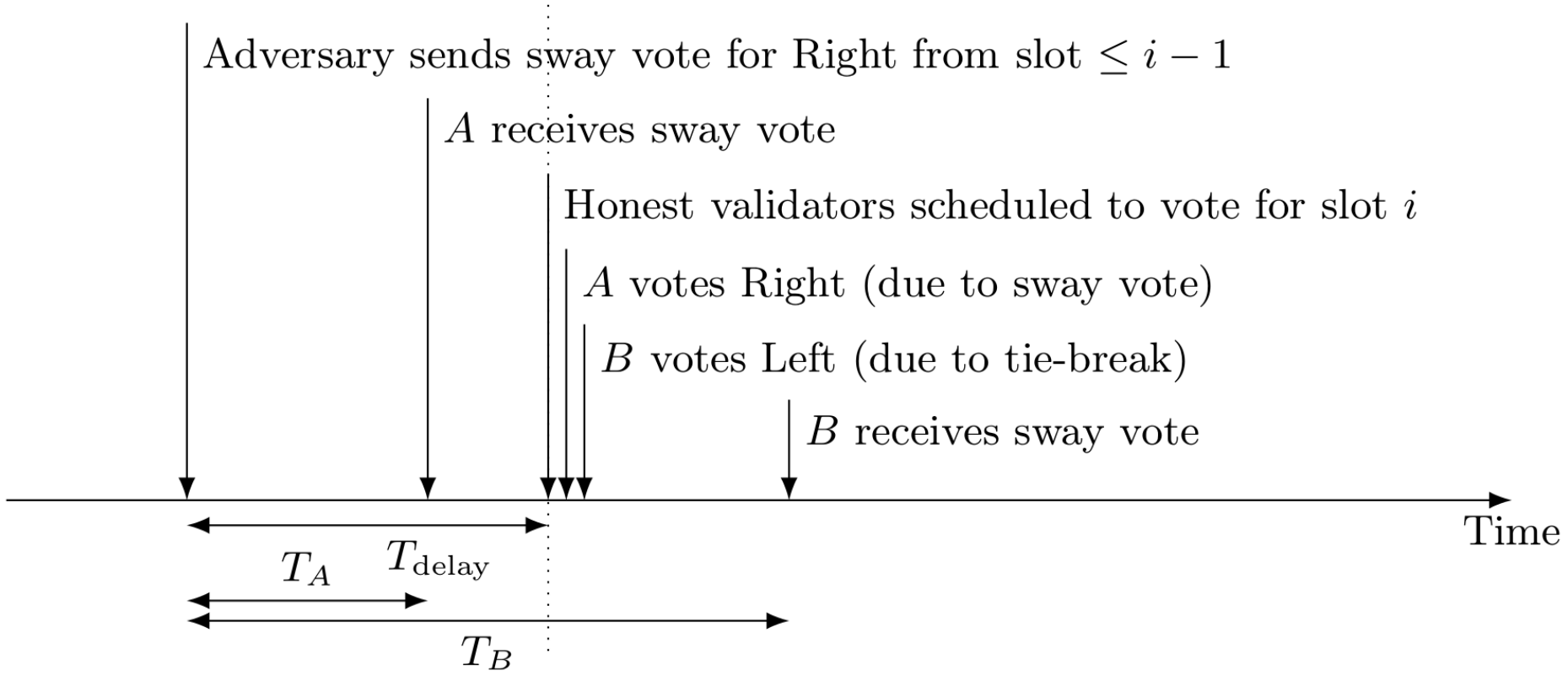
Delaying Consensus Attack

Goal: Prevent the protocol from finalizing blocks, or only allow it to finalize blocks slowly

Intuition:

- Attackers are globally distributed and propose competing blocks/chains
- Honest validators might attest to either chain/block
  - Split vote results in neither fork getting a 2/3 vote
- Attackers can amplify this stalemate by voting for one of the forks and "swaying" the honest nodes to one side or the other

Delaying Consensus Attack (cont.)



- Roughly, half of the nodes see the sway vote for the right branch
- The other nodes pick the left branch due to deterministic tie breaking

Discussion

- Difference between HotStuff, Ouroboros, and Gasper?
  - Ouroboros (the original paper) operates in a synchronous environment
  - HotStuff dynamically adapts to changes in latency, while Gasper does not guarantee liveness during high-latency periods

- Is Gasper/Ethereum2.0 a synchronous protocol?
  - It is partially synchronous protocol, but note the restrictions on liveness above
  - If global latency changes permanently, the protocol needs to be reconfigured. There's no built-in mechanism to do that.

That's all for today

- Next time: Stellar and Ripple
- Reminder, to send me updates about your project

Sources:

- Combining GHOST and Casper, Buterin et al., 2020
- Three Attacks on Proof-of-Stake Ethereum, Schwarz-Schilling et al., 2021