Lecture 4: Peer 2 Peer Networking

https://web3.princeton.edu/principles-of-blockchains/

Professor Pramod Viswanath
Princeton University

This lecture:
P2P Networking; Random and Structured Graphs;
Engineering issues; Privacy at the network layer

Networking Requirements

No centralized server (single point of failure, censorship)

Key Primitive

Broadcast blocks and transactions to all nodes

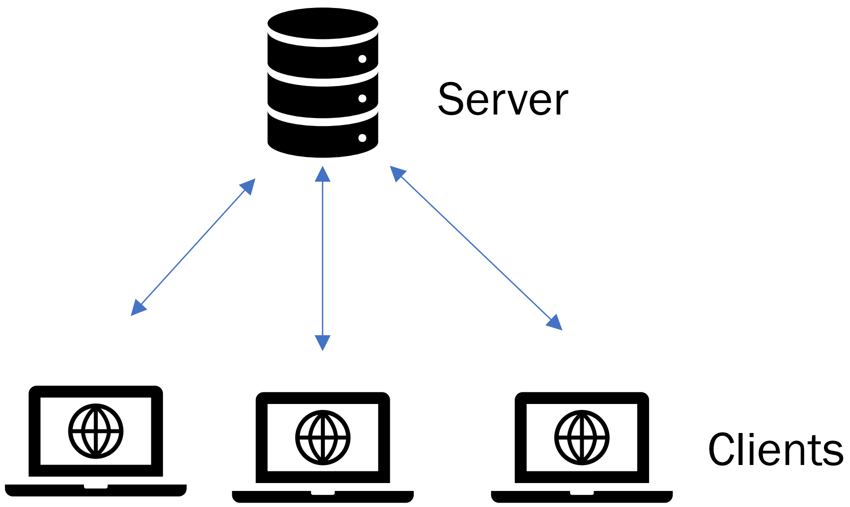
Robustness

some nodes go offline

new nodes join

Types of Network Architecture

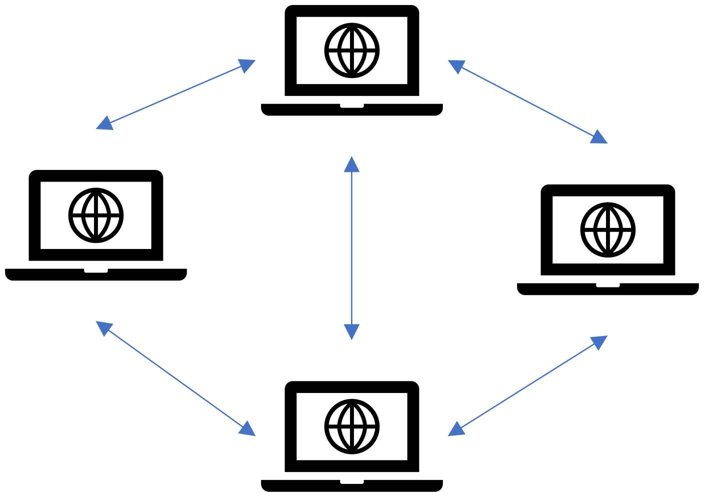
Client server



Server stores most of the data

COS/ECE 470 course website

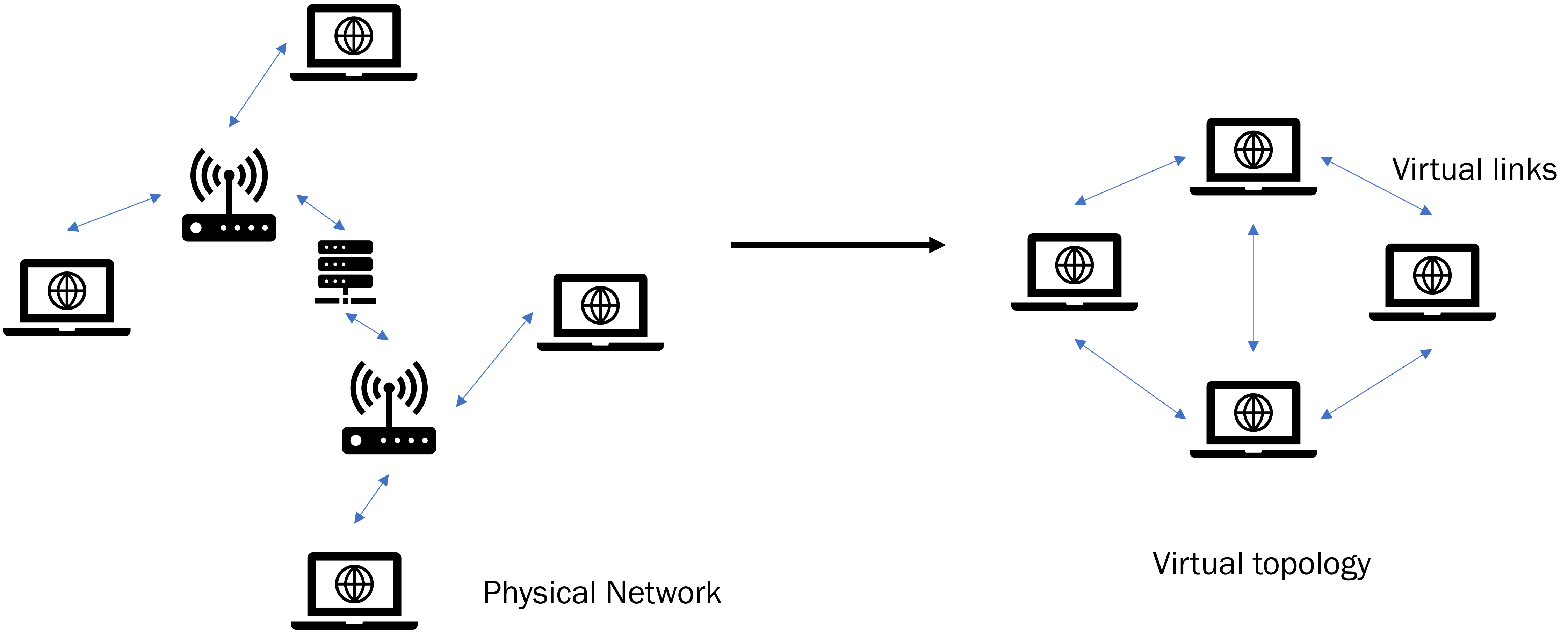
Peer to Peer



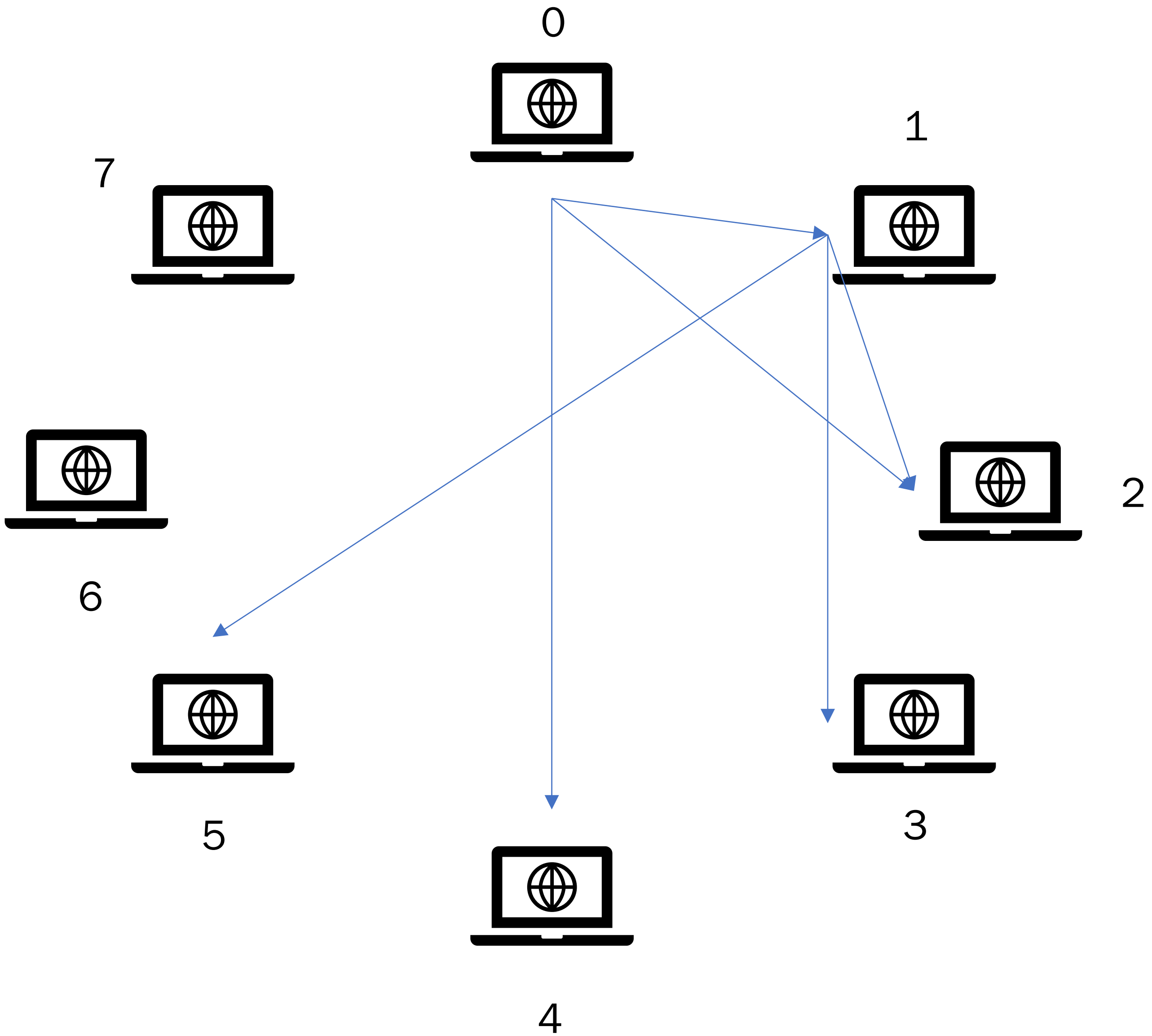
Each node acts as a client and a server

BitTorrent, Napster

Overlay Networks

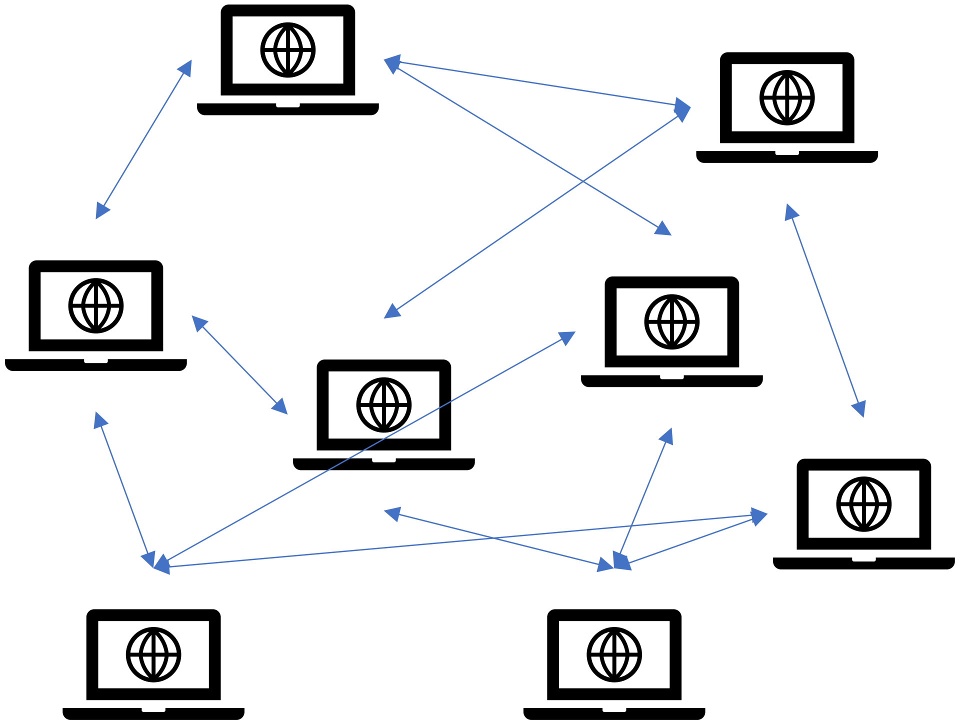


Structured Overlay Networks



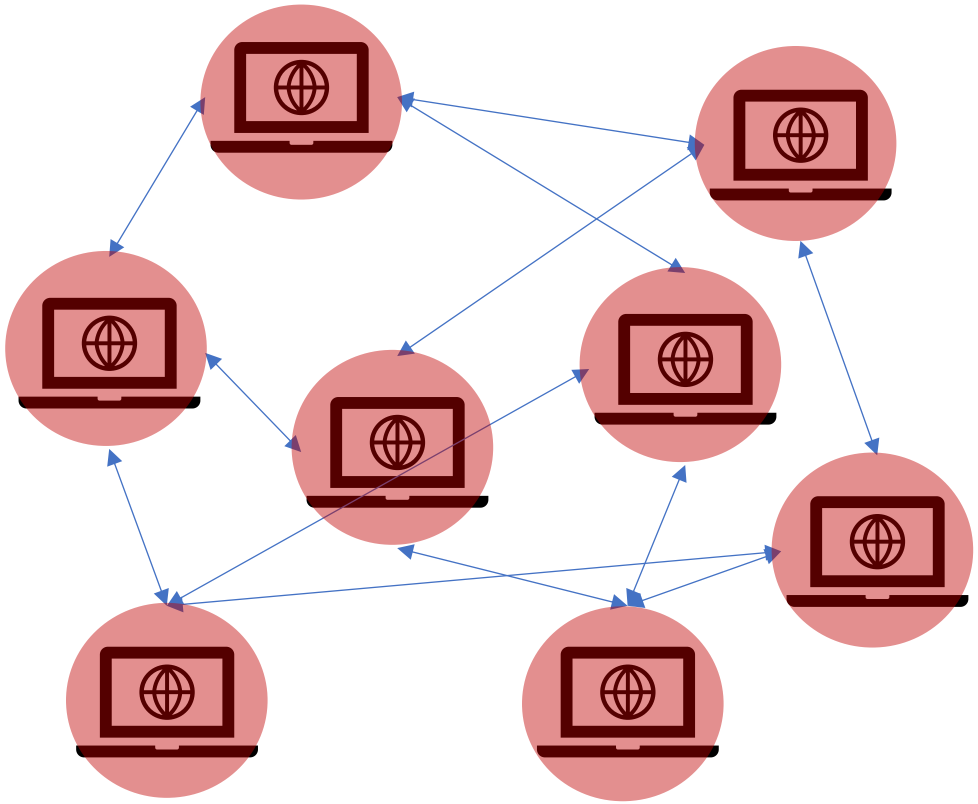
- Example: CHORD
- Assign a graph node identifier to each node
- Well defined Peer routing rules
- O(log N) routing
- O(log N) connections per node

Unstructured Overlay Networks



- Example: d-regular graph
- No node graph identifier
- Connect to any random d-nodes
- O(log N) routing (difficult to route)
- O(1) connections per node
- O(log N) broadcast

Gossip and Flooding



- Mimics the spread of an epidemic
- Once a node is “infected”, it “infects” its peers
- Information Spread exponentially and reaches nodes in O(log(N)) time

Expander graph

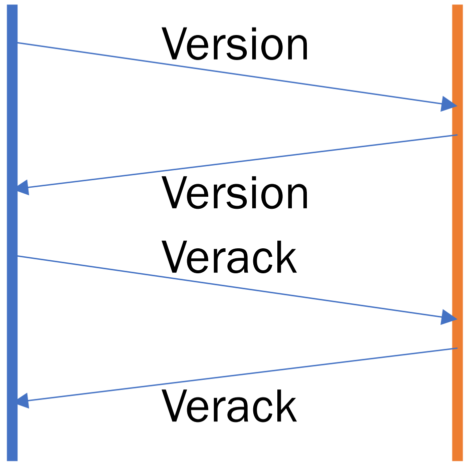
- Well connected but sparse graph
- Sparse graph G(V,E): |E| = O(|V|)
- Expander graph: $|\partial A| \geq \varepsilon |A|$
  - $|\partial A|$ = number of vertices outside A with at-least one neighbor in A.
- **A random d-regular (d $\geq$3) graph for large |V| is an expander graph** (with high probability)
- Intuition for O(log N) broadcast

Bitcoin network

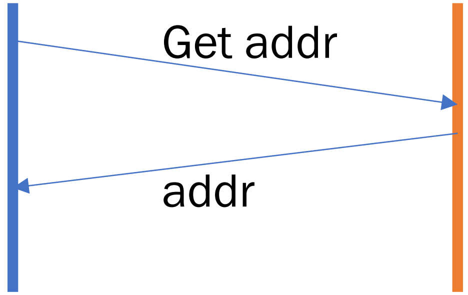
- TCP connection with peers
- At most 8 outbound TCP connections
- May accept up to 117 inbound TCP connections
- Maintains a large list of nodes (IP, port) on the bitcoin network
- Establishes connection to a subset of the stored nodes

Peer discovery

- DNS seed nodes (Hard coded in the codebase)
- Easy to be compromised, do not trust one seed node exclusively
- Hardcoded peers (fallback)
- Ask connected peers for additional peers



Connecting to a peer



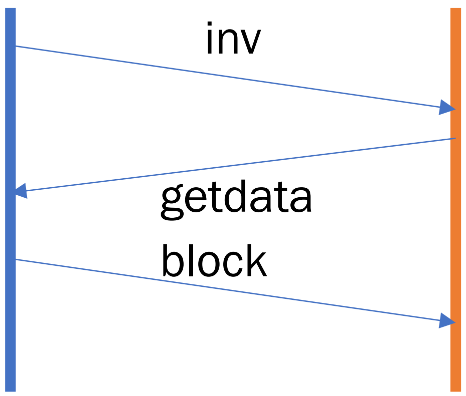
Gathering additional peers

Addr: contains list of up to 1000 nodes

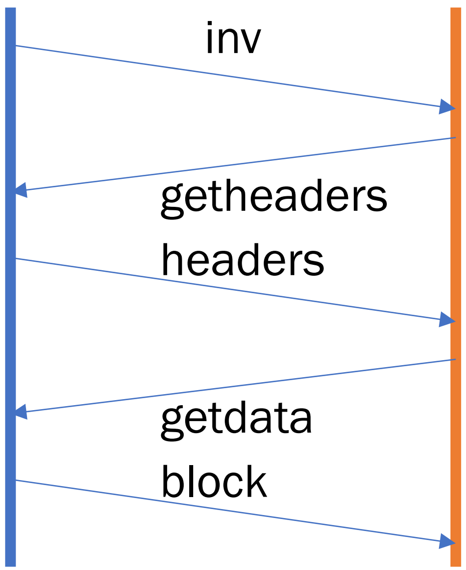
Block transmission

- Block is broadcasted to the network using gossip-flooding
- Standard block relay protocol to gossip blocks
- Relay after block validation
- Inv(blockhash): inventory message containing blockhash

- Block-First
- Getdata asks for the same block as inv
- Can download orphan blocks and keep it in memory



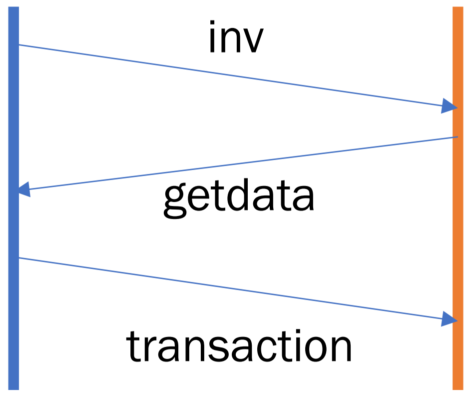
or



- Header-First
- Getheaders asks for the same block as inv or a few parent headers (in case of orphan block)
- Will not download orphan blocks if no header chain established

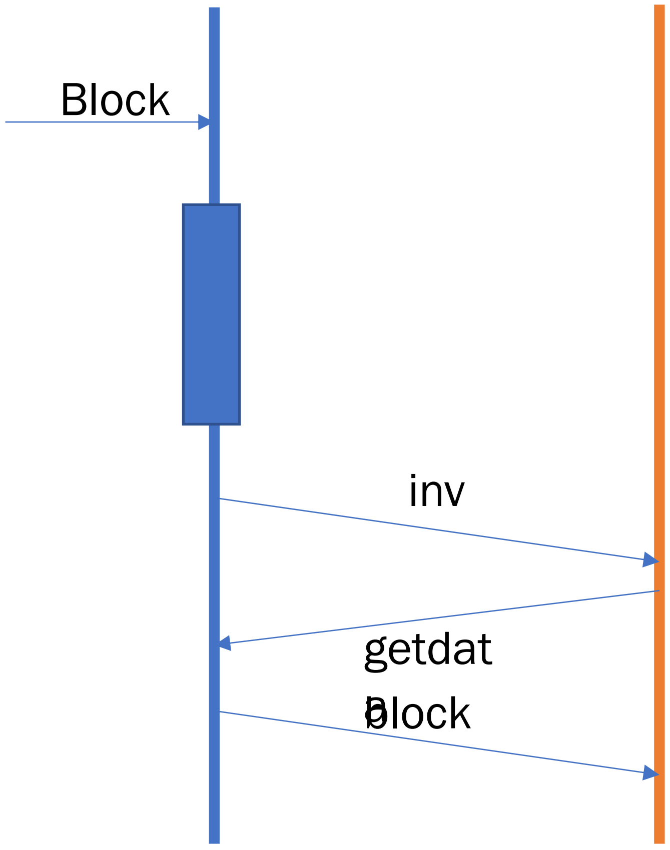
Data Broadcast

- Data (transactions) broadcasted using Gossip-flooding
- Each node maintains a non-persistent memory to store unconfirmed tx (mempool)
- inv(txid): Check if peer has a transaction with id: txid in mempool

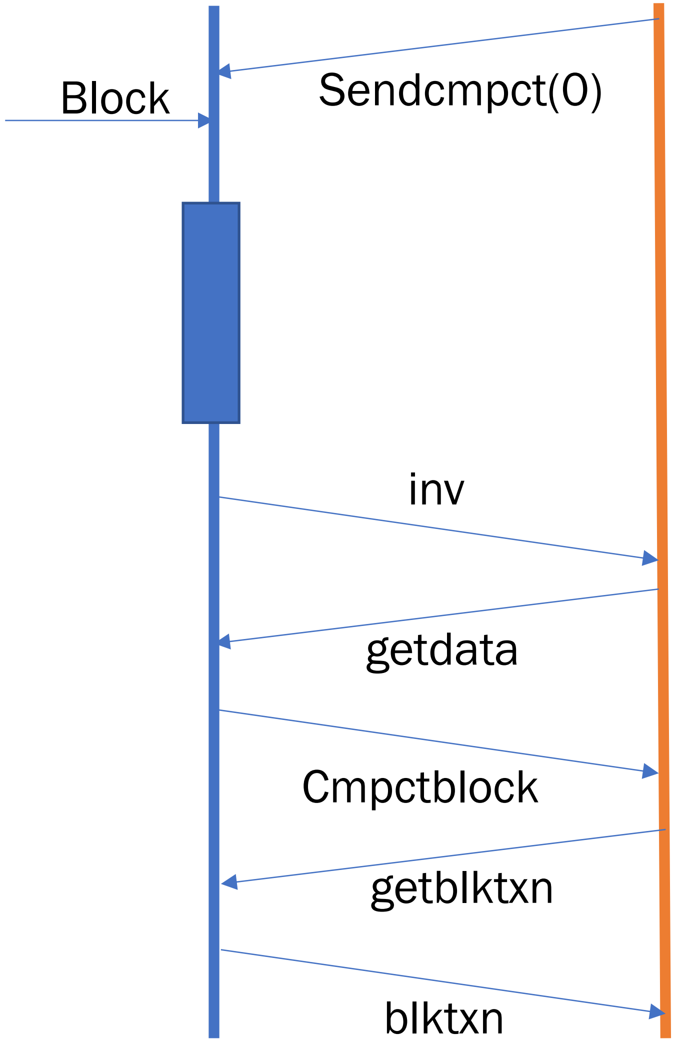


- Some unconfirmed tx might be removed from mempool

Compact blocks



Legacy relaying



Compact block relaying

- Legacy relaying sends transactions twice
- Guess the mempool of the receiving node
- Compact block has block header, txids, some full transactions
- The receiving node sends getblktxn to receive missing transactions

Capacity and Propagation Delay

- C = communication/processing capacity of the network (tx/s)

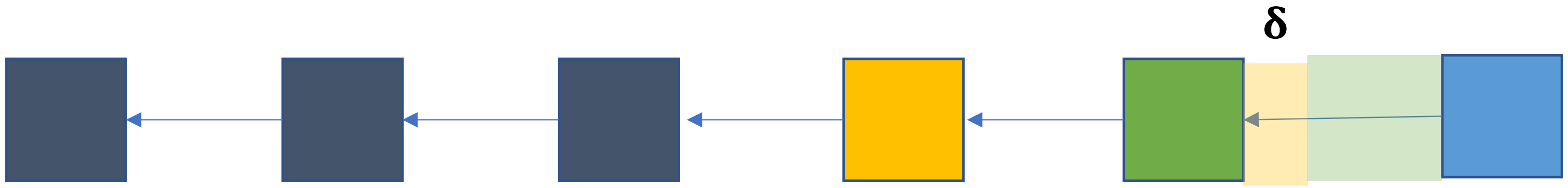
- D = speed-of-light propagation delay

- End to end delay
  1. Propagation delay: D
  2. Processing delay: B/C (where B is block size in tx)
  3. Queuing delay

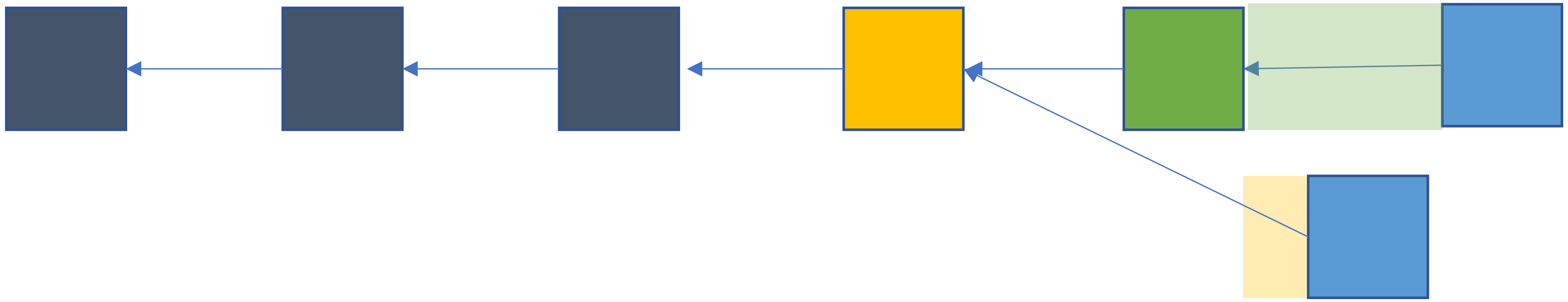
- Delay increases with increase in block size

Effects of delay

- Wasted hashpower



- Forking



Disadvantages of current p2p network

- Efficiency
  - Not efficient (total communication is O(Nd))
- Privacy
  - Can link transaction source to IP address
- Security
  - Plausible deniability for forking

Improving P2P Network Topology

- Random IP network
  - Not related to geographic distances
- Need a **geometric** random network
  - IP addresses do not necessarily reveal location
- Challenge
  - Self-adapting network topology based on measurements

Perigee

- A self-adaptive network topology algorithm
  - Goal: mimic random geometric network

- Decentralized algorithm that selects neighbors based on past interactions
  retain neighbors that relay blocks fast
  disconnect from neighbors that do not relay blocks fast
  explore unseen neighbors

- Motivated by the multi-armed bandit problem
  - Explore vs exploit tradeoff

Perigee Algorithm

- Assign scores for each subset of neighbors based on how fast they relay blocks
- Retain subset neighbors with best score
- Disconnect node not in the subset
- Form a connection to a random neighbor

Efficient Networking

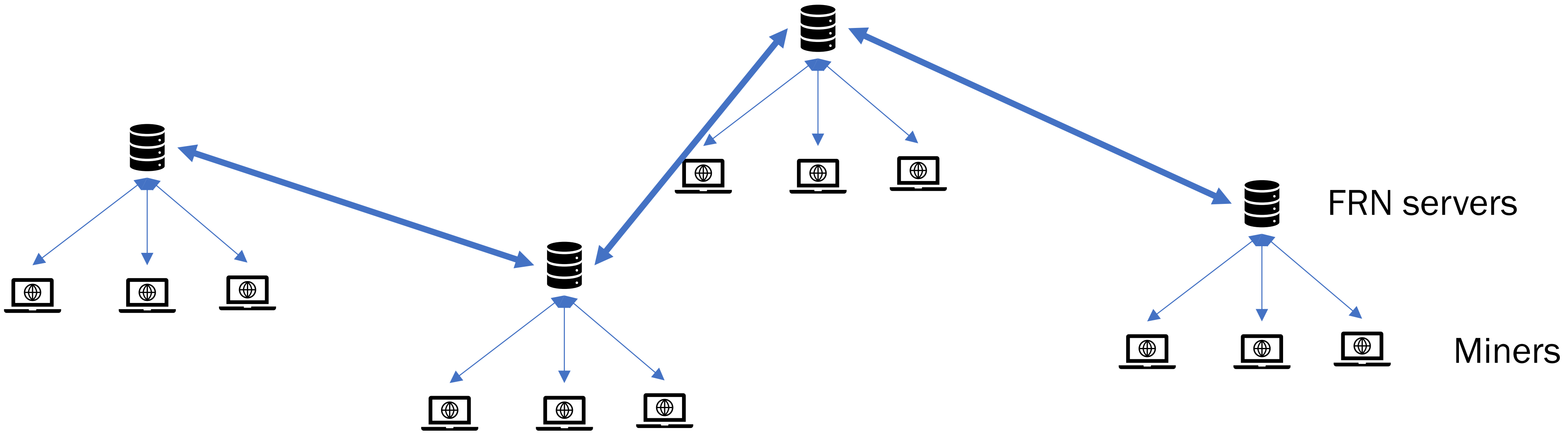
- **Trusted networks**

- Privacy
  - Can link transaction source to IP address

- Security
  - Plausible deniability for forking
  - Eclipse attacks

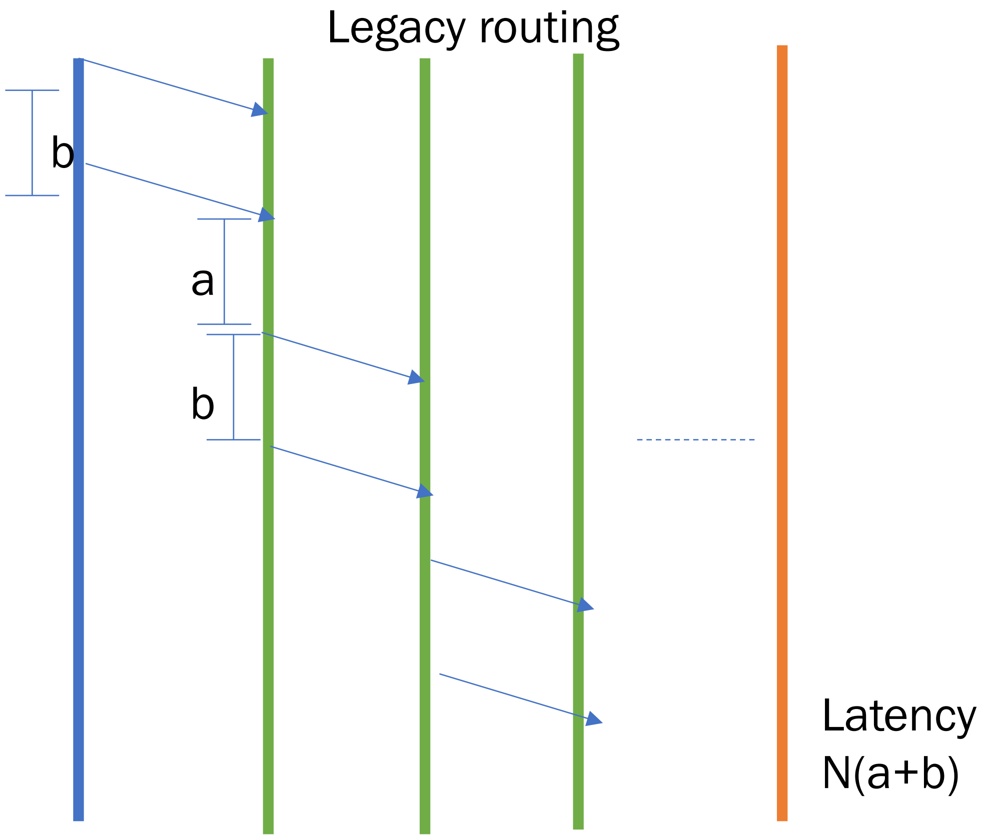
Trusted Networks

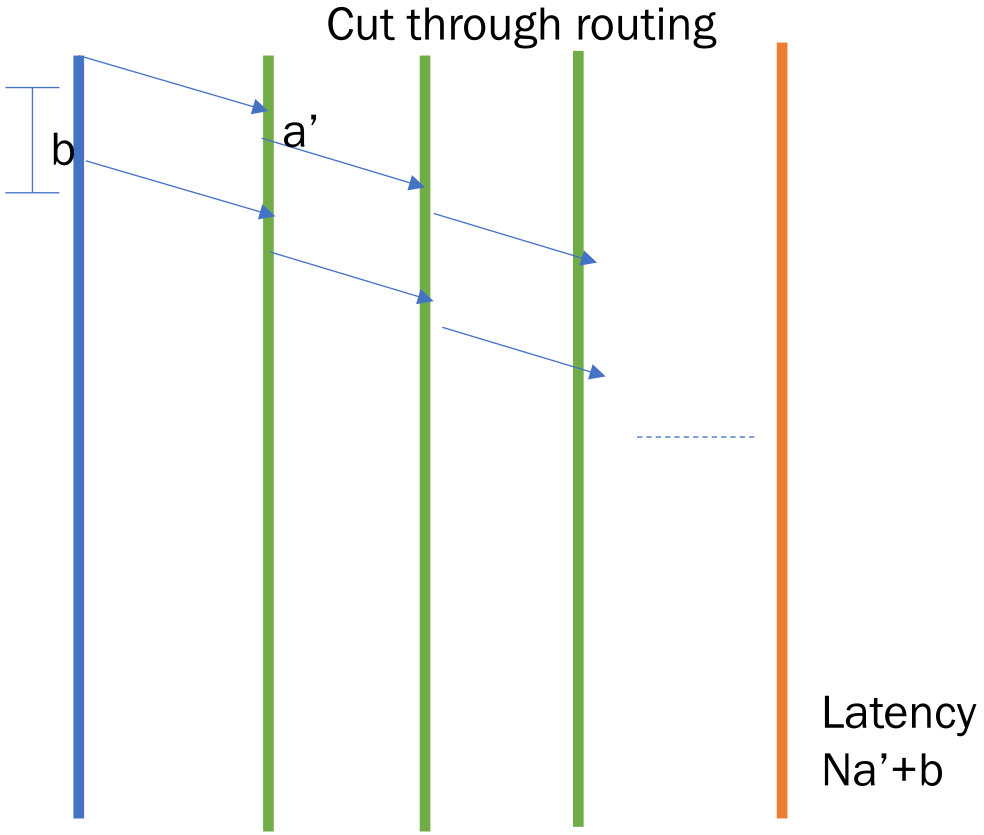
- FRN (Fast relay network): Hub and spoke model, trusted servers, servers are fast



Trusted Networks: Falcon

- Cut through routing for servers, only verify headers before forwarding



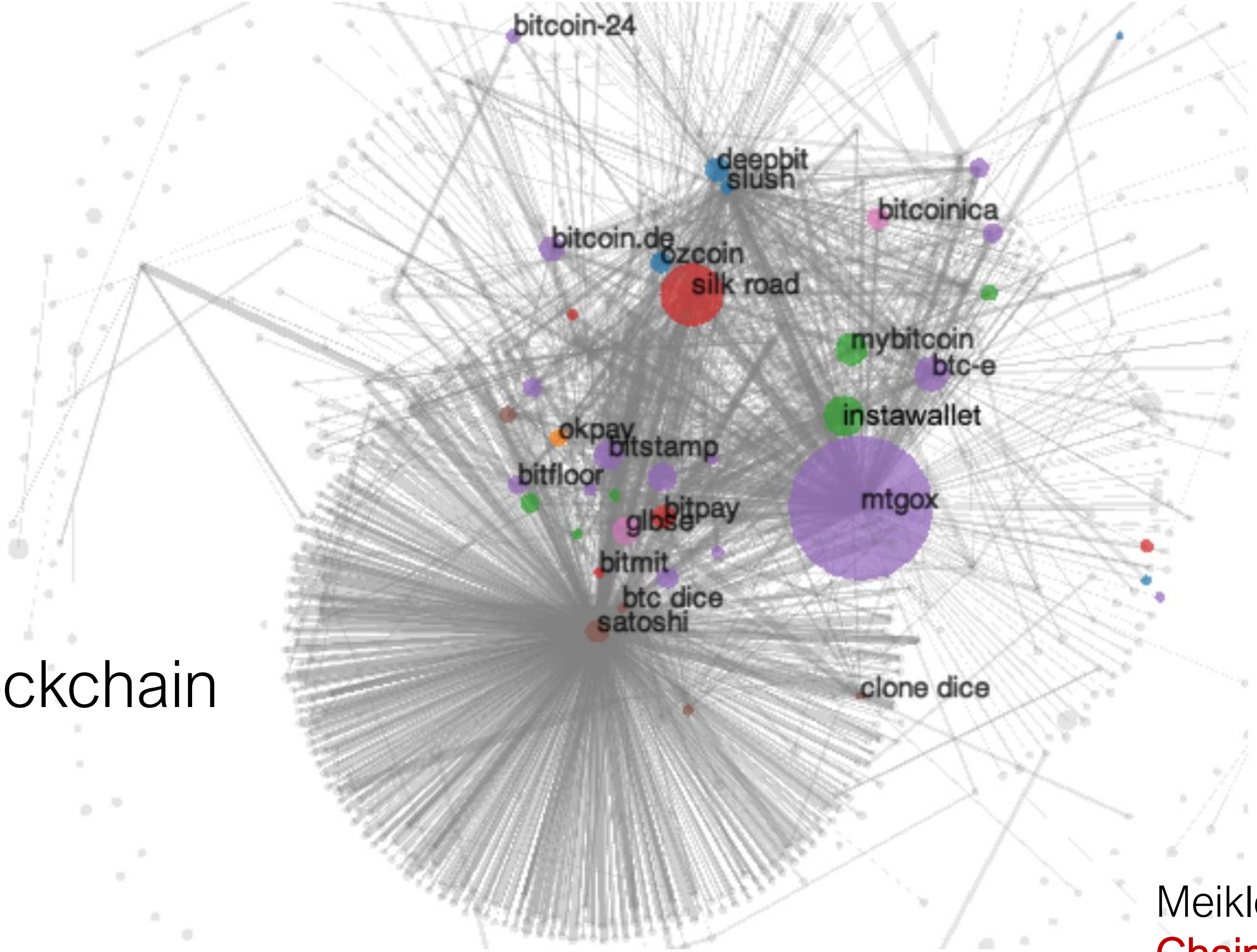


Network Privacy

Network privacy leakage

Scaling network anonymity

How can users be deanonymized?



Blockchain

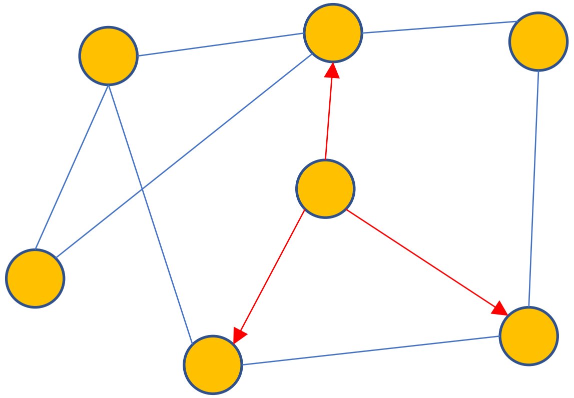
Meiklejohn et al., 2013
Chainalysis

What about the peer-to-peer network?

Public Key ⟷ IP Address

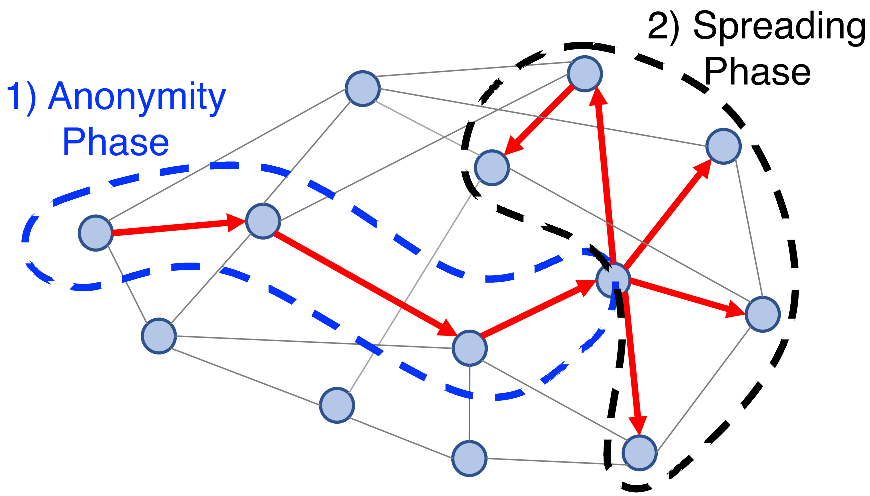
Dandelion

Deanonymization Analysis

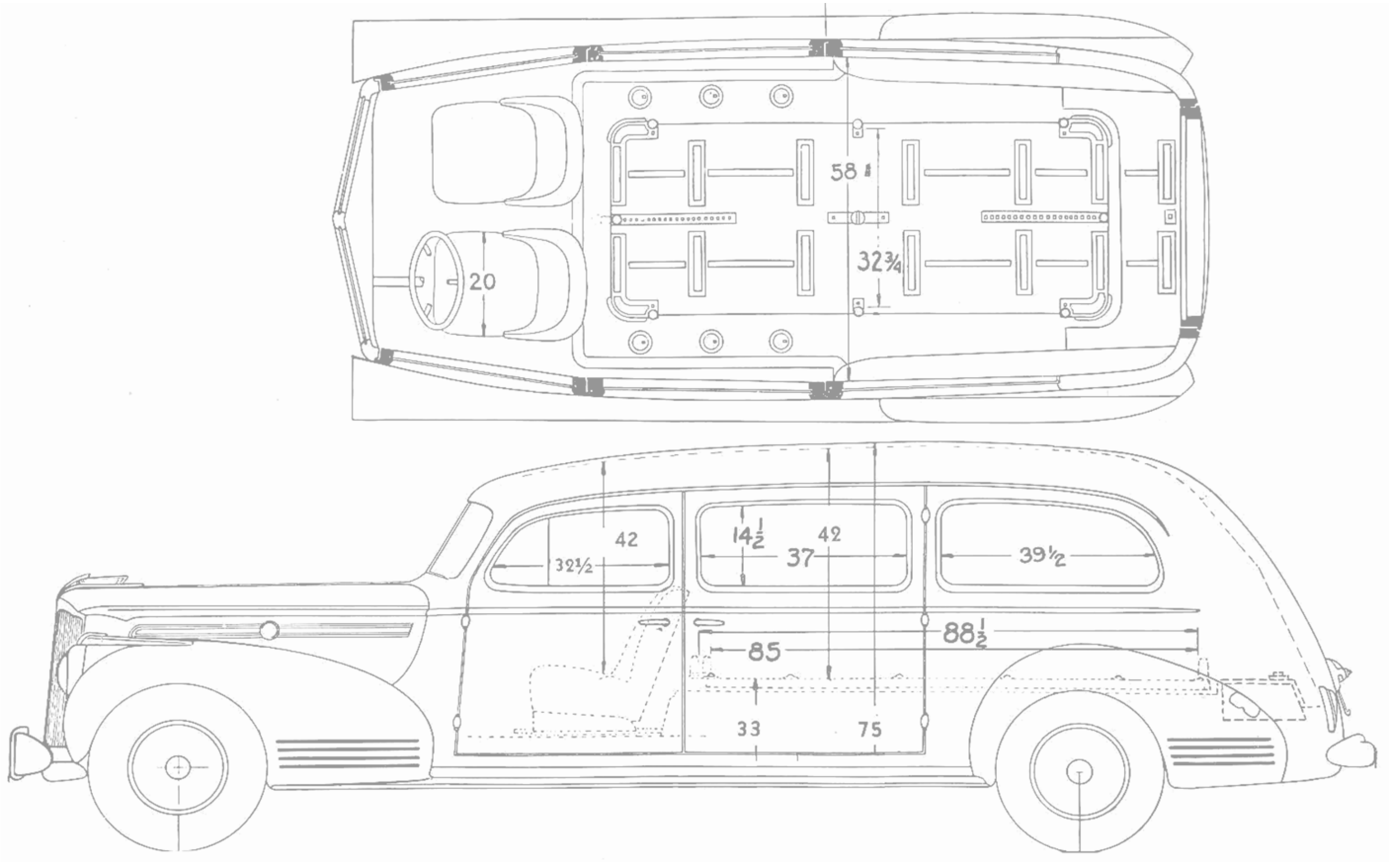


Pr(detection)

Redesign for Anonymity



Dandelion

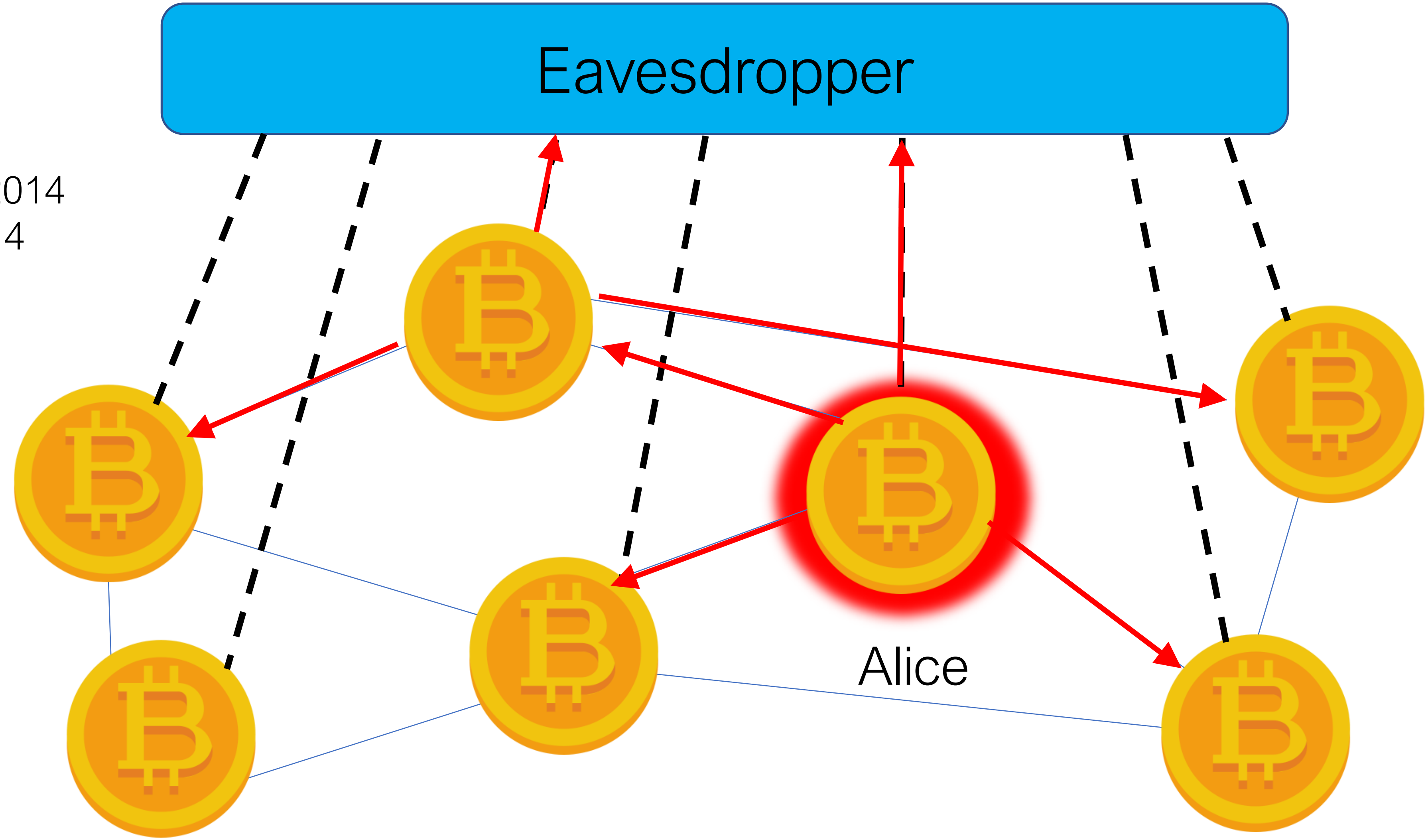


Model

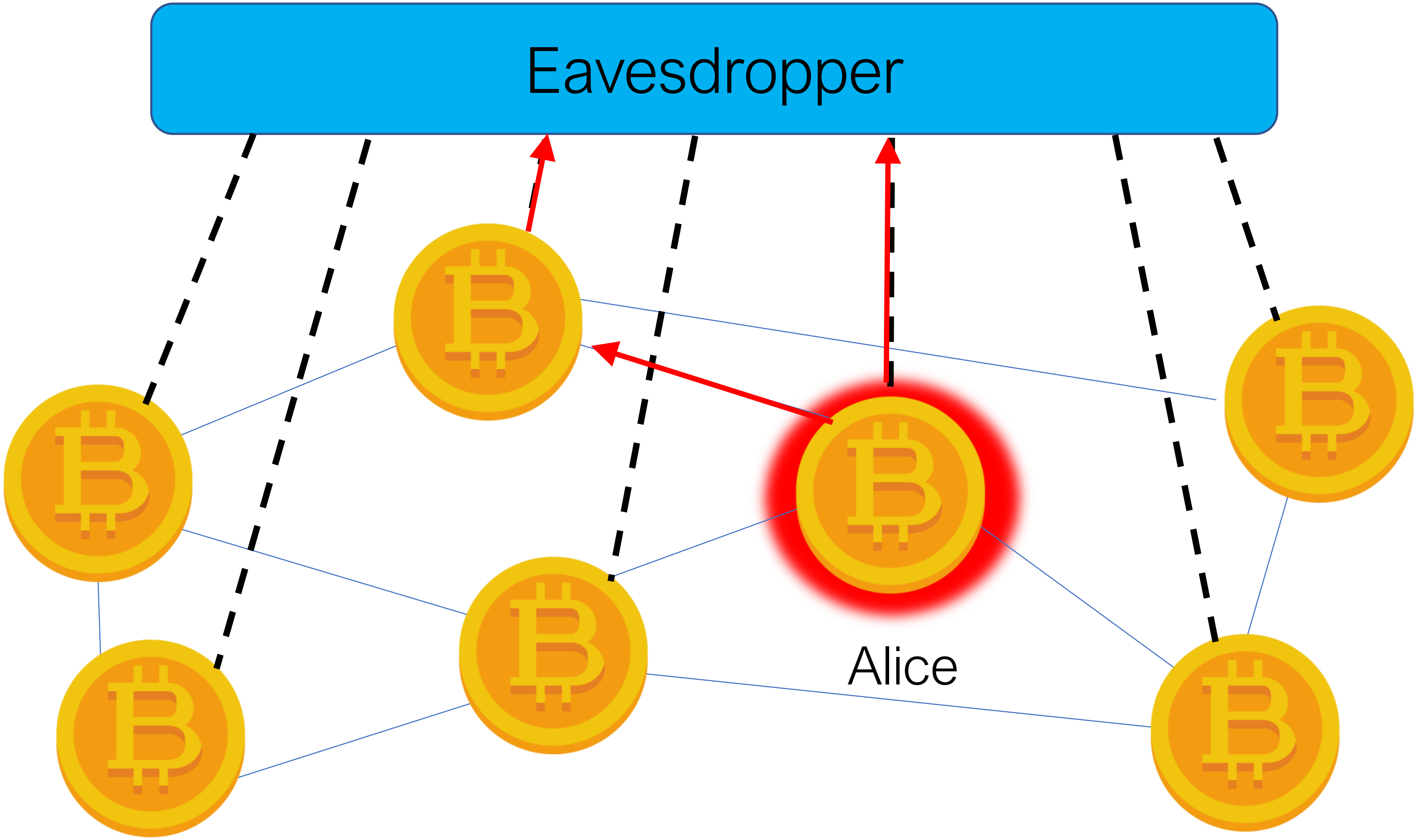
Assumptions and Notation

Attacks on the Network Layer

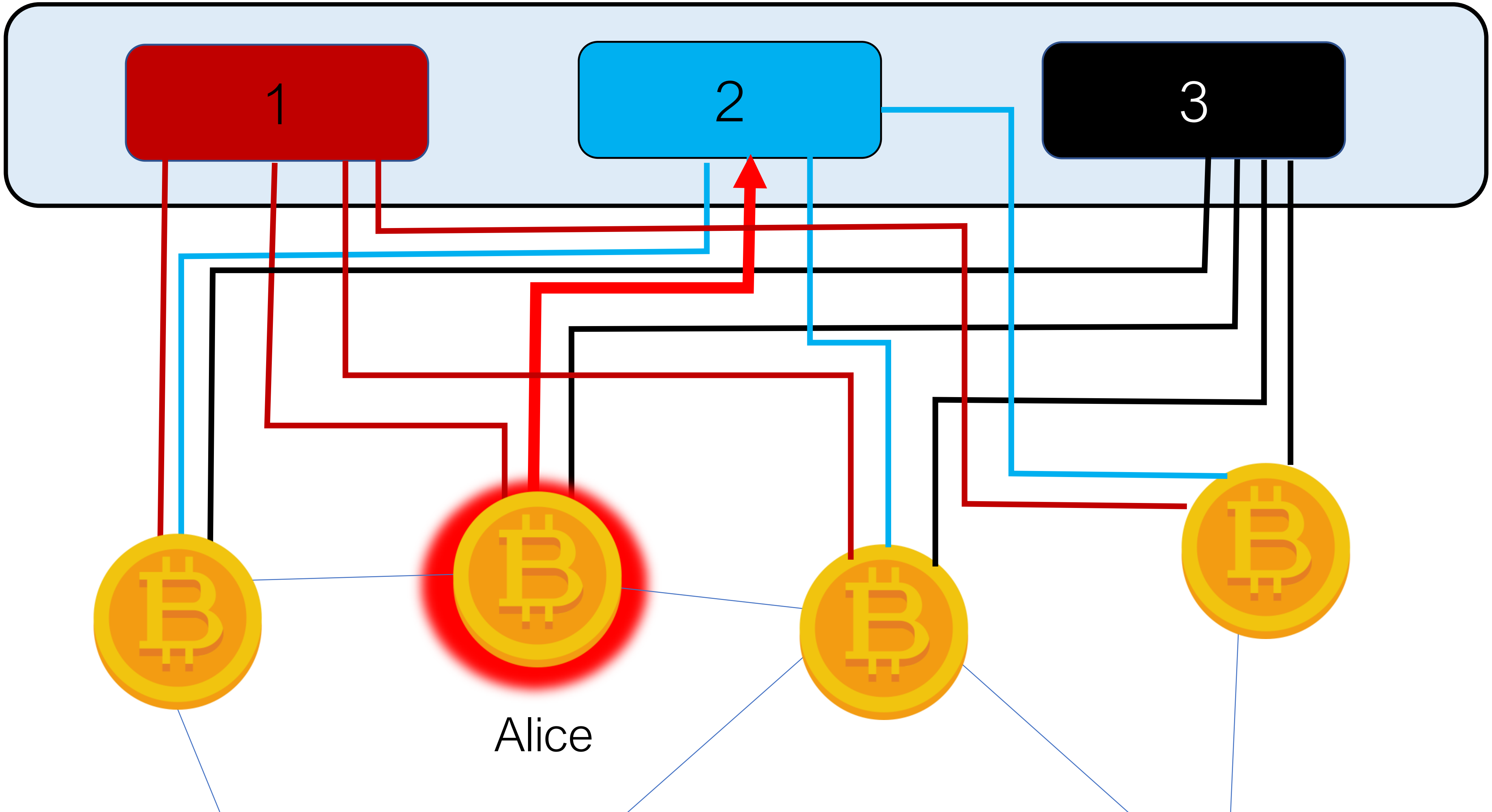
Biryukov et al., 2014
Koshy et al., 2014



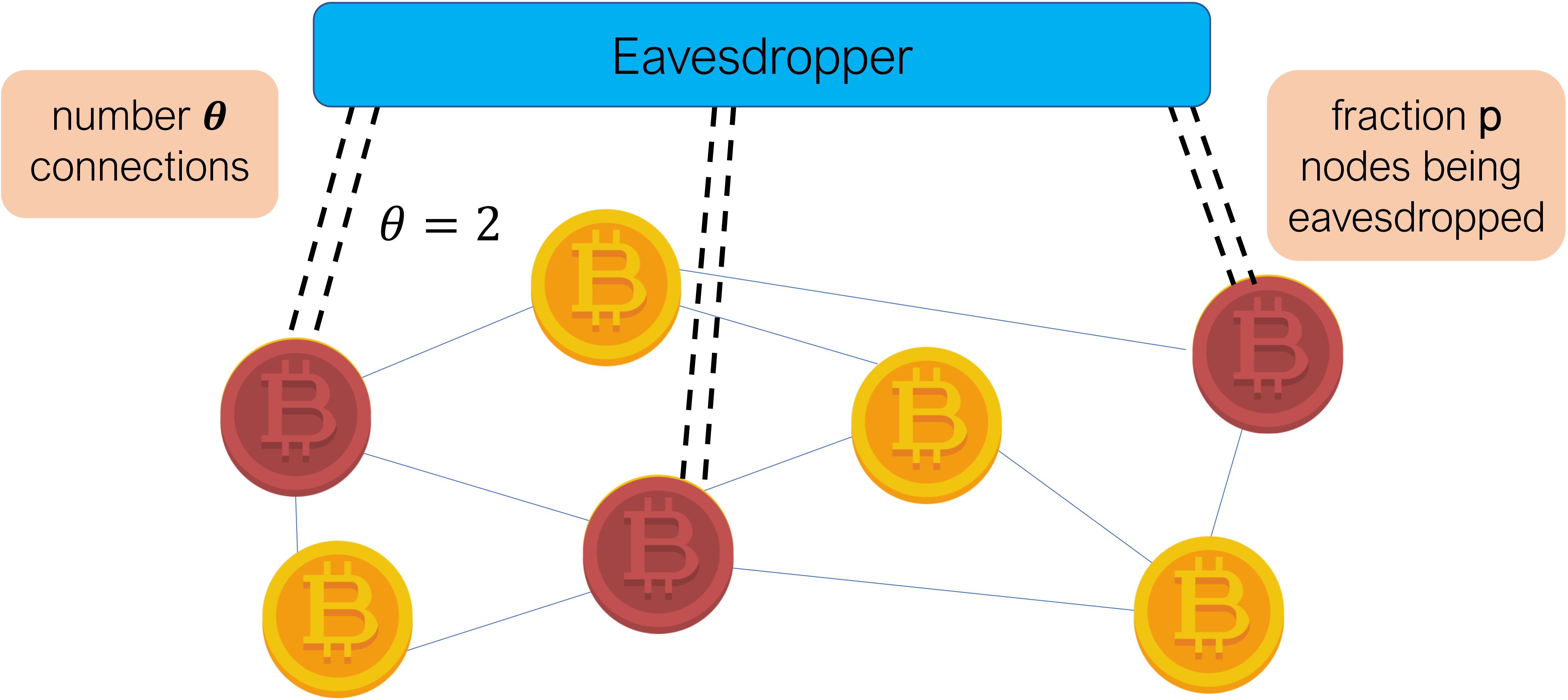
What can go wrong?

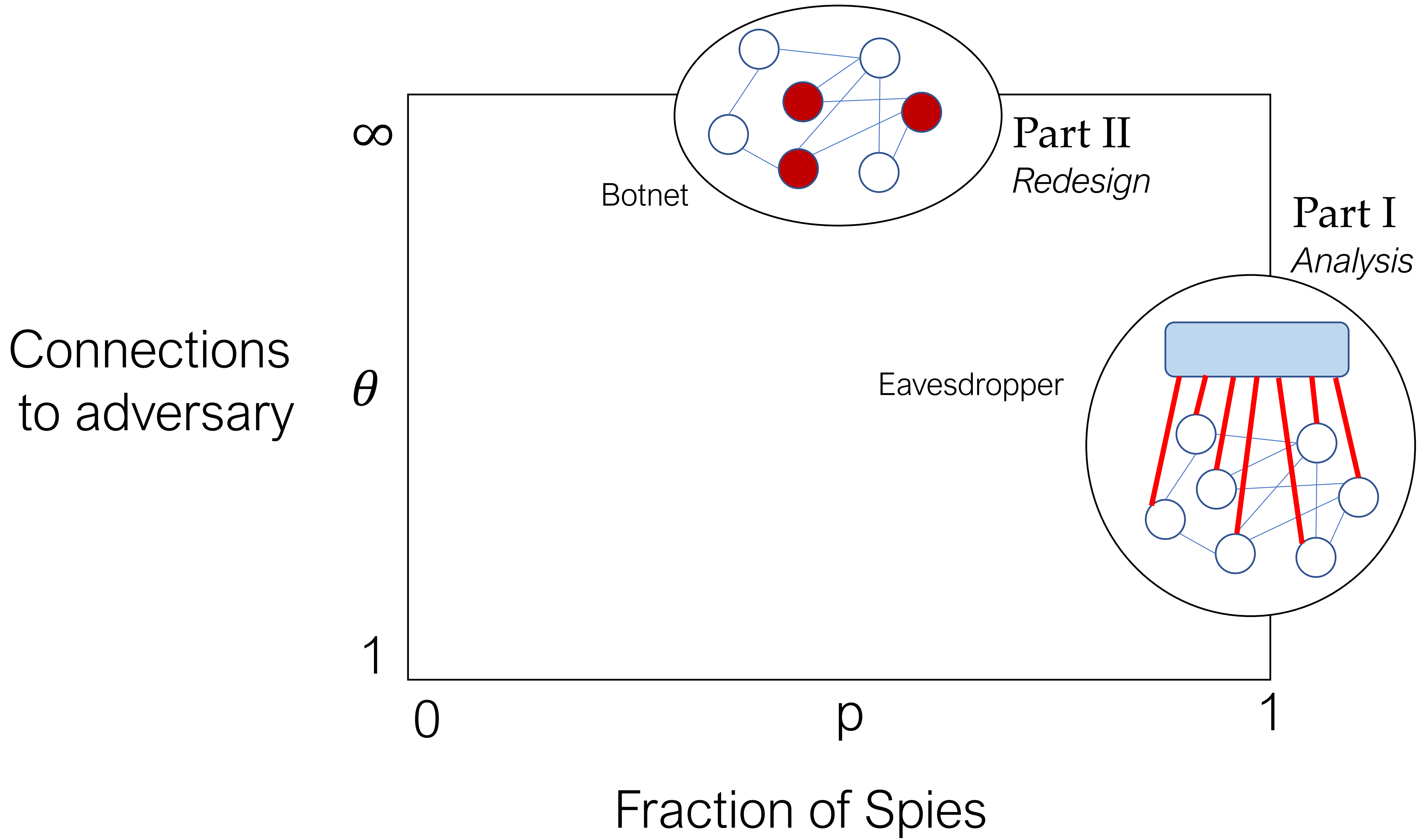


What the eavesdropper can do about it



Summary of adversarial model





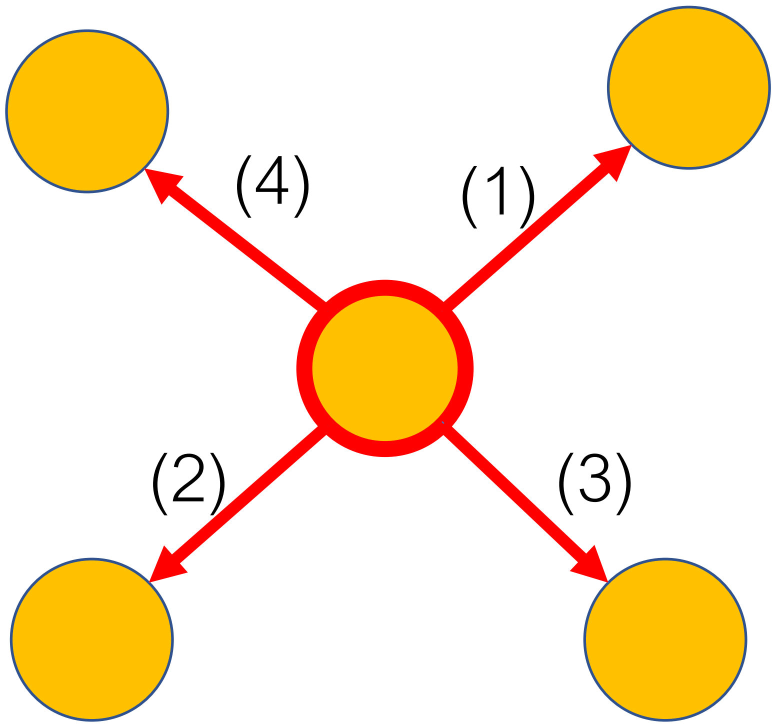
Part II
Redesign
Botnet
Part I
Analysis
Eavesdropper
Connections to adversary
∞
θ
1
0
p
1
Fraction of Spies

Analysis

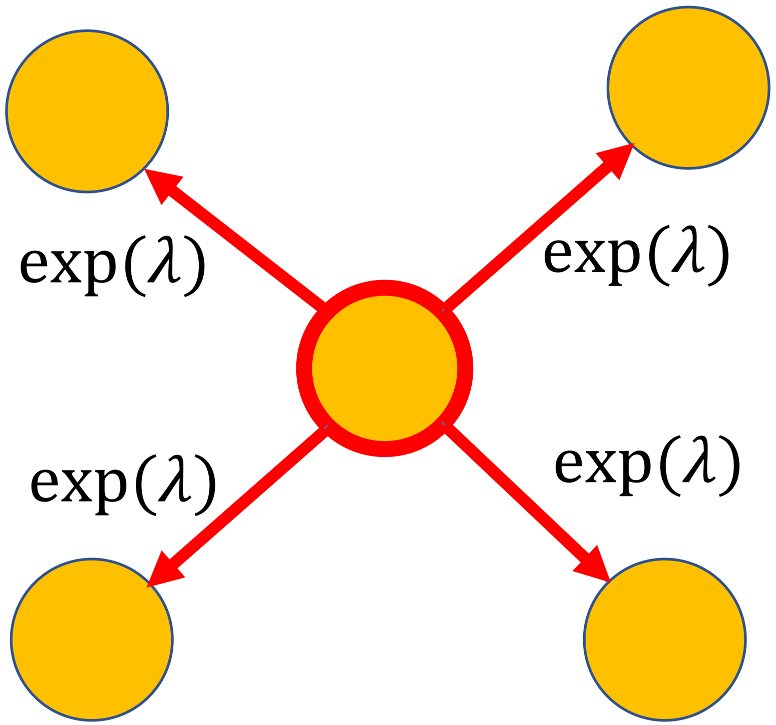
How bad is the problem?

Flooding Protocols

Trickle (pre-2015)



Diffusion (post-2015)

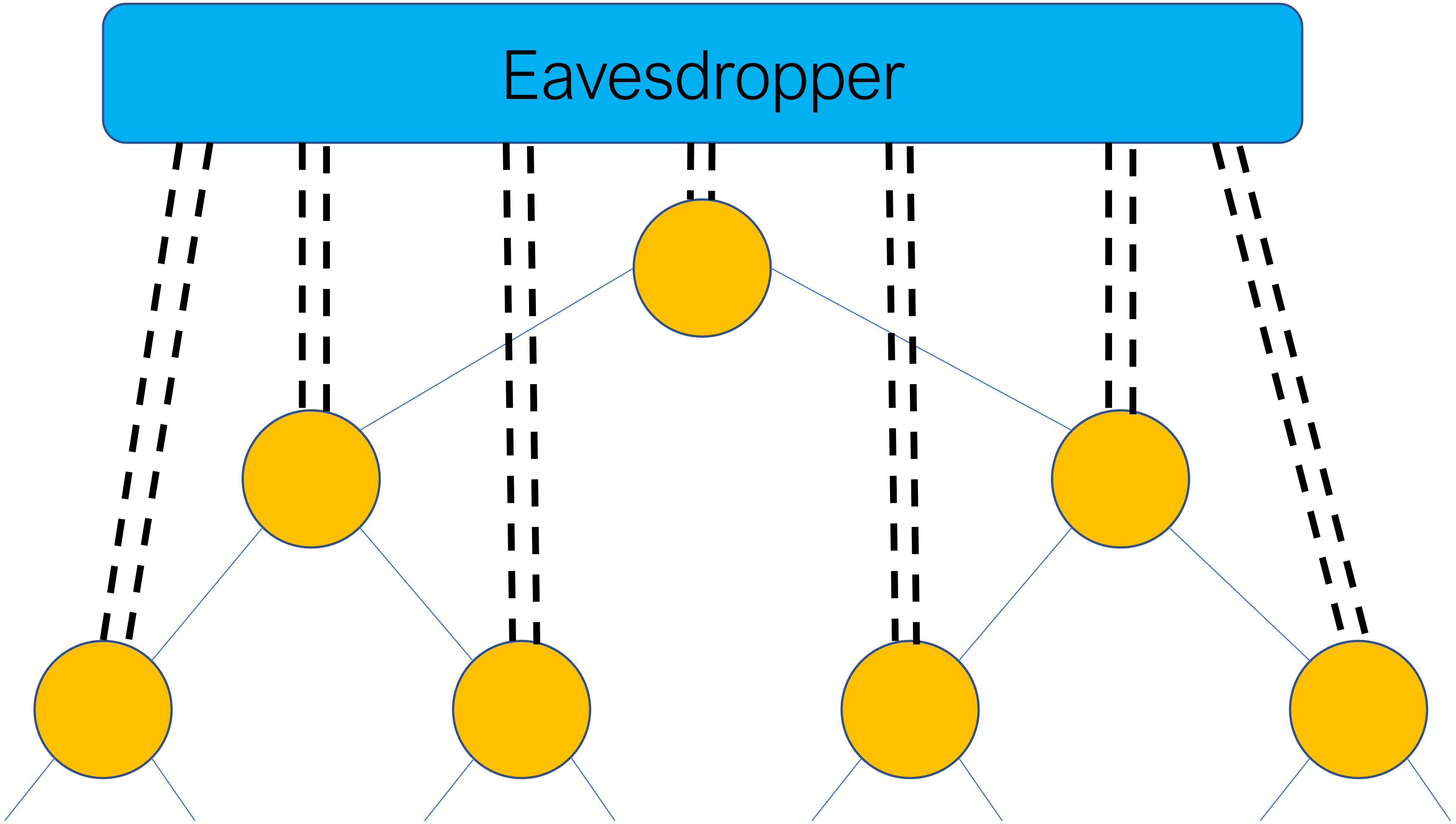


Does diffusion provide stronger anonymity than trickle spreading?

d-regular trees

Fraction of spies $p = 0$

Arbitrary number of connections θ



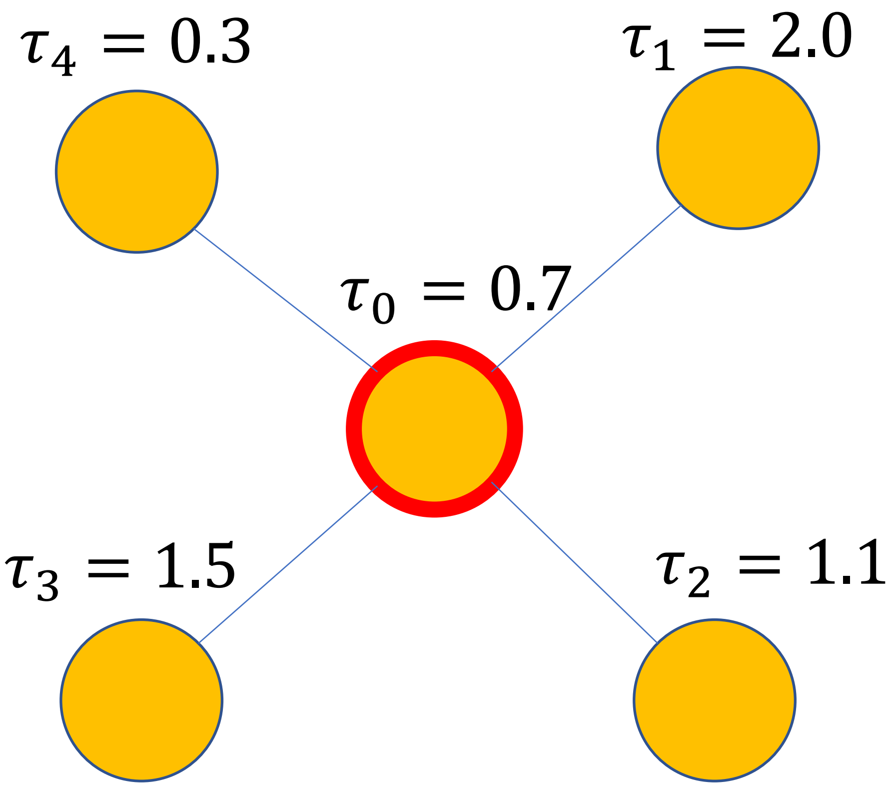
Anonymity Metric

timestamps

$$P(\text{detection}|\boldsymbol{\tau}, G)$$

graph

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \dots \\ \tau_n \end{bmatrix}$$

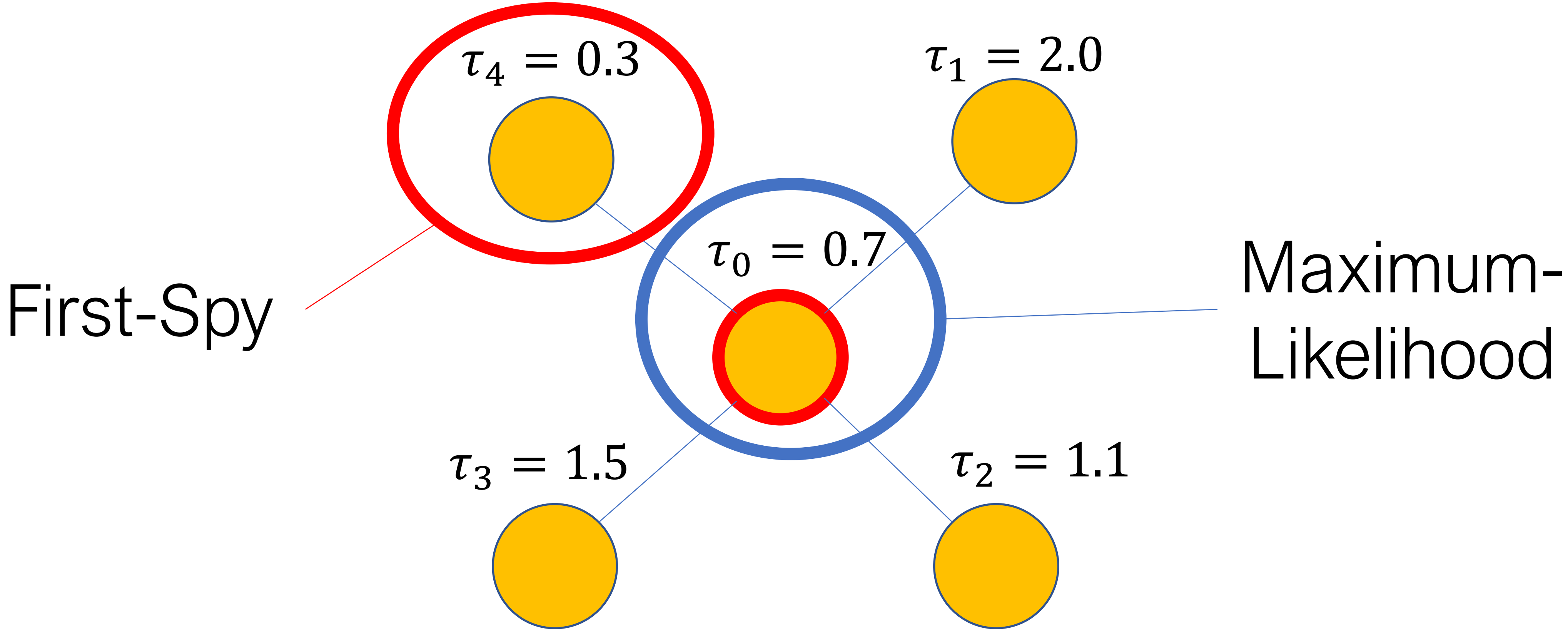


Estimators

timestamps

$$P(\text{detection}|\boldsymbol{\tau}, G)$$

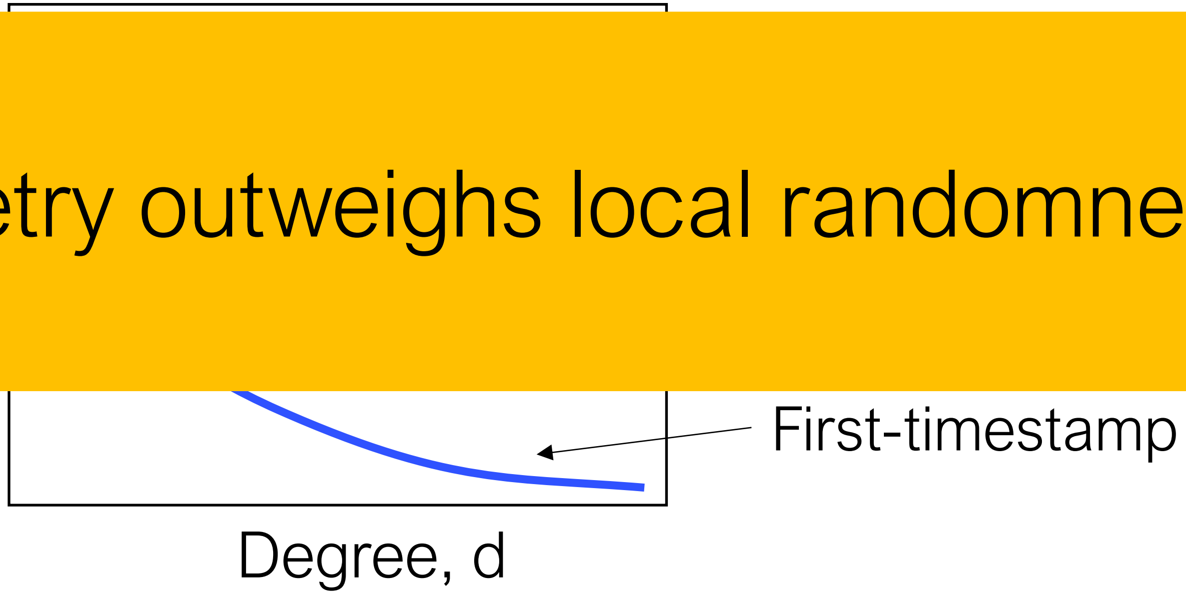
graph



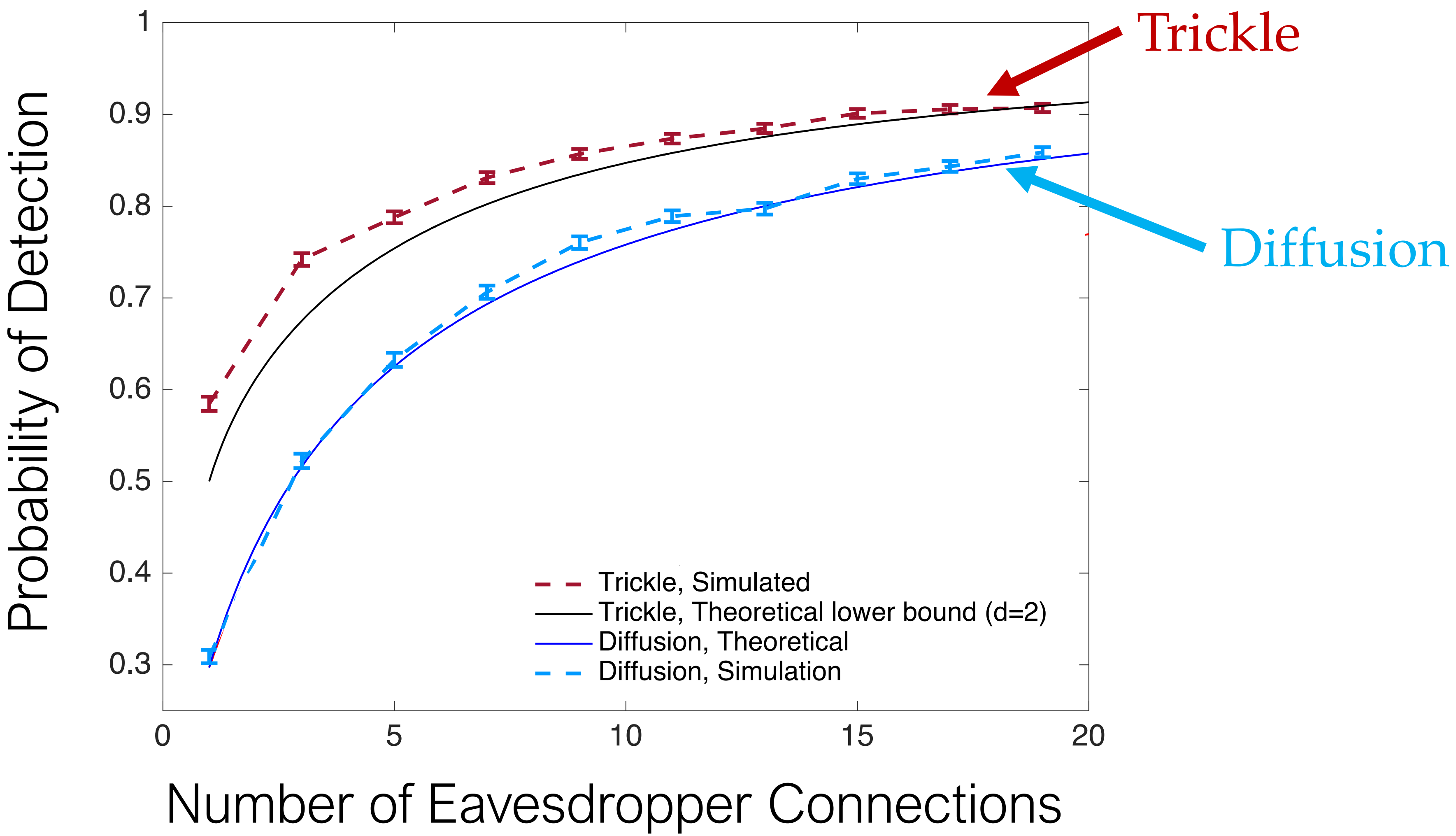
Detection Probability: d-Regular Trees

	Trickle	Diffusion
First-Timestamp	$O\left(\frac{\log d}{d}\right)$	$O\left(\frac{\log d}{d}\right)$
Maximum-Likelihood	$\Omega(1)$	$\Omega(1)$

Intuition: Symmetry outweighs local randomness!



Results: Bitcoin Graph

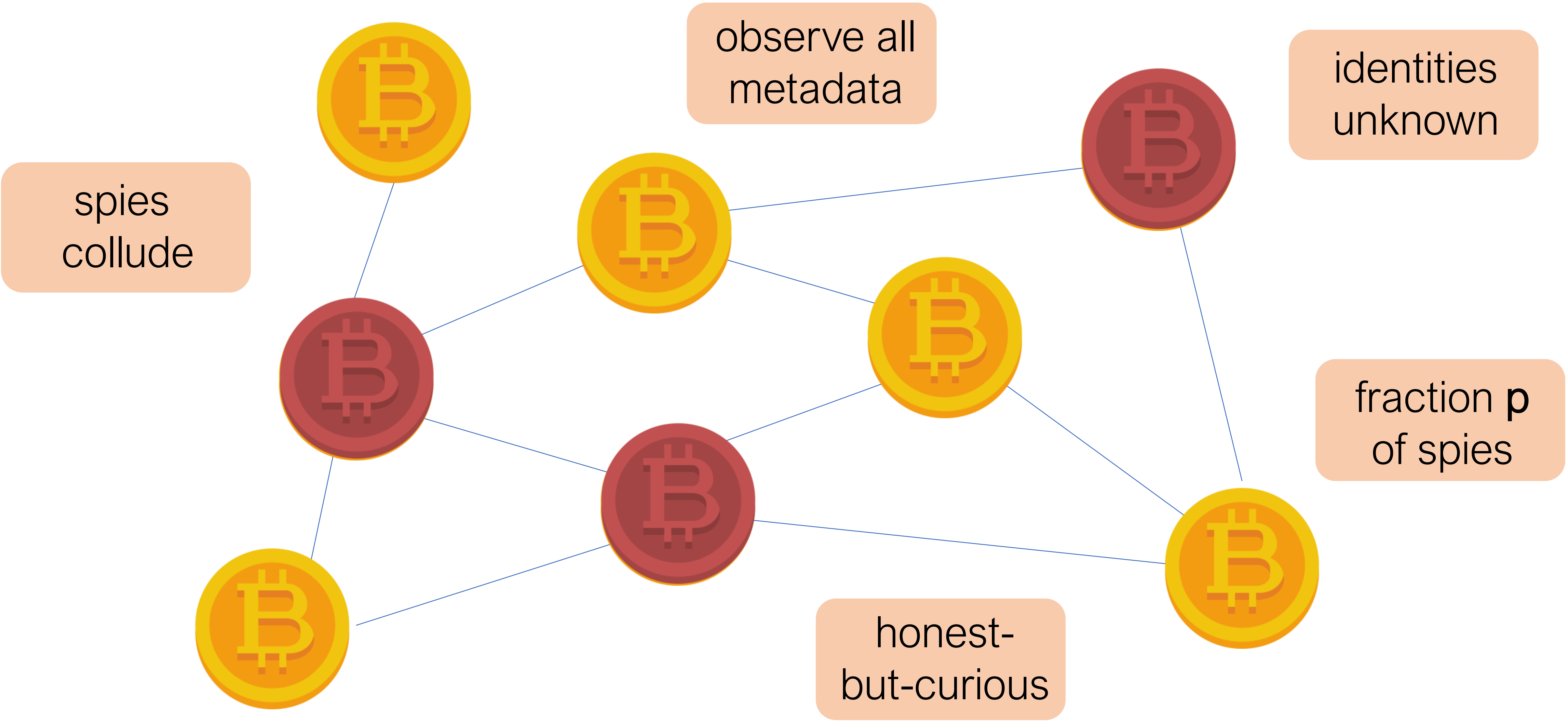


Diffusion does not have (significantly) better anonymity properties than trickle.

Redesign

Can we design a better network?

Botnet adversarial model



Metric for Anonymity

Recall

$$\frac{1}{n}\sum_{v} 1\{M(v'\text{s tx}) = v\}$$

Mapping

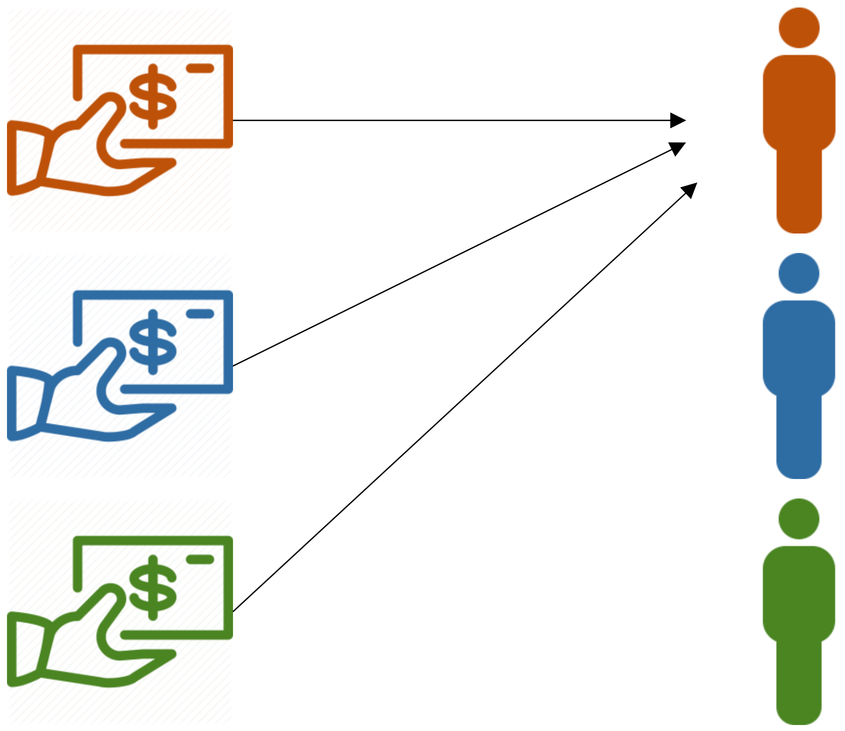
Number honest users

User

$\mathbb{E}[\text{Recall}] =$ Probability of Detection

Transactions

Users



Mapping M

Precision

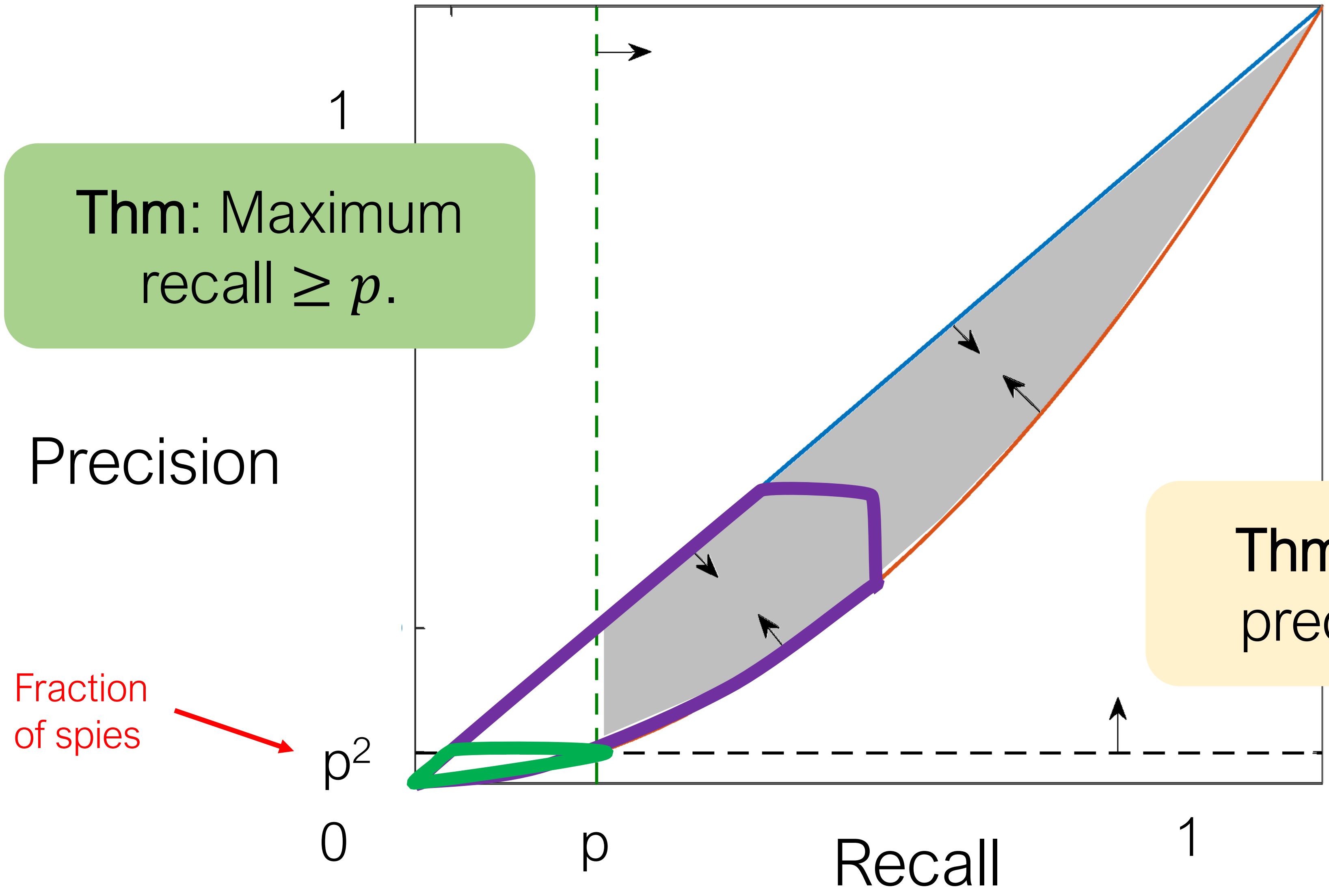
$$\frac{1}{n}\sum_{v} \frac{1\{M(v'\text{s tx}) = v\}}{\#\ \text{tx mapped to v}}$$

Goal

Design a distributed flooding protocol that minimizes the maximum precision and recall achievable by a computationally-unbounded adversary.

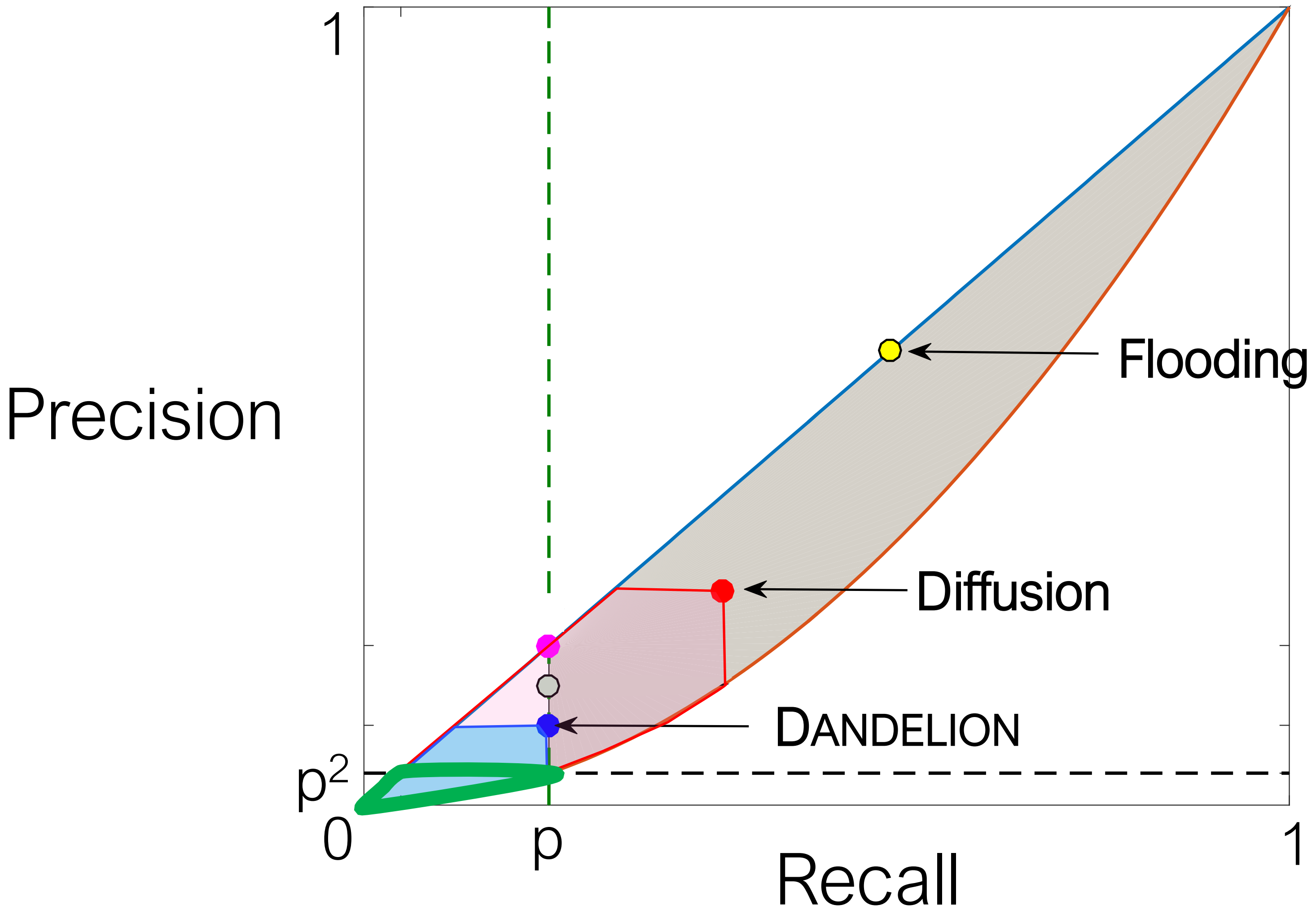
Fundamental Limits

Thm: Maximum recall $\geq p$.



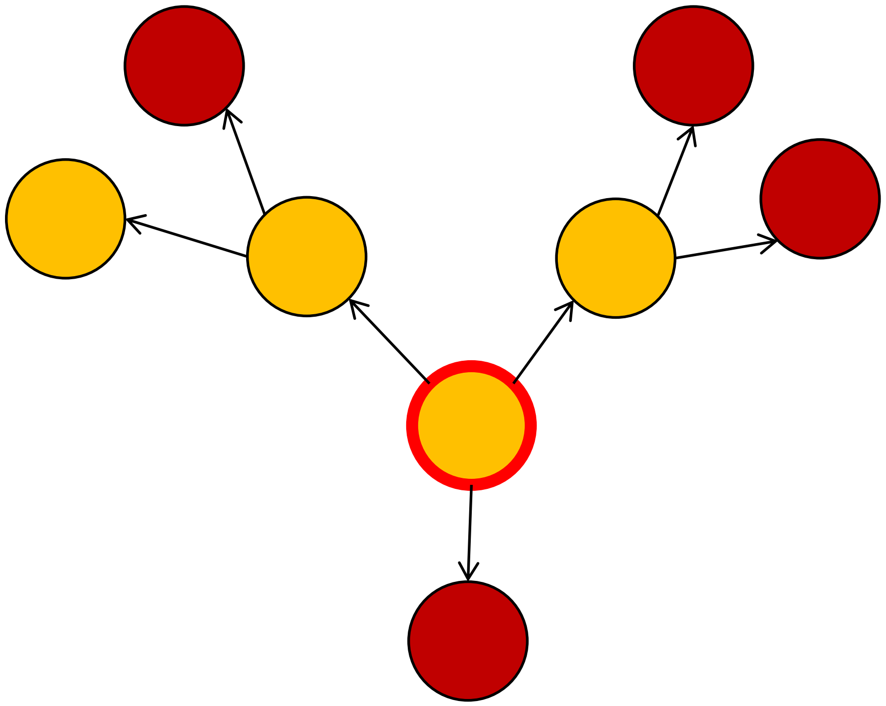
Thm: Maximum precision $\geq p^2$.

Performance: Achievable Region

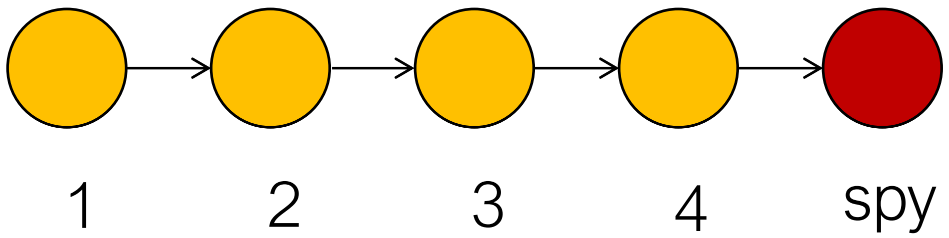


What are we looking for?

Asymmetry

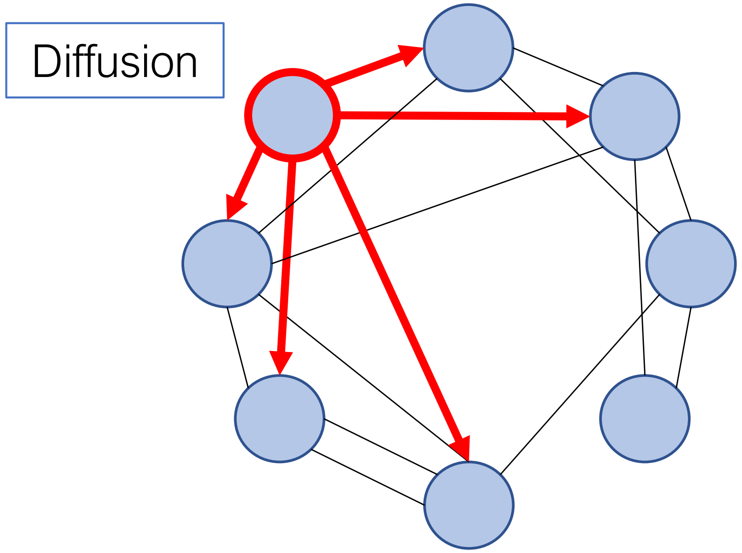


Mixing



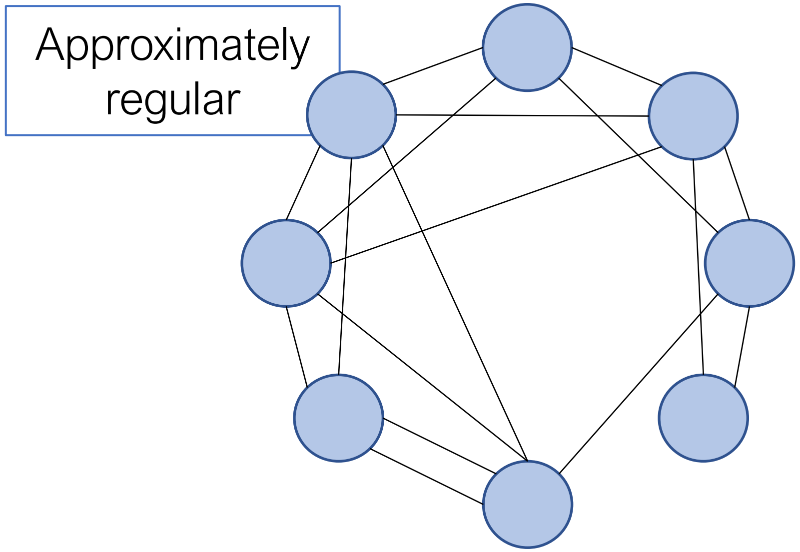
What can we control?

Spreading Protocol



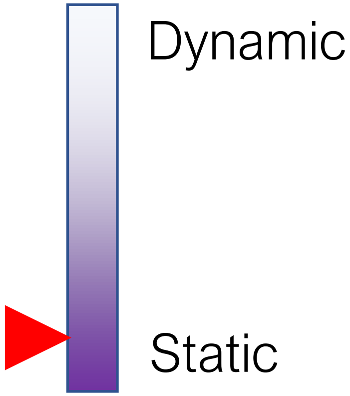
Given a graph, how do we spread content?

Topology



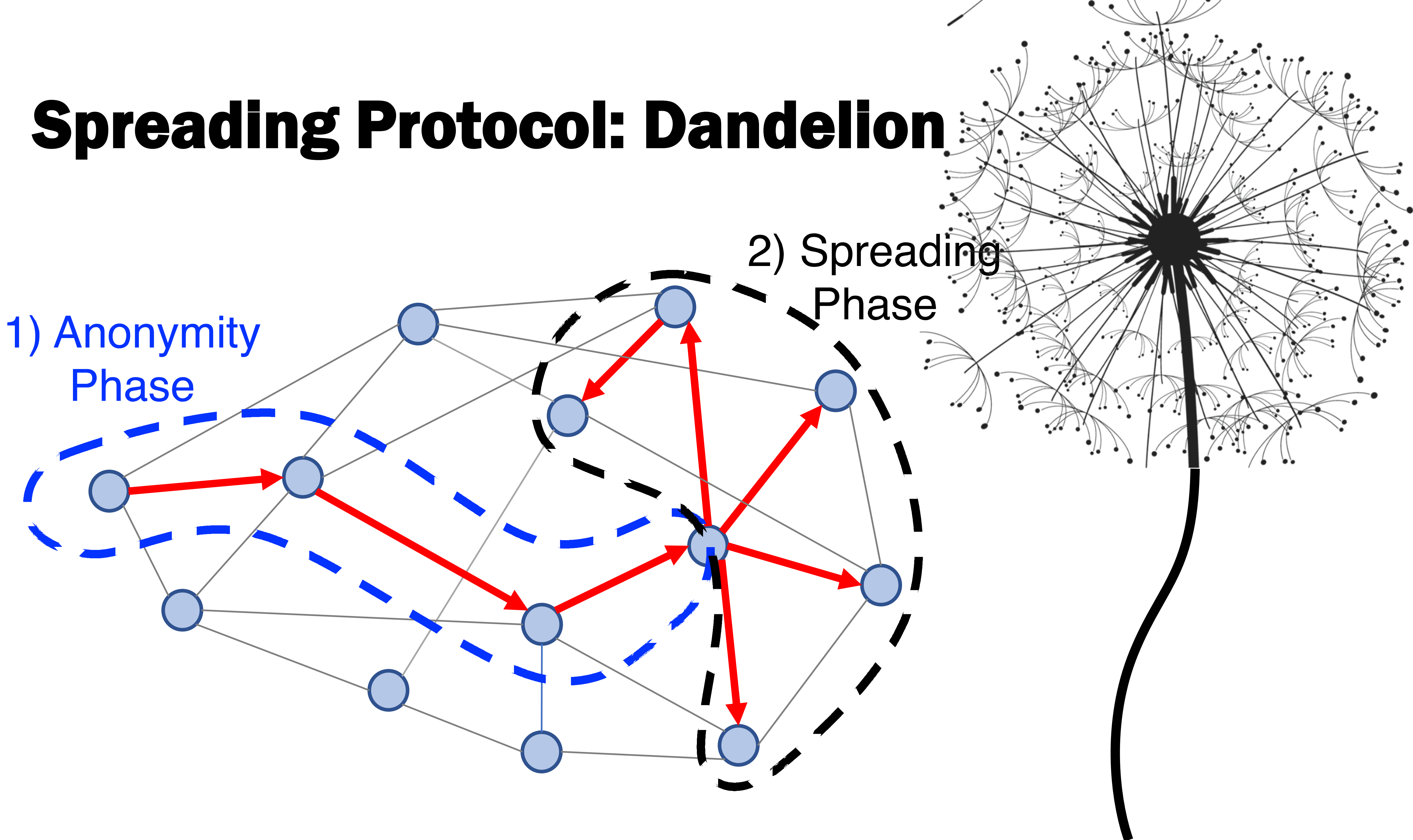
What is the underlying graph topology?

Dynamicity



How often does the graph change?

Spreading Protocol: Dandelion



Why Dandelion spreading?

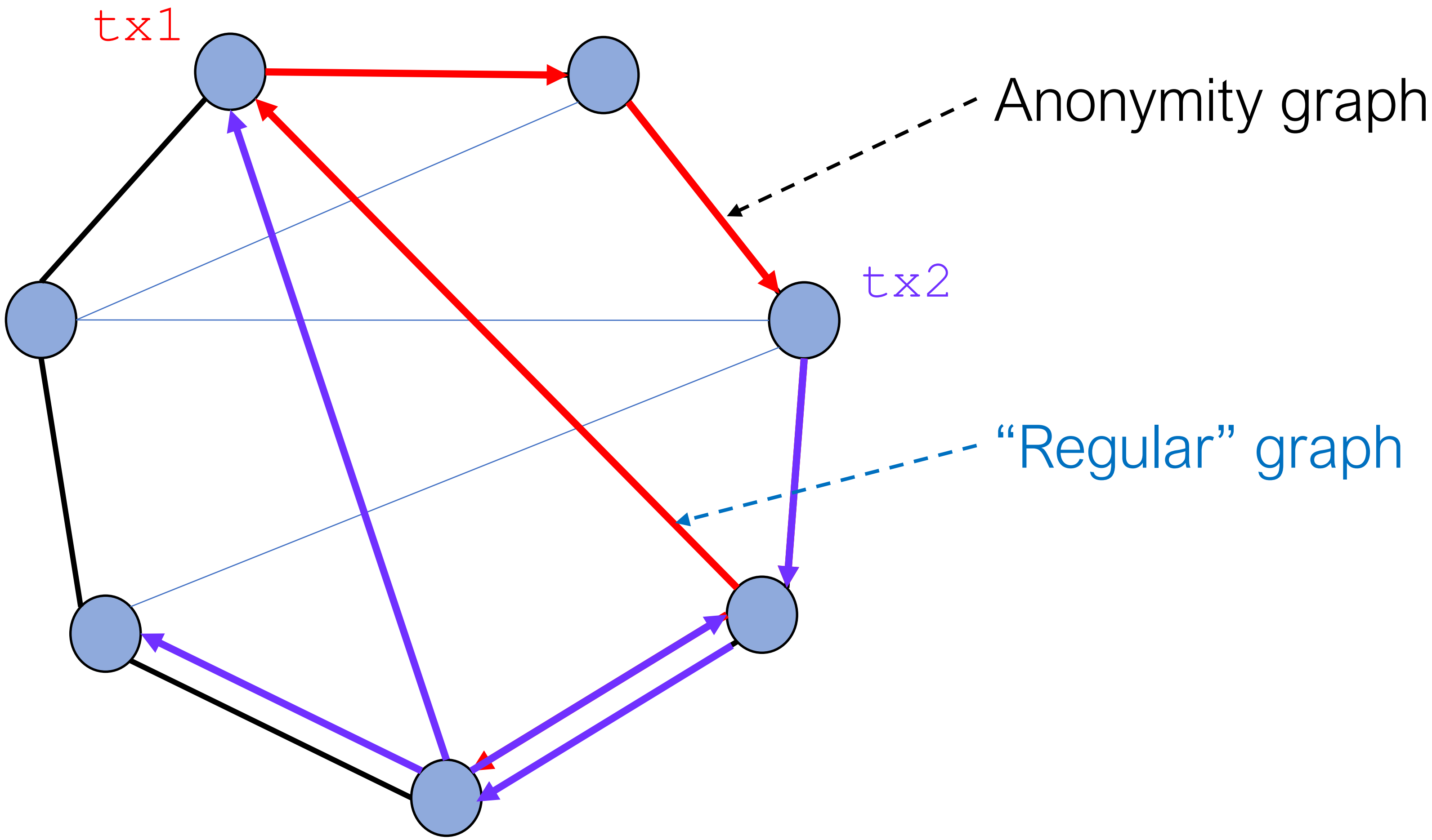
Theorem: Dandelion spreading has an optimally low maximum recall of $p + O\left(\frac{1}{n}\right)$.

Theorem: Fundamental lower bound = p

fraction of spies

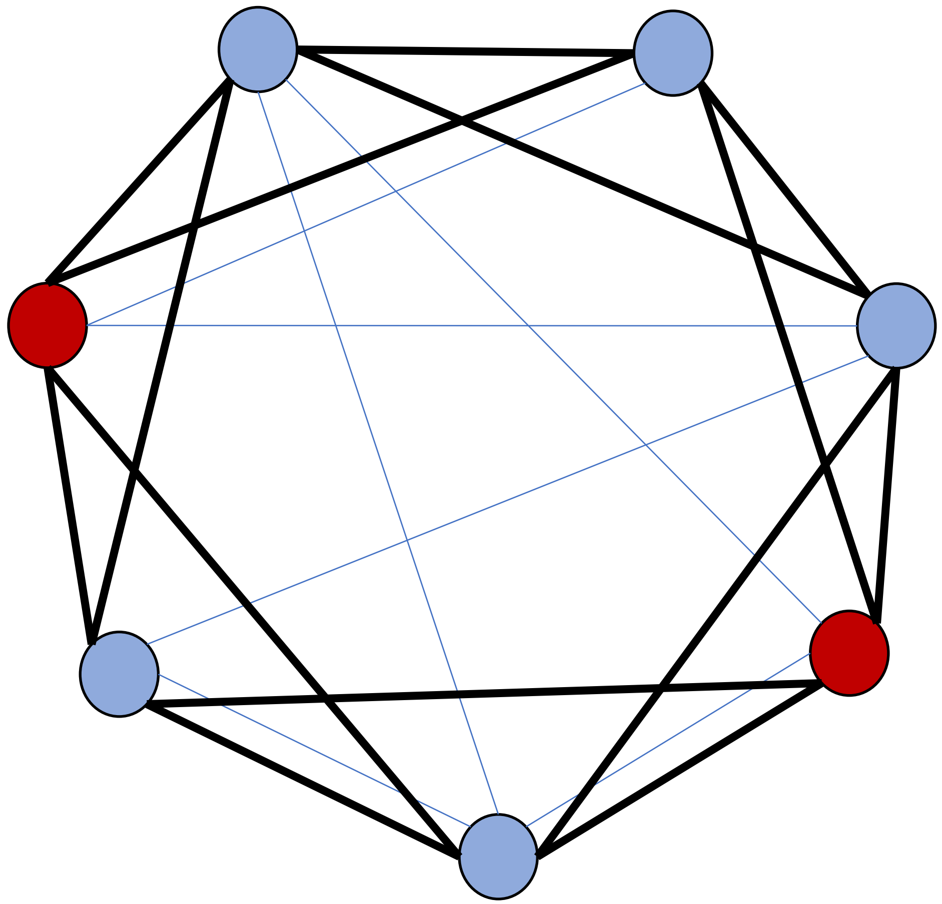
number of nodes

Graph Topology: Line



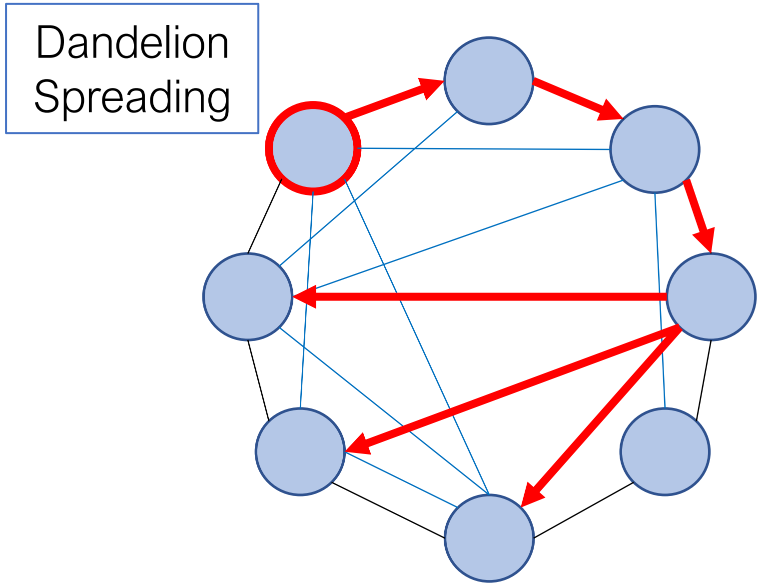
Dynamicity: High

Change the anonymity graph frequently.



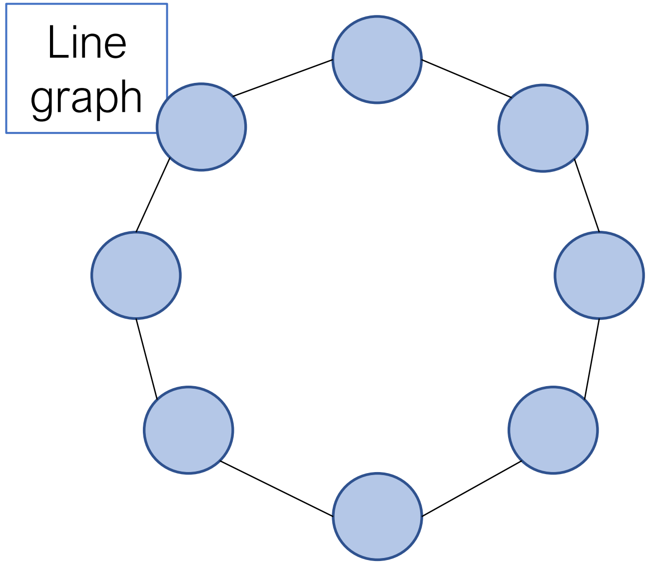
DANDELION Network Policy

Spreading Protocol



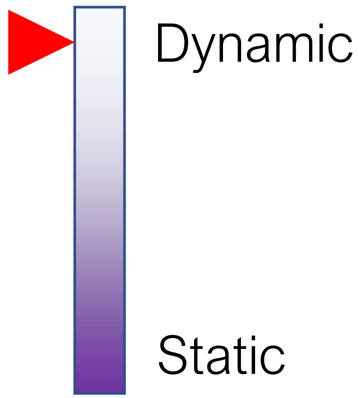
Given a graph, how do we spread content?

Topology



What is the anonymity graph topology?

Dynamicity



How often does the graph change?

Theorem: Fundamental lower bound = p^2

Theorem: DANDELION has a nearly-optimal maximum precision of $\frac{2p^2}{1-p}\log\left(\frac{2}{p}\right) + O\left(\frac{1}{n}\right).^*$

fraction of spies

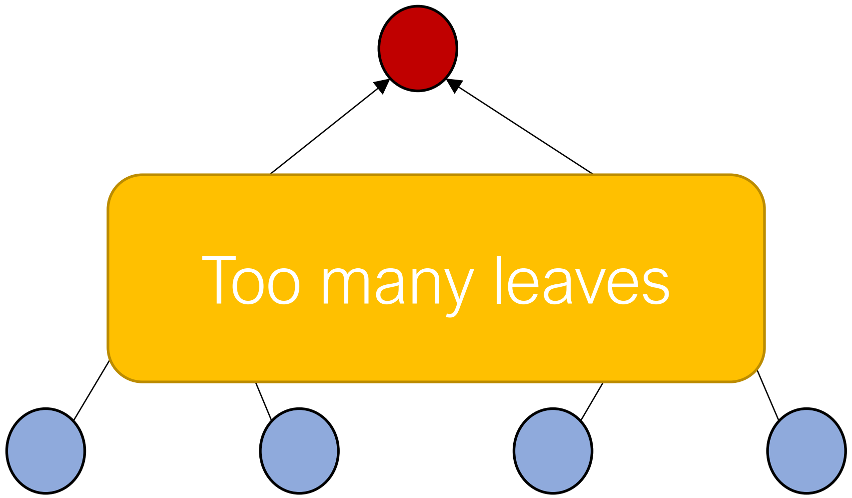
number of nodes

*For $p < \frac{1}{3}$

Why is DANDELION good?

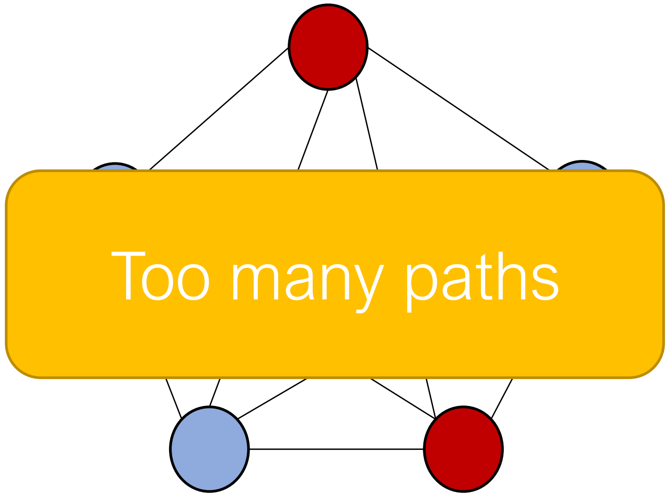
Strong mixing properties.

Tree



Precision: $O(p)$

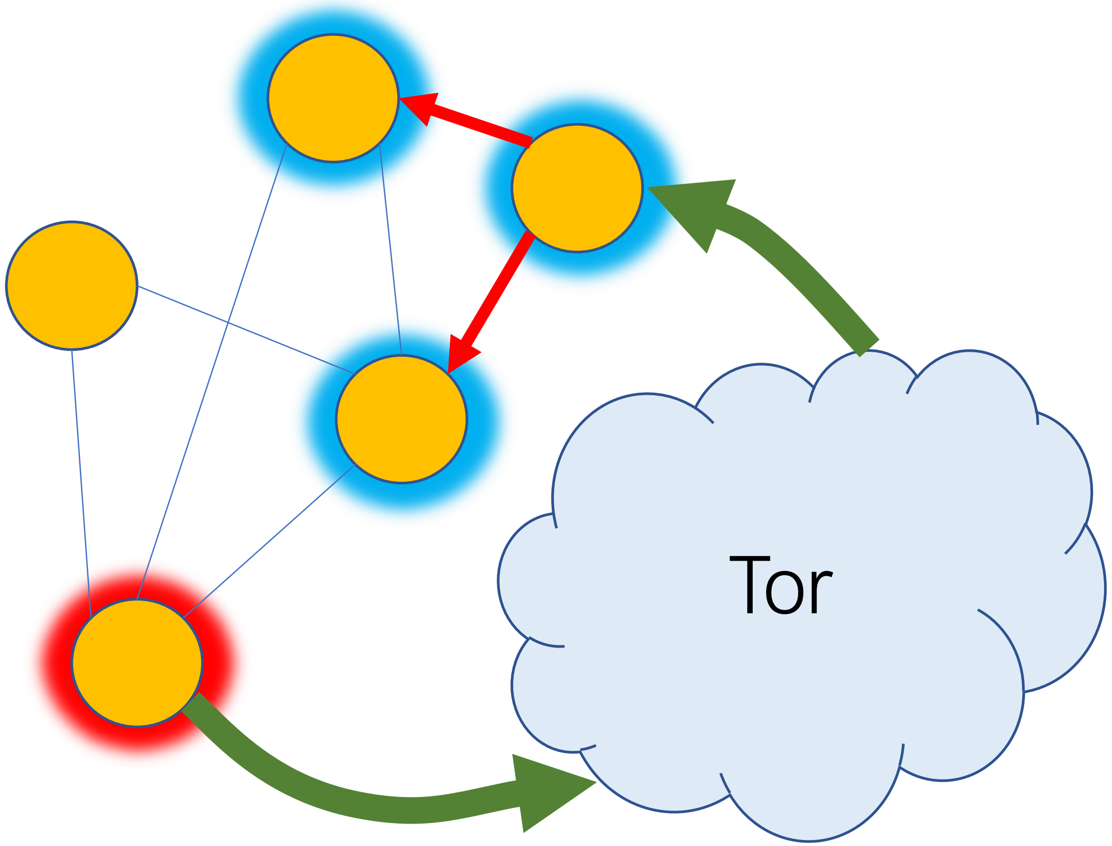
Complete graph



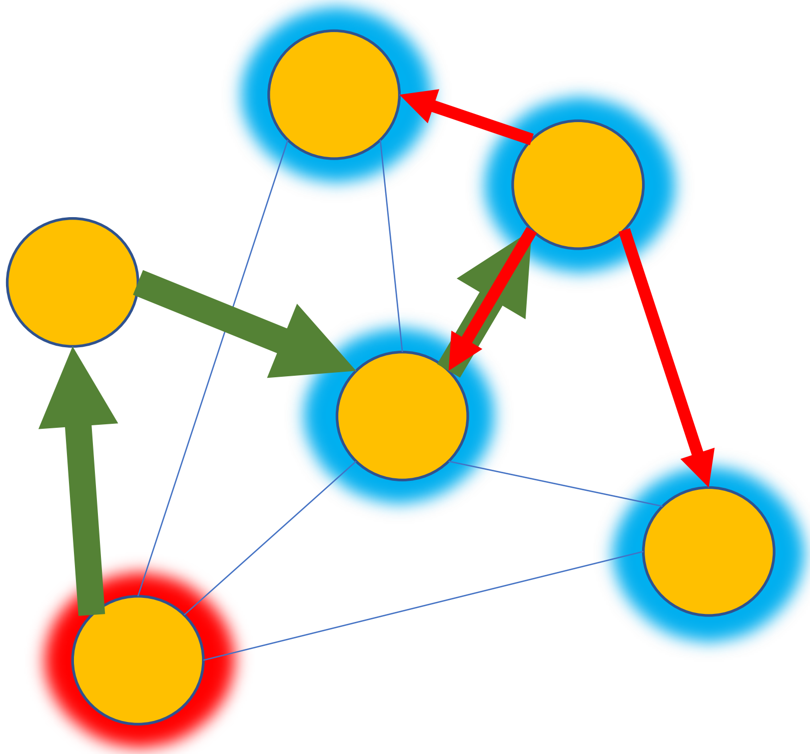
Precision: $\frac{p}{1-p}(1 - e^{p-1})$

Alternative solutions

Connect through Tor



I2P Integration (e.g. Monero)



Narayanan and Möser, 2017

Take-Home Messages

1) Bitcoin's P2P network has poor anonymity.

2) Moving from trickle to diffusion did not help.

3) DANDELION is a lightweight privacy solution for certain classes of adversaries.

4) We will revisit modern network design in light of frontrunning attacks

Attendance : NFT Drop

https://poap.website/dog-director-become

- Mint token to Metamask.
- Submit tx hash for attendance claim
- Instructions in Ed pinned posts.