The Effect of Faults on Network Expansion

Amitabha Bagchi * Ankur Bhargava† Amitabh Chaudhary‡ David Eppstein§
Christian Scheideler¶

February 2, 2006

Abstract

We study the problem of how resilient networks are to node faults. Specifically, we investigate the question of how many faults a network can sustain and still contain a large (i.e., linear-sized) connected component with approximately the same expansion as the original fault-free network. We use a pruning technique that culls away those parts of the faulty network that have poor expansion. The faults may occur at random or be caused by an adversary. Our techniques apply in either case. In the adversarial setting, we prove that for every network with expansion α, a large connected component with basically the same expansion as the original network exists for up to a constant times $\alpha \cdot n$ faults. We show this result is tight in the sense that every graph G of size n and uniform expansion $\alpha(\cdot)$ can be broken into components of size $o(n)$ with $\omega(\alpha(n) \cdot n)$ faults.

Unlike the adversarial case, the expansion of a graph gives a very weak bound on its resilience to random faults. While it is the case, as before, that there are networks of uniform expansion $\Omega(1/\log n)$ that are not resilient against a fault probability of a constant times $1/\log n$, it is also observed that there are networks of uniform expansion $O(1/\sqrt{n})$ that are resilient against a constant fault probability. Thus, we introduce a different parameter, called the *span* of a graph, which gives us a more precise handle on the maximum fault probability. We use the span to show the first known results for the effect of random faults on the expansion of d-dimensional meshes.

1 Introduction

Network nodes and communication links have always been susceptible to failure. Software or hardware faults (or phenomena outside the control of a network operator such as caterpillars) may cause nodes or links to go down. To be able to adapt to faults without serious degradation in service, fault-tolerant networks and routing protocols have to be set up. Although the study of communication in faulty networks is a classical field in network theory, there has been a recent renewal of interest in fault-tolerant routing due to the tremendous rise in popularity of mobile ad-hoc networks and peer-to-peer networks. In these networks, faults are actually not an exception but a frequently occurring event: in mobile ad-hoc networks, users may run out of battery power or may move out of reach of others, and in peer-to-peer networks, users may leave without notice.

Central questions in the theoretical study of faulty networks have been:

- How many faults can a network sustain and still have a large connected component that is a constant fraction of the original size?
- How many faults can a network sustain and still emulate its ideal counterpart with constant slowdown?

The first question has been heavily studied in the graph theory community, and the second question has been investigated by the parallel computing community in an attempt to find the point up to which a faulty parallel computer

*Dept. of Computer Science and Engineering, Indian Institute of Technology, Hauz Khas, New Delhi 110016, India. Email: bagchi@cse.iitd.ac.in
†Google, 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA. Email ankurb@google.com
‡Dept. of Computer Science and Engineering, Notre Dame University, Notre Dame, IN 46556, USA. Email: achaudha@cse.nd.edu
§Dept of Computer Science, University of California, Irvine, 92697. Email: eppstein@ics.uci.edu
¶Insitut für Informatik, Technische Universität München, Boltzmannstr. 3, D-85748 Garching, Germany. Email: scheideler@in.tum.de

can still emulate an ideal parallel computer with the same topology with constant slowdown. We refer the reader to [31] for a survey of results in these areas.

In this paper, we navigate a path between these two extremes. The fact that the network is still connected gives little solace if it has a severe bottleneck (e.g., one half is connected to the other by a single edge). At the other end, we might not need the property that we can fully emulate the original network in its faulty counterpart; all we might need is that the network's routing properties do not degrade excessively. We focus on this question: what happens to a network's routing capabilities when faults occur. To get a handle on the routing capabilities of the network we focus on a parameter that has been widely used - both in graph theory and in network theory - to measure the routing quality of a graph: expansion. More specifically, we are interested in the effect of faults on the expansion of a network, showing bounds on the number of adversarial faults and the fault probability that a network can suffer and still retain a large component with good enough expansion.

Before we proceed to our results, we discuss previous work related to connectivity and emulation in the face of faults.

1.1 Large connected components in faulty networks

We start with an overview of previous results for random faults and subsequently consider adversarial faults.

Given a graph G and a probability value p, let $G^{(p)}$ be the random graph obtained from G by keeping each edge of G alive with probability p (i.e., p is the *survival probability*, in the rest of the paper we consider *fault* probabilities, but in this section we talk in terms of survival probabilities.) Given a graph G, let $\gamma(G) \in [0, 1]$ be the fraction of the nodes of G contained in a largest connected component.

Let $\mathcal{G} = \{G_n \mid n \in \mathbb{N}\}$ be any family of graphs with parameter n. We call p^* the *critical probability* for the existence of a linear-sized connected component if for every constant $\epsilon > 0$ it holds:

1. For every $p > (1 + \epsilon)p^*$ there exists a constant $c > 0$ with $\lim_{n\to\infty} \Pr[\gamma(G_n^{(p)}) > c] = 1$.
2. For all constants $c > 0$ and for all $p < (1 - \epsilon)p^*$ it holds that $\lim_{n\to\infty} \Pr[\gamma(G_n^{(p)}) > c] = 0$.

Of course, it is not obvious whether critical probabilities exist. However, results by Erdős and Rényi [13] and their subsequent improvements (e.g. [8, 25]) imply that for the complete graph on n nodes, $p^* = 1/(n - 1)$, and that for a random graph with $d \cdot n/2$ edges, $p^* = 1/d$. For the 2-dimensional $n \times n$-mesh, Kesten showed that $p^* = 1/2$ [19]. Ajtai, Komlós and Szemerédi proved that for the hypercube of dimension n, $p^* = 1/n$ [1]. For the n-dimensional butterfly network, Karlin, Nelson and Tamaki showed that $0.337 < p^* < 0.436$ [18]. Leighton and Maggs [21] showed that there is an indirect constant-degree network connecting n inputs with n outputs via $\log n$ levels of n nodes each, called the multibutterfly, that has the following property: Up to a constant fault probability it is still possible to find $O(\log n)$ length paths from a constant fraction of the inputs to a constant fraction of the outputs. Subsequently Cole, Maggs and Sitaraman [10] extended this result for the butterfly.

Adversarial fault models have also been investigated. Leighton and Maggs [21] showed that no matter how an adversary chooses f failed nodes, there will be a connected component left in the multibutterfly with at least $n - O(f)$ inputs and at least $n - O(f)$ outputs. They further show that it is even possible to route packets between the inputs and outputs in this component in almost the same amount of time steps as in the ideal case. Subsequently Leighton, Maggs and Sitaraman [23] extended this result for the butterfly.

Upfal [32], following up on work by Dwork et. al. [12] and Alon and Chung [2], showed that there is also a direct constant-degree network on n nodes, a so-called expander, that has the property: no matter how an adversary chooses f failed nodes, there will be a connected component left in it with at least $n - O(f)$ nodes. Both results are optimal up to constants. Upfal uses a pruning technique, similar in spirit to the one we use later in this paper, to achieve his bound. Upfal gives a polynomial-time algorithm for pruning while we do not. But Upfal's pruning does not guarantee a large component of good expansion, while ours does. In fact, recent work indicates that there might not be any constant approximation algorithm to determine the expansion of a graph [20].

1.2 Simulation of fault-free networks by faulty networks

Next, we look at the problem of simulating fault-free networks by faulty networks. Let us suppose that there can be up to f node faults in the system at any time. One way to find out whether the largest remaining component still allows efficient communication is to check whether it is possible to embed a fault-free network of the same size and kind into the largest connected component of a faulty network. An *embedding* of a graph G into a graph H maps the nodes of G to non-faulty nodes of H and the edges of G to non-faulty paths in H. An embedding is called *static* if the mapping of the nodes and edges is fixed. A good embedding is one with minimum load, congestion, and dilation, where the *load* of an embedding is the maximum number of nodes of G that are mapped to any single node of H, the *congestion* of an embedding is the maximum number of paths that pass through any edge e of H, and the *dilation* of an embedding is the length of the longest path. The load, congestion, and dilation of the embedding determine the time required to emulate each step of G on H. In fact, Leighton, Maggs, and Rao have shown [22] that if there is an embedding of G into H with load ℓ, congestion c, and dilation d, then H can emulate any communication step (and also computation step) on G with slowdown $O(\ell + c + d)$.

Only a few results are known so far for constant slowdown in the worst-case faults setting. Leighton, Maggs and Sitaraman used dynamic embedding strategies to show that an n-input butterfly with $n^{1-\epsilon}$ worst-case faults (for any constant ϵ) can still emulate a fault-free butterfly of the same size with only constant slowdown [23]. Furthermore, Cole, Maggs and Sitaraman showed that an $n \times n$ mesh can sustain up to $n^{1-\epsilon}$ worst-case faults and still emulate a fault-free mesh of the same size with (amortized) constant slowdown [11]. It seems that the n-node hypercube can also achieve a constant slowdown for $n^{1-\epsilon}$ worst-case faults, but so far only partial answers have been obtained [23].

Random faults have also been studied. For example, Håstad, Leighton and Newman [15] showed that if each edge of the hypercube fails independently with any constant probability $p < 1$, then the functioning parts of the hypercube can be reconfigured to simulate the original hypercube with constant slowdown. Leighton, Maggs and Sitaraman [23] showed that a butterfly network whose nodes fail with some constant probability p can still emulate a fault-free butterfly of the same size with slowdown $2^{O(\log^* n)}$. Interestingly, in the conference version of [11], Cole, Maggs and Sitaraman claim that an $n \times n$ mesh in which each node is faulty independently with a constant fault probability is able to emulate a fault-free mesh with a constant slowdown [9]. The proof of this claim, which is stronger than the theorem we prove about the $n \times n$ mesh in this paper, is omitted in [9] and has not appeared elsewhere to the best of our knowledge.

For a list of further references concerning embeddings of fault-free into faulty networks see the paper by Leighton, Maggs and Sitaraman [23].

1.3 Our approach

The two common approaches – connectivity and emulation of fault-free by faulty networks – are too extreme for many practical applications. Knowing how long a network is still connected may not be very useful, because in extreme cases (just a single line connects one half to the other) the speed of communication may be reduced to a crawl, making it useless for applications that need fast interaction or large bandwidth such as interactive gaming or video conferencing. On the other hand, emulating a fault free network on a faulty network is like using a giant hammer to crack a small nut, so to speak. Emulation may not be needed when all we want is good routing properties in the faulty network ,i.e., reduced congestion or good expansion.

In ad-hoc network settings or peer-to-peer systems, applications are usually not concerned with the exact network topology. An application typically just requires that the network provides sufficient bandwidth and ensures sufficiently small delays. In this scenario, a more relevant question is:

> *How many faults can a network sustain and still contain a network at least a constant fraction of its original size with approximately the same expansion?*

An answer to this question would have many useful consequences for distributed data management, routing, and distributed computing. Research on load balancing has shown that if the expansion basically stays the same, the ability of a network to balance single-commodity or multi-commodity load basically stays the same, and this ability can be exploited through simple local algorithms [14, 5, 3]. Also, the ability of a network to route information is preserved because it is closely related to its expansion [29]. Furthermore, as long as the original network still has a large connected component of almost the same expansion, one can still achieve almost everywhere agreement, which is an important prerequisite for fundamental primitives such as atomic broadcast, Byzantine agreement, and clock synchronization [12, 32, 7].

In a work published after the conference version of this paper, Angel et. al. [4] have also taken the view that it is important to study routing in faulty networks. The main difference is that they study the properties of faulty networks in terms of the algorithmic complexity of finding paths between vertices. Our approach is less direct than theirs, but somewhat more general and widely applicable. We will discuss some specific implications of their work and its relation to our techniques in the concluding section of this paper.

Many different fault models have been studied in the literature: faults may be permanent or transient, nodes and/or edges may break down, and faults may happen at random or may be caused by an adversary or attacker. The former are called *random faults*, and the latter are called *adversarial faults*. We will concentrate on situations in which there are *static* node faults, i.e., once a node has become faulty either randomly or adversarially it remains faulty, and a node which is not faulty remains not faulty. A central parameter we will use in our investigations is the expansion.

Given a graph $G = (V, E)$ and a subset $U \subseteq V$, the *expansion* of U is defined as

$$\alpha(U) = \frac{|\Gamma(U)|}{|U|}$$

where $\Gamma(U)$ is the set of nodes in $V \setminus U$ that have an edge from U and $|S|$ denotes the size of a set S. The *expansion* of G is defined as $\alpha(G) = \min_{U, |U| \le |V|/2} \alpha(U)$. If G is clear from the context, we will also just write α.

We also use the *uniform expansion* of a graph. Suppose that we have a family of graphs $\mathcal{G}$. Let β be a function such that $\beta(|G|) \le \alpha(G)$ for all graphs $G \in \mathcal{G}$, where $|G|$ denotes the number of nodes in G. Then we say that $\mathcal{G}$ has a uniform expansion of β if for all subgraphs H of all graphs G in $\mathcal{G}$ it holds that $\alpha(H) = O(\beta(|H|))$.

Notice that $\alpha(H) = \min_{U, |U| \le |V(H)|/2} \alpha(U)$, i.e., it should not be confused with $\alpha(V(H))$ in G. The uniform expansion definition implies that the expansion of a subgraph H of G cannot be asymptotically better than the expansion of G (which also implies that $\beta(|G|) = \Theta(\alpha(G))$ for all $G \in \mathcal{G}$).

Despite its seemingly complex definition, the uniform expansion property is a quite natural property. In fact, all commonly used graph topologies in interconnection networks such as trees, d-dimensional meshes (with equal side length in each dimension), the hypercube, the butterfly, and the de Bruijn graph fulfill this uniform expansion property.

Uniform expansion is well illustrated by the example of the mesh. If we consider the $n \times n$-mesh (i.e., the 2-dimensional mesh of side length n), we can see that it has a uniform expansion of $\beta(x) = 1/\sqrt{x}$. This is demonstrated by observing that any subgraph of size m in that mesh can have a bisection width, defined as the minimum cut separating the graph into two subgraphs of approximately equal size, of at most $\sqrt{m}$ and therefore an expansion of at most $O(1/\sqrt{m})$.

1.4 Our main results

Adversarial faults

We give general upper and lower bounds for the number of node faults a graph can sustain and still retain a large component with basically the same expansion. The bounds are tight up to constant factors. More specifically, we show that the number of adversarial node faults a graph with expansion α and n nodes can sustain, with only a constant factor decrease in its expansion, is a constant times $\alpha \cdot n$. For (families of) graphs G of size n and uniform expansion β, this result is best possible up to a constant factor in the sense that $\omega(\beta(n) \cdot n)$ faults can break G into components of size $o(n)$.

Random faults

We also study random faults. We find that that tight dependence between the robustness of a network and the expansion that is observed in the case of adversarial faults is absent here. Although we are able to show examples of networks with expansion α which fall apart with high probability given a fault probability of a constant times α, we also find that there are networks which retain their expansion at fault probabilities which are $\omega(\alpha)$. Hence we need a better way of studying the robustness of networks to random faults. This motivates our main contribution: a new parameter for the study of the effect of random faults on network expansion. This parameter, call the *span*, may be of independent interest.

Consider a graph $G = (V, E)$. Let $U \subseteq V$ be any subset of nodes. U is defined to be *compact* if and only if U and $V \setminus U$ are connected in G. Let $\mathcal{U}$ be the set of all compact sets of G. Let $P(U)$ be the smallest tree in G which connects every node in $\Gamma(U)$ (i.e., it essentially spans the boundary of U). Note that the set of nodes in $P(U)$ need not be from U alone or from $V \setminus U$ alone. Then the *span* of a graph is defined as:

$$\sigma = \max_{U \in \mathcal{U}} \left\{ \frac{|P(U)|}{|\Gamma(U)|} \right\} \tag{1}$$

The span helps us characterize the resilience of the expansion to random faults. We show that a graph with maximum degree δ, span σ, and expansion α (fulfilling certain, weak conditions) can tolerate a fault probability of up to a constant times $1/\delta^{O(\sigma)}$ and still retain an expansion of at least α/δ.

We also show that d-dimensional meshes have constant span. The proof of this theorem is of independent value as it establishes an interesting property of the d-dimensional mesh: The boundary of any set of connected vertices in the d-dimensional mesh, whose complement is also connected, can be spanned by a tree of size at most twice the size of the boundary.

1.5 Outline of the paper

The rest of the paper is organized as follows: In Section 2, we consider adversarial faults, and in Section 3, we consider random faults. The paper ends in Section 4 with a discussion of how our results are related to previous research and some open problems.

2 Adversarial faults

In this section, we assume that a malicious adversary decides which nodes are faulty. More formally, we are given a graph $G = (V, E)$ with n nodes and expansion α. An adversary gives us a faulty version of this graph, called $G_f = (V_f, E_f)$, with f faulty nodes removed. An edge $\{u, v\} \in E$ remains in E_f if and only if both u and v are non-faulty. We first prove a general upper bound on the number of adversarial faults a graph can sustain, and afterwards we also prove asymptotically matching lower bounds for certain classes of graphs.

2.1 Upper bound on adversarial faults

We show that if an adversary is allowed no more than $O(\alpha \cdot n)$ faults, there always exists a subgraph of G_f called H which has $\Theta(n)$ nodes and an expansion of $\Omega(\alpha)$.

An adversary can simply disconnect parts of the graph by making a small number of nodes faulty, causing the expansion of G_f to be 0. Therefore, the subgraph H with good enough expansion is constructed by pruning away those parts of G_f whose expansion has degraded too much. This is a critically important step in our analysis. We formalize it as an algorithm called *Prune* described in Figure 1.

Note that the running time of *Prune* is not necessarily polynomial, nor are we claiming it is. *Prune* simply helps us prove an existential result. We discuss the algorithmic aspects of *Prune* in greater detail in Section 2.3

Before we get to the algorithm, we need to introduce some notation. Given a graph G, we define $\Gamma_G(S)$ to be the set of nodes in the neighborhood of a node set S in G. The algorithm generates a sequence of graphs G_0 to G_m, where the final graph G_m is the graph H we are looking for.

Algorithm *Prune(ϵ)*

```
G_0 ← G_f; i ← 0
while ∃S_i ⊆ V(G_i) such that |Γ_{G_i}(S_i)| ≤ ϵ · α|S_i| and |S_i| ≤ |V(G_i)|/2
    G_{i+1} ← G_i \ S_i
    i ← i + 1
end while
H ← G_i; m ← i
```

Figure 1: The pruning algorithm

Theorem 2.1 *Let G be any graph with n nodes, maximum degree δ and expansion α. Suppose that the adversary can select up to $f = \frac{\alpha n}{4\delta k^2}$ faulty nodes for some constant $k > 1$. Then,* Prune$(1 - \frac{1}{k})$ *returns a subgraph H of size at least $n - \frac{f \cdot k}{\alpha}$ with expansion at least $(1 - \frac{1}{k}) \cdot \alpha$.*

Proof. Denote $G_f \setminus H$ by $\mathcal{S}$, i.e., $\mathcal{S}$ is the union of all the regions culled by *Prune*. We will show that $|\mathcal{S}| \leq \frac{k \cdot f}{\alpha}$ by contradiction. For this we need the following lemma.

Lemma 2.2 *For all j with $0 \le j < m$,*

$$\left|\Gamma_{G_f}\left(\bigcup_{0\le i\le j} S_i\right)\right| \le \sum_{0\le i\le j} |\Gamma_{G_i}(S_i)| \le \alpha \cdot \left(1-\frac{1}{k}\right)\cdot \left|\bigcup_{0\le i\le j} S_i\right| .$$

Proof. Consider the first inequality. Any node v that lies in the neighborhood of $\bigcup_i S_i$ in G_f must lie in the neighborhood of some S_i in G_f. Thus, because v is outside of $\bigcup_i S_i$ and therefore belongs to H, there must be an S_i with $v \in \Gamma_{G_i}(S_i)$. Therefore, $\Gamma_{G_f}(\bigcup_i S_i) \subseteq \bigcup_i \Gamma_{G_i}(S_i)$. Hence the first inequality. Each set S_i that is culled by *Prune*($1-\frac{1}{k}$) has the property that $|\Gamma_{G_i}(S_i)| \le \alpha \cdot (1-\frac{1}{k}) \cdot |S_i|$. Since the sets S_i are disjoint, $\sum_i |S_i| = |\bigcup_i S_i|$. Hence the second inequality. ☐

Suppose now that $|\mathcal{S}| > \frac{k\cdot f}{\alpha}$. Since each S_i has a size of at most $n/2$, there must be a j so that one of the two following cases is true:

1. $\frac{k\cdot f}{\alpha} < \left|\bigcup_{0\le i\le j} S_i\right| \le n/2$
2. $\left|\bigcup_{0\le i<j} S_i\right| \le \frac{k\cdot f}{\alpha}$ and $n/2 - \frac{k\cdot f}{\alpha} < |S_j| \le n/2$.

In the case (1), it follows from Lemma 2.2 that for $\mathcal{S}' = \bigcup_{0\le i\le j} S_i$,

$$|\Gamma_{G_f}(\mathcal{S}')| \le \alpha \cdot \left(1-\frac{1}{k}\right) \cdot |\mathcal{S}'|.$$

We know by the definition of the expansion that in G, $|\Gamma(\mathcal{S}')|$ is at least $\alpha \cdot |\mathcal{S}'|$. Hence, the number of faulty nodes in $\mathcal{S}'$'s neighborhood must be at least $\alpha(1-(1-\frac{1}{k}))\cdot|\mathcal{S}'|$, which is greater than $\alpha \cdot \frac{1}{k} \cdot \frac{k\cdot f}{\alpha} = f$. Since the total number of faults the adversary is allowed to create is at most f, we have a contradiction.

Suppose now that case (2) above is true. Let $\mathcal{S}' = \bigcup_{0\le i<j} S_i$. It follows from *Prune*(ϵ) that $|\Gamma_{G_j}(S_j)| \le (1-1/k)\alpha|S_j|$. However, in order to upper bound $|\Gamma_{G_f}(S_j)|$, we also have to consider the neighbors S_j might have in $\mathcal{S}'$. According to Lemma 2.2, $|\Gamma_{G_f}(\mathcal{S}')| \le \alpha|\mathcal{S}'| \le k\cdot f$, and therefore, there can be at most $k \cdot f$ nodes in S_j that have neighbors in $\mathcal{S}'$. Since the maximum degree of G is δ, it follows that

$$|\Gamma_{G_f}(S_j)| \le \alpha \cdot \left(1-\frac{1}{k}\right) \cdot |S_j| + \delta \cdot k \cdot f \ .$$

On the other hand, we know that $|\Gamma_G(S_j)| \ge \alpha|S_j|$. Hence, the number of faults in G_f must be at least $\alpha|S_j|/k - \delta\cdot k\cdot f$. From case (2) and the definition of f it follows that $|S_j| \ge n/2 - n/(4\delta k) \ge 3n/8$ because $\delta \ge 2$. Furthermore, $\delta \cdot k \cdot f = \alpha n/(4k)$. Hence,

$$\alpha|S_j|/k - \delta \cdot k \cdot f \ge 3\alpha n/(8k) - \alpha n/(4k) \ge \alpha n/(8k) \ .$$

But from $k > 1$ and $\delta \ge 2$ it follows that $f = \alpha n/(4\delta k^2) < \alpha n/(8k)$, a contradiction.

Hence, H is at least $n - \frac{k\cdot f}{\alpha}$ in size and has an expansion of at least $(1-\frac{1}{k})\cdot\alpha$. ☐

2.2 Lower bounds for adversarial faults

The result given in Theorem 2.1 is the best possible up to constant factors in the sense that for every $\alpha > 0$ smaller than some constant there is an infinite family of graphs with expansion α which disintegrate into components of size $o(n)$ if $f \ge c \cdot \alpha n$ for some sufficiently large constant c.

Theorem 2.3 *There exists a constant γ such that, given any $\alpha < \gamma$, there is an infinite family of graphs with expansion α for which there is an adversarial selection of $c \cdot \alpha \cdot n$ faulty nodes causing the graph to break into components of size $o(n)$, where n is the number of nodes in the graph and c is an appropriately chosen constant.*

Proof. Consider an infinite family $\mathcal{G}$ of δ-regular expander graphs with constant degree δ, i.e., δ-regular graphs with the property that every subset of nodes containing at most half of the nodes in the graph has a constant expansion. It is well-known that random δ-regular graphs with $\delta \ge 3$ almost surely have this property.

For any fixed $G \in \mathcal{G}$ of size n and any k, let graph H be a copy of G with each edge being replaced by a chain of k nodes (between its two endpoints), where k is even. Then H has $k\cdot(\delta n)/2 + n = \Theta(k \cdot n)$ nodes.

Claim 2.4 *Graph H has expansion $\Theta(\frac{1}{k})$.*

Proof. We first prove a lower bound, i.e., every subset U of the node set of H such that $|U| < |H|/2$ has expansion at least $\Omega(|U|/k)$. Then we show an upper bound, i.e., there is a subset of H with expansion $O(|U|/k)$. These together will prove the claim.

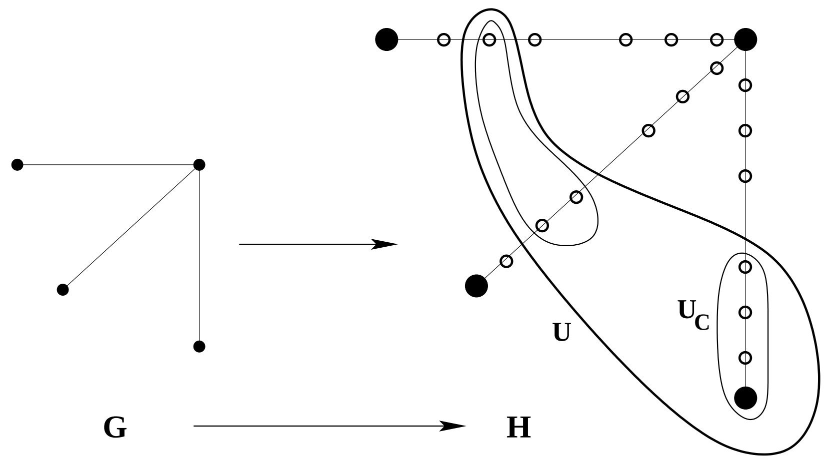

Figure 2: Distinguishing between two kinds of vertices for the lower bound.

For the lower bound, we have to show that every subset of nodes in H of size at most $|H|/2$ has an expansion of $\Omega(1/k)$. Consider any subset U of H-nodes with $|U| \leq |H|/2$. We differentiate between two sets of nodes within U. To do this we start by defining a set C as the set of all G-nodes with the property that all nodes within a distance of $k/2$ from them in H are in U. We name as U_C the set consisting of all H-nodes within a distance of $k/2$ of the nodes in C. Note that $U_C \subseteq U$. The other set of nodes we will consider is $U' = U \setminus U_C$. See Figure 2 for an illustration of this division of U into two subsets.

Let us first consider U_C. Since every G-node has exactly $\delta \cdot k/2 + 1$ many nodes within a distance of $k/2$ in H and $|U| \leq |H|/2$, it follows that $|C| \leq |G|/2$. Hence, C has an expansion of at least γ in G for some constant $\gamma > 0$. For every node $v \in \Gamma_G(C)$ there must be an H-node w within a distance of $k/2$ from v that is not in U. Let $\Gamma_H^U(U_C)$ denote the set of all of these nodes w. Since $|U_C| = (\delta \cdot k/2 + 1)|C|$, it holds that

$$|\Gamma_H^U(U_C)| \geq |\Gamma_G(C)| \geq \gamma|C| = \frac{\gamma}{\delta \cdot k/2 + 1} \cdot |U_C| \ . \tag{2}$$

Next, consider the set $U' = U \setminus U_C$ and let C' be the set of all G-nodes that have at least one U'-node within a distance of $k/2$ in H. From the definition of U' and C' it follows that for each $v \in C'$ there is at least one H-node w within a distance of $k/2$ from v that is not in U. Let $\Gamma_H^U(U')$ denote the set of all of these nodes w. Since $|U'| \leq (\delta \cdot k/2)|C'|$, we get

$$|\Gamma_H^U(U')| \geq |C'| \geq \frac{1}{\delta \cdot k/2} \cdot |U'| \ . \tag{3}$$

Combining inequalities (2) and (3), it follows that

$$|\Gamma_H(U)| \geq \max\left\{|\Gamma_H^U(U_C)|, |\Gamma_H^U(U')|\right\} \geq \max\left\{\frac{\gamma}{\delta \cdot k/2 + 1} \cdot |U_C|,\ \frac{1}{\delta \cdot k/2} \cdot |U'|\right\}$$

and since $U = U_C \cup U'$, we immediately get that $|\Gamma_H(U)| = \Omega(|U|/k)$. Hence, the lower bound holds.

To show that the upper bound holds as well, we just need to show that there exists a set of nodes U in H of size at most $|H|/2$ with an expansion of $O(1/k)$. Consider any two adjacent nodes in G. These two nodes are end points of a chain of length $k+2$ in H. Assigning the middle k nodes of this chain to the node set U gives a set with an expansion of $2/k$. Hence, the expansion of H is $O(1/k)$, which completes the proof of the claim. □

Now, from each chain of k nodes, we remove a central node. Then each connected component remaining has at most $1 + \delta \cdot \frac{k}{2}$ nodes left, which is $o(n)$, and the total number of nodes removed is $\frac{\delta}{2} \cdot n$, which is $\frac{1}{k}$ times the number of nodes in the graph. □

We now show a more far-reaching lower bound: For families of graphs that have a *uniform expansion*, our upper bound on the maximum number of node faults is not only true in some pathological cases, but for *every* case, i.e., for every graph in every such family of graphs. In the following, whenever we speak about a graph of uniform expansion, we assume that it belongs to a suitable family of graphs.

Theorem 2.5 *For every connected graph of size n and uniform expansion β there is an adversarial selection of $\omega(\beta(n) \cdot n)$ faulty nodes that causes the graph to break into components of size $o(n)$.*

Proof. Let $G = (V, E)$ be any graph of uniform expansion β that consists of n nodes. Then there must be a set $U_1 \subseteq V$, $|U_1| \leq n/2$, so that $|\Gamma(U_1)| \leq \beta(n) \cdot |U_1|$. Removing $\Gamma(U_1)$ leaves G with a set $\mathcal{V}_1 = \{V', V''\}$ of two node sets, $V' = U_1$ and $V'' = V \setminus (U_1 \cup \Gamma(U_1))$. Let V_1 be a set in $\mathcal{V}_1$ of maximum size. It follows from the uniformity of G that there must be a set $U_2 \subseteq V_1$, $|U_2| \leq |V_1|/2$, so that $|\Gamma(U_2)|$ w.r.t. $G(V_1)$ is $O(\beta(|V_1|)) \cdot |U_2|$. Removing U_2 results in a new set $\mathcal{V}_2$ of sets of nodes which is equal to $\mathcal{V}_1$ with V_1 being replaced by the node sets U_2 and $V_1 \setminus (U_2 \cup \Gamma(U_2))$. (That is, $\mathcal{V}_2$ contains three node sets.) We continue to take a node set V_i of largest size out of $\mathcal{V}_i$, remove nodes at the minimum expansion part in $G(V_i)$, and replace V_i by the two resulting node sets in $\mathcal{V}_i$ to get a set $\mathcal{V}_{i+1}$, until there is no subset in $\mathcal{V}_i$ left of size at least ϵn.

Our goal is to show that this process only removes $O(\frac{\log(1/\epsilon)}{\epsilon} \cdot \beta(n) \cdot n)$ nodes from G. If this were true, the theorem would follow immediately. We prove the bound with a charging strategy: Each time a set V_i is selected from $\mathcal{V}_i$, we charge all nodes in $\Gamma(U_{i+1})$ taken away from V_i to the nodes in U_{i+1}. Since

$$|\Gamma(U_{i+1})| = O(\beta(\epsilon n)) \cdot |U_{i+1}| = O\left(\frac{\beta(n)}{\epsilon} \cdot |U_{i+1}|\right)$$

for any $\beta(x) \geq 1/x$, this means that every node in U_{i+1} is charged with a value of $O(\epsilon^{-1} \cdot \beta(n))$. Every node can be charged at most $\log(1/\epsilon)$ times because each time a node is charged, it ends up in a node set U_{i+1} that is at most half as large as V_i, and we stop splitting a node set once it is of size less than ϵn. Hence, at the end, every node in V is charged with a value of $O(\frac{\log(1/\epsilon)}{\epsilon} \cdot \beta(n))$. Summing up over all nodes, the total charge is

$$O\left(\frac{\log(1/\epsilon)}{\epsilon} \cdot \beta(n) \cdot n\right) ,$$

which represents the number of nodes that have been removed from the graph. □

The proof of this theorem indicates that families of graphs with uniform expansion may have a smooth degradation in the size of their largest connected component. This is in fact true for the family of $n \times n$ meshes. However, since there are also families of graphs with a threshold behavior, like the graphs used in Theorem 2.3, Theorem 2.1 is, up to constant factors, the best one can show in general.

2.3 Algorithmic pruning

The pruning step is crucial to the results discussed in this section and it will be central to the handling of random faults in the next section. Before proceeding to that section, we briefly discuss the possibility of actually performing a pruning step in polynomial time.

The pruning algorithm can be seen as a solution to the decision version of the sparsest cut problem which can be stated as follows.

Sparsest Cut:
Given a graph $G = (V, E)$ with arbitrary non-negative edge capacities $c(e)$, $e \in E$, and some $\beta > 0$, find a set $U \subseteq V$ with $|U| \leq |V|/2$ and

$$\sum_{u \in U, v \in V \setminus U} c(u, v) \leq \beta \cdot |U|.$$

The sparsest cut problem is known to be NP-hard. However, if there existed a γ-approximation algorithm for the optimization version of the sparsest cut problem, we could use it in the following way for the pruning:

- Find a γ-approximate sparsest cut $(S, \bar{S})$ of the faulty graph.
- If S satisfies $\alpha(S) \leq \epsilon \cdot \alpha$ for some small constant $\epsilon > 0$, prune S away, otherwise stop.

Since the expansion of the cut found by the algorithm is at most γ times the expansion of the faulty graph, we know that the expansion of the remaining graph is $\Omega(\alpha/\gamma)$.

The proof that the size of this component is at least a constant fraction of the original size of the graph when allowing $c \cdot \alpha n$ faults for a sufficiently small constant c follows from the existential proof above.

Recently, the best known approximation to the sparsest cut problem was improved to $\sqrt{\log n}$ by Arora, Rao and Vazirani [6]. They had also conjectured that the sparsest cut problem should be approximable in polynomial time to within a constant factor. This would have had the effect of turning the existential results in this paper into algorithmic ones, making it possible to actually find large components with good expansion whose existence is proved here in polynomial time. However, recently obtained lower bounds indicate that this might not be possible [20].

3 Random faults

We now direct our attention to the case of random faults. We assume that each node in the graph can become faulty independently with a given probability p.

3.1 Random faults are not always easier to handle

Intuitively, it appears that random faults are easier to handle than adversarial faults, i.e., a graph should retain a linear sized component in the face of many more random faults than adversarial faults. However, as we will see in this section, this is not true in general. In fact, there are families of graphs for which a fault probability of $\Theta(\alpha)$ causes the graph to disintegrate into components of size $o(n)$, where α is the expansion of the graph. In other words, in these graphs $\Theta(\alpha n)$ random node failures can be as bad as adversarial node failures.

Theorem 3.1 *There exists a constant γ such that, given any $\alpha < \gamma$, there is an infinite family of graphs with expansion $\Theta(\alpha)$ for which a fault probability of $\Theta(\alpha)$ causes the graph to disintegrate into components of size $o(n)$, with high probability.*

Proof. We use the family of graphs constructed in the proof of Theorem 2.3, i.e., let $\mathcal{G}$ be an infinite family of constant degree expander graphs with constant expansion γ and constant degree δ. Further, let H be the graph resulting from G by replacing every edge in G by a chain of k nodes (between its two endpoints). Graph H has $\Theta(k \cdot n)$ nodes. From Claim 2.4, we know that H has an expansion of $\Theta(\frac{1}{k})$. We need to establish some important properties of H before we can complete the proof of this theorem. For this we need to define the *border* of a node set S as the set of all nodes in S which have an edge to a vertex outside S.

The following lemma helps us to count subgraphs of H with a certain property.

Lemma 3.2 *The number of connected subgraphs H' of H containing exactly r nodes of G and whose border lies entirely in G is at most $n \cdot \delta^{2(r-1)}$.*

Proof. It suffices to count all possible connected subsets of nodes of size r in G. This is because all the subgraphs we are concerned with are bordered by nodes of G. Any connected subset of nodes of size r can be spanned by a tree with $r - 1$ edges. This tree can be traversed by an Eulerian tour in which each edge is used at most twice. Hence, all nodes of the subset can be visited by a walk along at most $2(r - 1)$ edges in G. Since the starting point of the walk can be any one of the n vertices, there are at most $n \cdot \delta^{2(r-1)}$ ways of generating a connected subset of size r. □

Now, let us consider any subgraph H' of H with the properties of Lemma 3.2, i.e., H' contains exactly r nodes of G and its border lies entirely in G. Consider a spanning tree of H'. Since each node in $H' \cap G$ has to be spanned, the tree must contain at least $r - 1$ "edges" of G which are chains in H'. Each chain contains k vertices of $H' \setminus G$.

Further we can associate at least one vertex of G with each chain. Hence, H' contains at least $(k+1)(r-1)$ nodes. With this observation in place we are ready to proceed with the proof.

Let the failure probability of the nodes in H be $p = (2\ln\delta + 2)/k$. We say that a subgraph *survives* if none of its nodes become faulty. Hence, H' survives with probability at most $(1-p)^{(k+1)(r-1)} \leq e^{-k(r-1)\cdot p}$. By Lemma 3.2, there are no more than $n \cdot \delta^{2(r-1)}$ such subgraphs H'. Hence, for $r = \ln n + 1$, the probability that any such subgraph survives is at most

$$n \cdot \delta^{2(r-1)} \cdot e^{-k(r-1)\cdot p} = n \cdot \delta^{2(r-1)} \cdot e^{-(r-1)(2\ln\delta+2)} = n \cdot e^{-2\ln n} = 1/n \ .$$

But if no such subgraph survives, then a connected component in H can have a size of at most $O(\delta \cdot k \ln n)$. Thus, H breaks down into components of size $o(n)$, with high probability. ☐

However, the expansion of a graph is *not* necessarily the critical point (i.e., the point at which the graph disintegrates into components of size $o(n)$, with high probability) for all graphs. There are several important classes of graphs which can sustain a much higher fault probability and still yield a linear sized connected component with good expansion. One specific case is the mesh. In the following, we describe a general technique to quantify this higher fault probability.

3.2 Extracting a large subgraph from a graph with random faults

Consider any graph $G = (V, E)$ with n nodes, maximum degree δ, expansion α, and span σ. Let $G_f = (V, E_f)$ be the faulty version of G where each node is made faulty independently with probability p. An edge $\{u, v\} \in E$ remains in E_f if and only if both u and v are non-faulty.

We want to find a graph $H \subseteq G_f$ of size $\Theta(n)$ with expansion $\Omega(\alpha/\delta)$ for values of p up to a constant times $1/\delta^{c\cdot\sigma}$ where c is another constant. Here is a road map that might make it easier for the reader to follow as we attempt to find such an H.

- Recall the definition of *compact sets*: sets in the graph which are connected and whose complement is also connected. These sets will prove extremely useful for us. One convenient feature they have is that they are easy to count in terms of the span. (See **Proposition 3.3**.)
- Ease of counting in itself is not of much use. But we will also show that compact sets are the bottlenecks which determine the expansion of the graph. Formally, we will prove that if a graph has a set with expansion $< \gamma$ for any value of $\gamma > 0$ then it has a compact set of expansion $< \delta \cdot \gamma$. (See **Lemma 3.4**.)
- Hence, we know that pruning compact sets with bad expansion achieves what we want: the remaining graph has good expansion. So we will modify *Prune(ϵ)* to prune only compact sets of bad expansion from a faulty graph . (See algorithm *PruneCompact(ϵ)* in **Figure 4**.)
- We take care of an important technicality by showing that the connected components over the sets pruned by *PruneCompact(ϵ)* are also compact. (See **Claim 3.7**.)
- Finally, we show that the probability of the number of nodes culled being large is very small for values of the fault probability below a certain threshold. (See **Theorem 3.6**.)

  To show this, we handle two cases separately.

  1. *There is one pruned component with a large neighborhood.* We limit the probability of this by using the inclusion-exclusion principle. The fact that it is easier to count compact sets plays a major part here.
  2. *Each pruned component has a small neighborhood.* Here we have to make a careful counting argument looking at independent sets in the set of pruned connected components.

Now let us execute this road map.

Let $\mathcal{U}$ be the set of all compact sets of G. Recall that a set is compact if it and its complement are connected. In the following, given any graph G', $V(G')$ denotes the node set of G', $\Gamma_{G'}(U)$ denotes the neighborhood of node set U with respect to G' and $\alpha_{G'}(U)$ denotes the expansion of node set U with respect to G'.

We begin by relating the number of compact sets with a given neighborhood to two graph parameters: maximum degree and span.

Proposition 3.3 *For a graph G with span σ and maximum degree $\delta \geq 3$, there are at most $n \cdot \delta^{3\sigma \cdot k}$ compact sets with k nodes in their neighborhood.*

Proof. Consider compact sets with boundaries of size k. The proof of Lemma 3.2 implies that there are at most $n \cdot \delta^{2\sigma \cdot k}$ ways of generating a tree of size $\sigma \cdot k$ in G. Note that a tree of size $\sigma \cdot k$ can span up to $\binom{\sigma \cdot k}{k}$ sets of k vertices each. Hence, it can span at most $\binom{\sigma \cdot k}{k} \leq (e\sigma)^k$ compact sets with k nodes in their neighborhood. Therefore, for $\delta \geq 3 \geq e$, there are at most $n \cdot (e\sigma)^k \cdot \delta^{2\sigma \cdot k} \leq n \cdot \delta^{3\sigma \cdot k}$ compact sets with k nodes in their neighborhood. □

Before we describe the modified pruning algorithm, we set the stage for it. The following lemma will be used to show that our new pruning algorithm, called *PruneCompact* (see Figure 4), will find suitable candidates at every instance by showing that whenever a graph has a subset with bad expansion, it has a compact subset whose expansion is not much better.

Lemma 3.4 *Let $G = (V, E)$ be a connected graph with maximum degree δ and $G' = (V, E')$ be a subgraph of G. For any subset $S \subset V$ with $|S| \leq |V|/2$ there exists a compact set $K_{G'}(S)$ in G with $|K_{G'}(S)| \leq |V|/2$ whose expansion in G' is at most $\delta \cdot \alpha_{G'}(S)$.*

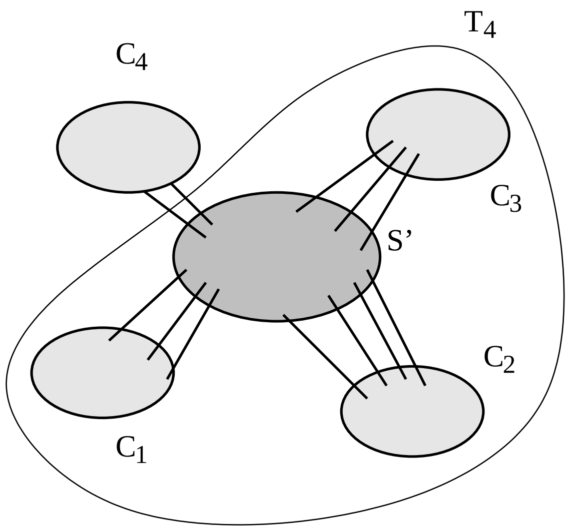

Figure 3: Note that though S is not compact, both C_4 and $S \cup C_1 \cup C_2 \cup C_3$ are.

Proof. Throughout this paper we have used the term expansion to mean node expansion, i.e., we have looked at expansion in terms of the size of the neighborhood of sets. However, to prove this lemma we will use the concept of *edge* expansion, which is defined as follows. Given a set $U \subseteq V$, we denote by $e(U)$ the set of edges with one endpoint in U and the other endpoint in $V \setminus U$. The *edge expansion* of U is as $\beta(U) = |e(U)|/|U|$. The *edge expansion* of a graph $G = (V, E)$ is

$$\beta(G) = \min_{U, |U| \leq |V|/2} \beta(U)$$

The edge expansion of a set and its node expansion are related as follows:

Fact 3.5 *For any set $U \subseteq V$ with $|U| \leq |V|/2$,*

$$\alpha(U) \leq \beta(U) \leq \delta \cdot \alpha(U).$$

We will now show that for any set $S \subseteq V$ with $|S| \leq |V|/2$ we can find a compact set S'' such that $\beta(S'') \leq \beta(S)$. This coupled with Fact 3.5 gives us the proof of the lemma.

Suppose first that S contains multiple connected components $S_1, S_2, \ldots, S_k$. Since no two of these connected components can share an edge of $e(S)$, it is clear that

$$\beta(S) = \frac{|e(S_1)| + |e(S_2)| + \cdots + |e(S_k)|}{|S_1| + |S_2| + \cdots + |S_k|}$$

Hence, by a simple averaging argument, there must be an i, $1 \le i \le k$, for which $\beta(S_i) \le \beta(S)$. Let us denote this S_i as S'.

If S' is compact, we are done and $K_{G'}(S) = S'$. So let us assume that S' is not compact. Suppose that $V \setminus S'$ has connected components $C_1, C_2, \ldots C_\ell$. Figure 3 clearly shows that every C_i is a compact set. That is because S' has connections to all the other components. Similarly, $T_i = (S' \cup C_1 \cup C_2 \cdots \cup C_\ell) \setminus C_i$ is a compact set for each i, $1 \le i \le \ell$. Figure 3 also shows us two other properties of each of these sets T_i: (1) $e(T_i) \le e(S')$ because some of the edges of $e(S')$ may now have become internal to T_i, and (2) $|T_i| \ge |S'|$ because each T_i contains S' in its entirety and may contain some more nodes from the components of $V \setminus S'$.

Hence, it follows that if there is an i^* for which $|T_{i^*}| \le |V|/2$, then

$$\beta(T_{i^*}) \le \beta(S')$$

and the S'' we are looking for is this T_{i^*}.

If $|T_i| > |V|/2$ for every $1 \le i \le \ell$, this means that $|C_i| \le |V|/2$ for each $1 \le i \le \ell$, because $|T_i| + |C_i| = |V|$. Now, let us look at the edge expansion of S' again:

$$\begin{aligned} \beta(S') &= \frac{|e(S')|}{|S'|} = \frac{|e(C_1)| + |e(C_2)| + \cdots + |e(C_\ell)|}{|S'|} \\ &\ge \frac{|e(C_1)| + |e(C_2)| + \cdots + |e(C_\ell)|}{|C_1| + |C_2| + \cdots + |C_\ell|} \end{aligned}$$

since $|S'| \le |V|/2$. By an averaging argument as before, there must be an i^*, $1 \le i^* \le \ell$ such that

$$\frac{|e(C_1)| + |e(C_2)| + \cdots + |e(C_\ell)|}{|C_1| + |C_2| + \cdots + |C_\ell|} \ge \frac{|e(C_{i^*})|}{|C_{i^*}|}$$

and since we have $|C_i| \le |V|/2$ for each i, it follows that $\beta(C_{i^*}) \le \beta(S')$. Hence, our choice of S'' is C_{i^*}.

This S'' we have chosen is the $K_{G'}(S)$ we set out to find. We have shown that $|K_{G'}(S)| \le |V|/2$ in each case, and Fact 3.5 ensures that it has the property that $\alpha(K_{G'}(S)) \le \delta \cdot \alpha(S)$. □

Algorithm *PruneCompact(ϵ)*

$G_0 \leftarrow G_f$; $i \leftarrow 0$
while $\exists S_i \subseteq V(G_i)$ s.t. $|\Gamma_{G_i}(S_i)| \le (\epsilon/\delta) \cdot \alpha |S_i|$ and $|S_i| \le |G_i|/2$
  $K_i \leftarrow K_{G_i}(S_i)$
  $G_{i+1} \leftarrow G_i \setminus K_i$
  $i \leftarrow i + 1$
end while
$H \leftarrow G_i$

Figure 4: The pruning algorithm

Consider now the algorithm *PruneCompact* in Figure 4. We use the notation from this algorithm in the proof of Theorem 3.6. Recall that the initial graph $G_0 = G_f$ contains *all* nodes of G, but the faulty nodes are isolated. In each iteration of the algorithm in which the expansion of G_i is still poor, a node set $K_{G_i}(S_i)$ is pruned away from G_i. Due to Lemma 3.4, $K_{G_i}(S_i)$ is compact in G restricted to the nodes in G_i. Notice that $K_{G_i}(S_i)$ may contain faulty nodes.

Theorem 3.6 *Consider any graph $G = (V, E)$ with maximum degree δ, span σ, and expansion $\alpha \ge (\gamma \delta \ln^3 n)/n$ for some sufficiently large constant γ and $|\Gamma(U)| \ge \log_\delta |U|$ for every node set $U \subseteq V$ with $|U| \le |V|/2$. Then, with high probability, provided the fault probability $p \le 1/(16e \cdot \delta^{8\sigma})$ and $\epsilon \le 1/4$,* PruneCompact(ϵ) *returns a graph $H \subseteq G$ of size $|H| \ge n/3$ with expansion at least $(\epsilon/\delta) \cdot \alpha$.*

Proof. In the following we will always assume that $\delta \ge 3$ since otherwise the lower bound on α is violated. We begin the proof by showing that the regions pruned are compact. Formally, let $\mathcal{T} = G \setminus H$. Hence, $\mathcal{T}$ is the union of all the culled regions. To prove the result, we will show that, with high probability, the size of $\mathcal{T}$ is no more than $n/2$. Let $T_1, T_2, \ldots, T_\ell$ be the connected components of $\mathcal{T}$ in G.

Claim 3.7 *Every $T_j \in \mathcal{T}$ is compact in G.*

Proof. We will denote by $G(V(G_i))$ the graph G induced by $V(G_i)$, i.e., the graph that contains all nodes in G_i and all edges in G that have both endpoints in G_i. Hence, $G(V(G_i))$ is the non-faulty version of G_i.

First of all, notice that $G(V(G_i))$ is connected for every i. This is true because only sets that are compact in $G(V(G_i))$ are removed from G_i for every i, and $G(V(G_0)) = G$. Moreover, every compact set K_i pruned away from G_i has at least one edge into $G_i \setminus K_i$ in $G(V(G_i))$. Also, every K_i must be completely contained in one of the T_j. This holds because the T_j's are connected components. If a K_i were distributed among at least two T_j's, then it cannot be connected, violating the compactness property.

Consider now any set T_j. Because it is a (maximal) connected component, it has no edges to any T_k with $k \neq j$. But T_j must represent the union of compact sets that were culled away, and the last of these that was culled away must have at least one edge into $G \setminus T_j$ because otherwise G would be disconnected. Hence, T_j must have at least one edge into H for every j. Thus, $G \setminus T_j$ is connected for every T_j implying that every T_j is compact in G. ☐

We now establish a bound on the probability that a given set T_i gets culled. This bound is in terms of δ, σ and the size of the neighborhood of T_i.

In the following, let $\Gamma(U)$ denote the neighborhood of U in G and $\Gamma_f(U)$ denote the neighborhood of U in G_f. From the definition of α it follows that $|\Gamma(T_i)| \geq \alpha|T_i|$ for every i. Furthermore, it follows from Lemma 3.4 that

$$|\Gamma_f(T_i)| \leq \sum_{K_j \in T_i} |\Gamma_{G_j}(K_j)| \leq \sum_{K_j \in T_i} \epsilon\alpha|K_j| = \epsilon\alpha|T_i| \leq \epsilon|\Gamma(T_i)| \ .$$

Hence, $|\Gamma(T_i)| - |\Gamma_f(T_i)| \geq (1-\epsilon)|\Gamma(T_i)|$. A set T_i is part of the culled region only in the event that the size of its neighborhood reduces from $|\Gamma(T_i)|$ to $|\Gamma_f(T_i)|$. This probability is at most

$$\binom{|\Gamma(T_i)|}{|\Gamma(T_i)| - |\Gamma_f(T_i)|} \cdot p^{|\Gamma(T_i)|-|\Gamma_f(T_i)|} \leq \left(\frac{ep|\Gamma(T_i)|}{|\Gamma(T_i)| - |\Gamma_f(T_i)|}\right)^{|\Gamma(T_i)|-|\Gamma_f(T_i)|} \tag{4}$$

$$\leq \left(\frac{ep}{1-\epsilon}\right)^{(1-\epsilon)|\Gamma(T_i)|} \tag{5}$$

If we set

$$\epsilon \leq \frac{1}{4} \quad \text{and} \quad ep \leq \frac{1}{16\ \delta^{8\sigma}},$$

then it follows that

$$\Pr[T_i \text{ is culled by } \textit{PruneCompact}] \leq \left(\frac{1}{12\delta^{8\sigma}}\right)^{3/4|\Gamma(T_i)|} \leq \frac{1}{12} \cdot \delta^{-6\sigma|\Gamma(T_i)|} \ .$$

In order to establish that the probability that the sizes of the T_is cumulatively add up to more than $2n/3$ is low, we need to distinguish between two cases based on the size of the neighborhood of the T_i's. In the following, let $k = \lceil \log_\delta n \rceil$.

Case 1: There exists an i with $|\Gamma(T_i)| > k$, i.e. there is a T_i with a large neighborhood.
From above we know that a given compact subgraph T_i is culled with probability at most $\delta^{-6\sigma|\Gamma(T_i)|}$. We multiply this probability with the number of ways of choosing such a subgraph. This gives us the probability that there is a T_i with such a large neighborhood. Lemma 3.2 implies that there are at most $n \cdot \delta^{3\sigma \cdot |\Gamma(T_i)|}$ compact sets with $|\Gamma(T_i)|$ nodes in the neighborhood. By definition, $\sigma \geq 1$. Hence,

$$\Pr[\exists T_i : \ |\Gamma(T_i)| > k] \quad \leq \textstyle\sum_{t=k}^{n} n \cdot \delta^{3\sigma \cdot t} \cdot \delta^{-6\sigma \cdot t} \tag{6}$$

$$\leq n^2 \cdot \delta^{-3\sigma k} \leq \tfrac{1}{n} \ . \tag{7}$$

Case 2: For all i, $|\Gamma(T_i)| \leq k$.
Without loss of generality, suppose that we halt *PruneCompact(ϵ)* once more than $2n/3$ nodes have been pruned away. Let $\mathcal{T}$ be the union of the culled sets before that round, and let K be the set of nodes pruned away so that the $2n/3$ bound is exceeded. Then it holds that $n/3 \leq |\mathcal{T}| \leq 2n/3$ because if $|\mathcal{T}| < n/3$, then $|K| > n/3 \geq (n - |\mathcal{T}|)/2$, which would violate the choice of K_i in step 3 of *PruneCompact(ϵ)* (see also Lemma 3.4).

Our goal is to show that a set $\mathcal{T}$ of pruned sets with $n/3 \leq |\mathcal{T}| \leq 2n/3$ cannot exist, with high probability. Let $T_1, \ldots, T_\ell$ be the connected components of such a $\mathcal{T}$. Recall that $T_1, \ldots, T_\ell$ are compact sets. First, suppose that there is a T_i with $|T_i| \geq |V|/2$. From the definition of the expansion it holds that

$$|\Gamma(T_i)| \geq \delta^{-1}|\Gamma(V \setminus T_i)| \geq \delta^{-1}\alpha n/2$$

because every node in $\Gamma(V \setminus T_i)$ must be in T_i and must have a neighbor in $\Gamma(T_i)$. Thus, from the lower bound on α in Theorem 3.6 it follows that

$$|\Gamma(T_i)| \geq \delta^{-1} \cdot \frac{\gamma\delta \ln^3 n}{n} \cdot n/2 = (\gamma/2)\ln^3 n > k$$

and, hence, we are back to case 1 above.

Therefore, we can assume in the following that $|T_i| \leq |V|/2$ for every i. In this case, we show that it is unlikely that $\Sigma_i|T_i|$ is more than $n/3$ with high probability. This is done by looking at the sets $\mathcal{S}_t$ of all possible candidate sets for the T_i's with t neighbors, $t \in \{1, \ldots, k\}$. For technical reasons, we count separately those t for which it is likely that we have $\Omega(\log^2 n)$ candidate sets: we show that the total number of nodes in these candidate sets is bounded by $n/6$, with high probability. We also show the same result for the remaining values of t, adding these up with the earlier nodes to a total of at most $n/3$ nodes that can be culled, with high probability.

Let $\mathcal{S}_t$ be the collection of all compact sets in G with a neighborhood of size exactly t. We know from above that $|\mathcal{S}_t| \leq n \cdot \delta^{3\sigma t}$. A subset $\mathcal{S}'_t$ of $\mathcal{S}_t$ is called a *valid candidate* for *PruneCompact(ϵ)* if and only if for any two sets $T, T' \in \mathcal{S}'_t$, T and T' do not intersect each other's neighborhoods (i.e., $T \cap \Gamma(T') = \emptyset$ and $T' \cap \Gamma(T) = \emptyset$). Only in this case, T and T' are separate connected components. A set $T \in \mathcal{S}_t$ is called *bad* if $|\Gamma_f(T)| \leq \epsilon \cdot \alpha|T|$. Certainly, if *PruneCompact(ϵ)* culls s compact sets T with $|\Gamma(T)| = t$, then there must exist a valid candidate $\mathcal{S}'_t$ in $\mathcal{S}_t$ of size at least s in which all sets are bad. But then, there must exist a set $\mathcal{S}''_t$ of size at least $s/(\delta t)$ in $\mathcal{S}_t$ with the property that all sets in $\mathcal{S}''_t$ are bad and for all sets $T, T' \in \mathcal{S}''_t$ it holds that $(T \cup \Gamma(T)) \cap (T' \cup \Gamma(T')) = \emptyset$. (A set $\mathcal{S}''_t$ fulfilling the latter property is also called an *independent collection* in the following.) This holds because for every node, a valid candidate $\mathcal{S}'_t$ can have at most δ sets that have this node in their neighborhood, and therefore a set in $\mathcal{S}'_t$ can share its neighborhood with at most $(\delta - 1)t \leq \delta t - 1$ other sets in $\mathcal{S}'_t$.

Independent collections of compact sets have the advantage that the events that compact sets are bad are independent. So our goal will be to bound the probability that there exists an independent collection $\mathcal{S}''_t$ in $\mathcal{S}_t$ of size at least s'' in order to bound the number of compact sets with neighborhood t that *PruneCompact(ϵ)* may cull.

Let the random variable X denote the maximum number of bad independent sets in $\mathcal{S}_t$. Since there are at most $\binom{n\cdot\delta^{3\sigma t}}{s''}$ ways of choosing an independent collection of size s'', it holds for $n/(4\delta^{3\sigma t}) \geq 12 \ln n$ that

$$\begin{aligned}
\Pr[X \geq n/(4\delta^{3\sigma t})] &\leq \binom{n \cdot \delta^{3\sigma t}}{n/(4\delta^{3\sigma t})} \cdot \left(\frac{1}{12\delta^{6\sigma t}}\right)^{n/(4\delta^{3\sigma t})} \\
&\leq \left(\frac{e \cdot n \cdot \delta^{3\sigma t}}{n/(4\delta^{3\sigma t})}\right)^{n/(4\delta^{3\sigma t})} \cdot \left(\frac{1}{12\delta^{6\sigma t}}\right)^{n/(4\delta^{3\sigma t})} \\
&\leq \left(\frac{e}{3}\right)^{n/(4\delta^{3\sigma t})} \leq e^{-1.1\ln n} = \frac{1}{n^{1.1}} \ .
\end{aligned}$$

Hence, for these values of t, the maximum size of an independent collection in $\mathcal{S}_t$ is bounded by $n/(4\delta^{3\sigma t})$, with high probability. From the previous arguments this implies that the maximum number of sets with neighborhood t that *PruneCompact(ϵ)* can cull is at most $(\delta t) \cdot n/(4\delta^{3\sigma t})$, with high probability. Since we assumed that $|\Gamma(U)| \geq \log_\delta |U|$ for every node set $U \subseteq V$ with $|U| \leq |V|/2$, it further holds that

$$|T| \leq \min\left[\delta^{|\Gamma(T)|},\ \alpha^{-1}|\Gamma(T)|\right] \ .$$

Hence, for any t so that $n/(4\delta^{3\sigma t}) \geq 12 \ln n$, *PruneCompact($\epsilon$)* can cull at most

$$m_t = \delta^t \cdot (\delta t) \cdot n/(4\delta^{3\sigma t}) \leq n/(4\delta^{\sigma t})$$

nodes, with high probability.

Let t_0 be the smallest t so that $n/(4\delta^{3\sigma t}) < 12\ln n$ (i.e., the bounds above do not apply). Then we define the random variable X as the maximum number of bad independent sets in $\cup_{t=t_0}^{k}\mathcal{S}_t$. Suppose that we have some canonical rule of specifying a unique independent collection of bad sets of maximum size (for example, add a bad set that is independent to the selected ones one by one, taking the one with lowest node ID if there is more than one). For any $i \in \cup_{t=t_0}^{k}\mathcal{S}_t$, let the binary random variable X_i be 1 if and only if i belongs to the canonical independent collection of bad sets. Then it holds that $X = \sum_i X_i$. Certainly, $\Pr[X_i = 1] \leq p_i$ where p_i is the probability that set i is bad, and

$$\mathrm{E}[\prod_{i\in S} X_i] \leq \prod_{i\in S} p_i$$

for any subset $S \subseteq \cup_{t=t_0}^{k}\mathcal{S}_t$ because only independent sets i can be in the canonical set. Hence, for an upper bound on X we can view X as the sum of negatively correlated random variables X_i with $\Pr[X_i = 1] = p_i$. In this case,

$$\mathrm{E}[X] \leq \sum_{t=t_0}^{k} n \cdot \delta^{3\sigma t} \cdot (1/12) \cdot \delta^{-6\sigma t} = \sum_{t=t_0}^{k} n/(12\delta^{3\sigma t}) \leq n/(8\delta^{3\sigma t_0}) \leq 6\ln n$$

and we can use the Chernoff bounds for sums of negatively correlated random variables in [30] (which are identical to the standard Chernoff bounds for an upper bound) to get

$$\Pr[X \geq 12\ln n] \leq e^{-(6\ln n)/3} = \frac{1}{n^2}\ .$$

Hence, *PruneCompact(ϵ)* can cull at most

$$\alpha^{-1}k \cdot (\delta k) \cdot 12\ln n \leq n/6$$

nodes in $\cup_{t=t_0}^{k}\mathcal{S}_t$, with high probability, if $\alpha \geq (72\ \delta \ln^3 n)/n$. Thus, overall, *PruneCompact(ϵ)* can cull at most

$$\sum_{t=1}^{t_0} m_t + n/6 = \sum_{t=1}^{t_0} n/(4\delta^{\sigma t}) + n/6 \leq n/6 + n/6 = n/3$$

nodes in case 2, with high probability, which implies that H must have at least $n/3$ nodes, with high probability.

Combining the two cases, we get

$$\Pr[\text{number of nodes pruned} \geq n/3] \leq \Pr[\text{Case 1}] + \Pr[\text{Case 2}] \leq \frac{1}{n}$$

which completes the proof. □

3.3 Span of the mesh

Theorem 3.8 *The d-dimensional mesh has span 2.*

Proof. Consider a d-dimensional mesh M with vertex set V. Suppose S is a compact set in this mesh. Let V be the set of vertices in M and let $B \subseteq V \setminus S$ be the neighborhood nodes $\Gamma(S)$. We place virtual edges between nodes in B. Two distinct nodes $u = (u_0, \ldots u_{d-1})$ and $v = (v_0, \ldots v_{d-1})$ have a virtual edge between them iff $|v_i - u_i| = 0$ for at least $d-2$ of its dimensions and $|v_i - u_i| \leq 1$ for the rest. Call the set of such virtual edges E_v. In Lemma 3.9, stated below, we claim that the graph (B, E_v) is connected. Therefore, we can find a spanning tree for B which has exactly $|B| - 1$ virtual edges. Since each edge in E_v can be simulated by at most 2 edges of M, we can say that there is a spanning tree in M for the nodes of B with at most $2 \cdot (|B| - 1)$ edges. □

Lemma 3.9 *Let $S \subset Z^d$ be a finite compact set in a d-dimensional mesh M, let B be the set of neighborhood nodes $\Gamma(S)$, and let E_v be the set of virtual edges. Then the graph (B, E_v) is connected.*

Proof. We will show that for any two points u and v in B, there is a path in E_v connecting the two; if this can be done for every two points, then B is connected as we hope to prove.

Our proof uses some basic and standard homology theory of cell complexes, which can be found in any introductory topology text; for instance, see [16]. Specifically, we use the Z_2 homology of d-dimensional Euclidean space R^d. We partition R^d into a complex of unit hypercube cells having the points of Z^d as their vertices. Each d-dimensional unit hypercube cell has as its neighborhood a set of $2d$ $(d-1)$-dimensional unit hypercube facets, again having Z^d as vertices, and so on. In this complex, a *k-chain* is defined to be any finite set of k-dimensional unit hypercubes having points of Z^d as vertices. The *neighborhood* of a k-chain C is the symmetric difference of the boundaries of its hypercubes; that is, it is the set of $(k-1)$-dimensional hypercubes that are on the neighborhood of an odd number of the k-dimensional hypercubes in C. A *k-cycle* is defined to be a k-chain that has an empty neighborhood, and a *k-neighborhood* is defined to be a k-chain that is the neighborhood of some $(k+1)$-chain. For quite general classes of cell complexes in R^d (and even more complicated topological spaces), every k-neighborhood is a k-cycle, but in R^d, the reverse is also known to be true: every k-cycle is a k-neighborhood.

Now, note that the given points u and v are in the complement of S, i.e., in $V \setminus S$ where V is the set of vertices in M. Since $V \setminus S$ is connected, we can find a path p_1 connecting u to v by a sequence of adjacent points in $V \setminus S$. We also find an edge e_1 connecting u to an adjacent point of Z^d in S, an edge e_2 connecting v to an adjacent point of Z^d in S, and a path p_2 connecting these two interior points by a sequence of adjacent points inside S (since S is connected as well). The union of p_1, p_2, and $\{e_1, e_2\}$ forms a 1-chain in the cubical complex described above. Moreover, this is a 1-cycle, because it has degree two at every vertex it touches. Therefore, it is the neighborhood of a 2-chain C; that is, C is a set of squares and $p_1 \cup p_2 \cup \{e_1, e_2\}$ is the set of edges in the cubical complex that touch odd numbers of squares in C.

Next, let U be the subset of R^d formed by a union of axis-aligned unit hypercubes, one for each member of $V \setminus S$, and having that member as its centroid; note that these hypercubes do not have integer vertices. Let B_U be the $(d-1)$-neighborhood of U, i.e., B_U consists of a collection of $(d-1)$-dimensional unit hypercubes that again do not have integer vertices. Now look upon C and B_U as two sets of points in R^d. Let G be the intersection of these two sets, i.e., $G = C \cap B_U$.

To understand the structure of G, consider the intersection of a square s of C and a $(d-1)$-dimensional hypercube h of B_U. Observe the following: (1) all points in s have at least $(d-2)$ coordinates with integer values, (2) the locus of points in h that have at least $(d-2)$ coordinates with integer values is a set of $(d-1)$ line segments that intersect at the centroid of h, and (3) if s intersects h, there are two adjacent vertices of s such that one is in S and the other in the complement of S. Thus, if s and h meet, they (see Figure 5) do so in a line segment of length $1/2$, that connects the centroid of h (where it is crossed by one edge of the square) to the centroid of one of its neighborhood $(d-2)$-dimensional hypercubes. Therefore G, the union of these line segments, can be viewed as a graph that connects vertices at these points. The degree of a vertex at the centroid of h is equal to the number of squares of C that touch that point (which is at most $2(d-1)$), and the degree of the other vertices can only be two or four depending on which of the four vertices of the square defining the vertex is interior to U.

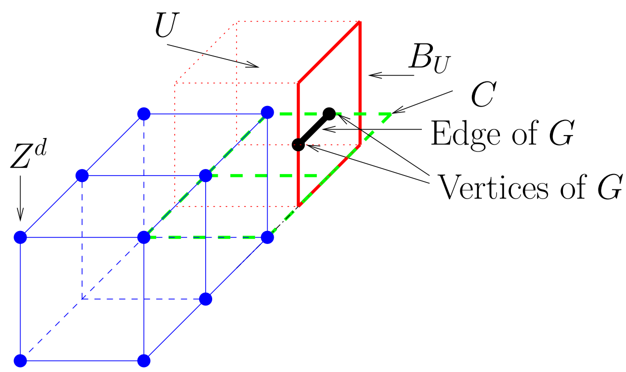

Figure 5: Vertices and edges of G formed by the intersection of squares in C and $(d-1)$-hypercubes in B_U.

Since the neighborhood of C crosses B_U only on the two edges e_1 and e_2, these two crossing points have odd degree and all the other vertices of G have even degree. Any connected component of any graph must have an even number of odd-degree vertices, so the two odd vertices $e_1 \cap B_U$ and $e_2 \cap B_U$ must belong to the same component and can be connected by a path p_3 in G.

Each length-$(1/2)$ segment of p_3 belongs to the neighborhood of a single hypercube in U, which, in turn, has as its centroid a point of B, our boundary of S. Let p_4 be the sequence of centroids corresponding to the sequence of edges in p_3. Then p_4 starts at u, and ends at v. Further, at each step, from one edge in p_3 to the next, either the current point from B in p_4 does not change, or it changes from one point in B to an adjacent point (when the corresponding pair of edges in p_4 form a $180°$ angle on two adjoining hypercubes), or it changes from one point in B to a point at distance $\sqrt{2}$ away (when the corresponding edges in p_4 form a $270°$ angle across a concavity on the neighborhood of U). Corresponding to both these changes there is a virtual edge in E_v that can be used to connect the distinct points in p_4.

So, we have constructed a path in E_v between an arbitrarily chosen pair of points u, v in B, and therefore the graph (B, E_v) is connected. □

Theorem 3.8 implies that the d-dimensional mesh can sustain a fault probability inversely polynomial in d and still have a large component whose expansion is no more than a factor of d worse than the original. In the next section, we elaborate on the significance of this result.

4 Conclusion

In this paper, we presented a general technique for determining the robustness of the expansion of different graphs against both adversarial and random faults. For random faults, we have come up with a new parameter, the span, which allows us to prove a strong result regarding the robustness of high dimensional meshes. Among other things, this result can provide useful insights into the robustness of peer-to-peer networks like CAN [28], which behaves like a d-dimensional mesh in its steady state. Basically, we have shown that CAN can tolerate a fault probability which is inversely polynomial in its dimension without losing too much in its expansion properties.

For the 2-dimensional mesh, our result is related to the line of research followed by Raghavan [27], Kaklamanis et. al. [17] and Mathies [26] who show that despite a constant fault probability (of as high as 0.4) a mesh with random failures can emulate a fault free mesh using paths with stretch factor at most $O(\log n)$. Since the diameter in a graph of expansion α is $O(\alpha^{-1} \log n)$ [24], our technique gives essentially the same result albeit with a lower fault probability. Additionally, for meshes of constant dimension greater than 2 our results imply a $O(\log n)$ dilation for path lengths, and hence a way to generalize these earlier results to higher dimensions.

The strength of our technique is that it is able to yield results for the 2-dimensional mesh which are comparable to previous results while giving new results for higher dimensional meshes and providing a general method suitable for analyzing any network whose span can be estimated.

As an aside we note that in an exciting new work, Angel et. al. [4] have shown that the dilation of the mesh does not suffer up to the critical probability and additionally it is possible to find paths between connected vertices which are no more than a constant times the length of the paths between them in the fault-free mesh. However, their work is silent on the question of congestion of path systems in the faulty network.

Open problems

We conjecture that the butterfly, shuffle-exchange, and de Bruijn graph all have a span of $O(1)$, which means that they can tolerate a constant fault probability. Though the span may provide tight results for these graphs, the exponential dependency of the fault probability on the span does not really give useful results if the span is beyond $\log n$. Hence, either a better dependency result or a parameter better than the span is needed. Clearly, as mentioned in the introduction, having a parameter that can accurately describe the fault tolerance of graphs w.r.t. expansion under random faults would be very useful for many applications.

5 Acknowledgments

The authors would like to thank the anonymous referees for their patience with early drafts and for their valuable and detailed comments.

References

[1] M. Ajtai, J. Komlós, and E. Szemerédi. Largest random component of a k-cube. *Combinatorica*, 2(1):1–7, 1982.

[2] N. Alon and F.R.K. Chung. Explicit construction of linear sized tolerant networks. *Discrete Math*, 72:15–19, 1989.

[3] A. Anagnostopoulos, A. Kirsch, and E. Upfal. Load balancing in arbitrary network topologies with stochastic adversarial input. *SIAM Journal on Computing*, 34(3):616–639, 2005.

[4] O. Angel, I. Benjamini, E. Ofek, and U. Wieder. Routing complexity of faulty networks. In *Proc. of 24th Annu. ACM Symp.on Principles of Distributed Computing (PODC '05)*, pages 209–217, 2005.

[5] E. Anshelevich, D. Kempe, and J. Kleinberg. Stability of load balancing algorithms in dynamic adversarial systems. In *Proc. of the 34th ACM Symp. on Theory of Computing (STOC)*, 2002.

[6] S. Arora, S.B. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. In *Proc. of 36th Annual ACM Symp. on Theory of Computing*, pages 222–231, 2004.

[7] M. Ben-Or and D. Ron. Agreement in the presence of faults, on networks of bounded degree. *Information Processing Letters*, 57(6), 1996.

[8] B. Bollobas. The evolution of random graphs. *Transactions of the AMS*, 286:257–274, 1984.

[9] R.J. Cole, B.M. Maggs, and R.K. Sitaraman. Multi-scale self emulation: a technique for reconfiguring arrays with faults. In *Proc. of the 25th ACM Symp. on Theory of Computing (STOC)*, pages 561–572, 1993.

[10] R.J. Cole, B.M. Maggs, and R.K. Sitaraman. Routing on butterfly networks with random faults. In *Proc. of the 36th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 558–570, 1995.

[11] R.J. Cole, B.M. Maggs, and R.K. Sitaraman. Reconfiguring arrays with faults part I: worst-case faults. *SIAM Journal on Computing*, 26(6):1581–1611, 1997.

[12] C. Dwork, D. Peleg, N. Pippenger, and E. Upfal. Fault tolerance in networks of bounded degree. *SIAM Journal on Computing*, 17:975–988, 1988.

[13] P. Erdős and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci.*, 5:17–61, 1960.

[14] B. Ghosh, F.T. Leighton, B.M. Maggs, S. Muthukrishnan, C.G. Plaxton, R. Rajaraman, A.W. Richa, R.E. Tarjan, and D. Zuckermann. Tight analyses of two local load balancing algorithms. *SIAM Journal on Computing*, 29(1):29–64, 1999.

[15] J. Håstad, F.T. Leighton, and M. Newman. Reconfiguring a hypercube in the presence of faults. In *Proc. of the 19th ACM Symp. on Theory of Computing (STOC)*, pages 274–284, 1987.

[16] M. Henle. *A Combinatorial Introduction to Topology*. W. H. Freeman, San Francisco, CA, 1979.

[17] C. Kaklamanis, A.R. Karlin, F.T. Leighton, V. Milenkovic, P. Raghavan, S.B. Rao, and A. Tsantilas. Asymptotically tight bounds for computing with faulty arrays of processors. In *Proc. of the 22nd ACM Symp. on Theory of Computing (STOC)*, pages 285–296, 1990.

[18] A.R. Karlin, G. Nelson, and H. Tamaki. On the fault tolerance of the butterfly. In *Proc. of the 26th ACM Symp. on Theory of Computing (STOC)*, pages 125–133, 1994.

[19] H. Kesten. The critical probability of bond percolation on the square lattice equals $1/2$. *Communication in Mathematical Physics*, 74:41–59, 1980.

[20] S. Khot and N. Vishnoi. The Unique Games Conjecture, integrality gap for cut problems and embeddability of negative type metrics into l_1. In *Proc. 46th Annu. IEEE Symp. on Foundations of Computer Science (FOCS '05)*, pages 53–62, 2005.

[21] F.T. Leighton and B.M. Maggs. Fast algorithms for routing around faults in multibutterflies and randomly-wired splitter networks. *IEEE Transactions on Computers*, 41(5):578–587, 1992.

[22] F.T. Leighton, B.M. Maggs, and S.B. Rao. Packet routing and job-shop scheduling in O(congestion + dilation) steps. *Combinatorica*, 14(2):167–186, 1994.

[23] F.T. Leighton, B.M. Maggs, and R.K. Sitaraman. On the fault tolerance of some popular bounded-degree networks. *SIAM Journal on Computing*, 27(5):1303–1333, 1998.

[24] F.T. Leighton and S.B. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *29th Annual Symposium on Foundations of Computer Science*, pages 422–431, 1988.

[25] T. Luczak, B. Pittel, and J.C. Wierman. The structure of a random graph at the point of the phase transition. *Transactions of the AMS*, 341:721–748, 1994.

[26] T.R. Mathies. Percolation theory and computing with faulty arrays of processors. In *Proc. of the 3rd ACM Symp. on Discrete Algorithms (SODA)*, pages 100–103, 1992.

[27] P. Raghavan. Robust algorithms for packet routing in a mesh. In *Proc. of the 1st ACM Symp. on Parallel Algorithms and Architectures (SPAA)*, pages 344–350, 1989.

[28] S. Ratnasamy, P. Francis, M. Handley, R.M. Karp, and S. Schenker. A scalable content-addressable network. In *Proc. of SIGCOMM 2001*, pages 161–172, 2001.

[29] C. Scheideler. *Universal Routing Strategies for Interconnection Networks*. Lecture Notes in Computer Science 1390. Spinger, 1998.

[30] C. Scheideler. *Probabilistic Methods for Coordination Problems*. HNI-Verlagsschriftenreihe 78, University of Paderborn, 2000.

[31] C. Scheideler. Models and techniques for communication in dynamic networks. In *Proc. of the 19th Symp. on Theoretical Aspects of Computer Science (STACS)*, pages 27–49, 2002.

[32] E. Upfal. Tolerating a linear number of faults in networks of bounded degree. *Information and Computation*, 115(2):312–320, 1994.