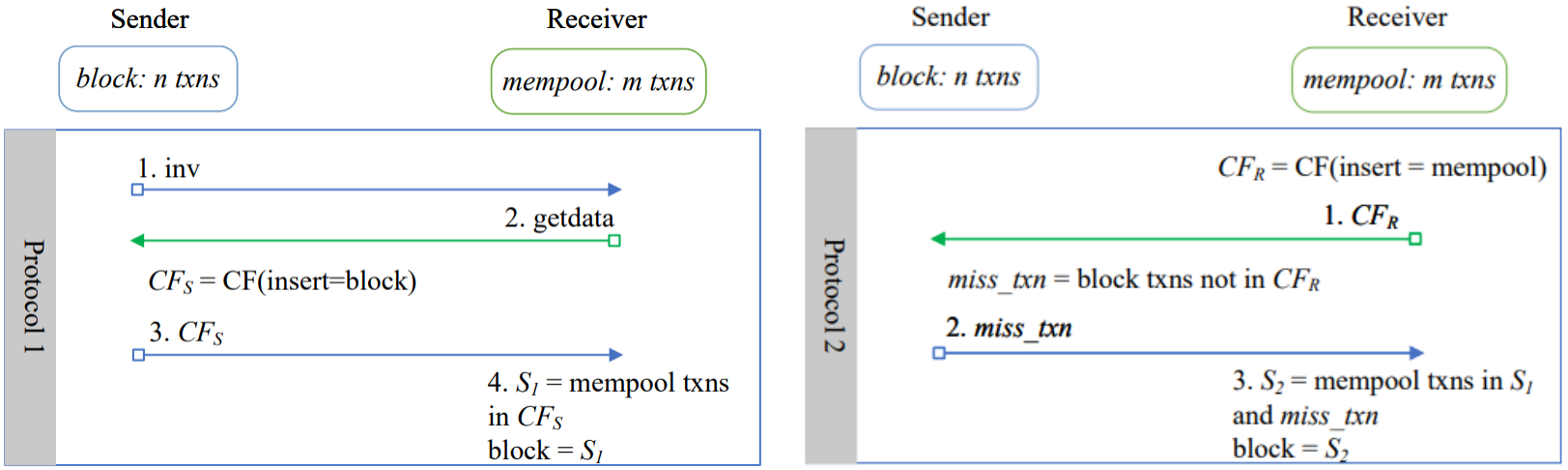
# Gauze: Enabling Communication-Friendly Block Synchronization with Cuckoo Filter

**Xiaoqiang Ding, Liushun Zhao, Lailong Luo, Junjie Xie, Deke Guo, Jinxi Li**

Frontiers of Computer Science, DOI: 10.1007/s11704-022-1685-5

# Problems & Ideas

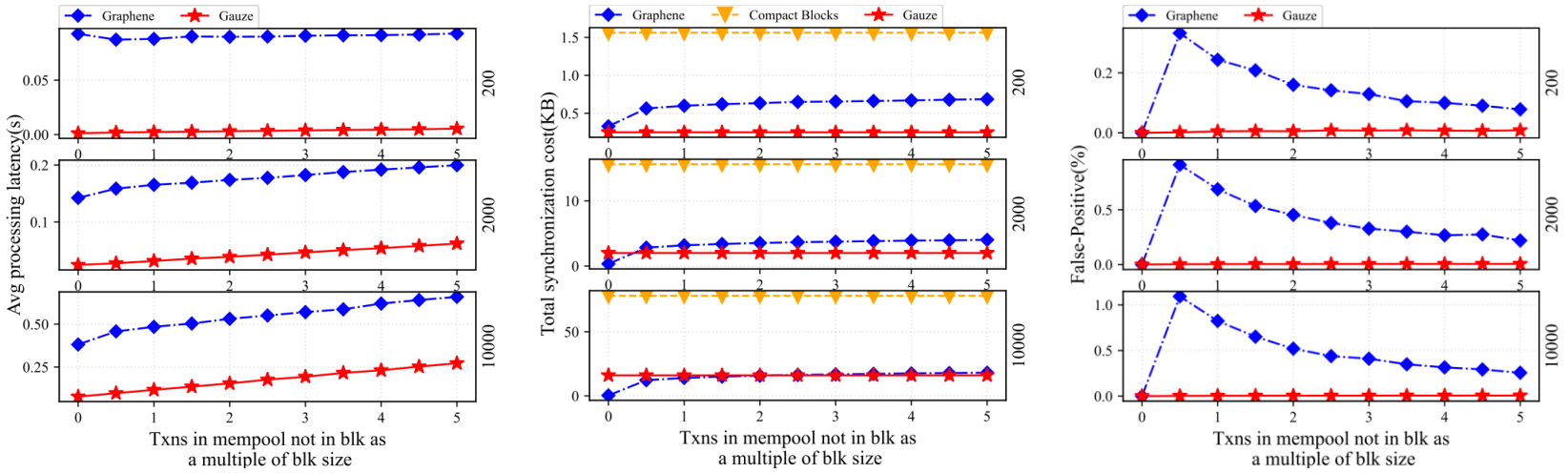
- Problems of conventional Block Synchronization approaches:
  - Traditionally, blockchain systems tend to send all the transactions from one node to another for synchronization.
  - Existing methods either take excessive time, require high network bandwidth or require additional roundtrips to complete block synchronization.
- Ideas: Using the Cuckoo filter to conduct the set reconciliation of the blocks and mempools.




Gauze Protocols. Left: Protocol 1: 1) The sender encodes the block's transactions as $CF_S$. 2) The receiver queries the $CF_S$ to identify which transactions are included in the block; Right: Protocol 2: 1) The receiver encodes the transactions of the mempool as $CF_R$. 2) The sender queries the $CF_R$ to identify the missed transactions of the receiver and adds them to the miss_txn. 3) The receiver uses the sender's $CF_S$ and miss_txn to decode the block.

# Main Contributions

- Contributions:
  - A novel approach for fast block synchronization which utilizes the Cuckoo filter to discern the transactions in the receiver's mempool and the block to verify, providing an efficient solution to the problem of set reconciliation in the P2P network;
  - A method that can save a significant portion of network bandwidth by reducing transmission traffic in the network;
  - A method that provides higher process performance, uses less space than other approaches, and does not require extra roundtrips to accomplish the block synchronization process.




The main results of Gauze compared with the Compact Blocks and Graphene from several aspects. Left: the average processing latency (s); Middle: the average space cost; Right: the average false positive rate with longer *f*.