Monoxide: Scale out Blockchains with Asynchronous Consensus Zones

Jiaping Wang, Hao Wang

Presenter

Tzu-Hsin Yang, Yi-Hsiang Chiu, Yu-Hsuan Lin, Pin-Ting Liu, Yu-Cheng Hwang

Outline

- Introduction
- Main Contribution
- System Design
- Chu-Ko-Nu
- Discussion

Introduction

- Obstacles of blockchain:
    - Low throughput hindered the scalability and usability of blockchain systems for increasing numbers of users and transactions
    - The requirement for every node to duplicate the communication, storage, and state representation of the entire network

- Partitioning workload into multiple zones can be a solution such that the consensus happened individually within each zone

UCDAVIS
COMPUTER SCIENCE

Challenges

- Correctness and Robustness
    - A transaction might involve multiple parties in different zones

- Efficiency
    - Efficient handling of such cases is the key to the throughput scalability and the performance

- Security
    - When the mining power is distributed to different zones, an attacker can gather the mining power toward a single zone and may easily exceed the 51% threshold within that zone

Concepts

- Asynchronous Consensus Zones
- Cross-Zone Atomicity
- Effective Mining Power Amplification

Contribution

- Divide workloads of communication, computation, storage, and memory for state representation into independent and parallel zones.

- **(Efficiency)** Eventual atomicity: efficiently handle cross-zone transactions, ensuring correctness and robustness.

- **(Security)** Chu-ko-nu mining: a novel PoW scheme, preventing lowering the attack bar when the mining power is dispersed into multiple zones.

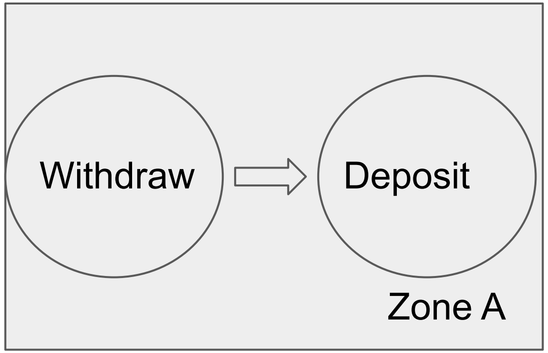
Background

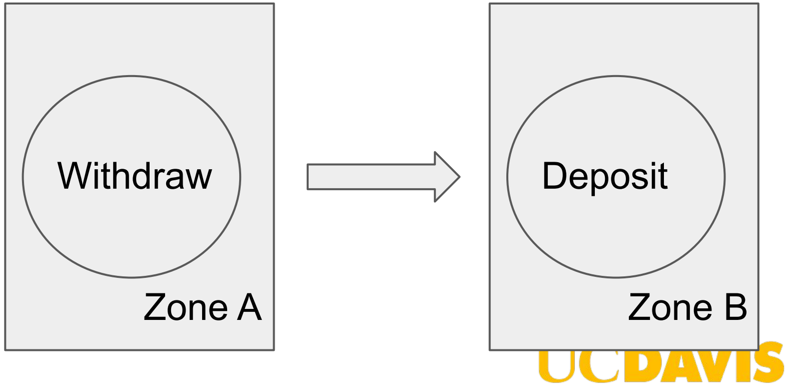
- Proof-of-Work (PoW)
- Account/Balance

System Design

System Design - Transactions in High Level View

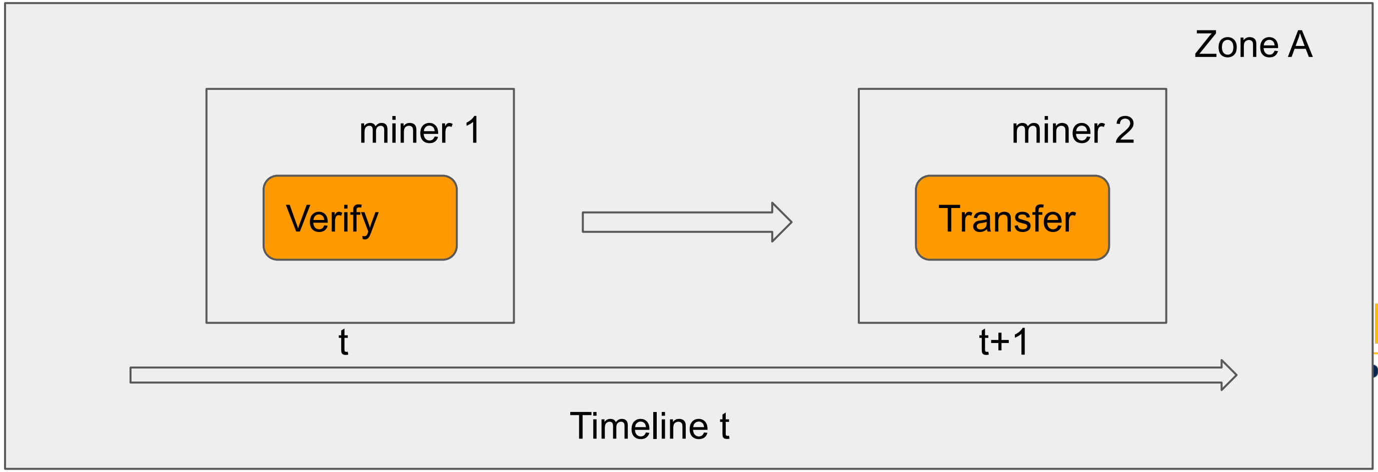
- Transactions type
  - Transactions happen in the single zone
  - Relay Transactions





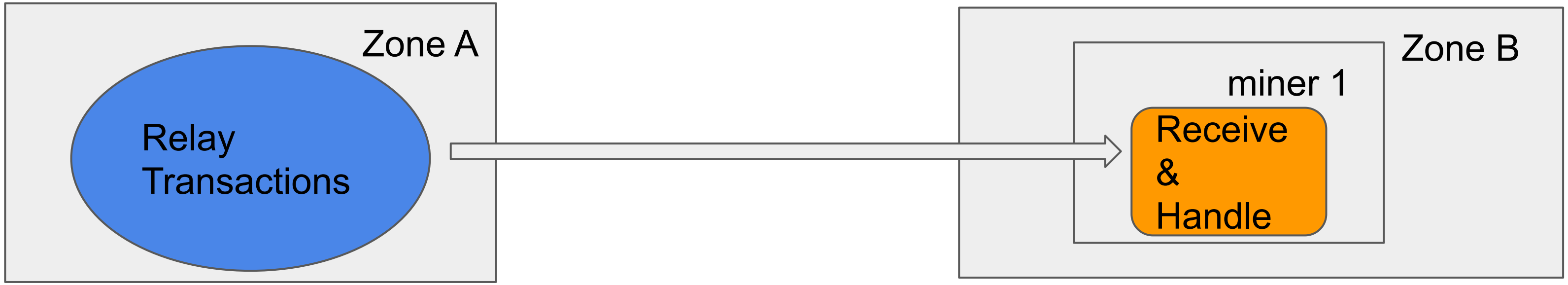
System Design - Relay Transactions

- Two users in different zones (e.g. zone A & zone B)
- Withdrawal operation in **zone A** picked up by a miner
  - In time t, miner verifies account balance
  - If valid, the t+1 block will take care of the transaction
  - If invalid, discard the transaction and record in Merkle tree



System Design - Relay Transactions

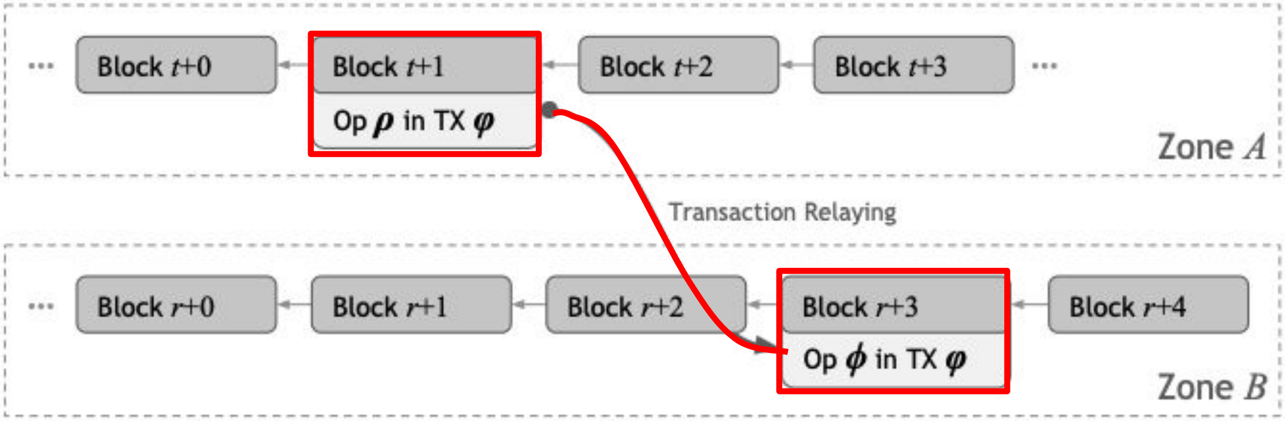
- Relay Transactions create in zone A
- Transferring to zone B
- Picking up by a miner in zone B



UCDAVIS COMPUTER SCIENCE

System Design - Relay Transactions

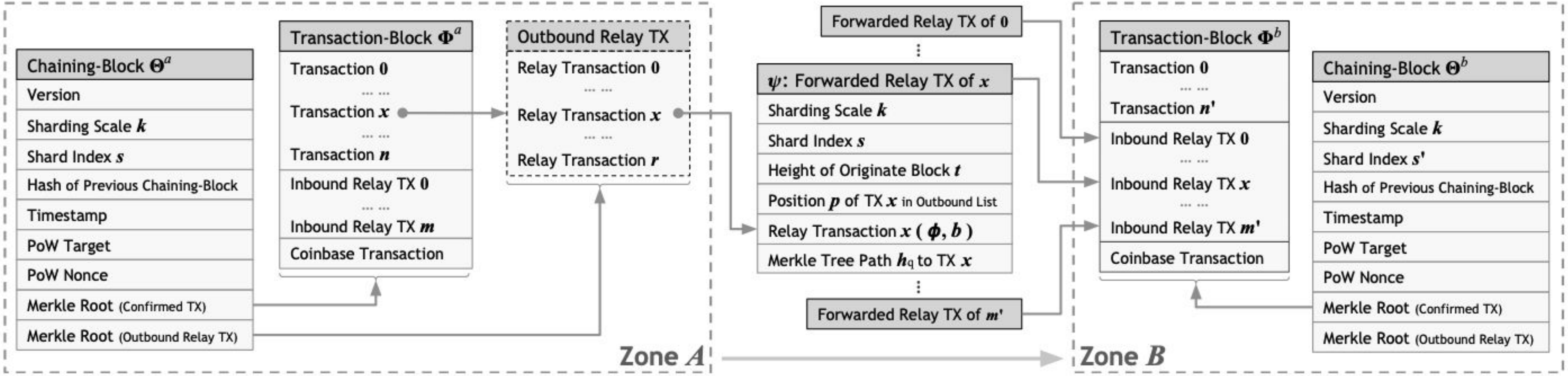
- Relay transactions



UC DAVIS
COMPUTER SCIENCE

Efficient Cross-Zone Atomicity

- Overlook



Chaining-Block Θ^a
- Version
- Sharding Scale k
- Shard Index s
- Hash of Previous Chaining-Block
- Timestamp
- PoW Target
- PoW Nonce
- Merkle Root (Confirmed TX)
- Merkle Root (Outbound Relay TX)

Transaction-Block Φ^a
- Transaction 0
- … …
- Transaction x
- … …
- Transaction n
- Inbound Relay TX 0
- … …
- Inbound Relay TX m
- Coinbase Transaction

Outbound Relay TX
- Relay Transaction 0
- … …
- Relay Transaction x
- … …
- Relay Transaction r

Zone A

Forwarded Relay TX of 0

⋮

ψ: Forwarded Relay TX of x
- Sharding Scale k
- Shard Index s
- Height of Originate Block t
- Position p of TX x in Outbound List
- Relay Transaction x (ϕ, b)
- Merkle Tree Path h_q to TX x

⋮

Forwarded Relay TX of m'

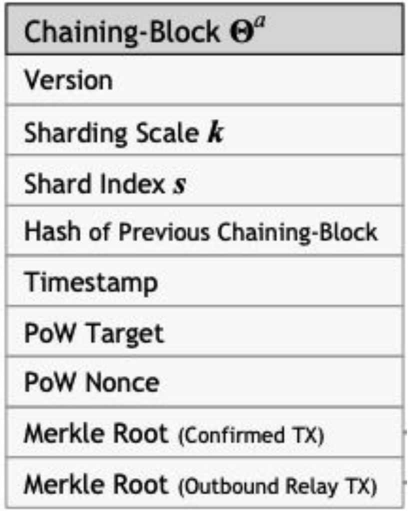
Transaction-Block Φ^b
- Transaction 0
- … …
- Transaction n'
- Inbound Relay TX 0
- … …
- Inbound Relay TX x
- … …
- Inbound Relay TX m'
- Coinbase Transaction

Chaining-Block Θ^b
- Version
- Sharding Scale k
- Shard Index s'
- Hash of Previous Chaining-Block
- Timestamp
- PoW Target
- PoW Nonce
- Merkle Root (Confirmed TX)
- Merkle Root (Outbound Relay TX)

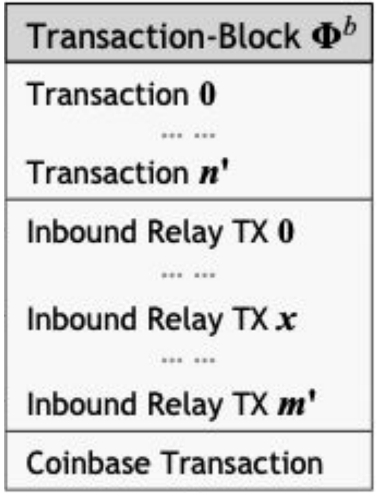
Zone B

Chaining-Block & Transaction-Block

- There are two parts in one zone
    - Chaining-Block
    - Transaction-Block



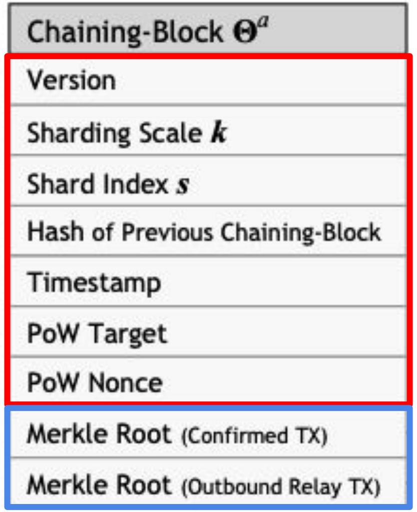
Chaining-Block Θ^a
Version
Sharding Scale k
Shard Index s
Hash of Previous Chaining-Block
Timestamp
PoW Target
PoW Nonce
Merkle Root (Confirmed TX)
Merkle Root (Outbound Relay TX)



Transaction-Block Φ^b
Transaction 0 … … Transaction n'
Inbound Relay TX 0 … … Inbound Relay TX x … … Inbound Relay TX m'
Coinbase Transaction

Chaining-Block

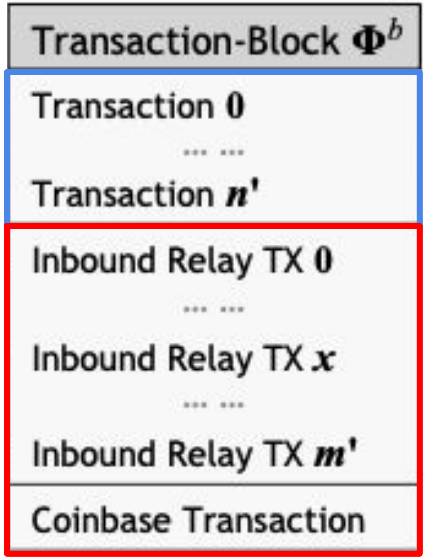
- In chaining-block
  - Block metadata
  - Transaction information of this particular zone



Chaining-Block Θ^a
Version
Sharding Scale k
Shard Index s
Hash of Previous Chaining-Block
Timestamp
PoW Target
PoW Nonce
Merkle Root (Confirmed TX)
Merkle Root (Outbound Relay TX)

Transaction-Block

- Confirmed transactions
- Transaction lists



Transaction-Block $\mathbf{\Phi}^b$

Transaction $\mathbf{0}$

... ...

Transaction $\boldsymbol{n'}$

Inbound Relay TX $\mathbf{0}$

... ...

Inbound Relay TX $\boldsymbol{x}$

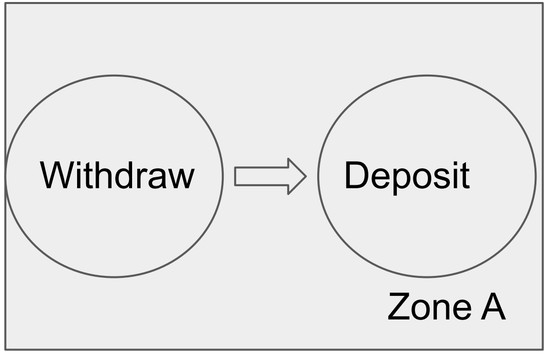
... ...

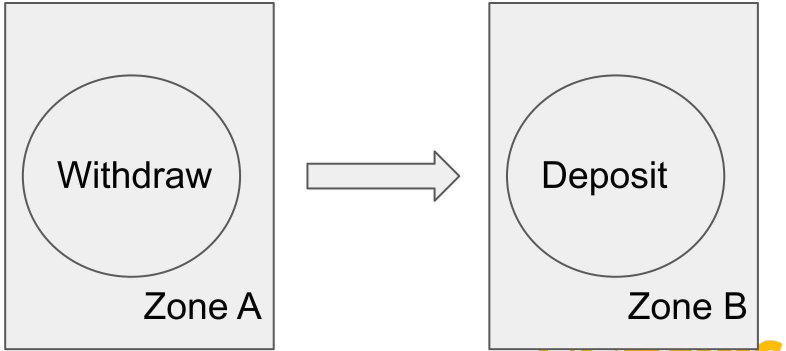
Inbound Relay TX $\boldsymbol{m'}$

Coinbase Transaction

Transaction Type

- Intra-Transaction
  - Taken care immediately
- Relay Transaction
  - Transferred to other zones

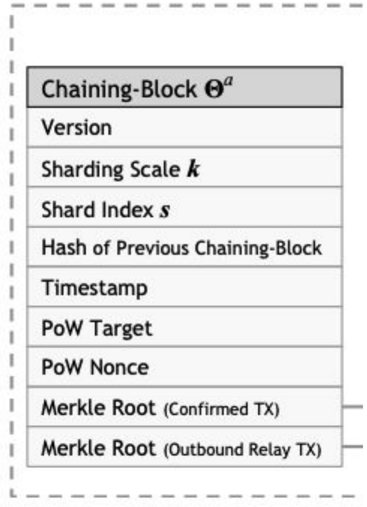




System Design - Relay Transaction

- Zone A

1. Unconfirmed transaction in zone A
2. Validating transaction while building a new block in zone A
   - Account balance > Transfer amount
3. Chaining-block builds



Chaining-Block Θ^a

Version

Sharding Scale k

Shard Index s

Hash of Previous Chaining-Block

Timestamp

PoW Target

PoW Nonce

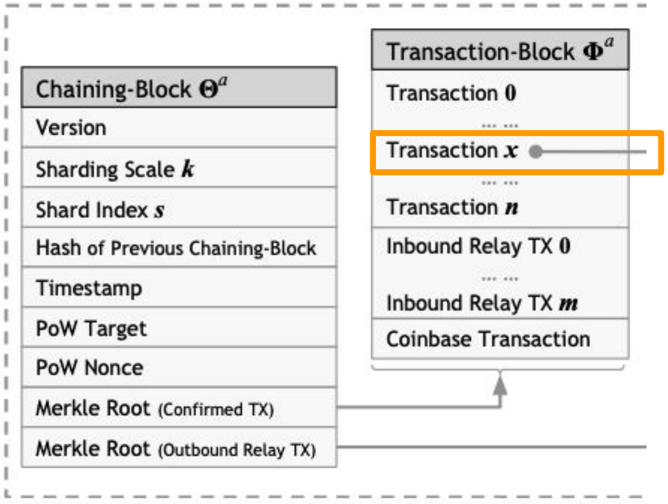
Merkle Root (Confirmed TX)

Merkle Root (Outbound Relay TX)

System Design - Relay Transaction

- Zone A

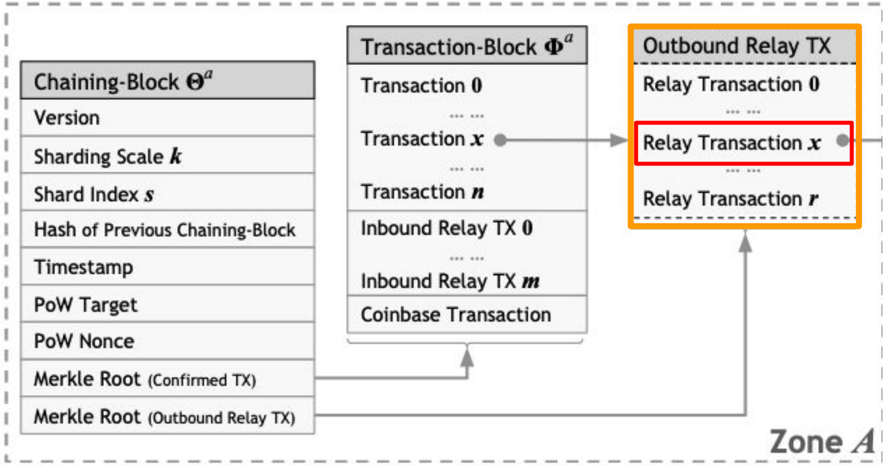
4. Transaction block builds



System Design - Relay Transaction

- Zone A

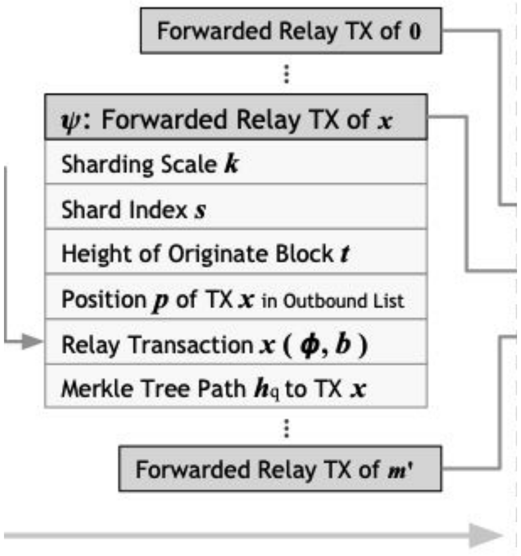
5. Chaining-block builds Outbound Relay TX



UCDAVIS COMPUTER SCIENCE

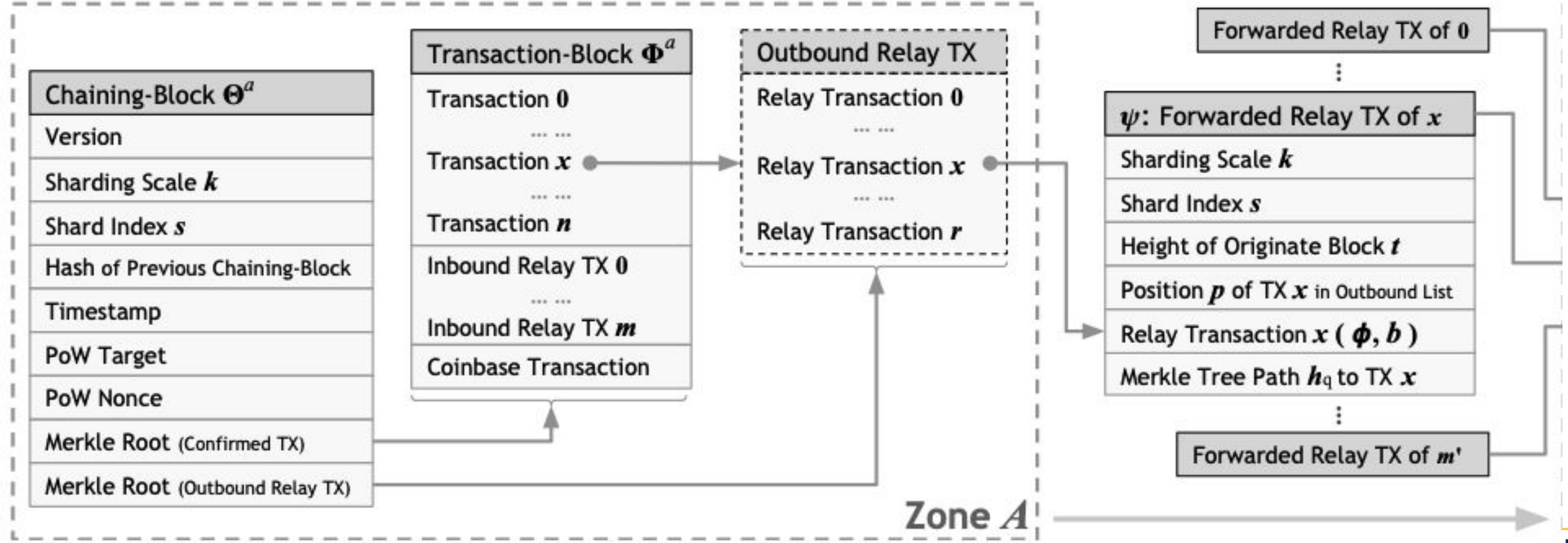
System Design - Relay Transaction

- Forwarded Relay TX
- Like a package


Forwarded Relay TX of 0
⋮
ψ: Forwarded Relay TX of x
Sharding Scale k
Shard Index s
Height of Originate Block t
Position p of TX x in Outbound List
Relay Transaction x (ϕ, b)
Merkle Tree Path h_q to TX x
⋮
Forwarded Relay TX of m'

System Design - Relay Transaction

- So far ...


Chaining-Block Θ^a
Version
Sharding Scale k
Shard Index s
Hash of Previous Chaining-Block
Timestamp
PoW Target
PoW Nonce
Merkle Root (Confirmed TX)
Merkle Root (Outbound Relay TX)
Transaction-Block Φ^a
Transaction 0
Transaction x
Transaction n
Inbound Relay TX 0
Inbound Relay TX m
Coinbase Transaction
Outbound Relay TX
Relay Transaction 0
Relay Transaction x
Relay Transaction r
Zone A
Forwarded Relay TX of 0
ψ: Forwarded Relay TX of x
Sharding Scale k
Shard Index s
Height of Originate Block t
Position p of TX x in Outbound List
Relay Transaction x (ϕ, b)
Merkle Tree Path h_q to TX x
Forwarded Relay TX of m'

System Design - Relay Transaction

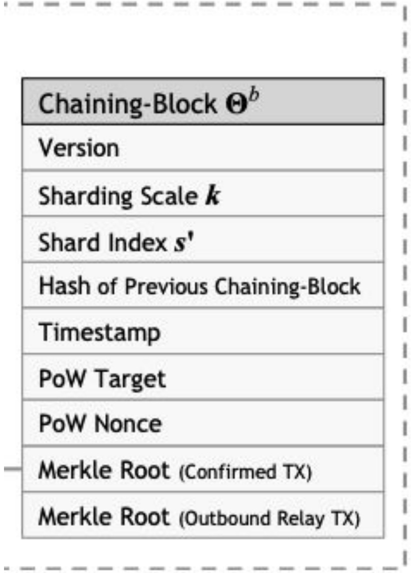
- Zone B

1. An inbound transaction is picked up by a miner
2. The miner verifies the inbound relay transaction

System Design - Relay Transaction

- Zone B

3. Chaining-block builds with inbound transaction information



Chaining-Block Θ^b

Version

Sharding Scale k

Shard Index s'

Hash of Previous Chaining-Block

Timestamp

PoW Target

PoW Nonce

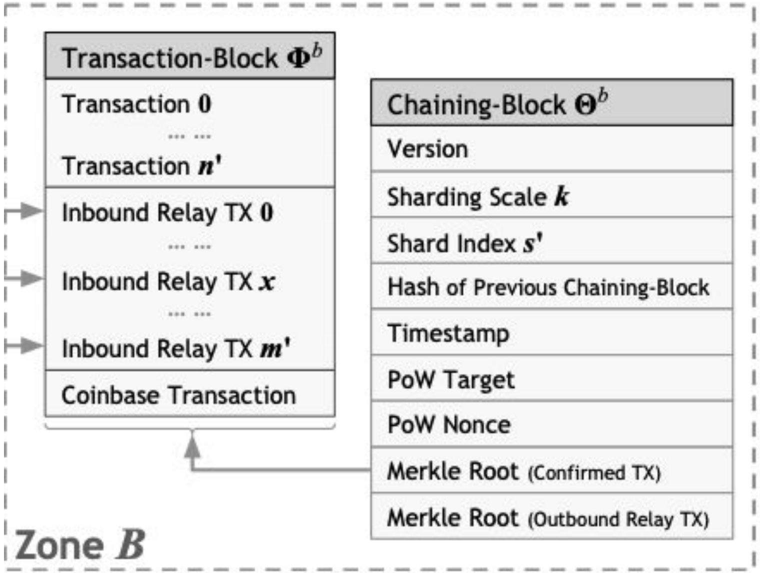
Merkle Root (Confirmed TX)

Merkle Root (Outbound Relay TX)

System Design - Relay Transaction

- Zone B

4. Transaction-block builds according to Chaining-block



Transaction-Block Φ^b

Transaction 0

... ...

Transaction n'

Inbound Relay TX 0

... ...

Inbound Relay TX x

... ...

Inbound Relay TX m'

Coinbase Transaction

Chaining-Block Θ^b

Version

Sharding Scale k

Shard Index s'

Hash of Previous Chaining-Block

Timestamp

PoW Target

PoW Nonce

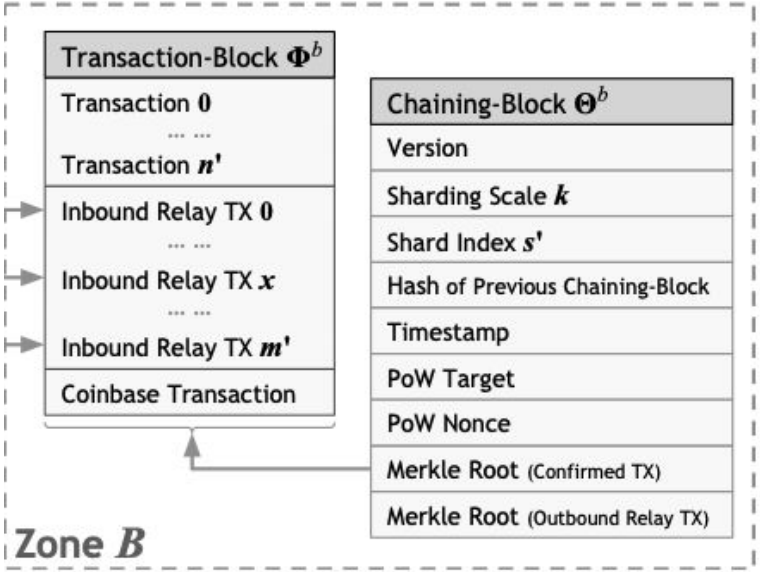
Merkle Root (Confirmed TX)

Merkle Root (Outbound Relay TX)

System Design - Relay Transaction

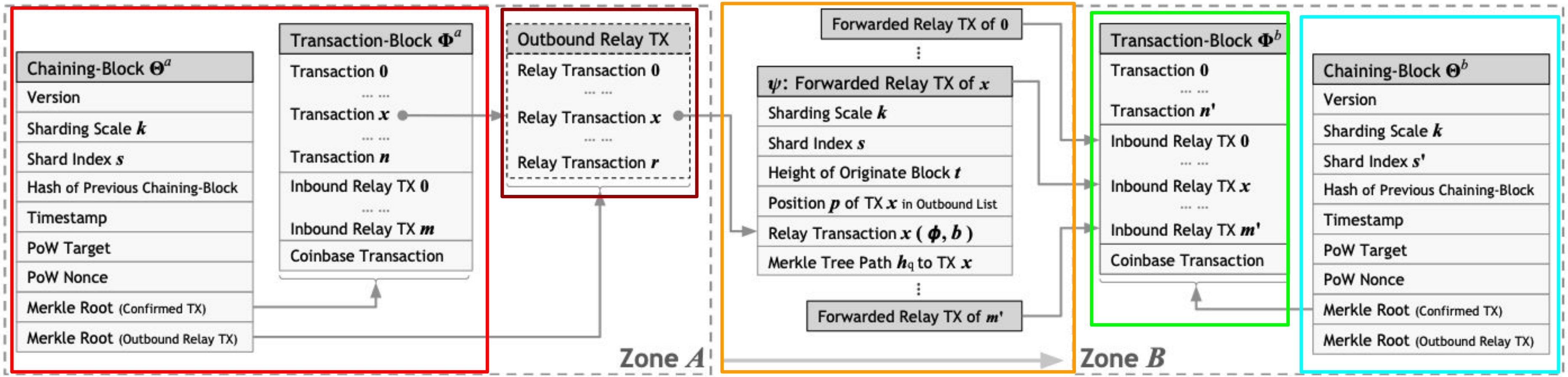
- Zone B

5. The deposit operation is executed, concluding the transaction process.



System Design - Relay Transaction

- The whole diagram



Chaining-Block Θ^a
Version
Sharding Scale k
Shard Index s
Hash of Previous Chaining-Block
Timestamp
PoW Target
PoW Nonce
Merkle Root (Confirmed TX)
Merkle Root (Outbound Relay TX)

Transaction-Block Φ^a
Transaction 0
... ...
Transaction x
... ...
Transaction n
Inbound Relay TX 0
... ...
Inbound Relay TX m
Coinbase Transaction

Outbound Relay TX
Relay Transaction 0
... ...
Relay Transaction x
... ...
Relay Transaction r

Zone A

Forwarded Relay TX of 0
⋮
ψ: Forwarded Relay TX of x
Sharding Scale k
Shard Index s
Height of Originate Block t
Position p of TX x in Outbound List
Relay Transaction x (ϕ, b)
Merkle Tree Path h_q to TX x
⋮
Forwarded Relay TX of m'

Zone B

Transaction-Block Φ^b
Transaction 0
... ...
Transaction n'
Inbound Relay TX 0
... ...
Inbound Relay TX x
... ...
Inbound Relay TX m'
Coinbase Transaction

Chaining-Block Θ^b
Version
Sharding Scale k
Shard Index s'
Hash of Previous Chaining-Block
Timestamp
PoW Target
PoW Nonce
Merkle Root (Confirmed TX)
Merkle Root (Outbound Relay TX)

Chu-Ko-Nu

UCDAVIS
COMPUTER SCIENCE

Defense Per-Zone Security

- A rational miner will ideally distribute its total mining power in different zones to maximize the rewards. This makes the mining power of the entire network H converge to be evenly distributed across zones => H/n.

- When a malicious miner gathers all its mining power T focuses on a single zone, the attack will succeed if $T > H/n$ × 50%, which will be unacceptably low when with a large n

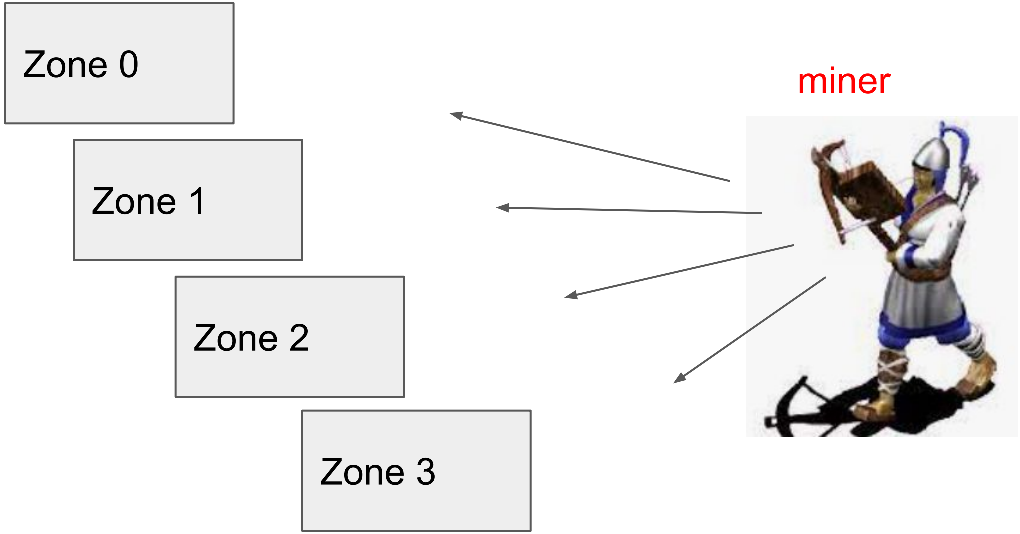
Chu-Ko-Nu Mining

- It ensures the effective mining power in each zone to be at the same level of the entire network

- It makes an attack on any individual zone as hard as that on the full network

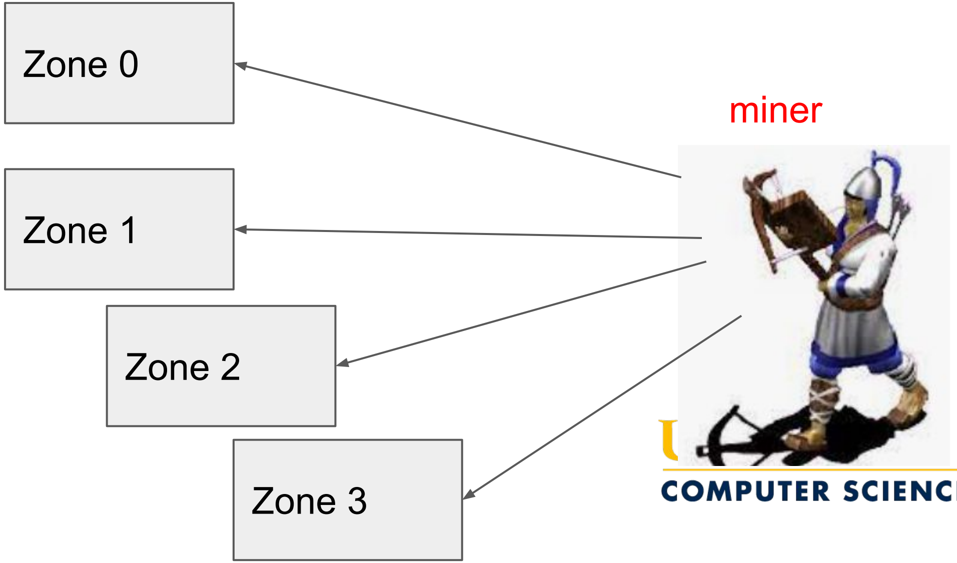
Concept of Chu-Ko-Nu

- A miner use a single PoW solution to create multiple blocks in different zones



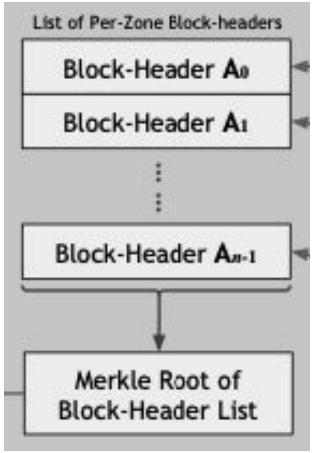
Concepts of Chu-Ko-Nu

- Contraint: each miner can create at most one block in each zone (Malicious miner can also be Chu-Ko-Nu)
- Chu-Ko-Nu miners coexist with conventional miners
- Every zone has the same PoW
- Best Scenario:
  - All miners shot all zones

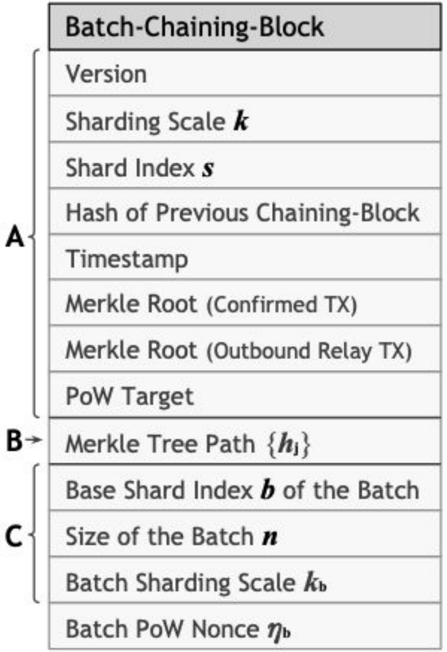


Chu-Ko-Nu Mining

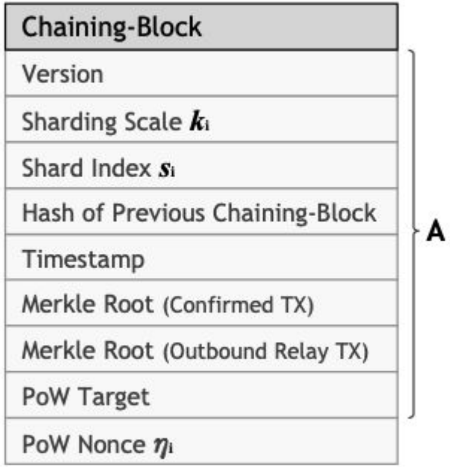
- Batch-Chaining-Block replaces Chaining-Block


List of Per-Zone Block-headers
Block-Header A_0
Block-Header A_1
Block-Header A_{n-1}
Merkle Root of Block-Header List

$$h_0$$



	Batch-Chaining-Block
A	Version
	Sharding Scale k
	Shard Index s
	Hash of Previous Chaining-Block
	Timestamp
	Merkle Root (Confirmed TX)
	Merkle Root (Outbound Relay TX)
	PoW Target
B	Merkle Tree Path $\{h_i\}$
C	Base Shard Index b of the Batch
	Size of the Batch n
	Batch Sharding Scale k_b
	Batch PoW Nonce η_b



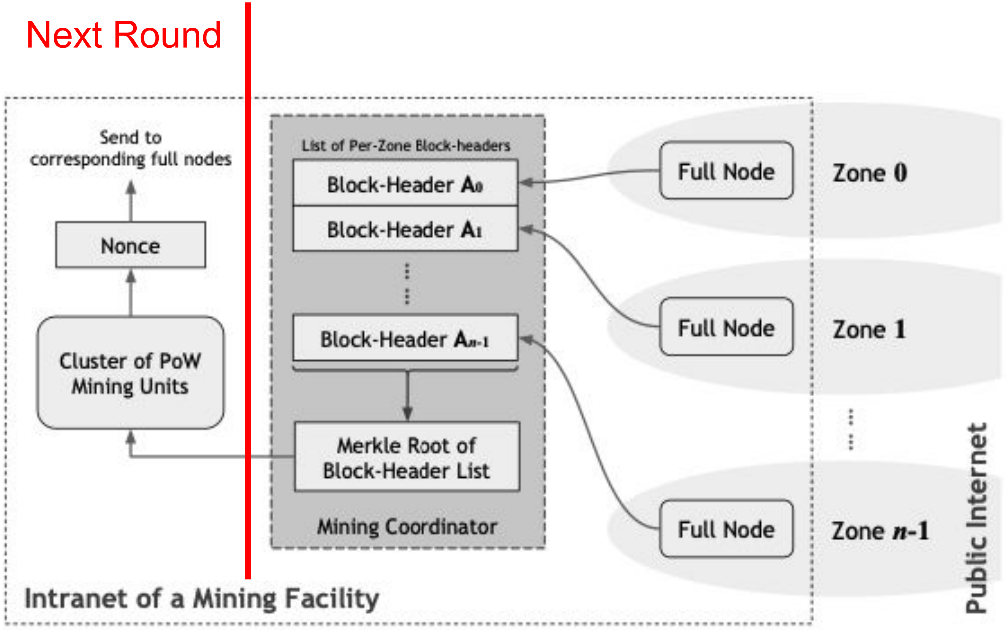
Chaining-Block	
Version	A
Sharding Scale k_i	
Shard Index s_i	
Hash of Previous Chaining-Block	
Timestamp	
Merkle Root (Confirmed TX)	
Merkle Root (Outbound Relay TX)	
PoW Target	
PoW Nonce η_i	

$$\text{hash}(\langle h_0, \text{C}, \eta_b \rangle) < \tau,$$

$$\text{hash}(\langle \text{A}_i, \eta_i \rangle) < \tau,$$

Chu-ko-nu Mining

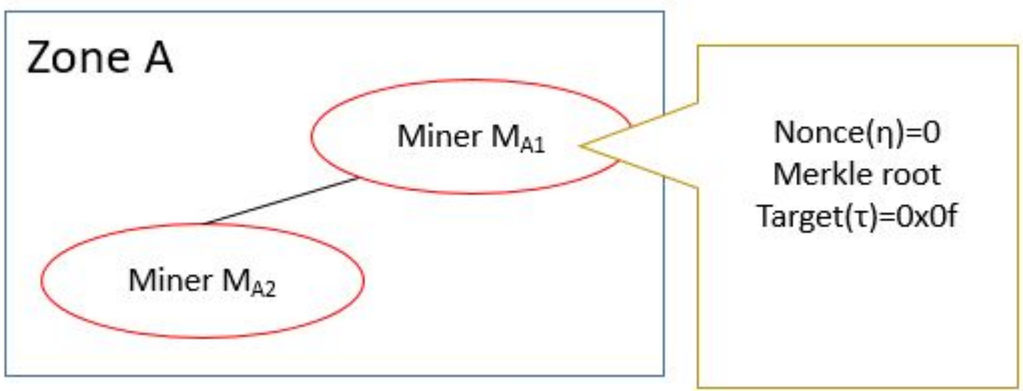
- Once Ai is updated in any zone i, new Ai will be sent to Mining Coordinator
- Mining Coordinator will recalculated the merkel tree of block-header list, and update Merkle tree root
- Broadcast new Merkel tree root to all miners
- All miners will recalculate nonce



UCDAVIS COMPUTER SCIENCE

Mining Mechanism

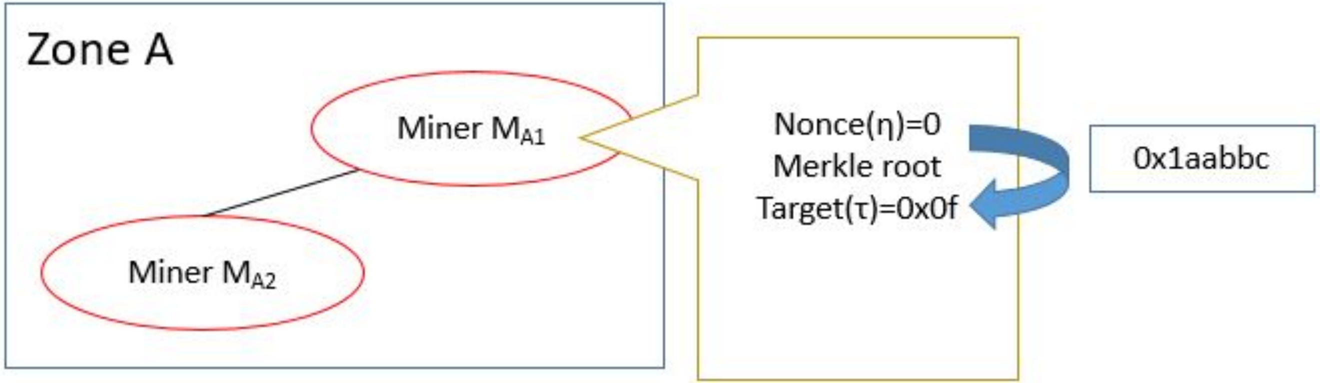
- Without Chu-Ko-Nu



Nonce(η)=0
Merkle root
Target(τ)=0x0f

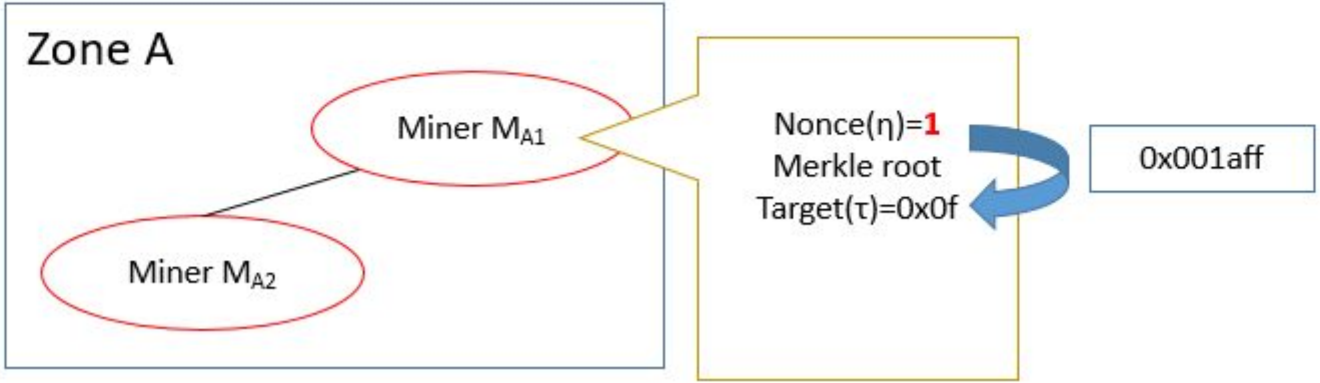
Mining Mechanism

- Without Chu-Ko-Nu



Mining Mechanism

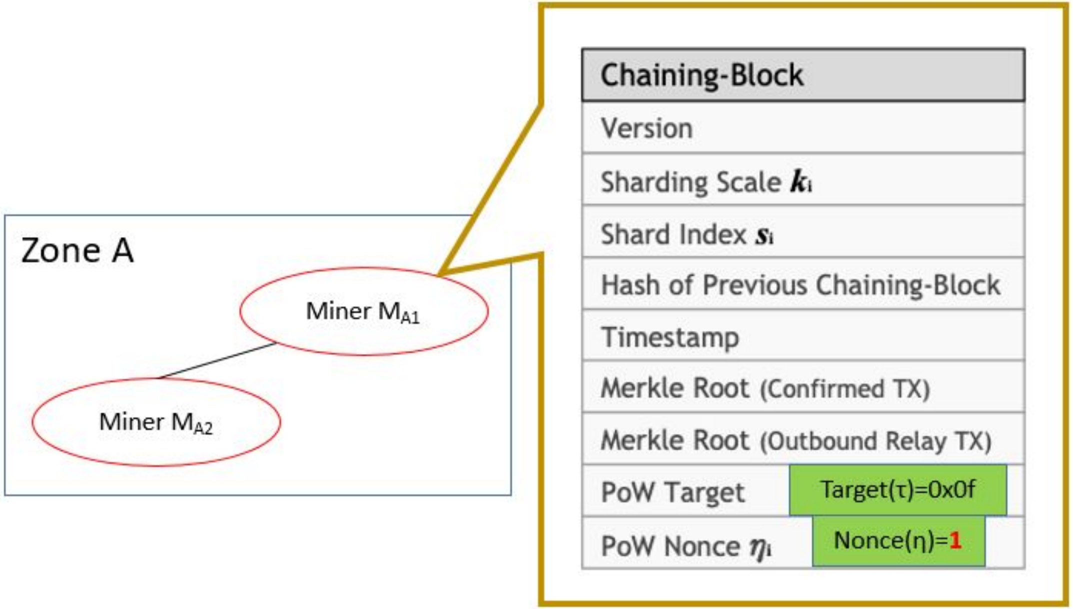
- Without Chu-Ko-Nu



UCDAVIS COMPUTER SCIENCE

Mining Mechanism

- Without Chu-Ko-Nu

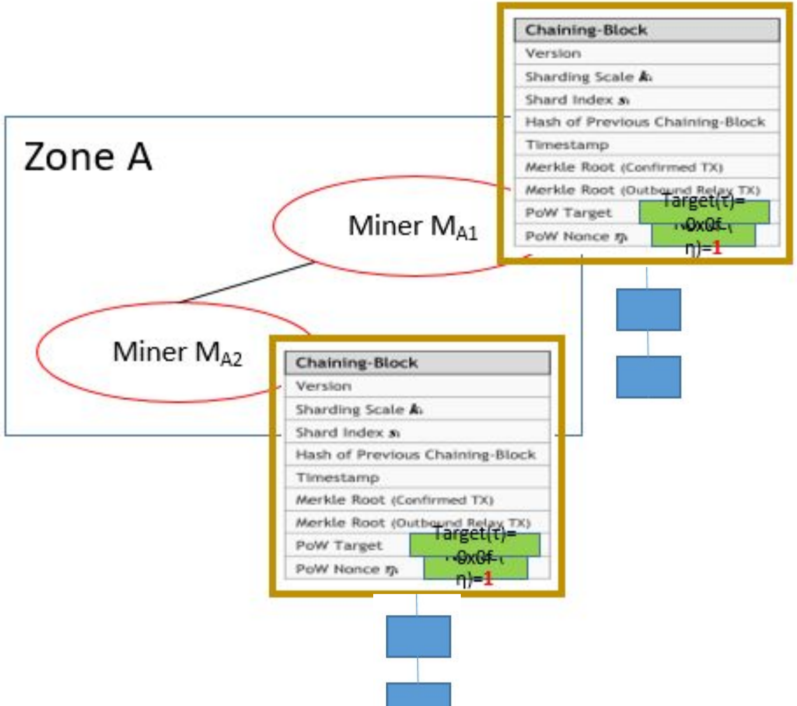


Chaining-Block	
Version	
Sharding Scale k_i	
Shard Index s_i	
Hash of Previous Chaining-Block	
Timestamp	
Merkle Root (Confirmed TX)	
Merkle Root (Outbound Relay TX)	
PoW Target	Target(τ)=0x0f
PoW Nonce η_i	Nonce(η)=1

UCDAVIS
COMPUTER SCIENCE

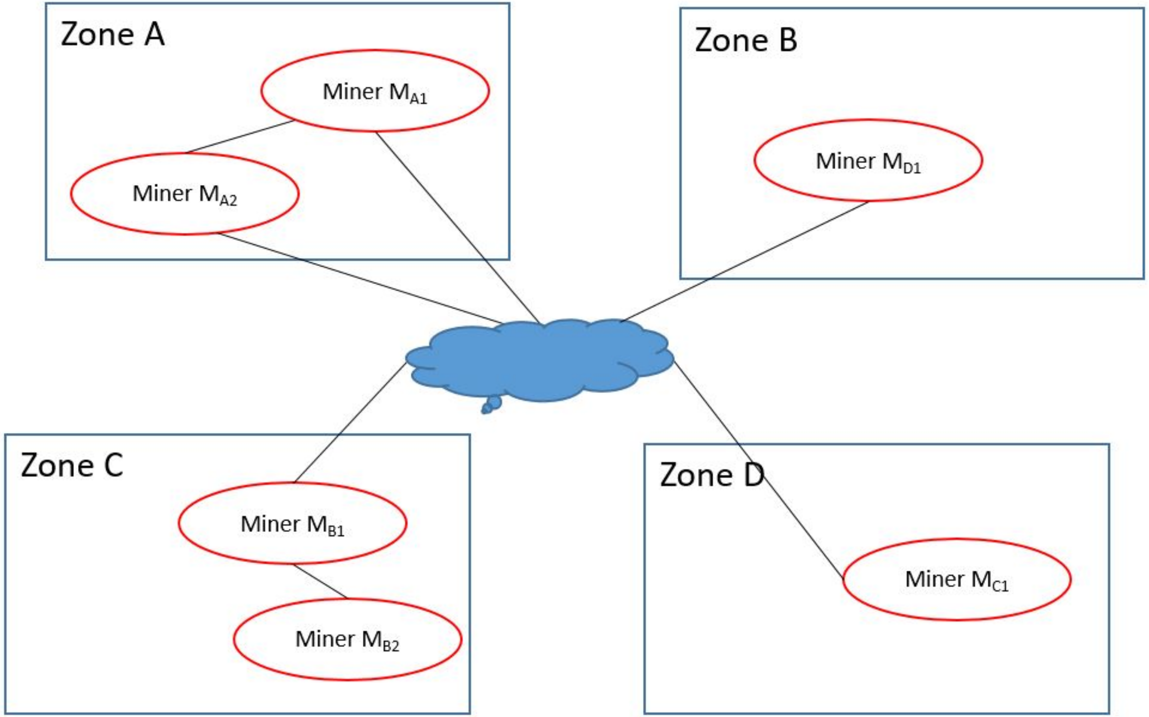
Mining Mechanism

- Without Chu-Ko-Nu



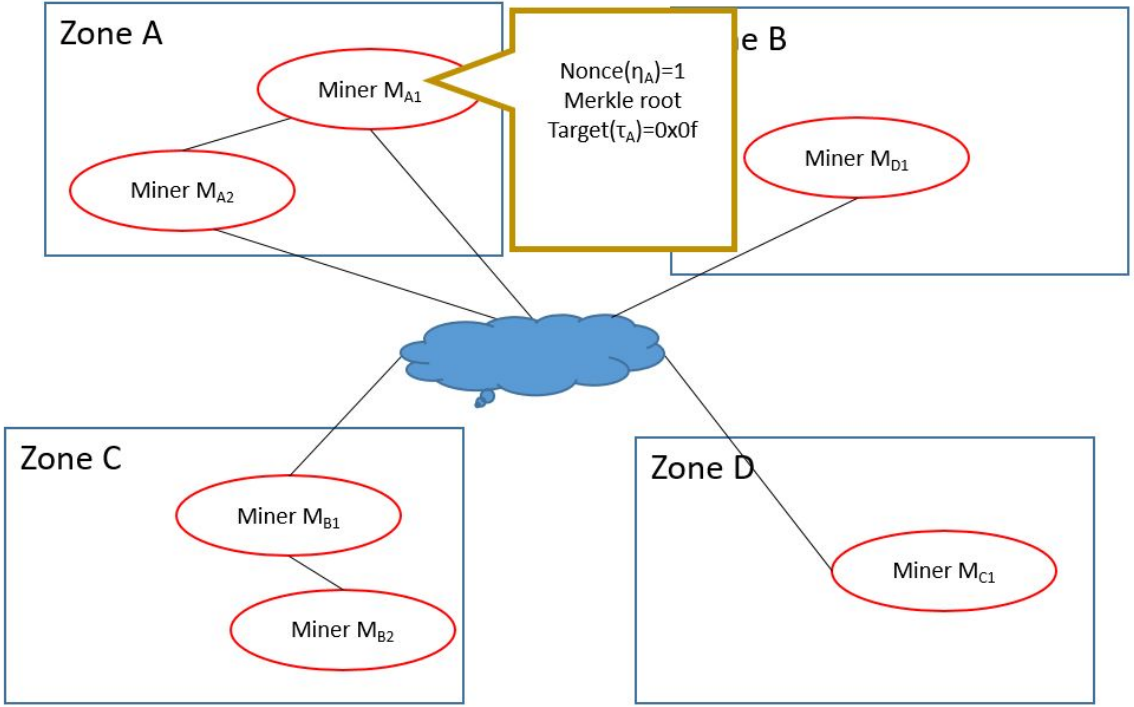
Mining Mechanism

- With Chu-Ko-Nu



Mining Mechanism

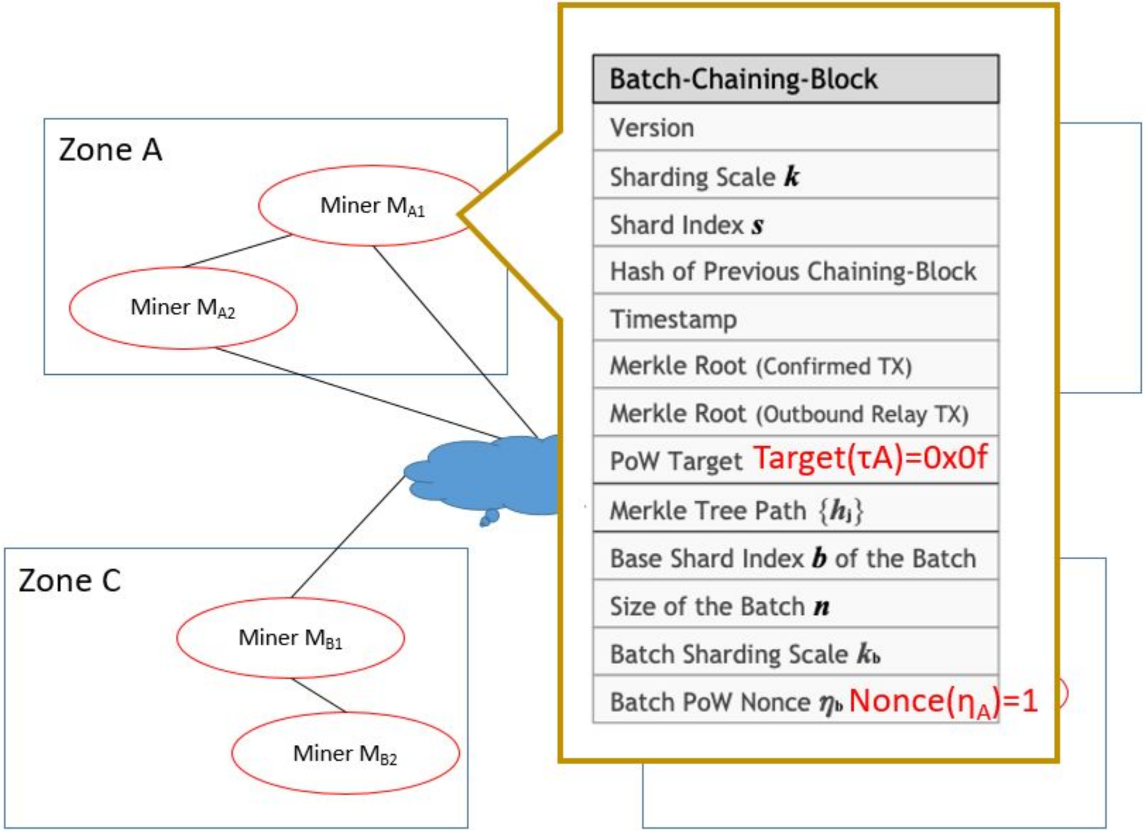
- With Chu-Ko-Nu



NCE

Mining Mechanism

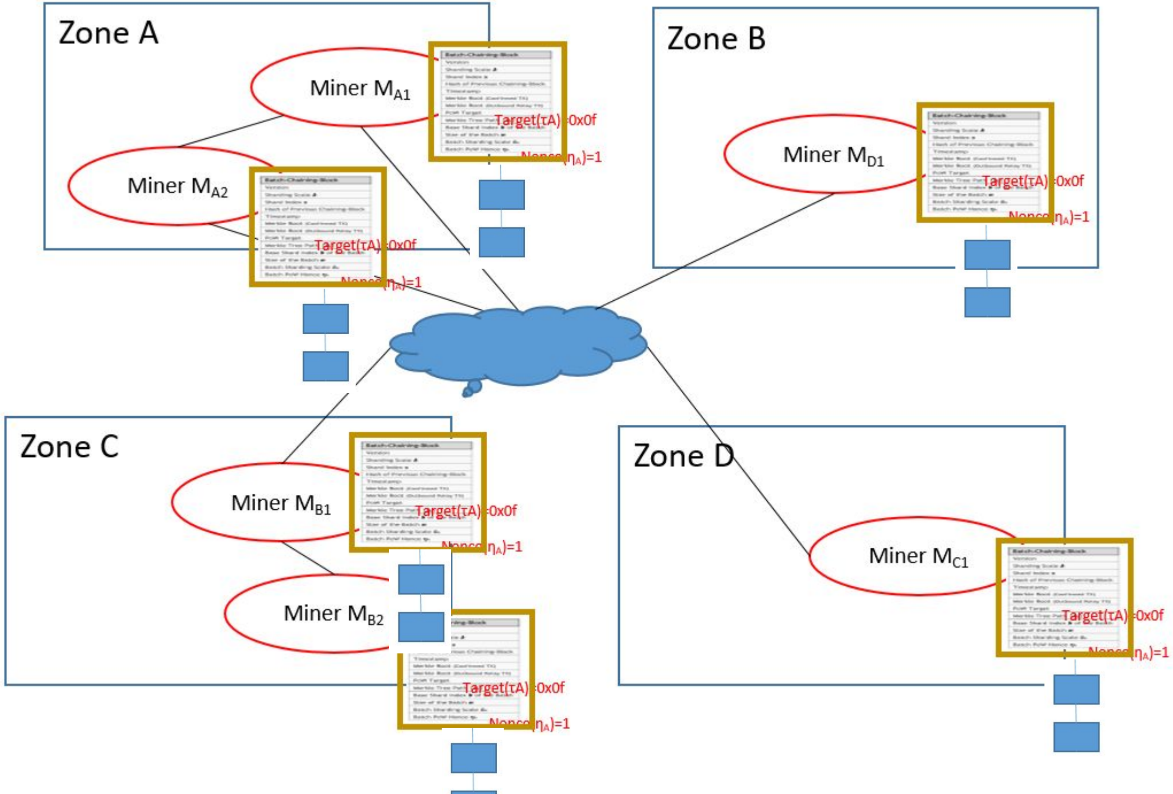
- With Chu-Ko-Nu



Batch-Chaining-Block
Version
Sharding Scale k
Shard Index s
Hash of Previous Chaining-Block
Timestamp
Merkle Root (Confirmed TX)
Merkle Root (Outbound Relay TX)
PoW Target Target(τA)=0x0f
Merkle Tree Path $\{h_j\}$
Base Shard Index b of the Batch
Size of the Batch n
Batch Sharding Scale k_b
Batch PoW Nonce η_b Nonce(η_A)=1

Mining Mechanism

- With Chu-Ko-Nu



Chu-ko-nu Mining

- m_p: the total physical hash rate of miners that participate in Chu-ko-nu mining
- m_d: the total physical hash rate of miners that don’t
- The effective hash rate m_s distributed in each zone can be calculated as

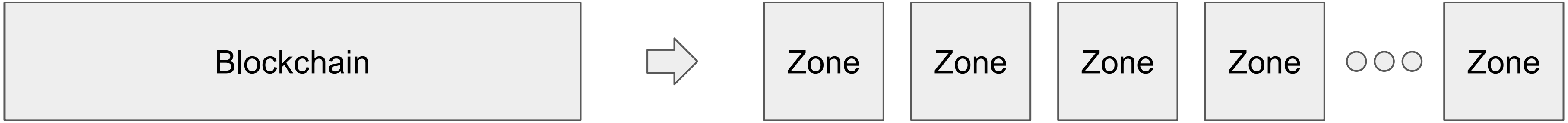
$$m_s = \frac{m_d}{2^k} + m_p.$$

- The attack bar in each zone

$$> \frac{m_s}{2 \cdot (m_d + m_p)} = 50\% - \frac{m_d \cdot (2^k - 1)/2^k}{2 \cdot (m_p + m_d)}$$

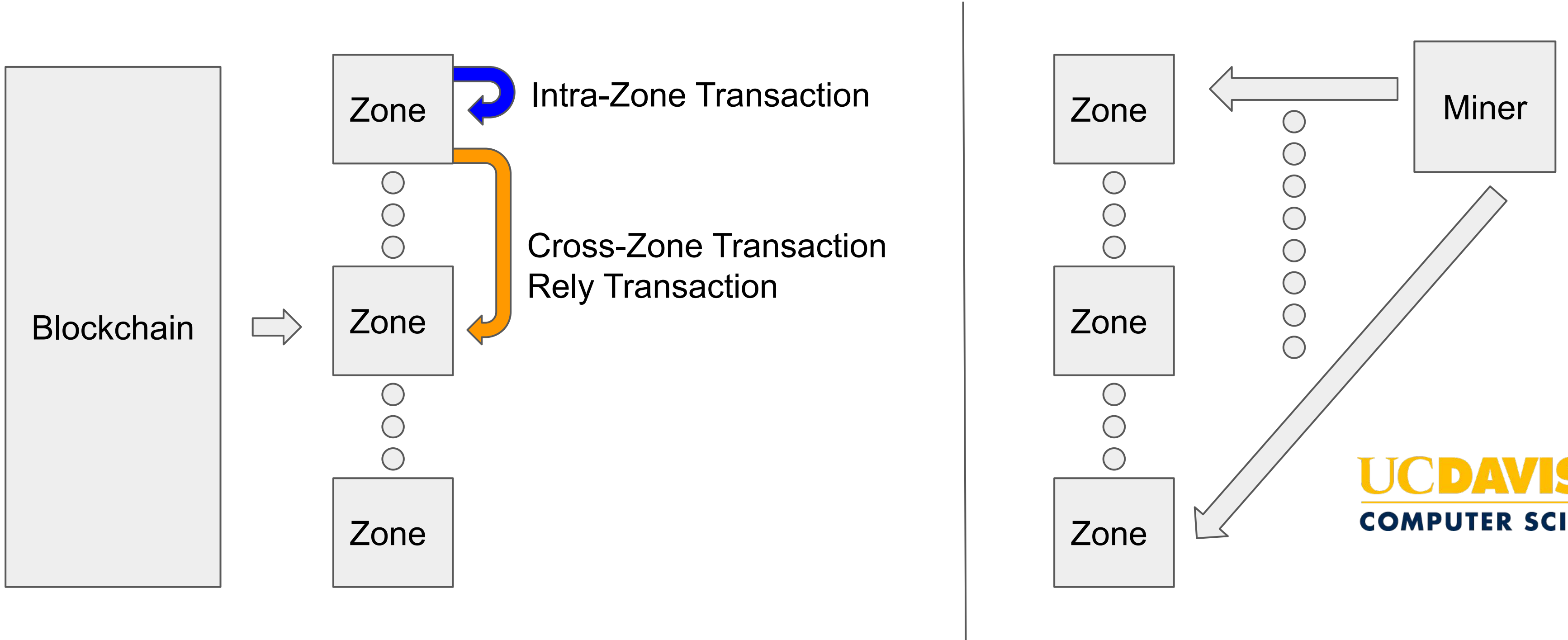
Discuss

- Main concept of the protocol - Sharding



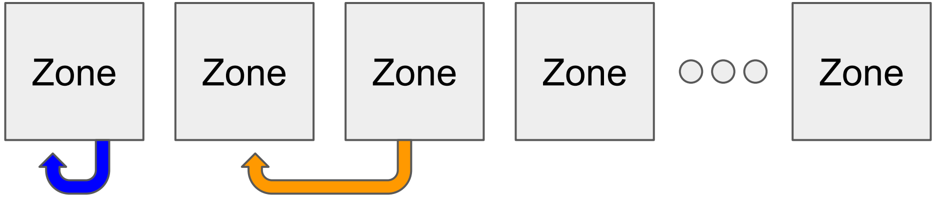
Overview

- Cross-Zone Transaction and Mining Protocol (Chu-ko-nu mining)



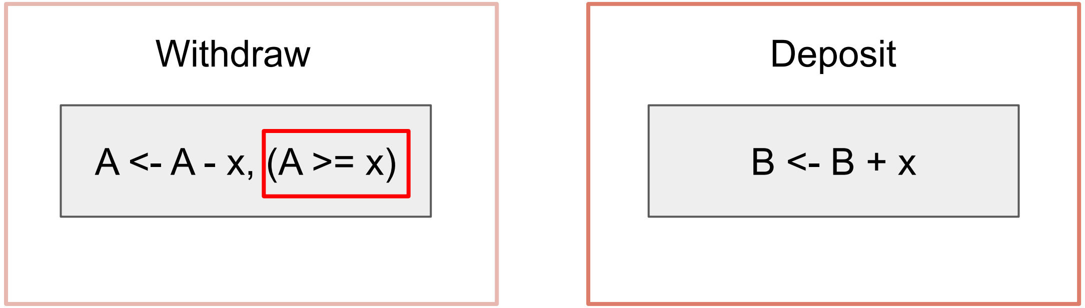
Asychornous Consenusu Zone

- Each zone work independently and all zones are asynchronous.
- A transaction in a same zone is straightforward.
- What about a cross-zone transaction?



Cross Zone

- For example, a user in zone A wants to transfer x tokens to another user in zone B.



Withdraw

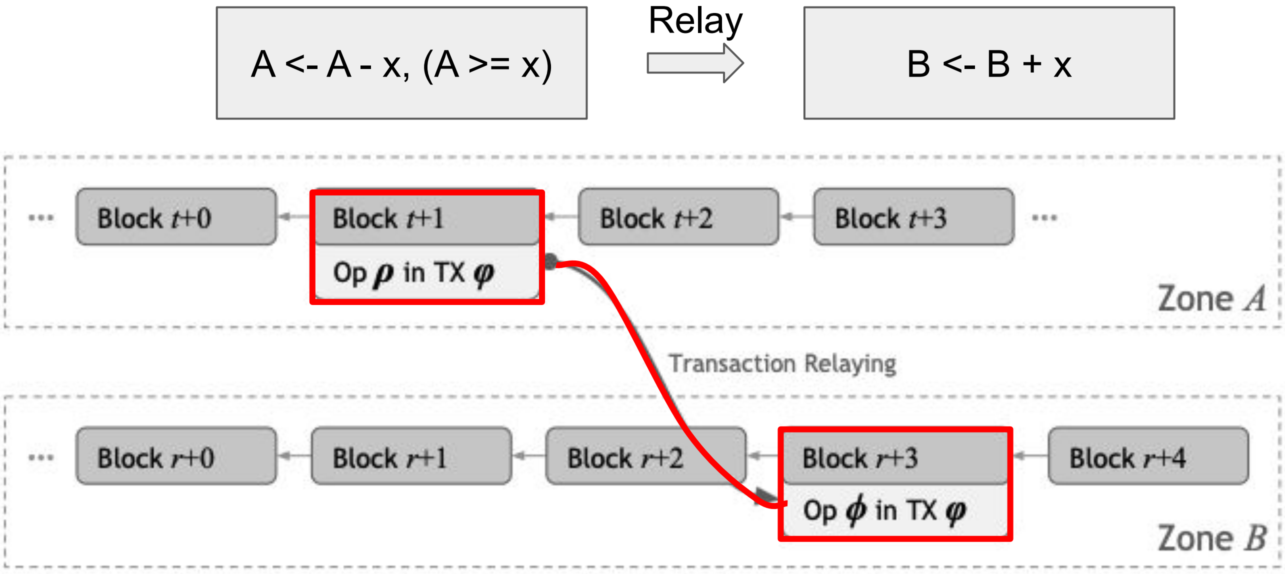
A <- A - x, (A >= x)

Deposit

B <- B + x

Cross Zone

- For example, a user in zone A wants to transfer x tokens to another user in zone B.



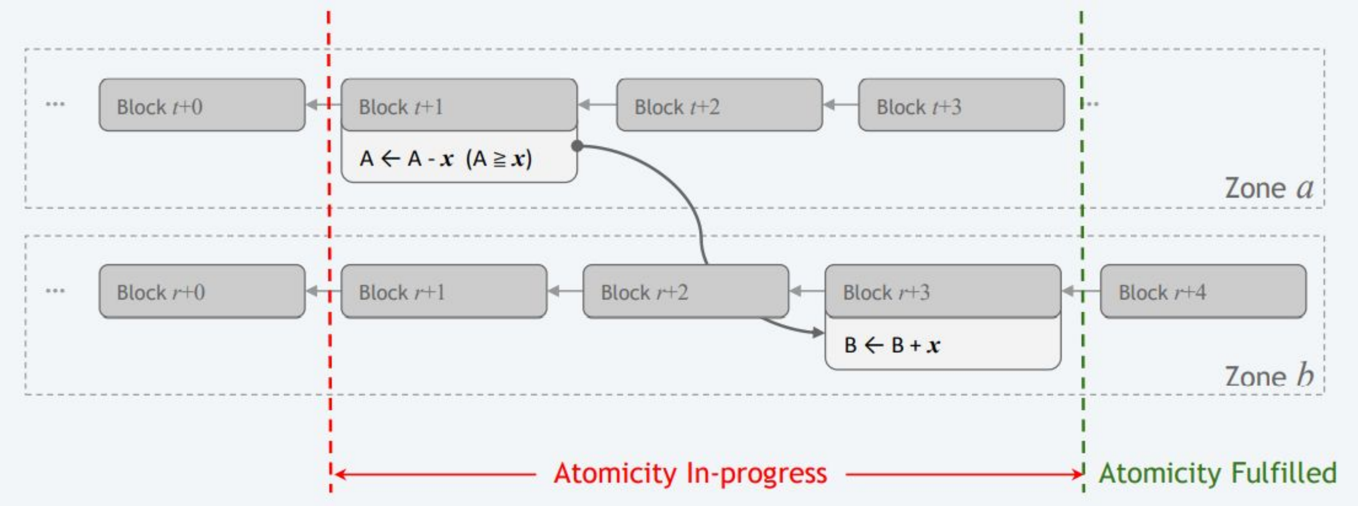
A <- A - x, (A >= x)

Relay

B <- B + x

Eventual Atomicity

- This paper purposes Eventual Atomicity to ensure that all relay transaction would be executed.

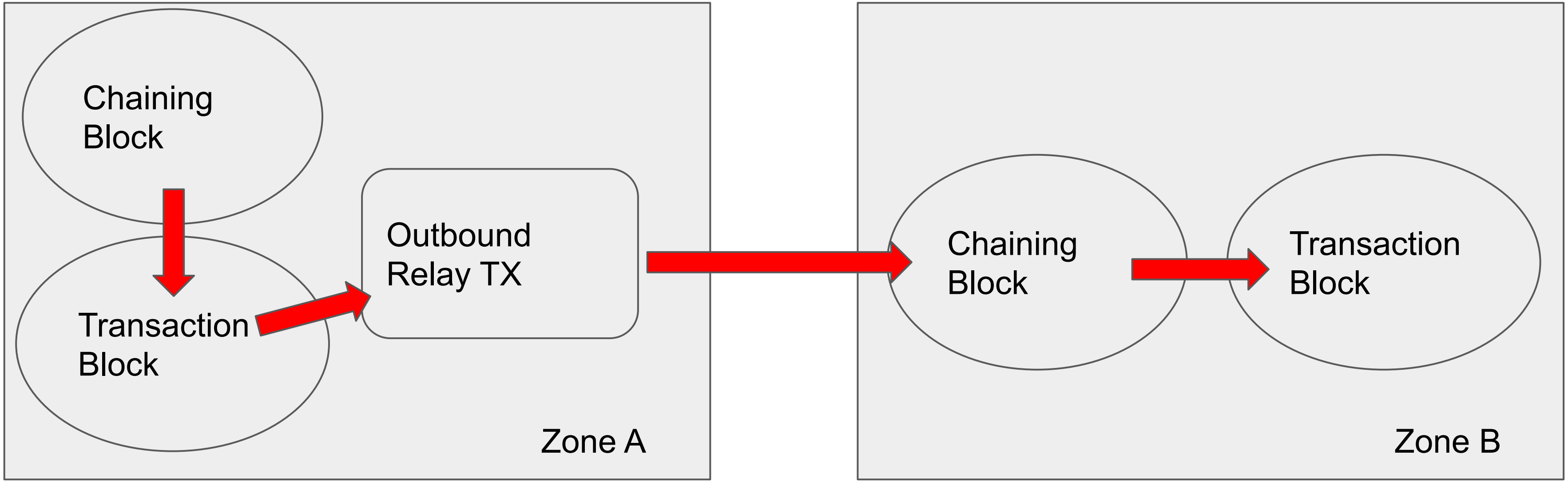


Ref

Eventual Atomicity

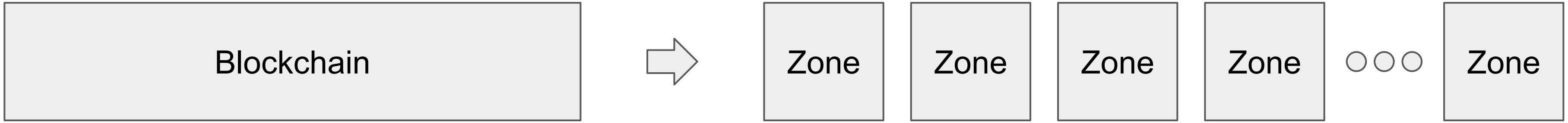
- In 2 phase commit, it has a roll back mechanism to prevent the invaild transaction.
- What about in this paper?

System Design

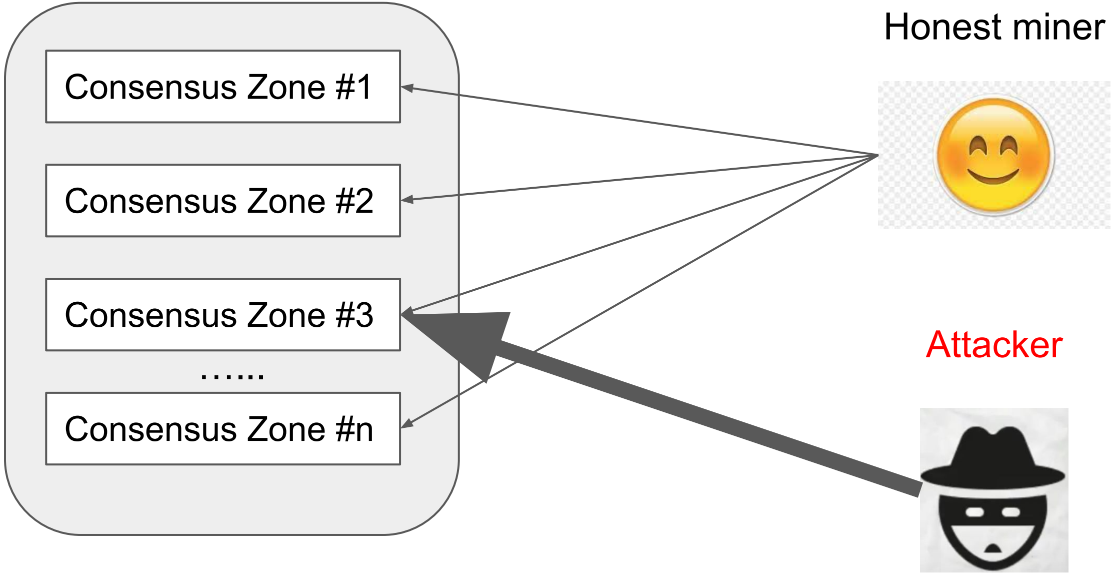


Chu-ko-nu Mining

- Goal: solve security issue of sharding



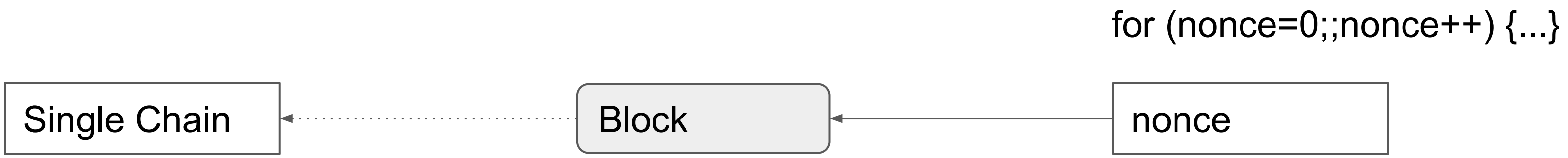
Security Issue: Single-Zone Focused Attack



UCDAVIS COMPUTER SCIENCE

Effective Mining Power

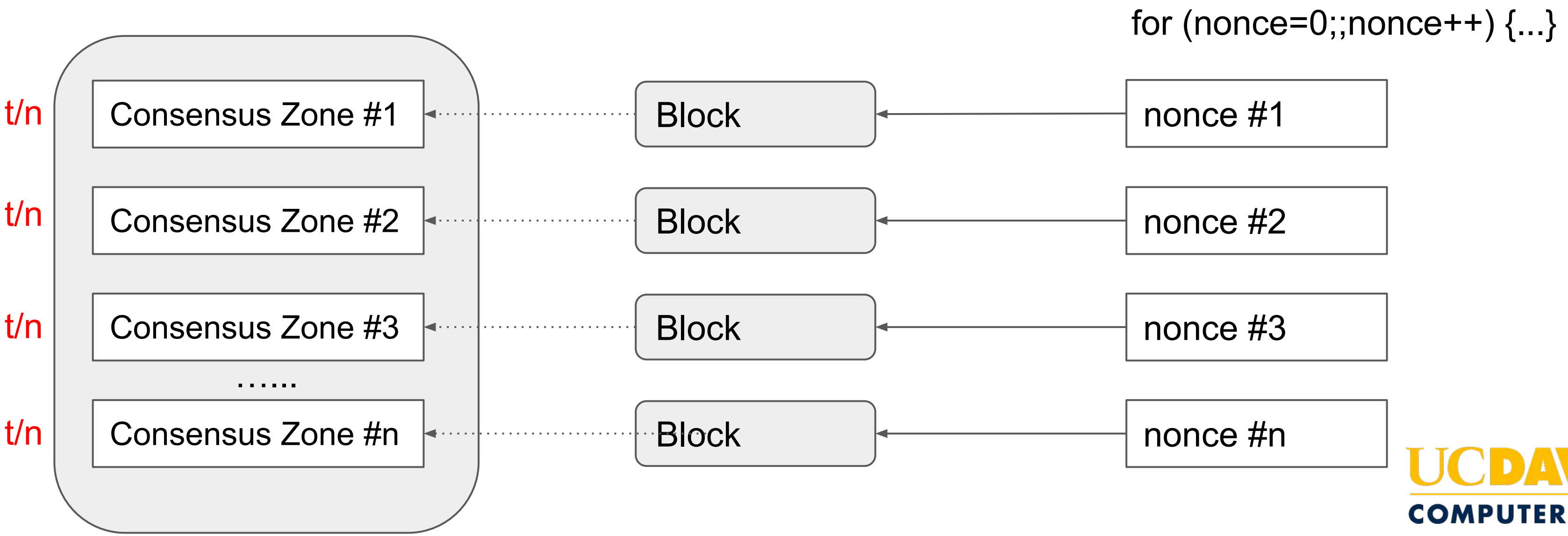
- Total Hashrate: t hash / sec
- Total Effective Mining Power: t hash / sec



```
for (nonce=0;;nonce++) {...}
```

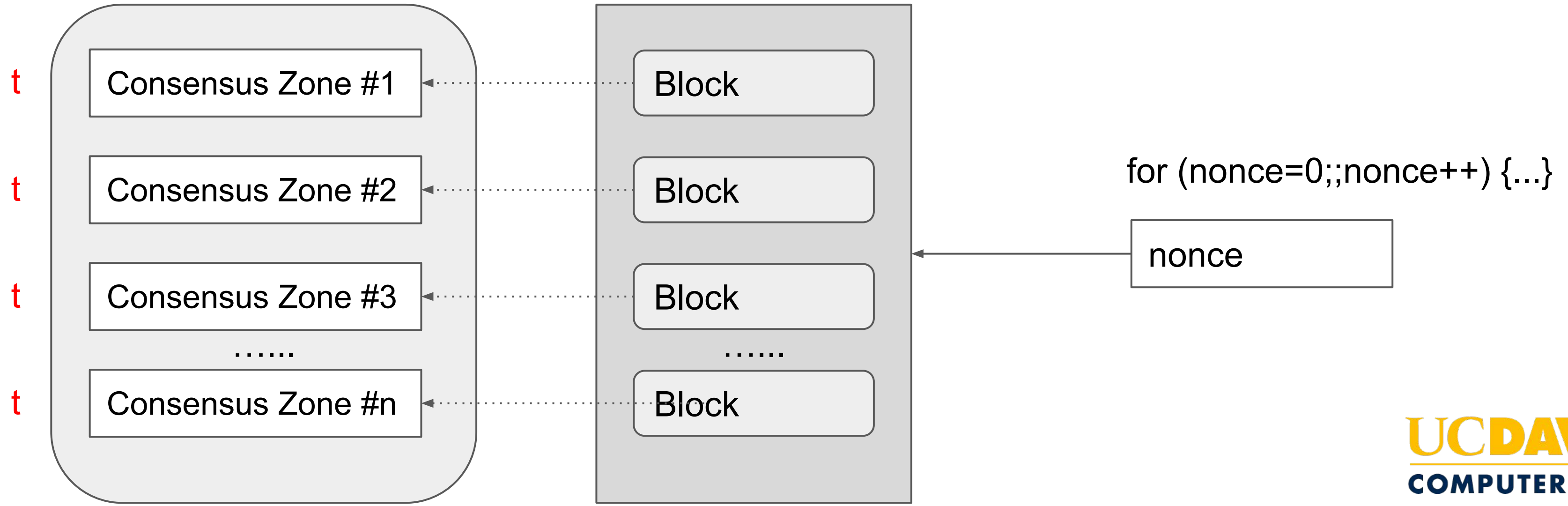
Effective Mining Power

- Total Hashrate: t hash / sec
- Total Effective Mining Power: t hash / sec



Chu-Ko-Nu Mining

- Total Hashrate: t hash / sec
- Total Effective Mining Power: t x n hash / sec



Experiment

Playback ERC20 historical payment transactions

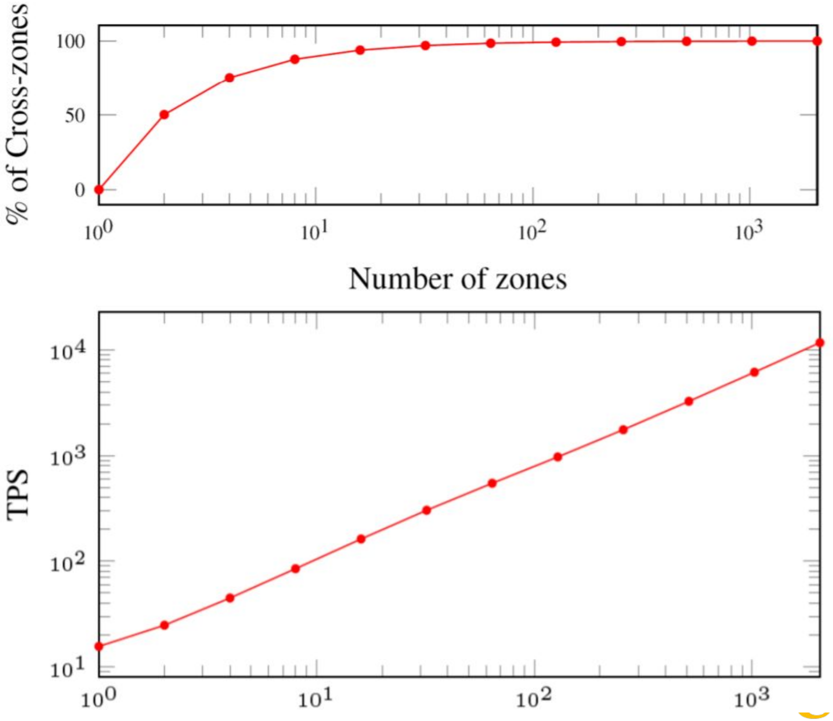
16.5 M Addresses

75.8 M Transactions

30Mbps per-node

15.6 TPS per-zone

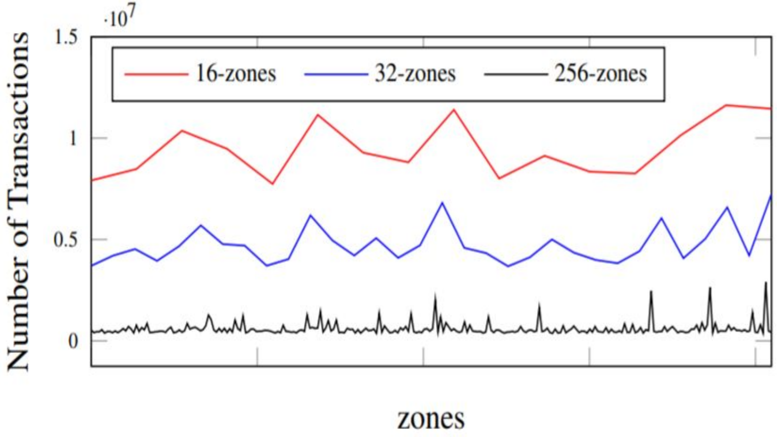
1 to 2048 zones



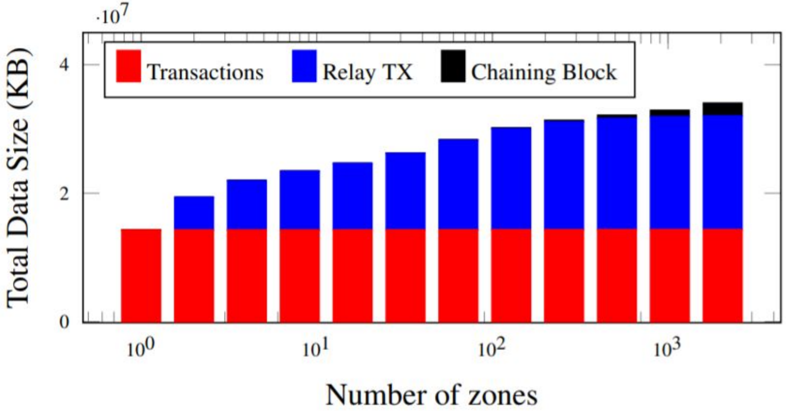
https://www.usenix.org/sites/default/files/conference/protected-files/nsdi19_slides_wang-jiaping.pdf

Experiment

Transaction distribution across zones



Sizes of the blockchain data in the entire network



Q & A

UCDAVIS
COMPUTER SCIENCE

what’s the difference between UTXO and Acc./Balance?

UTXO: T1 T2 T3 => O1 O2

A/B: A1B1-x => A2B2+x

Does invalid block will be recorded on the block(Merkle tree or real one block but tagged as invalid)?

Yes, when the block is orphaned, the block will no longer be valid.

Why the invalid block will be recorded?

The network latency.

Orphaned. It only exists on those miner who was synchronized when the fork occured.

How does a transaction be validated and committed in a block?

Each miner will attend to valid all the trasaction.

How does a BCB block be related to the block in other zones?

The Merkle tree path and configuration on the BCB.

Do zones need to synchronize? Or Ai could be updated very often and it will cause invalid and orphan block.