SkyChain: A Deep Reinforcement Learning-Empowered Dynamic Blockchain Sharding System

Jianting Zhang, Zicong Hong, Xiaoyu Qiu, Yufeng Zhan, Song Guo, Wuhui Chen

Aug 18, 2020

Talk Outline

Part I **Background**

Part II **Motivation**

Part III **SkyChain**

Part IV **Evaluation**

I

Background

1 The bottleneck of scalability in blockchain

Low TPS

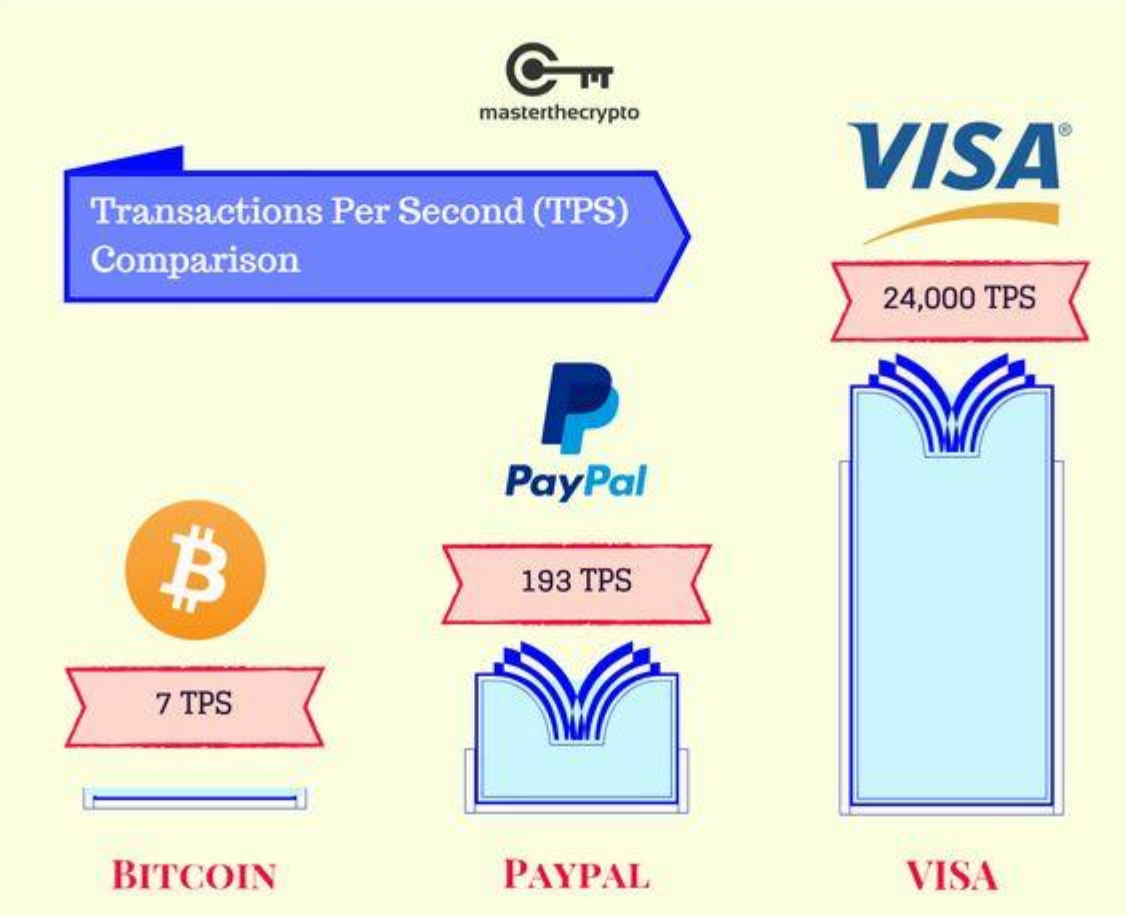
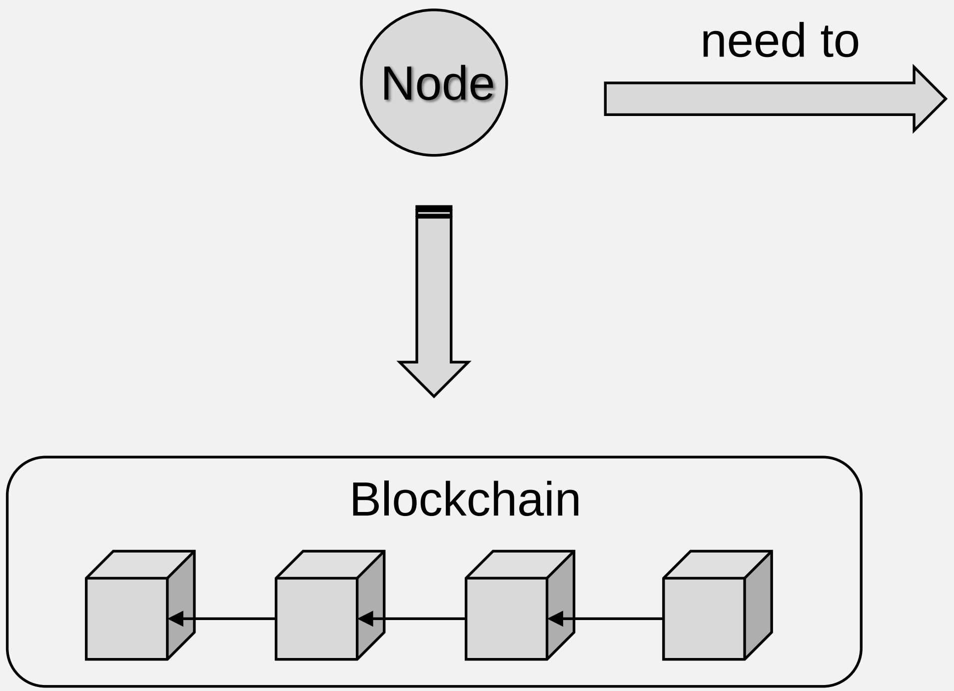


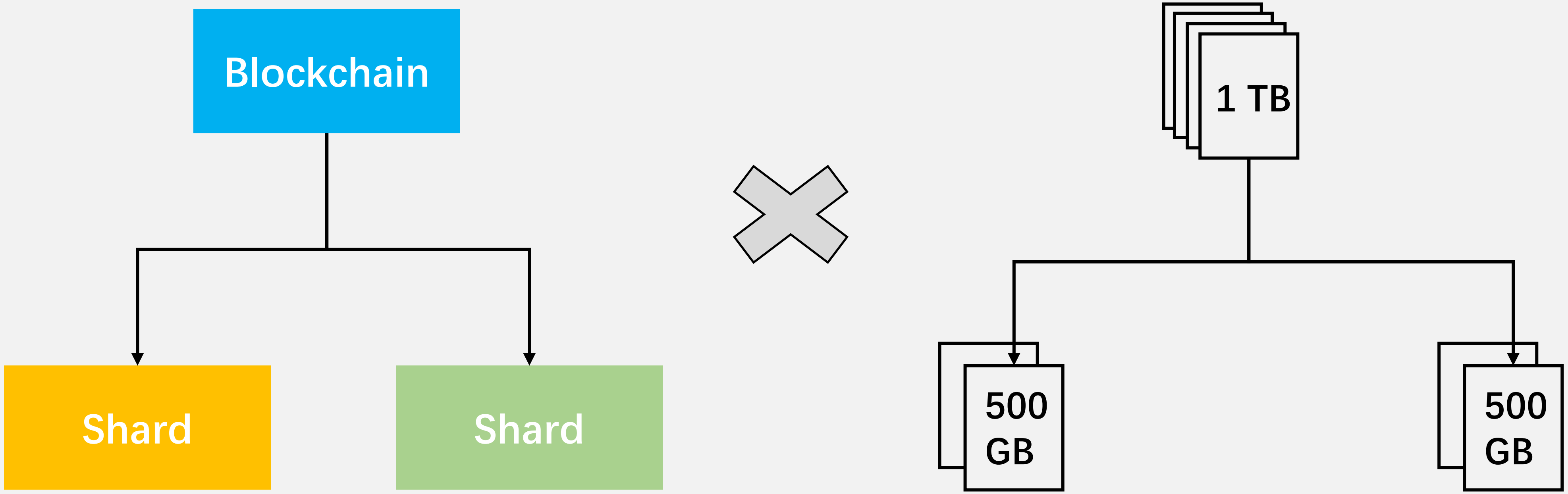
Fig.1:TPS Comparison

2 Why is it not scalable in blockchain?



- Communicate with all nodes — **Network I/O** Limit
- Validate all transactions — **CPU** Limit
- Store the whole ledger — **Disk** Limit

3 Blockchain sharding overview



Nodes sharding

Reducing the overhead of communication and computing

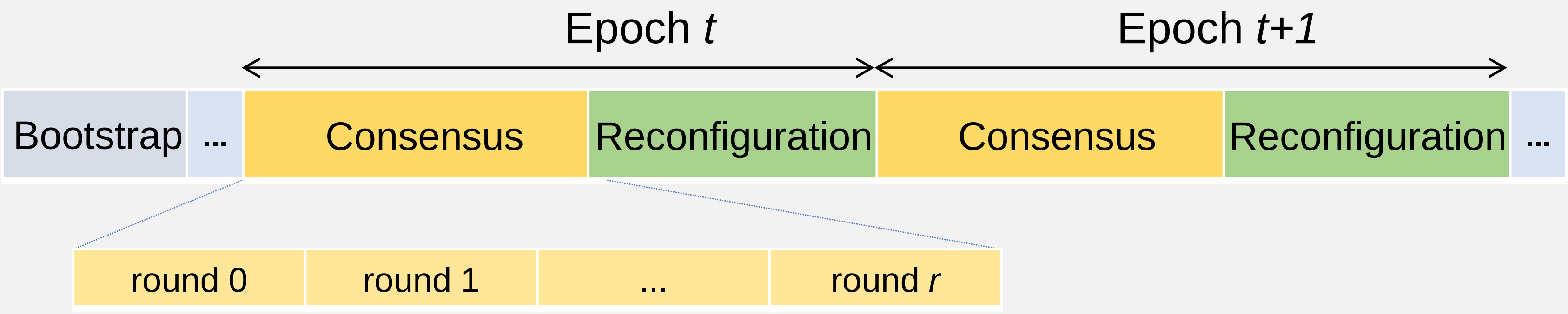
Storage sharding

Reducing the overhead of storage

II

Motivation

1 Privious sharding systems



a) Operating in **fixed** interval, called *epoch*. Each epoch includes:

- Consensus period: consisting of multiple rounds, in which a new block will be created
- Reconfiguration period: reshuffling shards to resist attacks

b) The number of shards is **fixed**

c) The size of block is **fixed**

Unadaptable for dynamic environment in blockchain

2 Dynamics of blockchain

- **New nodes joininig** ⟹ Enlarging shard size ⟹ Increasing communication overhead & Decreasing **throughput**
- **Old nodes leaving** ⟹ Decreasing shard size ⟹ Lowering the **security**
- **The continuous corrupting** ⟹ Increasing the rate of malicious nodes in a shard ⟹ Breaking the **security**

III

SkyChain
A dynamic blockchain sharding system

1 SkyChain design

Idea:

- Adjusting **the sharding policy** based on the dynamic environment

Goal

- Achieving a long-term balance between **performance** and **security**

Challenges

- How to make a suitable sharding policy
- How to efficiently merge or split the ledgers when shard number change

2 The sharding policy

a) The re-sharding interval

- Determining the duration of consensus period

b) The number of shards

- Determining how many shards can process transactions in parallel

c) The block size

- Determining how many transactions can be validated in each consensus round

a) The re-sharding interval

- Less frequent re-sharding can save more time for the consensus process, but it increases the risk of the sharding system.

- More frequent re-sharding can decrease the risk of the sharding system, but it intervenes the consensus process because validators stop processing consensus and suffers from extra cost for the communication and computation of re-sharding operation.

b) The number of shards

m: the size of shard

- Less shards can maintain the size of shards and increase the resiliency to malicious attacks.
- More shards can increase the parallelism for processing transactions and reduce communication overhead. However, the probability of forming an unsafe shard will be high if the shard size is small.

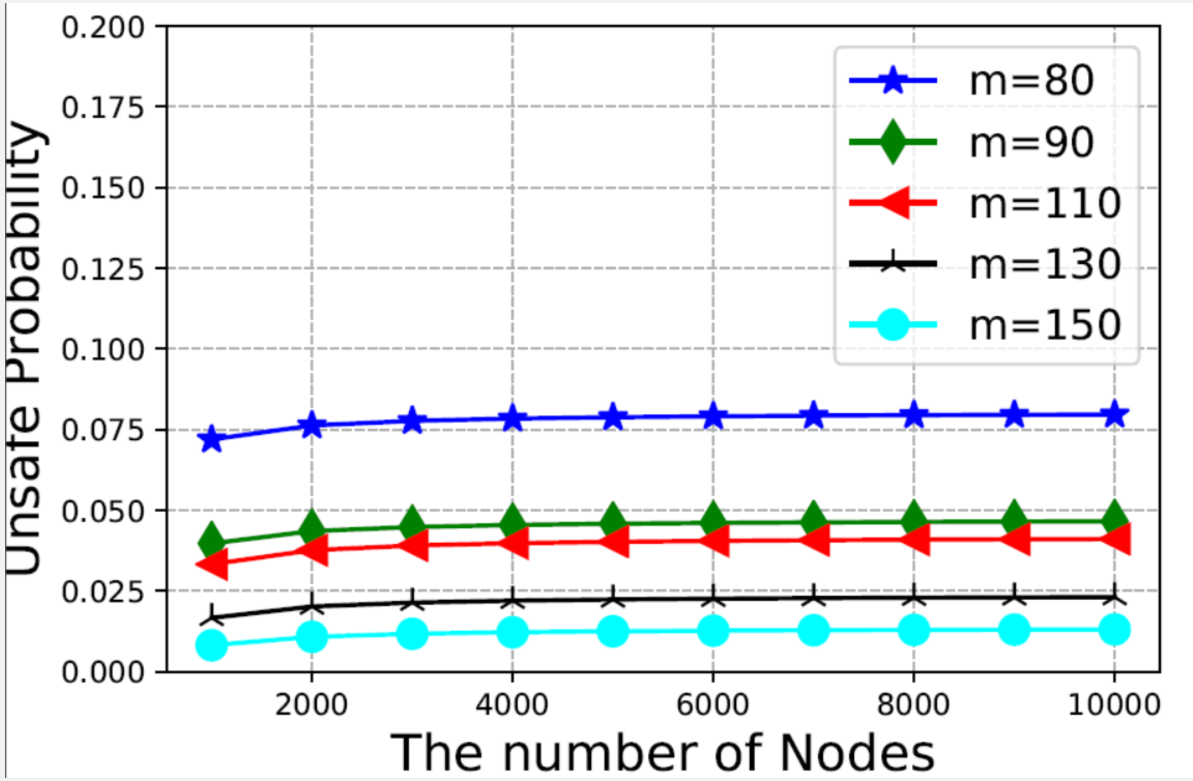


Fig.2:Unsafe probability with shard size and system size

c) The block size

m: the size of shard

- A bigger block can pack more transactions, thereby increasing the throughput.

- A bigger block also incurs a significant communication overhead and increases the latency of each consensus round when the shard size is big.

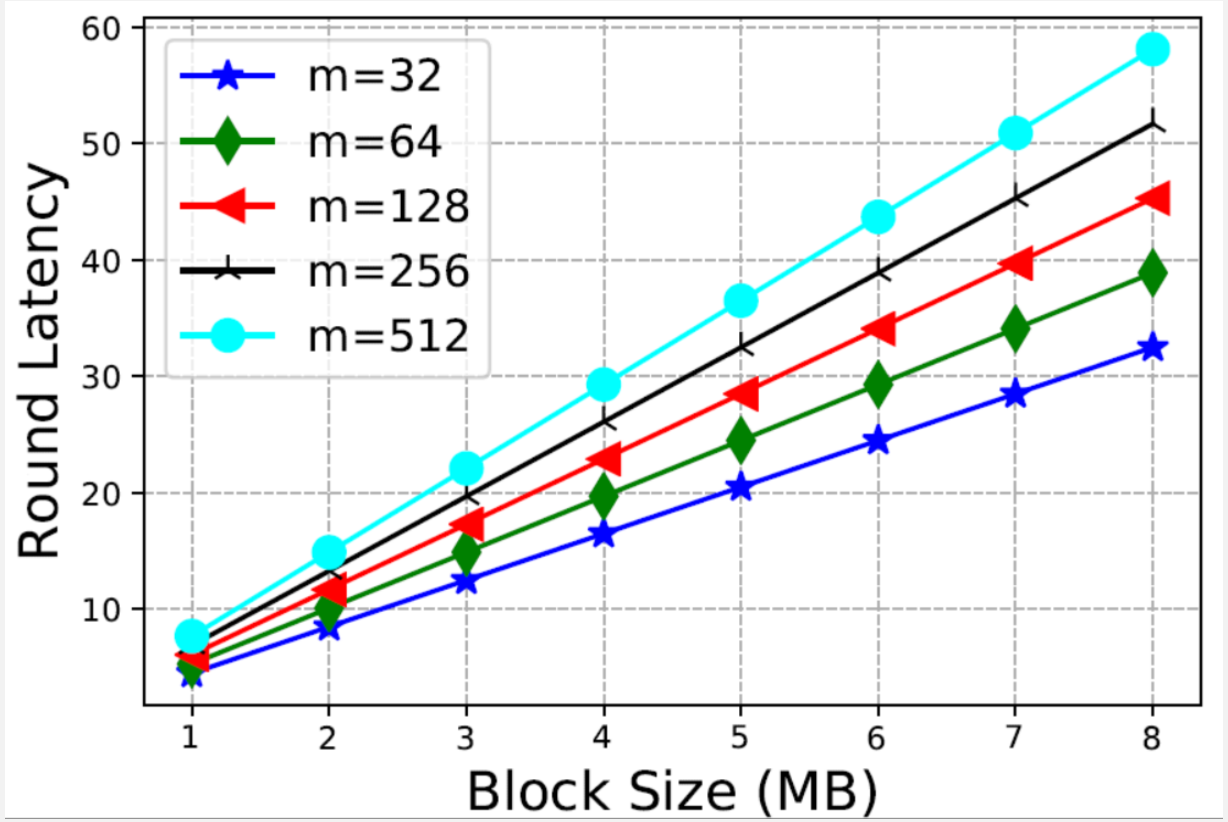


Fig.3:Round latency with block size

DRL-based dynamic sharding model

for making a suitable sharding policy

3 DRL-based sharding model

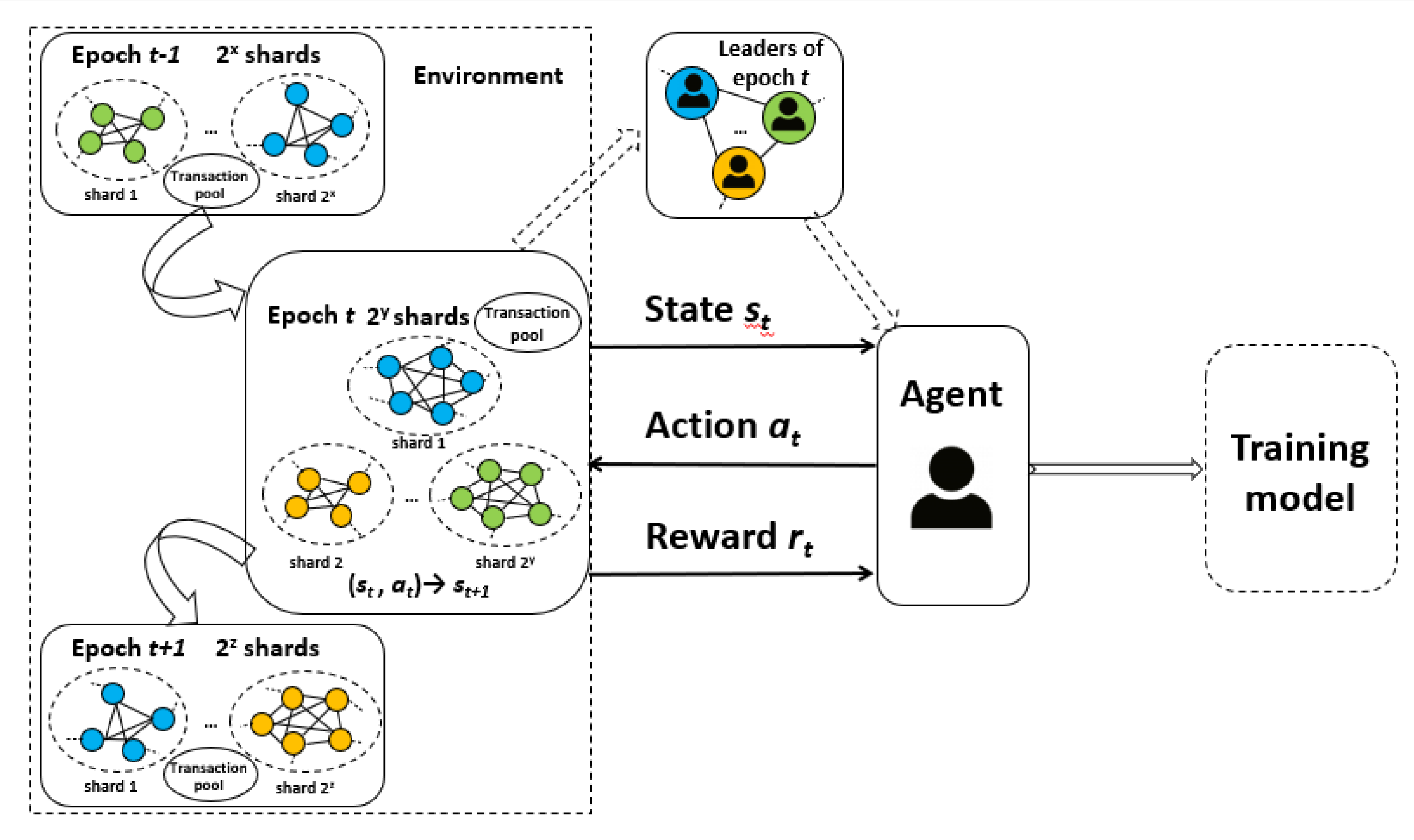
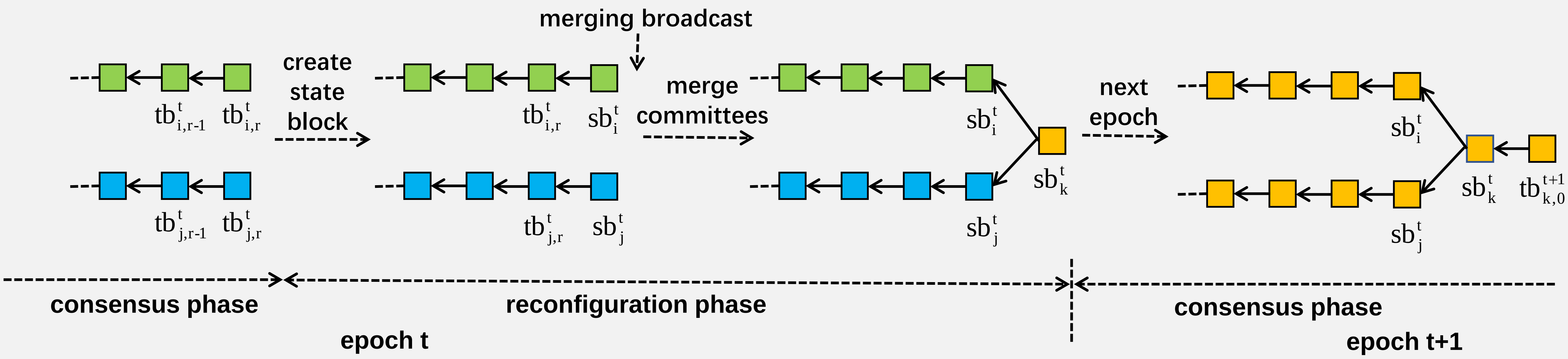


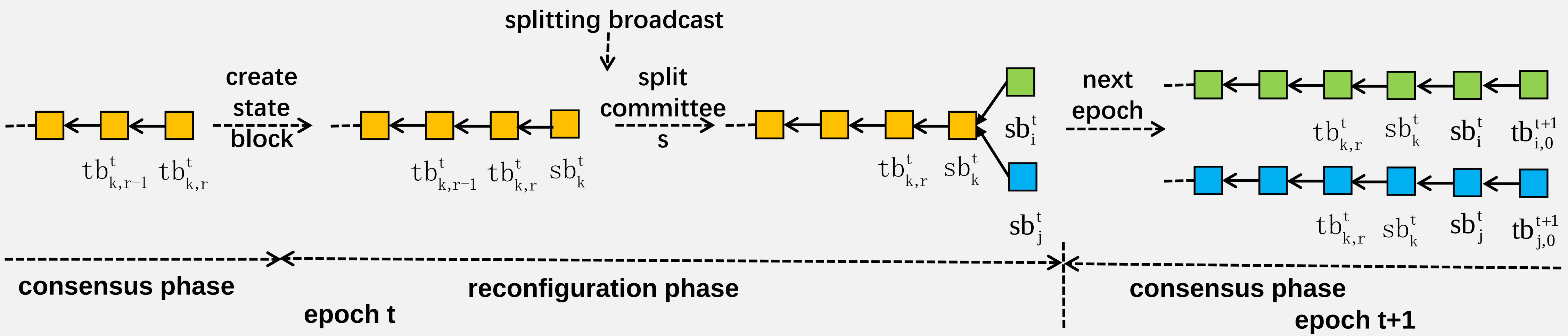
Fig.4:System model

Adaptive ledger protocol for merging or splitting the ledgers

4.1 Ledgers merging



4.2 Ledgers splitting



IV

Evaluation

Experimental Setting

The designs compared sharding algorithms

- The sharding policy with fixed re-sharding interval
- The sharding policy with fixed number of shards
- The sharding policy with fixed block size

Throughput vs Transaction size

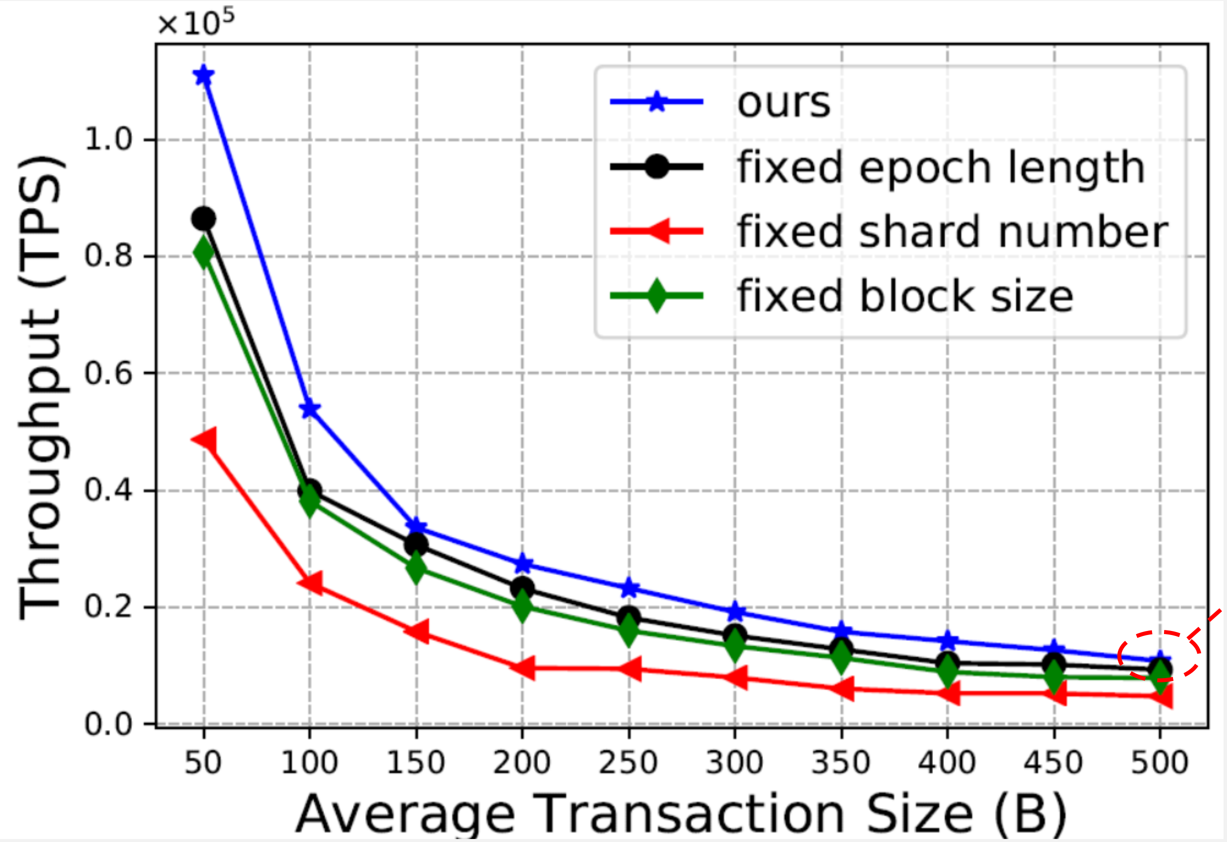
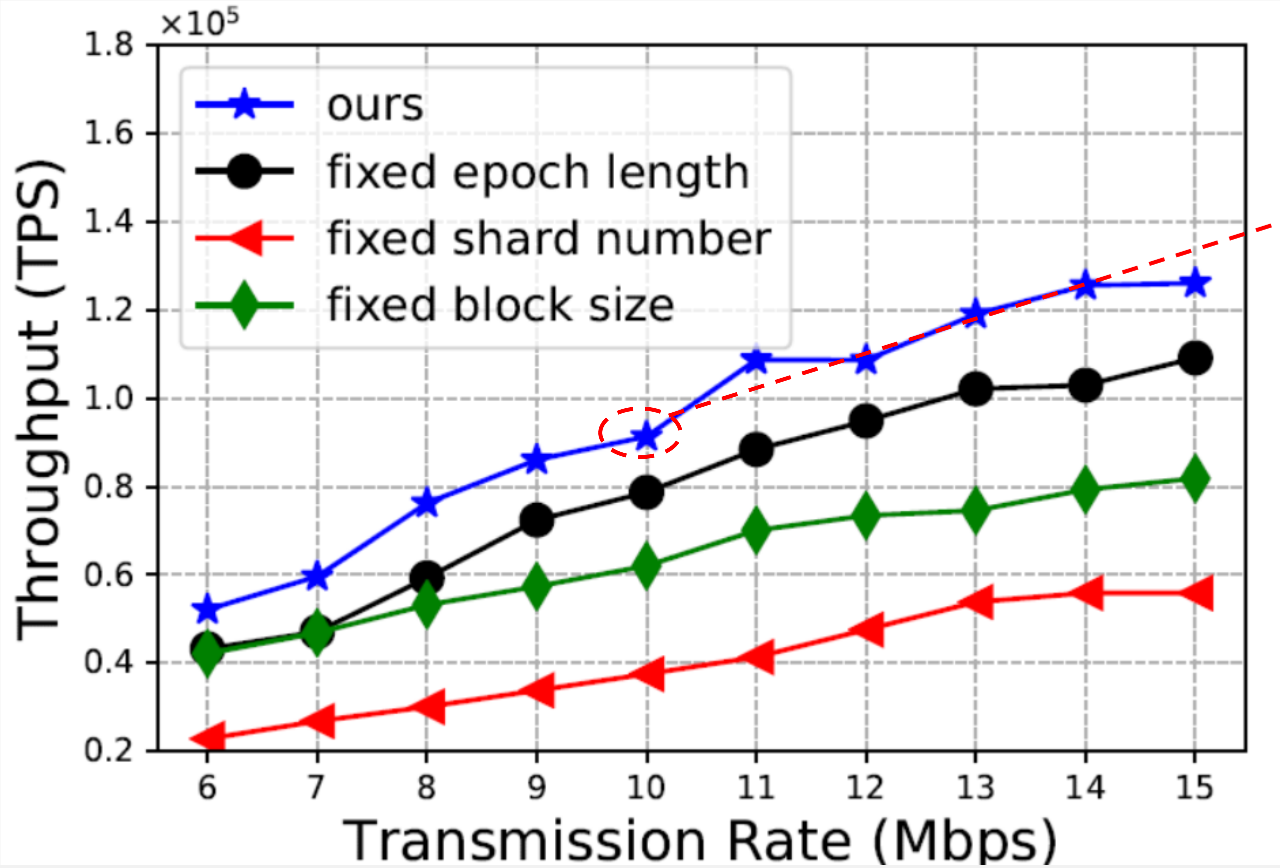


Fig.5:Throughput with average transaction size

10,000 TPS

Transmission Rate: 10 MB/s

Throughput vs Transmission Rate



90,000 TPS

Transaction size: 64 bytes

Fig.6:Throughput with transmission rate

Thank You!