Stabilization and expansion of simple dynamic random graph models for Bitcoin-like unstructured P2P networks

Francesco Pasquale[♦]

based on a joint work with

L. Becchetti[♥], A. Clementi[♦], E. Natale[♠], and L. Trevisan[♣]

[♦]Università di Roma Tor Vergata,[♥] Sapienza Università di Roma,

[♠]Université Côte d'Azur, [♣]Bocconi University

Seminari di Logica e Informatica Teorica
Università di Roma Tre

Rome, July 3rd, 2020

Cryptocurrencies: The Bitcoin Revolution

Bitcoin P2P e-cash paper

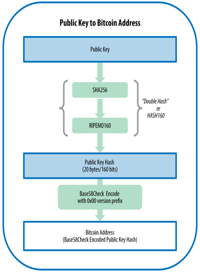
Satoshi Nakamoto | Sat, 01 Nov 2008 16:16:33 -0700

```
I've been working on a new electronic cash system that's fully
peer-to-peer, with no trusted third party.

The paper is available at:
http://www.bitcoin.org/bitcoin.pdf

The main properties:
 Double-spending is prevented with a peer-to-peer network.
 No mint or other trusted parties.
 Participants can be anonymous.
 New coins are made from Hashcash style proof-of-work.
 The proof-of-work for new coin generation also powers the
    network to prevent double-spending.
```

The Bitcoin system

Bitcoin

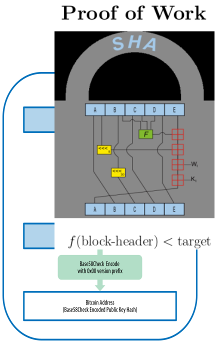
- **Addresses**



The Bitcoin system

Bitcoin

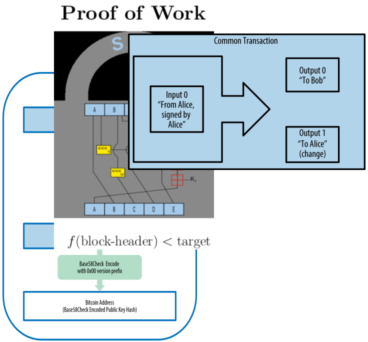
- Addresses
- **Mining**



The Bitcoin system

Bitcoin

- Addresses
- Mining
- **Transactions**

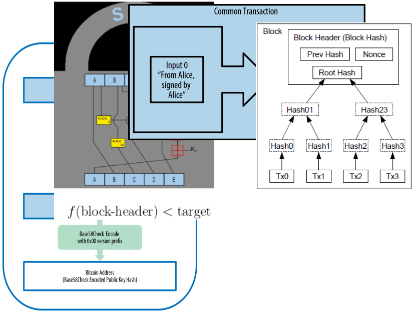


The Bitcoin system

Bitcoin

- Addresses
- Mining
- Transactions
- **Blocks**

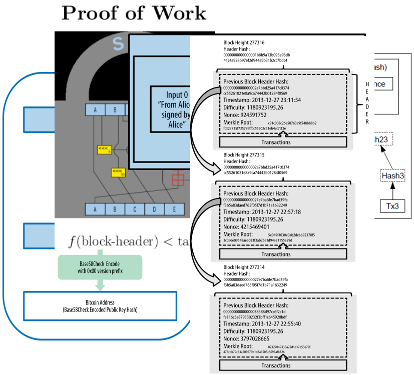
Proof of Work



The Bitcoin system

Bitcoin

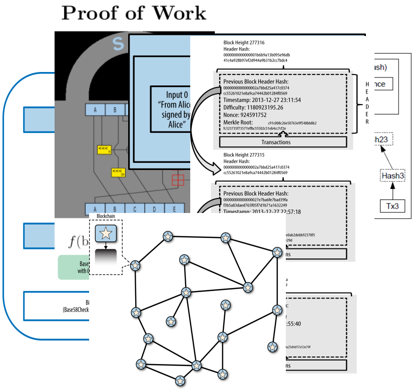
- Addresses
- Mining
- Transactions
- Blocks
- **Blockchain**



The Bitcoin system

Bitcoin

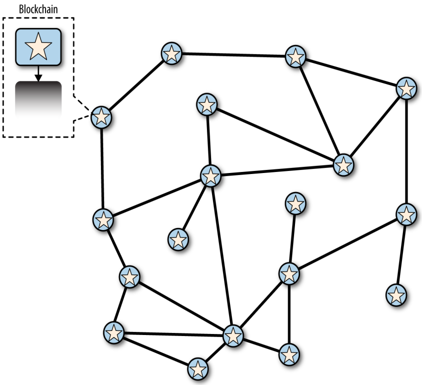
- Addresses
- Mining
- Transactions
- Blocks
- Blockchain
- **P2P network**



The Bitcoin system

Bitcoin

- Addresses
- Mining
- Transactions
- Blocks
- Blockchain
- P2P network



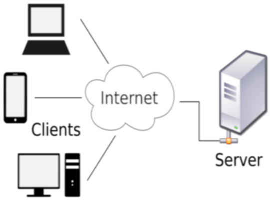
In this talk

1. Bitcoin P2P network formation process
2. Random graphs and dynamic random graph models
3. A dynamic random graph model for Bitcoin-like P2P networks
4. Hints at model analysis and proof techniques
5. Further results and research directions

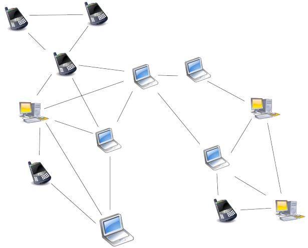
Bitcoin P2P network formation process

Peer-to-Peer Networks

Client-server architecture



P2P networks



- Each node is both client and server

The Bitcoin P2P Network

BITNODES

Bitnodes is currently being developed to estimate the size of the Bitcoin network by finding all the reachable nodes in the network.

GLOBAL BITCOIN NODES DISTRIBUTION

Reachable nodes as of Fri Jul 03 2020 12:38:23 GMT+0200 (GMT+02:00).

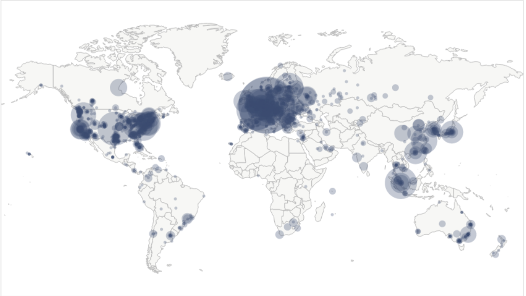
10370 NODES

24-hour charts »

Top 10 countries with their respective number of reachable nodes are as follow.

RANK	COUNTRY	NODES
1	n/a	2392 (23.07%)
2	United States	1950 (18.80%)
3	Germany	1795 (17.31%)
4	France	582 (5.61%)
5	Netherlands	430 (4.15%)
6	Canada	288 (2.78%)
7	Singapore	255 (2.46%)
8	United Kingdom	251 (2.42%)
9	Russian Federation	220 (2.12%)
10	China	188 (1.81%)

More (102) »



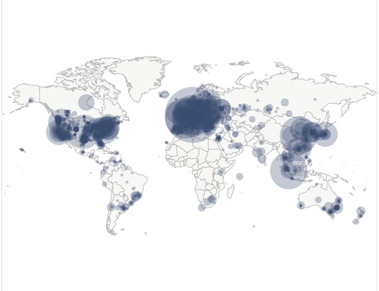
Map shows concentration of reachable Bitcoin nodes found in countries around the world.

LIVE MAP

The Bitcoin P2P Network

Network formation

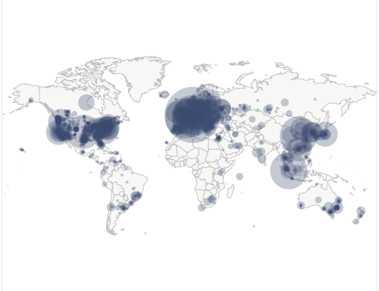
- Initially: DNS queries
- List of active nodes periodically updated and advertised
- Minimum of 8 connections initiated
- Maximum 125 connections



The Bitcoin P2P Network

Network formation

- Initially: DNS queries
- List of active nodes periodically updated and advertised
- Minimum of 8 connections initiated
- Maximum 125 connections



Question

Network structure?

Bitcoin Topology Inference

- Miller et al.
  Discovering bitcoins public topology and influential nodes
  2015
- Neudecker et al.
  Timing analysis for inferring the topology of the bitcoin peer-to-peer network
  2016
- Delgado-Segura et al.
  TxProbe: Discovering Bitcoins Network Topology Using Orphan Transactions
  2018

Random graphs and dynamic random graph models

Math models for networks: Graphs and Random graphs

Graph: G = (V, E)

- V finite set
- $E \subseteq \binom{V}{2}$

Math models for networks: Graphs and Random graphs

Graph: G = (V, E)

- V finite set
- $E \subseteq \binom{V}{2}$

Random Graph

Probability distribution over a set of graphs

Example: Erdös-Rényi $G_{n,p}$

Defined by two parameters: $n \in \mathbb{N}, p \in (0, 1)$

- $V = [n] = \{1, 2, \dots, n\}$
- Each pair of nodes $\{u, v\} \in \binom{V}{2}$ is an edge

Math models for networks: Graphs and Random graphs

Graph: G = (V, E)

- V finite set
- $E \subseteq \binom{V}{2}$

Random Graph

Probability distribution over a set of graphs

Example: Erdös-Rényi $G_{n,p}$

Defined by two parameters: $n \in \mathbb{N}, p \in (0, 1)$

- $V = [n] = \{1, 2, \ldots, n\}$
- Each pair of nodes $\{u, v\} \in \binom{V}{2}$ is an edge

Probabilistic questions

- What is the probability that a $G_{n,p}$ is connected?
- What is the expected diameter?

Math models for networks: Graphs and Random graphs

Graph: G = (V, E)

- V finite set
- $E \subseteq \binom{V}{2}$

Random Graph

Probability distribution over a set of graphs

Example: Erdös-Rényi $G_{n,p}$

Defined by two parameters: $n \in \mathbb{N}, p \in (0, 1)$

- $V = [n] = \{1, 2, \ldots, n\}$
- Each pair of nodes $\{u, v\} \in \binom{V}{2}$ is an edge

Probabilistic questions

- What is the probability that a $G_{n,p}$ is connected?
- What is the expected diameter?

Observation

Answers must be functions of n and p only.

Math models for networks: Graphs and Random graphs

Graph: G = (V, E)

- V finite set
- $E \subseteq \binom{V}{2}$

Random Graph

Probability distribution over a set of graphs

Example: Erdös-Rényi $G_{n,p}$

Defined by two parameters: $n \in \mathbb{N}, p \in (0,1)$

- $V = [n] = \{1, 2, \ldots, n\}$
- Each pair of nodes $\{u, v\} \in \binom{V}{2}$ is an edge

Probabilistic questions

- What is the probability that a $G_{n,p}$ is connected?

ER'58 Threshold at $p = \frac{\log n}{n}$

- What is the expected diameter?

ER'58 $\Theta(\log n / \log np)$

Observation

Answers must be functions of n and p only.

Dynamic Random Graphs

Dynamic Graphs and Random Models

Evolving graphs

An **evolving graph** $\mathcal{G}$ is a sequence of graphs $\mathcal{G} = \{G_t \, : \, t \in \mathbb{N}\}$.

- $G_t = (V_t, E_t)$ where
- V_t finite set
- $E_t \subseteq \binom{V_t}{2}$.

Dynamic Random Graphs

Dynamic Graphs and Random Models

Evolving graphs

An **evolving graph** $\mathcal{G}$ is a sequence of graphs $\mathcal{G} = \{G_t : t \in \mathbb{N}\}$.

- $G_t = (V_t, E_t)$ where
- V_t finite set
- $E_t \subseteq \binom{V_t}{2}$.

Random evolving graphs

V_t and E_t can be random sets.

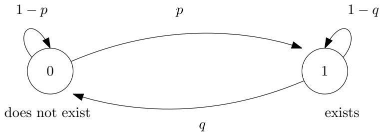
Dynamic Random Graphs

Edge-Markovian evolving graphs and Flooding

Example: Edge-Markovian evolving graph $\mathcal{G}(n, p, q, E_0)$

- $V_t = [n] = \{1, 2, \ldots, n\}$
- $E_t = \left\{ e \in \binom{[n]}{2} \; : \; X_t(e) = 1 \right\}$ for every $e \in \binom{[n]}{2}$, $\{X_t(e) \; : \; t \in \mathbb{N}\}$ is a Markov chain with transition matrix

$$M = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 1-p & p \\ 1 & q & 1-q \end{array}$$



- p: Edge **birth-rate**;
- q: Edge **death-rate**

Edge-Markovian Evolving Graphs

Properties

Markov chain

$$M = \begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix}$$

Stationary distribution

$$\pi = \left(\frac{q}{p+q}, \frac{p}{p+q} \right)$$

Edge-Markovian Evolving Graphs

Properties

Markov chain

$$M = \begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix}$$

Stationary distribution

$$\pi = \left(\frac{q}{p+q}, \frac{p}{p+q} \right)$$

Observations

- The graph eventually converge to an Erdös-Rényi random graph $G_{n,\hat{p}}$ with edge-probability
$$\hat{p} = \frac{p}{p+q}$$
- If $q = 1 - p$ the evolving graph is a sequence of independent $G_{n,p}$.

Edge-Markovian evolving graphs

Flooding

Flooding Process

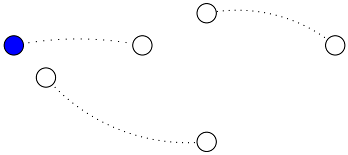
- Start with one single *informed* node;
- When a node **gets in touch** with an informed node, then it **collects** the information.

Edge-Markovian evolving graphs

Flooding

Flooding Process

- Start with one single *informed* node;
- When a node **gets in touch** with an informed node, then it **collects** the information.

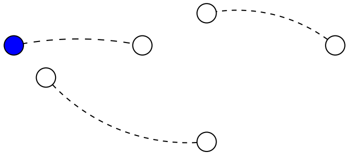


Edge-Markovian evolving graphs

Flooding

Flooding Process

- Start with one single *informed* node;
- When a node **gets in touch** with an informed node, then it **collects** the information.

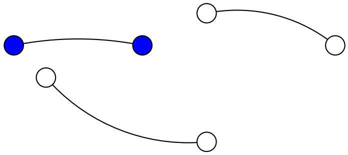


Edge-Markovian evolving graphs

Flooding

Flooding Process

- Start with one single *informed* node;
- When a node **gets in touch** with an informed node, then it **collects** the information.

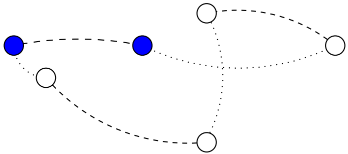


Edge-Markovian evolving graphs

Flooding

Flooding Process

- Start with one single *informed* node;
- When a node **gets in touch** with an informed node, then it **collects** the information.

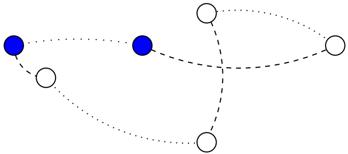


Edge-Markovian evolving graphs

Flooding

Flooding Process

- Start with one single *informed* node;
- When a node **gets in touch** with an informed node, then it **collects** the information.

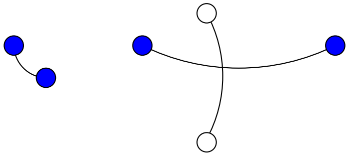


Edge-Markovian evolving graphs

Flooding

Flooding Process

- Start with one single *informed* node;
- When a node **gets in touch** with an informed node, then it **collects** the information.

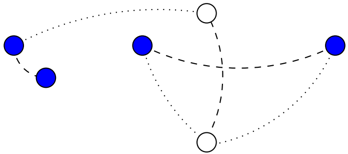


Edge-Markovian evolving graphs

Flooding

Flooding Process

- Start with one single *informed* node;
- When a node **gets in touch** with an informed node, then it **collects** the information.

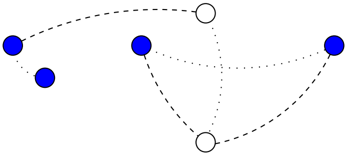


Edge-Markovian evolving graphs

Flooding

Flooding Process

- Start with one single *informed* node;
- When a node **gets in touch** with an informed node, then it **collects** the information.

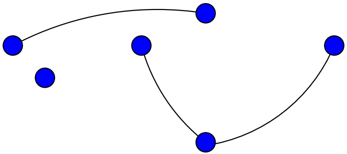


Edge-Markovian evolving graphs

Flooding

Flooding Process

- Start with one single *informed* node;
- When a node **gets in touch** with an informed node, then it **collects** the information.



Edge-Markovian evolving graphs

Flooding

Flooding Process

- Start with one single *informed* node;
- When a node **gets in touch** with an informed node, then it **collects** the information.

Theorem (Clementi et al., 2008)

Flooding Time of Edge-MEG $\mathcal{G}(n, p, q, E_0)$ *is w.h.p.*

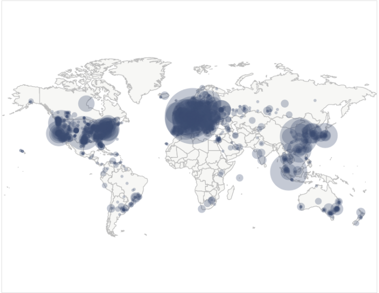
$$\mathcal{O}\left(\frac{\log n}{\log(1+np)}\right)$$

even for $E_0 = \emptyset$ *and* $q = 1$.

A dynamic random graph model for Bitcoin-like P2P networks

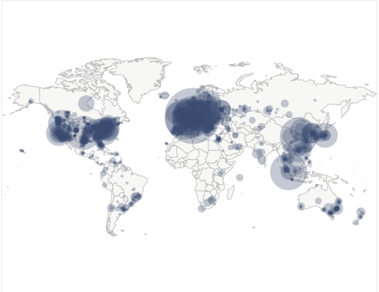
Dynamic Random Graph Model for Bitcoin P2P network

- Initially: DNS queries
- List of active nodes periodically updated and advertised
- Minimum of 8 connections initiated
- Maximum 125 connections



Dynamic Random Graph Model for Bitcoin P2P network

- Initially: DNS queries
- List of active nodes periodically updated and advertised
- Minimum of 8 connections initiated
- Maximum 125 connections



Idea: $G(n, d, c)$ model

n: Number of nodes

$d \in \mathbb{N}$: target degree

$c \geqslant 1$: tolerance

- Start with an empty graph
- When a node has less than d neighbors:
  Try to connect to new nodes u.a.r. among all nodes
- When a node has more than cd neigbors:
  Disconnect from some neighbors

RAES

Request a link, Accept if Enough Space

Directed graph $G = (V, E)$

- $d_{\text{out}} = d$ (outgoing links)
- $d_{\text{in}} \leqslant cd$ (max number of incoming links)

At each round, each node $u \in [n]$, independently of the other nodes:

- If $d_{out}(u) < d$:
"Request" link to $d - d_{out}$ new nodes u.a.r.

- If $d_{in}(u) > cd$:
"Reject" all requests of the last round

RAES

Request a link, Accept if Enough Space

Directed graph $G = (V, E)$

- $d_{\text{out}} = d$ (outgoing links)
- $d_{\text{in}} \leqslant cd$ (max number of incoming links)

At each round, each node $u \in [n]$, independently of the other nodes:

- If $d_{out}(u) < d$:
"Request" link to $d - d_{out}$ new nodes u.a.r.

- If $d_{in}(u) > cd$:
"Reject" all requests of the last round

Observation

Once a link is "accepted" it is "settles"

Questions: Termination time and Structure

Observation

When (If) the process terminates all nodes have $d_{\text{out}} = d$ and $d_{\text{in}} \leqslant cd$.

Questions: Termination time and Structure

Observation

When (If) the process terminates all nodes have $d_{\text{out}} = d$ and $d_{\text{in}} \leqslant cd$.

Question 1

How long it takes the graph to settle?

Questions: Termination time and Structure

Observation

When (If) the process terminates all nodes have $d_{\text{out}} = d$ and $d_{\text{in}} \leqslant cd$.

Question 1

How long it takes the graph to settle?

Question 2

What is the “structure” of the resulting graph?

Hints at model analysis and proof techniques

Termination time

Question 1

How long it takes the graph to settle?

Theorem (Termination)

For every $d \geqslant 1$, $c > 1$, and $\beta > 1$, process terminates within $\beta \log(n)/\log(c)$ rounds, with probability at least $1 - d/n^{\beta-1}$.

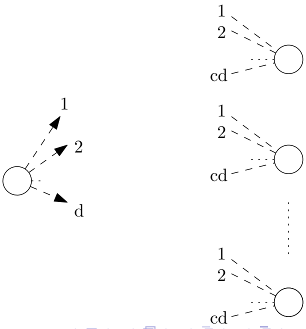
Termination time

Theorem (Termination)

For every $d \geqslant 1$, $c > 1$, and $\beta > 1$, process terminates within $\beta \log(n)/\log(c)$ rounds, with probability at least $1 - d/n^{\beta-1}$.

Proof sketch

- *nd* total links to be settled



Termination time

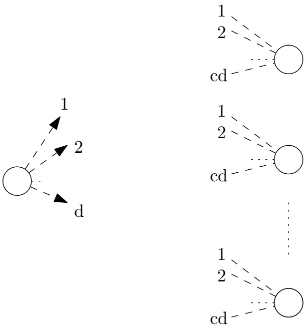
Theorem (Termination)

For every $d \geqslant 1$, $c > 1$, and $\beta > 1$, process terminates within $\beta \log(n)/\log(c)$ rounds, with probability at least $1 - d/n^{\beta-1}$.

Proof sketch

- nd total links to be settled
- For $i = 1, \ldots, nd$

$$X_i^{(t)} = \begin{cases} 1 & \text{if link requests } i \text{ is settled at round } t \\ 0 & \text{otherwise} \end{cases}$$

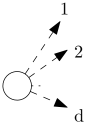


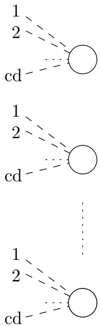
Termination time

Theorem (Termination)

For every $d \geqslant 1$, $c > 1$, and $\beta > 1$, process terminates within $\beta \log(n)/\log(c)$ rounds, with probability at least $1 - d/n^{\beta-1}$.

Proof sketch





- nd total links to be settled
- For $i = 1, \ldots, nd$

$$X_i^{(t)} = \begin{cases} 1 & \text{if link requests } i \text{ is settled at round } t \\ 0 & \text{otherwise} \end{cases}$$

- At each round there are at most $nd/cd = n/c$ nodes that receive cd or more link requests

Termination time

Theorem (Termination)

For every $d \geqslant 1$, $c > 1$, and $\beta > 1$, process terminates within $\beta \log(n)/\log(c)$ rounds, with probability at least $1 - d/n^{\beta-1}$.

Proof sketch

- Hence, at any attempt, prob that link i does not settle is at most $1/c$

$$\mathbf{P}\left(X_i^{(t)} = 0 \mid X_i^{(t-1)} = 0\right) \leqslant (n/c)/n = 1/c$$

Termination time

Theorem (Termination)

For every $d \geqslant 1$, $c > 1$, and $\beta > 1$, process terminates within $\beta \log(n)/\log(c)$ rounds, with probability at least $1 - d/n^{\beta-1}$.

Proof sketch

- Hence, at any attempt, prob that link i does not settle is at most $1/c$

$$\mathbf{P}\left(X_i^{(t)} = 0 \,|\, X_i^{(t-1)} = 0\right) \leqslant (n/c)/n = 1/c$$

- Hence $\mathbf{P}\left(X_i^{(t)} = 0\right) \leqslant (1/c)^t$

Termination time

Theorem (Termination)

For every $d \geqslant 1$, $c > 1$, and $\beta > 1$, process terminates within $\beta \log(n)/\log(c)$ rounds, with probability at least $1 - d/n^{\beta-1}$.

Proof sketch

- Hence, at any attempt, prob that link i does not settle is at most $1/c$

$$\mathbf{P}\left(X_i^{(t)} = 0 \,|\, X_i^{(t-1)} = 0\right) \leqslant (n/c)/n = 1/c$$

- Hence $\mathbf{P}\left(X_i^{(t)} = 0\right) \leqslant (1/c)^t$
- Finally, for $t > 2\log_c n$

$$\mathbf{P}\left(X_i^{(t)} = 0\right) \leqslant \frac{1}{n^2} \text{ and } \mathbf{P}\left(\exists i \in \{1, \ldots, nd\} \,:\, X_i^{(t)} = 0\right) \leqslant \frac{nd}{n^2} = d/n$$

Expansion

Question 2

"Structure" of the resulting graph?

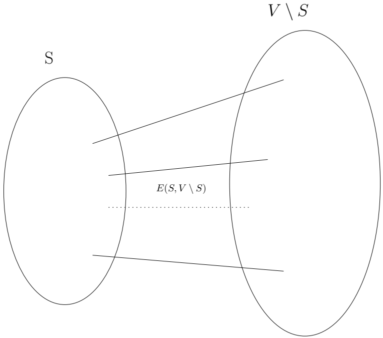
Expansion

Question 2

"Structure" of the resulting graph?

Expander Graph

$G = (V, E)$ is ε-*expander* if, for every subset $S \subset V$ with $|S| \leqslant |V|/2$, number of edges in the cut $E(S, V \setminus S)$ at least $\varepsilon \cdot |S|$.



- $E(S, V \setminus S) = \{\{u, v\} \in E \,:\, u \in S, v \in V \setminus S\}$.

Expansion

Question 2

"Structure" of the resulting graph?

Theorem (Expansion)

A sufficiently-small constant $\varepsilon > 0$ *exists such that for sufficiently large constants* d *and* c *resulting random graph* G *is* ε*-expander, w.h.p.*

Expansion

Question 2

"Structure" of the resulting graph?

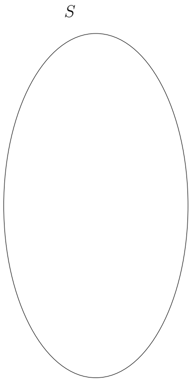
Theorem (Expansion)

A sufficiently-small constant $\varepsilon > 0$ exists such that for sufficiently large constants d and c resulting random graph G is ε-expander, w.h.p.

Proof based on "Encoding/Compression argument".

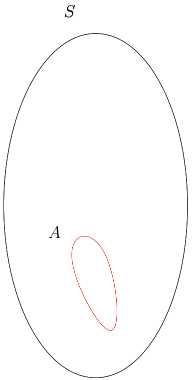
Encoding/Compression argument idea

- $S = \{0,1\}^n$ set of n-bit strings:
  $|S| = 2^n$



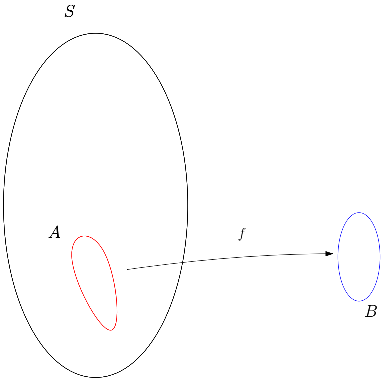
Encoding/Compression argument idea

- $S = \{0, 1\}^n$ set of n-bit strings: $|S| = 2^n$
- $A \subseteq S$ implicitly defined, we want to upper bound $|A|$



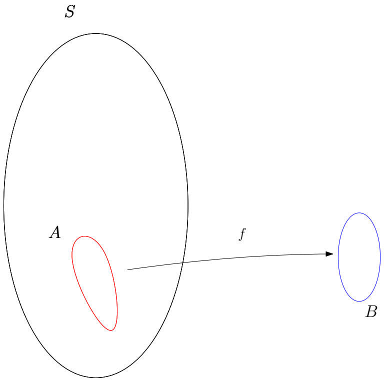
Encoding/Compression argument idea

- $S = \{0,1\}^n$ set of n-bit strings: $|S| = 2^n$
- $A \subseteq S$ implicitly defined, we want to upper bound $|A|$
- B set of $(n-k)$-bit strings: $|B| = 2^{n-k}$
- Injective map $f : A \longrightarrow B$



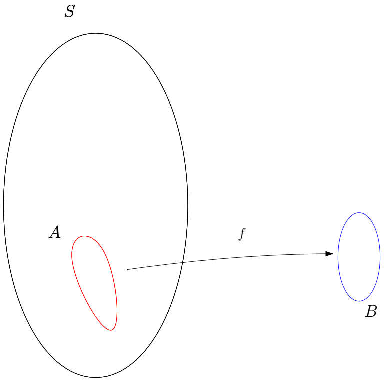
Encoding/Compression argument idea

- $S = \{0,1\}^n$ set of n-bit strings: $|S| = 2^n$
- $A \subseteq S$ implicitly defined, we want to upper bound $|A|$
- B set of $(n-k)$-bit strings: $|B| = 2^{n-k}$
- Injective map $f : A \longrightarrow B$
- Then $|A| \leqslant 2^{n-k}$



Encoding/Compression argument idea

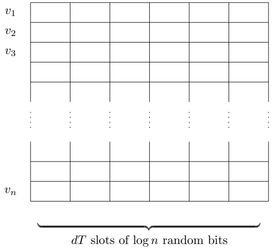
- $S = \{0,1\}^n$ set of n-bit strings: $|S| = 2^n$
- $A \subseteq S$ implicitly defined, we want to upper bound $|A|$
- B set of $(n-k)$-bit strings: $|B| = 2^{n-k}$
- Injective map $f : A \longrightarrow B$
- Then $|A| \leqslant 2^{n-k}$
- If one picks $x \in S$ u.a.r., then $\mathbf{P}(x \in A) \leqslant 2^{-k}$



Expansion

Encoding argument for RAES

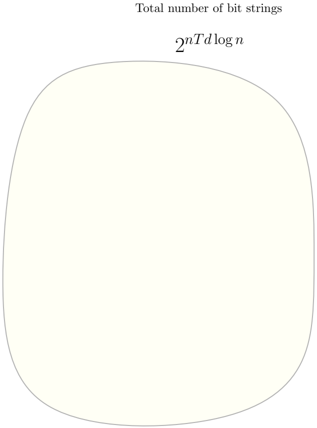
- An execution of the RAES graph dynamics for T rounds completely determined by a string of $nTd \log n$ bits
  - n nodes
  - T rounds
  - d out-neighbors
  - $\log n$ bits per sample



Expansion

Encoding argument for RAES

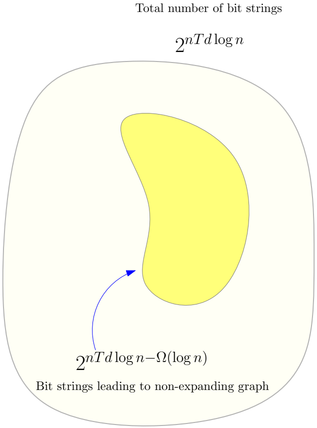
- An execution of the RAES graph dynamics for T rounds completely determined by a string of $nTd\log n$ bits



Expansion

Encoding argument for RAES

- An execution of the RAES graph dynamics for T rounds completely determined by a string of $nTd \log n$ bits
- Any bit string $R \in \{0,1\}^{nTd \log n}$ leading to a non-expanding graph can be "encoded" lossless with $nTd \log n - \Omega(\log n)$ bits.



Encoding argument

Proof Idea

- **If G is not an expander, then there is non-expanding set of nodes S...**

Encoding argument

Proof Idea

- If G is not an expander, then there is non-expanding set of nodes S...
- **...then the typical node in S will have a lot of neighbors in S...**

Encoding argument

Proof Idea

- If G is not an expander, then there is non-expanding set of nodes S...
- ...then the typical node in S will have a lot of neighbors in S...
- **...then we can "encode" those link requests with** $\log |S|$ **bits instead of** $\log n$**...**

Encoding argument

Proof Idea

- If G is not an expander, then there is non-expanding set of nodes S...
- ...then the typical node in S will have a lot of neighbors in S...
- ...then we can "encode" those link requests with $\log|S|$ bits instead of $\log n$...
- **...provided that we already "encoded" who's the set S...**

Encoding argument

Proof Idea

- If G is not an expander, then there is non-expanding set of nodes S...
- ...then the typical node in S will have a lot of neighbors in S...
- ...then we can "encode" those link requests with $\log |S|$ bits instead of $\log n$...
- ...provided that we already "encoded" who's the set S...
- **...this takes $\log \binom{n}{|S|}$ bits...**

Encoding argument

Proof Idea

- If G is not an expander, then there is non-expanding set of nodes S...
- ...then the typical node in S will have a lot of neighbors in S...
- ...then we can "encode" those link requests with $\log |S|$ bits instead of $\log n$...
- ...provided that we already "encoded" who's the set S...
- ...this takes $\log \binom{n}{|S|}$ bits...
- **...but we can save other bits suitably encoding accepted and rejected requests**

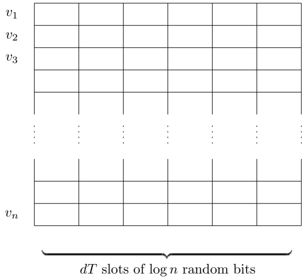
Encoding argument

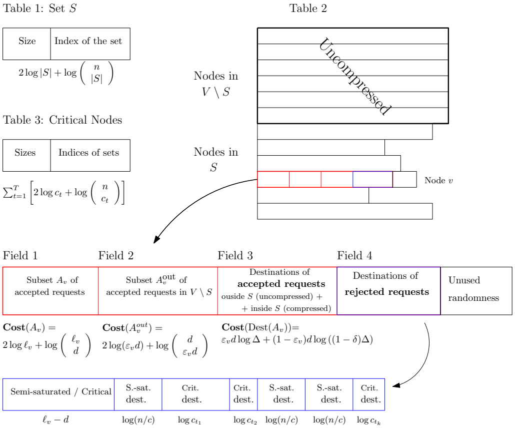
Proof Idea

- If G is not an expander, then there is non-expanding set of nodes S...
- ...then the typical node in S will have a lot of neighbors in S...
- ...then we can "encode" those link requests with $\log |S|$ bits instead of $\log n$...
- ...provided that we already "encoded" who's the set S...
- ...this takes $\log \binom{n}{|S|}$ bits...
- ...but we can save other bits suitably encoding accepted and rejected requests
- **...**

Encoding argument

Proof Idea





Further results and research directions

Bounded-degree expander inside a dense one

In this talk

Each node can sample **any** other node.

Bounded-degree expander inside a dense one

In this talk

Each node can sample **any** other node.

Slight generalization

Underlying Δ-regular graph G with $\Delta = \Theta(n)$. Nodes can sample only their neighbors in G.

Bounded-degree expander inside a dense one

In this talk

Each node can sample **any** other node.

Slight generalization

Underlying Δ-regular graph G with $\Delta = \Theta(n)$. Nodes can sample only their neighbors in G.

Theorem (Bounded-degree expander inside a dense one)

For sufficiently-small $\varepsilon > 0$, sufficiently-large d, and for any $0 < \alpha \leqslant 1$, a large-enough c exists such that for every Δ-regular underlying graph G with $\Delta = \alpha n$, if second largest eigenvalue of adjacency matrix of G is $\lambda_2 \leqslant \varepsilon \alpha^2 \Delta$, then RAES(G,d,c) produces a ε-expander, w.h.p.

Node churns

Observation

In RAES the set of nodes is fixed.

Node churns

Observation

In RAES the set of nodes is fixed.

Future work

Include **nodes joining and leaving** the network in the model.
At each round:

- N new nodes join (e.g., $N \sim Po(\lambda)$) and start executing RAES;
- Each node independently disappears with probability p.

Node churns

Observation

In RAES the set of nodes is fixed.

Future work

Include **nodes joining and leaving** the network in the model.
At each round:

- N new nodes join (e.g., $N \sim Po(\lambda)$) and start executing RAES;
- Each node independently disappears with probability p.

- Markovian process with no absorbing states
- Expected number of nodes at each round (in the long run) λ/p
- How to measure "quality" of the evolving random graph

L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and L. Trevisam.
Finding a Bounded-Degree Expander Inside a Dense One.
In *Proc. ACM-SIAM Symp. on Discrete Algorithms (SODA'20)*. SIAM, 2020.

Thank you!