Adapting Peer-to-Peer Topologies to Improve System Performance

Paul Silvey and Laurie Hurwitz
The MITRE Corporation
{psilvey, lhurwitz}@mitre.org

Abstract

Proposals for improving the performance of Peer-to-Peer file sharing systems like Gnutella often simply involve changes to the distributed search protocol. Since the effectiveness of any routing protocol is dependent on the P2P overlay network's interconnection topology, simultaneously controlling the network topology should enable performance enhancements as well. We consider how locally adaptive behaviors can lead to globally robust, scalable, and efficient P2P networks. We adapt topologies using operations of edge thinning, the removal of redundant links based on message passing utilities, and diameter folding, the selective addition of short-cut links between nodes at or near the diameter of the graph. Using network simulations, we establish how these locally selfish behaviors might help explain the ubiquitous natural occurrence of scale-free networks, and demonstrate how P2P networks that adapt their topologies toward more regular degree distributions improve in both performance and robustness.

1. Introduction

As our society and economy grow more network-centric, Peer-to-Peer (P2P) architectures are emerging as a viable technical approach to the construction of massively distributed information processing and file sharing systems. P2P systems are formed from dynamic connections among autonomous computer nodes, producing large and complex application-layer overlay networks. These overlay networks in turn are built upon physical network infrastructures like the Internet. Only recently have large real-world networks been studied carefully, resulting in the recurring discovery of a remarkably common structural property of naturally occurring networks—their inter-node connection distributions decay with a power-law tail. Networks with this property have been called scale-free, because there is no single characteristic scale as measured by node degree (number of links per node). Whereas random networks have degree distributions with a central tendency, scale-free networks do not; most of their nodes have very few connections, but a few are hubs with an extremely large number of connections [2]. There is some evidence that P2P overlay networks, just as the physical networks on which they depend, in general tend to be scale-free in nature [11][17].

The scale-free property appears to be explainable in networks that grow in a bottom-up fashion, where nodes that are joining exhibit a preferential attachment bias in selecting their neighbors. In contrast, networks engineered top-down, like those used by hypercube parallel processors or distributed hash table storage systems, exhibit regularity and determinism in their structure and function. While mathematicians have studied graph-theoretical properties of topological network families such as random, scale-free, and regular, they have typically not considered the P2P complexities of node transience, resource differentiation, and supply and demand distribution fluctuations. As part of our on-going research in distributed resource brokering, we are trying to identify and understand the many interdependent forces that affect the performance and resiliency of complex adaptive networks.

In many ways, the quality of a P2P system depends on the structural and behavioral properties of its network. For example, the distributed search protocols for routing queries through the network depend on the node interconnection topology for their efficiency. Flooding-based search algorithms like the one used by Gnutella [13] are very sensitive to the number of edges in the network graph. If the number of links is too small, all nodes will not be reachable in a reasonable amount of time or some will be excluded because they exceed a distance threshold (implemented using a time-to-live counter sent as part of the query message). Conversely, if there are too many links, numerous identical copies of the query message will arrive at many nodes from different directions, resulting in wasted bandwidth. Therefore, understanding how P2P networks grow and evolve is an important topic of study.

Our research is concerned with understanding how best to use P2P principles to create an efficient, scalable and fault-tolerant network of resource brokers to serve as an infrastructure for global information management applications. Initially, we have been interested in the effects that distributed search protocols have on particular network topologies, but our longer-term interests include many aspects of complexity in P2P networks, including peer-transience effects and general information economics, such as fluctuating information supply and demand, and the effects of market forces on autonomous and locally selfish peers. This paper summarizes the initial results of our experiments with adaptive topologies for P2P systems.

We conducted two separate experiments to evaluate the effects of topology adaptation strategies on the performance of a flooding-based distributed search protocol by implementing an instrumented computer simulation and capturing detailed performance metrics. Our first experiment evolved topologies from initial random graphs while maintaining a fixed number of links, while the second experiment allowed the number of links to increase or decrease, governed by target thresholds for key graph characteristics. The heuristics we used to guide our adaptation strategies are based on local cost/benefit analyses of existing or proposed neighbor links, and therefore the resulting topology changes represent reasonable behaviors for autonomous rational peers to perform themselves. We show these behaviors to be both locally and globally beneficial, but without some additional care, they appear to encourage the emergence of hubs, evolving toward scale-free networks.

This paper is structured as follows. Section 2 discusses the background of our research and other related work. Section 3 starts with our preliminary explorations and then describes in more detail the two sets of simulation experiments we performed to study the effects of heuristic adaptation strategies on the overall P2P network performance. In section 4 we discuss the results of these experiments, and then offer some conclusions and plans for future work in section 5. Finally, we list our references in section 6.

2. Background and Related Research

Since complex systems in general are easily modeled as dynamic networks, complex networks are an obvious topic of study in many academic disciplines, including sociology, economics, biology, and physics. They occur throughout our nation's critical infrastructure as the power, communication, transportation, and financial networks on which our collective prosperity depends. Particularly common in these natural settings are scale-free networks, which have been shown not only to be very robust against random failures, but also to have so-called small-world properties, where any two nodes are likely to be relatively close to each other via some path in the network. However, because their hubs are both few in number and highly connected, scale-free networks are vulnerable to catastrophic failures when a few select nodes are removed. As a result, system sciences find the results of these studies to be of foundational importance even though much of this attention on networks and their complex structure and behavior is relatively new.

Most of the Peer-to-Peer work to date has been in the form of real-world experimentation, based on a few simple ideas of autonomy, equality, and decentralization. For example, large-scale distributed processing systems like SETI@Home, and file-sharing systems like Gnutella and the FastTrack network (protocol for KaZaA nodes), have helped identify some practical problems with a few proposed approaches, but in general there is limited scientific understanding of the issues. Just now are P2P systems beginning to be studied academically, in laboratory models, simulations, algorithm designs, and mathematical theorems. More and more research groups like ours are now actively studying P2P problems without trying to get the public to 'test-drive' their proposed solutions. Lab testing certain key concepts before investing in large-scale implementations allows us to gain insights into why and how well various design strategies work, and to do so in a cost-effective manner.

Some researchers have focused on general principles and architectures for P2P systems, recognizing the complementary role of distributed indexing to distributed searching [8]. Similarly, others have emphasized the use of a P2P network as a distributed environment for pushing event notifications to subscribers [5], which we see as a natural complement to the "pull"-oriented file-sharing search and retrieval paradigms. In general, we view P2P architectures as most appropriate for managing distributed metadata to support resource discovery and access in diverse applications such as file sharing and Grid computing [12]. There is also some interesting related work on Distributed Hash Table (DHT) systems, which involves building distributed storage and lookup mechanisms based on various structured overlay topologies [6][9][15]. These researchers are interested in many of the same problems of network scalability, robustness, and efficiency that we are.

P2P researchers have considered several problems, including ways to make search more efficient through selective routing in scale-free networks [1], and ways to control the topology by developing protocols for joining nodes that actively maintain nearly balanced node degrees [16]. Emphasis on the join protocol is important, since scale-free networks have been shown to emerge when a graph grows by adding nodes, and the new nodes use a preferential attachment bias in selecting their neighbors. Simply increasing the chance of a new connection based on an existing node's age, for example, is sufficient to produce power law degree distributions [2]. In highly transient environments, the join and reconnect protocols could very likely have a dominant influence on topology, over adaptive strategies of the kind we have investigated. However, our initial approach is to understand how mature graphs can improve through adaptation without concentrating on the joining process, since the brokers we envision in our networks will seek to operate continuously like servers while directly interacting with each other as peers.

More generally, graph theorists have studied particular topological network families, and many precise results have been proven for regular structured graphs like k-ary d-cubes (a class of graph to which binary hypercubes belong), many forms of random graphs, and graphs with particular degree distributions including scale-free networks. For example, while we have long known that an N-node hypercube's diameter (the maximum shortest path between any pair of nodes) is log N, it has only recently been shown that the diameter of a scale-free network is even smaller [3][7].

Hypercube topologies are particularly interesting because they have been extensively studied in the context of massively parallel computer architectures, and they have many desirable peer-like properties [18]. We use them as baseline topologies in our simulation studies because they are regular graphs (i.e. each node has the same degree), are nicely bounded in diameter, and are completely symmetric (i.e. each node has exactly one node opposite it at the diameter). However, they are not realistic as a topology in environments with arbitrarily sized graphs and high node transience, since the number of nodes and their configuration are both highly constrained. As a result, we have additionally considered nearly regular graphs with randomized connections, a topological family known as random regular, often chosen to have link densities less than or equal to equivalently sized hypercubes.

While we have not formally treated it as such here, the problem we are addressing by adapting P2P topologies might be cast as a traditional kind of graph optimization problem, where we seek to minimize time and space costs, while imposing some constraints on the link density and distribution to ensure some level of robustness against malicious attack. Minimizing time requires small graph diameters, which is most easily achieved by adding links. On the other hand, minimizing space used corresponds to reducing the number of broadcast messages sent, which generally requires removing links. At the same time, maintaining some level of robustness requires keeping link densities above spanning tree levels, and balancing degrees so as to keep nodes indistinguishable from each other from the perspective of a malicious adversary. By considering this as a design problem with inherent tradeoffs and uncertainties, our problem also appears to be similar to those embodied in a broad class of complex systems recently characterized by the acronym HOT, which alternately stands for 'highly optimized tolerance' [4] or 'heuristically optimized trade-offs' [10]. The HOT mechanisms lead to systems that are efficient, yet fragile against certain unanticipated events. Furthermore, since they have been shown to produce various kinds of power-law phenomena, the emergence of scale-free network properties that we encountered in our simulations might be explainable in terms of these recent theories.

3. Topology Evolution Experiments

In this section, we chronicle the development and use of our adaptive topology heuristics, motivating and describing our specific algorithms and experiments. Utility-based edge thinning emerged as a means to save bandwidth during distributed query routing, and diameter folding in response to our desire to reach all nodes in a timely manner. In the process, we came to understand a few things about the inherent time-space trade-offs and their unexpected interactions with characteristics of network robustness.

Our simulated P2P networks were modeled as undirected connected graphs, with each node and link treated identically to the others in our distributed query evaluations. We first identified our overall goals for the network: based on a Gnutella-like flooding search protocol, we wanted to reduce bandwidth usage while maintaining a fast response time and a fault tolerant topology. To assess these criteria we examined the node degrees, the search cost to the network, minimum path lengths for all node pairs, and the redundancy of each link, by collecting simple statistics for each metric during simulation. We examined node degrees to support the analysis of fault tolerance, since any node with too few connections could easily become disconnected from the network, while having too many connections could lead its own failure to partition the network into disconnected subgraphs. We considered search cost, defined as the number of messages passed through the network per query, as a measure of bandwidth usage. The minimum path lengths gave us a gauge on the average search time as well as the maximum search time for a query. Finally, monitoring the redundancy of a link allowed us to know which links we might rearrange with the least disruption to the overall network.

In our preliminary explorations, we compared degree, search cost, and path length statistics across graphs of various sizes and starting topologies. After ensuring that a given initial topology was connected, we simulated a search of the whole graph beginning at each node and calculated our performance statistics. We modeled the search process as a simple parallel breadth-first graph traversal, with each message taking a unit amount of time to propagate to node neighbors not known to have seen it yet (i.e. all neighbors from which it did not receive the message). In each case we began with a binary hypercube as a baseline and generated other graph topologies to have a comparable number of links. All of the search costs came out similar and, at least for the topological families and link densities we studied, the search costs were approximately the same as the total number of links. We also noticed that the diameters of the graphs were similar. As a result of these explorations, we began developing a topology evolution algorithm to reduce the total number of links and therefore reduce the search cost. This resulted in our first adaptive topology heuristic, edge-thinning based on message passing utility scores.

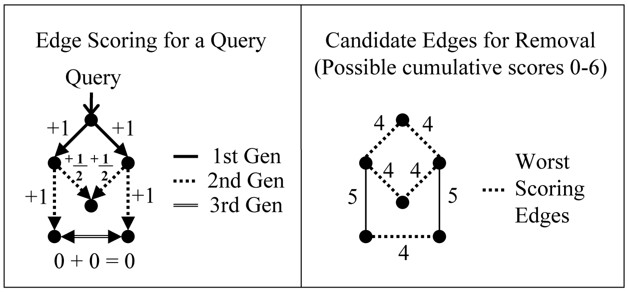

Figure 1 – Edge Thinning

Initially, we simply wanted to reduce search cost by removing the most redundant links without disconnecting the graph. In order to calculate redundancy, we needed a way to assign a 'score' to each link within the network to determine how useful it was across all searches. To implement this, we conducted a search using each node as the starting point, and accumulated utility scores to produce a total for each edge in the graph. The per-search utility scores are computed as follows. If a node only receives a given message from one neighbor, that link gets 100% of the credit, or a utility score of one. If it receives a new message simultaneously from exactly two neighbors, each of those links is given half the unit credit. In general, a node receiving a message for the first time assigns each link a fraction equal to 1/M where M is the total number of message instances received by the given node at that time (see Figure 1). By the rules of the search protocol, a node will ignore any redundant messages received at a later time, but the messages still consume bandwidth. Consequently, our scoring method gives these late message links a score of zero. Since each edge can get a maximum score of 1 per search and there are N such searches, the maximum utility possible for any edge is N. While developing this algorithm we discovered that, given our simplified model of the search process, the only places where late messages can be generated in the graph are within odd-length cycles. Because it is either receiving a message from a neighbor(s) or passing a message to its remaining neighbor(s), a node will only pass to and receive from the same neighbor in an odd cycle. These situations occur when a message originating from one node on the cycle follows different paths to arrive simultaneously at immediate neighbors on the opposite side. The message is then swapped on the next generation as illustrated by generation 3 of Figure 1. Even-length cycles also result in simultaneous redundant message delivery, but no late messages are passed as seen in generation 2 of Figure 1. Since some bandwidth is used without any benefit to the search in these odd cycles, we expected that our scoring strategy would find them. Based on preliminary analysis of our evolved topologies, our edge-thinning algorithm in fact does effectively pinpoint the odd cycles and disconnects links within them.

Our basic thinning algorithm first calculates the search statistics for a given topology and then randomly removes one of the worst scoring links. This calculation-removal cycle can be repeated until the graph becomes disconnected, at which point the last edge removed can be re-inserted. We found that this edge-thinning method reduced each of the test cases of size N to a spanning tree with exactly N-1 links. While this removed all redundancy and created a network with minimal bandwidth usage for a search, we were not aiming to evolve to such a brittle structure. So we either need to stop the thinning process at some earlier point, or replace some removed links by adding new ones somewhere else in the graph. Having created a way to remove the most redundant links, we began considering how and when we might add useful links.

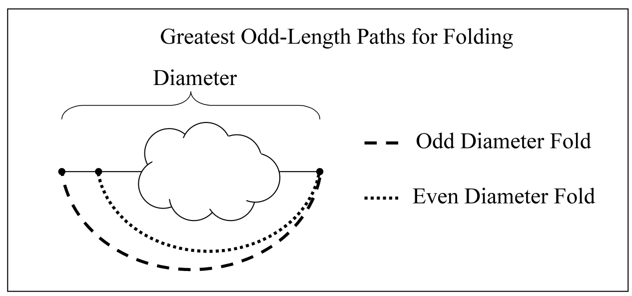

Figure 2 – Diameter Folding

Given our breadth-first search protocol, the worst-case time for a search to reach all other nodes is the same as the graph diameter. One way to reduce the search time is therefore to add shortcut links between all node-pairs whose minimum path length is the diameter, a technique that has been considered at least for hypercube topologies [18]. It is essentially an idealized version of the common-sense "cut out the middleman" heuristic. We call this technique diameter folding, as it folds the graph back in on itself the way one might fold back bread dough after rolling it out. At first we performed one large-scale folding operation, by creating links between all pairs of nodes at the diameter, and then performed thinning from the folded topology. The resulting statistics were similar to what we could already produce with just thinning, so we decided to create a small-scale folding operation that randomly connected just one of the node pairs at the diameter. We tried this operation by alternating it with the thinning operation for a few thousand iterations and discovered that when the graph had an even-length diameter, an edge would often be added that was immediately removed by the next thinning step because the folding step was creating an odd cycle within the graph. Therefore, for the efficiency of these simulations, our folding operation selects a pair of nodes at the diameter and connects them directly if the diameter is odd, or connects one of them to a neighbor along the path if the diameter is even (see Figure 2). This ensures that the new cycle that is being created is always of even length.

Armed with the complementary adaptive strategies just described, we set out to perform more thorough experiments. Our first experiment started with random graphs of various sizes, removing and adding a single link at each step using the thinning and folding operations. In Experiment 1, the overall link density stayed constant, but our hypothesis was that the resulting topologies would still show better performance. Experiment 2 considered the sensitivity to initial topological conditions, and compared the evolution of eight different 128-node graphs with various starting topologies. This second experiment also allowed the link density to evolve toward a target range. At each step in the Experiment 2 simulation, we used thresholds for diameter and mean degree as conditions for applying the folding and thinning operations respectively. The hypothesis here was that the graph could be improved no matter what the initial topology was.

In each experiment, we were effectively modeling two different selfish behaviors that the nodes could perform locally and in parallel, even though in our simulations the adaptation operations were actually applied in a globally sequential manner. The node-centric behaviors are as follows. If a node does not receive timely information across a link it will eventually drop that link, and any node that is extremely costly to reach indirectly will eventually become a candidate for a new direct link. We wanted to see whether these local heuristics would improve overall network performance, and if so by how much. Each of our hypotheses turned out to be verified partially, but scale-free network properties provided us a few surprises along the way. The details of these experiments are presented below.

In Experiment 1, we ran the constant link density tests for random graphs of node sizes 32, 64, 128, 256, and 512 that were generated to emulate the average degrees of binary hypercubes of the same size. Each simulation run proceeded for 5000 steps, alternating between a thinning operation and a folding operation. While we expected to remove odd cycles that were present in the initial graphs and rearrange links to create more tightly coupled topologies, we did not expect to create or reinforce hubs. However, when we plotted the data, it was clear that the maximum degree of the topologies was greatly increasing with evolution.

Though surprising at first, when we reconsidered the algorithm it became clear that while we penalized redundant links, we did nothing specifically to prevent the creation of hubs. In fact, as hubs grow, the heuristics indirectly encourage them, because more than likely the hubs offer the most efficient routes between nodes. In the end we not only remove the small outlying cycles because they are less useful to the overall network but also reinforce the hubs because they are on the shortest paths between many pairs of nodes. To the extent that a node can handle the load, being a hub is helpful to the whole network, and it simultaneously tends to place the node in an advantageous central position.

Table 1 – Experiment 1, Degree Statistics

Graph Size		Target Mean Degree	Actual Mean Degree	Initial Degree Statistics			Degree statistics after 5000 steps of alternating thin and fold					
							Without Balancing Bias			With Balancing Bias		
Nodes	Links			Min	Max	Std	Min	Max	Std	Min	Max	Std
32	66	5	4.1	1	8	2.0	2	17	4.4	2	8	1.4
64	178	6	5.6	1	10	2.1	3	33	7.6	2	9	1.7
128	455	7	7.1	2	15	2.8	3	54	9.7	4	15	2.0
256	1020	8	8.0	1	17	2.9	4	59	10.5	4	17	2.7
512	2371	9	9.3	1	20	3.1	5	37	5.6	6	21	2.3

NOTE: Random graphs generated by probabilistically deciding whether to include each possible link, with the probability determined by target mean degree. Here, the target degrees mimic a base two hypercube of the same size.

However, since one of our original goals in evolving the topologies was to reduce susceptibility to catastrophic failure, we realized that we now explicitly needed to encourage the nodes to balance their degrees in order to avoid creating hubs. We accomplished this by introducing a balancing bias in the final link selection, as follows. In thinning we favored the removal of poor scoring links between nodes with high degrees, and in folding we favored the addition of links between distant nodes with low degrees. The edge nomination process therefore became a two stage filtering operation. The first stage of candidate selection did not change; it still found the node pairs with the lowest edge scores for thinning and the greatest odd minimum path lengths for folding. A second stage of filtering was then added to narrow the set further by considering the effect each proposed change would have on the local link density. Specifically, we selected node pairs from the first stage filter with the highest combined degree for thinning and the lowest combined degree for folding. Any remaining ties after applying the second criteria were then broken randomly. As hoped, this balancing bias algorithm appears to prevent the creation of hubs, as can be seen by comparing the maximum degree with and without balancing to the initial degree statistics in Table 1. As a result, we used it in our subsequent variable link density experiment.

The second experiment considered eight different starting topologies, each with 128 nodes. We wanted to test the modified adaptive strategies that use the balancing bias to see if we could effectively evolve any graph toward some target characteristics. The number of distinct topologies for any graph with more than a just few nodes is astronomical in size. Since we could only attempt an extremely small sample of possible starting topologies, we applied a number of guiding heuristics in our selection. First we used topological families that have previously been studied, including random, random regular, hypercube, and scale-free. We also added two extreme link-density topology cases, a barely connected circle graph and a fully connected graph. Our target link densities and diameters were similarly motivated by heuristics. Here again the hypercube served as a guidepost, offering a desirable small world diameter of log N with a regular balanced degree. The rather high link density of a hypercube, however, appears overly conservative for a real P2P network, where link losses would not be expected to occur so frequently that nodes become completely disconnected before they can find replacement links. Since link density affects the search cost so directly, we chose to favor bandwidth savings over robustness, by aiming for relatively sparse graph topologies.

We set a threshold for maximum average degree and applied the thinning operation only on iterations of the simulation when the link density of the graph was above this threshold. Random graphs with low link densities are often disconnected at generation, and our hope was that a graph with a healthy number of initial links (easy to generate connected) could be trimmed down and made leaner. We chose a mean degree threshold of 3 for our target, striking a balance between a 128-node hypercube's dense regular degree of 7 and an overly thin spanning tree with an average of just under 2 links per node. We also wanted to temper the use of the folding operation, to apply it only when the diameter of the graph had grown unreasonably large. A shortcut link at the diameter always creates a cycle, reducing the maximum distance between any two nodes on that path to at most half the diameter. The diameter of a binary hypercube is $\log_2 N$, so if we folded whenever the diameter grew to more than twice that number we could avoid folding until the graph had stretched out significantly, and still bring it back to a diameter that was basically no worse than a hypercube. Using this reasoning, we chose a diameter threshold to be $2 \log_2 N$, which is 14 for 128 node graphs. A fold operation was performed on any step in the simulation when the diameter was above this threshold.

The initial topologies we evaluated included two extreme topologies, one minimally connected and one maximally connected. The minimal topology chosen was a circle, having one link more than a spanning tree and a large diameter of 64. The maximal topology was a fully connected graph, with each node directly connected to the other 127 nodes. A full network of this size has maximum redundancy in links and a diameter of 1. We expected the circle network mostly to exercise folding as it tried to shorten the diameter, and the full network to do likewise with thinning as it tried to reduce the mean degree. We also included the baseline hypercube topology, a random graph with mean degree of 7, two random regular graphs with different mean degrees, and two scale-free graphs constructed using the Barabási and Albert growth algorithm with different join parameters [2]. The random regular graphs had degrees of 3 and 7, the first being close to our target and the second being close to the hypercube baseline. The scale-free graphs used join rules of 3 and 6, meaning that new nodes added during topology generation randomly attached to 3 or 6 existing nodes respectively.

Table 2 – Experiment 2, Adaptation Statistics
(Initial statistics followed by shaded resulting statistics)

Initial Graph Topology	Degree			Path			Search		
	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
Circle Net	2	2.0	2	1	32.3	64	128	128.0	128
	2	3.0	4	1	4.7	8	193	193.0	193
Binary Hypercube	7	7.0	7	1	3.5	7	448	448.0	448
	2	3.1	5	1	4.4	7	201	201.0	201
Random Regular: degree 3	3	3.0	3	1	5.1	9	219	226.9	235
	1	3.0	5	1	4.6	7	213	222.6	233
Random Regular: degree 7	7	7.0	7	1	2.7	4	594	634.5	666
	1	3.0	7	1	4.2	7	210	221.9	232
Random Net	1	7.1	13	1	2.7	5	618	645.6	673
	1	3.0	13	1	3.8	7	218	225.9	233
Scale-Free Net (3 link join rule)	3	5.9	17	1	2.9	5	498	514.7	542
	1	3.0	15	1	3.6	6	214	224.6	232
Scale-Free Net (6 link join rule)	6	11.7	27	1	2.3	4	1066	1109.7	1177
	1	3.0	22	1	3.3	6	206	223.6	240
Full Net	127	127.0	127	1	1.0	1	16129	16129.0	16129
	1	3.0	10	1	4.2	7	192	192.0	192

Most of the topologies reached equilibrium after only a few thousand steps of simulated adaptation, the exception being the full network. Since it began with 8128 links, it required nearly that many steps of thinning to reach the target link density. Table 2 shows the initial and final statistics of the evolved topologies considered in Experiment 2. As seen by comparing the shaded rows in Table 2, all the topologies were able to evolve to roughly equivalent and improved levels of performance. However, pre-existing hubs that were rare in their initial graphs apparently did not nominate many links for thinning, so scale-free networks present a more challenging starting topology for our evolution strategies. We will discuss the implications of this and review the overall performance statistics of both experiments in the next section.

4. Analysis and Observations

In Experiment 1 we found that with or without a balancing bias, the act of alternately thinning and folding can rearrange the network to reduce overall search cost slightly in terms of bandwidth and time. In Experiment 2, most of the graphs were significantly reduced in their number of links, so their bandwidth performance improvements relative to the starting topologies were much greater than in Experiment 1. In both cases we note that our folding strategy had the benefit of making the network more tolerant to failure by increasing the degree of several nodes that were only sparsely connected originally (e.g. see the evolution of minimum degree values in Table 1).

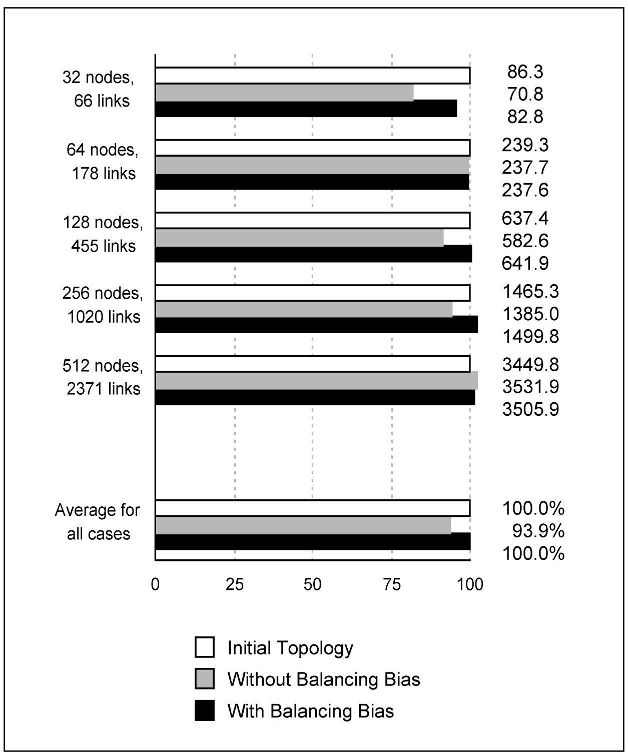

Figure 3 – Experiment 1, Bandwidth Usage
(Average Messages Passed per Query)

Let's take a closer look at what happened in Experiment 1. In Figure 3, we see that even though the overall link density stayed the same, in most of the graphs that evolved without the balancing bias, adapting the network topology did slightly reduce the bandwidth used in query routing. This makes sense because hubs serve as efficient message reflectors, and these graphs were the ones that appeared to be evolving toward scale-free topologies. We will present further supporting evidence of this in a moment. It is more apparent that the hubs are responsible for the savings when we observe that introducing the balancing bias resulted in evolved topologies whose overall bandwidth usage remained essentially the same.

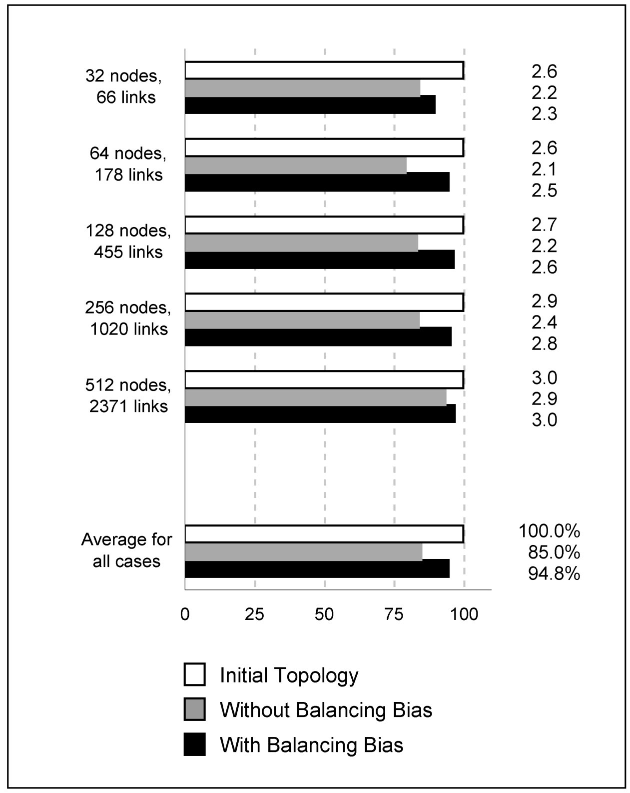

Figure 4 – Experiment 1, Search Time
(Average Minimum Path Length per Query)

Figure 4 shows that the adapted graphs in the first experiment did in fact consistently improve in one important performance metric, the time it takes to propagate a query through the network. Across all cases, the average distance between any two nodes was less for the adapted topologies than in the original random graphs. While this was true for the strategies that encouraged balancing, the mean path lengths were smaller still for the topologies where a few large hubs began to emerge. For example, in the 128-node network the non-balancing evolved topology had at least one hub with a degree of 54. The resulting average minimum path in this graph was over 18% shorter than the initial state and the diameter shrank from 5 to 3. Again, these effects are consistent with the properties of a scale-free network, whose natural diameter has been shown to be smaller than that of a random graph.

Figure 5 summarizes the overall effects of adaptation on degree that we saw in Experiment 1, and the significant differences that the balancing bias produces. To construct this composite figure we normalized the data by setting both the target degree and graph size of the five different sized graphs to the value 1 and then averaged the distributions together. The central tendency and slight skew in the initial degree distribution (column 1) matches the expected Poisson distribution well known for random graphs. The distribution for the graphs evolved with a balancing bias (column 3) shows a tighter central tendency, but is more skewed than the original because existing hubs tend to persist. The oddity is the long-tailed distribution for the networks evolved using the initial random tie-breaking strategy for thinning and folding candidates (column 2). We see that a few hubs with up to almost eight times the average degree of the network have been created but that most nodes have a small degree. This is similar to a power-law distribution though its minimum degree is higher than would be predicted by a pure power-law function. In both evolved cases, the minimum degree increased from its initial value, suggesting that single connection outliers were selected for folding operations.

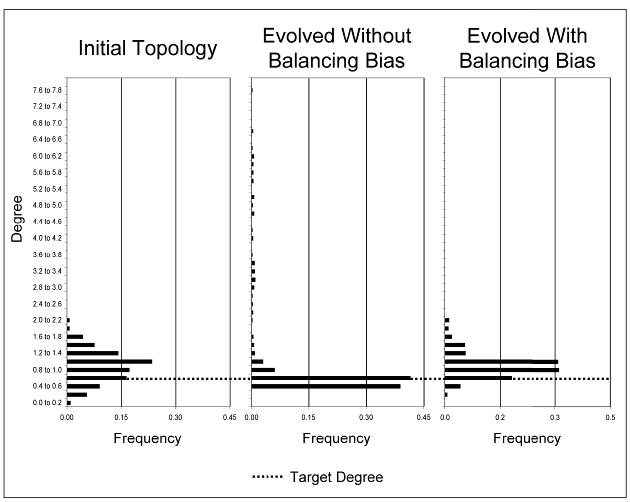

Figure 5 – Experiment 1, Avg. Degree Distributions (Normalized Frequency vs. Normalized Degree)

Overall in Experiment 1, we saw marginal improvements in search cost mostly due to savings in time, and an increase in the robustness of the graph through the effects of diameter folding and the use of the balancing bias during adaptation. The fact that scale-free networks could so easily emerge as an accidental result of our attempts to improve topologies was unexpected. Nevertheless, an intuitive explanation can be formed from the following simple observation. In order for hubs to emerge, links to existing hubs had to score well to avoid shrinking during thinning, and hubs had to be at or near the graph diameter to continue growing during folding. Short paths to many nodes pass through hubs, so their relatively good link scores make sense. The size of the graph does seem to have an effect on the hub growth rate, so being near the diameter may occur less frequently in larger graphs. We are currently developing some graph visualizations that should help us understand better what is going on in the evolution process.

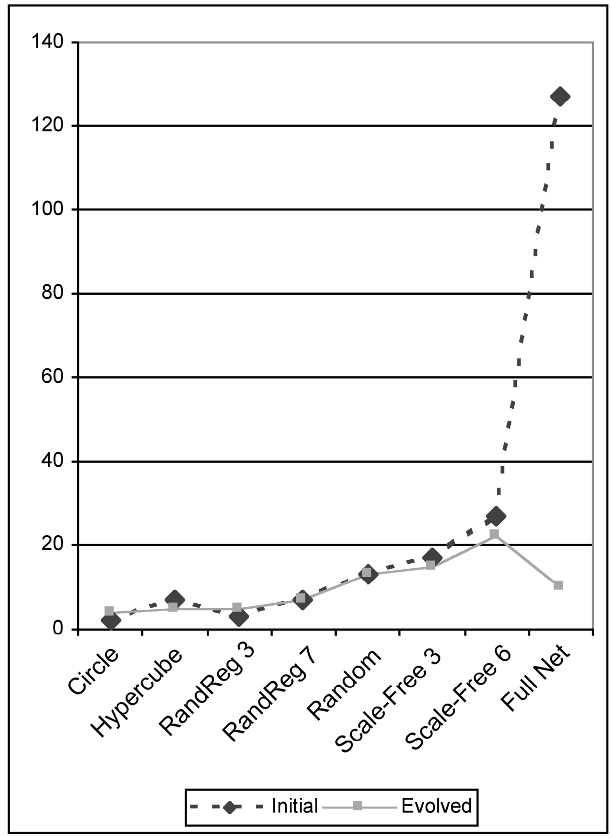

Figure 6 – Experiment 2, Maximum Degree

Experiment 2 gave us further evidence of the resilience of scale-free networks. We have already noted that each of the different starting topologies we considered successfully evolved to a more efficient and robust network, but one anomaly stood out. When we compared the maximum degrees of each topology before and after adaptation, we noticed that the full network starting topology was better able to balance the thinning than either of the scale-free starting topologies we tested. The full network started with nothing but hubs, in fact hubs of a maximum size, but unlike a scale-free network they were not rare in the graph. Our balancing bias had the effect of thinning the full network evenly, and in the end it was able to produce a topology that was very similar to what emerged from the circle, random, random-regular, and hypercube starting topologies. However, the scale-free starting topologies were resistant to this effect, tending to hold on to their power-law degree distributions even as they shed links. These results are visible in Figure 6, where the maximum degrees for each topology before and after adaptation are shown. Our conclusion is that our selfish node heuristics of thinning and folding, even with the addition of the degree balancing bias, are not sufficient to deconstruct existing scale-free networks. Further effort appears to be needed to penalize the hubs if we wish to not only prevent them from forming but also eliminate them when they pre-exist.

5. Conclusions and Future Work

Our basic hypothesis was that our local heuristics for adaptation would lead to network performance improvements. When we fixed the link density as we did in Experiment 1, we saw only a slight reduction in bandwidth used after topology evolution, but achieved a savings in query routing time and a strengthening of weakly connected nodes. When we allowed hubs to emerge during adaptation, our savings in time and bandwidth were greater than when we instead encouraged degree balancing. Experiment 2 then showed that many kinds of graphs could be significantly thinned without becoming too brittle, and graph diameters simultaneously kept from growing much larger than $\log_2 N$. At the same time, by making some effort to balance the degree of the nodes, we kept the graph from being as vulnerable to targeted attacks or accidental catastrophic failures.

The fact that hubs emerged as a side effect of thinning and folding was a surprise, leading us to conclude that there may be numerous forces at work in dynamic graphs to produce scale-free topologies. Growth with preferential attachment is certainly sufficient, but locally selfish incremental improvement choices may lead to power-law degree distributions as well. So hubs happen, and unless you try to prevent them, they seem to be nature's way of finding good solutions to many network problems.

The other major conclusion we drew from this work was that our basic selfish behaviors can be augmented to favor balancing degrees, and this can prevent a hub problem from getting worse. However, it is not sufficient to reverse the process if hubs already dominate the graph. In other words, scale-free networks are easy to make and hard to undo, at least using local adaptive strategies like our thinning and folding operations.

There are many directions we might wish to take this work, but our immediate research plan is to continue to add elements of complexity to the models, to make them more realistic. The message-passing model we have used so far does not capture real-world communication latencies, and perhaps that could be adequately modeled as statistical noise. This would affect the odd-cycle result, but overall we would still expect a graph to thin less-productive links over time. We mentioned the importance of join and reconnect protocols, which are the practical means to deal with the general peer transience problem. We would like to begin modeling both arrival and departure rates for peer nodes, as well as capacity differences and supply and demand fluctuations. For example, it would be nice to see whether the ultra or super peer strategies taken by existing P2P file-sharing implementations would emerge by adaptation to environmental conditions, versus being architected from the top-down.

Another approach we would like to take is to view this problem as an optimization problem with constraints, as mentioned briefly in Section 2. For example, given a target regular degree value (or constraints that bound its range and/or variance), can we find a network topology that minimizes overall bandwidth consumed by a set of distributed searches initiated from each node? Such a formulation might allow us to address this problem using probabilistic graph theory methods, or operations research techniques. Similarly, we would like to try additional forms of heuristic search such as a Genetic Algorithm [14] exploration of topological space (based on a fitness function that incorporates measures of time and space efficiency as well as network robustness).

Although our simulation results were computationally expensive to produce, we believe it is feasible to run these kinds of experiments on larger graphs than we have examined here. We need to ensure that the results hold for larger graphs, which perhaps can be established analytically if not empirically.

We want to examine more topological families as well, so we plan to implement a random graph generation algorithm that can produce graphs with degree distributions fit to arbitrary probability density functions. This would allow us to look at particular power law functions with different exponents, and to consider graphs with unusual degree distributions like normal or uniform.

Finally, our simulations need to become more like real networks with autonomous nodes operating in parallel. Changes to the graph topology can happen much quicker when many links are added or removed per simulation time step, and convergence may be harder to ensure in highly dynamic environments. Real P2P systems involve more than just a distributed search protocol, so we plan to simulate various joining and connectivity maintenance protocols as well as approaches to distributed indexing. A full discrete event simulation testbed will probably be needed to allow us to consider these kinds of additional complexities.

The initial explorations undertaken here have increased our basic understanding of tradeoffs between efficiency and robustness in P2P networks. We have demonstrated that active topological maintenance can be helpful for P2P systems, and we can now proceed to develop specific protocols to support such adaptive operations.

6. References

[1] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman, "Search in Power-Law Networks", Physical Review E 64, 046135, 2001.

[2] R. Albert, A. L. Barabási, "Statistical Mechanics of Complex Networks", Reviews of Modern Physics, 74, 2002, pp. 47-97.

[3] B. Bollobas and O. Riordan, "The Diameter of a Scale-free Random Graph", to appear in Combinatorica, 2003.

[4] J. Carlson and J. Doyle, "Highly Optimized Tolerance: A Mechanism for Power Laws in Designed Systems", Physical Review E 60, 1999, pp. 1412-1422.

[5] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf, "Achieving Scalability and Expressiveness in an Internet-Scale Event Notification Service", Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing, Portland OR, USA, July 2000, pp. 219-227.

[6] M. Castro, P. Druschel, Y. C. Hu, A. I. T. Rowstron, "Topology-Aware Routing in Structured Peer-to-Peer Overlay Networks", Future Directions in Distributed Computing, 2003, pp.103-107.

[7] R. Cohen and S. Havlin, "Scale-free Networks are Ultrasmall", Physical Review Letters. 90, 058701, 2003.

[8] B. F. Cooper, H. Garcia-Molina, "Studying Search Networks with SIL", 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03), Berkeley, CA, February 2003.

[9] S. El-Ansary, L. O. Alima, P. Brand, S. Haridi, "Efficient Broadcast in Structured P2P Networks", 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03), Berkeley, CA, February 2003.

[10] A. Fabrikant, E. Koutsoupias, and C. Papadimitriou, "Heuristically Optimized Trade-offs: A New Paradigm for Power Laws in the Internet, Ninth International Colloquium on Automata, Languages and Programming (ICALP '02). Málaga, Spain; 8-13 July 2002, pp. 110-122.

[11] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On Power-Law Relationships of the Internet Topology", ACM SIGCOM '99, Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, edited by L. Chapin, ACM, New York, 1999, pp. 251-262.

[12] I Foster and A. Iamnitchi, "On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing", 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03), Berkeley, CA, February 2003.

[13] The Gnutella Protocol Specification, version 0.4, rev 1.2. http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf

[14] J. H. Holland, Adaptation in Natural and Artificial Systems. Ann Arbor, MI: The University of Michigan Press, 1975.

[15] R. Mahajan, M. Castro and A. Rowstron, "Controlling the Cost of Reliability in Peer-to-Peer Overlays", 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03), Berkeley, CA, February 2003.

[16] G. Pandurangan, P. Raghavan, and E. Upfal, "Building Low-Diameter P2P Networks", Proceedings of the IEEE Symposium on Foundations of Computer Science, 2001.

[17] M. Ripeanu, I. Foster, and A. Iamnitchi, "Mapping the Gnutella Network: Properties of Large-scale Peer-to-Peer Systems and Implications for System Design", IEEE Internet Computing Journal, 6(1), 2002.

[18] S. G. Ziavras, "RH: A Versatile Family of Reduced Hypercube Interconnection Networks", IEEE Transactions on Parallel and Distributed Systems, Vol 5., No. 11, November 1994, pp. 1210-1220.