Random networks with tunable degree distribution and clustering

Erik Volz*
Cornell University, Ithaca, New York 14853, USA

(Received 4 June 2004; published 17 November 2004)

We present an algorithm for generating random networks with arbitrary degree distribution and clustering (frequency of triadic closure). We use this algorithm to generate networks with exponential, power law, and Poisson degree distributions with variable levels of clustering. Such networks may be used as models of social networks and as a testable null hypothesis about network structure. Finally, we explore the effects of clustering on the point of the phase transition where a giant component forms in a random network, and on the size of the giant component. Some analysis of these effects is presented.

DOI: 10.1103/PhysRevE.70.056115 PACS number(s): 89.75.Hc, 87.23.Ge, 89.20.Hh

I. INTRODUCTION

Many random network models have been proposed to replicate important aspects of the topology of real-world networks [1–14]. In particular, much attention has been paid to the degree distribution and the clustering coefficient. A great deal of progress has been made on network models which combine certain degree distributions with some level of clustering [13,15–19]. It has been an open problem to combine these two topologies in the most general way. Is it possible to have a network model which is flexible enough to accommodate any combination of degree distribution and clustering? In this paper we propose such a model and demonstrate its effectiveness by generating networks over a wide range of parameters.

Random network models have fallen into several broad categories. Some models have focused on Monte Carlo techniques to reproduce a specific topology [1,2,20]. Other models have specific topologies built into them (e.g., regular lattices) in order to explicate the so-called "small-world" problem [8,9]. Yet other models have focused on plausible mechanisms for how networks form, such as a growth process with preferential attachment [10,11,15]. In common with most mechanism-based models, we produce our networks by growing them from one initial node. We find that being able to construct a network one node at a time also offers sufficient flexibility to combine arbitrary degree distributions and clustering.

Once we have a network model that can combine arbitrary degree distributions and clustering, it is of interest to explore the effects of these parameters on the size of the giant component and the point of the phase transition where a giant component forms. This is true with regard to clustering in particular, as so far models capable of interpolating between extremes of this parameter have been lacking. In Sec. III we explore the effects of clustering on the size of the giant component and point of the phase transition. In Sec. IV we present some analysis.

Throughout this article we will rely on the following definitions. The *degree distribution* of a network describes how many neighbors a node in a network has. The probability of a node having degree k in a network is described by the degree distribution p_k, where p_k can take the form of any well defined discrete density function over the positive integers. Examples frequently employed in the literature are the following.

Poisson distribution: $p_k = z^k e^{-z}/k!$, $k \geqslant 0$.

Power-law distribution. For our experiments, we use power laws with finite cuttoffs κ: $p_k = k^{-\gamma} e^{-k/\kappa}/\mathrm{Li}_\gamma(e^{-1/\kappa})$, $k \geqslant 1$, where $\mathrm{Li}_n(x)$ is the nth polylogarithm of x.

Exponential distribution: $p_k = (1 - e^{-1/\lambda}) e^{-k/\lambda}$, $k \geqslant 0$.

Empirical distribution: The degree distribution is estimated from a network sample.

Gaussian distribution: The ordinary Gaussian must be modified to be positive and discrete.

The *clustering coefficient C* describes the proportion of triads in a network out of the total number of possible triads. The clustering coefficient is defined as

$$C = \frac{3N_\Delta}{N_3}$$

where N_Δ is the number of triads in the network and N_3 is the number of connected triples of nodes. Note that in every triad there are three connected triples.

There is also a measure of *local clustering* given by

$$C_i = \frac{N_\Delta(i)}{\binom{\delta(i)}{2}}$$

where $N_\Delta(k)$ is the number of triads connected to node i, $\delta(i)$ is the degree of node i, and $\binom{\delta(i)}{2}$ is the number of potential triads connected to a node of degree $\delta(i)$. The average value of local clustering (i.e., "Watts-Strogatz Clustering" [8]) is also of interest:

$$\frac{\sum C_i}{N}$$

where N is the number of nodes in the network. This value is frequently close to the clustering coefficient, and will be equal to the clustering coefficient if local clustering is constant throughout the network.

[*] Electronic address: emv7@cornell.edu

TABLE I. Detailed description of the clustering method.

(1) Initialize all nodes with a degree drawn independently from the degree distribution.

(2) Form a list of "stubs"—connections of nodes which have not yet been matched with neighbors. Call this list StubList.

(3) Pick a starting node v_0 uniformly at random from all nodes.

(4) For each of v_0's stubs, choose a new neighbor by picking an element v_1 from the StubList with probability $p_{v_1|d(v_0)}$ as described in the text. If the new neighbor is not the same node as v_0 or already connected to v_0 then form the connection. Otherwise, repeat the process until a valid neighbor is found. Add all of the new neighbors from this process to a list called NextWave.

(5) Copy all elements of NextWave to a list called CurrentWave. Remove all elements from NextWave. For all elements in CurrentWave do the following.

(a) Form a list of all nodes two steps away. If a node does not have any stubs left in StubList, throw it out. Call this list PotentialTriads.

(b) For all stubs which have not been assigned neighbors: (i) Scan through PotentialTriads. With probability C^{input}, connect to node $v_3 \in$ StubList. Remove element v_3 from PotentialTriads regardless of whether it was selected. If it was selected, also remove an instance of v_3 from the StubList. (ii) If no neighbors were selected from PotentialTriads, select a new neighbor by choosing from StubList as above. If the new neighbor is not in CurrentWave, and if the new neighbor is not already in NextWave, add them to NextWave.

(6) Repeat the last step until NextWave is empty following an iteration. Then, if StubList is empty, the process is complete—all connections have been formed. Otherwise, start a new component by choosing a new starting node uniformly at random from those not yet in the network.

II. RANDOM NETWORK MODEL

Introducing clustering into a network with a specified degree distribution is a nontrivial problem. Any method aspiring to introduce an arbitrary amount of clustering into a network must interpolate between two extremely different topologies. When clustering is 0%, the method must reproduce pure random networks with specified degree distributions. When clustering is 100%, there is only one configuration a network may have: each node must be connected to a small clique where every node has the same degree, and all of a node's neighbors are connected with one another. This challenge is made all the more difficult by trying to make the model networks general enough to accommodate any desired degree distribution.

The most obvious way of introducing triads is to simply define a *rewiring rule* whereby links are swapped between nodes so as to introduce triads while leaving the degree distribution the same. Such rewiring schemes quickly run into problems, as it is impossible to define a rule such that the number of triads is strictly increasing and the number of triads introduced does not max out. The problem is that when links are "swapped" among nodes, triads are not only created but can be destroyed. For example, in our simulations we have found that such schemes are effective only for introducing about 15% clustering into a poisson random network.

Rewiring algorithms have proven effective at the related challenge of adjusting the *average local clustering*. Kim [12] has recently used rewiring algorithms to introduce large amounts of *local clustering* into networks. Using Monte Carlo (MC) simulations at zero temperature (i.e., a triad is never destroyed in the rewiring process) and a Hamiltonian of $\Sigma - C_k$, Kim was able to modify various networks with diverse degree distributions to exhibit average local clustering ($\Sigma C_k / N$) ranging from 0% to 70%.

Newman [22] and Guillaume *et al.* [19] have had some success with another approach. These authors define a bipartite network of individuals and affiliations. Then they project the bipartite network onto a unipartite network of only nodes and no affiliations by connecting two nodes if they share a common affiliation. The distribution of affiliation size and the affiliation-degree distribution of the nodes are chosen in such a way as to produce a desired level of clustering. Tuning the degree distribution simultaneously has proven more challenging, however. While the bipartite projection method may actually have the potential to generate pure random networks with tunable degree distributions and clustering, so far its efficacy has been shown only for exponential and power-law random networks. It remains an open problem to implement it for arbitrary degree distributions.

Our method works by growing networks. The algorithm first initializes all nodes with a degree drawn independently from the desired degree distribution. Then the random network is constructed by an iterative procedure similar to a branching process. The premise is to start from a single node and then assign new connections entirely at random under the constraint that a certain amount of clustering must exist. The algorithm is described in detail in Table I, and is schematized in Fig. 1. Two example networks are shown in Fig. 2.

Our model has similarities to and differences from other models proposed in the literature. Like the algorithm of Milo *et al.* [20], each node is assigned a unique degree prior to any edges being formed between nodes. But like the model networks of Barabasi [3] and Dorogovtsev *et al.* [21] among others, the network is constructed via a growth process. The first node is chosen at random, and subsequently nodes are added to the graph by attaching them to nodes which still have stubs that have not been matched. When the new node forms its own connections, it first forms a list of all nodes which are two steps away. Then with probability C_{input}, that node is selected as the next neighbor.

One complicated feature of this algorithm concerns the probability of selecting a new neighbor from the stub list. In

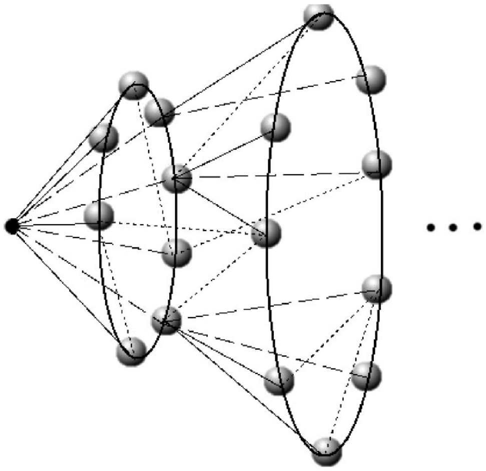

FIG. 1. Overview of the network construction process. The first node (far left) is chosen at random. Then neighbors for that node are chosen as described in the text. Subsequently, neighbors are chosen for the new nodes, but now we have new connections formed with nodes two steps away with probability C_{input}. Triadic connections are indicated with dotted lines. This process continues until the waves die out, and a new component is formed, or all nodes are exhausted.

fact, new neighbors cannot be selected uniformly at random from the stub list, as clustering implies a certain amount of degree assortativity among the nodes in the network. For example, a node connected to a degree k node has $k-1$ potential triads in common with that node, and on average will have $C(k-1)$ common triads. This implies that the node must have on average a degree at least equal to $C(k-1)$.

Because triads are distributed uniformly throughout the network, the number of triads connected to a node of degree k is distributed binomial $\left(\binom{k}{2},C\right)$. As noted above the number of common triads with a neighbor of degree k is distributed binomial $(k-1,C)$. Let τ_{ij} denote the number of triads node i has in common with node j, and τ_{ji} denote the number of triads j has in common with i. Of course these two random variables should be equal. We can calculate the probability of these two potential neighbors as having an equal number of common triads as:

$$p_{ij}^{c}=\sum_{x=0}^{\min[d(i),d(j)]} p(\tau_{ij}=x)p(\tau_{ji}=x).$$

Let q_j denote the probability of selecting node j from the stub list. Then the correct probability for selecting node j as a neighbor is

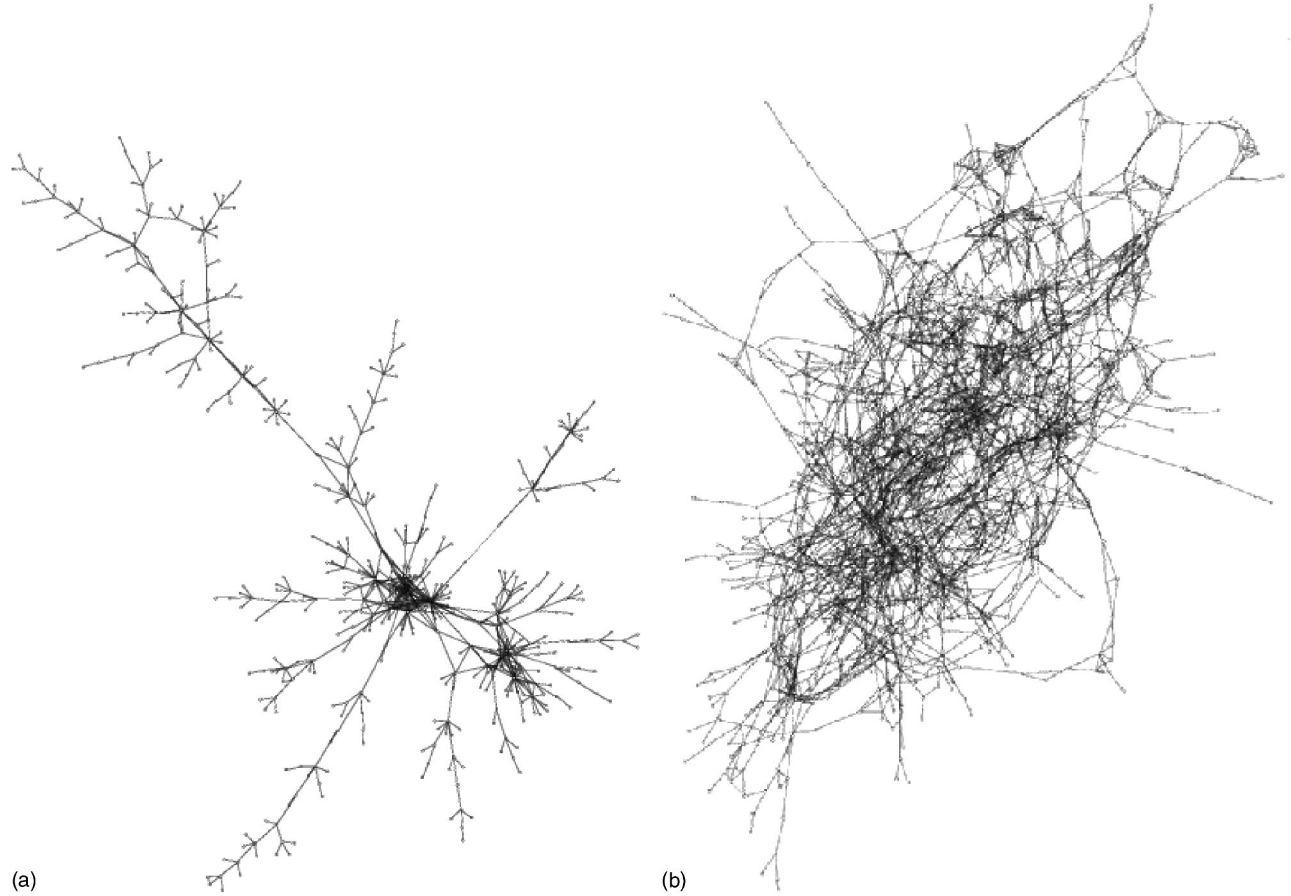

FIG. 2. Two examples of networks generated with the algorithm. Left: Random network with power-law degree distribution, $\kappa=15$, $\gamma=2$, $C=0.15$. Right: Random network with Poisson degree distribution, $z=4$, $C=0.40$ [38]. Note that these are abstract representations of random networks. The spatial embedding of the network does not have any meaning.

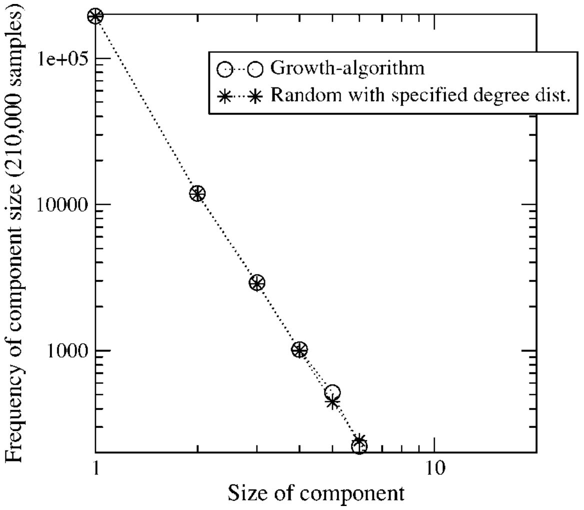

FIG. 3. Random graphs were generated with an exponential degree distribution $(\lambda=1.5)$ with two algorithms: (1) the clustering algorithm described in this text with $C=0$; (2) a "stub-matching" algorithm as in [2], known to produce true random graphs with specified degree distributions. The frequency of component sizes is illustrated above.

$$q_{ij} = \frac{q_j p_{ij}^c}{\sum_\alpha p_{i\alpha}^c}$$

which is just q_j weighted by the probability of the two neighbors having a compatible number of triads in common.

In order to sample from this distribution, we use Markov chain Monte Carlo techniques. For a large number of iterations we select a new node β from the stub list, then with probability $a_{\alpha\beta}$ we accept this new neighbor, where α is the currently selected node in the Markov process, and

$$a_{ij} = \frac{p_{i\mu}^c}{p_{i\alpha}^c}.$$

If β is not accepted, we keep α for the next iteration. The final neighbor is the node selected at the last iteration.

It is desirable that our algorithm selects networks as uniformly as possible from the ensemble of all networks which realize a given degree distribution and clustering coefficient. It is difficult to prove that our algorithm is truly unbiased in this sense, though our networks do have many of the properties of an unbiased random network. The algorithm can be tuned to produce exactly the right proportion of triads to triples in the limit of large graph size. Furthermore, the degree of the nodes were chosen as independently random variables, so in the limit of large graph size, the degree distribution is unbiased too. Furthermore, the triads are uniformly distributed throughout the network as reflected by the fact that the local clustering is independent of degree. Lastly, when this algorithm is used to produce networks with no clustering at all, it produces networks with the same statistical properties as true random graphs with a specified degree distribution. As shown in Fig. 3, the distribution of component sizes for networks made with this algorithm is identical to true random graphs with specified degree distribution without clustering.

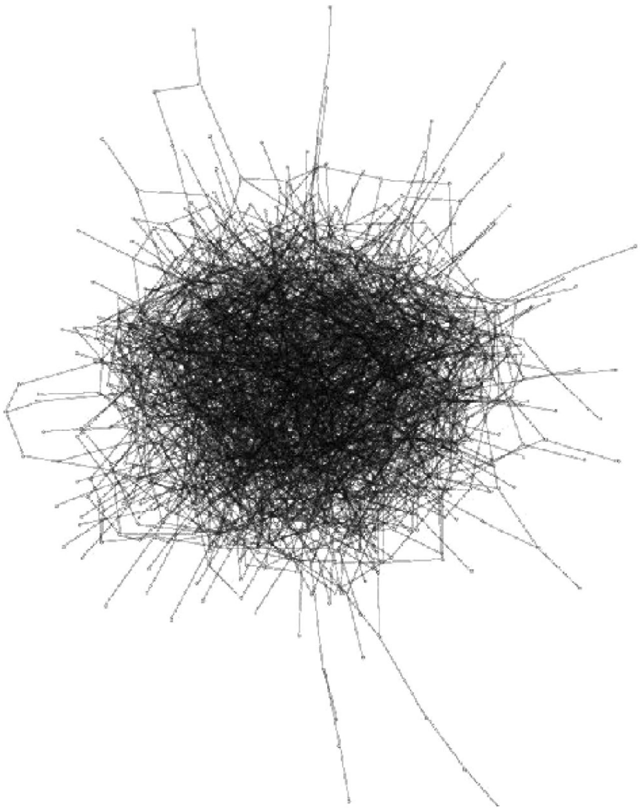

FIG. 4. Random network on 1500 nodes, Poisson degree distribution $(z=4)$, $C=0.00$. Compare with Figs. 2 (right) and 5.

It is worth noting that many real-world networks, particularly in the biological realm, have local clustering which scales as $1/k$ [23]. Our model in contrast produces constant local clustering, though it may be possible to generalize our method to create networks with any desired schedule of local clustering.

III. RESULTS

We have explored the effects of clustering and degree distribution over a wide range of parameters. Figures 2 (right), 4 and 5 illustrate the effect of clustering on the structure of a random networks with Poisson degree distributions $(z=3)$ as clustering is increased from 0 to 1.00. As C is increased, nodes tend to disaggregate into smaller tightly connected clusters of nodes with similar degree. This has the overall effect of decreasing the giant component size as clustering is increased. In the limit as C goes to 1, we find that the network breaks down into many small completely connected cliques with each node in a clique sharing a common degree.

Figure 6 shows the effects of clustering on the size of the giant component for a Poisson random network. Clustering varies from 0.05 to 0.90. The giant component seems to undergo a phase transition at a critical level of clustering around $C=0.60$. In the next section we will find that the

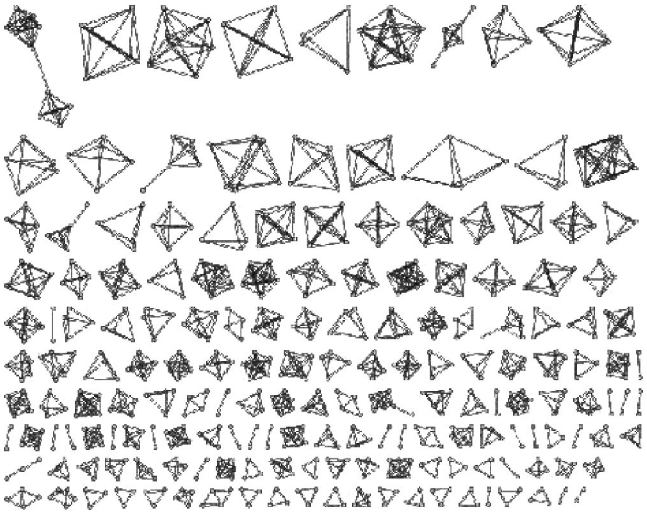

FIG. 5. Random network on 1500 nodes, Poisson degree distribution $(z=4)$, $C=0.97$.

critical clustering value is actually $C^*=0.618$. At this point, nodes suddenly disaggregate into much smaller, tightly interconnected groups. Similar phase transitions have been observed throughout the network literature, particularly concerning the targeted deletion of links and nodes in percolation phenomena [24]. This algorithm has similar disconnecting results without modifying the degree distribution of the network.

Regarding power-law networks (see Fig. 7), we note the striking tendency for moderate levels of clustering to limit the size of the giant component. Because the number of potential triads connected to a node scales as k^2, the high-degree vertices account for most of the clustering. In networks with highly skewed degree distributions, the high-degree nodes must connect to one another in order to realize the required number of triads. This has the effect of limiting the ability to act as hubs for low-degree vertices, and consequently the network disconnects into many small components. Large components can be preserved under much higher clustering with distributions such as the Poisson.

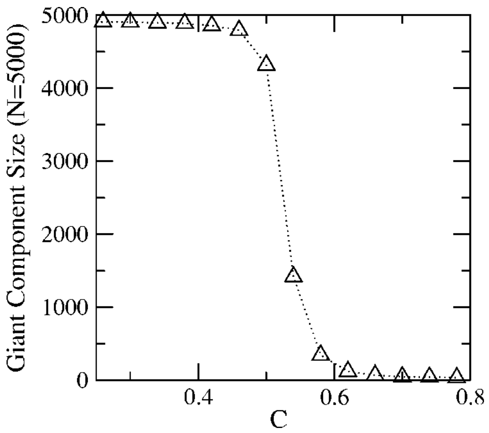

FIG. 6. Size of the giant component versus the clustering coefficient in a Poisson random network, $z=3$. Each point represents the average of 40 trials.

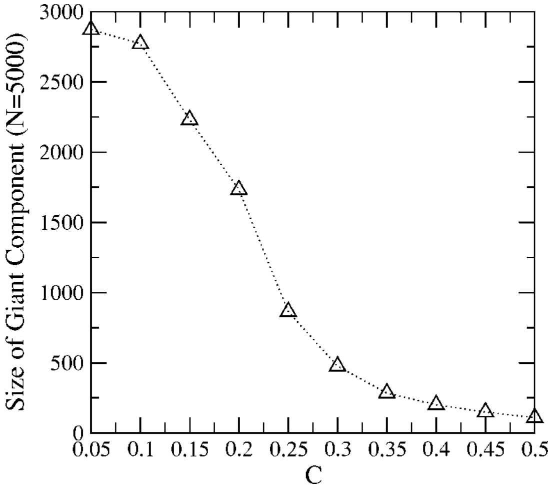

FIG. 7. $N=5000$ nodes. Power-law with parameters $\kappa=10$ and $\gamma=2$. Each point represents the average of 40 trials. Compare this with Fig. 6. The phase transition is much less sharp than for the Poisson random networks.

The phase transition also undergoes major changes with the introduction of clustering, although this effect seems to depend sensitively on the degree distribution. In Fig. 8 we see that the phase transition where a giant component forms is not significantly affected by the introduction of clustering for networks with power-law degree distributions. In contrast to the Poisson random networks, there is no sharp phase transition between the regime with a giant component and that without. This bears some resemblance to percolation phenomena, where the phase transition disappears for true power laws and an exponent of 2. But in Fig. 9 we see that the point of the phase transition was dramatically shifted forward for the Poisson random network. It is somewhat surprising to observe the phase transition being shifted *forward* as our algorithm features the introduction of degree assortativity into the network. Previous research has shown the ten-

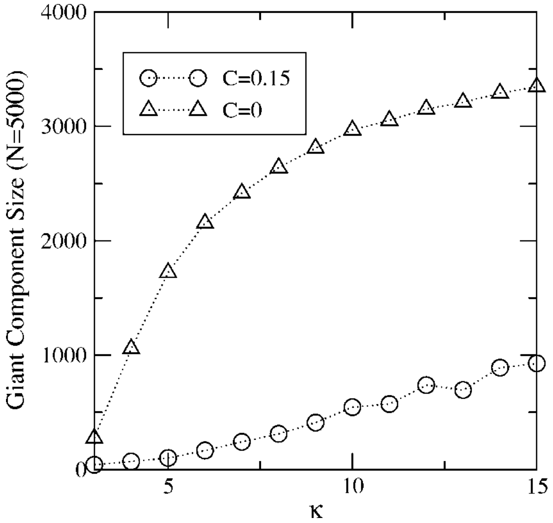

FIG. 8. Two random networks are compared over a range of parameter values for the power-law degree distribution with parameters κ and $\gamma=2$. Each point represents the average of 40 trials.

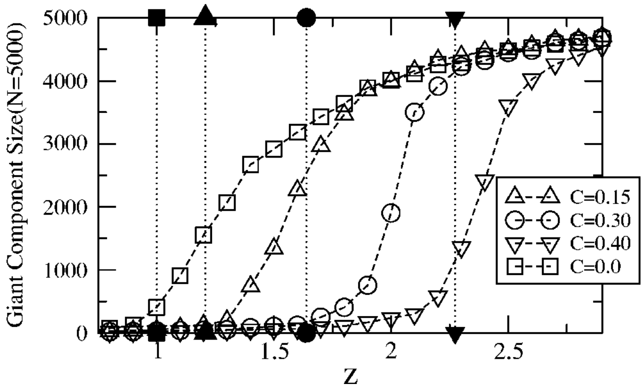

FIG. 9. The size of the giant component is shown vs z, the parameter of the Poisson degree distribution, for four levels of clustering ($C=0.0$, $C=0.15$, $C=0.30$, $C=0.40$). The vertical lines indicate the point of the phase transition for each level of clustering predicted by Eq. (4).

dency of degree assortativityto shift the point of the phase transition backward [25].

IV. PHASE TRANSITIONS

It is a necessary condition for a giant component to exist that if we pick a node at random, the average number of neighbors two steps away, s_2, exceeds the number of neighbors one step away, s_1 [26]. This is intuitive, since if it were not the case, the number of neighbors n steps away would decrease to zero on average, and the component would be finite in the limit of large network size. We can use this to approximate the point of the phase transition as clustering is varied in our random networks. Formally, we will solve for the point where

$$s_1 = s_2. \tag{1}$$

The necessary condition (1) will not quite be a sufficient condition in the presence of clustering as described below. Thus, our solution will only be a lower bound on the point of the phase transition, but in practice, this will serve as an excellent approximation.

For the Poisson degree distribution, the average number of nodes one step away is equal to the parameter of the distribution z, so we have $s_1=z$. As is well known [1], the number of edges emanating from a node if we pick an edge at random and follow it to one of its ends is also z for the Poisson degree distribution. Thus, in the absence of clustering we would have simply $s_2=s_1z=z^2$, where s_2 is the average number of nodes two steps away from a randomly chosen node.

In the presence of clustering, things become more complicated. Let us pick a node uniformly at random in the network and call this node v_0. A neighbor of this node, v_1, will have on average z connections not in common with v_0. Furthermore, there will be on average Cz triadic connections between v_0 and v_1 as each of those connections has a probability C of being a triad. We can simply deduct the triadic connections from s_2, so that we have

$$s_2 > z^2 - Cz^2 = z^2(1-C). \tag{2}$$

There is not equality in Eq. (2) because there is an additional force limiting the number of second neighbors: Once two neighbors of v_0, say v_1 and v_1', share a triadic connection, it becomes more likely that a node two steps away from v_0, say v_2, is a common neighbor of both v_1 and v_1'. In fact, such connections exist with probability C. Then, the number of connections we should deduct from every neighbor at distance 2 due to common connections of nodes at distance 1 is equal to C times the average number of triadic connections at distance 1, or in other words z^2C^2. Thus, we have

$$s_2 = z^2 - Cz^2 - C^2z^2 = z^2(1-C-C^2).$$

We can use this to solve for the critical z_C^* where a giant component forms given a level of clustering C:

$$z = z^2(1-C-C^2). \tag{3}$$

The nonzero root of this equation is given by

$$z_C^* = \frac{1}{1-C-C^2}. \tag{4}$$

Note that when $C=0$ we retrieve the well known result that a giant component forms when $z=1$ in the absence of clustering. Unfortunately, we can only say that this is a lower bound for the phase transition because the nodes at distance 2 are not identical to v_0. The number of outgoing connections from such nodes (to nodes not already counted) is less than $z-C^2z$ on average.

In Fig. 9 we have plotted the size of the giant component versus the parameter z for several levels of clustering. The vertical lines correspond to the phase transitions z_C^* as given by (4). We find good agreement between theory and simulation.

There is a singularity in (4) where $1-C-C^2=0$. At this point, $C^*=0.618$, the giant component disappears regardless of the average degree z of the degree distribution. C^* represents the critical level of clustering that can coexist in a network with a giant component.

V. FINITE SIZE EFFECTS

During the execution of the algorithm, it occasionally happens that a node cannot find a suitable neighbor due to the absence of a node left in the network which has free stubs and the correct degree to satisfy the degree assortativity requirements. This imperfection is due to the finite size of the network. In the limit of large size, it would always be possible to find a scale such that every node can find just the right profile of neighbors with the right degree. There is no perfect way to deal with such discprepancies. For the simulations used in this article, we have simply truncated the degree of that node so that it does not have to seek a new neighbor. Even with networks of only 5000 nodes, the number of corrections made is quite small.

Figures 10 and 11 show the effects of network size and clustering on the amount of degree corrections made by the algorithm. Figure 10 shows the effects of clustering on the

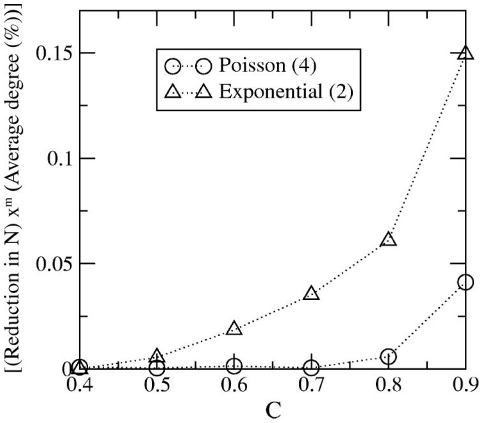

FIG. 10. The percentage reduction in the number of “stubs” is shown versus the clustering coefficient for two networks: (i) Poisson degree distribution with parameter=4; (ii) exponential degree distribution with parameter=2. $N=5000$ for both networks. Each point is based on the average of 20 trials.

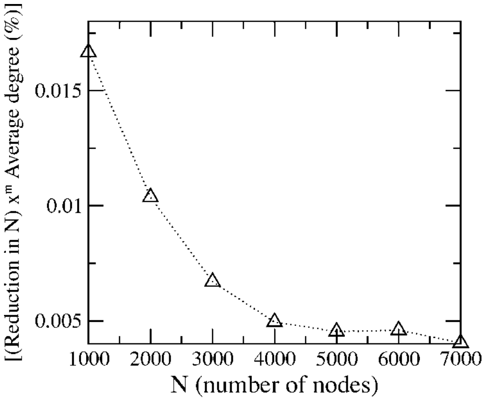

FIG. 11. The percentage reduction in the number of “stubs” is shown versus the network size. The network has a Poisson degree distribution with parameter=4, $C=0.80$. Each point is based on the average of 20 trial networks.

number of corrections made for two networks. Note that the total number of “stubs” in the network is equal to the average degree of the nodes times the population size. The corrections made is shown as the proportional reduction in the number of “stubs.” Even at 90% clustering, the Poisson random network only undergoes less than 5% reduction in its “stubs.”

Figure 11 shows the effects of network size on the number of corrections made. As expected, the number of corrections drops with the number of nodes in the network. For 7000 nodes and 80% clustering, a Poisson random network undergoes less than a 0.5% reduction in its “stubs.”

VI. DISCUSSION

We have presented a method for generating random networks which unite two frequently modeled topological features—clustering and the degree distribution.

Random network models can serve several important purposes. First, they can serve as a null hypothesis about the structure of a real-world network. Significant deviations in the structure of the real-world network from a corresponding random graph indicate that there are more forces at work shaping the network than are being accounted for in the random graph model. These deviations can then motivate further inquiry into the forces shaping real-world networks [1].

Second, real-world networks are very often of a scale that it is impossible to map them entirely. Various network sampling techniques have been devised to estimate features of the network topology in the absence of data on the entire network [27–29]. Given reliable estimates about network topology, a random network can then be generated which reproduces this topology. The random network may be used as a stand-in for modeling various dynamic models on networks.

Last, the family of random networks we have presented here enables the exploration of a huge parameter space for models on networks. There are a growing number of models which describe dynamic processes on networks. Examples are models of diffusion processes, such as models of epidemics [30–32], models of fads [33,34], the spread of rumors [35,36], and the migration of species among connected habitats [37]. Other models explore interactions among nodes embedded in a network. Examples include spin glasses, Kuramoto oscillators, and disordered neural networks [12]. There are many applications for exploring the effects of clustering and degree distributions on these and other models.

[1] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, Phys. Rev. E **64**, 026118 (2001).

[2] M. E. J. Newman, D. J. Watts, and S. H. Strogatz, Proc. Natl. Acad. Sci. U.S.A. **99**, 2566 (2002).

[3] R. Albert and A. L. Barabasi, Rev. Mod. Phys. **74**, 47 (2002).

[4] R. Albert, H. Jeong, and A. L. Barabasi, Nature (London) **406**, 378 (2000).

[5] L. Barabasi, *Linked: The New Science of Networks* (Perseus, Cambridge, U.K., 2002).

[6] E. Ravasz and A.-L. Barabási, e-print cond-mat/0206130.

[7] G. Caldarelli, A. Capocci, P. De Los Rios, and M. A. Muñoz, Phys. Rev. Lett. **89**, 258702 (2002).

[8] D. J. Watts and S. H. Strogatz, Nature (London) **393**, 440 (1998).

[9] D. J. Watts, *Small Worlds: The Dynamics of Networks between Order and Randomness* (Princeton University Press, Princeton, NJ, 1999).

[10] M. E. J. Newman, SIAM Rev. **45**, 167 (2003).

[11] L. A. N. Amaral, A. Scala, M. Barthélémy, and H. E. Stanley, Proc. Natl. Acad. Sci. U.S.A. **97**, 11 149 (2000).

[12] B. J. Kim, Phys. Rev. E **69**, 045101(R) (2004).

[13] S. N. Dorogovtsev and J. F. F. Mendes, *The Evolution of Net-*

works: From Biological Nets to the Internet and WWW (Oxford University Press, Oxford, 2003).

[14] P. L. Krapivsky and S. Redner, Phys. Rev. E **63**, 066123 (2001).

[15] J. Davidsen, H. Ebel, and S. Bornholdt, Phys. Rev. Lett. **88**, 128701 (2002).

[16] E. M. Jin, Michelle Girvan, and M. E. J. Newman, Phys. Rev. E **64**, 046132 (2001).

[17] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin, Phys. Rev. Lett. **85**, 4633 (2000).

[18] P. Holme and B. J. Kim, Phys. Rev. E **65**, 026107 (2002).

[19] J.-L. Guillaume and M. Latapy, e-print cond-mat/0307095.

[20] R. Milo, N. Kashtan, S. Itzkovitz, M. E. J. Newman, and U. Alon, e-print cond-mat/0312028.

[21] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin, e-print cond-mat/0206132.

[22] M. E. J. Newman, Phys. Rev. E **68**, 026121 (2003).

[23] E. Ravasz, A. L. Somera, D. A. Mongru, Z. Oltvai, and A.-L. Barabási, Science **297**, 1551 (2002).

[24] D. Stauffer and A. Aharony, *Introduction to Percolation Theory* (Taylor & Francis, London, 1992).

[25] M. E. J. Newman, Phys. Rev. E **67**, 026126 (2003).

[26] M. Molloy and B. Reed, Random Struct. Algorithms **6**, 161 (1995).

[27] D. Heckathorn, Soc. Probl. **49**, 11 (2002).

[28] M. Salganik and D. Heckathorn, Sociol. Methodol. (to be published).

[29] S. P. Blythe, C. Castillo-Chavez, and G. Casella, Math. Popul. Stud. **3**, 199 (1992).

[30] L. Ancel-Meyers, M. E. J. Newman, M. Martin, and S. Schrag, Emerg. Infect. Dis. **9**, 204 (2003).

[31] M. Kretzschmar and M. Morris, Math. Biosci. **133**, 165 (1996).

[32] R. Cohen, K. Erez, D. ben-Avraham, and S. Havlin, Phys. Rev. Lett. **86**, 3682 (2001).

[33] D. J. Watts, Proc. Natl. Acad. Sci. U.S.A. **99**(9), 5766 (2002).

[34] D. Centola, M. Macy, and V. Eguiluz (unpublished).

[35] X. Guardiola, A. Diaz-Guilera, C. J. Perez, A. Arenas, and M. Llas, e-print cond-mat/0204141.

[36] D. Zanette, Phys. Rev. E **65**, 041908 (2002).

[37] S. Ellner, Ecology **84**, 882 (2003).

[38] All networks were rendered with computer code YED© http://www.yworks.com, free for academic use.