On Static and Dynamic Partitioning Behavior of Large-Scale P2P Networks

Derek Leonard, *Student Member, IEEE*, Zhongmei Yao, *Student Member, IEEE*,
Xiaoming Wang, *Student Member, IEEE*, and Dmitri Loguinov, *Senior Member, IEEE*

***Abstract*—In this paper, we analyze the problem of network disconnection in the context of large-scale P2P networks and understand how both static and dynamic patterns of node failure affect the resilience of such graphs. We start by applying classical results from random graph theory to show that a large variety of deterministic and random P2P graphs almost surely (i.e., with probability $1 - o(1)$) remain connected under random failure if and only if they have no isolated nodes. This simple, yet powerful, result subsequently allows us to derive in closed-form the probability that a P2P network develops isolated nodes, and therefore partitions, under both types of node failure. We finish the paper by demonstrating that our models match simulations very well and that dynamic P2P systems are extremely resilient under node churn as long as the neighbor replacement delay is much smaller than the average user lifetime.**

***Index Terms*—Churn, dynamic resilience, graph disconnection, P2P.**

I. Introduction

During the recent explosion of P2P research, network resilience has become an important issue since forced user disconnection and graph partitioning significantly hinder the availability of the network to its users [17], [19], [29], [38]. The primary interest in this line of study is to understand how dynamic user arrivals and abrupt departures affect the connectivity (and sometimes other metrics) of the system. The original thrust [19], [20], [38] in this direction focused on *static* node failure, where a fully populated network experienced simultaneous node failure with independent probability p. While analytical results on the exact probability of disconnection under static failure are currently unavailable in the literature, prior analysis suggests that P2P networks are highly resilient to node faults and can survive the failure of up to 50% of the graph without significant degradation in performance [38].

Since users in P2P networks rarely fail simultaneously [4], a different approach [23], [26], [32] is to examine disconnection in *dynamic* systems, where users continuously join and leave the network according to some arrival/departure processes. The only analytical results available on the dynamic resilience of generic P2P networks correlate the rate of churn with user notification frequency [26] and examine how stabilization delays affect the consistency of Chord's finger table [23].

In this paper, we bridge the gap between static and dynamic disconnection analysis and show that the problem of graph partitioning under both types of failure can be reduced to computation of the probability that a P2P network develops at least one isolated[1] node during failure. Under the umbrella of this unifying model, we then derive a closed-form model for static resilience and examine the same issue in dynamic networks where users depart the system after spending random amounts of time online. Our results show that under p-percent static failure, almost every sufficiently large k-regular P2P graph G on n vertices remains connected with probability:

$$P(G \text{ is connected}) \approx e^{-n(1-p)p^k}. \tag{1}$$

Using degree $k = \log_2 n$ and the commonly used failure probability $p = 1/2$ [20], [38], it immediately follows that fully balanced Chord remains connected after half the nodes depart with probability $e^{-0.5} \approx 0.6$. Also notice that for $p < 1/2$, this probability converges to 1 (i.e., almost every graph is connected) as $n \to \infty$ and for $p > 1/2$, it converges to 0 (i.e., almost every graph is disconnected).

Outside of static resilience, our second result is the derivation of disconnection probabilities for dynamic systems, which frequently exhibit high levels of churn [4], [26] and are more mathematically elusive. To capture user behavior in such systems, we propose a simple node-failure model in which users stay in the system for random periods of time before deterministically failing at the end of their lifetime. To maintain a resilient system, we assume that each node monitors its neighbors and replaces them upon detecting their failure. Replacement delays S_i and lifetimes L_i are drawn from some (possibly heavy-tailed) distributions and generally determine the resilience of the system. We derive two models for the probability of user isolation ϕ and then demonstrate that many proposed k-regular P2P systems can survive N user joins without partitioning with probability at least:

$$P(G \text{ survives } N \text{ joins}) \geq \left(1 - \frac{\rho k}{(1+\rho)^k + \rho k - 1}\right)^N,$$

where $\rho = E[L_i]/E[S_i]$ is the ratio of the mean user lifetime to the mean neighbor replacement delay. To understand this result, consider the following example. Given a system with 5 million users that join the network once a day, $k = 12$ neighbors per node, mean user lifetime of 0.5 hours, and 1-minute search delay

Manuscript received January 26, 2006; revised June 20, 2006, December 11, 2006, and May 10, 2007. First published February 25, 2008; current version published December 17, 2008. Approved by IEEE/ACM Transactions on Networking Editor A. Somani. This work was supported by the National Science Foundation under Grants CCR-0306246, ANI-0312461, CNS-0434940, CNS-0519442, and CNS-0720571. An earlier version of the paper appeared in IEEE ICNP 2005.

The authors are with the Department of Computer Science, Texas A&M University, College Station, TX 77843 USA (e-mail: dleonard@cs.tamu.edu; mayyao@cs.tamu.edu; xmwang@cs.tamu.edu; dmitri@cs.tamu.edu).

Digital Object Identifier 10.1109/TNET.2007.911433

[1] A node is isolated when all of its neighbors are simultaneously in the failed state.

(i.e., $\rho = 30$), the network survives for 1 year without disconnecting with probability 99.99992%, for 100 years with probability 99.992%, and for 10,000 years with probability 99.2%. This implies that, even with a modest number of neighbors, P2P graphs remain connected for much longer than the anticipated session duration of their users.

This paper is organized as follows. Section II examines previous work. Section III discusses how isolated nodes affect graph connectivity under both static and dynamic node failure. Section IV focuses on static resilience and Section V discusses our lifetime model. Section VI provides an exact formula for the probability of isolation under the lifetime model and Section VII derives an asymptotic expansion of this result. We conclude the paper in Section VIII.

II. Background

A. Basics

We start by defining what we mean by the probability of an event occurring in an infinite sequence of graphs.

Definition 1: For a family of graphs $\{G_n\}_{n=1}^{\infty}$, where n is the number of nodes in G_n, property Q holds in *almost every* graph (or *almost surely*) if $\lim_{n\to\infty} P(G_n \text{ has } Q) = 1$.

We use "almost every" and "almost surely" interchangeably throughout the paper to mean that $P(G_n \text{ has } Q) = 1 - o(1)$. The next definition explains how the strength of a graph's "connectivity" can be expressed using the number of nodes needed to isolate each subset of the graph.

Definition 2: Consider a graph $G = (V, E)$, where V is the set of nodes and E is the set of edges, and some connected subset of nodes $S \subset V$. Define the *node boundary* of S to be $\partial S = \{v : (u,v) \in E, u \in S, v \in V \setminus S\}$ and *node expansion* of G to be the smallest ratio of the boundary size to the set size for all sets up to half the graph:

$$h(G) = \min_{S \subset V : |S| \le |V|/2} \frac{|\partial S|}{|S|}. \tag{2}$$

Large expansion means that each induced subset S contains few internal edges and is well connected to the remainder of the graph $V \setminus S$. As we discuss below, if $|\partial S|$ is proportional to set size $|S|$, the probability of disconnection in G can be reduced to that of node isolation.

B. Random Graph Theory

One of the first approaches to network reliability stems from random graph theory. The issue of partitioning and disconnection of random graphs $G(n,p)$ has a long history [13]. It is well-known that, as with any other monotone property, connectivity of $G(n,p)$ experiences a sharp transition from "almost never" to "almost always" at the threshold $p = \log n/n$; however, a more powerful result states that almost every random graph $G(n,p)$, $G(n,M)$, $G(n,k_{out})$ [7], [33] is connected *if and only if it has no isolated nodes*. Defining $\Phi(G_n)$ to be the probability that a random graph remains connected under p-percent node or edge failure and assuming that X is the number of isolated nodes in the graph immediately after the failure, the following holds [7]:

$$\lim_{n\to\infty} P\left(\Phi(G_n) = P(X=0)\right) = 1. \tag{3}$$

C. Deterministic Graphs

After some technical manipulation, a result similar to (3) can be shown to hold for certain deterministic networks as well. For example, Burtin [8] and later Bollobas [6] prove that under independent uniform failure, hypercubes are almost surely connected if and only if they have no isolated nodes. Intuitively, this result means that the conditional probability that a hypercube partitions along a set boundary ∂S, for some non-trivial set S, while having no isolated nodes is $o(1)$ as $n \to \infty$. We leverage these observations later in the paper.

Connectivity of *generic* deterministic graphs $G = (V, E)$ under independent node failure has also received significant attention in the literature [5], [16], [21]. In this line of work, $\Phi(G)$ is called *residual node connectivity* and can be expressed as [5]:

$$\Phi(G) = \sum_{i=1}^{n} S_i(G) p^{n-i} (1-p)^i, \tag{4}$$

where p is the failure probability of each node and $S_i(G)$ is the number of connected induced subgraphs of G with exactly i nodes. While this closed-form expansion is beneficial for simple graphs (such as trees), computation of $\Phi(G)$ for a generic graph requires the knowledge of an NP-complete metric $S_i(G)$, whose expression is unknown even for the basic hypercube [39].

Najjar and Gaudiot [31], however, noticed that several non-hypercube deterministic networks frequently develop disconnections around individual nodes rather than along boundaries of larger sets S, $|S| \ge 2$. This led to the following approximate model for the probability that an n-node, k-regular graph partitions under p-percent node failure [31]:

$$\Phi(G) \approx \sum_{i=0}^{n} Q_i \binom{n}{i} p^i (1-p)^{n-i}, \tag{5}$$

where

$$Q_i = \prod_{j=1}^{i} \left[1 - \frac{k(n-k-1)!(j-1)!(n-j)}{(n-1)!(j-k)!} \right]. \tag{6}$$

Other approaches that study disconnection of hypercubes include [11], [14], [18], [24]; however, none of them provide a practically usable model that is both accurate and simple to evaluate.

D. P2P Resilience

Given the wide variety of recently developed P2P systems, several techniques have been employed to evaluate the resilience of such graphs. One commonly used method is to monitor several performance metrics (e.g., percentage of successful queries, graph connectivity, consistency of links) under node failure and show how they change depending on system parameters. A seminal paper in this genre written by Gummadi *et al.* [19] explores the impact of different routing geometries on the static resilience of the graph, which is defined as the ability of the graph to route messages *before* the designed recovery algorithm repairs the graph. Other papers that examine static resilience in a similar fashion are [27], [34], and [36]. A more recent study by Chun *et al.* [12] uses simulations to analyze the impact of different types of neighbor-selection algorithms on static resilience of

P2P graphs under both random node failures and targeted attacks. The paper demonstrates that there is a distinct tradeoff between resilience and system performance.

The second approach is more analytical in nature. Chord [38] and Koorde [20] show that under independent uniform node failure, k-regular graphs require degree $k \geq \log_{1/p} n$ in order to upper-bound the probability of individual node isolation by $1/n$. Massoulie *et al.* [17], [29] develop a new P2P system based on random graphs and derive the probability that it remains connected under p-percent failure. Liben–Nowell *et al.* in [26] study the dynamic nature of P2P systems in regards to joins and unexpected departures and their impact on routing efficiency. The authors derive a lower bound on the number of users a node must be notified about in order for the system to avoid disconnection. In a more recent paper, Krishnamurthy *et al.* [23] focus on predicting the state of each finger pointer in a Chord system under dynamic failure conditions. They derive a probabilistic characterization of each neighbor and successor pointer, which allows them to obtain models for the percentage of failed queries in the system under user churn.

III. Unifying Model of Disconnection

In this section, we discuss how connectivity of P2P systems under static and dynamic node-failure patterns can be reduced to the problem of node isolation.

A. Generic Disconnection Model

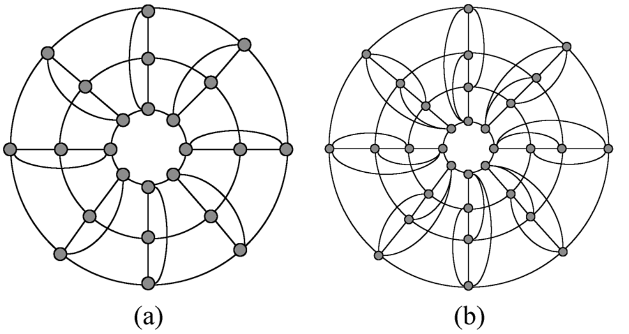
We first turn to the question of what properties a graph G must possess in order to satisfy (3) under random edge and node failure. Interestingly, the property that makes hypercubes (and classical random graphs) very unlikely to partition into non-trivial subgraphs without developing isolated nodes is that the number of edges leaving each set S is a certain *increasing* function of set size $|S|$. Burtin [8] showed that for each connected subset $S \subset V$ in a hypercube with $|S| \leq n/2$, the size of its edge boundary is at least:

$$|\{(u,v) \in E : u \in S, v \in V \setminus S\}| \geq |S| \left(k - \log_2 |S|\right), \quad (7)$$

where $k = \log_2 n$ is the degree of the graph. Condition (7) states that larger sets S are *always* better connected than smaller sets and ensures that the probability that any large subgraph disconnects after node failure is negligible compared to that of individual node isolation.

We are aware of only one effort to extend this result to generic graphs G. Najjar and Gaudiot [31] noticed that several other types of deterministic networks frequently develop disconnections around individual nodes rather than along boundaries of larger sets S, $|S| \geq 2$. This led to the conjecture that (3) holds for all k-connected graphs in which every set S exhibits sufficient boundary size $|\partial S|$.

Conjecture 1 (Najjar [31]): For any k-regular, k-node-connected graph $G = (V, E)$ in which the boundary $|\partial S|$ of every connected subset $S \subset V$ with $2 \leq |S| < |V|/2$ is larger than k, (3) holds under random node failure.


(a)

(b)

Fig. 1. (a) Graph $F_{24,4}$. (b) Graph $F_{32,5}$. Both have $m = 8$.

TABLE I
Two Examples of Disconnection for $n = 16384, k = 14$

p	P(connected)		P(no isolated)		$q(G)$	
	$F_{n,k}$	H_n	$F_{n,k}$	H_n	$F_{n,k}$	H_n
.3	.92946	.99946	.99822	.99946	.025	1
.35	.44712	.99564	.98789	.99564	.022	1
.4	.01427	.97403	.93590	.97403	.065	1
.45	0	.88236	.75723	.88239	.243	.999
.5	0	.60848	.36950	.60873	.631	.999

Note that the difference between this condition and (7) is that the latter requires that $|\partial S|$ grow as a function of set size $|S|$, while the former is lower-bounded by a constant that does not depend on $|S|$. Even without analyzing the proofs in [8], it is not difficult to see that this differences makes Conjecture 1 false. One simple counter-example is a graph we call $F_{n,k}$, which is a fusion of $k-1$ cycles constructed as follows. Start with $k-1$ cycles of size $m = n/(k-1)$ and assign sequential labels $c_{i1}, \ldots, c_{im}$ to all nodes in the ith cycle $(i = 1, \ldots, k-1)$. Then connect each node c_{ij} to $k-2$ other nodes in the same position j in other cycles: $c_{1j}, \ldots, c_{i-1,j}, c_{i+1,j}, \ldots, c_{k-1,j}$. An example of $F_{n,k}$ is shown in Fig. 1 for $k = 4$ and $k = 5$.

Notice that $F_{n,k}$ is k-regular, k-node-connected, and compliant with Conjecture 1; however, its bisection width is only $2(k-1)$ nodes regardless of the size of the graph, which contributes to its tendency to partition along the "weaker" dimension (i.e., along nodes $c_{1j}, \ldots, c_{k-1,j}$ for some j) as $n \to \infty$. It can be shown that by properly selecting p, a sequence of graphs $F_{n,k}$ can be constructed such that $P(X = 0)$ converges to 1 and $\Phi(G)$ to any constant in [0,1) as $n \to \infty$. We omit these results for brevity and instead provide simulations that demonstrate a similar result.

Before simulating $F_{n,k}$, we need another metric. Denote by $q(G)$ the fraction of disconnection events that contain at least one isolated node:

$$q(G) = P(X > 0 | G \text{ is disconnected}) = \frac{P(X > 0)}{1 - \Phi(G)}, \quad (8)$$

where as before X is the number of isolated nodes after the failure. Notice that in graphs satisfying (3), metric $q(G)$ must tend to 1 as $n \to \infty$.

Table I shows simulation results for $F_{n,k}$ and the hypercube H_n for $n = 16384$ and degree $k = 14$ (100,000 iterations). The three columns in the table contain 1) the probability that G is connected after node failure, 2) the probability that it has no isolated vertices, and 3) metric $q(G)$. Notice in the table that hypercubes are very likely to be connected if they have no isolated nodes and that their $q(G)$ is close to 1. This means that when the graph does partition, it almost certainly has at least one isolated node. At the same time, observe in the table that condition (3)

TABLE II
SIMULATIONS WITH DEGREE-REGULAR (FULLY BALANCED) DHTS

p	Chord: $n = 16384$, $k = 27$			CAN: $n = 16384$, $k = 14$			de Bruijn: $n = 20736$, $k = 24$			Pastry: $n = 15625$, $k = 24$		
	$\Phi(G)$	$P(X=0)$	$q(G)$	$\Phi(G)$	$P(X=0)$	$q(G)$	$\Phi(G)$	$P(X=0)$	$q(G)$	$\Phi(G)$	$P(X=0)$	$q(G)$
.4	.99999	.99999	1	.97321	.97321	1	.99999	.99999	1	1	1	N/A
.45	.99999	.99999	1	.88093	.88098	.9996	.99995	.99995	1	1	1	N/A
.5	.99996	.99996	1	.60704	.60735	.9992	.99930	.99930	1	.99950	.99950	1
.55	.99918	.99918	1	.18308	.18372	.9992	.99444	.99444	1	.99535	.99535	1
.6	.99354	.99354	1	.00645	.00661	.9998	.96181	.96194	.9966	.97105	.97105	1
.65	.95001	.95004	.9994	0	0	.9999	.79535	.79556	.9989	.83755	.83760	.9997
.7	.72619	.72650	.9988	0	0	1	.31999	.32119	.9982	.41305	.41395	.9985
.75	.17877	.18047	.9979	0	0	1	.00792	.00816	.9998	.02045	.02140	.9990
.8	.00040	.00043	.9999	0	0	1	0	0	1	0	0	1

TABLE III
SIMULATIONS WITH DEGREE-IRREGULAR GRAPHS FOR $n = 16384$

p	Symphony: $k_{out} = 14$			$G(n, k_{out}) : k_{out} = 14$			Randomized Chord: $k_{out} = 14$			RZ Chord: $k_{out} = 14$		
	$\Phi(G)$	$P(X=0)$	$q(G)$	$\Phi(G)$	$P(X=0)$	$q(G)$	$\Phi(G)$	$P(X=0)$	$q(G)$	$\Phi(G)$	$P(X=0)$	$q(G)$
.4	.99999	.99999	1	.99316	.99316	1	.99999	.99999	1	.9444	.9455	.9802
.45	.99998	.99998	1	.96609	.96609	1	.99999	.99999	1	.9057	.9089	.9661
.5	.99768	.99768	1	.86257	.86260	.9998	.99971	.99971	1	.8186	.8243	.9686
.55	.98750	.98750	1	.58042	.58064	.9995	.99747	.99747	1	.6248	.6367	.9683
.6	.93914	.93917	.9995	.17081	.17148	.9992	.98443	.98443	1	.3193	.3370	.9739
.65	.75520	.75527	.9997	.00547	.00560	.9998	.91624	.91625	.9999	.0585	.0673	.9907
.7	.31153	.31205	.9992	0	0	1	.63749	.63772	.9994	.0006	.0009	.9997
.75	.01269	.01296	.9997	0	0	1	.12993	.13076	.9990	0	0	1
.8	0	0	1	0	0	1	.00028	.00029	.9999	0	0	1

does not hold for $F_{n,k}$. For example, for $p = 0.4$, $\Phi(G) = 0.014$ while $P(X = 0)$ is over 0.93 and the probability that a partitioned graph has at least one isolated node is only 0.065.

While necessary conditions on G for (3) to hold are generally unknown, one can formulate a simple sufficient condition as follows.

Definition 3: A sequence of graphs $\{G_n\}_{n=1}^{\infty}$ is called *asymptotically strong*, if their expansion $h(G_n)$ is no smaller than that of hypercubes H_n or random graphs $G(n,p)$, $G(n,k_{out})$, $G(n,M)$ of the same size.

Using this definition, the following useful approximation immediately follows.

Observation 1: Almost every asymptotically strong graph G_n can be approximated as connected under random node failure if and only if it has no isolated nodes.

This statement is purposely generic so as to apply to a variety of different graphs. For example, this condition holds for DHTs that are isomorphic to or can be reduced to the hypercube, which includes Chord [38], logarithmic CAN with $d = \Theta(\log n)$ [34], randomized Chord [28], Tapestry [43], and Pastry [36]. It also holds for graphs that have better expansion than hypercubes (e.g., de Bruijn [27]) as long as $k = \Omega(\log n)$ and certain types of unstructured networks similar to $G(n, k_{out})$ that rely on unconstrained selection of neighbors during join.

Although traditional graph theory refers to graphs of asymptotically large size, extensive simulations below demonstrate the application of (3) to P2P graphs of *finite* size.

B. Static Resilience

Recall that static resilience alludes to the connectivity of a graph G after each node is removed from the graph independently with probability p. We next present simulation results of $\Phi(G)$, $P(X = 0)$, and $q(G)$ for a number of degree-regular and irregular P2P systems using 100,000 node-failure patterns for each value of p. To deal with directed graphs, we assumed that each node's in-degree and out-degree neighbors contributed to its resilience and that isolation happened when a node lost all of its in- *and* out-degree neighbors. Similarly, a directed P2P network was considered partitioned (disconnected) when its *undirected* version was, which is a measure of weak connectivity of directed graphs.

For each directed P2P system, denote by k_{out} its out-degree. Table II shows the above three metrics for degree-regular *fully populated* DHTs Chord [38] with $k_{out} = 14$ and $k = 27$, undirected CAN [34] with $k = 14$, de Bruin [20] with $k_{out} = 12$ and $k = 24$, and undirected Pastry [36] with $k = 24$. In all cases, we only simulate the default neighbors used in routing and do not consider auxiliary neighbor sets (e.g., Chord's successor list or Pastry's leaf set). As shown in the table, $\Phi(G)$ is very close to $P(X = 0)$ for all graphs and all values of p. Further notice that $q(G)$ is at least 0.9966, which confirms that an overwhelming majority of disconnections in these graphs occur with at least one isolation.

For degree-irregular graphs, we use fully populated Symphony [28], $G(n, k_{out})$ (i.e., a random graph with a fixed out-degree k_{out} and uniform neighbor selection) that models generic unstructured systems, fully populated Randomized Chord [28], and Random-Zone (RZ) Chord (i.e., Chord with a random partitioning of the circle). Table III demonstrates that these systems also follow the classical result well. Besides the fact that $\Phi(G)$ is very close to $P(X = 0)$, notice in Table III that the resilience of Chord with random zone sizes is inferior to balanced Chord since there is more possibility for nodes with smaller-than-average degree to disconnect the graph.[2] These nodes are also responsible for the deviation of $q(G)$ in RZ Chord from 1.0, which is a phenomenon that disappears as the number of nodes becomes larger (not shown for brevity).

C. *Dynamic Resilience*

While the use of p-percent uniform node failure provides an accurate approximation of actual network behavior in some cases, it has been noted that it has questionable applicability to real P2P networks [4], [26] where users join and leave the system asynchronously based on their individual browsing habits. One approach to modeling such systems is to assign each joining user a random lifetime L_i, which determines the duration that node i stays in the system before abruptly (i.e., without graceful notification of its neighbors) departing from the network and represents the amount of time a user spends in the network browsing for content and/or providing services to other peers.

Most structured P2P systems [28], [34], [38] use DHT-specific neighbor-replacement algorithms to repair the zones of failed nodes and maintain consistency of routing. Certain unstructured systems [10] also explicitly perform replacement of failed neighbors to achieve the desired level of routing and search performance. In addition to maintaining consistency of routing [38] and avoiding congestion in the graph [10], neighbor replacement serves the purpose of keeping the system resilient to disconnection. We next examine the question of how quickly failed neighbors should be replaced and what levels of resilience one should expect from churn-based P2P networks.

Throughout the paper, we assume that each node performs a "search" to find new neighbors as soon as it detects the failure. At this stage, we are not concerned with how this is accomplished and combine both failure detection and repair into a generic random variable S_i that measures the total delay required to perform these operations. Given this new paradigm of node-failure, we now define the probability ϕ that a given user v becomes isolated *during its lifetime* because its neighbors are failing at a faster rate than v is able to obtain their replacements from among the remaining nodes. We derive ϕ in later parts of the paper; however, we now show how the knowledge of this *local* metric can be used to study *global* resilience of lifetime-based P2P networks.

Define Z to be the random time (in terms of user joins) when graph G disconnects for the first time. Then assuming that G is asymtotically strong and each joining node i is assigned a Bernoulli random variable X_i that determines whether the user is isolated from the network during its lifetime, the probability that almost every graph stays connected for more than N user joins can be computed as:

$$P(Z > N) = P\left(\bigcap_{i=1}^{N}[X_i = 0]\right) \approx \prod_{i=1}^{N}\left(1 - E[X_i]\right), \tag{9}$$

where the independence between X_i is approximated from the asymptotically strong nature of the graph (see below for the reasoning).

TABLE IV
LIFETIME SIMULATIONS OF THE PROBABILITY $P(Z > N)$ THAT THE NETWORK SURVIVES AT LEAST N USER JOINS (FIXED SEARCH DELAYS)

Search	CAN: $N = 10^6$		RZ Chord: $N = 50,000$	
delay	Simulations	Model (10)	Simulations	Model (9)
6 min	.9732	.9728	.6295	.6251
7.5 min	.8118	.8124	.3284	.3184
8.5 min	.5669	.5659	.2189	.2206
9 min	.4065	.4028	.1460	.1483
9.5 min	.2613	.2645	.1211	.1274
10.5 min	.0482	.0471	.0493	.0493

For k-regular graphs, each user has the same probability of isolation (i.e., $E[X_i] = P(X_i = 1) = \phi$) and the above reduces to:

$$P(Z > N) \approx (1 - \phi)^N. \tag{10}$$

We next verify the applicability of (10) using simulations, where both $E[X_i]$ and ϕ are computed empirically, and defer the task of modeling ϕ until Section V. The simulations use two types of DHTs and two distributions of lifetimes: exponential with CDF $1 - e^{-\lambda x}$ and shifted Pareto with CDF $1 - (1 + x/\beta)^{-\alpha}$. The first system under study is a 12-regular fully populated CAN with exponential lifetimes, $\lambda = 2$ (mean lifetime 30 minutes), $n = 4096$ users, and $N = 10^6$. The second system is RZ Chord with Pareto lifetimes, $\alpha = 3, \beta = 1$ (mean lifetime also 30 minutes), $n = 128$ users, $k \approx 13$ (out-degree 7), and $N = 50,000$.

Simulation results are shown in Table IV, where both models (9), (10) match $P(Z > N)$ well. Observe in the table that zone-balanced CAN is significantly more resilient than RZ Chord since the latter frequently develops isolation around nodes with smaller-than-average degree. In fact, the resilience of CAN is quite impressive as it can survive 1 million user joins with probability 0.97 using 6-minute replacement delays. A more important result in Table IV, however, is that dynamic systems also stay connected as long as they do not contain isolated users, which confirms that resilience of P2P graphs under both types of failure can be reduced to the likelihood of user isolation.

IV. STATIC ISOLATION MODEL

This section develops a simple closed-form model for $P(X = 0)$, i.e., the probability that the graph contains no isolated nodes, under static node failure and compares this result to simulations of $\Phi(G)$. In the next section, we address the issue of dynamic node failure and derive a model for ϕ.

A. *Isolated Nodes*

Assume that each node i has k_i neighbors in some graph G and again define X_i to be a Bernoulli indicator variable of whether node i is isolated or not after each node is removed from the system with independent probability p:

$$X_i = \begin{cases} 1 & \text{isolated and alive} \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Denote by $p_i = P(X_i = 1) = (1 - p)p^{k_i}$ the probability that i is isolated and alive after the failure. Next, notice that $\{X_i\}$ may be identically or non-identically distributed, but they are

[2] More analysis of zone size distributions in DHTs can be found in [40].

almost certainly *dependent*. However, as $n \to \infty$, this dependency in asymptotically strong graphs becomes negligible and $\{X_i\}$ behave *as if* they were independent [2], [7]. This is a consequence of the fact that in P2P graphs under study, any two nodes i and j have a *fixed* number of common neighbors (e.g., in Chord, it is two), which becomes negligible compared to the total degree of each user $k_i = \Omega(\log n)$ as $n \to \infty$.

Next, let $X = \sum_{i=1}^{n} X_i$ be the total number of isolated nodes in G. Applying Markov's inequality $P(X \geq 1) \leq E[X]$, we directly obtain the next lower bound on the connectivity of the system.

Theorem 1: For asymptotically strong graphs, the following lower bound holds almost surely:

$$\Phi(G) \geq 1 - \sum_{i=1}^{n} p_i. \tag{12}$$

While this bound is very tight for small p and is better than those shown in [11] for all values of p, it produces negative values for sufficiently high failure rates. To overcome this limitation, an alternative approach is to notice that X is in fact a sum of a large number of Bernoulli random variables with certain well-know asymptotic properties. Due to the diminishing dependency between $\{X_i\}$ as $n \to \infty$, we apply the Chen–Stein method [2] to X and next obtain a much tighter result on $\Phi(G)$.

Theorem 2: For asymptotically strong graphs and probability of failure p satisfying the convergence conditions of the Chen–Stein theorem [2], the following holds: 1) random variables $\{X_i\}$ behave almost identically to a collection of independent variables with the same marginal distributions; 2) the number of isolated vertices X tends to a Poisson distribution with mean $\lambda = \sum_{i=1}^{n} p_i$; and 3) the probability $\Phi(G)$ of having a connected graph is $e^{-\lambda}$ almost surely.

We should note that the exact condition on p for this approximation to hold depends on the graph under study and the size of the overlap between different neighbor sets. Without this knowledge, a general rule of thumb is to ensure that $\sum_{i=1}^{n} p_i^2 \to 0$ as $n \to \infty$. We next show how well this approximation holds in both degree-regular and irregular P2P systems for several values of p. For degree-regular networks, Theorem 2 simplifies to a trivial closed-form expression:

$$\Phi(G) \approx e^{-n(1-p)p^k}. \tag{13}$$

To verify (13), we compare $\Phi(G)$ calculated in simulations over 100,000 node failure patterns to that of the model in Table V for fully balanced Chord [38] with $k = 27$ ($n = 16384$) and de Bruijn graphs [20] with $k = 24$ ($n = 20736$). As the table shows, simulations follow the model quite well for each graph over all values of p. For comparison purposes, the table also plots Najjar's model (5), which is surprisingly less accurate than (13) and significantly more complex to compute.

TABLE V
SIMULATION RESULTS AND MODEL (13) FOR TWO REGULAR GRAPHS

p	Chord: $n = 16384$, $k = 27$			de Bruijn: $n = 20736$, $k = 24$		
	$\Phi(G)$	Model	Najjar	$\Phi(G)$	Model	Najjar
.4	.9999	1	.9986	.9999	.9999	.9955
.45	.9999	1	.9984	.9999	.9999	.9948
.5	.9999	.9999	.9982	.9993	.9994	.9940
.55	.9992	.9993	.9976	.9944	.9945	.9892
.6	.9935	.9933	.9916	.9618	.9615	.9550
.65	.9500	.9503	.9463	.7954	.7907	.7750
.7	.7262	.7239	.7055	.3199	.3037	.2737
.75	.1788	.1766	.1501	.0079	.0055	.0033
.8	.0004	.0004	.0002	0	10^{-9}	10^{-10}

TABLE VI
SIMULATION RESULTS AND MODEL (14) FOR THREE IRREGULAR GRAPHS

p	Symphony		$G(n, k_{out})$		Randomized Chord	
	$\Phi(G)$	Model	$\Phi(G)$	Model	$\Phi(G)$	Model
.4	.9999	.9999	.9932	.9934	.9999	.9999
.45	.9998	.9996	.9661	.9666	.9999	.9999
.5	.9977	.9977	.8626	.8646	.9997	.9997
.55	.9875	.9875	.5804	.5829	.9975	.9976
.6	.9391	.9394	.1708	.1700	.9844	.9845
.65	.7552	.7535	.0055	.0053	.9162	.9151
.7	.3115	.3107	0	10^{-7}	.6375	.6372
.75	.0127	.0122	0	10^{-15}	.1299	.1282
.8	0	10^{-7}	0	10^{-34}	.0003	.0002

While many ideal DHTs are degree-regular, their instances under random node join and departure often exhibit degree irregularity that depends on random partitioning of the DHT space (e.g., zone-size distribution in Chord). Additional degree-irregular graphs include DHTs in which the in-degree is random (e.g., Symphony, Randomized Chord [28]) and unstructured P2P systems such as Gnutella. For such graphs, we obtain the probability of disconnection under static failure:

$$\Phi(G) \approx e^{-(1-p)\sum_i p^{k_i}} \approx e^{-n(1-p)E[p^{k_i}]}, \tag{14}$$

where $\sum_i p^{k_i}$ is approximated by $nE[p^{k_i}]$, treating k_i as a random variable.

To compute this model, we first use simulations to obtain $E[p^{k_i}]$ and then utilize this value in (14). Simulations of $\Phi(G)$ for Symphony [28], $G(n, k_{out})$ [7], and Randomized Chord [28], all with degree $k_{out} = 14$ and 16384 nodes, are shown in Table VI, which demonstrates that the model follows simulation results very accurately for all values of p.

To our knowledge there are no results on this topic for degree-irregular graphs with which to compare our model. As Najjar's result (5) is based on a complicated combinatorial argument that only applies to k-regular graphs, it cannot be easily extended to degree-irregular networks.

B. Discussion

The results of this section have confirmed that large-scale P2P networks generally disconnect through isolated nodes, both in degree-regular and irregular cases. Metric $q(G)$ in all simulations remained between 0.968 and 1, where deviation from 1 was more apparent in smaller graphs and cases when the degree of certain nodes was allowed to become much smaller than average (e.g., in RZ Chord). For larger graphs (hundreds of thousands or millions of nodes), the agreement between $\Phi(G)$ and $P(X = 0)$ becomes stronger (simulations not shown for brevity).

V. Concept of Dynamic Isolation

Using lifetime-based ideas developed in Section III, the rest of the paper deals with deriving the probability ϕ that all k neighbors of a given node v are simultaneously in the failed state before the lifetime of v expires. We start by formalizing churn-based P2P systems and explaining our assumptions.

A. Assumptions

Previous research suggests that the distribution of user lifetimes in real systems is often heavy-tailed (e.g., Pareto) [9], [37], where most users spend very little time browsing the network, while a small group remains logged in for weeks at a time providing services to other peers. Thus, to allow arbitrarily small lifetimes, we use a shifted Pareto distribution $F(x) = 1 - (1 + x/\beta)^{-\alpha}, x > 0, \alpha > 1$ to represent heavy-tailed user lifetimes, where scale parameter $\beta > 0$ can change the mean of the distribution without affecting its range $(0, \infty]$. The mean of this distribution $E[L_i] = \beta/(\alpha - 1)$ is finite only if $\alpha > 1$, which we assume holds throughout the paper.

Each existing user v in a P2P system has a number of pointers to other users in the system. As v continues to stay in the system, the identity of its neighbors may change over time as new users arrive and leave the system. There are two types of changes in neighbor tables—voluntary decisions by v to modify its links and abrupt departures of v's existing neighbors. The former type, which we call a *switch*, occurs when v decides to replace an existing neighbor with a new user either to improve its own connectivity to the system or in response to some external event (e.g., new user arrival in DHTs). The latter type of neighbor change, which we call a *recovery*, happens when an existing neighbor dies and v is forced to find a replacement from among the remaining alive nodes. It is during this search process that v is vulnerable to isolation from the graph and potentially unable to route to some of the peers in the system.

Note that switching is a common property of DHTs where arriving users split existing zones of the virtual space and assume the responsibility for all in-degree links assigned to their new zone. In unstructured P2P systems, switching may also occur, for example, when a user decides to "upgrade" its neighbor list to include better-connected or more reliable peers [10]. The analysis below, however, only considers systems that perform recovery, not switching. We assume throughout the rest of the paper that each link, once assigned to a neighbor, is tied to that user for the remainder of the user's lifetime. Unstructured P2P networks (e.g., Gnutella) naturally implement this policy, but DHTs can also be modified to replace neighbors *only* in response to failed links. For example, Randomized Chord [28] does not specify rigid peering rules for outgoing finger links and allows *any* user in a certain range of the DHT space to be v's neighbor. Thus, when new users arrive, v may continue linking to the original neighbor, which avoids switching and keeps v compliant with DHT rules.

We impose additional restrictions on the systems we study to maintain tractability. First, we only consider those networks that have evolved enough to allow asymptotic results from renewal process theory to hold (this usually applies in practice since real P2P systems continuously evolve and seldom or never restart). We also require certain stationarity of lifetime L_i, which means that all users joining the system have the same lifetime distribution $F(x)$. Second, we assume that users always accept incoming connections and do not impose an upper bound on their in-degree. Third, we constrain neighbor selection during recovery to be *uniformly random* within the graph, which can be easily implemented in unstructured P2P systems and certain DHTs that allow freedom in finger pointers (note that successor lists are not considered in this work). Assuming system size n is large, this results in the selection process being independent[3] of the potential neighbor's lifetime L_i or its current age A_i.

B. Modeling Neighbors

Next, we formalize the notion of residual lifetimes and understand how to model neighbor evolution. Define R_i to be the remaining lifetime of node i when it was selected by user v to be its neighbor during join or recovery. As before, let $F(x)$ be the CDF of lifetime L_i. Assuming that n is large and the system has reached stationarity, the CDF of residual lifetimes is given by [35]:

$$F_R(x) = \frac{1}{E[L_i]} \int_0^x (1 - F(z))\, dz. \tag{15}$$

For exponential lifetimes, which we study in the paper for comparison purposes, the residuals are trivially exponential using the memoryless property of $F(x) : F_R(x) = 1 - e^{-\lambda x}$; however, the residuals of Pareto distributions with shape α are *more* heavy-tailed and exhibit shape parameter $\alpha - 1$:

$$F_R(x) = 1 - \left(1 + \frac{x}{\beta}\right)^{1-\alpha}. \tag{16}$$

This means that Pareto-lifetime systems under churn are *more* resilient than the corresponding exponential systems for a given average lifetime since each user in the former case acquires neighbors with *larger* remaining lifetimes than those in the latter case. This can be explained by the fact that $E[R_i] = \beta/(\alpha - 2)$ is larger than $E[L_i] = \beta/(\alpha - 1)$ for all values of α and that residual lifetimes R_i in the Pareto case are stochastically larger than the corresponding lifetimes.

Next, assume that each neighbor $j (1 \le j \le k)$ of node v is either alive at any time t or v is searching for its replacement. Thus, neighbor j can be considered in the *on* state at time t if it is alive or in the *off* state otherwise. This neighbor failure/replacement procedure can be modeled as an alternating renewal process $Y_j(t)$:

$$Y_j(t) = \begin{cases} 1 & \text{neighbor } j \text{ alive at } t \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

Note that the average *on* delay of each process $Y_j(t)$ is $E[R_i]$ and the average *off* delay is $E[S_i]$. Using this notation, the degree of node v at time t is equal to $W(t) = \sum_{j=1}^{k} Y_j(t)$. Denote by T the first time at which a node is *isolated*, i.e., all of its neighbors are simultaneously in the *off* state. Thus, the maximum time a node can spend in the system before it is isolated can be written as the *first hitting time* of process $W(t)$ on level 0:

$$T = \inf\left(t > 0 : W(t) = 0 | W(0) = k\right). \tag{18}$$

[3] For age-dependent selection in unstructured networks, see [42].

Finally, observe that isolation happens only if T is smaller than user lifetime L_v, which means that isolation probability ϕ can be expressed as $\phi = P(T < L_v)$, where both T and L_v are random metrics.

The next two sections develop alternative models for ϕ and examine their accuracy in simulations.

VI. Exact Model of Dynamic Isolation

In this section, we build a rather general model for the probability ϕ that a node v becomes isolated due to all of its neighbors simultaneously reaching the failed state during v's lifetime. While closed-form derivation of ϕ for systems with non-exponential user lifetimes is difficult, certain cases identified below can be solved with arbitrary accuracy by replacing residual lifetimes and search delays with their hyper-exponential equivalents. The rest of this section deals with obtaining an exact model for ϕ using continuous Markov chains, while the next section studies its asymptotic approximation.

A. Hyper-Exponential Approximation

Recall that the hyper-exponential distribution H_m is a mixture of m exponential random variables with probability density function (PDF) in the form of [41]:

$$f_H(x) = \sum_{j=1}^{m} p_j \mu_j e^{-\mu_j x}, \tag{19}$$

where $\mu_j, p_j \geq 0$ for all j and $\sum_{j=1}^{m} p_j = 1$. The above distribution can be interpreted as generating exponential random variables $\exp(\mu_j)$ with probability p_j. It is well-known that any monotonic density function $f(x)$ can be represented with any desired accuracy using (19), i.e., $f_H(x) \to f(x)$ as $m \to \infty$ [3], [15]. In the analysis below, we leverage this property of hyper-exponentials and the fact that Pareto residual lifetime distributions are completely monotonic. While some of the prior literature [15] has used as many as 14 exponentials to approximate Pareto $f(x)$, our analysis suggests that as few as 3 are usually sufficient for achieving very accurate results on ϕ (see examples after the proofs).

Before we proceed with the derivations, it is useful to visualize the meaning of hyper-exponential distributions in our lifetime model. Assume that there are m_r different types of neighbors, where residual lifetimes of peers of type j are exponentially distributed with rate μ_j. When v requires a new neighbor, it selects a node of type j with probability p_j. Similarly, assume that there are m_s types of searches that can be currently in progress. A search of type j is instantiated by v with probability q_j and has duration exponentially distributed with rate λ_j. As long as neighbor residual lifetimes and search delays can be reduced to the hyper-exponential distribution, the resulting process $W(t)$ can be viewed as a homogenous continuous-time Markov chain as we show next.

Lemma 1: Given that the density function of residual lifetimes $f_R(t) = \sum_{j=1}^{m_r} p_j \mu_j e^{-\mu_j t}$ and the density function of search times $f_S(t) = \sum_{j=1}^{m_s} q_j \lambda_j e^{-\lambda_j t}$, $W(t)$ is a homogeneous continuous-time Markov chain.

Proof: Assuming m_r types of neighbors and m_s types of search processes, each state of $W(t)$ for a given user v can be written as:

$$(x_1, \ldots, x_{m_r}, y_1, \ldots, y_{m_s}), \tag{20}$$

where x_i is the number of v's neighbors of type i, y_j is the number of searches in progress of type j, $x_i \geq 0$, $y_j \geq 0$, and $\sum_{i=1}^{m_r} x_i + \sum_{j=1}^{m_s} y_j = k$. Also notice that $W(t)$ can be represented as $\sum_{i=1}^{m_r} x_i$. Since neighbors of type i are $\exp(\mu_i)$ and search processes of type j are $\exp(\lambda_j)$, the sojourn time in state $(x_1, \ldots, x_{m_r}, y_1, \ldots, y_{m_s})$ is exponential with rate:

$$\Lambda = \sum_{i=1}^{m_r} x_i \mu_i + \sum_{j=1}^{m_s} y_j \lambda_j. \tag{21}$$

Observe that when a neighbor dies, a search starts immediately and its properties are independent of those of the existing searches or neighbor lifetimes. Conversely, when a search ends and a new neighbor is found, the characteristics of this neighbor are independent of any previous behavior of $W(t)$. This independence allows us to easily write transition probabilities between adjacent states of $W(t)$. The first type of transition reduces $W(t)$ by 1 in response to the failure of one of v's neighbors, which is equivalent to a jump from state:

$$(x_1, \ldots, x_i, \ldots, x_{m_r}, y_1, \ldots, y_j, \ldots, y_{m_s}) \tag{22}$$

to state:

$$(x_1, \ldots, x_i - 1, \ldots, x_{m_r}, y_1, \ldots, y_j + 1, \ldots, y_{m_s}) \tag{23}$$

for any suitable $x_i \geq 1$. For simplicity of notation, we call the above transition $(x_i, y_j) \to (x_i - 1, y_j + 1)$. The corresponding probability that a neighbor of type i dies and a search of type j starts is $x_i \mu_i q_j / \Lambda$.

The second type of transition increases $W(t)$ by 1 as a result of finding a replacement neighbor, which corresponds to a jump from state:

$$(x_1, \ldots, x_i, \ldots, x_{m_r}, y_1, \ldots, y_j, \ldots, y_{m_s}) \tag{24}$$

to state:

$$(x_1, \ldots, x_i + 1, \ldots, x_{m_r}, y_1, \ldots, y_j - 1, \ldots, y_{m_s}) \tag{25}$$

for any $y_j \geq 1$. The corresponding notation for this transition is $(x_i, y_j) \to (x_i + 1, y_j - 1)$. The related probability that a search process of type j ends and finds a new neighbor of type i before any other event happens is $y_j \lambda_j p_i / \Lambda$.

By recognizing that the jumps behave like a discrete-time Markov chain and the sojourn times at each state are independent exponential random variables, we immediately conclude that $W(t)$ is a homogeneous continuous-time Markov chain with a transition rate matrix $Q = (q_{uu'})$ with:

$$q_{uu'} = \begin{cases} q_j x_i \mu_i & (x_i, y_j) \to (x_i - 1, y_j + 1) \\ p_i y_j \lambda_j & (x_i, y_j) \to (x_i + 1, y_j - 1) \\ -\Lambda & u' = u \\ 0 & \text{otherwise,} \end{cases} \tag{26}$$

where u and u' represent any suitable states in the form of (20) that satisfy transition requirements on the right side of (26). ■

The next step is to specify the initial state distribution of $W(t)$ and derive the PDF of the first-hitting time on state $W(t) = 0$ based on the transition rate matrix Q. While (26) initially appears complicated, placing states $(x_1, \dots, x_{m_r}, y_1, \dots, y_{m_s})$ in an increasing order reveals that Q is not much different from any other rate matrix. For small values of k, the matrix can be easily represented in memory and manipulated in software packages such as Matlab. For example, when $m_r = m_s = 3$, the size of Q is 252×252 for $k = 5$ and 792×792 for $k = 7$.[4]

The initial state distribution $\pi(0)$ is in form of:

$$\pi(0) = \left(\pi_{(x_1,\dots,x_{m_r},y_1,\dots,y_{m_s})}(0)\right), \tag{27}$$

where each entry in the vector represents the probability that the chain starts in state $(x_1, \dots, x_{m_r}, y_1, \dots, y_{m_s})$ for all possible permutations of variables x_i and y_j. Note, however, that the only "valid" starting states are those in which the number of alive neighbors $\sum_{i=1}^{m_r} x_i$ is exactly k and the number of searches in progress $\sum_{j=1}^{m_s} y_j$ is zero.

After rather straightforward manipulations, $\pi(0)$ can be obtained as follows (we omit the proof due to space limitations).

Lemma 2: Valid starting states have initial probabilities:

$$\pi_{(x_1,\dots,x_{m_r},0,\dots,0)}(0) = \prod_{i=1}^{m_r} \binom{k - \sum_{r=1}^{i-1} x_r}{x_i} p_i^{x_i}, \tag{28}$$

and all other states have initial probability 0.

Armed with this result, we next focus our attention on obtaining the distribution of T and deriving ϕ.

B. First Hitting Time

It is convenient to treat $W(t)$ as an absorbing Markov chain in order to derive the PDF of the first-hitting time of $W(t)$ on state 0. To this end, let $E = \{(0, \dots, 0, y_1, \dots, y_{m_s}) | \sum_{j=1}^{m_s} y_j = k\}$ be the set of all absorbing states. Then, for each non-absorbing state $u \notin E$, its transition rate to E is given by:

$$q_{uE} = \sum_{u' \in E} q_{uu'}, \tag{29}$$

where $q_{uu'}$ is the cell of matrix Q corresponding to transitions from state u to u'. We can then write Q in canonical form as:

$$Q = \begin{pmatrix} 0 & 0 \\ \mathbf{r} & Q_0 \end{pmatrix}, \tag{30}$$

where $\mathbf{r} = (q_{uE})^T$ for $u \notin E$ is a column vector representing the transition rates to the absorbing set E and Q_0 is the rate matrix obtained by removing the rows and columns corresponding to states in E from Q.

Generalizing the first hitting time from a starting state $w \notin E$ to any absorbing state in E as:

$$T_{wE} = \inf\{t > 0 : W(t) \in E | W(0) = w\}, \tag{31}$$

its density function can be obtained from the following lemma.

Lemma 3: Given that the chain starts from state $w \notin E$, where E is the set of absorbing states, the PDF $f_{T_{wE}}(t)$ of the first hitting time T_{wE} to any state in E is given by:

$$f_{T_{wE}}(t) = \sum_{u \notin E} p_{wu}(t) q_{uE}, \tag{32}$$

where $p_{wu}(t) = P(W(t) = u | W(0) = w)$ is the probability that the chain is in state u at time t given that it started in state w and q_{uE} is given in (29).

Proof: For Markov chains, it is not difficult to show that T_{wE} has a continuous density function $f_{T_{wE}}(t)$ such that for arbitrarily small dt:

$$P(t < T_{wE} < t + dt) = f_{T_{wE}}(t)dt + o(dt). \tag{33}$$

At the same time, from last-step analysis [22] we have:

$$P(t < T_{wE} < t + dt) = \sum_{u \notin E} p_{wu}(t) q_{uE} dt + o(dt). \tag{34}$$

Combining (33), (34) and letting $dt \to 0$, we easily obtain (32). ■

Computation of $f_{T_{wE}}(t)$ requires transition probabilities $p_{wu}(t)$ for all $u \notin E$, which are rather difficult to obtain in explicit closed-form for non-trivial Markov chains such as ours. Instead, we offer a solution in the next theorem that depends on spectral properties of Q_0 and a matrix representation of $p_{wu}(t)$.

Theorem 3: For hyper-exponential residual lifetimes and hyper-exponential search delays (19), the probability of isolation is:

$$\phi = \pi(0) V B V^{-1} \mathbf{r}, \tag{35}$$

where $\pi(0)$ is the initial state distribution in (28), V is a matrix of eigenvectors of Q_0, $B = \mathrm{diag}(b_j)$ is a diagonal matrix with:

$$b_j = \int_0^\infty (1 - F(t)) e^{\xi_j t} dt, \tag{36}$$

where $F(t)$ is the CDF of user lifetimes and ξ_j is the jth eigenvalue of Q_0.

Proof: Assuming hyper-exponential residuals and search delays, the probability of node isolation ϕ is simply:

$$\phi = P(T < L_v) = \int_0^\infty P(L_v > t) f_T(t) dt. \tag{37}$$

Invoking (32) and expressing it in matrix form, we have:

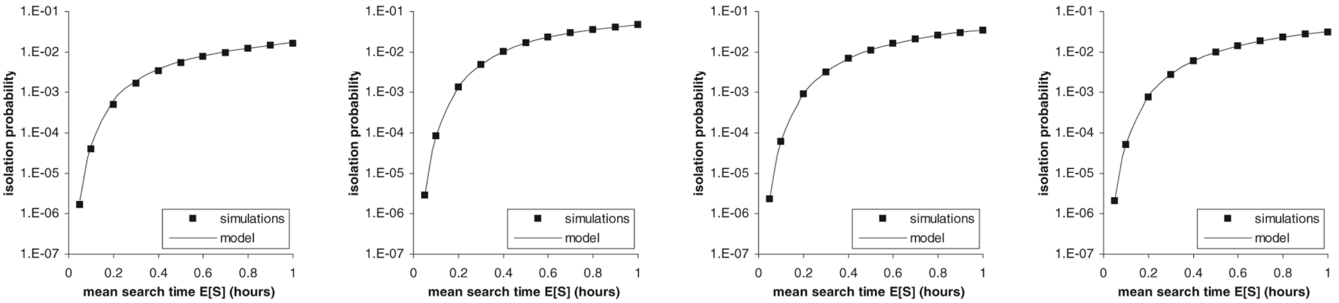
$$(f_{T_{wE}}(t))^T = P_0(t)\mathbf{r}, \quad w \notin E, \tag{38}$$

where $(f_{T_{wE}}(t))^T$ is a column vector and $P_0(t) = (p_{wu}(t))$ for $w \notin E$, $u \notin E$ are transition probability functions corresponding to non-absorbing states. Next representing $P_0(t) = e^{Q_0 t}$ using matrix exponential [35] and $Q_0 = V \Lambda V^{-1}$ using eigen-decomposition [30], we get:

$$P_0(t) = e^{Q_0 t} = V e^{\Lambda t} V^{-1} = V D(t) V^{-1}, \tag{39}$$

where $D(t) = \mathrm{diag}(e^{\xi_j t})$ and $\xi_j \le 0$ is the jth eigenvalue of Q_0. Recalling that there are multiple valid starting states, the

[4] The next section derives an asymptotic approximation to ϕ that does not depend on any matrix algebra; however, it is accurate only for exponential lifetimes.



(a) Wiebull S_i (b) exponential S_i (c) Pareto S_i with $\alpha = 3$ (d) Pareto S_i with $\alpha = 2.5$

Fig. 2. Comparison of model (42) to simulations using exponential lifetimes with $E[L_i] = 0.5$ and $k = 7$.

PDF $f_T(t)$ of the first hitting time T is simply the product of row vector $\pi(0)$ and column vector $(f_{T_{wE}}(t))^T$:

$$f_T(t)=\pi(0)\,(f_{T_{wE}}(t))^T=\pi(0)VD(t)V^{-1}\mathbf{r}, \quad w\notin E, \tag{40}$$

where $\pi(0)$ is given by (28). Substituting the above into (37), we get:

$$\phi = \int_0^\infty P(L_v > t)\pi(0)VD(t)V^{-1}\mathbf{r}dt, \tag{41}$$

which leads to (35) after removing the constants outside the integral and renaming variables. ■

Using (35), rate matrix Q_0, and vector $\mathbf{r}$, solution to node isolation probability ϕ can be easily computed using numerical methods. We next carry out this task using several distributions of user lifetimes and search delays and confirm the accuracy of the model.

C. Verification

We start with the exponential case.

Theorem 4: For exponential lifetimes $L_v \sim \exp(\mu)$ and hyper-exponential search delays, (36) is simply:

$$b_j = 1/(\mu - \xi_j). \tag{42}$$

We next test the accuracy of (42) combined with Theorem 3 in simulations over four distributions of search time for a graph with $k = 7$ and mean lifetime $E[L_i] = 0.5$ hours (additional simulations produce similar results and are omitted for brevity). The first distribution is Weibull with CDF $1 - e^{-(t/a)^c}$ and mean $E[S_i] = a\Gamma(1 + 1/c)$, the second is exponential with rate $1/E[S_i]$, the third is Pareto with $\alpha = 3$, and the fourth is again Pareto with $\alpha = 2.5$. For the Weibull distribution and each fixed $E[S_i]$, shape parameter c is set to 0.5 to produce reasonably heavy-tailed search durations and scale parameter a is kept at $E[S_i]/\Gamma(1+1/c)$. Both Pareto distributions are scaled using $\beta = E[S_i](\alpha - 1)$ to also keep their means equal to $E[S_i]$. For computing the model, each distribution is fitted with a hyper-exponential mixture model (19) using $m_s = 3$.

Fig. 2 illustrates the exemplary accuracy of (42) when compared to simulation results. Notice that the model tracks all four distributions of search delay for over five orders of magnitude and that ϕ becomes less sensitive to the distribution of S_i as $E[S_i] \to 0$. We leverage this observation in the next section and in the meantime discuss an example of how the model can be used to predict the performance of P2P networks. Assuming $k = 9$ neighbors, 30-minute average lifetimes, and 36-second search delays, $\phi = 1.8 \times 10^{-13}$. This demonstrates that P2P networks are extremely resilient against node isolation and can remain connected with a handful of neighbors and search delays on the order of tens of seconds.

Our next theorem derives ϕ for Pareto lifetimes and confirms that Pareto-based systems are more resilient than similar networks based on exponential lifetimes.

Theorem 5: For Pareto lifetimes $L_v \sim 1 - (1 + x/\beta)^{-\alpha}$ and hyper-exponential approximations for residuals and search delays, (36) becomes:

$$b_j = \beta e^{-\xi_j\beta} E_\alpha(-\xi_j\beta), \tag{43}$$

where $E_\alpha(x) = \int_1^\infty e^{-xu}u^{-\alpha}du$ is the generalized exponential integral.

Proof: Using the CDF of Pareto user lifetimes and (36), we get:

$$b_j = \int_0^\infty \left(1 + \frac{t}{\beta}\right)^{-\alpha} e^{\xi_j t}dt. \tag{44}$$

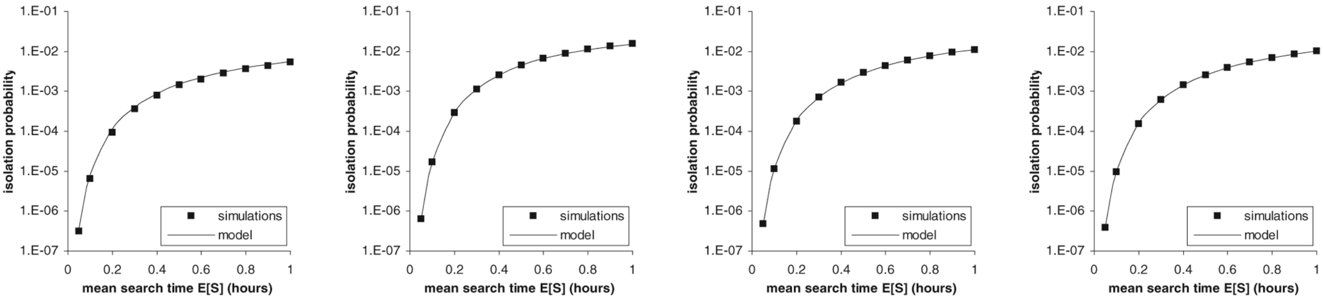
Setting $u = (1 + t/\beta)$, b_j can be reduced to:

$$b_j = \beta e^{-\xi_j\beta} \int_1^\infty u^{-\alpha}e^{\xi_j\beta u}du = \beta e^{-\xi_j\beta} E_\alpha(-\xi_j\beta), \tag{45}$$

which completes the proof. ■

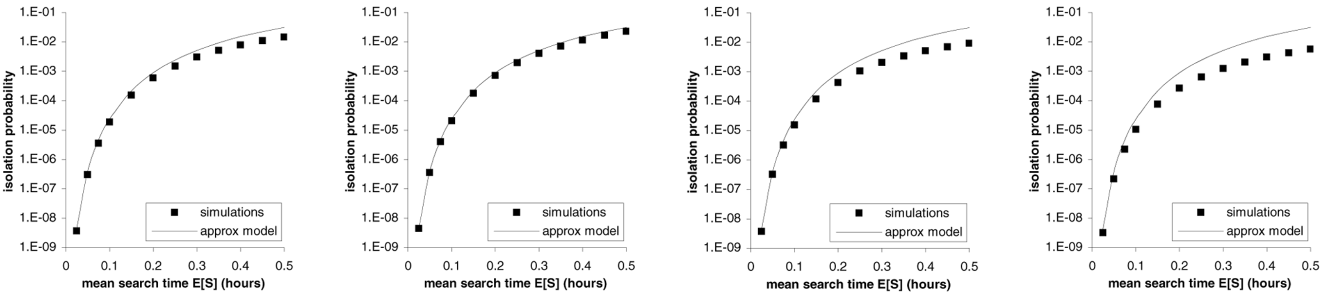
Simulation results for Pareto lifetimes with $k = 7$ and $E[L_i] = 0.5$ for the same four distributions of search delay are illustrated in Fig. 3. As in the case with exponential lifetimes and arbitrary search distributions, (43) combined with Theorem 3 is extremely accurate for all values of $E[S_i]$. Consider the same example of $k = 9$ neighbors, 30-minute average lifetimes, and 36-second search delays. In this case, Pareto $F(x)$ guarantees $\phi = 3.3 \times 10^{-14}$, which is an improvement in resilience by a factor of 5.5 over the exponential case with the same parameters.

While the results in this section are very accurate, the required matrix calculations make it difficult to compute ϕ for a large number of neighbors. For example, using $m_r = m_s = 3$ and $k = 15$, the matrix Q is very large at 15,504 × 15,504. For this reason we next derive a simple closed-form model that approaches the accuracy of (35), (36) for small $E[S_i]$.



(a) Wiebull S_i (b) exponential S_i (c) Pareto S_i with $\alpha = 3$ (d) Pareto S_i with $\alpha = 2.5$

Fig. 3. Comparison of model (43) to simulations using Pareto lifetimes with $E[L_i] = 0.5$ and $k = 7$.



(a) uniform S_i (b) constant S_i (c) exponential S_i (d) Pareto S_i with $\alpha = 3$

Fig. 4. Comparison of model (47) to simulations using exponential lifetimes with $E[L_i] = 0.5, k = 8$.

VII. Asymptotic Model of Dynamic Isolation

Since the previous result (35), (36) requires complex matrix and integral calculations, our next task is to simplify this model in the context of exponential lifetimes and obtain a simple closed-form expression for ϕ that is significantly easier to both understand and compute.

A. Exponential Lifetimes

We start by deriving the stationary distribution of $W(t)$.

Lemma 4 (Leonard [25]): For exponential lifetimes and exponential search delays, the stationary distribution of $W(t)$ is given by:

$$\pi_j = \lim_{t\to\infty} P\left(W(t) = j\right) = \binom{k}{j} \frac{\rho^j}{(1+\rho)^k}, \qquad (46)$$

where $\rho = E[L_i]/E[S_i]$.

We are now ready to present the main result of this section, which follows from Lemma 4 and inequalities for rare events in Markov chains [1], [25].

Theorem 6 (Leonard [25]): For exponential lifetimes and exponential search delays, the probability of isolation is:

$$\phi = \frac{\rho k}{(1+\rho)^k + \rho k - 1} + o(1), \qquad (47)$$

where $\rho = E[L_i]/E[S_i]$ is the ratio of the mean user lifetime to the mean search delay and $o(1)$ is a term that decays to zero as $E[S_i] \to 0$.

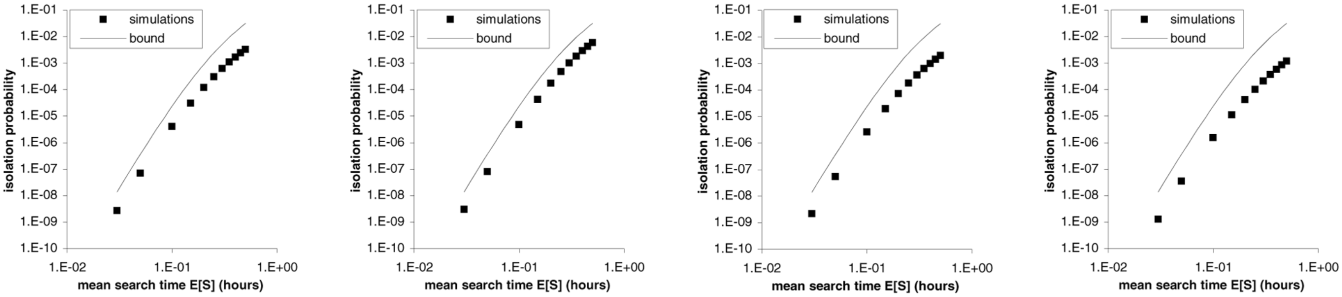
We next test the accuracy of (47) for exponential lifetimes under different distributions of search delay and verify that as $E[S_i] \to 0$ the asymptotic model indeed converges to the exact result (42).

B. Verification

Fig. 4 shows ϕ obtained in simulations using four distributions of search time for a graph with $k = 8$ and mean lifetime $E[L_i] = 0.5$ hours. We again use four different distributions of search delay, but in addition to exponential and Pareto search delays as before, we also include uniform S_i in $[0, 2s]$ and constant S_i equal to s for demonstration purposes that we discuss below. Notice that the asymptotic model is less accurate for exponential search delays, but provides an almost exact match to the constant-delay case (part (b) in the figure). Also observe that as $E[S_i]$ becomes smaller, all four cases indeed converge to (47) and achieve isolation probability $\phi \approx 4.2 \times 10^{-9}$ when the expected search time reduces to 1.5 minutes.

Further note that constant search delays provide the *worst-case* scenario for isolation, while highly variable distributions of S_i are the best. This observation can be explained by the non-negative nature of search times and the fact that for a given $E[S_i]$ higher variance of S_i implies that more probability mass is concentrated at values well below $E[S_i]$. We thus intuitively obtain that random search delays can only *improve* the resilience of the system compared to the worst-case scenario (i.e., constant S_i). This can be observed in Fig. 4 where ϕ in part (b) is the largest among the four cases. Since constant search delays happen to produce an almost ideal match to the approximate model, the result in (47) can be treated as an upper bound on ϕ for all cases with exponential lifetimes.

To finish this subsection, we examine the convergence of approximation (47) to the numerical model (42) in more detail. Table VII shows the values of ϕ produced by both models as $E[S_i]$ becomes very small. Observe in the table that both models



(a) uniform S_i (b) constant S_i (c) exponential S_i (d) Pareto S_i with $\alpha = 3$

Fig. 5. Upper bound (48) and simulations using Pareto lifetimes with $E[L_i] = 0.5$ hours and $k = 8$.

TABLE VII
CONVERGENCE OF (47) TO (42) FOR EXPONENTIAL SEARCH DELAYS AND EXPONENTIAL LIFETIMES WITH $E[L_i] = 0.5$ ($k = 8$)

$E[S_i]$	Model (42)	Model (47)	Ratio
1 hour	3.2480×10^{-2}	1.3971×10^{-1}	4.3017
6 min	1.5379×10^{-5}	2.3814×10^{-5}	1.5485
36 sec	8.2856×10^{-12}	8.7397×10^{-12}	1.0548
3.6 sec	1.0023×10^{-18}	1.0078×10^{-18}	1.0054
360 ms	1.0218×10^{-25}	1.0224×10^{-25}	1.0006

TABLE VIII
MINIMUM DEGREE NEEDED FOR A CERTAIN ϕ IN SYSTEMS WITH PARETO LIFETIMES WITH $\alpha = 3, \beta = 1$ AND $E[L_i] = 0.5$ hours

ϕ	Uniform	Lifetime	Mean Search time $E[S_i]$		
	$p = 1/2$	P2P	6 min	2 min	20 sec
10^{-4}	14	Bound (48)	8	5	4
		Simulations	7	5	4
10^{-6}	20	Bound (48)	10	7	5
		Simulations	10	7	5
10^{-8}	27	Bound (48)	13	9	6
		Simulations	13	8	6

indeed converge and that the relative difference diminishes to zero as $E[S_i]$ becomes small.

C. Bounding Pareto Lifetimes

Without the use of techniques described in Section V, $W(t)$ mixes very slowly under Pareto lifetimes and cannot be modeled as a Markov chain, so the derivation of ϕ for this case is very complicated. Furthermore, the result is expected to be sensitive to the exact value of parameters α and β of the Pareto distribution, which are difficult to measure and may vary from system to system. Thus, we instead utilize the exponential metric (47) as an upper bound on ϕ in systems with sufficiently heavy-tailed lifetime distributions and observe that this approach requires only an estimate of the average user lifetime $E[L_i]$. The result below follows from the fact that heavy-tailed L_i imply stochastically larger residual lifetimes R_i and that (47) provides an upper bound for all search delay distributions.

Corollary 1: For an arbitrary distribution of search delays and any lifetime distribution $F(x)$ with an exponential or heavier tail, which includes Pareto, lognormal, Weibull, and Cauchy distributions, the following upper bound holds:

$$\phi \leq \frac{\rho k}{(1+\rho)^k + \rho k - 1}, \tag{48}$$

where $\rho = E[L_i]/E[S_i]$ is the ratio of the mean user lifetime to the mean search delay.

For example, using 30-minute average lifetimes, 9 neighbors per node, and 1-minute average node replacement delay, the upper bound in (48) equals 1.02×10^{-11}, which allows each user in a 100-billion node network to stay connected to the graph for his/her entire life span with probability $1 - 1/n$. Using the uniform failure model of prior work and $p = 1/2$ [38], each user requires 37 neighbors to achieve the same ϕ *regardless of the actual dynamics of the system.*

To confirm that the upper bound (48) holds in practice, Fig. 5 shows ϕ in simulations with Pareto lifetimes with $E[L_i] = 0.5$ and $k = 8$. Observe in the figure that Pareto systems are in fact more resilient than those with exponential lifetimes. Also notice that constant search delays once again provide the worst-case resilience for a given $E[S_i]$ and that the difference between the Pareto and exponential ϕ is by a constant factor (i.e., the two curves become parallel as $E[S_i] \to 0$).

Even though exponential ϕ is often several times larger than the Pareto ϕ (the exact ratio depends on shape α), it turns out that the difference in node degree needed to achieve a certain level of resilience is usually negligible. To illustrate this result, Table VIII shows the minimum degree k that ensures a given ϕ for different values of search time $E[S_i]$ and Pareto lifetimes with $\alpha = 3, \beta = 1$ ($E[L_i] = 0.5$ hours). The column "uniform $p = 1/2$" contains degree k that can be deduced from the p-percent failure model (for $p = 1/2$) discussed in previous studies [38]. Observe in the table that the exponential case in fact provides a tight upper bound on the actual minimum degree and that the difference between the two cases is at most 1 neighbor.

D. Graph Disconnection

We now apply the newly acquired model for the probability of isolation ϕ to (10) and examine its accuracy in simulations. Re-writing (10), the dynamic resilience of a graph G is lower-bounded by:

$$P(Z > N) \geq \left(1 - \frac{\rho k}{(1+\rho)^k + \rho k - 1}\right)^N, \tag{49}$$

where Z is the number of user joins before the first disconnection of the system. Table IX contains $P(Z > N)$ obtained in simulations of 12-regular, fully populated CAN (i.e., each failed node is replaced after S_i time units by a new

TABLE IX
COMPARISON OF $P(Z > N)$ IN CAN

Fixed search time	Actual $P(Z > N)$	Model (10)	Model (49)	Metric $q(G)$
6 min	.9732	.9728	.9728	1
7.5 min	.8218	.8224	.8215	1
8.5 min	.5669	.5659	.5666	1
9 min	.4065	.4028	.4016	1
9.5 min	.2613	.2645	.2419	1
10.5 min	.0482	.0471	.0424	1

arrival with the same hash index) with exponential lifetimes, $E[L_i] = 0.5$ hours, $n = 4096$, and $N = 10^6$ user joins. The table also includes the value computed by model (10) using empirically measured ϕ along with the newly derived model (49) for comparison purposes. Note that even in the case of relatively large search delays (e.g., $S_i = 10.5$ minutes), the simulations still follow the model quite well and that the graph never partitions without having at least one isolated node (i.e., $q(G) = 1$).

To further illustrate the gravity of (49) when used as a lower bound on the performance of lifetime-based P2P systems, consider the example first mentioned in the introduction. In a k-regular P2P system with $k = 12$ for each neighbor, search delay $E[S_i] = 1$ minute, and average lifetime $E[L_i] = 0.5$ hours, the probability of isolation is $\phi = 4.57 \times 10^{-16}$. When ϕ is applied to (49) in which 35 million users join and leave the system each week, the probability that the network survives for 10,000 years without disconnecting is at least 99.2%. Model (49) further implies that the mean delay between disconnections is lower-bounded by $1/\phi$ user joins, or 1.2 million years. Relatively small systems are also very resilient based on this analysis. A system with $k = 8$, a search delay of 30 seconds, average lifetime $E[L_i] = 0.5$ hours, and 50,000 users join each day will survive for 100 years without disconnection with probability no less than 99.5%. These two examples show that both large and small-scale systems can easily achieve a high level of resilience.

E. Discussion

While the models described in this paper have shown that most current P2P systems are very resilient to node isolation and disconnection under many practical conditions, our results can also be exploited to develop even more resilient systems. However, the obvious solution of increasing k or decreasing $E[S_i]$ will likely cause increased network overhead and reduce scalability. A more cost-effective goal is to ensure that each node has a high probability of obtaining a neighbor with a large residual lifetime during its stay in the system. We propose intentionally monitoring the age of each node and giving more preference during neighbor selection to nodes with larger age, a technique that produces neighbors with larger residual lifetimes. Preliminary simulation results indicate that $E[R_i]$ of chosen neighbors increases by several times over uniformly random selection of neighbors.

This paper is instrumental in understanding the resilience of P2P networks, but there are several avenues that must be explored to fully understand how to prevent user isolation and graph disconnection in all manner of network topologies. One immediate goal is to study the growth of in-degree for nodes in P2P systems, the analysis of which is complicated by the inherent differences between DHTs and unstructured networks. Another goal is to analyze P2P systems with non-uniform neighbor selection techniques (e.g., age-based neighbor selection), an obstacle that is likely to require entirely different methods from those used in this paper.

VIII. CONCLUSION

This paper tackled the problem of P2P graph connectivity under both static and dynamic node-failure by establishing that almost every sufficiently large network remained connected if and only if it had no isolated nodes. We used this powerful result to derive models of graph connectivity for static and dynamic node failure that are much more accurate than previous efforts and are easily calculable. Our results show that most current P2P systems are extremely resilient to disconnection when the average lifetime of a user is at least several times larger than the average node-replacement delay.

Future work includes extending the lifetime model to include the in-degree of each node, analysis of DHTs that replace neighbors when a new user joins, construction of more resilient P2P networks under age-dependent neighbor selection, and measurement of existing P2P networks.

ACKNOWLEDGMENT

The authors would like to thank P. Brighten Godfrey and the anonymous reviewers for helpful feedback on earlier versions of the paper.

REFERENCES

[1] D. J. Aldous and M. Brown, “Inequalities for rare events in time-reversible markov chains II,” *Stochastic Processes Appl.*, vol. 44, pp. 15–25, 1993.

[2] R. Arratia, L. Goldstein, and L. Gordon, “Two moments suffice for poisson approximations: The Chen-Stein method,” *The Annals of Probability*, vol. 17, no. 1, pp. 9–25, Jan. 1989.

[3] S. N. Bernstein, “Sur Les Fonctions Absolument Monotones,” *Acta Math.*, vol. 52, pp. 1–66, 1928.

[4] R. Bhagwan, S. Savage, and G. M. Voelker, “Understanding availability,” in *Proc. IPTPS*, Feb. 2003, pp. 256–267.

[5] F. Boesch, D. Gross, and C. Suffel, “A coherent model for reliability of multiprocessor networks,” *IEEE Trans. Reliab.*, vol. 45, no. 4, pp. 678–684, Dec. 1996.

[6] B. Bollobás, “The evolution of the cube,” *Combinatorial Math.*, pp. 91–97, 1983.

[7] B. Bollobás, *Random Graphs*. Cambridge, U.K.: Cambridge University Press, 2001.

[8] Y. D. Burtin, “Connection probability of a random subgraph of an n-dimensional cube,” *Probl. Inf. Transm.*, vol. 13, no. 2, pp. 147–152, Apr.–June 1977.

[9] F. E. Bustamante and Y. Qiao, “Friendships that last: Peer lifespan and its role in P2P protocols,” in *Proc. Intl. Workshop on Web Content Caching and Distribution*, Sep. 2003.

[10] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, “Making Gnutella-like P2P systems scalable,” in *Proc. ACM SIGCOMM*, Aug. 2003, pp. 407–418.

[11] J. Chen, I. A. Kanj, and G. Wang, “Hypercube network fault tolerance: A probabilistic approach,” in *Proc. ICPP*, Aug. 2002.

[12] B.-G. Chun, B. Zhao, and J. Kubiatowicz, “Impact of neighbor selection on performance and resilience of structured P2P networks,” in *Proc. IPTPS*, Feb. 2005, pp. 264–274.

[13] P. Erdös and A. Rényi, “On the evolution of random graphs,” *Publ. Math. Inst. Hungarian. Acad. Sci.*, vol. 5, pp. 17–61, 1960.

[14] A.-H. Esfahanian, “Generalized measures of fault tolerance with application to n-cube networks,” *IEEE Trans. Comput.*, vol. 38, no. 11, pp. 1586–1591, Nov. 1989.

[15] A. Feldmann, A. C. Gilbert, W. Willinger, and T. G. Kurtz, "The changing nature of network traffic: Scaling phenomena," in *ACM SIGCOMM Comp. Comm. Rev.*, Apr. 1998, vol. 28, pp. 5–29.

[16] H. Frank, "Maximally reliable node weighted graphs," in *Proc. 3rd Ann. Conf. Information Sciences and Systems*, Mar. 1969, pp. 1–6.

[17] A. Ganesh and L. Massoulié, "Failure resilience in balanced overlay networks," in *Proc. Allerton Conf. Commu. Contr. Comput.*, Oct. 2003.

[18] Q.-P. Gu and S. Peng, "Unicast in hypercubes with a large number of faulty nodes," *IEEE Trans. Parallel Distrib. Syst.*, vol. 10, no. 10, pp. 964–975, 1999.

[19] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica, "The impact of DHT routing geometry on resilience and proximity," in *Proc. ACM SIGCOMM*, Aug. 2003, pp. 381–394.

[20] M. F. Kaashoek and D. Karger, "Koorde: A simple degree-optimal distributed hash table," in *Proc. IPTPS*, Feb. 2003, pp. 98–107.

[21] A. K. Kelmans, "Connectivity of probabilistic networks," *Auto. Remote Contr.*, vol. 29, pp. 444–460, 1967.

[22] M. Kijima, *Markov Processes for Stochastic Modeling*. London, U.K.: Chapman & Hall, 1997.

[23] S. Krishnamurthy, S. El-Ansary, E. Aurell, and S. Haridi, "A statistical theory of chord under churn," in *Proc. IPTPS*, Feb. 2005, pp. 93–103.

[24] S. Latifi, "Combinatorial analysis of the fault diameter of the n-cube," *IEEE Trans. Comput.*, vol. 42, no. 1, pp. 27–33, Jan. 1993.

[25] D. Leonard, Z. Yao, X. Wang, and D. Loguinov, "On static and dynamic partitioning behavior of large-scale networks," in *Proc. IEEE ICNP*, Nov. 2005, pp. 345–357.

[26] D. Liben-Nowell, H. Balakrishnan, and D. Karger, "Analysis of the evolution of the peer-to-peer systems," in *Proc. ACM PODC*, Jul. 2002, pp. 233–242.

[27] D. Loguinov, A. Kumar, V. Rai, and S. Ganesh, "Graph-theoretic analysis of structured peer-to-peer systems: Routing distances and fault resilience," in *Proc. ACM SIGCOMM*, Aug. 2003, pp. 395–406.

[28] G. S. Manku, M. Naor, and U. Weider, "Know thy neighbor's neighbor: The power of lookahead in randomized P2P networks," in *Proc. ACM STOC*, Jun. 2004, pp. 54–63.

[29] L. Massoulié, A.-M. Kermarrec, and A. Ganesh, "Network awareness and failure resilience in self-organising overlay networks," in *Proc. IEEE SRDS*, Oct. 2003, pp. 47–55.

[30] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. Philadelphia, PA: Society for Industrial and Applied Math, 2000.

[31] W. Najjar and J.-L. Gaudiot, "Network resilience: A measure of network fault tolerance," *IEEE Trans. Comput.*, vol. 39, no. 2, pp. 174–181, Feb. 1990.

[32] G. Pandurangan, P. Raghavan, and E. Upfal, "Building low-diameter peer-to-peer networks," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 6, pp. 995–1002, Aug. 2003.

[33] M. D. Penrose, "On k-connectivity for a geometric random graph," *Random Structures & Algorithms*, vol. 15, no. 2, pp. 145–164, Sep. 1999.

[34] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proc. ACM SIGCOMM*, Aug. 2001, pp. 161–172.

[35] S. Resnick, *Adventures in Stochastic Processes*. Boston, MA: Birkhäuser, 2002.

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

[37] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "A Measurement Study of peer-to-peer file sharing systems," in *Proc. SPIE/ACM Multimedia Computing and Networking*, Jan. 2002, vol. 4673, pp. 156–170.

[38] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proc. ACM SIGCOMM*, Aug. 2001, pp. 149–160.

[39] K. Sutner, A. Satyanarayana, and C. Suffel, "The complexity of the residual node connectedness reliability problem," *SIAM J. Comput.*, vol. 20, pp. 149–155, 1991.

[40] W. Wang, Y. Zhang, X. Li, and D. Loguinov, "On zone-balancing of peer-to-peer networks: Analysis of random node join," in *ACM SIGMETRICS*, Jun. 2004, pp. 211–222.

[41] R. W. Wolff, *Stochastic Modeling and the Theory of Queues*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[42] Z. Yao, X. Wang, D. Leonard, and D. Loguinov, "On node isolation under churn in unstructured P2P networks with heavy-tailed lifetimes," in *Proc. IEEE INFOCOM*, May 2007.

[43] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. Kubiatowicz, "Tapestry: A resilient global-scale overlay for service deployment," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 1, pp. 41–53, Jan. 2004.

Derek Leonard (S'05) received the B.A. degree (with distinction) in computer science and mathematics from Hendrix College, Conway, AR, in 2002. He is currently pursuing the Ph.D. degree in the Department of Computer Science, Texas A&M University, College Station, TX.

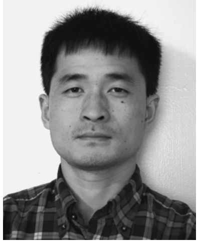
His research interests include peer-to-peer networks, optimization-based graph construction, and large-scale measurement studies of the Internet.

Mr. Leonard has been a student member of the Association for Computing Machinery (ACM) since 2006.

Zhongmei Yao (S'06) received the B.S. degree in engineering from Donghua University (formerly China Textile University), Shanghai, China, in 1997 and the M.S. degree in computer science from Louisiana Tech University, Ruston, LA, in 2004. She is currently pursuing the Ph.D. degree in the Department of Computer Science, Texas A&M University, College Station, TX.

Her research interests include peer-to-peer systems, Markov chains, and stochastic network modeling.

Ms. Yao has been a student member of the Association for Computing Machinery (ACM) since 2006.



Xiaoming Wang (S'04) received the B.S. degree in computer science and the M.S. degree in electronic engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1999 and 2002, respectively. He is currently pursuing the Ph.D. degree at Texas A&M University, College Station, TX.

From 2002 to 2003, he worked for Samsung Advanced Institute of Technology, South Korea. His research interests include peer-to-peer systems, probabilistic analysis of networks, and topology modeling.

Dmitri Loguinov (S'99–M'03–SM'08) received the B.S. degree (with honors) in computer science from Moscow State University, Moscow, Russia, in 1995 and the Ph.D. degree in computer science from the City University of New York, New York, in 2002.

From 2002 to 2007, he was an Assistant Professor of computer science with Texas A&M University, College Station. He is currently a tenured Associate Professor and Director of the Internet Research Lab (IRL) in the same department. His research interests include peer-to-peer networks, video streaming, congestion control, Internet measurement, and modeling.