Distributed Ledger over NDN for A Real-world Solar System

Zhiyi Zhang, Vishrant Vasavada, Randy King, Lixia Zhang

NDNcomm

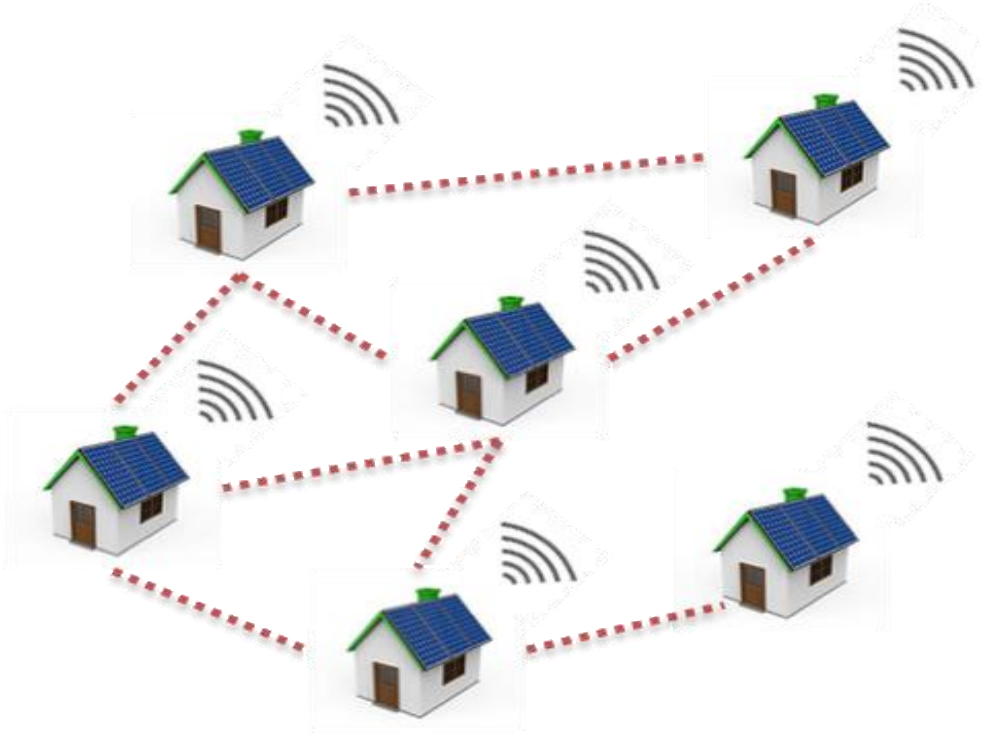
Sept 2018

Operant Solar System based on NDN

- Roof-top mesh networking over low-energy broadcast radio
- Implemented using NDN with geo forwarding
- Operant Solar issues each node a NDN certificate; all the Data packets are signed

Operant Solar:

http://www.operantsolar.com/



Distributed Ledger is needed

- Robust ledger system with the power of publicity: Unalterable, Undeniable, Monitored by the public
- Offers greatly improved security to commercial solar projects
  - Energy Production/Consumption data: business integrity
  - Detect abuse, outage, and intrusion

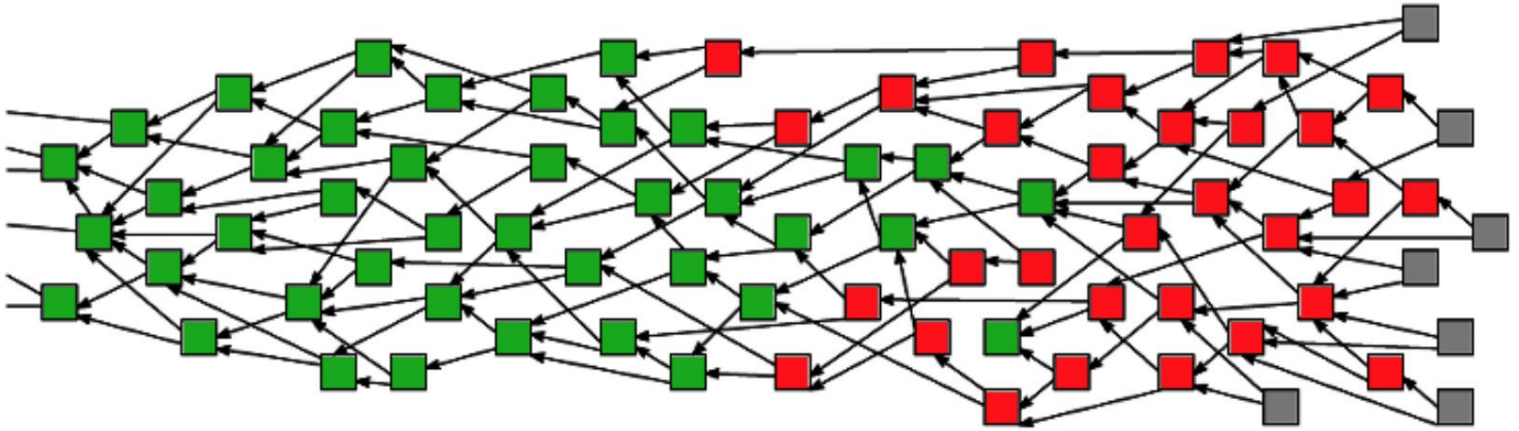
Notice: anonymity is not a big concern here (the company already knows their costumers)

Notice: common use case in many commercial projects, e.g., bank account, medical records

Background (1): Tangle

Tangle is a Directed Acyclic Graph (DAG) used by IOTA (cryptocurrency)

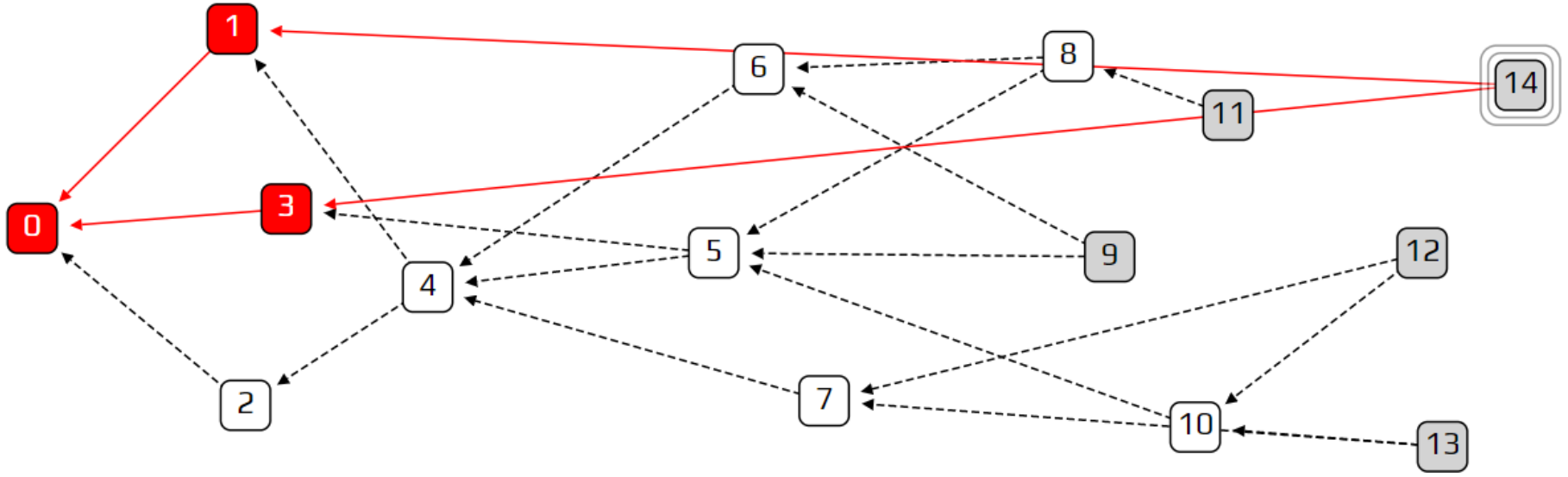
- Each block refers to (approves) two previous blocks
- A block gains weight when it is referred (approved) by later blocks
- A block that has not been referred yet is called a tip
- When a block gains enough weight, it’s accepted by the system



Background (2): Markov Chain Monte Carlo

An algorithm used by IOTA for tip selection

- Random walk from an ancestor node to a tip
- Weighted random walk: is more likely to select a heavy block
  - Prevent ”lazy” tip



NDN Distributed Ledger for Operant Solar

Design Overview

- Build over NDN instead of TCP/IP
- Use Data packet directly as Blocks
- Use Tangle instead of Blockchain
  - Better scalability
  - Ability to recover from Intermittent Connectivity
- Use Proof of Authentication (PoA) instead of Proof of Work (PoW)
  - A node signs the block instead of calculating a hashcash
  - PoA is already given by NDN’s built-in security

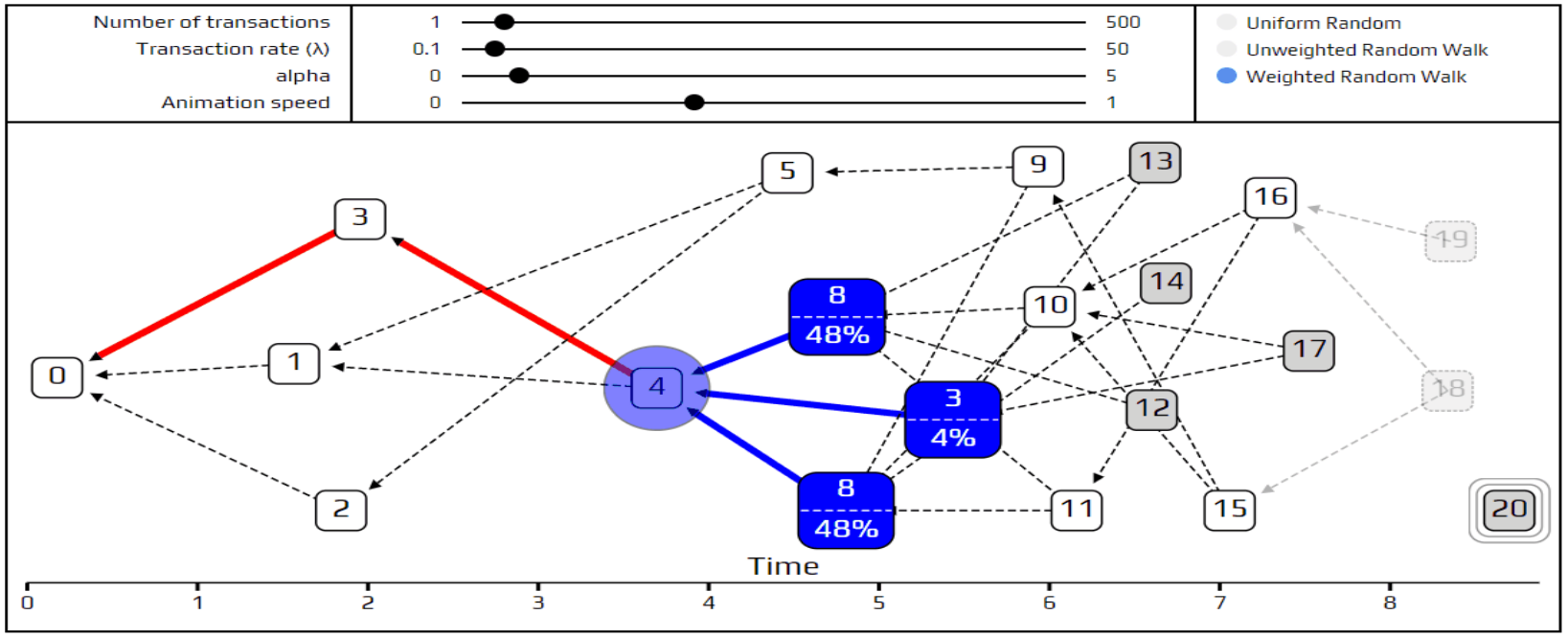
NDN enables difference

- Simple and straightforward design and implementation: “what” instead of “where”
- Data packet as the block
  - Fetch a block directly with its name
  - Light nodes don’t need to rely on full nodes
- Efficient content distribution
  - Broadcast Notification Interest to all peers
  - Data fetching Interest to fetch the new block

When a node wants to append a new Block

Each node periodically append a record block into the Tangle and at the same time, approve two existing tips. A block holds the recent energy production/consumption records

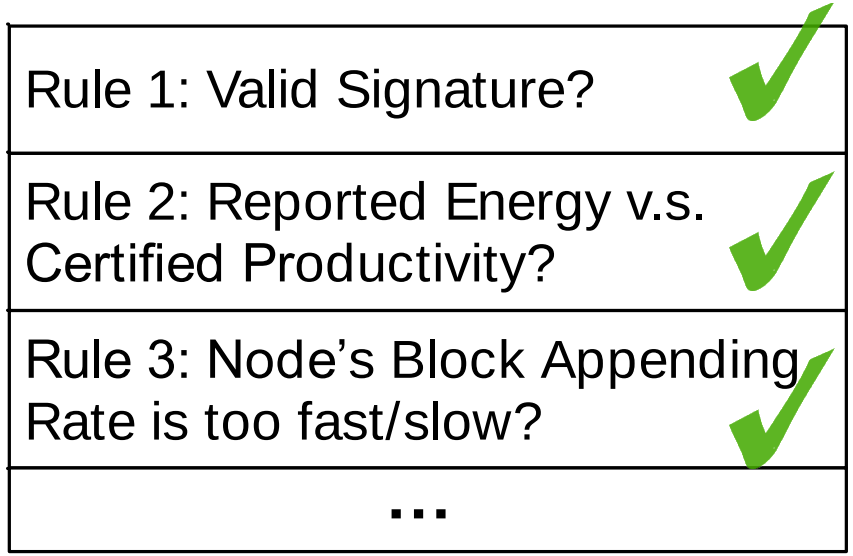
- **Run MCMC and select two tips**



When a node wants to append a new Block

Each node periodically append a record block into the Tangle and at the same time, approve two existing tips. A block holds the recent energy production/consumption records

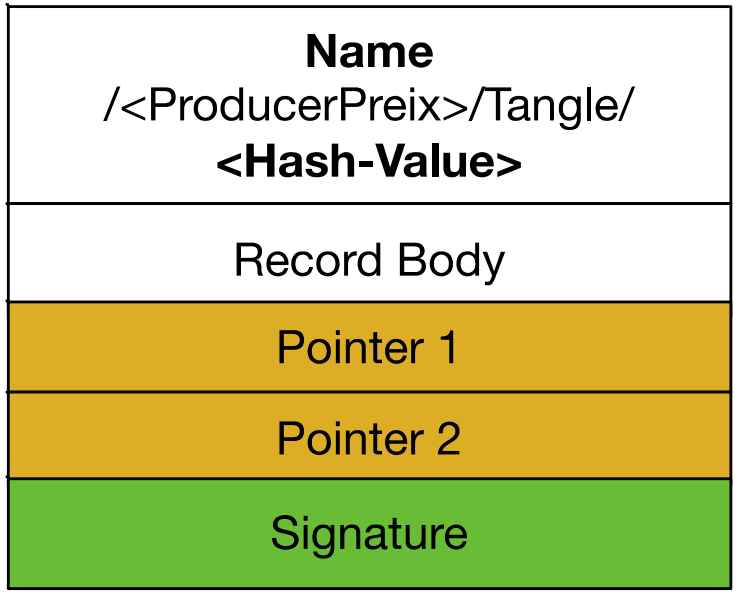
- Run MCMC and select two tips
- **Verify two tips by checking the rules**



When a node wants to append a new Block

Each node periodically append a record block into the Tangle and at the same time, approve two existing tips. A block holds the recent energy production/consumption records

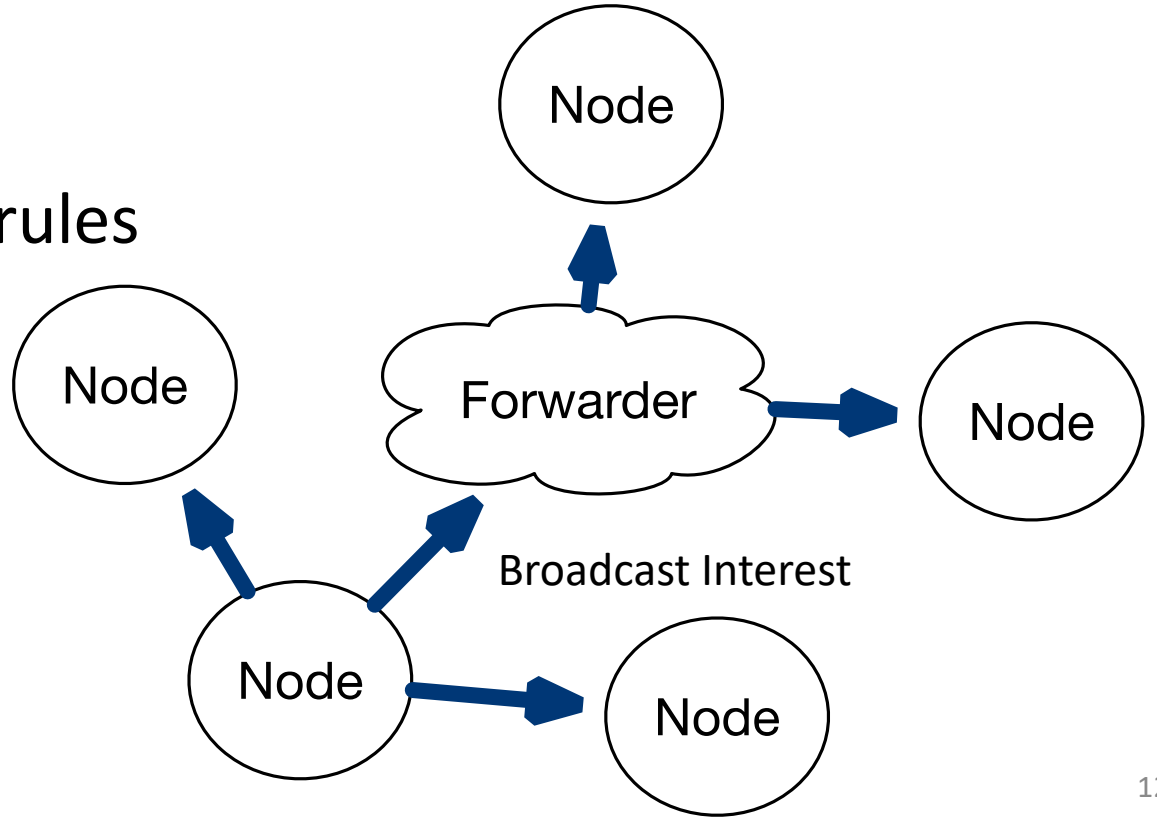
- Run MCMC and select two tips
- Verify two tips by checking the rules
- **Generate block and sign it**



When a node wants to append a new Block

Each node periodically append a record block into the Tangle and at the same time, approve two existing tips. A block holds the recent energy production/consumption records

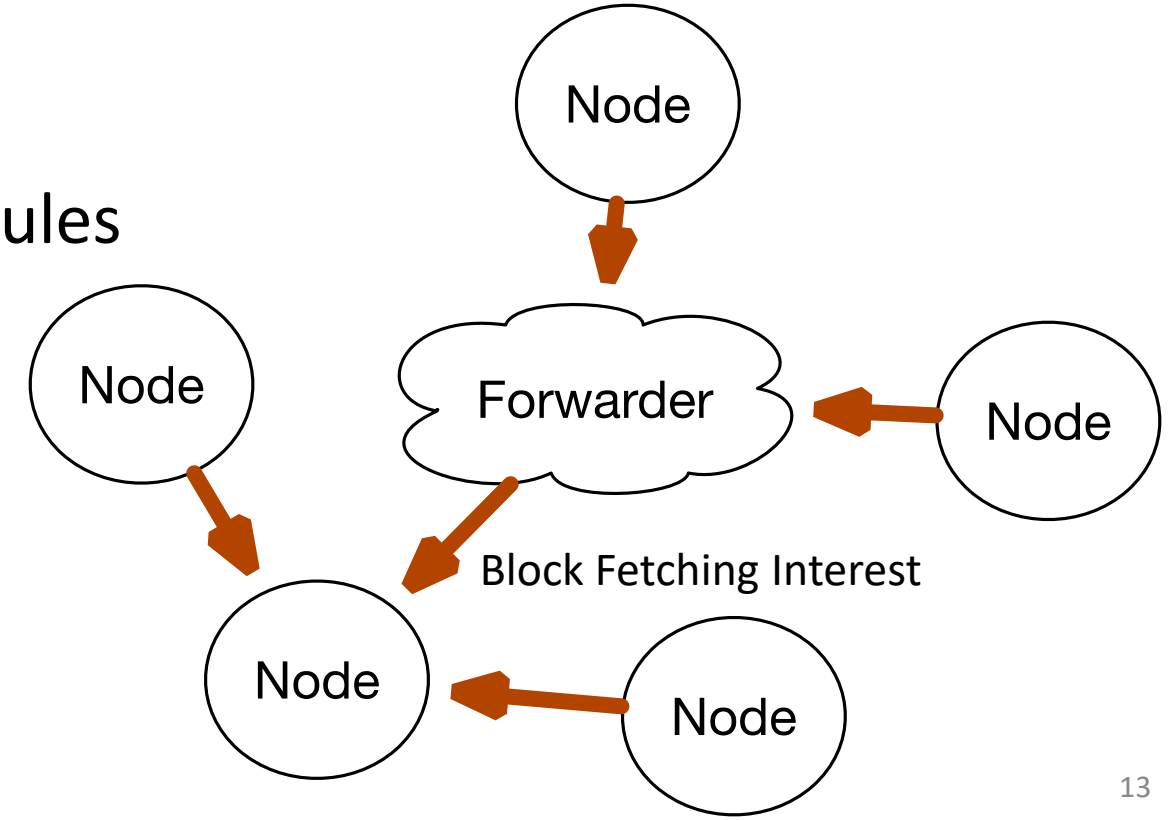
- Run MCMC and select two tips
- Verify two tips by checking the rules
- Generate block and sign it
- **Interest Broadcast**



When a node wants to append a new Block

Each node periodically append a record block into the Tangle and at the same time, approve two existing tips. A block holds the recent energy production/consumption records

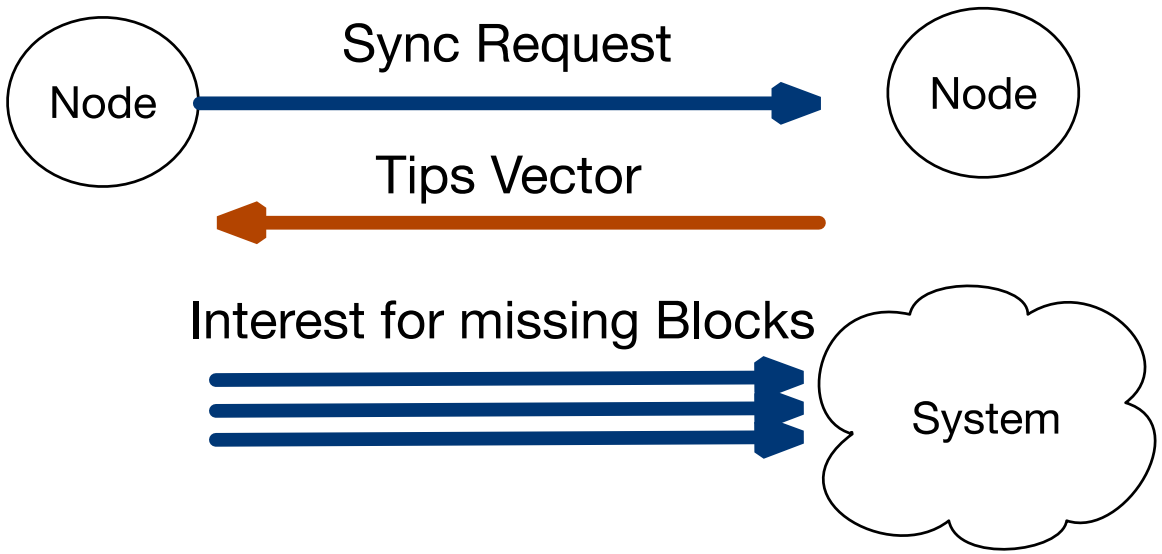
- Run MCMC and select two tips
- Verify two tips by checking the rules
- Generate block and sign it
- Interest Broadcast
- **Block Fetching**



Challenge 1: After back online

A Sync Protocol based on Tip Comparison

For each different tip
Step 1: hash look up to find whether the tip is already in the tangle
Step 2: If yes: ignore the tip (A is outdated)
Step 3: If no (B is outdated): walk back from the tip and fetch all the missing by checking the pointer fields in each block



Challenge 2: Malicious Block Appending

How to prevent a node from adding large amount of blocks to make a invalid block get accepted?

- The frequency of appending blocks is monitored by the whole system
- New blocks should not be accepted by peers if the frequency is larger than a threshold

Current Status

- Initial design without enough implementation
- Any comments are welcome

Thank You!

zhiyi@cs.ucla.edu