Network Layer and Information Propagation

Sergi Delgado Segura

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:** Bootstraping and file searching

P2P bootstraping

P2P bootstraping

- How do you find peers when you run a new node in the network?

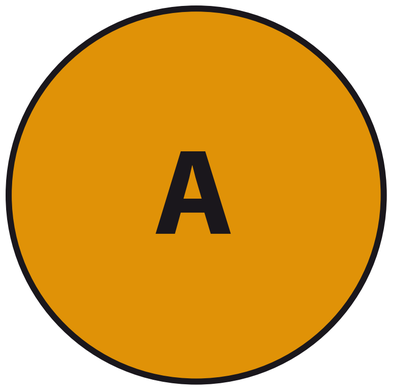
P2P bootstraping

- How do you find peers when you run a new node in the network?

- How peers announce its presence in the network?

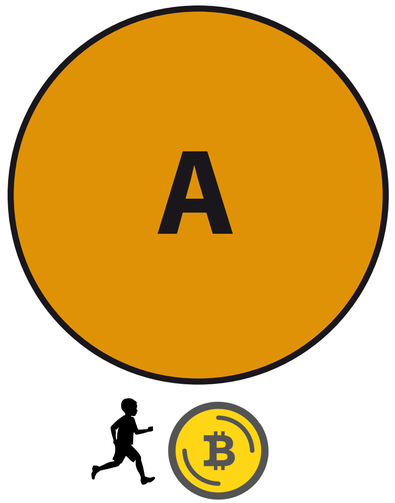
Peer discovery?

Peer discovery?

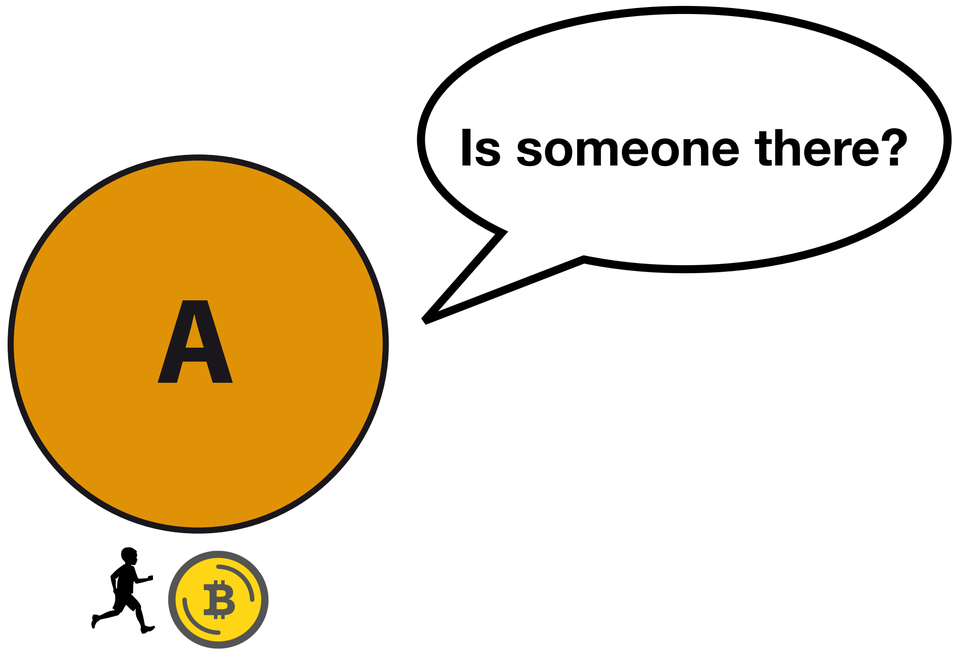


Peer discovery?

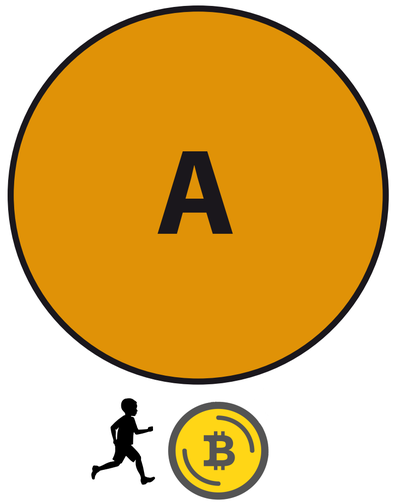
Peer discovery?



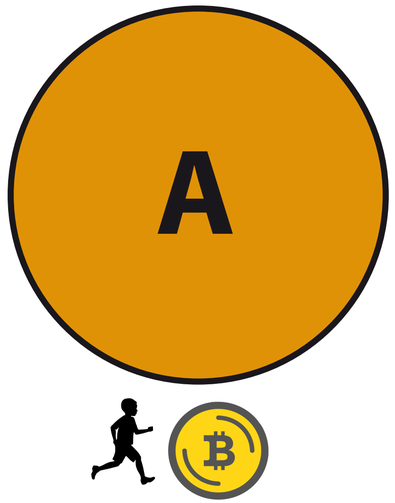
Peer discovery?



Peer discovery?

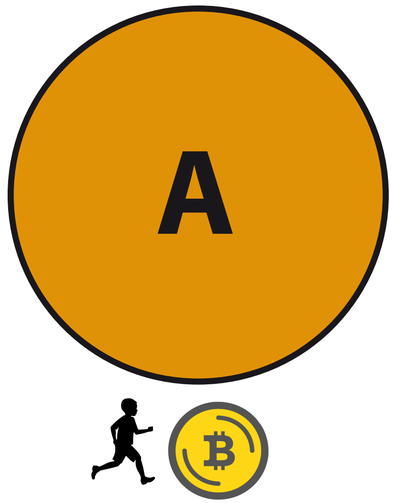


Peer discovery?



tumbleweed

Peer discovery?

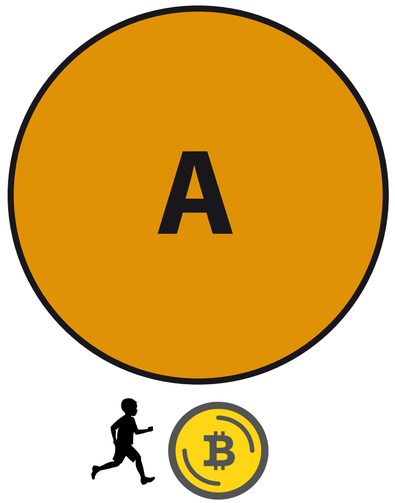


tumbleweed

Peer discovery?

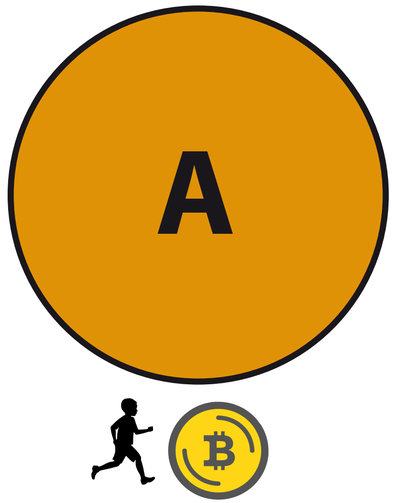
tumbleweed

Peer discovery?



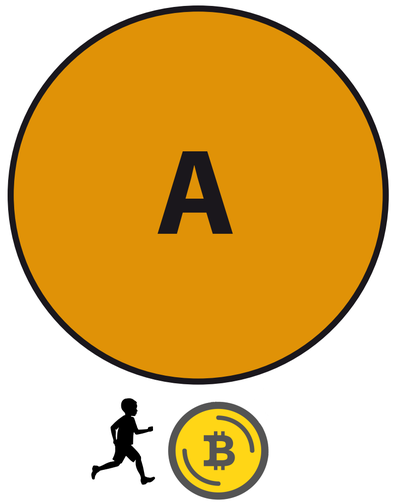
tumbleweed

Peer discovery?

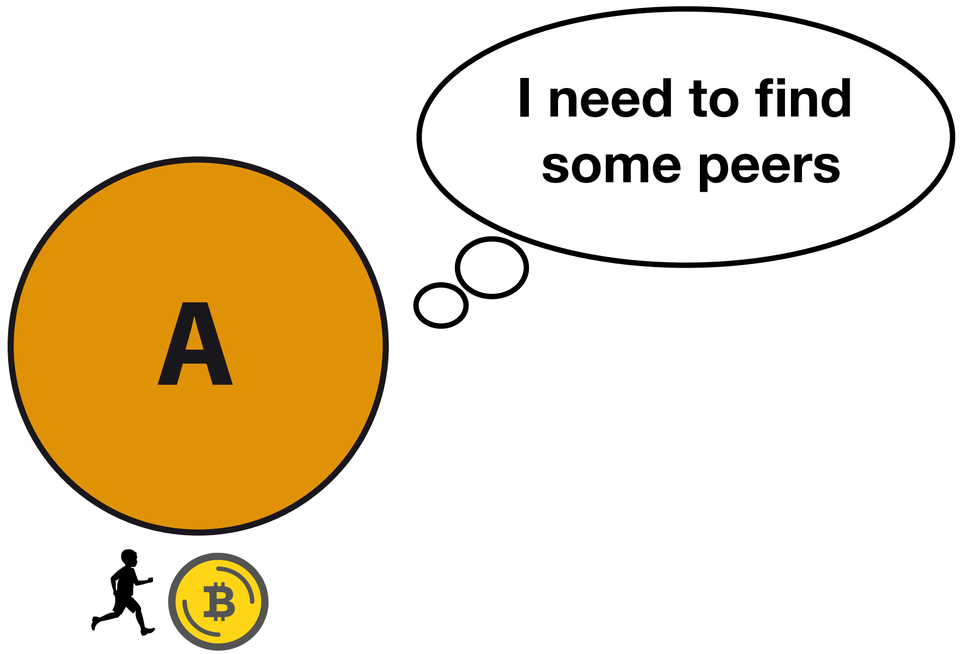


tumbleweed

Peer discovery?



Peer discovery?



P2P bootstraping

- How do you find peers when you run a new node in the network?

- How peers announce its presence in the network?

P2P bootstraping

- How do you find peers when you run a new node in the network?

- How peers announce its presence in the network?

- **Hardcoded trusted addresses / IRC bootstraping / Trusted DNS seeds / etc**

P2P file sharing

P2P file sharing

- How can you find files in a network when each peer can have different files?

P2P file sharing

- How can you find files in a network when each peer can have different files?

- How files are announced?

P2P file sharing

- How can you find files in a network when each peer can have different files?

- How files are announced?

- **Announce / Request**

P2P file sharing

- How can you find files in a network when each peer can have different files?

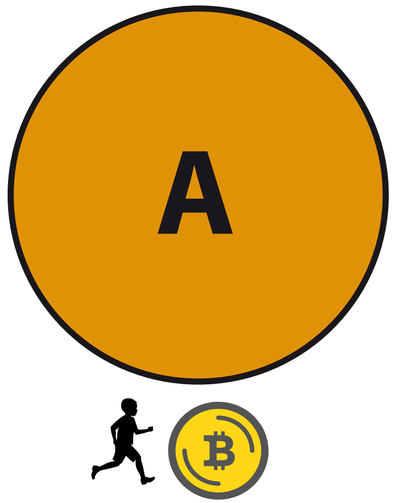
- How files are announced?

- **Announce / Request**

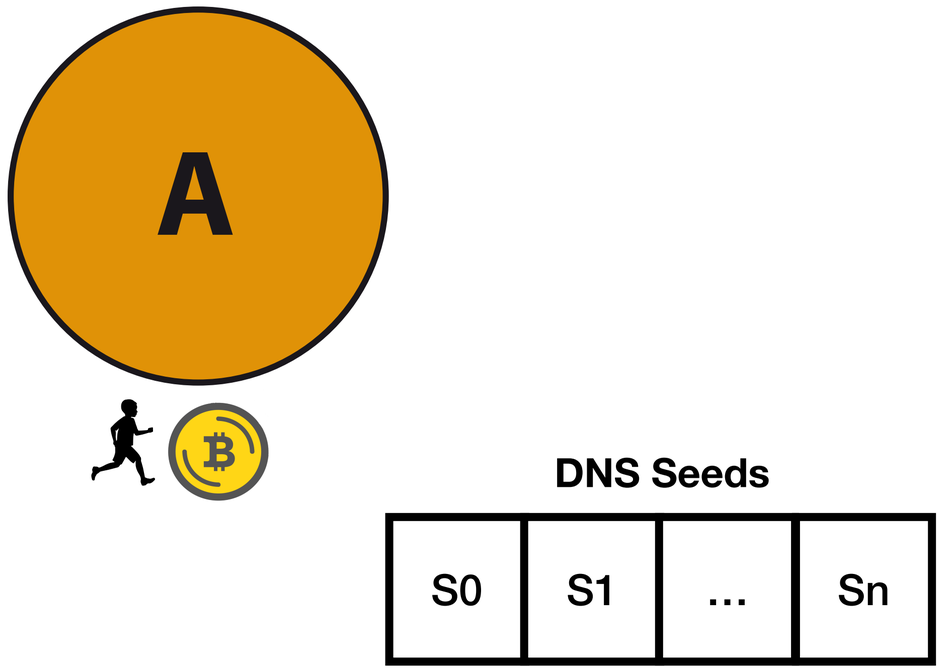
Cryptocurrency networks follow and announce paradigm instead of a request one

Node bootstraping and peer discovery

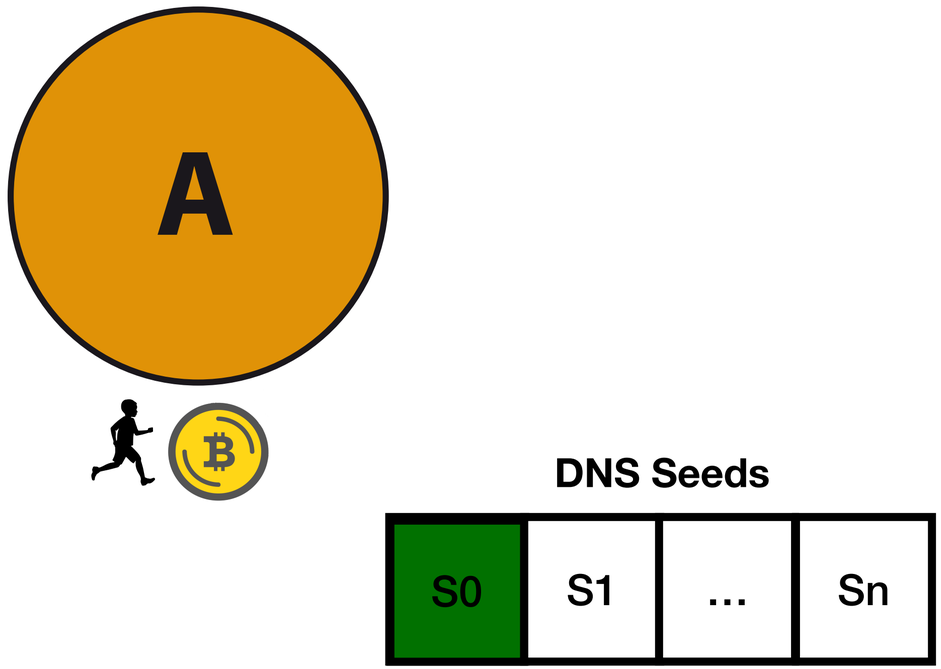
Bitcoin P2P bootstraping (peer discovery)



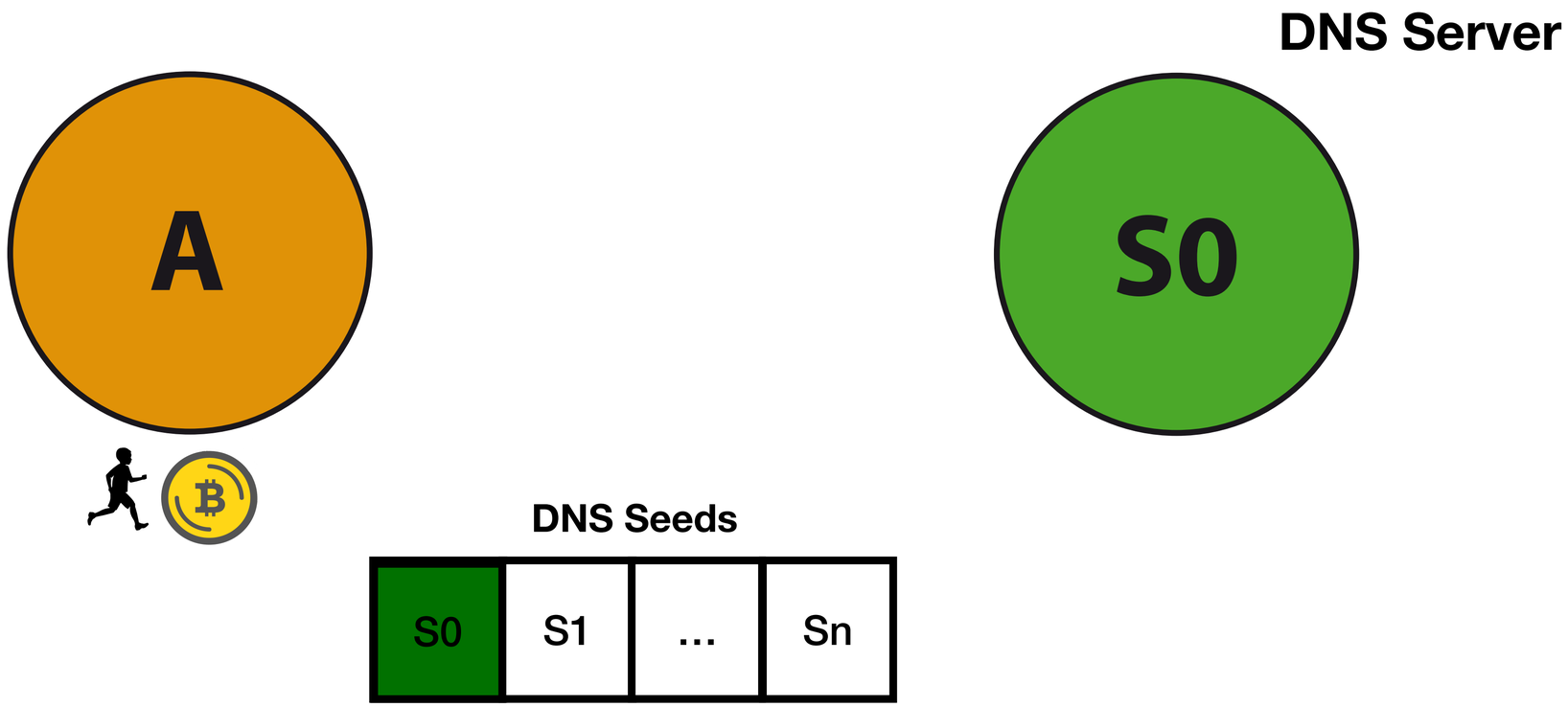
Bitcoin P2P bootstraping (peer discovery)



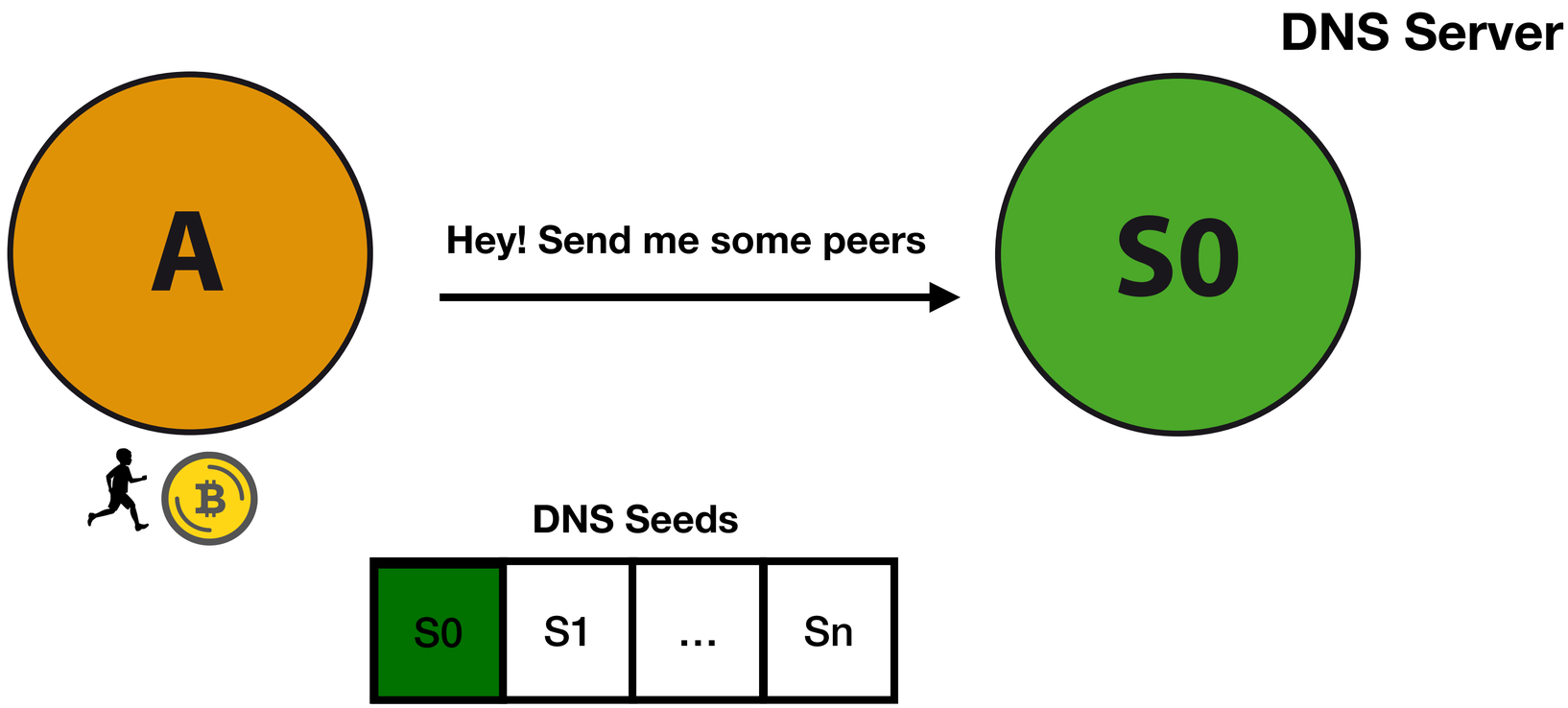
Bitcoin P2P bootstraping (peer discovery)



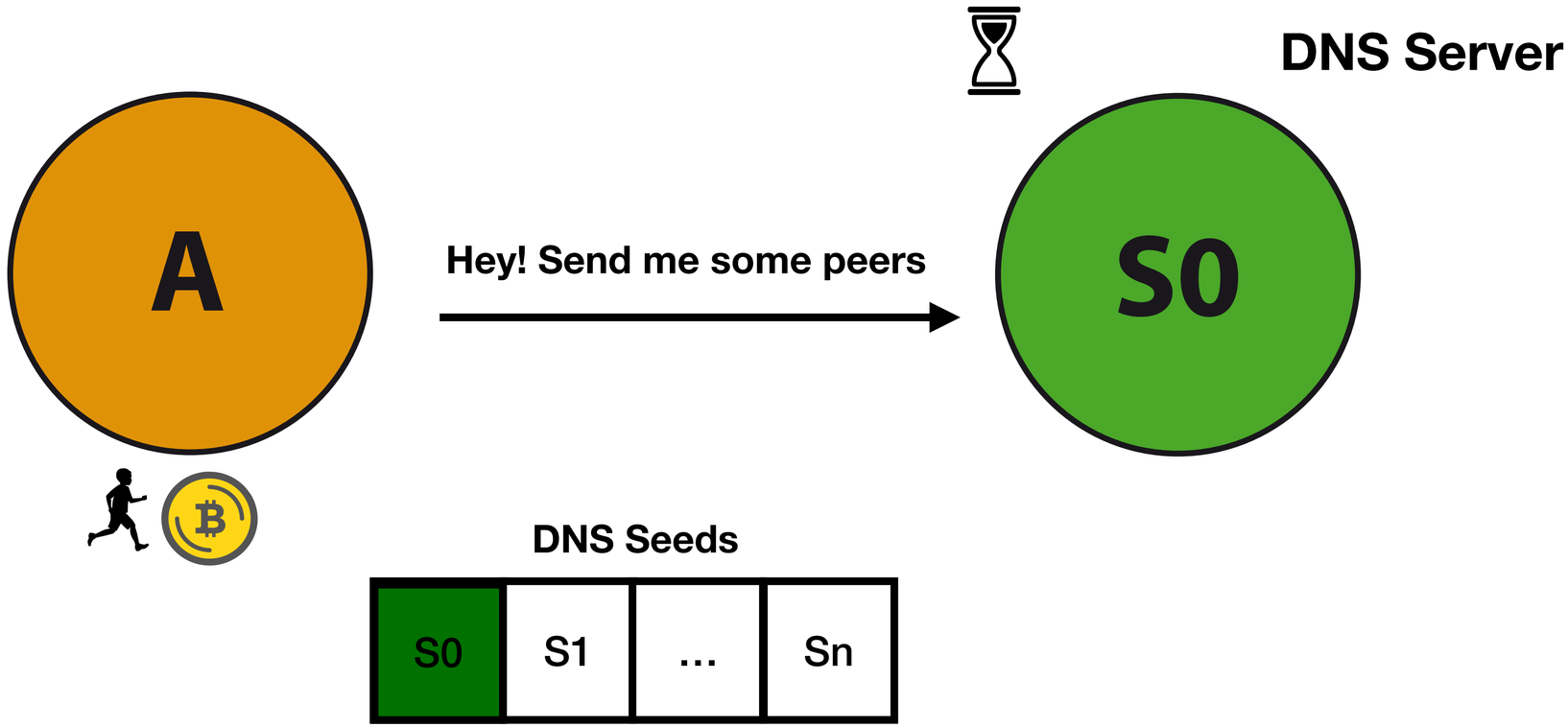
Bitcoin P2P bootstraping (peer discovery)



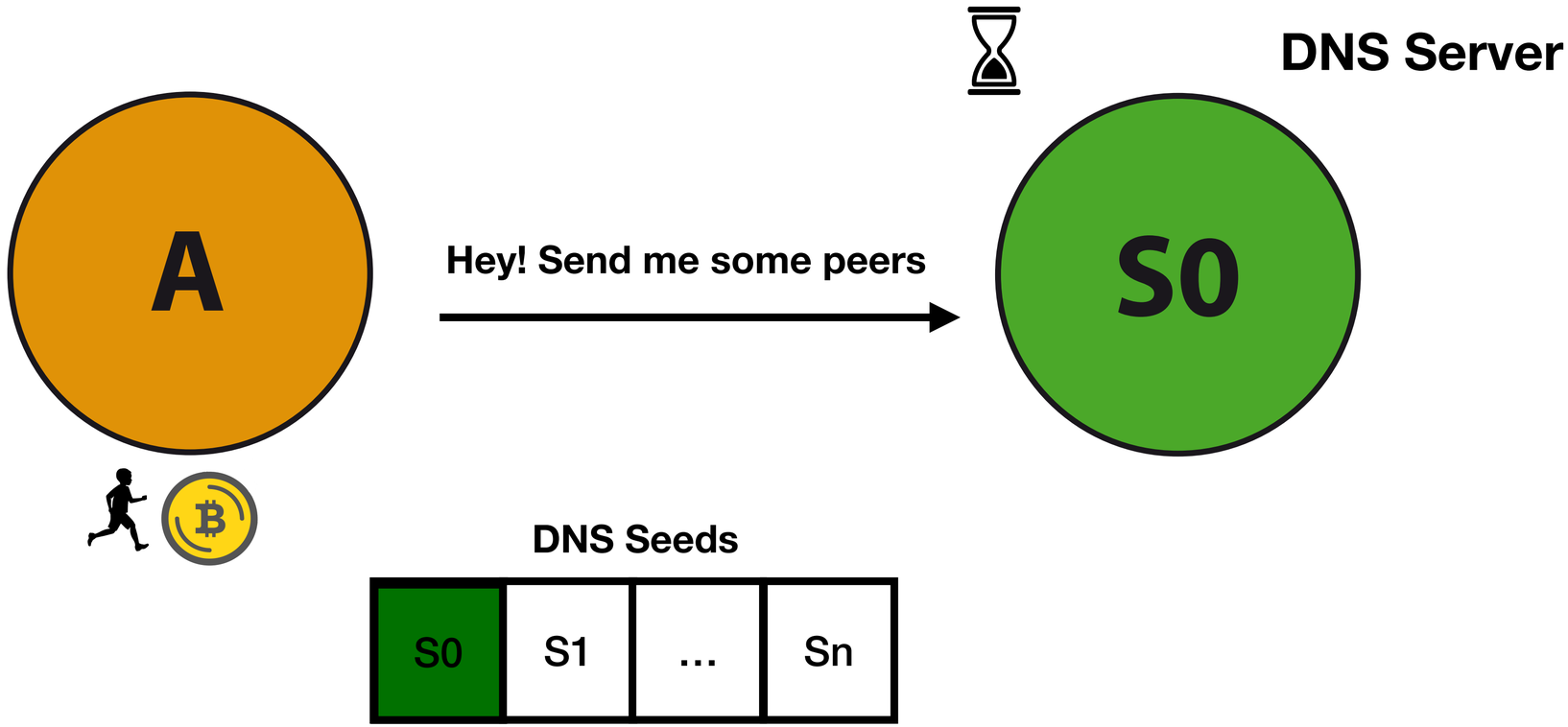
Bitcoin P2P bootstraping (peer discovery)



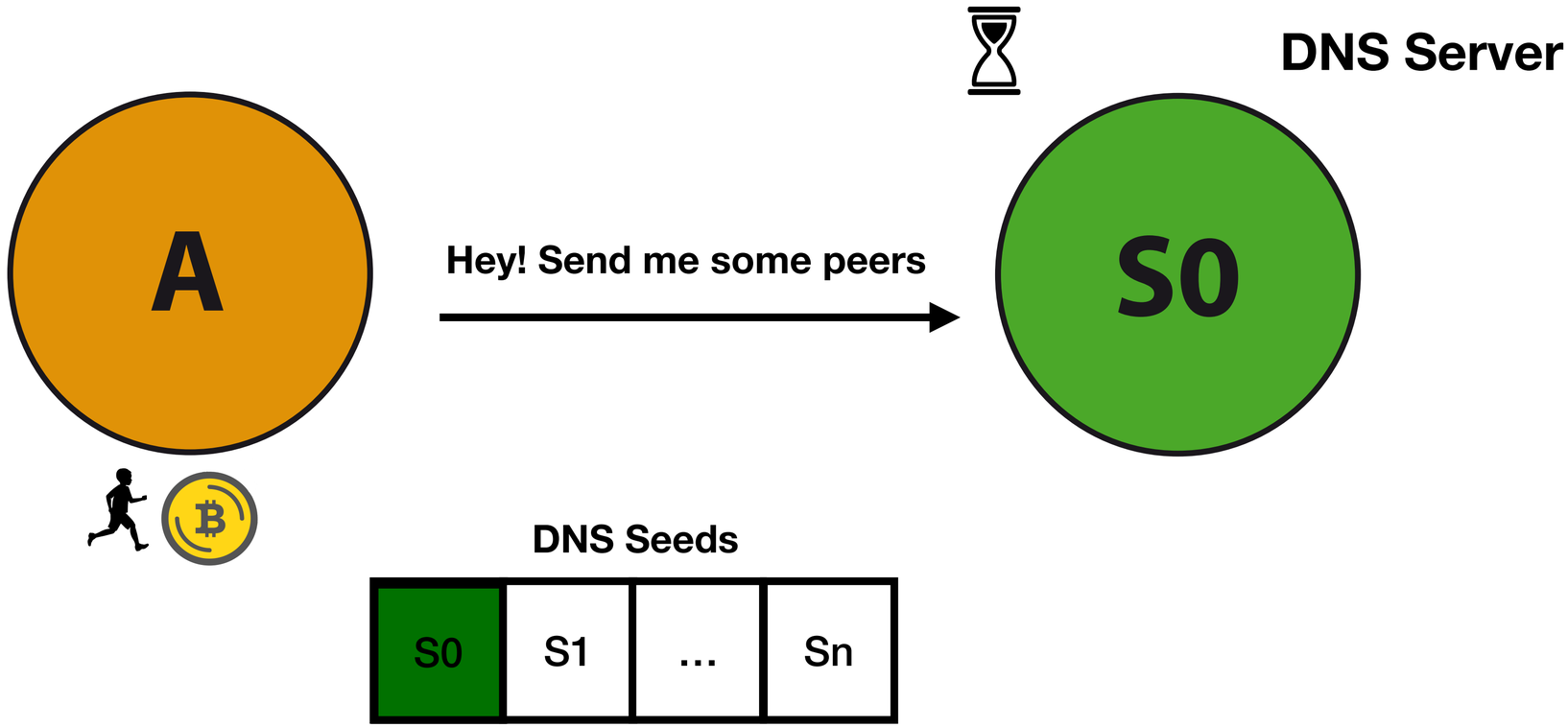
Bitcoin P2P bootstraping (peer discovery)



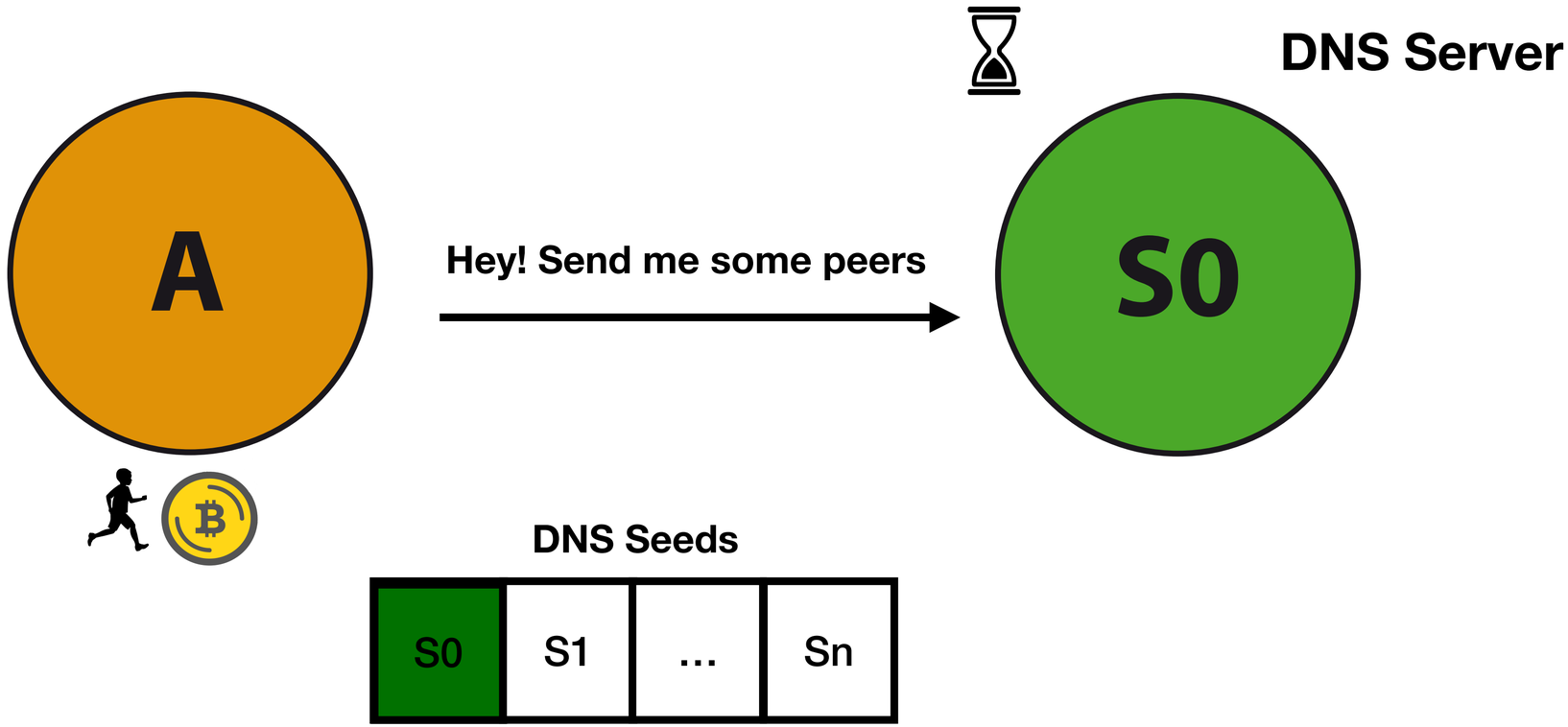
Bitcoin P2P bootstraping (peer discovery)



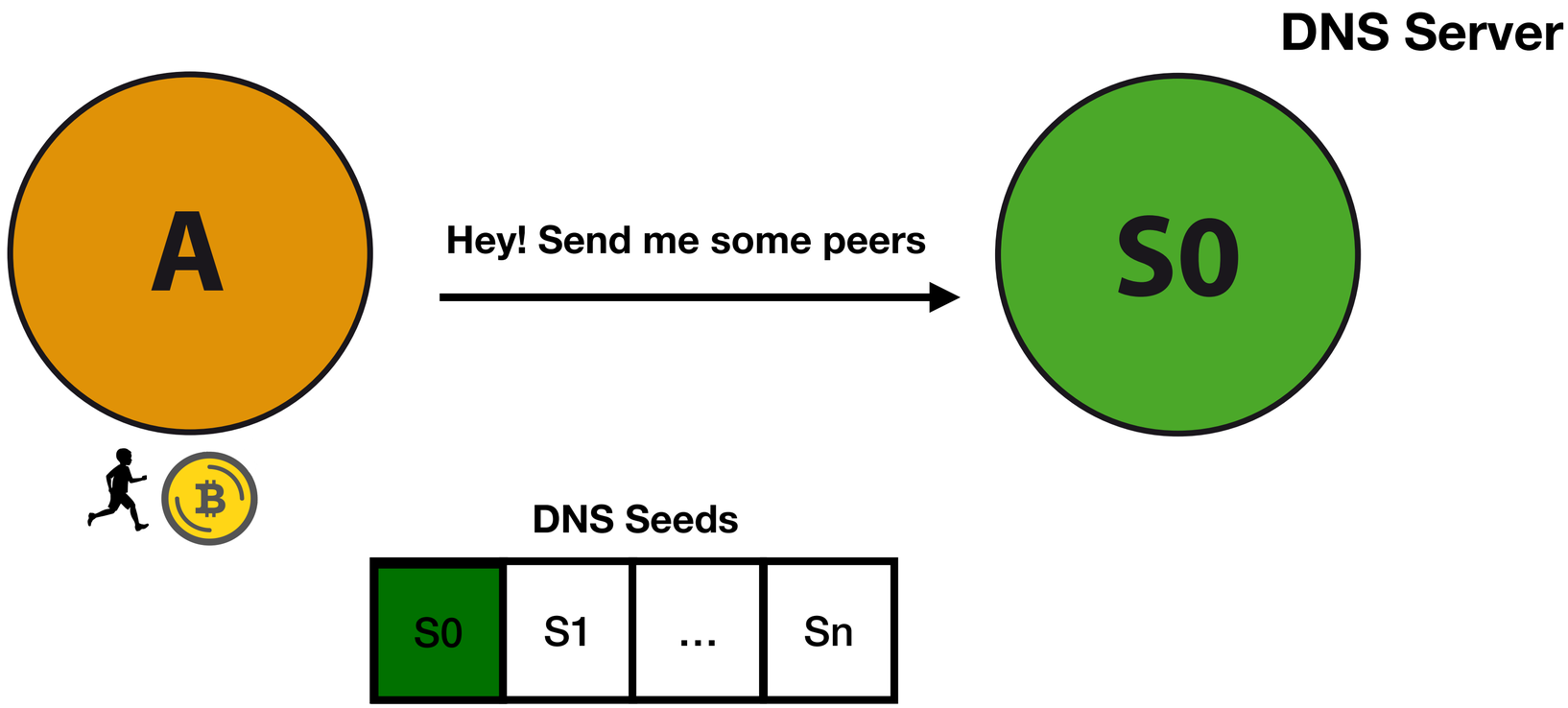
Bitcoin P2P bootstraping (peer discovery)



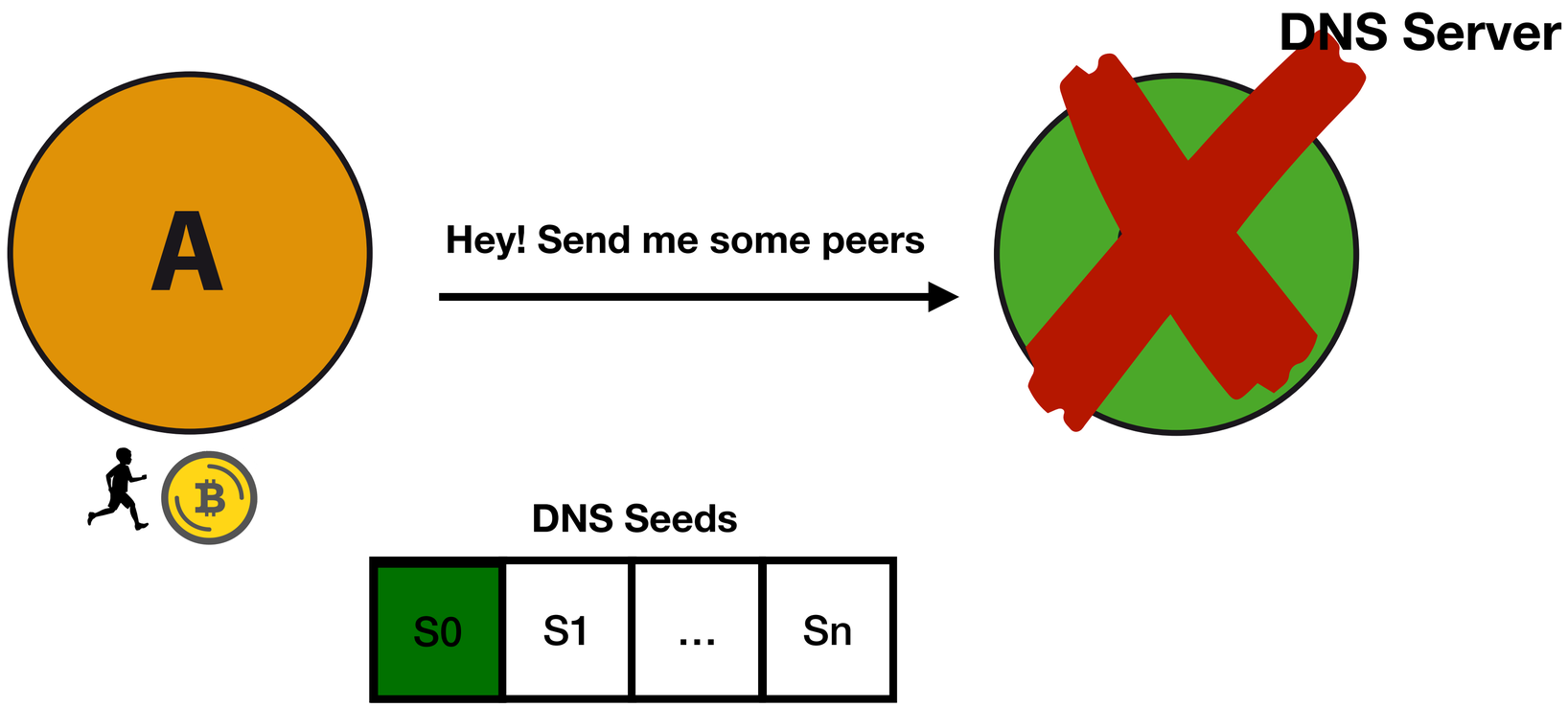
Bitcoin P2P bootstraping (peer discovery)



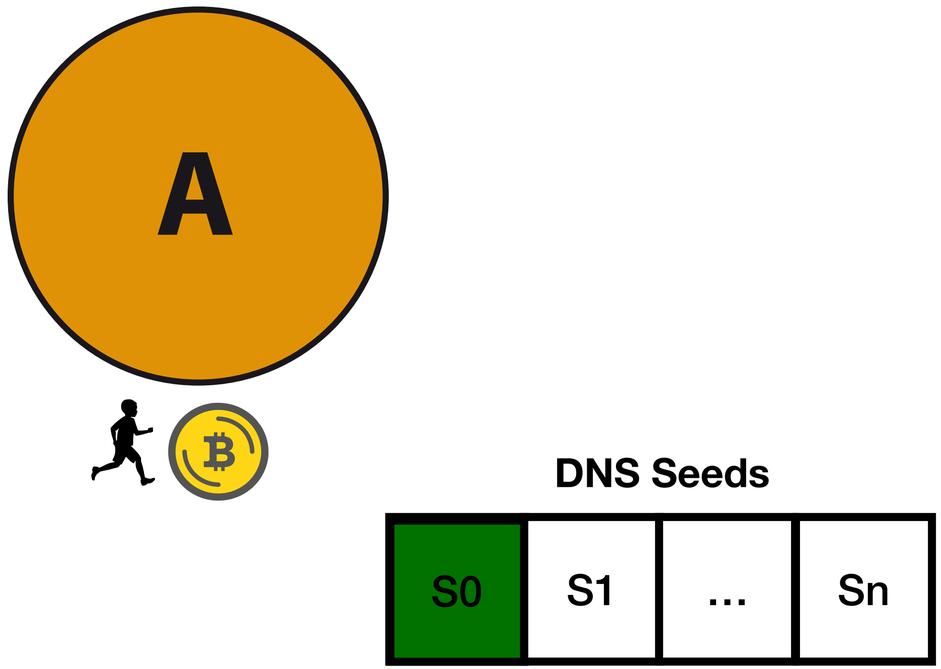
Bitcoin P2P bootstraping (peer discovery)



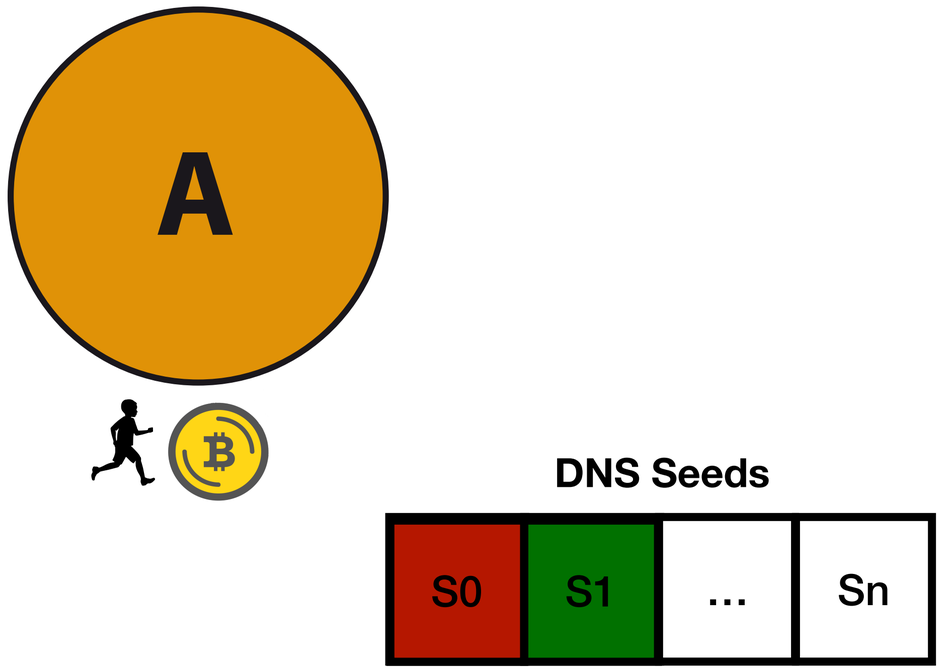
Bitcoin P2P bootstraping (peer discovery)



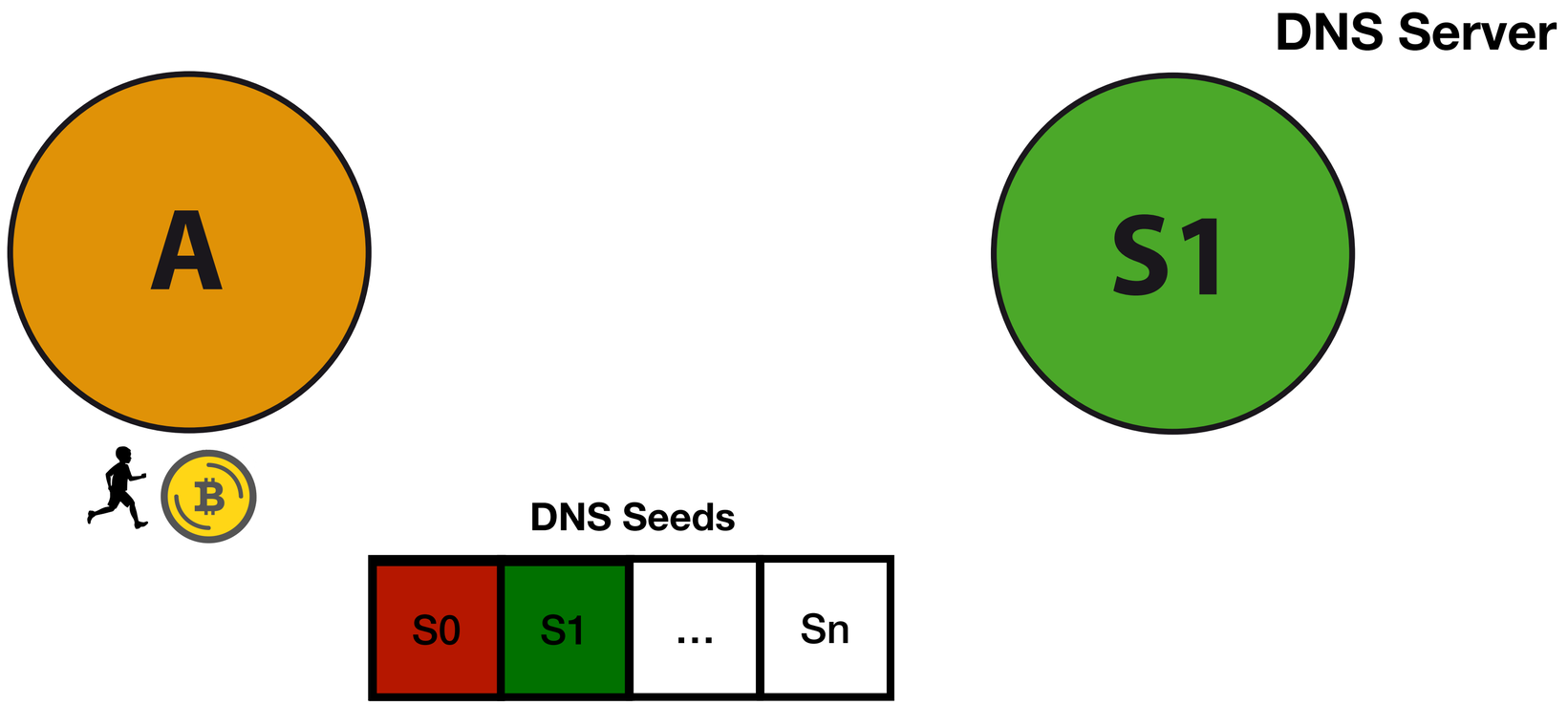
Bitcoin P2P bootstraping (peer discovery)



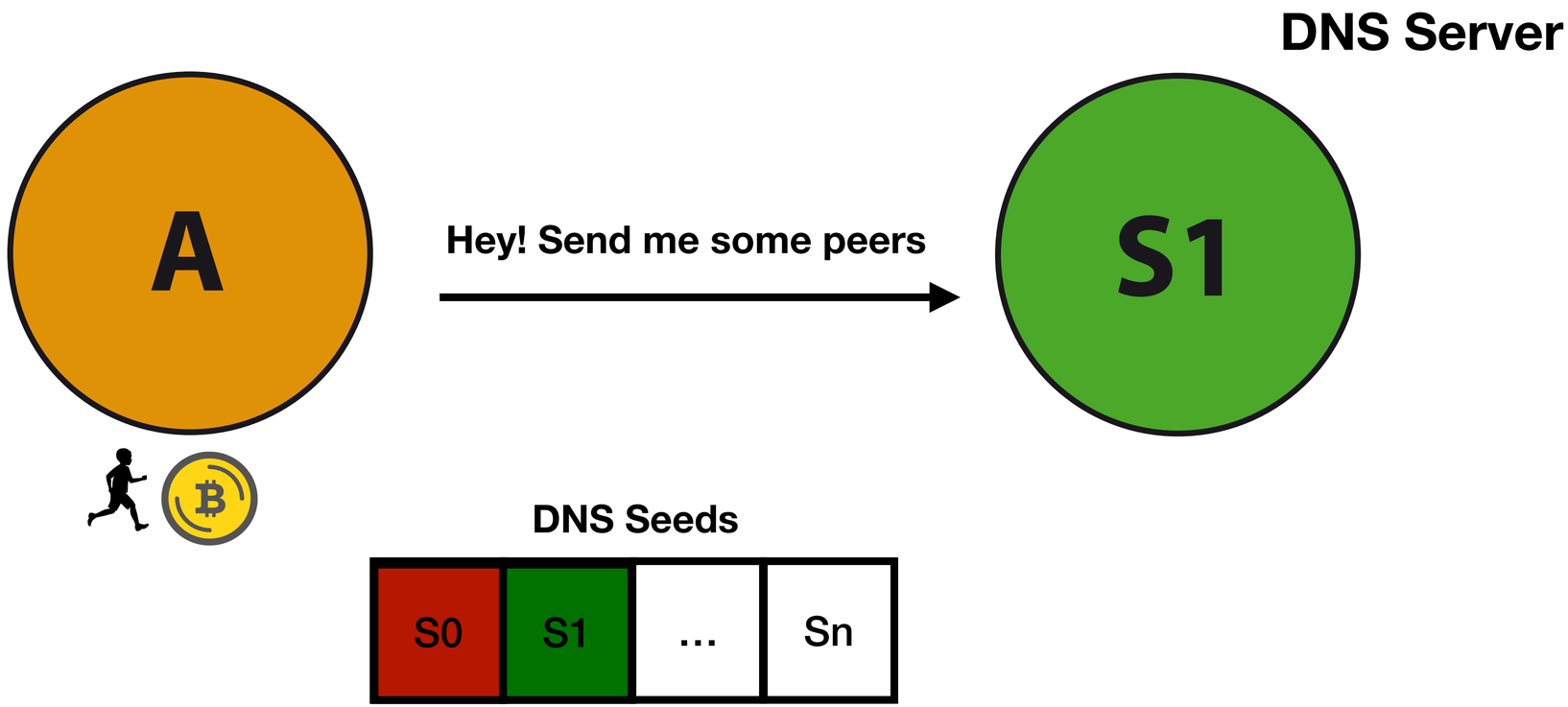
Bitcoin P2P bootstraping (peer discovery)



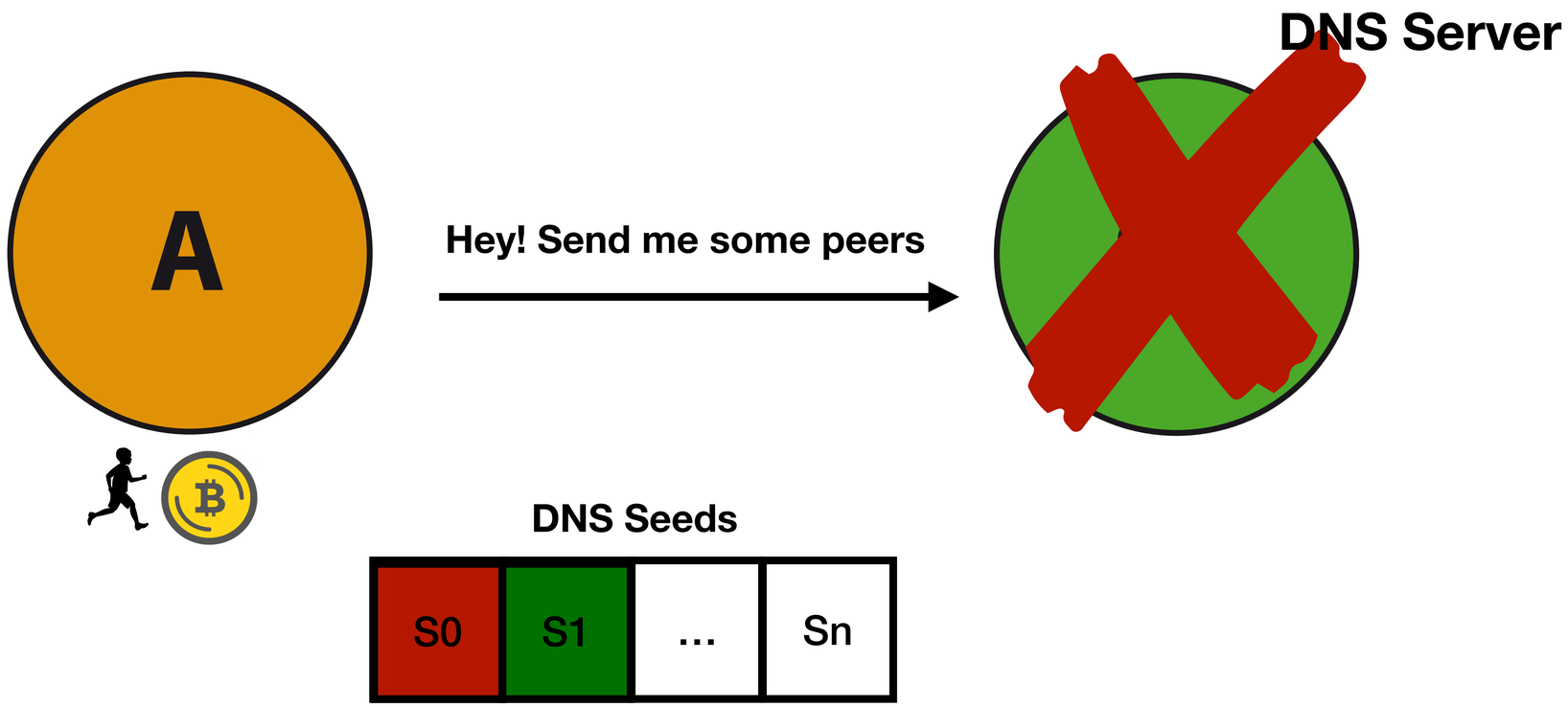
Bitcoin P2P bootstraping (peer discovery)



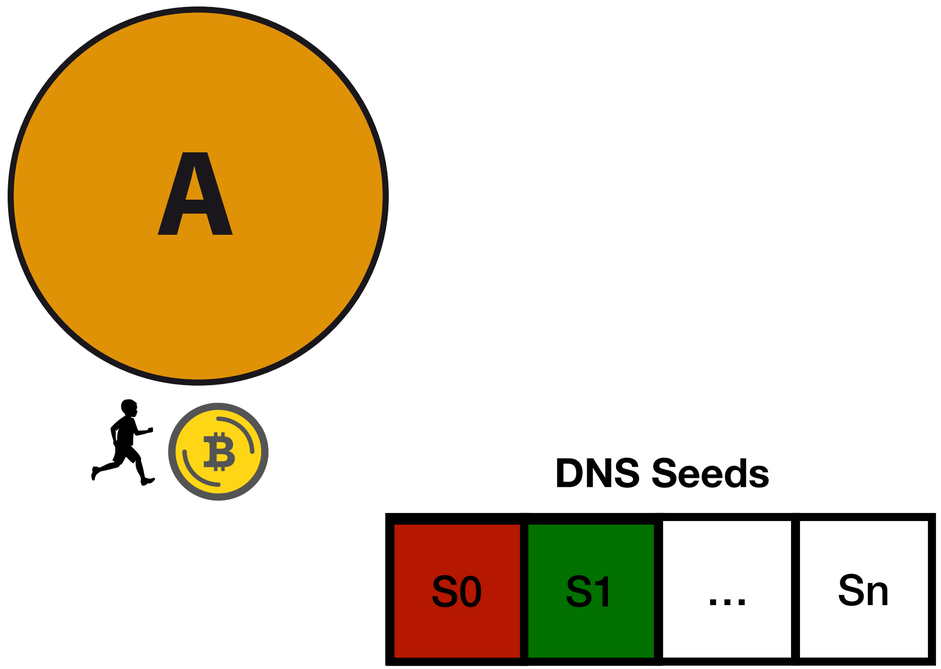
Bitcoin P2P bootstraping (peer discovery)



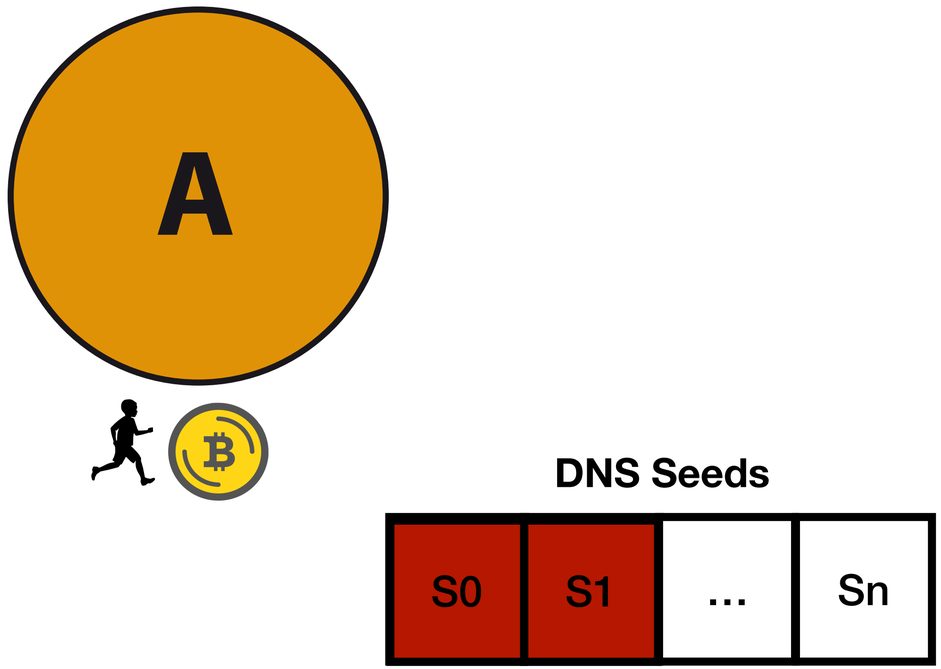
Bitcoin P2P bootstraping (peer discovery)



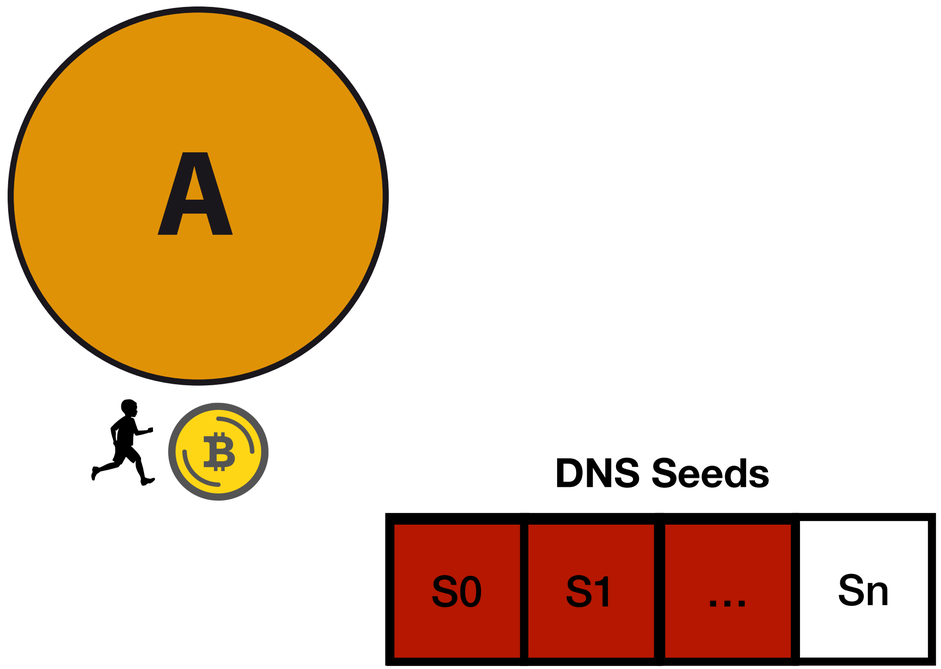
Bitcoin P2P bootstraping (peer discovery)



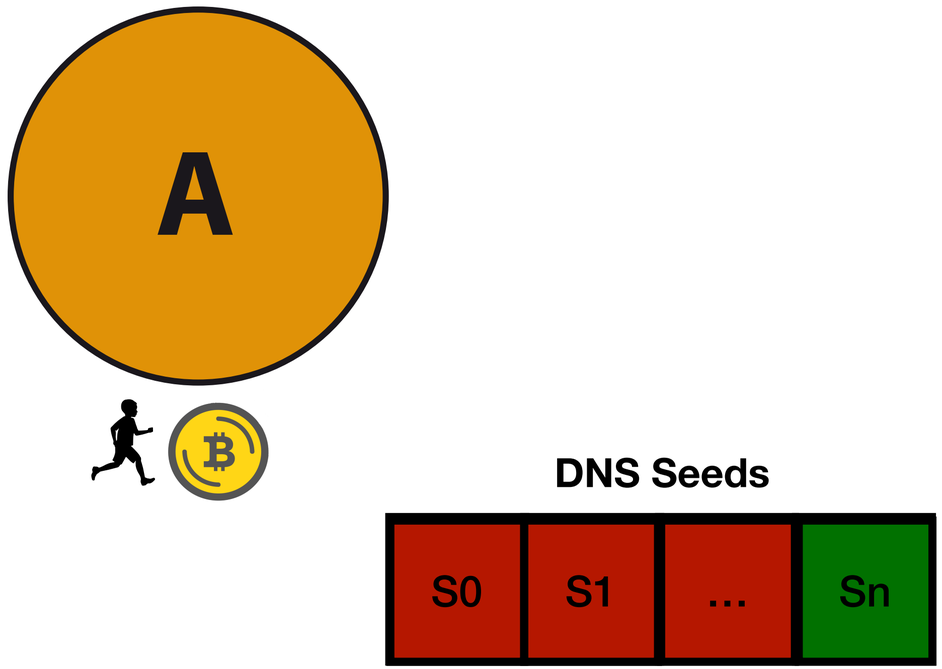
Bitcoin P2P bootstraping (peer discovery)



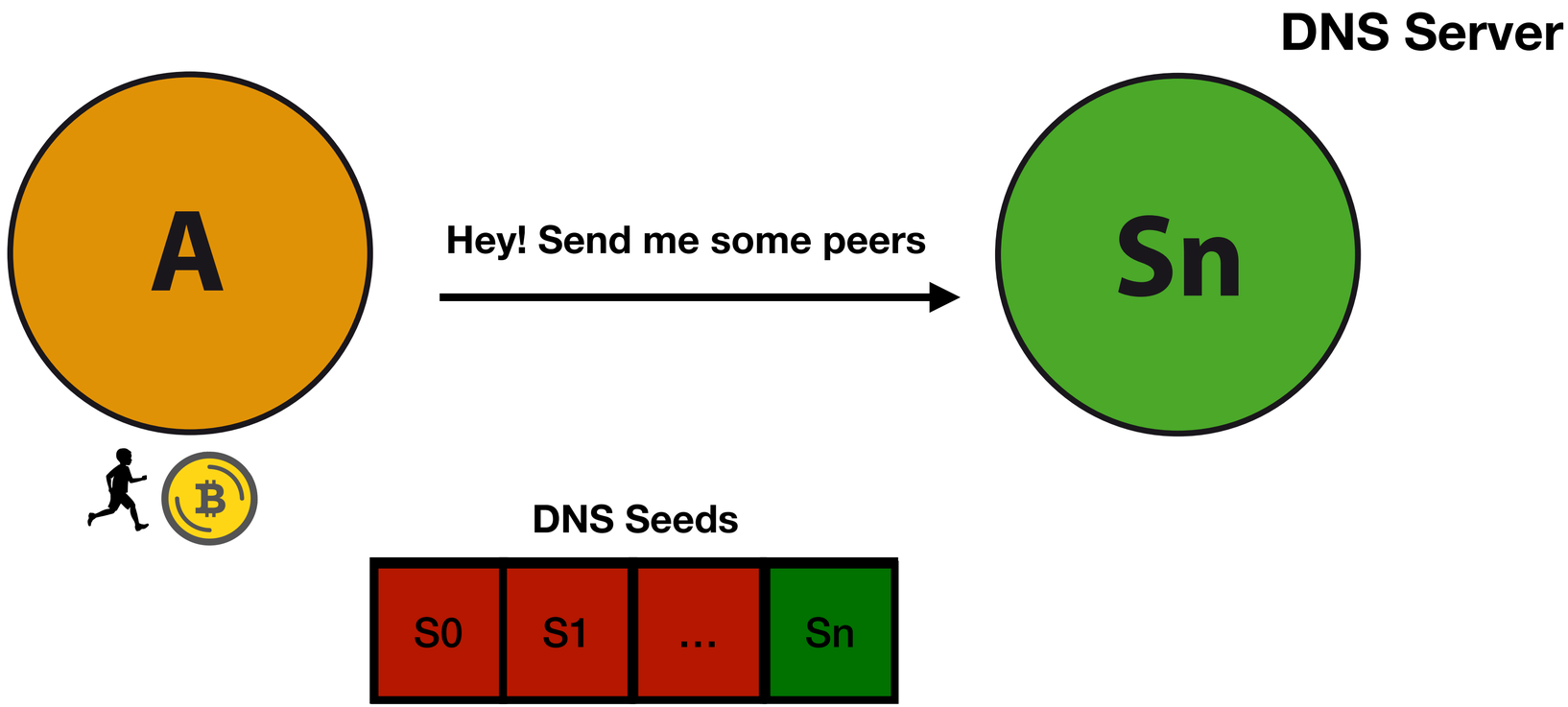
Bitcoin P2P bootstraping (peer discovery)



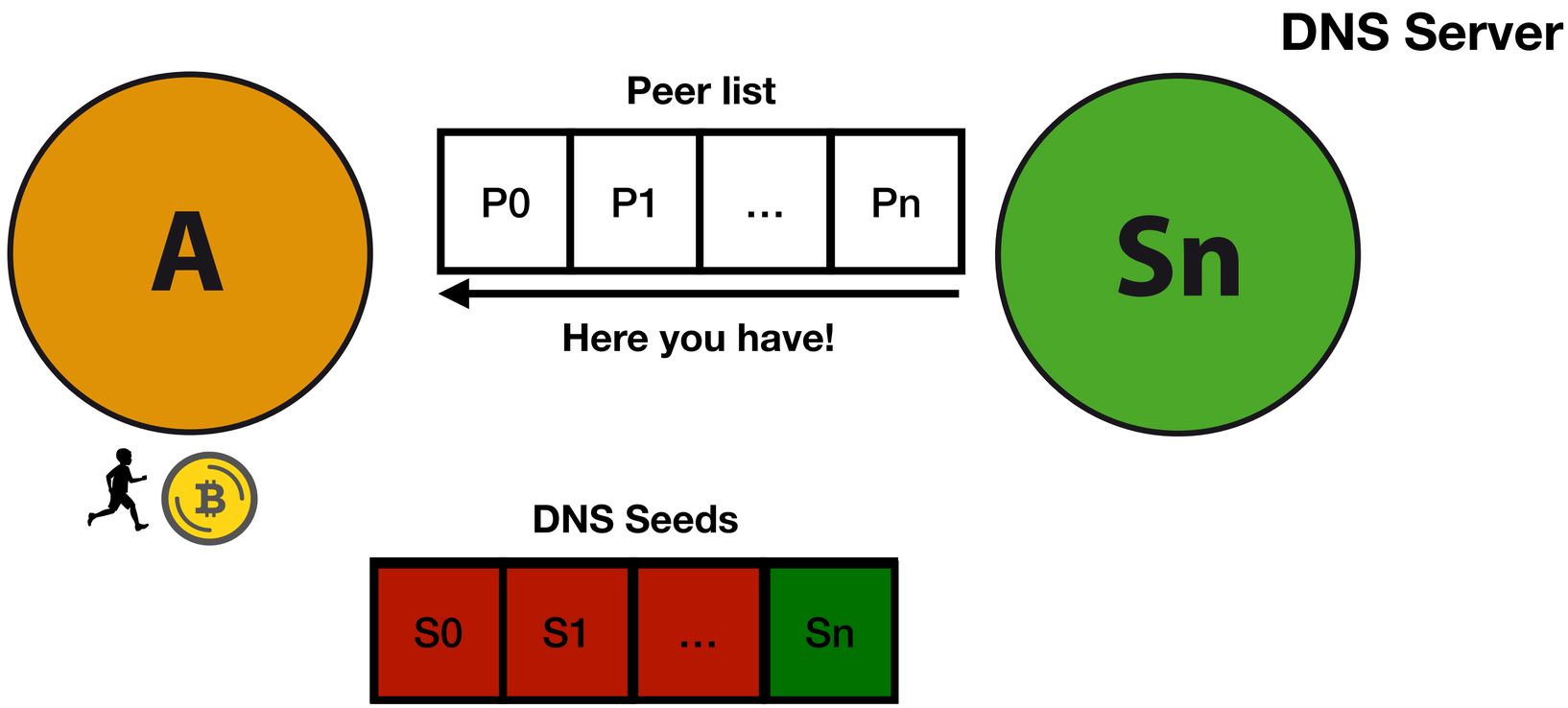
Bitcoin P2P bootstraping (peer discovery)



Bitcoin P2P bootstraping (peer discovery)



Bitcoin P2P bootstraping (peer discovery)

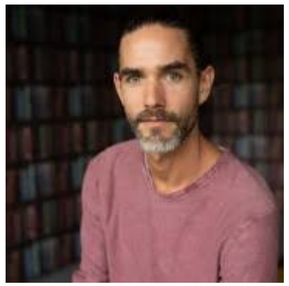


Bitcoin P2P bootstraping (DNS server hosts)

```
vSeeds.emplace_back("seed.bitcoin.sipa.be"); // Pieter Wuille
vSeeds.emplace_back("dnsseed.bluematt.me"); // Matt Corallo
vSeeds.emplace_back("dnsseed.bitcoin.dashjr.org"); // Luke Dashjr
vSeeds.emplace_back("seed.bitcoinstats.com"); // Christian Decker
vSeeds.emplace_back("seed.bitcoin.jonasschnelli.ch"); // Jonas Schnelli
vSeeds.emplace_back("seed.btc.petertodd.org"); // Peter Todd
vSeeds.emplace_back("seed.bitcoin.sprovoost.nl"); // Sjors Provoost
```





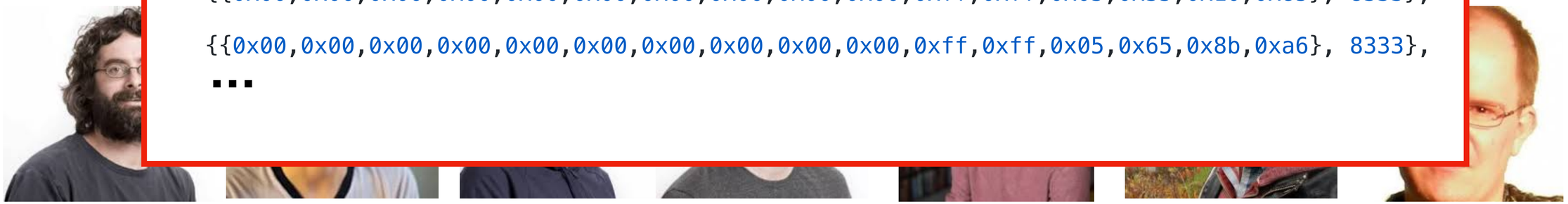


Bitcoin P2P bootstraping (DNS server hosts)

If DNS seeds do not work, a node will try to connect to a hardcoded list of nodes (fixed seed)

```
vSeeds.emplace_back("seed.bitcoin.sipa.be"); // Pieter Wuille
vSeed
vSeed
vSeed
vSeed                                                                    i
vSeed
vSeed
```

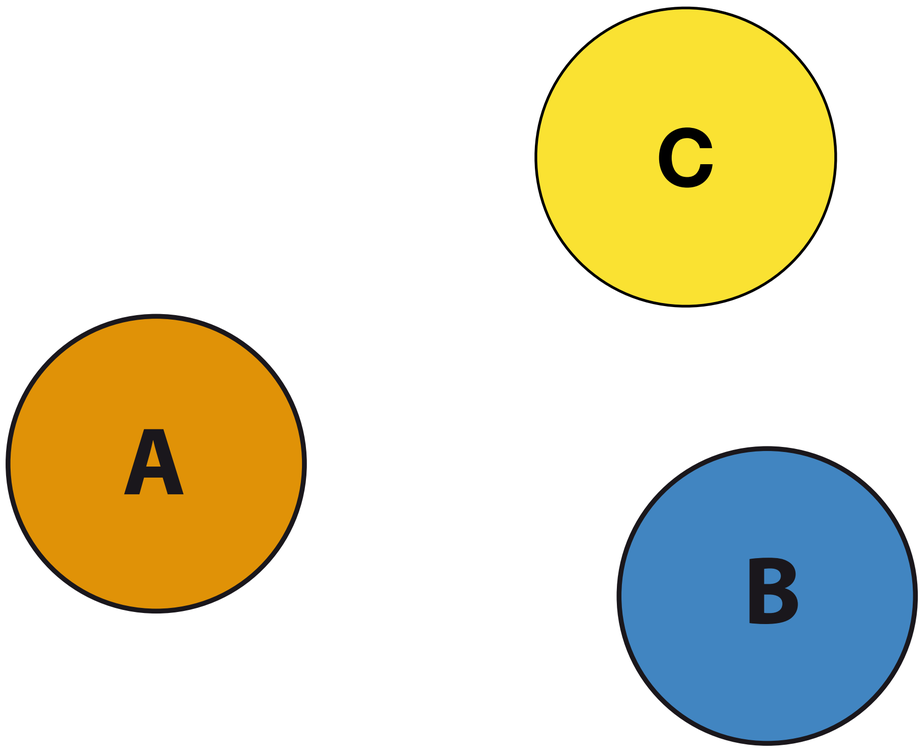
```
static SeedSpec6 pnSeed6_main[] = {
    {{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xff,0xff,0x02,0x84,0x64,0x2f}, 8333},
    {{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xff,0xff,0x05,0x01,0x61,0x04}, 8333},
    {{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xff,0xff,0x05,0x27,0xae,0x74}, 8333},
    {{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xff,0xff,0x05,0x2d,0x4f,0x0e}, 8333},
    {{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xff,0xff,0x05,0x35,0x10,0x85}, 8333},
    {{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xff,0xff,0x05,0x65,0x8b,0xa6}, 8333},
    ...
```



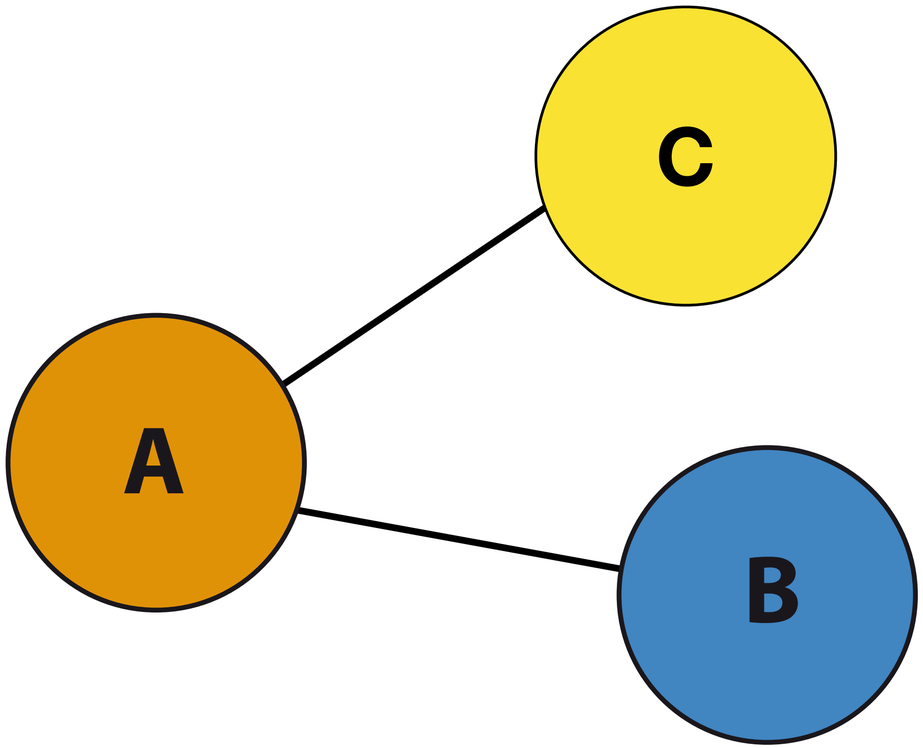
Bitcoin P2P bootstraping (summary)

- 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

Populating the peers database (addrman)

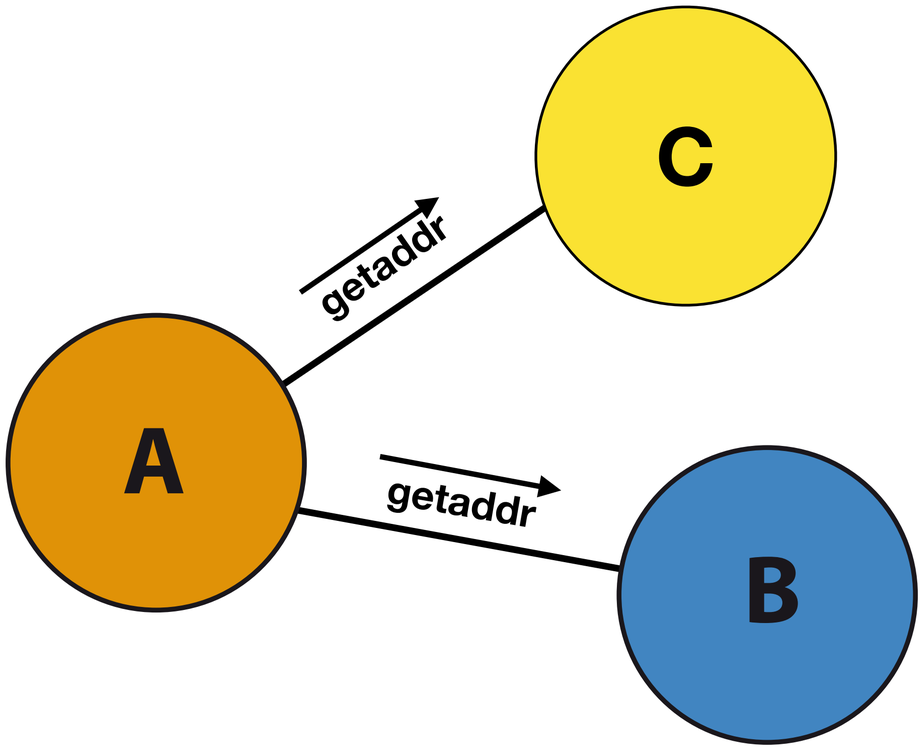


Populating the peers database (addrman)



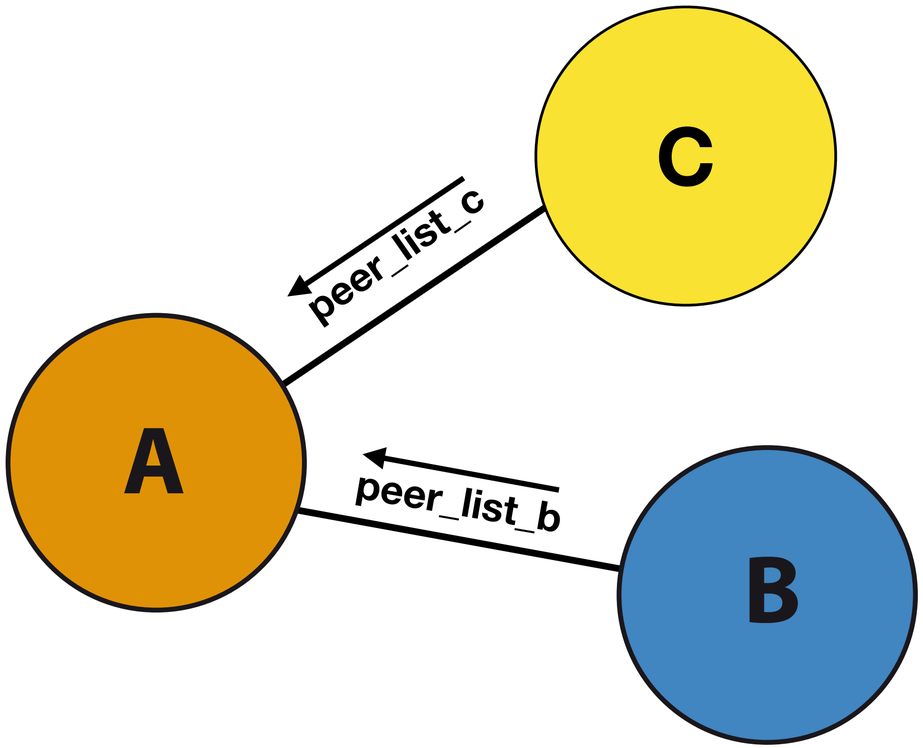
- A connects a subset of peers from the ones learned from the DNS seeds

Populating the peers database (addrman)



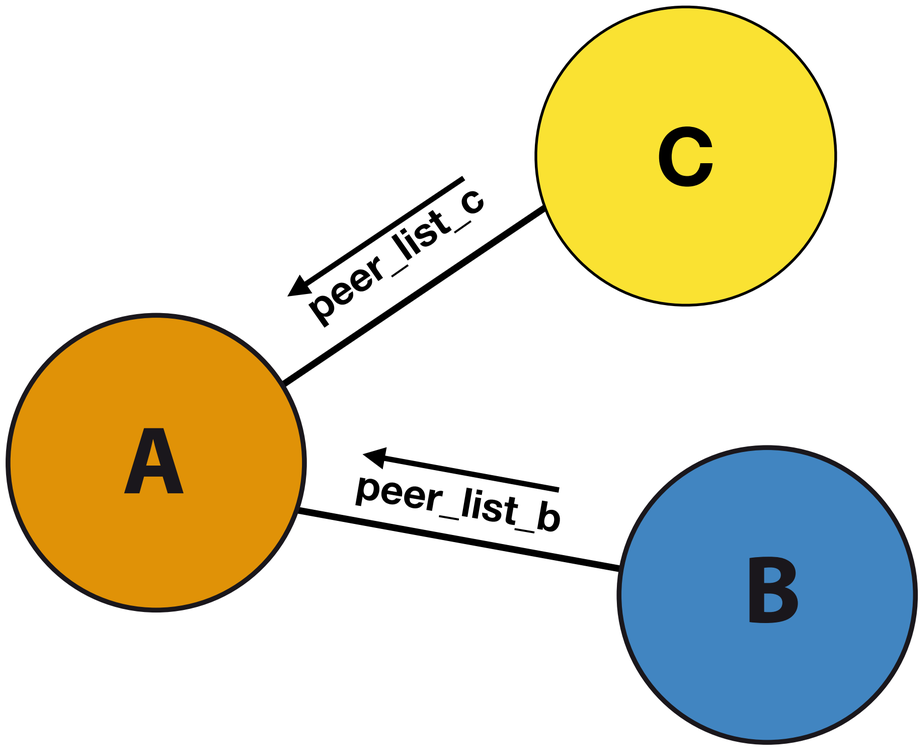
- A connects a subset of peers from the ones learned from the DNS seeds
- A requests more peers to his neighbors (**getaddr**)

Populating the peers database (addrman)



- A connects a subset of peers from the ones learned from the DNS seeds
- A requests more peers to his neighbors (**getaddr**)
- Peers reply with some addresses they known about (**addr, up to 1000 addresses**)

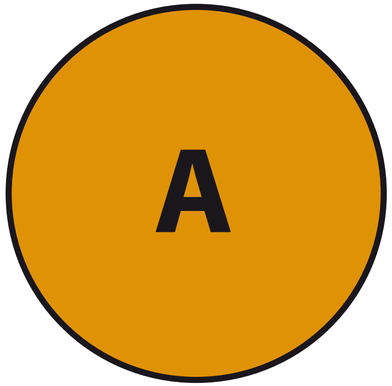
Populating the peers database (addrman)



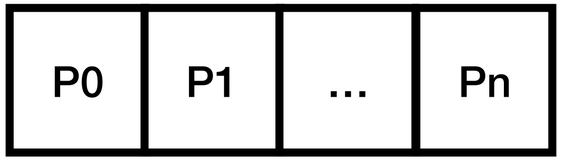
A's addrman = A's addrman U peer_list_c U peer_list_b

- A connects a subset of peers from the ones learned from the DNS seeds
- A requests more peers to his neighbors (**getaddr**)
- Peers reply with some addresses they known about (**addr, up to 1000 addresses**)
- A adds the new addresses to its peers database (or update the existing ones)

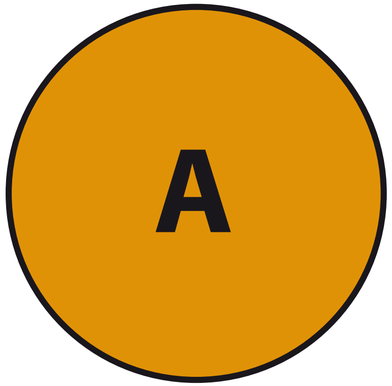
Bitcoin P2P bootstraping (connections)



Peer database (addrman)

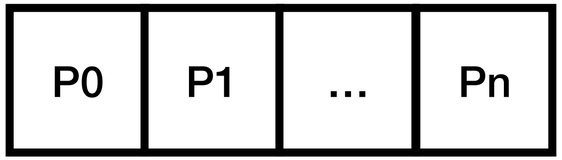


Bitcoin P2P bootstraping (connections)

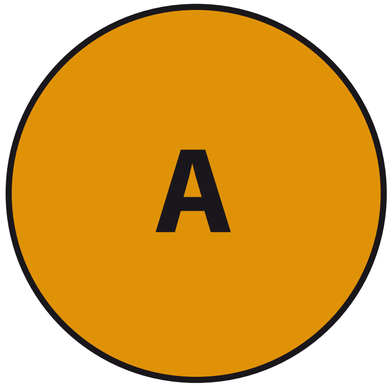


- During bootstrap, a node will start some **outgoing connections** with peers it has learn about (**8 by default**) and try to maintain them

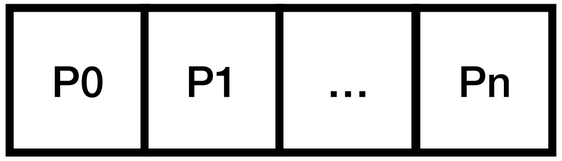
Peer database (addrman)



Bitcoin P2P bootstraping (connections)



Peer database (addrman)



- During bootstrap, a node will start some **outgoing connections** with peers it has learn about (**8 by default**) and try to maintain them

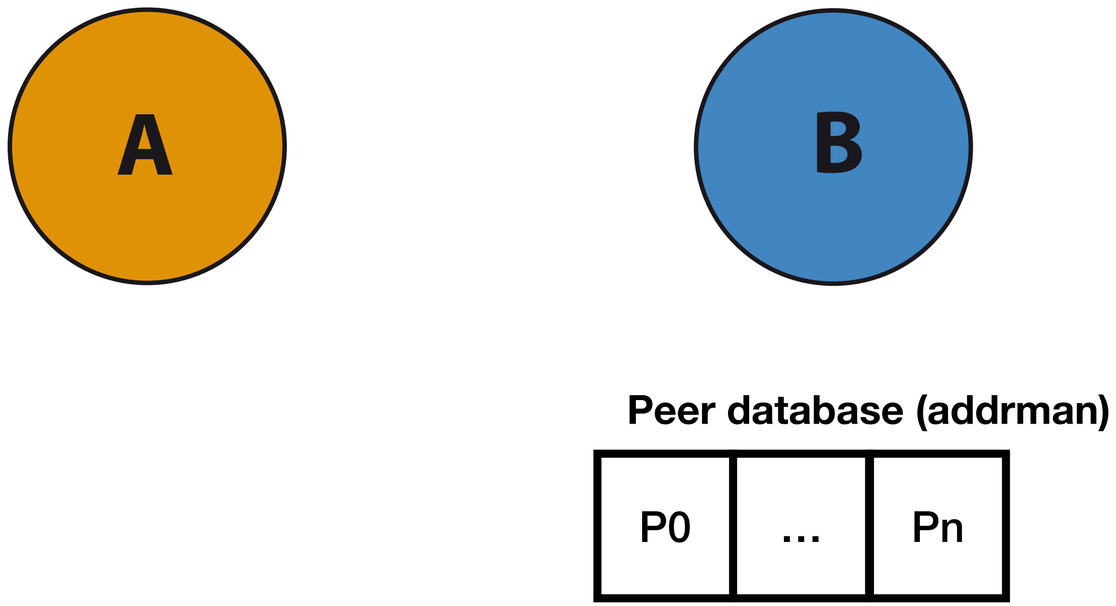
- A node will also accept **some incoming** connections (**117 by default**)

Hellow world!

- How does a node announce his presence to the rest of the network?

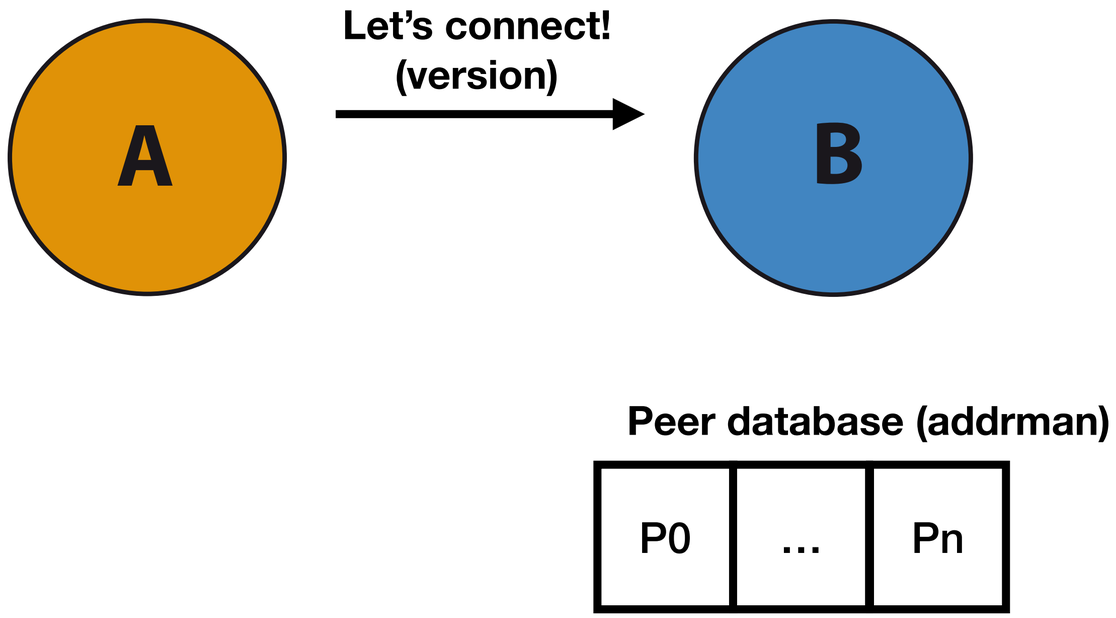
Hellow world!

- How does a node announce his presence to the rest of the network?



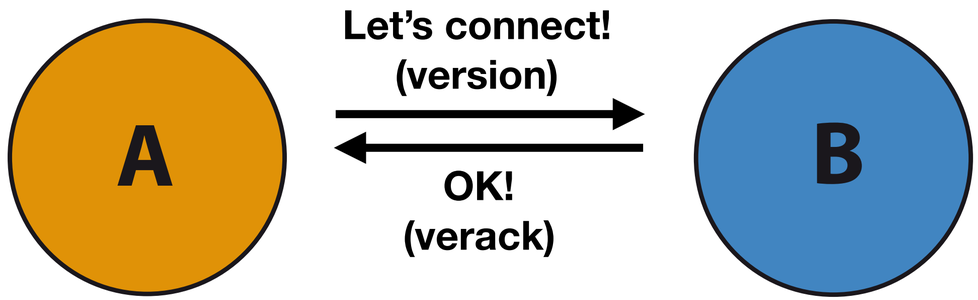
Hellow world!

- How does a node announce his presence to the rest of the network?



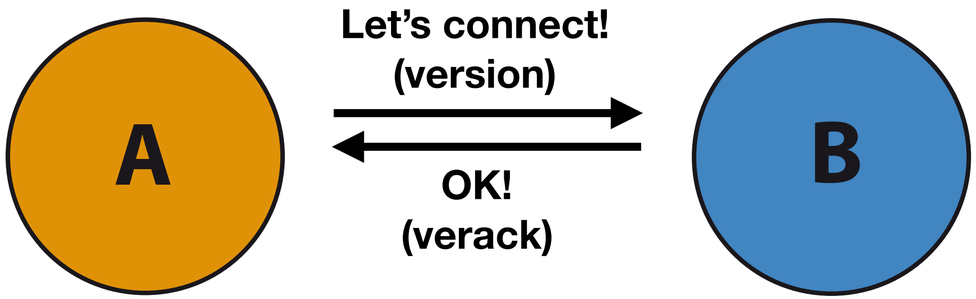
Hellow world!

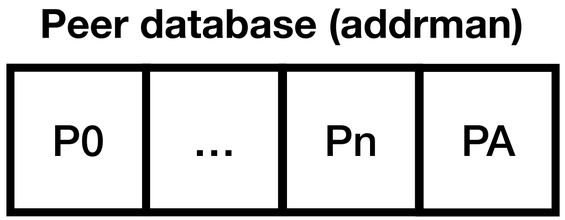
- How does a node announce his presence to the rest of the network?



Hellow world!

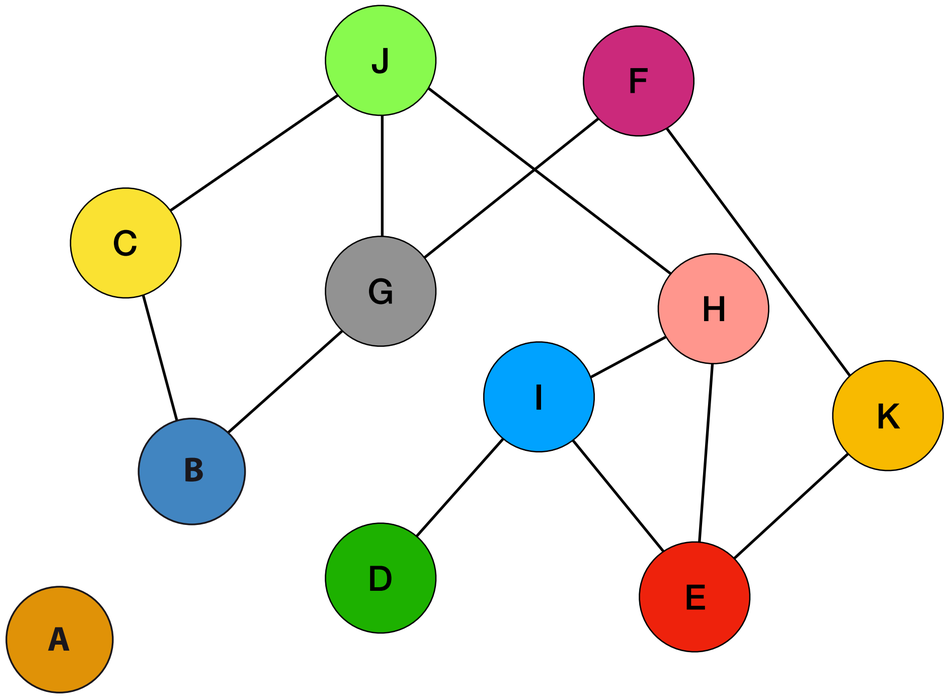
- How does a node announce his presence to the rest of the network?





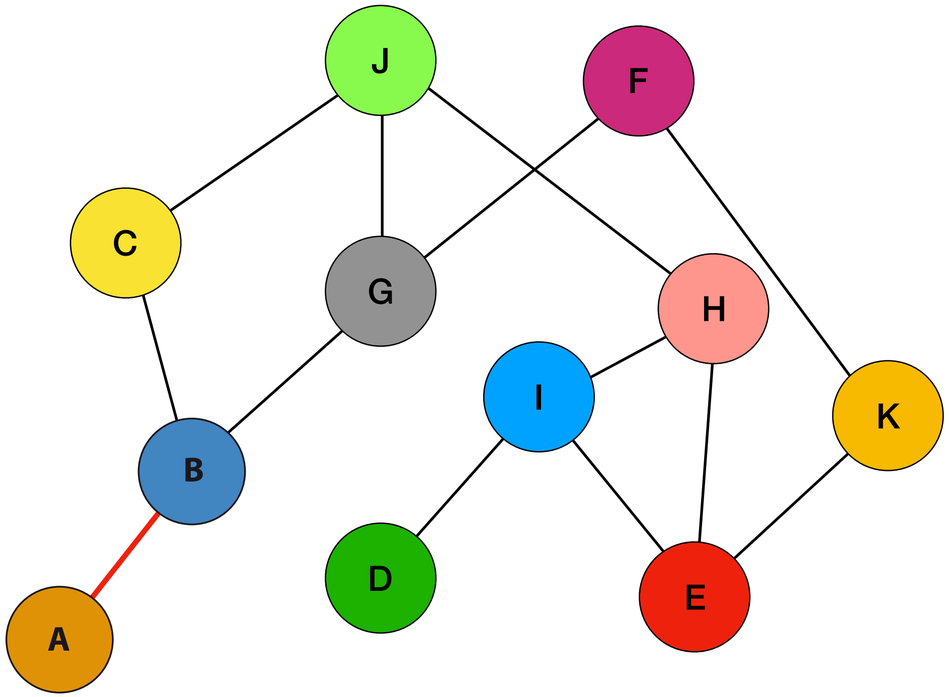
Hello world!

- How does a node announce his presence to the rest of the network?



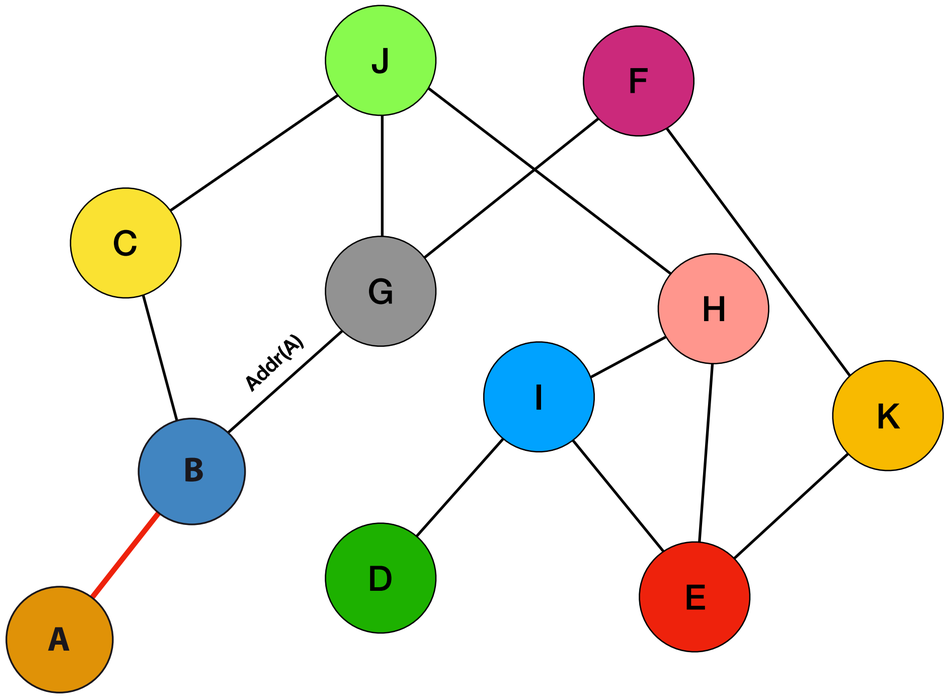
Hello world!

- How does a node announce his presence to the rest of the network?



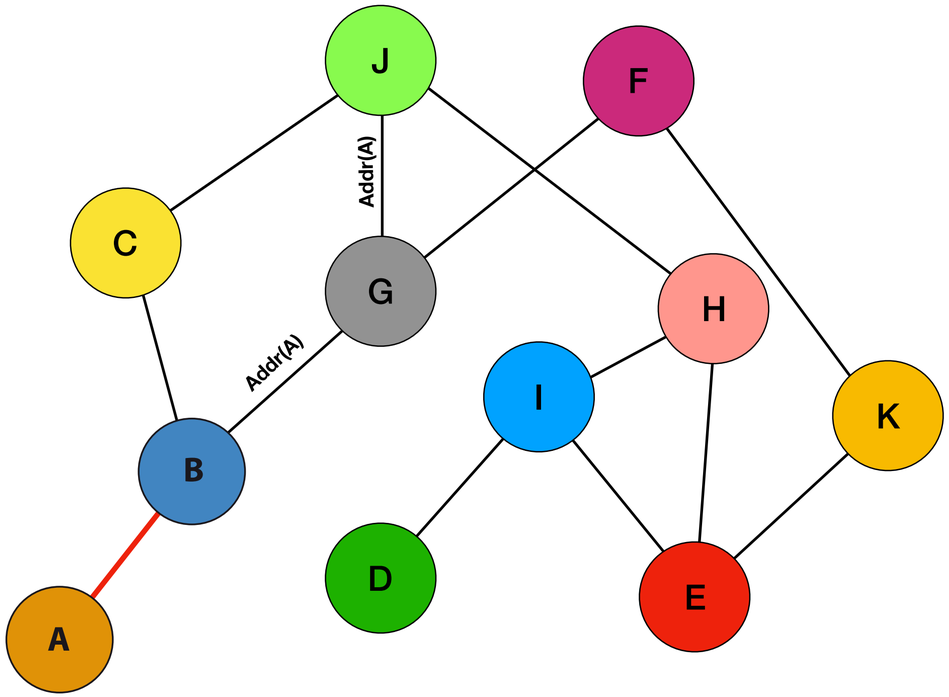
Hello world!

- How does a node announce his presence to the rest of the network?



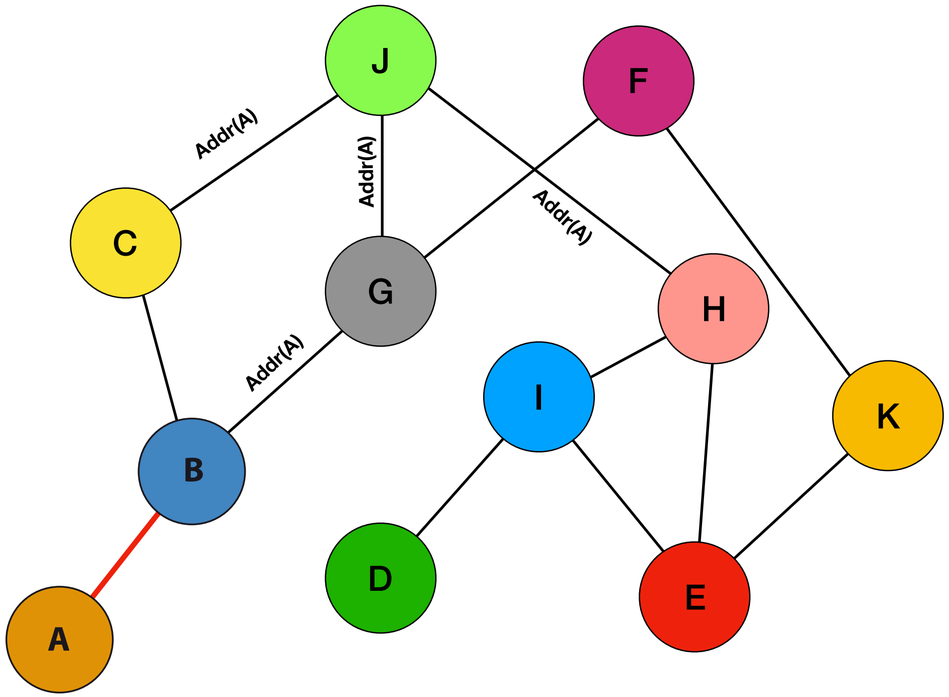
Hello world!

- How does a node announce his presence to the rest of the network?



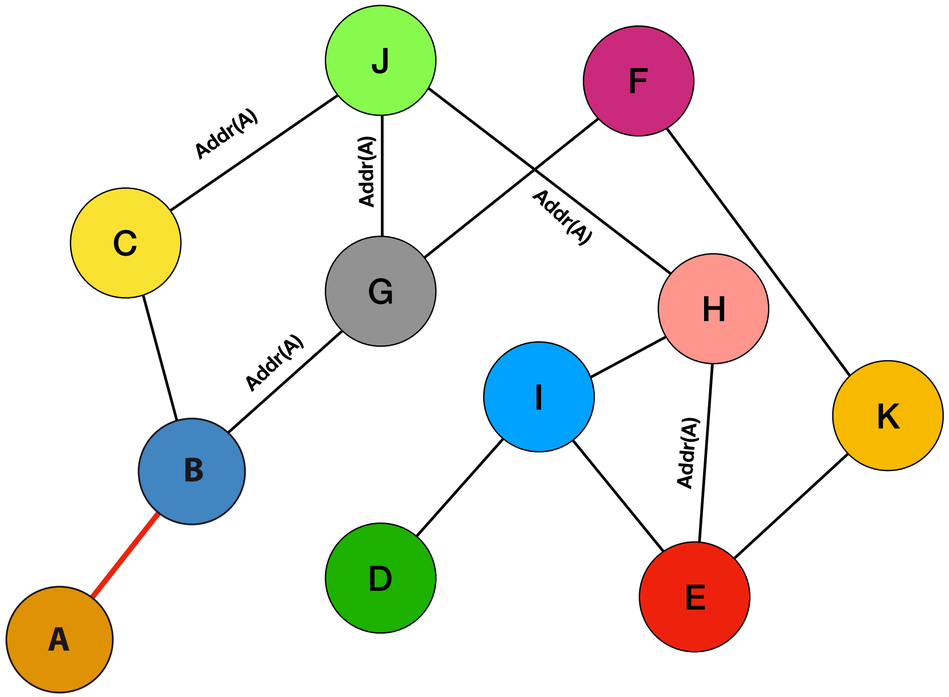
Hello world!

- How does a node announce his presence to the rest of the network?



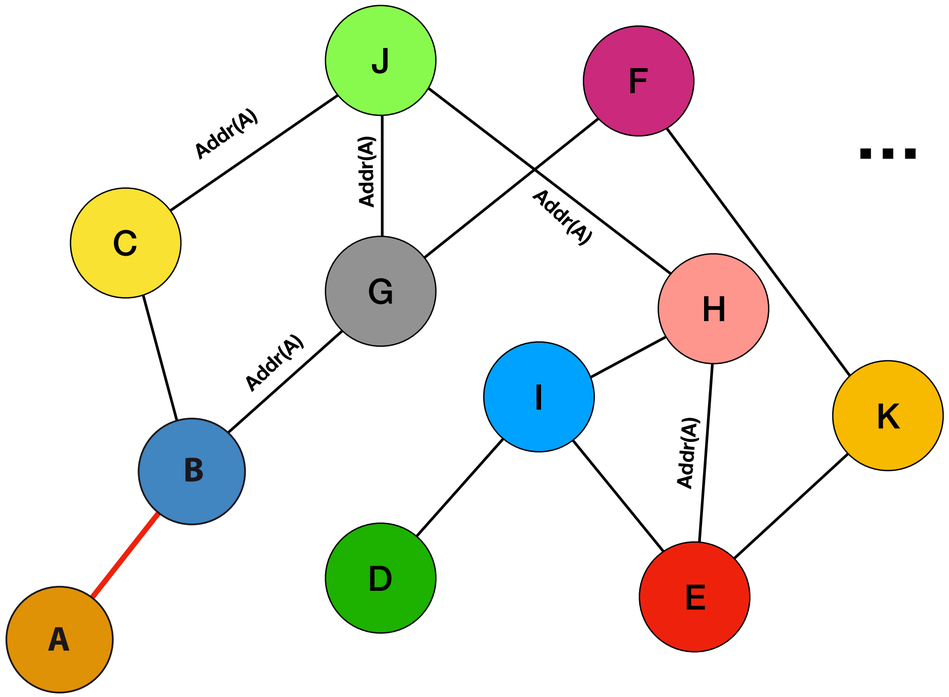
Hello world!

- How does a node announce his presence to the rest of the network?



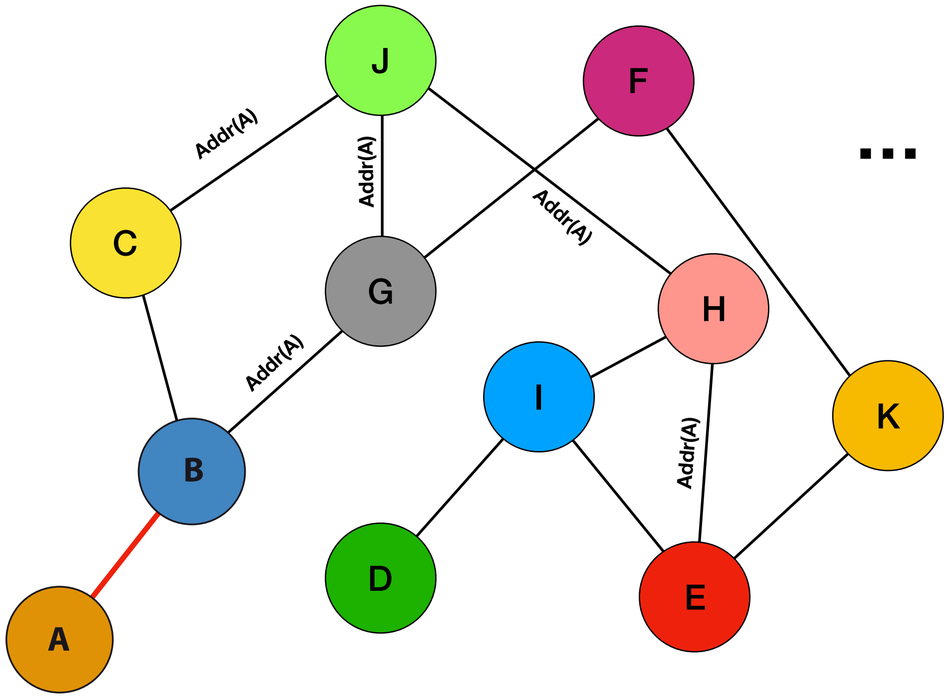
Hello world!

- How does a node announce his presence to the rest of the network?



Hello world!

- How does a node announce his presence to the rest of the network?

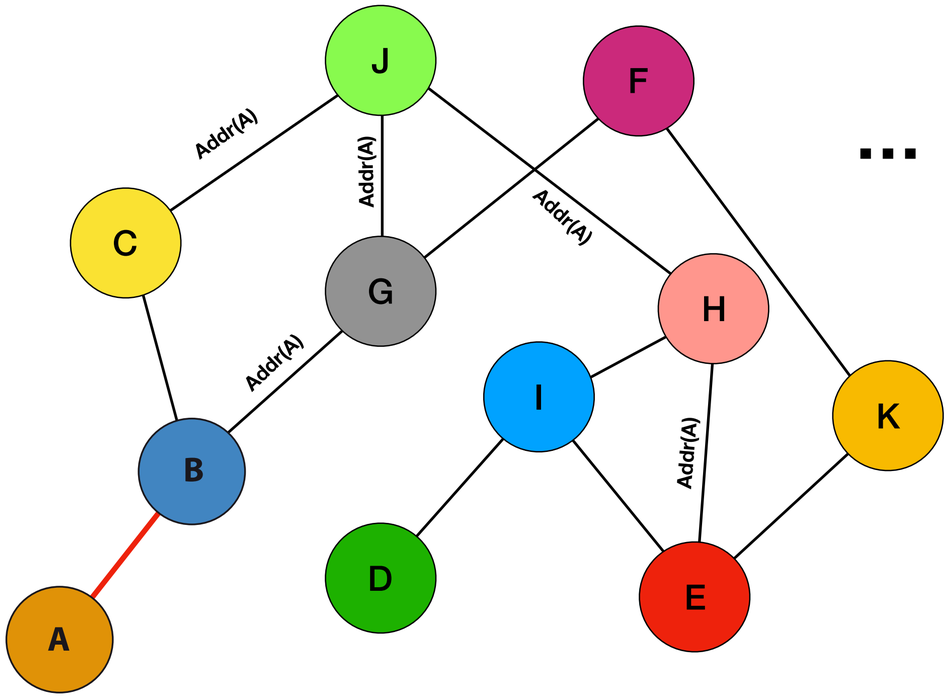


...

- The address will eventually be spread throughout the network

Hello world!

- How does a node announce his presence to the rest of the network?

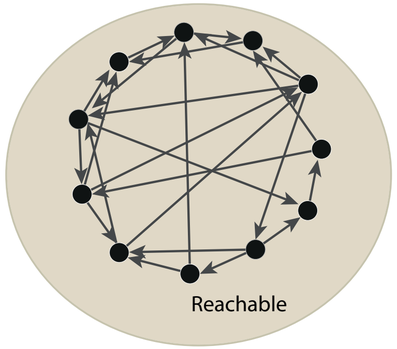


- The address will eventually be spread throughout the network
- Nodes learning about the new peer will add it to their peers database

Cryptocurrency networks taxonomy

Cryptocurrency networks taxonomy

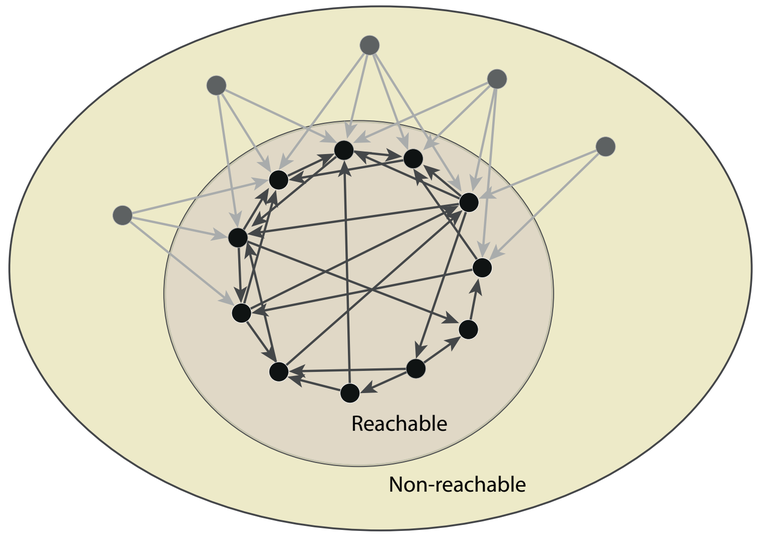
- Reachable network: all nodes accept incoming / outgoing connections



Cryptocurrency networks taxonomy

- Reachable network: all nodes accept incoming / outgoing connections

- Non-reachable: nodes do not accept incoming connections / cannot be reached (NAT/firewalls/…)

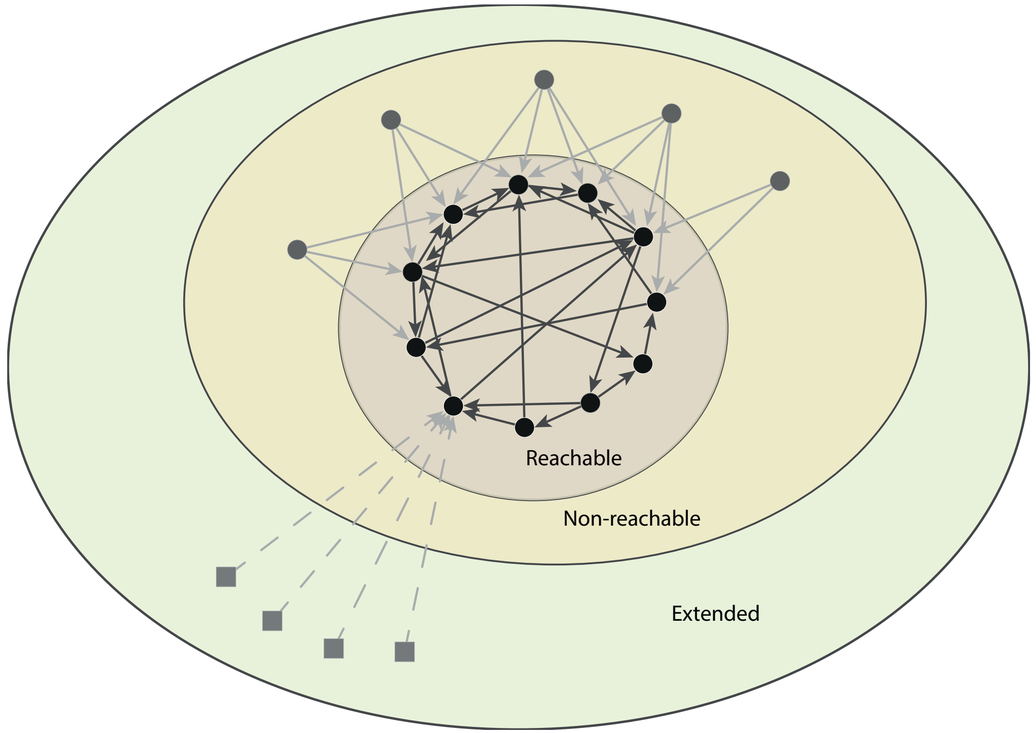


Cryptocurrency networks taxonomy

- 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)



Actors and purpose
(what, who, why, and how)

The data (what?)

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

The data (what?)

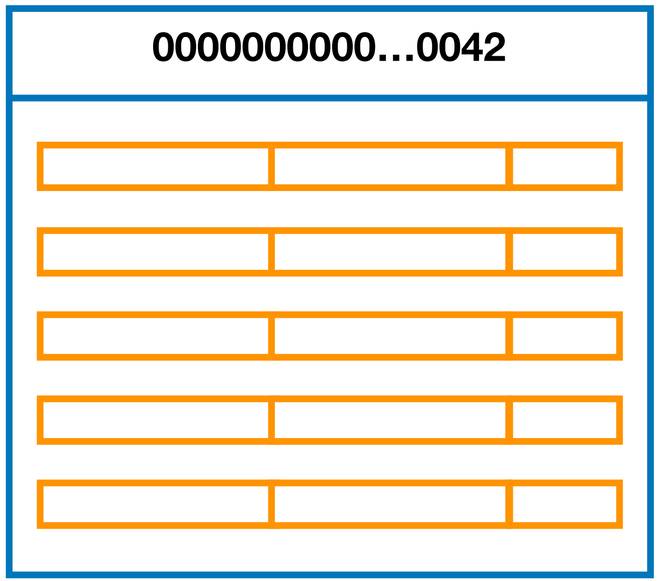
- There are two main items that peers share in a cryptocurrency P2P network: **transactions** and **blocks**

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)

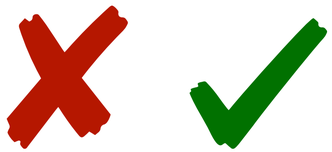
- There are two main roles followed by nodes: **peers** and **miners**

- (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)

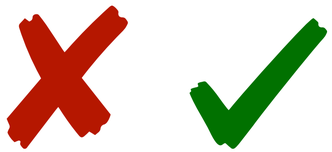
- There are two main roles followed by nodes: **peers** and **miners**

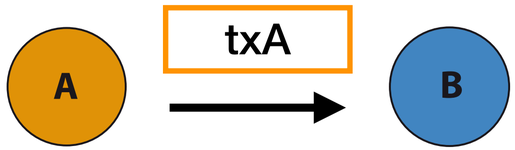
- (Normal) **Peers:**

  - 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)





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)

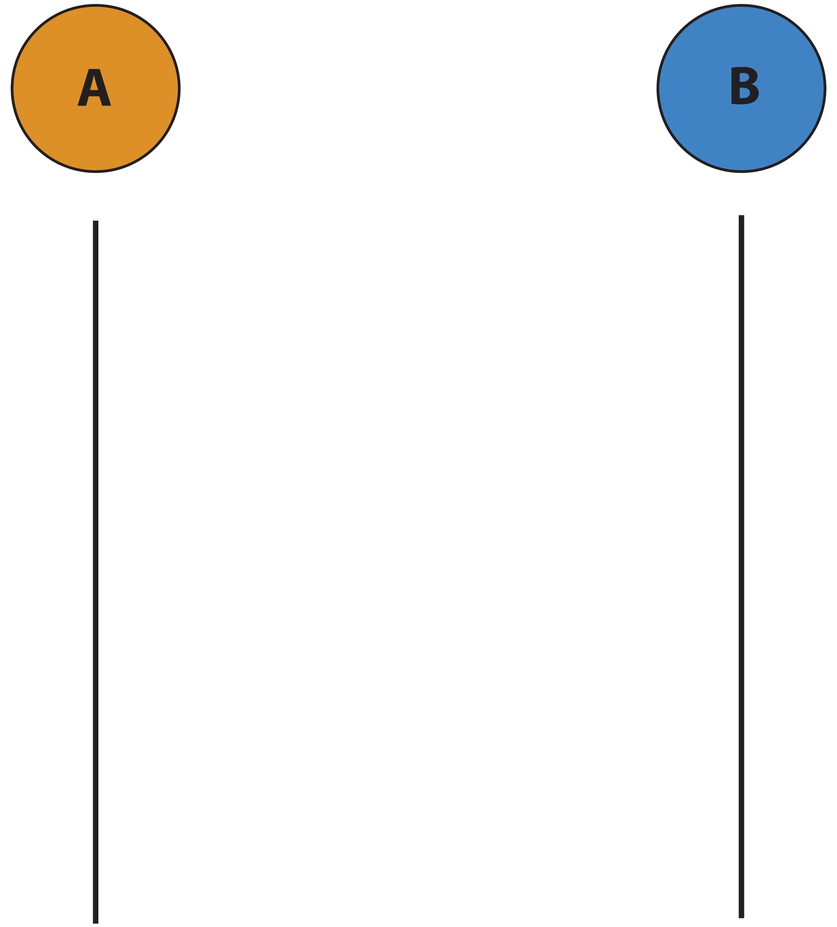
- Blocks are relayed to ultimately achieve a consistent view of the blockchain

- 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?)

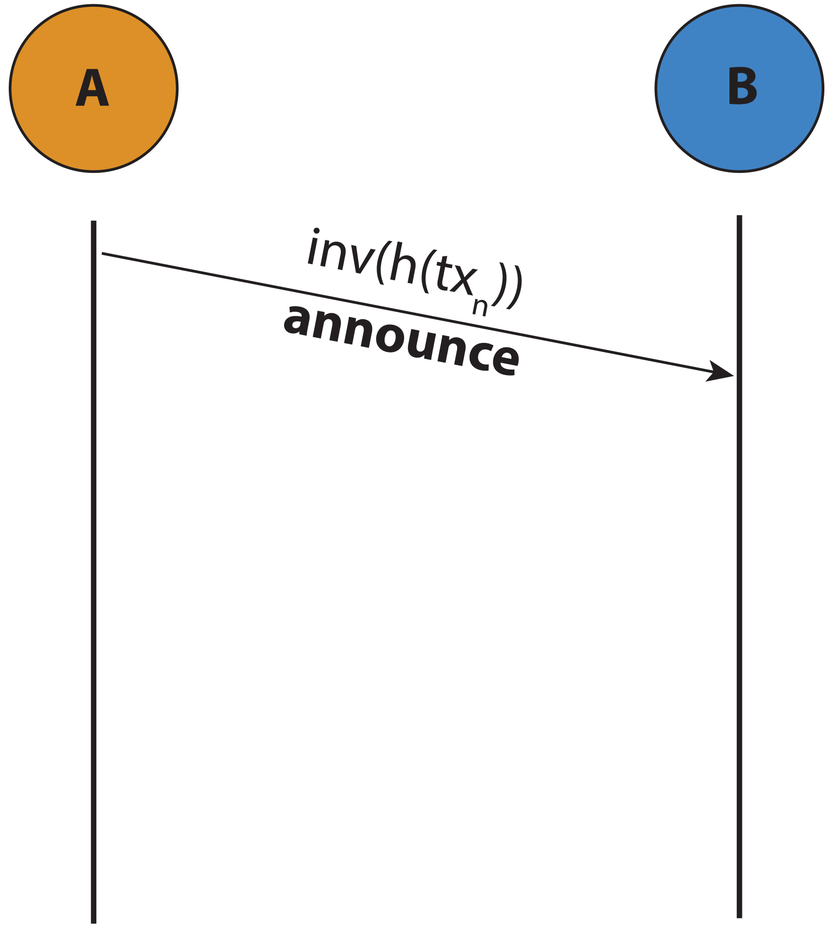
- Items (transactions and blocks) are shared between peers in a push manner



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**)

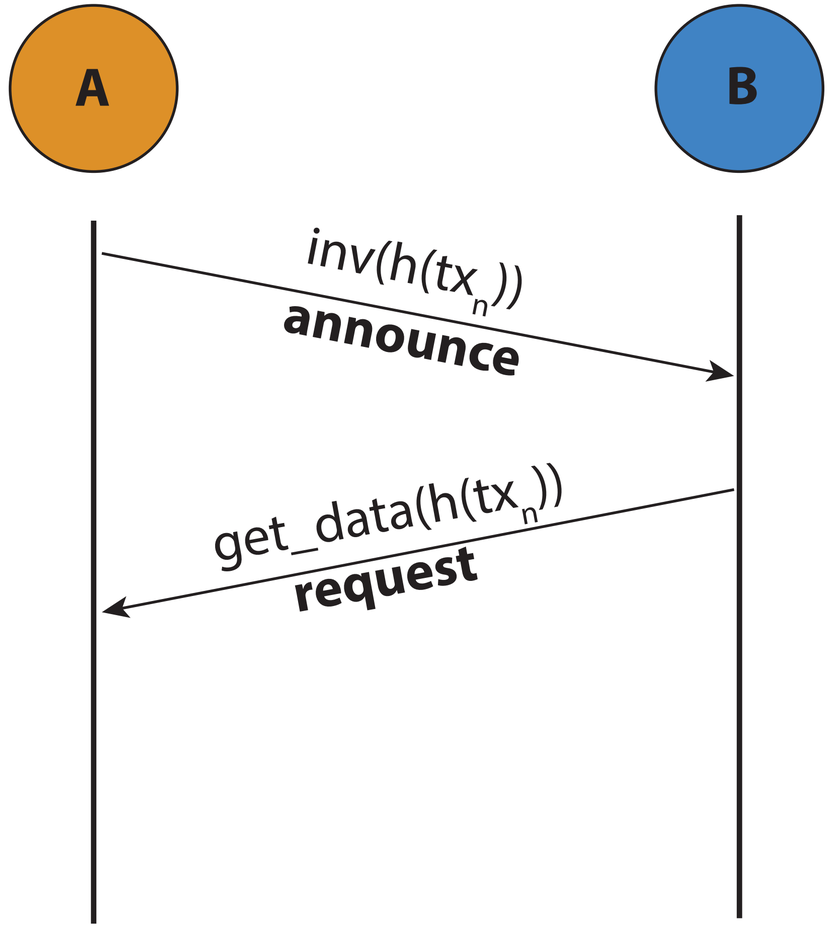


The gossip protocol (how?)

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

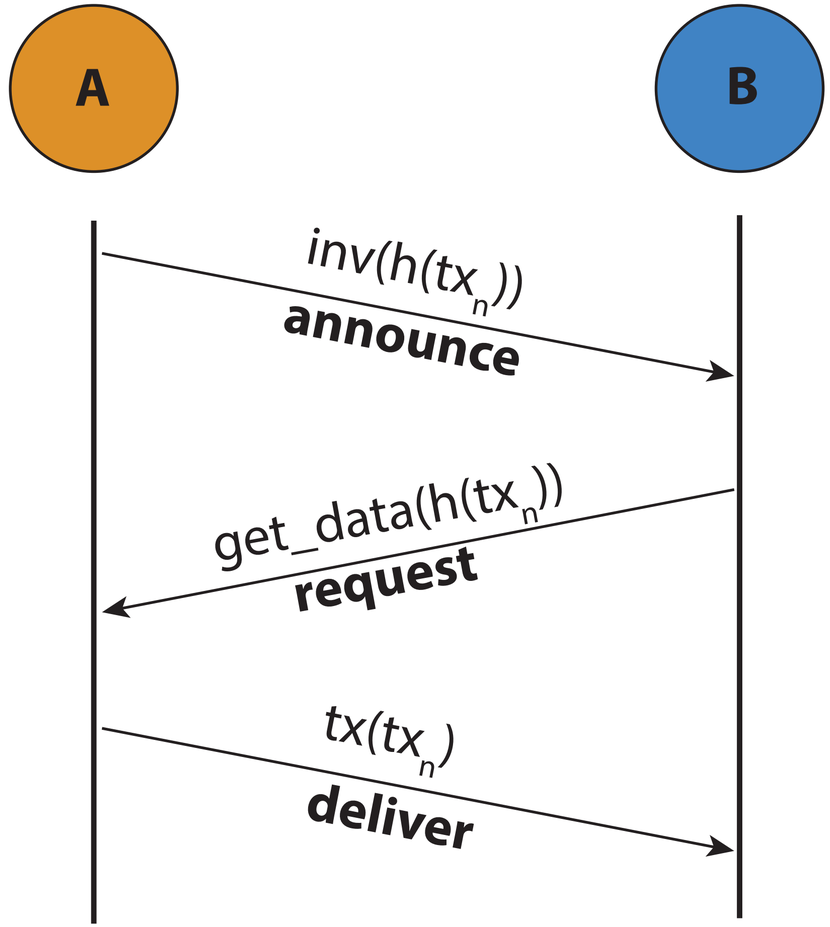
- When a peer receives / generates a new item he announce it to his neighbors (**announce**)

- 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?)

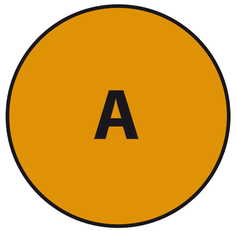
- Items (transactions and blocks) are shared between peers in a push manner
- When a peer receives / generates a new item he announce it to his neighbors (**announce**)
- Upon receiving an announce of an item, a node that does not know about it will request such item back to the announcer (**request**)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



Information propagation

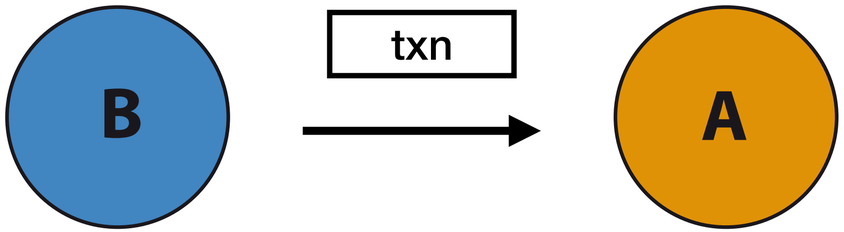
Information propagation (1/2)

Information propagation (1/2)

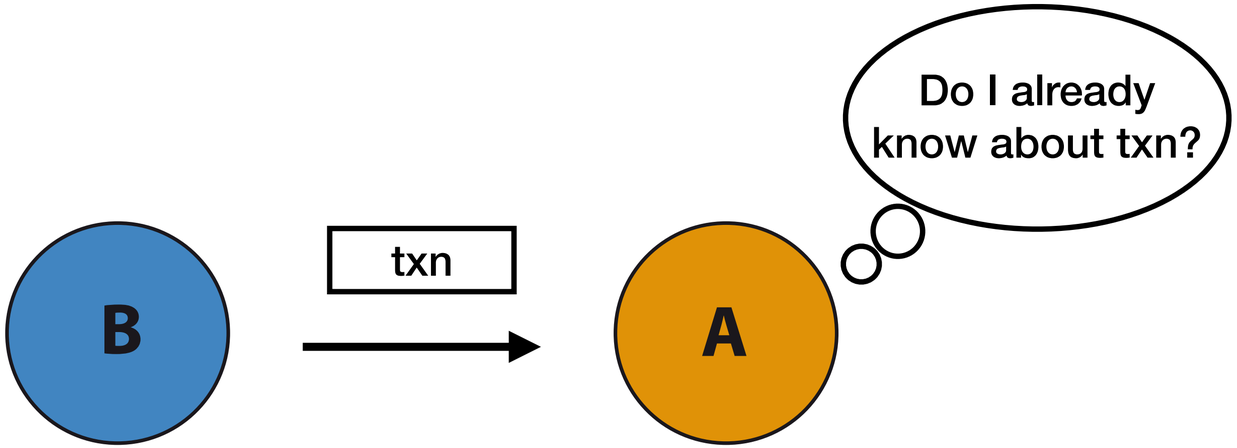


Information propagation (1/2)

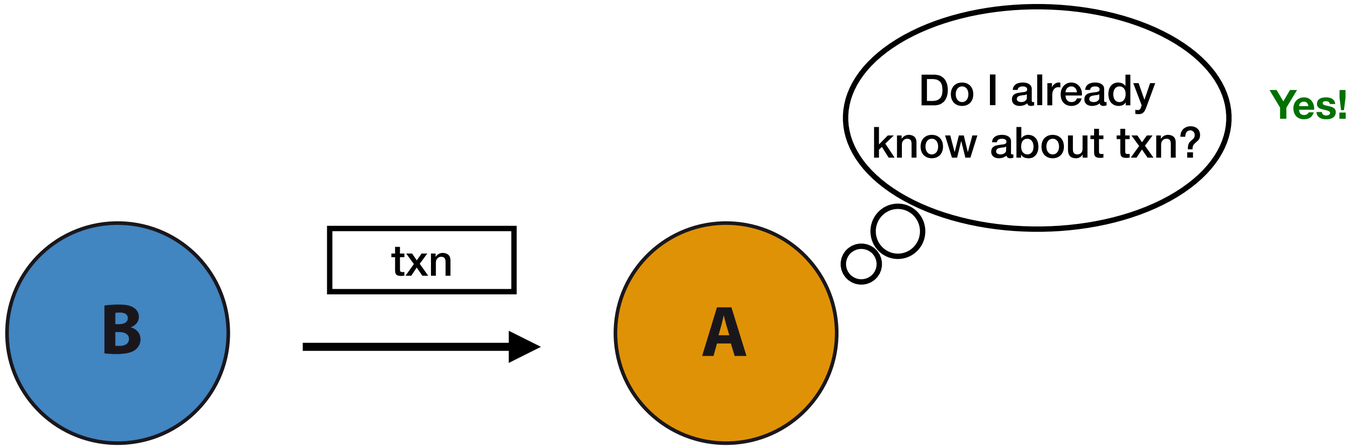
Information propagation (1/2)



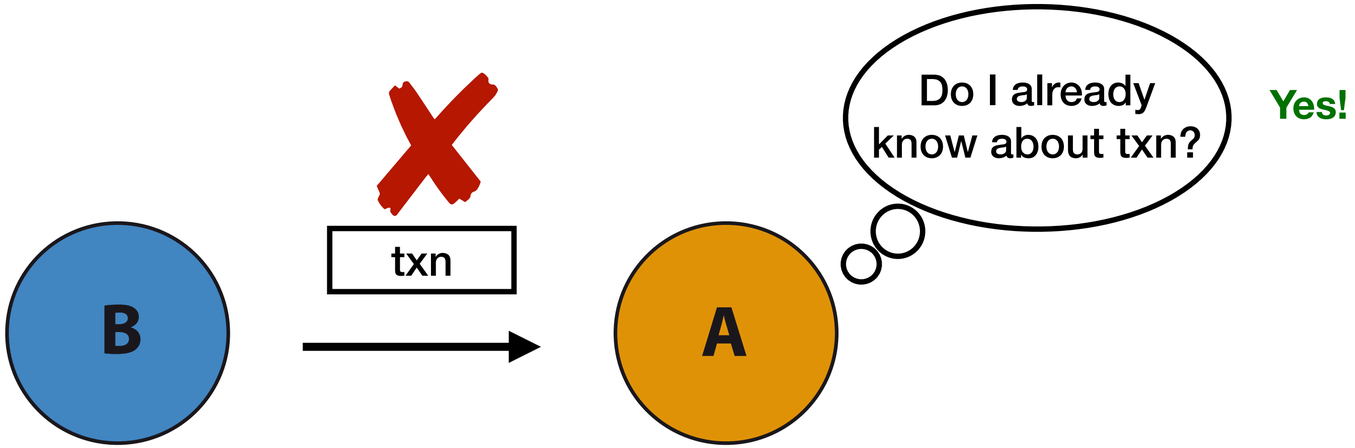
Information propagation (1/2)



Information propagation (1/2)

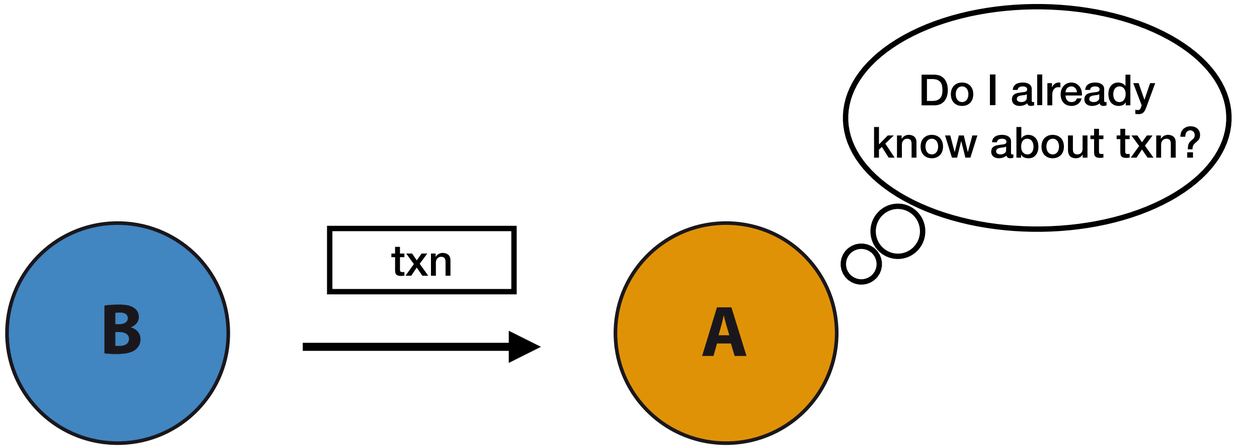


Information propagation (1/2)



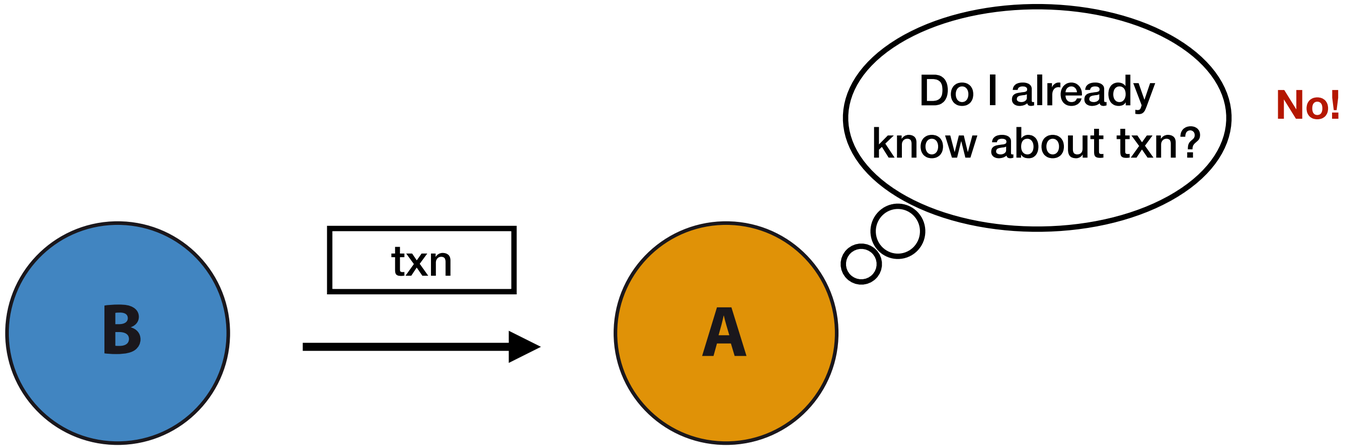
- Known transaction will be rejected

Information propagation (1/2)



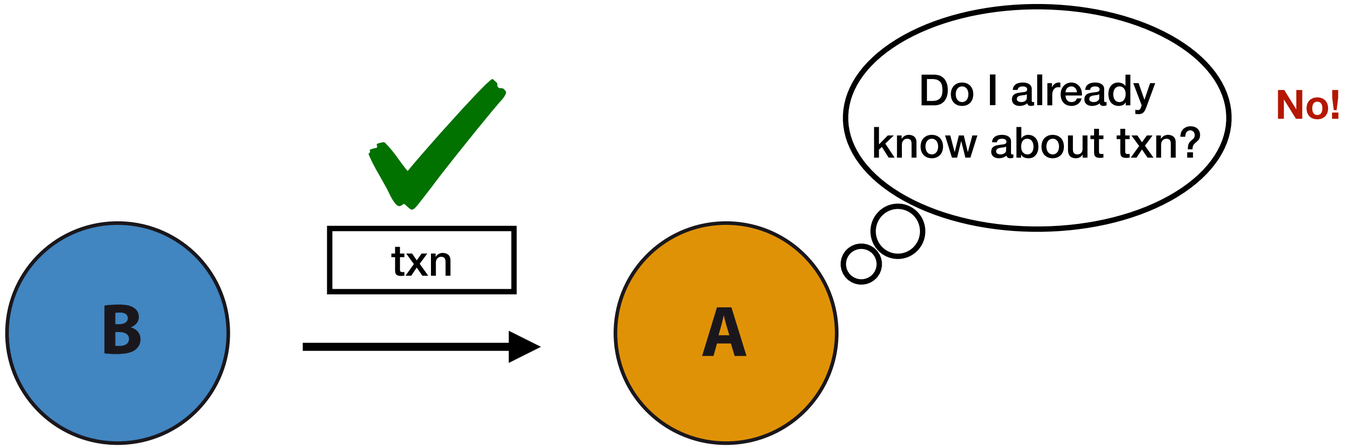
- Known transaction will be rejected

Information propagation (1/2)



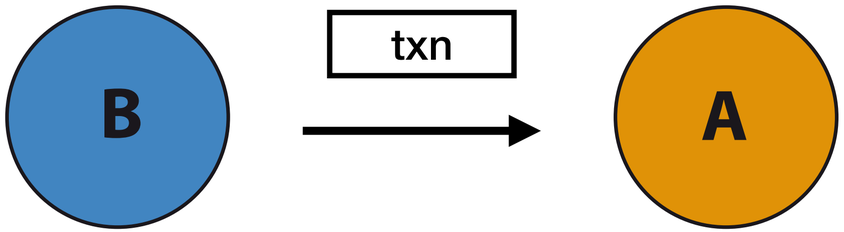
- Known transaction will be rejected

Information propagation (1/2)



- Known transaction will be rejected

Information propagation (1/2)



- Known transaction will be rejected

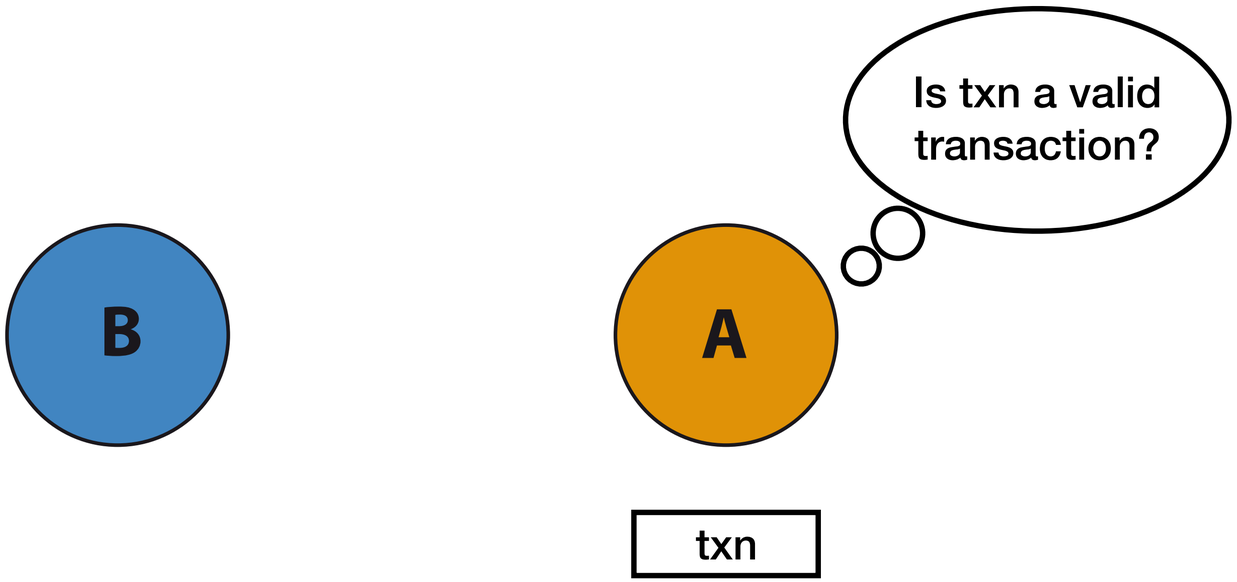
Information propagation (1/2)

- Known transaction will be rejected

Information propagation (1/2)

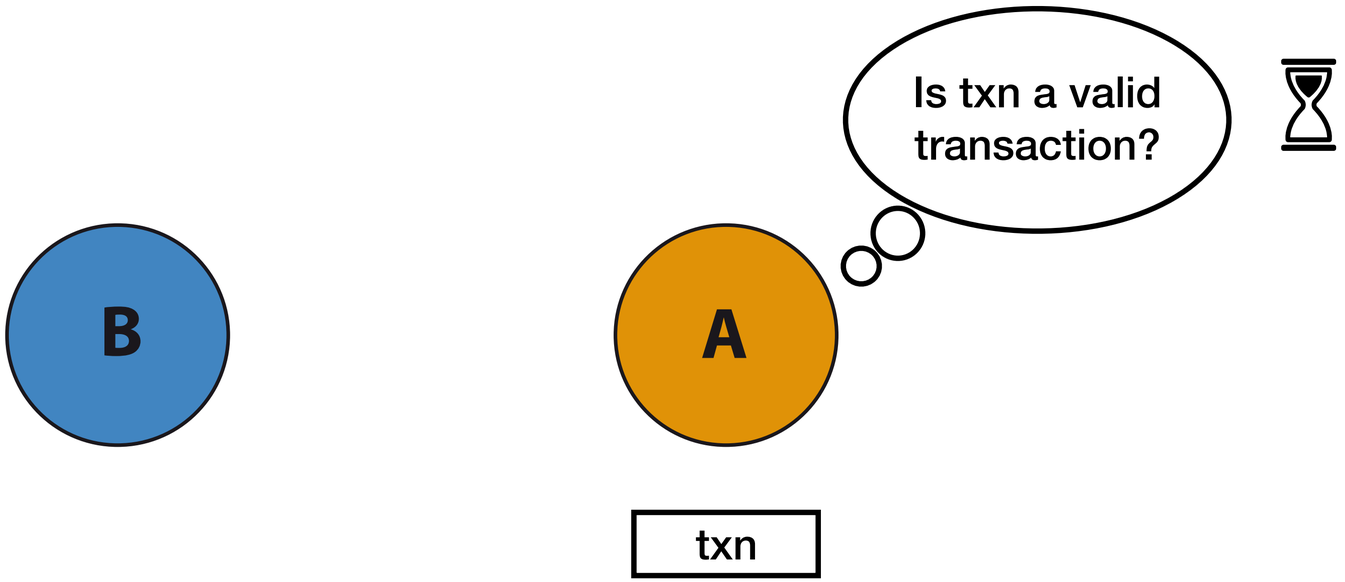
- Known transaction will be rejected

Information propagation (1/2)



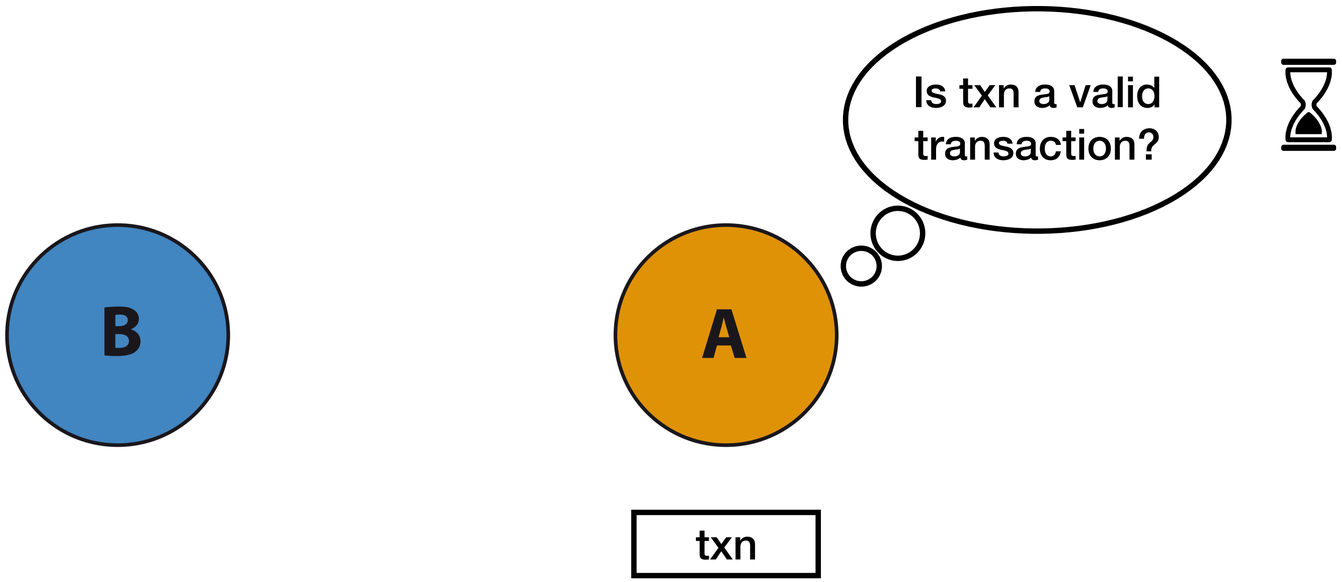
- Known transaction will be rejected

Information propagation (1/2)



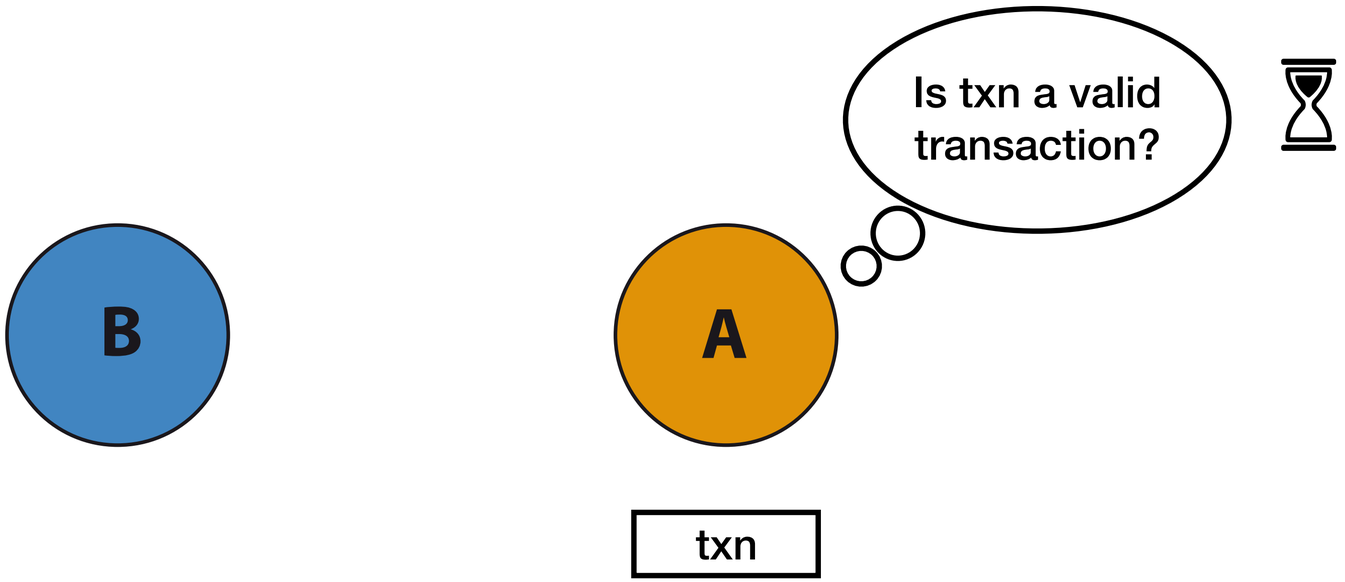
- Known transaction will be rejected

Information propagation (1/2)



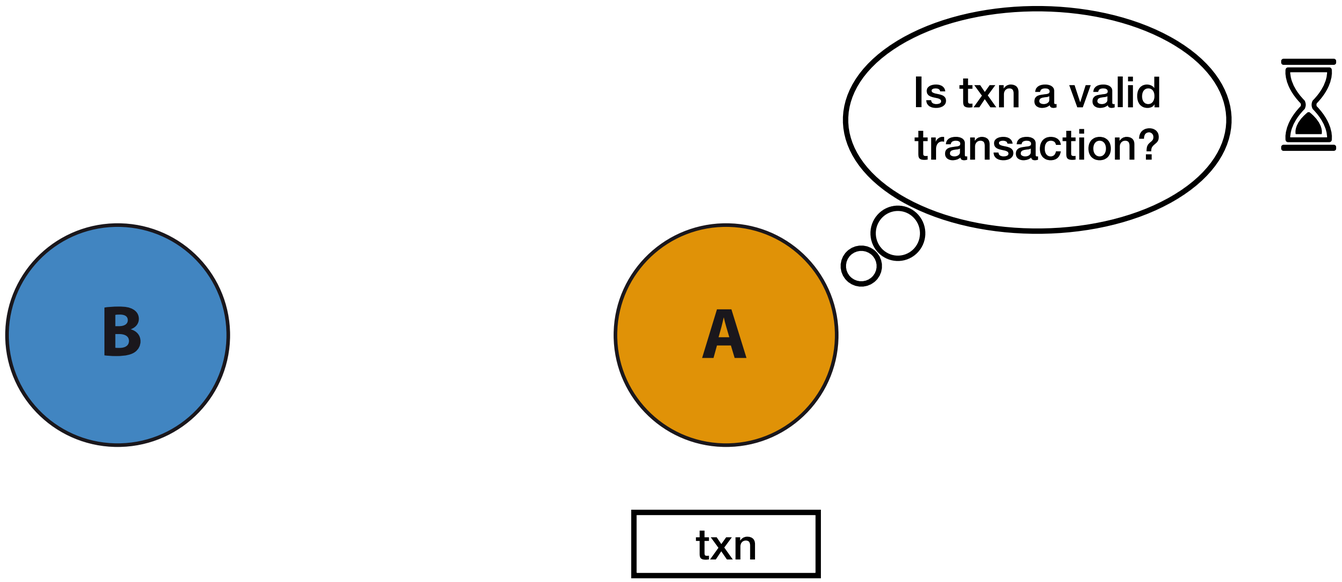
- Known transaction will be rejected

Information propagation (1/2)



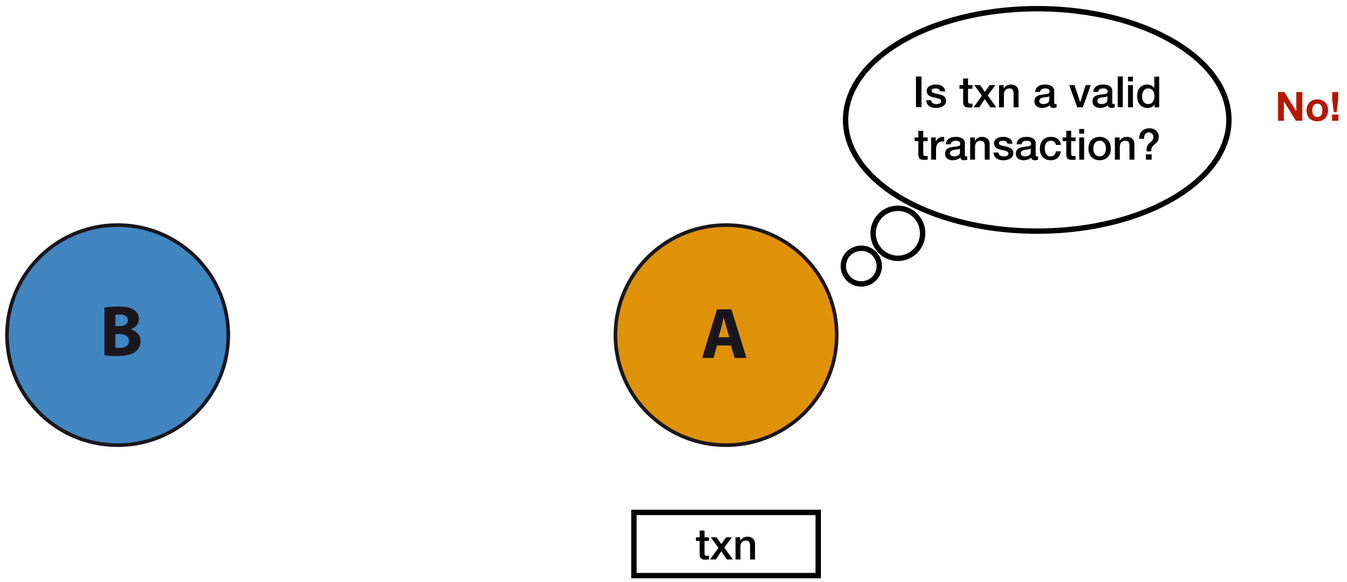
- Known transaction will be rejected

Information propagation (1/2)



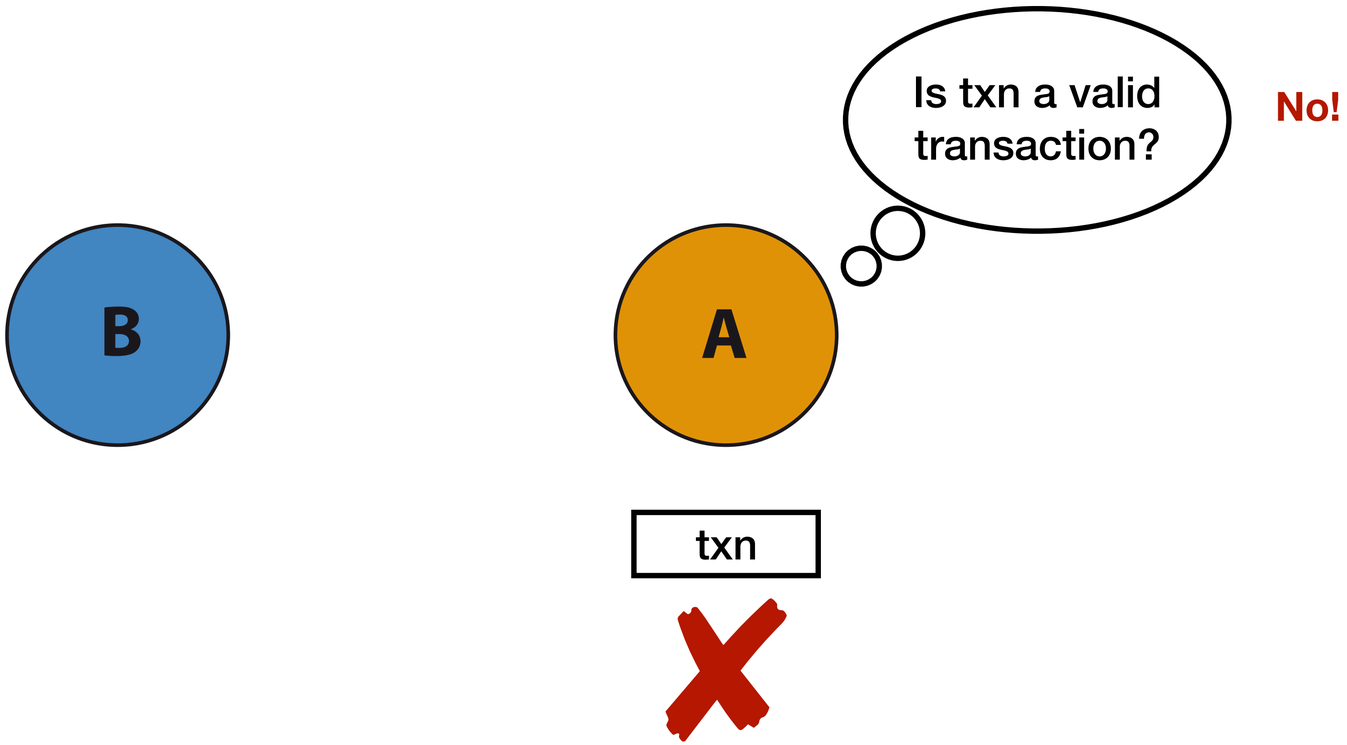
- Known transaction will be rejected

Information propagation (1/2)



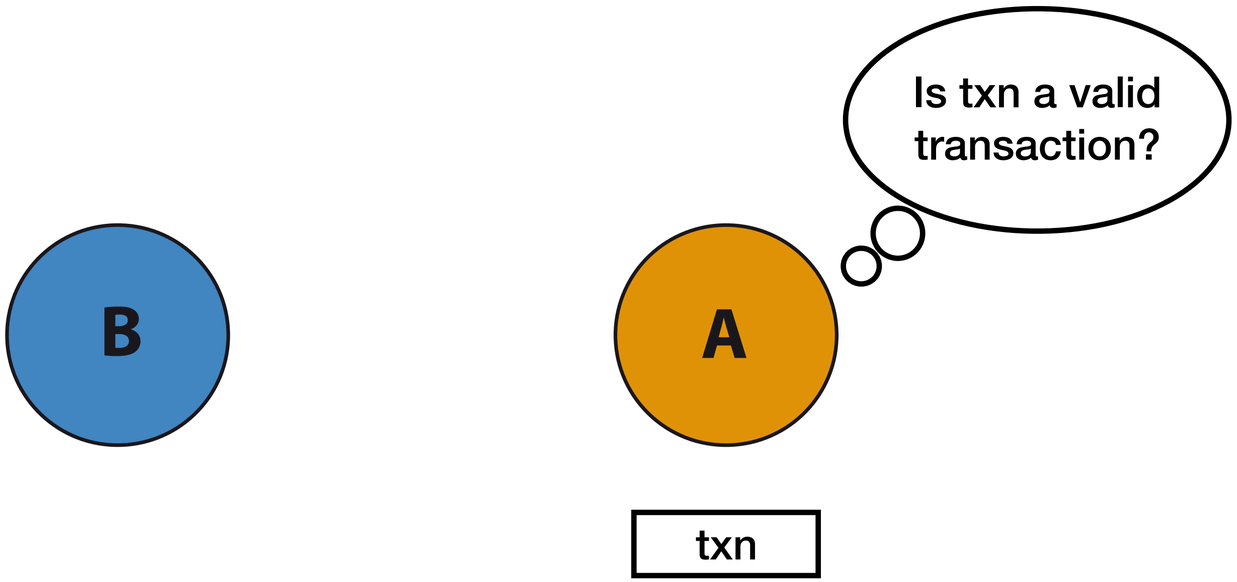
- Known transaction will be rejected

Information propagation (1/2)



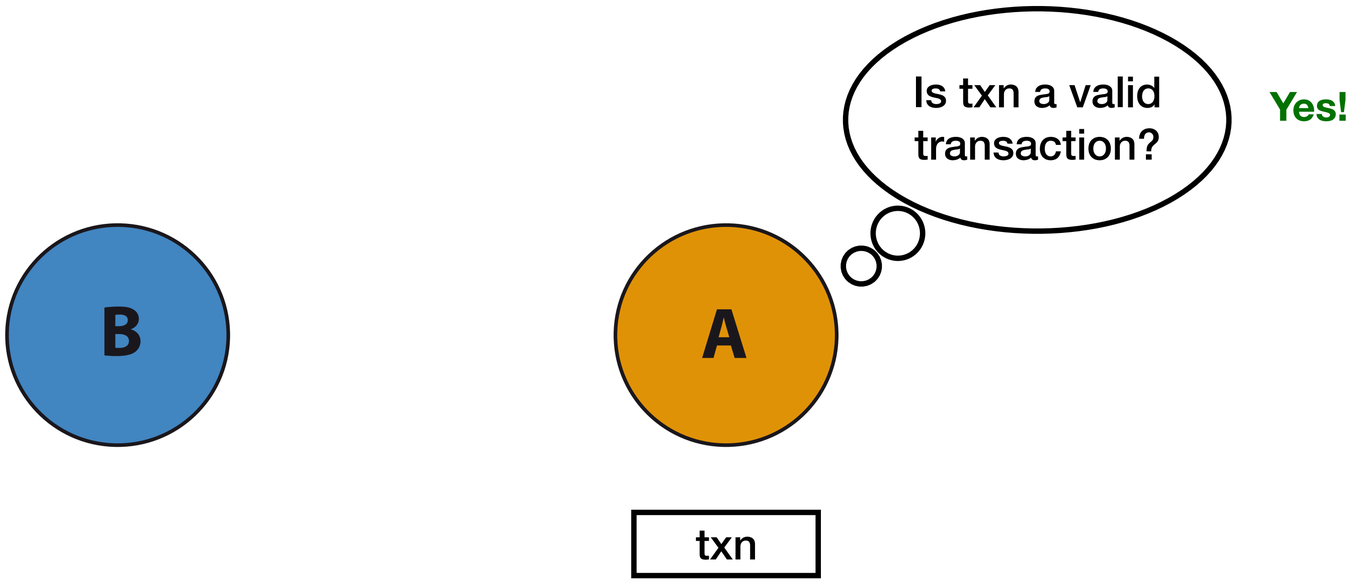
- Known transaction will be rejected
- Invalid transaction will also be rejected

Information propagation (1/2)



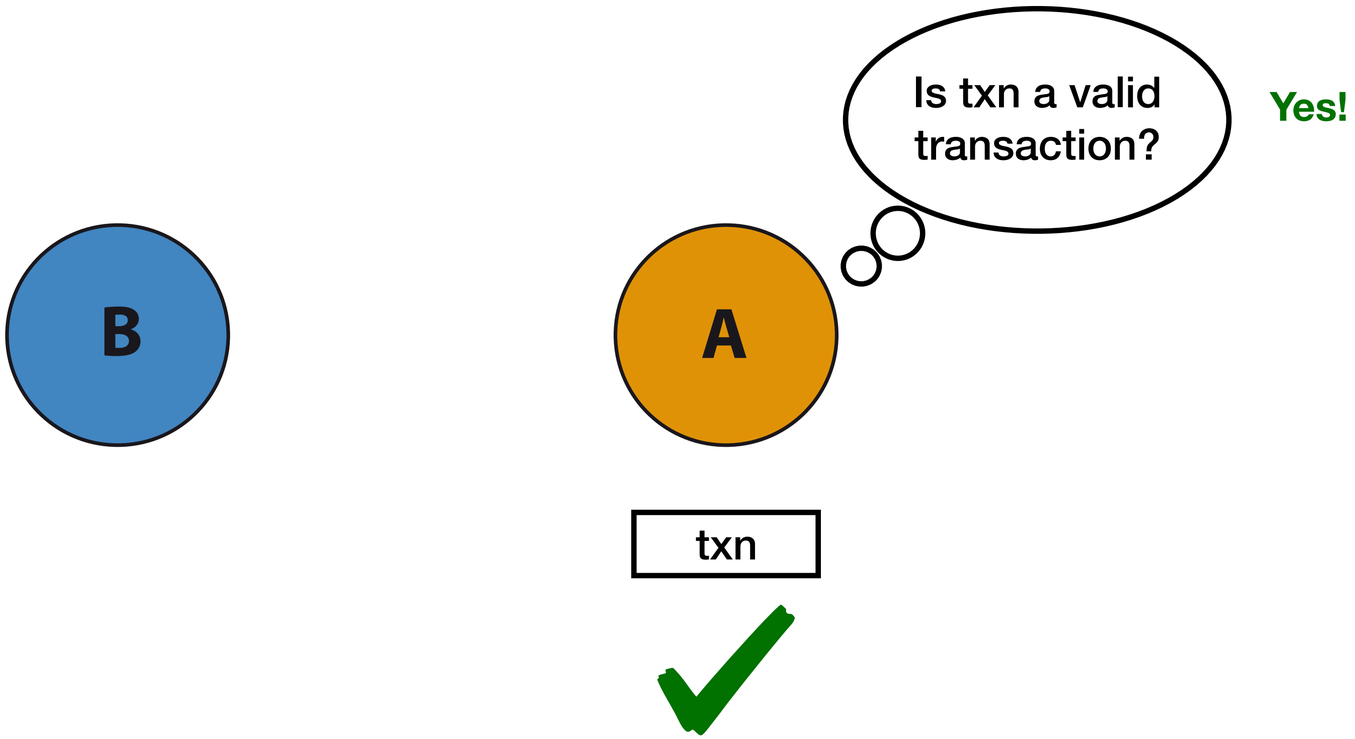
- Known transaction will be rejected
- Invalid transaction will also be rejected

Information propagation (1/2)



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

Information propagation (1/2)

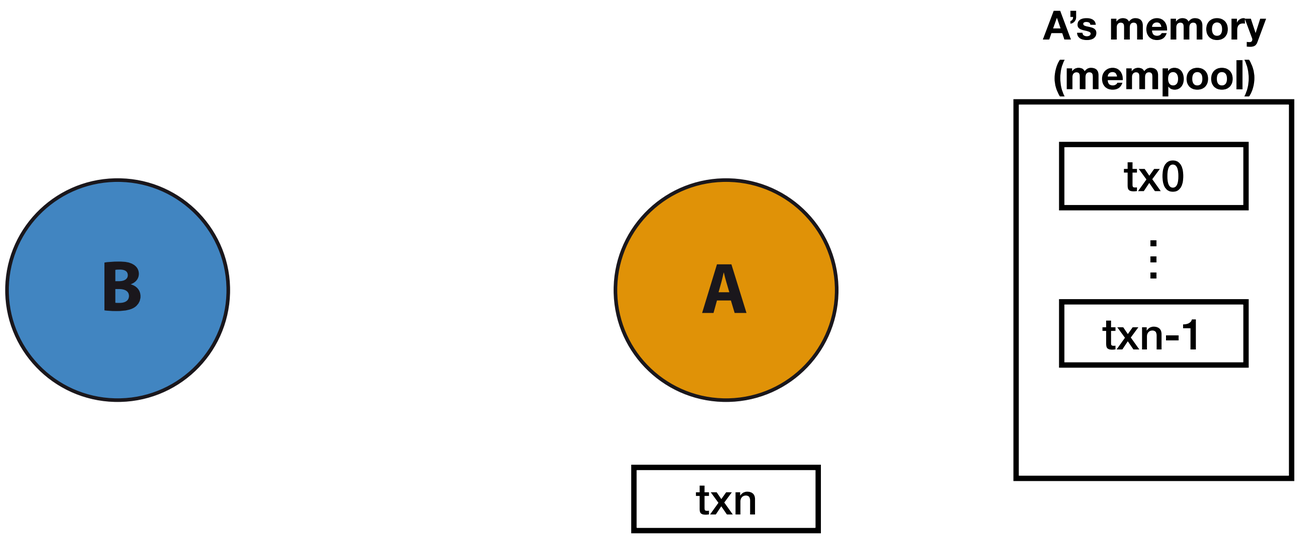


- Known transaction will be rejected
- Invalid transaction will also be rejected

Information propagation (1/2)

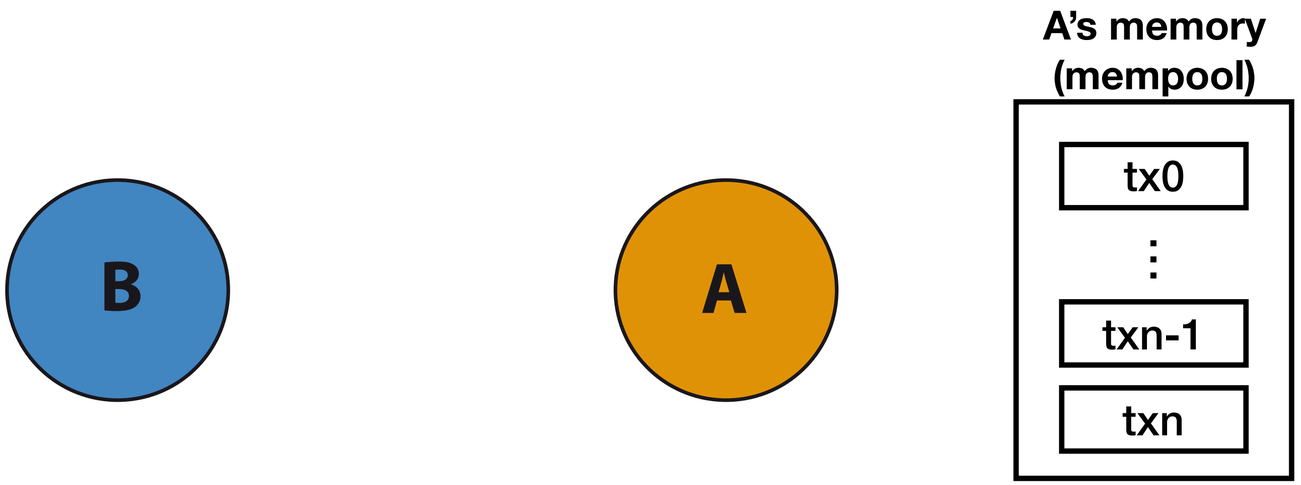
- Known transaction will be rejected
- Invalid transaction will also be rejected

Information propagation (1/2)



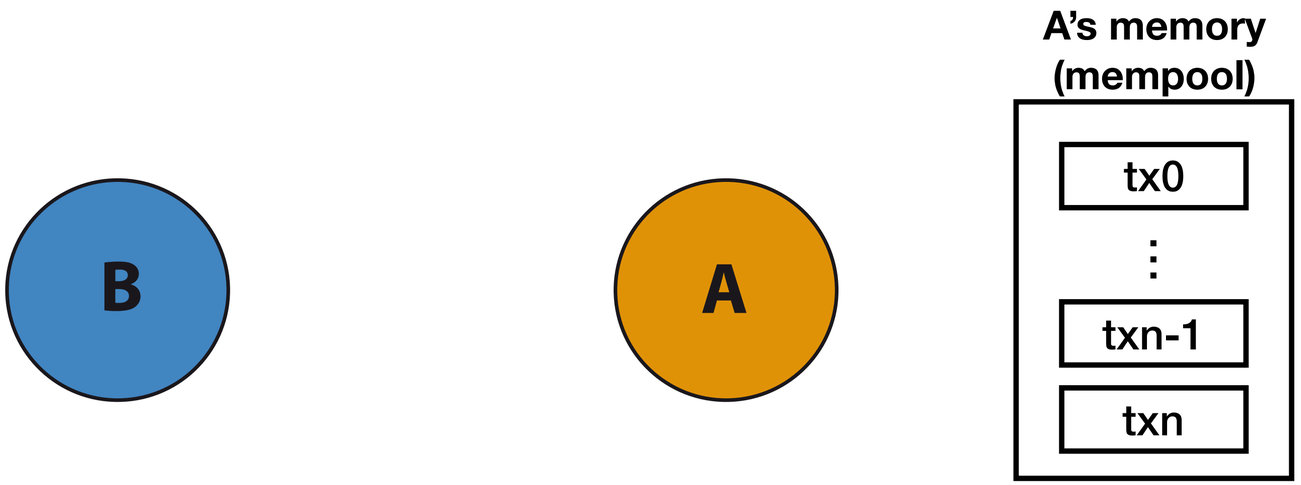
- Known transaction will be rejected
- Invalid transaction will also be rejected

Information propagation (1/2)



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

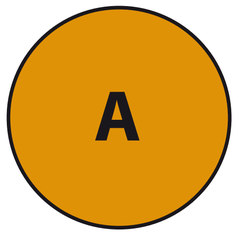
Information propagation (1/2)



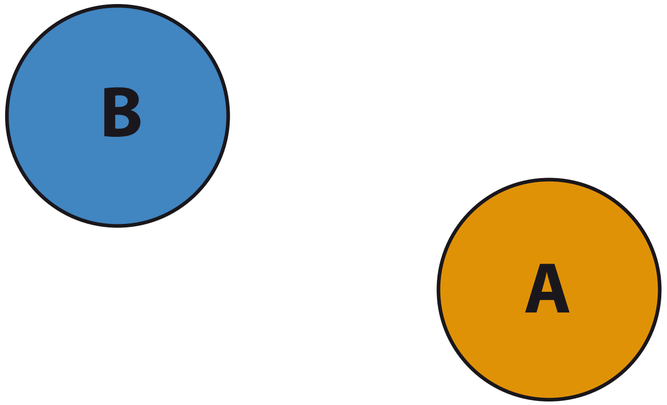
- Known transaction will be rejected
- Invalid transaction will also be rejected
- Valid (new) transactions will be kept in memory (mempool)

Information propagation (2/2)

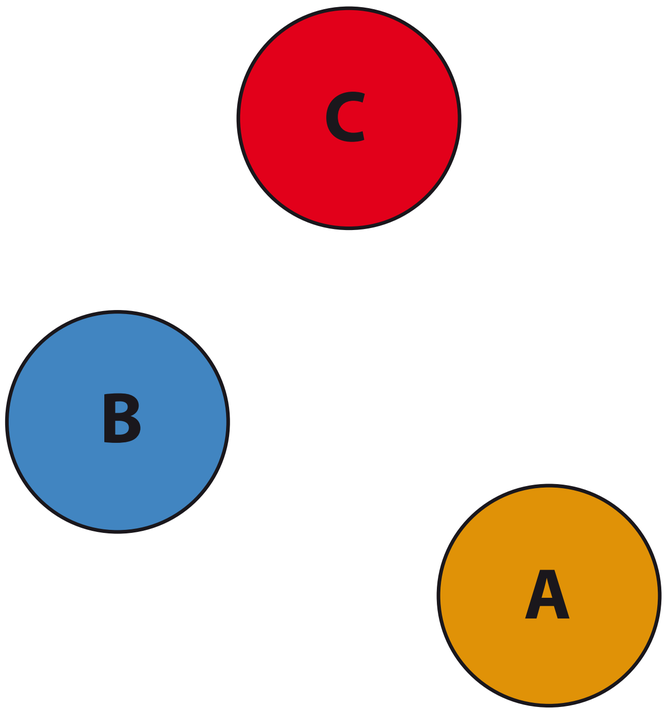
Information propagation (2/2)



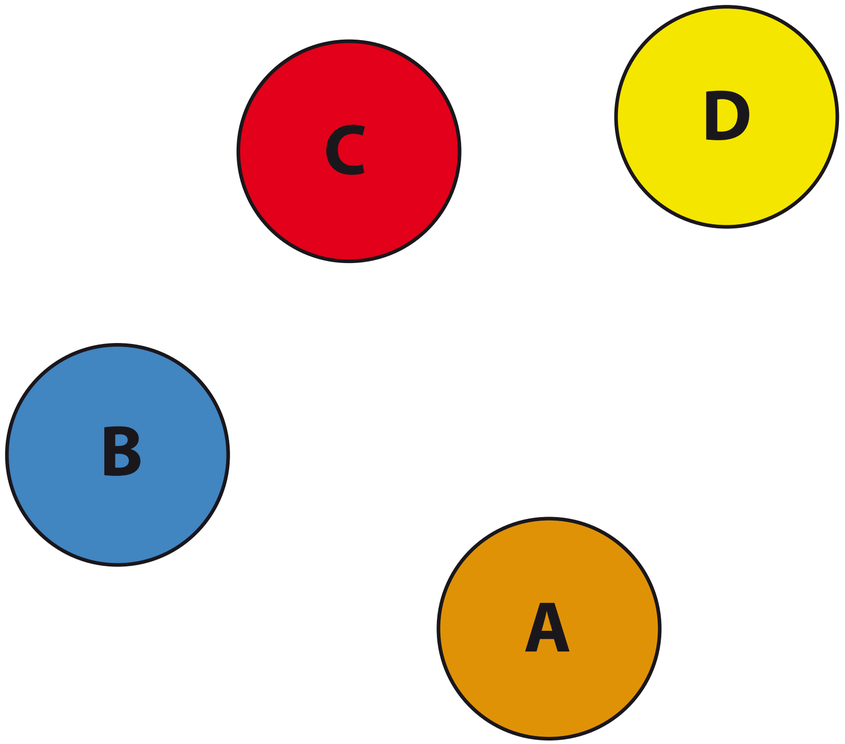
Information propagation (2/2)



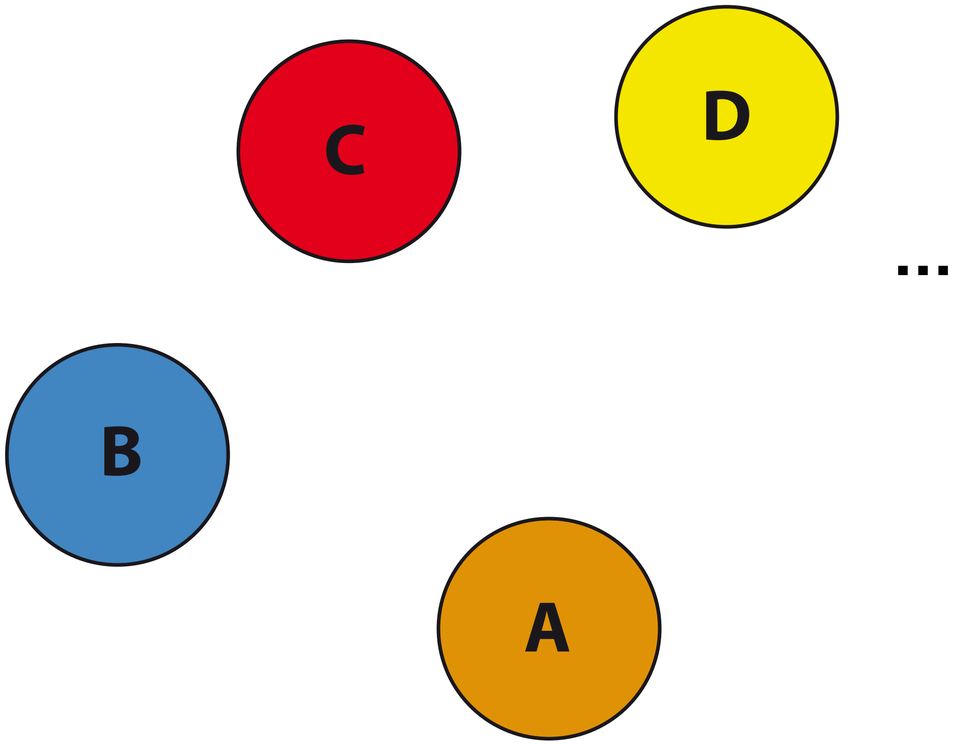
Information propagation (2/2)



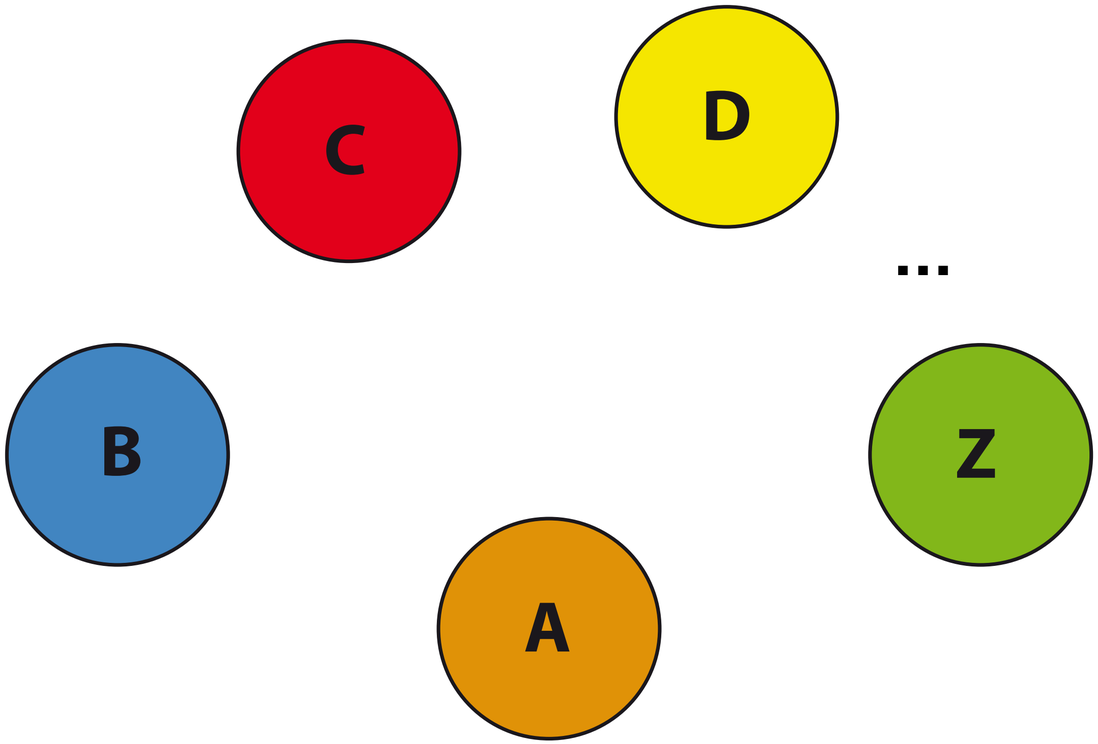
Information propagation (2/2)



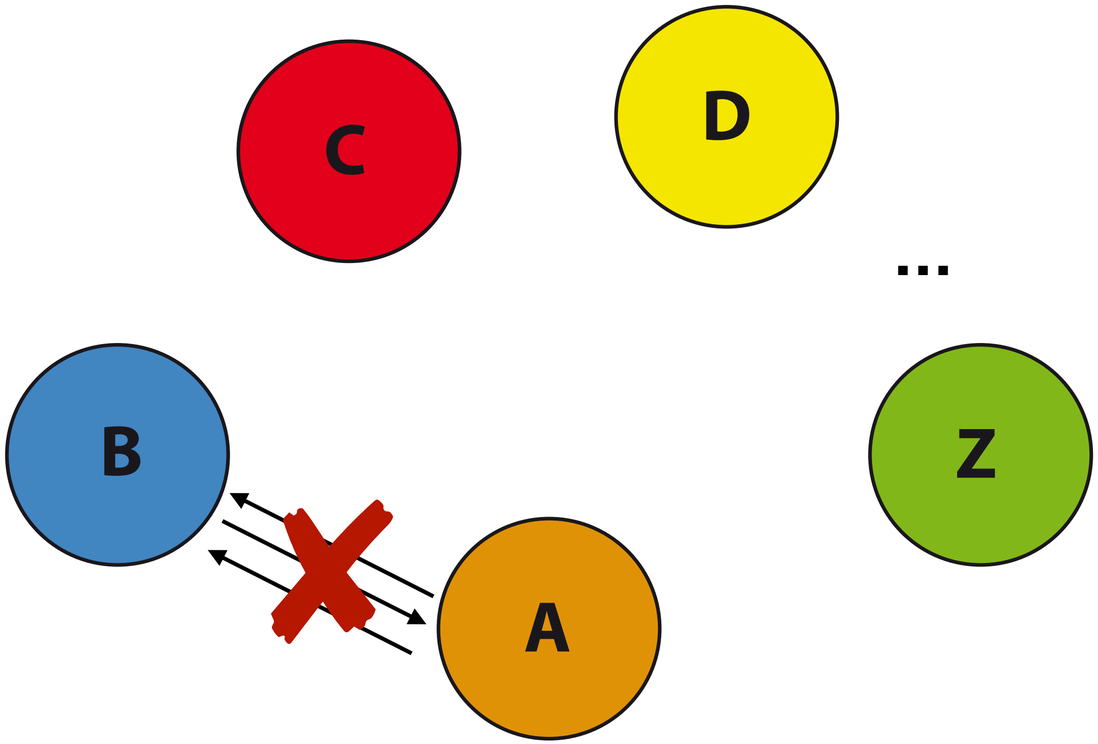
Information propagation (2/2)



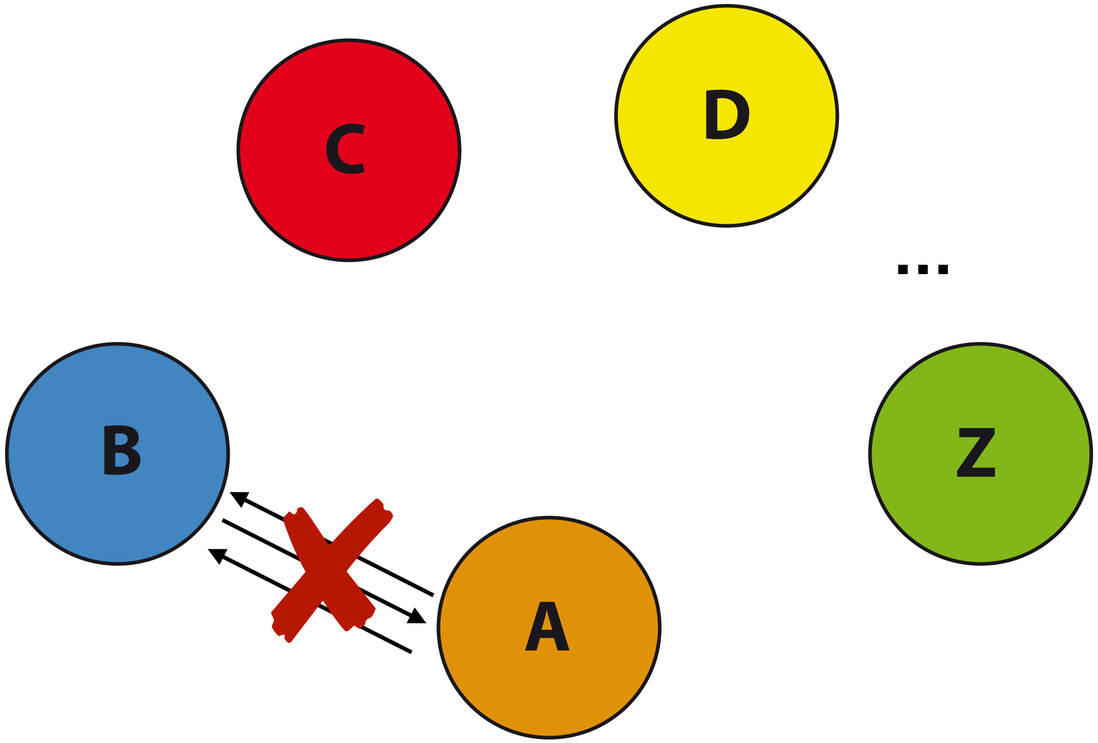
Information propagation (2/2)

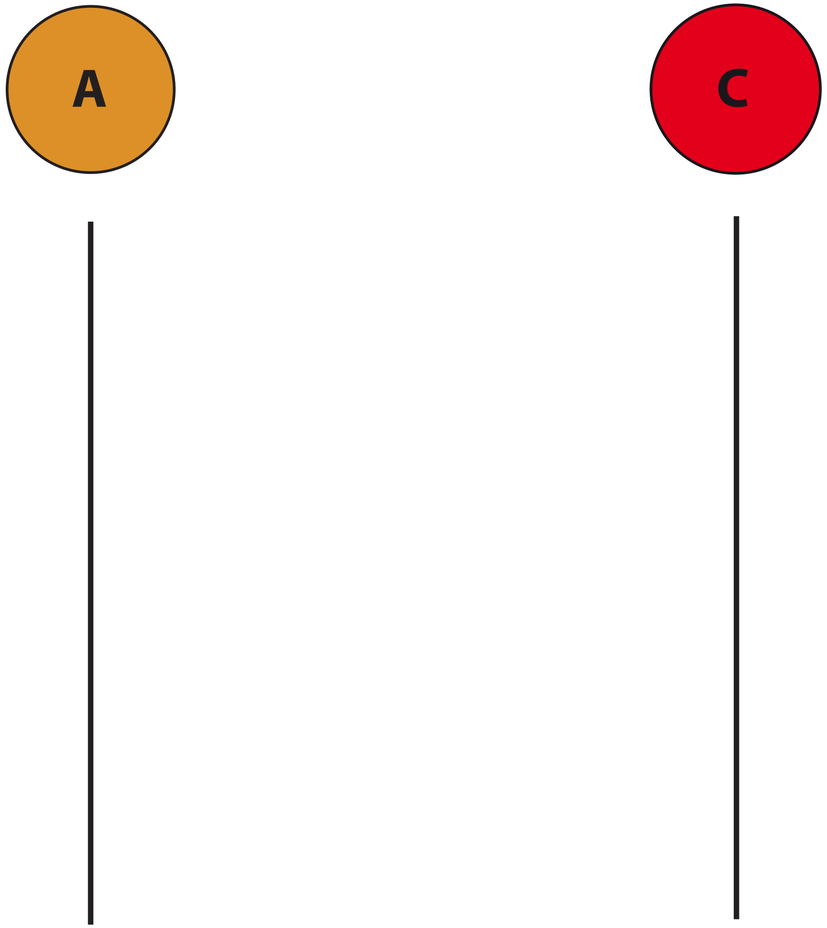


Information propagation (2/2)

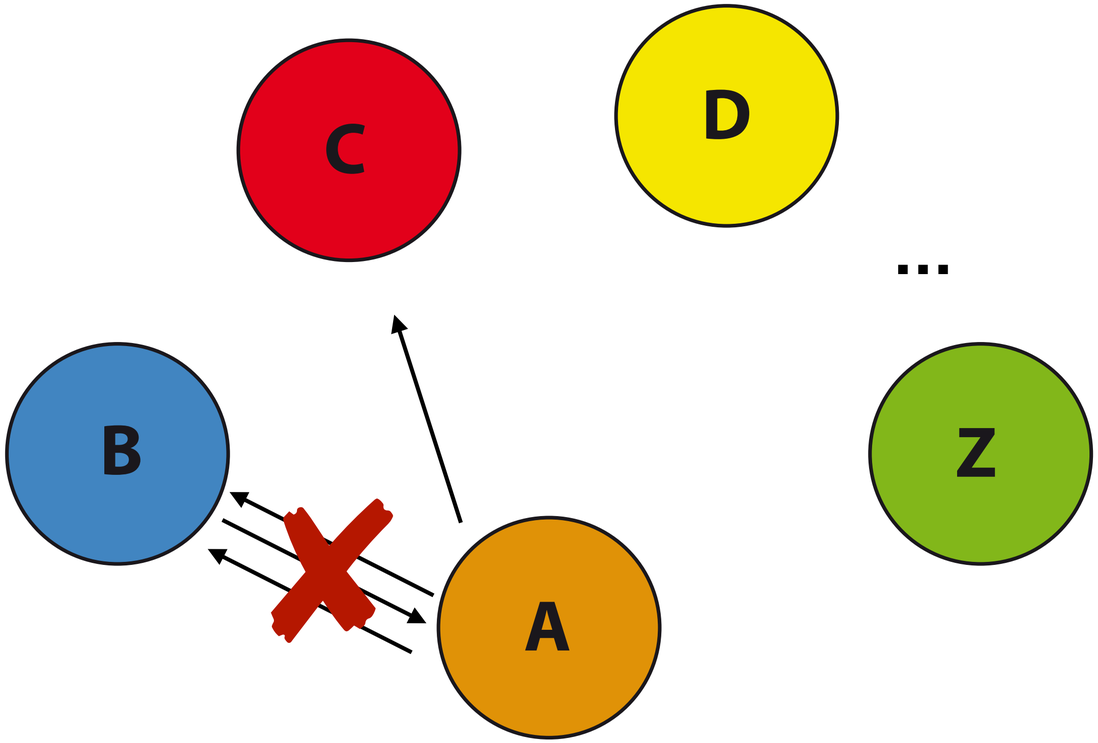


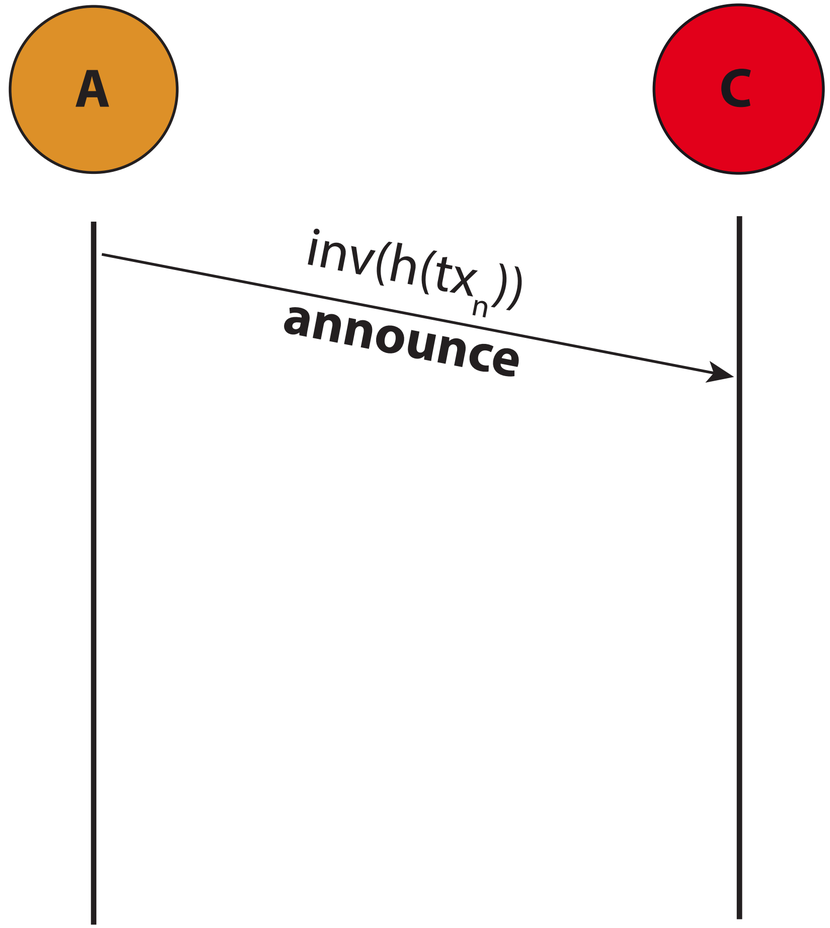
Information propagation (2/2)



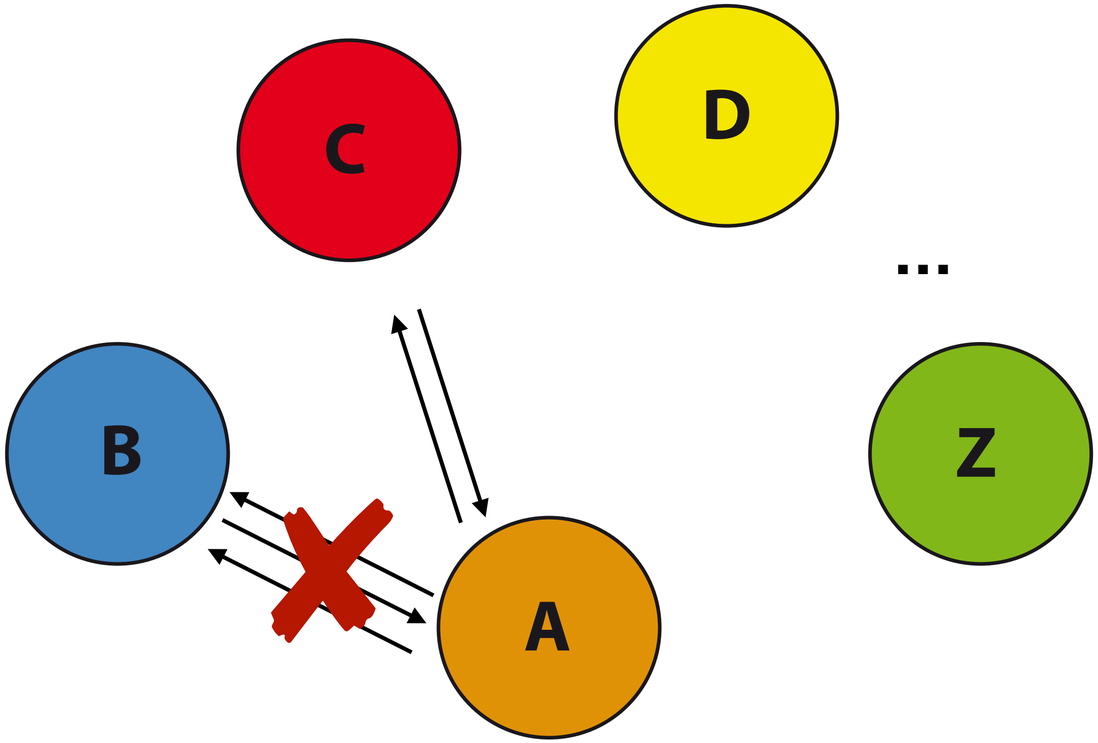


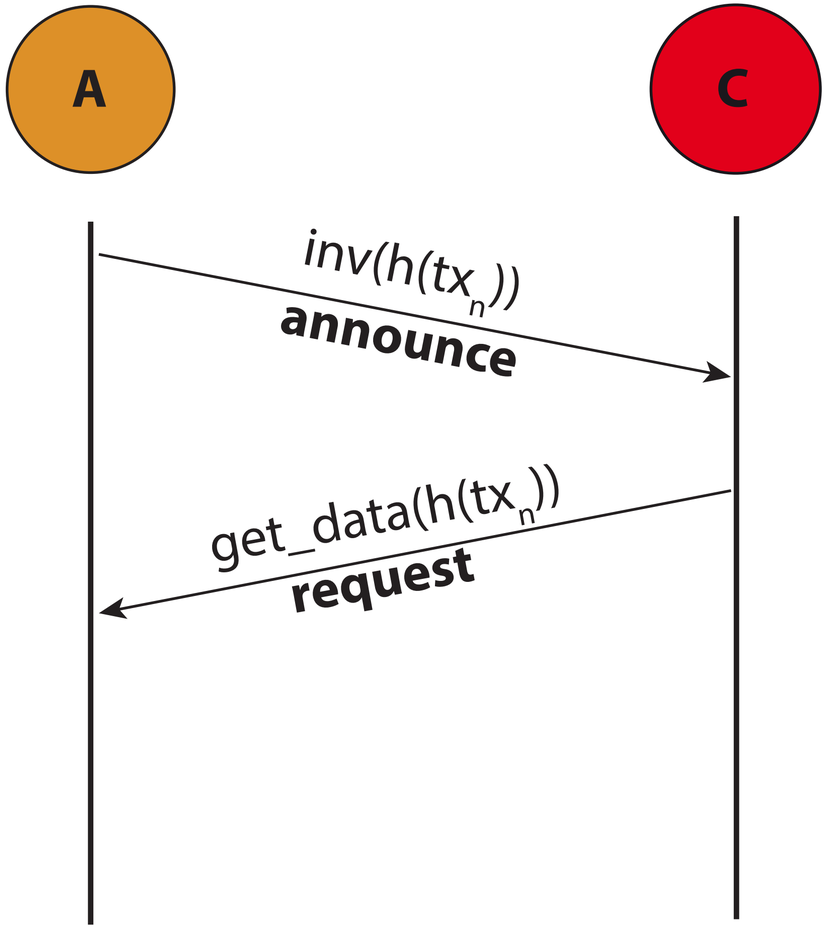
Information propagation (2/2)



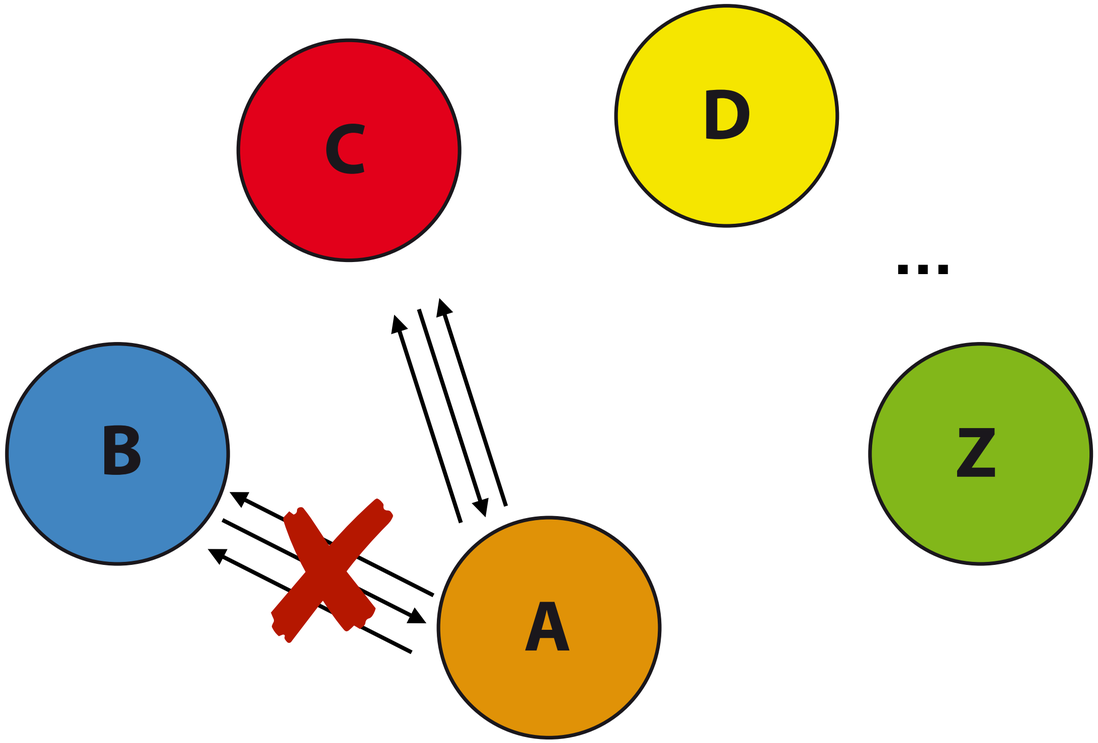


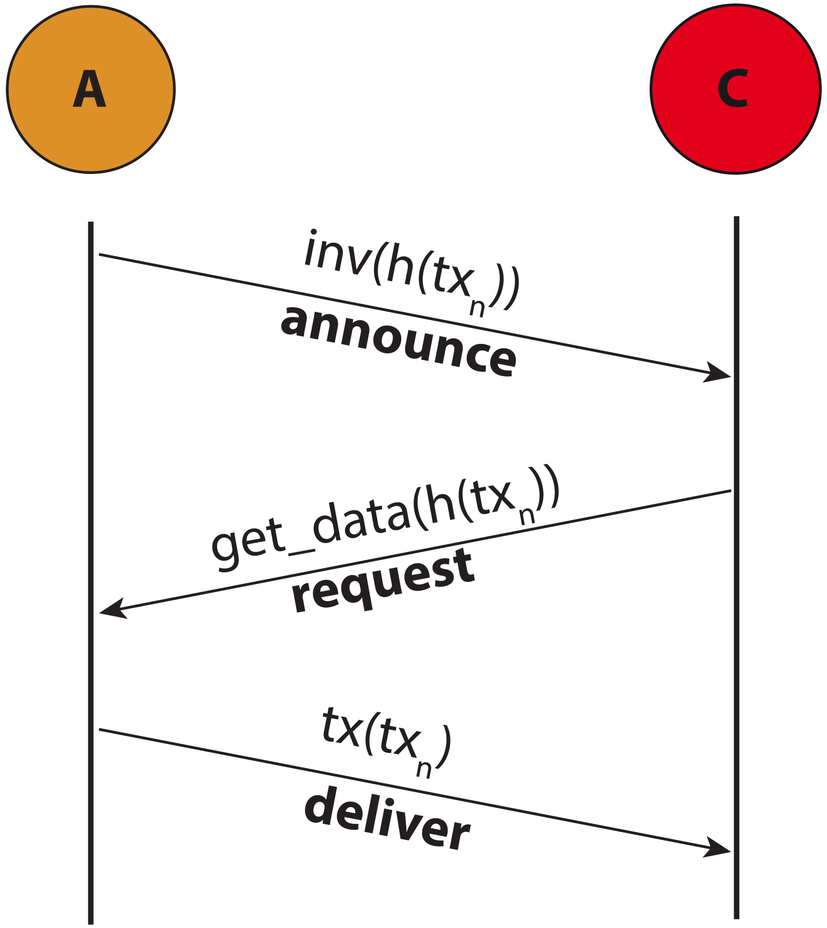
Information propagation (2/2)



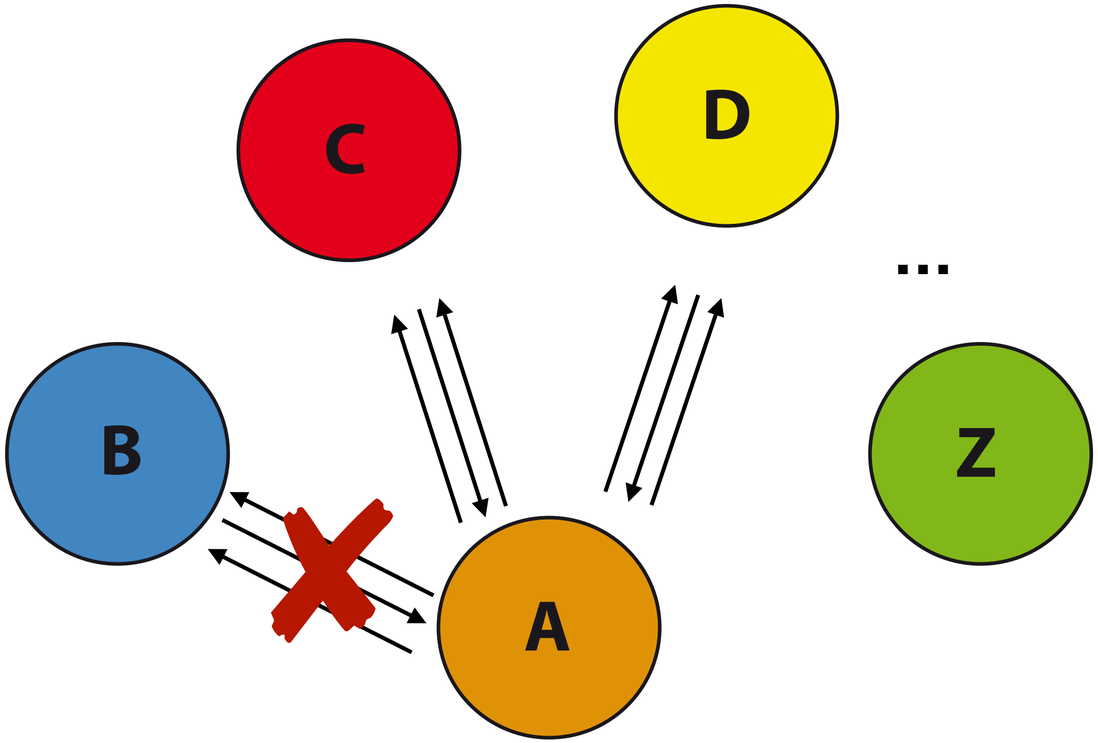


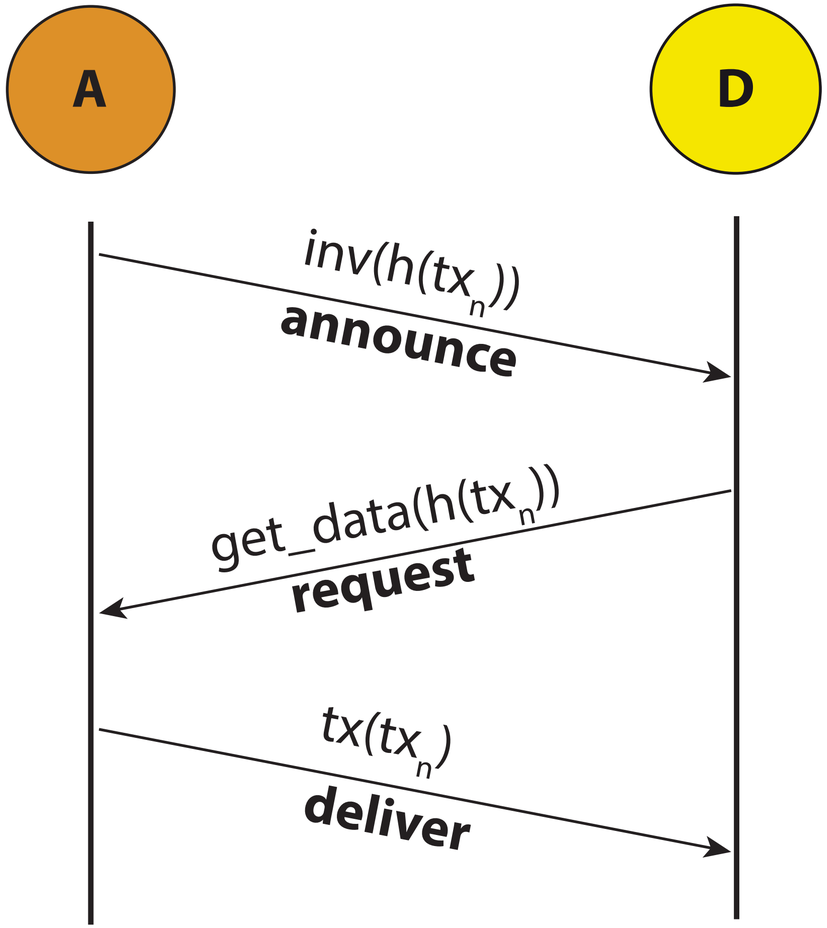
Information propagation (2/2)



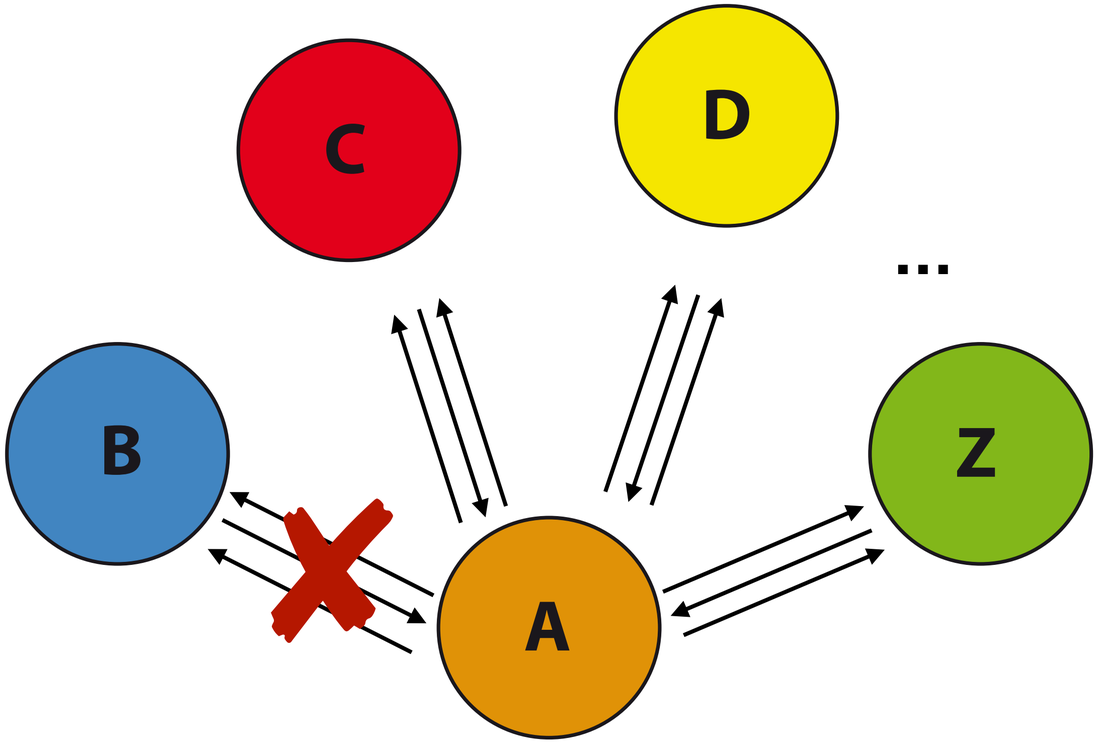


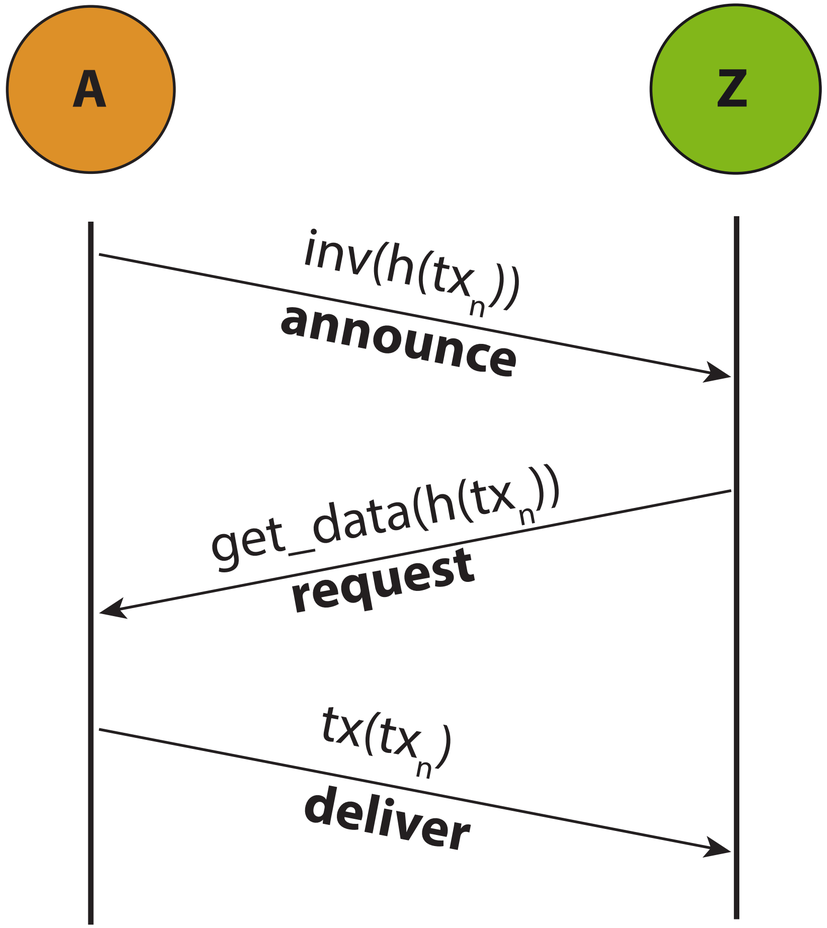
Information propagation (2/2)



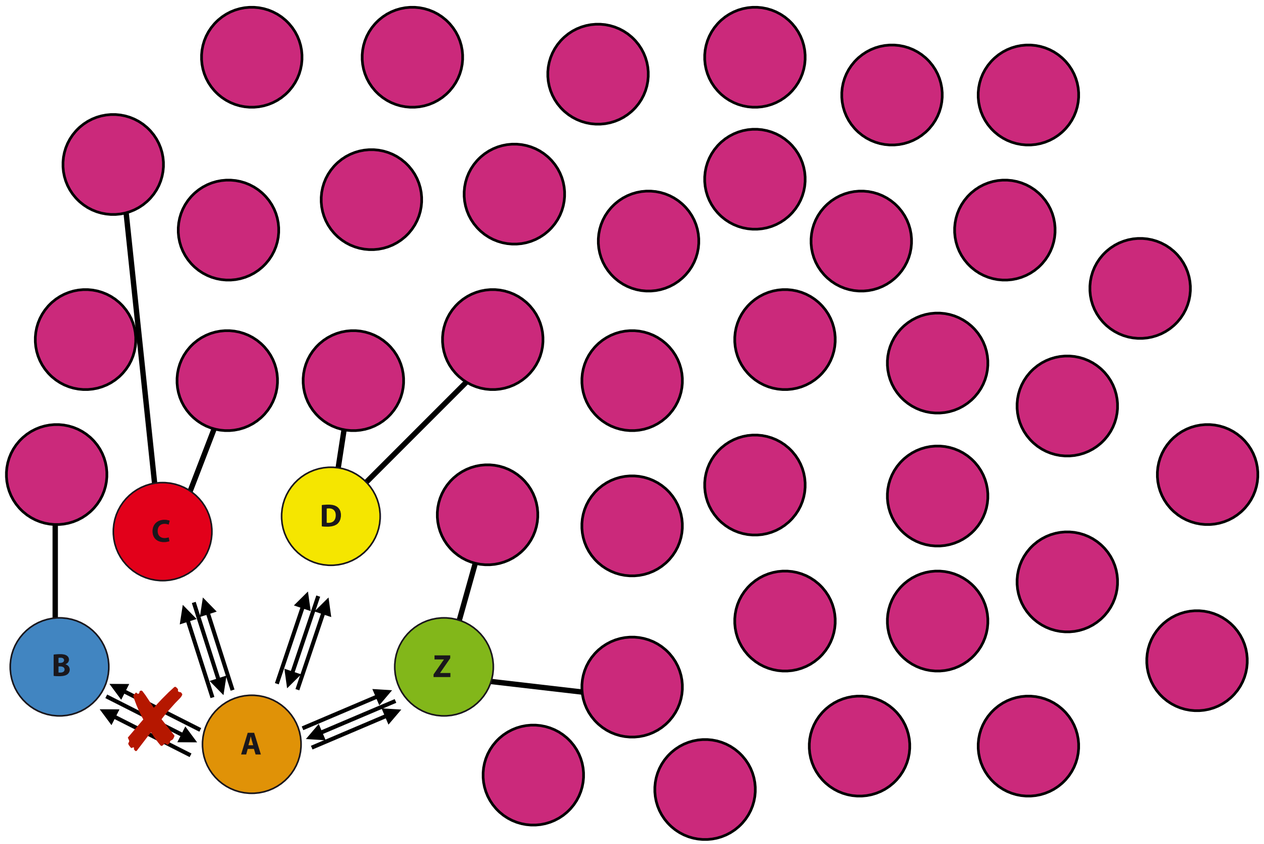


Information propagation (2/2)

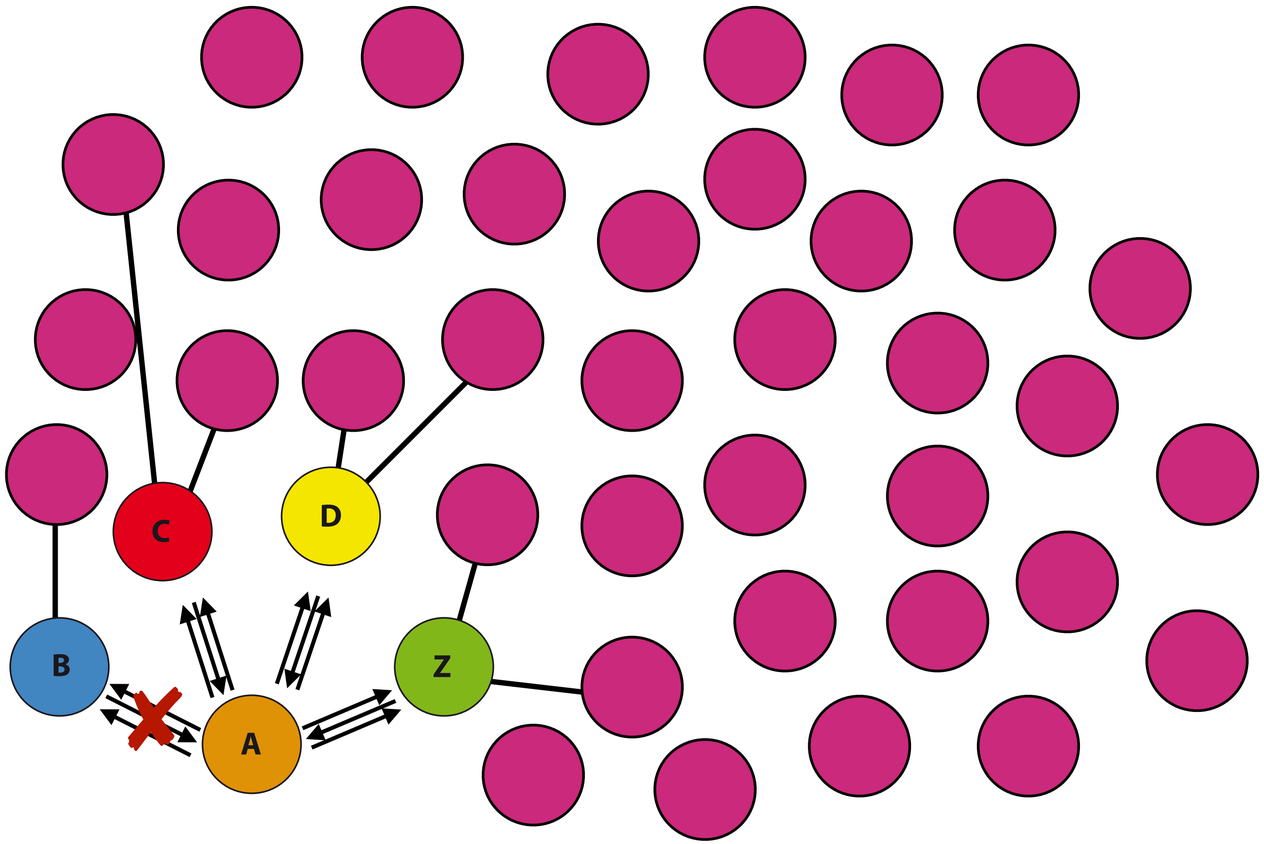




Information propagation (2/2)

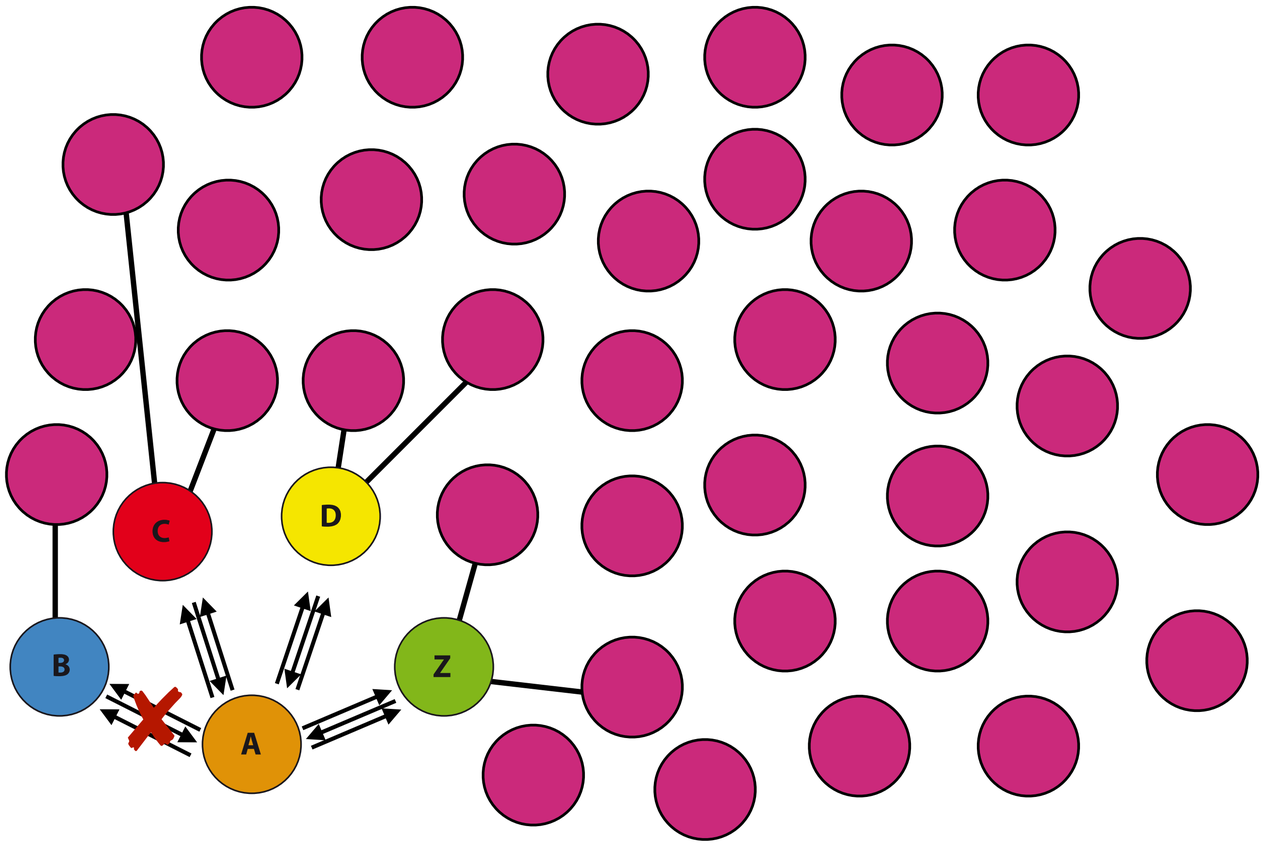


Information propagation (2/2)



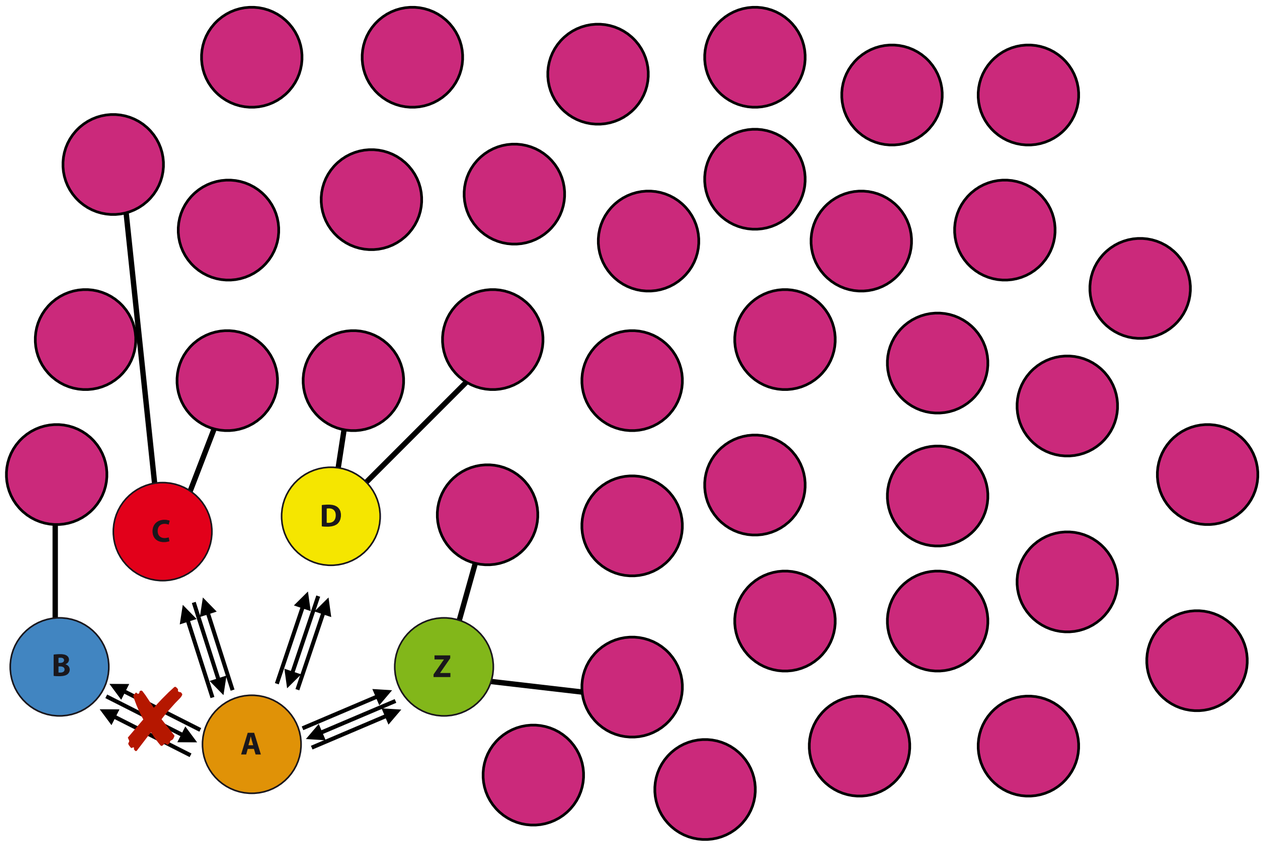
- And so on and so forth until all the nodes are reached

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*

Information propagation (propagation delays)

Information propagation (propagation delays)

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

- The transaction relay time currently follows a Poison distribution

- The propagation of blocks is not delayed, in order to reach full network coverage as soon as possible (* from previous slide)

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

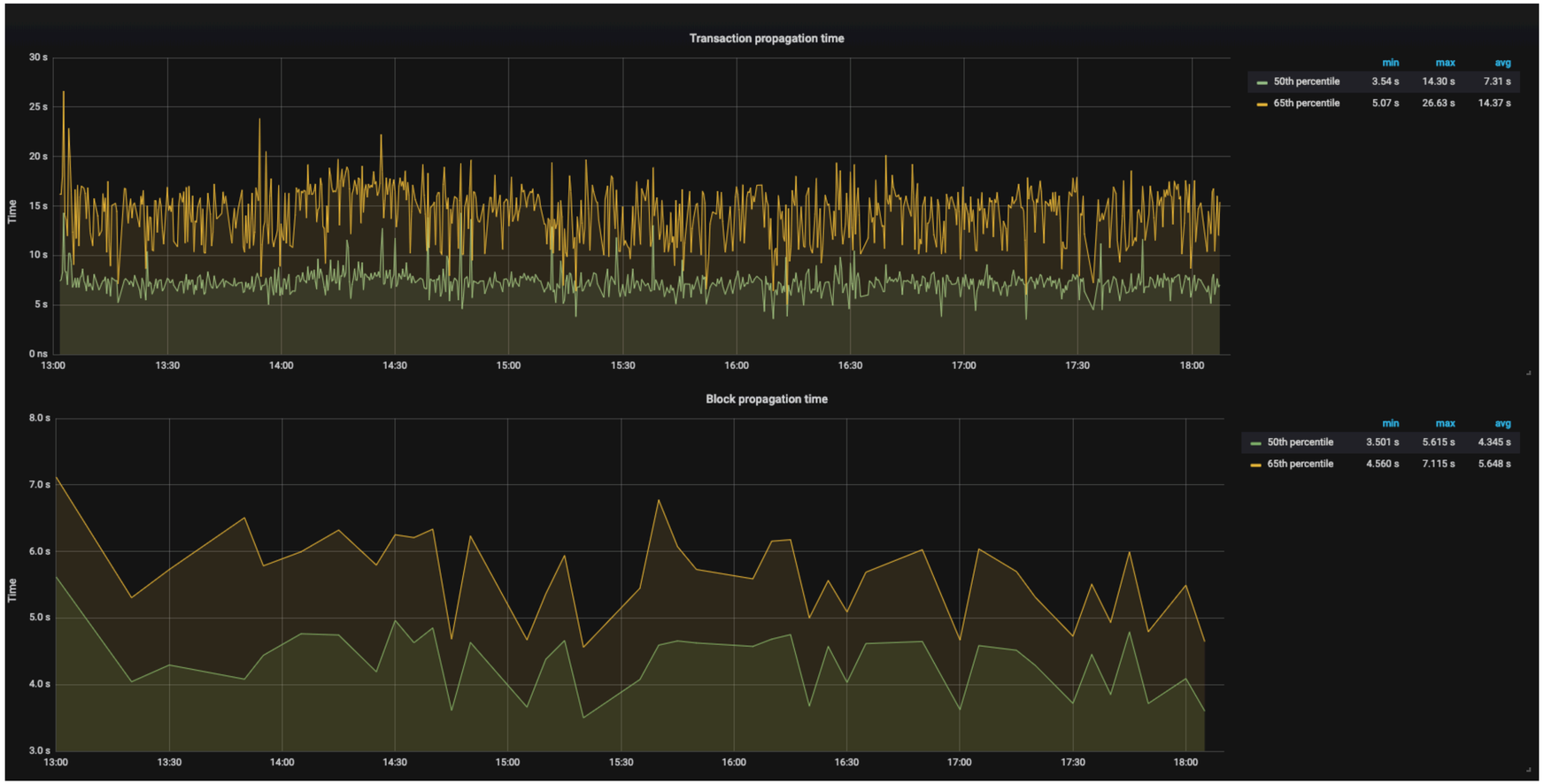
- The bigger the network the more it takes for an item to propagate (**this can be counterintuitive**)

- The bigger an item, the more it takes to validate and propagate

- Long propagation times (**for blocks**) imply bigger fork rate

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

Bitcoin testnet data propagation times



source: charts.satoshi.uab.cat

Title

Title

- Slide about FALCON, FIBRE, etc to talk about why block propagation is faster

0-conf transactions and double-spending

0-confirmation transactions (1/2)

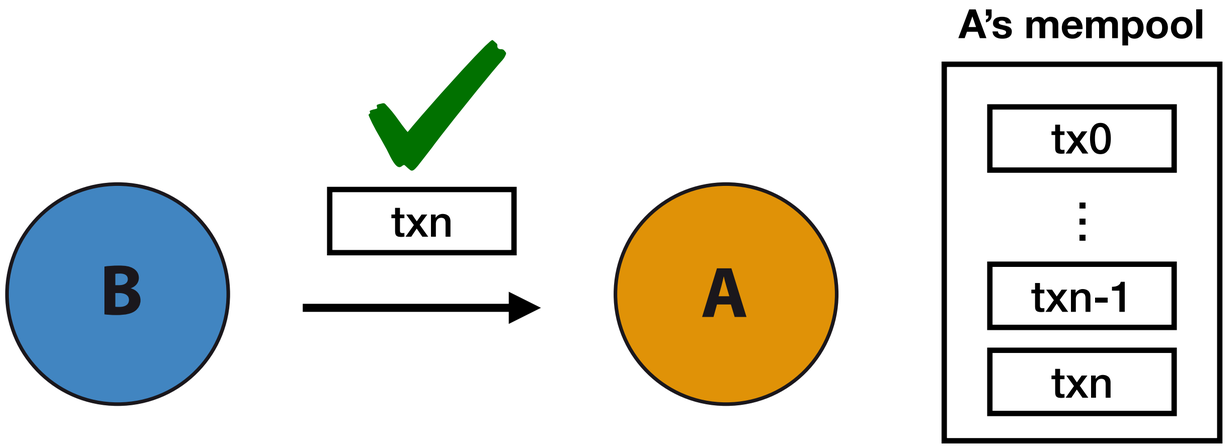
- **0-conf** transactions / **unconfirmed** transactions are those that are not part of the blockchain

- They can be valid transactions that are waiting to be included

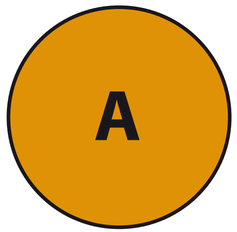
- But they can also be invalid (e.g: **double-spending transactions, etc**)

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

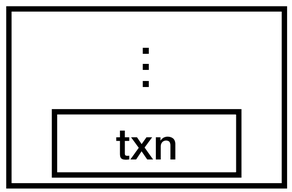
0-confirmation transactions (2/2)



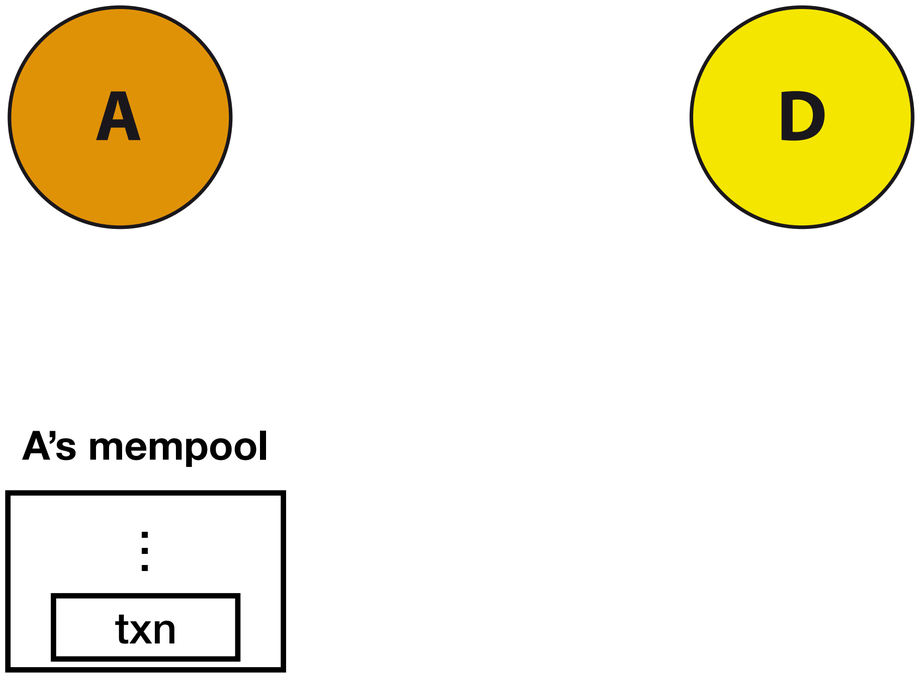
Confirmed transactions



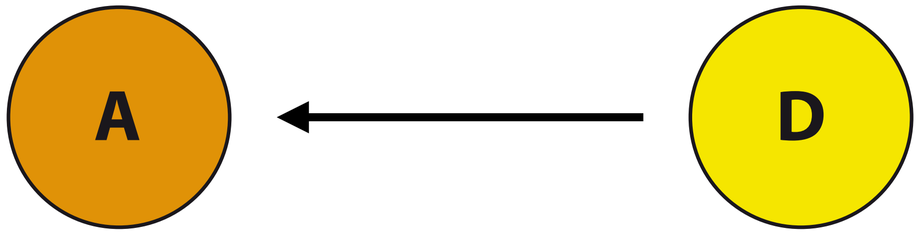
A's mempool



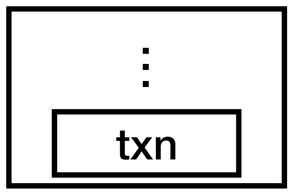
Confirmed transactions



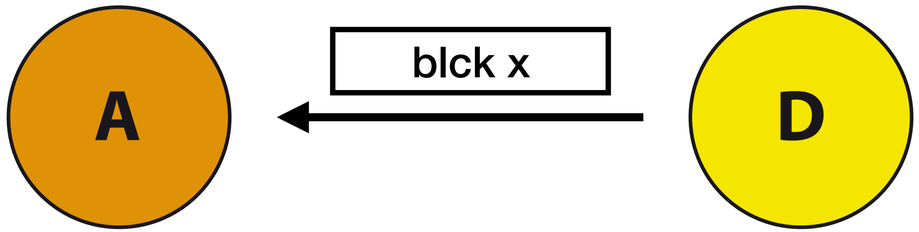
Confirmed transactions



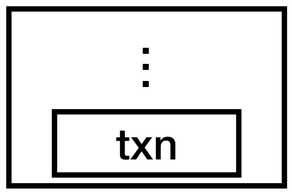
A's mempool



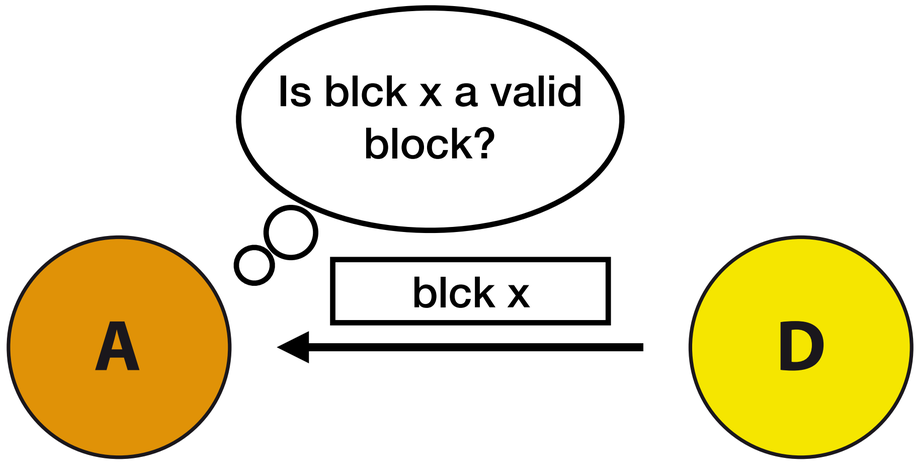
Confirmed transactions



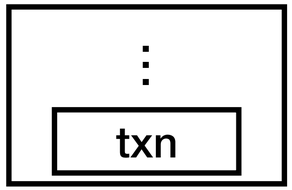
A's mempool



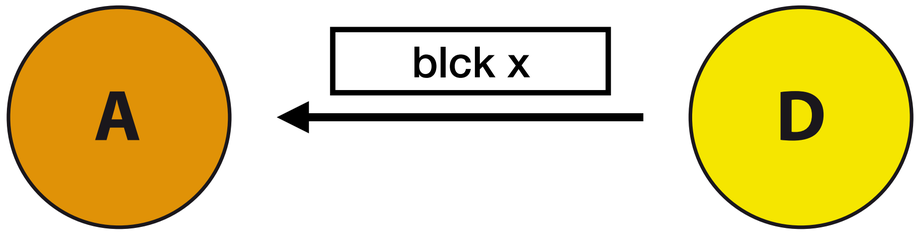
Confirmed transactions



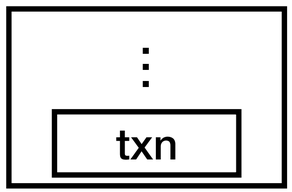
A's mempool



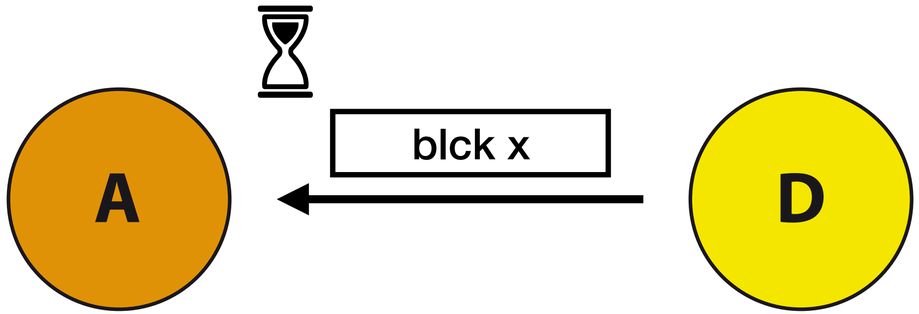
Confirmed transactions



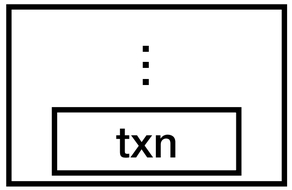
A's mempool



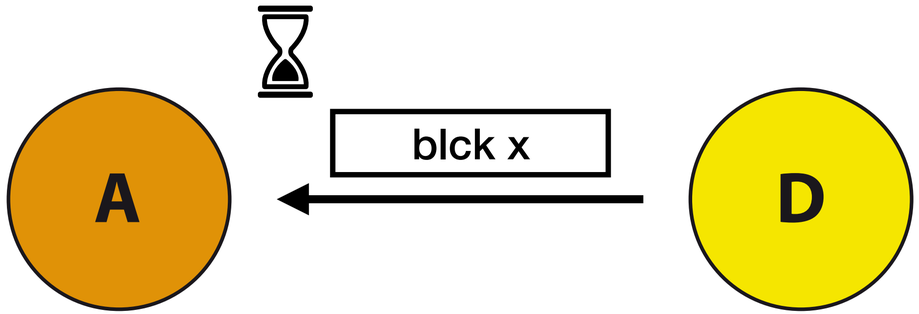
Confirmed transactions



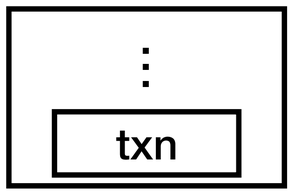
A's mempool



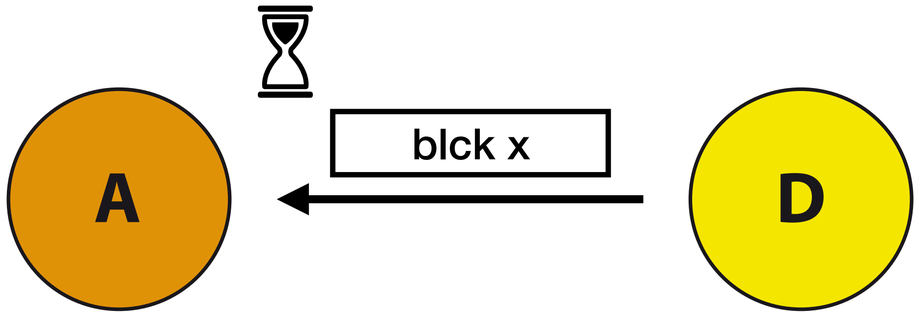
Confirmed transactions



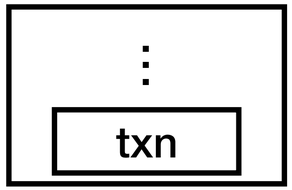
A's mempool



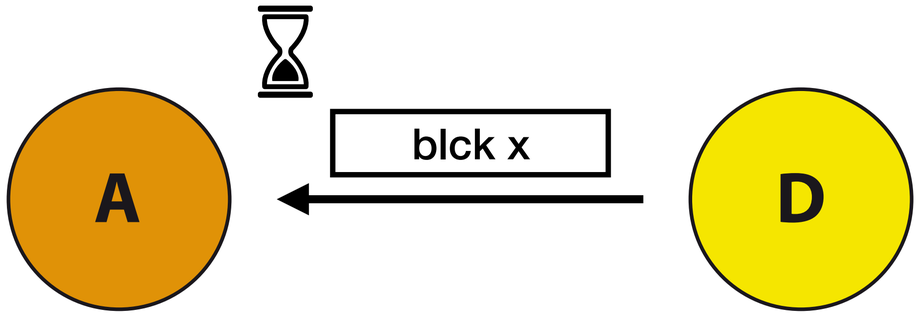
Confirmed transactions



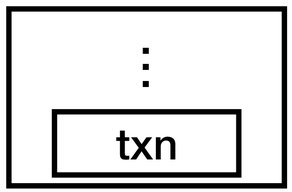
A's mempool



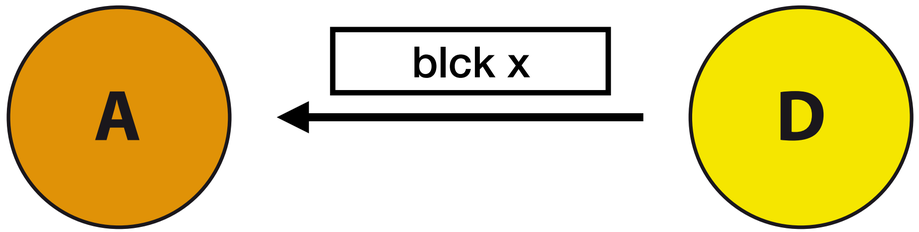
Confirmed transactions



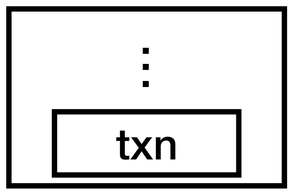
A's mempool



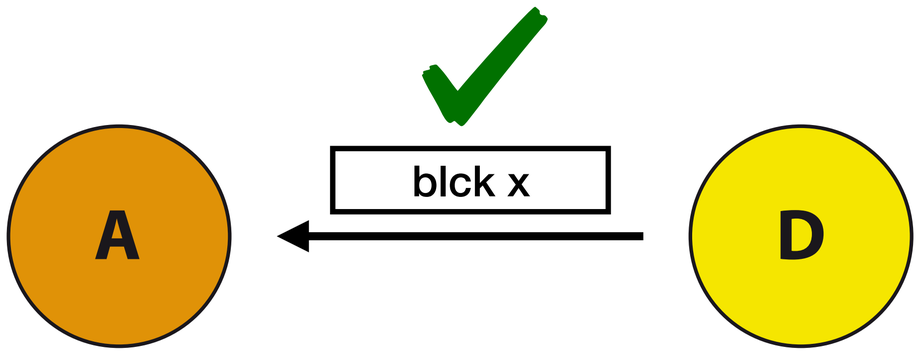
Confirmed transactions



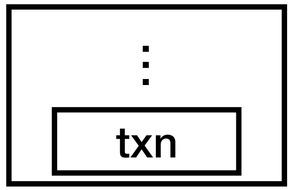
A's mempool



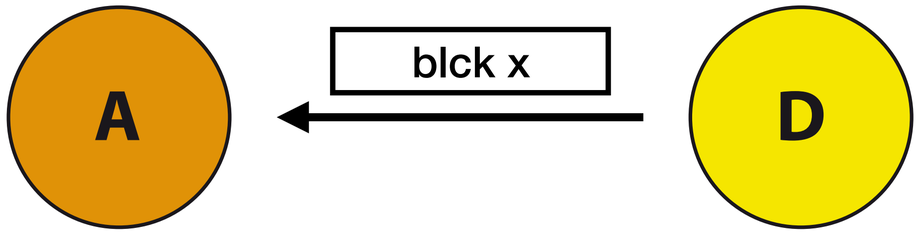
Confirmed transactions



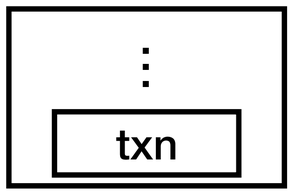
A's mempool



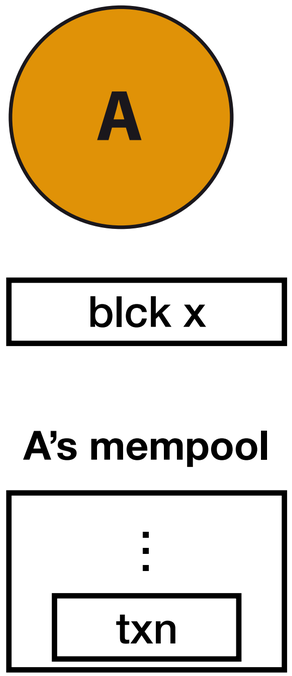
Confirmed transactions



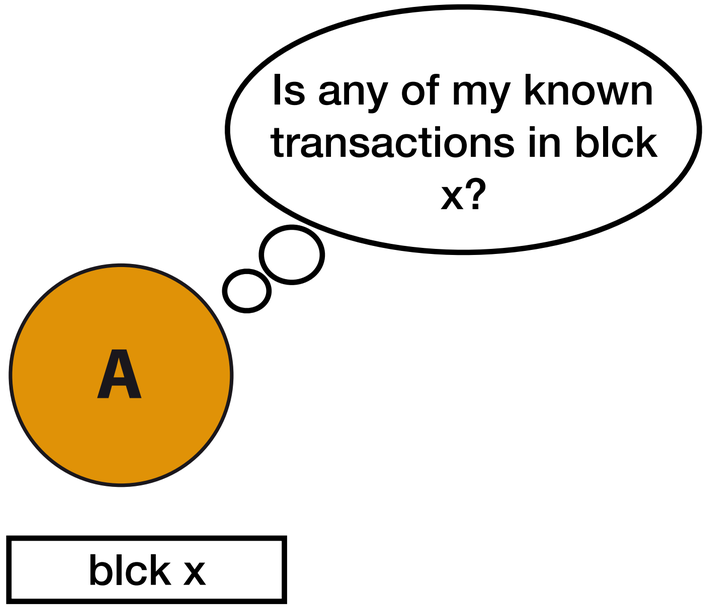
A's mempool



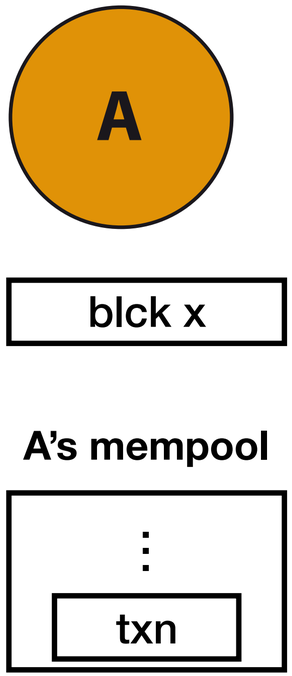
Confirmed transactions



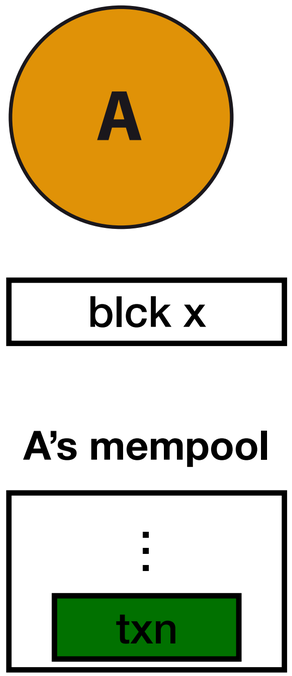
Confirmed transactions



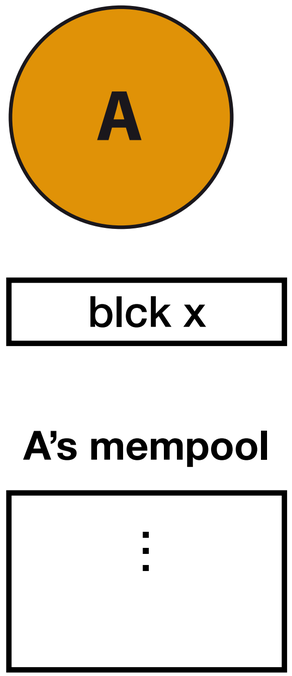
Confirmed transactions



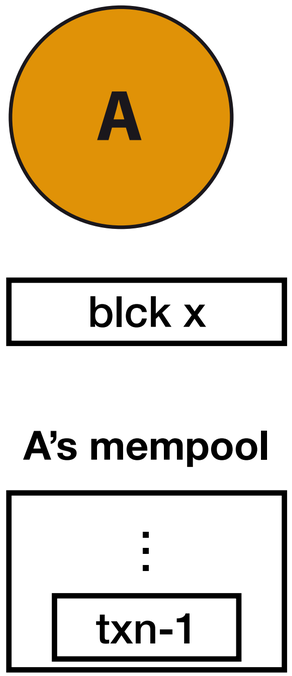
Confirmed transactions



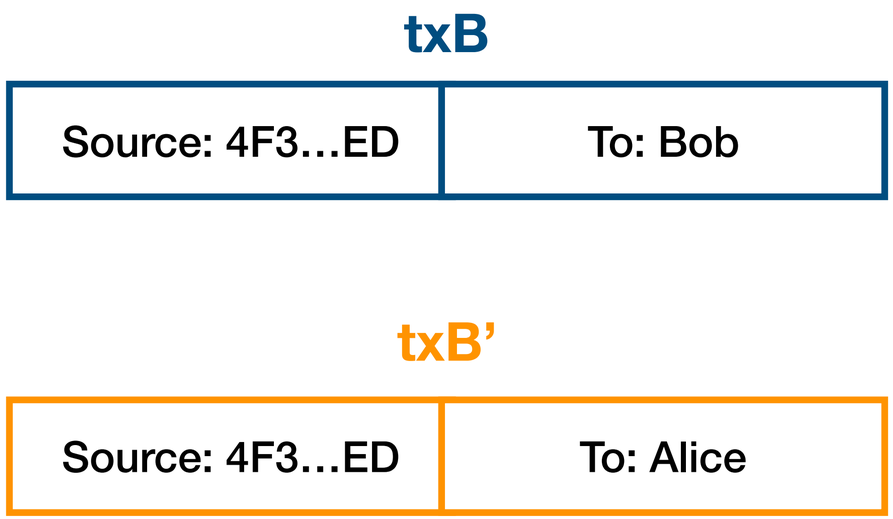
Confirmed transactions



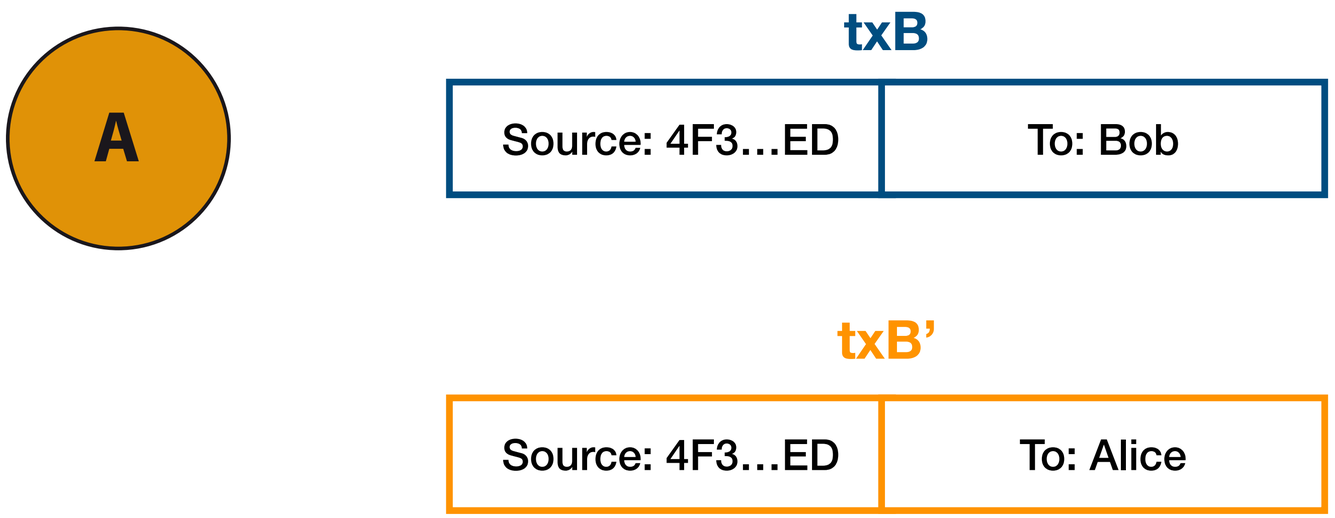
Confirmed transactions



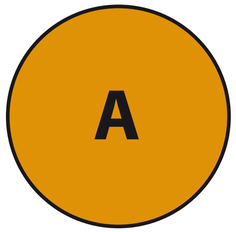
Double-spending transactions (1/2)



Double-spending transactions (1/2)



Double-spending transactions (1/2)



id = 4F3…ED

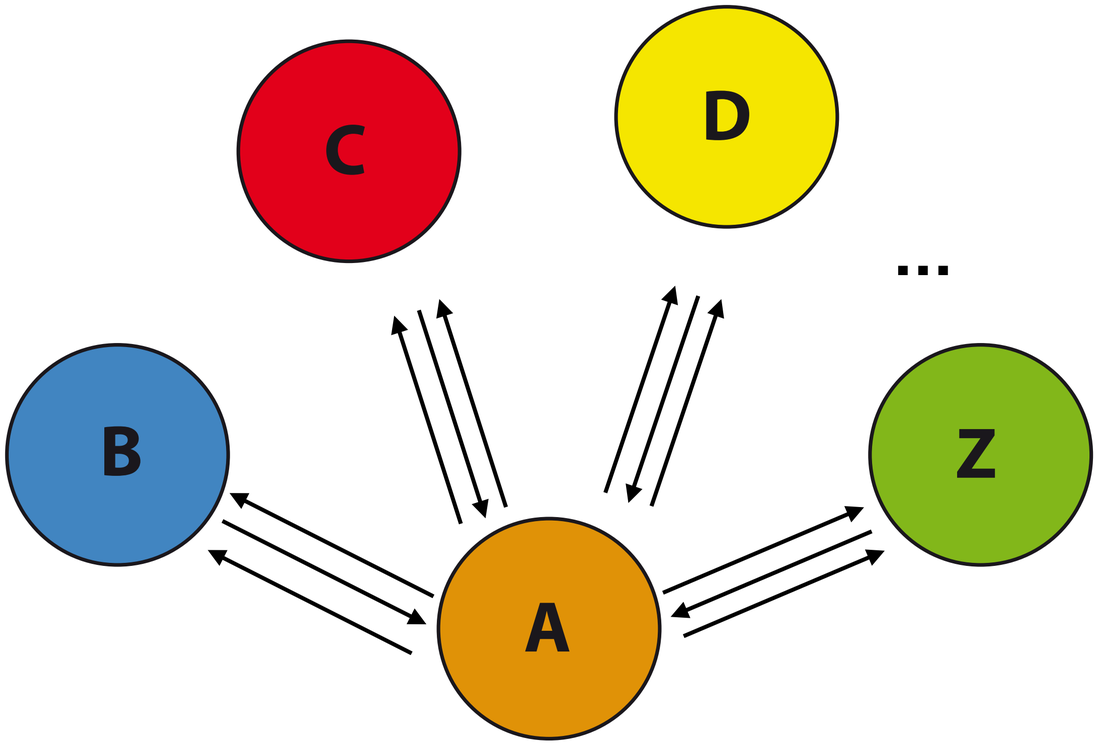
txB

Source: 4F3…ED	To: Bob

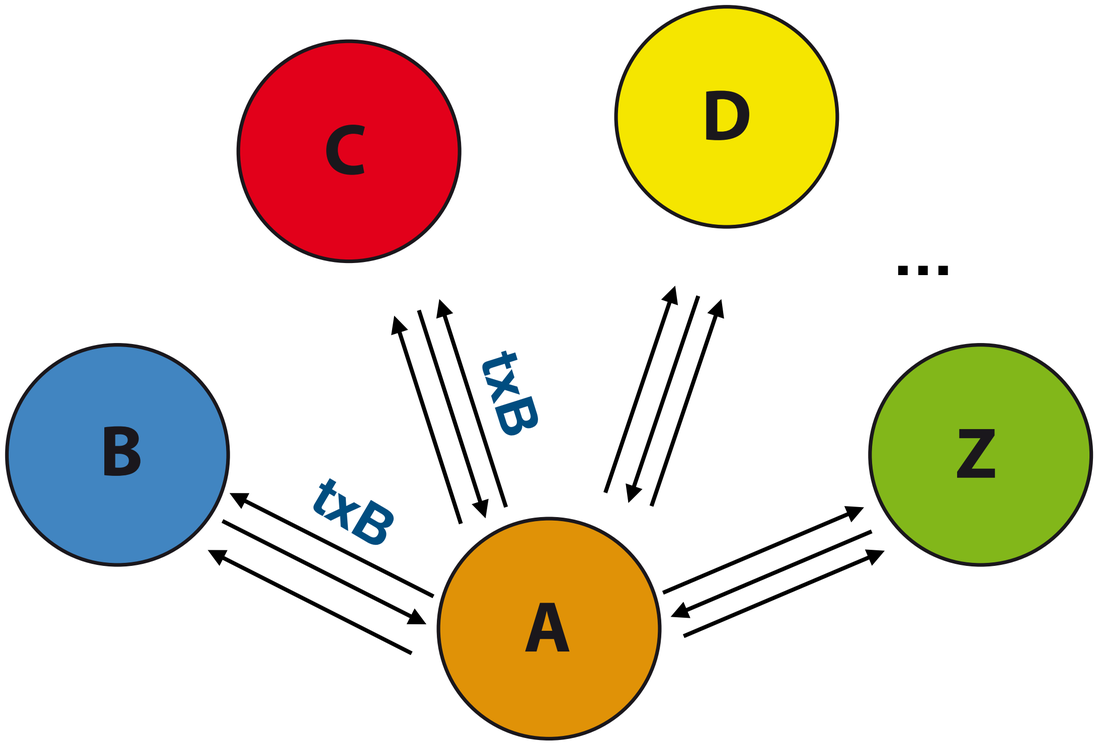
txB’

Source: 4F3…ED	To: Alice

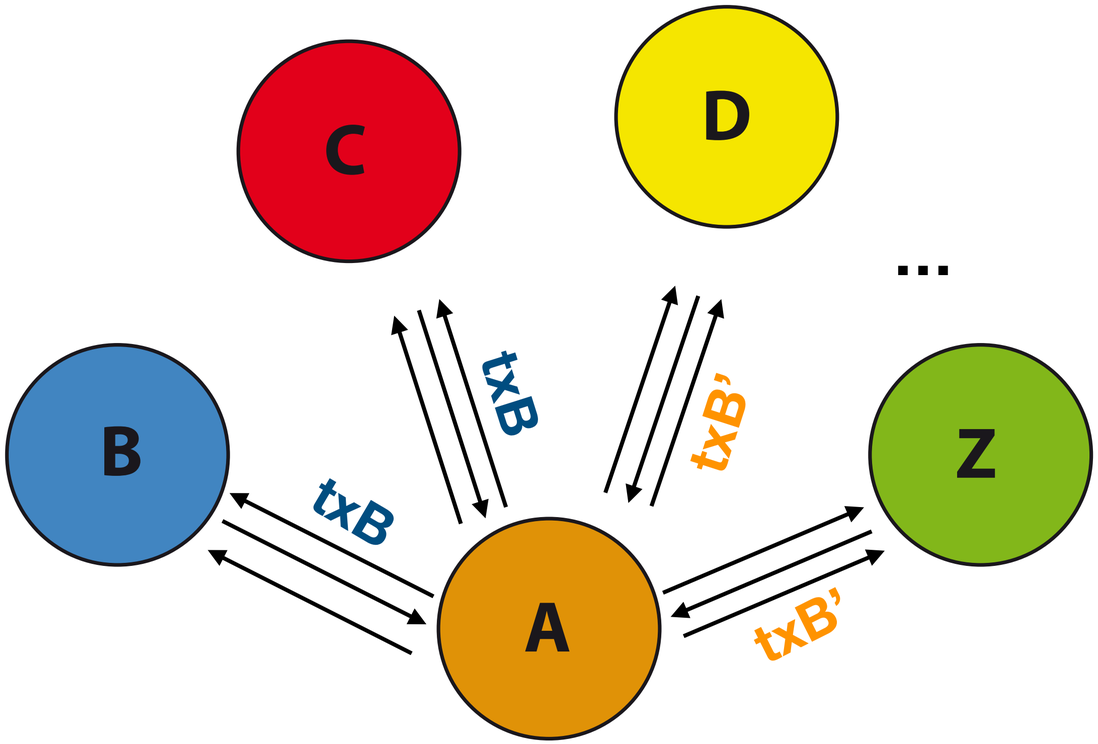
Double-spending transactions (2/2)



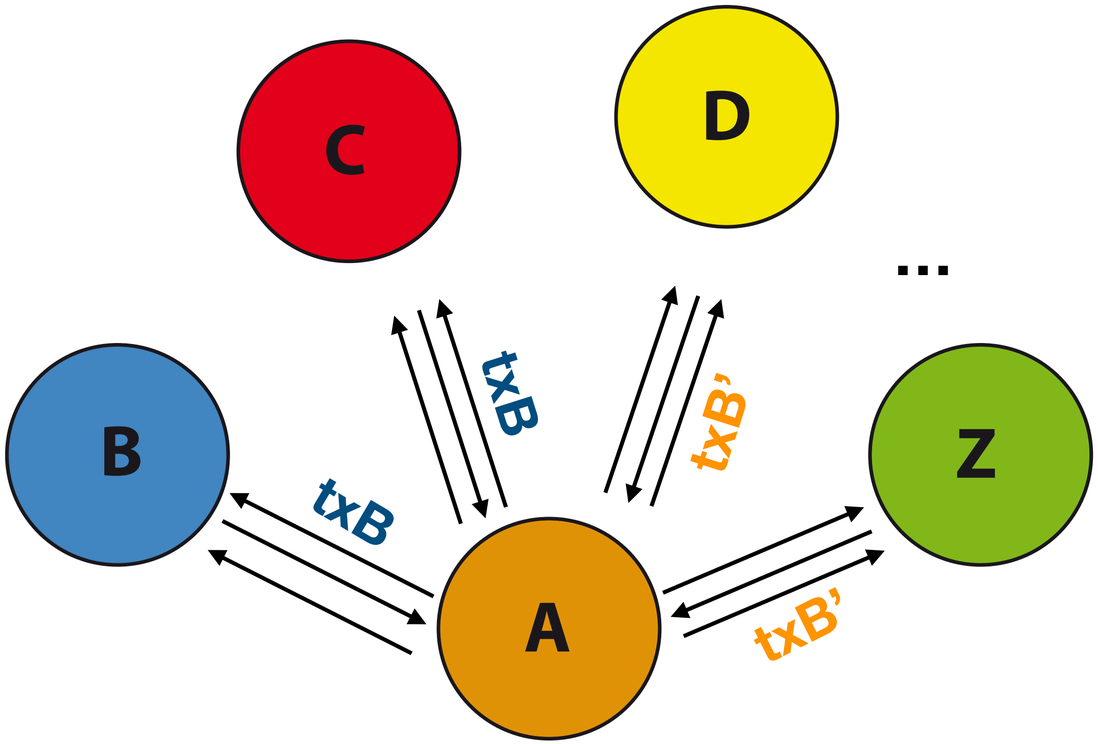
Double-spending transactions (2/2)



Double-spending transactions (2/2)

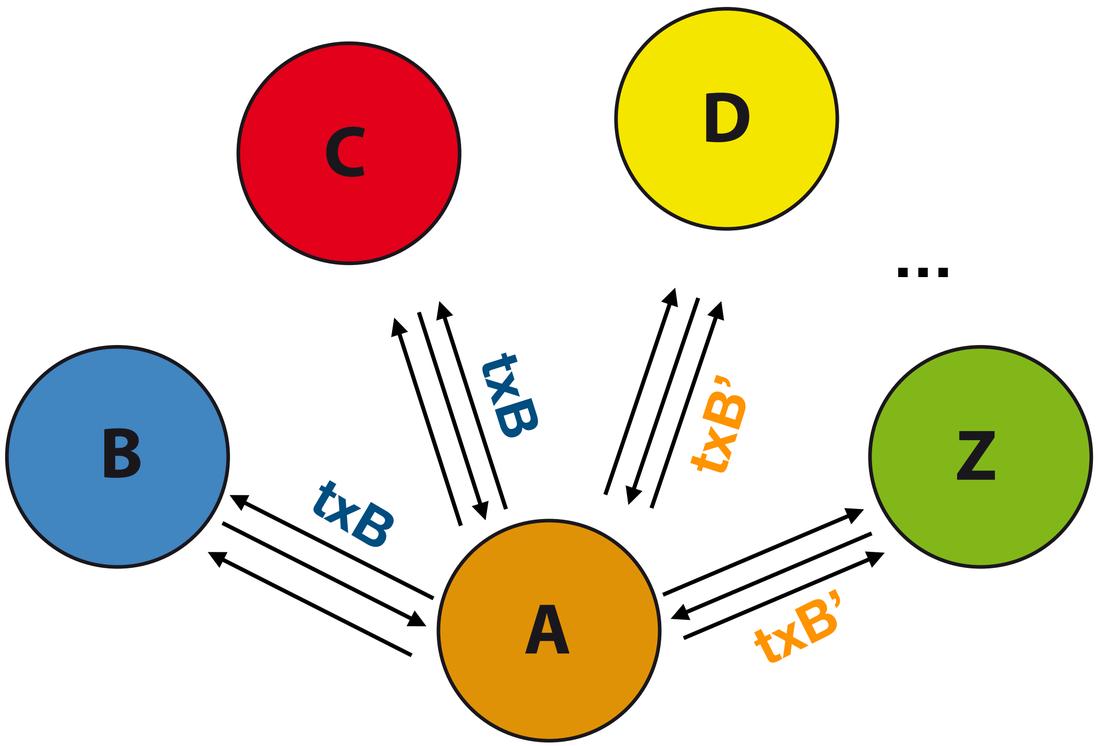


Double-spending transactions (2/2)



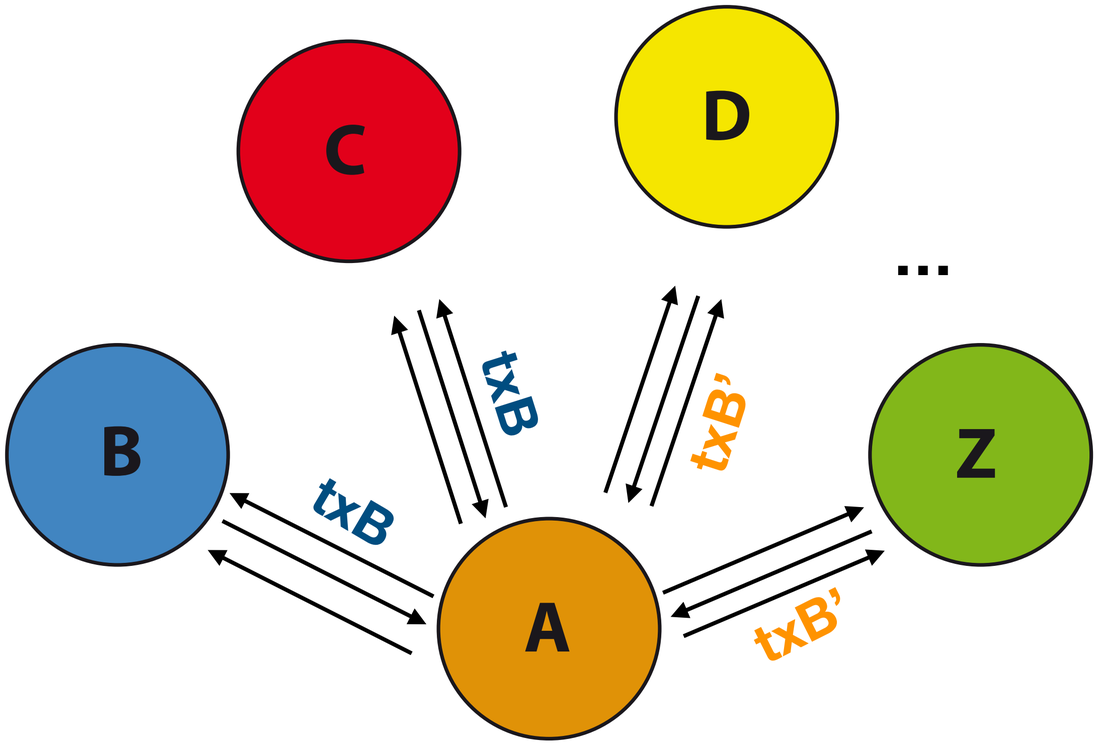
- **0-conf** transactions should not be trusted

Double-spending transactions (2/2)



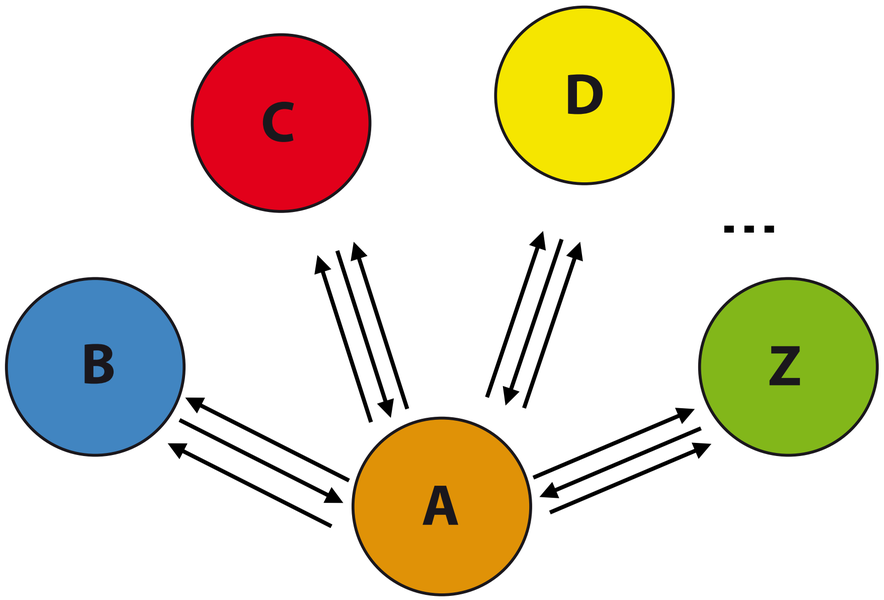
- **0-conf** transactions should not be trusted
- If B accepts txB before it appears in a block he **can be deceived** by A

Double-spending transactions (2/2)

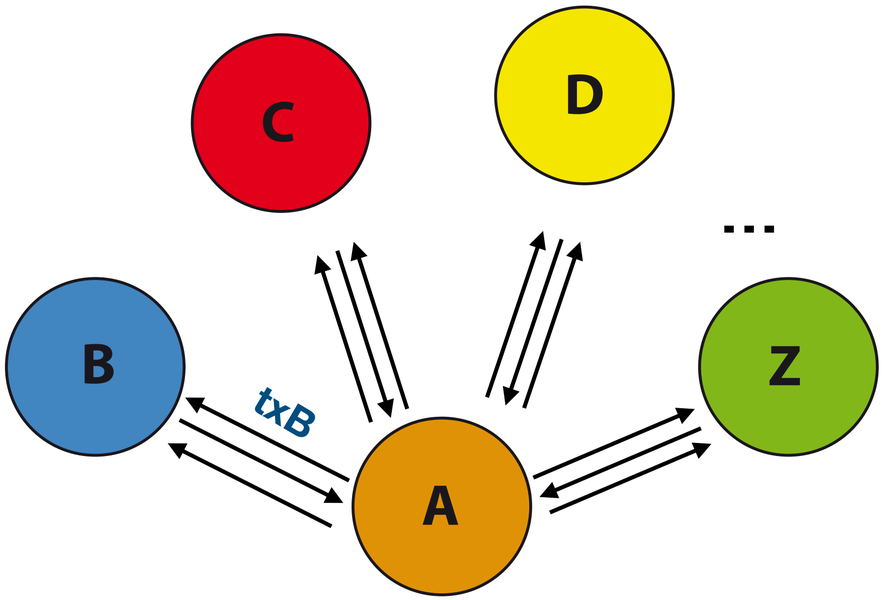


- **0-conf** transactions should not be trusted
- If B accepts txB before it appears in a block he **can be deceived** by A
- The de facto confirmation time is **6 blocks** (5 on top of the one including a certain transaction)

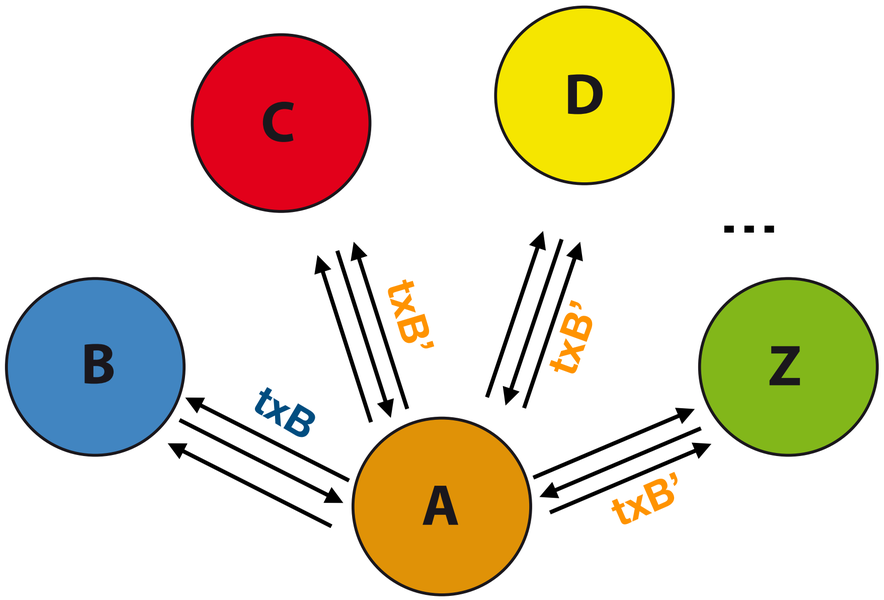
When things go south



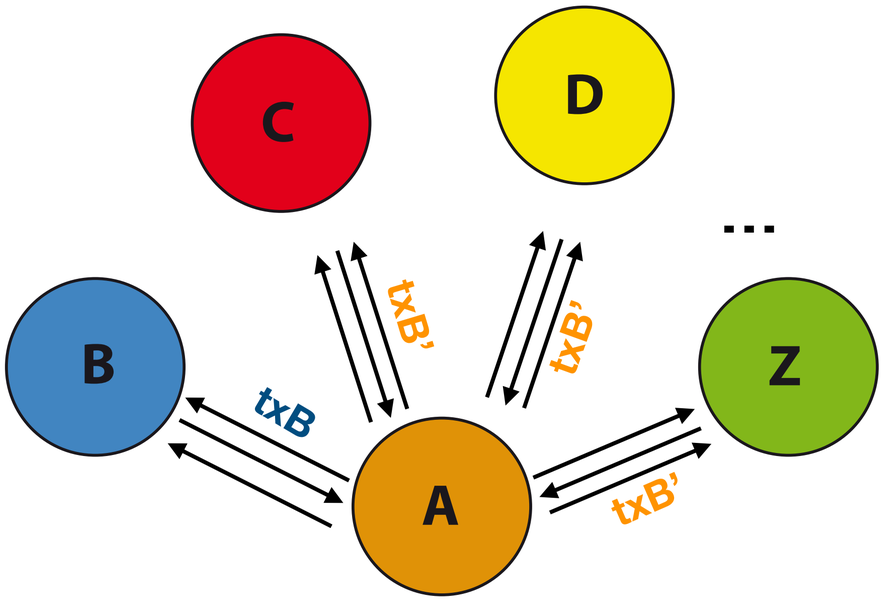
When things go south



When things go south

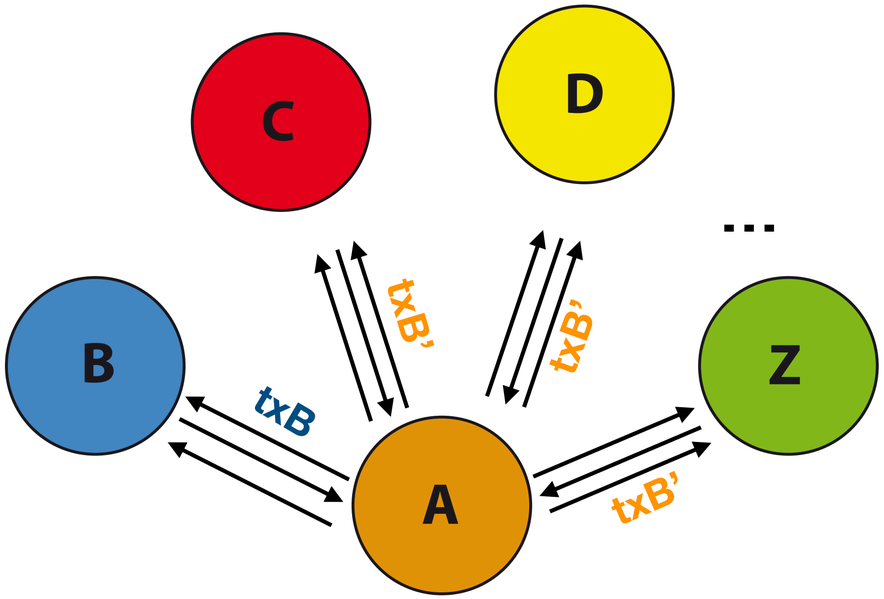


When things go south



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

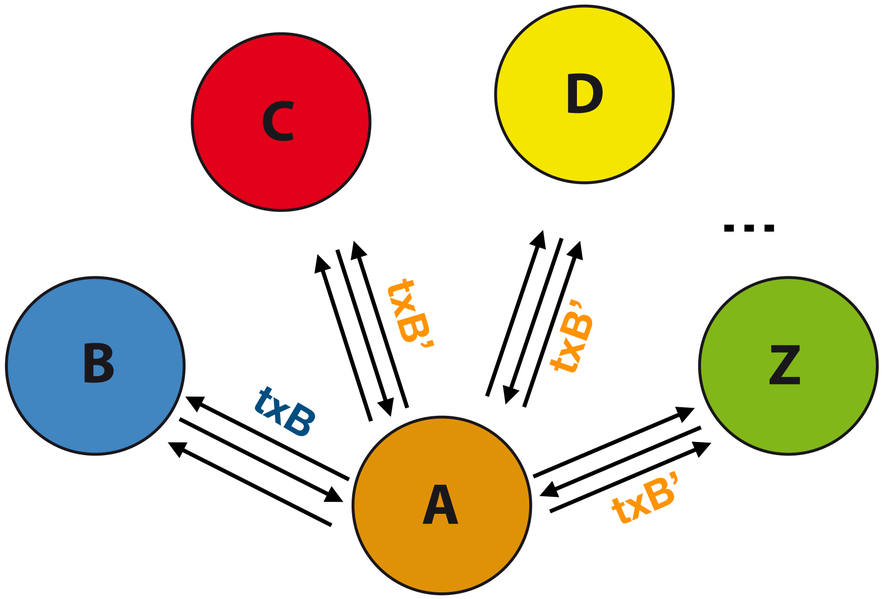
When things go south



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

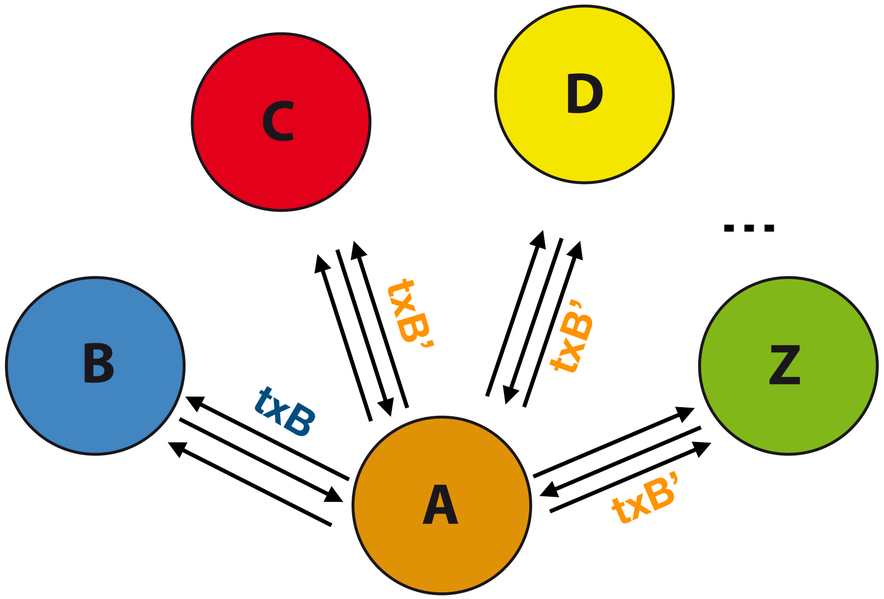
- B is said to be **eclipsed**

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

Title

Title

- Random topology and so on

Title

Title

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://arxiv.org/abs/1812.00942

Title

- A little bit about net topology and TxProbe picture

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://arxiv.org/abs/1812.00942

Title