CCGRID 2003, Third International Workshop on Global and Peer-to-Peer Computing (GP2PC)

Criticality-based Analysis and Design of Unstructured Peer-to-Peer Networks as “Complex Systems”

Farnoush Banaei-Kashani[†] and Cyrus Shahabi[†]

Abstract— **Due to enormous complexity of the unstructured peer-to-peer networks as *large-scale*, *self-configure*, and *dynamic* systems, the models used to characterize these systems are either inaccurate, because of oversimplification, or analytically inapplicable, due to their high complexity. By recognizing unstructured peer-to-peer networks as “complex systems”, we employ statistical models used before to characterize complex systems for formal analysis and efficient design of peer-to-peer networks. We provide two examples of application of this modeling approach that demonstrate its power. For instance, using this approach we have been able to formalize the main problem with normal flooding search, propose a remedial approach with our *probabilistic flooding* technique, and find the optimal operating point for probabilistic flooding rigorously, such that it improves scalability of the normal flooding by 99%.**

Keywords— **Unstructured peer-to-peer networks, complex systems, search, criticality.**

I. Introduction

Literally, a peer-to-peer network is an interconnection of *peer* nodes, i.e., nodes with equivalent functionality. Peer-to-peer networks are often implemented as overlay networks and provide resource/object sharing services to their users (nodes)[1]. The overlay network is required as the communication media between nodes for information management purposes (e.g., to search cooperatively for a shared object). Numerous applications comply with such a communication and computing model [1]: file-sharing, distributed storage, cooperative computing, distributed computing, etc. Hence, there is an immense interest in peer-to-peer networks both at academia and commercially.

Despite the variety of peer-to-peer applications and services, the core operation in most peer-to-peer systems is to locate the shared objects efficiently (which are distributed among the nodes). Consequently, the architecture of the peer-to-peer network, i.e., the topology of the overlay network and placement of the location metadata (pointers to the locations of the objects), should be designed for efficient search.

Several studies propose *structured* networks. Assuming *long-lived* and *cooperative* nodes, one can structurize the network by assigning responsibility of maintaining each particular location metadata to particular node(s). With such a structure, location queries can be routed efficiently towards the node(s) holding the corresponding location metadata for the requested object. For example, with hierarchical directory servers [2, 3] and hierarchical match-making regions [4] the responsibility of maintaining the location metadata for an object is assigned to the nodes located at the same locality where the object exists, and with DHTs (Distributed Hash Tables) [5, 6, 7, 8] the responsibility is assigned by hashing object identifiers to node identifiers.

However, with some applications such as file-sharing [9, 10, 11], nodes are often short-lived and autonomous with limited cooperative nature [12]. In such a dynamic environment assigning responsibility of maintaining location metadata is either impossible or impractical. Therefore, such peer-to-peer networks are either *loosely structured* [13], by self-configuring into dynamic unguaranteed structures, or in the extreme case totally *unstructured* [14] (i.e., without any imposed architecture). Without structure, location queries cannot be efficiently routed towards the node that holds the corresponding location metadata because there is no global consensus on the location of the metadata. Therefore, with a naive approach the location query is flooded to the network so that by exhaustive coverage eventually the node holding the corresponding location metadata receives the query and responds. Obviously, this search approach is not scalable.

Here, we focus on the peer-to-peer applications like file-sharing, which do not yield to structured search. To address the scalability problem with search in unstructured peer-to-peer networks, the research community has taken initial steps to gain a better understanding of the characteristics of these systems by performing measurement studies [15, 12, 16]. Moreover, assuming partially cooperative nodes, some has suggested replication of the metadata [17] or caching query responses [18] for more efficient flooding. Besides, researchers have tried different variations of classical search techniques, other than normal flooding: directed BFS and iterative deepening [19], expanding ring [20], and random walk [21, 22, 20]. Although some of these techniques improve the scalability of search, due to enormous complexity of the unstructured networks as *large-scale*, *self-configure*, and *dynamic* systems, these techniques often fail to characterize the unstructured networks with models that are both accurate and applicable, at the same time. They either assume oversimplified models that fail to capture important characteristics of the unstructured networks, or apply highly complex models that render analytical evaluations impossible.

We propose recognizing unstructured peer-to-peer networks as “complex systems” [23] and employing the accurate statistical models extensively used to characterize these systems for formal analysis and efficient design of peer-to-peer networks. Complex systems such as thermodynamic systems, biological networks, and social networks are large-scale systems with numerous locally communicating entities/nodes. Optimal global behaviors/properties of complex systems *emerge* from simple local interactions between their entities at *critical* operation points. In the context of peer-to-peer networks, efficient collaborative search for an object based on local interaction protocols (such as the Gnutella protocol [14]), and self-configuration of the network to a topology with particular characteristics (such as a power-law topology [24]) are both examples of emergence. Emergence and power-law relations are signatures of complex systems observed at peer-to-peer networks.

Statistical physics [25] (originally used with thermodynamic systems) and percolation theory [26] are among the theories that provide statistical models useful for analyzing complex systems. Since complex systems are large-scale and dynamic, taking account of individual entities of these systems results in highly complex and impractical models. Instead, statistical models

[†] Farnoush Banaei-Kashani and Cyrus Shahabi are with the Integrated Media Systems Center, Computer Science Department, University of Southern California, Los Angeles, California 90089; email: [banaeika,shahabi]@usc.edu

[1] The Internet, based on its design principle, only provides the basic data transportation services. Value-added resource-sharing services should be implemented at the application layer based on the basic services provided by the Internet.

consider the complex systems as probabilistic systems and capture their statistical properties. Due to pure statistical effects originating in the law of large numbers, these models are able to characterize properties of large-scale networks accurately, while being applicable. With these theoretical models, we can formally characterize the unstructured networks, understand the problems with search in these systems, provide design guides for new search algorithms, and analyze the performance of the proposed search algorithms (including those mentioned above) and find the optimal operating points for their parameters.

This modeling approach is successfully exploited with other large-scale computer networks. For example, with the Internet *criticality-based* analysis has been used to explain the self-similar scaling behavior of the Internet traffic flows [27] and the Internet topology [28]. For the Internet, some researchers believe this approach is merely evocative and not explanatory, mainly because it avoids modeling the details of the Internet [29]. As compared to the Internet, where co-existence of various protocols and applications complicate the detailed semantics of the system, with peer-to-peer networks semantics are rare and insignificant. Besides, even for the Internet an evocative model can be as useful as explanatory models in terms of characterizing the behaviors of the system.

In this paper, we show usefulness of this modeling approach by providing two explicit examples. First, with percolation theory we formalize the main problem with normal flooding, propose a remedial approach with our *probabilistic flooding* technique, and find the optimal operating point for probabilistic flooding rigorously, such that it improves scalability of the normal flooding by 99%. Second, we use percolation theory and diffusion equations to explain self-avoidance behavior of random-walk processes at networks with various connectedness levels.

II. Criticality-based Modeling Exercise

We intend to study and apply the phase transition phenomenon in the context of peer-to-peer networks, as complex systems, to identify *resource-efficient* operating points (i.e., optimal level of control variables) for their various emerging global properties. With complex systems, global behaviors/structures *emerge* from local interactions between system entities. The emergent phenomenon is characterized by abrupt transition in the phase (quantity or quality) of a global system property, so-called order variable, at the *critical* level of one or more parameters that define the local interactions, so-called control variable(s). The sudden transition of the order variable between two phases with changes in the control variable(s) is historically known as *phase transition*[2] (see Figure 1).

Phase transition is theoretically and empirically studied extensively in the context of various complex systems such as thermodynamic systems, biological networks, graph structures, social networks, economic systems, etc. Here, after an introductory discussion about percolation theory, one of the theories that is developed to analyze and explain this phenomenon, we introduce two instance applications of this phenomenon in the context of search in unstructured peer-to-peer networks.

A. Percolation Theory and Efficient Search

Percolation theory [26] is a quantitative (statistical-theoretical) and conceptual model for understanding and analyzing many concepts associated with complex systems such as clustering, fractals, diffusion, and particularly phase transition.

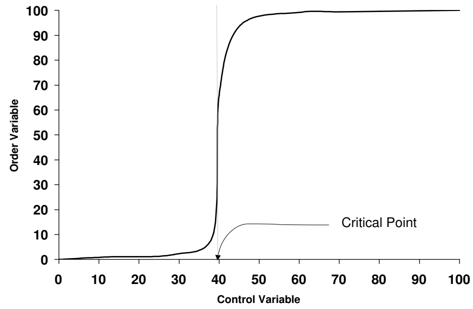

Fig. 1. Phase Transition

The significance of the percolation model is that many different problems can be mapped to the percolation problem [26]; e.g., forest-fire spread, oil field density estimation, diffusion in disordered media, etc. Gnutella [14], as a typical unstructured peer-to-peer network, has long been diagnosed with the scalability problem [30]. Gnutella does not scale because with its naive search approach, i.e., flooding, as the size of the network scales up the overhead imposed to each node exceeds the resources (particularly bandwidth) of the normal node; hence, fragmentation of the network is inevitable. On the other hand, empirical studies have verified that Gnutella network is a power-law network [15, 16]. Here, we intend to apply the percolation theory for power-law networks to analyze and address the scalability problem with unstructured peer-to-peer search algorithms.

First, we define a typical percolation model. Imagine a 2-dimensional lattice of dots, or so-called *sites*, large enough so that the effects from its boundaries are negligible (see Figure 2). Lines, or so-called *bonds*, are drawn between neighboring sites (i.e., sites in one hop distance from each other). Each bond can be open with probability p, or closed with probability $(1-p)$. A *cluster* (or component) is defined as a group of sites connected by open bonds. Percolation theory analyzes the statistical and geometrical properties of such clusters as the probability p changes; statistical distribution of the cluster size or cluster mass (number of sites within the cluster), cluster surface (length of the cluster perimeter), and cluster shape (fractal geometry of the cluster boundaries) are among the properties of interest in the percolation theory. We say a cluster *percolates* the lattice if it extends from one side of the lattice to the opposite side. The size of such a cluster, termed the *giant cluster*, is proportional to the size of the lattice. As p increases, emergence of the giant cluster for the first time marks the critical point where phase transition happens.

The percolation model described above (namely the lattice, short-range, bond percolation-model) has many different variants, each designed to model a particular problem. These variants differ based on the definition of the neighborhood (short-range versus long-range, and random versus deterministic), and definition of the nodes of the cluster (bonds versus sites). However, the principal analysis routines are the same with most of these variants. Scaling theories, namely renormalization group and finite-size scaling theory, are developed that generalize results on infinite-size, 2-dimensional lattice to infinite-size, infinite-dimensional lattice as well as finite-size lattice. Besides, more recently percolation is studied on various random graphs such as standard ER random graph [31], small-world network [32], and power-law network [33]. Percolation theory also extends to explain dynamic properties such as diffusion (or random walk) on the clusters of the percolation model.

[2] Phase transition was originally studied in the context of thermodynamic systems, as complex systems, in 70's. There it refers to sudden change in phase/form of the matter, e.g., from liquid to gas.

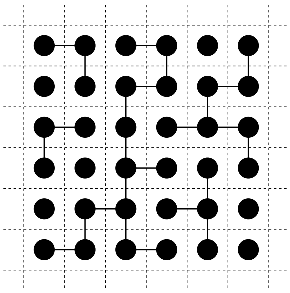

Fig. 2. Percolation lattice and clusters

B. *Probabilistic Flooding*

With flooding, the originator of a search sends a query to all its neighbors. The receivers of the query in turn forward the query to their neighbors, and so on. Each time the query is forwarded TTL (Time-To-Live) of the query is decremented by one. Therefore, the range of the query is restricted to the initial value of TTL set by the originator.

Although flooding the query results in the optimal response time, search by flooding is not scalable. As also recognized by other researchers [20], with an unstructured network, where in the extreme case there is no hints about the right direction for search, the best an efficient query can do is to cover as many nodes as possible in a reasonable time with minimal communications. Based on this view, the first time a node receives a query, the query is not considered as overhead, while duplicate queries are the main source of the overhead. Here, we introduce *probabilistic flooding* and by applying the percolation theory we prove that probabilistic flooding can be tuned to the optimal operating point where statistically duplicate queries are entirely eliminated. Therefore, with probabilistic flooding we can reduce the communication cost of the search to the minimal cost (i.e., the cost of coverage), which is less than 1% of the cost with normal flooding; hence, a scalable search algorithm.

With probabilistic flooding, a node forwards the query on a link to its neighbor with probability p, and drops the query with probability $(1-p)$. Normal flooding is an extreme case of the probabilistic flooding where p equals 1. Redundant connectivity of the unstructured network, i.e., existence of multiple (usually numerous) paths between two typical nodes, is the reason for existence of duplicate queries. Probabilistic flooding cuts the redundant paths effectively to eliminate duplicate queries and avoid the overhead cost of duplication. Tuning the value of p to an optimal operating point is crucial to effectiveness of the probabilistic flooding. At the optimal point, *statistically* the network is reduced to minimum connectivity such that full reachability is preserved while redundant paths are eliminated. Here, we apply percolation theory to find the optimal operating point.

The bond percolation model discussed before is an appropriate model for probabilistic flooding on unstructured networks: nodes are sites, links are bonds, and p is the probability of a bond being open. With $p=0$ all bonds are closed and there is no cluster at all. As p increases, isolated clusters start to appear. Percolation theory predicts the critical value of p at the phase transition where for the first time the cluster-fragments are unified to create a giant cluster (or giant component) that spans the network. Increasing p towards 1 furthermore only amplifies the connectivity (or so-called *strength*) of the giant cluster. Based on the above discussion, with probabilistic flooding the most dilute giant cluster, which appears at the critical point, results in the least communication cost. While this cluster spans the network for full reachability, it is not redundantly connected. Thus, for effective probabilistic flooding p must be tuned to its critical value at the phase transition. Here, we provide detailed criticality-based analysis to find the critical value of p.

B.1 Analysis

For a random graph of size N, $N \to \infty$, with connectivity distribution $P(k)$, percolation theory verifies that the criterion for criticality is as follows [34, 35]:

$$\frac{\langle k^2 \rangle}{\langle k \rangle} = 2$$

where the angular brackets denote ensemble average. With bond percolation, each link of the graph is cut with probability $(1-p)$, converting the actual connectivity distribution $P(k)$ of the graph to the *effective* distribution $P_e(k)$ as follows:

$$P_e(k) = \sum_{n=k}^{\infty} \binom{n}{k} p^k (1-p)^{n-k} P(n) \qquad (1)$$

To compute the critical probability p_c at which the effective graph is at phase transition we apply the criticality criterion:

$$\frac{\langle k^2 \rangle_e}{\langle k \rangle_e} = 2 \qquad (2)$$

but from equation (1), we can compute the first and the second moments of the effective distribution:

$$\begin{aligned}
\langle k \rangle_e &= \sum_{k=0}^{\infty} k \sum_{n=k}^{\infty} \binom{n}{k} p^k (1-p)^{n-k} P(n) \\
&= \sum_{n=0}^{\infty} P(n) \sum_{k=0}^{n} k \binom{n}{k} p^k (1-p)^{n-k} \\
&= p \sum_{n=0}^{\infty} n P(n) \\
&= p \ \langle k \rangle
\end{aligned}$$

and similarly:

$$\begin{aligned}
\langle k^2 \rangle_e &= \sum_{k=0}^{\infty} k^2 \sum_{n=k}^{\infty} \binom{n}{k} p^k (1-p)^{n-k} P(n) \\
&= \sum_{n=0}^{\infty} P(n) \sum_{k=0}^{n} k^2 \binom{n}{k} p^k (1-p)^{n-k} \\
&= \sum_{n=0}^{\infty} P(n) \left(np(1-p) + n^2 p^2 \right) \\
&= p^2 \ \langle k^2 \rangle + p(1-p) \ \langle k \rangle
\end{aligned}$$

therefore, from equation (2) we deduce:

$$p_c \tfrac{\langle k^2 \rangle}{\langle k \rangle} + (1-p_c) = 2 \quad \Rightarrow \quad p_c = \frac{1}{\alpha - 1} \qquad (3)$$

where $\alpha = \frac{\langle k^2 \rangle}{\langle k \rangle}$ is the ratio of the second to first moment of the actual graph.

As mentioned before, node connectivity in Gnutella network follows power-law distribution. Power-law graph is a special-case random graph. Here, we calculate α for a power-law graph. Following the traditional characterization of random graphs [33, 21], we use the generating-function formalism [36] to characterize the power-law graph. $G_0(x)$ is the generating function for the distribution of node-degree k:

$$G_0(x) = \sum_{k=0}^{\infty} p_k x^k \tag{4}$$

where p_k is the probability that a randomly chosen node on graph has degree k. From equation (4), one can deduce the n-th moment of the distribution:

$$\langle k^n \rangle = \Big[(x\frac{d}{dx})^n G_0(x) \Big]_{x=1} \tag{5}$$

For power-law graph, p_k can be defined as follows:

$$p_k = C k^{-\tau} e^{-k/\nu} \tag{6}$$

where C, τ, and ν are constants. The exponential cutoff factor is required for graphs representing real-world networks [37]; cutoff index ν is in the same order of magnitude as the maximum node degree in the graph. C is the normalization factor, given as $C = [\mathbf{Li}_\tau(e^{-1/\nu})]^{-1}$, where $\mathbf{Li}_\tau(x) = \sum_{k=1}^{\infty} \frac{x^k}{k^\tau}$ is the τ-th polylogarithm of x. With this definition, the generating function for power-law graphs is determined:

$$G_0(x) = \frac{\mathbf{Li}_\tau(xe^{-1/\nu})}{\mathbf{Li}_\tau(e^{-1/\nu})} \tag{7}$$

From equation (5), we can compute first and second moments of the power-law distribution[3]:

$$\begin{aligned} \langle k \rangle &= (x\frac{d}{dx})G_0(x)\Big|_{x=1} = \frac{\mathbf{Li}_{\tau-1}(xe^{-1/\nu})}{\mathbf{Li}_\tau(e^{-1/\nu})}\Bigg|_{x=1} \\ &= \frac{\mathbf{Li}_{\tau-1}(e^{-1/\nu})}{\mathbf{Li}_\tau(e^{-1/\nu})} \end{aligned}$$

and similarly:

$$\begin{aligned} \langle k^2 \rangle &= (x\frac{d}{dx})^2 G_0(x)\Big|_{x=1} = \frac{\mathbf{Li}_{\tau-2}(xe^{-1/\nu})}{\mathbf{Li}_\tau(e^{-1/\nu})}\Bigg|_{x=1} \\ &= \frac{\mathbf{Li}_{\tau-2}(e^{-1/\nu})}{\mathbf{Li}_\tau(e^{-1/\nu})} \end{aligned}$$

Therefore, for a power-law graph the ratio of the second to first moment α_{pl} can be computed as follows:

$$\alpha_{pl} = \frac{\langle k^2 \rangle}{\langle k \rangle} = \frac{\mathbf{Li}_{\tau-2}(e^{-1/\nu})}{\mathbf{Li}_{\tau-1}(e^{-1/\nu})}$$

which in turn gives the critical probability p_c:

$$p_c = \frac{1}{\alpha_{pl} - 1} = \frac{\mathbf{Li}_{\tau-1}(e^{-1/\nu})}{\mathbf{Li}_{\tau-2}(e^{-1/\nu}) - \mathbf{Li}_{\tau-1}(e^{-1/\nu})} \tag{8}$$

For Gnutella, the power-law exponent τ is estimated as low as $\tau = 1.4$ (see [24]) and as high as $\tau = 2.3$ (see [30]) in different cases/times. ν is in the range of 100 to 1000 (see [16]). In Figure 3, we illustrate p_c as a function of ν for various τ values. As depicted, the critical probability p_c can be less than 0.01; in the context of flooding-based search, it means the communication cost of probabilistic flooding can be less than 1% of that with normal flooding without losing the reachability.

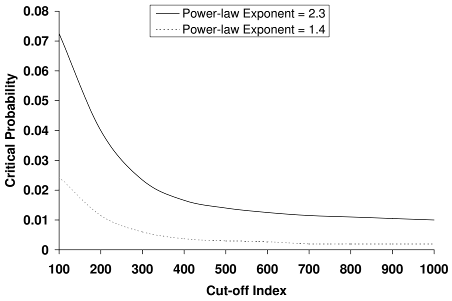

Fig. 3. Critical Probability

We prove unlike common impression of flooding-based search algorithms, search with probabilistic flooding seems to be scalable[4]. To make this search approach even more plausible, here, again we apply results achieved by statistical analysis of random graphs to address the second most important problem with flooding-based search, i.e., selecting appropriate initial value for TTL of the query. Applying the results provided by Newman et al. [33], we briefly propose a TTL selection policy that allows each node adaptively decide on the appropriate TTL value for its queries only based on the information collected locally; hence, scalable TTL selection.

In [33], the typical length λ of the shortest path between two randomly chosen nodes on any random graph is approximated as follows:

$$\lambda = \frac{\ln[(N-1)(z_2 - z_1) + z_1^2] - \ln(z_1^2)}{\ln(z_2/z_1)} \tag{9}$$

where z_i is the number of neighbors which are i hops away from the originator node, and N is an estimation of the total number of nodes available in the graph.

Typical path length between two nodes of the graph is a reasonable estimation of the distance between the originator of the query and the node that eventually serves the requested object. With an unstructured peer-to-peer network such as Gnutella, although activity of most network nodes is transient, the average number of the active nodes, N, is not heavily variant in short time-intervals [38]. Therefore, based on equation (9) each node can periodically estimate number of its first-level and second-level neighbors with local ping packets (with TTL=1 and TTL=2, respectively), and set TTL of its query packets to the estimated typical length of path between two nodes, λ, as observed by the node.

As a final note, we find it important to mention that here we are not advocating use of flood-based search instead of other search algorithms, although the probabilistic flooding approach described above seems to be a good candidate for search in unstructured networks. Instead, it is our intention to demonstrate the capabilities of the criticality-based analysis and the compliance of the nature of the unstructured peer-to-peer networks with the model that treats these systems as complex systems.

C. Random Walk or Diffusion

Different variations of random walk, a search approach in parallel with flooding-based search, are proposed by researchers

[3] Note: $\frac{d}{dx}\mathbf{Li}_\tau(x) = \frac{1}{x}\mathbf{Li}_{\tau-1}(x)$

[4] Currently, we are in the process of empirical verification of these results via Monte Carlo simulation.

[21, 20]. With random-walk search nodes forward the query to only one of their neighbors (selected randomly, or in some more intelligent way) instead of all neighbors, as compared to flooding-based search. This approach generally reduces the overhead of the query in trade with the increase in the time it takes to find the requested object.

With the percolation theory, researchers have extensively studied the notion of random walk as *diffusion* processes [39, 40, 26, 41]. Here, informally we provide an instance how the results of these studies can be exploited to understand some characteristics of random-walk processes at unstructured peer-to-peer networks and to derive design principles for efficient architect of these systems.

Consider the bond percolation model described at the beginning of the Section II-A[5]. Assume originator of a query initiates a large number of random-walk processes, termed random-walkers, to find a requested object. Variance θ of the distances (start-point to end-point) traversed by random-walkers can be used as an estimation of the size of the blob that contains the random-walkers:

$$\theta = \frac{1}{n}\sum_{i=1}^{n} R_i^2 - \left[\frac{1}{n}\sum_{i=1}^{n} R_i\right]^2 \qquad (10)$$

where R_i is the distance traversed by the i-th random-walker and n is number of random-walkers.

As mentioned in Section II-B, with unstructured networks the best an efficient query can do is to cover as many nodes as possible in a reasonable time with minimal communications. To achieve this goal, random-walkers must be *self-avoiding*, i.e., they should avoid re-traversing their own paths [21][6]. One can observe that θ, as a function of time, can be considered as a quality measure for self-avoidance: the faster θ (the linear blob size) grows with the time, the more self-avoiding the random-walkers. Assuming that at every time unit a random-walker hops once, θ is equivalently a function of the number of hops h made by the random-walkers.

With p (probability of a bond being open) equal 1, the central limit theorem characterizes θ as a function of h with the *normal* diffusion equation:

$$\theta = Dh \qquad (11)$$

where D, termed diffusivity, is a constant. With normal diffusion θ is proportional to h, and as we will see, this corresponds to the maximum growth rate of the variance, which in turn indicates the optimal self-avoidance. However, as p decreases towards zero, the normal diffusion equation does not hold anymore. Instead the diffusion, now termed subdiffusion, is characterized as follows:

$$\theta \propto h^{\gamma} \qquad (\gamma < 1) \qquad (12)$$

with γ monotonically decreasing as p converges to zero. Obviously, as far as self-avoidance is concerned, this behavior is undesirable. As $h \to \infty$, γ converges to the boundary points except at the critical point:

$$\begin{cases} \gamma \to 1 & (p_c < p < 1) \\ \gamma = \gamma_c & (p = p_c) \\ \gamma \to 0 & (0 \leq p < p_c) \end{cases} \qquad (13)$$

where γ_c $(0 < \gamma_c < 1)$ is the exponent of the *anomalous* diffusion at phase transition.

Percolation theory explains this behavior by analyzing the shape and strength of the clusters. Consider the inverse subdiffusion relation $h \propto \theta^{1/\gamma}$. With $p < p_c$, giant cluster does not exist and all clusters are finite. Therefore, with $h \to \infty$, θ cannot grow infinitely and $\gamma \to 0$. Of course, this state of the network is undesirable because it restricts the reachability. At $p = p_c$, the giant cluster appears for the first time. At the critical point the giant cluster is dilute and it is shaped as a fractal with high fractal dimension $1/\gamma_c$. The higher the dimension of a fractal, the more it takes for a random-walker to cover a region because of frequent back-tracks that decreases self-avoidance. At this state, although reachability is achieved, random-walkers are slow and cannot efficiently search the network. Finally, with $p > p_c$ as $p \to 1$, the strength of the giant cluster increases while its fractal dimension decreases, enabling random-walkers a more self-avoiding traversal of the network.

Thus, unlike probabilistic flooding which incurs less overhead with dilute network, random-walkers need more strengthful networks. Based on these observations, currently we are developing algorithms that enable network nodes to estimate the strength of the network (which is a global property) based on the local information and self-configure themselves to tune the strength for efficient random-walk search. It is important to note that tuning involves a tradeoff between network strength (characterized with p) and practical constraints, such as maximum connectivity of the nodes; these constraints limit the upper bound of p to be less than 1.

The percolation theory also studies diffusion fronts, the shape of the perimeter of the diffusion blob, as well as the probability distribution $P(r, h)$, the probability that a random-walker is at distance r from the originator after h hops. We let the reader imagine the uses of these analytic characterizations of random-walkers to design efficient random-walk processes for search in unstructured peer-to-peer networks.

III. Conclusion and Future Work

In this position paper, we introduced a novel approach to modeling, analysis, and design of unstructured peer-to-peer networks based on the rich theory of criticality and complex systems. We provided examples how this approach can enhance our analytical understanding of the unstructured peer-to-peer networks as well as ability to design efficient mechanisms (e.g., our probabilistic-flooding search mechanism improves the efficiency of normal flooding by up to 99%).

Currently, we pursue our interest in designing highly efficient information management mechanisms for peer-to-peer networks (such as search and replication techniques discussed here) based on the eye-opening theory of complex systems. There are two general conceptual frameworks proposed to study complex systems (see [23]), namely SOC (Self-Organized Criticality) and HOT (Highly Optimized Tolerance). In the context of peer-to-peer networks, SOC suggests an unstructured design while HOT leads to a loosely structured architect. We believe depending on the application characteristics both of these designs are meaningful. Therefore, we adopt both of the frameworks SOC and HOT with our architectures of peer-to-peer networks.

[5] This discussion is general and it is not constrained to 2-dimensional lattice percolation model.

[6] Here, we study self-avoidance not as a capability of the random-walkers, but as a characteristic of the shape of the network that makes self-avoiding random-walks possible in the first place.

Acknowledgments

This research has been funded in part by NSF grants EEC-9529152 (IMSC ERC) and IIS-0082826, and unrestricted cash gifts from Microsoft, NCR, and Okawa Foundation.

References

[1] A. Oram, *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, O'Reilly and Associates, 2001.

[2] M. van Steen, F.J. Hauck, and A.S. Tanenbaum, "A model for worldwide tracking of distributed objects," in *Proceedings of the TINA'96 Conference*, 1996.

[3] C. Shahabi and F. Banaei-Kashani, "Decentralized resource management for a distributed continuous media server," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 7, pp. 710–727, July 2002.

[4] B. Awerbuch and D. Peleg, "Concurrent online tracking of mobile users," in *In Proceedings of ACM SIGCOMM Symposium on Communication Architectures and Protocols*, 1991.

[5] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proceedings of ACM SIGCOMM '01*, August 2001.

[6] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of ACM SIGCOMM '01*, August 2001.

[7] B.Y. Zhao, J. Kubiatowicz, and A. Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing," Tech. Rep. UCB/CSD-01-1141, UCB, April 2001.

[8] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," in *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, November 2001, pp. 329–350.

[9] KaZaA, "Kazaa," 2002, http://www.kazaa.com/.

[10] Mojo Nation, "Mojo nation," 2002, http://www.mojonation.net/.

[11] Morpheus, "Morpheus," 2002, http://www.morpheus.com/.

[12] S. Saroiu, P.K. Gummadi, and S.D. Gribble, "A measurement study of peer-to-peer file sharing systems," in *Proceedings of Multimedia Computing and Networking (MMCN'02)*, January 2002.

[13] Freenet, "Freenet project," 2002, http://freenet.sourceforge.net/.

[14] Gnutella, "Gnutella RFC," 2002, http://rfc-gnutella.sourceforge.net/.

[15] M.A. Jovanovic, F.S. Annexstein, and K.A. Berman, "Scalability issues in large peer-to-peer networks - a case study of gnutella," Tech. Rep., ECECS Department, University of Cincinnati, 2001.

[16] M. Ripeanu, A. Iamnitchi, and I. Foster, "Mapping the gnutella network," *IEEE Internet Computing*, vol. 6, no. 1, pp. 50–57, January 2002.

[17] E. Cohen and S. Shenker, "Replication strategies in unstructured peer-to-peer networks," in *In Proceedings of ACM SIGCOMM 2002*, August 2002.

[18] A. Crespo and H. Garcia-Molina, "Routing indices for peer-to-peer systems," in *In Proceedings of the 22nd International Conference on Distributed Computing Systems*, July 2002.

[19] B. Yang and H. Garcia-Molina, "Efficient search in peer-to-peer networks," in *In Proceedings of the 22nd International Conference on Distributed Computing Systems*, July 2002.

[20] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *Proceedings of the 16th International Conference on supercomputing (ICS '02)*, June 2002.

[21] L. Adamic, R.M. Lukose, A.R. Puniyani, and B.A. Huberman, "Search in power-law networks," *Physics Review Letters*, , no. 64 046135, 2001.

[22] Q. Lv, S. Ratnasamy, and S. Shenker, "Can heterogeneity make gnutella scalable?," in *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02)*, 2002.

[23] J.M. Carlson and J. Doyle, "Complexity and robustness," *Proceedings of National Academy of Science of the USA*, vol. 99, pp. 2538–2545, February 2002.

[24] M. Jovanovic, "Modeling large-scale peer-to-peer networks and a case study of gnutella," M.S. thesis, University of Cincinnati, 2001.

[25] K. Huang, *Statistical Mechanics*, John Wiley and Sons, second edition, 1987.

[26] D. Stauffer and A. Aharony, *Introduction to Percolation Theory*, Taylor and Francis, second edition, 1992.

[27] Toru Ohira and Ryusuke Sawatari, "Phase transition in a computer network traffic model," *Physical Review E*, vol. 58, pp. 193195, 1998.

[28] R. Albert and A.L. Barab´asi, "Topology of evolving networks: Local events and universality," *Physical Review Letters*, vol. 85, pp. 5234–5237, 2000.

[29] W. Willinger, R. Govindan, S. Jamin, V. Paxson, and S. Shenker, "Scaling phenomena in the internet: Critically examining criticality," in *In Proceedings of the National Academy of Sciences of the USA*, 2001.

[30] CLIP2.com, "Gnutella: To the bandwidth barrier and beyond," November 2000, http://www.clip2.com/gnutella.html.

[31] B. Bollobás, *Random Graphs*, Academic Press, New York, 1985.

[32] C. Moore and M.E.J. Newman, "Exact solution of site and bond percolation on small-world networks," *Physical Review E*, vol. 62, no. 7059-7067, 2000.

[33] M.E.J. Newman, S.H. Strogatz, and D.J. Watts, "Random graphs with arbitrary degree distribution and their applications," *Physical Review E*, vol. 64, no. 026118, 2001.

[34] M. Molloy and B. Reed, "A critical point for random grraphs with a given degree sequence," *Random Structures and Algorithms*, vol. 6, pp. 161–180, 1995.

[35] R. Cohen, K. Erez, D. Avraham, and S. Havlin, "Resilience of the internet to random breakdowns," *Physical Review Letters*, vol. 85, no. 21, pp. 4626–4628, November 2000.

[36] H.S. Wilf, *GeneratingFunctionology*, Academic Press, second edition, 1994.

[37] L.A.N. Amaral, A. Scala, and M. Brathélémy, "Classes of small-world networks," *Proceedings of National Academy of Science of the USA*, vol. 97, no. 21, pp. 11149–11152, October 2000.

[38] Limewire.com, "Online gnutella statistics," 2002, http://www.limewire.com/.

[39] S. Havlin and D. Avraham, "Diffusion in disordered media," *Advances in Physics*, vol. 36, no. 6, pp. 695–798, 1987.

[40] A.B. Harris, Y. Meir, and A. Aharony, "Diffusion on percolation clusters," *Physical Review B*, vol. 36, no. 16, pp. 8752–8764, 1987.

[41] Š. Barta, "Anomalous diffusion on percolating clusters," *Physica A*, vol. 203, pp. 163–174, 1994.

[42] P. Cheeseman, B. Kanefsky, and W.M. Taylor, "Where the really hard problems are," in *In Proceedings of IJCAI-91*, 1991.

[43] D. L. Maremen and V. Lesser, "Problem structure and subproblem sharing in multi-agent systems," in *In Proceedings of the 3rd International Conference on Multi-Agent Systems*, 1998.

[44] R. Bejar, B. Krishnamachari, C. Gomes, and B. Selman, "Distributed constraint satisfaction in a wireless sensor tracking system," in *In Proceedings of the IJCAI-01 Workshop on Distributed Constraint Reasoning*, 2001.

[45] M. Yokoo, E.H. Durfee, T. Ishida, and K. Kuwabara, "The distributed constraint satisfaction problem: formalization and algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 10, no. 5, pp. 673–685, 1998.

[46] K. Sycara, K. Hirayama, and M. Yokoo, "The phase transition in distributed constraint satisfaction problems: first results," in *In proceedings of the International Workshop on Distributed Constraint Satisfaction*, 2000.

[47] D.J. Watts, P.S. Dodds, and M.E.J. Newman, "Identity and search in social networks," *Science*, vol. 296, pp. 1302–1305, 2002.

[48] Y. Hamadi, C. Bessière, and J. Quinqueton, "Backtracking in distributed constraint networks," in *In Proceedings of the 13th European Conference on Artificial Intelligence*, 1998.