

Giant Components in Random Temporal Graphs

Ruben Becker 

Ca' Foscari University of Venice, Italy

Arnaud Casteigts 

University of Geneva, Switzerland

Pierluigi Crescenzi 

Gran Sasso Science Institute, L'Aquila, Italy

Bojana Kodric 

Ca' Foscari University of Venice, Italy

Malte Renken 

Technical University of Berlin, Germany

Michael Raskin 

LaBRI, CNRS, University of Bordeaux, France

Viktor Zamaraev 

University of Liverpool, UK

Abstract

A temporal graph is a graph whose edges appear only at certain points in time. In these graphs, reachability among nodes relies on paths that traverse edges in chronological order (*temporal paths*). Unlike standard paths, temporal paths may not be composable, thus the reachability relation is *not transitive* and connected components (i.e., sets of pairwise temporally connected nodes) do not form equivalence classes, a fact with far-reaching consequences.

Recently, Casteigts et al. [FOCS 2021] proposed a natural temporal analog of the seminal Erdős-Rényi random graph model, based on the same parameters n and p . The proposed model is obtained by randomly permuting the edges of an Erdős-Rényi random graph and interpreting this permutation as an ordering of presence times. Casteigts et al. showed that the well-known single threshold for connectivity in the Erdős-Rényi model fans out into multiple phase transitions for several distinct notions of reachability in the temporal setting.

The second most basic phenomenon studied by Erdős and Rényi in static (i.e., non-temporal) random graphs is the emergence of a *giant connected component*. However, the existence of a similar phase transition in the temporal model was left open until now. In this paper, we settle this question. We identify a sharp threshold at $p = \log n/n$, where the size of the largest temporally connected component increases from $o(n)$ to $n - o(n)$ nodes. This transition occurs significantly later than in the static setting, where a giant component of size $n - o(n)$ already exists for any $p \in \omega(1/n)$ (i.e., as soon as p is larger than a constant fraction of n). Interestingly, the threshold that we obtain holds for both *open* and *closed* connected components, i.e., components that allow, respectively forbid, their connecting paths to use external nodes – a distinction arising from the absence of transitivity.

We achieve these results by strengthening the tools from Casteigts et al. and developing new techniques that provide means to decouple dependencies between past and future events in temporal Erdős-Rényi graphs, which could be of general interest in future investigations of these objects.

2012 ACM Subject Classification Theory of computation → Random network models; Mathematics of computing → Random graphs

Keywords and phrases random temporal graph, Erdős-Rényi random graph, sharp threshold, temporal connectivity, temporal connected component, edge-ordered graph

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2023.29

Category RANDOM

Related Version *Full Version*: <https://arxiv.org/abs/2205.14888v3>

Funding Some of the authors are supported by the German Research Foundation (DFG) under the project MATE (NI 369/17). Some of the authors are supported by the French ANR, project ANR-22-CE48-0001 (TEMPOGRAL). Some of the authors received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 787367 (PaVeS).



© Ruben Becker, Arnaud Casteigts, Pierluigi Crescenzi, Bojana Kodric, Malte Renken, Michael Raskin, and Viktor Zamaraev;

licensed under Creative Commons License CC-BY 4.0

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2023).

Editors: Nicole Megow and Adam D. Smith; Article No. 29; pp. 29:1–29:17



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Many real-world networks vary with time, as exemplified by the dynamic nature of today’s social media, telecommunication, transportation, and interaction in general in a complex network. Indeed, the examination of specific applications illustrates how networks endowed with temporal information enable more accurate and effective analysis of real-world systems compared to static networks [33].

This insight has motivated plethora of studies focusing on network modeling approaches that incorporate the time dimension [24, 25, 27]. A widely used model for these networks is given by *temporal graphs* (sometimes also called *time-varying graphs*, *evolving graphs*, or other names). A temporal graph is a pair $\mathcal{G} = (G, \lambda)$, where $G = (V, E)$ is an *underlying* (static) graph, and λ is an *edge labeling function* that assigns to every edge $e \in E$ a set of time labels $\lambda(e) \subseteq \mathbb{N}$ indicating when this edge is present. This definition, although simple, already captures two important aspects that determine temporal networks. Namely, (a) the topology of the network defined by the underlying graph G ; and (b) the schedule of edge availabilities represented by the labeling function λ .

Even though this model has gained much traction recently, the available tools for analyzing temporal graphs are still nowhere near the level of tools that have been developed for understanding static networks. One of the main challenges is the fundamentally changed notion of *reachability*. In temporal graphs, reachability is naturally based on paths that traverse edges in ascending time, a.k.a. *temporal paths*. A first difference with standard paths is that temporal paths are inherently directed, regardless of whether the graph itself is directed, due to the arrow of time. Even more significantly, temporal reachability is *not transitive*, i.e., the fact that node u can reach node v and node v can reach node w does not imply that u can reach w . The resulting non-composability is a source of complication for structural studies, as well as a frequent source of computational hardness. In fact, many problems related to reachability are hard in temporal graphs, even when their classical analogs are polynomial time solvable – see, for instance, the seminal paper by Kempe, Kleinberg, and Kumar [27] on k -disjoint temporal paths (and many further examples appearing in more recent works [1, 8, 13–15, 20, 22]). As observed by Bhadra and Ferreira [6], the fact that (*temporally*) *connected components* do not form equivalence classes and intersect in non trivial ways implies, among other consequences, that finding one of maximum size is NP-hard.

Random Models of Temporal Graphs

One of the most important tools in (static) network theory are random network models [30]. They allow reproducing characteristics of real networks and studying their statistical properties. The random perspective enables prediction of properties, anomaly detection, identification of phase transitions, and other conclusions about the nature of typical networks.

The cornerstone of random network theory is the Erdős-Rényi random graph model [3]. It has proven tremendously useful as a source of insight into the structure of networks [32]. An Erdős-Rényi random graph $G_{n,p}$ is obtained by placing an edge between each distinct pair of n vertices¹ independently with probability p . The study of this model was sparked by a series of seminal papers published by Erdős and Rényi starting in 1959 [16–19]. Since then, an important number of articles and books have been devoted to this model. These

¹ We use the terms *vertex* and *node* interchangeably.

results laid a solid foundation for the development of other models of more practical interest, including the configuration model [28, 29, 31], the small-world model [34], and the preferential attachment model [4].

The number of models of random *temporal* networks proposed in the literature is still limited and no systematic foundations are available [24]. In establishing such foundations, a natural question is: *What is the temporal analog of the Erdős-Rényi random graph model?* The answer to this question is not unique, as the time dimension can be incorporated in different ways [32]. Some candidates considered in the literature consider a sequence of independent Erdős-Rényi graphs, some others incorporate some dependencies in such a sequence (see for example [2, 5, 10–12, 23, 35]).

Temporal Erdős-Rényi Random Graphs

Recently, another natural and more direct temporal analog of the Erdős-Rényi random graphs was proposed by Casteigts et al. [9], based on the same parameters n and p . In this model, which we refer to as the *temporal Erdős-Rényi random graph model*, a random temporal graph is obtained from an Erdős-Rényi random graph $G_{n,p}$ by assigning to each edge a unique label (presence time) according to a uniformly random permutation of its edges. The main motivation is to obtain a temporal graph model whose properties (such as threshold values) can be directly compared to the classical Erdős-Rényi model, thereby highlighting the qualitative impact of the time dimension. A systematic study of this model may also set a benchmark for practical models.

As already remarked, the time dimension leads to a number of distinctions between static and temporal graphs. Many of them come from the conceptual difference between the notions of path and temporal paths. The reachability of a temporal graph is not symmetric (even in the undirected case) and not transitive, which is in stark contrast with static graphs. Indeed, the results of [9] revealed that even the notion of connectivity translates to a rich spectrum of phase transitions in the temporal setting. Namely, at $p = \log n/n$, any *fixed* pair of vertices can asymptotically almost surely (a.a.s.) reach each other; at $2 \log n/n$, at least one vertex (and in fact, any fixed vertex) can a.a.s. reach all the others; and at $3 \log n/n$, all the vertices can a.a.s. reach all others, i.e., the graph is temporally connected.

Connected Components in Temporal Erdős-Rényi Random Graphs

Perhaps the most investigated aspects of Erdős-Rényi random graphs is the emergence of a “giant” connected component [7, 21], which culminates in connectivity itself. The analogous question in a temporal setting is therefore natural. Interestingly, the lack of transitivity makes the very definition of temporal components ambiguous. If the vertices of the component need temporal paths traveling *outside* the component in order to reach each other, then the component is *open*; otherwise, it is *closed* [6].

Analyzing the emergence of (both types of) *temporally* connected components in the above model presents technical challenges that cannot be overcome by the only tools developed in [9]. These technical challenges and the importance of understanding connected components in temporal Erdős-Rényi random graphs motivated the present work.

1.1 Contributions

In this paper, we analyze the evolution of the largest connected component in a temporal Erdős-Rényi random graph with parameters n and p , as p increases (with $n \rightarrow \infty$). Our main result is that, in contrast to static graphs, the phase transition occurs at $p = \log n/n$. At this point, the size of the largest component jumps from $o(n)$ to $n - o(n)$.

► **Main Theorem** (informal). *There exists a function $\varepsilon(n) \in o(\log n/n)$ such that the size of a largest temporally connected component in a temporal Erdős-Rényi random graph is*

- (i) $o(n)$ a.a.s., if $p < \frac{\log n}{n} - \varepsilon(n)$; and
- (ii) $n - o(n)$ a.a.s., if $p > \frac{\log n}{n} + \varepsilon(n)$.

Notably, the same threshold holds for both open and closed connected components, although showing the latter requires more effort. We achieve these results by developing new techniques and combining them with strengthened versions of the tools from [9]. Informally, the new tools enable us to effectively contain the dependencies that exist between different time slices. Thus they facilitate building graph structures witnessing a desired property in *multiple independent phases*.

1.2 Significance of the Results & Techniques

Results. Our main result reveals a qualitative difference between the evolution of connected components in static random graphs and temporal random graphs. The emergence of a giant component in (static) Erdős-Rényi graphs follows a well-known pattern of events [19]. Below a critical probability $p_0 = 1/n$, almost all the components are trees, and no component is larger than $O(\log n)$. Then, at p_0 , a single “giant” component of size $\Theta(n^{2/3})$ arises. Then, at $p = c/n > 1/n$, this component contains a constant fraction $1 - x/c$ of all vertices (with $0 < x < 1$ being defined through $xe^{-x} = ce^{-c}$). As soon as $p \in \omega(1/n)$, the component contains all but $o(n)$ vertices. The case of directed static graphs is similar. Namely, for $p = c/n < 1/n$, a.a.s. all strongly connected components have size less than $3c^{-2} \log n$, and when $p = c/n > 1/n$, the graph contains a strongly connected component of size approximately $(1 - x/c)^2 n$ (with x as above) [21, 26], which implies that this component contains all but $o(n)$ vertices when $p \in \omega(1/n)$.

In the temporal setting, we show that the phase transition occurs at $p = \log n/n$. Namely, all components are of size $o(n)$ before that threshold and there is one component of size $n - o(n)$ afterwards. The fact that this transition occurs later in the temporal setting is not surprising, as the thresholds for *connectivity* is already known to be significantly smaller in the static setting than in the temporal setting; namely, connectivity occurs at $p = \log n/n$ in the static case (for both directed and undirected graphs) versus $p = 3 \log n/n$ for temporal connectivity [9]. However, while these thresholds for connectivity are within a multiplicative constant of each other, our results show that in the case of connected components the static and the temporal threshold are of distinct asymptotic orders.

Techniques. In the temporal Erdős-Rényi model, the unicity of presence times for the edges causes delicate dependencies between past and future events. To contain these dependencies, we introduce a multiphase analysis that consists of splitting the time interval into several phases where these dependencies are decoupled. We believe that many further temporal graph properties will require such a multiphase analysis and could benefit from the tools developed here. In contrast, the techniques from [9] are well suited for analyzing single-phase processes, where temporal paths do not interact across different time intervals (e.g. through composition).

In particular, our switch from a fixed base graph $G = K_n$ to an arbitrary graph of high minimum degree provides the possibility to “encapsulate” all dependencies on events occurring in some fixed “short” phase into the choice of base graph, effectively eliminating the need to deal with these dependencies individually. As an unsurprising but quite useful technical extension, we study also the behaviour of sets of journeys starting from any of a

set of source vertices. Furthermore, Lemma 5.2 is of independent interest for “bootstrapping” various such multiphase analysis attempts; it essentially proves that in the very early regime, there is only a small number of poorly connected vertices, and that these can be removed without compromising the connectivity of the remaining temporal graph.

Although our techniques handle specific dependencies of temporal Erdős-Rényi graphs, they remain general enough to be adaptable to models with less dependencies, such as models where several appearances of an edge is possible and these appearances follow an exponential distribution (Poisson process). The reasons for this are exactly the same as in [9]. Note, however, that weaker tools could suffice for such models, as past and future appearances of an edge are independent.

1.3 Organization

In Section 2, we provide all necessary definitions, and introduce the random temporal graph models used in the paper. In Section 3, we present the algorithm for constructing a foremost forest. We also state a core technical theorem (Theorem 3.3) concerned with reachability between two sets of nodes in a temporal graphs. The full proof of that theorem is deferred to the full version due to space restrictions, as are several other proofs and intermediate results. Using this theorem, we then prove in Section 4 that at $p = \log n/n$ the size of the largest *open* connected component jumps from $o(n)$ to $n - o(n)$. This also serves as a stepping stone towards Section 5, where we extend our technique to also apply to *closed* connected components. The proof is slightly more involved than for open components, as it requires further subdivisions of the phases. However, we show that both variants undergo essentially the same phase transition.

2 Preliminaries

In this paper, $[k]$ denotes the set of integers $\{1, \dots, k\}$, and $[a, b]$ denotes either the discrete interval from a to b , or the continuous interval from a to b , the distinction being clear from the context. All graphs are simple, i.e., without loops or multiple edges. For a graph G , we denote by $V(G)$ and $E(G)$ its vertex set and edge set respectively. We denote by $\delta(G)$ and $\Delta(G)$ the minimum and the maximum vertex degree of G respectively. As usual, K_n denotes the complete n -vertex graph.

2.1 Temporal Graphs

A *temporal graph* is a pair (G, λ) , where $G = (V, E)$ is a *static* graph and λ is a function that assigns to every edge $e \in E$ a finite set of numbers, interpreted as presence times. The graph G is called the *underlying graph* of the temporal graph and the elements of $\lambda(e)$ are called the *time labels* of e . We will denote temporal graphs by calligraphic letters, e.g., by \mathcal{G} . Instead of (G, λ) we will sometimes use the notation (V, E, λ) to denote the same temporal graph. In most cases, time labels will be elements of the real unit interval $[0, 1]$. Furthermore, in this paper, we restrict our consideration only to *simple* temporal graphs², i.e., temporal graphs in which every edge $e \in E$ is only present at a single point in time, i.e., $|\lambda(e)| = 1$. We sometimes write $V(\mathcal{G})$ and $E(\mathcal{G})$ for the node and edge set of a temporal graph \mathcal{G} respectively.

² We remark that all our results can be directly transferred to another, closely related model of non-simple temporal graphs; see Section 6.1.2 in [9].

A temporal graph $\mathcal{H} = (V_{\mathcal{H}}, E_{\mathcal{H}}, \lambda_{\mathcal{H}})$ is a *temporal subgraph* of a temporal graph $\mathcal{G} = (V_{\mathcal{G}}, E_{\mathcal{G}}, \lambda_{\mathcal{G}})$, if $V_{\mathcal{H}} \subseteq V_{\mathcal{G}}$, $E_{\mathcal{H}} \subseteq E_{\mathcal{G}}$ and $\lambda_{\mathcal{H}}(e) = \lambda_{\mathcal{G}}(e)$ for all $e \in E_{\mathcal{H}}$. For a vertex set $S \subseteq V(\mathcal{G})$, we denote by $\mathcal{G}[S]$ a temporal subgraph of \mathcal{G} induced by S . We use $\mathcal{G}_{[a,b]}$ to denote the temporal subgraph of \mathcal{G} with the same node set $V_{\mathcal{G}}$, the edge set $E' := \{e \in E_{\mathcal{G}} : \lambda_{\mathcal{G}}(e) \in [a, b]\}$, and the time labeling function $\lambda_{\mathcal{G}|_{E'}}$ which is the restriction of $\lambda_{\mathcal{G}}$ to E' .

A *temporal (u, v) -path* in $\mathcal{G} = (V, E, \lambda)$ between two nodes $u, v \in V$ is a sequence $u = u_0, u_1, \dots, u_{\ell} = v$ such that $e_i = \{u_{i-1}, u_i\} \in E$ for each $i \in [\ell]$, and time labels are increasing, i.e., $\lambda(e_1) < \dots < \lambda(e_{\ell})$. We call $\lambda(e_{\ell})$ the *arrival time* of the path. A temporal (u, v) -path is called *foremost (u, v) -path* if it has the earliest arrival time among all temporal (u, v) -paths. If there exists a temporal (u, v) -path, we say that u can reach v (every vertex reaches itself). A set $S \subseteq V$ is said to *reach* v if at least one of its elements reaches v . In that case, a *foremost (S, v) -path* in \mathcal{G} is a temporal (u, v) -path with earliest arrival time among all $u \in S$.

A vertex $u \in V$ is called *temporal source* in $\mathcal{G} = (V, E, \lambda)$ if there exists a temporal (u, v) -path for each $v \in V$. Similarly, a vertex $v \in V$ is called *temporal sink* in \mathcal{G} if there exists a temporal (u, v) -path for each $u \in V$.

A temporal graph $\mathcal{G} = (V, E, \lambda)$ is *temporally connected* if all nodes are temporal sources. We note that this also implies that all nodes are temporal sinks. An *open temporally connected component* or simply *connected component* in \mathcal{G} is an inclusion-wise maximal set $Z \subseteq V$ of nodes such that for every ordered pair of vertices $u, v \in Z$, there exists a temporal (u, v) -path in \mathcal{G} . We stress that such a temporal (u, v) -path can contain nodes from $V \setminus Z$. If for every ordered pair $u, v \in Z$, there exists a temporal (u, v) -path in $\mathcal{G}[Z]$, then Z is called *closed connected component*.

2.2 Random Temporal Graph Models

The model of *temporal Erdős-Rényi random graphs* was introduced in [9]³ as a natural temporal generalization of the classical Erdős-Rényi model $G_{n,p}$ of random graphs. An n -vertex temporal Erdős-Rényi random graph with the parameter $p \in [0, 1]$ is obtained by first drawing a *static* random Erdős-Rényi $G_{n,p}$ and then defining a temporal order on its edges by ordering them according to a uniformly random permutation. An equivalent and technically more convenient way of defining the temporal order on the edges is to draw, for every edge e , independently and uniformly at random a time label $\lambda(e)$ from the unit interval $[0, 1]$. Since the event that two edges get the same time label happens with probability 0, all edge orderings induced by such random time labels are equiprobable. Therefore, as long as the absolute values of time labels are irrelevant (which is the case for the questions studied in [9] and in the present paper), the two models are indeed equivalent. This latter model is denoted as $\mathcal{F}_{n,p}$. A possible way of generating objects from $\mathcal{F}_{n,p}$ is to first draw a temporal graph $\mathcal{G} = (G, \lambda)$ from $\mathcal{F}_{n,1}$ (thus the underlying graph G is complete), and to then consider $\mathcal{G}' = (G', \lambda') = (G, \lambda)|_{[0,p]}$, i.e., the temporal graph obtained from \mathcal{G} by removing edges with time labels greater than p . Observe that $G' \sim G_{n,p}$ and each time label $\lambda(e)$ is uniformly distributed on $[0, p]$. Hence, \mathcal{G}' is distributed according to $\mathcal{F}_{n,p}$ up to multiplying all labels by a factor of $\frac{1}{p}$, which we can ignore as it neither changes the relative order of time labels nor the absolute values of time labels are of any importance to us. For similar reasons, for any $0 \leq a \leq b \leq 1$, up to rescaling time labels, the temporal subgraph $\mathcal{G}|_{[a,b]}$ is distributed according to $\mathcal{F}_{n,q}$, where $q = b - a$.

³ In [9], this model was called Random Simple Temporal Graphs (RSTGs)

In order to overcome some technical challenges caused by interdependence of different temporal subgraphs, we define and study a natural generalization of $\mathcal{F}_{n,p}$ that we describe next. For an n -vertex graph G and a real value $p \in [0, 1]$, we denote by $\mathcal{F}_p(G)$ the following random temporal graph model. A random temporal graph $\mathcal{G} = (V, E, \lambda) \sim \mathcal{F}_p(G)$ is obtained by (1) independently and uniformly sampling a time label $\lambda'(e) \in [0, 1]$ for every $e \in E(G)$, and (2) setting $V = V(G)$, $E := \{e \in E(G) : \lambda(e) \leq p\}$ and $\lambda(e) = \lambda'(e)$ for every $e \in E$. We call G the *base graph* of $\mathcal{F}_p(G)$. We observe that the $\mathcal{F}_{n,p}$ model is obtained when choosing the base graph to be the complete n -vertex graph K_n .

In what follows we sometimes implicitly assume that $n = |V|$ is sufficiently large without restating this assumption. We note that some of our estimates hold only for rather large values of n . We did not attempt to reduce these bounds, but instead focused on achieving best possible readability.

At this point we refer the interested reader to Appendix A of the full version, where, as a warm-up, we give a simple upper bound on p which guarantees that $\mathcal{G} \sim \mathcal{F}_p(G)$ is temporally connected a.a.s.

3 The Foremost Forest Algorithm

The main aim of this section is to present an algorithm for constructing a foremost forest and to prove a property of this algorithm.

Foremost forests play a crucial role in the development of our main technical tool: for a fixed set of vertices S and a given number k , the estimation of the minimum value of p such that the vertices in S can reach k vertices in $\mathcal{G} = (V, E, \lambda) \sim \mathcal{F}_p(G)$ a.a.s.

We obtain such an estimation by examining the evolution of a foremost forest for S in \mathcal{G} via analysis of the execution of the foremost forest algorithm on random temporal graphs. To elaborate on this approach, let us consider $v \in V \setminus S$. We would like to estimate the probability that S reaches v in \mathcal{G} . For this, we follow an approach similar to the one used in [9]. Let $\mathcal{G}' \sim \mathcal{F}_1(G)$ and observe that the probability that S can reach v in \mathcal{G} is equal to the probability that the temporal subgraph $\mathcal{G}'_{[0,p]}$ contains a temporal (u, v) -path P for some node $u \in S$. This again is equivalent to the arrival time of P in \mathcal{G}' being at most p . Therefore, the estimation of the parameter p for which some node from S can reach v can be reduced to the estimation of the minimum arrival time of a foremost temporal path from S to v in $\mathcal{G}' \sim \mathcal{F}_1(G)$. A *foremost forest for S in \mathcal{G}* is a minimal temporal subgraph that preserves foremost reachabilities from S to all other vertices reachable from S in \mathcal{G} . We proceed with the necessary formal definitions.

► **Definition 3.1.** Let $\mathcal{G} = (V, E, \lambda)$ be a temporal graph and let $S \subseteq V$ be a set of vertices. The graph $\mathcal{G}_F = (V_F, E_F, \lambda_F)$ is an increasing temporal forest for S , if

- (a) \mathcal{G}_F is a temporal subgraph of \mathcal{G} ,
- (b) the graph $F = (V_F, E_F)$ is a forest (i.e. acyclic graph) with $|S|$ components,
- (c) for each $s \in S$ there is a connected component T_s of F such that s reaches all vertices of T_s in \mathcal{G}_F .

We are now ready to define (partial) foremost forests.

► **Definition 3.2.** Let $\mathcal{G} = (V, E, \lambda)$ be a temporal graph, let $S \subseteq V$ be a set of vertices and let $\mathcal{G}_F = (V_F, E_F, \lambda_F)$ be an increasing temporal forest for S .

1. Then \mathcal{G}_F is a partial foremost forest for S , if, for all $v \in V_F \setminus S$, the unique temporal (S, v) -path in \mathcal{G}_F is a foremost (S, v) -path in \mathcal{G} .
2. A partial foremost forest for S is a foremost forest for S if V_F contains all vertices reachable from S in \mathcal{G} , i.e., $V_F = \{v \in V : \exists (u, v)\text{-temporal path in } \mathcal{G} \text{ for some } u \in S\}$.
3. A (partial) foremost forest for $\{v\}$ is a (partial) foremost tree for v .

The Algorithm

Next, we present an algorithm that, given a temporal graph $\mathcal{G} = (V, E, \lambda)$ and a set of nodes $S \subseteq V$ constructs a foremost forest \mathcal{G}_F for S . This algorithm is a straightforward generalization of the foremost tree algorithm from [9], where the input set S is assumed to be singleton.

The idea of the algorithm similar to Prim's algorithm for minimum spanning trees in static graphs: Starting from $\mathcal{G}_F = (V_F, E_F, \lambda_F) = (S, \emptyset, \emptyset)$, which is trivially a partial foremost forest for S , the algorithm iteratively adds one node and one edge to V_F and E_F, λ_F , respectively, until \mathcal{G}_F becomes a foremost forest for S . The main difference to Prim's algorithm is that, in every iteration, the next edge to be added is chosen as the edge of minimum time label among all edges that extend the current increasing temporal forest. For brevity, we introduce the following notation. We write $\mathcal{G}_F \cup e$ for adding the edge $e = \{u, v\}$ to \mathcal{G}_F , i.e., the result is the temporal graph $(V_F \cup \{u, v\}, E_F \cup \{e\}, \lambda_F \cup \{(e, \lambda(e))\})$. The set of edges that extend the current partial forest can then be defined as

$$\text{ext}(\mathcal{G}_F) := \{e = \{u, v\} \in E : u \in V_F, v \in V \setminus V_F, \text{ and } \mathcal{G}_F \cup e \text{ is an increasing temporal forest for } S\}.$$

We are now ready to state the algorithm.

■ **Algorithm 1** FOREMOST FOREST.

Input : Simple temporal graph $\mathcal{G} = (V, E, \lambda)$; set of nodes $S \subseteq V$.
Output : Foremost forest for S .

- 1 $k = |S| - 1, \mathcal{G}_F^k = (S, \emptyset, \emptyset)$
- 2 **while** $\text{ext}(\mathcal{G}_F^k) \neq \emptyset$ **do**
- 3 $k := k + 1$
- 4 $e_k := \arg \min \{\lambda(e) \mid e \in \text{ext}(\mathcal{G}_F^{k-1})\}$
- 5 $\mathcal{G}_F^k := \mathcal{G}_F^{k-1} \cup e_k$
- 6 **return** \mathcal{G}_F^k

In Appendix B of the full version we prove that Algorithm 1 in fact builds a foremost forest. Furthermore, one of our main technical results is the following theorem which, for two given sets of nodes S and T , quantifies the probability that a foremost forest grown from set S reaches T .

► **Theorem 3.3** (Foremost Forest Target Set Reachability). *Let*

- G be a graph of minimum degree $\delta(G) \geq n - (\log n)^a$ for some $a \in \mathbb{N}$,
- let S and T be two sets of nodes in G of cardinalities $s \in [(\log n)^{13}, n/2]$ and t , respectively,
- let $z = z(n)$ be a function with $\varepsilon \leq z(n) \leq 1 - \varepsilon$ for some constant $\varepsilon \in (0, 1)$, and
- let $\mathcal{G} \sim \mathcal{F}_p(G)$ with $p \geq \frac{z \log n - \log s}{n} + \frac{3 \log \log n}{n}$.

Then the foremost forest algorithm from S on \mathcal{G} reaches T with probability at least $1 - \frac{5}{2} n^{-\log \log n} - e^{-\frac{t}{2n}(n^z - s)}$.

The formal proof of Theorem 3.3 is one of the technically more involved portions of this work. It is divided into a number of lemmas and has to be deferred to Appendix C of the full version due to lack of space; for improved accessibility, a high level overview of the proof structure is depicted in Figure 1. We proceed with a short proof sketch.

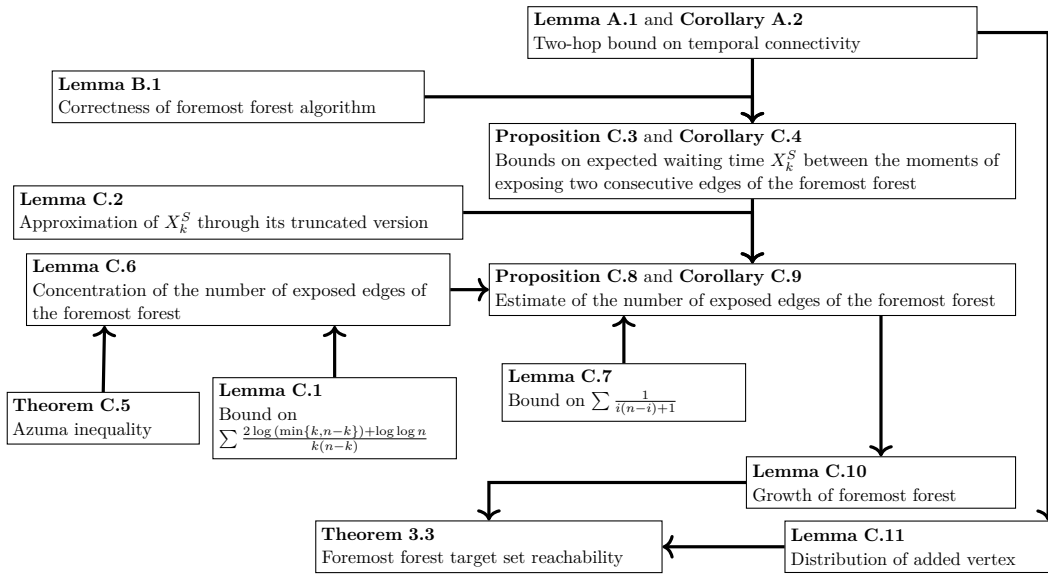


Figure 1 Overview of the proof of Theorem 3.3.

Proof Sketch. The theorem is deduced from Lemma C.10 and Lemma C.11 that can be found in Appendix C of the full version. Lemma C.10 essentially constitutes a generalization of the foremost tree growth analysis from [9], which estimates the number of vertices that a given vertex (referred to as a *source*) reaches by specific time in $\mathcal{F}_{n,p}$. Besides the difference that in Lemma C.10 we need to consider a fixed *set of source* vertices, the main technical challenge here is that we have to consider the $\mathcal{F}_p(G)$ model rather than the basic $\mathcal{F}_{n,p}$ model, resulting in fewer edges per node. While Lemma C.10 merely gives a statement over the number of nodes that are reached from a given source set, Lemma C.11 gives the second crucial ingredient for proving Theorem 3.3. It states that every new vertex reached by the foremost forest grown from S (i.e., every new vertex added to the foremost forest) is distributed almost uniformly on the vertices that are not reached yet and this allows us to estimate the probability that the forest reaches the target set T . ◀

4 Sharp Threshold for Giant Open Connected Component

In this section, we report on our first main result.

► **Theorem 4.1 (Main Result for Open Components).** *The function $\frac{\log n}{n}$ is a sharp threshold for Giant Open Connected Component. More specifically, there exists a function $\varepsilon(n) \in o\left(\frac{\log n}{n}\right)$, such that the size of a largest open temporally connected component in $\mathcal{G} \in \mathcal{F}_{n,p}$ is*

- (i) $o(n)$ a.a.s., if $p < \frac{\log n}{n} - \varepsilon(n)$; and
- (ii) $n - o(n)$ a.a.s., if $p > \frac{\log n}{n} + \varepsilon(n)$.

We prove the lower bound on the threshold (i.e. Theorem 4.1 (i)) in Section 4.1. The proof of this bound is a straightforward consequence of a result on foremost *tree* growth in $\mathcal{F}_p(K_n)$ from [9]. The upper bound (i.e. Theorem 4.1 (ii)) on the threshold is proved in Section 4.2 and is significantly more involved. In particular, it relies on Theorem 3.3 to measure foremost *forest* growth in $\mathcal{F}_p(G)$, where G is chosen to contain all edges that did *not* occur within some particular time window.

4.1 Lower Bound on the Threshold

We state the lower bound in form of the following theorem which says that a.a.s. there is no linear size component before time $\log n/n$. This theorem can be derived rather easily from results of Casteigts et al. [9]; we refer to Appendix D of the full version for the details.

► **Theorem 4.2** (Lower Bound in Theorem 4.1). *Let $\mathcal{G} \sim \mathcal{F}_p(K_n)$ with $p < \frac{\log n}{n} - \frac{3(\log n)^{0.8}}{n}$. Then, for any constant $c \in (0, 1)$, the graph \mathcal{G} does not contain a temporally connected component of size at least $c \cdot n$ with probability at least $1 - 2n^{-\sqrt{\log n}}$.*

4.2 Upper Bound on Threshold

Next, we present the first, weaker version of our main result, stating that an open temporally connected components containing almost all vertices appears already around time $\log n/n$.

► **Theorem 4.3** (Upper Bound in Theorem 4.1). *Let $\mathcal{G} \sim \mathcal{F}_p(K_n)$ with $p \geq (1 + \varepsilon(n)) \cdot \frac{\log n}{n}$. Then, the graph \mathcal{G} contains a temporally connected component of size $n - o(n)$ a.a.s.*

We begin by giving a sketch of the proof idea.

Proof Sketch. The strategy is as follows, see also Figure 2. We split the time interval $[0, p]$ into three intervals I_1 , I_2 , and I_3 of equal duration $p/3$, and reveal the edges of the graph in two phases.

In Phase 1, we reveal the edges whose time labels are in one of the intervals I_1 and I_3 . Using a result from [9] (Lemma D.2), we can conclude that there are $n - o(n)$ nodes (call them X), each of which a.a.s. reaches at least $\sqrt[3]{n} \log n$ vertices during I_1 , and there are at least $n - o(n)$ nodes (call them Y) that a.a.s. is reached by at least $\sqrt[3]{n} \log n$ vertices during I_3 .

In Phase 2, we reveal the edges appearing during the middle interval I_2 . We show that for every ordered pair of nodes x, y in the set $Z := X \cap Y$ (which is our intended connected component), the set of vertices that x can reach during I_1 , can reach during I_2 at least one vertex in the set of vertices that reach y during I_3 ; thus implying that x can reach y during $[0, p]$. For this purpose we can employ Theorem 3.3 with S being the set that x can reach during I_1 and T being the set of vertices that can reach y during I_3 . Note that the analysis of this phase is what requires us to develop the generalization $\mathcal{F}_p(G)$ of the model $\mathcal{F}_{n,p}$. In fact, the static base graph G used in the application of Theorem 3.3 is the graph obtained from K_n by removing the edges that appeared during either I_1 or I_3 . Finally a union bound over all pairs of nodes x and y yields the result. ◀

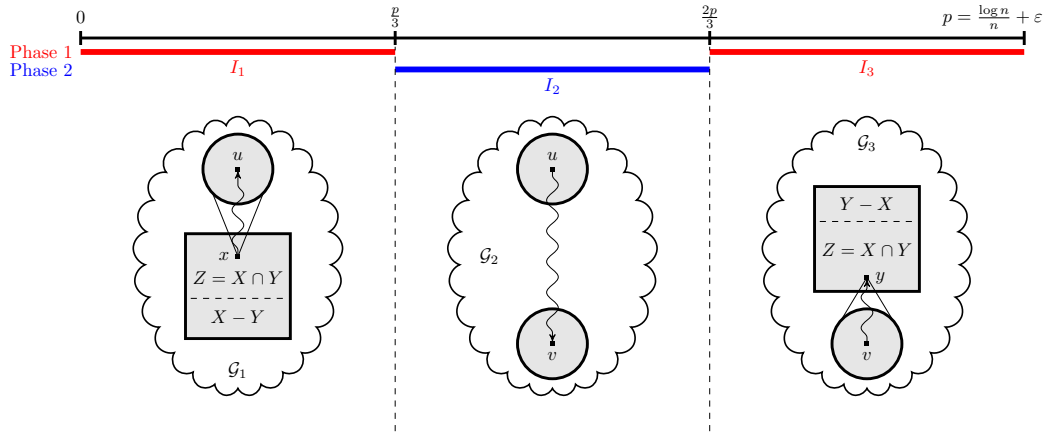
The remainder of this section is dedicated to proving Theorem 4.3. Throughout, we denote $\varepsilon(n) := \frac{1}{\log \log n}$.

Let $p = (1 + \varepsilon(n)) \cdot \frac{\log n}{n}$ and $\mathcal{G} \sim \mathcal{F}_p(K_n)$. We will prove Theorem 4.3 only for this value of p as it will then clearly follow for any larger value. Our strategy is to split the interval $[0, p]$ into three sub-intervals $[p_0, p_1]$, $[p_1, p_2]$, $[p_2, p_3]$, where $p_i := \frac{i}{3}(1 + \varepsilon(n)) \frac{\log n}{n}$ for $i \in [0, 3]$. We now first deduce the following corollary about the connectivity of the subgraphs $\mathcal{G}_{[p_i, p_{i+1}]}$ for $i \in [0, 2]$ of \mathcal{G} from Lemma D.2.

► **Corollary 4.4.** *For $i \in [0, 2]$, the number of vertices reached by (resp. reaching) a fixed vertex in $\mathcal{G}_{[p_i, p_{i+1}]}$ lies within $[n^{1/3} \log n, n^{1/3+\varepsilon(n)}]$ with probability at least $1 - \frac{10}{\log n}$.*

For space reasons, the proof of Corollary 4.4 is found in Appendix E of the full version.

Using Markov's inequality we can obtain that, a.a.s., almost all nodes can reach (resp. be reached by) the above number of nodes.



■ **Figure 2** General strategy for upper bounding the value of p in the case of open components. Here, \mathcal{G}_i denotes the restriction of the temporal graph to subinterval I_i . Wavy lines denote temporal paths. We show that any node $x \in Z$ can reach any other node $y \in Z$ by reaching a node u in \mathcal{G}_1 , then a node v in \mathcal{G}_2 , and finally y in \mathcal{G}_3 .

► **Lemma 4.5.** *Let $i \in \{0, 1, 2\}$. The number of vertices that can reach (resp. be reached by) at least $n^{1/3} \log n$ and at most $n^{1/3+\varepsilon(n)}$ vertices in $\mathcal{G}_{[p_i, p_{i+1}]}$ is at least $n - \frac{n}{\log \log n}$ with probability at least $1 - \frac{10 \log \log n}{\log n}$.*

Proof. Let \bar{X} denote the number of nodes in $\mathcal{G}_{[p_i, p_{i+1}]}$ that can reach (resp. be reached by) less than $n^{1/3} \log n$ or more than $n^{1/3+\varepsilon(n)}$ vertices in $\mathcal{G}_{[p_i, p_{i+1}]}$. Then $\mathbb{E}[\bar{X}] \leq 10n / \log n$ by Corollary 4.4. Using Markov's inequality $\mathbb{P}[\bar{X} \geq \frac{n}{\log \log n}] \leq \frac{10 \log \log n}{\log n}$. ◀

We now denote by X the set of nodes that can reach at least $n^{1/3} \log n$ and at most $n^{1/3+\varepsilon(n)}$ vertices in $\mathcal{G}_{[0, p_1]}$ and by Y the set of nodes that are reached by at least $n^{1/3} \log n$ and at most $n^{1/3+\varepsilon(n)}$ vertices in $\mathcal{G}_{[p_2, p_3]}$. Furthermore, we denote by $Z = X \cap Y$ their intersection. According to Lemma 4.5, it holds that $|Z| \geq n - \frac{2n}{\log \log n}$ with probability at least $1 - \frac{20 \log \log n}{\log n}$. The hardest part of our proof is to now show that, for a fixed ordered pair $x, y \in Z$, the probability that there is a temporal path from x to y is so large that we can take a union bound over all ordered pairs. To this end, let $A(x)$ be the set of nodes that x can reach in $\mathcal{G}_{[0, p_1]}$ and let $B(y)$ be the set of nodes that can reach y in $\mathcal{G}_{[p_2, p_3]}$. Furthermore, for $x \in X$, let

$$A'(x) := \{v \in V : \exists a \in A(x) \text{ s.t. } a \text{ reaches } v \text{ in } \mathcal{G}_{[p_1, p_2]}\}$$

be the set of nodes that x can reach in $\mathcal{G}_{[0, p_2]}$. Notice that x reaches y if and only if $A'(x)$ intersects $B(y)$.

Let $G'' = (V, E'')$ with $E'' = \{e \in \binom{V}{2} \mid \lambda(e) \in [0, p_1] \cup [p_2, p_3]\}$ be the graph containing all edges appearing in $\mathcal{G}_{[0, p_1]}$ or $\mathcal{G}_{[p_2, p_3]}$, and let $G' = (V, E')$ with $E' = \binom{V}{2} \setminus E''$ contain all other edges. Then we observe that the distribution of the set $A'(x)$ conditioned on the information about the edges appearing in $\mathcal{G}_{[0, p_1]}$ and $\mathcal{G}_{[p_2, p_3]}$ is identical to the node set of a foremost forest grown from $S := A(x)$ in $\mathcal{H} \sim \mathcal{F}_{p'}(G')$, where $p' = \frac{1}{3}(1 + \varepsilon(n)) \frac{\log n}{n}$. Furthermore, G'' is distributed as an Erdős-Rényi graph $G'' \sim G_{n, p}$ with $p := \frac{2}{3}(1 + \varepsilon(n)) \frac{\log n}{n}$. From a standard result regarding the maximum degree in $G_{n, p}$ we can thus conclude the following fact.

29:12 Giant Components in Random Temporal Graphs

► **Observation 4.6.** *It holds that $\Delta(G'') \leq 4 \log n$ a.a.s. and, thus, $\delta(G') \geq n - (\log n)^2$ a.a.s.*

Proof. Recall that G'' is distributed according to $G_{n,p}$ with $p := \frac{2}{3}(1 + \varepsilon(n)) \frac{\log n}{n}$. Following [7, Corollary 3.13], with $m = 1$ and $\omega(n) = \log n$, we have that a.a.s.

$$\Delta(G'') \leq pn + \sqrt{2pn \log n} + \log n \sqrt{\frac{pn}{\log n}} \leq \log n + \sqrt{2(\log n)^2} + \log n \leq 4 \log n.$$

The observation about the minimum degree now follows immediately for sufficiently large n . ◀

Thus, in order to lower bound the probability that $A'(x)$ intersects $B(y)$, we can use the following corollary of Theorem 3.3.

► **Corollary 4.7.** *Let*

- *G be a graph of minimum degree $\delta(G) \geq n - (\log n)^a$ for some $a \in \mathbb{N}$,*
- *let S and T be two sets of nodes in G , each of cardinality at least $n^{1/3} \log n$, and*
- *let $\mathcal{G} \sim \mathcal{F}_p(G)$ with $p \geq \frac{1}{3}(1 + \varepsilon(n)) \frac{\log n}{n}$.*

Then, the foremost forest algorithm from S on \mathcal{G} reaches T with probability at least $1 - 3n^{-\log \log n}$.

Proof. Set $s := |S|$, $t := |T|$. Without loss of generality, we may assume $s \leq n^{1/3 + \varepsilon(n)}$. Note that for large enough n it holds that

$$\begin{aligned} p &\geq \frac{1}{3} \left(1 + \frac{1}{\log \log n}\right) \frac{\log n}{n} \\ &\geq \frac{\frac{1}{3} \log n + 4 \log \log n}{n} \\ &= \frac{\frac{2}{3} \log n + 2 \log \log n - \frac{1}{3} \log n - \log \log n}{n} + \frac{3 \log \log n}{n} \\ &\geq \frac{z \log n - \log s}{n} + \frac{3 \log \log n}{n}, \end{aligned}$$

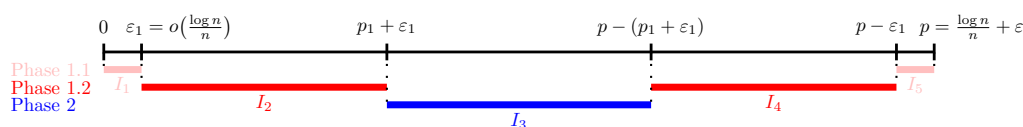
for $z = \frac{2}{3} + \frac{2 \log \log n}{\log n}$. From Theorem 3.3 it then follows that the foremost forest algorithm from S reaches T with probability at least

$$\begin{aligned} 1 - \frac{5}{2} n^{-\log \log n} - e^{-\frac{t}{2n}(n^z - s)} &\geq 1 - \frac{5}{2} n^{-\log \log n} - e^{-\frac{n^{1/3} \log n}{2n} (n^{2/3} (\log n)^2 - n^{1/3 + \varepsilon(n)})} \\ &\geq 1 - \frac{5}{2} n^{-\log \log n} - e^{-\frac{(\log n)^3}{4}} \geq 1 - 3n^{-\log \log n}, \end{aligned}$$

completing the proof. ◀

Using the above stated corollary, we can finally prove our first main result.

Proof of Theorem 4.3. Let $p = (1 + \varepsilon(n)) \cdot \frac{\log n}{n^{1/3 + \varepsilon(n)}}$ and $\mathcal{G} \sim \mathcal{F}_p(K_n)$. As above, let X be the nodes that can reach between $n^{1/3} \log n$ and $n^{1/3 + \varepsilon(n)}$ vertices in $\mathcal{G}_{[0, p_1]}$ and let Y be the nodes that are reached by between $n^{1/3} \log n$ and $n^{1/3 + \varepsilon(n)}$ vertices in $\mathcal{G}_{[p_2, p_3]}$. Furthermore, let $Z = X \cap Y$ be their intersection and recall that $|Z| \geq n - \frac{2n}{\log \log n}$ with probability at least $1 - \frac{20 \log \log n}{\log n}$ according to Lemma 4.5. Now, conditioned on the information about the edges appearing in $\mathcal{G}_{[0, p_1]}$ and $\mathcal{G}_{[p_2, p_3]}$, let $G' = (V, E')$ be the static graph with the same node set as \mathcal{G} and the edge set $E' = \{e \in \binom{V}{2} : \lambda(e) \notin [p_0, p_1] \cup [p_2, p_3]\}$, where λ is the time label function of \mathcal{G} . Note that according to Observation 4.6 the minimum degree in G' a.a.s. is at



■ **Figure 3** Illustration of the three different phases in our proof for the case of closed components. Here, the length p_1 of I_2 and I_4 and the length p_2 of I_3 are each roughly $\frac{1}{3} \frac{\log n}{n}$. In Phase 1.1, we reveal edges in I_1 and I_5 and identify our target closed connected component, a set of n' nodes V' each of which can reach (be reached by) poly-logarithmically many vertices within V' during I_1 (I_5) via temporal paths in V' . For Phase 1.2 (consisting of intervals I_2 and I_4) we show that every vertex in $v \in V'$ reaches (is reached by) polynomially many vertices in V' during $I_1 \cup I_2$ ($I_4 \cup I_5$). We then show that during Phase 2 (consisting of I_3), for each ordered pair of vertices $u, w \in V'$, the set of vertices reached by u during $I_1 \cup I_2$ can reach the set of vertices that reach w during $I_4 \cup I_5$, implying that u can reach w during $[0, p]$.

least $n - (\log n)^2$. Now, let $x, y \in Z$ be a fixed ordered pair of vertices. Applying Corollary 4.7 to $\mathcal{H} \sim \mathcal{F}_{\frac{p}{3}}(G')$ with $S = A(x)$, $a = 2$, and $T = B(y)$, we can conclude that $A'(x) \cap B(y) \neq \emptyset$ with probability at least $1 - 3n^{-\log \log n}$, and, thus, x reaches y with at least that probability. Hence, after a union bound over all ordered pairs, we get that all nodes in Z reach each other with probability at least $1 - 3n^{-\log \log n + 2}$. Therefore, \mathcal{G} has a temporally connected component of size at least $n - \frac{2n}{\log \log n} = n - o(n)$ a.a.s. ◀

5 Sharp Threshold for Giant Closed Connected Component

In this section we report on the result that $\frac{\log n}{n}$ is also a sharp threshold for the existence of a giant *closed* connected component. We first sketch the general proof idea; the formal proof given subsequently is based upon a lemma proven in Appendix F of the full version.

▶ **Theorem 5.1** (Main Result for Closed Components). *The function $\frac{\log n}{n}$ is a sharp threshold for Giant Closed Connected Component. More precisely, there exists a function $\varepsilon(n) \in o\left(\frac{\log n}{n}\right)$, such that the size of a largest closed temporally connected component in $\mathcal{G} \sim \mathcal{F}_{n,p}$ is*

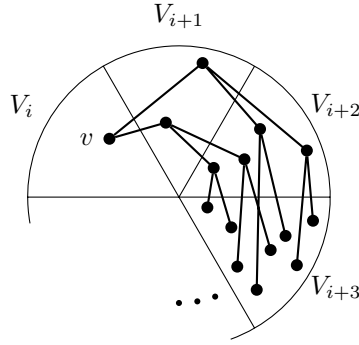
- (i) $o(n)$ a.a.s., if $p < \frac{\log n}{n} - \varepsilon(n)$; and
- (ii) $n - o(n)$ a.a.s., if $p > \frac{\log n}{n} + \varepsilon(n)$.

Proof Sketch. The lower bound of Theorem 5.1 is obviously a trivial consequence of the lower bound in Theorem 4.1. Thus, it remains to prove the upper bound. We start from our strategy of splitting the time into three intervals. We do not need to make any changes to our approach in the middle one (Phase 2), which previously required the most effort. However, we now need to do additional work in the first and last interval (Phase 1), which is the main technical contribution of this part. Recall that in the proof of Theorem 4.3, we only required that $n - o(n)$ vertices can all reach (resp. be reached by) at least $n^{1/3} \log n$ vertices within each of the three intervals from Figure 2. Now, we will need to prove that there exists a set V' of $n - o(n)$ vertices, such that every vertex in this set can reach (resp. be reached by) at least $n^{1/3} \log n$ vertices *via temporal paths that use only vertices in V'* . Once this is achieved, we can use the same approach as in the case of open components for Phase 2.

In order to obtain the set of vertices V' mentioned above, we have to insert an additional Phase 1.1, which looks only at a short time interval I_1 at the very beginning (and symmetrically I_5 at the very end). The purpose of this new phase is to “bootstrap” the closed component by identifying a set V' of $n' = n - o(n)$ vertices, which each reach at least *poly-logarithmically* many vertices by paths that are contained in V' . Lemma 5.2 formalizes this result.

A technical difficulty in Phase 1.1 is the need to control possible cascading effects, where removing low-degree vertices from the graph can cause further vertices to become low-degree vertices, etc. We overcome this difficulty by partitioning the vertices into sectors V_1, \dots, V_C and removing vertices from each sector V_i solely on the base of whether they have too few neighbors in the *next sector* V_{i+1} . This ensures that the sets of vertices removed from each sector are determined independently of each other. On this base, we are then able to prove that no cascading effects occur a.a.s. Subsequently, we show that after these removals, every remaining vertex can reach a poly-logarithmic number of others by considering *clocked* paths, which essentially march in lockstep, traversing the sectors in circular order (see Figure 4).

Subsequently, in Phase 1.2, we reveal edges that appear during I_2 or I_4 . We use the foremost forest technique developed earlier to show that, conditioned on the edges revealed in Phase 1.1, for every vertex v in V' the poly-logarithmic set of vertices reached by v during I_1 reaches *polynomially many* (by which we mean n^p for some fixed $p < 1$) vertices during I_2 . (Similarly, the set of vertices that reach v during I_5 is reached by polynomially many vertices during I_4 .) ◀



■ **Figure 4** Example of a temporal tree formed by clocked paths starting at a vertex $v \in V_i$. By restricting the edges used between sectors V_{i+j} and V_{i+j+1} to an appropriate time interval I_{j-i} , we ensure that the time labels of all these paths are monotonically clockwise increasing.

► **Lemma 5.2.** *Let $C \geq 3$, $\frac{1}{2} < \gamma < \alpha < 1$, and let $\mathcal{G} \sim \mathcal{F}_{n,p}$, where $p = 2C^2 \frac{(\log n)^\alpha}{n}$. Then a.a.s. \mathcal{G} contains a set V' of $n - o(n)$ vertices, such that, denoting $\mathcal{G}' := \mathcal{G}[V']$, every vertex in V' reaches at least $(\log n)^{(C-3)\gamma}$ vertices in $\mathcal{G}'_{[0,p/2]}$ and is reached by at least the same number of vertices in $\mathcal{G}'_{[p/2,p]}$.*

In the rest of this section we prove Theorem 5.1 using the above lemma, whose proof is found in Appendix F of the full version.

Proof of Theorem 5.1. Let $\gamma = 0.7$, $\alpha = 0.9$, $C = 30$, and let $n'(n) \in n - o(n)$ be the size of the vertex set guaranteed by Lemma 5.2. Set $\varepsilon_1 := C^2 \frac{(\log n)^\alpha}{n} \in o\left(\frac{\log n}{n}\right)$, $\varepsilon_2 := 4 \frac{\log \log n'}{\log n'} \in o(1)$, and $\varepsilon_3 := \frac{1}{3 \log \log n} \in o(1)$. Set also $p_1 := \left(\frac{1}{3} + \varepsilon_2\right) \frac{\log n'}{n'}$ and $p_2 := \left(\frac{1}{3} + \varepsilon_3\right) \frac{\log n'}{n'}$. Finally, define $p := 2\varepsilon_1 + 2p_1 + p_2$, which is equal to $\frac{\log n}{n} + \varepsilon$ for some $\varepsilon \in o\left(\frac{\log n}{n}\right)$.

Let $\mathcal{G} \sim \mathcal{F}_{n,p}$. We split $[0, p]$ into a total of five intervals I_i , $i \in [5]$. The first and the last interval are short and each has length ε_1 , i.e., $I_1 = [0, \varepsilon_1]$ and $I_5 = [p - \varepsilon_1, p]$. The three middle intervals are long and have lengths p_1, p_2 , and p_1 , respectively, i.e., $I_2 = [\varepsilon_1, \varepsilon_1 + p_1]$, $I_3 = [\varepsilon_1 + p_1, \varepsilon_1 + p_1 + p_2]$, $I_4 = [p - (\varepsilon_1 + p_1), p - \varepsilon_1]$. We will reveal the edges of the graph in three phases (Phase 1.1, Phase 1.2, and Phase 2), as was graphically summarized in Figure 3 in the introduction, and in each phase we condition on the edges revealed in the previous

phases. In Phase 1.1 we reveal edges in the intervals I_1 and I_5 and apply Lemma 5.2 to identify a large set of nodes V' , each of which can reach poly-logarithmically many vertices in V' during the first interval and can be reached by poly-logarithmically many vertices in V' during the last interval via temporal paths that use only nodes from V' . In the subsequent phases we restrict our attention to the subgraph induced by V' , which is the target giant closed connected component. In Phase 1.2, we reveal edges appearing in the intervals I_2 and I_4 . Because in Phase 1.1 a.a.s. we revealed poly-logarithmic number of edges for every vertex, we can use Lemma C.10 to argue that for every vertex $v \in V'$ the poly-logarithmic set of vertices reached by v during I_1 can reach polynomially many vertices during I_2 . Similarly, during I_4 polynomially many vertices can reach the poly-logarithmic set of vertices that reach v during I_5 . The main outcome of this phase is that every vertex in $v \in V'$ reaches polynomially many vertices in V' during $I_1 \cup I_2$ and is reached by at least as many vertices in $I_4 \cup I_5$. Finally, in Phase 2, because in the previous phases a.a.s. at most poly-logarithmically many edges were revealed for every vertex, we can apply Corollary 4.7 to prove that for each ordered pair of vertices $u, w \in V'$ the set of vertices reached by u during $I_1 \cup I_2$ can reach during I_3 the set of vertices that reach w during $I_4 \cup I_5$, implying that u can reach w during $[0, p]$. We now proceed with the formal details.

Phase 1.1. Let \mathcal{G}_1 be the temporal subgraph of \mathcal{G} formed by the edges with time labels in the intervals $I_1 \cup I_5$. Note that, up to shifting the time labels in the interval I_5 by $p - 2\varepsilon_1$, \mathcal{G}_1 is distributed according to $\mathcal{F}_{n, 2\varepsilon_1}$. Thus, by Lemma 5.2, a.a.s. there is a set $V' \subseteq V(\mathcal{G})$ containing n' vertices such that, denoting $\mathcal{G}' := \mathcal{G}[V']$, every vertex $v \in V'$ reaches a set $R_1(v)$ of at least $(\log n)^{(C-3)\gamma}$ vertices in \mathcal{G}'_{I_1} and is reached by a set $R'_1(v)$ of at least $(\log n)^{(C-3)\gamma}$ vertices in \mathcal{G}'_{I_5} .

Phase 1.2. Let G_1 be the underlying graph of \mathcal{G}_1 . Since G_1 is distributed as an Erdős-Rényi graph $G_{n, 2\varepsilon_1}$, similarly to Observation 4.6, we have that $\Delta(G_1) < 4 \log n$ a.a.s. Hence, in the graph $G'_2 = \left(V', \binom{V'}{2} \setminus E(G_1) \right)$ the minimum degree is at least $n' - 4 \log n \geq n' - (\log n')^2$. Observe that, up to shifting time labels, $\mathcal{G}'_{I_2} \sim \mathcal{F}_{p_1}(G'_2)$ when conditioning on the knowledge about all edges seen in $I_1 \cup I_5$. Therefore, since $|R_1(v)| \geq (\log n')^{13}$ for every vertex $v \in V'$, by applying Lemma C.10 to \mathcal{G}'_{I_2} and $R_1(v)$ (with parameter $z = 1/3 + \frac{\log \log n'}{\log n'}$), we conclude that the vertices in $R_1(v)$ reach in \mathcal{G}'_{I_2} at least $r := (n')^{\frac{1}{3} + \frac{\log \log n'}{\log n'}} = (n')^{1/3} \log n'$ vertices with probability at least $1 - 2(n')^{-\log \log n'}$. By the union bound, we have that with probability at least $1 - 2(n')^{1-\log \log n'} \in 1 - o(1)$, every vertex $v \in V'$ can reach in $\mathcal{G}'_{I_1 \cup I_2}$ a set $R_2(v)$ of at least r vertices. Symmetrically, with probability at least $1 - 2(n')^{1-\log \log n'} \in 1 - o(1)$, every vertex $v \in V'$ is reached in $\mathcal{G}'_{I_4 \cup I_5}$ by a set $R'_2(v)$ of at least r vertices.

Phase 2. Let G'_3 be the static graph defined by the vertex set V' and all edges appearing in $\mathcal{G}'_{I_1 \cup I_2}$ and $\mathcal{G}'_{I_4 \cup I_5}$. As in Phase 1.2, we can argue that the maximum degree of G'_3 is at most $4 \log n'$ a.a.s., and therefore the minimum degree of the graph $G'_4 = \left(V', \binom{V'}{2} \setminus E(G'_3) \right)$ is at least $n' - (\log n')^2$. Hence, up to shifting time labels, \mathcal{G}'_{I_3} is distributed according to $\mathcal{F}_{p_2}(G'_4)$ when conditioned on the knowledge of all edges revealed in $I_1 \cup I_2 \cup I_4 \cup I_5$. Thus, by Corollary 4.7, a given set of at least $(n')^{1/3} \log n'$ vertices in \mathcal{G}'_{I_3} can reach another given set of at least as many vertices with probability at least $1 - 3(n')^{-\log \log n'}$. Applying this to all ordered pairs of sets $(R_2(v), R'_2(w))$, $v, w \in V'$ and using the union bound, we conclude that the probability that all these pairs of sets reach each other in \mathcal{G}'_{I_3} is at least $1 - 3(n')^{2-\log \log n'} \in 1 - o(1)$.

Putting all together, we conclude that a.a.s. in $\mathcal{G}' = \mathcal{G}[V']$ any vertex reaches every other vertex. Thus, V' is, as desired, a giant closed connected component. \blacktriangleleft

References

- 1 Eleni C. Akrida, Leszek Gasieniec, George B. Mertzios, and Paul G. Spirakis. The complexity of optimal design of temporally connected graphs. *Theory of Computing Systems*, 61(3):907–944, 2017. doi:10.1007/s00224-017-9757-x.
- 2 Eleni C. Akrida, George B. Mertzios, Sotiris E. Nikolettseas, Christoforos L. Raptopoulos, Paul G. Spirakis, and Viktor Zamaraev. How fast can we reach a target vertex in stochastic temporal graphs? *Journal of Computer and System Sciences*, 114:65–83, 2020. doi:10.1016/j.jcss.2020.05.005.
- 3 Albert-László Barabási. *Network Science*. Cambridge University Press, 2016.
- 4 Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999. doi:10.1126/science.286.5439.509.
- 5 Hervé Baumann, Pierluigi Crescenzi, and Pierre Fraigniaud. Parsimonious flooding in dynamic graphs. *Distributed Computing*, 24(1):31–44, 2011. doi:10.1007/s00446-011-0133-9.
- 6 Sandeep Bhadra and Afonso Ferreira. Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks. In *Proceedings of the 2nd International Conference on Ad-Hoc Networks and Wireless (ADHOC-NOW)*, pages 259–270, 2003. doi:10.1007/978-3-540-39611-6_23.
- 7 Béla Bollobás. *Random Graphs*. Cambridge University Press, 2 edition, 2001. doi:10.1017/CB09780511814068.
- 8 Arnaud Casteigts, Anne-Sophie Himmel, Hendrik Molter, and Philipp Zschoche. Finding temporal paths under waiting time constraints. *Algorithmica*, 83(9):2754–2802, 2021. doi:10.1007/s00453-021-00831-w.
- 9 Arnaud Casteigts, Michael Raskin, Malte Renken, and Viktor Zamaraev. Sharp thresholds in random simple temporal graphs. In *Proceedings of the 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 319–326, 2021. doi:10.1109/FOCS52979.2021.00040.
- 10 Augustin Chaintreau, Abderrahmen Mtibaa, Laurent Massoulie, and Christophe Diot. The diameter of opportunistic mobile networks. In *Proceedings of the 3rd International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, pages 1–12, 2007. doi:10.1145/1364654.1364670.
- 11 Andrea Clementi, Angelo Monti, Francesco Pasquale, and Riccardo Silvestri. Broadcasting in dynamic radio networks. *Journal of Computer and System Sciences*, 75(4):213–230, 2009. doi:10.1016/J.JCSS.2008.10.004.
- 12 Andrea E. F. Clementi, Claudio Macci, Angelo Monti, Francesco Pasquale, and Riccardo Silvestri. Flooding time of edge-markovian evolving graphs. *SIAM Journal on Discrete Mathematics*, 24(4):1694–1712, 2010. doi:10.1137/090756053.
- 13 Alessio Conte, Pierluigi Crescenzi, Andrea Marino, and Giulia Punzi. Enumeration of s-d separators in DAGs with application to reliability analysis in temporal graphs. In *Proceedings of the 45th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 25:1–25:14, 2020. doi:10.4230/LIPIcs.MFCS.2020.25.
- 14 Jessica Enright, Kitty Meeks, George B. Mertzios, and Viktor Zamaraev. Deleting edges to restrict the size of an epidemic in temporal networks. *Journal of Computer and System Sciences*, 119:60–77, 2021. doi:10.1016/j.jcss.2021.01.007.
- 15 Jessica Enright, Kitty Meeks, and Fiona Skerman. Assigning times to minimise reachability in temporal graphs. *Journal of Computer and System Sciences*, 115:169–186, 2021. doi:10.1016/j.jcss.2020.08.001.
- 16 P. Erdős and A. Rényi. On random graphs I. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959. URL: https://publi.math.unideb.hu/load_doc.php?p=2769&t=pap.
- 17 P. Erdős and A. Rényi. On the strength of connectedness of a random graph. *Acta Mathematica Academiae Scientiarum Hungarica*, 12(1):261–267, 1964. doi:10.1007/BF02066689.
- 18 P. Erdős and A. Rényi. On the existence of a factor of degree one of a connected random graph. *Acta Mathematica Academiae Scientiarum Hungarica*, 17:359–368, 1966. doi:10.1007/BF01894879.

- 19 P. Erdős and A. Rényi. On the evolution of random graphs. *A Magyar Tudományos Akadémia. Matematikai Kutató Intézetének Közleményei*, 5:17–61, 1960. URL: http://www.renyi.hu/~p_erdos/1960-10.pdf.
- 20 Till Fluschnik, Hendrik Molter, Rolf Niedermeier, Malte Renken, and Philipp Zschoche. Temporal graph classes: A view through temporal separators. *Theoretical Computer Science*, 806:197–218, 2020. doi:10.1016/j.tcs.2019.03.031.
- 21 Alan Frieze and Michał Karoński. *Introduction to Random Graphs*. Cambridge University Press, 2015. doi:10.1017/CB09781316339831.
- 22 Emmanuel Godard and Dorian Mazaucic. Computing the dynamic diameter of non-deterministic dynamic networks is hard. In *Proceedings of the 10th International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics (ALGOSENSORS)*, pages 88–102, 2014. doi:10.1007/978-3-662-46018-4_6.
- 23 Peter Grindrod and Desmond J. Higham. Evolving graphs: dynamical models, inverse problems and propagation. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 466(2115):753–770, 2010. doi:10.1098/rspa.2009.0456.
- 24 Petter Holme and Jari Saramäki. Temporal networks. *Physics Reports*, 519(3):97–125, 2012. doi:10.1016/j.physrep.2012.03.001.
- 25 Petter Holme and Jari Saramäki, editors. *Temporal Network Theory*. Springer, 2019. doi:10.1007/978-3-030-23495-9.
- 26 Richard M. Karp. The transitive closure of a random digraph. *Random Structures & Algorithms*, 1(1):73–93, 1990. doi:10.1002/rsa.3240010106.
- 27 David Kempe, Jon Kleinberg, and Amit Kumar. Connectivity and inference problems for temporal networks. *Journal of Computer and System Sciences*, 64(4):820–842, 2002. doi:10.1006/jcss.2002.1829.
- 28 Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. *Random Structures & Algorithms*, 6(2-3):161–180, 1995. doi:10.1002/RSA.3240060204.
- 29 Michael Molloy and Bruce Reed. The size of the giant component of a random graph with a given degree sequence. *Combinatorics, Probability and Computing*, 7(3):295–305, 1998. doi:10.1017/S0963548398003526.
- 30 M. E. J. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003. doi:10.1137/S003614450342480.
- 31 M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(2):026118, 2001. doi:10.1103/PhysRevE.64.026118.
- 32 Mark Newman. *Networks*. Oxford University Press, 2018. doi:10.1093/oso/9780198805090.001.0001.
- 33 John Tang, Ilias Leontiadis, Salvatore Scellato, Vincenzo Nicosia, Cecilia Mascolo, Mirco Musolesi, and Vito Latora. Applications of temporal graph metrics to real-world networks. In *Temporal Networks*, pages 135–159. Springer, 2013. doi:10.1007/978-3-642-36461-7_7.
- 34 D. Watts and S. Strogatz. Collective dynamics of ‘small-world’ networks. *Nature*, 393(6684):440–442, 1998. doi:10.1038/30918.
- 35 Xiao Zhang, Cristopher Moore, and Mark Newman. Random graph models for dynamic networks. *The European Physical Journal B*, 90(10):1–14, 2017. doi:10.1140/epjb/e2017-80122-8.