Price of Structured Routing and Its Mitigation in P2P Systems under Churn

Joseph S. Kong and Vwani P. Roychowdhury
Department of Electrical Engineering, UCLA
Los Angeles, USA
{jskong,vwani}@ee.ucla.edu

Abstract

We study the resilience of structured peer-to-peer (P2P) networks in the presence of churn. Using the lifetime-based failure assumptions, we first show that a realistic churn model has an equivalent uniform failure model in the steady state. We then determine via percolation analysis and simulation the gap between the size of the connected component and the reachable component of a randomly picked node for Symphony and Chord. This gap represents the price of structured routing: the size of the set of nodes that are reachable by any unstructured routing method (e.g. broadcast on the unstructured overlay) from a randomly picked surviving node, but are not reachable using structured routing on the structured overlay. As an illustration, for 24*-minute average node lifetime with* 1*-minute average node-search delay, the gap is around 12 thousand nodes or 1.2% in a Chord network of around 1 million nodes. We finish by discussing potential techniques to mitigate the price of structured routing.*

1 Introduction

In the past few years, we have witnessed an explosion in peer-to-peer (P2P) research, especially in the area of structured P2P systems or distributed hash tables (DHTs). These structured overlays implement an efficient key-based routing interface that supports deterministic routing of messages to the destination typically with $O(\log n)$ latency, and with traffic (per query) that typically scales logarithmically with network size. The unstructured counterparts (i.e., networks where content distribution is not coupled to network connectivity, and any content could reside in any node), such as Gnutella, were widely criticized for their unscalable broadcast-based flooding schemes to search for content. As a result, the newly proposed structured systems were positioned to replace the unstructured P2P overlays as the next-generation of P2P systems.

Clearly, if there is no failure in a P2P system, the efficient structured routing algorithms deliver great performance. However, due to the continuous log-on and log-off of P2P end-users, the system can *never* be assumed to have achieved a perfect structure. As a result, we want to investigate what are the potential losses due to such imperfect structure and how can we characterize such losses?

We attempt to address this question by developing the concept of the *price of structured routing*, which is defined as the *gap* between the size of the connected component and the reachable component of a randomly selected node in a structured P2P system under random failure or churn. In other words, the gap is the size of the set of nodes that are reachable by any unstructured routing method such as simple broadcast from a randomly picked node, but are not reachable using a structured routing method. For hybrid P2P systems such as [3] that use the same structured topology for unstructured traffic, the price of structured routing corresponds to the difference in the number of nodes that can be reached on the unstructured and structured overlay.

The use of the term *price of structured routing* is not a one-sided criticism of structured P2P systems over their unstructured counterparts. Unstructured P2P systems surely have their many drawbacks as documented in the literature. The use of the term, *price of structured routing*, is to better focus this work on the potential drawbacks of structured routing under churn. In addition, the concept of *price of structured routing* is applicable to study any other network for which some form of structured routing is used; basically, any communication network that does not use a flood-based or gossip-based routing scheme uses structured routing. One prominent example would be the internet, since the BGP routing protocol can be classified as a structured routing scheme. Studying the price of structured routing for the internet is an interesting topic for future research.

In a real P2P network, nodes continuously log-on and log-off. When failed neighbors are encountered due to log-off, nodes employ neighbor-recovery algorithms to repair the failed links. However, neighbor recovery is not instantaneous. Using the lifetime model recently developed by Leonard et al. [6], the time required for repairing the

0-7695-2986-0/07 $25.00 © 2007 IEEE
DOI 10.1109/P2P.2007.9

failed links is denoted as the *search-time*, which is assumed to be strictly positive. As a result, for a large network in the steady state, there will always be nodes repairing failed links. Such imperfect structure has implications on the routing performance of the P2P system. Because messages are routed as specified by the underlying structured routing protocol, all pairs belonging to the same connected component need not be reachable under failure (see Fig. 1).

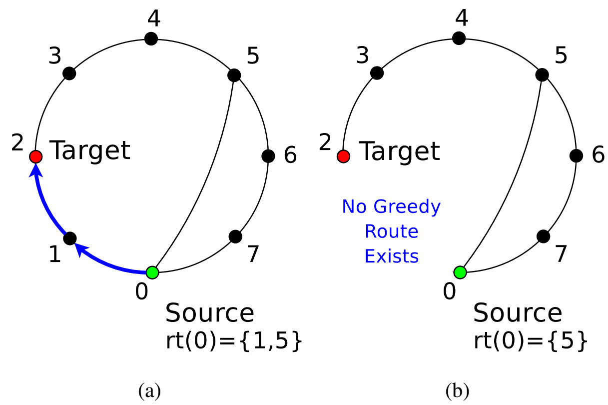


Figure 1. Connected structured networks are not always routable. In this example, we consider greedy, unidirectional routing on a ring with shortcuts. (a) Node 0 wishes to use greedy routing to reach node 2. (b) Node 1 has failed, $rt(0) = \{5\}$. Node 0 is closer to the target than the node in its routing table and thus greedy routing can get the message no closer to node 2.

In an effort to understand the magnitude of the price of structured routing, we performed large-scale simulation studies of the Symphony and Chord networks using realistic churn parameters obtained from empirical studies [2,11]. We simulated a Chord network of around 1 million nodes, where nodes have a Pareto lifetime distribution with a mean of 24 minutes and 1-minute average node-search delay. We found the gap between the size of the connected component and the reachable component of a randomly picked node to be 1.2%. In other words, a randomly selected node in the simulated network cannot reach approximately 12 thousand other nodes, although these nodes belong to the same connected component. For the Symphony network, the gap is found to be 4.0%.

To better understand the price of structured routing under churn, we developed analytical methods to study it. Using the lifetime model [6] and the widely-adopted assumption that the network remains of constant-size, we demonstrate that the lifetime-based churn model has an equivalent q-percent uniform failure model in the steady state (Sec. 2). Since a realistic churn model can be reduced to a uniform node failure model (also known as the site percolation model in physics parlance), we begin by studying the size of the giant connected component under site percolation.

We derive an analytical solution for the site percolation threshold and the size of the giant connected component (GCC) as a function of node survival probability p for Symphony [8] and randomized-Chord [4, 15]. Denoting $x(p)$ as size of the GCC (expressed as a fraction of number of system nodes), it is simple to show that the expected size of the *connected component* (CC) of a randomly chosen surviving node, S_c, is: $S_c = x^2(p)/p$. After analyzing the connected component, we apply the reachable component method [5] to derive the expression for the expected size of the *reachable component* (RC) of a randomly selected live node under random failure.

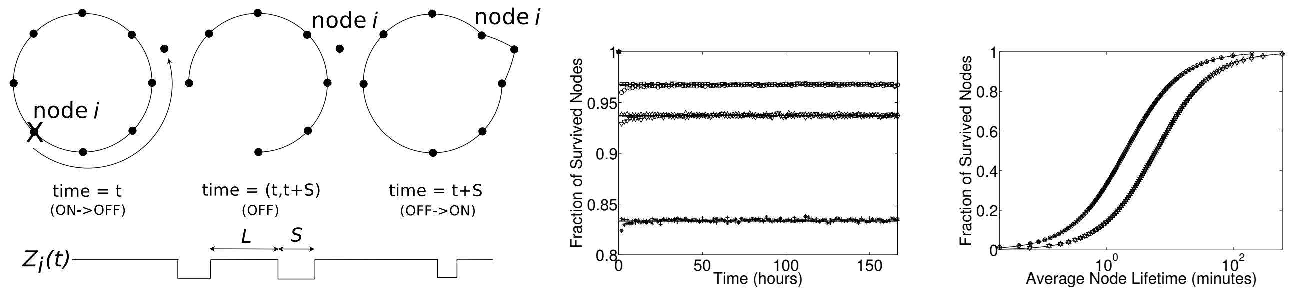
After combining results for the sizes of the connected and the reachable component of a randomly chosen node as a continuous function of p, we point out that the gap between the two curves is the *price of structured routing* for a P2P system, which is the penalty one pays for using structured routing versus an unstructured routing scheme such as a simple broadcast or any one of the more efficient unstructured methods [1, 12]. In addition, we show that the price of structured routing can be dramatically reduced by implementing certain mechanisms. Finally, we want to emphasize that the present paper is the first work that studies *both* the connectivity and routing properties of a faulty P2P network at the same time; previous works either focus on connectivity [6] or routing [7, 13].

2 Link between the Churn Model and the Uniform Failure Model

Leonard et al. [6] recently introduced the lifetime-based node failure model to characterize churn in P2P systems (referred to as the *lifetime model* in this paper). In this model, each joining node stays in the network for the lifetime L, then fails. After a node's neighbor has departed, the node spends the search-time S looking for a replacement neighbor. By the same token, a newly joining node spends S amount of time searching for required connections.

In the lifetime model, in order to prevent network size decreasing to zero, it is assumed that each failed node is to be immediately replaced by a new node (i.e. the failed node *rejoins*) with a randomized nodeID and a random lifetime L [6], where the nodeID determines the node's location in a DHT structure. This model assumption of a fixed-size network is widely used in the literature: the same assumption is used in the simulation studies of DHT systems under churn [7] and in the experimental studies of Bamboo [13]. This assumption of fixed-size network should yield similar results as the more realistic model of having nodes join and depart at the same rate to keep the time average size of the network constant.

Assuming that the network has evolved sufficiently long



(a) (b) $S = 2$ min, $L = \{10, 30, 60\}$ min (c) $S = \{2, 6\}$ min

Figure 2. (a) Illustration of a node failure event: at time t, node i fails and rejoins the network at a new randomized location as a "new" node; for the time period between t and $t + S$, node i's previous neighbors will be looking for replacement connections and node i will be looking for new connections at the rejoining location; at time $t + S$, all affected nodes obtain the required connections. Bottom: the ON/OFF process $Z_i(t)$ for node i. The length of the ON and OFF periods are given by the i.i.d. random variable L and S respectively. (b) Plot shows the time evolution of the fraction of ON nodes. The process quickly converges to the expected fraction of ON nodes as given by *Lemma 1*. (c) Plot shows that the simulation results match the theoretical curve as predicted by *Lemma 1*.

for renewal process theory to hold, the state of node i can be modeled as an ON/OFF process: at time t, the node is considered to be in the ON state if the node is alive with all the required connection as specified by the underlying DHT; on the other hand, the node is in the OFF state if the node has just failed and it is still searching for its required connections at its rejoining location (see Fig. 2(a)). We thus obtain the following simple yet important lemma.

Lemma 1 *Let $Z_i(t)$ denote the ON/OFF process for node i in a structured P2P system. The steady-state probability of finding the process in the ON state is:*

$$p = \lim_{t\to\infty} P(Z_i(t) = 1) = \frac{\bar{L}}{\bar{L} + \bar{S}} \qquad (1)$$

where $\bar{L}$ and $\bar{S}$ are the mean lifetime and search-time, respectively.

This result implies that once the network has evolved long enough to reach stationarity, the probability of finding a random node in the functioning ON state is given by p and the probability of a random node in the failed OFF state is $1 - p$ (i.e. technically alive but the node is still searching for neighbors). Note that the probabilities are independent since L and S are i.i.d. random variables. Furthermore, only the means of L and S should have an impact, since p is insensitive to the exact form of the lifetime and search-time distribution. Thus, given the dynamic lifetime model for churn, we can find an equivalent uniform failure model by invoking *Lemma 1*. This is a significant result since *all previous theoretical results derived for the uniform failure model are now valid for a dynamic churn model*.

We performed a simulation study to further support the validity of *Lemma 1*. The simulation study uses a Symphony network of 100,000 nodes. Each node's lifetime and search-time will be given by the i.i.d. random variables L and S each time the node rejoins the network and fails, respectively. As in [6], we will experiment with two lifetime distributions: exponential $F(x) = 1 - e^{-\lambda x}, x > 0$, and the shifted Pareto $F(x) = 1 - (1 + x/\beta)^{-\alpha}, x > 0, \alpha = 2$; S is modeled to be *constant*. We found that the network quickly converges to the stationary state and only the means of L and S impact the results (see Fig. 2(b) and 2(c)).

3 Price of Structured Routing: Simulations

Since a dynamic lifetime-based churn model can be reduced to a uniform node failure model (see Sec. 2), we begin our work by demonstrating the existence and magnitude of the price of structured routing through large-scale simulations under the uniform failure model.

3.1 Background and Simulation Setup

Symphony [8], a small-world routing network in the 1-dimensional case, has a ring-like address space where each node makes k_S short-range or near-neighbor connections and k_L long-range or shortcut connections according to the harmonic $1/y$ distance distribution (y is the numeric distance on the ring between the end-points of the shortcut). Routing is performed greedily by forwarding a message to the neighboring node closest to the destination. We only investigate the uni-directional routing protocol in this work for simplicity.

The simulation steps for Symphony are as follows: first, we simulate the Symphony graph with $N = 2^{20}$ nodes, with each node making k_S short-range connections and k_L long-range connections. All connections are uni-directional and

Size	$p = 0.8$ (Symphony)		$p = 0.96$ (Symphony)		$p = 0.8$ (Chord)		$p = 0.96$ (Chord)	
(N)	% of N	# of nodes (k)	% of N	# of nodes (k)	% of N	# of nodes (k)	% of N	# of nodes (k)
2^{16}	41.0	26.9	2.5	1.6	6.14	4.0	1.00	0.6
2^{18}	46.0	121	2.6	6.8	6.17	16.1	1.09	2.8
2^{20}	52.9	555	**3.7**	39.2	6.13	64.4	**1.09**	11.5

Table 1. Price of structured routing (Symphony): simulation results. The number of nodes *affected* is expressed in thousands.

$\bar{L}$	Pareto (Symphony)		Exponential (Symphony)		Pareto (Chord)		Exponential (Chord)	
(min.)	% of N	# of nodes (k)	% of N	# of nodes (k)	% of N	# of nodes (k)	% of N	# of nodes (k)
4	50.9	534	53.1	557	5.89	61.8	6.32	66.3
9	19.7	207	20.7	217	3.06	32.1	2.96	31.0
24	**4.0**	41.9	**3.6**	37.8	**1.17**	12.3	**1.17**	12.3

Table 2. Price of structured routing (Symphony): simulation results for the churn model. The search delay is 1 minute. Size of the network is 2^{20} nodes. The symbol $\bar{L}$ denotes the mean node lifetime.

are made in the clockwise direction. Second, for each node failure probability $q = 1 - p$, we delete the failed nodes along with all of their associated edges. Third, we randomly pick a survived node and find the size of the connected component and the reachable component. For obtaining the size of the reachable component, we perform the following: we count the number of nodes in the graph that the selected node can reach by following Symphony's uni-directional greedy routing algorithm. We repeat the third step fixed number of times and compute the sample average for the size of the connected and reachable component.

For ease of exposition, we will use the term "Chord" to denote randomized-Chord [4] in the rest of the paper. In Chord, nodes are placed in numerical order around a ring through successor and predecessor connections. Each node maintains $\log_2(N)$ connections or fingers, with the i^{th} finger at a distance uniformly chosen from an interval [2^{i-1},2^i]. Routing in Chord can be done greedily on the ring topology. That is, a message arriving at a node is routed to the node's neighbor with an address that is closest to the destination. For our simulation, a graph with $N = 2^{20}$ nodes is constructed with each node making 20 finger connections, and greedy routing is used. We follow the same procedure as in Symphony to compute the expected size of a randomly selected node's connected component and reachable component for different node failure probabilities.

3.2 Results for Symphony and Chord

After obtaining the simulation results, we can now examine the size of the gap between the *connected* and *reachable* components as a function of p for different structured P2P networks. We will interpret this gap as the price of structured routing for a DHT network under random failure. For Chord, the price is quite significant even for low rates of random failure (i.e. for $p = 0.96$). See Table 1. Recall that the price of structured routing is often expressed as the fraction of system size. Therefore, even a tiny gap of 1% means that around ten thousand nodes of this system with a million nodes are affected.

The price of structured routing is even more pronounced for the Symphony system with 2 near-neighbors and 2 shortcuts. In addition, note that the price of structured routing expressed as a fraction is strictly increasing as a function of system size for Symphony (see Table 1). This observation is expected since Symphony has constant number of state maintenance. In light of this discovery, a DHT system designer can provide an optional unstructured routing functionality in an implementation to allow nodes in a connected component to communicate with each other in the event that the network is in a badly damaged state. In fact, this is equivalent to building a hybrid P2P system.

3.3 Simulations for the Churn Model

Under the lifetime-based churn model, the simulations are performed as follows: every joining node is assigned a lifetime drawn according to a given distribution (e.g. Pareto or exponential); a node fails at the end of its lifetime, loses all of its connections and rejoins the structured network at a new randomized location; we record the size of the connected component and the reachable component at different time steps, and obtain the difference between the time average size of the CC and RC (i.e. the price of structured routing); we repeat the simulation for the Pareto and exponential lifetime distribution with different mean lifetimes. Symphony and Chord networks of 2^{20} nodes are used.

Simulation results are presented in Table 2. First, the exact form of the lifetime distribution has little impact. Second, recall that by invoking *Lemma 1*, the entries for $\bar{L} = \{4, 24\}$ minutes with $\bar{S} = 1$ minute correspond to $p = \{0.8, 0.96\}$. After examining Tables 1 and 2, we find that the corresponding entries match well with each other (the corresponding values for $p = 0.96$ and $\bar{L} = 24$ minutes are in bold). This observation further supports the link between the lifetime-based churn model and the uniform failure model.

As a practical illustration, for a realistic 24-minute average node lifetime [2,11] with 1-minute average node-search delay in Chord (approximately corresponding to a PING interval of 1 minute), the price of structured routing is 12.3 thousand nodes or 1.17% in a Chord network of 1.05 million nodes (Table 2). For any node characteristics with shorter lifetime or longer node-search delay, the price of structured routing is much more pronounced. Having examined the existence and the magnitude of the price of structured routing through simulations, we will now present the analytical methods to derive matching results.

4 Analysis

4.1 Site Percolation Analysis

Given a graph, site percolation theory deals with the emergence of a giant connected component that spans the graph, when nodes (i.e. sites) of the graph randomly fail with probability $q = 1-p$ [14]. Therefore, the uniform node failure model we assume in this work exactly corresponds to the site percolation problem. Note that the uniform node failure model is further linked to the lifetime-based churn model (Sec. 2).

Our primary figure of interest is $P(n)$, which is the probability that a randomly selected node belongs to a connected component of n nodes. Since $P(n)$ is difficult to evaluate directly, we invoke the generating function method by first defining $G(z)$ as: $G(z) = \sum_{n=0}^{\infty} P(n)z^n$.

The basic approach behind our solution is inspired by [10]. The roadmap to the solution is as follows: first, we find the self-consistent expression for $G(z)$; second, we compute the mean connected component size and the *percolation threshold*, which is the point at which the mean connected component size diverges to infinity; third, above the percolation threshold, $G(z)$ is redefined to denote the generating function for the distribution of the connected components that do not belong to the GCC; thus, the size of the GCC is given by $1 - G(1)$.

For illustration purpose, we will show how the generating function method is applied on the Erdos-Renyi (ER) random graph to derive the size of the GCC above the percolation threshold. Recall that the ER random graph has a Poisson degree distribution with the corresponding generating function given as: $H(z) = e^{\bar{k}(z-1)}$, where $\bar{k}$ is the average degree.

First, we note that a connected component consists of a node with any number of connected components attached to the node by a number of edges described by the degree distribution. Thus, we can write down the self-consistency equation: $G(z) = z\sum p_k[G(z)]^k = zH(G(z))$.

Thus, the recursive equation describing the size of the GCC, x, is given as: $x = 1 - G(1) = 1 - e^{-\bar{k}x}$. The size of the GCC is obtained by finding the solution to the above equation. The derivation for the Symphony graph under random node failures follows similarly as the derivation for ER graph.

4.2 Percolation Analysis of Symphony

Since the Symphony graph is arguably simpler than the Chord graph, we will begin with the derivation of the size of the giant connected component for the Symphony graph. For the Symphony graph under random failure, we will cluster *groups* of locally connected vertices on the ring as *nodes* on an overlay graph; the shortcut long-range connections become the edges of the overlay graph. We then apply the generating function method on the overlay graph.

As an illustration, we consider a Symphony graph where each node has 1 short-range ($k_S = 1$) and 1 long-range connection ($k_L = 1$). Now, every node in the system fails randomly with probability $q = 1 - p$. With approximately $k = Nq$ nodes failing, there will be k cuts separating the ring into approximately k local clusters or "islands", which form the *bottom layer* (see Fig. 3(b)). These local clusters will be joined by the long-range connections to form larger connected components making up the *top layer* (see Fig. 3(c)). Since the long-range connections are made in a random manner, an overlay graph with randomized connections is obtained by modeling each local cluster as a node and the long-range random connections as edges.

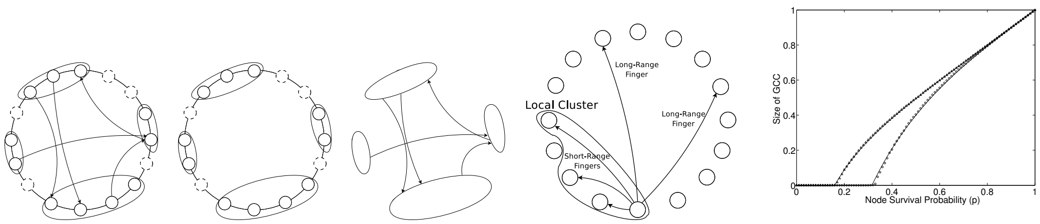
We now consider a general Symphony graph with any number of short-range and long-range connections. We first investigate the bottom ring layer. Let us use $P_0(n)$ to denote the probability that a randomly chosen node belongs to a local cluster of n nodes. We note that $P_0(n)$ is given as follows:

$$P_0(n) = \begin{cases} 1-p, & n = 0 \\ npr^{n-1}(1-r)^2, & n \geq 1 \end{cases} \quad (2)$$

where the variable r, which can be viewed as the probability of extending the local cluster in one direction, is defined as: $r = 1 - (1-p)^{k_S}$. The generating function for Eq. (2) is:

$$G_0(z) = \sum_{n=0}^{\infty} P_0(n)z^n = 1 - p + pz\frac{(1-r)^2}{(1-rz)^2} \quad (3)$$

In addition, we observe that every connected component that consists of a local cluster of n nodes has nk_L outgoing long-range connections from the local cluster and m incoming long-range connections attached to the local cluster;



(a) (b) (c) (d) (e) Symphony

Figure 3. (a) Symphony graph (k_S=1,k_L=1) with node failures: the ring is separated into 4 local clusters. (b) Symphony: the bottom layer of *local clusters* is formed by short-range connections. (c) Symphony: the top layer is an overlay graph obtained by modeling each local cluster as a node and the long-range connections as edges. (d) Long-range vs short-range fingers in Chord: the short-range fingers connect nodes to form *local clusters*, while the long-range fingers connect local clusters to form larger connected components. (e) Size of GCC in Symphony: diamonds (k_S=1,k_L=1), triangles (k_S=2,k_L=2).

these outbound and inbound long-range connections lead the local cluster to $m + nk_L$ other connected components.

Thus, we can write a self-consistent equation as follows:

$$G(z) = \sum_{n=0}^{\infty} P_0(n) z^n \sum_{m=0}^{\infty} P(m|n)[G(z)]^{c(m+nk_L)} \quad (4)$$

where c is the fraction of effective shortcuts and $P(m|n)$ denotes the probability of having m incoming long-range connections to a local cluster of size n.

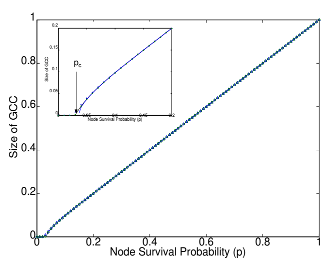


Figure 4. Size of GCC in Chord: simulation results (triangles). The simulation results match perfectly with analytical solutions (solid lines) for both Symphony and Chord ($N = 2^{20}$ nodes).

After performing a series of mathematical computations, we arrive at the self-consistency equation for finding the size of the giant connected component, x:

$$x = 1 - G_0((1-x)^{ck_L} e^{k_L((1-x)^c - 1)}) \quad (5)$$

where $G_0(z)$ is given by Eq. (3). The size of the GCC is obtained by finding the non-negative solution to Eq. (5). The analytical results match exactly with results from large-scale simulations (see Fig. 3(e)).

4.3 Percolation Analysis of Chord

The derivation for Chord follows closely as the derivation for Symphony in the section above, although there are several key differences. In Chord, each node makes $d = \log_2(N)$ connections, with the i^{th} connection made to the node at a distance uniformly chosen from an interval $[2^{i-1}, 2^i)$. We introduce the following concepts: the first k_S finger connections in a node's finger table are considered as "short-range" fingers. The rest of the k_L fingers are "long-range" fingers (see Fig. 3(d)). Note that we have: $k_S + k_L = d$. The cluster formed by only the short-range fingers is defined as the *local cluster*.

After performing similar computations as in the case for Symphony, we found that the size of the giant connected component, x, is obtained by solving the following equation:

$$x = 1 - G_0([(1-x)e^{-x}]^{k_L}) \quad (6)$$

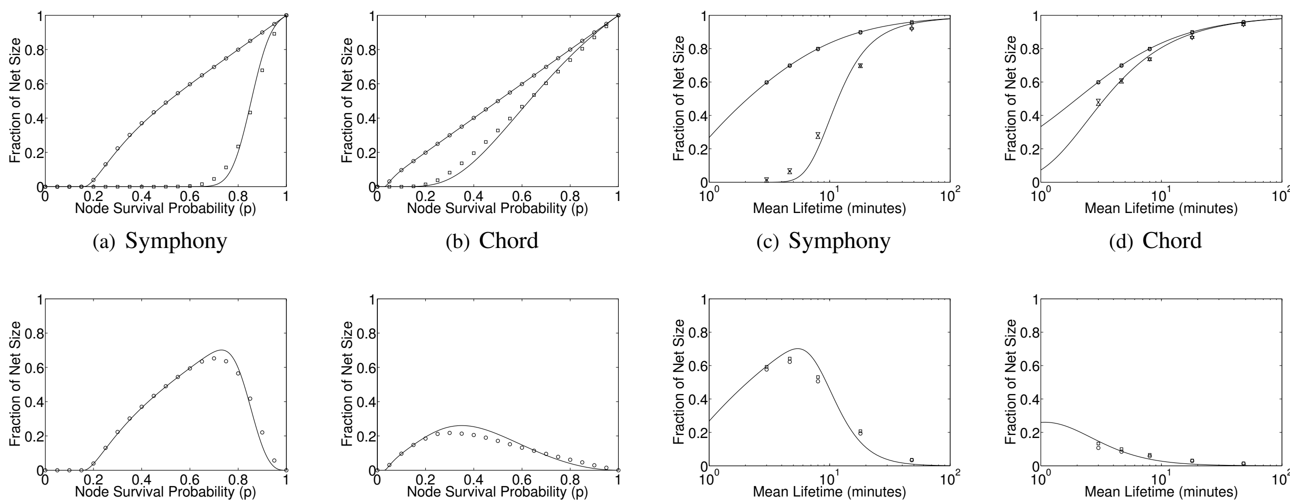
where $G_0(z)$ is given by Eq. (3). As shown in Fig. 4, the analytical results match exactly with the large-scale simulation results for a network of more than 1 million nodes.

4.4 Reachable Component Method

In the previous sections, we derived the size of the giant connected component of Symphony and Chord under random failure. In this section, we will examine the properties of the *reachable component* in a structured P2P system.

Using the *reachable component method* [5], the expected size of the reachable component of a randomly chosen node in a DHT routing systems can be analytically derived. Due to space constraints, we will simply present the results from [5]. We assume that the system has $N = 2^d$ nodes, where d is the node identifier length in bits.

For randomized Chord, the following expression for the hypercube geometry gives a good approximation of the ex-



(a) Symphony (b) Chord (c) Symphony (d) Chord

(e) Symphony (f) Chord (g) Symphony (h) Chord

Figure 5. (a,b) Uniform failure model: size of the connected component (CC) and the reachable component (RC) in Symphony ($k_S = 2, k_L = 2$) and Chord as a function of node survival probability p. The simulation results are shown as follows: size of CC (circles) and size of RC (squares). (c,d) Churn model: size of the CC and the RC in Symphony ($k_S = 2, k_L = 2$) and Chord as a function of average node lifetime (in minutes). The search delay, $\bar{S}$, is 2 minutes. The simulation points are time averages: CC (square,circle) and RC (up-triangle,down-triangle), with the format (exponential,Pareto). The simulation data points match with the analytical predictions (solid lines) closely. (e,f,g,h) Price of structured routing: the bottom figures simply plot the the gap between the CC and RC analytical curves and simulation points for the top figures. The price of structured routing is significant across a wide range. Simulations were performed on systems of $N = 2^{20}$ nodes.

pected size of the reachable component, since Chord routing can be shown as a slight variant of hypercubic routing:

$$E[S] \approx \sum_{h=1}^{d} \binom{d}{h} \prod_{m=1}^{h} (1 - q^m) \tag{7}$$

For Symphony routing, the expression for the expected size of the reachable component is as follows:

$$E[S] \approx \sum_{h=1}^{d} 2^{h-1} \prod_{m=1}^{h} (1 - Q_{sym}) \tag{8}$$

where

$$Q_{sym} \approx q^{1+k_L}\left(\frac{1 - (1 - \frac{k_L}{d} - q^{1+k_L})^{\frac{d}{1-q}+1}}{1 - (1 - \frac{k_L}{d} - q^{1+k_L})}\right)$$

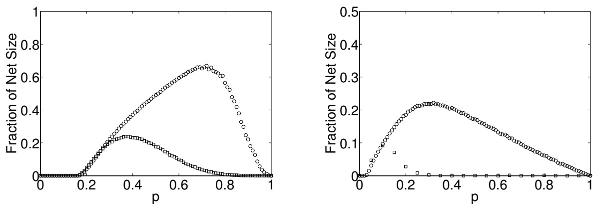
4.5 Analytical Results on the Price

Given the size of the giant connected component $x(p)$, the size of connected component of a randomly chosen node, $S_c(p)$, can be easily obtained by invoking the formula (Sec. 1): $S_c(p) = x^2(p)/p$. Finally, we have obtained the analytical results for both the size of connected component and reachable component of a randomly chosen node for Symphony and Chord under random failure. Thus, for each node survival probability p, the difference between the size of connected and reachable component yields the price of structured routing. Under a realistic churn model with average lifetime and search-time $\bar{L}$ and $\bar{S}$, we can find the price of structured routing analytically by first invoking *Lemma 1* and obtain the equivalent p value.

5 Analytical vs. Simulation Results

We combine the analytical and simulation results on the price of structured routing for both the uniform failure model and the lifetime-based churn model (Fig. 5). Note that the slight discrepancies between the theoretical curve and the simulation results are due to the approximations taken in deriving the size of the reachable component.

The churn simulations are performed as specified in Sec. 3.3. The theoretical curve under the churn model is obtained by first mapping the mean lifetime and search-time values to the node survival probability p by invoking *Lemma 1*, as discussed in Sec. 4.5. The churn simulation results match very closely with the theoretical predictions (see Fig. 5).



(a) Symphony ($N = 2^{20}$) (b) Chord ($N = 2^{20}$)

Figure 6. Mitigation techniques: simulation results. In (a), we compare the price of structured routing for a Symphony network ($k_S = 2, k_L = 2$) that implemented the NoN routing algorithm (squares) and for a network that uses simple greedy routing (circles). In (b), we compare the price of structured routing for a Chord network with 20 successor neighbors (squares) and for a network that has no successor neighbor (circles). The price of structured routing is significantly mitigated in both cases.

6 Mitigation Techniques

In this work, we use the *greedy routing* algorithm for Symphony and Chord to find the size of the reachable component under failure. It is conceivable that we can reduce the price of structured routing by allowing other routing algorithms. For example, one can allow nodes to perform neighbor-of-neighbor greedy routing (i.e. NoN-greedy routing: passing messages in a greedy manner to a neighbor or a neighbor's neighbor) [9]. Another method would be to increase the number of neighbors that each node keeps to provide additional fault tolerance. We implemented the NoN-greedy routing algorithm for Symphony and the mechanism for keeping a list of successor neighbors for Chord in our simulator. We found that the price of structured routing is greatly reduced (see Fig. 6). Nevertheless, these mechanisms would introduce additional costs such as message overheads and maintenance traffic. Another solution would be to provide an optional unstructured routing functionality [1, 12], which is equivalent to building a hybrid P2P system such as [3].

7 Concluding Remarks

In this work, we determined via analysis and simulation the magnitude of the price of structured routing for Symphony and Chord. Our analytical and simulation studies are applicable to other structured P2P systems such as Kademlia, Tapestry and Pastry. In sum, this work contributes to the further understanding of the resilience properties of P2P networks under churn.

8 Acknowledgments

The authors would like to thank Jesse Bridgewater and Linling He for helpful discussions and suggestions. This work was in part supported by the NSF grants ITR:ECF0300635 and BIC:EMT0524843.

References

[1] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman. Search in power-law networks. *Phys. Rev. E*, 64(4):046135, Sep 2001.
[2] F. E. Bustamante and Y. Qiao. Friendships that last: peer lifespan and its role in p2p protocols. In *IEEE WCW*, 2003.
[3] M. Castro, M. Costa, and A. Rowstron. Should we build gnutella on a structured overlay? *SIGCOMM Comput. Commun. Rev.*, 34(1):131–136, 2004.
[4] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica. The impact of DHT routing geometry on resilience and proximity. In *SIGCOMM*, 2003.
[5] J. Kong, J. Bridgewater, and V. Roychowdhury. A general framework for scalability and performance analysis of DHT routing systems. In *Proc. IEEE DSN*, 2006.
[6] D. Leonard, V. Rai, and D. Loguinov. On lifetime-based node failure and stochastic resilience of decentralized peer-to-peer networks. In *Proc. ACM SIGMETRICS*, 2005.
[7] J. Li, J. Stribling, R. Morris, M. F. Kaashoek, and T. M. Gil. A performance vs. cost framework for evaluating DHT design tradeoffs. In *Proc. IEEE INFOCOM*, 2005.
[8] G. S. Manku, M. Bawa, and P. Raghavan. Symphony: Distributed hashing in a small world. *Proc. 4th USENIX SITS*, pages 127–140, 2003.
[9] G. S. Manku, M. Naor, and U. Wieder. Know thy neighbor's neighbor. In *STOC '04*, 2004.
[10] M. E. J. Newman and D. J. Watts. Scaling and percolation in the small-world network model. *Physical Review E*, 60:7332, 2000.
[11] S. Saroiu, K. P. Gummadi, and S. D. Gribble. A measurement study of peer-to-peer file sharing systems. In *MMCN*, 2002.
[12] N. Sarshar, P. O. Boykin, and V. Roychowdhury. Percolation search in power law networks. In *Proc. IEEE P2P*, 2004.
[13] T. R. Sean Rhea, Dennis Geels and J. Kubiatowicz. Handling churn in a DHT. In *USENIX Annual Technical Conference*, 2004.
[14] D. Stauffer. *Introduction To Percolation Theory*. CRC, Philadelphia, PA, 1994. 2nd edition.
[15] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Trans. Netw.*, 11(1):17–32, 2003.