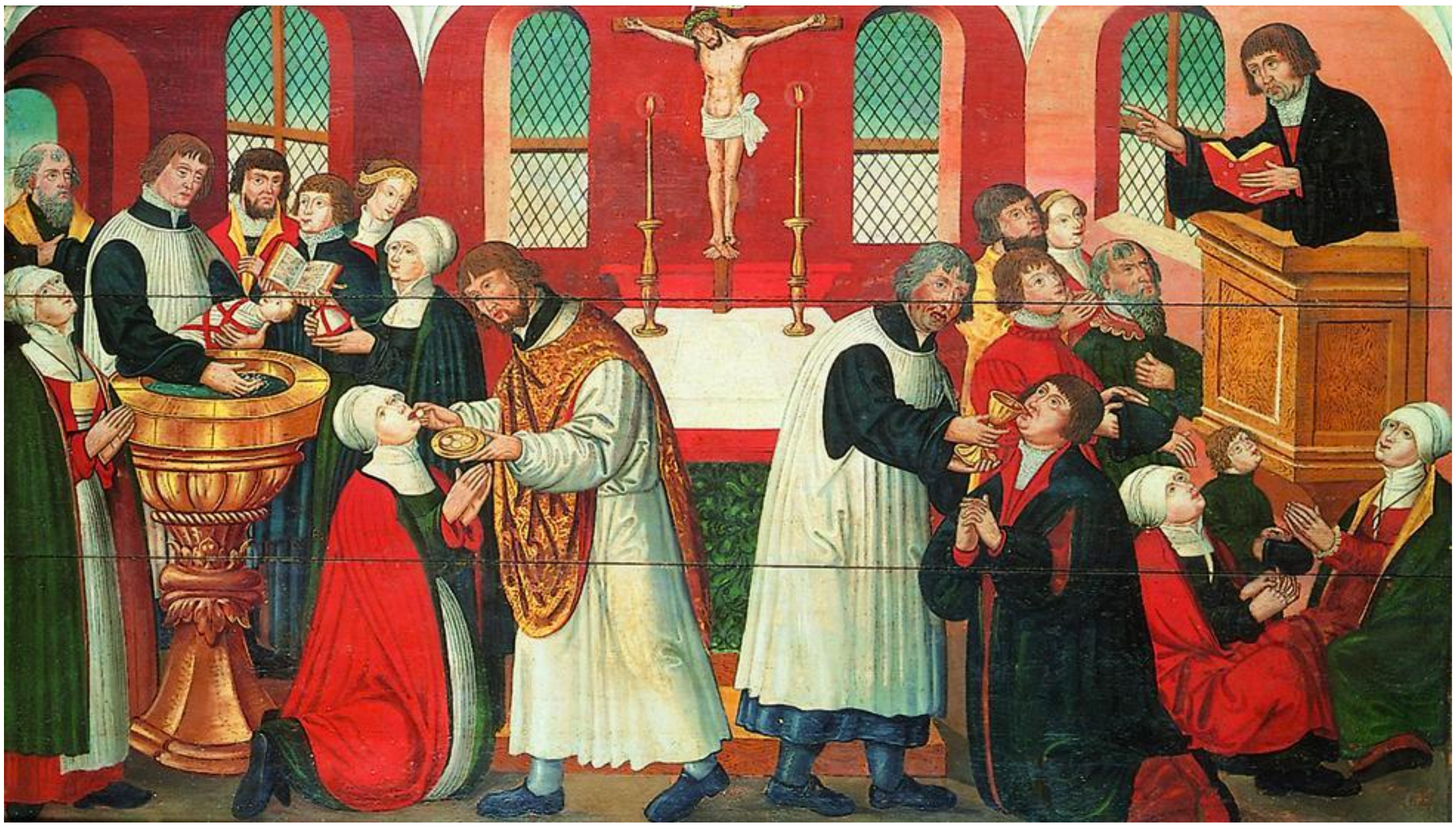
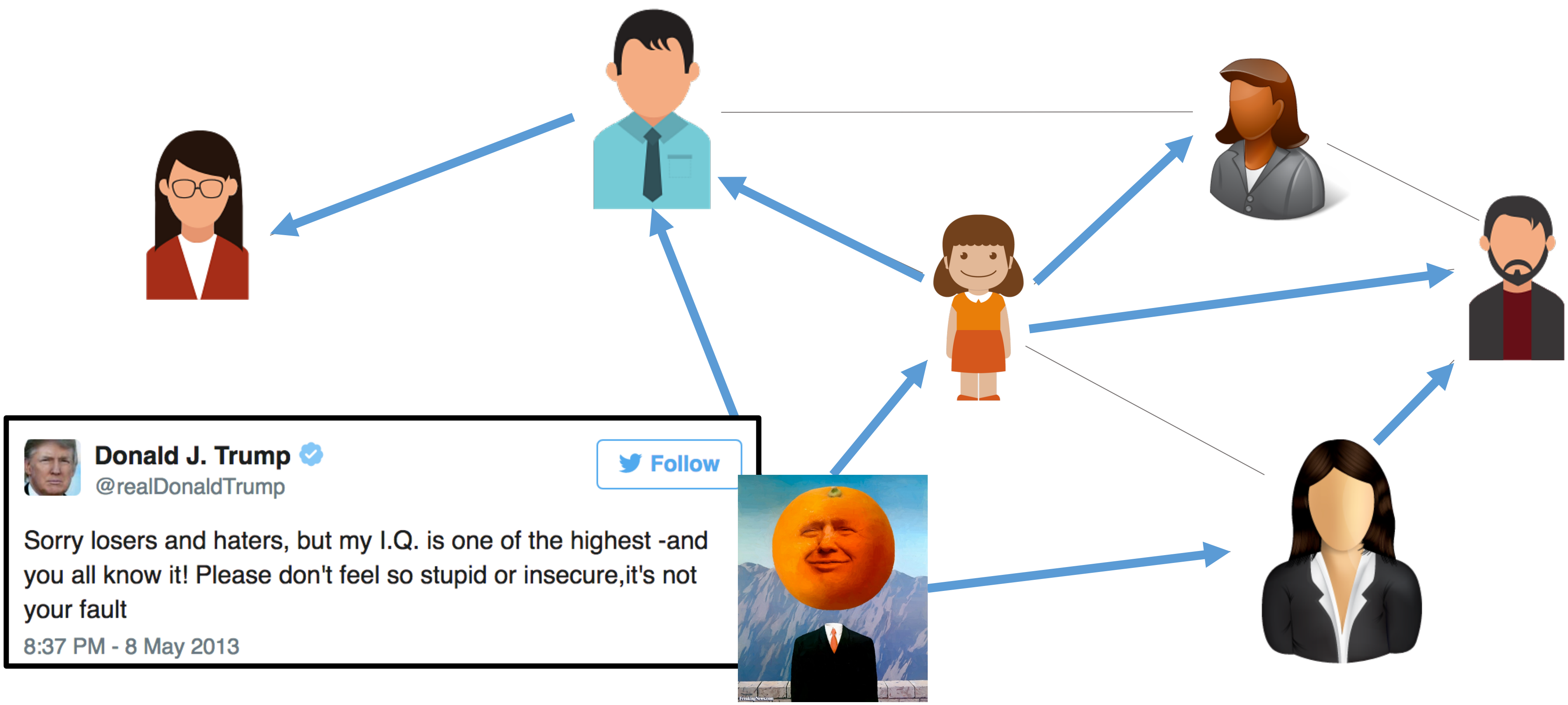
Finding and Hiding Message Sources in Networks:

Epidemics, Social Media, Cryptocurrencies

Broadcasting Information: Then



Broadcasting Information: Now



Broadcast communication is easier, cheaper, and **more democratic** than ever before.

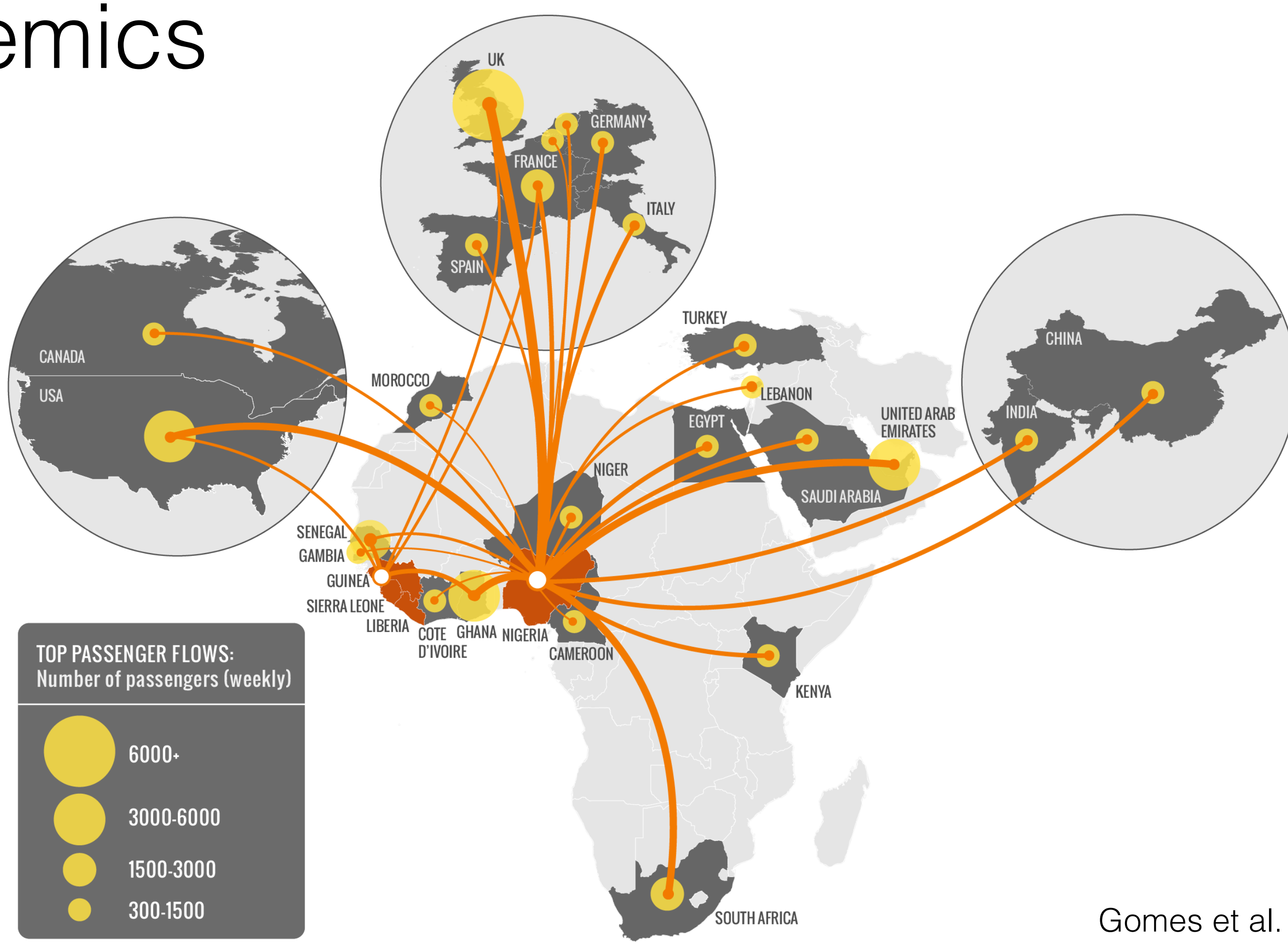
Distributed broadcasting

Epidemics

Social Networks

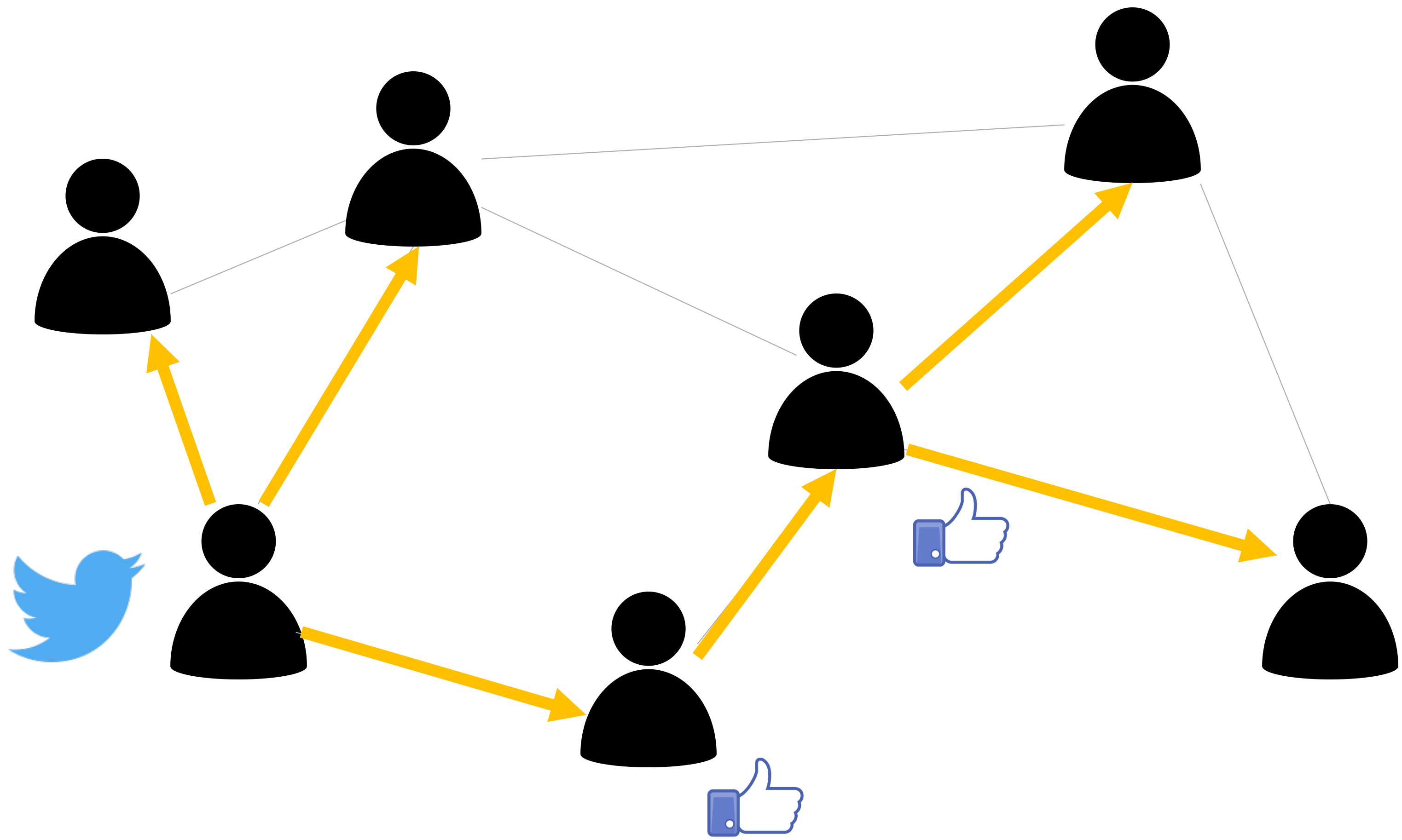
Cryptocurrencies

Epidemics

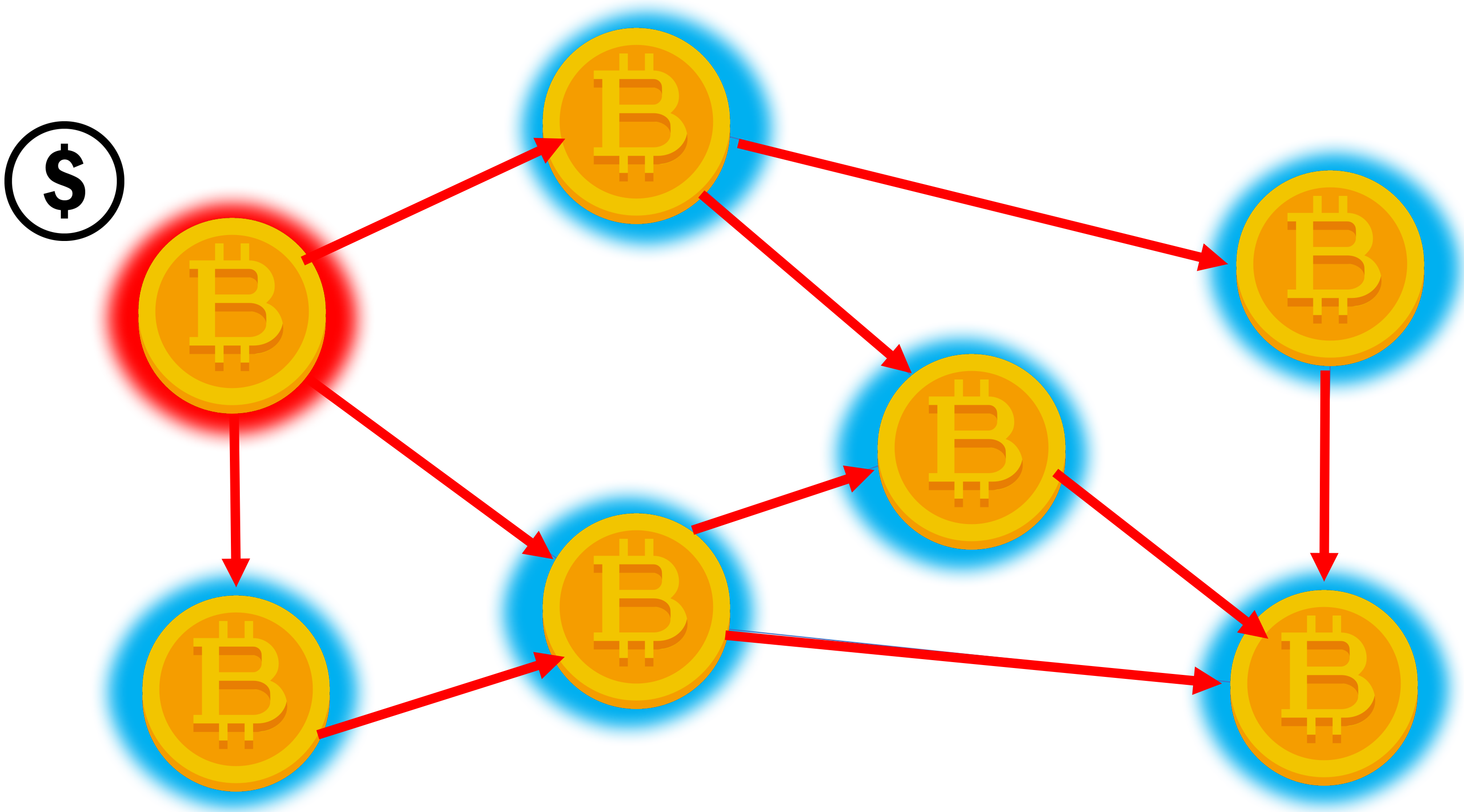


Gomes et al. 2014, PLOS

Social Networks



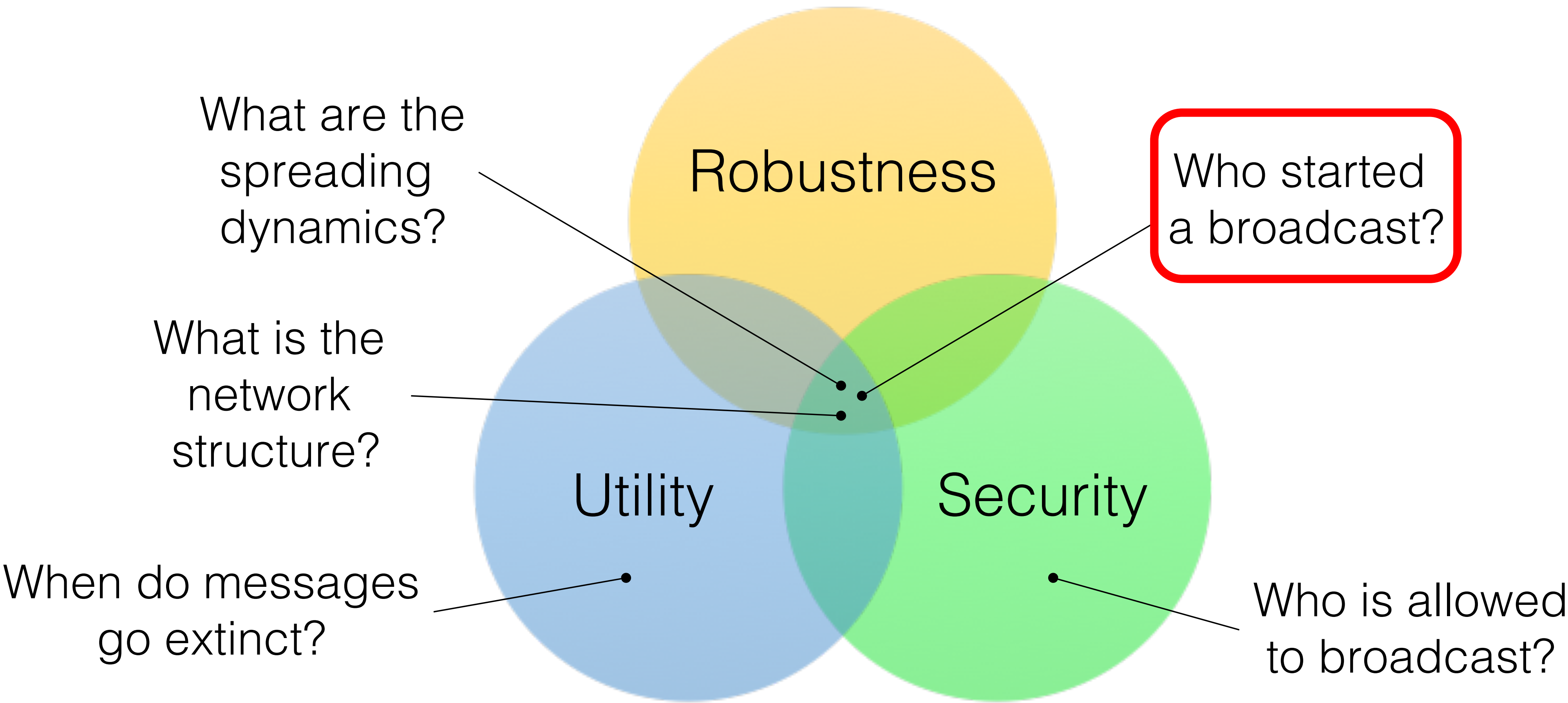
Cryptocurrencies



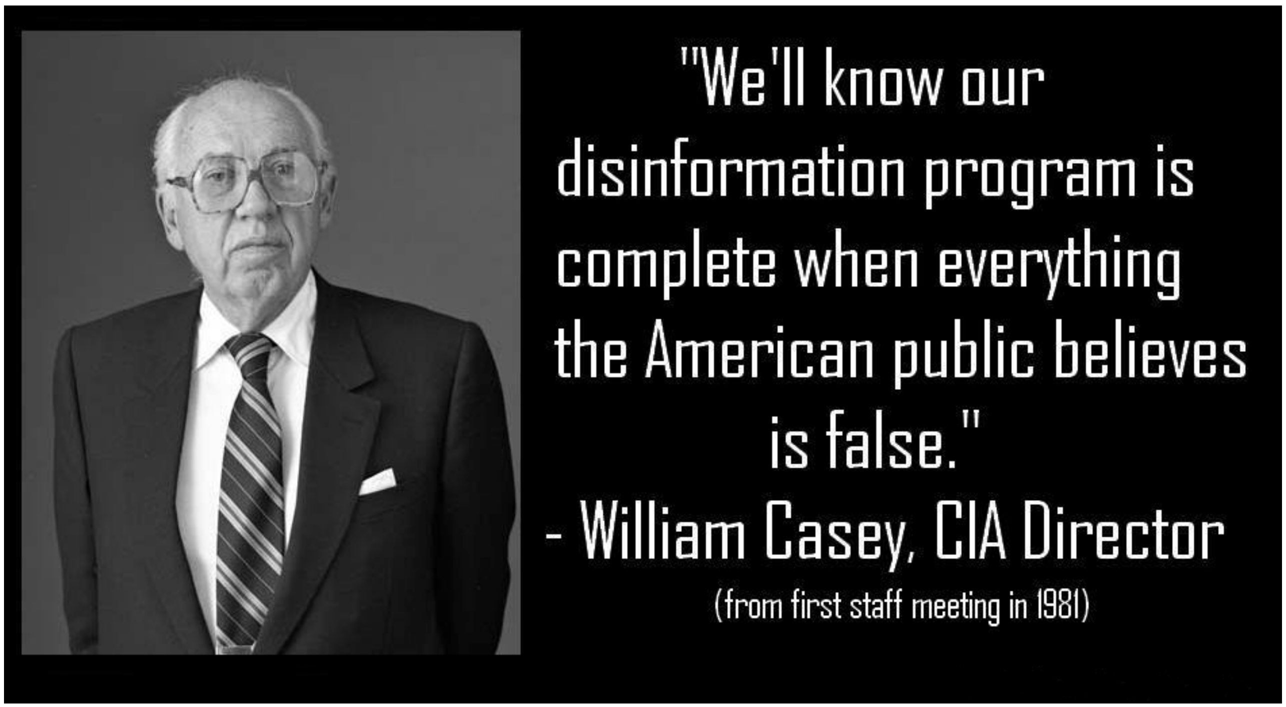
Broadcasting can impact the robustness, utility, and security of a network.

… but distributed network management poses new challenges!

Relevant Questions



Attribution is central to communication



This talk

- **Part I**: Systems and how to model them (1 hr)
  - Bitcoin primer (30 min)
  - Network models
  - Propagation models
  - Observation models
- **Part II**: Source finding (1 hr)
  - Algorithms for source detection
  - Analysis of these algorithms
  - Open problems
- **Part III**: Source hiding (1 hr)
  - Early results: crypto community
  - Statistical approaches
  - Open problems

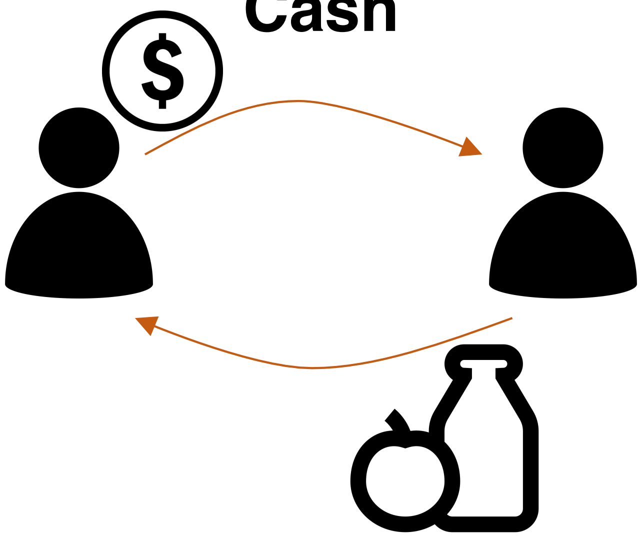
Cryptocurrencies Primer

The Origin of Bitcoin

Narayanan et al., *Bitcoin and Cryptocurrency Technologies,* 2016

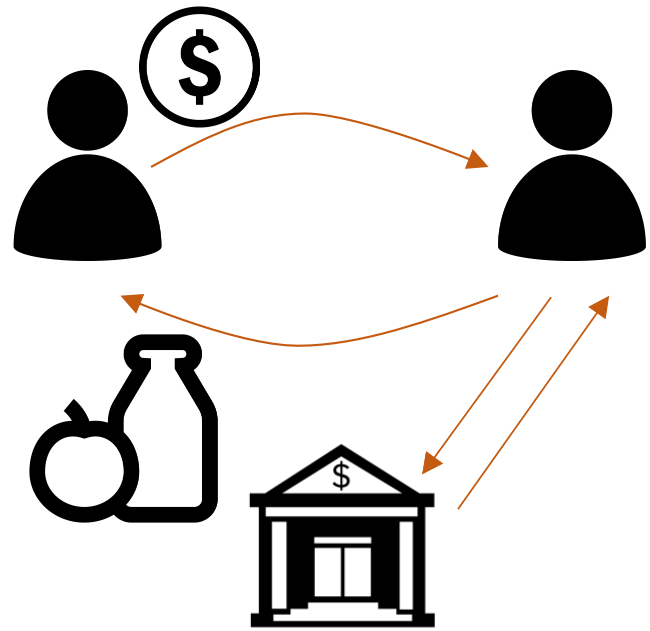
Financial systems

Cash



+ Offline transactions
+ Anonymous
- Requires initial seed cash

Credit



+ Exchanges can be digital
- Parties take on risk

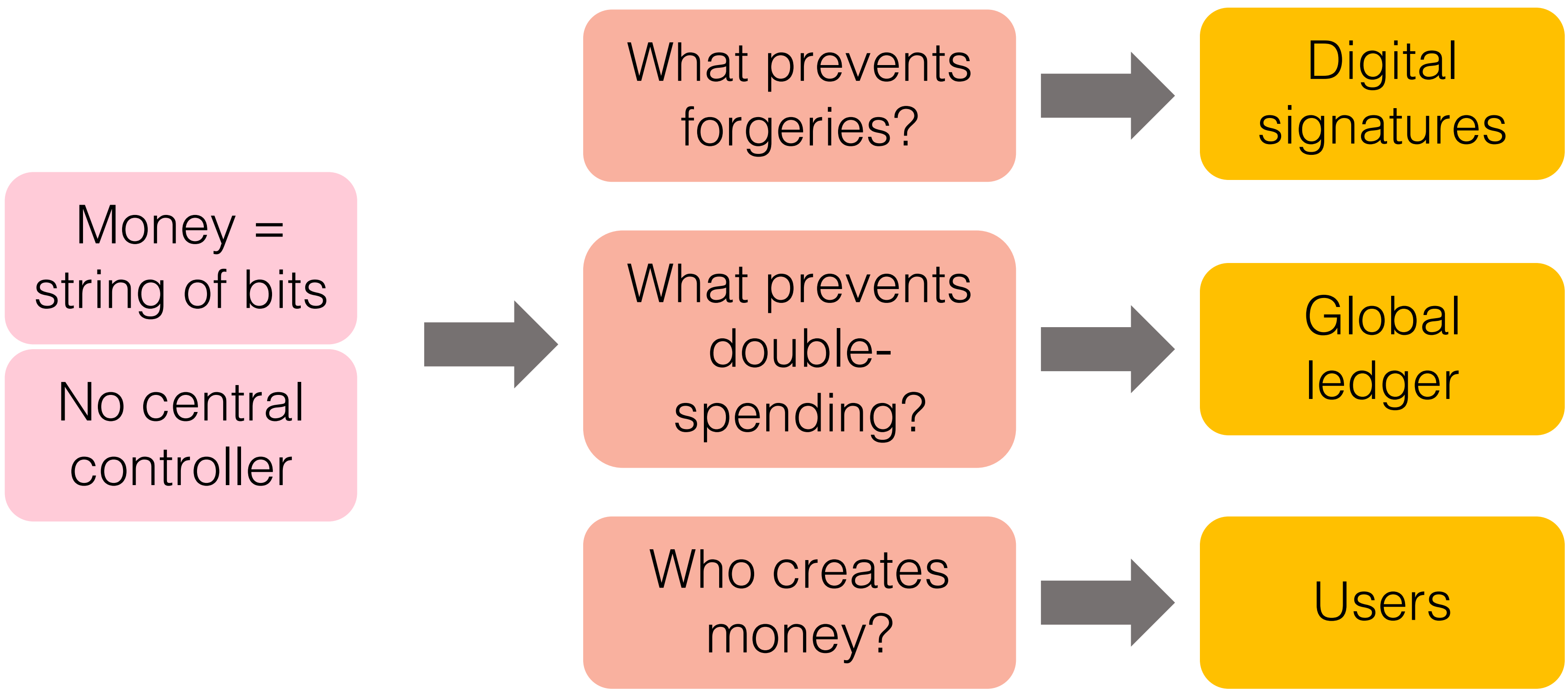
Bitcoin Objectives

- **Egalitarianism** → no central trusted party
- **Transparency** → transactions can be verified by all nodes
- **Privacy** → users need not reveal their identity to the currency

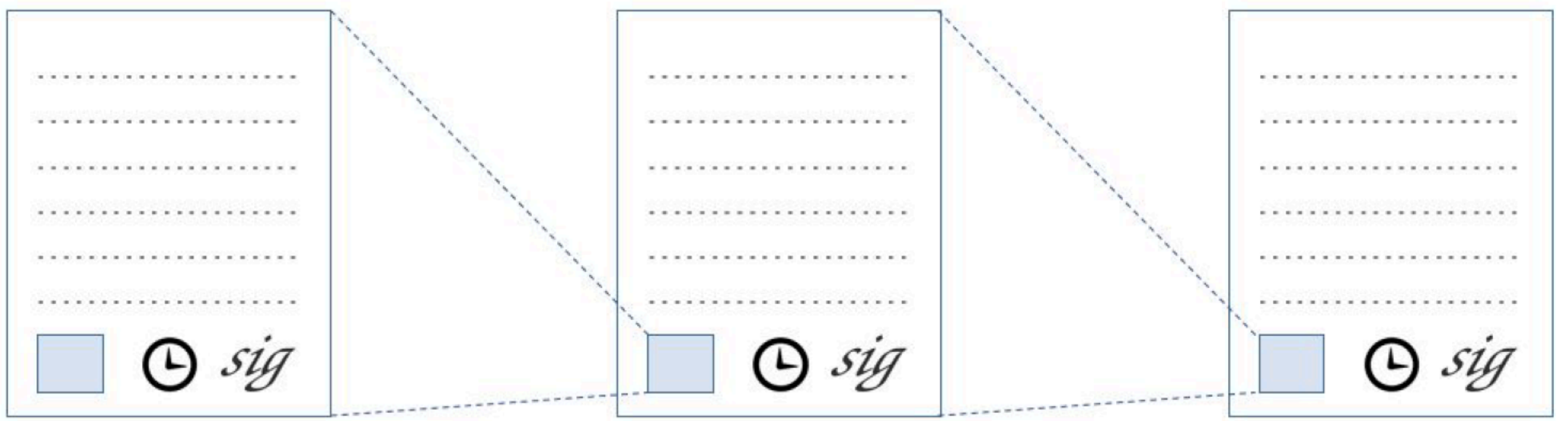
Bitcoin objectives

	Credit	Cash
Egalitarianism	✗	✗
Transparency	✗	✓
Privacy	✗	✓

Why this problem is hard



Append-only ledgers



Haber and
Stornetta, 1991

Image from Narayanan et al, 2016

Hierarchical structure

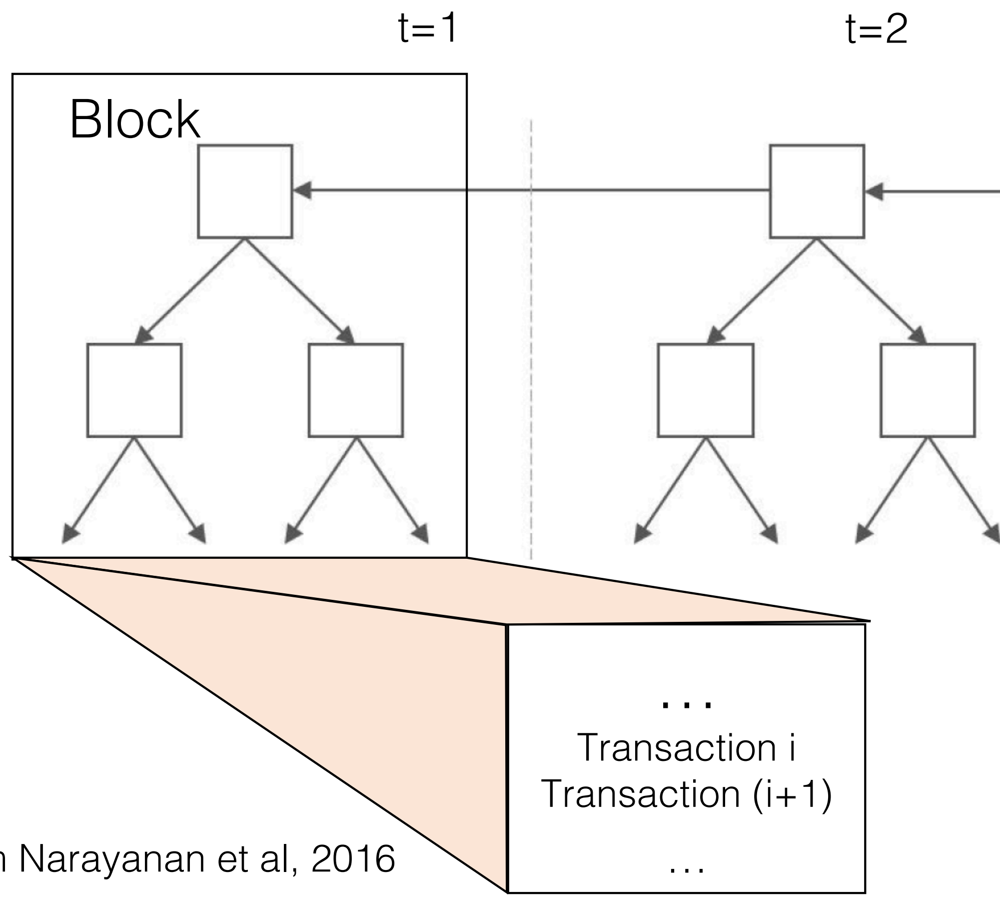
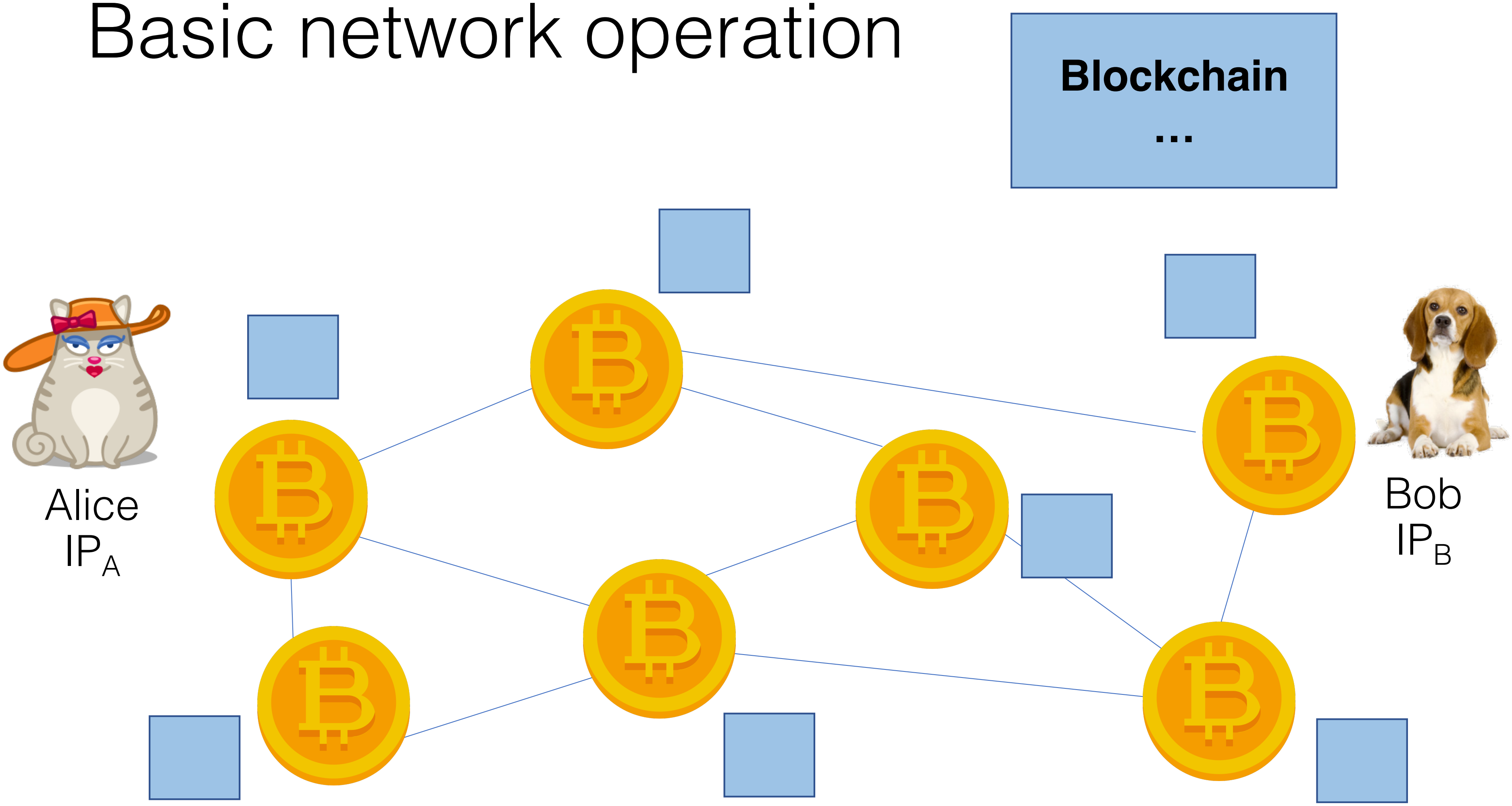


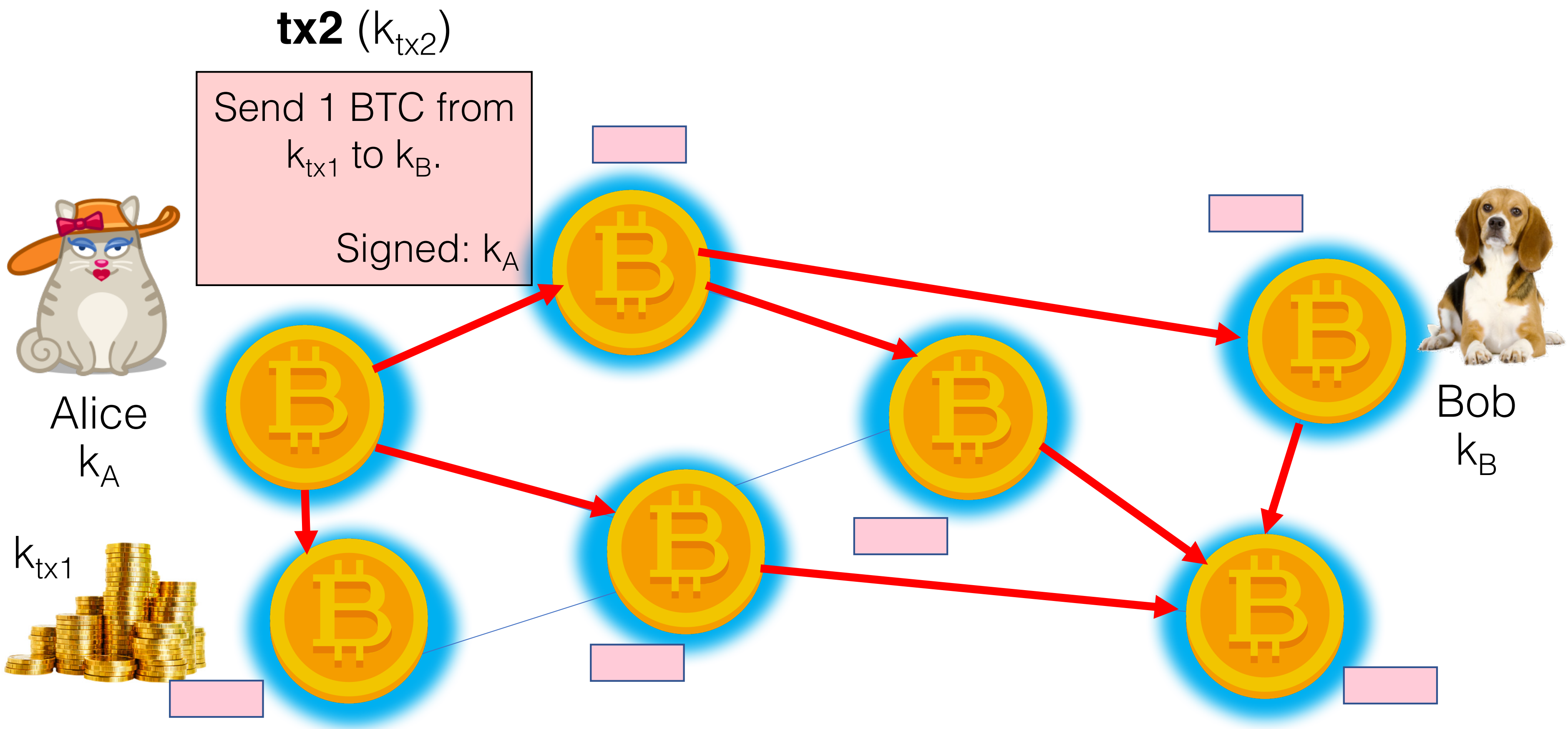
Image from Narayanan et al, 2016

Merkle Tree

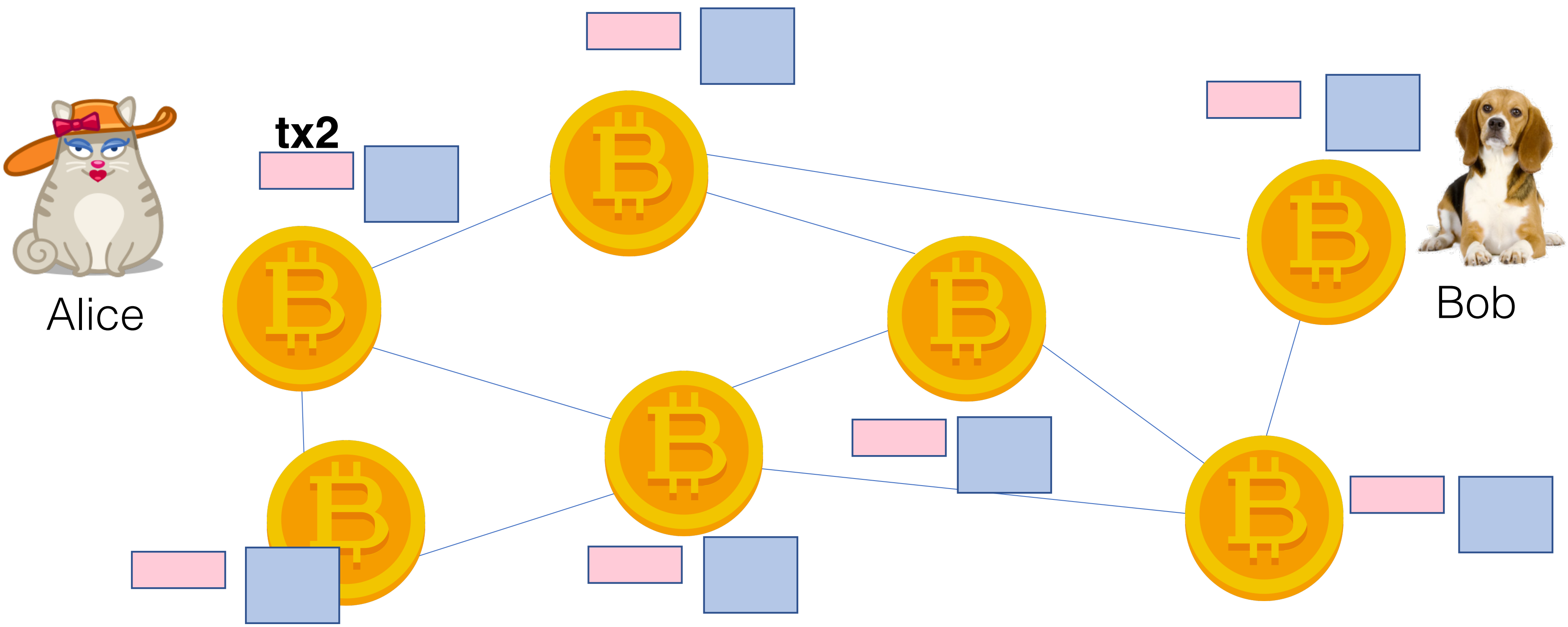
Basic network operation



Basic network operation



Adding to the Blockchain



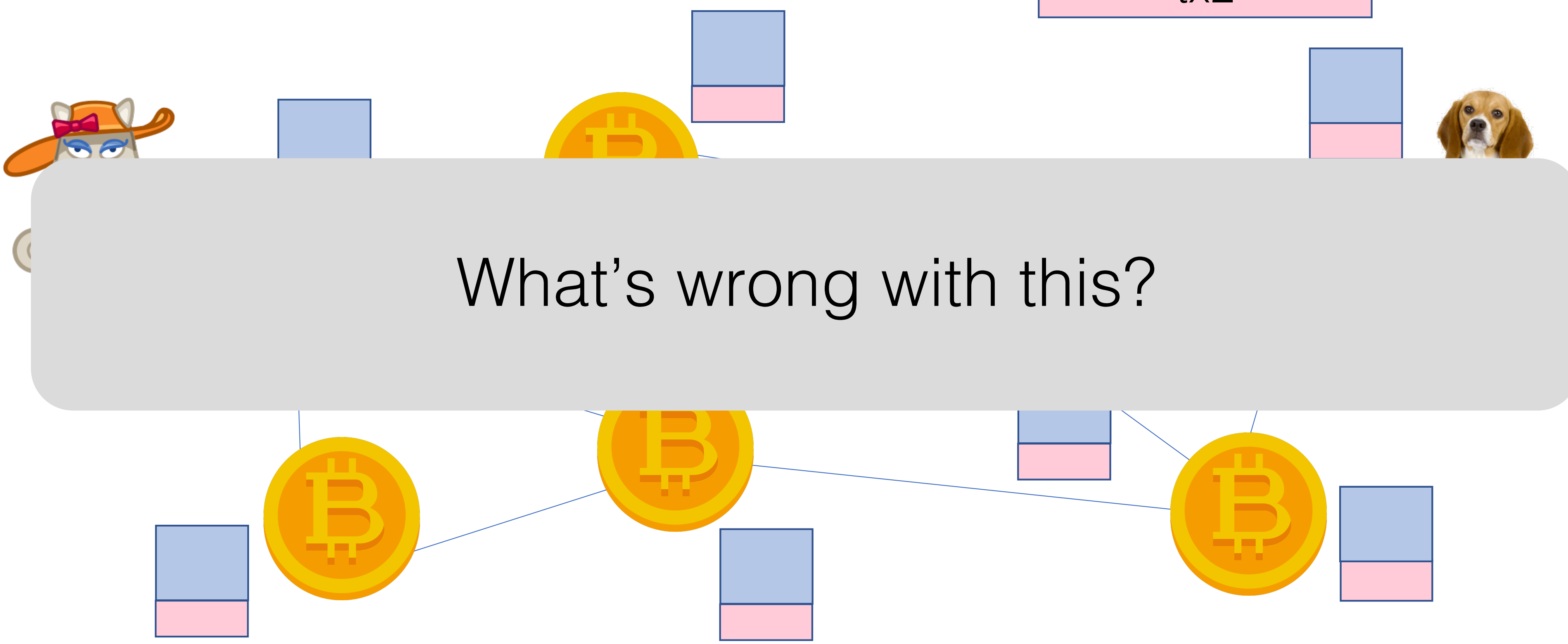
Adding to the Blockchain

Blockchain

…

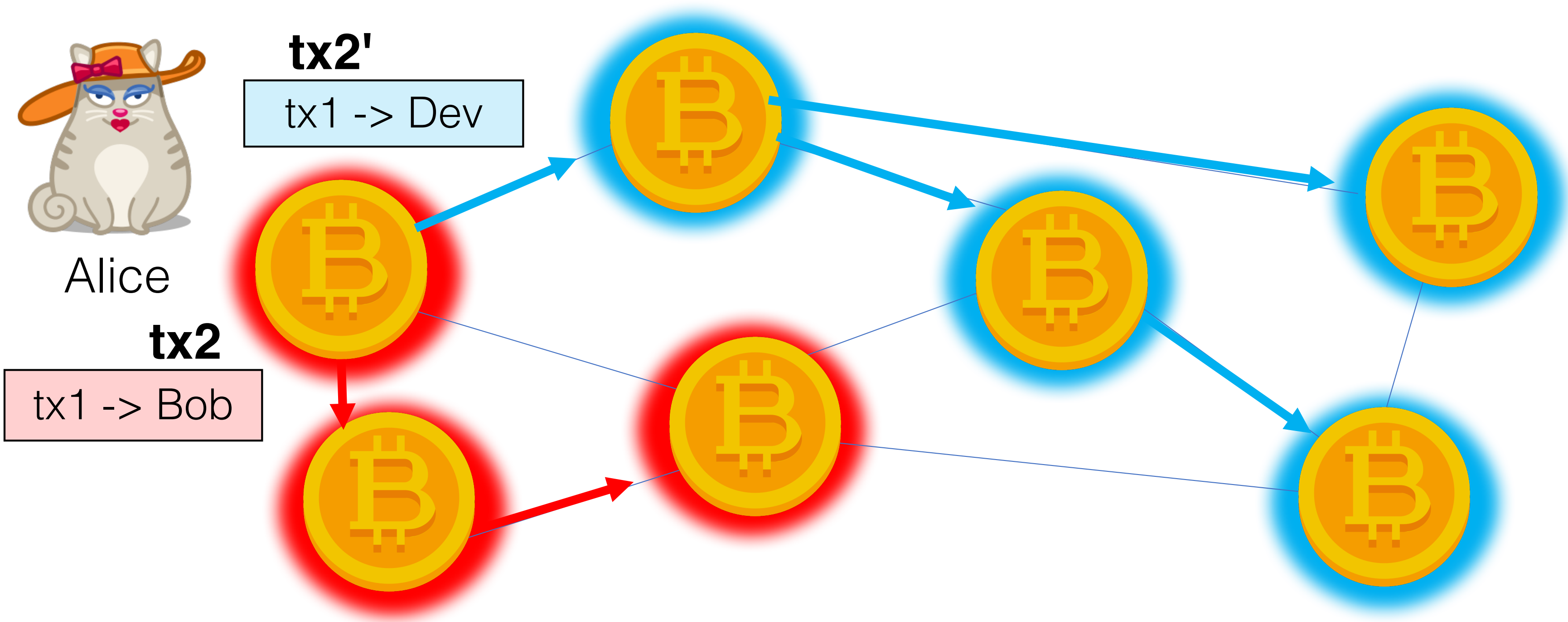
tx1

tx2

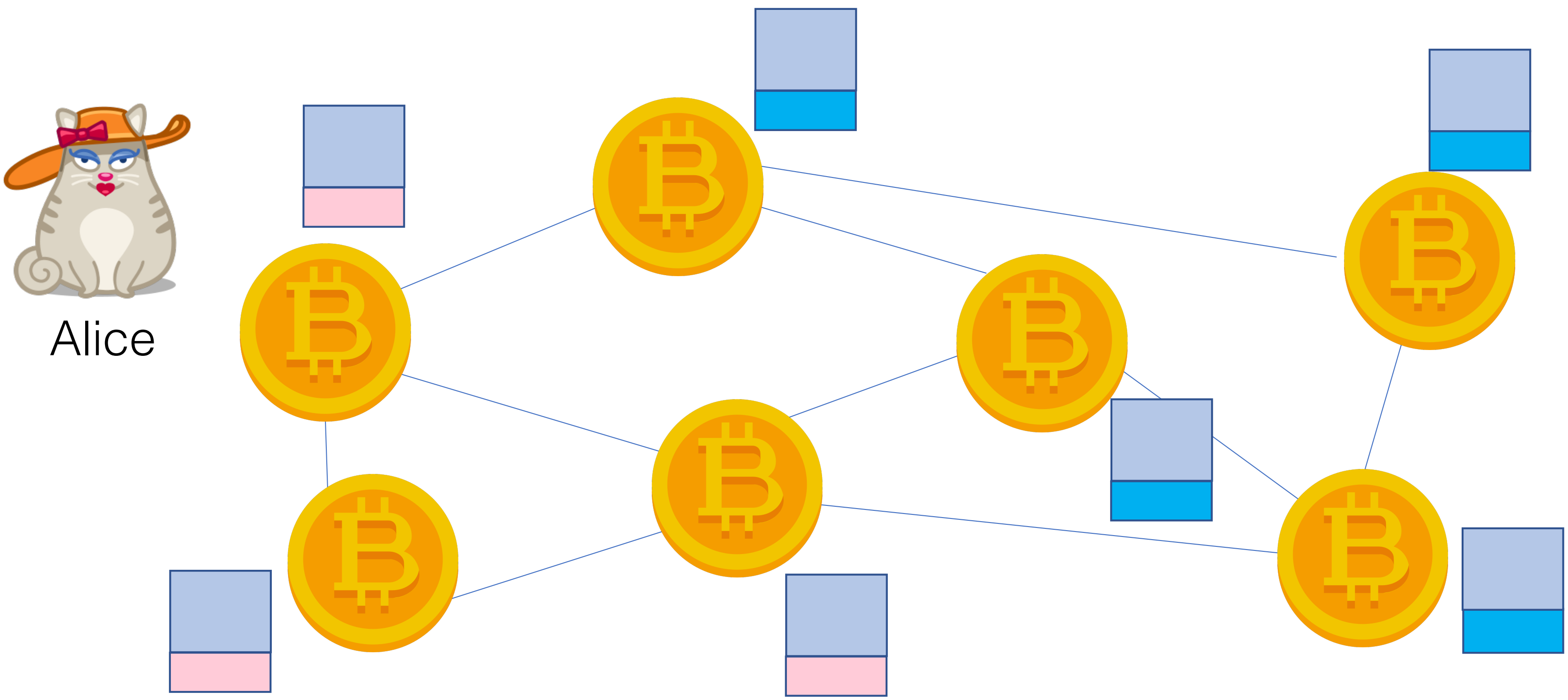


What's wrong with this?

Basic network operation



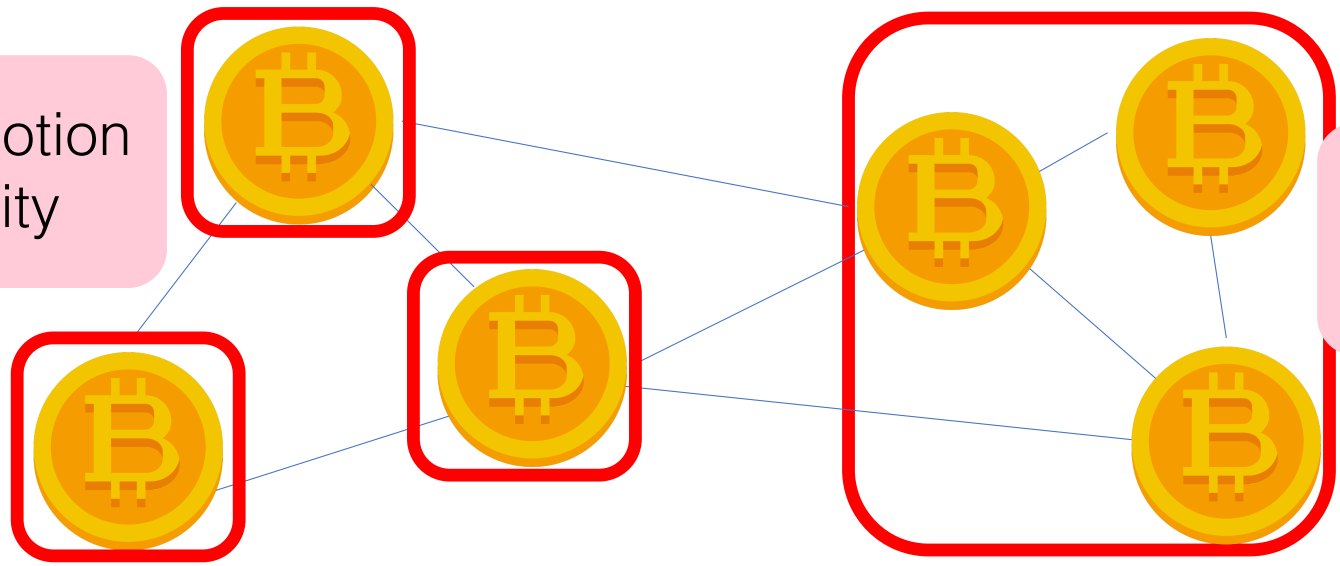
Adding to the Blockchain



Distributed Consensus in Bitcoin

Goal:
Pick 1 node uniformly at random

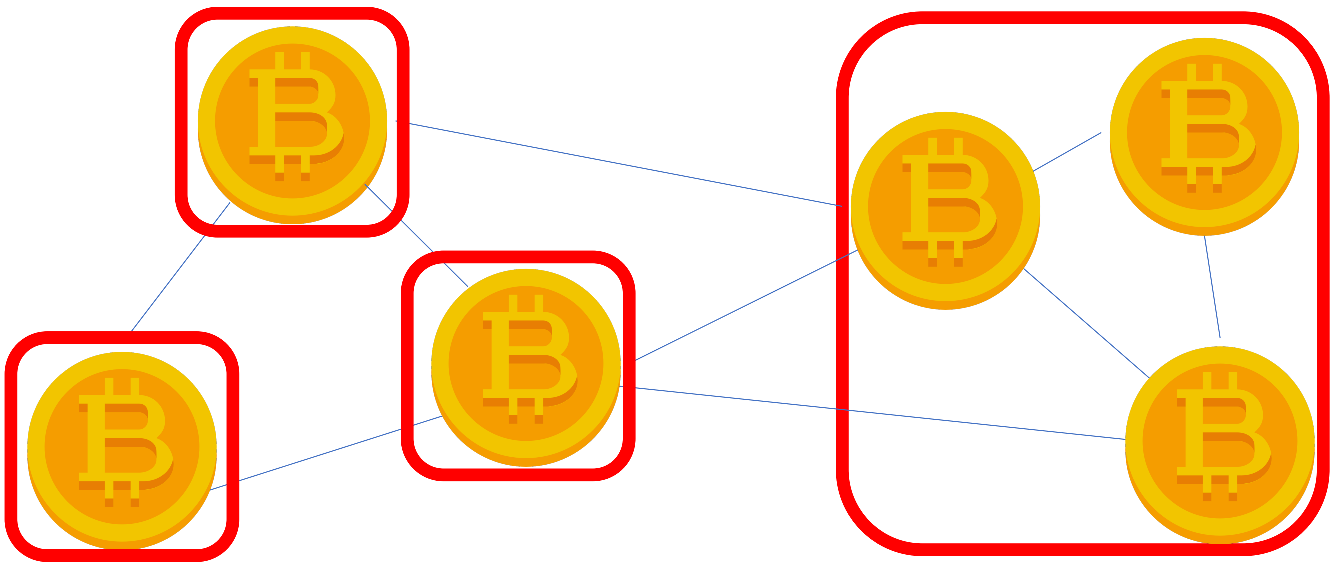
No fixed notion of identity



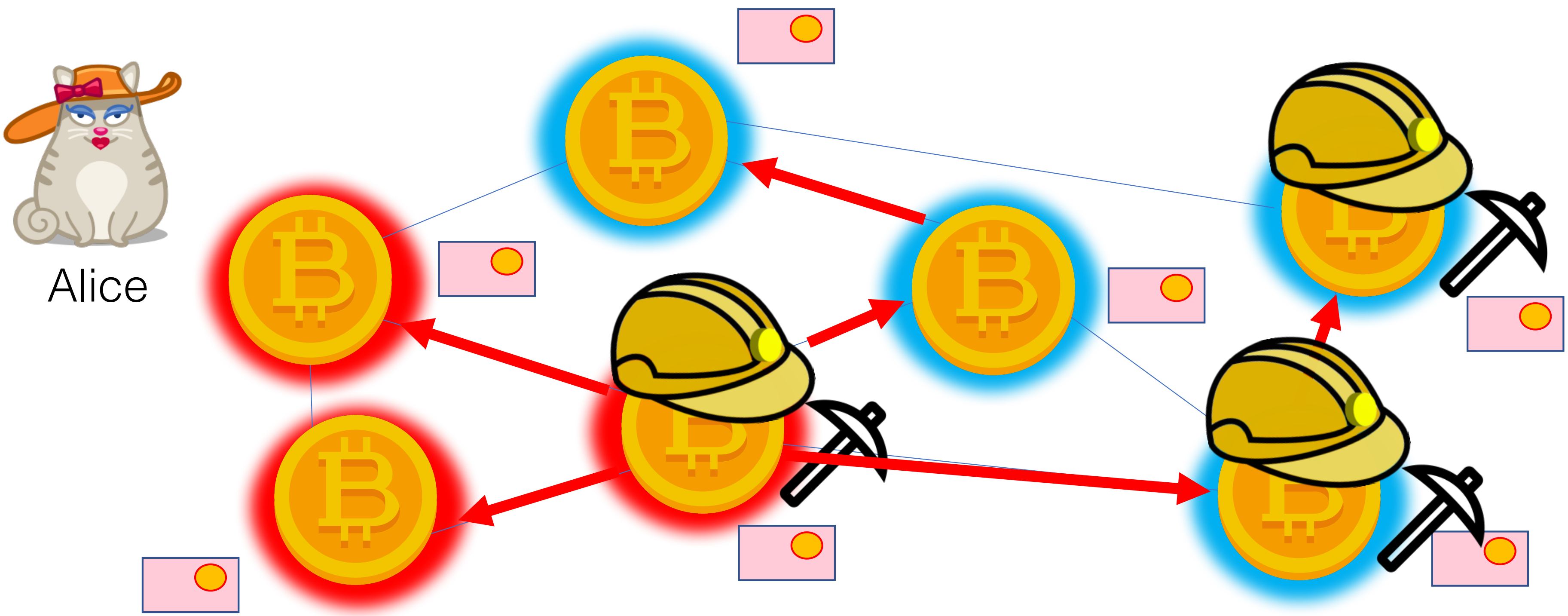
Robust to Sybils

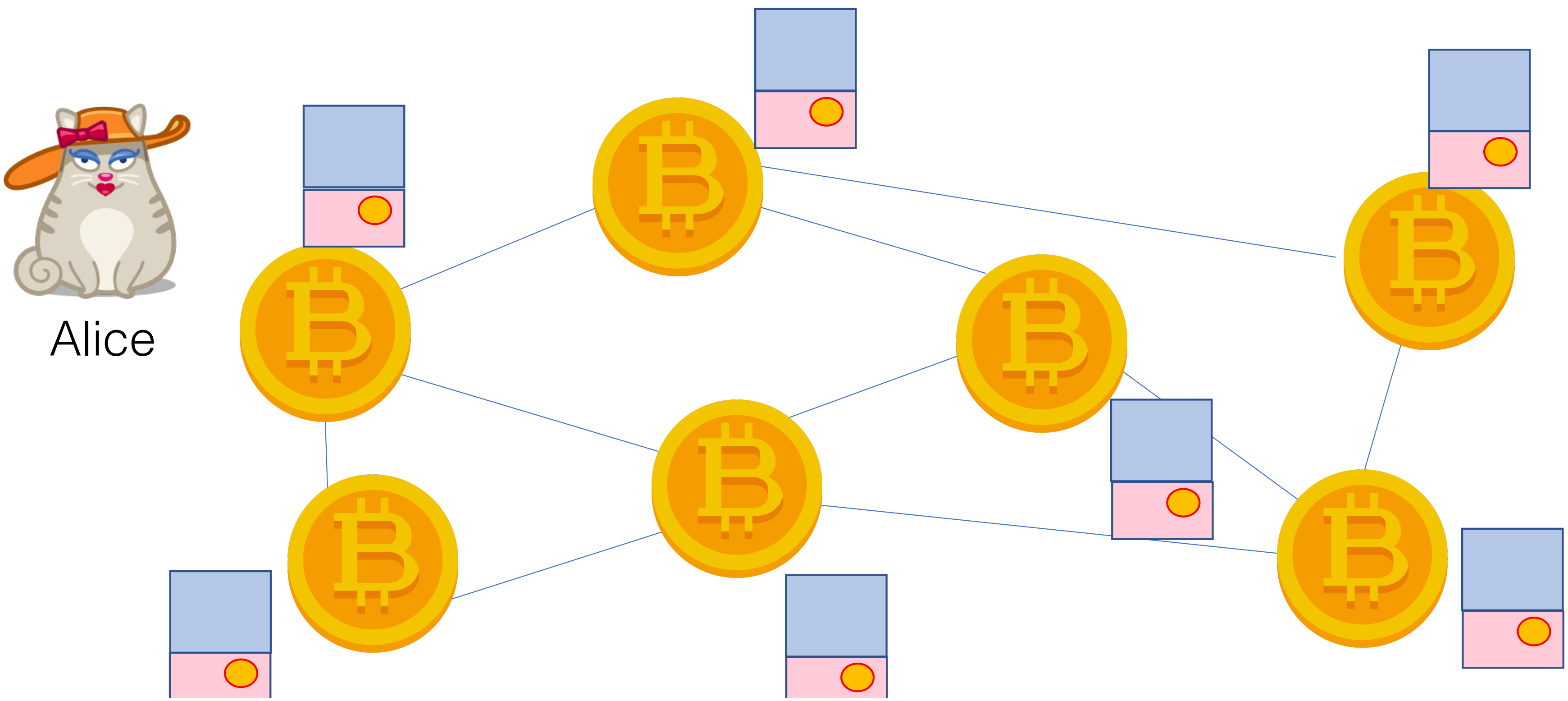
Proof-of-Work

Puzzle
Find x: $H(x) = f(tx, blockchain)$

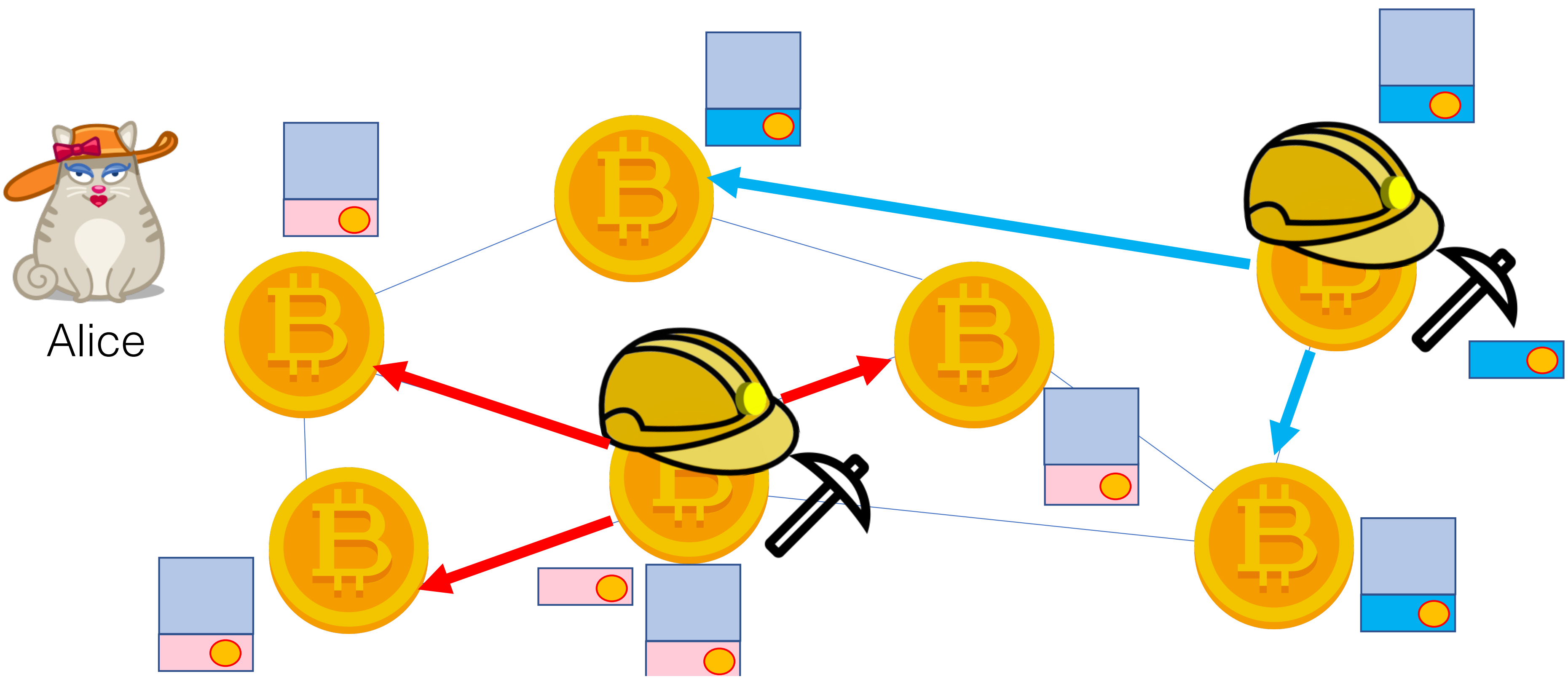


Mining

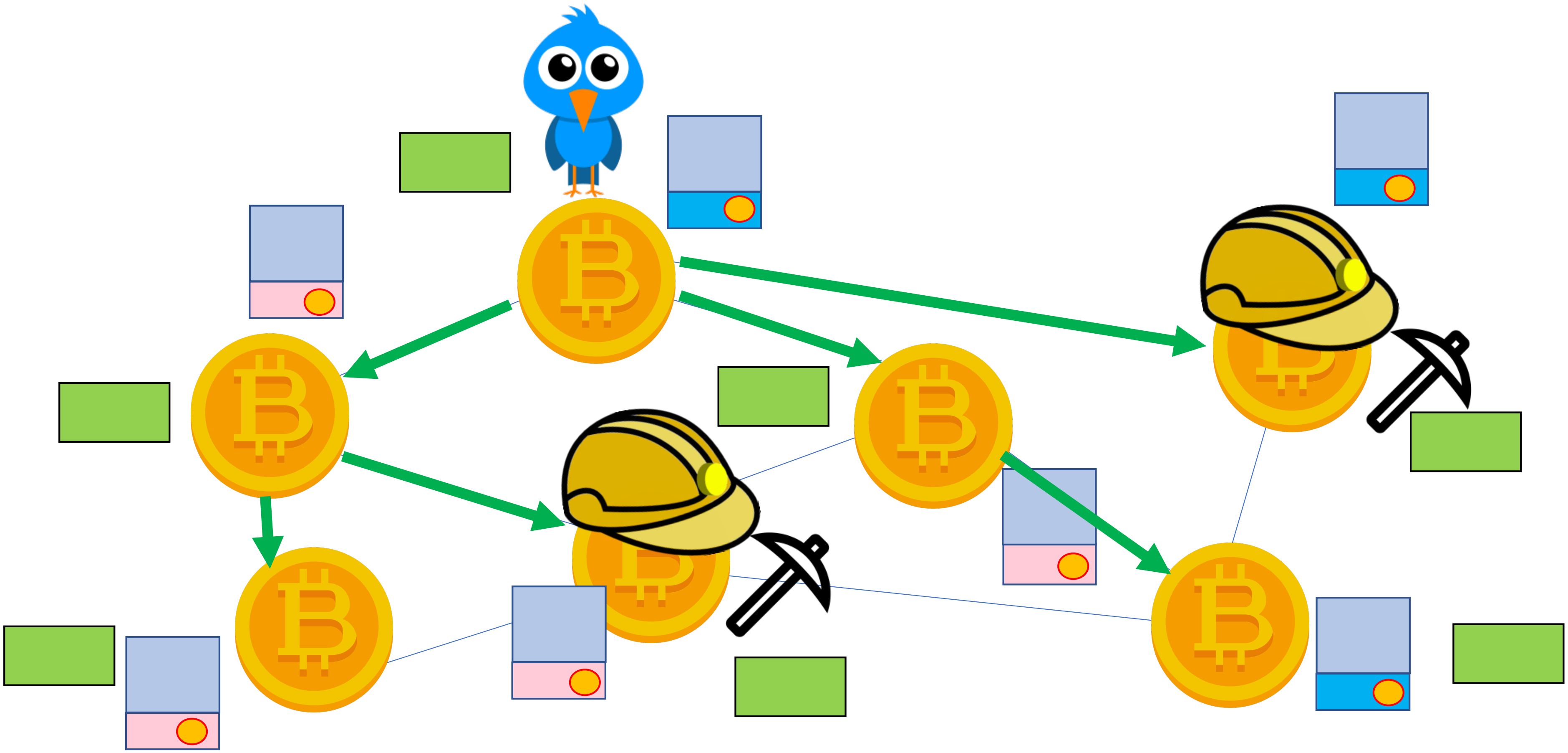



Alice

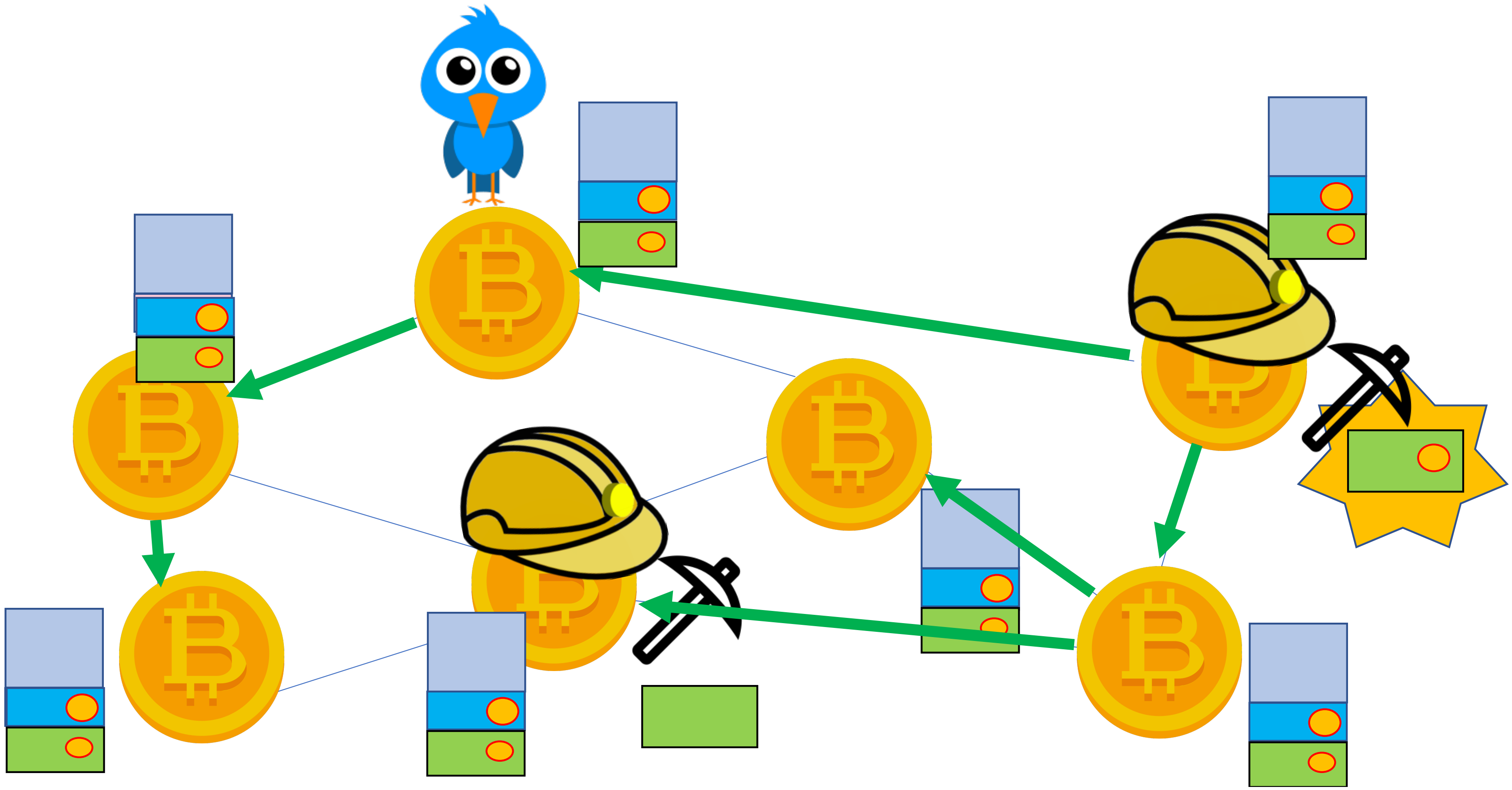
How are conflicts managed?



How are conflicts managed?



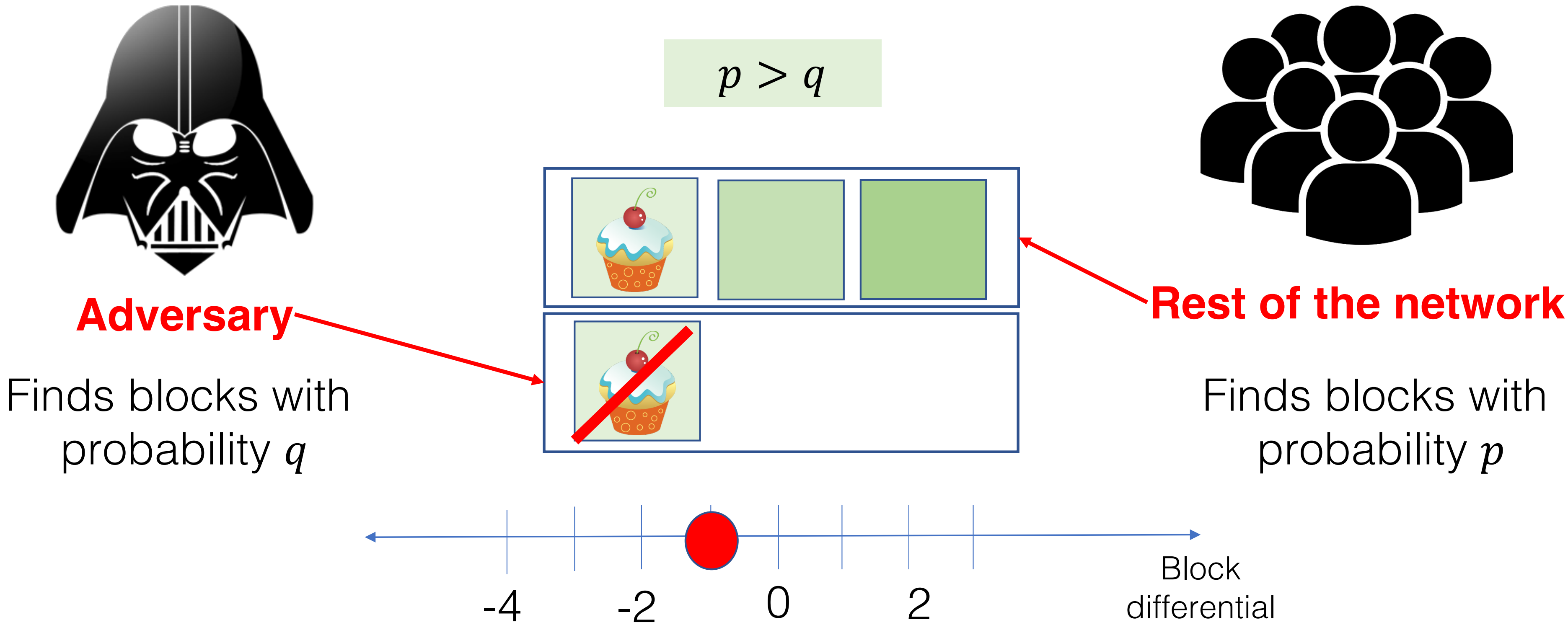
How are conflicts managed?



Bitcoin Consensus Protocol: Summary

- New transactions are broadcast
- Each node collects transactions into *blocks*
- One random node gets to broadcast its block / round
- Other nodes accept the block iff valid puzzle solution
- Miners “accept” blocks by referencing them in the next block

Probability of transaction reversal



Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System* (2008)

Probability of transaction reversal

p = Probability an honest node finds next block

q = Probability attacker finds next block

q_z = Probability attacker overtakes main blockchain starting from $-z$ differential

$$q_z = \begin{cases} 1, & if\ p \leq q \\ \left(\frac{q}{p}\right)^{-z}, & if\ p > q \end{cases}$$

This does not hold by assumption

Properties of Proofs of Work

	Cost	**Reward**
Measured in:	Computation	Bitcoins *(new-block reward, transaction fees)*
Scales according to:	Network's mining power *(1 block per 10 minutes)*	Geometric scaling

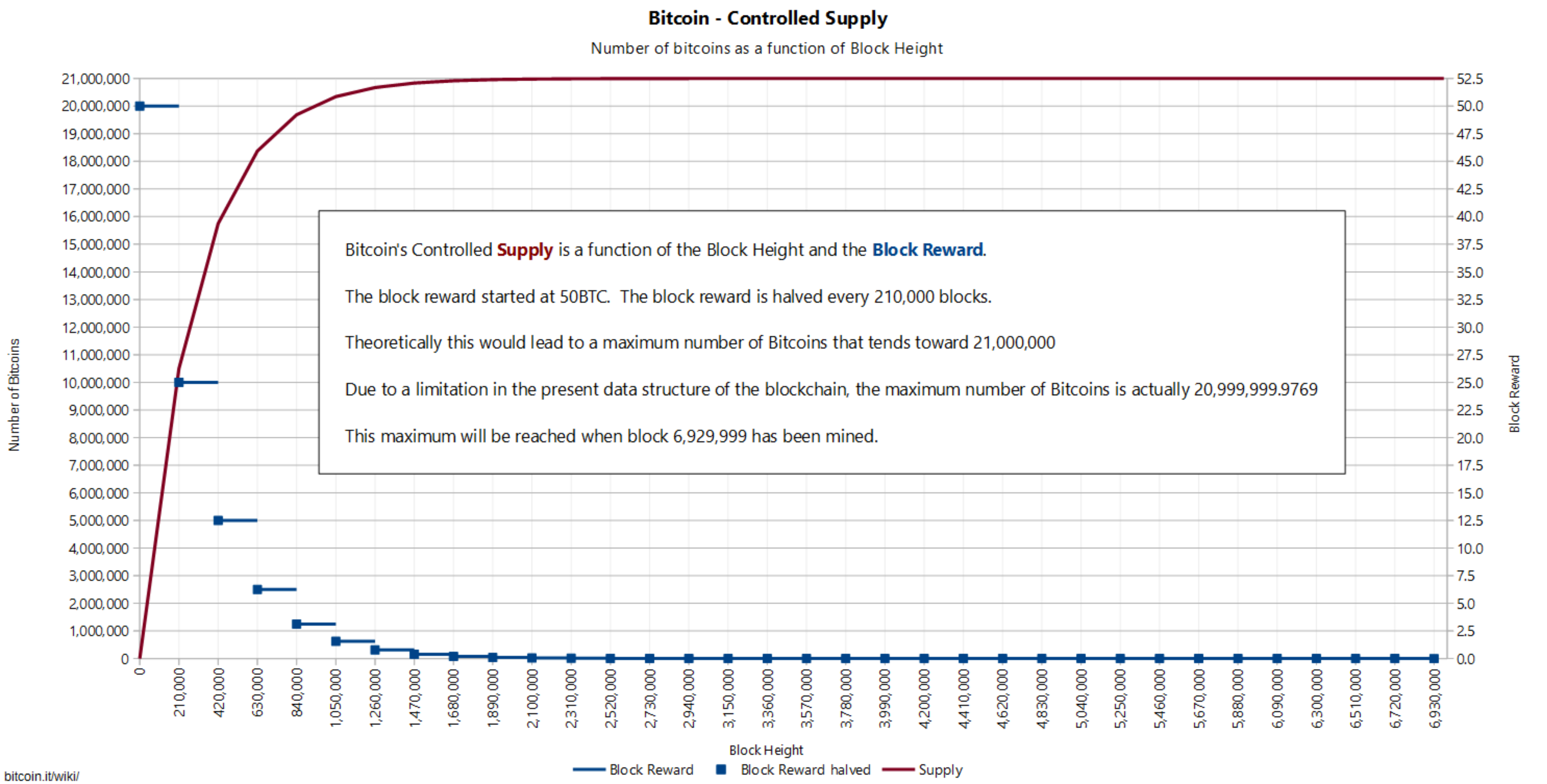


Image from BitcoinWiki

What purposes does mining serve?

Distributed consensus protocol

Limit rate of production

The Upshot

> Repeat after me: if you don't need concurrent access to a decentralized, mutable, singleton, you don't need a #blockchain.
>
> — ArthurB (@ArthurB) December 17, 2014

Why should the IT community care?

1. Network is central

2. Distributed storage

3. Game theory

This talk

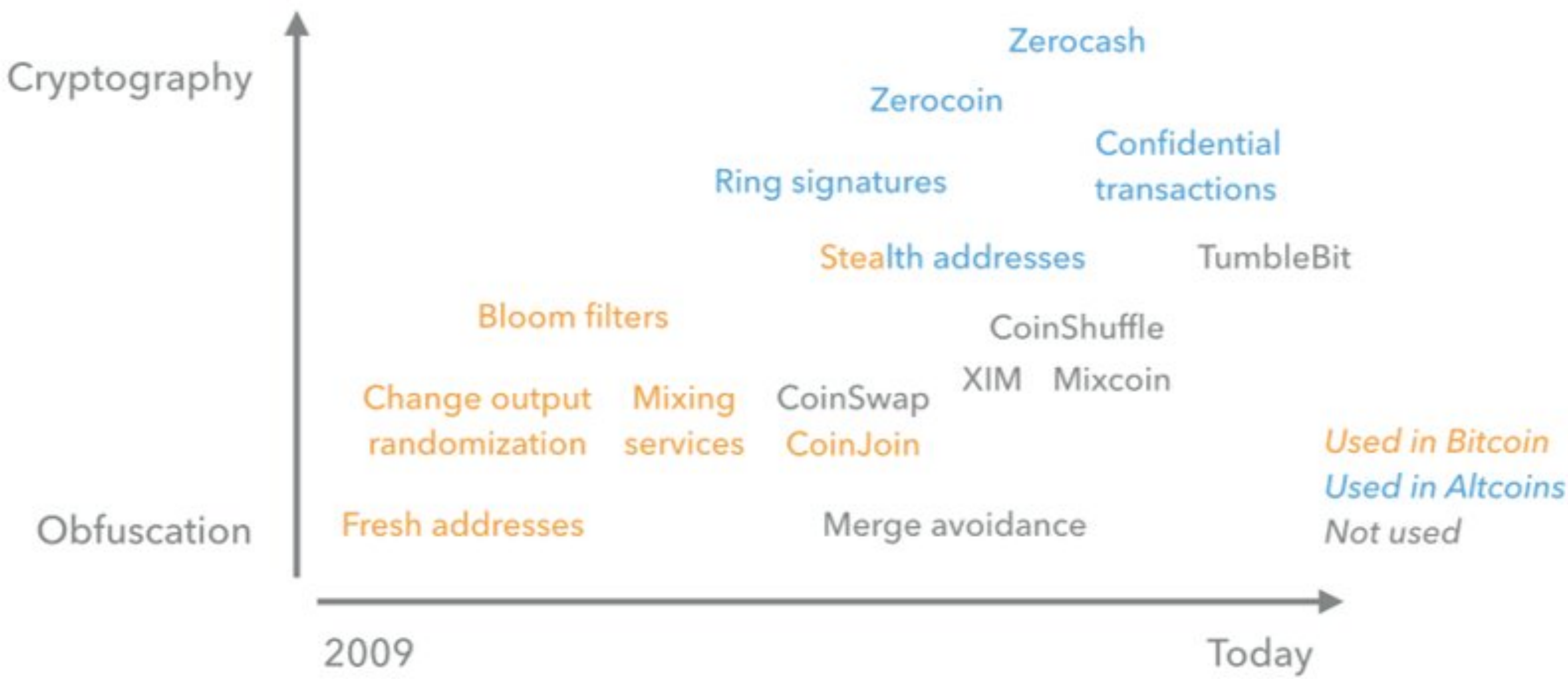


Figure 1: Privacy-Enhancing Technologies for Bitcoin. The X-axis is the date of invention and the Y-axis is an informal measure that combines the sophistication of the technique and the strength of the privacy guarantee. See Appendix 1 for references.

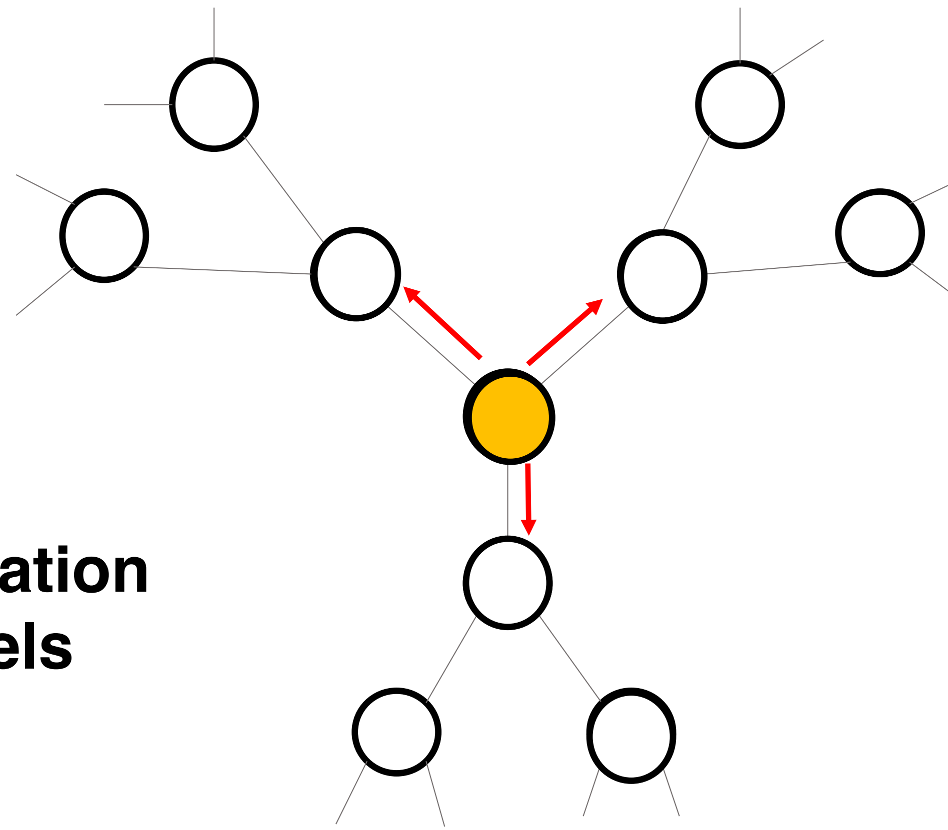
Narayanan and Moser, 2017

Models

Broadcasting over Networks

System Modeling

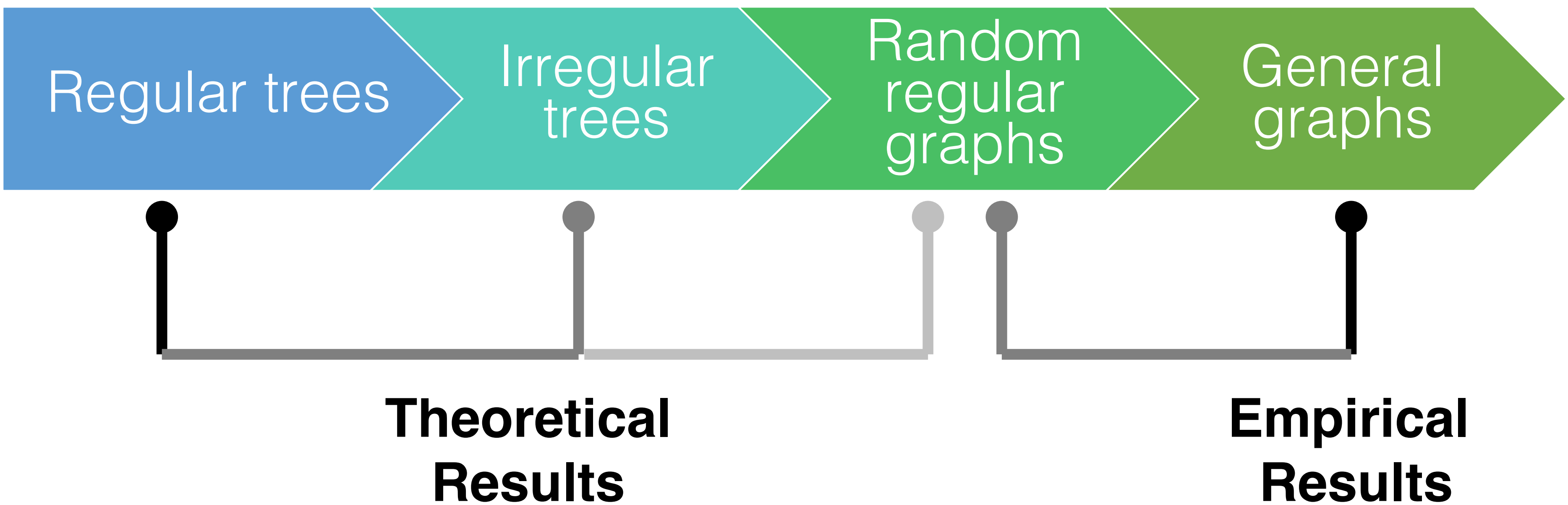
Network Models



Propagation Models

Observation/ Adversarial Models

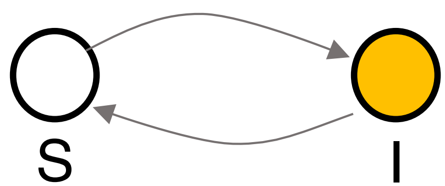
Network Models



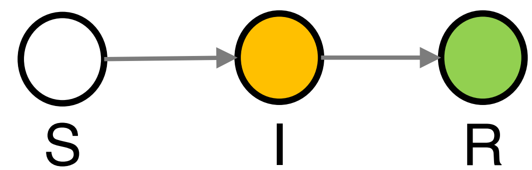
Propagation Models

Susceptible-Infected (SI)

Susceptible-Infected-Susceptible (SIS)



Susceptible-Infected-Recovered (SIR)

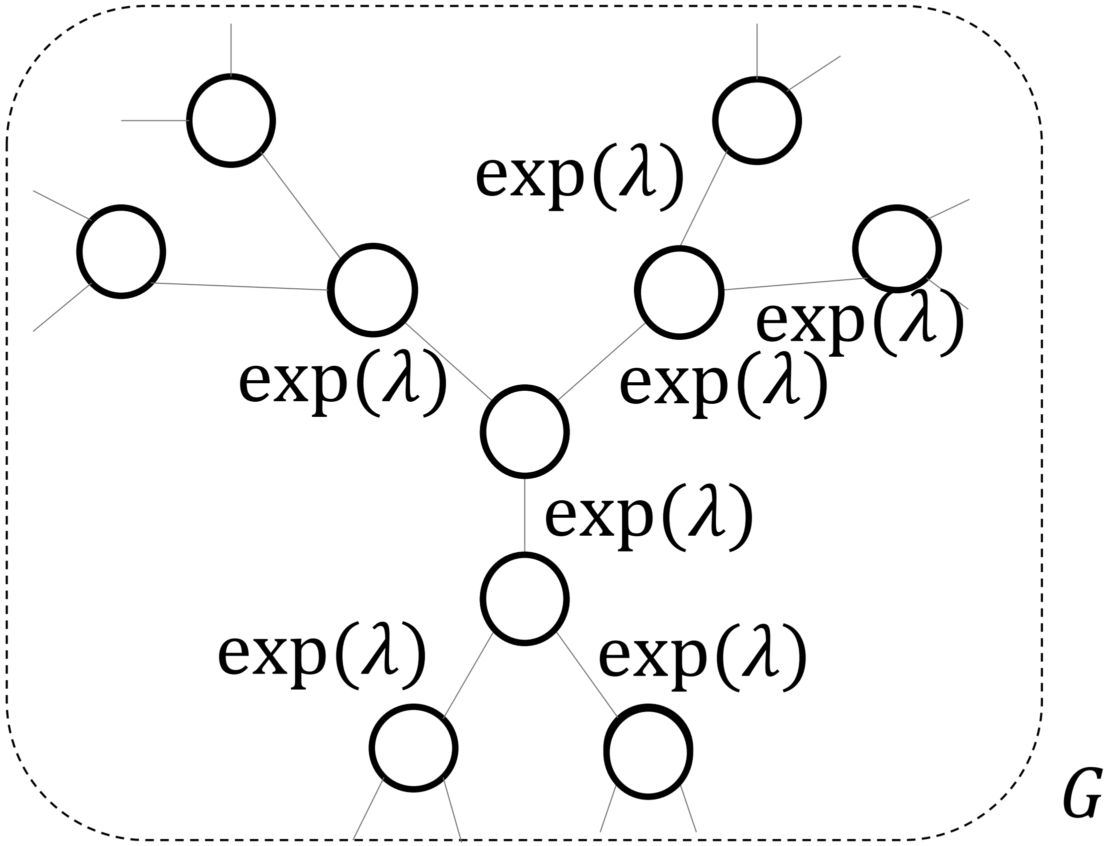


Propagation Models

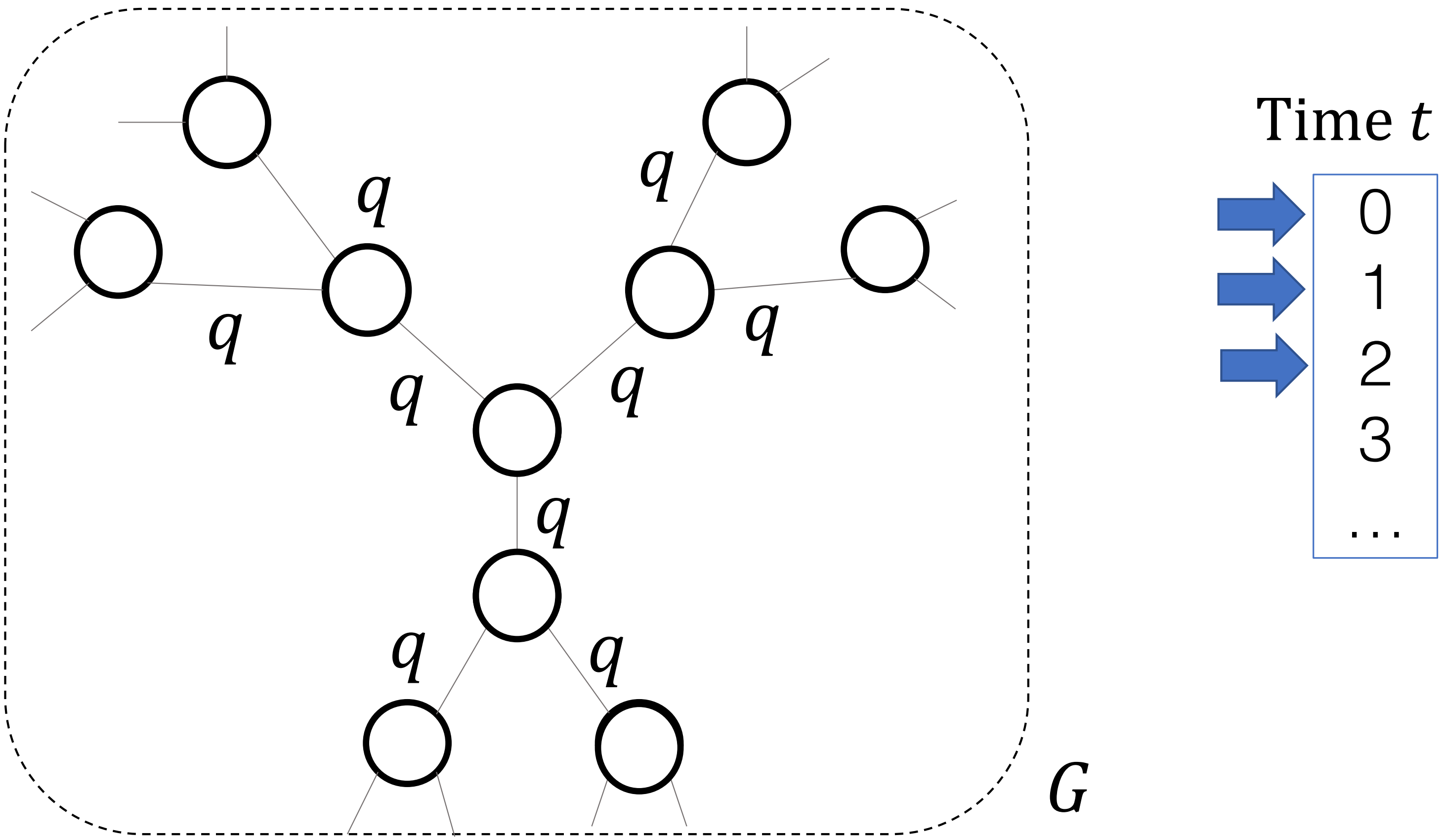
- Epidemics
- Social media
- Cryptocurrencies

	Susceptible-Infected (SI)	Susceptible-Infected-Susceptible (SIS)	Susceptible-Infected-Recovered (SIR)
Continuous-time			
Discrete-Time			

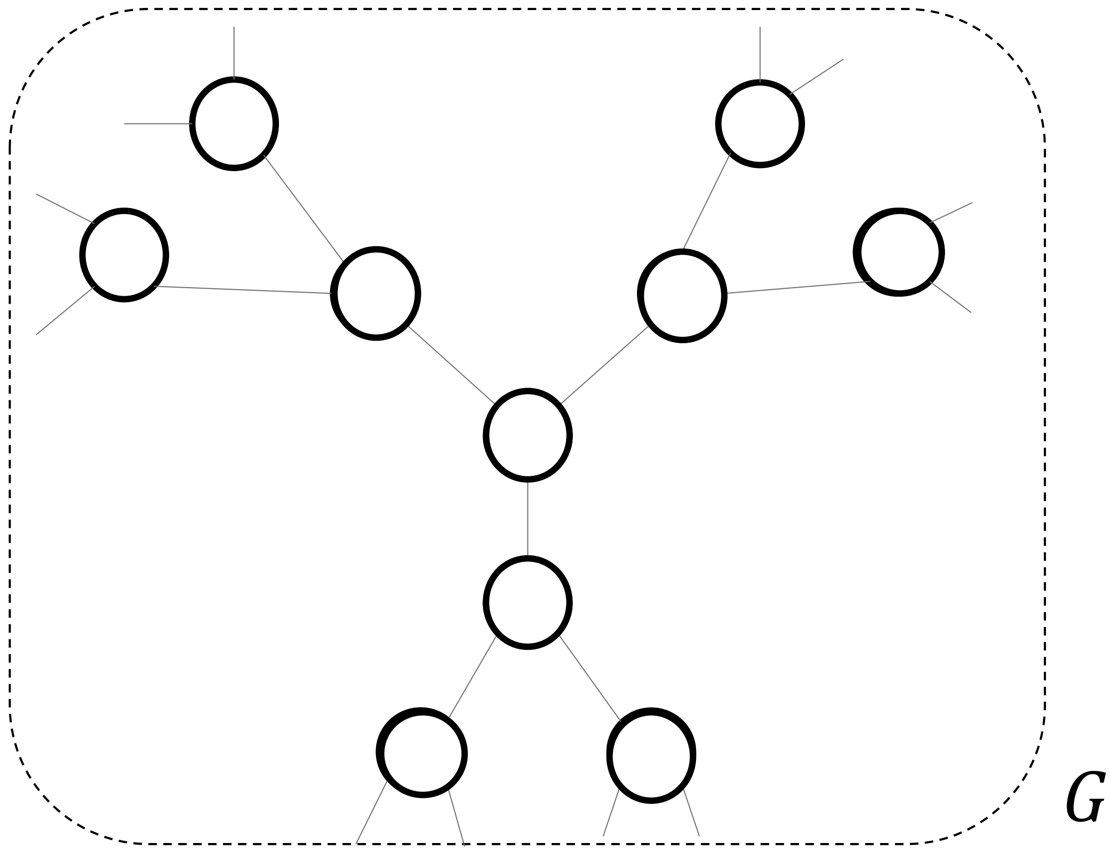
SI Diffusion (continuous-time)



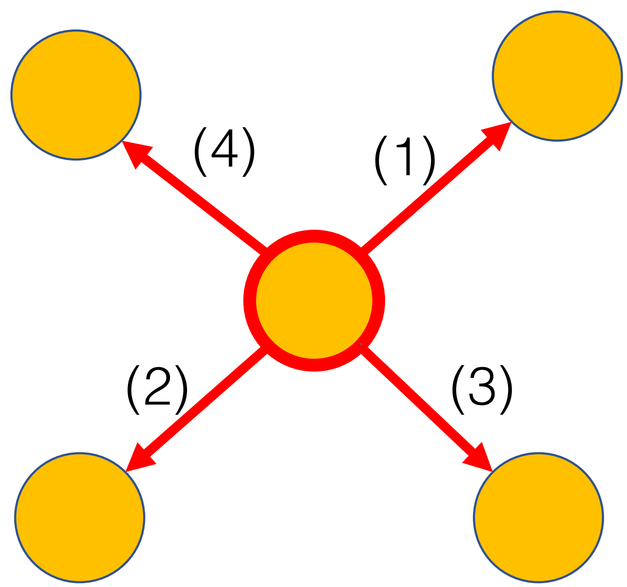
SI Diffusion (discrete-time)



SI Gossip (discrete-time)



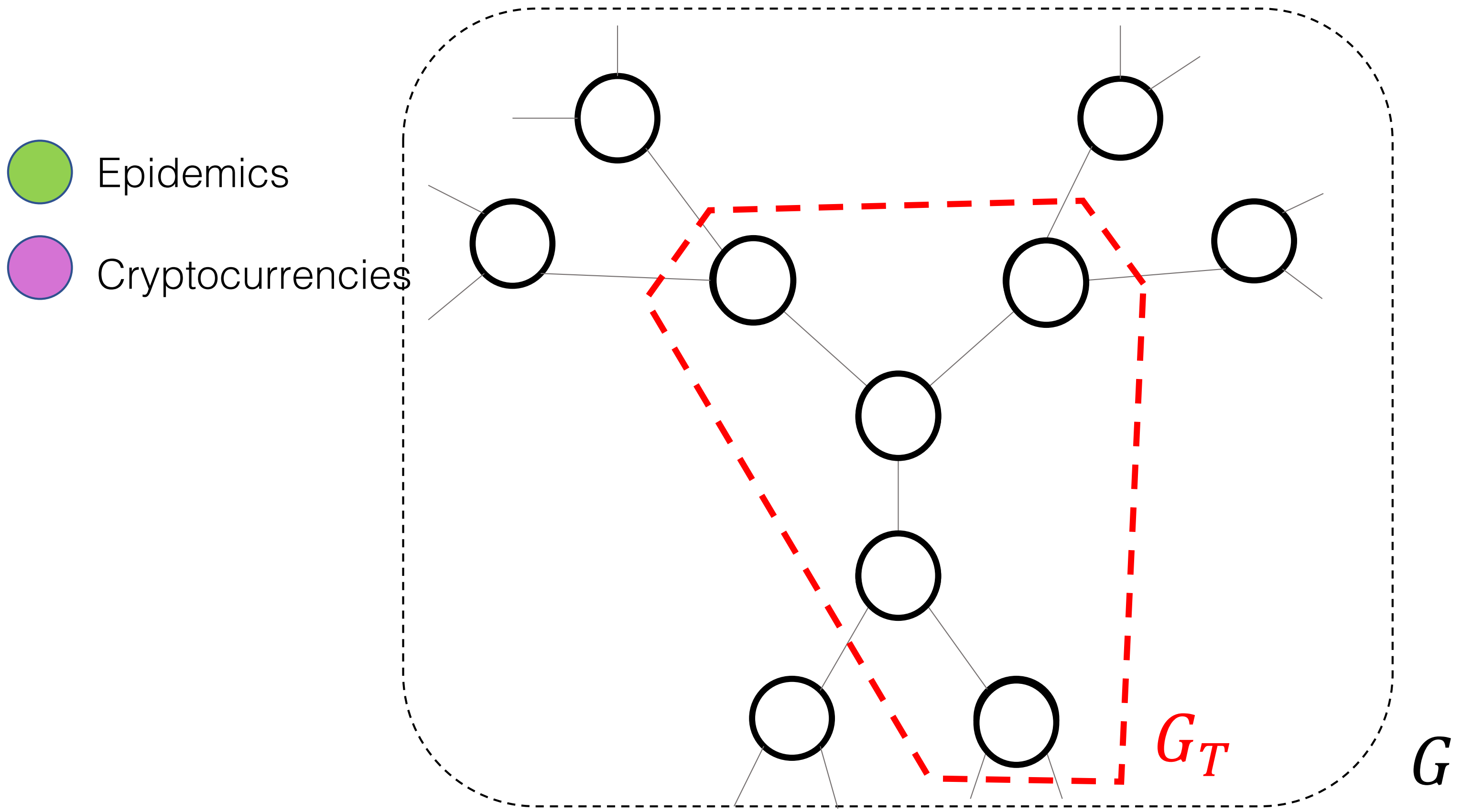
SI Gossip (discrete-time)



Propagation Models: Key attributes

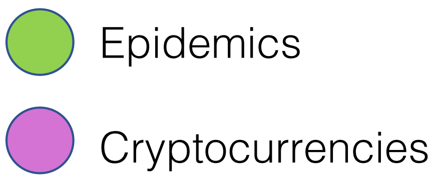
- Fully-distributed protocols
- Infection model can vary (**SI**, SIR, SIS)
- **Continuous**- vs. discrete-time systems
- Gossip vs. **diffusion**

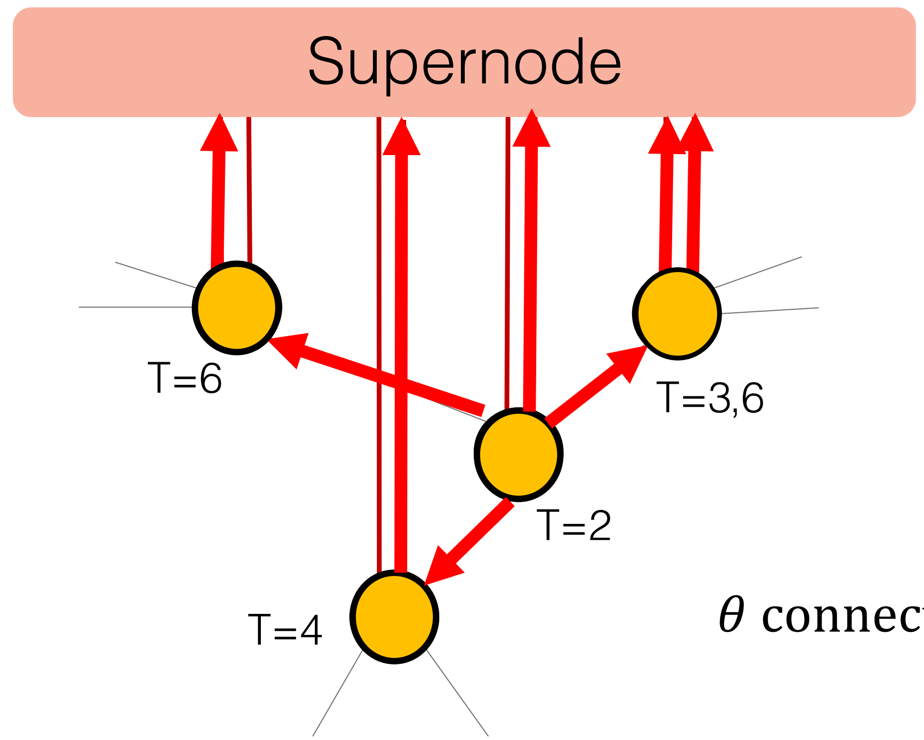
Snapshot Observer



Eavesdropping Observer

Eavesdropping Observer





θ connections per node

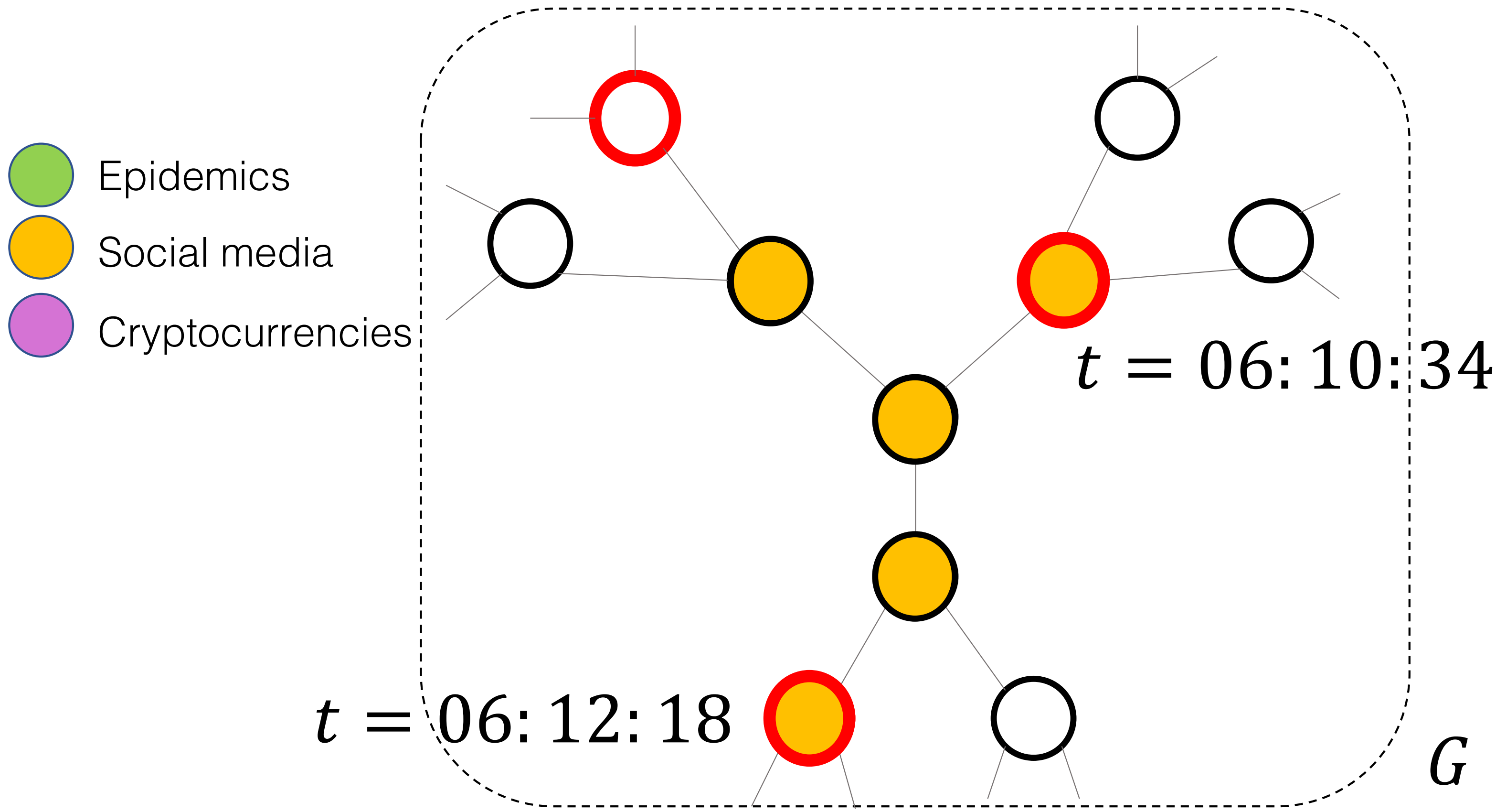
Spy-based Observer

The Facebook Squad: How Israel Police Tracks Activists on Social Media

It follows their Facebook pages, uses fake profiles to 'befriend' them and presents screenshots of posts in court – this is how Israel Police is adding social activists to its virtual surveillance list. 'They know what I write and do,' Ethiopian protest leader says.

Yaniv Kubovich | Feb 06, 2016 9:46 AM

Sampled Observers (Spies)



Observation Models: Key Attributes

- Fraction of nodes that can be observed (all nodes, subset)
- Delay of observation at those nodes (instantaneous / random)
- Nodes' adherence to protocol (honest-but-curious / malicious)

Summary: Modeling Epidemics

- Network models
  - **Trees**
  - General graphs (social networks, random graphs)

- Spreading models
  - **Diffusion**

- Observation/adversarial models
  - **Snapshot**
  - Spy-based, eavesdropper

Finding the Source

Part II

What you will learn in this hour

- Source detection algorithms
    - Rumor centrality
    - Other heuristics

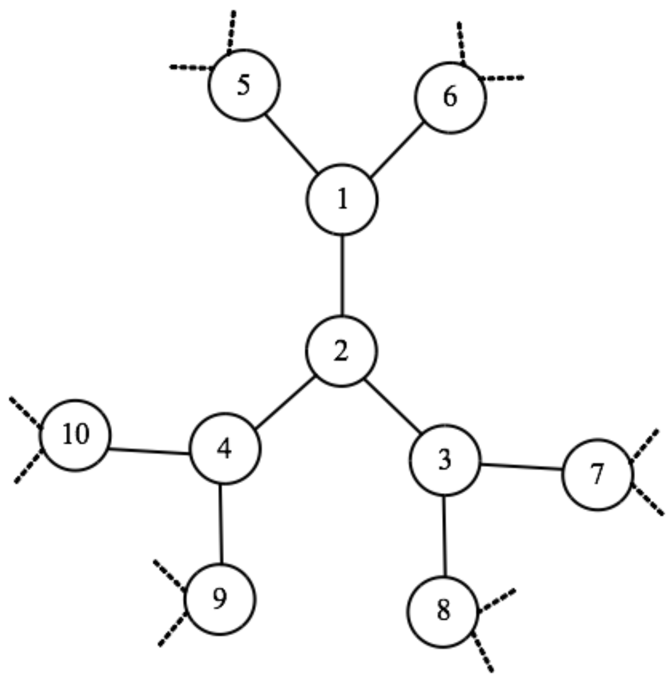
- Introduction to Pólya urns
    - Definition
    - Convergence results
    - Generalizations

- Using Pólya urn processes to analyze the probability of source detection in diffusion processes

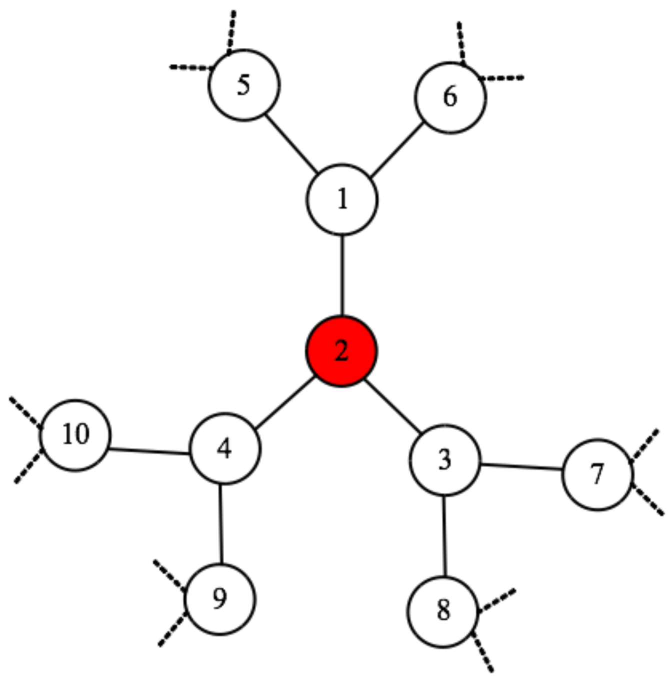
Source Detection Algorithms

Centrality measures

Rumors in networks

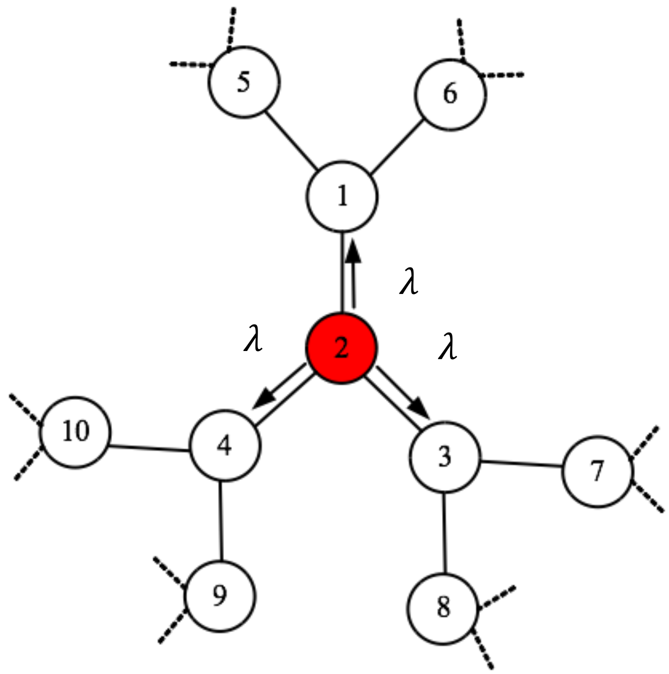


Rumors in networks



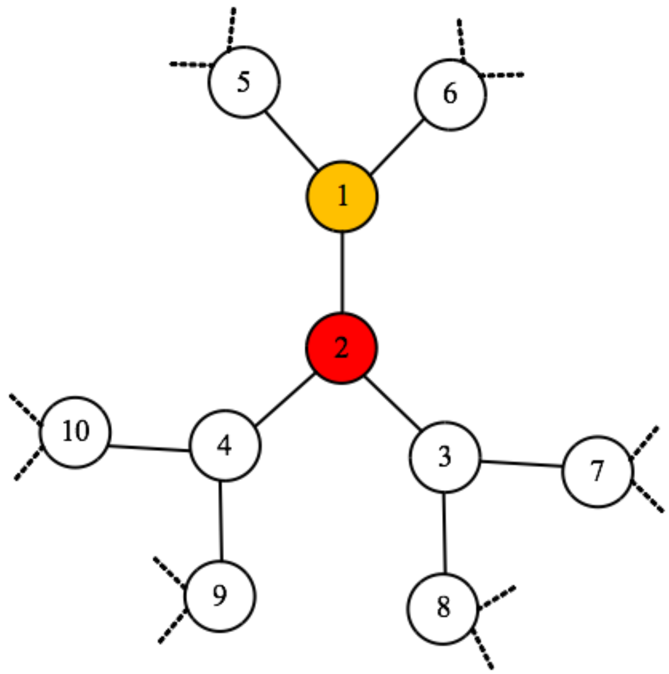
- a random node is the source of the rumor

Diffusion spreading

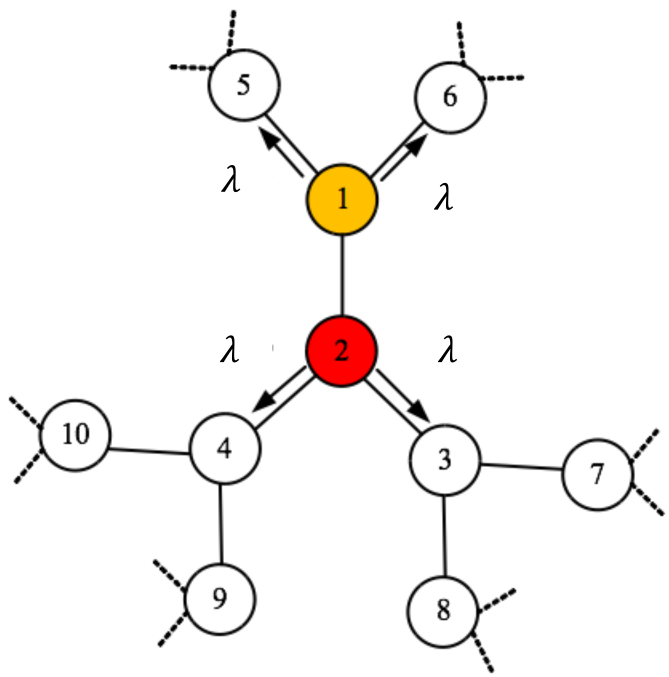


- Node 2 spreads the rumor to its neighbors iid along its edges

Rumors in networks

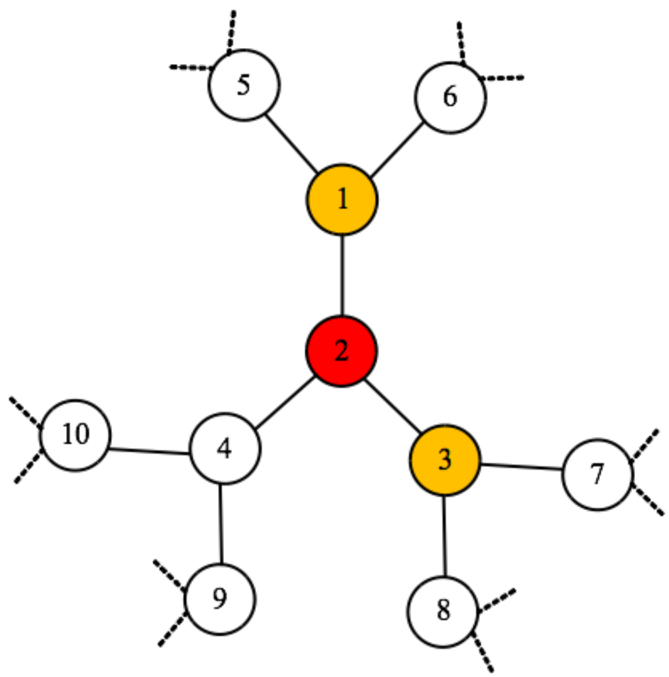


Diffusion Spreading



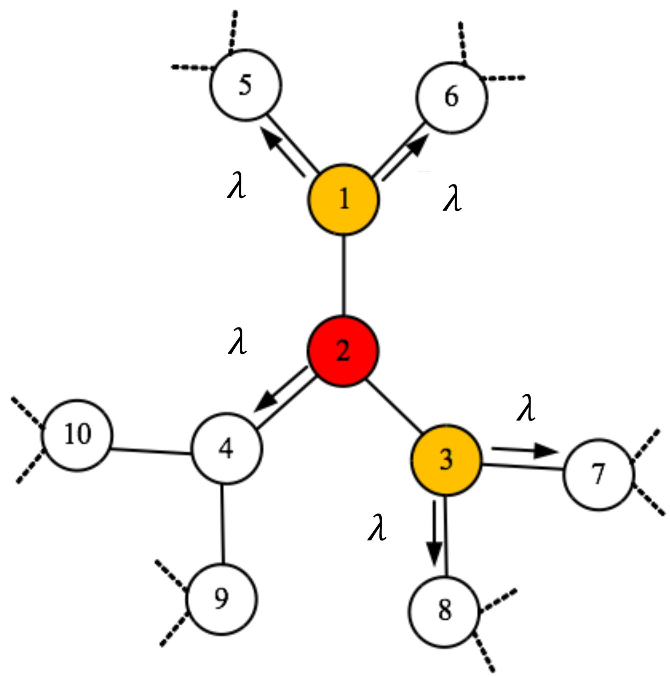
- Both nodes 1 and 2 spread the message along their edges

Diffusion Spreading

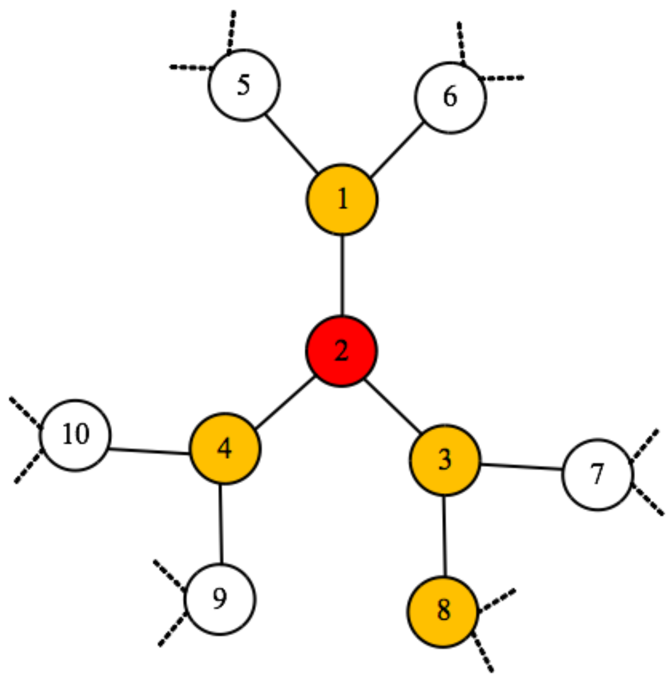


- Node 3 receives the message, say.

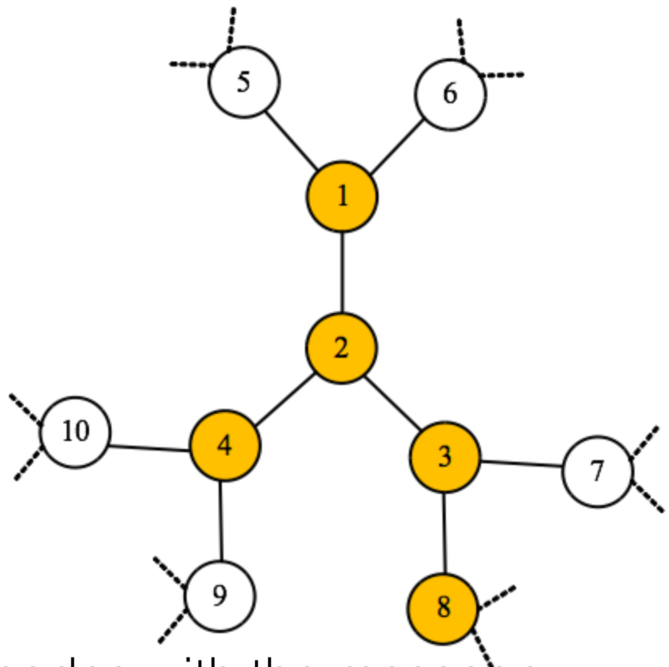
Diffusion Spreading



Diffusion Spreading

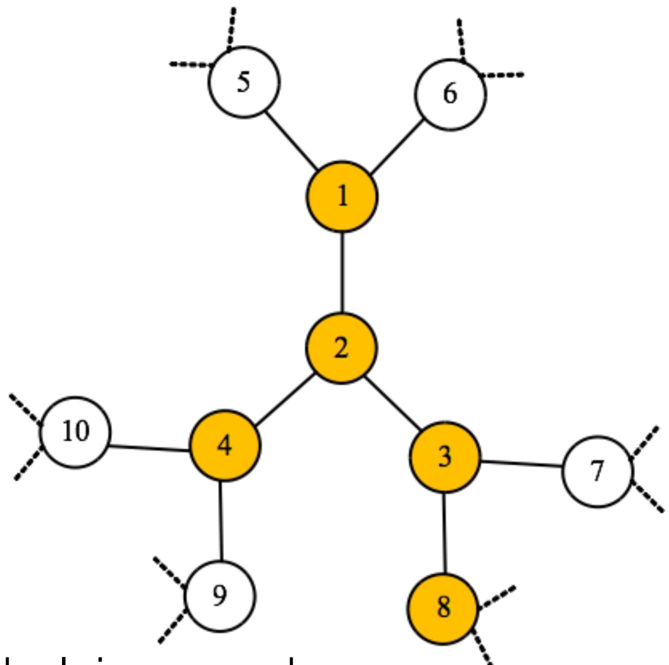


Snapshot observation



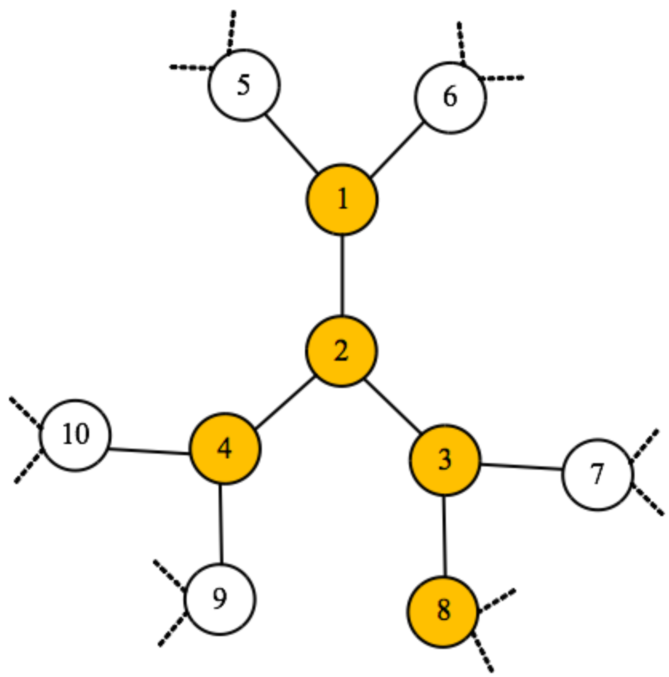
- Get to observe set of nodes with the message
- No timestamps

Source of Rumor



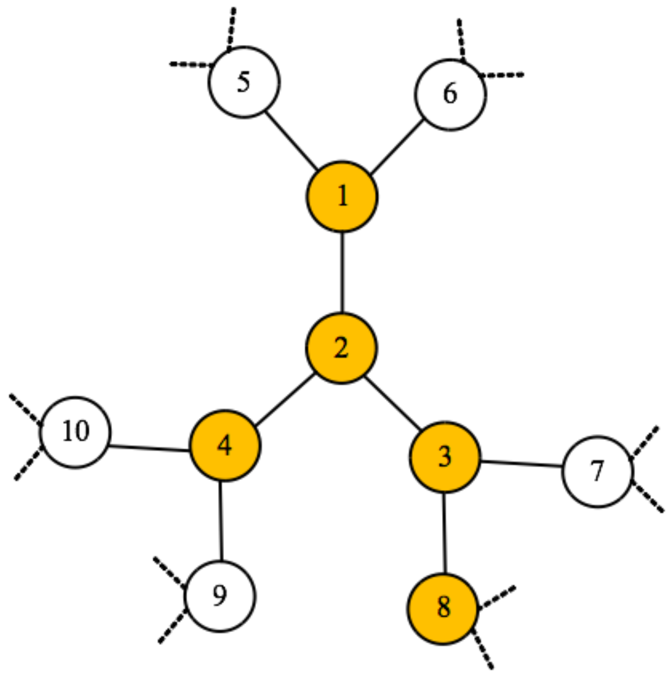
- Use knowledge of underlying graph
- knowledge of set of nodes with the message

Centrality



- Source is in the center

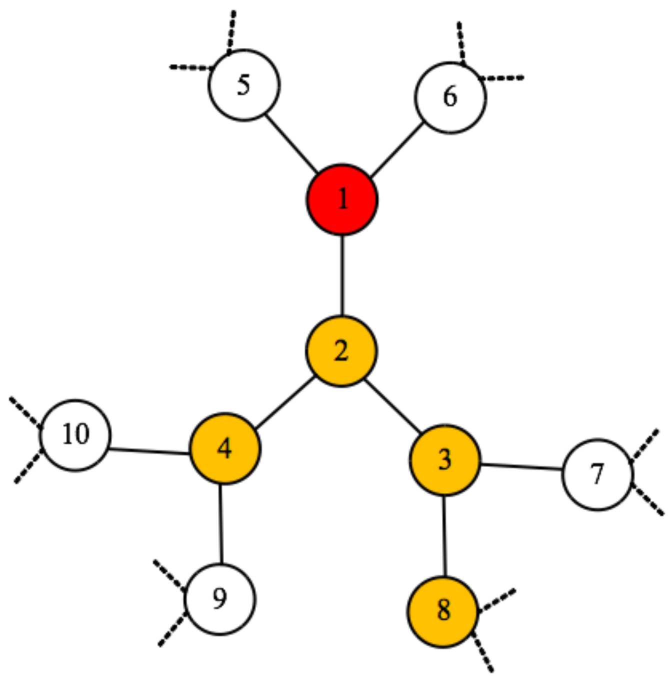
Rumor centrality



- Specific metric of centrality

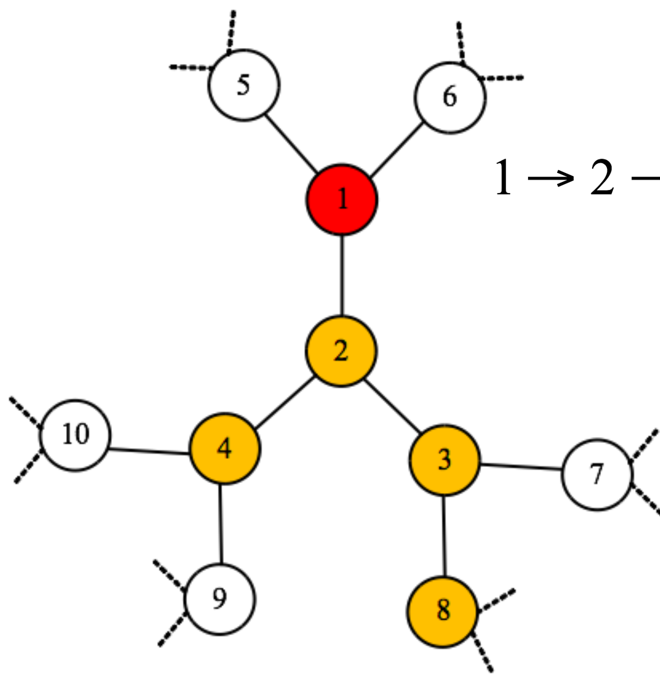
Shah and Zaman, *Rumors in a Network: Who's the Culprit?*, IT Transactions, 2011

Rumor centrality



- Hypothesis: node 1 is the source

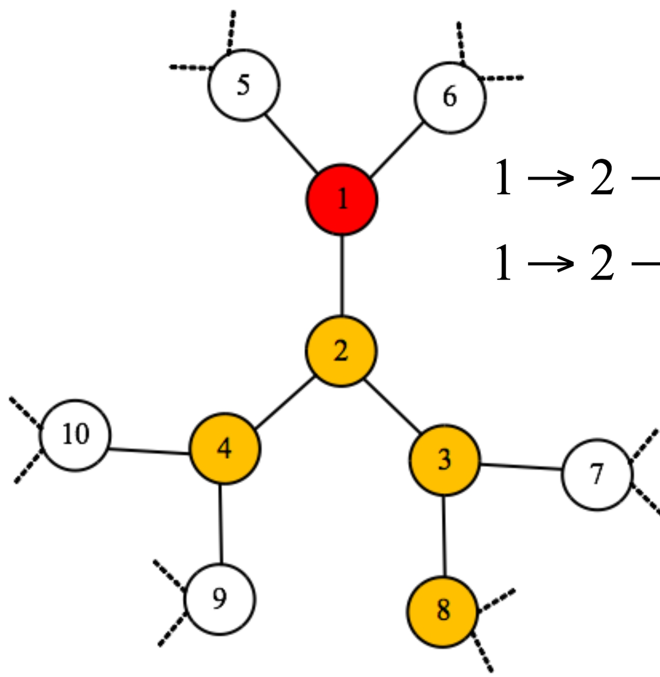
Rumor centrality



$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 8$

- Identify a possible spreading pattern

Rumor centrality

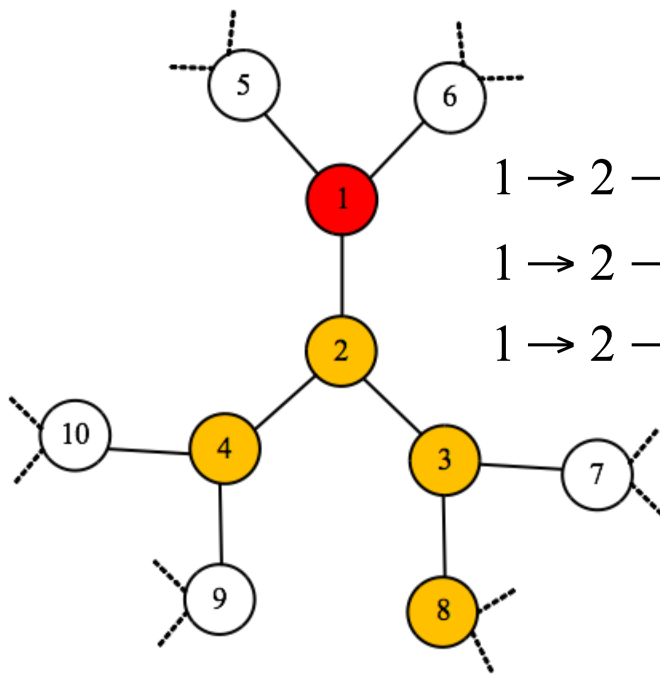


$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 8$$

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 8 \rightarrow 4$$

- Enumerate all possible spreading patterns

Rumor centrality

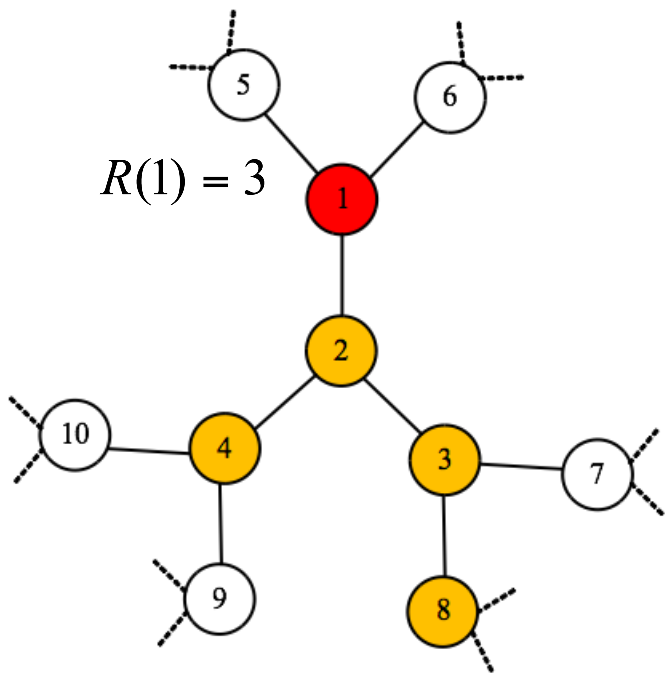


$$1 \to 2 \to 3 \to 4 \to 8$$

$$1 \to 2 \to 3 \to 8 \to 4$$

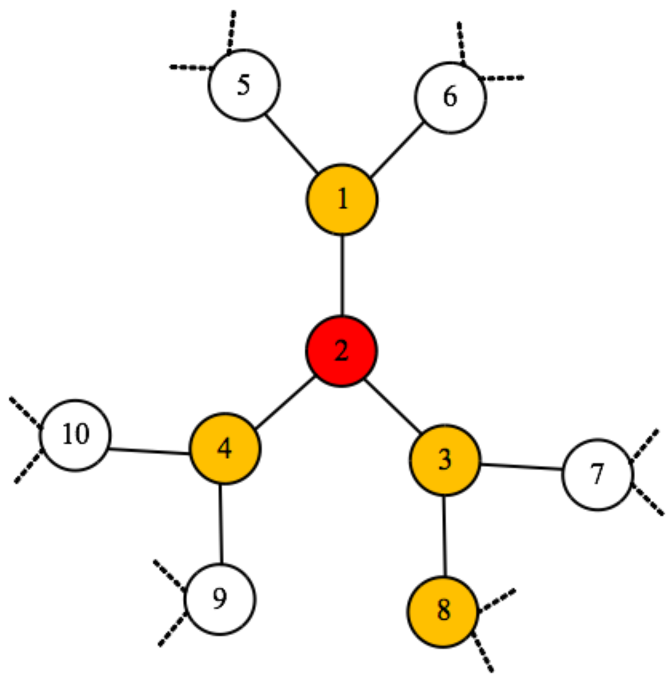
$$1 \to 2 \to 4 \to 3 \to 8$$

Rumor centrality



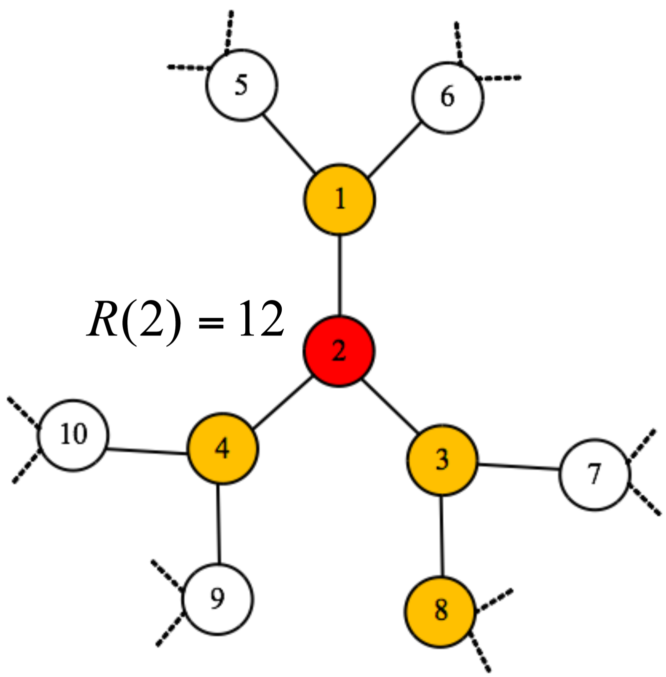
- Score = number of possible spreading patterns

Rumor centrality



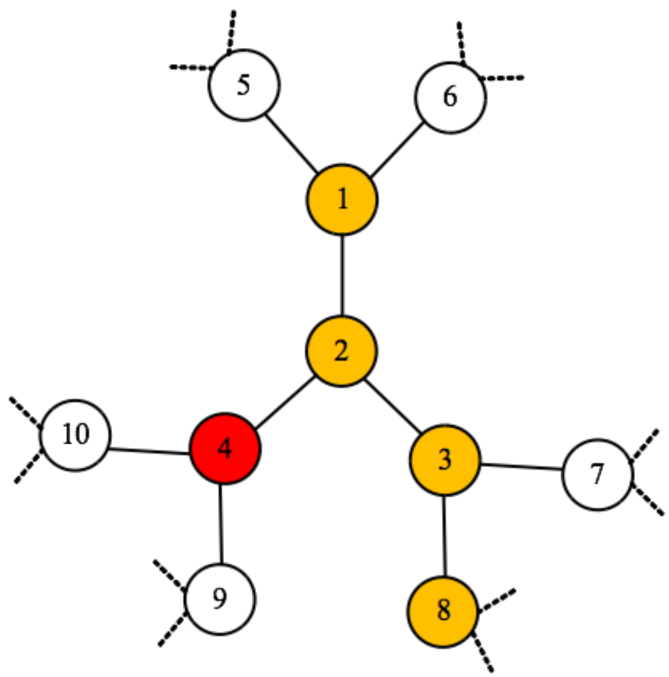
- Similar score for node 2

Rumor centrality

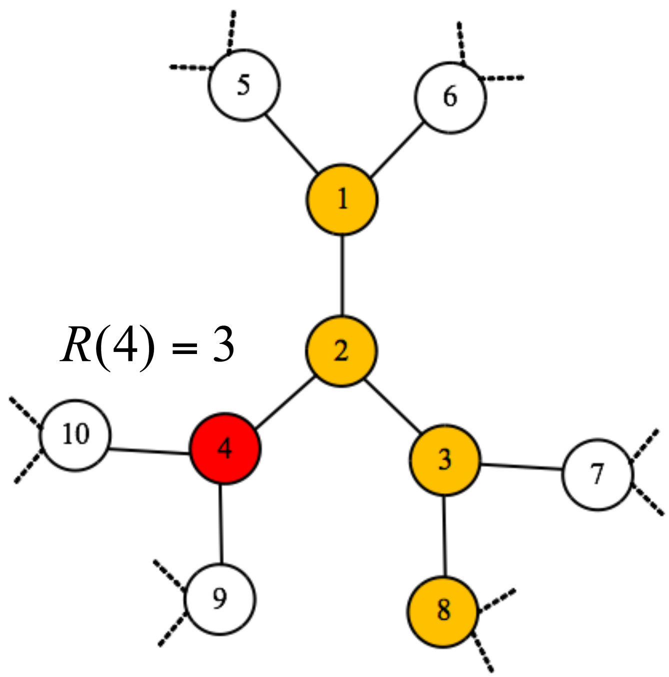


$$2 \to 1 \to 4 \to 3 \to 8$$
$$2 \to 1 \to 3 \to 4 \to 8$$
$$2 \to 1 \to 3 \to 8 \to 4$$
$$2 \to 4 \to 1 \to 3 \to 8$$
$$2 \to 4 \to 3 \to 1 \to 8$$
$$2 \to 4 \to 3 \to 8 \to 1$$
$$2 \to 3 \to 1 \to 4 \to 8$$
$$2 \to 3 \to 1 \to 8 \to 4$$
$$2 \to 3 \to 8 \to 1 \to 4$$
$$2 \to 3 \to 4 \to 1 \to 8$$
$$2 \to 3 \to 4 \to 8 \to 1$$

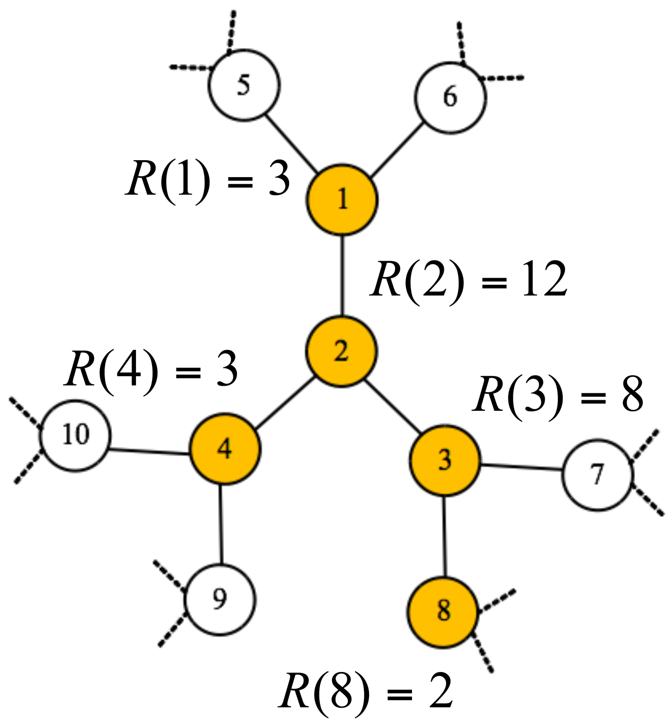
Rumor centrality



Rumor centrality

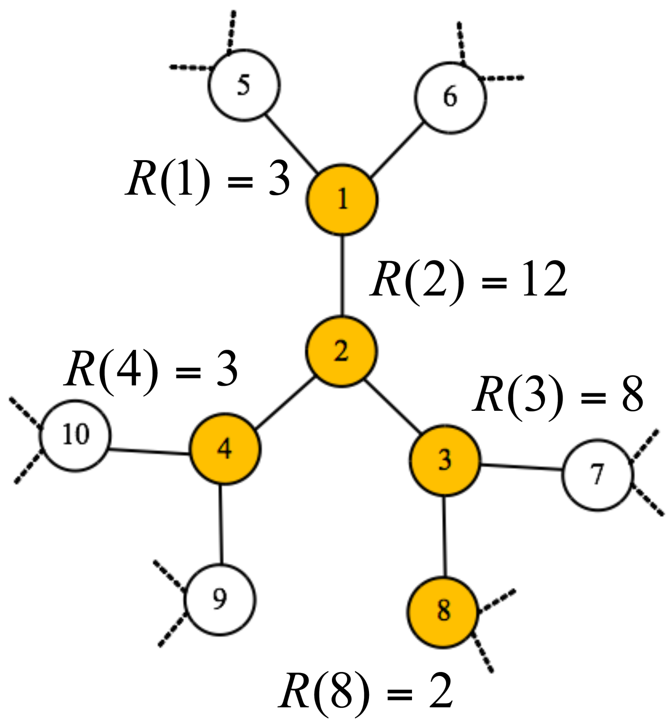


Rumor centrality



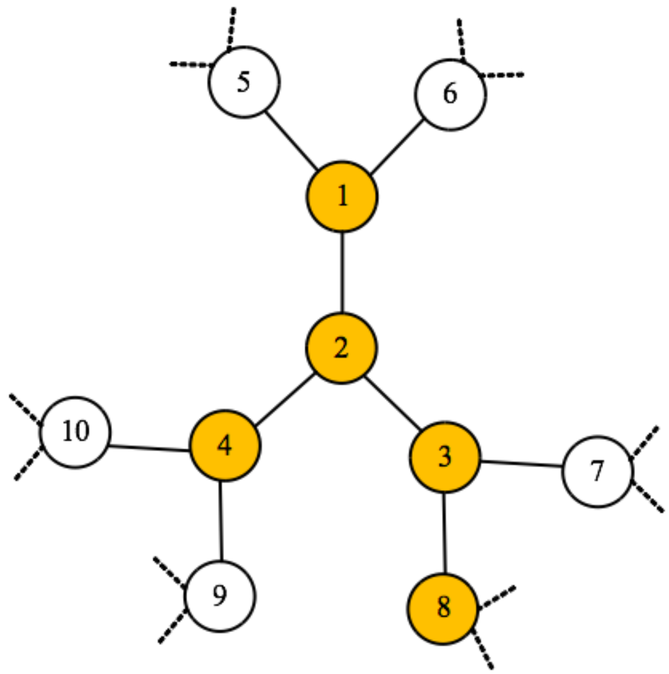
- Node 2 has the highest centrality score

Rumor centrality



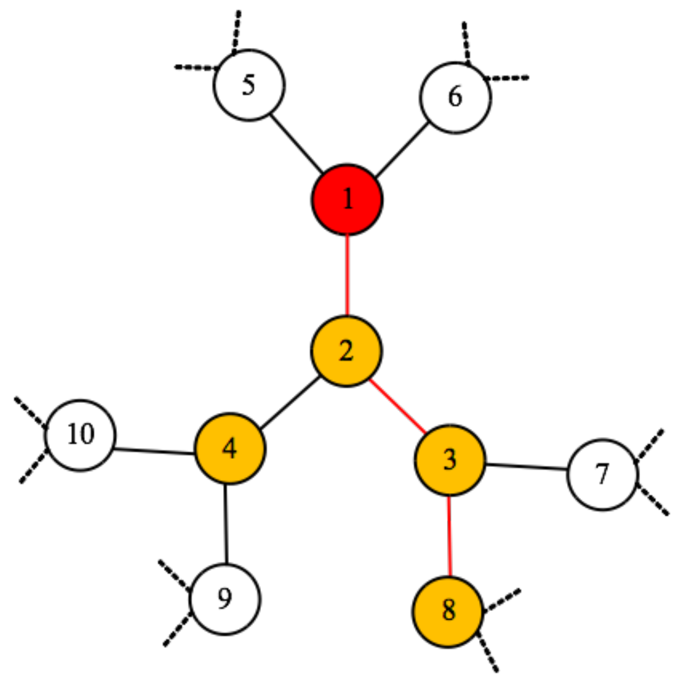
- Same as picking node with: smallest sum of distances to all nodes

Jordan centrality

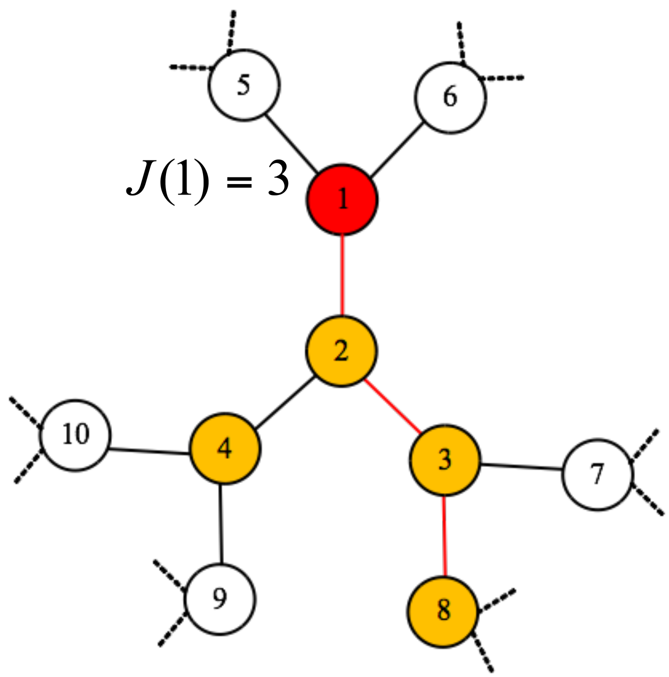


- Maximum distance from a node to another

Jordan centrality

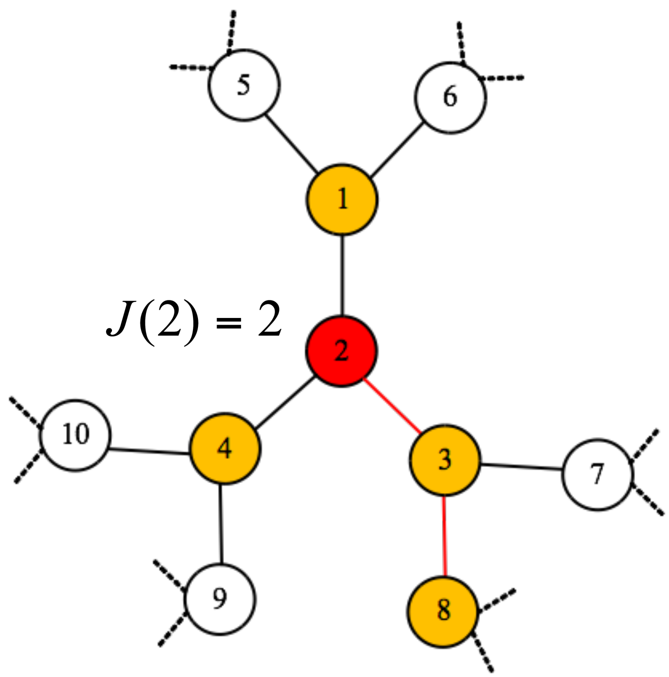


Jordan centrality

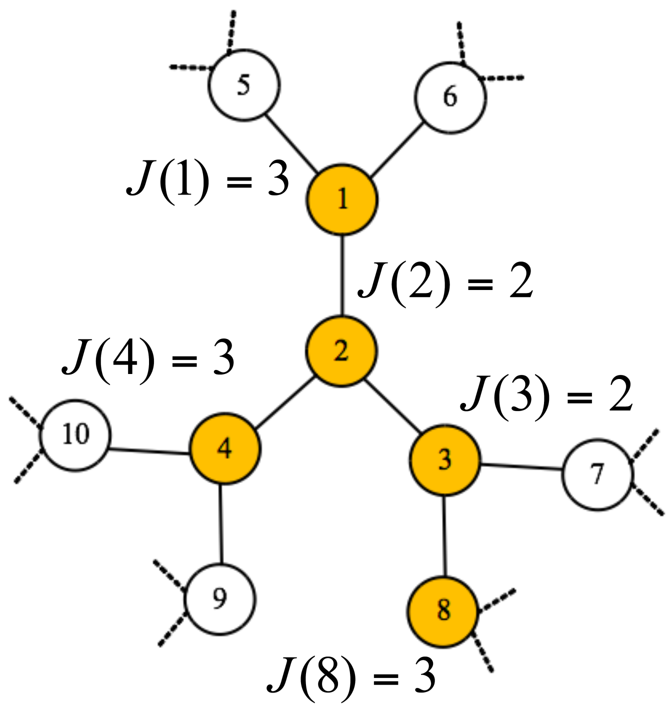


- Node 1's eccentricity is 3

Jordan centrality

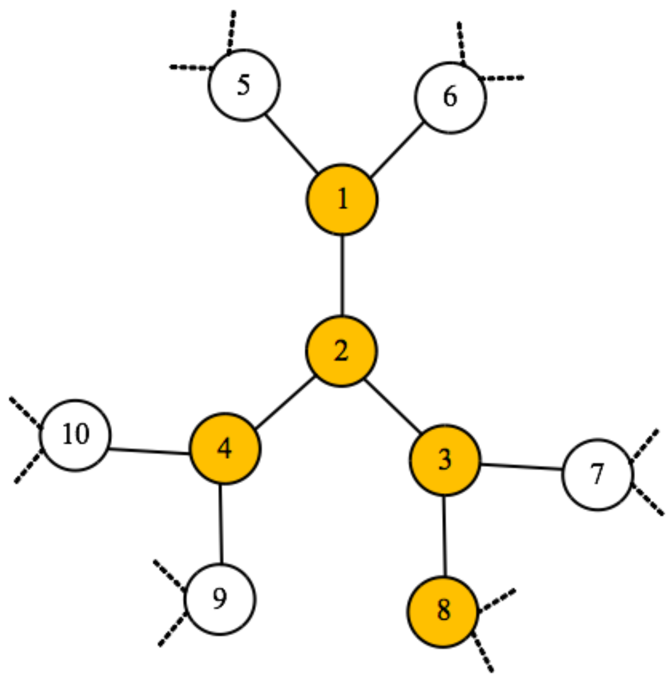


Jordan centrality



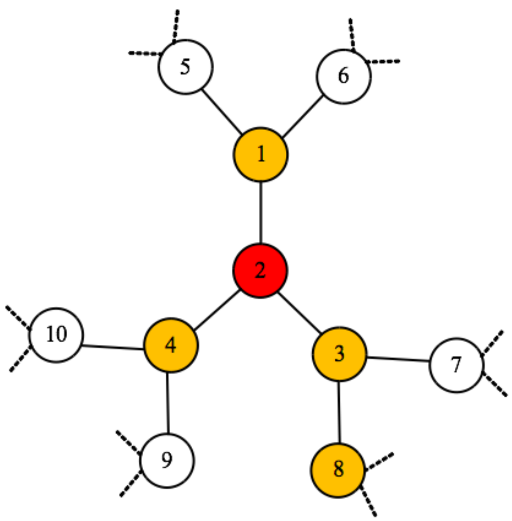
- Both nodes 2 and 3 are equally central

Counting Efficiently



- Naive counting is very inefficient

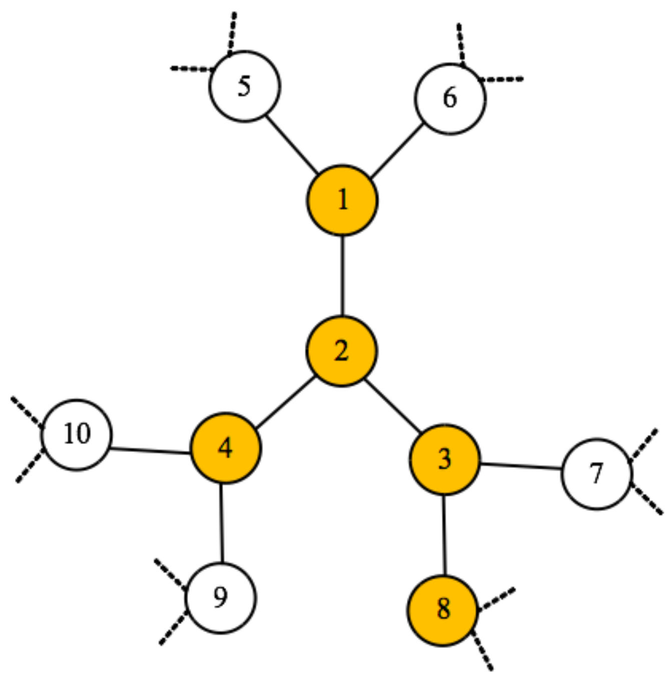
Naïve implementation of rumor centrality



$2 \to 1 \to 4 \to 3 \to 8$	$2 \to 4 \to 1 \to 8 \to 3$
$2 \to 1 \to 3 \to 4 \to 8$	$2 \to 1 \to 4 \to 8 \to 3$
$2 \to 1 \to 3 \to 8 \to 4$	$2 \to 4 \to 8 \to 1 \to 3$
$2 \to 4 \to 1 \to 3 \to 8$	$2 \to 4 \to 8 \to 3 \to 1$
$2 \to 4 \to 3 \to 1 \to 8$	$2 \to 1 \to 8 \to 4 \to 3$
$2 \to 4 \to 3 \to 8 \to 1$	$2 \to 1 \to 8 \to 3 \to 4$
$2 \to 3 \to 1 \to 4 \to 8$	$2 \to 8 \to 4 \to 1 \to 3$
$2 \to 3 \to 1 \to 8 \to 4$	$2 \to 8 \to 1 \to 3 \to 4$
$2 \to 3 \to 8 \to 1 \to 4$	$2 \to 8 \to 1 \to 4 \to 3$
$2 \to 3 \to 8 \to 4 \to 1$	$2 \to 8 \to 3 \to 1 \to 4$
$2 \to 3 \to 4 \to 1 \to 8$	$2 \to 8 \to 3 \to 4 \to 1$
$2 \to 3 \to 4 \to 8 \to 1$	$2 \to 8 \to 4 \to 3 \to 1$

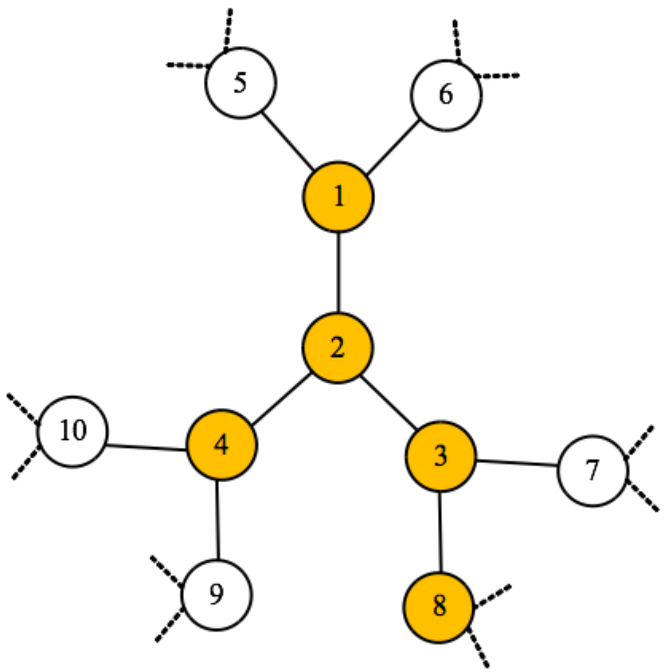
- Some orderings are valid, others not

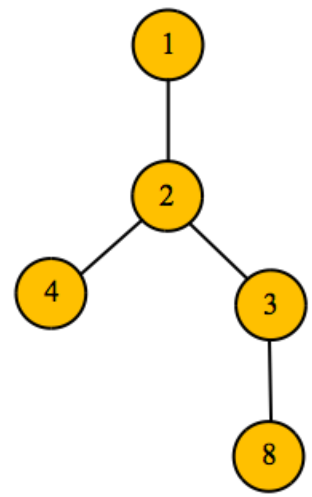
Rumor centrality via message passing



- Reuse computations

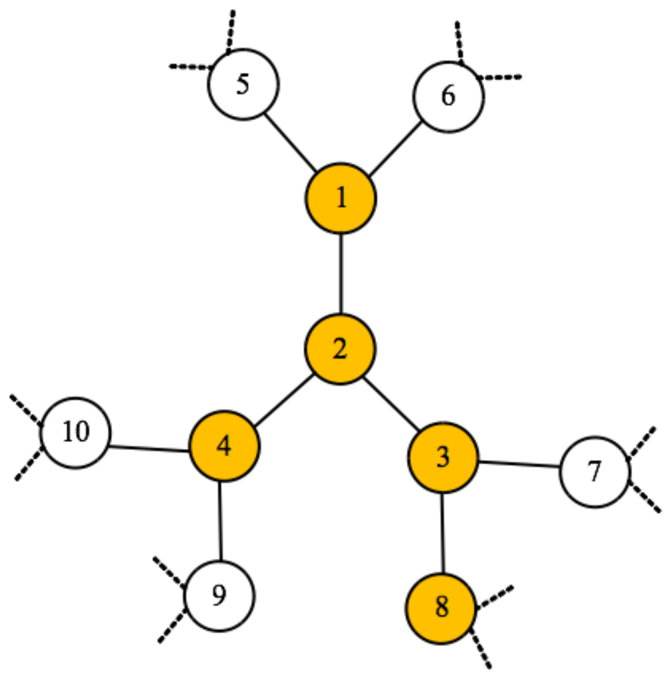
Rumor centrality via message passing





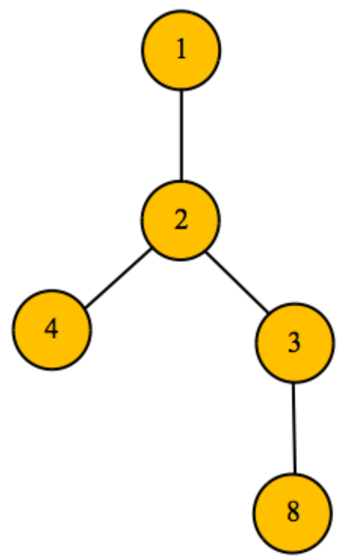
- Start with a node (1, say) and form a rooted tree

Rumor centrality via message passing



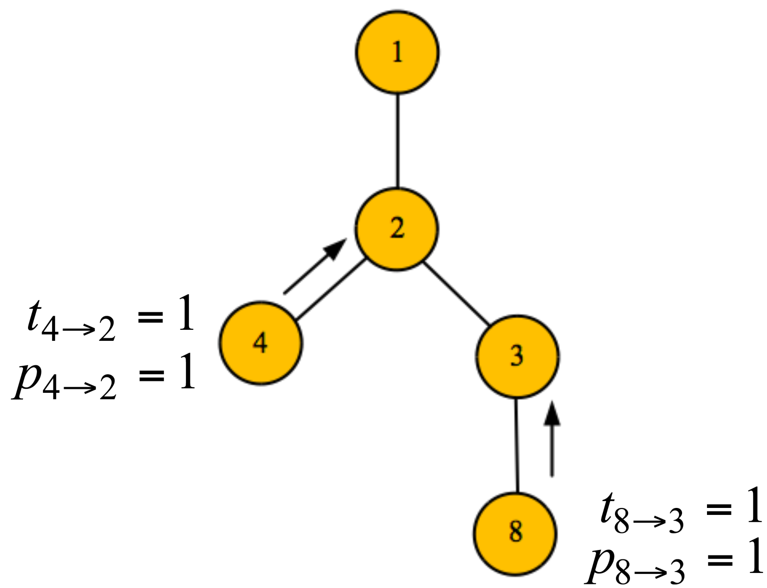
- Tree rooted at node 2

Upward pass



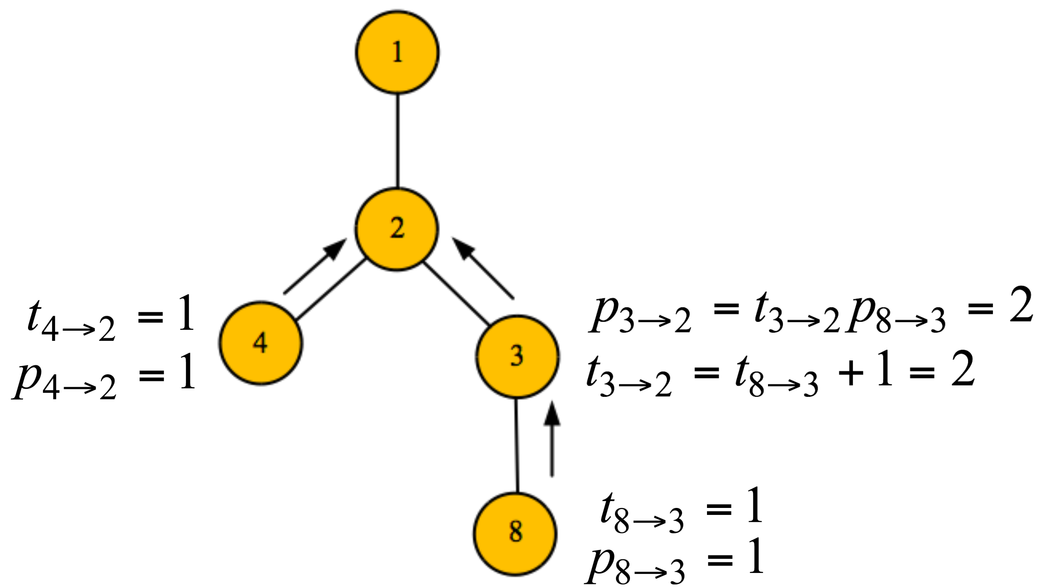
- Messages pass upwards from leaves to the root

Upward pass



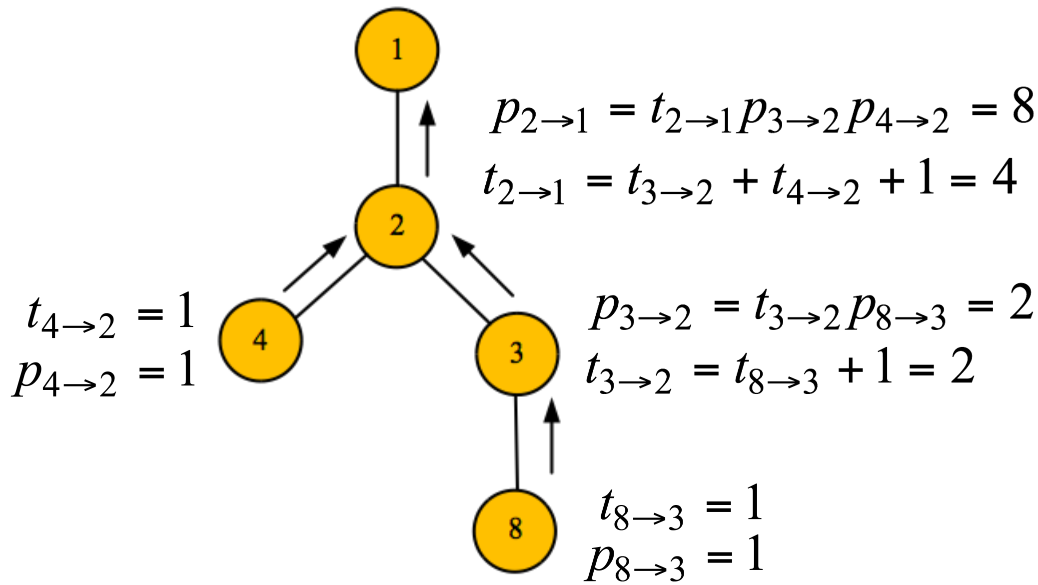
- Two types of messages

Upward pass



- Node 3 processes its message and sends it to its parent

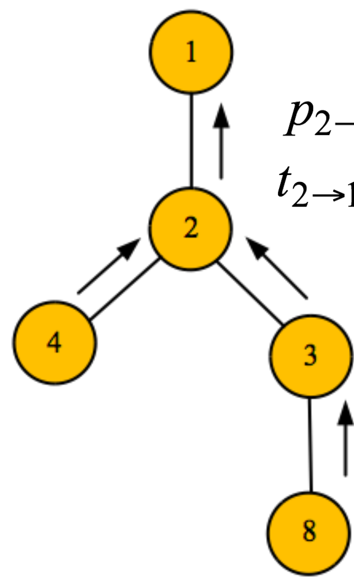
Upward pass



- Node 2 can now process its message and send it

Upward pass

$$R(1) = \frac{N!}{Np_{2\to1}} = \frac{5!}{5\times8} = 3$$



$$p_{2\to1} = t_{2\to1}p_{3\to2}p_{4\to2} = 8$$

$$t_{2\to1} = t_{3\to2} + t_{4\to2} + 1 = 4$$

$$t_{4\to2} = 1$$

$$p_{4\to2} = 1$$

$$p_{3\to2} = t_{3\to2}p_{8\to3} = 2$$

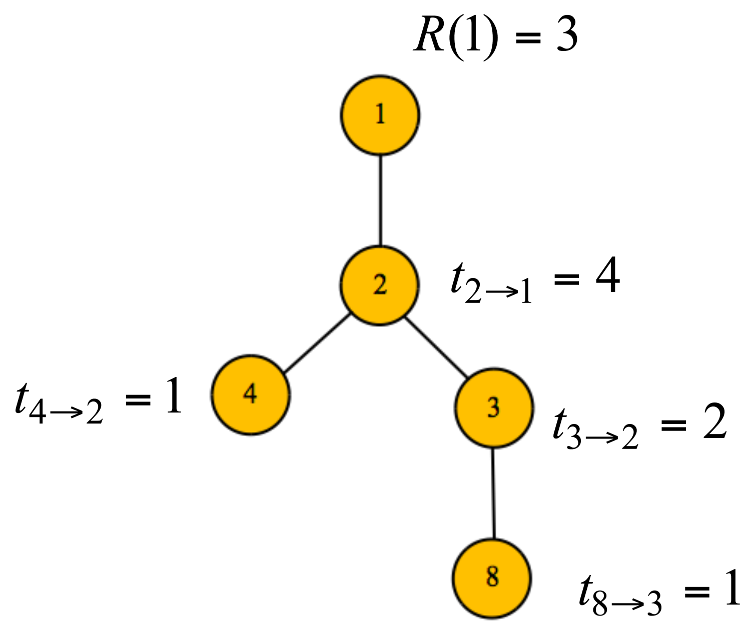
$$t_{3\to2} = t_{8\to3} + 1 = 2$$

$$t_{8\to3} = 1$$

$$p_{8\to3} = 1$$

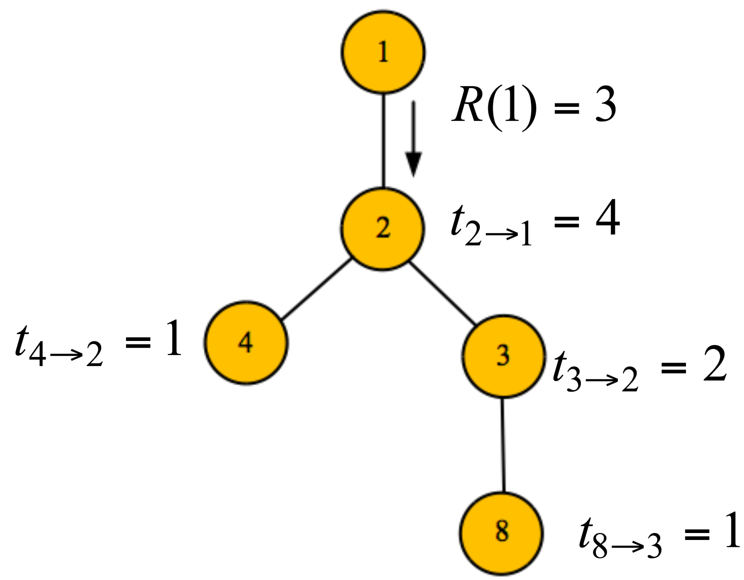
- Node 1 gets to calculate its rumor centrality score

Downward pass



- Messages pass downwards from root

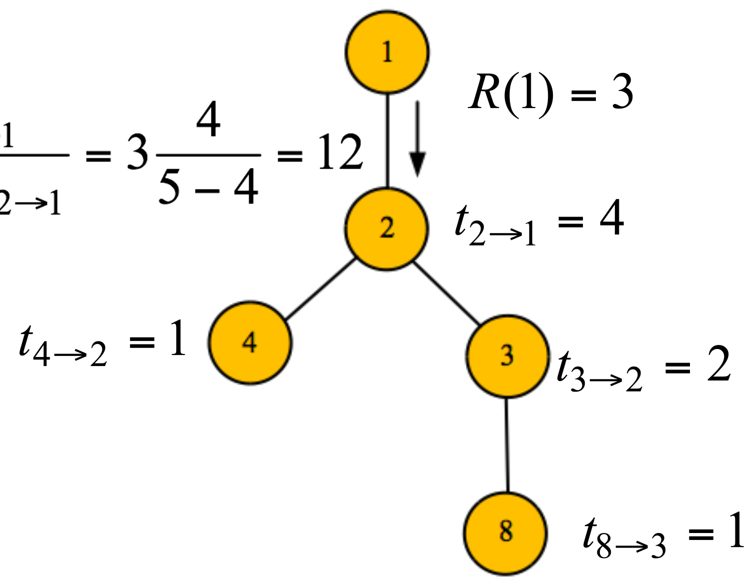
Downward pass



- Pass the rumor centrality score downwards

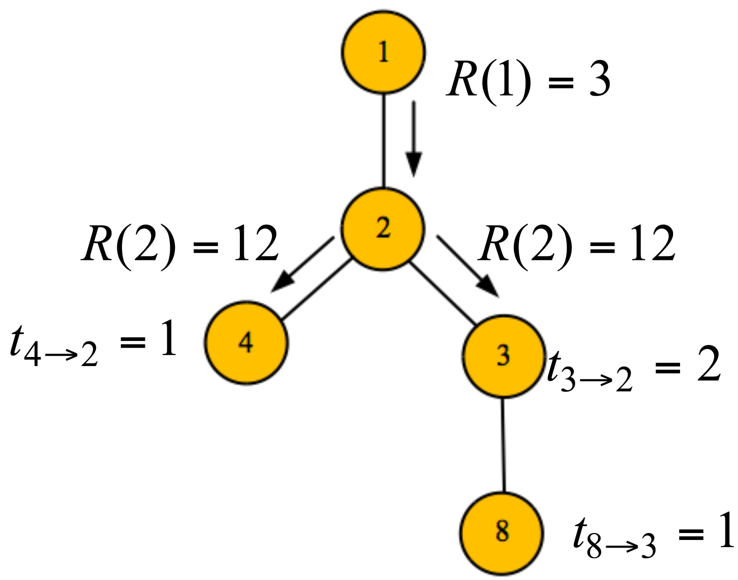
Downward pass

$$R(2) = R(1)\frac{t_{2\to1}}{N - t_{2\to1}} = 3\frac{4}{5-4} = 12$$

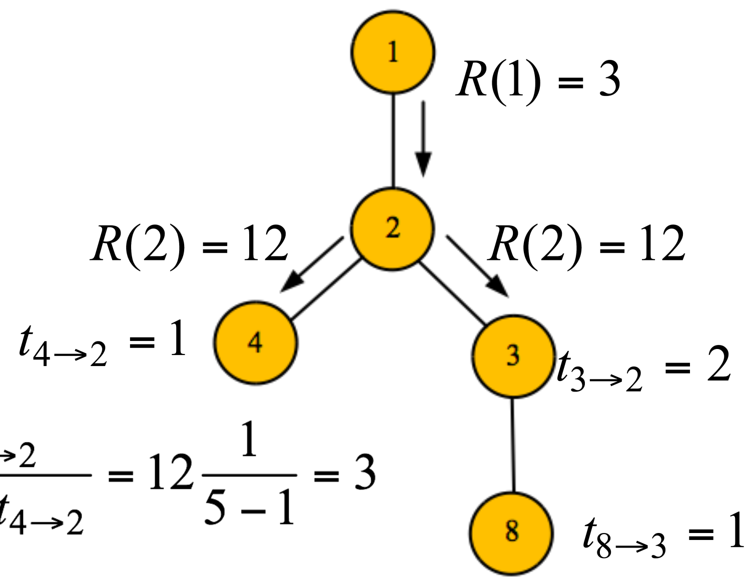


- Node 2 can compute its rumor centrality score

Downward pass



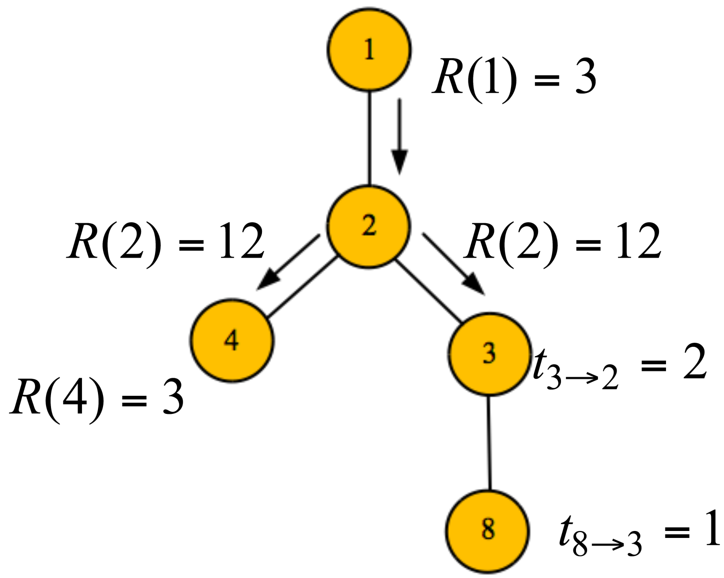
Downward pass



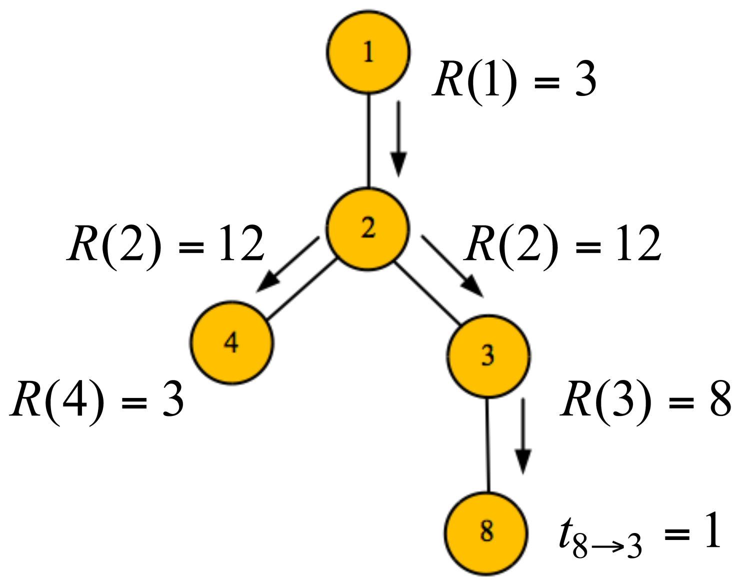
$$R(4) = R(2)\frac{t_{4\to2}}{N - t_{4\to2}} = 12\frac{1}{5-1} = 3$$

Downward pass

$$R(3) = R(2)\frac{t_{3\to 2}}{N - t_{3\to 2}} = 12\frac{2}{5-2} = 8$$

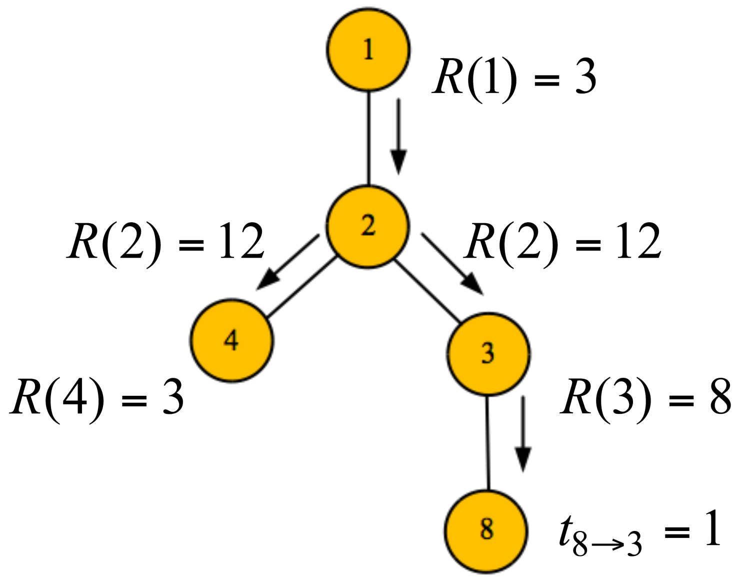


Downward pass

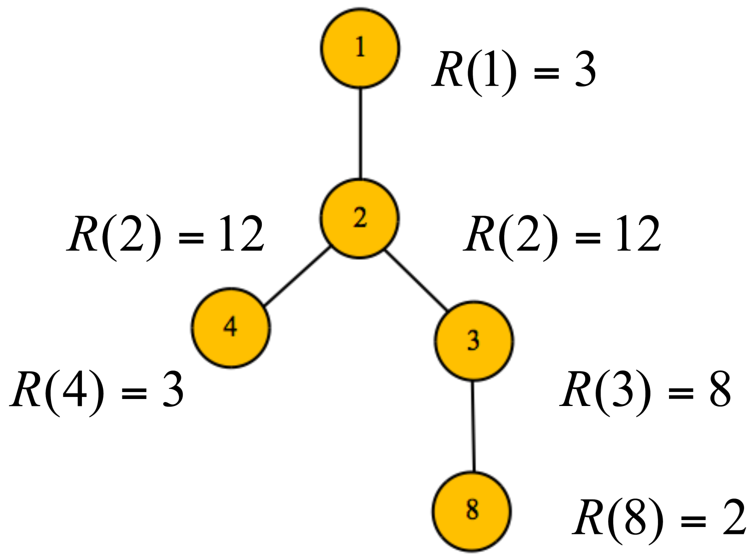


Downward pass

$$R(8) = R(3)\frac{t_{8\to3}}{N - t_{8\to3}} = 8\frac{1}{5-1} = 2$$

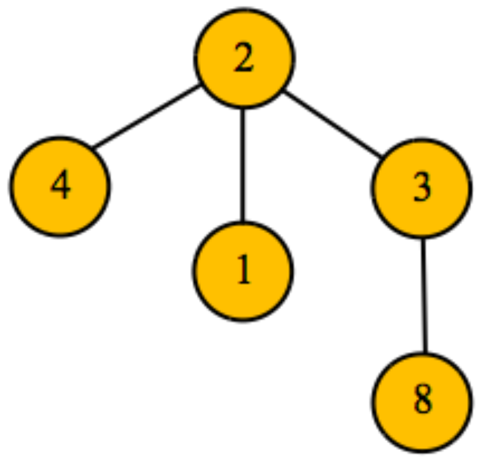


Computational complexity



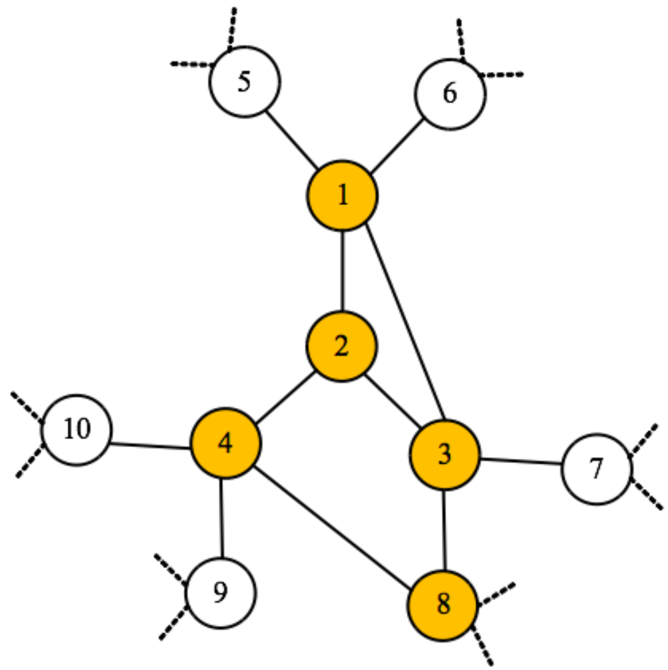
- 3N computations

Choice of root node



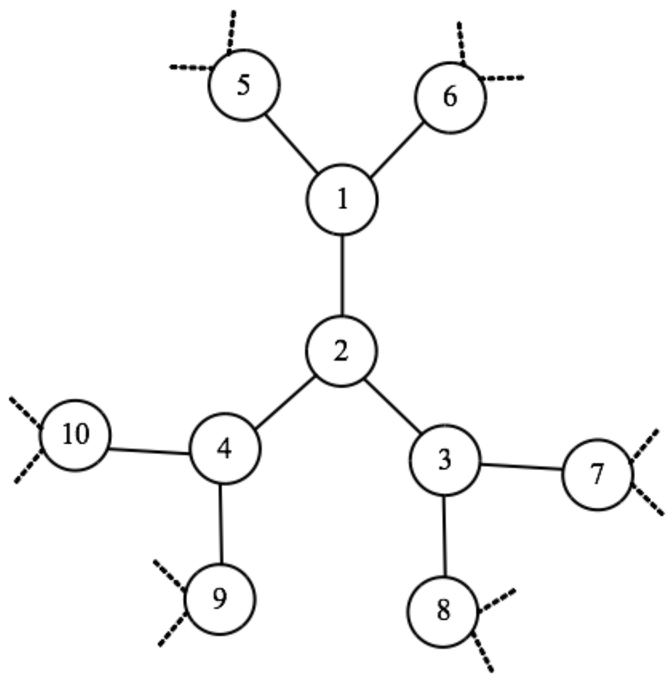
- Root node could have been 2
- Rumor centrality scores remain the same

Graphs with cycles?



- Heuristic: spreading occurs on a breadth-first tree

Regular tree



- Theorem: Rumor centrality = Maximum Likelihood
- Positive probability of detection, asymptotically

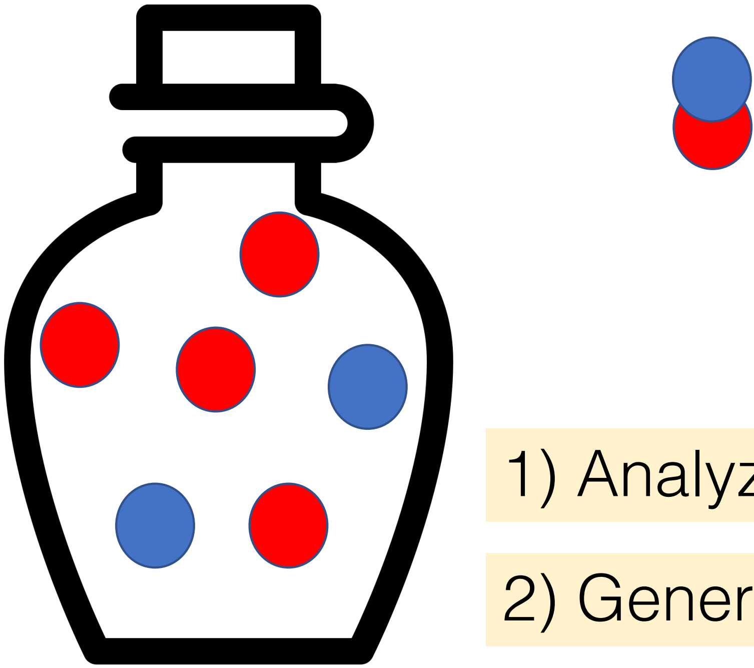
Shah and Zaman, *Rumor Centrality: A Universal Source Detector*, Sigmetrics 2012

Analyzing Diffusion Processes

Pólya Urns and More

Introduction to Pólya Urns

What is the fraction of red balls after n draws?

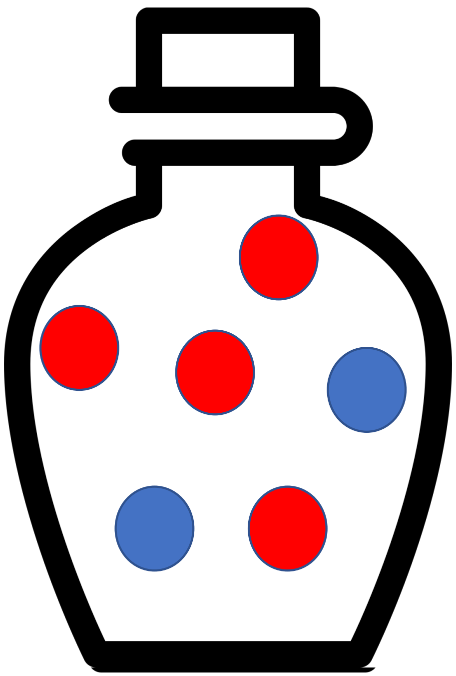


1) Analyze for 2 colors.

2) Generalize

Mahmoud, *Polya Urn Models*, CRC Press 2008

Does the order of draws matter?



$$\frac{1}{2} \quad \frac{2}{3} \quad \frac{1}{4} \quad \frac{3}{5} = \frac{3!\,1!}{5!}$$

$$\frac{1}{2} \quad \frac{1}{3} \quad \frac{2}{4} \quad \frac{3}{5} = \frac{3!\,1!}{5!}$$

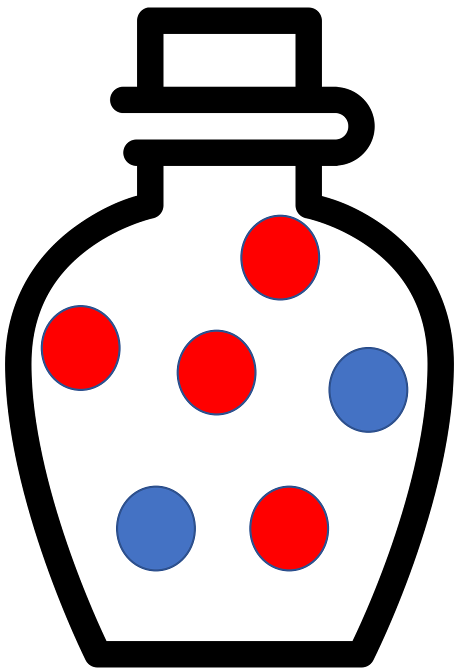
$$\mathrm{P}(r_n = k + 1) = \binom{\mathrm{n}}{\mathrm{k}} \beta(k + 1, n + 1 - k)$$

red balls at nth draw

$$\beta(x, y) = \int_0^1 m^{x-1}(1 - m)^{y-1} dm$$

Does the fraction of red balls converge?

r_n: Number of red balls $\quad$ R_n: Fraction of red balls $\quad$ $R_n = \frac{r_n}{n+2}$

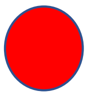


Approach

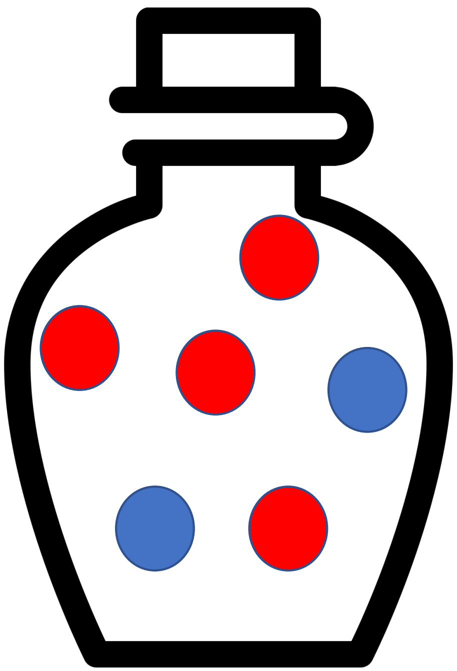
1) R_n is a martingale.

2) That martingale converges a.s.

1) R_n is a martingale.

 r_n: Number of red balls

R_n: Fraction of red balls $\quad R_n = \dfrac{r_n}{n+2}$



Fraction red balls

Num red balls +1

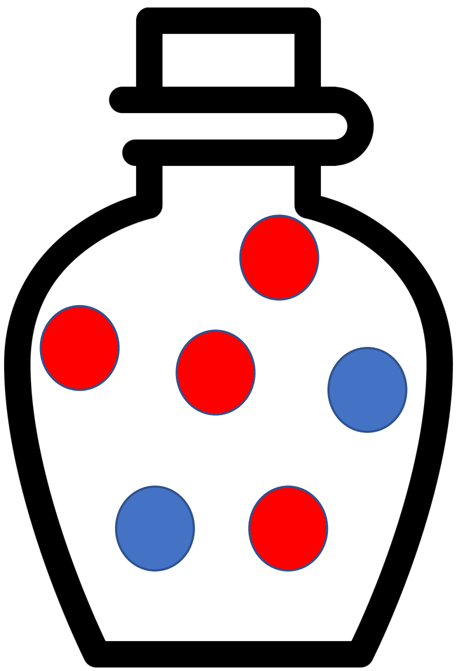
Fraction blue balls

Num red balls

$$E[R_n \mid R_{n-1}, \ldots, R_1] =$$

$$= \frac{R_{n-1} + r_{n-1}}{n+2} = \frac{r_{n-1}(n+2)}{(n+1)(n+2)} = R_{n-1}$$

2) This martingale converges a.s.



Martingale Convergence Theorem

$$R_n \in (0,1)$$

$$\rightarrow R(\omega) = \lim_{n\to\infty} R_n(\omega)$$

What is the limiting distribution?

Let's look at the moment-generating function

$$M_{R_n}(t) = E[\exp(tR_n)]$$

$$= \sum_{k=0}^{n} \exp(t\frac{k+1}{n+2})\mathrm{P}(R_n = \frac{k+1}{n+2})$$

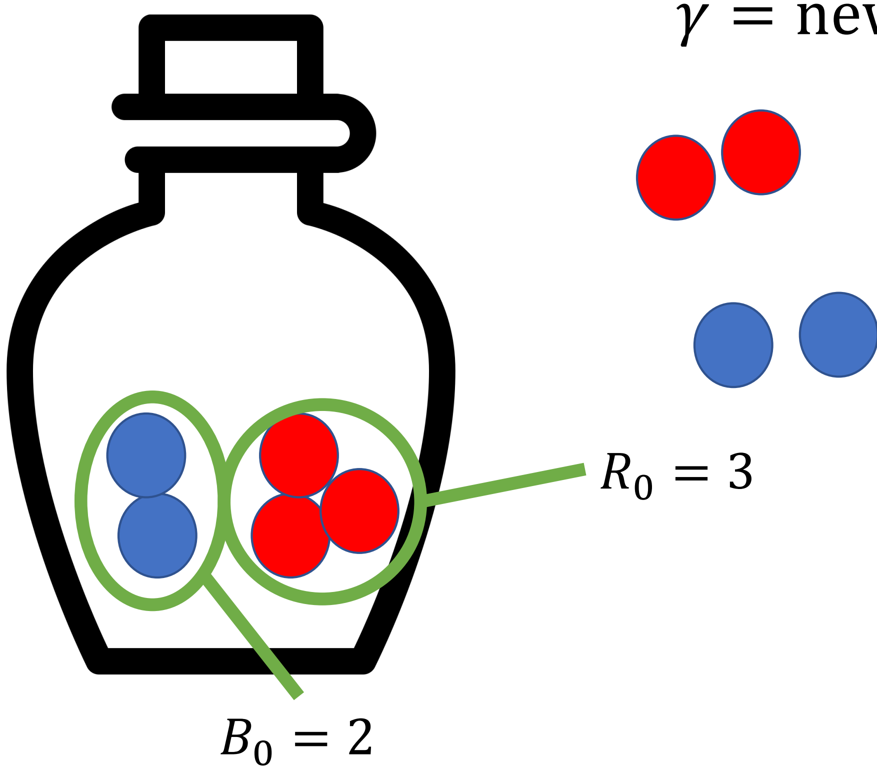
$$= \sum_{k=0}^{n} \exp\left(t\frac{k+1}{n+2}\right)\int_0^1 \binom{n}{k} m^k(1-m)^{n-k}\, dm$$

$$\xrightarrow[n\to\infty]{} \int_0^1 e^{tm}\, dm \qquad = \begin{cases} \dfrac{e^t - 1}{t}, & x \neq 0 \\ 1, & x = 0 \end{cases}$$

Moment-generating function of $\mathrm{Unif}(0,1)$

Generalization 1: Number of replacements

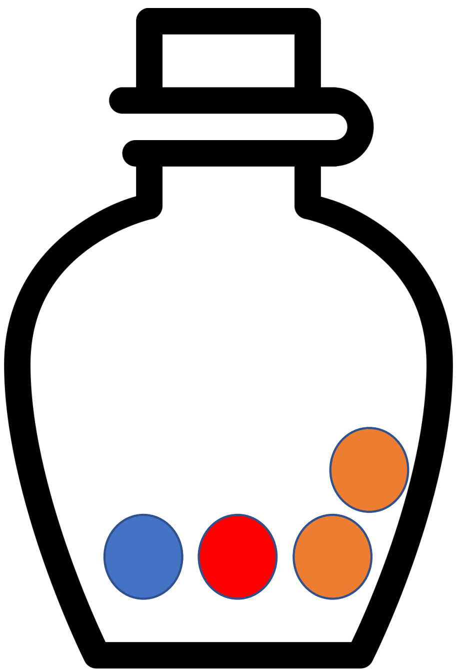
γ = new balls added of same color



$$R \sim \mathrm{Beta}\left(\frac{R_0}{\gamma}, \frac{B_0}{\gamma}\right)$$

Depends on initial conditions!

Generalization 2: Number of classes

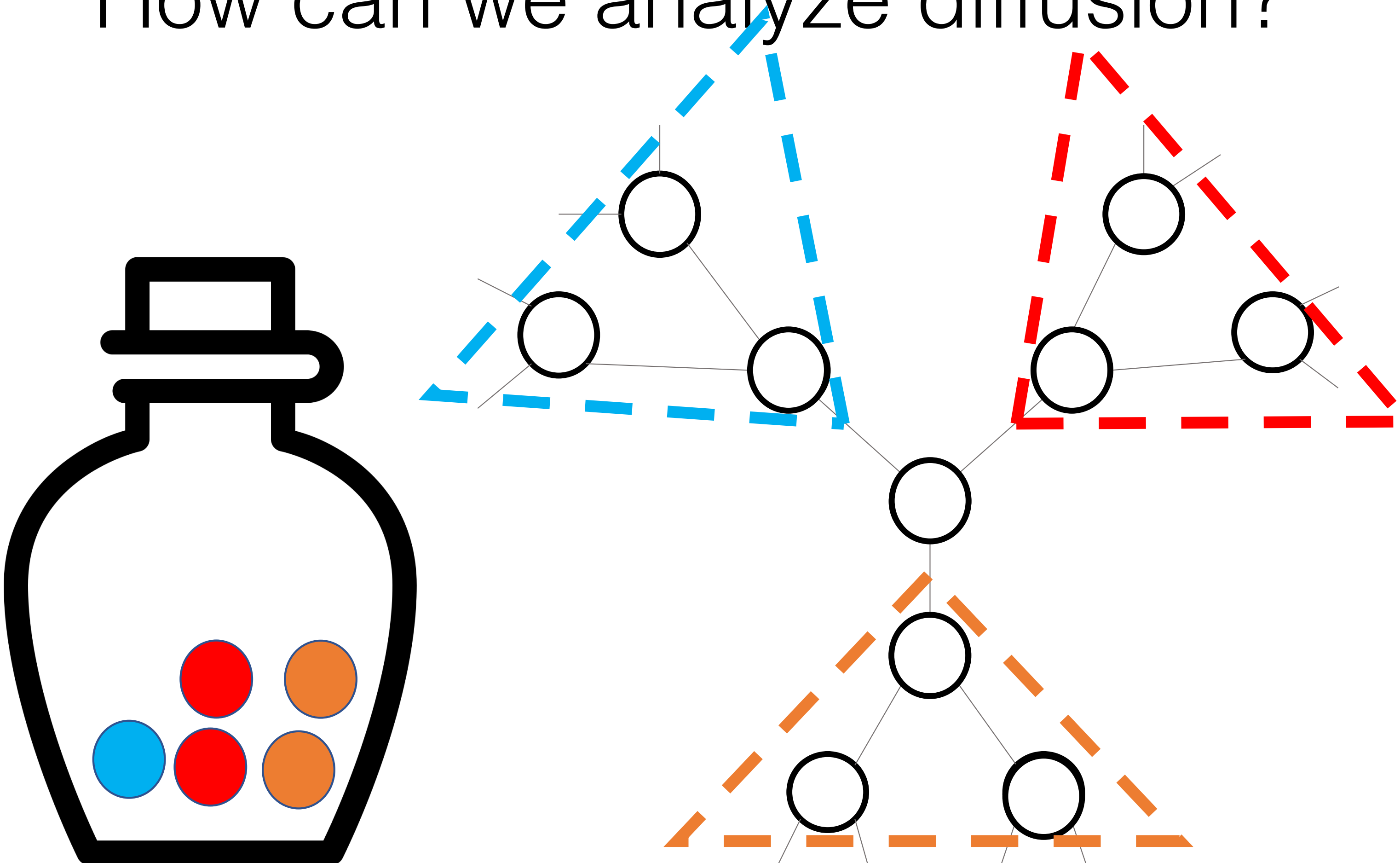


$\boldsymbol{\alpha} = [1\ 1\ 2]$ Initial values

$\gamma = 2$ # added balls of same color

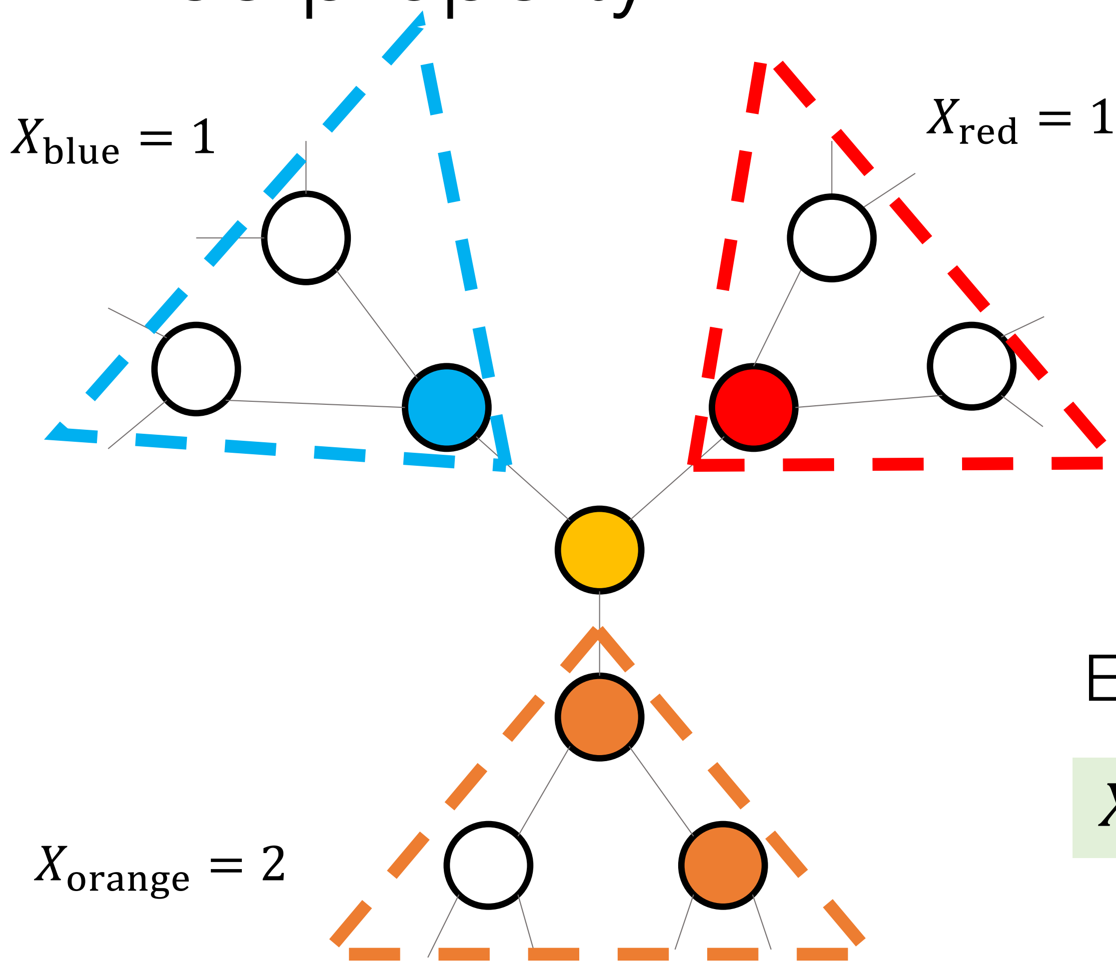
$$R \sim \mathrm{Dirichlet-Multinomial}(\boldsymbol{\alpha}, n)$$

How can we analyze diffusion?



Shah and Zaman, *Rumor Centrality: A Universal Source Detector*, 2012

A nice property



v is a rumor center iff

$$\mathrm{X_i}(T) \leq \frac{X_{\text{total}}}{2}$$

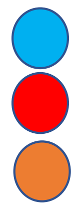
$\mathrm{i} \in \{\text{red, orange, blue}\}$

Number of "infected" nodes

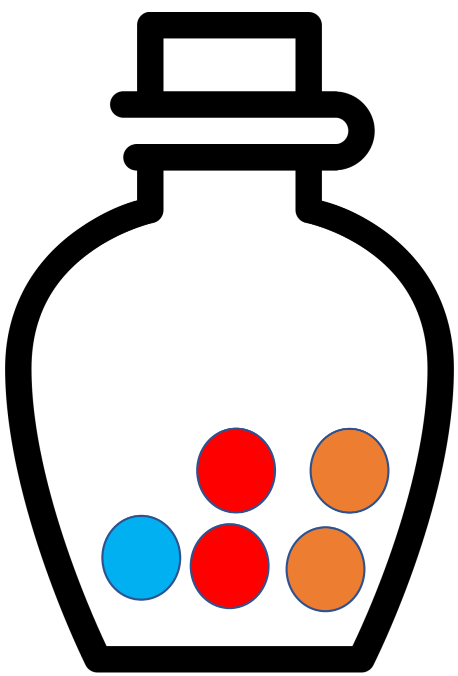
Example:

$$X_{\text{orange}} = 2 \leq \frac{n}{2} = \frac{5}{2}$$

What does this mean for our urn?



B_n: Fraction of
R_n: Fraction of
O_n: Fraction of

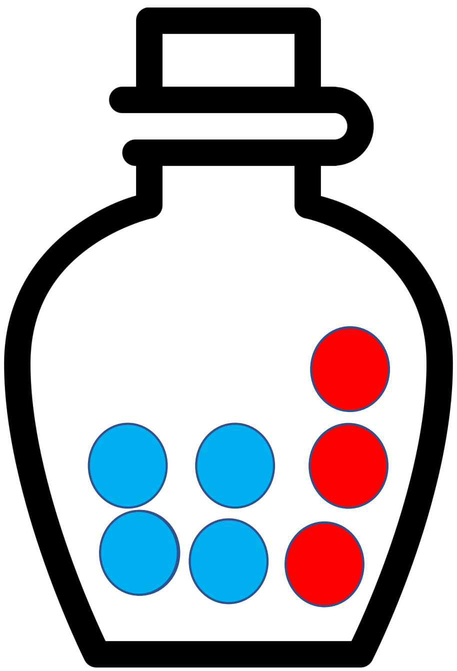


v is a rumor center iff

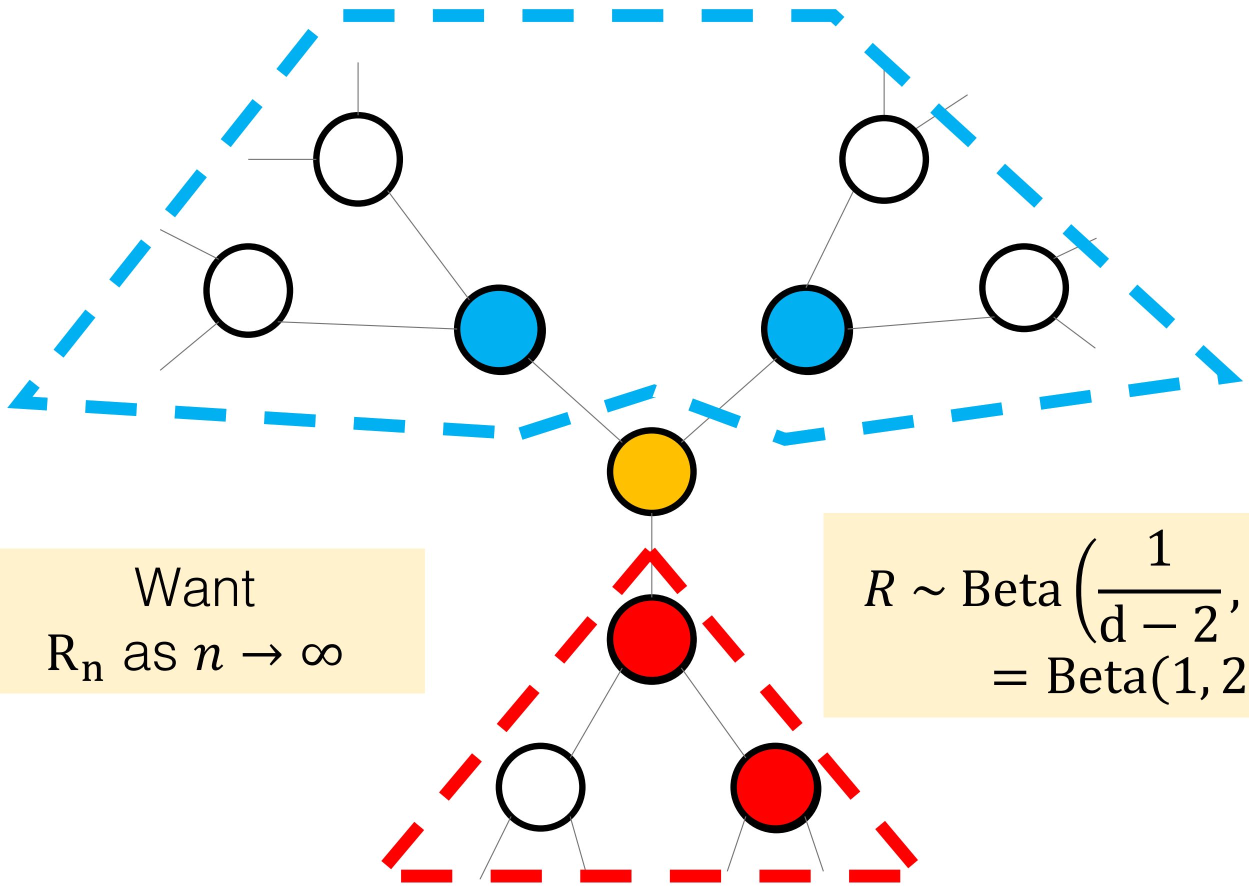
$$B_n, R_n, O_n \leq \frac{1}{2}$$

Let's use the convergence results from before.

Let's consider a slightly different urn.



Want
R_n as $n \to \infty$



$$R \sim \mathrm{Beta}\left(\frac{1}{d-2}, \frac{d-1}{d-2}\right) = \mathrm{Beta}(1,2)$$

Putting it all together

$$R \sim \text{Beta}\left(\frac{1}{\text{d}-2}, \frac{\text{d}-1}{\text{d}-2}\right)$$

Want $R \leq \frac{1}{2}$

$$I_{\frac{1}{2}}(a, b) \triangleq P(X \in [0, ½]) \text{ where } X \sim \text{Beta}(a, b)$$

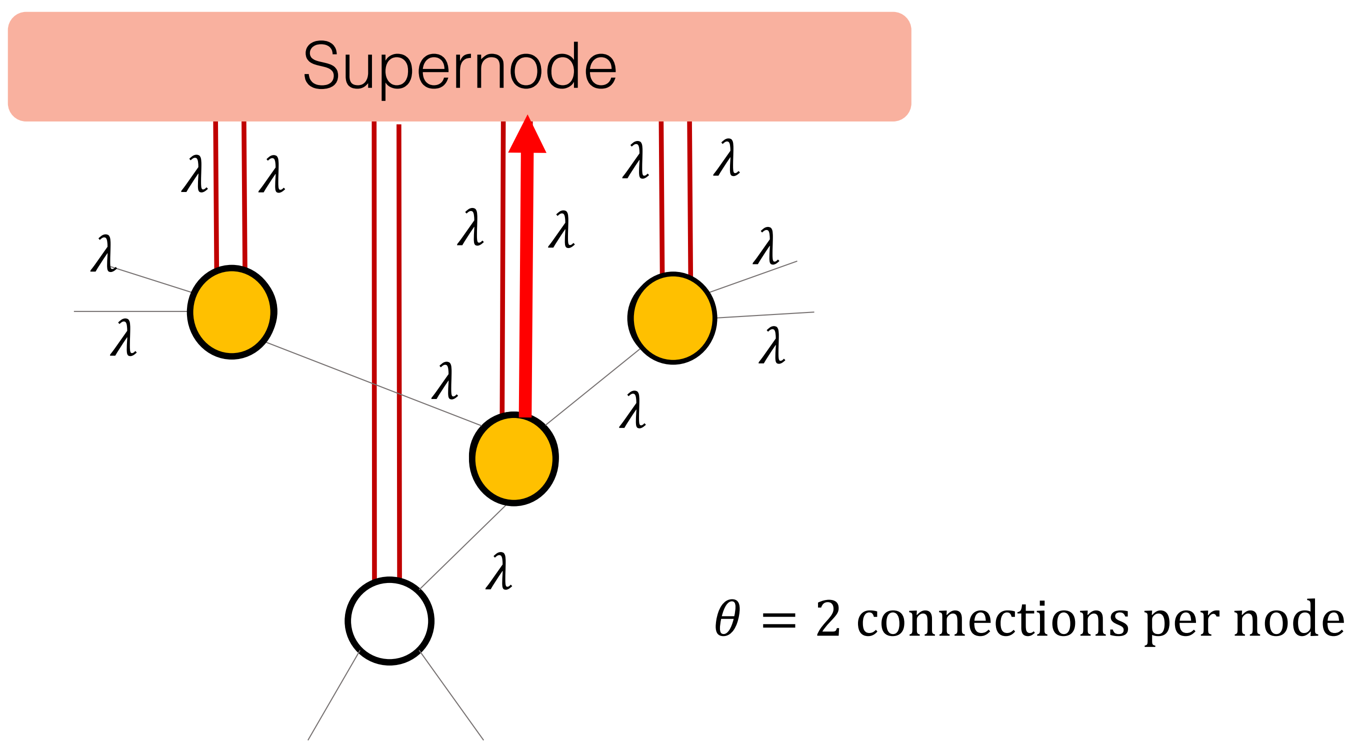
$$\lim_{t\to\infty} P(\text{detection}) = 1 - d(1 - I_{\frac{1}{2}}\left(\frac{1}{d-2}, \frac{d-1}{d-2}\right))$$

Example: $(d = 3) \rightarrow \lim_{t\to\infty} P(\text{detection}) = 0.25$

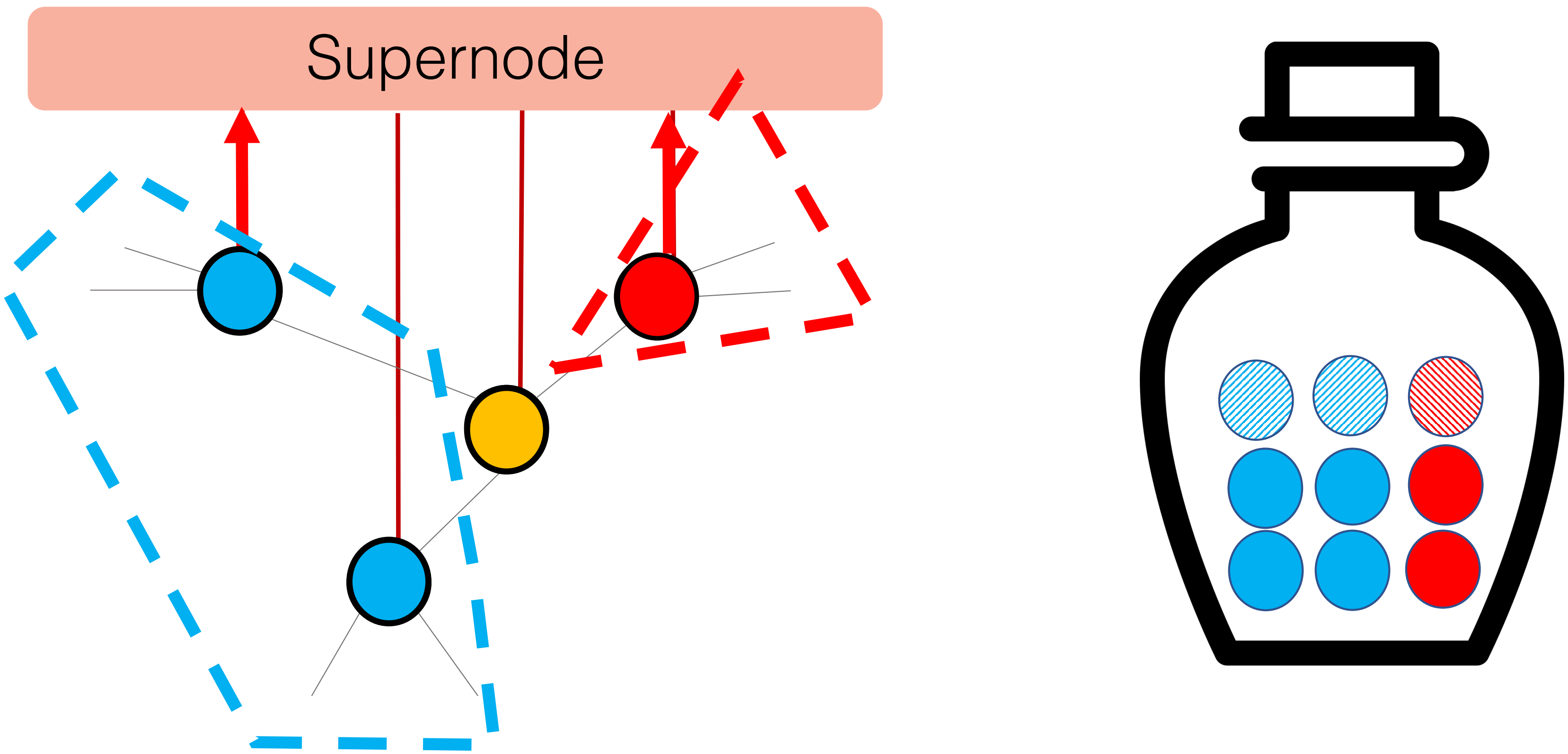
Rumor centrality: A Universal Source Detector, Shah and Zaman, 2012

What about other problems?

Eavesdropper Adversary



Let’s model this as an urn

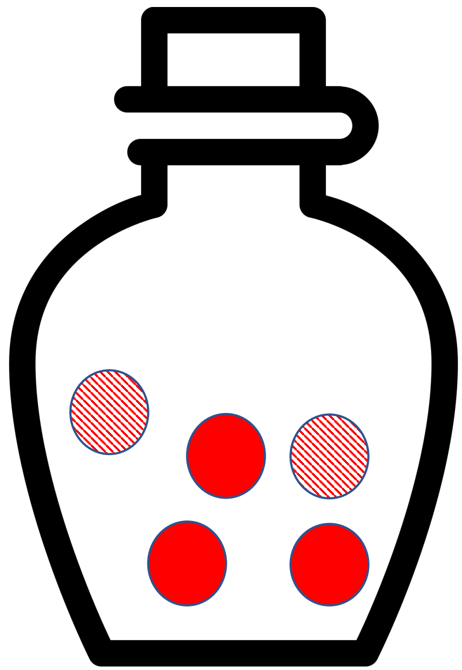


Generalized Polya Urns

Replacement Matrix

$$\begin{array}{cc} & \begin{array}{cc} \text{Solid} & \text{Striped} \end{array} \\ A = & \begin{bmatrix} d-2 & 1 \\ 0 & -1 \end{bmatrix} \begin{array}{l} \text{Solid} \\ \text{Striped} \end{array} \end{array}$$

Example



$$A = \begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix}$$

Convergence properties

$$A = \begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix}$$

Conditions

1) $A_{ij} \geq 0$ for $i \neq j$ and $A_{ii} \geq -1$
2) Largest real eigenvalue of A (λ_1) is
   1) positive
   2) simple
3) Start with ≥ 1 ball of a *dominating type*

Example

1) $A_{ij} \geq 0$ and $A_{ii} \geq -1$
2) $\lambda(A) = \{1,\text{-}1\}$
3) Solids are a *dominating type*

$$\begin{pmatrix} R_n \\ 1 - R_n \end{pmatrix} \xrightarrow{a.s.} \lambda_1 \; v_1$$

Fraction of solid balls

Fraction of striped balls

First eigenvalue

First (right) eigenvector

Athreya and Ney 1972, Jansen 2003

Comparing the two results

Classic Pólya Urns

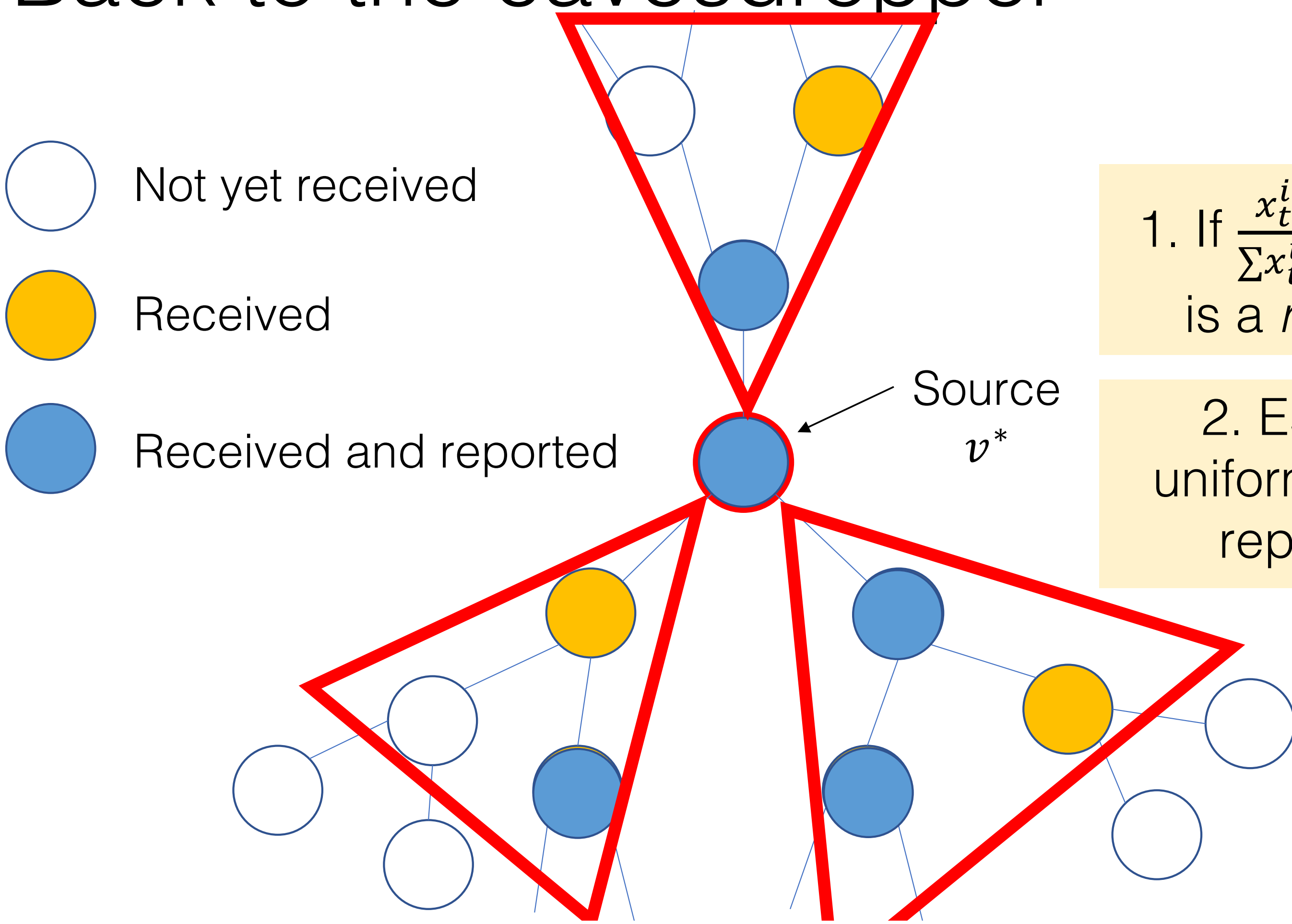
- Transition matrix
  - Nonsingular
  - **Not positive regular**
- $A = \begin{bmatrix} d-2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d-2 \end{bmatrix}$
- Converges to a **random variable** (Beta distribution)

Generalized Pólya Urns

- Transition matrix
  - Nonsingular
  - **Positive regular**
- $A = \begin{bmatrix} d-2 & 1 \\ 0 & -1 \end{bmatrix}$
- Converges to a **constant**

Back to the eavesdropper

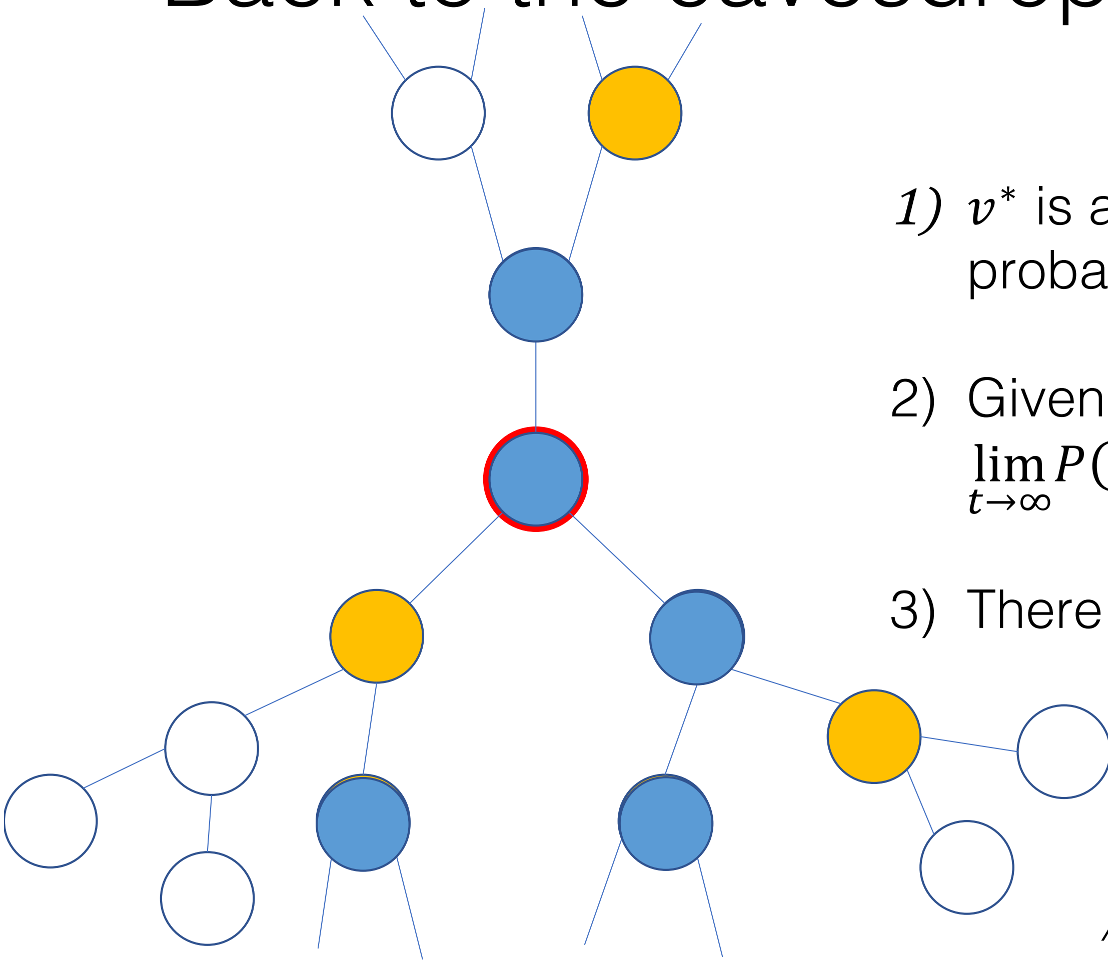
$x_t^i(v)$= # blue balls in ith subtree of v at time t



1. If $\frac{x_t^i(v)}{\sum x_t^i(v)} < \frac{1}{2}$, $\forall i$, then v is a *reporting source*.

2. Estimate $\hat{v}$ drawn uniformly from the set of reporting sources.

Back to the eavesdropper



Proof Sketch

1) v^* is a rumor center with known probability.

2) Given that v^* is a rumor center, $\lim_{t \to \infty} P(v^* \text{ is a reporting center}) = 1$

Uses urn results

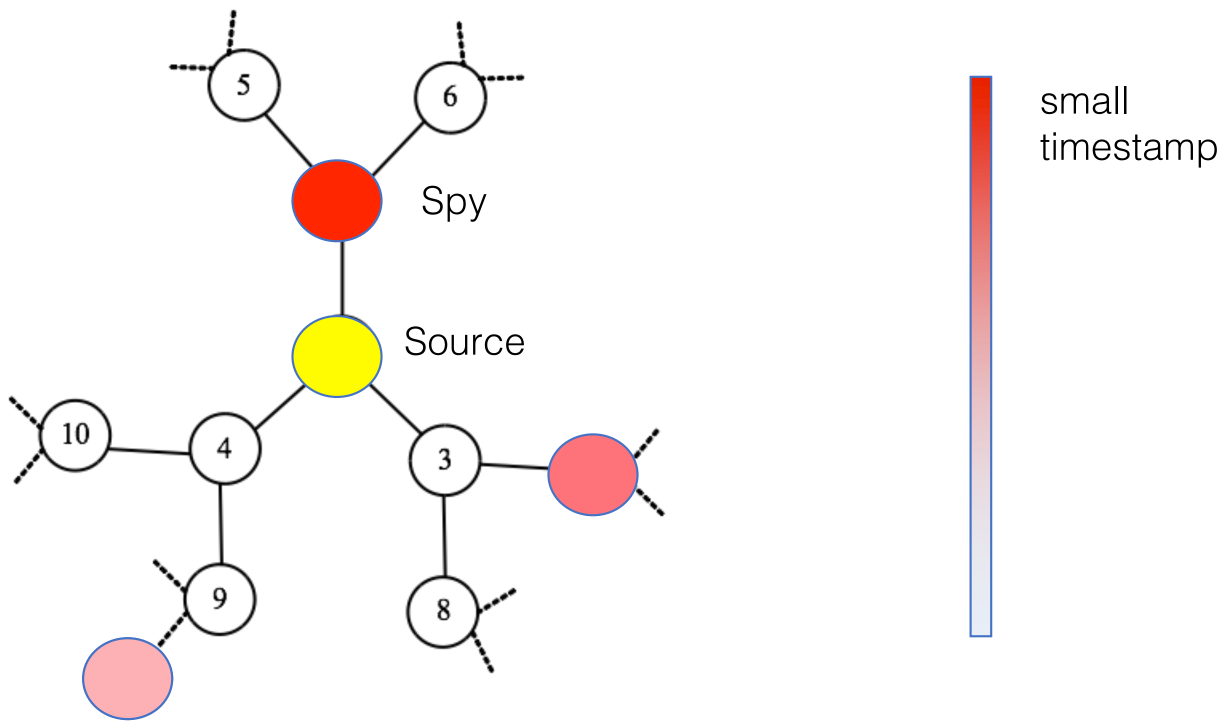
3) There is at most 1 reporting center.

Anonymity Properties of the Bitcoin P2P Network, 2017

Summary of Approach

- Extract a representation of the problem that can be modeled as a **Pólya Urn**
- Use known convergence results (Athreya and Ney 1972, Jansen 2003)

Spy Adversary



- Spy nodes observe time stamps

Centrality methods

- First spy estimator
    - source = node reporting earliest to spies
    - very easy to implement
    - no knowledge of underlying graph

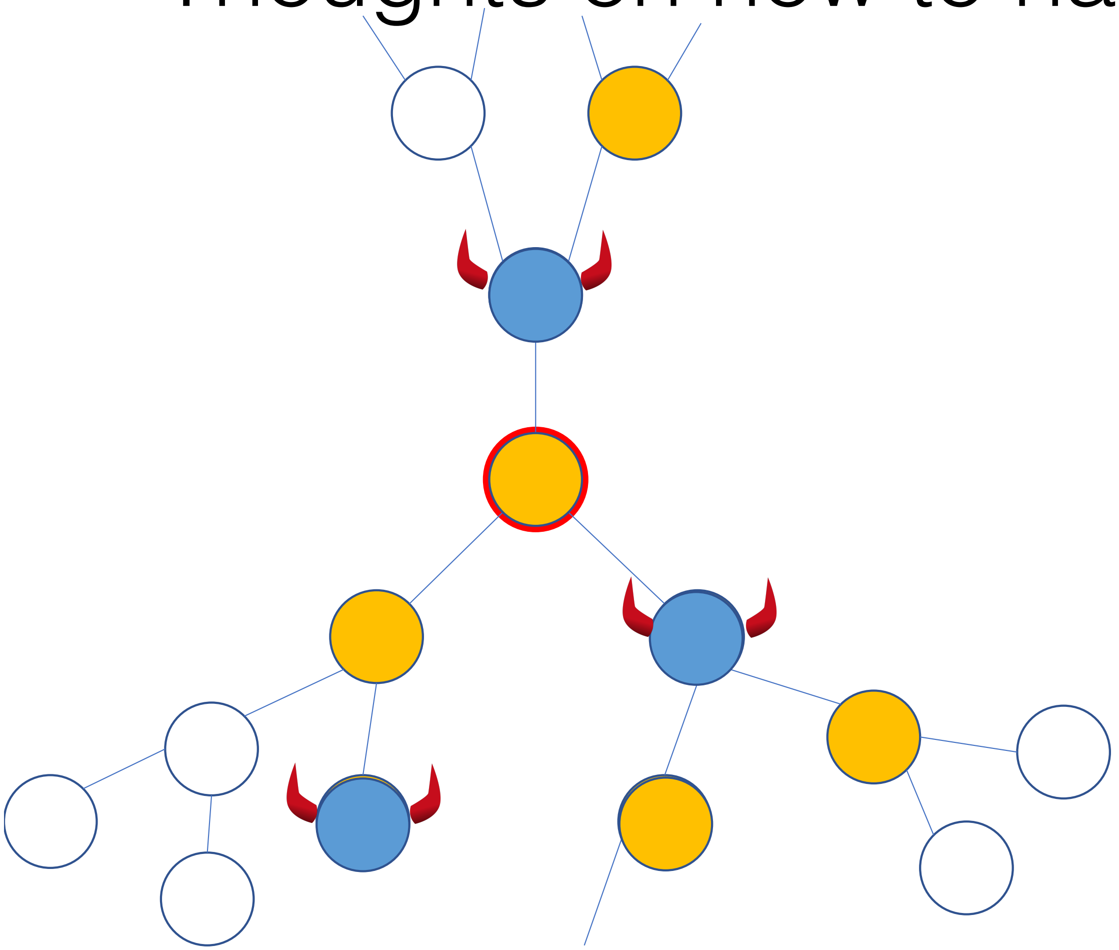
Centrality methods

- Earliest infection time estimator [Zhu, Chen, Ying, 2014]
  - estimate infection times of other nodes
  - eccentricity score =

$$\min_{\mathcal{T} \in \mathcal{P}_v} \min_{(u,v) \in \mathcal{T}} \sum_{u,v,\mu} (t_u - t_v - \mu)^2$$

  - pick node with smallest eccentricity
  - related estimator [Pinto, Thiran, Vetterli, 2012]

Thoughts on how to handle spies



- Use the same counting-based estimator
- Use randomized Polya urns

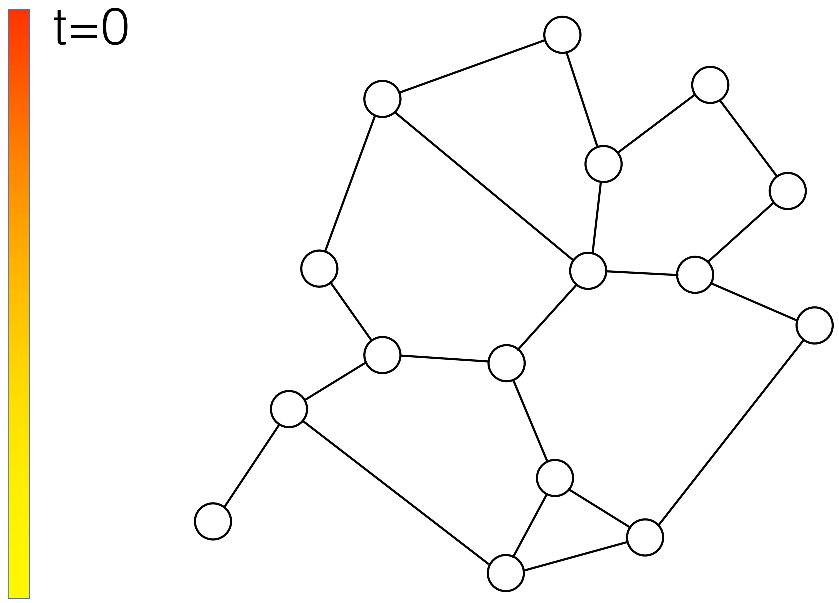
Open Problems

Moving Forward

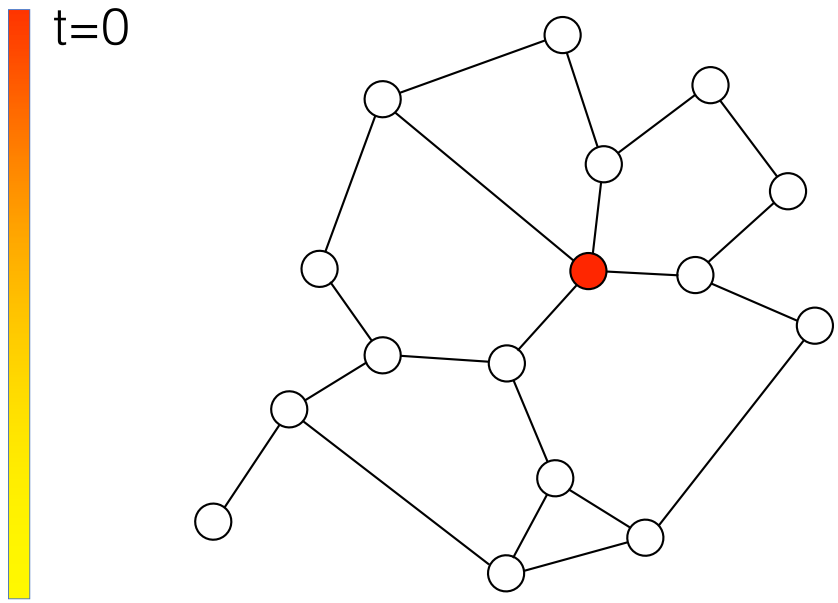
Other related questions

- Number of sources
- Detecting more than one source
- Combination of adversaries: snapshot+eavesdropper+spy
- Inferring the underlying network

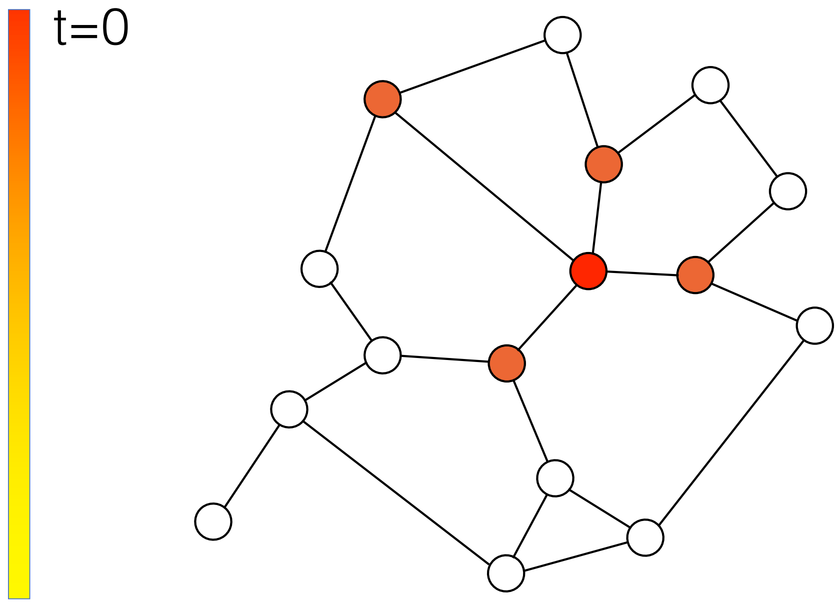
Inferring diffusion networks



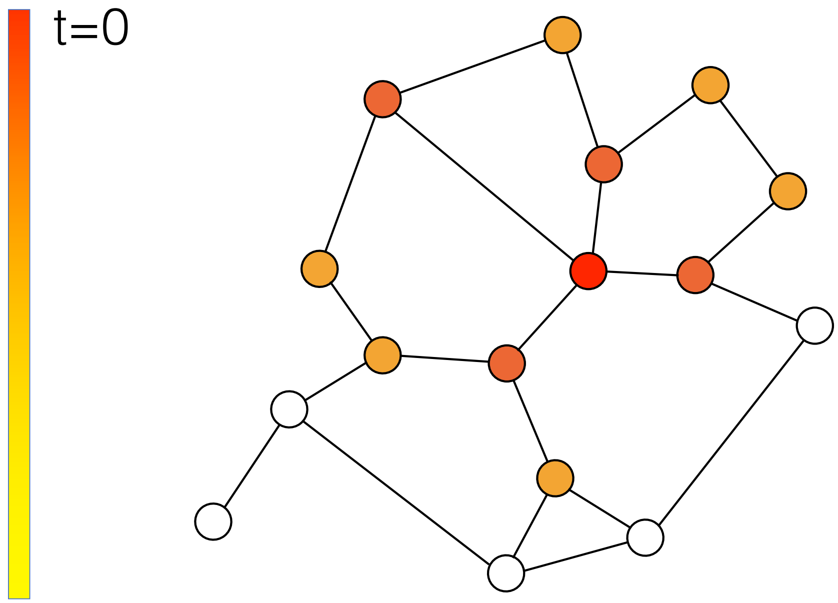
Inferring diffusion networks



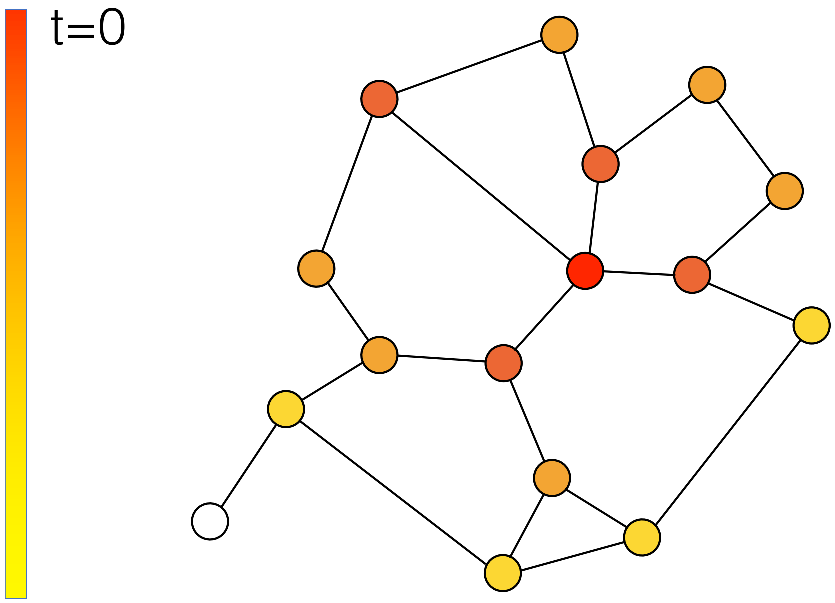
Inferring diffusion networks



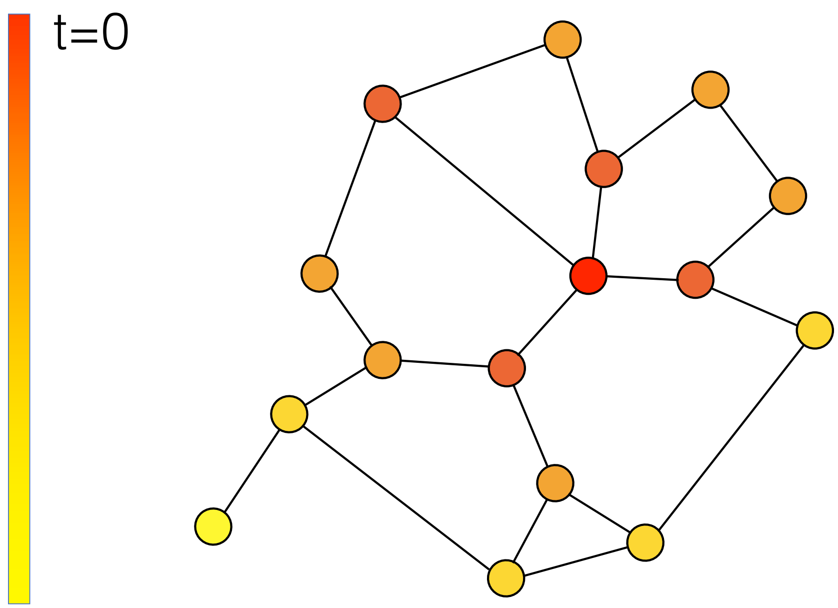
Inferring diffusion networks



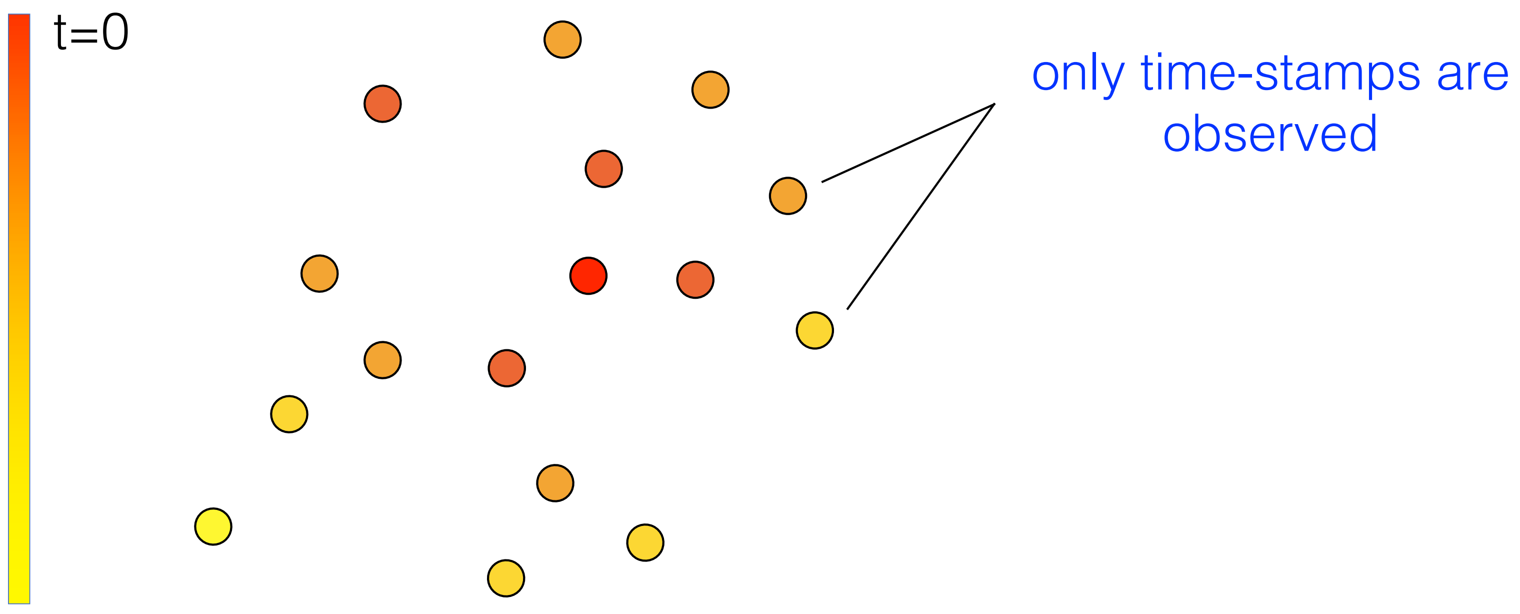
Inferring diffusion networks



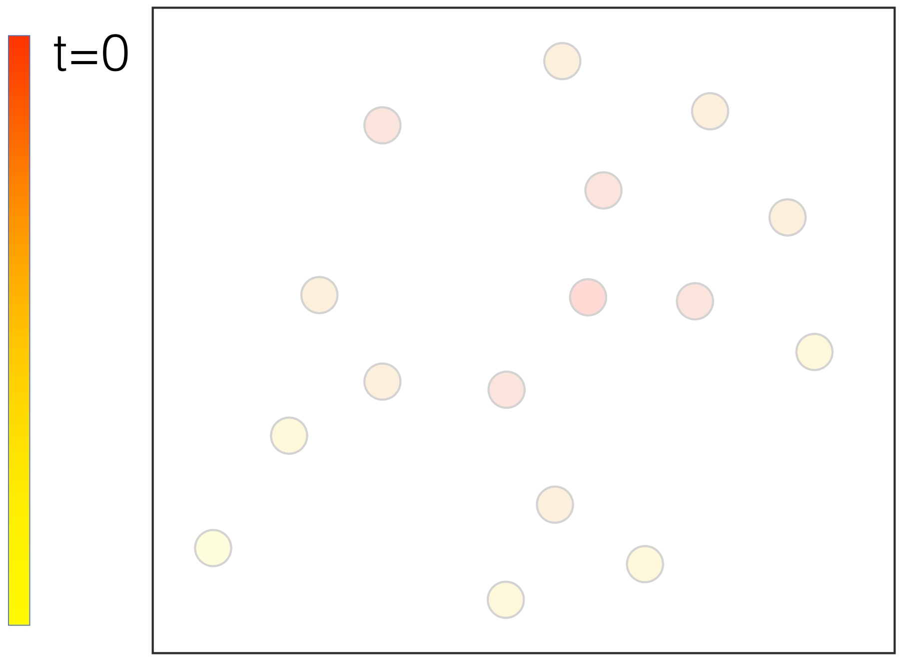
Inferring diffusion networks



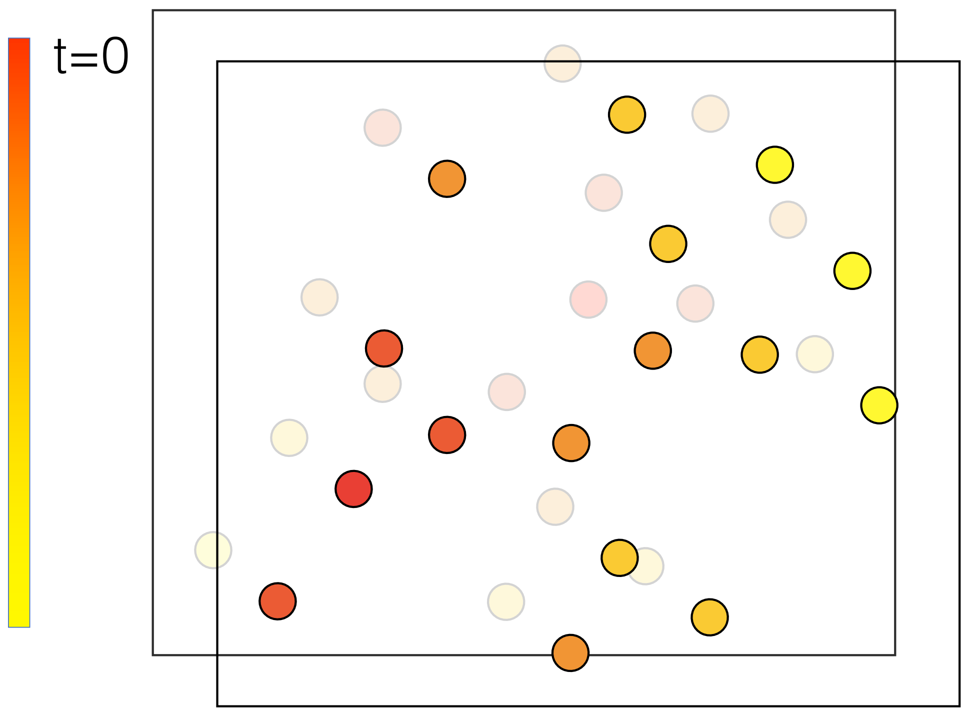
Inferring diffusion networks



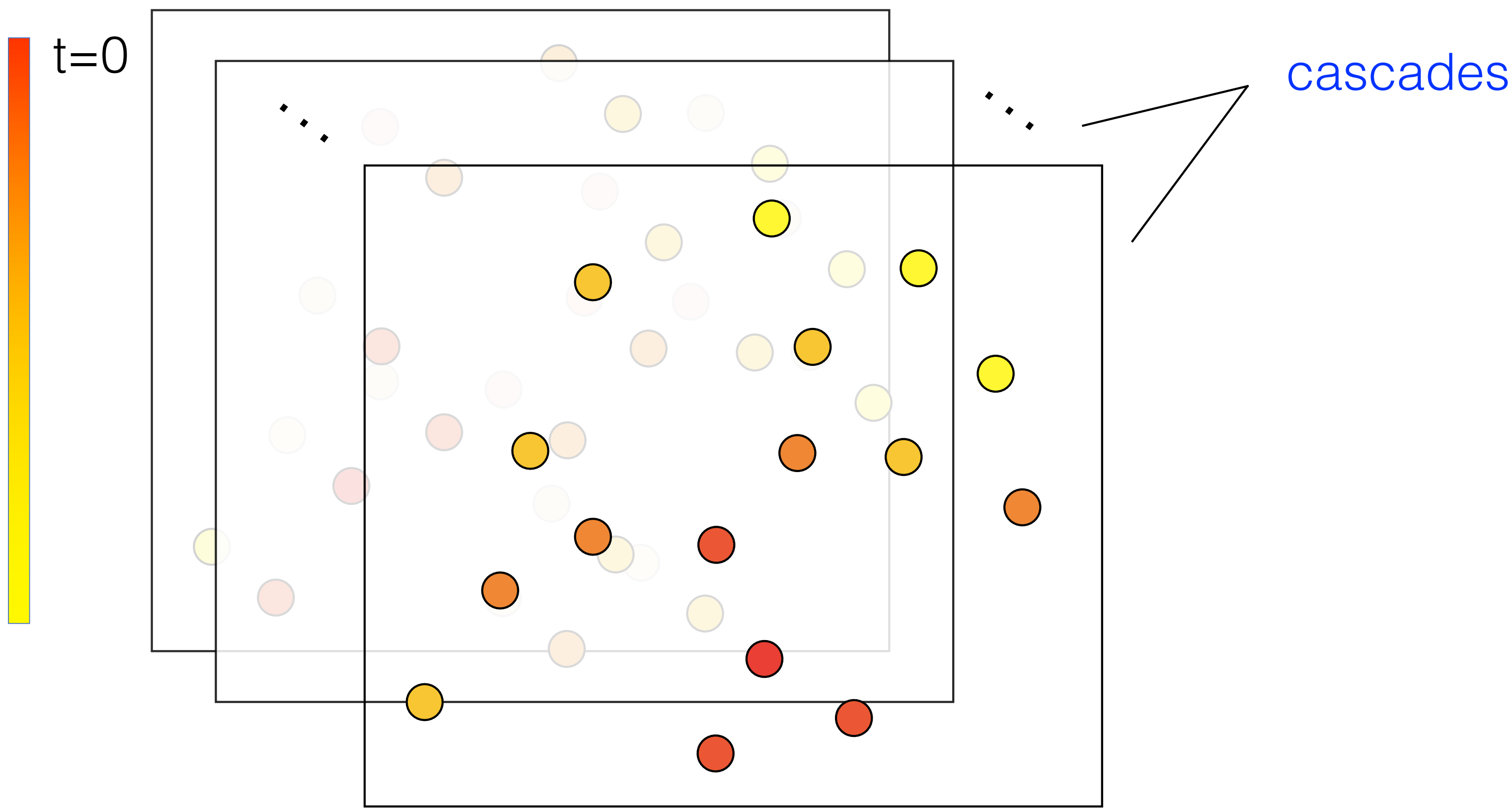
Inferring diffusion networks



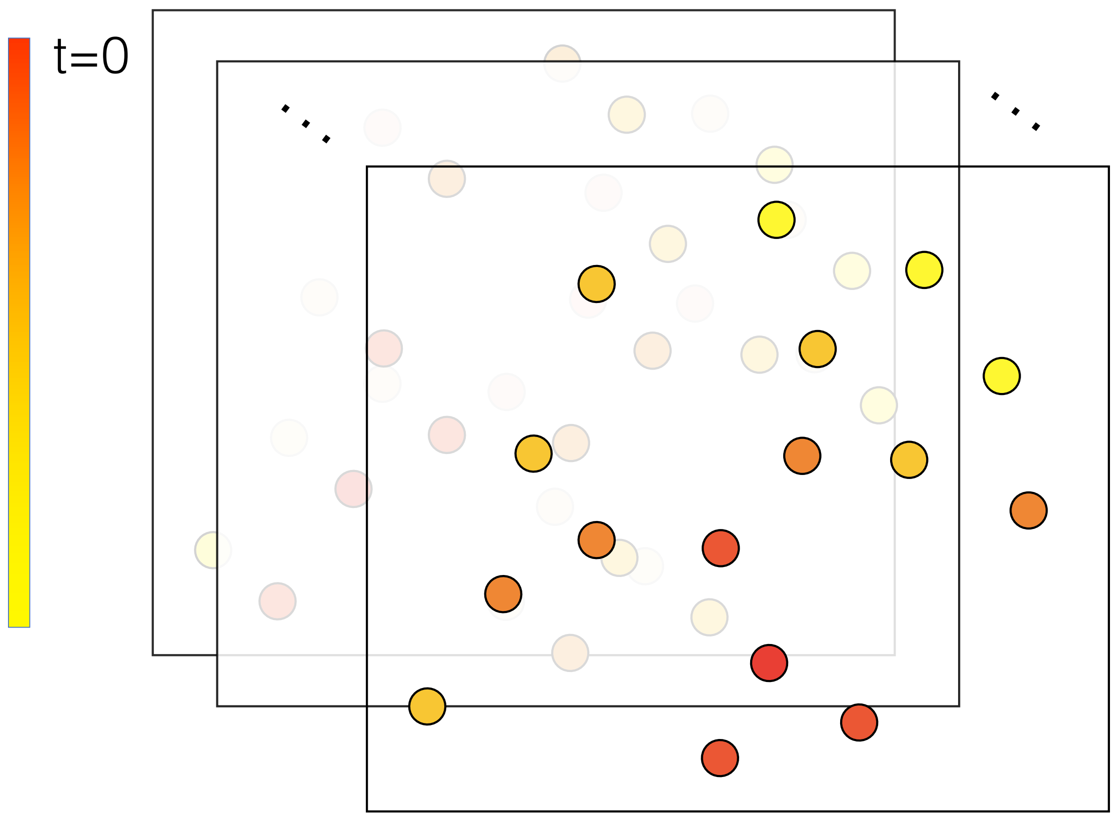
Inferring diffusion networks



Inferring diffusion networks



Inferring diffusion networks



Goal:

Estimate underlying graph topology

Models

- independent cascades model [Kempe, Kleinberg, Tardos ’03]
  - ❖ discrete-time
  - ❖ susceptible $\rightarrow$ active for one time-slot $\rightarrow$ inactive
  - ❖ node i infects j with probability p_{ij} if i is active

Algorithms

- estimate p_{ij} for all pairs (i,j):
    - ❖ log likelihood decouples, each term convex
- threshold to output graph
- sample complexity $O(d^2 \log n)$ for degree bound *d*

[Netrapalli, Sanghavi '12],

[Daneshmand, Gomez-Rodriguez, Song, Scholkopf '14]

Algorithms

- submodularity
- greedy algorithm; add one edge at a time to the graph estimate

[Gomez-Rodriguez, Leskovec, Krause ’12]

Hiding the Source

Part III

What you will learn in this hour

- Classical approach from the crypto community
  - Dining cryptographer networks

- Statistical approaches
  - **Static graph** is given

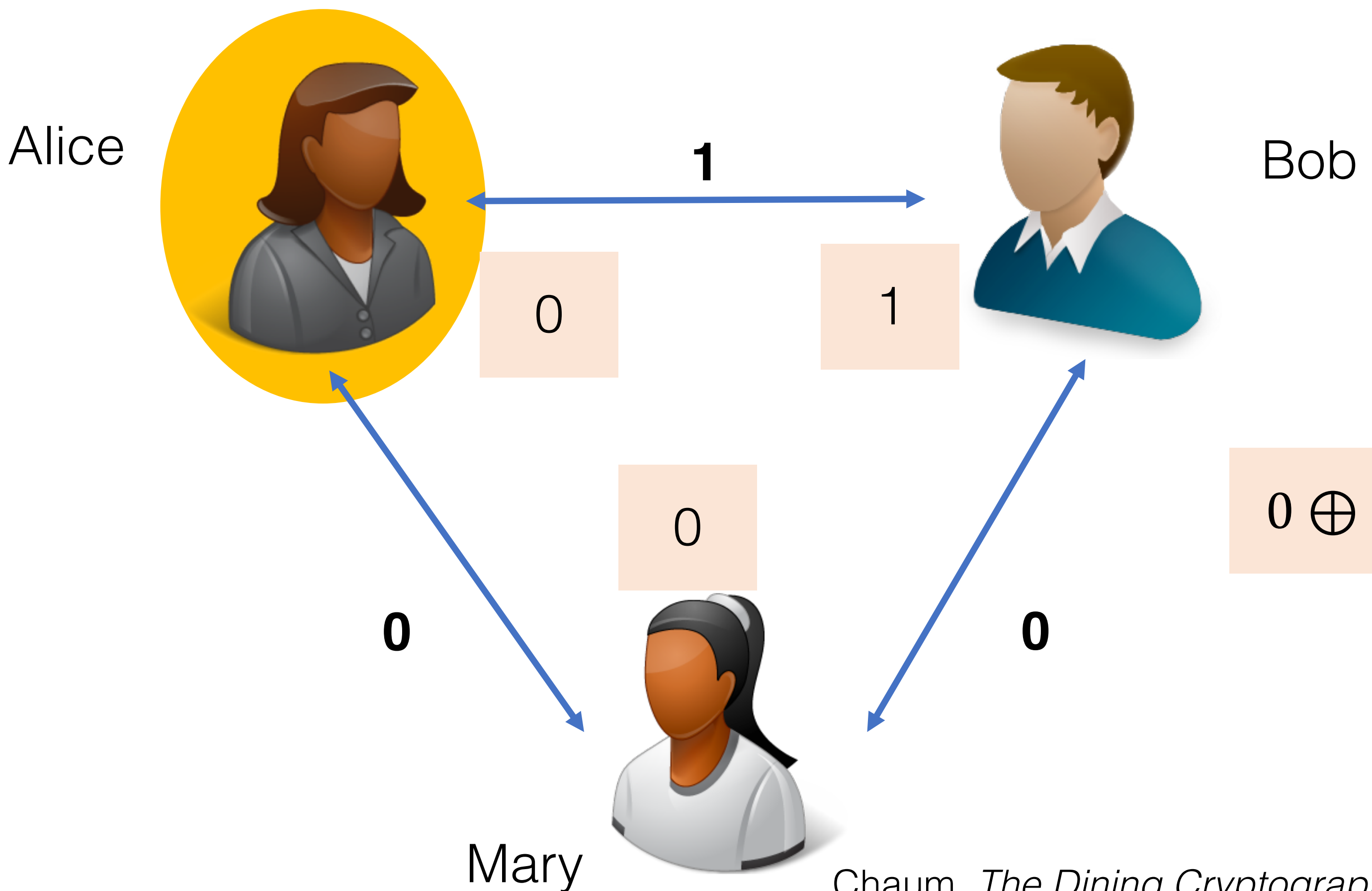
  - **Dynamic graph** can be chosen

- Open problems

General-Purpose Hiding

Dining Cryptographer Networks

Dining Cryptographer Networks



$$0 \oplus 0 \oplus 1 = 1$$

Chaum, *The Dining Cryptographers Problem*, 1988

What are some problems?

- High communication costs
- Cannot handle collisions
- Fragile to misbehaving nodes

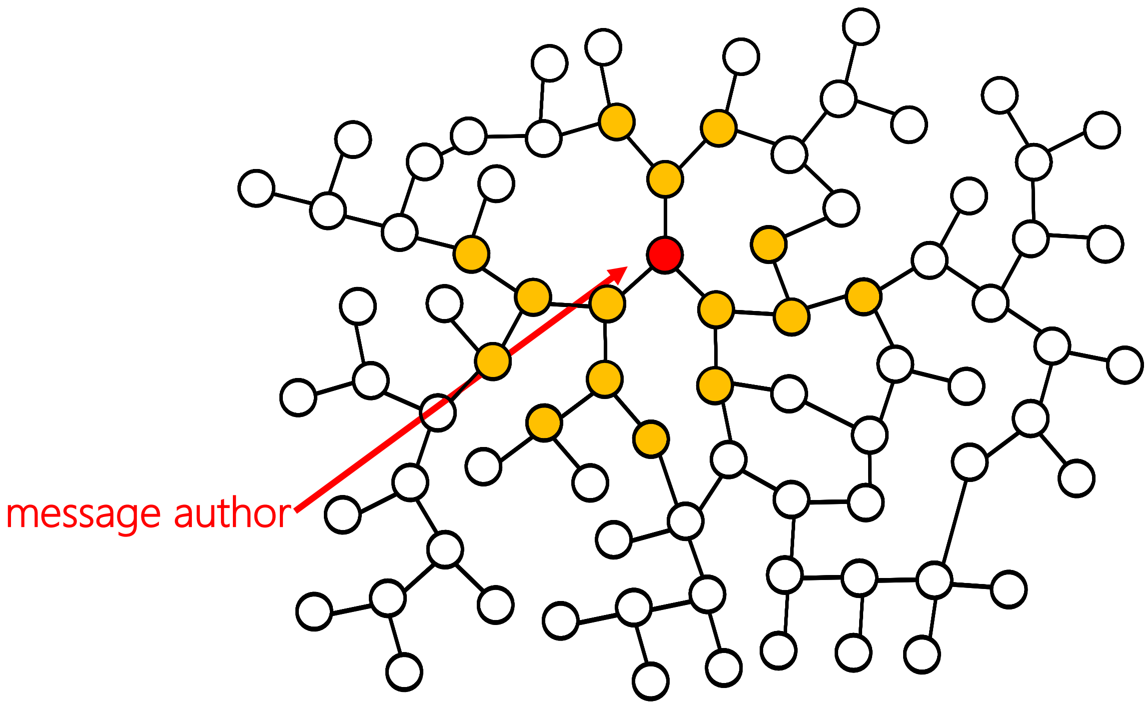
Golle and Juels, *Dining Cryptographers Revisited*, 2004
Sirer et al., *Eluding Carnivores: File Sharing with Strong Anonymity*, 2004
Franck, *New Directions for Dining Cryptographers*, 2008
Corrigan-Gibbs et al., *Dissent: Accountable Group Anonymity*, 2013
…

Worst-case solutions can be
too heavy to be practical.

Hiding on a Static Network

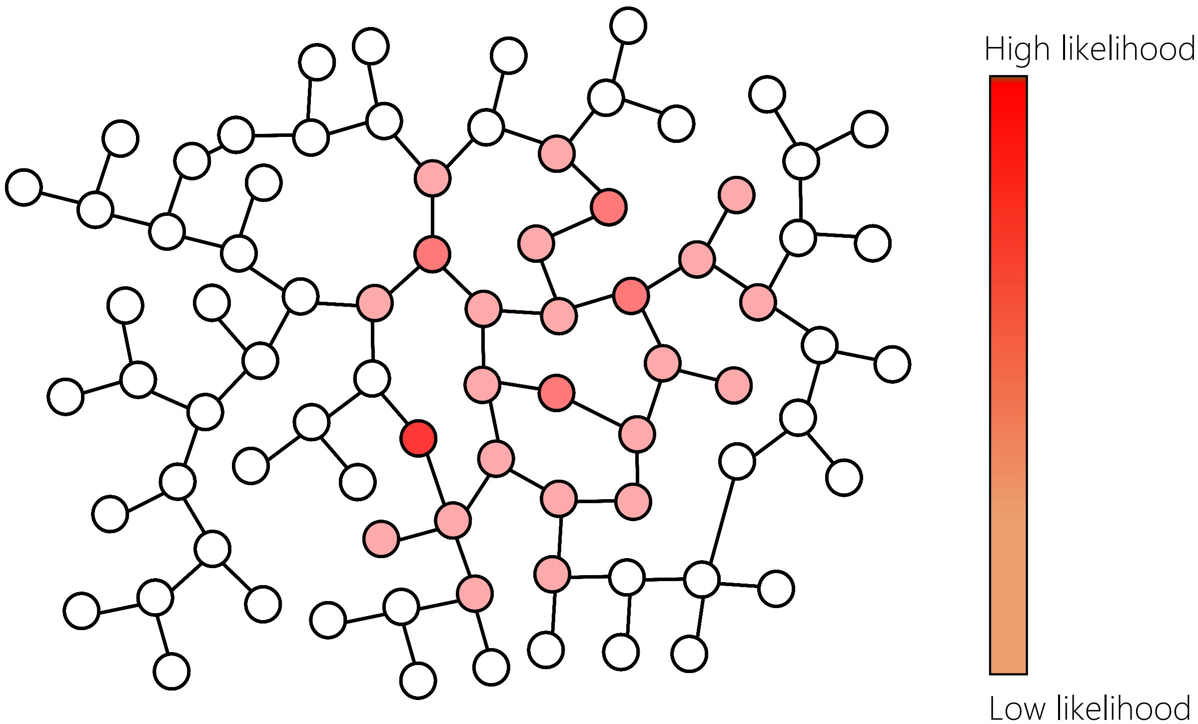
Applications in Social Networks

Information flow in social networks



Diffusion has statistical symmetry

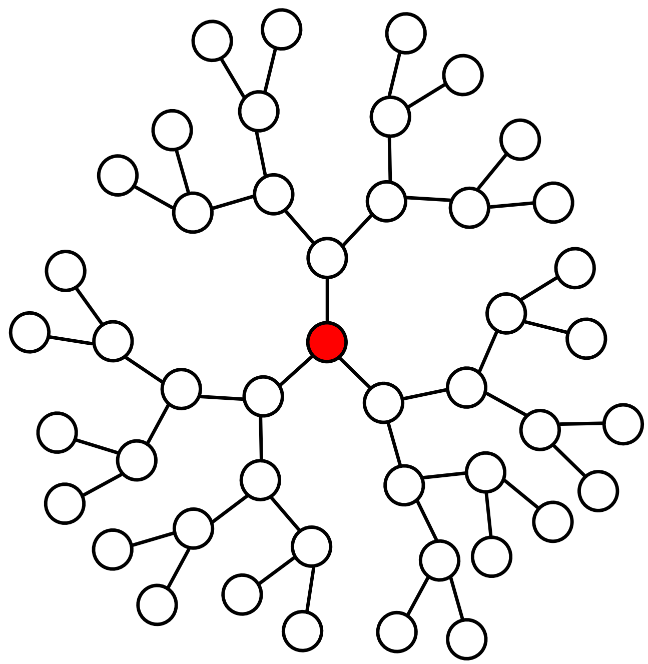
Breaking symmetry: Adaptive diffusion



Provides provable anonymity guarantees

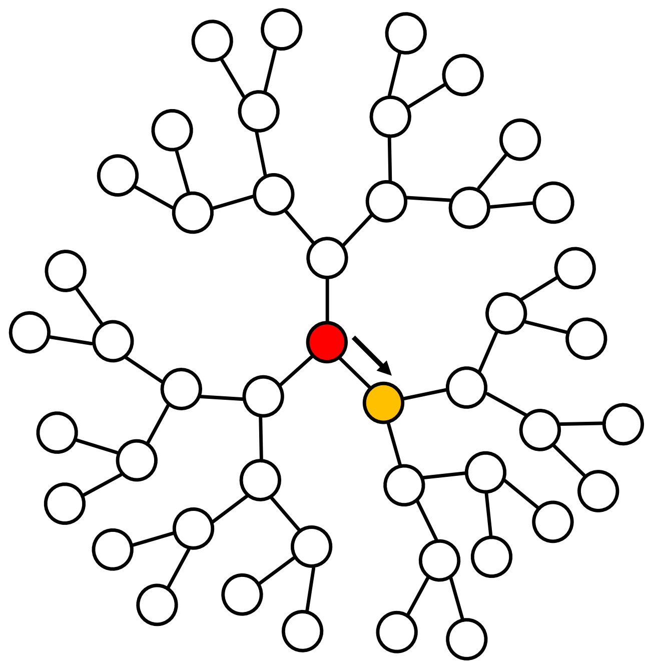
[*Spy vs. Spy: Rumor Source Obfuscation*, ACM Sigmetrics 2015]

d-regular trees: adaptive diffusion



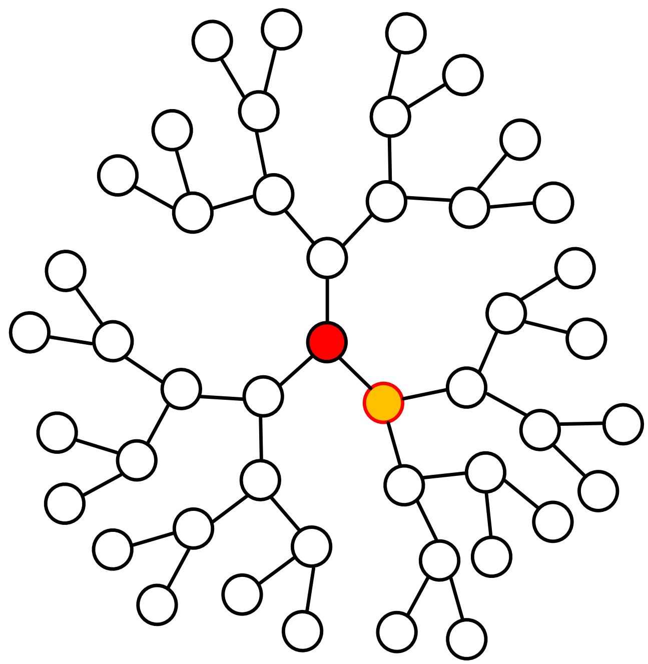
Initially, the author is also the "virtual source"

d-regular trees: adaptive diffusion



Break directional symmetry

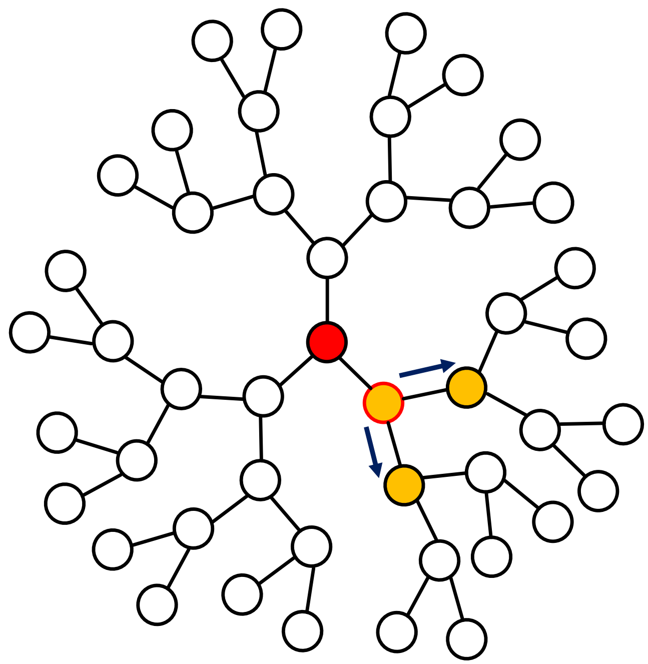
d-regular trees: adaptive diffusion



Break directional symmetry

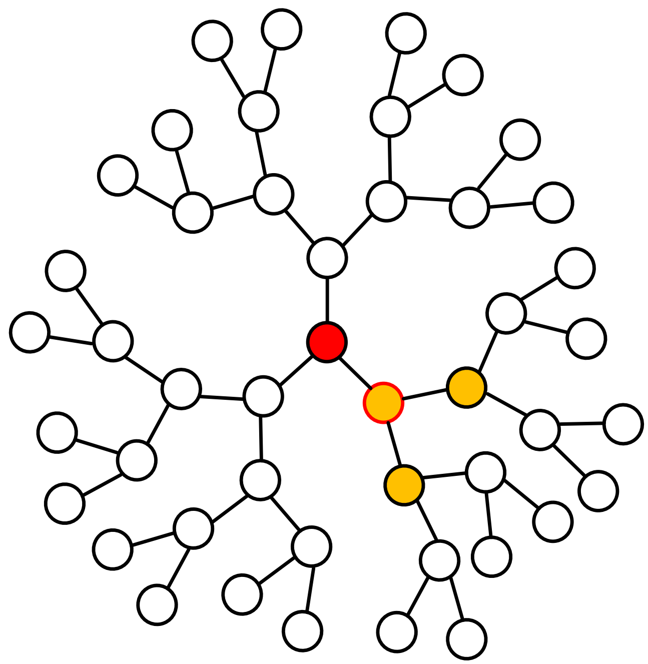
chosen neighbor = new virtual source

d-regular trees: adaptive diffusion



Break directional symmetry

d-regular trees: adaptive diffusion

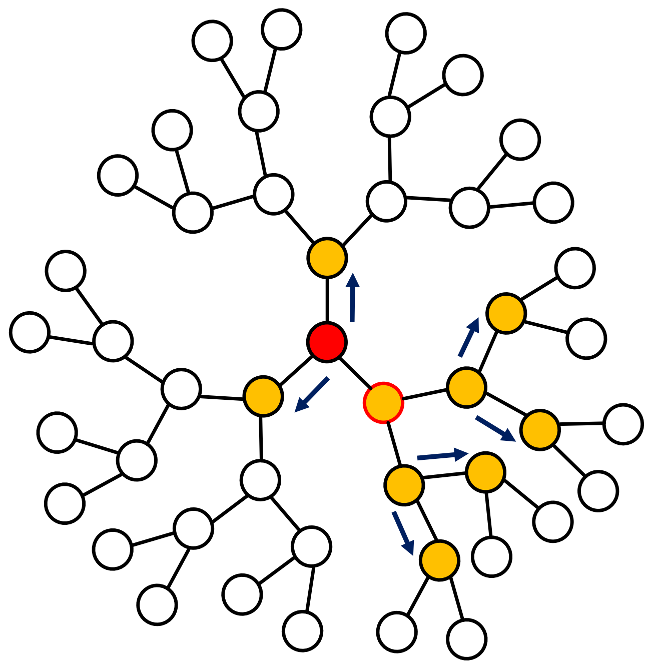


Break temporal symmetry

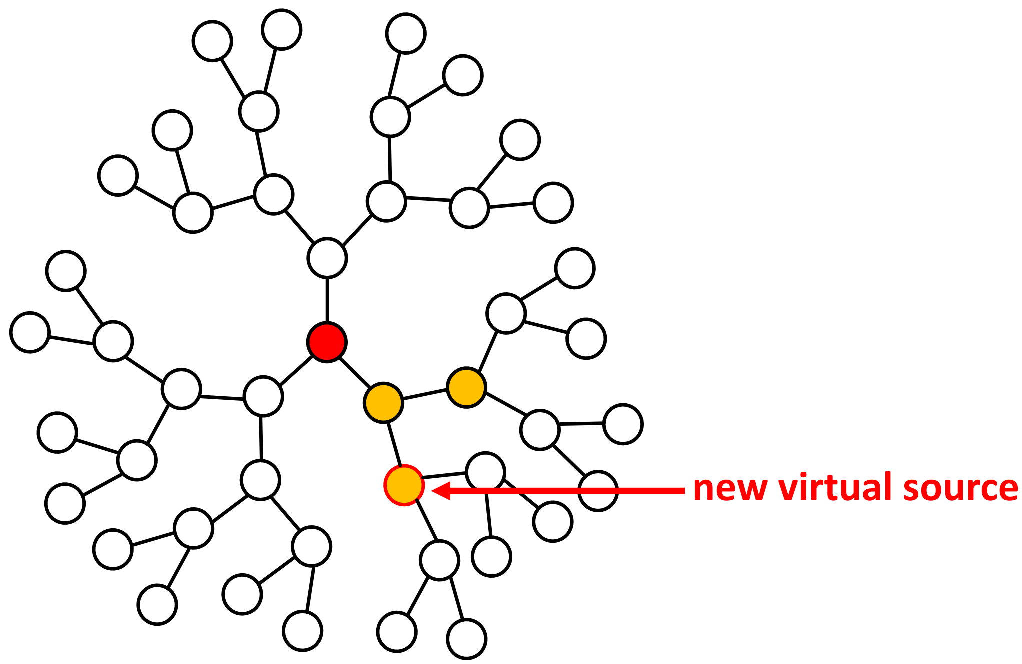
keep the virtual source token

pass the virtual source token

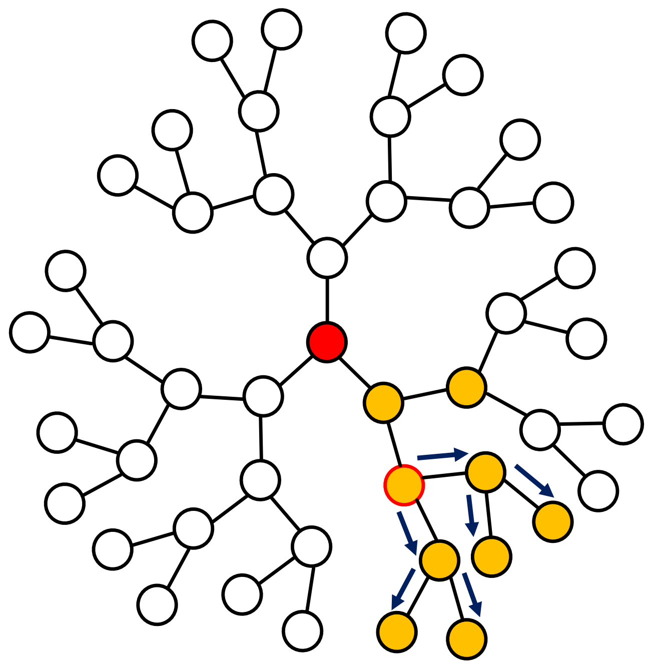
keep the virtual source token



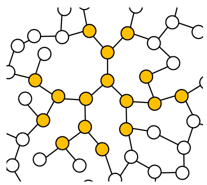
pass the virtual source token



pass the virtual source token

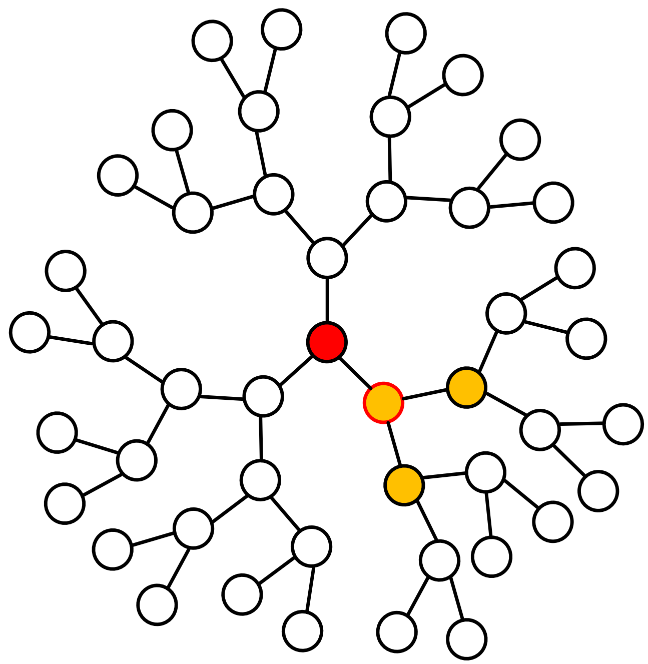


Results

	d-Regular trees	Irregular trees	Facebook graph
Snapshot 	[1]	[2]	[1]

[1] *Spy vs. Spy: Rumor Source Obfuscation*, Sigmetrics 2015
[2] *Rumor Source Obfuscation on Irregular Trees, Sigmetrics 2016*

When to keep the virtual source token?

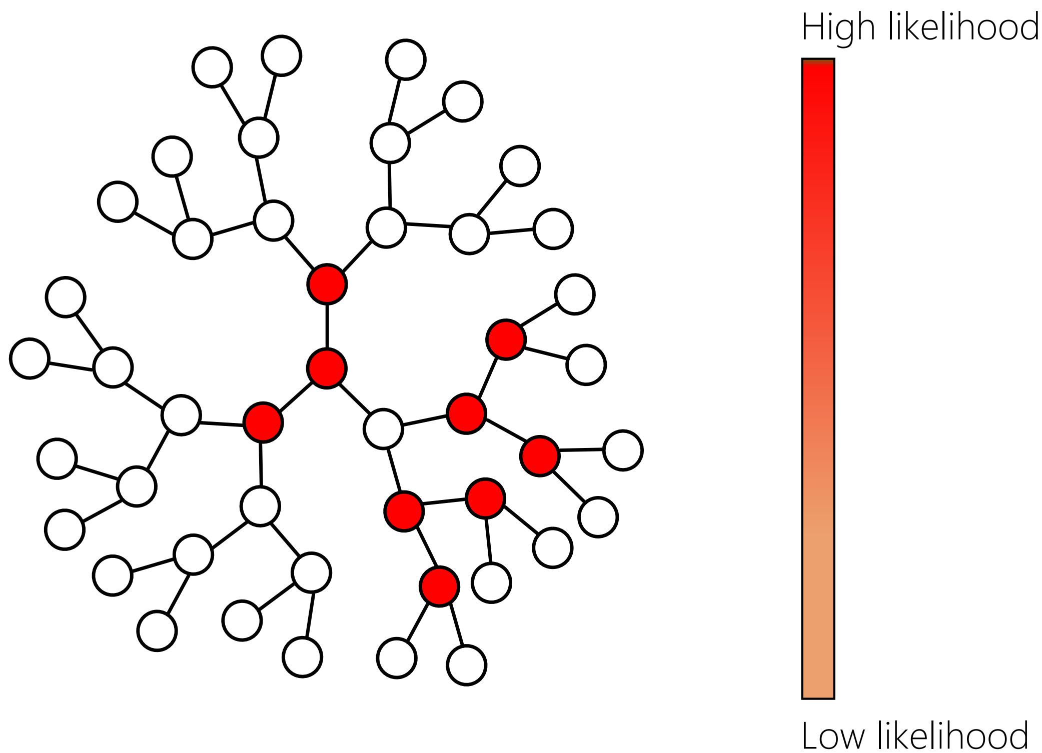


node degree

distance from source

Virtual source token is kept with probability $\boldsymbol{\alpha = (d - 1)^{-h}}$

Maximum likelihood detection

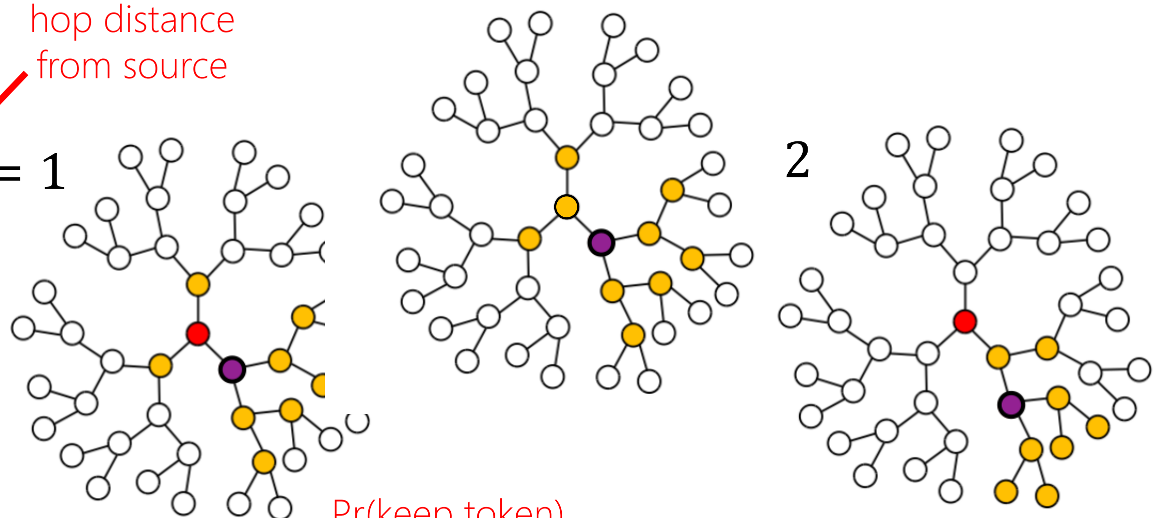


THEOREM: Probability of detection $= \frac{1}{N-1}$

hop distance from source

$h = 1$

2



Pr(keep token)

Likelihood $= \frac{1}{d} \cdot \alpha$

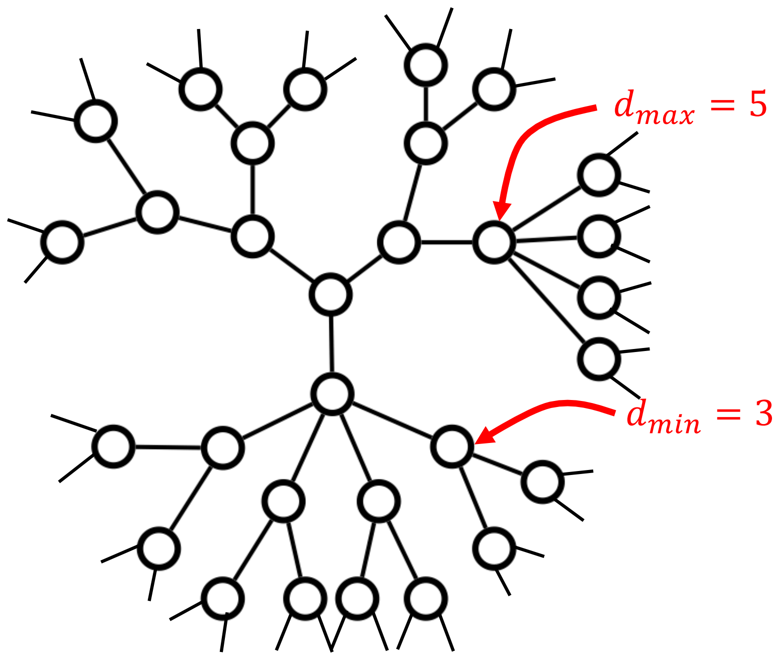
Tree degree

Likelihood $= \frac{1}{d} \cdot \frac{1-\alpha}{d-1}$

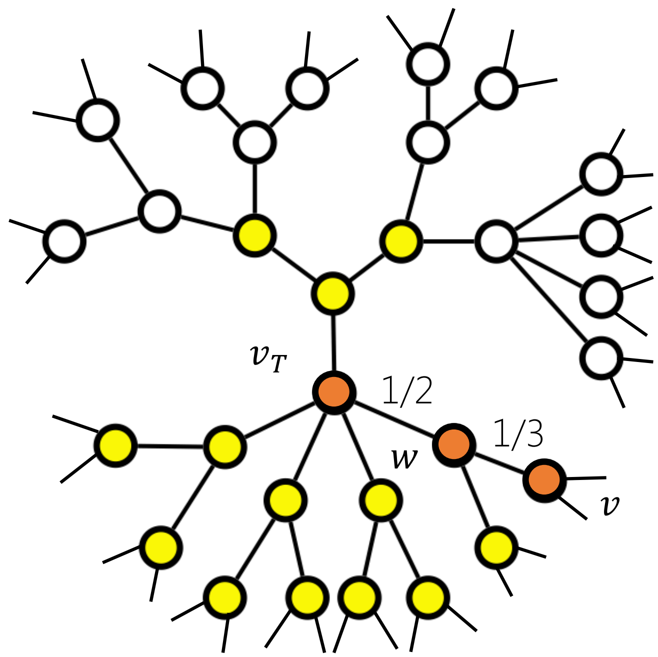
Want these to be equal: $\alpha = \frac{1}{d}$

Irregular trees

$$d_v = \begin{cases} 3 & w.p. \quad 0.7 \\ 5 & w.p. \quad 0.3 \end{cases}$$



How do we analyze this?



$$d_v = \begin{cases} d_{min} & w.p. \quad p_{min} \\ d_{max} & w.p. \quad p_{max} \end{cases}$$

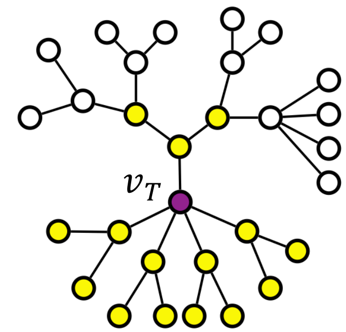
$$\hat{v}_{ML} = \arg \max_{v \in \text{leaves}} \frac{1}{d_v} \prod_{w \in P(v, v_T)} \frac{1}{d_w - 1}$$

Path from v to virtual source

Degree of node w

$$P(\text{detection} \mid \text{snapshot}) = \frac{1}{\boxed{\min_{v \in \text{leaves}} d_v \prod_{w \in P(v, v_T)} (d_w - 1)}}$$

Main result (special case)



$$\boxed{\Lambda_{G_T} \triangleq \min_{v \in \text{leaves}} d_v \prod_{w \in P(v, v_T)} (d_w - 1)}$$

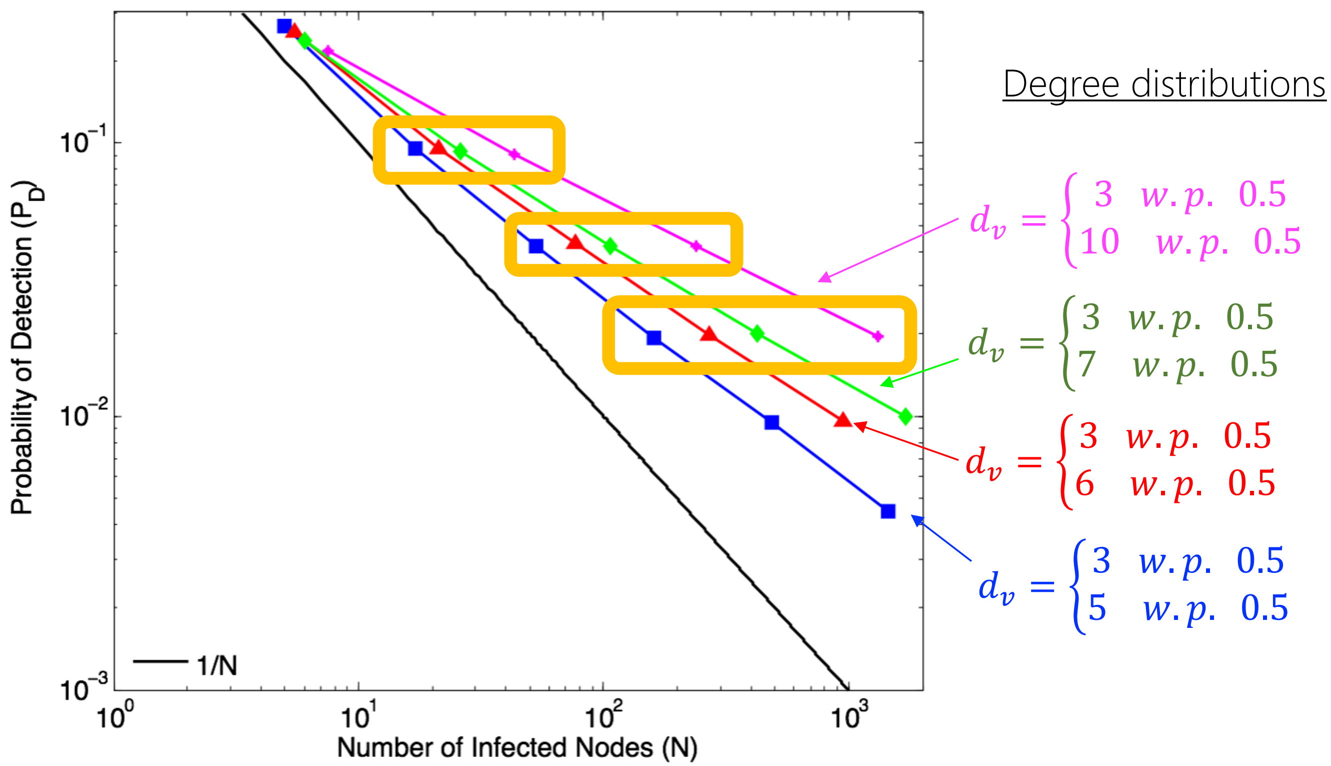
Probability of min degree

Min degree

$$\text{If} \;\; p_{min}(d_{min} - 1) > 1$$

$$P\left(\left|\frac{\log(\Lambda_{G_T})}{T} - \log(d_{min} - 1)\right| > \delta\right) \leq e^{-C_1 T}$$

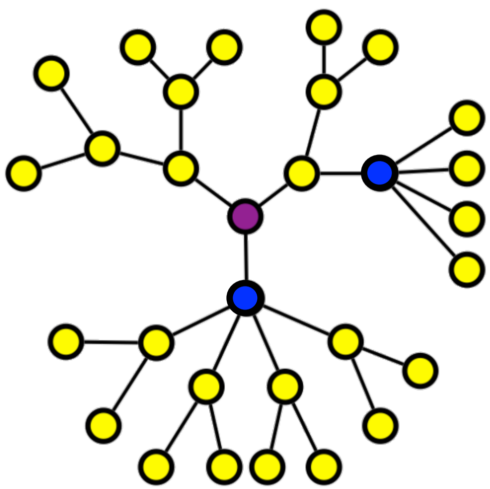
Theorem: Probability of detection $\approx \dfrac{1}{(d_{min}-1)^T}$



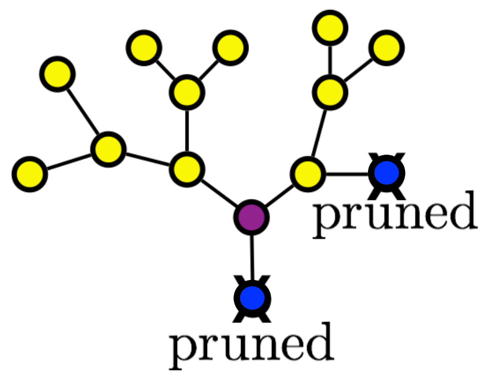
Theorem: Probability of detection $\approx \dfrac{1}{(d_{min}-1)^T}$

Proof sketch for $\min_{v \in \text{leaves}} d_v \prod_{w \in P(v, v_T)} (d_w - 1) \approx (d_{min} - 1)^T$

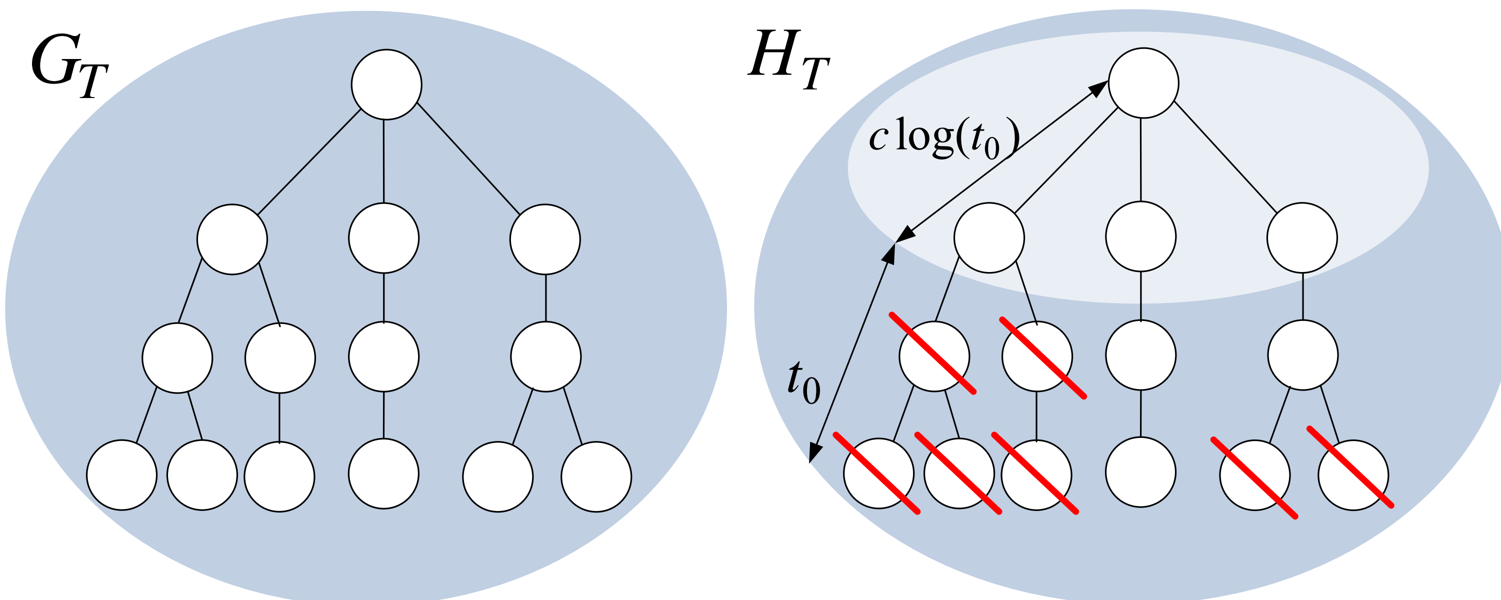
$$d_v = \begin{cases} 3 & w.p. \quad 0.7 \\ 5 & w.p. \quad 0.3 \end{cases}$$



$$d_v = \begin{cases} 3 & w.p. \quad 0.7 \\ 1 & w.p. \quad 0.3 \end{cases}$$



If $\underset{0.7}{p_{min}}(\underset{3}{d_{min}} - 1) > 1$ then the pruned process survives.



If $p_{min}(d_{min}-1) > 1$:

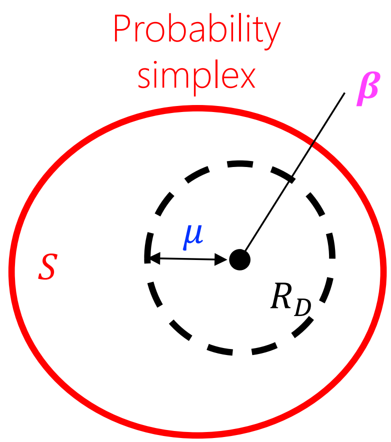
$$\min_{v\in\text{leaves}} d_v \prod_{w\in P(v,v_T)} d_w - 1 \;\approx\; (d_{min}-1)^T$$

Main result

$$\boxed{\Lambda_{G_T} \triangleq \min_{v \in \text{leaves}} d_v \prod_{w \in P(v, v_T)} (d_w - 1)}$$

$$d_v = \begin{cases} 3 & w.p. \quad 0.7 \\ 5 & w.p. \quad 0.3 \end{cases}$$

In general, $$P\left(\left|\frac{\log(\Lambda_{G_T})}{T} - r^*\right| > \delta\right) \le e^{-C_1 T}$$



$$\beta_i = \frac{p_i(d_i - 1)}{\sum_j p_j(d_j - 1)}$$

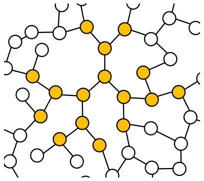
$$\mu = \log \sum_i p_i(d_i - 1)$$

KL-Divergence

$$R_D = \{\, \boldsymbol{r} \in S \mid \mu \ge D_{KL}(\boldsymbol{r} \parallel \boldsymbol{\beta})\}$$

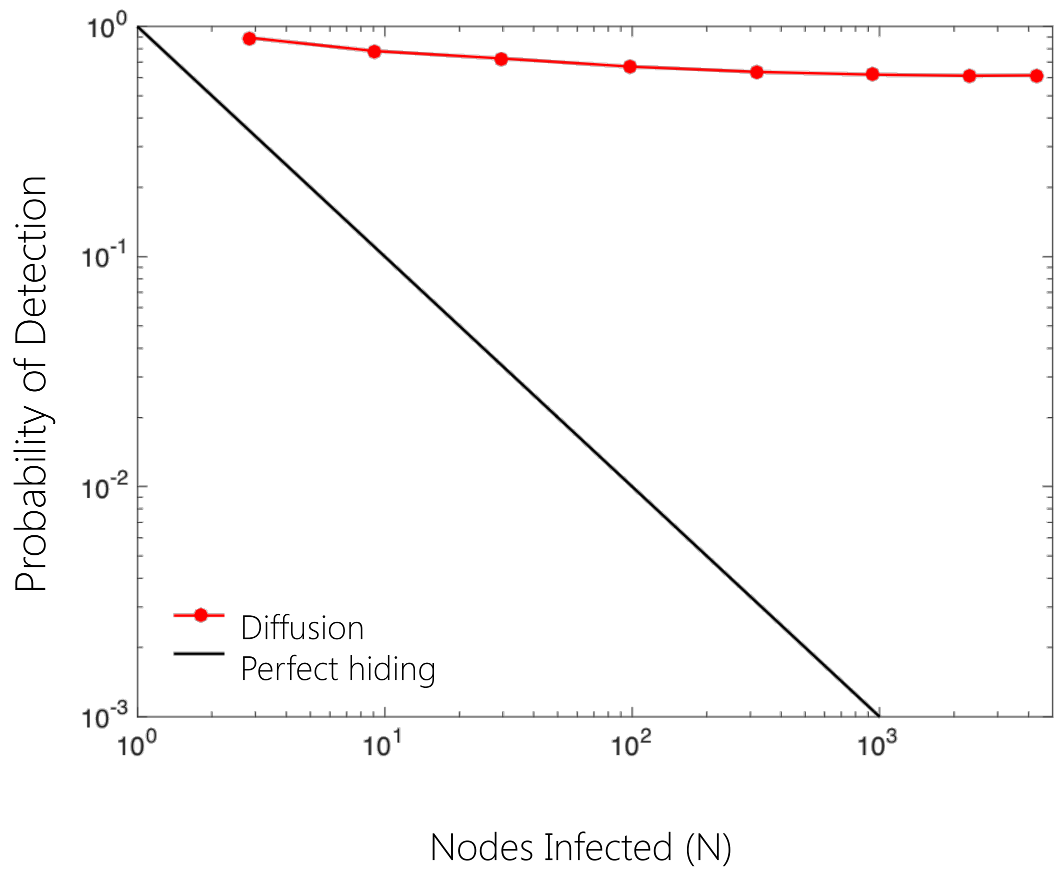
$$r^* = \min_{\mathbf{r} \in R_D} \langle \boldsymbol{r}, \log(\boldsymbol{d} - 1)\rangle$$

Results

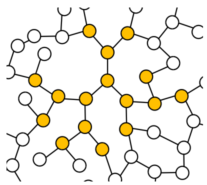
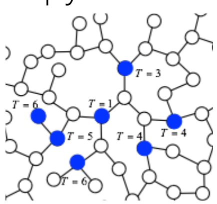
	d-Regular trees	Irregular trees	Facebook graph
Snapshot 	Optimal [1]	Near-optimal [2]	[1]

[1] *Spy vs. Spy: Rumor Source Obfuscation*, Sigmetrics 2015
[2] *Rumor Source Obfuscation on Irregular Trees, Sigmetrics 2016*

Facebook graph



Results

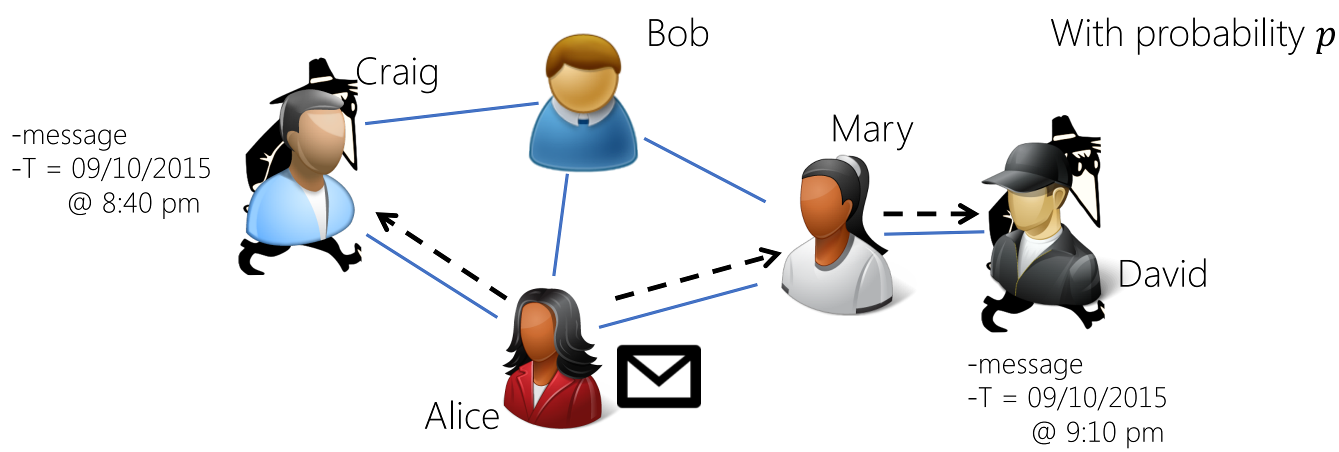
	d-Regular trees	Irregular trees	Facebook graph
Snapshot 	Optimal [1]	Near-Optimal [2]	Near lower bound [1]
Spy-based 	[3]	[3]	[3]

[1] *Spy vs. Spy: Rumor Source Obfuscation*, Sigmetrics 2015

[2] *Rumor Source Obfuscation on Irregular Trees, Sigmetrics 2016*

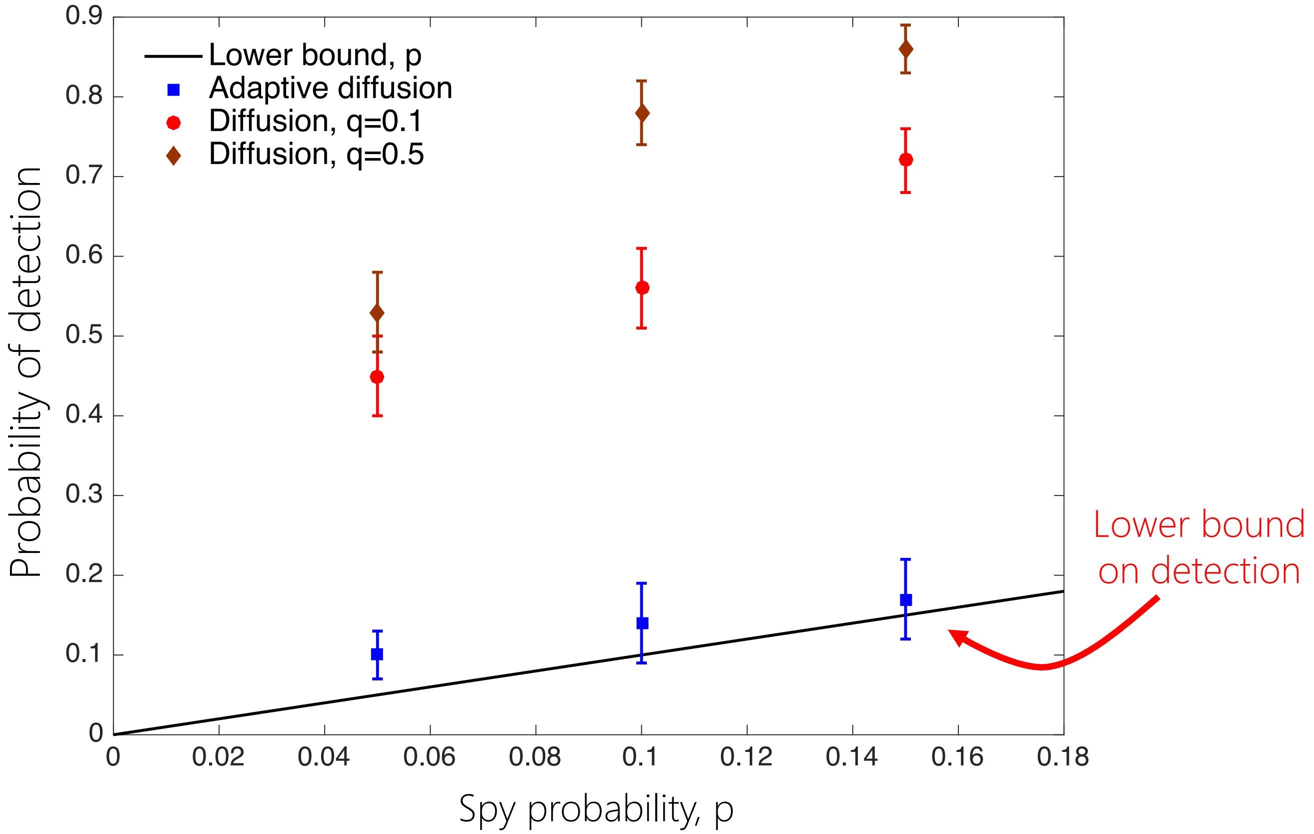
[3] *Metadata-Conscious Anonymous Messaging, ICML 2016*

Spy-based adversary

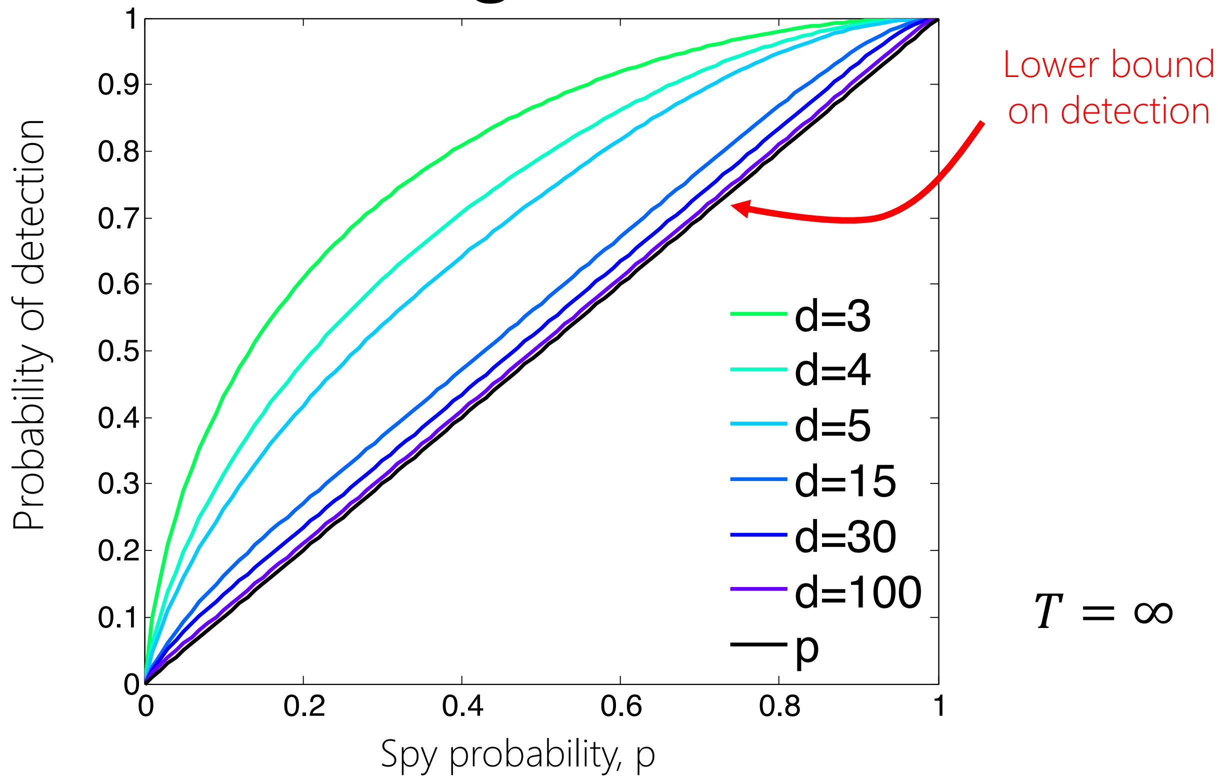


Adversary sees metadata at spy nodes

Facebook Graph



Result on d-regular trees



THEOREM: Probability of detection $= p + o(p)$

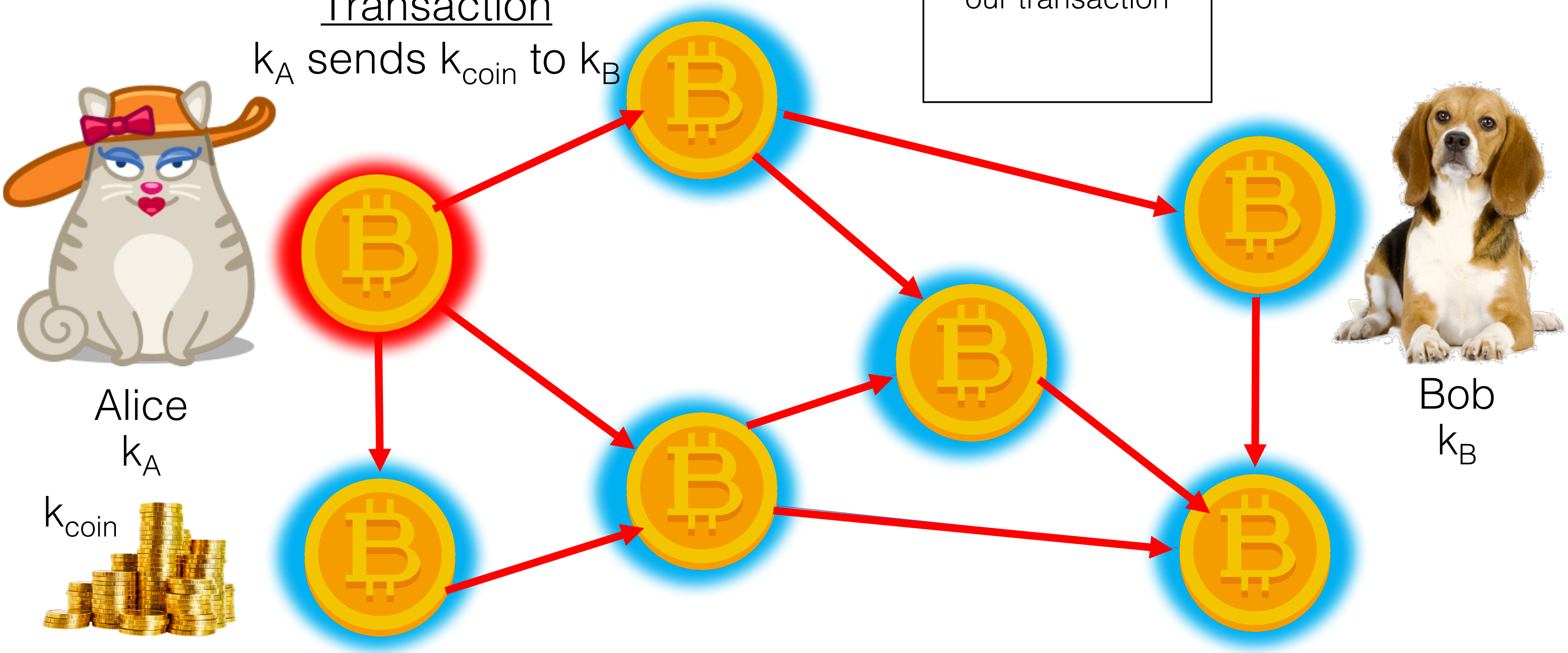
Hiding on a Dynamic Network

Applications in Cryptocurrencies

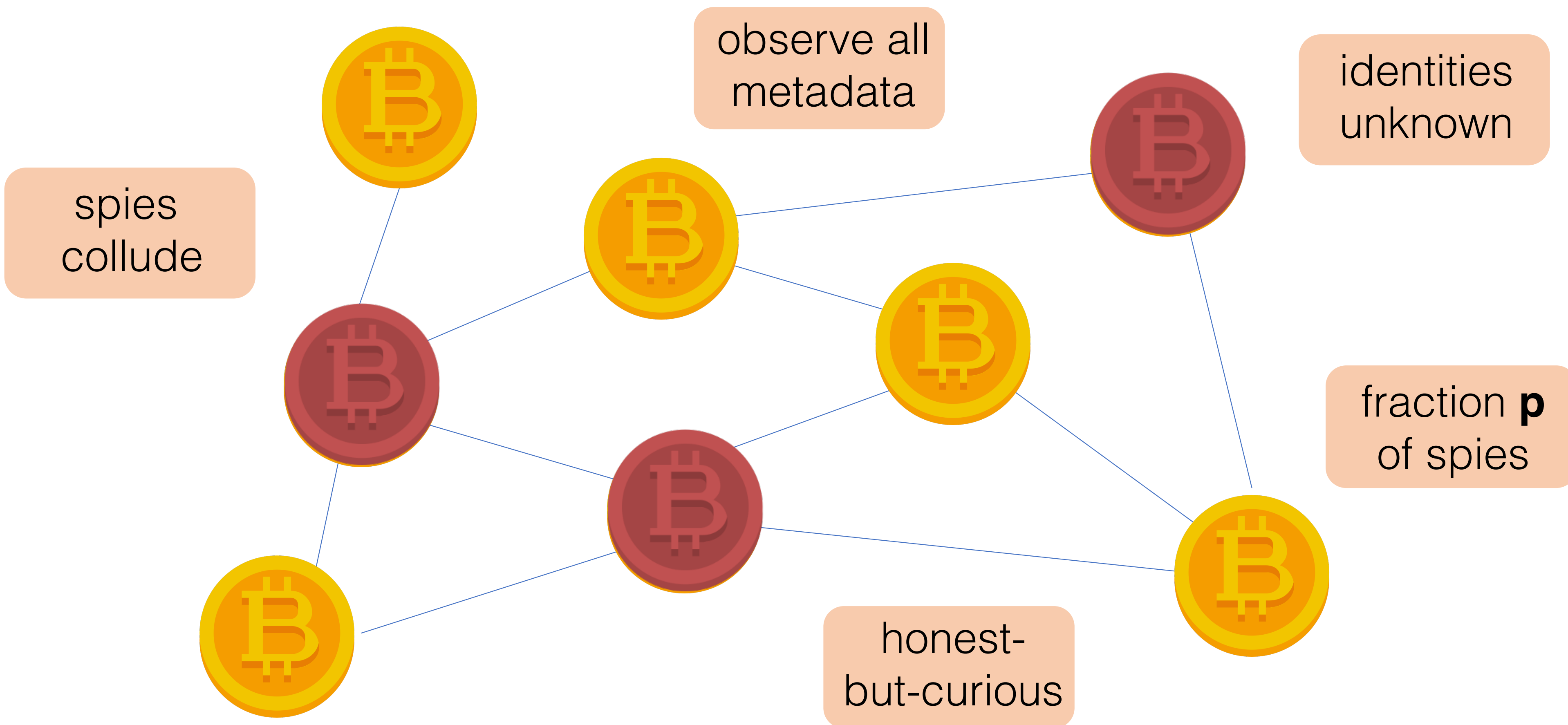
Bitcoin Reminder

Blockchain
sd93fjj2
pckrn29
...
our transaction

Transaction
k_A sends k_{coin} to k_B



Botnet (spy-based) adversarial model



Metric for Anonymity

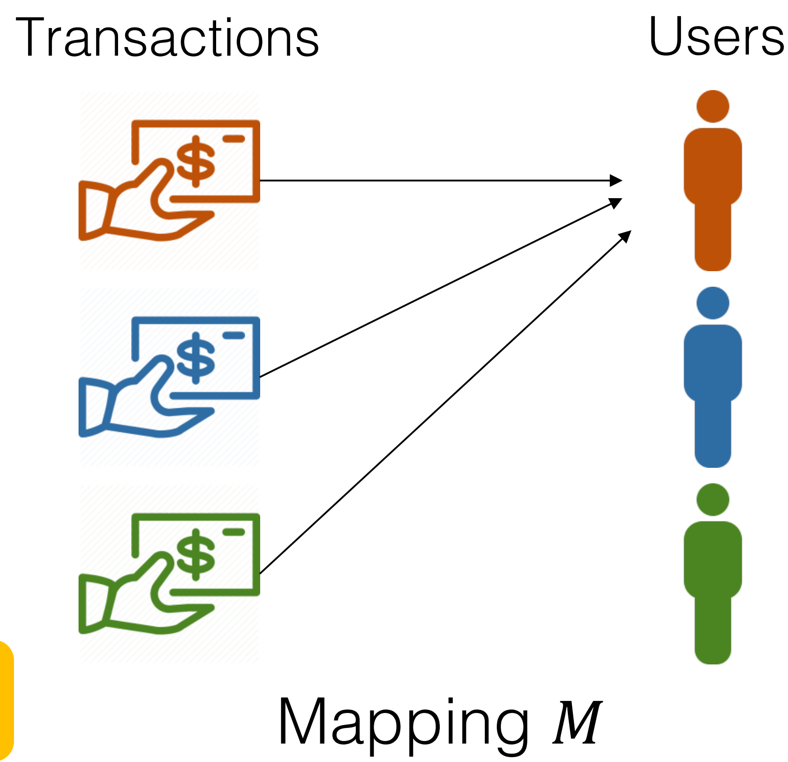
Recall

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

Mapping

Number honest users

User



Precision

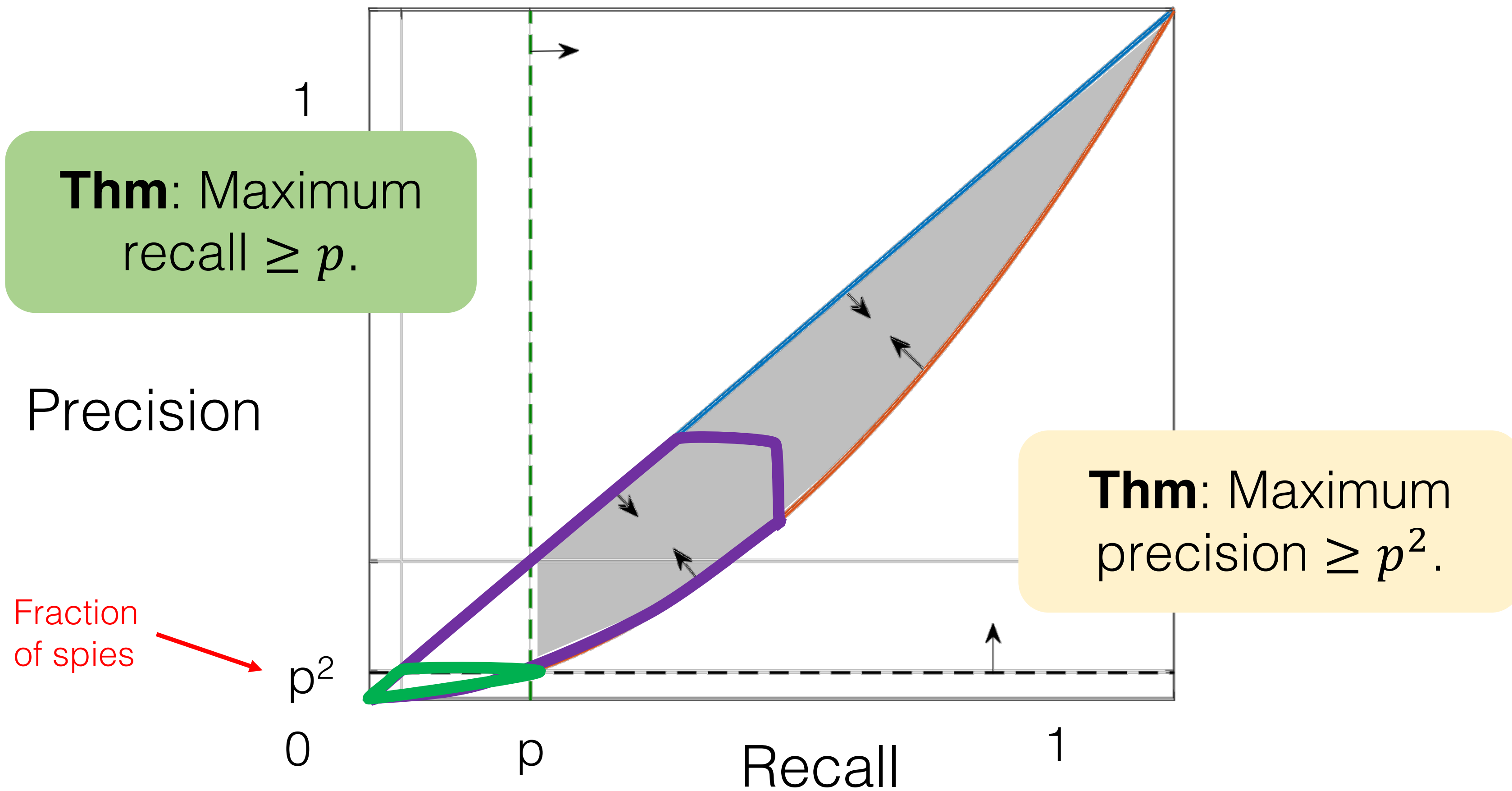
$$\frac{1}{n}\sum_{v} \frac{1\{M(v'\text{s tx}) = v\}}{\#\text{ tx mapped to v}}$$

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

Goal:

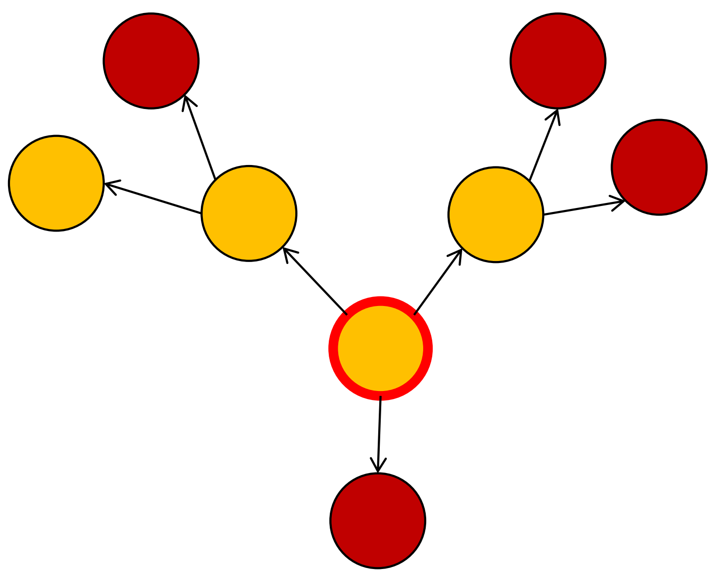
Design a distributed flooding protocol that minimizes the maximum precision and recall achievable by a computationally-unbounded adversary.

Fundamental Limits

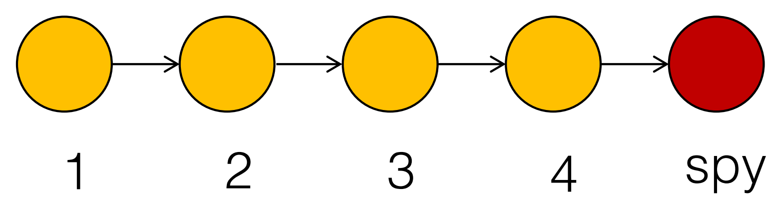


What are we looking for?

Asymmetry

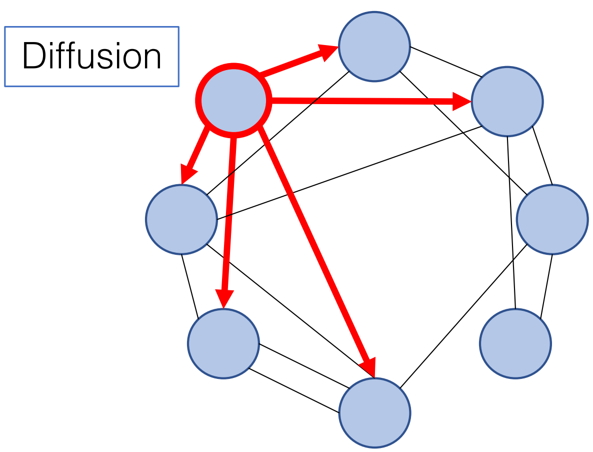


Mixing



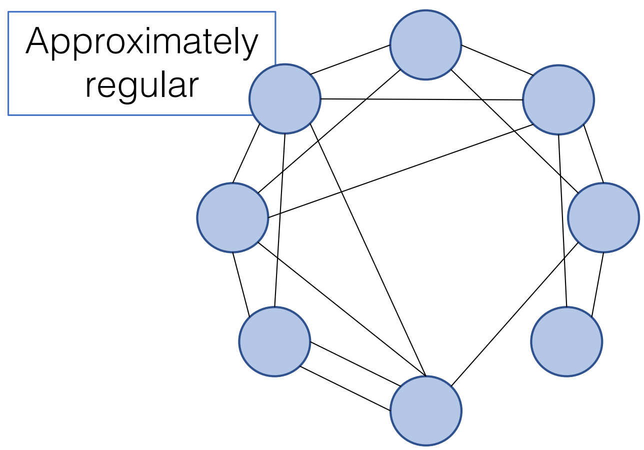
What can we control?

Spreading Protocol



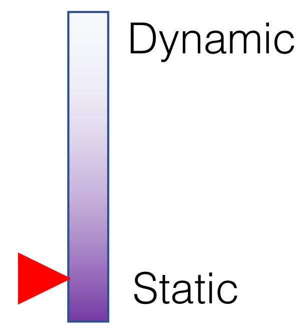
Given a graph, how do we spread content?

Topology



What is the underlying graph topology?

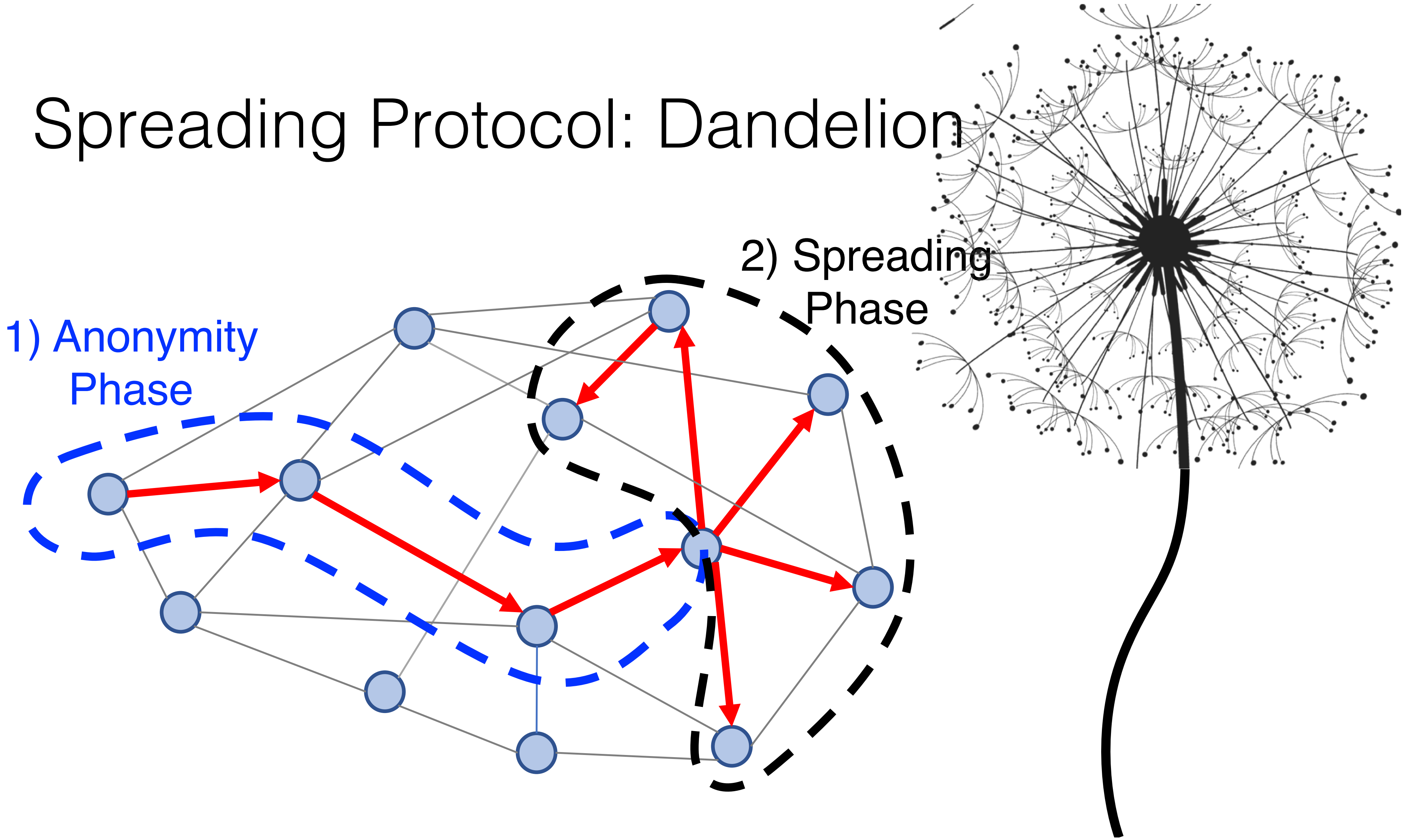
Dynamicity



How often does the graph change?

Dandelion: Redesigning the Bitcoin Network for Anonymity, Sigmetrics 2017

Spreading Protocol: Dandelion



Why Dandelion spreading?

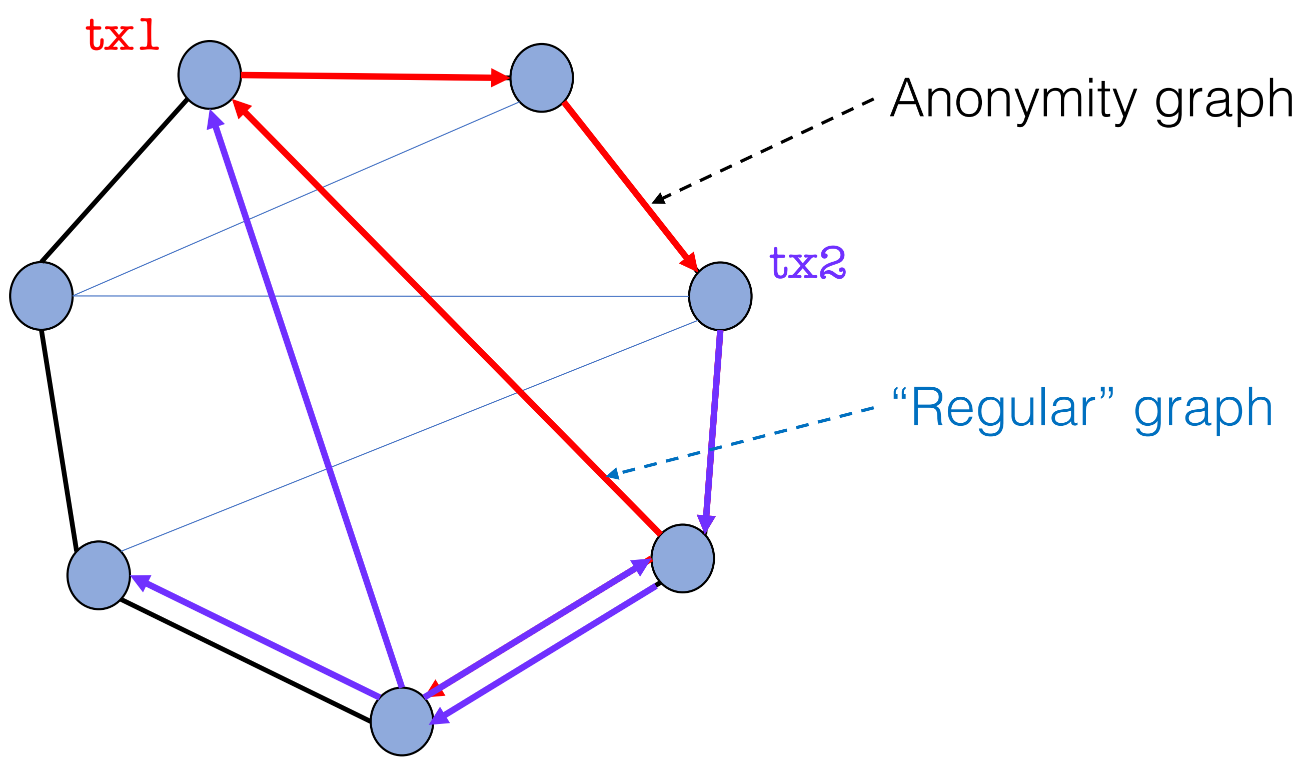
Theorem: Dandelion spreading has an optimally low maximum recall of $p + O\left(\frac{1}{n}\right)$.

lower bound = p

fraction of spies

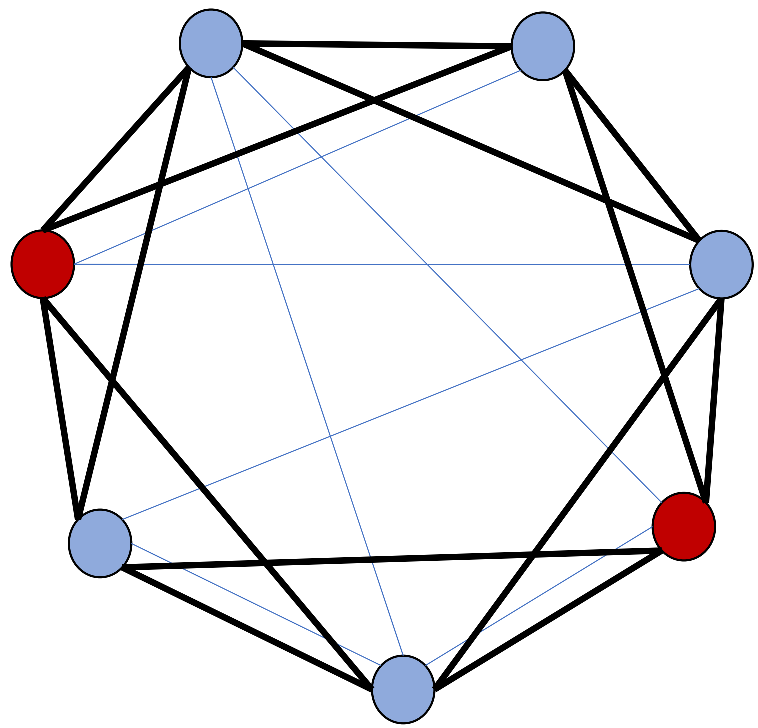
number of nodes

Graph Topology: Line



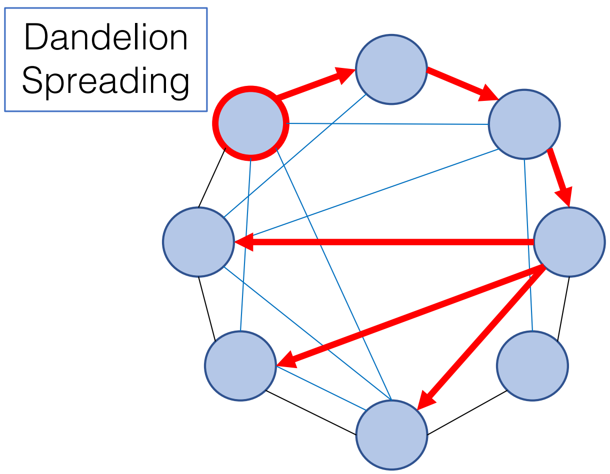
Dynamicity: High

Change the anonymity graph frequently.



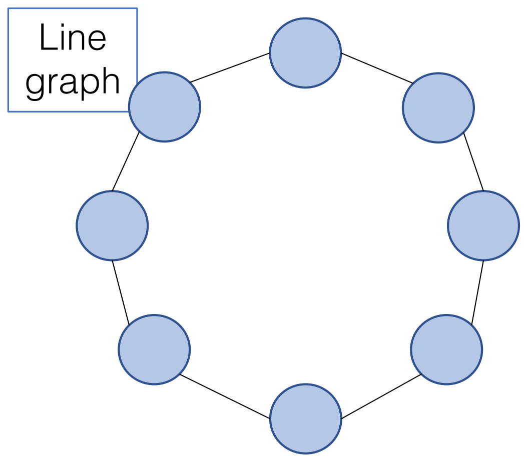
DANDELION Network Policy

Spreading Protocol



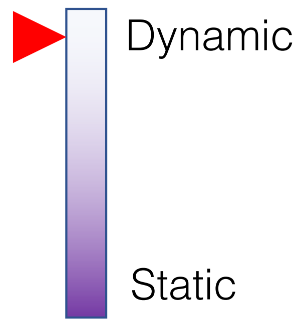
Given a graph, how do we spread content?

Topology



What is the anonymity graph topology?

Dynamicity



How often does the graph change?

lower bound = p^2

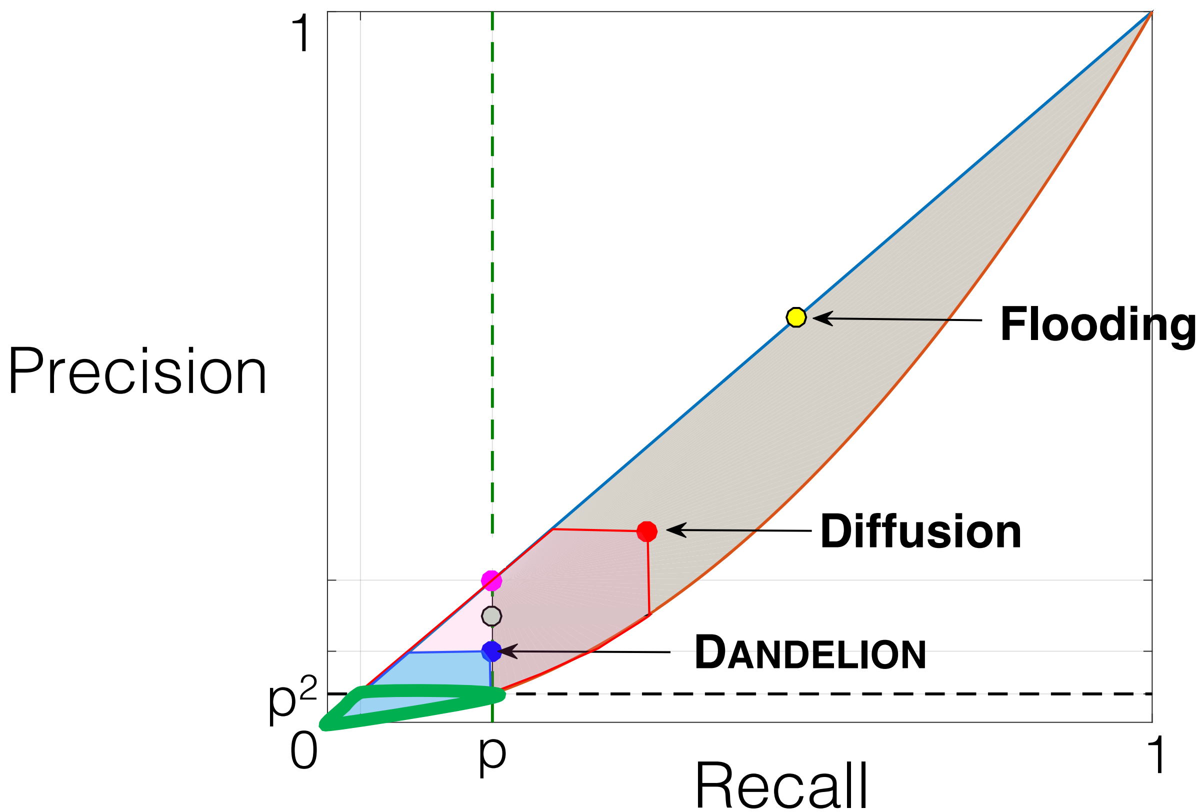
Theorem: DANDELION has a nearly-optimal maximum precision of $\frac{2p^2}{1-p}\log\left(\frac{2}{p}\right) + O\left(\frac{1}{n}\right)$.*

fraction of spies

number of nodes

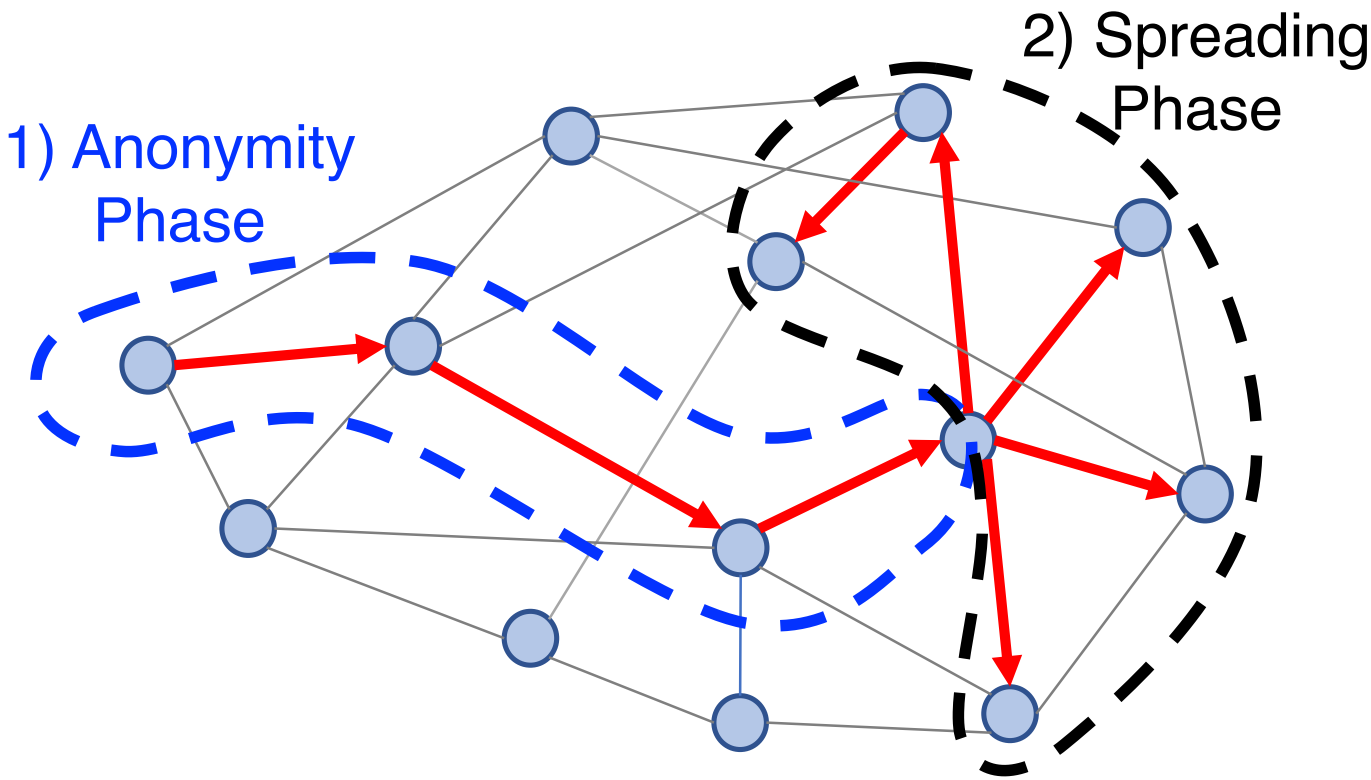
*For $p < \frac{1}{3}$

Performance: Achievable Region

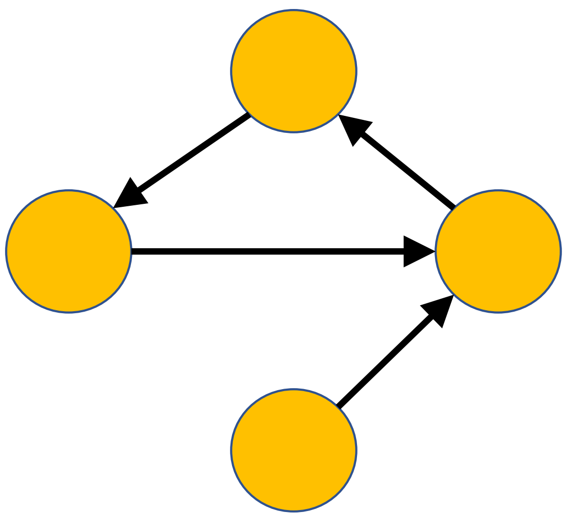


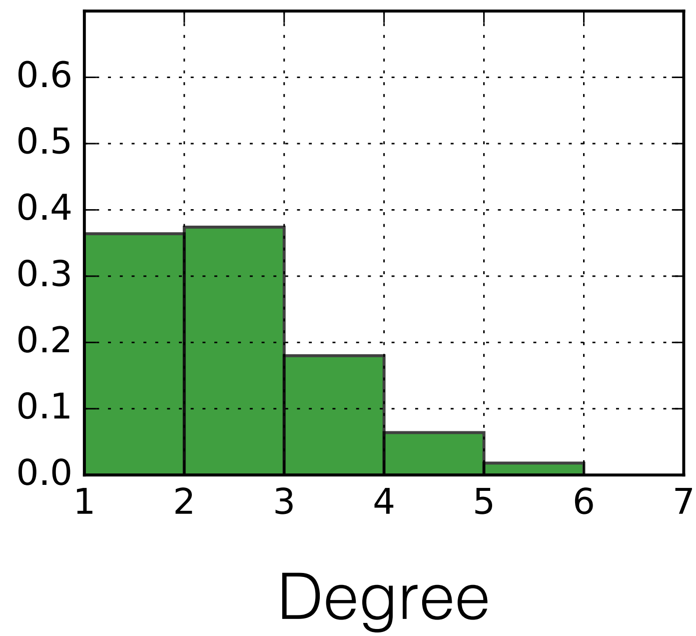
How practical is this?

Dandelion spreading

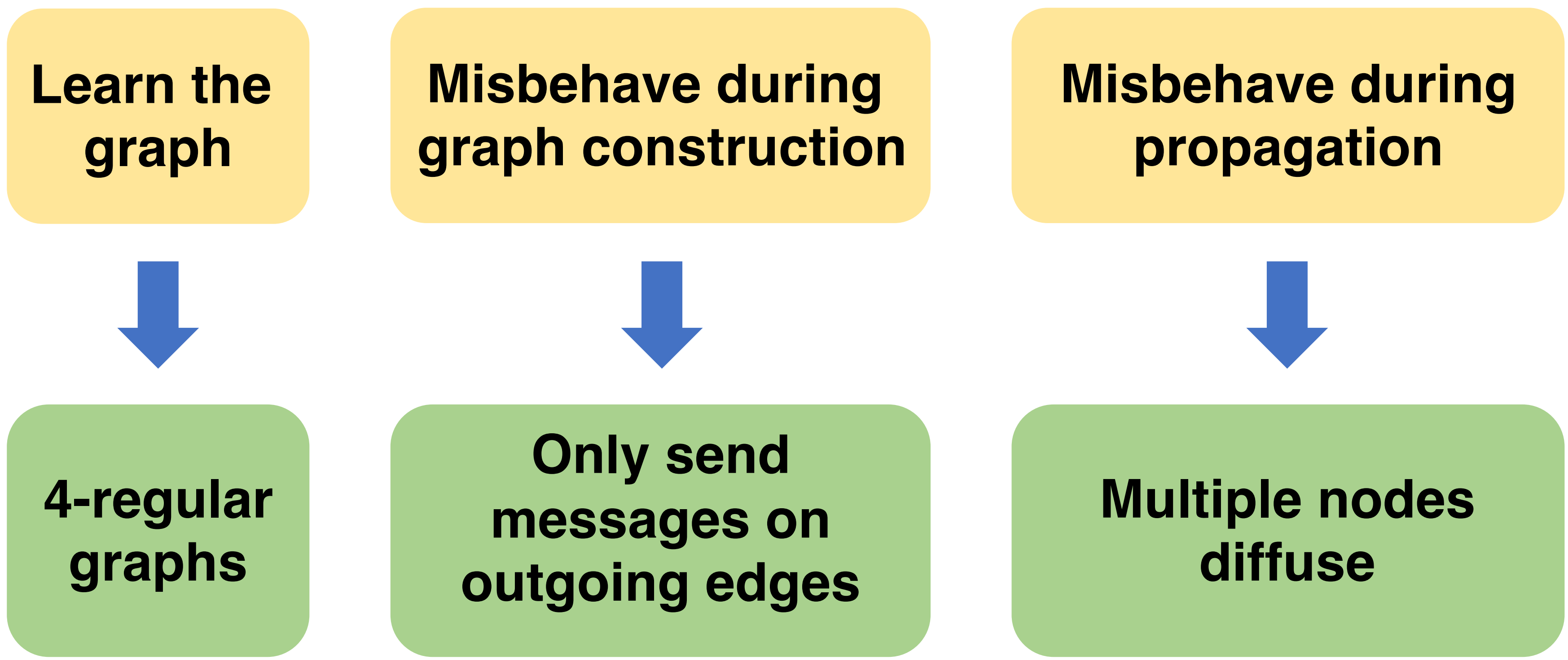


Anonymity graph construction

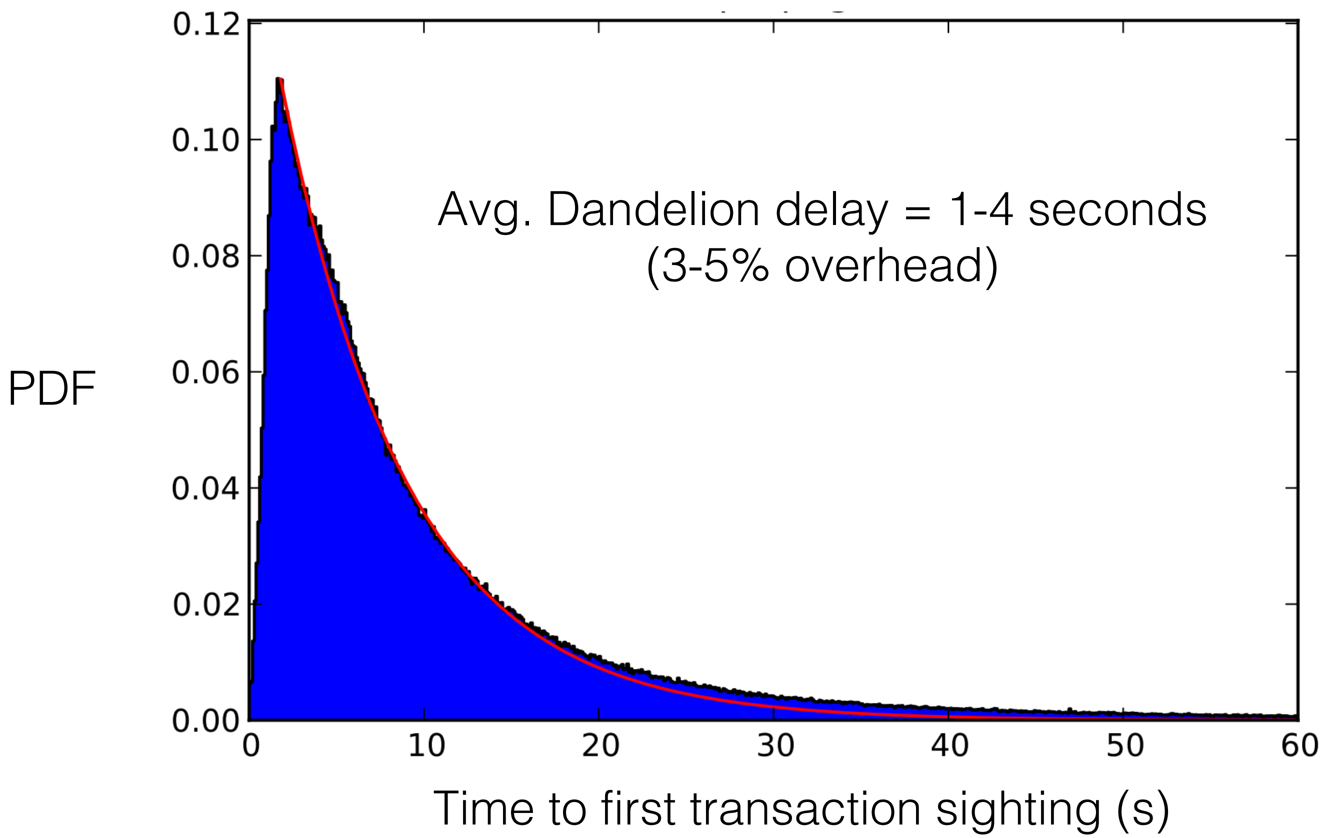




Dealing with stronger adversaries

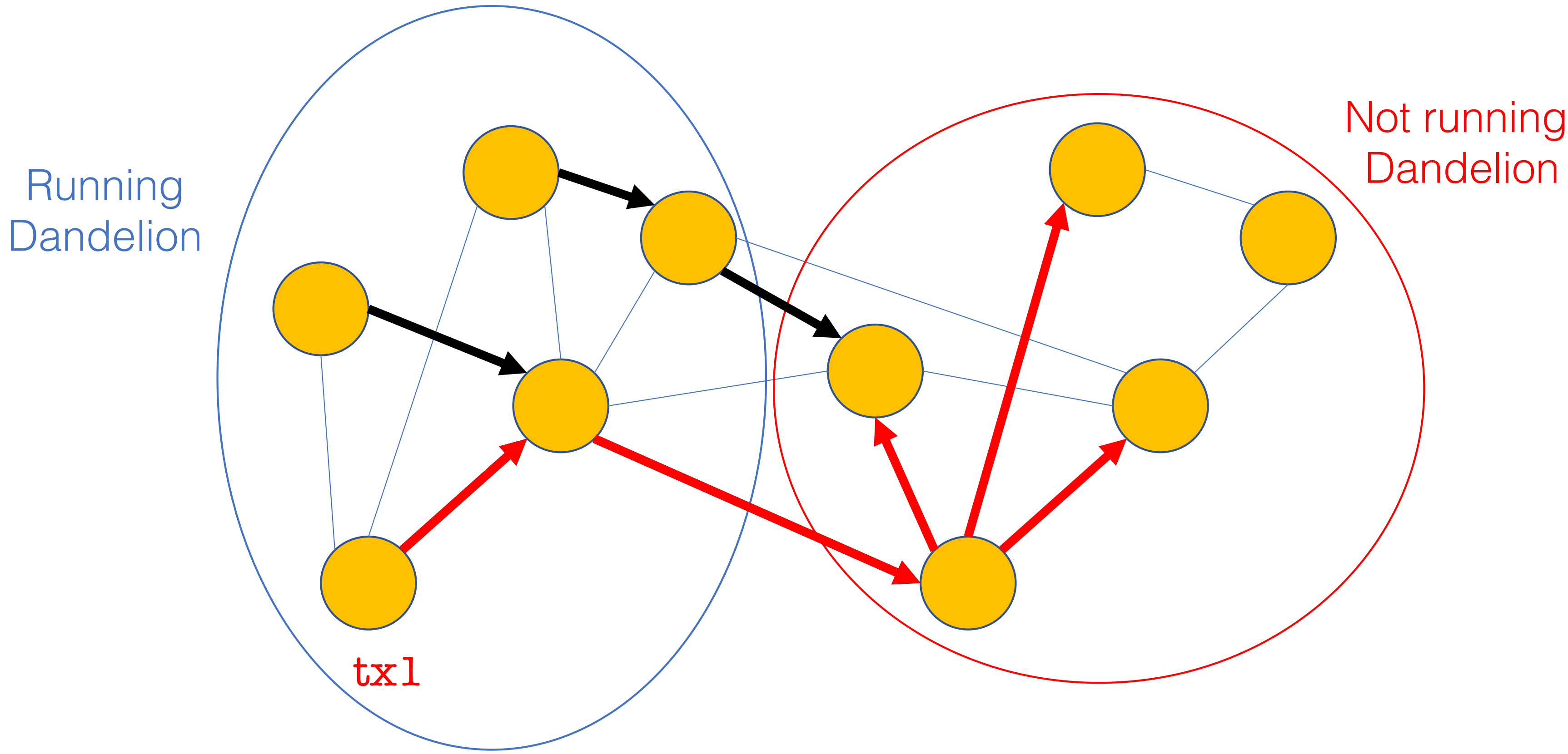


Latency Overhead: Estimate



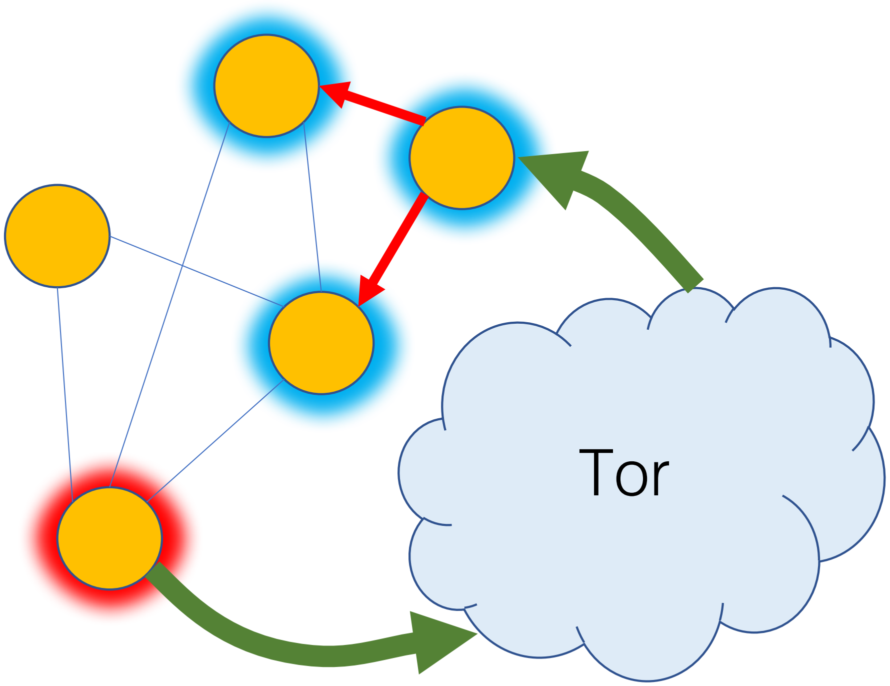
Information Propagation in the Bitcoin Network, Decker and Wattenhofer, 2013

Deployment considerations

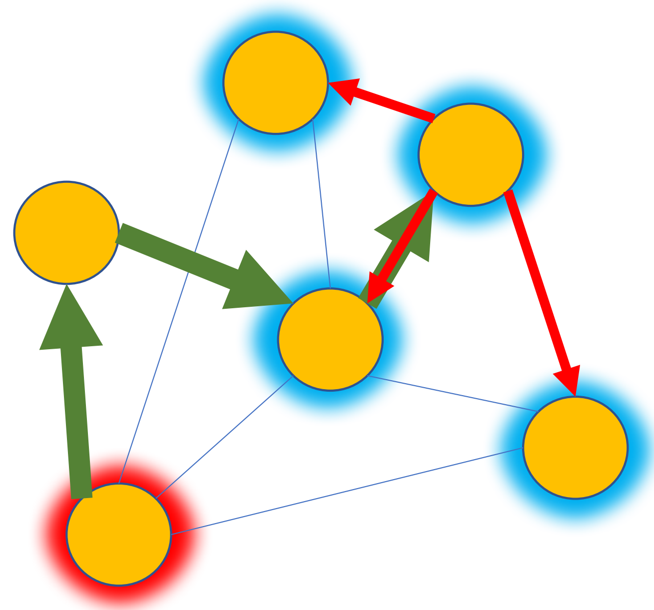


Why not alternative solutions?

Connect through Tor



I2P Integration (e.g. Monero)



Open Problems

- Static graph
  - Modeling user preferences
  - Using cliques for better anonymity on general graphs
- Dynamic graph
  - Characterizing graph learning rate
- Both
  - Intersection attacks!

Conclusion

- Broadcasting information
  - common primitive
  - modern applications
- Performance metrics
  - latency, spreading rate, coverage, anonymity
- Engineering choices
  - underlying topology, spreading protocol
- **Finding** the source
  - Inferring the network topology
- **Hiding** the source