University of Nevada, Reno

Ad-hoc Limited Scale-Free Models for Unstructured Peer-to-Peer Networks

A thesis submitted in partial fulfillment of the

requirements for the degree of Master of Science in

Computer Science

by

Durgesh R. Kumari

Dr. Murat Yuksel/Thesis Advisor

December, 2009

UNIVERSITY
OF NEVADA **THE GRADUATE SCHOOL**
RENO

We recommend that the thesis
prepared under our supervision by

DURGESH KUMARI

entitled

Ad-hoc Limited Scale-Free Models for Unstructured Peer-to-Peer Networks

be accepted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

Murat Yuksel, Ph.D., Advisor
Eelke Folmer, Ph.D., Committee Member
Banmali Rawat, Ph.D., Graduate School Representative
Marsha H. Read, Ph. D., Associate Dean, Graduate School

December, 2009

To my parents

Chandra Mauli Deo & Geeta Devi

Acknowledgments

I would like to take an opportunity to acknowledge all the people whose endless support and guidance helped me throughout my study and research.

First of all, I would like to thank my advisor Dr. Murat Yuksel for his enthusiastic supervision and guidance during my research work. I would also like to thank Dr. Hasan Guclu for his inputs and technical discussions during our meeting sessions. Dr. Eelke Folmer and Dr. Monica Nicolescu are equally thanked for the knowledge, they shared during my classes.

I want to show my gratitude to my friend Vandana Jha for being my surrogate family for past many years and for her continued moral and emotional support. I am also grateful to Manjari Sapre for always being there as a good friend and a source of inspiration. Sumant Jha, Hrishikesh Kulkarni and all my friends are thanked for their support, care and attention.

Finally, I am forever indebted to my parents, C.M.Deo and Geeta Devi and my siblings, Vandana, Amrita, Shalini and Prince for their understanding, selfless love and patience.

Durgesh R. Kumari

University of Nevada, Reno

December 2009

Abstract

Ad-hoc behavior of peers is one of the most important characteristics of any peer-to-peer (P2P) network that affects the scalability, stability, reachability, and search efficiency of the network [17]. These ad-hoc peers cause churn by untimely leaving and joining the network. It is important for a P2P network to sustain against these churns. Preserving the above mentioned characteristics becomes more difficult in a distributed and potentially uncooperative network environment. In this thesis, we propose a model for unstructured P2P networks which defines protocols for joining and rewiring process for a peer. Our model uses only local information for a peer for joining and rewiring, without needing to store the global state information [17]. We impose a hard cutoff on the network which limits the number of links a peer can have. This makes sure that there are no super hubs in the network and the load is distributed evenly among peers. We also investigate the effects of such hard cutoffs on degree distribution and search efficiency of the network [17].

Contents

List of Figures

Chapter 1

Introduction

The capabilities of a network to be stable and scalable play a vital role in determining its popularity and practicality. In order for any peer-to-peer network to be scalable it should respect the ad-hoc behavior of a node. It must have the sustainability to survive the churn caused by nodes joining or leaving the network. As the network grows larger, maintaining topological structure becomes difficult. In such scenarios, unstructured peer-to-peer networking provides a practical and feasible solution. However, it is being observed by researchers that the best search efficiency in any practical networks is achievable in cases when the topology of the network follows the scale-free (power-law) characteristics, which can be estimated as $O(\ln \ln N)$ [18]. Scale-free networks have one of the disadvantages that they impose high load on very few number of hub nodes. As we see in any unstructured peer-to-peer network, peers exhibit uncooperative behavior by not willing to take extra responsibility of preserving the structure and storing the information about overlay topology. Therefore, keeping these characteristics of peers in mind, we can state that the topology generation protocols cannot completely depend on methods which require or expect a peer to cooperate in generating the topology [18].

After observing this uncooperative behavior of the peers in the network, we impose

hard cutoffs on the degree of the peer, which results in making the whole network, a limited network [17]. We can define degree of a peer as the number of overlay links a peer has to other neighboring peers in the network. Limiting the network by imposing the hard cutoffs can have significant effects on the search capabilities of the network, we would discuss this in more details in the later part of the thesis.

Another major problem that needs to be addressed is how to make the overlay topology without using the global information. There are many ways which are discovered for generating and maintaining overlay topology using global information. However, assuming that peers in the network have global information about the whole network would be far beyond from the practical scenarios. Considering the fact that a peer cannot store a huge amount of state information, we use *local heuristics* to generate the overlay network. A good topology generation algorithm must provide a set of simple steps that needs to be done by a peer when it wants to join the network and similarly a set of simple operations that peers in the network should perform when one or more of their neighbors leave the network. These set of steps/operations must not be complicated and should not impact the search capabilities of the network [17].

This thesis first explores the construction and maintenance of scale-free overlay topologies by using local parameters available at each peer without having to probe for global information. In order to give our method a practical perspective, we limit the number of neighbors of a node to a predefined value indicated by hard cutoff. The ad-hoc nature of peers are exploited by rewiring the network upon failure or departure of a peer. In the later part of the thesis we examine the impact of imposing hard cutoffs and leveraging rewiring on search performance.

1.1 Contributions and Major Findings

As we stated earlier that the ad-hoc behavior of the peers impact the search performance of the network. We consider various parameters in our model in order to demonstrate the

relationship between peers behavior and search efficiency. The parameters are as follows– μ, defines the probability of a peer to leave the network, τ_j, defines the radius of local information about the peers available to the joining peer, and τ_l, defines the radius of local information available to the peers whose neighbor just left the network. We also define *hard cutoff*, k_c, as the maximum number of links a peer can have to the other peers at any point of time in the network. Our contributions include:

- *Instructions for generating scale-free topologies-considering ad-hoc behavior of peers:* We propose a generic model for generating the scale-free topologies that can a generate network based on different amount of information available to peers at the time of join and leave. We also provide a method for varying the ad-hocness of the network and for balancing the information needed to be stored at each peer in the network for join and leave. We also vary the frequency of peers leaving the network to study the realistic behavior of the network.

- *Impact of the imposed parameters on search efficiency of the network:* Through extensive simulations, we studied the performance of various search algorithms like Flooding (FL), Normalized Flooding (NF), and Random Walk (RW) on the different networks generated by our algorithm by varying the network parameters like μ, τ_j, τ_l, and k_c.

- *Guidelines for Rewiring the network after a peer has left:* Our research proposed several guidelines that can be performed by the neighbors of a leaving peer in an unstructured peer-to-peer network. Rewiring of the network once a peer has left ensures the stability, reachability and search performance of the network.

Rest of the thesis is organized as follows. Chapter 2 focuses on literature survey of the relevant researches. First we study several P2P network architectures followed by popular search techniques used for finding the desired content in the network. We also study some of the peer-to-peer network involving topology adaption for improving the search efficiency. Next, we define more specific features of scale-free networks in chapter 3. Networks dynamics

and design of our limited scale-free model are discussed in chapter 4. We also present our algorithm for growing a scale-free network considering the facts that at any point a node can join or leave the network, in this chapter. Chapter 5 confers the results and behavior of our model followed chapter 6 which concludes the thesis and discusses future goals.

Chapter 2

Literature Survey

Peer-to-peer (P2P) networking is a vast area of research. There has been a tremendous growth in this area of research in last five years. P2P networks exhibit three important characteristics: *Self-organization, symmetric communication* and *distributed control* [27]. As the name indicates, a self-organizing P2P network configures itself automatically on the arrival of new peers to the network and also adapts to any changes in the network structure like peers leaving the network. By virtue of this behavior, P2P networks survive the churn produced by its peers. P2P networks show symmetric communication, which is nothing but the fact that each peer in the network is treated equally. All the peers act as both client and server. Peers act as clients while asking for information, act as servers while providing the information and act as routers while forwarding search requests to other peers. P2P networks have no centralized control. It means that there is no single control point or server in the network for the regulations and processing of network. The body of P2P research can be divided into several categories– topology generation, storage, search, security and applications. Our work relates to topology generation and search of items in P2P networks, and thus we cover these two categories of P2P literature.

Based on overlay topology P2P network architecture has three forms [27]: centralized, distributed but structured and decentralized and unstructured.

Centralized P2P networks:

Centralized P2P networks, have a central server which stores the information which are needed to ease the searching process. For example in case of Napster [31], it stored the directory, listing all files shared by the participating peers. This directory was constantly updated upon arrival and departure of the peers [27]. Whenever a peer wanted to request a file, the search query was first routed to this central Napster server and where a look up was performed and then a list of the hosts, sharing the requested files, was sent back to the requester and then the requester chose a desired hosting peer and started downloading the file directly from that remote host peer [27]. People might confuse the centralized P2P network by client-server model, but these networks are not the same. In centralized P2P network, the central server does not store the actual contents. It only stores the directory containing the information about the peers sharing the information. Due to availability of this centrally located shared information centralized P2P network provides the best search performance but the main limitation of such P2P networks is that they are unscalable in the real network scenarios. Centralized P2P networks also have a single point of failure [12].

Decentralized but structured P2P networks:

As the name indicates the decentralized P2P networks do not have a centralized server. But they do define a structure in which the peer should join the network [27]. When a peer joins the network there are a set of operations performed to determine the exact connections that this new peer is allowed to make. There is a strict restriction on which peer can connect to which other peers in the network. However, there is no central server which hosts the directory of shared files and contents but there is slight control over placement of the files on the remote hosts or peers. Based on how much control the network has over storing the files, the decentralized structured P2P networks can have two forms– loosely structured P2P networks and highly structured P2P networks [27]. Freenet [15] is an example of structured P2P networks.

Decentralized and unstructured P2P networks:

Decentralized and unstructured P2P networks are the networks which neither have a central server nor a high control over the structure of the network or the place of the contents in the network. Such network inherit a kind of randomness in its behavior. The contents are stored randomly on the peers. Gnutella [5] is an unstructured P2P network. As the file placement is random in these networks, they have to rely on the search algorithms like blind flooding and random walk for finding the desired content in the network [25, 36]. Unstructured P2P systems are most commonly used in today's internet.

Despite of popularity of such unstructured P2P networks in the file sharing application, they are still inefficient when it comes to search performance. Request for searching the contents in the network can place a high load on the network traffic of P2P networks [27]. The condition becomes worst, when P2P networks start relying on blind flooding. A single query can generate a huge number of message in the network increasing the consumption of network resources to an extent which can limit the scalability and efficiency of the network. It becomes very important to choose the right search method for the system. However, picking a right search technique sometimes becomes a compulsion rather than a choice, due to the architectural properties and boundaries of the P2P network. Following subsection elaborate the most popular search techniques applied on P2P systems [27].

2.1 Search Techniques in P2P Networks

Depending on the architecture of P2P networks, the search techniques can be divided into two major categories: (i) Search techniques based on unstructured P2P networks and, (ii) Search techniques based on structured P2P networks. Considering the nature of our research, this section is focussed on search techniques built on unstructured P2P networks. Flooding, Expanding ring, Normalized Flooding, Random Walk, and Similar Content Search are some of the popular search methods built on top of unstructured P2P

networks.

- *Flooding:* This scheme is based on the simple broadcasting. This search technique is specially supported by unstructured peer-to-peer networks like Gnutella [21], [20], [12]. The query issued by the client peer is broadcasted to all peer via multi-hop flooding. Each node which is visited in the way of flooding evaluates the query locally and tries to find a match of the requested content in the list of items it has to share. If there is a match found then the response is returned to the requesting node or else it just broadcasts the query. Due to its nature, flooding provides high degree of fault tolerant. Figure 2.1-(a) shows the flooding. Time-to-Live (TTL) can be associated with each message to control the flooding radius.

  Gnutella typically sets this field to limit the flooding to 7-10 hops away [21] [20]. However, deciding the value of TTL is one the difficult task. If the value of TTL is too small, it reduces the network traffic and load on peers but at the same time reduces the success hit rate, which in turn affects the search performance. On the other hand large TTL values increase the success rate but loads the network with flood of messages. It can be a tough task to predict the TTL values, as there are different TTLs needed for different network topologies and differ for different replication ratios. Unfortunately, as its hard to determine the duplication of contents in the network, we are forced to set high value for TTL to ensure success of the query [27].

  One of the good things about this approach is that it supports arbitrarily complex queries and it does not impose any limitation on the placement of contents in the network or the topological structure. However, in order to support the flooding search algorithm each peer must connect to one other peer in the network so that there are no disconnected peers in the network, which limits scalability of such systems [34] [20]. Flooding also causes the duplication of the search request queries. Due to the nature of blind flooding a peer might receive multiple copies of same search query. Sometimes too many duplicate queries might overload the network traffic. However, there have

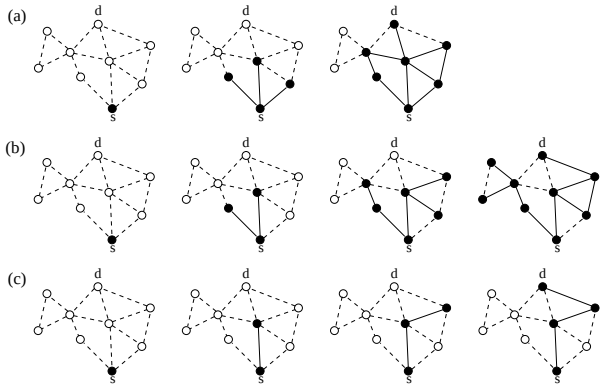


Figure 2.1: Search strategies: (a) Flooding (b) Normalized flooding (c) Random walk

been many approaches proposed to detect and prevent the duplication of the search query. But despite of these mechanisms applied it's hard to build a full proof system in case of flooding that will not generate a single duplicate message. These duplicate messages can worsen the situation when the query hit one of the hub peer in the network [27].

- *Expanding Ring:* To address the problem of selection of TTL, a new technique has been introduced which is known as Expanding Ring [27]. This method starts flooding the query with a small TTL in the beginning, if the search is not successful the TTL is increased before starting the next flood [27]. Researchers have shown the results by simulations that despite the successive retries, expanding ring still reduces message overhead significantly compared with regular flooding with a fixed TTL. The savings are obtained across all query and replication distributions [27]. Expanding ring achieves the savings at the expense of slight increase in the delays to find the object. Though expanding ring solves the TTL selection problem, it does not solve the problem of duplication of messages in the network and it continues to add to the overhead to the network.

- *Normalized Flooding:* Simple blind flooding mechanisms depend on the network structure. An irregular graph causes a significant deterioration in the the search perfor-

mance by increasing the number of messages floating in the system. There can be a sudden increase in the number of flooded messages, if flooding process reaches a node with high degree. Hence, a new technique has been introduce to subsidize this problem, known as Normalized Flooding [16]. Figure 2.1-(b) explains the working of this search technique. The main idea behind this search technique is to limit the number for same search request propagating in the network. First, it calculates the minimum degree of the network (d_{min}) and then forwards the request based on this minimum degree. If a peer with a degree equal or less than d_{min} receives the search request then it first processes the request to see if it has the requested contents, if it has the requested content then it replies back to the sender with the message specifying that there was a hit otherwise it forwards the request to all its neighbors.

Similarly, if a peer with degree more than d_{min} receives the search request then if there was no hit then it forwards the query on only d_{min} neighbors randomly chosen form the set of its neighbors except the one which forwarded the query.

- *Random Walk:* Random walk avoids the blind flooding mechanism of search. When a peer receives the search request then it randomly picks one neighbor from its set of neighbors and forwards the query to it. But this might result in delayed search success. To fasten the process it increases the number of independent walkers. To start with the starting peer can send the request to k randomly selected peers and then each of those k peers can perform their own independent random walks [27]. Random Walk addresses the scalability issues caused by normal blind flooding. Figure 2.1-(c) shows the random walk search. Two techniques have been proposed to terminate random walks: TTL (time-to-live) and checking. Inducing TTL terminates the search after certain number of hops, while checking means a walker periodically checks with the query originator before walking to the next node [27], [37].

- *Similar Content Based Search:* This search technique is popular in the P2P networks

like Gnutella [21] [12] [24]. The basic idea behind similar content based search is to organize the contents hosted by the peers in similar content groups [10] [33]. These groups are formed on top of the unstructured P2P overlay topology. It is intuitive that the peers within a certain group tend to have similar contents and are tentative hits for same queries. As a result this search technique will guide the queries to the peers which are more likely to have answers to the queries [10], thereby avoiding significant flooding.

2.2 Topology Adaption for Improving Search Efficiency

Previous section of this chapter discussed few popular search techniques built on top of existing P2P network topologies. These search techniques did not take into consideration the dynamic behavior (i.e. ad-hocness) of peers. One may doubt the efficiency of such search techniques for practical P2P networks where peers join and/or leave without giving any hints. In such scenarios, we must look at the search methods which use topology adaption to cope with the ad-hocness of the peers. Supernode-based P2P networks are good examples for such kind of P2P networks. In supernode-based P2P architecture, different peers are treated differently. Peers make unequal contribution to the network depending on their capabilities. The system workloads are neither uniformly distributed on all peers nor concentrated on few centralized serving peer but unevenly distributed among peers in the network. Supernodes take most of the system workload and more like servers and ordinary nodes take small portion of the workload and act as clients. However, unlike client-server model there is no strict separation between server and clients. Supernodes are chosen from the ordinary nodes based on certain criteria. These criteria differ from network to network. However, certain common criteria that can be applied are high connection bandwidth, good computational power and must be a stable peer (less join/leave frequency) [26]. The main problem in supernode-based P2P networks is defining a protocol for supernode selection. There have been several protocols introduced for addressing the supernode selection problem

in different networks like structured, unstructured and co-ordinate based [26].

This work is dedicated towards the unstructured P2P networks. The main focus of research in unstructured P2P networks have been developing a model where the peers do not need to store huge state information by keeping search efficiency optimal. The goal of this thesis is to develop a model for unstructured P2P networks which finds the right tradeoff between state complexity of peers and flooding based search efficiency [17]. In order to construct P2P overlay topology, participating peers need to store some state information. Our effort should be to minimize the state information needed by a peer in order to construct the overlay network. The minimal information that a peer must store is the list of its neighboring peers. Optionally, a peer can also store the *forwarding table* (also known as routing table in literature) for the searchable data items. We can also classify unstructured P2P networks into two categories based on state information stored by peers: (i) *per-data* unstructured P2P networks (i.e., peers maintain both the list of neighbor peers and the per-data forwarding table), and (ii) *non-per-data* unstructured P2P networks (i.e., peers maintain only the list of neighbor peers) [17].

Chapter 3

Growing Scale-Free Networks

3.1 Degree of a Node

In terms of graph theory or networks, *degree* of a node is defined as the number of connections or links the node has to the other nodes. In literature, degree of a node is also known as its connectivity. In a directed network or graph, a node can have two kind of degrees: *in-degree* defined by number of incoming connections and *out-degree* defined by number of outgoing connections.

Degree distribution is one of the most important factors to be considered for the study of networks, be theoretical or real. Degree distribution [3], denoted by $P(k)$, of a network is defined as the probability distribution of the degrees of nodes over whole network. $P(k)$ can be represented as the fraction of nodes in the network with degree k. For network if there are total n nodes and n_k of them have degree k, then $P(k) = n_k/n$.

3.2 The Power Law

After a careful analysis of networks, researchers ended up observing a very interesting behavior exhibited by most of the networks. Many natural and artificial systems such as the Internet [14], World Wide Web [4], scientific collaboration network [6], and e-mail

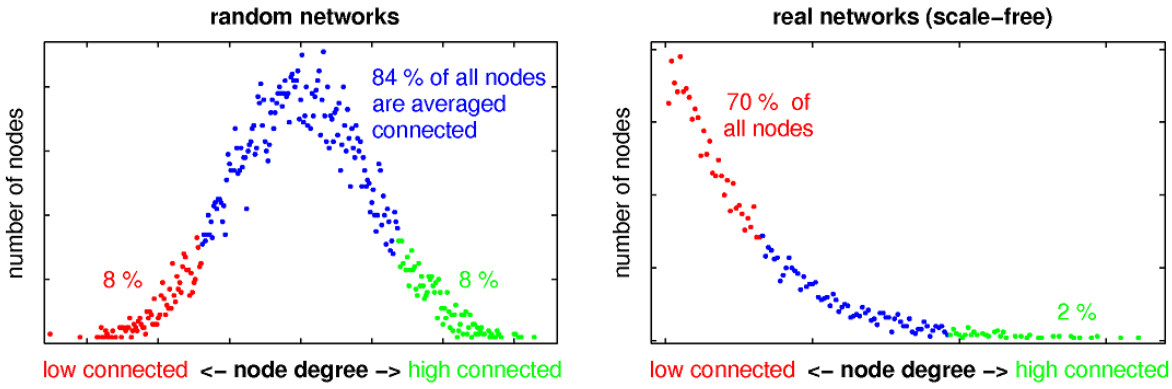


Figure 3.1: Power Law shown on random and real (scale-free) networks [32].

network [19] have power-law degree distributions [7]. These networks are also known as scale-free networks since their degree distributions are free of scale [7] (i.e. not a function of the number of network nodes N) and follow power-law distributions over many orders of magnitude. In simple words, we can state *power law* as the relationship between two variables such that one is proportional to the power of the other i.e. $y \propto x^a$, where x and y are the variables of interest and a is a constant [17]. In the field of networking this phenomenon can also be represented by probability of having nodes with k degrees as $P(k) \sim k^{-\gamma}$ where γ is usually between 2 and 3 [7]. Figure 3.1 shows power law on random and real (scale-free) networks.

3.3 Preferential Attachment (PA)

Scale-free networks grow by arrival of new nodes to the networks which connect to the already existing nodes in the network. Continuous addition of new nodes ensures the expansion of real-time scale-free networks. As we observe in a social network that a few handful of famous people tend to get more famous and known by the new people joining the society or as the"rich gets richer" philosophy, the scale-free networks also follow the same principle. So, whenever a a new node joins the network, it is more likely to be connected to a node which already has a large number of connections. Based on these unique behaviors of the scale-free networks, Barabási and Albert proposed a model for generating a network

Table 3.1: Scale-free network diameter behavior

Diameter d	Exponent γ	# of stubs m
$\ln \ln N$	(2,3)	≥ 1
$\ln N / \ln \ln N$	3	≥ 2
$\ln N$	3	1
$\ln N$	> 3	≥ 1

model which grew scale-freely and for which $P(k)$ follows a power law with γ=3. This model was given a name *preferential attachment* (PA) [7] and the resulting network was known as Barabási and Albert network [8].

In this thesis, we have used a simple version of PA model [7]. The model evolves by one node at a time and this new node is connected to m (number of stubs) different existing nodes with probability proportional to their degrees, i.e., P_i=$k_i / \sum_j k_j$ where k_i is the degree of the node i . The average degree per node in the resulting network is $2m$ and the minimum degree is m [17].

Randomness and PA characteristics make scale-free networks very robust against failure and attacks. As there are only few nodes with very large degrees so the probability of attacks on these nodes are small. On the other hand, attacking the low-degree satellite nodes does not do much harm to the network [17]. However, deliberate attacks targeted to the hubs which bare most of the traffic flows can shatter the network and severely damage the overall communication in the network. For the same reason the Internet is called "robust yet fragile" [13] or "Achilles' heel" [9].

3.4 The Cutoff

The term cutoff signifies the limit on maximum number of connections a node can have. It plays very important role in our model. Scale-free networks demonstrate one of the important characteristics i.e. the natural cutoff due to finite-size effects. Natural cutoff,

represented by k_{nc}, is the value of the degree above which one expects to find at most one vertex [17], i.e.

$$N \int_{k_{nc}}^{\infty} P(k)dk \sim 1 \ . \tag{3.1}$$

By using the degree distribution for the scale-free network and the exact form of probability distribution (i.e., $P(k)=(\gamma-1)m^{\gamma-1}/k^{\gamma}$), one obtains

$$k_{nc}(N) \sim mN^{1/(\gamma-1)}, \tag{3.2}$$

which is known as the *natural* cutoff of the network [17]. The scaling of the natural cutoff can also be calculated by using the extreme-value theory [28]. For the scale-free networks generated by PA model (γ=3) the natural cutoff becomes

$$k_{nc}(N) \sim m\sqrt{N}. \tag{3.3}$$

3.5 Preferential Attachment with Hard cutoffs

In theory, it is simple to define a natural cutoff for a scale-free network. But in practice the natural cutoff may not be always attainable [17]. There are several reasons for that. The most important reasons are the technical limitations on a node. The networks also put constraints on the number of connections a node can have. This holds special importance in case P2P networks. P2P networks obligate a node to have connections to too many other nodes. This drive us to impose an artificial cutoff on the degree of a node in P2P network. This artificial cutoff is referred as *hard cutoff*, denoted by k_c, in rest of the thesis [17].

For the implementation purposes, we did not allow a node in the network to have the links more than the fixed hard cutoff value during the attachment process [17]. This helped us in generating a network which was scale-free nature but there was not only few overloaded peers, instead there were many peers with degree equivalent to hard cutoff. The degree distribution of scale-free networks generated by PA model with different hard cutoff

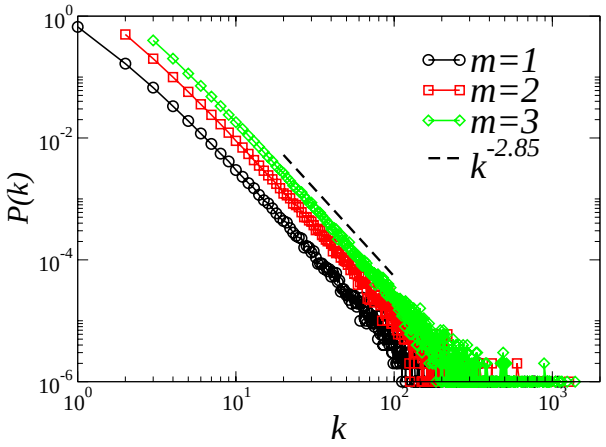
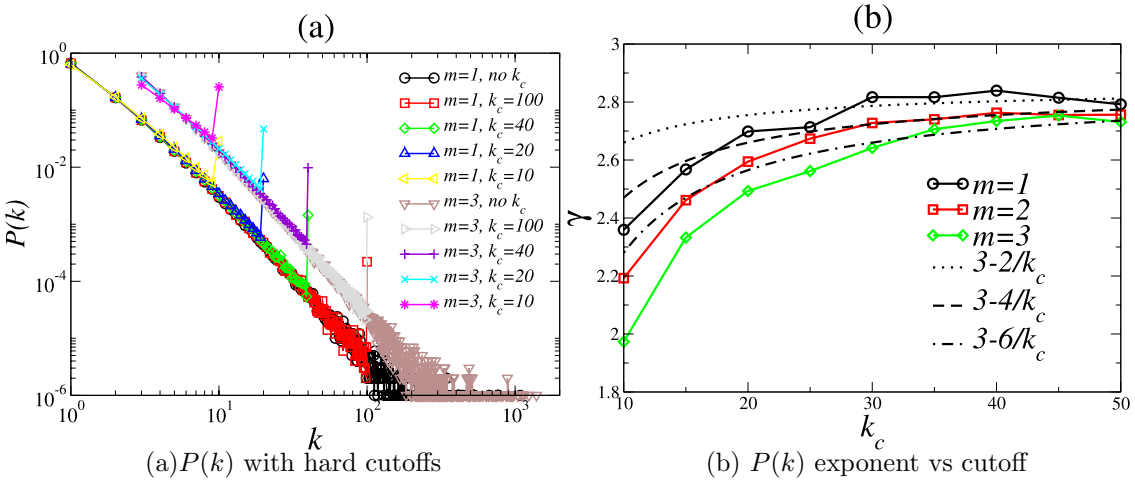


Figure 3.2: Degree distribution of PA model $P(k)$ without cutoffs.



(a)$P(k)$ with hard cutoffs (b) $P(k)$ exponent vs cutoff

Figure 3.3: Degree distribution of PA model

values are shown in figure 3.3.

It can be inferred from the figure that PA model with cutoff is slightly different than that of PA without a cutoff in terms of exponent and an accumulation of nodes with degree equal to hard cutoff [17].

PA model, in its original form, has a degree distribution exponent γ=3 for very large networks. However, when a hard cutoff is imposed it is observed that the absolute value of degree distribution exponent decreases [17, 18].

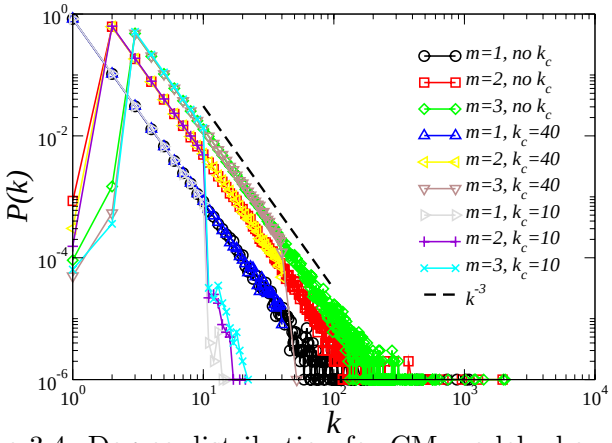


Figure 3.4: Degree distribution for CM model when $\gamma = 3$.

3.6 Configuration Model (CM)

In previous section, we have seen that in PA model shows lower degree distribution as the hard cutoff reduces. It was also required to reduce the spikes at hard cutoff value in figure 3.3(a) and to obtain a smooth power-law degree distribution. All these brought into picture the new models for topology generation like modified PA models such as nonlinear preferential attachment [22, 23], dynamic edge-rewiring [8] and fitness modes [1, 2]. Here, we choose to discuss configuration model with pre-defined degree distribution to generate a static scale-free network [29]. CM [11, 18, 29, 30] is a method to generated uncorrelated random networks with the given degree distribution. It also causes some very negligible number of nodes in the network to have degrees less than the fixed minimum degree value (denoted by m) or even nodes with degree zero are observed in figure 3.4 [18].

3.7 Growing Scale-Free Networks Using Local Heuristics

As observed in previous sections the PA model uses global information for the topology generation. In this model, when a new node tries to join the network it makes random attempt to connect to some of the existing nodes with the probability depending on the degree of the existing node. To implement such kind of model in P2P network or, for that matter any distributed real time network, the new node has to have the degree information

of all the nodes, in other words it requires the global topology information for joining the network. But it were only best to implement such model in reality if it had not required the global information in order to construct the topology. In reality it is impossible to maintain high state information at each node, moreover when the new nodes are continuously joining in and old nodes are frequently leaving the network. Such global topology information was also needed in CM method as well.

The need arises to have a model which at least did not require the global information for the construction of the topology. We must think of a model which requires only local information in order to form the topology. Of course, the cost of using only the local information is expected to be paid by the loss of scale-freeness (or, any other desired characteristics) of the whole overlay topology. The main loss would be the degradation in search efficiency. In this section, we present two practical methods HAPA and DAPA, which require only local state information to construct the network topology [17, 18].

3.7.1 Hop-and-Attempt Preferential Attachment (HAPA)

In HAPA, when a peer wants to join the network, it first randomly picks a peer from the existing network and make a link to it. Now, as the new node is connected to this randomly selected node, it can obtain the information about the peers which are neighbor to this peer. Similarly, now the new peer knows neighbors' neighbor too. Hence perform the same steps again and again until it has made the desired number connections i.e until the degree of this new reaches m [18]. Covering large number of peers while joining the network gives the new node a better probability to reach the hubs as it has been observed that in the scale-free networks the high degree nodes are only couple of hops away from any peer [18]. The old peers (the peers which joined the network initial) attract the new peers as these old peers are popular and have high probability that a new peer will join them. These super hubs have high degrees which of the order of network size. It is obvious from the procedure that it makes a a star-like topology if the network is not limited by cutoff [18].

3.7.2 Discover-and-Attempt Preferential Attachment (DAPA)

DAPA model first assumes that there is an existing network called *substrate network* with a pre-defined and pre-constructed topology at hand [18]. Then, we use preferential attachment technique to generate the overlay network, for that we first collect the set of peers which are reachable from the horizon of the new peer, this horizon is nothing but can be treated as a *local time-to-live* [18]. DAPA is used by several unstructured P2P networks like Gnutella [18] for topology generation.

Chapter 4

Growing Scale-Free Networks Under Churn

4.1 Network Dynamics

Before we get more focused towards the specifics of our model, it is important to discuss the factors that motivated us to design our model the way it is. Network be it social, artificial or real, show a dynamic behavior. It is hard to predict the exact behavior of the network as it grows over time. As discussed before, most of the network systems come across with frequent churn caused by leaving and joining nodes. A new node can join in at any time to the network or it can be just an old node coming up again after being away for sometime. Similarly, at any point of time an existing node in the network can decide to leave. There can be various reasons for a node to leave the network starting from a simple closing of the browser to some technical reasons. It is well known that most of the Internet users are opportunists and selfish by nature. It is not about the human nature that we are referring to, but it is about the purpose of a user for using the Internet and all they care about is how to solve that purpose, may be efficiently but that is a later thought. The moment the purpose is solved they might leave the network without any notification. It is not only hard to predict when a node is about to leave and join but it is also difficult to have control over the leave and join. There have been various researches performed in order to estimate the average time a node stays in the P2P network. It was shown in [9] that for a

scale-free network of size 10,000, the failure rate (i.e. nodes leaving the network) was 18%, the biggest connected component holds 8,000 nodes, whereas under the same conditions a random network can survive these failures by the biggest connected component of size 100.

The main concerns that this dynamic behavior of network gives birth to are [17], (a) When a new peer joins, how should it construct its list of neighbors? (b) When a new peer leaves, how should its neighbors rewire themselves to the network? It is very important to address these concerns in order to ensure the robustness and search efficiency of the any P2P network. As discussed earlier, there have been various protocols and algorithms suggested for diluting the effect of churn created by joining and leaving nodes, but most of them use global information about the network in order to construct and rewire the network. That is how our model proposed in this thesis stands out, because we construct and rewire the network based on local heuristics as discussed in next section of this chapter.

4.2 Growing Network using Local Heuristics

4.2.1 Bootstrapping

Our model considers a substrate network of $m+1$ fully connected nodes to start with. The possible values for m can be from 1 to 3. Algorithm 1 in the next section presents the pseudo code in order to explain the growth of the overlay topology. At each time step, a new node i is being added to the existing network by calling function *join()*. The new node first connects to a randomly selected node from the existing nodes and then it targets to connect to $m-1$ more nodes. Each of these $m-1$ nodes is randomly selected from the nodes in the radius of τ_j of its latest neighbor. Slowly the network grows from $m+1$ nodes to the targeted number of nodes (N_{target} as referred by the Algo 1) [17].

4.2.2 Sustaining against Churn

In order to simulate the effect of churn in our model, at each time step, we add a new node to the network and delete one randomly chosen node from the network. The new node can have m possible links with the existing nodes and the deletion of node is carried out with

the probability of μ. The nodes do not have global information about the whole network. So, the nodes can only choose from a subset of the network (horizon) in order to make a link. The parameters τ_j and τ_l are TTL values used by the nodes at the time of joining and leaving respectively [17]. The parameter τ_j limits the number nodes that are visible to the new nodes and τ_l is the parameter which helps in rewiring after a node has left the network by letting the abandoned nodes to randomly choose between nodes visible in τ_l radius and making new links [17].

The Join

A newly added node, first, selects a random node from the existing network and connects to it. Then, with the provided value of τ_j (may vary from 0 to 3), it constructs a set of nodes that are τ_j or less hops away from the node it has recently connected to. The set does not contain the nodes which already have degree equal to hard cutoff. Next, it randomly selects a node from this set and connects to it, with a probability proportional to its degree. This probability is normalized by the total degree of the nodes in the set. Subsequently, it selects other nodes from the set and tries to connect to it with the probability proportional to its degree, until its degree reaches m. If there is no nodes left in the set and the degree of the new node is still less than m, then it selects another random node and continue the same process until it fills all its stubs (m) [17].

The Leave

Our model also considers the churn created by the leaving nodes and does the rewiring process in order to maintain the reachability and robustness of the network. In order to simulate this in our model, we randomly select a node with a probability of μ and delete it from the network. Then the immediate neighbors of deleted node select a node randomly from a set of existing nodes reachable at τ_l hops or less form the deleted node and connect to it by applying rules of preferential attachment. Similar to join process, the leave process also does not include the nodes with the degree equal to cutoff, in the set of nodes that can

accept the connections from neighbors of the deleted node.

4.3 The Algorithm

4.3.1 The Parameters

In order to understand the Algorithm 1, first it is important to understand the involved parameters [17].

- *Ad-hocness of the nodes*, denoted by μ, gives the probability with which the nodes are deleted (i.e. nodes left) from the network.
- *Hard cutoff*, denoted by k_c. It defines the maximum limit on the number of links a node can have.
- *TTL at join*, denoted by τ_j, which specifies the horizon of the nodes in the network, a new node can connect to.
- *TTL at leave*, denoted by τ_l,which specifies the horizon of the nodes in the network, neighbors of the deleted node can connect to.
- *Number of stubs*, denoted by m, is minimum number of links a node should have.

Algorithm 1 Network growth using parameterized join and leave processes

//Global Variables and Functions
m - minimum degree
μ - probability of a node to leave the network
N - the maximum node ID of the existing network (the minimum node ID is 0)
G - graph of the existing network of M links and N nodes
PreferentialAttachment(G_1, G_2) *- a function that performs Preferential Attachment to G_1 by using the nodes in G_2, returns the number of successful new links*

// Join process of node i
void **Join**(i, τ_j)

1: N++
2: $numoflinks \leftarrow 0$
3: **while** $numoflinks < m$ **do**
4: $N_{rand} \leftarrow$ Randomize(1,N) {Pick a random node from the existing network}
5: $myG \leftarrow get\_subgraph(N_{rand}, \tau_j)$ {Get the subgraph including neighbor nodes of N_{rand} up to τ_j hops away}
6: $numoflinks$ += PreferentialAttachment(G, myG)
7: **end while**

//Leave process of node i
void **Leave**(i, τ_l)

1: $myG \leftarrow get\_subgraph(N_{rand}, \tau_l)$ {Get the subgraph including neighbor nodes of N_{rand} up to τ_l hops away}
2: $remove(N_{rand})$ {Delete N_{rand} from the existing network}
3: $N = N - 1$
4: PreferentialAttachment(G,myG)

// Growth process of a network with N_{target} nodes, parameterized with τ_j and τ_l
void **Grow**(N_{target}, τ_j, τ_l)

1: **for** i=m+1; i<N_{target}; i++ **do**
2: Join(N,τ_j)
3: $num \leftarrow$ Random(0,1)
4: **if** $N == N_{target}$ **then**
5: break;
6: **end if**
7: **if** $num < \mu$ **then**
8: $N_{del} \leftarrow$ Randomize(1,N)
9: Leave(N_{del}, τ_l)
10: **end if**
11: **end for**

Chapter 5

Simulations and Results

We simulated our theoretical model using programming language C and perl scripting in Unix environment. We ran our simulations on the University of Nevada, Reno research grid for faster results. We constructed different topologies by varying the network parameters like μ, τ_l, τ_j, k_c and m (discussed in previous chapter). We have five main parameters for our model namely, (i) churn, $0 < \mu < 1$, (ii) available information during join, $\tau_j \geq 0$, (iii) available information during leave, $\tau_l \geq 0$, (iv) hard cutoff, $k_c > 1$, and (v) minimum degree (number of stubs), m ≥ 1. Table 5.1 lists all the parameters with their range. By varying these parameters we generated topologies with 10,000 nodes. We used different values of k_c from 10 to 100 (mostly just a few in this range). In addition, we looked at the case with natural cutoff or no hard cutoff. We varied the values of τ_j and τ_l from 0 to 3. The minimum degree or number of stubs were varied from 1 to 3. We studied smaller values of churn from 0.0 to 0.3, reflecting no churn to 30% churn, respectively. We performed 5 realizations of our results.

5.1 Search Algorithms

5.1.1 Flooding

As discussed in chapter 2, flooding is the most common search algorithm. The interested node initiates the search request for the desired content by broadcasting the search query to all its neighbors. If any of the neighbors have the requested content then it responses back

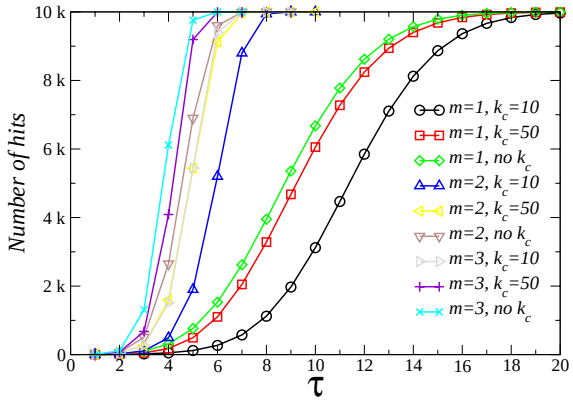


Figure 5.1: Flooding (FL) performance over topologies generated with m=3 and no churn.

to the source node, otherwise they forward the query to their neighbors excluding the source node and so on. The result for flooding is shown figure 5.1 shows the search performance over the topologies generated with $m = 3$ and considering no churn in the networks [17].

5.1.2 Normalized Flooding

In normalized flooding minimum degree of the node m holds an important place. As a node initiates the search query for the desired content, when it is received by a node with degree m, it forwards the request to its neighbors excluding the node which forwarded the request in previous step. However, when a node with degree higher than m receives the search query, it chooses m random nodes from its set of neighbors excluding the one which forwarded the message in previous step and forwards the search request to those neighbors. We performed normalized flooding search on topologies generated by tweaking the network parameters as shown in figure 5.6 [17].

5.1.3 Random Walk

Random walk or multiple random walks have been used to achieve better search granularity than normalized flooding. In random walk, the node receiving the search request forwards it to a randomly chosen neighbor excluding the one which sent the request. Then this randomly chosen neighbor again chooses one neighbor randomly from the set of its neighbors and forwards the request and so on. This process is repeated until the destination

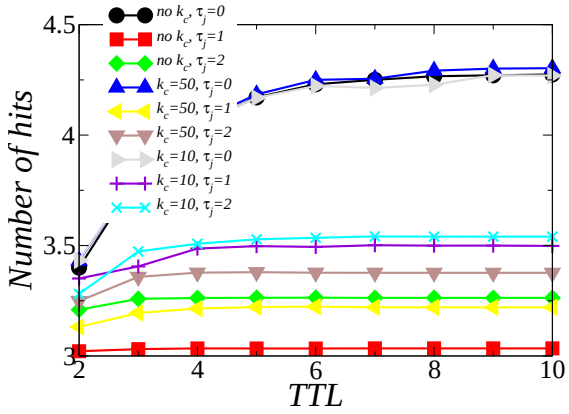


Figure 5.2: Performance of Random Walk over topologies with m=1 and no churn.

Table 5.1: Parameters of our topology construction framework

Symbol	Parameter Description	Range
μ	Ad-hocness of the nodes	[0,1)
τ_j	Available information at join	≥ 0
τ_l	Available information at leave	≥ 0
k_c	Hard cutoff	≥ 1
m	Minimum degree (# of stubs)	≥ 1

node is reached or the total number of hops becomes equal to TTL. Figure 5.2 shows the effectiveness of random walk search on certain chosen topologies [17].

5.2 Results

As the results of our simulation, we studied the effects of the network parameters on degree distribution, search performance and clustering coefficient of the network which are discussed in the rest of the chapter.

5.2.1 Effects on Degree Distribution

Figures 5.3 and 5.4 shows the behavior of degree distribution. Figure 5.3 shows the degree distributions of the network, when the value of $\mu = 0$, which indicates no churn. In simple words, there was no node leaving the network. Similarly, figure 5.4 shows the degree distributions when the nodes were baring the ad-hoc behavior with $\mu = 0.30$. There was a

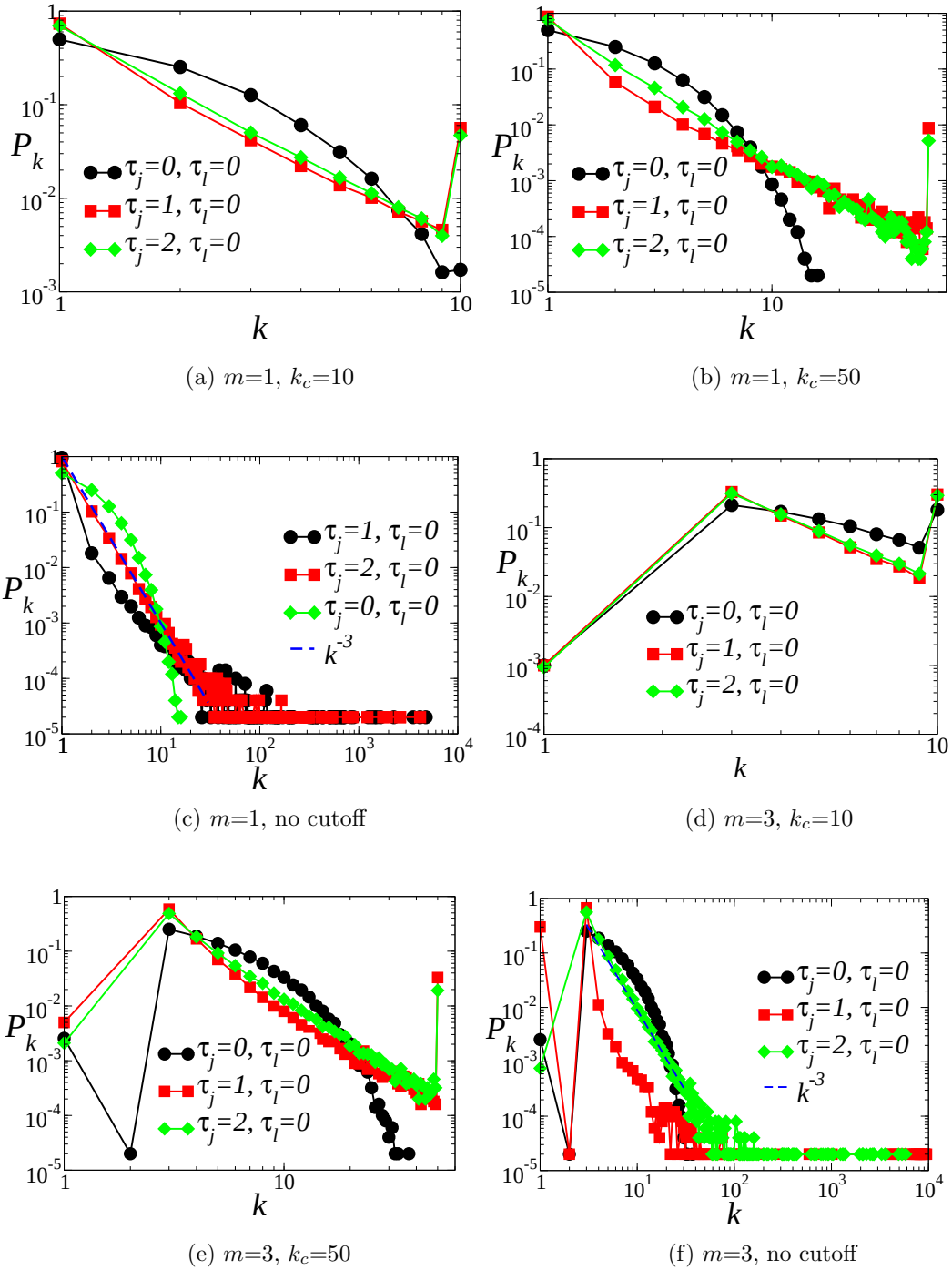
probability that 30% of the nodes will leave the network eventually.

It is evident that having global information about the topology during join and leave, helps generating better scale-free network with lower power-law exponent. Figure 3.2 shows this behavior very well. This graph shows the degree distributions with no cutoff, which is one of the ideal cases. Our goal is to generate a degree distribution which is close to the plots shown in this figure. As we analyze the behavior of degree distribution on different values of the parameters, we observe the shifts from an exponential one to a power-law one as τ_j increases from 1 to 2, as shown in figure 5.3 and 5.4, which means as there are more and more topological information available to a node while making new links the behavior of the network tends to be more power law [17]. It is also interesting to note the effect of m (i.e. the minimum degree). For both $m = 1$ and $m = 3$, we observe the similar trends of shift from exponential to power law with increasing value of m, though larger values of m makes the shift a little less apparent. We also observed the effect of hard cutoff, k_c. The hard cutoff only binds very large hubs to the cutoff without affecting the transition from exponential to power law [17].

We also studied the effects of τ_l which is shown in figure 5.4. It was interesting to note that τ_l has more significant in shifting degree distribution from exponential to power law than τ_j. This effect was more apparent for smaller values of cutoffs.

5.2.2 Effects on Search Efficiency

We analyzed the search performance of our model based on three search algorithms, flooding, normalized flooding and random walk. In flooding, the most important parameter when there is no deletion, is hard cutoff, which affects the search performance of flooding [17]. The cutoff determines, the number distinct nodes that can be reached for a particular search query, this effect can be seen well in figure 5.1. However, τ_j is also an important parameter responsible for changing the network's behavior from exponential to power law. It was also observed that increasing the value of minimum degree, improves the search performance,



(a) m=1, k_c=10

(b) m=1, k_c=50

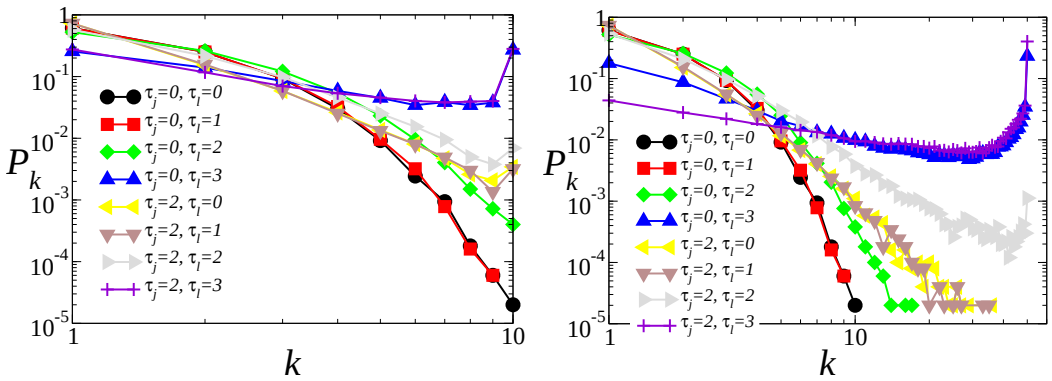
(c) m=1, no cutoff

(d) m=3, k_c=10

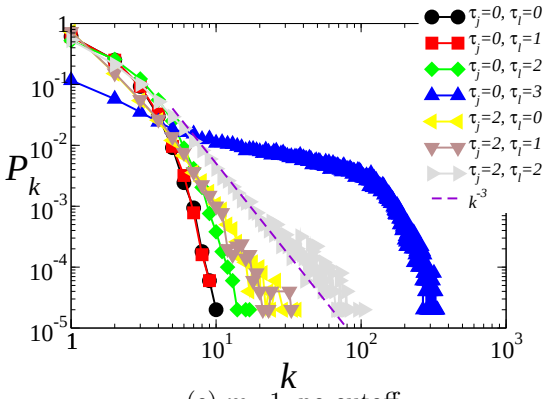
(e) m=3, k_c=50

(f) m=3, no cutoff

Figure 5.3: Degree distributions when there is no churn (i.e., μ=0): $P(k)$ for various networks generated by our framework for varying τ_j.



(a) m=1, k_c=10 (b) m=1, k_c=50



(c) m=1, no cutoff

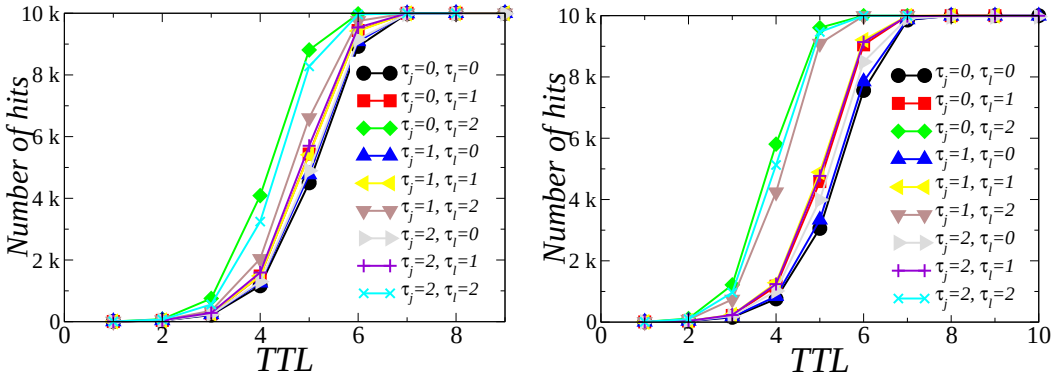
Figure 5.4: Degree distributions over ad-hoc nodes (i.e., μ=0.3): $P(k)$ for various networks generated by our framework for varying τ_j and τ_l.

it is clearly shown in figure 5.5. The churn also plays an important role in affecting the search performance [17]. Negative effect of high churn (i.e too many nodes leaving) can be eliminated by increasing the available information in rewiring the network (i.e. lager values of τ_l). Similar effects can be observed in case of normalized flooding. Figure 5.6 shows the performance of normalized flooding. In some cases, even high churn (high μ value) yields good search performance provided with large value of τ_l, like in figure 5.6 (f) where churn is 20% and values of τ_j and τ_l are 2 [17].

The next search algorithm we analyzed is random walk. Figures 5.2 and 5.7 show the search performance of random walk over the different values of parameters. We observed that the random walk qualitatively shows a similar behavior as normalized flooding except random walk is more vulnerable to isolated clusters. Both random walk and normalized flooding results show that the best performance is obtained when the available information in leaving is high ($\tau_l = 2$) but low in joining ($\tau_j = 0$) [17]. This demonstrates an interesting finding that using more information for rejoining of the nodes of a deleted node is more important than that in joining new nodes. We also observe that the for a fixed τ_l increasing τ_j does not increase search efficiency but decreases [17]. We believe the mechanism of rewiring after node deletion works in a peculiar way. It makes the network better connected in terms of search performance if the new nodes are not using too much global information in joining. We also believe this should manifest itself when put in a theoretical framework combining the structure and search performance [17].

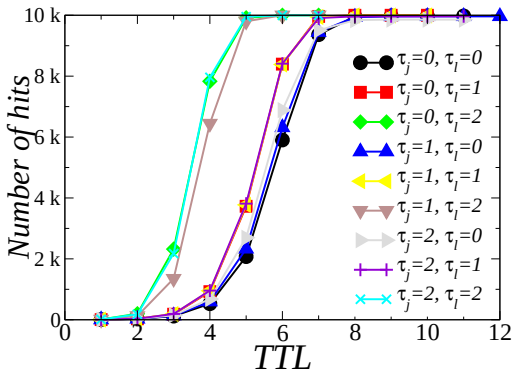
5.2.3 Effects on Clustering Co-efficient

In this part, we discuss about the effects of cutoff and churn on clustering coefficient. The clustering, also known as *transitivity* is the measure of how tightly connected the neighbors of a node are [17]. In other words transitivity means the presence of a high number of triangles in the network. An alternative definition of clustering coefficient which we also use in this paper is defined as follows [35]. Suppose c_i is the local clustering coefficient of



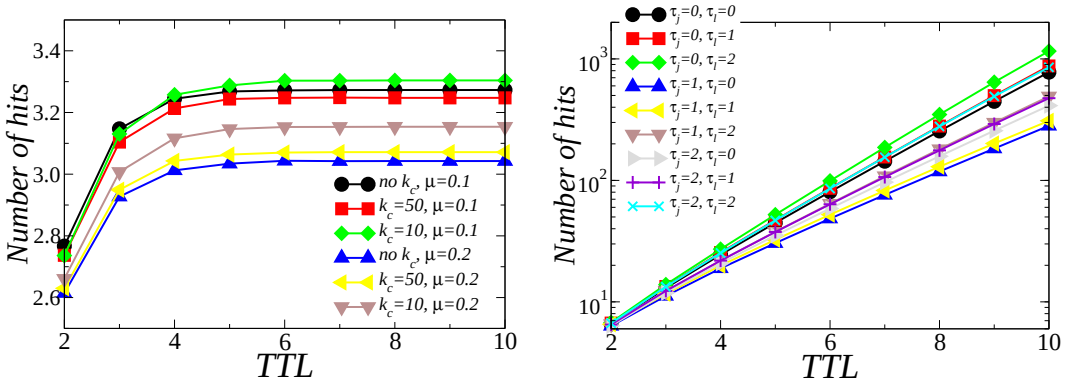
(a) μ=0.1

(b) μ=0.2



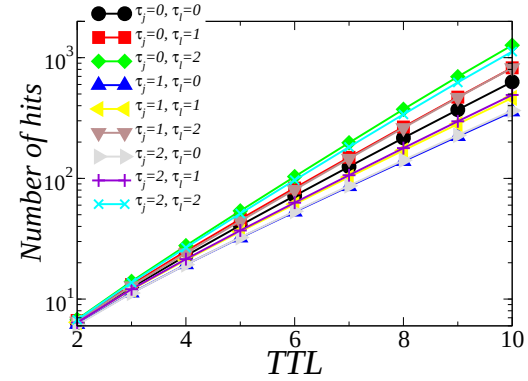
(c) μ=0.3

Figure 5.5: Flooding (FL) performance over topologies generated with k_c=10 and m=3.

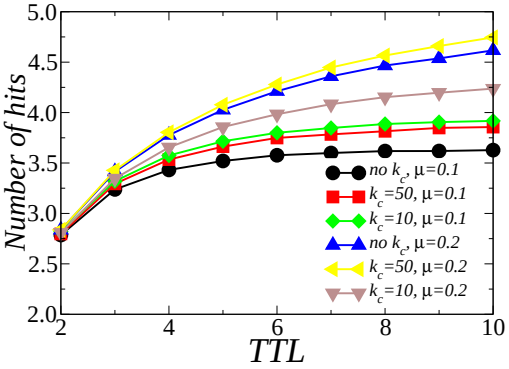


(a) m=1, τ_j=2, τ_l=0

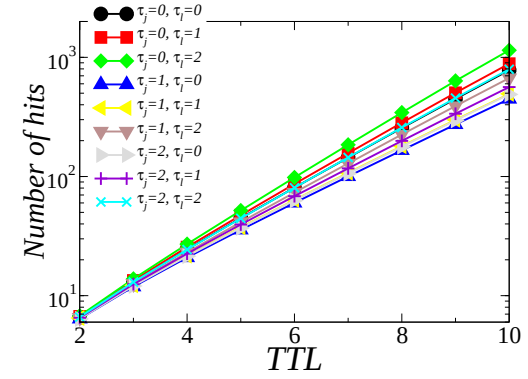
(b) m=2, μ=0.10, k_c=50



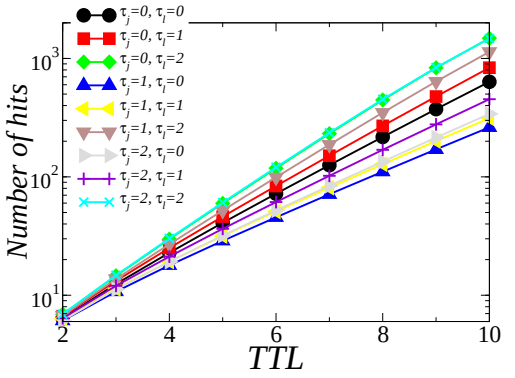
(c) m=2, μ=0.20, k_c=10



(d) m=1, τ_j=2, τ_l=2

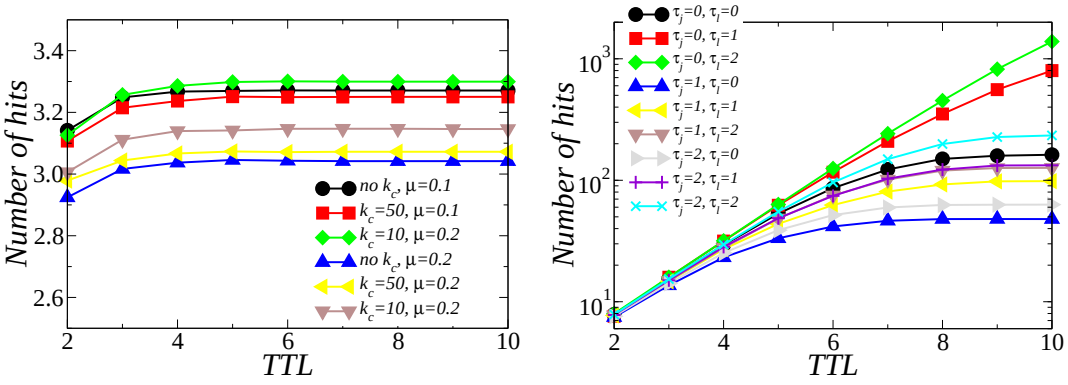


(e) m=2, μ=0.10, k_c=10

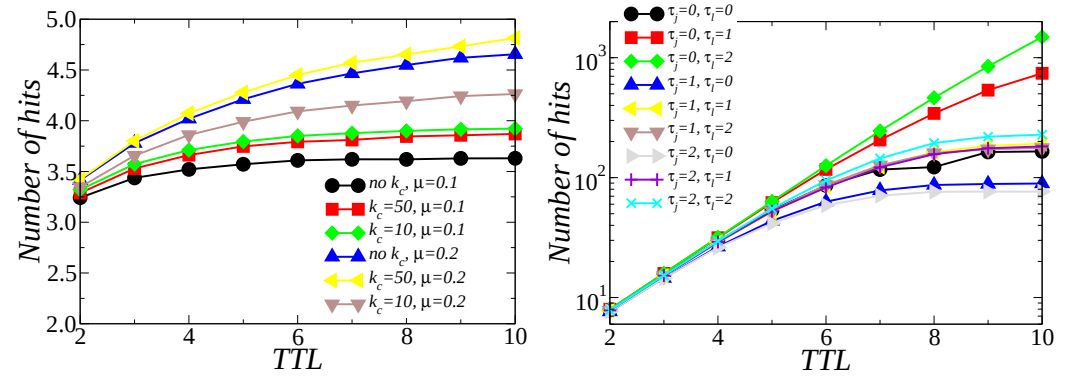


(f) m=2, μ=0.20, k_c=50

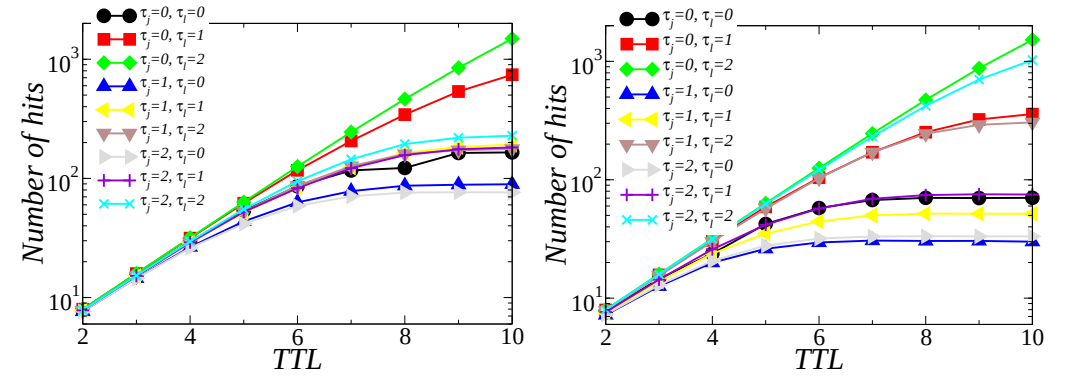
Figure 5.6: Normalized Flooding (NF) performance over topologies generated with various m, τ_j, and τ_l values.



(a) m=1, τ_j=2, τ_l=0 (b) m=2, μ=0.10, k_c=50



(c) m=2, μ=0.20, k_c=10 (d) m=1, τ_j=2, τ_l=2



(e) m=2, μ=0.10, k_c=10 (f) m=2, μ=0.20, k_c=50

Figure 5.7: Random Walk (RW) performance over topologies generated with various m, τ_j, τ_l, and k_c values.

the node i. Its value is obtained by counting the number of edges (denoted by e_i) among the neighbors of the node i and dividing this number to the maximum possible number of edges among the neighbors, i.e., $k_i(k_i-1)/2$, where k_i is the degree of the node i ($c_i = 0$ for $k_i = 0$ or 1) [17]:

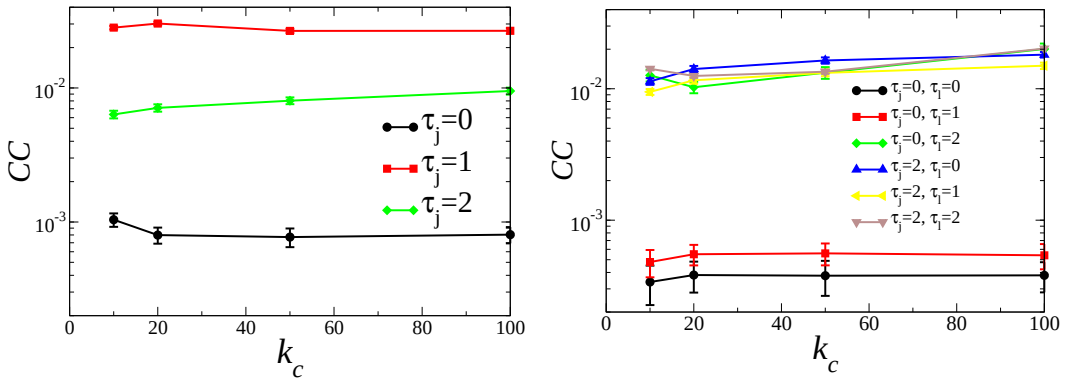
$$c_i = \frac{2e_i}{k_i(k_i-1)} \tag{5.1}$$

The clustering coefficient of the network is calculated by taking the average of the local clustering coefficients of all the nodes in the network:

$$C = \langle c \rangle = \frac{1}{N}\sum_i c_i \tag{5.2}$$

By definition, $0 \leq c_i \leq 1$ and $0 \leq C \leq 1$.

Cutoffs have a noticeable effect on clustering coefficient. Figure 5.8 illustrates this behavior very well. We have considered two cases to describe this characteristic (a) m=3, μ=0 and (b) m=3, μ=0.30. For the first case there is no effect of τ_l as μ=0 therefore we observed the behavior of clustering coefficient on increasing value of cutoffs on different values of τ_j [17]. However, for second case we observed this trend on different values of τ_j and τ_l. It's interesting to find out that increasing cutoffs, increased the values of clustering coefficients in most of the cases. After analyzing both the graphs it can be stated that the clustering coefficient of the network increases with the increase in the cutoff. This relationship between hard cutoffs and the clustering coefficient can be attributed to the fact that rewiring process gets limited in being able to connect to close by (i.e., within τ_l hops) peers [17]. We also observe that the clustering coefficient is very low, if not zero, for $\tau_j = 0$ and $\tau_l = 0, 1$, and increasing τ_l to 2 increases it considerably [17].



(a) $\mu = 0.00$ and m = 3

(b) $\mu = 0.30$ and m = 3

Figure 5.8: Clustering coefficient over cutoffs

Chapter 6

Conclusions and Future Work

To summarize, we worked on an ad-hoc limited scale-free network model for unstructured peer-to-peer networks. We did a detailed literature survey about the existing structured and unstructured peer-to-peer technologies and search algorithms. We developed an algorithm for joining and leaving procedure for the peers using local heuristics. We ran various search algorithms like flooding, normalized flooding and random walk on the generated network topologies by simulation to measure the search efficiency. Considering the ad-hocness of peers we parameterized our algorithm by imposing various criteria like hard cutoff, minimum degree (number of stubs), limiting information available to nodes at the time of joining and leaving and deleting nodes randomly to incorporate the churn in the network [17].

We analyzed the effects of these parameters on degree distributions and search performance of the network of 10,000 nodes generated by simulation. We also studied the effect on clustering coefficients of the network by varying the value of these parameters. Typically high values of these parameters will make the network a preferential attachment network with degree distribution exponent 3 [17]. By the results of our simulation we observed a very interesting fact that the negative effects of hard cutoff and high churn in the network can be compromised by providing more information at the time of joining and rewiring process (i.e. increasing the value of τ_j and τ_l). We take an opportunity to say that our findings are directly applicable to the current unstructured peer-to-peer networks, where a peer joins in and leaves the network at any point of time and each node have an upper limit

on the links that they can have with other nodes in the network.

As a future work, I propose to verify and support our theoretical model by some practical experiments. We can compare the search performance of our model with the existing ones. There are various ways we can quantify this goal. One should collect the network data of an existing peer-to-peer network, which is nothing but a set of bare nodes and links. This would provide us with the topological information and characteristics about which node is connected to which node. Once we have that information we can make an overlay topology on top of this network and then run the search algorithm and analyze the performances. The major work for this area of research would be collecting the topological information for a P2P network, which might not be available right away due to the security reasons.

The other way to serve the purpose of verifying our model would be to actually make our own client (peer) working on our algorithm and let it participate in the existing P2P network like Gnutella. As a survey on unstructured P2P network I came across with various client software for the Gnutella network which are open source like limewire [24]. We can modify these client software to work according to our algorithm. The main attention of this topic of research would be to tweak the existing code in such way that a peer can obtain the degree information about its neighbors based on the value of τ_j and τ_l. These two research studies are interesting and would easily validate our theoretical model.

Bibliography

[1] R. Albert and A.-L. Barabási. Bose-einstein condensation in complex networks. *Phys. Rev. Lett.*, 86:5632, 2001.

[2] R. Albert and A.-L. Barabási. Competition and multiscaling in evolving networks. *Europhys. Lett.*, 54:436, 2001.

[3] R. Albert and A. Barabsi. Statistical mechanics of complex networks. In *Reviews of Modern Physics*, 2002.

[4] R. Albert, H. Jeong, and A.-L. Barabasi. Diameter of the world wide web. In *Nature*, pages 401–130, 1999.

[5] P. Backx, T. Wauters, B. Dhoedt, and P. Demeester. A comparison of peer-to-peer architectures. In *IEEE*, 2002.

[6] A.-L. Barabasi. Evolution of the social network of scientific collaborations. In *Physica A*, volume 311:590, 2002.

[7] A. Barabsi and R. Albert. Emergence of scaling in random networks. In *Science*, volume 286, 1999.

[8] A. Barabsi and R. Albert. Topology of evolving networks: local events and universality. In *Physical Review Letters*, volume 85:5234, 2000.

[9] A. Barabsi, H. Jeong, and R. Albert. Error and attack tolerance of complex networks. In *Nature*, volume 409, 2000.

[10] M. Bawa, G. Manku, and P. Raghavan. Sets: Search enhanced by topic segmentation. In *Proceedings of the 26th Annual International ACM SIGIR Conference pages 306313, Toronto, Canada*, July 2003.

[11] E. Bender and E. Canfield. The asymptotic number of labeled graphs with given degree sequences. *J. Comb. Theory A*, 24:296, 1978.

[12] Clip2. The gnutella 0.4 protocol specification. In *http://dss.clip2.com/GnutellaProtocol04.pdf*, 2000.

[13] J. Doyle, D. L. Anderson, L. Li, S. Low, M. Roughan, S. Shalunov, R. Tanaka, and W. Willinger. The 'robust yet fragile' nature of internet. In *Proceedings of the National Academy of Sciences*, volume 102, 2005.

[14] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On powerlaw relationships of the internet topology. In *ACM Computer Communication Review*, pages 29–251, 1999.

[15] FreenetProject. Freenet project. In *http://freenetproject.org*, 2009.

[16] C. Gkantsidis, M. Mihail, and A. Saberi. Hybrid search schemes for unstructured peer-to-peer networks. In *IEEE Infocom*, 2005.

[17] H. Guclu, D. Kumari, and M. Yuksel. Ad-hoc limited scale-free models for unstructured peer-to-peer networks. In *Proceedings of IEEE International Conference on Peer-to-Peer Computing (P2P)*, 2008.

[18] H. Guclu and M. Yuksel. Scale-free overlay topologies with hard cutoffs for unstructured peer-to-peer networks. In *ICDCS Proceedings of the 27th International Conference on Distributed Computing Systems*, 2007.

[19] H.Ebel, M.-I. Mielsch, and S. Bornholdt. Scale-free topology of e-mail networks. In *Physical Review E*, volume 66:035103(R), 2002.

[20] D. Ilie and A. Popescu. Statistical models for gnutella signaling traffic. *Comput. Netw.*, 51(17):4816–4835, 2007.

[21] T. Klingberg and R. Manfredi. Gnutella 0.6, draft. *http://groups.yahoo.com/group/the gdf/files/, Development/GnutellaProtocol-v0.6-200206draft.txt*, 2002.

[22] P. Krapivsky and S. Redner. Organization of growing random networks. *Phys. Rev. E*, 63:66123, 2001.

[23] P. Krapivsky, S. Redner, and F. Leyvraz. Connectivity of growing random networks. *Phys. Rev. Lett.*, 85:4629, 2000.

[24] Limewire. The gnutella 0.6 protocol specification. In *http://www.limewire.org/*, 2002.

[25] Y. Liu, L. Xiao, X. Liu, L. M. Ni, and X. Zhang. Location awareness in unstructured peer-to-peer systems. In *IEEECS*, 2004.

[26] V. Lo, D. Zhou, Y. Liu, C. GauthierDickey, and J. Li. Scalable supernode selection in peer-to-peer overlay networks. In *Proceedings of the 2005 Second International Workshop on Hot Topics in Peer-to-Peer Systems (HOT-P2P'05)*, 2005.

[27] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and replication in unstructured peertopeer networks. In *ACM*, 2002.

[28] R. P.-S. M. Boguñá and A. Vespignani. Cut-offs and finite size effects in scale free networks. *The European Physical Journal B*, 38:205, 2004.

[29] M. Molloy and B. Reed. A critical-point for random graphs with a given degree sequence. *Random Structures & Algorithms*, 6:161, 1995.

[30] M. Molloy and B. Reed. The size of the giant component of a random graph with a given degree sequence. *Combinatorics, Probability and Computing*, 7:295, 1998.

[31] S. Saroiu, K. P. Gummadi, and S. D. Gribble. Measuring and analysing the characteristics of napster and gnutella hosts. In *Multimedia Systems*, 2004.

[32] M. Scholz. Network science. In *http://www.network-science.org*, 2009.

[33] K. Spripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interestbased locality in peer-to-peer system. In *Proceedings of IEEE INFOCOM, Volume 3, pages 21662176,San Francisco, CA*, March 2003.

[34] K. Sripanidkulchai. The popularity of gnutella queries and its implications on scalability. In *OReilly Peer-to-Peer and Web Services Conference*, 2001.

[35] D. Watts and S. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440, 1998.

[36] B. Yang and H. Garcia-Molina. Efficient search in peer-to-peer networks. In *The 22nd IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2002.

[37] Y. Zhu and Y. Hu. *Semantic Search in Peer-to-Peer Systems*, chapter 38. Taylor and Francis Group, LLC, 2006.