The Economic Limits of Permissionless Consensus

Andrew Lewis-Pye

(LSE)

SBC 2024

The Economic Limits of Permissionless Consensus

Andrew Lewis-Pye
(LSE)

SBC 2024

Eric Budish
(University of Chicago)

Tim Roughgarden
(a16z, Columbia University)

WHY NOT LAUNCH A 51% ATTACK?

Recall: Adversary controlling 51% of the hashrate of a PoW chain (e.g. Bitcoin) or 34% of the stake of a BFT-style PoS chain can cause consistency failures (e.g. to double-spend).

WHY NOT LAUNCH A 51% ATTACK?

Recall: Adversary controlling 51% of the hashrate of a PoW chain (e.g. Bitcoin) or 34% of the stake of a BFT-style PoS chain can cause consistency failures (e.g. to double-spend).

Question: Why wouldn't the adversary launch such an attack?

WHY NOT LAUNCH A 51% ATTACK?

Recall: Adversary controlling 51% of the hashrate of a PoW chain (e.g. Bitcoin) or 34% of the stake of a BFT-style PoS chain can cause consistency failures (e.g. to double-spend).

Question: Why wouldn't the adversary launch such an attack?

This is something Nakamoto considered in the original whitepaper...

NAKAMOTO'S ARGUMENT

51% hashrate $\Rightarrow$ positioned to earn lots of future BTC rewards

double spend $\Rightarrow$ sustained drop in BTC price (USD denominated)

NAKAMOTO'S ARGUMENT

51% hashrate $\Rightarrow$ positioned to earn lots of future BTC rewards

double spend $\Rightarrow$ sustained drop in BTC price (USD denominated)

Modern spin:
51% hashrate $\Rightarrow$ must be heavily invested in ASICS

double spend $\Rightarrow$ community will hard fork to change cryptographic hash function
$\Rightarrow$ ASICS will be worthless!

NAKAMOTO'S ARGUMENT

51% hashrate $\Rightarrow$ positioned to earn lots of future BTC rewards

double spend $\Rightarrow$ sustained drop in BTC price (USD denominated)

Modern spin:
51% hashrate $\Rightarrow$ must be heavily invested in ASICS

double spend $\Rightarrow$ community will hard fork to change cryptographic hash function
$\Rightarrow$ ASICS will be worthless!

Concern: "scorched earth style punishment"

- Honest BTC holders/ASIC owners are collateral damage

NAKAMOTO'S ARGUMENT

51% hashrate $\Rightarrow$ positioned to earn lots of future BTC rewards

double spend $\Rightarrow$ sustained drop in BTC price (USD denominated)

Modern spin:
51% hashrate $\Rightarrow$ must be heavily invested in ASICS

double spend $\Rightarrow$ community will hard fork to change cryptographic hash function
$\Rightarrow$ ASICS will be worthless!

Concern: "scorched earth style punishment"

- Honest BTC holders/ASIC owners are collateral damage

Question: Are these issues fundamental to permissionless consensus, or specific to Bitcoin's design decisions (e.g. PoW)?

THE PROMISE OF PROOF-OF-STAKE

Stake: controlled by the protocol, rather than being "of-chain".

THE PROMISE OF PROOF-OF-STAKE

Stake: controlled by the protocol, rather than being "of-chain".

Hope: double spend $\Rightarrow$ surgically confiscate (or "slash") stake of attacker, leaving honest participants unharmed

THE PROMISE OF PROOF-OF-STAKE

Stake: controlled by the protocol, rather than being "of-chain".

Hope: double spend $\Rightarrow$ surgically confiscate (or "slash") stake of attacker, leaving honest participants unharmed

Challenges:

- Must exist proofs of guilt (cf. accountability [Civit et al.] [Sheng et al.])

THE PROMISE OF PROOF-OF-STAKE

Stake: controlled by the protocol, rather than being "of-chain".

Hope: double spend $\Rightarrow$ surgically confiscate (or "slash") stake of attacker, leaving honest participants unharmed

Challenges:

- Must exist proofs of guilt (cf. accountability [Civit et al.] [Sheng et al.])
- Honest players must receive proof of guilt quickly
  - before adversary can cash out

THE PROMISE OF PROOF-OF-STAKE

Stake: controlled by the protocol, rather than being "of-chain".

Hope: double spend ⇒ surgically confiscate (or "slash") stake of attacker, leaving honest participants unharmed

Challenges:

- Must exist proofs of guilt (cf. accountability [Civit et al.] [Sheng et al.])
- Honest players must receive proof of guilt quickly
  - before adversary can cash out
- Honest players much reach consensus on punishment/new state
  - adversary might be large enough to prevent this

CASE STUDY: ETHEREUM

The intention is to make 51% attacks extremely expensive, so that even a majority of validators working together cannot roll back finalized blocks without undertaking an extremely large economic loss — a loss so large that

Minimal Slashing Conditions

Vitalik Buterin · Follow

14 min read · Mar 2, 2017

CASE STUDY: ETHEREUM

The intention is to make 51% attacks extremely expensive, so that even a majority of validators working together cannot roll back finalized blocks without undertaking an extremely large economic loss — a loss so large that

Minimal Slashing Conditions

Vitalik Buterin · Follow
14 min read · Mar 2, 2017

PROOF-OF-STAKE (POS)

Last edit: @nhsz, January 25, 2024

See contributors

Proof-of-stake (PoS) underlies Ethereum's consensus mechanism. Ethereum switched on its proof-of-stake mechanism in 2022 because it is more secure, less energy-intensive, and better for implementing new scaling solutions compared to the previous proof-of-work architecture.

PROS AND CONS

Pros

Proof-of-stake offers greater crypto-economic security than proof-of-work

- economic penalties for misbehavior make 51% style attacks more costly for an attacker compared to proof-of-work

source: official Ethereum documentation at https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/

CAN SLASHING FULFIL ITS PROMISE?

Questions:

- Is PoS more "economically secure" than PoW?
  - What does this even mean?

CAN SLASHING FULFIL ITS PROMISE?

Questions:

- Is PoS more "economically secure" than PoW?
  - What does this even mean?
- How "econonomically secure" can a protocol be?
  - Could there exist protocols more secure than today's Ethereum?

CAN SLASHING FULFIL ITS PROMISE?

Questions:

- Is PoS more "economically secure" than PoW?
  - What does this even mean?
- How "econonomically secure" can a protocol be?
  - Could there exist protocols more secure than today's Ethereum?
- How does the answer depend on our assumptions?
  - About message delays, active participation by honest players, etc.

CAN SLASHING FULFIL ITS PROMISE?

Questions:

- Is PoS more "economically secure" than PoW?
- How "econonomically secure" can a protocol be?
- How does the answer depend on our assumptions?

Key definition: An EAAC protocol [i.e. what we want]

(1) Honest players never lose resources (e.g. don't get slashed)
(2) Double Spend $\Rightarrow$ attacker guaranteed to lose resources

IMPOSSIBILITY RESULTS

Note: if protocol is secure (no consistency violations) with f-bounded attacker (f =fraction of stake), trivially EAAC.

- goal: want to be EAAC beyond the security threshold.

IMPOSSIBILITY RESULTS

Note: if protocol is secure (no consistency violations) with f-bounded attacker (f =fraction of stake), trivially EAAC.

- goal: want to be EAAC beyond the security threshold.

Theorem 1. In the dynamically available and synchronous setting, with a $\frac{1}{2}$-bounded attacker, no protocol is EAAC.

- Bitcoin is trivially EAAC for all $f < \frac{1}{2}$ [Garay/Kiayias/Leonardos 2015]
- Economic counterpart to [Neu/Tas/Tse 2022]

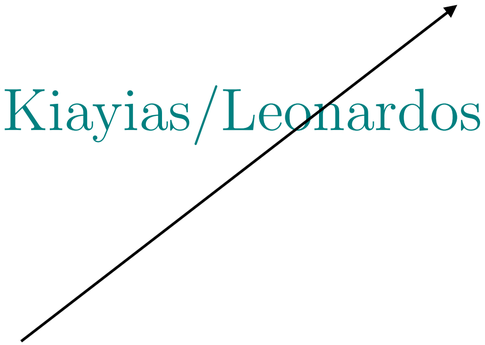
IMPOSSIBILITY RESULTS

Note: if protocol is secure (no consistency violations) with f-bounded attacker (f =fraction of stake), trivially EAAC.

- goal: want to be EAAC beyond the security threshold.

Theorem 1. In the dynamically available and synchronous setting, with a $\frac{1}{2}$-bounded attacker, no protocol is EAAC.

- Bitcoin is trivially EAAC for all $f < \frac{1}{2}$ [Garay/Kiayias/Leonardos 2015]
- Economic counterpart to [Neu/Tas/Tse 2022]



adversary controls 50% of stake/hashrate

IMPOSSIBILITY RESULTS

Note: if protocol is secure (no consistency violations) with f-bounded attacker (f =fraction of stake), trivially EAAC.

- goal: want to be EAAC beyond the security threshold.

Theorem 1. In the dynamically available and synchronous setting, with a $\frac{1}{2}$-bounded attacker, no protocol is EAAC.

- Bitcoin is trivially EAAC for all $f < \frac{1}{2}$ [Garay/Kiayias/Leonardos 2015]
- Economic counterpart to [Neu/Tas/Tse 2022]

Communication is reliable: there exists a known bound on message delay

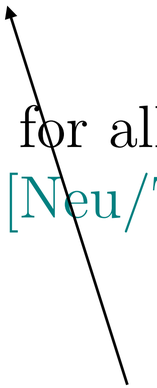
IMPOSSIBILITY RESULTS

Note: if protocol is secure (no consistency violations) with f-bounded attacker (f =fraction of stake), trivially EAAC.

- goal: want to be EAAC beyond the security threshold.

Theorem 1. In the dynamically available and synchronous setting, with a $\frac{1}{2}$-bounded attacker, no protocol is EAAC.

- Bitcoin is trivially EAAC for all $f < \frac{1}{2}$ [Garay/Kiayias/Leonardos 2015]
- Economic counterpart to [Neu/Tas/Tse 2022]



honest participants with resources may be inactive

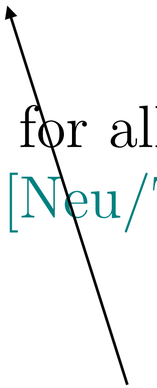
IMPOSSIBILITY RESULTS

Note: if protocol is secure (no consistency violations) with f-bounded attacker (f =fraction of stake), trivially EAAC.

- goal: want to be EAAC beyond the security threshold.

Theorem 1. In the dynamically available and synchronous setting, with a $\frac{1}{2}$-bounded attacker, no protocol is EAAC.

- Bitcoin is trivially EAAC for all $f < \frac{1}{2}$ [Garay/Kiayias/Leonardos 2015]
- Economic counterpart to [Neu/Tas/Tse 2022]



Note: BFT-style protocols are generally NOT live in this setting.

IMPOSSIBILITY RESULTS

Note: if protocol is secure (no consistency violations) with f-bounded attacker (f =fraction of stake), trivially EAAC.

- goal: want to be EAAC beyond the security threshold.

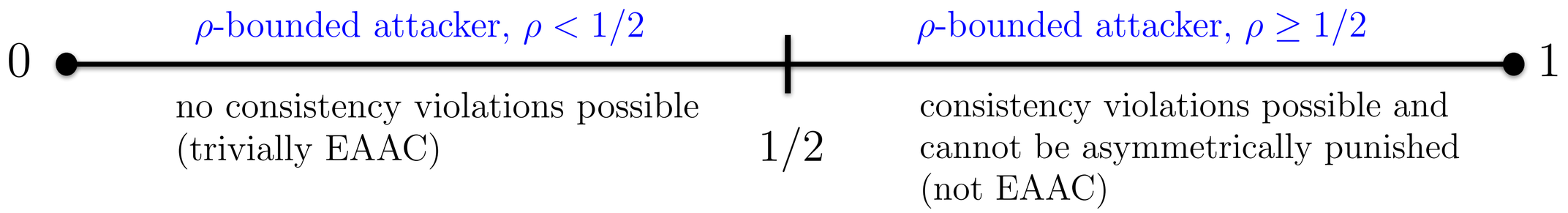
Theorem 1. In the dynamically available and synchronous setting, with a $\frac{1}{2}$-bounded attacker, no protocol is EAAC.

- Bitcoin is trivially EAAC for all $f < \frac{1}{2}$ [Garay/Kiayias/Leonardos 2015]
- Economic counterpart to [Neu/Tas/Tse 2022]

IMPOSSIBILITY RESULTS (CONTINUED)

Goal: want to be EAAC beyond the security threshold f.

Theorem 1. In the dynamically available and synchronous setting, with a $\frac{1}{2}$-bounded attacker, no protocol is EAAC.

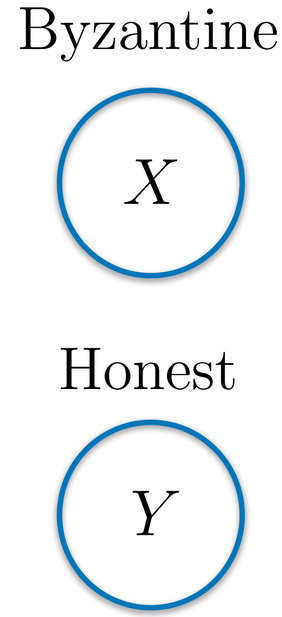


Theorem 1. In the dynamically available and synchronous setting, with a $\frac{1}{2}$-bounded attacker, no protocol is EAAC.

- can't implement slashing in a typical longest-chain-style protocol

Proof idea: let player sets X, Y have equal resources:

- X Byzantine, proceeds as if Y does not exist. Liveness in DA setting $\Rightarrow$ X, Y must confirm (conflicting) transactions

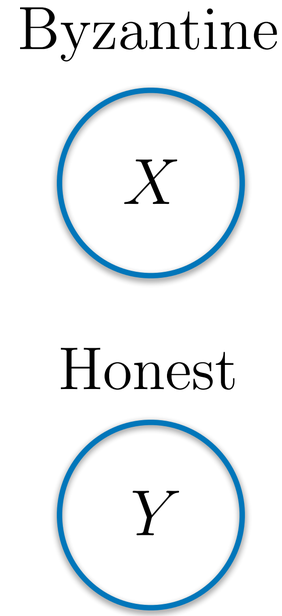


Theorem 1. In the dynamically available and synchronous setting, with a $\frac{1}{2}$-bounded attacker, no protocol is EAAC.

- can't implement slashing in a typical longest-chain-style protocol

Proof idea: let player sets X, Y have equal resources:

- X Byzantine, proceeds as if Y does not exist. Liveness in DA setting $\Rightarrow$ X, Y must confirm (conflicting) transactions
- Later, X disseminates all messages from its simulation

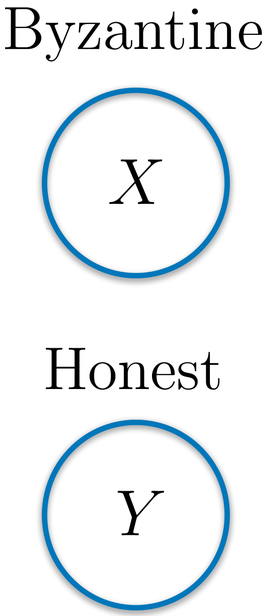


Theorem 1. In the dynamically available and synchronous setting, with a $\frac{1}{2}$-bounded attacker, no protocol is EAAC.

- can't implement slashing in a typical longest-chain-style protocol

Proof idea: let player sets X, Y have equal resources:

- X Byzantine, proceeds as if Y does not exist. Liveness in DA setting $\Rightarrow$ X, Y must confirm (conflicting) transactions
- Later, X disseminates all messages from its simulation
- Late arriving players can't distinguish which of X, Y is Byzantine

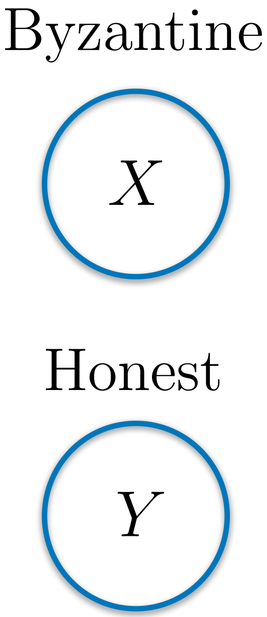


Theorem 1. In the dynamically available and synchronous setting, with a $\frac{1}{2}$-bounded attacker, no protocol is EAAC.

- can't implement slashing in a typical longest-chain-style protocol

Proof idea: let player sets X, Y have equal resources:

- X Byzantine, proceeds as if Y does not exist. Liveness in DA setting $\Rightarrow$ X, Y must confirm (conflicting) transactions
- Later, X disseminates all messages from its simulation
- Late arriving players can't distinguish which of X, Y is Byzantine
- No asymmetric punishment is possible!



IMPOSSIBILITY RESULTS (CONTINUED)

Goal: want to be EAAC beyond the security threshold f.

Theorem 1. In the dynamically available and synchronous setting, with a $\frac{1}{2}$-bounded attacker, no protocol is EAAC.

Theorem 2. In the quasi-permissionless and partially synchronous setting, with a $\frac{1}{3}$-bounded attacker, no protocol is EAAC.

- Security threshold is 1/3.
- Stronger: can't punish in the event of a consistency violation at all (even with collateral damage)!

IMPOSSIBILITY RESULTS (CONTINUED)

Goal: want to be EAAC beyond the security threshold f.

Theorem 1. In the dynamically available and synchronous setting, with a $\frac{1}{2}$-bounded attacker, no protocol is EAAC.

Theorem 2. In the quasi-permissionless and partially synchronous setting, with a $\frac{1}{3}$-bounded attacker, no protocol is EAAC.

- Security threshold is 1/3.
- Stronger: can't punish in the event of a consistency violation at all (even with collateral damage)!

honest players with resources are always active

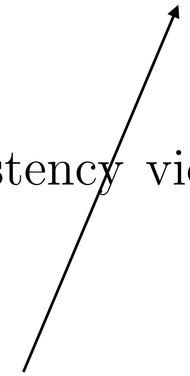
IMPOSSIBILITY RESULTS (CONTINUED)

Goal: want to be EAAC beyond the security threshold f.

Theorem 1. In the dynamically available and synchronous setting, with a $\frac{1}{2}$-bounded attacker, no protocol is EAAC.

Theorem 2. In the quasi-permissionless and partially synchronous setting, with a $\frac{1}{3}$-bounded attacker, no protocol is EAAC.

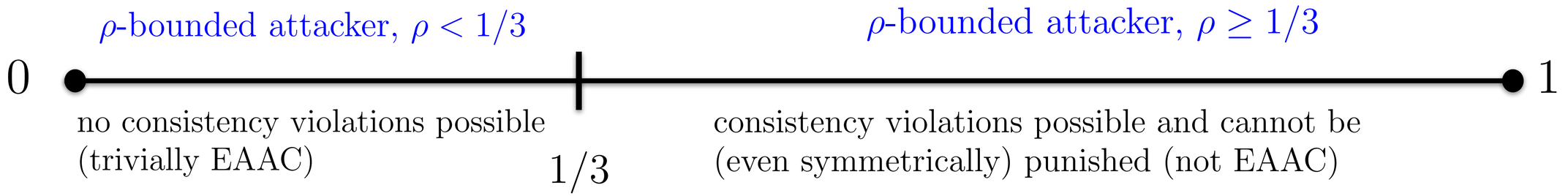
- Security threshold is 1/3.
- Stronger: can't punish in the event of a consistency violation at all (even with collateral damage)!



communication network may sometimes be unreliable

IMPOSSIBILITY RESULTS (CONTINUED)

Theorem 2. In the quasi-permissionless and partially synchronous setting, with a $\frac{1}{3}$-bounded attacker, no protocol is EAAC.



POSSIBILITY RESULT

Goal: want to be EAAC beyond the security threshold f.

POSSIBILITY RESULT

Goal: want to be EAAC beyond the security threshold f.

Theorem 3 (basic version). There is an EAAC protocol in the quasi-permissionless and synchronous setting with an f-bounded attacker for $f < 2/3$.

POSSIBILITY RESULT

Goal: want to be EAAC beyond the security threshold f.

Theorem 3 (basic version). There is an EAAC protocol in the quasi-permissionless and synchronous setting with an f-bounded attacker for $f < 2/3$.

Theorem 3 (refined version). For parameters Δ_1 (e.g. a second) and Δ_2 (e.g. hours), in the quasi-permissionless and partially synchronous setting (with parameter Δ_1), there is a protocol that is:

POSSIBILITY RESULT

Goal: want to be EAAC beyond the security threshold f.

Theorem 3 (basic version). There is an EAAC protocol in the quasi-permissionless and synchronous setting with an f-bounded attacker for $f < 2/3$.

Theorem 3 (refined version). For parameters Δ_1 (e.g. a second) and Δ_2 (e.g. hours), in the quasi-permissionless and partially synchronous setting (with parameter Δ_1), there is a protocol that is:

- Consistent and live with an f-bounded attacker for $f < 1/3$ (with latency $O(\Delta_1)$).

POSSIBILITY RESULT

Goal: want to be EAAC beyond the security threshold f.

Theorem 3 (basic version). There is an EAAC protocol in the quasi-permissionless and synchronous setting with an f-bounded attacker for $f < 2/3$.

Theorem 3 (refined version). For parameters Δ_1 (e.g. a second) and Δ_2 (e.g. hours), in the quasi-permissionless and partially synchronous setting (with parameter Δ_1), there is a protocol that is:

- Consistent and live with an f-bounded attacker for $f < 1/3$ (with latency $O(\Delta_1)$).
- EAAC for an f-bounded attacker with $f < 2/3$ whenever all message delays pre-GST are at most Δ_2.
    - NB: cooldown period for unstaking proportional to Δ_2.

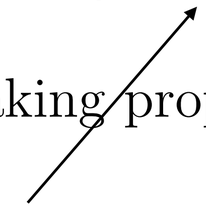
POSSIBILITY RESULT

Goal: want to be EAAC beyond the security threshold f.

Theorem 3 (basic version). There is an EAAC protocol in the quasi-permissionless and synchronous setting with an f-bounded attacker for $f < 2/3$.

Theorem 3 (refined version). For parameters Δ_1 (e.g. a second) and Δ_2 (e.g. hours), in the quasi-permissionless and partially synchronous setting (with parameter Δ_1), there is a protocol that is:

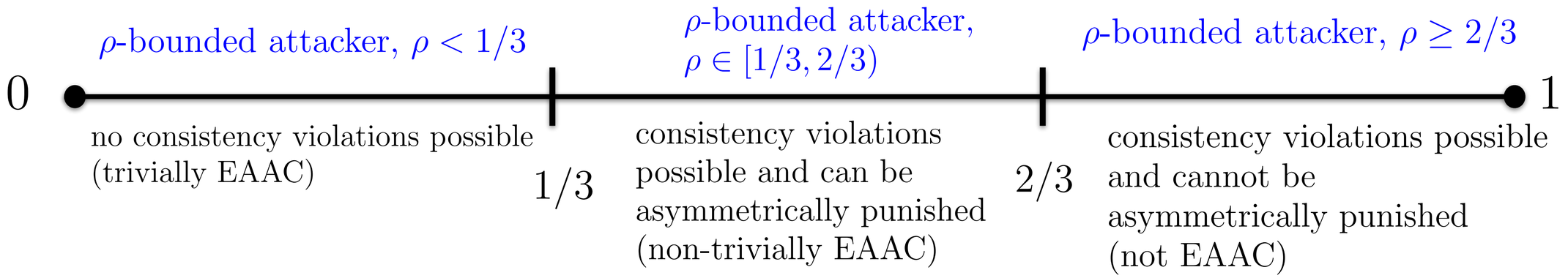
- Consistent and live with an f-bounded attacker for $f < 1/3$ (with latency $O(\Delta_1)$).
- EAAC for an f-bounded attacker with $f < 2/3$ whenever all message delays pre-GST are at most Δ_2.
  - NB: cooldown period for unstaking proportional to Δ_2.



2/3 is best possible [Tas et al. 2023]

Theorem 3 (refined version). For parameters Δ_1 (e.g. a second) and Δ_2 (e.g. hours), in the quasi-permissionless and partially synchronous setting (with parameter Δ_1), there is a protocol that is:

- Consistent and live with an f-bounded attacker for $f < 1/3$ (with latency $O(\Delta_1)$).
- EAAC for an f-bounded attacker with $f < 2/3$ whenever all message delays pre-GST are at most Δ_2.
  - NB: cooldown period for unstaking proportional to Δ_2.



The PoST protocol

Starting point: Tendermint (2 stages of voting per view)

- Execution divided into "epochs", each tasked with finalizing N blocks.
- Player set updated at each epoch (according to new stake amounts).

The PoST protocol

Starting point: Tendermint (2 stages of voting per view)

- Execution divided into "epochs", each tasked with finalizing N blocks.
- Player set updated at each epoch (according to new stake amounts).

Recall requirements:

- must exist proof of guilt

The PoST protocol

Starting point: Tendermint (2 stages of voting per view)

- Execution divided into "epochs", each tasked with finalizing N blocks.
- Player set updated at each epoch (according to new stake amounts).

Recall requirements:

- must exist proof of guilt
- honest players must receive proof of guilt quickly
  - worry: adversary might be powerful enough to prevent this

The PoST protocol

Starting point: Tendermint (2 stages of voting per view)

- Execution divided into "epochs", each tasked with finalizing N blocks.
- Player set updated at each epoch (according to new stake amounts).

Recall requirements:

- must exist proof of guilt
- honest players must receive proof of guilt quickly
  - worry: adversary might be powerful enough to prevent this
- honest players must reach consensus on punishment/new state
  - worry: ditto

The PoST protocol

Starting point: Tendermint (2 stages of voting per view)

- Execution divided into "epochs", each tasked with finalizing N blocks.
- Player set updated at each epoch (according to new stake amounts).

Challenge 1: adversary can delay release of conflicting stage-2 QCs

The PoST protocol

Starting point: Tendermint (2 stages of voting per view)

- Execution divided into "epochs", each tasked with finalizing N blocks.
- Player set updated at each epoch (according to new stake amounts).

Challenge 1: adversary can delay release of conflicting stage-2 QCs

- solution: 3 stages of voting rather than 2.
  - Cannot cause stage 3 conflicts without honest players seeing stage 2 conflicts.

The PoST protocol

Starting point: Tendermint (2 stages of voting per view)

- Execution divided into "epochs", each tasked with finalizing N blocks.
- Player set updated at each epoch (according to new stake amounts).

Challenge 1: adversary can delay release of conflicting stage-2 QCs

- solution: 3 stages of voting rather than 2.
  - Cannot cause stage 3 conflicts without honest players seeing stage 2 conflicts.

Challenge 2: if $f > 5/9$, post slashing, adversary still not 1/3-bounded

- Seems they will be able to kill liveness, preventing consensus on slashing conditions

Challenge 2: going beyond 5/9

Challenge 2: if $f > 5/9$, post slashing, adversary still not 1/3-bounded

Solution: upon seeing proof of guilt, honest players switch to "recovery mode", ensuring liveness via lower threshold for QCs

Challenge 2: going beyond 5/9

Challenge 2: if $f > 5/9$, post slashing, adversary still not 1/3-bounded

Solution: upon seeing proof of guilt, honest players switch to "recovery mode", ensuring liveness via lower threshold for QCs

Issue: consistency violations even easier in recovery mode

Challenge 2: going beyond 5/9

Challenge 2: if $f > 5/9$, post slashing, adversary still not 1/3-bounded

Solution: upon seeing proof of guilt, honest players switch to "recovery mode", ensuring liveness via lower threshold for QCs

Issue: consistency violations even easier in recovery mode

- solution: reboot epoch with consistency violation as many times as needed (slashing each time) until adversary $< 1/3$-bounded
  - increase QC threshold with each reboot, eventually get back to 2/3rds.

SUMMARY

- Cannot carry out slashing (or any asymmetric punishment) without strong assumptions on the active participation of honest stakers.
  - Rules out typical longest-chain protocols (e.g. Ethereum 1.0).

SUMMARY

- Cannot carry out slashing (or any asymmetric punishment) without strong assumptions on the active participation of honest stakers.
  - Rules out typical longest-chain protocols (e.g. Ethereum 1.0).
- Cannot carry out slashing without bounds on worst-case message delays.
  - Accords with use of long cooldown periods in Ethereum 2.0.

SUMMARY

- Cannot carry out slashing (or any asymmetric punishment) without strong assumptions on the active participation of honest stakers.
  - Rules out typical longest-chain protocols (e.g. Ethereum 1.0).
- Cannot carry out slashing without bounds on worst-case message delays.
  - Accords with use of long cooldown periods in Ethereum 2.0.
- Can augment PBFT/Tendermint-style protocols to carry out slashing, provably recover from any $< 2/3$-bounded attacker.
  - To go from 5/9 to 2/3 (efficiently), trade-off consistency for liveness in recovery mode.

Thanks for listening!