北京大学
PEKING UNIVERSITY

A Data Flow Framework with High Throughput and Low Latency for Permissioned Blockchains

Zhenxing Hu[‡ §], Shengjie Guan[‡ §], Wenbo Xu[§], Zhen Xiao[‡]
Jie Shi[§], Pengze Li[‡], Qiuyu Ding[‡ §], Hui Ding[§] and Chao Zeng[§]

[‡]*School of Computer Science, Peking University, China*
[§]*Ant Group, China*

Background and Motivation

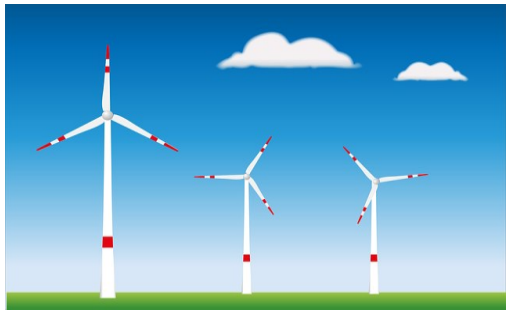
- Blockchain possesses the traits of transparency, immutability, and traceability.
- It is widely utilized in various sectors, such as payment, supply chain, healthcare, finance, and energy.

Payment

Supply chain

Healthcare

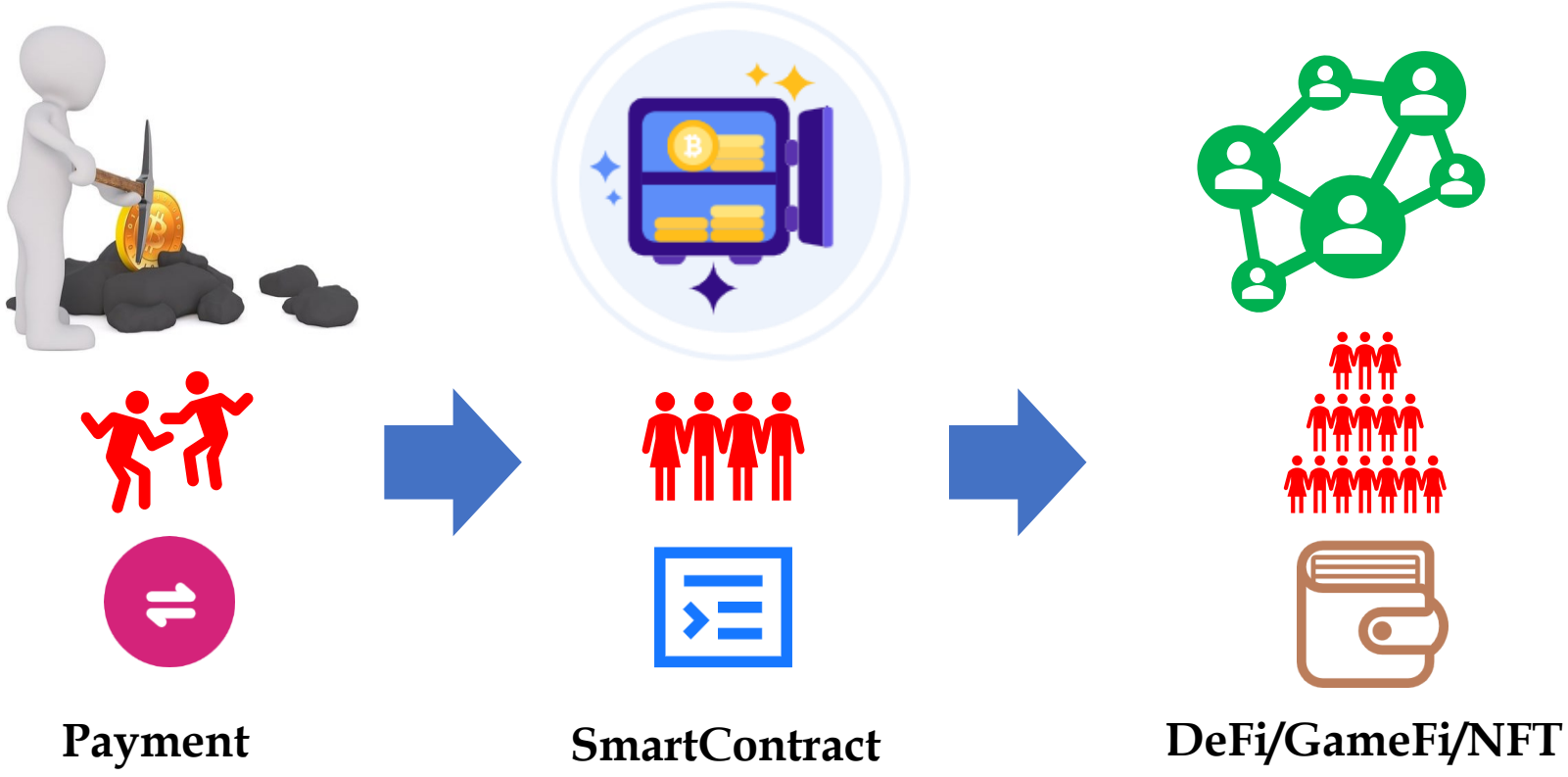
Finance



Energy

Background and Motivation

- The development of blockchain technology has evolved from the early days of Bitcoin's simple payments to Ethereum's smart contracts, and now to a wide range of rich DApps.
- As a result, the user scale of blockchain is also increasing.



Background and Motivation

- Due to its inherent pursuit of decentralization, high throughput has always been a common goal pursued by all blockchain enthusiasts.
- The consensus protocols have evolved from POW to POS, and subsequently to more advanced protocols such as DPOS, HotStuff, and so on.

POW（Bitcoin） → POS（Ethereum） PBFT → DPOS/HotStuff → ...

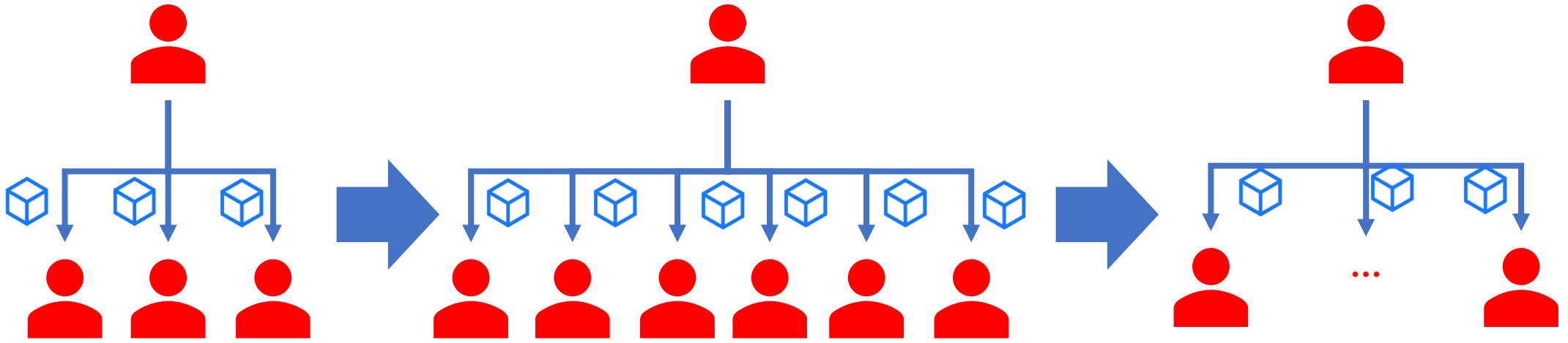
Payment

SmartContract

DeFi/GameFi/NFT

Background and Motivation

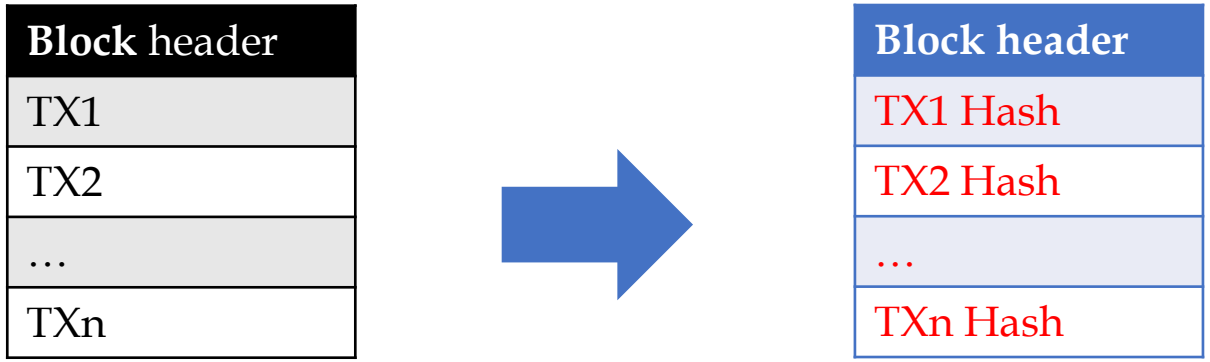
- Leader based BFT consensus protocols are mainly used in the current blockchains.
- The throughput of these protocols is always constrained by the bandwidth of the leader node.
- One more node implies that the leader has to send one more candidate block **in the propose phase.**



The leader node's bandwidth consumption

Background and Motivation

- To improve scalability and throughput, two methods are proposed: saving and borrowing bandwidth.
- **Saving bandwidth:**
  - The leader replaces each transaction with a short transaction hash and forms a compact block.
  - The leader distributes the compact block to other nodes in the propose phase.
  - Receivers recover the candidate block from the compact block and then enter the voting phase.



Candidate block **Compact block**

Bandwidth Consumption: $O(NMB) \rightarrow O(NMX)$

N: the number of consensus nodes

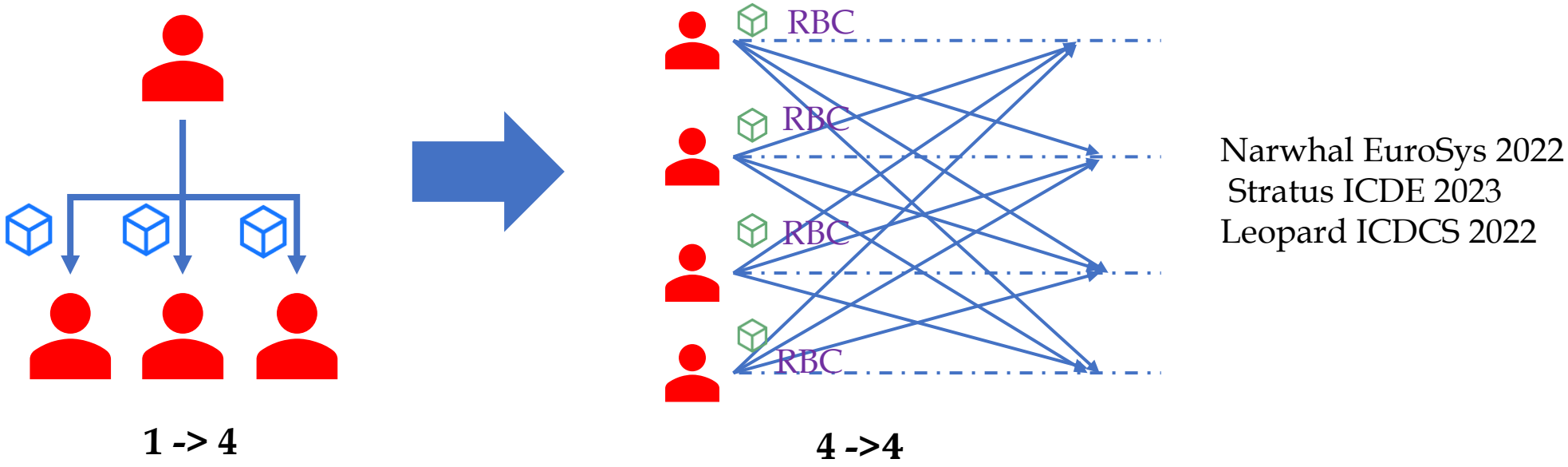
M: the number of transactions in a block

B: the size of a TX, X: the size of a short hash

Background and Motivation

Borrowing bandwidth

- All nodes pack transactions and produce micro-blocks
- Every producer uses reliable broadcast (RBC) to distribute micro-blocks.
- Every receiver needs to send a confirmation message to the producer.
- Every producer needs to collect at least n-f confirmations for every micro-block.



1 -> 4

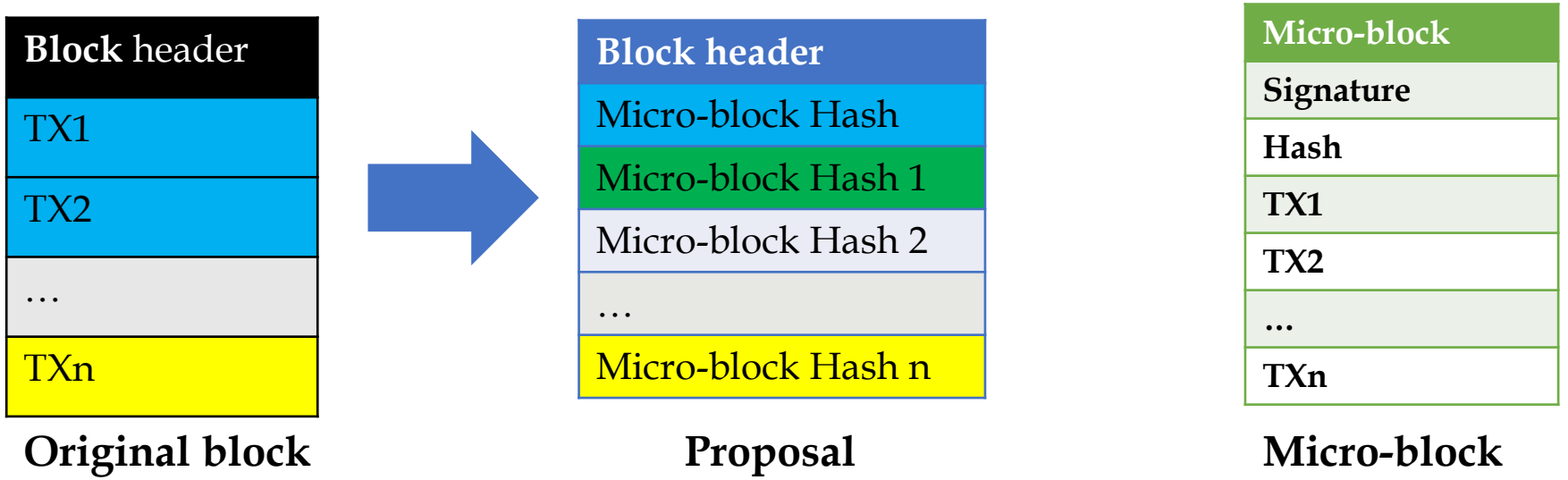
4 ->4

Narwhal EuroSys 2022
Stratus ICDE 2023
Leopard ICDCS 2022

Background and Motivation

Borrowing bandwidth

- A micro-block can be put into the candidate block if it has enough confirmations.
- In the propose phase, the leader puts each micro-block's hash into the candidate block.
- Receivers recover the candidate block from the proposal and then enter the voting phase.



Bandwidth Consumption :$O(NMB) \rightarrow O(NZX)$

N: the number of consensus nodes

M: the number of transactions in a block

B: the size of a TX, X: the size of a short hash

Z: the number of micro-block hashes

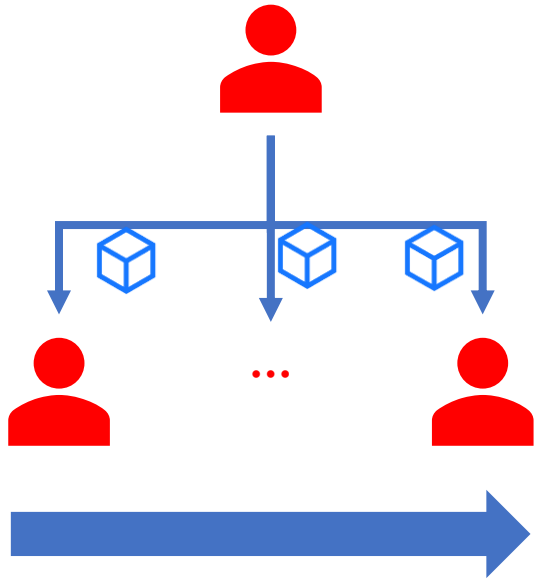
Background and Motivation

- We find that when we increase the transaction volume in the candidate block, the leader's bandwidth consumption of those methods also increases.
- Can we reduce the leader node's bandwidth consumption step further?

Save bandwidth : $O(NM)$

Borrow bandwidth : $O(NZ)$

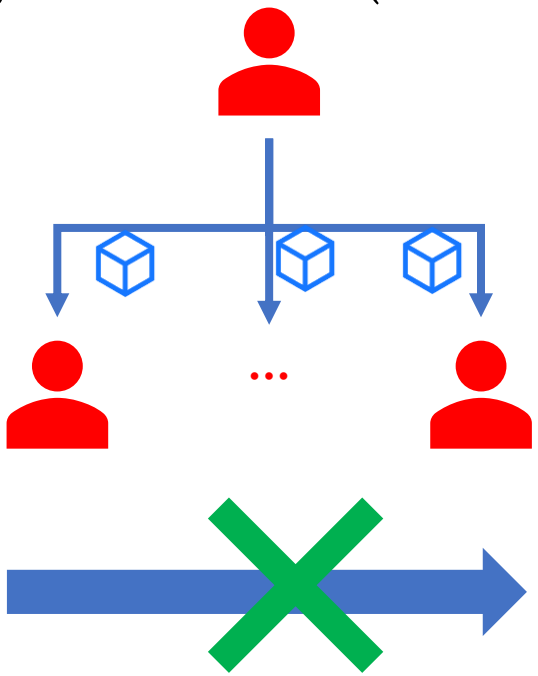
X is small, it is ignored.



Predis

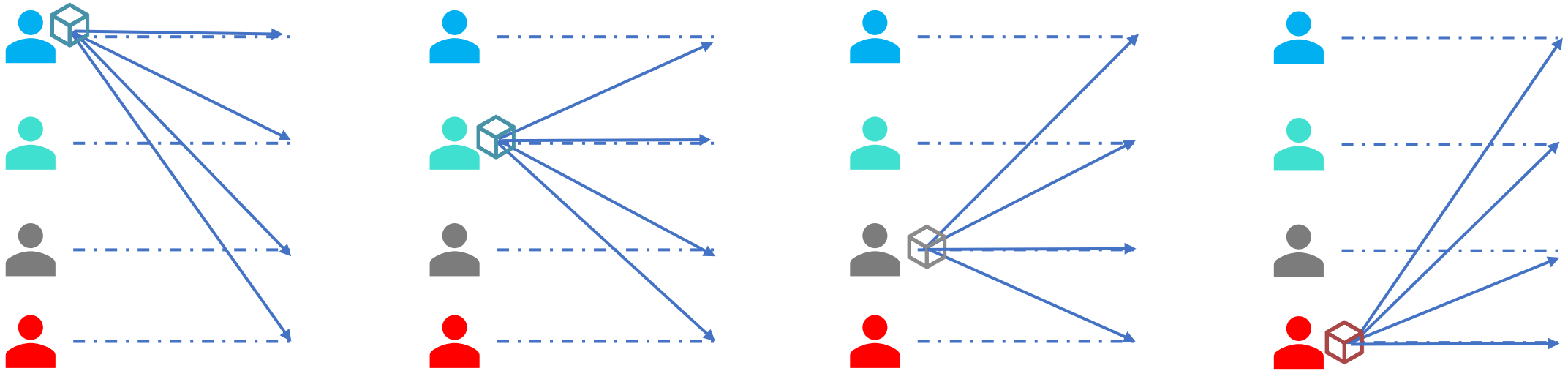
- We propose a new data distribution strategy called ***Predis,*** which is designed to enhance throughput of leader-based BFT protocols (PBFT, HotStuff, and etc.).
- ***Predis*** is also a method based borrowing bandwidth which reduces leader's bandwidth consumption to $O(N)$.
- ***Predis*** also has lower message complexity because producers do not need to collect confirmations for every micro-block (called bundle in ***Predis***) it produced.

$$\textbf{\textit{Predis}} = \boldsymbol{O(N)}$$



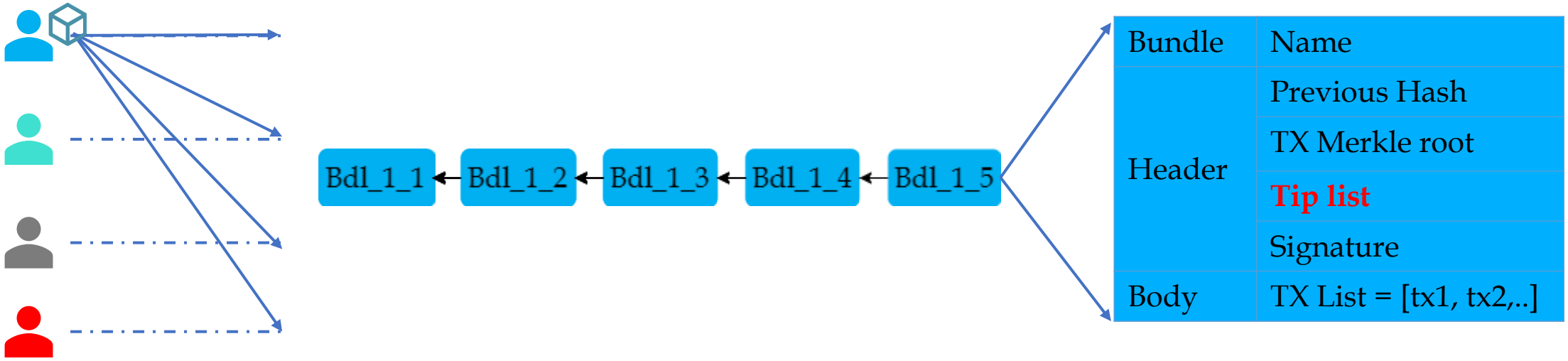
Predis

- In ***Predis,*** each consensus node continuously bundles up transactions and multicasts each bundle to other nodes.
- If there are n consensus nodes, there are n nodes producing bundles.



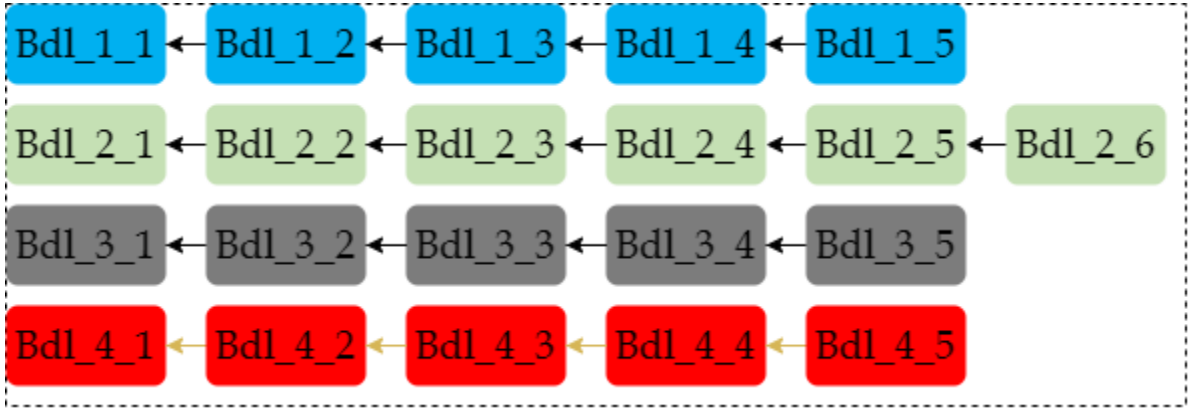
Predis

- The structure of a bundle is like the structure of a block in the blockchain.
- Each bundle has a header and a body.
- Each bundle has only one parent bundle.
- The Tip list contains the height of the latest bundle that the producer received for each bundle chain.



Predis

- Eventually, each node's mempool consists of n parallel bundle chains.
- With the tip list in each bundle, nodes do not need to send additional confirmation messages for every received bundle because the receiver can ascertain the sender's latest bundle height from the tip list of the receiving bundles.
- The ***tip list*** reduces message complexity and can achieve equivalent outcomes to reliable broadcast (RBC).

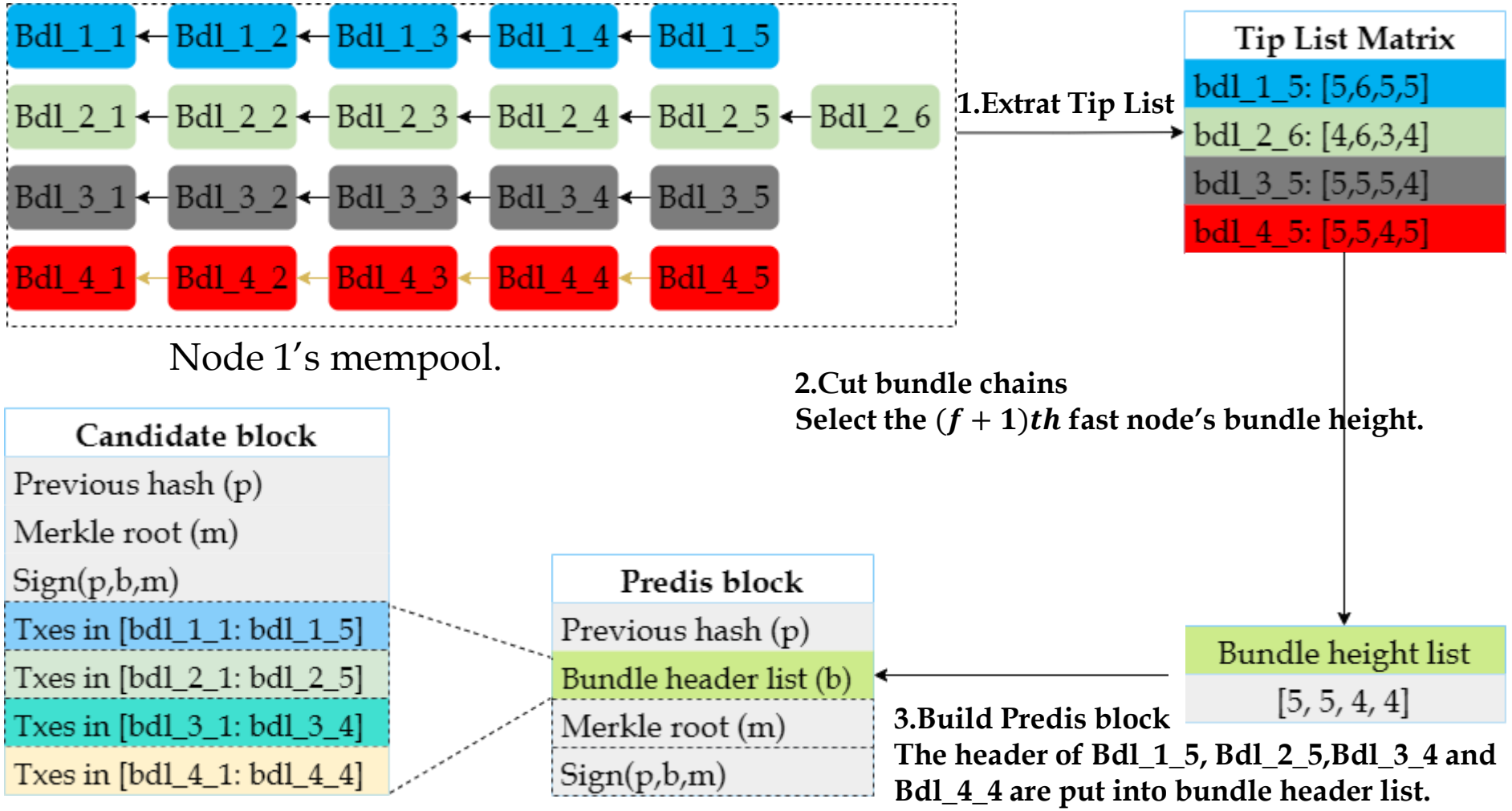


Node 1's mempool.

Bdl_1_5	Name
Header	Previous Hash (Bdl_1_4)
	TX Merkle root
	Tip list =[5,6,5,5]
	Signature
Body	TX List

Node 1 produces Bdl_1_5

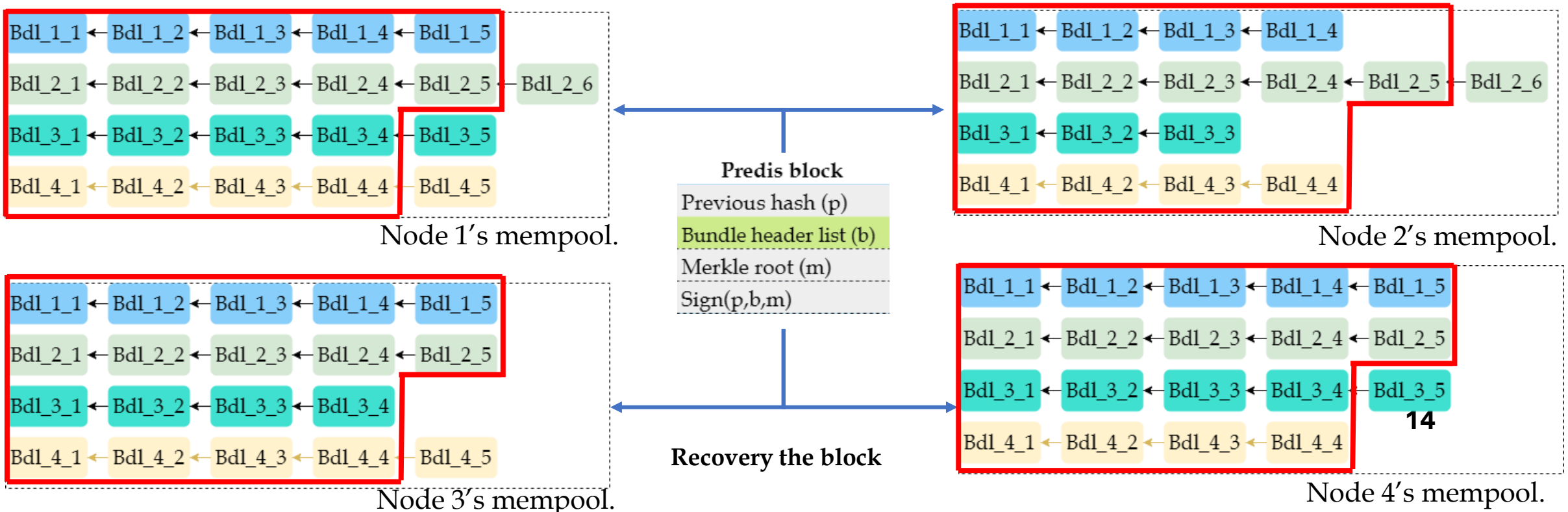
Predis



The leader proposes a ***Predis*** block.

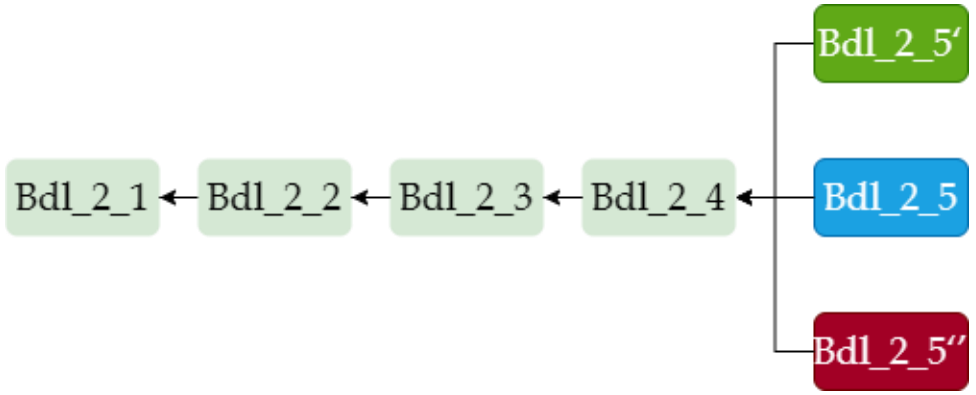
Predis

- Node 1, 3, and 4 can successfully recover the candidate block, while node 2 has to fetch missing bundles from other nodes.
- The cutting rule guarantees that node 2 can fetch each missing bundle from at least n-f nodes.
- If the computed transaction Merkle root in the candidate block equals that in the Predis block, a node will vote for that Predis block and enter the following consensus phases.



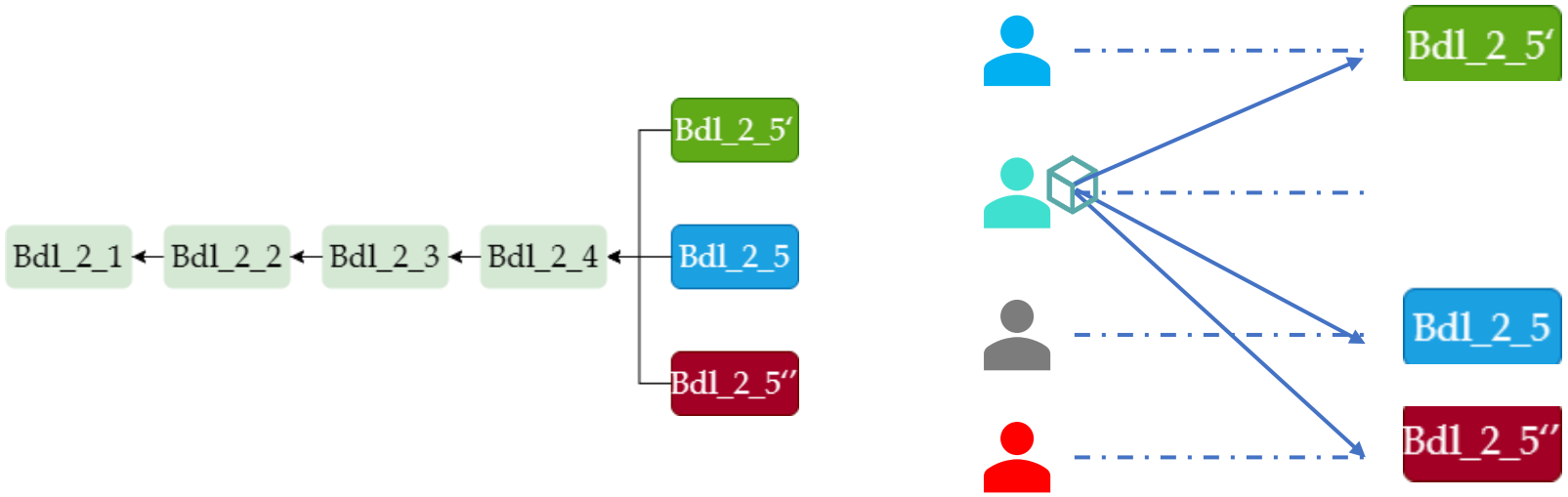
Predis-forking attack

- Malicious nodes can produce conflicting bundles.
- A node will vote for a Predis block provided that:
  - It has all transactions that Predis block contains.
  - The computed Merkle root of transactions in all bundles equals that in the Predis block.
- The forking attack cannot break ***Predis'*** safety.



Predis-forking attack

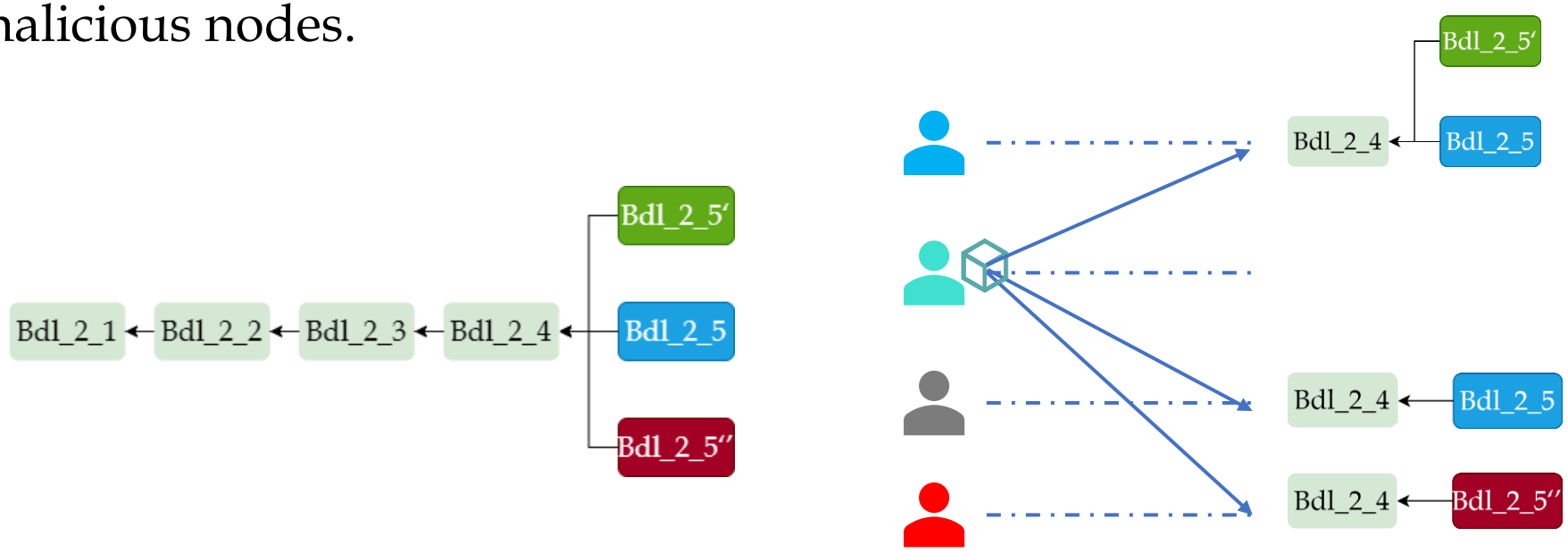
- **Case 1** : A producer distributes conflict bundles to different nodes.
- However, the leader can only cut bundle chains at one branch in propose phase.
- When a node receives a Predis block, it will fetch other conflict bundles. If it cannot obtain them, it will not vote for that Predis block.
- If the consensus time is out and the consensus is not reached, nodes will initiate view-change to keep liveness.



node 2 produces conflict bundles at height 5.

Predis-forking attack

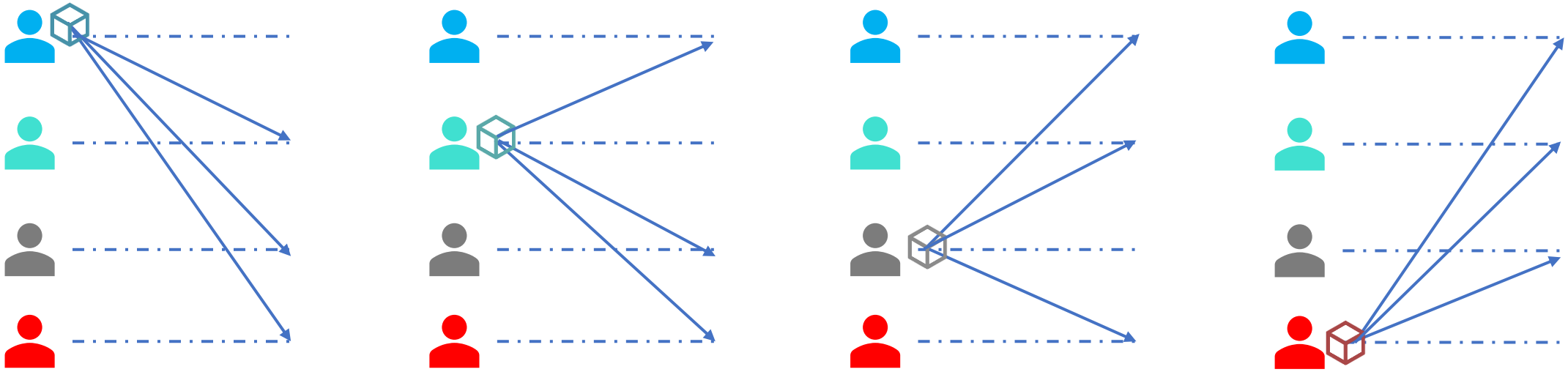
- **Case 2** : An honest node receives conflict bundles.
- The honest node puts the producer into a ban list for a punishment, and broadcasts conflict bundle headers to other nodes.
- An honest leader will not cut bundle chains produced by banned nodes.
- An honest node will never vote for a Predis block that contains bundles produced by malicious nodes.



node 2 produces conflict bundles at height 5.

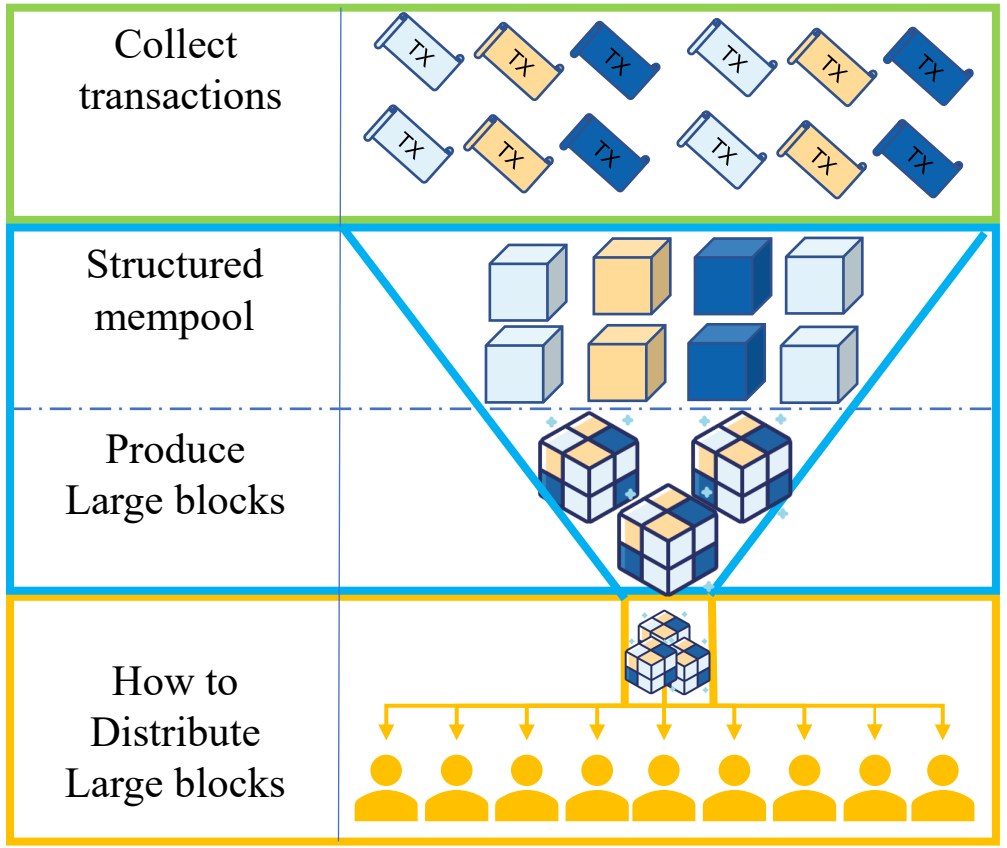
Predis' Advantages

- **High throughput:** One consensus round confirms massive transactions.
- **Low message complexity:** Achieve the same effect as RBC without requiring extra confirmation messages.
- **Low bandwidth consumption:** Leader's bandwidth consumption is O(N) in propose phase.



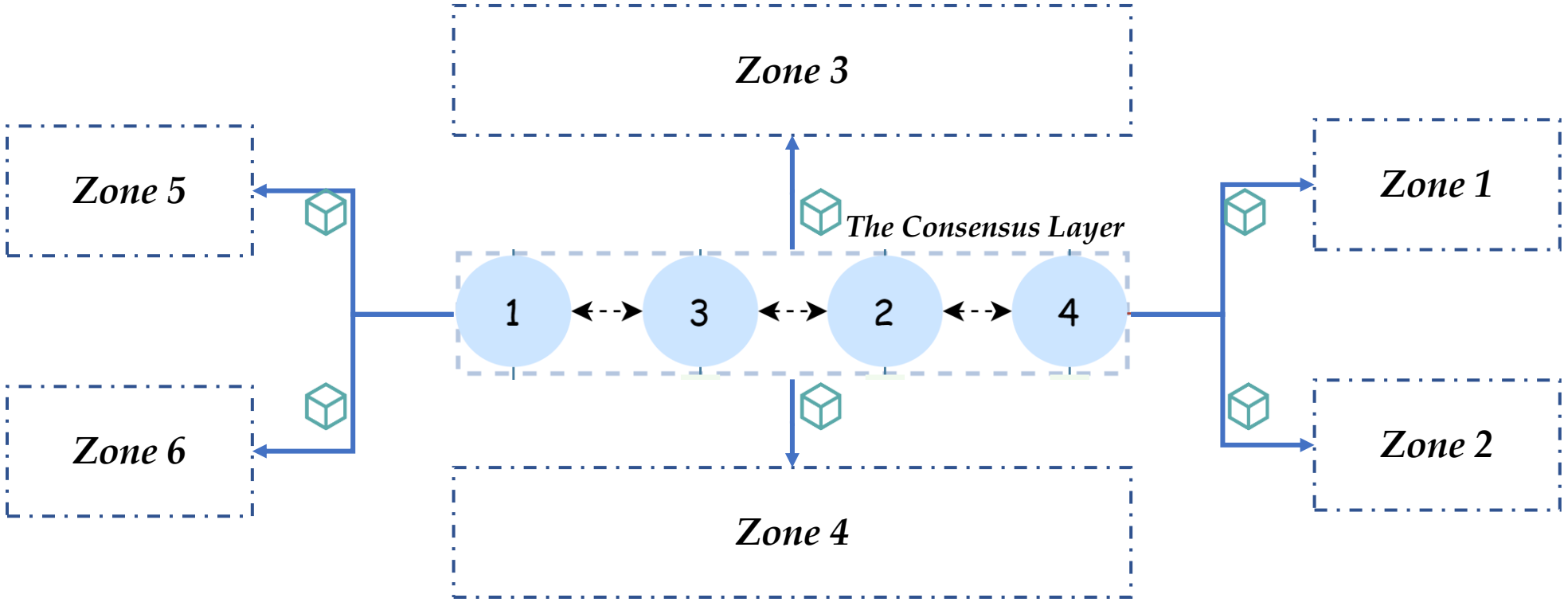
The next challenge

- In addition to consensus nodes, there may be many other full nodes in the blockchain.
- The ***Predis*** block maps to a large number of transactions, which means that the original blocks are very large.
- How to distribute these large blocks to other full nodes as fast as possible?
- We should build a proper network topology and data flow strategy between the consensus layer and the network layer.



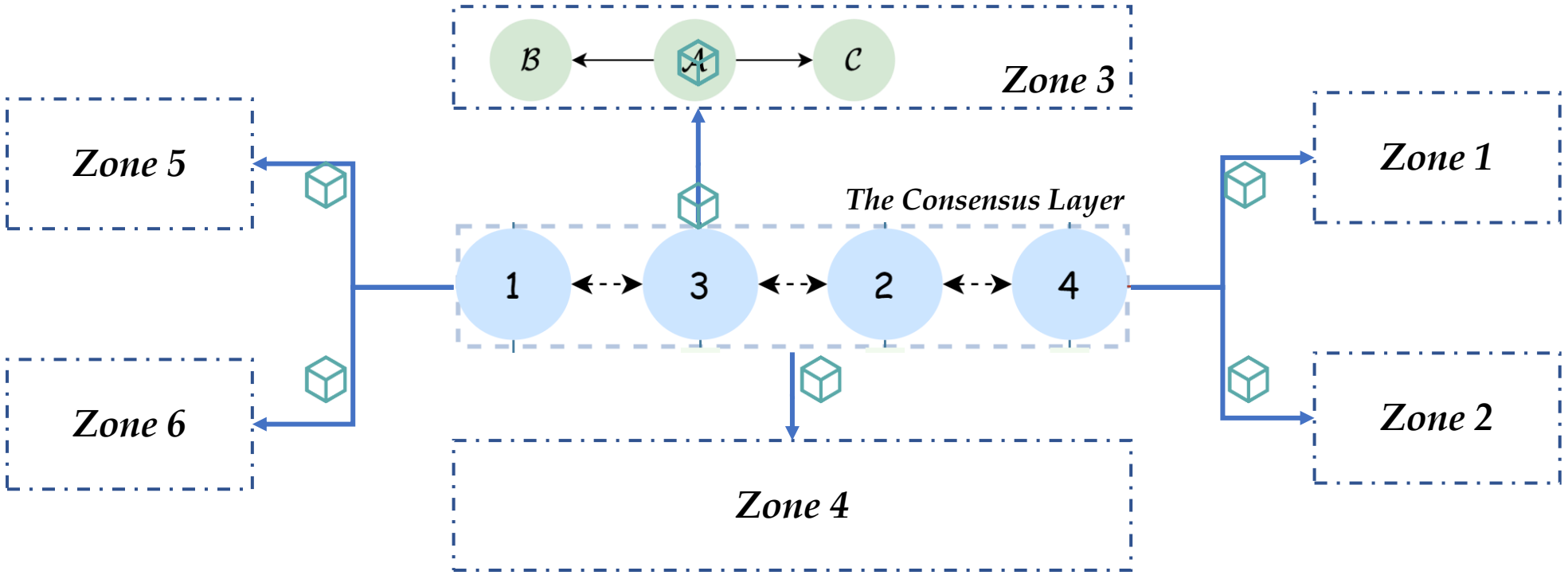
Multi-Zone

- ***Multi-Zone*** is a network topology to distribute bundles and ***Predis*** blocks to full nodes.
- It divides the network into several areas, each of which counts as a zone.
- This design simplifies the data flow from the consensus layer to the network layer.
- How does the consensus layer distribute data to the network layer?



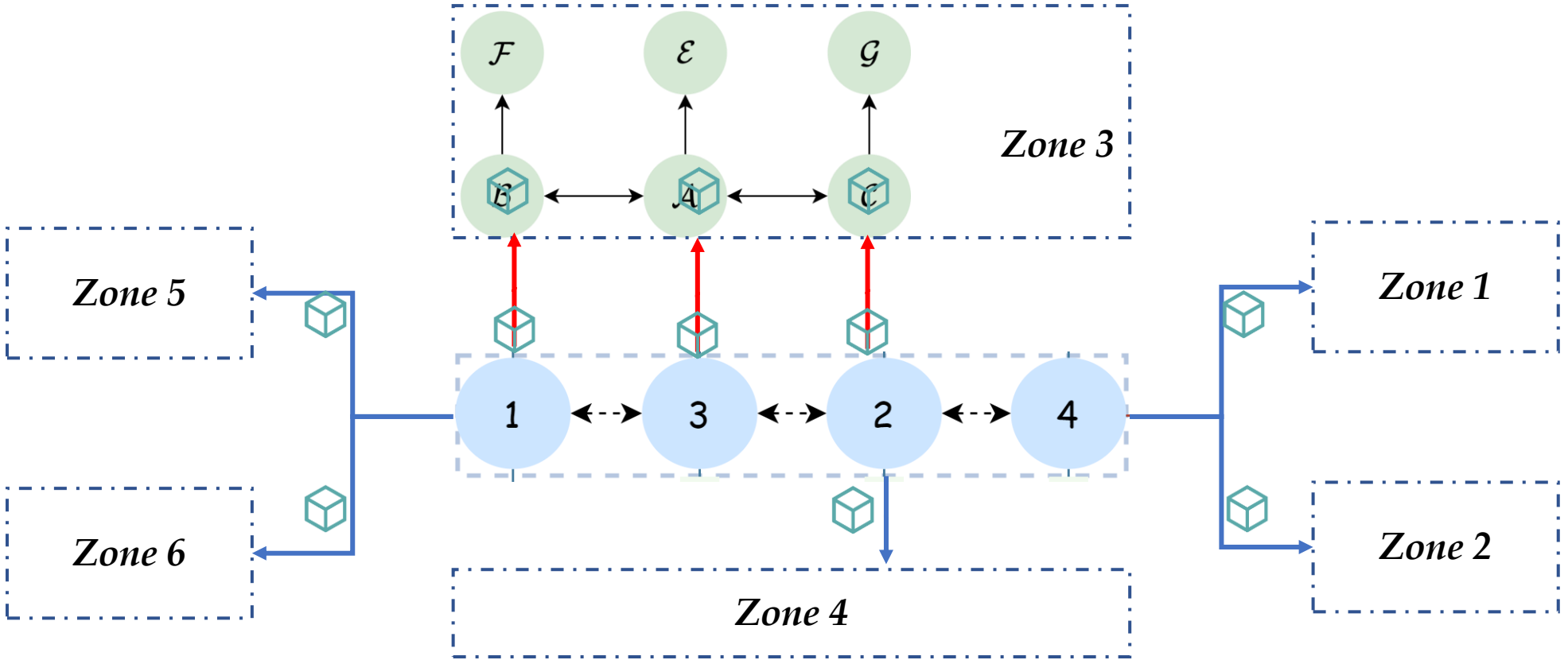
Multi-Zone

- When new data is produced, a consensus node is responsible for sending data to a zone.
- The nodes in each zone work together to finish data distribution.
- How to deal with node failure problem (Node 3 crashes, node A crashes).



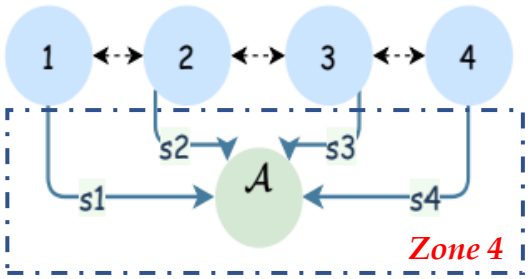
Multi-Zone

- More than one consensus nodes send data to every zone, and every zone uses gossip to handle node crashes.
- That consumes too much bandwidth of the consensus layer, degrading throughput.
- Gossip is not fast enough and also wastes bandwidth.

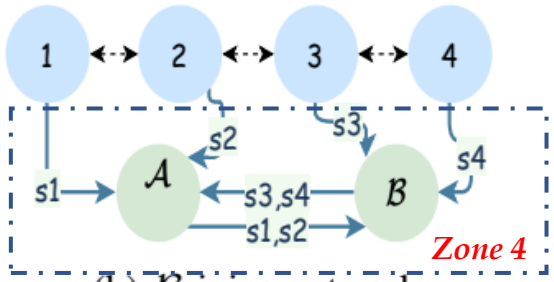


Multi-Zone

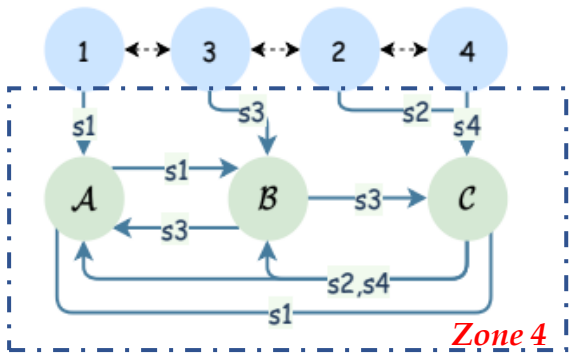
- There are three types of nodes in Multi-Zone: consensus nodes, relayer nodes, and ordinary nodes.
- Relayers receive data from consensus nodes and forward it to other relayers and ordinary nodes.
- The *ith* consensus nodes use erasure encoding to encode a bundle to n stripes and sends the *ith* stripes to relayers.
- n consensus nodes send n stripes to at most n relayers in each zone, if f consensus nodes crash or f relayers crash, other nodes can still receive a bundle.



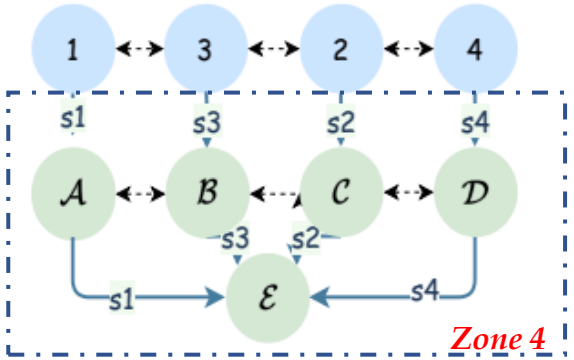
(a) $\mathcal{A}$ joins network



(b) $\mathcal{B}$ joins network



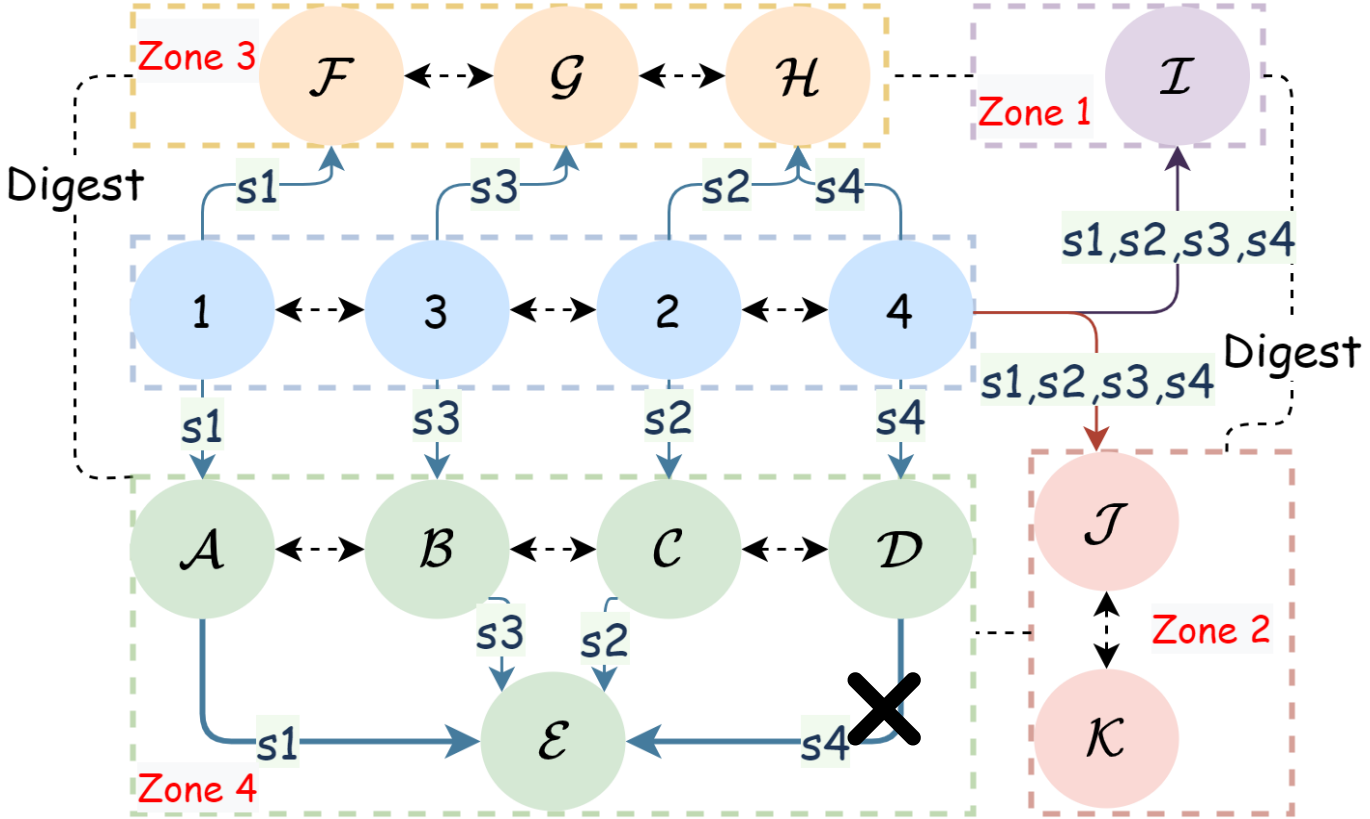
(c) $\mathcal{C}$ joins network



(d) $\mathcal{D}$ and $\mathcal{E}$ join network

Multi-Zone

- Each ***zone*** has at most ***n*** relayers.
- When relayers in a zone crashed, other ordinary nodes will become new relayers.
- As a node has enough parents, when it receives ***n-f*** stripes, it will decode a bundle.
- If one node's several parent crashes, it can still receive the data.



Multi-Zone's Advantages

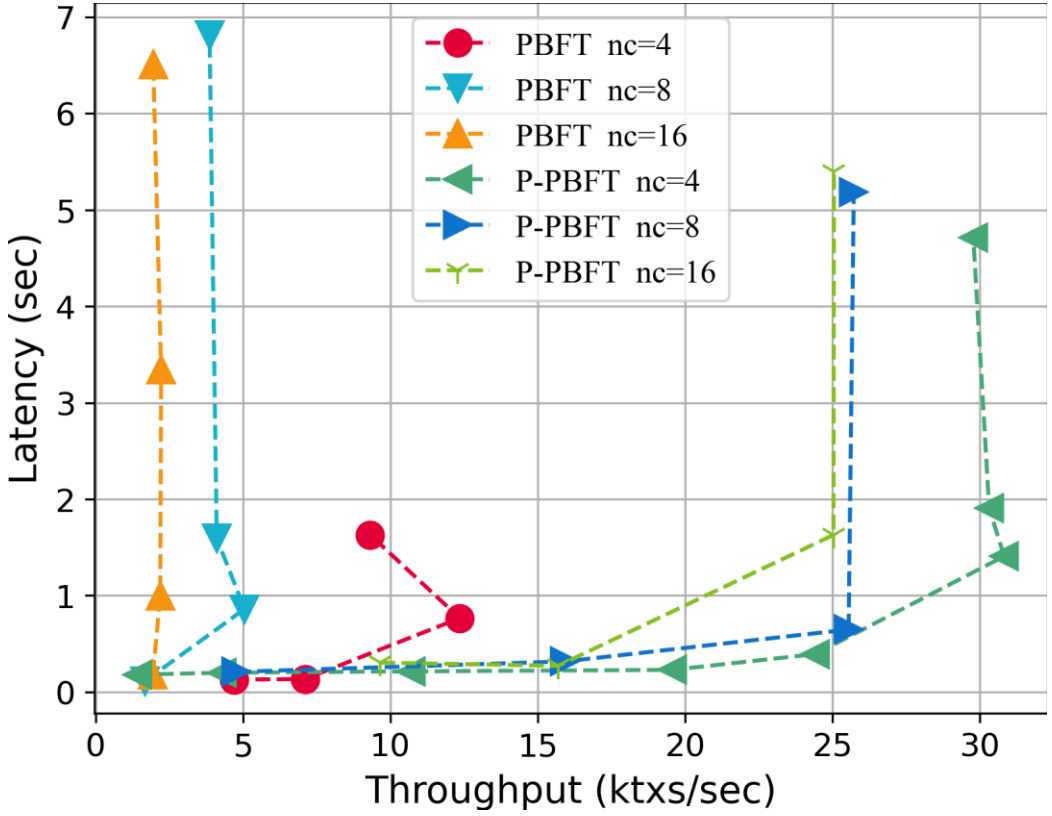
- **Good Scalability:** Keep the bandwidth consumption of distributing blocks by consensus nodes from growing as the number of full nodes rises.

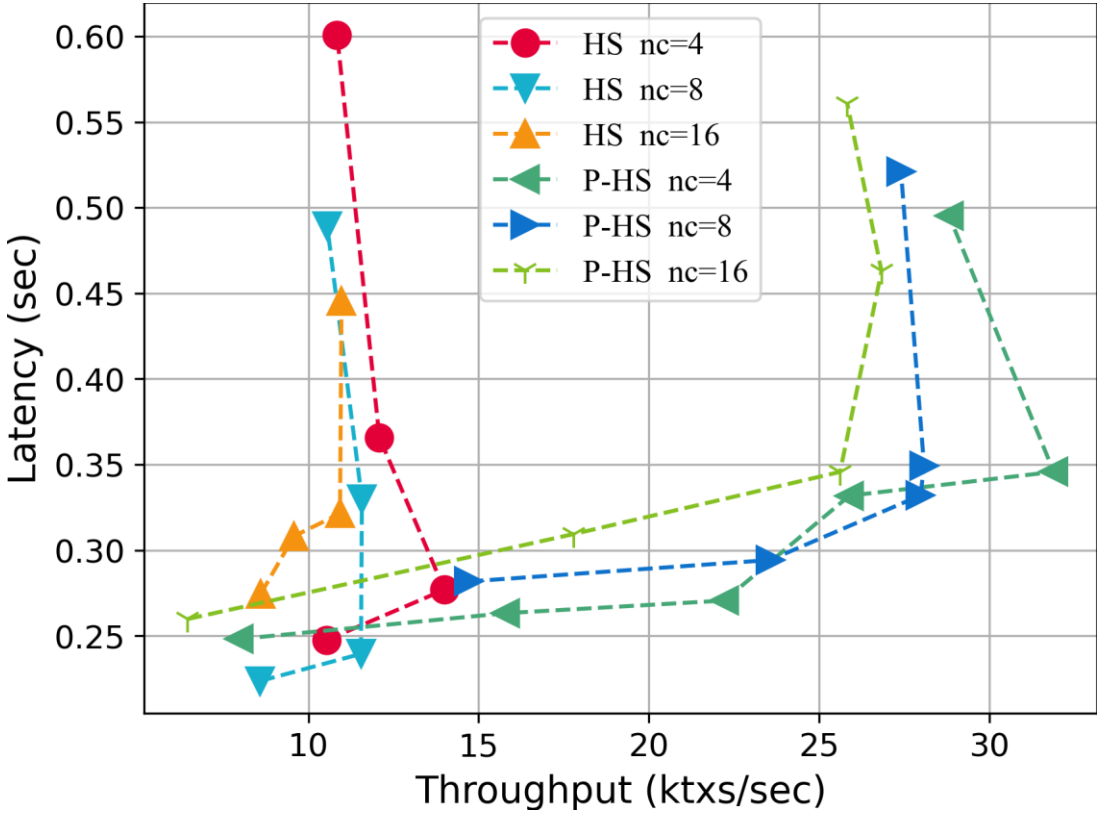
- **High Robustness**: Each node can have several parent nodes. Even if some of the parent nodes fail, a node can still receive complete data.

- **Low Latency**: When a new block is produced, the bundles it contains has already been distributed to other nodes, a tiny ***Predis*** block can be distributed to the network in a short time.

Evaluation- *Predis*

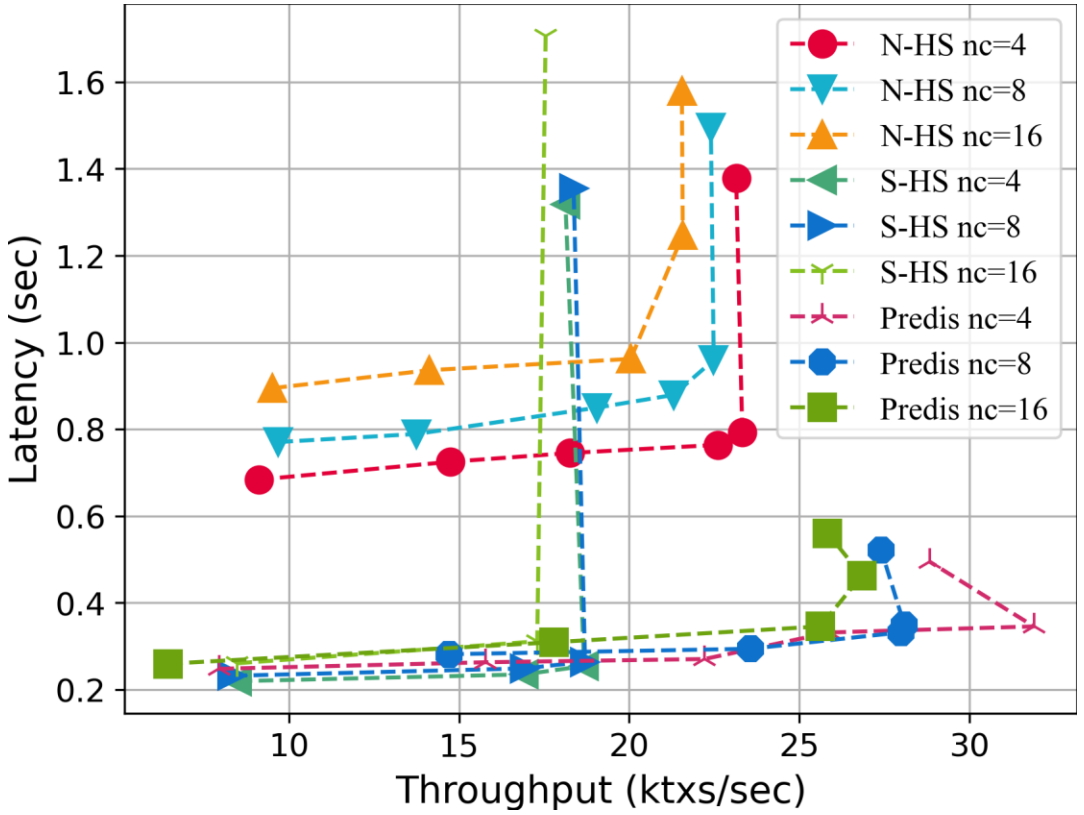
- ***Predis'*** improvement on PBFT and HotStuff.
- 100 Mbps at WAN environment in 4 regions.
- n_c means the number of consensus nodes.



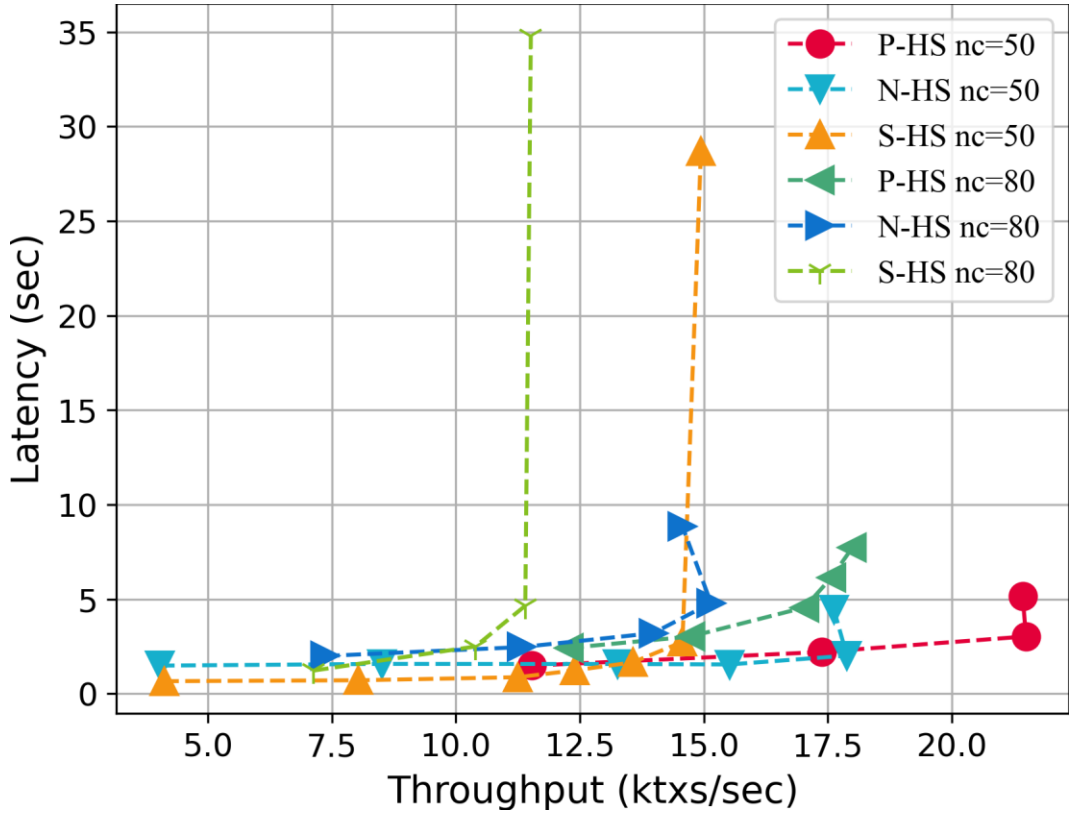


Evaluation- *Predis*

- ***Predis*** vs SOTA (Narwhal and Stratus).
- n_c means the number of consensus nodes.



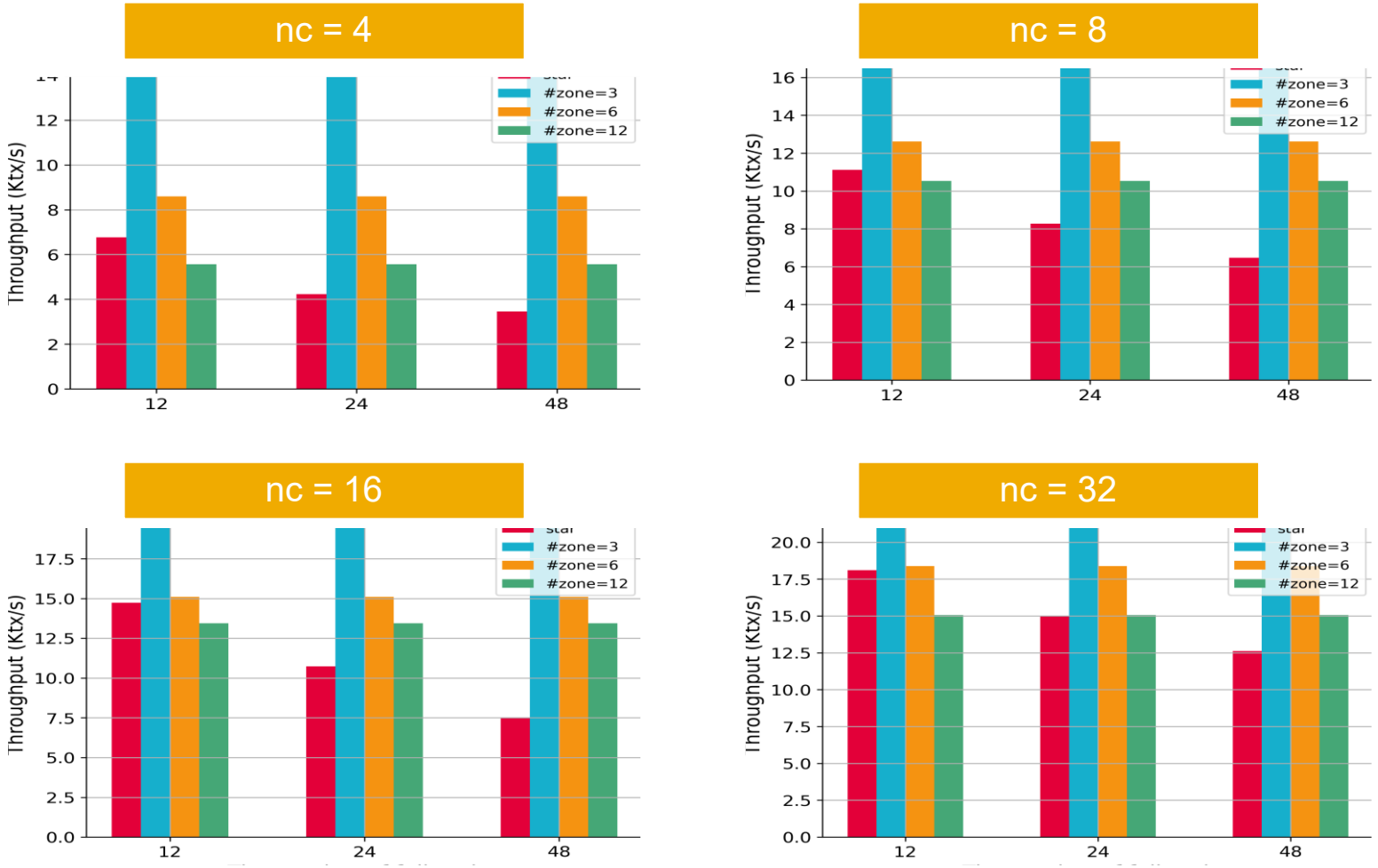
In the WAN environment



In the LAN environment

Evaluation- *Multi-Zone*

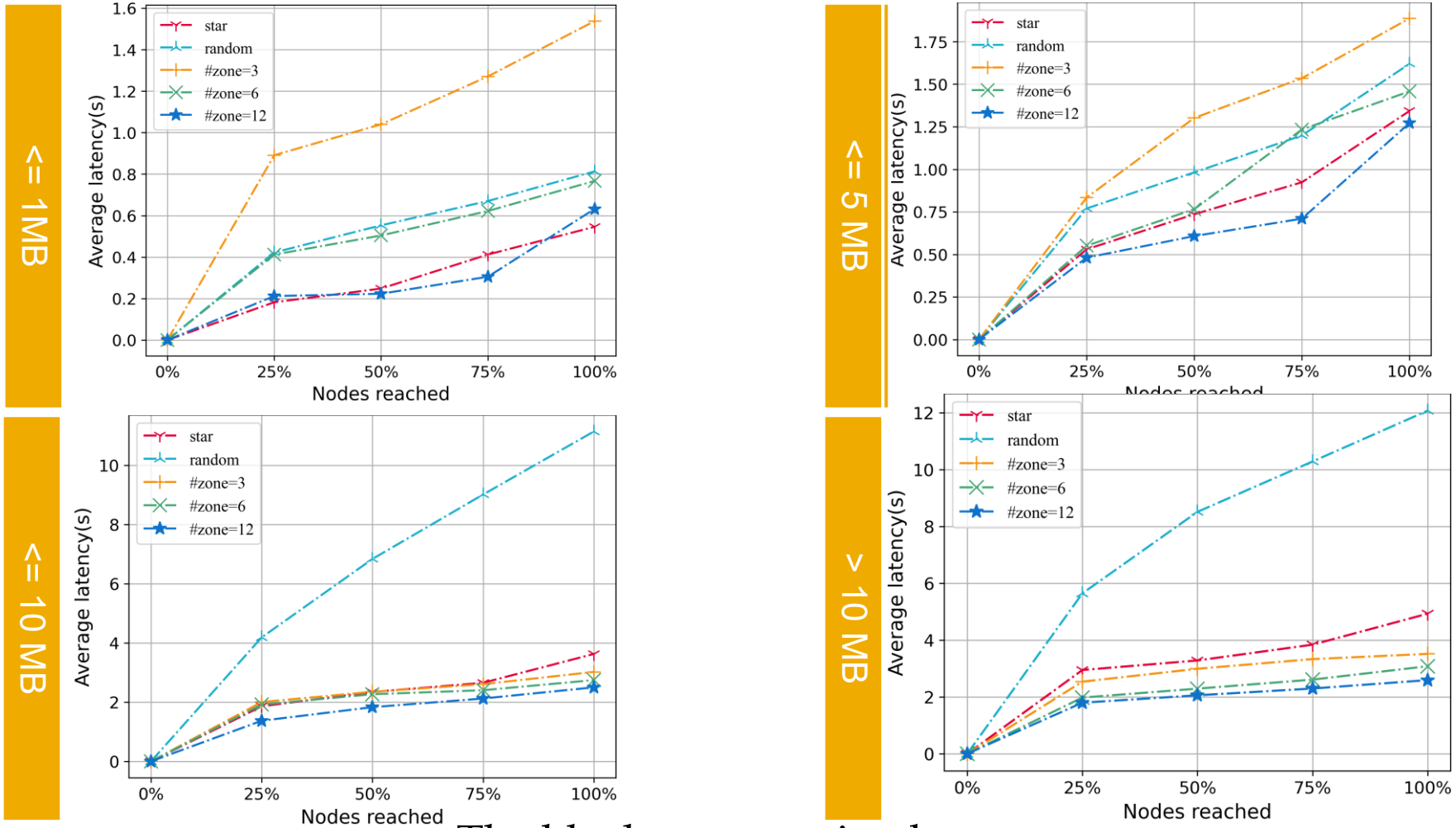
- ***Multi-Zone*** vs Star topology (on throughput).



The throughput of the consensus layer.

Evaluation- *Multi-Zone*

- ***Multi-Zone*** vs Star topology and random topology (**FEG ICDCS 2020**).



The block propagation latency.

Conclusion

- We propose a new data flow framework （***Predis*** and ***Multi-Zone***）with high throughput and low latency for blockchain network.

- We show that ***Predis*** can scale to large consensus group with a high throughput and ***Multi-Zone*** can transmit large blocks with low latency in large scale blockchain network.

- We hereby provide a novel viewpoint to design the data flow strategy for the large-scale blockchain network.

Thank you
Q&A?