Lecture 15: Blockchains with Finality

https://web3.princeton.edu/principles-of-blockchains/

Professor Pramod Viswanath
Princeton University

This lecture:
Finality of confirmation
Byzantine Fault Tolerant Consensus

Course Outline

- Module 1
  - Bitcoin (7 lectures)
- Module 2
  - Scaling Bitcoin (7 lectures)
- Module 3
  - Beyond Bitcoin (6 lectures) starting today

Pros and Cons of the Longest Chain Protocol

- Liveness
  - Even a single honest miner with a small hash power can extend the longest chain
- Safety
  - Guaranteed when hash power of honest nodes is more than 50%
  - But with 2 caveats
    - Probabilistic guarantee
    - Network must be synchronous

Byzantine Fault Tolerant (BFT) Protocols

- Deterministic safety even under asynchronous network

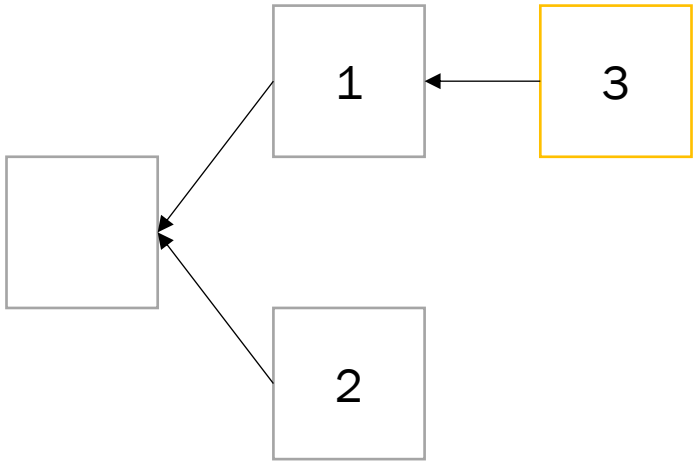
- Two closely related protocols:
  - Streamlet
  - HotStuff

BFT Protocol Setting

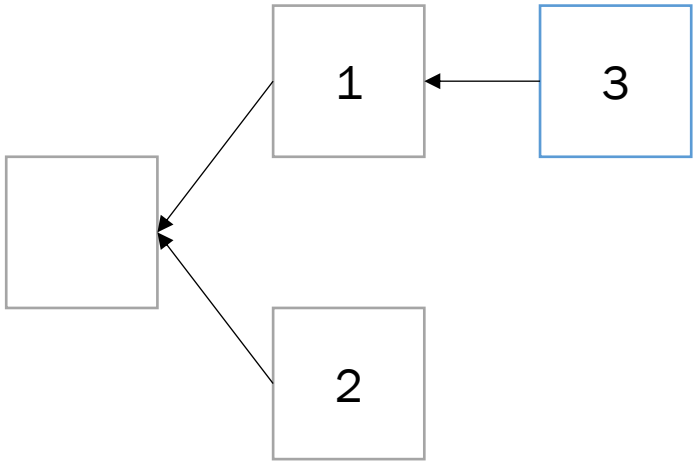
- Number of Participants is fixed
- Identities known (signatures) to all nodes
- In other words, “permissioned”

BFT Steps (Round-by-Round)

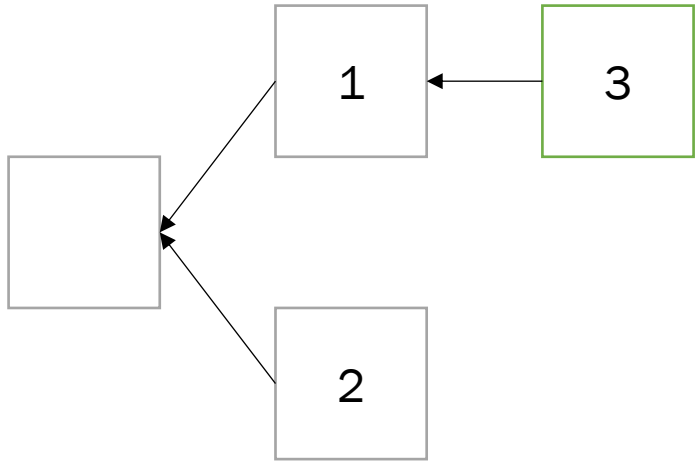
- In each round:



1. Propose a new block

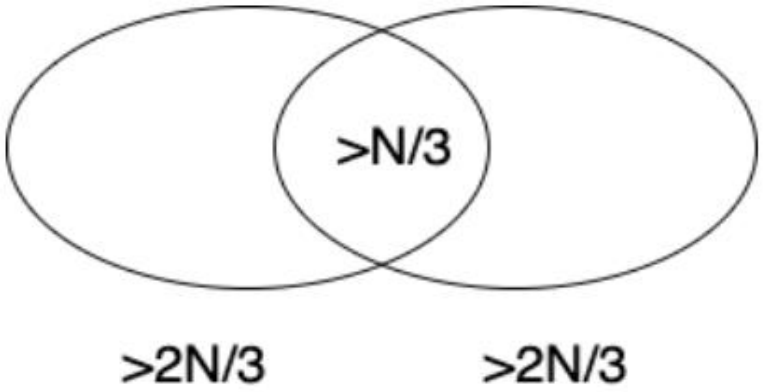


2. Vote



3. Notarized
= enough votes

BFT Confirmation Rule

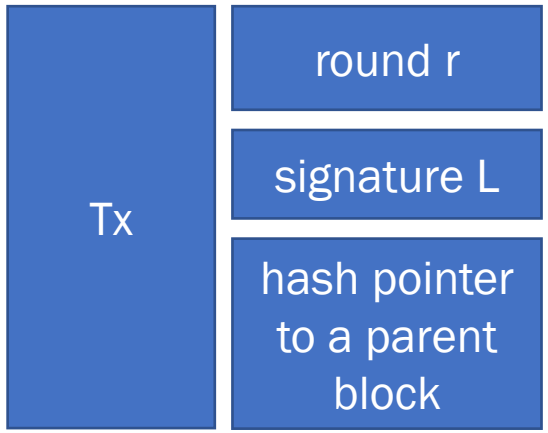


Streamlet

- Proceeds in lock-step rounds, each of which takes twice the communication delay
  - One leader elected every round
  - Leader will collect pending Tx and put a block together and propose
  - Whenever a block is proposed
    - Checks if signed by leader with rights to propose in round r
    - Vote on a proposed block if the block extends the longest notarized chain (a block is notarized if it receives at least 2N/3 votes)
- All nodes re-broadcast all messages they hear of
- A node does not vote for conflicting blocks (blocks at same height)

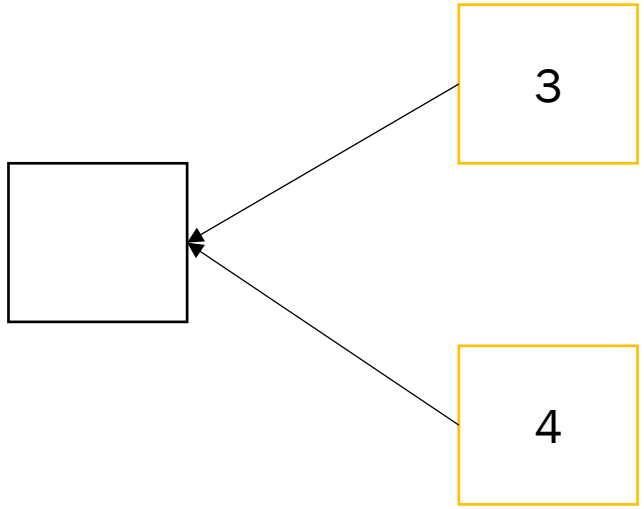
Streamlet

Block Structure



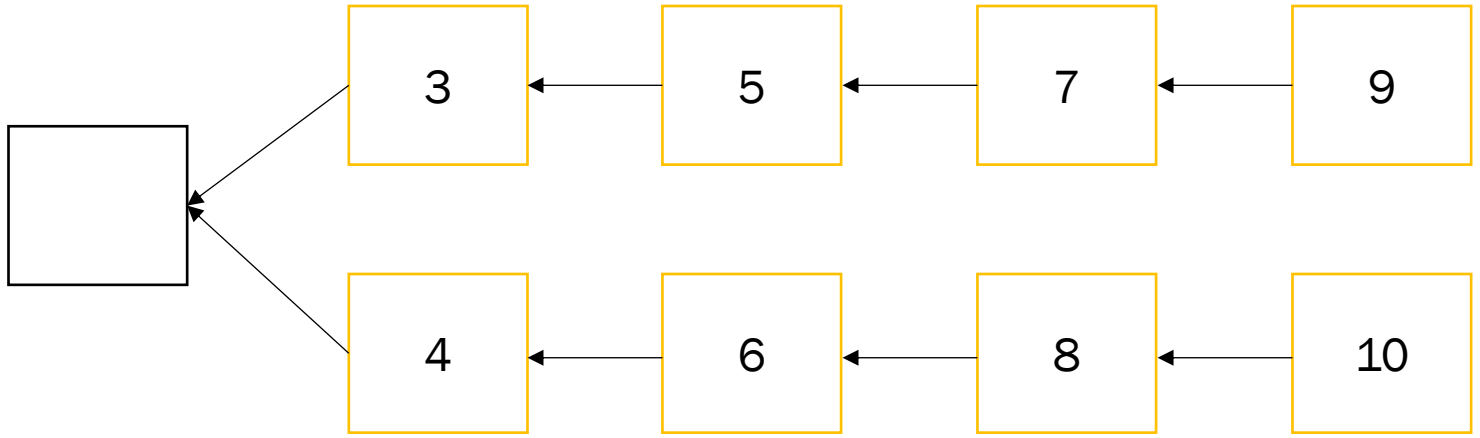
Streamlet Confirmation Rule

- Simple rule:
  - confirming a block as soon as it is notarized is not safe



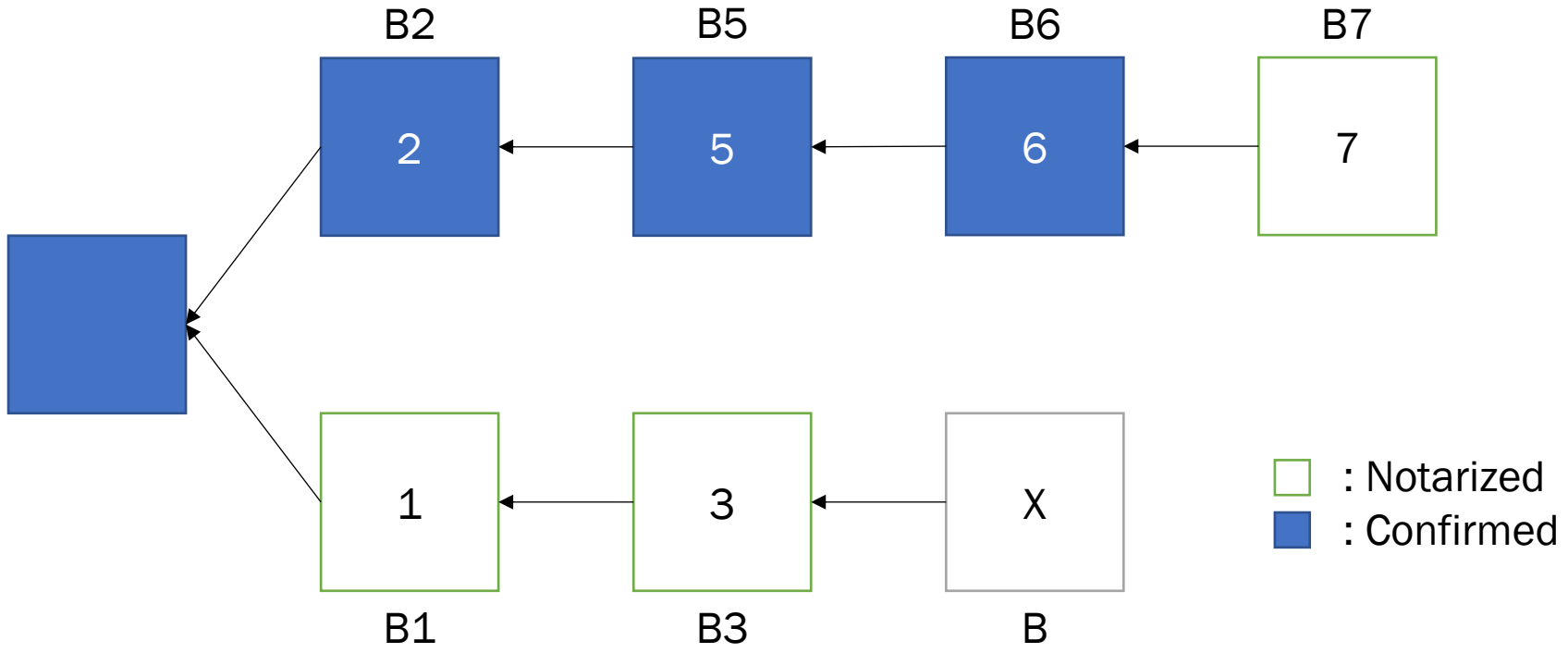
Streamlet Confirmation Rule

- Simple rule:
  - confirming a k-deep notarized block is not safe



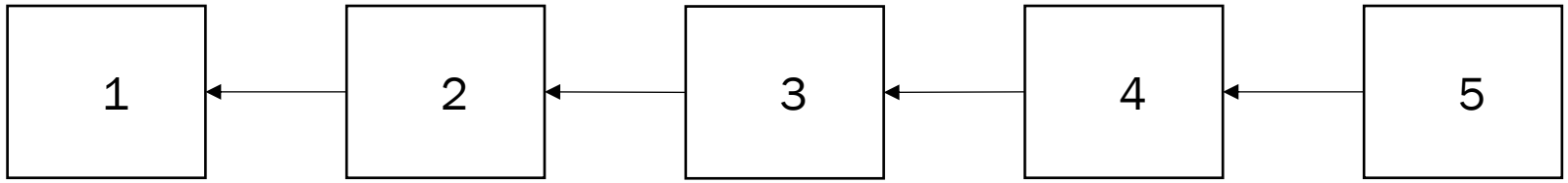
Streamlet Confirmation Rule

- Correct rule: On seeing three adjacent blocks in a notarized blockchain with consecutive round numbers, a player can confirm the second of the three blocks, and its entire prefix chain.



Streamlet Liveness

When network conditions are good, Streamlet makes progress whenever there are five consecutive rounds whose leaders are all honest.



Streamlet Performance

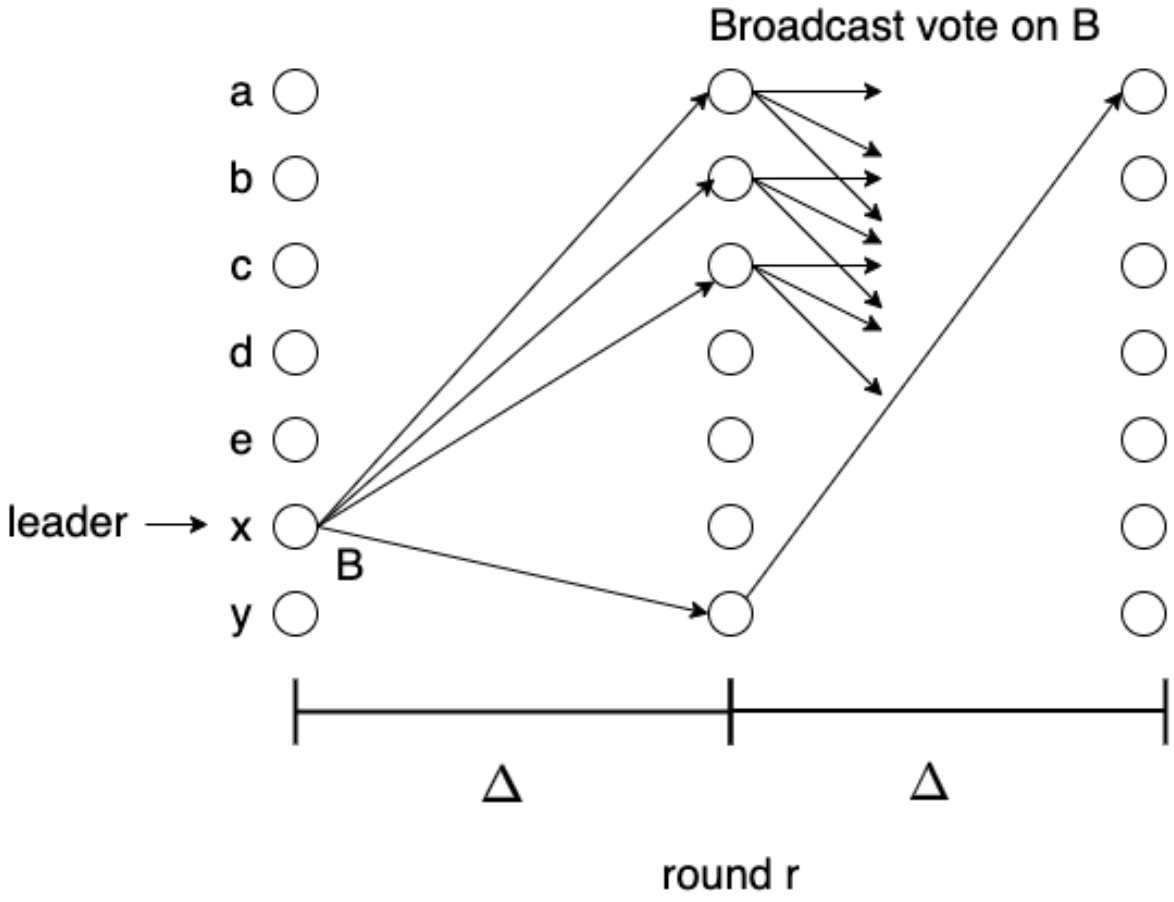
- Communication complexity
- Confirmation latency

Communication Complexity

Implicit echoing: upon observing a new transaction or message, a node always echos the transaction or message to everyone else.

- Echoing incurs n^3 voting messages per block
- Reducing the communication cost is non-trivial

Communication Complexity



Source: https://dahliamalkhi.github.io/posts/2020/12/what-they-didnt-teach-you-in-streamlet/

Streamlet Complexity

- Lower bound: At least n−1 messages are needed to spread the block among all nodes
- Linearity: communication complexity that is linear in the number of nodes
- Streamlet is not linear

Streamlet Latency

- Guarantee liveness whenever there are 5 consecutive honest proposers
  - Happens on an average once every $\frac{1}{(\frac{2}{3})^5} \approx 7.6$ rounds, about 15Δ
  - Recall Bitcoin latency is $O(log_e\left(\frac{1}{\epsilon}\right)\frac{1}{\lambda\Delta})\Delta$ where ϵ is the confirmation error probability and $\lambda\Delta \ll 1$ for security

Clock Synchronization

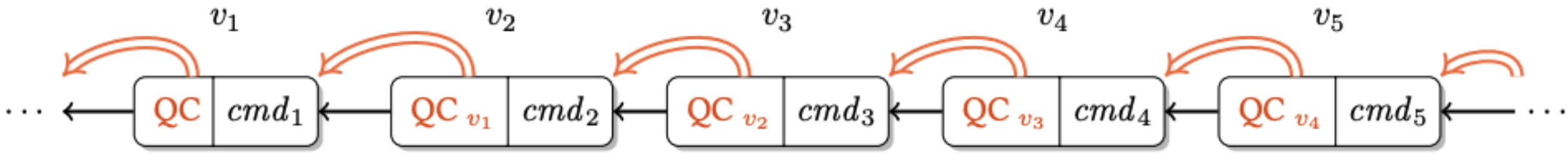
- Responsiveness: the ability to advance at the speed of the actual network delays without waiting maximal network delays
- Streamlet is not responsive

HotStuff

- State of the art BFT consensus protocol
- Similar to, but proposed earlier than, Streamlet
- HotStuff is linear and responsive

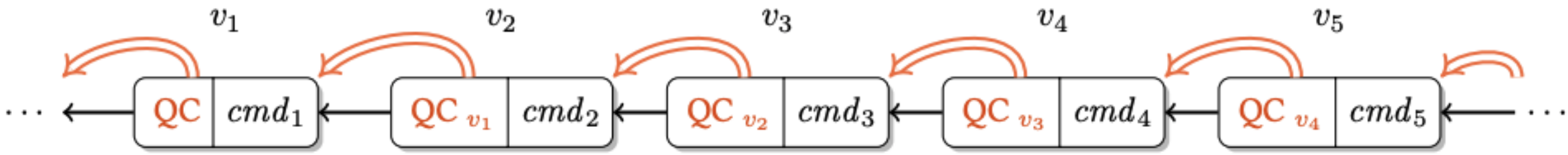
HotStuff

- n =3f+1, f Byzantine nodes
- Quorum certificate (QC): 2f+1 votes on one block
- QCs are on chain



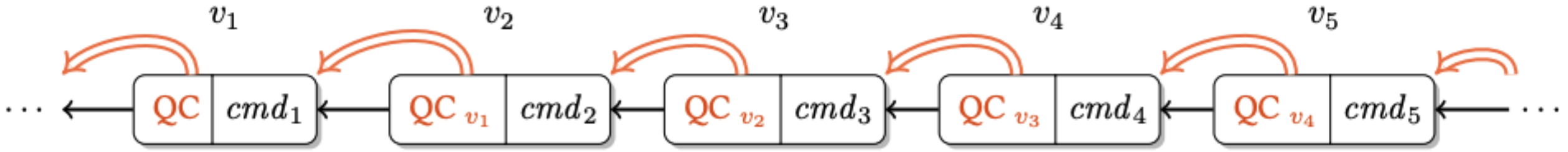
HotStuff Difference

- Each block is linked to its parent via the parent's quorum certificate (QC)
- A proposer can propose a new block right after it receives a new QC



From Streamlet to Hotstuff

- HighQC: Every node keeps the QC with the highest round number it knows of
- As in Streamlet
  - A leader proposal extends highQC
  - A node votes for a leader proposal if it extends the branch of its highQC, but only sends the vote to the leader of next round → Linearity
- Finalization rule: same as in Streamlet:



Clock Synchronization

- HotStuff does not require round synchronization for safety
- Time driven → event driven
- HotStuff is responsive

Implementation: Pacemaker

- Pacemaker: electing proposers
- Guarantees:
  - Infinitely often, all nodes spend a certain period of time jointly in a round
  - A unique correct proposer is elected for the round
- Naive way:
  - Double the round size until a decision is made
  - Round robin leader rotation

BFT Protocols summary

- Blockchain protocols with finality: Streamlet and HotStuff
  - Permissioned: fixed number of participants with known identities
  - Security: liveness is weakened in the pursuit of strengthening safety guarantee to a finality
- Relatively simple extension to the longest chain protocol

BFT Protocol Setting

- Number of Participants is fixed
- Identities known (signatures) to all nodes
- In other words, “permissioned”
- How do we go to permissionless?
  - Unknown number of participants
  - Arbitary identities?

Permissionless BFT Protocols

- Known upper bound on participants, known identities but variable participation at any time
- “Strategy”: elect a committee of N nodes from the set of participants.
  - The election can be verified in a distributed manner
  - Can be implemented via hash functions, and randomness from within the blockchain
- The committee proposes a block and implements consensus via a BFT protocol.

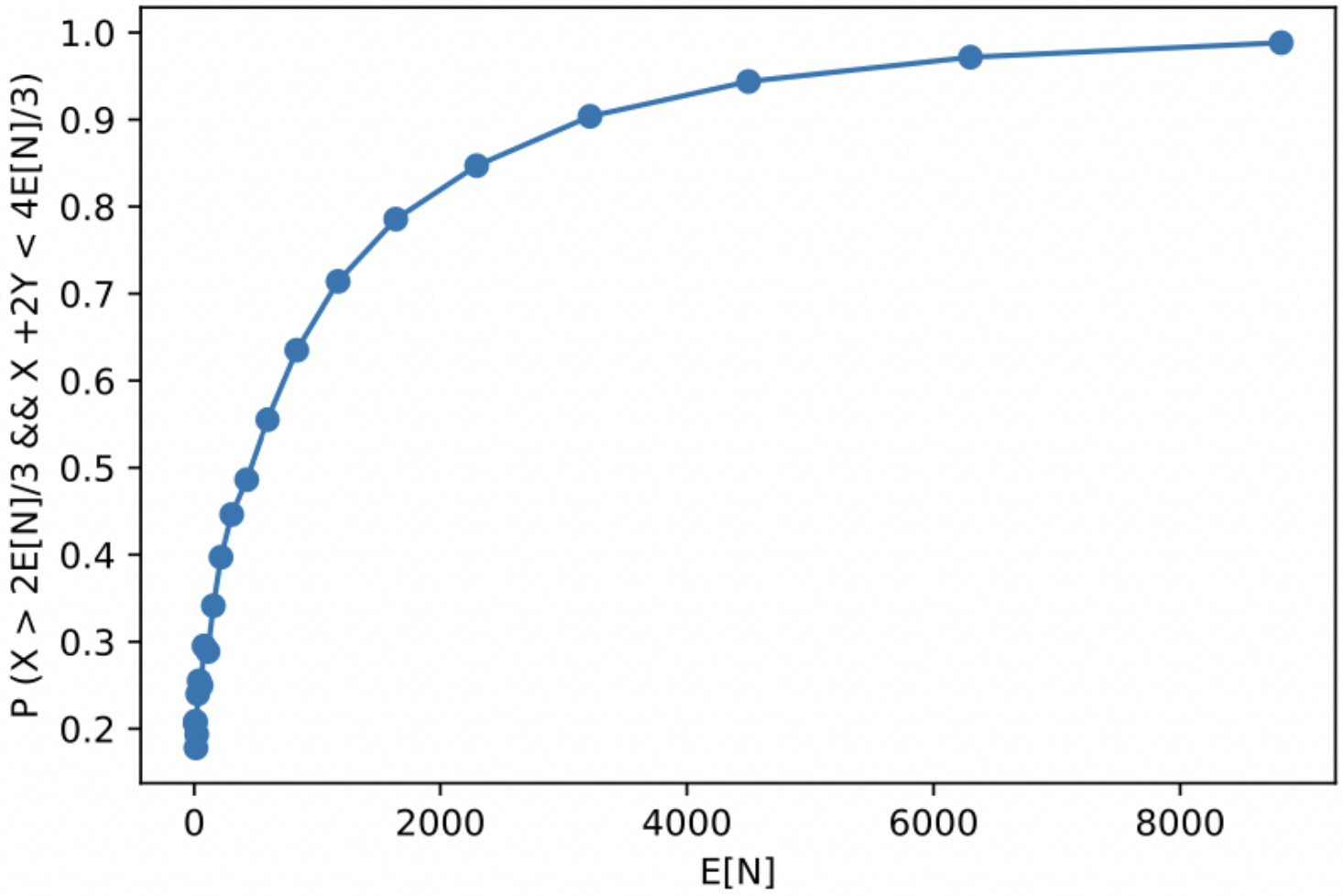
Random Committee Election

- As long as 2/3rd of the committee is live and honest, the BFT protocol is secure
- Electing a committee of fixed size (N) is hard
- Easier to elect a committee of random size
  - with desired mean value (N)
  - Hash functions

Random Committee Election in Practice

- Electing a committee of fixed size (N) is hard
- Easier to elect a committee of random size
- E.g., Total 1M users, E[N] = 1000, p = 0.001
  - X = #honest in committee ~ Binomial(700K, 0.001)
  - Y = #Byzantine in committee ~ Binomial(300K, 0.001)
  - Liveness: $X > 2N/3$
  - Safety: $X + 2Y < 4N/3$

Probabilities of Safety and Liveness



Vulnerability to Adaptive Adversary

- Adversary is static
  - So random number of Byzantine players
- Adaptive adversary
  - Can turn Byzantine after being elected into the committee
  - Long range attacks (can turn adversarial much later in time)
  - Fatal
- Need for randomness even within actions of adversary
  - **Player replaceability**
  - Separate topic: Algorand