

Network Layer and Information Propagation



Sergi Delgado Segura

Research associate @ UCL InfoSec



Before we start

We will use Bitcoin as example when explaining how certain parts of the network work. However, the same mechanisms apply to most of the existing cryptocurrencies with slight modifications (some times even without any).

Also keep in mind that for most of things within cryptocurrencies there is no formal specification but the live code. Therefore some details may change in the near future.



Introduction

Client-Server paradigm (1/2)

- Classic paradigm
- Actors are split in two types:
 - Servers:
 - serve specific resources upon request
 - can also provide different types of services

Client-Server paradigm (2/2)

- Clients:
 - resource/service requesters
 - do not share resources nor provide any service
- Clients initiate the communication and need to know the server endpoint
- Classical examples: WWW, DNS, Email, etc

Peer-to-peer (P2P) paradigm

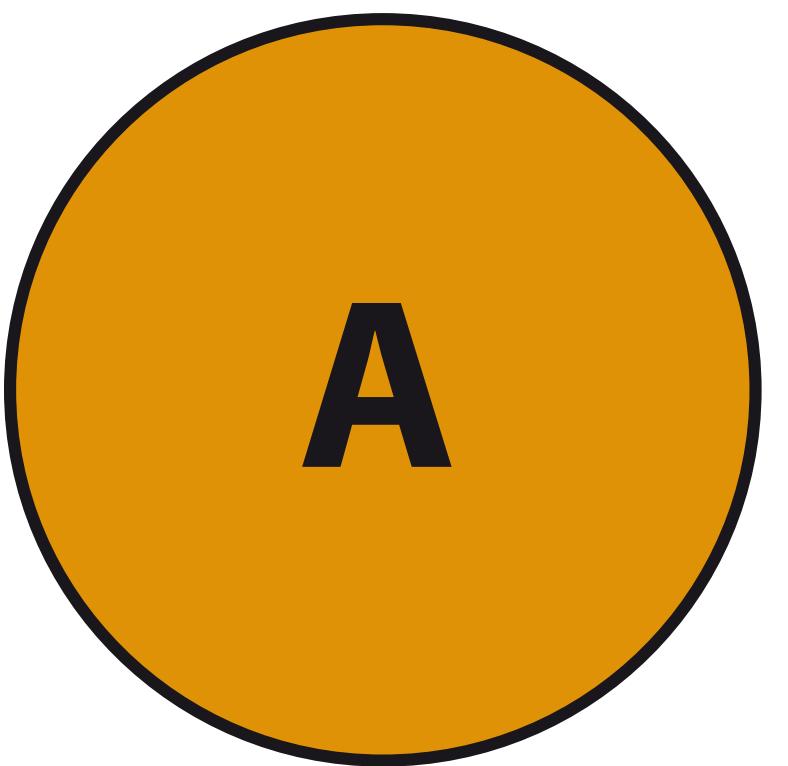
- All actors (**peers**) are equal and have both client and server capabilities
- Services / resources can be shared between several peers or found in a single location
- Each peer can choose what to serve/request
- Quite usual paradigm for distributed file sharing (e.g: Torrent)
- **Usual problems:** Bootstrapping and file searching



Bitcoin P2P bootstrapping (peer discovery)



Bitcoin P2P bootstrapping (peer discovery)



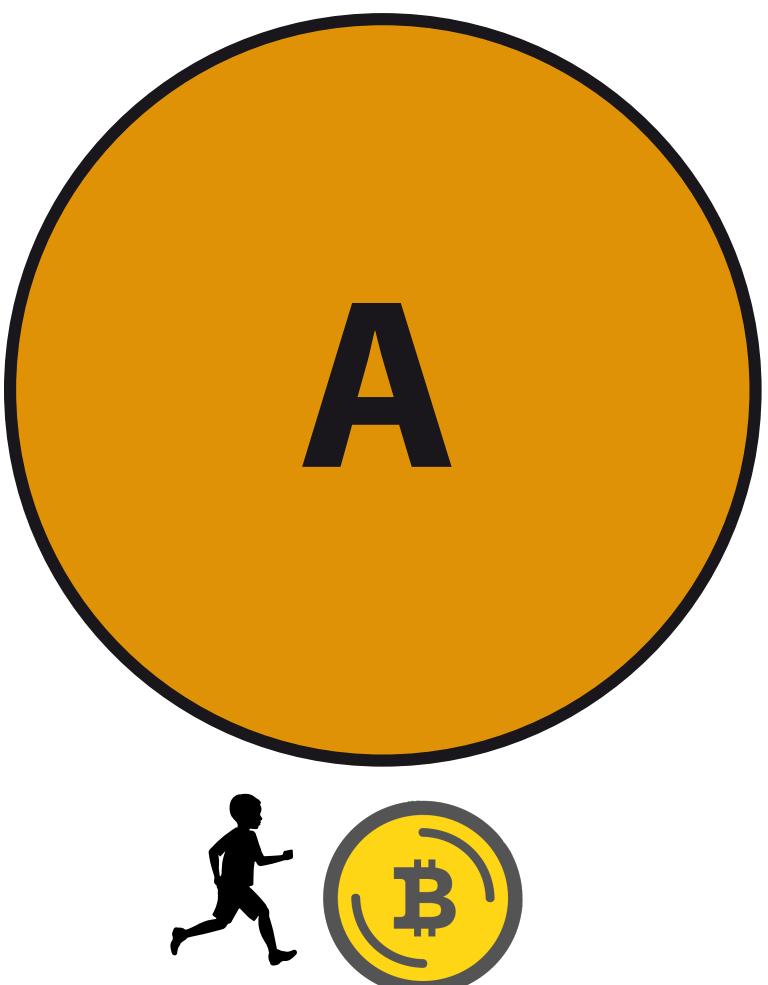


Bitcoin P2P bootstrapping (peer discovery)

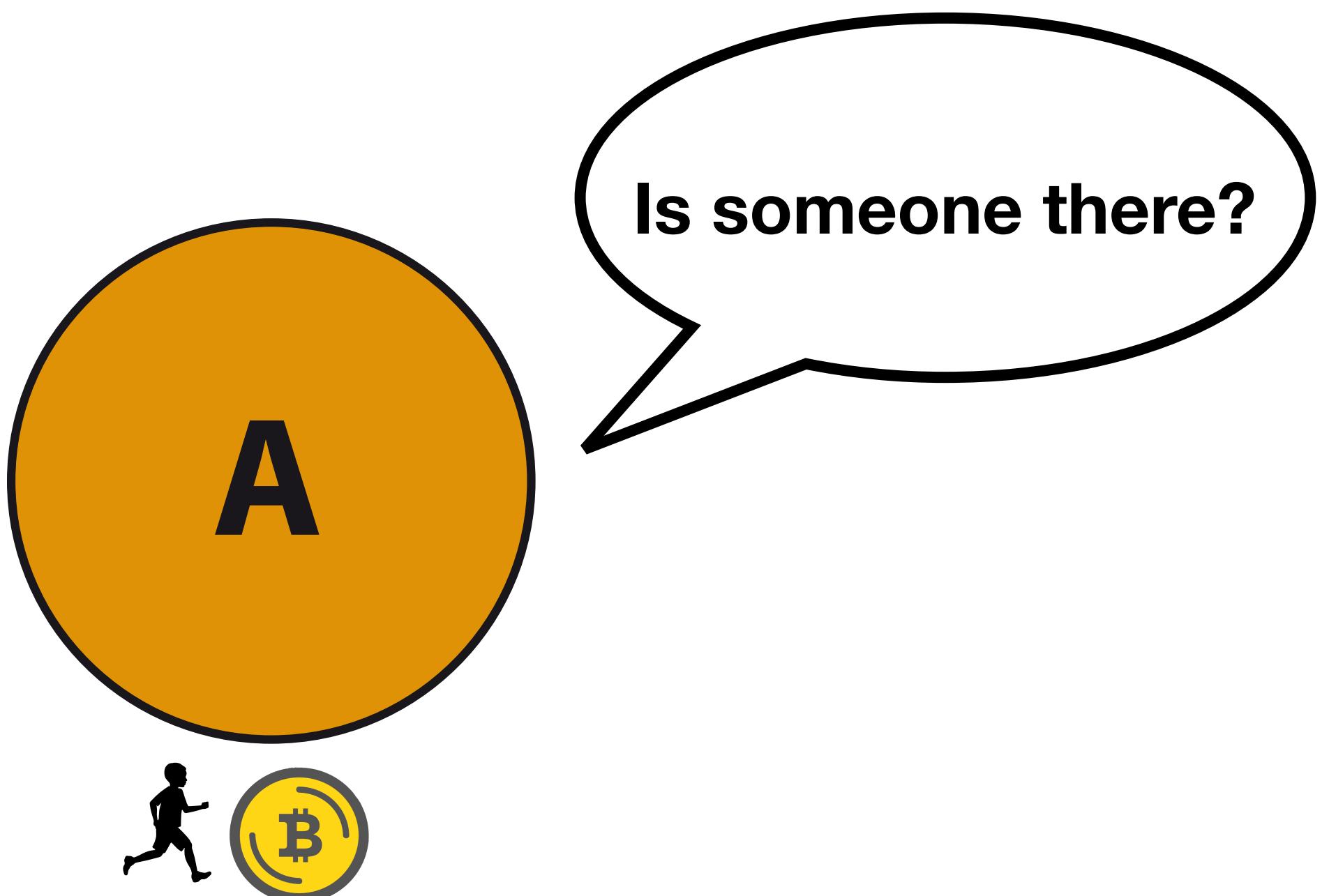




Bitcoin P2P bootstrapping (peer discovery)

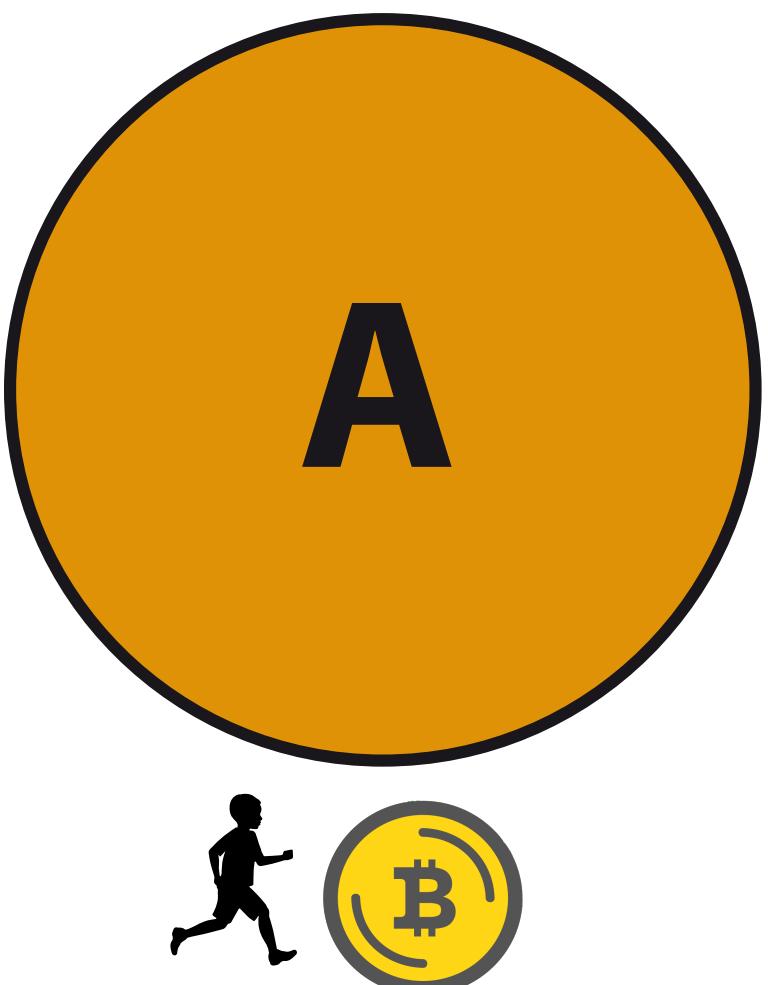


Bitcoin P2P bootstrapping (peer discovery)



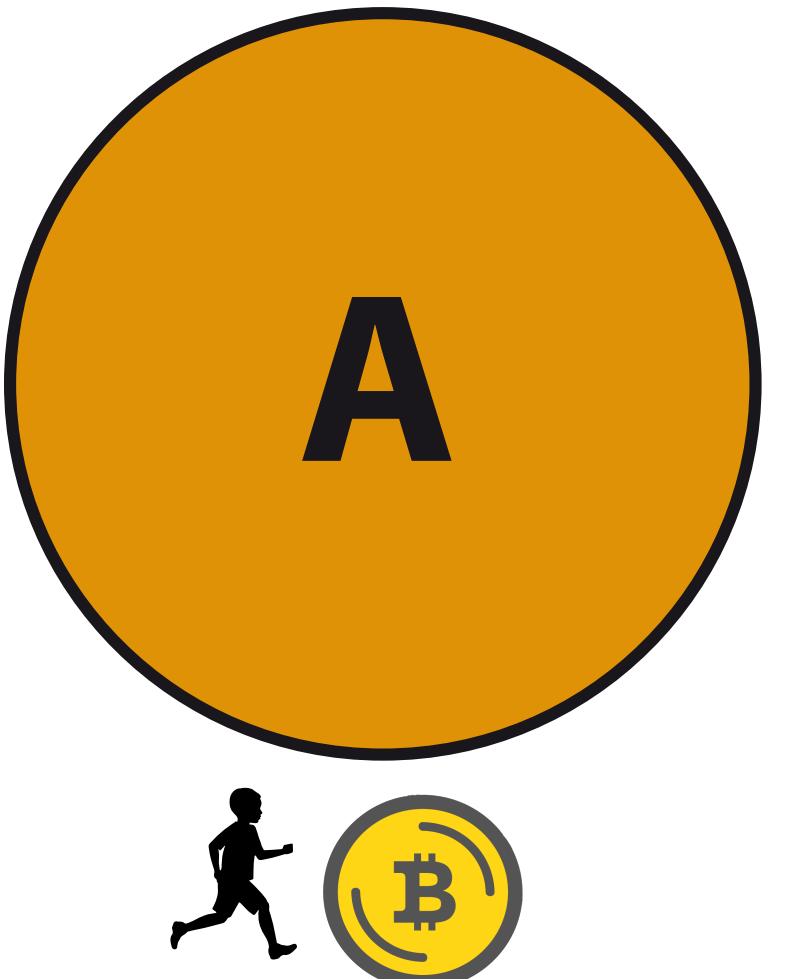


Bitcoin P2P bootstrapping (peer discovery)





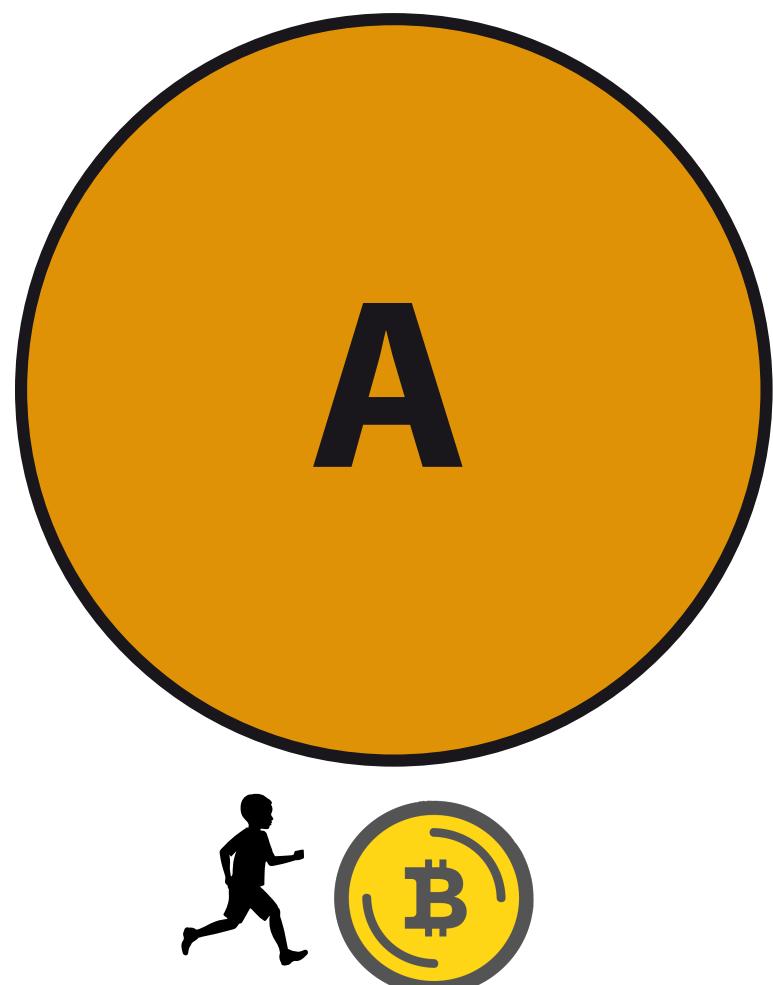
Bitcoin P2P bootstrapping (peer discovery)



****tumbleweed****



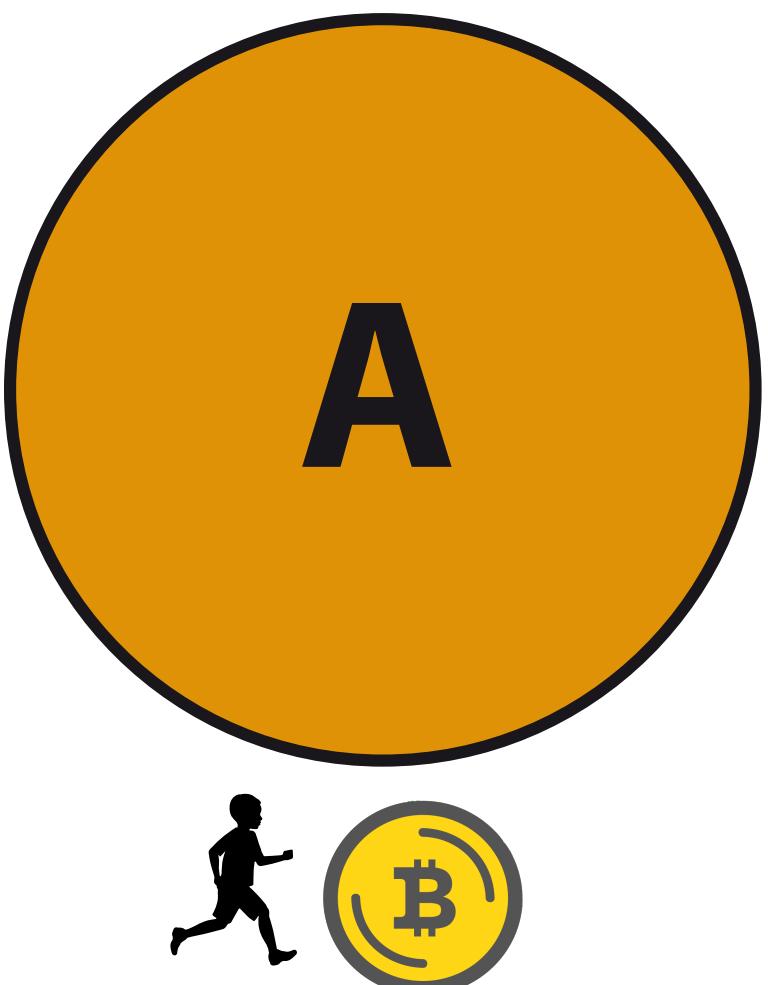
Bitcoin P2P bootstrapping (peer discovery)



****tumbleweed****



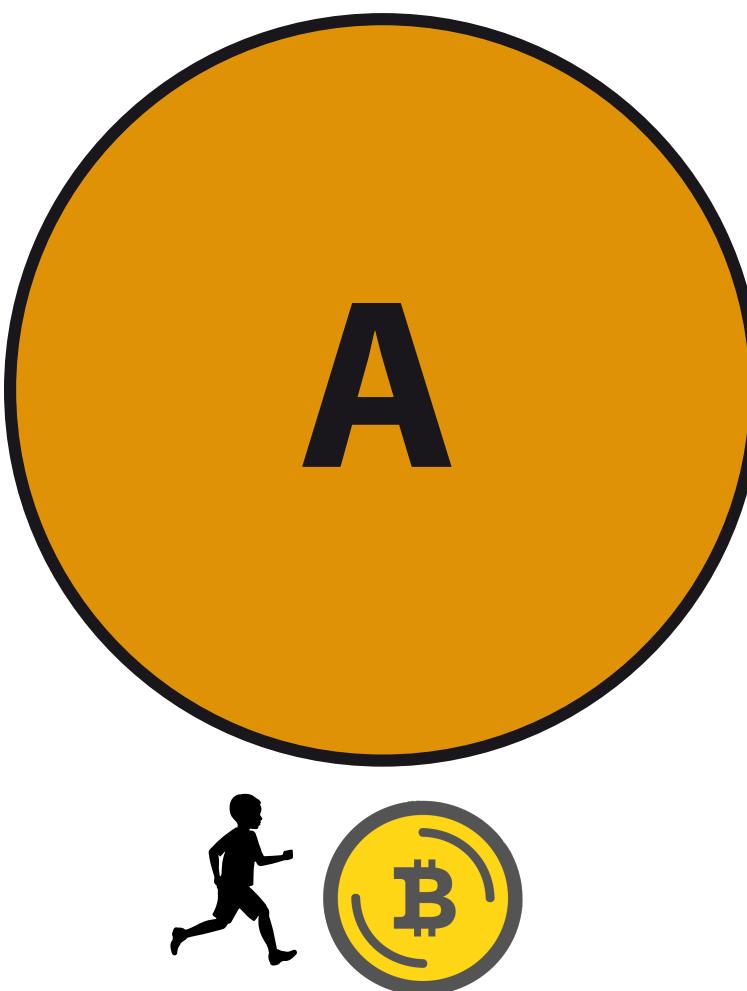
Bitcoin P2P bootstrapping (peer discovery)



****tumbleweed**



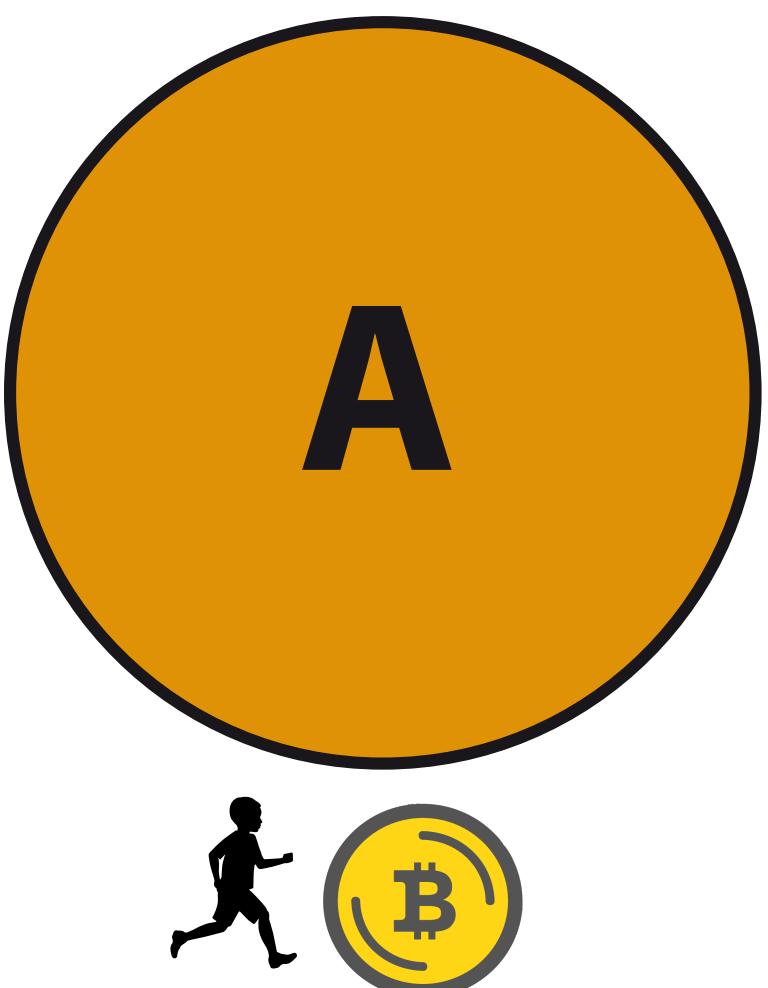
Bitcoin P2P bootstrapping (peer discovery)



****tumbleweed****



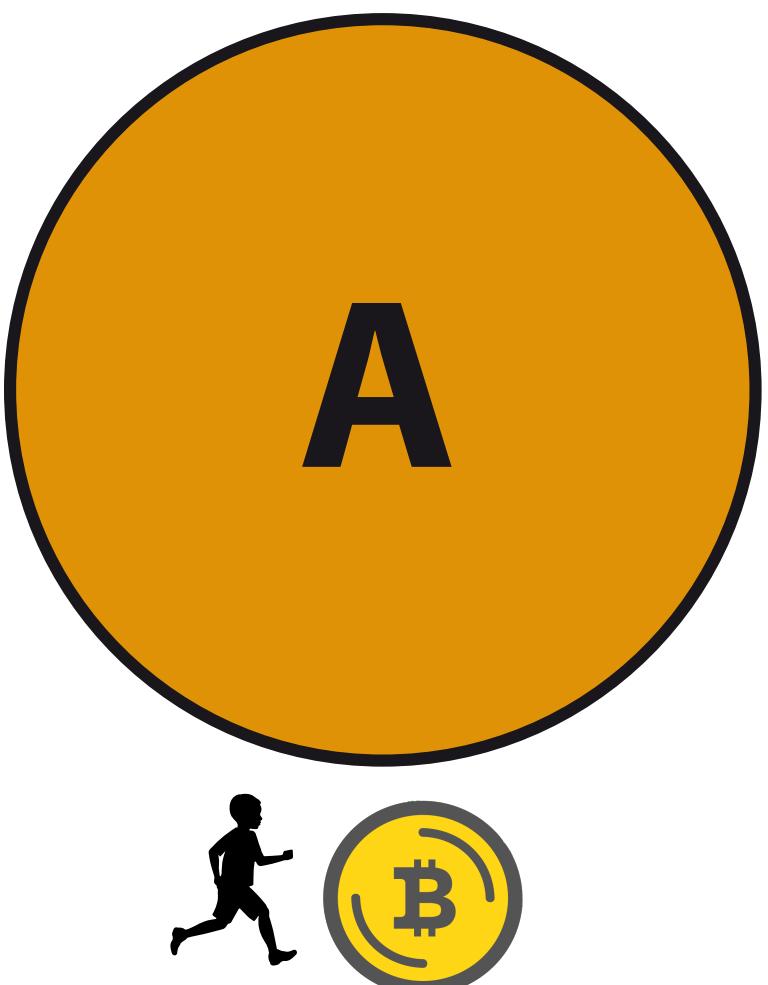
Bitcoin P2P bootstrapping (peer discovery)



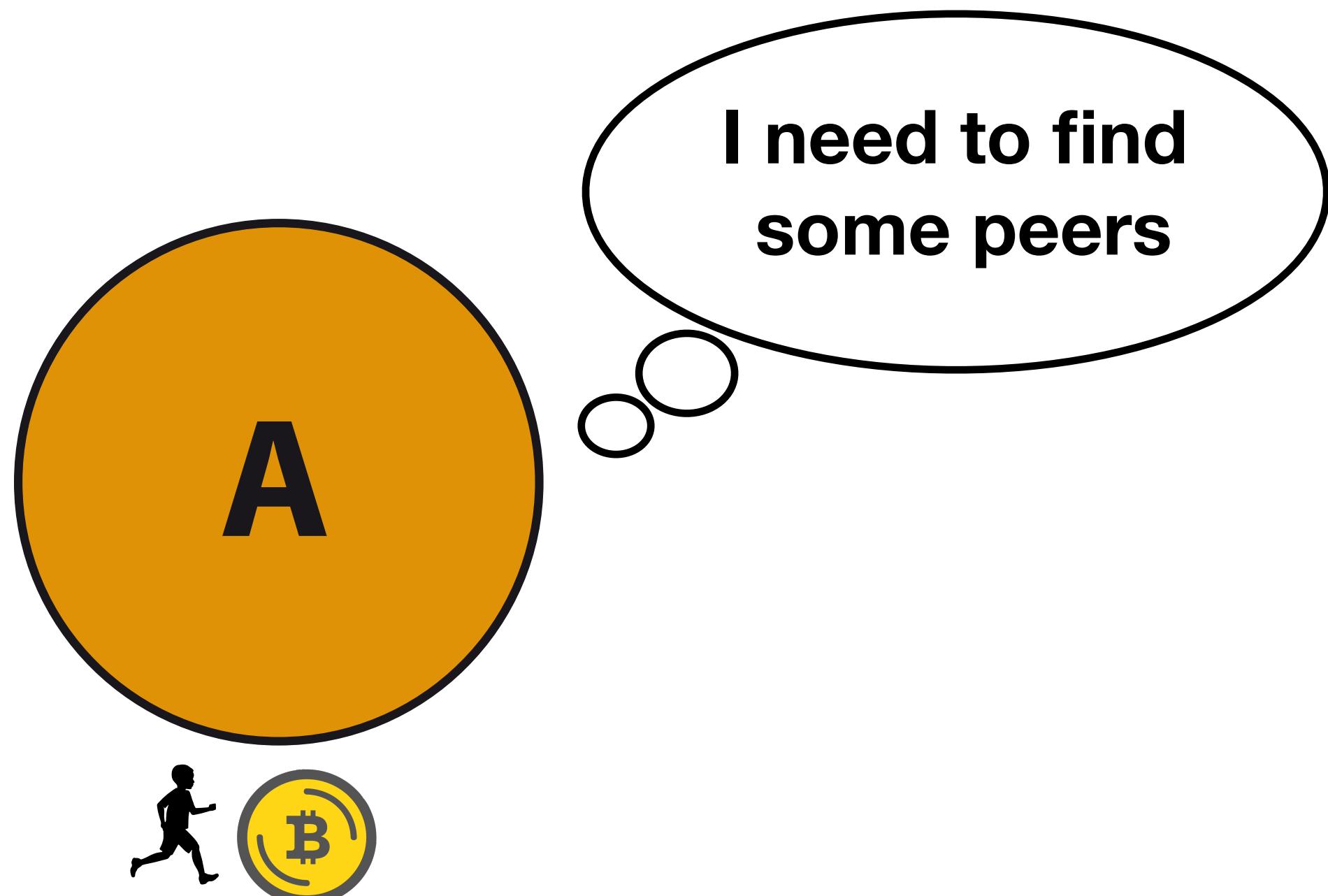
****tumbleweed****



Bitcoin P2P bootstrapping (peer discovery)

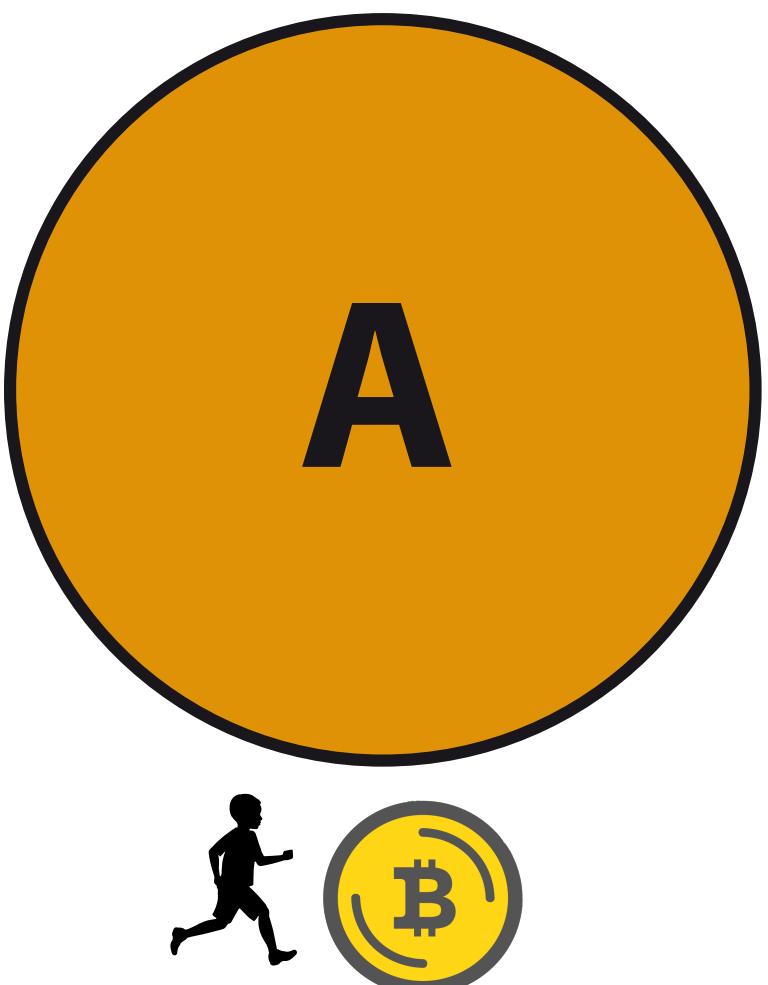


Bitcoin P2P bootstrapping (peer discovery)



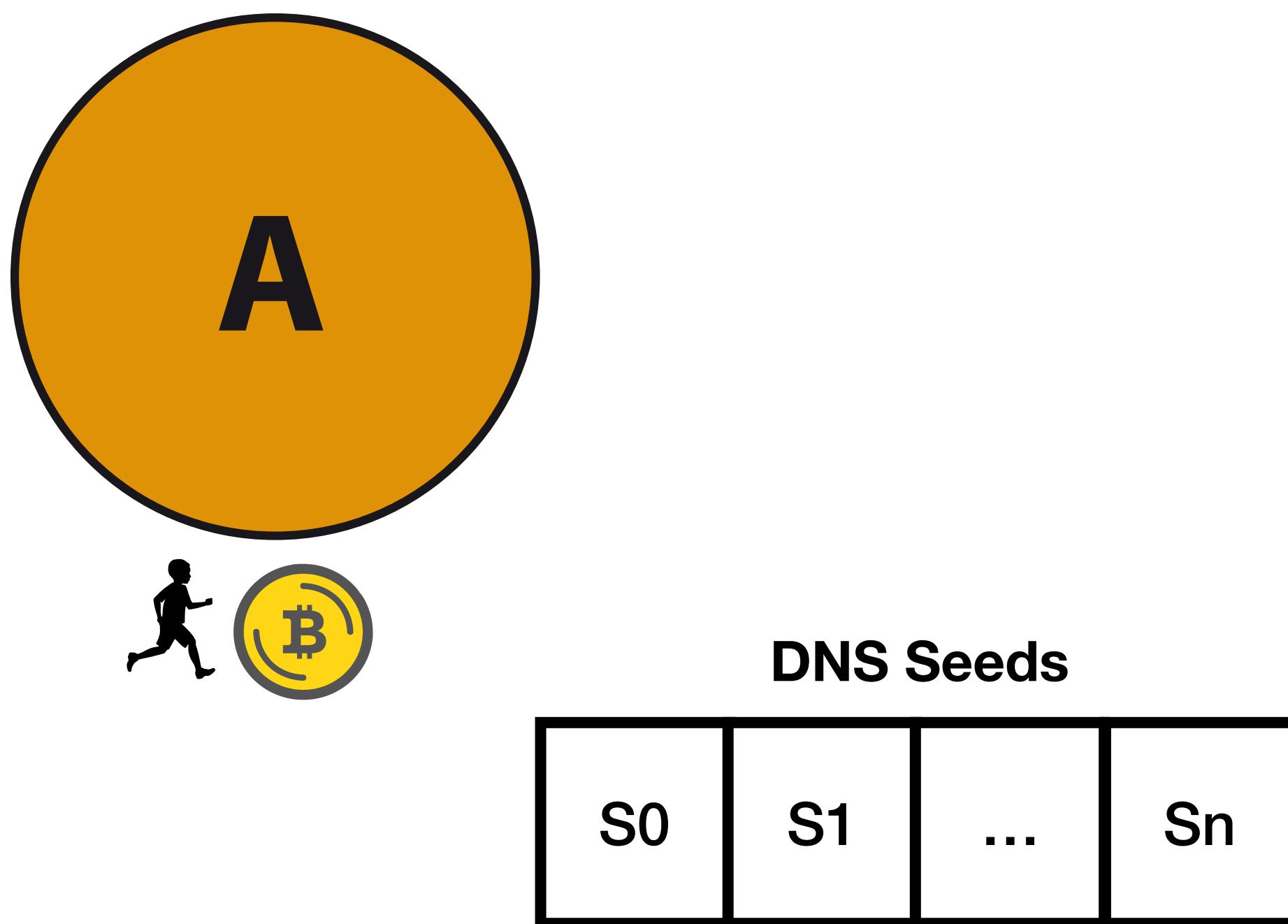


Bitcoin P2P bootstrapping (peer discovery)



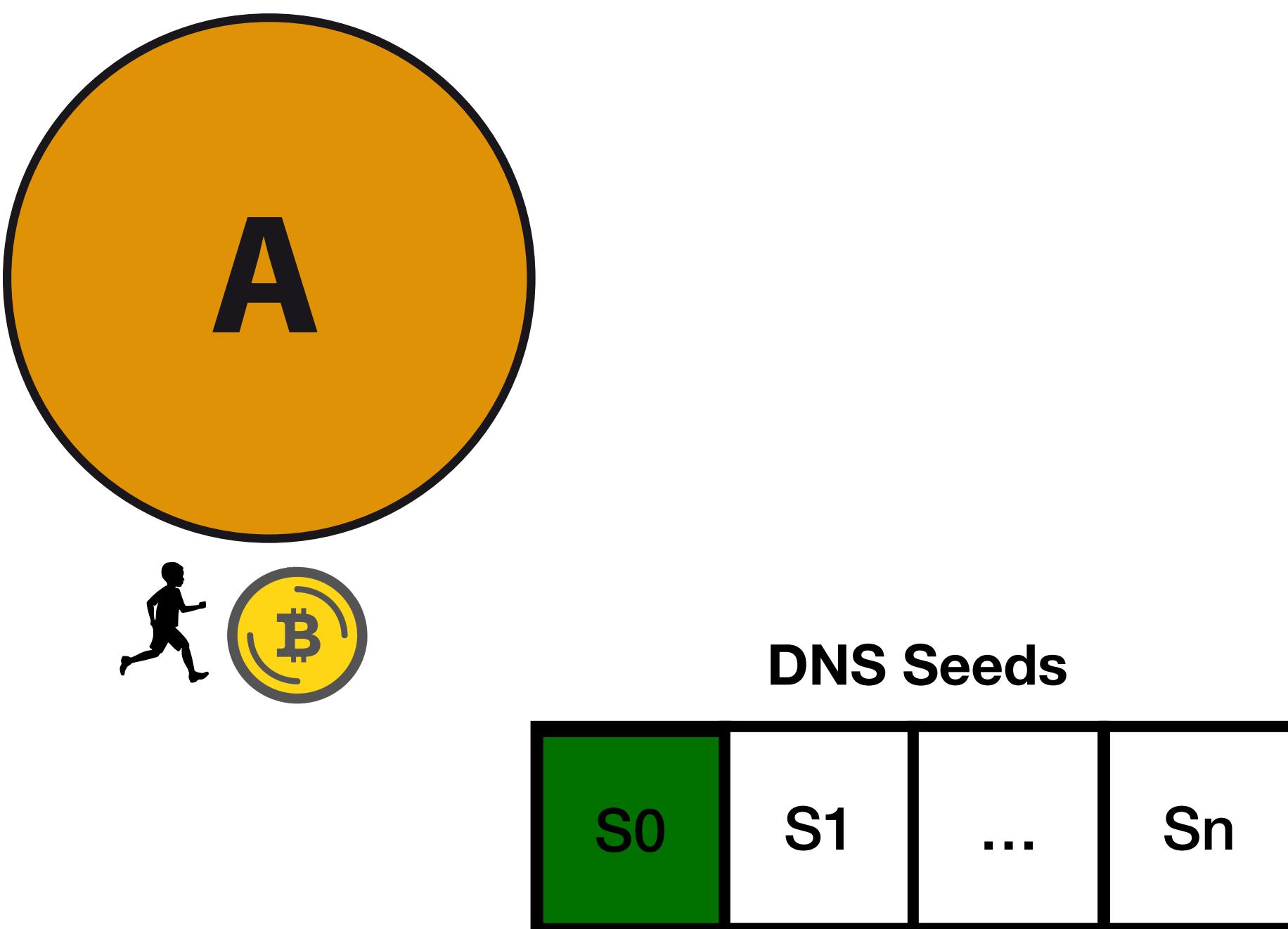


Bitcoin P2P bootstrapping (peer discovery)

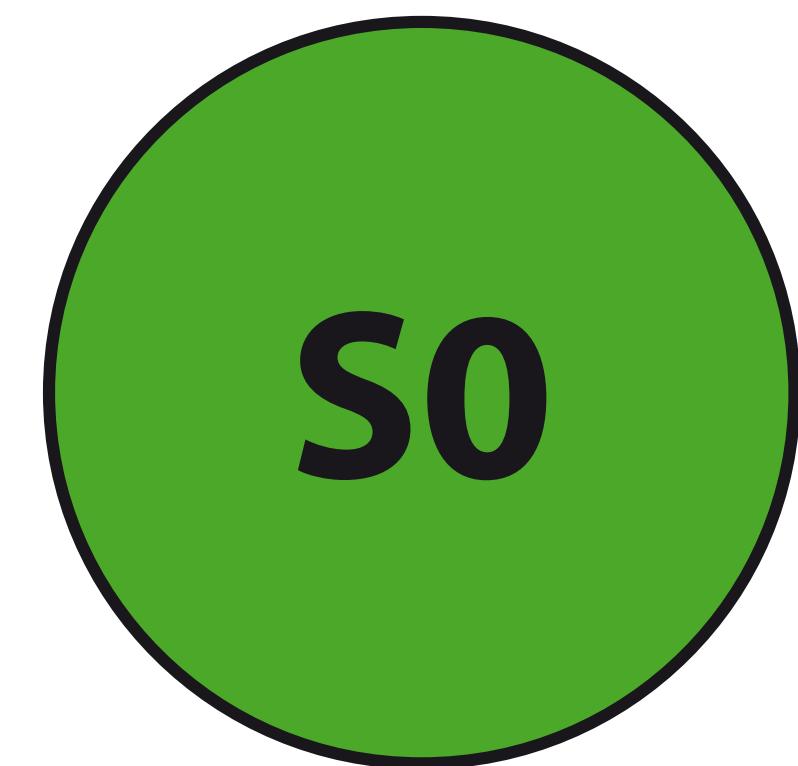
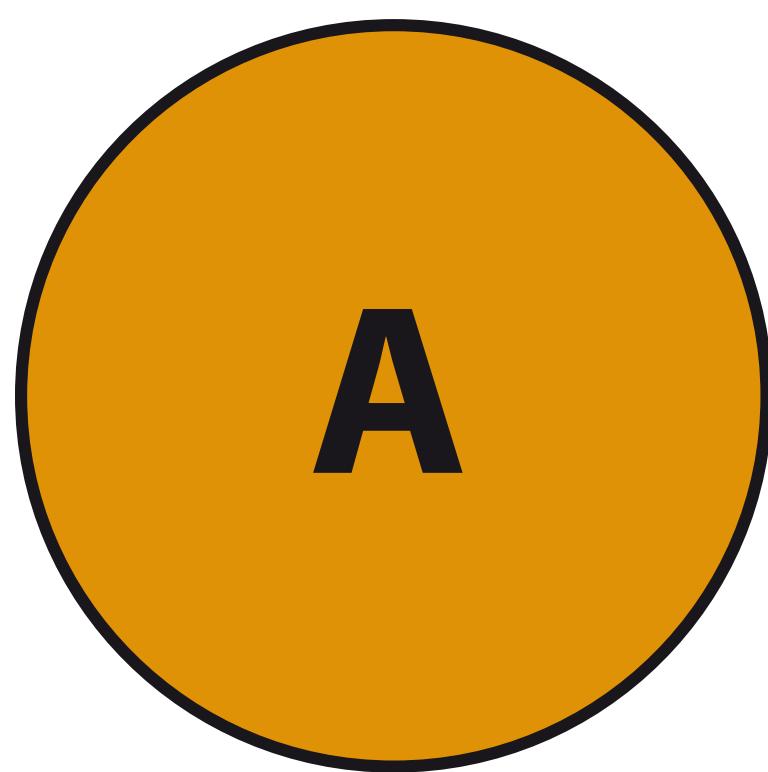




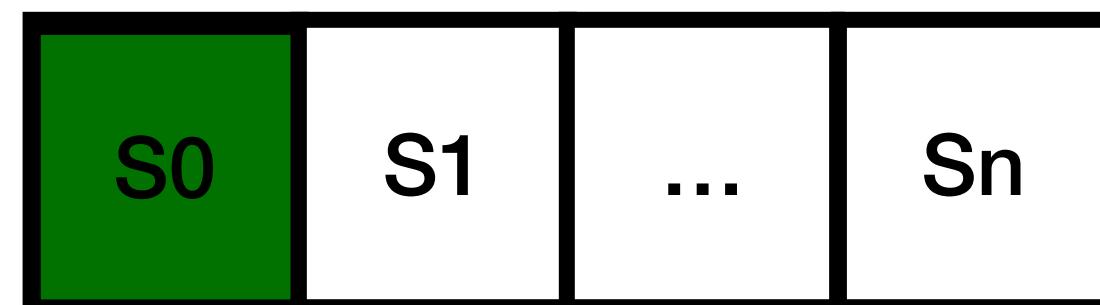
Bitcoin P2P bootstrapping (peer discovery)



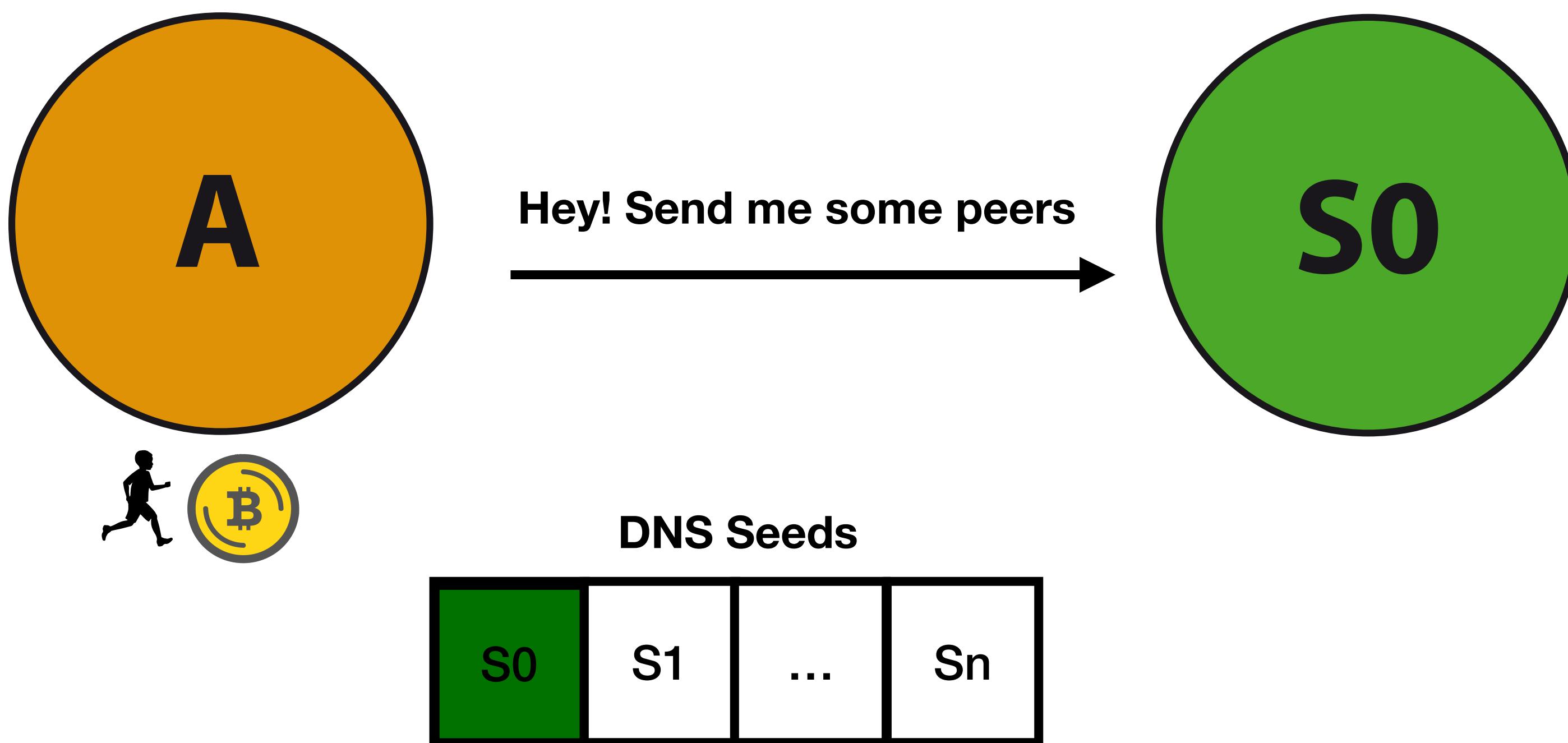
Bitcoin P2P bootstrapping (peer discovery)



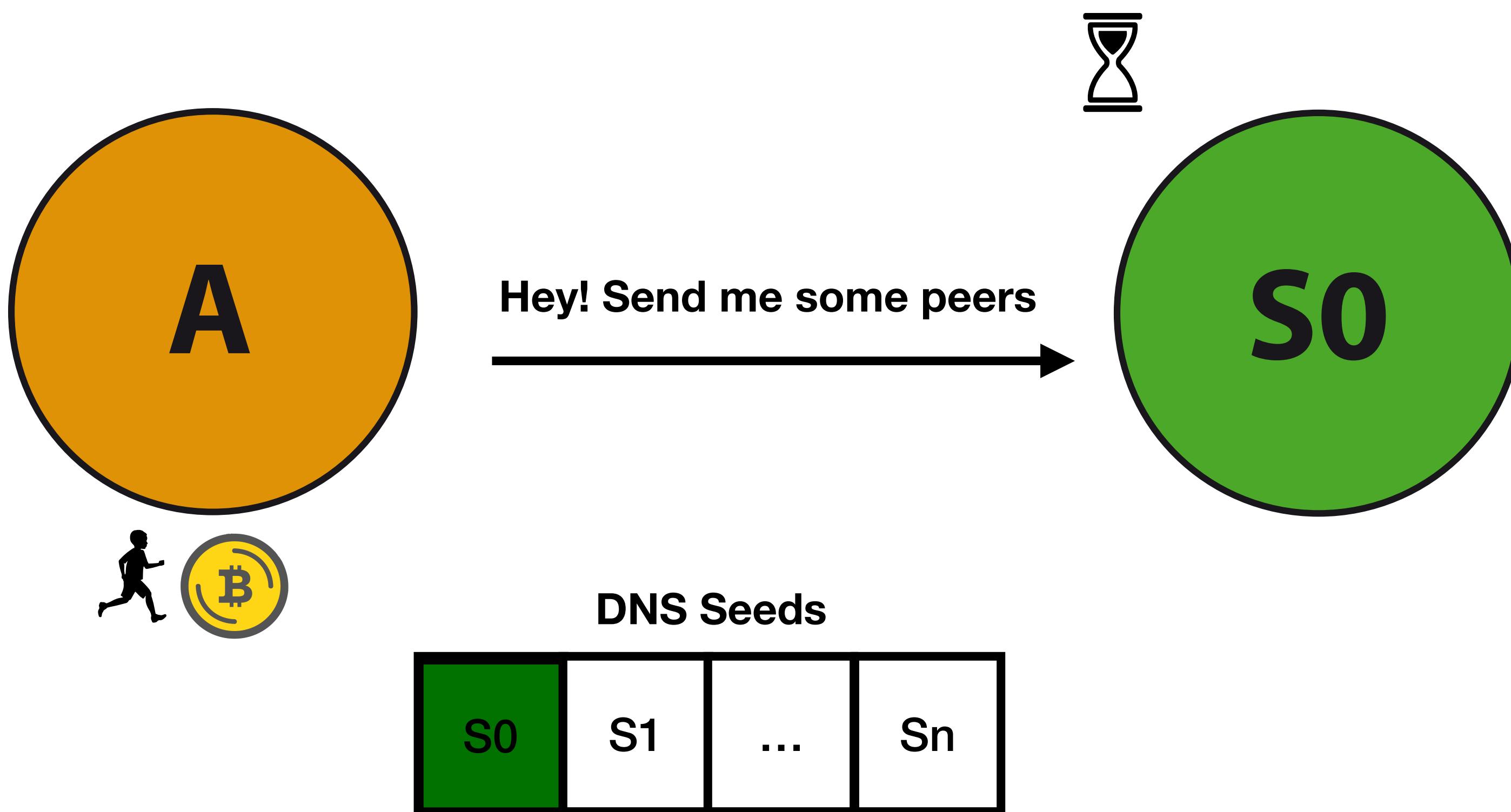
DNS Seeds



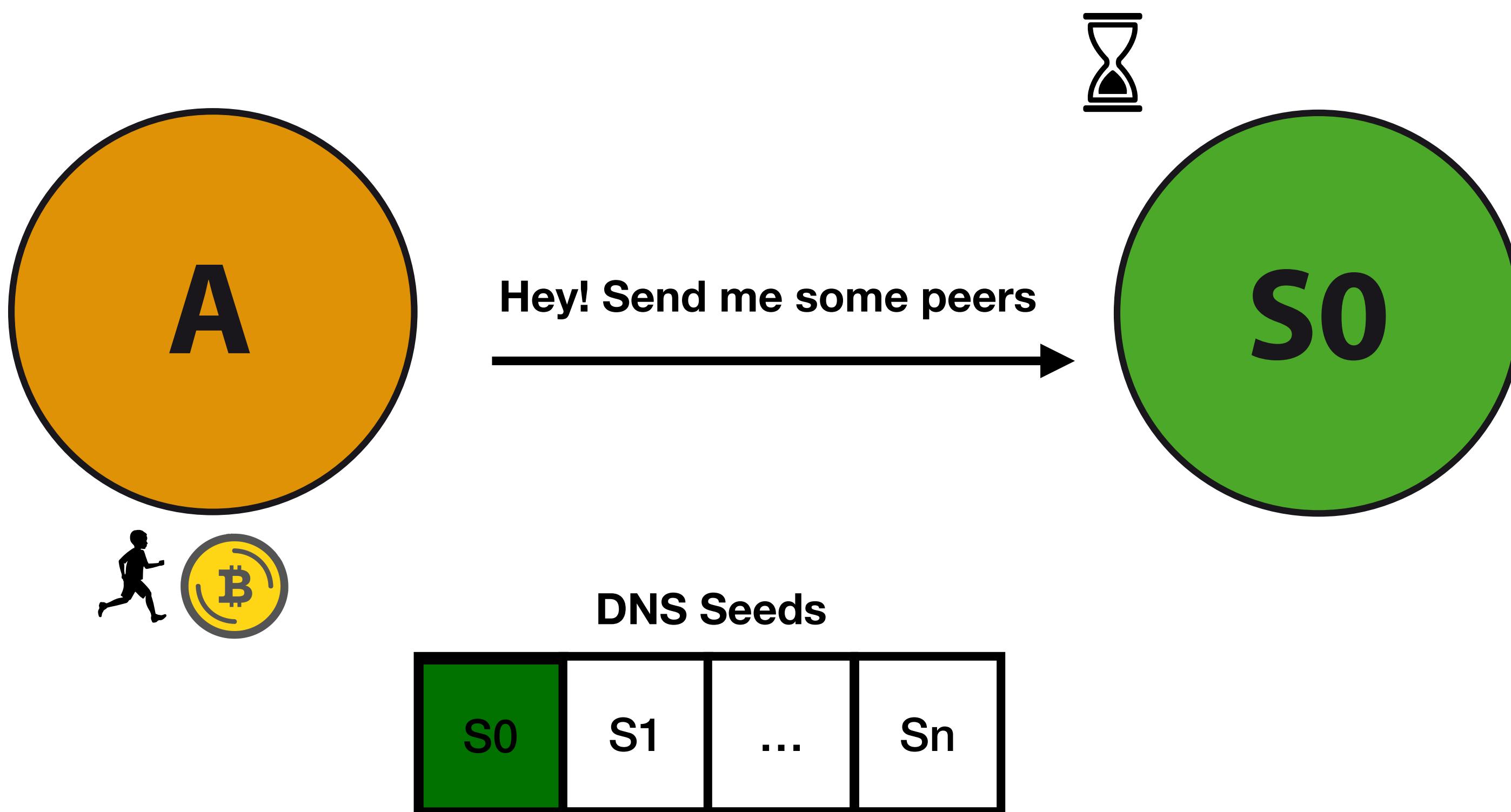
Bitcoin P2P bootstrapping (peer discovery)



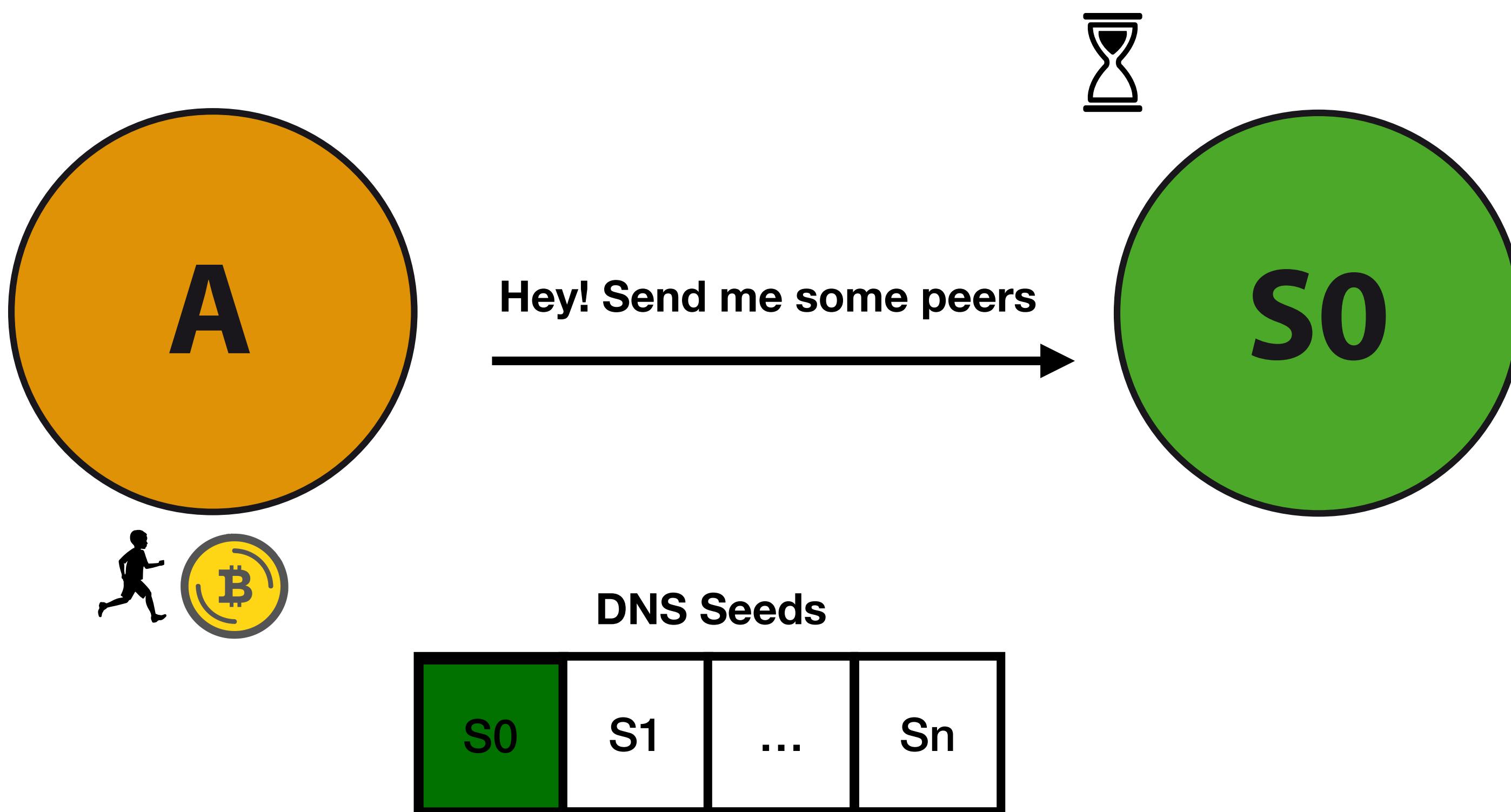
Bitcoin P2P bootstrapping (peer discovery)



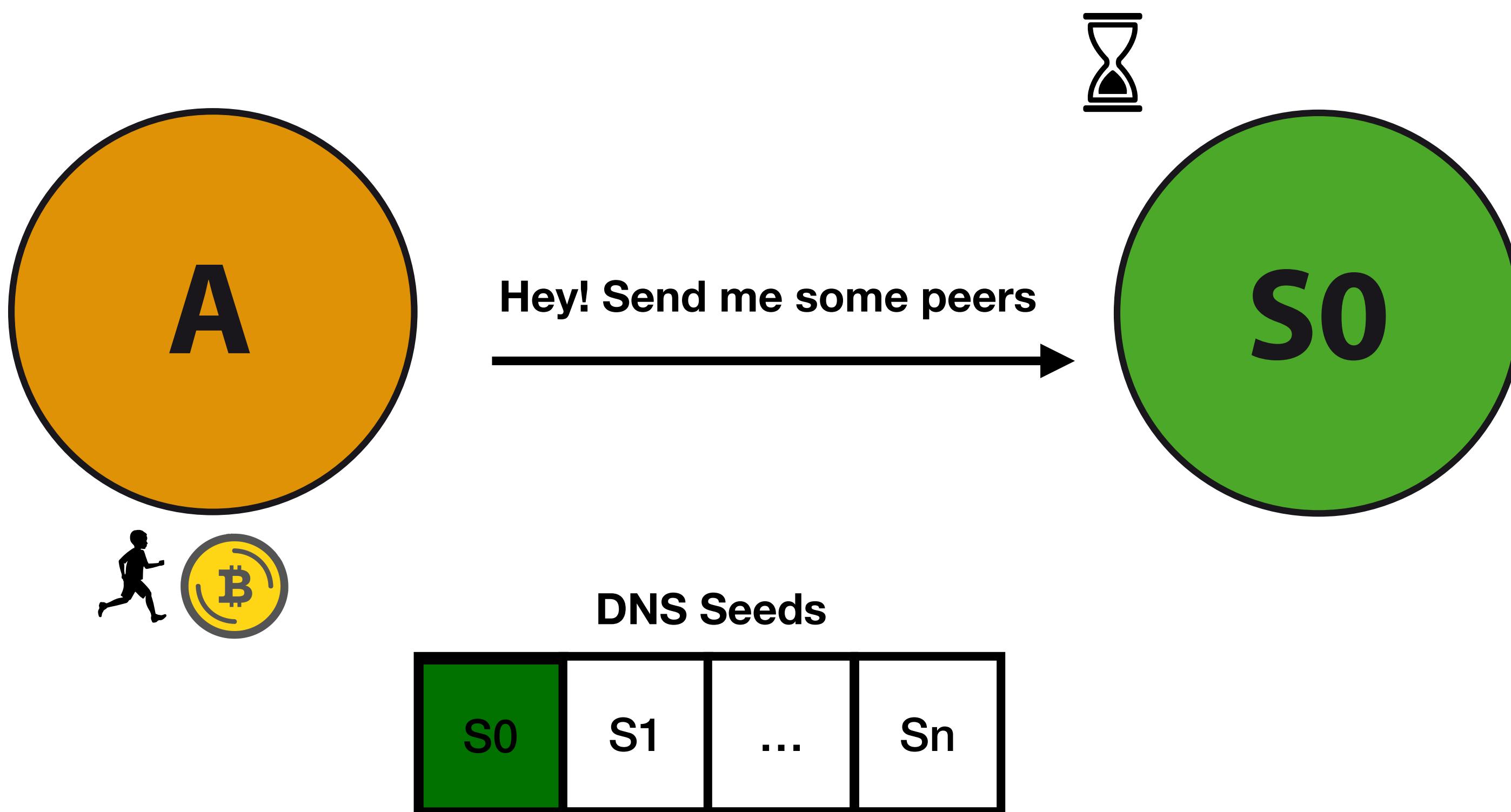
Bitcoin P2P bootstrapping (peer discovery)



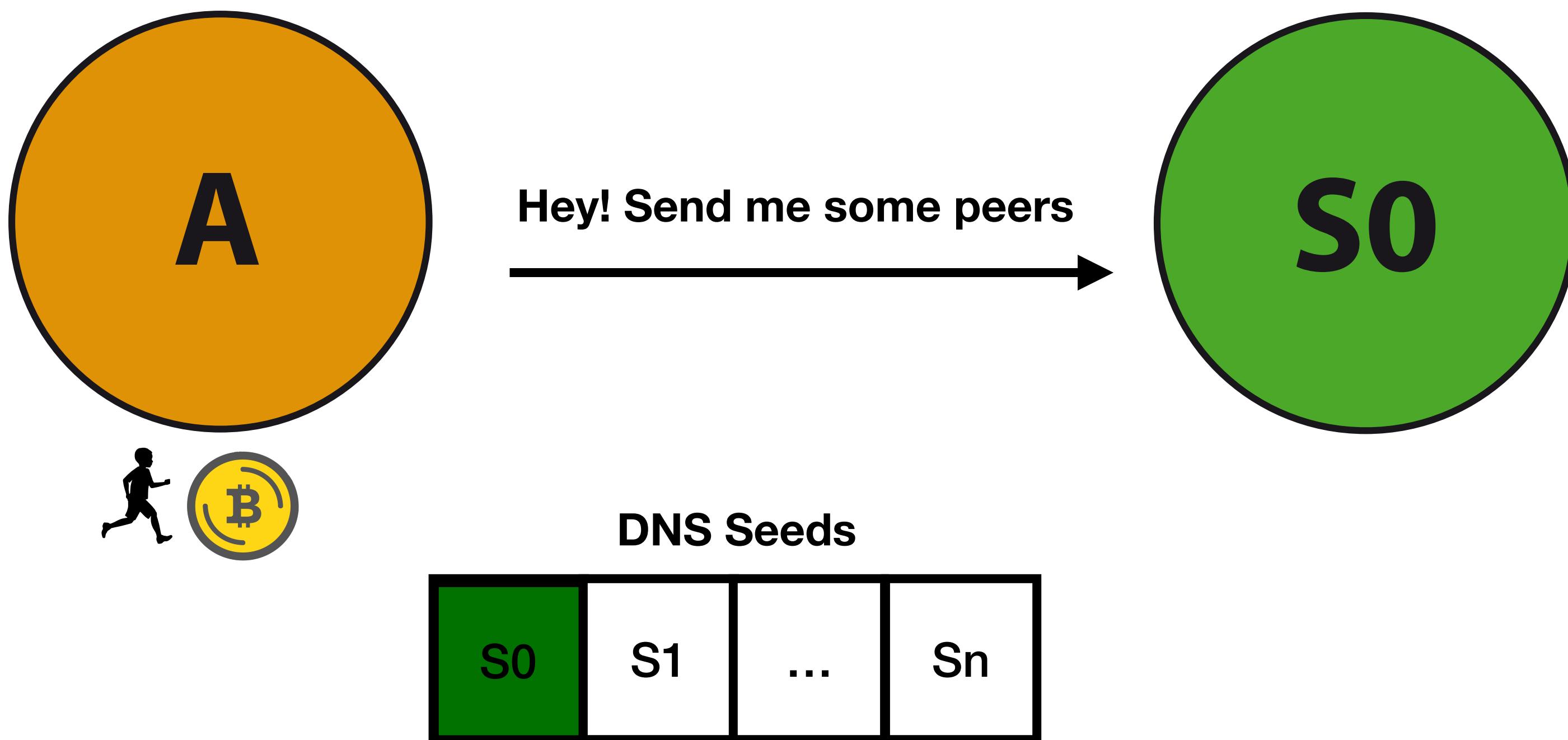
Bitcoin P2P bootstrapping (peer discovery)



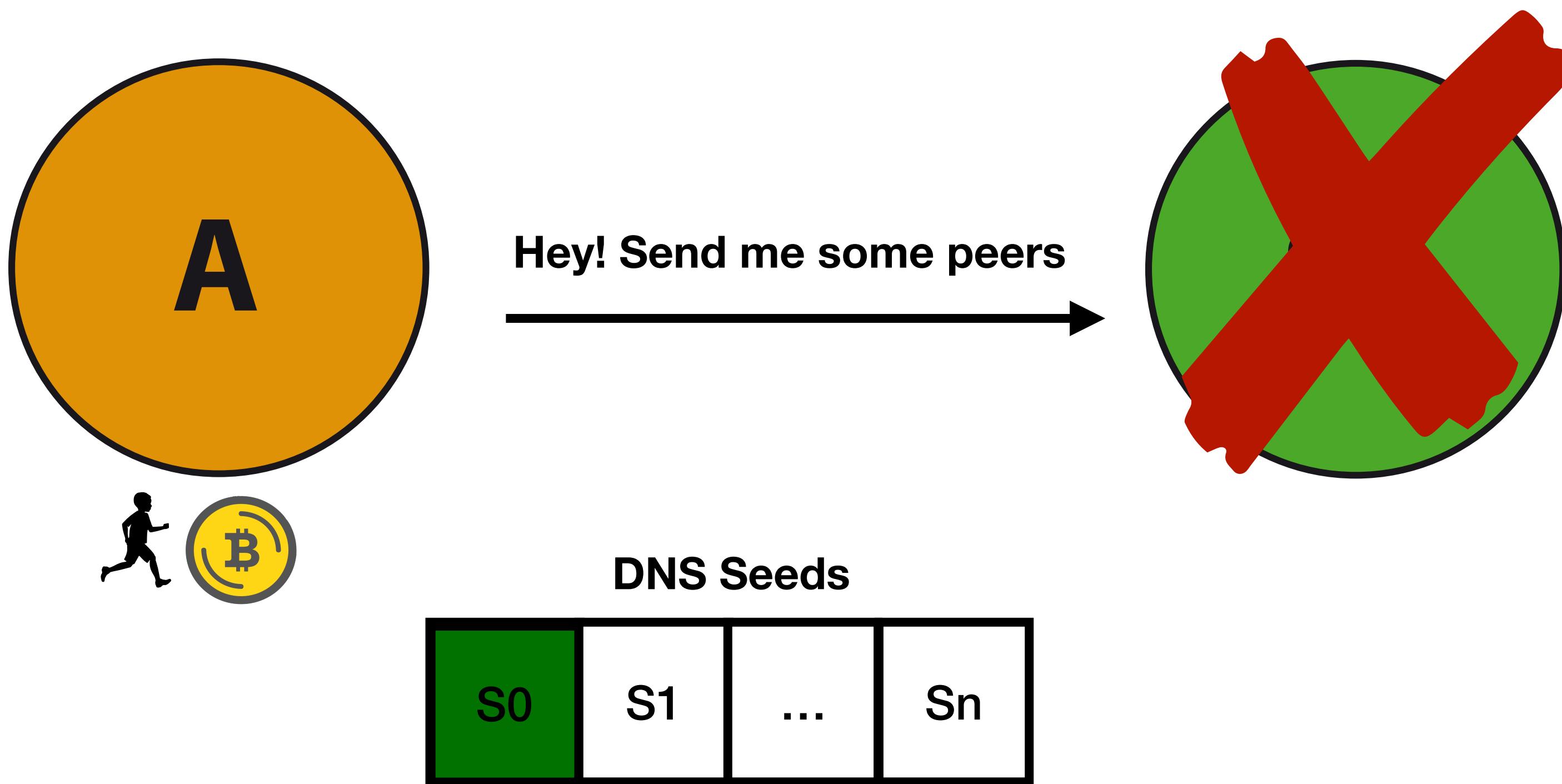
Bitcoin P2P bootstrapping (peer discovery)



Bitcoin P2P bootstrapping (peer discovery)

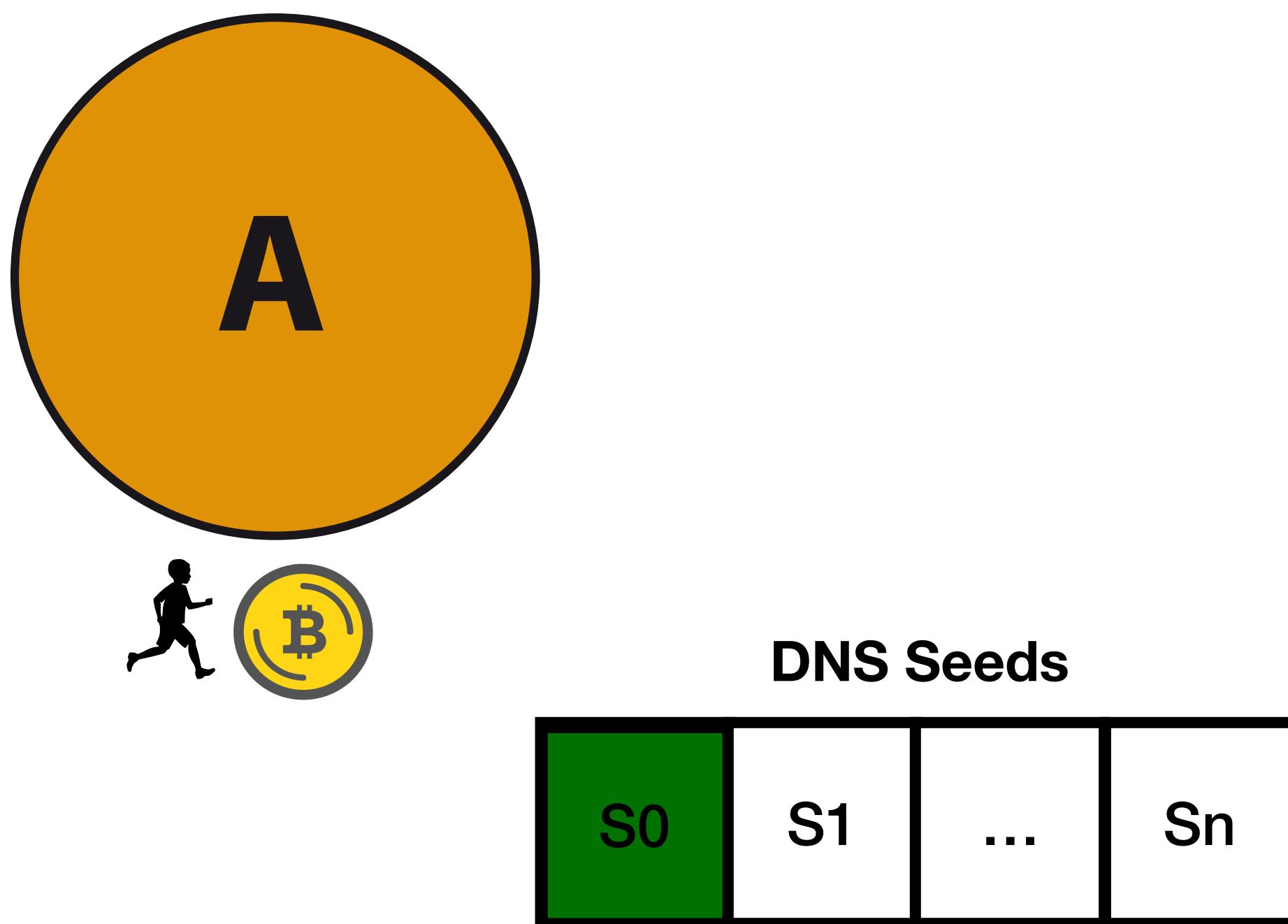


Bitcoin P2P bootstrapping (peer discovery)



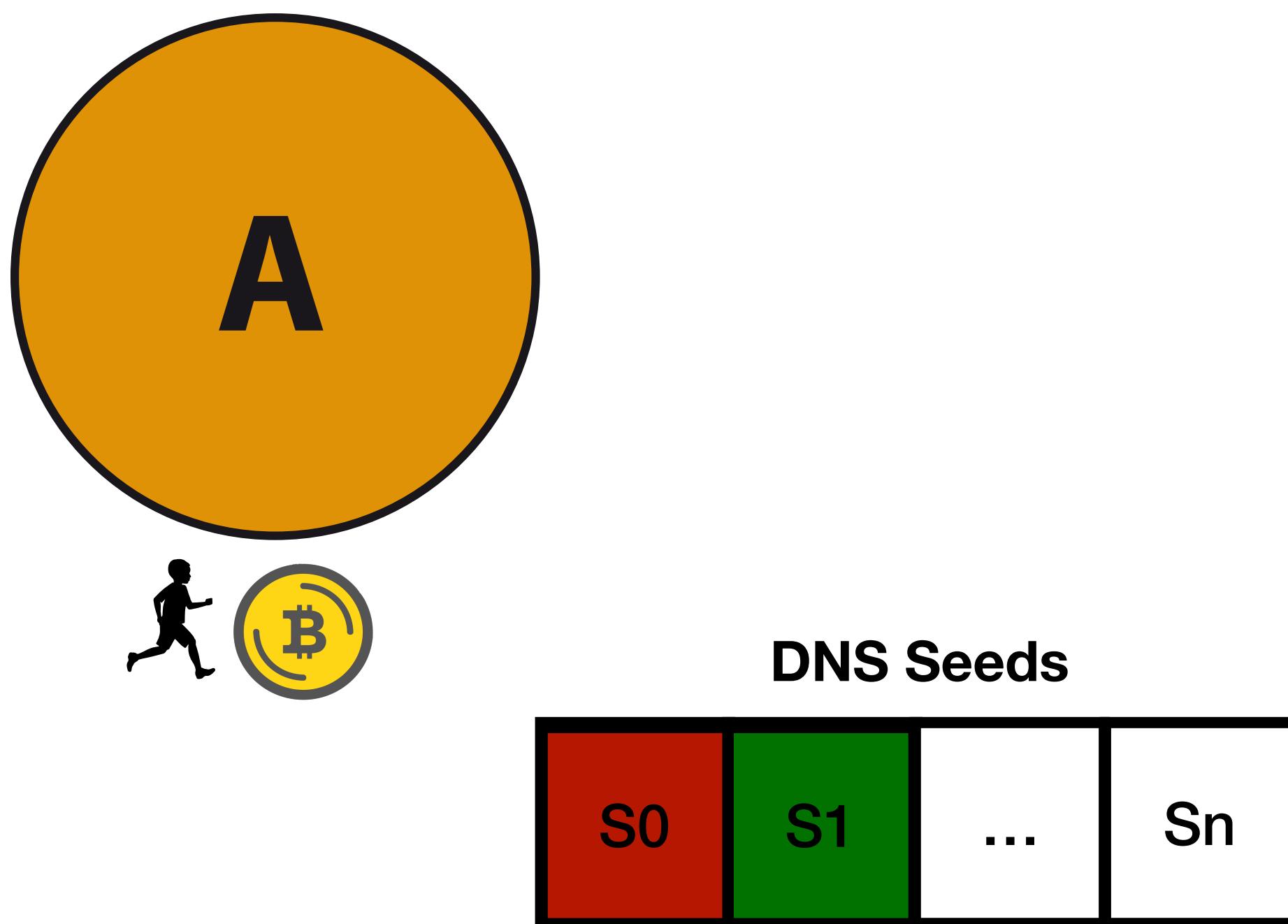


Bitcoin P2P bootstrapping (peer discovery)

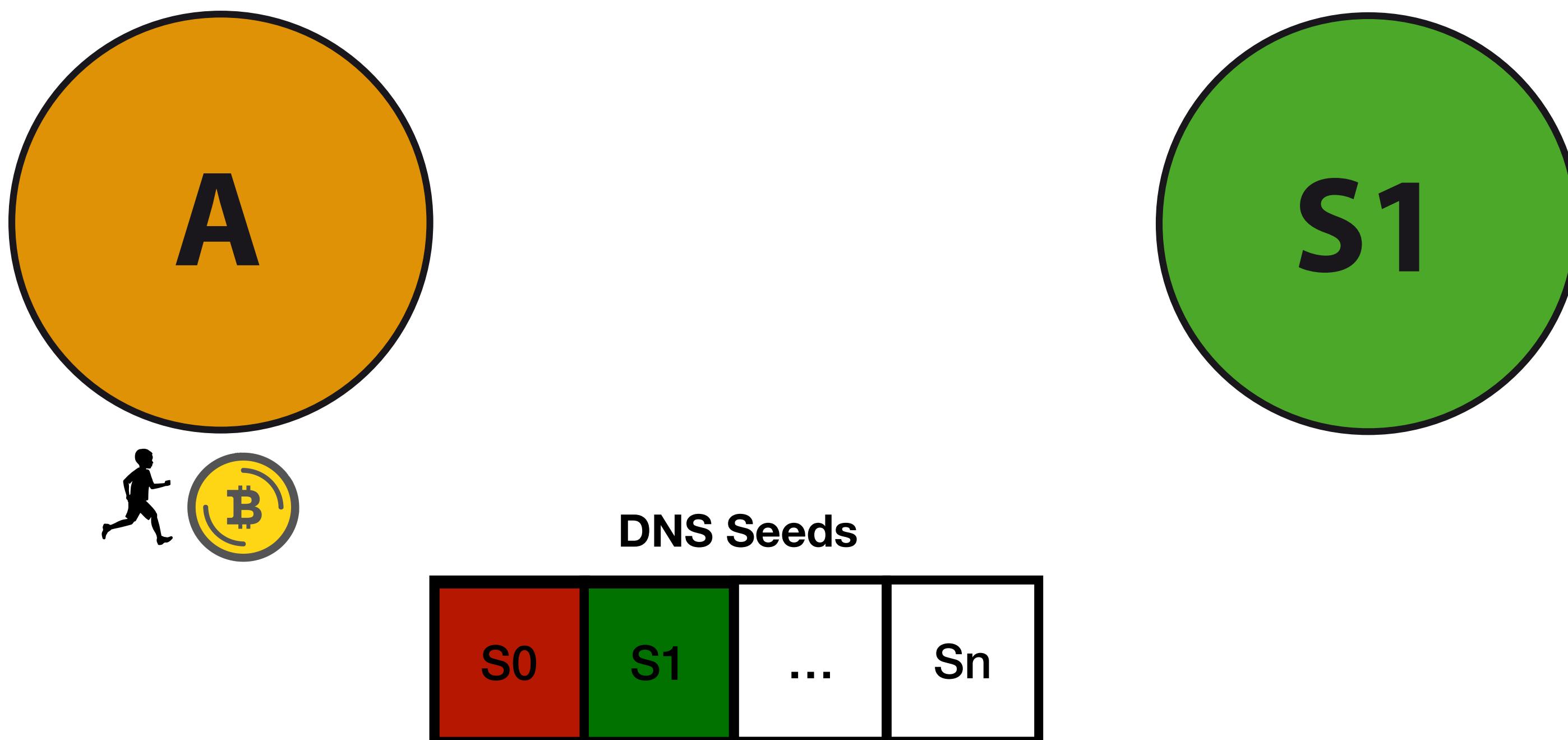




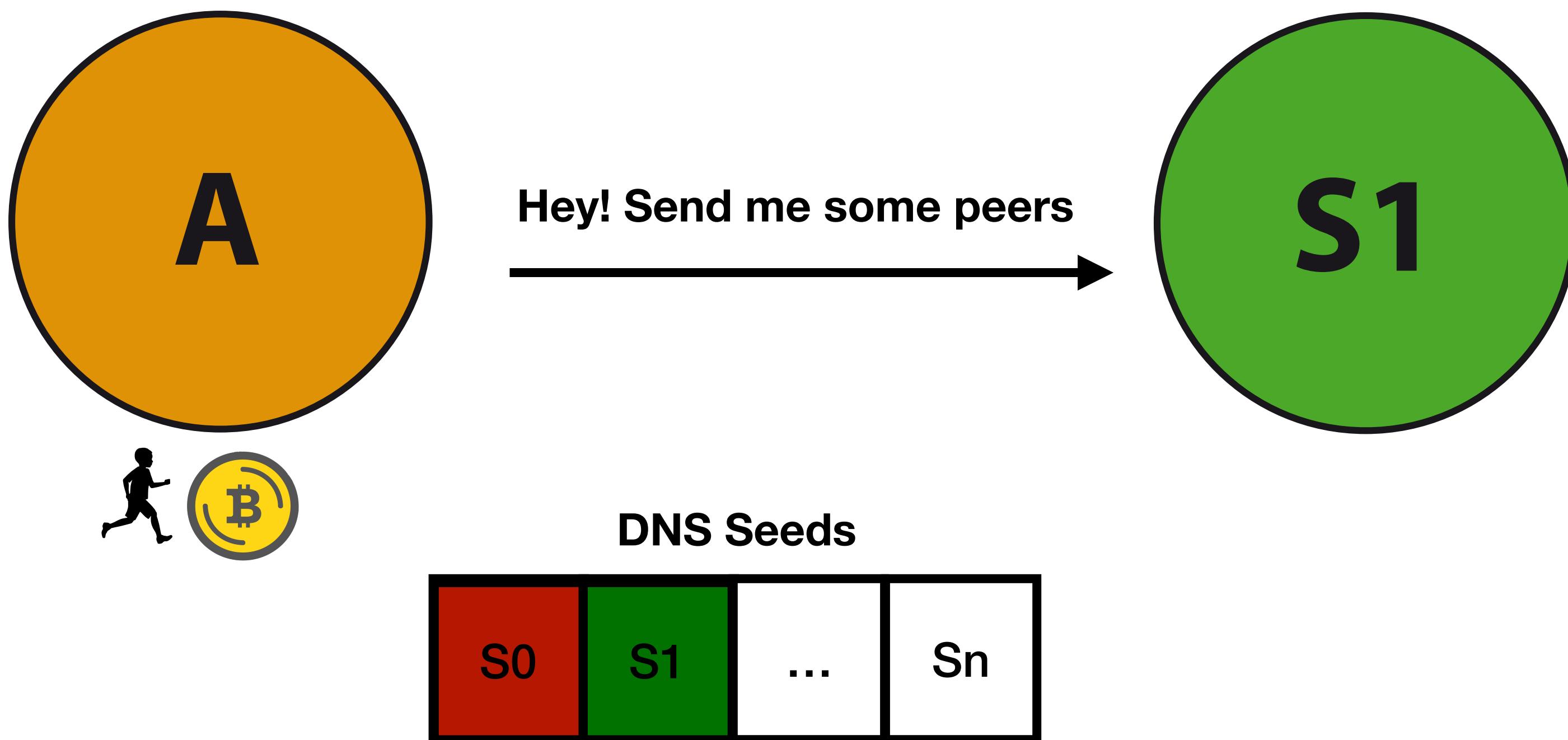
Bitcoin P2P bootstrapping (peer discovery)



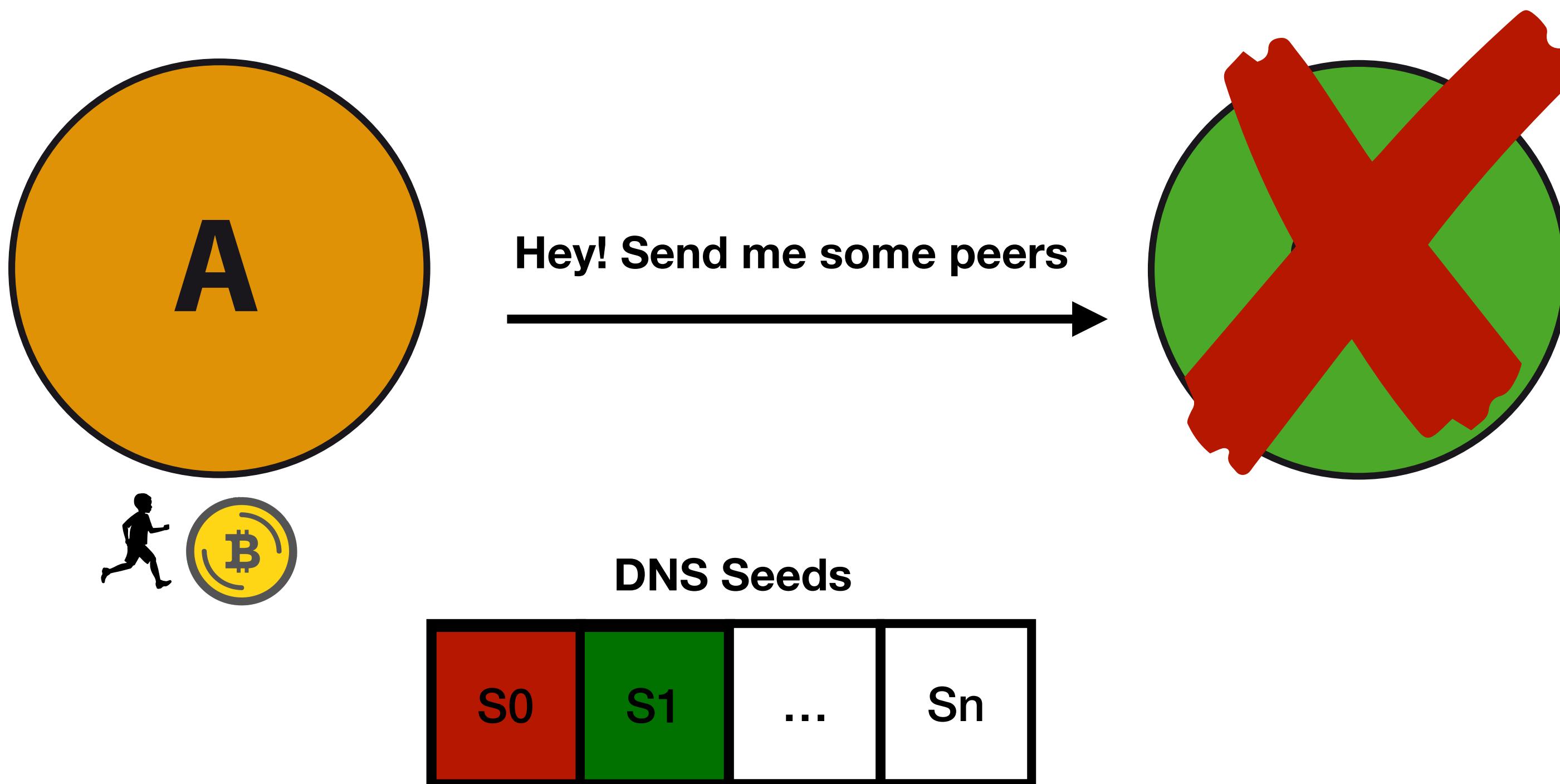
Bitcoin P2P bootstrapping (peer discovery)



Bitcoin P2P bootstrapping (peer discovery)

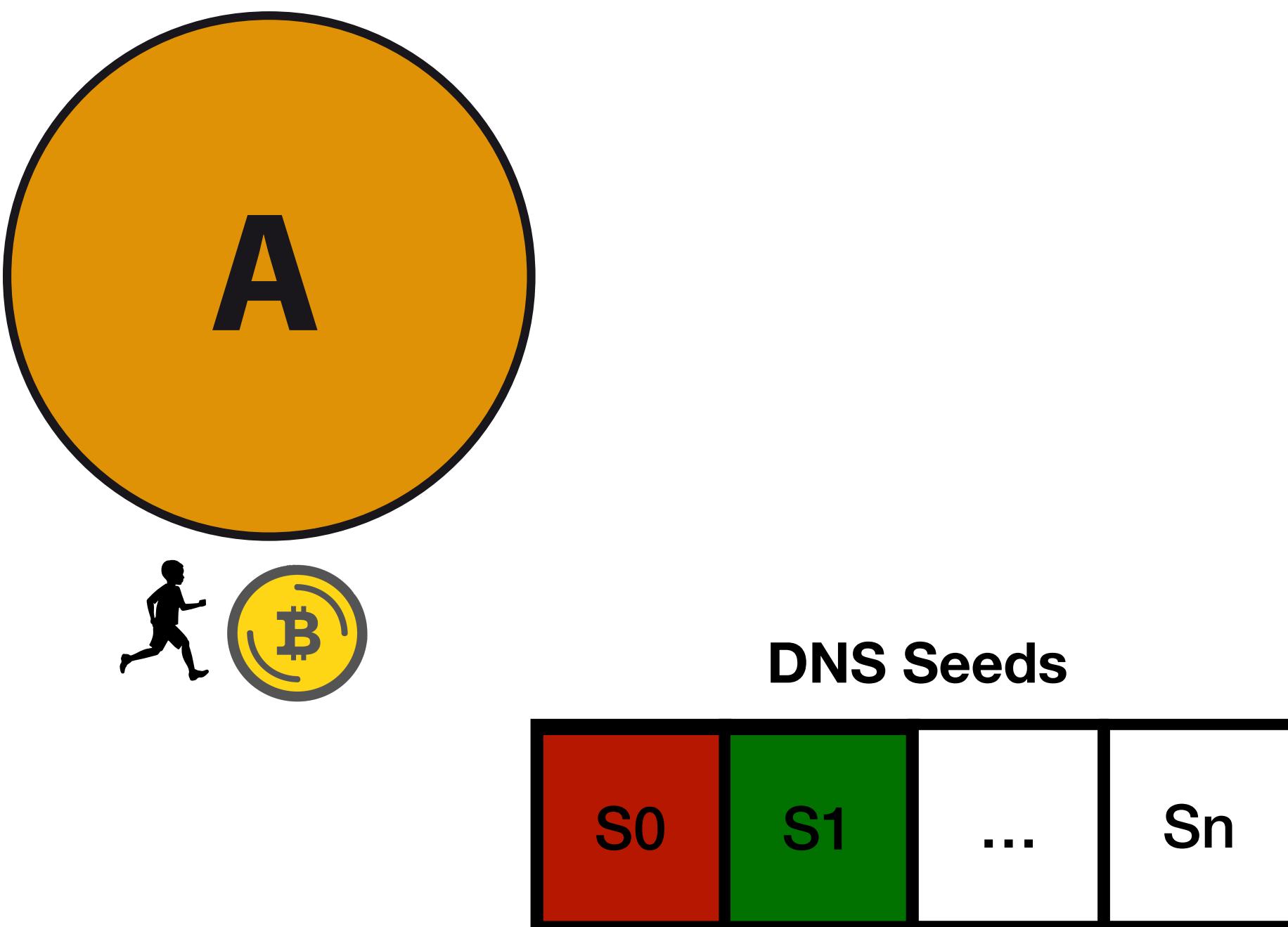


Bitcoin P2P bootstrapping (peer discovery)



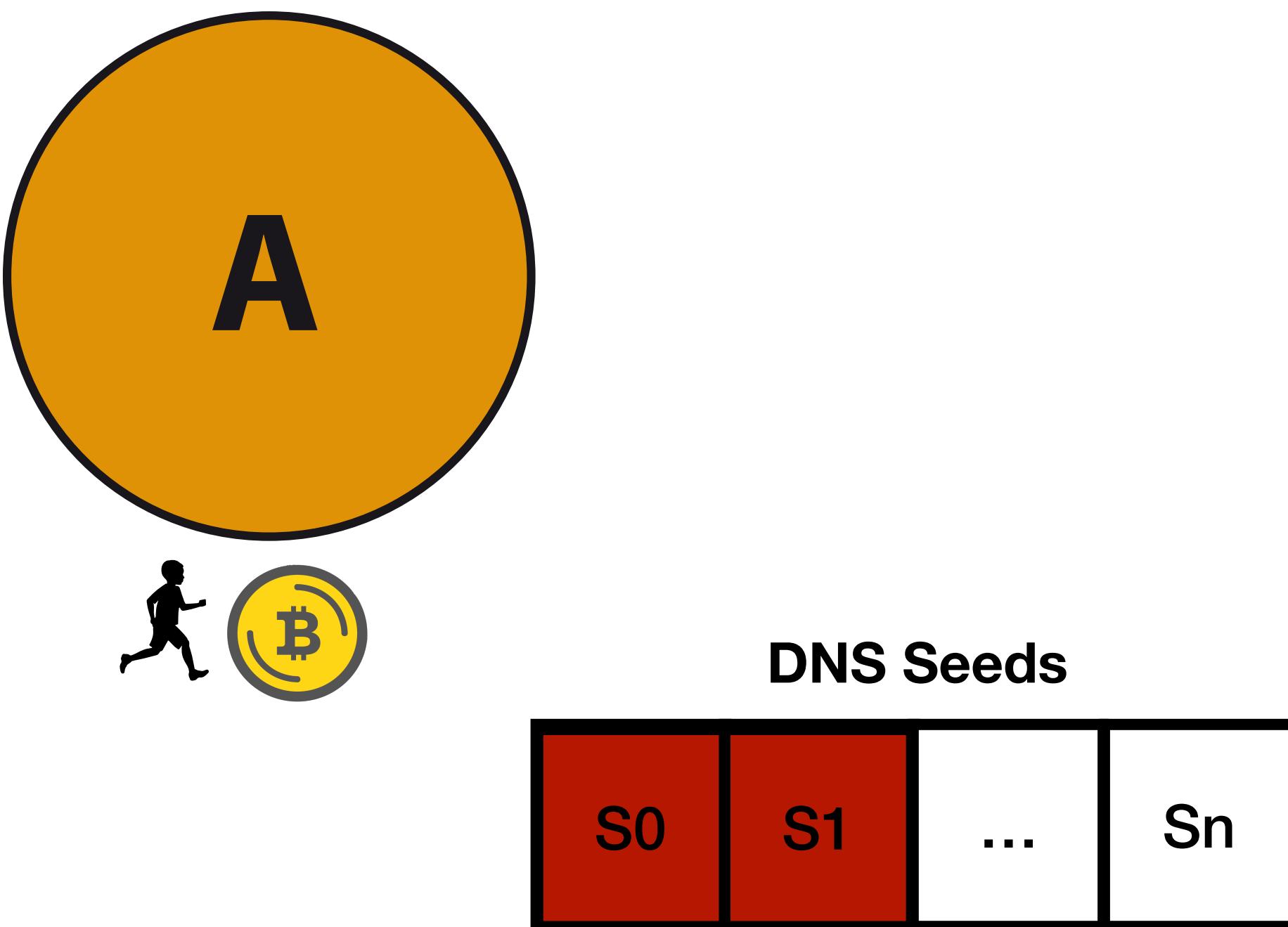


Bitcoin P2P bootstrapping (peer discovery)



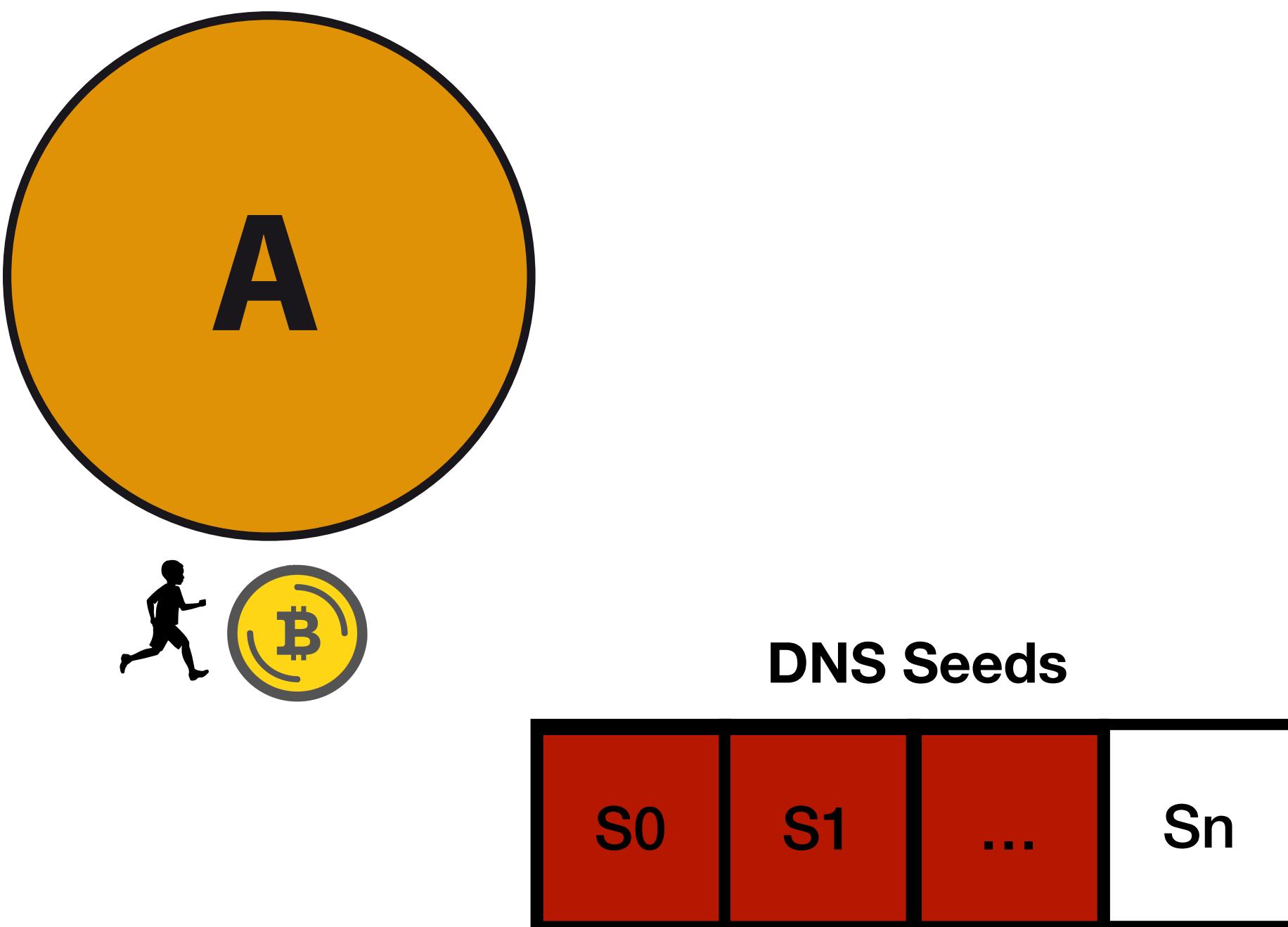


Bitcoin P2P bootstrapping (peer discovery)



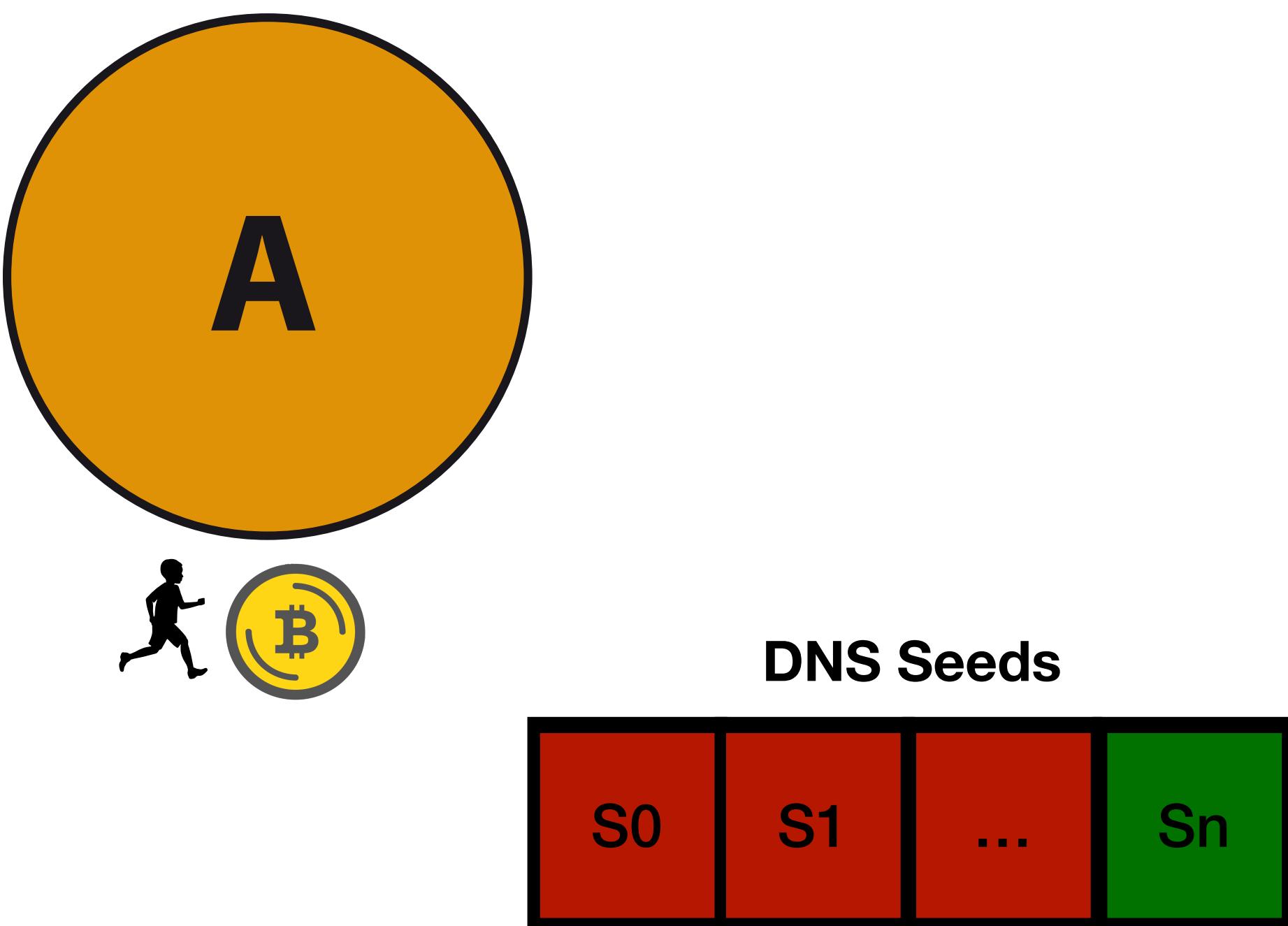


Bitcoin P2P bootstrapping (peer discovery)

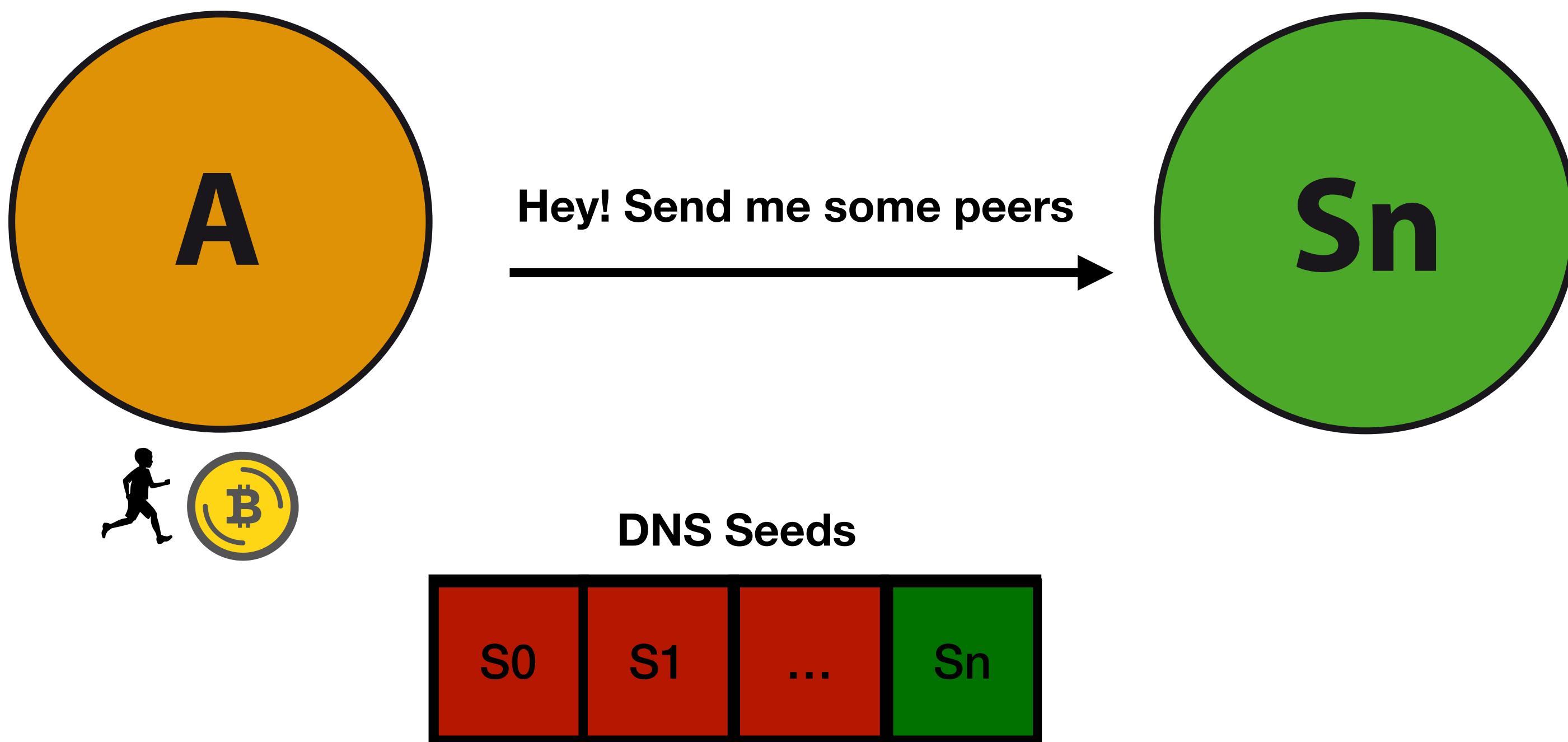




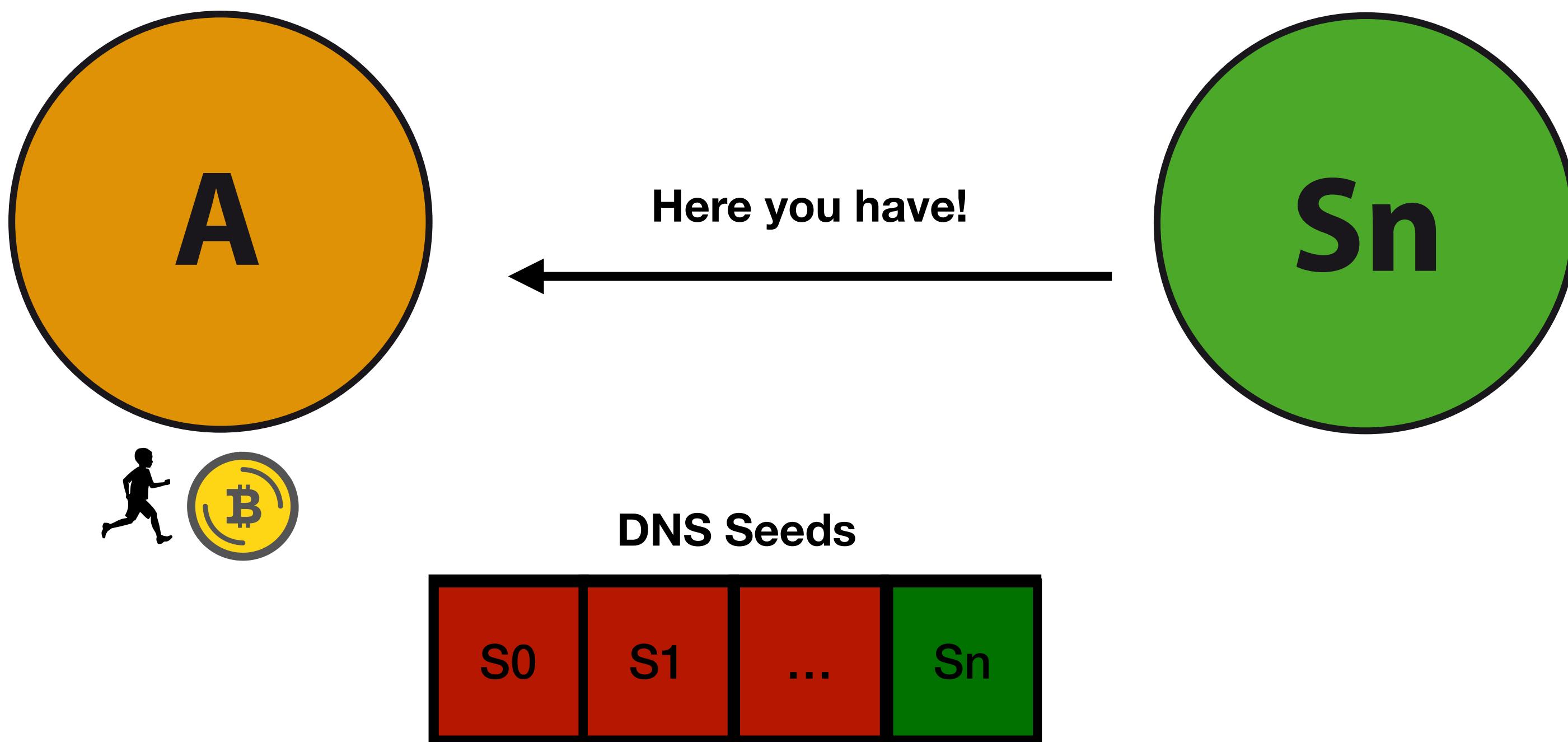
Bitcoin P2P bootstrapping (peer discovery)



Bitcoin P2P bootstrapping (peer discovery)



Bitcoin P2P bootstrapping (peer discovery)

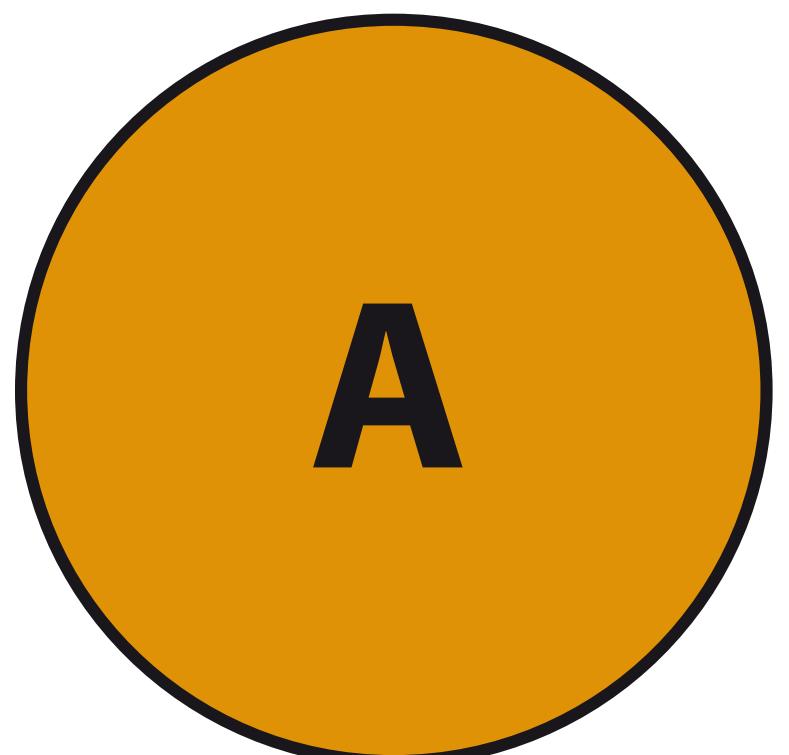




Bitcoin P2P bootstrapping (peer discovery)



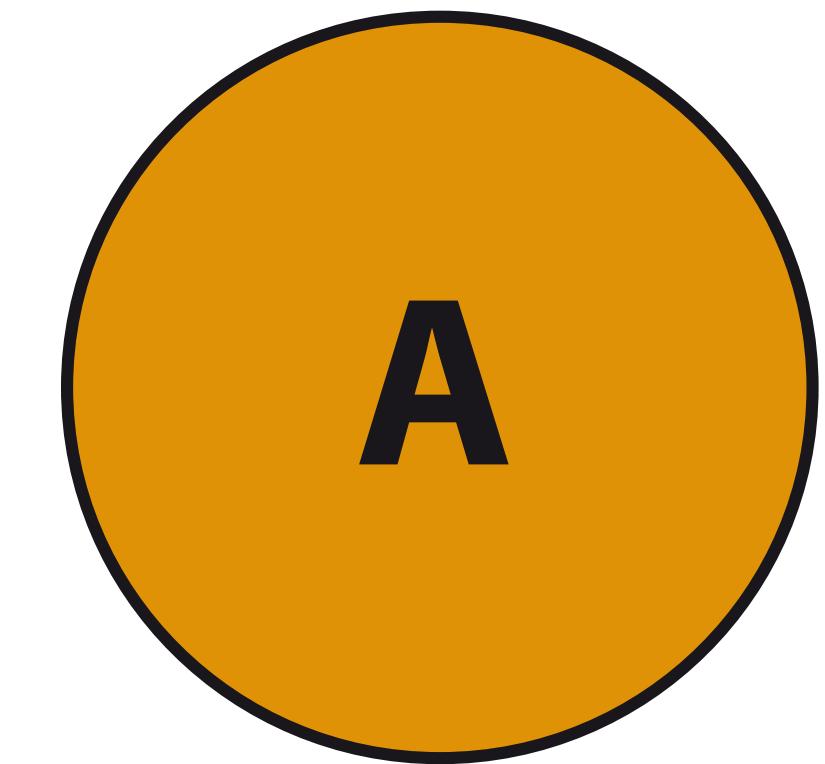
Bitcoin P2P bootstrapping (peer discovery)



Peer list (addrman)

P0	P1	...	Pn
----	----	-----	----

Bitcoin P2P bootstrapping (peer discovery)



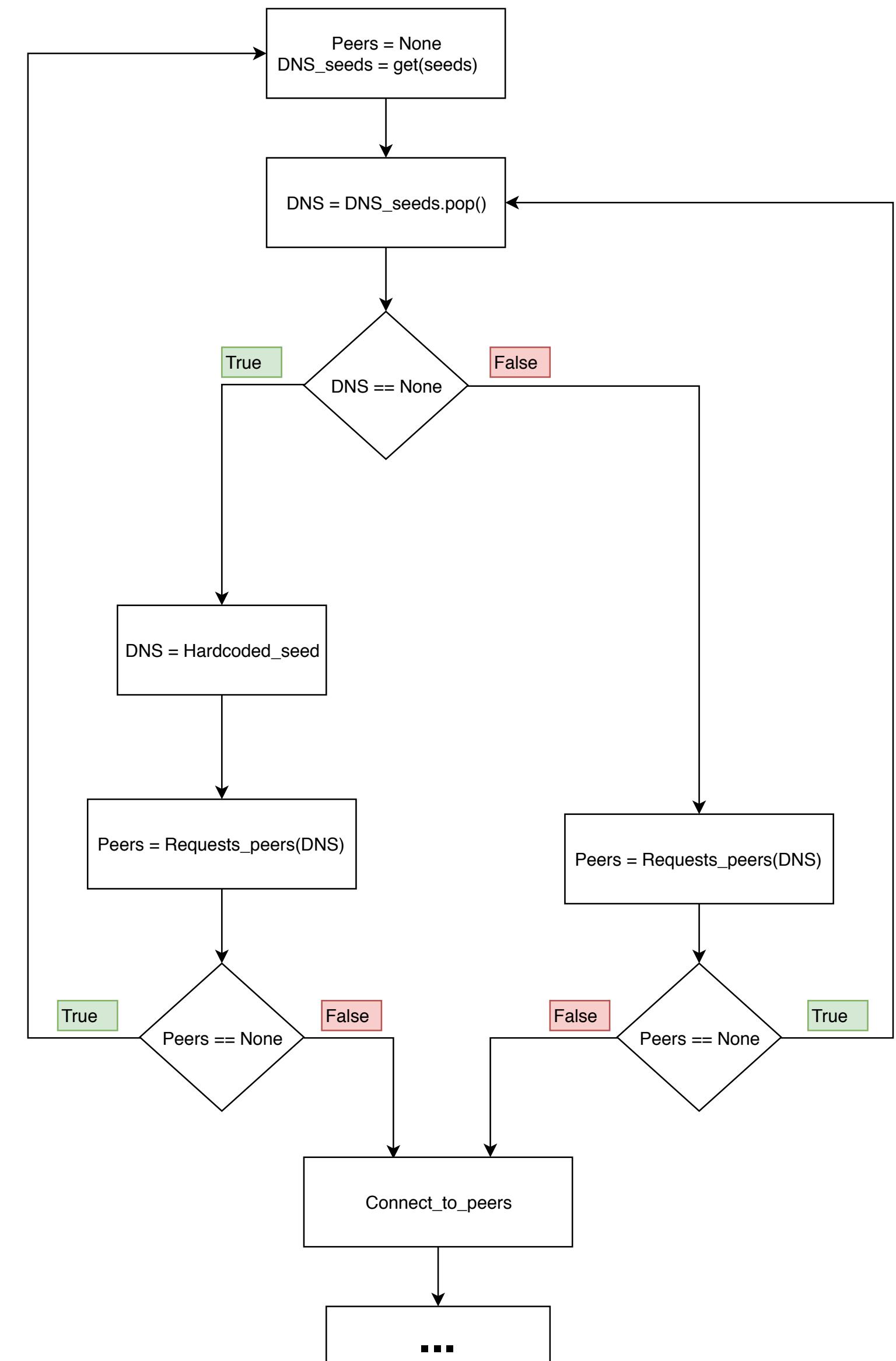
Peer list (addrman)

P0	P1	...	Pn
----	----	-----	----

- When a node receives a request for peers, it replies with randomly distributed subset of peers it knows about (**up to 1000**)
- Upon receiving some addresses, a bootstrapping node will store them and try to connect to a subset of his known peers (**8 by default**)
- A node will also accept some incoming connections (**117 by default**)

Bitcoin P2P bootstrapping (summary)

- A node bootstraps with no known peers
- First it tries to query a list of well known DNS seeds
- **As last resource it uses a hardcoded seed**

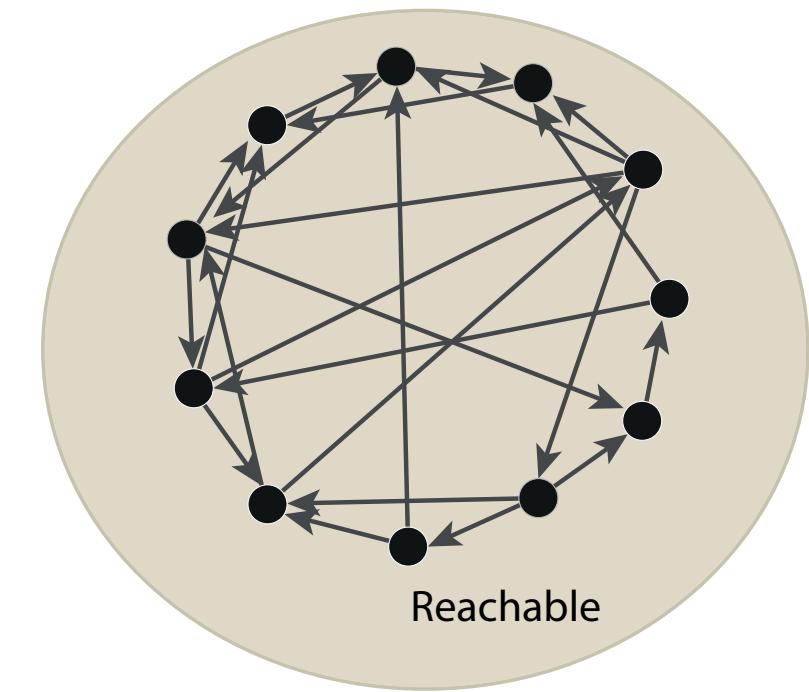




Cryptocurrency networks taxonomy

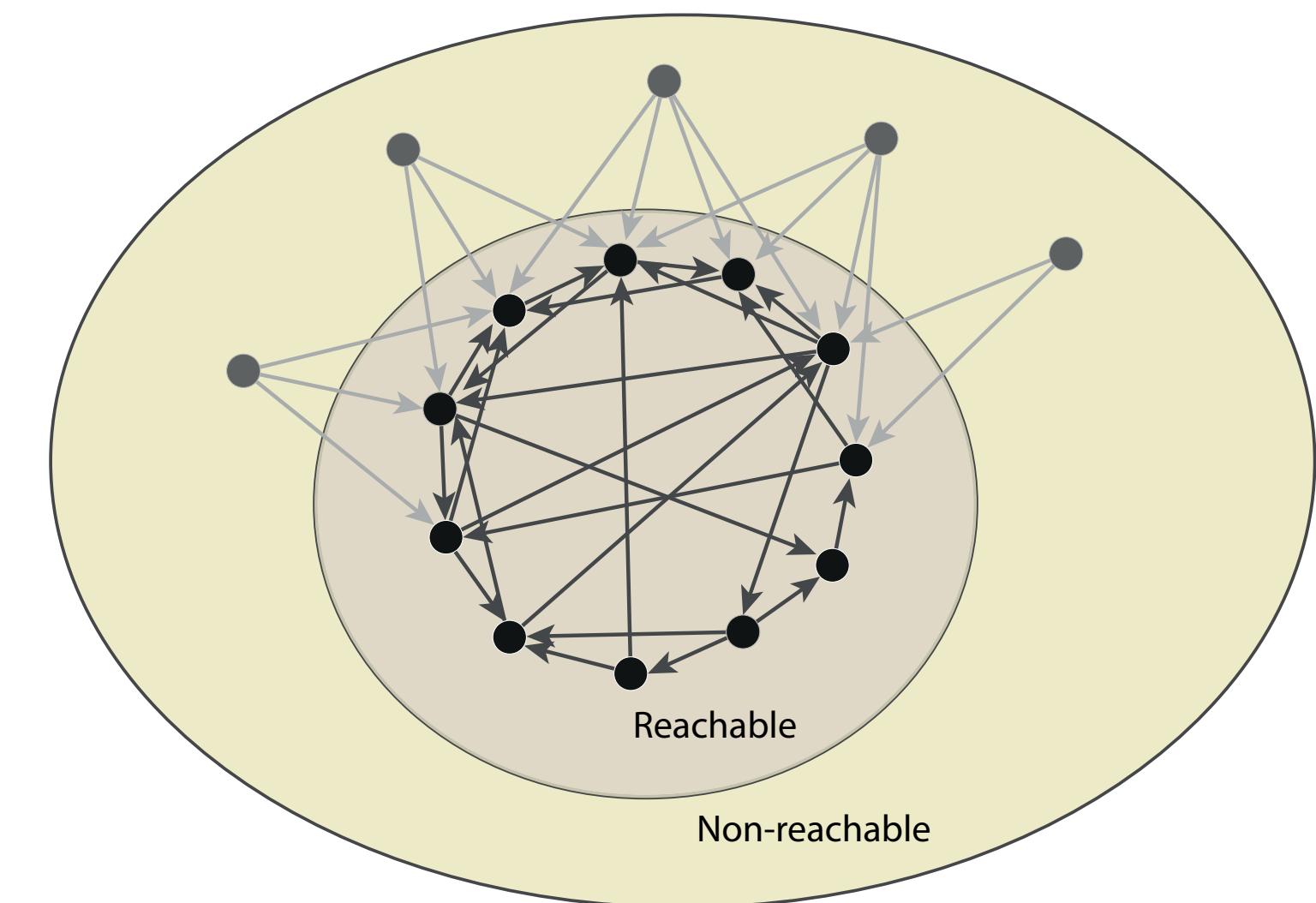
Cryptocurrency networks taxonomy

- Reachable network: all nodes accept incoming / outgoing connections



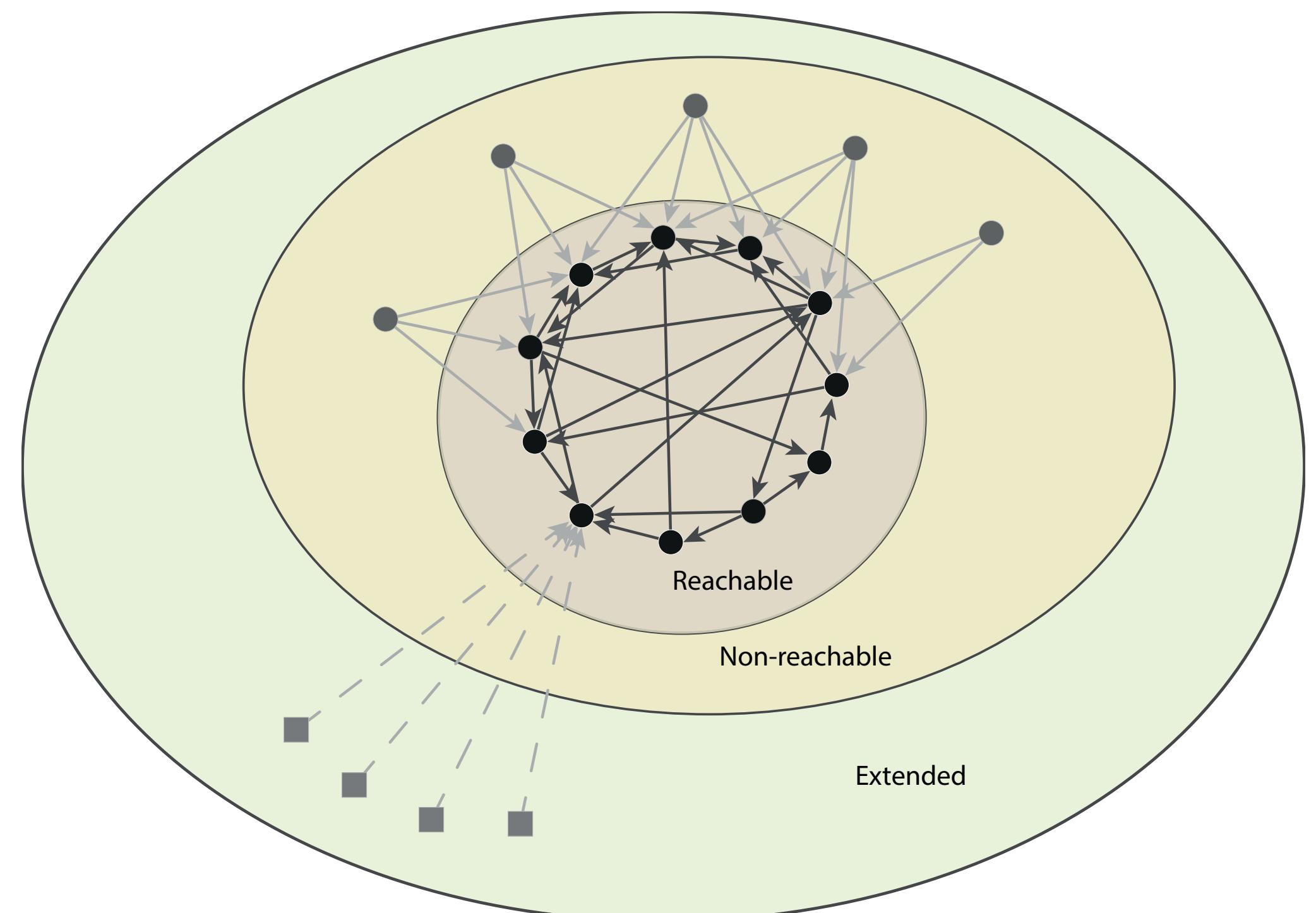
Cryptocurrency networks taxonomy

- Reachable network: all nodes accept incoming / outgoing connections
- Non-reachable: nodes do not accept incoming connections / cannot be reached (NAT/firewalls/...)



Cryptocurrency networks taxonomy

- Reachable network: all nodes accept incoming / outgoing connections
- Non-reachable: nodes do not accept incoming connections / cannot be reached (NAT/firewalls/...)
- Extended network: nodes use different protocol to communicate (not always P2P)



Cryptocurrency P2P 101



The data (what?)

- There are two main items that peers share in a cryptocurrency P2P network: **transactions** and **blocks**

The data (what?)

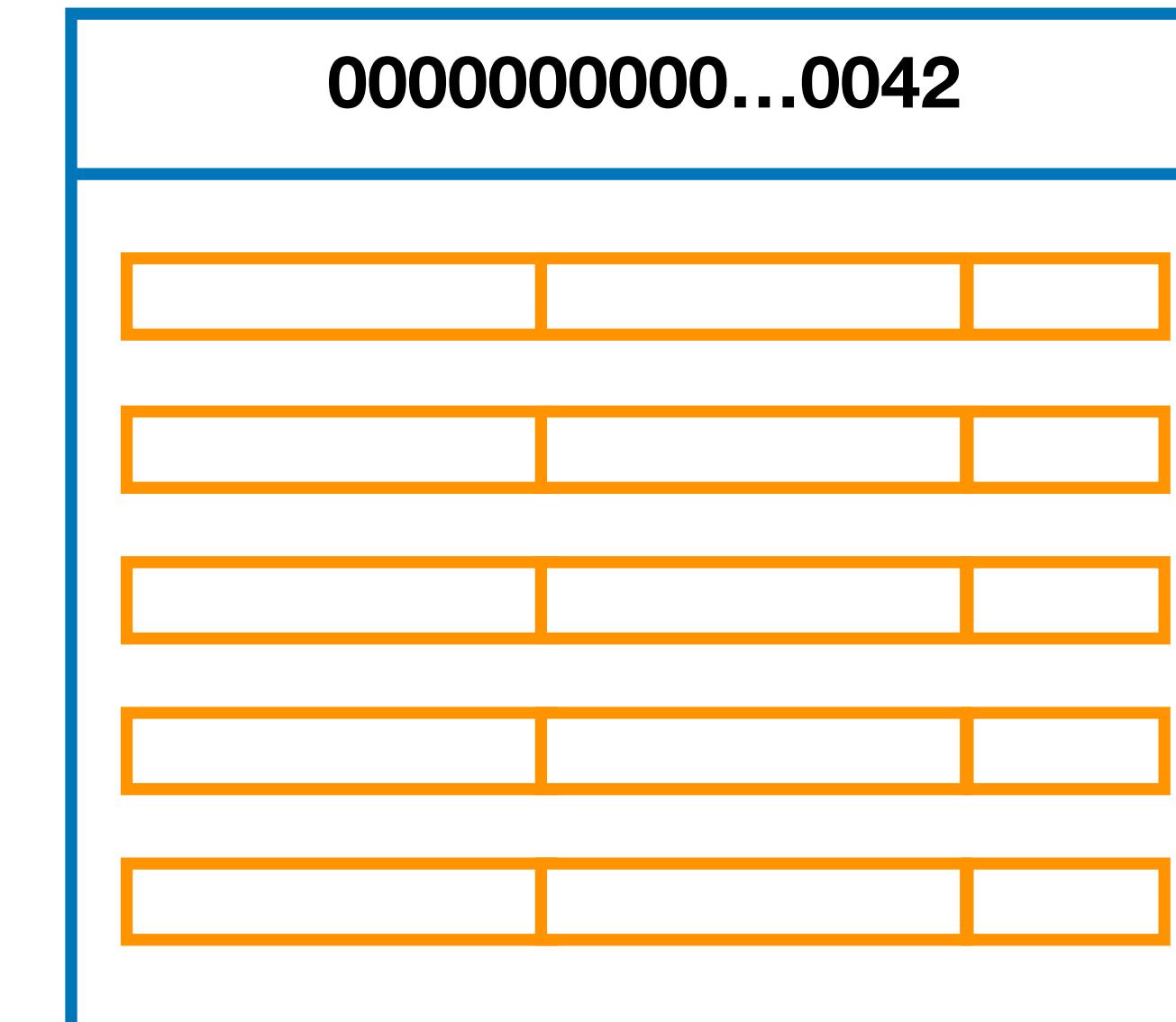
- There are two main items that peers share in a cryptocurrency P2P network: **transactions** and **blocks**

From: Alice	To: Bob	5
From: Alice	To: Carol	7
From: Carol	To: Dave	2
From: Dave	To: Bob	1

The data (what?)

- There are two main items that peers share in a cryptocurrency P2P network: **transactions** and **blocks**

From: Alice	To: Bob	5
From: Alice	To: Carol	7
From: Carol	To: Dave	2
From: Dave	To: Bob	1





The actors (who?) (1/2)



The actors (who?) (1/2)

- There are two main roles followed by nodes:
peers and **miners**



The actors (who?) (1/2)

- There are two main roles followed by nodes:
peers and **miners**
- (Normal) Peers:

The actors (who?) (1/2)

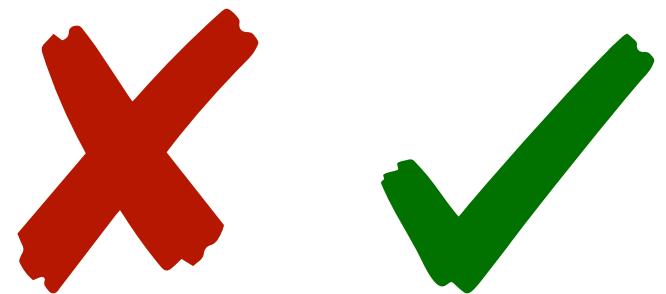
- There are two main roles followed by nodes:
peers and **miners**
- (Normal) Peers:
 - Can **create transactions** that spend some of their bitcoins

From: Alice	To: Bob	5
-------------	---------	---

The actors (who?) (1/2)

- There are two main roles followed by nodes:
peers and **miners**
- (Normal) Peers:
 - Can **create transactions** that spend some of their bitcoins
 - Do **verify** the correctness of received **transactions** and **blocks** (from other peers)

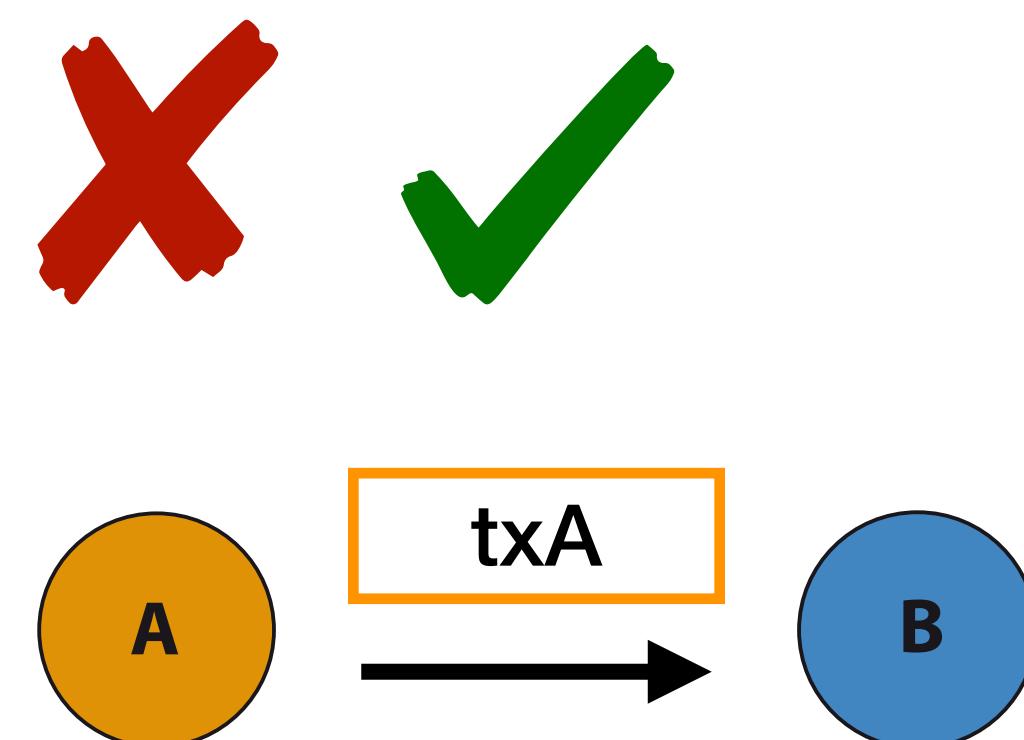
From: Alice	To: Bob	5
-------------	---------	---



The actors (who?) (1/2)

- There are two main roles followed by nodes:
peers and **miners**
- (Normal) Peers:
 - Can **create transactions** that spend some of their bitcoins
 - Do **verify** the correctness of received **transactions** and **blocks** (from other peers)
 - Do **relay** valid **transactions** and **blocks** (created by them or obtained from other peers)

From: Alice	To: Bob	5
-------------	---------	---





The actors (who?) (2/2)



The actors (who?) (2/2)

- Miners:



The actors (who?) (2/2)

- **Miners:**
 - Can do all what a **peer** could do*

The actors (who?) (2/2)

- Miners:
 - Can do all what a **peer** could do*
 - Can **generate blocks** through a process known as mining



The actors (who?) (2/2)

- Miners:
 - Can do all what a **peer** could do*
 - Can **generate blocks** through a process known as mining



* There are specific purpose miners (ASICs) that only perform mining

The purpose (why?)

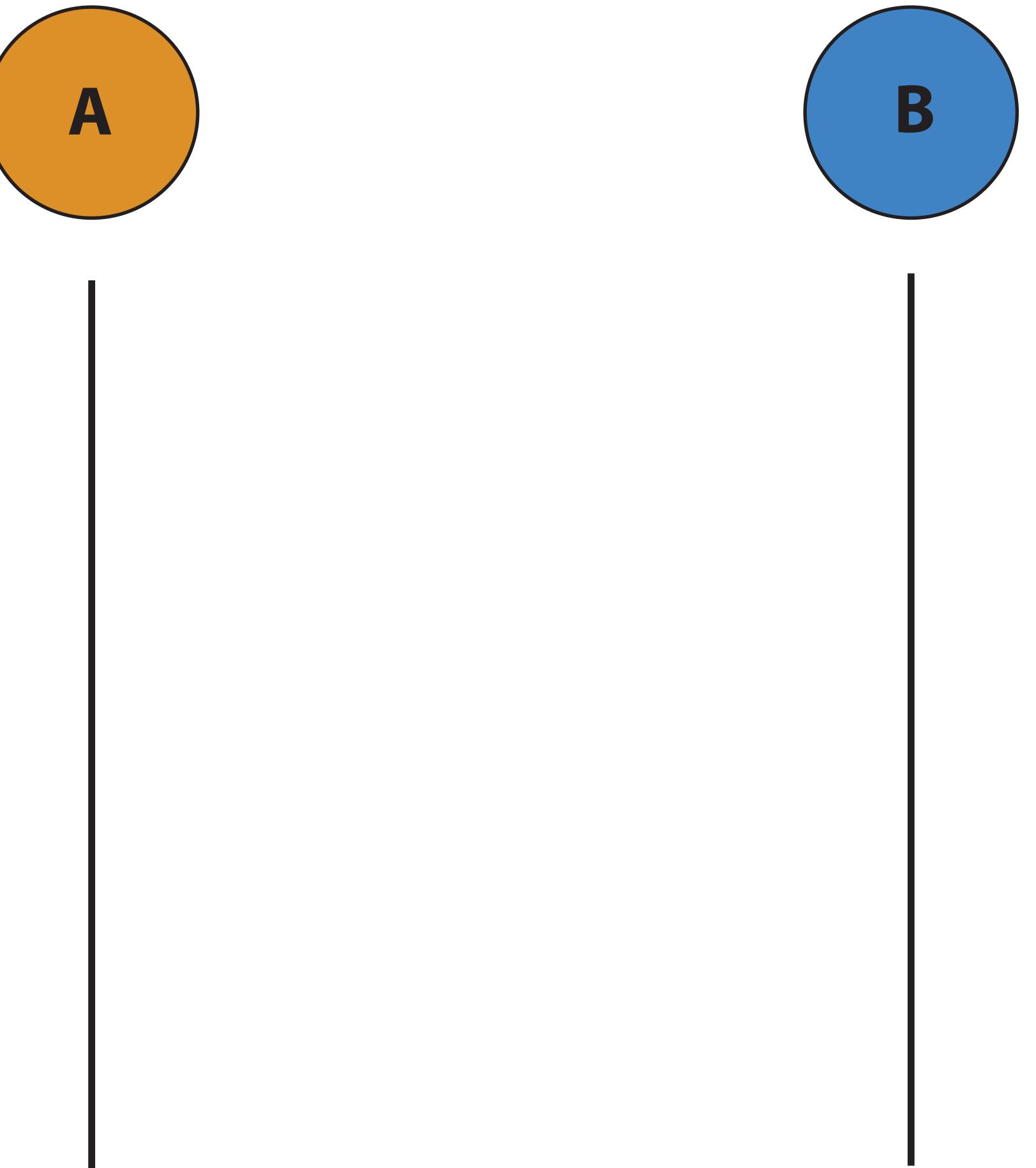
- Peers relay transactions in order to reach miners, which will include such transactions in future blocks
- Miners generate blocks to obtain its reward (and also the transactions fees)
- Blocks are relayed to ultimately achieve a consistent view of the blockchain
- Peers validate transactions and blocks (and relay only the valid ones) in order to avoid cheating (e.g: double-spending, coin forgery, etc)



The gossip protocol (how?)

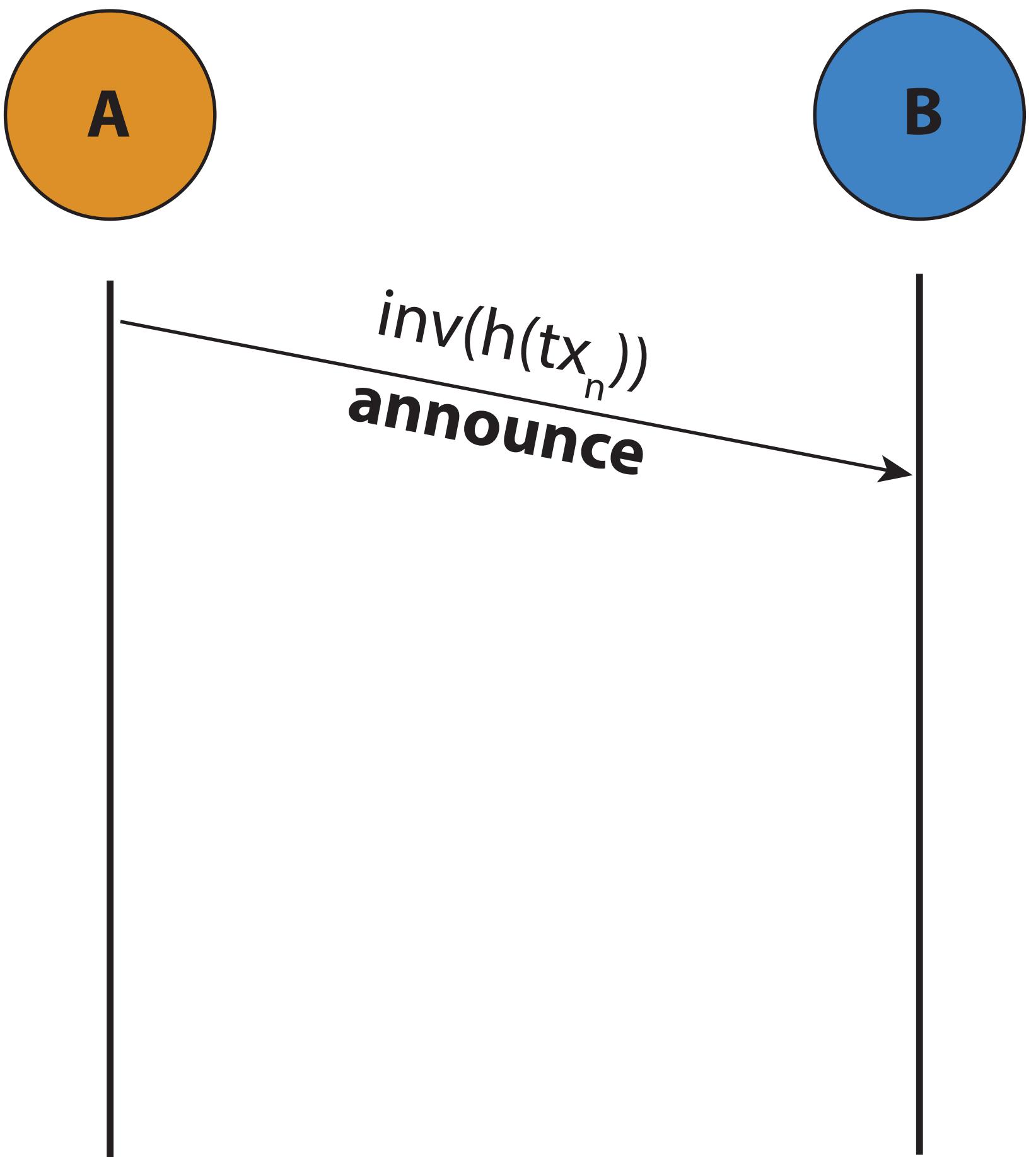
The gossip protocol (how?)

- Items (transactions and blocks) are shared between peers in a push manner



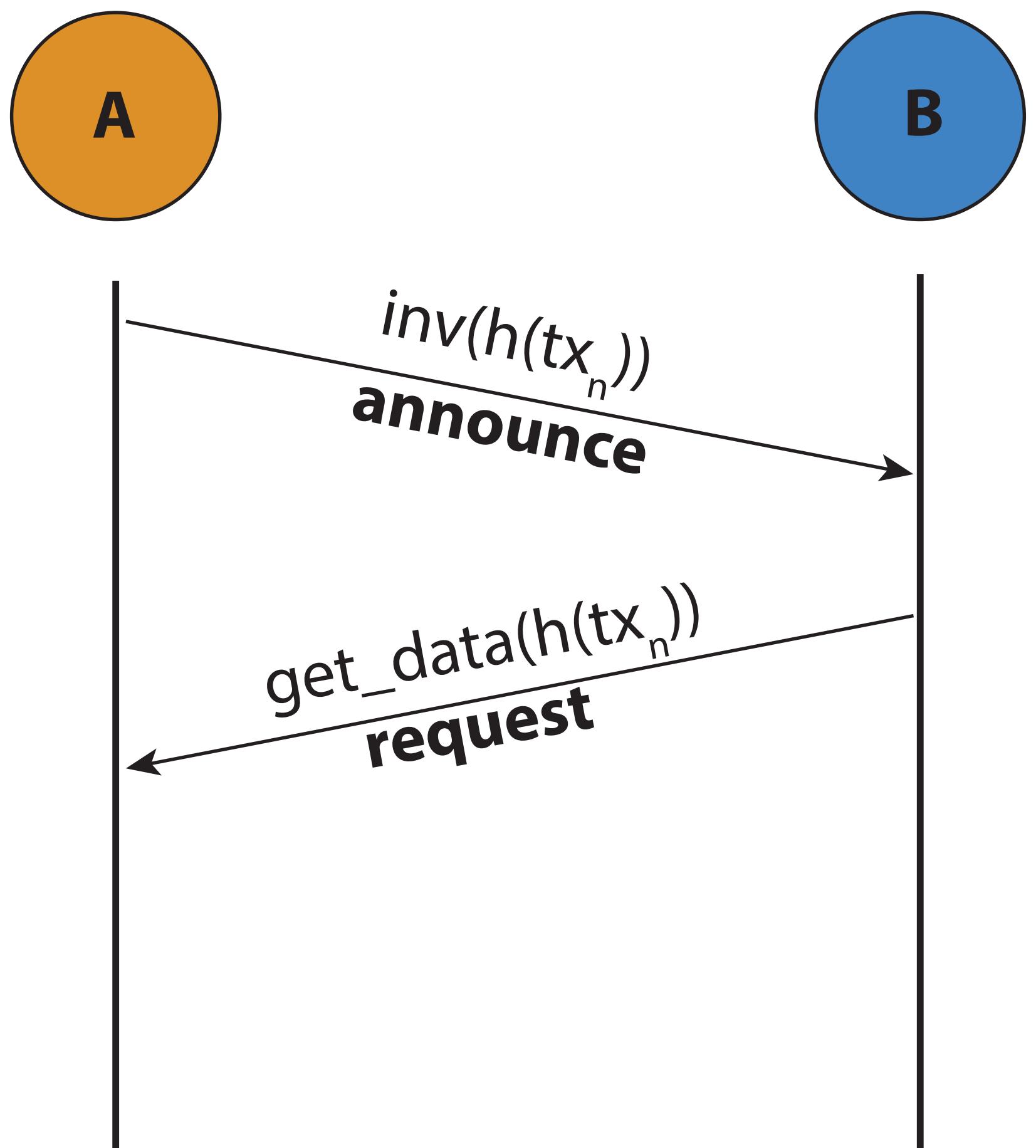
The gossip protocol (how?)

- Items (transactions and blocks) are shared between peers in a push manner
- When a peer receives / generates a new item he announce it to his neighbors (**announce**)



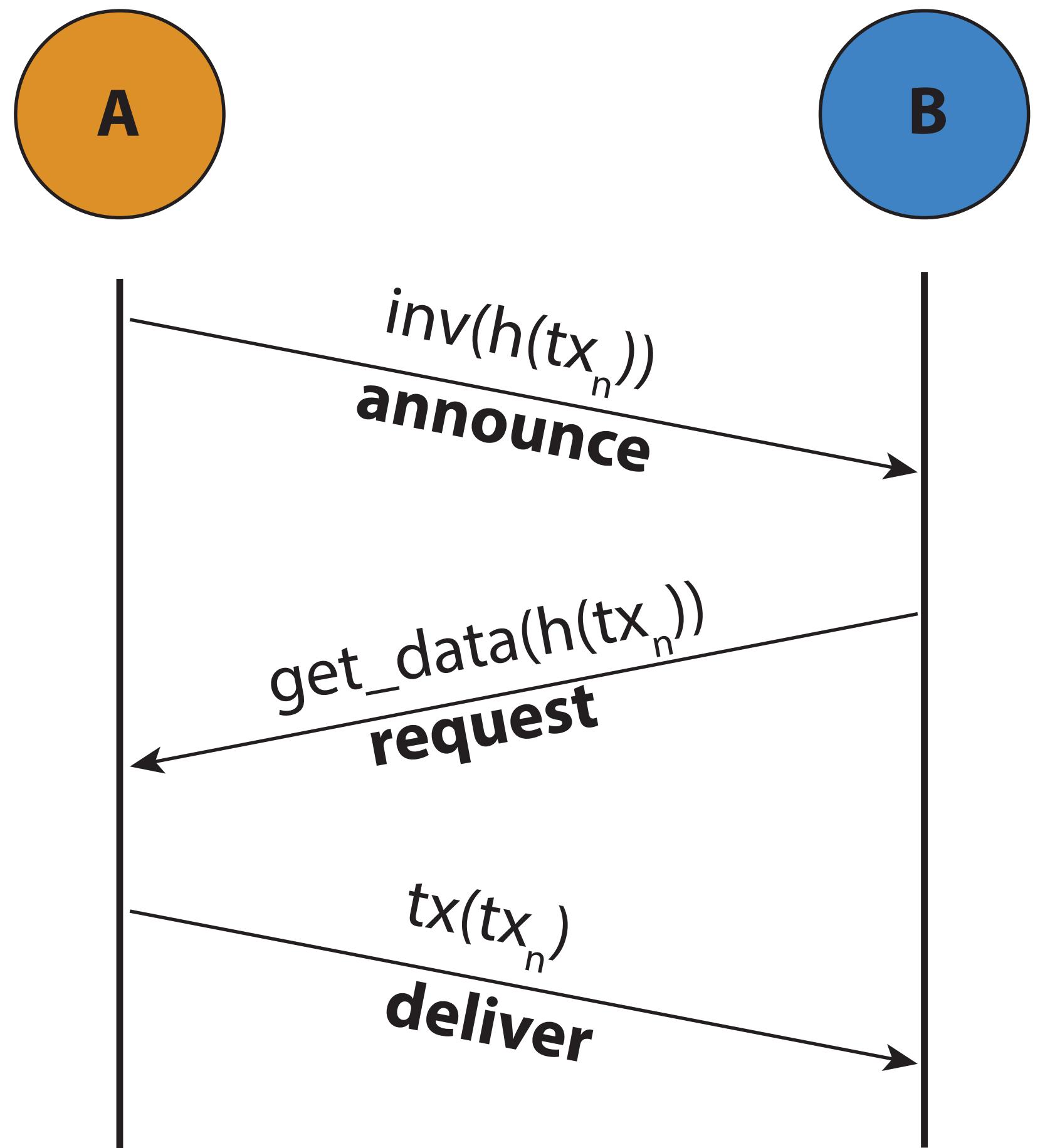
The gossip protocol (how?)

- Items (transactions and blocks) are shared between peers in a push manner
- When a peer receives / generates a new item he announce it to his neighbors (**announce**)
- Upon receiving an announce of an item, a node that does not know about it will request such item back to the announcer (**request**)



The gossip protocol (how?)

- Items (transactions and blocks) are shared between peers in a push manner
- When a peer receives / generates a new item he announce it to his neighbors (**announce**)
- Upon receiving an announce of an item, a node that does not know about it will request such item back to the announcer (**request**)
- Upon receiving a request of a known item, a node will reply back with it (**deliver**)



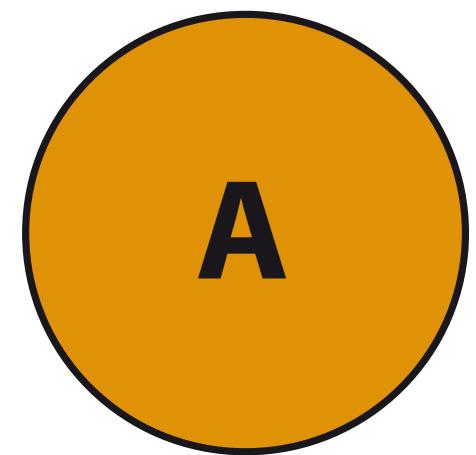
Information propagation



Information propagation (1/2)

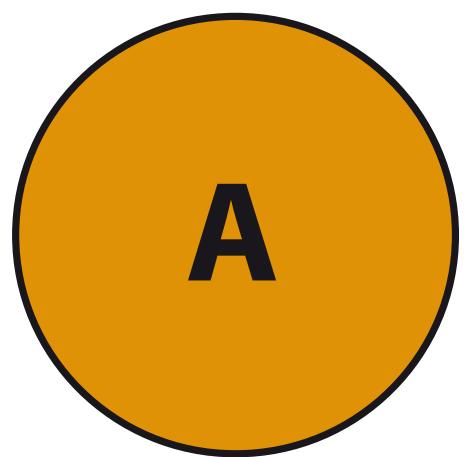
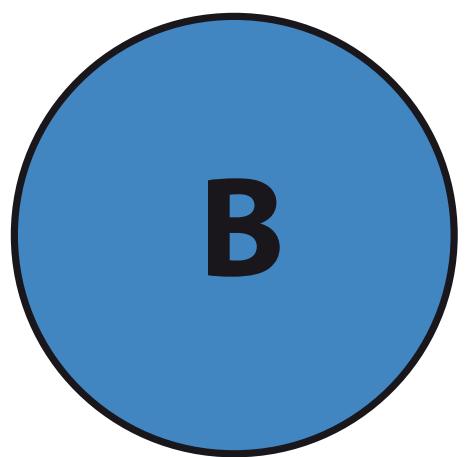


Information propagation (1/2)



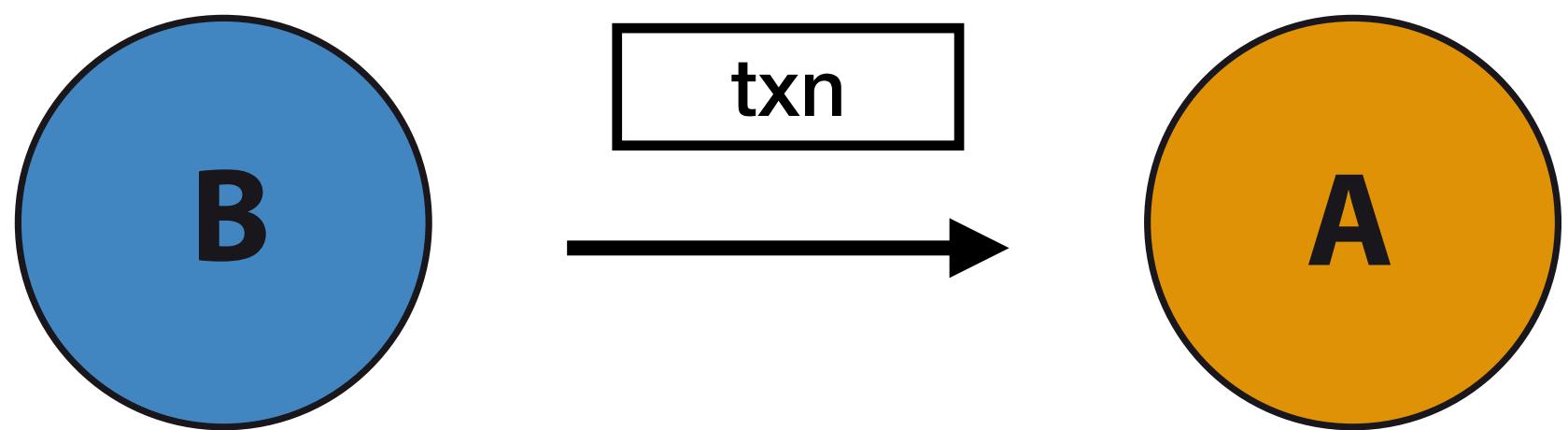


Information propagation (1/2)

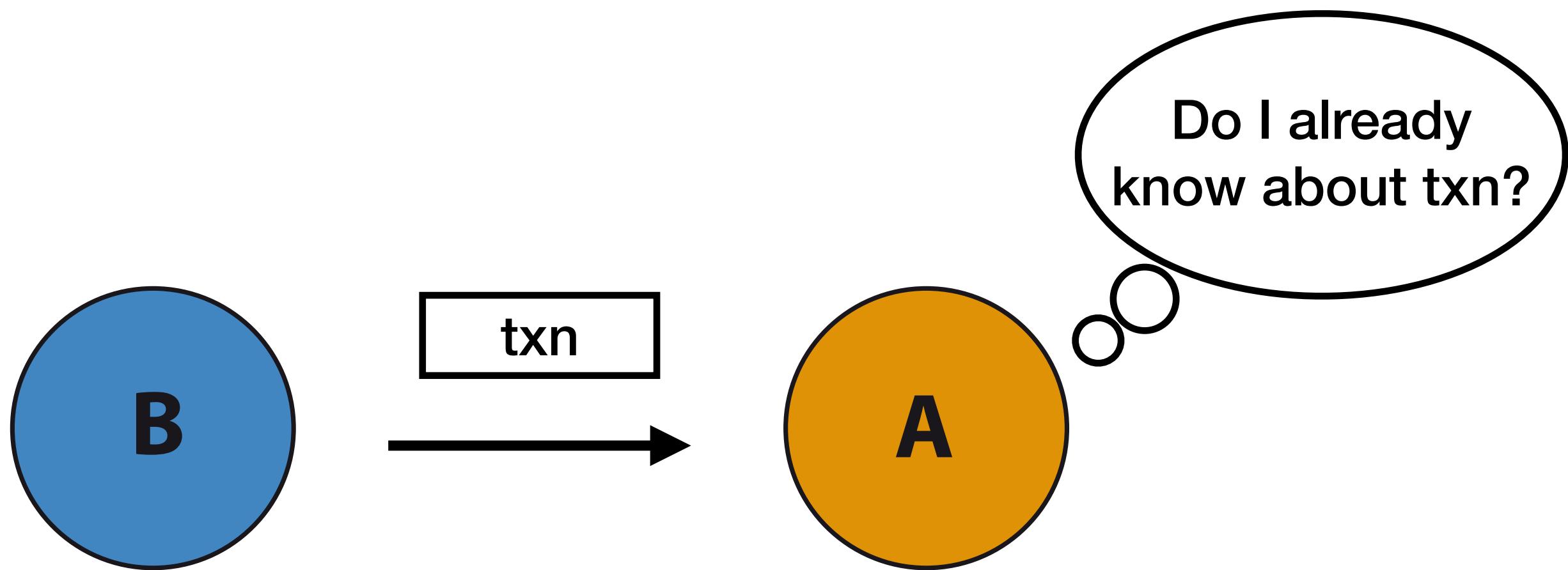




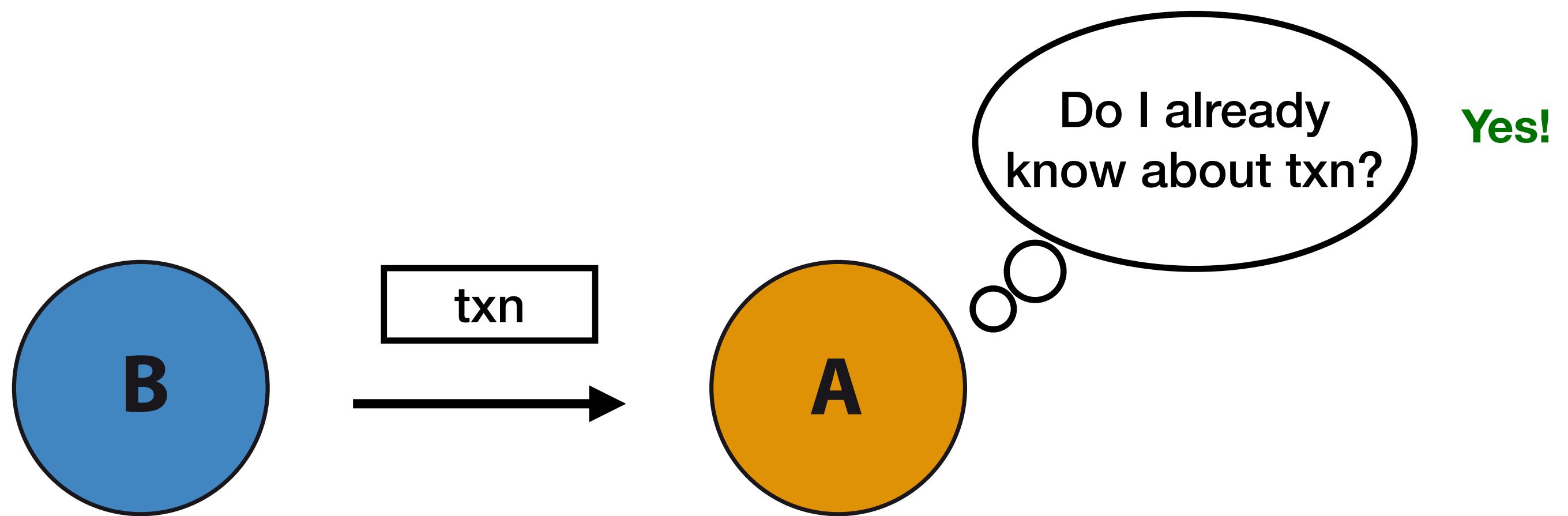
Information propagation (1/2)



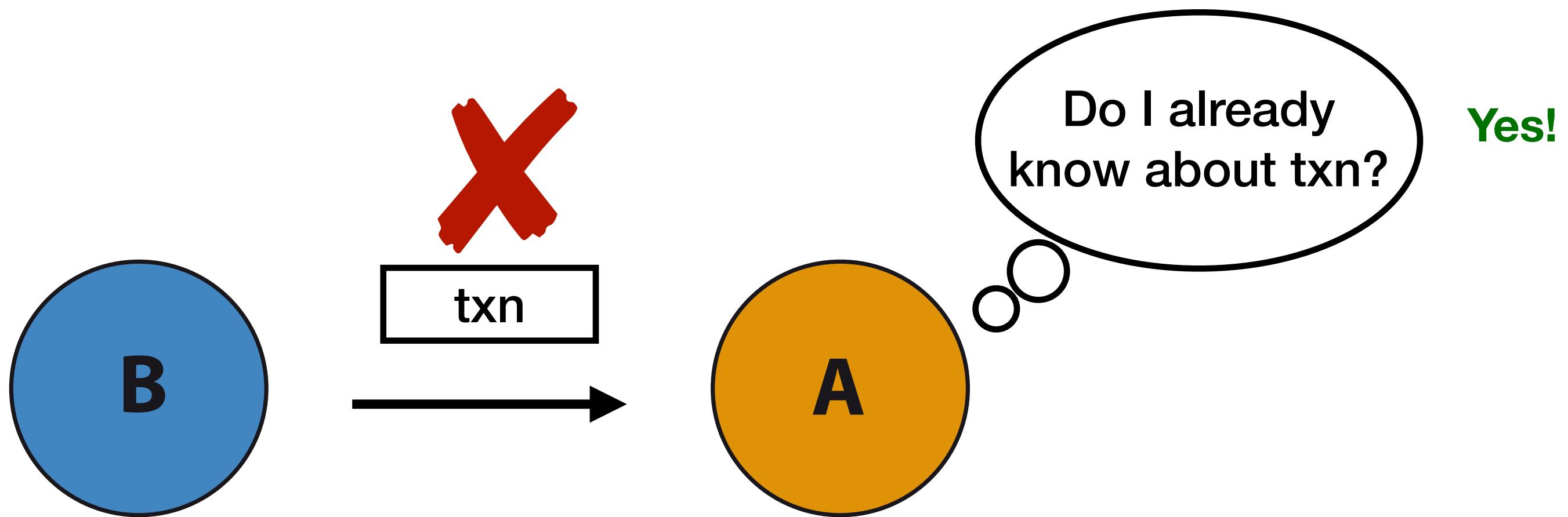
Information propagation (1/2)



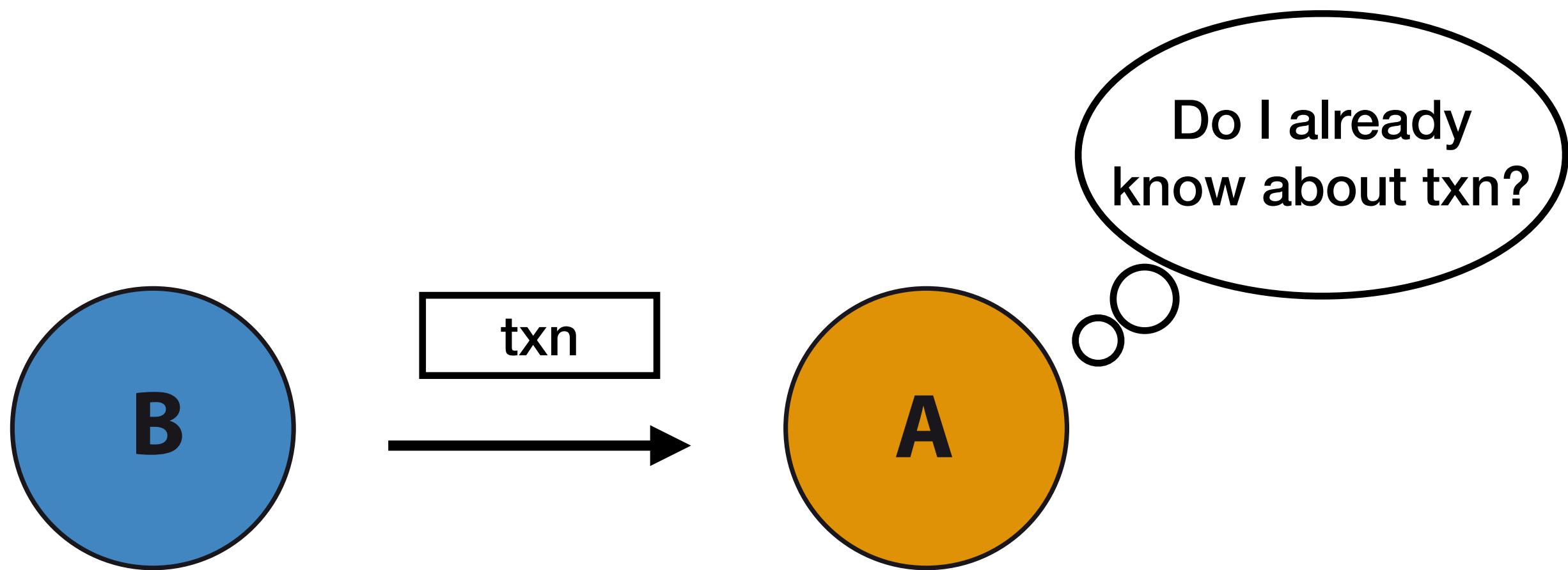
Information propagation (1/2)



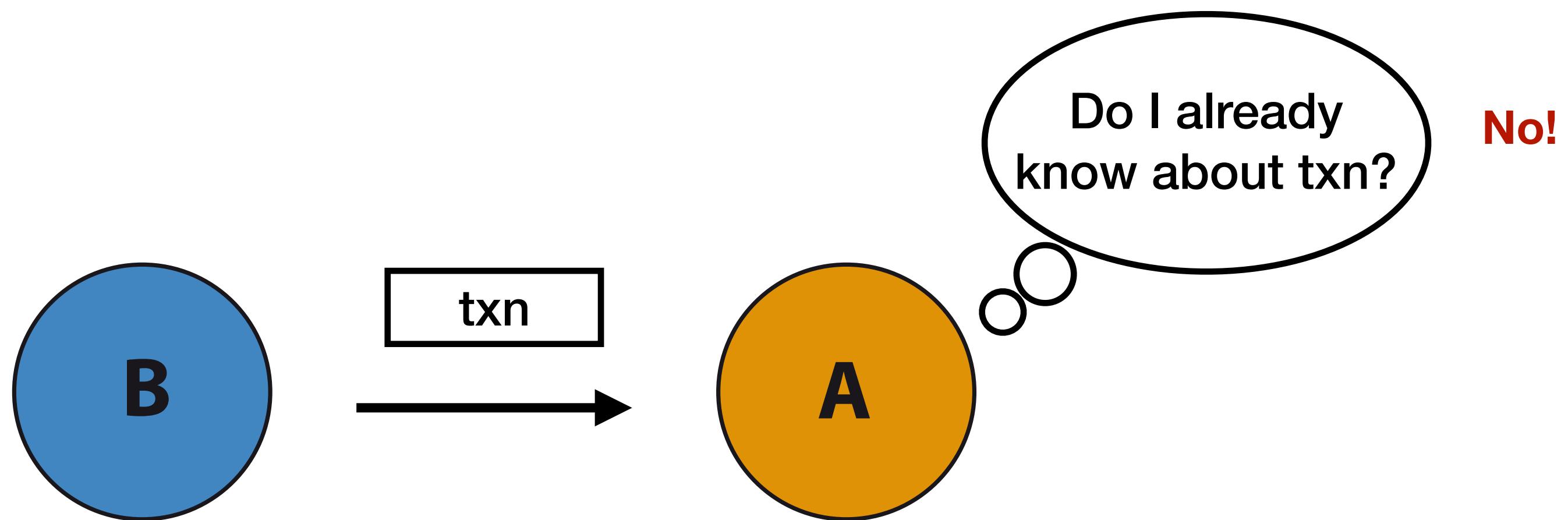
Information propagation (1/2)



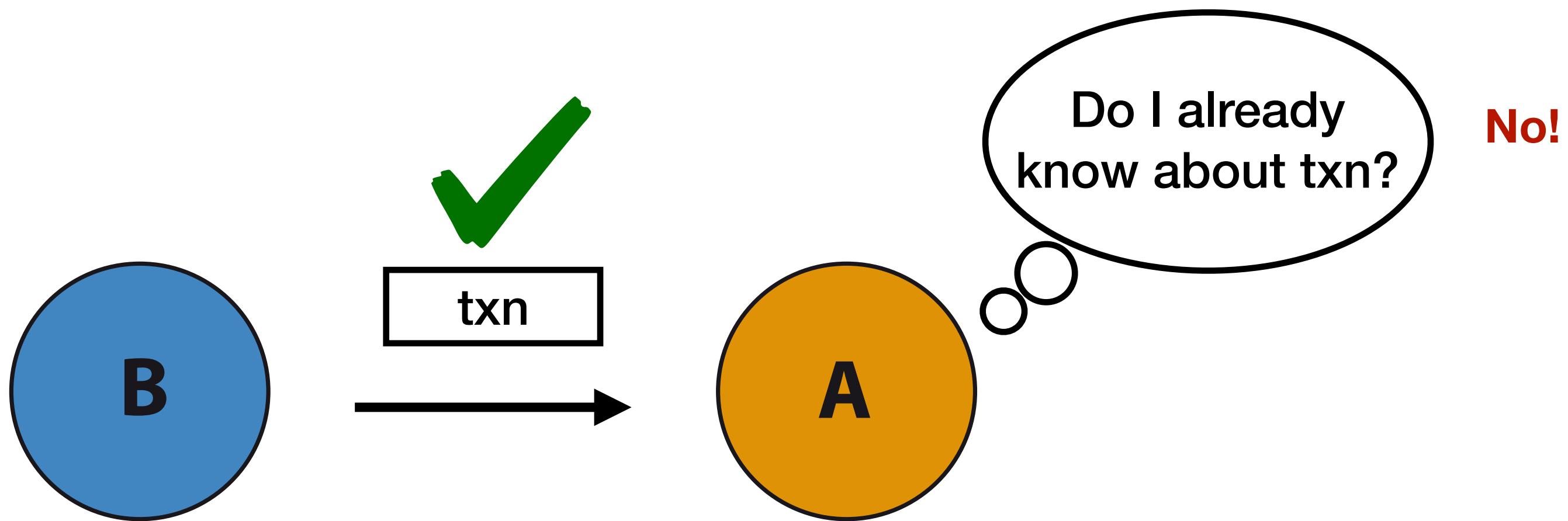
Information propagation (1/2)



Information propagation (1/2)

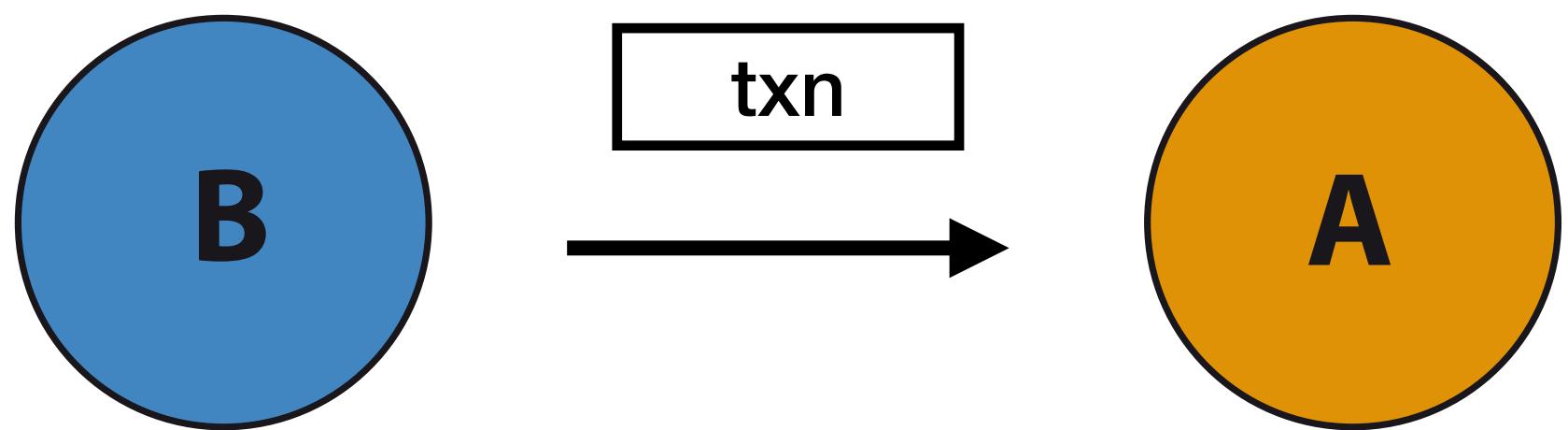


Information propagation (1/2)



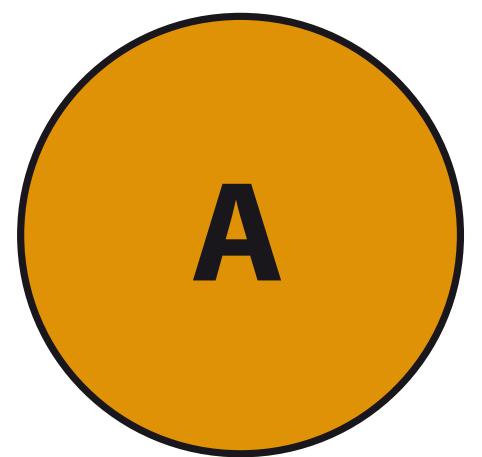
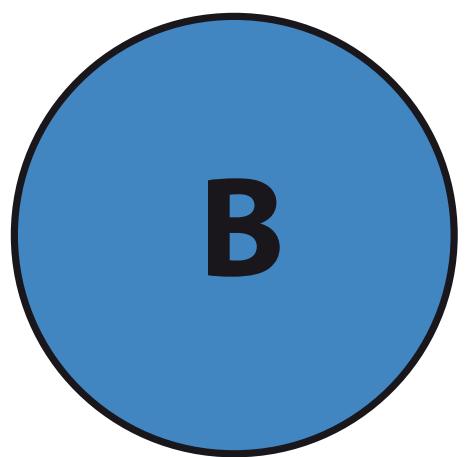


Information propagation (1/2)



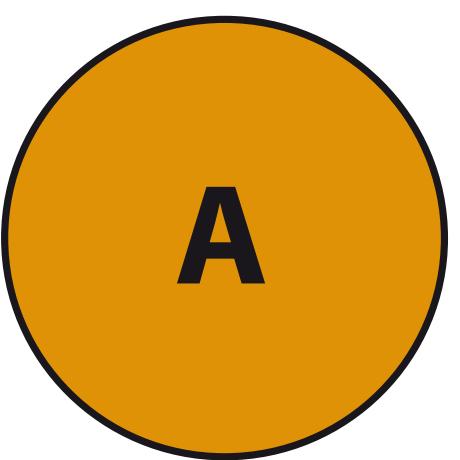
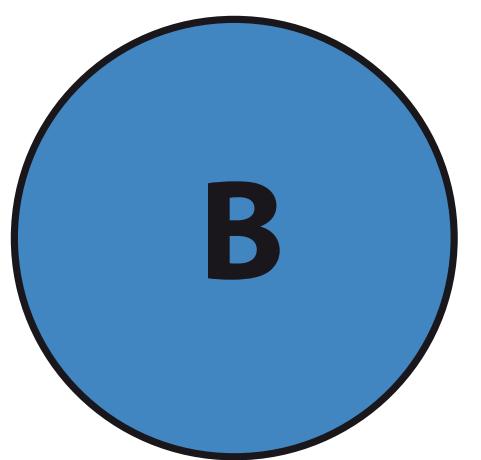


Information propagation (1/2)



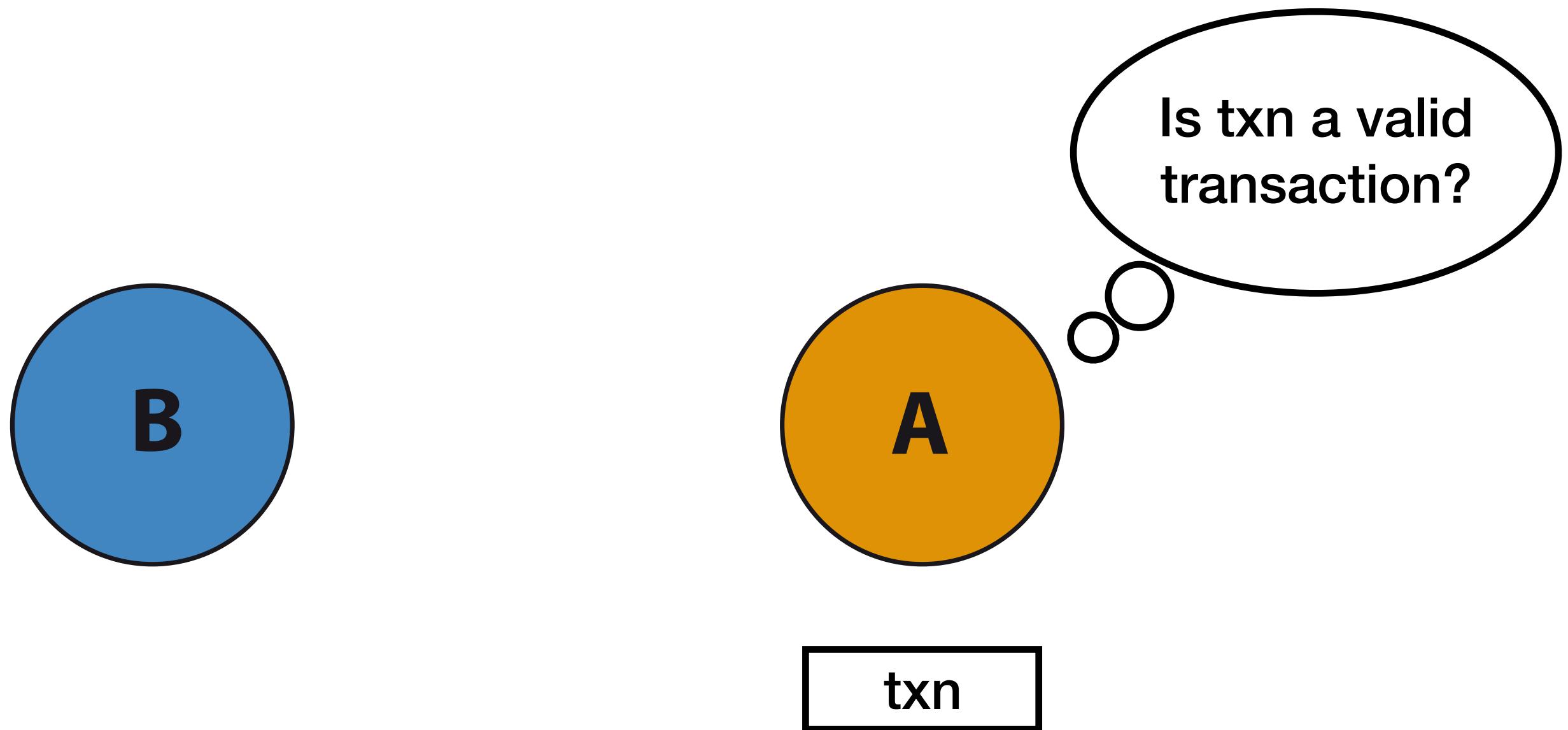


Information propagation (1/2)

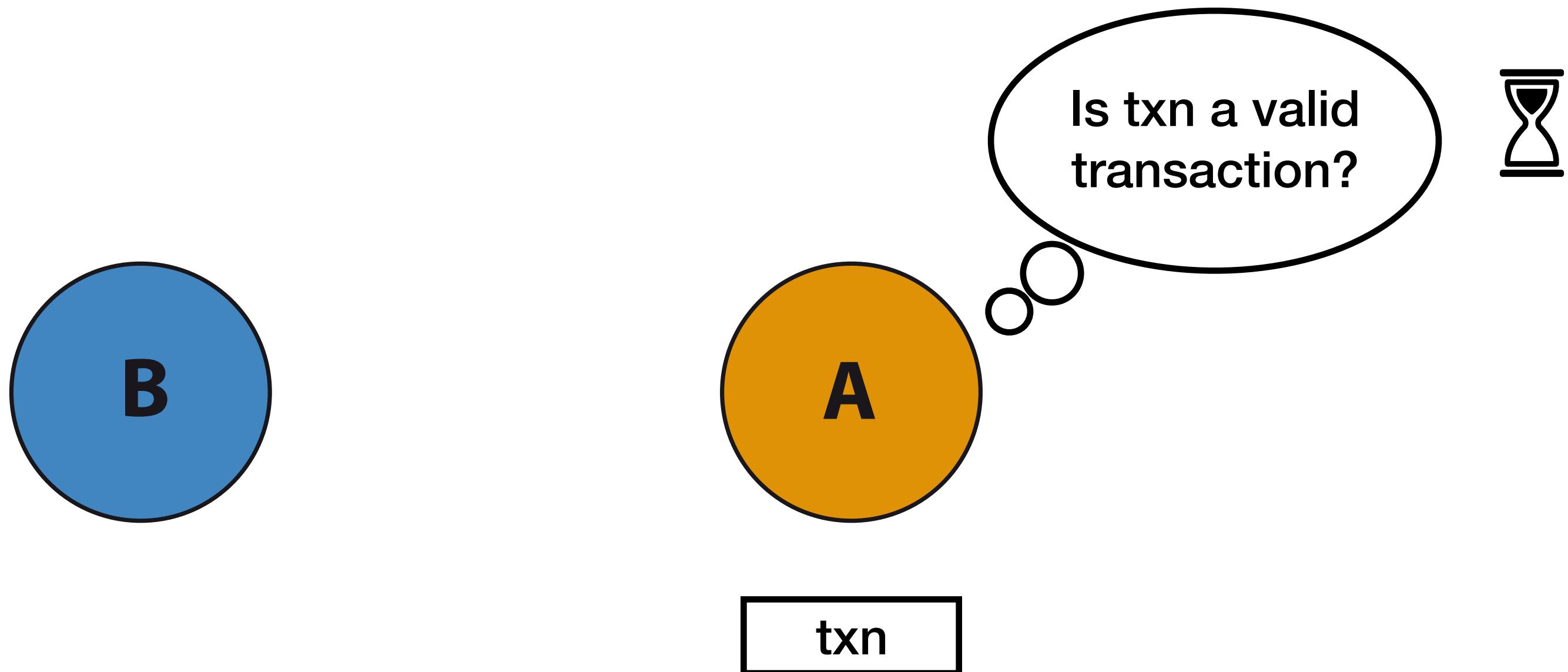


txn

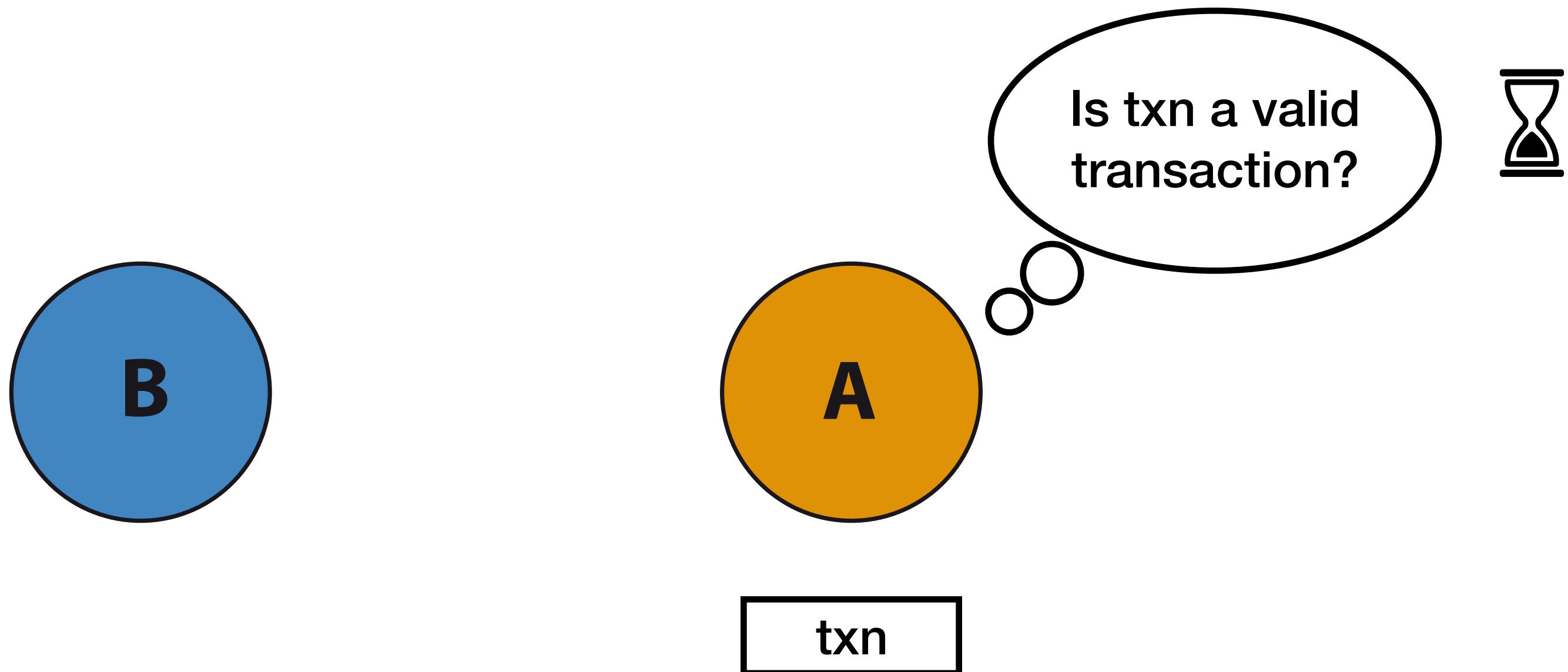
Information propagation (1/2)



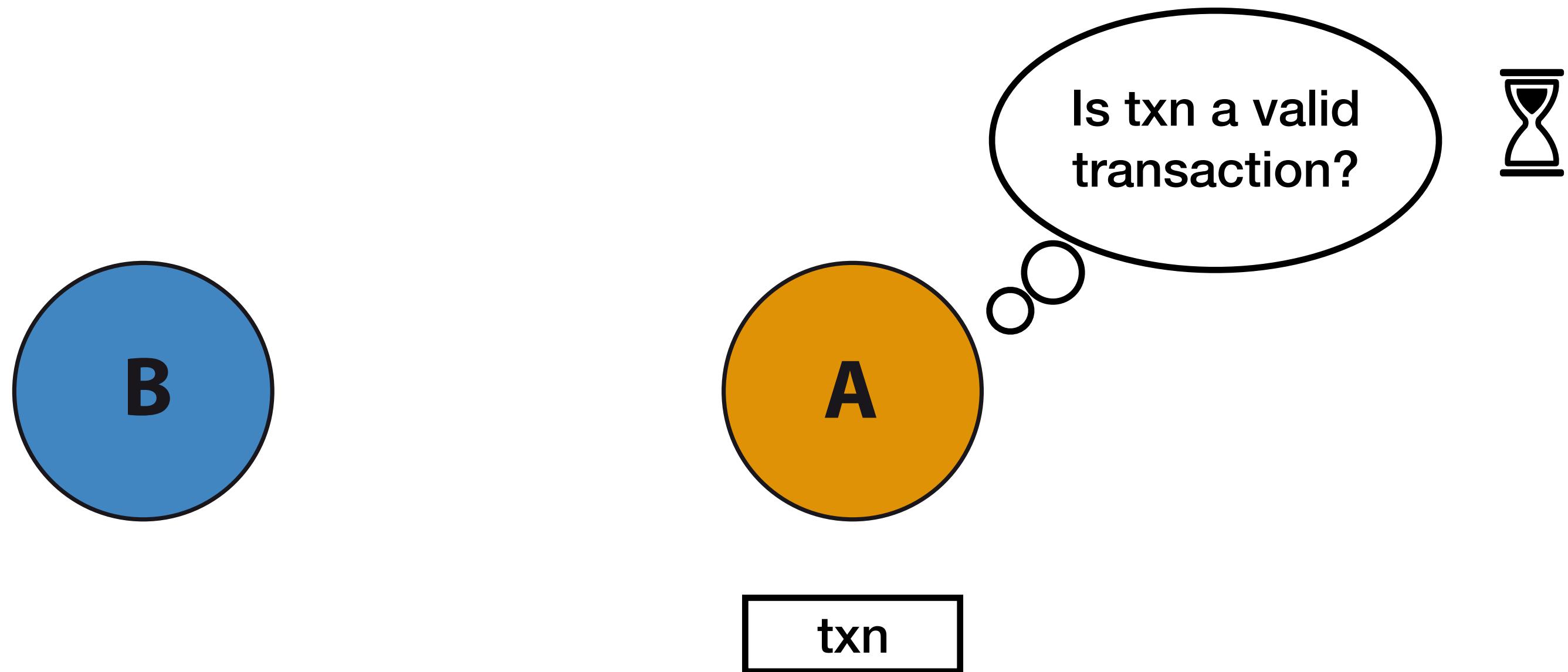
Information propagation (1/2)



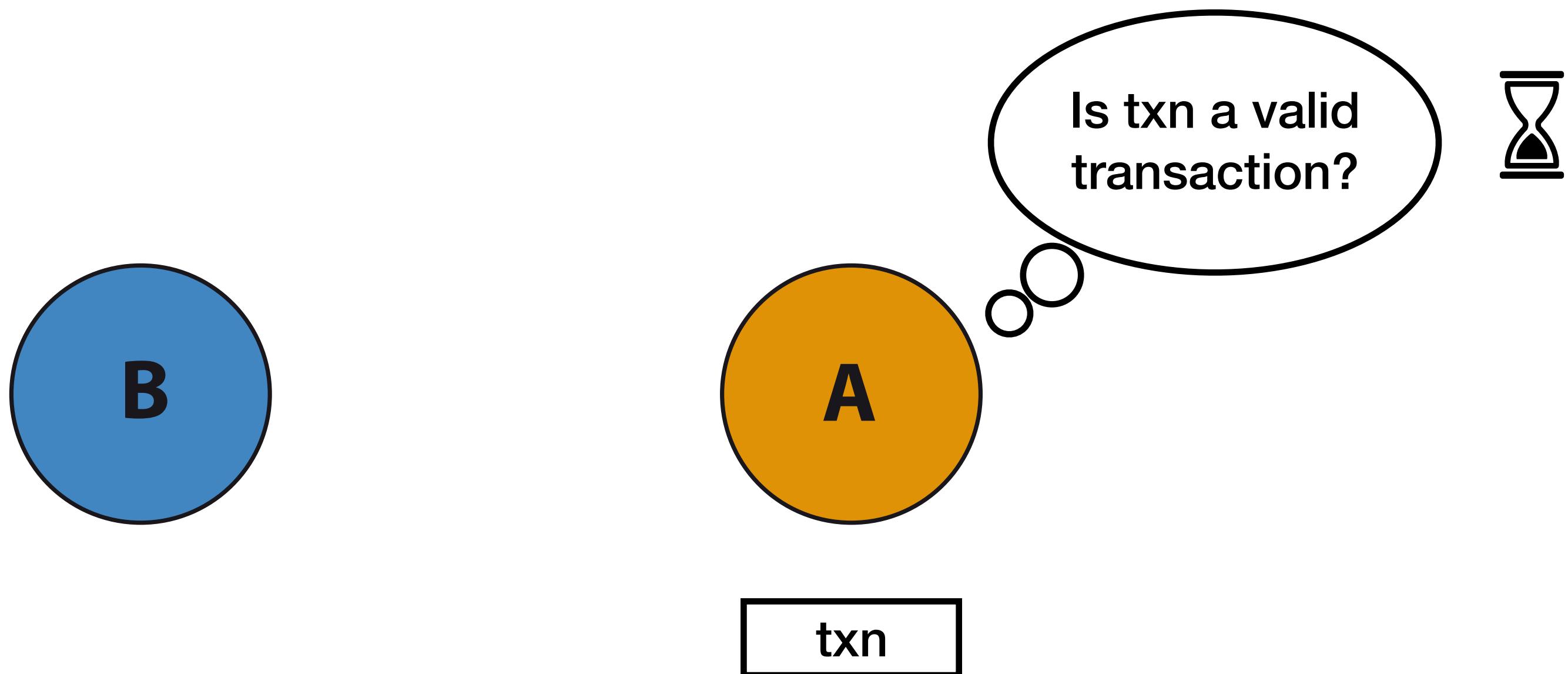
Information propagation (1/2)



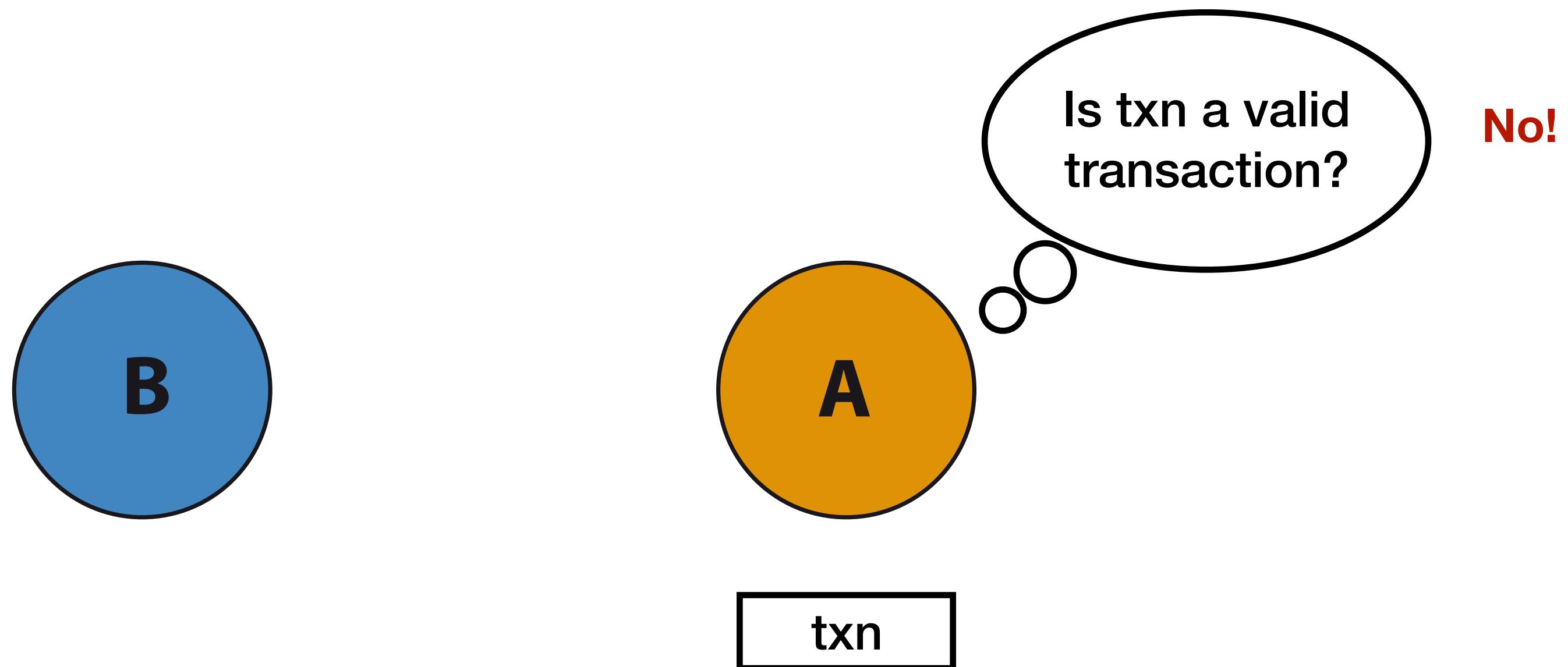
Information propagation (1/2)



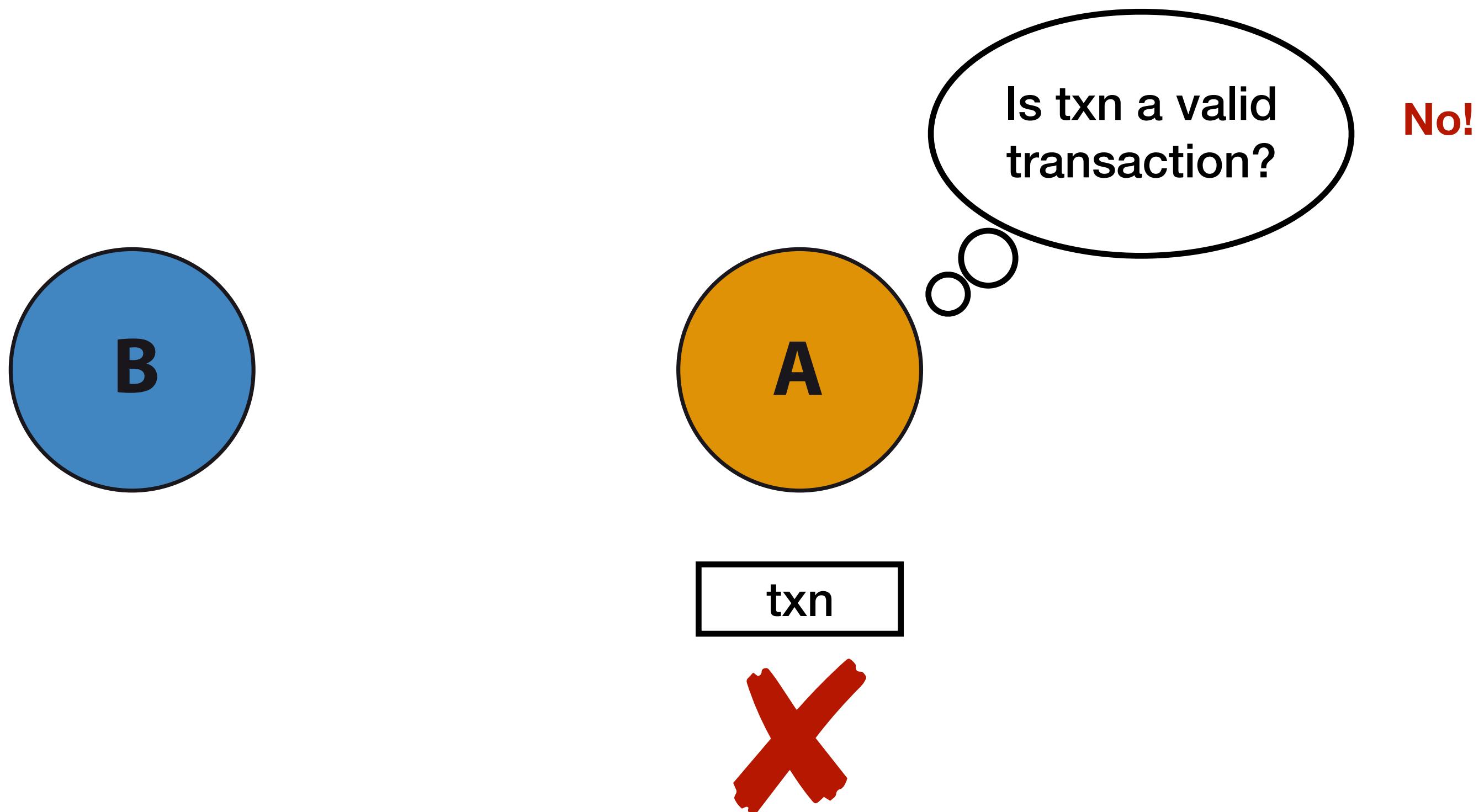
Information propagation (1/2)



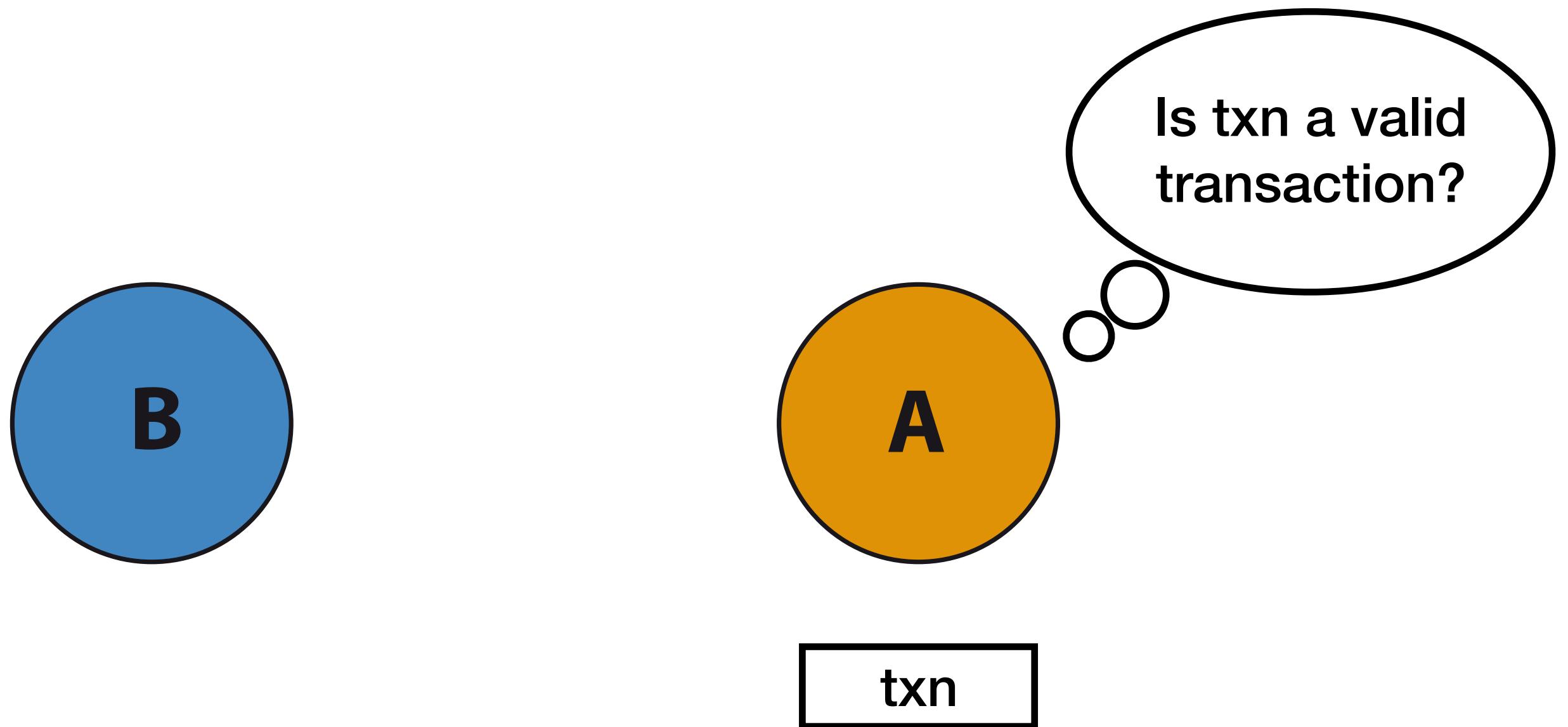
Information propagation (1/2)



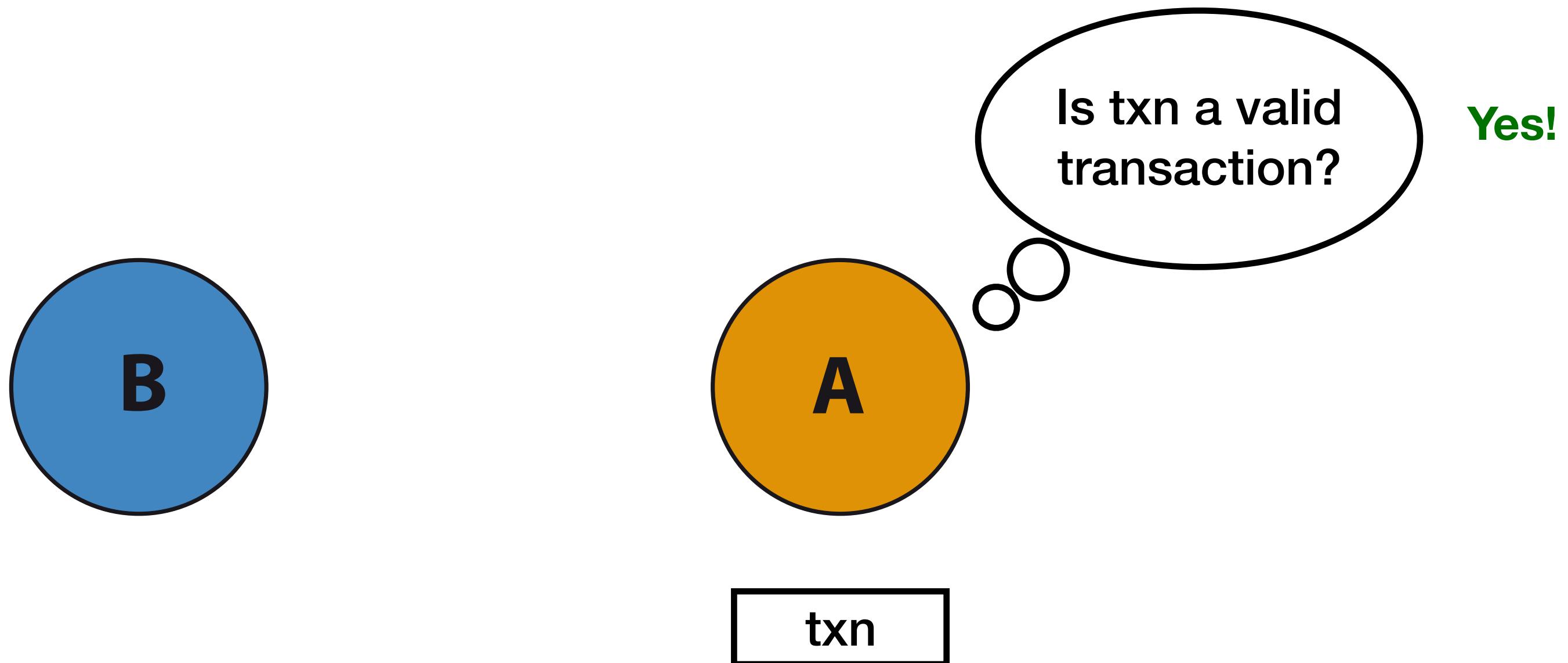
Information propagation (1/2)



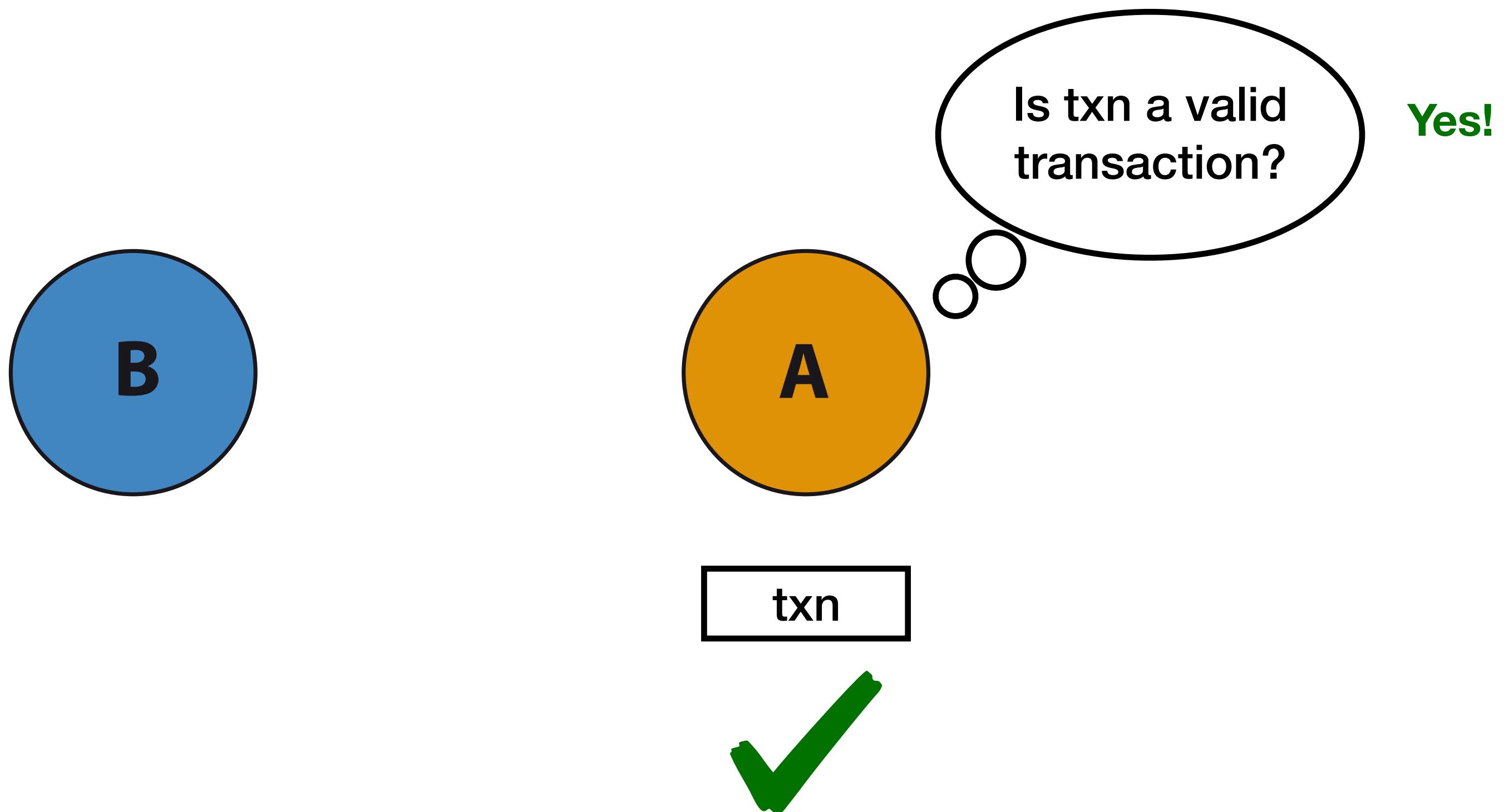
Information propagation (1/2)



Information propagation (1/2)

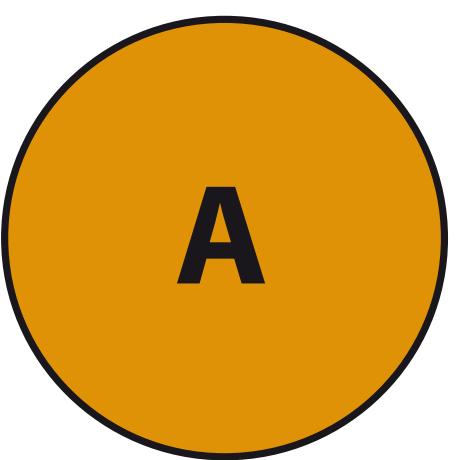
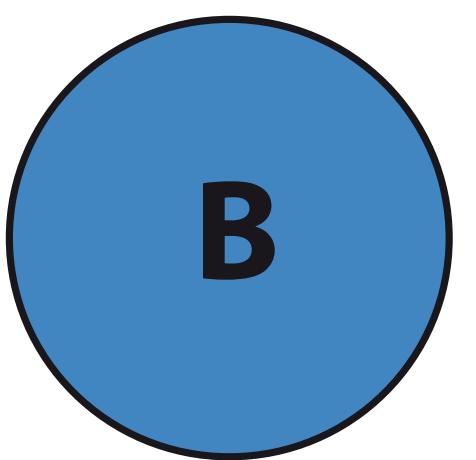


Information propagation (1/2)





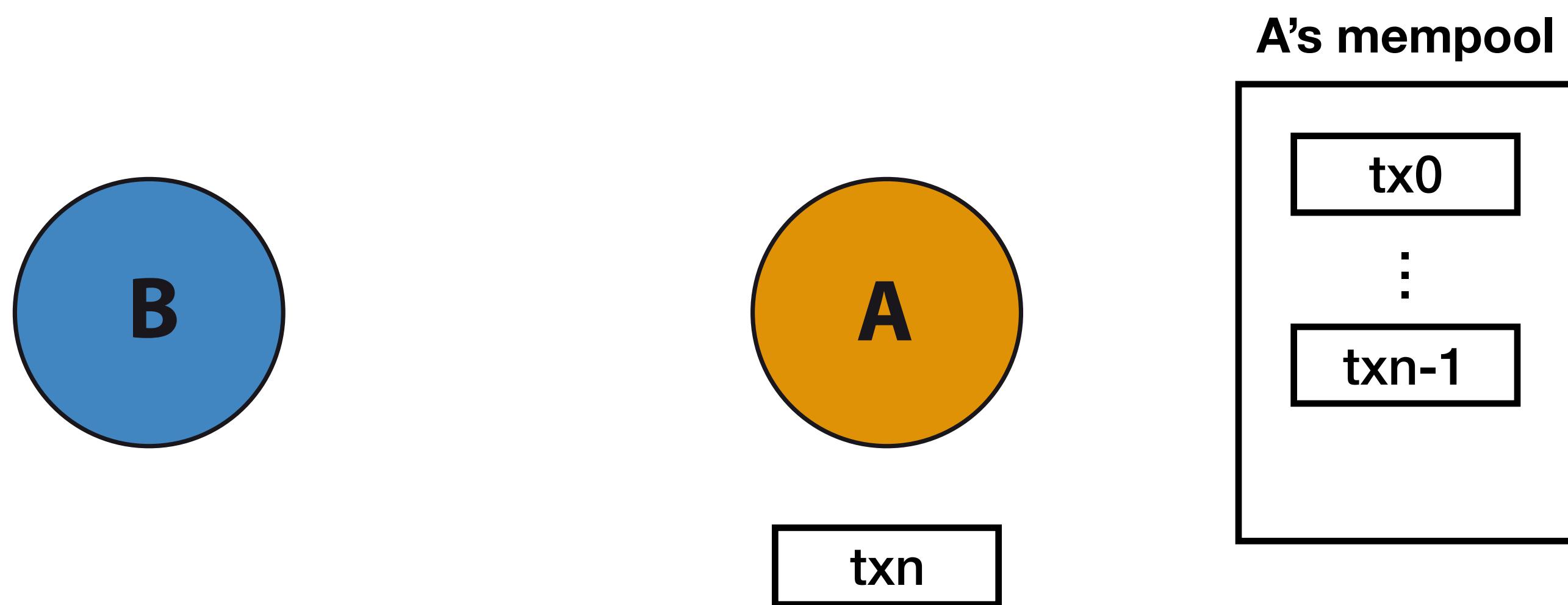
Information propagation (1/2)



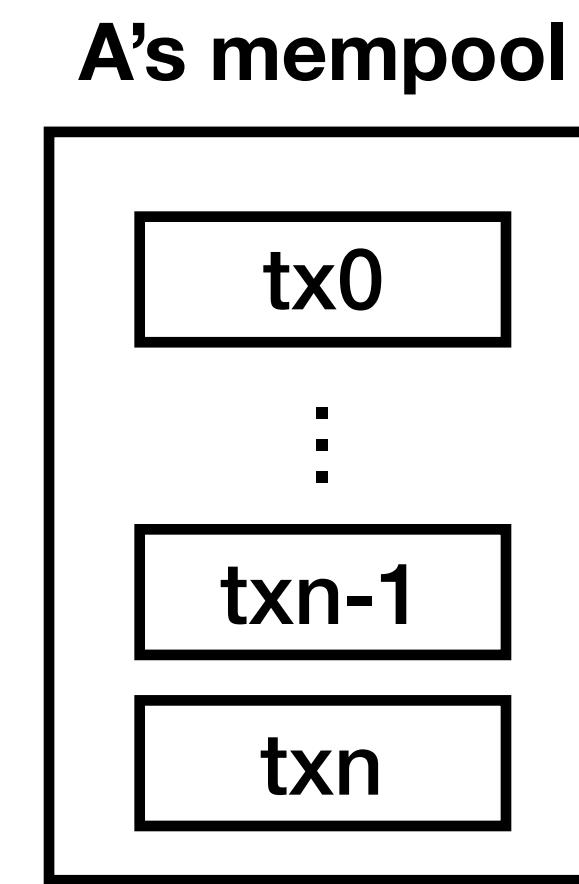
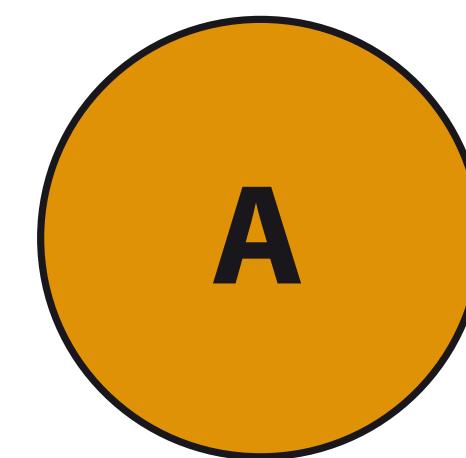
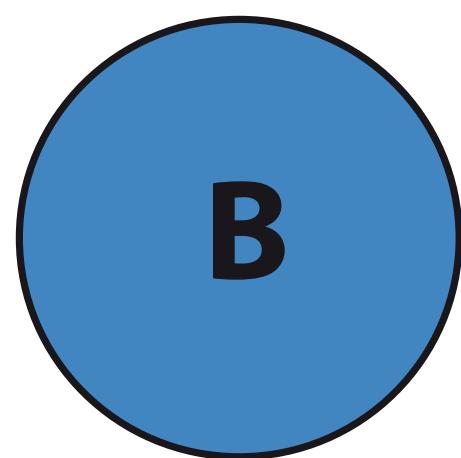
txn



Information propagation (1/2)



Information propagation (1/2)

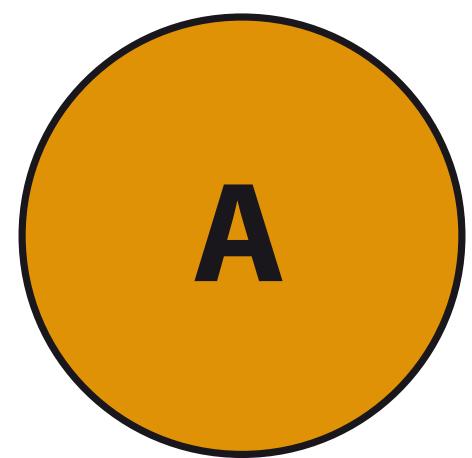




Information propagation (2/2)

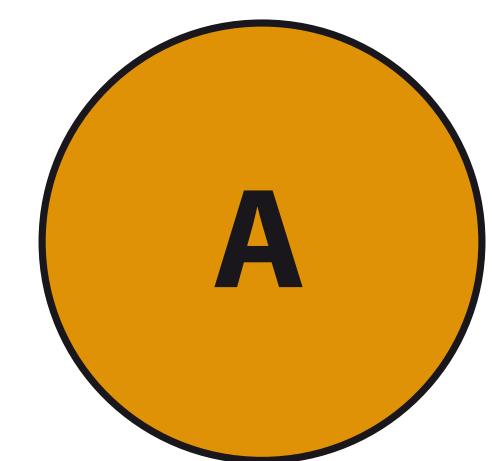
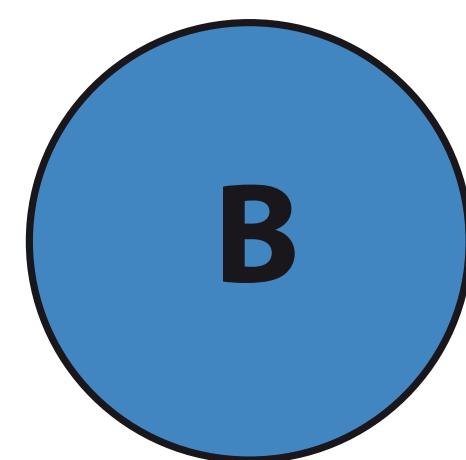


Information propagation (2/2)

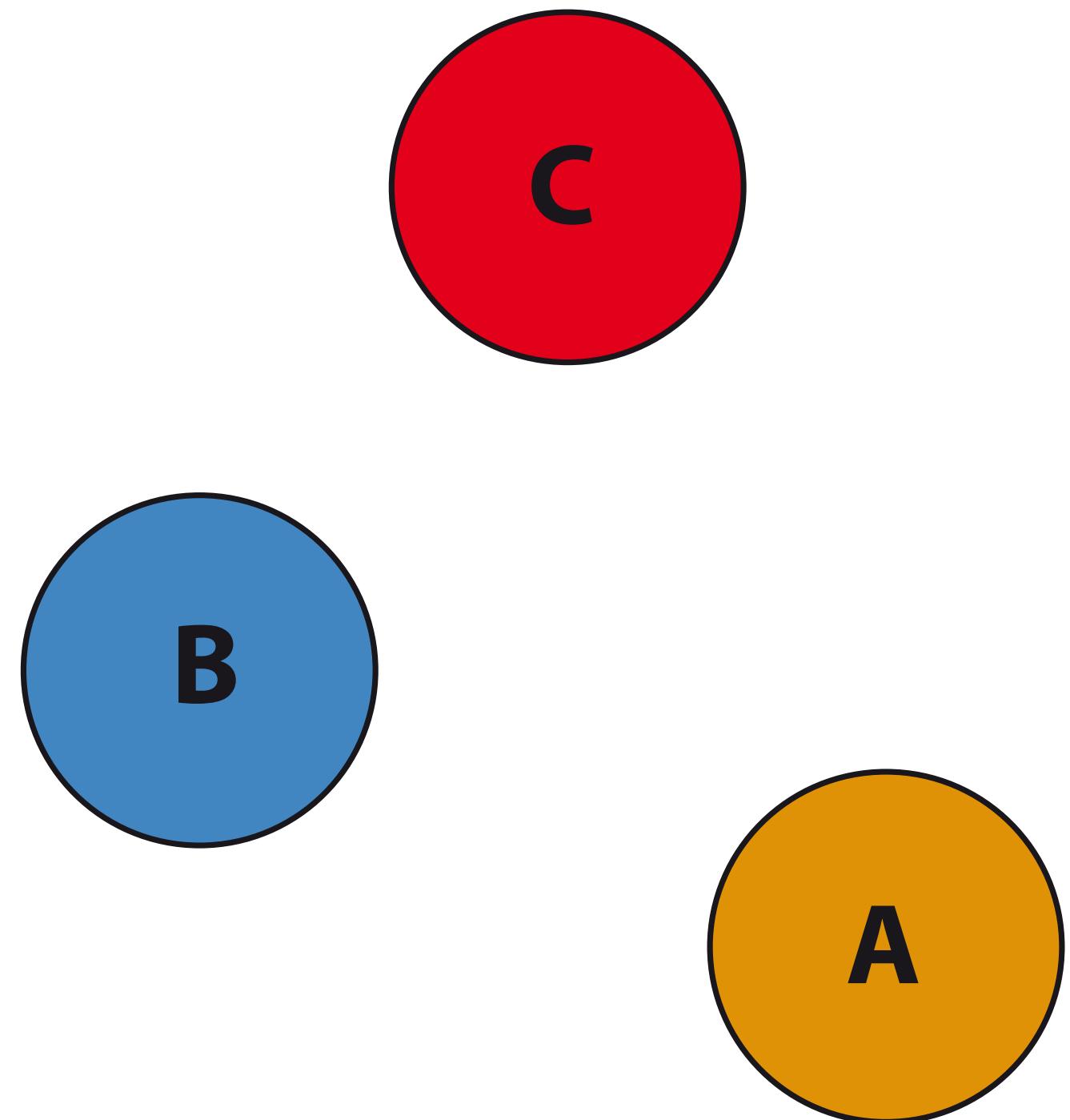




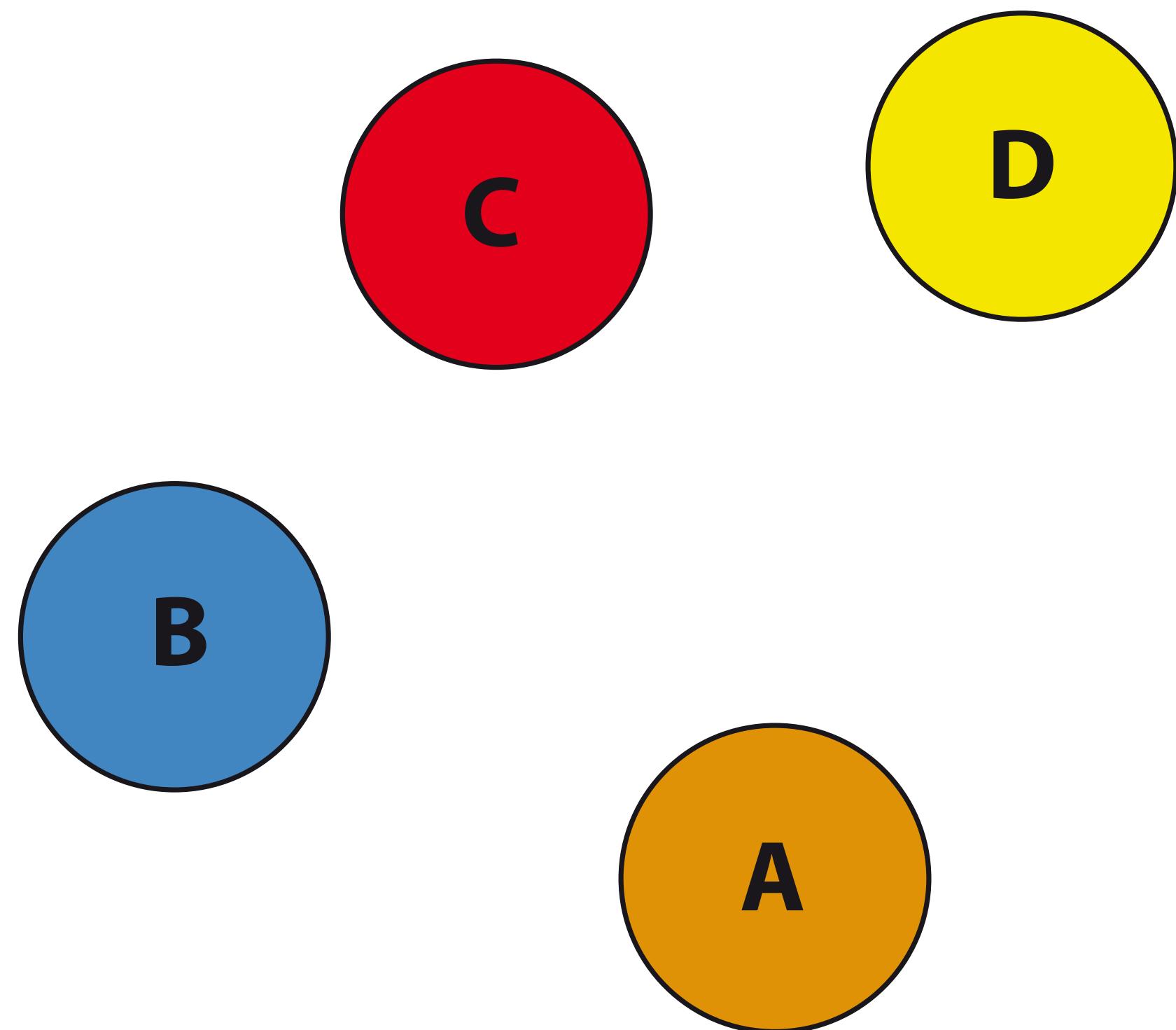
Information propagation (2/2)



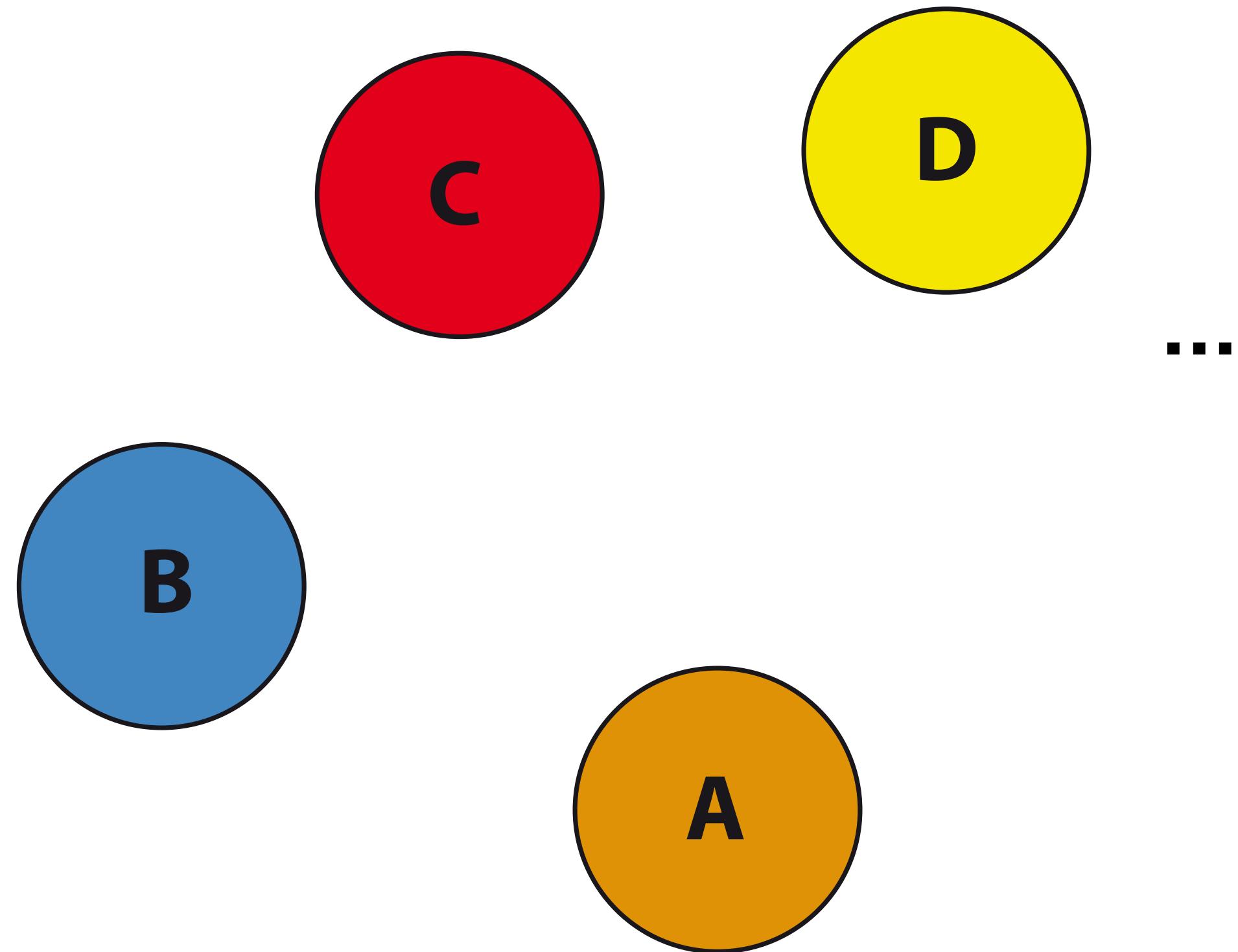
Information propagation (2/2)



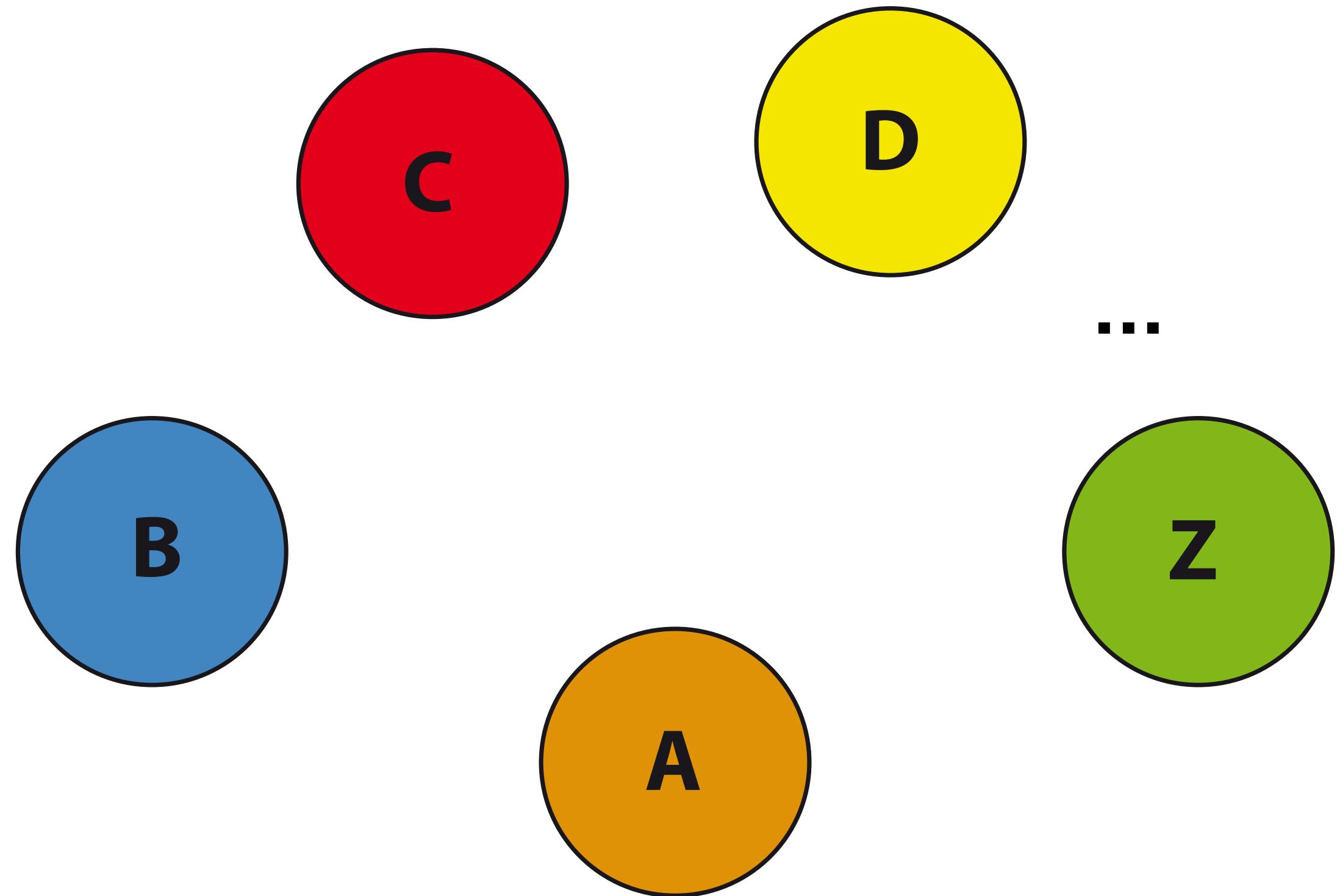
Information propagation (2/2)



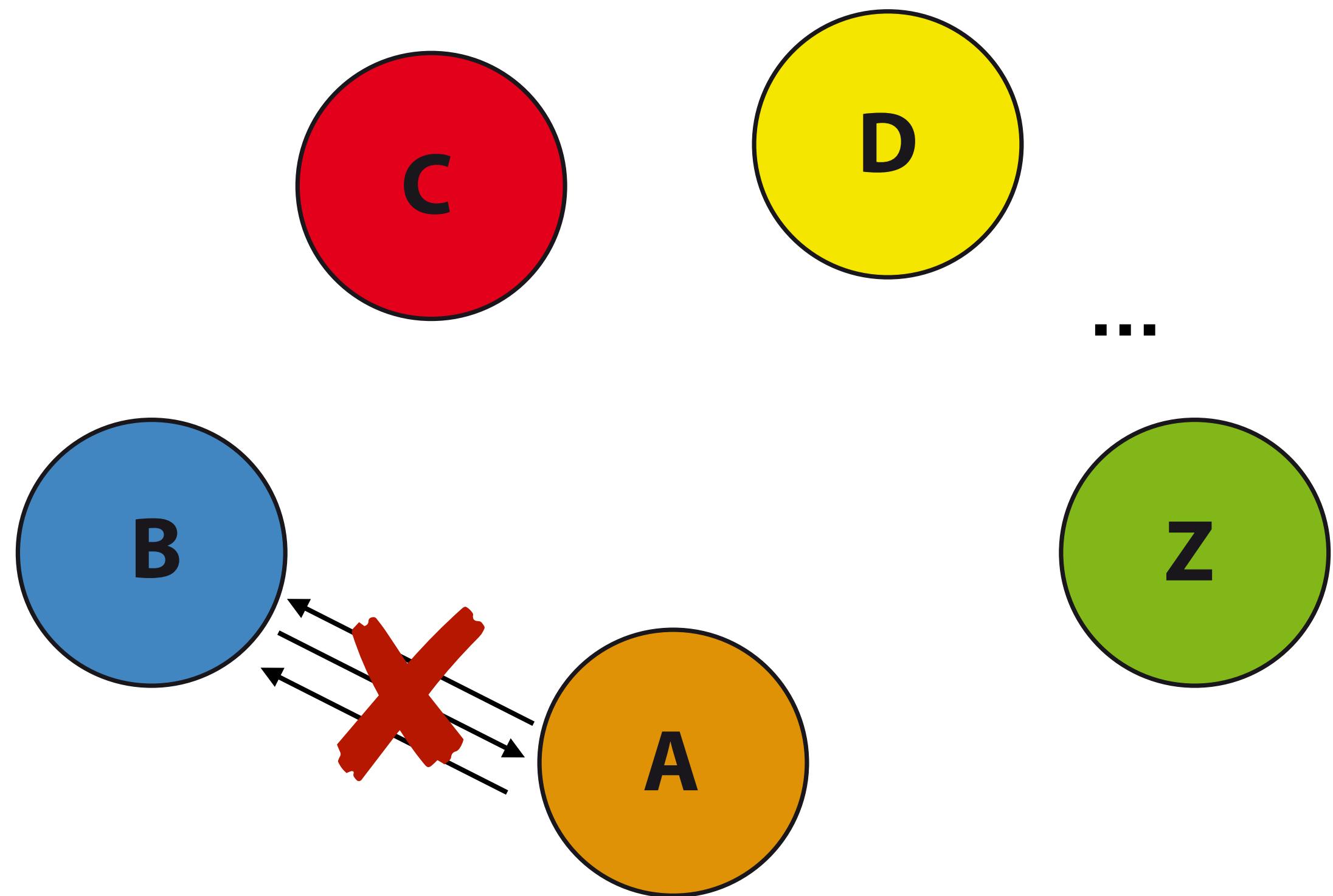
Information propagation (2/2)



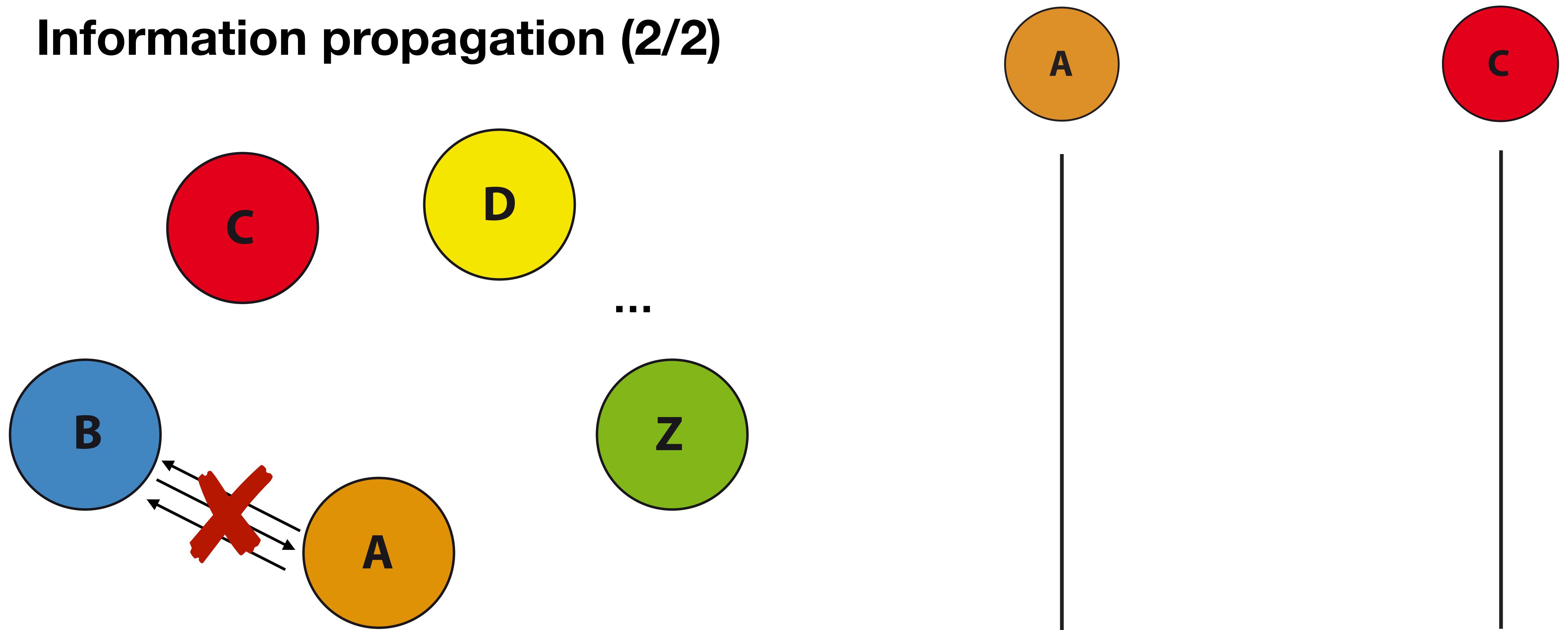
Information propagation (2/2)



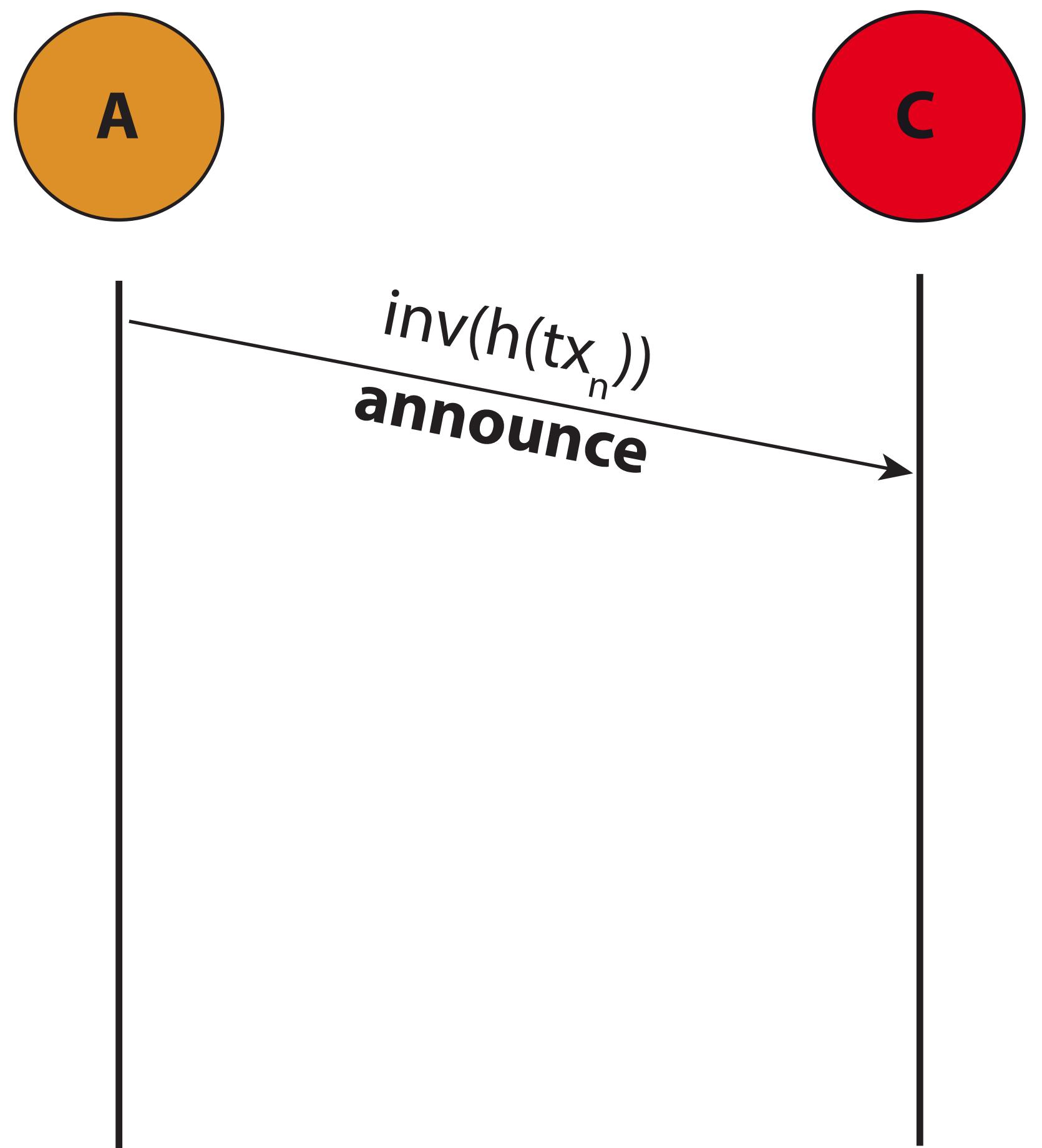
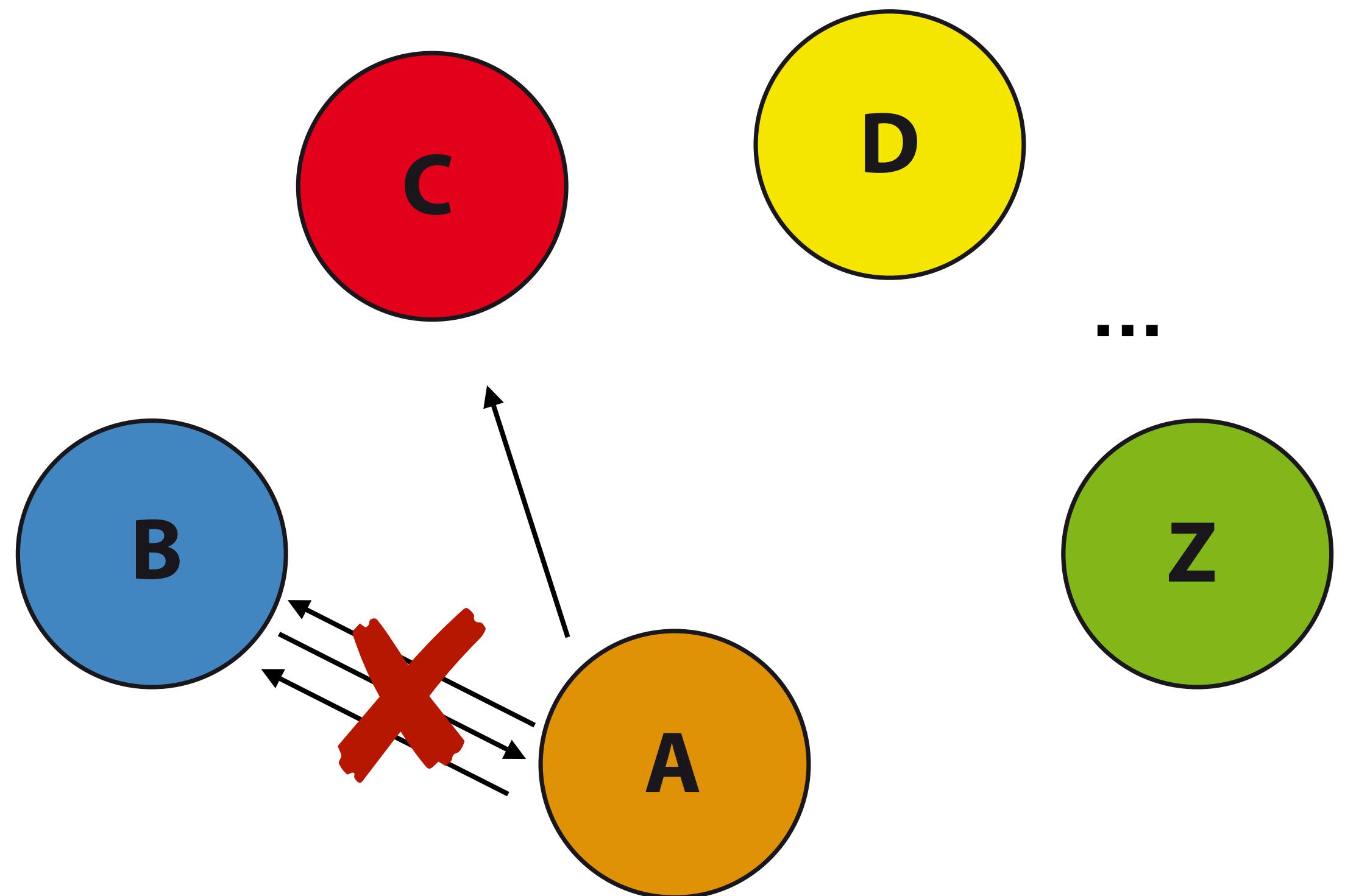
Information propagation (2/2)



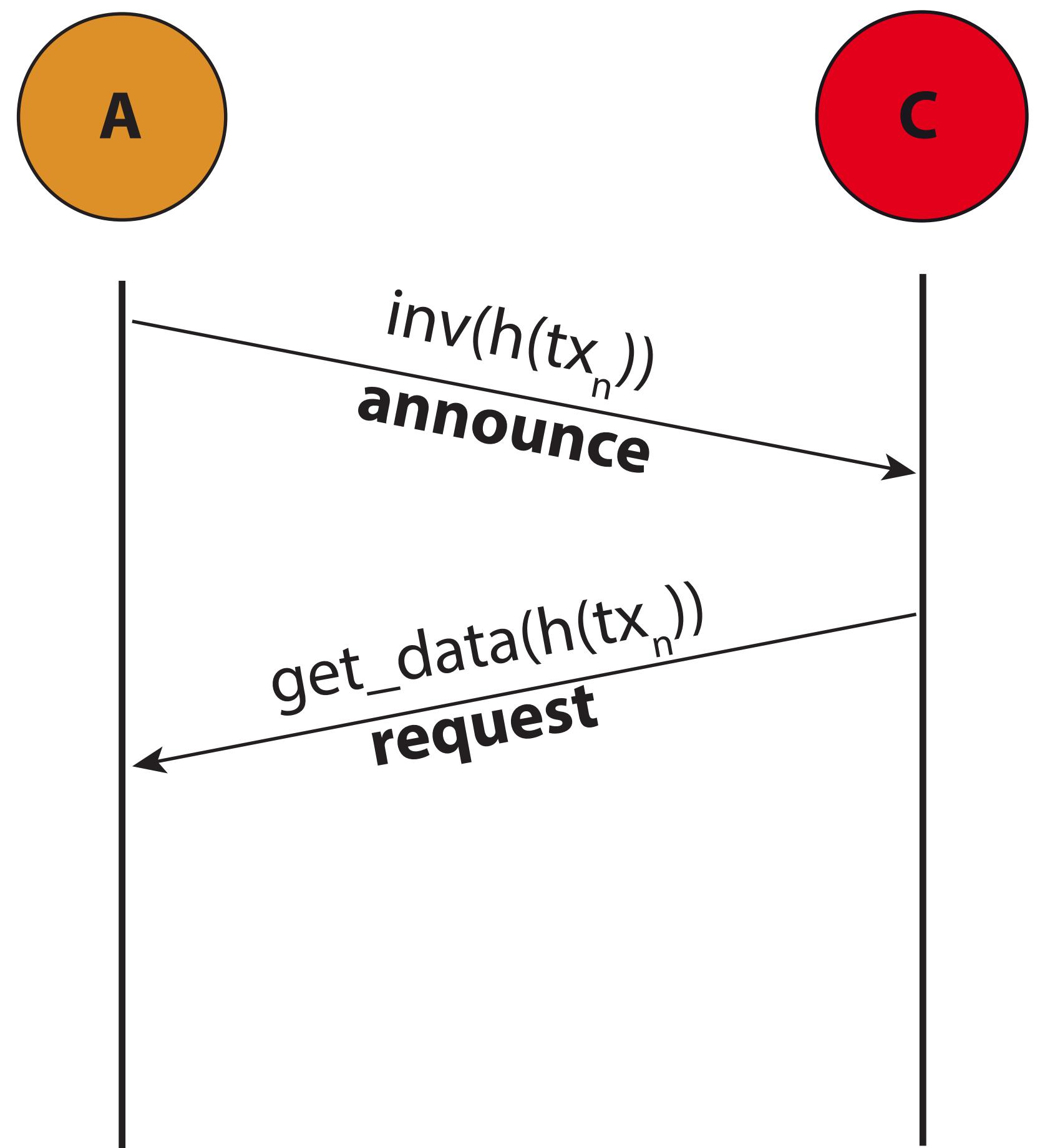
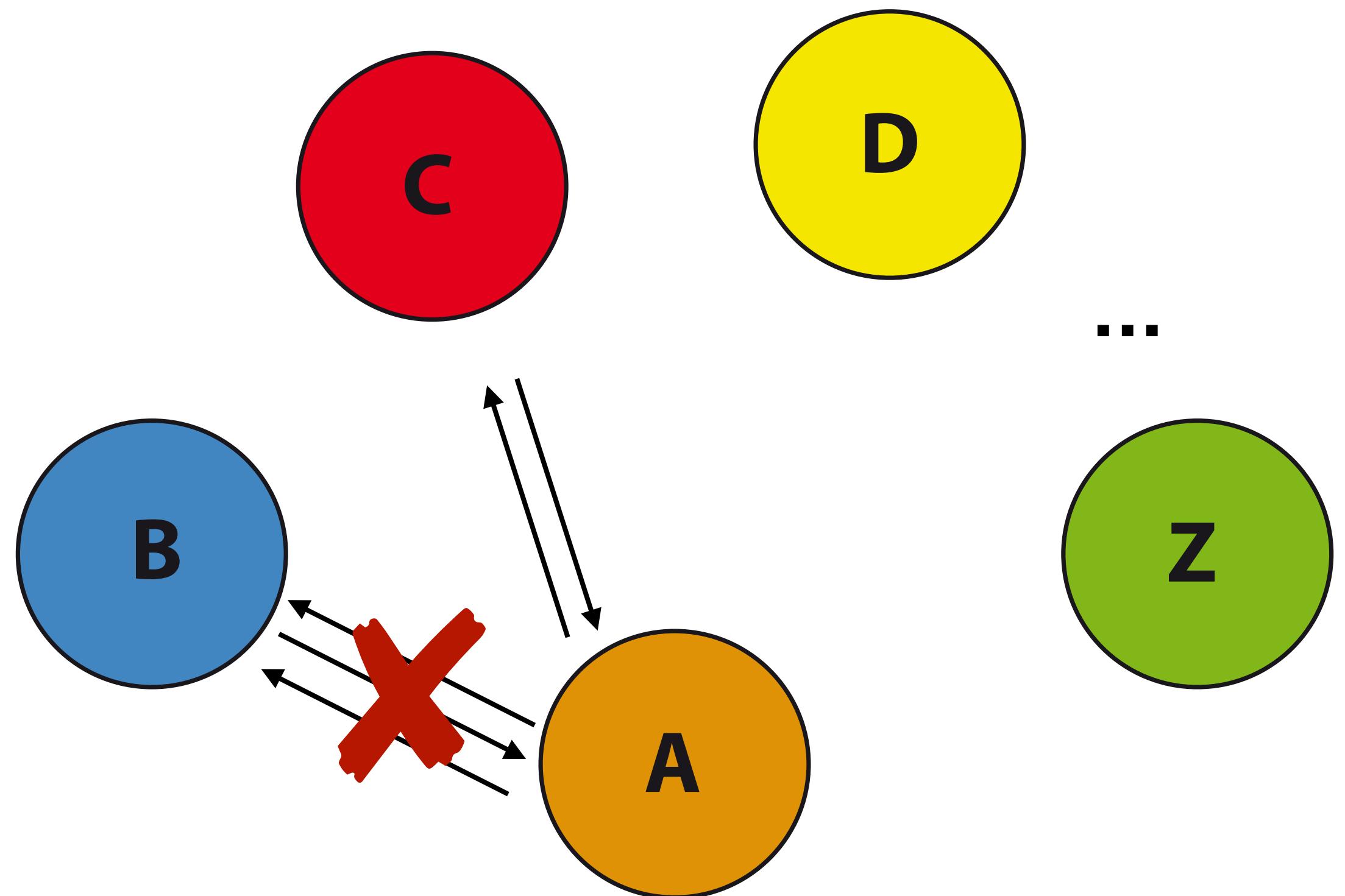
Information propagation (2/2)



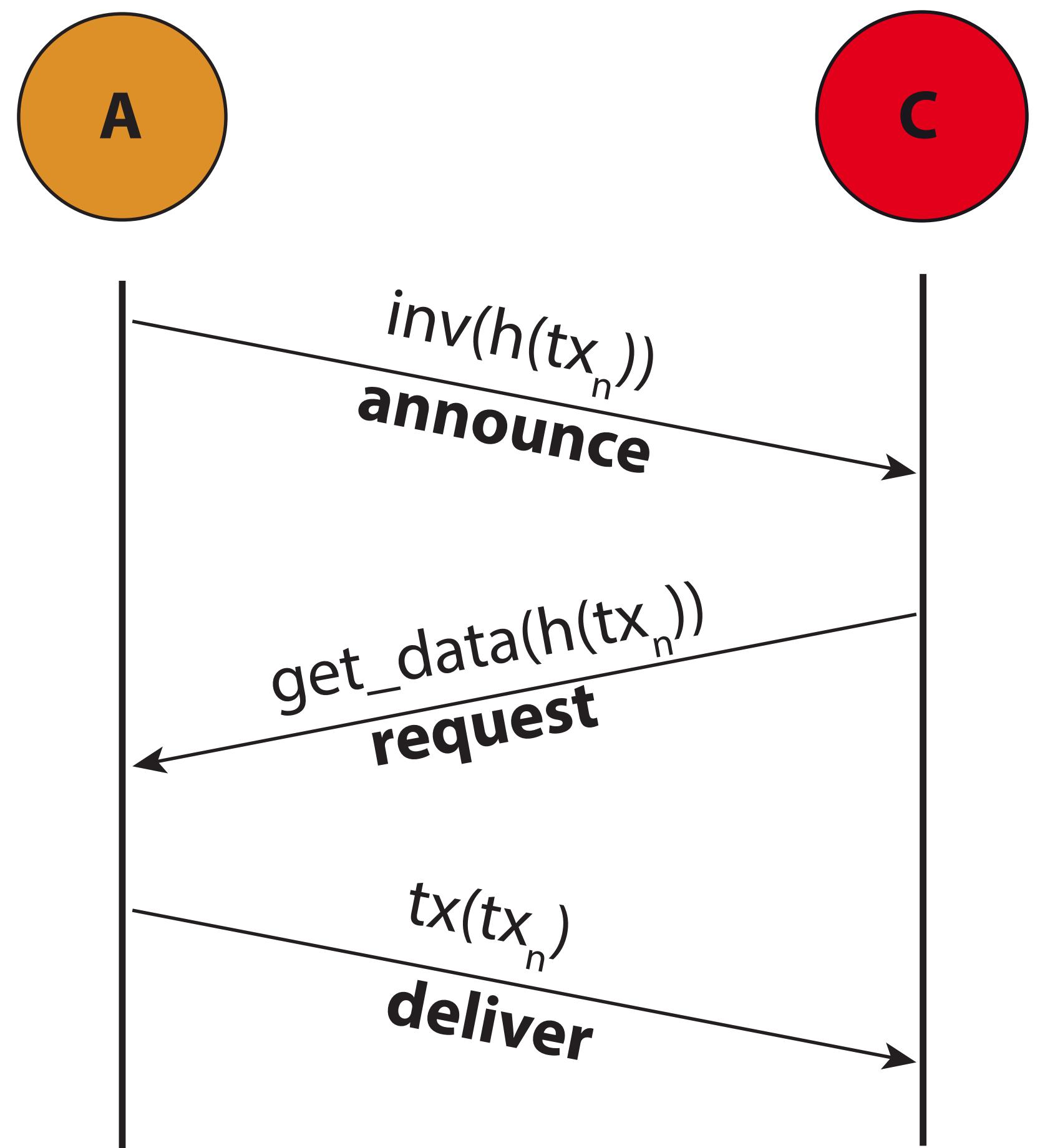
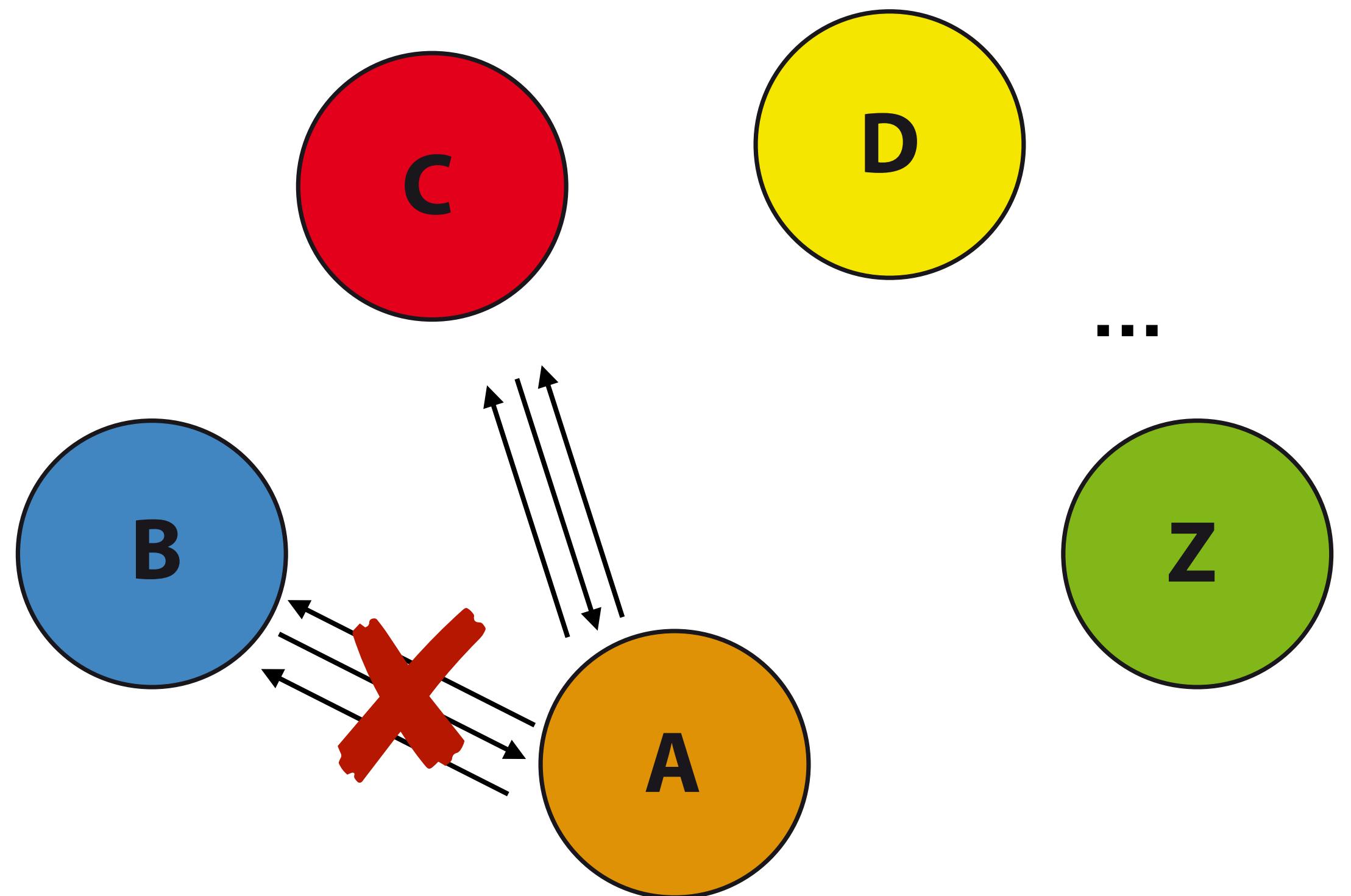
Information propagation (2/2)



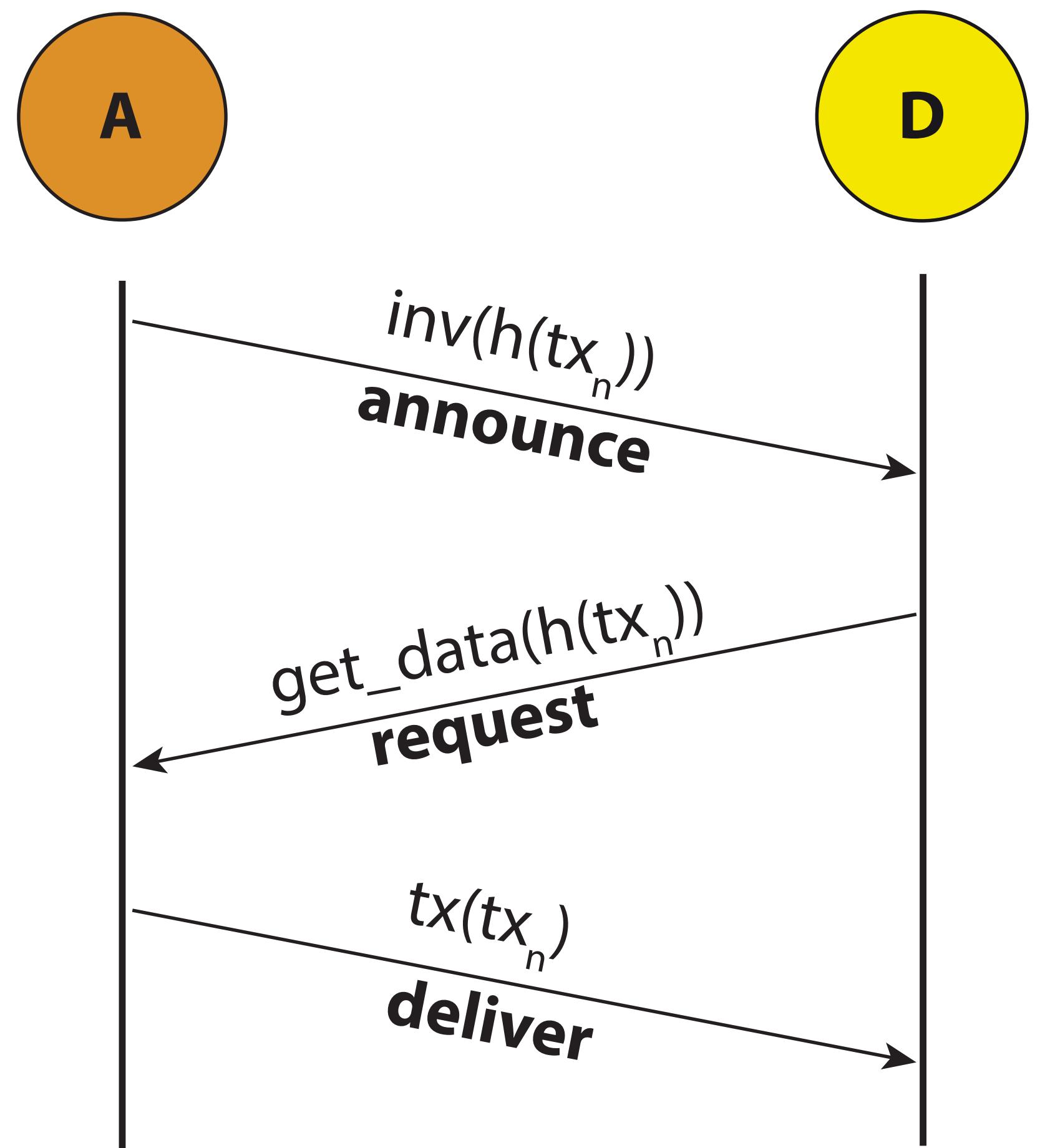
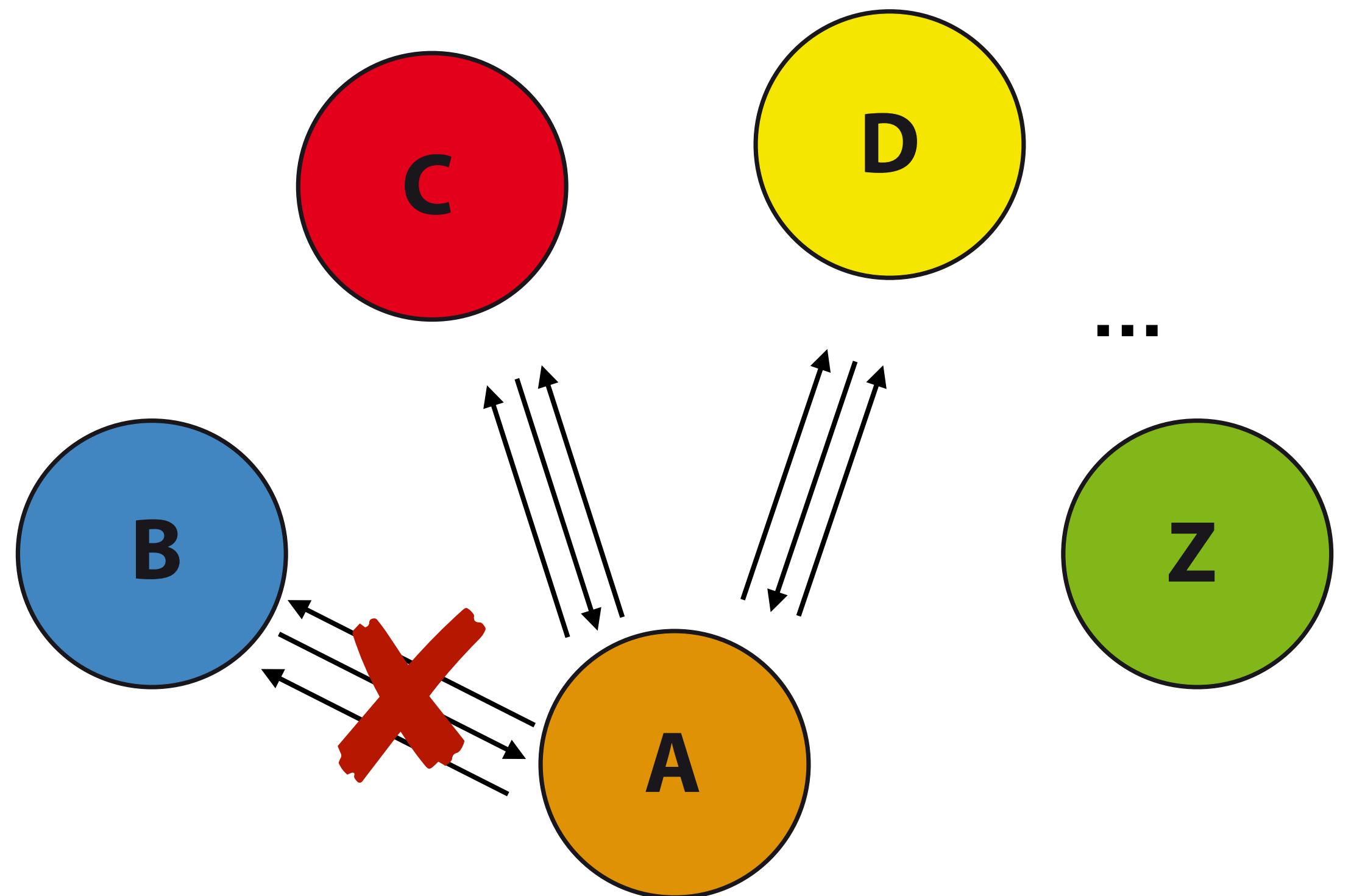
Information propagation (2/2)



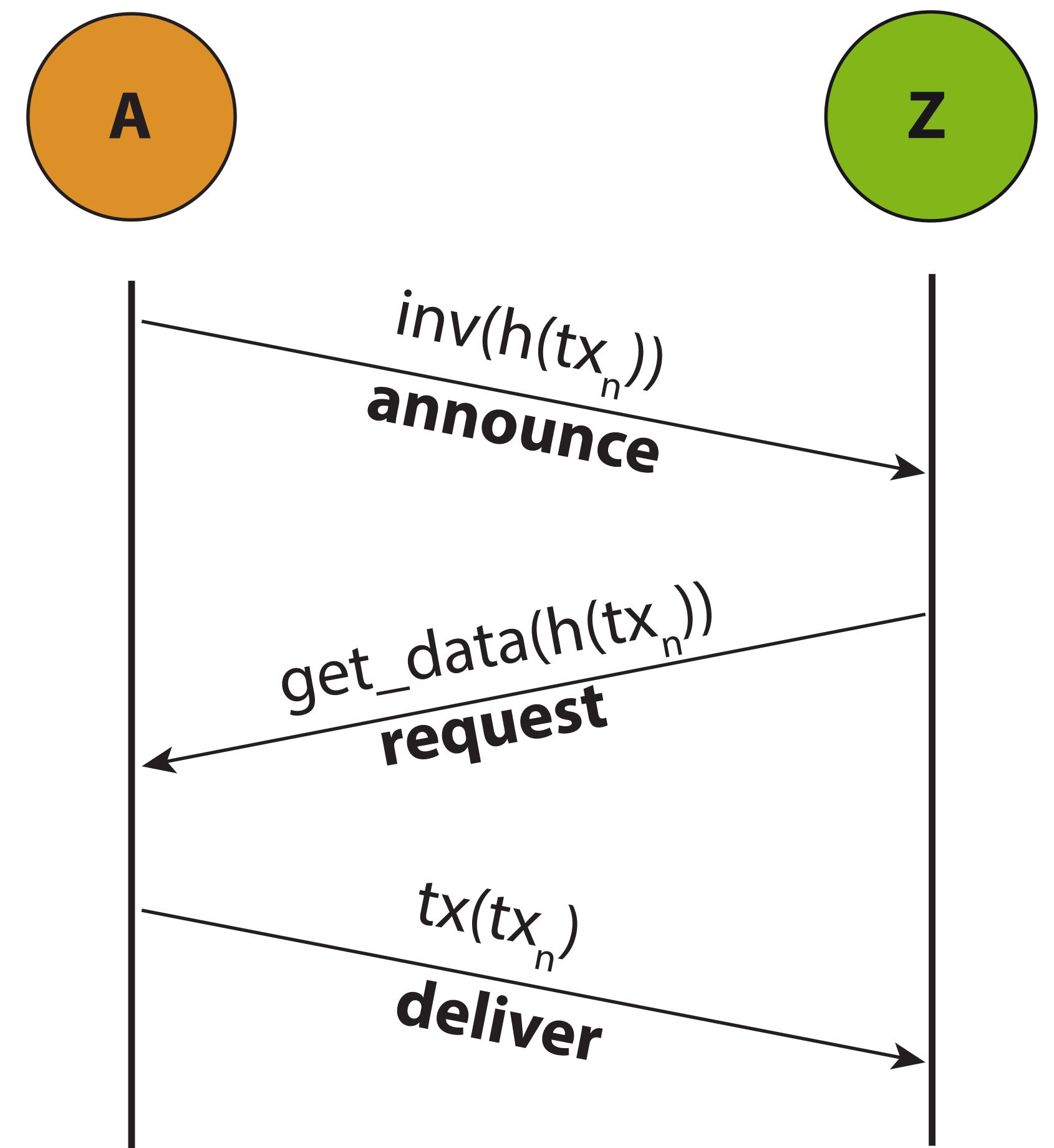
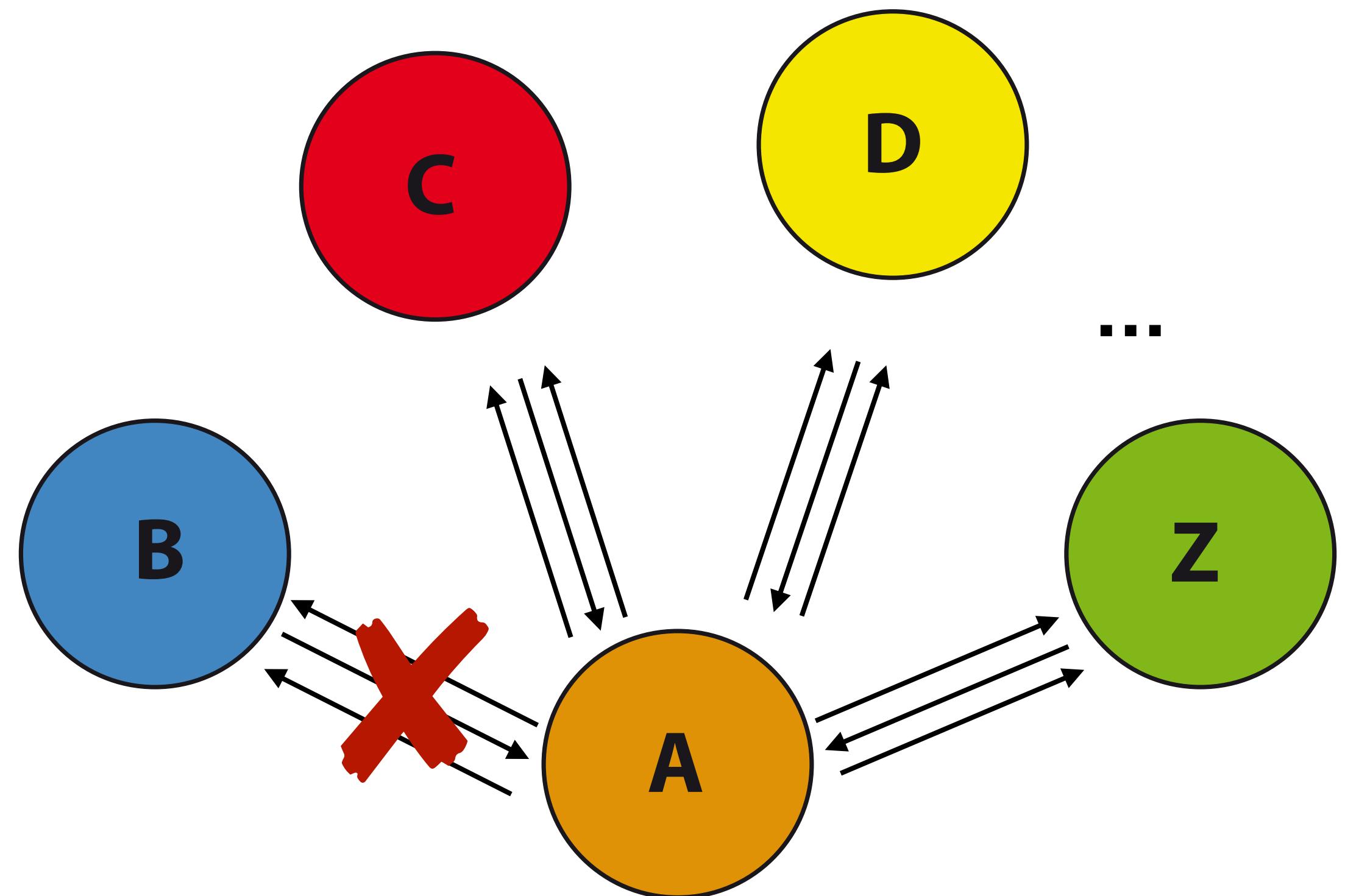
Information propagation (2/2)



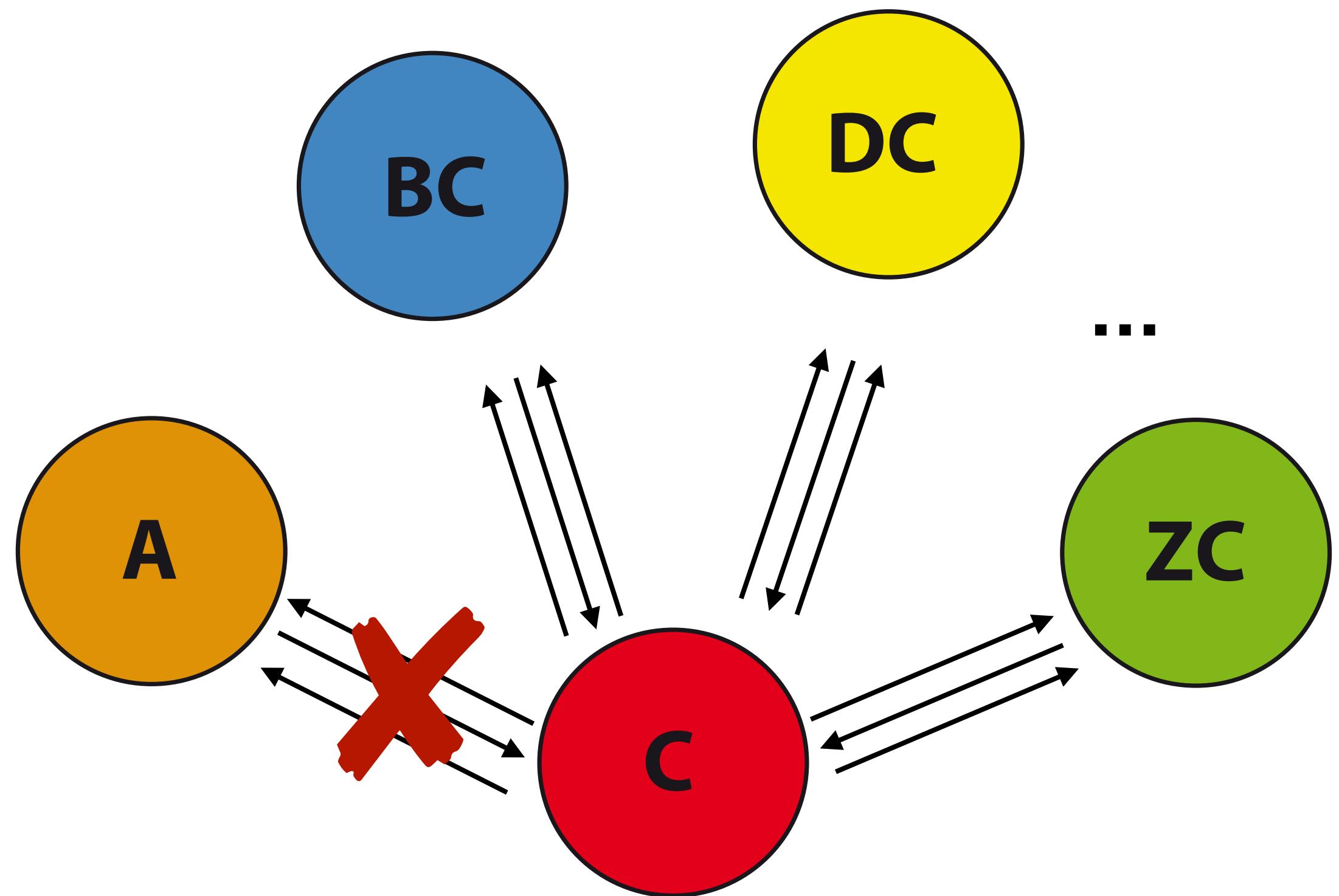
Information propagation (2/2)



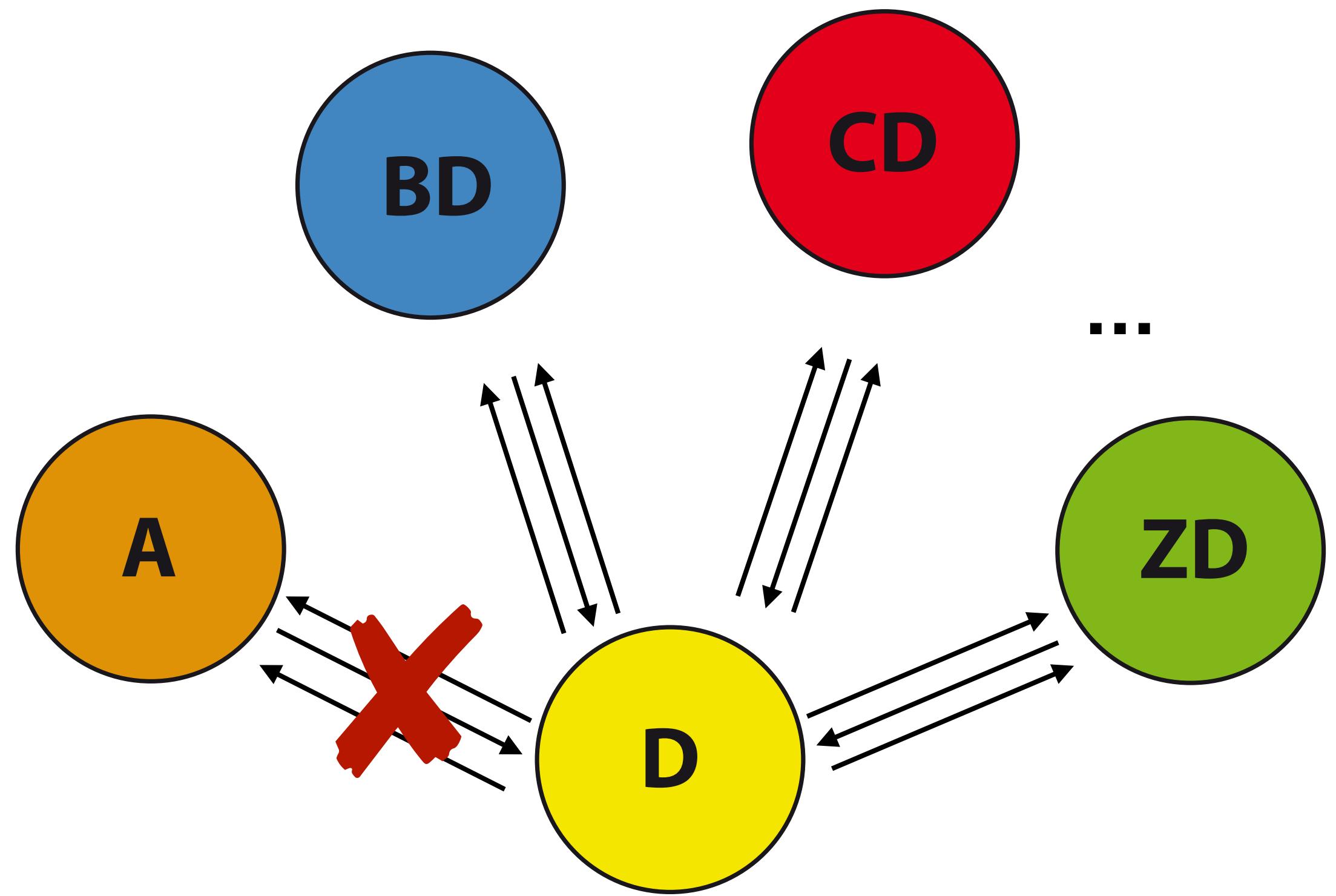
Information propagation (2/2)



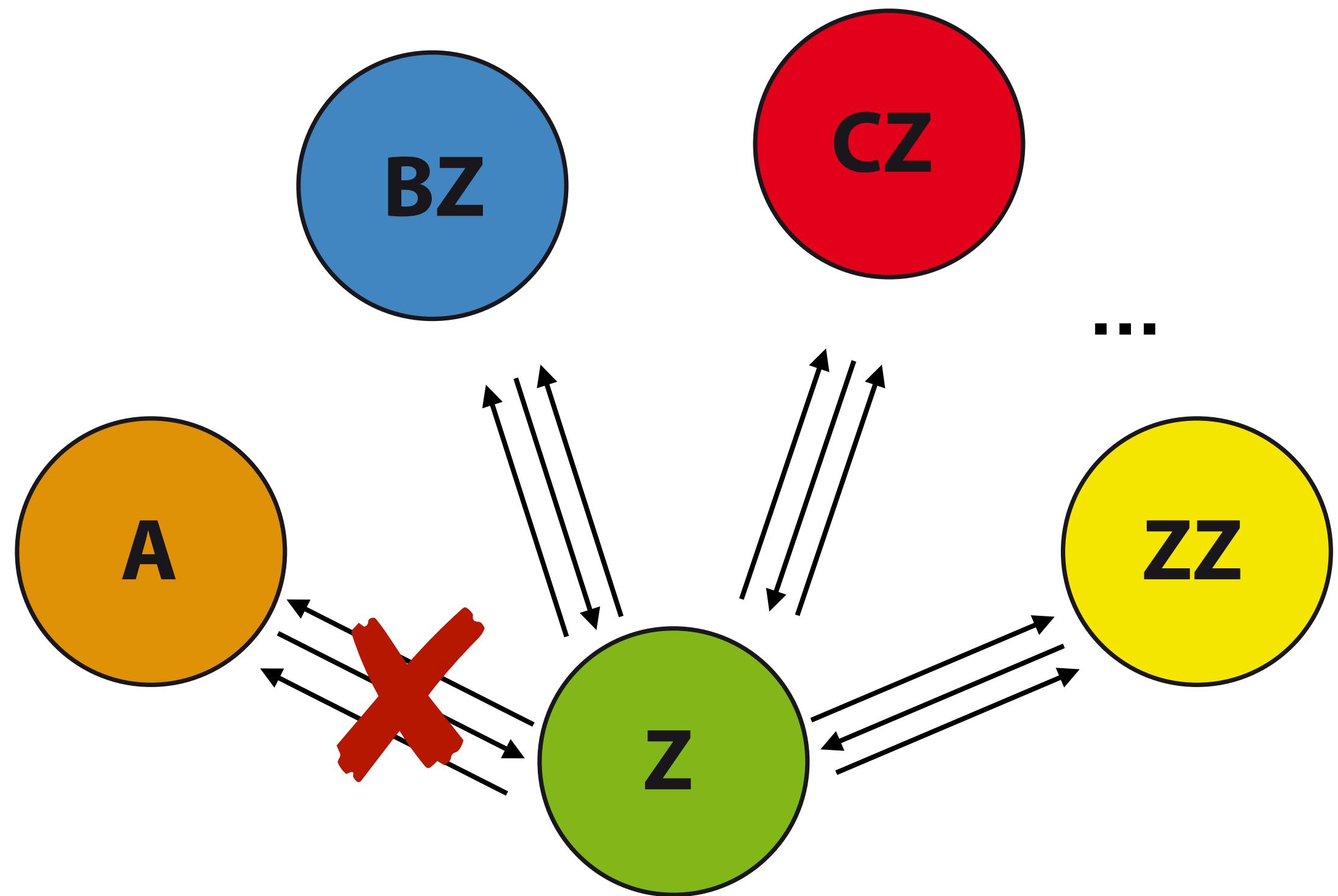
Information propagation (2/2)



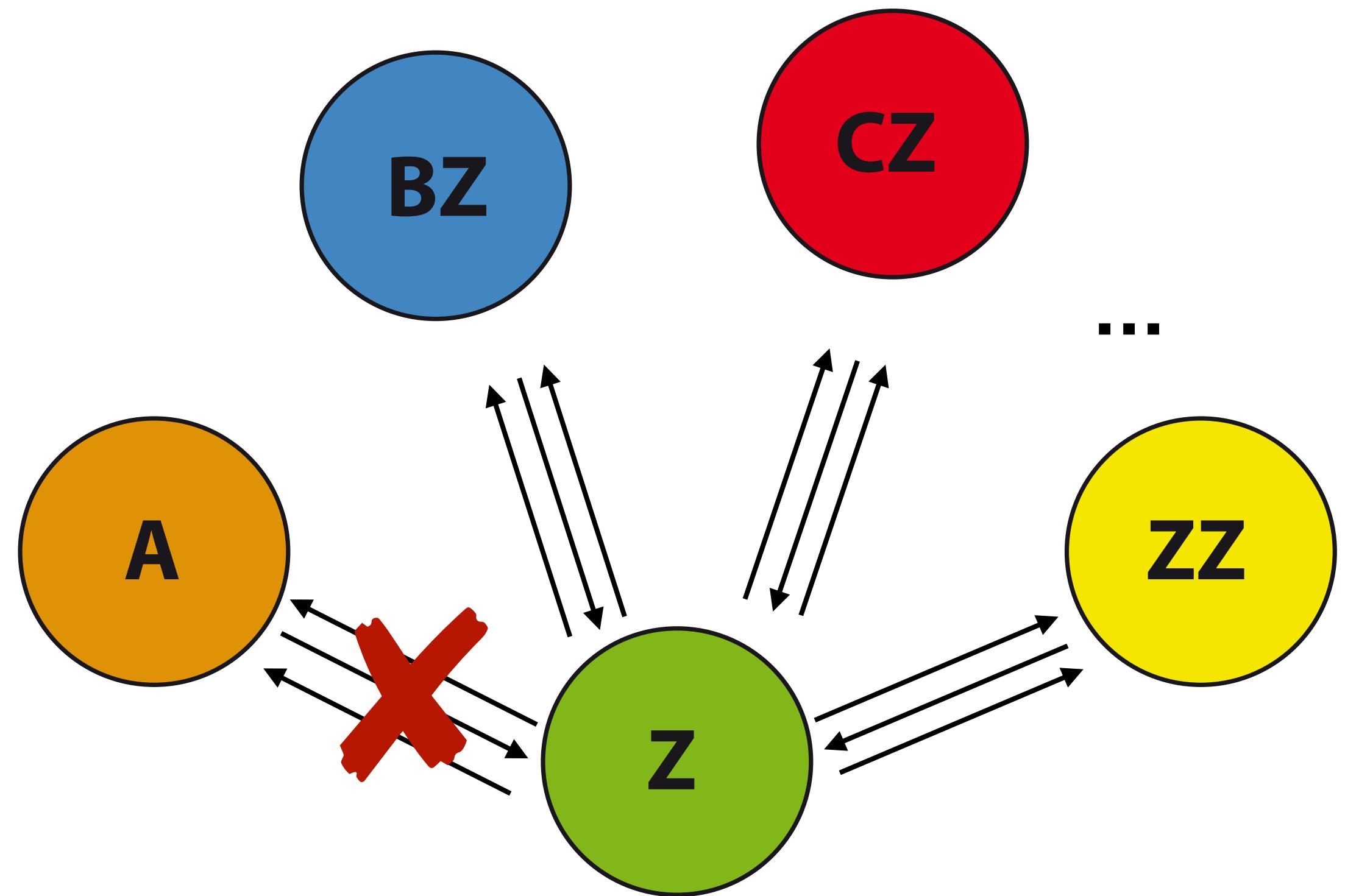
Information propagation (2/2)



Information propagation (2/2)

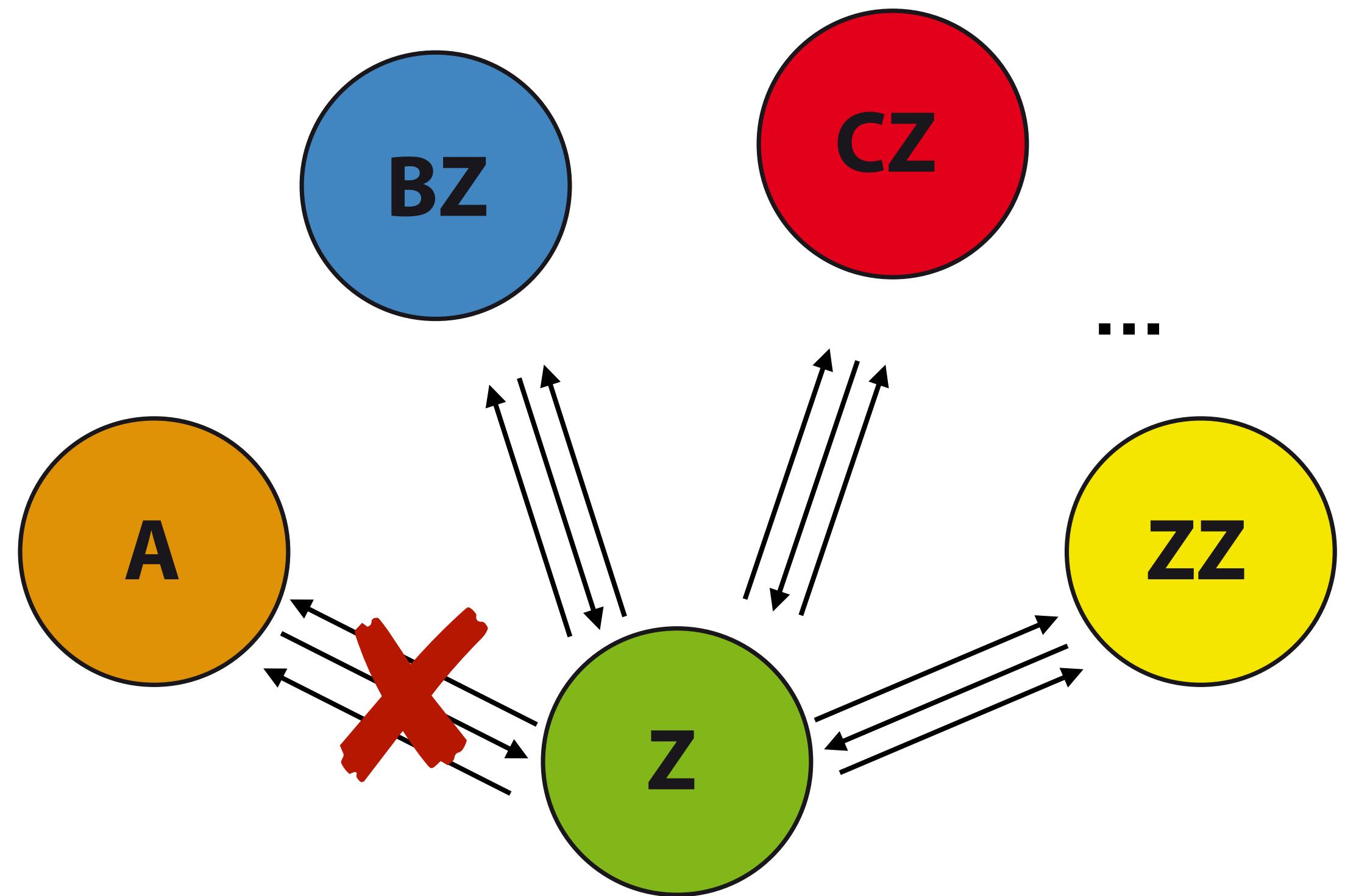


Information propagation (2/2)



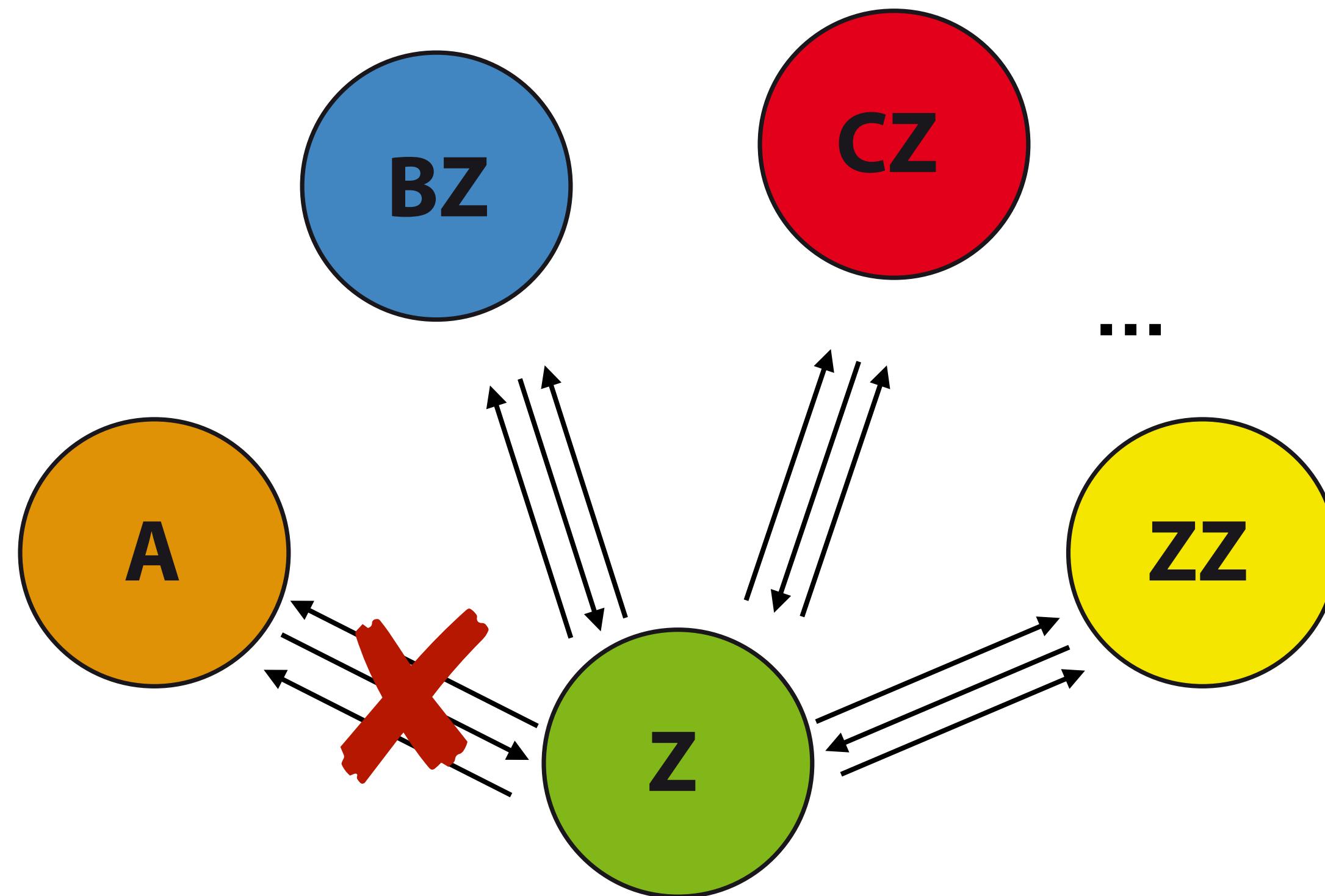
- And so on and so forth until all the nodes are reached

Information propagation (2/2)



- 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 (2/2)



- 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 exact same procedure applies for blocks



Implications

- The bigger the network the more it takes for an item to propagate (**this can be counterintuitive**)
- The bigger an item, the more it takes to validate and propagate
- Long propagation times (**for blocks**) imply bigger fork rate

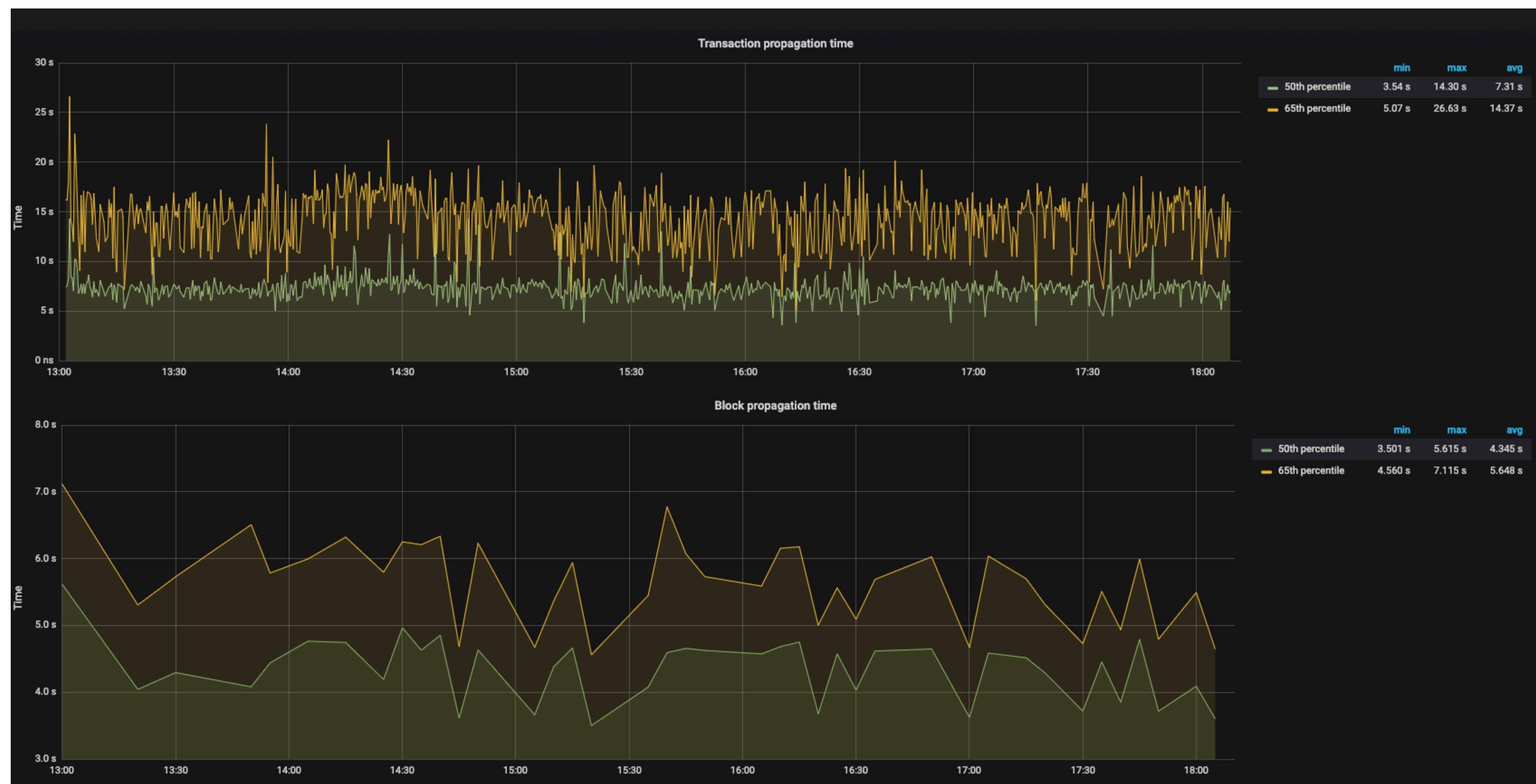
Implications

- The bigger the network the more it takes for an item to propagate (**this can be counterintuitive**)
- The bigger an item, the more it takes to validate and propagate
- Long propagation times (**for blocks**) imply bigger fork rate





Bitcoin testnet data propagation times



source: charts.satoshi.uab.cat

BREAK TIME



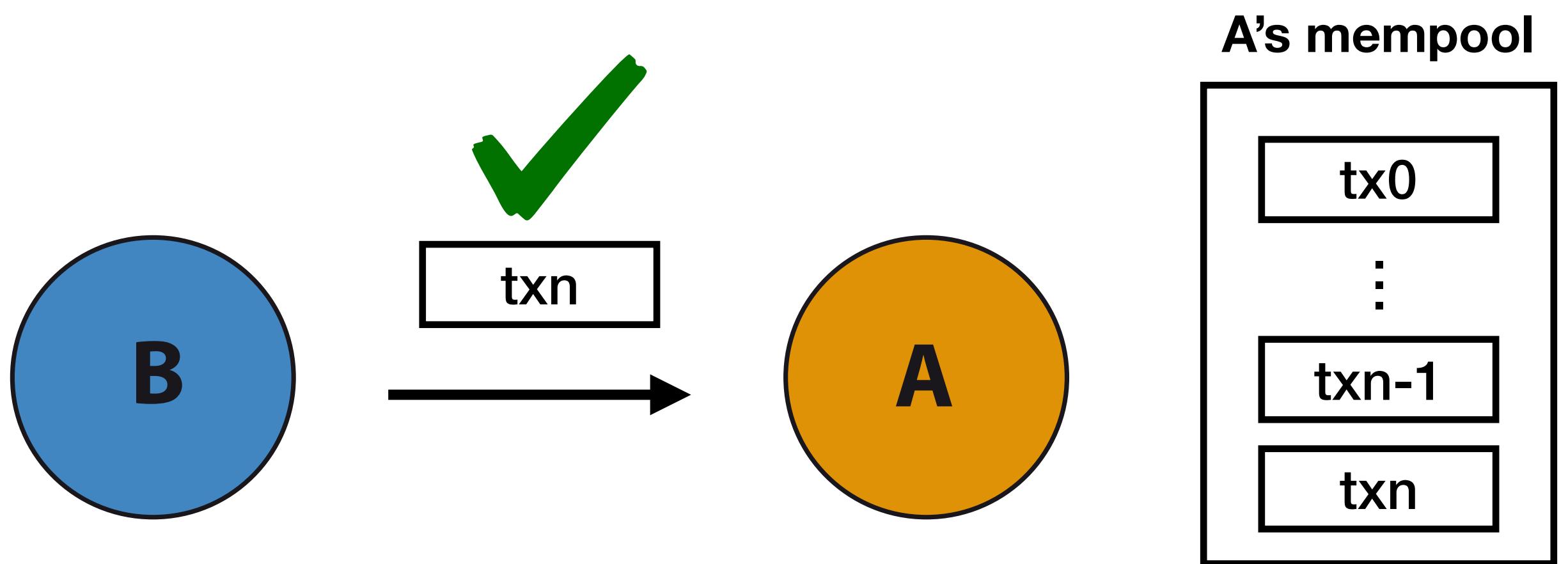
0-conf transactions and double-spending



0-confirmation transactions (1/2)

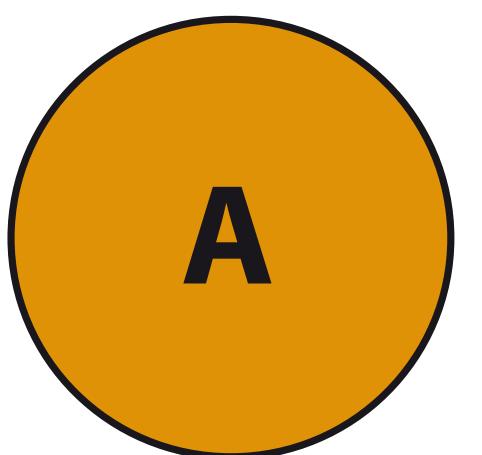
- **0-conf** transactions / **unconfirmed** transactions are those that are not part of the blockchain
- They can be valid transactions that are waiting to be included
- But they can also be invalid (e.g: **double-spending transactions, etc**)
- Different nodes can have conflicting version of the “**same transaction**”

0-confirmation transactions (2/2)

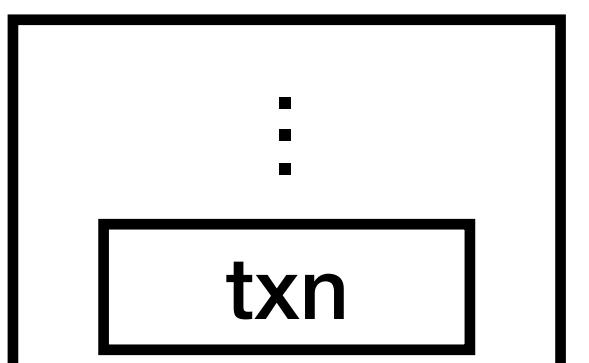




Confirmed transactions

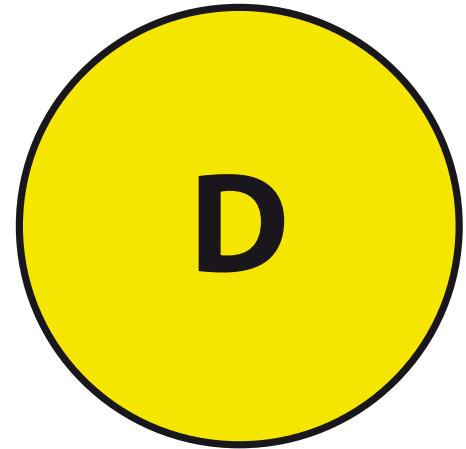
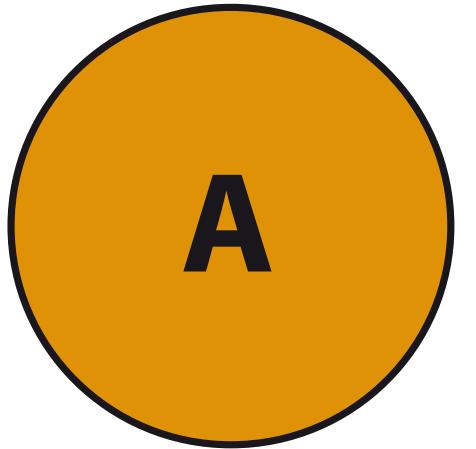


A's mempool

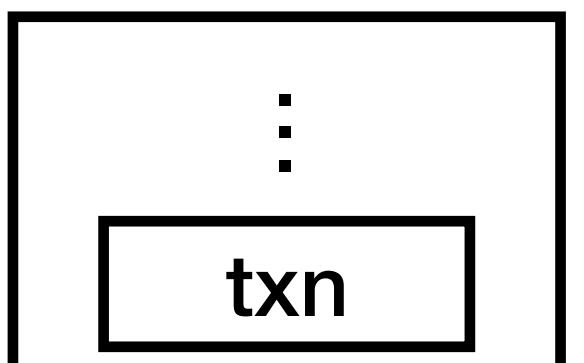




Confirmed transactions

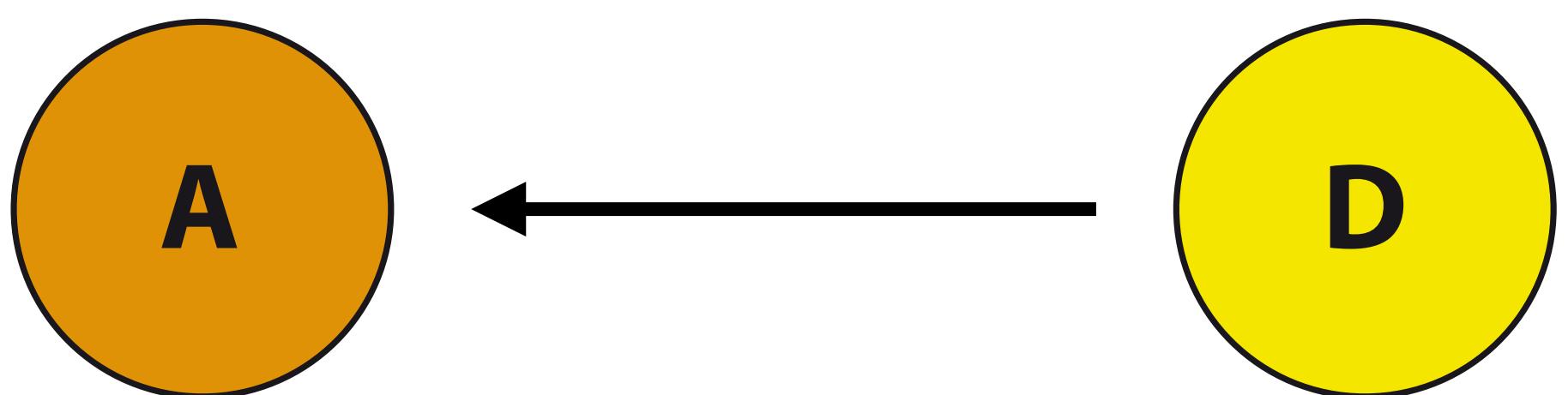


A's mempool

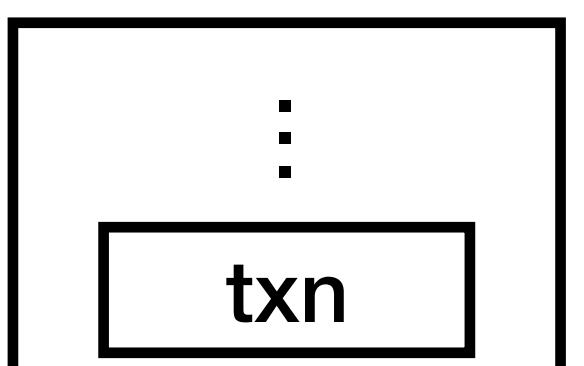




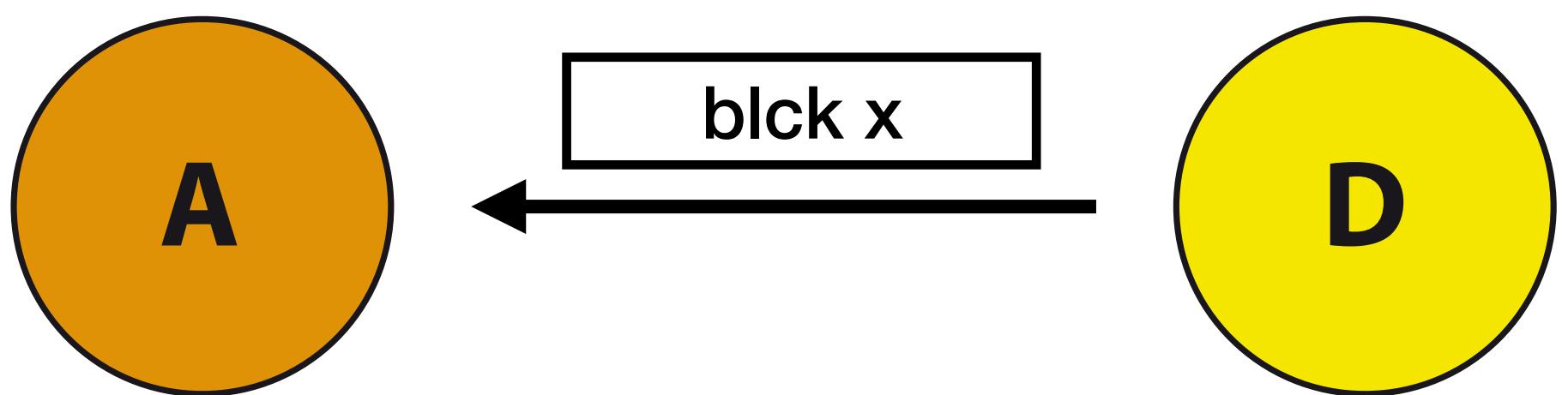
Confirmed transactions



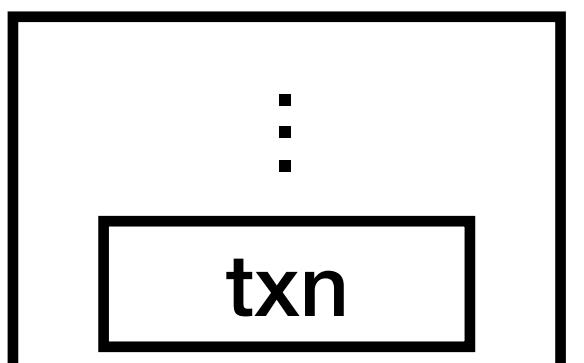
A's mempool



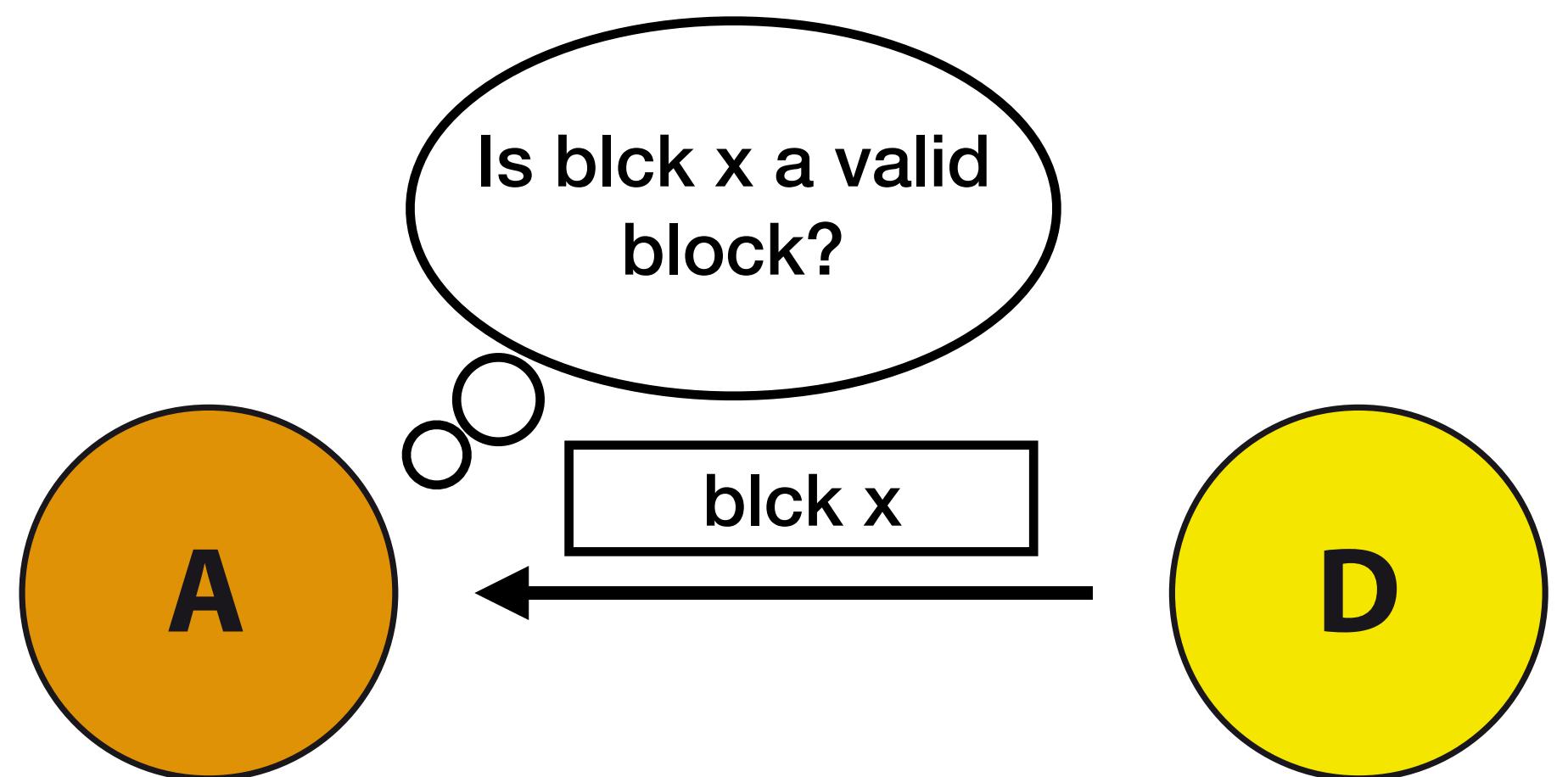
Confirmed transactions



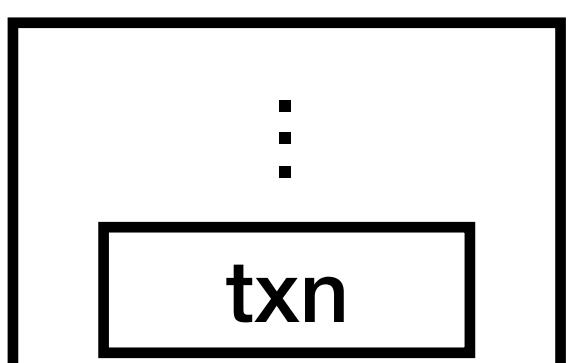
A's mempool



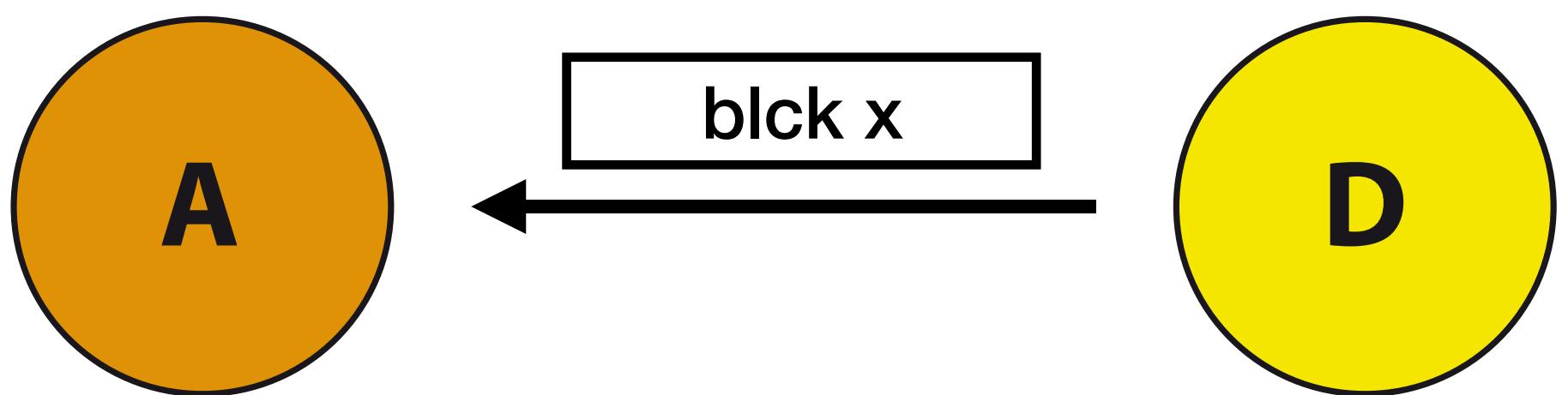
Confirmed transactions



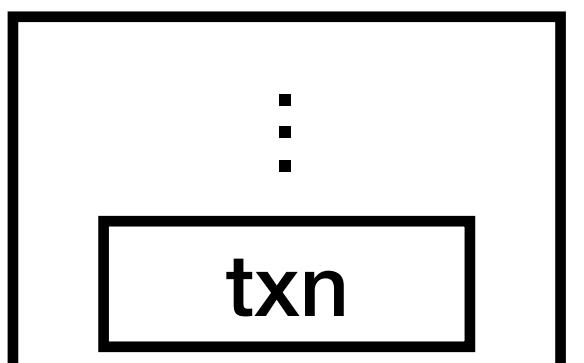
A's mempool



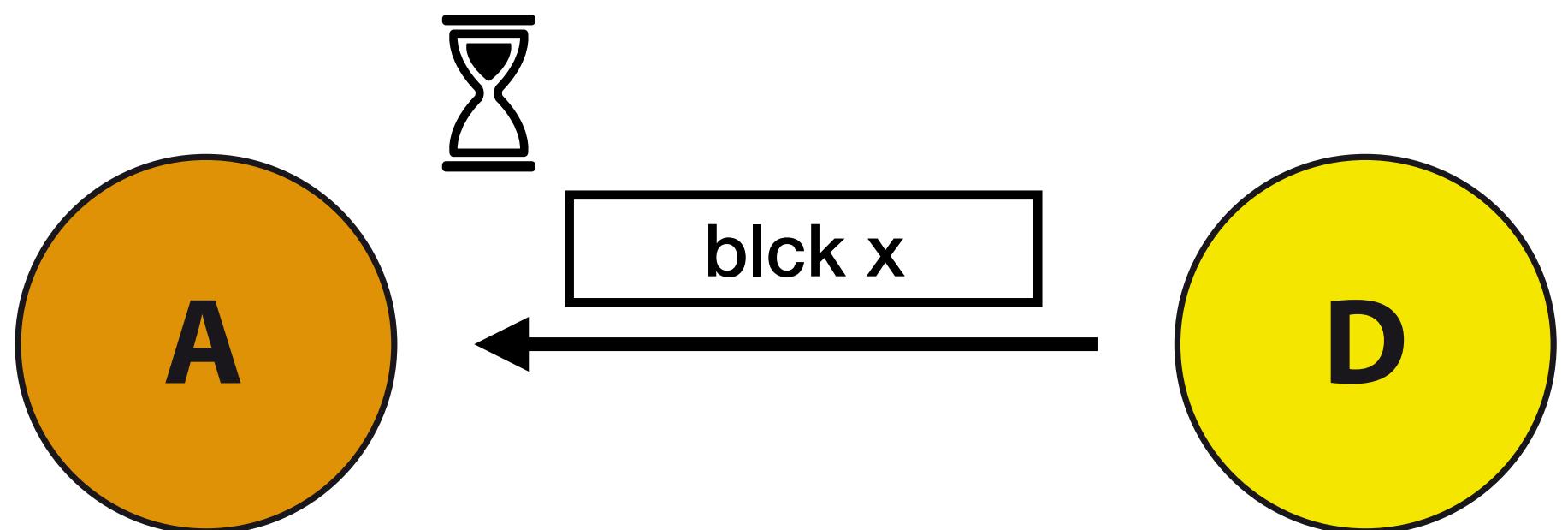
Confirmed transactions



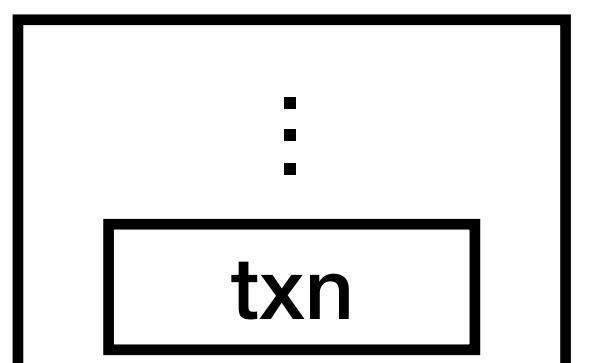
A's mempool



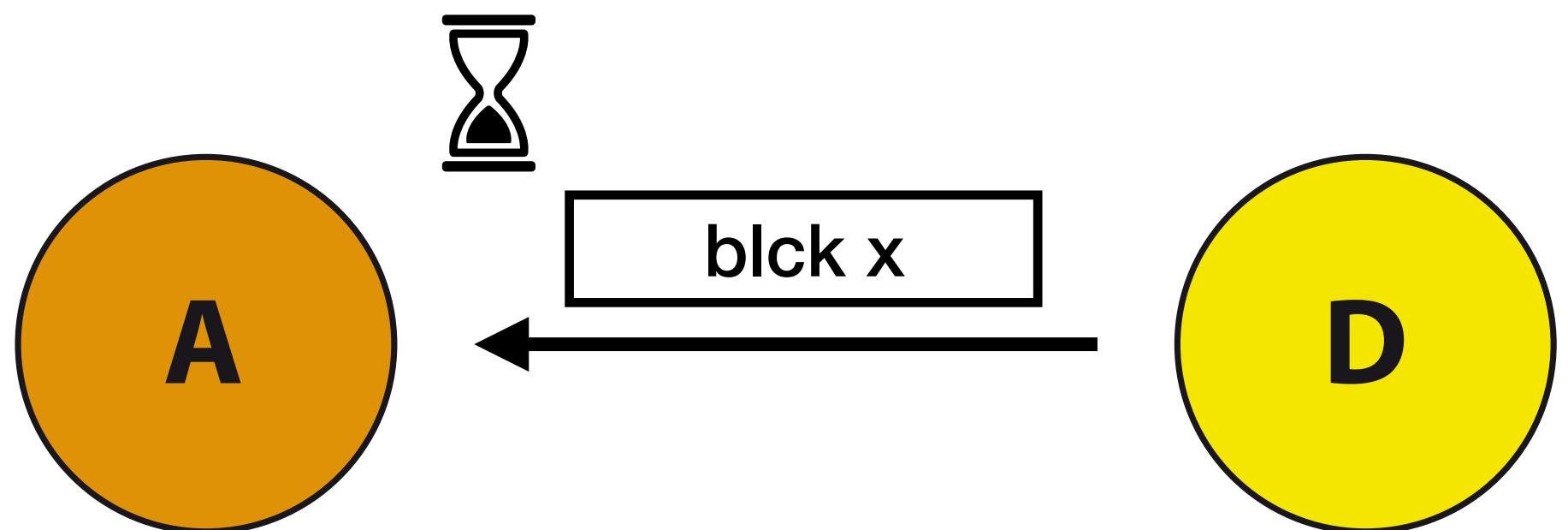
Confirmed transactions



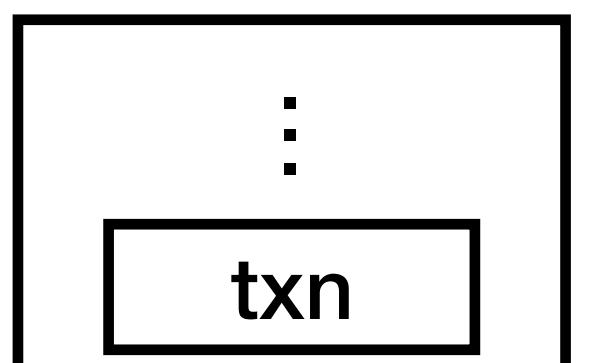
A's mempool



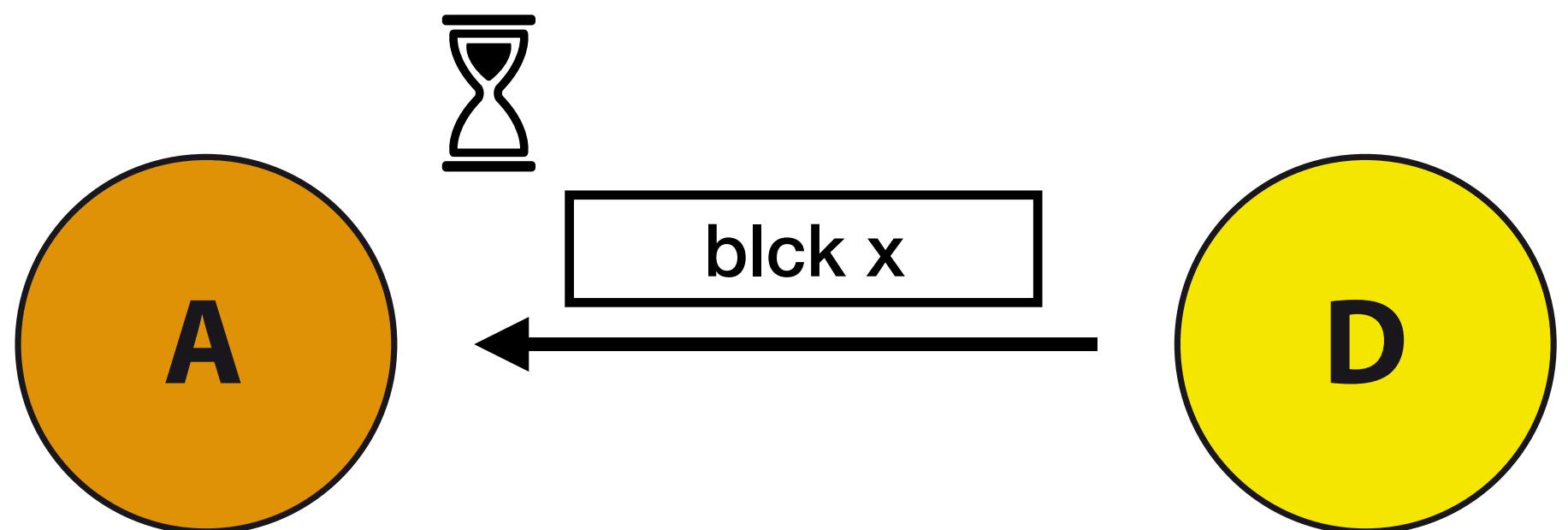
Confirmed transactions



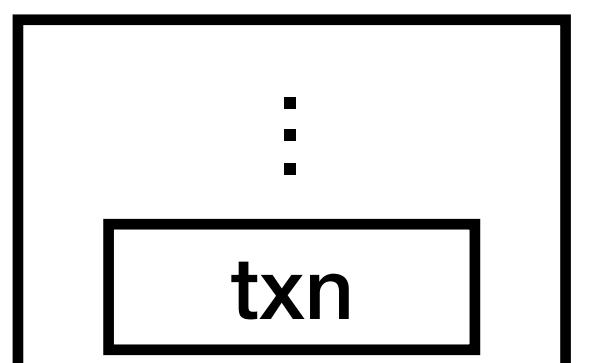
A's mempool



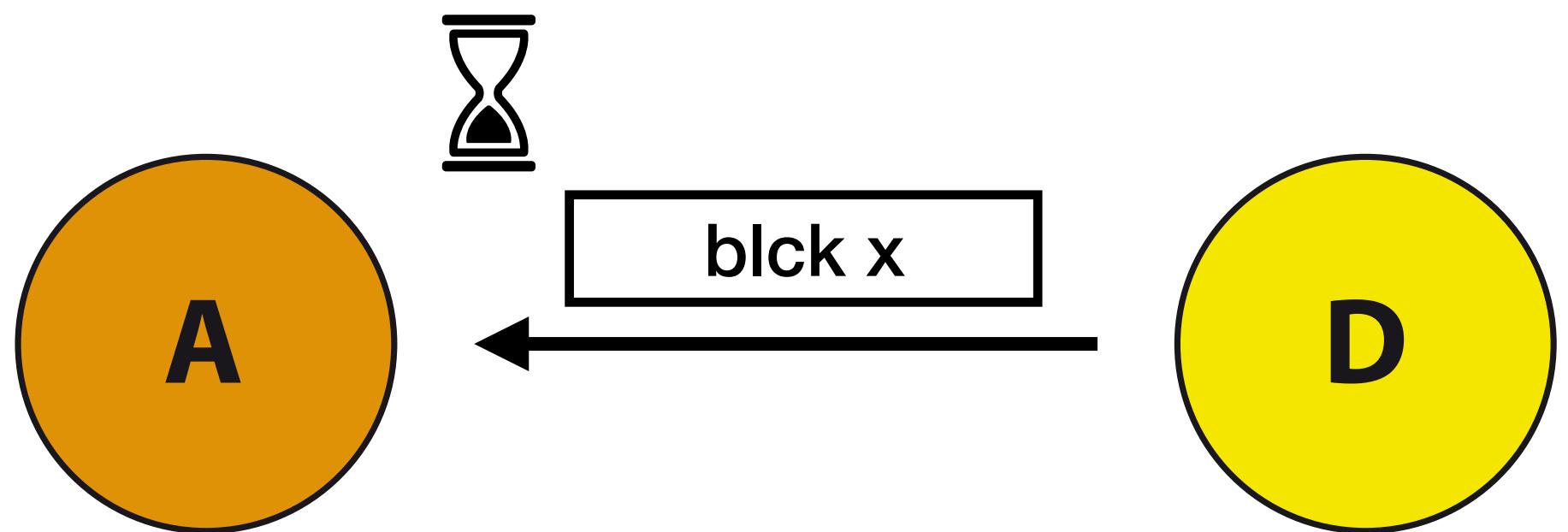
Confirmed transactions



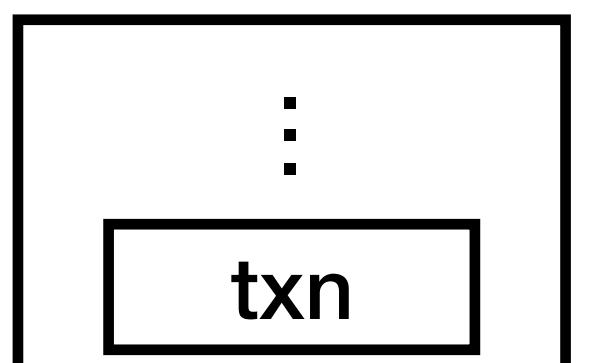
A's mempool



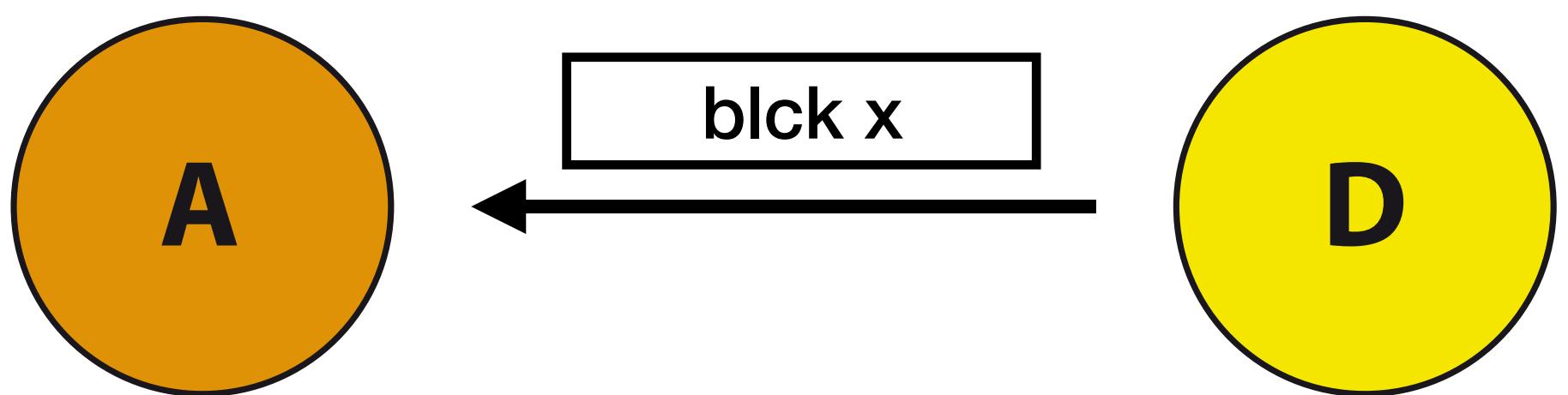
Confirmed transactions



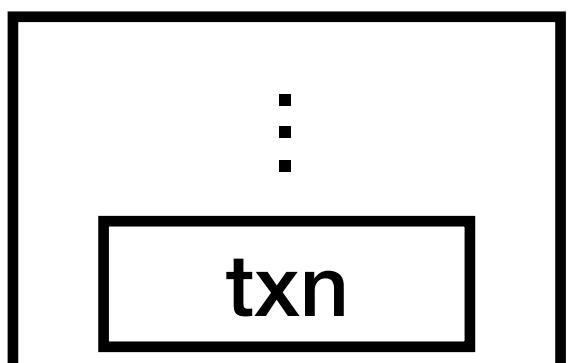
A's mempool



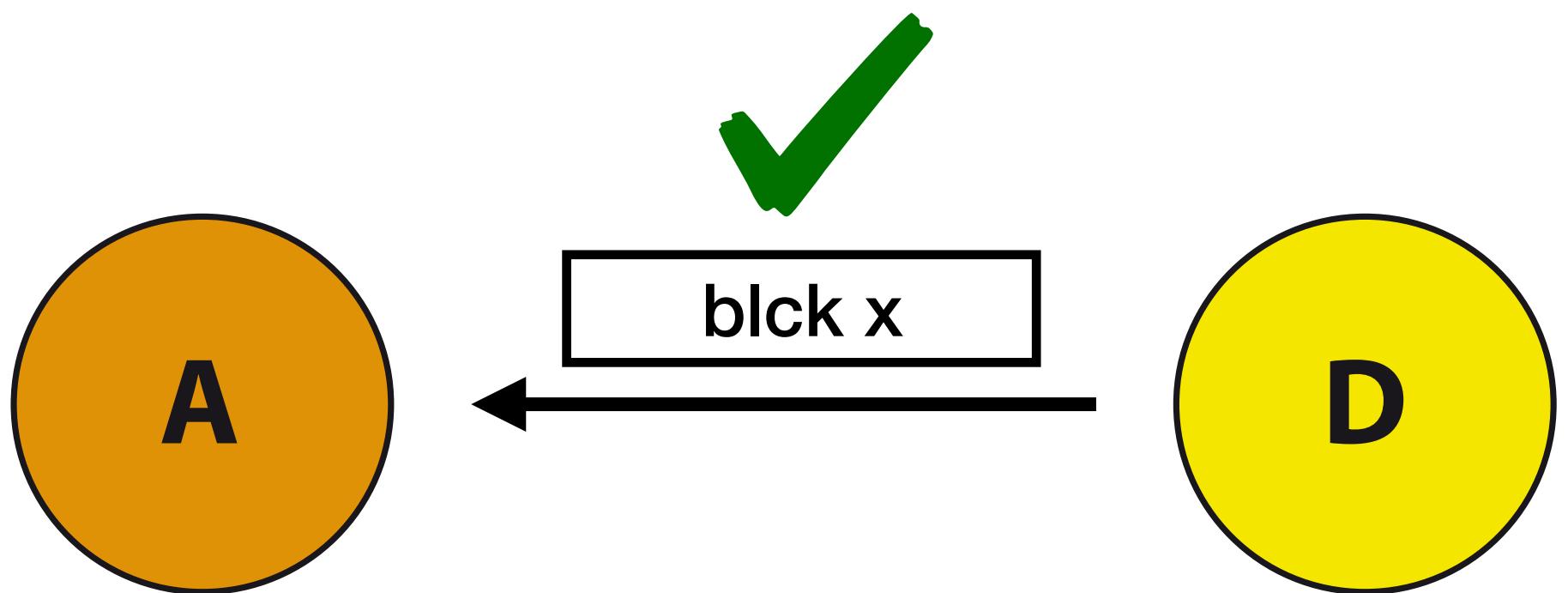
Confirmed transactions



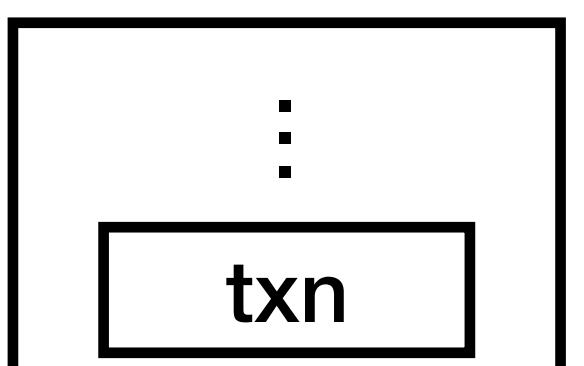
A's mempool



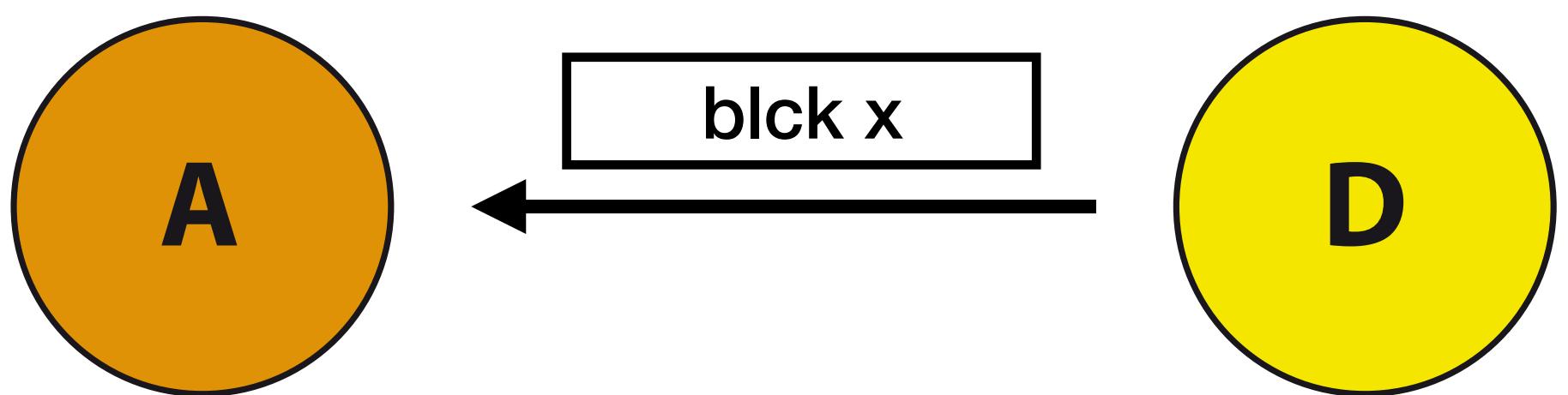
Confirmed transactions



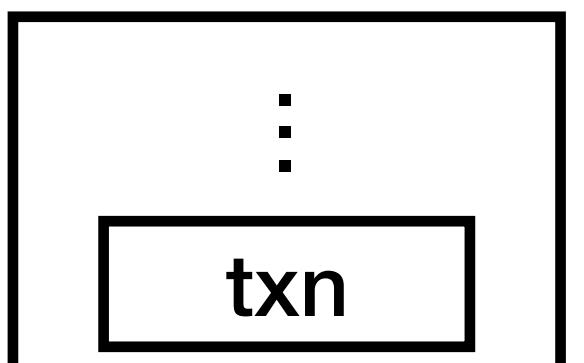
A's mempool



Confirmed transactions

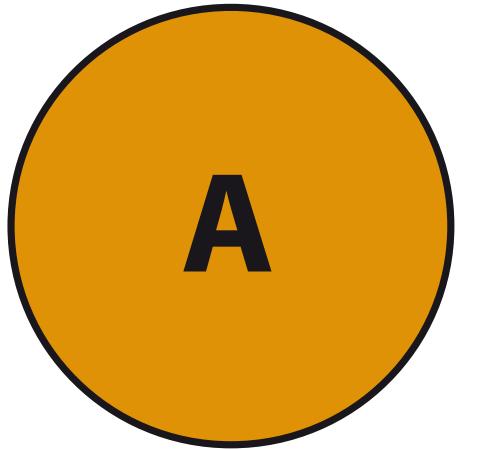


A's mempool

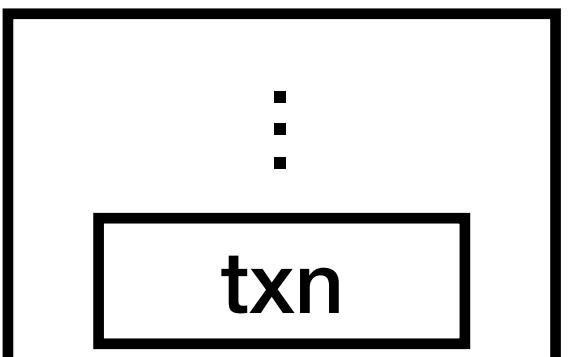




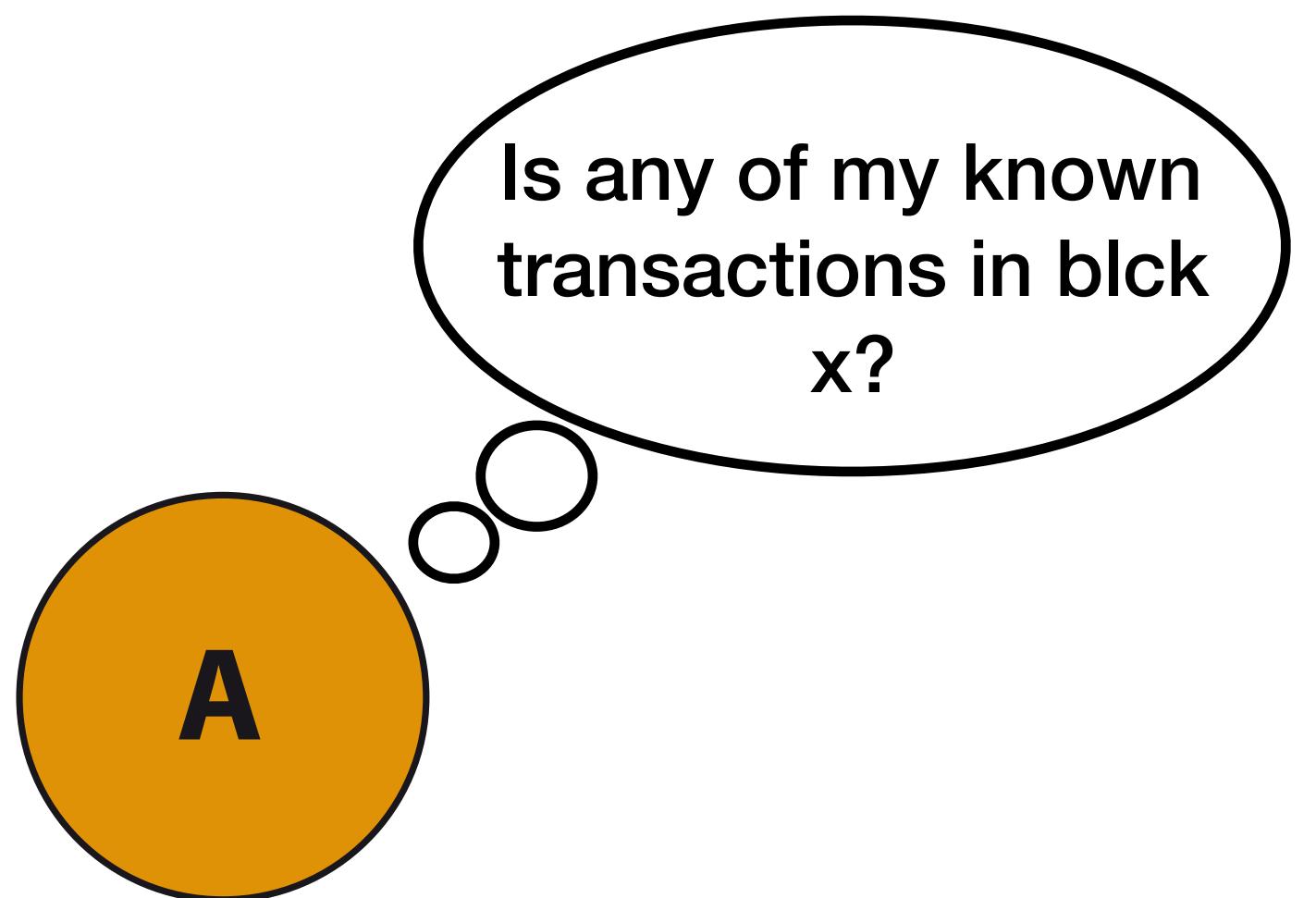
Confirmed transactions



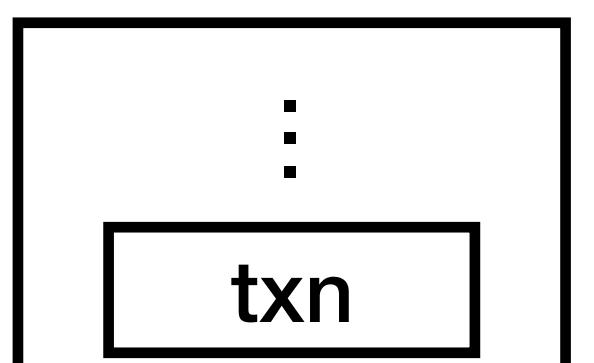
A's mempool



Confirmed transactions

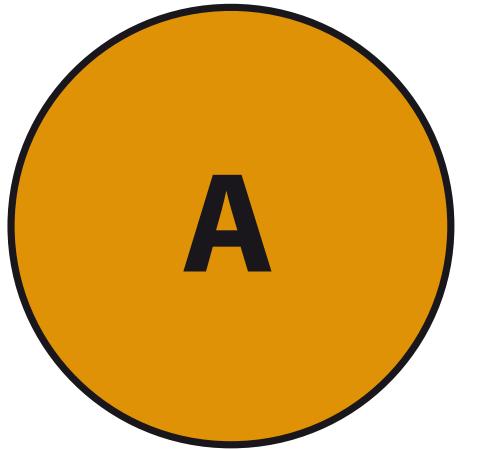


A's mempool

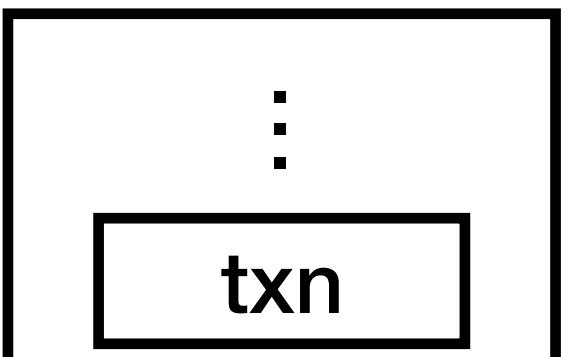




Confirmed transactions

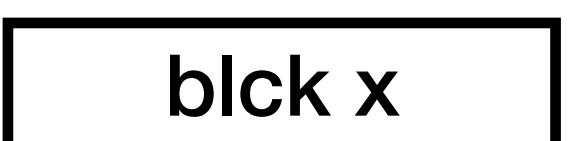
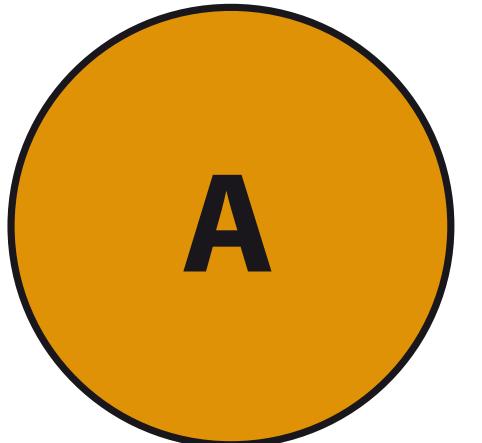


A's mempool

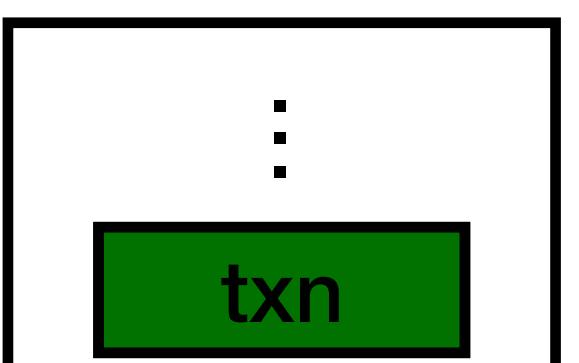




Confirmed transactions

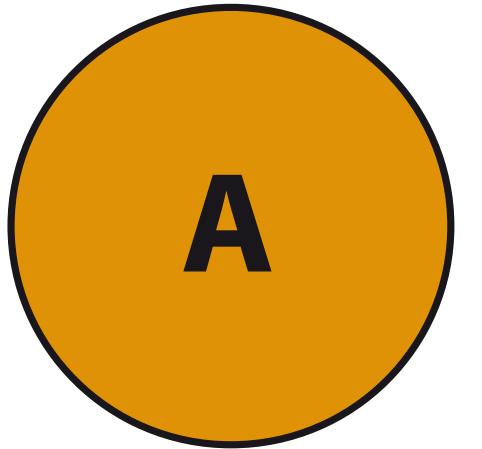


A's mempool

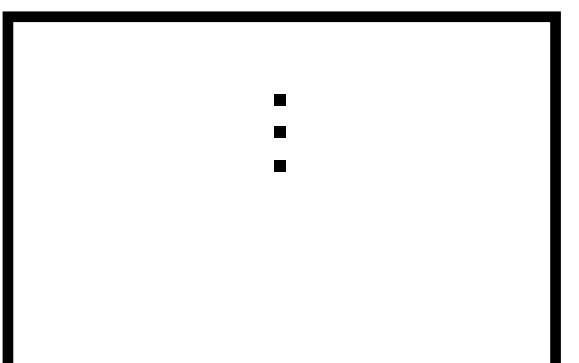




Confirmed transactions

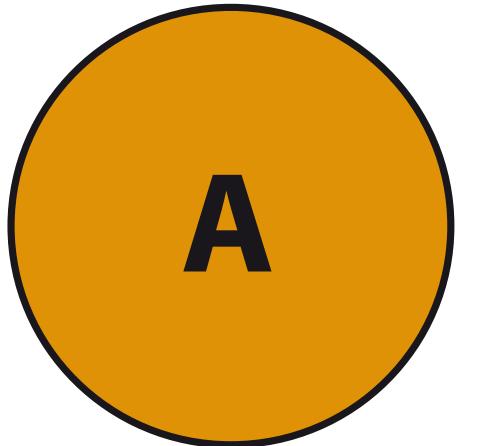


A's mempool

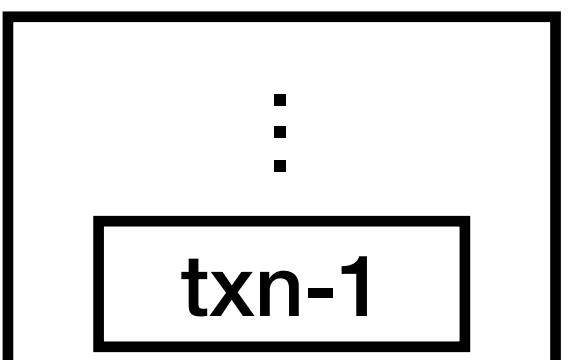




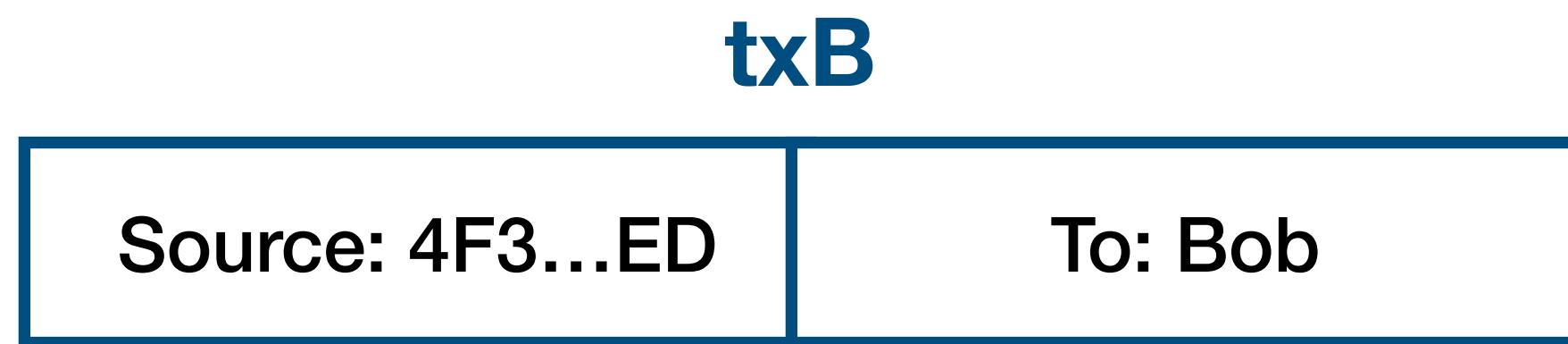
Confirmed transactions



A's mempool

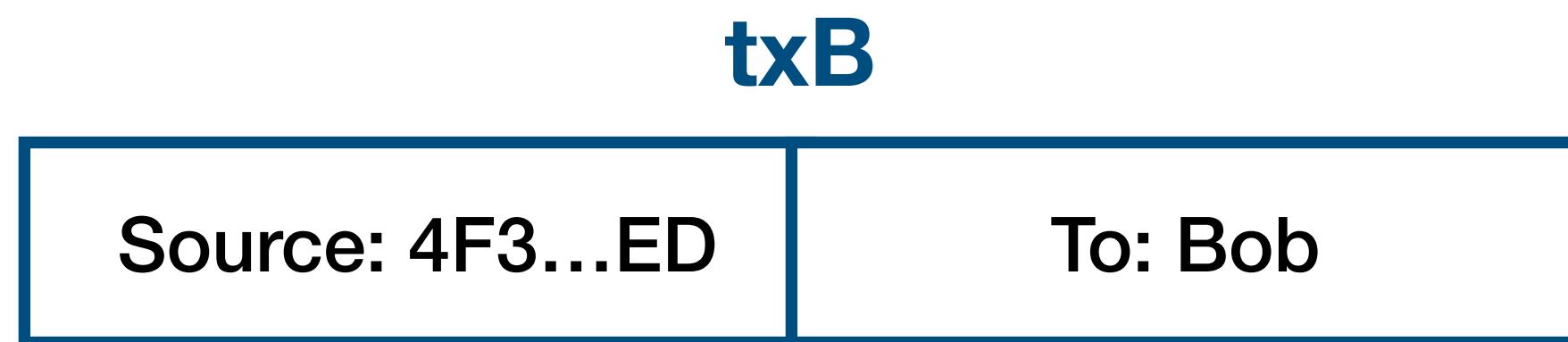
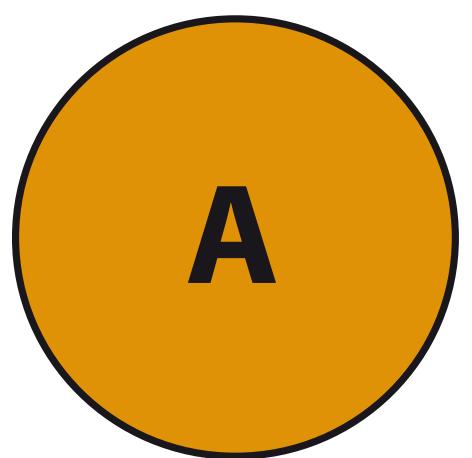


Double-spending transactions (1/2)

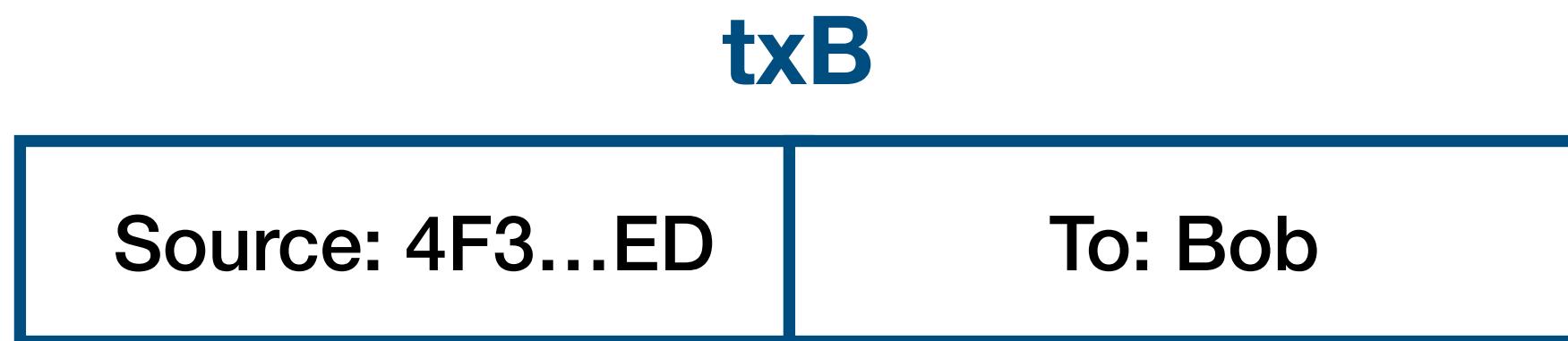
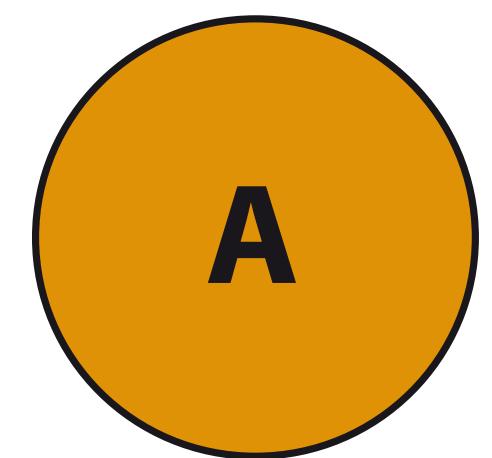




Double-spending transactions (1/2)



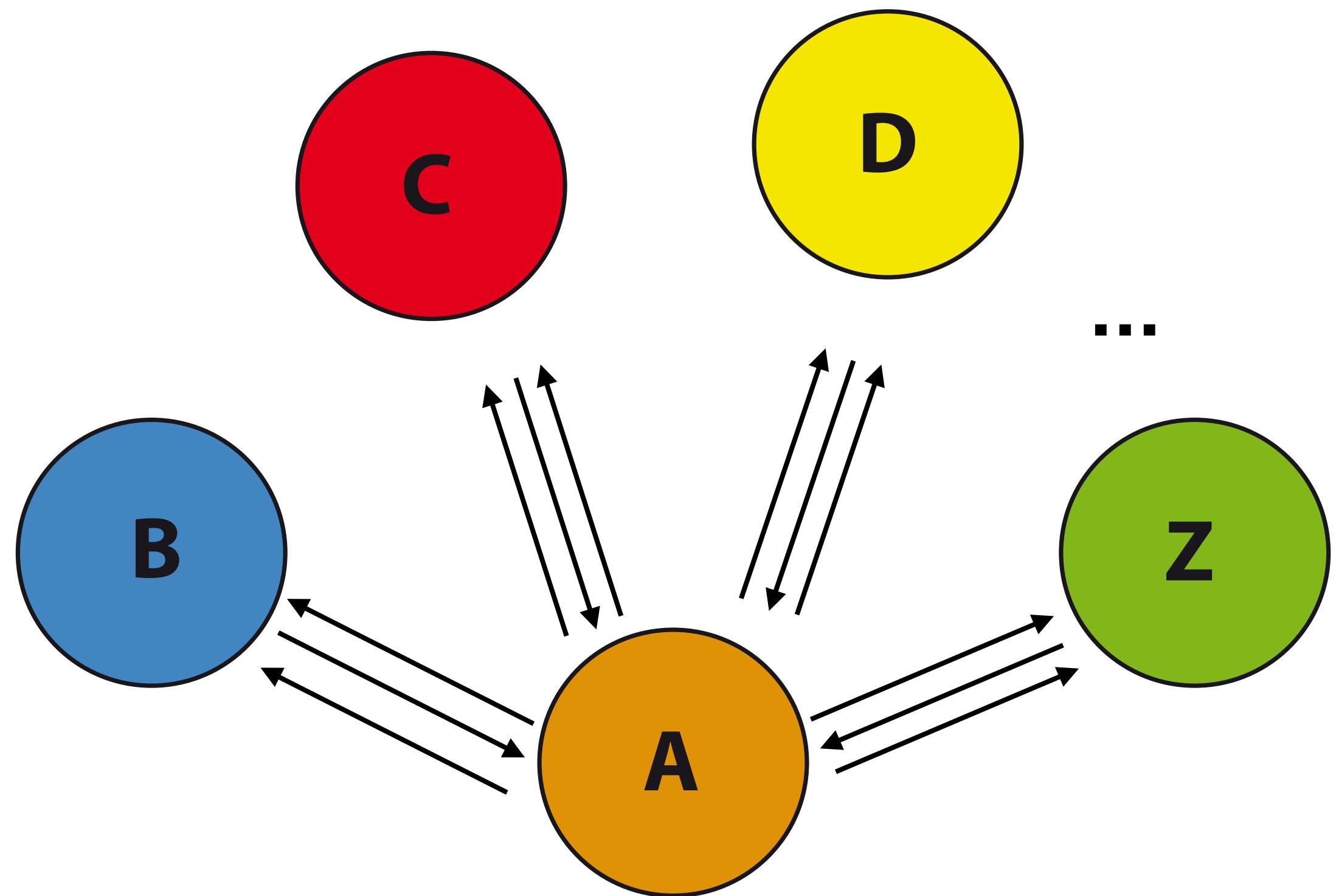
Double-spending transactions (1/2)



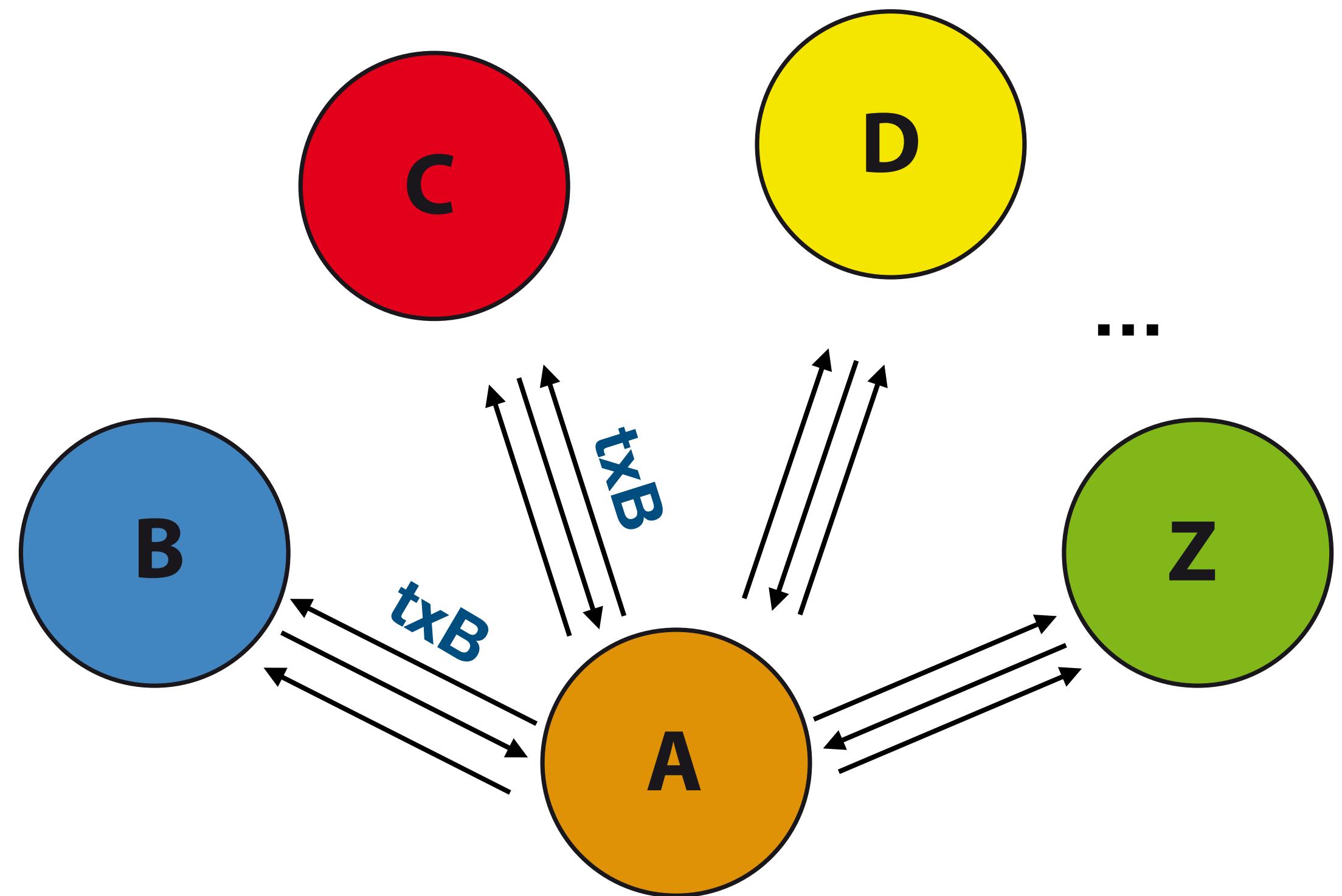
id = 4F3...ED



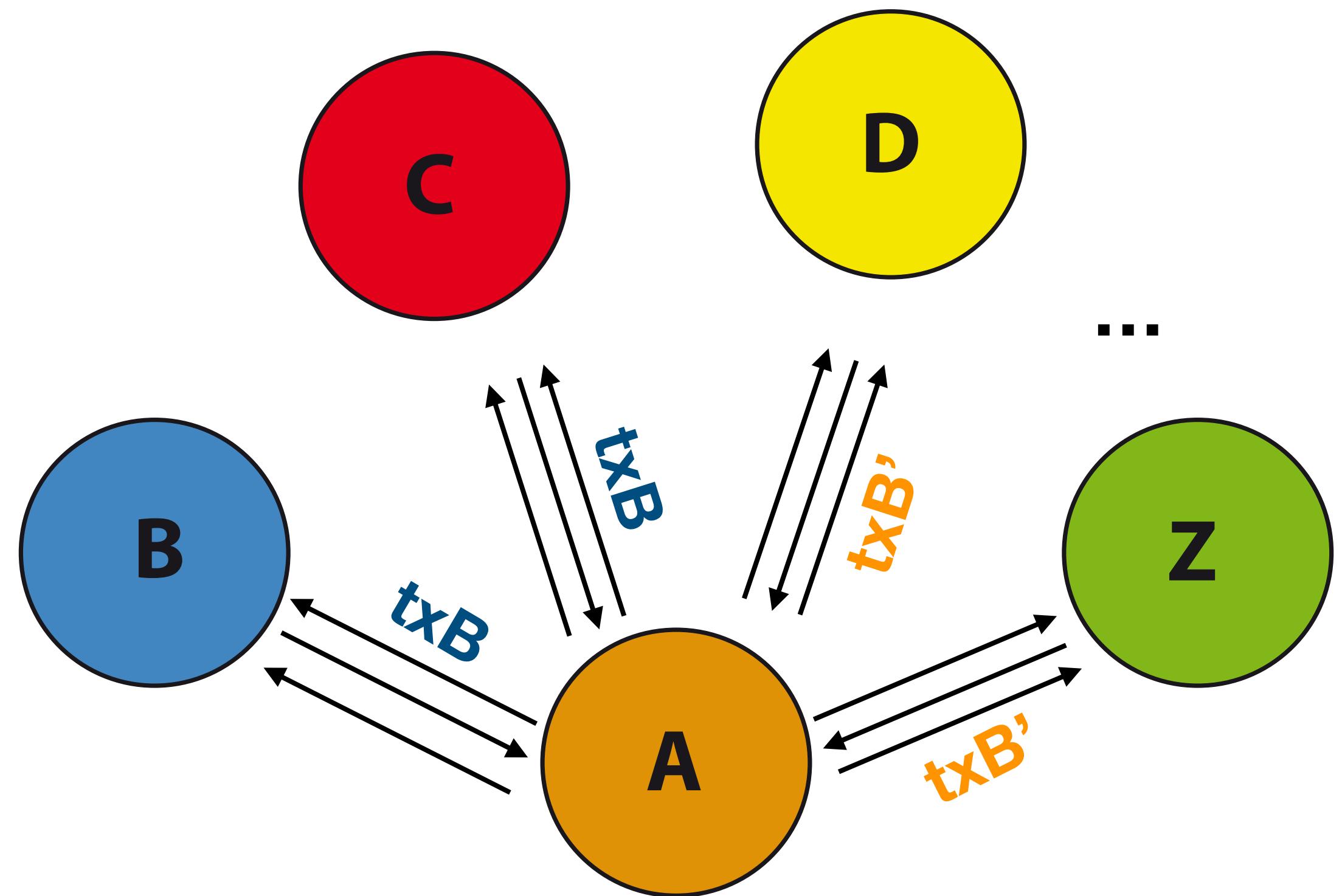
Double-spending transactions (1/2)



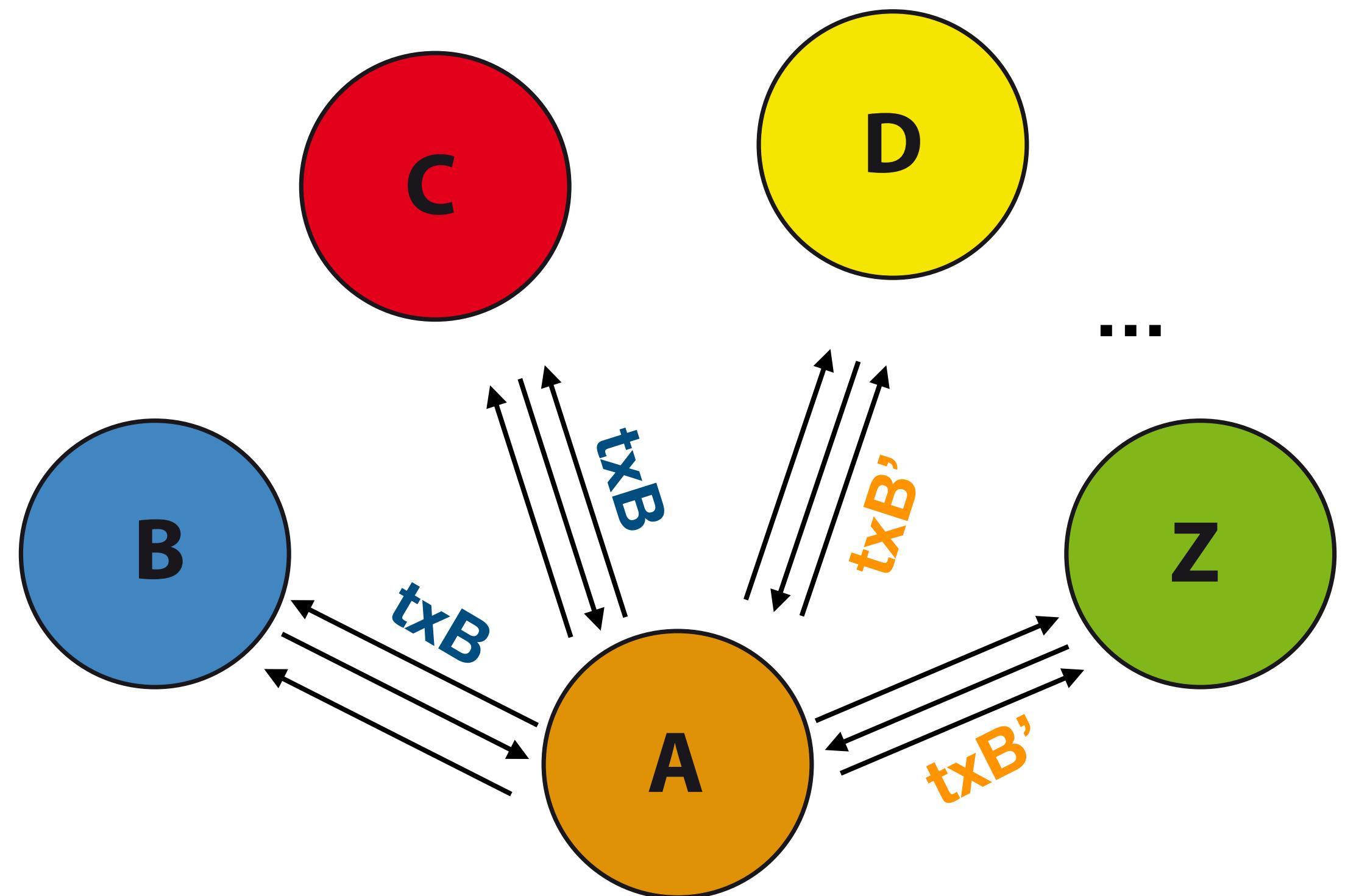
Double-spending transactions (1/2)



Double-spending transactions (1/2)

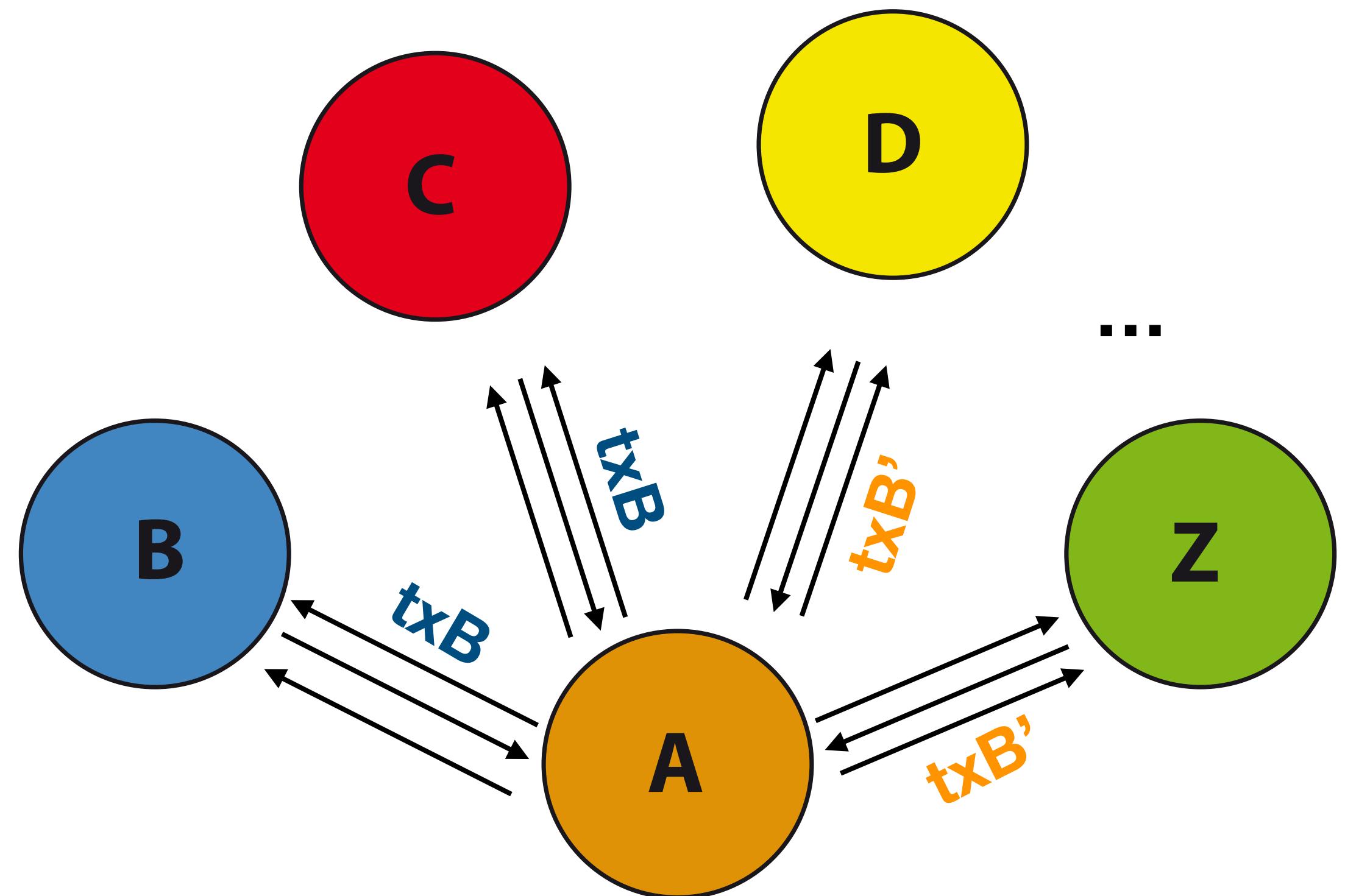


Double-spending transactions (1/2)



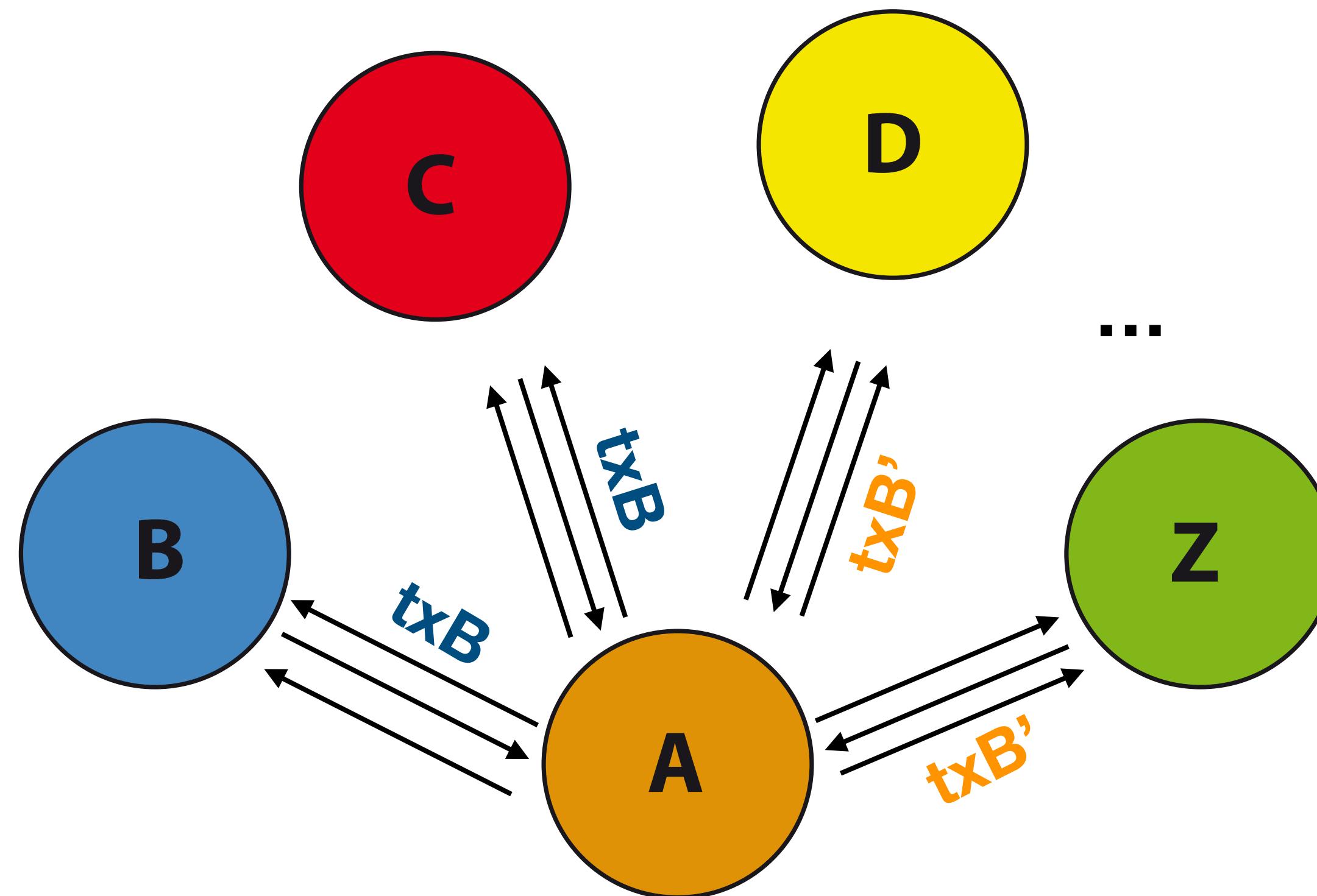
- 0-conf transactions should not be trusted

Double-spending transactions (1/2)



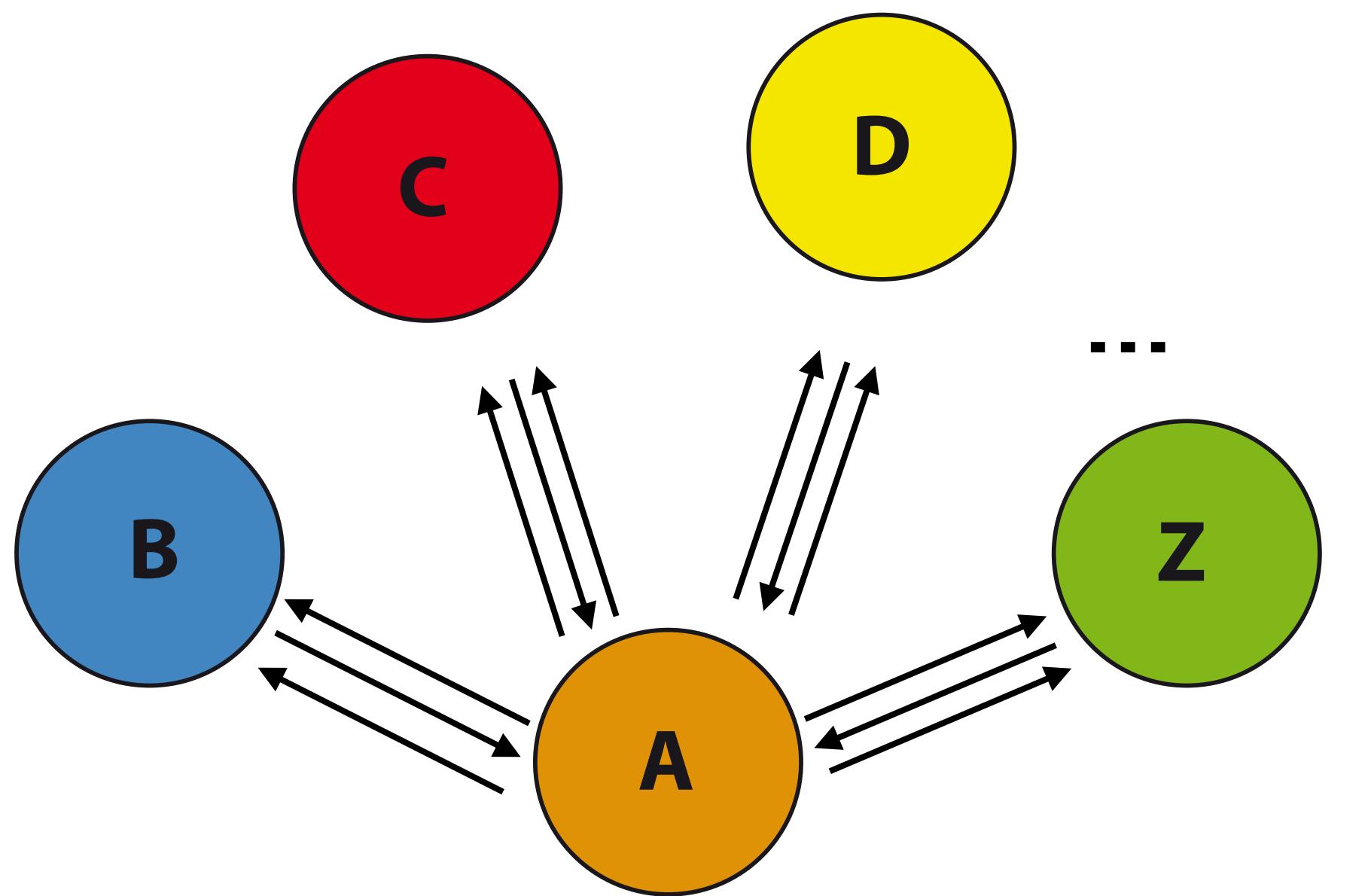
- 0-conf transactions should not be trusted
- If B accepts tx_B before it appears in a block he **can be deceived** by A

Double-spending transactions (1/2)

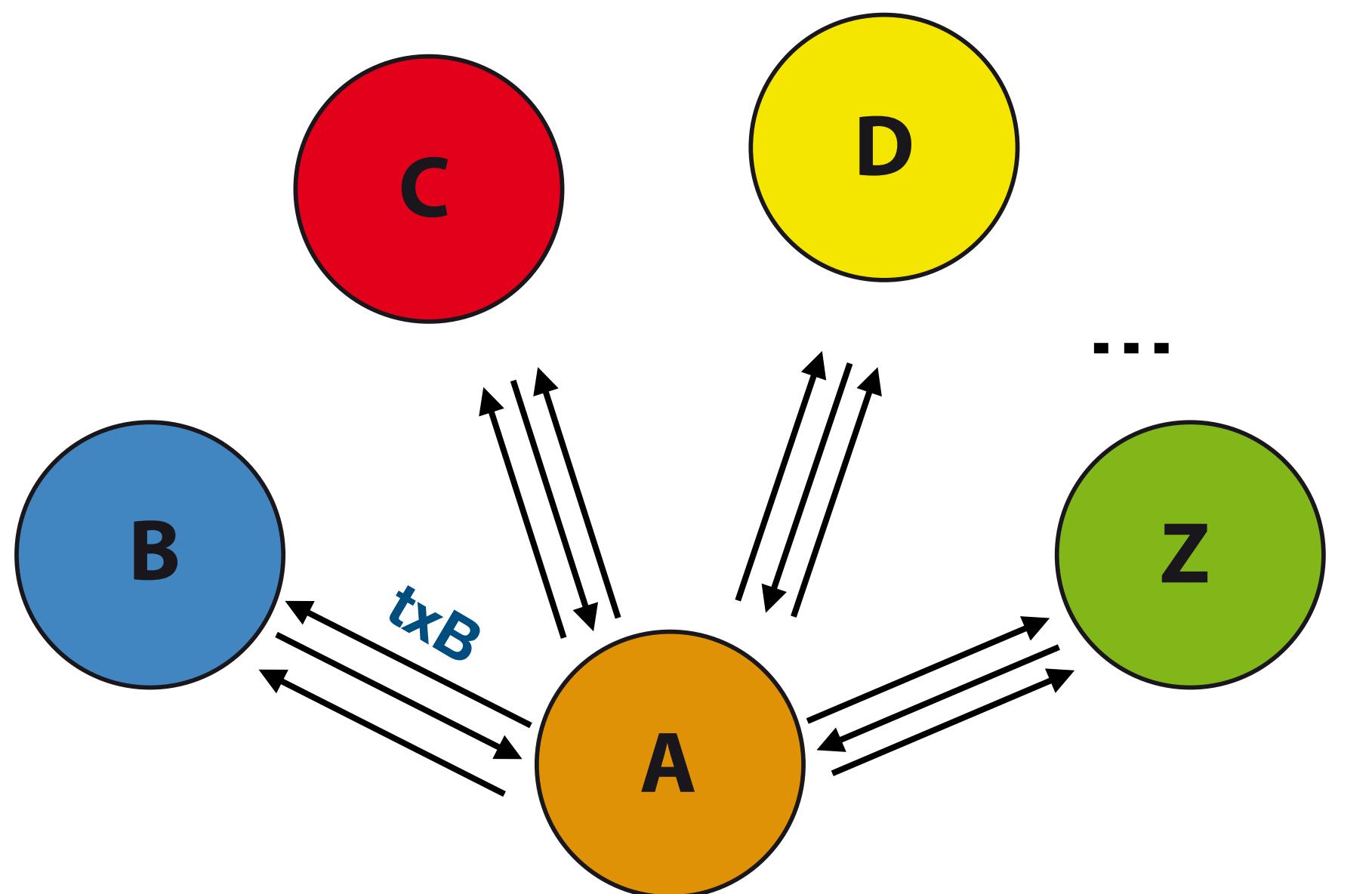


- 0-conf transactions should not be trusted
- If B accepts txB before it appears in a block he **can be deceived** by A
- The de facto confirmation time is **6 blocks** (5 on top of the one including a certain transaction)

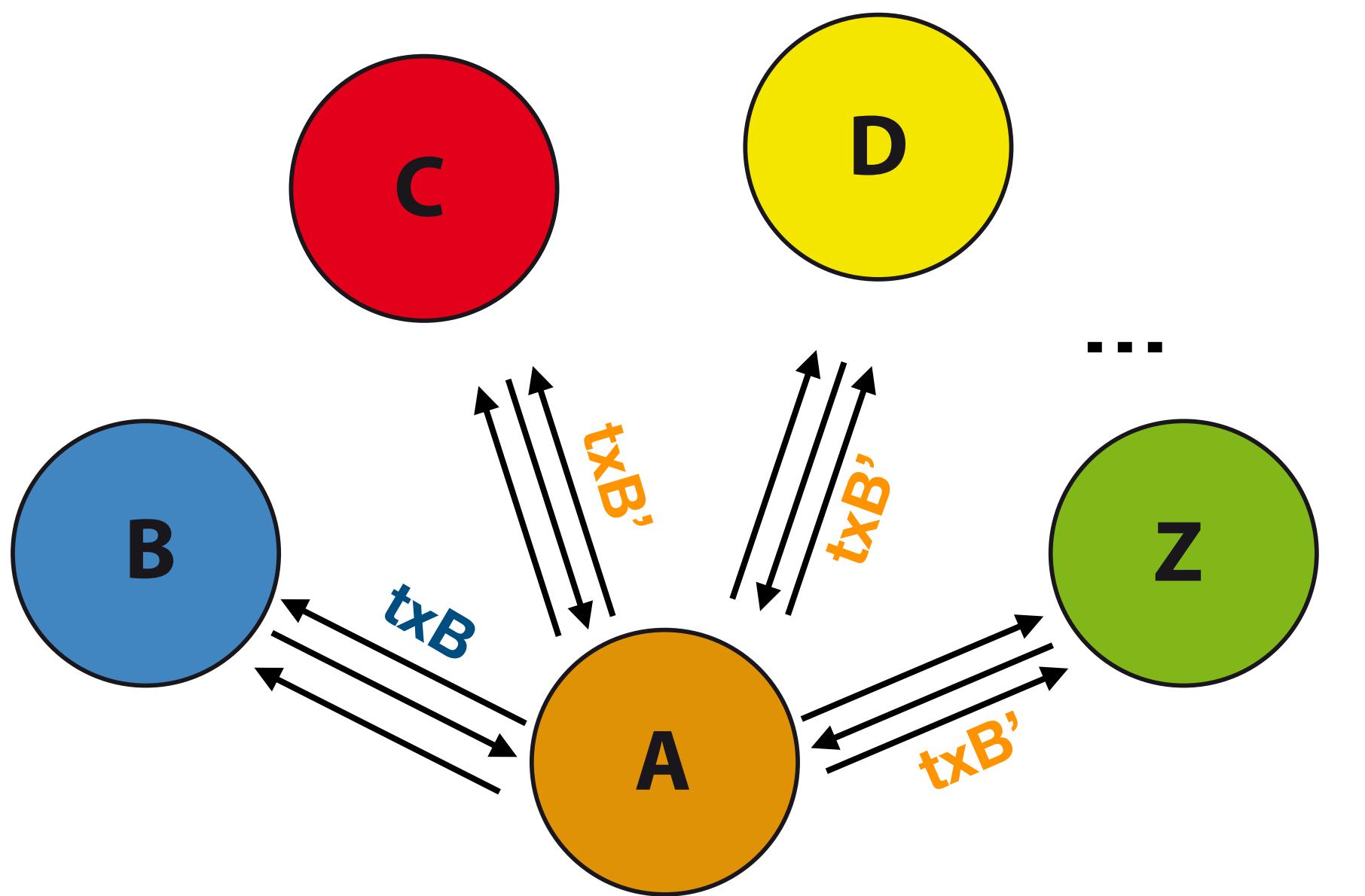
When things go south



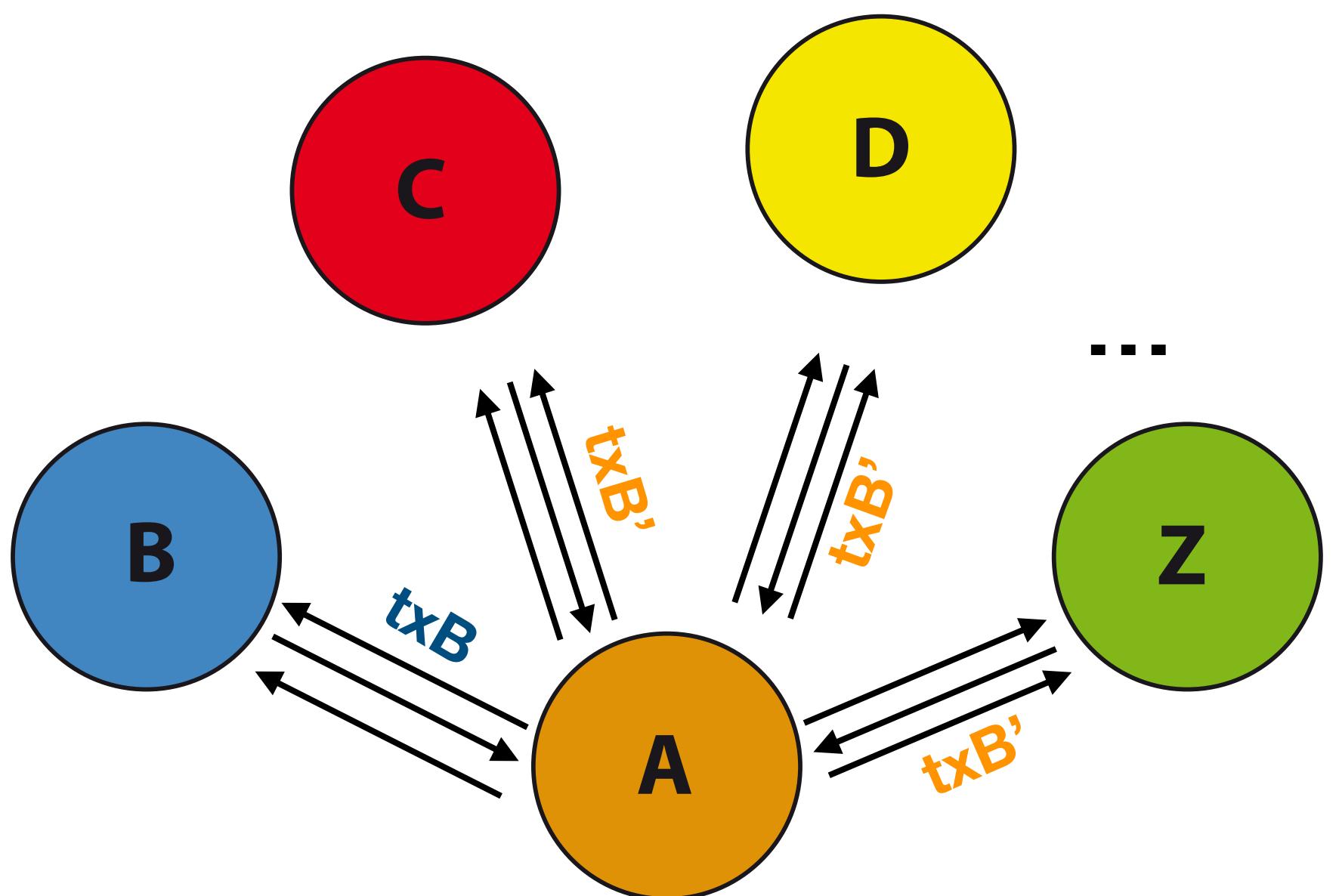
When things go south



When things go south

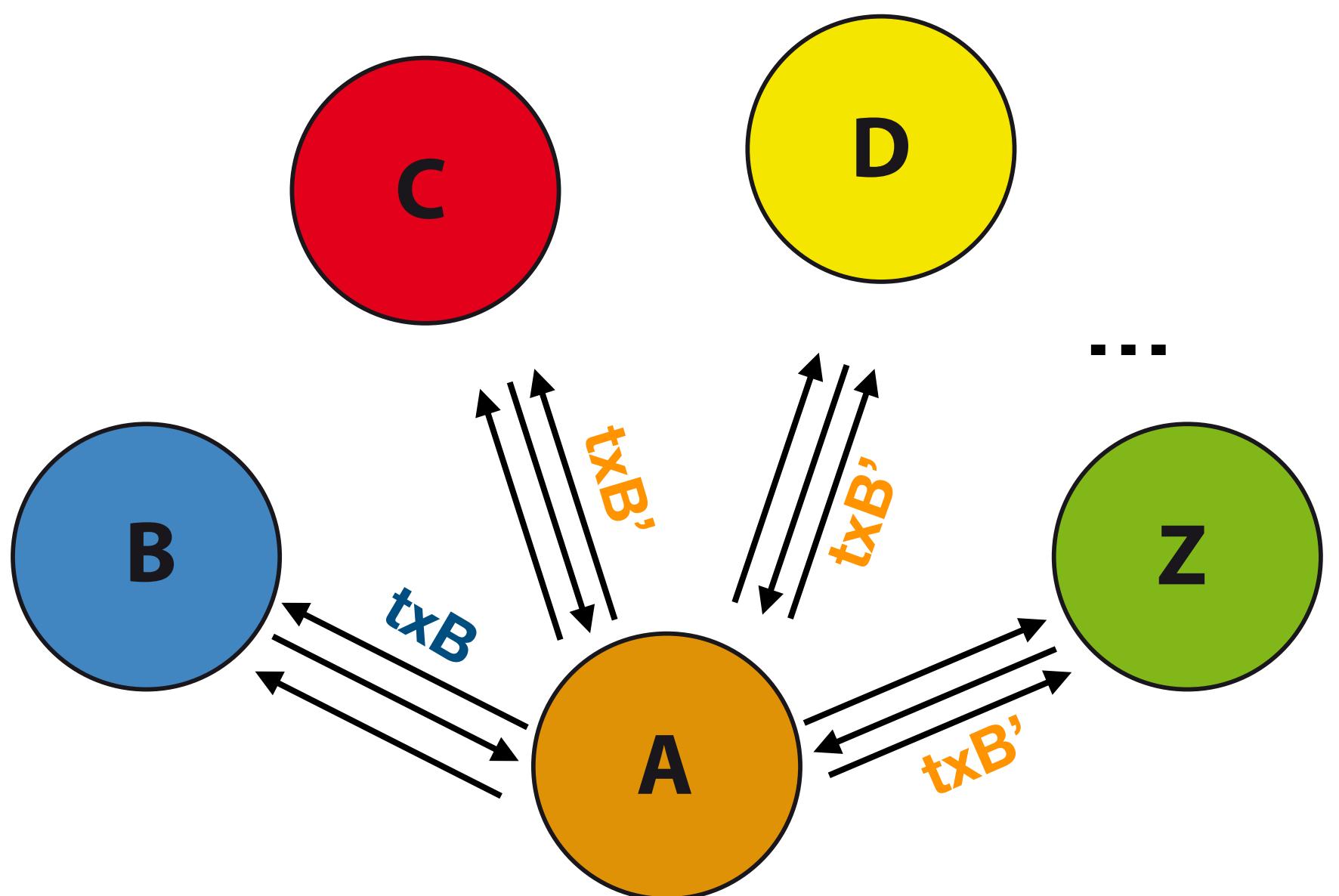


When things go south



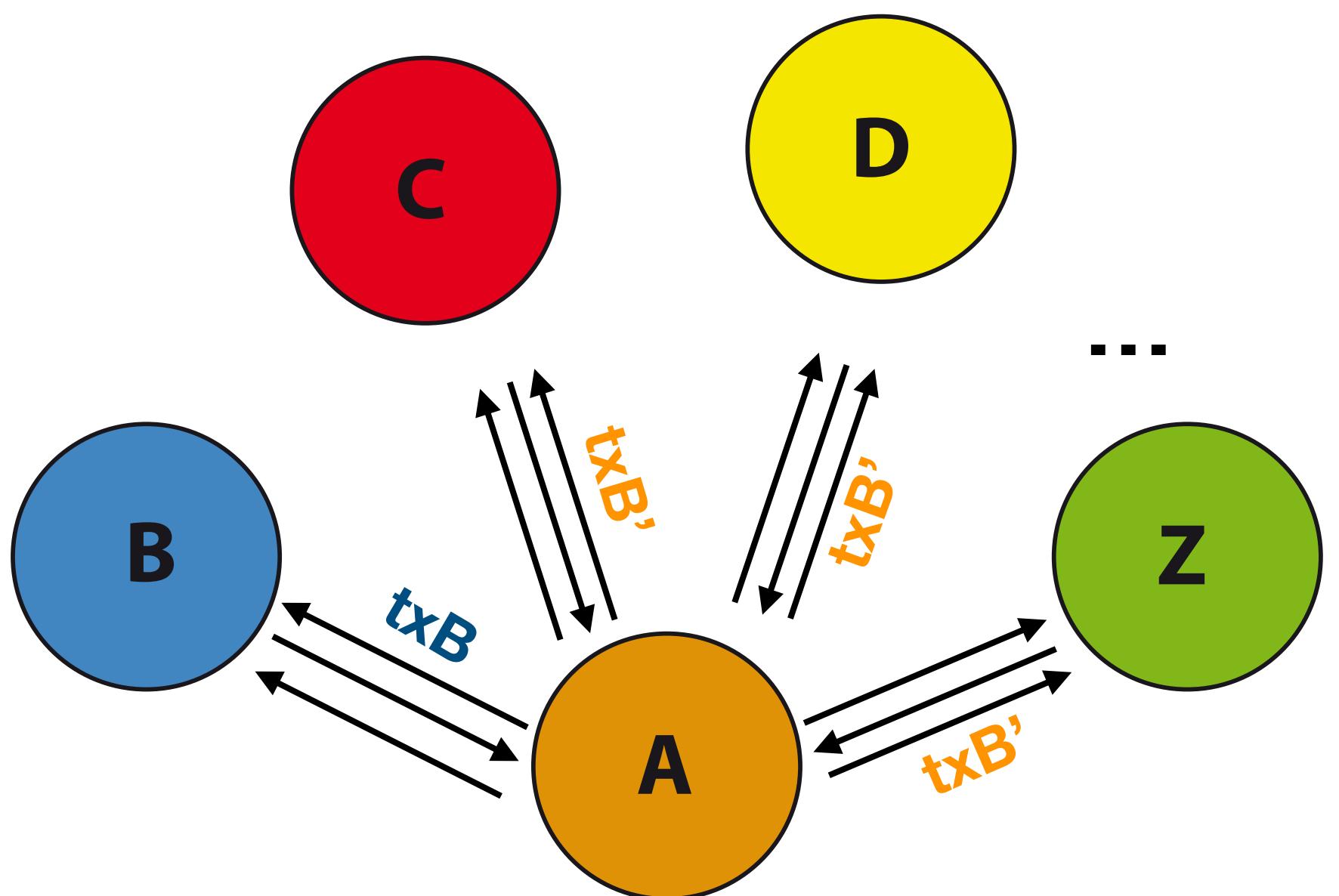
- If A controls the **network view** of B,
A can control what B know about the
currency

When things go south



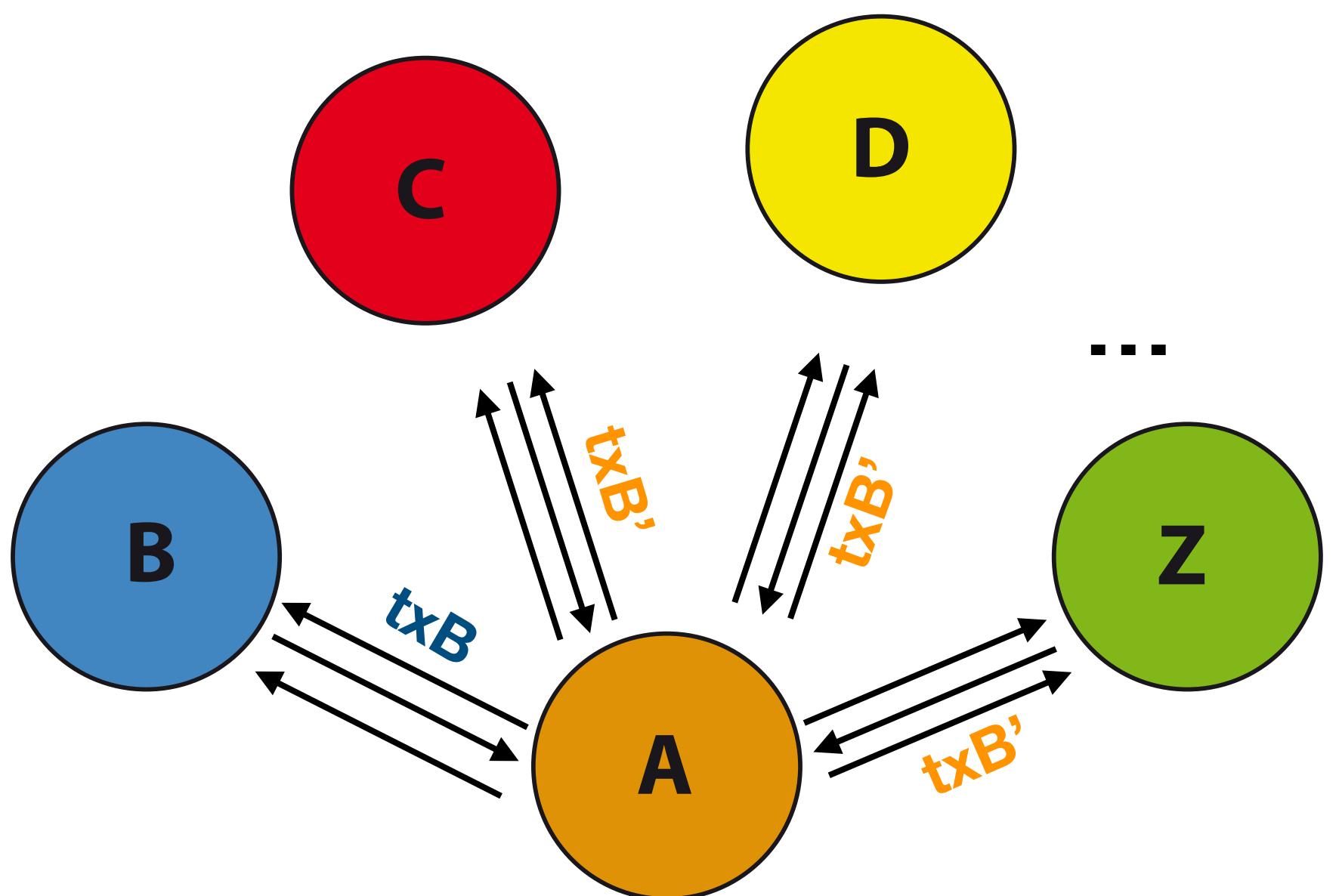
- If A controls the **network view** of B, A can control what B know about the currency
- B is said to be **eclipsed**

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



Network Layer and Information Propagation



Sergi Delgado Segura

Research associate @ UCL InfoSec

