Self-stabilization and expansion of a simple dynamic random graph model for Bitcoin-like unstructured P2P networks

Francesco Pasquale[♦]

based on a joint work with

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

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

[♣]UC Berkeley, [♠]Université Côte d'Azur

Algorithmic Aspects of Temporal Graphs II

Patras, Greece, July 9, 2019

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

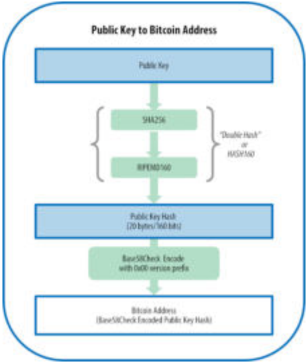
The main properties:
 Double-spending is prevented with a peer-to-peer network.
 No mint or other trusted parties.
 Participants can be anonymous.
 New coins are made from Hashcash style proof-of-work.
 The proof-of-work for new coin generation also powers the
 network to prevent double-spending.

Cryptocurrencies: The Bitcoin Revolution

Bitcoin

- **Addresses**

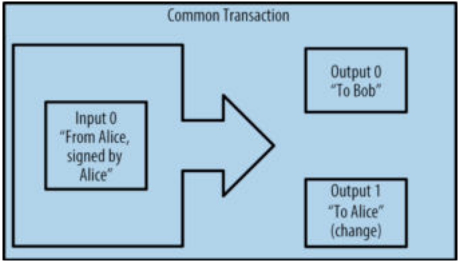
1C78STt6GqKNob7wnX9kWFcTLrPe39jRRQ



Cryptocurrencies: The Bitcoin Revolution

Bitcoin

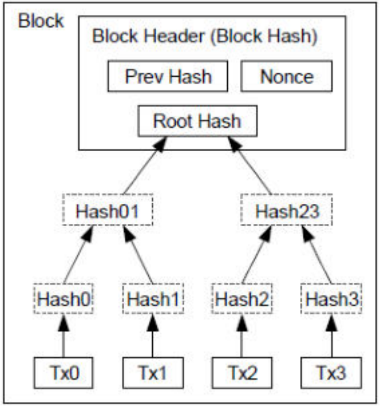
- Addresses
- **Transactions**



Cryptocurrencies: The Bitcoin Revolution

Bitcoin

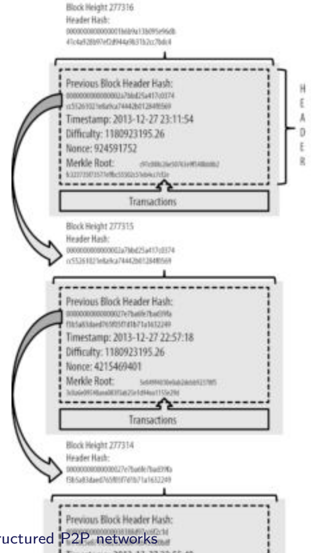
- Addresses
- Transactions
- **Blocks**



Cryptocurrencies: The Bitcoin Revolution

Bitcoin

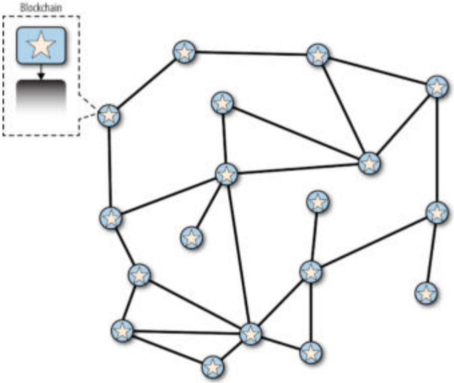
- Addresses
- Transactions
- Blocks
- **Blockchain**



Cryptocurrencies: The Bitcoin Revolution

Bitcoin

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

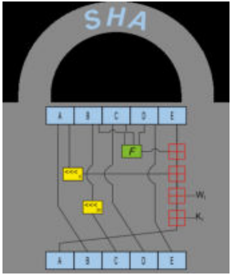


Cryptocurrencies: The Bitcoin Revolution

Bitcoin

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

Proof of Work



$f(\text{block-header}) < \text{target}$

The Bitcoin P2P Network

GLOBAL BITCOIN NODES DISTRIBUTION

Reachable nodes as of Sun Jul 07 2019 22:40:19 GMT+0200 (CEST).

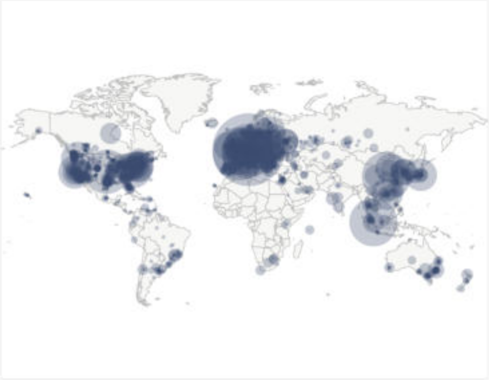
10118 NODES

24-hour charts »

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

RANK	COUNTRY	NODES
1	United States	2456 (24.27%)
2	Germany	1931 (19.08%)
3	n/a	624 (6.17%)
4	France	612 (6.05%)
5	Netherlands	514 (5.08%)
6	China	426 (4.21%)
7	Canada	344 (3.40%)
8	United Kingdom	295 (2.92%)
9	Singapore	286 (2.83%)
10	Russian Federation	246 (2.43%)

More (103) »

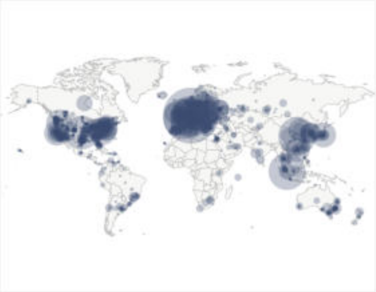


The Bitcoin P2P Network

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

The Bitcoin P2P Network

- 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 bitcoin's 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 Bitcoin's Network Topology Using Orphan Transactions
  2018

A simple dynamic graph model

$G(n, d, c)$ dynamic random graph

$$\left\{ \begin{array}{ccl} n & : & \text{number of nodes} \\ d & : & \text{minimum required degree} \\ c \geqslant 1 & : & \text{“tolerance”} \ (cd = \text{maximum degree}) \end{array} \right.$$

A simple dynamic graph model

$G(n, d, c)$ dynamic random graph

$$\begin{cases} n & : & \text{number of nodes} \\ d & : & \text{minimum required degree} \\ c \geqslant 1 & : & \text{“tolerance”} \ (cd = \text{maximum degree}) \end{cases}$$

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

A simple dynamic graph model

$G(n, d, c)$ dynamic random graph

$$\begin{cases} n & : \quad \text{number of nodes} \\ d & : \quad \text{minimum required degree} \\ c \geqslant 1 & : \quad \text{“tolerance”} \ (cd = \text{maximum degree}) \end{cases}$$

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

- If u has degree $< d$ then u picks a node v uniformly at random and adds the edge $\{u, v\}$ in E.

A simple dynamic graph model

$G(n, d, c)$ dynamic random graph

$$\begin{cases} n & : \quad \text{number of nodes} \\ d & : \quad \text{minimum required degree} \\ c \geqslant 1 & : \quad \text{“tolerance” } (cd = \text{maximum degree}) \end{cases}$$

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

- If u has degree $< d$ then u picks a node v uniformly at random and adds the edge $\{u, v\}$ in E.

- If u has degree $> cd$ then u picks one of its neighbors v uniformly at random and removes the edge $\{u, v\}$ from E

Self-stabilization and expansion

When (If) the process terminates all nodes have

$$d \leqslant \text{ degree } \leqslant cd$$

Self-stabilization and expansion

When (If) the process terminates all nodes have

$$d \leqslant \text{ degree } \leqslant cd$$

Question 1

How long it takes to settle?

Self-stabilization and expansion

When (If) the process terminates all nodes have

$$d \leqslant \text{ degree } \leqslant cd$$

Question 1

How long it takes to settle?

Question 2

Structure of the resulting graph?

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 u has $d_{out} < d$ outgoing links then u picks $d - d_{out}$ nodes uniformly at random $v_1, \ldots, v_{d-d_{out}}$ and "requests" edges $\{u, v_1\}, \ldots, \{u, v_{d-d_{out}}\}$*
- *If u receives $> cd$ incoming requests, u "rejects" 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 u has $d_{\text{out}} < d$ outgoing links then u picks $d - d_{\text{out}}$ nodes uniformly at random $v_1, \ldots, v_{d-d_{\text{out}}}$ and "requests" edges $\{u, v_1\}, \ldots, \{u, v_{d-d_{\text{out}}}\}$

- If u receives $> cd$ incoming requests, u "rejects" all requests of the last round

Observation

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

Self-stabilization

Question 1

How long it takes to settle?

Self-stabilization

Question 1

How long it takes to settle?

Stabilization

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

Self-stabilization

Question 1

How long it takes to settle?

Stabilization

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

- Parallel balls-into-bins problem
  (nd "link requests" [balls] in ncd "available slots" [bins])
- At each round each request has constant probability to be accepted

Expansion

Question 2

Structure of the resulting graph?

Expander Graph

A graph $G = (V, E)$ is an ε-*expander* if, for every subset $U \subset V$ with $|U| \leqslant n/2$, number of edges in the cut $(U, V - U)$ at least $\varepsilon \cdot \text{vol}(U)$.

Expansion

Question 2

Structure of the resulting graph?

Expander Graph

A graph $G = (V, E)$ is an ε-*expander* if, for every subset $U \subset V$ with $|U| \leqslant n/2$, number of edges in the cut $(U, V - U)$ at least $\varepsilon \cdot \mathrm{vol}(U)$.

Expansion

Question 2

Structure of the resulting graph?

Expander Graph

A graph $G = (V, E)$ is an ε-*expander* if, for every subset $U \subset V$ with $|U| \leqslant n/2$, number of edges in the cut $(U, V - U)$ at least $\varepsilon \cdot \mathrm{vol}(U)$.

Main difficulty

Complex dependencies among edges of the resulting graph

Expansion

Question 2

Structure of the resulting graph?

Expander Graph

A graph $G = (V, E)$ is an ε-*expander* if, for every subset $U \subset V$ with $|U| \leqslant n/2$, number of edges in the cut $(U, V - U)$ at least $\varepsilon \cdot \mathrm{vol}(U)$.

Main difficulty

Complex dependencies among edges of the resulting graph

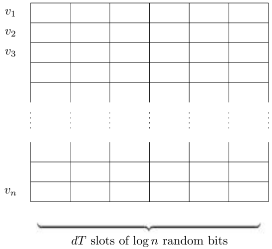
Theorem

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

Proof idea: **Encoding argument**

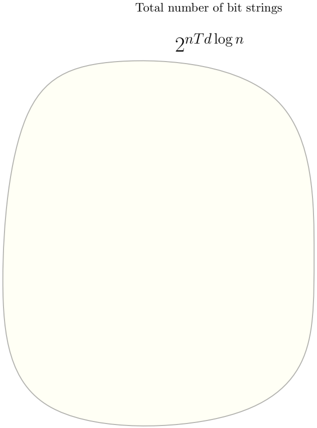
- $nTd \log n$ total random bits
  - n nodes
  - T rounds
  - d out-neighbors
  - $\log n$ bits per sample



Expansion

Proof idea: **Encoding argument**

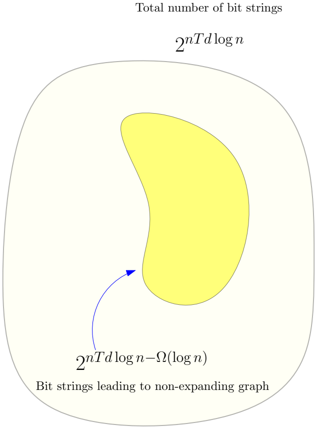
- $nTd \log n$ total random bits



Expansion

Proof idea: **Encoding argument**

- $nTd \log n$ total random bits
- Any bit string $R \in \{0,1\}^{nTd \log n}$ leading to a non-expanding graph can be "encoded" losslessy 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...**

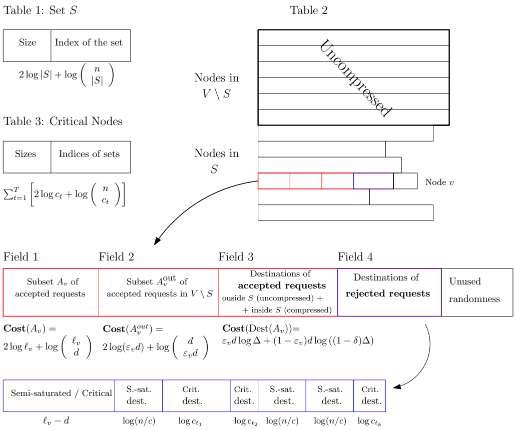
Encoding argument

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



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.

Results smoothly apply to a slight generalization

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

Bounded-degree expander inside a dense one

In this talk

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

Results smoothly apply to a slight generalization

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

Bounded-degree expander from a dense one

Parallel algorithm to find a bounded-degree expander inside a dense one.

Conclusions and future work

Conclusions

- Bitcoin P2P network as an interesting available dynamic network
- The protocol hides the network structure
- Encoding argument to prove properties of random graphs

Conclusions and future work

$G(n, d, c)$ dynamic random graph

$$\begin{cases} n & : \quad \text{number of nodes} \\ d & : \quad \text{minimum required degree} \\ c \geqslant 1 & : \quad \text{"tolerance" } (cd = \text{maximum degree}) \\ & \end{cases}$$

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

- *If u has degree $< d$ then u picks a node v uniformly at random and adds the edge $\{u, v\}$ in E.*
- *If u has degree $> cd$ then u picks one of its neighbors v uniformly at random and removes the edge $\{u, v\}$ from E*

Conclusions and future work

$G(n, d, c, p)$ dynamic random graph

$$
\begin{cases}
n & : \quad \text{number of nodes} \\
d & : \quad \text{minimum required degree} \\
c \geqslant 1 & : \quad \text{“tolerance” } (cd = \text{maximum degree}) \\
p & : \quad \text{edge failure probability}
\end{cases}
$$

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

- *If u has degree $< d$ then u picks a node v uniformly at random and adds the edge $\{u, v\}$ in E.*
- *If u has degree $> cd$ then u picks one of its neighbors v uniformly at random and removes the edge $\{u, v\}$ from E*
- **Each edge $e \in E$ disappears with probability p**

Conclusions and future work

Ergodic Markov Chain

- Mixing time?
- Stationary random graph properties?

Conclusions and future work

Ergodic Markov Chain

- Mixing time?
- Stationary random graph properties?

Proof techniques

Encodign argument doesn't seem to help.

Conclusions and future work

Ergodic Markov Chain

- Mixing time?
- Stationary random graph properties?

Proof techniques

Encodign argument doesn't seem to help.

Further

Nodes joining and leaving the network.

Thank you!