

Blockchain Consensus Protocol with Horizontal Scalability

K. Cong

Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

August 26, 2017

Outline

① Introduction

- The dangers of centralisation

- Related work

- Research question

② System architecture

- System model

- Architecture overview

- Extended TrustChain

- Consensus protocol

- Transaction protocol

- Validation protocol

③ Experimental results

④ Conclusion

Outline

1 Introduction

The dangers of centralisation

Related work

Research question

2 System architecture

System model

Architecture overview

Extended TrustChain

Consensus protocol

Transaction protocol

Validation protocol

3 Experimental results

4 Conclusion

The dangers of centralisation

- Technological advancements give us convenience
- But it puts central authorities in control
- Many are motivated by profit
- Not always in the interest of the “users”¹

¹Typically users of some free service X are, in fact, used by X.

The dangers of centralisation: Examples

- Baidu's promoted search result on experimental medical care caused death of a student [1]
- Facebook can predict your opinions and desires better than your spouse [2]
- With intimate knowledge of the individuals, Facebook creates "psychographic" profiles in political campaigns [3]

Blockchain: a new hope?

- Blockchains are distributed ledgers
- They enable large scale consensus
- An alternative to central authorities for the first time
- Some applications include:
 - Digital cash (e.g., Bitcoin)
 - Domain name system (e.g., Namecoin)
 - File sharing (e.g., Filecoin)
 - General purpose (e.g., Ethereum)

Blockchain: not there yet

- Early blockchain systems do not scale
- Bitcoin is limited to 7 transactions per second
- 100,000 transaction backlog in May 2017
- We require horizontal scalability for ubiquitous use
- More nodes = better performance and throughput

Related work

Table: Summary of the scalability properties of many blockchain systems. Scalability gets better from left to right.

Not scalable	Somewhat scalable	Limited horizontal scalability	True horizontal scalability
Bitcoin Ethereum etc.	Hyperledger ByzCoin Solidius	Elastico OmniLedger Lightning Network	CHECO (this work)

State-of-the-art: Sharding

- Split state into multiple shards
- Shards run consensus algorithm in parallel
- Difficult to perform atomic inter-shard transactions
 - Elastico: not possible
 - OmniLedger: via Atomic Commit protocol
- Additional complexity
- Diminishing return

Research question

How can we design a *blockchain consensus protocol* that is *fault tolerant*, *scalable* and able to reach *global consensus*?

Outline

① Introduction

The dangers of centralisation

Related work

Research question

② System architecture

System model

Architecture overview

Extended TrustChain

Consensus protocol

Transaction protocol

Validation protocol

③ Experimental results

④ Conclusion

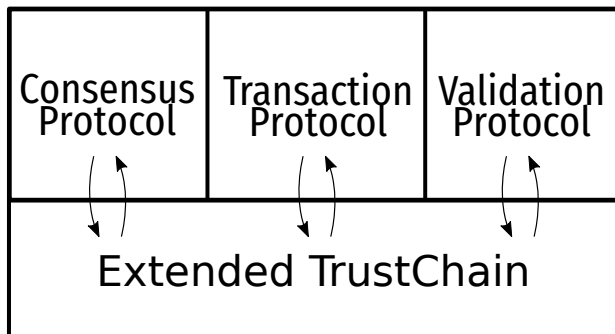
Intuition

- Traditionally: to reach consensus and check the validity of all transactions is expensive
- Our idea: we decouple consensus and validation
- Use a single digest to represent an arbitrarily large number of transactions
- Reach consensus on the small digest
- Nodes then independently check the validity of the transactions of interest

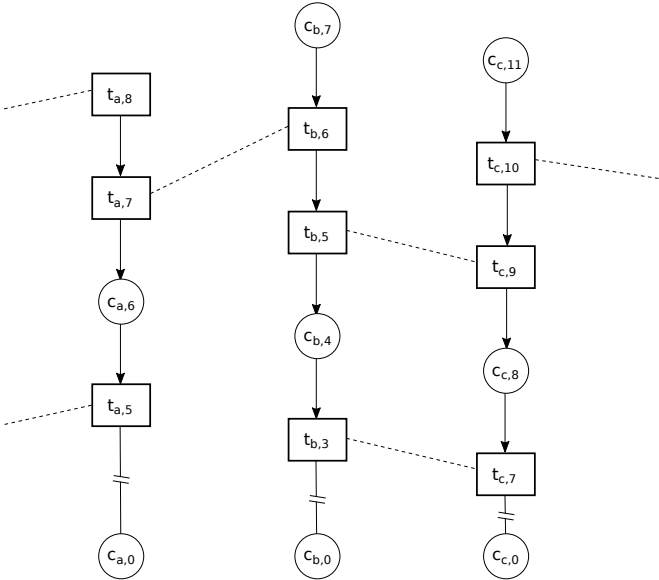
System model

- N is the population size
- Purely asynchronous channels with eventual delivery
- n nodes are facilitators
- t nodes are malicious, i.e. Byzantine
- $n \geq 3t + 1$
- $N \geq n + t$

The four components of CHECO



Extended TrustChain



Extended TrustChain: TX block

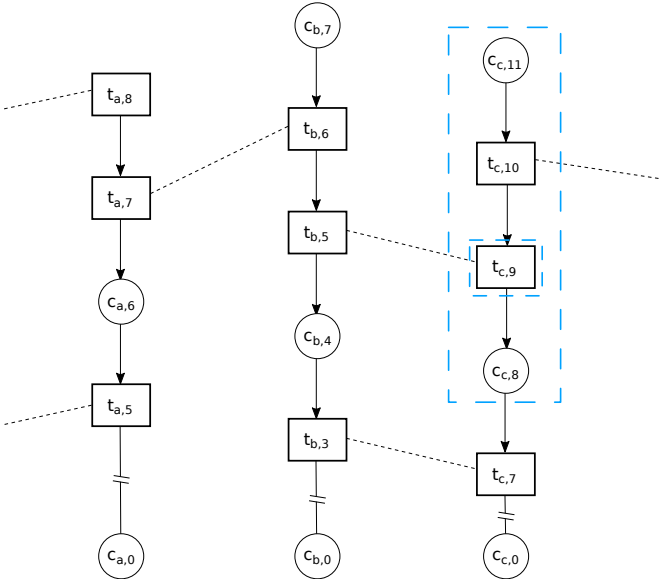
- ① Hash pointer to the previous block
- ② Sequence number
- ③ Transaction ID
- ④ Public key of the counterparty
- ⑤ Transaction message m
- ⑥ Signature the five items above

A transaction is represented by a *pair* of TX blocks

Extended TrustChain: CP block

- ① Hash pointer to the previous block
- ② Sequence number
- ③ Digest of consensus result, i.e. a set of CP blocks
- ④ Round number r
- ⑤ Signature on the four items above

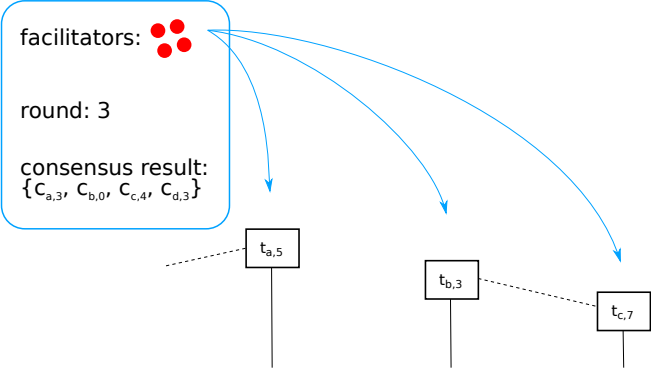
Extended TrustChain: Fragment of a TX block



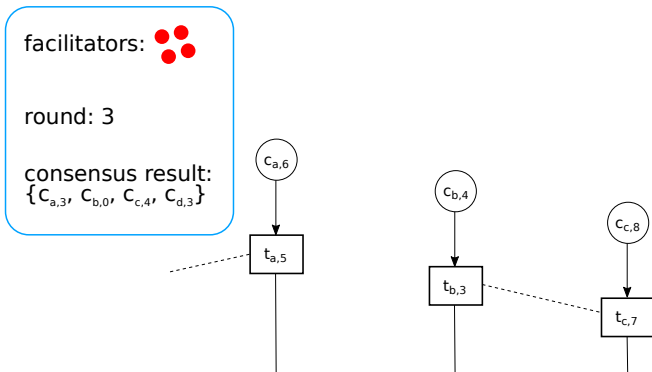
Consensus protocol—Background on ACS

- Asynchronous common subset
- A simplification of HoneyBadgerBFT [4]
- n nodes
- t nodes may be malicious
- Input: every node proposes a set of values, e.g., $\{A, B\}, \{B, C\}, \dots$
- Output: set union of the majority, e.g., $\{A, B, C, \dots\}$

Consensus protocol



Consensus protocol

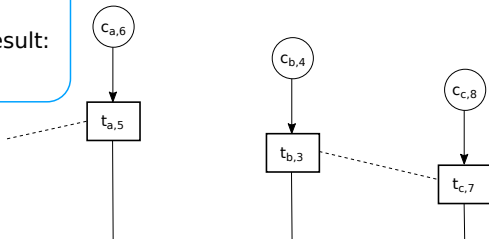


Consensus protocol

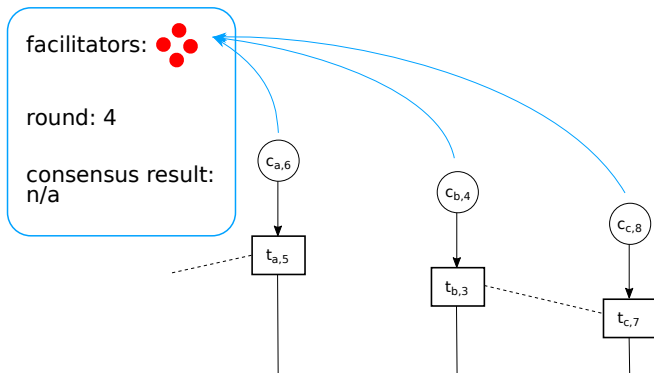
facilitators: lottery
 $\{c_{a,3}, c_{b,0}, c_{c,4}, c_{d,3}\}$

round: 4

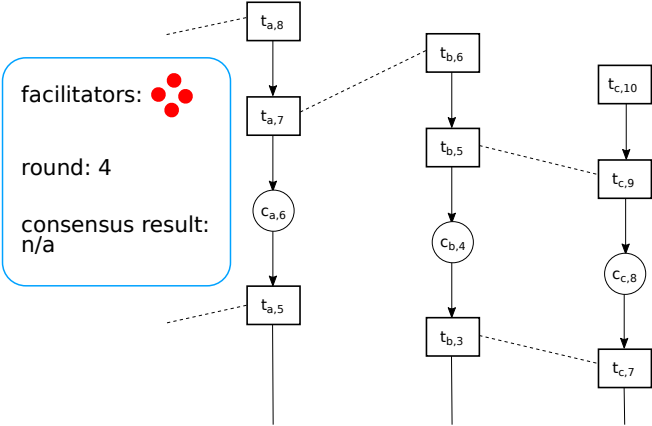
consensus result:
n/a



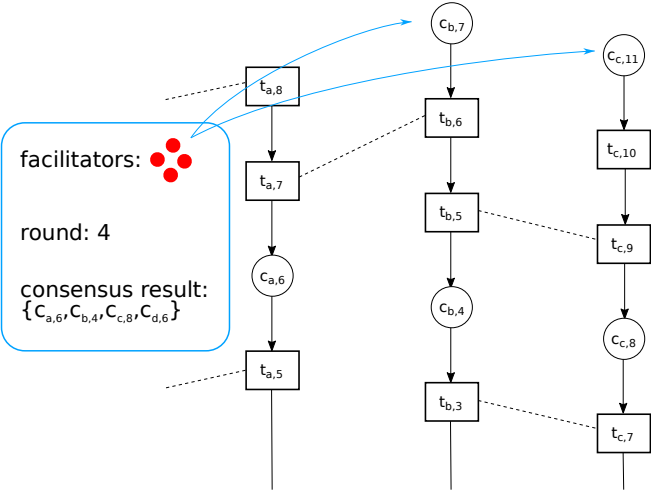
Consensus protocol



Consensus protocol



Consensus protocol

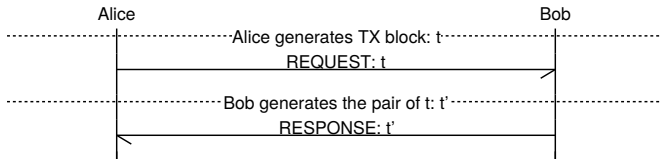


Consensus protocol: properties

The consensus protocol has the following properties in every round r .

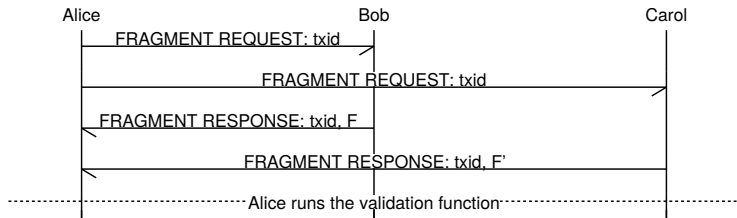
- ① *Agreement*: Every correct outputs the same set of facilitators.
- ② *Validity*: The consensus results is valid such that a new set of facilitators can be computed from it.
- ③ *Fairness*: Every node with a CP block in the consensus result should have an equal probability of becoming a facilitator.
- ④ *Termination*: Every correct node eventually outputs a set of facilitators.

Transaction protocol



- Request and response protocol
- Two TX blocks containing the same *txid* are generated
- Non-blocking

Validation protocol



- Alice needs the two fragments that belong to the transaction
- Validation function checks whether the fragments are OK and contain the transaction
- Non-blocking

Validation protocol: properties

- CP blocks of the fragments are “anchored” due to the consensus protocol
- It is difficult to modify the fragment once “anchored”
- Implicit consensus on transactions via CP blocks

Outline

① Introduction

The dangers of centralisation

Related work

Research question

② System architecture

System model

Architecture overview

Extended TrustChain

Consensus protocol

Transaction protocol

Validation protocol

③ Experimental results

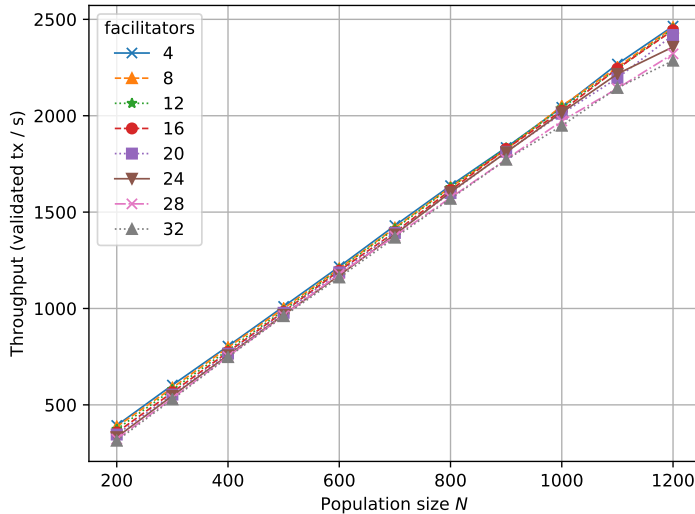
④ Conclusion

Implementation and experiment setup

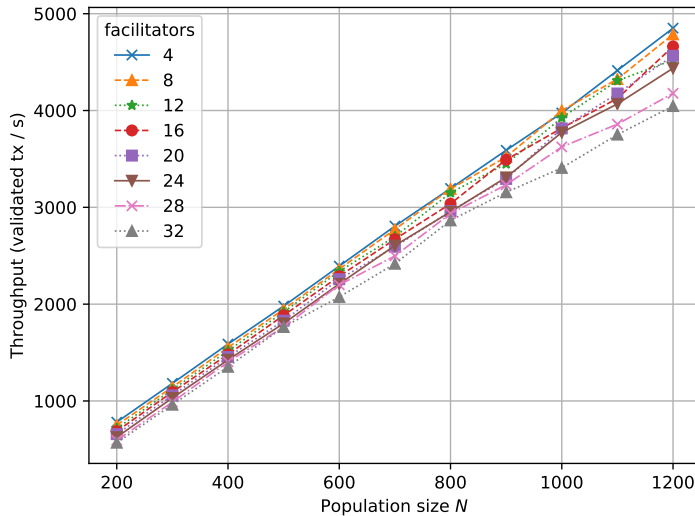
- Free and open source implementation on Github:
<https://github.com/kc1212/checo>
- SHA256 for hash functions and Ed25519 for digital signature
- Experiment on the DAS-5²

²<http://www.cs.vu.nl/das5/>

Throughput vs population size (random neighbour)



Throughput vs population size (fixed neighbour)



Outline

① Introduction

The dangers of centralisation

Related work

Research question

② System architecture

System model

Architecture overview

Extended TrustChain

Consensus protocol

Transaction protocol

Validation protocol

③ Experimental results

④ Conclusion

Conclusion

Research question

How can we design a *blockchain consensus protocol* that is *fault tolerant*, *scalable* and able to reach *global consensus*?

Our system achieve the following

- Fault tolerant up to t nodes
- Horizontal scalability
- Global consensus on CP blocks

Future work

- Improve fault tolerance
- Improve fork detection
- Analyse the system in the permissionless environment
- Concrete application

Bibliography



[Online]. Available:
https://en.wikipedia.org/wiki/Death_of_Wei_Zexi.



W. Youyou, M. Kosinski, and D. Stillwell, “Computer-based personality judgments are more accurate than those made by humans”, *Proceedings of the National Academy of Sciences*, vol. 112, no. 4, pp. 1036–1040, 2015.



[Online]. Available: <https://www.theguardian.com/technology/2017/jul/31/facebook-dark-ads-can-swing-opinions-politics-research-shows>.



A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, “The honey badger of bft protocols”, in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2016, pp. 31–42.