
CONSENSYS
QBFT
Presentation for the EEA
July 2021

Agenda

- Architecture of the algorithm (10-15 min) - Roberto
- Implementation (format messages, sub-protocol, etc.) (10-15 min) - Jason/Jitu
- Demo of a Hybrid Network (send transactions, add/remove validator) (10min) - Jason/Jitu
- Q&A

Overview

QBFT was developed in collaboration between JP Morgan and ConsenSys protocol development teams based on experience and learning from IBFT and IBFT 2.0

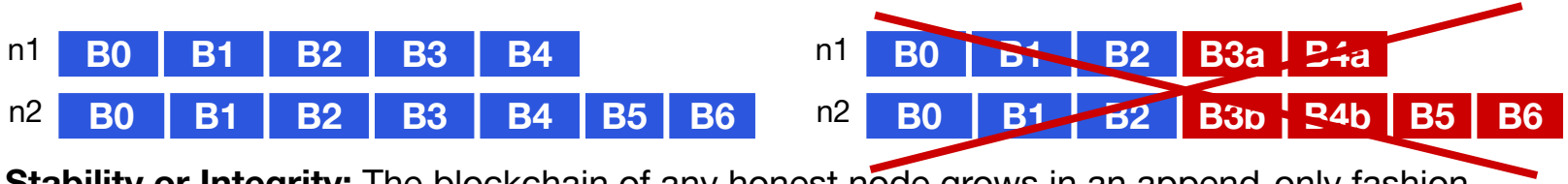
This new consensus mechanism has been developed in the latest release of Hyperledger Besu and GoQuorum.

Background

What is a BFT blockchain consensus protocol

It is a protocol that ensures the following properties:

- **Consistency or Agreement:** The blockchains of any two honest nodes are one the prefix of the other



- **Stability or Integrity:** The blockchain of any honest node grows in an append-only fashion
- **Liveness:** Any transaction submitted to the systems will eventually be included in the blockchain of all honest nodes

despite a fraction of the nodes (Byzantine) being able to collude to attack the system.

The maximum theoretical Byzantine resilience is $f = floor\left(\frac{n-1}{3}\right)$

n	*f*
3	0
4	1
5	1
6	1
7	2

There was once upon a time IBFT...

First proposal for an enterprise BFT blockchain consensus protocol

- A couple of issues were discovered with the protocol
  - Sub-optimal BFT resilience
  - Potential for the protocol to stop producing blocks

IBFT

Iteration on IBFT

- Fixes the issues identified with the IBFT protocol

IBFT2

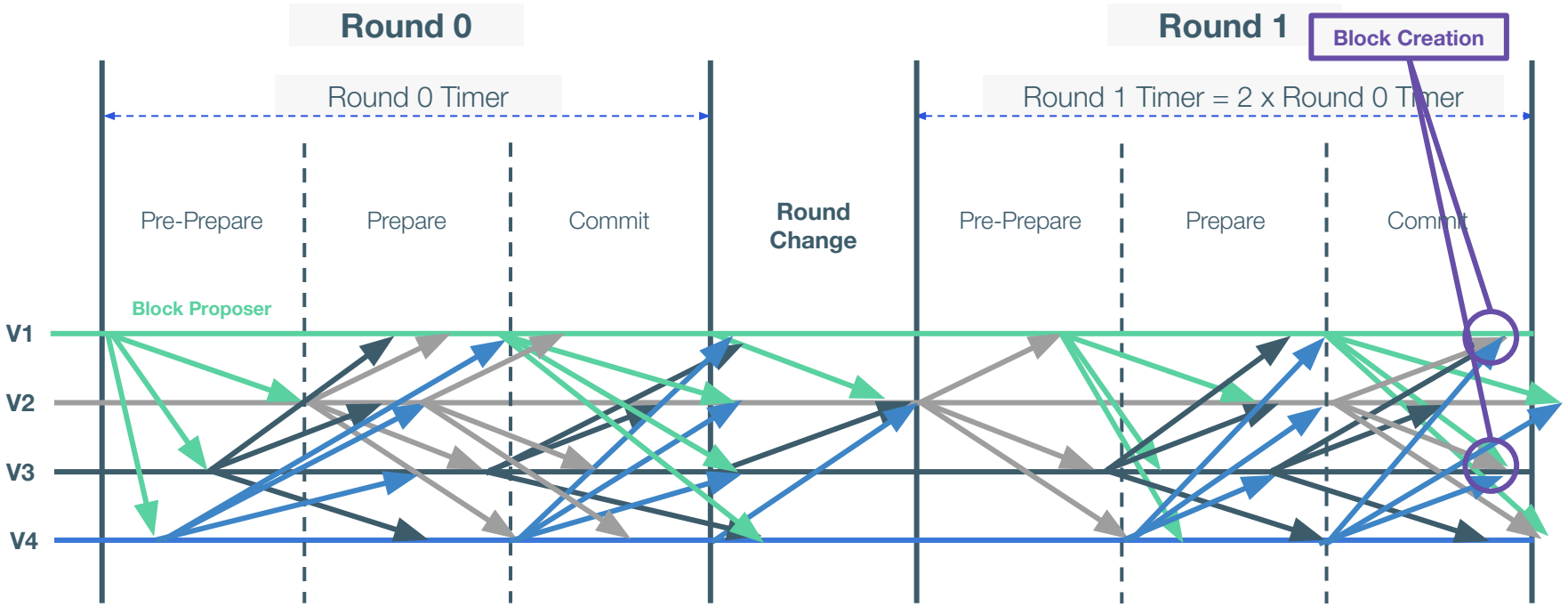
Reduction of the number of messages exchanged

- Round Change message complexity reduced from n^3 to n^2
- Formal protocol specifications

Original Idea: Henrique Moniz
https://arxiv.org/abs/2002.03613

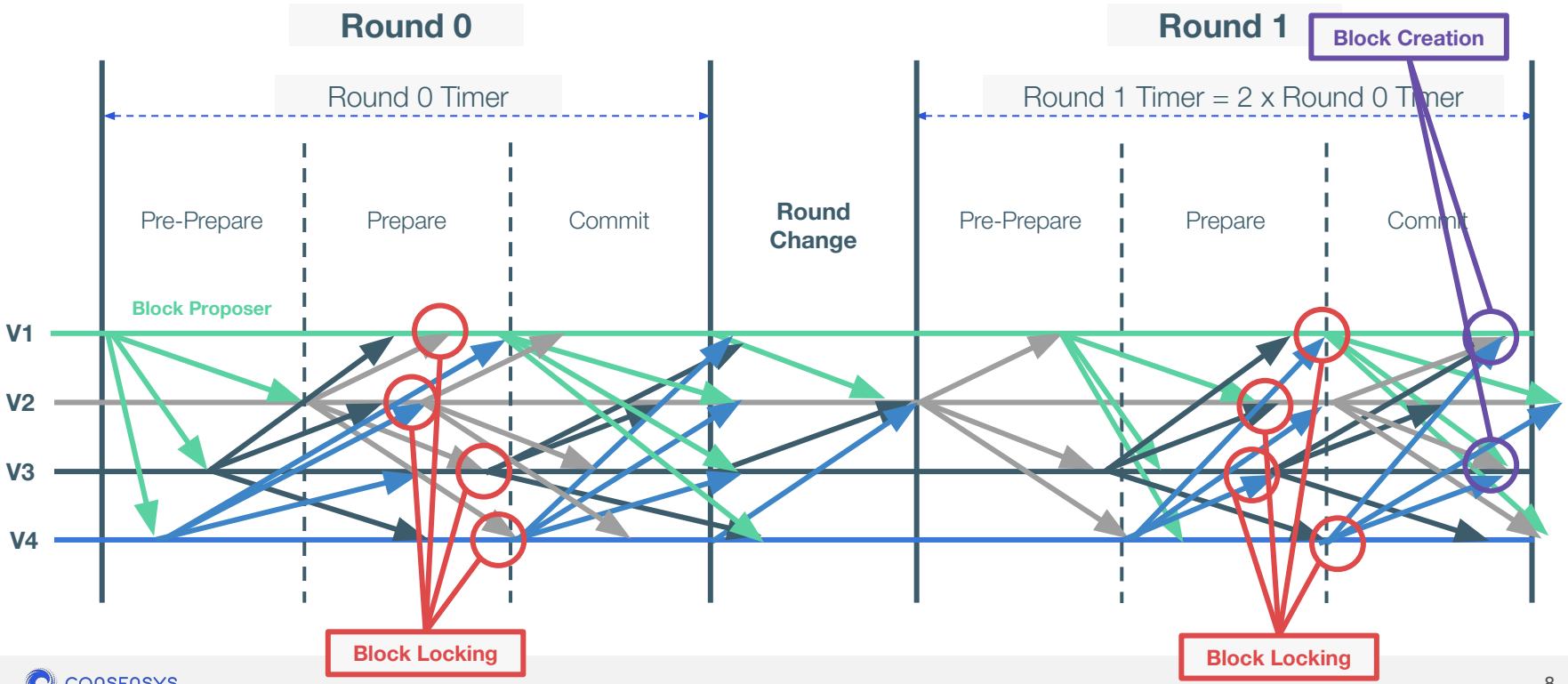
QBFT

IBFT/IBFT2/QBFT - Message Exchange Overview



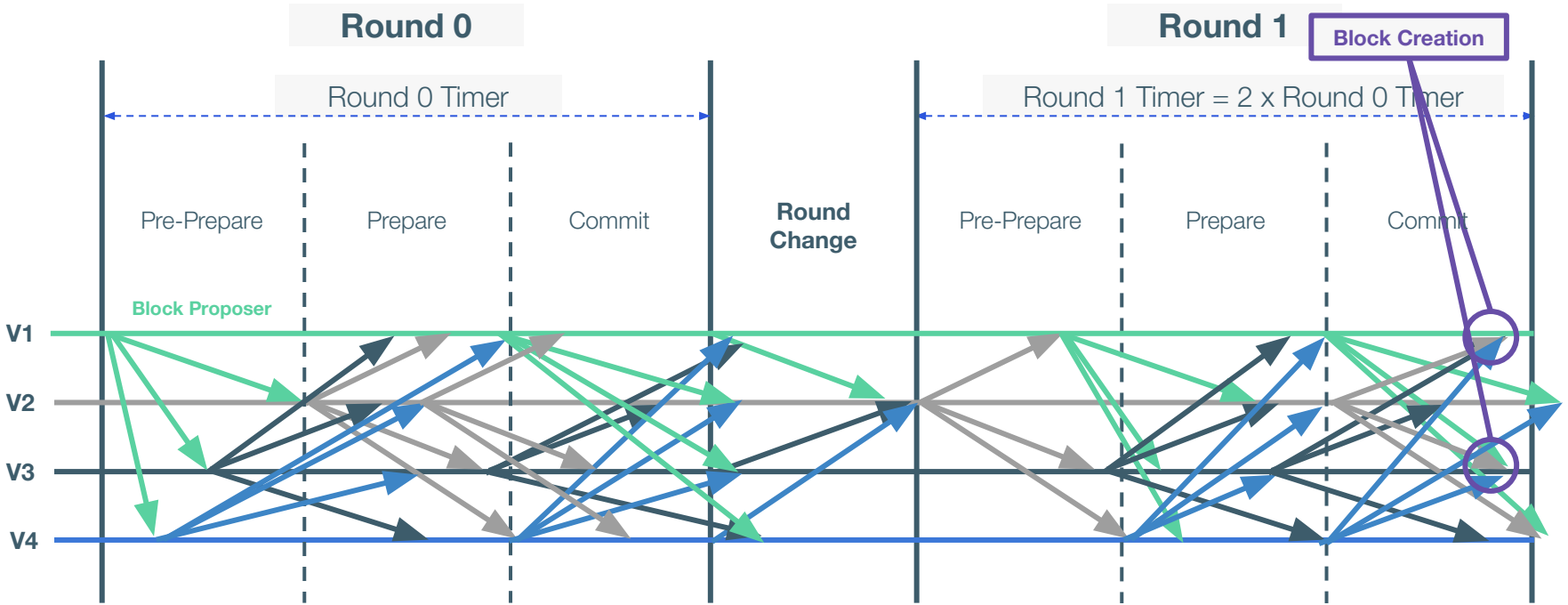
IBFT vs IBFT2 - Block Locking

- IBFT has the concept of block locking



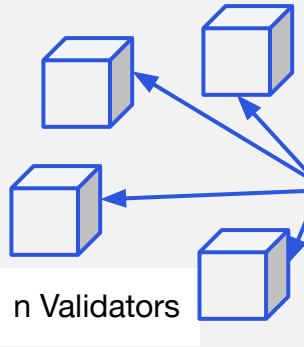
IBFT vs IBFT2 - Block Locking

- IBFT2 removes block locking



IBFT2 vs QBFT
How is the lower message complexity achieved?

IBFT2



Proposal
Sequence, Round
Block
⅔ Round Change

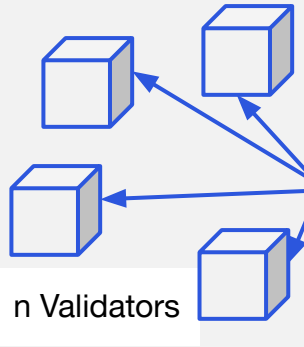
Round Change
Sequence, Round
Block Hash
⅔ Prepares

2n/3 Round Change * 2n/3 Prepares *

n =

$O(n^3)$

QBFT



Proposal
Sequence, Round
Block
⅔ Round Change
⅔ Prepares

Round Change
Sequence, Round
Block Hash
Prepared Round

(2n/3 Round Change + 2n/3 Prepares) *

n =

$O(n^2)$

Specification

Specification

Objectives

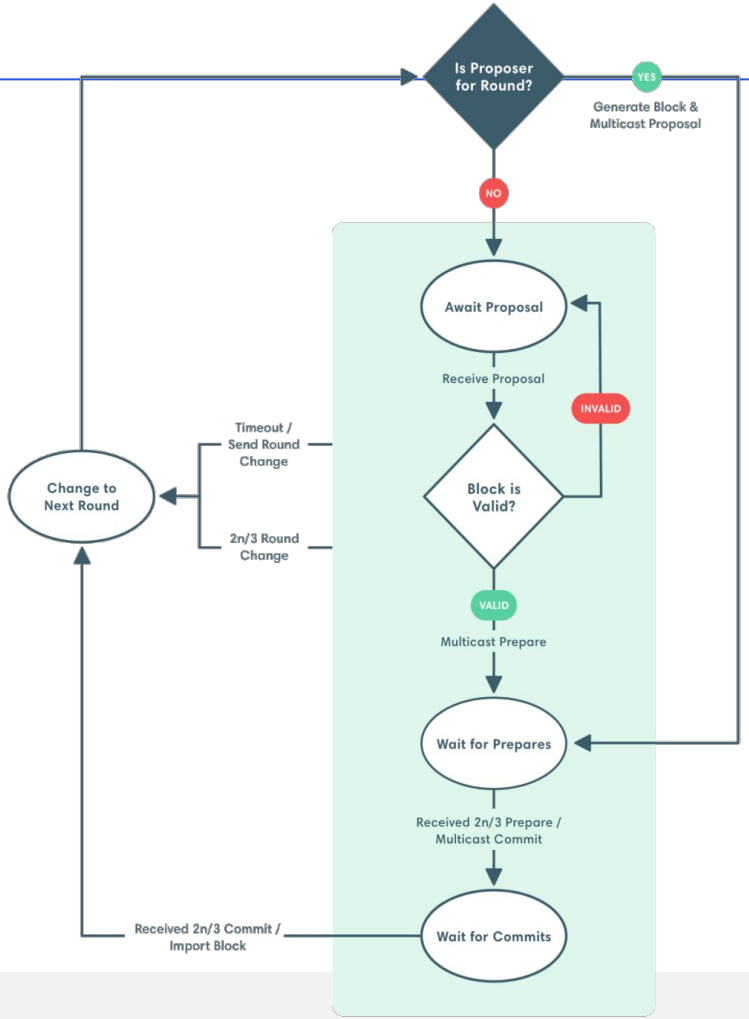
- Unambiguous Protocol Definition
- General
  - Allow for more than one compatible implementation
- Suitable for mechanised verification

Status

- Draft Version Available on the GitHub repo
- Formal Verification Ongoing
- This is not a blocker to run testnets

Implementation

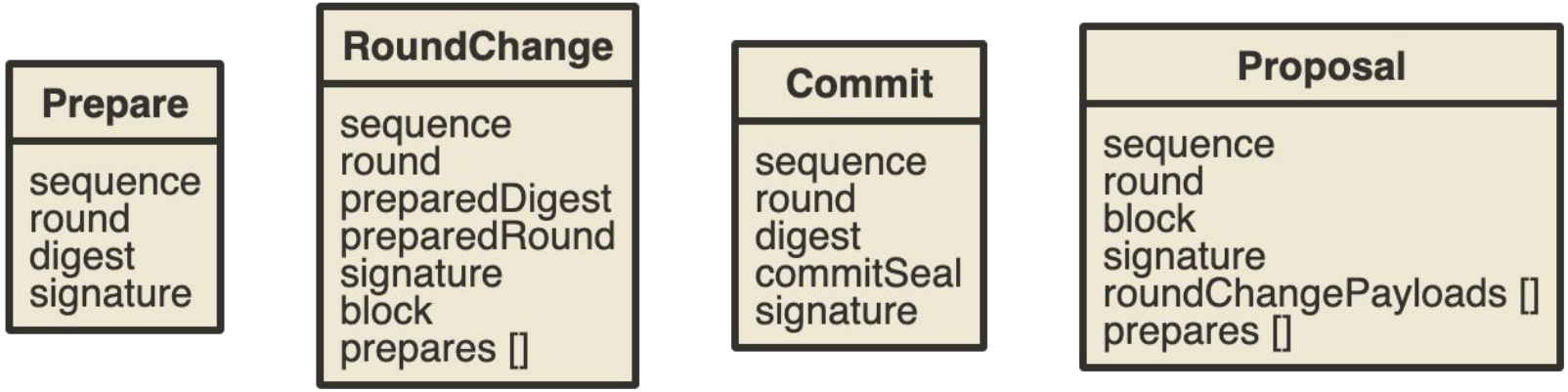
Message Workflow



Messages overview

- Message shapes
  - Signatures
    - All messages are signed by the transmitting node to prevent byzantine behaviour
  - Round specific
    - Every message specifies which Round it targets
- Messages are 1st class citizens
  - Need to be ordered (prior, current, future)
  - Validated
  - Gossipped
    - Messages are sent to peers using devp2p on the istanbul/100 subprotocol

Messages types



RoundChangePayload in the proposal is the signed RoundChange message without the block or prepare messages.

QBFT header details

- Similarly to IBFT and IBFT2 the block header extra data field contains
    - RLP encoded data containing seal, current set of validators, round block was sealed on and optionally a vote
- Mix hash value has a fixed value same as IBFT
- Unlike in IBFT the nonce and beneficiary fields are not used

Validators

- Can be added and removed by voting
- RPCs to add and remove validators
- Each validator maintains a vote tally and includes a vote when they next propose
- To change the validators set a quorum of (n/2) + 1 votes is required

Demo

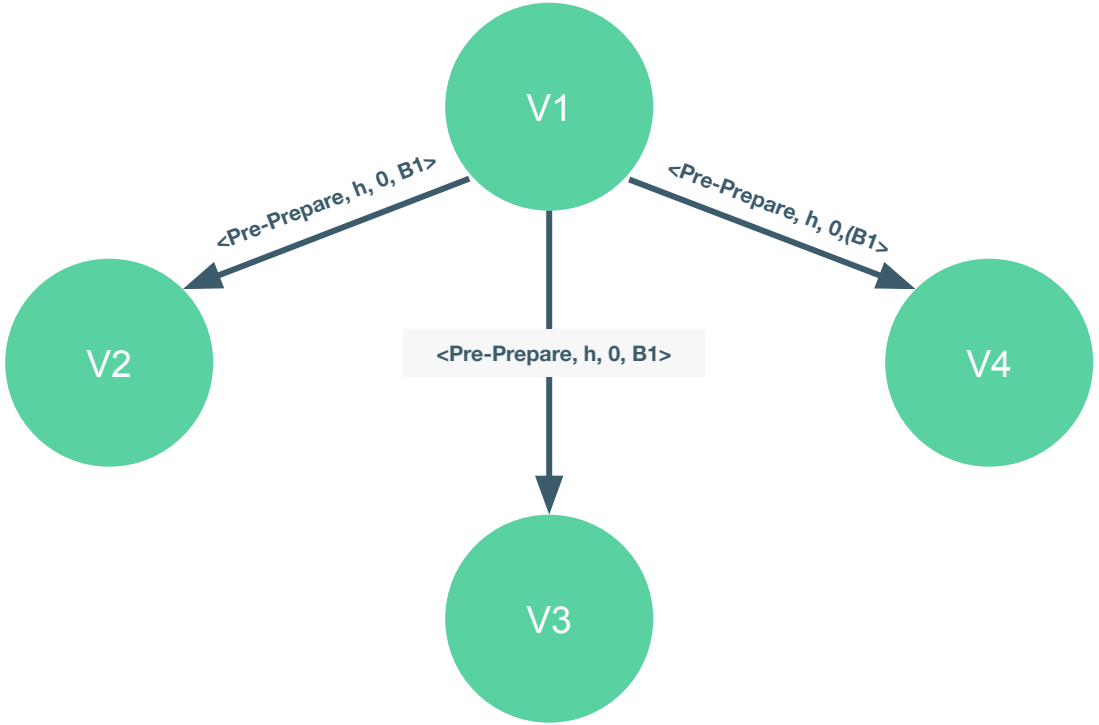
- 4 node interop network with 2 Besu and 2 Quorum nodes
- All nodes are validators

Questions

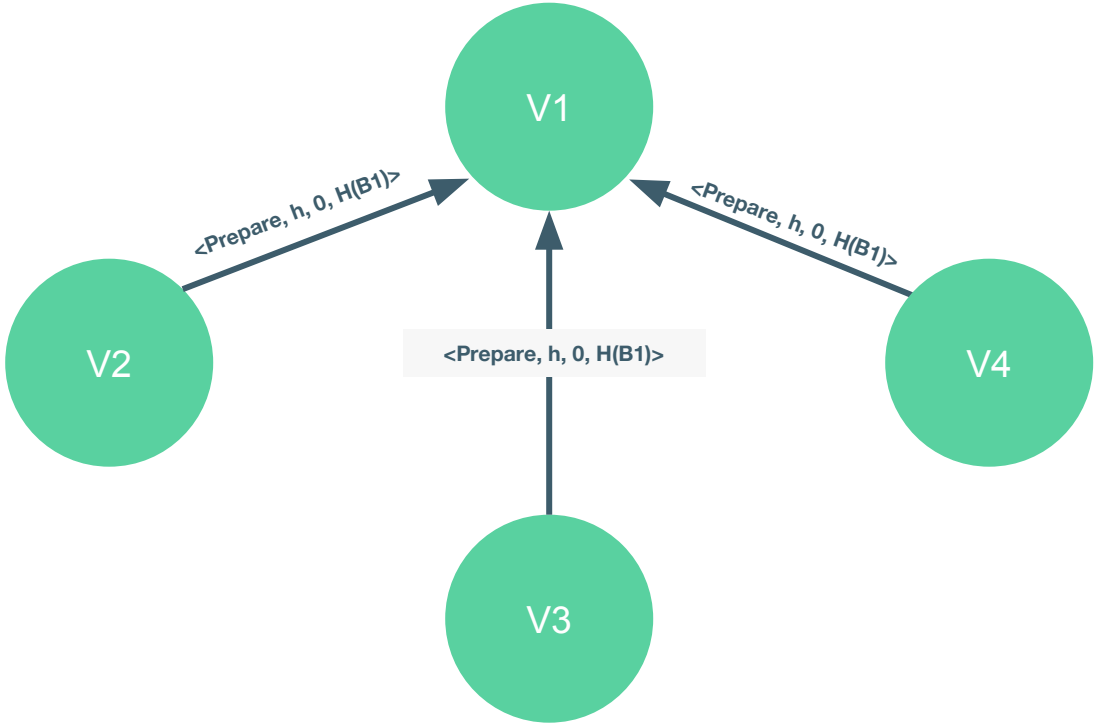
Appendix

IBFT Liveness Issue

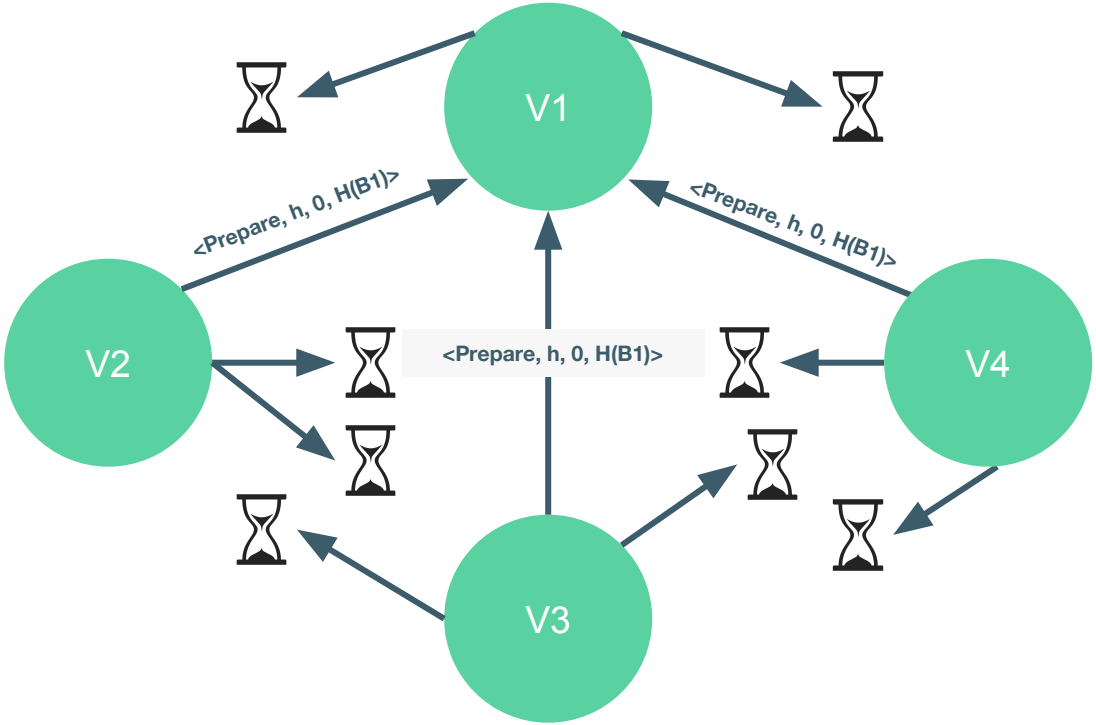
No termination is guaranteed if 1 validator stops



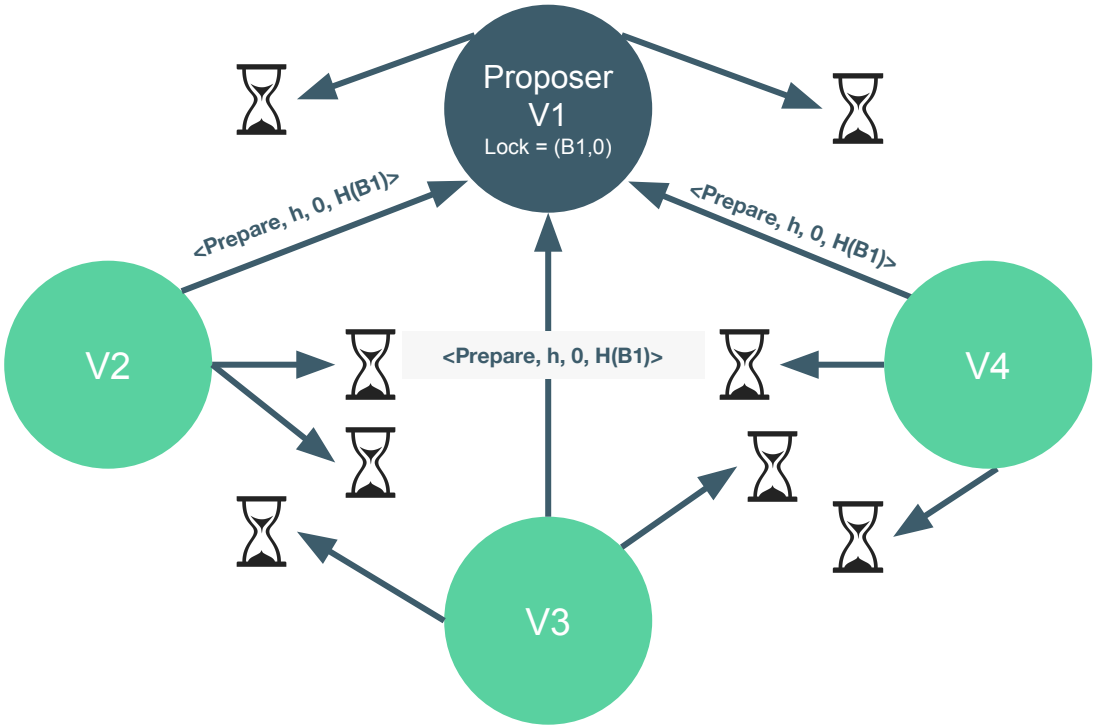
No termination is guaranteed if 1 validator stops



No termination is guaranteed if 1 validator stops



No termination is guaranteed if 1 validator stops

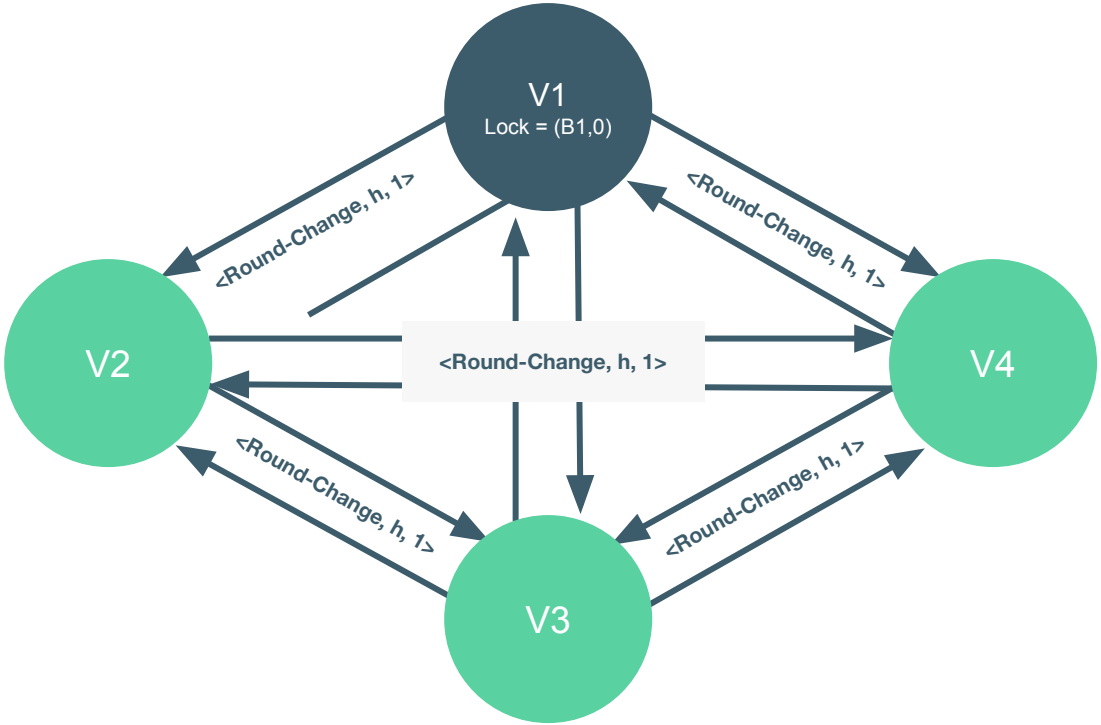


Formal Analysis of IBFT

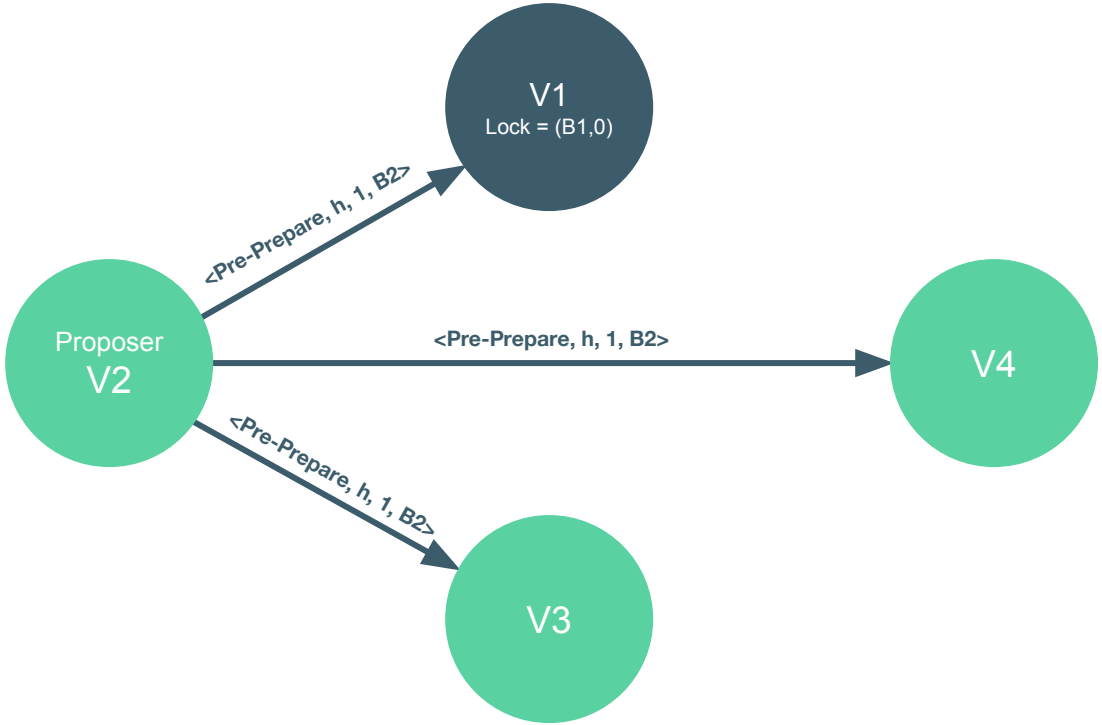
- Results
  - No termination is guaranteed if 1 validator stops

Round Timer Expired!!!

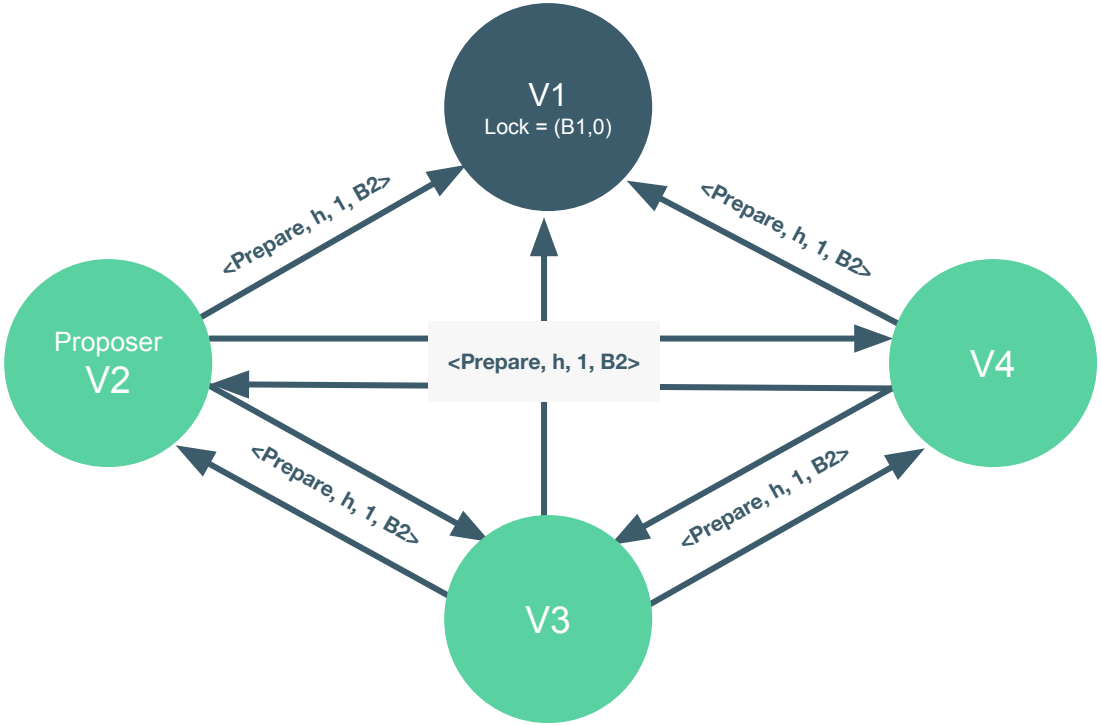
No termination is guaranteed if 1 validator stops



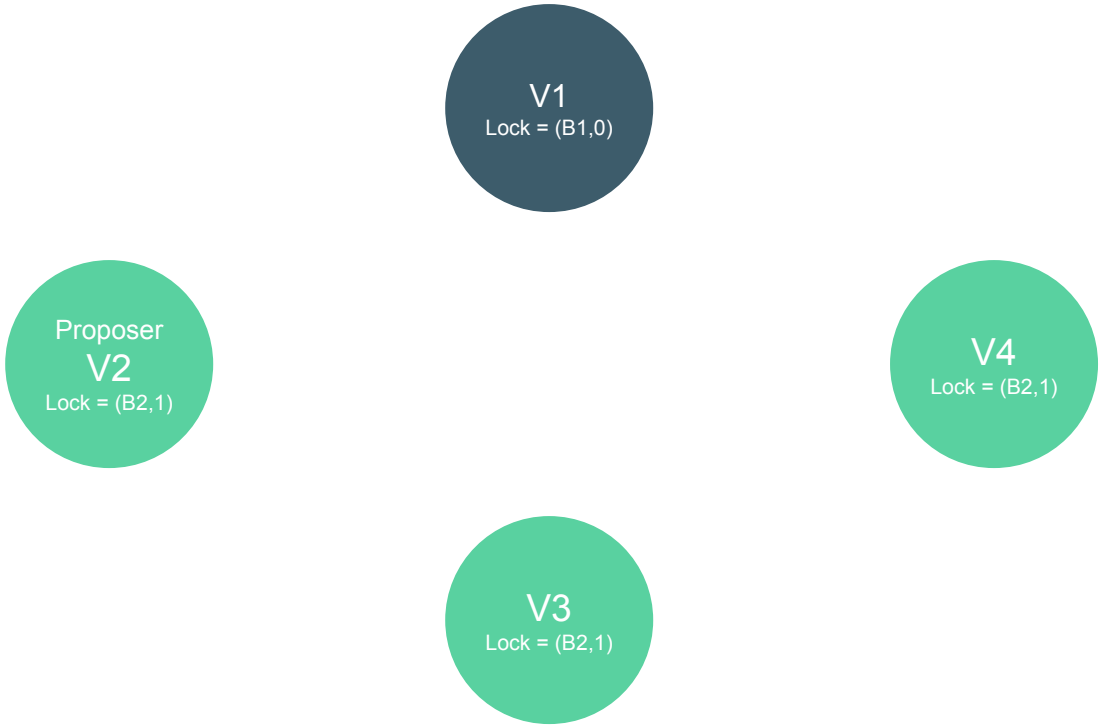
No termination is guaranteed if 1 validator stops



No termination is guaranteed if 1 validator stops



No termination is guaranteed if 1 validator stops



No termination is guaranteed if 1 validator stops

V1
Lock = (B1,0)

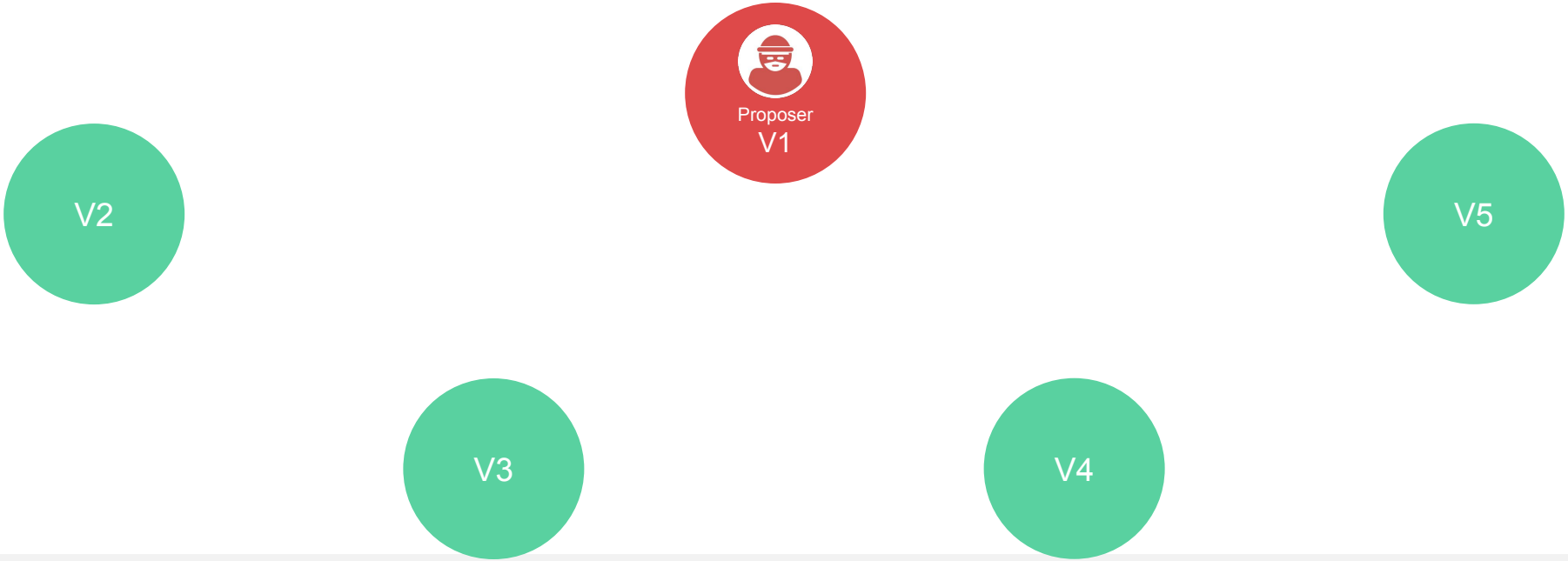
Proposer
V2
Lock = (B2,1)

V3
Lock = (B2,1)

IBFT Safety Issue

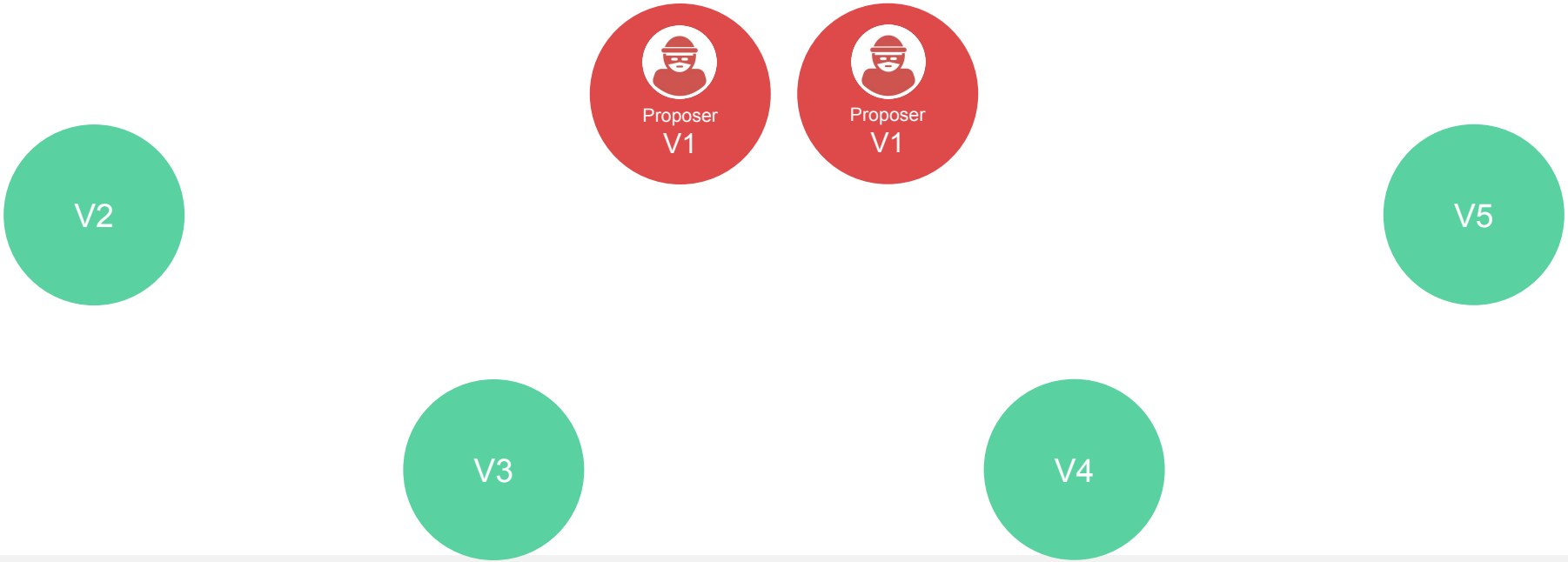
No optimal Byzantine-fault-tolerance

Minimum number of validators required by IBFT to participate in the protocol to decide on the next block to add to the blockchain: 3



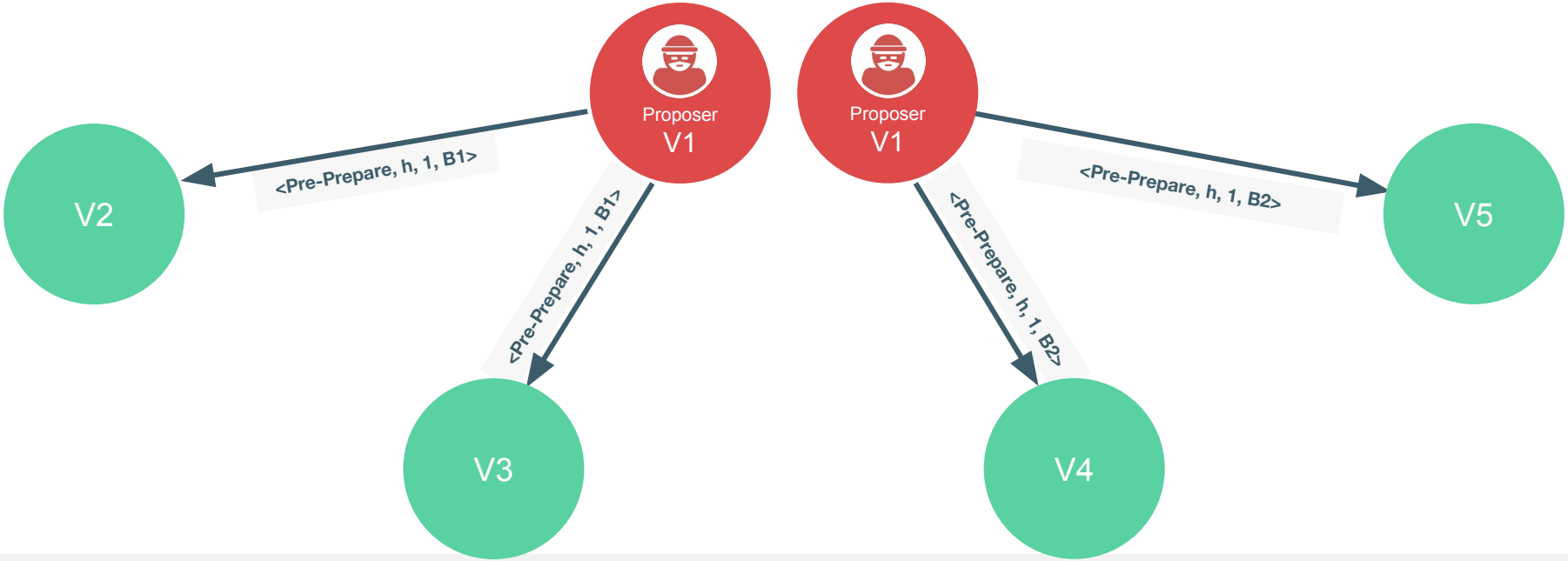
No optimal Byzantine-fault-tolerance

Minimum number of validators required by IBFT to participate in the protocol to decide on the next block to add to the blockchain: 3



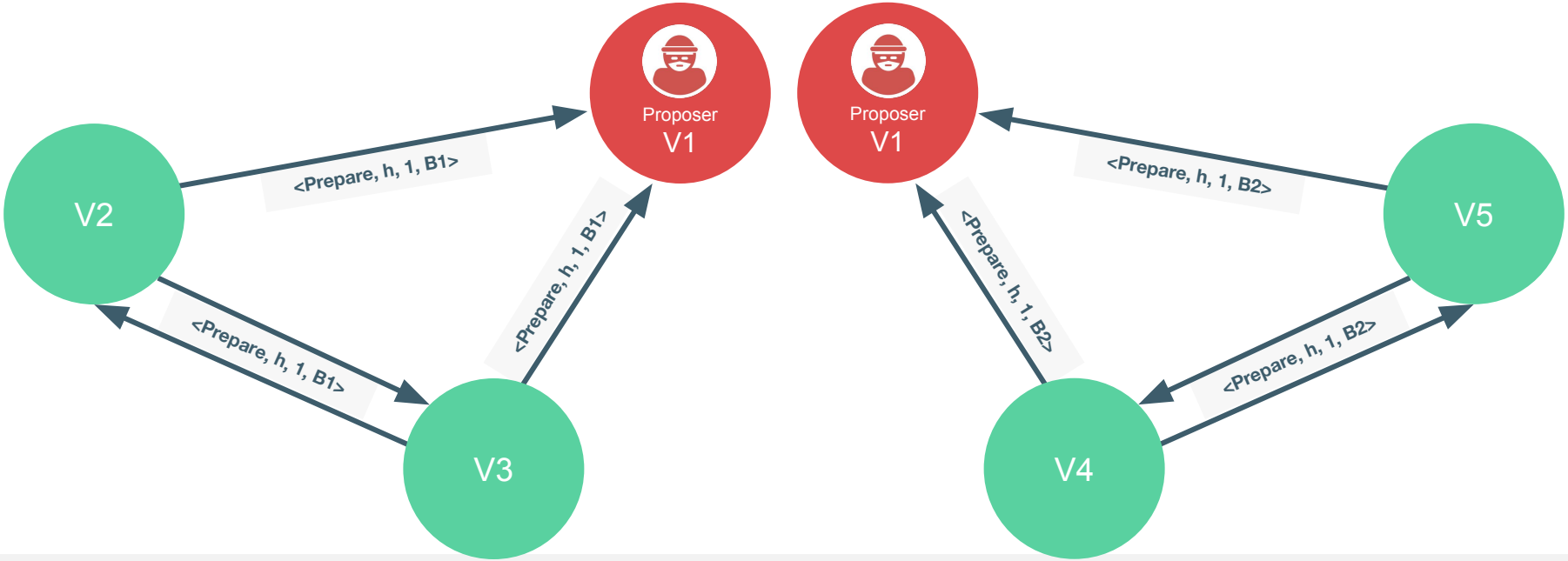
No optimal Byzantine-fault-tolerance

Minimum number of validators required by IBFT to participate in the protocol to decide on the next block to add to the blockchain: 3



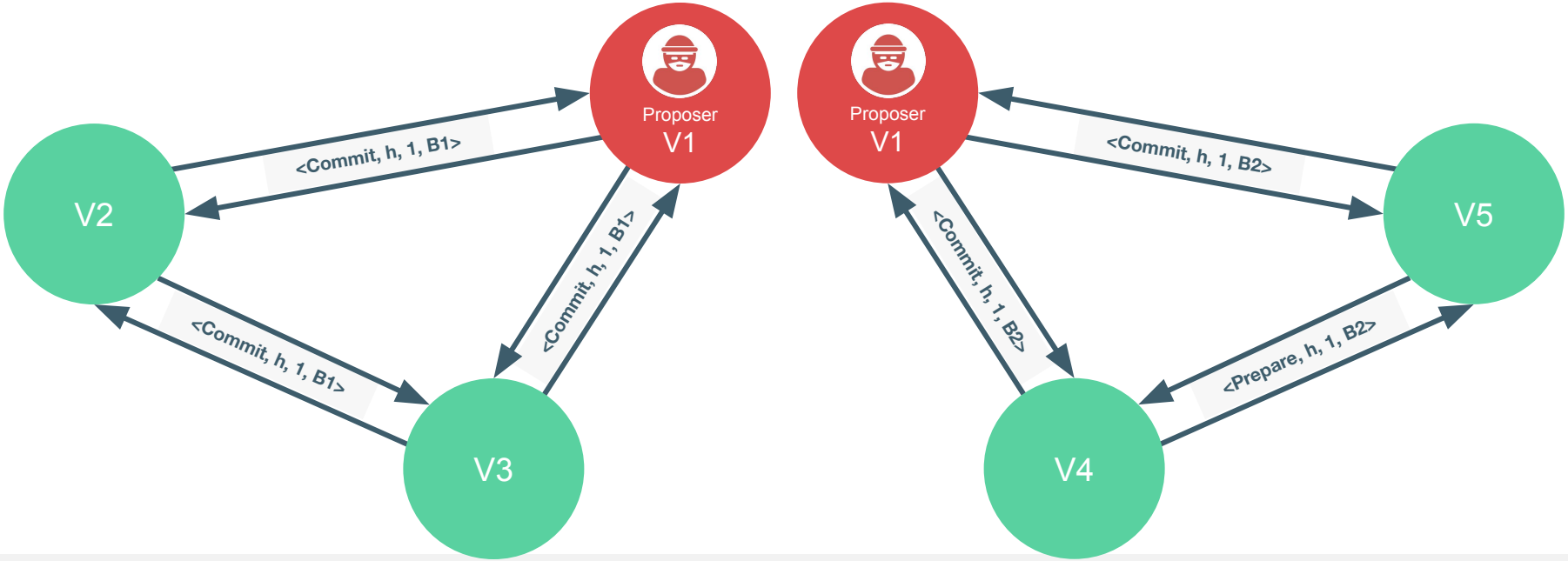
No optimal Byzantine-fault-tolerance

Minimum number of validators required by IBFT to participate in the protocol to decide on the next block to add to the blockchain: 3



No optimal Byzantine-fault-tolerance

Minimum number of validators required by IBFT to participate in the protocol to decide on the next block to add to the blockchain: 3



No optimal Byzantine-fault-tolerance

Minimum number of validators required by IBFT to participate in the protocol to decide on the next block to add to the blockchain: 3