Improving Network Robustness by Edge Modification

Alina Beygelzimer [*], Geoffrey Grinstein, Ralph Linsker, Irina Rish

IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
{beygel,ggrin,linsker,rish}@us.ibm.com

Abstract

An important property of networked systems is their robustness against removal of network nodes, through either random node failure or targeted attack. Although design methods have been proposed for creating, ab initio, a network that has optimal robustness according to a given measure, one is often instead faced with an existing network that cannot feasibly be substantially modified or redesigned, yet whose robustness can be improved by a lesser degree of modification. We present empirical results that show how robustness, as measured either by the size of the largest connected component or by the shortest path length between pairs of nodes, is affected by several different strategies that alter the network by rewiring a fraction of the edges or by adding new edges. We find that a modest alteration of an initially 'scale-free' network can usefully improve robustness against attack, particularly when the fraction of attacked nodes is small, and we identify modification schemes that are most effective for this purpose.

Key words: Complex networks, scale-free networks, robustness, largest connected component
PACS: 89.75.Da, 89.75.Hc

1 Introduction

Recently there has been a surge of interest in trying to understand robustness properties of realistic networks. A number of authors have observed that complex networks with heavy-tailed statistics are highly robust against random failures of nodes, but are hypersensitive to targeted attacks against nodes with large degrees, i.e., with

[*] Corresponding author. Tel: 1+ (914) 784-6960. Fax: 1+ (914) 784-6071

many connections to other nodes (see, e.g., refs. [1, 2, 3, 4, 5, 6, 7]). In such systems, the degree distribution p_k, i.e., the probability of a node having k connections to other nodes, typically decreases as a power of k. We follow a common abuse of terminology and use the term 'scale-free' to mean having such a degree distribution. Intuitively, a randomly chosen node is likely to have a low degree, so its removal has little effect on the network. On the other hand, removal of a high-degree node can have a significant effect, since such a node may hold a large part of the network together by connecting many other nodes.

This situation is often compared to that for the classical random graphs of Erdős and Rényi [8], in which every pair of nodes is connected with a fixed probability p, independently of every other pair. We use $G(n, p)$ to denote such a random graph on n nodes. The graph $G(n, p)$ has a binomial degree distribution, $P_b(k)$, which approaches a Poisson distribution as n becomes large. The tail of this distribution decreases exponentially in k, making it unlikely to encounter a hub, i.e., a node whose degree significantly exceeds the mean degree, $p(n - 1)$. Thus, there is little difference between random failures and targeted attacks for $G(n, p)$. As observed in the references cited above, $G(n, p)$ is more vulnerable to 'random failures,' but less vulnerable to 'attacks' than are comparable scale-free graphs. So, relative to random graphs, real, heavy-tailed networks seem to incur a penalty in robustness against targeted attacks in exchange for an increased robustness against random failures. An obvious question is whether this tradeoff is unavoidable. (See ref. [9] for a general discussion of tradeoffs of vulnerabilities.)

Several groups of authors, e.g., Shargel et al. [10], Paul et al. [11], Valente et al. [12] have considered the problem of designing networks that optimize or avoid the tradeoff between the two types of robustness. For example, Shargel et al. [10] considered the role of two mechanisms – growth and preferential attachment – in the formation of networks, and found heuristically that networks with preferential attachment but no growth show both types of robustness. This is in agreement with results of Bollobás and Riordan [5], who proved that there is a precise sense in which the robustness of scale-free graphs against random failures comes from the preferential attachment rather than the growth mechanism. ('Preferential attachment' means simply that the probability that a node acquires a new connection is proportional to the node's degree.)

However, a real network is the result of many different processes that may have little to do with robustness against node deletions, and it may not be desirable, or even possible, to make a substantial change to the network in order to increase a particular measure of robustness. For example, in current decentralized, highly dynamic systems such as peer-to-peer networks, one no longer has explicit control over the structure. It thus seems important to explore *already existing* networks to see if they can be simply modified to improve robustness against attacks without appreciably degrading either the network's performance or its robustness against random failures.

In this paper, we study one class of such modifications, wherein either existing edges are randomly rewired to connect different pairs of nodes, or else new edges are added randomly to the network. Such random perturbations decrease the network's dependence on its hubs, making it more robust against degree-based attacks. We explore the robustness of the modified networks as a function of the number of rewired or added edges, and investigate which structural properties of networks determine how rapidly the robustness increases as edges are modified. As we shall see, the rewiring process allows us to explore the tradeoff between robustness against random failure and attack, by interpolating between (a) (scale-free) graphs whose degree distributions are heavy-tailed, and (b) graphs whose distributions either have exponentially decreasing tails or are nearly regular (i.e., have node degrees falling within a narrow range of values). Heuristically, our modification schemes can be thought of as overlaying either a classical random graph, or a nearly regular graph, having good robustness against attack, on top of a subgraph of the existing network, thereby increasing the robustness of the latter.

Two types of costs are incurred by modifying a network. First, there are costs associated with adding or rewiring edges. Second, if the network was designed to have particular properties for reasons unrelated to robustness, then a modification that substantially alters these properties (in particular, a degree distribution) may be unacceptably costly. These costs are application-specific and are not dealt with here. We instead show how certain measures of robustness depend upon both the extent of network modification and the type of modification procedure used. We also show how the character of the network (viz., its node degree distribution) varies as a function of the fraction of edges that are modified. We find that, especially for low levels of attack (a small fraction of nodes removed), a modest number of edge modifications can be effective in increasing robustness against selective attacks without decreasing robustness against random failures. At higher levels of attack, even a substantial modification of the network tends to have a limited effect on increasing robustness. At intermediate levels of attack, robustness is little changed by a modest amount of rewiring, and then increases rapidly as the extent of rewiring is further increased.

2 Models of Networks

In our numerical experiments, we use both real-world networks and networks generated using scale-free graph models. There are two different classes of such models. The first class, degree-based and static (i.e., not changing with time), is an extension of the classical random graph model to general degree sequences. One either: (a) fixes a degree sequence (i.e., the complete set of node degrees for the network), and then generates a random graph from the space of all graphs realizing this sequence; or (b) specifies the distribution from which the sequence is to be drawn, and then generates a random graph realizing a degree sequence drawn from that distribution. Option (b) is often easier to analyze. This first class of models attempts to simulate the structures of real-world networks without attempting to explain their origin. Nonetheless, such models capture various other properties of real data [13] quite well.

The second class of models seeks to explain the origin of network properties, by allowing the networks to grow in a way that reflects some evolutionary principles. The degree sequence is not specified, but rather emerges as the result of a random graph process. Perhaps the most studied model in this class is the preferential attachment model [14, 15]. The model is parameterized by two integers n and m: the first specifies the number of steps of the random graph process (which equals the number of nodes in the generated graph), while the second specifies the number of edges with which each new node connects itself to existing nodes. These existing nodes ('neighbors') are chosen randomly, with probability proportional to their degrees. The growth process starts, for example, with a single node with m loops. It was suggested [14, 1], and later proved [16], that such a process results in the power-law degree distribution $p_k \sim k^{-3}$. Subsequent variations yield degree distributions with variable exponents [17, 18, 19, 20].

Such growth models produce graphs with n nodes and approximately nm edges (ignoring self-loops), and hence an average degree of $2m$. The case $m = 1$ is not very interesting, as observed in ref. [5]; the resulting graph is a tree, and the removal of any constant fraction of nodes disconnects most pairs, leading to small components of size sublinear in n. Thus only $m \geq 2$, i.e., an average degree of at least 4, is relevant for our purposes. Most real networks of interest, however, tend to have smaller average degrees. To allow comparison with realistic networks, we use the following modification to control the expected number of edges: As a new node is added, it chooses its first random connection as described above, thereby guaranteeing that it is connected to the network. Each of the remaining $m - 1$ possible connections is made with some fixed probability (according to the same distribution as before).

Besides generated graphs, we experimented with sample crawls of the Gnutella [1] network (from limewire.com). One of the crawls is shown in Figure 1(a). (Figure 1(b) shows robustness results for this network, and will be discussed later.) We also used networks generated by the Inet-3.0 topology generator [21], which predicts the degrees of a graph on a given number of nodes by extrapolating from the Internet data (at the Autonomous System or interdomain level), and then heuristically generates a connected graph satisfying this predicted degree sequence. Inet seems to be the most authoritative Internet modeling method in the networking community, and has been shown to predict various properties of the Internet data

[1] Gnutella is one of the most successful open-source, peer-to-peer file sharing protocols. The crawls represent sufficiently accurate snapshots of small portions of the network in 2002.

quite well.

3 Failures, Attacks, and Measures of Robustness

To study robustness against (a) random failures and (b) targeted attacks, we consider graphs that, respectively, are damaged (a) by deleting each node independently with probability q; and (b) by deleting the qn nodes (i.e., the fraction q) having the highest degrees. The same basic definition of attack was used in ref. [2, 1, 4]. It has been observed that there is little difference between this type of attack and adaptive attacks that remove the highest degree node remaining in a graph after previous deletions.

If enough nodes are removed, the network may break into disconnected components. We use the fraction of nodes remaining in the largest connected component (LCC) as an indication of the network's availability after the attack. This quantity is our first measure of robustness.

As nodes are removed, the length of the shortest path connecting a pair of nodes tends to increase, impeding the transmission of information. It seems natural to use the average shortest path length (averaged over all pairs of nodes) to measure the effect of the removal on the performance. Since node removal may render some other nodes disconnected, causing the average path length to diverge, we actually use the average *inverse* shortest path length (AISPL) after the removal of nodes as our second measure of robustness. (Note that disconnected pairs contribute 0 to this quantity.) In order to distinguish between the effects of graph fragmentation and slower communication within a connected subgraph, we also use as measures the average shortest path length and the diameter (maximum shortest path length) of the LCC.

4 Network Modification Schemes

We consider several simple rewiring schemes and their variations. In all cases, we are interested in how robustness against attack increases as a function of the number of modified edges. 'Random' as used here means 'randomly chosen with uniform probability.' Duplication of an edge connecting the same pair of nodes is not allowed (a new random selection is made instead); however, allowing duplications has little effect on the results.

- **S1a: Random addition.** Add a new edge by connecting two random nodes.
- **S1b: Preferential addition.** Add a new edge by connecting two unconnected nodes having the lowest degrees in the network.

- **S2a: Random edge rewiring.** Remove a random edge, then add an edge as in S1a.
- **S2b: Random neighbor rewiring.** Choose a random node, and then a random neighbor of that node, and remove the corresponding edge. Then add an edge as in S1a.
- **S3a: Preferential rewiring.** Disconnect a random edge from a highest-degree node, and reconnect that edge to a random node.
- **S3b: Preferential random edge rewiring.** Choose a random edge, disconnect it from its higher-degree node, and reconnect that edge to a random node.

The effect of scheme S2a is to interpolate between the original graph and the classical random graph with the same number of nodes and edges. Note that the probability that a node is chosen for edge removal at each step is proportional to the degree of that node. Figure 2(a) shows (on a log-log plot) how the node degree distributions tend toward the Poisson distribution corresponding to the classical random graph, starting with an Inet-generated graph with 3500 nodes and 5667 edges, as edges are randomly rewired.

In scheme S2b, only the second endpoint of each removed edge is chosen with a bias favoring higher-degree nodes; the first endpoint is chosen uniformly at random. In this case, the asymptotic degree distribution after rewiring (not shown) also has an exponential tail, but experimentally it is flatter than the corresponding Poisson distribution.

In the rewiring limit, Scheme S3a approaches a random regular graph with degree distribution concentrated on the mean degree, or on the two consecutive integer values flanking the mean degree if the latter is not an integer. In practice we observed a rather slow convergence to this asymptotic distribution (see Figure 2(b)).

Scheme S3b has a stronger bias toward equalizing the degrees of all nodes than does scheme S2a. The resulting degree distribution (not shown) has an exponential tail, but is much more sharply concentrated around the mean degree than is the corresponding Poisson distribution.

Note that rewiring an edge can disconnect the network. We could simply disallow such rewirings, but the complexity of checking that the network remains in one piece is at least linear in the number of nodes. This may be undesirable, especially if a decentralized algorithm is needed. One solution is simply *not* to perform the checks. This would significantly simplify both the implementation and the analysis of the rewiring schemes. We found empirically that rejecting rewirings that disconnect the network has little effect on the outcomes of the various rewiring schemes.

Figure 3(a) compares schemes S2a and S2b for an Inet-generated graph with 3500 vertices. The topmost solid curve shows the fraction of nodes in the largest connected component versus the number of edges rewired using random edge rewiring (S2a). The topmost dotted curve corresponds to random neighbor rewiring (S2b).

Note that vertex loss in this case is caused only by the rewiring process. Most of this loss occurs in the early stages of the process. This is probably because random rewirings quickly randomize the structure of the network, making it less likely that further random rewiring can disconnect a sizable part of it, even if the original network was less stable in this sense. Also, realistic networks are very unlikely to have 'bridges' – edges whose removal partitions the network into large disconnected components – so the probability of hitting one randomly is very small, even initially.

Heuristically, the likelihood of disconnecting the network during rewiring is expected to be smaller when the rewiring scheme is biased toward the removal of connections for which at least one node has high degree. This suggests that schemes S3a and S3b would be expected to preserve network integrity during rewiring better than S2a, and that S2a would perform better than S2b (since in S2a both endpoints are subject to this bias, and in S2b only one of the endpoints is subject to it). This ordering agrees with our empirical results.

5 Findings

We present and analyze results comparing the effectiveness of the various modification schemes using two principal measures of robustness: the fraction of sites remaining in the largest connected component (LCC) after an attack (or node failure), and the average inverse shortest path length (AISPL) between pairs of nodes. To interpret the AISPL results for graphs that have disconnected components, we also consider the average shortest path length and the diameter of the LCC. Throughout this section, each curve shown represents averages over 10 different runs, typical fluctuations from run to run being significantly smaller than the difference between curves. The curves are also representative in the sense that similar behavior is observed for many different networks.

5.1 Largest Connected Component

Here we assess robustness as measured by the fraction of sites remaining in the largest connected component (LCC) after an attack, first for the rewiring and then for the addition schemes.

5.1.1 Rewiring Schemes

We compare the effectiveness of the proposed *rewiring* schemes, at each of four attack levels: 0% (no attack), 1%, 10%, and 20%. The results are obtained on an

Inet-generated graph with 3500 vertices.

Figures 3 and 4 display the results. In Figure 3(a), the solid curves (from top to bottom) show LCC size as a function of the extent of rewiring, for scheme S2a (random edge rewiring) and for each of the four attack levels starting at 0%. The dashed curves refer to scheme S2b (random neighbor rewiring). Scheme S2a is always superior to S2b. Figure 3(b) shows LCC size for schemes S3a (solid curves), S3b (bold dashed curves), and S2a (light dotted curves), for the three nonzero attack levels (reading from top to bottom). Scheme S3a (preferential rewiring) is consistently superior, especially at high attack levels, but it is more difficult (than S2a or S3b) to implement in a distributed fashion, since at every step it requires knowledge of the highest-degree node.

The order of superiority in improving LCC robustness is thus found to be: S3a $>$ (S2a and S3b) $>$ S2b. We account for this behavior as follows: S3a $>$ S2a and S3b $>$ S2b because preferential schemes remove edges from high-degree nodes faster than do random rewiring schemes. Scheme S3a $>$ S3b and S2a $>$ S2b for the same reason, since within each pair of schemes the first one is more biased than the second toward removing edges from high-degree nodes.

Figure 4 shows the effect of schemes S2a and S3a on robustness against random failures (the six topmost curves) for the same network as in Figure 3. None of the rewiring schemes significantly changes robustness against random failures, so the improvements in robustness against attack (re-plotted here for comparison) are achieved with little or no increased vulnerability to random failures.

Returning to robustness against targeted attacks, Figure 3 shows that there are several regimes of behavior, depending upon the severity of the attack. When a small fraction of nodes is removed – at most 3% for the 3500-node network considered here – the LCC size tends to increase roughly linearly as one begins to rewire edges. (The small-attack regime seems the most realistic for applications.)

For an intermediate-level attack (see, e.g., the 10% curves in Figure 3(b)), the LCC curves exhibit a sigmoidal behavior: they start out almost flat, but then sharply increase at an intermediate value of the rewiring fraction, before saturating as the rewiring fraction is further increased. Thus improved robustness is obtained only when a threshold in the number of rewired edges is exceeded, at which point a very rapid improvement can occur (e.g., solid curve in Figure 3(b)). For example, for preferential rewiring, the fraction of nodes remaining connected after the 10% attack of Figure 3(b) increases from less than 1% to more than 50% as the number of modified edges increases from zero to roughly half of the initial edges. The steep portion of these sigmoidal curves shifts to the right (i.e., occurs at a larger value of the rewiring fraction) as the attack level is increased (compare the 10% and 20% curves in Figure 3(b)).

Finally, when the severity of the attack exceeds a certain limit (which depends on the average degree of the original network, as shown below), the system passes into a ‘saturated’ regime, in which the robustness ceases to increase regardless of how many edges are rewired.

To account for this behavior, we speculate that the position of the steep portion of the sigmoidal curve is related to the percolation threshold of the partially rewired network: the more edges are rewired, the larger the percolation threshold of the resulting network. (The percolation threshold of a network is the attack level below which the network retains a ‘giant’ percolating cluster, with an LCC that is a significant fraction of the full network.) Thus, as the attack level is increased, more edges need to be rewired before we reach a distribution for which that attack level lies below the percolation threshold of the rewired network. For sufficiently low attack levels, a rapid increase in LCC size occurs even at the onset of rewiring (i.e., the leftmost flat portion of the sigmoid is absent). For sufficiently high attack levels, even a network that has had arbitrarily many edges rewired will have a percolation threshold lower than the attack level. In that case, no amount of rewiring will improve the robustness (this is the ‘saturated’ regime).

We have also compared the effect of rewiring on scale-free networks that have the same average degree, but whose degree distributions have different power-law exponents. We find, as expected, that networks having heavier-tailed degree distributions (exponents of lesser absolute value) have worse tolerance to attacks, both initially and during rewiring. We also find that the behavior of the preferential rewiring scheme (S3a) under severe attacks is rather complicated. The tradeoff curves are not necessarily monotone, i.e., additional rewiring can actually harm the network. However, such reversals typically happen only when many edges are rewired, so the non-monotonicity should not be a concern in practice.

5.1.2 Edge Addition

Figure 5 shows the increase in robustness against attack for random (scheme S1a) and preferential (S1b) edge additions. Preferential addition generally outperforms random addition, as one would expect. As in random rewiring, however, there is an intermediate regime of attack severity (seen in the 10% attack curves in Figure 5) in which the robustness curves for the two schemes cross, preferential addition eventually winning for sufficiently large numbers of added edges. Not surprisingly, additions increase robustness more effectively than rewirings. For comparison, note that adding a number of new edges equal to roughly half of the initial number increases attack robustness (the fraction of the largest connected component) from less than 1% to about 70-80% when 10% of the nodes are attacked.

5.2 Average Inverse Shortest Path Length

We turn now to the path length measure of robustness. Recall that increased robustness in this context corresponds to several related (though not identical) changes in the network: an increase in the average inverse shortest path length (AISPL), or a decrease in the average shortest path length or the diameter of the largest connected component.

5.2.1 Rewiring Schemes

Figure 6 shows the AISPL as a function of the number of rewired edges following attacks and random failures of 1%, 3%, and 5% for each rewiring scheme, as well as the curves for the case when no nodes are removed and all changes are due to the rewiring scheme itself. Here the starting network is a partial snapshot of Gnutella with 737 nodes and 803 edges. In the face of severe attacks (10% level or greater), rewiring turns out to be relatively ineffective at increasing the performance of networks (as measured by the AISPL), compared with the effect of rewiring on increasing network availability (as measured by the LCC).

There are three competing factors that influence the AISPL in a network that is being fragmented into disconnected components. First, each pair of nodes that belong to different components contributes zero to the AISPL; on this basis, a smaller degree of fragmentation (fewer fragments and/or a larger LCC) will tend to increase the AISPL. Second, a smaller component will tend to have shorter path lengths (and a larger AISPL); on this basis, a *greater* degree of fragmentation (producing smaller components) will tend to increase the AISPL. Third, the average path length within the LCC of the network changes as a result of the evolution of the degree distribution caused by rewiring. The overall behavior of the AISPL is thus the result of an interplay among these three competing effects. This competition operates in ways that differ in detail for each rewiring scheme. We will distinguish among the contributions of these two effects below.

We first consider the robustness against attacks, displayed in the bottom three curves of each figure. In the regime of modest rewiring, the AISPL increases with the number of rewired edges, as expected. Preferential rewiring (scheme S3a) is the most effective of the four schemes in increasing the AISPL here, just as it was for the LCC.

For large numbers of rewired edges, however, the random rewiring schemes S2a and S2b produce larger AISPL values than do the preferential schemes S3a and S3b. This is true even in the absence of node deletions (compare the asymptotic values of the top curves in each panel of Figure 6), and it is a consequence of the different asymptotic behaviors of the rewiring schemes: Random rewiring leads to classical random graphs, while preferential rewiring leads to nearly regular graphs (whose degree distribution is concentrated about the average degree).

As shown in Figure 6(c), the AISPL for scheme S3a behaves non-monotonically as the number of rewired edges increases. This behavior is explained by considering the interplay between two of the effects mentioned above. For small rewirings, the preferential rewiring scheme S3a does best because, as we saw earlier, it tends to produce the most rapid growth in the largest connected component (see, e.g., Fig. 3(b)). As the extent of rewiring increases, the size of this largest component tends to saturate, at which point the behavior of the AISPL becomes dominated by the effect of further rewiring on the average path length within the largest component. Now we have seen that in the absence of attack, the asymptotic degree distribution for scheme S3a has support only for the two consecutive integers straddling the mean degree, which is approximately 2.2 for the network studied here. Thus approximately 80% of the nodes of the asymptotic distribution have degree 2, the rest having degree 3. This implies a network topology consisting of relatively long strings of nodes of degree 2 connecting nodes of degree 3 – a structure whose average path length will be larger than that of the original network, which contains hubs. Thus we expect the average path length of the largest connected cluster to increase with further rewiring, meaning that the AISPL should decrease as rewiring proceeds.

We turn now to the upper four curves in each panel of Figure 6, which illustrate the decreased robustness against random failures produced by rewiring. In each figure, these four upper curves are roughly parallel, demonstrating that, at least for the relatively modest attacks considered here, the reduction in AISPL produced by rewiring is nearly independent of the level of random failure: Virtually all of the decrease is already present in the 0% curve. Again, this decrease results from the interplay between the fragmentation of the network caused by rewiring and the effect of rewiring on the average path length of the largest connected component.

For the same initial network, we also studied the effect of rewiring on the diameter of the largest connected component (LCC). Like the average path length, this is a measure of the efficiency of communication between nodes within a cluster, small diameters obviously being desirable. One expects and finds that the diameter and average path length behave similarly. Figure 7 show plots of the ratio of the diameter of the LCC to the number of nodes in the LCC, as a function of the number of rewired edges, for the four rewiring schemes. This ratio compensates, albeit crudely, for the increase of diameter with increasing numbers of nodes in the LCC. Lower ratios, of course, imply better communications within the cluster.

In these figures, the top three curves represent (from top to bottom) attacks of 5%, 3%, and 1%, while the bottom four curves represent random failures of 5%, 3%, 1%, and 0%. In every case, these lower four curves are clustered very closely together – sometimes indistinguishably so – showing that the ratio of diameter to largest cluster size is largely unaffected by modest numbers of failures. For the random schemes, S2a and S2b, the four curves are virtually horizontal, implying a roughly linear increase of diameter (or average path length) with LCC size. In schemes S3a and S3b, on the other hand, the four curves are roughly linear with small but nonzero slope. Since the LCC sizes (corresponding to the 0% failure case) are essentially independent of rewiring, this implies that the cluster diameters increase roughly linearly with the number of rewired edges as the degree distributions of the clusters evolve towards their asymptotic limits.

The three upper curves in all of these figures show an initial decrease, quite sharp for the 3% and 5% curves, as edges are rewired. This is a result of the LCC growing more rapidly with rewiring than the diameter (or average path length), and is another measure of the effectiveness of rewiring in mitigating the consequences of attack. The rate of decrease in the ratio slows as rewiring proceeds, reflecting the interplay of effects that we saw in the figures for the AISPL. For scheme S3a (and for the 1% curve of scheme S3b), the ratios begin to increase as enough edges are rewired. As in the nonmonotonic AISPL curves for scheme S3a, this reversal results from the saturation of the LCC size, and the increase in diameter associated with the evolution of the degree distribution towards its limit, which includes no large hubs.

To summarize, preferential rewiring (S3b) was found to be the most efficient in increasing the AISPL for modest rewirings, but the random edge (S2a) and random neighbor (S2b) schemes yielded the largest AISPL values when larger numbers of edges were rewired. Not surprisingly, none of these schemes performed as well as random addition.

5.2.2 Edge Addition

Figure 1(b) shows the AISPL following removal of the 5% highest-degree nodes, as a function of the number of random edges added. The three curves correspond to three networks, each having 435 nodes and 459 edges. The dotted curve that lower-bounds the other two curves corresponds to the partial Gnutella topology in Figure 1(a). The other dotted curve corresponds to a graph with the same degree sequence but optimized to withstand the type of attacks considered here; this is achieved by carefully ensuring that there exist alternative “chains” between nodes connected through hubs. The solid curve corresponds to a scale-free graph with a heavier tail, but the same number of nodes and edges as the first two graphs. Notice that the AISPL is fairly small because the graphs are very sparse. By the time the number of edges has increased by 33% (or 150 edges), this performance measure has increased roughly four-fold.

A question raised by Figure 1(b) is whether the shape of the robustness curves depends on the structure of the graph for a fixed degree distribution. For example, is a random graph with a given degree sequence more robust against attacks than a grown graph realizing the same degree sequence? To explore this question, we started with our original connected graphs and ran the following Markov chain to generate a random graph with the same degree sequence (see [22]). A step of the Markov chain picks two random edges (u, v) and (x, y) with distinct end points, and cross-wires them, i.e. replaces them with (u, y) and (x, v), unless at least one of (u, y), (x, v) is already an edge, in which case nothing is done. It also does not do the switching if removing (u, v) or (x, y) disconnects the graph. The above Markov chain walks on the space of all connected realizations of the original degree sequence, and it is known to converge to the uniform distribution on this space. (generating a random graph with the given degree sequence). Empirically, we find that the differences in robustness within the class of graphs with the same degree sequence seem to vanish after some critical number of random edges has been added.

6 Analysis

In this section, we analyze degree distributions of intermediate graphs generated by random rewiring. Let $n_k(t)$ be the expected number of nodes with degree k at time t. At each step, $n_k(t)$ (for non-boundary k) will increase by an amount proportional to the probability that a degree $k+1$ node loses an edge (becoming a degree k node), and decrease by an amount proportional to the probability that a degree k node loses an edge at that step. There is also an influx of $2n_{k-1}(t)/n$, since a node of degree $k-1$ can acquire an edge. There is a similar outflow to account for nodes of degree k that acquired an edge. Thus we have

$$n_k(t+1) = n_k(t) + \frac{2n_{k-1}(t)}{n} - \frac{2n_k(t)}{n} + \frac{(k+1)n_{k+1}(t)}{nm} - \frac{kn_k(t)}{nm}.$$

For random addition, we have only the first three terms, together with the boundary conditions: $n_1(t+1) = n_1(t)(1-2/n)$ and $n_{n-1}(t+1) = n_{n-1}(t)$. Letting $p = 2/n$, we immediately have

$$\begin{aligned} n_1(t) &= n_1(0)(1-p)^t, \\ n_2(t) &= n_2(0)(1-p)^t + n_1(0)p(1-p)^{t-1}, \end{aligned}$$

and it is not hard to see that the general solution is

$$n_k(t) = \sum_{i=0}^{k-1} n_{k-i}(0)\binom{t}{i}(1-p)^{t-i}p^i. \tag{1}$$

Notice that approximating by the sum of Poisson distributions is not suitable since we are interested in finite values of t. In the limit of large t, we will obviously get the complete graph on n nodes with multiple edges.

If the original network is very large, so that it is undesirable to simulate the addition process in order to compute the robustness curve, the impact of addition can be predicted numerically using the explicit form of intermediate degree distributions given in Equation (1) and the approach of Callaway et al. [2]. In particular, ref.[2] gives a numerical method for computing the percolation threshold for a network with a given degree distribution. The method allows one to remove vertices from the network in any order that depends on their degree (including removal in order of decreasing degree and random failure).

7 Applications

Our approach can be used, in principle, for the design of communication protocols that provide globally robust topologies by performing local operations in highly decentralized, dynamic systems. The techniques are applicable in any scenario that supports edge modifications. The best example is the class of peer-to-peer networks which create logical topologies on top of the physical Internet. Rewiring an edge in such a network amounts, for example, to closing one and establishing another TCP connection between a pair of peers. Edge additions may also be meaningful in some social networks.

The approach could also be used for the Web graph. Of course, rewiring or adding explicit links on web pages is unreasonable, but it may be possible to add hidden links visible only to Web crawlers. Such a random hidden web on top of the visible Web graph may be used to guarantee that large parts of the Web graph are not lost if the graph is attacked.

We believe that the proposed approach is feasible, particularly to protect against modest attacks of less than a few percent node removal, in which case the fractional increase in robustness can exceed the fraction of modified edges. It would also be likely to improve other network properties, in particular the speed with which information propagates through the network. Randomization has already been successfully used in the design of peer-to-peer systems (see, e.g., [23]). The protocol for joining the network in [23] basically consists of connecting to a randomly chosen existing peer. Uniform sampling from the set of peers can be simulated by a random walk (of constant length) on the network topology [24], provided that the network has sufficiently good connectivity properties. Sampling also becomes easier as more edges are modified: the more random edges, the better the expansion (graph parameter quantifying the rate at which information propagates through the network), and the easier it is to sample a random peer, even if the initial expansion was not good. Another sampling approach based on random walks was proposed by Henzinger et al. [25] in the context of sampling URLs uniformly at random from the web graph. Thus random edge modifications can be implemented in a highly decentralized, scalable manner.

8 Known Results and Their Relation to This Paper

There has been considerable interest in understanding the robustness properties of networks and their dependence on degree distribution. In particular, much effort has been focused on obtaining q_c, the largest value of q for which there exists a constant $c = c(q) > 0$, dependent only on q, such that with high probability (i.e., probability approaching 1 as the number of nodes n goes to infinity), the graph remaining after the removal of qn nodes still has a component of size at least cn. The quantity q_c, familiar from studies of classical random graphs, corresponds to the critical probability in percolation; q_c need not exist, i.e., a giant component may exist for all values of $q < 1$. In fact, Callaway et al. [2] and Cohen et al. [3] showed that in the case of random deletions, there is no critical probability for random graphs having power-law degree distributions $p_k \sim k^{-\alpha}$ with $\alpha \leq 3$ (which is true for most networks of interest). The result actually holds for any distribution with a diverging second moment. The generating-function methods of ref. [2] are general enough to analyze the removal of nodes in any order that depends on their degree – in particular, in order of decreasing degree.

Bollobás and Riordan [5] rigorously proved the absence of the critical probability for random failures in the preferential attachment model (with $m > 1$), rather than in the degree-based model. They showed that for any $0 < q < 1$, a random deletion of qn nodes from such an n node graph leaves a giant component of order $c(q)n$ for some constant $c(q) > 0$. In contrast, a comparable classical random graph with connection probability $p = 2m/(n-1)$ has a well-understood threshold probability: a giant component exists if and only if the expected degree $p(1-q)(n-1)$ in the remaining graph is greater than one, implying the critical probability $q_c = 1 - 1/2m$. Thus, as proved in ref. [5], scale-free graphs in the preferential attachment model are infinitely more robust to random deletions than comparable random graphs. However, the constants $c(q)$ are so small for severe failures (corresponding to q approaching 1) that the "giant" component may not be seen in practice.

All the above papers also concluded that scale-free random graphs are more vulnerable to attacks than classical random graphs. Results of Cohen et al. [4] and Callaway et al. [2] apply for the degree-based model. Bollobás and Riordan [5] proved, for the preferential attachment growth process, that for *any* choice of qn nodes, there is a threshold probability $q_c = 1 - 2/(m+1)$. More precisely, if $q \geq q_c$, then with high probability there is no giant component. Otherwise there is a giant component of size roughly at least $O(q)n$.

9 Summary, Discussion, and Conclusions

We have demonstrated empirically that simple randomization of existing networks increases their robustness against attacks on the largest-degree nodes, without appreciably decreasing their robustness against random node failure. We found that for attacks limited to a few percent of the largest nodes (probably the most likely regime for realistic networks), the robustness initially increases roughly linearly, sometimes fairly steeply, with the number of rewired nodes. In such cases a modest amount of randomness can noticeably improve network availability and performance. For more severe attacks, we have seen that little benefit is obtained by randomization until the number of rewired edges exceeds a threshold of roughly 1/3 to 1/2 the original number of edges (hardly a modest amount of modification), depending on the rewiring scheme and network. Above this threshold (which is related to the percolation threshold of the graph produced by the rewirings), improvement in robustness can be sudden and quite dramatic. For large enough attacks (exceeding 20%, very roughly), the robustness can simply saturate as rewiring proceeds, saturation sometimes setting in so quickly that rewiring offers virtually no protection against attack.

Though there were some exceptions in the intermediate regime of attack severity, we found that the typical order of performance of the different rewiring schemes using the LCC measure of robustness was, from best to worst: preferential rewiring (S3a), preferential random edge rewiring (S3b), random edge rewiring (S2a), and random neighbor rewiring (S2b). Not surprisingly, random addition was superior to rewiring, though it increases the average degree. Preferential addition (S1b), superior to random addition (S1a), was the overall champion for robustness enhancement. As discussed earlier, this order of performance is easy to understand heuristically. The situation is more complicated for the AISPL measure of robustness, though the random addition schemes (S1a and S1b) remained superior to the rewiring schemes. Among the latter, preferential rewiring (S3a) was best for small rewirings, though the purely random schemes (S2a) and (S2b) were the ultimate winners.

The superiority of preferential rewiring to other schemes shares a qualitative feature with the findings of Valente et al. [12]; namely, that the preferred node degree distribution in each case is concentrated, with each node having a degree that takes one of only two values. Subject to a fixed average degree and a given constraint on the smallest and largest degrees, Valente et al. showed that the degree distribution maximizing the percolation threshold for attacks is a two-point distribution in which any node has either the smallest degree or some intermediate degree (smaller than the upper cut-off), with the partition between these two degrees determined by the value of the average degree. The value of the larger degree varies depending on the level of attack. Ref. [12] provides only a numerical solution for the second value. Recall that the stationary distribution of our preferential wiring scheme is concentrated on two consecutive integers which bracket the average degree (typically not an integer); the partition between these two values is determined by the average degree in the obvious way. A better understanding of the optimal 2-point distribution for a given level of attack and a given average degree may suggest other rewiring schemes that would concentrate all degrees on an optimal pair of values.

Of course, different node-removal strategies can have different effects, so it is possible that in optimizing the network for withstanding a specific, obvious attack strategy, one may inadvertently increase the network's vulnerability to other modes of attack [9]. Even if this is the case, however, one might hope that exploiting these new vulnerabilities will require detailed knowledge of the global structure of the network. Such knowledge will be increasingly hard to acquire as the complexity of real networks continues to grow. Hence, basing attacks on such local information as node degrees may be the most sensible approach for the adversary. Even though one can easily construct networks in which, say, some low-degree nodes are 'central' to how the network functions (e.g., link together different, almost disjoint parts of the network), locating such nodes (even if they exist in practice), would require some global knowledge about the role nodes play in the existence of paths between other nodes, and such information may not be easy to infer.

References

[1] R. Albert, H. Jeong, and A.-L. Barabási. Error and attack tolerance of complex networks. *Nature*, 406:378–382, 2000.

[2] D. Callaway, M. Newman, S. Strogatz, and D. Watts. Network robustness and fragility: Percolation on random graphs. *Phys. Rev. Letters*, 85:5468–5471, 2000.

[3] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin. Resilience of the internet to random breakdown. *Phys. Rev. Letters*, 85:4626–4628, 2000.

[4] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin. Breakdown of the internet under intentional attack. *Phys. Rev. Letters*, 86:3682–3685, 2001.

[5] B. Bollobás and O. Riordan. Robustness and vulnerability of scale-free random graphs. *Internet Mathematics*, 1(1):1–35, 2003.

[6] B. Bollobás and O. Riordan. Coupling scale-free and classical random graphs. *Internet Mathematics*, 1(2):215–225, 2004.

[7] A. Broder, S. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, J. Wiener. Graph structure in the Web. *Proceedings of the 9th International World Wide Web conference*, 309–320, 2000.

[8] P. Erdős and A. Rényi. On the evolution of random graphs. *Bull. Inst. Internat. Statist.*, 38:343–347, 1961.

[9] J. M. Carlson and J. Doyle. Highly optimized tolerance. *Phys. Rev. E*, 60:1412–1427, 1999.

[10] B. Shargel, H. Sayama, I. Epstein, and Y. Bar-Yam. Optimization of robustness and connectivity in complex networks. *Phys. Rev. Letters*, 90(6):068701, 2003.

[11] G. Paul, T. Tanizawa, S. Havlin, and H. Stanley. Optimization of Robustness of Complex Networks. *Euro. Phys. J. B* 38, 187–191, 2004.

[12] A. Valente, A. Sarkar, and H. Stone. 2-Peak and 3-Peak optimal complex networks. *Proceedings of the 5th International Conference on Complex Systems (ICCS)*, 2004.
[13] H. Tangmunarunkit, R. Govindan, S. Jamin, and S. Shenker. Network topology generators: Degree-based vs. structural. *ACM SIGCOMM*, pages 147–159, 2002.
[14] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.
[15] B. Bollobás and O. Riordan. The diameter of a scale free random graph. *Combinatorica*, 4:5–34, 2004.
[16] B. Bollobás, O. Riordan, J. Spencer, and G. Tusnády. The degree sequence of a scale-free random graph process. *Random Structures and Algorithms*, 18:279–290, 2001.
[17] E. Drinea, M. Enachescu, and M. Mitzenmacher. Variations on Random Graph Models for the Web. Technical Report TR-06-01, Harvard University, Computer Science Group, Cambridge, MA, 2001.
[18] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic Models for the Web Graph. *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science (FOCS)*, 57–65, 2000.
[19] S. Dorogovtsev, J. Mendes, and A. Samukhin. Structure of growing networks with preferential linking. *Phys. Rev. Letters*, 85, 4633, 2000.
[20] D. J. de S. Price. A general theory of bibliometric and other cumulative advantage processes. *J. Amer. Soc. Inform. Sci.*, **27**, 292–306, 1976.
[21] J. Winick and S. Jamin. Inet-3.0: Internet topology generator. Tech. Report CSE-TR-456-02, Univ. of Michigan, 2002.
[22] C. Gkantsidis, M. Mihail, and E. Zegura. The Markov Chain Simulation Method for Generating Connected Power Law Random Graphs. Alenex, 2003.
[23] C. Law and K.-Y. Siu. Distributed construction of random expander networks. In *IEEE Infocom*, 2003.
[24] C. Gkantsidis, M. Mihail, and A. Saberi. Random walks in peer-to-peer networks. In *Infocom*, 2004.
[25] M. Henzinger, A. Heydon, M. Mitzenmacher, M. Najork. On Near-Uniform URL Sampling. *Proceedings of the 9th International World Wide Web Conference*, 2000.

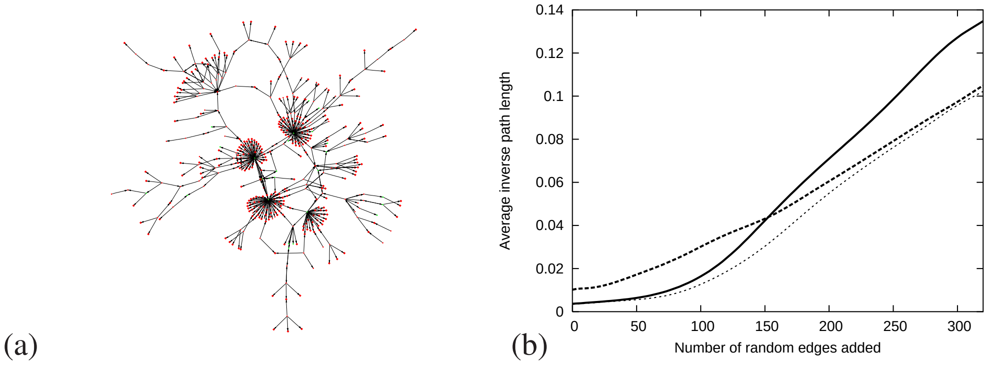

Fig. 1. (a) A sample crawl of the Gnutella network (435 nodes), (b) Average inverse shortest path length versus the number of added edges, for three networks with 435 nodes and 459 edges, but with some differences in their topologies (see text). The curve corresponding to the network in (a) lower-bounds the other two curves.

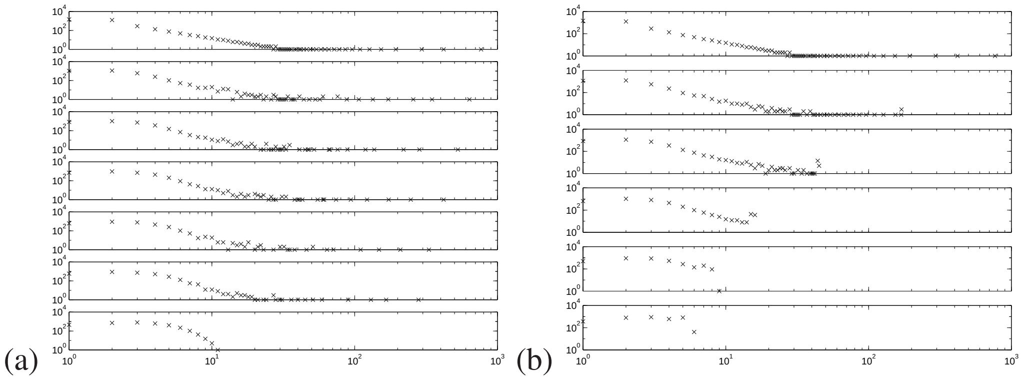

Fig. 2. (a) Evolution of the node degree distributions (log-log plot), starting with an Inet-generated graph with 3500 nodes and 5667 edges, using random edge rewiring (scheme S2a). Abscissa denotes the node degree, ordinate denotes the number of nodes having that degree. From top to bottom: the original network; the network after random rewiring of 1000, 2000, 3000, 4000, and 5000 edges; the corresponding Poisson distribution for comparison. (b) Similar to (a), except that preferential rewiring (scheme S3a) is performed. From top to bottom: the original network; and the network after preferential rewiring of 1000, 2000, 3000, 4000, and 5000 edges.

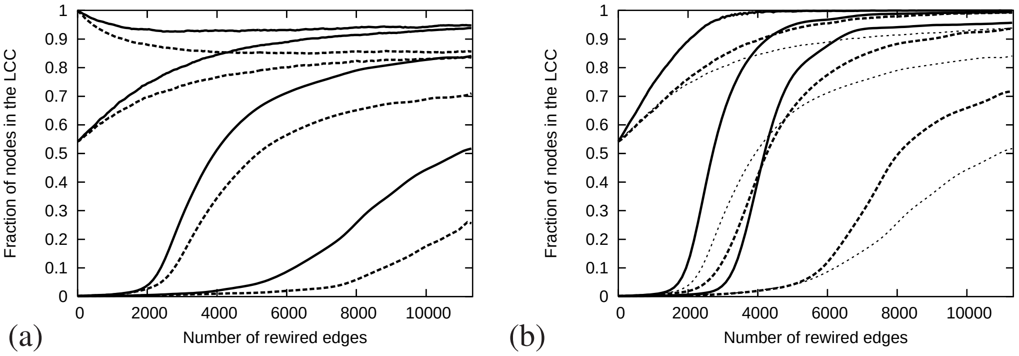

Fig. 3. (a) An Inet-generated network with 3500 nodes and 5667 edges. Comparison of random edge rewiring (scheme S2a, solid curves) and random neighbor rewiring (scheme S2b, dashed curves). Starting from the topmost pair of curves, the plots show the fraction of nodes in the largest connected component, versus the number of rewired edges after, respectively, 0% (none), 1%, 10%, and 20% of nodes (in decreasing order of degree) are attacked. (b) A similar comparison of preferential rewiring (scheme S3a, solid curves), preferential random edge rewiring (S3b, bold dashed curves), and random edge rewiring (S2a, thin dotted curves). Starting from the topmost curves, the plots correspond to 1%, 10%, and 20% attacks. Each curve is an average over 10 runs, and no connectivity checks were performed (see text).

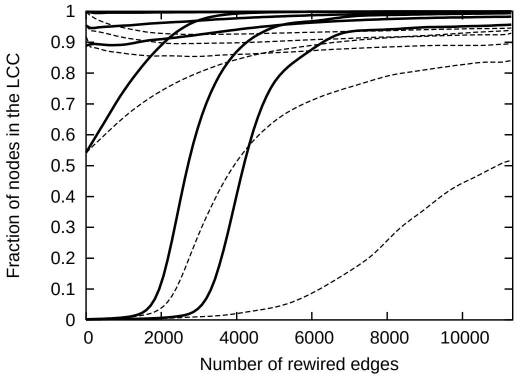

Fig. 4. Robustness against random failures and attacks for the network of Figure 3. Solid lines correspond to preferential rewiring (scheme S3a), dashed lines to random edge rewiring (S2a). The topmost horizontal pairs of curves correspond to random failures of 1%, 10%, and 20% of the nodes, respectively. The other three pairs of curves correspond to attacks on 1%, 10%, and 20% of the nodes (from left to right).

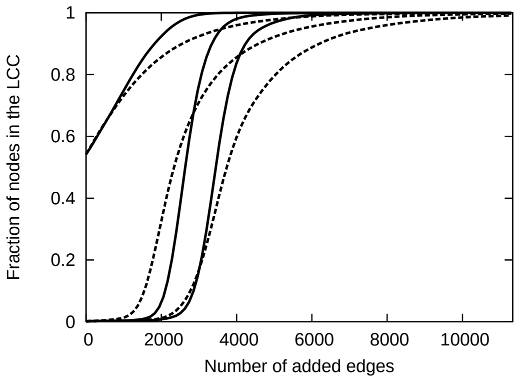

Fig. 5. Robustness against attacks for the network of Fig. 3 for preferential addition (S1b, solid curves) and random addition (S1a, dashed curves) schemes. Robustness against random failures is not shown since edge additions do not decrease it.

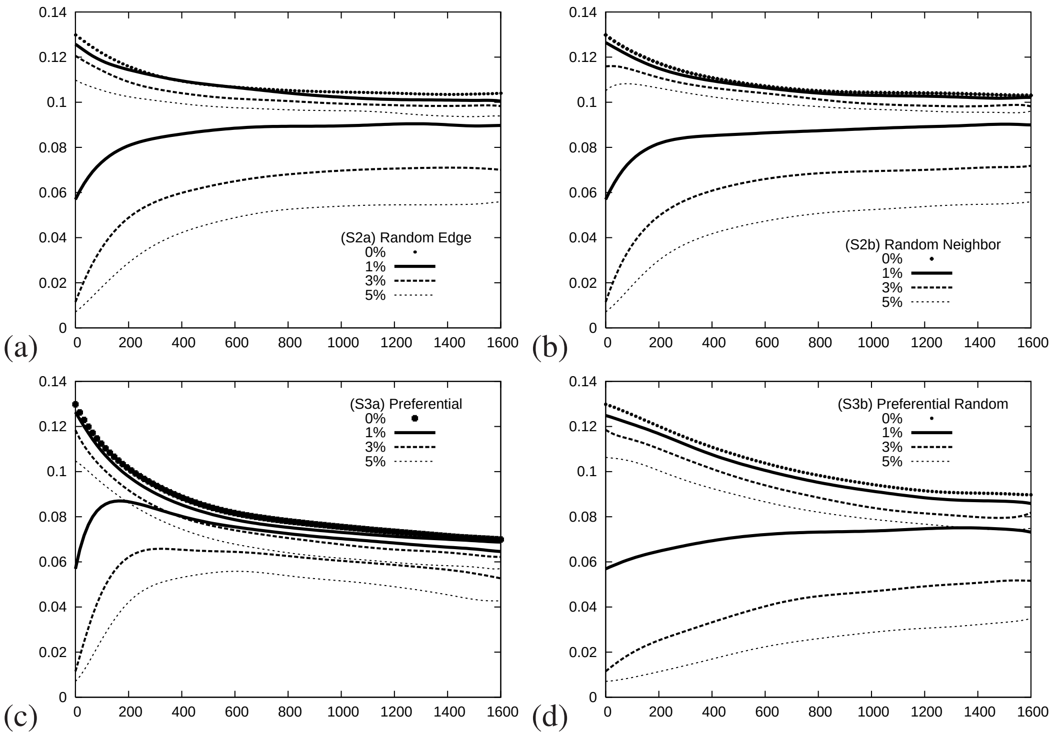

Fig. 6. Average inverse shortest path length (AISPL) versus number of rewired edges, for each rewiring scheme. In each panel, the seven curves (in order, at the left edge, from top to bottom) show AISPL for: no random failure or attack; random failures of 1%, 3%, and 5%; and attacks of 1%, 3%, and 5%. (a) Random edge rewiring (scheme S2a). (b) Random neighbor rewiring (scheme S2b). (c) Preferential rewiring (scheme S3a). (d) Preferential random edge rewiring (scheme S3b).

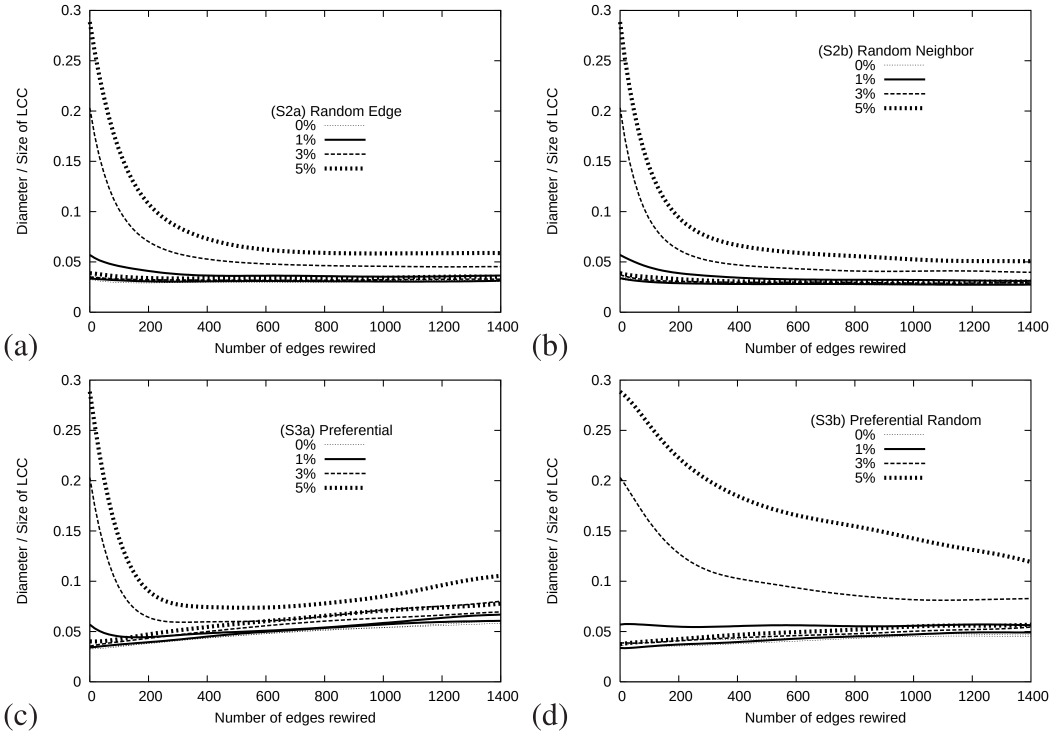

Fig. 7. Ratio of diameter of largest connected component (LCC) to number of nodes in LCC versus number of rewired edges, for each of our rewiring schemes. In each panel, the seven curves (in order, at the left edge, from bottom to top) show AISPL for: no random failure or attack; random failures of 1%, 3%, and 5%; and attacks of 1%, 3%, and 5%. The bottom four curves in each panel are clustered closely together. (a) Random edge rewiring (scheme S2a). (b) Random neighbor rewiring (scheme S2b). (c) Preferential rewiring (scheme S3a). (d) Preferential random edge rewiring (scheme S3b).