Dynamic graph models inspired by the Bitcoin network-formation process

Antonio Cruciani
Gran Sasso Science Institute
L'Aquila, Italy
antonio.cruciani@gssi.it

Francesco Pasquale
Università di Roma "Tor Vergata"
Roma, Italy
pasquale@mat.uniroma2.it

ABSTRACT

The network formation process in the Bitcoin protocol is designed to hide the global network structure: while most of the nodes of the network can be easily discovered, the existence of an edge between two nodes is only known by the two endpoints. In [Becchetti et al., SODA2020] the authors propose a dynamic random graph model inspired by the network formation process in the Bitcoin protocol and they prove that the evolution of the graph quickly terminates and that the resulting graph is an expander, with high probability.

In this paper we extend the model in [Becchetti et al., SODA2020] to obtain dynamic random graph models that evolve forever: in the first model, edges can be faulty, i.e., each edge at each round disappears with some probability; in the second one, at every round new nodes join the network according to a Poisson process and each node currently in the network disappears with certain probability; in the third one, we consider a combination of the two models above, in which edges can be faulty and nodes can join and leave the network. We run extensive simulations to measure the "flooding time" in the three models, i.e., how long it takes a message starting at a random node to reach all, or almost all, the nodes. The simulations show that, for large ranges of the parameters of the models, the flooding time is short, i.e., compatible with a logarithmic growth, as a function of the number of nodes in the network. Our results also suggest that the default values of the network formation parameters used in the main implementation of the Bitcoin protocol seem overwhelmingly safe with respect to the stability of the network, and they might safely be tuned to reduce network traffic.

CCS CONCEPTS

• **Theory of computation** → **Random network models**; • **Networks** → *Network design principles*; • **Mathematics of computing** → *Markov processes.*

KEYWORDS

Dynamic graphs, Markov chains, P2P networks, simulations

A preliminary version of this paper appeared as Brief Announcement in the Proceedings of the 24th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'22).

ICDCN 2023, January 4–7, 2023, Kharagpur, India

ACM ISBN 978-1-4503-9796-4/23/01…$15.00
https://doi.org/10.1145/3571306.3571398

ACM Reference Format:
Antonio Cruciani and Francesco Pasquale. 2023. Dynamic graph models inspired by the Bitcoin network-formation process. In *24th International Conference on Distributed Computing and Networking (ICDCN 2023), January 4–7, 2023, Kharagpur, India.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3571306.3571398

1 INTRODUCTION

Bitcoin is a cryptocurrency proposed in 2008 by an unknown person or group of people under the pseudonym of Satoshi Nakamoto [23]. The system is built using a clever combination of a few classical cryptographic concepts: cryptographic hash functions [12], digital signature schemes [14], and hash-cash style proof-of-work [15]. Nodes participating in the Bitcoin system are connected toward an unstructured peer-to-peer network [7] running on top of the Internet. The first version of the software was released by Satoshi Nakamoto in January 2009. The most widely used implementation coming from that initial release, Bitcoin-core [24], is currently under active development. In this paper we are concerned with dynamic graph models inspired by the network formation process of the Bitcoin P2P network. We refer the reader interested in a complete description of the Bitcoin system to [2, 25].

After an initial bootstrap in which they rely on DNS seeds for node discovery, nodes running the Bitcoin-core implementation turn to a fully-decentralized policy to regenerate their neighbors when their degree drops below the configured threshold [11]. Each node has a "target out-degree value" and a "maximum degree value" (respectively 8 and 125, in the default configuration) and it locally stores a large list of (ip addresses of) "active" nodes. Every time the number of current neighbors of a node is below the configured target value it tries to create new connections with nodes sampled from its list. The list stored by a node is initially started with nodes received in response to queries to DNS seeds, then it is periodically advertised to its neighbors and updated with the lists advertised by the neighbors. Hence, in the long run each node samples its out-neighbors from a list formed by a "sufficiently random" subset of all the nodes of the network.

While most of the nodes of the network can be easily discovered [36], the existence of an edge between two nodes is only known by the two endpoints. The topology of the Bitcoin network is thus hidden by the network formation protocol. Indeed, discovering the network structure has been recently an active research topic [13, 27].

1.1 Our contribution.

RAES (Request a link, then Accept if Enough Space) [8] is a directed random graph model defined by three parameters $n \in \mathbb{N}, d \in$

$\{1, \ldots, n-1\}, c > 1$, in which each one of n nodes has out-degree exactly d and in-degree at most cd. The random graph is generated according to the following discrete random process: The graph starts with no edges, and at every round each node u with out-degree $d_u^{\text{out}} < d$ picks $d - d_u^{\text{out}}$ nodes *uniformly at random (u.a.r.)* (with repetitions) and, for each such node v, u "requests" a directed link (u, v); If a node v receives a number of link-requests that would make its in-degree larger than cd, then v rejects all requests received in the current round, otherwise v accepts all requests of the round. The process terminates when all nodes have out-degree d (and in-degree at most cd).

The RAES model can be seen as a simplified version of the network-formation process implemented in Bitcoin-core [11]. However, it lacks one of the crucial aspects of the real network: the *dynamics*, i.e., the fact that nodes can join and leave the network at any time and edges can be faulty. In this paper we consider an undirected version of RAES and we extend the random graph model in two ways, both of them generating *dynamic* random graphs that perpetually evolve. We run extensive simulations of both models to grasp the "stationary" structural properties of the dynamic random graphs and to measure the time it takes a message generated from one node to reach all (or almost all) the nodes.

In the first model, *edge-dynamic RAES (E-RAES)*, we add an "edge-evolution" parameter $p \in [0, 1]$ with the following role: At every round, each accepted edge disappears with probability p. A detailed description of this model is presented in Section 2.1. Since the set of nodes of the graph is fixed while the set of edges evolve in discrete rounds, the dynamic random graph is a sequence $\{G_t = (V, E_t) : t \in \mathbb{N}\}$ where the distribution of the edges at round t only depends on the set of edges at round $t - 1$. In order to empirically measure when the dynamic random graph can be considered stable, we compute the sequence of spectral gaps γ_t of the transition matrices P_t of the snapshots G_t of the dynamic graph and we consider that the dynamic graph is in a stable regime when γ_t remains in a sufficiently small interval for a sufficiently large window of consecutive rounds. The spectral gap of the transition matrix is also a measure of how "well-connected" a graph is and the results of the simulations show that the model generates, on average, sequences of graphs that are well-connected even with large values for the edge disappearing rate p. Indeed, even when a large fraction of edges disappear at any round, one single step of the RAES procedure is typically sufficient to rebuild a well-connected graph.

In the second model, *vertex-dynamic RAES (V-RAES)*, we add two "node-evolution" parameters, $\lambda \in \mathbb{R}^+$ and $q \in [0, 1]$, with the following roles: At every round t, $N_\lambda(t)$ new nodes enter the network, where $N_\lambda(t)$ is a Poisson random variable with rate λ, and each node leaves the network with probability q, independently of the other nodes. As soon as a new node joins the network, it starts requesting links to the nodes already in the network; one round later, i.e., when the presence of the new node has been revealed to the network, the node also starts receiving incoming link requests from other nodes; when a node leaves the network, all its incident links disappear. A detailed description of this model is given in Section 2.2. In the V-RAES model the network evolution is a sequence of random graphs $\{G_t = (V_t, E_t) : t \in \mathbb{N}\}$ in which both the set of nodes and the set of edges are random sets at any round. It is easy to see that the expected number of nodes in the graph converges to λ/q, if we consider it *before* the node-leaving step, and to $\lambda(1 - q)/q$ if we consider it *after* the node-leaving step.

The dissemination protocol in Bitcoin-core is a gossip-based flooding: When a node receives a valid transaction, it announces it to all its neighbors (see, e.g., https://en.bitcoin.it/wiki/Network#Standard_relaying). As far as we know there are recent proposals to modify the dissemination mechanism aiming at improving network bandwidth usage [26] or limiting de-anonymization attacks [18], but to the best of our knowledge they have not been implemented in Bitcoin-core so far (see, e.g., https://github.com/bitcoin/bips/blob/master/bip-0330.mediawiki). In both our models, E-RAES and V-RAES, we simulate the flooding process and we measure the *flooding time*, i.e., how long it takes a message starting at a random node to reach all (or almost all) the nodes of the graph.

For the E-RAES model, the results of the simulations show that the flooding time is short (i.e., compatible with a logarithmic growth, as a function of the number of nodes), for every value of the edge-disappearance rate p. For the V-RAES model, the results of the simulations show that, as long as the fraction of nodes that leave the network at any round is not too large, e.g., if it stays below 70%, a message starting at a random node typically quickly reaches nearly all of the nodes.

We also simulate a combination of the two models, in which nodes join and leave the network as in the V-RAES and edges can be faulty as in the E-RAES. In this paper we present the set of results obtained by simulating the models with only a few representative ranges for parameters d and c that determine the neighborhood size of the nodes: the smallest possible values for which the underlying graph turns out well-connected and the default values used in the main Bitcoin implementation. However, we remark that simulations with different values of d and c exhibit similar qualitative behavior. The interested reader can find the libraries developed to run the simulations on the github repository of the first author (https://github.com/Antonio-Cruciani/dynamic-random-graph-generator).

Notice that the topology of the evolving random graphs generated according to our models is almost surely quite far from the evolving topology of the real Bitcoin network, since each node of the real network can autonomously decide how many neighbors it wants to have and how to try to connect to them, and typically nodes choose different strategies based on their different needs. However, the topology of the evolving random graphs generated according to our models is probably close to the topology that the Bitcoin network would have if all full-nodes used the Bitcoin-core implementation with the default parameters. The study of our models thus allows us to give evidence of the long-term stability of the network generation process implemented in Bitcoin-core. This, in turn, gives an indication about the long-term stability of a large part of the real network without revealing its topology.

As noted in [28], most design decisions implemented at the network layer of permissionless blockchains imply some tradeoffs that typically are not yet well-understood. In this respect, the results of our simulations suggest that the default values used in the main Bitcoin implementation that determine the size of the neighborhood of a full-node could be safely reduced by most of the full-nodes to save network bandwidth without compromising the stability of the network.

1.2 Related work

The topology of the Bitcoin network is hidden by the network formation protocol. However several approaches in the last decade proved effective in revealing some portion of the network. Miller et al. [22] developed a set of tools and an infrastructure to discover the public Bitcoin network. Their approach was subsequently made ineffective by an update in the Bitcoin protocol. Neudecker et al. [27] proposed a timing analysis that is able to infer the network topology with a sufficient degree of precision. Delgado-Segura et al. [13] proposed a new approach to reconstruct the network structure and tested it on the Bitcoin *testnet* network revealing a network with 733 nodes and 6090 edges, with an average degree of 16.6 and with most of the nodes having between 7 and 14 neighbors. As far as we know it has never been tested on the Bitcoin *main* network.

Peer-to-Peer (P2P) networks received a lot of attention in the last twenty years and several (static and dynamic) network models have been proposed so far. A random network model for unstructured P2P networks was introduced and analyzed by Panduragan et al. [30]. Their model was inspired by the Gnutella P2P network and is based on the existence of a *host server* that maintains a *cache* of constant size with addresses of nodes accepting connections that can be reached at any time by other nodes. In [5] the authors introduced a class of dynamic graphs called *Dynamic Networks with Churn* (in short, *DNC*) where both node insertion/deletion and edge evolution are considered. The authors assume that the dynamic graph consists of a sequence of *d-regular expander graphs*; for the purpose of that paper, such an assumption is justified by the results in [4], where the authors presented a distributed protocol that guarantees the maintenance of a bounded degree topology that, with high probability, contains an expander subgraph whose set of vertices has size $n - o(n)$, where n is the "stable" network size. In [10] the authors defined two churn processes: in the first one, at every round a new node is added to the network while no node leaves it; in the second one, the size of the vertex set is n and when a new node joins the network the oldest node leaves it. The authors designed a protocol where each node u starts $c \cdot m$ independent random walks (containing the ID-label of the node) until they are picked up by new nodes joining the network, that connects to the peers that contributed to the tokens. The resultant dynamic topology is shown to keep diameter $O(\log n)$ and to be fault-tolerant against adversarial deletion of both edges and vertices. The tokens in the graphs must be circulating at each time step in order to ensure that they are well-mixed; this implies that the rate at which new nodes can join the system is limited, as they must wait while the existing tokens mix before they can use them. Bagchi et al. [6] studied the number of adversarial and random faults that an expander graph can tolerate while preserving approximately the same expansion factor and a linear number of nodes. Becchetti et al. [8] introduced and analyzed the *RAES* network formation model, in which after a logarithmic number of rounds the network evolution terminates in a state in which every node has a specified out-degree and in-degree upper bounded by a constant. In a recent work Becchetti et al. [9] introduced and studied a similar model in which nodes can also join and leave the network, but the in-degree of the nodes is not upper bounded by a constant.

Several well known problems have been studied in the context of dynamic networks: (byzantine) agreement, search and storage, (byzantine) leader election, expander maintenance, information spreading, membership management (we refer the reader to [3] for a survey). For information spreading, early works considered gossip-based broadcast algorithms (see, e.g., [16]). However, for privacy oriented P2P networks, such as the Bitcoin P2P network, some of these algorithms have been shown to expose the network to privacy vulnerabilities [19] and motivated the design of more sophisticated information spreading algorithms with low overhead as well as strong resistance to de-anonymization attacks [18, 26, 33]. Being able to randomly select other peers as new neighbors to maintain a random-graph like overall structure (low diameter, bounded degree, etc.) is another critical issue in such networks that has been extensively studied (see, e.g., [20, 31, 34]).

1.3 Roadmap

In Section 2 we give the formal description of the two dynamic random graph models and of the parameters that we are measuring with the simulations. In Sections 3 and 4 we describe the results obtained from the simulations of the two models and in Section 5 we consider a combination of the two models. Finally, in Section 6 we draw some conclusions.

2 THE MODELS AND THE PROBLEM

A *dynamic graph* $\mathcal{G}$ is a sequence of graphs $\mathcal{G} = \{G_t = (V_t, E_t) : t \in \mathbb{N}\}$ where the sets of nodes and edges can change at any discrete round. If they change randomly, we call the corresponding random process a *dynamic random graph*. In this section we introduce two dynamic random graph models, that we call *Edge-dynamic RAES (E-RAES)* and *Vertex-dynamic RAES (V-RAES)*, that extend the RAES model introduced in [8].

2.1 Edge-dynamic RAES (E-RAES)

The E-RAES model is defined by four parameters, n, d, c, and p, where $n \in \mathbb{N}$ is the number of nodes, $d \in \mathbb{N}$ is the *minimum target degree*, $c \cdot d$ with $c \geqslant 1$ is the *maximum acceptable degree*, and $p \in [0, 1]$ is the *edge-failure probability*. The set of n nodes is fixed, while the set of edges evolves, at each round, in three steps. In the first step, each node with less than d neighbors connects with randomly chosen nodes in order to reach its minimum target degree; in the second step, each node with more than $c \cdot d$ neighbors, disconnects from randomly chosen neighbors in order to remain within its maximum acceptable degree; in the third step, each edge disappears with probability p, independently of the other edges.

Starting from an arbitrary initial graph $G_0 = (V, E_0)$.
At each round $t \in \mathbb{N}$:

Step 1: For each node $u \in V$, let N_u^1 be the set of neighbors of u at the beginning of Step 1. If $|N_u^1| < d$ then u samples $d - |N_u^1|$ nodes from the set $V \setminus N_u^1$, independently and u.a.r. with replacement, and connects to them.

Step 2: For each node $u \in V$, let N_u^2 be the set of neighbors of u at the beginning of Step 2. If $|N_u^2| > c \cdot d$ then u samples $|N_u^2| - (c \cdot d)$ neighbors from the set N_u^2, independently and u.a.r. with replacement, and disconnects from them.

Step 3: Each edge $\{u, v\}$ currently in the graph disappears with probability p, independently of the other edges.

The E-RAES model defines a Markov chain with the set of all graphs with n nodes as state space. It is not difficult to see that the chain is aperiodic and that the empty graph is a *recurrent* state (see, e.g., Chapter 1.5 in [29] for some background). Hence, if we consider the recurrent class containing the empty graph, the Markov chain defined by the E-RAES model starting at the empty graph will converge to a stationary distribution π. From a theoretical point of view, it would be interesting to analyze the expansion properties of the stationary random graph (i.e., a random graph sampled according to the stationary distribution) and to estimate the *mixing time* of the Markov chain, i.e., the time it takes the distribution of the chain starting at the empty graph to get close to the stationary distribution. However, a theoretical analysis of the mixing time appears quite challenging due to the complexity of the Markov chain: for example, in Appendix A we observe that the chain is not *reversible*, thus it not possible to apply the large body of tools developed for the analysis of reversible chains (see, e.g., [1]). In Section 3 we propose an empirical convergence criterion, we present the results on the expansion properties of the snapshots of the dynamic graph obtained by simulating the E-RAES, and the results on the time it takes a message starting at a random node to reach all the nodes.

2.2 Vertex-dynamic RAES (V-RAES)

The V-RAES model is defined by four parameters, λ, d, c, and q, where $\lambda > 0$ is the *arrival rate* of new nodes, d and $c \cdot d$ are the *minimum target degree* and the *maximum acceptable degree* as described in the E-RAES model, and $q \in [0, 1]$ is the *node-leaving probability*. At each round t the graph evolves in four steps. In step *zero* $N_\lambda(t)$ new nodes join the graph, where $N_\lambda(t)$ is a Poisson random variable with rate λ. In step *one*, each node with less than d neighbors (hence, including the $N_\lambda(t)$ newly arrived ones) connects with randomly chosen nodes among those that are in the graph at the current round and were also present in the graph at the previous round (hence, excluding the $N_\lambda(t)$ newly-arrived nodes). In step *two*, each node with more than $c \cdot d$ neighbors, disconnects from randomly chosen neighbors in order to remain within its maximum acceptable degree. In step *three*, each node u disappears with probability q, independently of the other nodes (all edges incident to u disappear as well).

Starting from an arbitrary initial graph $G_0 = (V_0, E_0)$.
At each round $t \in \mathbb{N}$:

- **Step 0:** $N_\lambda(t)$ new nodes join the graph, where $N_\lambda(t)$ is a Poisson random variable with rate λ.
- **Step 1:** For each node u, let N_u^1 be the set of neighbors of u at the beginning of Step 1. If $|N_u^1| < d$ then u samples $d - |N_u^1|$ nodes from the set $(V_t \setminus N_\lambda(t)) \setminus N_u^1$, independently and u.a.r. with replacement, and connects to them.
- **Step 2:** For each node u, let N_u^2 be the set of neighbors of u at the beginning of Step 2. If $|N_u^2| > c \cdot d$ then u samples $|N_u^2| - (c \cdot d)$ neighbors from the set N_u^2, independently and u.a.r. with replacement, and disconnects from them.
- **Step 3:** Each node u disappears with probability q, independently of the other nodes, together with its incident edges.

The size of the vertex set V_t in the V-RAES model converges to $\lambda(1-q)/q$, if measured at the end of the round, and it converges to λ/q if measured at the end of step *two* of the round, i.e., before the node-leaving step. Indeed, consider the following informal argument: Let us name f_t the expected number of nodes at round t, then $f_t = (f_{t-1} + \lambda)(1-q)$, if computed at the end of the round, since in expectation λ new nodes join the network at round t and each node in the graph remains in the network with probability $(1-q)$. Solving the recurrence with initial condition $f_0 = 0$ gives $f_t = \lambda \sum_{i=1}^{t} (1-q)^i = \lambda \left(1 - q - (1-q)^{t+1}\right)/q$, that converges to $\lambda(1-q)/q$ for t that goes to infinity. More formally, the size of the vertex set is actually a Markovian queue $M\backslash G\backslash\infty$ and it converges to a Poisson random variable of rate λ/q (see, e.g., [30]). In Section 4 we present the results of the simulations of the V-RAES model.

2.3 Preliminaries

Spectral gap. Let $G = (V, E)$ be an undirected graph with no self-loops. The *transition* matrix of a simple random walk on G (we will refer to it as the transition matrix of G) is the $|V| \times |V|$ matrix $P = D^{-1}A$, where A is the adjacency matrix of G and D is the diagonal matrix whose entries are the degrees of the nodes (for each node $u \in V$, $D(u,u)$ is the degree of u in G). It is well-known that P is *reversible*, all its eigenvalues are real and they belong to the interval $[-1, 1]$ and the largest eigenvalue is $\lambda_1 = 1$. Moreover, the second largest eigenvalue $\lambda_2 < 1$ if and only if G is connected. In this case the *spectral gap* $\gamma = 1 - \lambda_2$ is a measure of how quickly the random walk converges to its stationary distribution (the largest the spectral gap the fastest the convergence rate). In the following paragraph we recall that the spectral gap is also a measure of how "well-connected" the underlying graph G is.

Expanders and spectral gaps. A graph $G = (V, E)$ with $|V| = n$ nodes is a $(1+\delta)$-*vertex expander*, for some $\delta > 0$, if for every set S of size at most $n/2$, the neighborhood $N(S) = \{v \in V : \{u, v\} \in E \text{ for some } u \in S\}$ has size at least $(1+\delta)|S|$. It is known that, for a regular graph G, if we define $\gamma = 1 - \max\{\lambda_2, |\lambda_n|\}$, where λ_2 and λ_n are respectively the second-largest eigenvalue and the smallest eigenvalue of the transition matrix P, then the graph G is a $(1+\gamma)$-vertex expander (see e.g. Chapter 4 in [32]). Thus, larger values of the spectral gap γ of the transition matrix P correspond to better expansion of the underlying graph G.

Let $\mathcal{G} = \{G_t = (V_t, E_t) : t \in \mathbb{N}\}$ be a dynamic (random) graph. For the purpose of this paper, we measure how well-connected are the snapshots G_t of the dynamic graph by computing the spectral gaps of their transition matrices.

Flooding. To measure the time it takes a message sent by a node to reach all (or a large fraction of) nodes we use the following *flooding process*. Let $\mathcal{G} = \{G_t = (V_t, E_t) : t \in \mathbb{N}\}$ be a dynamic random graph. The flooding process over $\mathcal{G}$ starting at round t_0 from the initiator $u_0 \in V_{t_0}$ is the sequence of (random) sets of nodes $\{I_t : t \in \mathbb{N}\}$ such that: $I_t = \emptyset$ for $t < t_0$; $I_{t_0} = \{u_0\}$; and for $t > t_0$

$$I_t = (I_{t-1} \cup N(I_{t-1})) \cap V_t$$

where $N(I_{t-1})$ is the set of nodes in $V_{t-1} \setminus I_{t-1}$ that in graph G_{t-1} have at least one neighbor in I_{t-1}

$$N(I_{t-1}) = \{v \in V_{t-1} \setminus I_{t-1} : \{u, v\} \in E_{t-1} \text{ for some } u \in I_{t-1}\}$$

We say that I_t is the subset of *informed* nodes at round t. If at some round t all nodes currently in the network are informed, i.e. $I_t = V_t$, we say that the flooding is *complete*. The *flooding time* is the number of rounds between t_0 and the first round t such that $I_t = V_t$.

3 E-RAES SIMULATIONS

In this section, we present the results of the simulations of the E-RAES model. On the one hand we are interested in the structural properties of the snapshots of the evolving graphs, on the other hand we want to measure how long it takes a message starting at one node to reach all the others. To evaluate the structural properties, we use the *spectral gap* of the transition matrix; to evaluate the speed of information spreading we use the *flooding time* (see Section 2.3).

In Section 3.1 we define an empirical converge criterion that we will use to decide the starting round for the simulations computing the average spectral gap of the snapshots of the evolving graph and the average duration of the flooding process. In Section 3.2 we present the results of the simulations for the spectral gap and in Section 3.3 those for the flooding process.

We present only the results for some representative parameters d and c: the minimum target degree $d = 4$ is small enough to guarantee that the resulting snapshots of the evolving graph are quite "sparse"; the value $c = 1.5$ makes the nodes quite "inflexible" about their target degree (each node only accepts to have degree 4, 5, or 6). Despite these strict requirements about the graph structure, our simulations show that the random process quickly stabilizes on a stationary regime, where the snapshots of the graph are often very good expanders, even for large values of the edge-failure probability p. We remark that results qualitatively very similar to those presented for $d = 4$ and $c = 1.5$ appear for different values of d and c.

3.1 Convergence criterion

We want to study how fast the information spreads from a node to all the other nodes when the network evolution is *stationary*. In order to decide the starting round for the flooding process, we need a criterion to establish when the network evolution reaches stationarity. In principle, it would be possible to give theoretical bounds on the number of rounds needed to reach stationarity by analyzing the *mixing time* of the Markov chain induced by the E-RAES model; however, as we mentioned in Section 2, the analysis of such a Markov chain appears far from easy. For the purpose of this paper, we use a heuristic criterion based on the stabilization of the spectral gap. We set an $\varepsilon > 0$ and we declare that the graph stabilizes when the spectral gap remains in a range of width 2ε for $\log n$ consecutive rounds. More formally, at the generic round $t \geqslant \log n$, if all spectral gaps $\gamma_{t-\log n}, \gamma_{(t-\log n)+1}, \ldots, \gamma_t$ differ from γ_t for at most ε then we declare that the dynamic random graph mixed. The choice of ε is dynamically computed by the following rule: we simulate a long-run of the evolving graph for 100 rounds and we set ε as the mean absolute deviation [17] of the non-zero values.

Figure 1 shows a representative sample of the evolution of the spectral gap during the first rounds of the E-RAES model with $n = 2^{15}$ nodes, $d = 4$, $c = 1.5$, and edge-failure probability $p = 0.1$, starting from the empty graph. The spectral gap of the snapshots of the evolving graph is computed before the edge failure step. The picture shows that, after about 15 rounds, the spectral gap stabilizes with very small oscillations, from round to round in each execution and with little difference from one execution to another.

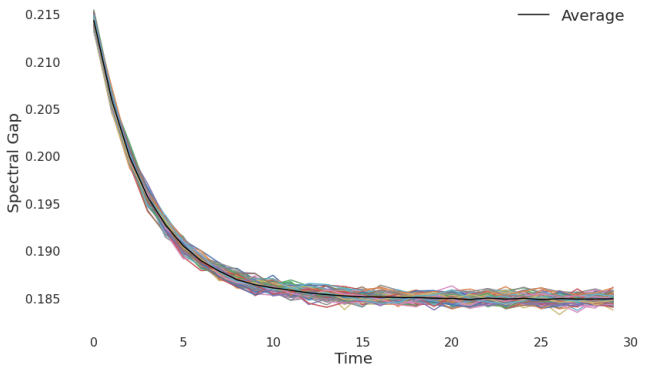


Figure 1: The initial evolution of the spectral gap and its stabilization for the dynamic graph with 2^{15} nodes, $d = 4$, $c = 1.5$, and starting from the empty graph. The spectral gap at each round is computed before the edge-failure step. Each line in the picture plots one out of one hundred executions. The bold black line plots the average of the spectral gaps computed at each round, over all the executions.

3.2 Average spectral gap in the long run

To measure the expansion properties of the typical snapshot of the evolving graph, we simulate the E-RAES model and compute the average of the spectral gaps of the snapshots. The table and the plot in Figure 2 show the results of the simulations for different values for the number of nodes n and edge-failure probability p. Each number in the table is the average over 100 runs of 100 rounds each. We computed the spectral gap both before and after the edge-failure step.

For small values of p, e.g., $p = 0.1$, the first column of the table in Figure 2 shows that the snapshots are on average connected (the spectral gap is non-zero) even *after* the edge failure step. Although the differences between the spectral gaps computed before and after the edge-failure step, that increases with the number of nodes, indicates that after the edge-failure step the resulting graph tend to become a much weaker expander, even when only 10% of the edges disappear on average.

For larger values of p the snapshots of the graph after the edge faults turn out to be mostly disconnected (spectral gap equals to zero), however the spectral gap computed *before* the edge-failure step indicates that every time the graph becomes disconnected, just one more step of the RAES process is sufficient to rebuild a connected graph with good expansion properties.

In Figure 2 it is also interesting to notice the unimodal trend of the spectral gap as a function of the edge-failure probability: it decreases for p from 0 to 0.1 and it increases for $p > 0.1$. This indicates that the snapshots of *highly-dynamic* graphs, in which nodes are forced to frequently regenerate their neighborhoods, are

Average spectral gap								
Nodes \ p		0.0	0.1	0.3	0.5	0.7	0.9	1.0
1024	B	0.345	0.19	0.208	0.234	0.269	0.317	0.345
	A	0.345	0.157	0.0	0.0	0.0	0.0	0.0
2048	B	0.342	0.189	0.206	0.232	0.268	0.315	0.342
	A	0.342	0.155	0.0	0.0	0.0	0.0	0.0
4096	B	0.341	0.187	0.204	0.231	0.267	0.314	0.341
	A	0.341	0.144	0.0	0.0	0.0	0.0	0.0
8192	B	0.341	0.186	0.204	0.23	0.266	0.313	0.341
	A	0.341	0.09	0.0	0.0	0.0	0.0	0.0
16384	B	0.34	0.186	0.203	0.23	0.265	0.313	0.34
	A	0.34	0.053	0.0	0.0	0.0	0.0	0.0
32768	B	0.34	0.185	0.203	0.23	0.265	0.313	0.34
	A	0.34	0.006	0.0	0.0	0.0	0.0	0.0

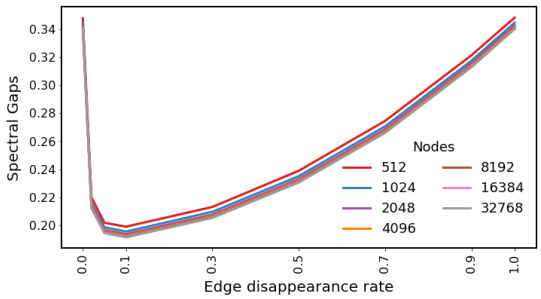


Figure 2: Average spectral gap for E-RAES of 100 runs of 100 rounds each, for $d = 4$, $c = 1.5$, and increasing values for number of nodes n and edge-failure probability p. The spectral gap is computed before (B) and after (A) the edge-failure step.

better expanders than the snapshots of less dynamic graphs, in which the connections between nodes are more stable.

3.3 Flooding Time Analysis

We here present the results of the simulations of the flooding process (see Section 2.3) on the E-RAES model (see Section 2.1). The simulation proceeds as follows: Starting from the empty graph, we wait for the first round t_0 in which the dynamic graph $\{G_t = (V, E_t) : t \in \mathbb{N}\}$ meets the criterion defined in Section 3.1, then we pick a node $u_0 \in V$ uniformly at random, we simulate the flooding process with initiator u_0, and we measure the number of rounds until the flooding is complete.

Figure 3 shows the results of the simulations obtained by setting in the E-RAES model the parameters $d = 4$, $c = 1.5$ and different values for number of nodes n and edge-failure probability p. Each point in the plot is the average, over 100 runs, of the number of rounds required by the flooding process to complete.

The picture quite clearly highlights that the flooding time, as a function of the number of nodes, is compatible with a logarithmic growth, for every value of the edge-failure probability p. The value of p seems to determine the multiplicative constant of the logarithm. We remark that in the simulations the message-passing step of the flooding process is scheduled *after* the edge-failure step of the E-RAES model, i.e., when for values of p larger than 0.1 the snapshot of the graph is typically disconnected. Thus it is interesting to notice that, even for large value of p, e.g. when 90% of the edges disappear at each round, the time required to get all nodes informed is quite short. These results suggest that a new message rapidly "floods"

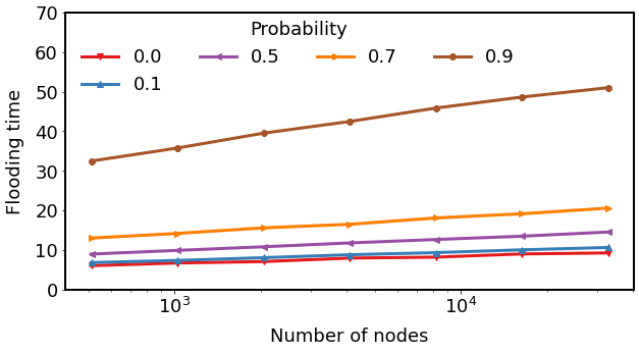


Figure 3: Semi-log-plot of the average flooding time of $\mathcal{G}(n, 4, 1.5, p)$ with $2^9 \leqslant n \leqslant 2^{15}$, $p \leqslant 0.9$.

the dynamic network even if every snapshot of the dynamic graph is completely sparse and disconnected.

4 V-RAES SIMULATIONS

In this section, we present the results of the simulations of the V-RAES model. As for the structural properties, we recall that (see Section 2.2) when new nodes arrive they can connect to nodes currently in the graph, but they cannot be asked for connections from other nodes. At each round thus the snapshot of the evolving graph is formed by a *core*, i.e., the nodes that were present in the graph in the previous round as well, and a *periphery*, i.e., the nodes arrived in the current round, that are connected only to nodes in the core. Hence, the snapshots of the evolving graph are not good expanders. Nevertheless, our simulations show that the "flooding time" in the V-RAES model is fast.

The definition of "flooding time" as described in Section 2.3 needs to be appropriately adapted in the V-RAES model to take into account the fact that new nodes join the network at any round and thus the process could (and typically does) never reach a state in which all nodes currently in the network are informed.

As we did for the E-RAES model, we want to start simulating the flooding process when the network evolution is "stationary". In Section 4.1 we thus define an heuristic convergence criterion and in Section 4.2 we present the results on the flooding process. We remark that we here present the results only for some representative parameters d and c, other choices for those parameters produce similar results.

4.1 Convergence criterion

Since in the V-RAES model new nodes arrive at any round with rate λ and each node leaves the network with probability q, the stationary expected number of nodes in the network is λ/q and the actual number of nodes is concentrated around its expected value. We thus consider the network evolution for the V-RAES model to have reached a stationary regime when the number of nodes in the network is close to λ/q.

Figure 4 shows the evolution of the number of nodes in the graph during the first rounds of the V-RAES, with parameters $d = 4$ and $c = 1.5$, starting from the empty graph. All plots in the picture

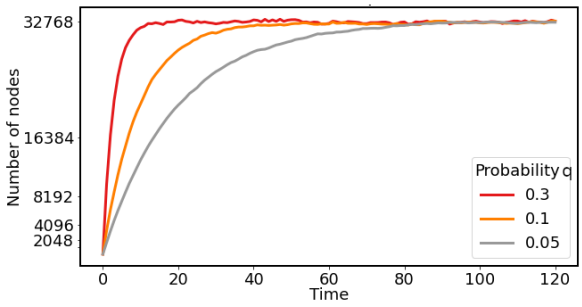


Figure 4: The evolution of the number of nodes for some sample runs with $d = 4$, $c = 1.5$, and $\lambda/q = 2^{15}$ and $q = 0.05, 0.1, 0.3$

refer to the ratio $\lambda/q = 2^{15}$, each plot with a different value for the node-leaving probability q (and with the corresponding value for λ). The number of nodes is considered *before* the node-leaving step.

4.2 Flooding Time Analysis

Since nodes join and leave the network at any round, in the V-RAES model a message sent from an initiator node might not reach neither all the nodes in the graph nor a large fraction of them. For example, if the initiator node and all its neighbors leave the network one round after the message departure, then the message will never reach any of the other nodes. In order to measure the speed of information spreading in the V-RAES model, we thus run the simulations as follows: Starting from the empty graph, we wait for the first round t_0 in which the dynamic graph $\{G_t = (V_t, E_t) : t \in \mathbb{N}\}$ meets the criterion defined in Section 4.1, then we pick a node $u_0 \in V_{t_0}$ uniformly at random, we simulate the flooding process (see Section 2) with initiator u_0, and we monitor the fraction of informed nodes $\alpha_t := |I_t|/|V_t|$ at each round.

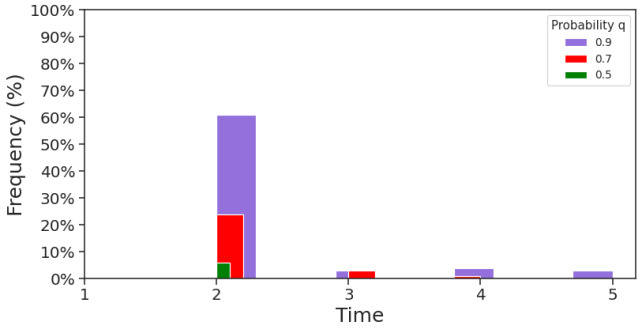


Figure 5: Percentage of the failed flooding executions. Each bar of the histogram indicates the number of times in which all the informed nodes left the network at the corresponding round. The ratio λ/q is fixed to 2^{15}.

Fig. 5 shows the fraction of simulations in which, at some round after t_0, all the informed nodes disappeared simultaneously, thus leaving the network without any informed node. The first observation emerging from the histograms is that all the times this event happened, it was within five rounds from t_0.

For $q = 0.9$, i.e. when about 90% of the nodes disappear at every round, in about 60% of the simulations all the informed nodes left at the second round of the flooding process. On the other hand, for $q = 0.5$, i.e when about half of the nodes disappear at every round, the fraction of times in which the message of the initiator node u_0 fails to spread in the network is very small.

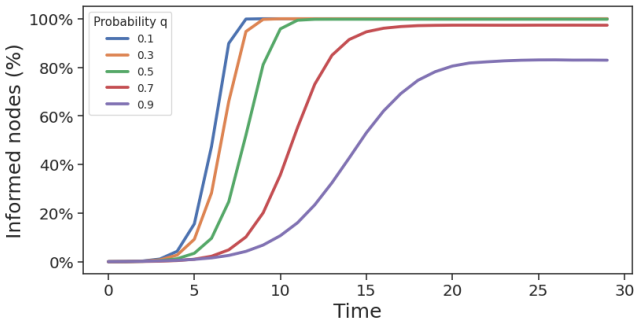


Figure 6: Average over 100 runs of the evolution of the fraction of informed nodes α_t, at each time step. In the plots the ratio λ/q is fixed to 2^{15}.

In Fig. 6 we plot the evolution of the fraction α_t of informed nodes, for all the simulations in which the message of the initiator node u_0 does spread in the network. The plots show that, when the set of informed nodes do not disappear during the very first rounds, the fraction of informed nodes quickly stabilizes over precise values that depend on the node-leaving probability q: for $q \leqslant 0.7$ the number of informed nodes reaches a stationary phase in which almost all the nodes in the network are informed; even for larger values of the node-leaving probability, e.g., when $q = 0.9$, in all simulations in which the informed nodes do not simultaneously disappear within the first five rounds, the fraction of informed nodes stabilizes around 80%.

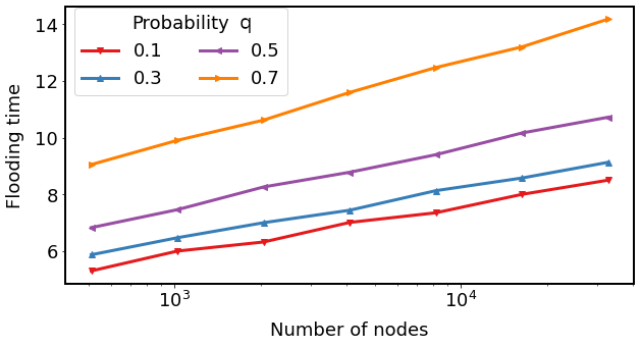


Figure 7: Semi-log-plot of the average flooding time trend of $\mathcal{G}(\lambda, q, 4, 1.5)$ with $2^9 \leqslant \lambda/q \leqslant 2^{15}$

As a measure of *flooding time* in the V-RAES model, we thus can consider the number of rounds required to reach the stable value α_t, as it is determined by the node-leaving probability q. For

example, in Figure 7 we plot the number of rounds required by the flooding process to reach a fraction α_t of informed nodes of at least 90%, for all the values of the node-failure probability q such that the fraction of informed nodes stabilizes above 90%. The picture clearly highlights that such number of rounds is compatible with a logarithmic growth, as a function of λ/q.

5 EV-RAES AND THE PARAMETERS OF THE REAL BITCOIN NETWORKS

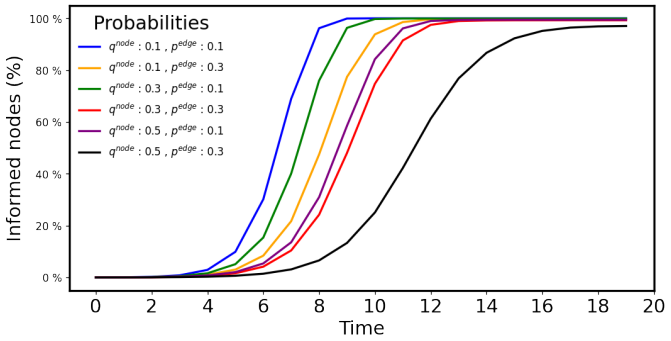
In this section we present a combination of the E-RAES and V-RAES models, that we call EV-RAES model, in which nodes join and leave the network as in the V-RAES and edges can be faulty as in the E-RAES. We simulate the flooding process on such a model first using the same values for d and c that we used in the V-RAES section and then using the default values of the main implementation of Bitcoin.

Starting from an arbitrary initial graph $G_0 = (V_0, E_0)$.
At each round $t \in \mathbb{N}$:

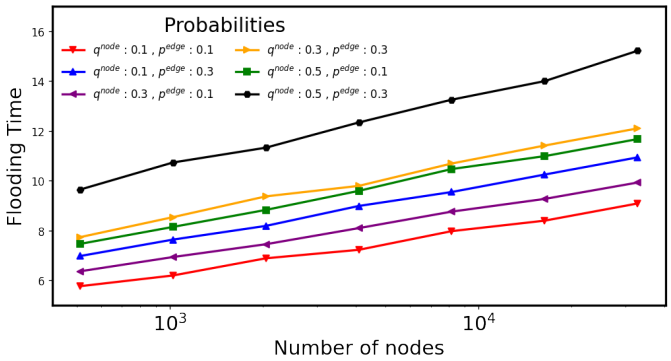
Step 0: $N_\lambda(t)$ new nodes join the graph, where $N_\lambda(t)$ is a Poisson random variable with rate λ.
Step 1: For each node u, let N_u^1 be the set of neighbors of u at the beginning of Step 1. If $|N_u^1| < d$ then u samples $d - |N_u^1|$ nodes from the set $(V_t \setminus N_\lambda(t)) \setminus N_u^1$, independently and u.a.r. with replacement, and connects to them.
Step 2: For each node u, let N_u^2 be the set of neighbors of u at the beginning of Step 2. If $|N_u^2| > c \cdot d$ then u samples $|N_u^2| - (c \cdot d)$ neighbors from the set N_u^2, independently and u.a.r. with replacement, and disconnects from them.
Step 3: Each edge $\{u, v\}$ currently in the graph disappears with probability p, independently of the other edges.
Step 4: Each node u disappears with probability q, independently of the other nodes, together with its incident edges.

We first observe the impact of the edge failures on the fraction of nodes reached in the flooding procedure and on the flooding time.

Figure 8 shows the results of the simulations on the fraction of informed nodes and the flooding time for the EV-RAES model, with the same values for d and c used in Section 4 in the simulations of the V-RAES. A comparison of Figure 8 with Figures 6 and 7 highlights that the impact of the edge failures on the final fraction of informed nodes and on the flooding time is quite negligible. For example, for node-leaving probability q up to 0.3, even with edge-disappearance rate $p = 0.3$ all nodes receive the message within the same amount of rounds needed when the edges do not disappear. For larger values of q, e.g. $q = 0.5$, the fraction of nodes that receive the message turns out smaller for $p = 0.3$ with respect to the case in which edges do not disappear. Notice that such large values for q and p are only useful to test the limits of model, since they generate dynamic networks in which 50% of the nodes join and leave the network and 30% of the edges disappear at every round. In any realistic scenario, the fraction of nodes that join and leave the network at any round and the number of connections that fail is likely to be much smaller. In those scenarios, our simulations indicate that all nodes receive the message, within a number of rounds that is compatible with a logarithmic growth as a function of the number of nodes in the network.



(a) Evolution of the fraction of informed nodes α_t. The ratio λ/q is fixed to 2^{15}.



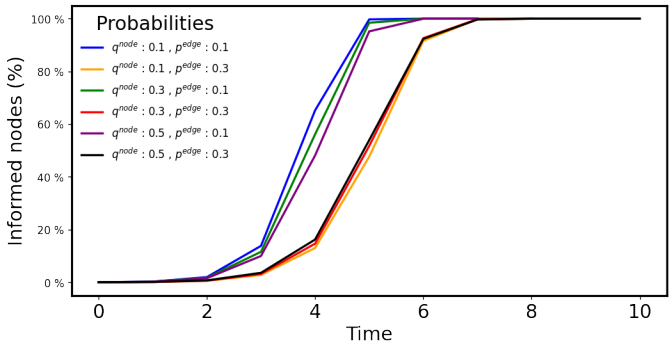
(b) Semi-log-plot of the average flooding time of the EV-RAES with $2^9 \leqslant \lambda/q \leqslant 2^{15}$, node disappearance rate $q = 0.1, 0.3, 0.5$, and edge disappearance rate $p = 0.1, 0.3$.

Figure 8: EV-RAES with $d = 4$ and $c = 1.5$, fraction of informed nodes and flooding time

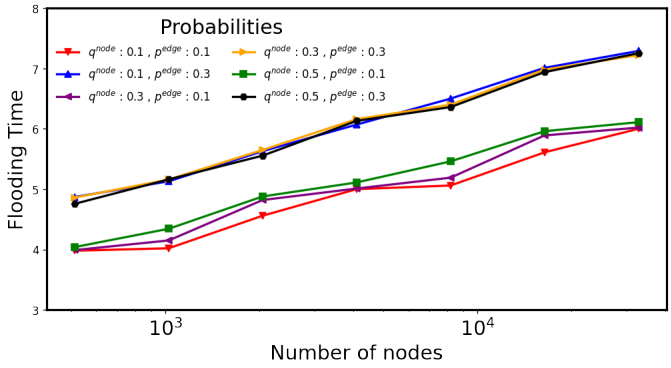
5.1 The degree of the full-nodes and network traffic

In the Bitcoin network currently there are approximately $14 \cdot 10^3$ reachable nodes (see https://bitnodes.io/ for periodic crawls of the Bitcoin P2P Network) and several hidden ones [35]. In the default configuration of the main implementation, lower and upper bounds on the number of connections that a full-node can have are set to 8 and 128, respectively. We thus also simulated the EV-RAES model with values for parameters d and c corresponding to the above values of the real Bitcoin P2P Network: namely, $\lambda/q = 2^{14}$ and $d = 8$ and $c = 15.625$.

On the one hand, a comparison of Figures 9 and 8 shows that the advantage of having such a large number of neighbors, i.e. up to 128 in Figure 9 as opposed to up to 6 in Figure 8, is limited in terms of fraction of informed nodes and flooding time. On the other hand, the number of neighbors of a full-node is directly proportional to the amount of network traffic going through the node. Indeed, "*it's common for full nodes on high-speed connections to use* 200 *gigabytes upload or more a month*" (see https://bitcoin.org/en/full-node#minimum-requirements). In order to measure the impact of the number of neighbors on the network traffic, we installed a



(a) Evolution of the fraction of informed nodes α_t. The ratio λ/q is fixed to 2^{15}.



(b) Semi-log-plot of the average flooding time of the EV-RAES with $2^9 \leqslant \lambda/q \leqslant 2^{15}$, node disappearance rate $q = 0.1, 0.3, 0.5$, and edge disappearance rate $p = 0.1, 0.3$.

Figure 9: EV-RAES with $d = 8$ and $c = 15.625$: Fraction of informed nodes and flooding time

Bitcoin-core full-node, we reduced the default number of connections of the node from 125 to 25 and, after the completion of the initial block download, we monitored the upload network traffic observing an average upload traffic between 400 and 500 MB per day, hence less than 15 GB per month.

6 CONCLUSIONS

In this paper we introduced two models of dynamic random graphs inspired by the network formation protocol of the Bitcoin P2P network. We simulated the models to evaluate the structural properties of the snapshots of the dynamic graphs and to measure the time it takes a message starting at a random node to reach all, or almost all, the nodes. The results of our simulations show that the network structure generated by the E-RAES, by the V-RAES, and by a combination of the two models is globally very robust, in the sense that the network can quickly rebuild itself after node and edge failures. Moreover, the simulations of the flooding procedure show that the information spreading in the two models is fast and reliable, for the E-RAES essentially at any edge-failure rate, and for the V-RAES up to a node-failure rate as high as 70%. The outcomes of the simulations on the combined model EV-RAES are similar to those obtained in the V-RAES model for a large range of the parameters.

Since the degree of a full-node in the Bitcoin network is directly correlated to the amount of traffic going through the node, our results suggest that it is quite safe, for full-nodes of the Bitcoin network that need to reduce the bandwidth usage, to change the default value of the maximum number of connections from 125 to much smaller values. On the one hand this significantly reduces the upload network traffic and, on the other hand, our simulations suggest that it does not compromise the overall stability and reliability of the network.

REFERENCES

[1] David Aldous and James Allen Fill. 2002. Reversible Markov Chains and Random Walks on Graphs. Unfinished monograph, recompiled 2014, available at http://www.stat.berkeley.edu/$\sim$aldous/RWG/book.html.

[2] Andreas M. Antonopoulos. 2017. *Mastering Bitcoin: Programming the open blockchain.* "O'Reilly Media, Inc.".

[3] John Augustine, Gopal Pandurangan, and Peter Robinson. 2016. Distributed Algorithmic Foundations of Dynamic Networks. *SIGACT News* (2016).

[4] John Augustine, Gopal Pandurangan, Peter Robinson, Scott T. Roche, and Eli Upfal. 2015. Enabling Robust and Efficient Distributed Computation in Dynamic Peer-to-Peer Networks. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015.* IEEE Computer Society.

[5] John Augustine, Gopal Pandurangan, Peter Robinson, and Eli Upfal. 2012. Towards robust and efficient computation in dynamic peer-to-peer networks. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012.* SIAM.

[6] Amitabha Bagchi, Ankur Bhargava, Amitabh Chaudhary, David Eppstein, and Christian Scheideler. 2006. The effect of faults on network expansion. *Theory of Computing Systems* 39, 6 (2006), 903–928. Preliminary version in SPAA'04.

[7] David Barkai. 2001. *Peer-to-peer computing: technologies for sharing and collaborating on the net.* Intel Press.

[8] Luca Becchetti, Andrea Clementi, Emanuele Natale, Francesco Pasquale, and Luca Trevisan. 2020. Finding a bounded-degree expander inside a dense one. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms.* SIAM, 1320–1336.

[9] Luca Becchetti, Andrea Clementi, Francesco Pasquale, Luca Trevisan, and Isabella Ziccardi. 2021. Expansion and flooding in dynamic random networks with node churn. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS).* IEEE, 976–986.

[10] Colin Cooper, Martin E. Dyer, and Catherine S. Greenhill. 2005. Sampling regular graphs and a peer-to-peer network. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver, British Columbia, Canada, January 23-25, 2005.* SIAM.

[11] Bitcoin Core. 2022. Bitcoin Core 0.11 (ch 4): P2P Network. https://en.bitcoin.it/wiki/Bitcoin_Core_0.11_(ch_4):_P2P_Network. Accessed: 2022-07-22.

[12] Ivan Bjerre Damgård. 1987. Collision free hash functions and public key signature schemes. In *Workshop on the Theory and Application of of Cryptographic Techniques.* Springer, 203–216.

[13] Sergi Delgado-Segura, Surya Bakshi, Cristina Pérez-Solà, James Litton, Andrew Pachulski, Andrew Miller, and Bobby Bhattacharjee. 2019. TxProbe: Discovering Bitcoin's network topology using orphan transactions. In *International Conference on Financial Cryptography and Data Security.* Springer, 550–566.

[14] Whitfield Diffie and Martin Hellman. 1976. New directions in cryptography. *IEEE transactions on Information Theory* 22, 6 (1976), 644–654.

[15] Cynthia Dwork and Moni Naor. 1992. Pricing via processing or combatting junk mail. In *Annual International Cryptology Conference.* Springer, 139–147.

[16] Patrick Th. Eugster, Rachid Guerraoui, Sidath B. Handurukande, Petr Kouznetsov, and Anne-Marie Kermarrec. 2003. Lightweight probabilistic broadcast. *ACM Trans. Comput. Syst.* (2003).

[17] Brian Everitt and Anders Skrondal. 2002. *The Cambridge dictionary of statistics.* Vol. 106. Cambridge university press Cambridge.

[18] Giulia Fanti, Shaileshh Bojja Venkatakrishnan, Surya Bakshi, Bradley Denby, Shruti Bhargava, Andrew Miller, and Pramod Viswanath. 2018. Dandelion++: Lightweight Cryptocurrency Networking with Formal Anonymity Guarantees. *Proc. ACM Meas. Anal. Comput. Syst.* (2018).

[19] Giulia Fanti and Pramod Viswanath. 2017. Deanonymization in the Bitcoin P2P Network. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.).

[20] Márk Jelasity, Spyros Voulgaris, Rachid Guerraoui, Anne-Marie Kermarrec, and Maarten van Steen. 2007. Gossip-based peer sampling. *ACM Trans. Comput. Syst.* (2007).
[21] David A Levin and Yuval Peres. 2017. *Markov chains and mixing times*. Vol. 107. American Mathematical Soc.
[22] Andrew Miller, James Litton, Andrew Pachulski, Neal Gupta, Dave Levin, Neil Spring, and Bobby Bhattacharjee. 2015. Discovering bitcoin's public topology and influential nodes. *et al* (2015).
[23] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf.
[24] Satoshi Nakamoto and et Al. 2008. Bitcoin Core. https://github.com/bitcoin/bitcoin. Accessed: 2022-07-22.
[25] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. 2016. *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press.
[26] Gleb Naumenko, Gregory Maxwell, Pieter Wuille, Alexandra Fedorova, and Ivan Beschastnikh. 2019. Erlay: Efficient Transaction Relay for Bitcoin. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*. ACM.
[27] Till Neudecker, Philipp Andelfinger, and Hannes Hartenstein. 2016. Timing analysis for inferring the topology of the bitcoin peer-to-peer network. In *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*. IEEE, 358–367.
[28] Till Neudecker and Hannes Hartenstein. 2018. Network layer aspects of permissionless blockchains. *IEEE Communications Surveys & Tutorials* 21, 1 (2018), 838–857.
[29] James R. Norris. 1998. *Markov chains*. Number 2. Cambridge university press.
[30] Gopal Pandurangan, Prabhakar Raghavan, and Eli Upfal. 2003. Building low-diameter peer-to-peer networks. *IEEE Journal on selected areas in communications* 21, 6 (2003), 995–1002. Preliminary version in FOCS'01.
[31] Angelos Stavrou, Dan Rubenstein, and Sambit Sahu. 2004. A lightweight, robust P2P system to handle flash crowds. *IEEE J. Sel. Areas Commun.* (2004).
[32] Salil P. Vadhan et al. 2012. *Pseudorandomness*. Vol. 7. Now Delft.
[33] Shaileshh Bojja Venkatakrishnan, Giulia Fanti, and Pramod Viswanath. 2017. Dandelion: Redesigning the Bitcoin Network for Anonymity. *Proc. ACM Meas. Anal. Comput. Syst.* (2017).
[34] Spyros Voulgaris, Daniela Gavidia, and Maarten van Steen. 2005. CYCLON: Inexpensive Membership Management for Unstructured P2P Overlays. *J. Netw. Syst. Manag.* (2005).
[35] Liang Wang and Ivan Pustogarov. 2017. Towards better understanding of bitcoin unreachable peers. *arXiv preprint arXiv:1709.06837* (2017).
[36] Addy Yeow. 2013. Global Bitcoin Nodes Distribution. https://bitnodes.io/. Accessed: 2022-07-22.

A E-RAES NON-REVERSIBILITY

A Markov chain $\{X_t\}_t$ with state space Ω and transition matrix P is *reversible* if a probability distribution π over Ω exists such that for every pair of states $x, y \in \Omega$ the following *detailed balanced equation* holds: $\pi(x)P(x,y) = \pi(y)P(y,x)$. The analysis of reversible Markov chains can take advantage of several mathematical tools (see, e.g., [1]) that typically are not available for non-reversible chains. In this appendix we observe that the Markov chain defined by the E-RAES model is non-reversible.

The E-RAES model defines a Markov chain $\mathcal{M}$ where the state space Ω is formed by all the graphs with n nodes and, for two states/graphs $x, y \in \Omega$, $P(x,y)$ is the probability to reach state y from state x following the three steps of the E-RAES model, as defined in Section 2.1. Observe that, given two arbitrary states $x, y \in \Omega$, in general it is not possible to reach state y starting from state x with a sequence of states $x = z_0, z_1, \ldots, z_k = y$ such that $P(z_i, z_{i+1}) > 0$ for every $i = 0, 1, \ldots, k-1$. However, if we restrict the state space to the subset $\hat{\Omega} \subseteq \Omega$ of all the states that can be reached starting from the empty graph $G_0 = (V, \emptyset)$, it is easy to see that the Markov chain restricted to state space $\hat{\Omega}$ is *irreducible* and *aperiodic* (see, e.g., Chapters 1.5-1.7 in [21] for some background). Hence, there is a unique stationary distribution $\hat{\pi}$ over $\hat{\Omega}$ such that, starting from any state $x \in \hat{\Omega}$, $P^t(x, \cdot)$ converges to $\hat{\pi}$ as t goes to infinity (see, e.g., Theorem 4.9 in [21]).

If, by contradiction, there was a probability distribution π over $\hat{\Omega}$ satisfying the detailed balanced equation $\pi(x)P(x,y) = \pi(y)P(y,x)$, then π would be stationary for P (see, e.g., Proposition 1.20 in [21]) and, for the uniqueness of the stationary distribution, we would have $\pi = \hat{\pi}$. Notice that, since $\hat{\pi}$ is the stationary distribution of an irreducible Markov chain with state space $\hat{\Omega}$, then $\hat{\pi}(x) > 0$ for every $x \in \hat{\Omega}$. However, it is not difficult to find two states $x, y \in \hat{\Omega}$ such that $P(x,y) > 0$ and $P(y,x) = 0$. Indeed, consider two graphs x and y such that in both of them each node has degree between d and cd, and the set of edges in y is a subset of the set of edges in x. Clearly $P(x,y) > 0$, since $P(x,y)$ is at least as large as the probability that exactly all the edges in x that are not in y disappear during Step 3 of the E-RAES, and $P(y,x) = 0$, since in y every node has degree between d and cd thus no new edges are created during Steps 1 and 2 of the E-RAES model. Hence for such two states it must be $0 < \hat{\pi}(x)P(x,y) \neq \hat{\pi}(y)P(y,x) = 0$.