Data Recovery After Geographic Correlated Attacks

Guy Grebla Alon Efrat Esther Ezra Rom Pinchasi Swaminathan Sankararaman

***Abstract*—In distributed storage networks, ensuring data availability in the presence of hardware faults is an important requirement. Typically, redundancy schemes such as replication and erasure coding are used to ensure this. In case of hardware failures, these networks may be disconnected into multiple components, each of which may require access to the data. In addition, the placement of redundant information must also be optimized as it is ever-changing and requires constant updating.**

We study the problem of selecting a set of nodes in networks of this kind so that data availability is maintained in the face of *geographically correlated failures*. We model failure events of arbitrary shapes as the union of disks or line segments in the plane and present approximation algorithms for the problem of selecting a minimum number of redundant information locations (such as replicas or coded file segments) so that data recovery is guaranteed at every node in the face of *any* failure event. Using tools from computational geometry, our algorithms are efficient and provide good guarantees.

I. Introduction

Distributed data storage is fast becoming the go-to solution for providing reliable and fast access to data in the cloud (see, e.g., [8],and [10]). A major issue to consider when designing protocols for such distributed file storage is ensuring reliability and maintaining availability of data in the presence of hardware failures. Of particular importance are failures in the backbone network employed for communication among the data storage "nodes". Such failures may result in network disconnection and thus, loss of access to large portions of the network. For example, in content delivery networks, communication takes place over the internet and data access must be ensured even when failures occur in the internet backbone. At a lower level (e.g. within data-centers), failures in the local backbone must not result in loss of data access. Ensuring reliability in the event of such failures necessitates the presence of an effective data redundancy scheme.

In current systems, there exist two main redundancy schemes for ensuring these requirements: (i) replication of data, where multiple copies of the data are stored (preferably in diverse locations), and (ii) erasure coding, where a set of data fragments is generated (whose total size is larger than the actual data size) and distributed across the network so that we may recover the original data even if a few of these fragments are lost or "erased". These redundancy schemes are usually deployed in a fairly ad-hoc manner by trying to distribute data as widely as possible in the network. However, network failure scenarios often *correlated* as they are often caused by external events such as natural disasters, human error or malicious attacks. Therefore, it is important to study methods of exploiting this correlation in a principled manner to ensure reliability and availability of data. Further, data is often being modified/updated very frequently and maintaining consistency of data is an important requirement. Reconfiguration of the replicas may also be necessary due to efficiency constraints. Thus, it is imperative to reduce communication by minimizing the number of replicas/segments stored in the network while not compromising on reliability and availability.

/978-1-4799-7795-6/15/$31.00 ©2015 IEEE

Geographically Correlated Failures. In this work, we are concerned with *geographically correlated failures*, i.e., where a set of network components fail due to some external events whose effects are geographically correlated. Geographically correlated failures have recently gained attention in the study of network survivability (see [1], [2], [4], [5], [23]–[25], [28]). These types of failures may arise at a higher scale due to the occurrence of natural disasters or even at a lower scale (e.g., within a data-center) due to events such as fires. Fig. 1 demonstrates the effects of a geographically correlated failure on the giber backbone of a major network provider [20]. We model the shapes of such failure regions by considering them as the union of a set of disks/lines in the plane. This models a variety of natural disasters such as earthquakes, hurricanes and solar flares or man-made ones such as Electromagnetic Pulse (EMP) attacks, fires or dragging anchors.

Network, Failure and Redundancy Models. We model the backbone network as a connected graph $G = (V, E)$ embedded in the plane where V is a set of n node locations (corresponding to routers/switches) and E is a set of m links connecting the nodes (modeled as

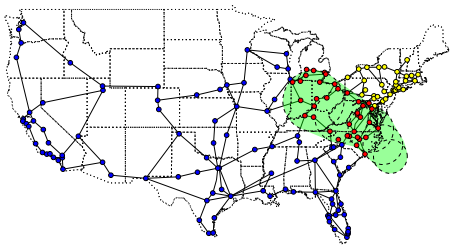


Fig. 1. Impact of a disaster on the fiber backbone of a major service provider [20]. Links in the region shaded green assumed to fail. Red nodes are effected directly, but are disconnected from the blue nodes.

linear or piece-wise linear approximations of the curves in the plane corresponding to the cables running between the routers). In many networks, such as fiber-optic backbones of Internet Service Providers (ISPs) or even smaller-scale backbone networks such as those in data-centers, the graph is "almost" planar (see Fig. 1), i.e., only few cables cross each other, and thus we assume that G is a planar graph. Individual data servers are connected to the routers and when we refer to storing/placing data at a node, it implies that the data is actually stored at one of these servers.

A failure event destroys nodes and links from G confined to the shape of the failure events. As a result, the ***residual network*** (surviving network after the failure event) may consist of several maximal connected pieces or components. Even though communication via different routes may be possible, these are likely to be congested, hence, the network is effectively disconnected. Formally, we define the failure model as a collection Γ of possible failure regions dictated by a specific underlying shape γ. In this paper, we consider the failure models where γ could be disks or line segments. For the case of disks, we consider fixed-radius or unit disks, i.e., all possible translations of a disk γ as well as arbitrary radius disks, i.e., all possible translations, rotations and scalings of γ. For line segments, we consider all possible endpoint of arbitrarily-lengeth segment.

Redundancy may be achieved in one of two ways: ***replication*** and ***erasure coding*** [12]. Replication involves placing copies of a data file in multiple nodes such that given a disaster, every surviving node has access to at least one copy. Erasure coding aims to reduce the storage requirement of replication by computing some encoding of the data file and storing small portions of it in different locations so that a node may reconstruct the entire file by retrieving a few of these encoded portions. Formally, a (k', k)-*erasure code* (where $k' \geq k$) splits the data file f into k segments, encodes these k segments in some manner to obtain a set of k' segments to be stored in the network. The original file can be reconstructed in any node in the network by retrieving and decoding any k of k' segments. Some types of erasure codes such as Fountain codes are "rateless", i.e., the number k' need not be fixed and we can generate as many coded segments as needed.

In what follows, we assume in the replica model that each node can store *at most one replica*, whereas in the erasure-code model we do not have a restriction on the number of segments to be stored at a node.

Our results. Given a network G modeled as above, a redundancy scheme to be used and a file f, we consider the following file distribution problem: Compute a set of nodes V_D so that, if f is stored at nodes in V_D using a redundancy scheme (replication or erasure-coding), in the event of *any* geographically correlated failure scenario, f may be recovered at all surviving nodes. That is, if the failure event disconnects the network into several connected components, each component must contain the entirety of the file in itself. We also consider the case where only those nodes that lie in sufficiently large components will need the file. In this case, we show a considerably better bound. We denote by k^* the size of the optimum solution of V_D and as defined earlier, n is the number of nodes in the network.

Theorem I.1 (Replication I)**.** *Given a file f, a network model G and a failure model Γ of disks/lines, one can compute in expected time $O(k^*\lambda(n)\log^2 n)$ a subset of nodes V_D of size $O(k^* \log k^*)$ to store replicas of f such that, given a failure event, every node in the residual network has a path to a node in V_D and can retrieve f. Here, $\lambda(n) = O(n^2)$ for unit/fixed radius disks or line segments and $\lambda(n) = O(n^3)$ for arbitrary radius disks.*

We note that k^* is likely to be much smaller than n, and thus our algorithms "beat" the standard greedy algorithm, which has only $O(\log n)$ guarantee. As a special case of the problem above, where we are only interested in recovering the large connected components in the residual graph, we obtain:

Theorem I.2 (Replication II)**.** *Given a network model G and a failure model Γ of disks/lines, one can compute in expected time $O(\lambda(n)\log^2 n)$ a subset of nodes V_D of size $O((1/\varepsilon)\log(1/\varepsilon))$ to store replicas of a file such that in the event of a failure at* any *location, every node in the residual network which has paths to atleast εn other nodes has a path to a node in V_D with high probability. Here, $\lambda(n) = O(n^2)$ for unit/fixed radius disks or lines and $\lambda(n) = O(n^3)$ for arbitrary radius disks.*

We also show that our scheme are applicable for more general disaster shapes as well. If the shape can be

approximated as the union of b disks and/or lines, for some integer parameter $b > 0$, we get the following.

Theorem I.3 (Replication III)**.** *Given a file f, network model G and a failure model Γ of aribitrary shapes approximated as the union of b disks and/or line segments, one can compute in expected time $n^{O(b)}$ a subset of nodes V_D of size $O(bk^* \log k^*)$ to store replicas of f such that, given any failure event, every node in the residual network has a path to at least one node in V_D.*

Finally, when applying an erasure code (but not restricting node storage capacity), we show:

Theorem I.4 (Erasure-coding)**.** *Given a file f, network model G and failure model Γ of disks/lines, and suppose a (k', k)-erasure code is used, one can compute in polynomial time, a file distribution scheme for $O(k^* \log k^*)$ file segments through the network such that, given any failure event, every node containing a path to at least k nodes in the residual graph has access to at least k encoded segments and can, thus reconstructed the file.*

Theorem I.4 implies that employing a (k', k)-coding scheme or a rateless coding scheme to generate k' encoded segments would guarantee recovery of f at every node with sufficient connectivity.

Related Work. Data replication is used in many large-scale distributed storage systems such as Google's BigTable [8] and Amazon's Dynamo [10]. The advantage of replication schemes is that they are simple and easier to implement than coding schemes while the disadvantages lie in the storage requirements. Error-correcting codes have been utilized in distributed storage systems such as RAID [27] and DPSS [21] to ensure data availability while being efficient. The question of how to recover data at a node has been studied in the context of bandwidth requirements [11], [12]. These works consider the tradeoff between bandwidth and storage and focus on data recovery at a single node. Works that consider network topology to design redundancy schemes [18], [22] focus on efficiency of data recovery when data loss occurs while the network is intact. In this, our work differs from these by considering the causes of such data loss directly to decide where to store redundant information. The key drawback of all of these is that they do not explicitly consider the geographic correlation of failures into account.

Whereas the focus of some of the previous works in geographic correlated failures is on (i) assessing the vulnerability of (geographical) networks to such disasters [1], [24], [25] (e.g., identifying disaster locations that have the maximum effect on network capacity), and the impact of such failures [24], (ii) analyzing network vulnerability or evaluate its robustness [2], [4], and measuring its reliability [23], we propose a scheme to recover such networks and achieve data reliability in the face of *any* geographic correlated disaster.To the best of our knowledge, the only previous study which considers the problem of designing data distribution schemes in order to achieve tolerance to geographically correlated failures is the recent work of Banerjee *et al.* [5]. Their work devises a scheme to distribute a set of coded segments generated from the data using an erasure coding scheme into the network, where each node in the network has a storage capacity for storing the file segments. Recovery of the data is only guaranteed in the largest surviving component of the network. Our work is different in two ways: (i) first, we guarantee a recovery in *every* connected component of the residual network rather just the largest surviving component. , and (ii) using erasure codes, although we assume that the storage capacity is not limited, we are able to use techniques from geometric optimization and learning theory reduce the approximation factor to $O(\log k^*)$ in polynomial time, where k^* is the size of the optimal solution, and the overall majority of the running time is derived by producing a solution to a linear programming system on n variables. The latter is a significant improvement over the $O(\log n)$-approximation of their algorithm since in realistic settings, we expect k^* to be much smaller than n. Thus we obtain better guaranteed performance bounds than those in [5] (although the model is slightly different) and in many cases, suggest faster algorithms.

II. Preliminaries

In this section we describe the various steps of our algorithm, so that given a network $G = (V, E)$, modelled as a planar graph, and a geographic correlated failure modeled as a family Γ of translations and rotations of a planar region γ, compute a smallest set of nodes $V_D \subseteq V$, so that each connected component in the residual network G_γ (as defined in Section I), for any $\gamma \in \Gamma$, contains a node of V_D. We consider cases where γ is either a disk or a straight line.

Range spaces. We first review a few concepts from random sampling and statistical learning theory [26]. Let X be a (possibly infinite) set of objects also referred to as the "space", and let $\mathcal{R}$ be a (possibly infinite) family of subsets of X, those subsets are called the "ranges". The pair $\Sigma = (X, \mathcal{R})$ is called a *range space*.

In our context, we define the range space $\mathcal{H} = (V, \mathcal{C}_\Gamma)$, where V is the set of nodes of G as above, and $\mathcal{C}_\Gamma$ consists of the collection of all subsets of V that appear in a connected component of the residual graph $\mathcal{G}_\gamma$ (that is, after excluding from G all nodes and links meeting

γ), for each $\gamma \in \Gamma$. Although at first glance it may appear that $\mathcal{C}_\Gamma$ is a very large collection (e.g., it may contain exponentially many connected components), a main property shown in Theorem III.1 is the fact that $|\mathcal{C}_\Gamma|$ is only *polynomial* in $|V|$ (and this property holds for any restriction of $\mathcal{C}_\Gamma$ to a subset of V)—see below.

Hitting sets and ε-nets. A *hitting set* for a range space $(X, \mathcal{R})$ is a subset $H \subseteq X$ such that each range $R \in \mathcal{R}$ intersects H. The *hitting-set problem* is to find a smallest-size hitting-set. A related (dual) problem is the *set-cover problem* is to find a smallest subcollection of ranges $\mathcal{S} \subseteq \mathcal{R}$ that altogether cover X. The general (primal and dual) problems are NP-Hard to solve (even approximately) [14], and the simple greedy algorithm yields the (asymptotically) best known approximation factor of $O(\log n)$ computable by a polynomial-time algorithm.Most of these problems remain NP-Hard even in geometric settings.However, one can attain an improved approximation factor of $O(\log k^*)$ in polynomial time for many scenarios, where k^* is the size of the optimal solution. This improvement is based on the iterative reweighting scheme of Brönnimann and Goodrich [6], which is strongly related to the theory of ε-nets.

An ε-net for a range space $X, \mathcal{R})$ (for some $\varepsilon > 0$) is a subset $N \subset X$ such that each range $R \in \mathcal{R}$ intersects N if $\|R \cap X| \geq \varepsilon \|X\|$, i.e., an ε-net is a hitting set for all "heavy" ranges. A seminal result by Haussler and Welzl [17] shows that under some favorable scenarios (including various geometric settings), a random subset $N \subseteq X$ of size $O(1/\varepsilon \log(1/\varepsilon))$ is an ε-net with constant probability. The notion of ε-nets can be extended to the case where the elements in X are assigned weights. In this case a *weighted ε-net* is a subset $N \subseteq X$ that hits all ranges $R \in \mathcal{R}$ whose total weight (that is, the sum of the weights over the elements $x \in R$) is at least ε of the total weight of X.

Iterative reweighting scheme. We are given a range space $(X, \mathcal{R})$, and the goal is to find a small subset H of X which meets every range in $\mathcal{R}$. The technique assumes the availability of two black-box routines: (i) an *ε-net finder*, which is a procedure that, given any weight function w on X and $\varepsilon > 0$, constructs a *weighted ε-net* N for $(X, \mathcal{R})$, in the sense that N hits each range whose weight is at least $\varepsilon w(X)$; (ii) a *verifier*, which is a procedure that, given a subset $H \subseteq X$, either determines that H is a hitting set, or returns a nonempty range $R \in \mathcal{R}$ such that $R \cap H = \emptyset$. The algorithm runs an exponential search to guess the value of k^*, which is the size of the smallest hitting set for $(X, \mathcal{R})$. At each step, denote by k^* the current guess for this value. The algorithm assigns weights (initially, uniform) to the elements of X, and uses the net finder to select a (weighted) $\frac{1}{2k^*}$-net N. If the verifier determines that N is a hitting set, it outputs N and stops. Otherwise, it returns some range R not hit by N. We double the weights of the elements in R, and repeat the above procedure until N hits all the ranges in $\mathcal{R}$, or abort after a pre-specified number of iterations, concluding then (with high probability) that the current guess for k^* is too small. Thus, upon termination, the size of the reported hitting set has the same upper bound as that for $\frac{1}{2k^*}$-nets. If we are in the scenario where this bound is $O(k^* \log k^*)$ (as mentioned above), then the resulting approximation factor is $O(\log k^*)$.

The analysis of [6] shows that, at the right guess for k^*, the algorithm terminates after at most $O(k^* \log (n/k^*))$ rounds. Thus the overall running time of the algorithm is $O\left(k^* \log (n/k^*)(T_N + T_V)\right)$, where T_N, T_V are the respective running time bounds for the net-finder and the verifier.

Our problem. Our problem of computing a smallest set of nodes $V_D \subseteq V$, so that each connected component in the residual network contains a node of V_D is nothing but a hitting-set problem. Indeed, recall that our range space $\mathcal{H} = (V, \mathcal{C}_\Gamma)$ consists of the nodes V of G and the collection of all subsets of V that appear in a connected component of the residual graph. Thus our actual goal is to compute a smallest subset of $V_D \subseteq V$ that hits each range in $\mathcal{C}_\Gamma$ (and thus each connected component in the residual graph).

We note that connected components caused by (infinite-length) lines are always contained in connected components caused by line-segments. Hence it is enough to consider the case where events are either disks or infinite lines.

III. Algorithm for Replica Placement

A broad overview of the algorithm is as follows. We first describe the specification of an ε-net and a verifier in the context of our problem, and then describe the algorithms implementing these specifications. We show below that one can (i) construct in expected polynomial time an ε-net of size $O(1/\varepsilon \log (1/\varepsilon))$, and (ii) implement the verifier procedure in polynomial time. We plug these into the iterative reweighting scheme from Section II to get an expected polynomial-time $O(\log k^*)$-approximation algorithm for our hitting-set problem.

A. An ε-net Finder

In our setting, an ε-net for the range space $(V, \mathcal{C}_\Gamma)$ is a subset $N \subseteq V$ that hits each connected component of size at least εn in the residual graph G_γ, for any $\gamma \in \Gamma$.

The main challenge that we face in this part of the analysis is *combinatorial*, that is, we show that our range space $(V, \mathcal{C}_\Gamma)$ admits an ε-net of size $O(1/\varepsilon \log(1/\varepsilon))$.

VC-dimension. Given a range space $\Sigma = (X, \mathcal{R})$, a finite subset $Y \subseteq X$ is *shattered* by Σ if $\{R \cap Y \mid R \in \mathcal{R}\} = 2^Y$, i.e., the restriction of $\mathcal{R}$ to Y can realize all subsets of Y. The *VC-dimension* of Σ is defined to be the largest size of a subset of X shattered by Σ; the VC-dimension is infinite if this is arbitrarily large. Many natural range spaces have finite VC-dimension. The bound on the size of ε-nets (see Section II) is in fact $O((d/\varepsilon)\log(d/\varepsilon))$ [17], i.e., a random sample of that size is an ε-net with constant probability. This was later improved to $O(\frac{d}{\varepsilon}\log\frac{1}{\varepsilon})$ in [19].

Our setting. In order to show that our setting admits an ε-net of size $O(1/\varepsilon \log(1/\varepsilon))$, we show that the VC-dimension d of $(V, \mathcal{C}_\Gamma)$ is finite. Then, we construct such an ε-net by randomly sampling $O(d/\varepsilon \log(1/\varepsilon))$ nodes of V and obtain an appropriate sample after a constant number of trials on average. The construction of a weighted ε-net is somewhat more involved but can be done in $O(n)$ time using standard machinery[1].Thus the actual ε-net construction is standard, while the bound on its size requires a considerably more involved analysis. We devote the remainder of the analysis concerning the net-finder to show that $(V, \mathcal{C}_\Gamma)$ has a finite VC-dimension, which is proved in Theorem III.1.

A major difficulty in our setting $(V, \mathcal{C}_\Gamma)$ is the fact that G does not have a constant description complexity, i.e., a connected component in $\mathcal{C}_\Gamma$ is not necessarily defined by a constant number of nodes (we no longer have the property that a range is defined by a constant number of elements from the space). Moreover, by definition of restricted range spaces, in the restriction of $(V, \mathcal{C}_\Gamma)$ to a subset $V' \subseteq V$, we do not modify the paths connecting nodes in the original graph G, but only consider the connectivity among the nodes in V' (w.r.t. the original graph). Thus each connected component $C \in \mathcal{C}_\Gamma$ is now confined to V'. Given this setting, the problem of showing that the number of such restricted ranges is only polynomial in $|V'|$ becomes non-trivial. Due to lack of space, we only provide a sketch of our proof, and defer the rest of the rather intricate analysis to the full version [13, cf. Appendix].

Theorem III.1. *The range space $(V, \mathcal{C}_\Gamma)$ has VC-dimension 4, when Γ is either a set of disks or lines in the plane.*

[1]Roughly speaking, each node of V is "duplicated" according to its weight, and then, after an appropriate weight normalization, a random sample of $O(1/\varepsilon \log(1/\varepsilon))$ elements from this new setting is a weighted ε-net with constant probability

Proof sketch: We only present the case where Γ consists of disks. The case of lines is an easy generalization. We exploit the fact that for any pair of disks their boundaries are either disjoint or meet exactly twice; these regions are referred to as *pseudo-disks*.

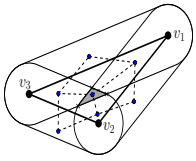
Recall that G is planar and connected. We show that one cannot shatter a subset of nodes of size 5, i.e., given any subset $U \subseteq V$ of five nodes, one cannot obtain all 2^U subsets appearing as connected components in the residual graph G_γ, for any $\gamma \in \Gamma$. In this case, we assume by contradiction that this shattering is possible, implying that all pairs of nodes $\{u, u'\}$ in U are realizable. Thus, there is always a region $\gamma \in \Gamma$ such that (i) it leaves u, u' and the path (in G) connecting them intact, and (ii) intersects all other paths connecting the remaining pairs in U (and so now these pairs are disconnected). Consider now all these pairs $\{u, u'\}$ and the paths $P_{u,u'}$ connecting them, they form a "clique graph" of five nodes (with edges replaced by these paths). It is well known from combinatorial geometry that this clique cannot be embedded in the plane so that all its edges are crossing-free [15]. Combined with a theorem of Hannani and Tutte [15], [29], there must be at least one such pair of paths $P_{u,u'}$, $P_{v,v'}$ that meet an odd number of times (as this theorem implies that when each pair of paths cross an even number of times, the corresponding graph can be embedded crossing-free in the plane). Since G is planar and Γ is a set of pseudodisks, we obtain that $P_{u,u'}$, $P_{v,v'}$ meet an even number of times, which is a contradiction. □

B. The Verifier

Our verification process proceeds as follows. Given a planar graph $G = (V, E)$ modeling our network, a set Γ of geographic correlated failures modeled as a collection of planar regions (each of which is a translation or rotation of a fixed planar region γ), and a set $V_D \subset V$ of replicas, determines whether V_D hits each connected component in the residual graph G_γ, for any $\gamma \in \Gamma$. In case of a missed component, we return this component. The procedure is based on the algorithm of Agarwal *et al.* [2]. We thus only briefly describe our approach, then apply the algorithm in [2], and then state its running time when Γ is either a set of disks or lines in the plane.

In each of these cases, Γ consists of a continuous collection of regions (each of which is a translation and rotation of a fixed region γ). It is, however, sufficient to consider a set of special events, referred to as *critical events* in which the structure of the residual graph G_γ changes, and new connected components may be created (as a split of existing components), or existing components may be merged together. Only in the events of the former kind the (given) set V_D may miss a connected

component (just created). The "merging" events must be tracked as well, in order to be able to report all connected components of the residual graph at any time (as this components may be split later on). The machinery in [2] enables us to efficiently track all these events. Informally, the algorithm in [2] performs a walk through the cells of a planar subdivision induced by regions corresponding to to all locations of γ in which it meets an edge e of G, for each e. Each cell of this subdivision corresponds to placements of γ where the residual graph G_γ (in each of these locations) remains unchanged. The transition between adjacent cells constitute the critical events, as in each transition γ meets a new edge or becomes apart from an edge. The running time of the algorithm is proportional to the complexity of this subdivision, with a multiplicative factor of $O(\log^2 n)$. Thus we only need to show these bounds in our case in order to bound the running time of the verifier procedure. Note that we can also check each newly created component whether it contains a node of V_D, and count the number of these nodes with the same running time. From the theory of "arrangements" (see, e.g., [3]) subdivisions of simply-shaped regions in d-space have only $O(n^d)$ cells. When γ is a unit disk, the regions comprising our subdivision are *hippodromes* [2], i.e., the *Minkowski Sum* of an edge $e \in E$ and a unit disk [26], or formally, the set $\{x \in \mathbb{R}^2 \mid C(x) \cap e \neq \emptyset\}$, where $C(p)$ is the unit disk centered at p; see Figure to the right.



We have $O(n)$ hippodromes, as there are $O(n)$ edges in G and the planar subdivision will contain only $O(n^2)$ cells, implying that the verifier runs in $O(n^2 \log^2 n)$ time. For arbitrary radius disks, we represent γ by a point $(x, y, r) \in \mathbb{R}^3$ where r is the radius of γ and (x, y) is its center. Thus, we get a three-dimensional subdivision of simply-shaped regions with only $O(n^3)$ cells, from which it will follow that the verifier runs in $O(n^3 \log^2 n)$ time. Specifically, for each link $e \in E$, we consider the bi-variate function $f_e : \mathbb{R}^2 \to \mathbb{R}$ defined by $f_e(p) = d(p, e)$ where $d(p, e)$ is the distance from p to the closest point on e. The region thus created is the Minkowski Sums of the edge $e \in E$ and a *cone*, and, by definition, each point (x, y, r) inside this region represents a disk of radius r centered at (x, y) which intersects e. The hippodrome of e defined for unit disks is obtained at the level set $\{p \in \mathbb{R}^2 \mid f_e(p) = 1\}$. We defer the case of lines to the full version [13] due to lack of space.

Corollary III.1. *The running time of the verifier procedure is $O(n^2 \log^2 n)$ if γ is a line or a unit disk, or $O(n^3 \log^2 n)$ if γ is an arbitrary-radius disk.*

Recall that the ε-net finder runs in $O(n)$ time. Putting this together with our verifier procedures into the iterative reweighting scheme,we prove Theorem I.1. To prove Theorem I.2, we note from our discussion on ε-nets that $O((1/\varepsilon)\log(1/\varepsilon))$ replicas suffice to recover from all components of size at least εn for some threshold $\varepsilon > 0$. We can, thus, produce this set by randomly sampling followed by testing only sufficiently large components.

Remark. Fiber backbone networks are typically much less complex than worse-case planar networks due to constraints on the way the fiber is laid; for example, they tend to follow major arteries of the transportation network for easy servicability. Thus, as was noted in [1], for fixed-radius disasters, the corresponding planar subdivision has only a near-linear complexity rather than quadratic, from which it follows the running time for the verifier is nearly-linear. This speeds up the running time of our hitting-set algorithm to be close to $O(nk^*)$ (up to some polylogarithmic factor).

C. Extension: Modeling failure events of arbitrary shape

We now resolve the case of arbitrary-shape disasters in the replica model, where these shapes are modeled as union of (possibly, arbitrary-radius) disks and/or lines; in a typical setting an arbitrary shape is approximated in this manner. Letting Γ be a set of (arbitrary-radius) disks and lines, we now consider the residual graph of G after placing b distinct regions $\gamma_1, \ldots, \gamma_b \in \Gamma$, where $b > 0$ is an integer parameter. We obtain the range space $(V, \mathcal{C}_\Gamma^{(b)})$ consisting of the nodes V of G and the collection of all subsets of V that appear in a connected component of $G_{\bigcup_{i=1}^b \gamma_i}$ for some b-tuple of regions $\gamma_1, \ldots, \gamma_b \in \Gamma$, where $G_{\bigcup_{i=1}^b \gamma_i}$ is the residual graph after excluding from G all nodes and links meeting $\bigcup_{i=1}^b \gamma_i$.

We now observe that each connected component of $G_{\bigcup_{i=1}^b \gamma_i}$ is obtained by the *intersection* of b connected components (that is, b ranges) from the original range space $(V, \mathcal{C}_\Gamma)$. Indeed, such a connected component is affected by b distinct regions $\gamma_1, \ldots, \gamma_b$, each of which defines a connected component C_{γ_i} in its own (individual) residual graph G_{γ_i}, $i = 1 \ldots, b$. Thus a connected component of $G_{\bigcup_{i=1}^b \gamma_i}$ appears in the intersection $\bigcap_{i=1}^b C_{\gamma_i}$ (note that this intersection may not be connected). Using standard VC-dimension properties and the fact that the original range space $(V, \mathcal{C}_\Gamma)$ has VC-dimension 4, one can show that the VC-dimension of $(V, \mathcal{C}_\Gamma^{(b)})$ is at most $O(b)$; see, e.g., [16].

ε-net finder and verifier procedures. By the ε-net bound in [19] it follows that $(V, \mathcal{C}_\Gamma^{(b)})$ has an ε-net of size $O\left(\frac{b \log(1/\varepsilon)}{\varepsilon}\right)$ (in fact, a random sample of that

size is an ε-net with constant probability). We are not aware of how to efficiently extend the verifier to the case $b > 1$. Instead, we just resort to a brute-force algorithm, which exhausts all $n^{O(b)}$ ranges in $(V, \mathcal{C}_\Gamma^{(b)})$, and then tests each of them with respect to the candidate hitting set. Omitting any further details, the running time is $n^{O(b)}$ (with a constant of proportionality in the exponent being slightly larger than that in the bound on the number of ranges). Incorporating into the iterative reweighting scheme we obtain Theorem I.3.

D. Extension: Erasure Coding

In this section, we extend the previous algorithm to the case when erasure coding is used. We are given a file f split into k segments and assume that a (k', k)-coding scheme is applied where $k' \geq k$. Our goal is that, after a disastrous event, each surviving node connected to at least k other nodes will have access to at least k segments of f, to enable the recovery of the data in each such node. We aim to choose a smallest subset of coding nodes that guarantees this property. This is particularly applicable in scenarios where rateless codes are used and as many encoded segments may be generated as necessary.

A reduction to multi-hitting set. Analogous to hitting sets, a ***k-hitting set*** for a range space $(X, \mathcal{R})$ is a set $S \subseteq X$ such that, for every $R \in \mathcal{R}$, $|S \cap R| \geq k$. The MIN-k-HITTING-SET problem is that of computing a minimum cardinality k-hitting set. In order to maintain an accessibility of each surviving node v to at least k other nodes storing segments of f (so that v can fully reconstruct f), we need to solve a MIN-k-HITTING-SET problem for the range space $(V, \mathcal{C}_\Gamma)$. A subset $V_D \subset V$ is a k-hitting set for $(V, \mathcal{C}_\Gamma)$, if each component in $\mathcal{C}_\Gamma$ (of size $\geq k$) contains at least k nodes of V_D.

We derive a solution for this problem by using the machinery of Chekuri *et al.* [9], who obtained an $O(d \log k^*)$-approximation factor (computed in expected polynomial time, given a polynomial-time verifier procedure) for the MIN-k-HITTING-SET problem when d is the VC-dimension. Thus in our scenario, since $d = 4$, we obtain a $O(\log k^*)$-approximation factor. We note that the algorithm in [9] was originally presented for the set-cover problem (dual to hitting set) and allows repetitions, which is why we allow arbitrary storage capacity. The majority of the time is in computing a solution a linear programming system (which we then relax). Using standard algorithms this can be completed in polynomial time, completing the proof of Theorem I.4.

IV. SIMULATIONS

In this section, we conduct a simulation study to evaluate the effectiveness of a realistic data storage network to disasters when using the proposed algorithms for realistic networks. We consider the Qwest fiber backbone network in the USA from 2010 which consists of 153 nodes. Current network maps can be found in [7]. We consider the case where redundancy is carried out by replication using the following modified replication scheme: replication happens continuously in time and at each time step, a node stores a replication with a predetermined probability p. This results with an expected number of pn nodes updated in each time step, where n is the total number of nodes (in practice either there is a centralized entity that sends a replication to the selected nodes, or, the replication is broadcasted to all nodes and saved in each node with probability p to save storage). This models a scenario where new versions are periodically sent and we wish to recover as recent a version as possible. Using this model of file versions enables us to quantitatively measure the effectiveness of the algorithm in terms of the version staleness rather than a simple yes or no to whether the file can be retrieved.

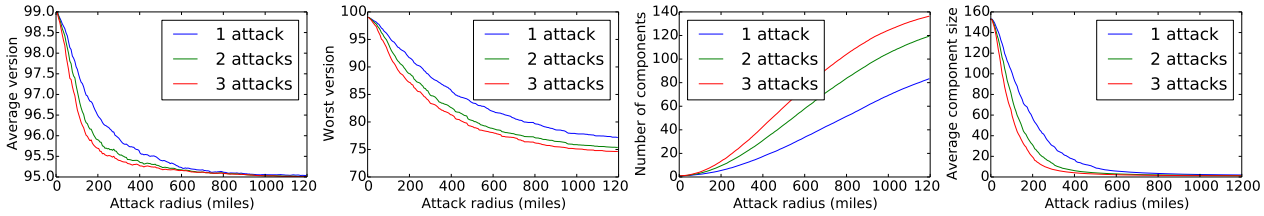
We use values of $p = 0.2$ and consider 100 time steps. After we distribute the file to the selected nodes in each time step, we cause a geographically correlated failure by simulating up to 3 attacks modeled as a disks with radius roughly between 0 and 1200 miles. Each edge that touches the attack fails. Each point in Fig. 2 is averaged over 200 random realizations of attack locations.

We first measure the average staleness at each node as a function of attack radius, i.e., we average over the newest version in each connected component post-attack; see Fig. 2(a). Note that as we consider 100 time steps, version i is distributed in time step i ($0 \leq i \leq 99$). We note that even for large attacks radii, the average version is not lower than 95. This is expected and can be explained based on the following equation:

$$P_m^x = (1-p)^{m(100-x-1)}(1-(1-p)^m),$$

where P_m^x is the probability that the best version in a component of size m is x. As the attack radius increases, in the most extreme case, all nodes are isolated and the expected version in a component can be computed by $\sum_{x=0}^{99} x P_1^x$. As we selected $p = 0.2$, the expected version is indeed 95. We note that the above equation can't be used when the components vary in size; however, the figure shows the behavior for the network that for radius values of 200 miles it is already possible to obtain an average value close to 95 with only three attacks.

Fig. 2(b) shows the worst version which is defined as the minimum value amongst the newest versions for the various components. This metric shows what on average would be the value for the worst component. The expected worst version for the worst case (i.e., each



(a) (b) (c) (d)

Fig. 2. The performance of the proposed model under different number of attacks of different radii. (a) Average of best version in each connected component; (b) Average of the version in the worst component; (c) The average number of connected components; (d) The average size of a connected component

component is of size 1), is as follows:

$$P_n^{'x} = \left(1-(1-p)^{(100-x)}\right)^n - \left(1-(1-p)^{(100-x-1)}\right)^n,$$

where $P_n^{'x}$ is the probability that the worst version amongst n individual nodes equals x. Using $P_n^{'x}$ to compute the expectation and as $n = 153$, the expected value for the worst version is 74.3. We see that compared to the average version, the worst component on average has a much lower version values than the average version taken over all components. However, to obtain a worst version of less than 90, 3 attacks of radius greater than 200 miles are needed. Fig. 2(c) and 2(d) provide insight on the change in network topology with number and radius of attacks. While the number of connected components (Fig. 2(c)) increases in a nearly linear fashion (different inclines for different number of attacks), the average size of the component dramatically decreases as the attack radius increases (Fig. 2(d)).

Acknowledgments: The work of AE and SS was supported by NSF grant CNS-1017114 and CCF-12-16689, and by DTRA grant HDTRA1-13-1-0021.

REFERENCES

[1] P. Agarwal, A. Efrat, S. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman. Network vulnerability to single, multiple, and probabilistic physical attacks. *MILCOM*, 2010.

[2] P. Agarwal, A. Efrat, S. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman. The resilience of wdm networks to probabilistic geographical failures. *Trans. Netw.*, 21(5):1525–1538, 2013.

[3] P. K. Agarwal and M. Sharir. Arrangements and their applications. *Handbook of Computational Geometry*, 49–119, 2000.

[4] S. Banerjee, S. Shirazipourazad, P. Ghosh, and A. Sen. Beyond connectivity - new metrics to evaluate robustness of networks. In *Proc. 12th IEEE HPSR*, pages 171 –177, 2011.

[5] S. Banerjee, S. Shirazipourazad, and A. Sen. On region-based fault tolerant design of distributed file storage in networks. *INFOCOM* 2012.

[6] H. Brönnimann and M. Goodrich. Almost optimal set covers in finite vc-dimension. *Discrete Comput. Geom.*, 463–479, 1995.

[7] CenturyLink Network Maps.

[8] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. Bigtable: A distributed storage system for structured data. *ACM Trans. Comput. Syst.*, 26(2):4:1–4:26, 2008.

[9] C. Chekuri, K. L. Clarkson, and S. Har-Peled. On the set multicover problem in geometric settings. *ACM Trans. Algorithms*, 9(1):9:1–9:17, 2012.

[10] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels. Dynamo: amazon's highly available key-value store. *SIGOPS Oper. Syst. Rev.*, 41(6):205–220, 2007.

[11] A. Dimakis, V. Prabhakaran, and K. Ramchandran. Decentralized erasure codes for distributed networked storage. *Information Theory, IEEE Transactions on*, 52(6):2809 – 2816, 2006.

[12] A. Dimakis, K. Ramchandran, Y. Wu, and C. Suh. A survey on network codes for distributed storage. *Proc. IEEE*, 99(3):476 –489, 2011.

[13] http://www.cs.arizona.edu/~alon/papers/data_recovery.pdf.

[14] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[15] H. Hanani (Chaim Chojnacki). Über wesentlich unpläättbare Kurven im drei-dimensionalen Raume. *Fundamenta Mathematicae*, 23:135–142, 1934.

[16] S. Har-Peled. *Geometric Approximation Algorithms*. Mathematical Surveys and Monographs, American Mathematical Society, 2011.

[17] D. Haussler and E. Welzl. ε-nets and simplex range queries. *Discrete Comput. Geom.*, 2(1):127–151, 1987.

[18] A. A. Jiang and J. Bruck. Network file storage with graceful performance degradation. *Trans. Storage*, 1(2):171–189, 2005.

[19] J. Komlós, J. Pach, and G. Woeginger. Almost tight bounds for ε-nets. *Discrete Comput. Geom.*, 7(2):163–173, 1992.

[20] Level 3 Communications. Network map. http://www.level3.com/interacts/map.html.

[21] Q. Malluhi and W. Johnston. Coding for high availability of a distributed-parallel storage system. *IEEE Trans. Parallel Distrib. Syst.*, 9(12):1237 –1252, 1998.

[22] M. Naor and R. M. Roth. Optimal file sharing in distributed networks. *SIAM J. Comput.*, 24(1):158–183, 1995.

[23] S. Neumayer and E. Modiano. Network reliability with geographically correlated failures. *INFOCOM*, 2010.

[24] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano. Assessing the impact of geographically correlated network failures. In *MILCOM*, 2008.

[25] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano. Assessing the vulnerability of the fiber infrastructure to disasters. *INFOCOM*, 2009.

[26] J. Pach and P. Agarwal. *Combinatorial Geometry*. Wiley, 2011.

[27] D.A. Patterson, G. Gibson, and R. H. Katz. A case for redundant arrays of inexpensive disks (raid). *SIGMOD* , 109–116, 1988.

[28] A. Sen, S. Murthy, and S. Banerjee. Region-based connectivity: a new paradigm for design of fault-tolerant networks. In *Proc. IEEE HPSR*, 2009.

[29] W. Tutte. Toward a theory of crossing numbers. *J. Comb. Theory*, 8(1):45–53, 1970.

V. Proof of Theorem III.1

Assume to the contrary that there are five vertices $v_1, \ldots, v_5$ of G such that $\{v_1, \ldots, v_5\}$ is shattered by $H(G, \mathcal{C})$. In particular this means that for every $1 \leq i < j \leq 5$ there is $C_{ij} \in \mathcal{C}$ such that v_i and v_j belong to the same connected component of $G_{C_{ij}}$ and this connected component does not contain any of $\{v_1, \ldots, v_5\} \setminus \{v_i, v_j\}$. Hence for every $1 \leq i < j \leq 5$ there exists a simple path P_{ij} in $G_{C_{ij}}$ connecting v_i and v_j. Observe that, by definition, P_{ij} does not meet the curve C_{ij}.

In order to handle more elegantly the possible case in which two paths P_{ij} and $P_{k\ell}$ share common edges and vertices other than their endpoints, we do the following: For every $1 \leq i < j \leq 5$ we draw a curve Q_{ij} connecting v_i and v_j such that Q_{ij} follows very closely the path P_{ij}. We make sure that Q_{ij} and P_{ij} meet exactly the same set of curves in $\mathcal{C}$. The difference between P_{ij} and Q_{ij} is that Q_{ij} is drawn in a rather free fashion and it does not meet any edge of G more than at a finite number of points (see Figure 3 for an illustration). We will also make sure that at every point that two curves Q_{ij} and $Q_{k\ell}$ meet they either properly cross or share a common endpoint.

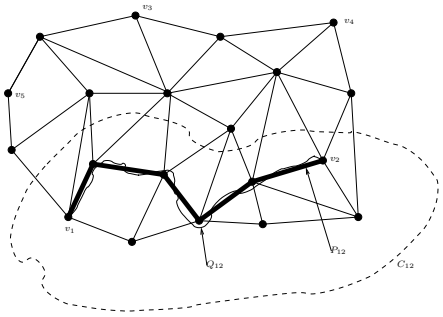


Fig. 3. The path P_{12} and the curve Q_{12}.

Considering only the curves Q_{ij} for $1 \leq i < j \leq 5$ and the vertices $v_1, \ldots, v_5$, we obtain a drawing of K_5 in the plane. As K_5 is not a planar graph, by a theorem of Hannani and Tutte [15], [29], there exist four distinct indices, that without loss of generality we assume they are $1, 2, 3$, and 4, such that Q_{12} and Q_{34} cross an odd number of times.

Recall that the curve Q_{12} is disjoint from C_{12}. We may assume, without loss of generality, that Q_{12} is contained in the area bounded by the simple closed curve C_{12} rather than in its complement. Indeed, if it happens that Q_{12} is contained in the region that is not bounded by C_{12}, we can apply inversion mapping with respect to a point w_0 that lie in the area bounded by C_{12} (specifically, this mapping takes a point z in the complex plane to the point $\frac{1}{z-w_0}$). Since the inversion mapping is continuous it does not effect any incidence relation and the global combinatorial picture of our curves remains the same. However, after applying the inversion mapping the curve Q_{12} is now contained in the region bounded by C_{12} (see Figure 4 for an illustration).

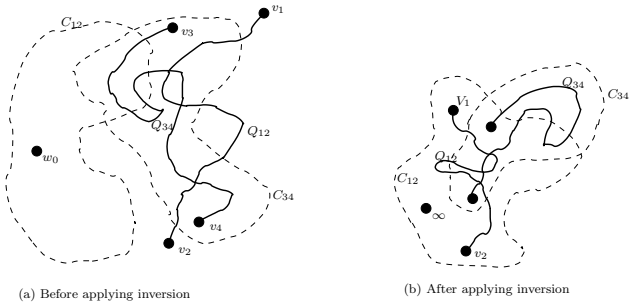


Fig. 4. Applying the inversion mapping.

Let x_1 be the first intersection point of Q_{12} with Q_{34} as we traverse Q_{12} from v_1 in the direction of v_2. Let x_2 be the first intersection point of Q_{12} with Q_{34} as we traverse Q_{12} from v_2 in the direction of v_1 (it may happen that $x_1 = x_2$, in which case Q_{12} and Q_{34} meet only at $x_1 = x_2$). Similarly, let x_3 be the first intersection point of Q_{34} with Q_{12} as we traverse Q_{34} from v_3 in the direction of v_4. Let x_4 be the first intersection point of Q_{34} with Q_{12} as we traverse Q_{34} from v_4 in the direction of v_3 (see Figure 5).

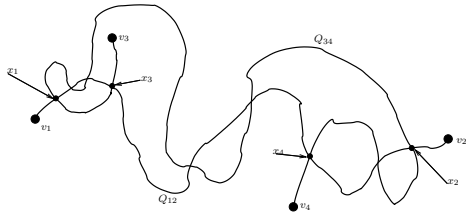


Fig. 5. The definition of x_1, x_2, x_3, and x_4.

For convenience we introduce the following notation. Given a curve Q and two points a and b on Q we denote by $Q(a, b)$ the portion of Q delimited by a and b (we assume Q is not a closed curve).

We note the following simple yet important observation.

Claim V.1. 1) *The curve C_{34} must cross $Q_{12}(v_1, x_1)$.*

2) *The curve C_{34} must cross $Q_{12}(v_2, x_2)$.*
3) *The curve C_{12} must cross $Q_{34}(v_3, x_3)$.*
4) *The curve C_{12} must cross $Q_{34}(v_4, x_4)$.*

Proof. Because the four statements are very similar we prove only the first one leaving the rest to the reader. Assume to the contrary that C_{34} does not cross $Q_{12}(v_1, x_1)$. As we traverse the path P_{12} from v_1 in the direction of v_2, let v be the first vertex on P_{12} that is also a vertex of P_{34}. Such a vertex must exist because P_{12} and P_{34} meet as Q_{12} and Q_{34} cross each other. Because G is a planar map P_{12} and P_{34} must first meet at a vertex of G. Now it is easy to see that, because C_{34} does not cross $Q_{12}(v_1, x_1)$, v_1 belongs to the same connected component of $G_{C_{34}}$ as v_3 and v_3, contradicting the definition of C_{34}. □

Consider the curve Q_{12}. As we traverse this curve from v_1 in the direction of v_2 let $z_1, \ldots, z_k$ be the intersection points of Q_{12} and C_{34} listed in the order they are encountered.

We claim that $Q_{12}(v_1, z_1)$ does not meet the curve Q_{34}. Indeed, this follows from Claim V.1 because if $Q_{12}(v_1, z_1)$ meets Q_{34} at a point w, then the point x_1 must lie on $Q_{12}(v_1, w)$. In particular $Q_{12}(v_1, x_1)$ does not meet the curve Q_{34}, as z_1 is the first meeting point of Q_{12} and Q_{34} as we traverse Q_{12} from v_1 in the direction of v_2. This is a contradiction to Claim V.1. In a completely analogous manner one can show that $Q_{12}(v_2, z_k)$ does not meet the curve Q_{34}.

We will reach a contradiction by showing that each of the portions $Q_{12}(z_i, z_{i+1})$ meets Q_{34} an even number of times. Indeed, this will contradict the assumption that Q_{12} and Q_{34} meet an odd number of times.

Let $1 \leq i < k$ and consider the portion $Q_{12}(z_i, z_{i+1})$ of Q_{12}. Recall that z_i and z_{i+1} are two points on C_{34}. The points z_i and z_{i+1} divide C_{34} into two subarcs that we denote by A and B. Observe that the union of A and $Q_{12}(z_i, z_{i+1})$ is a simple closed curve that we denote by W_A (see Figure 6). Similarly, the union of B and $Q_{12}(z_i, z_{i+1})$ is a simple closed curve that we denote by W_B.

Because every two simple closed curves cross each other an even number of times, C_{12} crosses W_A an even number of times. It follows that C_{12} crosses A an even number of times, as C_{12} is disjoint from $Q_{12}(z_i, z_{i+1})$. Similarly, C_{12} crosses B an even number of times. However, because $\mathcal{C}$ is a family of pseudo-circles C_{12} and C_{34} may cross each other at most twice. This implies that C_{12} is disjoint either from A or from B.

Without loss of generality assume that C_{12} is disjoint from A. We claim that C_{12} does not intersect

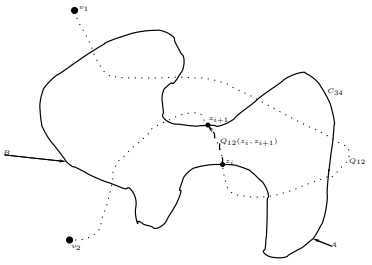


Fig. 6. The definition of A and B.

the region bounded by W_A. Indeed, because C_{12} is disjoint from A, then it is also disjoint from W_A as $W_A = A \cup Q_{12}(z_i, z_{i+1})$ and C_{12} is disjoint from $Q_{12} \supset Q_{12}(z_i, z_{i+1})$. It follows that if C_{12} intersect the region bounded by W_A, then it is contained in it. But this is impossible because the region bounded by C_{12} contains, by assumption, the curve Q_{12}.

Next, we claim that neither v_3 nor v_4 can be located in the region bounded by W_A. This is because, by Claim V.1, as we traverse Q_{34} from say v_3 in the direction of v_4 we must hit the curve C_{12} before hitting x_3, the first intersection point we encounter with Q_{12}. However, if v_3 is located in the region bounded by W_A, then because C_{12} does not intersect this region it follows that as we traverse Q_{34} from say v_3 in the direction of v_4 we must first hit a point of W_A. Of course we cannot hit a point of A because $A \subset C_{34}$ and Q_{34} is disjoint from C_{34}. Therefore, we must hit a point of $W_A \setminus A = Q_{12}(z_i, z_{i+1}) \subset Q_{12}$. This contradicts our assumption. Similarly, we show that v_4 is not contained in the region bounded by W_A. It now follows from Jordan's theorem that Q_{34} meets W_A an even number of times. This implies that Q_{34} meets $Q_{12}(z_i, z_{i+1})$ an even number of times, as Q_{34} is disjoint from A.

We have thus reached the desired contradiction showing that Q_{34} meets each of the portions $Q_{12}(z_i, z_{i+1})$ an even number of times and therefore Q_{34} meets the entire curve Q_{12} an even number of times, contrary to our assumption.

We note that the proof applies to families of pseudo-line as well as pseudo-circles. A family $\mathcal{C}$ of curves in the plane is called a family of pseudo-lines if every two curves meet at most once and (very importantly for us) the ends of each curve in $\mathcal{C}$ are at infinity. A family of lines in the plane is an example for such a family.

By applying the inversion mapping that takes z in the complex plane to $\frac{1}{z}$ (assuming that the origin does not belong to any of the curves) a family of pseudo-lines becomes a family of (pairwise intersecting) pseudocir-

cles meeting each other also at the origin. Therefore, the proof for pseudo-lines follows immediately.