The Resilience of WDM Networks to Probabilistic Geographical Failures

Pankaj K. Agarwal, Alon Efrat, Shashidhara K. Ganjugunte,
David Hay, Swaminathan Sankararaman and Gil Zussman

***Abstract*—Telecommunications networks, and in particular optical WDM networks, are vulnerable to large-scale failures in their physical infrastructure, resulting from physical attacks (such as an Electromagnetic Pulse attack) or natural disasters (such as solar flares, earthquakes, and floods). Such events happen at *specific geographical locations* and disrupt specific parts of the network but their *effects cannot be determined exactly in advance*. Therefore, we provide a unified framework to model network vulnerability when the event has a *probabilistic nature*, defined by an arbitrary probability density function. Our framework captures scenarios with a number of simultaneous attacks, when network components consist of several dependent sub-components, and in which either a 1+1 or a 1:1 protection plan is in place. We use computational geometric tools to provide efficient algorithms to identify vulnerable points within the network under various metrics. Then, we obtain numerical results for specific backbone networks, demonstrating the applicability of our algorithms to real-world scenarios. Our novel approach allows to identify locations which require additional protection efforts (e.g., equipment shielding). Overall, the paper demonstrates that using computational geometric techniques can significantly contribute to our understanding of network resilience.**

***Index Terms*—Network survivability, geographic networks, network protection, computational geometry, optical networks.**

I. Introduction

TELECOMMUNICATION networks are crucial for the normal operation of all sectors of our society. During a crisis, telecommunication is essential to facilitate the control of physically remote agents, provide connections between emergency response personnel, and eventually enable reconstitution of societal functions. However, telecommunication networks rely heavily on physical infrastructure (such as optical fibers, amplifiers, routers, and switches), making them vulnerable to physical attacks, such as Electromagnetic Pulse (EMP) attacks, as well as natural disasters, such as solar flares, earthquakes, hurricanes, and floods [11], [19], [20], [54], [55].

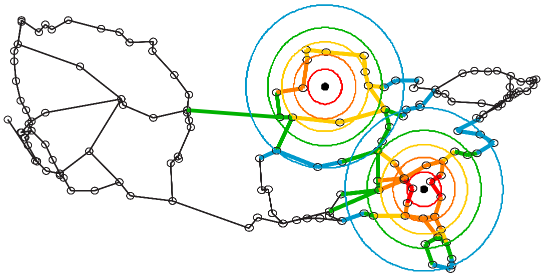


Fig. 1. The fiber backbone operated by a major U.S. network provider [43] and an example of two attacks with probabilistic effects (the link colors represent their failure probabilities).

Physical attacks or disasters affect specific geographical area and will result in failures of neighboring components. Therefore, it is important to consider their effects on the physical (fiber) layer as well as on the (logical) network layer. Increasingly, networks use a shared infrastructure to carry voice, data, and video simultaneously. Thus, failures in the this infrastructure will lead to a breakdown of vital services.

Although there has been significant research on network survivability, most previous works consider a small number of isolated failures or focus on shared risk groups (e.g., [9], [16], [35], [40], [49], [57] and references therein). On the other hand, work on large-scale attacks focused mostly on cyber-attacks (viruses and worms) (e.g., [8], [22], [33]). In contrast, we consider events causing a large number of failures in a specific geographical region.

This emerging field of *geographically correlated failures* has started gaining attention only recently [2], [25], [26], [37]–[39], [45], [46], [55]. However, unlike most of the recent work in this field, we focus on probabilistic attacks and on multiple simultaneous attacks. One example of such a scenario is shown in Fig. 1 which depicts the fiber backbone operated by a major U.S. network provider [43] and two attacks with probabilistic effects (the link colors represent their failure probabilities).

The effects of physical attacks can rarely be determined exactly in advance. The probability that a component is affected by the attacks depends on various factors, such as the distance from the attack's epicenter to the component, the topography of the surrounding area, the component's specifications, and even its location within a building or

Partial and preliminary versions of this paper appeared in Proc. Infocom'10 [3] and Proc. Milcom'10 [2].

Pankaj K. Agarwal and Swaminathan Sankararaman are with the Department of Computer Science, Duke University. email:{pankaj, swami}@cs.duke.edu. This work was done while Swaminathan Sankararaman was at the University of Arizona.

Alon Efrat is with the Department of Computer Science, The University of Arizona. email:alon@cs.arizona.edu

Shashidhara K. Ganjugunte is with Mentor Graphics. email:shashigk@cs.duke.edu. This work was done while Shashidhara K. Ganjugunte was at Duke University.

David Hay is with the Department of Engineering and Computer Science, Hebrew University. email:dhay@cs.huji.ac.il. This work was done while David Hay was with Columbia University.

Gil Zussman is with the Department of Electrical Engineering, Columbia University. email:gil@ee.columbia.edu

a system.[1] In this paper, we consider probability functions which are non-increasing functions of the distance between the epicenter and the component. We assume these functions have a constant description complexity, and allow them to be either continuous or discontinuous (e.g. histograms). Then, we develop algorithms that obtain the expected vulnerability of the network. Furthermore, while [25], [26], [37]–[39], [45], [46], [55] consider only a single event, our algorithms allow the assessment of the effects of several simultaneous events.

We focus on wavelength-routed WDM optical networks, especially at the backbone [40], [49]. We model the network as a graph, embedded in the plane, in which each node corresponds to an *optical cross-connect (OXC)* and each link corresponds to an optical fiber (which are usually hundreds or thousands of kilometers long). Along each link there are amplifiers, which are spaced-out approximately equally and are crucial to traffic delivery on the fiber. Data is transmitted on this graph on *lightpaths*, which are circuits between nodes. While lightpaths can be established by the network dynamically, lightpath-provisioning is a resource-intensive process which is usually slow. If many links fail simultaneously (as in the case of a physical attack or a large-scale disaster), current technology will not be able to handle very large-scale re-provisioning (see for example, the CORONET project [13]). Therefore, we assume that lightpaths are static, implying that if a lightpath is destroyed, all the data that it carries is lost. We note that our results are applicable to any network in which end-to-end paths are static and known in advance. This includes, for example, MPLS networks without label swapping at intermediate nodes.

We also consider networks that are protected by a *dedicated path protection* plan. Under such plans, every (primary) lightpath has a predefined backup lightpath on which data can be transmitted if the primary lightpath fails. Protection plans are pre-computed before a failure event, and therefore, it is reasonable to assume that they can be applied even after large-scale failures. Common approaches include 1+1 or 1:1 dedicated protection plans (see [40], [49]). Conceptually, in the 1+1 protection plan, the data is sent twice along primary and backup lightpaths, implying that data is lost only when both lightpaths fail simultaneously. A 1:1 dedicated protection, on the other hand, allows using a backup lightpath for low-priority traffic. Once the primary lightpath fails, traffic is shifted to the backup lightpath, and the low-priority traffic is disregarded.

Finally, we consider networks with dynamic restoration capabilities, i.e., where traffic may be dynamically rerouted in the event of an attack to avoid data loss. In general, devising efficient restoration algorithms, especially when required to handle large-scale failures, is a challenging task. Dynamic restoration schemes are more efficient in utilizing network capacity, but have slower recovery time and often cannot guarantee quality of restoration. With the current technology, large-scale dynamic restoration is mostly infeasible. However, this capability will emerge in future optical networks [13].

We note that, in between dedicated path protection and fully-dynamic restoration, there are other protection techniques that trade between the robustness of the network and the complexity of the technique (cf. [40] for a complete survey of these techniques). In this paper, we focus only on the two endpoints of this scale, while we leave for future research how to adapt our proposed algorithms to more sophisticated techniques.

Our goal is to identify the most vulnerable locations in the network, where vulnerability is measured either by expected number of failed components or by the expected total data loss. Our model allows for the consideration of failure probabilities of compound components by evaluating the effect of the attack on their sub-components (e.g., the failure probability of a fiber, due to failure of some amplifiers). We consider the vulnerability of the network in terms of three measures: (i) *expected component damage*: The expected number of network components directly damaged by attacks or the expected amount of traffic lost due to the attacks, (ii) *average two-terminal reliability*: The expected number of node pairs in the network which are able to communicate post-attack and (iii) *expected maximum flow*: the maximum post-attack flow.

We first develop algorithms for a single attack scenario under the first two vulnerability measures outlined above. Our algorithms provide a tradeoff between accuracy and running time; we can provide arbitrarily small errors, albeit with high running time. Although these algorithms have to be executed offline in preparation for disasters, efficiency is important as numerous options and topologies need to be considered. Moreover, our algorithms also work under deterministic attack effects and achieve better results than the prior ones [38].

Next, we consider the case of k simultaneous attacks under the vulnerability measure of expected component damage and provide approximation algorithms for computing the most vulnerable set of k locations. This problem is hard not only due to its probabilistic nature but also due to the combinatorial hardness of the deterministic problem.

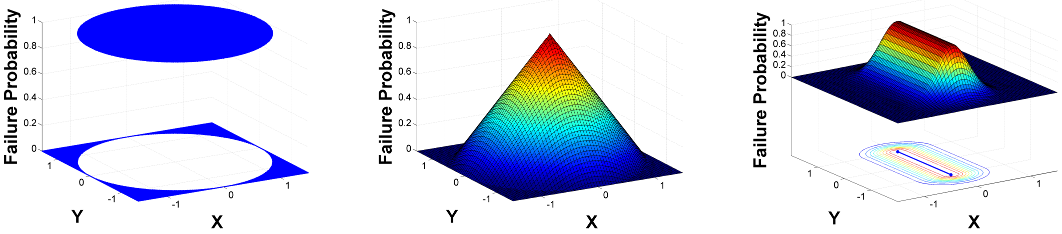
For networks with protection plans, we provide approximation algorithms to identify pairs of vulnerable locations that will have a high effect on both primary and backup paths. For future networks with dynamic restoration capability, network resilience can be measured in terms of the expected maximum flow measure. However, we show that computing this measure is *#P-Complete* and hence cannot be found in any reasonable time. We discuss options for mitigating this evaluation barrier.

Finally, we provide experimental results demonstrating the applicability of our algorithms to real backbone networks. Among other things, we show that even when the approximation algorithms only guarantee low accuracy (thereby, having low running time), the results are very close to optimal. This would allow checking various scenarios and settings relatively fast.

In summary, the contributions of this paper are fourfold:

1) This is the first paper to present a general probabilistic model for geographically-correlated failures, as well as efficient approximation algorithms for finding the most vulnerable locations in the network under two measures. Our algorithms trade accuracy with efficiency, where we can provide arbitrarily small errors, albeit with high running time. In addition, we provide the first set of

[1] Characterizing the failure probability function of each component is orthogonal to this research, and we assume it is given as an input.



(a) (b) (c)

Fig. 2. Failure probability function: (a) deterministic model, (b) probabilistic attack (inverse distance function) for a node, (c) probabilistic attack (Gaussian function) for a link.

algorithms that deal with simultaneous attacks.

2) We provide algorithms that take into account pre-computed protection plans.
3) For networks with dynamic restoration capabilities, the network resilience corresponds to the maximum post-attack flow. We show that computing this measure is *#P-Complete* and discuss options for mitigating this evaluation barrier.
4) Importantly, this paper demonstrates that geometric techniques can significantly contribute to our understanding of network resilience.

The rest of the paper is organized as follows: Section II reviews related work, and Section III states the network model and the problem. we present in Section IV algorithms for analyzing network vulnerability by a single location, and extend them to multiple attacks in Section V. We study the effect of protection and restoration plans in Sections VI and VII. We present experimental results in VIII and conclude and discuss future work in Section IX.

II. Related Work

Network survivability and resilience is a well-established research area (e.g., [9], [40], [49], [57] and references therein). However, most of the previous work in this area and, in particular in the area of physical topology and fiber networks (e.g., [16], [35]), focused on a small number of fiber failures (e.g., simultaneous failures of links sharing a common physical resource, such as a cable, conduit, etc.). Such correlated link failures are often addressed systematically by the concept of *shared risk link group* (SRLG) [28]. Additional works explore dependent failures, but do not specifically make use of the causes of dependence [32], [50], [52].

In contrast with these works, we focus on failures within a specific geographical region (e.g., [10], [20], [54]), implying that the failed components do not necessarily share the same physical resource. To the best of our knowledge, *geographically correlated failures* have been considered only in a few papers and under very specific assumptions [25], [26], [37]–[39], [45], [55]. In most cases, the assumption is that the failures of the components are deterministic and that there is a single failure. Perhaps closest to this paper are the problems studied in [10], [17], [18], [38], [46], and [51]. In particular, Neumayer *et al.* [38] recently obtained results about the resilience of fiber networks to geographically correlated failures when attacks have a circular area of effect in which links and nodes may fail. However, they only consider a single attack scenario with deterministic effects. Rahnamay-Naeini *et al.* [44], on the other hand, consider a stochastic setting with multiple attacks. However, unlike our paper, they deal with random attack locations and not with probabilistic effects of a failure on nearby components.

Another closely related theoretical problem is the *network inhibition problem* [41], [42], in which the objective is to minimize the value of a maximum flow in the graph, where there is a cost associated with destroying each edge, and a fixed budget is given for an orchestrated attack (namely, removing a set of edges whose total destruction cost is less than the budget). However, previous works dealing with this setting and its variants (e.g., [12], [42]) did not study the removal of (geographically) neighboring links.

Notice that when the logical (i.e., IP) topology is considered, wide-spread failures have been extensively studied [22], [33]. Most of these works consider the topology of the Internet as a random graph [8] and use percolation theory to study the effects of random link and node failures on these graphs. These studies are motivated by failures of routers due to attacks by viruses and worms rather than physical attacks.

III. Model and Problem Formulation

The optical network is represented as a graph $G = (V, E)$, where V is a finite set of nodes in the plane, and E is a set of links. We assume that each link is a straight line segment. Recall that each node corresponds to an optical cross-connect (OXC) and each link corresponds to an optical fiber. Each link $e \in E$ has a capacity $c_e \geq 0$. A lightpath π is a path in G; let t_π be the amount of data transmitted over π per unit of time.

In certain types of attacks, links are not affected directly. Recall that each link has a sequence of amplifiers. A link becomes unusable if any of the amplifiers becomes unusable. In such a case, we model amplifiers also as nodes of G and the portions of a link between two adjacent amplifiers are considered edges. We consider nodes and edges of G as *simple components*, and lightpaths as *compound components*. A link is a simple or compound component, depending on whether it is regarded as a single edge of G or a sequence of amplifiers.

The input is a set $Q = \{q_1, \ldots, q_m\}$ of network components; each component q has an associated *weight* w_q indicating either lightpath traffic or link capacity.

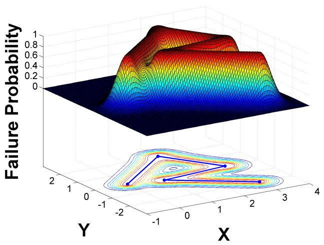


Fig. 3. Failure probability for a compound component.

Probabilistic Attack Model. An attack induces a spatial probability distribution on the plane, specifying the damage probability at each location (see Fig. 2). First consider simple components. We define the probability distribution function $f : Q \times \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$. Given an attack location $p \in \mathbb{R}^2$ and $q \in Q$, $f(q,p)$ is the probability that q is affected by an attack at p. Let $d(q,p)$ be the Euclidean distance between p and q.[2] We assume f is non-increasing of $d(p,q)$ and of constant description complexity.[3] We allow f to be discontinuous, e.g., a piecewise-constant function (histogram). For a fixed $q \in Q$, we use the function $f_q : \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$ to denote its probability distribution function, as the function of the location of attack. Here are some examples of f_q: in the deterministic setting, $f(q,p)$ is 1 if $d(p,q) \leq r$ and 0 otherwise, for some parameter r. Alternatively, one could use a more sophisticated function, for example, where $f_q(p)$ depends on the distance from p to q or the length of the portion of link q within the attack radius, which also decreases with distance. In many applications $f(q,p)$ is given, or can be computed as a function of the distance from p to q. Two examples of f_q that we use in our experimental results are:

- f_q decreases linearly with the Euclidean distance, e.g., $f(q,p) = \max\{0, 1 - d(q,p)\}$ (the circular area of effect is similar to the geometric models in [38], [45]); and
- f_q decreases exponentially with $d(p,q)$, e.g. Gaussian distribution $f(q,p) = \beta e^{-\alpha d(q,p)^2}$ for constants $\alpha, \beta > 0$, chosen appropriately to normalize the distribution.

For a compound component π composed of a sequence of simple components $\langle q_1, \ldots, q_r \rangle$, we define its probability of being damaged by an attack at p, $f_\pi(p)$, to be the probability that at least one of its simple component if damaged, i.e.,

$$f_\pi(p) = 1 - \prod_{q \in \pi} (1 - f_q(p)). \tag{1}$$

Figures 2 and 3 illustrate cases where f_π decreases exponentially with the distance for both types of components. A simpler definition of the probability of failure of π is $f_\pi(p) = |\pi'|/|\pi|$ where $\pi' = \{q \in \pi \mid f_q(p) \geq \delta\}$ for some fixed parameter $\delta > 0$. For networks with protection plans (see Section VI), we assume that data is lost, if and only if both the primary and backup lightpaths are affected.

Given Q and a fixed integer $k \geq 1$, our goal is to find a set P of k locations so that simultaneous attacks at P have the highest expected impact on the network. We consider three measures of impact of P on the network: (i) expected component damage, (ii) average two-terminal reliability, and (iii) expected maximum flow.

Expected Component Damage. For a set of attack locations P, let $\Phi(Q,P)$ denote the expected total weight of failed components in Q (see the example in Fig. 4). By linearity of expectation, we get

$$\Phi(Q,P) = \sum_{q \in Q} w_q \left(1 - \prod_{p \in P} (1 - f_q(p)) \right). \tag{2}$$

If $P = \{p\}$, we set

$$\Phi(Q,p) := \Phi(Q,P) = \sum_{q \in Q} w_q f_q(p).$$

For a given integer $k \geq 1$, let $\Phi(Q,k) = \max_{|P|=k} \Phi(Q,P)$ and $\Phi(Q) = \Phi(Q,1)$.

The weight w_q of each component enables us to define various measures in a unified manner: if Q is the set of amplifiers and w_q is set to 1 (for all q), then $\Phi(Q,P)$ is the expected number of failed amplifiers. Similarly, if Q is the set of fibers and for any fiber q, $w_q = c_q$ (q's capacity), then $\Phi(Q,P)$ yields the expected capacity loss of attacks in P. Finally, if Q is the set of lightpaths and $w_q = t_q$, then $\Phi(Q,P)$ is the expected loss in traffic, unless there is a protection (or restoration) plan in place. It is important to notice that, by linearity of expectation, $\Phi(Q,P)$ corresponds to the expected value of the measure under consideration, regardless of any dependency between the various components in Q. Therefore, even in the extreme situations in which two components share the same physical resource (e.g., lightpaths that share the same fiber, or fibers that share the same conduit), one can evaluate $\Phi(Q,P)$ by considering each component separately.

When components are points in the plane (that is, amplifiers or OXCs) and $f_q(p) = \max\{0, 1 - d(p,q)\}$, the problem is related to the Fermat-Weber problem [21], [34] (i.e., finding a point that minimizes the average distance to a given set of points). However, the approximate solutions for the Fermat-Weber problem and our problem can be quite different.

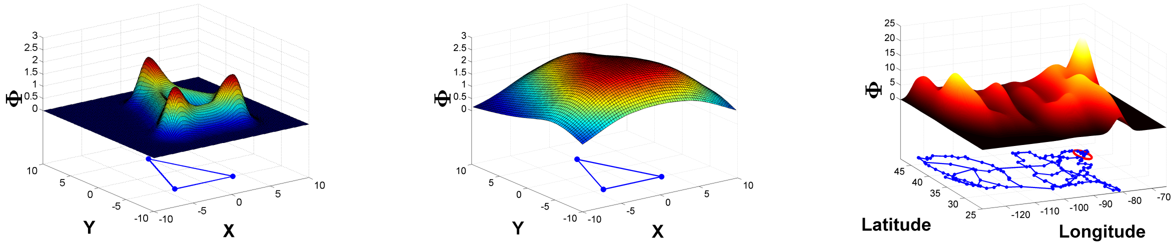
Average Two-Terminal Reliability. Given a set of probabilities of failure on the network components (induced by the attacks at locations P), the two-terminal reliability for a given node pair s, d in the network is the probability that they remain connected after the attack. The average two-terminal reliability, denoted by $\chi(Q,P)$ is the expected number of node-pairs which remain connected after the attack. Formally,

$$\chi(Q,P) = \frac{1}{|V|^2} \sum_{i,j \in V} \chi_{ij}(P)), \tag{3}$$

where $\chi_{ij}(P)$ is the probability that i is connected to j given the set P of attack locations. The quantity χ measures the

[2] More precisely, $d(p,q)$ is the minimal Euclidean distance between p to any point along q: $d(p,q) = \min_{x \in q} \|pq\|$, where $\|\cdot\|$ is the Euclidean distance.

[3] Intuitively, by *constant description complexity* we mean functions that can be expressed as a constant number of polynomials of constant maximum degree or simple distributions like the Gaussian distribution.



(a)

(b)

(c)

Fig. 4. (a,b): Expected damage for a triangle network and Gaussian probability distribution function with (a) small variance, (b) large variance. (c) Expected damage for a fiber network.

network's post-attack connectivity. For an integer $k \geq 1$, let $\chi(Q,k) = \max_{|P|=k} \chi(Q,P)$ and $\chi(Q) = \chi(Q,1)$.

Given probabilities of failure on links/nodes in a network, the problem of computing the *two-terminal reliability* (the probability that a specific pair of nodes is connected) and *all-terminal reliability* (the probability that some pair of nodes is disconnected) are well-known intractable problems [15] indicating that our problem is intractable as well. In the case of the all-terminal reliability problem, there exists a randomized fully polynomial time approximation scheme [29], [30] but the problem is significantly different from our problem.

Expected Maximum Post-Attack Flow. This quantity measures the maximum flow in the network once the components have failed between predetermined source and destination nodes. This is useful to determine the location maximizing data loss in networks with dynamic restoration capabilities (which re-route traffic to avoid data loss). We show that the problem of finding such a location is intractable.

IV. Assessing Vulnerability to a Single Attack

In this section, we present algorithms for computing the vulnerability of a set Q of simple or compound components to a single attack. Section IV-A describes an approximation algorithm for computing the maximum expected damage by a single attack for the case of simple components, IV-B extends this algorithm to compound components and IV-C describes approximation algorithms for computing a location minimizing average two-terminal reliability. Our algorithms have a tunable parameter $\varepsilon > 0$ providing a tradeoff between accuracy and efficiency. We note that, in order to measure the performance of our algorithms, we introduce other parameters but these are not inputs to the algorithm. We begin by introducing two geometric concepts, which will be used by the algorithms.

Arrangement. Let $\Gamma = \{\gamma_1, \ldots, \gamma_m\}$ be a set of (simple) geometric regions (e.g., disks, triangles, hippodromes) in $\mathbb{R}^2$; regions in Γ may overlap. The *arrangement* of Γ, denoted by $\mathcal{A}(\Gamma)$, is the planar subdivision induced by Γ. Namely, its vertices are the intersection points of the boundaries of regions in Γ, its edges are the maximal connected portions of the boundaries of the regions not containing a vertex, and its faces are the maximal connected regions of $\mathbb{R}^2$ not containing the boundary of any region;[4] see Fig. 5. The complexity of $\mathcal{A}(\Gamma)$, which we denote by $\kappa(\Gamma)$, is the total number of its vertices, edges, and faces. Since $\mathcal{A}(\Gamma)$ is a planar graph, this quantity is proportional to the number of edges. In the worst case $\kappa(\Gamma) = O(m^2)$ provided that any pair of boundaries intersect in $O(1)$ points, but in our cases it will be much smaller – closer to $O(m)$. Let $\mathcal{D}(\Gamma)$ be the planar dual graph of $\mathcal{A}(\Gamma)$ – its nodes (resp. edges, faces) are the faces (resp. edges, nodes) of $\mathcal{A}(\Gamma)$. We label each edge of $\mathcal{D}(\Gamma)$ with the region of Γ whose boundary contains the corresponding dual edge of $\mathcal{A}(\Gamma)$ [23, Page 44]. See [5] for details on arrangements.

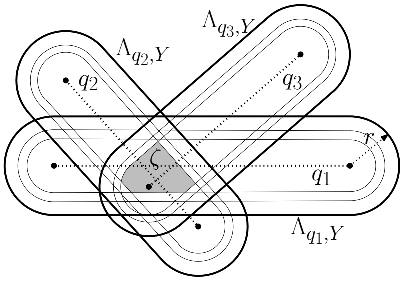


Fig. 5. The arrangement which corresponds to probabilistic attacks of 3 links q_1, q_2, and q_3, such that each has 3 superlevel sets. The shaded region ζ is an example of one of the faces of the arrangement.

Let $\alpha : \Gamma \to \mathbb{R}^+$ be a weight function. For a point $p \in \mathbb{R}^2$, we define its *depth* with respect to Γ and α to be

$$\Delta(\Gamma, \alpha, p) = \sum_{\{\gamma \in \Gamma | p \in \gamma\}} \alpha(\gamma).$$

If $\alpha(\gamma) = 1$ for all $\gamma \in \Gamma$, then $\Delta(\Gamma, p) := \Delta(\Gamma, \alpha, p)$ is the number of regions of Γ containing p. We set $\Delta(\Gamma, \alpha) = \max_{p \in \mathbb{R}^2} \Delta(\Gamma, \alpha, p)$ to be the maximum depth and denote the maximum number of regions containing a point by $\Delta(\Gamma)$.

Superlevel sets. Let $h : \mathbb{R}^2 \to \mathbb{R}$ be a bivariate function. For a value $t \in \mathbb{R}$, we define the *t-superlevel set* of h to be the closure of the set $h_{\geq t} = \{x \in \mathbb{R}^2 \mid h(x) \geq t\}$. If h is continuous, then $h(x) = t$ for all points on the boundary of the t-superlevel set. Given two parameters $\delta > 0$ and $s \in \mathbb{Z}^+$, we define

$$Y(\delta, s) = \{y_i := (1-\delta)^i \mid 0 \leq i \leq s\}.$$

[4] We assume that the boundaries of two regions are either disjoint or intersect transversally at a finite number of points.

Given $Y := Y(\delta, s)$ and a simple component q, let $\lambda_{q,i}$ be the y_i-superlevel set of f_q. Let $\Lambda_{q,Y} = \{\lambda_{q,i} \mid 0 \le i \le s\}$. Since we have assumed f_q to be a non-increasing function of distance from q, the two following properties hold.

1) For all $i \le s$, $\lambda_{q,i}$ is a simply connected region and $\lambda_{q,i} \subseteq \lambda_{q,i+1}$. Hence $\mathcal{A}(\Lambda_{q,Y})$ is a set of "nested" faces; see Fig. 2.
2) Let a, b be two points lying in the same face of $\mathcal{A}(\Lambda_{q,Y})$. If a, b lie in the outermost (unbounded) face, then $f_q(a), f_q(b) \le (1-\delta)^s$, otherwise $f_q(a) \ge (1-\delta) f_q(b)$.

A. Expected Damage for Simple Components

Let Q be a set of m weighted simple components—Q is a set of links or a set of nodes (amplifiers), let $w_q > 0$ be the weight of $q \in Q$, and let $\varepsilon > 0$ be a parameter. We describe two algorithms for computing a point $\tilde{p}$ such that $\Phi(Q, \tilde{p}) \ge (1-\varepsilon)\Phi(Q)$. We first describe our basic algorithm MaxExpectedDamageLocation, which is a Las Vegas algorithm to compute $\tilde{p}$. Then, we present a faster Monte Carlo algorithm, albeit with a slight probability of finding a point whose induced damage is less than $(1-\varepsilon)\Phi(Q)$. MaxExpectedDamageLocation is a building block in our other algorithms for more sophisticated scenarios.

MaxExpectedDamageLocation — A Las Vegas algorithm. The algorithm, whose running time is, in practice, $O((m/\varepsilon)\log^2(m/\varepsilon))$, has the following four main steps:

1) Superlevel sets generation for each component q, taking into account the approximation parameter ε.
2) Computation of the corresponding arrangement. This procedure is randomized with guaranteed expected running time. The arrangement induces a function $\tilde{f}$ approximating f, which is constant within each face.
3) Efficient computation of $\tilde{f}$ for each face.
4) MaxExpectedDamageLocation returns an arbitrary point in the face whose $\tilde{f}$ is maximal.

We turn now to the details of each step and prove the correctness of the algorithm. Without loss of generality, we assume that $\max_{p \in \mathbb{R}^2} f_q(p) = 1$ for all q and that $\max_{q \in Q} w_q = 1$. If necessary, we scale the weights and probability distribution functions so that this may be true.

We set $\delta = \varepsilon/4, s = \lceil \log_{1-\varepsilon}(\delta/m) \rceil$, and $Y = Y(\delta, s)$. Note that $s = O((m/\varepsilon)\log(m/\varepsilon))$. Set $\Lambda_q = \Lambda_{q,Y}$ for all $q \in Q$ and $\Lambda = \bigcup_{q \in Q} \Lambda_q$. We assume that the superlevel sets of different components intersect transversally, i.e., if two superlevel sets intersect, there always exists a region adjacent to the intersection points. We compute $\mathcal{A}(\Lambda)$ and its dual graph $\mathcal{D}(\Lambda)$. For each $q \in Q$, we define a new function $\tilde{f}_q : \mathbb{R}^2 \to \mathbb{R}$:

$$\tilde{f}_q(p) = \begin{cases} 0, & \text{if } p \notin \lambda_{q,s}, \\ (1-\delta)^i, & \text{if } i = \min\{j \mid p \in \lambda_{q,j}\}. \end{cases} \tag{4}$$

Set

$$\tilde{\Phi}(Q, p) = \sum_{q \in Q} w_q \tilde{f}_q(p). \tag{5}$$

Note that $\tilde{f}_q(p)$, and thus $\tilde{\Phi}(Q, p)$ is the same for all points p in a face $\zeta \in \mathcal{A}(\Lambda)$, and we use $\tilde{f}_q(\zeta)$ and $\tilde{\Phi}(Q, \zeta)$ to denote these values, respectively. Further, let ζ_1 and ζ_2 be two adjacent faces of $\mathcal{A}(\Lambda)$ sharing an edge $e \subseteq \lambda_{q,i}$. $\tilde{f}_{q'}(\zeta_1) = \tilde{f}_{q'}(\zeta_2)$ for all components $q' \neq q$. Thus, if we have computed $\tilde{\Phi}(Q, \zeta_1)$ then we can compute $\tilde{\Phi}(Q, \zeta_2)$ from $\tilde{\Phi}(Q, \zeta_1)$ by updating a single term in (5). By performing a depth-first search on $\mathcal{D}(\Lambda)$, we compute $\Phi_\zeta = \Phi(Q, \zeta)$ for each node ζ of $\mathcal{D}(\Lambda)$ (each face ζ of $\mathcal{A}(\Lambda)$) and return a point $\tilde{p}$ from a face $\tilde{\zeta}$ of $\mathcal{A}(\Lambda)$ that maximizes Φ_ζ, i.e., $\tilde{\zeta} = \arg\max_{\zeta \in \mathcal{D}(\Lambda)} \tilde{\Phi}_\zeta$. Since the boundary curves of superlevel sets of two components intersect transversally, we can prove that $\tilde{\Phi}(\tilde{p}) = \max_{p \in \mathbb{R}^2} \tilde{\Phi}(p)$. The correctness of the algorithm follows from the following two lemmas.

Lemma 1. *For any $q \in Q$ and for any point $p \in \mathbb{R}^2$,*

$$f_q(p) \ge \tilde{f}_q(p) \ge \begin{cases} f_q(p) - \delta/m, & \textit{if } p \notin \lambda_{q,s}, \\ (1-\delta) f_q(p), & \textit{otherwise.} \end{cases}$$

Proof: By construction of superlevel sets and the definition of $\tilde{f}_q$, $f_q(p) \ge \tilde{f}_q(p)$ for all $p \in \mathbb{R}^2$. If $p \notin \lambda_{q,s}$ then $f_q(p) < \delta/m$, and $\tilde{f}_q(p) = 0$. If $i \le s$ is the smallest index such that $p \in \lambda_{q,i}$, then $f_q(p) \le (1-\delta)^{i-1}$ and $f_q(p) = (1-\delta)^i$, implying that $\tilde{f}_q(p) \ge (1-\delta) f_q(p)$. ■

Lemma 2. $\Phi(Q, \tilde{p}) \ge (1-\varepsilon/2)\Phi(Q)$.

Proof: Let $p^* = \arg\max_{p \in \mathbb{R}^2} \Phi(Q, p)$. Let $Q_I = \{q \in Q \mid p^* \in \lambda_{q,s}\}$ and $Q_E = Q \setminus Q_I$. Then using Lemma 1,

$$\begin{aligned} \tilde{\Phi}(Q, p^*) &= \sum_{q \in Q_I} w_q \tilde{f}_q(p^*) + \sum_{q \in Q_E} w_q \tilde{f}_q(p^*) \\ &\ge \sum_{q \in Q_I} w_q (1-\delta) f_q(p^*) + \sum_{q \in Q_E} w_q (f_q(p^*) - \delta/m) \\ &\ge (1-\delta) \sum_{q \in Q_I} w_q f_q(p^*) - \sum_{q \in Q_E} w_q \delta/m \\ &\ge (1-\delta)\Phi(Q, p^*) - \delta \ge (1-\varepsilon/2)\Phi(Q), \end{aligned}$$

since $\Phi(Q, p^*) \ge 1$ by our normalization. The lemma now follows because $\tilde{\Phi}(Q, \tilde{p}) \ge \tilde{\Phi}(Q, p^*)$. ■

Finally, the running time of the algorithm is bounded by the time spent in computing $\mathcal{A}(\Lambda)$, plus $O(\kappa(\Lambda))$ to compute $\tilde{\Phi}_\zeta$ for all faces $\zeta \in \mathcal{A}(\Lambda)$. The former takes $O(|\Lambda| \log |\Lambda| + \kappa(\Lambda)) = O((m/\varepsilon)\log^2(m/\varepsilon) + \kappa(\Lambda))$ expected time (see [5]). Note that $\kappa(\Lambda)$ is $O(|\Lambda|)$ plus the number of pairs of superlevel sets in Λ whose boundaries intersect. Let $\kappa(Q, \delta)$ denote the number of such pairs in Λ for a given parameter δ. In the worst case $\kappa(Q, \delta) = O((m^2/\varepsilon^2)\log^2(m/\varepsilon))$ but in practice it is closer to $(m/\varepsilon)\log(m/\varepsilon)$. We conclude the following.

Theorem 1. *Let Q be a set of m simple components, and let f be a probability function defined for the components in Q, let $\varepsilon > 0$ be a parameter. Then a point $\tilde{p}$ can be computed in expected time $O((m/\varepsilon)\log^2(m/\varepsilon) + \kappa(Q, \varepsilon/4))$ such that $\Phi(Q, \tilde{p}) \ge (1-\varepsilon)\Phi(Q)$.*

Monte Carlo Algorithm. We now describe a faster Monte Carlo algorithm to compute such a point $\tilde{p}$ (the exact running time of the algorithm depends on the type of simple components considered, see Theorem 2). We do this by formulating the problems as computing a point of maximum depth in a set of weighted regions and adapting the algorithms of Agarwal *et al.* [4] or Aronov and Har-Peled [6]. Λ_q and Λ

are as above. We define a weight function $\beta : \Lambda \to \mathbb{R}^+$ as follows. For a superlevel set $\lambda_{q,i}$, we define

$$\beta(\lambda_{q,i}) = \begin{cases} w_q(1-\delta)^s & \text{if } i = s,\\ w_q\delta(1-\delta)^i & \text{otherwise.}\end{cases}$$

Lemma 3. *For any point $p \in \mathbb{R}^2$, $\Delta(\Lambda, p) = \tilde{\Phi}(Q,p)$.*

Proof: Fix a component q. If $p \notin \lambda_{q,s}$, then $\tilde{\Phi}(Q,p) = 0$ and p does not lie in any superlevel set of Λ_q. Let $i \leq s$ be the smallest index such that $p \in \lambda_{q,i}$. If $i = s$, $\Delta(\Lambda_q, p) = w_q(1-\delta)^s$ and $\tilde{f}_q(p) = (1-\delta)^s$. If $i < s$, then

$$\begin{aligned}\Delta(\Lambda_q,p) &= \sum_{j\geq i} w_q\beta(\lambda_q, i) = w_q\sum_{j=i}^{s-1}\delta(1-\delta)^j + w_q(1-\delta)^s\\ &= w_q(1-\delta)^i(1-(1-\delta)^{s-i}) + w_q(1-\delta)^s\\ &= w_q(1-\delta)^i = w_q\tilde{f}_q(p).\end{aligned}$$

Hence $\Delta(\Lambda,p) = \sum_{q\in Q}\Delta(\Lambda_q,p) = \tilde{\Phi}(Q,p)$. ■

The problem of computing $\tilde{p}$ thus reduces to computing the point of the maximum depth in Λ. We reduce this problem to computing the deepest point in a multiset of unweighted regions. Set $|\Lambda| = n$. For each component $\lambda_{q,i} \in \Lambda$, we set

$$\tilde{\beta}(\lambda_{q,i}) = \left\lfloor \frac{2n}{\delta}\beta(\lambda_{q,i})\right\rfloor$$

and keep only those superlevel sets for which $\tilde{\beta}(\lambda_{q,i}) \geq 1$. Next we make $\tilde{\beta}(\lambda_{q,i})$ copies of $\lambda_{q,i}$ and let $\tilde{\Lambda}$ be the resulting multiset of superlevel sets. By construction, $|\tilde{\Lambda}| \leq \sum_{i=1}^{n}\sum_{j\geq 1}(2n/\delta)(1-\delta)^j = O\left((n^2/\delta^2)\log(n/\delta)\right)$. We do not compute the set $\tilde{\Lambda}$ explicitly; we will generate various subsets of $\tilde{\Lambda}$ as needed. The following lemma is the crux of our algorithm.

Lemma 4. $(\delta/2n)\Delta(\tilde{\Lambda}) \geq (1-\delta/2)\Delta(\Lambda,\beta)$.

Proof: Let $\tilde{p}$ be the point of maximum depth with respect to Λ, β. If $\tilde{p} \in \lambda_{q,i}$ then $\tilde{p}$ lies in all $\lfloor(2n/\delta)\beta(\lambda_{q,i})\rfloor$ copies of $\lambda_{q,i}$ in $\tilde{\Lambda}$. Hence,

$$\begin{aligned}\Delta(\tilde{\Lambda},\tilde{p}) &\geq \sum_{\lambda_{q,i}|\tilde{p}\in\lambda_{q,i}}\left(\frac{2n}{\delta}\beta(\lambda_{q,i}) - 1\right)\\ &= \frac{2n}{\delta}\Delta(\Lambda,\beta,\tilde{p}) - n = \frac{2n}{\delta}(1-\delta/2)\Delta(\Lambda,\beta,\tilde{p}),\end{aligned}$$

since $\Delta(\Lambda,\beta) \geq 1$. Hence, the lemma holds. ■

We now describe an algorithm that computes a point $\bar{p}$ such that $\Delta(\tilde{\Lambda},\bar{p}) \geq (1-\delta/2)\Delta(\tilde{\Lambda})$. The following lemma is a slightly adapted form of the lemma in [6] (cf. Corollary 3.2).

Lemma 5. *Let $\Delta = \Delta(\tilde{\Lambda})$, $\tilde{n} = |\tilde{\Lambda}|$, $0 < \tilde{\delta} < 1/2$ be fixed, $r \geq \Delta/4$ be an integer, and $\tilde{R} \subseteq \tilde{\Lambda}$ be a multiset formed by picking each region $\tilde{\Lambda}$ with probability $\psi = \psi(\tilde{\delta}, r) := \min\left(c_1(\log\tilde{n}/r\tilde{\delta}^2), 1\right)$, independently, where c_1 is an appropriate constant. Then:*

(i) *If $\Delta(\tilde{R}) \geq 2r\psi$, then with high probability $\Delta \geq 3r/2$.*

(ii) *If $\Delta(\tilde{R}) \leq (1-\tilde{\delta})r\psi$, then with high probability $\Delta \leq r$.*

(iii) *For all $p \in \mathbb{R}^2$, such that $\Delta(\tilde{R},p) \geq (1-\tilde{\delta})r\psi$, $(1-\tilde{\delta})\Delta(\tilde{\Lambda},p) \leq \frac{\Delta(\tilde{R},p)}{\psi} \leq (1+\tilde{\delta})\Delta(\tilde{\Lambda},p)$, with high probability.*

In view of Lemma 5, the point $\bar{p}$ can be computed by doing an exponential search, as described in [6]. There are two non-trivial steps: (i) Choosing the multiset $\tilde{R}$. (ii) A depth threshold procedure that determines whether $\Delta(\tilde{R}) \geq (1-\tilde{\delta})r\psi$. If so, then return a point p such $\Delta(\tilde{R},p) \geq (1-\tilde{\delta})r\psi$. Recall that we do not compute the set $\tilde{\Lambda}$ explicitly. We observe that the number of copies of $\lambda_{q,i}$ chosen in $\tilde{R}$ follows a *binomial distribution* $\mathbb{B}(\tilde{\beta}_{q,i},\psi)$ with parameters $\tilde{\beta}_{q,i}$ and ψ. So we draw a value $\nu_{q,i} \sim \mathbb{B}(\tilde{\beta}_{q,i},\psi)$ in $O(\log\tilde{\beta}_{q,i}) = O(\log m)$ time and associate $\nu_{q,i}$ as the weight $\nu(\lambda_{q,i})$ of $\lambda_{q,i}$. If $\nu_{q,i} = 0$, we ignore $\lambda_{q,i}$. Let $R \subseteq \Lambda$ be the resulting subset, and let $\tilde{R}$ be the resulting multiset. Then for any point $p \in \mathbb{R}^2$, $\Delta(\tilde{R},p) = \Delta(R,\nu,p)$. We can use the procedure described in [1], [6] to check whether $\Delta(R,\nu,p) \geq (1-\tilde{\delta})r\psi$. The expected running time of these procedures is $O(|R|\log|R| + \rho)$ where ρ is the number of vertices in $\mathcal{A}(R)$ whose depth with respect to R,ν (i.e., depth w.r.t. $\tilde{R}$) is at most $r\psi$.

Since $\nu(\lambda_{q,i}) \geq 1$ for all superlevel sets in R, ρ is bounded by the number of vertices whose unweighted depth is at most ρ. Using the argument in Clarkson and Shor [14] (see also [48]), it can be shown that $\rho = O(\sigma(Q)\log^2(n)/\varepsilon^4)$, where $\sigma(Q)$ is the maximum number of vertices on the boundary of the union of a subset of superlevel sets in Λ, If Q is a set of nodes then $\sigma(Q) = m$, but if Q is a set of links, then $\sigma(l)$ can be $\Omega(m^2)$ in the worst case even though it is $O(m)$ in practice. Since the decision procedure is invoked $\log m$ times, the overall running time of this procedure is $O(\sigma(Q)\log^4(m/\varepsilon)/\varepsilon^4)$. Lemmas 2, 3, 4 imply that $\Phi(Q,\bar{p}) \geq (1-\varepsilon)\Phi(Q)$. Indeed,

$$\begin{aligned}\frac{\delta}{2n}\Delta(\tilde{\Lambda},\bar{p}) &\geq \frac{\delta}{2n}(1-\delta/2)\Delta(\tilde{\Lambda}) \geq (1-\delta/2)\Delta(\Lambda,\beta)\\ &= (1-\delta/2)^2\Phi(Q) \geq (1-\delta/2)^2(1-2\delta)\Phi(Q)\\ &\geq (1-3\delta)\Phi(Q) \geq (1-\varepsilon)\Phi(Q).\end{aligned}$$

Hence, we obtain the following:

Theorem 2. *Let Q be a set of m simple components, f a probability distribution function, and $\varepsilon > 0$ be a parameter. A point $\tilde{p} \in \mathbb{R}^2$ can be computed in $O(\sigma(Q)\log^3(m/\varepsilon)/\varepsilon^4)$ expected time such that with high probability $\Phi(Q,\tilde{p}) \geq (1-\varepsilon)\Phi(Q)$, where $\sigma(Q)$ is the parameter as defined above and its value lies between m and m^2.*

B. Expected Damage for Compound Components

Let $\Pi = \{\pi_1,\ldots,\pi_m\}$ be a set of m compound components and let $0 < \varepsilon < 1$ be a parameter. We wish to compute a point $\tilde{p} \in \mathbb{R}^2$ such that $\Phi(\Pi,p) \geq (1-\varepsilon)\Phi(\Pi)$. We can use the algorithm described for simple components but the difficulty is that a superlevel set of f_π, is not a simply connected region of constant size – its boundary may be disconnected and may have too many edges; see Fig. 3. So computing a superlevel set of π is expensive. Hence, we use a slightly different approach.

Let Q be the set of simple components in the compound components of Π. Set $\sum_{\pi\in\Pi}|\pi| = n$. We say that a simple component q is *affected* by an attack at a location p if $f_q(p) \geq \varepsilon/(4mn)$, otherwise we say that an attack at location p has no affect on of q. For a compound component $\pi \in \Pi$, σ_π be the maximum number of simple components in π that can be affected by an attack at some location, and let

$\sigma_\Pi = \max_{\pi\in\Pi}\sigma_\pi$. In practice $\sigma := \sigma_\Pi$ is a constant, though it may be as large as $|Q|$ in the worst case.

We set $\delta = \varepsilon/4\sigma$, $s = \log_{1-\delta}(\varepsilon/4mn) = O((\sigma/\varepsilon)\log(n/\varepsilon))$. For each $q \in Q$, let $\Lambda_q = \Lambda_{q,Y}$, and $Y = Y(\delta, s)$ and let $\Lambda = \bigcup_{q\in Q}\Lambda_q$, $|\Lambda| = O((\sigma n/\varepsilon)\log(n/\varepsilon))$. Next, let $\tilde{f}_q$ be the same as in (4), and we now define: $\tilde{f}_\pi(p) = 1 - \prod_{q\in\pi}(1-\tilde{f}_q(p))$, $\tilde{\Phi}(\Pi, p) = \sum_{\pi\in\Pi} w_\pi \tilde{f}_\pi(p)$.

Note that for any $q \in Q$ if a point $p \notin \lambda_{q,s}$ then q is not affected by an attack p. We compute $\mathcal{A}(\Lambda)$, compute $\tilde{\Phi}(\Pi, \zeta)$ for each face ζ of $\mathcal{A}(\Lambda)$, and return a point $\tilde{p}$ from a face $\tilde{\zeta}$ that maximizes the value of $\tilde{\Phi}(\Pi, \zeta)$. The total time taken by this algorithm is $O(|\Lambda|\log|\Lambda| + \kappa(\Lambda))$. The correctness of the algorithm follows from the following two lemmas.

Lemma 6. *Let $X_i \in (0,1)$ for $1 \le i \le k$, let $0 < \delta < 1/k$ be a parameter, and for $1 \le i \le k$ let $\tilde{X}_i$ be a value such that $X_i \ge \tilde{X}_i \ge (1-\delta)X_i$. If $g(X_1, \ldots, X_k) = 1 - \prod_{i=1}^{k}(1-X_i)$, then $g(X_1,\ldots,X_k) \ge g(\tilde{X}_1,\ldots,\tilde{X}_k) \ge (1-k\delta)g(X_1,\ldots,X_k)$.*

Proof: Since $\tilde{X}_i \le X_i$, the first inequality is true. For $j \le k$, let $\binom{X}{j}$ denote the family of subsets of $X_1, \ldots, X_k$ of size j. Then, $g(X_1,\ldots,X_k) = \sum_{j\ge1}(-1)^{j+1}\sum_{R\in\binom{X}{j}}\prod_{X_i\in R}X_i$. The value of $g(\tilde{X}_1,\ldots,\tilde{X}_k)$ is minimum when $\tilde{X}_i = (1-\delta)X_i$. Therefore,

$$\begin{aligned} g(\tilde{X}_1,\ldots,\tilde{X}_k) &\ge \sum_{j\ge1}(-1)^{j+1}\sum_{R\in\binom{X}{j}}(1-\delta)^j\prod_{X_i\in R}X_i \\ &\ge (1-\delta)^k\sum_{j\ge1}(-1)^{j+1}\sum_{R\in\binom{X}{j}}\prod_{X_i\in R}X_i \\ &\ge (1-k\delta)g(X_1,\ldots,X_k). \qquad\blacksquare \end{aligned}$$

We now prove the main lemma.

Lemma 7. *For any $\pi \in \Pi$ and for any $p \in \mathbb{R}^2$, $f_\pi(p) \ge \tilde{f}_\pi(p) \ge (1-\varepsilon/2)f_\pi(p) - \frac{\varepsilon}{2m}$.*

Proof: It suffices to prove the second inequality. Fix a point $p \in \mathbb{R}^2$. Let $\Pi_A \subseteq \Pi$ be the set of simple components affected by p, and let $\Pi_{NA} \subseteq \Pi$ be the set of remaining simple components; set $t = |\Pi_{NA}|$. The $\tilde{f}_q(p) \ge 0$ and $f_q(p) \le \varepsilon/4mn$ for all components $q \in \Pi_{NA}$. Therefore, by Lemma 6,

$$\begin{aligned} \tilde{f}_\pi(p) &\ge 1 - \prod_{q\in\Pi_A}(1-\tilde{f}_q(p)) \\ &\ge (1-\varepsilon/4)\left[1 - \prod_{q\in\Pi_A}(1-f_q(p))\right] \\ &= \frac{1-\varepsilon/4}{(1-\varepsilon/4mn)^t}\Bigg[(1-\varepsilon/4mn)^t - \\ &\qquad (1-\varepsilon/4mn)^t\prod_{q\in\Pi_A}(1-f_q(p))\Bigg] \\ &\ge (1-\varepsilon/2)\left[(1-\varepsilon/4m) - \prod_{q\in\pi}(1-f_q(p))\right] \\ &\ge (1-\varepsilon/2)f_\pi(p) - \frac{\varepsilon}{4m}. \qquad\blacksquare \end{aligned}$$

Using the above lemma and following the proof of Lemma 2, we obtain the following.

Corollary 1. $\Phi(\Pi) \ge \tilde{\Phi}(\Pi, \tilde{p}) \ge (1-\varepsilon)\Phi(\Pi)$.

Putting everything together, we obtain the following:

Theorem 3. *Let Π be a set of m compound components, let Q be the set of simple components in them, and let $n = \sum_{\pi\in\Pi}|\pi|$. Let f be a probability distribution function, and let $0 < \varepsilon < 1$ be a parameter. A point $\tilde{p}$ such that $\Phi(\Pi, \tilde{p}) \ge (1-\varepsilon)\Phi(\Pi)$ can be computed in expected time $O(\frac{\sigma n}{\varepsilon}\log^2(n/\varepsilon) + \kappa(Q, \varepsilon/4\sigma))$ time, where σ is the maximum number of simple components of a component in Π that are affected by an attack and $\kappa(Q, \varepsilon/4\sigma)$ is the same as defined above.*

We note that in the worst case $\sigma = n$ and $\kappa(Q, \varepsilon/4\sigma) = O(\sigma^2 n^2 \log^2(n/\varepsilon)/\varepsilon^2) = O((n^4/\varepsilon^2)\log^2(n/\varepsilon))$, but in practice σ is a small constant and $\kappa(Q, \varepsilon/4\sigma) = O(|\Lambda|) = O((n/\varepsilon)\log(n/\varepsilon))$. Furthermore, we can also use the Monte Carlo algorithm by sampling components in Π. However, it is hard to prove an improved bound on its running time because the complexity of superlevel sets can be large.

C. Average Two-Terminal Reliability

Let Q be a set of m simple components and let $0 < \varepsilon < 1/2$ be a parameter. We describe an algorithm for computing a point $\tilde{p}$ such that $\chi(Q, \tilde{p}) \le (1+\varepsilon)\chi(Q)$ (as defined in Section III). Our algorithm follows the same paradigm as in Sections IV-A and IV-B: (i) compute a set of superlevel sets, (ii) compute $\chi(Q, p_\varphi)$ for a point p_φ in each face φ of their arrangement and (iii) return a point p_{φ^*} with lowest χ.

We make two assumptions on the effects of the attacks:

- **A1:** We assume a *local attack* whose range is limited to r which is small compared to the network's environment size. Formally, we assume that an attack at p can only affect a small number of components, i.e., if, for an attack at p, $Q_p = \{q \in Q \mid d(p,q) \le r\}$, then $|Q_p| \le k$. We assume k to be a constant. In the context of this section, we call k the maximum depth (note that this is different from the weighted depth defined in Section IV).
- **A2:** An attack on the network cannot destroy any component with very low or very high probability: If a component has a probability smaller than ε to fail (or survive), we assume that this is indeed the case. More formally, for an attack at p affecting a component q, we assume that $f_q(p)$ is either 0 or 1 or lies in the interval $(\varepsilon, 1-\varepsilon)$.

For each component q, we construct the $y_{\min}$-superlevel set of f_q where $y_{\min} = \varepsilon$ and denote this by $\lambda_{q,\min}$. Note that, by A2, for every location outside $\lambda_{q,\min}$, f_q takes the value 0. Let $\Lambda_{\min} = \{\lambda_{q,\min} \mid \forall q \in Q\}$ and $\mathcal{A}_{\min} = \mathcal{A}(\Lambda_{\min})$ denote the arrangement of $\Lambda_{\min}$. We call $\mathcal{A}_{\min}$ the *coarse arrangement*. In every face φ of $\mathcal{A}_{\min}$, the set of components in Q which have positive probability of failure stays the same. We denote this set by $Q(\varphi)$. Note that $|Q(\varphi)| \le k$.

At a higher level, the algorithm traverses the faces of $\mathcal{A}_{\min}$ so that, at each face φ, we may maintain the connected components of $Q\backslash Q(\varphi)$. For more details on how this may be done, we refer the reader [23, Chapter V] in which a procedure for performing this traversal efficiently is described.

At each face, we construct a *fine arrangement* in a manner similar to Sections IV-A and IV-B. We set $\delta = \varepsilon/8k$, $s =$

$\log_{1-\delta}(\varepsilon/(1-\varepsilon)) = O((k/\varepsilon)\log(1/\varepsilon)$ and compute $Y := Y(\delta, s)$. For a component q, let $\lambda_{q,i}$ denote the y_i-superlevel set of f_q and let $\lambda'_{q,i}$ denote the y_i-superlevel set of $1-f_q$. Let $\Lambda_q = \{\lambda_{q,i} \mid 0 \le i \le s\} \cup \{\lambda'_{q,i} \mid 0 \le i \le s\}$ and let $\Lambda(\varphi) = \cup_{q \in Q(\varphi)} \Lambda_q$. By the properties of superlevel sets, for all $i \le s$, $\lambda_{q,i}$ and $\lambda'_{q,i}$ are simply connected regions and $\lambda_{q,i} \subseteq \lambda_{q,i+1}$ (similarly, $\lambda'_{q,i} \subseteq \lambda'_{q,i-1}$). Thus, the arrangement $\mathcal{A}(\Lambda(\varphi))$ is a set of "nested" faces. In each case, we choose $\delta = \varepsilon/4k$ and s such that $\varepsilon(1+\delta)^s \ge (1-\varepsilon)$. Therefore, $s = O(\frac{k}{\varepsilon}\log\frac{1}{\varepsilon})$. Let ζ be a face of a fine arrangement $\mathcal{A}(\Lambda(\varphi))$ contained inside a face φ of the coarse arrangement $\mathcal{A}_{\min}$.

Recall that at face φ of $\mathcal{A}_{\min}$, we maintain the set of connected components of $Q \setminus Q(\varphi)$. At a face φ of $\mathcal{A}_{\min}$, the algorithm traverses the faces of $\mathcal{A}(\Lambda(\varphi))$ inside φ by performing a depth-first search on the dual graph $D(\Lambda(\varphi))$ of $\mathcal{A}(\Lambda(\varphi))$ similar to the algorithm in Section IV-A. At each face ζ, we compute the probabilities of all possible failure scenarios of components $Q(\varphi)$ (since $|Q(\varphi)| \le k$, there are possible 2^k such scenarios corresponding to each subset of $Q(\varphi)$ failing). For each scenario, we insert the components which are not failed into the set of connected components of $Q \setminus Q(\varphi)$ and compute the number of pairs of nodes connected. A weighted sum over all scenarios with the weights corresponding to the probabilities gives the value of $\chi(\varphi)$. Finally, the algorithm reports the minimum over all faces. We refer the reader to [23, Chapter V] for the complete details of this procedure.

The correctness of the algorithm follows from the following lemma. Consider a face ζ of $\mathcal{A}(\Lambda(\varphi))$ contained in a face φ of $\mathcal{A}_{\min}$ and a single scenario of failure of components in $Q(\varphi)$ where only the components in a set $Q_f \subset Q(\varphi)$ fail. Further, we denote by $\chi_{Q_f}(p)$, the probability of this scenario taking place when the attack is at a point $p \in \zeta$.

Lemma 8. *For two points p_1 and p_2 in the same face ζ of $\mathcal{A}(\Lambda(\varphi))$ and a specific subset $Q_f \subseteq Q(\varphi)$ failing, $\chi_{Q_f}(p_1) \ge \chi_{Q_f}(p_2)$, then $\chi_{Q_f}(p_1) \le (1+\varepsilon)\chi_{Q_f}(p_2)$.*

Proof: For an attack at a point $p \in \varphi$, we have

$$\chi_{Q_f}(p) = \prod_{q \in Q_f} f_q(p) \cdot \prod_{q \in Q(\varphi) \setminus Q_f} (1 - f_q(p))$$

Since for each $q \in Q_f$, $f_q(p_2) \ge (1-\delta) f_q(p_1)$ and similarly, for each $q \in Q(\varphi) \setminus Q_f$, $1 - f_q(p_2) \ge (1-\delta) f_q(p_1)$ where $\delta = \varepsilon/8k$, we have:

$$\chi_{Q_f}(p_2) \ge \left(1 + \frac{\varepsilon}{4k}\right)^k \chi_{Qf}(p_1),$$

since $1/(1-(\varepsilon/8k)) \le 1 + (\varepsilon/4k)$ for $0 < \varepsilon < 1/2$. The proof follows since $(1+\frac{\varepsilon}{4k})^k \le (1+\frac{\varepsilon}{4}(e-1)) \le (1+\varepsilon)$. ∎

Summing over all scenarios, clearly, the algorithm provides a $(1+\varepsilon)-$approximation of the optimal value.

We now analyze the running time of the algorithm. The arrangement $\mathcal{A}_{\min}$ may be computed in time $O(m\log m + |\mathcal{A}_{\min}|)$. $|\mathcal{A}_{\min|}| = km$ since the maximum depth is k and $\Lambda_{\min}$ is a set of pseudo-disks (see [14], [47]). For each face φ of $\mathcal{A}_{\min}$, the time spent in computing χ is exponential in k (since we examine 2^k failure scenarios of $Q(\varphi)$) and independent of m (since $|\Lambda(\varphi)|$ is independent of m). The traversal of the faces of $\mathcal{A}_{\min}$ may be accomplished in time $O(km\log^2 m + km\log k)$ steps for each of which we need to traverse the fine arrangement (see [23, Chapter V] for full details). Thus, the total time for the algorithm is $O(c_k m(\log^2 m + \log k))$ where c_k is exponential in k and independent of m.

Theorem 4. *Under assumptions A1 and A2, given a set Q of m simple components, a point $\tilde{p}$ such that $\chi(Q, \tilde{p}) \le (1+\varepsilon)\chi(Q, p^*)$, where p^* is the location that minimizes χ, can be computed in $O(c_k m(\log^2 m + \log k))$ time. Here c_k is independent of m but exponential in the maximum depth k.*

V. Assessing Vulnerability to Multiple Simultaneous Attacks

We now consider scenarios in which k attacks may happen simultaneously. Our goal is therefore to identify the set P of k locations, for which $\Phi(Q, P)$ is maximized over all possible choices of k locations. In general, finding this set P is NP-hard, since maximizing the value of Φ is a generalization of the well-known *maximum set cover problem* [27]. Nevertheless, we show that the function Φ satisfies two key properties *monotonicity* and *submodularity*, which are used to develop an approximation algorithm. Again, as before, this approximation algorithm has a tunable parameter ε which provides a tradeoff between the approximation factor and running time.

At a high level, the greedy algorithm works in k iterations. At each iteration, we choose a location for an attack. Let $P_i = \{p_1, p_2, \ldots, p_i\}$ be the set of locations chosen after i iterations. At iteration $i+1$, we pick the location that has the highest impact in terms of expected component damage given that we have already chosen P_1. In order to quantify this impact, we define the notion of *revenue* of a location p given P_i, which is denoted by $\mathrm{Rev}(p, P_i)$ and defined as follows:

$$\mathrm{Rev}(p, P_i) = \Phi(Q, P_i \cup \{p\}) - \Phi(Q, P_i).$$

A perfect greedy algorithm would pick a point $p^*_{i+1} \notin P_i$ which maximizes the revenue $\mathrm{Rev}(p, P_i)$ over all points $p \in \mathbb{R}^2$. However, implementing the greedy algorithm exactly may be possible for certain functions $f_q(\cdot)$ (e.g., square of the Euclidean distance), but in general it might be difficult. Thus, our approximate greedy algorithm finds a location $\hat{p}_{i+1}$ such that $\mathrm{Rev}(\hat{p}_{i+1}, P_i) \ge (1-\varepsilon)\mathrm{Rev}(p^*_{i+1}, P_i)$. Notice that $\mathrm{Rev}(p, P_i) = \sum_{q \in Q} \mu(q, P_i) f_q(p)$, where $\mu(q, P_i) = w'_q \prod_{p_i \in P_i}(1 - f_q(p_i))$. Thus, the approximate greedy procedure may be implemented using the algorithms from in Section IV after modifying the weights of the components to $\mu(q, P_i)$ (instead of w'_q).

Let P^* be the set of k locations which maximizes $\Phi(Q, P)$ over all possible P. We now show that Φ satisfies the key properties: *monotonicity* and *submodularity*. These two properties immediately imply that a perfect greedy algorithm achieves a $(1-1/e)$–approximation [36]. Since our algorithm is only approximately greedy, this results in an overall approximation factor of $(1 - \frac{1}{e^{1-\varepsilon}})$ [24], for any $0 < \varepsilon < 1$.

Monotonicity intuitively means that the expected damage only increases with number of attacks. Formally, $\Phi(Q, \cdot)$ is monotonically non-decreasing, i.e., $\Phi(Q, P_1) \le \Phi(Q, P_2)$, for any set $P_2 \supseteq P_1$ (this stems from the fact that $\mu(q, P_2) \le$

$\mu(q, P_1)$, for any $q \in Q$). $\Phi(Q, \cdot)$ also exhibits the "law of diminishing returns" property or *submodularity*: for a given attack p and two sets of attacks P_1 and P_2 such that $P_2 \supseteq P_1$, the revenue of p is lower with respect to P_2 than with respect to P_1. The following lemma captures this property.

Lemma 9. *$\Phi(Q, \cdot)$ is a submodular function. Namely, for any two set of points P_1 and P_2, such that $P_2 \supseteq P_1$, and any point $p \in \mathbb{R}^2$, $\Phi(Q, P_1 \cup \{p\}) - \Phi(Q, P_1) \geq \Phi(Q, P_2 \cup \{p\}) - \Phi(Q, P_2)$, i.e., $\mathrm{Rev}(p, P_1) \geq \mathrm{Rev}(p, P_2)$.*

Proof: If $p \in P_2$, $\Phi(Q, P_2 \cup \{p\}) - \Phi(Q, P_2) = 0$ and the proof is trivial. If $p \notin P_2$, $\mathrm{Rev}(p, P_2) = \sum_{q \in Q} \mu(q, P_2) f_q(p)$ and $\mathrm{Rev}(p, P_1) = \sum_{q \in Q} \mu(q, P_1) f_q(p)$. Since $\mu(q, P_2) \leq \mu(q, P_1)$ for any $q \in Q$, the claim follows. ■

It is important to note that our proof holds for both types of components (simple and compound), and hence, the greedy algorithm works for both cases. Clearly, our algorithm takes $O(kg(Q))$ where $g(Q)$ is the time required to perform each step, i.e., the running time of MAXEXPECTEDDAMAGELOCATION from Sections IV-A and IV-B. Thus, in practice, for a set Q of m simple components, the running time would be $O(k(m/\varepsilon)\log^2(m/\varepsilon))$ and for a set Q of compound components such that $n = \sum_{\pi \in Q} |\pi|$, the runnning time would be $O(k(n/\varepsilon)\log^2(n/\varepsilon))$.

Theorem 5. *Let Q be a set of m simple or compound components, let f be a probability function defined for the simple components in Q and let $0 < \varepsilon < 1$ be a parameter. A set of k points $\tilde{P}$ such that $\Phi(Q, \tilde{P}) \geq (1 - (1/e^{1-\varepsilon}))\Phi(Q, k)$ can be found in time $O(kg(Q))$ where $g(Q)$ is the time required for finding a single location maximizing $\Phi(Q)$.*

VI. NETWORKS WITH A PROTECTION PLAN

In networks with a protection plan in place at time of deployment, the determination of paths (both primary and backup) during design-time often takes geographical correlation into account. The primary and backup lightpaths tend to be fiber-disjoint or even to be part of different Shared Risk Link Groups (SRLGs). For example, the fibers should not be close physically. Thus, it is likely that a reasonable protection plan will cope with a single attack. In this section, we evaluate the resilience of a protection plan to *two simultaneous attacks*.

Formally, we are given a set Π of pairs of lightpaths (π_i, π'_i), where π_i and π'_i are the primary and backup paths. Let T_i and t_i be, respectively, the high-priority and low-priority traffic on these lightpaths (for 1+1 protection, t_i is always 0). Thus, one loses t_i when either π_i or π'_i fails, or $T_i + t_i$ if both fail at once. We may consider three possible events at which there is a loss of traffic: (i) π_i fails and π'_i does not fail, denoted by E_1, (ii) π_i does not fail and π'_i fails, denoted by E_2, and (iii) both π_i and π'_i fail, denoted by E_3. Given two attack locations p_1 and p_2, the probabilities of the three events are as follows:

$$\begin{aligned}
Pr(E_1) &= g_{\pi'_i}(p_1)g_{\pi'_i}(p_2)(f_{\pi_i}(p_1) + f_{\pi_i}(p_2)g_{\pi_i}(p_1)) \\
Pr(E_2) &= g_{\pi_i}(p_1)g_{\pi_i}(p_2)(f_{\pi'_i}(p_1) + f_{\pi'_i}(p_2)g_{\pi'_i}(p_1)) \\
Pr(E_3) &= f_{\pi_i}(p_1)f_{\pi'_i}(p_1)g_{\pi'_i}(p_2) + f_{\pi_i}(p_2)f_{\pi'_i}(p_2)g_{\pi'_i}(p_1) \\
&\quad + f_{\pi_i}(p_1)f_{\pi'_i}(p_2)g_{\pi_i}(p_2) + f_{\pi_i}(p_2)f_{\pi'_i}(p_1)g_{\pi_i}(p_1) \\
&\quad + f_{\pi_i}(p_1)f_{\pi'_i}(p_1)f_{\pi_i}(p_2)f_{\pi'_i}(p_2)
\end{aligned}$$

where, $g_\pi(p)$ denotes $1 - f_\pi(p)$. Hence, the *expected loss on the i^{th} pair* is given by:

$$\begin{aligned}
\Phi_i(\{p_1, p_2\}) = t_i(Pr(E_1) + Pr(E_2) + Pr(E_3)) \\
+(t_i + T_i)Pr(E_3) \qquad (6)
\end{aligned}$$

For the entire network, we get $\Phi(\Pi, \{p_1, p_2\}) = \sum_i \Phi_i(\{p_1, p_2\})$. We next show how to find locations $\{\tilde{p_1}, \tilde{p_2}\}$ such that $\Phi(\Pi, \{\tilde{p_1}, \tilde{p_2}\})$ approximates $\Phi(\Pi, 2)$, the maximum expected loss over all pairs of locations. Notice that one can also measure the *worst-case vulnerability* of the protection plan by the value of $\Phi(\Pi, 2)$ and use this value to compare the resilience of alternative plans.

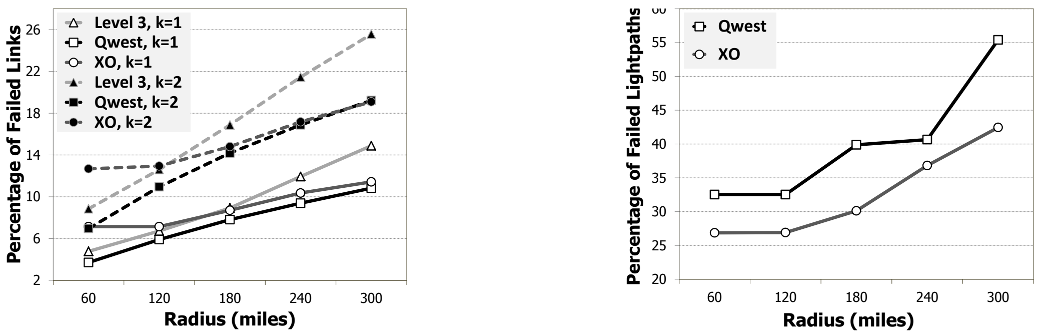
The algorithm proceeds in a manner similar to MAXEXPECTEDDAMAGELOCATION in Section IV. First, we scale the values of t_i and T_i for every pair (π_i, π'_i) such that $\max_i(T_i + t_i) = 1$. Next, similar to IV-B, we choose $\delta = \varepsilon/(c_1\sigma)$ where σ is the maximum number of simple components in any path (primary or backup) and c_1 is a constant whose choice is described later. We also choose s such that $(1 - \delta)^s \leq (\varepsilon/2n)$, where n is the sum of the lengths of all paths. Note that $s = O((\sigma/\varepsilon)\log(n/\varepsilon))$. With these values, we compute $Y := Y(\delta, s)$ and compute the arrangement Λ of superlevel sets of both f and $g = 1 - f$ for all simple components based on Y (similar to Section IV-C).

The approximation factor of both f and g for a single path follows similar to Lemma 7. Now, we compute the value $\Phi(\Pi, \{p_1, p_2\})$ for every pair of faces of Λ where p_1 and p_2 may be located and pick the pair which maximizes Φ, say $\{\tilde{p_1}, \tilde{p_2}\}$. Since there is at most a multiplication of four terms in Eq. (6), we choose the constant c_1 needed for determining s in such a manner that $c_1\varepsilon \leq 1 - \sqrt[4]{1 - \varepsilon}$. With this choice, following the proof of Lemma 2, we may show that the $\Phi(\Pi, \{\tilde{p}_1, \tilde{p}_2\}) \geq (1 - \varepsilon)\Phi(\Pi, 2)$. The running time of the algorithm is quadratic in the size of the arrangement.

Theorem 6. *Let Π be a set of lightpath pairs designating the protected paths, let Q be the constituent simple components and let $n = \sigma_{(\pi,\pi')\in\Pi}|\pi|+|\pi'|$. Let f be a probability function defined for the simple components in Q and let $0 < \varepsilon < 1$ be a parameter. A set of 2 points $\{\tilde{p}_1, \tilde{p}_2\}$ such that $\Phi(\{\tilde{p}_1, \tilde{p}_2\}) \geq (1 - \varepsilon)\Phi(\Pi, 2)$ can be found in time $O(((\sigma)/\varepsilon)\log^2(n/\varepsilon) + (\kappa(Q, \varepsilon/(c_1\sigma)))^2)$ where σ is the maximum number of simple components in any lightpath and κ is the complexity of the arrangement of superlevel sets of Q.*

VII. NETWORKS WITH RESTORATION ALGORITHMS

In a network with *dynamic restoration capabilities*, where traffic may be re-routed dynamically based on failed components, the optimal quality of restoration (in terms of post-attack traffic carried by the network between predetermined source nodes and destination nodes) is the *maximum flow* of the residual network. Therefore, finding the most vulnerable location in such a setting is equivalent to finding the location whose corresponding attack minimizes the *expected maximum flow*. However, under a probabilistic setting, finding the expected maximum flow of a graph is #P-complete. This is true even if all links have unit weight (that is, a connectivity problem), and even if the graphs are planar. It is important



(a) *Links*

(b) *Lightpaths*

Fig. 6. Variation of Φ, normalized by the sum over the entire network, with the attack radius for a *linear* failure probability function.

to note that although one is not directly required to compute the exact *value* of the expected maximum flow in order to find the most vulnerable location, and, in some cases, one can compare the effects of two locations without such computation (e.g., when the failure probability of one location dominates the other), in the general case, such computation is necessary (e.g., two locations affecting disjoint sets of links and there is no third location that can be used for comparison). Thus, we obtain the following result. whose complete proof appears in Appendix A.

Theorem 7. *Computing the most vulnerable location in term of expected maximum flow is #P-complete.*

Essentially, this hardness result implies that finding the most vulnerable location requires an exponential-time algorithm *in the number of affected links*. Such algorithms might be feasible to implement when the number of these links is bounded by a small constant κ. The most intuitive approach is by *complete state enumeration*. Such an algorithm considers one candidate location at a time (obtained by the corresponding arrangement, as in Section IV); each location p defines a probabilistic graph $G = (V, E)$ where every edge $e \in E$ has a failure probability $f(e, p)$. Let E_1 denote the edges with zero failure probability, and E_2 the rest of the edges. The algorithm enumerates all subsets of E_2 and for each such subset S, it first computes the probability for such a failure pattern: $\Pr_S = \prod_{e\in S} f(e,p) \prod_{e\in E_2\setminus S}(1 - f(e,p))$; then, it computes the maximum flow F_S in $G_S = (V, E_1 \cup S)$. The expected maximum flow is $\sum_{S\subseteq E_2} \Pr_S \cdot F_S$, and its computation requires $2^{|E_2|} \leq 2^{\kappa}$ maximum-flow computations.[5] Alternative techniques, such as graph simplification, graph factoring, and inclusion-exclusion based approaches have also been studied [15]. However, all suggested algorithms still require exponential running time.

VIII. Experimental Results

We have obtained numerical results of the algorithms of Section IV for three networks within continental USA: Level 3's network of 230 links [31], Qwest's fiber-optic network of 181 links [43], and XO Communications' long-haul network of 71 links [56]. We used lightpath information (compound components) for the last two (65 lightpaths in case of Qwest's network and 37 for XO). In addition, for Qwest's network, we used the transmission rates of lightpaths to determine their weights. We conducted simulations with five accuracy values $\varepsilon = \{0.1, 0.2, \ldots, 0.5\}$ for links (simple components) for $k = 1, 2$, i.e., up to two attacks. For lightpaths, we used three values $0.2, 0.35, 0.5$ and simulated one attack ($k = 1$). For each case, we considered five attack radii, ranging between 60 and 300 miles and two f functions: one that decreases *linearly* with the distance, and the other that follows a *Gaussian* distribution (see Section III).

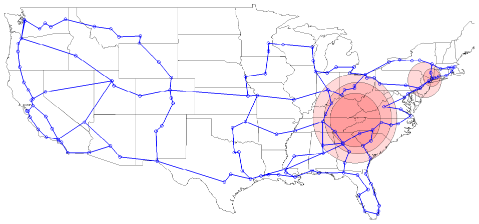


Fig. 7. Locations found by MaxExpectedDamageLocation on Qwest's network, a Gaussian f-function on simple components for various radii.

Fig. 6 shows the change in Φ with attack radius for a linear f for both links and lightpaths. We normalized the value of Φ, so that 100% implies the sum of the weights of all network components. Clearly, the marginal gain for increasing the attack radius is limited, and even small attacks of radius 60 miles can cause large damage, if they are placed in vulnerable locations. Moreover, increasing k to just two causes more than 1.5 times the number of links to fail indicating that there are multiple locations with similarly high impact. Fig. 7 depicts examples of best single attack locations on Qwest's network for a Gaussian f on links, and various radii.

We also compared Φ for different values of ε. Table I shows the results for links and lightpaths when the attack radius (resp., standard deviation) is 180 miles for the linear (resp., Gaussian) f-function. Here, Φ_L and Φ_G respectively denote Φ under linear and Gaussian f-functions. Our results show *no perceptible change in Φ when ε is changed, neither for links*

[5] Note that the arrangement of Section IV induces only an approximate solution. In this case, we need to scale the error parameter ε inversely with κ to avoid accumulating errors in the computation.

TABLE I

VALUES OF Φ FOR LINKS AND LIGHTPATHS UNDER LINEAR f-FUNCTION (Φ_L) AND GAUSSIAN f-FUNCTION (Φ_G). THE TOTAL WEIGHT OF LIGHTPATHS IS 1193 AND 37 FOR QWEST AND XO RESPECTIVELY.

	Level3		**Qwest**		**XO**	
	Φ_L	Φ_G	Φ_L	Φ_G	Φ_L	Φ_G
Links, k=1	20.5	69.4	14.1	37.2	6.1	15.6
Links, k=2	38.8	105.2	25.7	62.9	10.6	25.9
Lightpaths, k=1	-	-	475.7	615.1	11.1	15.8

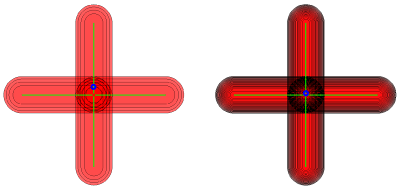


Fig. 8. Example with four links, where Φ varies significantly with ε. MAXEXPECTEDDAMAGELOCATION selects attack location with $\Phi = 2.677$ for $\varepsilon = 0.5$ (see arrangement on the left) and $\Phi = 3.788$–approximately 40% more–for $\varepsilon = 0.1$ (arrangement on the right). Notice that the fiber-links (in green) do not intersect (but their end-points are in close proximity).

nor for lightpaths. This conclusion holds for all 3 networks, for both f-functions and for various attack radii. This may be due to the fact that, in these networks, the locations found lie extremely close to a link and in many cases, close to a node, avoiding the worst-case (in terms of accuracy).

However, there do exist cases where Φ varies significantly with ε: in Fig. 8, 4 links of length 5 units are placed as shown with a small gap at the center. When the f-function is Gaussian with a standard deviation of 2.2 units and $\varepsilon = \{0.1, 0.5\}$, the values of Φ computed by MAXEXPECTEDDAMAGELOCATION are 3.79 and 2.68, respectively. While such cases where Φ varies significantly with ε exist, our results show that, *in practice, the dependence on ε is very limited.*

To validate our algorithm, we also computed Φ for all three networks when attack locations are restricted to a fine grid of cell size 0.6×0.6 miles. Fig. 4 (c) shows the effects on Qwest's network, of attacks of radius 180 miles centered at grid points. The point corresponding to the maximum value of Φ lies less than 0.5 miles from our algorithm's output (shown in red in Fig. 4(c)) and the values of Φ are also almost the same. These results further reinforce the conclusion that our algorithm is, in practice, close to optimal.

Finally, we compared the total number of locations examined by MAXEXPECTEDDAMAGELOCATION as well as the actual execution times for all values of ε. Fig. 9 shows the results for Qwest's network for three attack radii. We see that the complexity of the arrangement as well as the execution times for smaller values of ε are far higher than that for larger values showing that using larger values provides large reductions in running time with minimal loss of accuracy.

IX. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we provided a unified framework to identify vulnerable point(s), given a WDM network embedded in the Euclidean plane. A unique feature of our framework is its ability to cope with a *wide range of probabilistic attack and failure models*.

The basic building block of our framework is the algorithm MAXEXPECTEDDAMAGELOCATION, which locates efficiently a point in the plane that causes arbitrarily close to maximum expected damage on a network comprised of simple components. By its tolerance factor ε, MAXEXPECTEDDAMAGELOCATION trades accuracy with running time. We further extended and improved MAXEXPECTEDDAMAGELOCATION in various ways that allow it to deal with compound components, simultaneous attacks, networks equipped with a protection plan and to deal faster with simpler networks or distributions. We also evaluated its performance by simulation on three real WDM networks. Our numerical results show, quite surprisingly, that MAXEXPECTEDDAMAGELOCATION finds a location very close to optimal, even when taking a high tolerance factor ε (e.g., when it runs very fast but with a loose guarantee on the quality of its output). This makes MAXEXPECTEDDAMAGELOCATION an even more attractive tool for assessing network resilience.

Future research directions include adapting our algorithms to more sophisticated protection techniques, developing efficient planning methods for geographically-resilient networks and investigating the effect of adding minimal infrastructure (e.g., lighting-up dark fibers) on network resilience, as well as proving hardness results for approximation schemes. Moreover, we plan to determine how to use low-cost shielding for existing components to mitigate large-scale physical attacks.

REFERENCES

[1] P. K. Agarwal, D. Z. Chen, S. K. Ganjugunte, E. Misiołek, M. Sharir, and K. Tang, "Stabbing convex polygons with a segment or a polygon," in *Proc. ESA*, Sep. 2008.

[2] P. K. Agarwal, A. Efrat, S. K. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman, "Network vulnerability to single, multiple, and probabilistic physical attacks," in *Proc. MILCOM*, Nov. 2010.

[3] ——, "The resilience of wdm networks to probabilistic geographical failures," in *Proc. IEEE INFOCOM*, Apr. 2011.

[4] P. K. Agarwal, T. Hagerup, R. Ray, M. Sharir, M. H. M. Smid, and E. Welzl, "Translating a planar object to maximize point containment," in *Proc. 10th Annu. European Sympos. Algorithms*, 2002, pp. 42–53.

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

[6] B. Aronov and S. Har-Peled, "On approximating the depth and related problems," in *Proc. ACM-SIAM SODA*, Jan. 2005.

[7] M. O. Ball, C. J. Colbourn, and J. S. Provan, "Network reliability," in *Network Models*, ser. Handbooks in Operations Research and Management Science. Elsevier, 1995, vol. 7, ch. 11, pp. 673– 62.

[8] A. L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.

[9] R. Bhandari, *Survivable networks: algorithms for diverse routing*. Kluwer, 1999.

[10] D. Bienstock, "Some generalized max-flow min-cut problems in the plane," *Math. Oper. Res.*, vol. 16, no. 2, pp. 310–333, 1991.

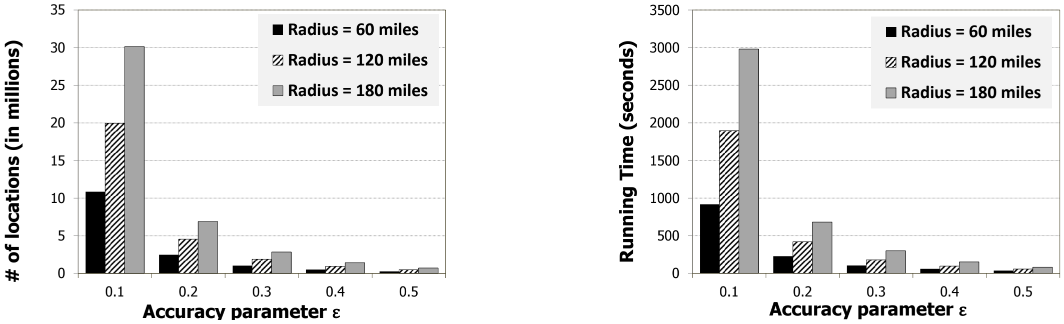
[11] J. Borland, "Analyzing the Internet collapse," *MIT Technology Review*, Feb. 2008. [Online]. Available: http://www.technologyreview.com/Infotech/20152/?a=f

[12] R. L. Church, M. P. Scaparra, and R. S. Middleton, "Identifying critical infrastructure: the median and covering facility interdiction problems," *Ann. Assoc. Amer. Geographers*, vol. 94, no. 3, pp. 491–502, 2004.

[13] G. Clapp, R. Doverspike, R. Skoog, J. Strand, and A. V. Lehmen, "Lessons learned from CORONET," in *OSA OFC*, Mar. 2010.

[14] K. L. Clarkson and P. W. Shor, "Applications of random sampling in computational geometry, II," *Discrete Comput. Geom.*, vol. 4, pp. 387–421, Sep. 1989.

[15] C. J. Colbourn, *The Combinatorics of Network Reliability*. Oxford University Press, 1987.



(a) *# of locations vs ε*

(b) *Running time vs ε*

Fig. 9. Number of locations examined by MaxExpectedDamageLocation and running times vs ε for various radii given a *linear* probability function.

[16] O. Crochat, J.-Y. Le Boudec, and O. Gerstel, "Protection interoperability for WDM optical networks," *IEEE/ACM Trans. Netw.*, vol. 8, no. 3, pp. 384–395, 2000.

[17] N. Dinh, Y. Xuan, M. T. Thai, P. Pardalos, and T. Znati, "On new approaches of assessing network vulnerability: hardness and approximation," *IEEE/ACM Trans. Netw.*, vol. 20, no. 2, pp. 609–619, Apr 2012.

[18] N. Dinh, Y. Xuan, M. T. Thai, E. K. Park, and T. Znati, "On approximation of new optimization methods for assessing network vulnerability," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 2678–2686.

[19] W. R. Forstchen, *One Second After*. Tom Doherty Associates, 2009.

[20] J. S. Foster, E. Gjelde, W. R. Graham, R. J. Hermann, H. M. Kluepfel, R. L. Lawson, G. K. Soper, L. L. Wood, and J. B. Woodard, "Report of the commission to assess the threat to the United States from electromagnetic pulse (EMP) attack, critical national infrastructures," Apr. 2008.

[21] R. L. Francis, *Facility Layout and Location: An Analytical Approach*. Prentice-Hall, Englewood Cliffs, NJ, 1974.

[22] L. K. Gallos, R. Cohen, P. Argyrakis, A. Bunde, and S. Havlin, "Stability and topology of scale-free networks under attack and defense strategies," *Phys. Rev. Lett.*, vol. 94, no. 18, 2005.

[23] S. K. Ganjugunte, "Geometric hitting sets and their variants," Ph.D. dissertation, Duke University, 2011.

[24] P. R. Goundan and A. S. Schulz, "Revisiting the greedy approach to submodular set function maximization," *Working paper*, 2008.

[25] A. F. Hansen, A. Kvalbein, T. Cicic, and S. Gjessing, "Resilient routing layers for network disaster planning," in *Proc. ICN*, Apr. 2005.

[26] M. M. Hayat, J. E. Pezoa, D. Dietz, and S. Dhakal, "Dynamic load balancing for robust distributed computing in the presence of topological impairments," *Wiley Handbook of Science and Technology for Homeland Security*, 2009.

[27] D. Hochbaum and A. Pathria, "Analysis of the greedy approach in problems of maximum k-coverage," *Naval Research Logistics (NRL)*, vol. 45, no. 6, pp. 615–627, 1998.

[28] IETF Internet Working Group, "Inference of Shared Risk Link Groups," Nov. 2001, Internet Draft. [Online]. Available: http://tools.ietf.org/html/draft-many-inference-srlg-02

[29] D. R. Karger, "A randomized fully polynomial time approximation scheme for the all-terminal network reliability problem," *SIAM Rev.*, vol. 43, no. 3, pp. 499–522, 2001.

[30] D. R. Karger and R. P. Tai, "Implementing a fully polynomial time approximation scheme for all terminal network reliability," in *Proc. ACM-SIAM SODA*, Jan. 1997.

[31] Level 3 Communications, Network Map. [Online]. Available: http://www.level3.com/interacts/map.html

[32] G. Liu and C. Ji, "Scalability of network-failure resilience: Analysis using multi-layer probabilistic graphical models," *IEEE/ACM Trans. Netw.*, vol. 17, no. 1, pp. 319 –331, Feb. 2009.

[33] D. Magoni, "Tearing down the Internet," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 6, pp. 949–960, Aug. 2003.

[34] Z. A. Melzak, *Companion to Concrete Mathematics; Mathematical Techniques and Various Applications*. Wiley, New York, 1973.

[35] A. Narula-Tam, E. Modiano, and A. Brzezinski, "Physical topology design for survivable routing of logical rings in WDM-based networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 8, pp. 1525–1538, Oct. 2004.

[36] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions - I," *Math. Prog.*, vol. 14, no. 1, pp. 265–294, Dec. 1978.

[37] S. Neumayer and E. Modiano, "Network reliability with geographically correlated failures," in *Proc. IEEE INFOCOM*, Mar. 2010.

[38] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano, "Assessing the vulnerability of the fiber infrastructure to disasters," in *Proc. IEEE INFOCOM*, Apr. 2009.

[39] ——, "Assessing the impact of geographically correlated network failures," in *Proc. IEEE MILCOM*, Nov. 2008.

[40] C. Ou and B. Mukherjee, *Survivable Optical WDM Networks*. Springer-Verlag, 2005.

[41] C. A. Phillips, "The network inhibition problem," in *Proc. ACM STOC*, May 1993.

[42] A. Pinar, Y. Fogel, and B. Lesieutre, "The inhibiting bisection problem," in *Proc. ACM SPAA*, Jun. 2007.

[43] Qwest, Network Map. [Online]. Available: http://www.qwest.com/largebusiness/enterprisesolutions/networkMaps/

[44] M. Rahnamay-Naeini, J. Pezoa, G. Azar, N. Ghani, and M. Hayat, "Modeling stochastic correlated failures and their effects on network reliability," in *Proc. IEEE ICCCN*, Aug. 2011.

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

[46] A. Sen, B. Shen, L. Zhou, and B. Hao, "Fault-tolerance in sensor networks: a new evaluation metric," in *Proc. IEEE INFOCOM*, Apr. 2006.

[47] M. Sharir, "The clarkson-shor technique revisited and extended," in *Proc. ACM SoCG*, Jun. 2001, pp. 252–256.

[48] M. Sharir and P. K. Agarwal, *Davenport-Schinzel Sequences and their Geometric Applications*. Cambridge University Press, 1995.

[49] A. K. Somani, *Survivability and Traffic Grooming in WDM Optical Networks*. Cambridge University Press, 2005.

[50] J. Spragins, "Dependent failures in data communication systems," *IEEE Trans. Commun.*, vol. 25, no. 12, pp. 1494 – 1499, Dec. 1977.

[51] J. P. Sterbenz, E. K. Çetinkaya, M. A. Hameed, A. Jabbar, Q. Shi, and J. P. Rohrer, "Evaluation of network resilience, survivability, and disruption tolerance: Analysis, topology generation, simulation, and experimentation (invited paper)," *Springer Telecommunication Systems*, 2011, to appear.

[52] K. Trivedi, D. S. Kim, and R. Ghosh, "Resilience in computer systems and networks," in *Proc. IEEE/ACM ICCAD*, Nov. 2009, pp. 74 –77.

[53] A. Vigneron, "Geometric optimzation and sums of algebraic functions," in *Proc. ACM-SIAM SODA*, Jan. 2010.

[54] C. Wilson, "High altitude electromagnetic pulse (HEMP) and high power microwave (HPM) devices: Threat assessments," CRS Report for Congress, July 2008. [Online]. Available: http://www.ntia.doc.gov/broadbandgrants/comments/7926.pdf

[55] W. Wu, B. Moran, J. Manton, and M. Zukerman, "Topology design of undersea cables considering survivability under major disasters," in *Proc. WAINA*, May 2009.

[56] XO Communications, Network Map. [Online]. Available: http://www.xo.com/about/network/Pages/maps.aspx

[57] D. Zhou and S. Subramaniam, "Survivability in optical networks," *IEEE Network*, vol. 14, no. 6, pp. 16–23, Nov.-Dec. 2000.

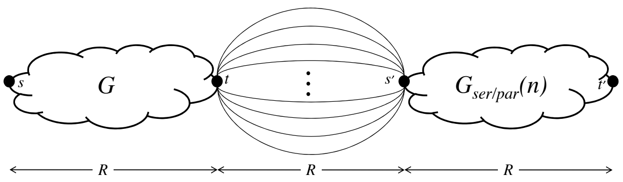


Fig. 10. Illustration of the proof of Theorem 7.

Appendix A
Proof of Theorem 7

The proof is by reduction from the $s{-}t$ expected maximum flow problem, in which one need to compute the expected maximum flow from node s to node t. We restrict our graphs to be with capacity 1 and all edge failure probabilities to be the same (in our case, $1/2$). It is known that the problem is still #P-complete [7]. In addition, by enumerating over all possible combinations, and since for each instance the maximum flow is integral, one can verify that the *value* of the expectation is a multiple of $\frac{1}{2^n}$. Trivially, the expected maximum flow is bounded by n.

Let G be such a network of n links (all their weights are 1) and let R denote its physical diameter. Assume, without loss of generality, that $d(s,t) = R$. We define the following f-function to determine geographical failures:

$$f(p,q) = \begin{cases} \frac{1}{2} & d(p,q) \leq R \\ 0 & \text{otherwise} \end{cases}$$

Note that f induces a uniform reliability problem on G.

We next show how to construct a family of graphs so that by finding their most vulnerable location (in terms of expected $s{-}t$ flow), one can compute the value of the expected $s{-}t$ flow on the original graph, hence establishing that finding the most vulnerable location is in #P. Note that our family graphs will contain exponential number of graphs (each of polynomial size), however we will need to consider only a polynomial number of them (and can construct them on-the-fly).

We first consider a simple serial-parallel construction: Given a graph G, where the probability that s' is connected to t' is C, then *(i)* if one adds an edge (s'', s') then the probability that s'' is connected to t' is $C/2$; *(ii)* if one add an edge (s', t') then the probability that s' is connected to t' is $1/2 + C/2$. With a combination of x serial-parallel compositions, one can build a sequence of graphs, $G_{ser/par}(i)$, (for any $1 \leq i < 2^x$) whose distinguished nodes are connected with probability $i/2^x$. We scale the physical length of the edges so that the physical size of the entire graph is R (implying that a single attack can affect all edges). In the rest of the proof, we fix x to be $2n$.

Let F be the value of the maximum flow of G had all the edges not failed (this can be computed in polynomial time). At each iteration, we use some graph $G_{ser/par}(i)$, where the capacity of all its edges is F. We connect nodes t and s' by n^2 parallel edges of capacity F and length larger than R (see Fig 10). Note that the expected minimum cut size (and hence induced bottleneck) on $G_{ser/par}(i)$ is exactly $F\frac{i}{2^{2n}}$. Our reduction follows by a binary search on the parameter i of the family of the graphs: In general, at each iteration, we compute the most vulnerable location in term of expected maximum flow between s to t' and distinguish between five different cases:

1) The location affects only G. This implies that the expected s-t flow is strictly less than $F\frac{i+1}{2^{2n}}$. We will continue in the next iteration with smaller value of i (in a binary search manner).
2) The location affects only G and the n^2 parallel edges. Failure on the parallel edges affects the maximum flow if and only if all edge fail. Since this happens with probability $\frac{1}{2^{n^2}}$, the expected s-t flow is strictly less than $F\frac{i+1}{2^{2n}}$. We will continue in the next iteration with smaller value of i (in a binary search manner).
3) The location affects only the n^2 parallel edges. This implies that the expected maximum flow is more than $F(1 - \frac{1}{2^{n^2}})$, which by the granularity of the values of the expected maximum flow implies that it is F (and the algorithm finishes).
4) The location affects only $G_{ser/par}(i)$. This implies that the expected s-t flow is strictly more than $F\frac{i-1}{2^{2n}}$. We will continue in the next iteration with higher value of i (in a binary search manner).
5) The location affects $G_{ser/par}(i)$ and the n^2 parallel edges. This, again, implies that the expected s-t flow is strictly more than $F\frac{i-1}{2^{2n}}$. We will continue in the next iteration with higher value of i (in a binary search manner).

We start with $i = 2^{2n-1}$. When the binary search completes, we get the accurate value of G's expected maximum s-t flow. The number of iterations is bounded $O(\log 2^{2n}) = O(n)$ and therefore our reduction is polynomial, as required.

PLACE
PHOTO
HERE

Pankaj K. Agarwal earned his PhD in Computer Science from the Courant Institute of Mathematical Sciences at New York University. He joined the Department of Computer Science of Duke University in 1989 where he is now the RJR Nabisco Professor of Computer Science and Professor of Mathematics. His research interests include geometric algorithms and data structures, computational molecular biology, spatial databases, global change, geographic information systems, sensor networks, and robotics. He has authored four books, and more than 250 scholarly articles in various journals, edited volumes, and international conferences. He has received many awards, including National Young Investigator, Sloan Fellow, and ACM Fellow, and he serves on the editorial boards of a number of journals.

PLACE
PHOTO
HERE

Alon Efrat is an associate professor in the Department of Computer Science at the University of Arizona. He earned his PhD from Tel-Aviv University under the supervision of Prof. Micha Sharir. He was also a postdoctorate research assistant at Stanford University, and at IBM Almaden Research Center. His research areas include geometric algorithms and their applications to sensor networks, robotics and computer vision. He is the author or co-author of nearly 95 publications, almost all in peer-reviewed, prestigious venues. He won the NSF CAREER award in 2004. He has served on many NSF panels and technical program committees in different areas, on the editorial board of the International Journal of Computational Geometry and its Application (IJCGA), and was a guest editor of this journal.

PLACE
PHOTO
HERE

Shashidhara K. Ganjugunte received his B.E. from Bangalore University in 2001. He then worked as software design engineer at Microsoft India, Hyderabad until 2003. He received his Master's degree from University of Maryland, Baltimore County in 2005, and his Ph.D in 2011 from Duke University. His research involves developing algorithms for geometric problems with applications in sensor networks, robotics and structural biology.

PLACE
PHOTO
HERE

David Hay received his BA (summa cum laude) and PhD degree in computer science from the Technion - Israel Institute of Technology in 2001 and 2007, respectively. He is currently a senior lecturer (assistant professor) at the Rachel and Selim Benin School of Computer Science and Engineering, Hebrew University, Jerusalem, Israel. Prior to joining the Hebrew University, David Hay was with IBM Haifa Research Labs (1999-2002), Cisco Systems (2006), Ben-Gurion University of the Negev (2007-2008), Politecnico di Torino (2008-2009), and Columbia University (2009-2010). His main research interests are algorithmic aspects of high-performance switches and routers—in particular, QoS provisioning, competitive analysis, and packet classification.

PLACE
PHOTO
HERE

Swaminathan Sankararaman received the B.E. degree in Computer Science and Engineering from Anna University, Chennai, India, in 2006. He received his M.S. and Ph.D. degrees in Computer Science in 2008 and 2011 from the University of Arizona, Tucson and is currently a Postdoctoral Associate at the Department of Computer Science, Duke University. His research interests lie in the applications of geometric algorithms to optimization problems in wired/wireless networking.

PLACE
PHOTO
HERE

Gil Zussman (S'02-M'05-SM'07) received the Ph.D. degree in Electrical Engineering from the Technion - Israel Institute of Technology in 2004. In 2004–2007, he was a Postdoctoral Associate at MIT. He is currently an Assistant Professor at the Department of Electrical Engineering in Columbia University. His research interests are in the area of wireless networks. Gil received the Marie Curie Outgoing International Fellowship, the Fulbright Fellowship, the IFIP Networking 2002 Best Student Paper Award, the OPNETWORK 2002 and the ACM SIGMETRICS/IFIP Performance 2006 Best Paper Awards, and the 2011 IEEE Communications Society Award for Outstanding Paper on New Communication Topic. He was a member of a team that won the 1st place in the 2009 Vodafone Americas Foundation Wireless Innovation Competition, and received the DTRA Young Investigator Award and the NSF CAREER Award.