CRYPTOCURRENCIES (COMP0143): The Bitcoin Network Layer

Sergi Delgado Segura

LAST TIME (WEEK 3)

Differences between client/server and peer-to-peer paradigms

How a new node joins the network

- How it learns about the network
- How others learn about it

Actors and their role in the network

The gossip protocol

THIS TIME (WEEK 3)

Nodes misbehavior

Information propagation

Network based attacks

Network topology

Nodes misbehavior

Nodes misbehavior

Every node maintains a **banscore** with each of its neighbors

If a node finds that one if its peers is misbehaving, the former will increase the banscore of the later

If the banscore of a neighbor reaches (or surpasses) its maximum (**100 by default)**, the node will ban that neighbor for a certain time (**24h by default**)

The banscore increase depends on how the neighbor is misbehaving

Banscore

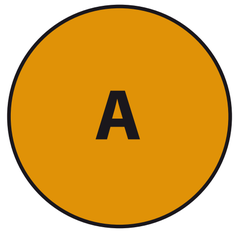
Examples of banscore increase:

- Not sending a version message as the first message in a handshake (**1**)
- Sending more than 1000 addresses in a single address message (**1**)
- Sending more than 50000 ids in a single inventory message (**20**)
- Sending a transaction with a script too big (**100)**

src/net_processing.cpp for more

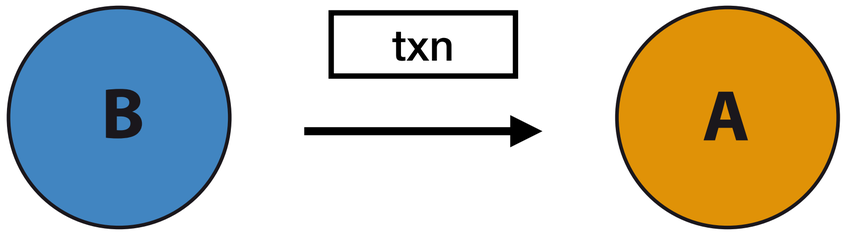
Information propagation

INFORMATION PROPAGATION (1/3)

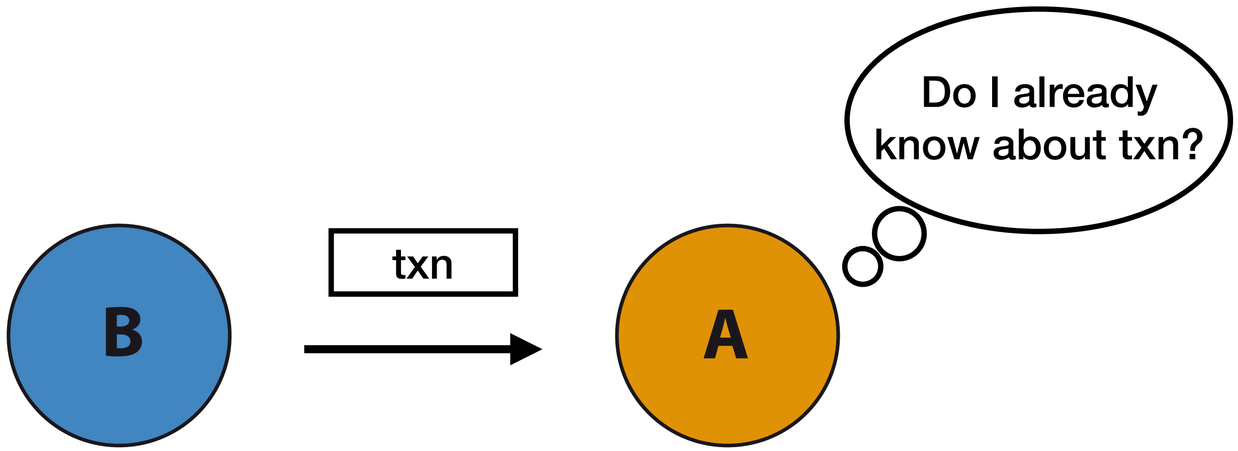


INFORMATION PROPAGATION (1/3)

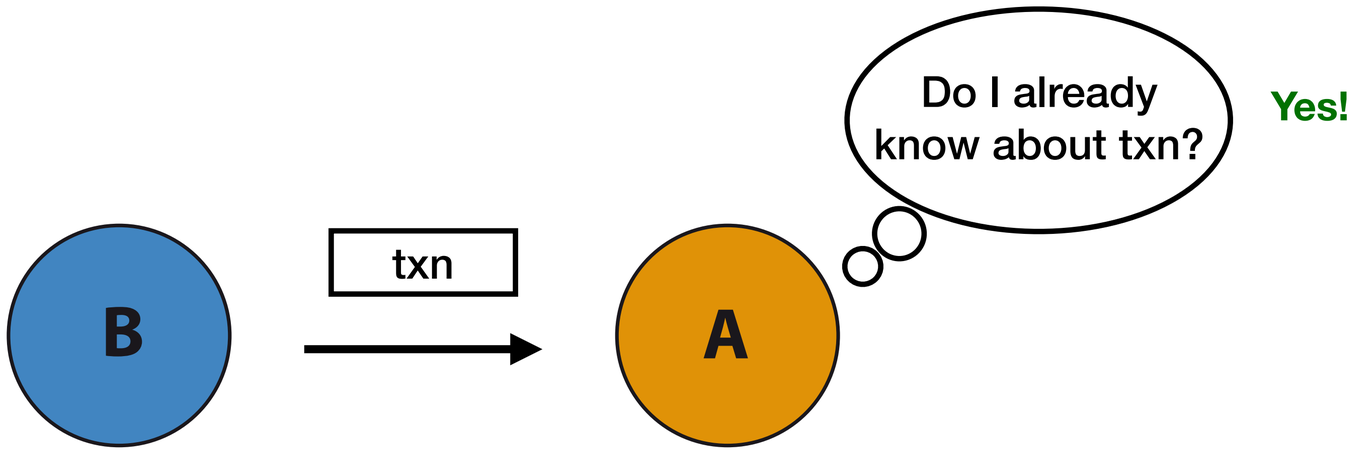
INFORMATION PROPAGATION (1/3)



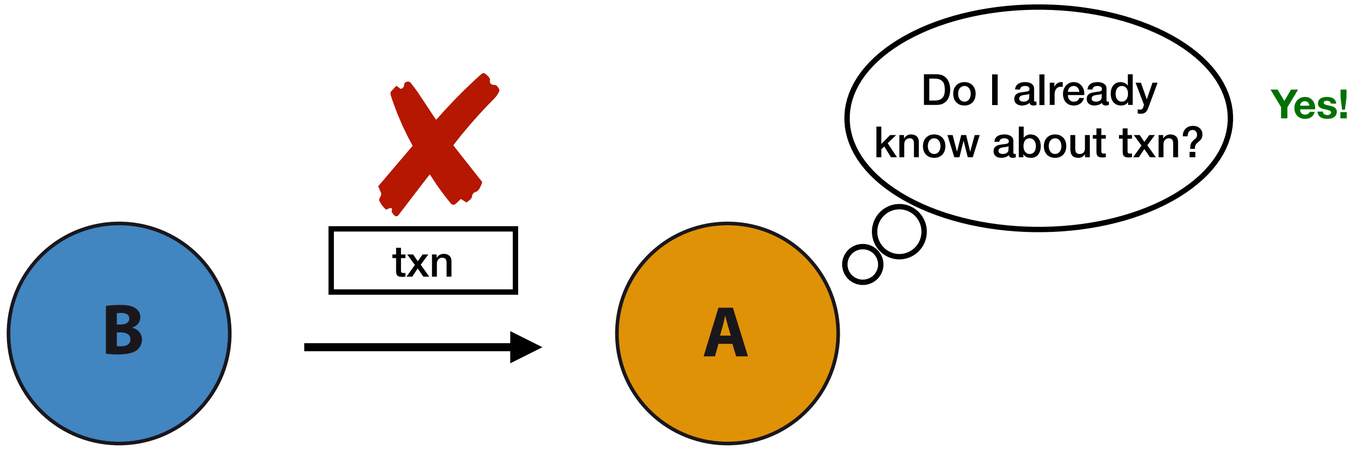
INFORMATION PROPAGATION (1/3)



INFORMATION PROPAGATION (1/3)

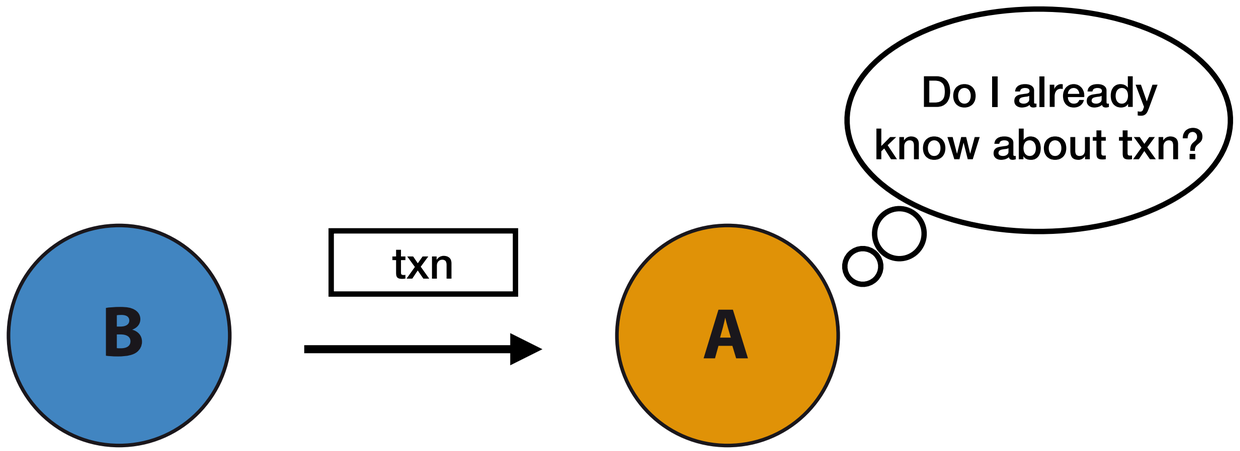


INFORMATION PROPAGATION (1/3)



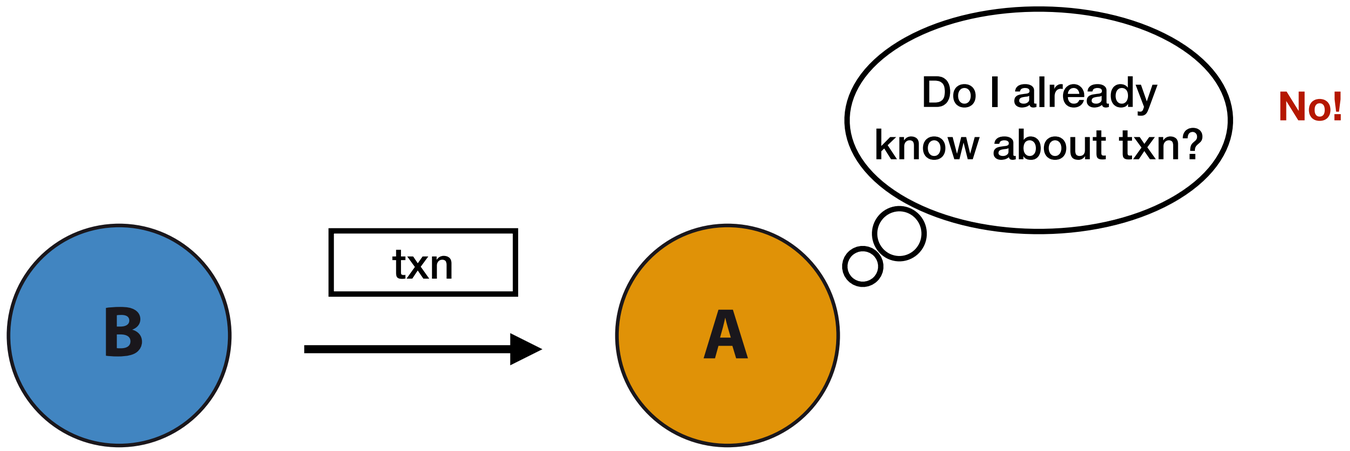
- Known transaction will be rejected

INFORMATION PROPAGATION (1/3)



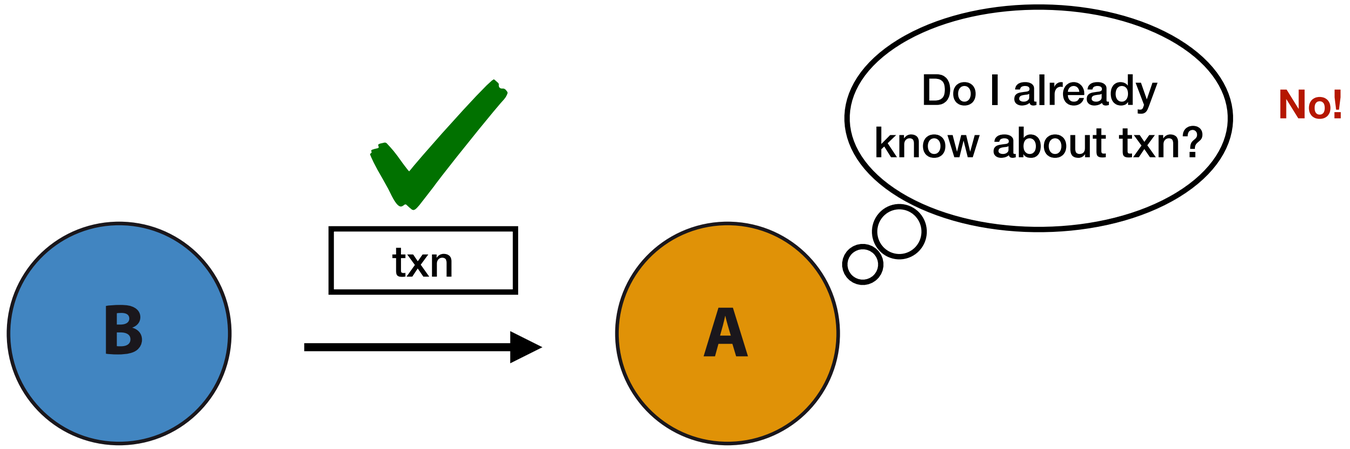
- Known transaction will be rejected

INFORMATION PROPAGATION (1/3)



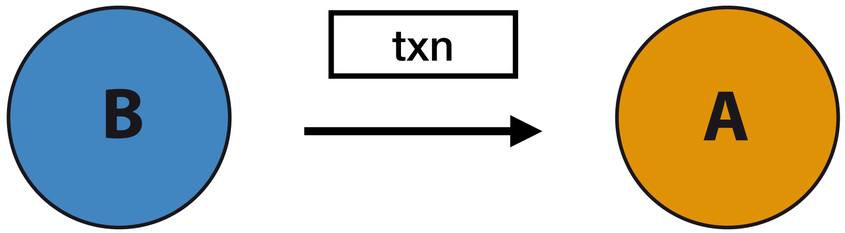
- Known transaction will be rejected

INFORMATION PROPAGATION (1/3)



- Known transaction will be rejected

INFORMATION PROPAGATION (1/3)



- Known transaction will be rejected

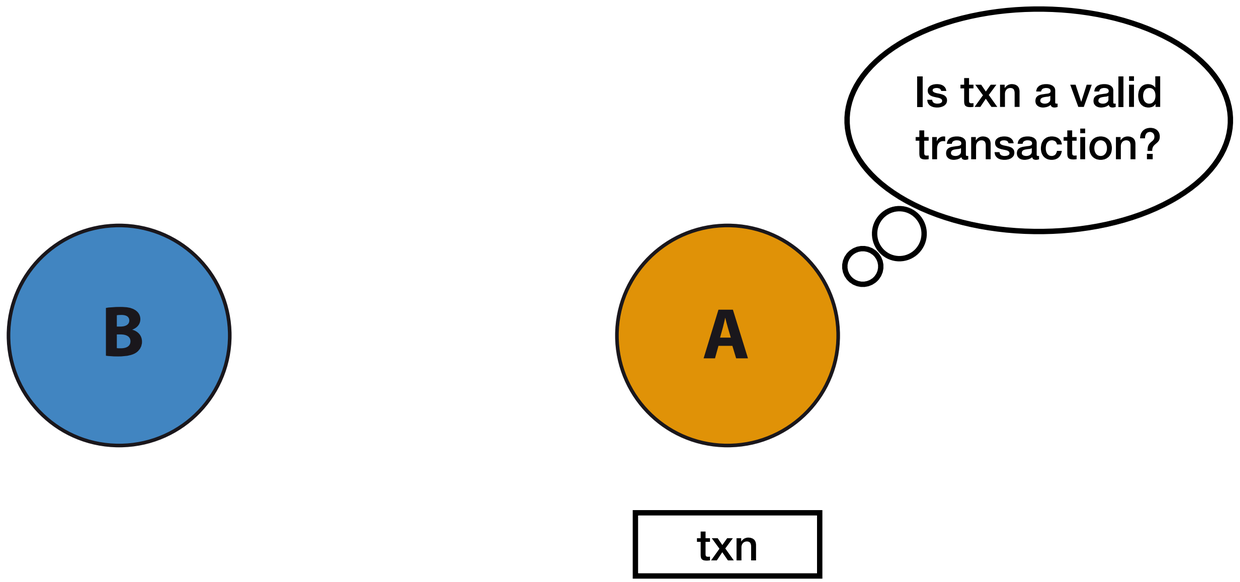
INFORMATION PROPAGATION (1/3)

- Known transaction will be rejected

INFORMATION PROPAGATION (1/3)

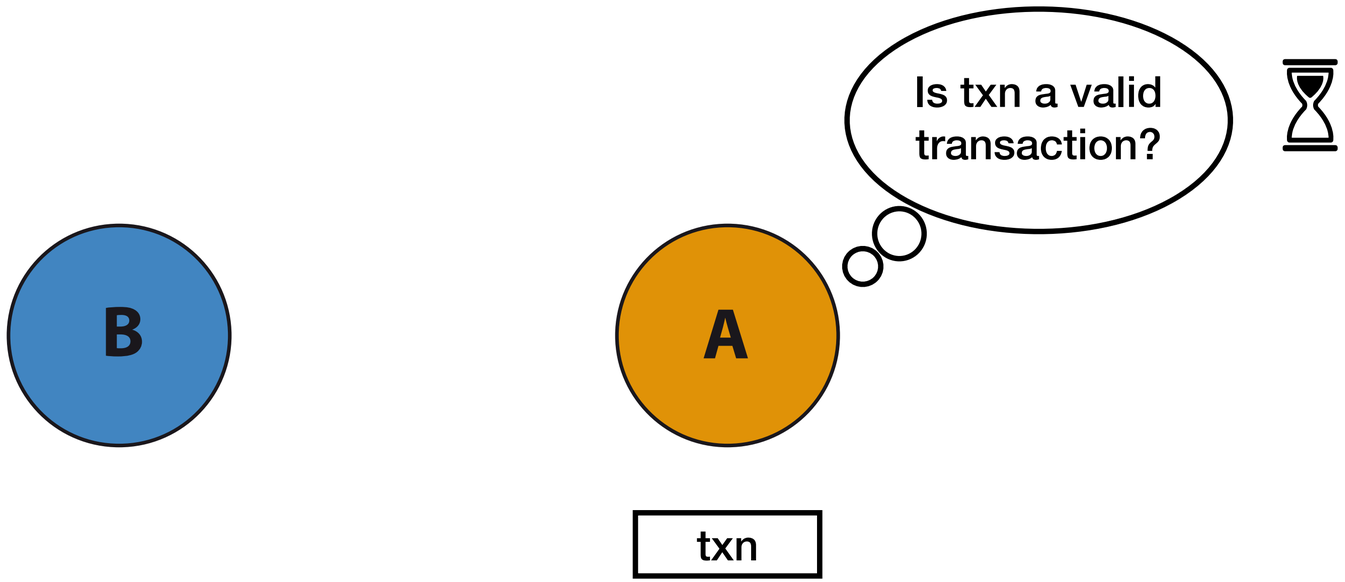
- Known transaction will be rejected

INFORMATION PROPAGATION (1/3)



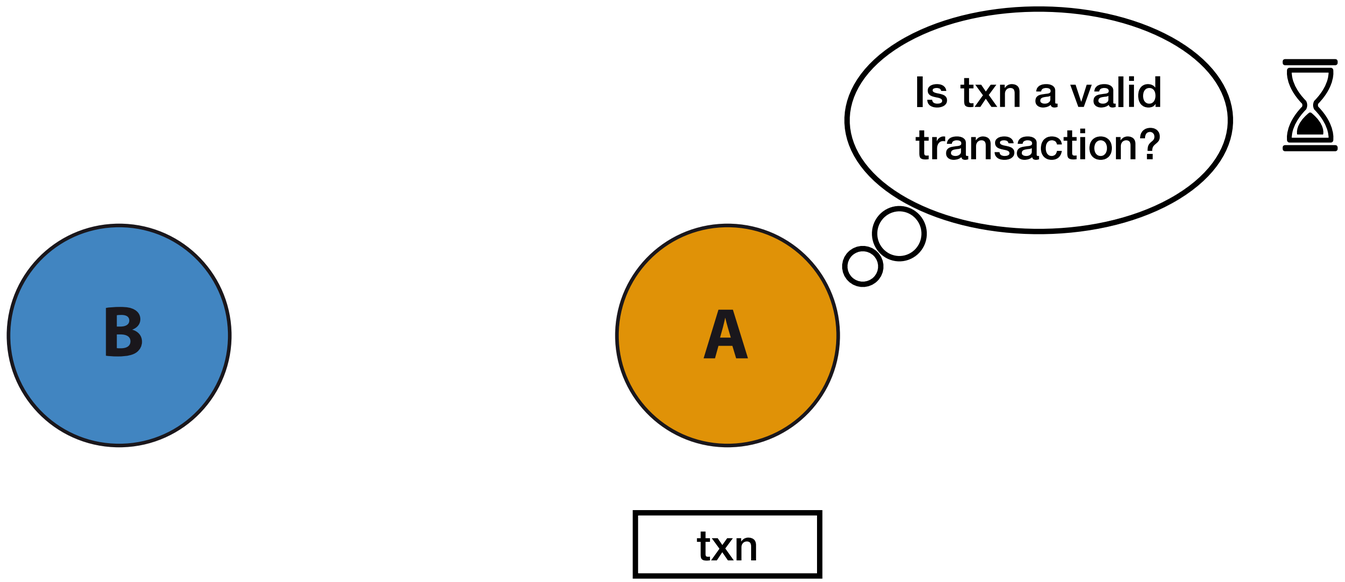
- Known transaction will be rejected

INFORMATION PROPAGATION (1/3)



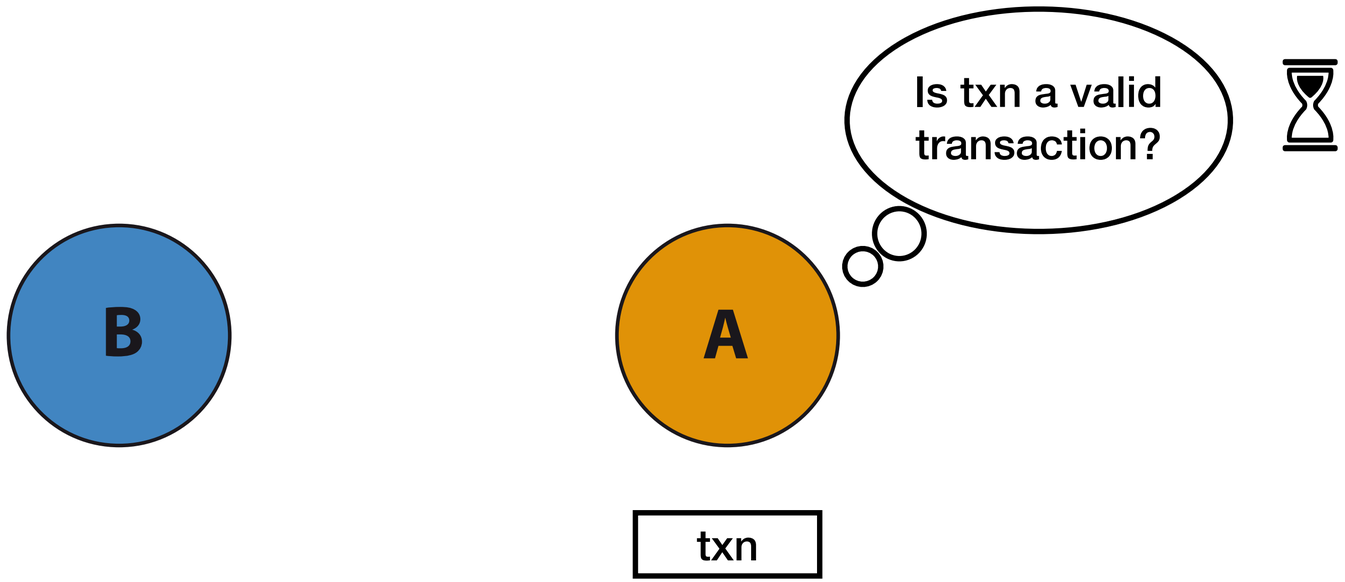
- Known transaction will be rejected

INFORMATION PROPAGATION (1/3)



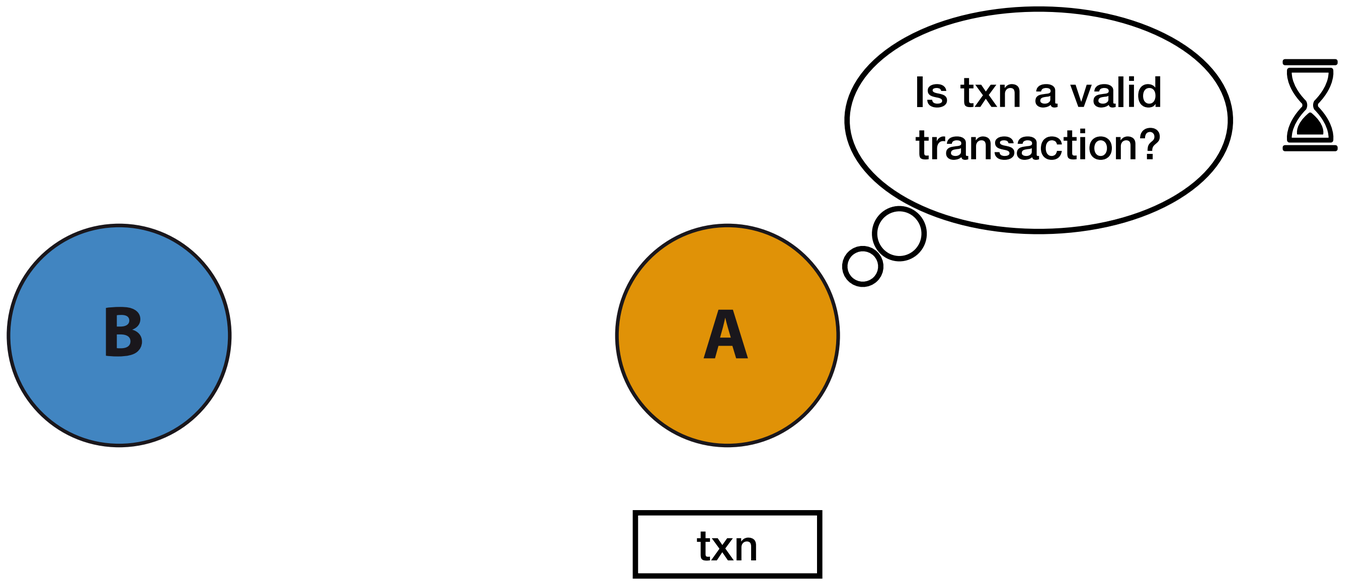
- Known transaction will be rejected

INFORMATION PROPAGATION (1/3)



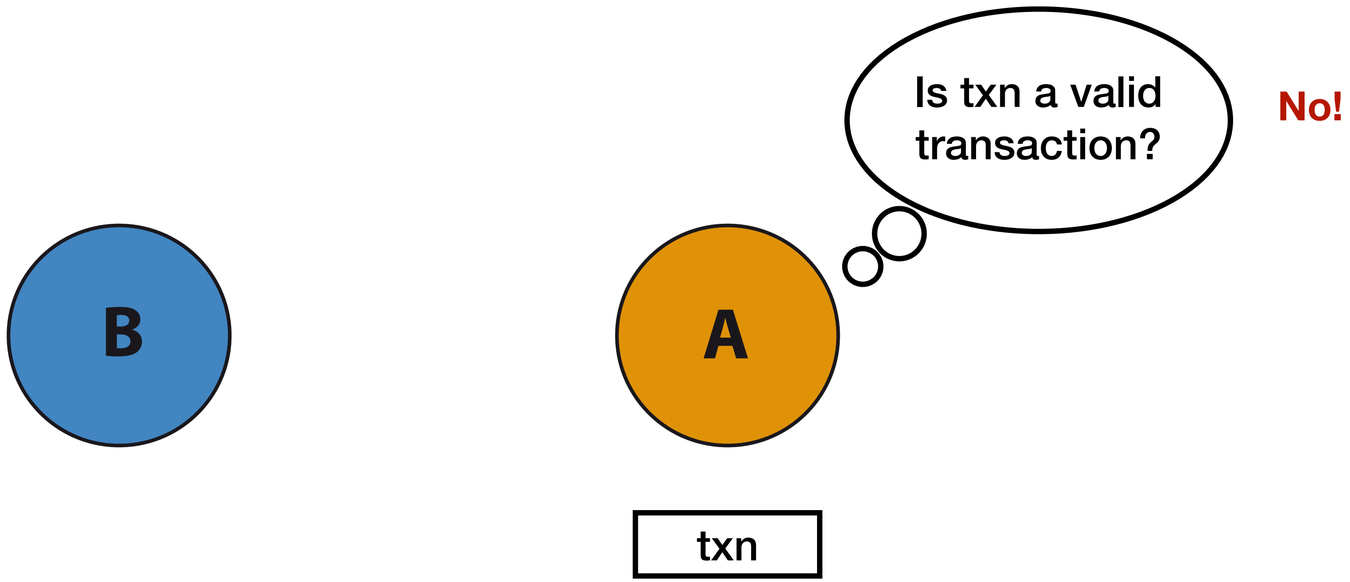
- Known transaction will be rejected

INFORMATION PROPAGATION (1/3)



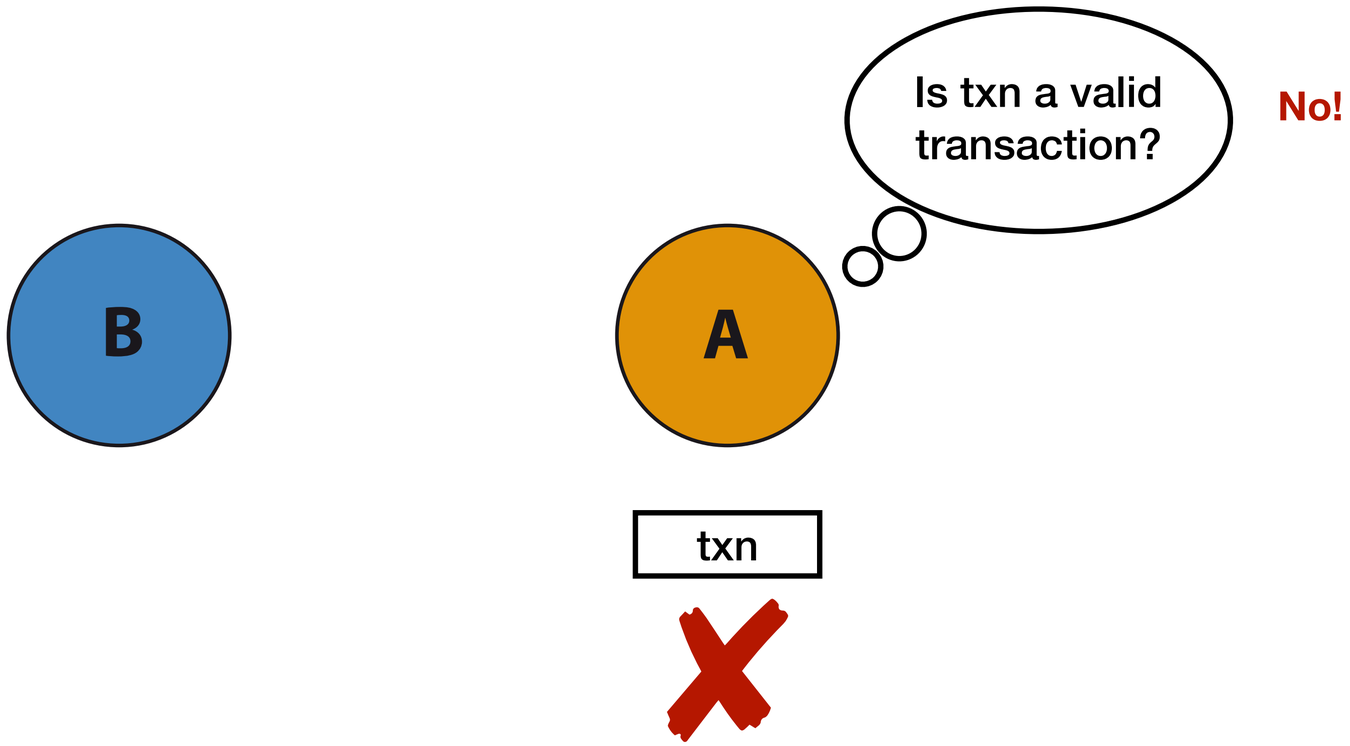
- Known transaction will be rejected

INFORMATION PROPAGATION (1/3)



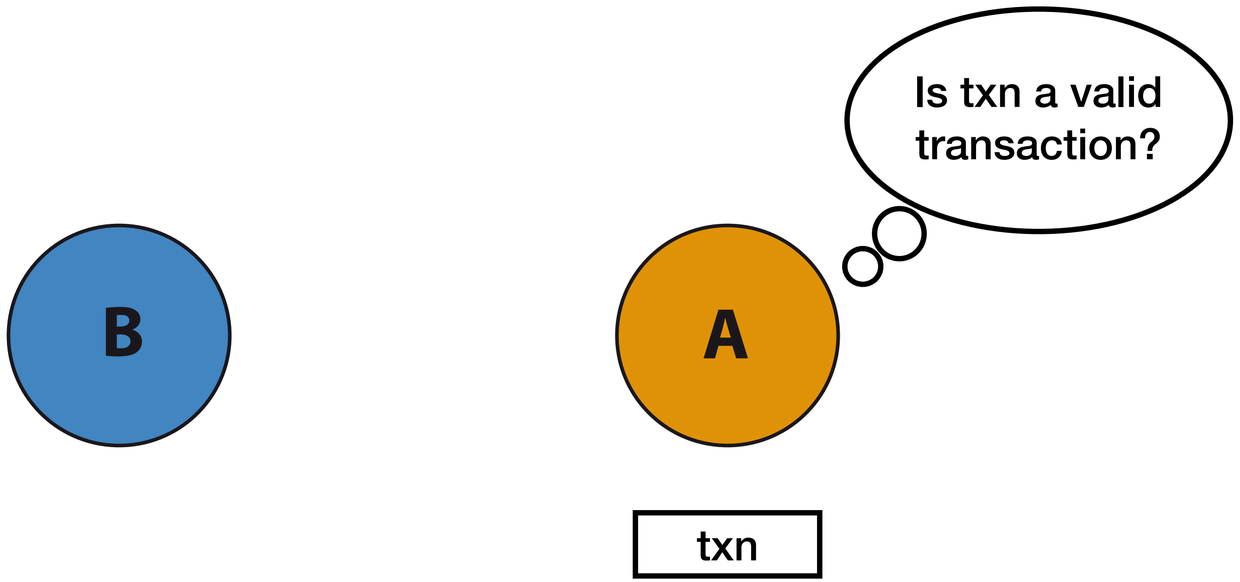
- Known transaction will be rejected

INFORMATION PROPAGATION (1/3)



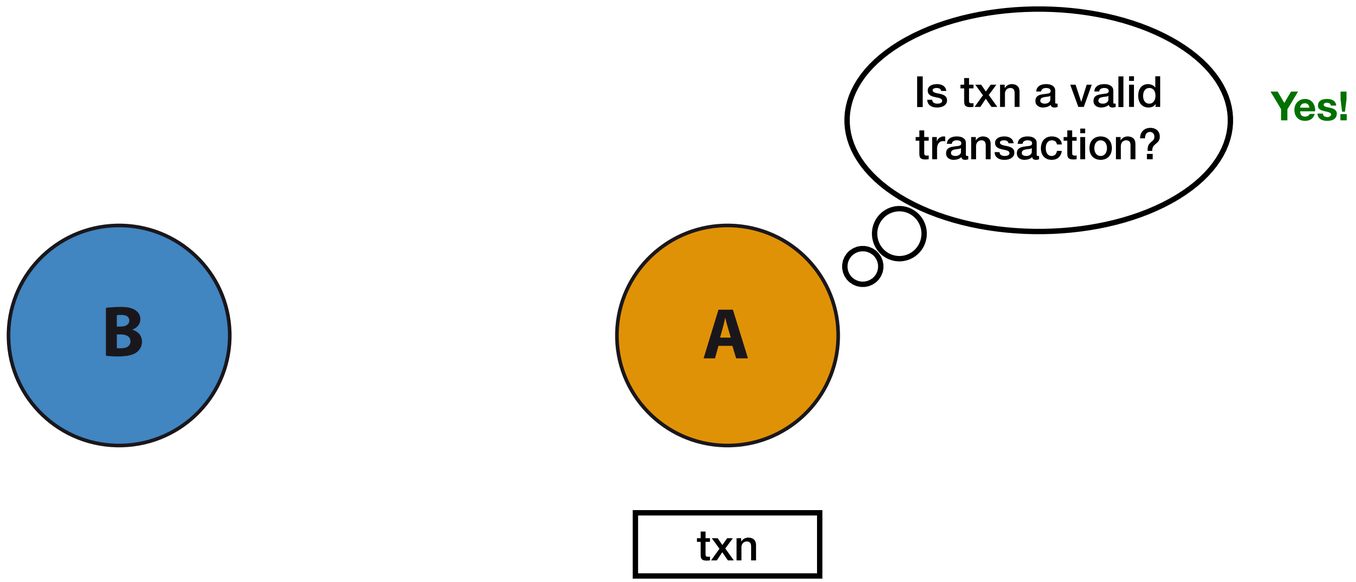
- Known transaction will be rejected
- Invalid transaction will also be rejected

INFORMATION PROPAGATION (1/3)



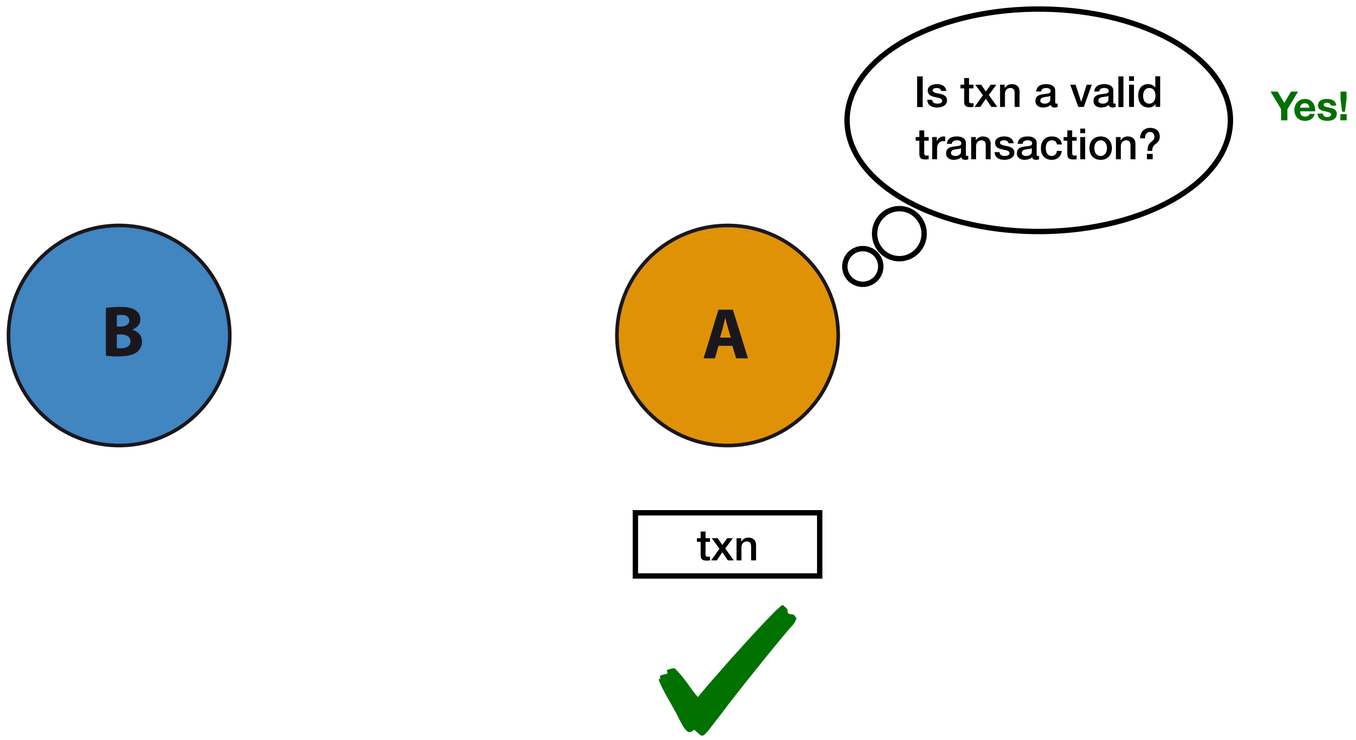
- Known transaction will be rejected
- Invalid transaction will also be rejected

INFORMATION PROPAGATION (1/3)



- Known transaction will be rejected
- Invalid transaction will also be rejected

INFORMATION PROPAGATION (1/3)

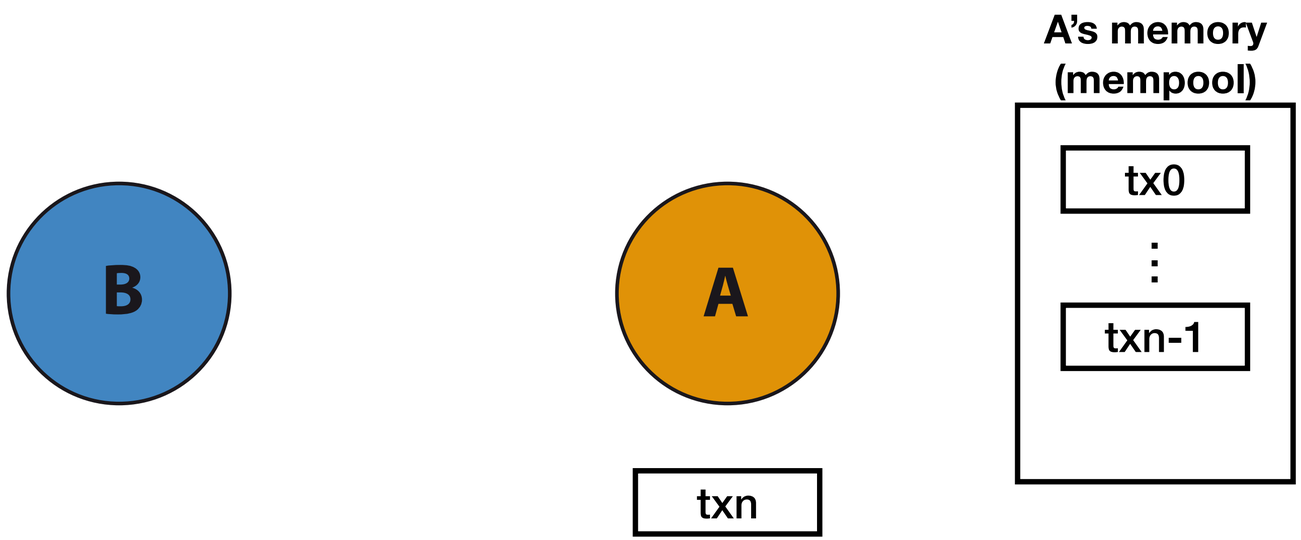


- Known transaction will be rejected
- Invalid transaction will also be rejected

INFORMATION PROPAGATION (1/3)

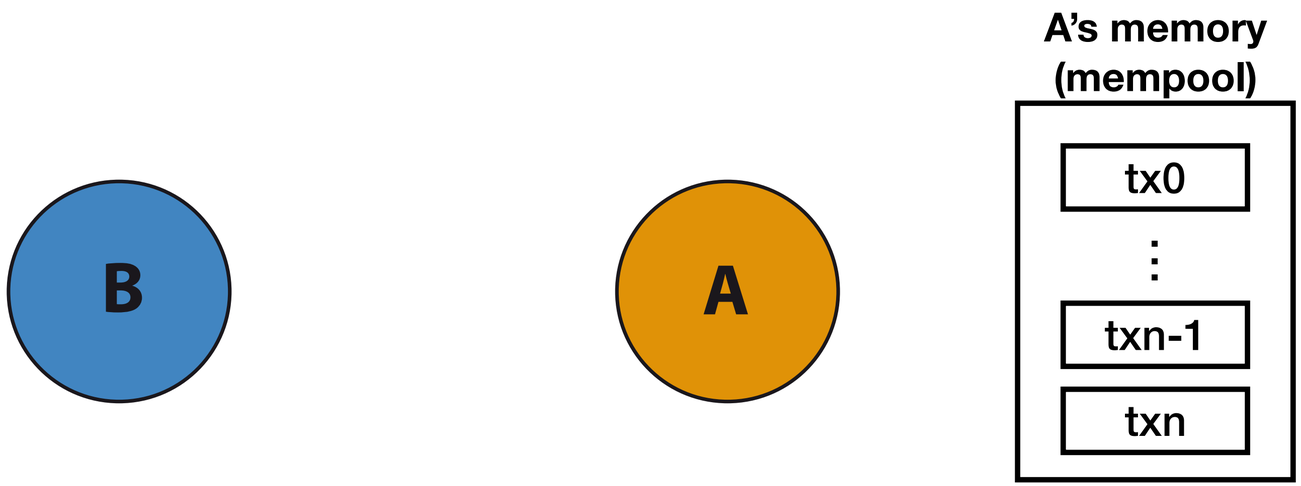
- Known transaction will be rejected
- Invalid transaction will also be rejected

INFORMATION PROPAGATION (1/3)



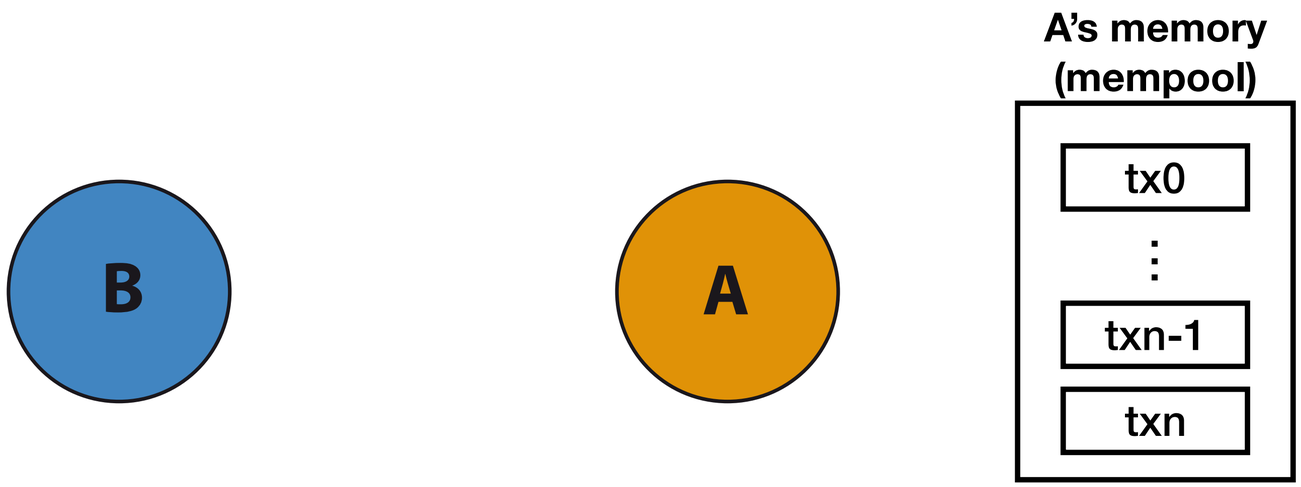
- Known transaction will be rejected
- Invalid transaction will also be rejected

INFORMATION PROPAGATION (1/3)



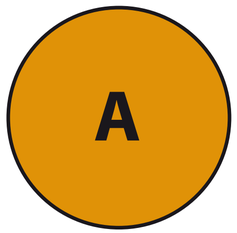
- Known transaction will be rejected
- Invalid transaction will also be rejected

INFORMATION PROPAGATION (1/3)

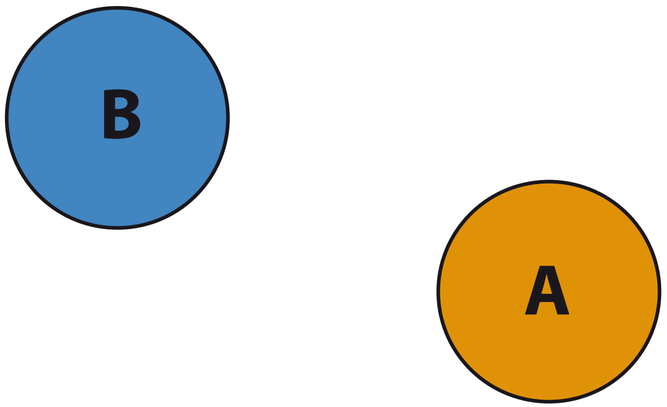


- Known transaction will be rejected
- Invalid transaction will also be rejected
- Valid (new) transactions will be kept in memory (mempool)

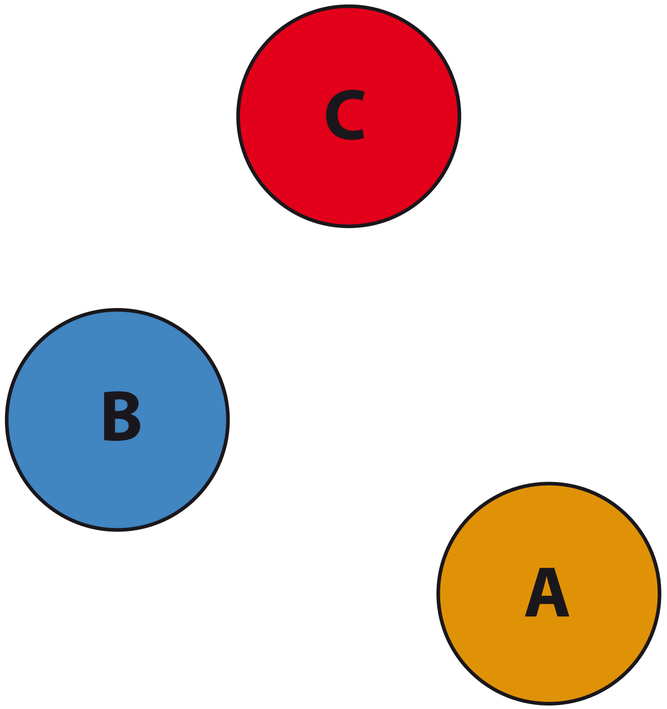
INFORMATION PROPAGATION (2/3)



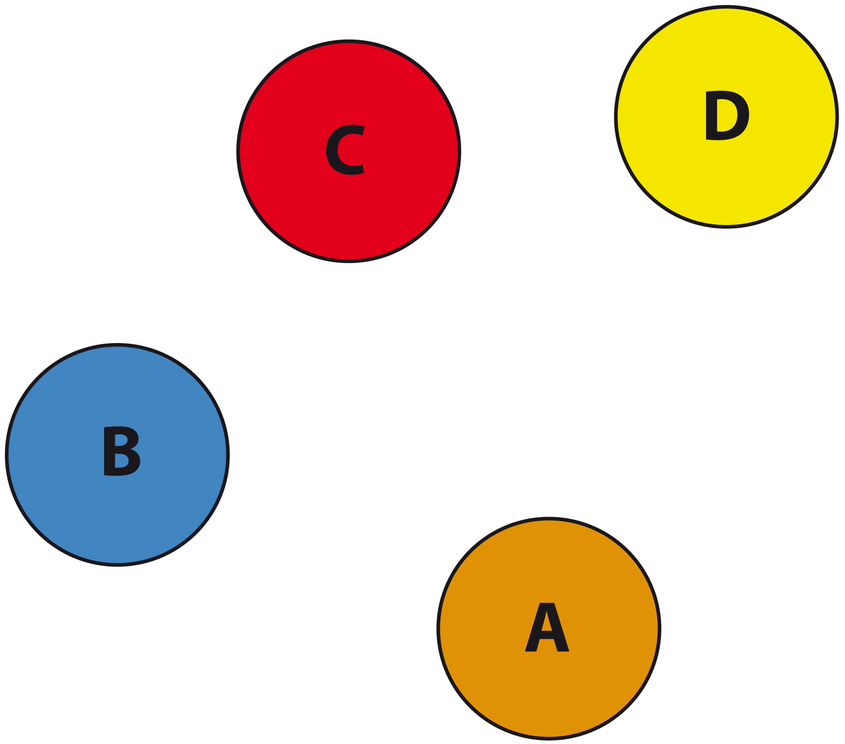
INFORMATION PROPAGATION (2/3)



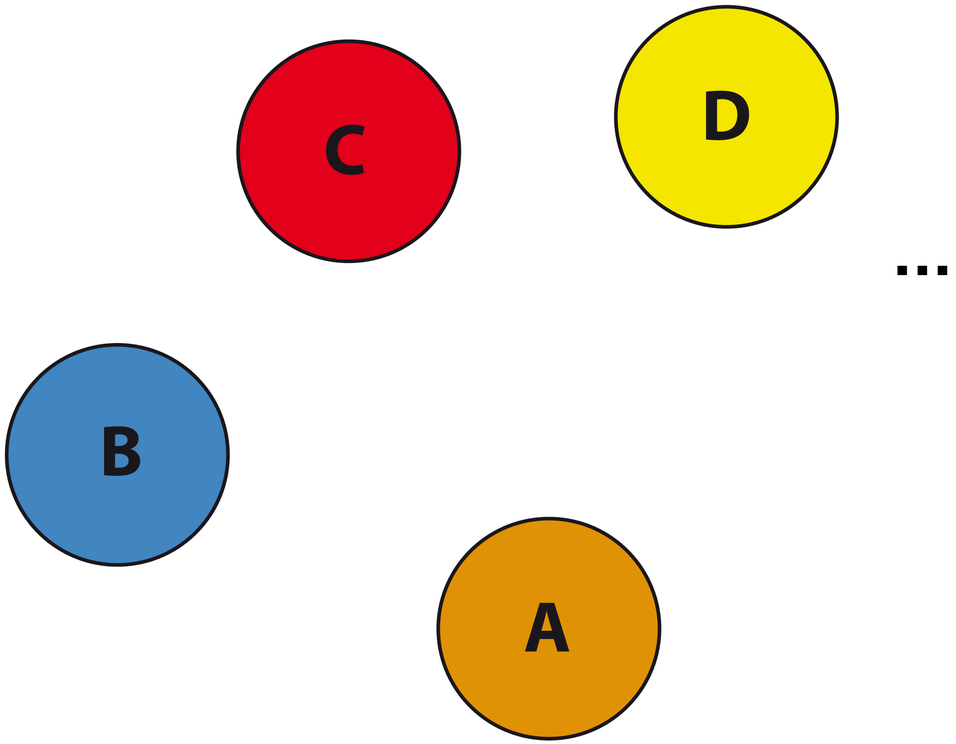
INFORMATION PROPAGATION (2/3)



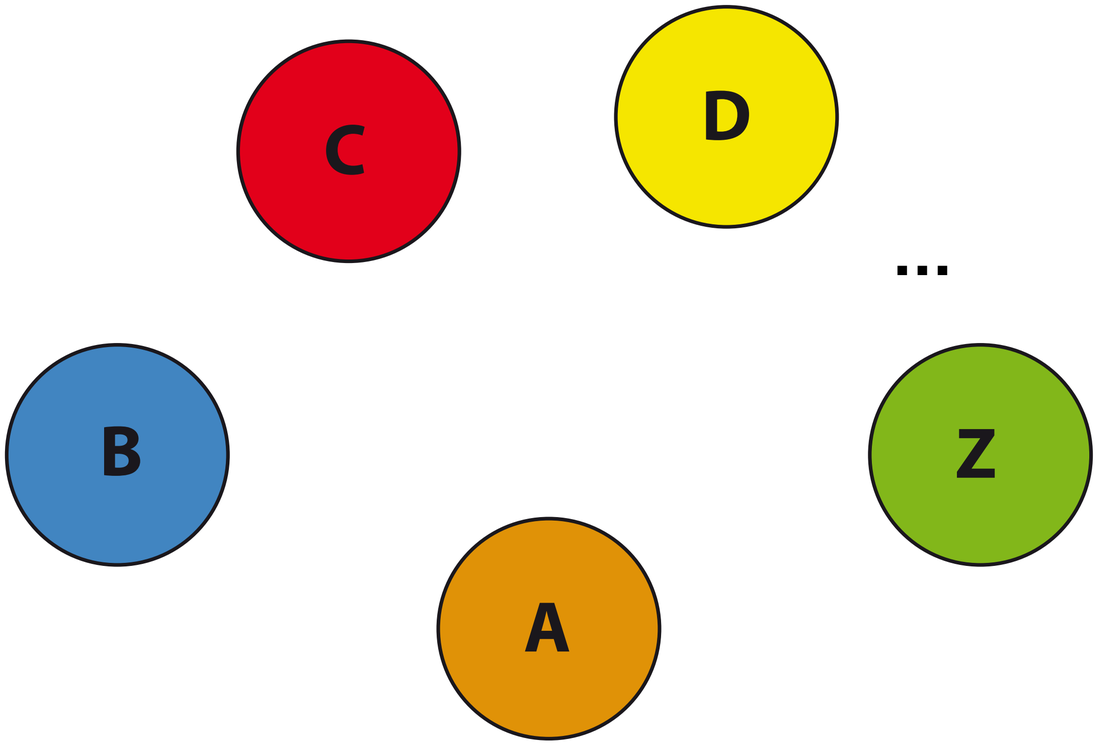
INFORMATION PROPAGATION (2/3)



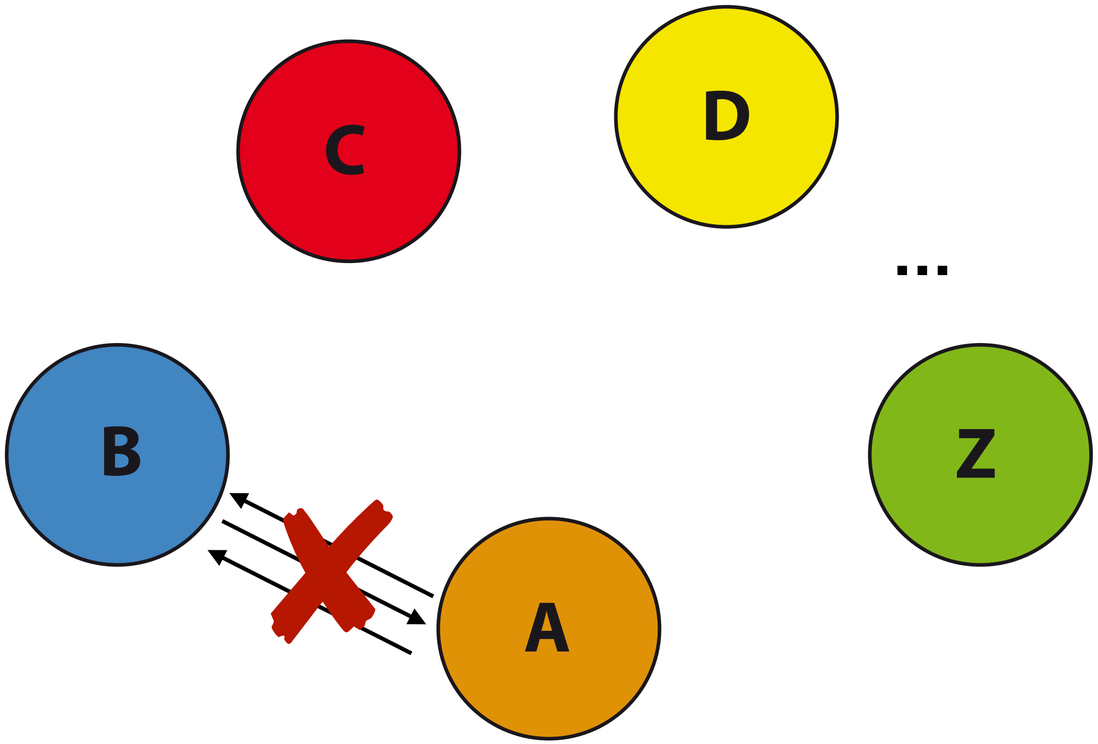
INFORMATION PROPAGATION (2/3)



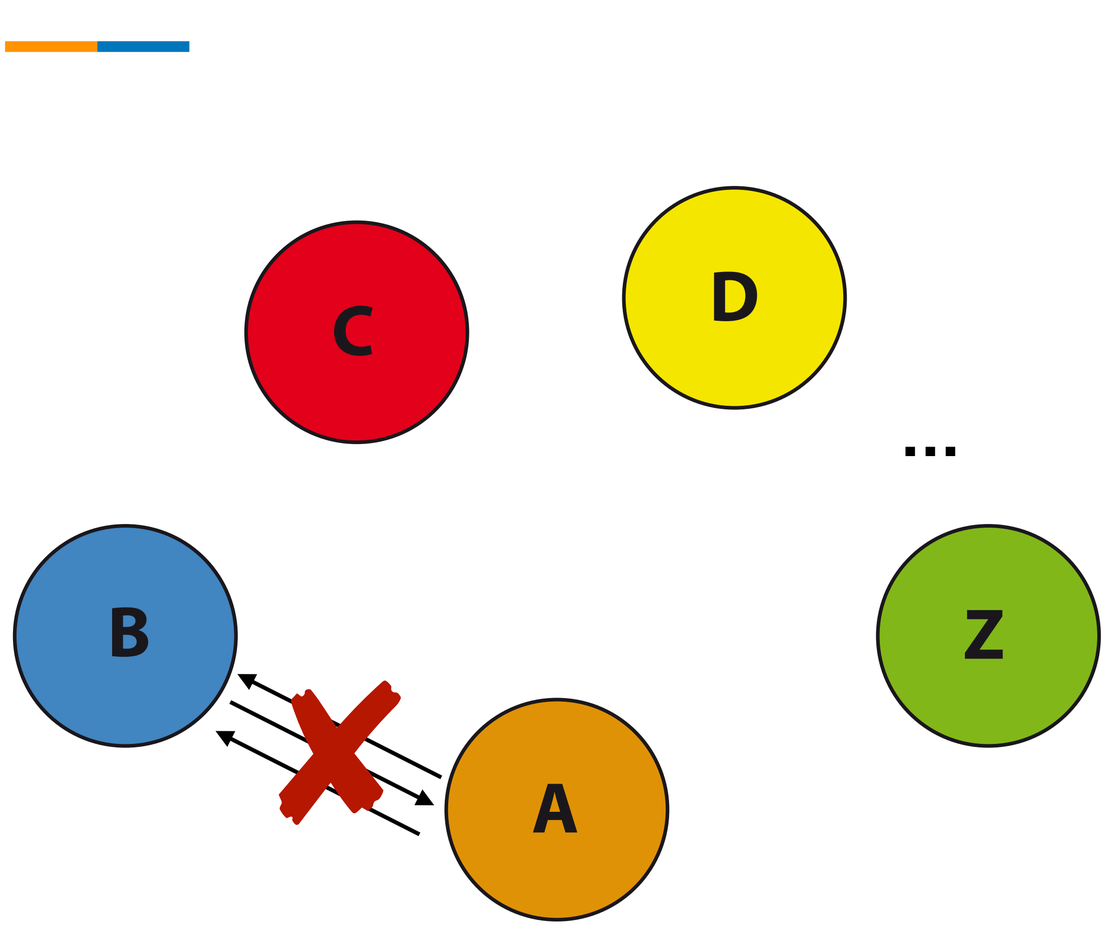
INFORMATION PROPAGATION (2/3)

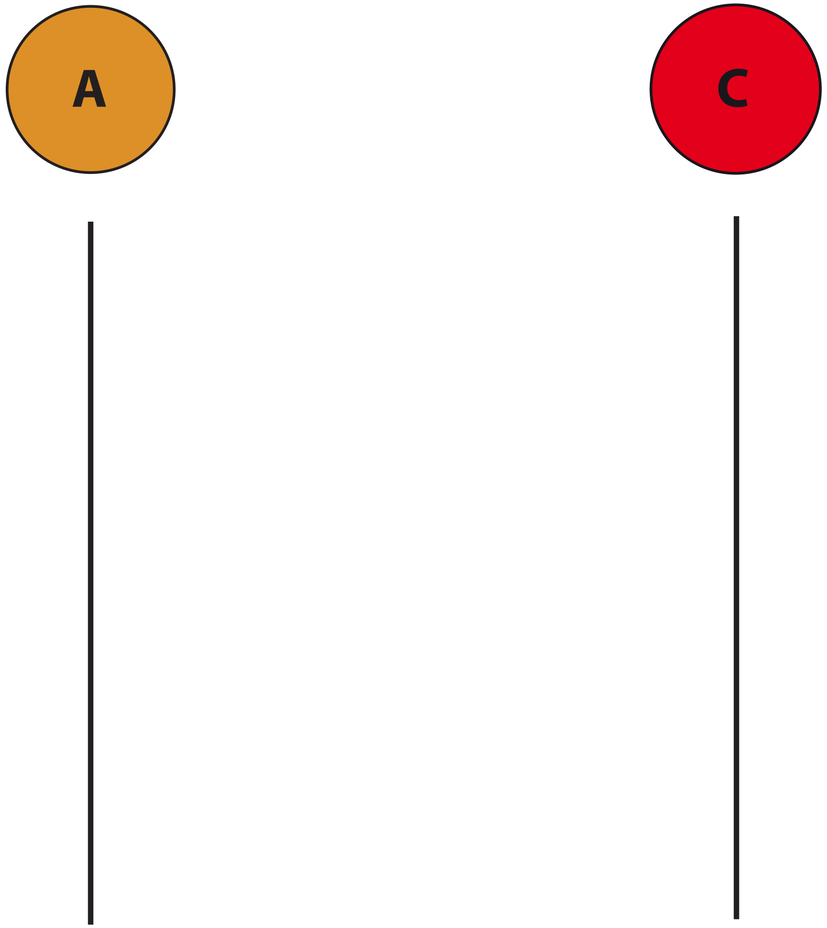


INFORMATION PROPAGATION (2/3)

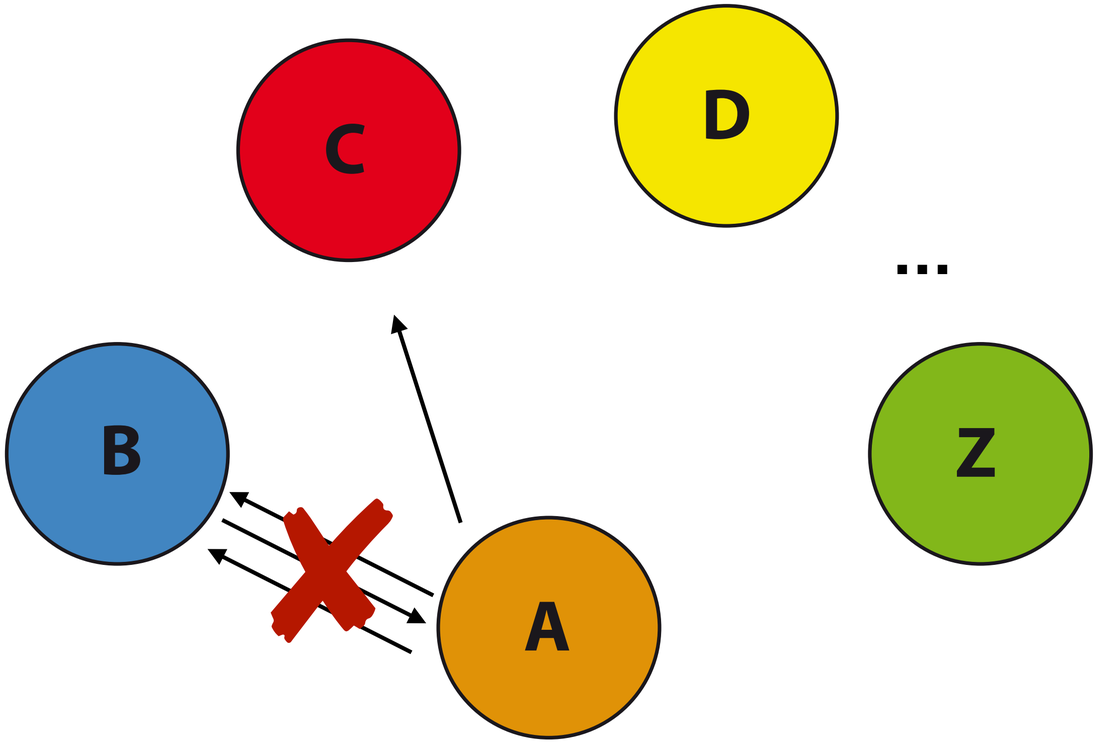


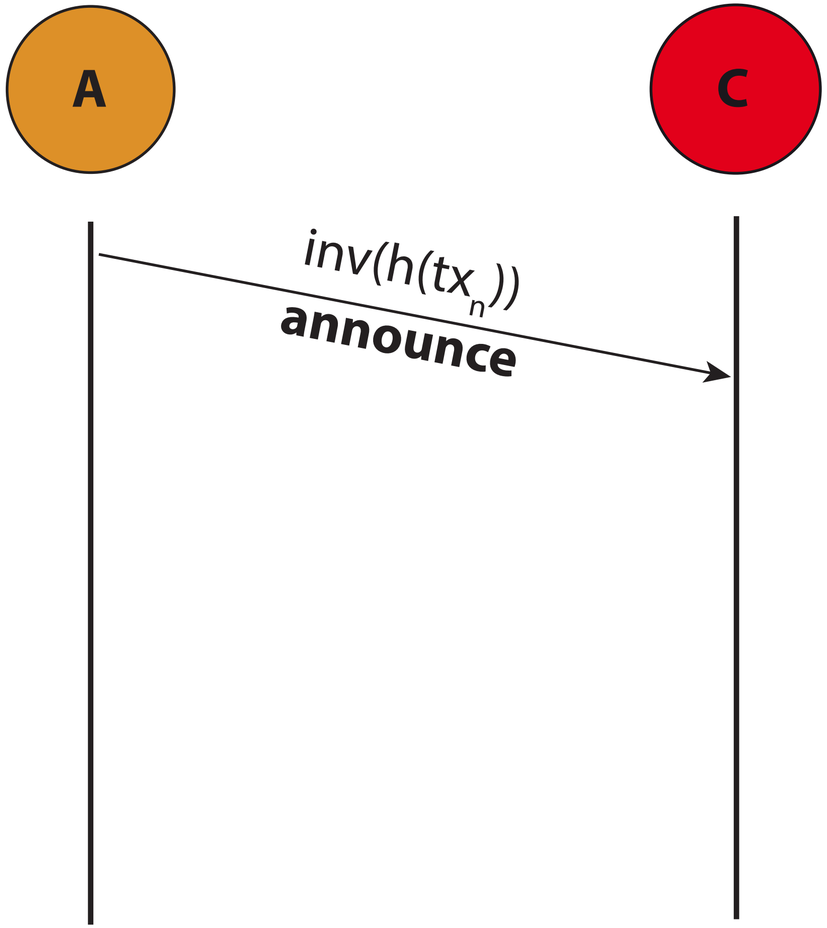
INFORMATION PROPAGATION (2/3)



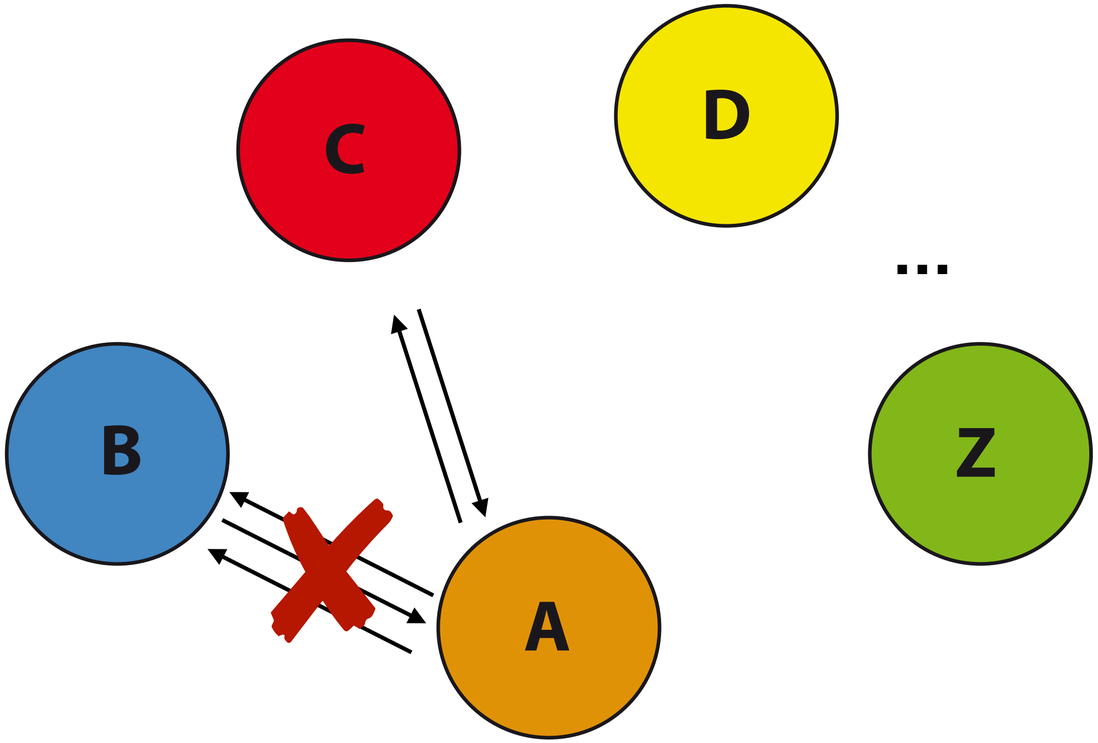


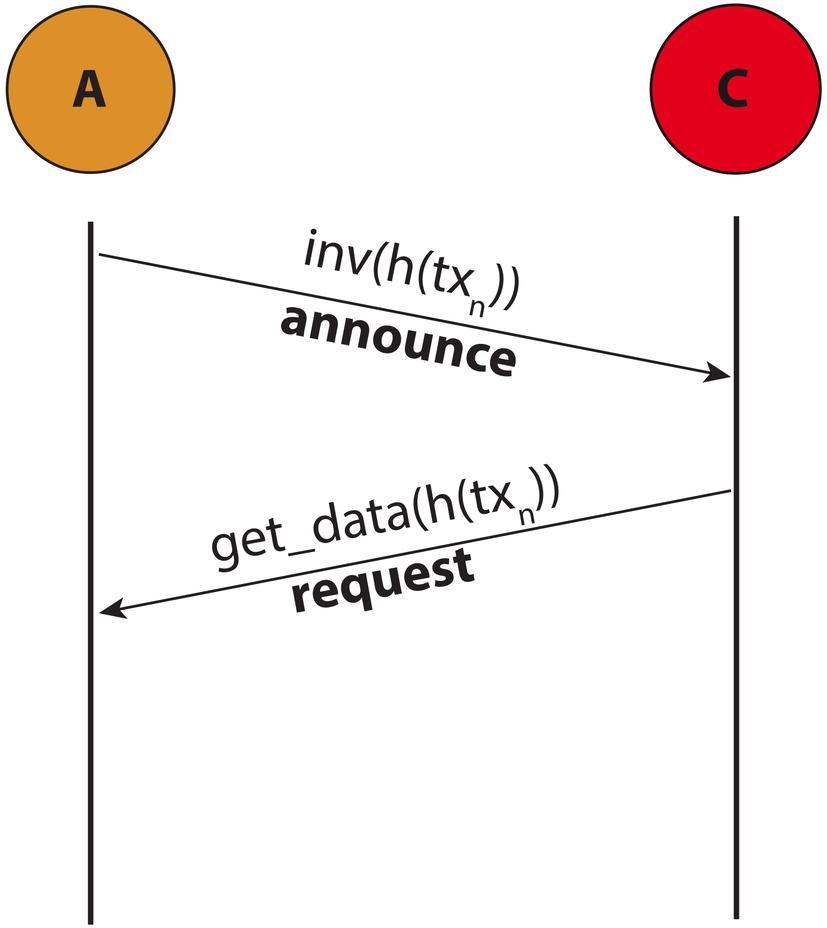
INFORMATION PROPAGATION (2/3)



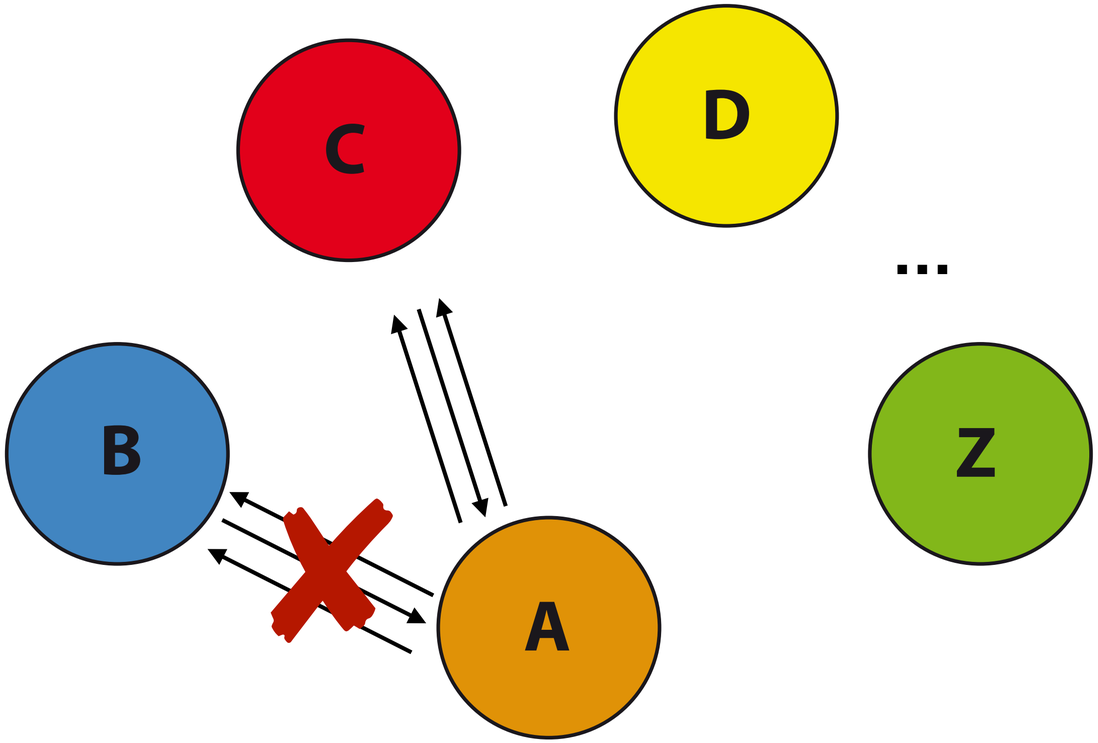


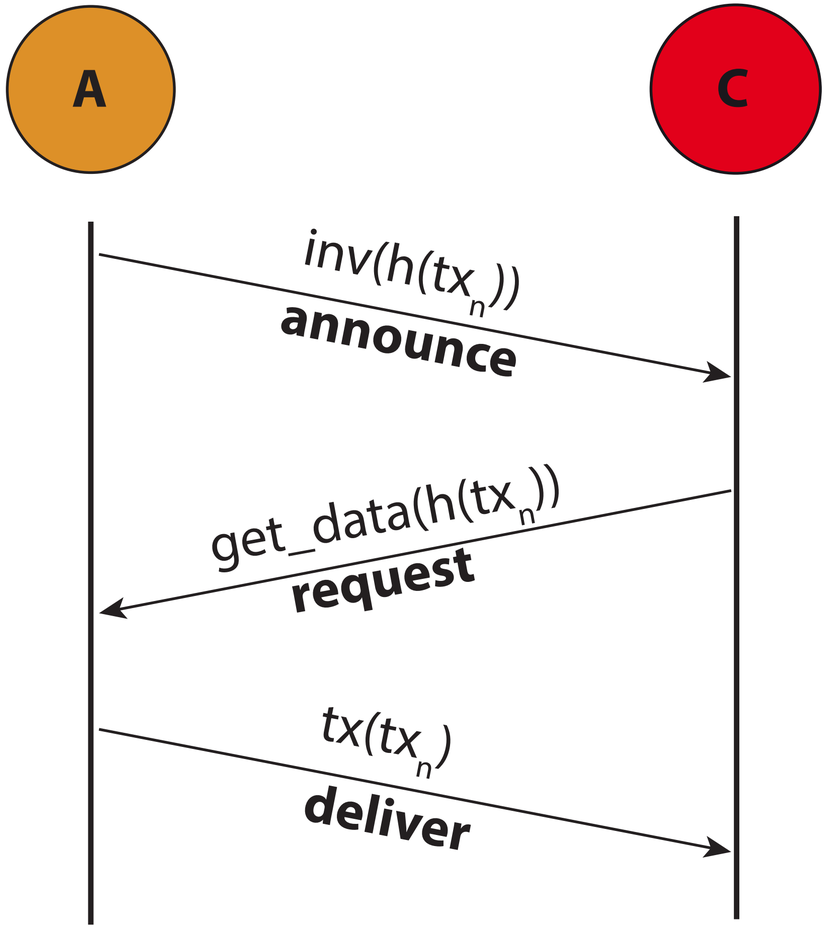
INFORMATION PROPAGATION (2/3)



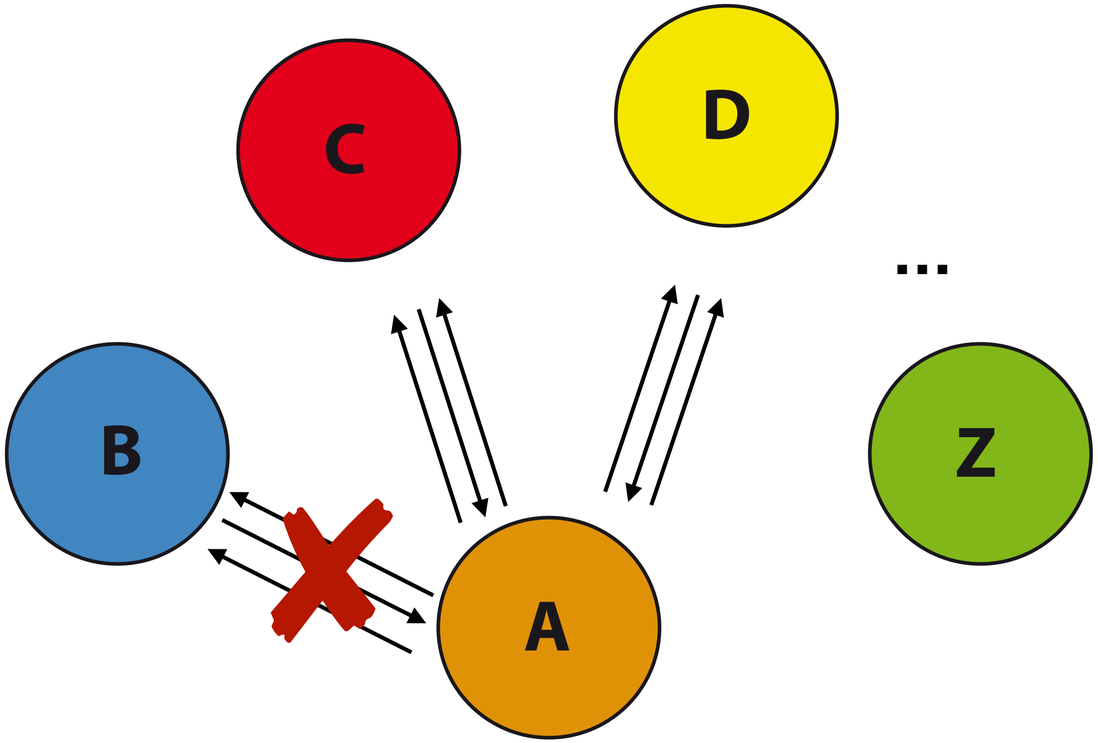


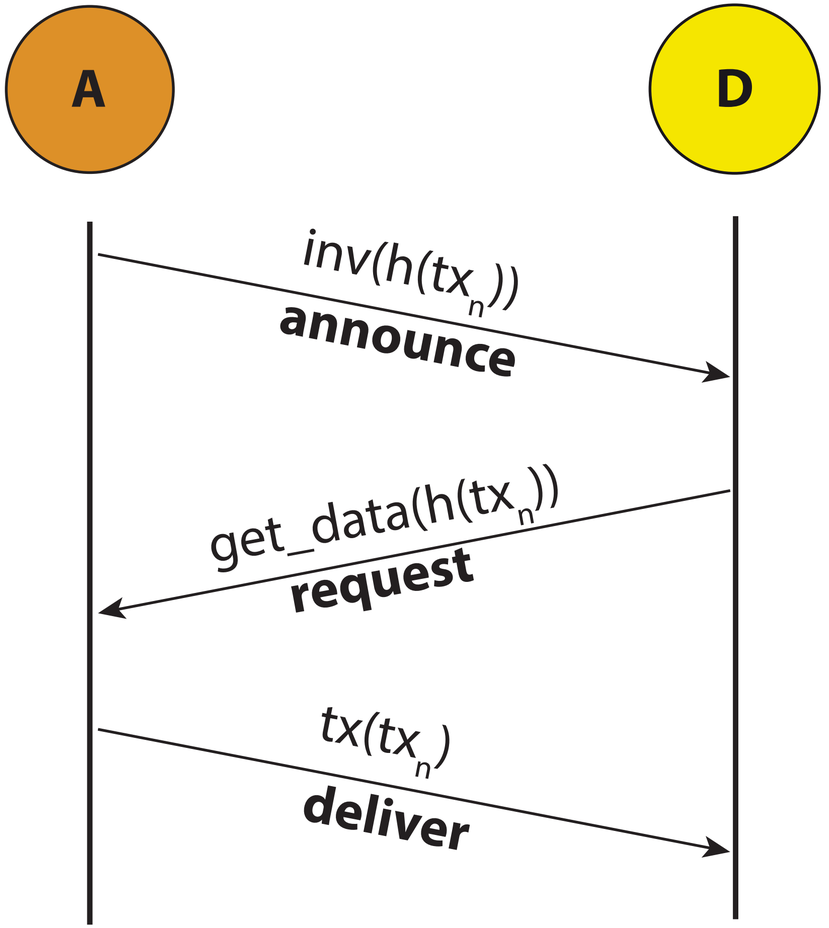
INFORMATION PROPAGATION (2/3)



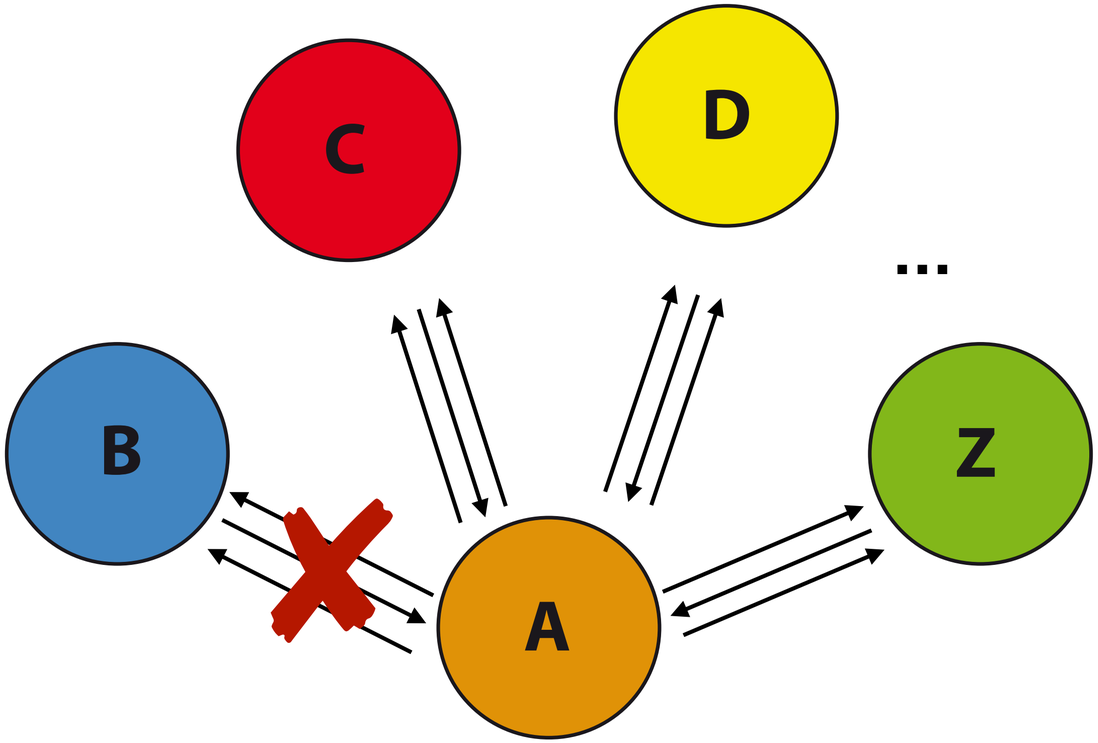


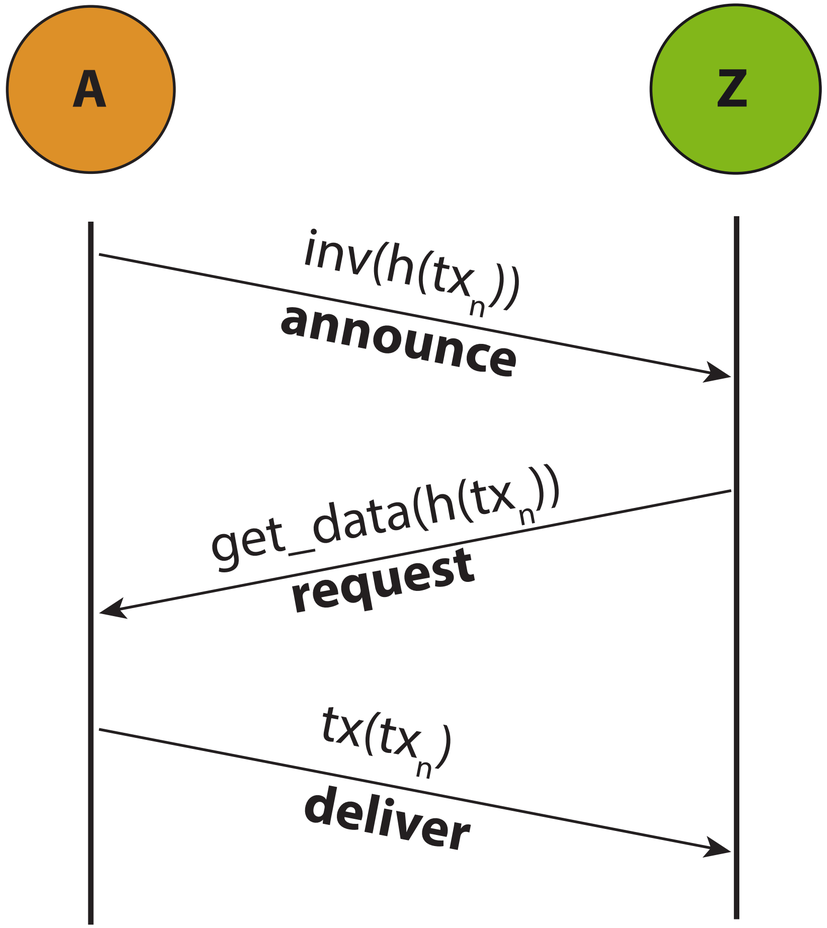
INFORMATION PROPAGATION (2/3)



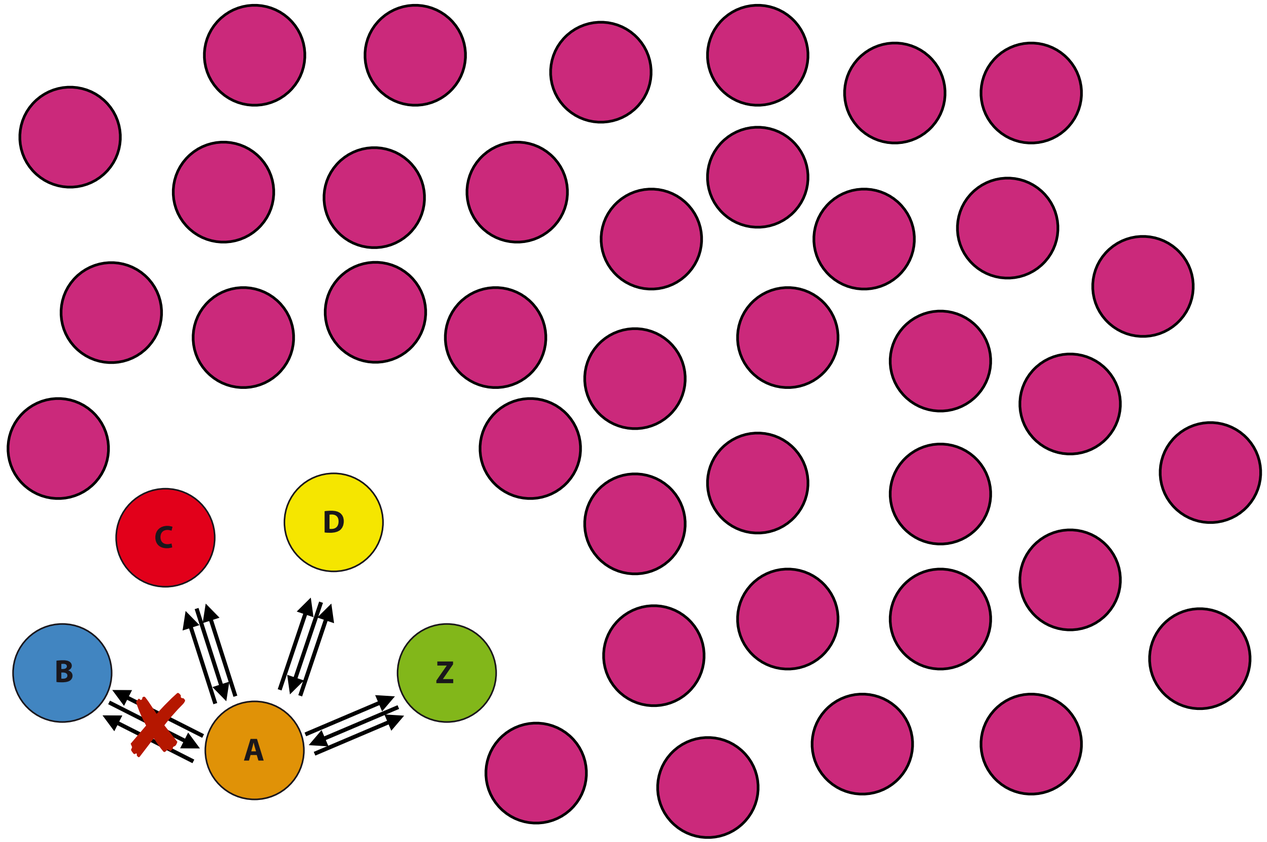


INFORMATION PROPAGATION (2/3)

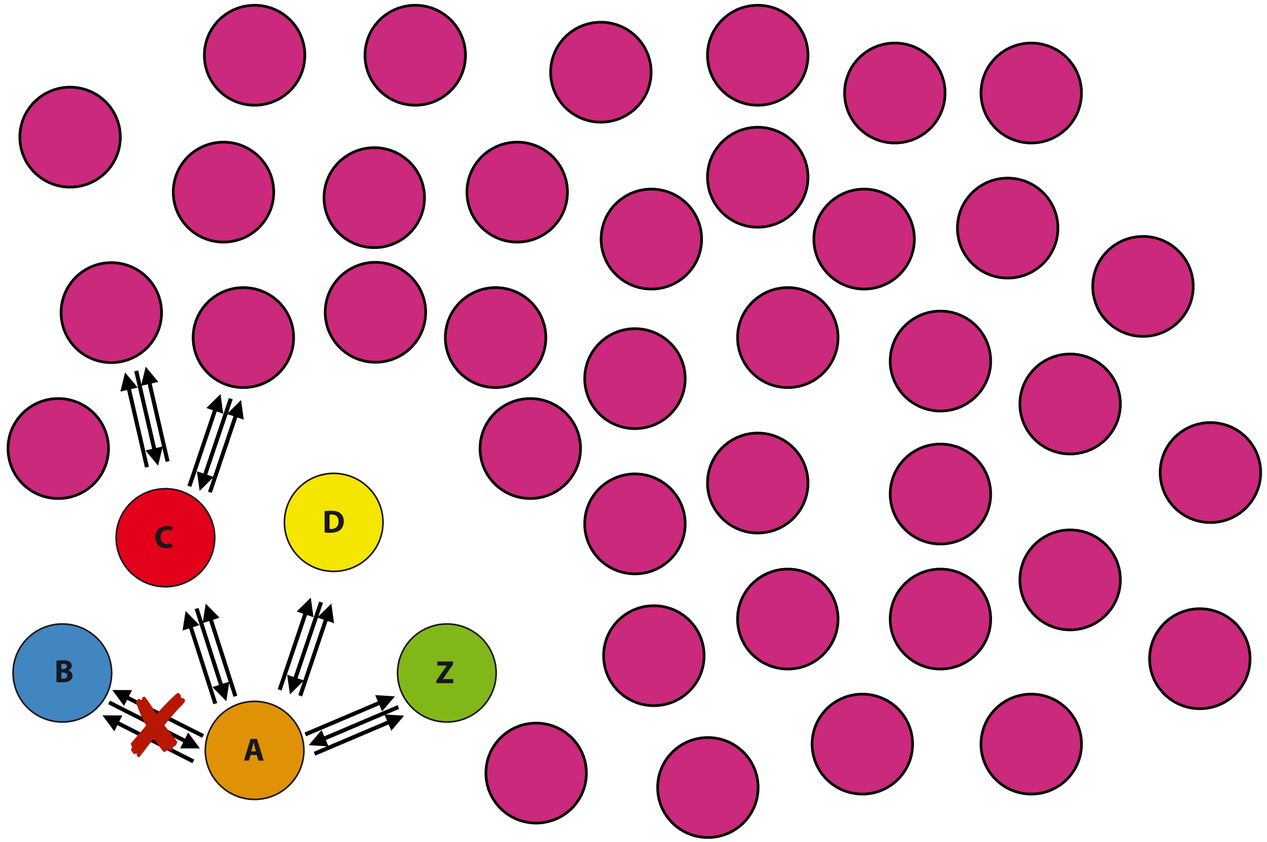




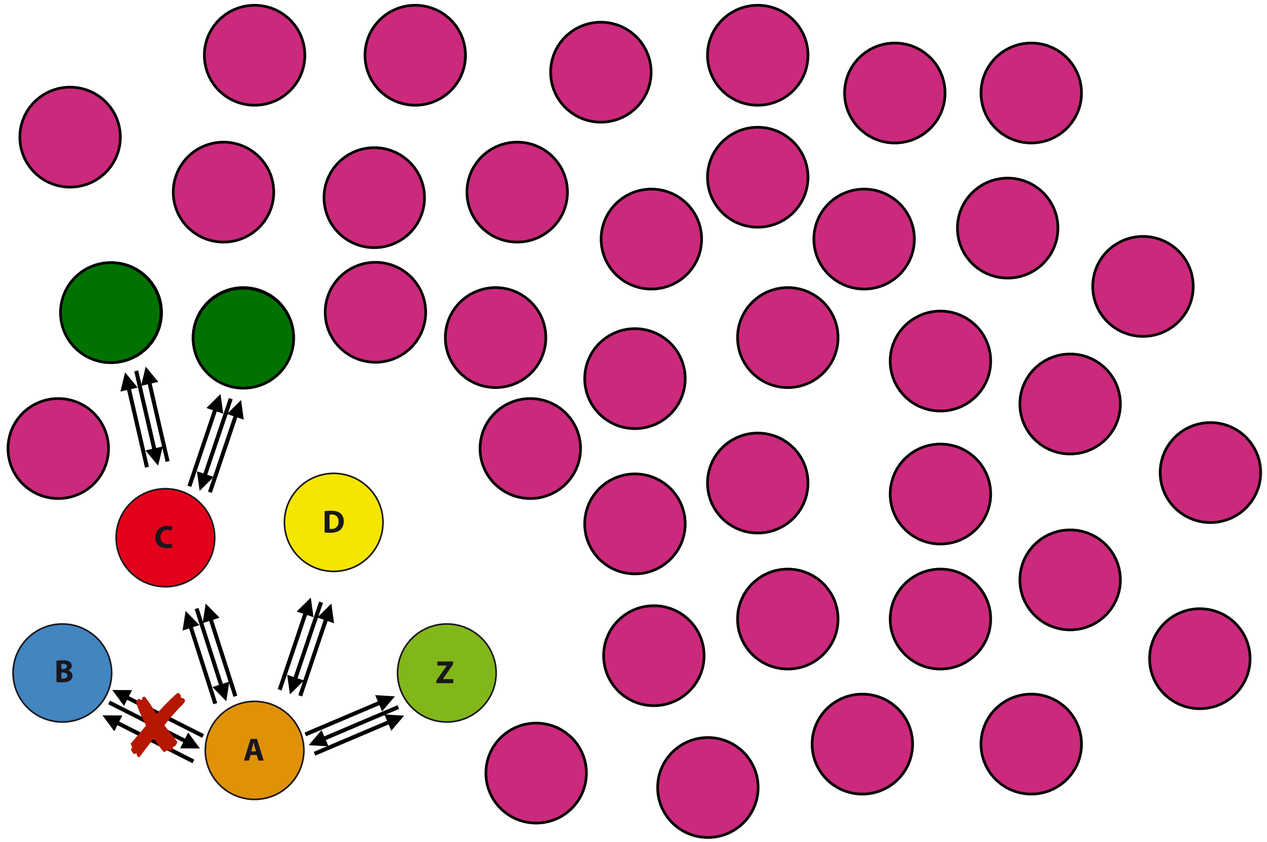
INFORMATION PROPAGATION (3/3)



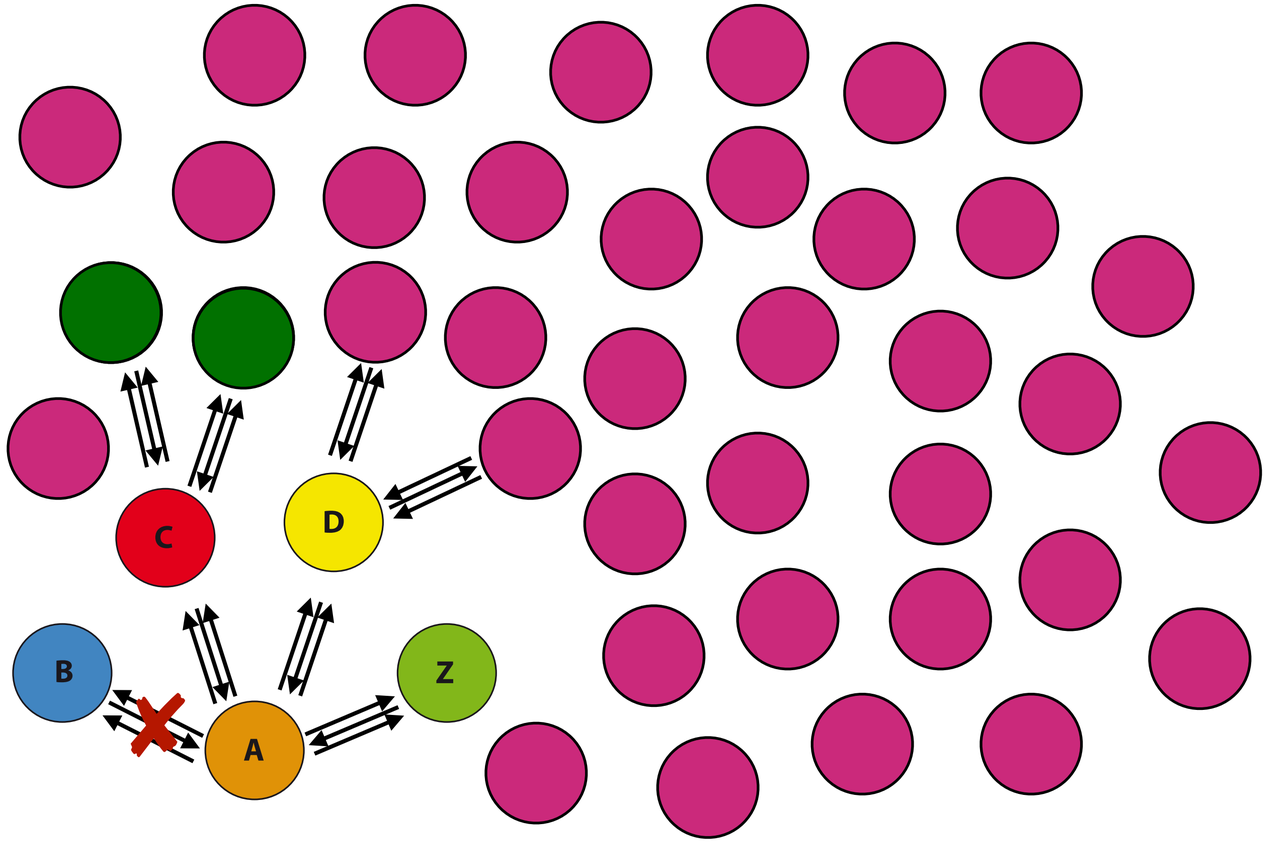
INFORMATION PROPAGATION (3/3)



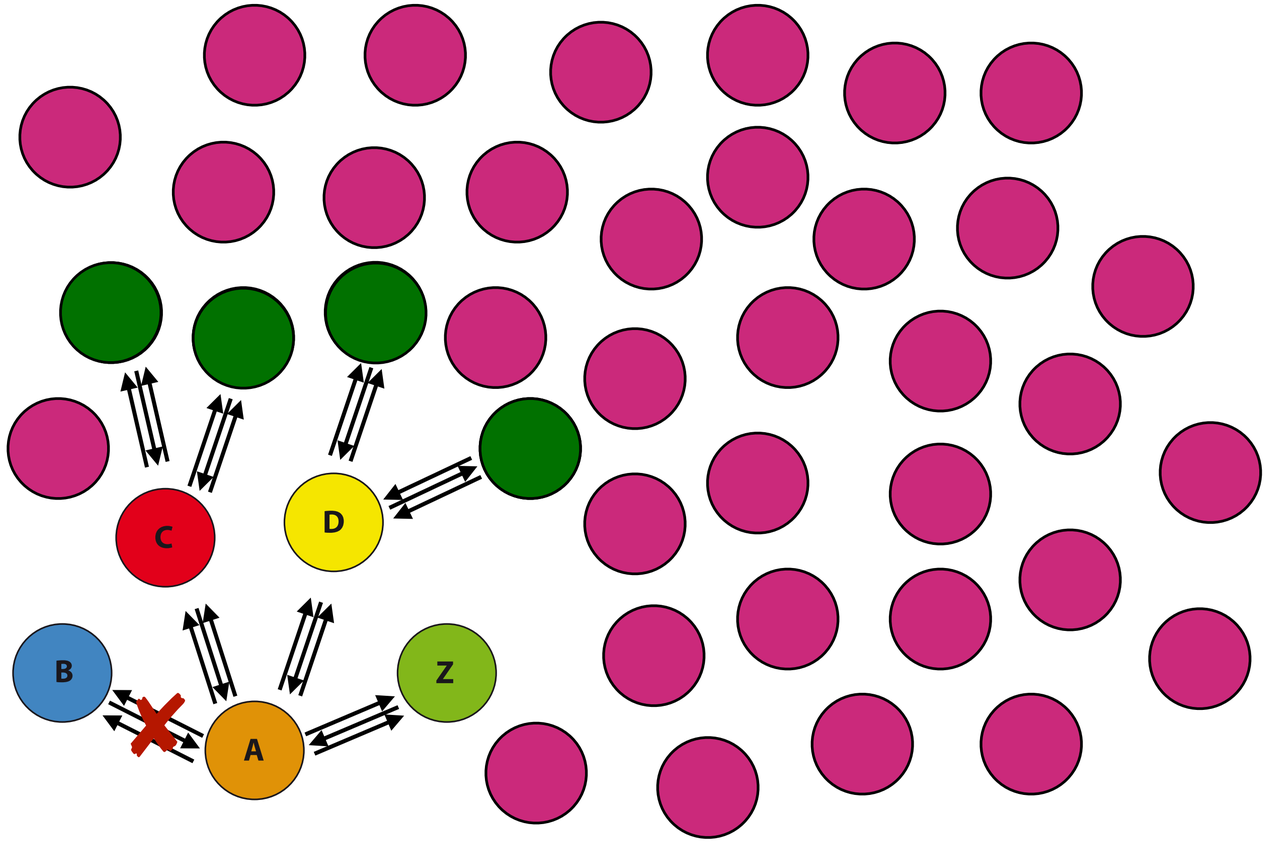
INFORMATION PROPAGATION (3/3)



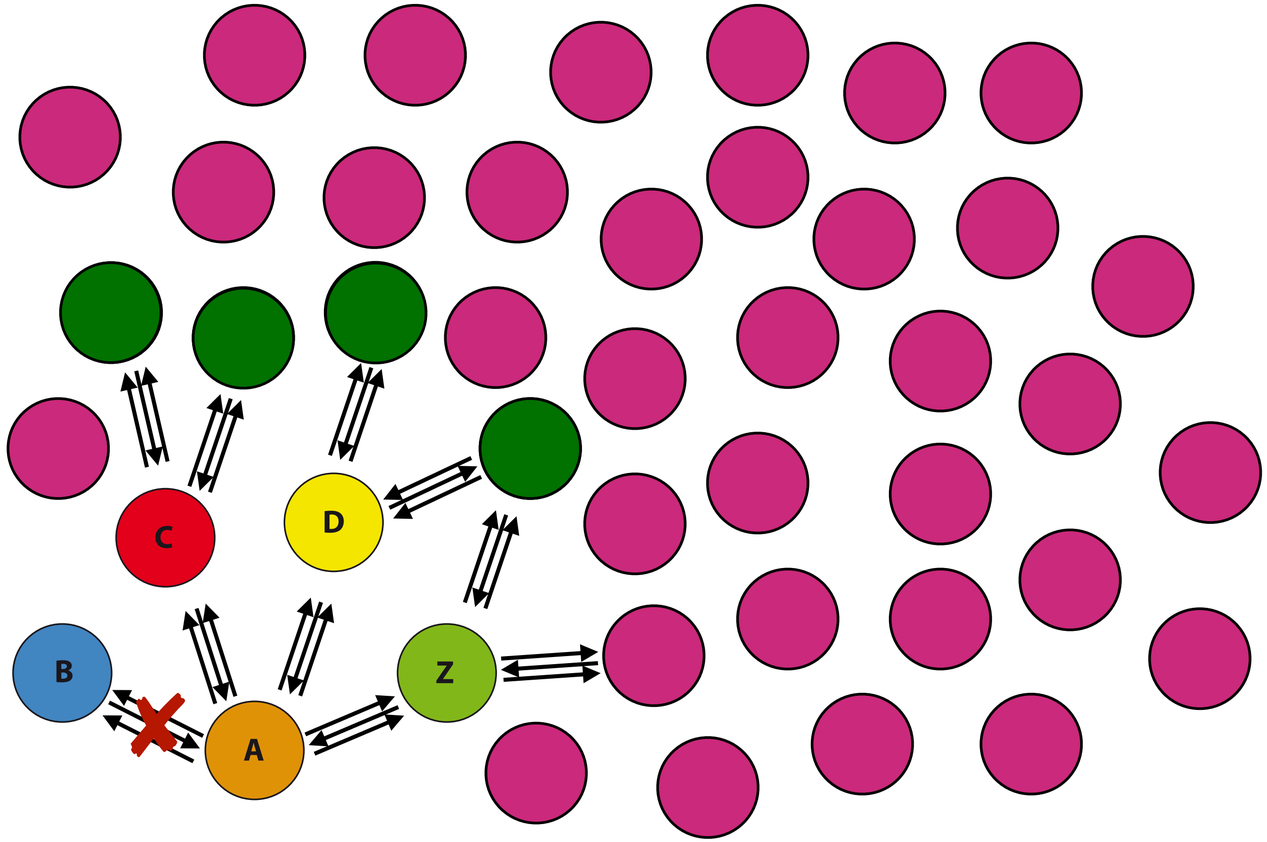
INFORMATION PROPAGATION (3/3)



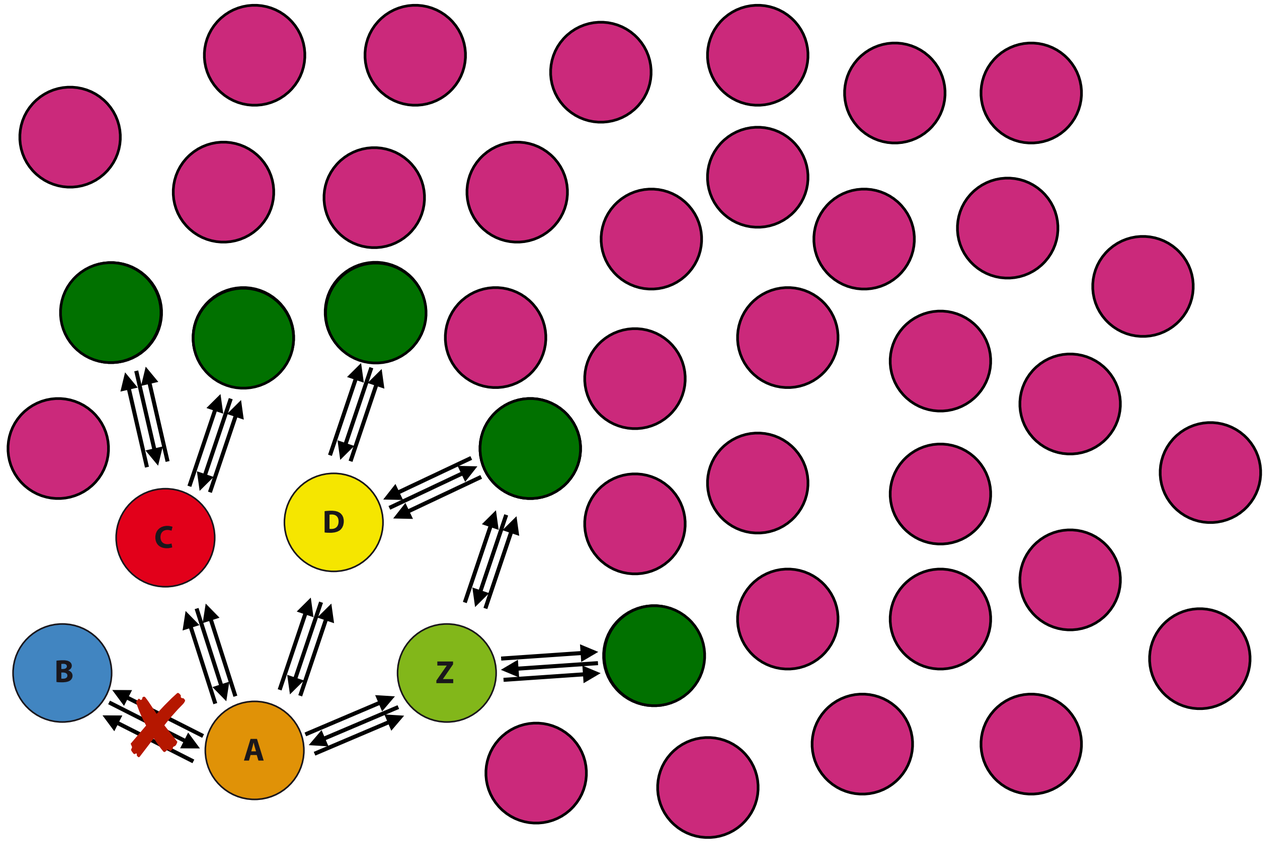
INFORMATION PROPAGATION (3/3)



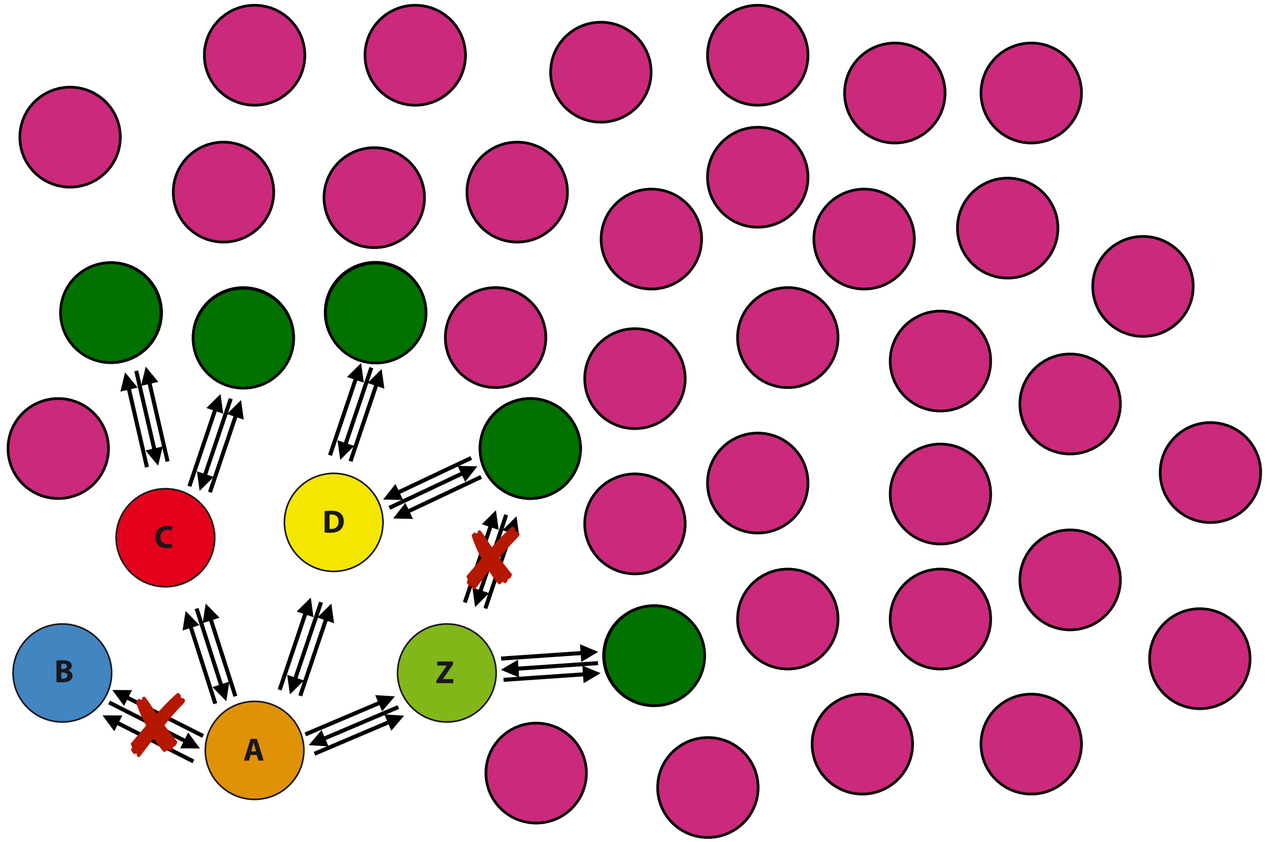
INFORMATION PROPAGATION (3/3)



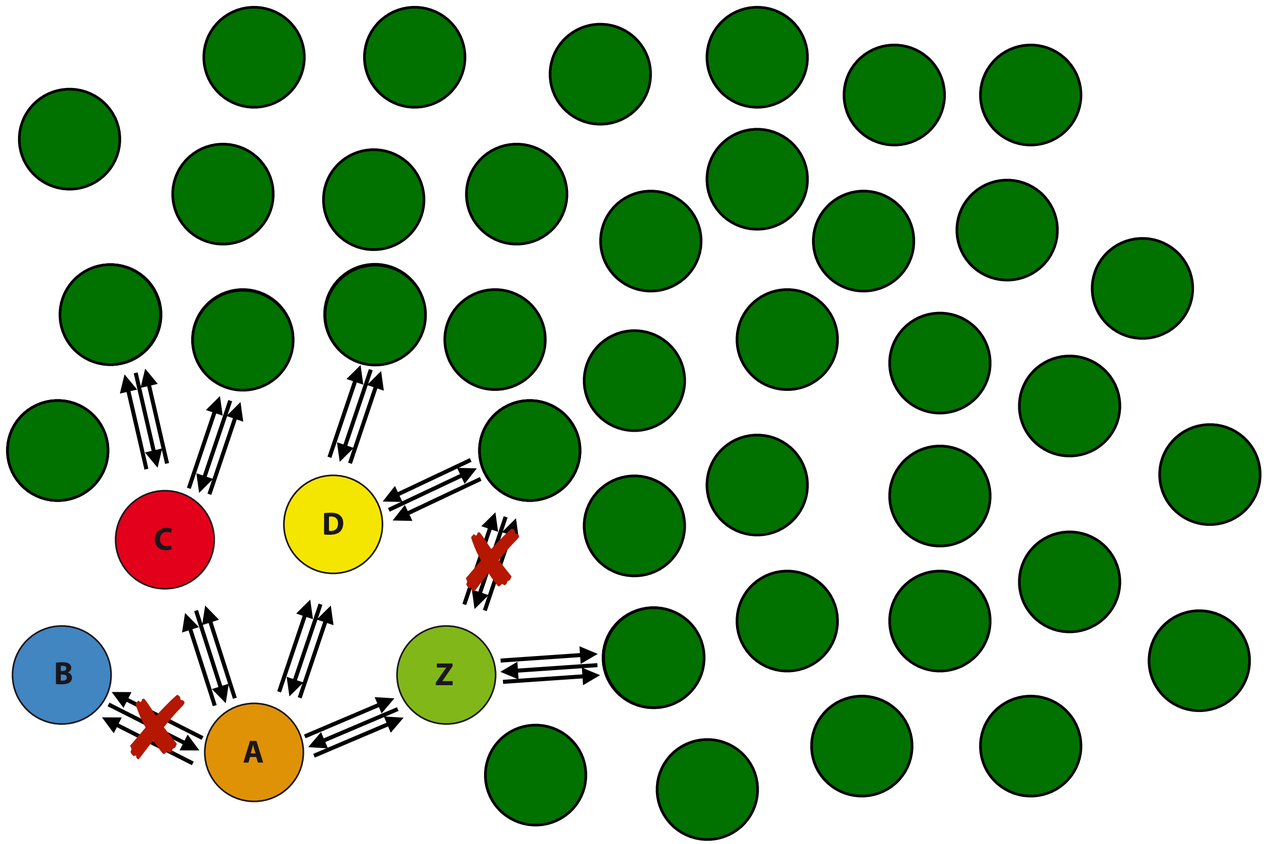
INFORMATION PROPAGATION (3/3)



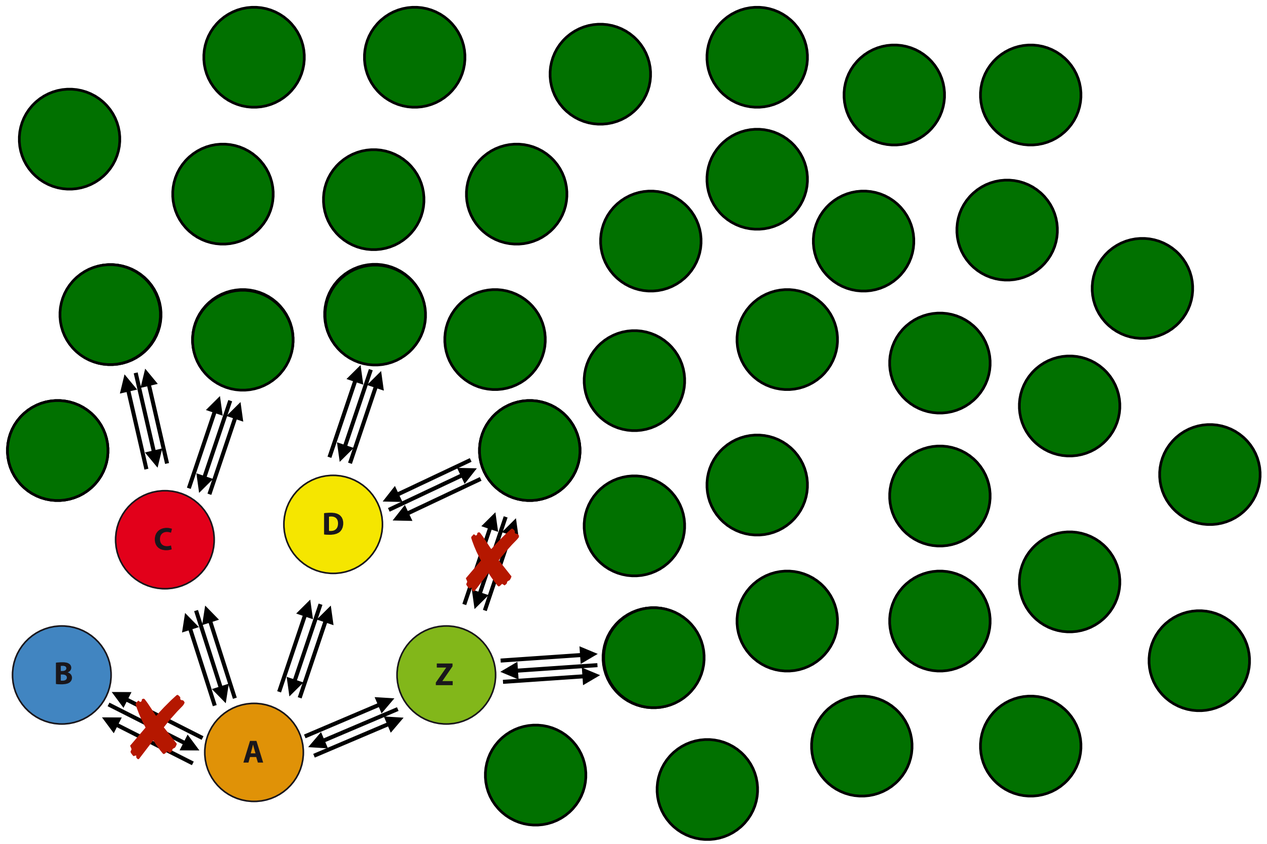
INFORMATION PROPAGATION (3/3)



INFORMATION PROPAGATION (3/3)

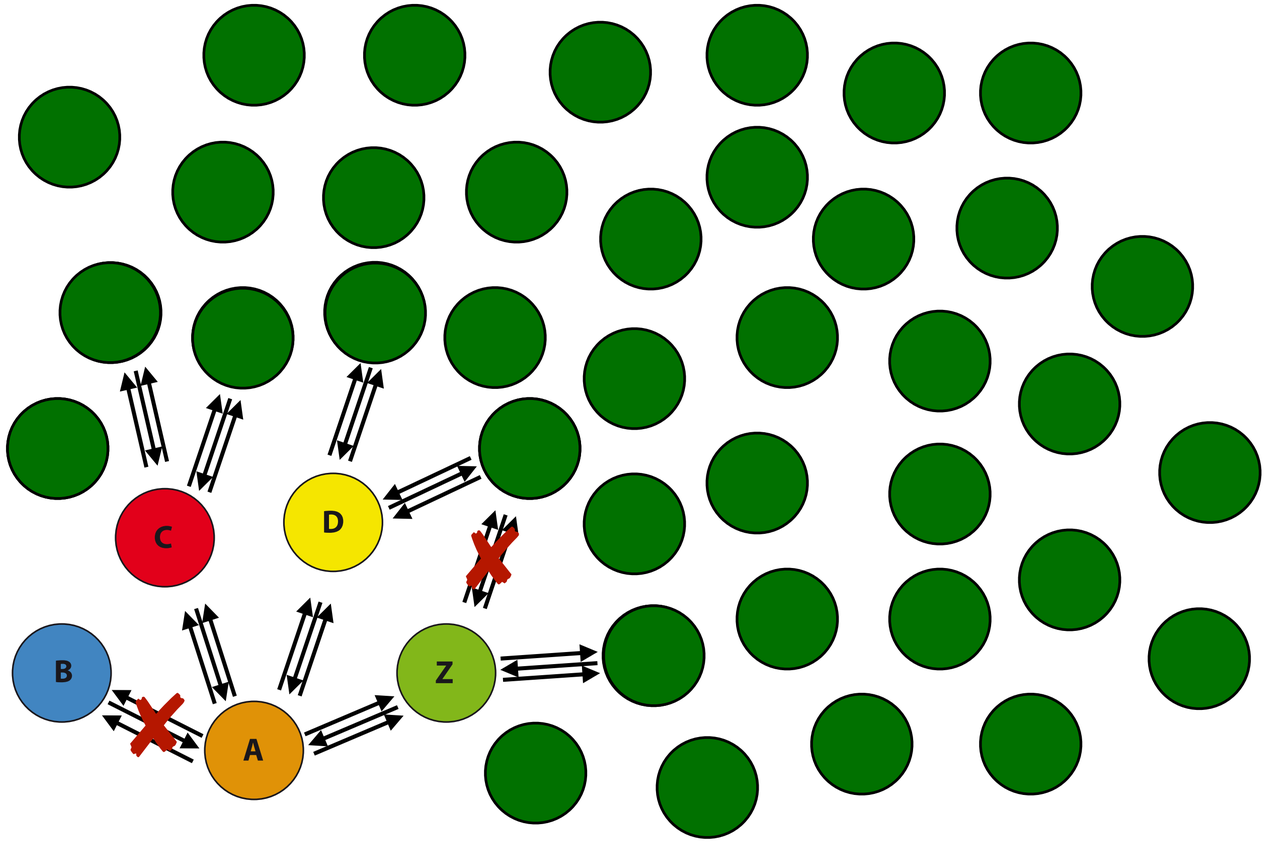


INFORMATION PROPAGATION (3/3)



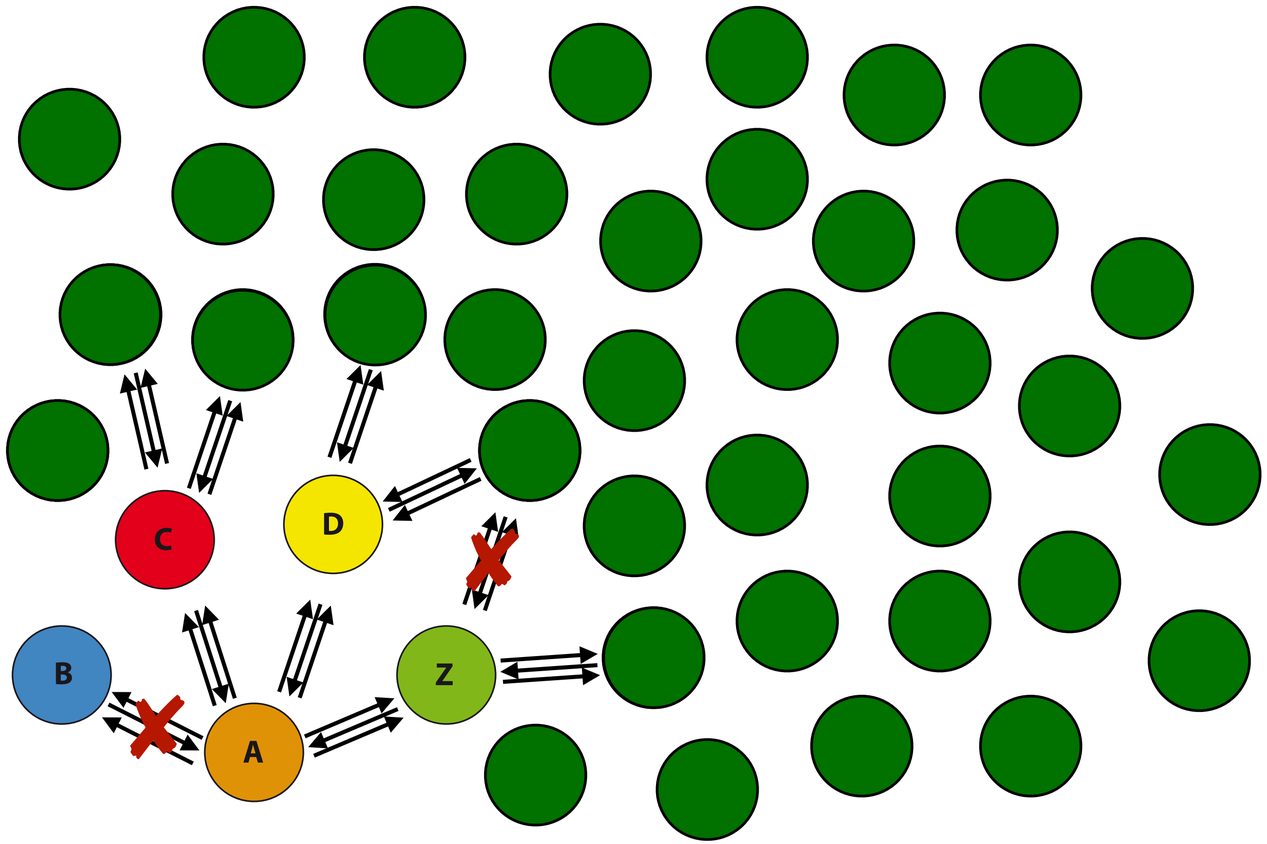
- And so on and so forth until all the nodes are reached

INFORMATION PROPAGATION (3/3)



- And so on and so forth until all the nodes are reached
- Recall that a node will reject a transaction if it has already learnt about it from any of its neighbors

INFORMATION PROPAGATION (3/3)



- And so on and so forth until all the nodes are reached
- Recall that a node will reject a transaction if it has already learnt about it from any of its neighbors
- The same procedure applies for blocks

IMPLICATIONS

The bigger the network the more it takes for an item to propagate (**this can be counterintuitive**)

Long propagation times (**for blocks**) imply bigger likelihood of forking the blockchain

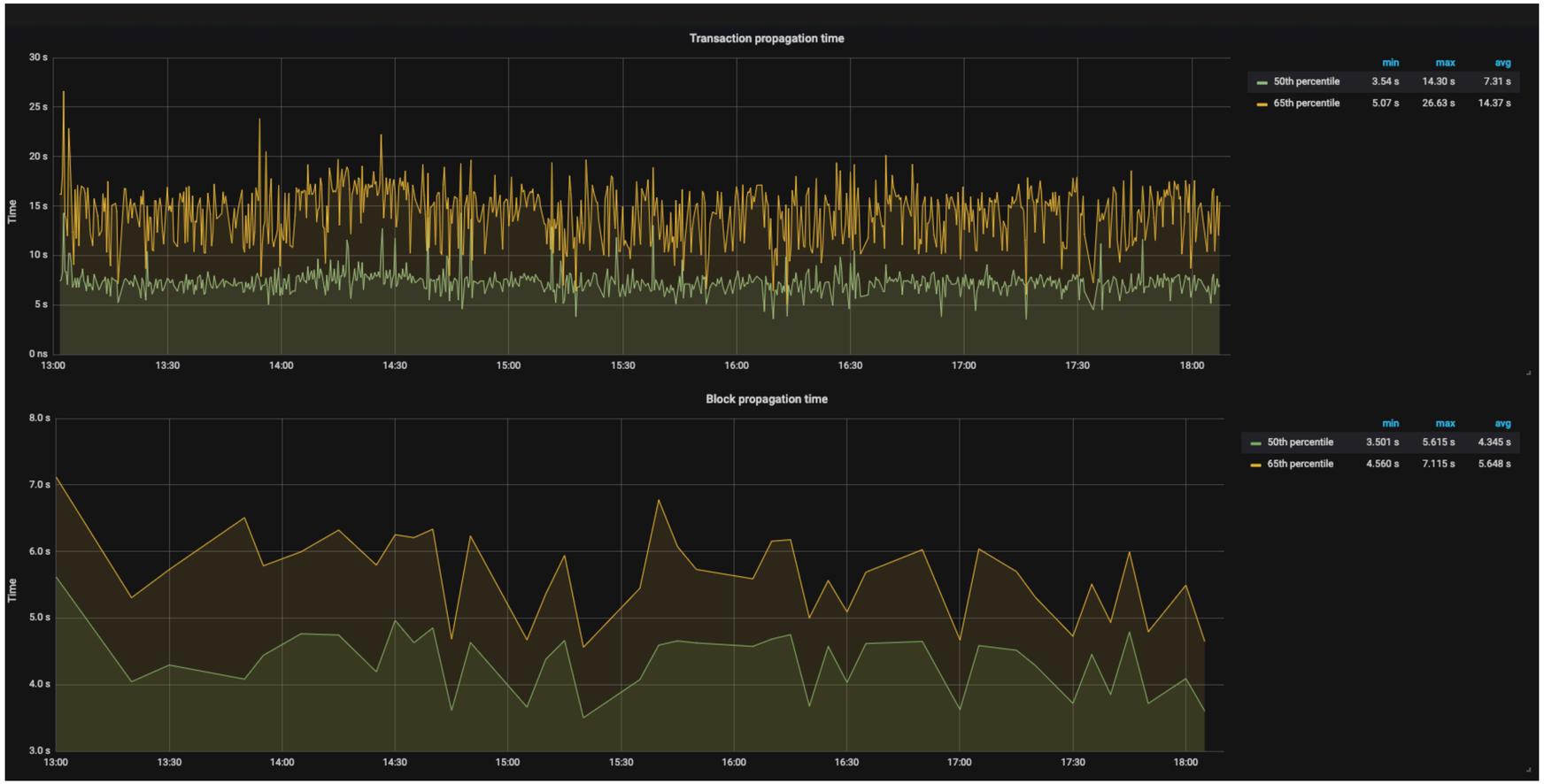
IMPLICATIONS

The bigger the network the more it takes for an item to propagate (**this can be counterintuitive**)

Long propagation times (**for blocks**) imply bigger likelihood of forking the blockchain

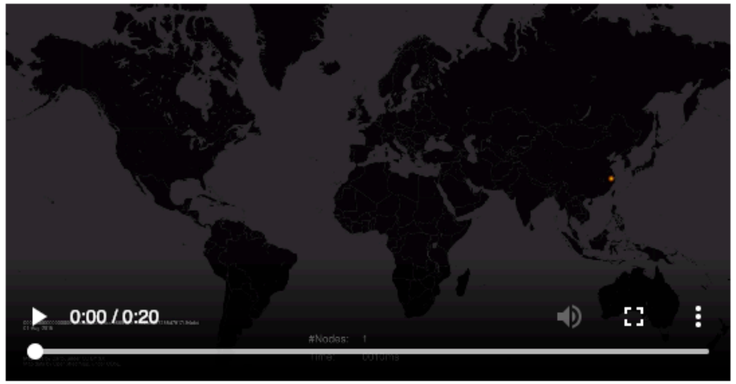
Christian Decker and Roger Wattenhofer
Information propagation in the Bitcoin network
https://ieeexplore.ieee.org/document/6688704

DATA PROPAGATION TIMES (TESTNET)

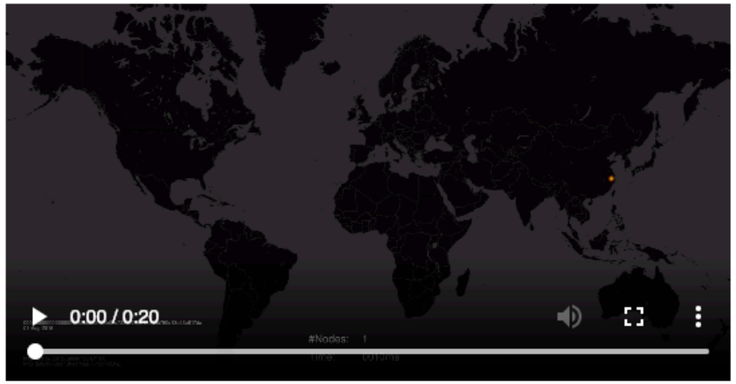


source: charts.satoshi.uab.cat

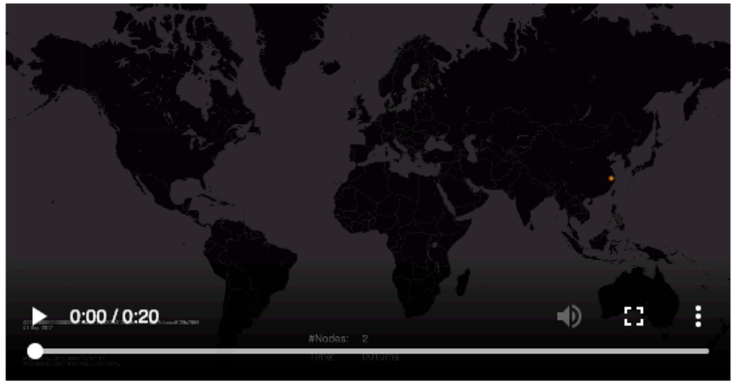
MORE ABOUT PROPAGATION TIMES



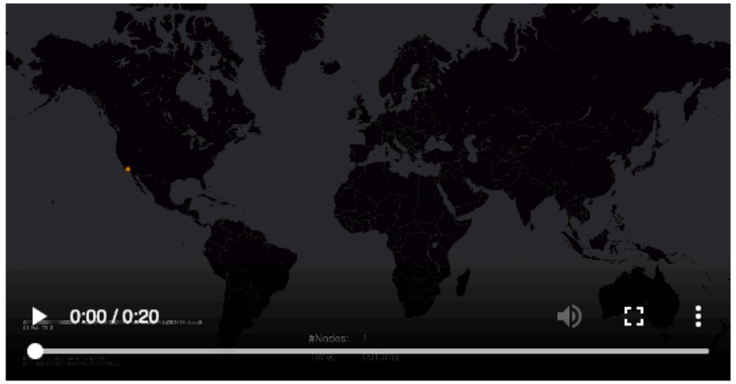
Block propagation | 01.08.2015



Block propagation | 01.08.2016



Block propagation | 01.09.2017



Block propagation | 01.02.2018

source: https://dsn.tm.kit.edu/bitcoin/videos.html

PROPAGATION DELAYS (1/2)

How can blocks propagate faster than transactions if the former are bigger than the later?

PROPAGATION DELAYS (1/2)

How can blocks propagate faster than transactions if the former are bigger than the later?

- Transactions are accumulated in buffers and forwarded in batches to break the link between first relayer and origin of a transaction

PROPAGATION DELAYS (1/2)

How can blocks propagate faster than transactions if the former are bigger than the later?

- Transactions are accumulated in buffers and forwarded in batches to break the link between first relayer and origin of a transaction
- The propagation of blocks is not delayed, in order to reach full network coverage as soon as possible

PROPAGATION DELAYS (2/2)

But blocks are way bigger than transactions, how can they be propagated so fast!?

PROPAGATION DELAYS (2/2)

But blocks are way bigger than transactions, how can they be propagated so fast!?

- Fast relay networks on top of Bitcoin exists (Falcon, FIBRE, etc) to enhance the propagation time of blocks

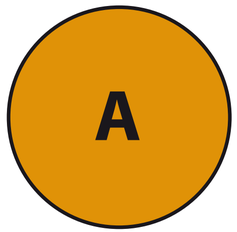
PROPAGATION DELAYS (2/2)

But blocks are way bigger than transactions, how can they be propagated so fast!?

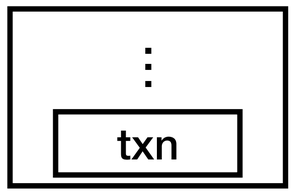
- Fast relay networks on top of Bitcoin exists (Falcon, FIBRE, etc) to enhance the propagation time of blocks
- Miners use such networks to ensure minimal propagation times as well as ensure being mining on top of the most recent block

0-conf transactions and double-spending

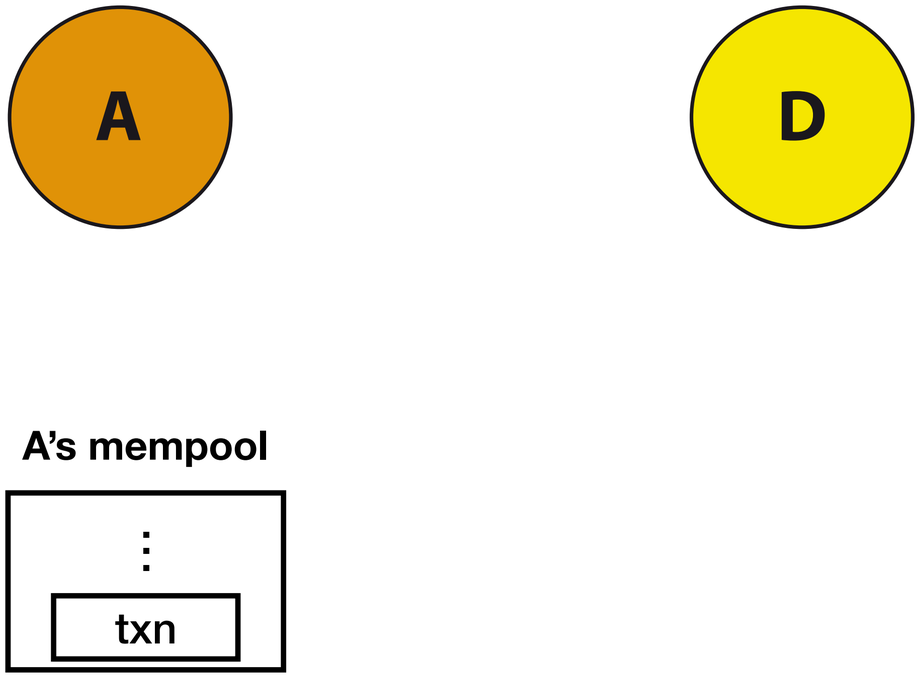
CONFIRMED TRANSACTIONS



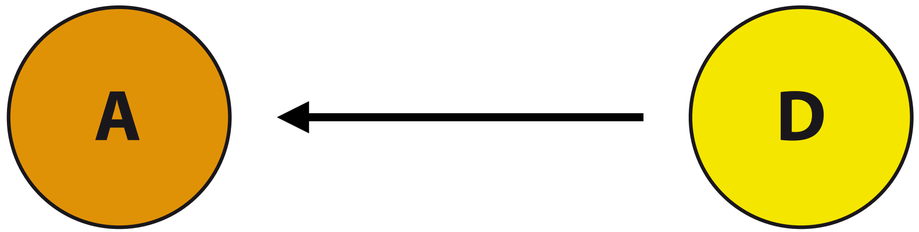
A's mempool

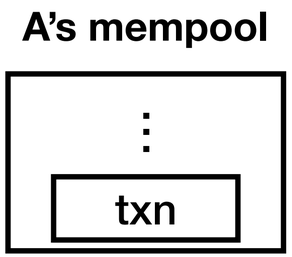


CONFIRMED TRANSACTIONS

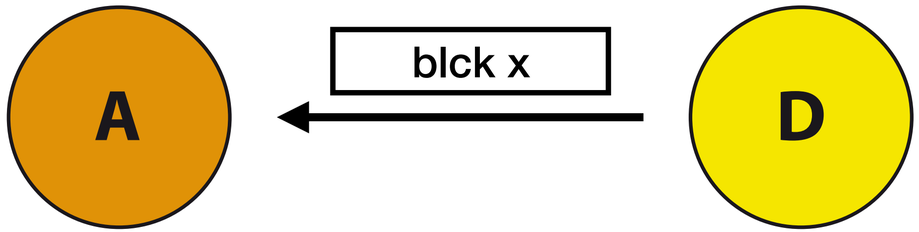


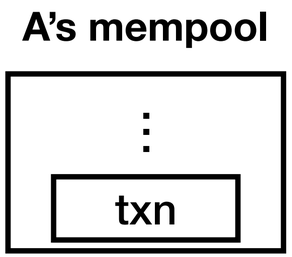
CONFIRMED TRANSACTIONS



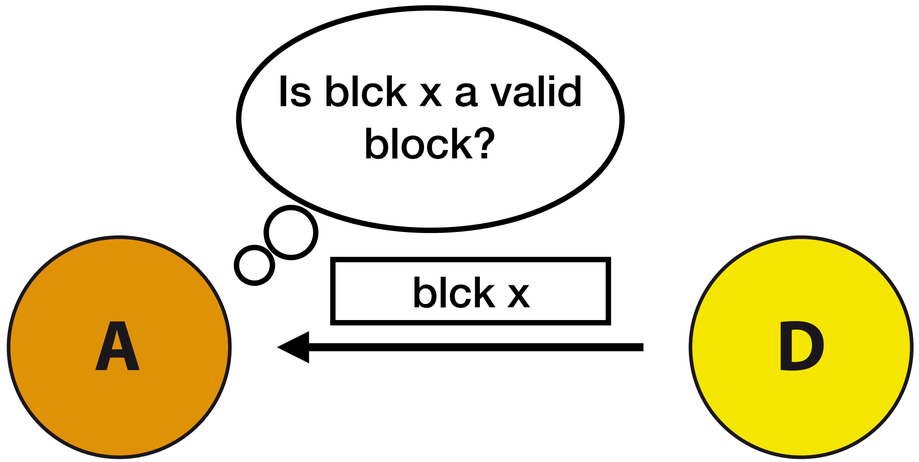


CONFIRMED TRANSACTIONS

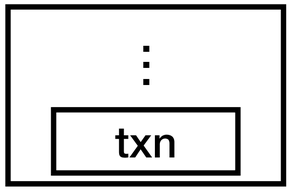




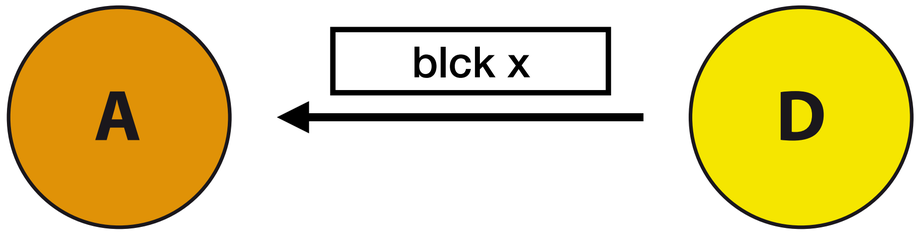
CONFIRMED TRANSACTIONS

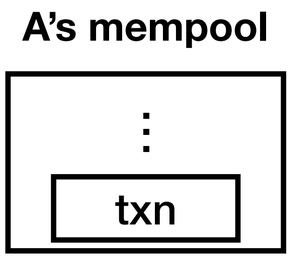


A's mempool

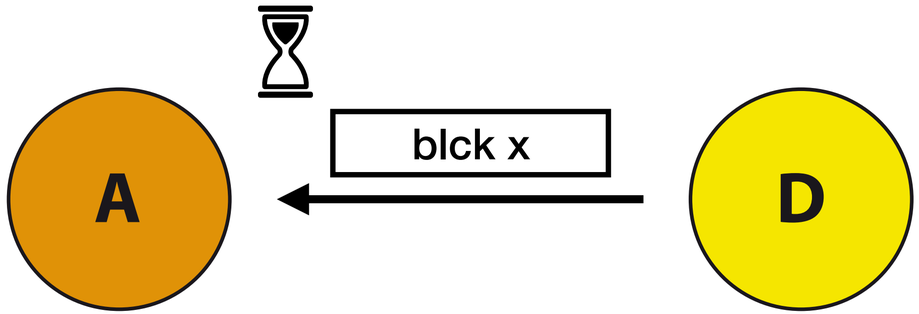


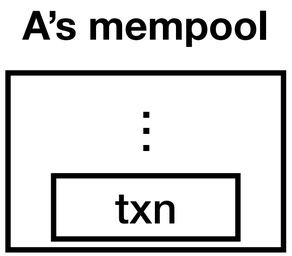
CONFIRMED TRANSACTIONS



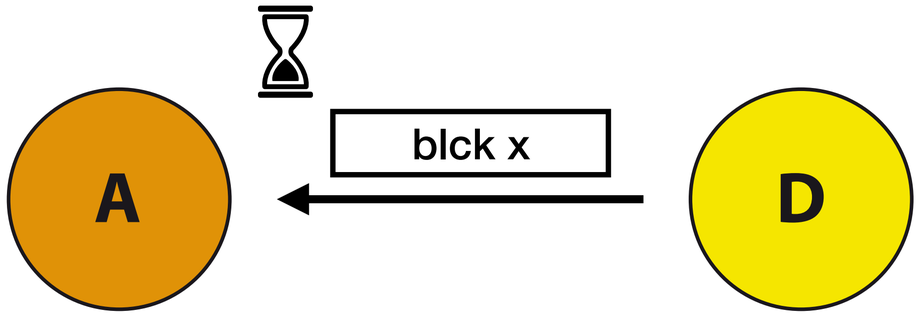


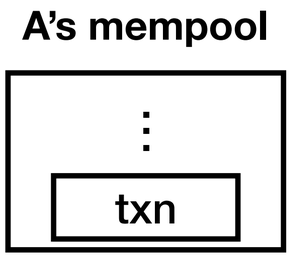
CONFIRMED TRANSACTIONS



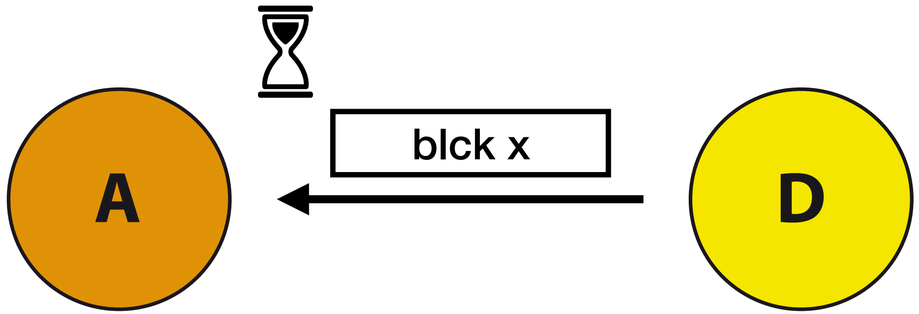


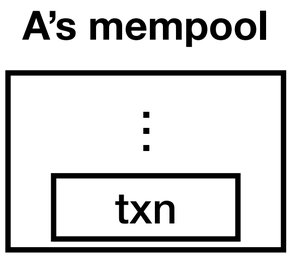
CONFIRMED TRANSACTIONS



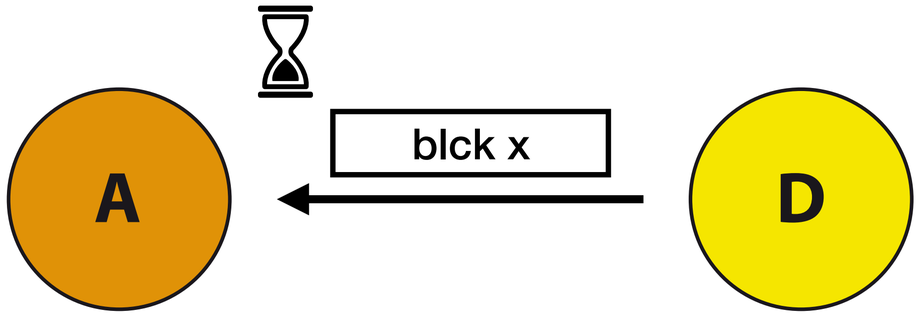


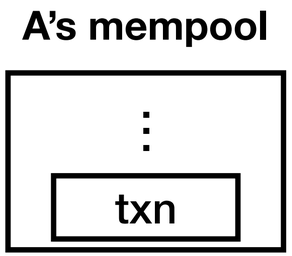
CONFIRMED TRANSACTIONS



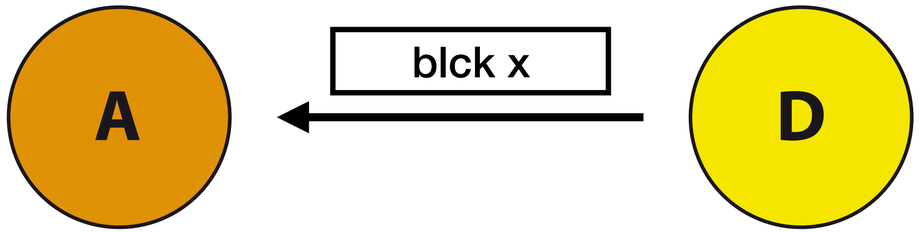


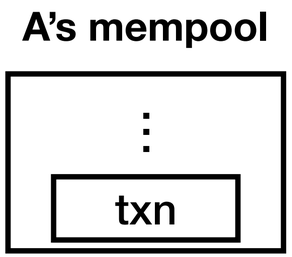
CONFIRMED TRANSACTIONS



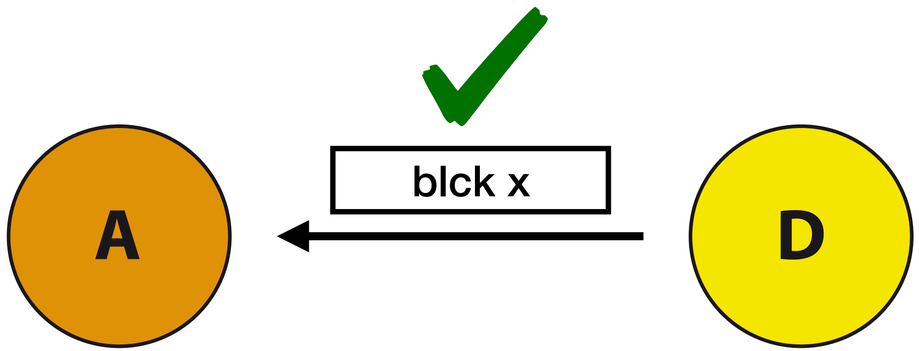


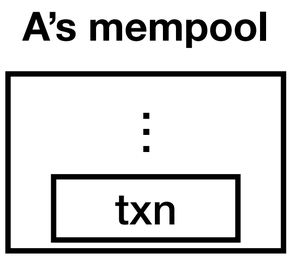
CONFIRMED TRANSACTIONS



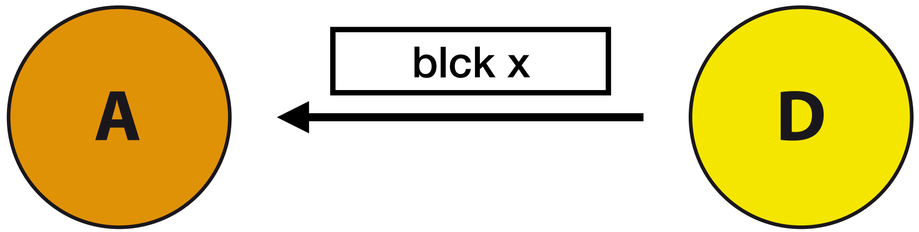


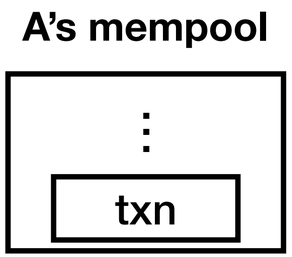
CONFIRMED TRANSACTIONS



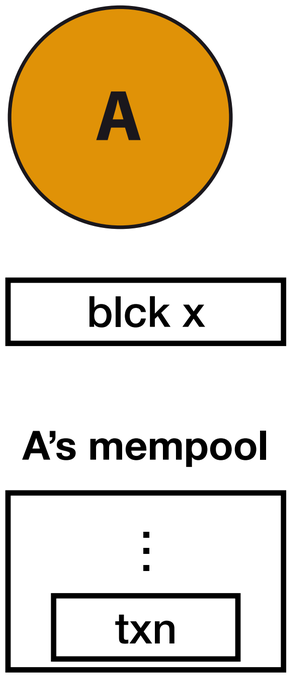


CONFIRMED TRANSACTIONS

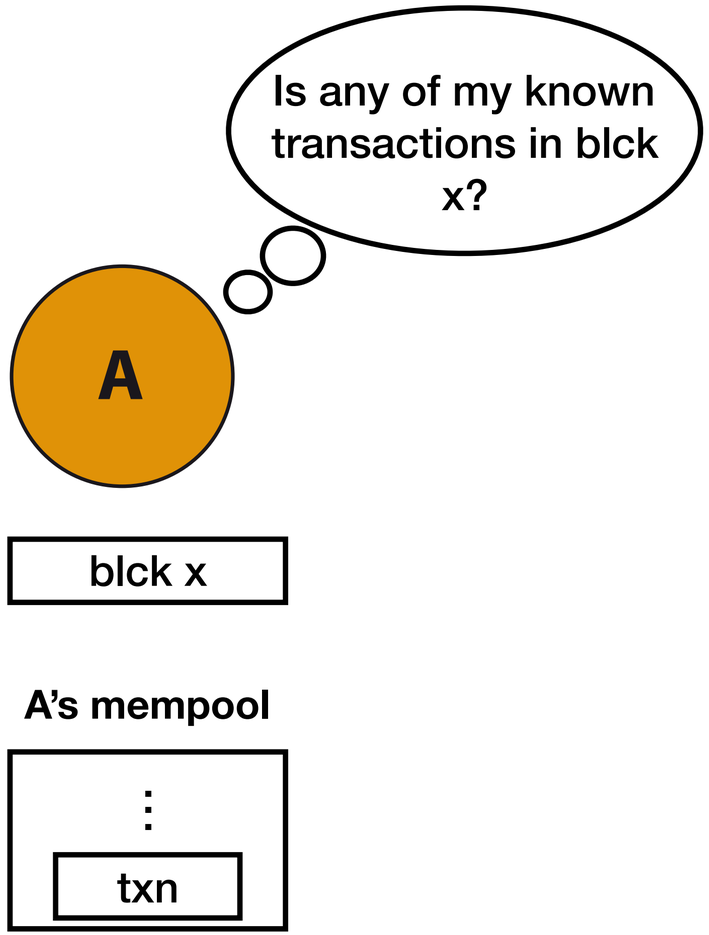




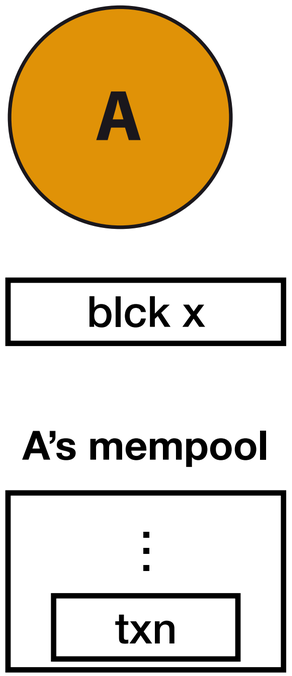
CONFIRMED TRANSACTIONS



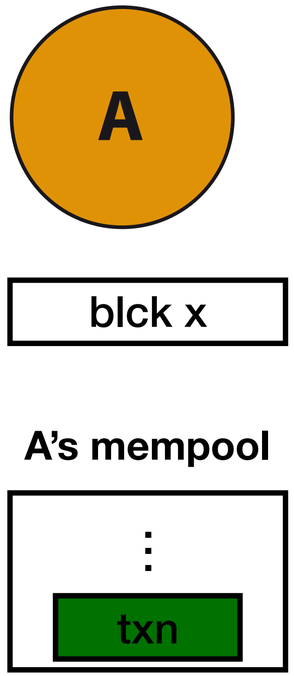
CONFIRMED TRANSACTIONS



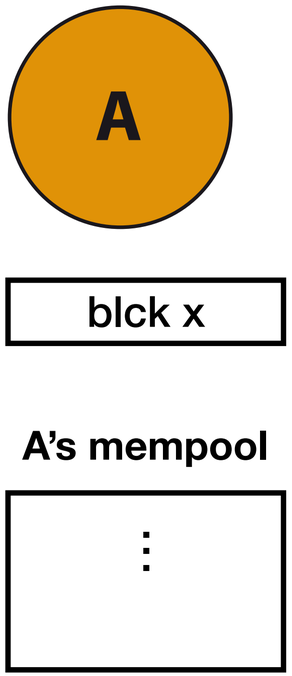
CONFIRMED TRANSACTIONS



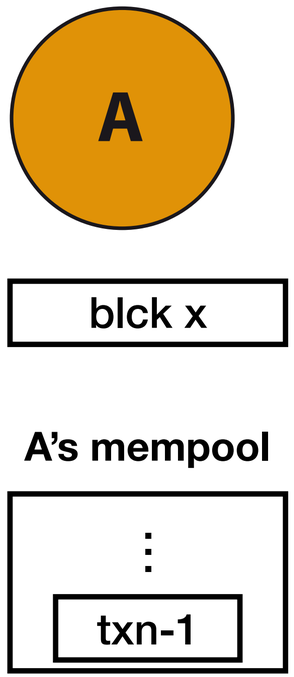
CONFIRMED TRANSACTIONS



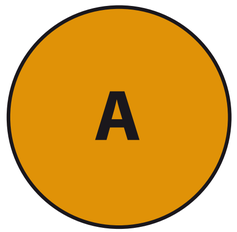
CONFIRMED TRANSACTIONS



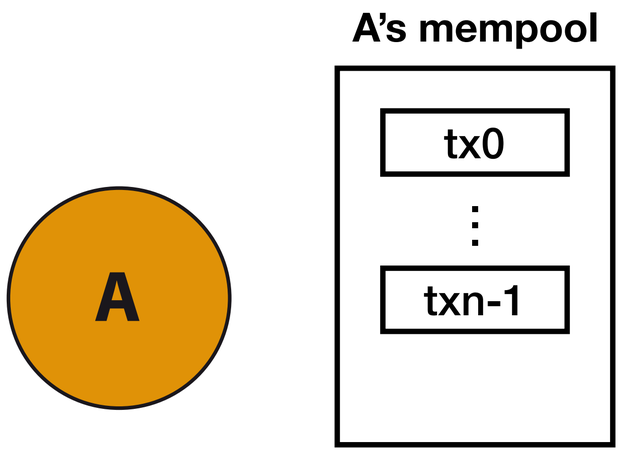
CONFIRMED TRANSACTIONS



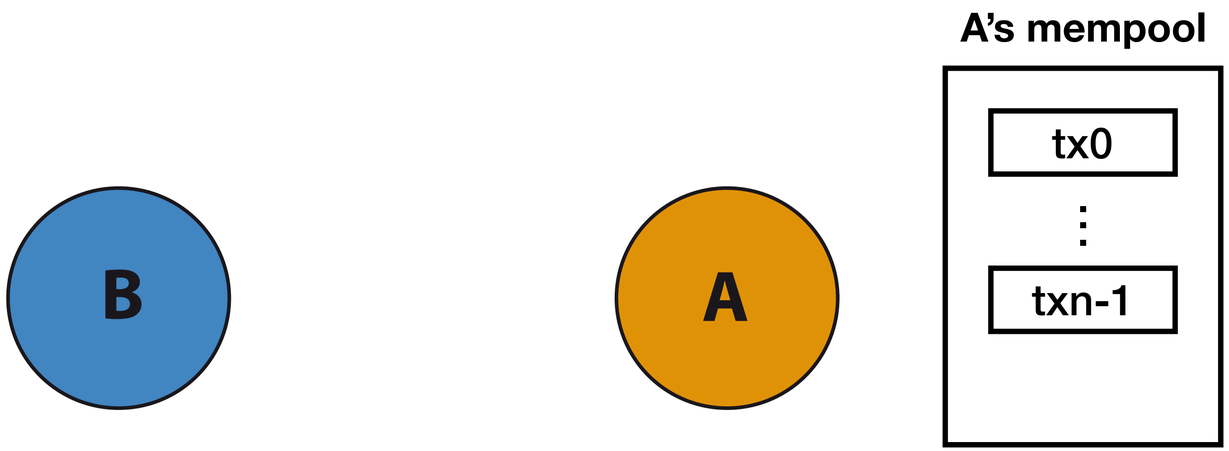
0-CONFIRMATION TRANSACTIONS (1/2)



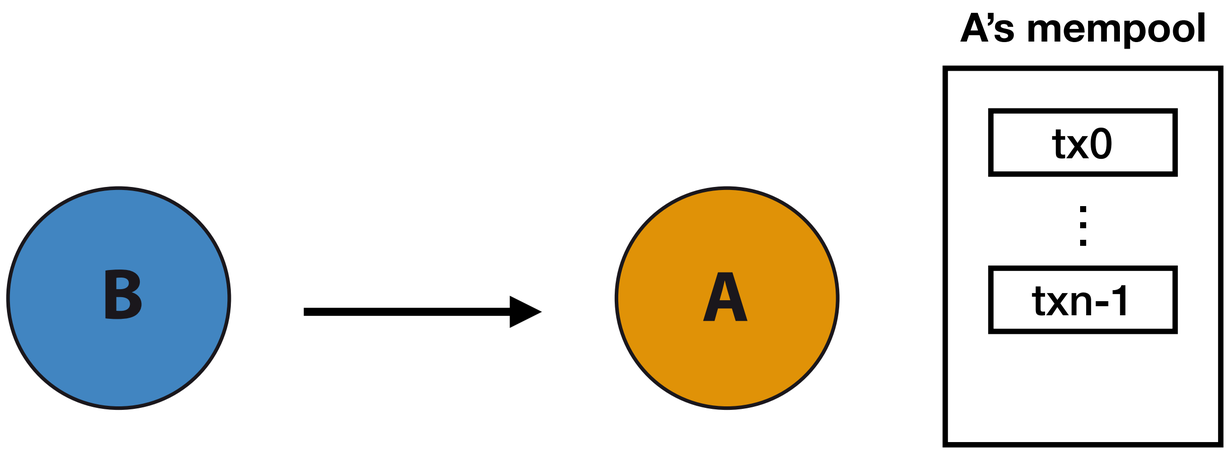
0-CONFIRMATION TRANSACTIONS (1/2)



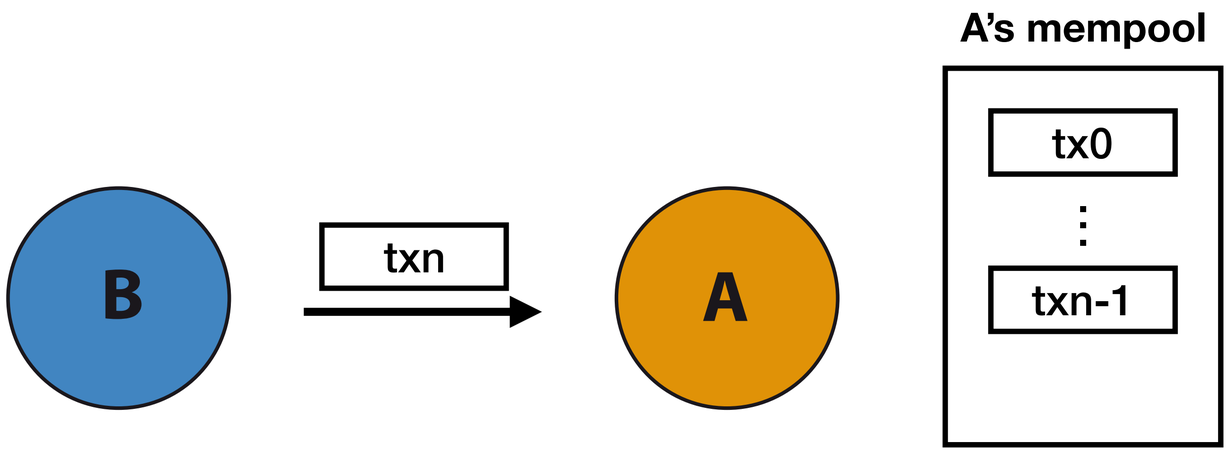
0-CONFIRMATION TRANSACTIONS (1/2)



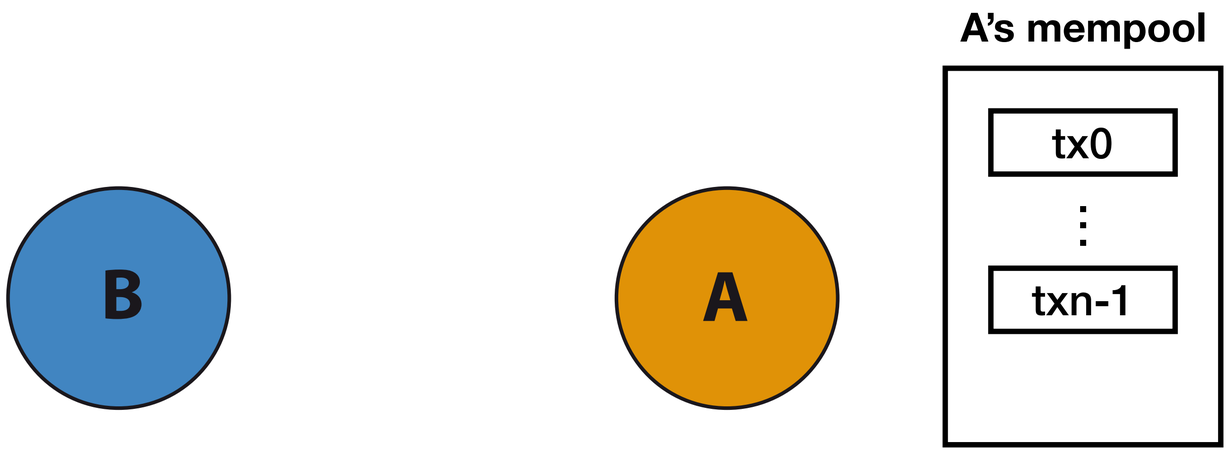
0-CONFIRMATION TRANSACTIONS (1/2)



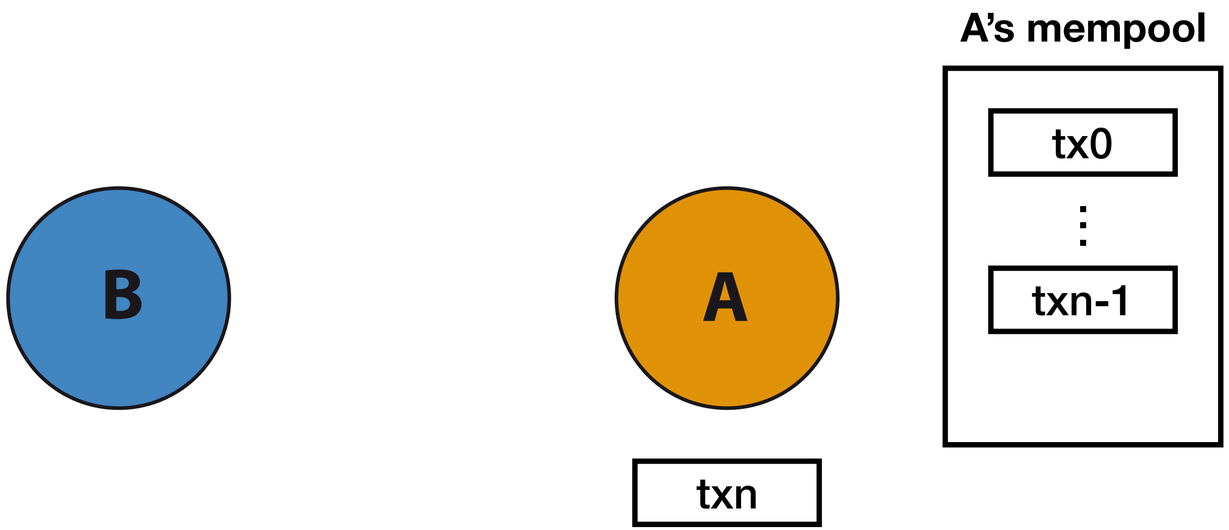
0-CONFIRMATION TRANSACTIONS (1/2)



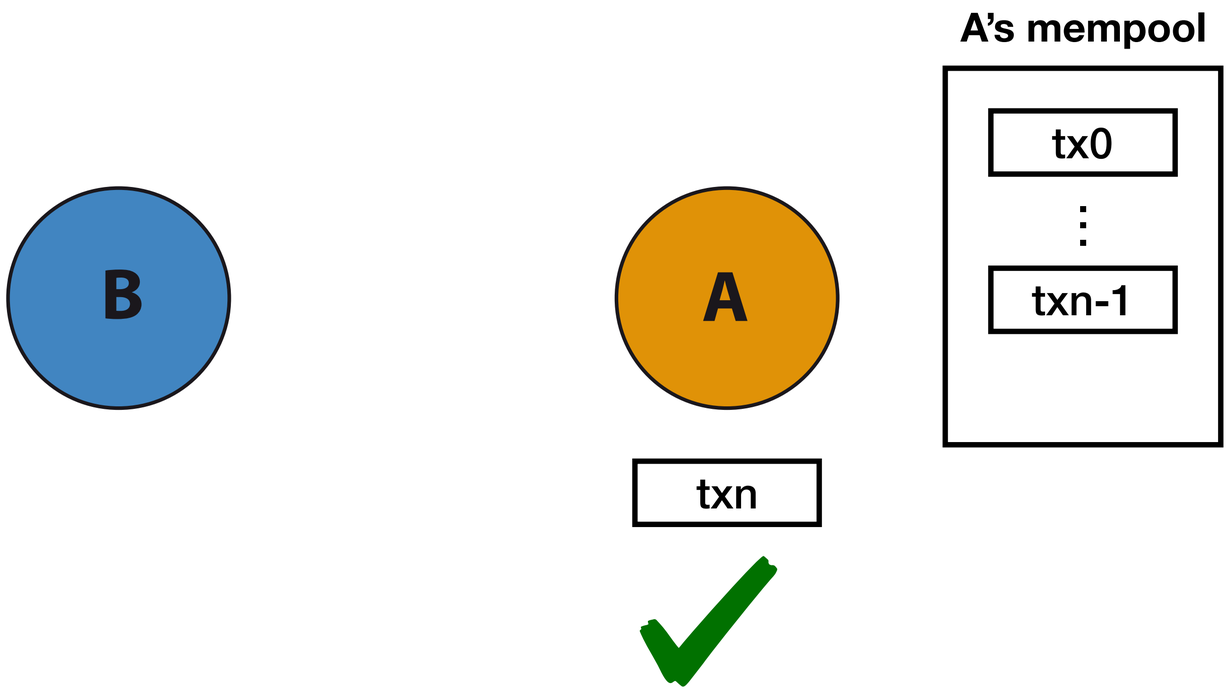
0-CONFIRMATION TRANSACTIONS (1/2)



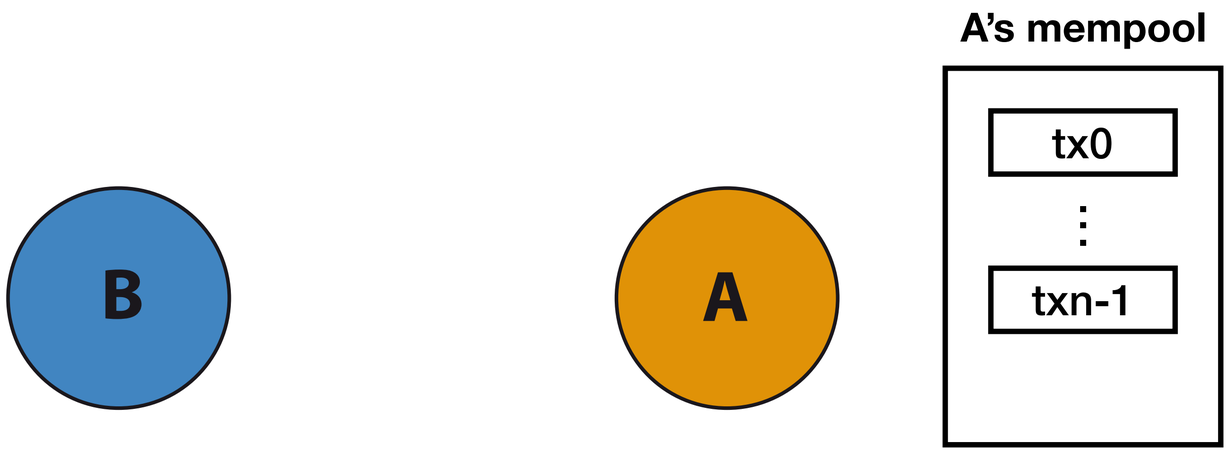
0-CONFIRMATION TRANSACTIONS (1/2)



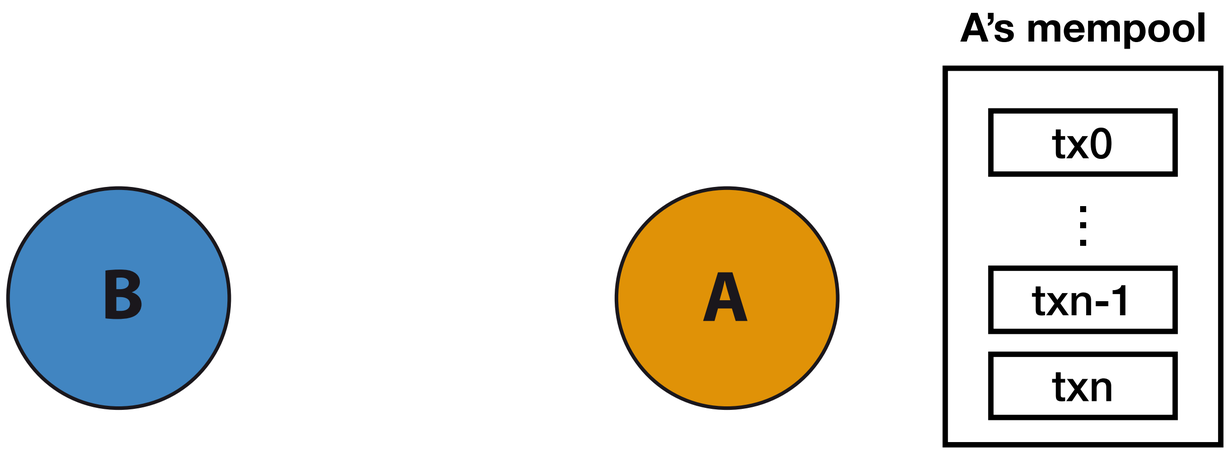
0-CONFIRMATION TRANSACTIONS (1/2)



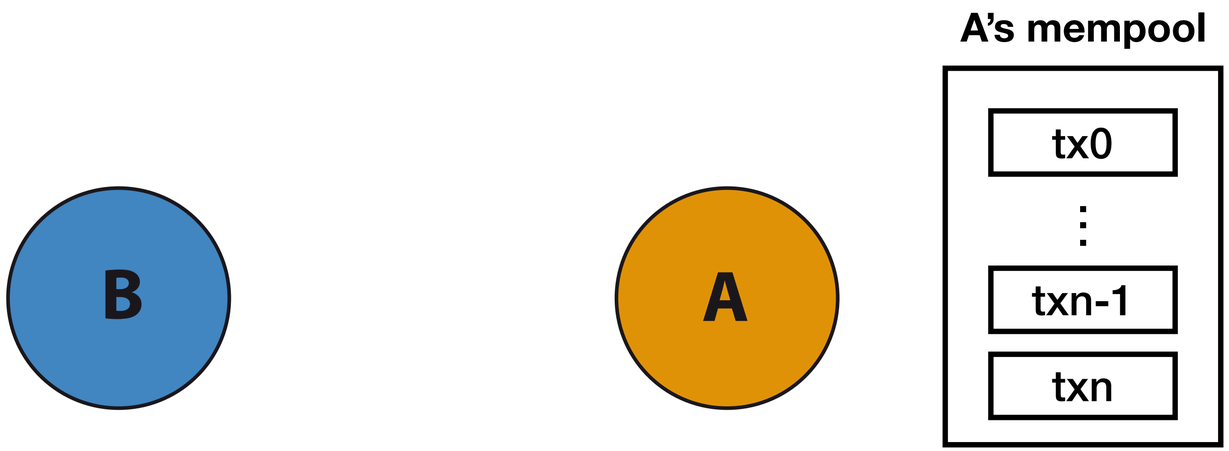
0-CONFIRMATION TRANSACTIONS (1/2)



0-CONFIRMATION TRANSACTIONS (1/2)



0-CONFIRMATION TRANSACTIONS (1/2)



Transactions sitting in memory (mempool) are unconfirmed and should not be trusted

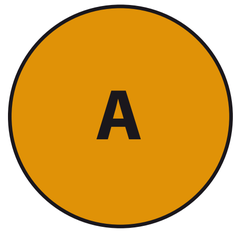
0-CONFIRMATION TRANSACTIONS (2/2)

0-conf transactions / **unconfirmed** transactions are those that are not part of the blockchain (**they are stored in the mempool**)

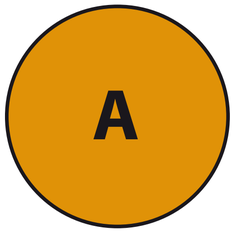
0-conf transactions are not covered by the double-spending protection offered by the blockchain (they are not part of it)

Different nodes can have conflicting version of the **“same transaction”**

DOUBLE-SPENDING TRANSACTIONS (1/2)

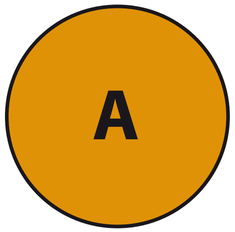


DOUBLE-SPENDING TRANSACTIONS (1/2)



id = 4F3…ED

DOUBLE-SPENDING TRANSACTIONS (1/2)



id = 4F3…ED

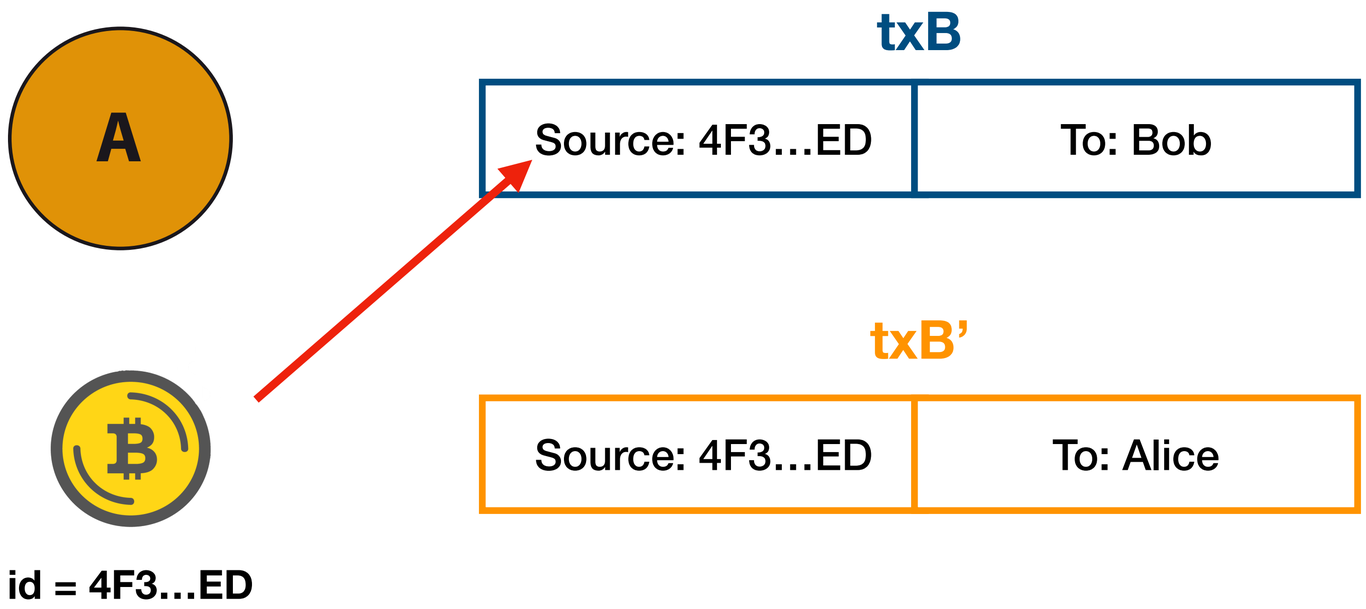
txB

Source: 4F3…ED	To: Bob

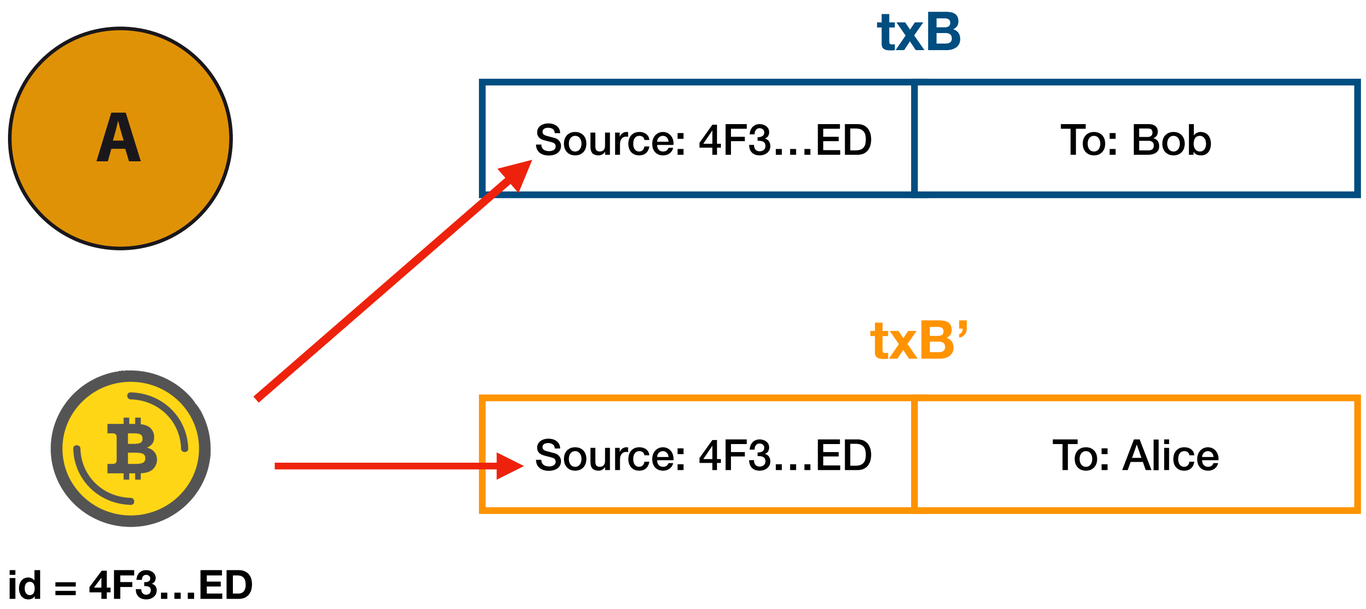
txB’

Source: 4F3…ED	To: Alice

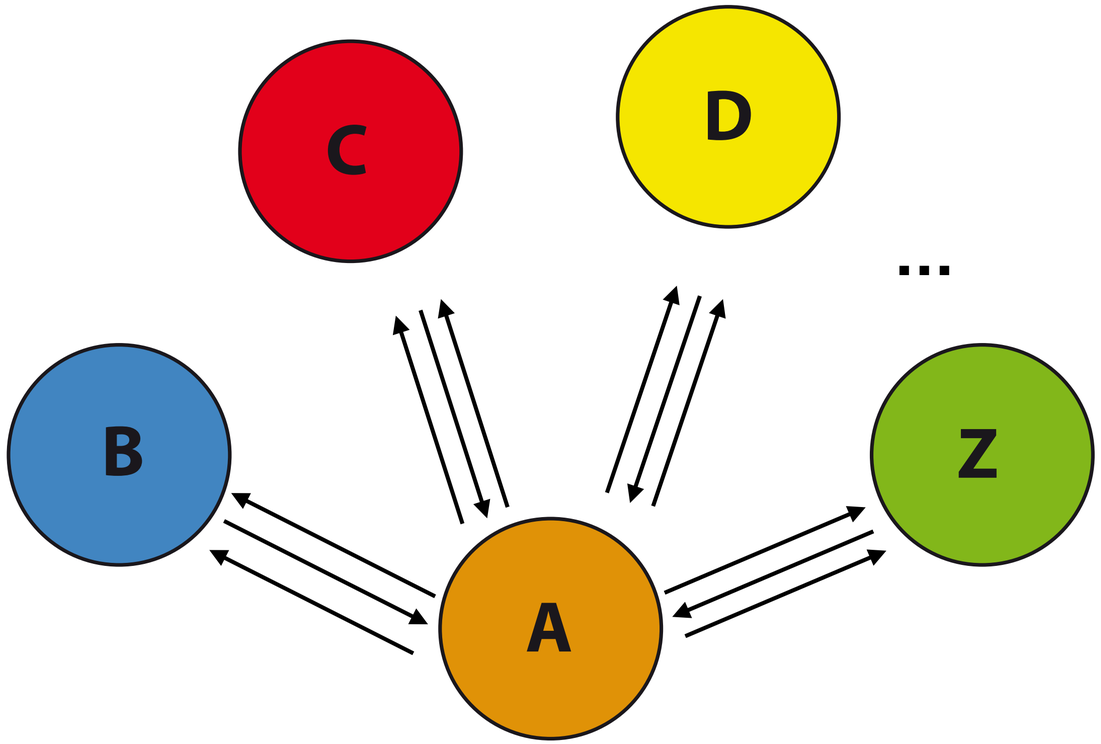
DOUBLE-SPENDING TRANSACTIONS (1/2)



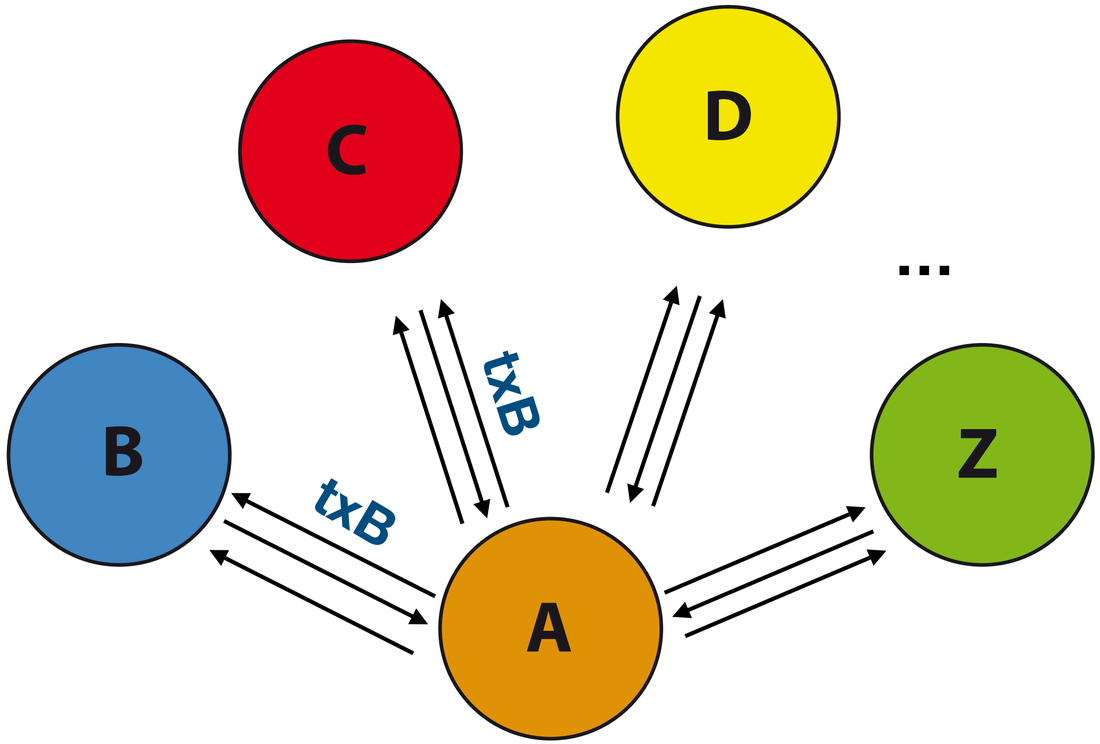
DOUBLE-SPENDING TRANSACTIONS (1/2)



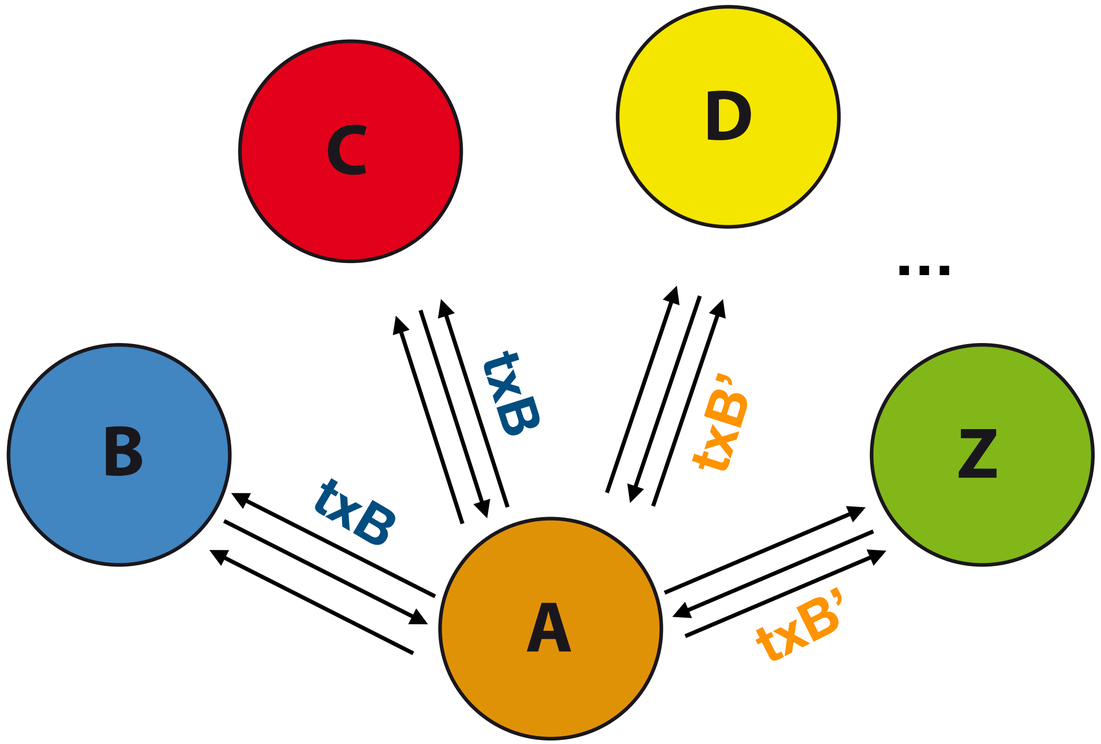
DOUBLE-SPENDING TRANSACTIONS (2/2)



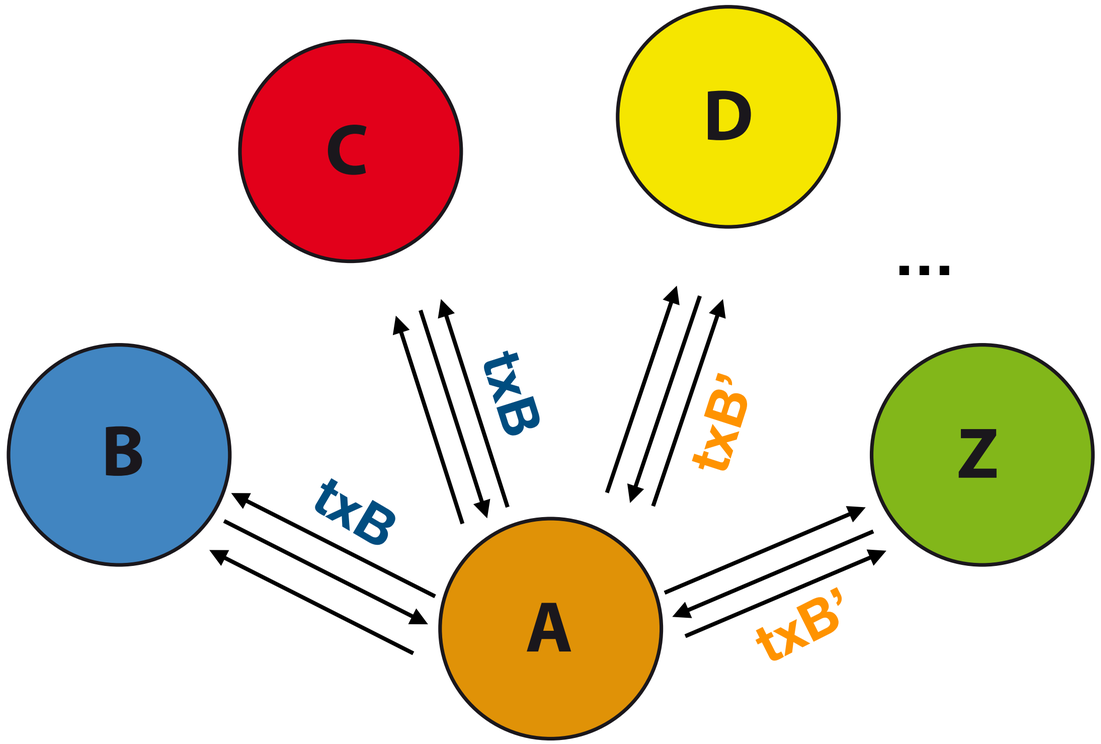
DOUBLE-SPENDING TRANSACTIONS (2/2)



DOUBLE-SPENDING TRANSACTIONS (2/2)

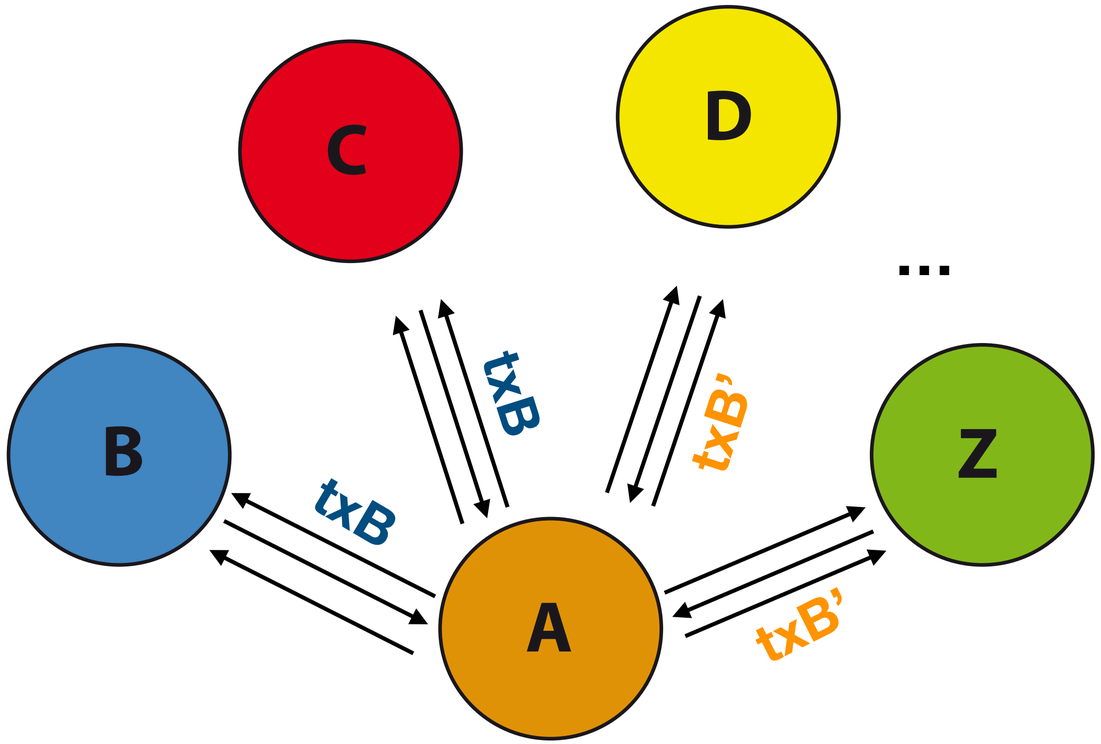


DOUBLE-SPENDING TRANSACTIONS (2/2)



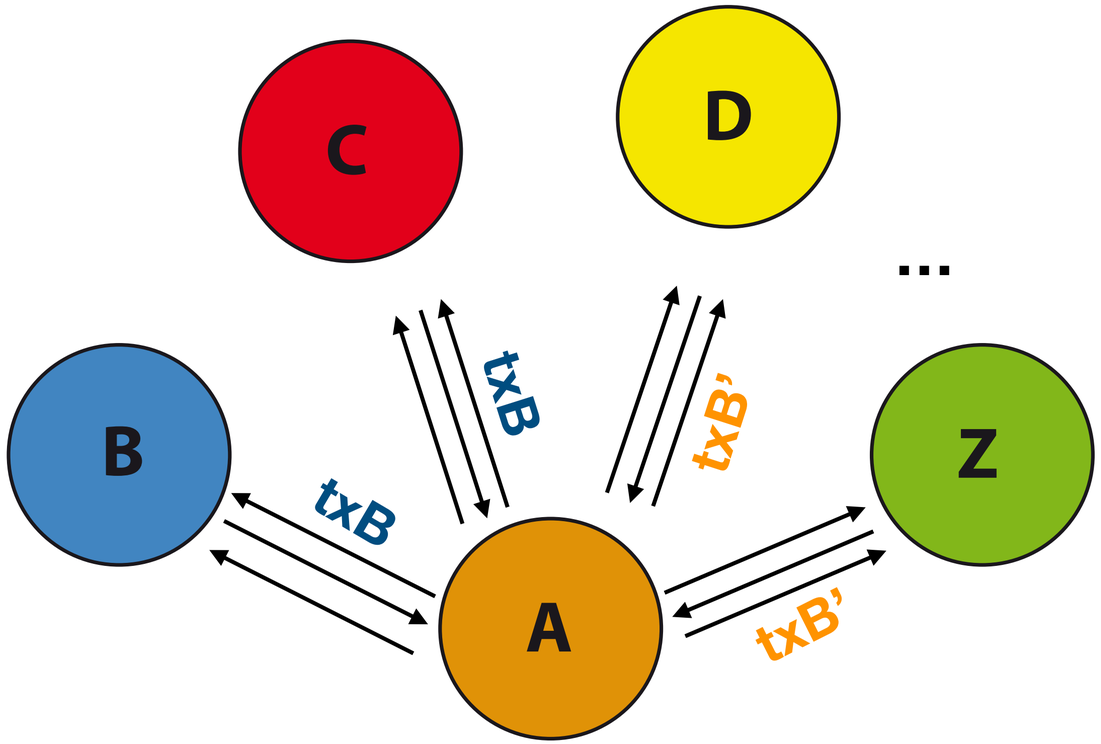
- **0-conf** transactions should not be trusted

DOUBLE-SPENDING TRANSACTIONS (2/2)



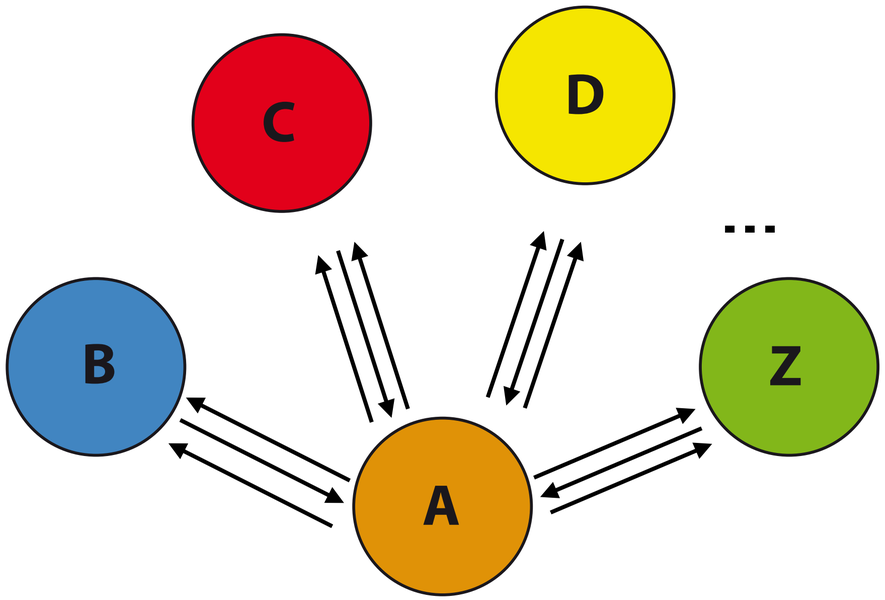
- **0-conf** transactions should not be trusted
- If B accepts txB before it appears in a block he **can be deceived** by A

DOUBLE-SPENDING TRANSACTIONS (2/2)

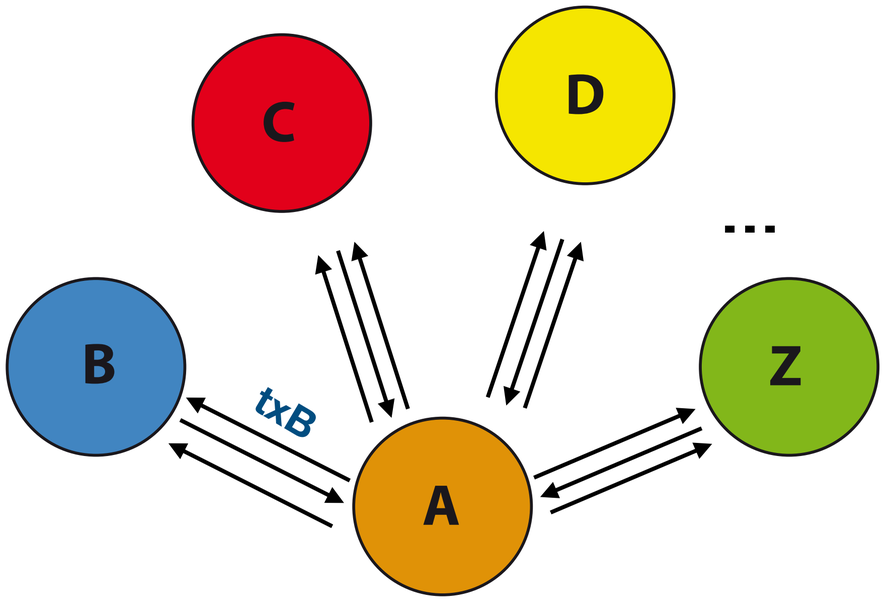


- **0-conf** transactions should not be trusted
- If B accepts txB before it appears in a block he **can be deceived** by A
- The de facto confirmation time is **6 blocks** (5 on top of the one including a certain transaction)

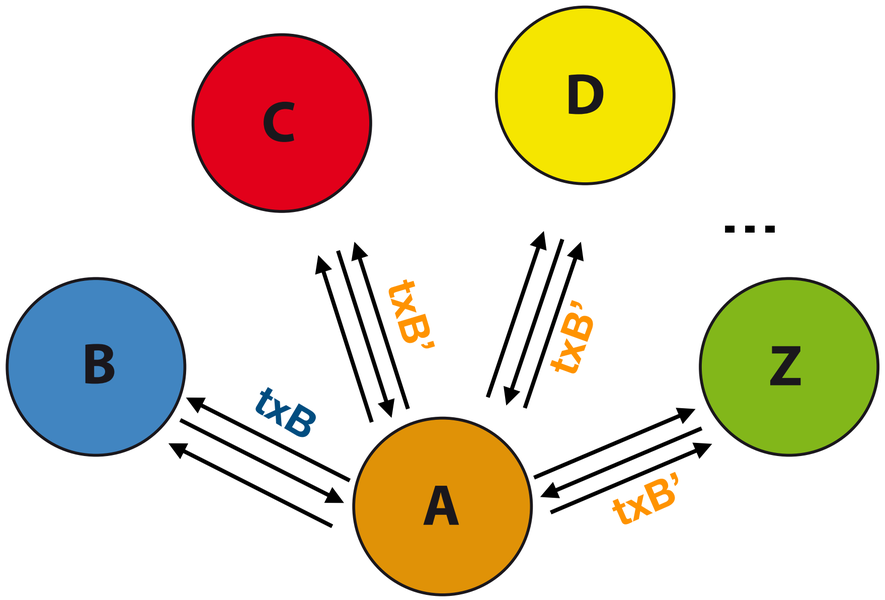
WHEN THINGS GO SOUTH



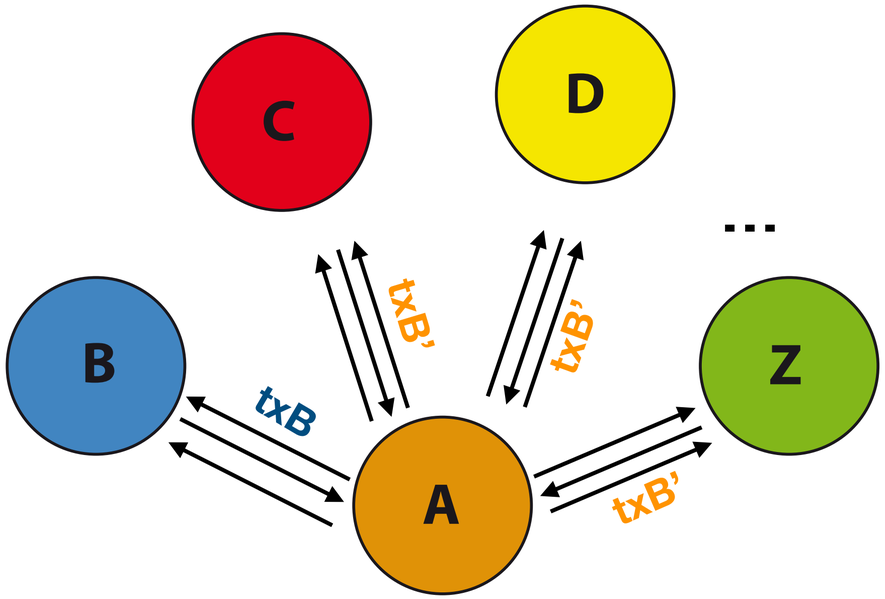
WHEN THINGS GO SOUTH



WHEN THINGS GO SOUTH

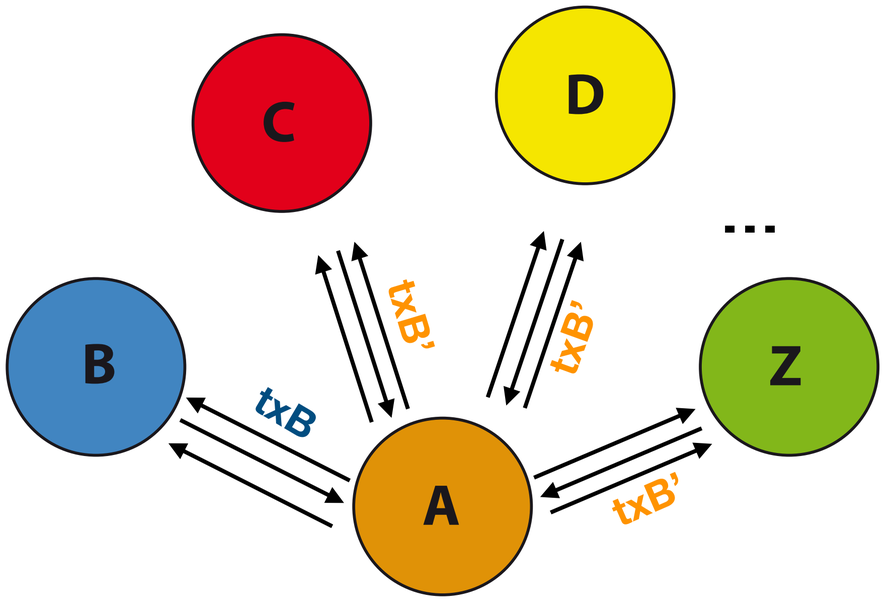


WHEN THINGS GO SOUTH



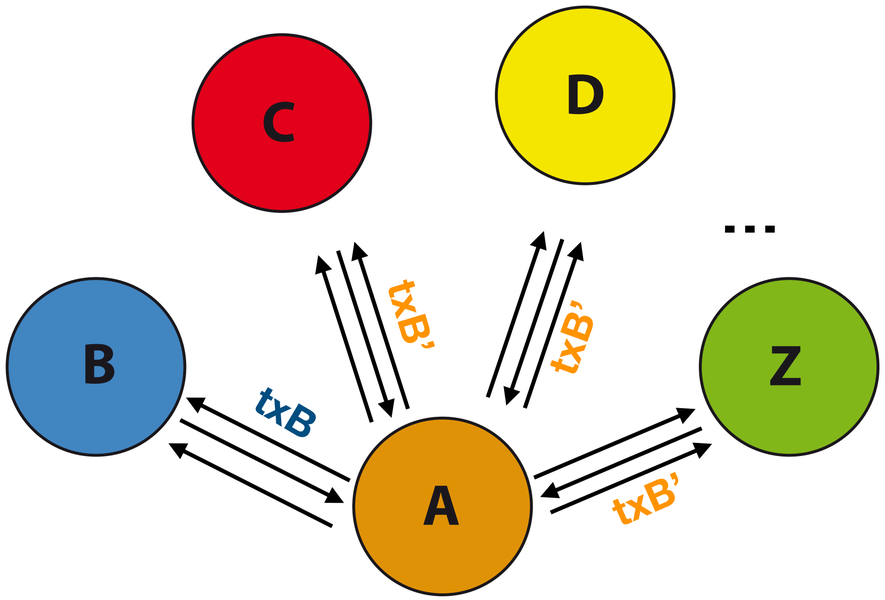
- If A controls the **network view** of B, A can control what B know about the currency

WHEN THINGS GO SOUTH



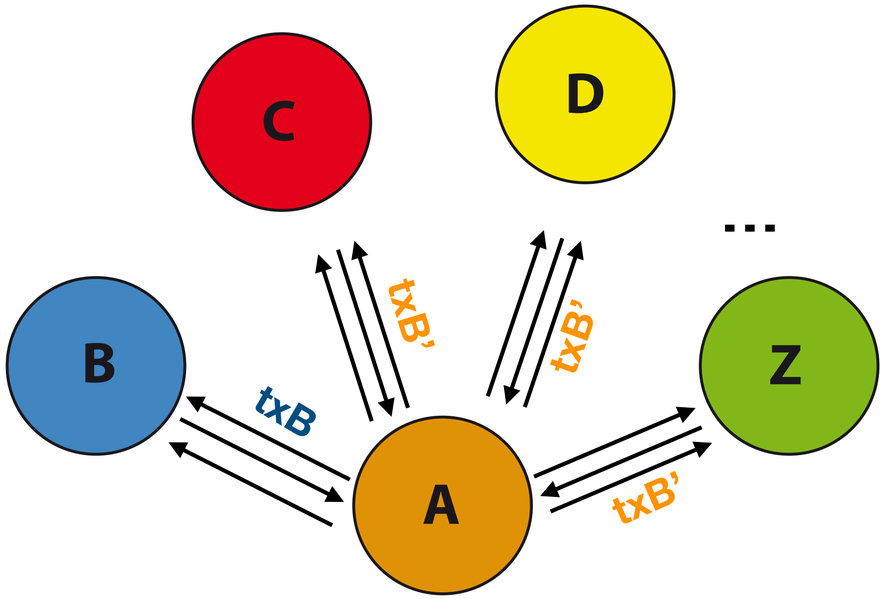
- If A controls the **network view** of B, A can control what B know about the currency
- B is said to be **eclipsed**

WHEN THINGS GO SOUTH



- If A controls the **network view** of B, A can control what B know about the currency
- B is said to be **eclipsed**
- A will be able to **easily fool** B

WHEN THINGS GO SOUTH



- If A controls the **network view** of B, A can control what B know about the currency
- B is said to be **eclipsed**
- A will be able to **easily fool** B

Ethan Heilman, Alison Kendler, Aviv Zohar and Sharon Goldberg
Eclipse Attacks on Bitcoin's Peer-to-Peer Network
https://www.usenix.org/node/190891

Network topology

UNKNOWN TOPOLOGY BY DESIGN

Peers are chosen pseudorandomly from the peer database of a node in order to become neighbors

UNKNOWN TOPOLOGY BY DESIGN

Peers are chosen pseudorandomly from the peer database of a node in order to become neighbors

Peers can be requested from other peers, but no information about whether the responder is (or has been) a neighbor of any of the provided peers is given

UNKNOWN TOPOLOGY BY DESIGN

Peers are chosen pseudorandomly from the peer database of a node in order to become neighbors

Peers can be requested from other peers, but no information about whether the responder is (or has been) a neighbor of any of the provided peers is given

The network topology should mimic a random network

INFERRING THE TOPOLOGY

Does the network really look random?

INFERRING THE TOPOLOGY

Does the network really look random?

How can we known if we don’t know what the topology looks like?

INFERRING THE TOPOLOGY

Does the network really look random?

How can we known if we don’t know what the topology looks like?

Can we do anything to infer the topology?

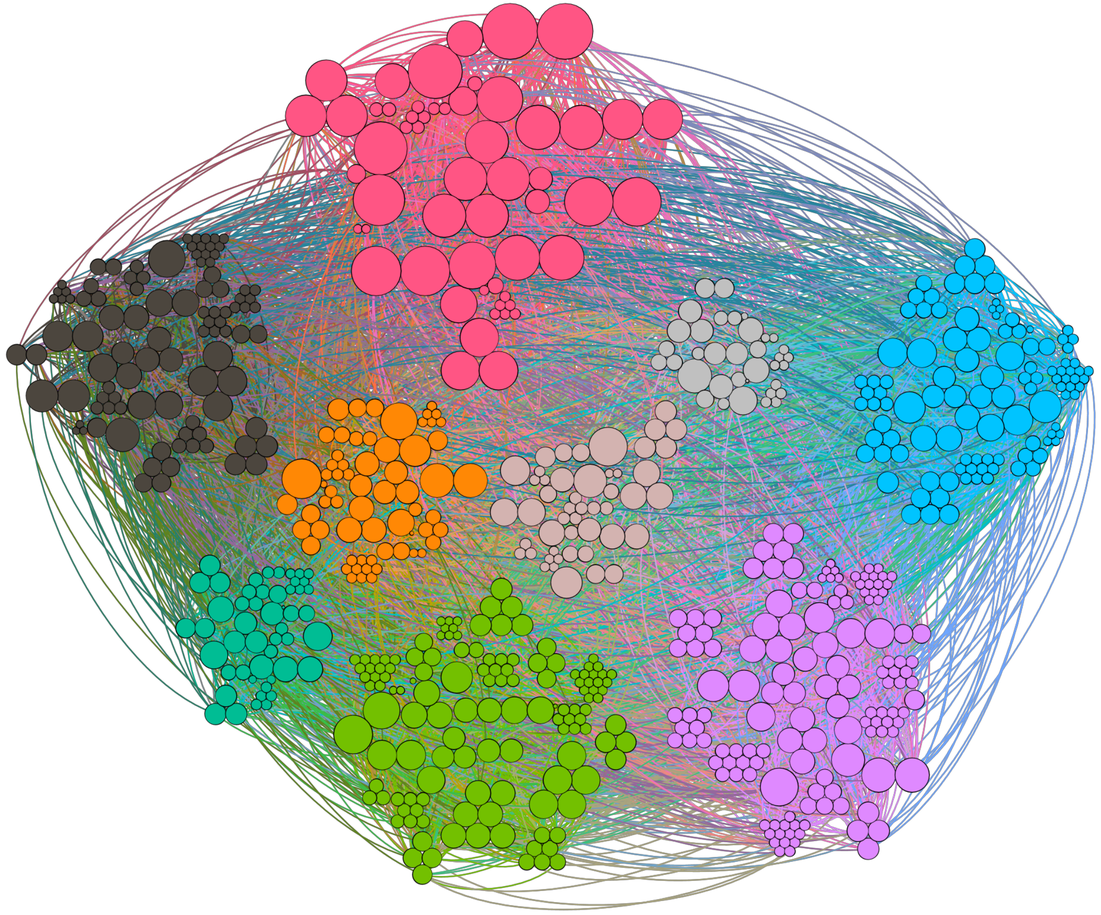
INFERRING THE TOPOLOGY

Does the network really look random?

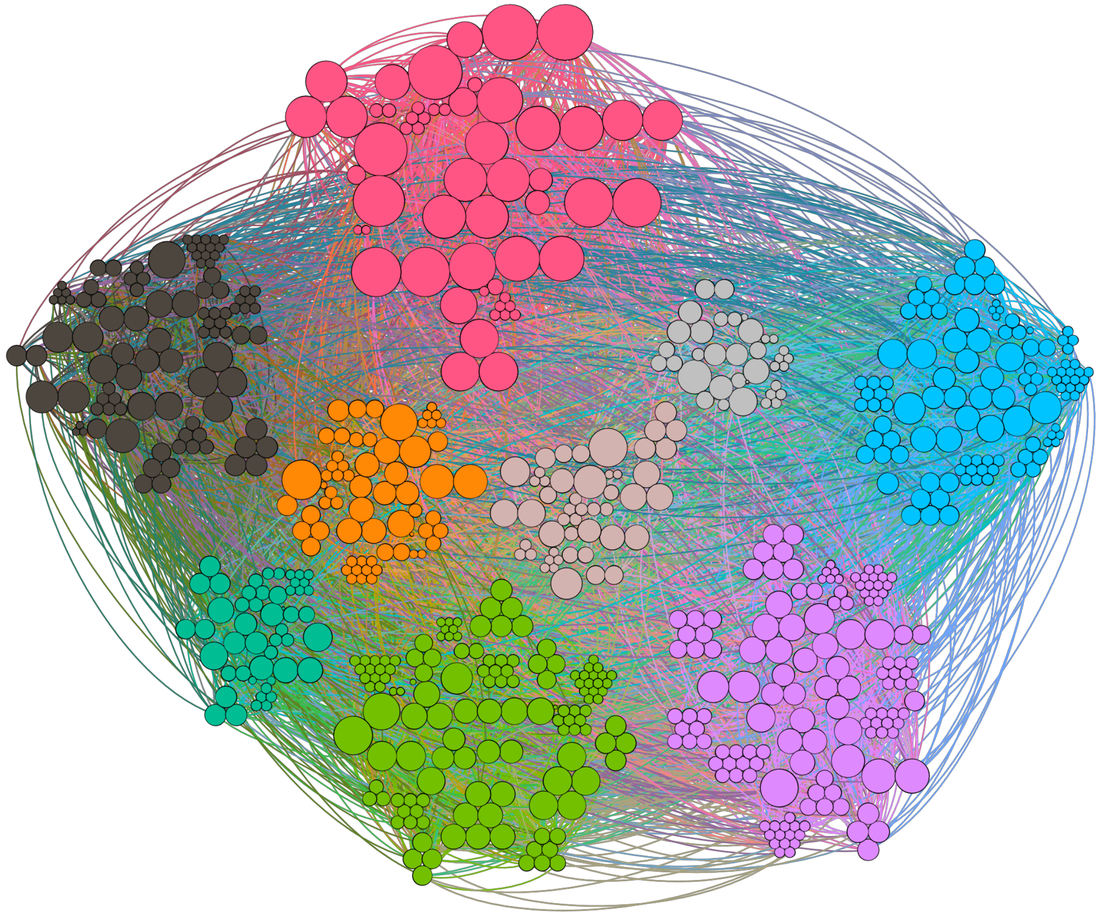
How can we known if we don’t know what the topology looks like?

Can we do anything to infer the topology?

TESTNET TOPOLOGY

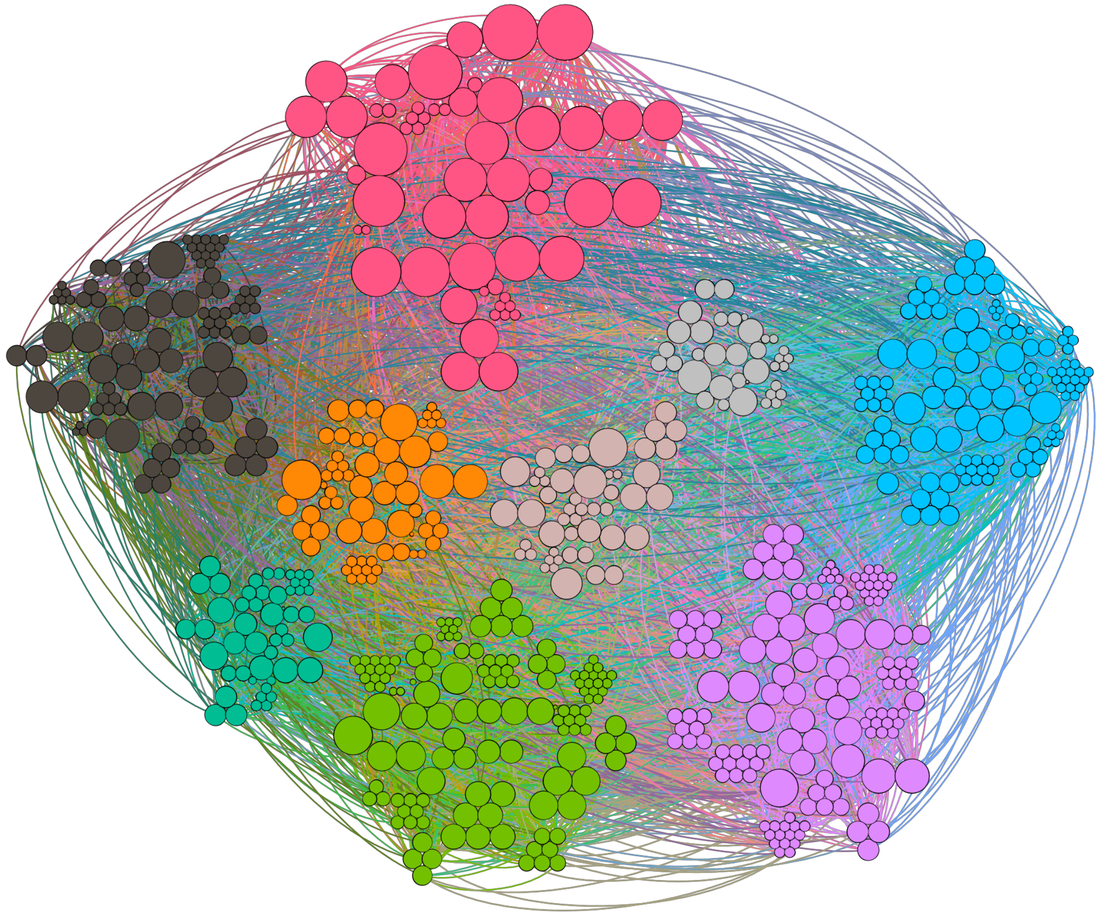


TESTNET TOPOLOGY



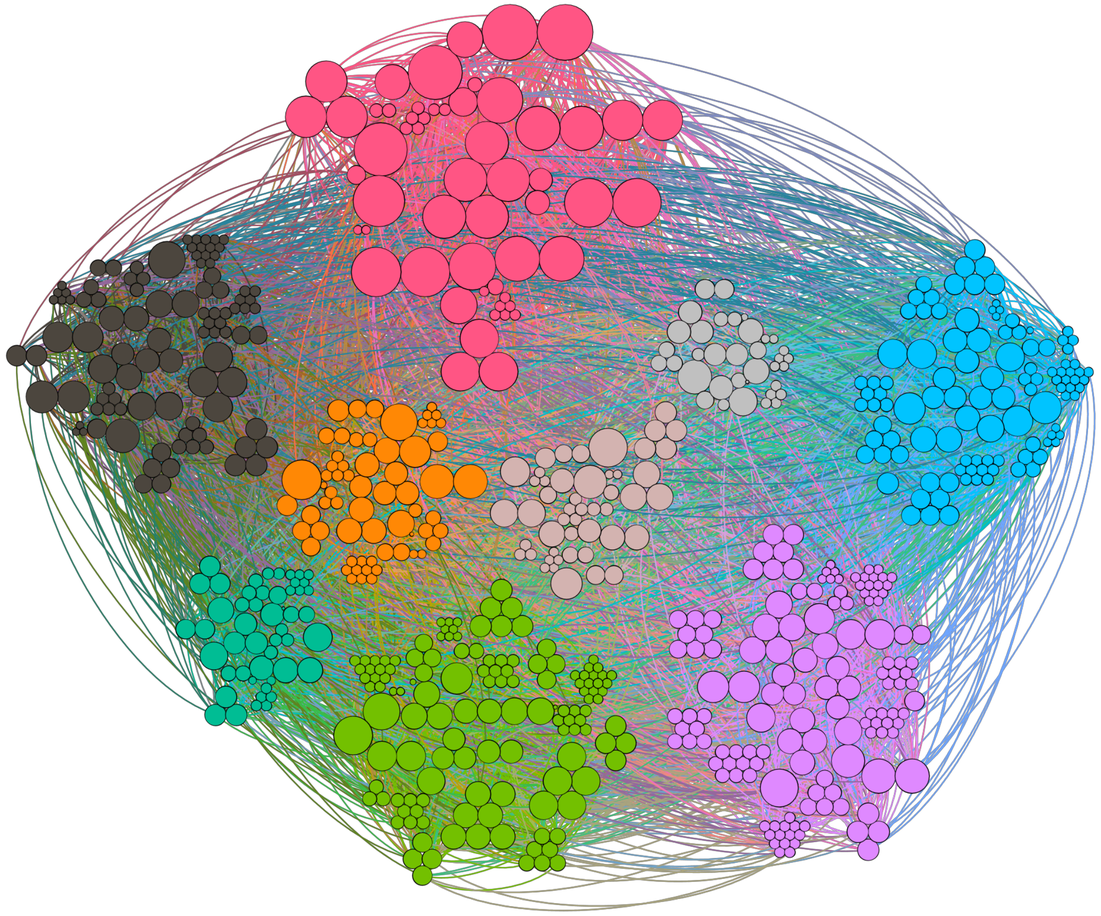
- Several communities can be easily identified

TESTNET TOPOLOGY



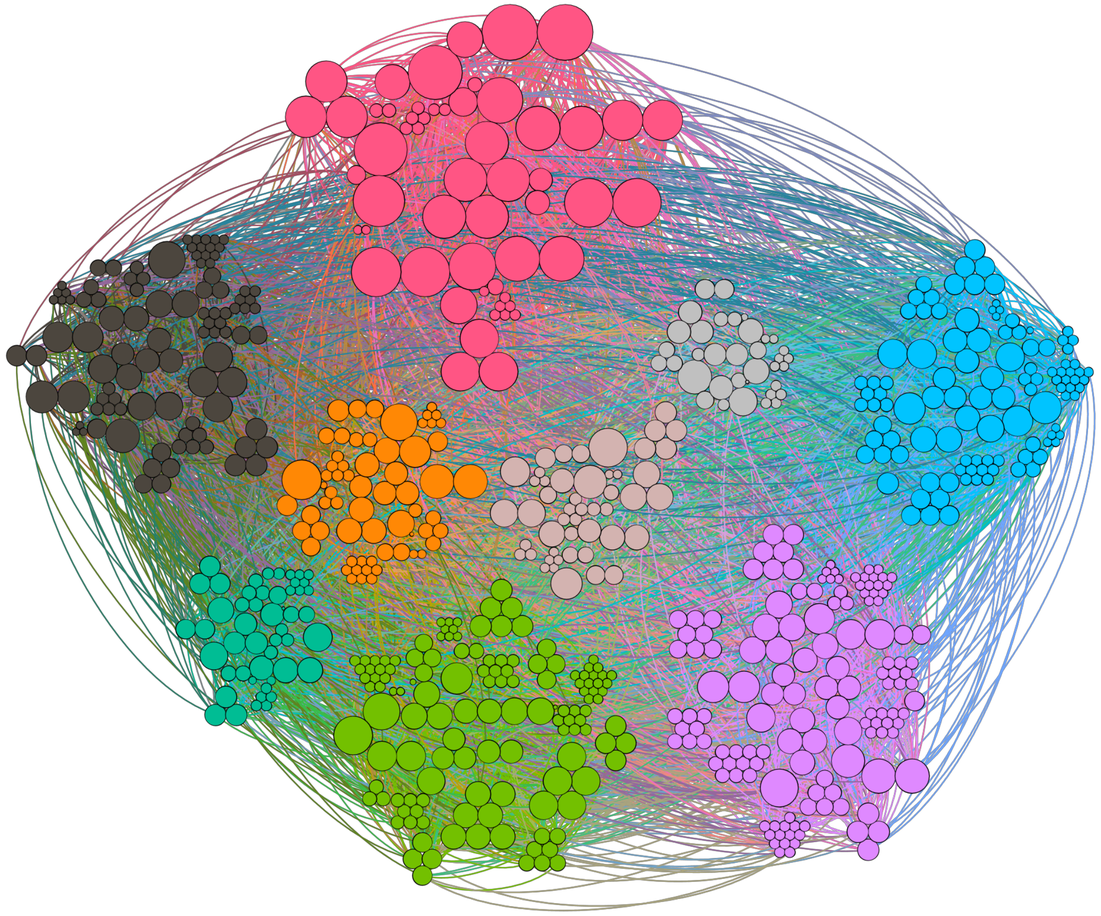
- Several communities can be easily identified
- The network looks far from a random graph of similar characteristics

TESTNET TOPOLOGY



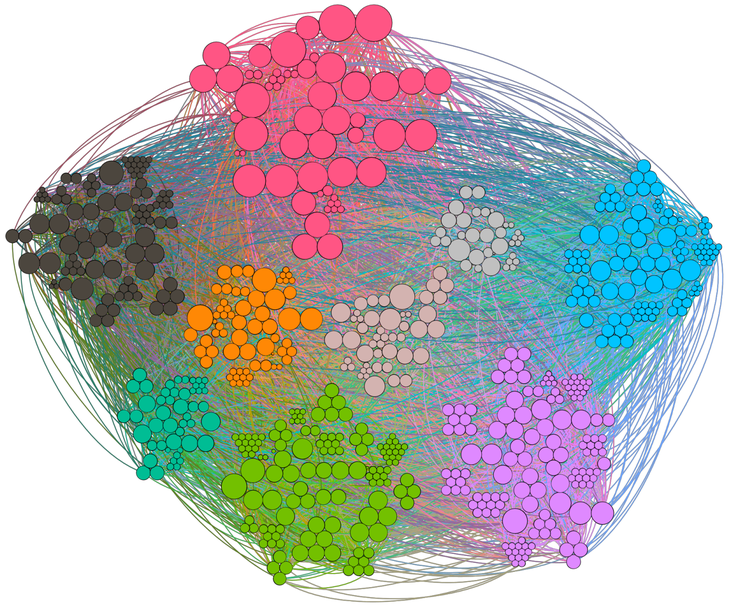
- Several communities can be easily identified
- The network looks far from a random graph of similar characteristics
- The topology can be used to identify undesired centralization

TESTNET TOPOLOGY



- Several communities can be easily identified
- The network looks far from a random graph of similar characteristics
- The topology can be used to identify undesired centralization
- But also to target some nodes (first step in several network based attacks)

TESTNET TOPOLOGY



- Several communities can be easily identified
- The bigger the node the higher its degree
- The network looks far from a random graph of similar characteristics

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

TxProbe: Discovering Bitcoin's Network Topology Using Orphan Transactions

https://fc19.ifca.ai/preproceedings/58-preproceedings.pdf