Study and Improvement of Robustness of Overlay Networks

A thesis submitted in partial fulfillment of
the requirements for the degree of

Master of Technology

by

Hema Swetha Koppula

Under the guidance of

Prof. Niloy Ganguly

" YOGA KARMASU
KAUSALAM "

Department of Computer Science & Engineering,
Indian Institute of Technology – Kharagpur

2008

Certificate

This is to certify that the thesis titled Study and Improvement of Robustness of Overlay Networks submitted by Hema Swetha Koppula (03CS3016) to the Department of Computer Science and Engineering in partial fulfillment for the award of the degree of Bachelor of Technology (Hons.) & Master in Technology in Computer Science and Engineering, is a bonafide record of work carried out under my supervision and guidance. The thesis has fulfilled all the requirements as per as regulation of this institute and, in my opinion reached the standard for submission.

Date: May 7th, 2008

Prof. Niloy Ganguly

Assistant Professor
Department of CSE
IIT Kharagpur

Acknowledgement

With great pleasure and deep sense of gratitude, I express my indebtedness to Prof. Niloy Ganguly for his invaluable guidance and constant encouragement at each and every step of my project work. He exposed us to the intricacies of relevant topics through paper counseling and discussions and always showed great interest in providing timely support and suitable suggestions.

I would also like to express my gratitude to all my friends and colleagues for their constant support and encouragement. Words are not enough to express my gratitude towards my parents to whom I owe every success and achievements of my life. Their constant support and encouragement under all odds has brought me where I stand today.

Date: May 7th, 2008

Hema Swetha Koppula

03CS3016
Department of CSE
IIT Kharagpur

Abstract

The heterogeneity present in the real-world networks like peer-to-peer networks make them particularly vulnerable to attacks as large-scale cascade may be triggered by disabling a set of key nodes. In addition to this vulnerability towards dynamic events, real world networks react quite strongly towards certain types of attacks which may adversely affect their static properties. This brings an obvious concern for the security and robustness of these systems. In this thesis, empirical results are presented that show how robustness of overlay networks, measured in terms of different parameters like size of largest connected component, number of components and diameter, percolation point and number of nodes failed, can be improved by applying various edge modification schemes. The dynamic effect of node removal along with its static impact on the network is observed in order to study the impact network topology has on its robustness. Also by assuming simple models of communication between the network nodes, the impact that the routing schemes have on the robustness of the network given its topology is studied.

Contents

Abstract iii

List of Figures vi

List of Tables viii

1. Introduction…..……………………………….....………………………1

2. Background and Related Work…………………………………………………3

3. Network Modification: Simulation Model…………………………………..….5

3.1. Edge Modification Schemes……………………………………………........5

3.2. Metrics to calculate Robustness……………………………………..………..6

3.3. Simulation Methodology…………………………………...………………9

3.3.1. Network Graph……………………………………………………..…...10

3.3.2. Edge Modification Model…..…………….………………………….11

3.3.3. Attack Model……………………..…………………………..11

3.3.4. Cascaded Failure Model…………………………………..……….12

4. Network Modification: Results & Analysis…………………………………13

4.1. Static Analysis……………………………………………………………13

4.2. Dynamic Analysis……………..……………………………………….16

5. Effect of Routing Strategies……………………………..…………………...22

5.1. Introduction……………………………………..……………......22

5.2. Routing Model…………………………………………………………23

5.3. Random Walks…………………………………………………..……..25

5.4. Cascading Effect………………………………………………………..…...30

5.4.1. Random Walker 1……..……………………………………….………..30

5.4.2. Random Walker 2……………………..…………………….………..32

5.4.3. Comparison with results on power-law graph……………….……….35

5.5. Routing Analysis ………………………………………..………..….36

6. Conclusion…………………………………………………….………..……….41

Bibliography………………………………………………………………........... 48

List of Figures

3.1 : The Component Size Distribution for Gnutella Network………………………8

3.2 : Simulation Flow Chart …………………………………………………………9

4.1 : The effect of modification schemes on the Diameter ………………………...14

4.2 : The effect of modification schemes on the Size of the LCC…………………..14

4.3 : The effect of modification schemes on the Number of Components………….15

4.4 : The effect of modification schemes on Percolation Point……………………..16

4.5 : Node Failure due to Cascading Effect for different types of Attacks…………16

4.6 : Node Failure due to Cascading Effect as a function of the % of Addition…....18

5.1 : Effect of No. of Communicating Pairs on the cascading effect.…………….……24

5.2 : Random Betweenness vs degree of nodes for a power law graph…….….…. 27

5.3 : Random Betweenness vs Random Walker1 Betweenness……….……….…...29

5.3 : Random Betweenness vs Random Walker2 Betweenness ……….…………...29

5.5 : Cascading effect - number of nodes failed for Random Walker1 ……………. 31

5.6 : Cascading effect – number of components for Random Walker1………….......31

5.7 : Cascading effect - size of LCC for Random Walker1……………….................32

5.8 : Cascading effect - number of nodes failed for Random Walker 2……….……..33

5.9 : Cascading effect - number of components for Random Walker 2……….……..34

5.10 : Cascading effect - number of components for Random Walker2…………..….34

5.11 : Number of nodes failed - Random Walker1 on power-law graph ……….…..35

5.12 : Number of nodes failed - Random Walker2 on power-law graph …..…….…36

5.13 : Total degree of nodes removed – Random Walker1…………………………..37

5.14 : Total degree of nodes removed – Random Walker2………………………….37

5.15 : Probability of choosing a node of degree k - Random Walker1 …………….39

5.16 : Probability of choosing a node of degree k - Random Walker2 …….……..... 40

List of Tables

Table 1: Results of Edge Addition Schemes on Gnutella Network ….……………...13

Table 2: Results of Edge Rewiring Schemes on Gnutella Network …..……………..13

Table 3: Cascading effect on Removal of Highest Degree Node of the Network…...17

Table 4: Cascading Effect with Edge Addition Schemes on Gnutella Network….....18

Table 5: Cascading Effect with Rewiring Schemes on Gnutella Network…………..19

Chapter 1

Introduction

The study of attacks on complex networks is important in order to identify the vulnerabilities of real-world networks, which can be used either for protection (e.g., of infrastructures) or for destruction (e.g., in the control of epidemic diseases). Additionally, it can provide guidance in designing more robust artificial networks (e.g., communication networks). An important property of networked systems is their robustness against various types of failures and attacks on network nodes. Although several design methods have been proposed for creating a network that has optimal robustness according to a given measure, in most real world situations we are often faced with an existing network that cannot be substantially modified or redesigned. Moreover, real world networks are result of many different processes that may not take the robustness into account. For example we can consider the peer-to-peer networks, which are largely decentralized and highly dynamic systems. One cannot have explicit control over their structure to ensure properties like robustness under various types of disrupting events such as a random failure or an intended attack. The robustness of such networks can be improved by a small degree of modification [1].

The modification could be in the form of either edge addition or edge rewiring. The network can be modified at two different stages to increase the robustness. One is a preventive stage in which the network is made more robust so that it does not breakdown under attack or failure. The second stage is after a disrupting event, by applying some repair strategies to restore the original properties of the network. For applying any kind of edge modification to a network to improve its robustness, it is important to understand how the existing topologies deal with failures and attacks. In this work, the effect of random failure and targeted attack on network nodes in a particular peer-to-peer overlay network, a crawl of Gnutella super-peer network, has been studied. Both static and dynamic effects of the node removal have been considered to see if, by suitably modifying the network we can improve it robustness against failures and attacks without appreciably degrading its performance.

Two types of cost are incurred by modifying a network. The first type of cost is that associated with adding or rewiring edges, i.e. modifying the network. Second is the cost incurred due to the changes in particular properties, which the network is designed to have, due to the application of various modification schemes on the network. These costs are application-specific. The modification schemes are evaluated based on the cost, along with the improvement in the robustness measures.

The propagation of the node failure in the network depends both on the network structure as well as the routing strategy followed to pass messages in the network. Different routing strategies choose different intermediate nodes to route messages between the same end nodes. This leads to congestion at different nodes and hence causes their failure. Therefore, the routing strategies also affect the cascading effect of node removals in the network. By choosing an appropriate routing strategy the network can be made more stable without making any change in the network topology. This method would not have costs due to modification of the network, but might lead to the increase in communication costs based on the routing strategy. To understand the effect of the routing strategies on the robustness of the network, a few routing models have been simulated on the network and the cascading effect was observed.

The remainder of this thesis is organized as follows. Chapter 2 provides background and related work on various studies on robustness of complex graphs. Chapter 3 describes our edge modification schemes and the metrics used to measure robustness, and Chapter 4 describes the simulation methodology. Chapter 5 discusses implications of the study of robustness as a measure of modification. Chapter 6 studied the effect of routing strategy has on robustness and Chapter 6 concludes this thesis.

Chapter 2

Background and Related Work

Many authors have studied the effect of failures and attacks on various complex networks. Scale-free networks are known to be sensitive to targeted attacks, which are biased towards higher degree, in comparison with random attacks[2]. This is due to the heterogeneity present in the scale free networks. In these networks, degree distribution i.e. probability of a node having degree **k**, decreases with power of **k**[3]. Therefore randomly chosen node is likely to have a low degree, so its removal has little effect on the network. Removal of a high degree node can have a significant effect since such a node may hold a large part of the network together by connecting many other nodes. For Erdos-Renyi random graphs[6], there is not much difference between random failures and targeted attacks due to the homogeneous nature of these networks. In these graphs every pair of nodes is connected with a fixed probability **p**, independently of every other pair. They have a binomial degree distribution, $\mathbf{P_b(k)}$, which approaches a Poisson distribution as the number of nodes becomes large. Hence, there is very less chance of encountering a hub. Therefore, targeted attacks have less effect on these graphs. It is found that these networks are more vulnerable to *random failures* than to *intended attacks*, compared to scale-free networks.

A convenient way to address the robustness of a complex network is to examine how the diameter, size of the largest connected component and number of connected components, which measure the efficiency of communication (or information flow) within the network, are changing under random or intentional attacks. But these measures address only the static properties of the networks. Cascading failures have been reported for numerous networks, which refer to subsequent failure of other parts of the network induced by the failure of or attacks on few key nodes. Researchers have investigated mechanisms leading to cascades of overload failures in complex networks by constructing models incorporating the flow of physical quantities in the network [4]. An important question for many real-world situations is how attacks affect the functioning of a network when the flow of information or other physical quantities in the network are taken into consideration. In particular, the removal of nodes changes the balance of flows and it may trigger a cascading failure, as the one that happened on August 10, 1996 in the western U.S. power grid. Authors have shown that for networks where network flow can redistribute among the nodes, intentional attacks on highly loaded nodes can trigger a large-scale cascade of overload failures[7].

Chapter 3

Network Modification: *Simulation Model*

The various schemes which are used to increase robustness of networks are discussed here. In addition to that some simple measures which can quantify the robustness of any network are also discussed.

3.1. Edge Modification Schemes

Various edge modification schemes have been proposed in the literature, which aim at improving the robustness of these complex networks [1]. These can be broadly categorized into - Edge Addition schemes and Edge Rewiring schemes. Edge addition schemes result in increased number of edges or connectivity in the network whereas rewiring schemes change the properties of the network while keeping the number of edges constant.

The following modification schemes are considered: (Note that 'Random' as used here means randomly chosen with uniform probability and duplication of edges between any two already connected nodes is not allowed)

1. Random Edge Addition - An edge is added between any two randomly chosen nodes.

2. Preferential Edge Addition - An edge is added between two unconnected nodes having the lowest degrees in the network.

3. Random Edge Rewiring - A random edge is removed and then a random edge is added between two random nodes.

4. Random Neighbor Rewiring - A node is chosen at random and an edge to a random neighbor is disconnected from that node. The loose end of this edge is connected to a random node.

The *Random neighbor rewiring* is a new edge modification scheme that we have introduced. It is a variation from the previously stated Random neighbor rewiring schemes [1].

If we choose a random neighbor of a randomly chosen node, the probability of the neighbor node having degree $\mathbf{k}$ is proportional to $\mathbf{kp_k}$, where $\mathbf{p_k}$ is the probability that the randomly chosen node has degree $\mathbf{k}$. Therefore the random neighbors of randomly chosen nodes have higher degree, given that the assortativity is low. In such cases, where assortativity is low, the *Random neighbor rewiring* scheme disconnects the edge connected to a high degree neighbor and reconnects it to a random node, which would be a lower degree node given the power law nature of the scale-free graphs. This tends to bring in a degree of homogeneity into the graph structure, the extent of which depends on the amount of rewiring.

These edge modification schemes can be mapped to different network management processes that take place in unstructured peer-to-peer overlay networks. For example, the superpeers connect to new superpeers which come into the network and disconnect old superpeers with time, in order to exchange network information, as well as to handle the network churn. This process is equivalent to random rewiring if no preference is used in choosing new neighbors. Therefore, studying the effect of these modification schemes on the robustness of the overlay network can help in designing robust network management protocols.

3.2. Metrics to calculate Robustness

We measure the robustness of the networks on the basis of following parameters:

1. Diameter of the graph

2. Size of the largest connected component (LCC)

3. Number of components
4. Percolation Point
5. Node Failure

The first three parameters are static measures of robustness of the network, i.e. they do not capture the effect of cascading of the network flow upon a failure or an attack. These three metrics were chosen as they are simple and also capture the essential requirements for a robust network without flow considerations. The *diameter* of graph is the maximum of all longest paths between any two nodes in the network. *Size of largest connected component* is the maximum number of nodes present in a component among all the components present. *Number of components* is the number of weakly connected components present in the network. These metrics quantify robustness of a network as the *diameter* is a measure of the maximum time for information propagation in the network, whereas the *size of LCC* and *number of components* measure the availability of the network. A network is considered more robust if it has a low diameter and high availability. Therefore any increase in diameter, or an increase in the number of components, or a decrease in the size of LCC can be considered as degradation of the network and hence reflects on the low robustness of the network graph. The last metric *Node Failure* measures the dynamics of node removals. It shows how many nodes go down due to the overload of flow in the network caused by the previously removed nodes. It is a measure of the cascading effect created due to removing any set of nodes from the network and the breakdown in the information flow caused by it. We show that networks where load can be redistributed among the remaining nodes, targeted attacks on key nodes can lead to breakdown of the whole network.

The stability of superpeer networks are measured in terms of a certain fraction of nodes (f_c) called *percolation threshold* [8], removal of which disintegrates the network into large number of small, disconnected components. Below that threshold, there exists a connected component which spans the entire network. The value of the percolation threshold f_c theoretically signifies the stability of the network; higher values indicate greater stability against attack. During the experiment, we remove a fraction of nodes f_t from the network in step t and check whether we reach the percolation point. If not then in the next step t + 1 we remove $f_{t+1} = f_t + \epsilon$ fraction of nodes from the network and check again. This process is continued until we reach the percolation point. After each step, we find out the status of the network in terms of the number and size of the components formed. We collect the statistics of s and n_s where s denotes size of the components and n_s, number of components of size s and define the normalized component size distribution $CS_t(s) = sn_s / \Sigma_s sn_s$ at step t. We compute $CS_t(s)$ for all the steps starting from t = 1 and observe the behavior of $CS_t(s)$ after each step. Initially the CSt(s) shows unimodal character confirming a single connected component or bimodal character confirming a large component along with a set of small components. As the fraction of nodes removed from the network increases gradually, the network disintegrates into several components. This leads to the change in the behavior of $CS_t(s)$ whereby at a particular step, t_n, $CS_{tn}(s)$ becomes monotonically decreasing function indicating t_n as the percolation point. Therefore t_n is considered as the time step where percolation occurs and the total fraction of nodes removed at that step f_{tn} specifies the percolation threshold. Figure 3.1 shows the graphs of the component size distributions at various time instances taking ϵ value as 0.01. It can be seen that initially there was one big connected component. Later at time t = 10, there is a bimodal distribution of the component sizes signifying presence of large components spanning the network. At time t = 31, the network completely disintegrates into very small components. This corresponds to the percolation point of the network.

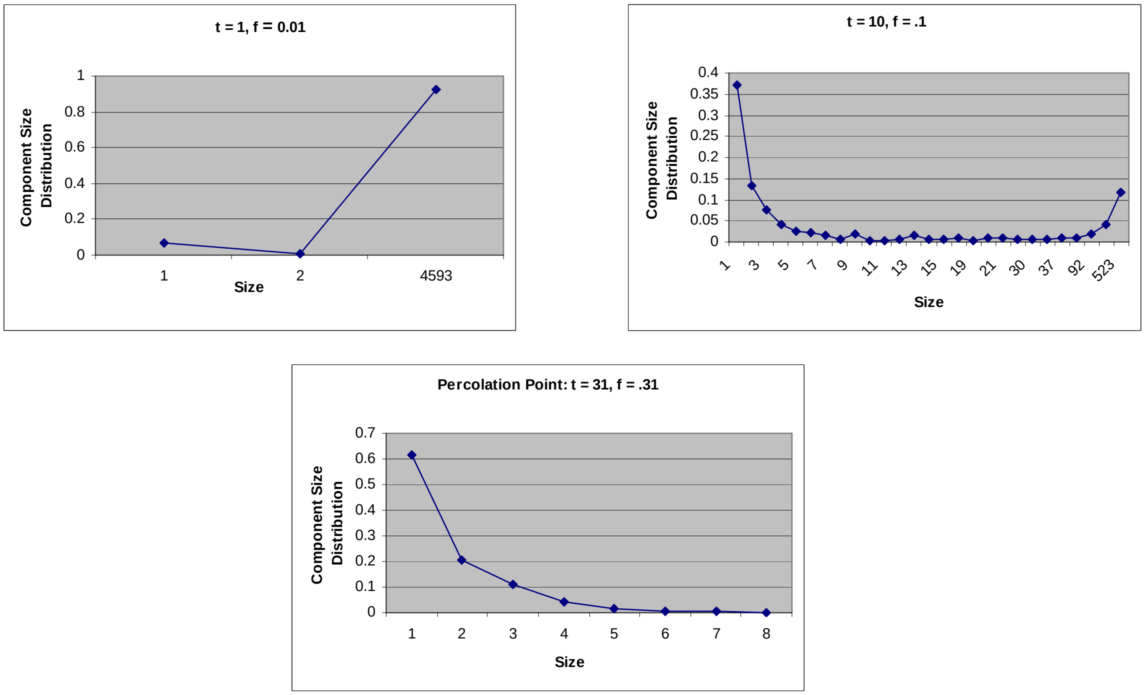


Figure 3.2. : The Component Size Distribution for Gnutella Network for Є = .01, showing the occurrence of percolation point at t = 31.

The various edge modification schemes are studied under the light of how they affect these metrics which are computed as a function of percentage modification for a given percentage of removed nodes. These metrics give us insight into making the network more robust against attack on nodes by taking proper preventive measures.

3.3. Simulation Methodology

The simulations are mainly concentrated around the preventive measures introduced in the first section of this thesis. The effect of failure and attack on the original graph and to understand the nature of the network topology is studied first. Then the various edge modification schemes are simulated on the network graph to get modified graphs. The effect of attacks and failures are observed on these modified graphs and the results are compared for the various modification schemes. The methodology is described in the flowchart given in Figure 4.1. The network graph, modification and attack analysis models are described here.

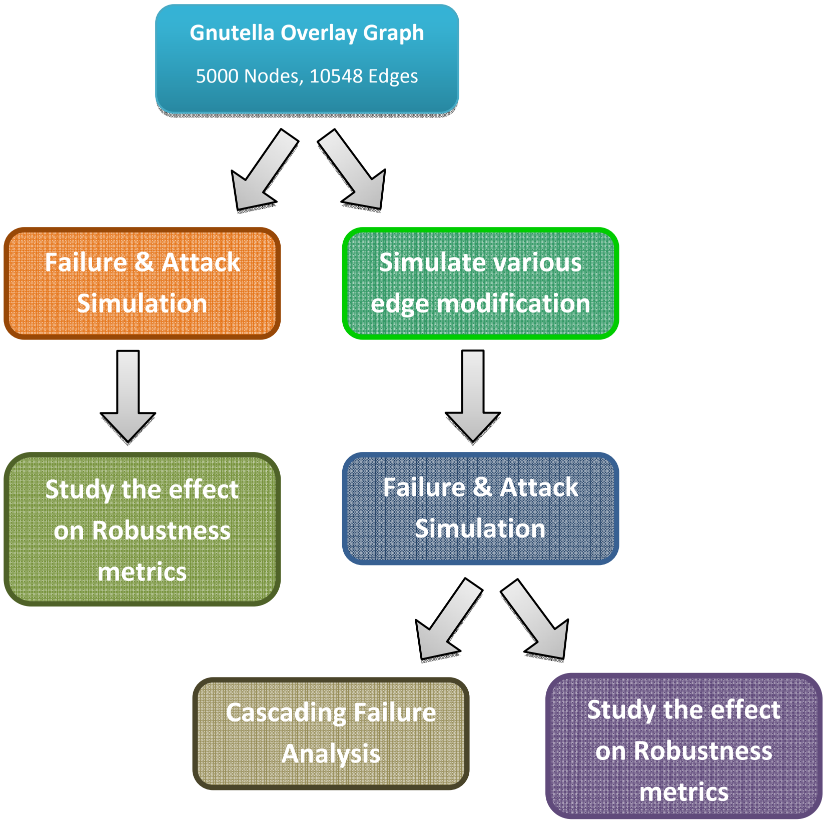

Gnutella Overlay Graph
5000 Nodes, 10548 Edges
Failure & Attack Simulation
Simulate various edge modification
Study the effect on Robustness metrics
Failure & Attack Simulation
Cascading Failure Analysis
Study the effect on Robustness metrics

Figure 3.2 : Simulation Flow Chart: Showing the steps executed for studying the effect of various modification schemes and node failures on the graph

3.3.1.Network Graph

Attack and edge modification schemes were simulated and their effects upon the peer-to-peer overlay networks are studied. The simulations were performed on the overlay network of size 5000 nodes, obtained by crawling Gnutella. The original network contained more than a million nodes but we selected a connected subset of the original graph for simulation purpose, since the computation of certain metrics is very costly. This subgraph has a heterogeneous degree distribution but does not follow power law. Its an hybrid between ER and Power Law graphs. Even though real world networks follow power law and are scale free in nature when the graph is considered as a whole, subgraphs of these networks might not posses these characteristics fully. But they surely have a certain degree of heterogeneity as they are random subgraphs of huge heterogeneous graphs. Since one of the motivations behind the study of the various edge modification schemes is to help in designing robust network management protocols, and since these protocols are most effective when based on local knowledge, it justifies studying the robustness and the effect of the edge modification schemes on random subgraphs of the full network.

3.3.2.Edge Modification Model

The edge modification schemes used are *random edge addition*, *preferential edge addition*, *random edge rewiring* and *random neighbor rewiring* as explained in the previous section. First two modification schemes add edges between two nodes which didn't have any edge between them in the original graph. The last two modification schemes try to rewire the edges i.e; number of edges in the network essentially remains the same. Edge modification is applied on the original graph at various percentages (5, 10, 15, 20, 30, 50, 70 %) for each of the four schemes mentioned above.

3.3.3.Attack Model

Two types of node removal are studied, *Random Failure* and *Preferential Attack*. In random failure a set of random nodes are removed from the network. In case of preferential attack, a set of nodes with high degree are removed from the network. This type of attacks can be very crucial to the network as the more important nodes, hub-like nodes which contribute more in network management, will be down which might lead to a break down of the network into various partitions. On each of the original as well as the modified graphs, three levels of failure and attack (5, 10, 15 %) are simulated and the values for the above mentioned metrics were observed. Therefore, the effect of the edge modification is studied by seeing how the measured parameters of the network change with the amount of modification for various levels of failure and attack.

3.3.4.Cascaded Failure Model

For studying the cascaded effect of failures, we assume that the number of messages being transmitted through a node is proportional to the betweenness of that node in the network. Also, initially the network is in a stationary state where the load at each node is less than the capacity of that node. Therefore we assign capacities to each node on the basis of its initial betweenness centrality in the network, **$(1 + \alpha)L$**, where L is the initial load (initial betweenness centrality) at each node and **α** is a small positive fraction. For our simulations we used the value **$\alpha = 0.3$**. The load at each node at any time step is computed as a function of total number of shortest paths passing through that node. We have used a modification of dijkstra algorithm for computing betweenness centrality of each node[8]. Then a small percentage of nodes is removed using either the Random Failure model or Targeted Attack model. After attack step, loads of the removed nodes are redistributed in the network which changes the betweenness centralities of the remaining nodes. Then each node is checked to see if the load i.e; the betweenness centrality of that node, has exceeded its capacity or not. If yes, the node is treated as failed and removed from the network. This way the cascading of node failures was simulated for a fixed number of time steps or until the network had become stable again.

Chapter 4

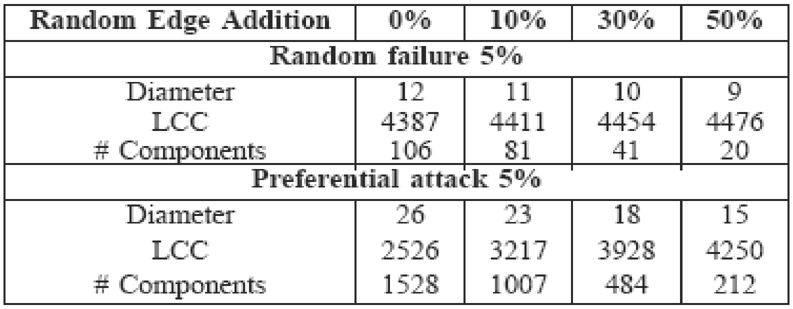
Network Modification: *Results & Analysis*

4.1. Static Analysis

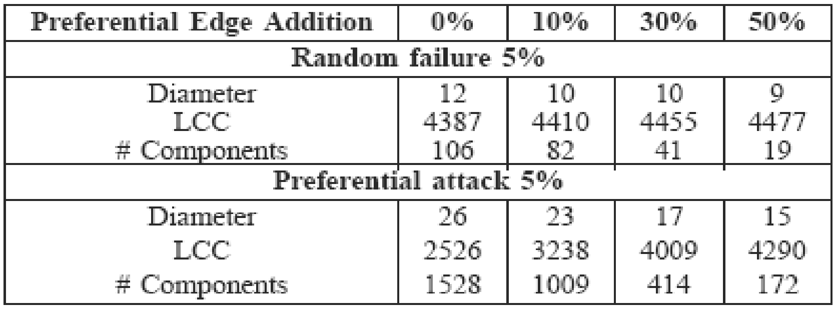
The static analysis deals with the first three metrics mentioned in the last chapter. It considers the static network, i.e. it measures only the network properties after few nodes are removed form the network and doesn't consider the propagation of this failure in the network due to the initial node removal. The results from the static analysis show that both the addition schemes perform better than the rewiring schemes. Addition of new edges increases redundancy in the paths between any two nodes, and hence increases the *size of largest connected component*, while decreasing the *diameter* and the *number of components*. But edge addition is *costly* as it would lead to extra bandwidth usage in the overlay network.

Table 1 and 2 show some of the simulation results for the various schemes. It can be observed that the number of components increase drastically in case of targeted attack as compared to random failure. As we increase the percentage of rewiring, number of components decrease indicating increased connectivity in the network. Similarly, size of largest connected component (LCC) also grows with the percentage of edges rewired.

Table 1: Results of Edge Addition Schemes on Gnutella Network

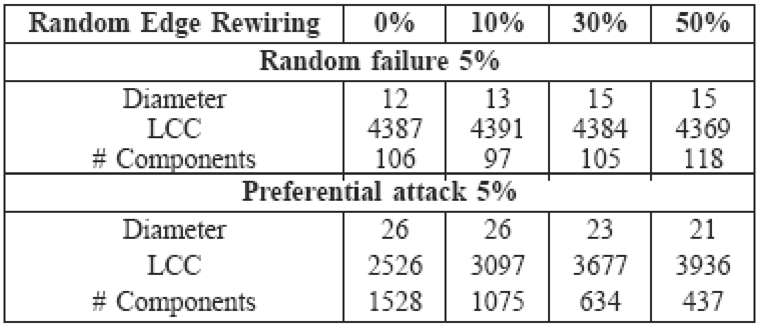


Random Edge Addition	0%	10%	30%	50%
Random failure 5%				
Diameter	12	11	10	9
LCC	4387	4411	4454	4476
# Components	106	81	41	20
Preferential attack 5%				
Diameter	26	23	18	15
LCC	2526	3217	3928	4250
# Components	1528	1007	484	212

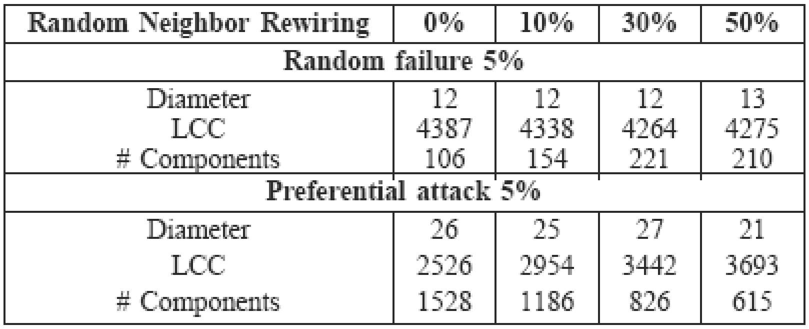


Preferential Edge Addition	0%	10%	30%	50%
Random failure 5%				
Diameter	12	10	10	9
LCC	4387	4410	4455	4477
# Components	106	82	41	19
Preferential attack 5%				
Diameter	26	23	17	15
LCC	2526	3238	4009	4290
# Components	1528	1009	414	172

Table 2: Results of Edge Rewiring Schemes on Gnutella Network



Random Edge Rewiring	0%	10%	30%	50%
Random failure 5%				
Diameter	12	13	15	15
LCC	4387	4391	4384	4369
# Components	106	97	105	118
Preferential attack 5%				
Diameter	26	26	23	21
LCC	2526	3097	3677	3936
# Components	1528	1075	634	437



Random Neighbor Rewiring	0%	10%	30%	50%
Random failure 5%				
Diameter	12	12	12	13
LCC	4387	4338	4264	4275
# Components	106	154	221	210
Preferential attack 5%				
Diameter	26	25	27	21
LCC	2526	2954	3442	3693
# Components	1528	1186	826	615

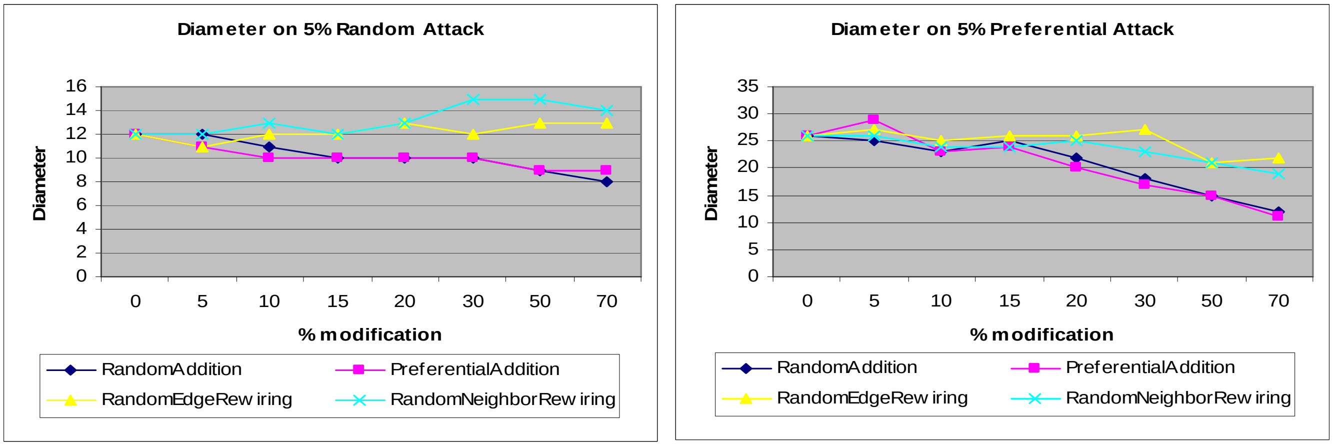


Figure 4.1 : The effect of modification schemes on the Diameter of the network for random and preferential attack.

Figure 4.1 shows how the diameter changes with the percentage of modification for all the four modification schemes. Figure 4.2 shows how the size of the largest connected component changes with percentage modification and Figure 4.3 shows the same results for the number of components. The left side graphs shows the results when 5% of nodes are removed randomly from the network and the right side graphs show for the case when 5% nodes are removed due to preferential attack.

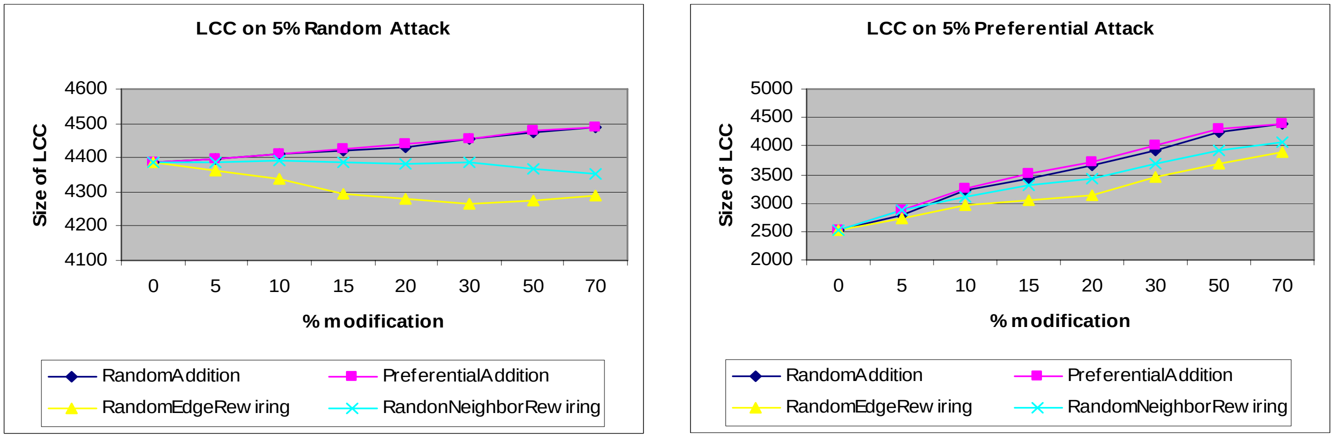


Figure 4.2 : The effect of modification schemes on the Size of the Largest Connected Component of the network for both random and preferential attack.

It can be seen from the results that *Random Neighbor Rewiring* outperforms other schemes in the static analysis of the network, considering the cost of modifications. This can be explained by the assortativity of the network, having an initial value of -0.19, which means that there is low correlation between the degree of neighboring nodes. Hence, as mentioned before the *Random Neighbor Rewiring* tries to make the network more homogeneous and increases the robustness in terms of availability of the network.

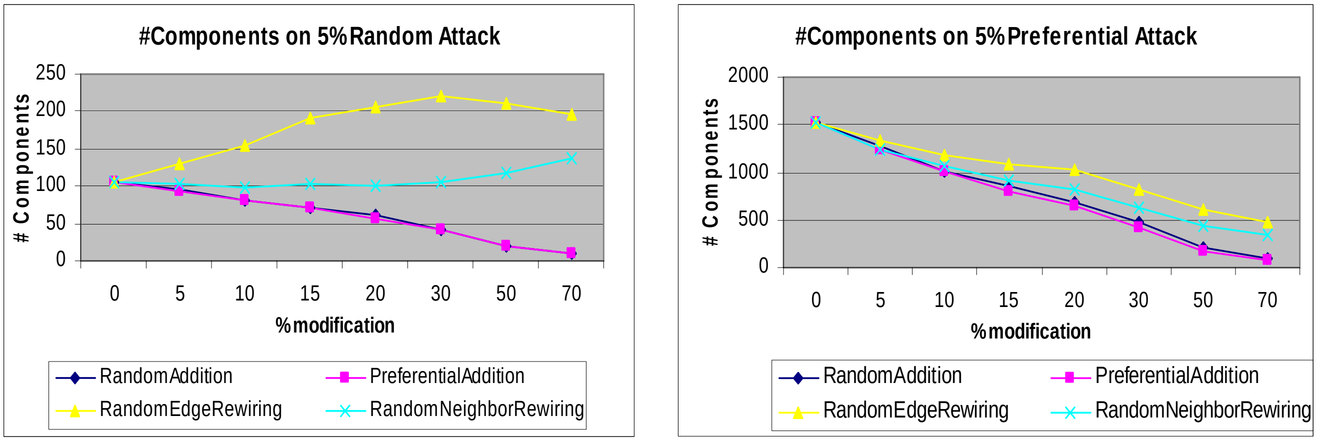


Figure 4.3 : The effect of modification schemes on the Number of Components of the network for both random and preferential attack.

4.2. Dynamic Analysis

The dynamic analysis considers how the network changes with time. The metrics we measure in the dynamic analysis are the percolation point, which is the point at which the network disintegrates on removing a constant fraction of the nodes at each instant, and number of nodes failed due to the cascading effect of the model of communication we proposed earlier.

Figure 4.4 shows how the percolation point changes as a function of the percentage modification for each of the modification schemes. It can be seen that the percolation point is reached slower when Addition Schemes are used and also gets slower with in increase in the % modification. From the graph we can also see that Random Neighbor Rewiring Performs as well as the Addition Schemes.

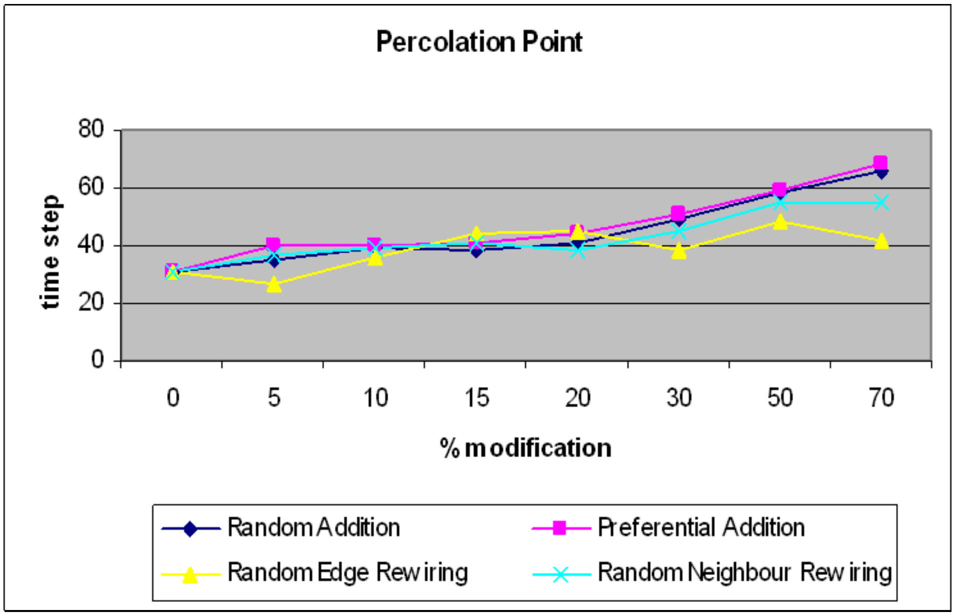


Figure 4.4 : The effect of modification schemes on Percolation Point of the network.

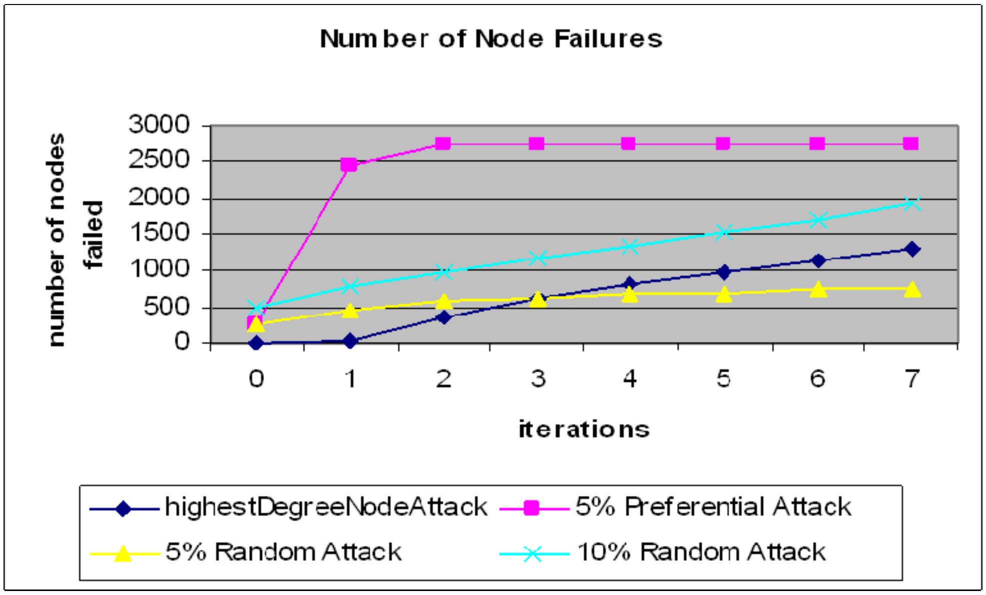
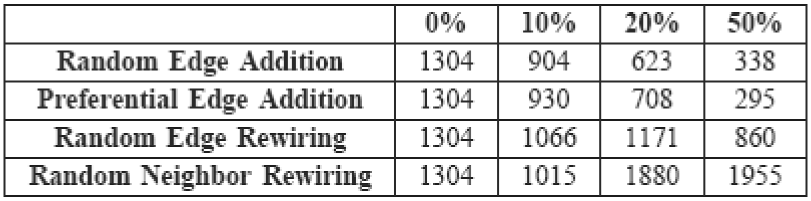


Figure 4.5 : Node Failure due to Cascading Effect for different types of Attacks

Figure 4.5 shows the failure rate of nodes for random and targeted attacks when cascading is considered. Preferential attack on 5% nodes causes more than half of the nodes to fail in the network (in only two iterations), as expected. It can be seen that the removal of highest degree node is more devastating for the network than attacking 5% nodes of the network randomly.

Table 3 : Cascading effect on Removal of Highest Degree Node of Gnutella Network

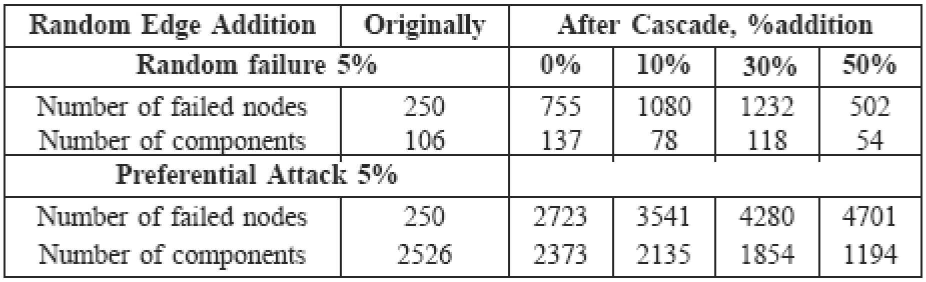


	0%	10%	20%	50%
Random Edge Addition	1304	904	623	338
Preferential Edge Addition	1304	930	708	295
Random Edge Rewiring	1304	1066	1171	860
Random Neighbor Rewiring	1304	1015	1880	1955

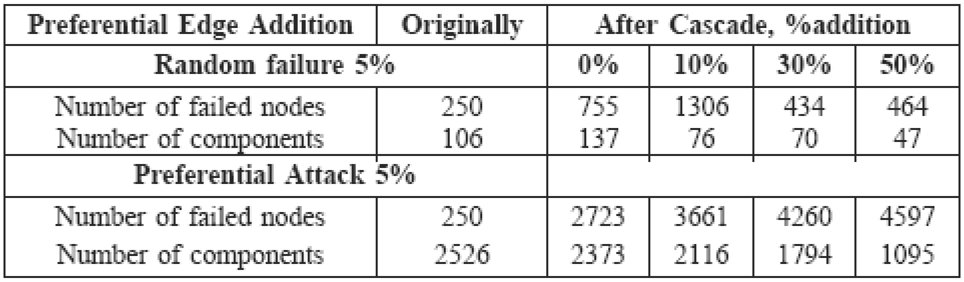
Table 3 shows the cascading effects on removal of the highest degree node from the network. As we had stated earlier, removal of a highly connected node from the network adversely affects the information flow capability of the network. This fact can be easily seen here as removal of the highest degree node from the network causes 1304 nodes to fail in 8 iterations of cascaded analysis. Figure

Table 3 also highlights the performance of various edge modification schemes. Clearly, *edge addition schemes* perform better than the *edge rewiring schemes* as they increase the connectivity between nodes. They create more shortest paths between nodes not passing through the highest degree node. Therefore the amount of load to be redistributed after the removal is less, and hence causes less nodes to fail due to the redistribution. The *edge rewiring schemes* do not perform well, as they do not contribute much in shifting the betweenness of the highest degree node to other nodes in the network.

Table 4 : Cascading Effect with Edge Addition Schemes on Gnutella Network



Random Edge Addition	Originally	After Cascade, %addition			
Random failure 5%		0%	10%	30%	50%
Number of failed nodes	250	755	1080	1232	502
Number of components	106	137	78	118	54
Preferential Attack 5%					
Number of failed nodes	250	2723	3541	4280	4701
Number of components	2526	2373	2135	1854	1194



Preferential Edge Addition	Originally	After Cascade, %addition			
Random failure 5%		0%	10%	30%	50%
Number of failed nodes	250	755	1306	434	464
Number of components	106	137	76	70	47
Preferential Attack 5%					
Number of failed nodes	250	2723	3661	4260	4597
Number of components	2526	2373	2116	1794	1095

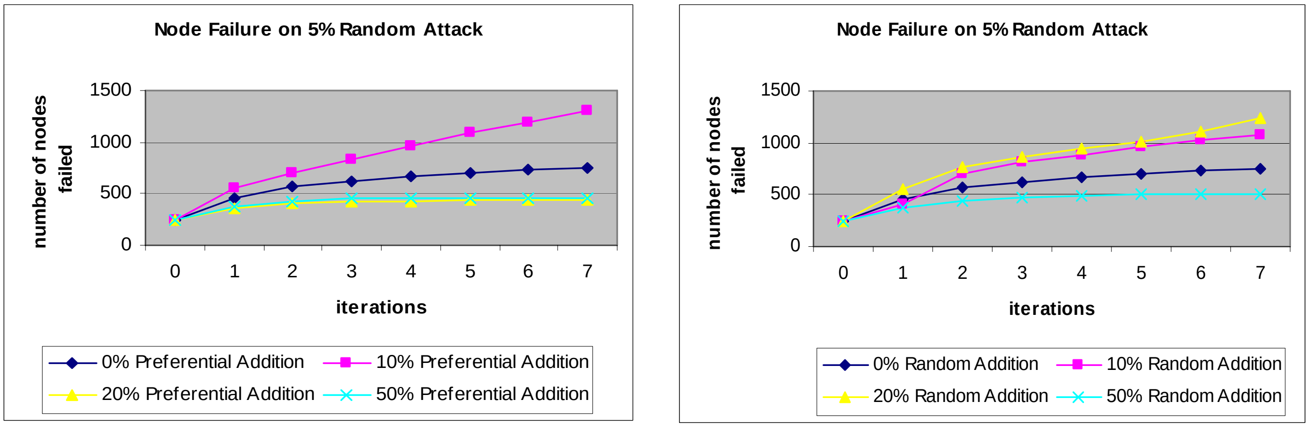
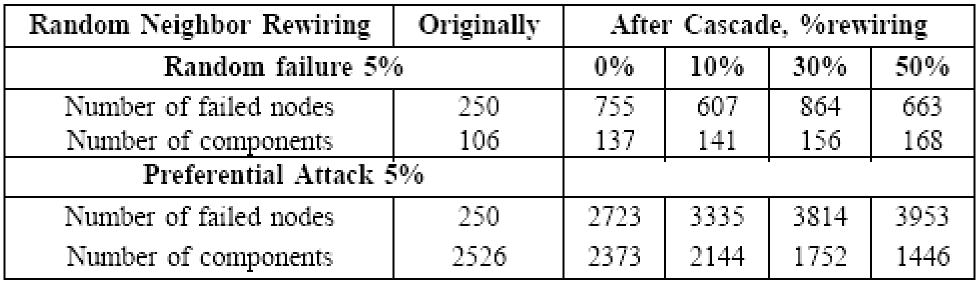


Figure 4.6 : Node Failure due to Cascading Effect as a function of the amount of Addition

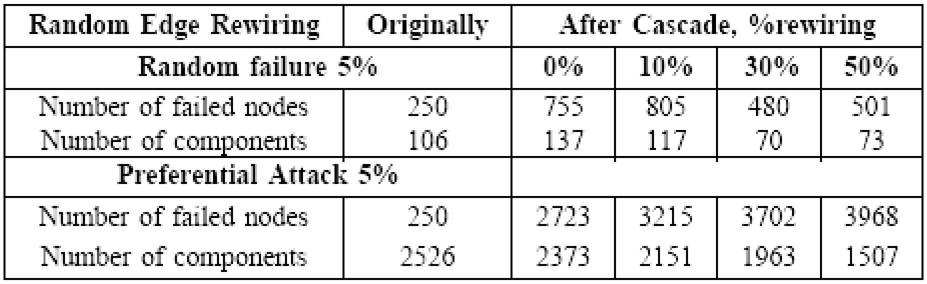
We also evaluated different edge modification strategies when a small fraction of the network nodes are removed. We show the simulation results obtained for 5% random and preferential attacks. Table 4 shows the results for edge addition schemes and we find that when a larger number of nodes in the network are *randomly* removed, *preferential addition* is more efficient. This is represented graphically in Figure 4.6. Random addition loses out to preferential addition scheme as the randomly chosen nodes which gain edges and contribute in new shortest paths are most likely removed in random failure. In case of *preferential attacks* both the schemes fail to make any improvement in the network.

Results on using rewiring schemes are shown in Table 5. It is seen that rewiring schemes also do not perform well in case of *preferential attack* as compared to *random failure*. But it can be seen that at lower modification percentages the *rewiring schemes* are better than *addition schemes*.

Table 5 : Cascading Effect with Rewiring Schemes on Gnutella Network



Random Neighbor Rewiring	Originally	After Cascade, %rewiring			
Random failure 5%		0%	10%	30%	50%
Number of failed nodes	250	755	607	864	663
Number of components	106	137	141	156	168
Preferential Attack 5%					
Number of failed nodes	250	2723	3335	3814	3953
Number of components	2526	2373	2144	1752	1446



Random Edge Rewiring	Originally	After Cascade, %rewiring			
Random failure 5%		0%	10%	30%	50%
Number of failed nodes	250	755	805	480	501
Number of components	106	137	117	70	73
Preferential Attack 5%					
Number of failed nodes	250	2723	3215	3702	3968
Number of components	2526	2373	2151	1963	1507

A high percentage of addition is required to gain more advantage than the rewiring schemes. This observation is particularly important because in case of removing a set of nodes and not just the highest degree node, rewiring is more beneficial than and also not as costly as addition. At high modification percentages, edge addition schemes outperform both the rewiring schemes which is expected, but high percentage of addition would also be extremely costly.

Chapter 5

Effect of Routing Strategies

5.1. Introduction

The routing strategy used to route packets in the network decides the flow of traffic among the network nodes. A few examples of these strategies could be to forward the packet to a random neighbor (random-walk), or to send the packet to all the neighbors (flooding), or to send it to the highest degree neighbor (preferential-walk), or to send the packet to the destination by the shortest path. Each of these routing strategies require different amount of information of the network, for example, only the degree of neighbors in preferential-walk or the information of the whole network which routing using the shortest paths.

Each of these strategies might choose different nodes while routing between the same source and destination based on the criteria they emphasize. This would lead to an uneven distribution of the load at each node based on its location in the graph causing congestion at some nodes. Therefore the routing scheme along with the network topology results in the congestion of certain nodes. If these nodes stay congested, stalling the passing of messages between other nodes, the congestion would spread causing more delay or failure of transmission. The amount and speed of the spread of the congestion in the rest of the network depends on the network topology as we have seen in the earlier results.

Therefore, the amount of propagation of the node failure in the network depends both on the network structure as well as the routing strategy followed to route messages in the network. The cascading effect on the gnutella network is studied while using different routing schemes. The impact of the routing scheme is observed by comparing the various schemes on the same network, whereas the importance of the topology is observed by comparing the results on gnutella network to those on a scale-free power law graph.

We tried to study the cascading effect when random routes were used to communicate. It is important to study the random paths in the network because on absence of global information in the network, the node has to route packets based wholly on local knowledge. The basic routing strategy is to send the message to a random neighbor when no information is available. This is also called a random walk in the network. We also show the simulation results when partial global data is available. That is we follow the shortest path to route when we know it and we use a random path when the shortest path information is not available. We see how the network is affected as a function of deviation from following shortest paths.

5.2. Routing Model

The model considered is as follows. At any time instant, some random source nodes try to communicate with random destination nodes by sending packets. This time instant is assigned for the transmission of these packets from source to destination. Each node in the network has a capacity. Two case were analyzed, one with all nodes having constant capacity and the second where the capacity of the node is proportional to its degree. A node is considered congested if the number of packets routed through this node at a time instant exceeds its capacity. Congested nodes are considered failed, since these nodes can't be used for routing immediately. As in the earlier analysis, we don't consider the recovery of nodes from the congested state. Therefore they are removed from the network. This process is repeated for 100 steps and its effect on the network is studied.

First the number of pairs of nodes allowed to communicate at a time instant is identified by simulating the model with different values of this number and then a value which stabilizes the total node removals is chosen. Shortest paths are used to route messages in the network and a constant capacity of 5 is assigned to each node. Figure 6.1 shows the cascading effect as a measure of number of node failed over time. It can be seen that at 100 pairs per iteration, the number of nodes failed stabilized even though it reaches a high fraction of the network. Therefore, the number of pairs to communicate per iteration is taken as 100, i.e., 100 random pairs are allowed to communicate before the congested nodes are checked for and removed from the network.

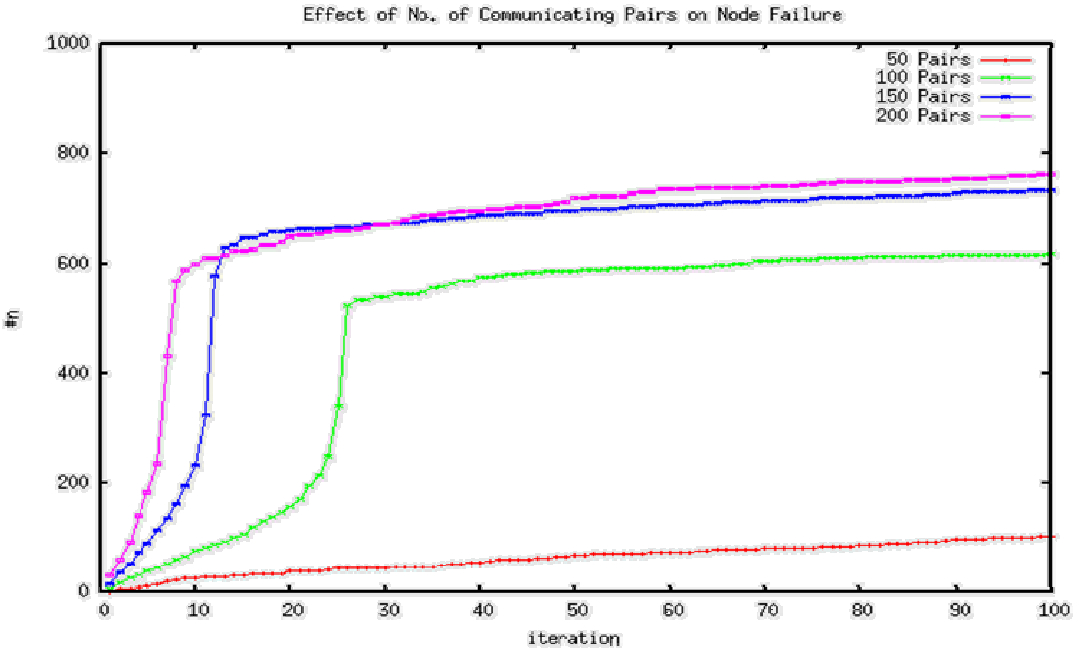


Figure 5.1 : Effect of No. of Communicating Pairs on the cascading effect of node failure on using the shortest paths to route the packets and a constant capacity of 5 at each node.

The routing schemes considered for routing packets in the network are the “random walk” routing and “shortest path” routing. As mentioned earlier these are the two extreme cases in terms of the amount of network knowledge required to route packets. We compare these two routing strategies on a given network. Also, how the network is affected due to deviation from one routing scheme to the other is studied by considering deviation percentages of 20, 40, 60 & 80.

5.3. Random Walks

More recently, random walks on finite graphs have received much attention, and mostly to measure the quantitative aspects such as how long we have to walk before we return to the starting node? , before we see a given node?, before we see all nodes?, etc.[11]. Work has also been done on the relation of random walks with the node properties, which are a direct consequence of the network topology. Some important results regarding the node properties of the nodes chosen in Random Walks are understood better before trying to study the effect of the network topology in the model proposed above.

The basic definition of a random walk on a graph is as follows: given a graph and a starting point, we select a neighbor of it at random, and move to this neighbor; then we select a neighbor of this point at random, and move to it etc. The (random) sequence of points selected this way is a random walk on the graph[9]. Let **G = (V,E)** be a connected graph with ***n*** nodes and ***m*** edges. Consider a random walk on **G** which starts at the initial node v_0. If at the ***t***-th step we are at a node ***i***, we move to ***j***, a neighbor of ***i*** with a probability ($1/K_i$); ie., the walker selects the neighbors of ***i*** with equal probability. Therefore the transition probability P_{ij} to go to node ***j*** from node ***i*** at time ***t*** is:

$$P_{ij}(t) = \sum_k \frac{A_{kj}}{K_k} P_{ik}(t-1), \tag{1}$$

where A_{kj} is the entry in the adjacency matrix (equal to 1 if ***k*** and ***j*** are neighbors, otherwise 0) and K_k is the degree of node ***k***. The explicit expression for the transition probability $P_{ij}(t)$ to go from node I to node j in t steps is obtained by iterating equation (1) as follows:

$$P_{ij}(t) = \sum_{j_1 j_2 \ldots\ldots j_{t-1}} \frac{A_{ij_1}}{K_i} . \frac{A_{j_1 j_2}}{K_{j_1}} \cdots \frac{A_{j_{t-1} j}}{K_{j_{t-1}}} . \tag{2}$$

While comparing the expressions for $P_{ij}(t)$ and $P_{ji}(t)$, due to the undirectedness of the network, we can see that

$$K_i P_{ij}(t) = K_j P_{ji}(t). \qquad (3)$$

For the stationary solution, i.e. at the infinite time limit, equation (3) implies that $\boldsymbol{K_i\, P_j^{\infty} = K_j\, P_i^{\infty}}$, where $\boldsymbol{P_j^{\infty}} = \mathbf{lim}_{\mathbf{t\,->\,0}}\, \boldsymbol{P_{ij}(t)}$. Therefore we get

$$P_i^{\infty}(t) = \frac{K_i}{\mathcal{N}}, \quad \text{where } \mathcal{N} = \sum_i K_i. \qquad (4)$$

This shows that the more links a node has to other nodes in the network, the more often it will be visited by a random walker. We have tried to represent this result also in terms of the Random Betweenness Centrality. Betweenness Centrality of node is defined as the fraction of shortest paths between node pairs that pass through this node. Therefore, betweenness reflects the amount of the influence a node has over the spread of information through the network. By considering only shortest paths, the betweenness centrality shows the influence of any node when the information routing is based on shortest paths. We could take into consideration the routing strategy while measuring the influence of a node in the spread of information. This would help us to find the ‘important’ nodes of the given network topology specific to the routing strategy used. A measure based on the random walks, random-walk betweenness, is proposed by M.E.J Newman [10] which counts the expected number of times a node is traverse by a random walk between two other nodes.

The method used to compute this measure of random-walk betweenness is briefly explained here:

1. Construct the matrix **D−A**, where **D** is the diagonal matrix of vertex degrees and A is the adjacency matrix.

2. Remove any single row, and the corresponding column. For example, one could remove the last row and column.

3. Invert the resulting matrix and then add back in a new row and column consisting of all zeros in the position from which the row and column were previously removed (e.g., the last row and column). Call the resulting matrix **T**, with elements $\mathbf{T_{ij}}$.

4. Calculate the betweenness from Eq. (7), using the values of **I_i** from Eqs. (5) and (6).

$$I_i^{(st)} = \frac{1}{2}\sum_j A_{ij}\left|T_{is} - T_{it} - T_{js} + T_{jt}\right|, \text{ for } i \neq s, t. \qquad (5)$$

$$I_s^{(st)} = 1, \; I_t^{(st)} = 1. \qquad (6)$$

$$b_i = \frac{\sum_{s<t} I_i^{(st)}}{\frac{1}{2}n(n-1)} \qquad (7)$$

Since the random betweenness of a node counts the number of paths through it, i.e. the number of times the node is visited by a random walker, it should be proportional to the degree of the node. Figure 6.2 shows this result true for a scale-free power-law graph. It can be seen that the random betweenness values are indeed proportional to the degree of the nodes.

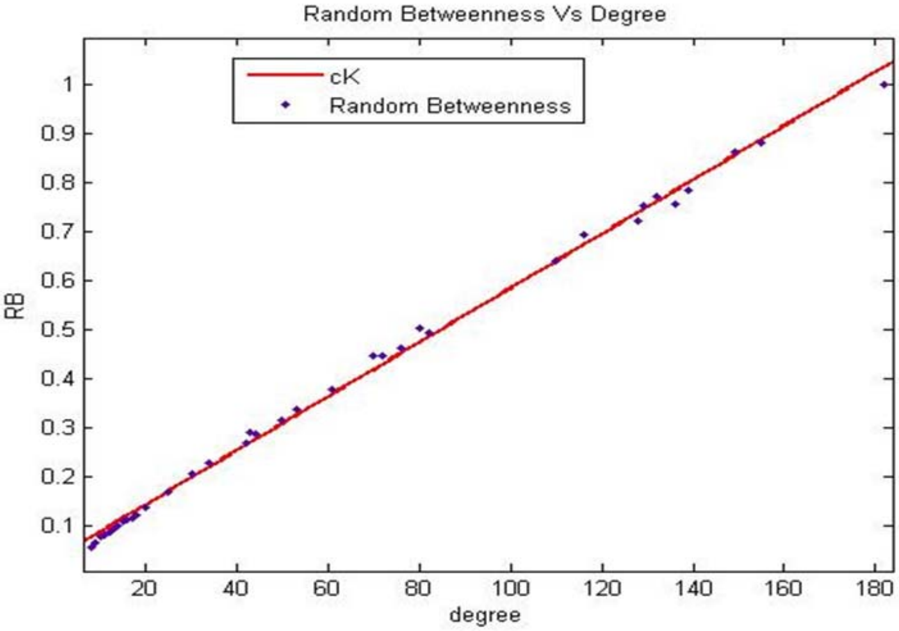


Figure 5.2 : Graph showing the relation between random betweenness of the nodes and their degree for a power law graph.

We considered two variations of random walks. One is the random walk as defined above but with the restriction that a node visited once cannot be visited again when going from the source node to the destination. This is obtained by constructing a random spanning on the network. We call this scheme the ***Random Walker 1*** from here on. The second one is a variation to the first scheme. In the first the spanning tree is built by expanding the tree from a random node. The second scheme uses selection criteria for the node which would lead to the expansion of the tree. The node chosen to expand is the node to which path degree is the highest. We define the path degree as the sum of degrees of all the nodes on a path. We call this scheme the ***Random Walker 2*** from here on. The betweenness of nodes can be computed by simulating the routing schemes on the network and counting the number of times a node is selected in a path between a pair of nodes. Since to compute this measure for all pairs of nodes is very computationally intensive, we simulated the routing schemes to communicate from about 150 nodes to all other nodes. That is, the random walkers are initiated 4999 times at each of the 150 nodes. The computed Random Walkers Betweenness are compared with the Random Betweenness of the nodes.

Figure 6.3 compares the betweenness of nodes when Random Walker 1 is simulated to the theoretical Random Betweenness. It can be seen that the trend of increase in value of betweenness with the increase in degree is followed in the simulated values also. But it can be seen that the betweenness is not proportional the degree. In fact the betweenness values are proportion to the square of degree. Similar behavior can be seen in Figure 6.4 for the case of Random Walker 2. This deviation could be due to observing only a fraction of communicating pairs in the simulation of the random walkers. There is also a restriction on the number of times a node can appear on a path between two nodes, whereas no such restriction is present while calculating the theoretical Random Betweenness.

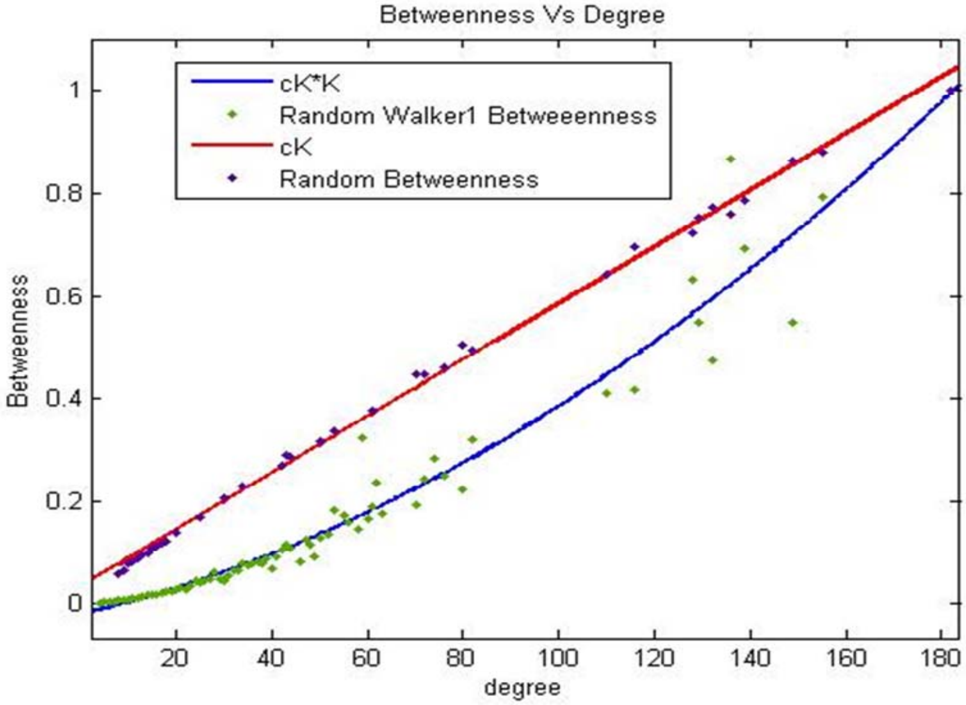


Figure 5.3 : Graph showing the comparison between Random Betweenness and Random Walker1 Betweenness for power-law graph.

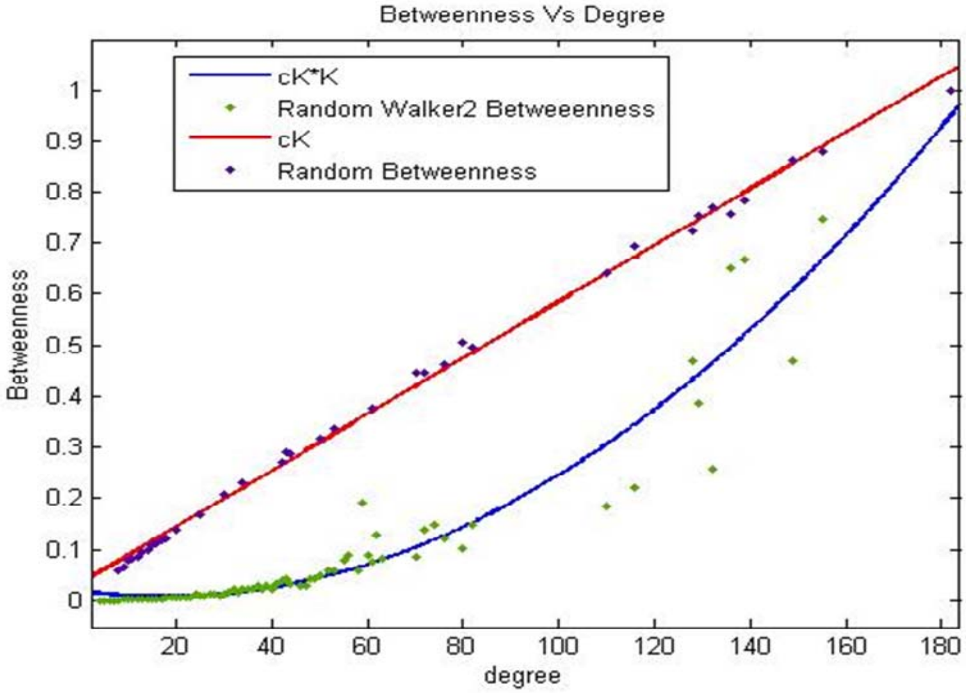


Figure 5.3 : Graph showing the comparison between Random Betweenness and Random Walker2 Betweenness for power-law graph.

5.4. Cascading Effect

The routing model is simulated on the Gnutella graph described in Chapter 4. The cascading effect is studied for both the random walking variations and compared with the shortest path routes to see how many nodes are getting removed, and how the network is affected by these removals, i.e. the size of the LCC and the number of components, after each iteration. The effect of the amount of global information present is also observed by applying various percentages of deviation from the number of shortest paths used for routing.

5.4.1. Random Walker 1

Figure 6.5 shows the cascading effect in terms of number of nodes failed due to congestion for the case where random walker 1 is used to route packets between the random source and destination pairs. It can be seen that on average the number of nodes failed after about 30 iterations is almost equal in both Random Path Routing and Shortest Path Routing. This value reaches to about 600 nodes and stabilizes. The more important observation is how the two routing schemes affect the network before they reach the point after which there is not much difference between the two. The steep raise in the value of the number of nodes failed occurs much sooner in case of Random Path Routing (12-17th iteration), compared to Shortest Path Routing (20-25th iteration).

The point at which the number of nodes removed is stabilized varies between 500 and 600 for the various % deviations from shortest paths. This shows that there are at max 500 important nodes in the network on whose removal the whole network disintegrates. And these nodes are made to fail in all three routing schemes given sufficient time, 30 iterations for the considered routing model. Also it can be seen that the number of node failures never exceeds 600 nodes in all the cases.

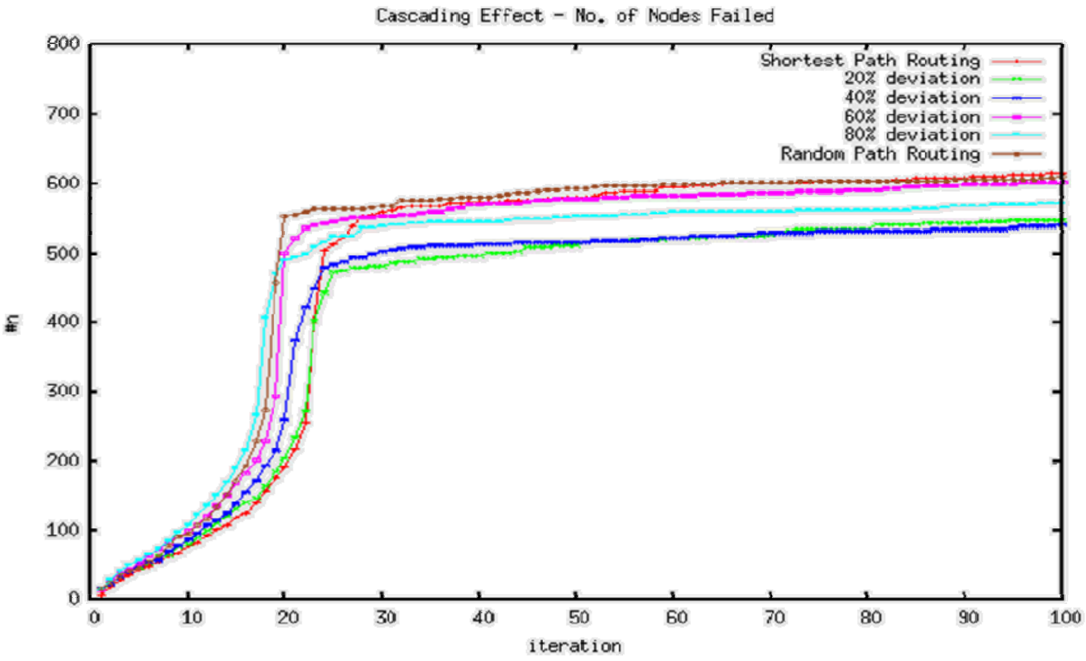


Figure 5.5 : Cascading effect in terms of number of nodes failed for Random Walker 1

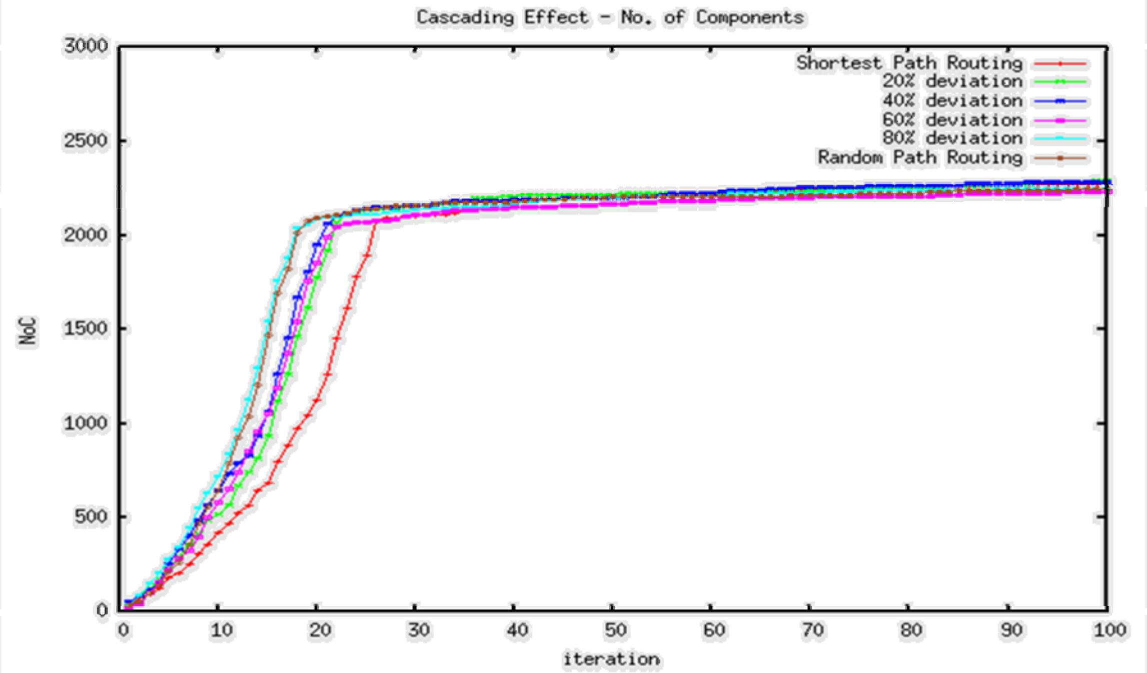


Figure 5.6 : Cascading effect in terms of number of components for Random Walker 1

Similar trends are observed while measuring the number of components and the size of the largest connected component as shown in Figure 6.6 and Figure 6.7 respectively. The number of components increases drastically due to the removal of nodes and reaches a value about 2225. At this point the network is completely disconnected having a lot of small components, mostly single disconnected nodes. This can be confirmed by looking at how the size of the largest connected component decreases drastically and reaches to a very low value.

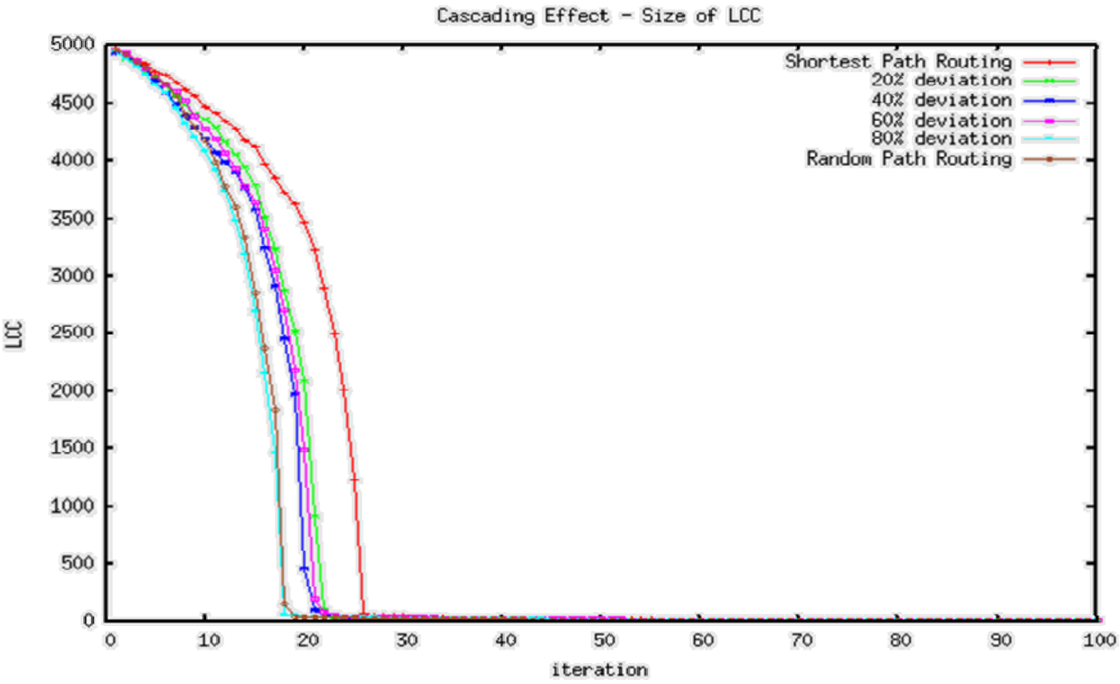


Figure 5.7 : Cascading effect in terms of size of LCC for Random Walker 1.

5.4.2. Random Walker 2

Figure 6.8 shows the cascading effect in terms of number of nodes failed due to congestion for the case where random walker 2 is used to route packets between the random source and destination pairs. Similar results are observed as in the case of Random Walker 2. On average the number of nodes failed after about 30 iterations is almost equal in both Random Path Routing and Shortest Path Routing. Here also this value reaches to about 600 nodes and stabilizes as in the case Random Walker 1. This agrees with our earlier observation of the presence of important nodes, which are about one-tenth the size of the network in number, on whose removal the whole network disintegrates.

The difference in the points where the number of failures takes a steep raise is also present and is more prominent as they are more spread over the time. The steep raise even much earlier for Random Path Routing (6-8th iteration) and there is a smooth increase in the value of the point with the decrease in the percentage of deviation and finally reaches the point of rise for the Shortest Routing Scheme (24-26th iteration).

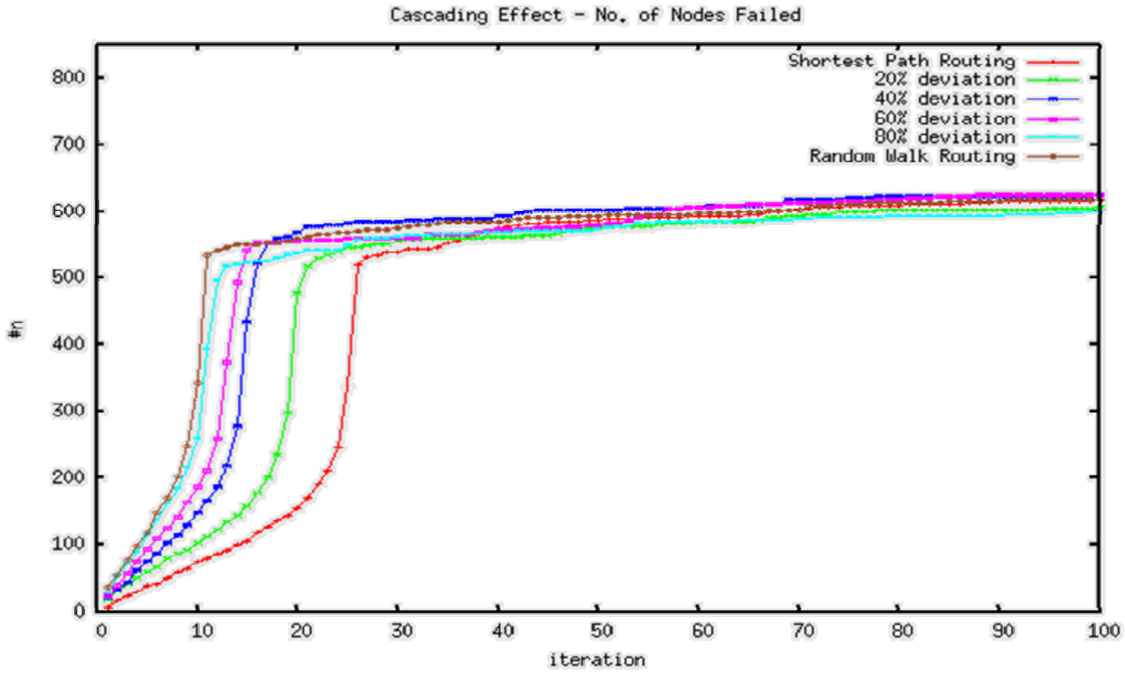


Figure 5.8 : Cascading effect in terms of number of nodes failed for Random Walker 2.

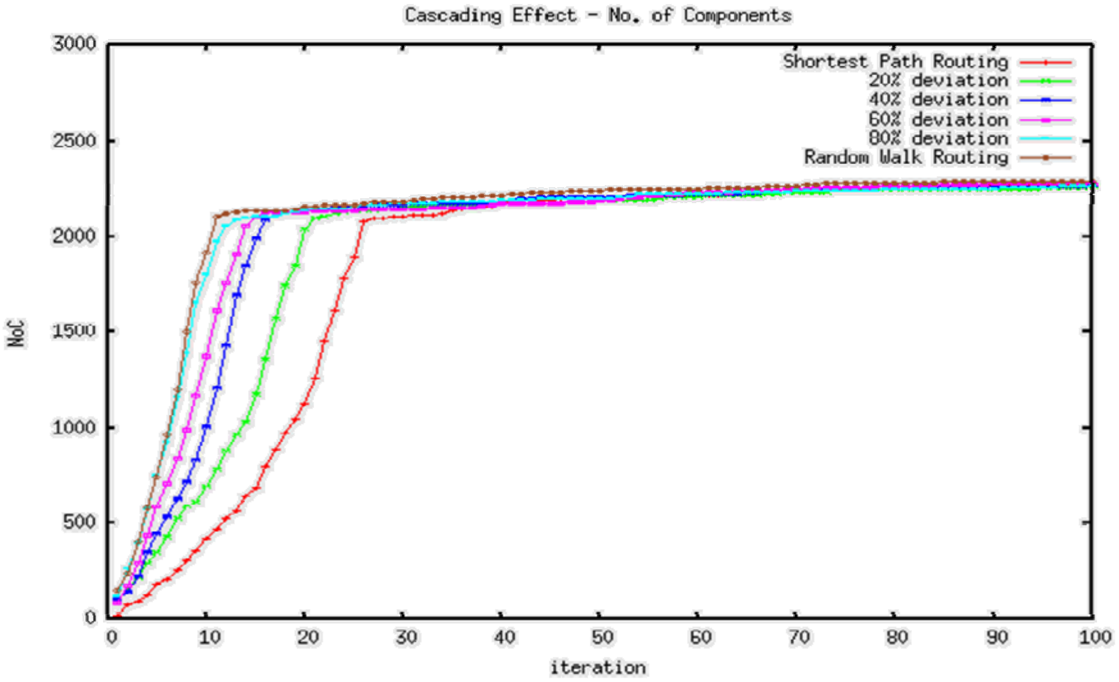


Figure 5.9 : Cascading effect in terms of number of components for Random Walker 2.

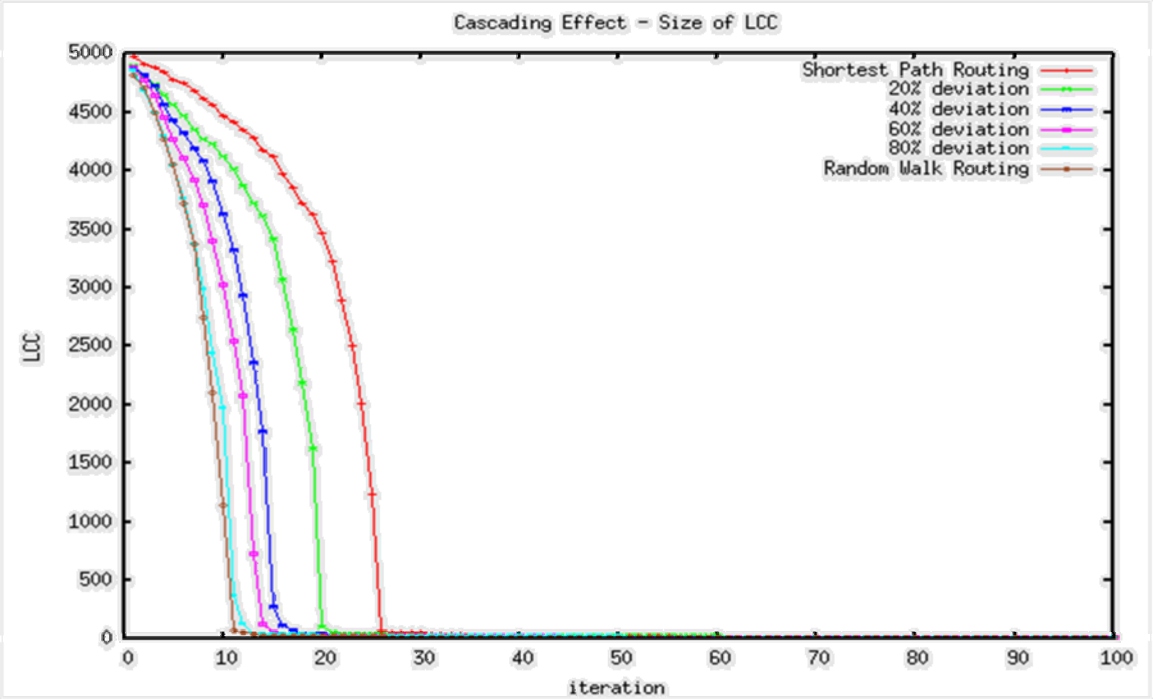


Figure 5.10 : Cascading effect in terms of number of components for Random Walker 2.

5.4.3. Comparison with results on power-law graph

The results on Gnutella network are compared with those on scale-free power-law graph to understand the effect of the topology. Figure 6.11 and Figure 6.12 show the cascading effect in terms of number of nodes failed due to congestion when random walker 1 and random walker 2 are used respectively. It can be seen that the number of nodes failed do not stabilize till the 100th iteration, and the maximum number is attained when Random Walker 2 is used. This implies that the power-law graph topology is more robust than the Gnutella network topology. Also, it can be seen that the power-law network is more vulnerable to Random Walker 2 and least vulnerable to Random Walker 1.

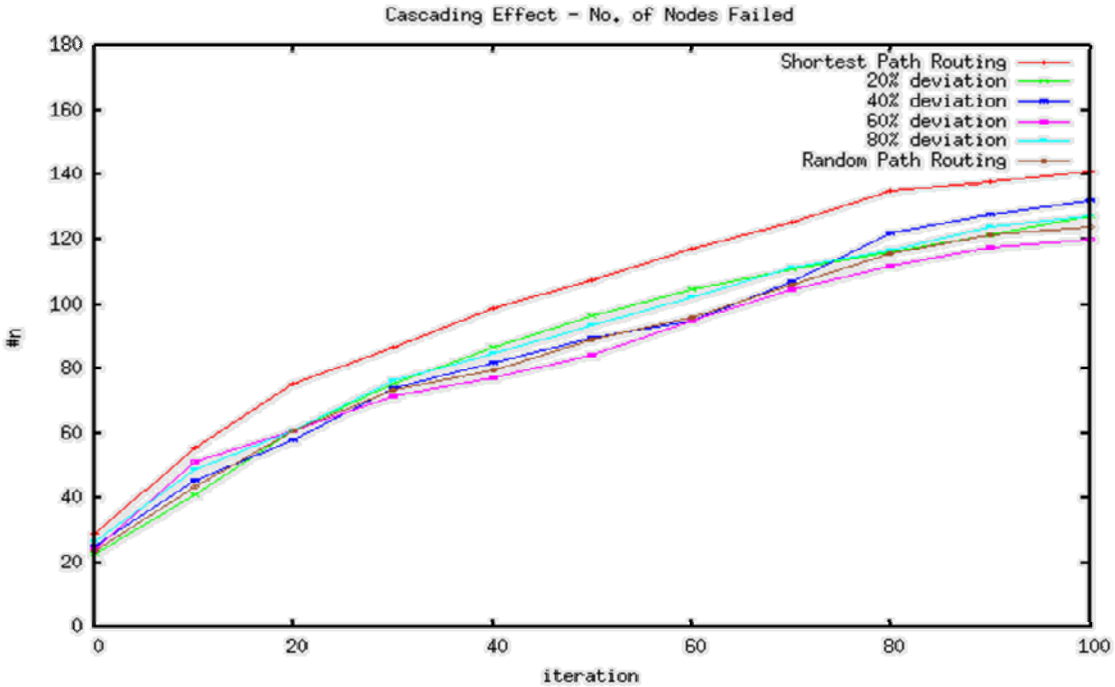


Figure 5.11 : Cascading effect in terms of number of nodes failed for Random Walker1.

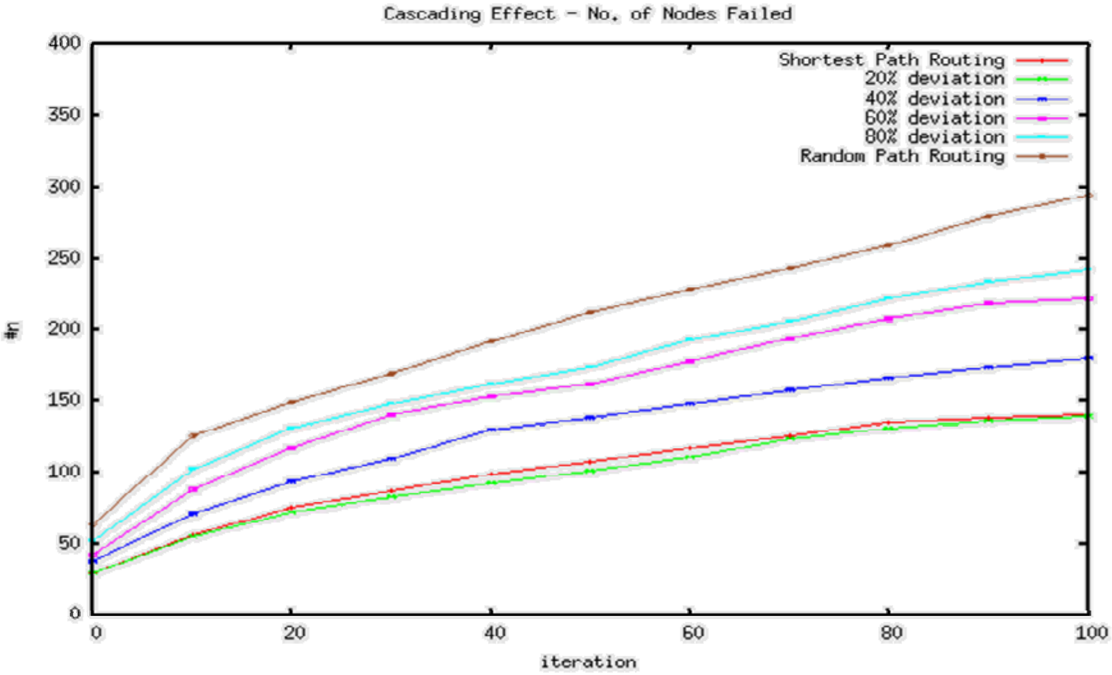


Figure 5.12 : Cascading effect in terms of number of nodes failed for Random Walker 2.

5.5. Routing Analysis

The above results of Cascading effect can be explained by looking at the nodes that have been removed at each of the iteration. For this let us look into the total degree of the nodes removed at each iteration. Figure 6.13 compares the total degree of the nodes removed per iteration for Random Walker 1 and Shortest Path Routing, and Figure 6.14 compares the total degree of nodes removed per iteration for Random Walker 2 and Shortest Path Routing.

It can be seen that the total degree of removed nodes is very high initially for the Random Walk Routing compared to Shortest Path Routing. This means either high degree nodes are removed or lots of smaller degree nodes are removed. Due to this there is an earlier degradation of network in case of Random Path Routing. On observing the data it has been seen that both high degree nodes and also large number of medium degree nodes are removed initially while using Random Path Routing schemes.

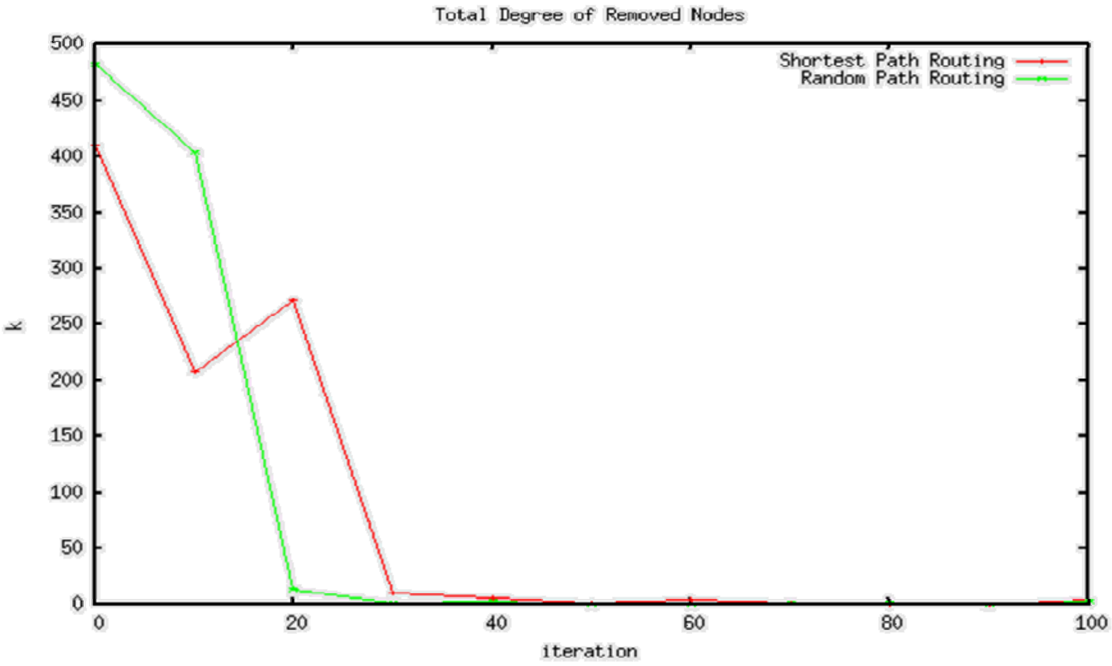


Figure 5.13 : Total degree of nodes removed – Random Walker 1.

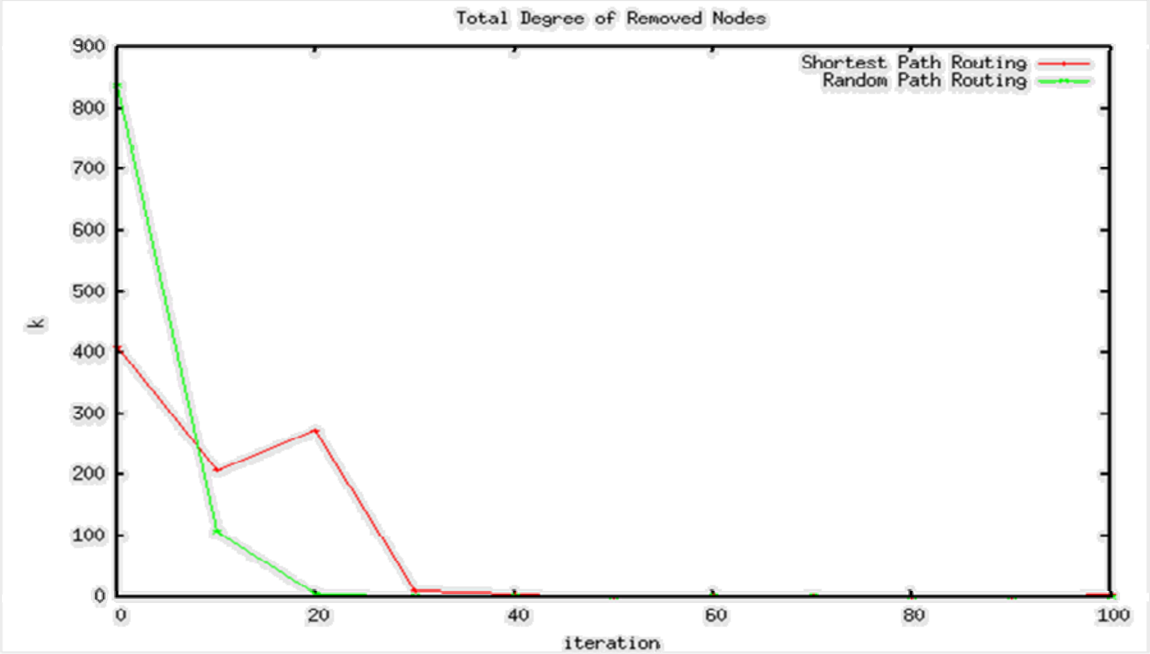


Figure 5.14 : Total degree of nodes removed – Random Walker 2.

Since the knowledge of the degree of the nodes removed by the routing schemes in important in analyzing the effect they have on the network, we tried to formulate the degree of the nodes removed due to random walks. Let the probability that a node had degree $\boldsymbol{k}$ be $\boldsymbol{p_k}$ and the probability that a given node fails due to congestion be $\boldsymbol{q_k}$. Therefore, the probability that a node of degree $\boldsymbol{k}$ fails due to congestion is $\boldsymbol{q_k p_k}$. This probability can be expressed in terms of generating functions as follows

$$G_0(x) = \sum_k q_k p_k x^k.$$

Taking a power-law scale-free graph with the degree distribution $\boldsymbol{p_k \propto k^{-1}}$. As discussed in the previous section, in random walks the number of times a node gets visited is proportional to its degree $\boldsymbol{k}$. Since the capacity of the node is assumed to be constant, the probability of a node failing, i.e. the probability of a node exceeding its capacity is proportional to the number of times it gets selected as a node in the paths between other nodes. Therefore the probability of failure of a given node is proportional to its degree. Hence, we have $\boldsymbol{q_k \propto k}$. So, we can write the generating function of the probability of a node of degree $\boldsymbol{k}$ to fail as

$$G_0(x) = \sum_k ckk^{-1}x^k = \sum_k cx^k$$, where $\boldsymbol{c}$ is a constant.

The probability of a node to get congested and eventually removed from the network is shown to be independent of its degree. Therefore higher degree nodes tend to fail much more than the lower degree nodes.

We try to verify the above formulation by simulating the random walkers on the power-law graph to route packets between 10,000 random pairs and seeing which nodes exceed their capacity and noting their degree. Figure 6.15 shows the results for random walker1 and Figure 6.16 shows the results for random walker 2. It can be seen that the distribution is proportional to $\boldsymbol{k}$. This follows our earlier result, where the betweenness measured for the random walkers was proportional to $\boldsymbol{k^2}$ for lower degrees. By replacing $\boldsymbol{q_k}$ by $\boldsymbol{k^2}$ instead of $\boldsymbol{k}$ we get a more approximate equation for our random walkers which says the probability of failure of a node of degree $\boldsymbol{k}$ is proportional to its degree. Therefore, the simulation results support our formulation.

Though Figure 6.15 shows a better fit for lower degrees is a ***k***^{2} curve, above the degree 40, the data points show that the probability is linear to the degree.

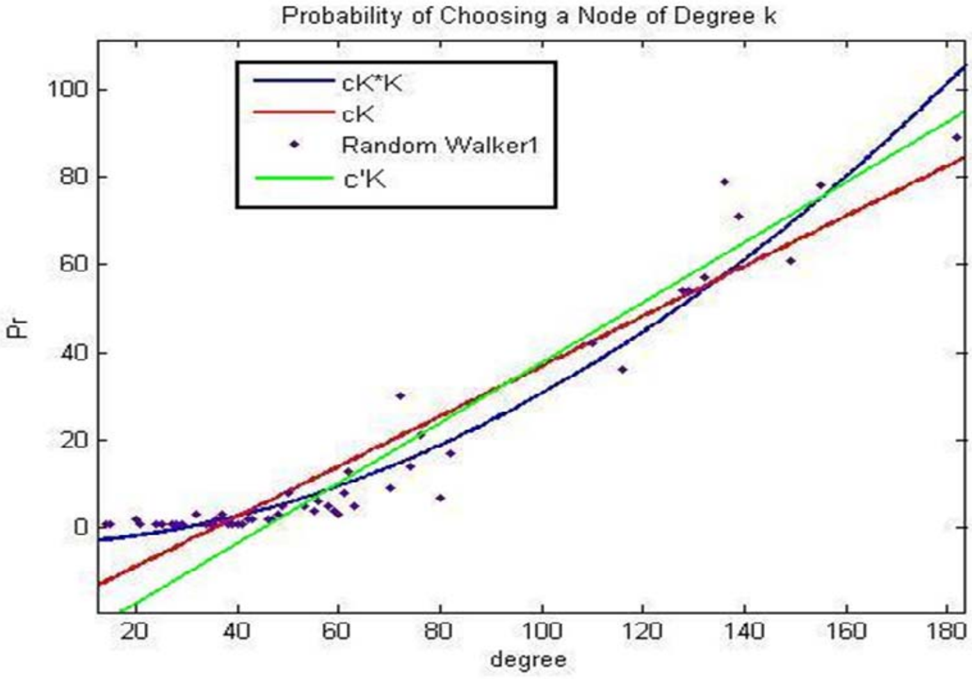


Figure 5.15 : Graph showing the probability of choosing a node of degree k while using Random Walker1 on a power-law graph.

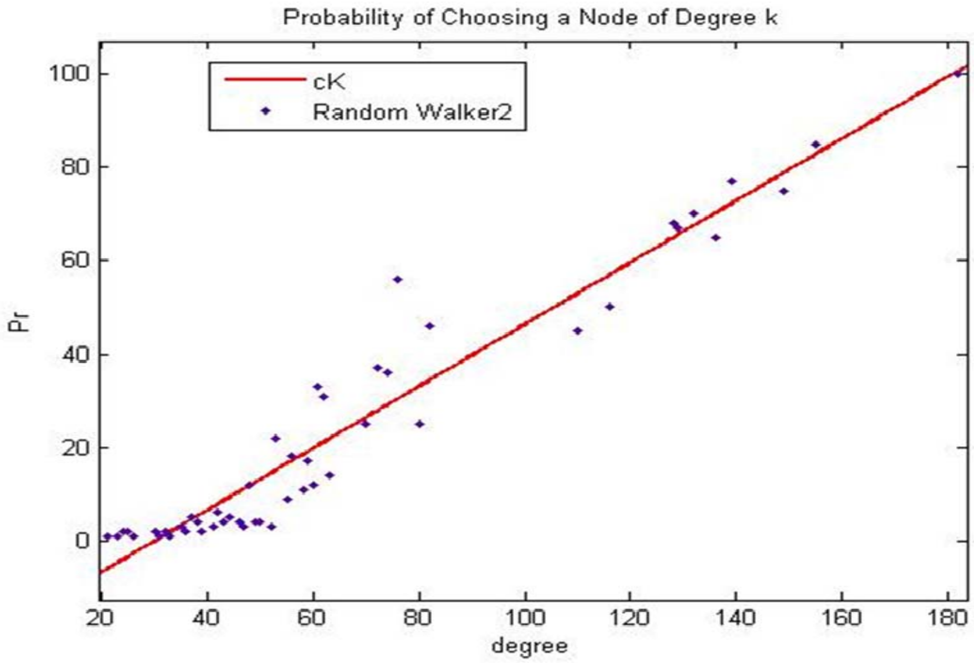


Figure 5.16 : Graph showing the probability of choosing a node of degree k while using Random Walker2 on a power-law graph.

The graphs also explain the difference of the affect the Random Path Routing scheme had on the cascading effect results observed on the scale-free power-law graph. In case of Random Walker 2, not only the high degree nodes have a high probability of failure but also the medium degree nodes. This leads to a larger number of medium and high degree nodes to fail hence increasing the cascading effect in the network.

Chapter 6

Conclusion

In peer-to-peer networks, it is very important to know how to tackle random failures and targeted attacks in an efficient way as they are very common. We have shown that with small modifications we can improve robustness of these networks. We have dealt with the 'preventive' methodology i.e., trying to modify the network to make it robust against attacks and failures. In our simulation for static analysis, we have noticed that addition schemes perform better than the rewiring schemes as expected, but they are expensive. Considering the cost incurred while rewiring or adding the edges, we see that the Random neighbor rewiring performs better than the others as it tries to equalize the degree among all the nodes, making the network more robust against targeted attacks. The cascading effects in the peer-to-peer networks are demonstrated by taking a simple data flow model. We have also performed the dynamic analysis for the various modification schemes which has given us more insight into the usefulness of the rewiring schemes over addition schemes when a small fraction of network nodes are removed. The knowledge of how the various modification schemes affect the robustness of the network can be used to design better distributed network management protocols.

The effect of routing on the dynamics of the network has been studied. This gives us an insight into how different routing strategies can lead to congestion at different nodes. This knowledge is useful in selecting a suitable routing scheme, given a network topology, which leads to efficient network communication. Therefore, in cases where we have no control on the topology of the network or changing the topology is very costly, we can implement a routing strategy with increases the throughput of the network.

Bibliography

[1] A.Beygelzimer, G.Grinstein, R.Linsker and I.Rish - *Network Robustness by Edge Modification*, Physica A, Volume 357, Issue 3-4,p.593-612.

[2] P.Crucittia, V.Latorab, M.Marchioric and A.Rapisardab - *Error and Attack Tolerance of Complex Networks*, Nature. 2000 Jul 27, 406(6794):378-82.

[3] R.Albert and A.Barabasi - *Statistical Mechanics of Complex Networks,* Reviews of Modern Physics 74, 47 (2002)

[4] Ying-Cheng Lai, A.E.Motter and T.Nishikawa - *Attacks and Cascades in Complex Networks*, Lecture Notes in Physics, 2004, Springer.

[5] Jian-jun Wu, Zi-you Gao and Hui-jun Sun - *Cascade and Breakdown in scale-free Networks with Community Structures*, Physical Review E, 2006, APS.

[6] P. Erdos, A. Renyi - *On the Evolution of Random Graphs*, Publ. Math. Inst. Hangar Acad. Sci., 5, 1960, 17-61.

[7] A. E. Motter and Ting-cheng Lai - *Cascade-based attacks on Complex Networks*, Physical Review E, 2002, APS.

[8] U. Brandes - *A Faster algorithm for Betweenness Centrality*, Journal of Mathematical Sociology, 2001.

[9] J. D. Noh, H. Rieger - *Random Walks on Complex Networks*, Physics Review Letter 92, 118701.

[10] M.E.J Newman – *A measure of betweenness centrality based on random walk*, Social Networks, 2005- Elsevier.

[11] László Lovász – *Random Walks on Graphs: A Survey*, Tech. Report Dept. Computer Science Yale Univ., New Haven, Conn.