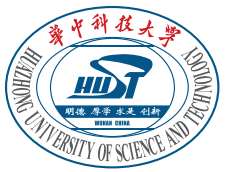
2026 Network and Distributed System Security Symposium (NDSS)

Icarus: Achieving Performant Asynchronous BFT with Only Optimistic Paths

Xiaohai Dai*, Yiming Yu*, Sisi Duan†, Rui Hao‡, Jiang Xiao*, and Hai Jin*

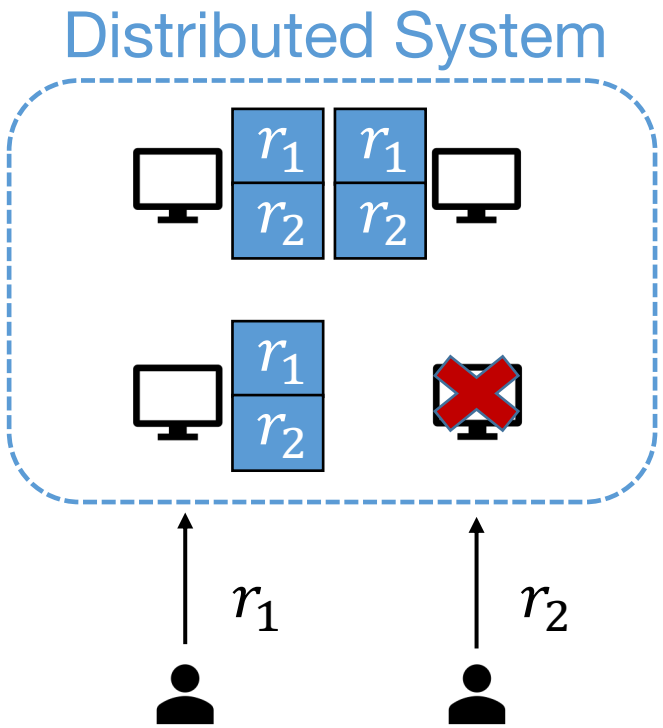
*Huazhong University of Science and Technology, China
†Tsinghua University, China
‡Wuhan University of Technology, China



Background

Consensus

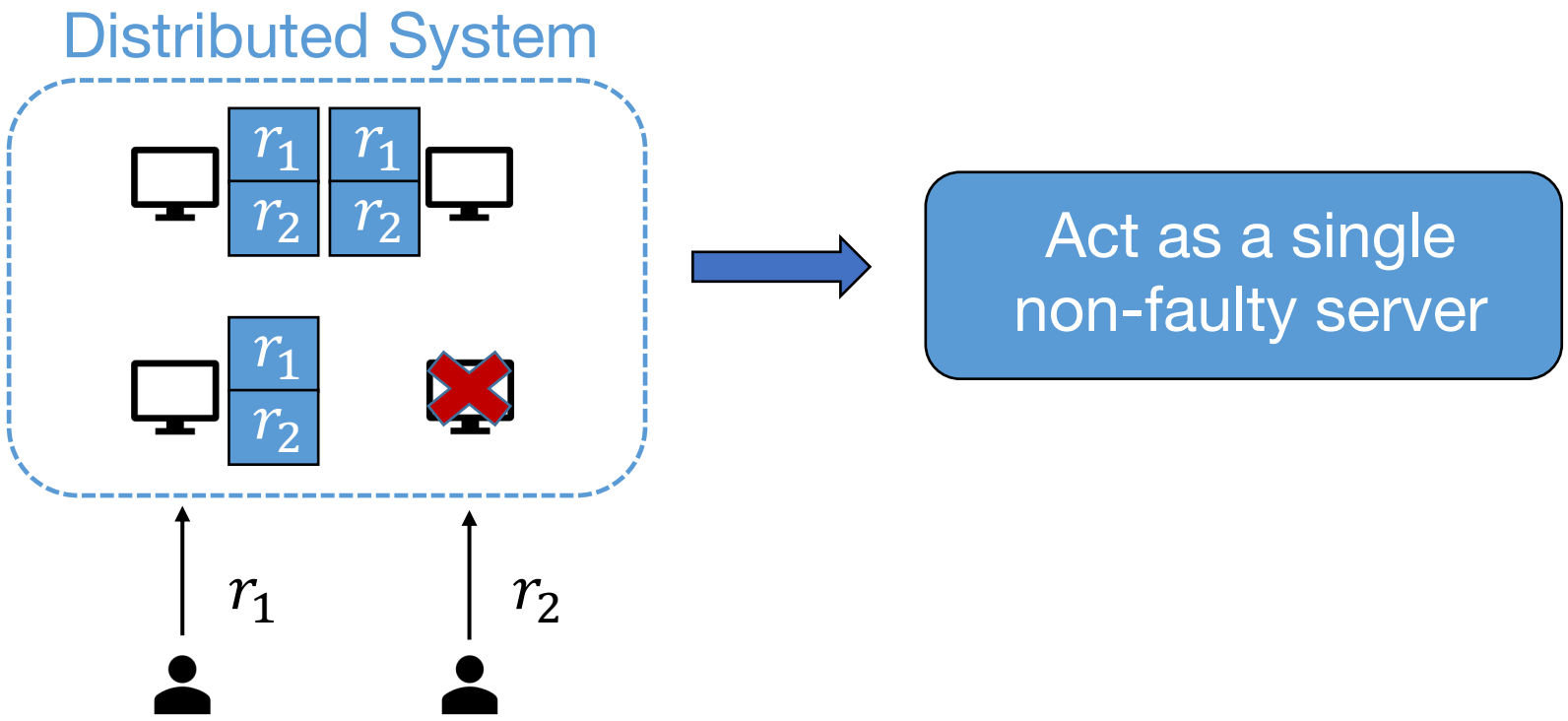
- Reach an agreement on requests even some nodes are Byzantine.



Background

Consensus

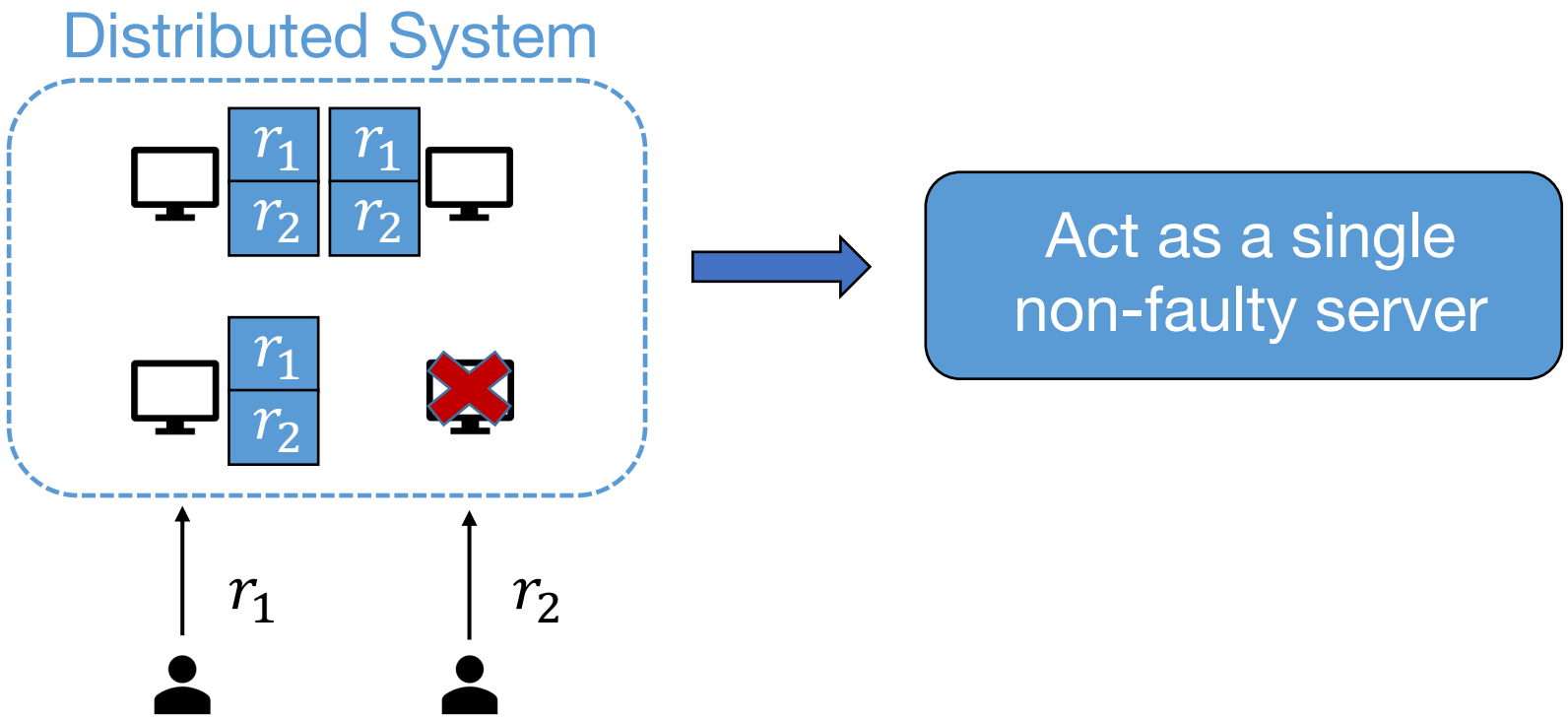
- Reach an agreement on requests even some nodes are Byzantine.



Background

Byzantine Fault Tolerant (BFT) Consensus

- Reach an agreement on requests even some nodes are Byzantine.



Background

Timing assumptions

- **Synchronous network**
  - Messages will be delivered in Δ
- **Partially-synchronous network**
  - Synchronous after GST
- **Asynchronous network**
  - No assumption is made on message delay.

Background

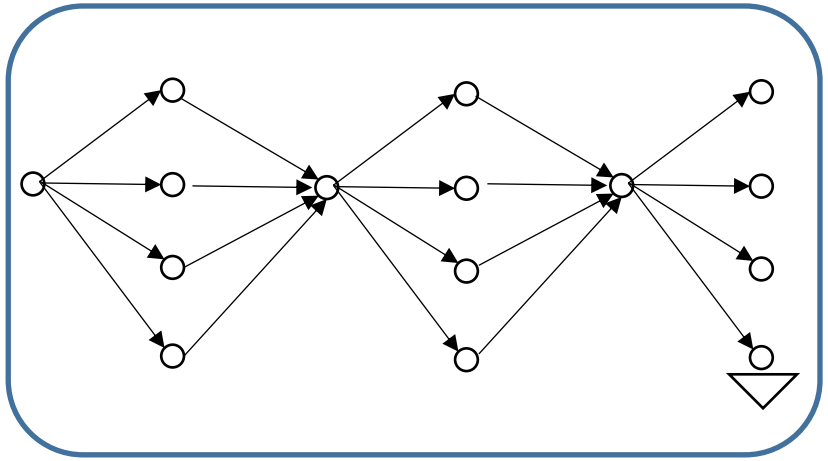
Timing assumptions

- **Synchronous network**
  - Messages will be delivered in Δ
  - **Protocols:** Pili, Sync-HotStuff, …
- **Partially-synchronous network**
  - Synchronous after GST
  - **Protocols:** PBFT, HotStuff, SBFT, …
- **Asynchronous network**
  - No assumption is made on message delay.
  - **Protocols:** HoneybadgerBFT, Dumbo, Tusk, MVBA, …

Background

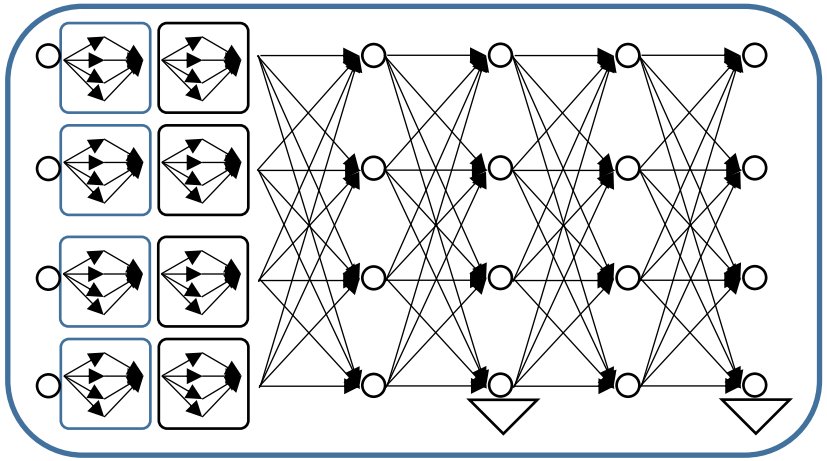
Pros and cons.

(Partially-)synchronous protocols



- **Pros**: High efficiency
- **Cons**: **Weak robustness**

Asynchronous protocols



- **Pros**: Strong robustness
- **Cons**: **Low efficiency**

Background

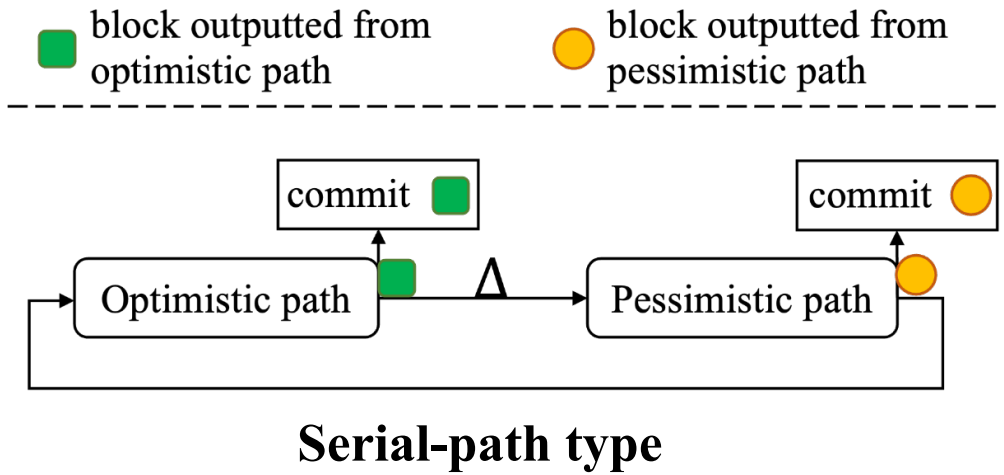
Dual-path paradigm: optimistic path + pessimistic path

- **Optimistic path:** partially synchronous protocol
- **Pessimistic path:** purely asynchronous protocol

Background

Dual-path paradigm: optimistic path + pessimistic path

- **Serial-path:** execute two paths in serial[1][2]



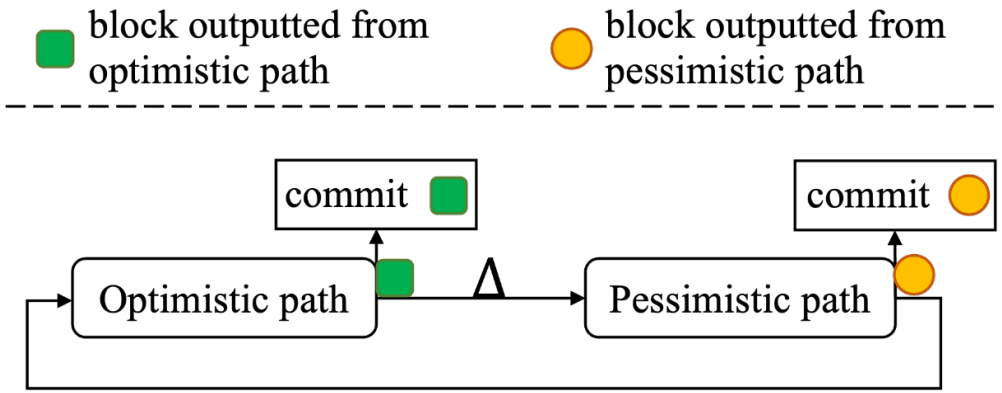
Serial-path type

[1] Bolt-dumbo transformer: Asynchronous consensus as fast as the pipelined bft. CCS'22
[2] Jolteon and ditto: Network-adaptive efficient consensus with asynchronous fallback. FC'22

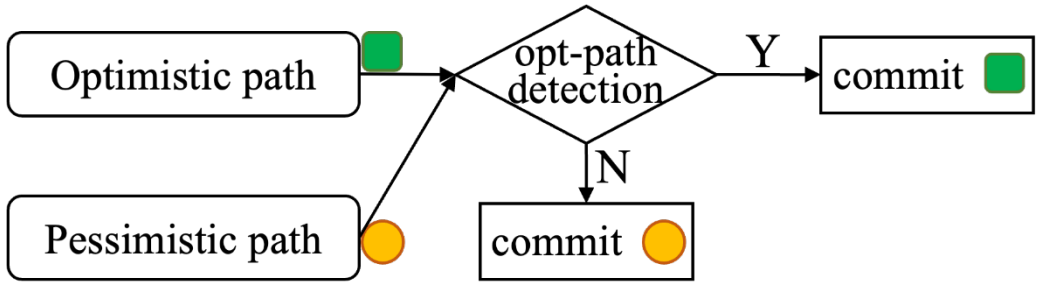
Background

Dual-path paradigm: optimistic path + pessimistic path

- **Serial-path:** execute two paths in serial[1][2]
- **Parallel-path:** execute two paths in parallel[3][4]



Serial-path type



Parallel-path type

[1] Bolt-dumbo transformer: Asynchronous consensus as fast as the pipelined bft. CCS'22
[2] Jolteon and ditto: Network-adaptive efficient consensus with asynchronous fallback. FC'22
[3] ParBFT: Faster Asynchronous BFT Consensus with a Parallel Optimistic Path. CCS'23
[4] Abraxas: Throughput-efficient hybrid asynchronous consensus. CCS'23

Motivation

Limitations of dual-path paradigm

- Employ MVBA as the pessimistic path
  - Structurally complex
  - High overhead
  - Low throughput

Motivation

Limitations of dual-path paradigm

- Employ MVBA as the pessimistic path
  - Structurally complex
  - High overhead
  - Low throughput

Limitations of serial-path type

- Rely on a good **estimation of Δ**
- Improper Δ causes significant **performance degradation**

Motivation

Limitations of dual-path paradigm

- Employ MVBA as the pessimistic path
  - Structurally complex
  - High overhead
  - Low throughput

Limitations of serial-path type

- Rely on a good **estimation of Δ**
- Improper Δ causes significant **performance degradation**

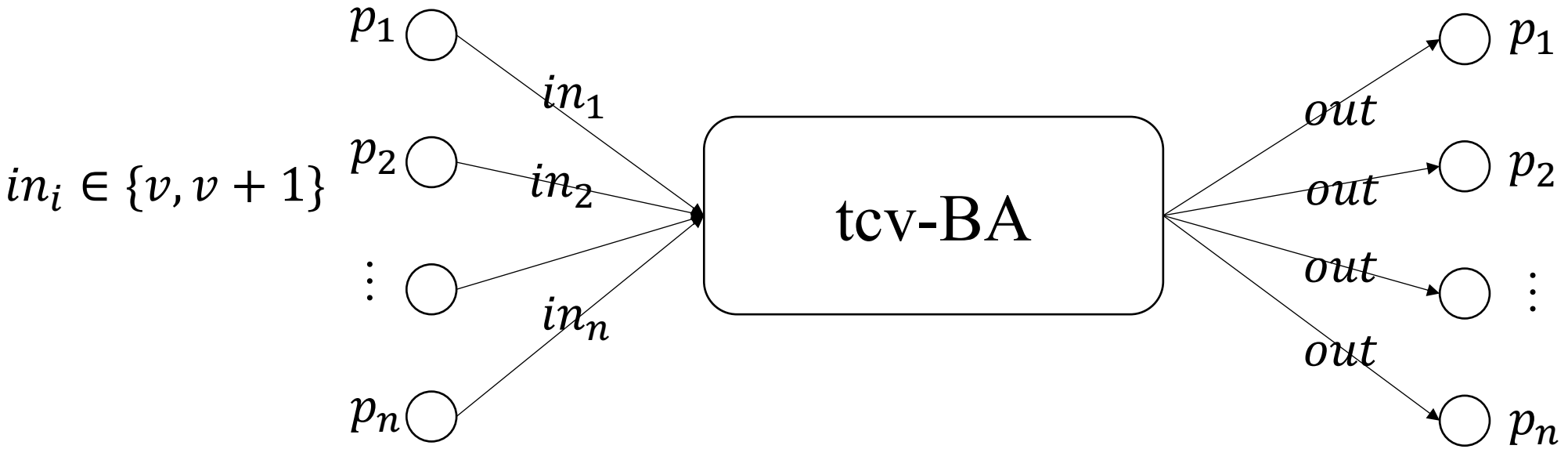
Limitations of parallel-path type

- Commit **only blocks from the optimistic path** under favorable conditions
- **Resource waste** due to the redundant execution of the pessimistic path

Preliminaries

tcv-BA (two-consecutive-value Byzantine Agreement) [1]

- A variant of ABA (asynchronous binary agreement)
- **Input**: a value in $S = \{v, v+1\}$
- **Output**: a consistent value

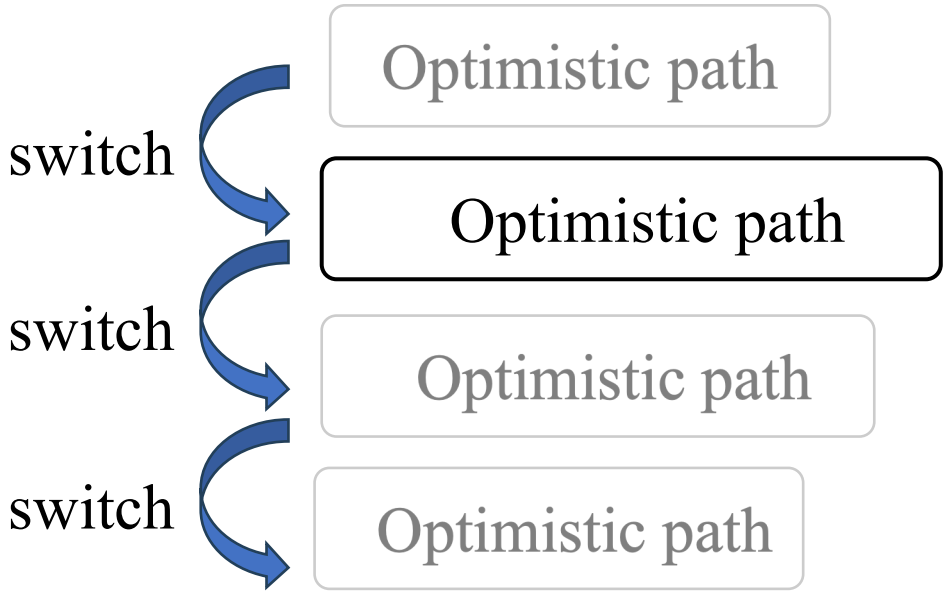


[1] Bolt-Dumbo transformer: Asynchronous consensus as fast as the pipelined BFT. CCS'22

New Paradigm: Only Optimistic Paths

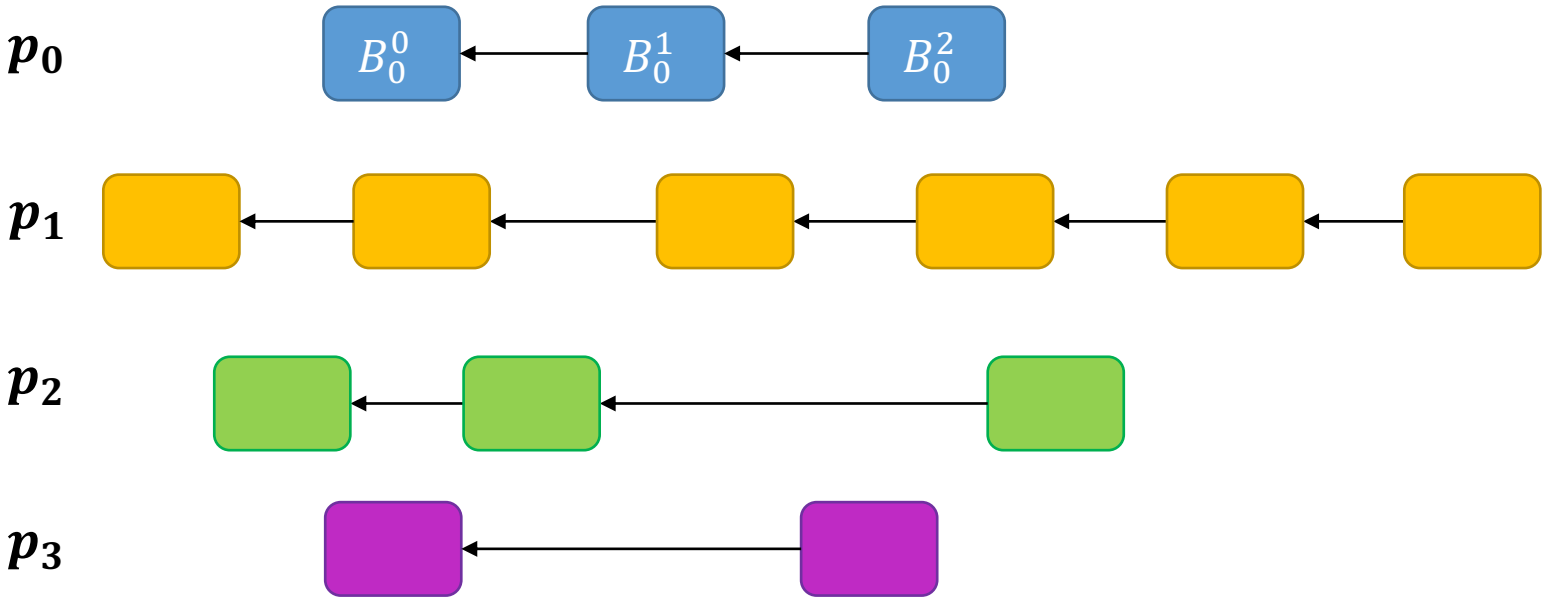
An alternative paradigm: only optimistic paths

- Remove pessimistic path
- Switch to another optimistic path when the current optimistic path fails



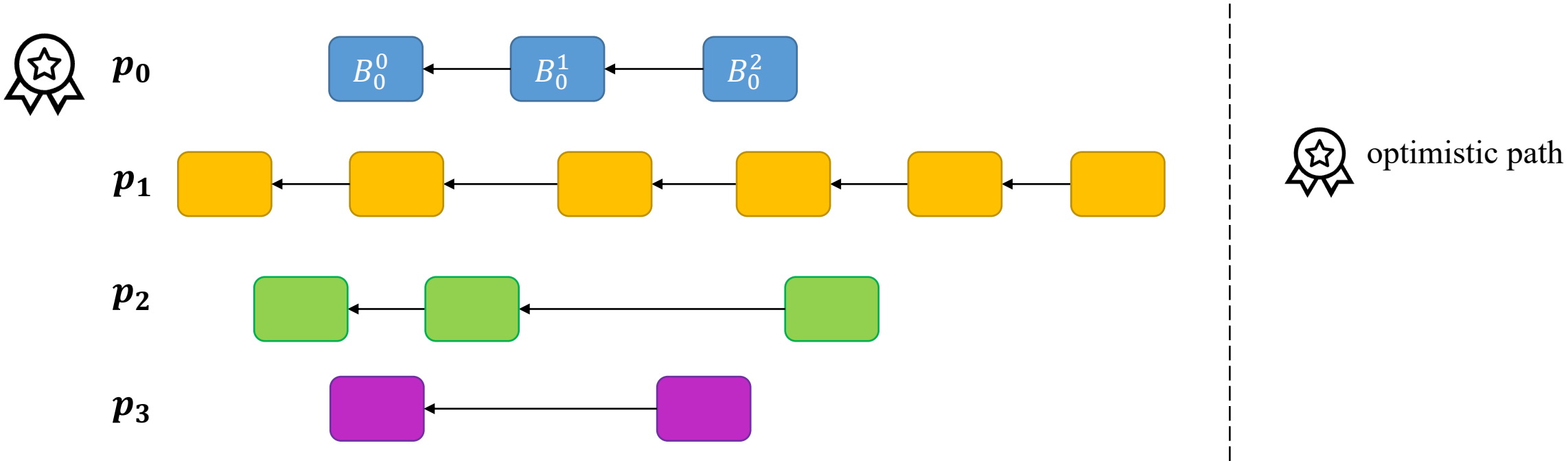
Icarus Design

- **Parallel chains with path rotation**
  - Each node continuously **broadcasts a chain** linked by QC (Quorum Certificate)



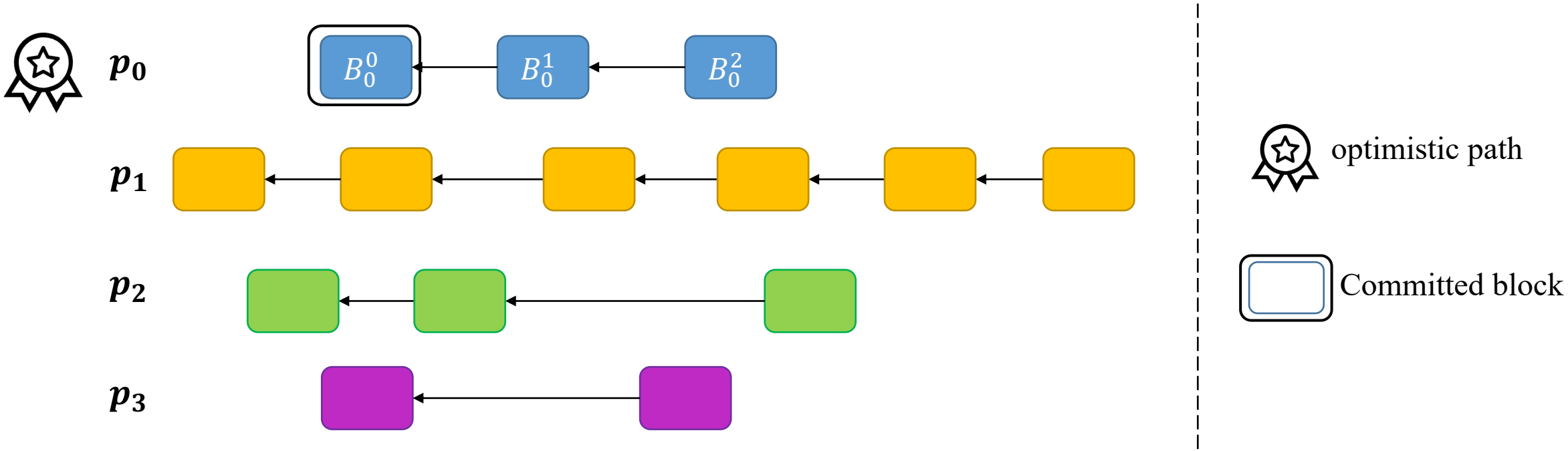
Icarus Design

- **Parallel chains with path rotation**
  - Each node continuously **broadcasts a chain** linked by QC (Quorum Certificate)
  - Chains **take turn** serving as the optimistic paths



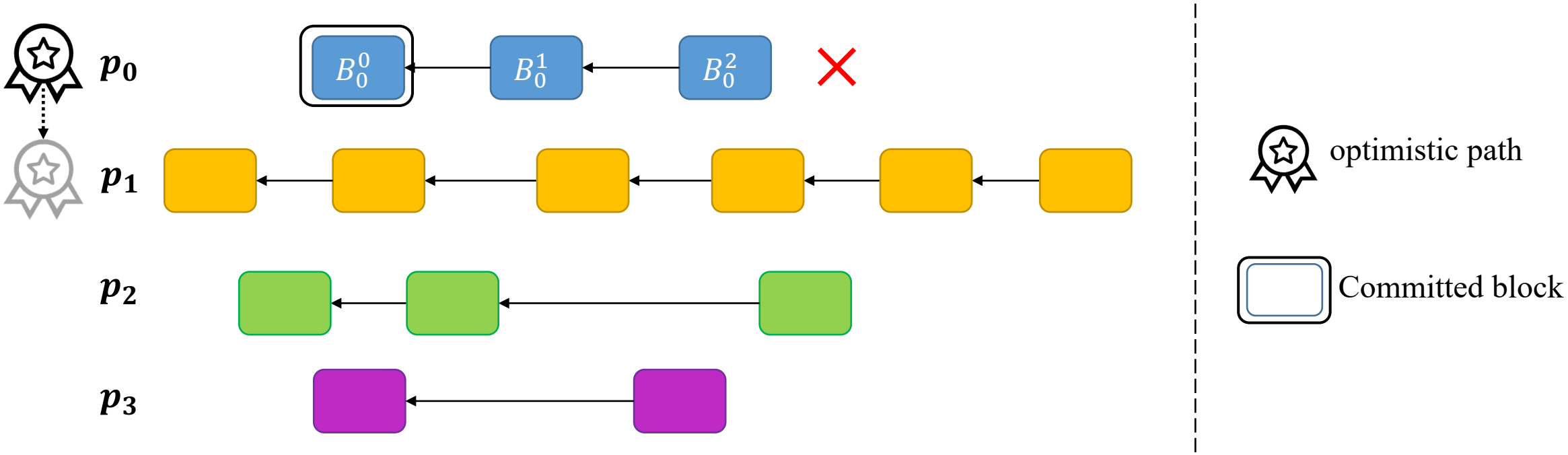
Icarus Design

- **Parallel chains with path rotation**
  - Each node continuously **broadcasts a chain** linked by QC (Quorum Certificate)
  - Chains **take turn** serving as the optimistic paths
  - Blocks on the optimistic path can be committed through **two-phase rule**



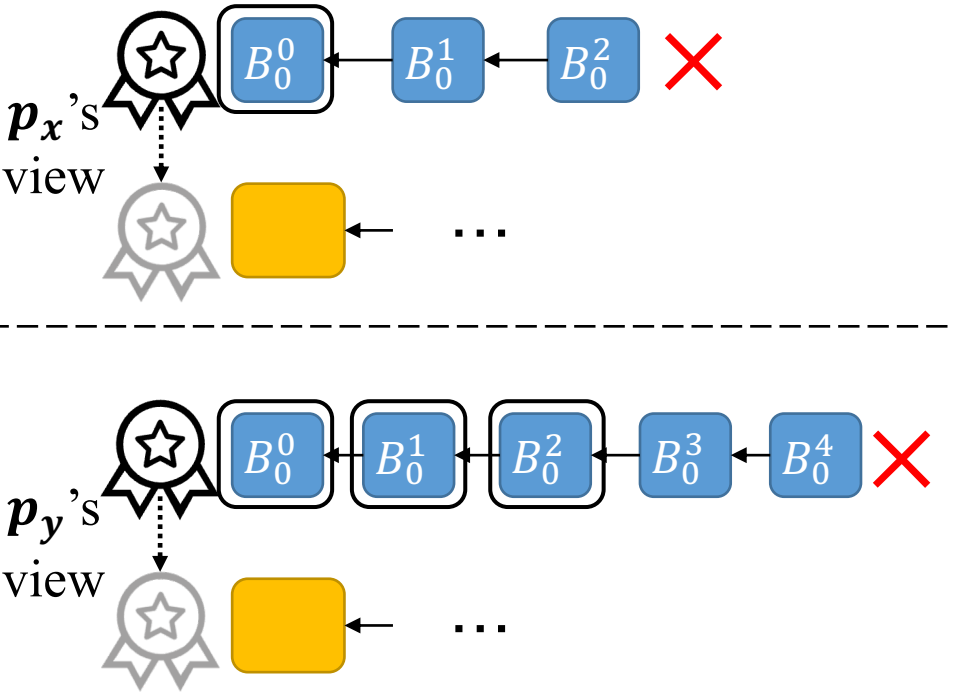
Icarus Design

- **Parallel chains with path rotation**
  - Each node continuously **broadcasts a chain** linked by QC (Quorum Certificate)
  - Chains **take turn** serving as the optimistic paths
  - Blocks on the optimistic path can be committed through **two-phase rule**



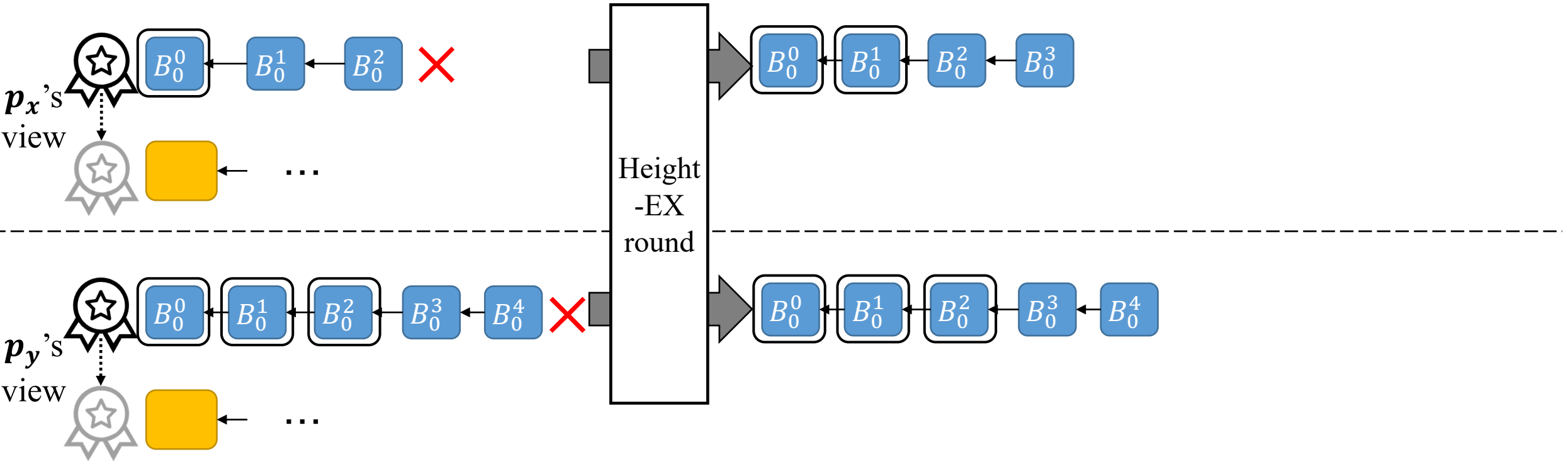
Icarus Design

- **Leverage tcv-BA to align heights before path switching**
  - **Challenge:** nodes may initiate switching at different heights



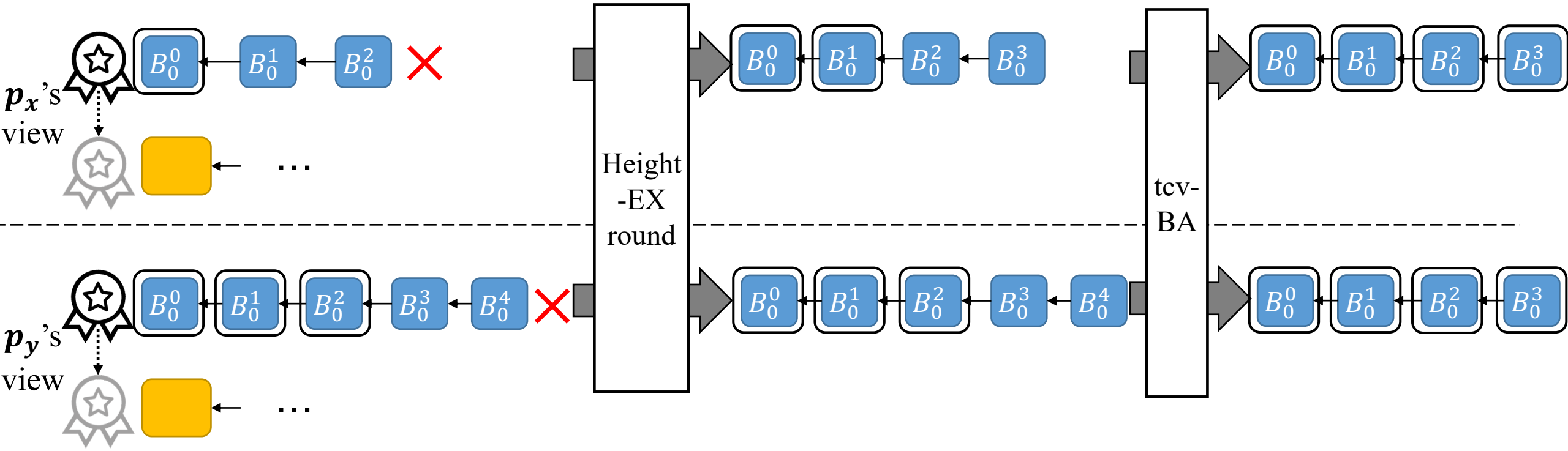
Icarus Design

- **Leverage tcv-BA to align heights before path switching**
  - **Challenge:** nodes may initiate switching at different heights
  - Conduct a **height-exchange round** to reduce heights to two consecutive values



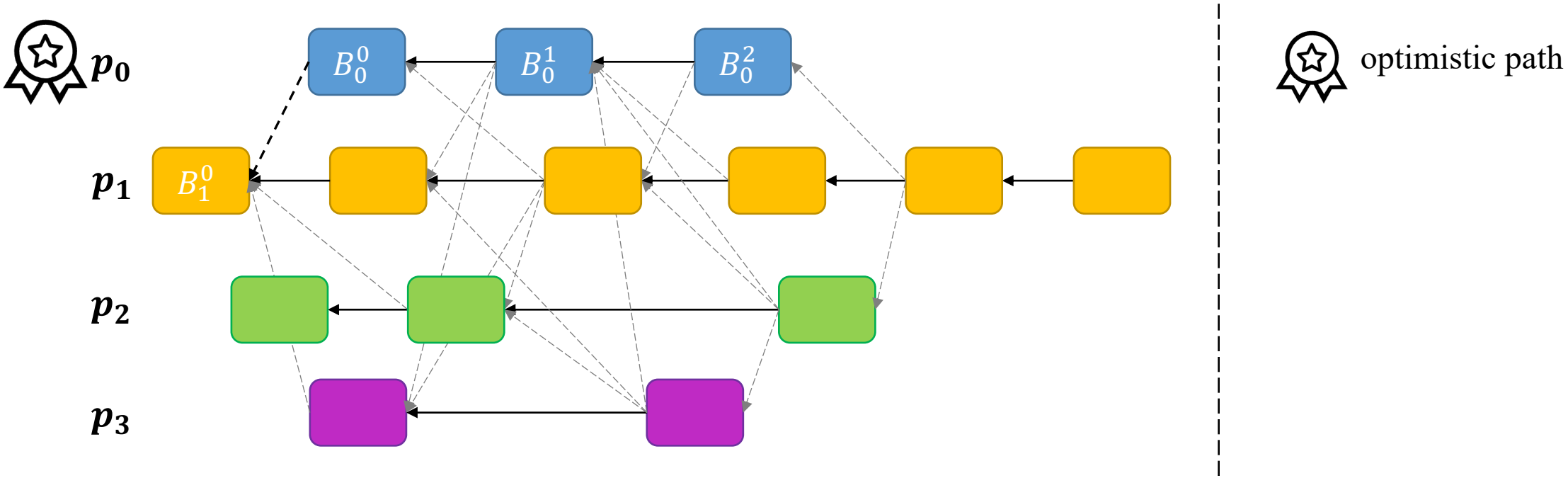
Icarus Design

- **Leverage tcv-BA to align heights before path switching**
  - **Challenge:** nodes may initiate switching at different heights
  - Conduct a **height-exchange round** to reduce heights to two consecutive values
  - Leverage **tcv-BA** to align heights and commit pending blocks before switching



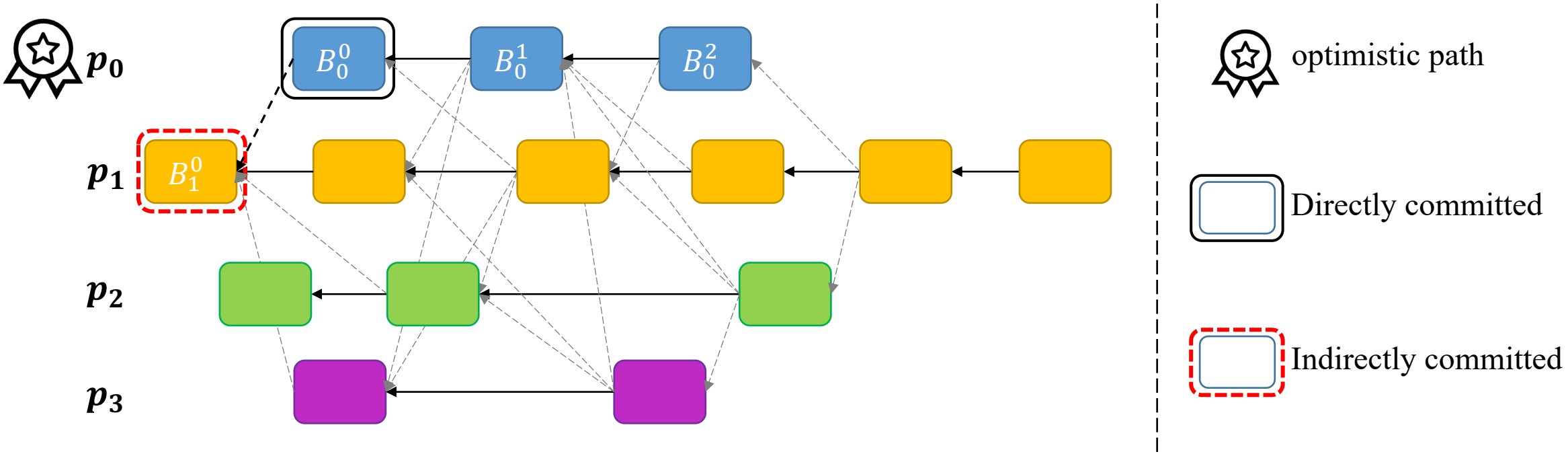
Icarus Design

- **Inter-chain references for high throughput**
  - Each block also references **blocks from other chains**



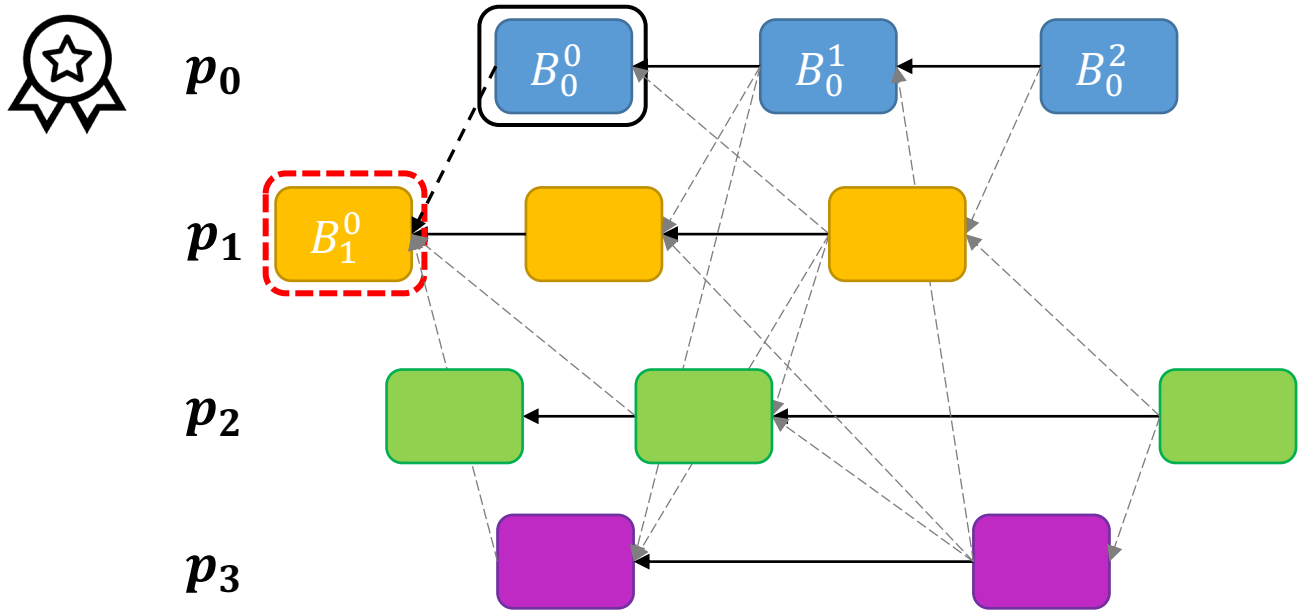
Icarus Design

- **Inter-chain references for high throughput**
  - Each block also references **blocks from other chains**
  - Committing a block triggers committing of its referenced blocks from other chains



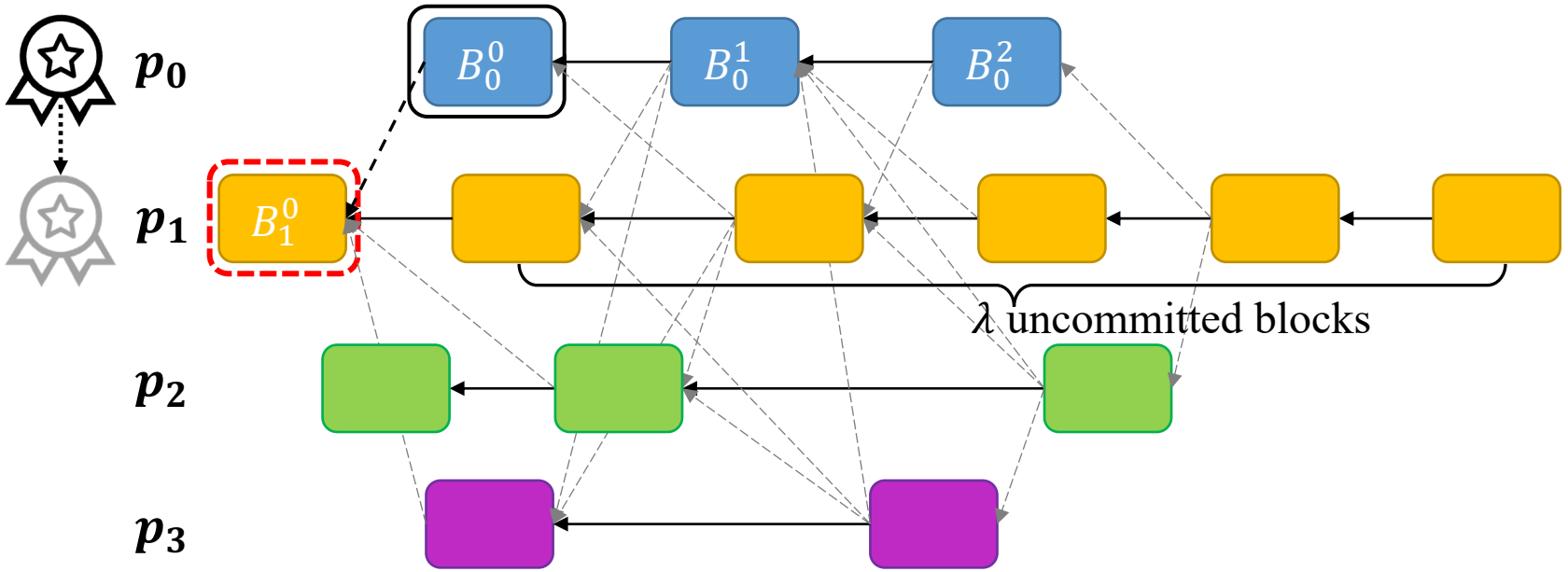
Icarus Design

- **Counting uncommitted blocks on other chains to initiate path switching**
  - If the **opt-path works well**, the number of uncommitted blocks on each chain is **low**



Icarus Design

- **Counting uncommitted blocks on other chains to initiate path switching**
  - If the **opt-path works well**, the number of uncommitted blocks on each chain is **low**
  - Conversely, if any non-opt-path chain accumulates λ **uncommitted blocks**, the system is prompted to **initiate a path rotation**.



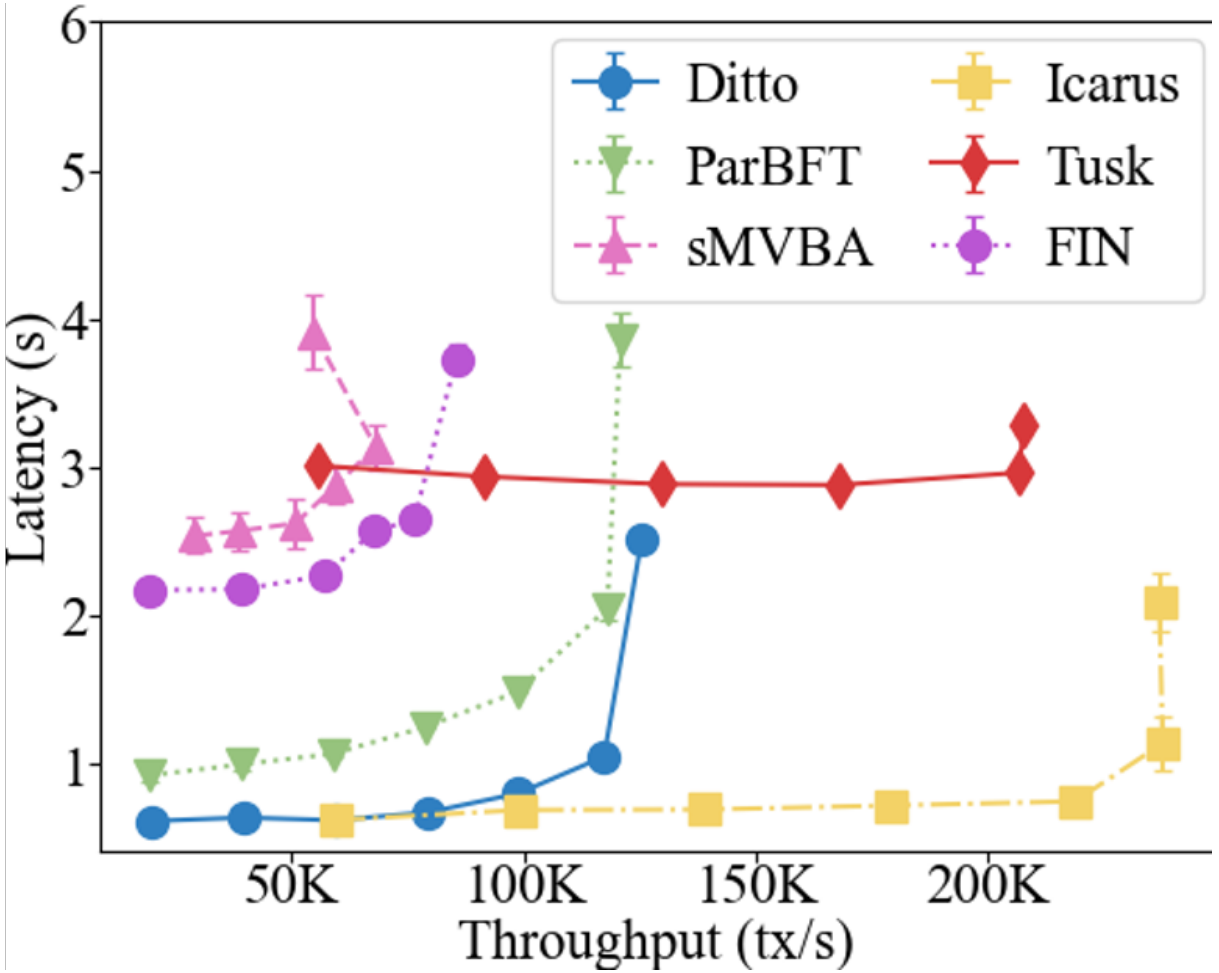
Evaluation

- **Implementation and baseline**
  - **Dual-path:** Ditto & ParBFT
  - **Purely asynchronous:** sMVBA & FIN
  - **DAG-based:** Tusk
  - All protocols are implemented in Rust using a unified framework

- **Experimental details**
  - AWS m5d.xlarge EC2 instance.
  - Spanning five regions across the world.
  - **Metrics:** End-to-end latency and throughput

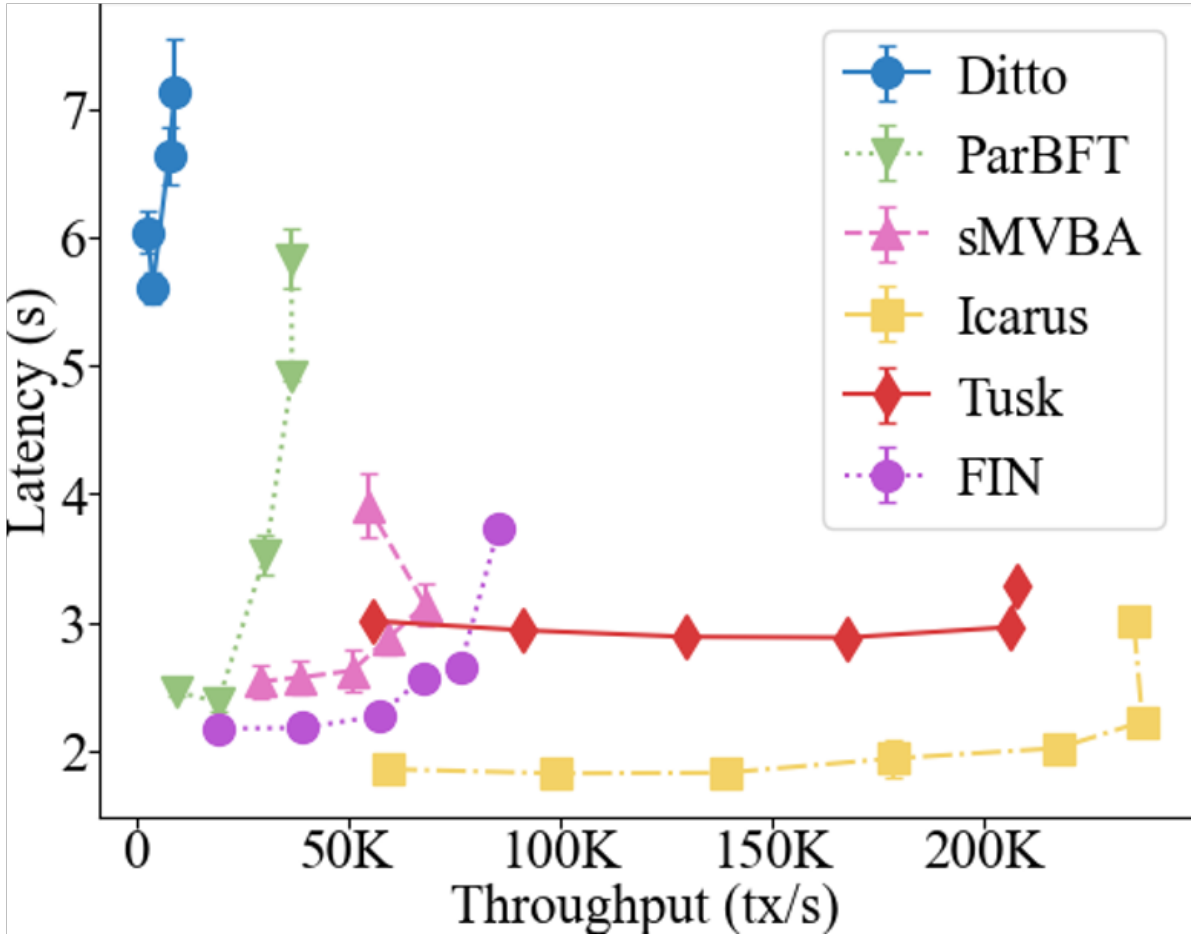
Performance in favorable situations

- **Settings:**
  - Non-faulty nodes
  - A stable network
- **Results:**
  - Icarus achieves the **lowest latency** due to the usage of the **optimistic path**
  - Icarus achieves the **highest throughput**, attributed to its **parallel-chain design**



Performance when leaders are delayed

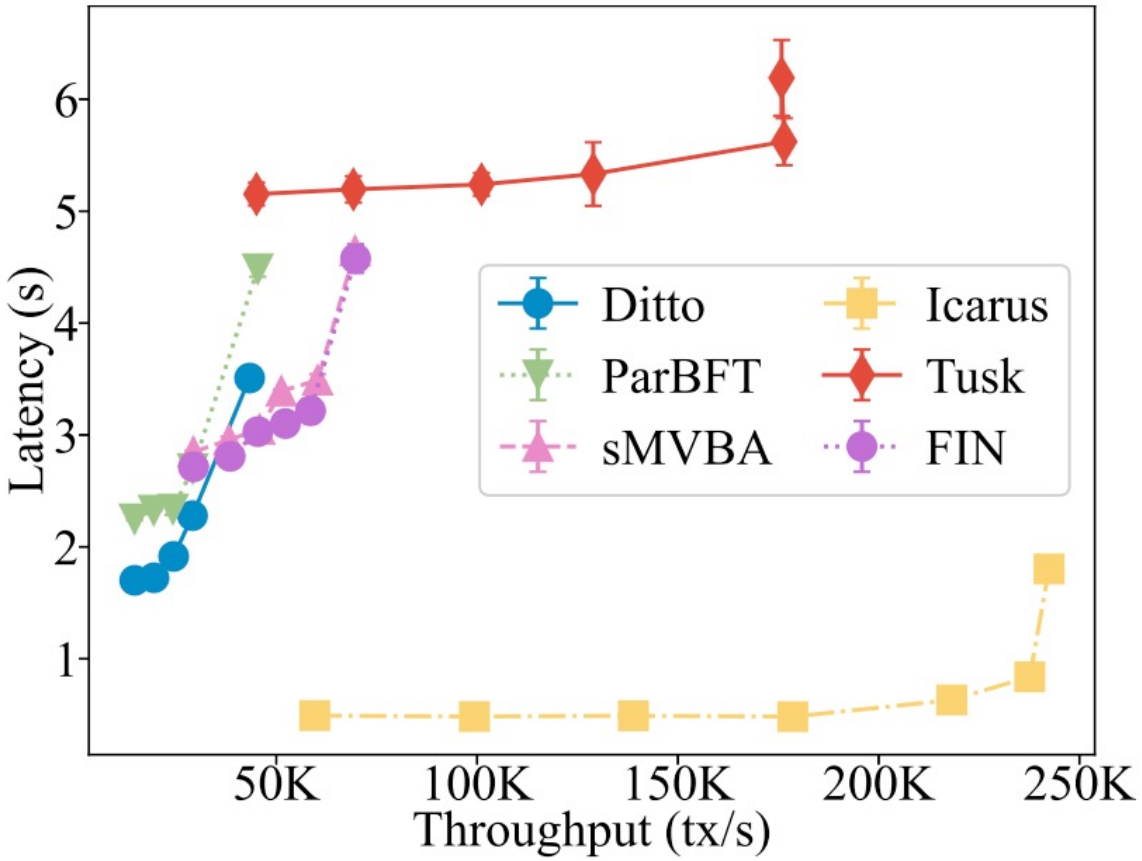
- **Settings:**
  - Non-faulty nodes
  - Leaders are delayed by 20s
- **Results:**
  - Icarus maintains the **highest throughput** because the growth of other chains is **unaffected** even when the optimistic path fails
  - Icarus still achieves the **lowest latency** by **immediately** committing blocks on **new optimistic paths**



Performance with a rapid node

- **Settings:**
  - Simulate a situation where p_2 produces blocks much faster than others
  - Delay the block broadcasting of all nodes by 500ms, except p_2

- **Results:**
  - Icarus achieves significantly **lower latency and higher throughput** than other protocols
  - This advantage stems from Icarus's ability to **swiftly switch to and remain on** the optimistic path maintained by the faster p_2.



Conclusion

- We identify key limitations in existing dual-path asynchronous BFT protocols, particularly the inherent complexity and inefficiency of their pessimistic path (i.e., MVBA).

- We propose a new paradigm that only exclusively executes the optimistic paths and instantiate it as Icarus.

- We conduct various experiments to evaluate Icarus, whose results demonstrate its superior performance.

Thanks!
Q&A

More questions, please feel free to contact us:
xhdai@hust.edu.cn