BEYOND STAKE

BE
ST

Implementing Diversity Policies on PoS.

Klaus Kursawe
vega.xyz

Profiles

Fons Bruekers and Stefan Katzenbeisser and Klaus Kursawe and Pim Tuyls

2002/134 (PS PS.GZ PDF)

Asynchronous Verifiable Secret Sharing and Proactive Cryptosystems

Christian Cachin and Klaus Kursawe and Anna Lysyanskaya and Reto Strobl

(PS PS.GZ PDF)

...ic Asynchronous Atomic

...d Victor Shoup

...chronous

Vega

Derivative trading platform running on a dedicated and specialised chain.

Improving on the chain (MEV/fairness, latency, diversity)

Hello!

By History:

- PhD on Byzantine Fault Tolerant Ordering protocols in 2001 at IBM Zurich
  - First fully asynchronous, leaderless, practical BFT protocol (with implementation & formal verification)

As noone cared about this back then:

- Other stuff in Security and Privacy
- Security of Critical Infrastructures

Comeback:

- Former advisor to ChainSpace.io, Libra
- VEGA

WHY VALIDATOR POLICIES ?

Bitcoin Original & Cypherpunk vision:

The chain is secured by thousands of students in their dorm rooms.

Modern Reality:

Mining/Validating is a serious business

This undermines some of the basic assumptions we've got

THE SEARCH FOR POLICY

An Axiomatic Approach to Block Rewards

Xi Chen, Columbia University, xichen@cs.columbia.edu
Christos Papadimitriou, Columbia University, christos@cs.columbia.edu
Tim Roughgarden, Columbia University, tr@cs.columbia.edu

September 25, 2019

arXiv:1909.10645v1 [cs.GT] 23 Sep 2019

Abstract

Proof-of-work blockchains reward each miner for one completed block by an amount that is, in expectation, proportional to the number of hashes the miner contributed to the mining of the block. Is this *proportional allocation rule* optimal? And in what sense? And what other rules are possible? In particular, what are the desirable properties that any "good" allocation rule should satisfy? To answer these questions, we embark on an axiomatic theory of incentives in proof-of-work blockchains at the time scale of a single block. We consider desirable properties of allocation rules including: symmetry; budget balance (weak or strong); sybil-proofness; and various grades of collusion-proofness. We show that Bitcoin's proportional allocation rule is the unique allocation rule satisfying a certain system of properties, but this does not hold for slightly weaker sets of properties, or when the miners are not risk-neutral. We also point out that a rich class of allocation rules can be approximately implemented in a proof-of-work blockchain.

1 Introduction

The Bitcoin protocol was a remarkable feat: eleven years after its sudden appearance [7], and without much adjustment and debugging, it has been used by millions of people and has launched the blockchain industry. Arguably, the most crucial and ingenious aspect of its design lies in the *incentives* the protocol provides to its miners to participate and follow it faithfully. We believe it is of great importance and interest to understand and scrutinize the incentives provided by blockchain protocols—and to do so through the point of view and the methodology of Economic Theory, the science of incentives.

Flaws in the incentives of a blockchain protocol can manifest themselves at multiple timescales. For longest-chain proof-of-work blockchains like Bitcoin, the most well-studied incentive-based attacks, such as selfish mining [4, 10, 5] and transaction sniping [3], concern miners reasoning strategically over multiple block creation epochs. For example, in selfish mining, a miner relinquishes revenue in the short term to achieve greater revenue (in expectation) in the long run via a type of forking attack.

This paper studies incentive issues and potential deviations from intended miner behavior at the most basic time scale, that of a single block creation epoch. We focus on the allocation of *block rewards*, which drives the incentive structure in Bitcoin and many other similar protocols. The dominant paradigm in proof-of-work blockchains is to fix a per-block reward, and for each block to allocate the entire reward to whichever miner first solves a difficult cryptopuzzle. Assuming that miners independently and randomly guess and check possible solutions to the cryptopuzzle,

Beyond Staking

An Aphoristic design for Staking and Rewards

Klaus Kursawe*

Vega Protocol

king Aphorisms

In a decentralised system, it is vital to align the mechanisms that steer the consensus protocol – both through economy and through protocol design – to assure that validators and miners are likely to behave in the best interest of the overal system. What this means in detail, however, and how ideal properties can be married with implementable policies, is still an open question. The first attempt towards a structured approach we are aware of has been done by Chen et al. [1], though their approach is more aimed at incentive structures for proof-of-work protocols. They propose five axioms as the base of a reward system, namely Non-negativity, budget balance, Symmetry, Sybil-proofness, and Collusion proofness. While these are logical choices for desirable properties, there are also as logical arguments for directly contradicting axioms; if we start from a different angle to prioritize decentralisation and diversity, as well as the

There's logical policies we'd want that contradict each other

- Sybill freeness: A validator shouldn't profit from splitting into several pseudo-entities
- Anti-Whaling: No single validator should have more than x% of the vote

THE SEARCH FOR POLICY

Focus for this talk: Diversity in Validator properties

An Axiomatic Approach to Block Rewards

Xi Chen, Columbia University, xichen@cs.columbia.edu
Christos Papadimitriou, Columbia University, christos@cs.columbia.edu
Tim Roughgarden, Columbia University, tr@cs.columbia.edu

September 25, 2019

arXiv:1909.10645v1 [cs.GT] 23 Sep 2019

Abstract

Proof-of-work blockchains reward each miner for one completed block by an amount that is, in expectation, proportional to the number of hashes the miner contributed to the mining of the block. Is this *proportional allocation rule* optimal? And in what sense? And what other rules are possible? In particular, what are the desirable properties that any "good" allocation rule should satisfy? To answer these questions, we embark on an axiomatic theory of incentives in proof-of-work blockchains at the time scale of a single block. We consider desirable properties of allocation rules including: symmetry; budget balance (weak or strong); sybil-proofness; and various grades of collusion-proofness. We show that Bitcoin's proportional allocation rule is the unique allocation rule satisfying a certain system of properties, but this does not hold for slightly weaker sets of properties, or when the miners are not risk-neutral. We also point out that a rich class of allocation rules can be approximately implemented in a proof-of-work blockchain.

1 Introduction

The Bitcoin protocol was a remarkable feat: eleven years after its sudden appearance [7], and without much adjustment and debugging, it has been used by millions of people and has launched the blockchain industry. …

Beyond Staking

An Aphoristic design for Staking and Rewards

Vega Protocol

ing Aphorisms

In a decentralised system, it is vital to align the mechanisms that steer the consensus protocol – both through economy and through protocol design – to assure that validators and miners are likely to behave in the best interest of the overal system. What this means in detail, however, and how ideal properties can be married with implementable policies, is still an open question. The first attempt towards a structured approach we are aware of has been done by Chen et al. [1], though their approach is more aimed at incentive structures for proof-of-work protocols. They propose five axioms as the base of a reward system, namely Non-negativity, budget balance, Symmetry, Sybil-proofness, and Collusion proofness. While these are logical choices for desirable properties, there are also as logical arguments for directly contradicting axioms; if we start from a different angle to prioritize decentralisation and diversity, as well as the

There's logical policies we'd want that contradict each other

- Sybill freeness: A validator shouldn't profit from splitting into several pseudo-entities
- Anti-Whaling: No single validator should have more than x% of the vote

The need for validator-diversity

- “When I introduced Byzantine failures, it was meant to model arbitrary but independent failures, not coordinated malicious ones. The assumption that a dedicated attacker is bound by attacking only one third of all parties is ridiculous.”

  Leslie Lamport, 2001

- “China hosts around 75% of the world's bitcoin mining capacity—or “hashrate”—due to its established technology supply chains and extremely cheap electricity.”

  Time, June 2, 2021

- “The basic answer is that 37.07% of stake is in AWS. That is quite frankly not good. But they are almost all "private validators" - run by institutions that don't care much about the health of Solana as long as they can make some money.”

  Reddit.com

- “The ETH 2.0 testnet ‘Medalla’ came to a grinding halt due to a time-bug that took a majority of testnet validators offline. This is the first instance of the network coming to a stop. Although Ethereum has experienced bottlenecks in the past, it has never come to a full stop like it did due to the testnet time-bug. […] As a result, the percentage of individuals successfully validating blocks on the ETH 2.0 testnet dropped from 75% to 5%.”

  Coingeek.com

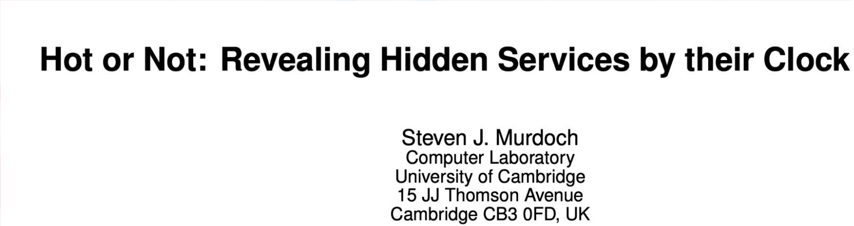
- **In a bold and potentially unprecedented move buried in the lawsuit's 69th paragraph, the SEC today claimed it had the right to sue Balina not only because his case concerns transactions made in the United States, but also because, essentially, the entire Ethereum network falls under the US government's purview.**

  Decrypt.co

CONTROVERSY: SEMI-ENFORCABILITY

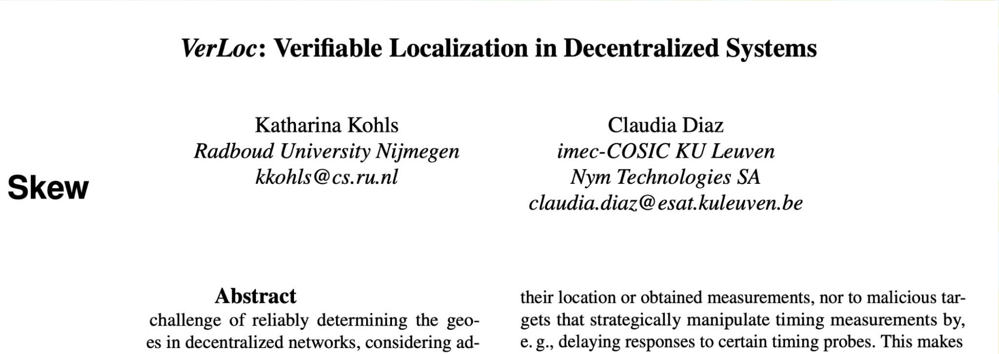
There may not be a reliable way to reliably measure a property (is a validator is situated where they claim they are, what operating system do they run/…).

- We have a security policy that we may not be able to enforce to 100%
- Not having anything is a worse idea (See Bitcoin & China)
- We probably don't need to be completely secure, it is sufficient if it's more effort/risk to cheat than not to, or at least that breaking the policy requires criminal intent
- We already have nice work to make cheating expensive/hard/dangerous (at least in the PoS world)



Hot or Not: Revealing Hidden Services by their Clock Skew

Steven J. Murdoch
Computer Laboratory
University of Cambridge
15 JJ Thomson Avenue
Cambridge CB3 0FD, UK



***VerLoc*: Verifiable Localization in Decentralized Systems**

Katharina Kohls
Radboud University Nijmegen
kkohls@cs.ru.nl

Claudia Diaz
imec-COSIC KU Leuven
Nym Technologies SA
claudia.diaz@esat.kuleuven.be

Abstract

challenge of reliably determining the geo-
es in decentralized networks, considering ad-

their location or obtained measurements, nor to malicious tar-
gets that strategically manipulate timing measurements by,
e. g., delaying responses to certain timing probes. This makes

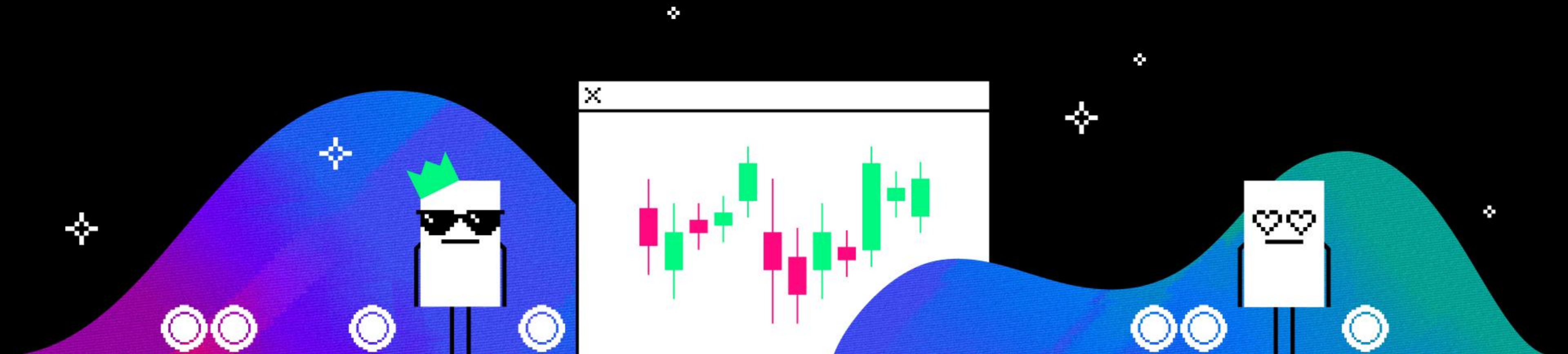
ECONOMIC POLICY IMPLEMENTATION

- Negative incentive:
  - Slashing/reward withholding for misbehaving validators
- Positive Incentive
  - Diversity Rewards: Give extra rewards to validators that add to system diversity
- Indirect Economics
  - Delegated Proof of Stake: Loss of revenue/reputation results in loss of delegation and thus, weight
  - This might also be implemented through secondary markets

LIMITS OF ECONOMIC IMPLEMENTATIONS

- Contradictory Policies (e.g., geographic diversity vs. performance)
- Different Validator Businessmodels
  - MEV
  - Cross-Domain-MEV; other aspects of multi-chain validators
  - VC/Custodian-Relations
  - …
- New Financial Instruments
  - Outsource risks of slashing, e.g. through derivatives, selling deposits, …
  - Flashloans
- Higher motivation to cheat
  - There’s now value in lying about properties
  - If you measure something, someone will find a way to game the scoring system

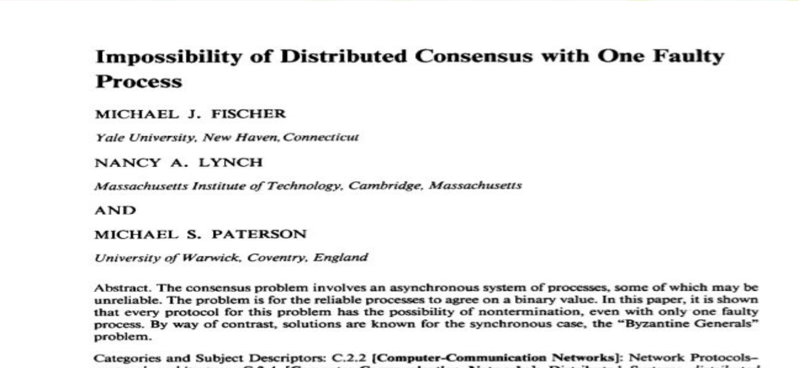
DIVERSITY IMPLEMENTATION ON CONSENSUS LEVEL



CONTEXT: WHY CONSENSUS IS MESSY

Consensus is (sorta) impossible: More precisely:

“No deterministic asynchronous protocol can guarantee termination even in the presence of one crash failure”

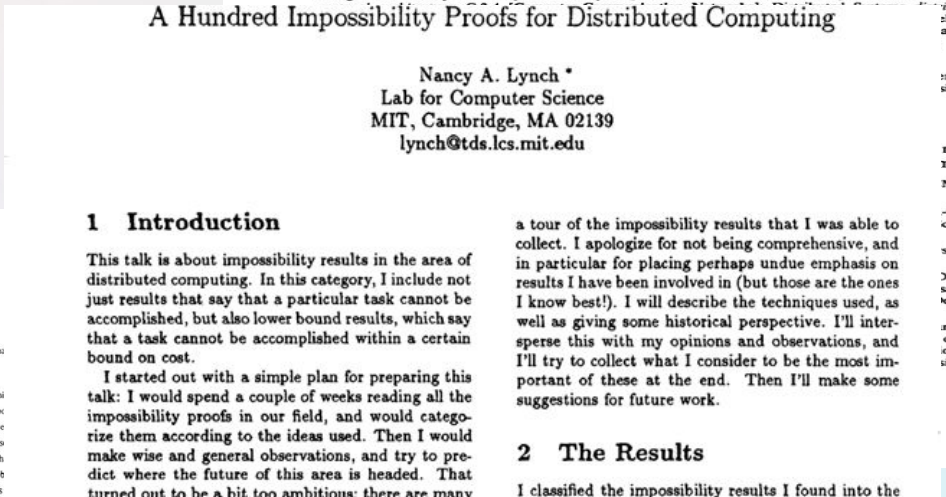


Impossibility of Distributed Consensus with One Faulty Process

MICHAEL J. FISCHER
Yale University, New Haven, Connecticut
NANCY A. LYNCH
Massachusetts Institute of Technology, Cambridge, Massachusetts
AND
MICHAEL S. PATERSON
University of Warwick, Coventry, England

Abstract. The consensus problem involves an asynchronous system of processes, some of which may be unreliable. The problem is for the reliable processes to agree on a binary value. In this paper, it is shown that every protocol for this problem has the possibility of nontermination, even with only one faulty process. By way of contrast, solutions are known for the synchronous case, the “Byzantine Generals” problem.

Categories and Subject Descriptors: C.2.2 **[Computer-Communication Networks]**: Network Protocols–



A Hundred Impossibility Proofs for Distributed Computing

Nancy A. Lynch *
Lab for Computer Science
MIT, Cambridge, MA 02139
lynch@tds.lcs.mit.edu

1 Introduction

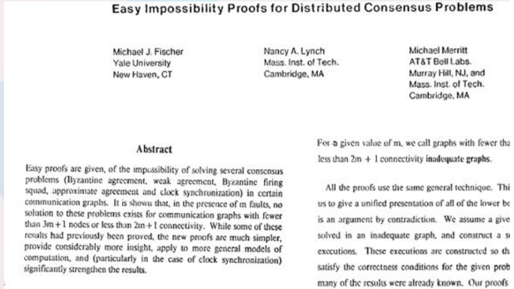
This talk is about impossibility results in the area of distributed computing. In this category, I include not just results that say that a particular task cannot be accomplished, but also lower bound results, which say that a task cannot be accomplished within a certain bound on cost.

I started out with a simple plan for preparing this talk: I would spend a couple of weeks reading all the impossibility proofs in our field, and would categorize them according to the ideas used. Then I would make wise and general observations, and try to predict where the future of this area is headed. That turned out to be a bit too ambitious; there are many

a tour of the impossibility results that I was able to collect. I apologize for not being comprehensive, and in particular for placing perhaps undue emphasis on results I have been involved in (but those are the ones I know best!). I will describe the techniques used, as well as giving some historical perspective. I'll intersperse this with my opinions and observations, and I'll try to collect what I consider to be the most important of these at the end. Then I'll make some suggestions for future work.

2 The Results

I classified the impossibility results I found into the



Easy Impossibility Proofs for Distributed Consensus Problems

Michael J. Fischer
Yale University
New Haven, CT

Nancy A. Lynch
Mass. Inst. of Tech.
Cambridge, MA

Michael Merritt
AT&T Bell Labs.
Murray Hill, NJ, and
Mass. Inst. of Tech.
Cambridge, MA

Abstract

Easy proofs are given, of the impossibility of solving several consensus problems (Byzantine agreement, weak agreement, Byzantine firing squad, approximate agreement and clock synchronization) in certain communication graphs. It is shown that, in the presence of m faults, no solution to these problems exists for communication graphs with fewer than 3m+1 nodes or less than 2m+1 connectivity. While some of these results had previously been proved, the new proofs are much simpler, provide considerably more insight, apply to more general models of computation, and (particularly in the case of clock synchronization) significantly strengthen the results.

For a given value of m, we call graphs with fewer tha less than 2m + 1 connectivity inadequate graphs.

All the proofs use the same general technique. Thi us to give a unified presentation of all of the lower bo is an argument by contradiction. We assume a give solved in an inadequate graph, and construct a s executions. These executions are constructed so th satisfy the correctness conditions for the given prob many of the results were already known. Our proofs

CONTEXT: WHY CONSENSUS IS MESSY

Consensus is (sorta) impossible: More precisely:

“No deterministic asynchronous protocol can guarantee termination even in the presence of one crash failure”

- Use time. This is the most efficient way to get around this, unless your timing assumption was wrong.
- Use probability. Terminating with probability 1 is good enough. Slightly slower, but fully asynchronous
- Don’t terminate/finalize. We can live with some probability of rollbacks.

Impossibility of Distributed Consensus with One Faulty Process

MICHAEL J. FISCHER
Yale University, New Haven, Connecticut
NANCY A. LYNCH
Massachusetts Institute of Technology, Cambridge, Massachusetts
AND
MICHAEL S. PATERSON
University of Warwick, Coventry, England

Abstract. The consensus problem involves an asynchronous system of processes, some of which may be unreliable. The problem is for the reliable processes to agree on a binary value. In this paper, it is shown that every protocol for this problem has the possibility of nontermination, even with only one faulty process. By way of contrast, solutions are known for the synchronous case, the “Byzantine Generals” problem.

Categories and Subject Descriptors: C.2.2 **[Computer-Communication Networks]**: Network Protocols–

A Hundred Impossibility Proofs for Distributed Computing

Nancy A. Lynch *
Lab for Computer Science
MIT, Cambridge, MA 02139
lynch@tds.lcs.mit.edu

1 Introduction

This talk is about impossibility results in the area of distributed computing. In this category, I include not just results that say that a particular task cannot be accomplished, but also lower bound results, which say that a task cannot be accomplished within a certain bound on cost.

I started out with a simple plan for preparing this talk: I would spend a couple of weeks reading all the impossibility proofs in our field, and would categorize them according to the ideas used. Then I would make wise and general observations, and try to predict where the future of this area is headed. That turned out to be a bit too ambitious; there are many

a tour of the impossibility results that I was able to collect. I apologize for not being comprehensive, and in particular for placing perhaps undue emphasis on results I have been involved in (but those are the ones I know best!). I will describe the techniques used, as well as giving some historical perspective. I'll intersperse this with my opinions and observations, and I'll try to collect what I consider to be the most important of these at the end. Then I'll make some suggestions for future work.

2 The Results

I classified the impossibility results I found into the

Easy Impossibility Proofs for Distributed Consensus Problems

Michael J. Fischer
Yale University
New Haven, CT

Nancy A. Lynch
Mass. Inst. of Tech.
Cambridge, MA

Michael Merritt
AT&T Bell Labs.
Murray Hill, NJ, and
Mass. Inst. of Tech.
Cambridge, MA

Abstract

Easy proofs are given, of the impossibility of solving several consensus problems (Byzantine agreement, weak agreement, Byzantine firing squad, approximate agreement and clock synchronization) in certain communication graphs. It is shown that, in the presence of m faults, no solution to these problems exists for communication graphs with fewer than 3m+1 nodes or less than 2m+1 connectivity. While some of these results had previously been proved, the new proofs are much simpler, provide considerably more insight, apply to more general models of computation, and (particularly in the case of clock synchronization) significantly strengthen the results.

For a given value of m, we call graphs with fewer tha less than 2m + 1 connectivity inadequate graphs.

All the proofs use the same general technique. Thi us to give a unified presentation of all of the lower bo is an argument by contradiction. We assume a give solved in an inadequate graph, and construct a s executions. These executions are constructed so th satisfy the correctness conditions for the given prob many of the results were already known. Our proofs

THE CONSENSUS MAP

	Randomized	pBFT/partial synchronous	Longest Chain
	Committee Based		
PoS	CKPS01, Sintra, HoneyBadger,… Finalizing 2/3 honest no timing assumptions Leaderless Bypass FLP by probabilistic termination	CL99, Tendermint, Algorand, Hotstuff … Finalizing 2/3 honest timing requirements for liveness/performance Bypass FLP by timing assumption	Solana, Ouroboros,… **Gasper** Non Finalizing 51% honest timing requirements for safety Bypass FLP by non-finalization/timing assumption
PoW	*It's possible (probably ?)…*		Bitcoin, Ethereum PoW, … Non Finalizing 51% honest timing requirements for safety

THE CONSENSUS MAP

	Randomized	pBFT/partial synchronous	Longest Chain
	Committee Based		
PoS	CKPS01, Sintra, HoneyBadger,… Finalizing 2/3 honest no timing assumptions Bypass FLP by probabilistic termination	CL99, Tendermint, Algorand, Hotstuff… Finalizing 2/3 honest timing requirements for liveness/performance Bypass FLP by timing assumption	Solana, Ouroboros,… Non Finalizing 51% honest timing requirements for safety Bypass FLP by non-finalization/timing assumption
PoW	*It's possible (probably ?)…*		Bitcoin, Ethereum PoW,… Non Finalizing 51% honest timing requirements for safety

Pretty much understood

Gasper (ETH PoS)
We need to talk…

Needs experimentation & statistical evaluation

GENERAL ADVERSARY STRUCTURES

In the normal model, we can do consensus if we have less than 1/3 (51%) of stake corrupted. This is boring.

General Adversary Structures:

- Explicitly write down all sets of validators we want to tolerate to collude
  - This is the most flexible notion; we want to scale it down later to be more manageable
  - The latter is also required for registrationlessness*
- Modify our protocols to replace stake by those sets
- Re-Examine the impossibility proofs to define requirements for the sets

*We can be permissionless (i.e., noone can tell you to not validate), but still require registration (i.e., validators know of each other.)

GENERAL ADVERSARY STRUCTURES ON COMMITTEE BASED PROTOCOLS

upon $\langle \mathsf{PROPOSAL}, h_p, round_p, v, vr \rangle$ **from** $\mathsf{proposer}(h_p, round_p)$ AND $2f+1$ $\langle \mathsf{PREVOTE}, h_p, vr, id(v) \rangle$ **while** $step_p = propose \wedge (vr \geq 0 \wedge vr < round_p)$ **do**

  if $valid(v) \wedge (lockedRound_p \leq vr \vee lockedValue_p = v)$ **then**

    broadcast $\langle \mathsf{PREVOTE}, h_p, round_p, id(v) \rangle$

  else

    broadcast $\langle \mathsf{PREVOTE}, h_p, round_p, nil \rangle$

  $step_p \leftarrow prevote$

Tendermint code extraxt
f: number of tolerated failures (a.k.a. t)
(or, tolerated represented stake)

In modern protocols, there’s pretty much only three thresholds:
n-t
2t+1 (usuallty the same as n-t, as n=3t+1)
t+1

What we really want from thresholds

If we use thresholds in our protocols, what do we actually mean ?

wait for t+1 messages

Property: you can expect to have input from at least one honest validator

- wait until you heard from people from at least 5 countries
- wait until you heard from at least 3 different implementations

wait for 2t+1 messages

Property: you expect to have an honest majority / two of these set intersect in one honest party

- wait until you heard from people of at least 9 countries
- wait until you heard from at least 5 different implementations

wait for n-t messages

Property: it doesn't make sense to wait any longer

- wait until you heard of all countries active in the last 24 hours minus 4
- Wait until you got 2/3 of all people active in the last 24 hours
- Wait until you heard from people that sum up to 2/3 of the combined votes of the last 3 months

Transforming protocols & proofs

Let P be the set of all participants, and Z the set of subsets of P, such that Z contains all sets of parties that we allow to be corrupted simultaneously. Then

t+1 → a minimal set of parties that is not contained in Z

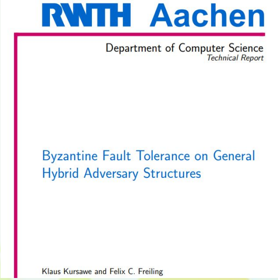
2+1 →a minimal set of parties that is not covered by the union of two sets in Z

n-t → a set of parties that is P without any set in Z

For most modern (committee based) protocols, we can simply replace the thresholds with these sets and have them run on general adversary structures. Similarly, most proofs transform straightforwardly

We can also extend the model to hybrid byzantine/crash failures (n>3b+2c) without needing to change the protocol logic; in this case, each set is a set (C,B) of parties that can crash and parties that can go bad. This allows for tolerating more failures overall.

Limits


RWTH Aachen

Department of Computer Science
Technical Report

Byzantine Fault Tolerance on General Hybrid Adversary Structures

Klaus Kursawe and Felix C. Freiling

Not all sets are possible; just like we have ⅓ and ½ in the threshold model to make consensus possible, we have limits for the set composition.

Let $\mathcal{P}$ be the set of all participants. An *adversary structure* $\mathcal{Z}$ is a monotone set of classes (C, B) of subsets of $\mathcal{P}$ (i.e., $C, B \subset \mathcal{P}$) [FHM99]. The adversary structure $\mathcal{Z}$ satisfies the predicate $Q^{(3,2)}(\mathcal{P}, \mathcal{Z})$, if $\forall (B_1, C_1), (B_2, C_2), (B_3, C_3) \in \mathcal{Z} : \{B_1 \cup B_2 \cup B_3 \cup C_1 \cup C_2\} \neq \mathcal{P}$.

Requiremenrts (necessary and sufficient)

n > 3t+1 → no union of three such sets covers all validators (requirement for asynchronous protocols). T

This is called Q(3)

n> 2t+1/51%. →no union of two such sets covers all validators (requirement for timed protocols).

This is called Q(2)

We can use that to compute the number of validators needed for a given policy. Generally: The more complex the policy, the more validators I need to be able to implement it.

*Tested or
pBFT 99
CKPS01
KS01
Wendy
Tendermin
Hotstuff

http://sunsite.informatik.rwth-aachen.de/Publications/AIB/2005/2005-09.pdf

SIMPLE LONGEST CHAIN PROTOCOLS

- Longest chain protocols don't have thresholds, but they have
- A leader selection algorithm
- A longest chain rule

The length of a block is 0.95^(maximum number of directly preceeding blocks it shares a corruption set with)

Thus, any chain that doesn't get out of some corruption set will eventually be shorter than competing chains.

This has a number of details that need consideration, e.g., the number of block confirmations

The 51% rule would be replaced by Q(2).

Needs more careful analysis on

is 0.95 a good number

how does this affect confirmation times

Since we can't use simple Bitcoin-analysis style probabilities anymore, what do we base such recommendations on ?

LONGEST CHAINS: PARAMETER CHOICE ?

- In normal longest chain, we can compute probabilities of fork-length by assuming every leader is honest with $p>0.5$
- As the whole point here is to eliminate failure independence, this doesn't work anymore.
- We can still give some indications, but they change with the sets and are somewhat harder to compute. Using a good leader choice algorithm will probably help
- Also, the number 0.95 is completely arbitrary.

GASPER

This is a committee based protocol which can use a pBFT style conversion

Algorithm 4.2 Hybrid LMD GHOST Fork Choice Rule

procedure HLMD(G)
  $L \leftarrow$ set of leaf blocks B_l in G
  $(B_J, j) \leftarrow$ the justified pair with highest attestation epoch j in $J(\mathsf{ffgview}(B_l))$ over $B_l \in L$
  $L' \leftarrow$ set of leaf blocks B_l in G such that $(B_j, j) \in J(\mathsf{ffgview}(B_l))$
  $G' \leftarrow$ the union of all chains $\mathsf{chain}(B_l)$ over $B_l \in L'$
  $B \leftarrow B_J$
  $M \leftarrow$ most recent attestations (one per validator)
  while B is not a leaf block in G' **do**
    $B \leftarrow \underset{B' \text{ child of } B}{\arg\max}\ w(G', B', M)$
    (ties are broken by hash of the block header)
  return B

This is the longest chain approach mentioned before

Definition 3.4. Given a view G, Let M be the set of latest attestations, one per validator. The *weight* $w(G, B, M)$ is defined to be the sum of the stake of the validators i whose last attestation in M is to B or descendants of B.

MANAGING THE SETS

- Manualy defining the sets is too flexible. The most natural way to generate them on the fly is property based:
- We want to tolerate failure of nodes representing 1/3 of stake (now)
- We want to tolerate failure of 1/3 of the nodes and all nodes in 1/3 of the countries
- We want to tolerate failure of all nodes in 1/3 of countries + 2 cloud providers
- We want to tolerate failure of all nodes with the same implementation, plus all nodes in 1/3 of the countries, plus 1/3 of the represented stake
- The only limit is the Q(3) predicate; the more attributes we want to cover, the more difficult that gets. We can also change the policy dynamically.
- Given this limitation, what policies are desirable for a working ecosystem ?
  - Especially, along what properties do we want to diversify (geography, cloud-provider, implementation, running MEVBoost/different proposers
- How do we handle that Ghost and Casper have different conditions ?
- How do we avoid ‘minority stacking’ ?

PARTING SUMMARY

- Plain blockchain implementations can get serious issues when the interests of validators and the network don't allign
- Economic Incentivisation is the most used tools to re-align them, but that has limits
- Consensus level policy implementations are a great tool here; general adversary structures offer great (too much) flexibility, and can be integrated relatively naturally into existing protocols
- Diversity:

  This can be implemented relatively painlessly (though we do need to make sure nothing explodes, especially with complex protocols like GASPER. We're not done here.

  There's a limit on how complex diversity policies can be if we want to be diverse along several attributes. Given we need to prioritize somewhere, this would be a great discussion to have/

  A separate question is on how to measure those attributes

  To all lawyers: Does that help arguing about decentralisation, too ?

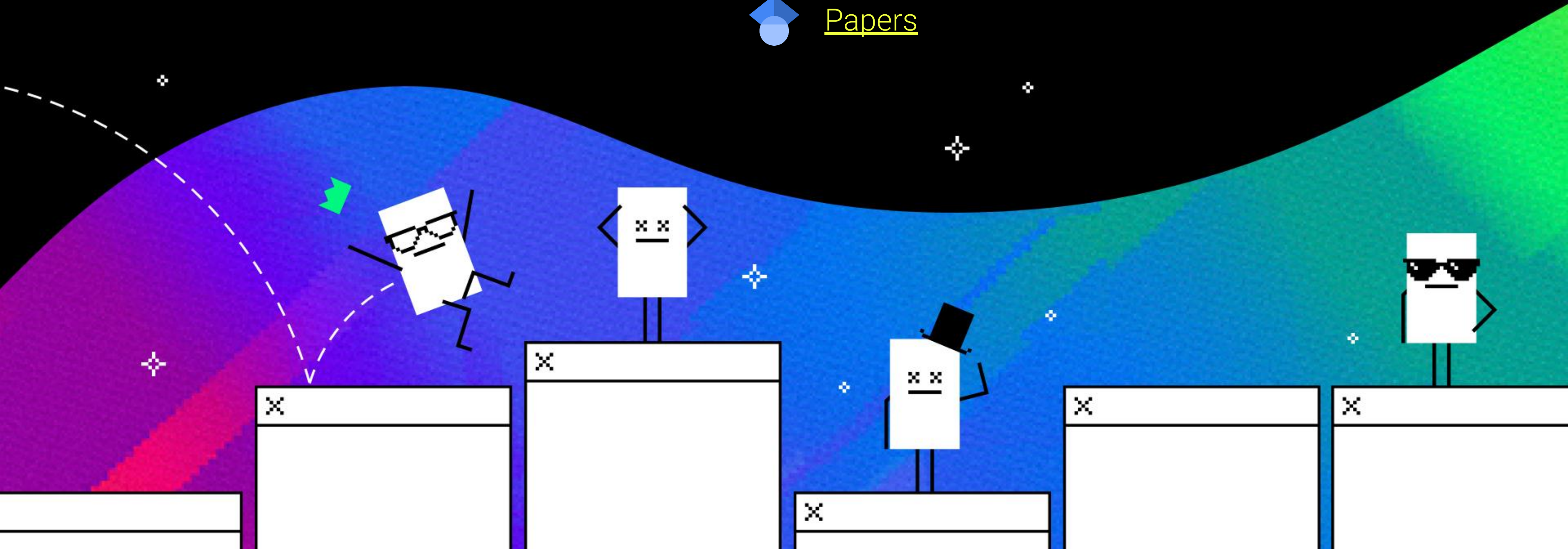
Klaus Kursawe, Klaus@vega.xyz

THANK YOU!

Klaus Kursawe
Vega Protocol

vega.xyz
klaus@vega.xyz
Twitter: @chezklaus

[Papers](#)



LONGEST CHAIN PROTOCOLS

- Thresholds are not part of the protocol, but implicit

  Finallity is an external policy, and 51% rule is needed for that to make sense

  Thus, we can’t replace protocol thresholds with adversary structures

- Policies can be implemented in chain length weight

  Currently, every block adds 1 to the chain length. This isn’t necessary

  Length can be modified to represent an adversary structure:

  A block length is counted as 0.95^x, where x is the number of blocks directly preeceeding it that have been generated by validators in the same adversary set.

- Leader Selection can also take Avs into account

  Finallity is an external policy, and 51% rule is needed for that to make sense

  Thus, we can’t replace protocol thresholds with adversary structures

Section 1 title here.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

- Sollicitudin
- Consectetur
  - Condimentum
    - **Magna**
    - **Ligula**

Section 1 details with an image. Enter title here.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Enter your main point / statement here.

Section 1 details with a main point. Enter title here.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Section 2 title here.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

- Sollicitudin
- Consectetur
  - Condimentum
    - **Magna**
    - **Ligula**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

- Sollicitudin
- Consectetur
  - Condimentum
    - **Magna**
    - **Ligula**

Section 2 details with an image. Enter title here.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Section 2 details with a main point. Enter title here.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Enter your main point / statement here.

Section 3

Section 3 title here.

Enter your main point / statement here.

Section 3 details with a main point. Enter title here.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Section 4

Section 4 title here.

Section 4 details with a main point. Enter title here.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Enter your main point / statement here.

Enter your main point / statement here.

Here's the timeline.

Event 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.

Event 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.

Event 3

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.