# Formal Analysis of a Proof-of-Stake Blockchain

Wai Yan M. M. Thin, Naipeng Dong, **Guangdong Bai**, Jin Song Dong

National University of Singapore

Griffith University

Dec 12, 2018

# Outline

- Problem Statement
- Background
- Tendermint Consensus Algorithm
- Formal Analysis
- Conclusion

# Problem Statement

- Consensus protocols and algorithms are being developed rapidly
- They are fundamental to the chains
- Formal analysis of these consensus protocols is necessary

# Background

# Background

•Blockchain – sequence of blocks

•Block – maintains the metadata (the hash value of itself, link to the previous block, signatures) and payload

•Consensus algorithm – protocol used by the nodes in the network to agree on a new block

# Consensus Algorithms

## Proof-of-work

- Nodes provide the proof by solving a mathematical problem (e.g. Bitcoin)
- Rewarded for performing an operation agreed by majority
- Not punished for performing a malicious operation
- E.g. Bitcoin

## Proof-of-stake

- Nodes provide a stake for voting/validating a new block
- Stakes are slashed if a malicious activity is detected
- E.g. Ethereum's Casper, Tendermint

## Others: Delegated Proof-of-stake , Proof-of-burn …

# Focus on Proof-of-stake

- **Proof-of-work**
  - Scalability concerns
  - Waste energy and resources (solving hash puzzles)
- **Proof-of-stake**
  - Alternative to the wasteful proof-of-work
  - More scalable and robust against certain attacks (E.g. 51% attack)
  - Employed by popular blockchain systems - Peercoin, Ethereum's Casper, Tendermint (Cosmos)

# Tendermint Consensus Algorithm

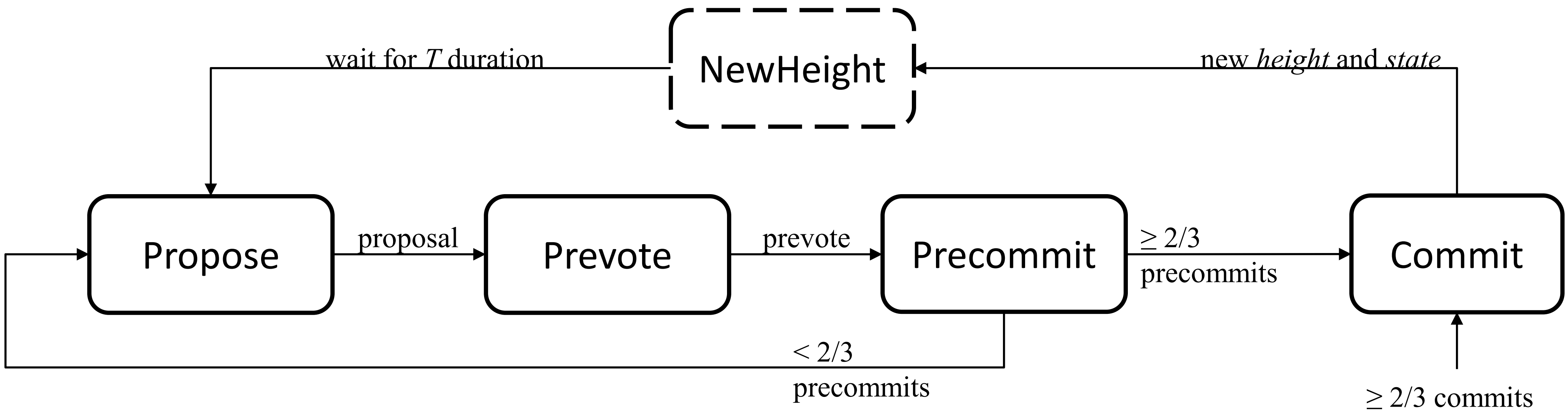
# Tendermint Consensus Algorithm

- **Proposals**
  - A new block must be proposed by the correct proposer at each round, and gossiped to the other validators
- **Votes**
  - Two phases of voting occur to ensure optimal Byzantine fault tolerance: *pre-vote* and *pre-commit*
- **Locks**
  - Prevent two different blocks to be committed at two different rounds at the same height

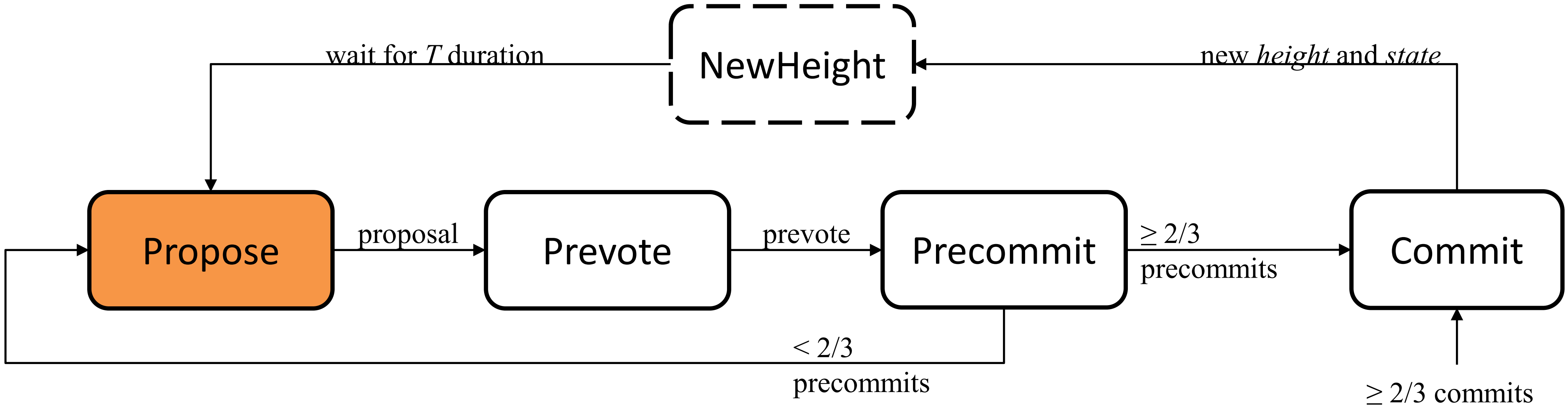
# Tendermint Consensus Algorithm

- **Validators chosen in round-robin to become the proposer**
- **Proposer in charge of proposing a block for the current round**
- **Proposer/validators**
  - Receive proposal/votes from neighbours
  - Validate the block in proposal/votes
  - Post a bond transaction to vote
  - *Gossip* the proposal/votes

# Tendermint Consensus Algorithm



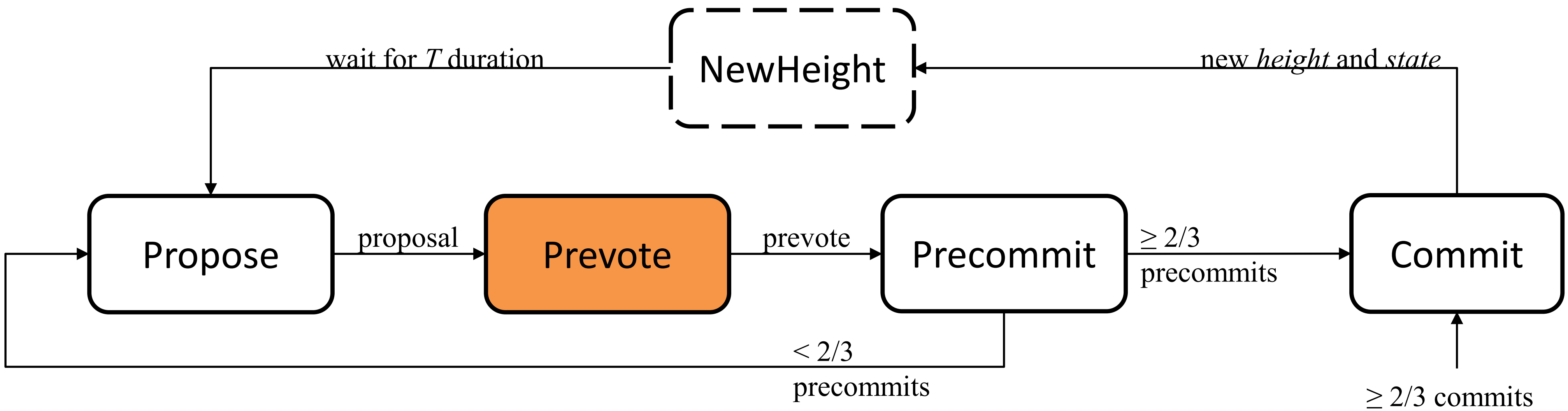
# Tendermint Consensus Algorithm



# Tendermint Consensus Algorithm: Propose

- **Proposer broadcasts a proposal to its peers**
- **If the proposer has already locked on a block during the *Precommit* of the *previous* round**
  - Propose the block
- **Otherwise**
  - Create a new block

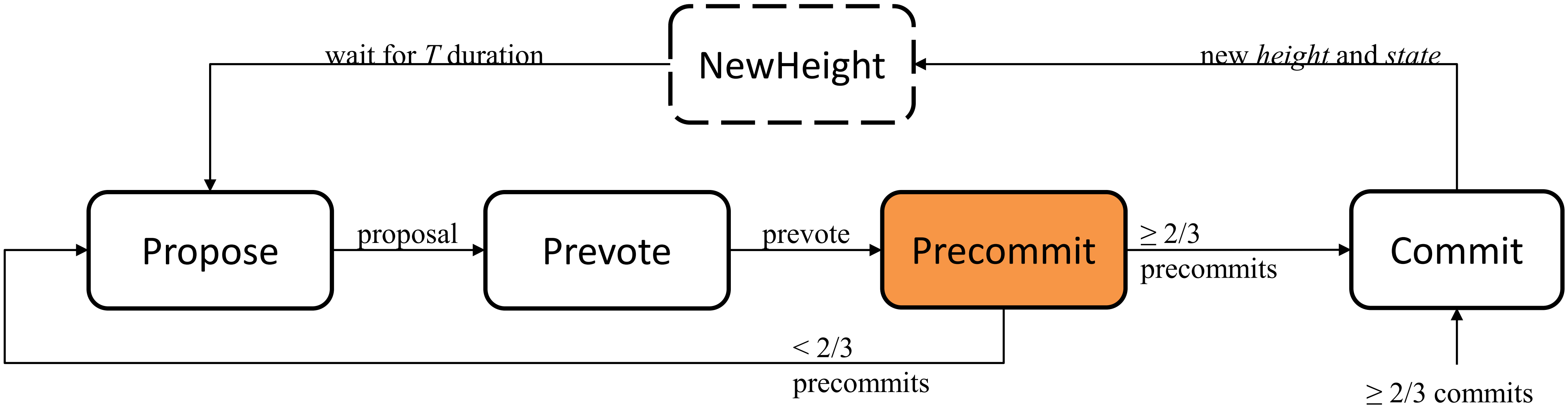
# Tendermint Consensus Algorithm



# Tendermint Consensus Algorithm: Prevote

- **Each validator will vote for a block and gossip it to the neighbours.**
- **The block to be included is chosen in the following order:**
  - A locked proposed block from prior rounds
  - A valid acceptable block from the current proposal
  - NIL if neither is available

# Tendermint Consensus Algorithm



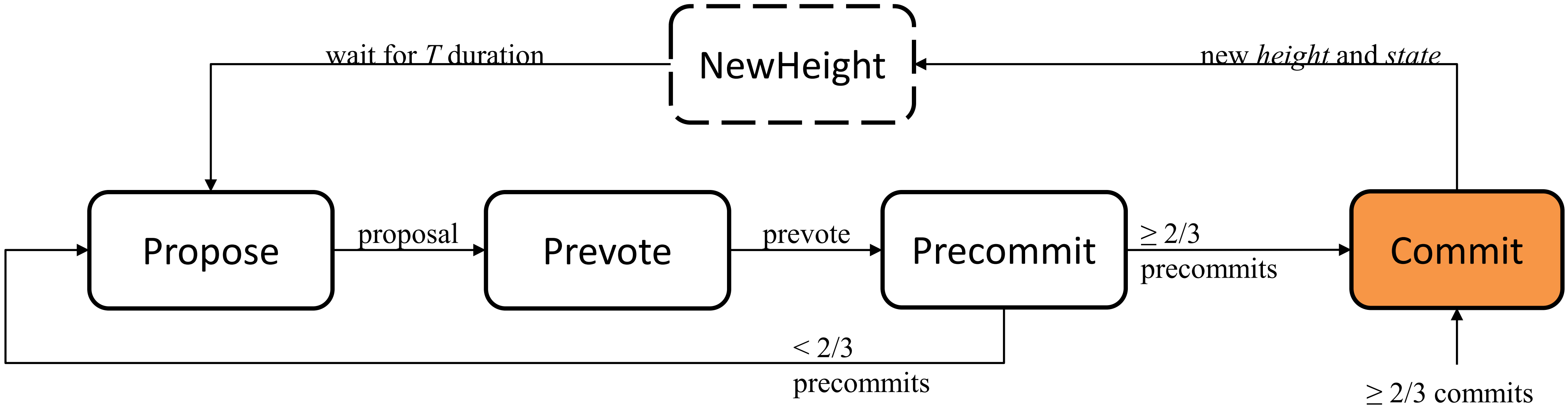
# Tendermint Consensus Algorithm: Precommit (1/2)

- **If validator has more than 2/3 of prevotes for an acceptable block**
  - Releases the existing lock
  - Locks onto this block
  - Signs and broadcasts a precommit vote for this block
  - Packages the prevotes for the locked block into a proof-of-lock
- **Otherwise**
  - Neither signs nor locks on any block

# Tendermint Consensus Algorithm: Precommit (2/2)

- **If received more than 2/3 of precommits for a block**
  - Proceed to *Commit* phase for this round
- **Otherwise**
  - Proceed to *Propose* phase for next round

# Tendermint Consensus Algorithm



# Tendermint Consensus Algorithm: Commit

- **Receive the block from one of its peers**
- **Sign and broadcast a commit to other peers**
- **When > 2/3 commits of the block are received by the network**
  - Proceed to *NewHeight*
  - Wait for a fixed duration to receive additional commits of the block
  - Proceed to *Propose*
- **At anytime during the protocol, if > 2/3 commits for a particular block is received,**
  - Proceed to *Commit*

# Modelling & Checking

# Modelling & Checking

- **Built using CSP# and verified using PAT model checker**
- **Two sets of verifications with 3 validators and 4 validators**
- **Assumptions**
  - All the nodes in the network are connected to each other
  - Existing nodes will not leave the network and no new nodes will join the network
  - All nodes have the same voting power/stake
  - No network latency

# Properties

1. **Deadlockfree-ness (T1)**
2. **Ability to reach consensus  (T2)**
3. **Immunity against block overwrites (A1)**
4. **Immunity against Invalid blocks (A2)**
5. **Immunity against Censorship attacks (A3)**
   - The network can reach consensus even with the absence of malicious nodes in the voting process who refuse to broadcast or vote a valid block in order to censor a particular content of the block or censor the node itself

# Modelling

```
BlockChain() = (||x:{0..N-1} @ (Propose(x);
Prevote(x); Precommit(x); PreparePOL(x); Commit(x)));
NextRound();
```

```
where  P ; Q  → process P followed by process Q
       P || Q → synchronous processes P and Q.
```

# Attacker Models (1/3)

**P0. BlockChain()**

**P1. BlockChainWithMinorityOverwrite()**

```
SimulateMalicious(MINORITY, OVERWRITE_VOTING); BlockChain();
```

**P2. BlockChainWithHalfOverwrite()**

```
SimulateMalicious(HALF, OVERWRITE_VOTING); BlockChain();
```

**P3. BlockChainWithMajorityOverwrite()**

```
SimulateMalicious(MAJORITY, OVERWRITE_VOTING); BlockChain();
```

# Attacker Models (2/3)

**P4. BlockChainWithMinorityInvalid()**

```
SimulateMalicious(MINORITY, INVALID_BLOCK_VOTING); BlockChain();
```

**P5. BlockChainWithHalfInvalid()**

```
SimulateMalicious(HALF, INVALID_BLOCK_VOTING); BlockChain();
```

**P6. BlockChainWithMajorityInvalid()**

```
SimulateMalicious(MAJORITY, INVALID_BLOCK_VOTING); BlockChain();
```

# Attacker Models (3/3)

**P7. BlockChainWithMinorityCensor()**

```
SimulateMalicious(MINORITY, NO_VOTING); BlockChain();
```

**P8. BlockChainWithHalfCensor()**

```
SimulateMalicious(HALF, NO_VOTING); BlockChain();
```

**P9. BlockChainWithMajorityCensor()**

```
SimulateMalicious(MAJORITY, NO_VOTING); BlockChain();
```

# Verification Results

**Deadlockfree-ness (T1)**
**Ability to reach consensus (T2)**
**Immunity against block overwrites (A1)**
**Immunity against Invalid blocks (A2)**
**Immunity against Censorship attacks (A3)**

| | T1 | T2 | A1 | A2 | A3 |
|---|---|---|---|---|---|
| P0 BlockChain | ✔ | ✔ | ✔ | ✔ | ✔ |
| P1 (overwrite ≤ 1/3) | ✔ | ✔ | ✔ | | |
| P2 (1/3 < overwrite < 2/3) | ✔ | ✗ | ✔ | | |
| P3 (overwrite ≥ 2/3) | ✔ | ✔ | ✗ | | |
| P4 (invalid ≤ 1/3) | ✔ | ✔ | | ✔ | |
| P5 (1/3 < invalid < 2/3) | ✔ | ✗ | | ✔ | |
| P6 (invalid ≥ 2/3) | ✔ | ✗ | | ✔ | |
| P7 (no_vote ≤ 1/3) | ✔ | ✔ | | | ✔ |
| P8 (1/3 < no_vote < 2/3) | ✔ | ✗ | | | ✗ |
| P9 (no_vote ≥ 2/3) | ✔ | ✗ | | | ✗ |

# Benchmarks (1/3)

| Deadlock-free | | BlockChain | MinorityForking | HalfForking | MajorityForking | MinorityInvalid | HalfInvalid | MajorityInvalid | MinorityCensor | HalfCensor | MajorityCensor |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Visited States | 3 Validators | 748 | 749 | | 749 | 749 | | 749 | 598 | | 873 |
| | 4 Validators | 17,644 | 17,645 | 17,645 | 17,645 | 17,645 | 17,645 | 17,645 | 3,249 | 865 | 423 |
| | 5 Validators | 4,279,260 | 4,279,261 | 4,279,261 | 4,279,261 | 4,279,261 | 4,279,261 | 4,279,261 | 314,709 | 4,125 | 1,335 |
| | 6 Validators | | | | | | | | | | |
| Transistions | 3 Validators | 1,972 | 1,973 | | 1,973 | 1,973 | | 1,973 | 1,385 | | 1,824 |
| | 4 Validators | 103,000 | 103,001 | 103,001 | 103,001 | 103,001 | 103,001 | 103,001 | 13,201 | 2,385 | 937 |
| | 5 Validators | 42,530,784 | 42,530,785 | 42,530,785 | 42,530,785 | 42,530,785 | 42,530,785 | 42,530,785 | 2,431,909 | 15,629 | 3,853 |
| | 6 Validators | | | | | | | | | | |
| Time Taken(s) | 3 Validators | 0.06 | 0.06 | | 0.05 | 0.06 | | 0.05 | 0.04 | | 0.04 |
| | 4 Validators | 3.52 | 3.48 | 3.47 | 3.49 | 3.42 | 3.22 | 3.43 | 0.47 | 0.07 | 0.03 |
| | 5 Validators | 1486.10 | 1430.37 | 1454.97 | 1531.58 | 1638.59 | 1512.43 | 1504.90 | 89.90 | 0.52 | 0.11 |
| | 6 Validators | | | | | | | | | | |
| Memory Used (MB) | 3 Validators | 138.99 | 144.44 | | 138.95 | 143.75 | | 142.66 | 138.22 | | 140.59 |
| | 4 Validators | 146.39 | 143.03 | 145.44 | 146.81 | 143.69 | 140.24 | 144.29 | 140.67 | 140.60 | 137.79 |
| | 5 Validators | 624.86 | 109.29 | 116.63 | 166.40 | 460.14 | 84.17 | 77.92 | 121.55 | 14.52 | 14.95 |
| | 6 Validators | | | | | | | | | | |

| LEGEND | |
|---|---|
| (orange) | Property being verified |
| (blue) | BlockChain Model |
| (green) | Verified TRUE |
| (pink) | Verified FALSE |
| (grey) | Verification Invalid |
| (yellow) | Verification not run due to state space complexity |

| Distribution of validators | | | |
|---|---|---|---|
| Validators | Minority | Half | Majority |
| 3 | 1 | - | 2 |
| 4 | 1 | 2 | 3 |
| 5 | 1 | 3 | 4 |
| 6 | 2 | 3 | 4 |

# Benchmarks (2/3)

| Consensus | | BlockChain | MinorityForking | HalfForking | MajorityForking | MinorityInvalid | HalfInvalid | MajorityInvalid | MinorityCensor | HalfCensor | MajorityCensor |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Visited States | 3 Validators | 66 | 67 | | 67 | 67 | | 750 | 65 | | 881 |
| | 4 Validators | 142 | 143 | 17,646 | 143 | 143 | 17,646 | 17,646 | 129 | 866 | 424 |
| | 5 Validators | 266 | 267 | 4,279,262 | 267 | 267 | 4,279,262 | 4,279,262 | 239 | 4,126 | 1,336 |
| | 6 Validators | 450 | 451 | | 451 | 451 | | | 335 | 131,274 | 4,480 |
| | | | | | | | | | | | |
| Transitions | 3 Validators | 65 | 66 | | 66 | 66 | | 1,973 | 64 | | 1,824 |
| | 4 Validators | 141 | 142 | 103,001 | 142 | 142 | 103,001 | 103,001 | 128 | 2,385 | 937 |
| | 5 Validators | 265 | 266 | 42,530,785 | 266 | 266 | 42,530,785 | 42,530,785 | 238 | 15,629 | 3,853 |
| | 6 Validators | 449 | 450 | | 450 | 450 | | | 334 | 931,969 | 21,648 |
| | | | | | | | | | | | |
| Time Taken(s) | 3 Validators | 0.01 | 0.01 | | 0.01 | 0.01 | | 0.05 | 0.01 | | 0.04 |
| | 4 Validators | 0.02 | 0.01 | 3.22 | 0.01 | 0.01 | 3.07 | 3.10 | 0.01 | 0.07 | 0.03 |
| | 5 Validators | 0.02 | 0.02 | 1573.97 | 0.02 | 0.02 | 1512.57 | 1583.24 | 0.01 | 0.58 | 0.12 |
| | 6 Validators | 0.05 | 0.03 | | 0.04 | 0.03 | | | 0.02 | 39.78 | 0.81 |
| | | | | | | | | | | | |
| Memory Used (MB) | 3 Validators | 138.11 | 138.19 | | 138.22 | 138.18 | | 143.05 | 137.98 | | 140.71 |
| | 4 Validators | 141.89 | 142.06 | 140.86 | 142.20 | 142.04 | 139.22 | 140.36 | 140.86 | 140.79 | 137.71 |
| | 5 Validators | 12.30 | 12.59 | 1007.11 | 12.96 | 12.55 | 524.55 | 690.11 | 15.10 | 15.56 | 15.47 |
| | 6 Validators | 143.81 | 146.76 | | 142.08 | 146.63 | | | 140.81 | 236.94 | 141.52 |

| LEGEND | |
|---|---|
| | Property being verified |
| | BlockChain Model |
| | Verified TRUE |
| | Verified FALSE |
| | Verification Invalid |
| | Verification not run due to state space complexity |

| Distribution of validators | | | |
|---|---|---|---|
| Validators | Minority | Half | Majority |
| 3 | 1 | - | 2 |
| 4 | 1 | 2 | 3 |
| 5 | 1 | 3 | 4 |
| 6 | 2 | 3 | 4 |

# Benchmarks (3/3)

| Forking Attack | | BlockChain | MinorityForking | HalfForking | MajorityForking |
|---|---|---|---|---|---|
| Visited States | 3 Validators | 66 | 67 | | 750 |
| | 4 Validators | 142 | 143 | 144 | 17,646 |
| | 5 Validators | 266 | 267 | 268 | 4,279,262 |
| | 6 Validators | 450 | 451 | | |
| | | | | | |
| Transistions | 3 Validators | 65 | 66 | | 1,973 |
| | 4 Validators | 141 | 142 | 143 | 103,001 |
| | 5 Validators | 265 | 266 | 267 | 42,530,785 |
| | 6 Validators | 449 | 450 | | |
| | | | | | |
| Time Taken(s) | 3 Validators | 0.01 | 0.01 | | 0.05 |
| | 4 Validators | 0.01 | 0.01 | 0.01 | 3.31 |
| | 5 Validators | 0.02 | 0.02 | 0.02 | 1618.90 |
| | 6 Validators | 0.03 | 0.03 | | |
| | | | | | |
| Memory Used (MB) | 3 Validators | 138.12 | 138.18 | | 139.91 |
| | 4 Validators | 141.90 | 142.07 | 142.16 | 141.01 |
| | 5 Validators | 12.29 | 12.58 | 12.80 | 492.65 |
| | 6 Validators | 146.06 | 146.71 | | |

| Invalid Block Insertion | | BlockChain | MinorityInvalid | HalfInvalid | MajorityInvalid |
|---|---|---|---|---|---|
| Visited States | 3 Validators | 68 | 69 | | 69 |
| | 4 Validators | 143 | 144 | 144 | 144 |
| | 5 Validators | 267 | 268 | 268 | 268 |
| | 6 Validators | | | | |
| | | | | | |
| Transistions | 3 Validators | 67 | 68 | | 68 |
| | 4 Validators | 142 | 143 | 143 | 143 |
| | 5 Validators | 266 | 267 | 267 | 267 |
| | 6 Validators | | | | |
| | | | | | |
| Time Taken(s) | 3 Validators | 0.01 | 0.01 | | 0.00 |
| | 4 Validators | 0.01 | 0.02 | 0.01 | 0.01 |
| | 5 Validators | 0.02 | 0.02 | 0.02 | 0.02 |
| | 6 Validators | | | | |
| | | | | | |
| Memory Used (KB) | 3 Validators | 138.13 | 138.21 | | 138.22 |
| | 4 Validators | 141.91 | 142.05 | 142.14 | 142.10 |
| | 5 Validators | 12.34 | 12.59 | 12.79 | 12.68 |
| | 6 Validators | | | | |

| LEGEND | |
|---|---|
| (orange) | Property being verified |
| (blue) | BlockChain Model |
| (green) | Verified TRUE |
| (pink) | Verified FALSE |
| (grey) | Verification Invalid |
| (yellow) | Verification not run due to state space complexity |

| Distribution of validators | | | |
|---|---|---|---|
| Validators | Minority | Half | Majority |
| 3 | 1 | - | 2 |
| 4 | 1 | 2 | 3 |
| 5 | 1 | 3 | 4 |
| 6 | 2 | 3 | 4 |

# Conclusions

- **We made a preliminary step towards the formal verification of consensus protocols**
  - We modelled the Tendermint consensus algorithm in CSP# with 10 models to simulate several attacks
  - We verified five preliminary properties using PAT
- **Additional measures are required to ensure the protocol can withstand censorship attacks**
- **Models available at https://goo.gl/Jzym4B**

# Future Works

- **Automatic formal verification is limited in verifying consensus protocols with larger numbers of nodes**
- **Current models and properties are restricted**
- **We are interested in**
  - Studying verification algorithms catered towards blockchains
  - Modelling sophisticated attacks and verifying more complex security properties

# Thank you