CREDENCE Workshop

Performance Modeling and Analysis of the Bitcoin Inventory Protocol

Yahya Shahsavari, Kaiwen Zhang, and Chamseddine Talhi

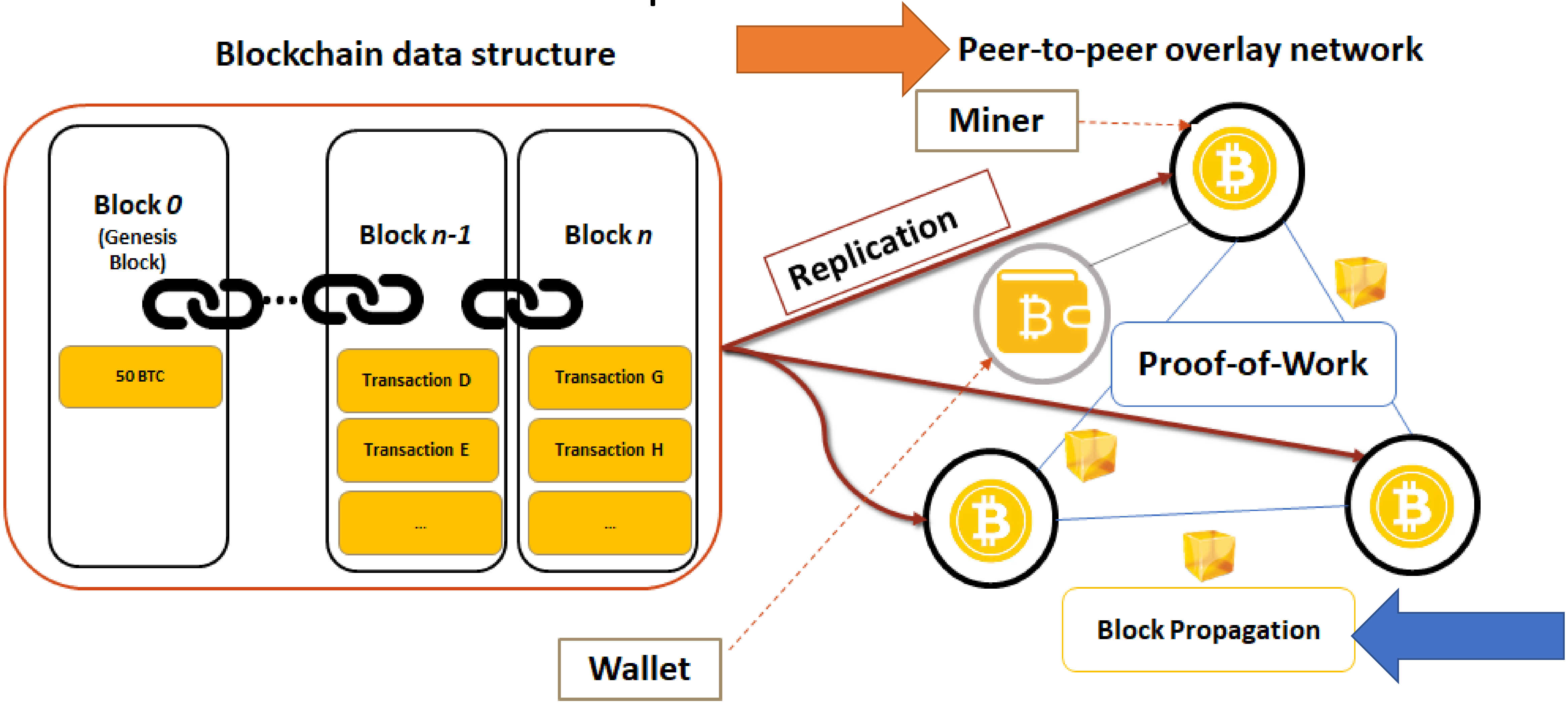
École de technologie supérieure (ÉTS)
University of Québec
Montréal, Canada

August 2019
Stavanger

Why use performance modeling?

- Performance modeling: Calculate/predict performance metrics
- DApps: Modeling helps choosing and tuning the right DLT system
- Bitcoin and altcoins: Impact of varying **blockchain parameters and network conditions** to assess the health of a particular Bitcoin-based system

Overview of the scope of the work

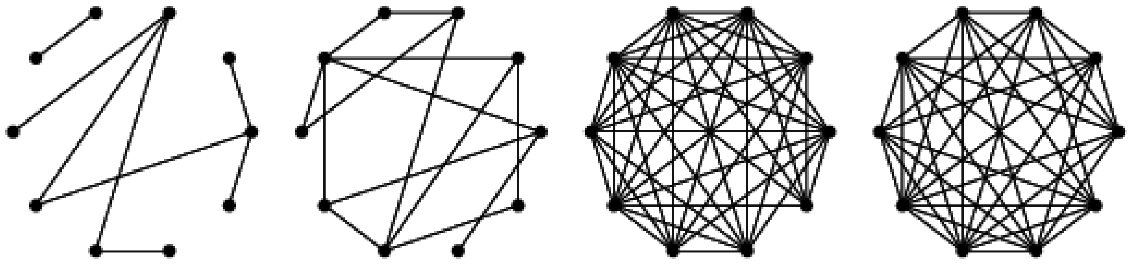


Contributions of the work

1. Model the Bitcoin overlay network using random **graphs.**
2. Model the block propagation algorithm of Bitcoin using **waves.**
3. Present mathematical equations for important performance metrics: **block propagation delay** and **traffic overhead,** as well as **fork occurrence probability .**
4. Implement the model using a **network simulator (OMNet++)** and **validate** the results with Bitcoin historical data.
5. Demonstrate the impact of the **block size** and **average number of connections per node** and **P2P bandwidth** on the block propagation delay and fork occurrence probability.
6. We estimate the **weight of each branch** in case of fork occurrence.

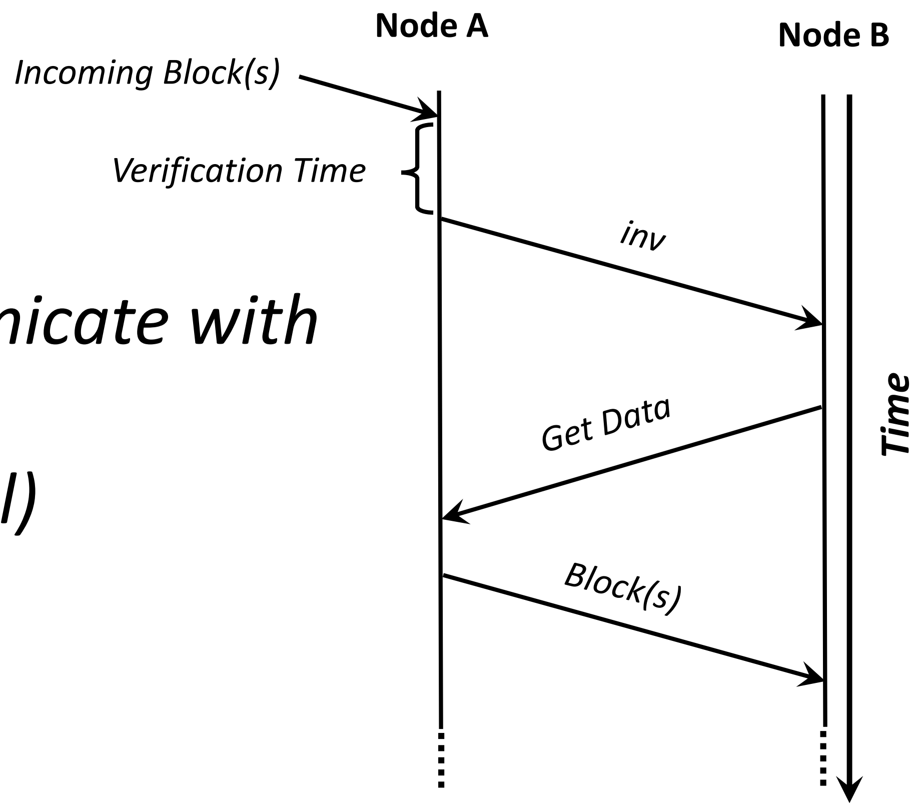
Modeling the Bitcoin overlay network

- Overlay network as a graph: $G(V, L)$.
- If there is a link between node i and node j, then $(i, j) \in L$.
- Random graph $G_p(N)$: N nodes and link probability p.
- A random graph models an ***ideal decentralized network***
  - Relay nodes or mining pools are considered in extension of our work.
- Can use *p* to derive *M* (average number of connections per node)



(increasing p in a random graph)

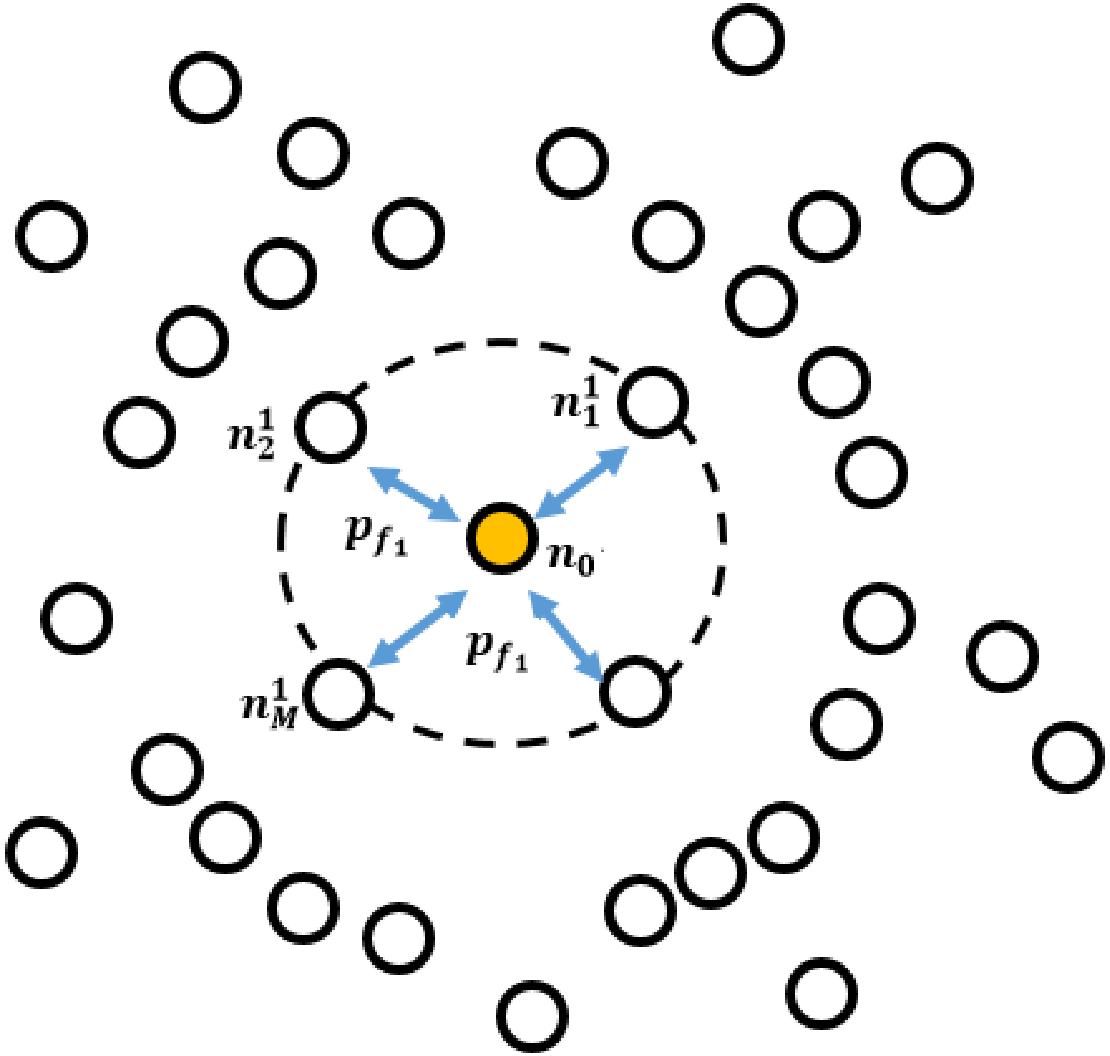
Inventory-based protocol of Bitcoin



Each node communicate with all its neighbors (Gossiping protocol)

Block dissemination (Wave 1)

- n_0 : node who mined a block
- Receiving nodes in wave 1:
  $$W_1 = \{n_1^1, n_2^1, \dots, n_M^1\}$$
- Each node has a **forwarding probability** p_f to reply the inv message with $getdata$ message
- Forwarding probability for the first wave: $p_{f_1} = 1$



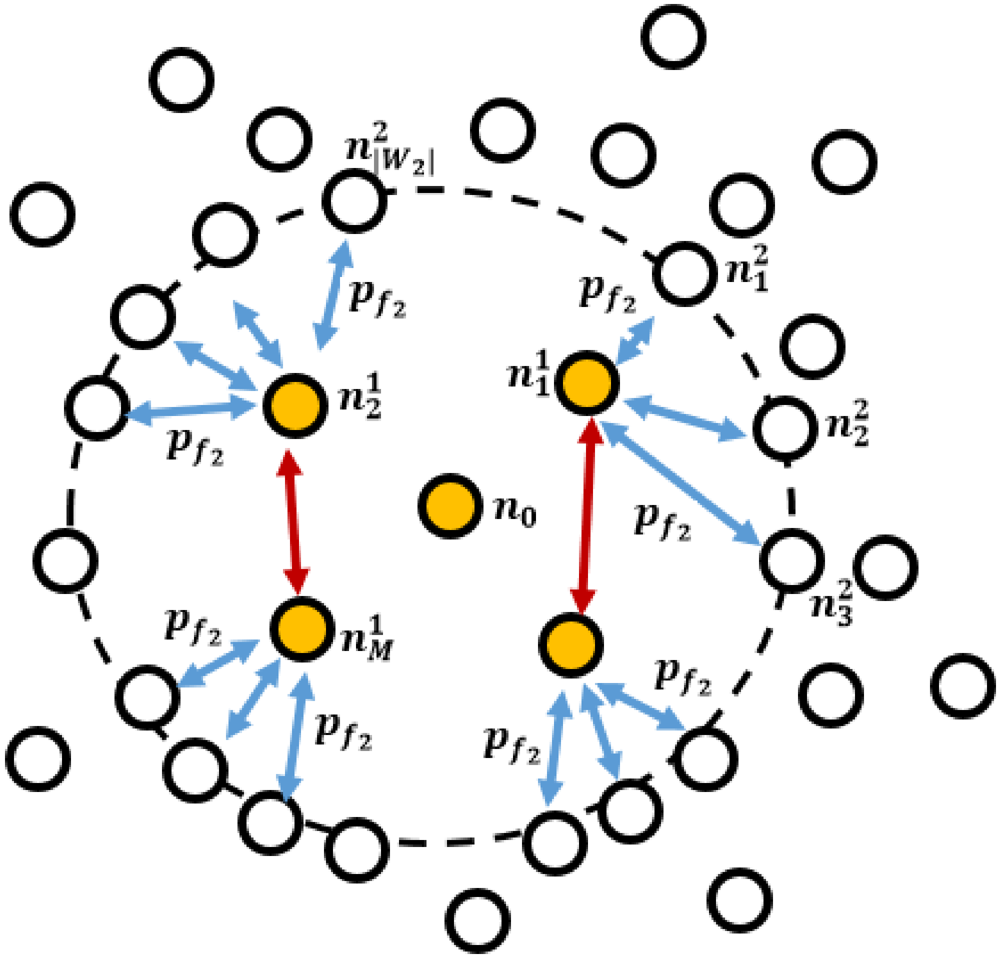
Wave 2 analysis

- In wave 2, $p_{f_2} \neq 1$
- Some nodes contacted in wave 2 received it in wave 1 already:

$$P_{f_2} = \frac{N - 1 - |W_1|}{N - 1}$$

- Number of block copies obtained during this wave:

$$|W_2| = \lceil p_{f_1} p_{f_2} \boldsymbol{M}^2 \rceil$$



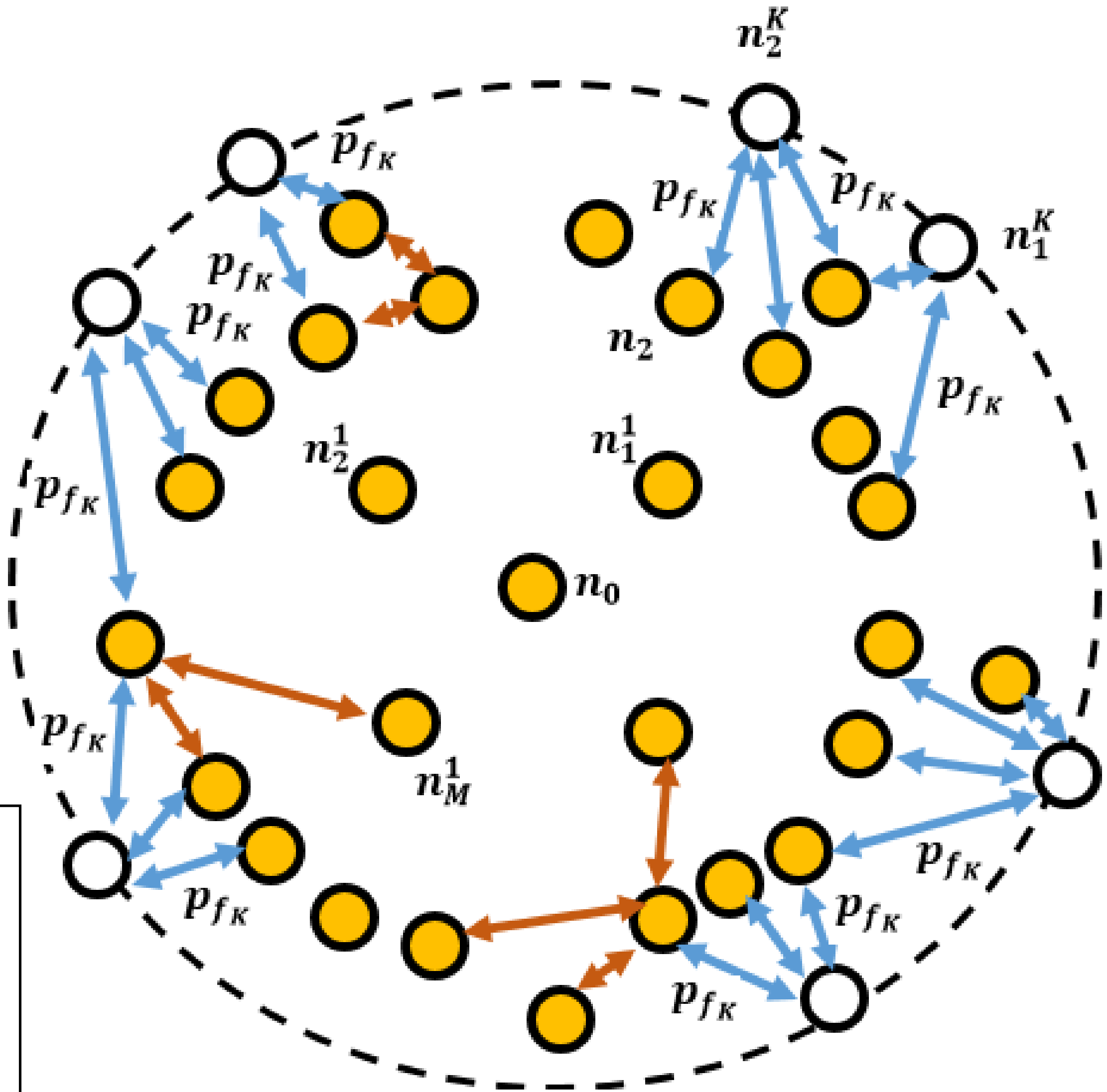
Analysis for wave *i*

$$|W_j| = \lceil \mathbf{M}^j \prod_{k=1}^{j-1} p_{f_k} \rceil$$

$$p_{f_i} = \frac{N - 1 - \sum_{j=0}^{i-1} \mathbf{M}^j \prod_{k=1}^{j-1} p_{f_k}}{N - 1}$$

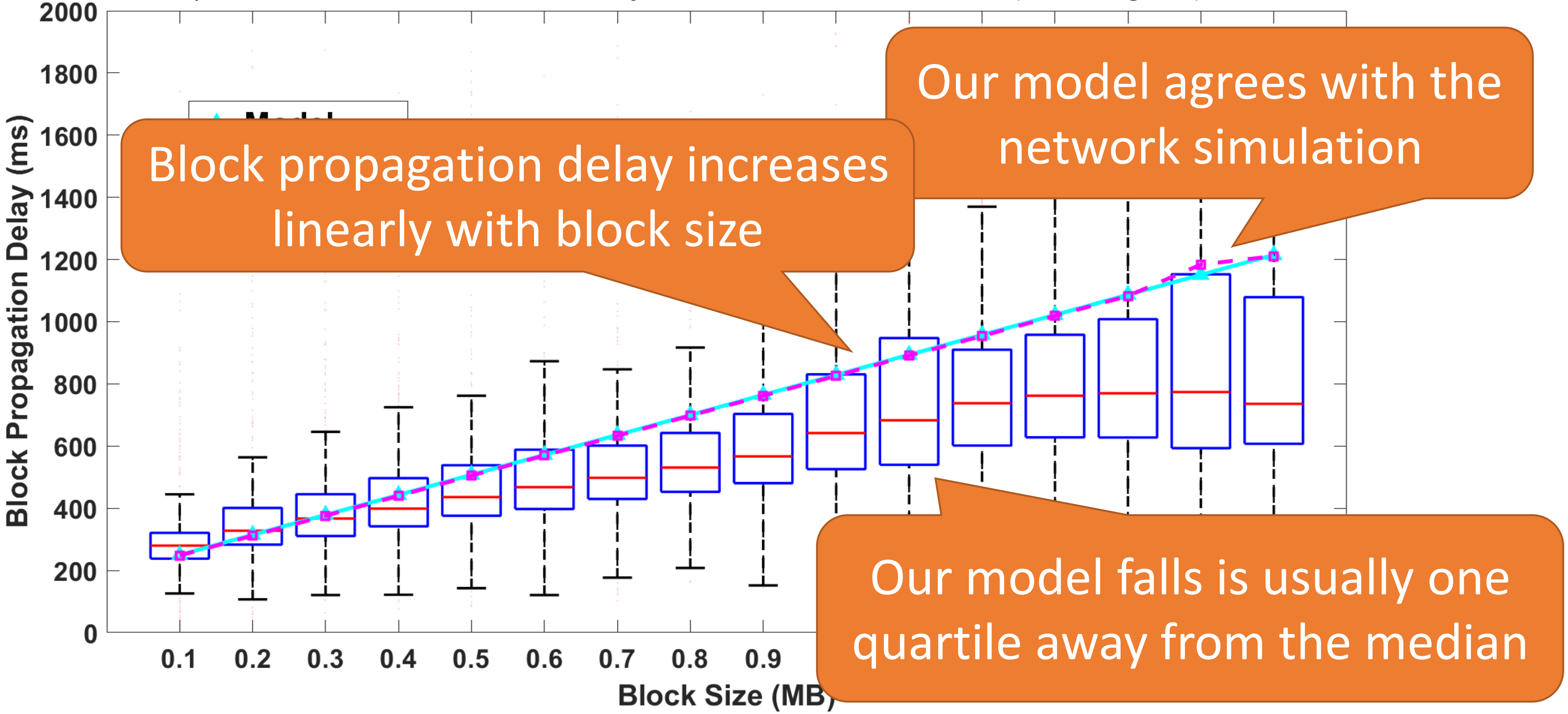
We use this model to create formulas for calculating:

1. Block propagation delay
2. Traffic overhead
3. Fork probability
4. Branch weights during a fork



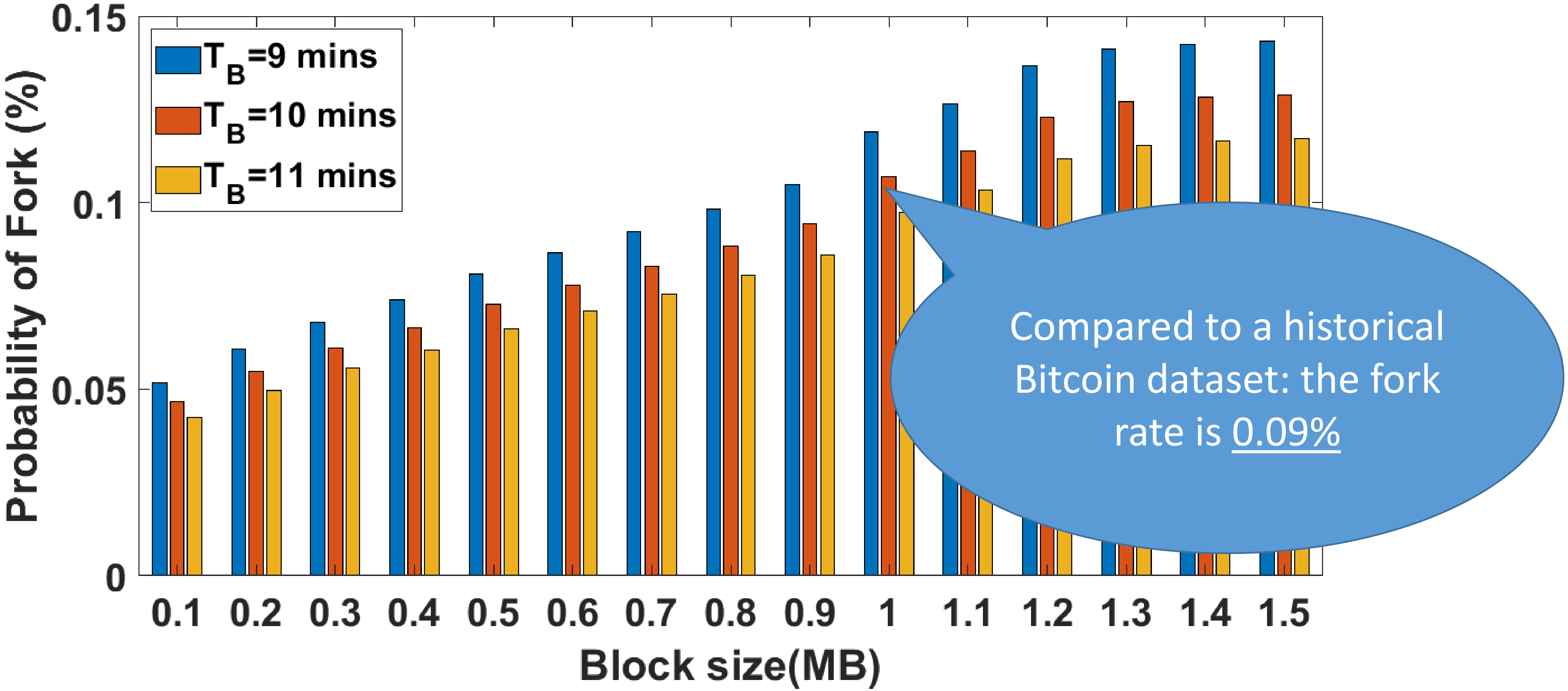
Comparison with Bitcoin data (M=32)

Box plots: Historical data extracted from 15,000 Bitcoin blocks (with SegWit)



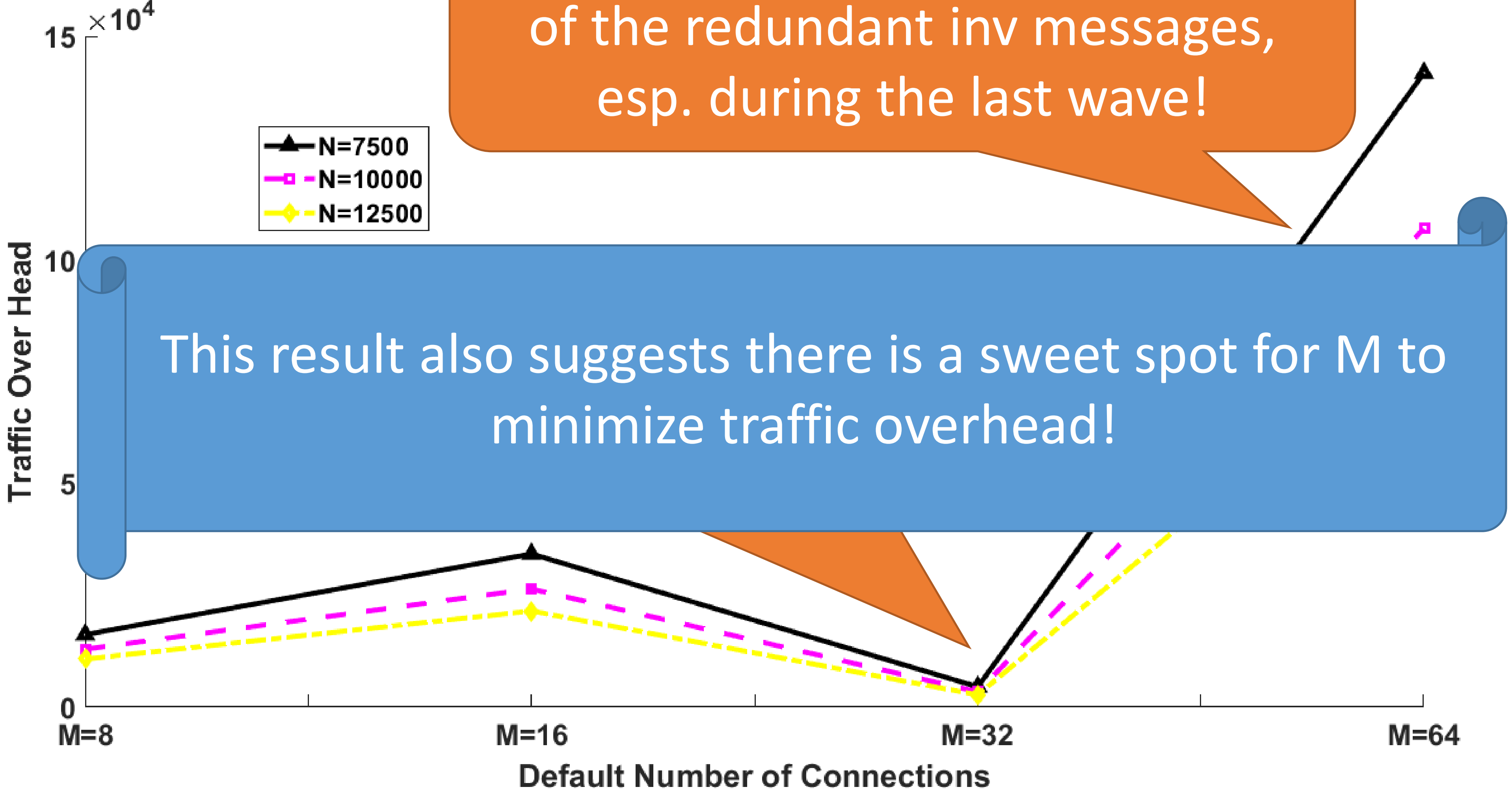
Shahsavari, Zhang, Talhi

Block size impact on the fork probability



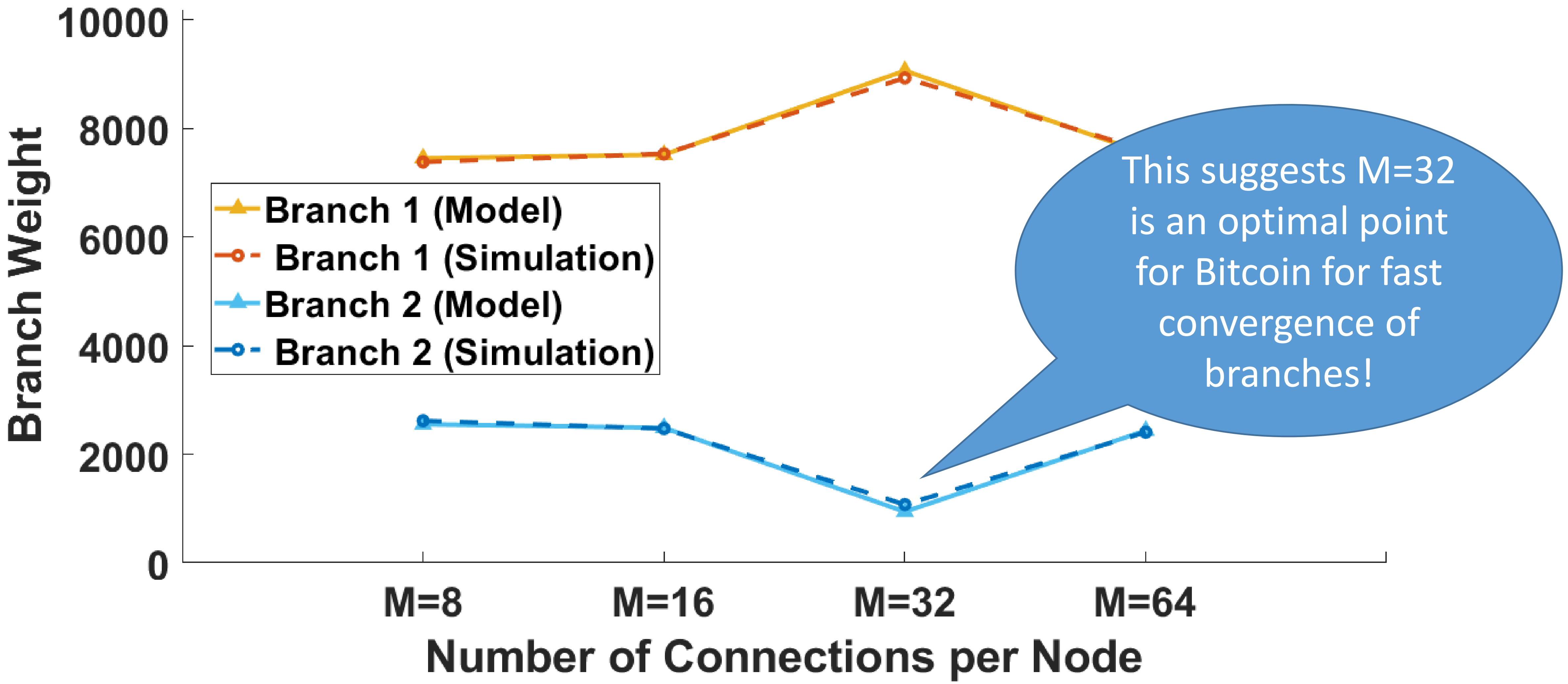
Traffic overhe

Overhead at M=64 is high because of the redundant inv messages, esp. during the last wave!



This result also suggests there is a sweet spot for M to minimize traffic overhead!

Branch weights ($t < t' < t + T$)



Publications

- **Performance Modeling and Analysis of the Bitcoin Inventory Protocol.**
  Yahya Shahsavari, Kaiwen Zhang, Chamseddine Talhi. *IEEE DAPPs 2019*. **Best Paper Award**.
- **A Theoretical Model for Fork Analysis in the Bitcoin Network.**
  Yahya Shahsavari, Kaiwen Zhang, Chamseddine Talhi. *IEEE BLOCKCHAIN 2019*.

Takeaway points

- Current **lower bound on number of connections** for safety: **4** for Bitcoin (10,000 nodes)
- Formulas for calculating:
  - Block propagation delay
  - Traffic overhead
  - Fork probability
  - Branch weights
- Important parameters include, but are not limited to:
  - Number of connections
  - Block size
  - Block time
  - Inter-block time (time between leading block and trailing block at the same height)
- Key observation: **32 connections is the sweet spot** for the current Bitcoin network
  - Reduces traffic overhead and branch weight of the trailing block
  - Validated by reports which determined that the current average is 32 in Bitcoin
- Currently working on considering relay networks and mining pools

Backup slides

How to choose a good value of *p* and *M*?

- Finding a good **lower bound** for *p*:
  - If *p* is too low, the network contains partitions: blocks cannot be fully propagated (perpetual branching!)
- If $p \geq \frac{\log(N)}{N}$, then $G_p(N)$ becomes a connected graph with **very high probability**.
- This is therefore a very **critical lower bound** for safety!
- Each node on average has M connections to other nodes
- To form a connected graph with high probability, it is sufficient that:

$$M \geq \lceil \frac{N-1}{N} log(N) \rceil$$

Current lower bound for Bitcoin network

- Current size of 10,000 node can be supported with M = 4
- Bitcoin protocol imposes a default limit of **8 outgoing connections**
- This limit is sufficient for a size of ~**100,000,000 nodes**
- However, the reported average number of connections in Bitcoin is **32**
- Next: what is a good value of *M* **beyond the lower bound?**
  - To answer this, we need to model block propagation

GLOBAL BITCOIN NODES DISTRIBUTION

Reachable nodes as of Sat Apr 06 2019 21:55:05 GMT-0700 (Pacific Daylight Time).

9659 NODES

24-hour charts »

Top 10 countries with their respective number of reachable nodes are as follow.

RANK	COUNTRY	NODES
1	United States	2459 (25.46%)
2	Germany	1881 (19.47%)
3	France	603 (6.24%)
4	Netherlands	500 (5.18%)
5	China	355 (3.68%)
6	Canada	350 (3.62%)
7	United Kingdom	309 (3.20%)
8	Singapore	298 (3.09%)
9	n/a	273 (2.83%)
10	Russian Federation	263 (2.72%)

More (97) »

https://bitnodes.earn.com/

Calculating the block propagation delay

- 100% block propagation: $\sum_{i=1}^{\boldsymbol{K}} \boldsymbol{M}^{i} \prod_{j=1}^{i-1} p_{f_j} = \boldsymbol{N}$
- $\boldsymbol{K}$ = **Total number of waves** needed for 100% propagation
- D = **Block propagation delay**

$$D = \boldsymbol{K}\left(D_v + \frac{S_i}{B} + Y_I + D_g + \frac{S_g}{B} + Y_G + D_b + \frac{\boldsymbol{S_b}}{B} + Y_B\right)$$

- B = Bandwidth of each link
- D_v: Block validation time, D_g: inv message processing time, D_b: $getdata$ message processing time
- Y_I, Y_G, Y_B: Signal propagation delay for: inv message, $getdata$ message, and the propagated block, respectively
- $S_i, S_g, \boldsymbol{S_b}$: Size of inv message, $getdata$ message, **and the block**, respectively

Calculating the traffic overhead

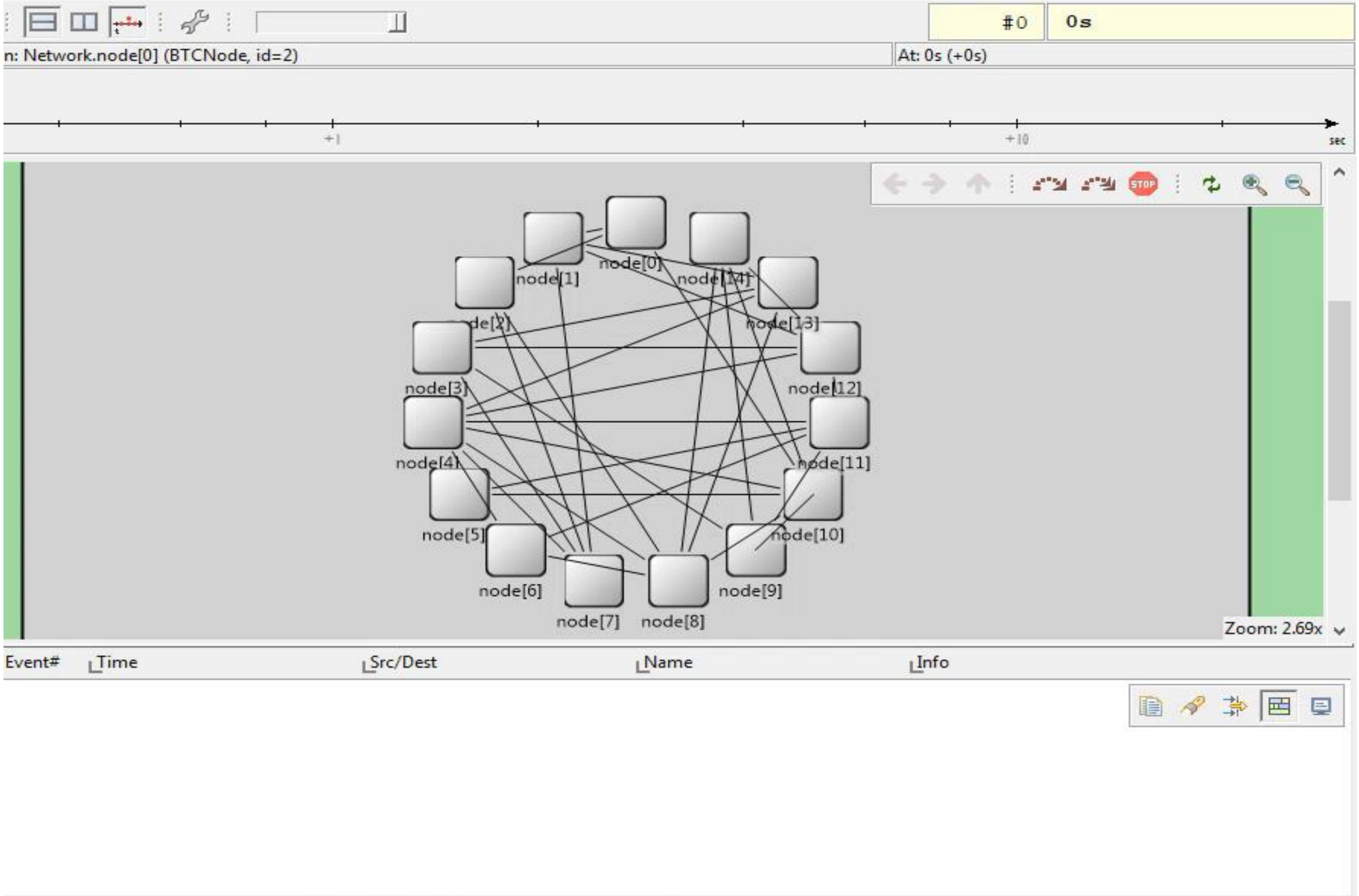
- Traffic overhead: % of timed-out inv messages.
- Wave i:

$$H_i = \frac{(1 - p_{f_i})\boldsymbol{M}^i \prod_{j=1}^{i-1} p_{f_j}}{N - 1}$$

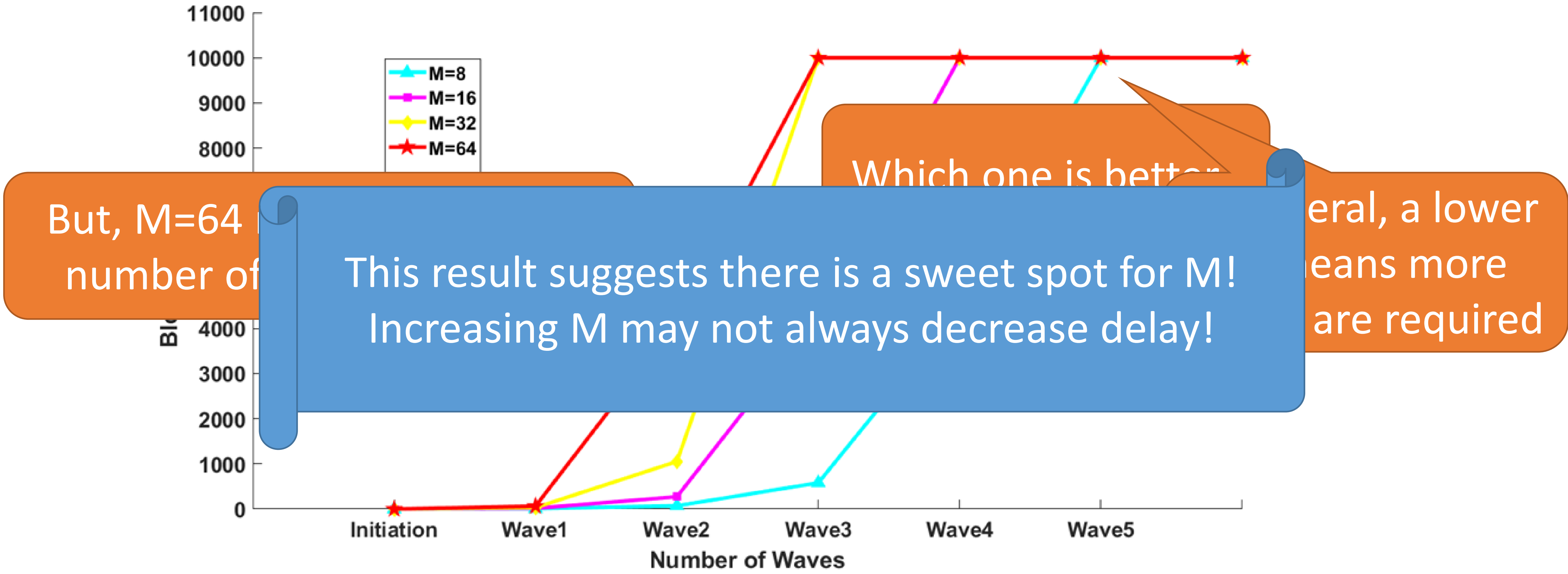
- Overall overhead:

$$\bar{H} = \frac{1}{N - 1} \sum_{i=1}^{\boldsymbol{K}} (1 - p_{f_i})\boldsymbol{M}^i \prod_{j=1}^{i-1} p_{f_j}$$

Simulating block propagation using OMNET++

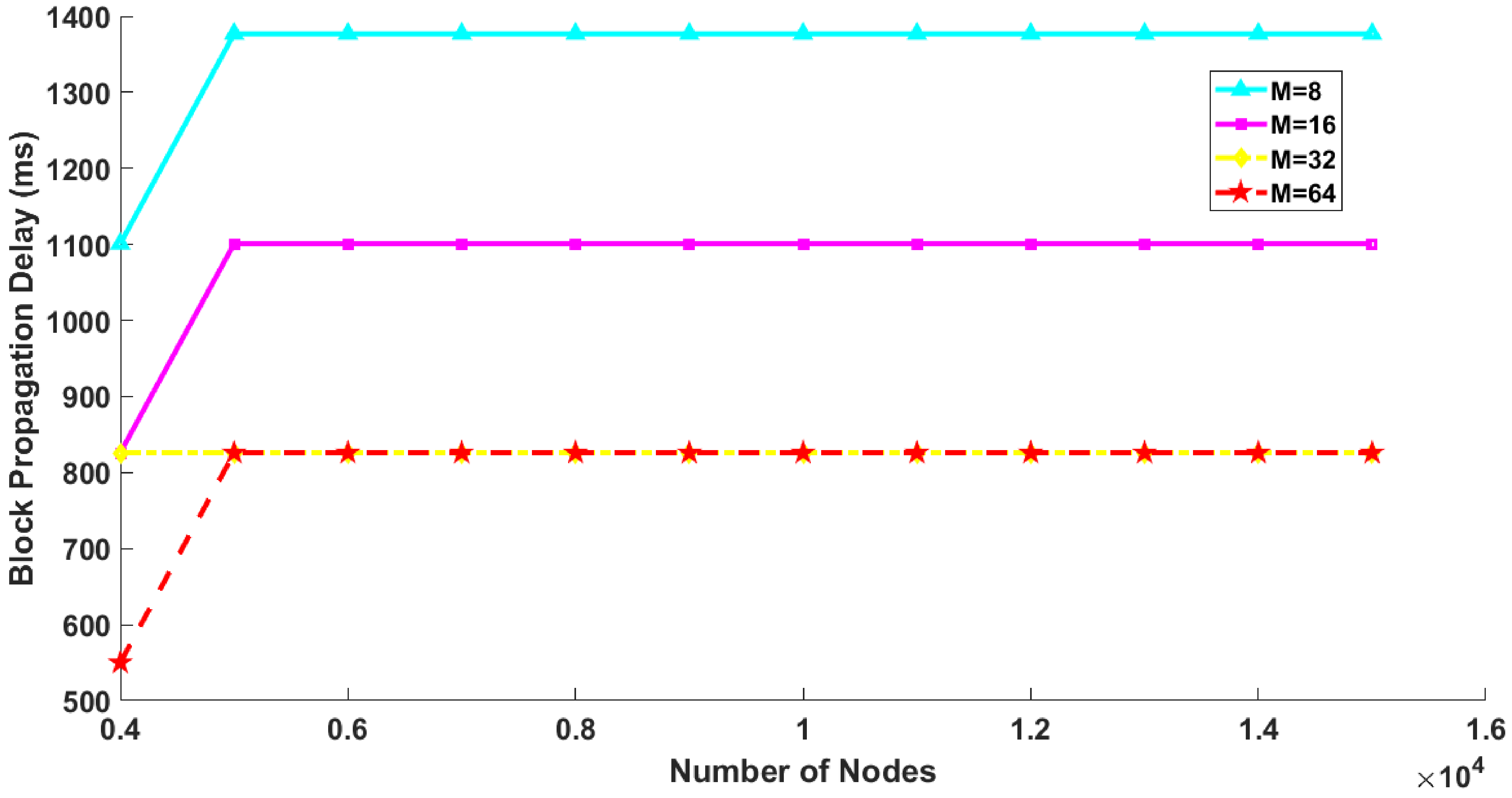


Utility of each wave with varying M

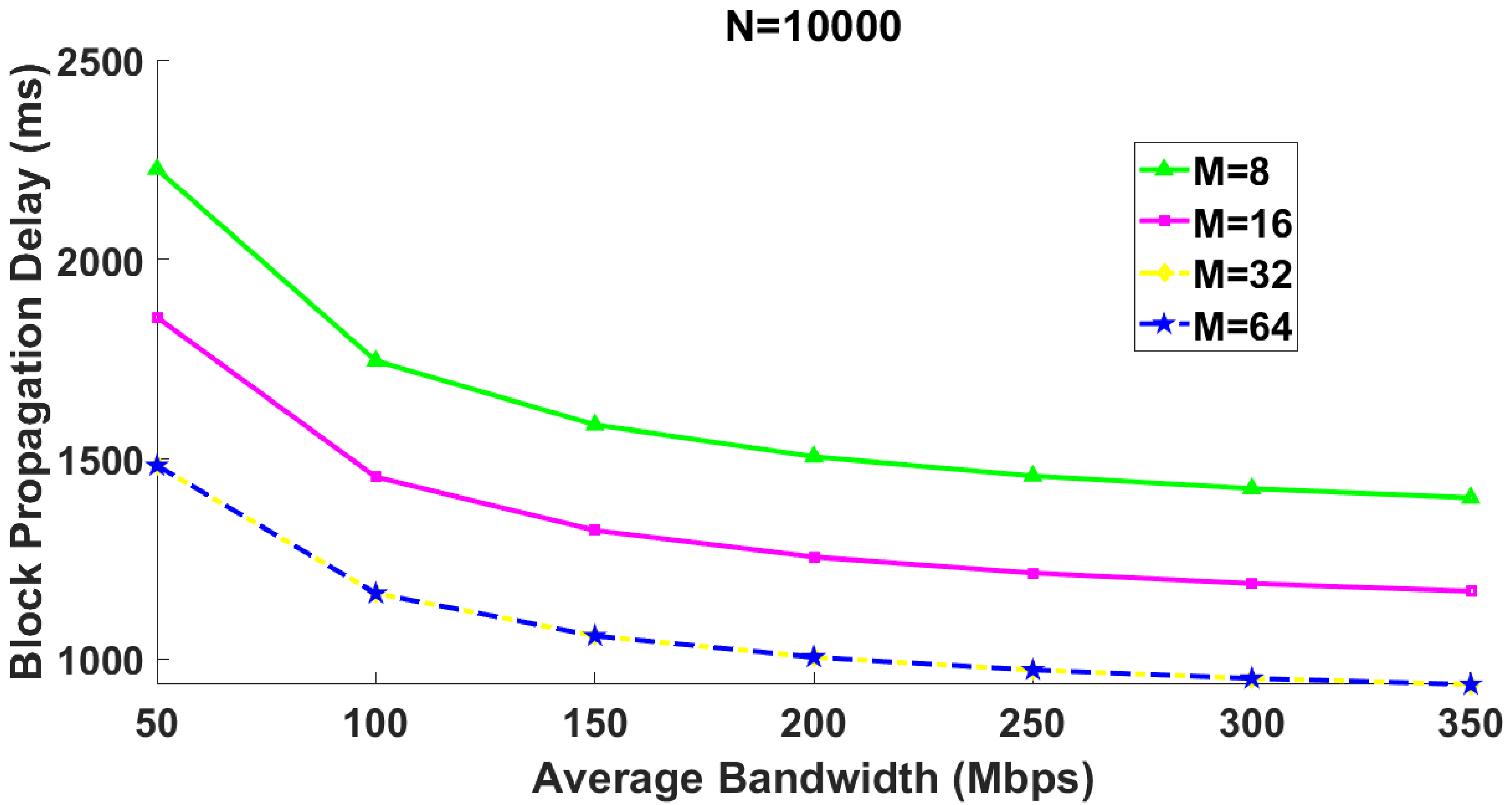


- M=8: minimum for Bitcoin protocol
- M=32: observed Bitcoin network

Block propagation analysis



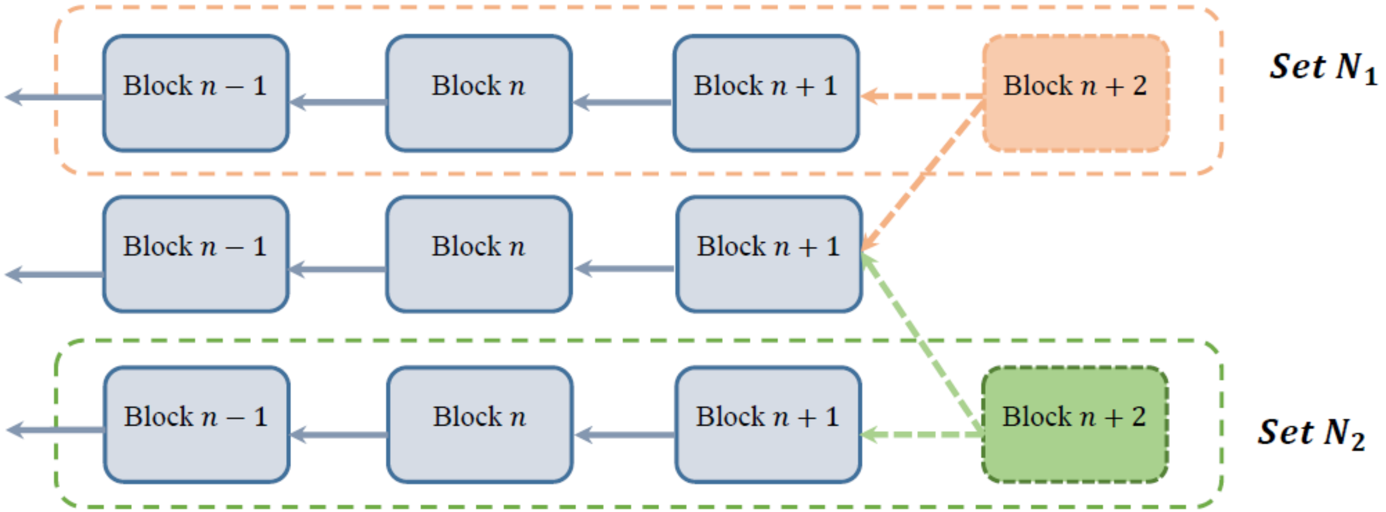
Block propagation analysis



Conclusion

- Although the throughput of the Bitcoin can be increased by choosing a bigger size for blocks, this can cause a significant increase in block propagations delay.
- The delay can be reduced by increasing number of connections per node, but this has the drawback of increased traffic overhead.

What is a Fork?!



$$\exists\, b, b' \in \mathcal{B} \text{ and } b \neq b' \,|\, h_b = h_{b'}$$

Forks can occur in one of these situations:

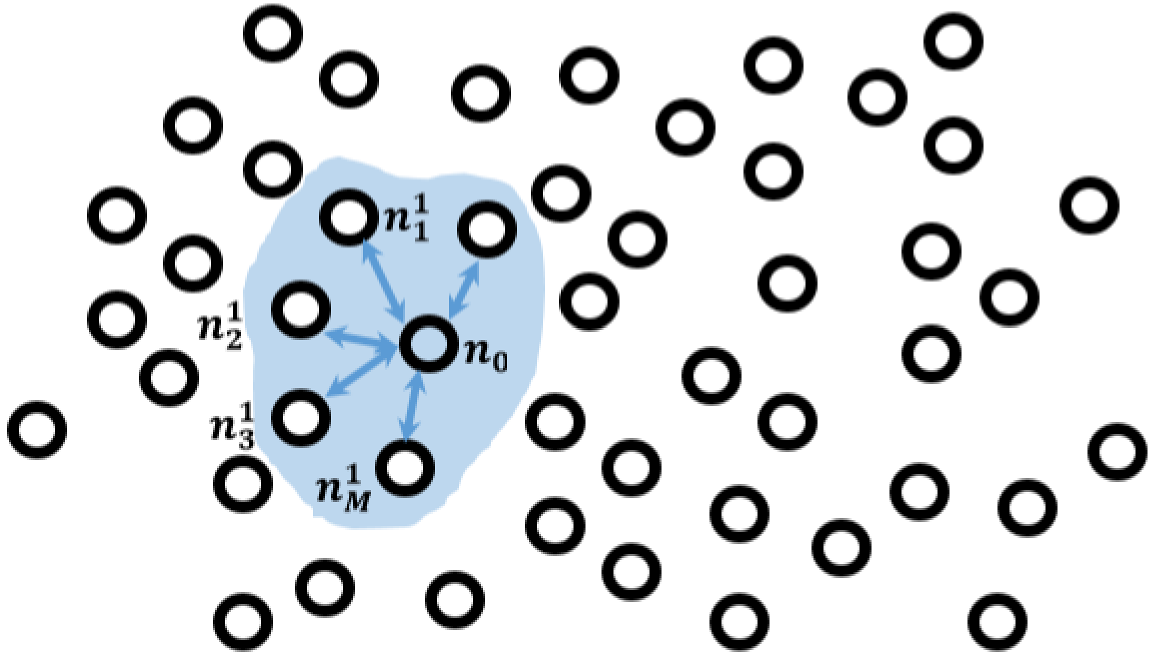
- **Network isolation**
    - Due to poor connectivity, network may become partitioned
- **Changes in core components of the blockchain protocol**
    - Soft forks
    - Hard forks
- **Miners deviation from the standard protocol**
    - Temporary block withholding
    - Selfish mining
    - Feather forking attacks
- **Block propagation delay**
    - Two different miners mine a block at almost the same time

Fork dissemination model

- n_0 : node who mined the block *b*
- Receiving nodes in wave 1:

$$W_1 = \{n_1^1, n_2^1, \dots, n_M^1\}$$

- Each node has a **forwarding probability** p_f to reply the inv message with $getdata$ message
- Forwarding probability for the first wave: $p_{f_1} = 1$
- At this point, the competing block *b*' **has not been mined yet**

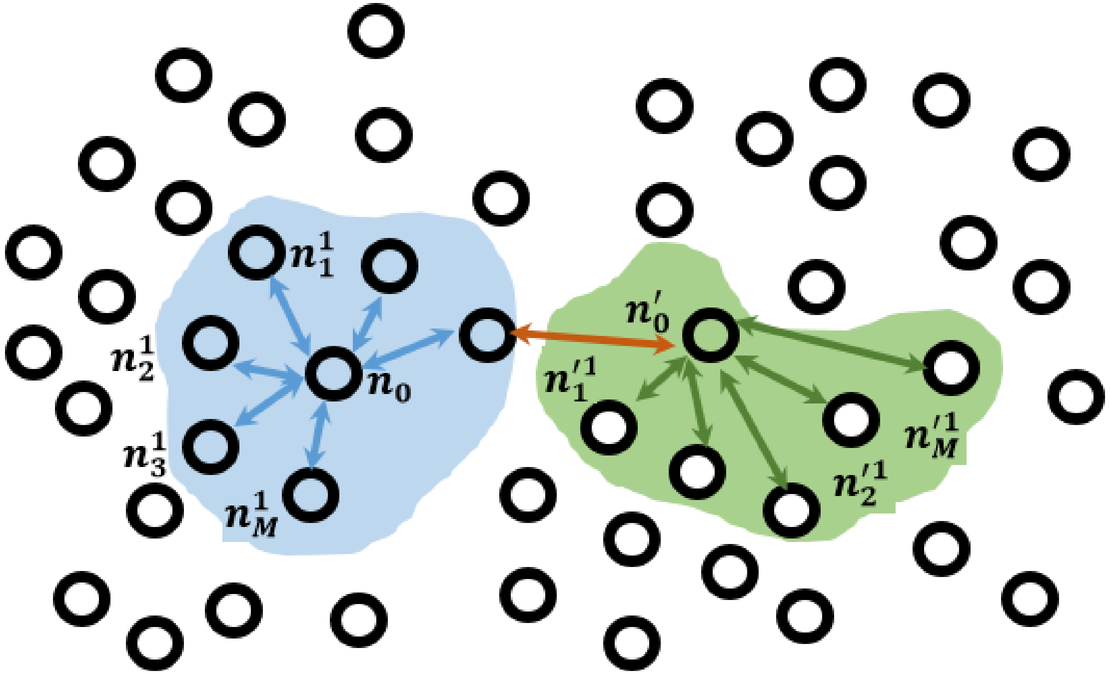


Wave 1 for block b’

- Suppose b' is mined at time t’: $t < t' < t + T$

T: wave length (1 wave time length)

- Receiving nodes in wave 1: $W_1' = \{n'^1_1, n'^1_2, \ldots, n'^1_M\}$
- Forwarding probability for the first wave for the block b': $p'_{f_1} \neq 1$.
- $p_{f'_1} = \frac{N-1-|W_1|}{N-1}$
- $|W_1| = \lceil p_{f_1} \boldsymbol{M} \rceil$



General formulas for wave i and time t'

- Recursive function calculated with results from **previous waves of both blocks**
- Mining time t' is generalized as follows:

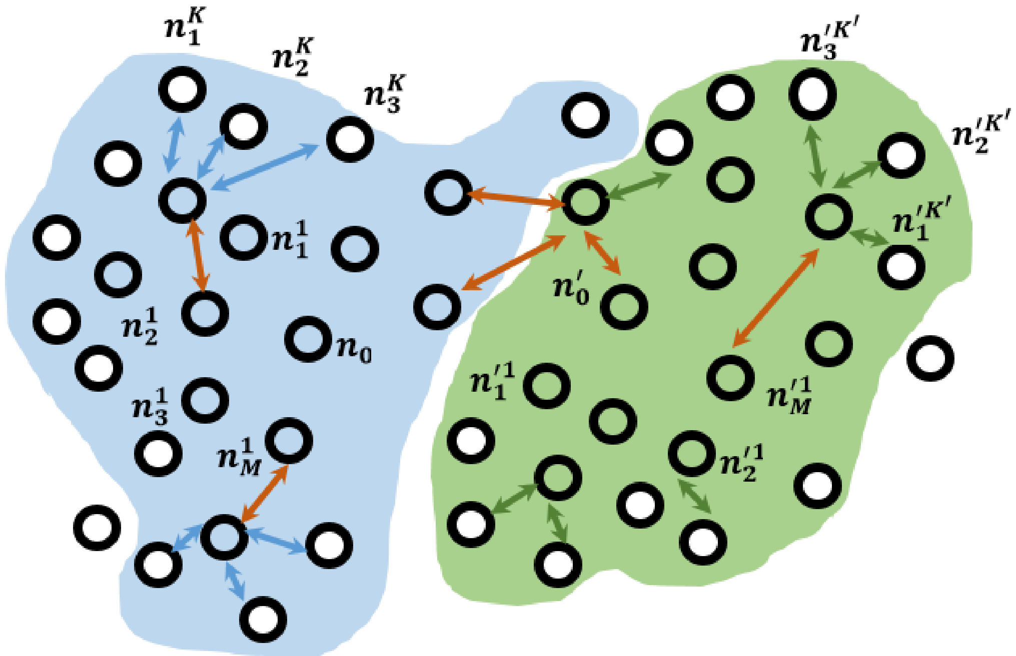
$$t + (m-1)T < t' < t + mT$$

- $p_{f_i} = \dfrac{N-1-\sum_{j=0}^{i-1} \boldsymbol{M}^j \prod_{k=1}^{j} p_{f_k} - \sum_{j=0}^{i-1} \boldsymbol{M}^j \prod_{k=1}^{j} p_{f'_k}}{N-1}$

$(1 < i \le K)$

- $p_{f'_i} = \dfrac{N-1-\sum_{j=0}^{i} \boldsymbol{M}^j \prod_{k=1}^{j} p_{f_k} - \sum_{j=0}^{i-1} \boldsymbol{M}^j \prod_{k=1}^{j} p_{f'_k}}{N-1}$

$(1 < i - m \le K)$

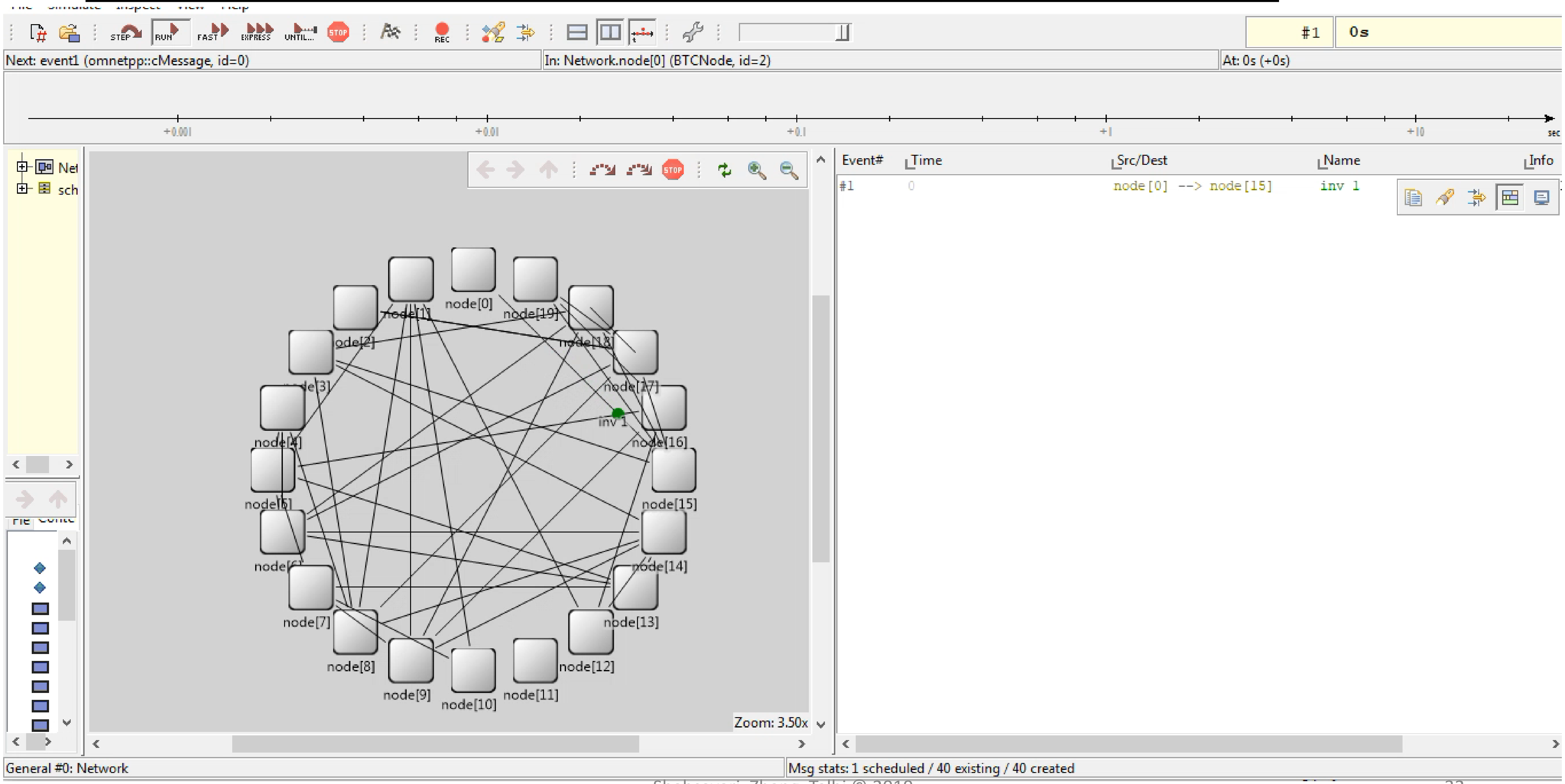


Fork dissemination model for wave 2

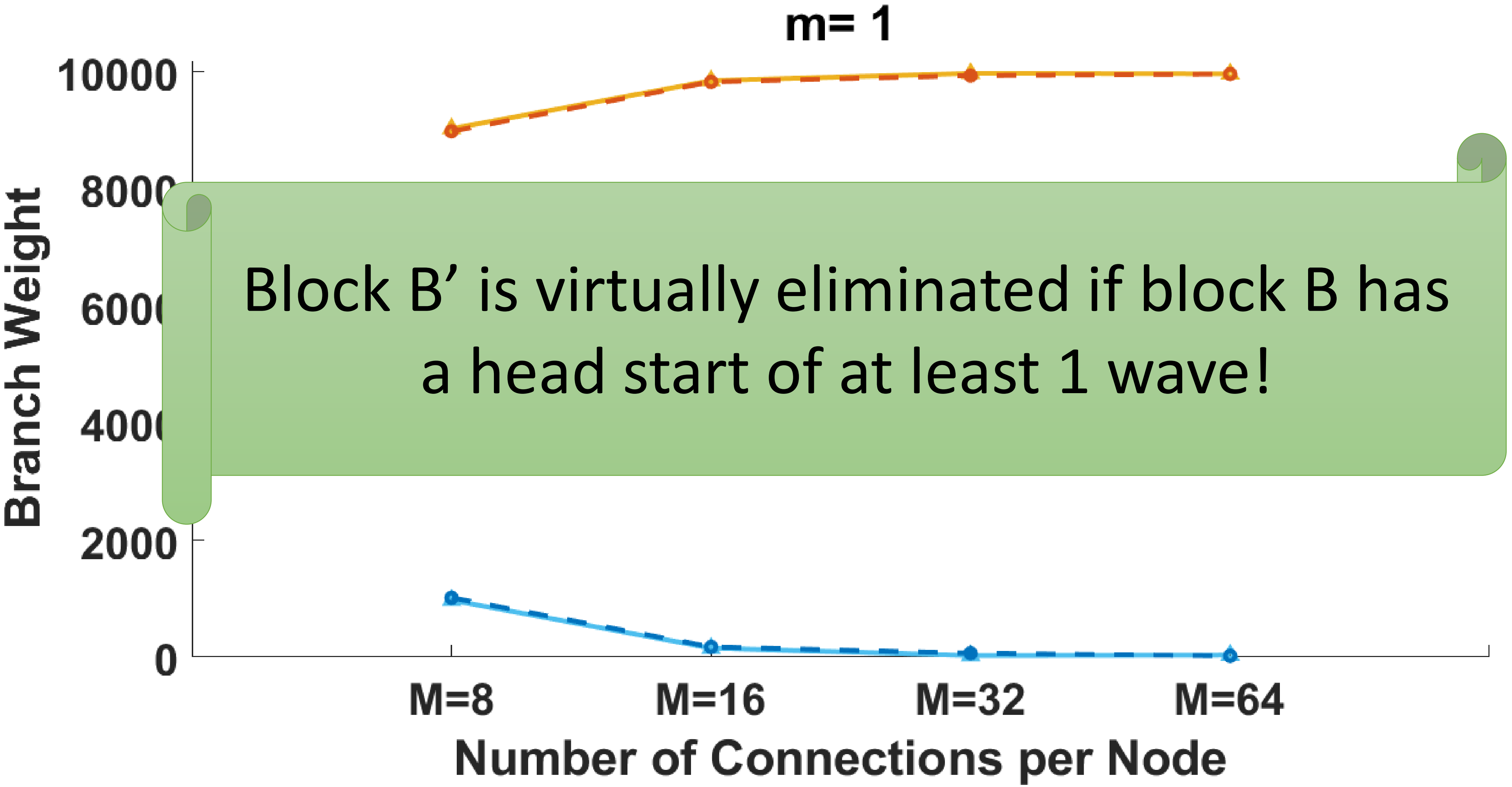
- Forwarding probability for the second wave: $p_{f_2} \neq 1$.
- $P_{f_2} = \frac{N-1-|W_1|-|W_1'|}{N-1}$
- $|W_1| = \lceil p_{f_1} M \rceil$
- $|W_1'| = \lceil p_{f_1'} M \rceil$

- Forwarding probability for the second wave: $p_{f_2} \neq 1$.
- $P_{f_2'} = \frac{N-1-|W_1|-|W_1'|-|W_2|}{N-1}$
- $|W_2| = \lceil p_{f_1} p_{f_2} M^2 \rceil$

Demo of our simulation using OMNET++

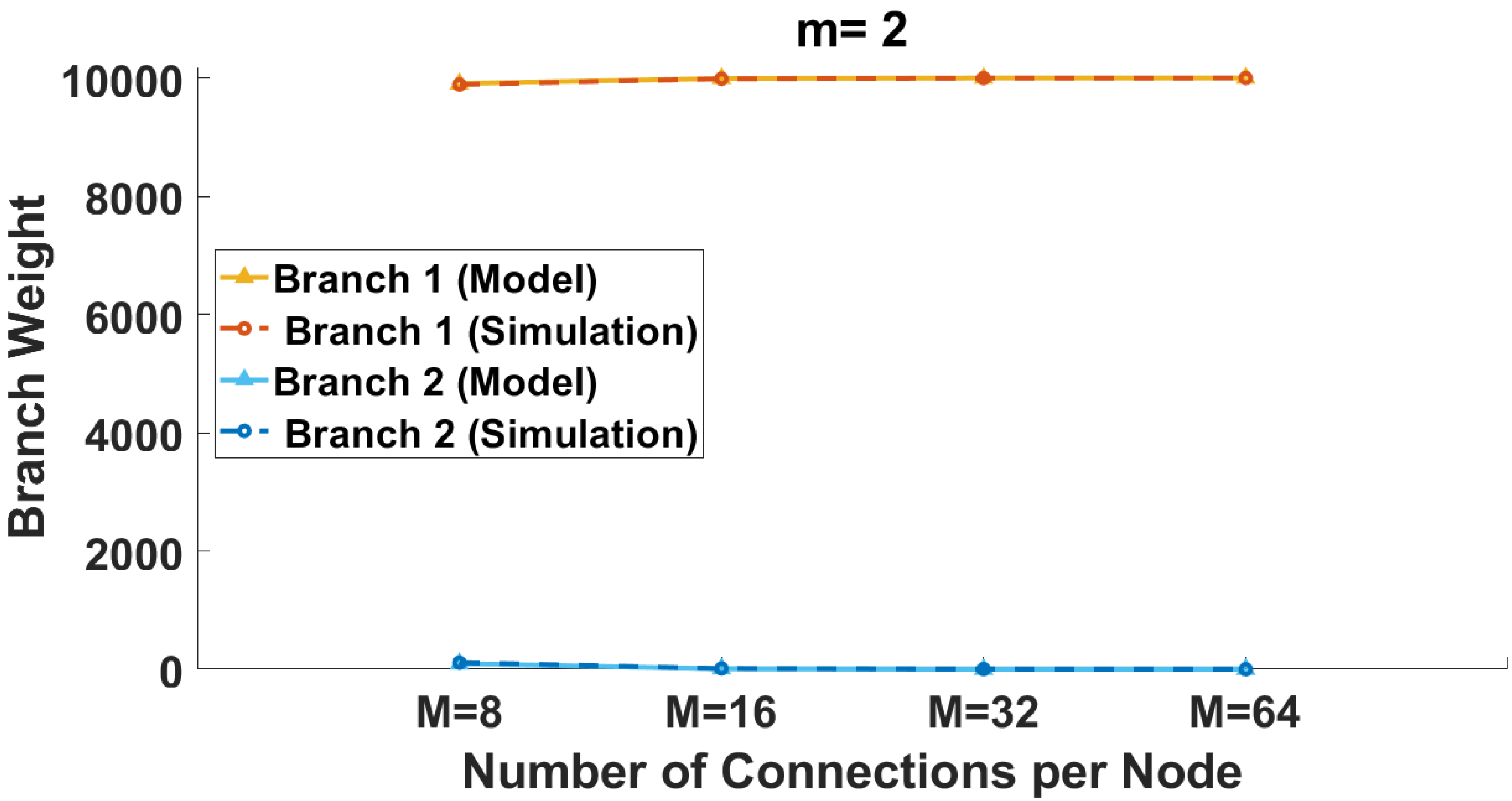


Branch weights ($t + T < t' < t + 2T$)



Block B’ is virtually eliminated if block B has a head start of at least 1 wave!

Branch weights ($t + 2T < t' < t + 3T$)



P2P bandwidth impact on the fork probability