THESIS

ON THE ROLE OF TOPOLOGY IN AUTONOMOUSLY COPING WITH FAILURES IN CONTENT DISSEMINATION SYSTEMS

Submitted by

Ryan Stern

Department of Computer Science

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Fall 2014

Master's Committee:

Advisor: Shrideep Pallickara

Michelle Strout
Daniel Turk

ABSTRACT

ON THE ROLE OF TOPOLOGY IN AUTONOMOUSLY COPING WITH FAILURES IN CONTENT DISSEMINATION SYSTEMS

Content dissemination systems provide a substrate that allows large numbers of entities to communicate with each other. These entities could be processes, sensors, and networked instruments that produce and consume data streams. To ensure scaling, the content dissemination substrate comprises a large number of distributed nodes. As the number of participating nodes increases, the likelihood of failures also increases. These failures can occur for any number of reasons, including: faulty hardware, programmer or user error, power failure, and network outages. Node failures can result in *partitions* with the original set of connected nodes disintegrating into smaller, disjoint subsets. Brewer's CAP theorem limits the choices for a partitioned system: availability or consistency but not both. It is therefore desirable to ensure that partitions are less likely.

This thesis explores how nodes comprising the content dissemination system can be organized into topologies with the objective of improved partition tolerance. The topologies we consider are based on random, regular, power law, and Watts-Strogatz small world graphs. Connections within these topologies can account for network proximity and are suitable for real-time communications. We explore specific attributes of a topology that contribute to its partition resiliency, such as clustering coefficients, distribution of random links, and preferential attachment. Metrics we use to profile suitability of different topologies include: communication path lengths, migration of workloads, and the impact on system throughput. This research will allow designers to choose topologies or configure metrics to achieve performance objectives and the degree of partition tolerance.

ACKNOWLEDGEMENT

This research is supported by a grant from the US National Science Foundation's Computer Systems Research Program (CNS-1253908).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Content dissemination systems are used to distribute content between large sets of producers and consumers. The underlying dissemination mechanisms in these systems can be based on publish/subscribe [1] [2] [3], peer-to-peer [4] [5] [6], ad hoc routing [7] [8], or IP multicast [9] [10]. These content dissemination systems are large and routinely comprise tens of thousands of nodes.

As the scale in such systems increase, the probability of failure also increases. A reactive strategy deals with such failures after the fact by initiating measures to restore system properties (such as number of connections and network partitions) that were violated. Systems also prepare for failure by building redundancies into the system. In such systems, the redundancies are in the form of hot standbys (or shadow processes) that are ready to take over as soon as the primary fails.

Publish/subscribe systems demarcate the producer and consumer roles. These systems rely on a middleware to route content from the producer to the consumer. The middleware is extensible though statically configured with distributed nodes that provide dissemination capabilities. The middleware allows for decoupled interactions between the entities: producers and consumers need not know about each other and they may also not be up-and-running at the same time.

Peer to peer systems can either be structured or unstructured [5] [6] [11]. In unstructured P2P systems, peers rely on time-to-live (TTL) counters to control the scope of disseminations. Structured P2P systems on the other hand impose a logical overlay on the peers. The most dominant form of these structured P2P systems are Distributed Hash Tables (DHTs). Structured DHTs are highly suited for discovery operations with their efficient bounds for lookups. In DHT-based systems, each node maintains local information about a small subset of nodes ($log\ N$) and uses this when routing discovery requests. The objective while routing requests is to ensure that each hop takes the request closer (in ID space) to the destination. Routing schemes rely on increasing the number of matching bits between the intermediate nodes and the final destination as hops are taken en route to reaching it.

Dissemination of content in DHTs is performed in the ID space that generally has no correlation with network proximity [5]. Nodes that are close to each other in the ID space may incur considerable network delays during communications. Similarly, nodes within the same cluster may be assigned identifiers that are far apart in the ID-space. Nodes are assigned identifiers randomly, and these IDs are usually 128-bits when UUIDs are used or 160-bits when SHA-1 digests are used on randomly generated strings. Identifier assignments are dissociated from network proximity. In DHTs it is often the case that logical hops between "neighboring" peers can often involve intermediate hops through the logical-ID space that are multiple network hops away.

In P2P systems such as BitTorrent [6] that are not based on DHTs, chunks of a file are dispersed on multiple nodes. Retrievals of a file target multiple nodes that hold different chunks and the transfers take place in parallel. The focus is on reducing the load at a given node and not necessarily on reducing network footprints during communications. DHTs and BitTorrent are not intended for real time routing of content.

We posit that topology plays a key role in systems designed for real-time routing of content. Our research is based on systems that exploit or account for network proximity. In such systems, proximate nodes comprise logical clusters. A *topology* is the organization of the links and nodes within a graph. Based on connections maintained within the system, these topologies may have different properties that determine suitability for disseminating content.

Some properties of a topology include path length and clustering coefficient. The *path length* corresponds to the average number of hops that must be traversed during communications between any pair of nodes in the system. A topology's *clustering coefficient* measures the degree to which nodes tend to cluster together. A connected topology refers to a situation where a node can reach any other node in the system. Network partitioning refers to a situation where the topology disintegrates into two or more disjoint subgraphs, where a node can only reach other nodes in the subgraph that it belongs to.

The connectivity between the nodes leads to different types of topologies. There are several aspects that we consider when accounting for connectivity within a topology:

- The average number of connections originating from any given node.
- The degree of randomness in the connections. This taps into the notion of *locality* of connections. These are situations where the topology includes subgraphs where the constituent nodes are densely connected to each other i.e. the number of connections originating from a node in the subgraph is dominated by connections to other nodes within the subgraph.
- Variability in the number of connections that originate from different nodes within the topology, which plays an important role in power-law topologies.

Associated with every link in the graph is a weight that indicates desirability of using that link for transmissions. This weight can be biased for latency, throughput, jitter reduction, or loss rates. We rely on these link weights to compute a Minimum Spanning Tree (MST). This can then be used to support network-aware one-to-one and one-to-many communications.

We investigate the behavior of potential topologies used in Funnel, a new decentralized publish/subscribe content dissemination system that we are building. In our system, each node connects to a subset of the nodes within the system. Once these connections are set up, we explore the resilience of different topologies to failures and partitioning. Nodes work with local information and failures that they can observe i.e. situations where connections to a node from other nodes are severed.

Research Questions:

In this thesis we explore the following research questions:

(1) *What is the impact of topology on resilience to network partitions?* Specifically, we are interested in exploring when partitions occur in different topologies. For this thesis, nodes do not initiate steps to preserve connection level thresholds.

(2) *Which features of the topology contribute to this resilience?* Two aspects of connectivity play a role here. The first aspect is the degree of randomness among connections within the graph as it relates to locality. The second aspect is the number of connections originating from a node and whether variability in this number leads to nodes with a high degree of connectivity.

(3) *How well do different topologies cope with failures?* Specifically, how does the impact of failures manifest themselves in terms of variations in the load experienced by other nodes in the system? Is this redistribution of workloads skewed or is it more or less evenly distributed across nodes comprising the topology?

Thesis Contributions

This thesis describes how topologies play a role in autonomously preserving several features of interest. Specifically, we have explored not just how topologies are resilient to failures but also what aspects contribute to this resilience. This research also quantifies the impact of failures on communications, partition tolerance, migration and distributions of workloads, and the concomitant impact on system throughputs. The findings of this research will be incorporated into our content dissemination system, Funnel. Designers of new and existing content dissemination systems (including those mentioned previously) can also utilize this research to choose between topologies or configure metrics for a specific topology in order to achieve their performance objectives.

Thesis Organization

The remainder of this thesis is organized as follows. In Chapter 2, we provide background information about Funnel and the topologies being investigated in this thesis. In Chapter 3, we provide the methodology for our experiments on the influence of clustering coefficient, failure resistance, and the influence of topology on throughput. In Chapter 4, we present our findings based on the experiment results. Chapter 5 concludes the thesis.

CHAPTER 2

BACKGROUND

When working with decentralized distributed systems such as Funnel, it is important to understand what effect failures have on the topology in use. These failures may change the properties of the topology including average path length and clustering coefficient (see Section 3.2), or may change the traffic patterns experienced within the system, leading to bottlenecks and lower throughput. Knowledge of how failures affect the system is important for determining the best policy for local decisions made by each decentralized node in the system, decisions that either maintain or degrade the topology.

Funnel is a content distribution system that we are building. The results of this work will determine the topologies that are used by our system moving forwards. The system combines aspects of peer-to-peer systems with that of publish/subscribe. Unlike traditional publish/subscribe systems that require a dedicated middleware to perform the content dissemination, in Funnel the nodes (either producers or consumers) organize themselves into a topology that is then used for disseminating content.

Funnel allows nodes to organize into different topologies including the ones discussed in this thesis. A registry controls the composition of the logical overlay and this registry can have shadow processes that allow it to sustain failures. For each topology, once a steady state is reached, decisions at individual nodes allow preservation of the topology by maintaining the requisite local properties such as the degree, locality, and randomness in the connections initiated by a node. Funnel relies on Minimum Spanning Trees (MST) to route data. A failure of a node within a cluster of proximate nodes is repaired locally without having to recompute the global MST. The nodes utilized in our simulations are full-fledged Funnel nodes with support for disseminating content. During simulations, rather than using TCP or UDP for communications, the system relies on memory-based communications.

This thesis looked at four topologies, the first of which was the regular ring lattice. Furthermore, three topology parameters —N, K, and R — control how these topologies are created. N specifies the number of nodes in the topology, K specifies the mean degree (number of links) of a node, and R

specifies the percentage of links that are random, long links. The regular ring lattice, shown in Figure 1, is a graph with nodes connected in a ring. With K equal to two, a node P_i will have a connection to nodes P_{i-1} and P_{i+1}. With K equal to 4 there will also be additional connections to nodes P_{i-2} and P_{i+2}, and so on for higher K. Regular ring lattices have some distinctive properties when it comes to path length and clustering coefficient. The average path length of a regular ring lattice is longer than other topologies investigated in this thesis, since the only route is around the ring, potentially skipping every few nodes based on K. The ring of connections also gives the topology its high clustering coefficient, approaching .75 as K increases, since nearby nodes will be connected to each other.

If regular ring lattices exist on one end of a spectrum, then random graphs would exist on the other. As its name suggests, this topology consists of random connections between N nodes. There is no guarantee that every node will have the same number of links, but the average number for all nodes will be K. Random graphs have a desirable short path length, but a low clustering coefficient. A random graph is shown in Figure 1.

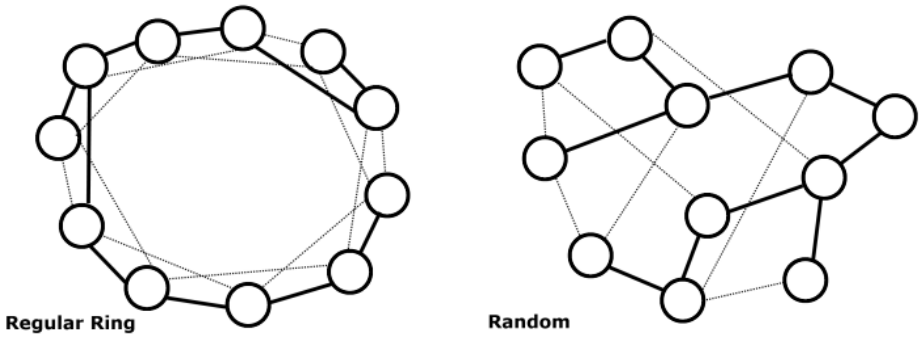

Figure 1: Regular ring lattice (left) and Random (right).
An example of the regular ring lattice and random graph topologies consisting of only a few nodes and links. The bold lines represent a possible MST.

Small-world networks can be considered the middle ground between regular ring lattices and random graphs [12]. These small-world networks consist mostly of short links to neighboring nodes as in a regular ring lattice, but a small percentage of links are random. All three topologies can be created using the same algorithm by adjusting the parameter R. This process will be described in greater detail in

Section 3.1. Small-world graphs can be identified by their high clustering coefficient and low path length. To be considered small-world, the path length should grow proportionately to $\log N$ as more nodes are added to the topology. An example small-world network is shown in Figure 2.

In a power-law based topology, the greater the number of connections at a node, the greater the probability of a new connection being initiated to it [13] [14]. New nodes preferentially initiate connections to nodes with a higher number of connections. This results in the emergence of *hubs*, which are nodes with a disproportionate share of the number of connections in the system. Power-law networks have a very short path length, which often grows proportionally to $\log(\log N)$. This is due to the hubs that form in power-law networks. The hubs have the highest number of connections to other nodes, and often connect entire sub graphs. The probability of a node having a given number of connections to other nodes follows a power law, hence the name. An example power-law network is shown in Figure 2.

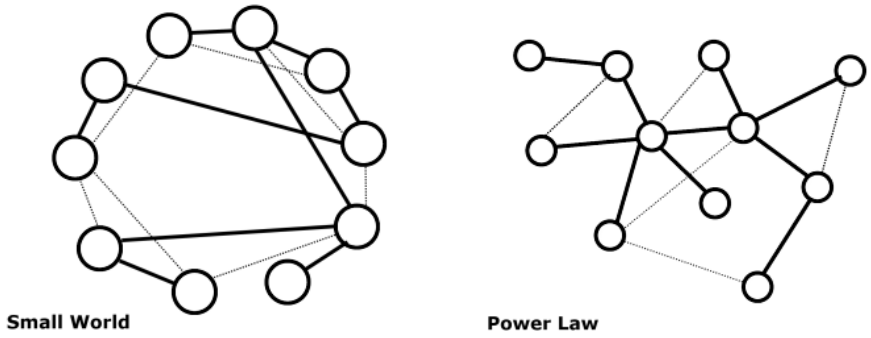

Figure 2: Small world and power law networks.
An example of small-world and power-law topologies. The bold lines represent a possible MST.

CHAPTER 3

METHODOLOGY

The results in this thesis were gathered from three simulation experiments. The first of the three gathers average path length and clustering coefficient data from topologies with different parameters. This experiment produces data that is used to reason about the type of topology being considered in the remaining two experiments. The second experiment begins looking at various topologies' robustness in the face of failures. The third and final experiment included in this thesis investigates throughput and bottlenecks as these failures are introduced. The methodology for each of the three experiments is described in detail in the following sections. Results are detailed later, in Chapter 4.

3.1 GENERATING TOPOLOGIES

The simulation starts by creating an instance of the topology being investigated. Although the results of the experiments will be used to aid decisions in a decentralized setting, the experiments are controlled by a process with access to the global state.

Regular ring lattice, *small-world*, and *random* topologies are created using the same method, based on [12], with various parametric adjustments to achieve a particular topology. The addition of links within the system is governed by two parameters: R which is related to the proximity/ locality of a connection and K which measures the number of connections at a node. The method begins by creating N nodes and arranging them in a ring. Links are then added to the nodes, based on the value of R. In this thesis, R represents the probability that a given link will be *long* as opposed to *short*. A short link is a link that forms a connection to a node in a local cluster. These nodes will have alternate paths between them also consisting of short links. The other parameter used in this thesis is K. K is related to the degree of a node, there will be on average K links for each node. That means that when R is equal to 0, all links are short and each node has K links to its nearest neighbors. This is commonly referred to as a regular ring lattice.

The actual process of generating the topology is as follows. Short links are added, connecting node i to node $i + j$ for all $0 \leq i < N$, $1 \leq j \leq \frac{K}{2}$. The links have a random chance of being placed or skipped for each possible link position, but it is guaranteed that exactly $N(1 - R)\frac{K}{2}$ short links will be added in this phase.

At this point, the topology does not have any of the desired small-world properties. The second stage remedies this by adding long links. Long links connect a node to a randomly selected distant node in the graph. The generator used in this thesis makes no distinction between a distant node and one forming a local cluster when randomly selecting a node. This process continues until the average number of links per node is K. Since the node is selected at random, an R value of one will produce a random graph.

The *power-law* topology is generated using a separate algorithm. The algorithm begins by assigning the number of outgoing links that each node is allowed to have. Iterating from degree $d = 1000$ down to $d = 1$, the probability of a node having d links is computed. The formula used is $P(d) = d^{-a}$, where a is a configurable variable. The variable is automatically adjusted such that 20 percent of the nodes hold 80 percent of the links. For K equal to 10 and N equal to 10000, a is near 2.1. Following the computation of $P(d)$, the probability is multiplied by N to give the number of nodes with the given degree d. For example, if $P(2) = 0.5$ and $N = 10000$, 5000 nodes will have degree 2. For $P(d) * N < 1$, there is $P(d)$ chance that one node will have that degree.

Once the degrees are assigned, the links are added. The generator iterates over the nodes several times, adding one link per node as long as that node has not exhausted its assigned number of links. The other endpoint of the link is selected at random from the remaining nodes that have not had their assigned degree satisfied.

3.2 INFLUENCE OF CLUSTERING COEFFICIENT

Here we investigate relationships between the average path length and average clustering coefficient across various topologies. Following the analysis, this information will be used to find

relationships between average path length, clustering coefficient, and the topology's tolerance to failures. This information can also be used to infer the *type* of topology. We analyze various types of topologies including random graphs, small-world networks, power-law networks, and variations in-between. Although multiple topologies are covered, extra attention was paid to the small-world network topologies with a large number of nodes, as these topologies showed the best results.

The K and R parameters are adjusted to change the topology. Our experiments tested K values ranging from 2 to 30. For R, values of 1, 0.8, 0.5, 0.2, 0.1, 0.08, 0.05, 0.02, 0.01, 0.008, 0.005, 0.002, 0.001, and 0 were tested. Since the topology being tested can potentially contain thousands of nodes, low values of K can still add a significant number of links to the topology. We investigated properties of various topologies comprising 100, 1000, 10000 nodes. This helps in identifying which of the tested topologies are, in fact, small-world, power-law, random, or regular ring lattice.

Once the topology construction is completed, the topology's properties can be tested and recorded. The simulation computes the average path length and clustering coefficient to test if the generated network has the properties of a small-word network or power-law network.

The *path length* refers to the shortest number of hops to get from one node to another. We do not consider weights that may exist per link in a live system when determining the average path length, only the hop count is considered. To begin computing the average path length, two nodes are selected at random. Breadth-first search is used to find the smallest number of hops from the first randomly selected node to the second randomly selected node. This is repeated for 250 random pairs of nodes, and an average taken.

The *clustering coefficient* is a measure of how much nodes cluster together. To compute the network average clustering coefficient, the local clustering coefficient is computed for each node n in the network, and an average taken. The local clustering coefficient of a node is computed by dividing the total number of all directly neighboring nodes' links that connect to other direct neighbors by the number of *all* possible links between those neighbors. For example, let some node n have 4 neighboring nodes, as is depicted by bottom node in Figure 3. In this figure, solid lines are actual links and dashed lines are the

remaining possible links. The leftmost neighbor has four links, but only two of those links connect to another neighbor of *n*; only two of the three links is counted in the clustering coefficient. There are six possible links between the four neighboring nodes, two of which are actual links. This leads to a local clustering coefficient of 2/6 for *n*. In the general case where n has c neighbors, the total number of possible links is $\frac{(c)(c-1)}{2}$.

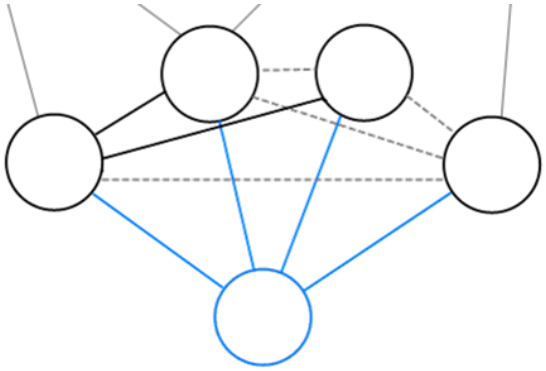

Figure 3: Computing local clustering coefficient.
Solid lines are actual links, dashed lines are other possible links. The bottom node has four direct neighbors. Two of six possible links exist between these neighbors.

3.3 FAILURE RESILIENCE

Here, we investigate the resilience of topologies to sustain failures. Specifically, we determine the percentage of nodes that can fail before a partition forms. We are particularly interested in examining how resilient small-world networks are compared to other well-known topologies.

In this experiment, we are only looking at the ability to reach any other node after a failure takes place. In other words, we are only checking if a partition has formed; how each topology affects the ability to reach any given node using only local decisions is a subject for future work. Our objective is to identify the parameters and properties, such as clustering coefficient and proportion of random links, which provide the best resilience to partitions. Before that can happen, though, data on how resilient each topology is to failures is collected. This is achieved using the process described below.

Topologies consisting of 10000 nodes were tested. From there, 100 nodes (1%) in the topology are randomly removed, or dropped. All of the dropped nodes' links are also removed at that time, a worst

case scenario compared with a situation involving only individual links failing. We are checking to see the point at which random failures cause the topology to partition into two or more distinct subgraphs. If the topology has not partitioned, another 100 nodes are removed, repeated until a partition forms. The percentage of nodes that were removed is then recorded. We remove 1% of the nodes at a time since checking for a partition is an expensive operation when working with large graphs.

The above process is repeated 500 times and an average is taken, generating a new instance of the topology each time. Averaging over several runs should reduce the impact of slightly overshooting the true percentage.

3.4 INFLUENCE OF TOPOLOGY ON THROUGHPUT

In this experiment we investigate the influence of topology on throughput in the face of failures. Like the previous experiments, a topology with 10000 nodes and a given K and R is generated with randomly assigned links based on the previously described rules. Following the generation of this topology, routes are determined using an MST and packets of data are sent across the system.

When the simulation starts, the 10000 nodes are randomly ordered in a list. This will be the order the nodes will be dropped in. Next, 300 pairs of source and destination nodes are selected from the end of this list. This process ensures that the source and destination nodes are not removed, only the intermediate nodes. These pairs are then used as the endpoints for all future routing. By repeating the same routes, the results truly reflect how throughput and bottlenecks are affected by routing changes caused by an increasing percentage of node failures, instead of path length differences introduced by using a different endpoint altogether. Following the initial topology generation and selection of endpoint pairs, the simulator routes data and removes nodes in iterations (or rounds) until the graph partitions. Each iteration involves two phases. The first phase involves determining the route for each source and destination node pair and then recording the nodes involved in that route. This process will be described in detail below. The second phase involves dropping a percentage of nodes. For this experiment, as with the rest of this work, 1% of the nodes were dropped randomly during each iteration. As before, when a node is dropped,

all of its links are also removed. Dropping the nodes will shift some of the burden onto other nodes in the system. One of the goals of this experiment is to identify in which of the investigated topologies this burden has the smallest effect on throughput and bottlenecks. Bottlenecks occur when the available resources have been consumed and nodes are unable to cope with demand. This leads to a reduction in throughput at those nodes.

For each routing pair, the data is always routed between that pair's source node and destination node. Note however, that the path in between these two nodes may not be the same each failure iteration. As mentioned, the selected source and destination nodes are the last to fail. With 300 paths selected, or 600 safe nodes out of 10000, the graph will often partition before these nodes would need to be dropped. The routing process begins by computing the MST for the given graph. This simulation uses Kruskal's algorithm to create the MST [15]. To aid in this process, all links in the graph are assigned a random weight value that remains constant for the length of the simulation. After generation, the source node is treated as the root of the MST, and the path to the destination node is extracted. From there, that path is used to determine the intermediate nodes to route a data packet through. When a node happens to receive one of these packets, a counter is incremented. Therefore, the counter at each node represents the total number of times that a given node was utilized to route data. Higher numbers indicate a busier node. This data can be used to identify which topologies are more prone to bottlenecks when faced with failures.

CHAPTER 4

RESULTS

The results of the experiments are presented over the next few sections, with one section per experiment. Following each result will be a discussion. Conclusions will be drawn later.

4.1 INFLUENCE OF CLUSTERING COEFFICIENT RESULTS

Our results from this experiment report the average path length and clustering coefficient for topologies containing 100, 1000, and 10000 nodes with various values of K and R. These results are useful in identifying properties associated with different topologies. The results gathered here are most valuable in conjunction with data from experiments described in sections 4.2 and 4.3.

Table 1 depicts some of the results collected during this simulation. Due to the large number of combinations of N, K and R, only the results from topologies with $N = 10000$ and $K = 10$ are shown in the table. The topologies shown have been labeled by type. This type is determined based on the properties of the topology; these properties have been listed in Chapter 2. Small-world topologies are determined by comparing the average path length when N is 100, 1000, and 10000. Topologies that show the average path length to be proportional to $\log(N)$ are labeled small world. Topologies that do not show this behavior, but do not have the properties of a regular ring lattice are labeled intermediate.

Average path lengths do not have a correlation to the fault tolerance of the topology; however, the same may not be true for the clustering coefficient. Looking at the clustering coefficients for the best performing topologies in Figures 4 and 5 reveals a trend, however. The clustering coefficient for topologies shown in the figures around $R = 0.1$ all tend to be near 0.57. The lowest coefficient was 0.46, when K was equal to 4. This quickly reached 0.57 as K grew, even as K was incremented to 30. Further investigation is required to determine if this value is significant for topologies with N other than 10,000.

Table 1: Average path length and clustering coefficient for various topologies with K=10.

Type	R Value	Avg. Path Length (Start)	Clustering Coefficient (Start)
Regular	0.0	501.75	0.667
Intermediate	0.0001	250.83	0.667
Intermediate	0.0002	168.36	0.666
Intermediate	0.0005	90.25	0.666
Intermediate	0.0008	65.02	0.666
Intermediate	0.001	55.46	0.666
Small World	0.002	34.41	0.664
Small World	0.005	19.48	0.661
Small World	0.008	15.11	0.657
Small World	0.01	13.53	0.655
Small World	0.02	9.97	0.643
Small World	0.05	7.23	0.610
Small World	0.08	6.29	0.579
Small World	0.1	5.92	0.559
Small World	0.2	5.23	0.426
Small World	0.5	4.37	0.189
Random	0.8	4.27	0.001
Random	1.0	4.26	0.001
Power Law	N/A	3.82	0.012

4.2 FAILURE RESILIENCE RESULTS

The results of the experiments testing failure resilience are depicted in Figure 4, Figure 5, and Figure 6. Data was generated independently for each pair of K and R values, as described earlier. To aid in identifying patterns and trends, the data from multiple topologies is grouped together by either K or R and then plotted. The average results from a topology with a given K and R is represented by a single point on the plot.

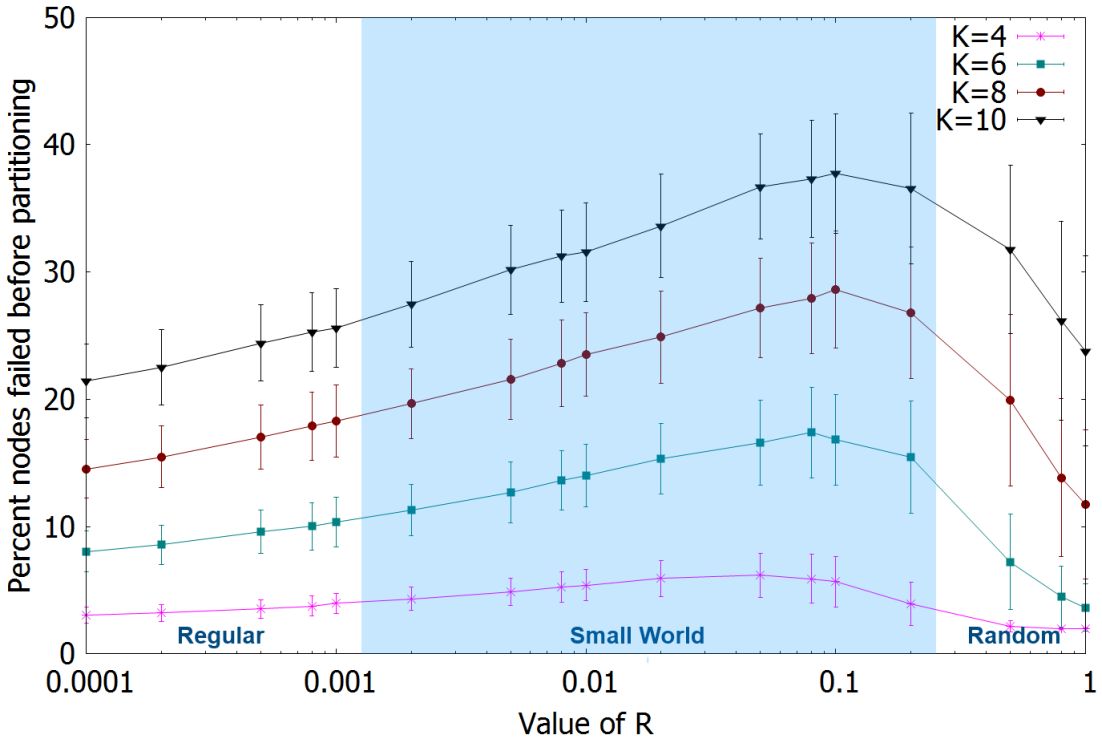

Figure 4: Percent nodes dropped for low values of K.
Topologies consisting of 10,000 nodes are subjected to random node failures. For various values of K and R, the percent of nodes that failed before the topology partitioned is plotted. Small world topologies can be found in the shaded region.

Figure 4 depicts a plot of the generated data after it has been grouped by the value of K, the mean node degree. The plot depicts failure resilience in terms of the percentage of nodes that failed before a partition emerged vs. the value of R for that topology. This figure is limited to K values 4, 6, 8, and 10 to improve readability. Similarly, Figure 5 depicts results of K values 10, 20, and 30. Note that both of these plots use a logarithmic scale for R.

The different lines on Figures 4 and 5 represent topologies with different values of K. While each of those lines represents topologies that have a different number of links, all of the points along a single line have the same number of links in the system. All variations in the percentage of nodes that could be dropped before the graph partitioned is influenced only by the way links connect the nodes in the graph, which varies from topology to topology.

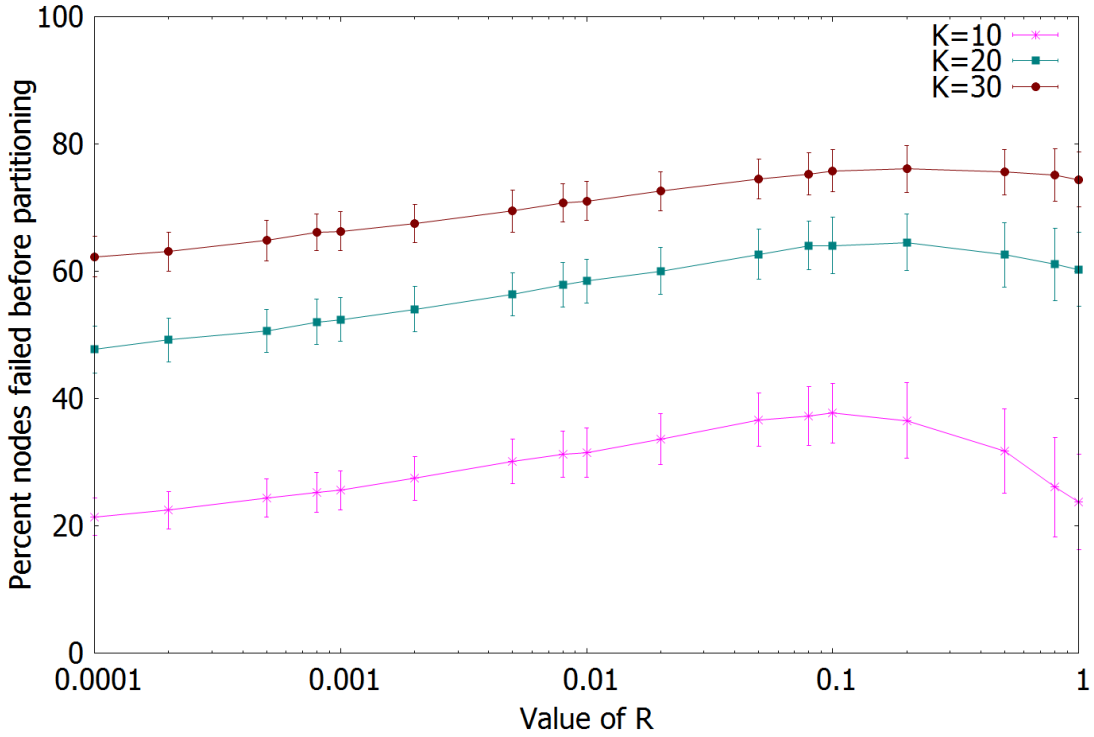

Figure 5: Percent nodes dropped for high values of K.
Topologies consisting of 10,000 nodes are subjected to random node failures. R is not as significant as K grows larger.

This second experiment provided us with important insight about the nature of various topologies with respect to failures. One such insight is that the significance that one topology plays compared to another varies with the number of links in the system. Refer to Figures 4 and 5. In Figure 4, the topologies consist of much fewer links compared with those depicted in Figure 5. The topologies generated with K equal to 6, 8 and 10 shows a distinctive bulge near R values equal to 0.1. This bulge represents a noteworthy difference in the percentage of nodes that could be dropped before partitioning. In the context of topologies with a different number of links, however, having a higher number of links reduced the topology's susceptibility to faults. For these topologies with a larger population of links, the bulge at $R = .1$, while still there, is more gradual and much less significant. The existence of these additional links increases the number of possible paths available to get from one node in the system to any other. This reduces the probability that a random failure will break all of the remaining paths. Also consider that although the topologies with a high value of K show greater resistance to failures than lower K, such

topologies may not be feasible for some applications due to the large number of connections, at 20 links per node on average. Most systems only have a single physical link connecting a machine to the rest of the network. This single physical link will need to be split 20 times in order to use such a topology, as the bandwidth available from a physical link is limited and must be shared. This places increased importance on the properties of topologies with lower values of K.

From Figure 4, we can see that topologies with R equal to 0 are less tolerant to failures than the topologies with $0 < R < 1$. These are regular ring lattices, since 0% links were random long links. Similarly, the topologies with R equal to 1 are random, since we randomly connected 100% of the links in the system. Refer to Table 1 for the type of topology between these values of R. Identifying the topologies in this range is important, since the topologies in this range were able to tolerate a greater percentage of failures.

Figure 4 indicates that the tested topologies that had R near 0.1 demonstrate the best tolerance to random failures. Interestingly, all topologies tested were able to handle the highest number of failures when R was very close to $R = 0.1$, no matter the number of links in the topology. Compared with regular ring lattices and random graphs, between a 5% and 15% more of the total nodes in the system can also fail before a partition forms when the R value was near 0.1. This was largely dependent on the value of K, with 10, seeing the best improvement over random graphs and regular ring lattices.

In addition to the results grouped by K, the results were also grouped by R. Figure 6 depicts this grouping for the two best and two worst performing values of R. Grouping the data this way allows additional conclusions to be drawn that were not obvious from grouping by K alone. Figure 6 allows us to determine which values of R produce topologies that are the least susceptible to failures. The plot indicates that the topologies performed relatively poorly when R equaled 0, the regular ring lattice. Figure 6 reinforces the findings from Figures 4 and 5. We now clearly see the influence of more links on the topologies resilience to failures. From $K = 4$ through $K = 16$, there is a steep improvement in the percentage of nodes that can fail. The rate that nodes can fail versus the number of links added to the

topology begins to slow after this point. We can further see that the influence of *R* diminishes as more links are added to the system. Despite this, topologies with *R* set to 0.1 and 0.2 appear to do much better than other values of *R*, no matter what value of *K* is being investigated.

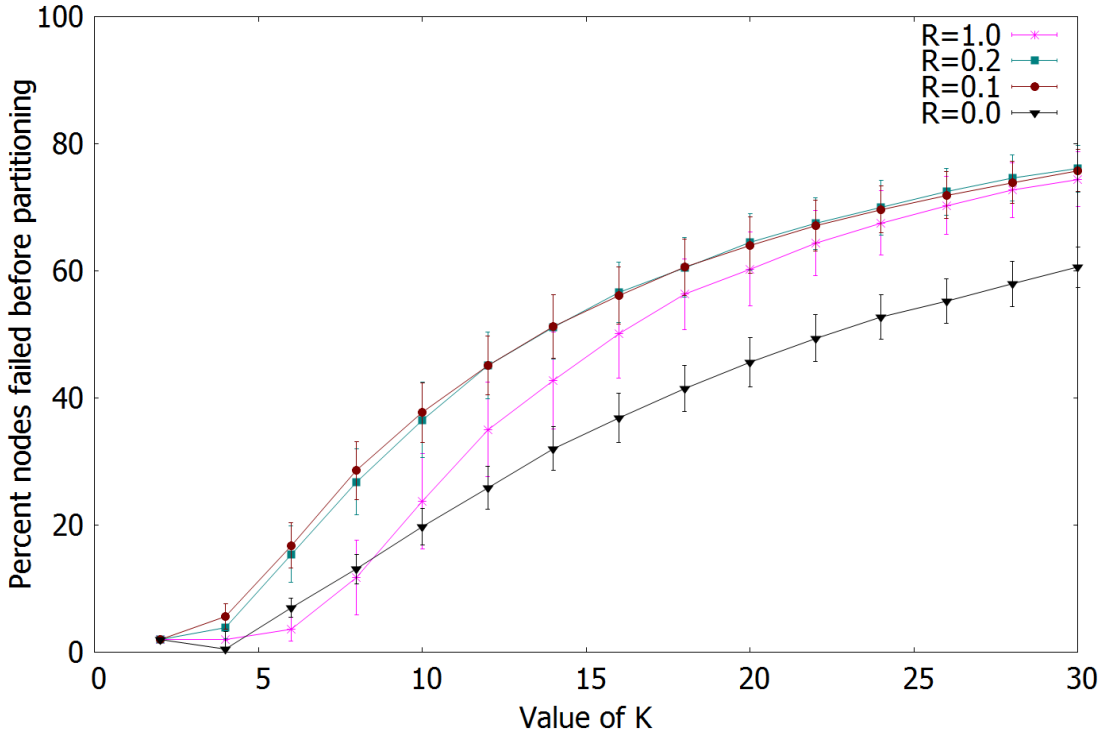

Figure 6: Failure resilience results grouped by R.
A different view of the data shown in Figures 4 and 5. This figure shows the influence of K on the failure resilience. R near 0.1 shows the best resilience to failures, regardless of K.

The final topology investigated in this thesis, power-law networks, faired very poorly compared with the other topologies investigated so far. In power-law networks 20 percent of the nodes contain 80 percent of the links in the topology. The average degree of each node, *K*, was 10, the same as the most promising small world topology discussed previously. This means that there are the same number of nodes and links in both topologies. The largest degree of a node in the system was near 105, and varied slightly from run to run due to the probability of the topology containing nodes with a higher degree.

When subjected to random node failures, the power law topologies partitioned during the first round of failures. Recall that one percent of the nodes randomly fail during each round. The reasoning for this is in the topology's hubs. Hubs are nodes with a very high proportion of links. Many smaller groups

of nodes connect to each other via a hub. Hubs often have direct connections to other hubs in the system, allowing distant nodes to be reached in only a few hops. Nodes in one group almost always connect to other groups via hubs, due to the higher probability of nodes with low degree. The hubs constitute single points of failure: if a hub fails, the topology will partition.

Table 2 shows the percent of nodes that failed for various topologies being tested. As in Table 1, this table reflects topologies with $N = 10000$ and $K = 10$. Topologies with K other than 10 show similar results. It can be seen that the small-world topologies are best at surviving random node failures. They regularly survive over 30 percent node failures, sometimes reaching the low 40s, as when $R = 0.1$. This is significantly better than random, regular, and power law topologies, handling over twice the number of failures. It is clear that topology can strongly influence failure resilience.

Table 2: Average path length and clustering coefficient after failures.

Type	R Value	Percent Failures	Avg. Path Length (Failures)	Clustering Coefficient (Failures)
Regular	0.0	19.6	564.65	0.667
Intermediate	0.0001	21.6	373.14	0.666
Intermediate	0.001	26.2	99.37	0.665
Small World	0.01	31.8	21.43	0.654
Small World	0.02	34.4	14.97	0.643
Small World	0.05	37.5	10.03	0.609
Small World	0.08	39.3	8.46	0.578
Small World	0.1	40.3	7.86	0.558
Small World	0.2	36.5	6.40	0.425
Small World	0.5	37.3	5.11	0.188
Random	0.8	22.6	4.61	0.001
Random	1.0	11.7	4.41	0.001
Power Law	N/A	1.0	3.86	0.012

4.3 INFLUENCE ON THROUGHPUT RESULTS

In this section we discuss how throughput changes in the face of failures. Figures 7 through 13 depict data collected from testing a regular graph, small-world graph with $R = 0.1$, and a random graph. Each of these topologies initially consisted of 10,000 nodes and K set to 10. The figures depict the node

ID from 0 to 9999 on the x axis against the number of packets out of 300 seen at that node during that failure iteration. Except for power-law, there is one figure depicting the topology before any failures, and one figure depicting the topology soon before it partitioned. Throughout this section, the terms low usage, medium usage, and high usage are used. These terms refer to nodes receiving less than 10% of the packets, between 10% and 33% of the packets, and over 33% of the packets respectively.

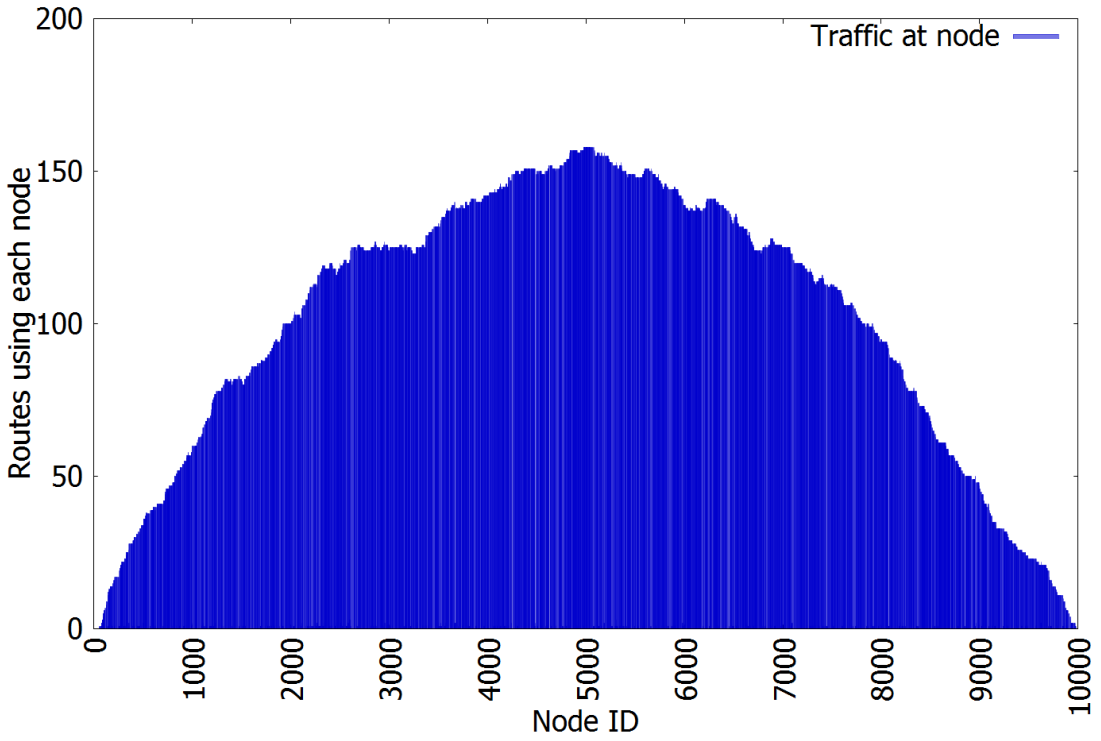

Figure 7: Regular ring lattice without failures.
Shows how many data packets were routed through each node in the topology, out of 300. This figure depicts the topology before any failures.

The first of the topologies discussed in this section are regular ring lattices. Figure 7 depicts the results of the throughput test before any failures. The results depicted in this figure were unexpected, but make sense considered how the experiment was carried out. Since the topology being investigated is a regular ring lattice, the only way to get from one node to another is to travel along the ring. Only a few nodes can be skipped over at a time, depending on what K happens to be. With K equal to 10, 4 nodes can potentially be skipped over per hop. If the shortest path was being taken, one would expect the number of times a data packet was seen at a given node to be roughly equal. For this experiment, however, the data

packets are not being routed using an algorithm that provides the shortest path between nodes, but with an MST instead.

Instead of providing the shortest path between two endpoints, an MST provides the shortest path to reach all nodes in the tree. We use an MST since that is how we plan to initially route data in Funnel. Unlike selecting shortest paths for each pair of endpoints, with an MST, there is a branching path through the system. When only routing data between two endpoints, this tree reduces down to a long chain of nodes, the branches that would take data to extraneous nodes in the topology are simply ignored. Since the MST is only generated once and all pairs of endpoints use it, the traffic for all pairs of endpoints is sent through the same nodes on the way to the destination. This long single path results in the distribution seen in Figure 7, even though the selected endpoints are random. The data packets are routed around the ring, traveling through a large number of nodes along the way. There are no hubs that become overloaded, all nodes in the system see a significant portion of the traffic, with 25% of the nodes routing nearly 50% of the packets sent. Over 50% of the nodes were responsible for routing 30% of the packets.

Figure 8 depicts the same regular ring lattice examined in Figure 7, except now the plot depicts the topology as it was after nodes had been dropped, but before it partitioned. Most of the MST generated at this phase is similar to the one used to route traffic without failures, since the weights per link are constant for the duration of the simulation. In a real system, the weights governing the selection of links to be used in the MST would likely have changed at some point in response to the current load. Dropping nodes did not significantly change the overall throughput, although it did shift the center of the MST. This shift greatly changed traffic seen at individual nodes. At least when using an MST for routing, failures do not seem to have any significant effect on overall system throughput and bottlenecks for regular ring lattices, but it can have a significant impact on individual nodes. We were not planning on using regular ring lattices in Funnel to begin with due to the long average path length, but the additional high utilization of a large percentage of nodes when using a MST ended any possibility of it being used at all. Of course, it is possible to generate multiple MSTs at various points to help lessen the load at some of the nodes, but there are other topologies that perform better still.

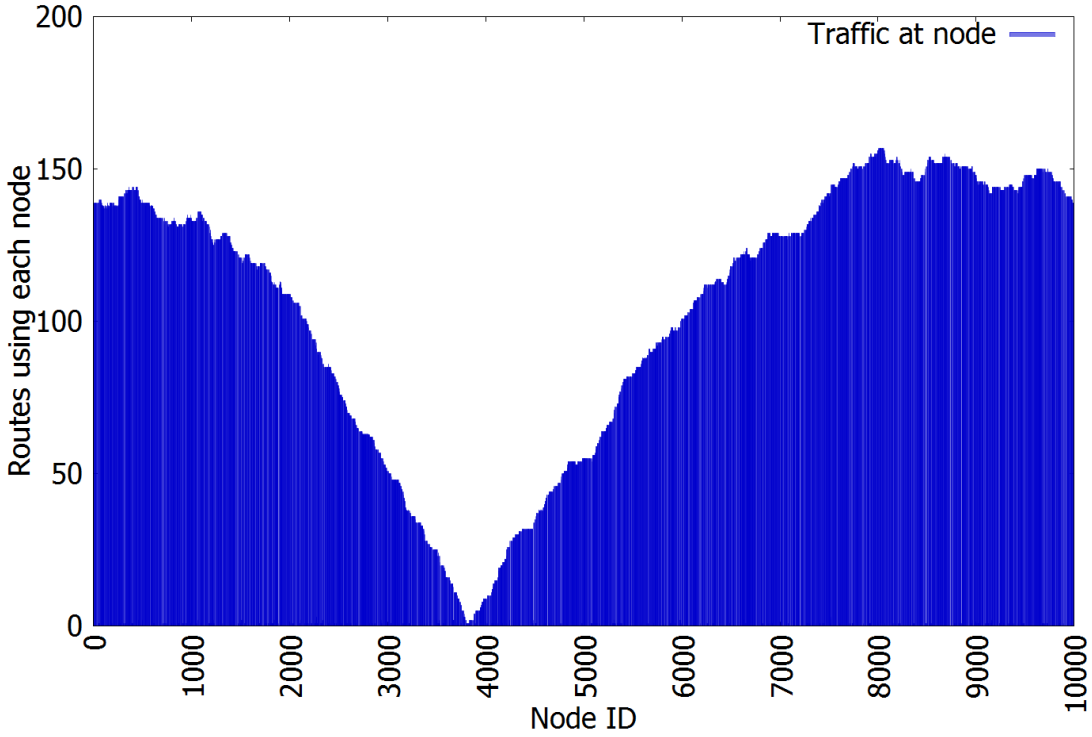

Figure 8: Regular ring lattice before partitioning.
Shows how many data packets were routed through each node in the topology, out of 300. This figure depicts the topology after failures have occurred. Notice MST paths have changed, but the overall load is roughly the same.

Small-world networks were investigated next. Figure 9 depicts a small-world network before any failures have taken place. Compared with the regular ring lattice, the influence of a shorter path length on throughput when using an MST for routing is clear. The small-world network had much fewer nodes processing a high percentage of messages. Instead, the vast majority of nodes were tasked with forwarding less than 10% of the packets, while some clusters of nodes reported high usage. The reasoning for the appearance of theses clusters lies in the small-world network's high clustering coefficient. The high clustering coefficient at a given node means that the MST likely will include a link to a neighboring node, resulting in a chain through the neighboring nodes in the cluster. This chain means that if one node in the cluster has a high usage, the neighboring nodes also have high usage. This is similar to regular ring lattices, but here the MST branches sooner since not all nodes have high clustering. Note that a single

MST is still being used to route between all 300 pairs of randomly selected endpoints, yet the busiest of these clusters did not significantly exceed the busiest of regular ring lattice nodes.

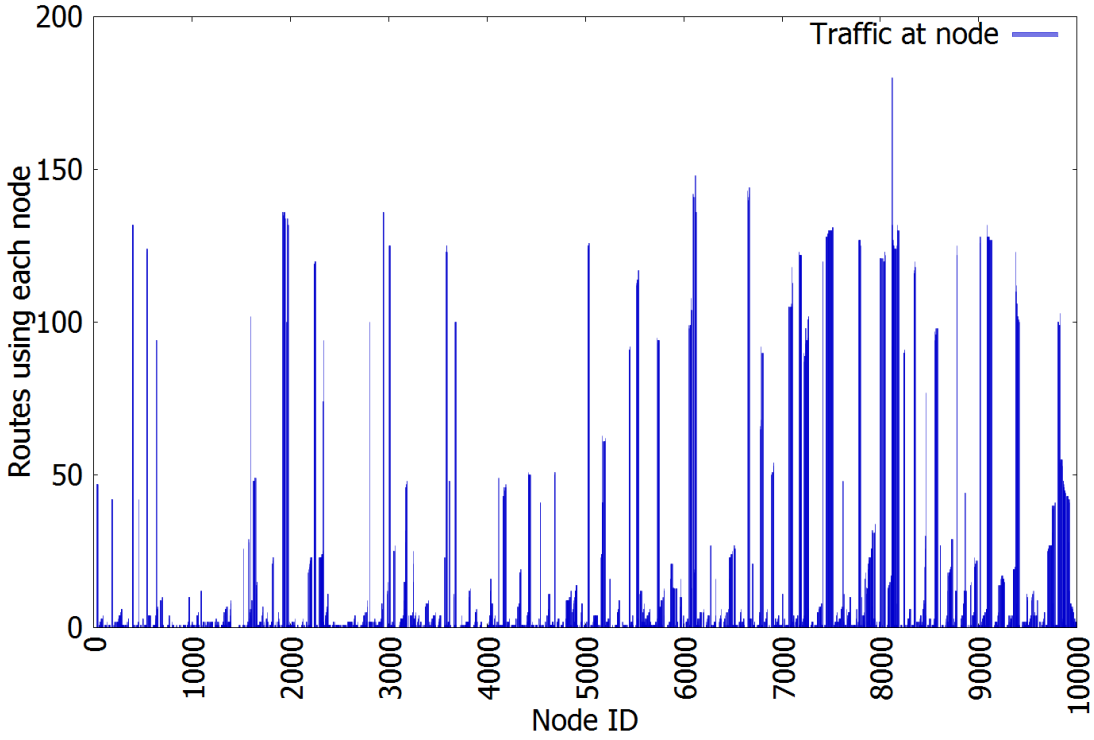

Figure 9: Small world graph without failures.
This figure depicts the bottlenecks in small world graphs when routing with an MST. Notice that there are busy clusters of nodes separated by regions of nodes with light usage.

After nearly 32% of the nodes had failed, the results for throughput right before the graph partitioned were collected, and are shown in Figure 10. These results were surprising because we had expected to see either more pronounced large usage spikes or increased density of medium usage spikes. Instead of finding increases in usage spike density, we found that the load was lighter across the majority of nodes in the system. The plot revealed that the throughput and bottlenecks appeared to improve, with much fewer busy nodes.

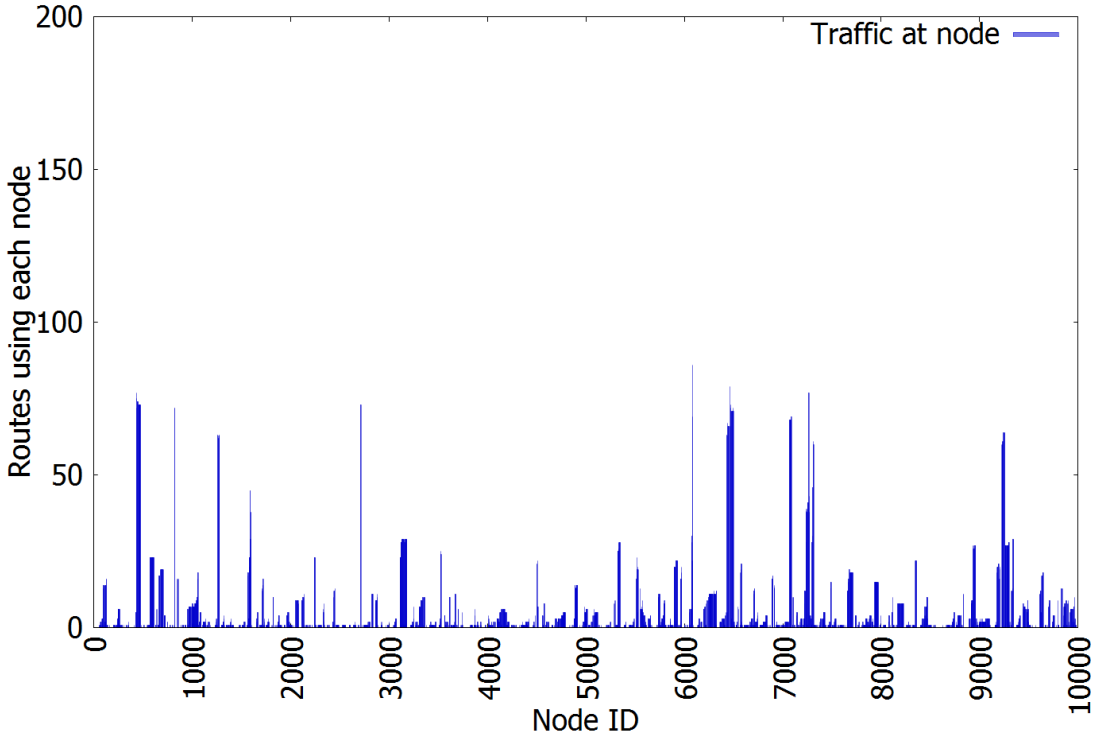

Figure 10: Small world before partitioning.
This figure depicts the small world topology after failures have taken place. The busy clusters still exist, but have migrated. Over 3000 nodes are no longer reporting usage.

This figure shows two high usage nodes forwarding over half of the messages, but it is not busier than the nodes in Figure 9. There are a few explanations for this behavior. Over 30% of the nodes have failed by the time the data for Figure 10 is recorded. This means that over 30% of 10,000 nodes are no longer reporting any usage, since data packets no longer travel over them. To ensure that this is not a visual anomaly, the actual number of nodes with high usage above 100 along with the number of nodes with medium usage between 30 and 100 is determined both before and after failures, averaged over 100 runs. Before failures, 318 nodes had medium usage and 173 nodes had high usage. After failures, 151 had medium usage and 34 nodes reported high usage. This is half of the medium usage nodes and most of the high usage nodes. Nodes seeing at least 50% of the packets dropped from 29 to 1. This is especially significant in terms of bottlenecks considering that the maximum usage seen at a single node dropped on average from 182 packets to 87 packets. Failures appear to have a significant, yet positive, impact on the throughput and bottlenecks seen in the system.

A likely reason for the lower usage levels relates to the usage of an MST for routing. As mentioned previously, the MST is not the shortest path between two nodes. The MST may take a long roundabout path if that path has the minimum total distance to reach all nodes. Using the MST to route between two endpoints means that the roundabout path must be taken, even if the nodes happened to have a direct connection to each other. Once nodes have failed, the MST no longer needs to include a path to them. This means that there can be a more direct path between the remaining nodes. When over 30% of the nodes have failed, especially ones that are part of a cluster, the effect on the MST path becomes quite significant. This is especially true if the MST is allowed to branch at the hubs following the removal of desirable links, producing paths closer to the shortest path possible in terms of hops. It would be interesting to use a shortest path routing algorithm alongside of the MST routing algorithm to gauge how much of an impact using an MST has on node usage. From what we have gathered from this experiment, it appears that the load reduces slightly, and is shifted to other nodes in the system, at least when routing with an MST.

After the throughput of small-world networks were investigated, the throughput of random graphs was examined. These results, shown in Figure 11, shared some similarities with those of the small world network, but were more in line with what we expected to see. Over 100 runs of the simulation, there were fewer nodes with the high and medium usage levels seen with small-world graphs. Before any failures, there were 204 nodes with medium usage and 108 nodes with high usage. 16 nodes reported receiving at least 50% of the packets, and the maximum usage was 181 of 300 packets.

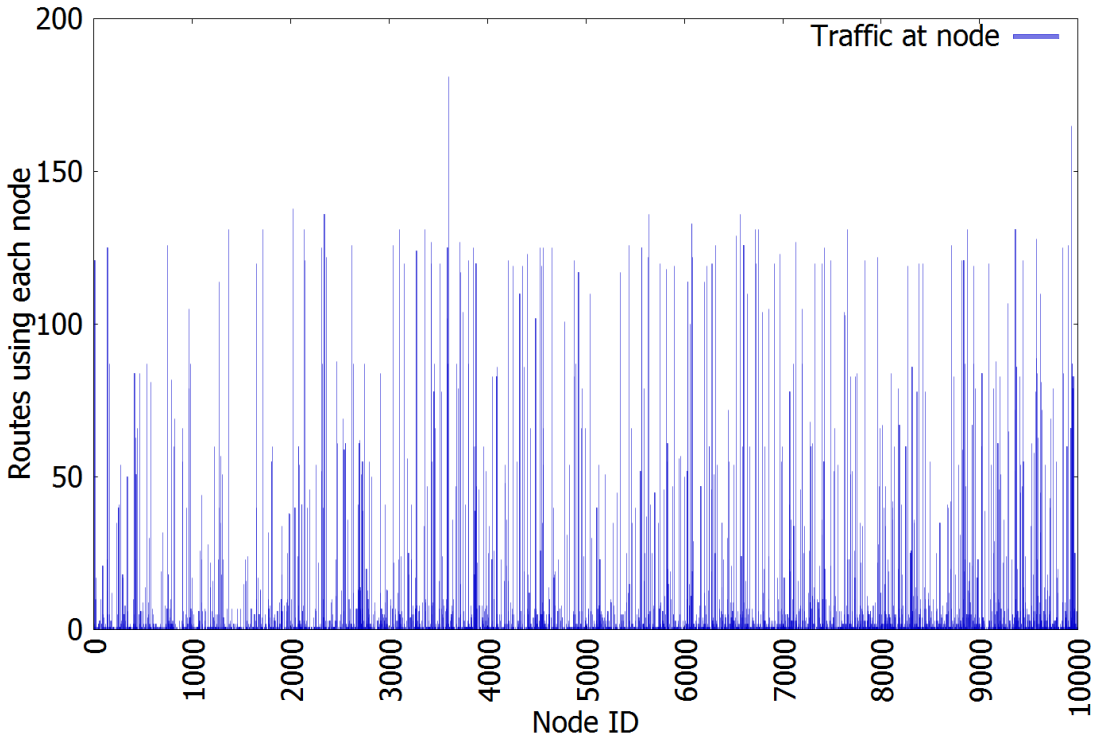

Figure 11: Random graph before failures.
Depicts the busy nodes in a random graph before any failures. There are roughly the same number of busy nodes as in small world, but they are spread evenly across the topology.

Once failures were introduced into the random graph, shown in Figure 12, the number of nodes with medium and high levels of usages remained relatively the same, but had potential to increase. After failures, 189 nodes had medium usage and 103 were high usage. The number of nodes forwarding over 50% of the packets and the maximum usage reported remained the same. When it comes to throughput and bottlenecks when faced with random failures, the random graphs absorb the failures well, but do not improve with failures as the small-world topologies do. As a tradeoff, the random graphs start with roughly 200 fewer medium and high usage nodes than small-world networks. In addition, the busy nodes appear to be better spaced in the random graph topology than the small-world topology, a trait related to the clustering coefficients for each topology. As mentioned previously, small-world topologies have a high clustering coefficient close to that of a regular ring lattice. Random graphs, on the other hand, have a very low clustering coefficient. This means that random graphs have busy nodes spread randomly across the system, while the small-world topologies form busy clusters of nodes.

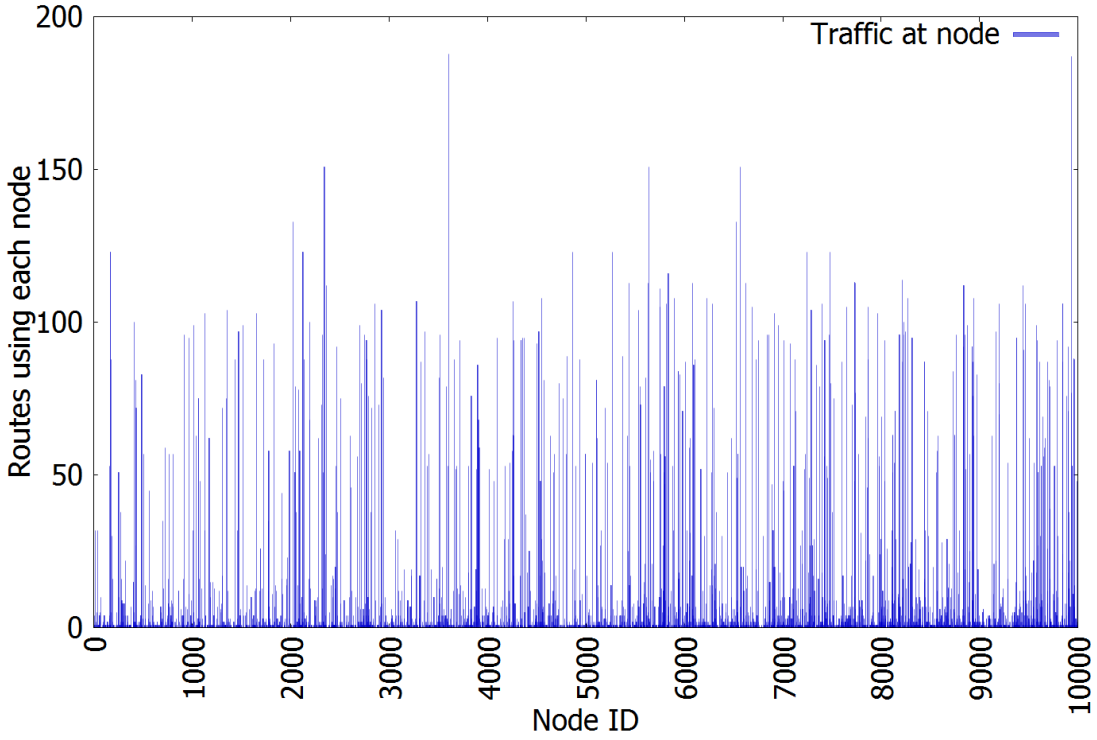

Figure 12: Random graph before partitioning.
Depicts the busy nodes in a random graph. There are roughly the same number of busy nodes as in small world, but they are spread evenly across the topology.

This experiment did not look into which of the topologies were better at dynamically adapting to changes in load. It may be that when using an MST to route large amounts of data, randomly spaced busy nodes are better at adapting their local routing policy. This could be accomplished by adjusting link weights in response to the load. One the other hand, with a local cluster of busy nodes all choking on the same data, it may be easier to make local decisions that lessen the load experienced by some nodes in the cluster. Either way, the trade off with fault tolerance needs to be taken into account as well. The random graph topology with 10,000 nodes performed poorly compared to the small-world network. In general, the small-world network was able to sustain twice the amount of failures.

If the only requirement for the system is bottlenecks and throughput when routing data with an MST, power-law networks show promising results. Figure 13 depicts the results of routing between 300 pairs of random endpoints using an MST. Unlike regular ring lattices, regular graphs, and small world networks, the nodes are much less busy. Even the few hubs, depicted as spikes in the figure, are less busy

in power law networks than a larger portion of nodes small-world and random graphs. Recall the discussion earlier about how MSTs are not the shortest path between two endpoints. Since links between smaller groups of nodes exist almost exclusively via hubs, the MST will contain these links. This means that the MST will branch at each hub toward the leaf nodes. As a result, the resulting MST contains paths between pairs of nodes that are very short, if not the shortest path available in the system. In small-world topologies, the MST contains longer branches, with chains of nodes shared between larger portions of endpoints. In contrast, MSTs for power-law topologies have shorter, more direct branches. Chains of nodes are shared between fewer pairs of endpoints.

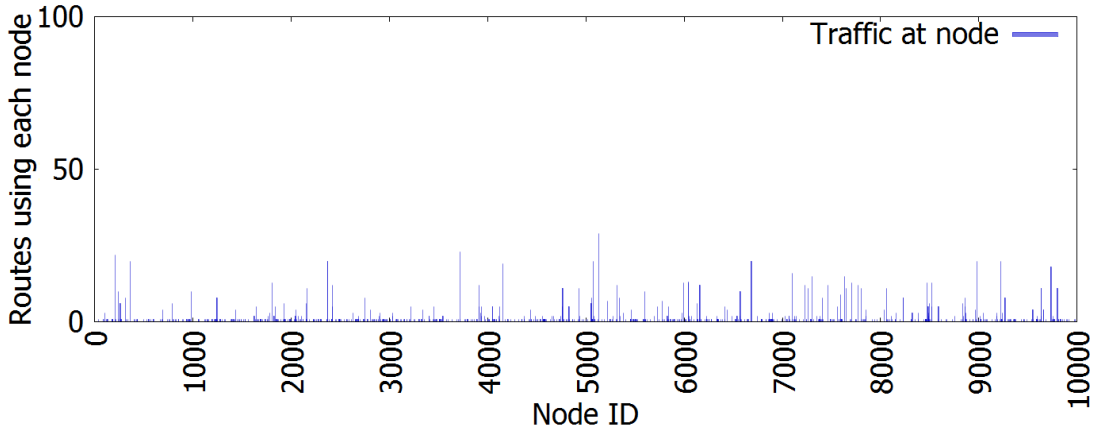

Figure 13: Power-law graph before failures.
Depicts the number of data packets routed though each node in a power law topology. The MST branches at the hubs, spikes in the figure, resulting in path lengths close to the shortest path.

As appealing as power law networks are when using an MST, the topology is not suitable for handling failures. The failure of a single hub node can partition the system, which is not desirable. As mentioned previously, the topology partitioned before even one percent of the nodes failed. With respect to the goals of this thesis, the poor failure resilience of power-law topologies make them unacceptable for consideration in real systems.

CHAPTER 5

CONCLUSION

We explored the role of topology in coping with failures during content disseminations. Our benchmarks tested topologies comprising up to 10,000 nodes. Features of the topology such as the degree, randomness, and locality of connections contribute to this resilience and also the distribution of loads as a result of these failures. In general, average path lengths do not have a correlation to the resilience of the topology. The R value of the topology plays a crucial role in failure resilience: an R value between 0.1 and 0.2 outperforms all other values of regardless of the K value within that topology.

Small-world networks handle failures particularly well. They can sustain a high percentage of node failures before the topology partitions. The load distributions as a result of failures also do not introduce bottlenecks. When using an MST for routing, the path lengths and average workload at each node improve with failures.

Random topologies handle throughput and bottlenecks in the presence of failures reasonably well, though not to the same degree as small-world topologies do.

Regular ring lattices and power-law networks performed poorly in the presence of failures with partitions forming rather fast. Though power-law networks show promising results when routing data, their poor failure resilience makes them unacceptable in real settings.

Our future efforts will be focused on dynamic reconfigurations of the underlying topologies. The reconfigurations will be triggered to preserve performance in the presence of bottlenecks and also in the face of node and link failures.

REFERENCES

[1] Eugster, P. T., Felber, P. A., Guerraoui, R., and Kermarrec, A. 2003. The many faces of publish/subscribe. *ACM Computing Surveys (CSUR)*, 35, 114-131.

[2] Pallickara, S., Ekanayake, J., and Fox, G. 2009. Granules: A lightweight, streaming runtime for cloud computing with support, for map-reduce. In *IEEE International Conference on Cluster Computing (CLUSTER'09)*, 1-10.

[3] Pallickara, S. and Fox, G. 2003. NaradaBrokering: a distributed middleware framework and architecture for enabling durable peer-to-peer grids. In *Middleware 2003*, 998-999.

[4] Rowstron, A. and Druschel, P. 2001. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Middleware 2001*, 329-350.

[5] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., and Balakrishnan, H. 2001. Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM Computer Communication Review*, 31, 4, 149-160.

[6] Cohen, B. 2003. Incentives build robustness in BitTorrent. In *Workshop on Economics of Peer-to-Peer systems*, 68-72.

[7] Abolhasan, M., Wysocki, T., and Dutkiewicz, E. 2004. A review of routing protocols for mobile ad hoc networks. *Ad hoc networks*, 2, 1, 1-22.

[8] Baldoni, R., Beraldi, R., Cugola, G., Migliavacca, M., and Querzoni, L. 2005. Structure-less content-based routing in mobile ad hoc networks. In *IEEE International Conference on Pervasive Services, 2005 (ICPS'05)*, 37-46.

[9] Opyrchal, L., Astley, M., Auerbach, J., Banavar, G., Strom, R., and Sturman, D. 2000. Exploiting IP multicast in content-based publish-subscribe systems. In *IFIP/ACM International Conference on Distributed systems platforms*, 185-207.

[10] Quinn, B. and Almeroth, K. 2001. IP multicast applications: Challenges and solutions.

[11] Ripeanu, M. and Foster, I. 2002. Mapping the gnutella network: Macroscopic properties of large-scale peer-to-peer systems. *Peer-to-Peer Systems*, 85-93.

[12] Watts, D. and Strogatz, S. 1998. The small world problem. *Collective Dynamics of Small-World Networks*, 393, 440-442.

[13] Faloutsos, M., Faloutsos, P., and Faloutsos, C. 1999. On power-law relationships of the internet topology. In *ACM SIGCOMM Computer Communication Review*, 251-262.

[14] Barabasi, A., Albert, R., and Jeong, H. 2000. Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A: Statistical Mechanics and its Applications*, 281, 1, 69-77.

[15] Kruskal, J. B. 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7, 1, 48-50.