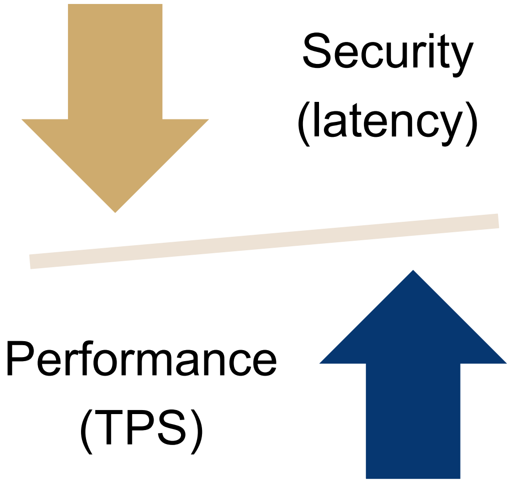
北京大學
PEKING UNIVERSITY

Dino: A Block Transmission Protocol with Low Bandwidth Consumption and Propagation Latency

Authors: Zhenxing Hu and Zhen Xiao
School of Computer Science, Peking University

Background and Motivation

- A general method to improve TPS is to increase block capacity to contain more transactions(TX), but it prolongs block propagation latency and degrades system security.
- The incompatible of TPS and block size is the concrete embodiment of the paradox of performance and security in blockchain.
- A best-of-both-world solution is to compress the block size while increasing the number of TXes in it.



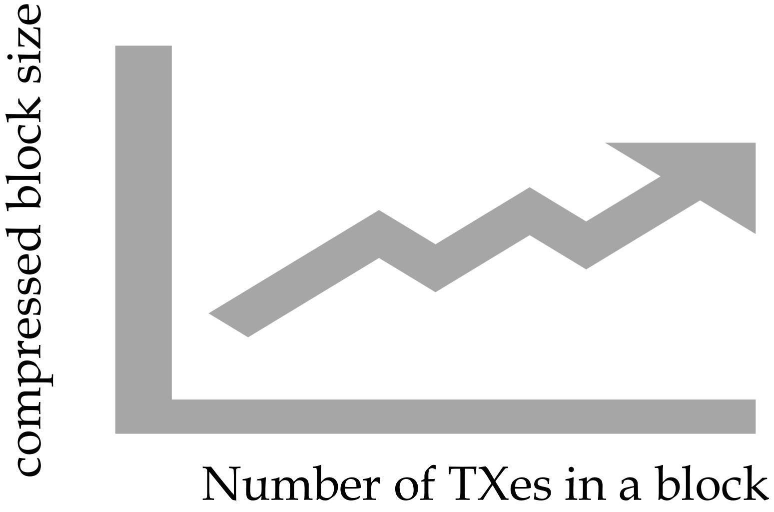
$$TPS = \frac{block\ size}{block\ interval} = \frac{number\ of\ TXes\ in\ one\ block}{block\ interval}$$

Background and Motivation

- Bitcoin (BTC) and Bitcoin Cash (BCH) have already deployed block compression methods, namely the Compact, Graphene, Xthinner, and XThin.

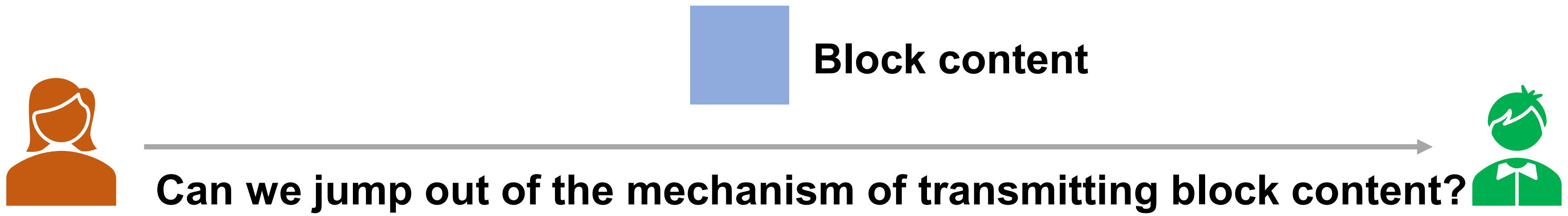
- Those approaches cannot avoid the increased block size due to increased transaction volume since their mechanisms depend on compressing the original block content.

- When the block capacity increases, the size of its associated compressed block also increases.



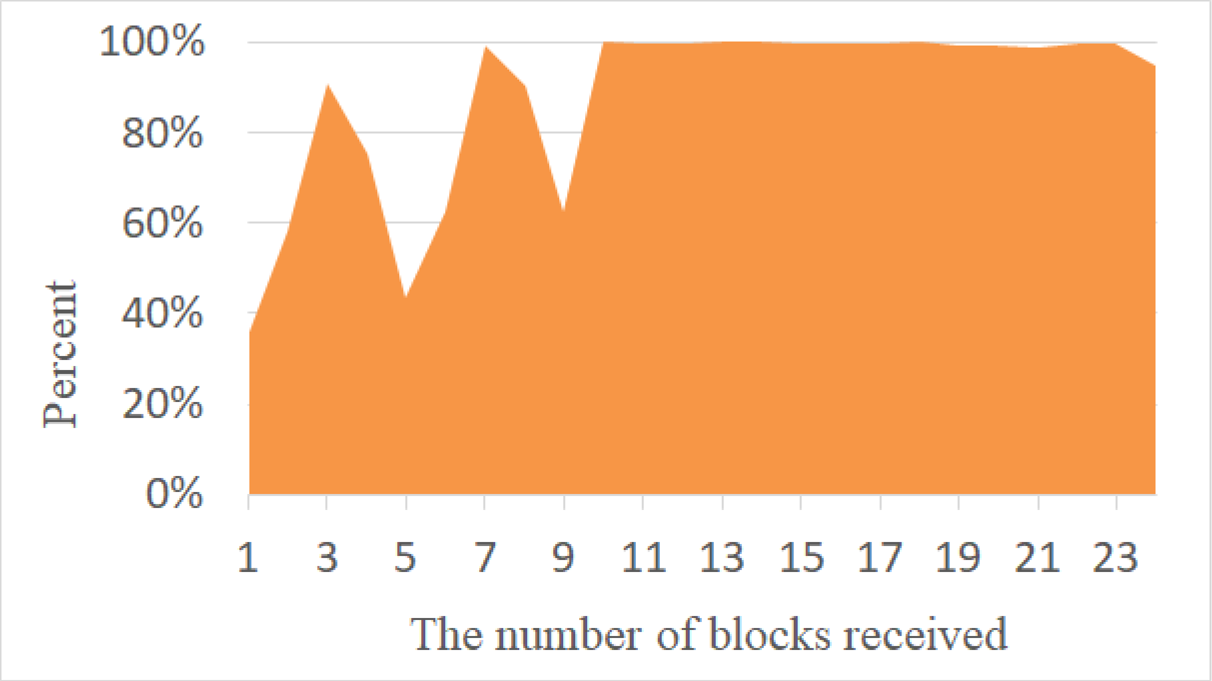
Our Solution

- Existing research is based on the assumption that a node always need to transmit block content to its neighbors.
- Can we jump out of the limitation of block compression?
- We propose a new block transmission protocol: Dino.
- It solves the drawback of the previous approaches by **transmitting block reconstruction rules** instead of compressed block content.



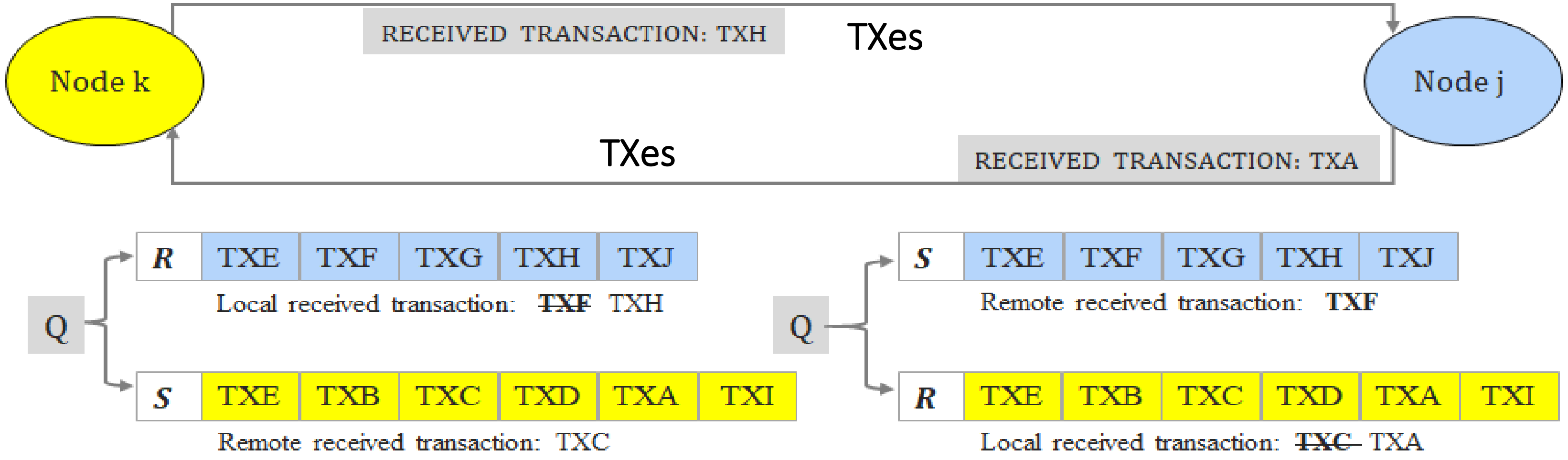
Dino’s Assumptions

- 1. **Almost all transactions** in a new block **already exists** in the mempools of other nodes.
- 2. **Miners** are **profit-oriented** and prefer to package transactions with higher fee rates into blocks.



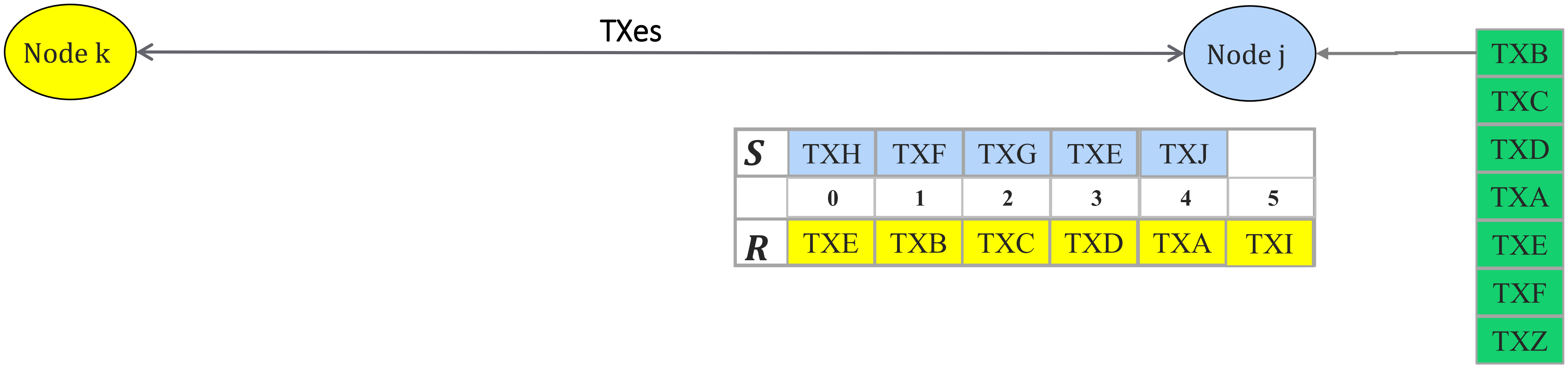
The Details of Dino

- In Dino, a node maintains a sending list(S) and a receiving list(R) for each of its neighbors.
- S stores the **TX hashes** it sends to the other node.
- R stores the **TX hashes** it receives from the other node.
- Periodically, each node sends a message to tell the other node the TXes it has received so far.
- Sender's S is equal to Receiver's R.



The Details of Dino

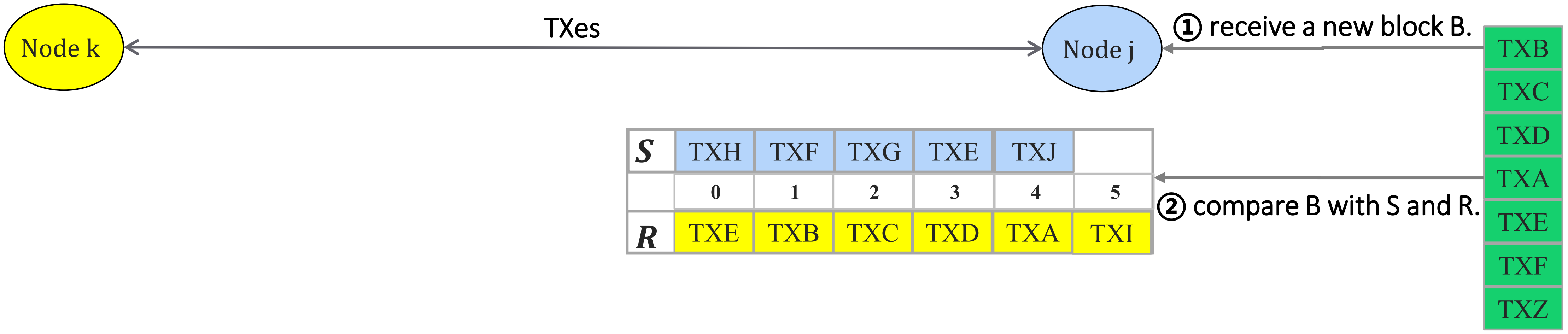
- Let $\boldsymbol{F}$ be the transaction package algorithm: $\boldsymbol{block} = \boldsymbol{F}(\boldsymbol{block\ capacity}, \boldsymbol{TX\ set})$, $\boldsymbol{F}$ chooses the maximum profitable TXes to generate a block.
- Suppose node j and k are neighbors in the Dino protocol. When node j receives a new block B and wants to send it to node k.



S	TXH	TXF	TXG	TXE	TXJ	
	0	1	2	3	4	5
R	TXE	TXB	TXC	TXD	TXA	TXI

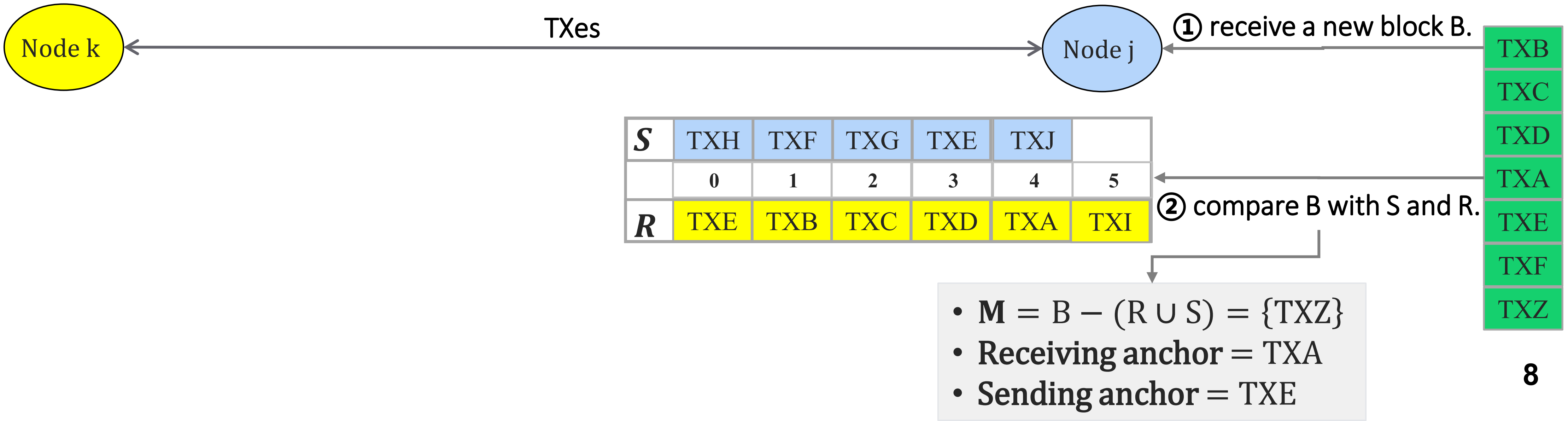
The Details of Dino

- Step 1. Node j receives a new block B.
- Step 2. Node j compares B with its S and R associated with node k.



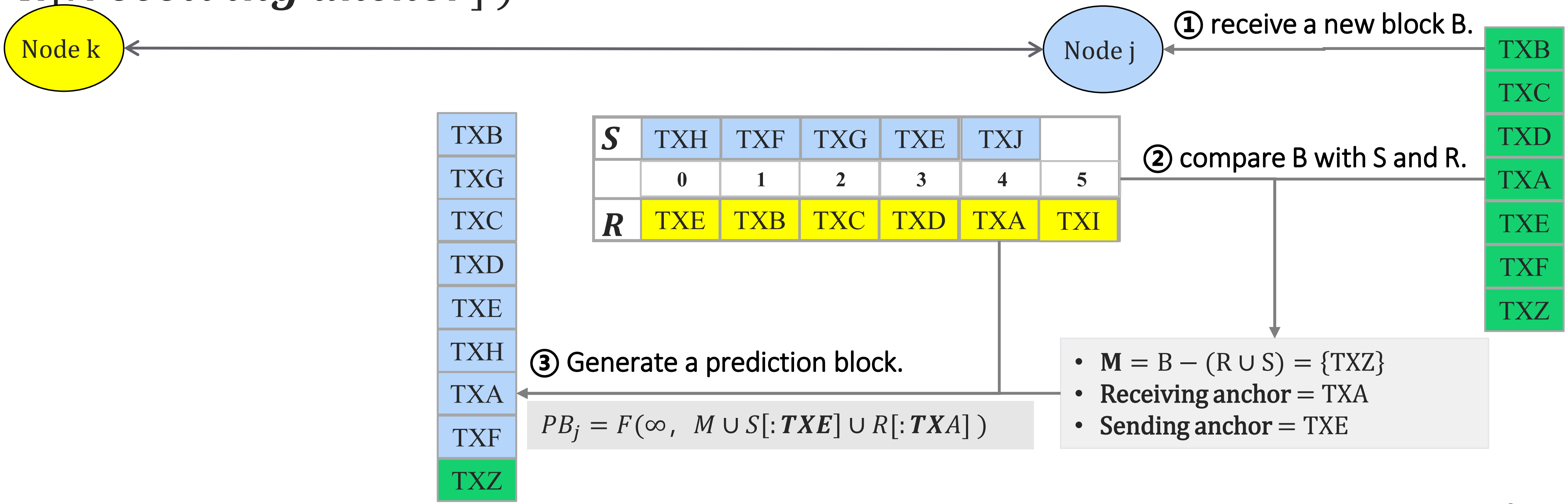
The Details of Dino

- Step 1. Node j receives a new block B.
- Step 2. Node j compares B and its S and R associated with node k.
  - Anchor is a TX hash. TXes located after it do not exist in block B.
  - Node j finds the **sending anchor** and the **receiving anchor** in B.
  - The missing TXes set(**M**) is a set of TXes that exists in B but not in S and R.



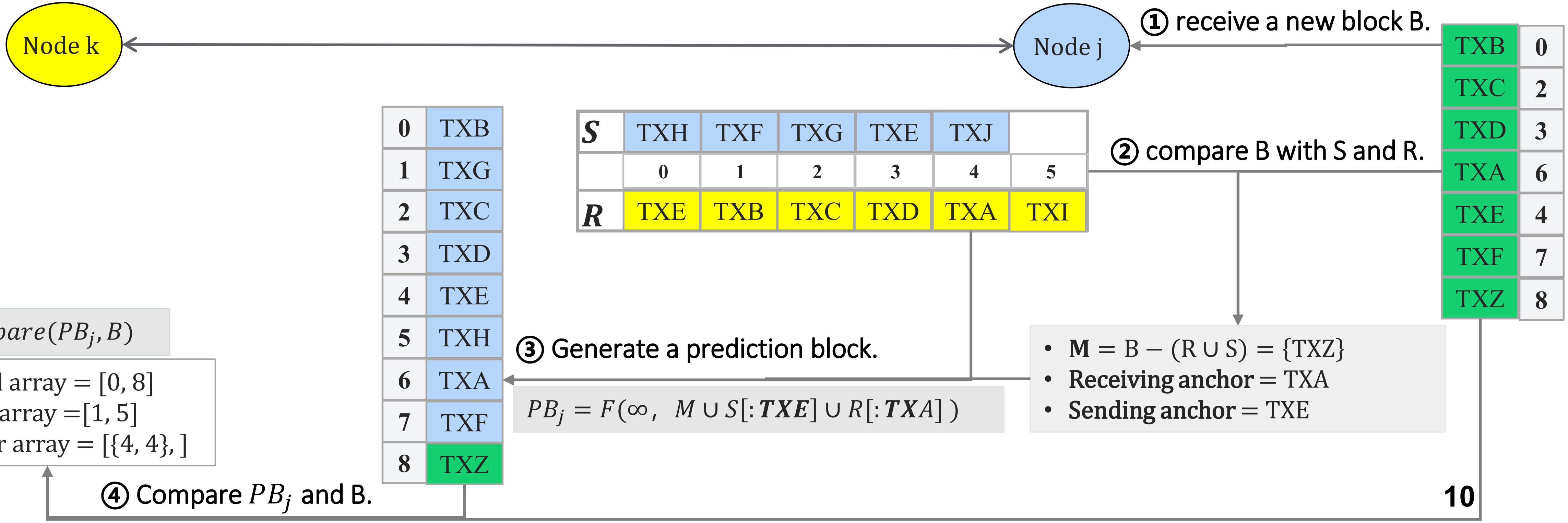
The Details of Dino

- Step 1. Node j receives a new block B.
- Step 2. Node j compares B and its S and R associated with node k.
- Step 3. Node j generates a prediction block PB_j with TXes package algorithm F.
- $\boldsymbol{PB_j} = F(\text{block capacity} = \infty,\ TXes\ set = M \cup S[:\boldsymbol{sending\ anchor}] \cup R[:\boldsymbol{receiving\ anchor}])$



The Details of Dino

- Step 1. Node j receives a new block B.
- Step 2. Node j compares B and its S and R associated with node k.
- Step 3. Node j generates a prediction block PB_j with TXes package algorithm F.
- Step 4. Node j compares PB_j with B to generate a transformation message **T**.



The Details of Dino

- Step 1. Node j receives a new block B.
- Step 2. Node j compares B and its S and R associated with node k.
- Step 3. Node j generate a prediction block PB_j with TXes package algorithm F.
- Step 4. Compare PB_j and B to generate a transformation message **T**.
- Step 5. Node j constructs a Dino block and sends it to node k.
- Dino block contains the receiving anchor, the sending anchor, the missing TXes set and the transformation message **T**.

$\boldsymbol{T} = Compare(PB_j, B)$

- Interval array = [0, 8]
- Delete array =[1, 5]
- Reorder array = [{4, 4},]

- **M** = B – (R ∪ S) = {TXZ}
- **Receiving anchor** = TXA
- **Sending anchor** = TXE

⑤ Construct a Dino block

- *Header + Coinbase TX*
- *Receiving anchor*
- *Sending anchor*
- *M*
- *T*

The Details of Dino

- Step 1. Node j receives a new block B.
- Step 2. Node j compares B and its S and R associated with node k.
- Step 3. Node j generates a prediction block PB_j with TXes package algorithm F.
- Step 4. Node j compares PB_j and B to generate a transformation message **T**.
- Step 5. Node j constructs a Dino block and sends it to node k.
- Step 6. Node k receives the Dino block and generates a prediction block PB_k.
  - $PB_k = F(\infty,\ M \cup R[:\text{sending anchor}] \cup S[:receiving\ anchor])$
  - Tx order in node k's S is the same as the TX order in node j's R.
  - Tx order in node k's R is the same as the TX order in node j's S.
  - PB_k is identical to PB_j.
- Step 7. Node k rebuilds **B** with PB_k and transformation message **T** in the Dino block.
  - $B = Rebuild(PB_k, T)$

Analysis of Dino

- A Dino block contains following ingredients:
  - Block header and Coinbase TX.
  - Missing TXes set **M**
  - Receiving anchor and Sending Anchor.
  - Transformation message **T**.

- The size of a Dino block is determined by the number of missing TXes and the size of transformation message T.

Assumption 1: Almost all transactions in a new block already exists in the mempools of other nodes.

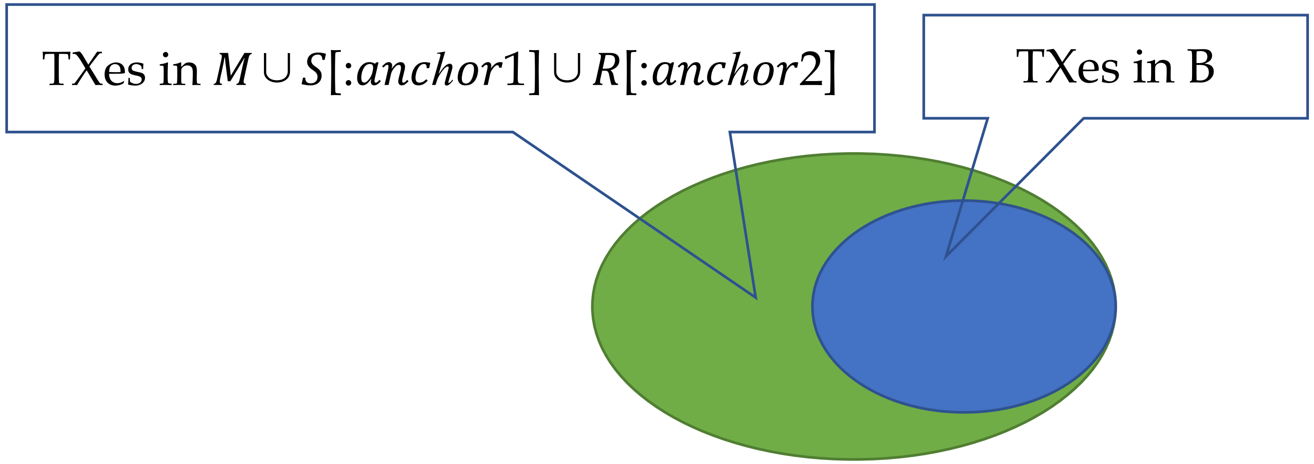
- When this assumption is satisfied, the number of missing TXes could be zero.

Analysis of Dino

- The size of a transformation message T is determined by the similarity between the transaction order in the prediction block and that in the original block.

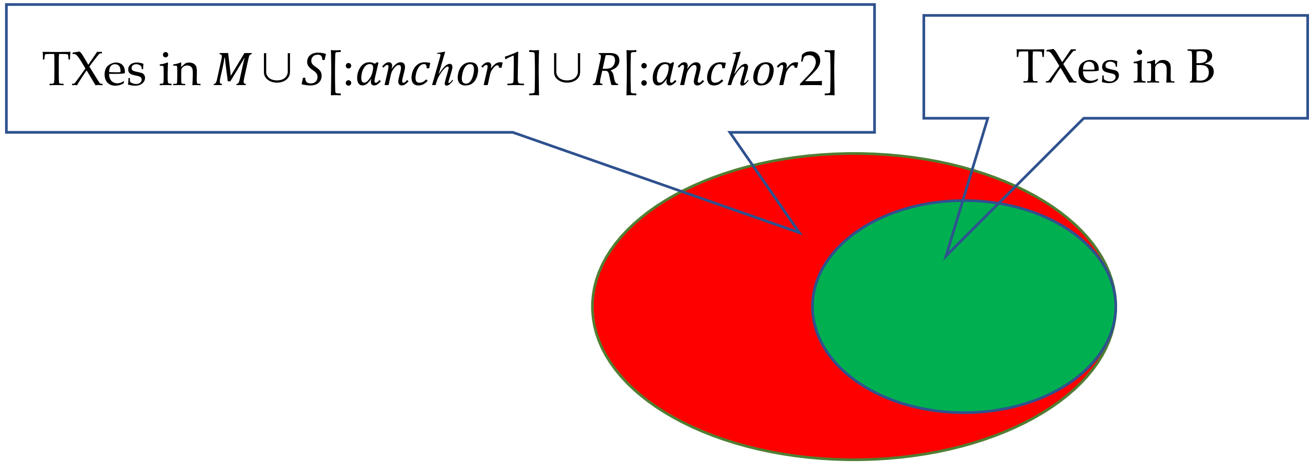
Assumption 2: miners are **profit-oriented**.

- When this assumption is satisfied,
  - 1. Transactions in B is a subset of transactions in M, S, and R.
  - 2. Both B and PB are generated by the same algorithm F.
- The TX order in PB is almost the same as the TX order in B.



Analysis of Dino

- The TX order in PB is almost the same as the TX order in B.
- TXes in B must exist in a range of PB.
- We can reconstruct B by **deleting** and **reordering** some TXes in PB.
- The number of TXes which needs to be deleted and reordered are impacted by two factors:
  - The transaction generation rate v in the blockchain network.
  - The latency that a TX disseminates to the whole network

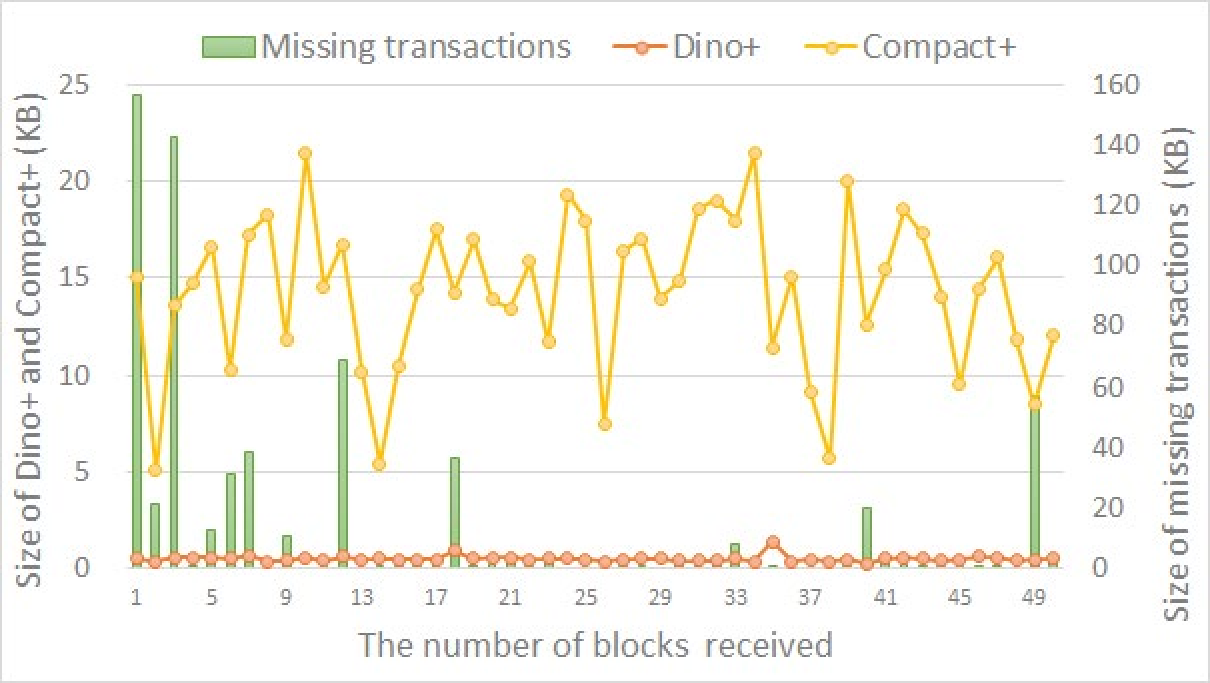


0	TXB
1	TXG
2	TXC
3	TXD
4	TXE
5	TXH
6	TXA
7	TXF
8	TXZ

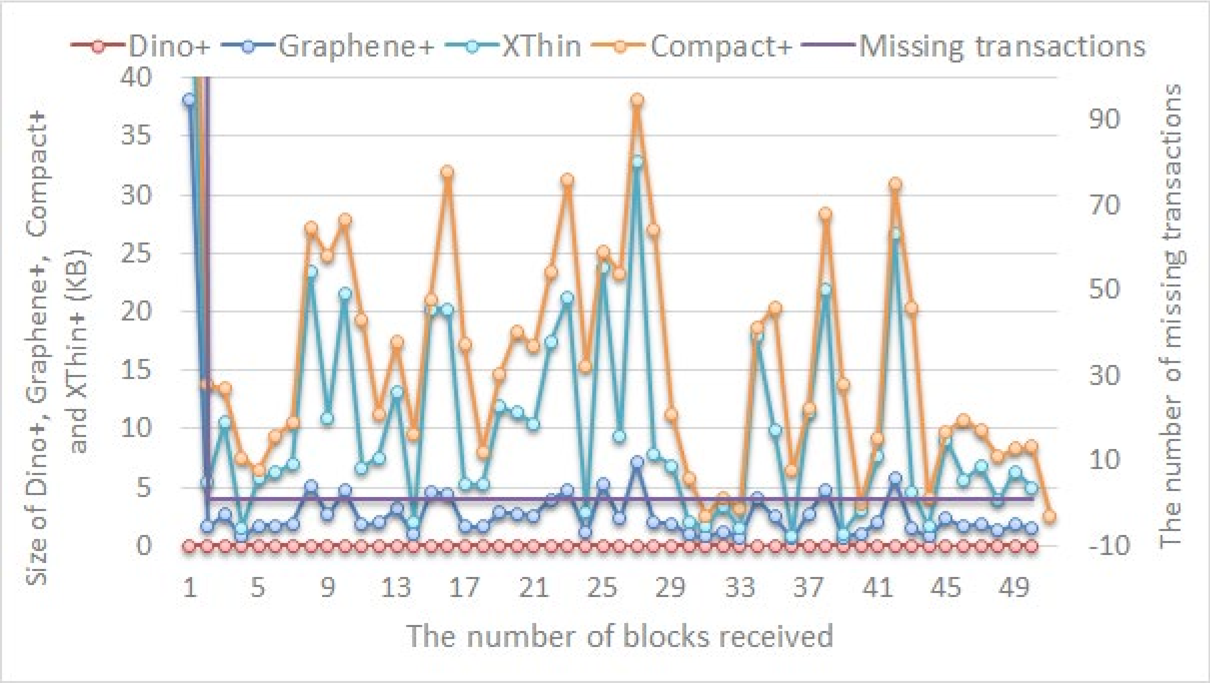
TXB	0
TXC	2
TXD	3
TXA	6
TXE	4
TXF	7
TXZ	8

Evaluation

The experiment was conducted in a cluster with 16 nodes.



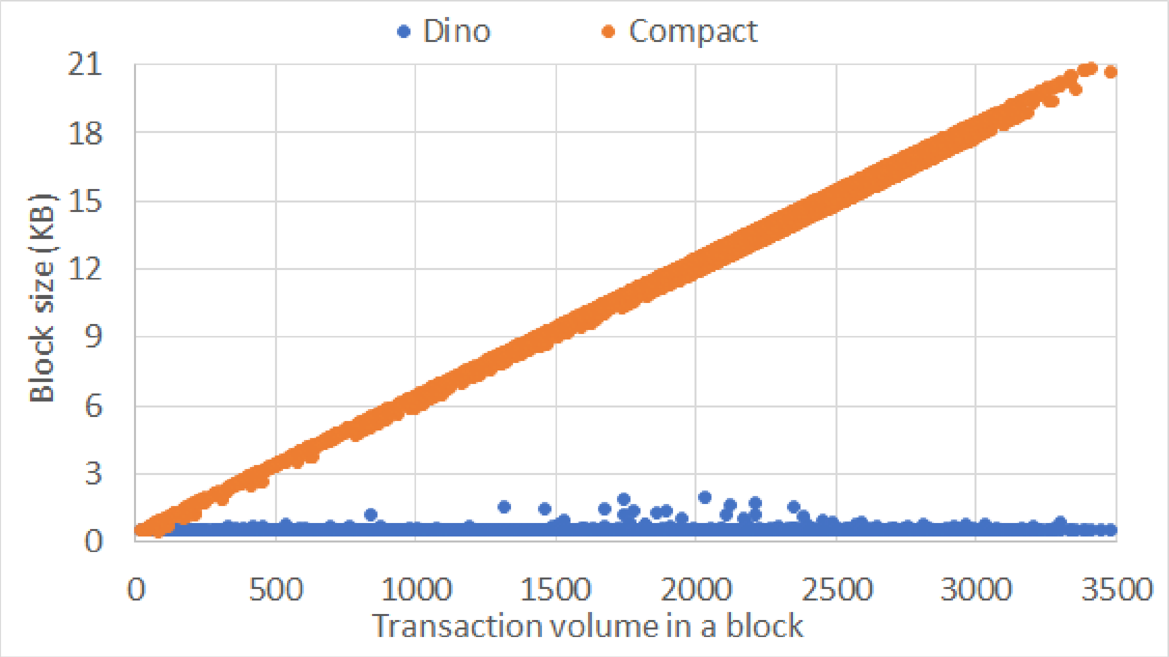
Dino vs Compact in Bitcoin



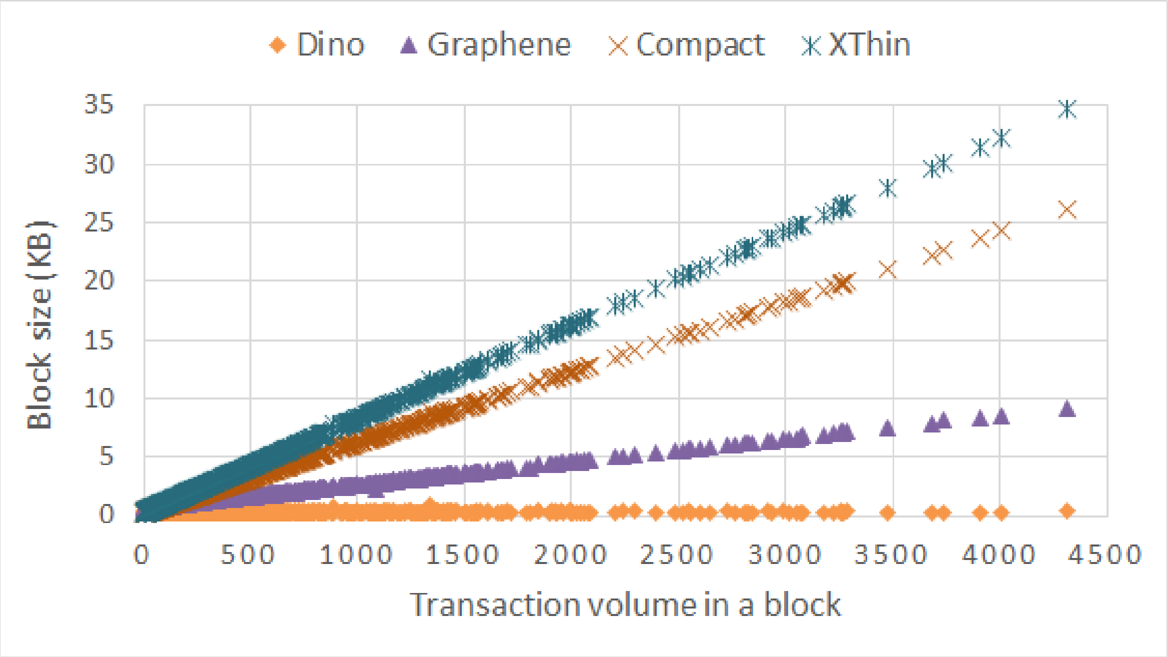
Dino, Graphene, Compact, XThin in Bitcoin Cash

Evaluation

- When there is no missing transaction, the size of a Dino block keeps constant when the transaction volume increases from zero to the max limit.



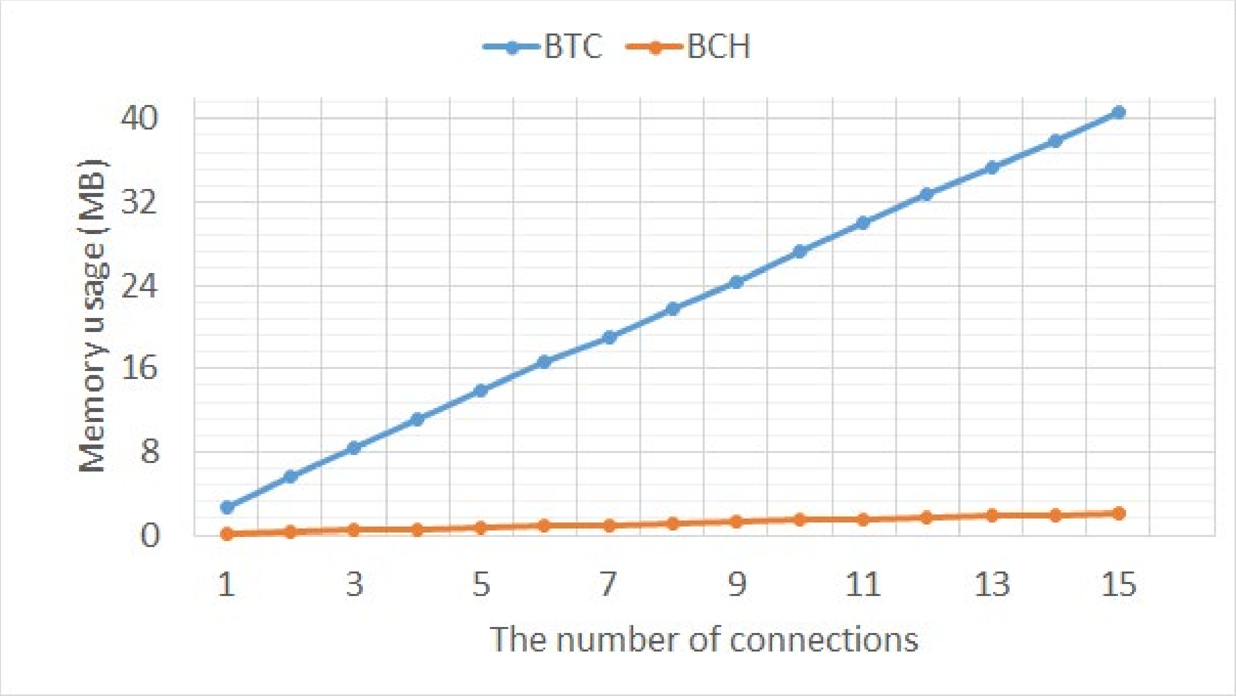
Dino vs Compact in Bitcoin



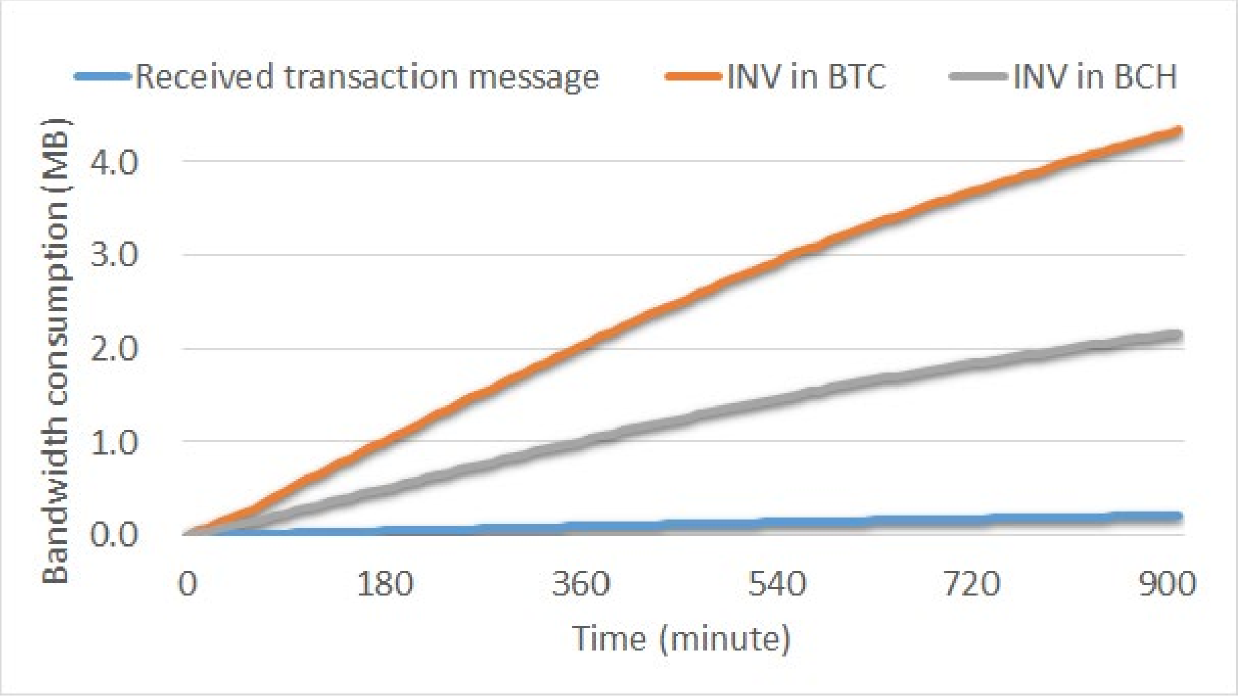
Dino, Graphene, Compact, XThin in Bitcoin Cash

Evaluation

- When there is no missing transaction, the size of a Dino block keeps constant when the transaction volume increases from zero to the max limit.



Dino's memory usage

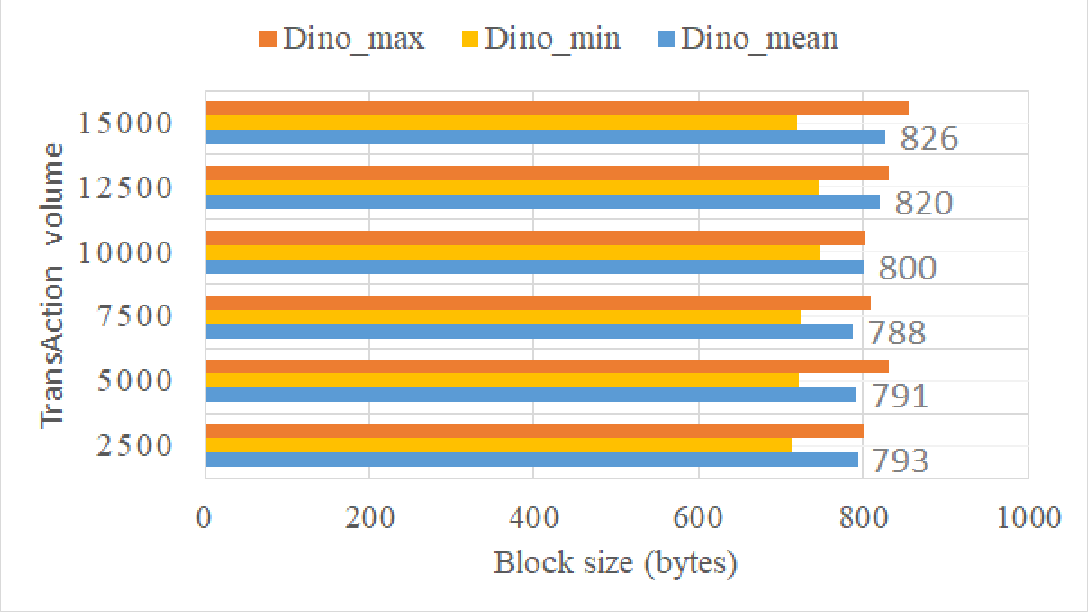


Dino's bandwidth costs

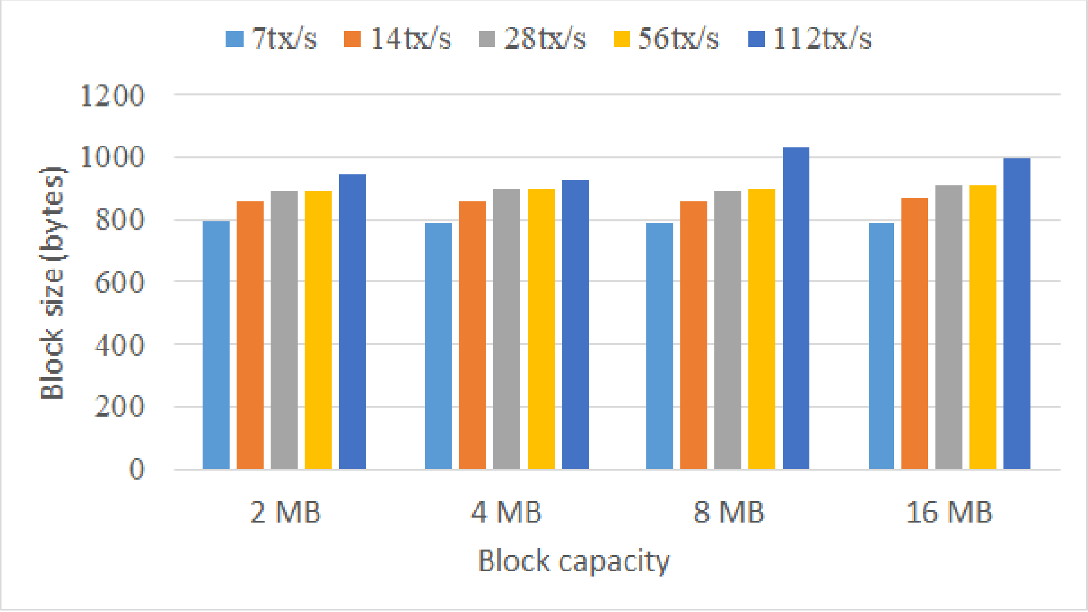
Simulation

A Dino block is always no more than 1 KB when

- The transaction volume is increased from 2500 to 15000.
- The TPS is increased from 7 to 112.
- The block capacity is increased from 1MB to 16 MB.



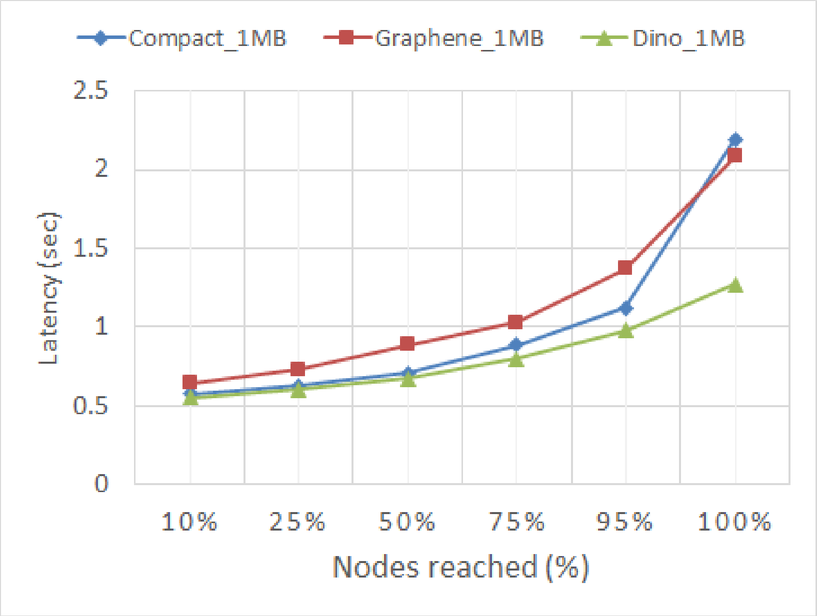
Dino's stability



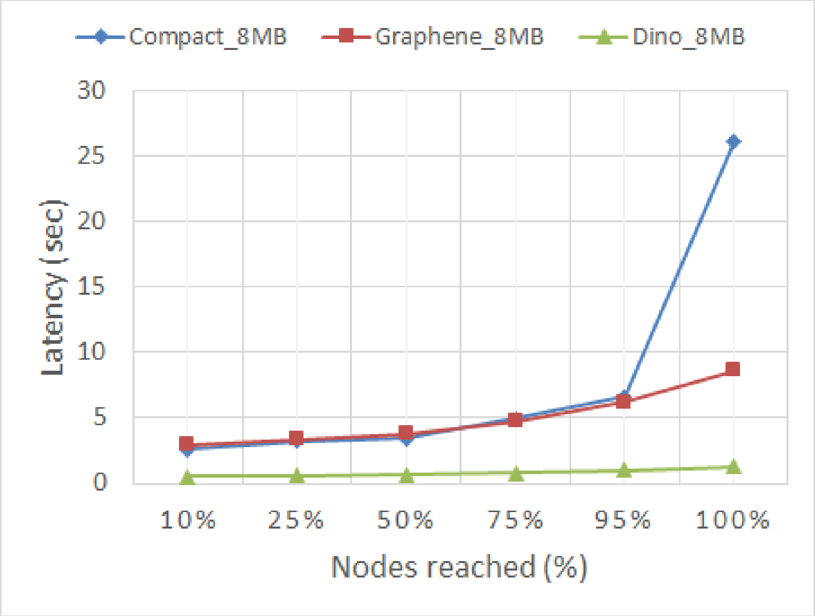
Dino's scalability

Simulation

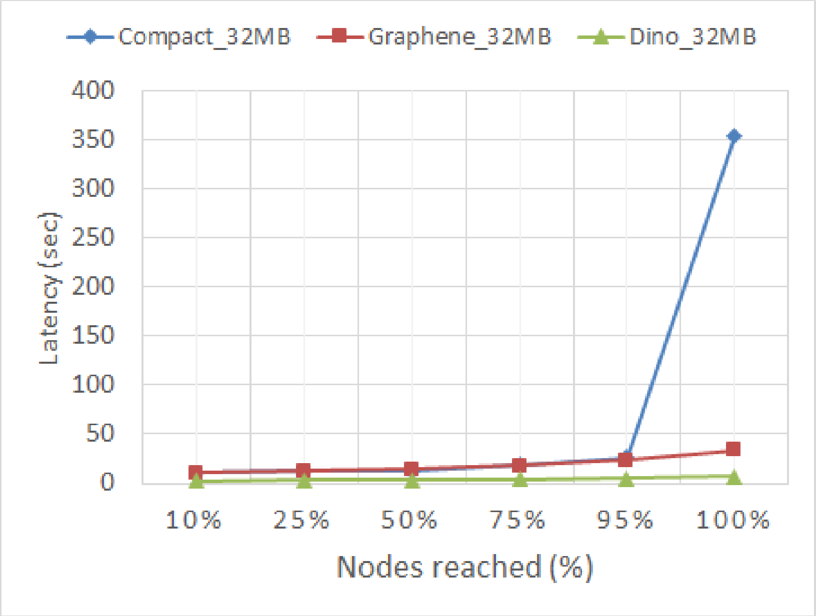
- Block propagation latency comparison among Dino, Compact, and Graphene.
- As the Dino block is smaller, it takes less time to reach all nodes.



1MB Block Propagation latency



8MB Block Propagation latency



32 MB Block Propagation latency

Conclusion

- We present a block dissemination protocol that transmits block construction rules instead of block content.
- Our results illustrate that Dino has a substantial advantage of scaling to larger transaction volume and higher transaction generation rates.
- We point out a new direction of block transmission that is promising to overcome the contradiction between blockchain networks' performance and security.

Thank you