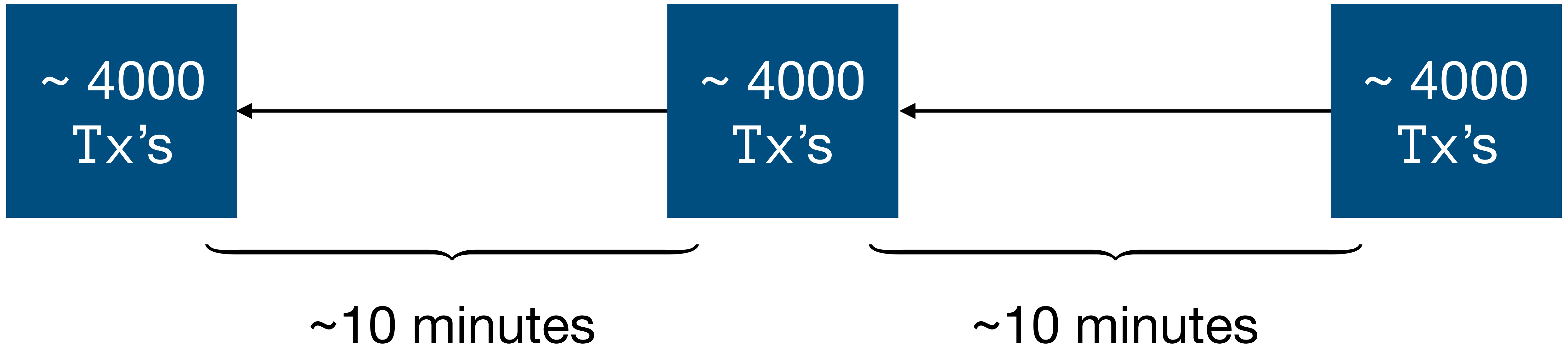
Consensus Mechanism Design based on Structured DAGs

Jiheng Zhang

HKUST

Blockchain



TPS = ~7

WHAT IS GREAT

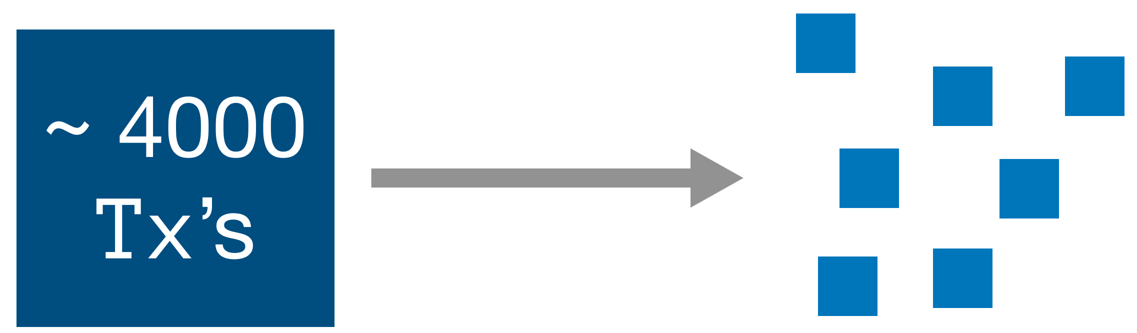
✓ Secure: Nakamoto consensus

✓ Decentralized: PoW

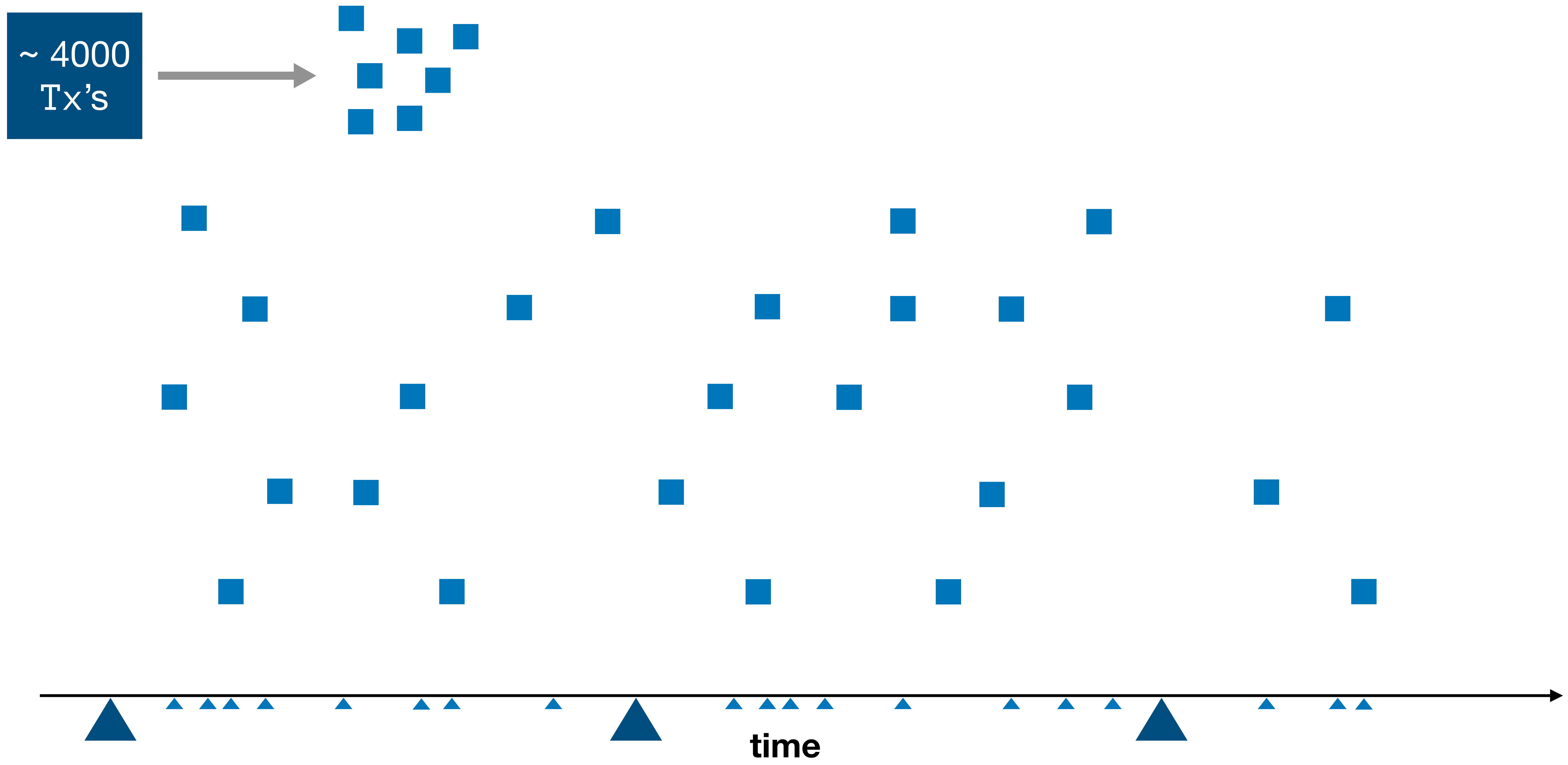
WHAT NEEDS IMPROVEMENT

- ☐ TPS
- ☐ Latency: waiting + confirmation
- ☐ Concentration of mining power
- ☐ Transactions with little fees

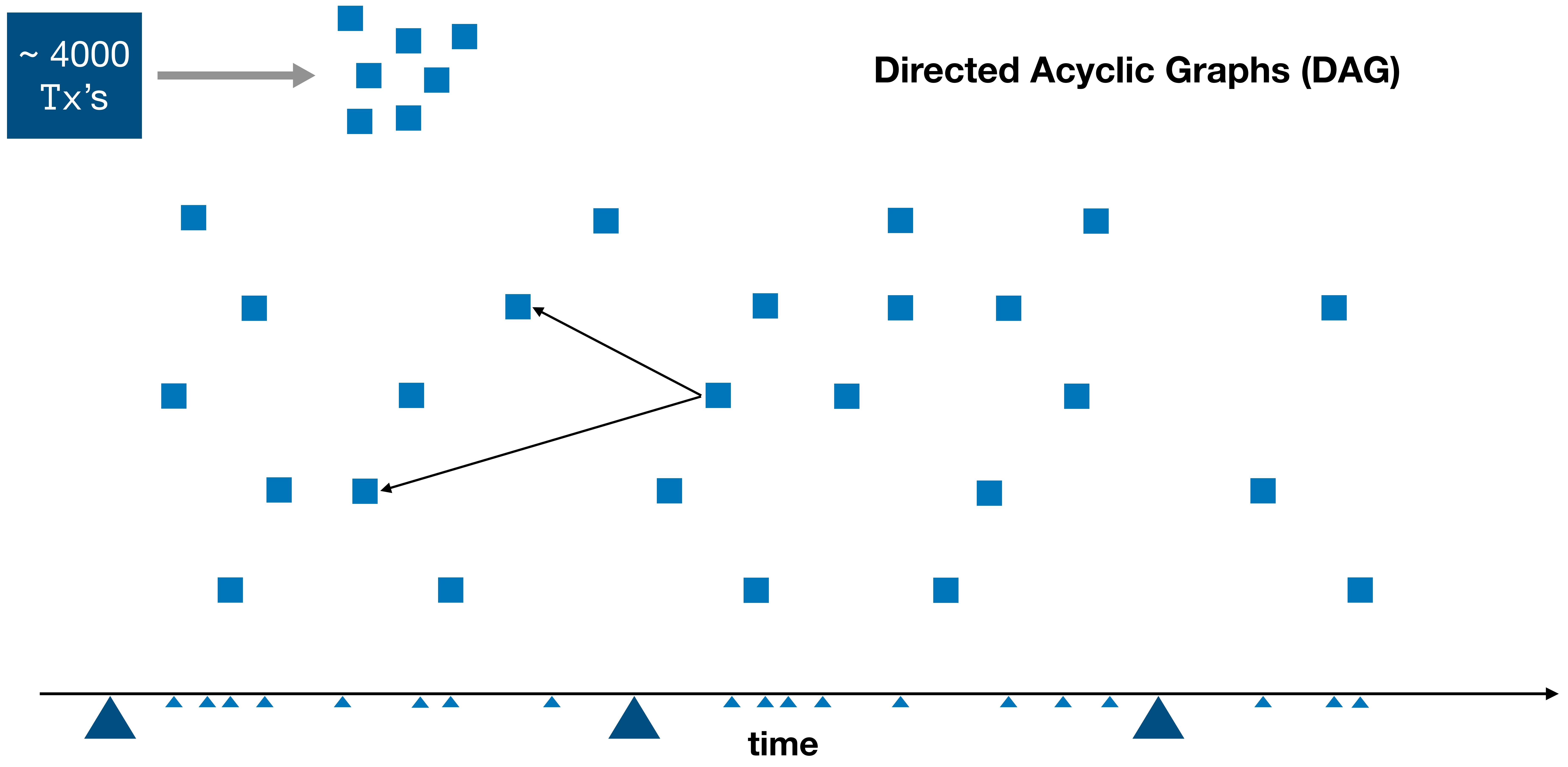
Parallelism



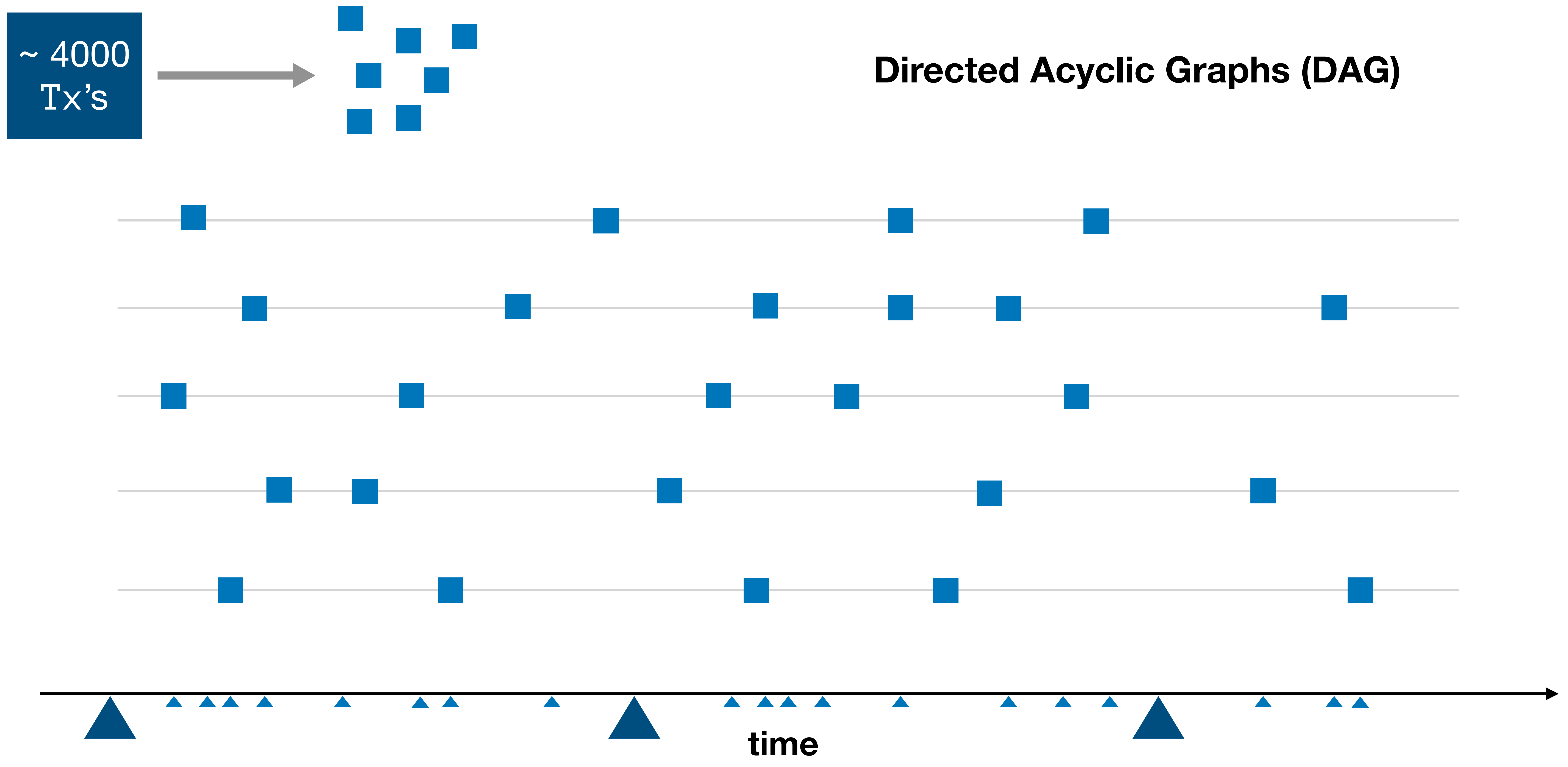
Parallelism



Parallelism



Parallelism



Structured DAGs

Structured DAGs

Block Structure

Block Size in bytes	
$\texttt{id}_{\texttt{prev}}$	32
$\texttt{id}_{\texttt{ms}}$	32
$\texttt{id}_{\texttt{tip}}$	32
`nonce`	4
`peer`	65
`message`	~ 500

Block Structure

Peer chain

Previous block by the same peer

Block Size in bytes	
id_{prev}	32
id_{ms}	32
id_{tip}	32
nonce	4
peer	65
message	~ 500

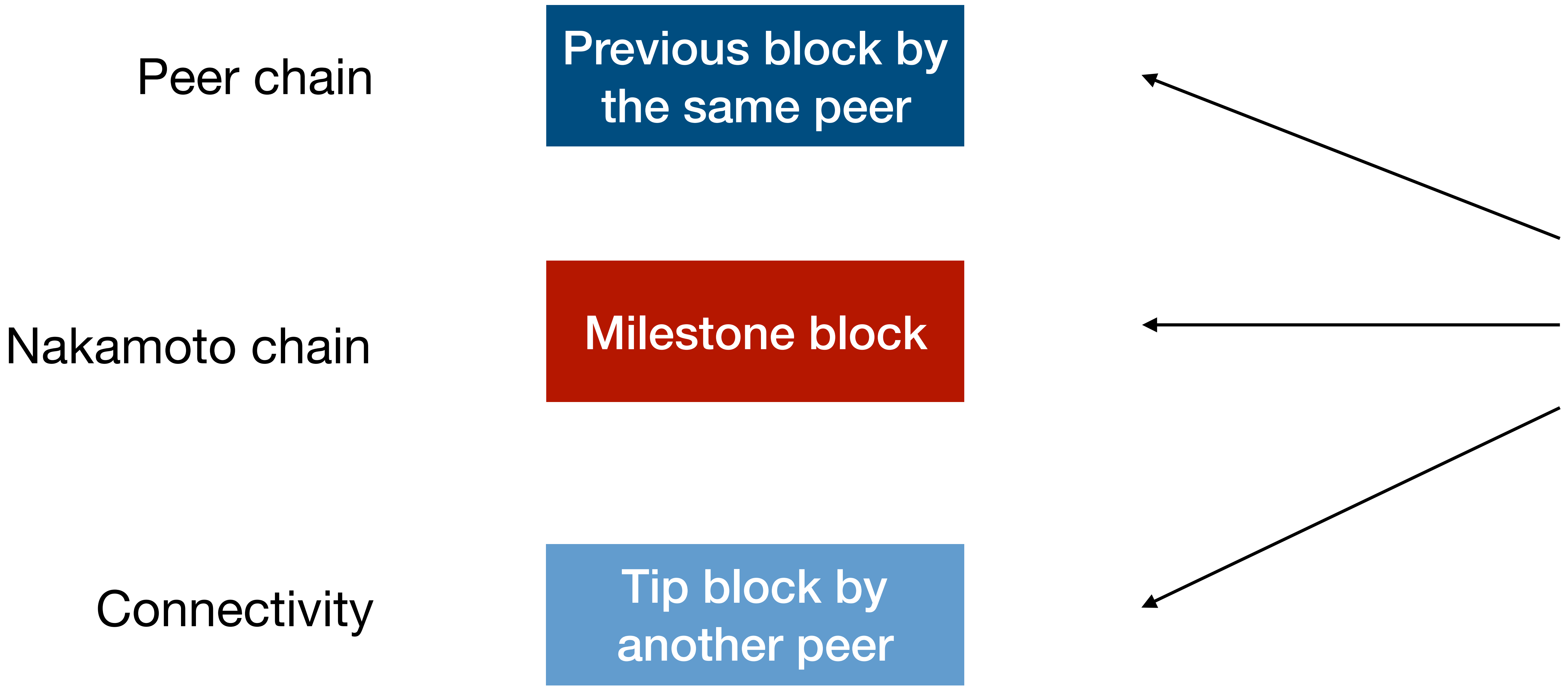
Block Structure

Peer chain — Previous block by the same peer

Nakamoto chain — Milestone block

Block Size in bytes	
id_{prev}	32
id_{ms}	32
id_{tip}	32
nonce	4
peer	65
message	~ 500

Block Structure



Block Size in bytes	
id_{prev}	32
id_{ms}	32
id_{tip}	32
nonce	4
peer	65
message	~ 500

Block Structure

Peer chain — Previous block by the same peer

Nakamoto chain — Milestone block

Connectivity — Tip block by another peer

Block Size in bytes	
id_{prev}	32
id_{ms}	32
id_{tip}	32
nonce	4
~~peer~~	~~65~~
message	~ 500

$$\textbf{Overhead} = \frac{\textbf{Header}}{\textbf{Header} + \textbf{Message}}$$

1 Tx	2 Tx's	3 Tx's	4 Tx's
17%	9%	6%	5%

Proof of Work

$$\mathtt{B} = (\mathtt{id}_{\mathtt{prev}}, \mathtt{id}_{\mathtt{ms}}, \mathtt{id}_{\mathtt{tip}}, \mathtt{nonce}, \mathtt{message})$$

$$H(\mathtt{B}) = \underbrace{0 \cdots\cdots 0}_{\text{leading x bits are all 0}} * * * * * * *$$

Regular Block

regular block reward + Tx fee

$$H(\mathtt{B}) = \underbrace{0 \cdots\cdots\cdots\cdots 0}_{\text{leading y bits are all 0}} * * * *$$

Milestone Block

additional bonus*

Proof of Work

$$\mathtt{B} = (\mathtt{id}_{\mathtt{prev}}, \mathtt{id}_{\mathtt{ms}}, \mathtt{id}_{\mathtt{tip}}, \mathtt{nonce}, \mathtt{message})$$

$$H(\mathrm{B}) = \underbrace{0\cdots\cdots0}_{\text{leading x bits are all 0}} * * * * * * * $$

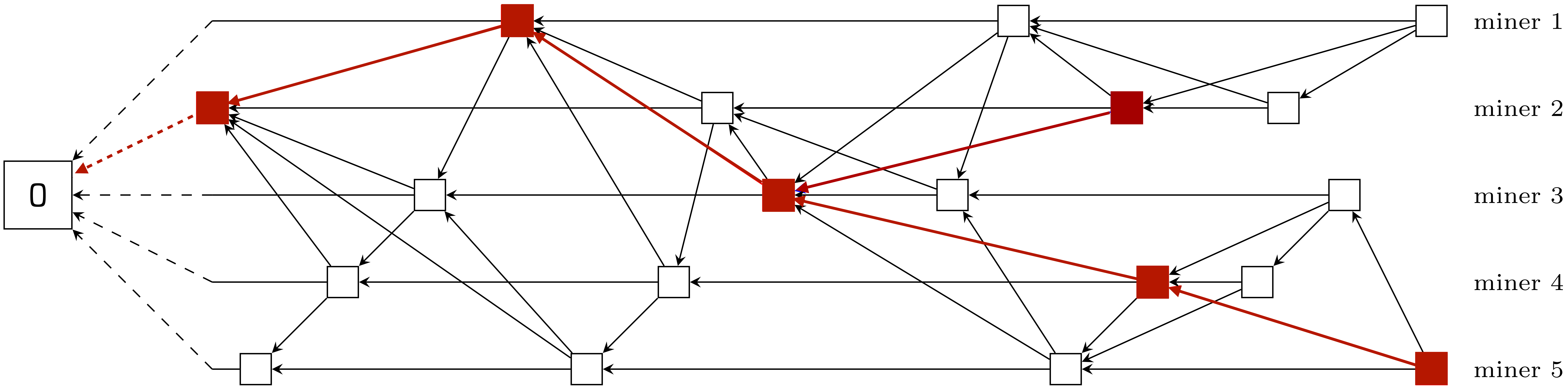
Regular Block — **regular block reward + Tx fee**

$$H(\mathrm{B}) = \underbrace{0\cdots\cdots\cdots\cdots0}_{\text{leading y bits are all 0}} * * * * $$

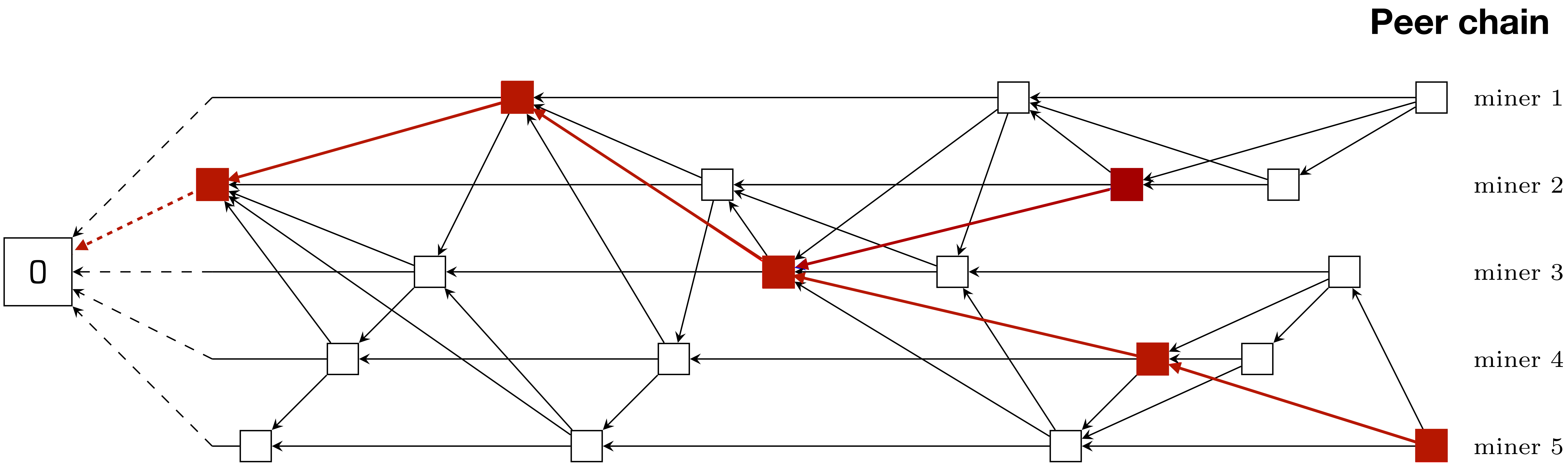
Milestone Block — **additional bonus***

Three pointers have to be specified before knowing the type!

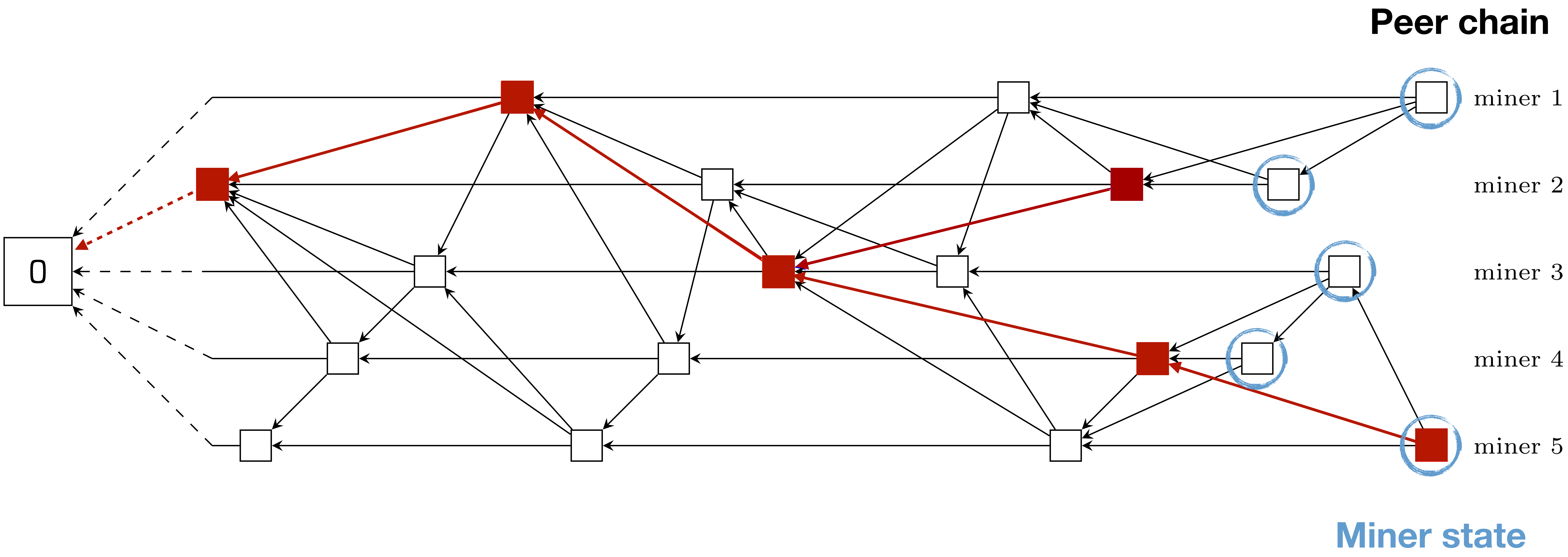
Structured DAG



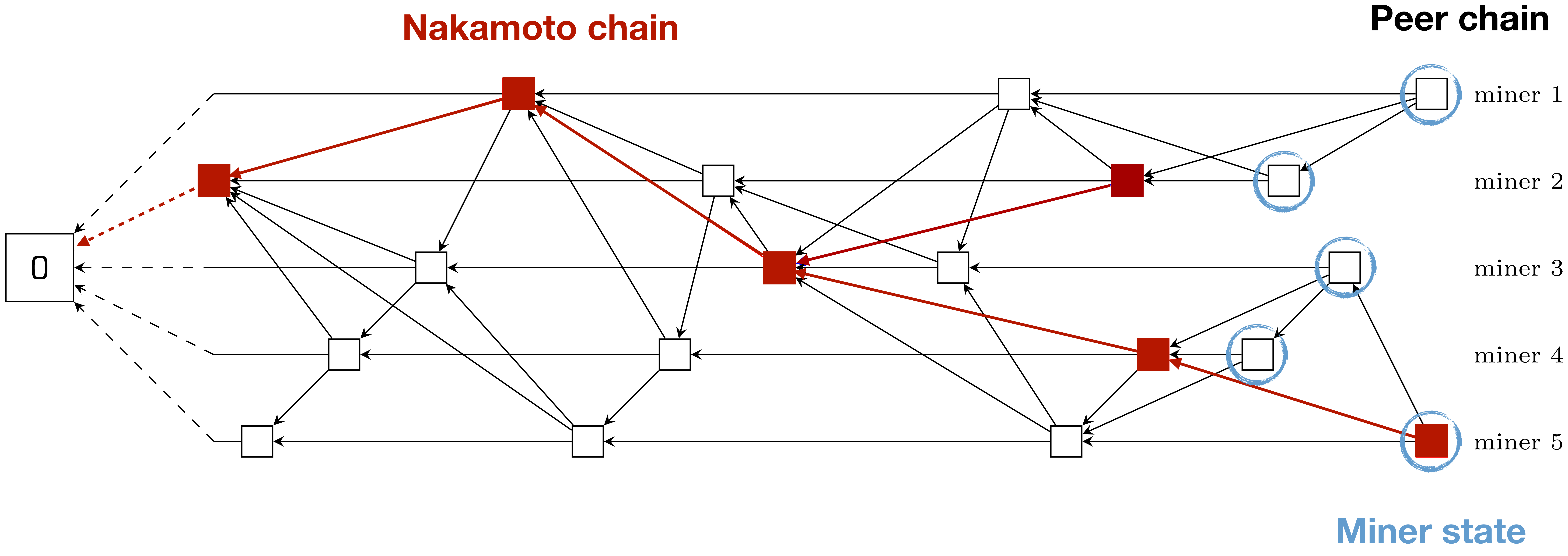
Structured DAG



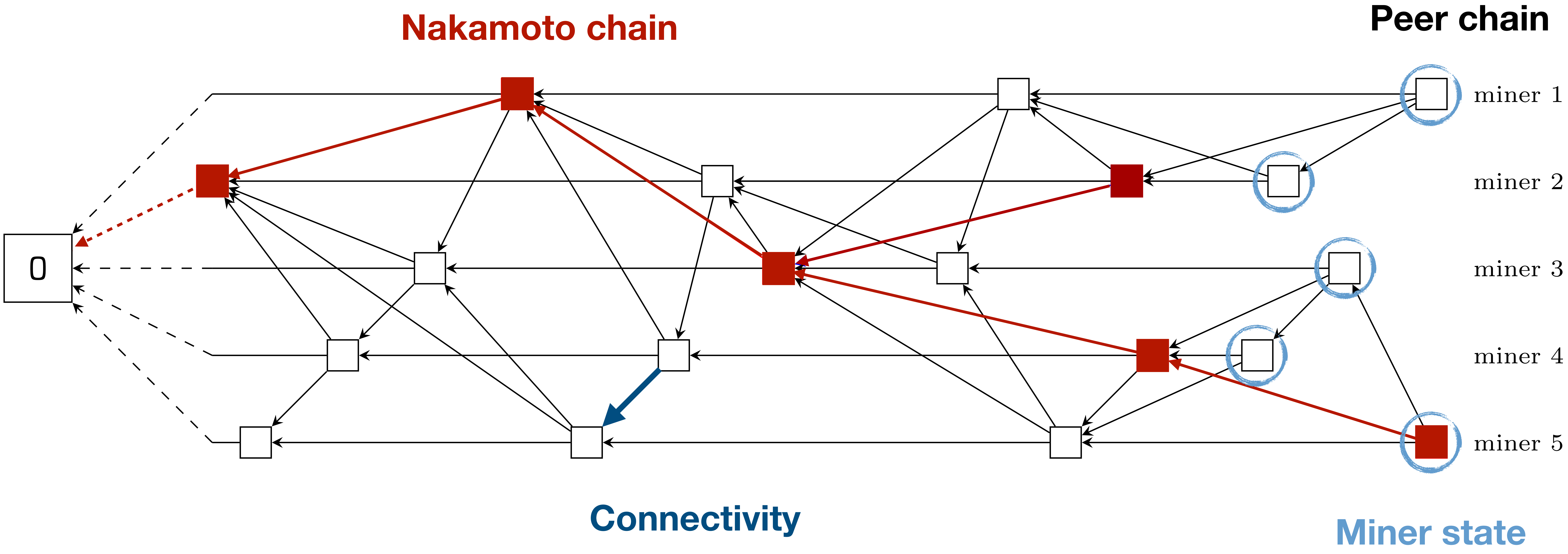
Structured DAG



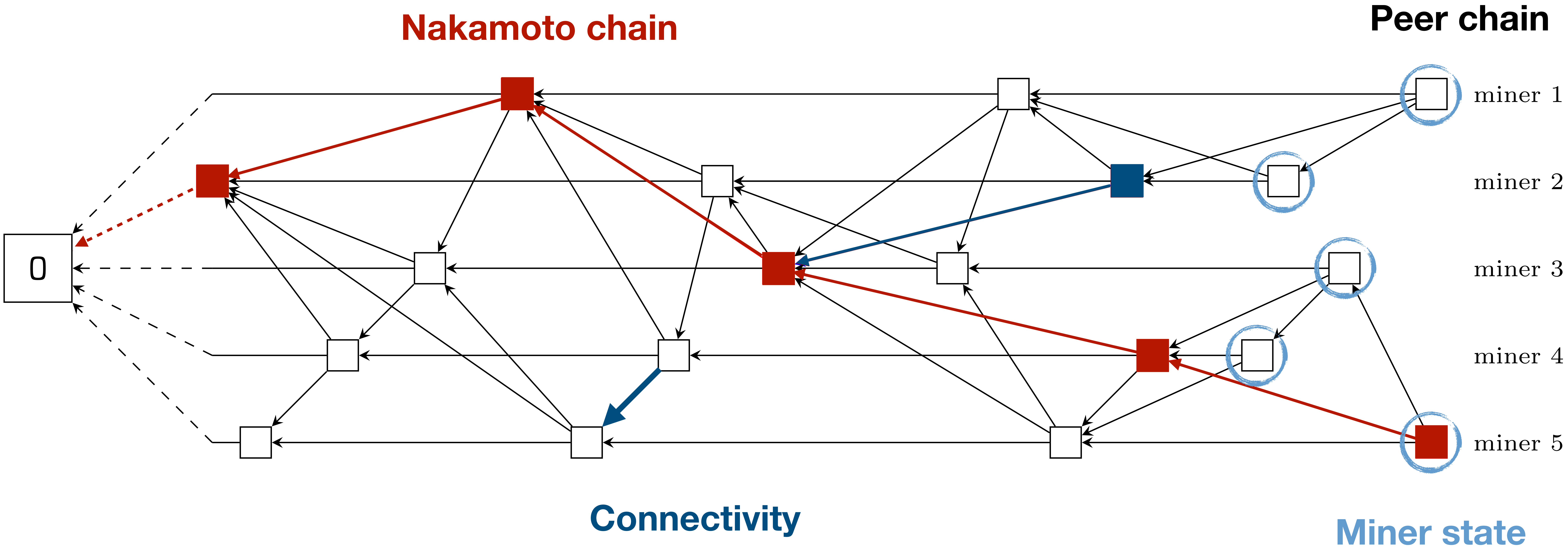
Structured DAG



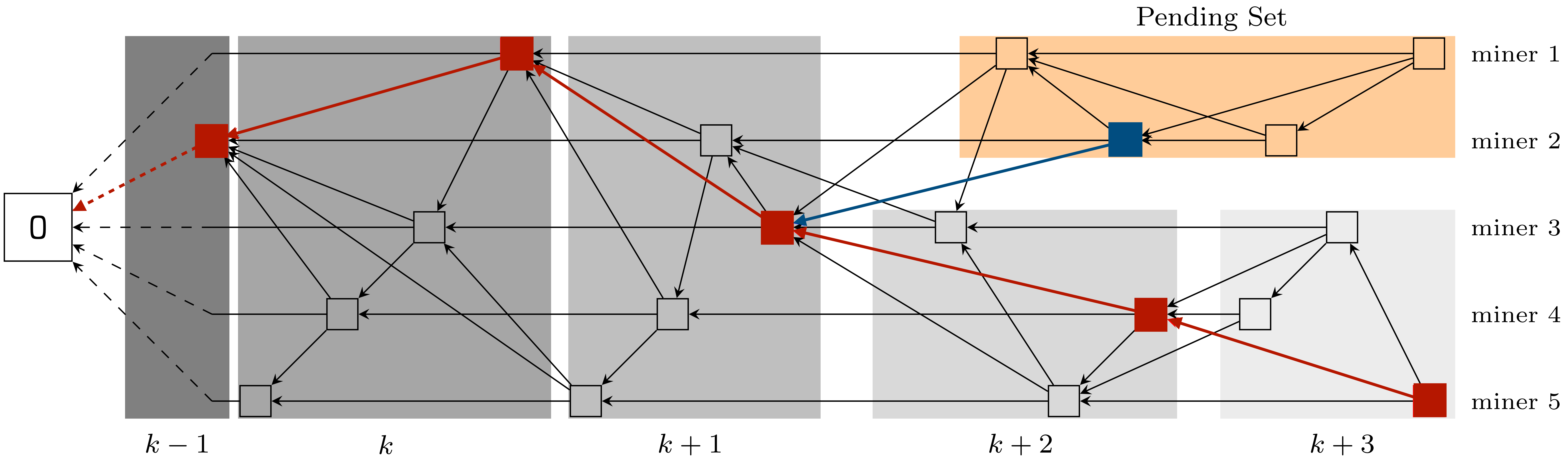
Structured DAG



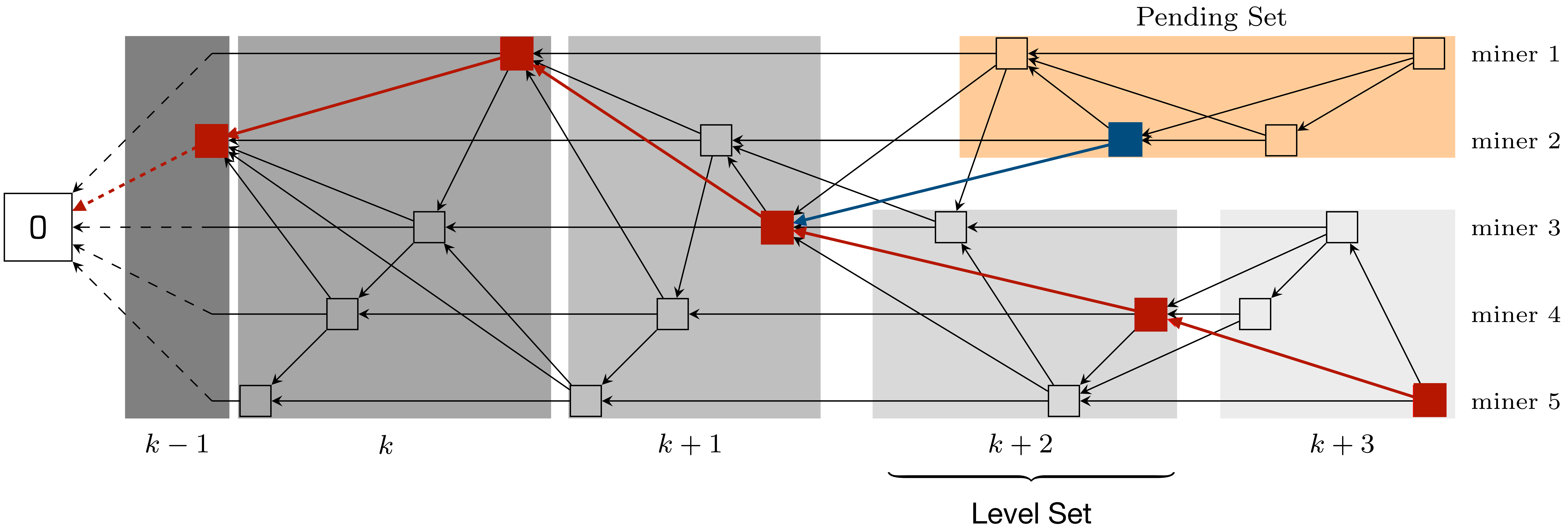
Structured DAG



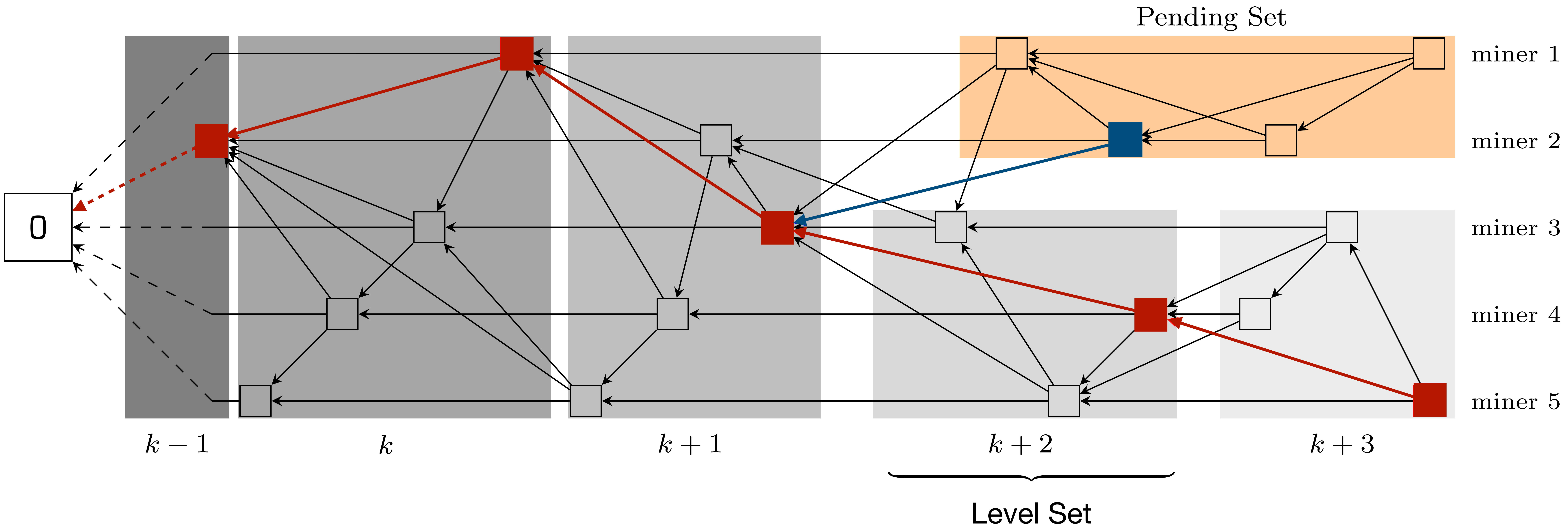
Structured DAG



Structured DAG

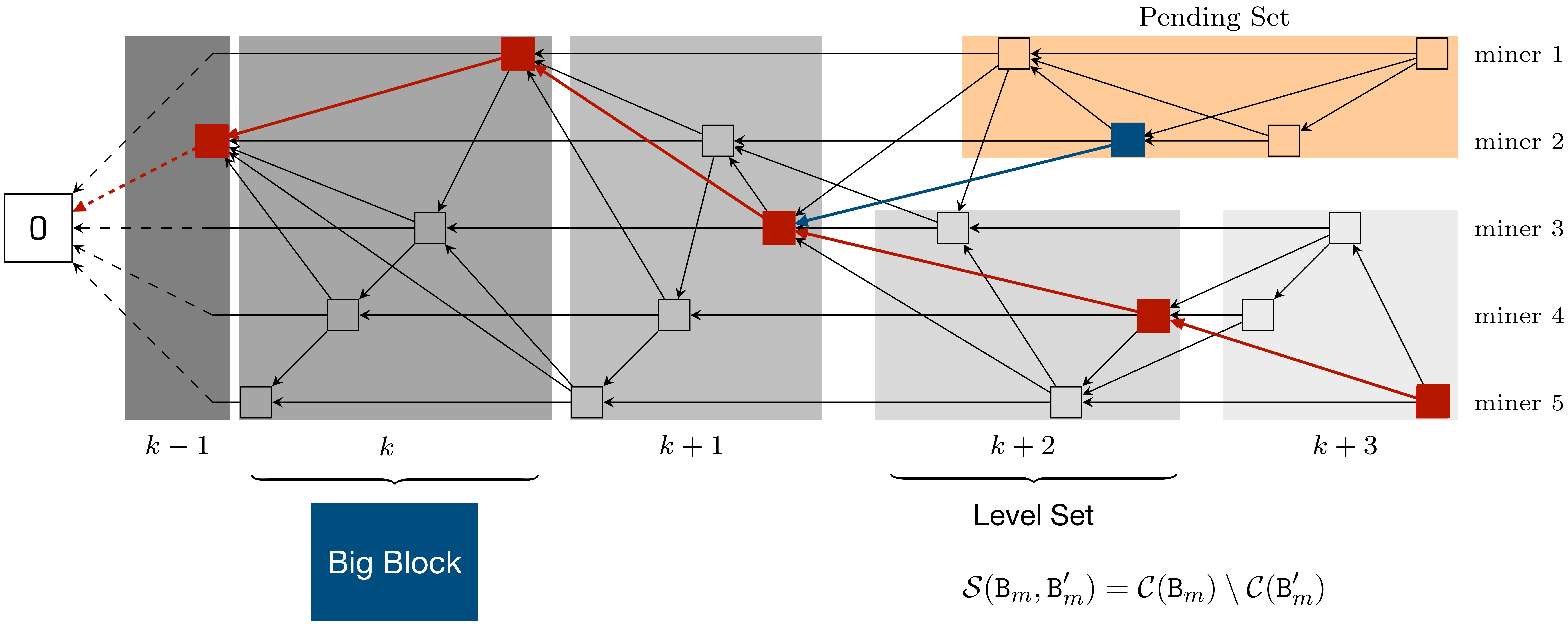


Structured DAG



$$\mathcal{S}(\mathtt{B}_m, \mathtt{B}'_m) = \mathcal{C}(\mathtt{B}_m) \setminus \mathcal{C}(\mathtt{B}'_m)$$

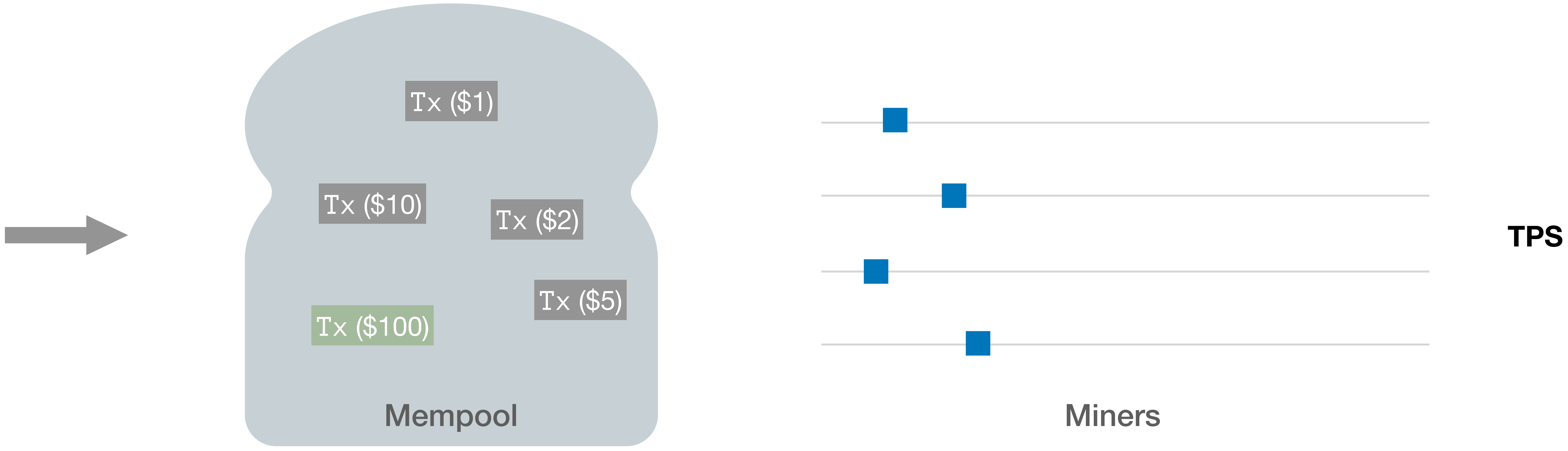
Structured DAG



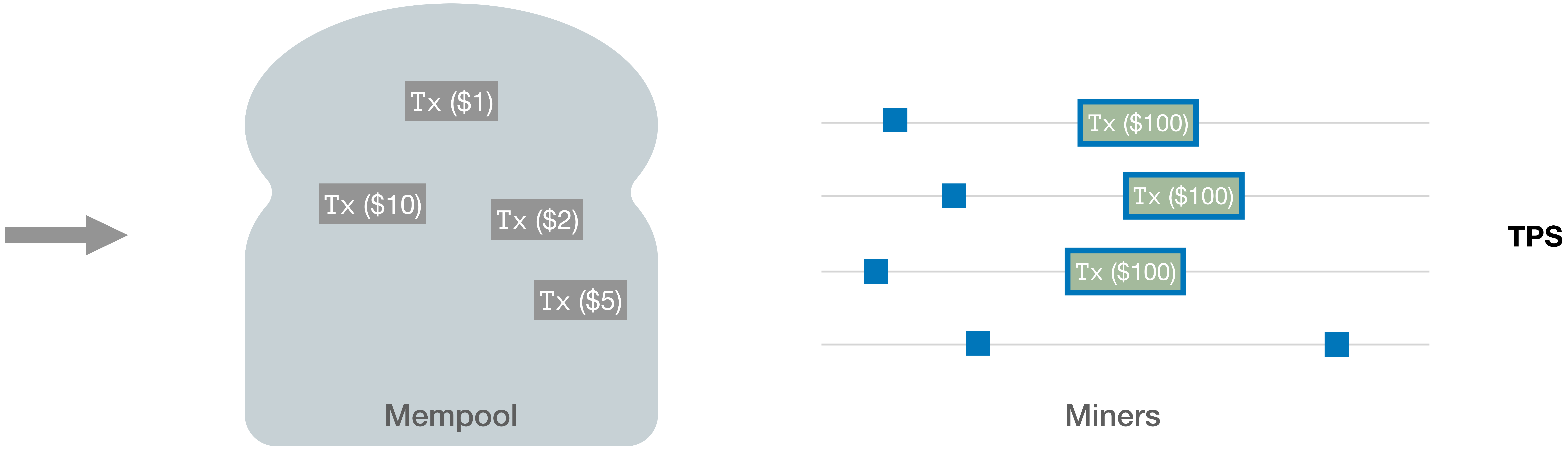
$$\mathcal{S}(\mathtt{B}_m, \mathtt{B}'_m) = \mathcal{C}(\mathtt{B}_m) \setminus \mathcal{C}(\mathtt{B}'_m)$$

Peers obtain different parts of the big block from different peers continuously over time.

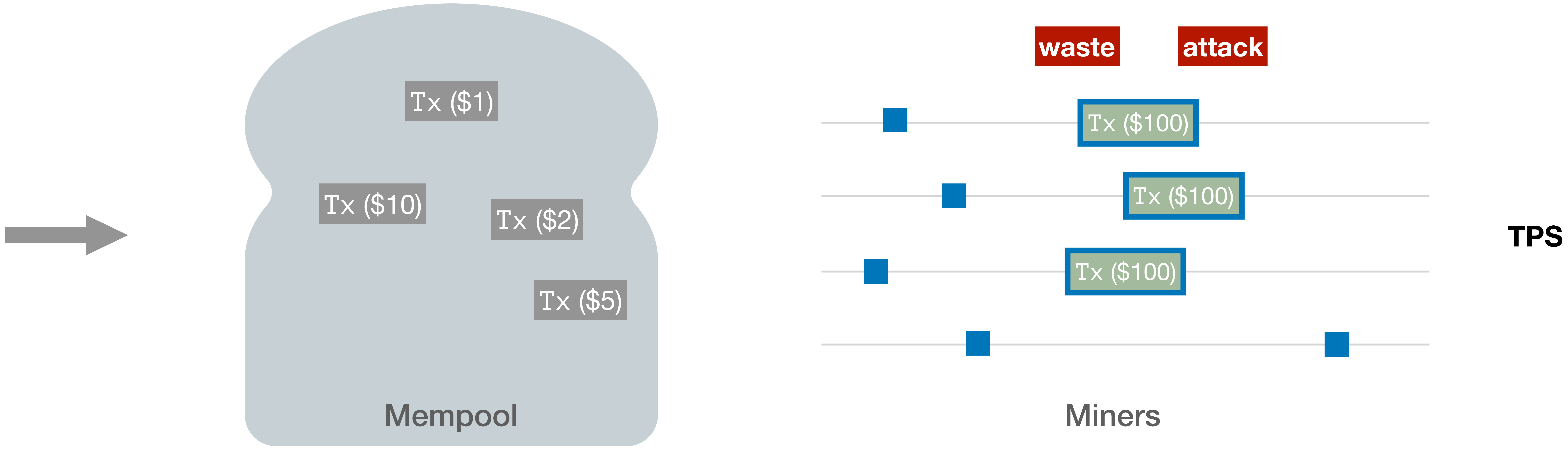
Transaction Assignment



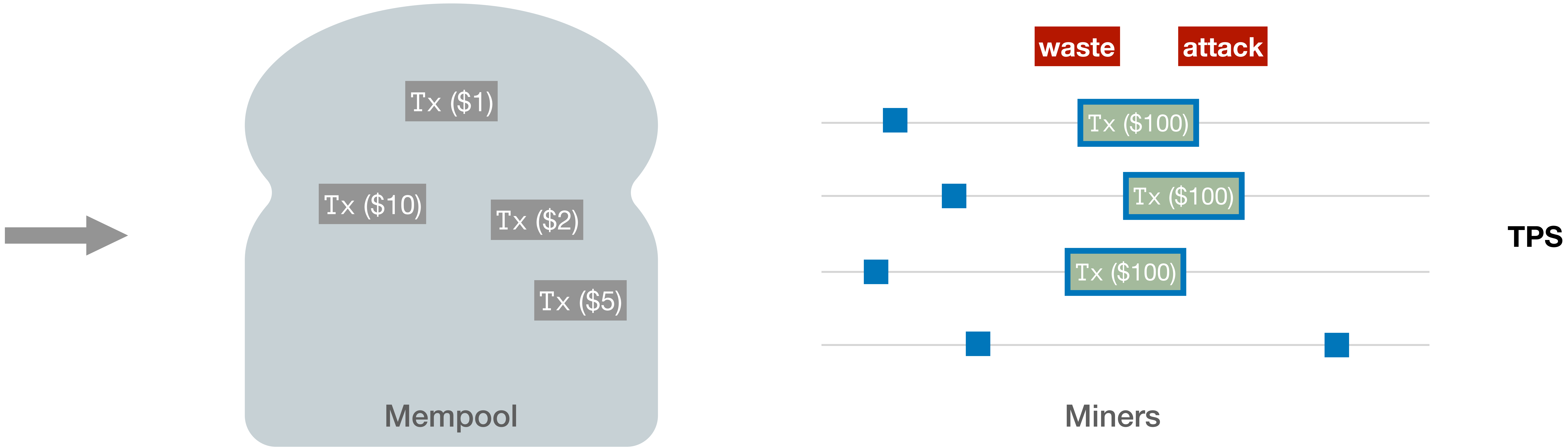
Transaction Assignment



Transaction Assignment

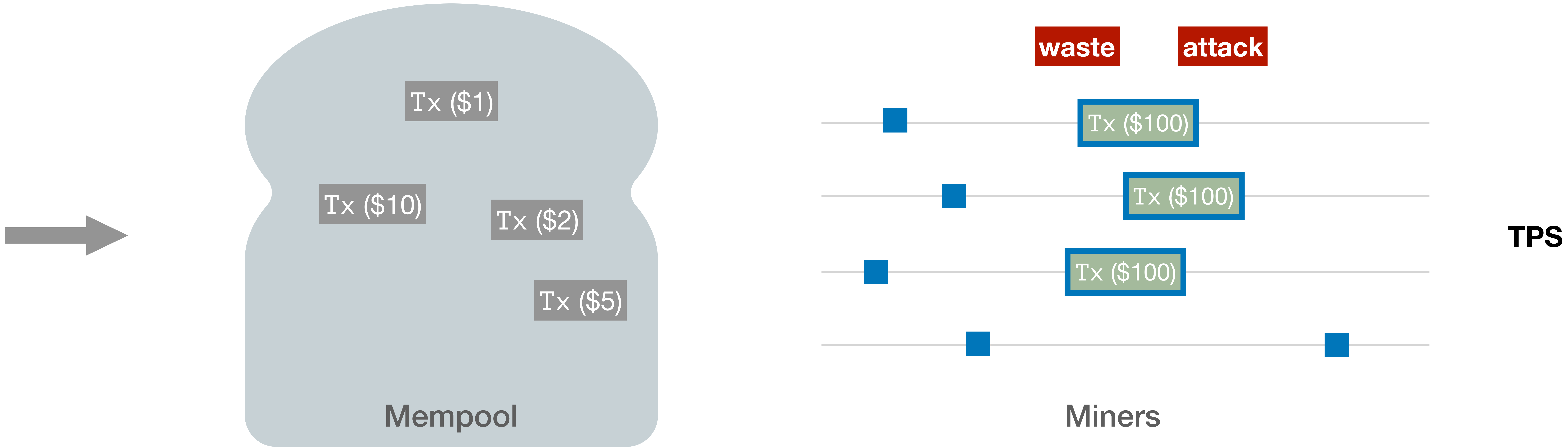


Transaction Assignment



$$. H(\text{miner's state}, \texttt{Tx}) < c \times \underbrace{\text{miner's hashing power}}_{q_i}$$

Transaction Assignment

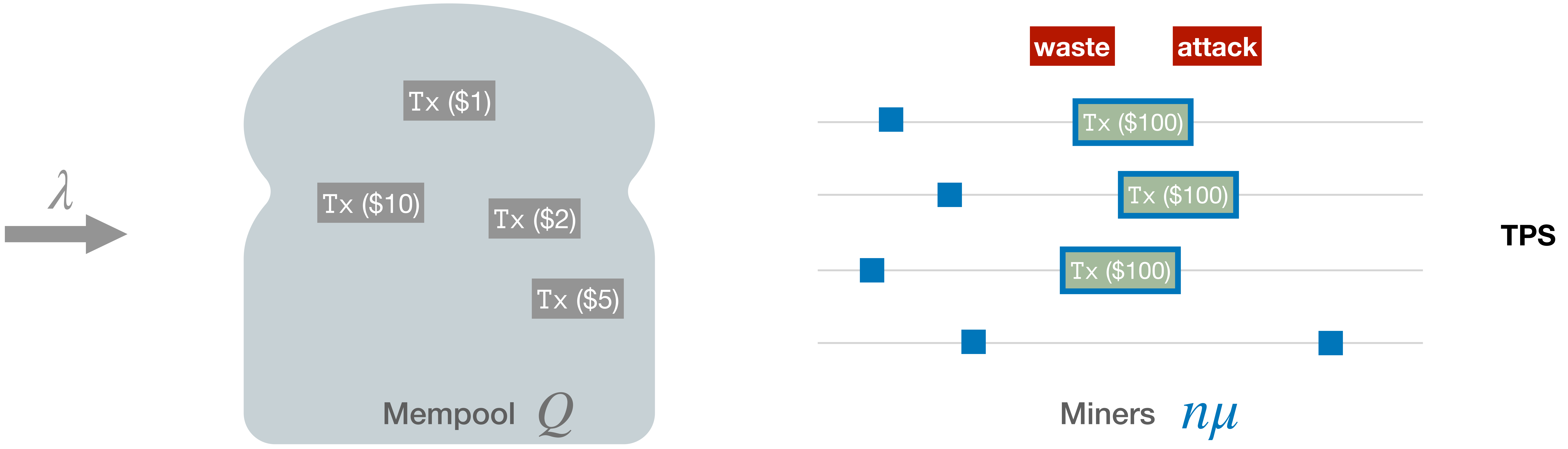


$$. H(\text{miner's state}, \mathtt{Tx}) < c \times \underbrace{\text{miner's hashing power}}_{q_i}$$

verifiable

consensus

Transaction Assignment



$$. H(\text{miner's state}, \mathtt{Tx}) < c \times \underbrace{\text{miner's hashing power}}_{q_i}$$

verifiable

consensus

Protocol

RECEIVING A BLOCK

Suppose Alice has a local DAG

$$\mathcal{G}_a = \mathcal{C}(\mathtt{B}_m)$$

1. Download if height is too small
2. Solidify + topological sort
3. Add blocks one by one

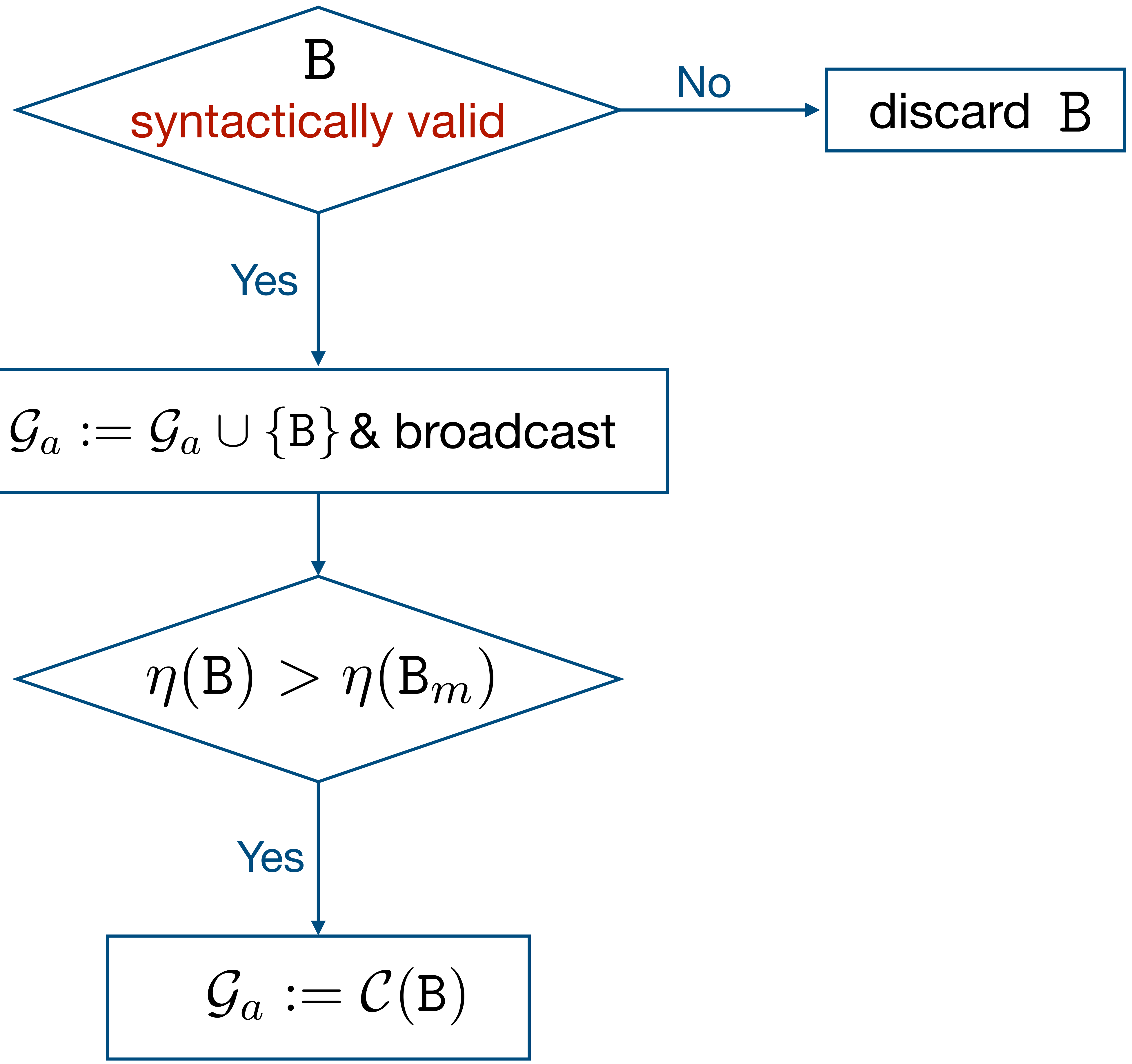
Protocol

RECEIVING A BLOCK

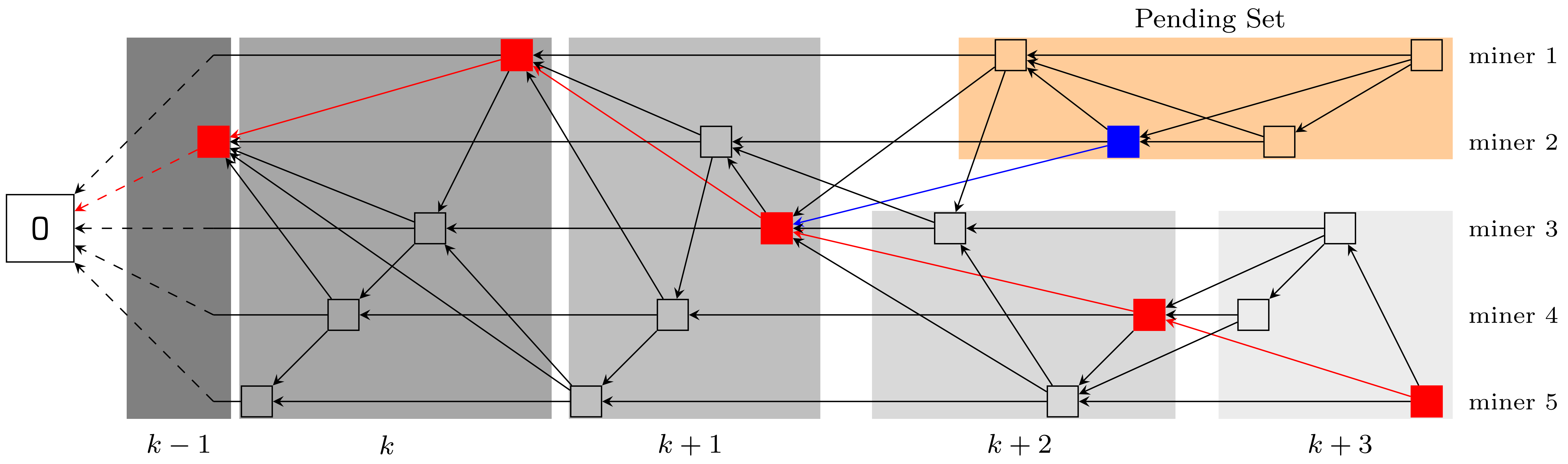
Suppose Alice has a local DAG

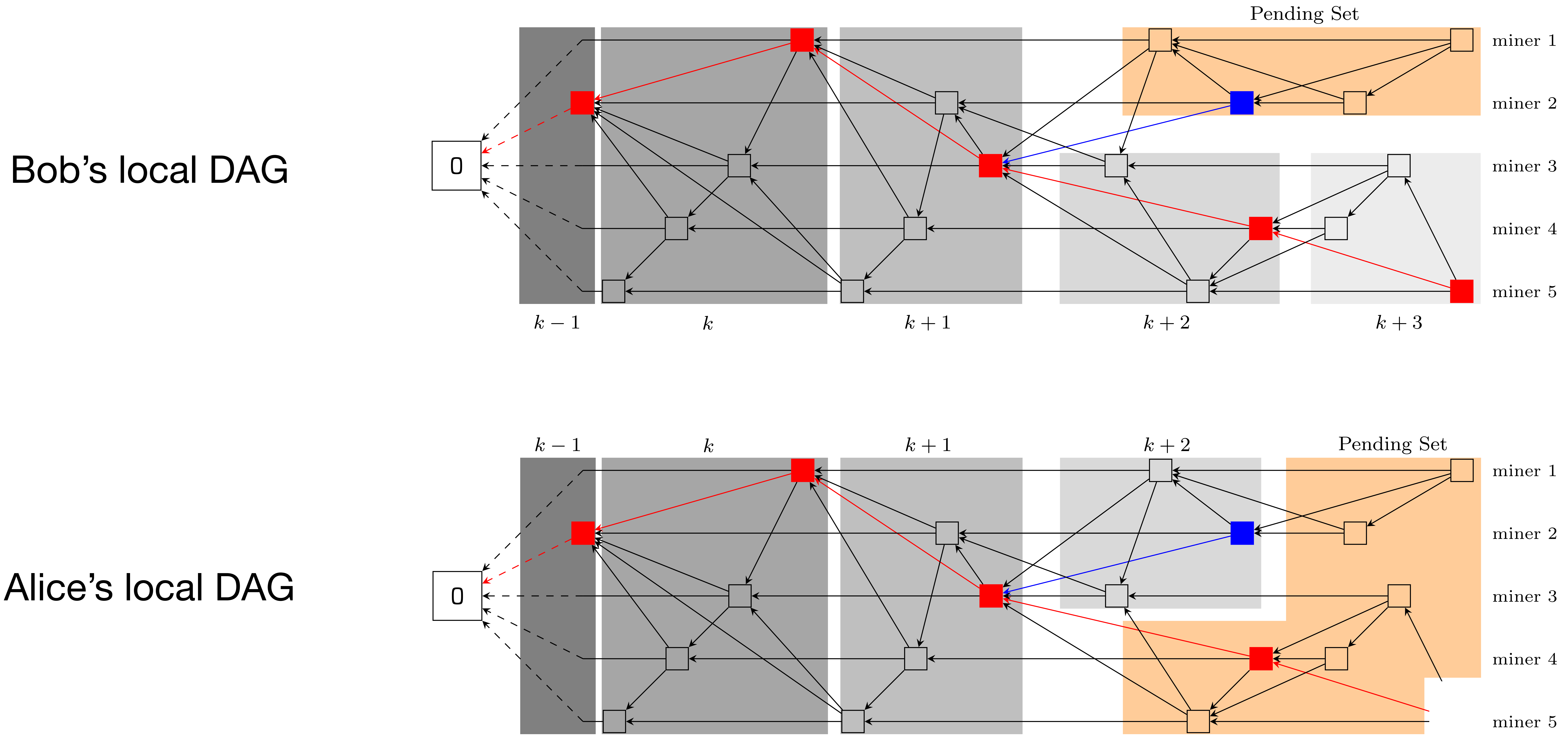
$$\mathcal{G}_a = \mathcal{C}(\mathtt{B}_m)$$

1. Download if height is too small
2. Solidify + topological sort
3. Add blocks one by one

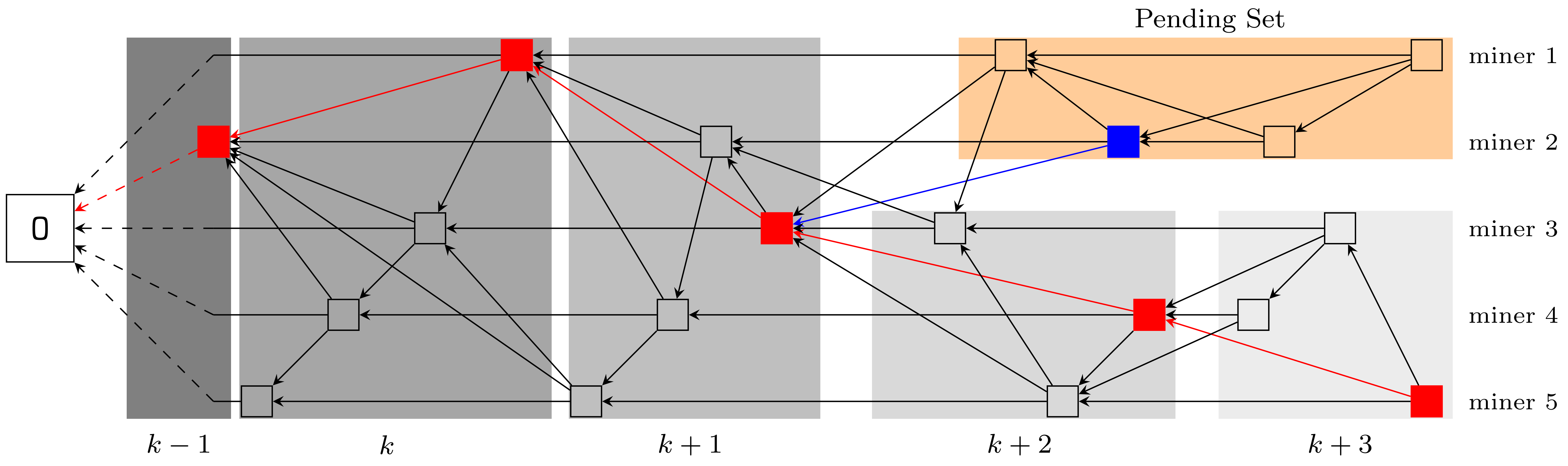


Bob's local DAG

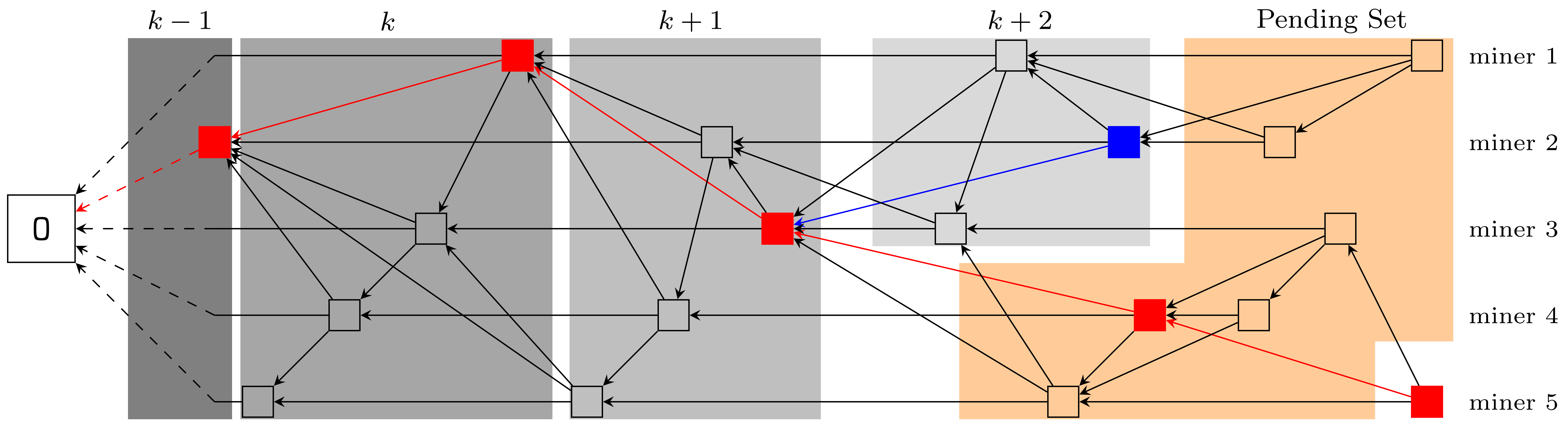



Bob's local DAG
Pending Set
miner 1
miner 2
miner 3
miner 4
miner 5
0
k − 1
k
k + 1
k + 2
k + 3
Alice's local DAG
k − 1
k
k + 1
k + 2
Pending Set
miner 1
miner 2
miner 3
miner 4
miner 5
0

Bob’s local DAG



Alice’s local DAG



Protocol

CREATING A BLOCK

Block Size in bytes	
	32
	32
	32
	4
	~ 500

Protocol

CREATING A BLOCK

1. Find a transaction Alice can process

Block Size in bytes	
	32
	32
	32
	4
`message`	~ 500

Protocol

CREATING A BLOCK

1. Find a transaction Alice can process
2. Prepare **three pointers**

Block Size in bytes	
id_{prev}	32
id_{ms}	32
id_{tip}	32
	4
message	~ 500

Protocol

CREATING A BLOCK

1. Find a transaction Alice can process
2. Prepare **three pointers**
3. Solve the cryptographic puzzle

Block Size in bytes	
$\texttt{id}_{\texttt{prev}}$	32
$\texttt{id}_{\texttt{ms}}$	32
$\texttt{id}_{\texttt{tip}}$	32
`nonce`	4
`message`	~ 500

Protocol

CREATING A BLOCK

1. Find a transaction Alice can process
2. Prepare **three pointers**
3. Solve the cryptographic puzzle
4. Broadcast the block

Block Size in bytes	
$\mathtt{id}_{\mathtt{prev}}$	32
$\mathtt{id}_{\mathtt{ms}}$	32
$\mathtt{id}_{\mathtt{tip}}$	32
`nonce`	4
`message`	~ 500

syntactically valid

Protocol

CREATING A BLOCK

1. Find a transaction Alice can process
2. Prepare **three pointers**
3. Solve the cryptographic puzzle
4. Broadcast the block

Block Size in bytes	
id_{prev}	32
id_{ms}	32
id_{tip}	32
nonce	4
message	~ 500

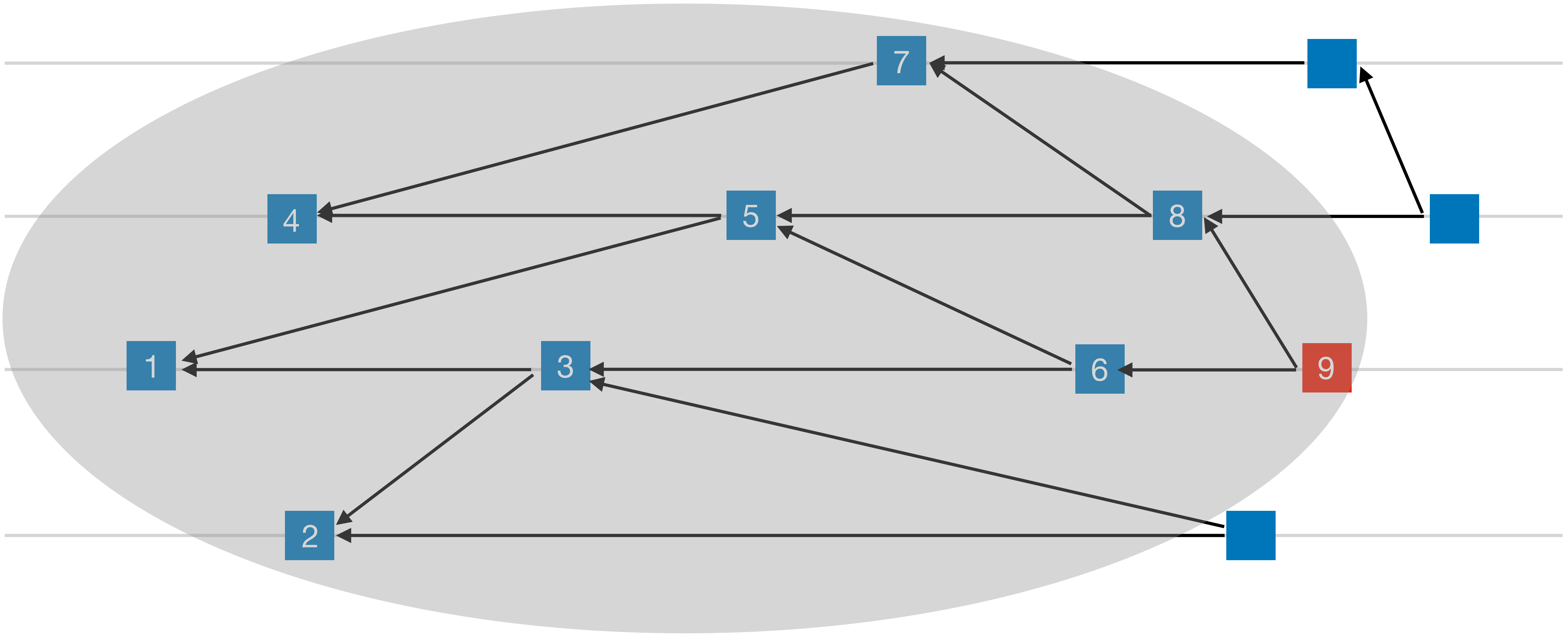
syntactically valid

Peer chain forked — ☐ What if not pointing to the miner's previous block

Consensus failed — ☐ What if not pointing to the most recent milestone

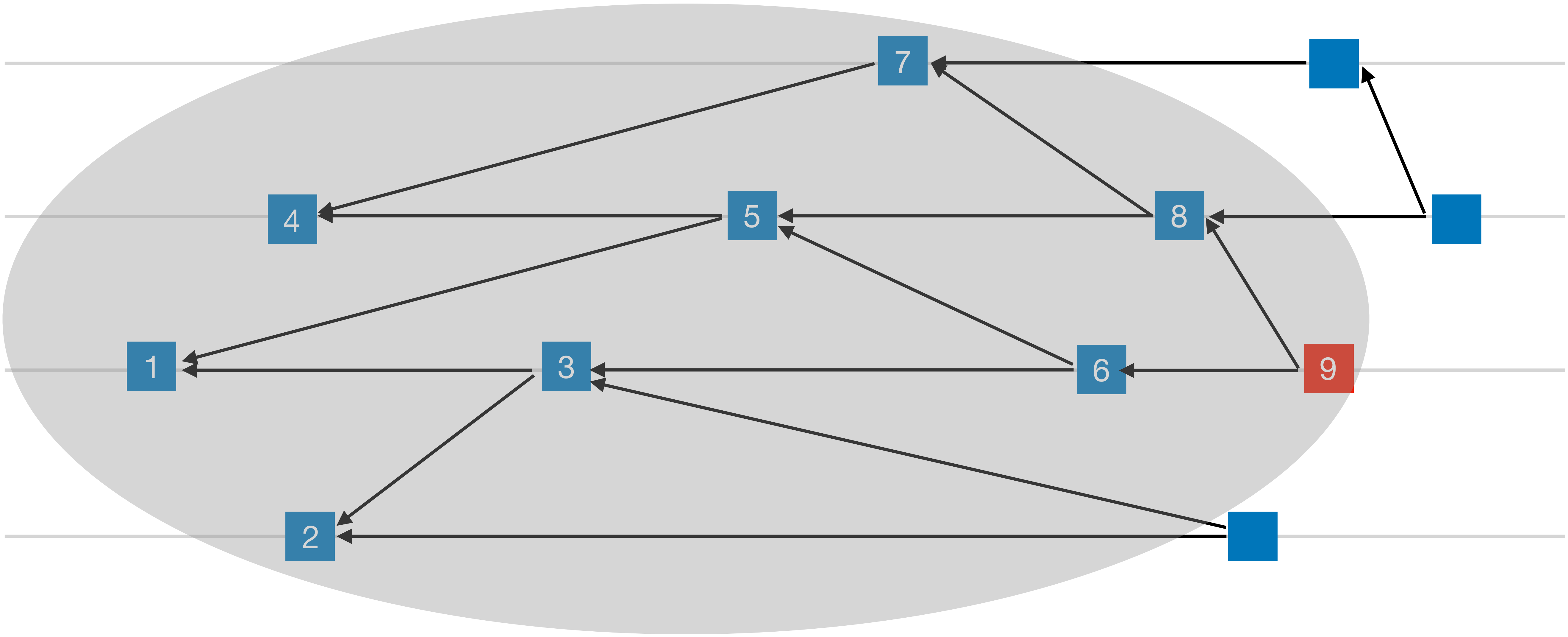
Connectivity broken — ☐ What if not pointing to a recent regular block by others

Building a Ledger



Post-order DFS

Building a Ledger



Post-order DFS

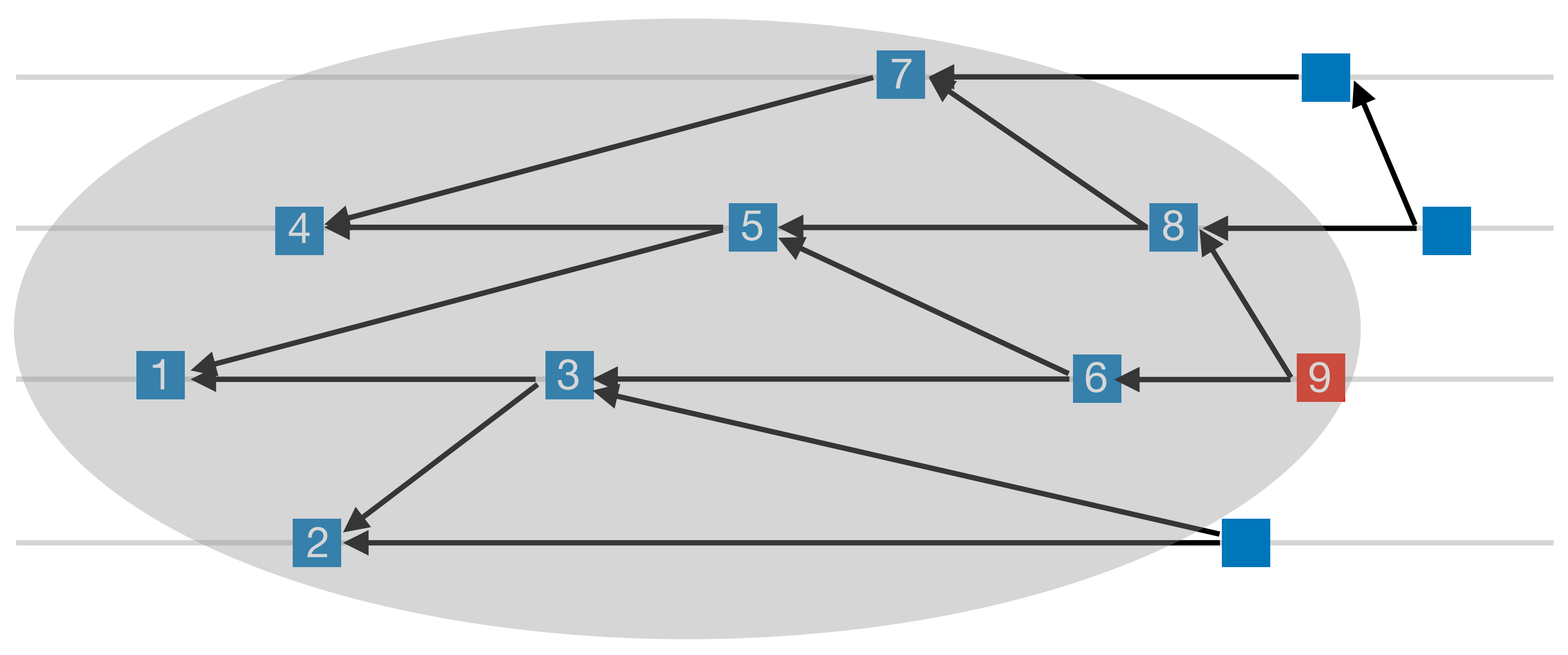
✓ chronological order of blocks on a peer chain

✓ approximate chronological order of all blocks

Building a Ledger

Order all transactions
in a level set

$\mathrm{Tx}_1, \mathrm{Tx}_2, \mathrm{Tx}_3, \ldots$



Post-order DFS

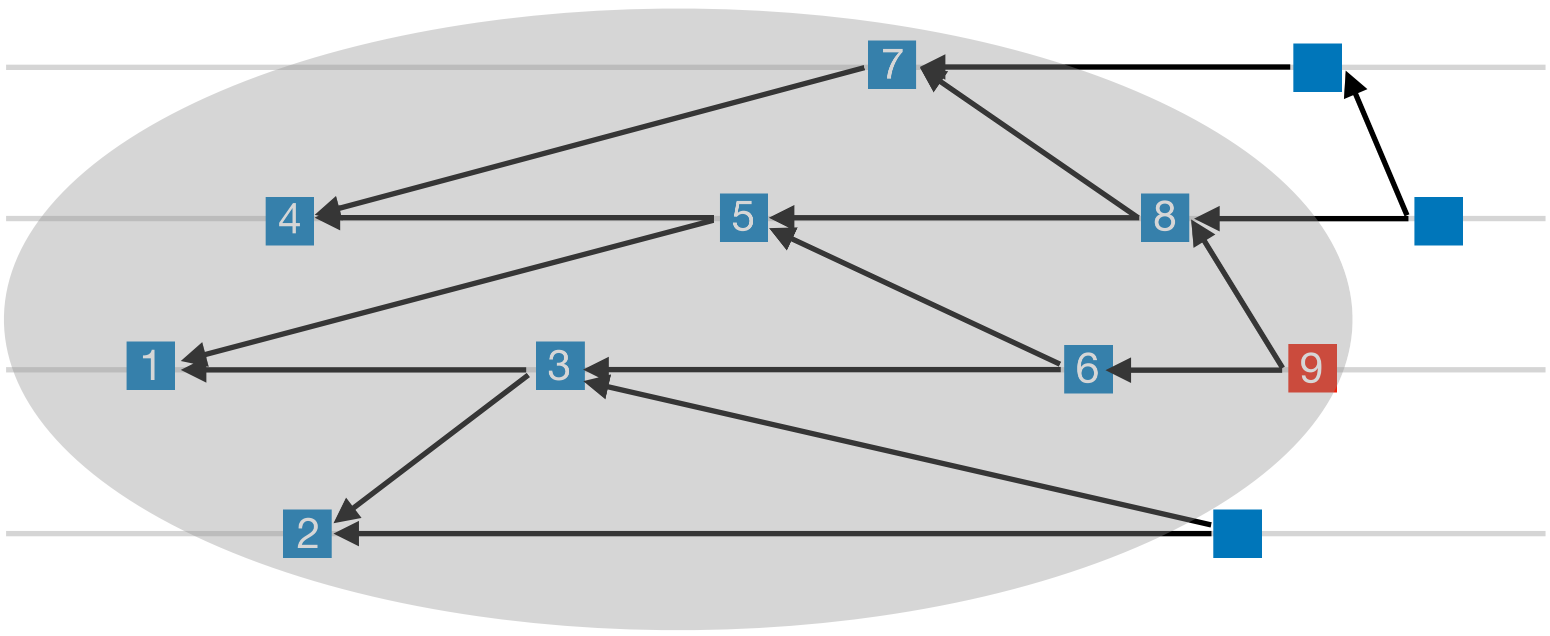
✓ chronological order of blocks on a peer chain

✓ approximate chronological order of all blocks

Building a Ledger

Order all transactions in a level set

$\texttt{Tx}_1, \texttt{Tx}_2, \texttt{Tx}_3, \ldots$



Post-order DFS

✓ chronological order of blocks on a peer chain

✓ approximate chronological order of all blocks

$\texttt{Tx}_{k+1}$ valid

Yes:

$$\mathcal{L}_{k+1} = \mathcal{L}_{k+1} \cup (\texttt{Tx}_{k+1}),$$
$$\mathcal{U}_{k+1} = \mathcal{U}_k \cup \{\text{outputs from } \texttt{Tx}_{k+1}\} \setminus \{\text{inputs from } \texttt{Tx}_{k+1}\}$$

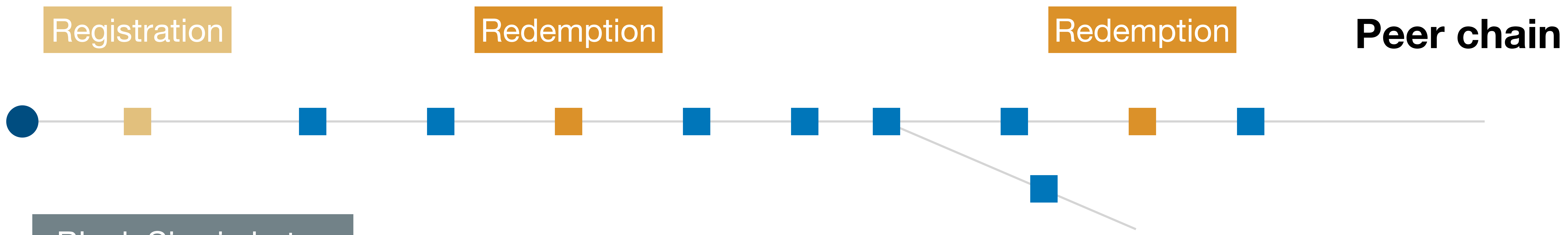
No:

$$(\mathcal{L}_{k+1}, \mathcal{U}_{k+1}) = (\mathcal{L}_k, \mathcal{U}_k)$$

Reward Scheme

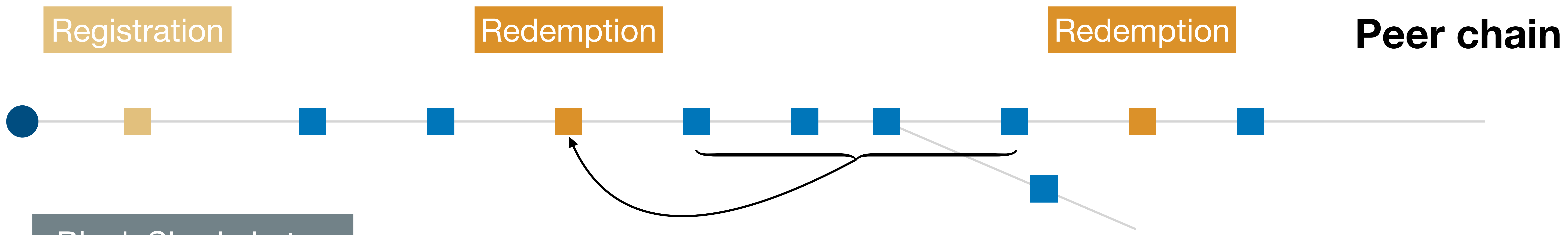
Block Size in bytes	
id_{prev}	32
id_{ms}	32
id_{tip}	32
nonce	4
~~peer~~	~~65~~
message	~ 500

Reward Scheme



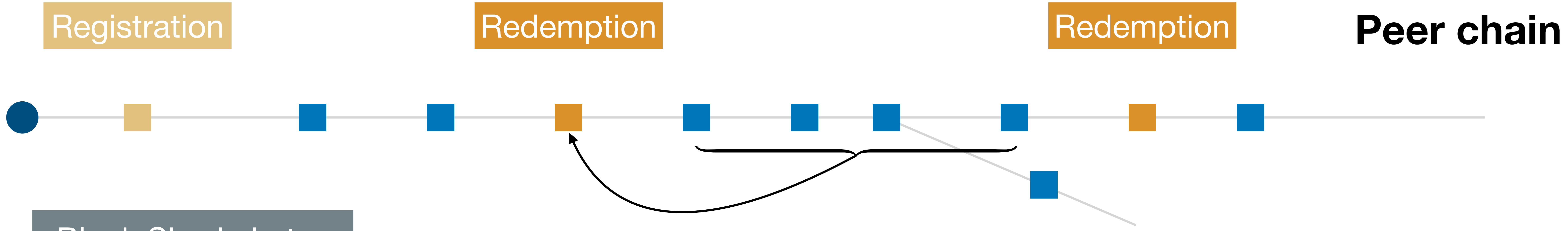
Block Size in bytes	
$\mathtt{id}_{\mathtt{prev}}$	32
$\mathtt{id}_{\mathtt{ms}}$	32
$\mathtt{id}_{\mathtt{tip}}$	32
nonce	4
~~peer~~	~~65~~
message	~ 500

Reward Scheme



Block Size in bytes	
id_{prev}	32
id_{ms}	32
id_{tip}	32
nonce	4
~~peer~~	~~65~~
message	~ 500

Reward Scheme



Block Size in bytes	
id_{prev}	32
id_{ms}	32
id_{tip}	32
nonce	4
~~peer~~	~~65~~
message	~ 500

✓ No peer id/sig to save message space

✓ No coinbase to reduce the number of UTXOs

Reward Scheme

Type	Status	Transaction	Reward	Bonus
regular +	on peer chain	Valid	r + Tx fee	0
regular +	on peer chain	Invalid	r	0
regular +	forked	Valid	0	0
regular +	forked	Invalid	0	0
milestone	longest MS chain	Valid	r + Tx fee	% × r × level set
milestone	longest MS chain	Invalid	r	% × r × level set

Reward Scheme

Type	Status	Transaction	Reward	Bonus
regular +	on peer chain	Valid	r + Tx fee	0
regular +	on peer chain	Invalid	r	0
regular +	forked	Valid	0	0
regular +	forked	Invalid	0	0
milestone	longest MS chain	Valid	r + Tx fee	% × r × level set
milestone	longest MS chain	Invalid	r	% × r × level set

☑ not to fork peer chain

Reward Scheme

Type	Status	Transaction	Reward	Bonus
regular +	on peer chain	Valid	r + Tx fee	0
regular +	on peer chain	Invalid	r	0
regular +	forked	Valid	0	0
regular +	forked	Invalid	0	0
milestone	longest MS chain	Valid	r + Tx fee	% × r × level set
milestone	longest MS chain	Invalid	r	% × r × level set

- ☑ try luck on another block
- ☑ not to fork peer chain

Reward Scheme

Type	Status	Transaction	Reward	Bonus
regular +	on peer chain	Valid	r + Tx fee	0
regular +	on peer chain	Invalid	r	0
regular +	forked	Valid	0	0
regular +	forked	Invalid	0	0
milestone	longest MS chain	Valid	r + Tx fee	% × r × level set
milestone	longest MS chain	Invalid	r	% × r × level set

☑ try luck on another block

☑ not to fork peer chain

☑ try best to make your milestone on the longest chain

Reward Scheme

Type	Status	Transaction	Reward	Bonus
regular +	on peer chain	Valid	r + Tx fee	0
regular +	on peer chain	Invalid	r	0
regular +	forked	Valid	0	0
regular +	forked	Invalid	0	0
milestone	longest MS chain	Valid	r + Tx fee	% × r × level set
milestone	longest MS chain	Invalid	r	% × r × level set

- ☑ try luck on another block (rows 1–2)
- ☑ not to fork peer chain (rows 3–4)
- ☑ contribute to connectivity (rows 5–6)
- ☑ try best to make your milestone on the longest chain (Bonus column)

Model Analysis

$. H(\text{miner's state}, \texttt{Tx}) < c \times \text{miner's hashing power}$

$$\mathbb{P}(\texttt{Tx} \text{ is not workable for any miner}) \approx e^{-c}$$

Model Analysis

Waste

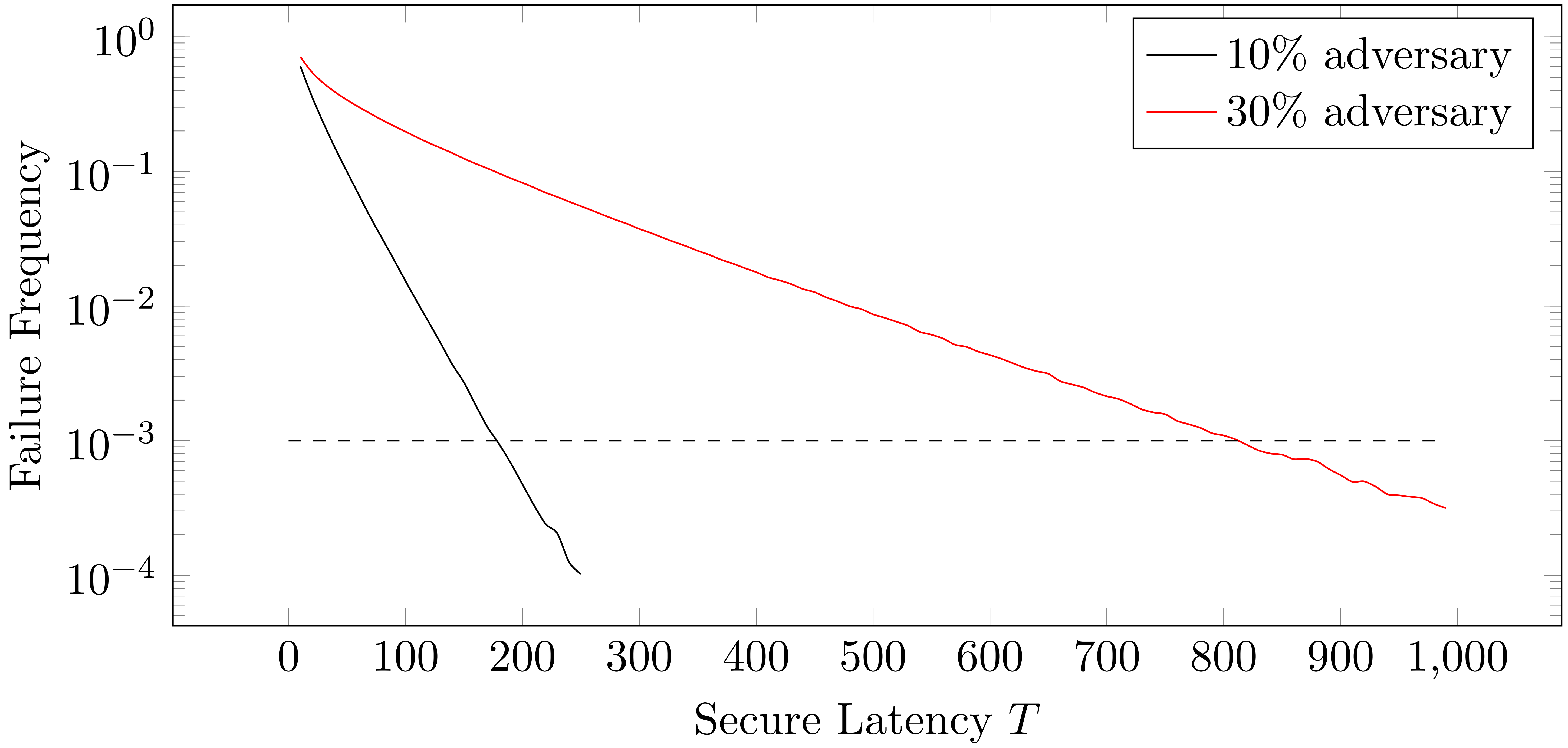
$$\theta(c) = \frac{(1 - e^{-\mu\bar{t}})\mu c\bar{t}}{1 + (1 - e^{-\mu\bar{t}})\mu c\bar{t}}$$

Queueing latency

$$\frac{1}{c}\frac{1}{\rho\mu}\ln\left(\frac{1}{1 - \frac{\rho}{1-\theta(c)}}\right)$$

Infection delay

$$\frac{2 + 2\ln(n)}{\mu} + \frac{1}{np\mu}$$



Performance

System parameters

Partition factor *c*	Block generation speed	MS interval	Avg. # of blocks per level set
0.01	1200 blocks/second	10 seconds	12000

Performance

System parameters

Partition factor *c*	Block generation speed	MS interval	Avg. # of blocks per level set
0.01	1200 blocks/second	10 seconds	12000

Modeling assumptions

Tx arrival rate	Percentage of malicious hashing power
1000	30%

Performance

System parameters

Partition factor *c*	Block generation speed	MS interval	Avg. # of blocks per level set
0.01	1200 blocks/second	10 seconds	12000

Modeling assumptions

Tx arrival rate	Percentage of malicious hashing power
1000	30%

Performance

Queueing latency	Infection delay	Security latency	Wasted capacity
188 seconds	23 seconds	810 seconds	1.7%

Summary

consensus reward algorithm

- ☑ TPS
- ☑ Latency: waiting + confirmation
- ☑ Concentration of mining power
- ☑ Transactions with little fees

Summary

consensus **reward** **algorithm** **economic model**

- ☑ TPS
- ☑ Latency: waiting + confirmation
- ☑ Concentration of mining power
- ☑ Transactions with little fees

- ☐ How to issue new coins: inflation, limited supply, …?
- ☐ How to incentivize people to provide storage and bandwidth service?

Summary

consensus | reward | algorithm | economic model | optimization

- [x] TPS
- [x] Latency: waiting + confirmation
- [x] Concentration of mining power
- [x] Transactions with little fees

- [ ] How to issue new coins: inflation, limited supply, …?
- [ ] How to incentivize people to provide storage and bandwidth service?
- [ ] How to adaptively adjust the number of blocks created per unit of time?

Consensus Mechanism Design based on Structured Directed Acyclic Graphs

Jiahao He[*], Guangju Wang[*], Guangyuan Zhang[*], and Jiheng Zhang[†]

[*†]DMAC Lab
[*†]Department of Industrial Engineering and Decision Analytics
[†]Department of Mathematics
[*†]The Hong Kong University of Science and Technology

January 9, 2019

Abstract

We introduce a structure for the directed acyclic graph (DAG) and a mechanism design based on that structure so that peers can reach consensus at large scale based on proof of work (PoW). We also design a mempool transaction assignment method based on the DAG structure to render negligible the probability that a transaction being processed by more than one miners. The result is a significant scale-up of the capacity without sacrificing security and decentralization.

Key words: consensus; directed acyclic graph; proof of work; transaction assignment.

arXiv.org

arXiv:1901.02755

GitHub

coming soon