A Two-Layer Blockchain Sharding Protocol Leveraging Safety and Liveness for Enhanced Performance

Yibin Xu, Jingyi Zheng, Boris D¨udder, Tijs Slaats, Yongluan Zhou

Department of Computer Science, University of Copenhagen

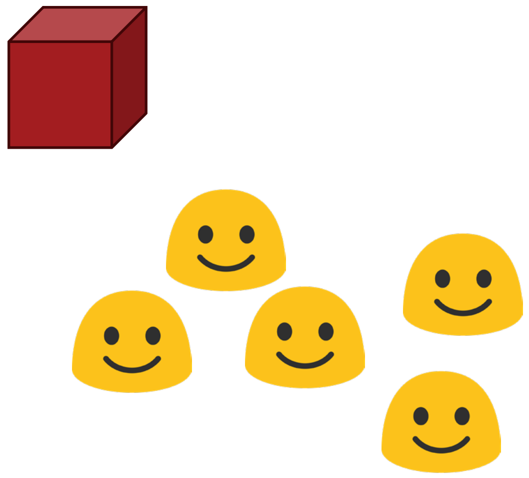
KØBENHAVNS UNIVERSITET

NDSS SYMPOSIUM/2024

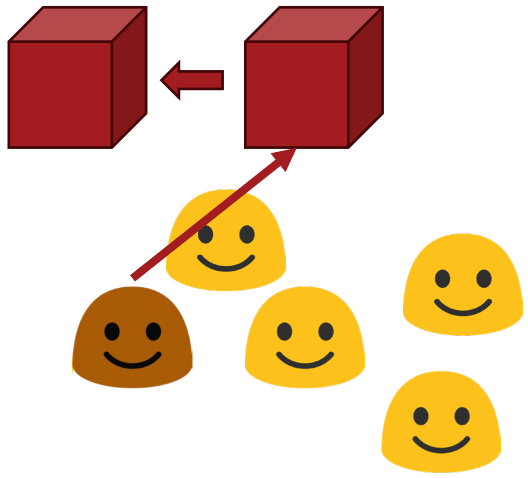
Presented by Internet Society

#NDSSSymposium2024

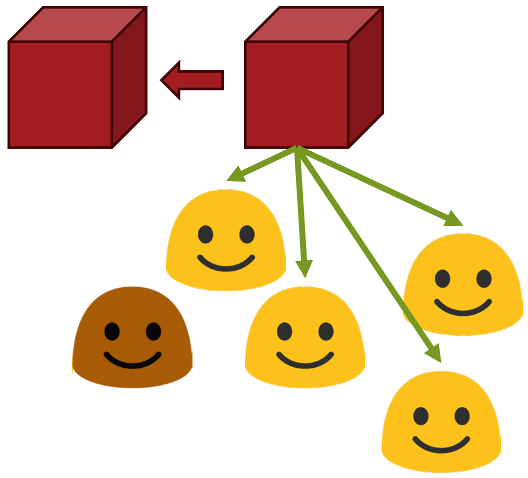
Blockchain without sharding



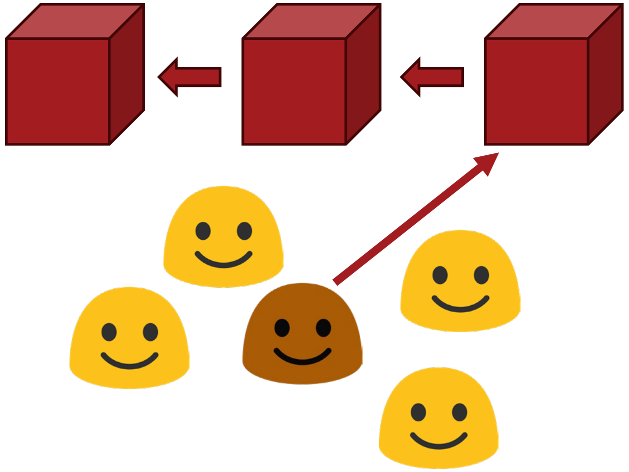
Blockchain without sharding



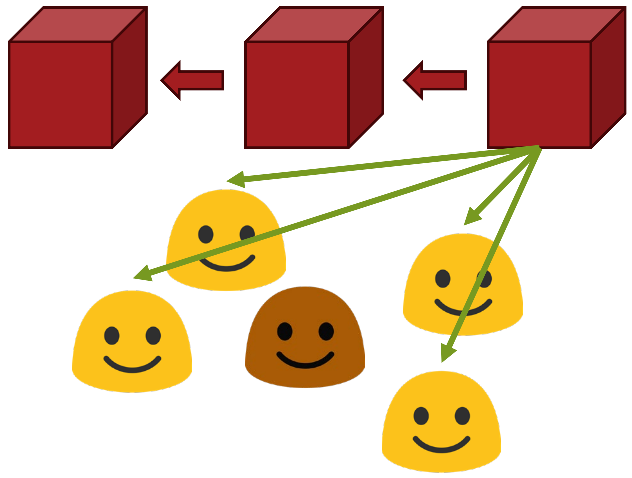
Blockchain without sharding



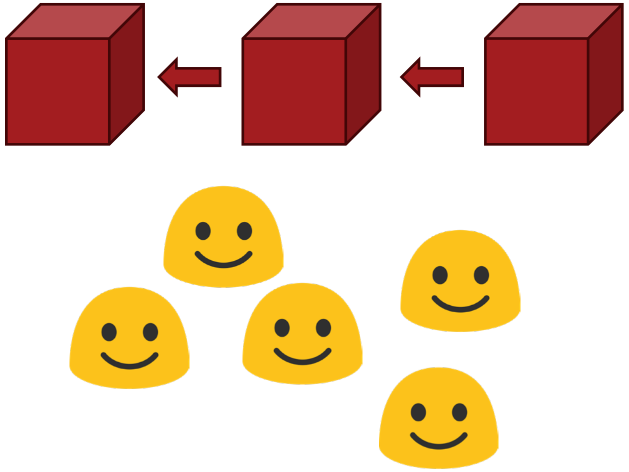
Blockchain without sharding



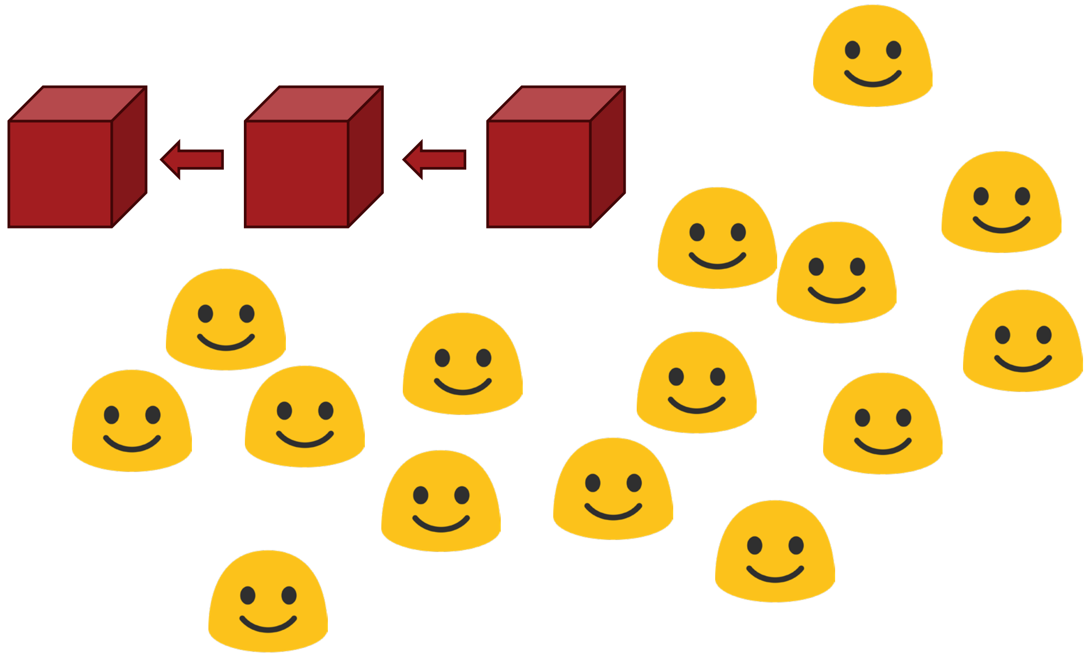
Blockchain without sharding



Blockchain without sharding

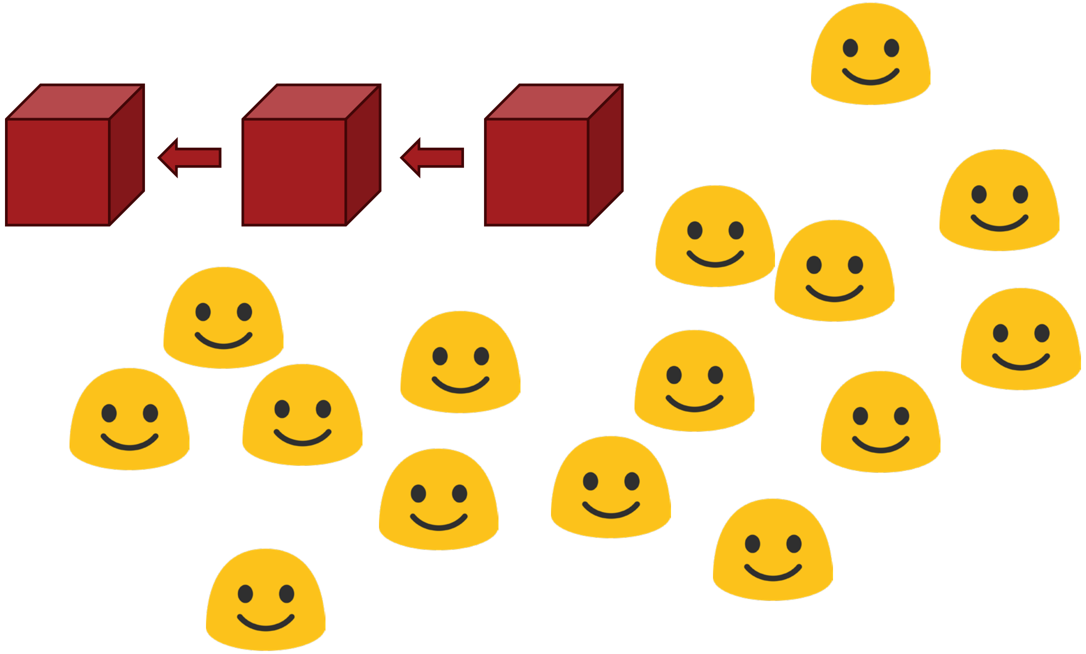


Blockchain without sharding



Scalability problem with increased servers.
Not able to increase the size of blocks

Blockchain without sharding

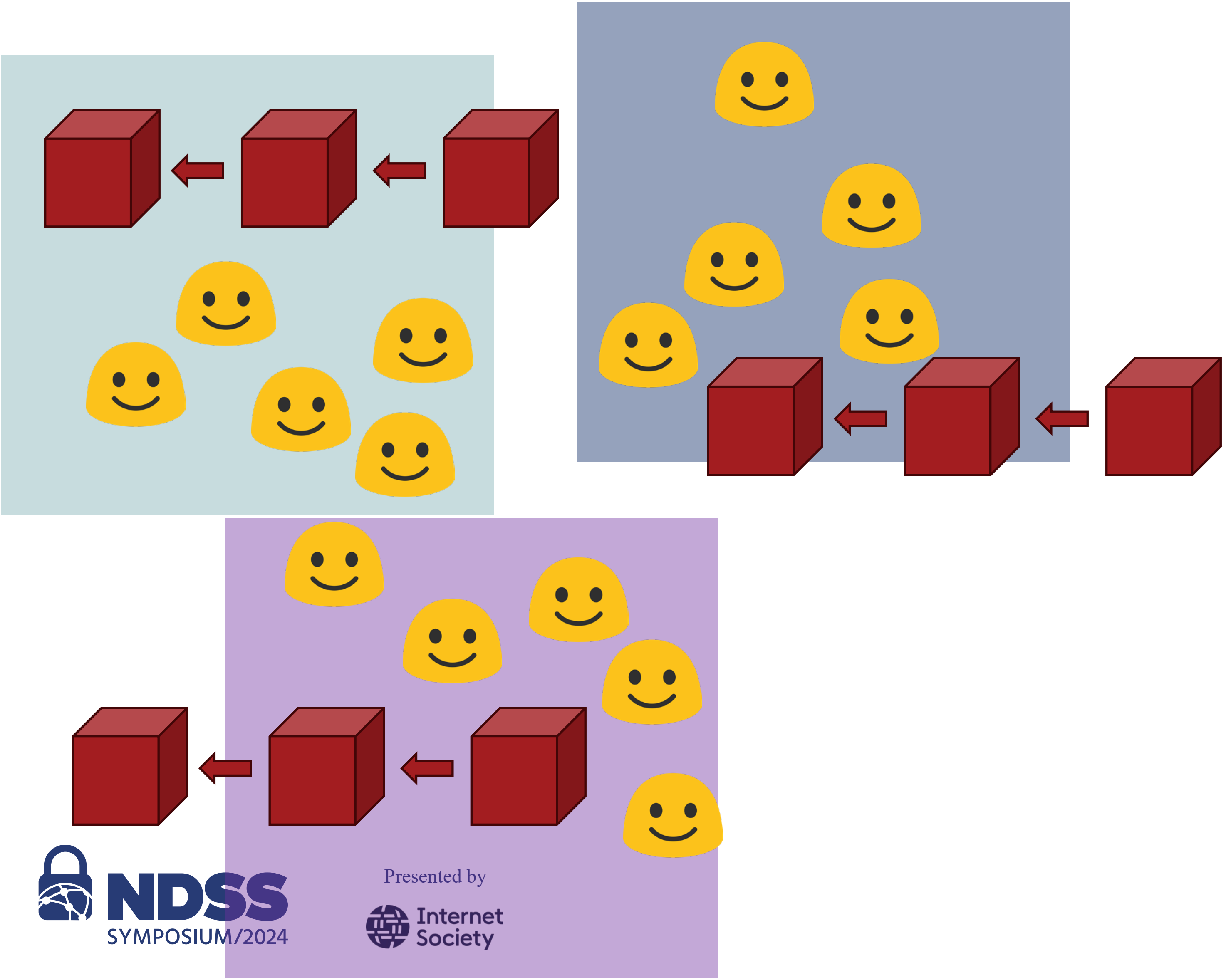


All nodes verify all blocks

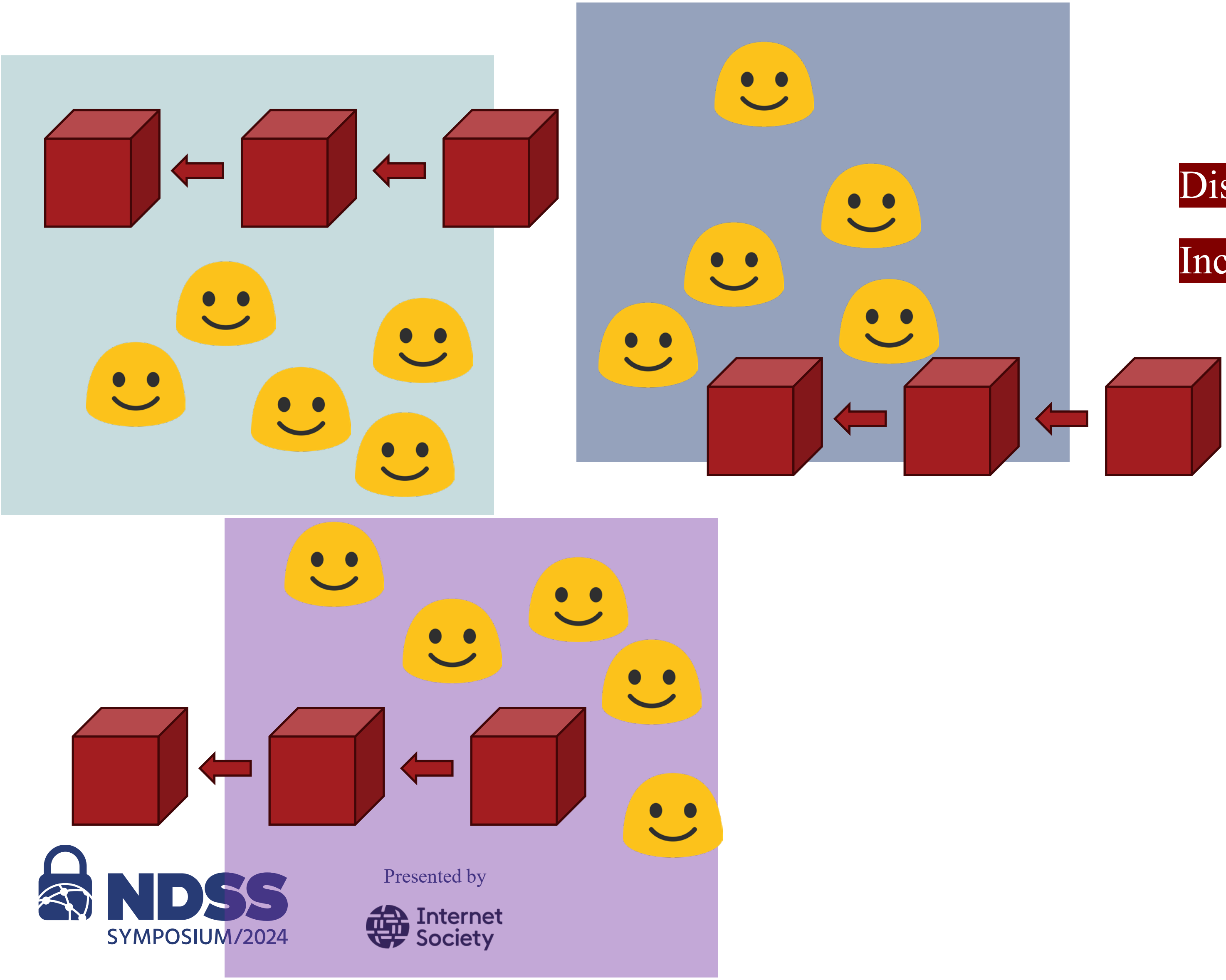
Increasing # nodes does not improve efficiency

Scalability problem with increased servers.
Not able to increase the size of blocks

Basic Idea of Sharding



Basic Idea of Sharding



Distributed nodes to separate shards (chains)

Increasing # nodes improves efficiency

Basic Approach

Randomly assign nodes to shards + Run BFT protocols in each shard

Asynchronous: Kokoris-Kogias, Eleftherios, et al. "Omniledger: A secure, scale-out, decentralized ledger via sharding." 2018 IEEE symposium on security and privacy (SP). IEEE, 2018.

Synchronous: Zamani, Mahdi, Mahnush Movahedi, and Mariana Raykova. "Rapidchain: Scaling blockchain via full sharding." Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. 2018.

Basic Approach

How large a shard needs to be?

Randomly assign nodes to shards

$$P_f = \sum_{i=\lfloor \frac{1}{2} M \rfloor + 1}^{M} \binom{M}{i} (P_a)^i (1 - P_a)^{M-i} \leq 2^{-\sigma}$$

Synchronous bound

M: Number of nodes in a shard.
Pa: The population percentage of adversarial nodes in maximum
σ: The failure probability.

Basic Approach

How large a shard needs to be?

Randomly assign nodes to shards

$$P_f = \sum_{i=\lfloor \frac{1}{2}M \rfloor + 1}^{M} \binom{M}{i} (P_a)^i (1 - P_a)^{M-i} \leq 2^{-\sigma}$$

Synchronous bound

M: Number of nodes in a shard.
Pa: The population percentage of adversarial nodes in maximum
σ: The failure probability.

The size of a shard is determined by the security parameter σ and the adversarial population percentage Pa

$$P_f = \sum_{i=\lfloor \frac{1}{2}M \rfloor+1}^{M} \binom{M}{i} (P_a)^i (1-P_a)^{M-i} \leq 2^{-\sigma}$$

$$P_f = \sum_{i=\lfloor \cancel{\frac{1}{2}} M \rfloor + 1}^{M} \binom{M}{i} (P_a)^i (1 - P_a)^{M-i} \leq 2^{-\sigma}$$

$$P_f = \sum_{i=\lfloor \underset{\frac{2}{3}}{\cancel{\frac{1}{2}}} M \rfloor + 1}^{M} \binom{M}{i} (P_a)^i (1 - P_a)^{M-i} \leq 2^{-\sigma}$$

$$P_f = \sum_{i=\lfloor \cancel{\frac{1}{2}} M \rfloor + 1}^{M} \binom{M}{i} (P_a)^i (1 - P_a)^{M-i} \leq 2^{-\sigma}$$

$$\frac{2}{3}$$

$$P_f = \sum_{i=\lfloor \cancel{\frac{1}{2}} M \rfloor + 1}^{M} \binom{M}{i} (P_a)^i (1 - P_a)^{M-i} \leq 2^{-\sigma}$$

$$\cancel{\frac{2}{3}}$$

$$\frac{5}{6}$$

$$P_f = \sum_{i=\lfloor \cancel{\frac{1}{2}} M \rfloor+1}^{M} \binom{M}{i} (P_a)^i (1 - P_a)^{M-i} \leq 2^{-\sigma}$$

$$\cancel{\frac{2}{3}} \quad \cancel{\frac{5}{6}}$$

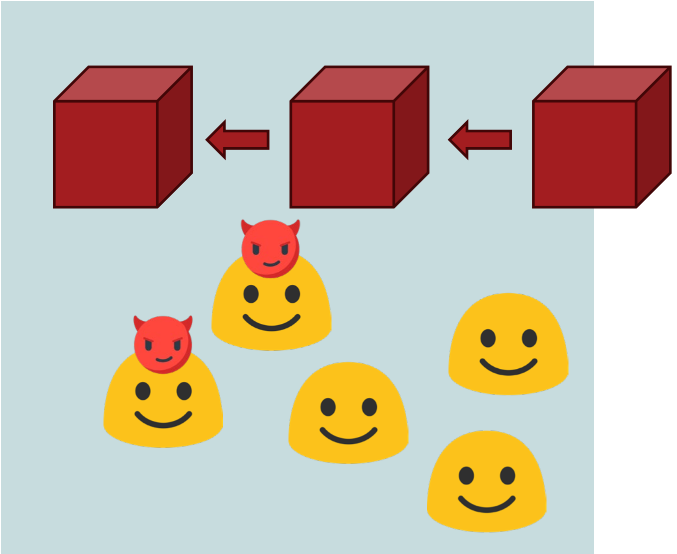
$$P_f = \sum_{i=\lfloor \cancel{\frac{1}{2}} M \rfloor + 1}^{M} \binom{M}{i} (P_a)^i (1 - P_a)^{M-i} \leq 2^{-\sigma}$$

$$\cancel{\frac{2}{3}}$$

$$\cancel{\frac{5}{6}}$$

Decrease M to allow smaller shards

A normal shard

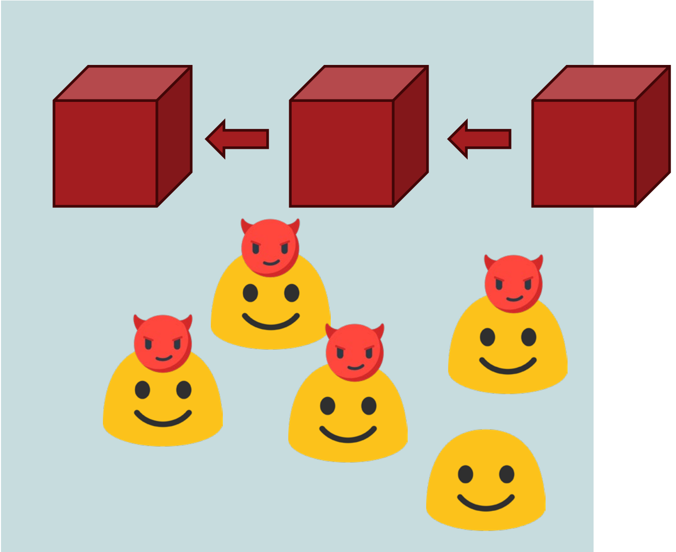


S=L<50%

S: The maximum percentage of nodes being adversarial allowed in a shard to generate a correct verdict.

L: The maximum percentage of nodes being adversarial allowed in a shard to generate a verdict.

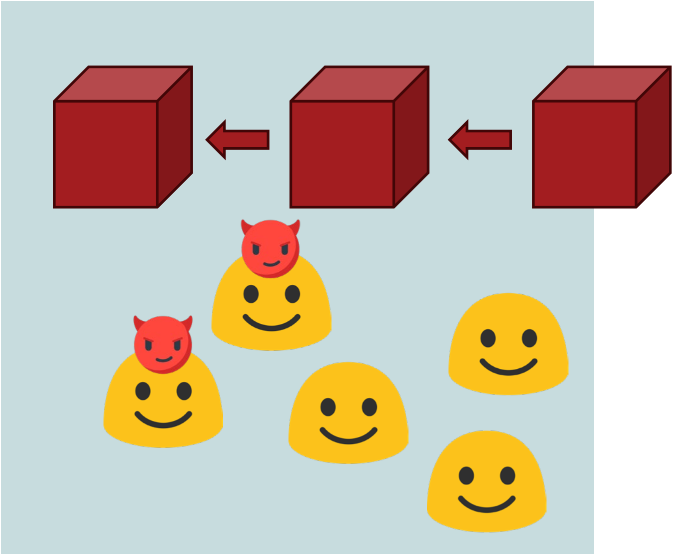
A shard leveraging liveness and safety



S=80%

L<20%

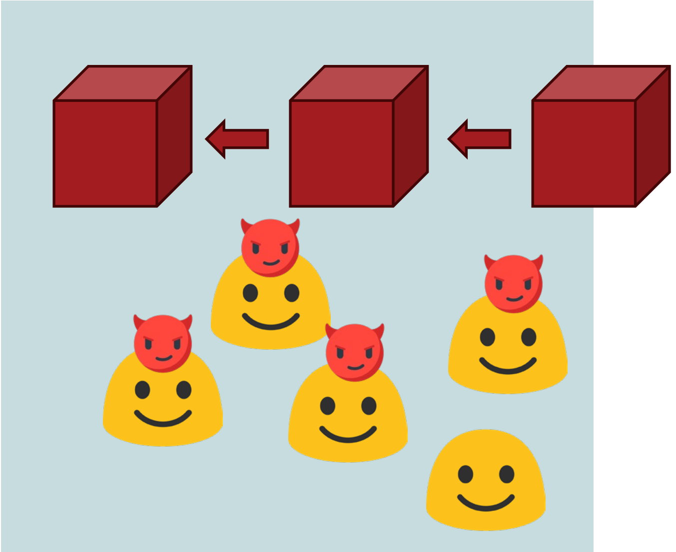
A normal shard



S=L<50%

Liveness issue!!!

A shard leveraging liveness and safety



S=80%

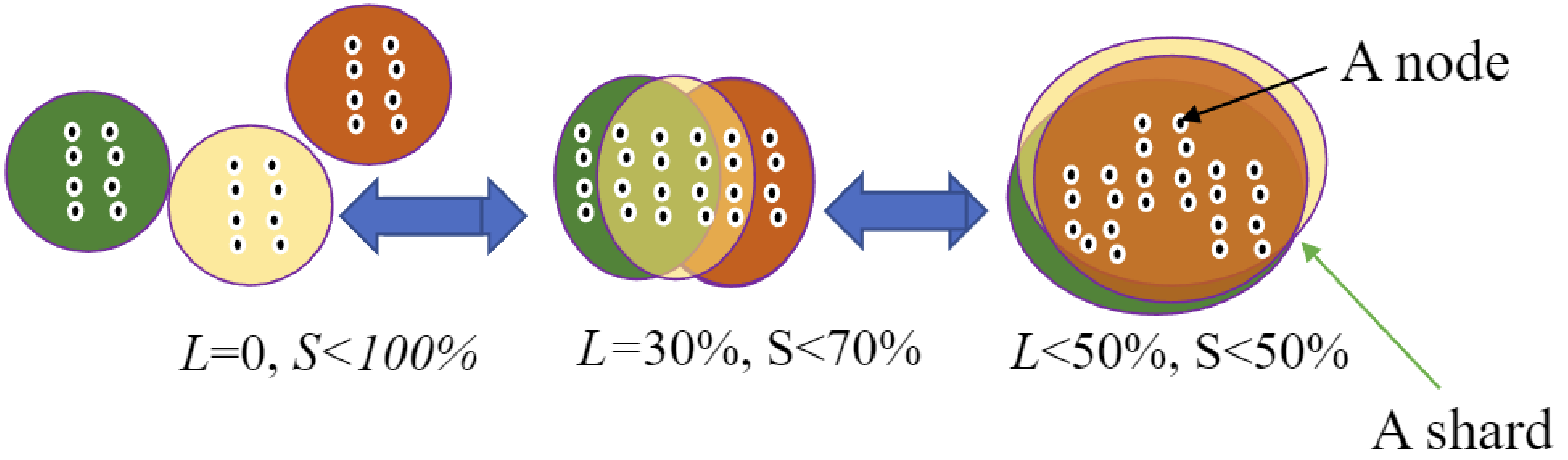
L<20%

S: The maximum percentage of nodes being adversarial allowed in a shard to generate a correct verdict.

L: The maximum percentage of nodes being adversarial allowed in a shard to generate a verdict.

Leveraging Safety and Liveness to increase the number of shards

Randomly assign nodes to shards + Run BFT protocols in each shard + Global consensus to relive shards



Relive the shards by reconstruct all the shards

Xu, Yibin, et al. "A flexible n/2 adversary node resistant and halting recoverable blockchain sharding protocol." Concurrency and Computation: Practice and Experience 32.19 (2020): e5773.

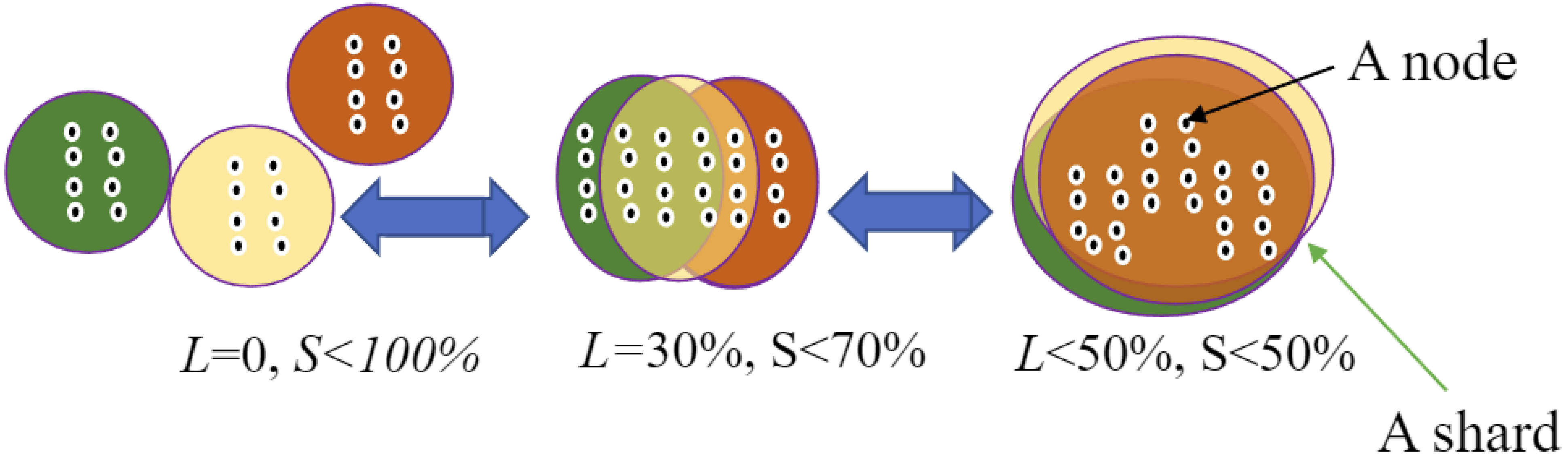
Relive the shards extend the shard size and overlap with other shards

David, Bernardo, et al. "GearBox: Optimal-size Shard Committees by Leveraging the Safety-Liveness Dichotomy." Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. 2022.

Leveraging Safety and Liveness to increase the number of shards

Randomly assign nodes to shards + Run BFT protocols in each shard + Global consensus to relive shards

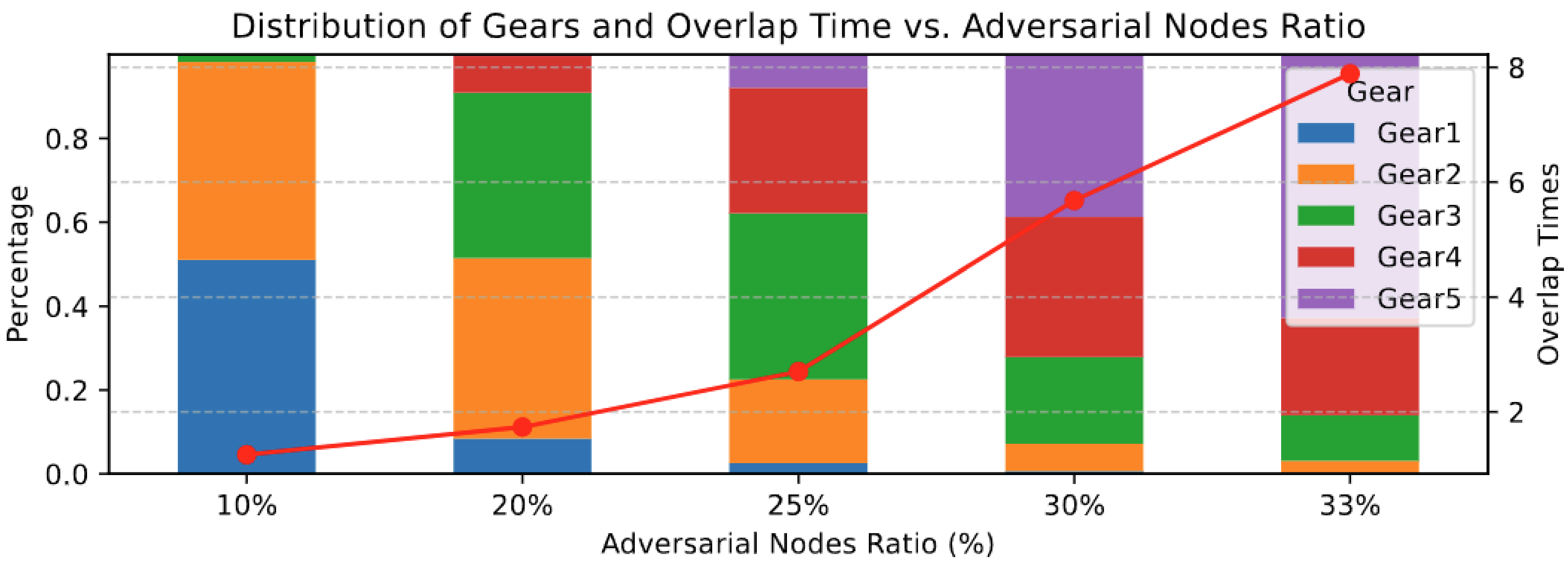
Adjust only the shard size but not shard number, resulting huge overlapping.



Leveraging Safety and Liveness to increase the number of shards

Randomly assign nodes to shards + Run BFT protocols in each shard + Global consensus to relive shards

Adjust only the shard size but not shard number, resulting huge overlapping.

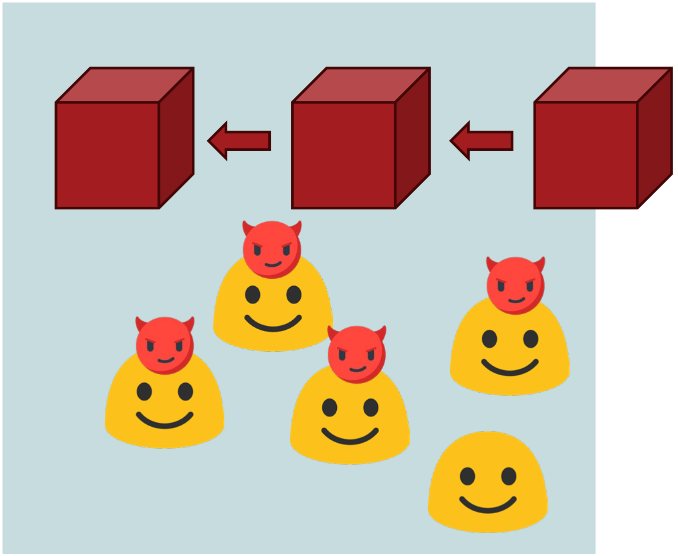


Security issues

The communication model only guarantees that the honest nodes receive the message from each other

A shard leveraging liveness and safety

We need the help from an adversary to reach a verdict.



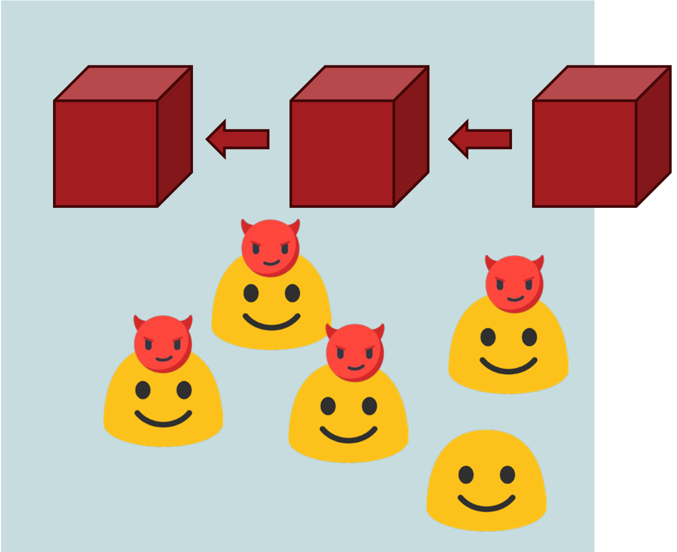
This verdict requires at least one honest node, so the verdict is correct.

However, there is no guarantee that the adversary's votes are received by other nodes. Equivocation problem!!

S=80%

L<20%

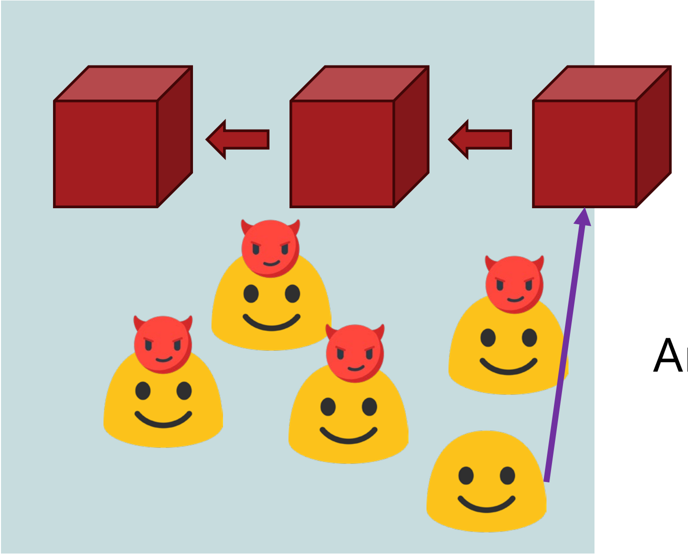
A shard leveraging liveness and safety



S=80%

L<20%

A shard leveraging liveness and safety

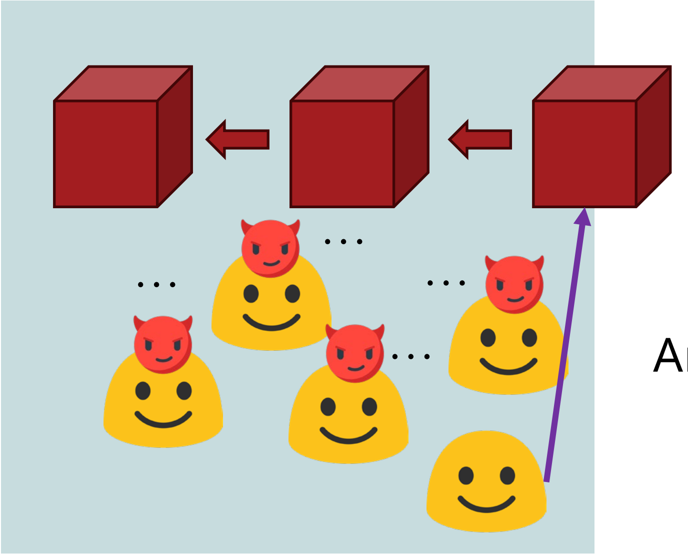


An honest node voted to accept this block.

S=80%

L<20%

A shard leveraging liveness and safety



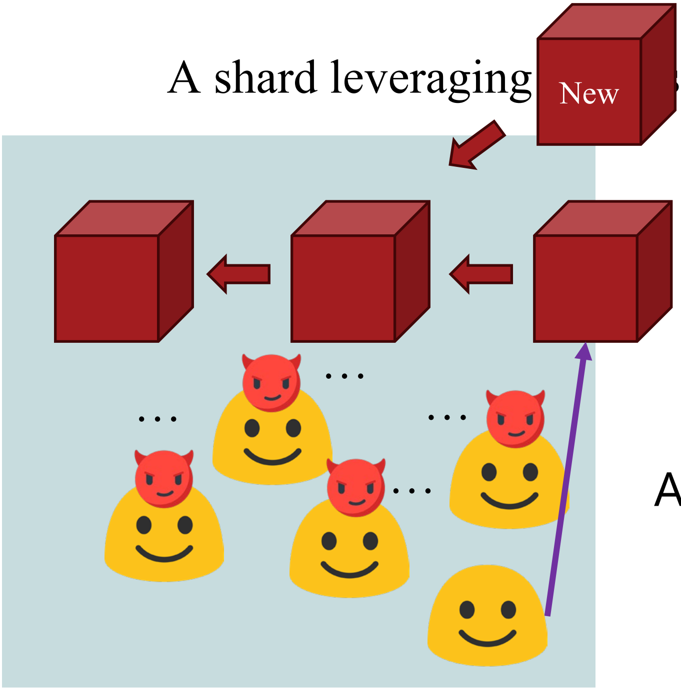
Other nodes remain silent.

An honest node voted to accept this block.

S=80%

L<20%

A shard leveraging [...]s and safety



After a timeout, a new block is proposed.

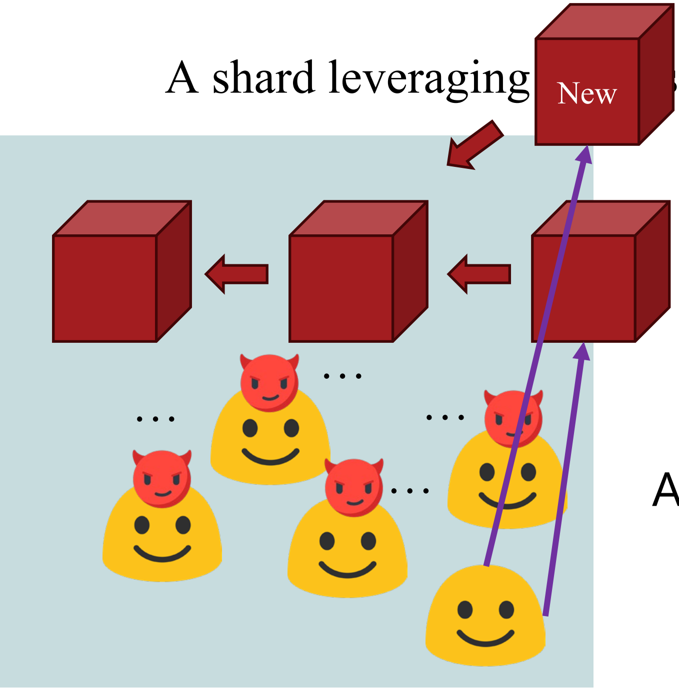
Other nodes remain silent.

An honest node voted to accept this block.

S=80%

L<20%

A shard leveraging s and safety



After a timeout, a new block is proposed.

Vote for the new block.

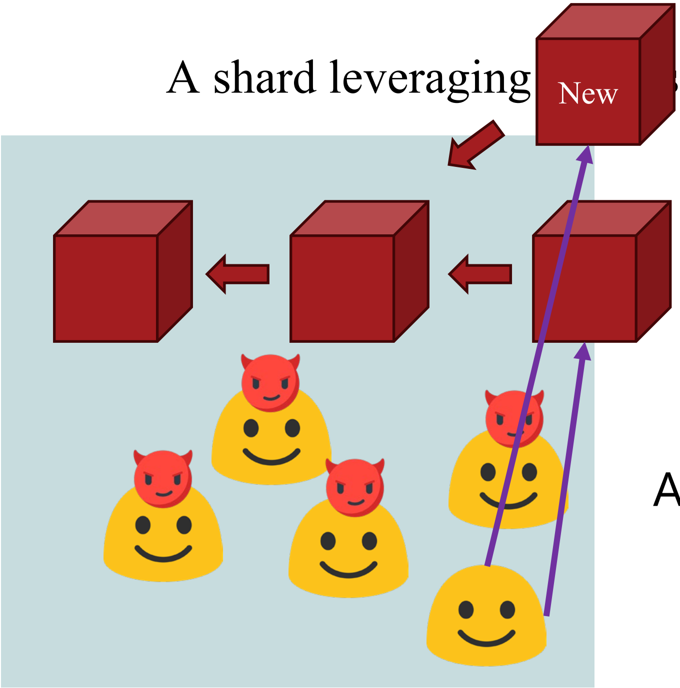
Other nodes remain silent.

An honest node voted to accept this block.

S=80%

L<20%

A shard leveraging ... s and safety



After a timeout, a new block is proposed.

Vote for the new block.

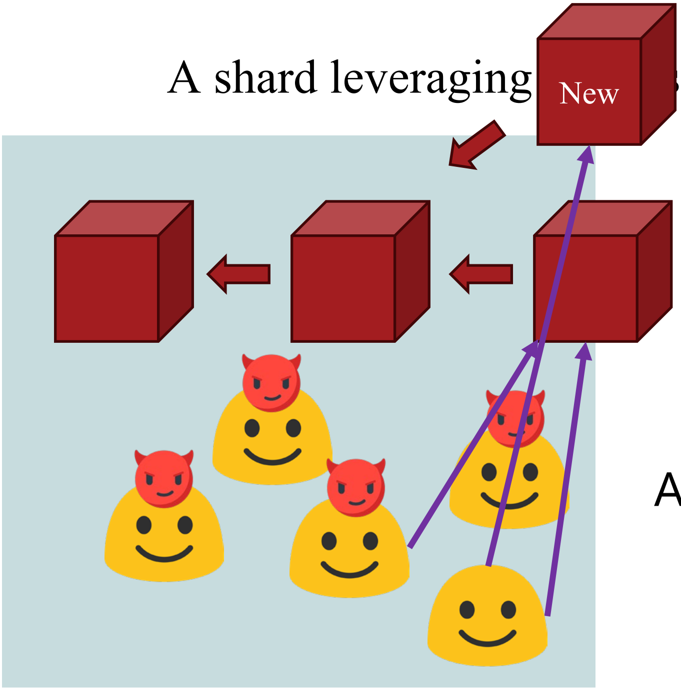
Other nodes remain silent.

An honest node voted to accept this block.

S=80%

L<20%

A shard leveraging … s and safety



After a timeout, a new block is proposed.

Vote for the new block.

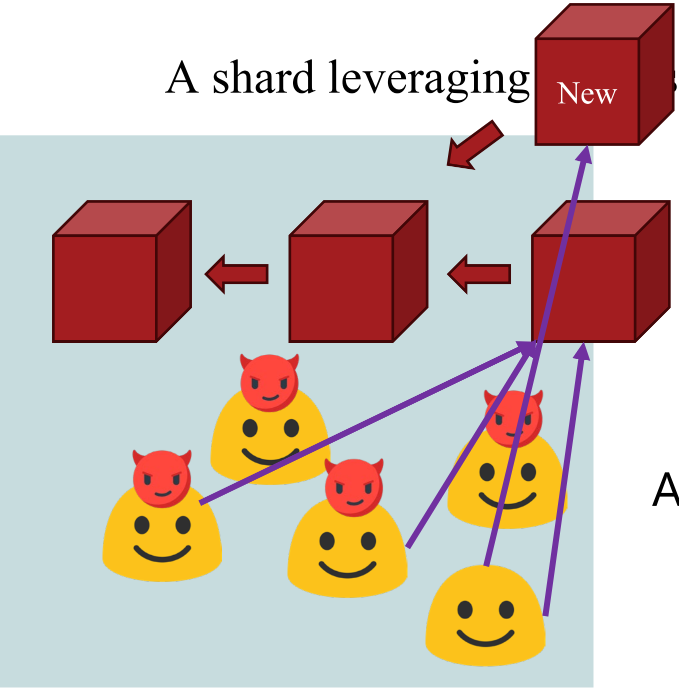
Other nodes remain silent.

An honest node voted to accept this block.

S=80%

L<20%

A shard leveraging s and safety



After a timeout, a new block is proposed.

Vote for the new block.

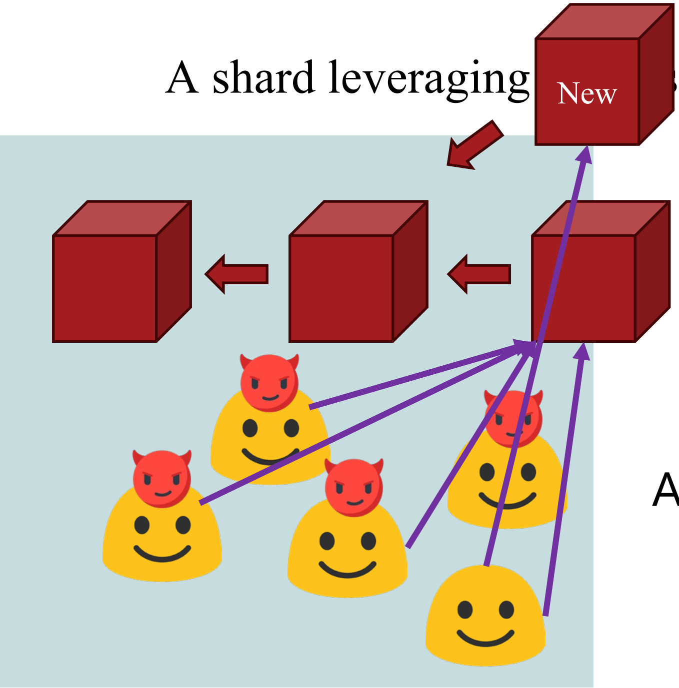
Other nodes remain silent.

An honest node voted to accept this block.

S=80%

L<20%

A shard leveraging s and safety



After a timeout, a new block is proposed.

Vote for the new block.

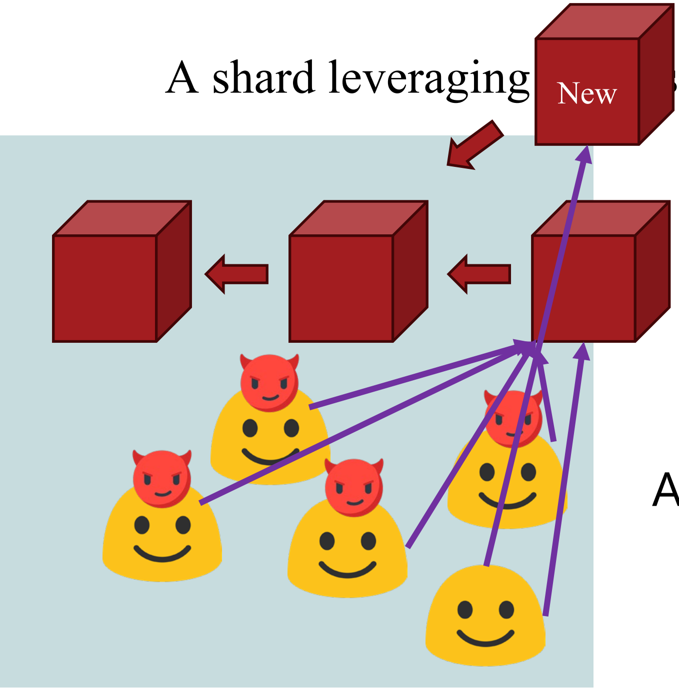
Other nodes remain silent.

An honest node voted to accept this block.

S=80%

L<20%

A shard leveraging s and safety



After a timeout, a new block is proposed.

Vote for the new block.

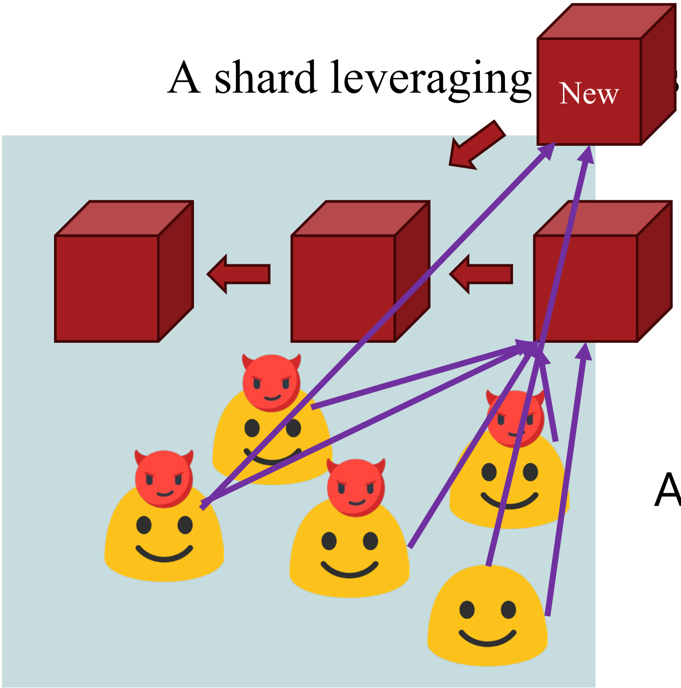
Other nodes remain silent.

An honest node voted to accept this block.

S=80%

L<20%

A shard leveraging [...]s and safety



After a timeout, a new block is proposed.

Vote for the new block.

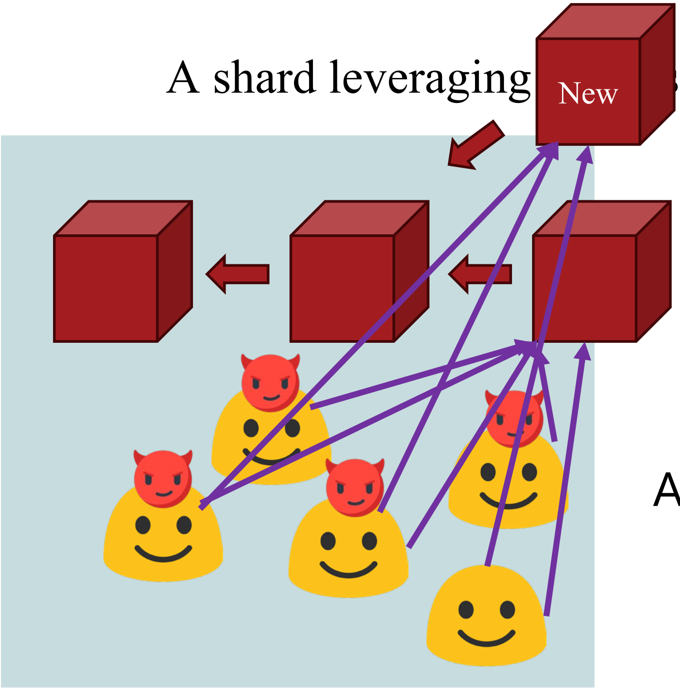
Other nodes remain silent.

An honest node voted to accept this block.

S=80%

L<20%

A shard leveraging s and safety



After a timeout, a new block is proposed.

Vote for the new block.

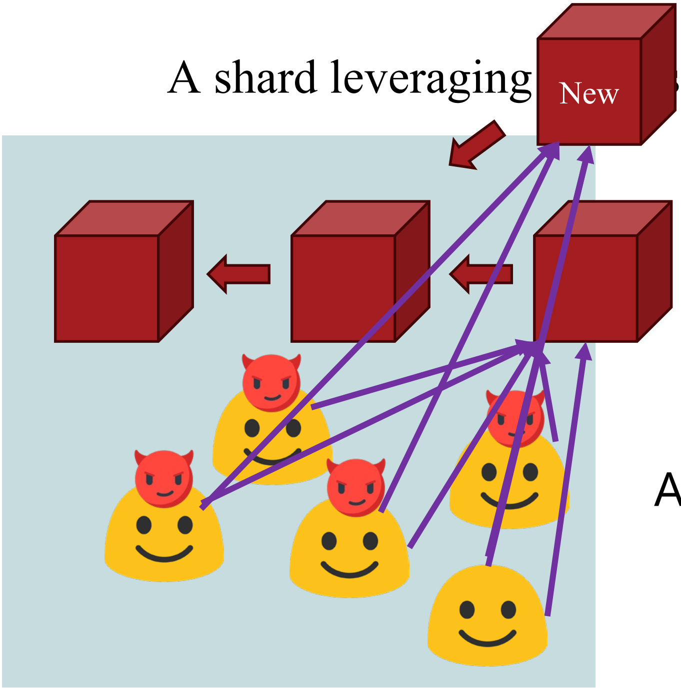
Other nodes remain silent.

An honest node voted to accept this block.

S=80%

L<20%

A shard leveraging s and safety



After a timeout, a new block is proposed.

Vote for the new block.

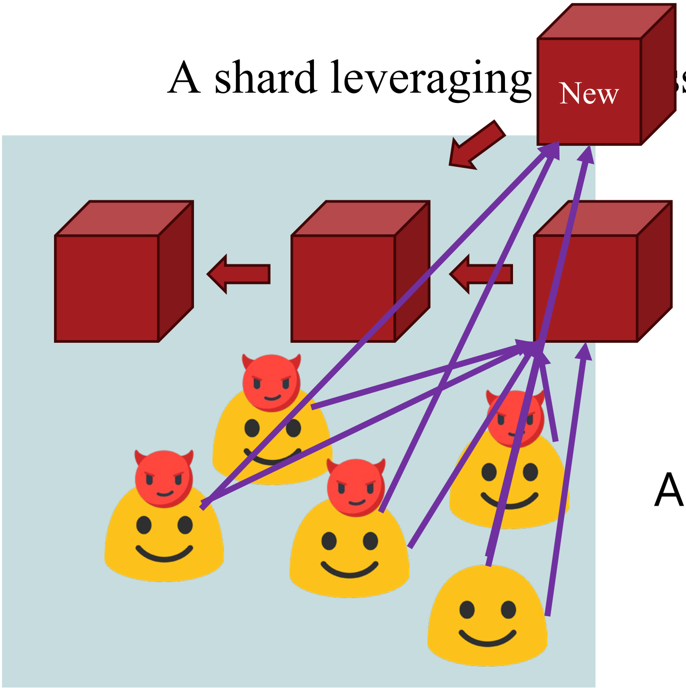
Other nodes remain silent.

An honest node voted to accept this block.

S=80%

L<20%

A shard leveraging ...s and safety



After a timeout, a new block is proposed.

Vote for the new block.

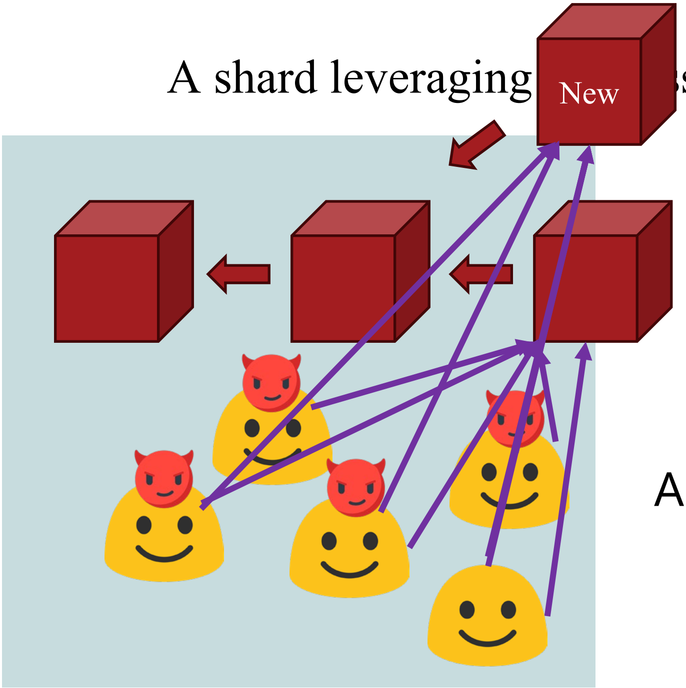
Other nodes remain silent.

An honest node voted to accept this block.

S=80%

L<20%

A shard leveraging s and safety



After a timeout, a new block is proposed.

Vote for the new block.

Other nodes remain silent.

An honest node voted to accept this block.

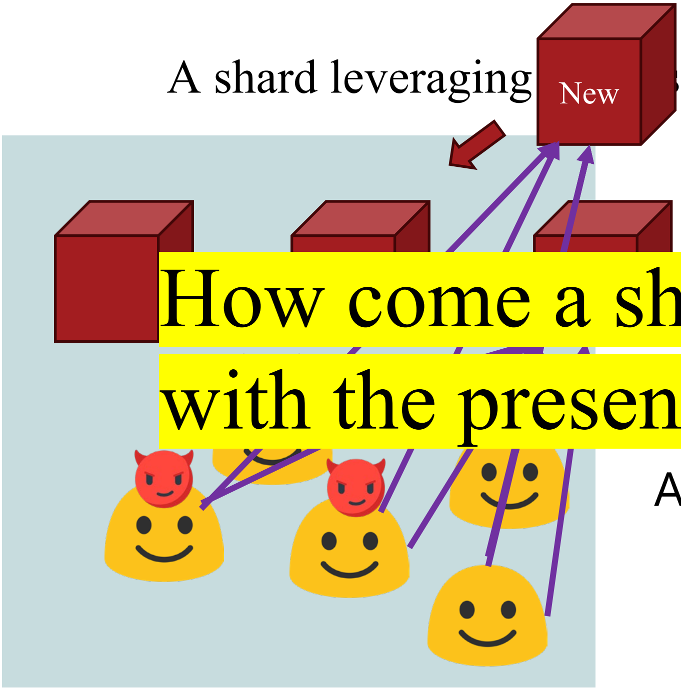
S=80%

L<20%

Can not tell if a vote was casted on time.
Cannot avoid equivocation.

Unless the decision reached is confirmed globally.

A shard leveraging s and safety



Vote for the new block

After a timeout, a new block is proposed.

How come a shard reaching an honest consensus independently with the presence of the majority adversarial nodes?

An honest node voted to accept this block.

S=80%

L<20%

Can not tell if a vote was casted on time.
Cannot avoid equivocation.

Unless the decision reached is confirmed globally.

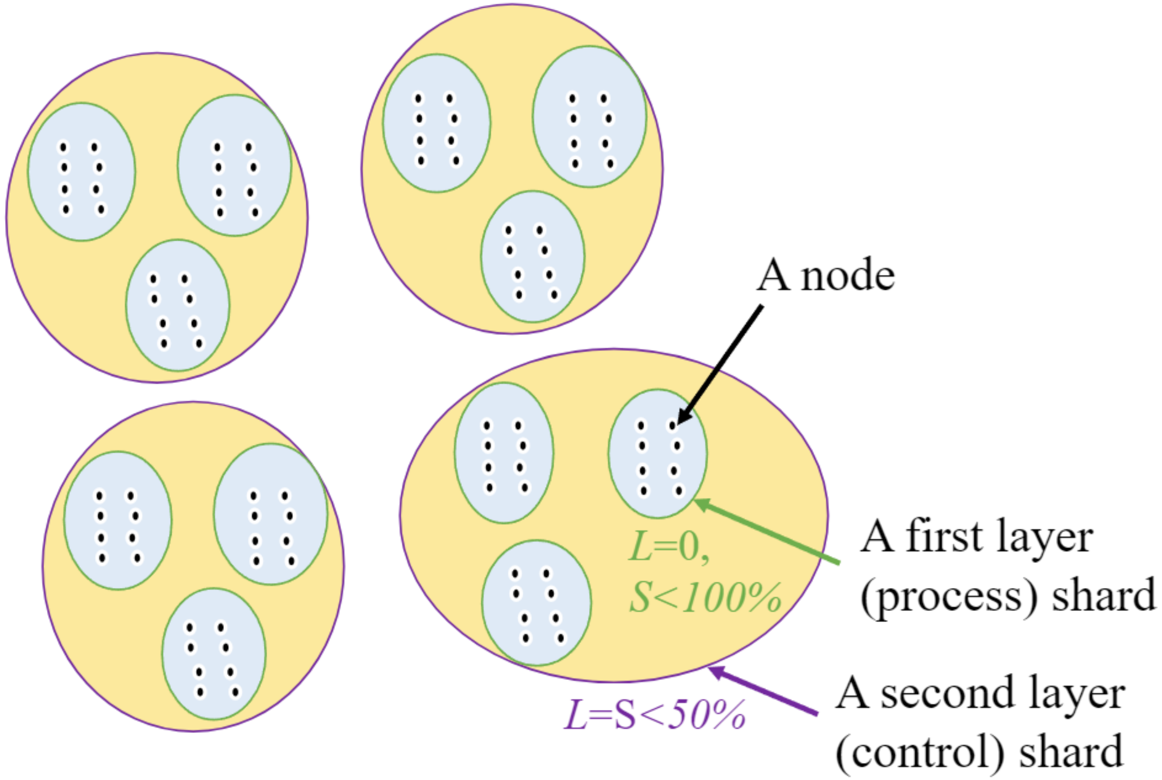
Reticulum: A Two-Layer Blockchain Sharding Protocol Leveraging Safety and Liveness

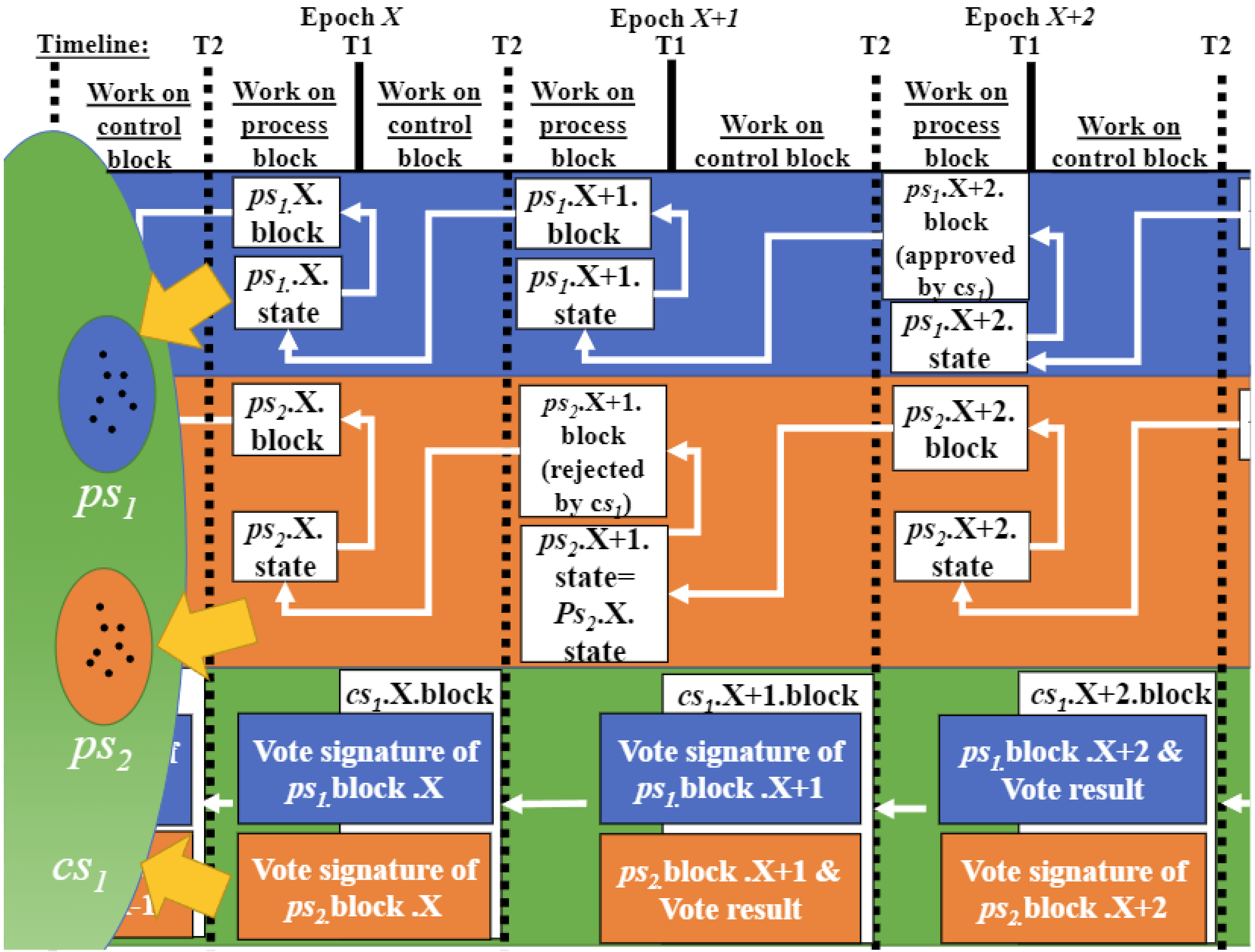
Nodes are in a process shard and the corresponding control shard.

The votes within a process shard are Byzantine broadcast to all nodes in the control shard.

Only being confirmed in the control shard, a new blockchain epoch starts in the process shard.

If a process shard is dead, its work is carried on by the control shard.





T1: Timebound for the process block.
T2: Timebound for the control block.

T2 is adjusted according to how many process blocks requiring intervention.

Responsibility and punishment can be built!

τ liveness guarantees

The honest nodes are remain active at all times.
Other nodes can absent from voting in the process shard once in every τ rounds.

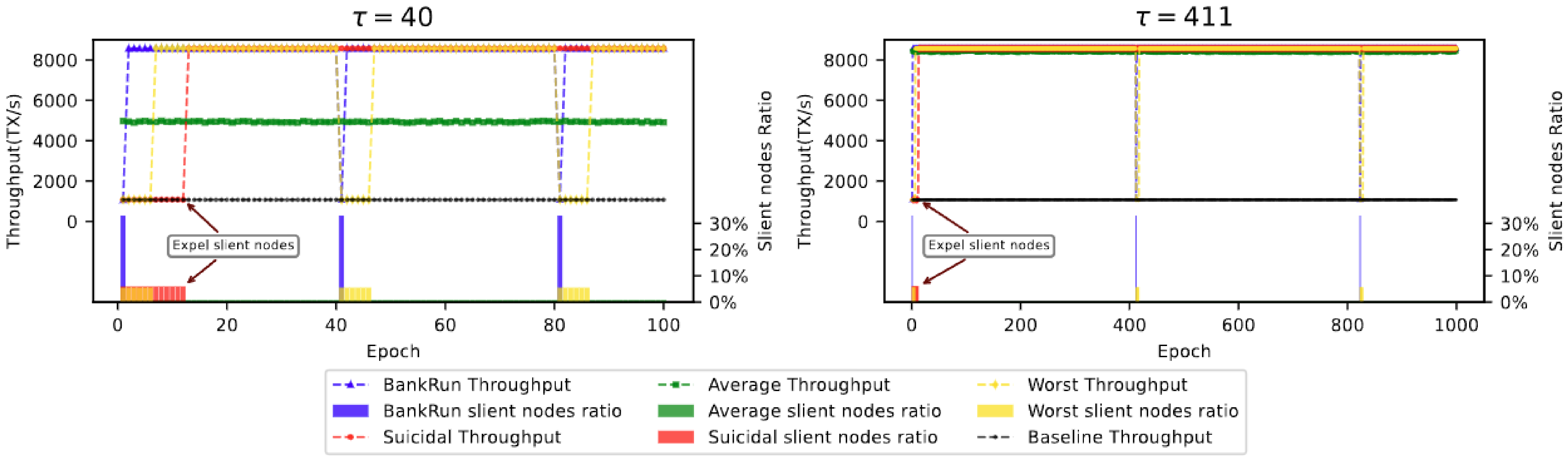
BankRun: all adversarial nodes do not vote for the process blocks at a single epoch.
BankRun can only occur once in every τ epochs.

Average: each adversarial node does not vote once in a random epoch in every τ epochs.

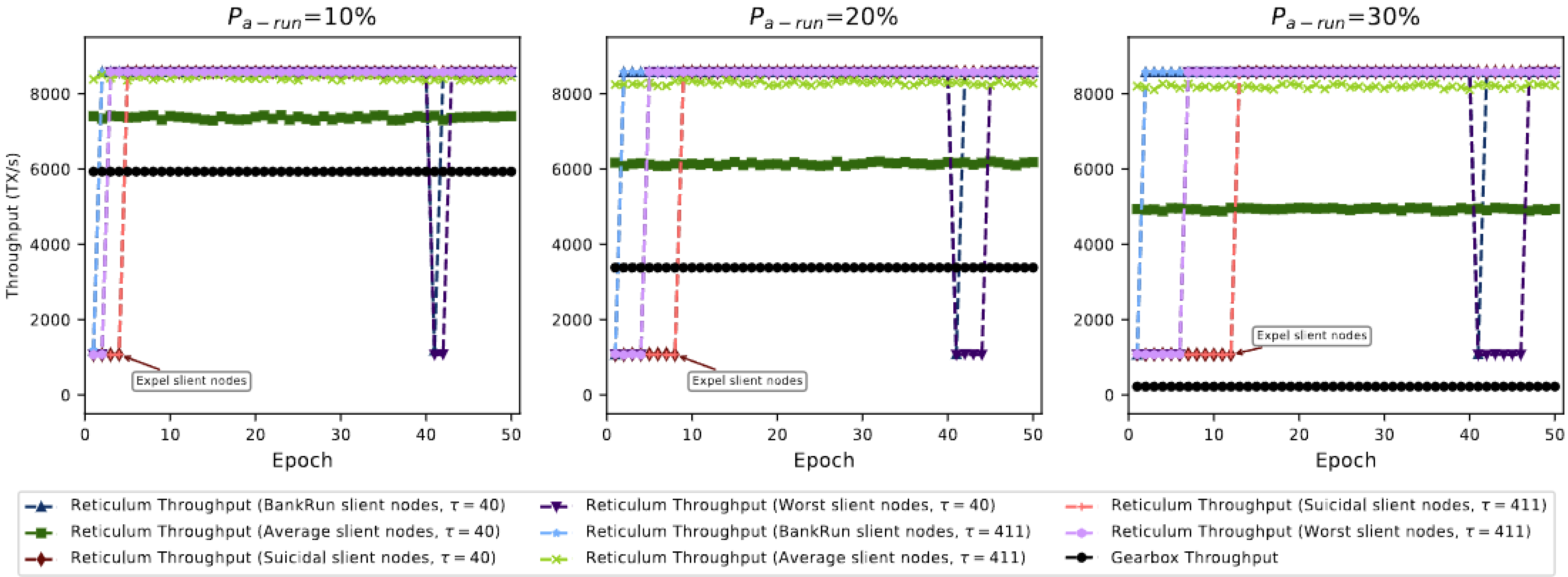
Worst: only one adversarial node refuses to vote at each process shard in every epoch. The adversary can stop a process shard for $i < \tau$ epochs in every τ epochs where i is the number of adversarial nodes inside this shard.

Suicidal: is based on the worst strategy but all adversarial nodes vote at most τ -2 epochs in every τ epoch, and be expelled at the second time when they remain silent in voting.

Experiment



$\tau = 40$
$\tau = 411$
Throughput(TX/s)
8000
6000
4000
2000
0
Expel slient nodes
30%
20%
10%
0%
Slient nodes Ratio
0
20
40
60
80
100
200
400
600
800
1000
Epoch
BankRun Throughput
BankRun slient nodes ratio
Suicidal Throughput
Average Throughput
Average slient nodes ratio
Suicidal slient nodes ratio
Worst Throughput
Worst slient nodes ratio
Baseline Throughput


P_{a-run}=10%
P_{a-run}=20%
P_{a-run}=30%
Throughput (TX/s)
Epoch
Expel slient nodes
Reticulum Throughput (BankRun slient nodes, τ = 40)
Reticulum Throughput (Average slient nodes, τ = 40)
Reticulum Throughput (Suicidal slient nodes, τ = 40)
Reticulum Throughput (Worst slient nodes, τ = 40)
Reticulum Throughput (BankRun slient nodes, τ = 411)
Reticulum Throughput (Average slient nodes, τ = 411)
Reticulum Throughput (Suicidal slient nodes, τ = 411)
Reticulum Throughput (Worst slient nodes, τ = 411)
Gearbox Throughput

Thank you ☺