Analysis of the Blockchain Protocol in Asynchronous Networks

Rafael Pass     Lior Seeman     abhi shelat
Cornell Tech    Uber            Northeastern
Traditional distributed systems: The “Permissioned” Model

- Consistency
- Liveness

Paxos/PBFT
The “Permissionless” Model:
Bitcoin/Blockchain

The Times 03/Jan/2009
Chancellor on brink of second bailout for banks.
The “Permissionless” Model

- Nodes do not know each other a-priori
- Nodes come and go
- ANYONE can join
- No network synchronization
The “Permissionless” Model

- Strong impossibility results known in the “permissionless” ("unauthenticated") model [BCLPR05]
  - Consistency is impossible
  - Sybil attacks unavoidable.
  - [BCLPR05] defined “weakened” security model (w/o consistency)
Nakamoto’s Blockchain [Nak’08]

Prevents Sybil attacks with Proofs-of-Work Puzzles [DN’92]

**Claims** blockchain achieves “public ledger” assuming “honest majority of computing power”:

- **Consistency**: everyone sees the same history
- **Liveness**: everyone can add new transactions
Nakamoto’s Blockchain [Nak’08]

Prevents Sybil attacks with Proofs-of-Work Puzzles [DN’92]

2 amazing aspects:

- Overcomes permissionless barrier [BCLPR’05]
- Overcomes $\frac{1}{3}$ barrier even in permissioned setting [LSP’83]
● **WHAT IS** a blockchain?
  ○ no definition of an “abstract blockchain”

● Does Nakamoto’s protocol achieve **CONSISTENCY**?
  ○ “Specific attacks” don’t work [N’08, GKL’15, SZ’15]
  ○ 49.1% attack (with 10s network delays) claimed [DW’14]
What is a blockchain?
How to build a “blockchain”
How to build a “blockchain”
How to build a “blockchain”
Search for a puzzle solution

D > H \((\text{-}, \text{coins}, \text{puzzle})\)
We found a new block
Best way to find a solution is brute-force search: model H as RO
Honest nodes only “believe” longest chain
Jesper wants to erase this transaction
For Jesper to erase his transaction, he has to find a longer chain.
"If transaction is sufficiently deep, he cannot do this unless he has majority hashpower"
“If transaction is sufficiently deep, he cannot do this unless he has majority hashpower”

- [Nak’08]: “simply trying to mine alternative chain fails”
- [GKL’15]: in synchronous network
- [SZ’15]: “non-withholding attacks” fail also with $\Delta$-delays
Blockchain abstraction (a la GKL,KL) w/ prob $\exp(-k)$

1. **Consistency**: Honest nodes agree on all but last $k$ blocks

\[ \leq k \text{ unstable} \]
Blockchain abstraction

Future-self consistency

Consistency: Honest nodes agree on all but last $k$ blocks

$\leq k$ unstable

$\exp(-k)$
Blockchain abstraction

1. **Consistency**: Honest nodes agree on all but last $k$ blocks

2. **Chain quality**: Any consecutive $k$ blocks contain “sufficiently many” honest blocks

$\text{w/ prob } \exp(-k)$
Blockchain abstraction

1. **Consistency**: Honest nodes agree on all but last $k$ blocks with prob $\exp(-k)$

2. **Chain quality**: Any consecutive $k$ blocks contain “sufficiently many” honest blocks

3. **Chain growth**: Chain grows at a steady rate
Blockchain implies “state machine replication” in the permissionless model.

1. Consistency
2. Chain quality
3. Chain growth

Traditional “state machine replication”

1. Consistency
2. Liveness
Theorem:

For every $\rho < 1/2$, if “mining difficulty” is appropriately set (as a function of the network delay $\Delta$, and total mining power), Nakamoto’s blockchain guarantees:

- Consistency
- Chain quality: $1 - \rho/(1-\rho)$
- Chain growth: $O(1/\Delta)$

where $\rho$ adv’s fraction of hashpower, and adv controls the network.
Theorem:

For every $\rho < 1/3$, if “mining difficulty” is appropriately set (as a function of the network delay $\Delta$, and total mining power), Nakamoto’s blockchain guarantees:

- Consistency
- Chain quality: $1 - (1/3)/(2/3) = 1/2$
- Chain growth: $O(1/\Delta)$

where $\rho$ adv’s fraction of hashpower, and adv controls the network.
Theorem:
For every $\rho < 1/2$, if “mining difficulty” is appropriately set (as a function of the network delay $\Delta$, and total mining power), Nakamoto’s blockchain guarantees:

- Consistency
- Chain quality: $1 - \rho/(1-\rho)$
- Chain growth: $O(1/\Delta)$

where $\rho$ adv’s fraction of hashpower, and adv controls the network
Theorem:

For every $\rho < 1/2$, if “mining difficulty” is appropriately set (as a function of the network delay $\Delta$, and total mining power), Nakamoto’s blockchain guarantees:

- Consistency
- Chain quality: $1 - \rho/(1-\rho)$
- Chain growth: $O(1/\Delta)$

“Blocks are found SLOWER than $\Delta$”

where $\rho$ adv’s fraction of hashpower, and adv controls the network
Theorem:

For every $\rho < 1/2$, if “mining difficulty” is appropriately set (as a function of the network delay $\Delta$, and total mining power), Nakamoto’s blockchain guarantees:

- Consistency
- Chain quality: $1 - \rho/(1-\rho)$
- Chain growth: $O(1/\Delta)$

“Blocktime” $>> \Delta$

where $\rho$ adv’s fraction of hashpower, and adv controls the network
When \( c = 60 \) (10 min blocktime, 10s network delays)

**Secure**: \( \rho < 49.57 \) (contradicts [DW’14]’attack!)

**Attack**: \( \rho > 49.79 \)
“Appropriately set”

\[ \alpha(1 - 2(\Delta + 1)\alpha) > \beta. \]
Theorem [Security of Nakamoto]
For every $\rho < 1/2$, if mining difficulty is appropriately set (as a function of the network delay, and total mining power), Nakamoto’s blockchain guarantees a) consistency, b) chain quality $1 - \rho/(1-\rho)$, and c) Chain growth: $O(1/\Delta)$

Theorem [Blatant attack]:
For every $\rho > 0$, for every mining difficulty, there exists a network delay such that Nakamoto’s blockchain is inconsistent and has 0 chain quality
Nakamoto’s protocol achieves strong robustness properties:

• assuming “honest majority of computational power”
• assuming puzzle difficulty is appropriately set as a function of network delay \( \Delta \)
Nakamoto’s protocol achieves strong robustness properties:
• assuming “honest majority of computational power”
• assuming puzzle difficulty is appropriately set as a function of network delay $\Delta$

**BUT 1:** Blocktime need to be roughly $10 \times \Delta$ to handle $\rho > 0.45$; thus, slow confirmation times
Nakamoto’s protocol achieves strong robustness properties:

- assuming “honest majority of computational power”
- assuming puzzle difficulty is appropriately set as a function of network delay $\Delta$

**BUT 1:** Blocktime need to be roughly $10 \times \Delta$ to handle $p > 0.45$; thus, slow confirmation times

**BUT 2:** not fair, not incentive compatible!
Follow-up Works

Incentive Compatibility: The Fruit Chain [PS’17]

All use our abstraction of a blockchain, as well as our analysis of Naka
Follow-up Works

Incentive Compatibility: The Fruit Chain [PS’17]

Fast confirmation:

All use our abstraction of a blockchain, as well as our analysis of Naka
Follow-up Works

Incentive Compatibility: **The Fruit Chain** [PS’17]

Fast confirmation:

- Assuming 2/3 honesty: **Hybrid Consensus** [PS’16]

*All use our abstraction of a blockchain, as well as our analysis of Naka*
Follow-up Works

Incentive Compatibility: The Fruit Chain [PS’17]

Fast confirmation:

- Assuming 2/3 honesty: Hybrid Consensus [PS’16]
- Impossible if only 2/3-\eps honest

All use our abstraction of a blockchain, as well as our analysis of Naka
Follow-up Works

Incentive Compatibility: **The Fruit Chain** [PS’17]

Fast confirmation:

- Assuming 2/3 honesty: **Hybrid Consensus** [PS’16]
- Impossible if only 2/3-\eps honest
- Optimistically Instant Confirmation: **Thunderella** [PS’17]

*All use our abstraction of a blockchain, as well as our analysis of Naka*
Follow-up Works

Incentive Compatibility: The Fruit Chain [PS’17]

Fast confirmation:

- Assuming 2/3 honesty: Hybrid Consensus [PS’16]
- Impossible if only 2/3-\eps honest
- Optimistically Instant Confirmation: Thunderella [PS’17]

All use our abstraction of a blockchain, as well as our analysis of Naka