Damage detection via shortest-path network sampling

Fabio Ciulla,[1,*] Nicola Perra,[1] Andrea Baronchelli,[2] and Alessandro Vespignani[1,3]
[1] *Laboratory for the Modeling of Biological and Socio-technical Systems, Northeastern University, Boston, Massachusetts 02115, USA*
[2] *Department of Mathematics, City University London, Northampton Square, London EC1V 0HB, United Kingdom*
[3] *Institute for Scientific Interchange (ISI), Torino, Italy*

(Received 27 January 2014; published 30 May 2014)

Large networked systems are constantly exposed to local damages and failures that can alter their functionality. The knowledge of the structure of these systems is, however, often derived through sampling strategies whose effectiveness at damage detection has not been thoroughly investigated so far. Here, we study the performance of shortest-path sampling for damage detection in large-scale networks. We define appropriate metrics to characterize the sampling process before and after the damage, providing statistical estimates for the status of nodes (damaged, not damaged). The proposed methodology is flexible and allows tuning the trade-off between the accuracy of the damage detection and the number of probes used to sample the network. We test and measure the efficiency of our approach considering both synthetic and real networks data. Remarkably, in all of the systems studied, the number of correctly identified damaged nodes exceeds the number of false positives, allowing us to uncover the damage precisely.

DOI: 10.1103/PhysRevE.89.052816 PACS number(s): 89.75.Hc, 89.20.Hh

I. INTRODUCTION

Real-world networks are often the result of a self-organized evolution without central control [1–3]. The physical Internet and the World Wide Web (WWW) are two prototypical examples where the interplay of local and nonlocal evolution mechanisms defines the global structure of the network. In absence of determined blueprints, the only way to characterize the global structure of self-organizing systems is by devising sampling experiments, such as *traceroute* probing for the physical Internet [4–16] and *web crawlers* for the WWW [17–19]. However, sampling processes are limited by time, physical constraints, and in general, guarantee access only to a part of the network. In this context, the detection of network damage is a difficult task. The lack of information on the exact structure of the system makes it extremely difficult to identify what damage has actually been suffered and segregate damaged elements from those that have simply not yet been probed by the sampling process.

Here, we numerically study the effectiveness of shortest-path sampling strategies for damage detection in large-scale networks. These methods are currently used in Internet probing [1,20–24], including failure detection via traceroute-like tools and end-to-end measurements. We consider different attack strategies, namely, random or connectivity based, and introduce a global measure, M, that allows us to quickly identify damages that induce large variations in the routing patterns of networks. We then propose a statistical method able to classify nodes as damaged or functioning in the case of partial network sampling. In our analysis, we first sample a given network structure via shortest-path probes [1,20–24] obtaining a partial representation of its nodes and connectivity patterns. Then, we damage the network by removing nodes according to different strategies and sample again the damaged network supposing not to know the location and magnitude of the damage. During this sampling, we constantly monitor whether the probability not to have seen a node exceeds the expected value calculated on the basis of the sampling of the undamaged graph. We define a statistical criterion to assess which nodes of the network can be considered damaged, and we test the performance of the method by looking at the number of true and false positives it identifies.

We perform numerical experiments on synthetic networks, either heterogeneous, generated with the uncorrelated configurational model (UCM) [25], or homogeneous, obtained through the Erdös-Rènyi model (ER) [26]. Remarkably, the magnitude and location of the damage can be detected with fairly good confidence. The accuracy improves for nodes that play a central role in the network's connectivity. Namely, the detection of damaged hubs is more reliable than that of peripheral nodes. As a practical application, we consider the damage detection on the physical Internet at the level of the autonomous system (AS). We use the AS topology provided by the DIMES project [11]. In this case, to simulate realistic damages, we remove nodes according to their geographical position. In doing so, we simulate critical events, such as large-scale power outages, deliberate server switch offs [27], or other major localized catastrophic events [28]. Interestingly, also in this case, our methodology allows us to statistically identify the extent and location of the damage with reasonable accuracy.

The paper is organized as follows: In Sec. II, we present the sampling method used. In Sec. III, we introduce a measure that provides a general estimation of the damage extension in sampled networks. In Sec. IV, we provide a method to infer the status of single nodes using a p-value test. In Sec. V, we validate this method by applying it to the physical Internet network. Finally, in Sec. VI we present our conclusions and final remarks.

II. SHORTEST-PATH SAMPLING OF UNDAMAGED NETWORKS

The sampling of networks via shortest paths consists in sending probes from a set of nodes that have been defined as sources toward another set of nodes chosen to be targets. Each probe

[*] f.ciulla@neu.edu

travels through the network following a shortest path and records each node and link visited, returning a path. This is, to a first approximation, what is executed by Internet mapping projects that use the traceroute tool [8] as a probing method. This methodology infers paths by transmitting a sequence of limited-time-to-live transmission control protocol/Internet protocol (TCP/IP) packets from a source node to a specified target on the Internet. The nodes visited along the way send their IP addresses as a response and create the path. The union of all the paths returned by the probes creates the sampled picture of the network. However, mapping the real physical Internet is complicated, and existing approaches have major limitations [4]. For example, visited nodes can fail to provide their IP address, and wrong or outdated forwarding route registries can result in forwarding route indications that are not optimal. Our study is inspired by the traceroute tool, although we assume that probes follow the shortest paths, neglecting the real-world limitations aforementioned.

Here, we focus on undirected and unweighted networks consisting of N nodes and M edges $\mathcal{G}(N,M)$. We fix a set of sources $\mathbf{S} = (s_1,s_2,\ldots,s_{N_S})$ and a set of targets $\mathbf{T} = (t_1,t_2,\ldots,t_{N_T})$ among the N nodes, withN_S and N_T being the total number of sources and targets, respectively. For each pair of nodes, taken in the two sets, a shortest-path probe is sent. After all of the $N_S \times N_T$ probes have reached their targets, all the resulting paths are merged in a sampled network that we denote as $\mathcal{G}^*(N^*,M^*)$. Here, and in the rest of the paper, the star symbol indicates sampled quantities, so N^* is the number of discovered nodes and M^* is the number of discovered links via the shortest-path sampling.

In general, for each source-destination pair, we can have two or more equivalent shortest paths. More precisely, there could be different strategies to numerically simulate the shortest-path probing:

(1) *Unique shortest path*. The shortest path between a node i and a target T is always the same independently of the source S. Each shortest path is selected initially, and they will never change.

(2) *Random shortest path (RSP)*. Each shortest path is randomly selected every time among the equivalent ones.

(3) *All shortest paths*. All possible, equivalent, shortest paths between shortest paths are discovered.

In the following, we will use the RSP probing strategy [20,29–31]. Both sources and targets are chosen randomly among all the nodes. Inspired by real Internet probing, we investigate scenarios in which the order of magnitude of sources is $N_S = O(10)$, while the order of magnitude of the density of targets, ρ_T, is $O(10^{-1})$. Along with the raw number of discovered nodes, we also keep track of the visit probability p_i for each visited node i, defined as the ratio between the number of shortest-path probes passed through the node i and the total number of probes sent $N_S \times N_T$:

$$p_i = \frac{\sum_{j=1}^{N_S \times N_T} \delta_{i,j}}{N_S \times N_T}, \tag{1}$$

where $\delta_{i,j}$ is equal to 1 if the node i is seen by the probe j. In the limit in which both N_S and N_T approach N, the probability p becomes the betweenness [20]. Instead, in more realistic cases where the number of sources and targets is small, the nodes visit probability is just an approximation of this quantity [20]. Considering this limit, we show the distribution of p in Fig. 1 in both UCM and ER networks with 10^5 nodes. The number of sources is 15 and the target density is 0.2. The curves show a power-law behavior, except for the presence of two peaks, representing the visit probability of sources and targets. The peak for large values of p_i is the consequence of the sources' visits and appears in correspondence to $p_i = p_{\text{source}} = 1/N_S$, $\forall i \in \mathbf{S}$. The other peak is due to targets and occurs for $p_i = p_{\text{targets}} = 1/N_T$, $\forall i \in \mathbf{T}$ [32].

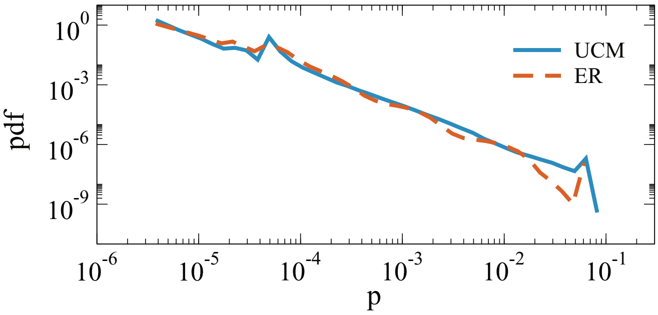

FIG. 1. (Color online) Probability distribution function (pdf) of nodes visit probability for undamaged UCM (blue solid line) and ER (orange dashed line) networks. The two peaks that deviate from the overall heavy-tailed behavior occur at $p_i = 1/N_S$ and $p_i = 1/N_T$, and represent the visit probability of sources and targets, respectively. The curves are the average over 100 independent simulations.

III. DAMAGE DETECTION

In order to introduce damage in the network, we consider that N_D nodes are not functional, i.e., a fraction $\rho_D = N_D/N$ of nodes and all of their links are removed from the network $\mathcal{G}$. We define the damaged network $\mathcal{G}_D(N_D,M_D)$, where the subscript D denotes damage. Damaged nodes are selected either randomly or according to a degree based strategy in which nodes are removed according to their position in the degree ranking (hubs first or leaves first). While target nodes can be damaged, we assume that no sources are damaged. In these settings, we aim at inferring the damage using shortest-path sampling by looking at the number of nodes discovered before and after the damage occurs.

Shortest-path probes are sent between each pair of source-target nodes. The sampled network after the damage $\mathcal{G}_D^*$ differs in general from the sampled view $\mathcal{G}^*$ of the original undamaged network because of the changed topology due to the missing nodes. To quantify the damage, we introduce the quantity

$$M = 1 - \frac{N_D^*}{N^*}, \tag{2}$$

where N^* and N_D^* are the number of discovered nodes in the undamaged and damaged network, respectively. If no damage occurs, the number of nodes discovered in $\mathcal{G}_D$ is similar to the one discovered in $\mathcal{G}$ so that $N^* \simeq N_D^*$ and $M \simeq 0$ [33]. If less nodes are seen in $\mathcal{G}_D$ than in $\mathcal{G}$, then $M > 0$, with $M = 1$ representing the extreme case in which no nodes are discovered in the damaged network. Interestingly, the quantity M can also assume negative values. Indeed, it is possible to see more nodes in $\mathcal{G}_D$ with respect to $\mathcal{G}$. Although this case

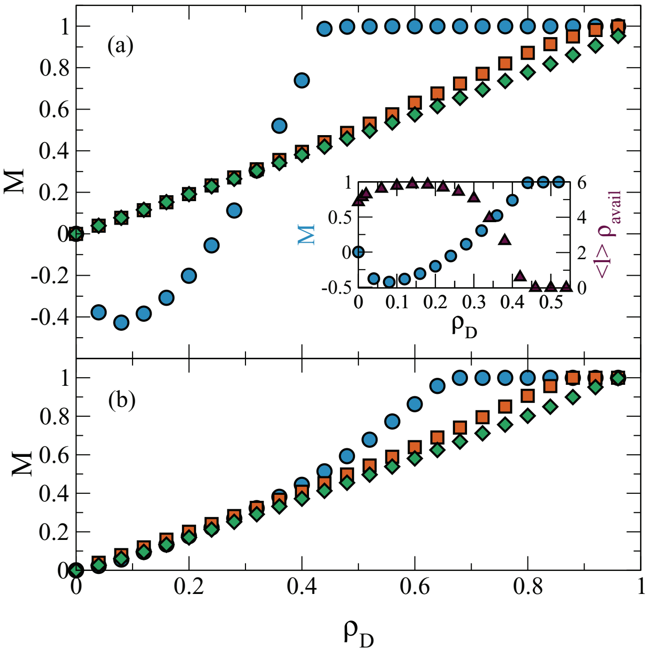

FIG. 2. (Color online) The behavior of M is shown for three different damage strategies: random (red squares), small degree nodes first (green diamonds), and hubs first (blue circles). (a) UCM scale-free graph. Inset: The behavior of M when high degree nodes are removed first is compared to the quantity $\langle l\rangle\rho_{\mathrm{av}}$ as a function of ρ_D. (b) ER random graph. For the UCM network, the minimum indicates the enhanced discovery given by the lack of hubs. Each plot is the median among 100 independent assignments of sources and targets. Error bars illustrating the 95% confidence interval are too small to be visible at this scale.

may sound counterintuitive at first, a closer look at the effect in the topology induced by removing nodes clearly explains its meaning. Indeed, by removing some central nodes in the network (in the next section, we discuss this point in detail), the length of the shortest paths might increase on average as well as the number of discovered nodes.

Numerical simulations

We measure the quantity M in damaged homogeneous and uncorrelated heterogeneous networks [1–3,34] generated through the ER and the UCM algorithm, respectively. The network size is fixed at $N = 10^5$ nodes, and the number of sources and target density are $N_S = 15$, and $\rho_T = 0.2$, respectively. The average degree is $\bar{k} = 8$ for both the topologies. The exponent γ for UCM is 2.5. As mentioned above, sources and targets are randomly selected. We consider different damage strategies in which the removed nodes are selected at random or based on their degree. We further divide the latter strategy considering two cases in which nodes are removed in increasing (hubs first) or decreasing order of degree (leaves first).

Figure 2 shows M as a function of ρ_D for the three different attack mechanisms, for the two different types of network. The top panel presents data for the UCM network. The random nodes removal strategy gives the same qualitative behavior as the one in which the small degree nodes are attacked first. This is not surprising, since the probability that a randomly selected node has a small degree is extremely high due to the power-law degree distribution of the network. The two strategies select, on average, the same category of nodes. A big difference can be noted when nodes are attacked in decreasing order of degree (high degree nodes, hubs, are attacked first). For small values of ρ_D, the value of M assumes negative values, meaning that more nodes are discovered in the damaged network than in the undamaged one. Indeed, hubs act as shortcuts for network connectivity. Their failure causes the rerouting of probes toward lower degree nodes and the consequent growth of the average length of the shortest paths $\langle l\rangle$. As ρ_D increases, this trend is contrasted by the progressive fragmentation of the network in many disconnected components.

In order to estimate how much the network has been fragmented by the damaging process, we define the quantity ρ_{av} as the average of the ratio between the nodes of the components in which each source is located by the total number of nodes N of the undamaged network. After a certain amount of nodes are removed, the graph undergoes disconnection and more than one component appears. At this point, a shortest-path probe can reach only the nodes belonging to the component where the source is located. Components with no sources will no longer be accessible. ρ_{av} is a decreasing function of ρ_D, assuming the value 1 when there is no damage, whereas it becomes $\rho_{\mathrm{av}} = N_S/N$ in the limit of $\rho_D \simeq 1$, when only the sources survive and each of them constitutes one component. Neither $\langle l\rangle$ or ρ_{av} alone explains the presence of the minimum quantity M in the plots. Instead, the product of the two $\langle l\rangle\rho_{\mathrm{av}}$ does: It represents the average number of nodes discovered by each shortest-path probe rescaled by the number of nodes effectively available to be discovered. The relation of this quantity with the minimum for M is shown in the inset of Fig. 2(a). The argument above is confirmed by the behavior of M in ER graphs. Here, removing the nodes with higher degree has a much smaller impact on the topology, and consequently, there is no increase in the amount of nodes discovered in the damaged graph $\mathcal{G}_D$ with respect to the original one $\mathcal{G}$. The plot of M for hubs removal in the ER network does not show a minimum, and substantially, the damage detection works similarly for all damage strategies.

IV. SINGLE NODE DAMAGE DETECTION

While the measure M quantifies the damage at the global level, it does not provide any information about specific nodes of the network. In this section, we address the damage of individual nodes by assuming that the information gathered during the exploration of the undamaged network constitutes the null hypothesis of our measure, namely, that none of the nodes is damaged. We start by monitoring the network $\mathcal{G}$, assuming that it is not damaged. Every time we send a shortest-path probe, we obtain a better approximation of the sampled network $\mathcal{G}^*$ with increasing number of discovered nodes N^*. At the same time, we collect information about how many times a probe passes through a node i resulting in visit probability p_i defined in Eq. (1). Later, the network is damaged according to one of the strategies discussed above and sampled via shortest-path probes. By definition, any node that is discovered during the sampling is not damaged.

However, the situation is less clear for nodes that have not been discovered. Indeed, the reason why a node is not seen can be either that it is actually damaged or that the sampling has missed it because the damage has altered the shortest-path routing. In order to infer the state of undiscovered nodes, we use a p-value test [35] applied to the visit probability p_i. More precisely, we calculate the probability $(1-p_i)^\tau$ of not seeing the node i after a number τ of shortest-path probes. τ can assume any integer value from 1 to $N_S \times N_T$. The p-value test consists of imposing the equality between this quantity and an arbitrary confidence level C:

$$C = (1-p_i)^{\tau_i}. \tag{3}$$

Note that after imposing the equality, τ_i has the index i as for p_i. This is because τ_i is different for each node. By taking the logarithm on both sides of the equation, we obtain

$$\tau_i = \frac{\ln C}{\ln(1-p_i)}. \tag{4}$$

If the node i has not been seen at least once before τ_i probes have been sent, then we can state that i is damaged with statistical confidence C. Here, we are assuming that the visit probability of nodes does not change after the damage. This holds up when the damage is a relatively small perturbation and does not change the connectivity of the network or its dynamical properties [3,36,37]. After the p_i values have been determined for all of the nodes, the value of C tunes the number of probes to be sent before declaring a node damaged. If C is selected to be large, nodes will be considered as damaged much earlier but with a small statistical confidence, leading to a large number of false positive (F_P) detections. Conversely, if C is set to be small, more probes are needed to state if a node is damaged or not. The accuracy improves, and the final response will eventually return only actual damaged nodes, the true positive damaged nodes (T_P). On the other hand, the number of probes needed to reach this level of statistical confidence will be much higher, resulting in a longer sampling process.

The value of C is an input of the method, and it can be chosen by opportunely tuning the trade-off between a poor but fast sampling that produces a high number of F_P and an accurate but slow sampling that generates more T_P. We evaluate the performances of the damage detection strategy measuring its precision and recall. In particular, the precision α is defined as

$$\alpha = \frac{T_P}{T_P + F_P}. \tag{5}$$

The recall r is instead

$$r = \frac{T_P}{T_P + F_N}, \tag{6}$$

where F_N indicates the number of false negatives, i.e., nodes damaged but not detected. In a given network, precision and recall are functions of the parameter C. In Fig. 3 we plot α for different values of C in both UCM and ER networks. In the top panel, we remove top degree nodes and set $\rho_D = 10^{-3}$. As expected α increases as C decreases. Interestingly, in the case of the UCM topology, the increase is slower. Indeed, we can notice that α reaches an arbitrary level of 90% (dashed line) for $C = 10^{-10}$, while in the case of the ER network, the same level is reached for $C = 10^{-5}$. The extremely low values of C, especially in the case of the UCM network, is justified by the presence of the logarithm function in the numerator of Eq. (4). Considering the big absolute value of the denominator for big p_i, very low values of C are required to have τ_i ranging from 0 to $N_S \times N_T$. The quite different value of C in the two networks, for top degree nodes removal, can be explained considering the distribution of p_i of the nodes that will be damaged in the case of hubs removal (Fig. 4). In the UCM network, the top degree

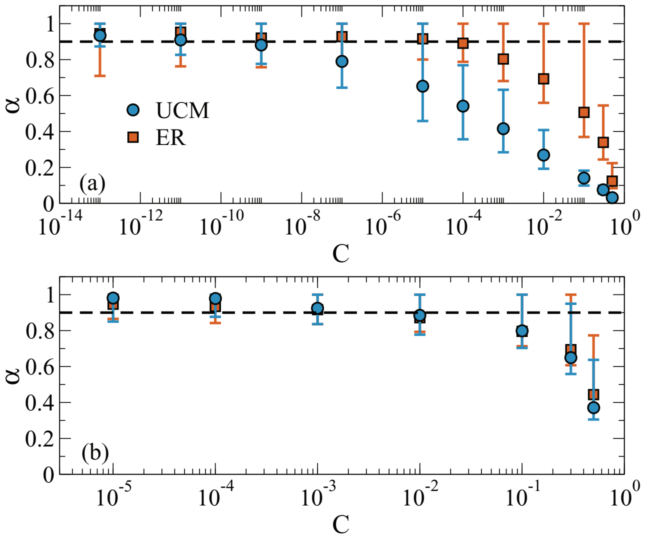

FIG. 3. (Color online) Precision α as a function of C for (a) hubs removal with fraction of removed nodes $\rho_D = 0.001$ and (b) random nodes removal with fraction of removed nodes $\rho_D = 0.01$. The black dashed line indicates $\alpha = 0.9$. Dots represent the average over 100 independent simulations, and error bars illustrate the 95% confidence interval.

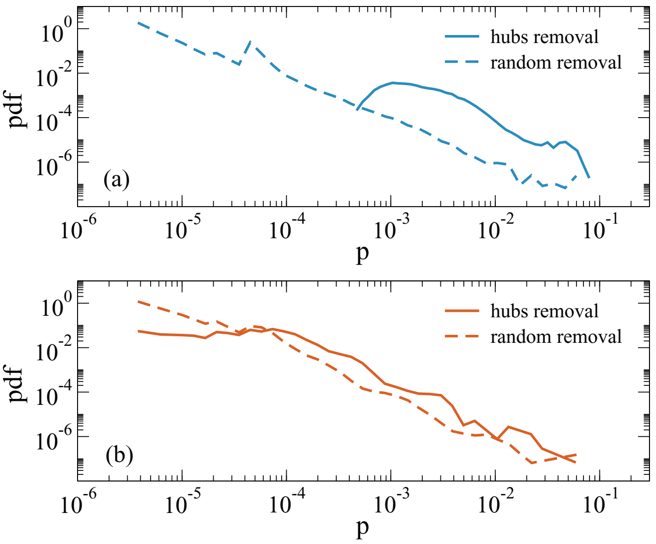

FIG. 4. (Color online) Probability distribution function (pdf) of nodes visit probability in the undamaged UCM (a) and ER (b) networks restricted to nodes that will be later damaged with two different strategies. Curves are the average over 100 independent simulations.

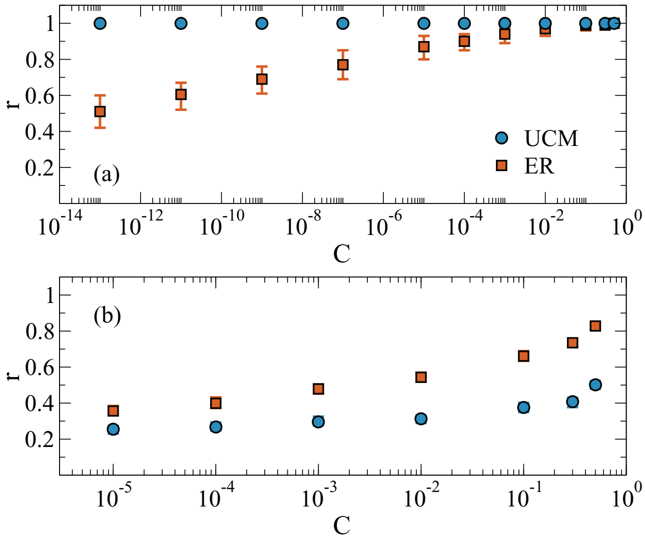

FIG. 5. (Color online) Recall r as a function of C for (a) hubs removal with fraction of removed nodes $\rho_D = 0.001$ and (b) random nodes removal with fraction of removed nodes $\rho_D = 0.01$. Dots represent the average over 100 independent simulations, and error bars illustrate the 95% confidence interval.

nodes have a much higher visit probability than the rest of the nodes. Large values of p_i combined with C via Eq. (4) produce small values of τ_i, allowing all removed nodes to be promptly declared damaged. The downside of this effect is the onset of a large number of F_P that leads to smaller precision. In the ER network, instead, the role of high degree nodes is not so determinant, and their visit probability is almost indistinguishable from that of random nodes. As a consequence, larger values of C are required to produce τ_i small enough to allow the algorithm to declare the nodes damaged.

In Fig. 3(b), we show the same curve for the random damaging strategy setting $\rho_D = 10^{-2}$. In this case, the behavior of α in the two topologies is very similar, and it is due to the p distributions that in both networks span the entire range of possible visit probabilities, thus reproducing the same curves shown in Fig. 1.

In Fig. 5, we study the recall r as a function of C. In this case, we can see that r reaches 1 for all the values of C investigated when damaging hubs in an UCM network. All damaged nodes are detected during the sampling. This is, again, a consequence of the very large visit probabilities for high degree nodes in the UCM network, which allows prompt detection of hubs removal. In the ER network, instead, the presence of low visit probability nodes among those in the top degree ranks require larger values of C in order to send enough probes to declare nodes damaged and improve the recall. It is crucial to stress that $\tau_{\max} = N_S \times N_T$. Any node that for a given C is characterized by $\tau_i > \tau_{\max}$ will not be evaluated by the algorithm. In the bottom panel, we see the recall for random nodes removal in both topologies. Here, the behavior of r is similar for both UCM and ER networks as a consequence of the similar visit probability distributions.

Numerical simulations in synthetic networks

We apply the statistical criterion developed in the previous section to the two types of synthetic networks, UCM and ER, with 10^5 nodes and two damaging strategies, high degree and random nodes removal. We send shortest-path probes from 15 sources to a number of targets equal to a fraction $\rho_T = 0.2$ of total nodes. In order to compare the results of this part of the study for different topologies and damage strategies, we arbitrarily fixed the C value of the one correspondent to a precision of $\alpha = 0.9$ in each system.

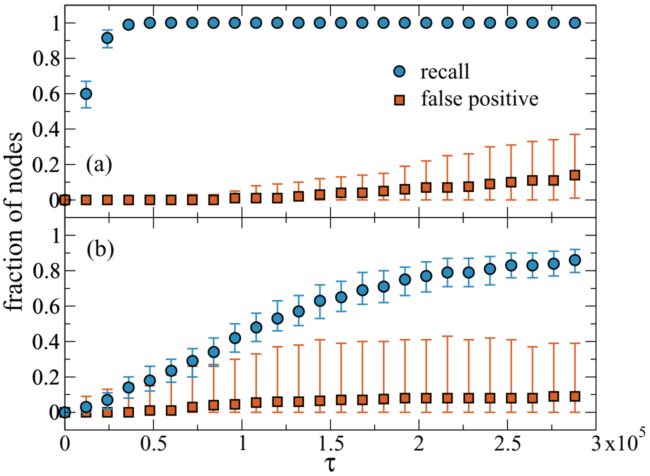

FIG. 6. (Color online) Recall r (blue circles) and normalized number of false positive f_p (orange squares) for high degree nodes removal as a function of number of probes in UCM (a) and ER (b) networks. The fraction of removed nodes is $\rho_D = 0.001$. The values of confidence level C are 10^{-10} and 10^{-5} for UCM and ER networks, respectively. Points are the median among 100 realizations with independent choice of sources and target. Error bars illustrate the 95% confidence interval.

Let us first consider UCM networks subject to the removal of the top 100 nodes ranked according to the degree ($\rho_D = 10^{-3}$). In Fig. 6, we plot the recall r and the normalized number of F_P, $f_p = F_P/(T_P + F_N)$, as a function of τ. Interestingly, the recall reaches 1 quickly. The absence of the hubs is promptly detected by the method. In Fig. 7, we show the behavior of the same quantities in the case of the random removal of nodes considering $\rho_D = 10^{-2}$. In this case, the recall increases slowly, while f_p remains constant after an initial increase. An interesting feature of the r curve is the presence of a jump. This is the consequence of the peak in the distribution of p_i that is mapped into τ_i via Eq. (4). It occurs at the value of τ corresponding to $p_{\text{targets}} = 1/N_T$ and is caused by the enhanced visit probability of target nodes. Since targets are assigned randomly in UCM networks, they are probably small degree nodes that are visited as if they are set to be targets. This implies that a specific number of probes equal to

$$\tau_{\text{targets}} = \frac{\ln C}{\ln(1 - p_{\text{targets}})} = \frac{\ln C}{\ln(1 - 1/N_T)} \tag{7}$$

is necessary to be able to say if targets are damaged or not. Since targets are 20% of the nodes, once τ_{targets} is reached, a conspicuous amount of nodes can be declared damaged or not. It is worth noting that only the number of T_P increase in correspondence with the jump, and f_p do not exhibit any discontinuity. This means that we have a better view of the damage without affecting the accuracy.

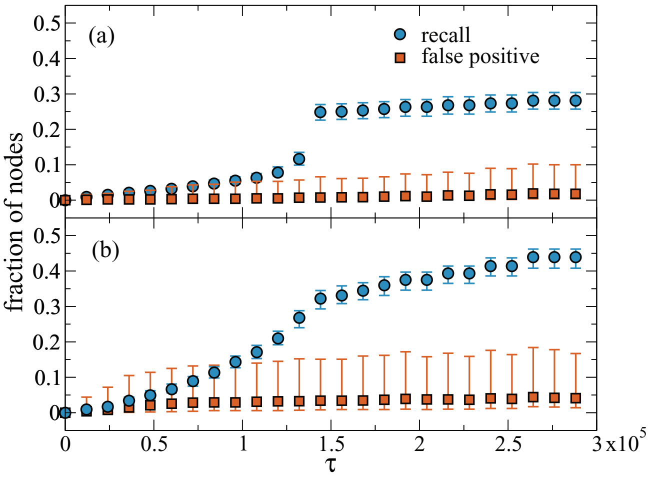

FIG. 7. (Color online) Recall r (blue circles) and normalized number of false positives f_p (orange squares) for random nodes removal as a function of the number of probes in UCM (a) and ER (b) networks. The fraction of removed nodes is $\rho_D = 0.01$. The value of confidence level C is 10^{-3} for both the UCM and ER networks. Points are the median among 100 realizations with independent choice of sources and target. Error bars illustrate the 95% confidence interval.

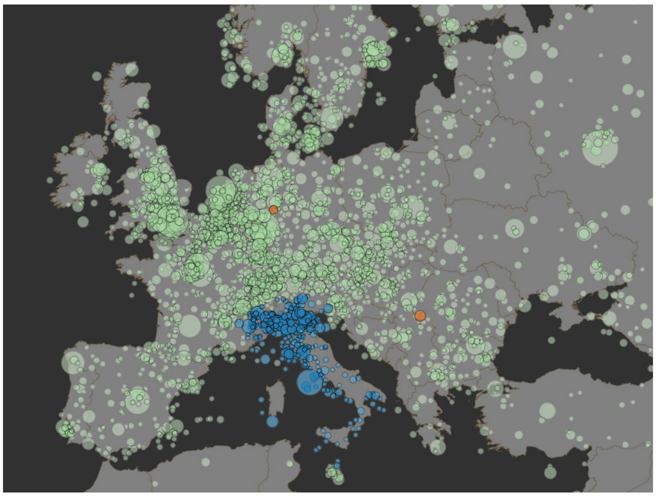

FIG. 8. (Color online) Map of Europe showing the detected Italian damaged nodes in the real AS network. Each circle represents one AS, and its size is proportional to the degree. Green circles are the working nodes, blue ones are the T_P, and orange ones are the F_P. Nodes located at sea are an effect of the finite accuracy of geographical coordinates provided.

Let us now consider ER networks subject to removal of nodes in decreasing order of degree. In Fig. 6(b), we plot r and f_p as a function of τ. As for the case of the UCM network, the recall increases, even if slower, and reaches the maximum values at 0.9. Similar behavior for both the topologies is observed in the case of the random removal of nodes (see Fig. 7).

V. NUMERICAL DAMAGE DETECTION IN GEOLOCALIZED NETWORKS

In this section, we consider a sample of the real Internet topology network at the level of AS where each node is an autonomous system of known geographical location [10,21,38], and links represent the physical connections among them. Topologies are available for download in the DIMES project webpage [11]. We focus on the largest connected component of ASs that is made by 32 852 nodes.

We test damage detection in two relevant classes of realistic attacks that affect either all nodes in the same country, or all nodes inside a radius ξ with epicenter E. Both of these strategies are geography based, but they describe different scenarios. The first represents a deliberate shut-down such as what allegedly happened in several countries during the Arab spring [27]. The second one refers to localized events such as blackouts, earthquakes, or other catastrophic events [28]. Also, in this case, we fix the number of sources $N_S = 15$ and the density of targets $\rho_T = 0.2$. According to one of the two geographic based strategies, we remove N_D nodes from the original AS network. The main difference between this case and those discussed in the previous sections is that the networks here have geographical attributes. The measure of damage detection should then be able to return both the entity and the geographical location of the damage. We use the same method already discussed for synthetic networks. We decided to damage all of the AS nodes in Italy as an example of an entire country switch off. This translates to removing $N_D = 1246$ nodes, equal to a fraction $\rho_D = 0.038$ of total nodes. Figure 8 shows the outcome of our analysis. We want to stress that the algorithm does not have any *a priori* information about the location of the damage. Despite that, the method clearly returns Italy as an affected country. Few other nodes are wrongly classified as damaged. The reason for the presence of F_P can be due to statistical fluctuations or to some F_P nodes strongly linked to the Italian T_P so that the deletion of the latter prevents them from being visited. For the second type of geographical damage, we decide to switch off all the ASs within a radius of 50 km around the city of Boston, MA, in the USA. This corresponds to $N_D = 176$ and $\rho_D = 0.0054$. Also, in this case,

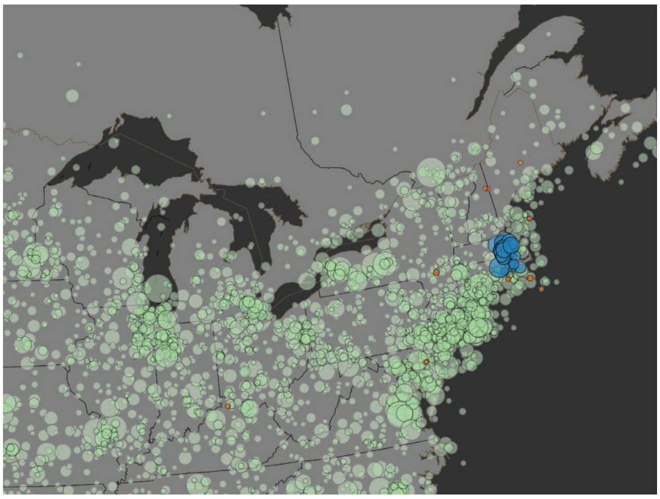

FIG. 9. (Color online) Map of part of the United States east coast, which shows the nodes in the real AS network after damaging the city of Boston within a radius of 50 km. Each circle represents one AS, and its size is proportional to the degree. Green circles are the working nodes, blue ones are the T_P, and orange ones are the F_P.

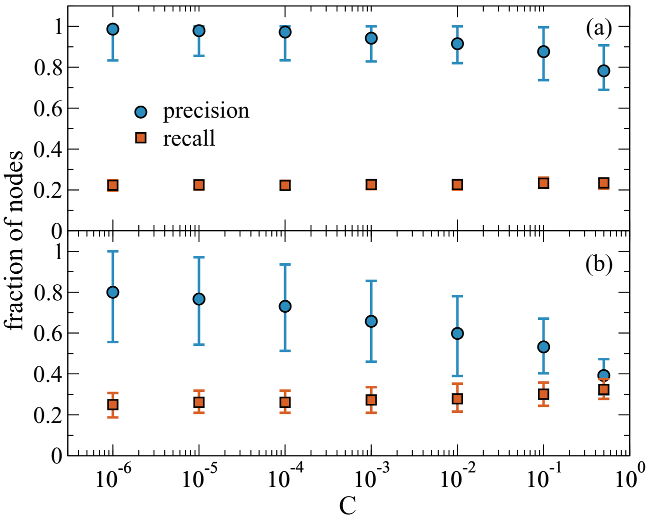

FIG. 10. (Color online) Precision α (blue circles) and recall r (orange squares) as a function of C for Italian nodes removal (a) and damaging of nodes around the city of Boston within a radius of 50 km (b) in the real AS network.

the method is able to detect the correct location of the damage as shown in Fig. 9. In both cases of geographical damaging, the recall is almost constant and close to the value 0.2 as shown in Fig. 10. Indeed, the algorithm is almost only detecting the fraction of damaged nodes that are also targeted. Because of the homogeneous distribution of target nodes, this fraction corresponds to $\rho_T = 0.2$. The comparison between Figs. 10(a) and 10(b) reveals that both precision and recall vary more with C when damaging local areas than for the shutdown of entire countries. In the first case, for big values of C, the precision drops while the recall slightly increases. This means that a less strict choice of C allows the discovery of more nodes. However, the F_P grow more than the T_P. So, a little gain in recall is contrasted by a big loss in precision.

As for the artificial networks, in the case of one country damaged, we choose C to achieve a precision of 0.9 ($C = 10^{-2}$). In the case of local damaging, there is no value of C that allows us to reach such a precision. For this reason, and considering the diverse nature of the two strategies, we fix the arbitrary value to $\alpha = 0.75$ ($C = 10^{-5}$). Although the recall never exceeds 0.3 in both damage detections, the method provides a good result considering the small number of damaged nodes, the completely random displacement of source and target nodes, and the lack of any *ad hoc* search strategy.

VI. CONCLUSIONS

In this paper, we addressed the problem of damage detection in large-scale networks. We assessed the effectiveness of shortest-path probing for damage detection in the case of incomplete network sampling. We considered different network topologies, damage strategies, and defined basic metrics for the measurement of damage. We provided a statistical criterion for the classification (damaged and undamaged) of single nodes based on the p-value test. Although this criterion allows false positives, i.e., nodes wrongly considered as damaged, it consents to fine-tune the statistical confidence level in order to optimize the trade-off between precision and probing load in the system. The numerical investigation according to this criterion allows the study of damages in partially sampled networks with tunable precision. In the case of real-world networks such as the Internet AS graph, we damaged the network according to geographical features simulating critical events on specific areas or deliberate shutdown of an entire country, as for political reasons. Also in this case our methodology is able to identify the entity of the damage and, more importantly, its location.

The method we have proposed can represent a first step towards a strategy for the continuous monitoring of large-scale, self-organizing networks. Possible variations of the shortest-path sampling can be envisioned and combined with more elaborate, diffusive walkers strategies that optimize network discovery. Furthermore, we have studied only the random displacement of sources and targets. Detection of damages could be improved by opportune choice of sources and targets or by a different schedule of probes delivery. This point will be addressed in future works.

ACKNOWLEDGMENTS

The authors thank Bruno Gonçalves for valuable discussions and the DIMES Project for providing autonomous system topology free to download. This work was partially funded by the DTRA-1-0910039 award to A.V.

[1] M. E. J. Newman, *Networks, An Introduction* (Oxford University Press, New York, 2010).

[2] G. Caldarelli, *Scale-Free Networks: Complex Webs in Nature and Technology*, OUP Catalogue Series (Oxford University Press, New York, 2007).

[3] A. Barrat, M. Barthèlemy, and A. Vespignani, *Dynamical Processes on Complex Networks* (Cambridge University Press, New York, 2008).

[4] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz, in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM '03* (ACM, New York, 2003), pp. 365–378.

[5] M. Luckie, Y. Hyun, and B. Huffaker, in *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement, Vouliagmeni, Greece* (ACM, New York, 2008), pp. 311–324.

[6] B. Huffaker, D. Plummer, D. Moore, and K. Claffy, in *Proceedings of the Symposium on Applications and the Internet (SAINT) Workshops, Nara, Japan* (IEEE, Piscataway, NJ, 2002), pp. 90–96.

[7] M. Luckie, in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, IMC '10* (ACM, New York, 2010), pp. 239–245.

[8] Van Jacobson, traceroute, ftp://ftp.ee.lbl.gov/traceroute.tar.gz

[9] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, IEEE/ACM Trans. Netw. **12**, 2 (2004).

[10] *The Cooperative Association for Internet Data Analysis*, *CAIDA*, http://www.caida.org/home/

[11] *The DIMES Project*, http://www.netdimes.org

[12] *Topology Project*, Electric Engineering and Computer Science Department, University of Michigan, http://topology.eecs.umich.edu/

[13] *The Internet Mapping Project at Bell Labs*, http://www.cheswick.com/ches/map/index.html

[14] *Rocketfuel: An ISP Topology Mapping Engine*, http://www.cs.washington.edu/research/networking/rocketfuel/

[15] *Routeviews Project*, http://www.routeviews.org

[16] T. Bu, N. Duffield, F. L. Presti, and D. Towsley, SIGMETRICS Perform. Eval. Rev. **30**, 21 (2002).

[17] L. Bing, *Web Data Mining* (Springer, New York, 2011).

[18] F. Menczer, G. Pant, and P. Srinivasan, ACM Trans. Internet Technol. **4**, 378 (2004).

[19] S. M. Mirtaheri, M. E. Dinçtürk, S. Hooshmand, G. V. Bochmann, G.-V. Jourdan, and I.-V. Onut, in *Proceedings of the 2013 Conference of the Center for Advanced Studies on Collaborative Research* (IBM, Armonk, NY, 2013), pp. 40–54.

[20] L. Dall'Asta, I. Alvarez-Hamelin, A. Barrat, A. Vàzquez, and A. Vespignani, Theor. Comput. Sci. **355**, 6 (2006).

[21] R. Pastor-Satorras and A. Vespignani, *Evolution and Structure of Internet: A Statistical Physics Approach* (Cambridge University Press, Cambridge, UK, 2004).

[22] K. Claffy, T. E. Monk, and D. McRobb, Nature (London) **7** (1999).

[23] Y. Shavitt and E. Shir, SIGCOMM Comput. Commun. Rev. **35**, 71 (2005).

[24] A. Abdelkefi, Y. Efthekhari, and Y. Jiang, arXiv:1209.5074 (2012).

[25] M. Catanzaro, M. Boguñá, and R. Pastor-Satorras, Phys. Rev. E **71**, 027103 (2005).

[26] P. Erdös and A. Rényi, Publ. Math. Inst. Hung. Acad. Sci. **5**, 17 (1960).

[27] A. Dainotti, C. Squarcella, E. Aben, K. C. Claffy, M. Chiesa, M. Russo, and A. Pescapé, in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference, IMC '11* (ACM, New York, 2011), pp. 1–18.

[28] V. S. A. Kumar, M. V. Marathe, R. Sundaram, M. Thakur, and S. Thulasidasan, *Proceedings are in Electronic Form.; 2006. ISMA 2006 WIT: Workshop on the Internat Topology.*, http://www.ccs.neu.edu/home/koods/papers/kumar06scaling.pdf

[29] A. Lakhina, J. W. Byers, M. Crovella, and P. Xie, Technical Report BUCS-TR-2002-021, Department of Computer Sciences, Boston University, 2002.

[30] T. Petermann and P. De Los Rios, Europhys. J. B: Condens. Matter Complex Syst. **38**, 201 (2004).

[31] D. Achilioptas, A. Clauset, D. Kempe, and C. Moore, J. ACM **56**, 21 (2009).

[32] Both sources and targets are randomly selected, so we can assume that on average, they will be visited N_T times if they are sources or N_S times if they are targets. Considering this and applying Eq. (1), we get $p_i = \sum_{j=1}^{N_S \times N_T} \delta_{i,j}/(N_S \times N_T) = \frac{N_T}{N_S \times N_T} = \frac{1}{N_S}$ and $p_i = \frac{N_S}{N_S \times N_T} = \frac{1}{N_T}$, respectively.

[33] The number N^* may differ from N_D^* also when no damage is present because of the not deterministic behavior of the shortest-path algorithm. As we wrote in Sec. II, we use the RPS probing strategy that randomly returns one of the possibly equivalent shortest paths, hence providing a different view of the same network.

[34] R. Albert and A.-L. Barabási, Rev. Mod. Phys. **74**, 47 (2002).

[35] G. Cowan, *Statistical Data Analysis* (Oxford University Press, New York, 1998).

[36] R. Cohen and S. Havlin, *Complex Networks: Structure, Robustness and Function* (Cambridge University Press, Cambridge, 2010).

[37] J. Platig, E. Ott, and M. Girvan, Phys. Rev. E **88**, 062812 (2013).

[38] A. Vázquez, R. Pastor-Satorras, and A. Vespignani, Phys. Rev. E **65**, 066130 (2002).