

# Models of Smoothing in Dynamic Networks

Uri Meir

Tel Aviv University, Israel

Ami Paz

Faculty of Computer Science, Universität Wien, Austria

Gregory Schwartzman

JAIST, Ishikawa, Japan

---

## Abstract

---

Smoothed analysis is a framework suggested for mediating gaps between worst-case and average-case complexities. In a recent work, Dinitz et al. [Distributed Computing, 2018] suggested to use smoothed analysis in order to study dynamic networks. Their aim was to explain the gaps between real-world networks that function well despite being dynamic, and the strong theoretical lower bounds for arbitrary networks. To this end, they introduced a basic model of smoothing in dynamic networks, where an adversary picks a sequence of graphs, representing the topology of the network over time, and then each of these graphs is slightly perturbed in a random manner.

The model suggested above is based on a per-round noise, and our aim in this work is to extend it to models of noise more suited for multiple rounds. This is motivated by long-lived networks, where the amount and location of noise may vary over time. To this end, we present several different models of noise. First, we extend the previous model to cases where the amount of noise is very small. Then, we move to more refined models, where the amount of noise can change between different rounds, e.g., as a function of the number of changes the network undergoes. We also study a model where the noise is not arbitrarily spread among the network, but focuses in each round in the areas where changes have occurred. Finally, we study the power of an adaptive adversary, who can choose its actions in accordance with the changes that have occurred so far. We use the flooding problem as a running case-study, presenting very different behaviors under the different models of noise, and analyze the flooding time in different models.

**2012 ACM Subject Classification** Theory of computation → Distributed computing models; Theory of computation → Dynamic graph algorithms

**Keywords and phrases** Distributed dynamic graph algorithms, Smoothed analysis, Flooding

**Digital Object Identifier** 10.4230/LIPIcs.DISC.2020.36

**Related Version** A full version of the paper is available at <https://arxiv.org/abs/2009.13006>.

**Funding** *Ami Paz*: Supported by the Austrian Science Fund (FWF): P 33775-N, Fast Algorithms for a Reactive Network Layer.

*Gregory Schwartzman*: This work was supported by JSPS Kakenhi Grant Number JP19K20216 and JP18H052.

**Acknowledgements** The authors are thankful to Seth Gilbert for interesting discussions.

## 1 Introduction

Distributed graph algorithms give a formal theoretical framework for studying networks. There is abundant literature on the topic for static networks, and recently, an increasing amount of work on dynamic networks, under the title of distributed dynamic graph algorithms. Yet, our understanding of the interconnections between the theoretical models and real world networks is unsatisfactory. That is, we have strong theoretical lower bounds for many distributed settings, both static and dynamic, yet real-world communication networks are functioning, and in a satisfactory manner.



© Uri Meir, Ami Paz, and Gregory Schwartzman;

licensed under Creative Commons License CC-BY

34th International Symposium on Distributed Computing (DISC 2020).

Editor: Hagit Attiya; Article No. 36; pp. 36:1–36:16

Leibniz International Proceedings in Informatics



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

One approach to bridging the gap between the strong theoretical lower bounds and the behaviour of real-world instances, is the idea of *smoothed analysis*. Smoothed analysis was first introduced by Spielman and Teng [18, 19], in an attempt to explain the fact that some problems admit strong theoretical lower bounds, but in practice are solved on a daily basis. The explanation smoothed analysis suggests for this gap, is that lower bounds are proved using very specific, pathological instances of a problem, which are highly unlikely to happen in practice. They support this idea by showing that some lower bound instances are extremely *fragile*, i.e., a small random perturbation turns a hard instance into an easy one. Spielman and Teng applied this idea to the simplex algorithm, and showed that, while requiring an exponential time in the *worst case*, if we apply a small random noise to our instance before executing the simplex algorithm on it, the running time becomes polynomial in expectation.

Smoothed analysis was first introduced to the distributed setting in the seminal work of Dinitz et al. [8], who considered distributed dynamic networks. These are networks that changes over time, and capture many real world systems, from Blockchain, through vehicle networks, and to peer-to-peer networks. They are modeled by a sequence of communication graphs, on which the algorithm needs to solve a problem despite the changes in the communication topology. Adapting the ideas of smoothed analysis in this setting is not an easy task. First, while the classical smoothed analysis is applied to a single input, in the dynamic setting we deal with an infinite sequence of communication rounds, on ever-changing graphs. And second, defining the correct perturbation which the input undergoes is more challenging, as the input is discrete, as opposed to the continuous input of the simplex algorithm, where Gaussian noise is a very natural candidate.

To address the above challenges, Dinitz et al. fix a *noise parameter*  $k > 0$ , and then, after the adversary picks an infinite sequence of graphs  $\{G_i\}$ , they derive a new series of graphs  $\{H_i\}$  by perturbing every  $G_i$  with an addition or deletion of roughly  $k$  random edges. Specifically, the perturbation ( $k$ -smoothing) is done by choosing uniformly at random a graph within Hamming distance at most  $k$  of  $G_i$  (i.e., a graph that is different from  $G_i$  by at most  $k$  edges), which also has some desired properties (say, connectivity). Note that in the typical case  $\Omega(k)$  edges are perturbed, as most of the graphs with Hamming distance at most  $k$  from  $G_i$  have Hamming distance  $\Omega(k)$  from it. Using this model, they analyze the problems of flooding, hitting time, and aggregation.

In this paper we present a study of *models of smoothing*, or put differently, a study of models of noise in dynamic networks. To this end, we focus on one of the three fundamental problems presented in the previous work – the flooding problem. In this problem, a source vertex has a message it wishes to disseminate to all other vertices in the network. In each round, every vertex which has the message forwards it to all of its neighbors, until all vertices are informed. First, note that the problem is trivially solvable in static networks, in diameter time. Second, in the dynamic setting, it is easy to construct a sequence of graph where flooding takes  $\Omega(n)$  rounds, even if each of them has a small diameter. It is thus not surprising that adding random links between vertices may accelerate the flooding process, and indeed, it was shown in [8] that in  $k$ -smoothed dynamic networks, the complexity of flooding drops to  $\tilde{O}(n^{2/3}/k^{1/3})$  (where the  $\tilde{\cdot}$  sign hides polylog  $n$  factors). Note the huge improvement, from  $\Omega(n)$  to  $\tilde{O}(n^{2/3})$ , even for  $k = 1$ .

## 1.1 Motivation

The significant step made by Dinitz et al. in introducing smoothed analysis to the distributed dynamic environment left many fascinating questions unanswered. One natural direction of research has to do with applying smoothed analysis for a variety of problems, but here, we

take a different approach. In an attempt to understand the essential properties of smoothed analysis in our setting, we focus on questions related to the very basic concept of smoothing and study several models of noise. We outline some of the related questions below.

**The curious gap between  $k = 0$  and  $k = 1$ .** Dinitz et al. show a tight bound of  $\tilde{\Theta}(n^{2/3}/k^{1/3})$  for the problem of flooding in a dynamic networks with noise  $k$ . That is, as  $k$  decreases, flooding becomes harder, but only up to  $\tilde{\Theta}(n^{2/3})$  for a constant  $k$ . However, when there is no noise at all, a *linear*, in  $n$ , number of rounds is required. That is, there is a curious gap between just a tiny amount of noise and no noise at all, which stands in sharp contrast with the smooth change in the flooding time as a function of  $k$ , when  $k$  ranges from 1 and to  $\Omega(n)$ . One may wonder if there is a natural way to extend the model presented by Dinitz et al. to bridge this gap.

**An oblivious adversary.** The results of Dinitz et al. assume an oblivious adversary. That is, the evolution of the network must be decided by the adversary in advance, without taking into account the noise added by the smoothing process. It is natural to ask what is the power of an *adaptive* adversary compared to an oblivious one. As smoothed analysis aims to be the middle ground between worst-case and average-case analysis, it is very natural to consider the effect of noise on the strongest possible adversary, i.e., an adaptive adversary.

**Change-dependent noise.** The type of noise considered by Dinitz et al. remains constant between rounds, regardless of the topology or the actions of the adversary, which corresponds to “background noise” in the network. However, it is very natural to consider cases where the amount of noise itself changes over time. More specifically, if we think of a noise as an artifact of the changes the adversary performs to the system, and thus, it is natural to expect different amounts of noise if the adversary performs a single change, or changes the whole topology of the graph. Hence, a natural model of noise in a network is one where the added noise is *proportional* to the amount of change attempted by the adversary, that is,  $k$  is proportional to the Hamming distance between the current graph and the next one. A different, natural definition of a changes-dependent noise, is where the noise occurs only in the changing edges, and not all the graph.

## 1.2 Our results

To address the above points of interest, we prove upper and lower bounds, as summarized in Table 1. First, we show a natural extension of the previously presented noise model to *fractional* values of  $k$ . For  $k \geq 1$ , our model roughly coincides with the prior model and results, while for  $0 < k < 1$ , in our model a single random change occurs in the graph with probability  $k$ . In our model, we show a bound of  $\tilde{\Theta}(\min\{n^{2/3}/k^{1/3}, n\})$  for flooding, for values of  $k$  ranging from  $k = 0$  to  $k = \Theta(n)$ , even fractional – see theorems 6, 7, and 8. The flooding time thus has a clean and continuous behavior, even for the range  $0 < k < 1$  which was not studied beforehand, providing a very natural extension of the previous model.

Next, we focus our attention on an adaptive adversary, that can choose the next graph depending on the smoothing occurred in the present one. Here, we show that the added power indeed makes the adversary stronger, and yet, flooding can be solved in this case faster than in a setting where no smoothing is applied. Specifically, in theorems 9 and 10 we show that in this setting flooding can be done in  $\tilde{O}(n/\sqrt{k})$  rounds, and takes at least  $\tilde{\Omega}(n/k)$  rounds. This result presents a strict separation between the power of an adaptive and an oblivious adversaries.

■ **Table 1** Bounds on flooding time in different models of smoothing.

Model	Upper bound	Lower bound	Reference
Integer noise Non-adaptive adv.	$O(n^{2/3}(1/k)^{1/3} \log n)$	$\Omega(n^{2/3}/k^{1/3})$ , for $k \leq \sqrt{n}$	Dinitz et al. [8]
Fractional noise Non-adaptive adv.	$O(\min\{n, n^{2/3}(\log n/k)^{1/3}\})$	$\Omega(\min\{n, n/k, n^{2/3}/k^{1/3}\})$	Thm. 6, 7, 8
Fractional noise Adaptive adv.	$O(\min\{n, n\sqrt{\log n/k}\})$	$\Omega(\min\{n, n \log k/k\})$	Thm. 9, 10
Proportional noise Non-adaptive adv.	$O(n^{2/3} \cdot (D \log n/\epsilon)^{1/3})$		Thm. 12
Proportional noise Adaptive adv.	$O(n)$	$\Omega(n)$	Thm. 13
Targeted noise	$O(\min\{n, D \log n/\epsilon^{D^2}\})$	$\Omega(n)$ , for $D \in \Theta(\log n)$	Thm 14, 15

We then turn our attention to a different type of noise, and introduce two new models of *responsive noise* – noise which depends on the changes the adversary performs. The goal of studying responsive noise is to better understand systems where the noise is an artifact of the changes, and more changes induce more noise. We consider two, completely different cases: if only the *amount* of noise depends on the changes, then the system is less stable than in the prior models, and flooding can be delayed a lot by the adversary. On the other hand, if the same amount of noise is *targeted* at the changing links themselves, then the system remains stable, and flooding occurs much faster. In both models, our results surprisingly depend on a new attribute – the static diameter of the graph,  $D$ .

The first model of responsive noise we introduce is the *proportional noise* model, where the noise is randomly spread among the graph edges as before, and its amount is proportional to the number of proposed changes. We consider two types of adversaries in this setting – adaptive, and oblivious. Theorem 12 shows that  $\tilde{O}(\min\{n^{2/3}D^{1/3}/\epsilon^{1/3}, n\})$  rounds are sufficient for flooding in this model with an oblivious adversary. Here, the static diameter  $D$  comes into play, since the adversary can now force change-free rounds: if the adversary does not make any change, no noise occurs, the graph remains intact and no random “shortcut edges” are created. Current lower bounds for flooding with oblivious adversaries seem to require many changes, but in the proportional noise model this results in the addition of many random edges, thus speeding up the flooding process. In addition, the upper bound suggests that the static diameter  $D$  should come into play in the lower bounds, which is not the case in previous constructions. While we believe our upper bound is tight, proving lower bounds appears to be beyond the reach of current techniques.

In the proportional noise model with adaptive adversary, we encounter another interesting phenomenon. Adjusting the upper bound analysis in a straightforward manner gives the trivial upper bound of  $O(n)$ , and for a good reason: an adaptive adversary can slow down the flooding time all the way to  $\Omega(n)$ , as shown in Theorem 13. The adversary for this lower bound makes only a few changes in each round, and only in necessary spots (using its adaptivity). The fact that the noise is proportional boils down to almost no noise occurring, which allows the adversary to maintain control over the network.

The second model of responsive noise we study is that of a *targeted noise*. Here, the expected amount of noise applied is roughly the same as above, but only the edges that undergo changes are perturbed by the noise. More concretely, each change proposed by the

adversary does not happen with some probability  $\epsilon$ . The aim here is to model networks where every attempted change may have some probability of failure. In this setting, the case of an adaptive adversary seems more natural; however, we prove our upper bound for an adaptive adversary, and the lower bound for an oblivious one, thus handling the harder cases of both sides.

Our upper bound shows significant speedup in flooding on graphs with constant static diameter –  $O(\log n)$  rounds suffice for flooding with targeted noise. This phenomenon holds even for higher static diameters: Theorem 14 gives an upper bound of  $O((D \log_{1/\epsilon} n)/\epsilon^{D^2})$  rounds for flooding. This improves upon the trivial  $O(n)$  whenever  $D = O(\sqrt{\log_{1/\epsilon} n})$ . Finally, in Theorem 15 we show that for larger static diameter,  $D = \Theta(\log n)$ , the number of rounds required for flooding is  $\Omega(n)$ .

**Our techniques.** Analysing our new models of smoothing require a new and more global techniques. While we adopt the results of [8] for changes in a single round, our models introduce new technical challenges, as they require multiple-round based analysis.

The main technical challenge for proving upper bounds comes from the fact that one cannot even guarantee that noise occurs at every round, and when it does – the amount of noise is not fixed through the execution of the whole algorithm. This requires us to conduct a much more global analysis, taking into account sequences of rounds with a varying amount of noise, instead of analyzing a single round at a time. In several cases, the static diameter  $D$  appears as a parameter in our results and analysis: in our model, the adversary can force change-free rounds where no noise occurs, in which case flooding happens on a static graph.

In an attempt to study the exact limitations of our noise models, we present specifically crafted lower bound networks for each model. Note that in many models our adversary is adaptive, potentially making it much stronger. This makes our lower bound more strategy-based, and not merely a fixed instance of a dynamic graph. We revise the original flooding lower bound of Dinitz et al., in order to make it more applicable to other models. We present a more detailed and rigorous proof, that indeed implies tight bounds in most of our models, and specifically, when considering adaptive adversaries.

### 1.3 Related work

Smoothed analysis was introduced by Spielman and Teng [18,19] in relation to using pivoting rules in the simplex algorithm. Since, it have received much attention in sequential algorithm design; see, e.g., the survey in [19]. The first application of smoothed analysis to the distributed setting is due to Dinitz et al. [8], who apply it to the well studied problems of aggregation, flooding and hitting time in dynamic networks, as described above. Chatterjee et al. [5] considered the problem of a minimum spanning tree construction in a distributed setting of a synchronous network, where the smoothing is in the form of random links added in each round, which can be used for communication but not as a part of the tree.

The field of distributed algorithm design for dynamic networks has received considerable attention in recent years [16]. Most of the works considered models of adversarial changes [1, 2, 4, 9, 11, 12, 14, 15], while others considered random changes, or random interactions between the vertices [3, 6, 7, 10, 13]. Yet, as far as we are aware, only the two aforementioned works take the smoothed analysis approach in this context.

## 2 Preliminaries

### 2.1 Dynamic graphs

All graphs are of the form  $G = (V, E)$ , where  $V = [n] = \{1, \dots, n\}$  is the set of vertices. A dynamic graph  $\mathcal{H}$  is a sequence  $\mathcal{H} = (G_1, G_2, \dots)$  of graphs,  $G_i = (V, E_i)$  on the same set of vertices, which can be finite or infinite. The *diameter* of a (static) graph is the largest distance between a pair of vertices in it, and the *static diameter* of a dynamic graph  $\mathcal{H}$  is the minimal upper bound  $D$  on all the diameters of the graphs in the sequence  $\mathcal{H}$ .

We study synchronous distributed algorithms in this setting, where the time is split into rounds, and in round  $i$  the vertices communicate over the edges of the graph  $G_i$ .

Given two graphs  $G = (V, E)$  and  $G' = (V, E')$  on the same set of vertices, we denote  $G - G' = (V, E \setminus E')$ . We also denote  $G \oplus G' = (V, E \oplus E')$ , where  $E \oplus E'$  is the set of edges appearing in exactly one of the graphs. The size of set  $E \oplus E'$  is called the *Hamming distance between  $G$  and  $G'$* , and is denoted by  $|G \oplus G'|$ .

### 2.2 Models of noise

Our smoothed analysis is based on three models of noise, defined in this section.

For a single step, we recall the definition of  $t$ -smoothing [8], and add the new notion of  $\epsilon$ -smoothing, for a parameter  $0 < \epsilon < 1$ . At each step, we think of  $G_{\text{old}}$  as the current graph, and of  $G_{\text{adv}}$  as the future graph suggested by the adversary. The actual future graph,  $G_{\text{new}}$ , will be a modified version of  $G_{\text{adv}}$ , randomly chosen as a function of  $G_{\text{old}}$  and  $G_{\text{adv}}$ . In addition, we consider the set  $\mathcal{G}_{\text{allowed}}$  of *allowed graphs* for a specific problem. For flooding, this is just the set of all connected graphs.

► **Definition 1.** Let  $0 \leq \epsilon \leq 1$  and  $t \in \mathbb{N}$  be two parameters,  $\mathcal{G}_{\text{allowed}}$  a family of graphs, and  $G_{\text{old}}$  and  $G_{\text{adv}}$  two graphs in  $\mathcal{G}_{\text{allowed}}$ .

- A  $t$ -smoothing of  $G_{\text{adv}}$  is a graph  $G_{\text{new}}$  which is picked uniformly at random from all the graphs of  $\mathcal{G}_{\text{allowed}}$  that are at Hamming distance at most  $t$  from  $G_{\text{adv}}$ . The parameter  $t$  is called the noise parameter.
- An  $\epsilon$ -targeted smoothing of a graph  $G_{\text{adv}}$  with respect to  $G_{\text{old}}$  is a graph  $G_{\text{new}}$  which is constructed from  $G_{\text{adv}}$  by adding to it each edge of  $G_{\text{old}} - G_{\text{adv}}$  independently at random with probability  $\epsilon$ , and removing each edge of  $G_{\text{adv}} - G_{\text{old}}$  with the same probability. If the created graph is not in  $\mathcal{G}_{\text{allowed}}$ , the process is repeated.

We are now ready to define the three types of smoothing for dynamic graphs. The first extends the definition of [8] for non-integer values  $k \in \mathbb{R}_+$ , and the other two incorporate noise that depends on the latest modifications in the graph.

For a positive real number  $x$ , we define the random variable  $\text{round}(x)$ , which takes the value  $\lceil x \rceil$  with probability  $x - \lfloor x \rfloor$  (the fractional part of  $x$ ), and  $\lfloor x \rfloor$  otherwise.

► **Definition 2.** Let  $0 \leq \epsilon \leq 1$  and  $k \in \mathbb{R}_+$  be two parameters, and  $\mathcal{G}_{\text{allowed}}$  a family of graphs.

Let  $\mathcal{H} = (G_1, G_2, \dots)$  be a dynamic graph, i.e., sequence of “temporary” graphs. Let  $G'_0$  be a graph (if  $G'_0$  is not explicitly specified, we assume  $G'_0 = G_1$ ).

- A  $k$ -smoothed dynamic graph is the dynamic graph  $\mathcal{H}' = (G'_0, G'_1, G'_2, \dots)$  defined from  $\mathcal{H}$ , where for each round  $i > 0$ ,  $G'_i$  is the  $t_i$ -smoothing of  $G_i$ , where  $t_i = \text{round}(k)$ .
- An  $\epsilon$ -proportionally smoothed dynamic graph  $\mathcal{H}' = (G'_0, G'_1, G'_2, \dots)$  is defined from  $\mathcal{H}$ , where for each round  $i > 0$ ,  $G'_i$  is the  $t_i$ -smoothing of  $G_i$ , where  $t_i = \text{round}(\epsilon \cdot |G'_{i-1} \oplus G_i|)$ .

- An  $\epsilon$ -targeted smoothed dynamic graph is the dynamic graph  $\mathcal{H}' = (G'_0, G'_1, G'_2, \dots)$  defined iteratively from  $\mathcal{H}$ , where for each round  $i > 0$ ,  $G'_i$  is the  $\epsilon$ -targeted smoothing of  $G_i$  w.r.t.  $G'_{i-1}$ .

All the above definitions also have adaptive versions, where each  $\mathcal{H}$  and  $\mathcal{H}'$  are defined in parallel: the graph  $G_i$  is chosen by an adversary after the graphs  $G_0, G_1, G_2, \dots, G_{i-1}$  and  $G'_0, G'_1, G'_2, \dots, G'_{i-1}$  are already chosen, and then  $G'_i$  is defined as above. We use adaptive versions for the first two definitions.

## 2.3 Auxiliary lemmas

We use two basic lemmas, which help analyzing the probability of the noisy to add or remove certain edges from a given set of potential edges. These lemmas address a single round, we apply them with a different parameter in each round. The lemmas appeared as Lemmas 4.1 and 4.3 in [8], where they were used with the same parameter throughout the process.

► **Lemma 3.** *There exists a constant  $c_1 > 0$  such that the following holds. Let  $G_{\text{adv}} \in \mathcal{G}_{\text{allowed}}$  be a graph, and  $\emptyset \neq S \subseteq \binom{[n]}{2}$  a set of potential edges. Let  $t \in \mathbb{N}$  be an integer such that  $t \leq n/16$  and  $t \cdot |S| \leq n^2/2$ .*

*If  $G_{\text{new}}$  is a  $t$ -smoothing of  $G_{\text{adv}}$ , then the probability that  $G_{\text{new}}$  contains at least one edge from  $S$  is at least  $c_1 \cdot t |S| / n^2$ .*

► **Lemma 4.** *There exists a constant  $c_2 > 0$  such that the following holds.*

*Let  $G_{\text{adv}} \in \mathcal{G}_{\text{allowed}}$  be a graph. Let  $S \subseteq E_{\text{adv}}$  be a set of potential edges such that  $S \cap E_{\text{adv}} = \emptyset$ . Let  $t \in \mathbb{N}$ , such that  $t \leq n/16$ .*

*If  $G_{\text{new}}$  is an  $t$ -smoothing of  $G_{\text{adv}}$ , then the probability that  $G_{\text{new}}$  contains at least one edge from  $S$  is at most  $c_2 \cdot t |S| / n^2$ .*

## 2.4 Probability basics

In all our upper bound theorems, we show that flooding succeeds with high probability (w.h.p.), i.e., with probability at least  $1 - n^{-c}$  for a constant  $c$  of choice. For simplicity, we prove success probability  $1 - n^{-1}$ , but this can be amplified by increasing the flooding time by a multiplicative constant factor. Our lower bounds show that flooding does not succeed with probability more than  $1/2$ , so it definitely does not succeeds w.h.p.

We will also use the following Chernoff bound (see, e.g. [17]):

► **Lemma 5.** *Suppose  $X_1, \dots, X_n$  are independent Bernoulli random variables, with  $\mathbb{E}[X_i] = \mu$  for every  $i$ . Then for any  $0 \leq \delta \leq 1$ :*

$$\Pr \left[ \sum_{i=1}^n X_i \leq (1 - \delta)\mu n \right] \leq e^{-\delta^2 \mu n / 2}.$$

We usually apply this bound with  $\delta = 0.9$ .

# 3 Flooding in Dynamic Networks

## 3.1 Fractional amount of noise

We address an interesting phenomenon which occurs in [8]: by merely adding a single noisy edge per round, the flooding process dramatically speeds up from  $\Theta(n)$  in the worst case, to only  $\Theta(n^{2/3})$  in the worst case. To explain this gap, we present a natural notion of *fractional*

noise: if the amount of noise  $k$  is not an integer, we use an integer close to  $k$ , using the function  $\text{round}(k)$ . An easy assertion of the result in [8], is that whenever  $k > 1$ , the analysis does not change by much, and the same asymptotic bound holds. This leaves open the big gap between  $k = 0$  (no noise) and  $k = 1$ . We address this question in the next two theorems, showing a smooth transition in the worst-case flooding time between these two values of  $k$ .

Next, we revise the upper bound in [8] to handling fractional values of  $k$  and values smaller than 1. In addition, we show a somewhat cleaner argument, that carries three additional advantages: (1) It can be easily extended to adaptive adversaries; (2) It extends well to other models, as seen in subsequent sections; (3) It decreases the multiplicative logarithmic factor, dropping the asymptotic gap between upper and lower bound to only  $O(\log^{1/3}(n))$ . Hence, the proof of the next theorem serves as a prototype for later proofs, of upper bounds for adaptive adversaries and proportional noise.

► **Theorem 6.** *Fix a real number  $0 < k \leq n/16$ . For any  $k$ -smoothed dynamic graph, flooding takes  $O(\min\{n, n^{2/3}(\log n/k)^{1/3}\})$  rounds, w.h.p.*

Note that an  $n$ -round upper bound is trivial (using the connectivity guarantee), and that whenever  $k \leq c \cdot \log n/n$  for some constant  $c$ , the term  $n$  is the smaller one, and we need not prove anything more.<sup>1</sup> We therefore focus on the event of  $k > c \cdot \log n/n$ , and for this case we show an upper bound of  $O(n^{2/3}(\log n/k)^{1/3})$ .

**Proof.** Fix  $u$  to be the starting vertex, and  $v$  to be some other vertex. We show that within  $R = 3r$  rounds, the message has been flooded to  $v$  with constant probability (over the random noise). We split the  $3r$  rounds into three phases as follows.

1. First  $r$  rounds, for spreading the message to a large set of vertices we call  $U$ .
2. Next  $r$  rounds, for transferring the message from the set  $U$  to another large set  $V$  to be defined next.
3. Last  $r$  rounds, for “converging” and passing the message from the potential set  $V$  to the specific vertex  $v$ .

We now quantify the message flooding in the three phases.

**After the first  $r$  rounds, the message has reached many vertices.** Let  $U_i$  denote the set of vertices that have been flooded at time  $i$ . Using merely the fact that the graph is connected at all times, and assuming the flooding process is not over, we know that at each round a new vertex receives the message, implying  $|U_{i+1}| \geq |U_i| + 1$ . Hence,  $|U_r| \geq r$  (with probability 1).

**In the last  $r$  rounds, many vertices can potentially flood the message to  $v$ .** We denote by  $V_i$  the set of vertices from which  $v$  can receive the message within the last  $i$  rounds ( $R - i + 1, \dots, R$ ). Formally, fix the sequence of graphs chosen by the oblivious adversary, and apply the random noise on these graphs to attain the last  $r$  graphs of our *smoothed* dynamic graph. Define  $V_0 = \{v\}$ , and for  $i > 0$ , define  $V_i$  as the union of  $V_{i-1}$  with the neighbors of it in  $G_{R-i+1}$ . That is,  $V_i$  is defined analogously to  $U_i$  but with the opposite order of graphs. We point out that we are dealing with a stochastic process: each  $V_i$  is a random variable that depends on the noise in the last  $i$  rounds. Still, the connectivity guarantees that  $|V_{i+1}| \geq |V_i| + 1$ . Therefore, we have  $|V_r| \geq r$  (with probability 1).<sup>2</sup>

<sup>1</sup> This is rather intuitive, as if  $k$  is very small (e.g.,  $k \leq c/n$ ), when considering  $\delta n$  rounds for small enough  $\delta$ , no noise occurs. It is known that when no noise occurs, a linear amount of rounds is required for flooding. This intuition is formalized in the lower bound later in this section.

<sup>2</sup> Note that here we strongly rely on the obliviousness of the adversary: with an adaptive adversary, one cannot define  $V_r$  properly before the end of the execution of the algorithm, as the adversary’s choices



**The middle rounds.** The above processes define two randomly chosen sets,  $U_r$  and  $V_r$ , each of size at least  $r$ . If  $U_r \cap V_r \neq \emptyset$ , then we are done as  $v$  is guaranteed to receive the message even if we ignore the middle rounds. Otherwise, we consider the set  $S = U_r \times V_r$  of potential edges, and show that one of them belongs to our smoothed dynamic graph in at least one of the middle graphs, with probability at least  $1 - n^{-2}$ , for the right value of  $r$ .

Let us consider separately the two cases:  $k \geq 1$  (note that non-integer  $k$  was not handled before), and the case  $0 < k < 1$ .

**The case of  $k \geq 1$ .** In this case, we essentially claim the following: we are guaranteed to have either  $\lfloor k \rfloor$  noise or  $\lceil k \rceil$  at each and every round. Applying Lemma 3 for each such round, the probability of not adding any edge from  $S$  is at most

$$(1 - c_1 \lfloor k \rfloor |S|/n^2) \leq (1 - c_1 k r^2/2n^2),$$

where the inequality follows from  $\lfloor k \rfloor \geq k/2$  and  $|S| = |U_r| \cdot |V_r| \geq r^2$ . Thus, the probability of not adding any edge from  $S$  in any of these  $r$  noisy rounds is at most

$$(1 - c_1 k r^2/2n^2)^r \leq e^{-c_1 r^3 k/(2n^2)},$$

which is upper bounded by  $n^{-2}$ , whenever  $r \geq n^{2/3} \cdot [(4 \log n)/(c_1 k)]^{1/3}$ .

**The case of  $0 < k < 1$ .** Note that here we can no longer use  $\lfloor k \rfloor \geq k/2$ , and so we turn to a somewhat more global analysis which guarantees that “enough” rounds produce a (single) noisy edge: at each round we essentially add a noisy edge with probability  $k < 1$ , and otherwise do nothing. Since we have  $r$  rounds, and in each of them noise occurs independently, we can use a standard Chernoff bound to say that with all but probability at most  $e^{-0.4kr}$ , we have  $(k/10)r$  rounds in which a noisy edge was added.<sup>3</sup> We later bound this term. For each of the  $(k/10)r$  rounds we again apply Lemma 3 to claim that no edge from  $S$  was added, with probability at most

$$(1 - c_1 |S|/n^2) \leq (1 - c_1 r^2/n^2),$$

where the inequality follows again from  $|S| = |U_r| \cdot |V_r| \geq r^2$ . Thus, the probability of not adding any edge from  $S$  in any of these  $(k/10)r$  noisy rounds is upper bounded by

$$(1 - c_1 r^2/n^2)^{(k/10)r} \leq e^{-c_1 k r^3/(10n^2)},$$

which is upper bounded by  $1/(2n^2)$  whenever  $r \geq n^{2/3} \cdot [(10 \log n)/(c_1 k)]^{1/3}$ .

Recalling that we only deal with the case  $k \geq c \cdot \log n/n$ , choosing the constant  $c$  according to the constant  $c_1$ , we also get  $r \geq 10 \log n/k$ . This allows us to bound the error of the Chernoff inequality:  $e^{-0.4kr} \leq e^{-3 \log n} \leq 1/(2n^2)$ . Union bounding on both possible errors, we know that with probability at least  $1 - n^{-2}$  the vertex  $v$  is informed after  $3r$  rounds.

To conclude, we use  $r = n^{2/3} \cdot [(10 \log n)/(c_1 k)]^{1/3}$ . For any value  $k \geq c \cdot \log n/n$ , and for any realization of  $U_r$  and  $V_r$ , the message has been passed to  $V_r$  within the  $r$  middle rounds, with probability at least  $1 - n^{-2}$ . So  $v$  received the message after  $R = 3r$  rounds with the same probability. Taking a union bound over all the vertices implies that  $R$  rounds are enough to flood to the whole network with probability at least  $1 - 1/n$ . ◀

---

were not yet made. The case of an adaptive adversary is discussed in the next section.

<sup>3</sup> The expectation over all rounds is  $kr$  noisy edges.

We also show a matching lower bound, restructuring the proof of the lower bound in [8]. We note that their proof actually states a lower bound of  $\Omega(\min\{n/k, n^{2/3}/k^{1/3}\})$  that apply to any  $k < n/16$  (and is indeed tight for  $k < O(\sqrt{n})$ ). We restructure the analysis of the lower bound to fit the different constructions given in the following sections, as follows.

First, we consider the first  $R$  rounds, for some parameter  $R$ , and show inductively that the following two main claims continue to hold for every round  $i < R$  with very high probability.

- Some pre-defined set of vertices stays uninformed
- Given the above, the *expected* number of informed vertices in total does not grow rapidly. The growth (in the expected number of informed vertices) has a multiplicative-additive progression, potentially becoming an exponential growth (with a small exponential).

We choose  $R$  such that the expected number of informed vertices is upper bounded by  $\delta n$ , and use Markov inequality to show that with high probability the flooding is not yet over after  $R$  rounds. We then apply a union bound over all the inductive steps, where each has a small probability to fail the inductive argument. Altogether, we show that with high probability, the flooding is not over after  $R$  rounds.

In the full version of this paper, we use this argument in order to prove the following extensions of the lower bound from [8] to non-integer values and to fractional values (where  $k < 1$ ).

► **Theorem 7.** *Fix  $1 \leq k \leq n/16$  (not necessarily an integer). For any  $k$ -smoothed dynamic graph, for the flooding process to succeed with probability at least  $1/2$ , it must run for  $\Omega(\min\{n/k, n^{2/3}/k^{1/3}\})$  rounds.*

In particular, whenever  $k = O(\sqrt{n})$ , the dominant term is  $\Omega(n^{2/3}/k^{1/3})$ , matching the upper bound (up to logarithmic factors).

► **Theorem 8.** *Fix  $0 < k < 1$ . For any  $k$ -smoothed dynamic graph, for the flooding process to succeed with probability at least  $1/2$ , it must run for  $\Omega(\min\{n, n^{2/3}/k^{1/3}\})$  rounds.*

### 3.2 Adaptive vs. oblivious adversary

Here we note that the results of [8] are for the case of oblivious (non-adaptive) adversary. We extend our results (in the more generalized, fractional noise regime) to the adaptive case, obtaining bounds that are different than the ones in [8]. Interestingly, our results show separation between adaptive and oblivious adversary for a wide range of network noise: for constant  $k$ , in particular, we get a separation for the adaptive case, where no constant amount of noise speeds up the flooding below  $\Omega(n)$ , and the oblivious case where  $k = \omega(1/n)$  already speeds the flooding to  $o(n)$  rounds.

► **Theorem 9.** *Fix  $0 < k \leq n/16$ , not necessarily an integer. For any  $k$ -smoothed adaptive dynamic graph, flooding takes  $O(\min\{n, n\sqrt{\log n/k}\})$  rounds, w.h.p.*

The proof follows an argument similar to the one of Theorem 6, where the process is broken to three phases:  $u$  spreads the message to  $U_r$ , which carry it to  $V_r$ , who are sure to deliver it to the destination  $v$ . The major difference here is that we can no longer rely on the last phase, since an adaptive adversary *sees* the informed vertices and is able to act accordingly. We thus turn to a statistical analysis of the second phase instead, where the informed set  $U_r$  delivers the message directly to  $v$ . The full proof appears in the full version of this paper.

When  $k$  is a constant, the above result is tight, which we prove by adjusting the oblivious dynamic network from Theorem 7 to use the adversary's adaptivity, as follows. The basic structure of the hard instance for flooding is achieved by roughly splitting the graph into

*informed* and *uninformed* vertices. In order to ensure low diameter, all the uninformed vertices are connected as a star, centered at a changing uninformed vertex called a *head*, which in turn connects them to a star composed of the informed vertices. The key idea in the analysis of this process, is that the head at each round should not become informed via noisy edges at an earlier round, as in this case it will immediately propagate the message to all the uninformed vertices, completing the flooding. An oblivious adversary picks a sequence of heads at the beginning of the process, and this invariant is kept since the probability of any of the selected heads to get informed too early is low. However, after roughly  $n^{2/3}$  rounds, the probability that a selected head is informed becomes too high. An adaptive adversary, on the other hand, can continue crafting a hard instance graph for a linear number of rounds – in each round, the adversary knows which vertices are uninformed and can pick one of them as the new head, thus overcoming this obstacle.

► **Theorem 10.** *Fix  $0 < k \leq n/16$ , not necessarily an integer. For any  $k$ -smoothed adaptive dynamic graph, for the flooding process to succeed with probability at least  $1/2$  it must run for  $\Omega(\min\{n, n \log k/k\})$  rounds.*

**Proof.** We start by formally defining a hard case for flooding with noise. The base to our definition is the *spooling graph*, which was defined by Dinitz et al. [8]: at time  $i$ , the edges are  $\{(j, i)\}_{j=1}^{i-1}$ , a star around the vertex  $i$  (this will be the informed spool) and edges  $\{(i+1, j)\}_{j=i+2}^n$ , a star around the vertex  $i+1$ , which will be the right spool, with vertex  $i+1$  crowned as the *head*. The spools are then connected using the additional edge  $(i, i+1)$ .

We define the *adaptive spooling graph*, as follows: at first  $E_1 = ([n] \setminus \{2\}) \times \{2\}$ . After the first iteration, 2 learns the message and is removed to the left (informed) spool. We denote for each round  $i$  the set of informed vertices at the end with  $I_i$ . We also use  $u_i$  for the lowest index of an uninformed vertex at the end of round  $i$ . Formally  $u_i = \min\{\bar{I}_i\}$ . We define the adaptive spooling graph at time  $i+1$  to have the edge set

$$E_{i+1} = \{1\} \times (I_i \setminus \{1\}) \cup \{(1, u_i)\} \cup \{u_i\} \times (\bar{I}_i \setminus \{u_i\}).$$

In this graph, essentially, 1 is connected to all already-informed vertices, and also to the next head  $u_i$ , which is connected to all other uninformed vertices. The static diameter stays 3 at each iteration, but now we are promised that at iteration  $i+1$ , the new head  $u_i$  is uninformed at the beginning of said round.

We next show that the *expected* number of informed players cannot grow by much.

▷ **Claim 11.** When taking expectation over the noise we add, we have  $\mathbb{E}[|I_{i+1}|] \leq (1 + c_2 k/n) \cdot \mathbb{E}[|I_i|] + 1$ .

Note that  $I_i$  does not depend on the noise of rounds  $i+1, \dots, R$ , and so the left side takes expectation over  $i+1$  rounds of noise, and the right side over  $i$  rounds.

Intuitively, the claim states that the expected growth in the number of *informed* vertices is bounded by an additive-multiplicative progression (i.e., at each step the amount grows multiplicatively and then a constant is added). This is true as the noisy edges induce a multiplicative growth, and connectivity forces that at least 1 additional vertex ( $u_i$  itself) receives the message. The proof of this claim can be found in the full version.

Using the claim, we analyze the progression of  $A_i = \mathbb{E}[|I_i|]$ , splitting to 2 cases:

1. For  $A_i \leq n/(c_2 k)$ , the additive term is larger, and so in this range  $A_{i+1} \leq A_i + 2$ .
  2. For  $A_i \geq n/(c_2 k)$ , the multiplicative term overcomes and we get  $A_{i+1} \leq (1 + 2c_2 k/n)A_i$ .
- Note that we also have  $A_{i+1} \geq A_i + 1$ , using connectivity, showing this bounds on the progression are rather tight.

We next split into cases: if  $k < 1/c_2$ , the additive term controls the process through  $\Theta(n)$  iterations, giving us  $A_{n/20} \leq n/10$ . Otherwise, the multiplicative term would come into play. We denote the number of additive rounds by  $r_0$ : the minimal index such that  $A_{r_0} \geq n/(c_2k)$ . Since  $A_0 = 1$  (only the vertex 1 knows the message at the beginning), and we have good bounds on the progression, we conclude  $r_0 = \Theta(n/k)$ .

Next, we allow additional  $r_1$  rounds in which the multiplicative term is dominant. We get

$$A_{r_0+r_1} \leq A_{r_0}(1 + 2c_2k/n)^{r_1} \leq \Theta(n/k)(1 + 2c_2k/n)^{r_1}.$$

Taking  $r_1 = \delta(n \log k/k)$  with small enough  $\delta$ , we have  $A_{r_0+r_1} = \mathbb{E}[|I_{r_0+r_1}|] \leq n/10$ .

In both cases, there is a round  $R$  with  $\mathbb{E}[|I_R|] \leq n/10$ . By Markov inequality, strictly less than  $n$  vertices are informed after  $R$  rounds, w.p. at least 0.9. Note that for the first case  $R = \Theta(n)$ , and for the second case  $R = r_0 + r_1 = O(n \log k/k)$ , concluding the proof. ◀

## 4 Responsive noise

In this section we consider *responsive* noise in the dynamic network, where some elements of the noise incurred in each round relate to the changes done at this round. We consider this model as complement to one discussed in the previous section: the parameter  $k$  of “noise per round” is fit to model some internal background noise in a system. However, in a more realistic model we would expect the noise to work in different patterns in times of major changes in the network, as opposed to rounds where no changes were made at all.

To this end, we introduce two variants of a responsive noise: one is that of *proportional* noise, and the other is *targeted* noise. In the first, the amount of noisy edges would relate to the amount of changes that occurred in the last step. In the second variant, we expect the noise to specifically “target” newly modified edges. This could relate to a somewhat limited adversary (in aspect of ability to change the graph).

In the responsive model, an interesting phenomenon occurs: an adversarial network can choose to stay still, in order to force a round where no noise occurs. For the flooding problem, this strength is limited: whenever the static diameter of each iteration is upper bounded by  $D$ , the waiting game can only last  $D - 1$  rounds. To that end, we show how this model affects the analysis of the upper bound, and yet again incorporate the new phenomenon described to devise a non-trivial lower bound.

### 4.1 Proportional noise

► **Theorem 12.** *Fix  $0 < \epsilon$ . For any  $\epsilon$ -proportionally smoothed dynamic graph with static diameter at most  $D$ . If no noise invokes more than  $n/16$  changes, the flooding process finishes after  $O(n^{2/3} \cdot (D \log n/\epsilon)^{1/3})$  rounds, w.h.p.*

The proof resembles the proof of Theorem 6 with three phases, which handles the “waiting game” mentioned above. Given static diameter  $D$ , an adversarial network can only stay intact for  $D - 1$  rounds, thus in the second phase at least  $1/D$  of the rounds introduce one changed edge (or more), inferring that w.h.p. noise is incurred in  $\Omega(\epsilon/D)$  rounds. The proof appears in the full version of this paper.

Adjusting the same argument for an *adaptive* network, using only two phases (same as Theorem 9), would give an upper bound of  $O(n \cdot \sqrt{D \log n/\epsilon})$ , which in this case does not improve upon the trivial  $O(n)$  promised by connectivity alone.

We continue to show that this is tight, and in the proportional noise model, the changes in the network can be so subtle that they would invoke little noise and the flooding would barely speed up at all, asymptotically. We note that the following construction uses constant  $D$ , where it is hardest to manipulate.

► **Theorem 13.** *Fix  $\epsilon \leq 1/5$ . There exists an  $\epsilon$ -proportionally smoothed dynamic graph with an adaptive adversary, where a flooding process must run for  $\Omega(n)$  rounds in order to succeed with probability at least  $1/2$ .*

In the current, adaptive model, the spooling graph no longer stands as a slow-flooding graph. Changing the center of the uninformed star takes  $\Omega(n)$  edge changes in early rounds, and the proportional noise invoked by these changes would be devastating (e.g., for  $\epsilon = \Omega(1)$ ).

Instead, in the proof of this theorem, which appears in the full version of our paper, the adversary relies on its adaptivity in order to maintain control over the network. This is done using a sequence of graphs that require only  $O(1)$  changes at each round.

We turn to a variant of the spooling graph, composed of two star graphs, one is made of informed vertices and one of uninformed vertices. The last vertex to be informed is the center of the uninformed vertices star, which remains the same vertex throughout the process. Connectivity is kept using a connector vertex, which changes each round. Every time a connector learns the message, it gets disconnected from the uninformed star moved to the informed star, and another vertex becomes the connector.

As before, we need to deal with noisy edges that allow random vertices to learn the message, even when these are not the designated connector vertices. Here, we utilize the adaptivity once more: every time this happens, we immediately remove the newly informed vertex from the uninformed star and add it to the informed one, keeping the center of the uninformed star ignorant of the message. The only bad case is when the center vertex of the uninformed star gets connected to an informed vertex at random, but this happens with very low probability.

The key observation in our analysis is that handling a randomly informed vertex only costs additional  $O(1)$  changes. and even combined with the planned changes to the network, one can upper bound the amount of necessary changes at each round. This creates a positive feedback loop: the adversary can maintain the desired structure of the graph, using very few changes, and provoking almost no noise.

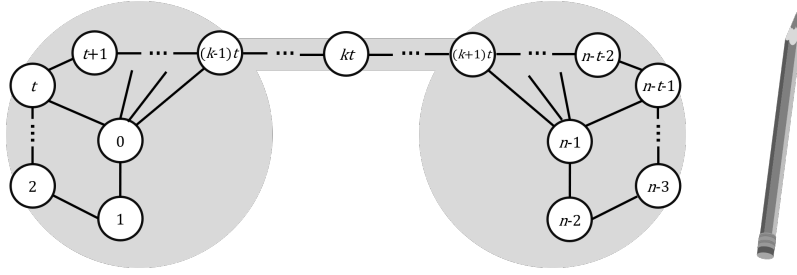
Finally, we note that each round at most one vertex learns the message due to a randomly added edge, in addition to the connector. Thus, only a constant amount of vertices (in fact, 2) can become informed in each round, implying the need for a linear number of rounds.

## 4.2 Targeted noise

For targeted noise this new phenomenon is surpassed by the limited ability of the adversary to make changes. We can think of targeted noise as some sort of “slow down”, as if the network repeatedly tries to modify some of the edges, eventually succeeding, but not before a number of rounds has passed. In this last model we show just how strong the waiting game can be: for a graph with constant static diameter (which makes the waiting game obsolete), flooding will take  $O(\log n)$  rounds, with high probability. However, for larger value of  $D$ , the same analysis would quickly fail. For  $D = \Theta(\sqrt{\log n})$  we get the trivial bound of  $O(n)$ . We finish by showing an explicit construction with static diameter  $D = \Theta(\log n)$  that relies strongly on the waiting game and admits a lower bound of  $\Omega(n)$  rounds.

► **Theorem 14.** *For any  $\epsilon$ -targeted smoothed dynamic graph with static diameter  $D$ , flooding can be done in  $O(D \log n / \epsilon^{D^2})$  rounds, w.h.p.*

Note that for  $D = o(\sqrt{\log_{1/\epsilon} n})$ , this improves upon the trivial fact that  $n$  rounds are enough (as at each round, one new vertex must be informed, since the graph is connected). Specifically, for a constant static diameter, we get  $O(\log n)$  rounds.



■ **Figure 1** The cassette graph,  $G_j^t$ , where  $j = kt$  and  $n$  is some multiple of  $t$ .

**Proof.** Fix the starting vertex  $v_0$  which is informed. First, we show that the probability of a vertex  $v \neq v_0$  staying uninformed after  $D$  rounds is at most  $1 - \epsilon^{D^2}$ .

Consider  $D$  consecutive graphs  $G_1, \dots, G_D$  in the smoothed graph. As the static diameter is  $D$  in every round, there exists a path  $P_v$  from  $v_0$  to  $v$  in  $G_1$ , whose length is at most  $D$ . Since we deal with targeted smoothing, each edge of  $P_v$  that exists in a graph  $G_i$  for some  $1 \leq i < D$  exists in  $G_{i+1}$  probability either 1 or  $\epsilon$ . So, for each such  $i$ , if all the edges of  $P_v$  exist in  $G_i$  then they all exist in  $G_{i+1}$  with probability at least  $\epsilon^D$ . Hence, the probability of the path  $P_v$  existing in all  $D$  graphs is at least  $(\epsilon^D)^D = \epsilon^{D^2}$ .

Fix a positive integer  $t$ , and consider  $t$  consecutive sequences of  $D$  graphs each. On each sequence, we apply the above claim, and conclude that the probability of  $v$  staying uninformed after these  $tD$  rounds is at most

$$(1 - \epsilon^{D^2})^t \leq e^{-t \cdot \epsilon^{D^2}}.$$

For  $t = (c+1) \log n \cdot (1/\epsilon)^{D^2}$  sequences, the probability of  $v$  not being informed after  $tD$  rounds is at most  $n^{-(c+1)}$ . A union bound over all  $n-1$  vertices that need to be informed implies that  $tD$  rounds suffice to fully flood the network with probability at least  $1 - n^{-c}$ . ◀

The above theorem implies that if the static diameter of all graphs in the sequence is small, roughly  $O(\sqrt{\log n})$ , flooding is fast. Next, we show that this is almost tight: if the diameter is in  $\Omega(\log n)$ , flooding cannot be done faster than in non-smoothed graphs.

► **Theorem 15.** *For every constant  $0 < \epsilon < 1$ , there is a value  $D \in \Theta(\log n)$  and an  $\epsilon$ -targeted smoothed dynamic graph such that with high probability, the diameter of the graph is  $D$  and flooding on it takes  $n-1$  rounds.*

In order to prove this theorem, we present the *dynamic cassette graph* (see Figure 1). Fix  $\epsilon$  and  $n$  as in the theorem statement, and let  $t = \lfloor c \log_{1/\epsilon} n \rfloor$  for a constant  $c$  of choice. The dynamic cassette graph on vertices  $V = \{0, \dots, n-1\}$  is the dynamic graph  $\mathcal{H} = \{G_1, \dots, G_n\}$ , where  $G_i = (V, E_i)$  is defined by

$$E_i = \{(j, j+1) \mid 0 \leq j < n-1\} \cup \{(0, jt) \mid 1 \leq j \leq \lfloor (i-1)/t \rfloor\} \cup \{(jt, n-1) \mid \lfloor (i-1)/t \rfloor + 2 \leq j \leq \lfloor (n-2)/t \rfloor\}.$$

This graph is the path on  $n$  vertices, with some additional edges connecting the first and last vertices to vertices in the set  $\{jt \mid 1 \leq j \leq (n-2)/t\}$ ; these will be referred to as *shortcut vertices*, and the additional edges to them, *shortcut edges*. At the first graph,  $G_1$ , all shortcut vertices but the first are connected to the last vertex,  $n-1$ . Then, one by one, the shortcut vertices disconnect from  $n-1$ , and soon after – connect to 0. At each time interval  $[(j-1)t+1, jt]$ , all the shortcut vertices with index strictly smaller than  $jt$  are connected to the vertex 0, and all those with index strictly higher than  $jt$  are connected to  $n-1$ .

Consider the *smoothed cassette graph*  $\mathcal{H}'$ , i.e., the  $\epsilon$ -targeted smoothed dynamic graph derived from  $\mathcal{H}$ . The dynamic graph  $\mathcal{H}'$  can be interpreted as undergoing the following process: during each time interval  $[(j-1)t+1, jt]$ , the adversary repeatedly tries to add a new edge  $(0, (j-1)t)$ , and remove the edge  $((j+1)t, n-1)$ . The targeted noise creates a slowdown that might prevent this from happening right away, yet for the right value of  $t$ , both changes indeed happen by the end of the time interval w.h.p. We state the following claim and direct the reader to the full version for its proof.

▷ **Claim 16.** For each  $2 \leq j \leq (n-2)/t$ , the smoothed graph  $G'_{jt}$  does not contain the edge  $(0, (j-1)t)$  with probability at most  $n^{-c}$ , and contains the edge  $((j+1)t, n-1)$  with the same probability.

If the edge  $(0, (j-1)t)$  exists in the smoothed graph  $G'_{jt}$  then it also exists in all later graphs,  $G_{j'}$  with  $j' > jt$ . Similarly, if the edge  $((j+1)t, n-1)$  does not exist in this graph, it also does not appear in later graphs. A union bound thus extends the last claim as follows.

▷ **Claim 17.** For each  $2 \leq j \leq (n-2)/t$ , the smoothed graph  $G'_{jt}$  and all later graphs contain the edge  $(0, (j-1)t)$  with probability at least  $1 - n^{-c+1}$ , and all these graphs do not contain the edge  $((j+1)t, n-1)$  with the same probability.

Using this claim, Theorem 15 can easily be proven.

**Proof of Theorem 15.** Consider the smoothed dynamic cassette graph  $\mathcal{H}'$ . We start by analyzing its diameter.

Let  $G'_{j'}$  be a graph in  $\mathcal{H}'$ , and pick a  $j$  such that  $jt \leq j' < (j+1)t$ . By Claim 17, the graph  $G'_{jt}$  and all later graphs contain the edge  $(0, (j-1)t)$  with probability at least  $1 - n^{-c+2}$ . In addition, all the graphs  $G_1, \dots, G_{(j+1)t-1}$  contain the edge  $((j+2)t, n-1)$  (and note that this is not a probabilistic claim).

The distance between every two shortcut vertices is  $t$ , so the distance from every vertex in the graph to the closest shortcut vertex is at most  $t/2$ . Each shortcut vertex is directly connected to either 0 or  $n-1$  w.h.p., except for  $jt$ , who is connected to both by a  $t+1$  path. Finally, between the vertices there is a path of length  $2t+2$ , through  $(j-1)t$  and  $(j+1)t$ . Let us bound the length of a path between two vertices  $i, i'$  (with upper bounds on each part): from  $i$  to its closest shortcut vertex ( $t/2$  hops), to 0 or  $n-1$  ( $t+1$  hop), maybe to the other vertex between 0 and  $n-1$  ( $2t+2$  hops), to the shortcut vertex closest to  $i'$  ( $t+1$  hop), and to  $i'$  ( $t/2$  hops). This sums to a path of length at most  $5t+4 = \Theta(\log n)$ . A more detailed analysis can reduce this to roughly  $3t$  hops.

For the flooding time, we use Claim 17 again, but now for the edges  $((j+1)t, n-1)$  that do not appear  $G'_{jt}$  and all later graphs w.h.p. A simple induction on  $j' = 0, \dots, n-1$  shows that after  $j'$  rounds, i.e. in graph  $G_{j'}$ , only vertices  $0, \dots, j'$  are informed. The base case is trivial. For the step, the only edges connecting informed vertices to uninformed vertices are  $(j', j'+1)$ , and edges from 0 to shortcut vertices, which have the form  $(0, tj)$  with  $tj \leq j'$  – this yields from the construction and always holds. The only other type of possible edges connecting informed and uninformed vertices are of the form  $(jt, n-1)$ , with  $jt \leq j'$ . However, the claim implies that w.h.p., by round  $j'$  none of these edges exist. Hence, before round  $n-1$  not all vertices are informed, and flooding takes  $n-1$  rounds w.h.p. ◀

---

**References**

---

- 1 John Augustine, Gopal Pandurangan, Peter Robinson, and Eli Upfal. Towards robust and efficient computation in dynamic peer-to-peer networks. In *SODA*, pages 551–569. SIAM, 2012.
- 2 Philipp Bamberger, Fabian Kuhn, and Yannic Maus. Local distributed algorithms in highly dynamic networks. In *IPDPS*, pages 33–42. IEEE, 2019.
- 3 Leran Cai, Thomas Sauerwald, and Luca Zanetti. Random walks on randomly evolving graphs. In *Structural Information and Communication Complexity - 27th International Colloquium, SIROCCO*, volume 12156 of *Lecture Notes in Computer Science*, pages 111–128. Springer, 2020.
- 4 Keren Censor-Hillel, Neta Dafni, Victor I. Kolobov, Ami Paz, and Gregory Schwartzman. Fast and simple deterministic algorithms for highly-dynamic networks. *CoRR*, abs/1901.04008, 2019.
- 5 Soumyottam Chatterjee, Gopal Pandurangan, and Nguyen Dinh Pham. Distributed MST: A smoothed analysis. In *ICDCN 2020: 21st International Conference on Distributed Computing and Networking*, pages 15:1–15:10. ACM, 2020. doi:10.1145/3369740.3369778.
- 6 Andrea E. F. Clementi, Riccardo Silvestri, and Luca Trevisan. Information spreading in dynamic graphs. *Distributed Computing*, 28(1):55–73, 2015.
- 7 Oksana Denysyuk and Luís E. T. Rodrigues. Random walks on evolving graphs with recurring topologies. In *DISC*, volume 8784 of *Lecture Notes in Computer Science*, pages 333–345. Springer, 2014.
- 8 Michael Dinitz, Jeremy T Fineman, Seth Gilbert, and Calvin Newport. Smoothed analysis of dynamic networks. *Distributed Computing*, 31(4):273–287, 2018.
- 9 Chinmoy Dutta, Gopal Pandurangan, Rajmohan Rajaraman, Zhifeng Sun, and Emanuele Viola. On the complexity of information spreading in dynamic networks. In *SODA*, pages 717–736. SIAM, 2013.
- 10 Leszek Gąsieniec and Grzegorz Stachowiak. Fast space optimal leader election in population protocols. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 2653–2667. SIAM, 2018.
- 11 Mohsen Ghaffari, Nancy A. Lynch, and Calvin C. Newport. The cost of radio network broadcast for different models of unreliable links. In *PODC*, pages 345–354. ACM, 2013.
- 12 Bernhard Haeupler and David R. Karger. Faster information dissemination in dynamic networks via network coding. In *PODC*, pages 381–390. ACM, 2011.
- 13 Dariusz R. Kowalski and Miguel A. Mosteiro. Polynomial counting in anonymous dynamic networks with applications to anonymous dynamic algebraic computations. *Journal of the ACM*, 67:1–17, 2020.
- 14 Fabian Kuhn, Nancy A. Lynch, and Rotem Oshman. Distributed computation in dynamic networks. In *STOC*, pages 513–522. ACM, 2010.
- 15 Fabian Kuhn, Yoram Moses, and Rotem Oshman. Coordinated consensus in dynamic networks. In *Proceedings of the 30th Annual ACM Symposium on Principles of Distributed Computing, PODC*, pages 1–10. ACM, 2011. doi:10.1145/1993806.1993808.
- 16 Othon Michail. An introduction to temporal graphs: An algorithmic perspective. *Internet Math.*, 12(4):239–280, 2016.
- 17 Michael Mitzenmacher and Eli Upfal. *Probability and computing — randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.
- 18 Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *J. ACM*, 51(3):385–463, 2004.
- 19 Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis: an attempt to explain the behavior of algorithms in practice. *Commun. ACM*, 52(10):76–84, 2009.