

# TxProbe: Discovering Bitcoin's Network Topology Using Orphan Transactions

---

**Sergi Delgado-Segura**, Surya Bakshi, Cristina Pérez-Solà, James Litton, Andrew Pachulski, Andrew Miller and Bobby Bhattacharjee

# MOTIVATION



**What is the big deal with knowing the network topology?**

- Is the network really decentralized?
- Are there supernodes in the network?
- Are there weak spots that can be easily isolated?

**Currently, we do not know**

# THE TOPOLOGY SHOULD LOOK RANDOM



How Bitcoin (Core client) nodes choose their peers?

- Pseudorandomly from the **addrman**
- **8 outbound** connections by default
  - No pair of nodes in the same **/16** (IPv4)
- **117 inbound** connection by default (no IP restriction here)

Bitcoin forks based on the Core client follow the same approach

# REACHABLE AND NON-REACHABLE NODES

## GLOBAL BITCOIN NODES DISTRIBUTION

Reachable nodes as of Thu Feb 07 2019  
10:26:44 GMT+0000 (Greenwich Mean Time).

10365 NODES

**Difference between reachable and non-reachable?**

**Reachable nodes accept incoming connections, non-reachable do not**

**How many non-reachable nodes are out there?**

**There is not good approximation (AFAIK)**

**Probing techniques generally focus on the reachable network**

10 Japan 228 (2.20%)

[More \(100\) »](#)

Map shows concentration of reachable Bitcoin nodes found in countries around the world.

LIVE MAP

Source: <https://bitnodes.earn.com/>

# TRANSACTION PROPAGATION IN BITCOIN



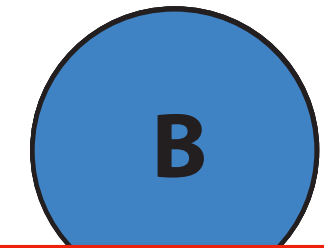
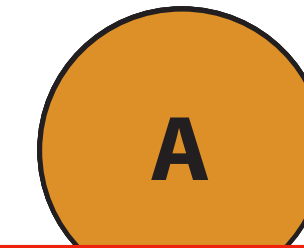
Our inferring technique is based on transaction propagation

We take advantage of how some kind of transactions (**orphans** and **double-spending**) are handled by nodes

How does it roughly work?

# TRANSACTION PROPAGATION IN BITCOIN

Transactions are shared between peers in a push manner



When  
he a

Valid transaction are stored in **mempool**

Transaction in **mempool** are eventually **propagated throughout the node neighborhood**

Upon  
does  
the

A node will wait **up to 2 minutes** for a response to a **getaddr message** before asking any other peer

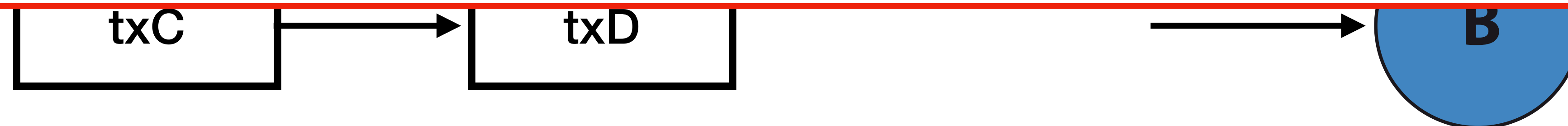
Wait for tx  
up to 2 min

Upon receiving a request or a transaction, a node  
will reply back with it (**deliver**)

# ORPHAN TRANSACTIONS

A transaction is flagged as orphan when a node receiving it does not know about

Transactions in MapOrphanTransactions **are NOT forwarded** to any node  
If the same transactions is offered again to the node (**inv message**), it will not ask back for it (**getaddr**)



**txD is orphan from  
B's point of view**

# DOUBLE-SPENDING TRANSACTIONS



Given a pair of double-spending transactions, a node will **pick the first one** it learns about

id = 4F3...ED

:

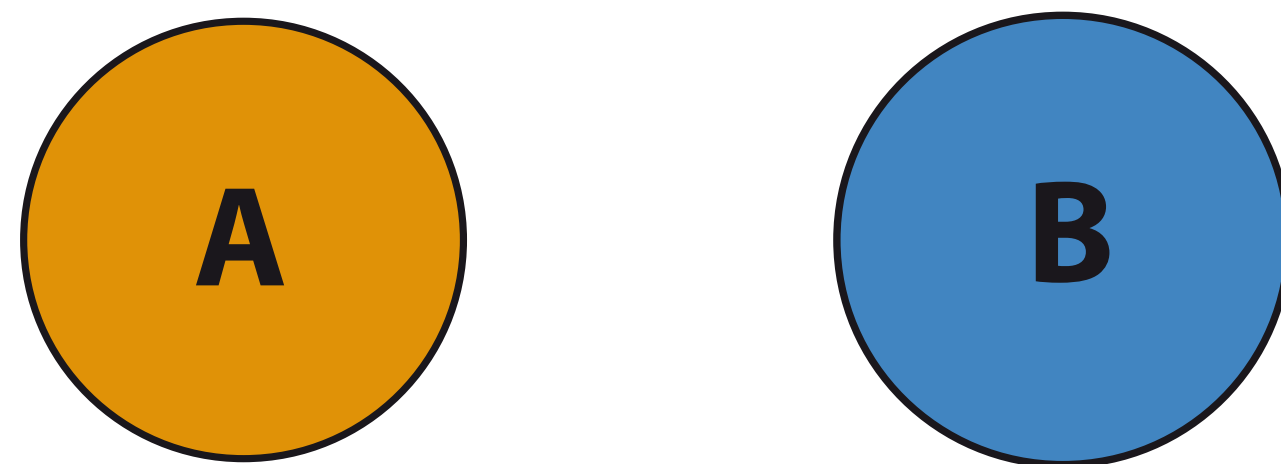
txn-1

txB



# A BASIC TOPOLOGY INFERRING TECHNIQUE

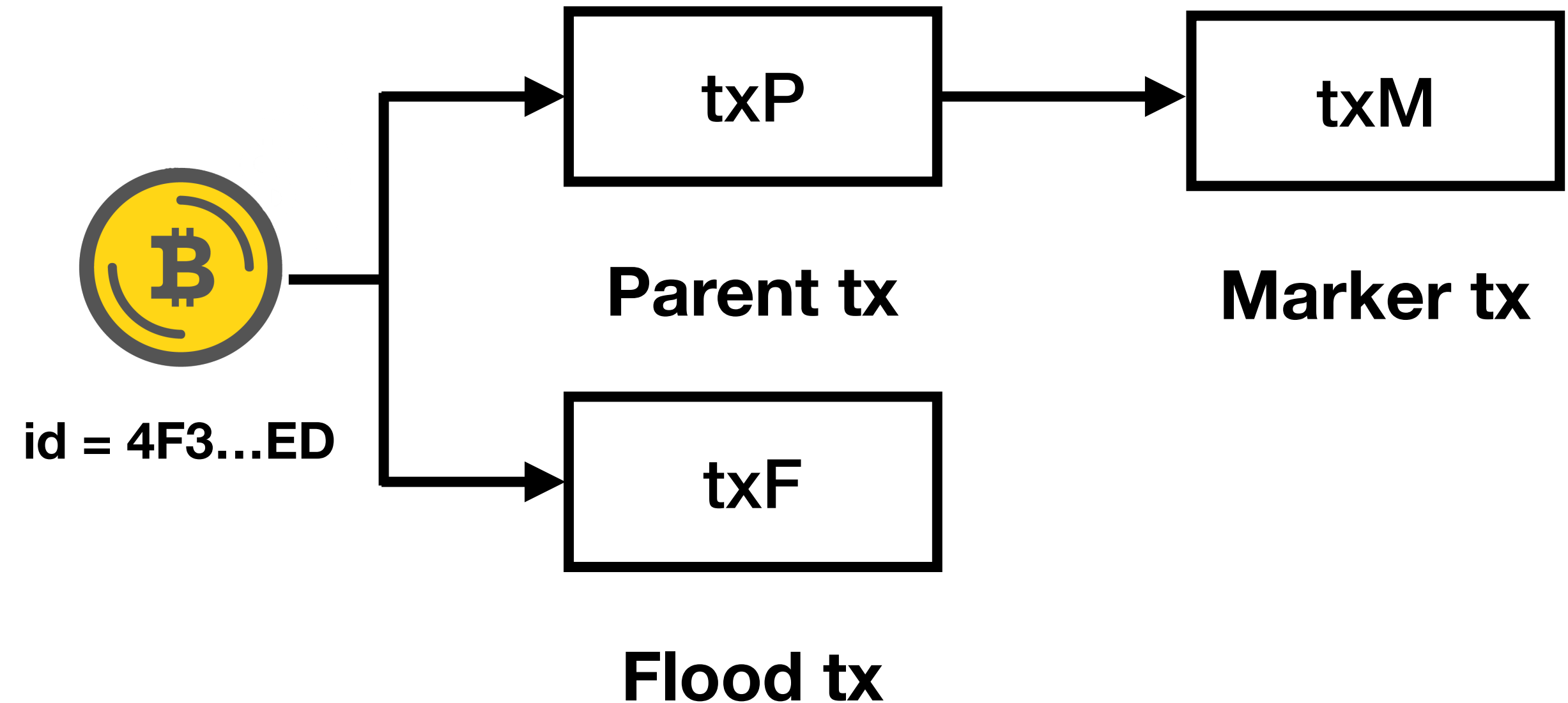
## Two nodes



## Observation tool

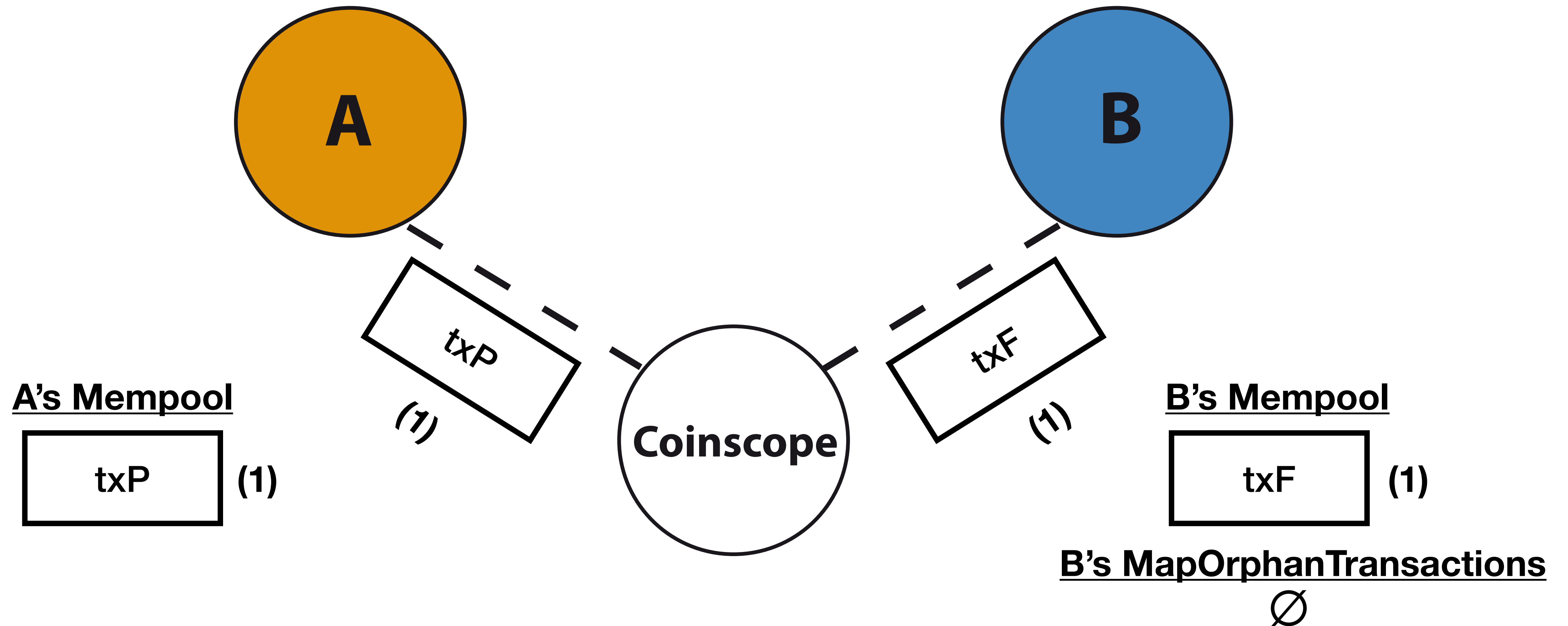


## Three transactions

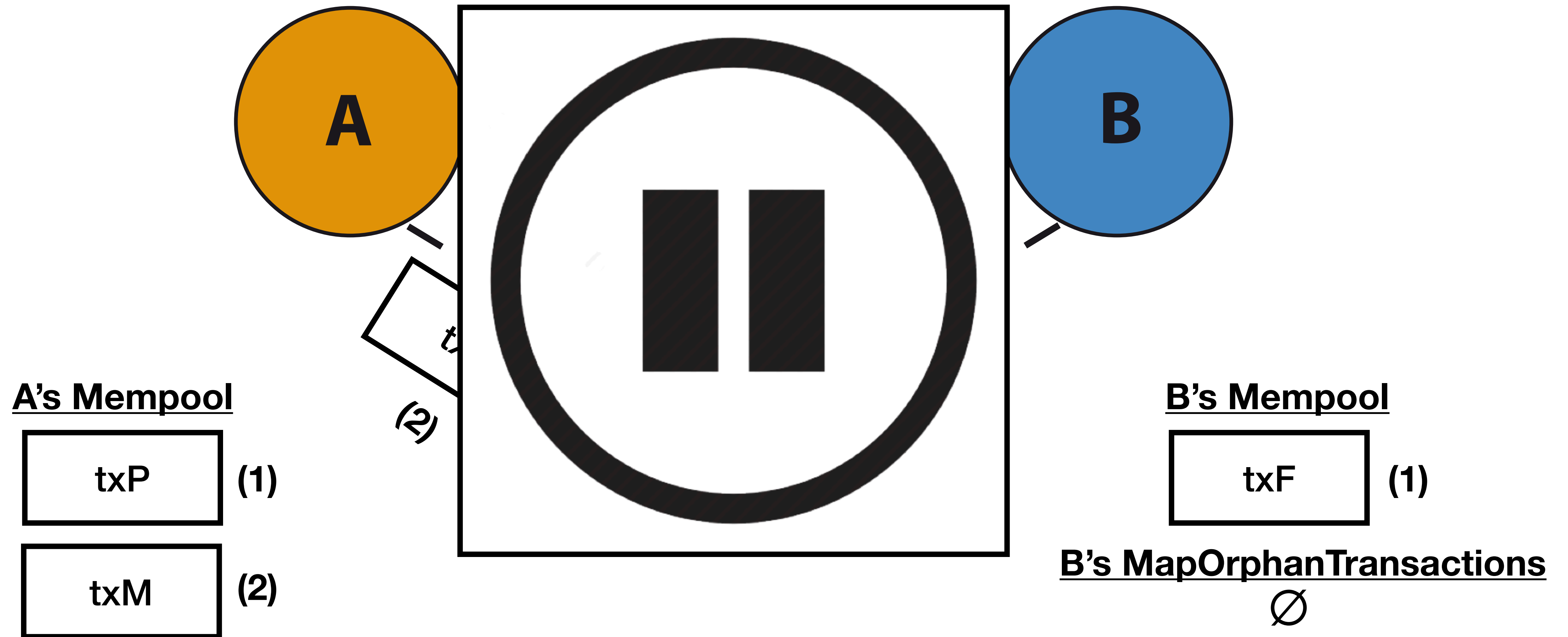


Andrew Miller, James Litton, Andrew Pachulski, Neal Gupta, Dave Levin, Neil Spring,  
Bobby Bhattacharjee  
**Discovering Bitcoin's Public Topology and Influential Nodes**

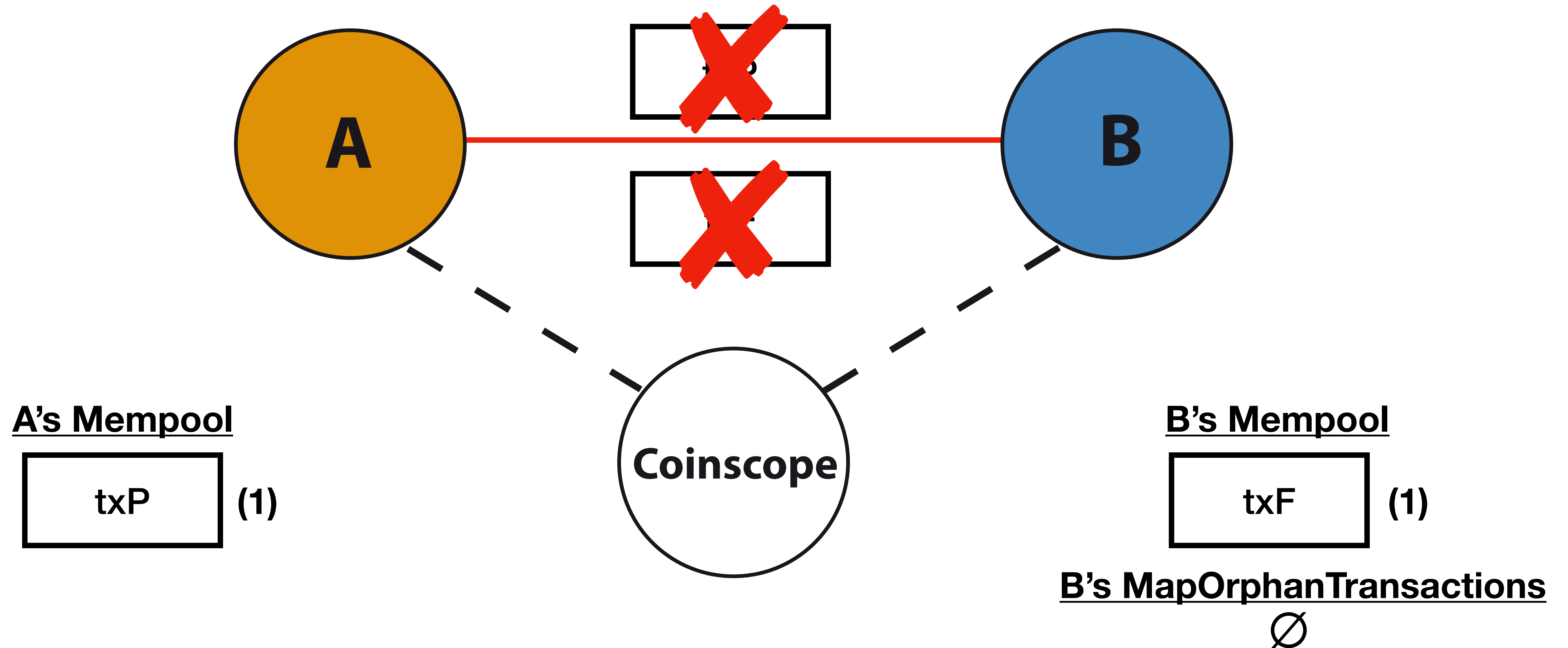
# NEGATIVE INFERRING TECHNIQUE



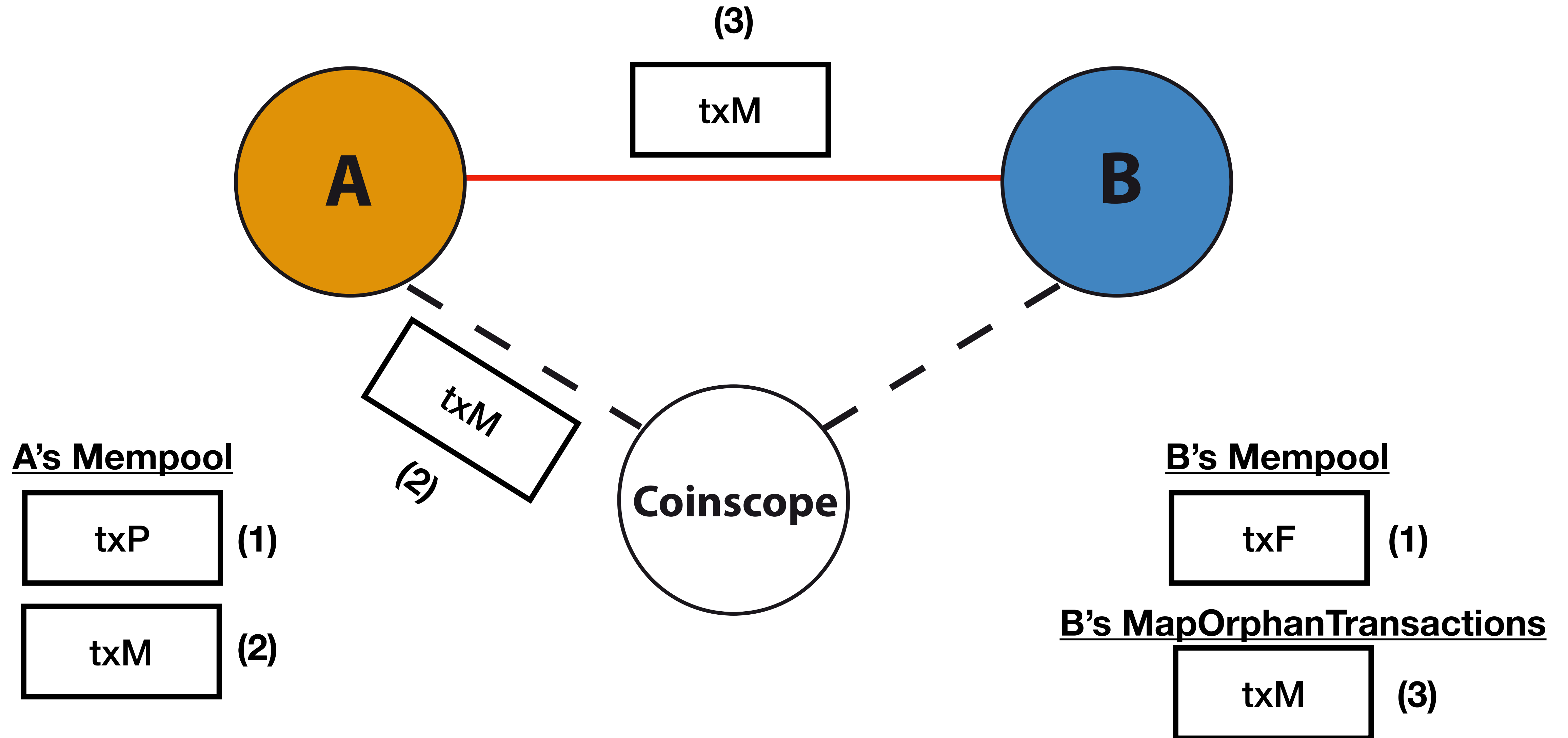
# NEGATIVE INFERRING TECHNIQUE



# POSITIVE INFERRING TECHNIQUE



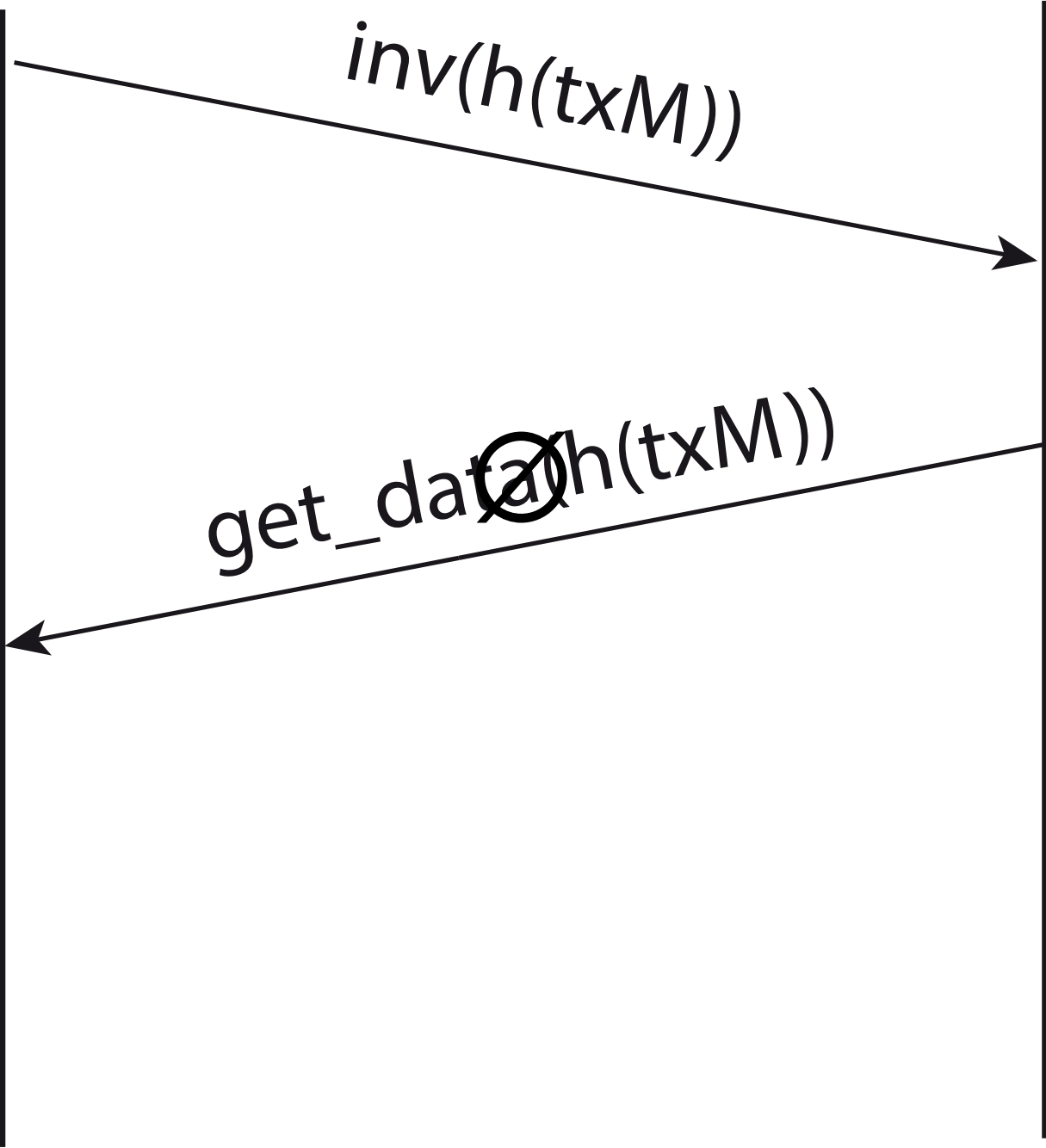
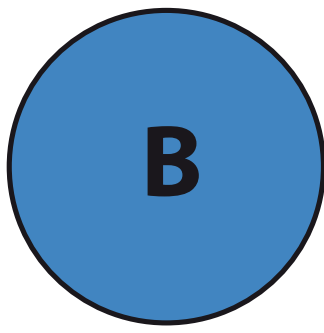
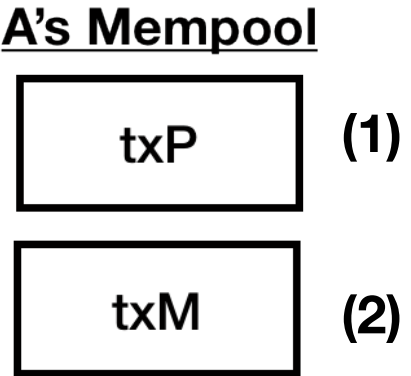
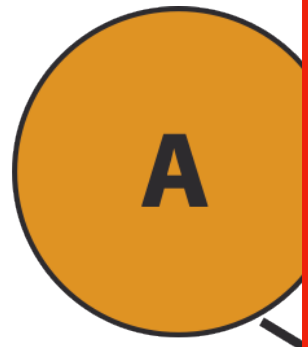
# POSITIVE INFERRING TECHNIQUE



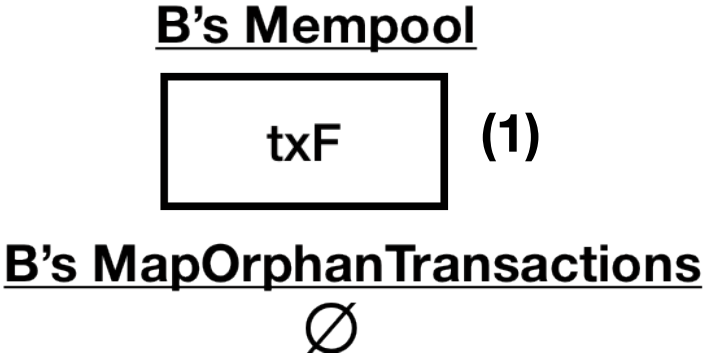
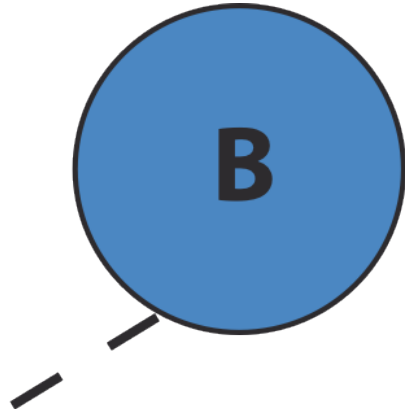
# A BASIC TOPOLOGY INFERRING TECHNIQUE



Position



inferring





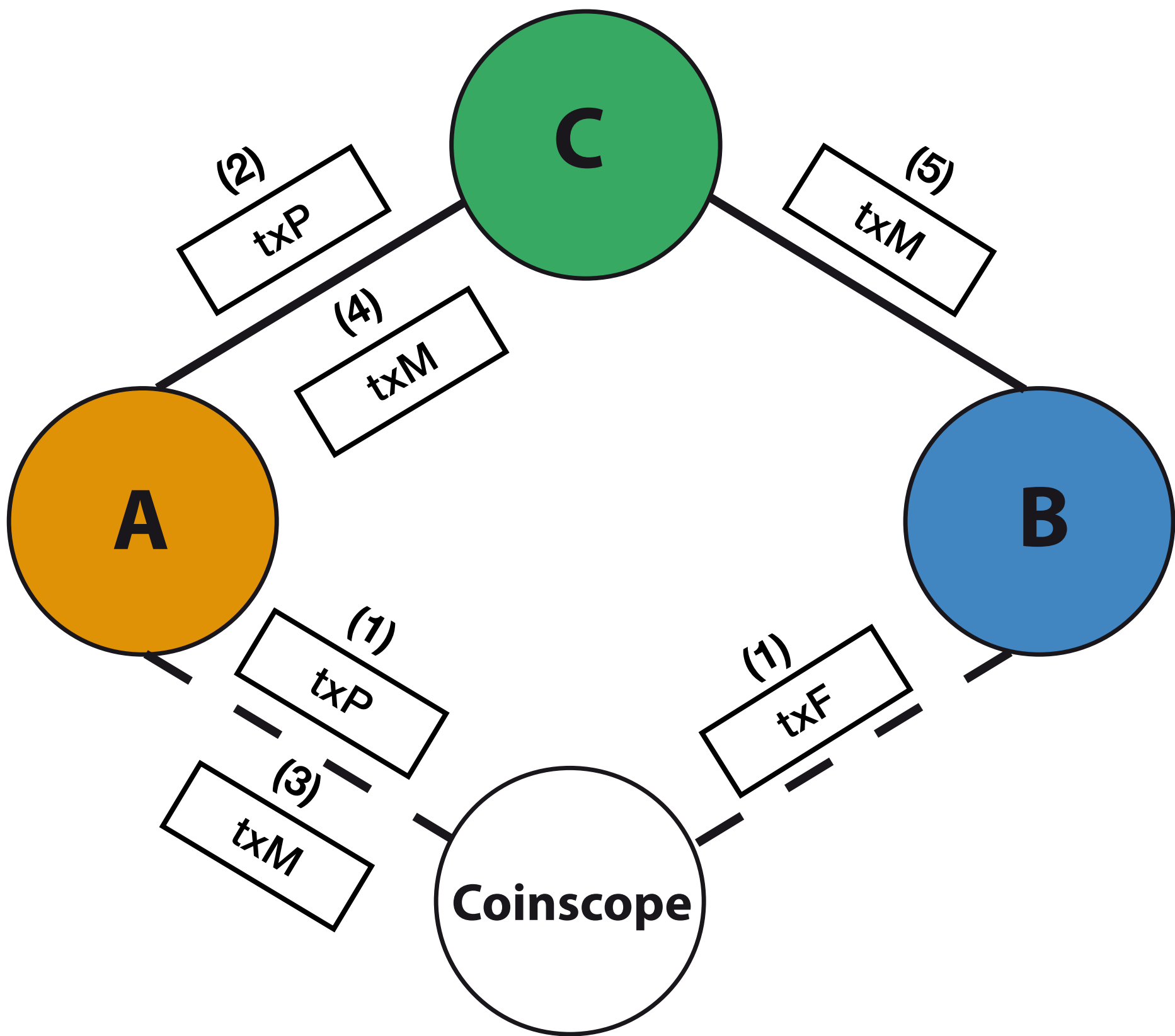
# ITS NOT THAT EASY



# ITS NOT THAT EASY



Long story short, if you add an additional node to the equation it will fail



A's Mempool

txP	(1)
txM	(3)

C's Mempool

txP	(2)
txM	(4)

B's Mempool

txF	(1)
-----	-----

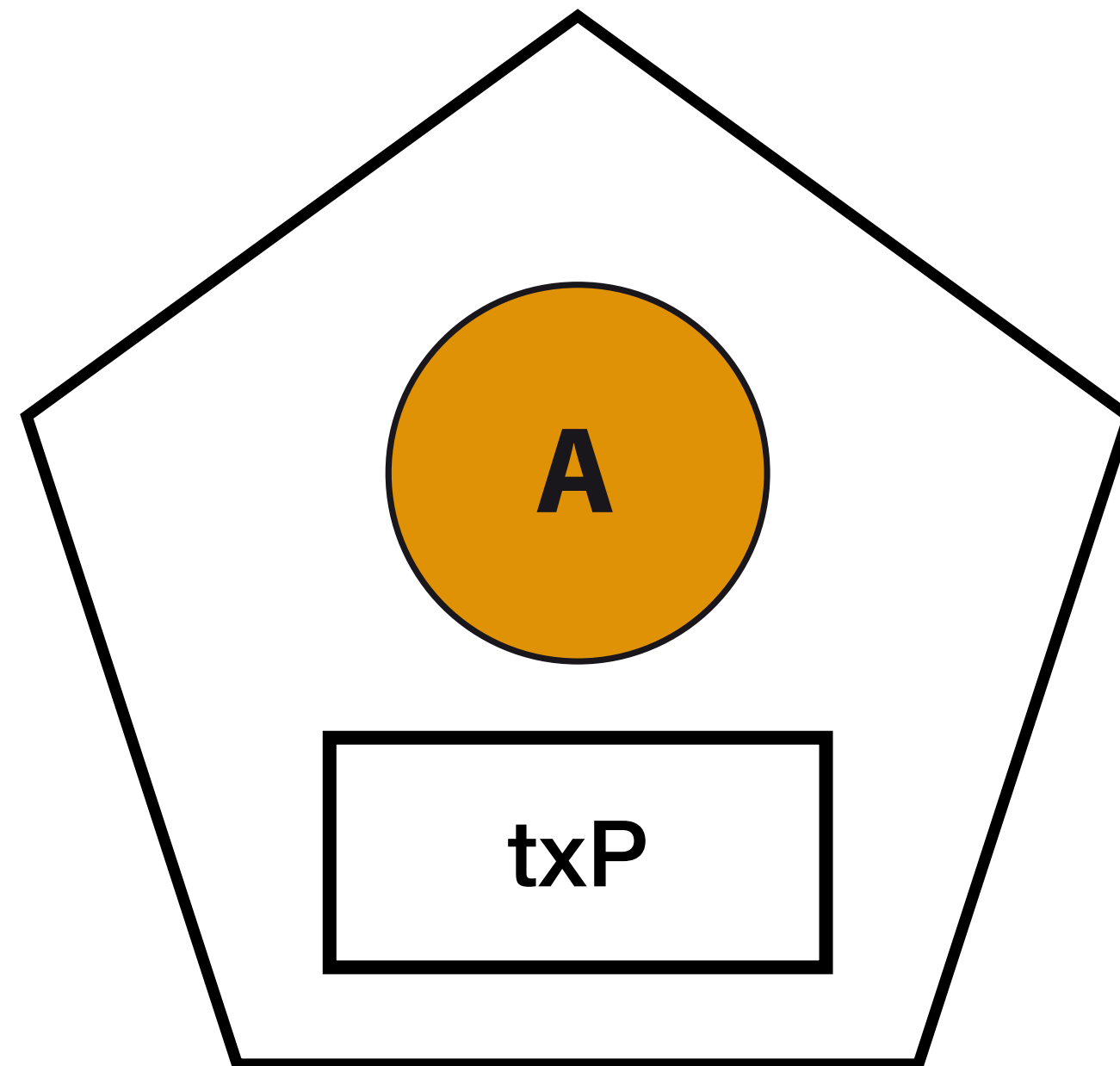
B's MapOrphanTransactions

txM	(5)
-----	-----

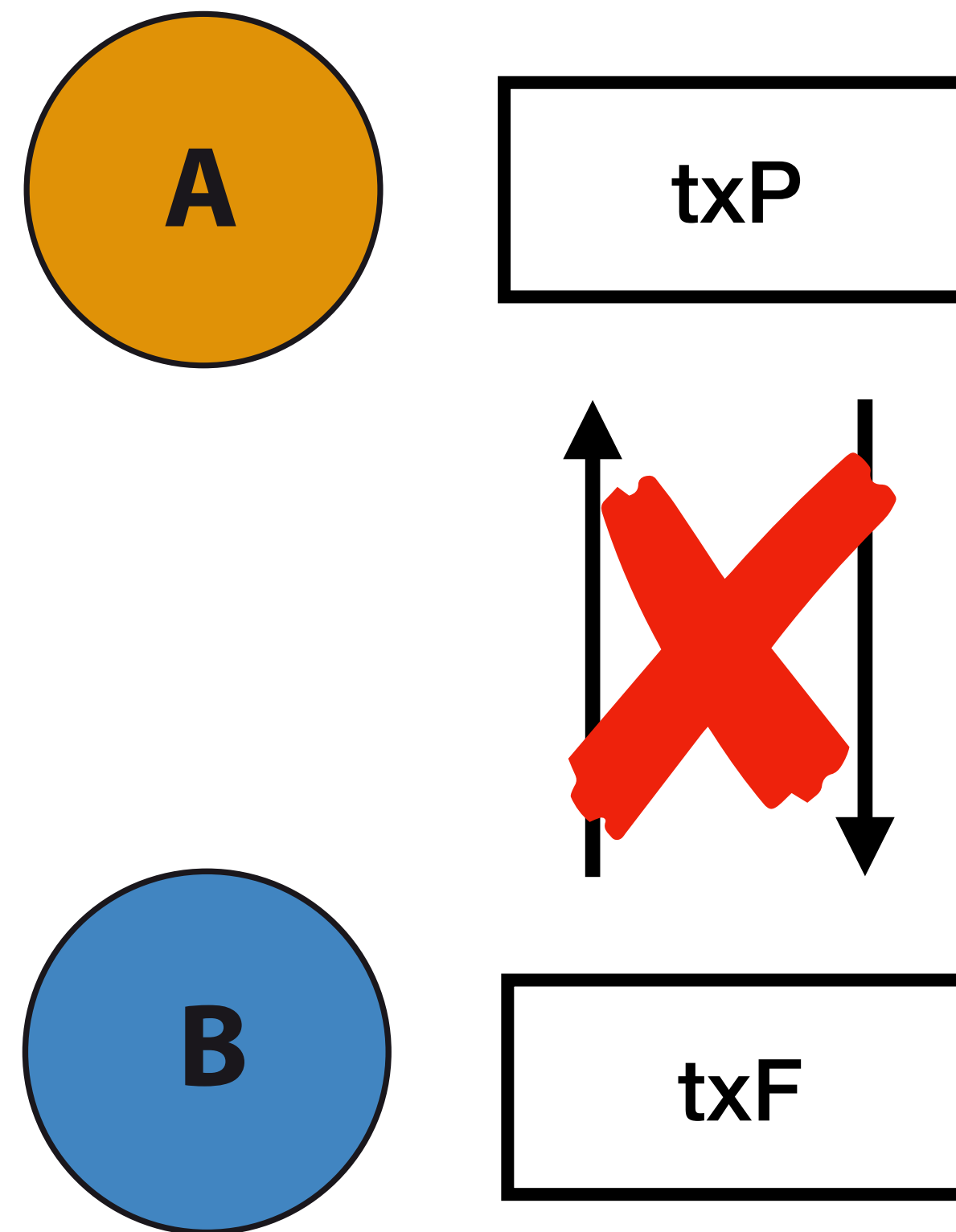


# MAKE THIS WORK IN A REAL NETWORK

## Isolation



## Synchronicity



## Scalability

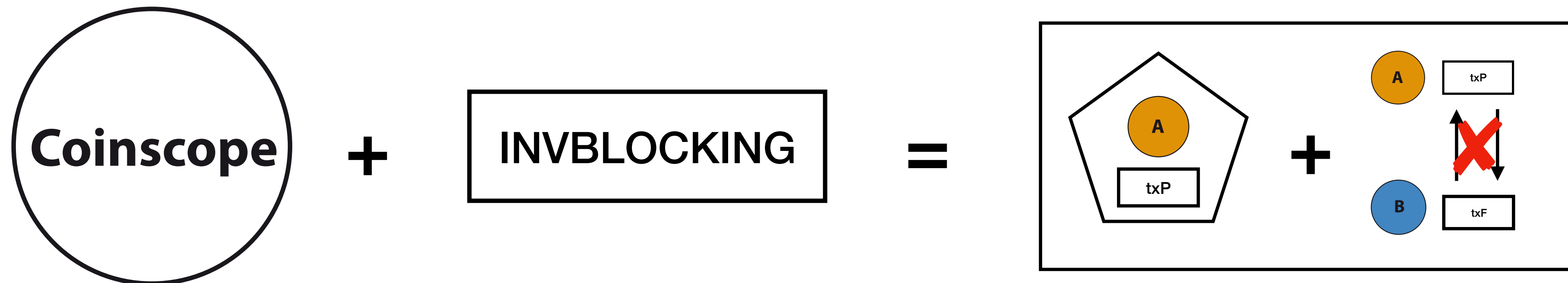
$$\approx 3n \text{ txs}$$



$$\approx 2\sqrt{n} \text{ txs}$$

$$n = \#nodes$$

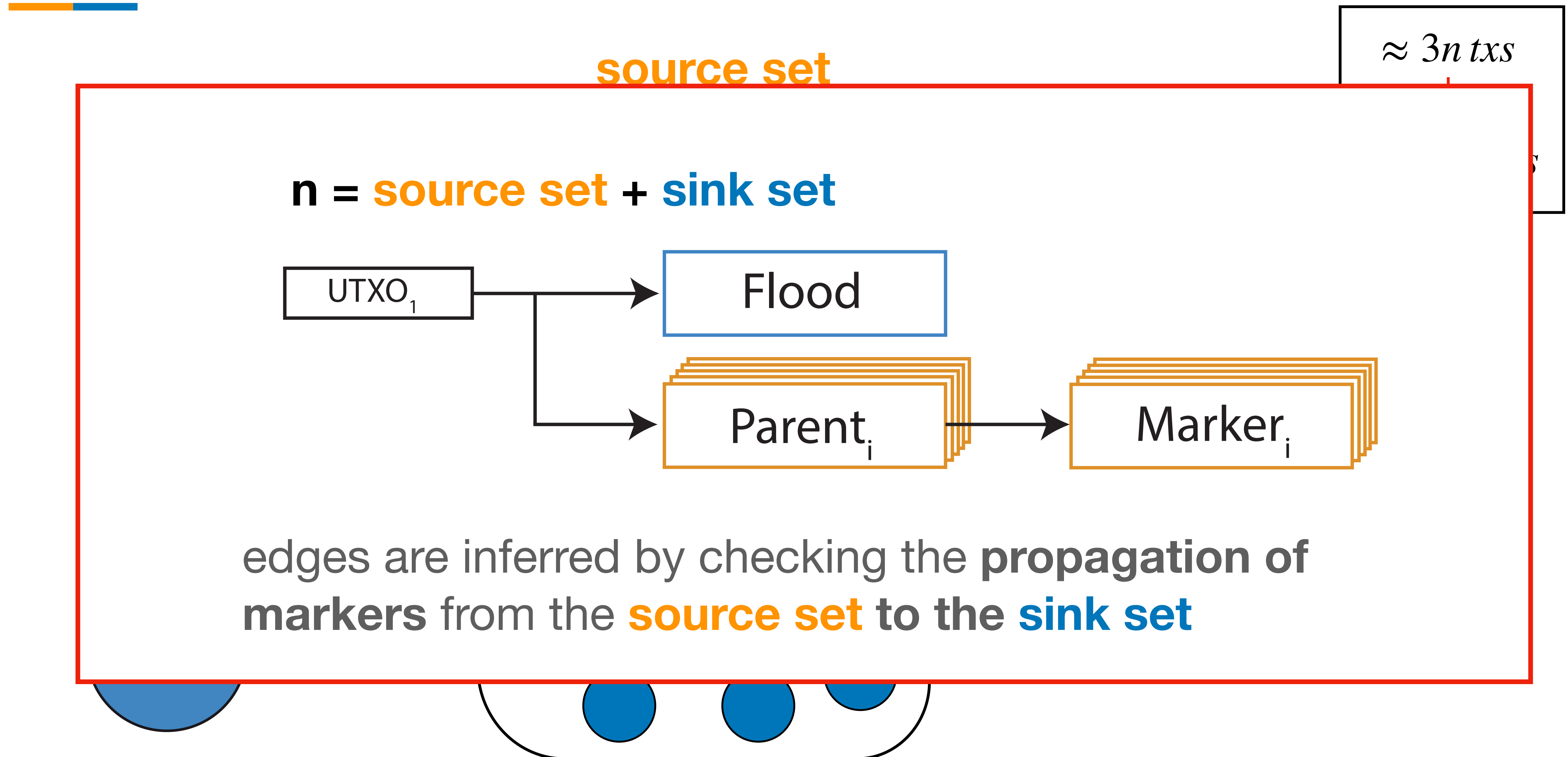
# ACHIEVING ISOLATION AND SYNCHRONICITY



## INVBLOCKING

- Send **INV** messages with **txP** and **txF** to the whole network
- Nodes will **ask us** about **txP** and **txF**
- We **withhold** the information effectively **blocking the propagation of txP and txF**

# ACHIEVING SCALABILITY



# ACHIEVING SCALABILITY (CONT)



To satisfy the **scalability property**, we are inferring several edges at the same time

We need to make sure that the MapOrphanTransactions pool **have enough room** to store all our orphans

Otherwise **false negatives** could occur



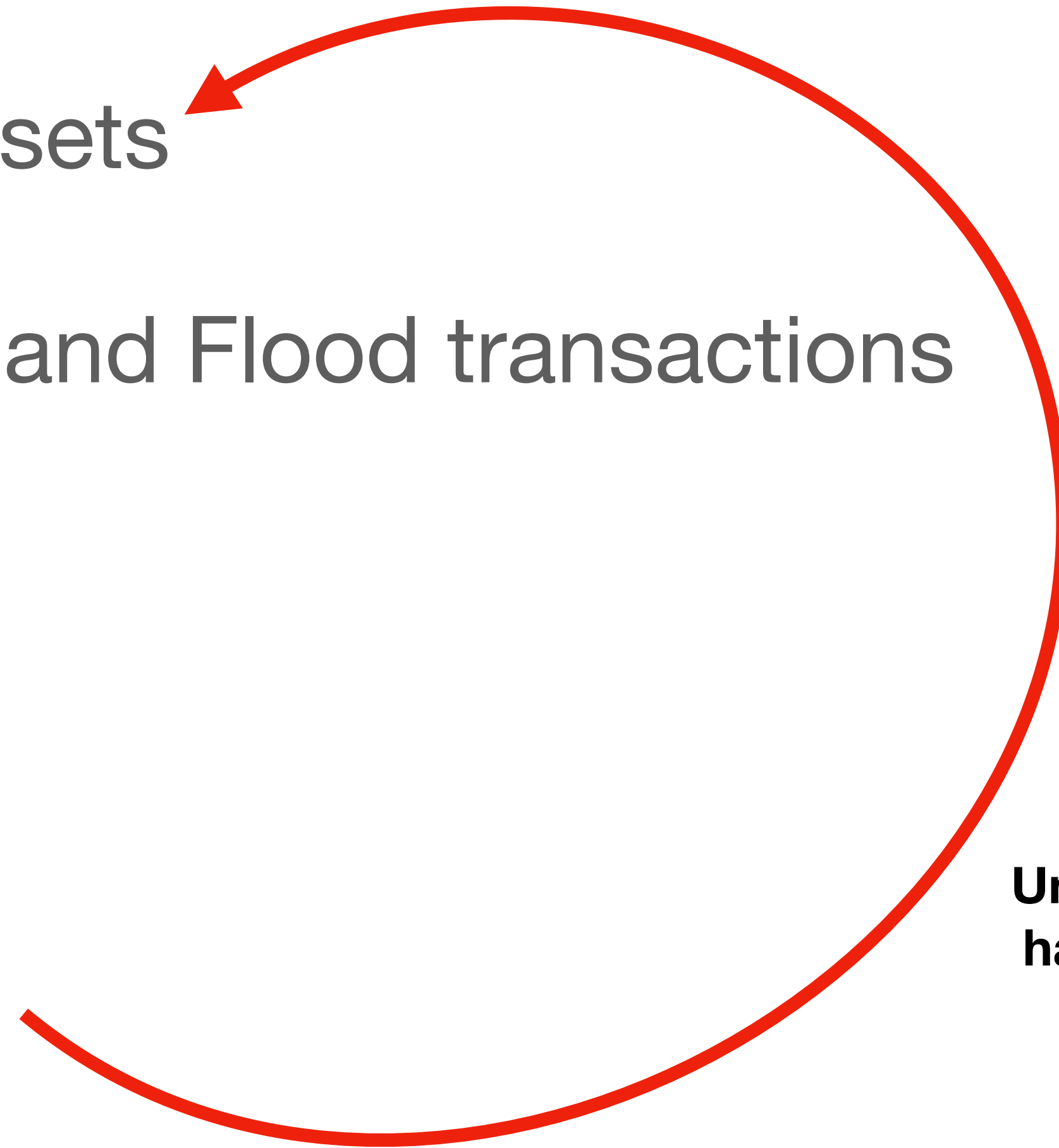
Refer to the paper for details about the orphan pool handling

# TXPROBE - PROTOCOL OVERVIEW



- **Choose** source and sink sets
- **Create** Parents, Markers and Flood transactions
- **INVBLOCK** the network
- **Send** transactions
- **Request** markers back

**Until every pair of nodes  
have been in a different  
set at least once**



# TXPROBE - DATA VALIDATION



- We run 5 Bitcoin Core nodes as **ground truth**
- Nodes are included as part of the source set
- We define our precision / recall by checking how well can we infer the ground truth nodes connections

# TXPROBE - COSTS



For a network like **Bitcoin mainnet**:

nodes  $\approx$  10000

time  $\approx$  8.25 hours

cost = 573210-764280 satoshi (5 sat/byte)  $\approx$  **\$(20-30)**

**We can actually do better, add  
some notes about this**

# WHY TESTNET AND NO MAINNET?



- TxProbe is rather invasive: it empties the **MapOrphanTransactions pool** of all nodes in the network every round
- We could not measure the implication that such behavior may have had on the **propagation of regular transactions**
- The technique **could** have also **be seen as an attack** to the network