

Percolation on Sparse Networks

Brian Karrer,¹ M. E. J. Newman,¹ and Lenka Zdeborová²

¹*Department of Physics, University of Michigan, Ann Arbor, Michigan 48109, USA*

²*Institut de Physique Théorique, CEA Saclay and URA 2306, CNRS, 91191 Gif-sur-Yvette, France*

(Received 12 June 2014; published 12 November 2014)

We study percolation on networks, which is used as a model of the resilience of networked systems such as the Internet to attack or failure and as a simple model of the spread of disease over human contact networks. We reformulate percolation as a message passing process and demonstrate how the resulting equations can be used to calculate, among other things, the size of the percolating cluster and the average cluster size. The calculations are exact for sparse networks when the number of short loops in the network is small, but even on networks with many short loops we find them to be highly accurate when compared with direct numerical simulations. By considering the fixed points of the message passing process, we also show that the percolation threshold on a network with few loops is given by the inverse of the leading eigenvalue of the so-called nonbacktracking matrix.

DOI: 10.1103/PhysRevLett.113.208702

PACS numbers: 89.75.Hc, 64.60.ah, 05.70.Fh

Percolation, the random occupation of sites or bonds on a lattice or network with independent probability p , is one of the best-studied processes in statistical physics. It is used as a model of porous media [1,2], granular and composite materials [3–6], resistor networks [7], forest fires [8], and many other systems of scientific interest. In this Letter we study the bond (or edge) percolation process on general networks or graphs, which is used to model the spread of disease [9,10] and network robustness [11–13] in social and technological networks, among other things. Although percolation has been studied extensively on simple model networks such as random graphs [11,12,14,15], there are few analytic results for real-world networks, whose structure is typically more complicated. We show that percolation properties of networks can be calculated using a message passing technique, leading to a range of new results. In particular, we derive equations for the size of the percolating cluster and the average size of nonpercolating clusters, which can be solved rapidly by numerical iteration given the structure of a network and the value of p . By expanding the message passing equations about the critical point we also derive an expression for the position of the percolation threshold, showing that the critical value of p is given by the inverse of the leading eigenvalue of the so-called nonbacktracking matrix [16,17], an edge-based matrix representation of network structure that has found recent use in studies of community detection and centrality in networks [17,18]. The quantities we calculate are averages over all possible realizations of the randomness inherent in the percolation process, rather than over a single realization, obviating the need for a separate average over realizations as is typically required in direct numerical simulations.

We focus in particular on sparse networks, those for which only a small fraction of possible edges are present,

which includes most real-world networks. Our results are exact for large, sparse networks that contain a vanishing density of short loops, but even for networks that do contain loops, as most real-world networks do, we find the cluster size calculations to be highly accurate and the threshold calculations can be shown to give a lower bound on the true percolation threshold.

Consider, then, a bond percolation process on an arbitrary undirected network of n nodes and m edges. Edges are occupied uniformly at random with probability p or unoccupied with probability $1 - p$. The primary entities of interest are the percolation clusters, sets of nodes connected by occupied edges. Since percolation is a random process, one cannot know with certainty the identity of the clusters ahead of time, but some things are known. In general there will (with high probability) be at most one percolating cluster, a cluster that fills a nonvanishing fraction of the network in the limit of large n , plus an extensive number of small clusters of finite average size. The percolating cluster appears only for sufficiently large values of p and the percolation threshold p_c is the value above which it appears; below p_c there are only small clusters.

Define $\pi_i(s)$ to be the probability that node i belongs to a small cluster of exactly s nodes, averaged over many realizations of the random percolation process. If the network is a perfect tree—if it contains no loops—then the size s of the cluster is equal to 1 (for node i itself) plus the sum of the numbers of nodes reachable along each edge attached to i , which is zero if the edge is unoccupied or nonzero otherwise. If, on the other hand, there are loops in the network then this calculation will not, in general, give the exact value of s , since it may be possible to reach the same node along two different occupied edges, which leads to overcounting. If the network is sparse, however, and

locally treelike, meaning that in the limit of large network size an arbitrarily large neighborhood around any node takes the form of a tree (and hence contains no loops), then our calculation gives a good approximation, which becomes exact in the $n \rightarrow \infty$ limit.

Working in the large n limit then and assuming the network to be locally treelike, the probability $\pi_i(s)$ is equal to the probability that the numbers of nodes reachable along each edge from i add up to $s - 1$, which we can write as

$$\pi_i(s) = \sum_{\{s_j: j \in \mathcal{N}_i\}} \left[\prod_{j \in \mathcal{N}_i} \pi_{i \leftarrow j}(s_j) \right] \delta\left(s - 1, \sum_{j \in \mathcal{N}_i} s_j\right), \quad (1)$$

where $\pi_{i \leftarrow j}(s)$ is the probability that exactly s nodes are reachable along the edge connecting i and j , \mathcal{N}_i is the set of immediate network neighbors of node i , and $\delta(a, b)$ is the Kronecker delta function.

We now introduce a probability generating function $G_i(z) = \sum_{s=1}^{\infty} \pi_i(s) z^s$, whose value is given by

$$\begin{aligned} G_i(z) &= \sum_{s=1}^{\infty} z^s \sum_{\{s_j: j \in \mathcal{N}_i\}} \left[\prod_{j \in \mathcal{N}_i} \pi_{i \leftarrow j}(s_j) \right] \delta\left(s - 1, \sum_{j \in \mathcal{N}_i} s_j\right) \\ &= z \prod_{j \in \mathcal{N}_i} \sum_{s_j=0}^{\infty} \pi_{i \leftarrow j}(s_j) z^{s_j}, \end{aligned} \quad (2)$$

which can be conveniently written as

$$G_i(z) = z \prod_{j \in \mathcal{N}_i} H_{i \leftarrow j}(z), \quad (3)$$

where $H_{i \leftarrow j}(z) = \sum_{s=0}^{\infty} \pi_{i \leftarrow j}(s) z^s$ is the generating function for $\pi_{i \leftarrow j}(s)$.

To calculate $H_{i \leftarrow j}(z)$, we note that $\pi_{i \leftarrow j}(s)$ is zero if the edge between i and j is unoccupied (which happens with probability $1 - p$) and nonzero otherwise (probability p), which means that $\pi_{i \leftarrow j}(0) = 1 - p$, and for $s \geq 1$

$$\pi_{i \leftarrow j}(s) = p \sum_{\{s_k: k \in \mathcal{N}_j \setminus i\}} \left[\prod_{k \in \mathcal{N}_j \setminus i} \pi_{j \leftarrow k}(s_k) \right] \delta\left(s - 1, \sum_{k \in \mathcal{N}_j \setminus i} s_k\right), \quad (4)$$

where the notation $\mathcal{N}_j \setminus i$ denotes the set of neighbors of j excluding i . Substituting this expression into the definition of $H_{i \leftarrow j}(z)$ above, we then find that

$$H_{i \leftarrow j}(z) = 1 - p + pz \prod_{k \in \mathcal{N}_j \setminus i} H_{j \leftarrow k}(z). \quad (5)$$

This self-consistent equation for the generating function $H_{i \leftarrow j}(z)$ suggests a message-passing algorithm: for any value of z one guesses (for instance, at random) an initial set of values for the $H_{i \leftarrow j}$ and feeds them into the

right-hand side of Eq. (5), giving a new set of values on the left. Repeating this process to convergence gives a solution for the generating functions, which can then be substituted into Eq. (3) to give the generating function for the cluster probabilities $\pi_i(s)$, from which we can recover the probabilities themselves by differentiating.

As an example application of this method, note that, since $\pi_i(s)$ is the probability that i belongs to a small (nonpercolating) cluster of size s , the probability that i belongs to a small cluster of any size is $\sum_s \pi_i(s) = G_i(1)$ and the probability that it belongs to the percolating cluster is 1 minus this. Then the expected fraction S of the network occupied by the entire percolating cluster is given by the average over all nodes:

$$S = \frac{1}{n} \sum_{i=1}^n [1 - G_i(1)] = 1 - \frac{1}{n} \sum_{i=1}^n \prod_{j \in \mathcal{N}_i} H_{i \leftarrow j}(1). \quad (6)$$

Setting $z = 1$ in Eq. (5) we have

$$H_{i \leftarrow j}(1) = 1 - p + p \prod_{k \in \mathcal{N}_j \setminus i} H_{j \leftarrow k}(1), \quad (7)$$

and the solution of this equation, for instance, by iteration from a random initial guess, allows us to calculate the size of the percolating cluster [19]. We give illustrative applications to several networks below.

As another example, consider the case where vertex i does not belong to the percolating cluster. Then the expected size $\langle n_i \rangle$ of the cluster it does belong to is given by

$$\langle n_i \rangle = \frac{\sum_s s \pi_i(s)}{\sum_s \pi_i(s)} = \frac{G_i'(1)}{G_i(1)} = 1 + \sum_{j \in \mathcal{N}_i} \frac{H_{i \leftarrow j}'(1)}{H_{i \leftarrow j}(1)}, \quad (8)$$

and, differentiating Eq. (5), we have

$$H_{i \leftarrow j}'(1) = p \left[1 + \sum_{k \in \mathcal{N}_j \setminus i} \frac{H_{j \leftarrow k}'(1)}{H_{j \leftarrow k}(1)} \right] \prod_{k \in \mathcal{N}_j \setminus i} H_{j \leftarrow k}(1). \quad (9)$$

By iterating both Eqs. (7) and (9) from random initial values and substituting the results into Eq. (8) we can calculate the expected cluster size. Or we can average over all vertices to calculate the network-wide average size of a nonpercolating cluster.

Figure 1 shows results from the application of these techniques to the calculation of cluster sizes for three networks: a computer-generated network that is genuinely treelike (so the method should work well), and two real-world networks for which percolation could be useful as a model of resilience—a network representation of the Internet at the level of autonomous systems and a peer-to-peer file sharing network. Also shown on the figure are results from direct numerical simulations of percolation on the same networks. As the figure shows, the message

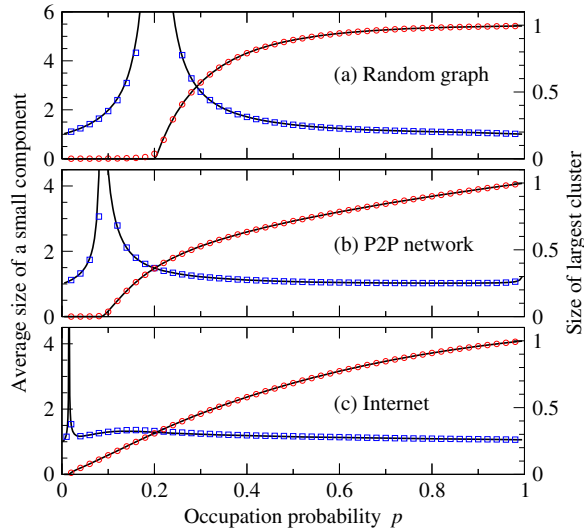


FIG. 1 (color online). Results for three example networks. Solid lines are the message passing calculations and the points, included for comparison, are from direct numerical simulations on the same networks. For each network we show the average size of small clusters (blue squares) and the size of the percolating cluster (red circles). The simulations are averaged over at least 100 repetitions of the percolation process in each case. The networks are (a) an Erdős-Rényi random graph of 10000 nodes and mean degree 5, (b) a peer-to-peer file sharing network of 62586 nodes [20], and (c) a 22963-node snapshot of the structure of the Internet at the level of autonomous systems.

passing and numerical results are in excellent agreement, not only for the computer-generated example but also for the two real-world networks, even though the latter are not treelike. Both the size of the percolating cluster and the mean size of the small clusters are given accurately by the message passing method.

One might ask what the virtue of the message passing method is if one can perform direct percolation simulations of the kind used in Fig. 1. There are two answers to this question. First, simulation algorithms calculate cluster sizes for only a single realization of the randomness inherent in the percolation process. To get accurate results the simulation must be repeated over many realizations, but this can take a significant amount of time and even then the final results still contain statistical errors. The message passing method on the other hand returns the cluster size distribution over all realizations of the randomness in a single calculation.

Second, and perhaps more intriguing, the message passing method not only provides a numerical algorithm for percolation calculations but also allows us to derive new fundamental results by analyzing the behavior of the algorithm itself. As an example, we can calculate the exact position of the percolation threshold on an arbitrarily large, locally treelike network, as follows.

The value $H_{i \leftarrow j}(1) = 1$ for all i, j is trivially a solution of Eq. (7) and hence also a fixed point under the iteration of

that equation. Since $H_{i \leftarrow j}(1)$ is the probability that vertex i does not belong to the percolating cluster, this solution corresponds to the situation in which no vertex is in the percolating cluster. If the solution is a stable fixed point of Eq. (7), then the iteration will converge to it and our message passing algorithm will tell us there is no percolating cluster. If it is unstable, we will end up at a different solution and there is a percolating cluster. Thus, the point at which the trivial fixed point $H_{i \leftarrow j}(1) = 1$ goes from being stable to being unstable is precisely the percolation threshold.

We can determine the stability of the fixed point by linearizing: we write $H_{i \leftarrow j}(1) = 1 - \epsilon_{i \leftarrow j}$ and expand Eq. (7) to leading order in $\epsilon_{i \leftarrow j}$, which gives $\epsilon_{i \leftarrow j} = p \sum_{k \in \mathcal{N}_j} \epsilon_{j \leftarrow k}$, or in matrix notation $\mathbf{\epsilon} = p \mathbf{B} \mathbf{\epsilon}$, where $\mathbf{\epsilon}$ is the $2m$ -element vector with elements $\epsilon_{i \leftarrow j}$ and \mathbf{B} is a $2m \times 2m$ nonsymmetric matrix with rows and columns indexed by directed edges $i \leftarrow j$ and elements $B_{i \leftarrow j, k \leftarrow l} = \delta_{jk}(1 - \delta_{il})$. This matrix is known as the Hashimoto or nonbacktracking matrix and has been a focus of recent attention for its role in community detection and centrality calculations on networks [17,18].

The vector $\mathbf{\epsilon}$ tends to zero and hence the fixed point is stable under iteration of $\mathbf{\epsilon} = p \mathbf{B} \mathbf{\epsilon}$ if and only if p times the leading eigenvalue of \mathbf{B} is less than unity. Hence, we conclude that the critical percolation probability p_c of a sparse, locally treelike network is equal to the reciprocal of the leading eigenvalue of the nonbacktracking matrix.

A different result, reminiscent of this one, has been given recently by Bollobás *et al.* [21], who show that in the limit of large network size the critical occupation probability for percolation on a dense network is equal to the reciprocal of the leading eigenvalue of the adjacency matrix. The result given here is the equivalent for sparse networks.

As a simple example consider a random k -regular graph, i.e., a network in which every node has exactly k edges but connections are otherwise made at random. For such a graph the nonbacktracking matrix has $k-1$ nonzero elements in each row and column and hence its largest eigenvalue is exactly $k-1$, giving $p_c = 1/(k-1)$, which can easily be confirmed to be the correct answer using other methods [11,12]. The leading eigenvalue of the adjacency matrix on the other hand, which gives the dense-graph percolation threshold as discussed above, is k and hence would give a lower, and incorrect, result of $p_c = 1/k$.

In fact, the leading eigenvalue of the adjacency matrix is never less than the leading eigenvalue of the nonbacktracking matrix. To see this, consider a matrix \mathbf{B}' , which is a slight variant of the nonbacktracking matrix having elements $B'_{i \leftarrow j, k \leftarrow l} = \delta_{jk}$. An eigenvector \mathbf{v} of this matrix with elements $v_{i \leftarrow j}$ and eigenvalue λ satisfies $\lambda v_{i \leftarrow j} = \sum_{k \leftarrow l} \delta_{jk} v_{k \leftarrow l} = \sum_{kl} A_{kl} \delta_{jk} v_{k \leftarrow l} = \sum_l A_{jl} v_{j \leftarrow l}$, which has solutions $v_{i \leftarrow j} = w_j$ where w_j are the elements of any eigenvector \mathbf{w} of the adjacency matrix \mathbf{A} and λ is the

corresponding eigenvalue. Thus, \mathbf{B}' and \mathbf{A} have the same eigenvalues and in particular the leading eigenvalue of \mathbf{B}' is also the leading eigenvalue λ_A of \mathbf{A} .

We now observe that the difference $\Delta\mathbf{B} = \mathbf{B}' - \mathbf{B}$ has elements $\Delta B_{i \leftarrow j, k \leftarrow l} = \delta_{jk} \delta_{il}$, which are all non-negative, and we apply the so-called Collatz-Wielandt formula, a corollary of the Perron-Frobenius theorem which says that for any real vector \mathbf{x} the leading eigenvalue of \mathbf{B}' (which, as we have said, is equal to λ_A) satisfies

$$\lambda_A \geq \min_{i: x_i \neq 0} \frac{[\mathbf{B}'\mathbf{x}]_i}{x_i} = \min_{i: x_i \neq 0} \left(\frac{[\mathbf{B}\mathbf{x}]_i}{x_i} + \frac{[\Delta\mathbf{B}\mathbf{x}]_i}{x_i} \right). \quad (10)$$

Let us choose \mathbf{x} to be the leading eigenvector of \mathbf{B} , which has all elements non-negative by the Perron-Frobenius theorem. Then $[\mathbf{B}\mathbf{x}]_i/x_i = \lambda_B$, where λ_B is the leading eigenvalue of \mathbf{B} , and $[\Delta\mathbf{B}\mathbf{x}]_i/x_i \geq 0$ for all i , so Eq. (10) implies that $\lambda_A \geq \lambda_B$.

This in turn implies that the dense-matrix result for the percolation threshold based on the adjacency matrix is a lower bound on the percolation threshold of a sparse treelike graph.

An interesting special case is that of a perfect tree, a network with no loops at all. Percolation, in the sense of a percolating cluster that fills a nonzero fraction of the network in the large- n limit, never occurs on such a network—for all $p < 1$ the largest cluster occupies only a vanishing fraction of the network and our formalism gives this result correctly. The diagonal elements of powers of the nonbacktracking matrix count numbers of closed nonbacktracking walks on a graph [17,22] (hence the name “nonbacktracking matrix”), but a perfect tree has no such walks, so the trace of every power of the matrix is zero and hence so also are all eigenvalues. Thus, the reciprocal of the largest eigenvalue diverges and there is no percolation threshold. The leading eigenvalue of the adjacency matrix, on the other hand, is nonzero on a tree. On a k -regular tree, for instance, the leading eigenvalue of the adjacency matrix for large n is k again, implying a percolation threshold of $1/k$. This is, indeed, a lower bound on the true percolation threshold, as it must be, but it is in error by a wide margin.

All of our results so far have been for treelike networks, but most real-world networks are not trees. We can nonetheless use the techniques developed here to say something about the nontreelike case. On a tree the number of nodes reachable along the edge from i to j is 1 (for node j itself) plus the sum of the numbers $n_{j \leftarrow k}$ reachable along every other edge attached to j . On a nontree, on the other hand, this sum overestimates the number of reachable nodes because some nodes are reachable along more than one edge from j . This means that for $z \leq 1$ the generating function $H_{i \leftarrow j}(z)$ for the true number of reachable nodes will be greater than or equal to the value given by a naive estimate calculated from a simple average over the randomness:

$$\begin{aligned} H_{i \leftarrow j}(z) &\geq 1 - p + pz \left\langle z^{\sum_{k \in \mathcal{N}_j \setminus i} n_{j \leftarrow k}} \right\rangle \\ &= 1 - p + pz \left\langle \prod_{k \in \mathcal{N}_j \setminus i} z^{n_{j \leftarrow k}} \right\rangle \\ &\geq 1 - p + pz \prod_{k \in \mathcal{N}_j \setminus i} \langle z^{n_{j \leftarrow k}} \rangle, \end{aligned} \quad (11)$$

where the second inequality follows by an application of the Chebyshev integral inequality [23]. But $\langle z^{n_{j \leftarrow k}} \rangle = H_{j \leftarrow k}(z)$ by definition, so we find that on a nontreelike network the exact equality of Eq. (5) is replaced with an inequality:

$$H_{i \leftarrow j}(z) \geq 1 - p + pz \prod_{k \in \mathcal{N}_j \setminus i} H_{j \leftarrow k}(z). \quad (12)$$

Suppose, however, that we nonetheless decide to use the exact equality of Eq. (5), iterating to estimate the generating functions. If we start from an initial value of $H_{i \leftarrow j}$ equal to the true answer we are looking for (which we do not know, but let us suppose momentarily that we do), then it is straightforward to see from Eq. (12) that the value of $H_{i \leftarrow j}$ will never increase under the iteration, implying that the value we calculate will be a lower bound on the true value for all $z \leq 1$. As we approach the percolation threshold from above in the large size limit, the true value of $H_{i \leftarrow j}(1)$, which represents the probability that the edge from i to j connects to a small cluster, approaches 1, while the value calculated from Eq. (5), which is less than or equal to the true value, must reach 1 later, i.e., at a lower or equal value of p . Thus, the percolation threshold estimated from Eq. (5) is never higher than the true percolation threshold. Equivalently, we can say that for any network, p_c is always greater than or equal to the inverse of the leading eigenvalue of the nonbacktracking matrix. The only exception is for the case of a perfect tree, for which the largest eigenvalue is zero, as discussed above. Thus, the leading eigenvalue gives us a bound on the percolation threshold.

We can also combine this result with our earlier observation that the leading eigenvalue of the adjacency matrix is never less than that of the nonbacktracking matrix to make the further statement that p_c for any network is always greater than or equal to the inverse of the leading eigenvalue of the adjacency matrix. Thus, both eigenvalues place lower bounds on p_c , but the bound given by the nonbacktracking matrix is better (or at least never worse) than the one given by the adjacency matrix. Numerical tests of these results on various networks are given in the Supplemental Material [24].

In summary, we have in this Letter shown that percolation on sparse, locally treelike networks can be reformulated as a message passing process, allowing us to solve for average percolation properties such as the size of the percolating cluster and the average size of the nonpercolating clusters. Tests on both computer generated and

real-world networks show good agreement with numerical simulations of percolation on the same networks. By analyzing the message passing equations we have also shown that the position of the percolation threshold on treelike networks is given by the inverse of the leading eigenvalue of the nonbacktracking matrix. On nontreelike networks this result is not exact but it gives a bound on the exact result.

The authors thank Cris Moore, Leonid Pryadko, and Pan Zhang for useful conversations. This work was funded in part by the National Science Foundation under Grants No. DMS-1107796 and No. DMS-1407207 and by DARPA under Grant No. FA9550-12-1-0432.

Note added.—Recently we learned of concurrent work by Hamilton and Pryadko [25] in which a similar result for the percolation threshold is derived.

-
- [1] J. Machta, *Phys. Rev. Lett.* **66**, 169 (1991).
 [2] K. Moon and S. M. Girvin, *Phys. Rev. Lett.* **75**, 1328 (1995).
 [3] T. Odagaki and S. Toyofuku, *J. Phys. Condens. Matter* **10**, 6447 (1998).
 [4] J. Tobochnik, *Phys. Rev. E* **60**, 7137 (1999).
 [5] S. de Bondt, L. Froyen, and A. Deruyttere, *J. Mater. Sci.* **27**, 1983 (1992).
 [6] D. P. Bentz and E. J. Garboczi, *Materials and Structures* **25**, 523 (1992).
 [7] L. de Arcangelis, S. Redner, and A. Coniglio, *Phys. Rev. B* **31**, 4725 (1985).
 [8] C. L. Henley, *Phys. Rev. Lett.* **71**, 2741 (1993).
 [9] P. Grassberger, *Math. Biosci.* **63**, 157 (1983).
 [10] M. E. J. Newman, *Phys. Rev. E* **66**, 016128 (2002).
 [11] R. Cohen, K. Erez, D. ben-Avraham, and S. Havlin, *Phys. Rev. Lett.* **85**, 4626 (2000).
 [12] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts, *Phys. Rev. Lett.* **85**, 5468 (2000).
 [13] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, *Phys. Rev. E* **65**, 056109 (2002).
 [14] A. V. Goltsev, S. N. Dorogovtsev, and J. F. F. Mendes, *Phys. Rev. E* **78**, 051105 (2008).
 [15] S. Janson, *Electron. J. Probab.* **14**, 5 (2009).
 [16] K. Hashimoto, *Adv. Stud. Pure Math.* **15**, 211 (1989).
 [17] F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborová, and P. Zhang, *Proc. Natl. Acad. Sci. U.S.A.* **110**, 20935 (2013).
 [18] T. Martin, X. Zhang, and M. E. J. Newman, *arXiv:1401.5093*.
 [19] Equations equivalent to Eq. (7) have appeared previously in O. Rivoire, Ph.D. thesis, Université Paris Sud-Paris XI, 2005.
 [20] A. Iamnitchi, M. Ripeanu, and I. Foster, in *Proceedings of the First International Workshop on Peer-to-Peer Systems*, edited by P. Druschel, F. Kaashoek, and A. Rowstron, Lecture Notes in Computer Science Vol. 2429 (Springer, Berlin, 2002), pp. 232–241.
 [21] B. Bollobás, C. Borgs, J. Chayes, and O. Riordan, *Ann. Probab.* **38**, 150 (2010).
 [22] N. Alon, I. Benjamini, E. Lubetzky, and S. Sodin, *Commun. Contemp. Math.* **09**, 585 (2007).
 [23] B. Karrer and M. E. J. Newman, *Phys. Rev. E* **82**, 016101 (2010).
 [24] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/PhysRevLett.113.208702> for percolation threshold results and a discussion of numerical methods.
 [25] K. E. Hamilton and L. P. Pryadko, preceding Letter, *Phys. Rev. Lett.* **113**, 208701 (2014).