Network Failure Detection and Graph Connectivity*

Jon Kleinberg† Mark Sandler‡ Aleksandrs Slivkins§

June, 2003
Minor revision: July 2007.

Abstract

We consider a model for monitoring the connectivity of a network subject to node or edge failures. In particular, we are concerned with detecting (ϵ, k)-*failures*: events in which an adversary deletes up to k network elements (nodes or edges), after which there are two sets of nodes A and B, each at least an ϵ fraction of the network, that are disconnected from one another. We say that a set D of nodes is an (ϵ, k)-detection set if, for any (ϵ, k)-failure of the network, some two nodes in D are no longer able to communicate; in this way, D "witnesses" any such failure. Recent results show that for any graph G, there is an (ϵ, k)-detection set of size bounded by a polynomial in k and ϵ, independent of the size of G.

In this paper, we expose some relationships between bounds on detection sets and the edge-connectivity λ and node-connectivity κ of the underlying graph. Specifically, we show that detection set bounds can be made considerably stronger when parameterized by these connectivity values. We show that for an adversary that can delete $k\lambda$ edges, there is always a detection set of size $O(\frac{k}{\epsilon}\log\frac{1}{\epsilon})$ which can be found by random sampling. Moreover, an (ϵ, λ)-detection set of minimum size (which is at most $\frac{1}{\epsilon}$) can be computed in polynomial time. A crucial point is that these bounds are independent not just of the size of G but also of the value of λ.

Extending these bounds to node failures is much more challenging. The most technically difficult result of this paper is that a random sample of $O(\frac{1}{\epsilon}\log\frac{1}{\epsilon})$ nodes is a detection set for adversaries that can delete a number of nodes up to κ, the node-connectivity.

For the case of edge-failures we use VC-dimension techniques and the cactus representation of all minimum edge-cuts of a graph; for node failures, we develop a novel approach for working with the much more complex set of all minimum node-cuts of a graph.

Keywords. network failires, detection sets, connectivity, minimal cuts, cactus representation, VC-dimension.

AMS subject classification. Primary classification: 68Q25 (CS/ Theory of Computing: analysis of algorithms and problem complexity). Secondary classifications: 68R10 (CS/ Discrete Mathematics in relation to CS: Graph Theory), 05C40 (Combinatorics/ Graph Theory: connectivity).

---

*Preliminary version of this paper appeared in *15th ACM-SIAM Symp. on Discrete Algorithms*, 2004. This work has been completed while the second and the third authors were graduate students at Cornell University. This work has been supported in part by a David and Lucile Packard Foundation Fellowship and NSF ITR/IM Grant IIS-0081334 of Jon Kleinberg.

†Department of Computer Science, Cornell University, Ithaca, NY 14853 Email: *kleinber@cs.cornell.edu*.

‡Google Inc., New York, NY 10011, Email: *sandler@google.com*.

§Microsoft Research, Mountain View, CA 94043. Email: *slivkins at microsoft.com*.

1 Introduction

Monitoring network connectivity. As links or nodes fail in a network, it is important to maintain information about basic properties such as connectivity. For large, unstructured networks, this is often done by recourse to sampling and other approximate measurements; performing such measurements in a robust and accurate way is an active research topic (e.g. [4, 5, 19, 21, 20]). A general problem here is to minimize the cost of network monitoring and measurement, in terms of communication, computation, and resource usage.

Here we consider a model proposed by the first author for monitoring network connectivity [16]. We are given a connected node graph G on n nodes, and we want to detect "failure events" in which at most k network elements (nodes or edges) are deleted, after which there are two sets of nodes A and B, each of size $\geq \epsilon n$, such that no node in A has a path to any node in B. We will call such a pair of sets *separated*, and we will call such an event an (ϵ, k)-*failure*. (To reflect the fact that the k node or edge failures can be arbitrary, we will sometimes speak of them as being selected by an adversary.)

To detect such failures, we consider the strategy of placing "detectors" at a subset D of the nodes of G. Now, if we find that two detectors are unable to communicate — either because there is no path between them, or because one has been deleted — we can record a fault in the network. We would like our set D to have the property that *whenever* an (ϵ, k)-failure occurs, some two detectors are unable to communicate; we will refer to such a set D as an (ϵ, k)-*detection set.* Note the nature of this condition: D must detect all possible (ϵ, k)-failures, so we imagine D as being chosen *before* the adversary selects a set of k network elements to delete. The emphasis in [16] was on finding a bound on the number of nodes that must be randomly selected from a graph G in order to obtain an (ϵ, k)-detection set with high probability. Improvements to these bounds were obtained by [7].[1]

In this paper, we adopt a somewhat different approach to this issue, by exposing some interesting and non-trivial connections between the size of the smallest detection set for a graph G and the values of its edge- and node-connectivity. (The edge-connectivity of G, denoted $\lambda(G)$, is the smallest number of edges that must be deleted in order to disconnect G. The node-connectivity of G, denoted $\kappa(G)$, is the analogous quantity for node deletions.) We show that stronger bounds on detection set size can be obtained if we parameterize these bounds by the connectivity values λ and κ; and for some cases, we use this relationship with connectivity to provide the first *per-instance* guarantees for detection sets.

Because our results are different depending on whether the adversary is deleting edges or nodes, we consider these two cases separately.

Detection sets for edge failures. We begin with adversaries that can delete up to k edges; as such, we will be concerned with (ϵ, k)-*edge-failures*, which are (ϵ, k)-failures in which only edges are deleted. It is known that a random set of $O(\frac{k}{\epsilon} \log \frac{1}{\epsilon})$ nodes is an (ϵ, k)-detection set for edge failures with high probability [16], and that every graph contains an (ϵ, k)-detection set for edge failures of size $O(\frac{k}{\epsilon})$ [7]; note that both bounds are independent of the size of the graph G. It is not difficult to show that both bounds are tight, and so there is no prospect of obtaining an improvement that applies to all graphs. However, it makes sense to ask whether better bounds are possible in terms of natural parameters of the graph G.

An obvious parameter to consider here is the edge-connectivity λ; indeed, there can be no (ϵ, k)-edge-failures in G if $k < \lambda$. Our first main result establishes that λ is indeed a natural way to parameterize the problem; we show that every graph G has an (ϵ, λ)-detection set for edge failures of size at most $\frac{1}{\epsilon}$. Note that there is no leading constant in this bound, and that it is independent not just of the size of G but also of the value of λ. Extending this result, we show further that an (ϵ, λ)-detection set for edge failures of

[1] Following the publication of the conference version of this paper, further improvements have been obtained in [8].

minimum size for a graph G can be computed in polynomial time. The algorithms used to establish these results are based on the cactus representation of all minimum edge-cuts of G [6, 9].

Given that strong bounds are possible for detecting an adversary that can delete one minimum cut's worth of edges, it is natural to ask what can be said about an adversary capable of deleting a number of edges equal to k times the size of a minimum cut. We show that a random set of $O(\frac{k}{\epsilon} \log \frac{1}{\epsilon})$ nodes is a $(k\lambda, \epsilon)$-detection set for edge failures with high probability. This is essentially a factor of λ times stronger than the bounds of [7, 16], which did not take edge connectivity into account. Our proof of this result uses a VC-dimension argument in the style of [16]; the bound on the VC-dimension is obtained using a result of Mader [18, 10]. that extends results of Lovász [17] and of Cherkasskij [3] on edge-disjoint paths in graphs.

Detection sets for node failures. We now consider adversaries that can delete up to k nodes. By analogy with our results for edge failures, we consider the size of detection sets in terms of the node-connectivity κ. Our main result here is that every graph G (with $\kappa = O(\epsilon^2 n)$) has an (ϵ, κ)-detection set for node failures of size $O(\frac{1}{\epsilon})$; moreover, a random set of $O(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$ nodes forms an (ϵ, κ)-detection set for node failures with high probability. Again, note that these bounds are independent not just of the size of G but also of the value of κ. Extending our results to adversaries that delete $k\kappa$ nodes for $k > 1$ is a very interesting and apparently difficult open question.

We note the distinction, raised by Gupta [13], between *strong* and *weak* detection sets for node failures. A strong detection set D has the property that, after any (ϵ, k)-node-failure, two nodes of D lie in different connected components. A weak detection set D' has the property that, after any (ϵ, k)-node-failure, two nodes of D lie in different connected components *or* an element of D has been deleted. Either of these definitions arguably forms a plausible definition of network failure detection. Improving a bound of [16], Fakcharoenphol showed that a random set of $O(\frac{k}{\epsilon} \log k \log \frac{1}{\epsilon})$ nodes is a strong (ϵ, k)-detection set for node failures [7], and Gupta showed that every graph has a weak (ϵ, k)-detection set for node failures of size $O(\frac{k}{\epsilon})$. As we note in Section 4, weak detection appears to be a more useful notion when the problem is parameterized by node connectivity; in particular, our main result is about weak detection sets. Henceforth, we will assume that all detection sets for node failures are weak unless otherwise specified.

Our analysis for node failures is significantly more complicated than for edge failures, and this is not surprising; not only is no analogue of the cactus representation known for min-node-cuts, but this appears to be intrinsic due to the $\#P$-completeness of even counting the number of min-node-cuts [2]. Indeed, given the lack of tractable representations for min-node-cuts, we believe that our analysis develops some useful properties of their structure. We begin by constructing a detection set of minimum size for adversaries that can delete *shredders* [2, 15] — min-node-cuts whose deletion produces at least three components. The construction of the detection set then proceeds by greedily isolating a maximal collection of relatively balanced min-node-cuts that produce just two components, and whose "small sides" are disjoint; the small side of each such cut is required to have at least $\frac{\epsilon n}{10}$ nodes. We then show that by placing detectors so that one lies on the small side of each of these cuts, there is no way for a min-node-cut producing two components of size at least ϵn each to avoid being detected.

Further Discussion. A simple calculation based on Karger's algorithm gives an upper bound of $O(\frac{k}{\epsilon} \log n)$ on a random sample of nodes that forms an $(\epsilon, k\lambda)$-detection set for edge failures.[2] However, our goal in this paper is to find bounds that do not depend on the size of the graph.

Following [16], we can extend our results to a model in which the nodes of the network G are partitioned into two sets — a set V_0 of *end nodes* and a set V_1 of *internal nodes*. We assume that we are only allowed to place detectors at end nodes, and correspondingly are only interested in monitoring the connectivity of the

[2]Note that no such simple bound is available for the case of node-failures, which is yet another evidence of its difficulty.

end nodes. Specifically, we re-define (ϵ, k)-failures as failures of $\leq k$ network elements, after which two disjoint subsets of V_0, each of size $\geq \epsilon |V_0|$, are separated from each other. We can show that the bounds obtained above carry over to this more general setting; we omit further discussion of the generalization from this version of the paper.

Our work is similar in spirit to some of the work on vertex connectivity and augmentation thereof, e.g. [14, 2, 15, 12]. The actual technical issues are quite different, however, since we are only interested in balanced cuts. In general one could view our work here as integrating notions from edge- and node-connectivity with the problem of balanced separators of graphs — two topics that have traditionally been approached separately due to their great differences in tractability.

Notation. In this paper all graphs are assumed undirected; our standard notation for a graph is $G = (V, E)$. An edge(node)-cut is a set X of edges (nodes) such that $G \setminus X$ is disconnected.

A min-edge(node)-cut is an edge(node)-cut of minimum size. This size is also known as edge(node)-*connectivity* and denoted by λ and κ respectively. We will write *min-cut* when it is clear whether we are talking about edge-cuts or node-cuts. A set of nodes is *tight* if it is a union of some (but not all) components of a min-cut. A cut X is called ϵ*-balanced* if there are two sets of vertices of size $\geq \epsilon n$ that are disconnected from one another in $G \setminus X$. An ϵ-balanced cut of $\leq k$ edges(nodes) is called an (ϵ, k)-cut.

If sets X, Y have a non-empty intersection, we say X *meets* Y. To help clarify the notation in places, we will sometimes write $X + Y$ to denote the union of disjoint sets X and Y, and $X - Y$ to denote the difference of sets X and Y for which $Y \subseteq X$.

2 Detection sets for edge failures

In this section we present our results on edge failures. For edge failures that correspond to min-edge-cuts, our algorithms are based on the *cactus representation* of all min-cuts in a graph [6, 9]. We include a self-contained review of the relevant definitions and facts. Our result for the general edge-failures proof uses a VC-dimension argument in the style of [16]; the bound on the VC-dimension is obtained using a result of Mader [18, 10] on edge-disjoint paths in graphs.

Throughout the section, all cuts are edge-cuts, and all detection sets are for edge failures. Let D be a set of nodes, representing the locations of our detectors. We say that D *detects* a cut X if some pair of detectors is separated in $G \setminus X$. We call D an (ϵ, k)*-detection set* if it detects every (ϵ, k)-edge-cut.

In Section 2.1 we review the cactus representation. Section 2.2 is on min-edge-cuts: we construct a smallest (ϵ, λ)-detection set and prove it has size $\leq \frac{1}{\epsilon}$. Section 2.3 is on general edge-failures: we prove that a set of $O(\frac{k}{\lambda\epsilon} \log \frac{1}{\epsilon})$ randomly sampled nodes is an (ϵ, k)-detection set with high probability.

2.1 Review: cactus representation

Edges will be viewed as cycles of length two; cycles of length 3 or more are called *proper*. A *cactus* is a connected graph such that any two of its cycles have at most one vertex in common. An arbitrary cactus can be obtained starting from a cycle and recursively adding new cycles that share a single vertex with the existing graph. In a cactus, some edges are contained in a proper cycle (*cycle edges*), and some aren't (*path edges*). Each cycle edge has capacity $\frac{1}{2}$, and each path edge has capacity 1. It follows that min-cuts of a cactus have capacity 1. We can characterize them as follows:

Fact 2.1 *Consider a cactus T. Then (a) each path edge is a min-cut, (b) any pair of cycle edges from the same cycle is a min-cut, and (c) there are no other min-cuts.*

Proof: Clearly any cut in T has capacity at least 1. For part (a), let uv be a path edge. If there exists a uv-path p not containing the edge uv, then $p + uv$ is a cycle, contradicting the definition of a path edge. Therefore u and v are separated in $T - uv$. So uv is a cut in T, hence a min-cut.

For part (b), let e_1, e_2 be cycle edges from the same cycle C. $e_1 + e_2$ splits C into two arcs, call them C_1 and C_2. Suppose C_1 and C_2 are connected in $T - e_1 - e_2$. Then there exist vertices $u \in C_1$, $v \in C_2$ such that there is a uv-path p that does not intersect with C except for the endpoints. Let C' be the uv-arc of C that contains e_1. Then $p + C'$ is a cycle in T that shares ≥ 2 vertices with C, contradiction. So C_1 and C_2 are not connected in $T - e_1 - e_2$. Therefore $e_1 + e_2$ is a cut in T, hence a min-cut.

For part (c), suppose X is a min-cut of T that is neither a path edge nor a pair of cycle edges from the same cycle. Since the capacity of X is ≤ 1, it consists of one or two cycle edges. So there is a (proper) cycle C such that X contains exactly one edge $uv \in C$. Since X is a min-cut, it must separate u and v. However, they are connected by $C - uv$. Contradiction. □

In a cactus, nodes of degree one will be called *leaves*, nodes of degree two that are contained in a cycle will be called *cycle nodes*, and all other nodes will be called *branch nodes*.

Fact 2.2 *Let v be a branch node of a cactus T. Then the cycles that contain v are pairwise disconnected in $T - v$.*

Proof: Let C, C' be cycles that contain v. Let uv, $u'v$ be edges in C, C', resp. Suppose u and u' are connected in $T - v$, i.e. there is a uu'-path p not containing v. Then $p + uv + vu'$ is a cycle that shares ≥ 2 vertices with C (and C'), contradiction. So C and C' are disconnected in $T - v$. □

Consider a branch node v of a cactus T. It connects two or more cycles. By Fact 2.2, the removal of v splits T into two or more connected components (v*-components*). Each v-component X is tight: for some cycle C containing v, it is obtained by removing the edge(s) of C that are adjacent to v.

Fact 2.3 *Suppose S is a tight set in cactus T, v is a branch node. Then:*

(a) if $v \in S$ then S contains at least one v-component.

(c) if $v \notin S$ then S is contained in a v-component.

(c) for any v-component X of T, either $X \subset S$, or $S \subset X$, or $X \subset V - S$, or $V - S \subset X$.

Proof: For part (a), let S be a component of a min-cut C. By Fact 2.1 C is contained in a cycle, so $C \subset T[X + v]$ for some v-component X. Therefore if Y is any other v-component then $Y + v$ is connected in $T \setminus C$. $Y \subset S$ follows since $v \in S$ and S is connected in $T \setminus C$.

For part (a), suppose S meets two v-components then they are connected in $T - v$ (via S), contradiction.

For part (c), suppose X meets both S and $V - S$. Then by part (b) if $v \in S$ then $V - S \subset X$, else we have $S \subset X$. □

Let G be a weighted graph on n vertices. A *cactus-pair* of G is a pair (T, π) where T is a cactus, and π is a mapping from $V(G)$ to $V(T)$ such that if M is a tight set in T then $\pi^{-1}(M)$ is a tight set in G. For each tight set M of T say that (T, π) *represents* the min-cut C of G such that $\pi^{-1}(M)$ is a C-component. A *cactus representation* of G is a cactus-pair of G that represents all min-cuts of G. Dinits et al. [6] proved that every capacitated graph has a cactus representation of size $O(n)$. Further results show that a cactus representation of size $O(n)$ can be efficiently constructed. See the introduction of [9] for discussion.

2.2 Detection sets for min-edge-cuts

Here we are only interested in ϵ-balanced min-cuts, and so the cactus representation is too general for our purposes. This motivates the following definitions.

Definition 2.4 *Let an* ϵ-cactus-pair *be a cactus-pair that represents all* ϵ*-balanced min-cuts. Let an* ϵ-cactus *be the cactus in such cactus-pair (if the mapping is clear). A subset* S *of vertices of a cactus is* heavy *if* $|\pi^{-1}(S)| \geq \epsilon n$*. Call a cactus-pair* reduced *if every* v*-component is heavy.*

A reduced ϵ-cactus-pair can be efficiently computed from a standard cactus representation by consecutively applying the following reduction.

Lemma 2.5 *Suppose* T *is an* ϵ*-cactus,* v *is a branch node,* X *is a* v*-component that is not heavy. Let* T' *be* T *with* X *contracted into* v*. Then* T' *is also an* ϵ*-cactus.*

Proof: For each ϵ-balanced min-cut C of G there is a min-cut C' of T that represents it. By Fact 2.3c there is a component S of C' such that $X \subset S$ or $S \subset X$. Since S is heavy and X isn't, it must be the case that X is a proper subset of S. Then $v \in S$, so C' is a min-cut in T', too. Therefore T' represents C. □

Let G be a capacitated graph. Let (T, π) be a reduced ϵ-cactus-pair of G. We will characterize (ϵ, λ)-detection sets of minimum size in terms of T.

Let a *subcycle* be a set of consecutive cycle nodes of a (proper) cycle in T. Consider the non-degenerate case when there is at least one branch node. Then the weight $|\pi^{-1}(\cdot)|$ of each leaf and each subcycle is at most $(1-\epsilon)n$. Let a *canonical subcactus* be a set of nodes of T that contains each leaf, has an element in every heavy subcycle, and contains no branch nodes. Let $D \subset V(G)$ be a set of detectors (not necessarily an (ϵ, λ)-detection set). Say D is T*-canonical* if $\pi(D)$ is a canonical subcactus, and at most one detector is mapped to each node of T. The following two lemmas show that any smallest (ϵ, λ)-detection set is in fact a smallest T-canonical set.

Call $S \subset V$ *heavy* if $|S| \geq \epsilon n$, and *balanced* if both S and $V \setminus S$ are heavy. Call $S' \subset V(T)$ *balanced* if $\pi^{-1}(S')$ is balanced. For each balanced tight set S of G let $\pi'(S)$ be a (balanced) tight set S' of T such that $S = \pi^{-1}(S')$.

Lemma 2.6 *Any smallest* (ϵ, λ)*-detection set is* T*-canonical.*

Proof: Let D be a smallest (ϵ, λ)-detection set. Call elements of D *detectors*. We need to show that (1) at most one detector is mapped to each node of T, (2) there is a detector mapped to each leaf and each heavy subcycle of T, (3) and no detectors are mapped to branch nodes of T. Let us prove these three statements in order.

(1) Suppose two detectors d_1, d_2 map to a node v of T. To obtain a contradiction it suffices to show an (ϵ, λ)-detection set smaller than D. We claim that $D - d_1$ is also an (ϵ, λ)-detection set. Suppose not. Then there is a balanced tight set S of G that contains $D - d_1$. Obviously $d_1 \notin S$. Let $S' = \pi'(S)$. Since $d_2 \in S$, $v = \pi(d_2) \in S'$, so $d_1 \in S$, too, a contradiction.

(2) There is a detector mapped to each heavy tight set of T, in particular, to each leaf and each heavy subcycle.

(3) Suppose a detector d is mapped to a branch node v of T. By analogy with (1), we claim that $D - d$ is also an (ϵ, λ)-detection set. For suppose not. Then $D - d$ is disjoint with some balanced tight set S. Let $S' = \pi'(S)$. Since D is an (ϵ, λ)-detection set, $d \in S$, so $v \in S'$. Therefore by Fact 2.3a S' contains some v-component S''. Since T is reduced, S'' is heavy, so there is a detector mapped to it. So S contains a detector other than d, a contradiction. □

Lemma 2.7 *Any* T*-canonical set is an* (ϵ, λ)*-detection set.*

Proof: Suppose $D \subset V$ and $\pi(D)$ meets each leaf and each heavy subcycle of T. We need to prove that $\pi(D)$ meets each heavy tight set of T. To show this we claim that any heavy tight set S of T contains a leaf or a heavy subcycle.

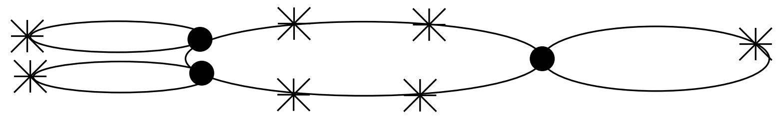


Figure 1: An ϵ-cactus with detectors. Branch nodes are denoted by '•', detectors by '*'. In the central cycle, there are three subcycles between the branch nodes. The smallest of them is not heavy, hence does not contain a detector. The other two are big enough so that they need two detectors each. Each of the three smaller cycles is heavy (even without its branch node), since otherwise it would have been contracted.

We'll use induction on the size of S. The base case corresponds to an S that consists of one vertex, say v. By Fact 2.3a v cannot be a branch node. So either v is a leaf or it is a heavy subcycle consisting of a single cycle node.

For the induction step, note that if S contains a branch node v then by Fact 2.3a S contains some (heavy) v-components S', to which the induction hypothesis applies. If S does not contain any branch nodes, then it lies within a single cycle, so S is a (heavy) subcycle. The claim follows. □

Theorem 2.8 *A smallest (ϵ, λ)-detection set is of size at most $\frac{1}{\epsilon}$. There is a polynomial-time algorithm to construct it.*

Proof: Let (T, π) be a reduced ϵ-cactus-pair of G. We have seen that smallest (ϵ, λ)-detection sets are (mapped to) smallest canonical subcacti of T. Therefore it suffices to compute a smallest canonical subcactus of T.

Let S be a subset of a proper cycle C in T. Call S a C-detection set if S does not contain any branch nodes, and every heavy subcycle of C contains an element of S. By definition, if there are no heavy subcycles in C then an empty set is a C-detection set. Obviously, a subset of T is a canonical subcactus iff it is a union of leaves of T and (disjoint) C-detection sets, one for each proper cycle of T. Therefore to compute a smallest canonical subcactus of T it suffices to construct a smallest C-detection set for each proper cycle C of T.

The construction is as follows. Assuming T consists of more than one cycle, C contains one or more branch nodes. Assuming C contains cycle nodes, pick any branch node v_b followed by a cycle node v. Start with v. In the iterative step, start with a cycle node and move clockwise along C till a heavy subcycle is detected (call this subcycle *selected*) or a branch node is reached. Start a new step with the next cycle node. Stop when v_b is reached. Let S be the set of the last nodes (clockwise) of selected subcycles.

Obviously S is a C-detection set. S is a smallest such set by the following observation. Let S' be a C-detection set. Let $v \in C$ be a branch node or an element of S'. Let v' be the next node clockwise. Let C' be the smallest heavy subcycle starting with v', if it exists. Let w be the last node of C'. Then C' contains at least one element of S. The observation is that $S' - C' + w$ is a C-detection set with the same or smaller number of elements. Consecutively applying this observation, we can transform S' to S without increasing the number of detectors.

Our construction puts one detector into each leaf of T and each selected subcycle. Since leaves of T are heavy and selected subcycles are heavy and disjoint, our construction covers at least ϵn weight with each detector. Since the total weight of (nodes of) T is n, the total number of detectors is at most $\frac{1}{\epsilon}$. □

2.3 Smaller detection sets for edge failures

A set S of nodes is *k-edge-separable* if there exists a set Z of $\leq k$ edges such that S is a union of components of $G \setminus Z$. Let $\mathcal{F}$ be the family of all k-edge-separable sets. We say that $A \subseteq V$ is *shattered* by $\mathcal{F}$ if for all

$B \subseteq A$ there exists an $F \in \mathcal{F}$ such that $B = A \cap F$. The *VC-dimension* of $\mathcal{F}$ is defined to be the maximum cardinality of a subset of V that is shattered by $\mathcal{F}$.

In [16], it was shown that one can connect the VC-dimension d of $\mathcal{F}$ with (ϵ, k)-detection sets via the notion of an *ϵ-net*, which is a set that meets each $F \in \mathcal{F}$ of size $\geq \epsilon n$. Specifically, a theorem by [1] says that a set of $O(\frac{d}{\epsilon}\log\frac{1}{\epsilon} + \frac{1}{\epsilon}\log\frac{1}{\delta})$ randomly sampled nodes is an ϵ-net for $\mathcal{F}$ with probability at least $1-\delta$.[3] Moreover, it is easy to show [16] that an ϵ-net for $\mathcal{F}$ is an (ϵ, k)-detection set.

In [16], it was shown that the VC-dimension of $\mathcal{F}$ is at most $2k+1$, yielding a bound of $O(\frac{k}{\epsilon}\log\frac{1}{\epsilon})$ on the size of an (ϵ, k)-detection set. In this section, we strengthen the VC-dimension bound on $\mathcal{F}$ to $O(\frac{k}{\lambda})$. Thereforem we obtain the following theorem.

Theorem 2.9 *A set of $O(\frac{k}{\lambda\epsilon}\log\frac{1}{\epsilon})$ randomly sampled nodes is an (ϵ, k)-detection set with high probability.*

We now turn to the new bound on the VC-dimension; to prove it, we will use the following theorem by Mader [18] on edge-disjoint paths between elements of a given set of vertices. Let R be a subset of V of size r. Let $d(R)$ be the number of edges leaving R. Let $q(R)$ be the number of components C of $G - R$ for which $d(C)$ is odd. Let an *R-path* be a path connecting distinct elements of R.

Theorem 2.10 [18] *The maximal number of edge-disjoint R-paths is $\frac{1}{2}\min(\sum d(V_i) - q(\cup V_i))$, where the minimum is taken over all collections of disjoint subsets of vertices $V_1, V_2, \ldots, V_r$ such that $|V_i \cap R| = 1$.*

Corollary 2.11 *There are $\Omega(r\lambda)$ edge-disjoint R-paths.*

Proof: Consider a collection of disjoint subsets of vertices $V_1, V_2, \ldots, V_r$ such that $|V_i \cap R| = 1$. Let $d = \sum d(V_i)$, $q = q(\cup V_i)$. By the above theorem it suffices to prove that $d - q = \Omega(r\lambda)$.

Note that $d \geq r\lambda$ since $d(V_i) \geq \lambda$. Let $C_1 \ldots C_q$ be the components C of $G - \cup V_i$ such that $d(C)$ is odd. All edges exiting each C_i are to $\cup V_i$. So $d \geq d(\cup V_i) \geq \sum d(C_i) \geq q\lambda$. If $r \geq q$ then $d - q \geq r\lambda - q \geq r(\lambda - 1)$. If $r < q$ then $d - q \geq q\lambda - q \geq r(\lambda - 1)$. Therefore $d - q = \Omega(r\lambda)$. $\square$

The following is a well-known application of the probabilistic method.

Lemma 2.12 *Let (R, F) be a multi-graph on R. Then there exists a partition of R into sets R_1, R_2 such that there are at least $\frac{1}{2}|F|$ edges between R_1 and R_2.*

Lemma 2.13 *The VC-dimension of $\mathcal{F}$ is $O(\frac{k}{\lambda})$.*

Proof: Let R be a subset of V of size r. By Cor. 2.11 there exists a family $\mathcal{P}$ of $\Omega(rc)$ edge-disjoint R-paths. Let (R, F) be a multi-graph on R such that there is a 1-1 correspondence between uv-paths in $\mathcal{P}$ and edges $uv \in F$. By Lemma 2.12 there exists a partition of R into sets R_1, R_2 such that (in the original graph) there are $\Omega(r\lambda)$ edge-disjoint paths between R_1 and R_2. We can choose $r = \Theta(\frac{k}{\lambda})$ so that there is guaranteed to be a family $\mathcal{P}'$ of (at least) $k+1$ edge-disjoint paths between R_1 and R_2.

We claim that R cannot be shattered by $\mathcal{F}$. Suppose not. Then there exists $X \in \mathcal{F}$ such that $X \cap R = R_1$. X is a union of components of some cut Z of k or less edges. Z is disjoint with (at least) one path $p \in \mathcal{P}'$. The ends of p are in the same Z-component, so either they are both in X, or both not in X. In both cases this contradicts $X \cap R = R_1$. Thus, the claim is proved, and it follows that the VC-dimension of $\mathcal{F}$ is $r = O(\frac{k}{\lambda})$. $\square$

[3] Both [16] and [7] used a slightly weaker theorem, with a corresponding bound of $O(\frac{d}{\epsilon}\log\frac{d}{\epsilon} + \frac{1}{\epsilon}\log\frac{1}{\delta})$.

3 Detection sets for node failures

The main theorem of this section (Theorem 3.6) is that for $\kappa < O(\epsilon^2 n)$ a set of $O(\frac{1}{\epsilon}\log\frac{1}{\epsilon})$ randomly sampled nodes is a weak (ϵ,κ)-detection set with high probability. We rely on a special case of ϵ-shredders, which is a corollary of our result on *strong* detection thereof (Theorem 3.1). We also present a partial result (Theorem 3.15) on extending strong detection sets for ϵ-shredders to those for general (ϵ,κ)-cuts.

Before we proceed, let's review the definitions. In this section all cuts are node-cuts, all detection sets are for node failures. A cut X is called *two-way* if $G \setminus X$ has exactly two connected components, called the *sides* of X. A *shredder* is a min-cut with three or more components. An ϵ*-shredder* is an ϵ-balanced shredder. A set D of nodes *strongly detects* a cut X if some pair of detectors is separated in $G \setminus X$. If D either meets or strongly detects X, say D *weakly* detects X. D detects (is a detection set for) a family of cuts if it detects every cut in the family.

The rest of this section is organized as follows. In Section 3.1 we show how to find a strong detection sets for ϵ-shredders. In Section 3.2 we use shredders to get a detection set for two-way ϵ-balanced min-cuts. In Section 3.3 we combine these two results and obtain the main theorem. Finally, in Section 3.4 we present our partial result on strong detection sets.

3.1 Strong detection sets for shredders

It is a well-known fact that there can be exponentially many min-cuts. Furthermore, even *counting* min-cuts is #P-complete [2]. However, there can be only $O(n)$ shredders [15], with a polynomial-time enumeration algorithm [2]. We start by stating the main result of this subsection.

Theorem 3.1 *Suppose $\kappa < \epsilon n$. Then a set of $O(\frac{1}{\epsilon}\log\frac{1}{\epsilon\delta})$ randomly sampled nodes is a strong detection set for ϵ-shredders with probability at least $1-\delta$. Moreover, a* smallest *strong detection set for ϵ-shredders has size $\leq \frac{1}{\epsilon}$ and can be constructed in polynomial time.*

Before we prove this theorem we need to establish some basic facts about min-cuts. For a cut X the connected components of $G \setminus X$ are also called X*-components*. Let S, T be min-cuts. Say S *meshes* T if S meets at least two T-components. By [2, Lemma 4.3(1)] if S meshes T then T meets every S-component. Thus meshing is a symmetric relation. If S meshes T (and T meshes S), the two cuts are *meshing*. Else S and T are *non-meshing*.

Lemma 3.2 ([2], Lemma 4.3(2)) *If min-cuts S and T are meshing, then there is a component Q of either S or T such that Q contains $V - S - T$.*

Corollary 3.3 *If $\kappa < \epsilon n$ then any two ϵ-shredders are non-meshing.*

Lemma 3.4 *Let S and T be non-meshing shredders. Let C be the S-component that meets T. Then C contains all T-components but one, call it C'. Moreover, C' contains $V - S - C$, i.e. all S-components other than C.*

Proof: Pick any $v \in S - T$. By minimality of S, v has edges to each S-component (else, $S - v$ is a cut). Thus, $V - S - C + \{v\}$ is connected. Since $T \subset S \cup C$, $V - T - C$ is connected, and hence lies in a T-component C'. So all other T-components are contained in C and $V - S - C \subset V - T - C \subset C'$. □

For a family $\mathcal{F}$ of ϵ-shredders, we call a component of a shredder an $\mathcal{F}$*-head* if it meets at least one shredder in $\mathcal{F}$. Now, suppose we have an (ϵ, k)-detection set for shredders, and S is an ϵ-shredder with an $\mathcal{F}$-head H. Then there exists $T \in \mathcal{F}$ that meets H; so by Lemma 3.4 H contains all T-components but one, and hence contains a detector. This gives the following lemma.

Lemma 3.5 *Let $\mathcal{F}$ be a family of ϵ-shredders, with $\kappa < \epsilon n$, and let S be an ϵ-shredder with an $\mathcal{F}$-head H. Then any detection set for $\mathcal{F}$ meets H.*

Proof of Theorem 3.1: Let $\mathcal{F}_0$ be the family of all ϵ-shredders. Start with $\mathcal{F} = \mathcal{F}_0$. While there exists an ϵ-shredder $S \in \mathcal{F}$ with two or more $\mathcal{F}$-heads, delete S from $\mathcal{F}$. Let $\mathcal{F}_1$ be the resulting family of shredders. By Lemma 3.5 any strong detection set for $\mathcal{F}_1$ is a strong detection set for $\mathcal{F}_0$.

Let $S \in \mathcal{F}_1$. Let the *head* H of S be the (single) $\mathcal{F}_1$-head of S. Let the *tail* of S be $V - S - H$. Note that by Lemma 3.4 for any $S, T \in \mathcal{F}_1$ the tail of S is contained in the head of T (and vice versa). In particular, tails are pairwise disjoint. Since each head contains someone else's tail, a set D of nodes is a detection set for $\mathcal{F}_1$ iff D meets the tail of each $S \in \mathcal{F}_1$. Therefore, a smallest detection set for $\mathcal{F}_1$ has size $|\mathcal{F}_1|$. Since tails are of size $\geq \epsilon n$ each, $|\mathcal{F}_1| \leq \frac{1}{\epsilon}$. The random sampling result follows by a simple probabilistic computation. □

3.2 Detecting two-way min-cuts

In this subsection we construct a weak detection set for two-way (ϵ, κ)-cuts. First we give a non-efficient deterministic construction. We consider $(\frac{\epsilon}{10}, \kappa)$-cuts and use a greedy-type algorithm to construct a "maximal" family of two-way $(\frac{\epsilon}{10}, \kappa)$-cuts with sides A_i and B_i such that $A_i \subseteq B_j$ for all $i \neq j$. In particular A_i's are pairwise disjoint, so there are at most $\frac{10}{\epsilon}$ of them. It turns out that if $\kappa < O(\epsilon^2 n)$ then putting a detector into each A_i suffices. More precisely we show (Theorem 3.8) that these detectors together with any weak detection set for shredders give a weak (ϵ, κ)-detection set. Then a simple probabilistic argument yields a randomized result stated below.

Theorem 3.6 *Suppose $\kappa < \frac{\epsilon^2 n}{20}$. Then a set of $O(\frac{1}{\epsilon} \log \frac{1}{\epsilon\delta})$ randomly sampled nodes is a weak (ϵ, κ)-detection set with probability at least $1 - \delta$.*

We start with some notation and a simple but very useful lemma about crossing min-cuts. Let S be a set of nodes. Call S *connected* if the subgraph of G induced by S is connected. Else say S is *disconnected*. Say a cut X *preserves* S if X disjoint with S and S lies in one component of $G \setminus X$. Note that a connected set of nodes is preserved by X if and only if it is disjoint with X. $N(S)$ denotes the set of neighbors of S, i.e. the set of all nodes in $V - S$ that have an edge to S. Note that if $V - S - N(S)$ is non-empty then $N(S)$ is a cut.

Say two-way min-cuts X and Y are *strongly crossing* if each side of X meets each side of Y. Say X and Y are *weakly crossing* if X meets both sides of Y and vice versa.[4] It is easy to see that strong crossing implies weak crossing, but not the other way round.

To formulate the promised lemma, we will use the following notation. The sides of X and Y are respectively P_1, P_2 and Q_1, Q_2. Their intersections ("quarters") are $C_{ij} = P_i \cap Q_j$. Also let $X_i = Q_i \cap X$ and $Y_i = P_i \cap Y$ and $X \cap Y = S$.

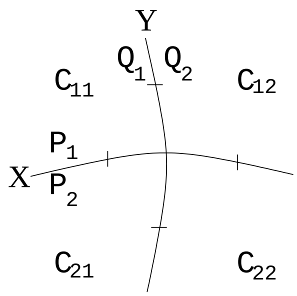
Lemma 3.7 (The Two-Quarters Lemma) *Suppose two-way min-cuts X and Y are weakly crossing so that the two quarters C_{21} and C_{12} are non-empty. Then*

(a) $|X_1| = |Y_1|$ *and* $|Y_2| = |X_2|$,

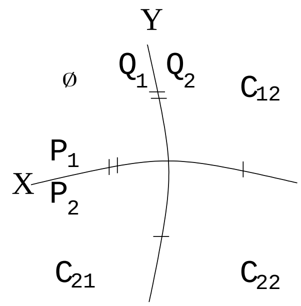
(b) C_{21} *and* C_{12} *are tight, with* $N(C_{ij}) = Y_j + X_i + S$,

(c) $V - C_{21} - N(C_{21})$ *is connected, same for* C_{12}.

[4] Note that if X meets both sides of Y, say at v_1 and v_2, respectively, then Y meets both sides of X. Indeed, for the sake of contradiction suppose Y does not meet a side P_1 of X. Then, since any node in X has at least one edge to P_1 and P_1 is connected, there is a $v_1 v_2$ path in G/Y, contradiction.



(a) X and Y are strongly crossing



(b) X and Y are weakly crossing

Figure 2: Two applications of the Two-Quarters Lemma.

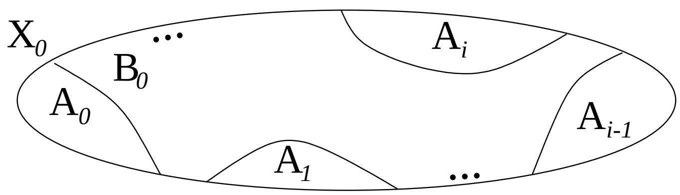


Figure 3: Partitioning of the graph after the ith iteration of the algorithm

Proof: $T = X_1 + Y_2 + S$ and $U = X_2 + Y_1 + S$ separate C_{21} and C_{12} respectively from the rest of the graph. It follows that $Y_2 \geq X_2$ (else $|T| < |X|$), $X_1 \geq Y_1$ (else $|T| < |Y|$), $X_2 \geq Y_2$ (else $|U| < |Y|$) and $Y_1 \geq X_1$ (else $|U| < |X|$). Therefore $|X_1| = |Y_1|$ and $|X_2| = |Y_2|$, so U and T are min-cuts and C_{12} and C_{21} are tight. Finally, $V - C_{21} - N(C_{21})$ is connected as a union of two connected sets (Q_1 and P_2) with a non-empty intersection (C_{12}). □

This lemma is similar to the result of Jordán [14] on intersecting tight sets. Note that if X and Y are strongly crossing our lemma yields $|X_1| = |X_2| = |Y_1| = |Y_2|$ (Figure 2a). We will also use it for $\frac{\epsilon}{10}$-balanced min-cuts that are crossing weakly but not strongly. Then one of the "quarters", say C_{11}, is empty, so, assuming $\kappa < \frac{\epsilon n}{10}$, C_{21} and C_{12} are not (Figure 2b).

Now we are ready to describe the construction.

Construction

1. Let $\mathcal{F}$ denote family of all $\frac{\epsilon}{10}$-balanced two-way min-cuts, and let $\mathcal{A}(\mathcal{F})$ denote the family of the sides of all $F \in \mathcal{F}$. Stop if $\mathcal{F}$ is empty.

2. Choose any inclusion-wise minimal component A_0 from $\mathcal{A}(\mathcal{F})$, let $X_0 = N(A_0)$ be the corresponding cut and B_0 be the second component of X_0. Put detectors in A_0 and B_0.

3. Delete from $\mathcal{F}$ all cuts which do not preserve A_0. For $X \in \mathcal{F}$ let $A(X)$ be the side of X that *does not* contain A_0.

4. Start with the first iteration. For the i-th iteration choose a cut $X_i \in \mathcal{F}$ so that $A(X_i)$ does not contain any other $A(X)$ for $X \in \mathcal{F}$. Let $A_i = A(X_i)$. Let B_i be the other side of X_i.

5. Put a detector into A_i. Remove from $\mathcal{F}$ all cuts which do not preserve $A_0 \cup A_1 \cup \cdots \cup A_i$. Stop If $\mathcal{F}$ is empty; else iterate.

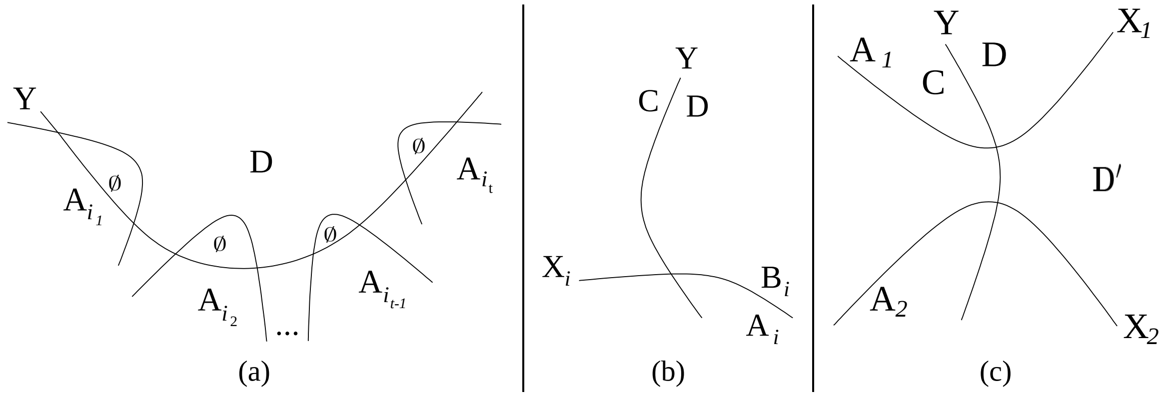


Figure 4: Three different options of how Y can interact with X_i's. For (c) we prove that the portion of Y between cuts X_1 and X_2 shrinks to an empty set, and $X_1 \cap Y = X_2 \cap Y$.

By construction all A_i's are pairwise disjoint, and each $A_i \geq \frac{\epsilon}{10}$. Therefore our algorithm will terminate after at most $\frac{10}{\epsilon}$ steps after putting at most $\frac{10}{\epsilon}$ detectors. Denote this set of detectors by $\mathcal{D}_2$. Let $\mathcal{D}_1$ be any weak detection set for shredders, $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$.

Theorem 3.8 *If $\kappa \leq \frac{\epsilon^2}{20} n$ then any ϵ-balanced two-way min-cut is weakly detected by $\mathcal{D}$.*

Before proving this theorem we will state some simple properties of our construction.

Lemma 3.9 *For all $i \neq j$ $A_j \subseteq B_i$. In particular X_i is disjoint with A_j.*

Proof: We will prove that for any $i \neq j$, X_i is disjoint with A_j (which would immediately imply $A_j \subseteq B_i$). If $j < i$ then by construction X_i is disjoint with all A_j for $j \leq i$ and $A_j \subseteq B_i$. On the other hand, if $j > i$ then B_j contains A_i and suppose $A_j \cap X_i \neq \emptyset$ then $v \in A_j \cap X_i$ has at least one edge to A_i and thus to B_j, so A_j and B_j are not separated, a contradiction. □

Corollary 3.10 *Each B_i contains at least one detector.*

Lemma 3.11 *If a tight set $A \subset A_i$ is of size $\geq \frac{\epsilon n}{10}$ then the cut $N(A)$ is a shredder.*
Proof: Suppose not. Then $N(A)$ is a two-way $(\frac{\epsilon}{10}, \kappa)$-cut preserving B_i and hence $\bigcup_{j=0}^{i-1} A_j$. Thus $N(A)$ was not deleted from $\mathcal{F}$ until iteration i, so it should have been chosen instead of X_i, contradiction. □

In what follows we assume $\kappa \leq \frac{\epsilon^2}{20} n$. The next lemma shows how $\mathcal{D}_1$ (a detection set for shredders) helps to detect two-way min-cuts.

Lemma 3.12 *Let Y be an $\frac{\epsilon}{10}$-balanced two-way min-cut with sides C and D. Suppose D contains a set W of size at least $\frac{\epsilon n}{10}$ such that $N(W)$ is a shredder. Then $D + Y$ contains at least one detector from $\mathcal{D}_1$.*
Proof: The shredder $Z = N(W)$ is $\frac{\epsilon}{10}$-balanced, so it is weakly detected by $\mathcal{D}_1$. Since Y is a cut, there are no edges between W and C, i.e. Z lies in $D + Y$. It follows that C is connected in $G \setminus Z$, hence lies in a single connected component thereof. Thus at least one detector from $\mathcal{D}_1$ is not in C, so it is in $D + Y$. □

Now we are ready to sketch the proof of Theorem 3.8; the details are in the next subsection.

Proof Sketch of Theorem 3.8. Let Y be an ϵ-balanced two-way min-cut with sides C and D. We need to show that $\mathcal{D}$ meets Y or both sides thereof. For the sake of contradiction suppose it is not so. Then without loss of generality $\mathcal{D} \subset C$, which implies that C meets every A_i and B_i. Clearly then $A_i \not\subseteq D + Y$, for every i. Also note that by Lemma 3.12 D cannot contain disconnected tight sets larger than $\frac{\epsilon n}{10}$.

There are now three cases to consider, depending on the relation of Y to the sets X_i. First, suppose Y does not strongly cross any X_i. We show that $N(D \setminus \cup X_i)$ is a two-way $\frac{\epsilon}{10}$-balanced cut that was not excluded from $\mathcal{F}$ (see Figure 4a), and this contradicts the stopping condition of the algorithm. If Y strongly crosses exactly one X_i, then we replace Y by the cut $Y' = N(D \cap B_i)$ (see Figure 4b). Y' does not strongly cross any X_i, so we apply the argument from the case above to show that Y' is detected. Therefore there is at least one detector in set D, which contradicts our assumption. Finally if none of these two cases apply then Y strongly crosses at least two sets among $\{X_i\}$, say X_i and X_j. An argument using the Two-Quarters Lemma then shows that X_i and X_j partition Y into the same subsets (see Figure 4c). We then prove that X_i and X_j cut off a large connected subset D' of D such that $N(D')$ is a two-way $(\frac{\epsilon}{10}, \kappa)$-cut not deleted from $\mathcal{F}$, which thus violates the stopping condition. □

3.3 Full proof of Theorem 3.8

Lemma 3.13 *Suppose Y is ϵ-balanced and A_i meets D. Then either there is a detector in $D + Y$ or the following conditions hold:*

(a) *Y strongly crosses X_i, and*

(b) *$N(D \cap B_i)$ is a two-way $\frac{8\epsilon}{10}$-balanced min-cut.*

Proof: Suppose there is no detector in $D + Y$. Since A_i and B_i each contain a detector, they meet C. Now we can invoke the Two-Quarters Lemma to quarters $B_i \cap C$ and $A_i \cap D$ and conclude that $A_i \cap D$ is tight. We claim that $|B_i \cap D| \geq \frac{8\epsilon}{10} n$. Indeed, otherwise $|A_i \cap D| \geq \frac{\epsilon n}{10}$, so by Lemma 3.12 $N(A_i \cap D)$ is a two-way cut, which contradicts Lemma 3.11. Claim proved.

This proves (a) and shows that $N(B_i \cap D)$ is an $\frac{8\epsilon}{10}$-balanced cut. To complete (b), note that $B_i \cap D$ is tight by the Two-Quarters Lemma, so by Lemma 3.12 $N(B_i \cap D)$ is two-way. □

Let Y be an ϵ-balanced two-way min-cut with sides C and D. We need to show that $\mathcal{D}$ meets Y or both sides thereof. For the sake of contradiction suppose it is not so. Then without loss of generality $\mathcal{D} \subset C$, which implies that C meets every A_i and B_i. Clearly then $A_i \not\subseteq D + Y$, for every i. Also note that, by Lemma 3.12 D cannot contain disconnected tight sets larger than $\frac{\epsilon n}{10}$. There are three possible cases which we prove separately: (1) cut Y does not strongly cross any X_i, (2) cut Y strongly crosses exactly one X_i, and (3) cut Y strongly crosses at least two X_i's.

Case 1: cut Y does not strongly cross any X_i. To re-use this proof for the second case, we will assume that Y is only $\frac{8\epsilon}{10}$-balanced, rather than ϵ-balanced,

Since we assumed that X_i does not strongly cross Y by Lemma 3.13 we have all A_i's are disjoint with D. Using this fact we show that each X_i excises a small a piece of size at most κ from D, and finally we show that $D \setminus \cup X_i$ is large, tight, connected, and preserves $\cup A_i$, and thus algorithm could have made at least one more step.

Let $X_{i_1}, X_{i_2}, \ldots X_{i_t}$ be all cuts which are intersecting with D. Let $D_j = D - D \cap \bigcup_{h=1}^{j} X_{i_h}$, $Y_j = N(D_j)$ and $C_j = V - Y_j - D_j$. First of all

$$|D_j| \geq |D| - \sum_{h=1}^{j} |X_{i_h}| \geq \frac{8\epsilon}{10} n - \kappa \frac{10}{\epsilon} \geq \frac{3\epsilon}{10} n$$

The last transition is because $\kappa \leq \frac{\epsilon^2}{20}n$.

We will prove by induction that each D_j is tight, connected and corresponding cut $Y_j = N(D_j)$ is two-way for every $0 \leq j \leq t$.

Suppose we did that, then D_t by its construction is disjoint with any X_i, and thus all A_i's are disjoint with Y_t, and hence lie in $V - D_t - Y_t$, therefore Y_t preserves $\bigcup A_i$ (because Y_t is a two-way cut). On the other hand $|D_t| \geq \frac{2\epsilon}{10}\, n$ and $|C_t| \geq |C| \geq \epsilon N$. So Y_t is $\frac{2\epsilon}{10}$-balanced two-way min-cut and preserves $\bigcup A_i$, thus our algorithm could have made one more step, and so we come to contradiction.

Now we have to prove our claim. Clearly D_0 is tight, connected and $N(D_0) = Y$ is two-way by our definition of Y and D. Suppose the claim holds for D_{j-1}, we now prove it for D_j. We have

$$D_j = D_{j-1} - D_{j-1} \cap X_{i_j} = B_{i_j} \cap D_{j-1}.$$

If D_j is disjoint with X_{i_j} then $D_j = D_{j-1}$ and we are immediately done. Otherwise, Y_{j-1} weakly crosses X_{i_j}. (Indeed, D_{j-1} is not preserved by X_{i_j}, and $C_{j-1} \supseteq C$ and hence meets both A_{i_j} and B_{i_j} and so not preserved.) But then we satisfy conditions of the Two-Quarters Lemma, where $A_{i_j} \cap C_{j-1}$ and $B_{i_j} \cap D_{j-1}$ is not empty, and thus $D_j = B_{i_j} \cap D_{j-1}$ is tight. Therefore by Lemma 3.12 D_j is connected and $N(D_j)$ is a two-way cut. This proves the claim.

Case 2: cut Y strongly crosses exactly one X_i. Indeed, consider set $D' = D \cap B_i$. By Lemma 3.13 and our assumption that there were no detectors in $D + Y$, it has size at least $\frac{8\epsilon}{10}\, n$, is tight and corresponding cut $Y' = D \cap X_i + X_i \cap Y + Y \cap B_i$ is two-way min-cut.

Since $D' \subseteq D$, and only one A_i meets D (and it does not meet with D' by our construction), no A_i meets with D'. Therefore by Lemma 3.13 Y' does not strongly cross any X_i and thus by the case (1) Y' is detected by $\mathcal{D}$. This proves that there is at least one detector in $Y' + D'$, and by construction $Y' + D' \subseteq D + Y$, and therefore there is at least one detector in $D + Y$, contradiction.

Case 3: cut Y strongly crosses at least two X_i's. We need to prove that either $D + Y$ contains at least one detector from $\mathcal{D}$ (and thus contradicting our assumption), or we could have done one more step of the algorithm $A2$. Without loss of generality Y strongly crosses X_1 and X_2 (see Figure 4c).

First we prove that each of the triples (A_1, X_1, B_1) and (A_2, X_2, B_2) partitions set Y into the same subsets.

Claim 3.14 *$X_1 \cap Y = X_2 \cap Y$ and $A_1 \cap Y = B_2 \cap Y$ and $B_1 \cap Y = A_2 \cap Y$.*

Proof: Note that $Y = Y \cap A_i + Y \cap X_i + Y \cap B_i$, and since $A_1 \subseteq B_2$, we have that $Y \cap A_1 \subseteq Y \cap B_2$, and analogously $Y \cap A_2 \subseteq Y \cap B_1$, but by the Two-Quarters Lemma we have $|Y \cap A_1| = |Y \cap B_1|$ and $|Y \cap A_2| = |Y \cap B_2|$ and thus $Y \cap A_1 = Y \cap B_2$ and $Y \cap A_2 = Y \cap B_1$, and thus $X_1 \cap Y = X_2 \cap Y$. □

We will prove that either there is a leftover part D' in D, which could have used for the next step of the algorithm, or Y is detected.

For each $i = 1, 2$, since X_i strongly crosses Y, set D is partitioned by X_i into three non empty parts $D'_i = D \cap B_i$, $D''_i = D \cap A_i$ and $D'''_i = D \cap X_i$. Now, by Lemma 3.13 and our assumption that $\mathcal{D} \cap (D + Y) = \emptyset$, we conclude that D'_i is tight, its cardinality is at least $\frac{8\epsilon}{10}\, n$ and $N(D'_i)$ is two-way min-cut.

Consider $D' = D'_1 \cap D'_2$. We claim that the corresponding cut $Z = N(D')$ is a two-way $(\frac{\epsilon}{10}, \kappa)$-cut that preserves $\cup A_i$. This contradicts the stopping condition of the algorithm: it could have made one more iteration. Therefore it remains to prove the claim.

Firstly, D' is tight by the Two-Quarters Lemma applied to cuts $N(D'_1)$ and $N(D'_2)$. Its size is

$$|D'| = |D'_1 \cap D'_2| = |D - (D''_1 + D'''_1) \cup (D''_2 + D'''_2)| \geq \epsilon n - 2(\frac{\epsilon n}{10} + \kappa) \geq \frac{6\epsilon}{10}\, n,$$

so Z is $\frac{\epsilon}{10}$-balanced, and moreover D' is connected (this is by Lemma 3.12 and the assumption that $\mathcal{D}$ is disjoint with $Y+D$). Since $Z = (X_1 \cup X_2) \cap (D \cup Y)$, we conclude that (1) Z is two-way, since $V - D' - Z$ is connected as a union of three non-disjoint connected subsets C, A_1 and A_2, and (2) Z is disjoint with $\cup A_i$ by Lemma 3.9.

To prove that Z preserves $\cup A_i$ it remains to show that all A_i's are disjoint with D'. Indeed, suppose some A_i meets D'. It cannot be properly contained in D, hence in D'. So, since A_i is connected, it meets Z, contradiction. Claim proved. This completes the proof of Theorem 3.8.

3.4 Strong detection sets

We present a partial result on extending strong detection sets for ϵ-shredders to those for general (ϵ, κ)-cuts. Essentially, we show that it suffices to have a strong (ϵ, κ)-detection set D' for some subgraph $G' = (V, E')$ of G of the same connectivity κ. In particular, we can without loss of generality assume that G is *minimally* k-connected.

Theorem 3.15 *Suppose $\kappa < \epsilon n$ and we have a strong (ϵ, κ)-detection set D' for a κ-connected subgraph $G' = (V, E')$ of G. Then we can use D' to construct a strong (ϵ, κ)-detection set for G. Specifically, for a high-probability result it suffices to add $O(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$ randomly sampled detectors. Alternatively, it suffices to add at most $\frac{2}{\epsilon}$ detectors, and there is a polynomial-time algorithm to construct them.*

Proof: Let D'' be a smallest detection set for ϵ-shredders of G. Let S be an (ϵ, κ)-cut G. Then S is an (ϵ, κ)-cut in G' such that each S-component in G is a union of S-components in G'. Obviously, if S-components are the same in G and in G', then D' detects S. Therefore, if $D' \cup D''$ does not detect S, then S is a two-way (ϵ, κ)-cut in G but a shredder in G'. Call such cuts *evil*. Therefore it suffices to detect all evil cuts.

For an evil cut S, the two components of S in G are called S*-shores*. We need to put a detector in each S-shore. For the rest of the proof we can forget about G. We operate (only) on G' and treat S-shores as unions of components of S in G'. The proof is similar to that of Theorem 3.1.

Evil cuts are ϵ-shredders in G', so there are at most n of them and they can be efficiently listed. Let $\mathcal{F}_0$ be the family of *all* evil cuts. Start with $\mathcal{F} = \mathcal{F}_0$. While there exists $S \in \mathcal{F}$ such that each S-shore contains an $\mathcal{F}$-head of S, delete S from $\mathcal{F}$ (because by Lemma 3.5 S is detected by D'). Let $\mathcal{F}_1$ be the resulting family of evil cuts. Clearly if D is a detection set for $\mathcal{F}_1$ then $D \cup D'$ is a detection set for $\mathcal{F}_0$.

Say $H \subset V$ is a *head* of S if H is an $\mathcal{F}_1$-head of S. Let the *tail shore* of $S \in \mathcal{F}_1$ be the S-shore that does not contain any heads of S (such shore exists by construction of $\mathcal{F}_1$). Observe that for any two $S, T \in \mathcal{F}_1$ the tail shore of T is contained in a head of S (and vice versa). Why? T meets exactly one component of S, say H (so H is a head). By Lemma 3.4 H contains all T-components but one, call it C. C meets S, thus C is a head. Therefore, the tail shore of T is contained in H.

By the observation above, the tail shores of cuts in $\mathcal{F}_1$ are pairwise disjoint and moreover (assuming $\mathcal{F}_1$ consists of at least two cuts) putting a detector in each of these shores strongly detects $\mathcal{F}_1$. Since the tail shores have size $\geq \epsilon n$ each, $|\mathcal{F}_1| \leq \frac{1}{\epsilon}$. Therefore we need $\frac{1}{\epsilon}$ detectors for F_0, which together with D'' is $\leq \frac{2}{\epsilon}$ detectors. For a random sampling result note that it suffices to augment D' by a hitting set for the tail shores of $\mathcal{F}_1$ and the tails of ϵ-shredders of G, as defined in the proof of Theorem 3.1. $\square$

4 Extensions and further directions

There are a number of natural questions left open by this work. One is to investigate whether an (ϵ, κ)-detection set for node failures of minimum size can be computed in polynomial time for a given graph G; this would parallel the per-instance result we obtain for edge failures. We note that Section 3.1 provides such an optimality result for node failures when the adversary is restricted to deleting a shredder.

We believe it would be interesting to extend our results on node failures to obtain bounds for strong detection sets. In fact, our bounds for shredders apply already to the case of strong detection; and in Theorem 3.15 we provide a further step in this direction, proving that we can without loss of generality assume that G is *minimally* k-connected.

It would clearly be interesting to obtain results on detection sets with respect to adversaries that can delete a number of nodes equal to a constant times the node-connectivity, by analogy with our results for edge-connectivity. To obtain detection set bounds here that are independent of the value of κ, it is not difficult to see that we need to focus on weak detection; indeed, there exist graphs in which we would need at least $k - \kappa$ nodes in any strong (ϵ, k)-detection set for node failures.

Finally, the problem of deciding whether a given set D is an (ϵ, k)-detection set provides another clear connection to the problem of balanced separators in graphs: indeed, deciding whether the empty set is an (ϵ, k)-detection set is coNP-complete because of its equivalence to a balanced separator problem. On the other hand, using techniques from [11, 22], we can obtain a polynomial-time algorithm for deciding whether D is an (ϵ, k)-detection set for node failures when $k = \kappa$; this is non-trivial due to the fact that there can be exponentially many min-node-cuts.

Acknowledgments. It is our pleasure to acknowledge the contribution of Laszlo Lovász; discussions with him about the prospect of parameterizing detection sets by the minimum cut size provided a portion of the motivation for this work, and also led to the results described in Section 2.3.

References

[1] A. Blumer, A. Ehrenfeucht, D. Haussler and M. Warmuth, "Learnability and the Vapnik-Chervonenkis Dimension," *J. of the ACM*, **36**(4) (1989), pp. 929-965.

[2] J. Cheriyan and R. Thurimella, "Fast Algorithms for k-shredders and k-node Connectivity Augmentation," *Proc. 28th Annual ACM Symposium on the Theory of Computing* , 1996.

[3] B.Cherkasskij, "Solution of a Problem on Multicommodity Flows in a Network," *Ekon. Mat. Metody* 13:1(1977), pp. 143-151. (in Russian; the result is cited in [10]).

[4] F. Chung, M. Garrett, R. Graham and D. Shallcross, "Distance realization problems with applications to Internet tomography," *Journal of Computer and System Sciences* 63(3): 432-448 (2001).

[5] K. Claffy, T. Monk and D. McRobb, "Internet Tomography," *Nature, Web Matters*, Jan. 7, 1999.

[6] E.Dinits, A.Karzanov and M.Lomonosov, "On the Structure of a Family of Minimal Weighted Cuts in a Graph", in A.Fridman, ed., *Studies in Discrete Optimization*, pp. 290-306, Nauka, Moscow, 1976 (in Russian).

[7] J.Fakcharoenphol, *An improved VC-dimension bound for finding network failures* . Master's thesis, Dept. of EECS, UC Berkeley, 2001.

[8] U. Feige and M. Mahdian, "Finding small balanced separators", *Proc. 38th ACM Symposium on Theory of Computing*, 2006.

[9] L. Fleischer, "Building Chain and Cactus Representation of all Minimum Cuts from Hao-Orlin in the Same Asymptotic Run Time". *J. Algorithms*, **33**(1) (1999), pp. 51-72.

[10] A. Frank, “Packing paths, circuits, and cuts – a survey,” in: *Paths, Flows and VLSI-Layouts*, B. Korte, L. Lovász, H-J. Prömel, A. Schrijver, eds., Springer-Verlag (1990) pp. 47-100.

[11] H. Gabow, “Centroids, Representations, and Submodular Flows,” *J. Algorithms* **18**(1995) pp. 586-628.

[12] H. Gabow, “Using expander graphs to find vertex connectivity,” *Proc. 41st Annual IEEE Symposium on Foundations of Computer Science*, 2000.

[13] A. Gupta, November 2000, personal communication. See Section 1.4.2 of [7] for a write-up.

[14] T. Jordán, “On the optimal vertex-connectivity augmentation,” *J. Combin. Theory*, Ser. B **63**(1) (1995), pp. 8-20.

[15] T. Jordán, “On the Number of Shredders,” *J. Graph Theory*, **31** (1999), pp. 195-200.

[16] J. Kleinberg, “Detecting a Network Failure,” *Proc. 41st Annual IEEE Symposium on Foundations of Computer Science*, 2000.

[17] L. Lovász, “On Some Connectivity Properties of Eulerian Graphs,” *Acta Math. Hung.* 28 (1976), pp. 129-138.

[18] W. Mader, “Über die Maximalzahl kantendisjunkter A-Wege,” *Arch. Math.* **30**(1978), pp. 325-336.

[19] J. Mahdavi and V. Paxson, “IPPM Metrics for Measuring Connectivity,” *RFC 2678, Network Working Group*, Sept. 1999.

[20] S. Muthukrishnan, T. Suel and R. Vingralek, “Inferring Tree Topologies Using Flow Tests,” *Proc. 14th ACM-SIAM Symposium on Discrete Algorithms*, 2003.

[21] V. Paxson, “Towards a Framework for Defining Internet Performance Metrics,” *Proc. INET* 1996.

[22] J.-C. Picard and M.Queyranne, “On the Structure of All Minimum Cuts in a Network and Applications,” *Mathematical Programming Study* **13**(1980) pp.8-16.