# EFFECT OF OVERLAY TOPOLOGY ON PEER-TO-PEER DATA DISSEMINATION AND BUFFER MANAGEMENT

by

Emre İskender

A Thesis Submitted to the
Graduate School of Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Electrical & Computer Engineering

Koç University

July, 2009

Koç University
Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Emre İskender

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Associate Prof. Öznur Özkasap, (Advisor)

Associate Prof. Mine Çağlar, (Advisor)

Prof. A. Murat Tekalp

Associate Prof. Selda Küçükçifçi

Associate Prof. Engin Erzin

Date: 27/07/2009

*To my parents Güler and Ömer and my brother İlker, I gratefully dedicate this thesis. Thank you for always supporting me.*

# ABSTRACT

Keeping every node updated about the newly generated data in the dynamic and rapidly changing environment of a network is achieved via various data dissemination algorithms. Epidemics is one of the widely accepted algorithms because of its reliability and robustness. Message loss recovery for maintaining the reliability of the content delivery in case of message losses is achieved via several buffer management techniques. Efficient usage of limited memory resources is the basic deal for buffer management.

In this thesis, we present our analysis of peer-to-peer (P2P) networking phenomena, namely data dissemination and buffer management, focusing on topological perspectives. For data dissemination, we examine spreading of epidemics for anti-entropy algorithms on several overlay network topologies, considering peer proximity. We derive nodes' exact probability distributions of being infected in each epidemic cycle of data dissemination. For buffer management, we examine buffering with an efficient algorithm, Stepwise Fair-share Buffering, that uses memory resources effectively and distributes the buffering load uniformly throughout the system. We analyze the effect of different topologies on buffer management, using hierarchical and power-law topologies, two basic types of topology modeling the Internet.

For data dissemination, the effect of topological properties is studied using numerical evaluations. The rate of dissemination is found to be related to the adjacency matrix in a nonlinear way. For buffering, performance evaluation of various models with hierarchical and power-law topologies are conducted. Scalability, reliability, dissemination delays and uniformity are considered as basic performance parameters. We have shown that Stepwise Fair-share Buffering method facilitate better uniformity in distribution of buffering load, in view of our simulations. We expect to have higher delays due to decision process performed for bufferer selection. However, it is also shown that dissemination delay performance drawback is eliminated when power-law topologies are considered.

# ÖZETÇE

Enerjik ve hızlı değişen ağ ortamlarında, yeni gelen her bir bilgi hakkında, ağdaki kullanıcıları bilgilendirme işlemi, çeşitli bilgi yayılımı algoritmalarıyla sağlanır. Güvenilirliği ve sağlamlığı ile, salgın yayılımı algoritmaları, bu algoritmalar arasında en yaygın olanlardan biridir. İçerik dağıtımı sırasındaki güvenilirliğin sağlanması için, herhangi bir mesaj kaybı durumunda, kaybolan mesajların yeniden temini, çesitli ara bellek yönetimi yöntemleri ile sağlanır. Sınırlı bellek kaynaklarının etkili bir şekilde kullanımı, ara bellek yönetiminin en temel amaçlarından biridir.

Bu tez çalışmasında, bilgi yayılımı ve ara bellek yönetimi olmak üzere, görevdeş ağlardaki iki temel kavram, topolojik yönlerden analiz edilmektedir. Bilgi yayılımı için, görevdeş ağlarda, verilen herhangi bir topoloji ile, komşuluk bilgisine bağlı, entropi-önler algoritmalarıyla yayılım incelenmektedir. Ağdaki bütün düğümlerin, her bir salgın döngüsü esnasındaki enfekte olma olasılıkları bulunmaktadır. Ara bellek yönetimi için, sistemdeki bellek kaynaklarını etkin bir biçimde kullanan ve bellek yükünü sistem üzerindeki kullanıcılar üzerine dengeli bir biçimde dağıtan bir algoritma olan, Adımsal Eşit Dağılımlı Ara Bellek algoritması ile bellek dağıtımı incelenmektedir. İnterneti örnekleyen sıra-düzensel ve üs kanunu temel topolojilerinin, ara bellek yönetimine olan farklı etkileri incelenmektedir.

Topolojik özelliklerin bilgi yayılımına olan etkileri, sayısal hesaplamalarla incelenmiştir. Yayılım hızının, komşuluk matrisine, doğrusal olmayan bir yolla bağlı olduğu bulunmuştur. Ara bellek modelinin, sıra-düzensel ve üs kanunu topolojilerdeki başarım hesaplamaları benzetim sonuçlarıyla bulunmuştur. Temel başarım parametreleri olarak, ölçeklenebilirlik, güvenilirlik, yayılım gecikme zamanları ve dengeli dağılım dikkate alınmıştır. Adımsal Eşit Dağılımlı Ara Bellek algoritmasının, bellek yükünü sistem üzerindeki kullanıcılara dengeli bir biçimde dağıttığı benzetim sonuçlarıyla gösterilmiştir. Ara bellek seçimindeki karar verme sürecinin gecikmeye sebep olmasını beklediğimiz halde, üs kanunu topolojileri ele alındığında, bu gecikmenin büyük ölçüde giderildiği görülmüştür.

# ACKNOWLEDGMENTS

I wish to express my sincere gratitude to my thesis supervisors Associate Prof. Öznur Özkasap and Associate Prof. Mine Çağlar for their support, valuable criticism, and endless patience throughout my study. I am grateful to members of my thesis committee for critical reading of this thesis and for their valuable comments. I would like to thank Prof. A. Murat Tekalp, Associate Prof. Selda Küçükçifçi and Associate Prof. Engin Erzin for the critical reading of this thesis and serving on my advisory committee. I gratefully acknowledge the opportunity given by the Department of Electrical & Computer Engineering at Koç University. I have studied for more than three years in this department with a truly enriching environment. The financial support of the Turkish National Science Foundation (TÜBİTAK) is also sincerely acknowledged. I would like to thank to my friends Emrah Cem, Nargiz Kalantarova, Ergun Biçici and Emrah Ahi for their encouragement and support. I would also like to thank to Mahmut Bibinoğlu, my manager in Nortel Netaş, for his great tolerance during my research. Finally I thank my family and my fiancée Hilal Alkan for providing me a morale support that helps me in hard days of my research.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

## NOMENCLATURE

| | |
|---|---|
| ACK | Acknowledgement |
| BF | Buffer Fullness Ratio of a Peer |
| LRU | Least Recently Used |
| NAK | Negative Acknowledgement |
| NH | Neighbor History Information |
| P2P | Peer-to-Peer |
| RMTP | Reliable Message Transport Protocol |
| RRMP | Randomized Reliable Multicast Protocol |
| SRM | Scalable Reliable Multicast Protocol |
| TTL | Time-to-Live |

Chapter 1

# INTRODUCTION

## 1.1 Motivation

Growth of networks, especially the Internet, results in considerable interest on distributed systems [1]. P2P networking is a popular paradigm of distributed systems in which there exists diverse connectivity among the participants, and cumulative usage of network resources, especially the bandwidth, is the core value of P2P networks. P2P model has become a very powerful paradigm for developing Internet-scale systems and sharing resources (i.e., CPU cycles, memory, storage space, network bandwidth) over large scale geographical areas [2]. There are popular P2P applications on file sharing such as Bit Torrent, Gnutella, Freenet and Morpheus.

In the last few years, a new class of distributed stream processing applications have emerged in domains such as network traffic monitoring, financial, health-care, sensor data acquisition and multimedia. In distributed stream processing applications, data produced by heterogeneous, autonomous and large numbers of globally-distributed data sources are composed dynamically to generate results of interest. These offer scalability and availability advantages by harnessing distributed processing elements in a cost-effective way. More advantages of distributed stream processing applications include their ability for customized delivery, for adaptation to different loads, and for resiliency to node failures. Distributed stream processing can also be applied to multimedia streams, to eliminate the need for a dedicated server with a high bandwidth connection and offer media services that can be composed on demand.

Real time data is also efficiently transmitted using P2P technology. For Internet based video broadcasting applications such as IPTV, the P2P streaming scheme has been found to be an effective solution [3]. P2P live streaming has become a viable solution for IPTV services with medium quality video for a large number of concurrent users. With the popularity of video on demand applications over the Internet, the traditional client-server and content server at edge solutions are not adequate in handling dynamic viewer behaviors and do not scale well with a large audience. On the other hand, the P2P based solutions utilizing application layer overlay are becoming popular, because it is easy to implement and cheaper than duplicating content servers at edges. The core benefit of P2P based solution is that it utilizes the buffering and uploading capacities of the participating peers, and provides a more scalable and robust content delivery solution. PPLive[4], PPStream[5], CoolStreaming[6] and Tribler[7] are the well known examples for the great success of P2P streaming systems.

Tribler[7] is a software for watching TV online. It is an open source P2P client with various features for watching videos online and it is based on the BitTorrent protocol. It uses an overlay network for content searching and adds keyword search ability to the BitTorrent file download protocol using a gossip protocol. The software includes the ability to recommend content. After a dozen downloads the Tribler software can roughly estimate the download taste of the user and recommends content. This feature is based on collaborative filtering. Another feature of Tribler is a limited form of social networking and donation of upload capacity. Tribler includes the ability to mark specific users as online friends. Such friends can be used to increase the download speed of files by using their upload capacity.

There are two significant issues in P2P networking. The first issue of our interest is data dissemination. P2P networks are dynamic networks and peers in the network may need to be informed about newly generated messages throughout the system in order to keep the network up to date [8]. There are two different methods for modeling data dissemination: Simple epidemics and anti-entropy algorithms [9]. In simple epidemics, epidemics disseminate from an infectious peer to a subset of its neighbors, defined by the fan-out parameter, in each epidemic round. There is no mutual exchange of state information and an infectious peer may receive a particular data message multiple times. This causes redundant message transmission in the network, but overhead is reduced. In anti-entropy algorithms, peers in the network choose one or a group of its neighbors determined by fan-out and exchange status information prior to actual data dissemination. This phase is called epidemics. There are three approaches for data exchange, namely pull, push and hybrid, as particular models of anti-entropy. In anti-entropy algorithms, data carried on each peer is compared prior to data exchange to avoid the pitfall of sending unnecessary data as in simple epidemics. The algorithm causes no overhead.

Epidemics is the most popular algorithm because of its reliability concern [10, 11]. Additionally, the effect of epidemics is that data can spread within a group just as it would in real life [12]. A critical ratio for detecting if the epidemics will spread to entire network or not is named as epidemic threshold. In earlier studies, the effect of network topology on dissemination is examined and different epidemic thresholds are identified in relation to various topological properties of the underlying network, such as average connectivity, connectivity divergence of the topology and maximum eigenvalue of the adjacency matrix.

The second issue of our interest is buffer management. In order to provide reliable dissemination throughout the system, missing messages need to be retrieved successfully in case of message losses. Retrieval of lost messages is achieved via several buffer management techniques. However, limited memory resources in the system requires clever buffer management techniques in order use memory resources efficiently. In earlier studies, both topological and non-topological methods for efficient buffer management are proposed. For non-topological methods; in Bimodal Multicast [13], receiving peer buffers the messages for a fixed amount of time; in [14], NAK based retransmission control scheme is used in order to overcome failures due to high message generation rate; in [15] the message is discarded from the buffer after the nodes with the least reliable and slowest links are monitored based on both ACK and NAK messages; in [16], safe messages are detected and discarded from the buffers; in Search Party algorithm [17], messages are kept in the buffers for a fixed amount of time and discarded at the end of this time period without any restriction; in LRU discard method [18], least recently used (LRU) message is discarded from the buffer in case of buffer overflow; in [19], history buffers are used during data dissemination, in order to overcome multiple delivery of the messages; in hash-based buffering [20], each message is buffered by a small set of peers determined by a hash function. For topological methods; in Stepwise Probabilistic Buffering [21], sending peer randomly chooses the bufferer corresponding to the message prior to message transmission; in Stepwise Fair-share Buffering [21], the bufferers are selected during data dissemination through an adaptive scheme considering locality information of the topology; in Randomized Reliable Multicast Protocol (RRMP)

[22], the peers in the network are grouped in local regions to keep buffering load in each local region balanced; in Reliable Multicast Transport Protocol (RMTP) [23], there is a supervising peer responsible for collecting ACK messages from the peers in its local region and retransmitting lost messages to corresponding receivers. Among these methods, Stepwise Fair-share Buffering is proposed as a mechanism that uses memory resources effectively [21]. Additionally, this algorithm distributes the buffering load uniformly throughout the network elements, satisfying the fairness requirement of a P2P network. However, a general framework for analyzing the effect of different network topologies on buffer management has not been considered yet.

## 1.2 Contribution

In this thesis, we examine spreading of epidemics for anti-entropy algorithms on several overlay network topologies, considering peer proximity. We derive nodes' exact probability distributions of being infected in each epidemic cycle of data dissemination. The effect of topological properties on data dissemination is studied using numerical evaluations. The rate of dissemination is found to be related to the adjacency matrix in a nonlinear way [9].

We also analyze and compare the effect of different topologies (using hierarchical and power-law topologies, two basic types of topology modeling the Internet) on buffer management, with various buffer management techniques. Simulation results show that, fair-share approach performs balanced buffering load very close to perfect balancing, due to the cleverness in buffer selection process. Fair-share approach is better on buffering load performance compared to other approaches, but selection process is time consuming and results in extra delay during data dissemination. However, we have shown that power-law topologies, due to the nature of topology, achieved faster dissemination performance compared to hierarchical topologies and disadvantage of fair-share approach is eliminated by the faster dissemination performance of power-law topology structure.

As the contribution to this thesis for data dissemination, we have derived an analytical model of pull type anti-entropy approach for SI epidemic data dissemination. The rate of dissemination is found to be related to the topological properties such as degree distribution and eigenvalues of the gradient matrix over Erdös-Rényi and power-law random graphs. Rather than the maximum eigenvalue, the mean and the standard deviation of all eigenvalues are found to be effective in predicting the rate of diffusion.

As the contribution to this thesis for buffer management, performance evaluation of various models with hierarchical and power-law topologies are conducted. Scalability, reliability, dissemination delays and uniformity are considered as basic performance parameters. We have shown that Stepwise Fair-share Buffering method facilitate better uniformity in distribution of buffering load, in view of our simulations. We expect to have higher delays due to decision process performed for bufferer selection, however, it is also shown that dissemination delay performance drawback is eliminated when power-law topologies are considered.

The rest of this thesis is organized as follows. The related work is summarized in the next chapter. Chapter 3 gives the details of the proposed model and the results for data dissemination concept. Chapter 4 describes the analysis of buffer management with various buffering approaches. Simulation results of buffer management study are presented in Chapter 5. Finally, concluding remarks are given in Chapter 6.

Chapter 2

# RELATED WORK

Data dissemination and buffer management are important issues in P2P network applications. Data dissemination is important because of the dynamic and rapidly changing environment of a network requires keeping every node updated about the new data [8]. Maintaining the reliability of the content delivery is another subject of interest and to prevent message loss throughout the network, efficient buffer management is required [21]. The first part of this chapter covers related studies about data dissemination and the second part covers related studies about buffer management.

## *2.1 Data Dissemination*

Regularly informing peers in a dynamic network is achieved by data dissemination. Epidemic algorithms are first discovered to achieve reliable data dissemination in large-scale, distributed networks. In addition to reliability; simplicity, robustness, high resilience to failures and flexibility of these algorithms make them popular [1, 10, 12]. Moreover, in addition to data dissemination; it is discovered that these algorithms are also efficient for data aggregation, overlay maintenance, and resource allocation, making them widely accepted.

Epidemics in distributed systems refers to the repeated probabilistic exchange of data among members [10]. The effect of epidemics is that data can spread within a group just as it would in real life. In a sense, this is strongly related to epidemics, by which a disease is spread by infecting members of a group, which in turn can contact in others [12].

Peers in the network choose one or a group of network members determined by fan-out and exchange status information prior to actual data dissemination. This phase is called epidemics. In these algorithms, each peer compares their present data with the selected peer prior to data exchange in order to overcome unnecessary data exchange. This way, the algorithm faces some delay constraints but epidemics is a required phase, in order not to cause any overhead [9]. There are three approaches for data exchange, namely pull, push and hybrid, as particular models of anti-entropy algorithms. [24].

We define an infectious peer as the peer that holds data to be shared and susceptible peer as the peer that lacks the specific data in a network. Epidemic spreading in the network takes place from infectious nodes to susceptible nodes, and it is modeled as a process in an undirected graph with nodes where every infectious node exchanges data with one of the previously chosen members. Modeling the spread of epidemics by taking into account the topological and nodes' neighborhood information provides benefits such as predicting the future spreading behavior, developing methods to control epidemics or achieving faster epidemic data dissemination [9].

Topologically, we can classify anti-entropy algorithms according to selected epidemic member set, as epidemic anti-entropy with full membership knowledge, epidemic anti-entropy with partial membership knowledge and hybrid approaches. Next, we will describe these methods and give information about the related studies.

### *2.1.1 Epidemic anti-entropy with full membership knowledge*

In this method, data dissemination is achieved through uniformly selected peers over the entire topology. This method requires the full view of the topology for selecting the peers. Let's denote $N$ to be the total number peers in the network: Previous analysis shows that, all nodes receive a copy of the newly generated message with a delay that is logarithmic in the full network size, in $O(log(N))$ steps of its initial appearance, which is the best possible time interval for the complete dissemination of data throughout the network. However, in many network applications, when some new data is generated at individual nodes, this newly generated data is mostly interesting for the nodes nearby. For example when an alarm is generated at an individual node, we want to alert nearby nodes earlier than nodes further away. Since this method is not effective on data dissemination through nearby nodes, it is not preferred in data dissemination methodology [8].

In a previous study [24], this method is considered assuming that each peer has global knowledge of all peers. That is, any other peer in the network can be chosen as an epidemic target. Although this assumption is not realistic, it is a crucial simplification for the exact probability calculations performed in. The probability distribution of the number of newly infected peers at each round is derived for push, pull and hybrid anti-entropy algorithms.

### 2.1.2 Epidemic anti-entropy with partial membership knowledge

In this method, data is disseminated through the neighboring peers. Peers in the network choose one or a group of its neighbors as epidemic targets and disseminate data through these selected neighbors. This method requires only a partial view of the network topology for anti-entropy peer selection. As a basic necessity, newly generated data is firstly disseminated to the peers in the close proximity and additionally this approach doesn't require the view of the full network topology. However, this approach has some drawbacks. Let's denote $N$ to be the total number peers in the network: The time it takes for all nodes to obtain a given message under this scheme is $\theta(\sqrt{n})$, which is very slow compared to epidemics through entire topology. In addition to its delay incompetence, this method is also more delicate to link failures around the source peers [8].

In an earlier work using this method [25], the effect of network topology on dissemination is examined and different epidemic thresholds are identified in relation to various topological properties of the underlying network, such as average connectivity, connectivity divergence of the topology and maximum eigenvalue of the adjacency matrix. A critical ratio for detecting 'if the epidemics will spread to entire network or not' is named as epidemic threshold. It has been shown that infection eventually dies out if $\frac{\phi}{\delta} < \textit{epidemic threshold}$ where $\phi$ is the infection rate and $\delta$ is the cure rate. The average connectivity in the network is denoted by $\langle k \rangle$, and the connectivity divergence is by $\langle k^2 \rangle$, the mean and the second moment of the degree distribution, respectively. It has been suggested that an epidemic threshold is $\tau = \frac{1}{\langle k \rangle}$ for homogenous Erdös-Rényi networks and $\tau = \frac{\langle k \rangle}{\langle k^2 \rangle}$ for power-law topologies. A general epidemic threshold of $\tau = \frac{1}{\lambda_{1,A}}$ is also suggested for an arbitrary network where $\lambda_{1,A}$ is the largest eigenvalue of the adjacency matrix.

### 2.1.3 Hybrid approaches

Dissemination through full membership knowledge and dissemination through partial topology view are the two fundamental approaches in anti-entropy methodologies. However, deficiencies of the pure applications of these approaches motivate people about finding a combination of the two. Distance-based epidemics (or in other words spatial epidemics) and hierarchical-adaptive epidemics are the famous topology related studies combining epidemic anti-entropy with full membership knowledge and epidemic anti-entropy with partial membership knowledge in order to achieve better performance.

Distance-based (spatial) epidemics [8] exhibits the best qualitative features of the two epidemic approaches, by guaranteeing that a message can be propagated to any node at distance $d$ from its originator, with high probability, in time bounded by a polynomial in $log(d)$. In distance-based epidemics, fast spreading of data is achieved as in epidemic anti-entropy with full membership knowledge $O(log(N))$ and initial information of nearby nodes about the newly generated messages is achieved as in epidemic anti-entropy with partial membership knowledge. This study is a good example showing how beneficial an approach can be by combining two different basic approaches in a single algorithm.

Epidemics has emerged as a famous technique since it achieves a reliable dissemination in large, distributed networks. However, ordinary epidemics has two major drawbacks: Large number of packets generated causes network overhead and imposing the same load on peers in case of failure without applying any adaptive scheme decreases performance of dissemination. The method named as hierarchical-adaptive epidemics [11], overcomes the network overhead phenomena by organizing members into a hierarchical structure that reflects their proximity according to some network-related metric and forcing peer groups to disseminate data through this hierarchical structure. Also, the algorithm adaptively adjusts the dissemination load imposed on peers in order to overcome performance decrease in case of failures.

## 2.2 Buffer Management

In order to achieve reliable dissemination in a network, messages should be kept in temporary storage areas, namely buffers, such that any peer could request any missing message when a failure occurs during regular dissemination [21]. This mechanism is called buffering. Limited memory resources phenomena is the basic subject to deal with in buffering. Fairness, in other words uniformly balancing the load among peers is also another subject of matter in P2P networks. But, fairness is not only specific for buffer management purposes and we will briefly explain fairness as a general issue in subsection 2.2.3. Storage and retrieval of the messages causes a great cost to the system and this extra cost should be minimized. Limited memory resources should be used effectively and any redundant data should be discarded from the system. Clever buffer management techniques are developed to minimize the extra cost of buffering while preserving a reliable dissemination.

There are many solutions of efficient memory usage for buffer management concepts in the literature. Likewise in data dissemination, approaches involving network topology also have important place in buffer management phenomena. Considering our focus on topological properties, we can classify previously recommended solutions as; buffer management without topology consideration and buffer management focusing on topology information.

### *2.2.1 Buffer Management without topology consideration*

Previous solutions for buffer management focus on decreasing memory usage simply by discarding messages from the buffers without considering effect of topology. In these solutions, messages are discarded from the buffers using the information of time passed during buffering, ACK-NAK messages generated throughout the system, random discard, LRU discard policies, discarding safe messages, etc. For instance, in Bimodal Multicast [13], a receiving peer buffers the messages for a fixed amount of time after their initial reception and then discards the message from its buffer, resulting in the reduction of memory resources.

The system may encounter failures due to limited buffer capacity, bandwidth or CPU speed in network elements when message generation rate is above a threshold. Flow control mechanisms are used in order to overcome failures due to high message generation rate. In NAK based retransmission control scheme, the sender reduces its transmission rate adaptively whenever it receives too many NAKs from the receivers. In [14], buffer capacity information of each peer is disseminated to the network and source peers adjust their particular message transmission rate according to this feedback. In these approaches, the buffer overflow at the receivers is minimized with the expense of higher transmission delay.

There is another buffer management algorithm [15] based on both ACK and NAK messages. The peers in the network are ranked according to their error rates using generated ACK and NAK messages. The nodes with the least reliable and slowest links are monitored. It is assumed that if a message is correctly received by these nodes, it has been probably received by all other nodes and the message can safely be discarded from the buffer.

In hash-based buffering [20], selection of bufferers is achieved using a hash-based method prior to the beginning of dissemination. Before the source peer starts generating messages, a bufferer is assigned for each message using a hash table with uniform hashing. The topology information used in this approach is only the number of peers in the system. Upon receiving the message, peer decides to be the bufferer for that message, using a cleverly designed hash function that distributes the buffering load uniformly among the network. However, this approach is not immune in dynamic systems, since the hash table only considers initial network elements.

In [16], there is a mechanism for discarding safe messages from the buffers. The members periodically exchange messages to inform each other about the messages they have received. When a particular message reaches all of the members, it can safely be discarded from the buffer. System wide buffer space is reduced but high traffic is caused due to frequent exchange of history messages.

In Search Party algorithm [17], messages are kept in the buffers for a fixed amount of time and discarded at the end of this time period without any restriction. It is possible for the system to face with buffer overflow due to high message generation or limited memory resources. In buffer overflow conditions, the peers are forced to discard messages from the buffers. There are different selection techniques for discarding messages in case of a buffer overflow. The oldest of the messages can be discarded, the message to be discarded from the buffer can be chosen randomly or least recently used (LRU) message can be discarded from the buffer. LRU discard method [18]is proven to have better buffer hit rate compared to other methods.

History buffers are used during data dissemination, in order to overcome multiple delivery of the messages [19]. The list of the messages that the peer has already received are kept in the history buffer. Although history buffers perform well on preventing multiple delivery of messages, it suffers from high memory consumption. Keeping a history buffer for every peer in the network, prevents the efficient usage of memory resources. Moreover, the history information of formerly disseminated messages lose the necessity, since it becomes less probable for the multiple delivery of formerly generated messages. FIFO scheme is applicable for history buffer scheme. Therefore, size of history buffer can be chosen large enough to guarantee safe delivery and not to give rise to multiple deliveries of the same message to the application. Choosing an appropriate history buffer size saves memory resources while reducing multiple deliveries.

### *2.2.2 Buffer Management focusing on topology information*

It is shown that topology has an important effect on data dissemination and buffering in P2P networks [9, 21]. Better buffer management performance is achieved when topology knowledge is considered together with messages generated throughout the topology.

Stepwise Probabilistic Buffering [21] is proposed to distribute the load of buffering evenly to the entire system where all peers have only partial knowledge of the participants. For determining the bufferers of a data message, the source sends buffering request messages to randomly selected $b$ peers in its partial view. Parameter $b$ is the number of bufferers per message. For a data message, if $b > 1$ then its bufferers are determined in parallel. Buffer fullness ratio of a peer (BF) is the ratio of the number of messages that are stored in the peers buffer to its long-term buffer capacity. Time-to-Live (TTL) value attached to a buffering request message indicates the maximum number of times that request message can be forwarded among peers. When a peer receives a buffering request message for a particular data, it accepts the request with probability $1 - BF$. Otherwise, it forwards the message to a randomly selected peer from its partial view with a probability equal to BF. For example, if 90% of the long-term buffer is full, then the peer becomes the bufferer of the message with probability of 0.1 and sends the buffering request to one of its neighbors with probability of 0.9. Initially, assuming that all buffers are empty, peers that are in the partial view of the source will accept the buffering requests with higher probabilities. Then, as the buffer level of these neighboring peers will approach their capacity, they will begin to forward the buffering requests with higher probabilities to their neighboring nodes. Likewise, as the data dissemination continues, the peers with one or more hops away from the source will begin to reach their buffer capacities and forward the buffering requests to their neighbors. Thus, a stepwise probabilistic buffering takes place.

In Stepwise Fair-share Buffering [21], the bufferers are selected during data dissemination through an adaptive scheme considering locality information of the topology that distributes the buffering load uniformly. In the method, every peer stores the number of messages that its neighbors have ever buffered. This is called the neighbor history information (NH). This information is used for determination of the bufferers. At specific time intervals, the peers update their neighbor history information. The bufferer determination phase is initiated by the source to one of its neighbors through a selection mechanism. Time-to-live (TTL) value attached to a buffering request message indicates the maximum number of times that request message can be forwarded among peers. When a peer receives a buffering request it decreases the TTL value attached to a buffering request message. If the TTL value becomes zero, then the peer accepts the buffering request. If TTL value is greater than zero, the peer multicasts neighbor history request messages to its neighbors. As soon as the peer receives all the responses from the neighbors, it updates its neighbor history information. Then, it detects the peers with the minimum number of messages buffered. If the corresponding peer is the peer itself it accepts the buffering request, otherwise if it is one of the neighboring peers it sends the buffering request to that neighbor. If there is more than one peer with the minimum number of buffered messages, the peer chooses randomly one of them. Similarly, if the peer is one of these candidate peers and it chooses itself then it accepts the request. Another advantage of this approach is its immunity in dynamic systems since newly joining peers are considered in this adaptive scheme.

In Randomized Reliable Multicast Protocol (RRMP) [22], the peers in the network are grouped in local regions. The peers have two separate buffer spaces: short-term and long-term. Upon initially receiving the message, receiving peer keeps the message in its short-term buffer and waits for a while until the message completely disseminates to the local region. Then the peer makes a random choice and decides whether to be the long-term bufferer for the message or discard the message. Buffering load in each local region is kept balanced after this decision process. The message is buffered in the long-term buffer for a fixed amount of time and during this period newly incoming messages are discarded. Discovery of the repair node takes a long time in this approach.

A tree based reliable multicast protocol in this category is the reliable multicast transport protocol (RMTP) [23]. The protocol is designed for reliable delivery of data from one sender to a group of receivers. In RMTP a hierarchical tree-based approach is used. Receivers are grouped into local regions or domains and in each region there is a special receiver called designated receiver. Each designated receiver has the knowledge of the members in its local region and the sender. A designated receiver in each local region is responsible for sending acknowledgments periodically to the sender, for processing acknowledgment from receivers in its domain, and for retransmitting lost packets to the corresponding receivers. The sender multicasts data to all receivers but only designated receivers inform the sender about their status. Each receiver periodically sends an ACK to its designated receiver instead of sending an ACK for every received packet. This ACK contains the maximum packet number that each receiver has successfully received. However, error recovery is delayed by this periodic feedback policy. Hence, RMTP is not suitable for applications that transmit time sensitive data. In addition, in RMTP the whole multicast session data is in the secondary storage of the repair node for retransmission. Therefore, it is not applicable to large groups or long-lived sessions.

### *2.2.3 Importance of Fairness in P2P networks*

Fair distribution of content and load balancing is one of the primitive requirements of P2P networks. Every peer in a P2P network participates as both server and client and every peer is the basic building block for the system. Highly loaded peers become more vulnerable and therefore equal distribution of the system load over the peers is necessary in order to keep the stability of the system.

Distribution of the content in a fair and fully decentralized manner among the peers is important because it can improve resource usage, minimize network latencies and reduce the volume of unnecessary traffic incurred in large-scale P2P systems. Load balancing can be achieved by replication [26]. Firstly, popular documents are determined and replicated and then the replications are sent to the less loaded peers in the system. By replication, the system become more fair and stable with a little the expense of increase in memory usage. As another way of fair content distribution, we can specify the less loaded peers in the system and we can route the newly upcoming messages to these less loaded peers. This method fairly distributes the load over the system with the expense of specifying less loaded peers in the network.

Similar to fair distribution of content, buffering load should also be fairly distributed over the peers in the network. Any algorithm designed for performance increase in a P2P network should consider fair distribution of content as well as fair distribution buffering load. Since all peers participate in a P2P system with equal functionality, highly loaded peers become more vulnerable and fairness of the buffering load becomes a critical issue for the stability of the system. Stepwise Probabilistic Buffering and Stepwise Fair-share Buffering [21] are good examples for buffering approaches having a fair distribution of bufferers in the network.

In order to measure the degree of fairness, a powerful fairness metric is proposed: Fairness index proposed by Ray Jain [27], given below, is a very useful fairness indicator. If a system allocates resources to $n$ contending users, such that the $i^{\text{th}}$ user receives an allocation $x_i$, the following index called fairness index for the system is proposed:

$$F(x) = \frac{(\sum x_i)^2}{n \sum x_i^2}$$

This index is independent of scale, continuous, applies to any number of users. The result ranges from $1/n$ (worst case) to 1 (best case) and it has a intuitive relationship with user perception. This fairness index is recommended to use in any kind of fairness measure.

Chapter 3

# ANALYTICAL MODEL FOR TOPOLOGY DEPENDENCE IN PEER-TO-PEER ANTI-ENTROPY SPREADING

We examine spreading of epidemics for an anti-entropy algorithm in networks with various P2P overlay topologies. Neighborhood knowledge among peers and data exchange based on proximity are considered. Our analytical model for SI (Susceptible-Infected) epidemics involves equations for calculating the infection probability of each peer in consecutive epidemic rounds as a function of the topology. Using numerical evaluations, we study the effect of graph properties on dissemination as an aspect of real world P2P overlays.

## *3.1 Problem Overview*

In this chapter, we investigate the impact of topology using SI epidemic model, which is a suitable model for data dissemination applications. Dissemination of a single message is our subject of interest. Epidemic spreading in a network takes place from infectious nodes to susceptible nodes. It is modeled as a process in an undirected graph where every infectious node exchanges data with one of its neighbors. Modeling the spread of epidemics by taking into account the topology and neighborhood information provides benefits such as predicting the future spreading behavior, developing methods to control epidemics or achieving faster data dissemination. Topological properties considered for SIS model previously and graph invariants such as degree distribution and eigenvalues are studied as an aspect of real world P2P networks. In P2P content dissemination systems such as BitTorrent [28] and SeCond [29], each peer exchanges data with a group of its neighbors on the overlay. We introduce a model for calculating the infection probabilities of the nodes as a function of the topology through a general adjacency matrix and show numerical results on various power-law and Erdös-Rényi random topologies.

## 3.2 Principles of Epidemic Spreading

In this section, we give information about the types of epidemic models and define epidemic dissemination approaches.

### 3.2.1 Epidemic Models

In SI (Susceptible-Infected) model, infectious peers are never cured and continue to infect the remaining susceptible peers until the infection is spread among the network. SI model is mostly applicable for data dissemination purposes over a network.

In SIS (Susceptible-Infected-Susceptible) model an infectious peer turns to be a susceptible peer after the cure. But the nodes may become infected again without any restriction. SIS model is applicable in security services in particular to spread of Internet worms and e-mail viruses.

SIR (Susceptible-Infected-Removed) model is used to represent virus/worm propagation in distributed systems [30]. There are two different proposed models for SIR model: In the first model, each infectious peer is detected and removed from the system. In this model, there exist only infectious and susceptible peers and the population size decreases dynamically due to removals. In the second model, each infectious peer is cured and gains immunity such that it does not receive infection again. In this model, there exist only infectious, susceptible and immune peers.

### 3.2.2 Dissemination Algorithms

In Simple epidemics algorithm, epidemics disseminate from an infectious peer to a subset of its neighbors, defined by the fan-out parameter, in each epidemic round. Since there is no mutual exchange of state information, an infectious peer may receive a particular data message multiple times. Hence, this causes redundant message transmission in the network. However, simple epidemics has reduced overhead in comparison to broadcasting/flooding.

In Anti-entropy algorithms, peers in the network choose one or a group of its neighbors determined by fan-out and exchange status information prior to actual data dissemination. This phase is called epidemics. There exist three approaches for data exchange, namely pull, push and hybrid, as particular models of anti-entropy [24]. In anti-entropy algorithms, data carried on each peer is compared prior to data exchange to avoid the pitfall of sending unnecessary data as in simple epidemics. The algorithm causes no overhead but epidemics is a required phase.

Epidemic spreading is examined by calculating the infection probabilities of all nodes in the network for every epidemic round with the pull based anti-entropy algorithm [24]. In the pull approach, when an infectious peer (holding data to be shared) picks a susceptible peer (lacking the specific data) randomly, this triggers data dissemination from infectious peer to the susceptible. Spreading updates are triggered by susceptible peers when they are picked as targets by infectious peers.

## *3.3 Proposed Model*

Our model examines epidemic dissemination with pull based anti-entropy algorithm and SI epidemic spreading. The pull algorithm is given in Algorithm 1 below. In SI model, the infectious peers are never cured and continue to infect the remaining susceptible peers until the infection is spread over the network as in data diffusion. The analytical model we develop in this section is an extension of earlier work developed for SIS simple epidemic which is used for spreading of viruses in particular and a peer becomes susceptible after a cure [25, 31]. In prior work for SIS (Susceptible - Infected - Susceptible) model, various epidemic thresholds are identified in relation to various topological properties of the underlying network [25, 32]. Such properties include average connectivity, connectivity divergence of the topology and maximum eigenvalue of the adjacency matrix. The epidemic threshold is important for detecting whether the epidemics will spread to the entire network or not.

**Algorithm 1** Pull Algorithm: Epidemic anti-entropy data dissemination

```
Node $I$ is infectious and node $S$ is susceptible. When $I$ picks a neighbor $S$ as the epidemic target, infection is triggered:
1. After state exchange via epidemics, $S$ requests missing data from $I$ to initiate the pull action.
2. $S$ receives (pulls) the data from $I$.
3. Upon receiving the data, $S$ becomes infectious.
```

We derive equations to calculate the infection probability of each peer (node) in consecutive epidemic rounds. We assume that an infected node equally likely chooses one of its neighbors and infects the neighbor if it is healthy. The following notation is used:

$p_{i,t}$ :probability that node $i$ is infected at time $t$

$\zeta_{i,t}$ :the probability that a node $i$ will not receive infections from its neighbors at time $t$

$n_j$ :total number of neighbors of a node $j$, that is,

$$n_j = \sum_{k=1}^{N} A(j,k) \tag{3.1}$$

where $A$ is the adjacency matrix and $N$ is the total number of nodes.

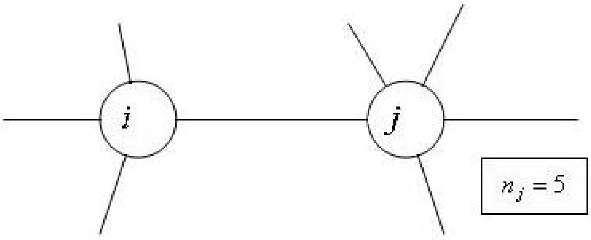

Figure 3.1: Node Selection

The selection process for a node $i$ by node $j$ in the pull approach is illustrated in Fig. 3.1 where node $j$ has 5 neighbors and hence $i$ becomes infectious with probability 1/5. Clearly, if there are multiple neighbors of $i$ which are infectious, then the probability of $i$ being selected increases in a given round.

A node $i$ remains susceptible at time $t$ when either one of the following occurs

- neighbor node $j$ is susceptible at time $t-1$, which has probability $1 - p_{j,t-1}$
- neighbor node $j$ is infected at time $t-1$ but chooses a neighbor other than $i$, which happens with probability $(n_j - 1)/n_j$

Since the neighbors act independently in anti-entropy model, we can write the probability that a node $i$ remains susceptible at time $t$ as

$$\begin{aligned} \zeta_{i,t} &= \prod_{j\text{: neighbor of } i} \left[ (1 - p_{j,t-1}) + \left( p_{j,t-1} \left( \frac{n_j - 1}{n_j} \right) \right) \right] \\ &= \prod_{j\text{: neighbor of } i} \left( 1 - \frac{p_{j,t-1}}{n_j} \right) \end{aligned}$$

Then, the probability that a node $i$ is susceptible at time $t$ is the product of the probability that it is susceptible at time $t-1$ and the probability that it does not receive infection from its neighbors. That is,

$$1 - p_{i,t} = (1 - p_{i,t-1}) \prod_{j\text{: neighbor of } i} \left[1 - \left(\frac{p_{j,t-1}}{n_j}\right)\right] \tag{3.2}$$

We can illustrate the idea of our epidemic spreading model on a sample network. Our sample network consists of 7 nodes as shown in Fig. 3.2. The adjacency matrix corresponding to this network is represented with matrix $A$ which is given below.

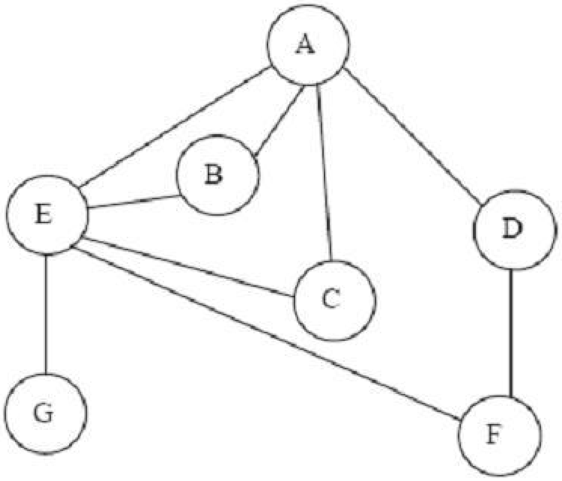

Figure 3.2: Sample network

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

In our sample network with the given adjacency matrix A, dissemination is almost complete at the end of the 6th round and we calculate the infection probabilities of the nodes using Equation (3.2). Infection probabilities, namely $p_{i,t}$, from 0th to 6th epidemic rounds of dissemination are given in Table 3.1. Here, $i$=A, B, ..., G and $t$= 0, 1, ..., 6.

| | **0** | **1** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|---|---|
| **A** | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **B** | 0 | 0.25 | 0.47 | 0.64 | 0.78 | 0.87 | 0.92 |
| **C** | 0 | 0.25 | 0.47 | 0.64 | 0.78 | 0.87 | 0.92 |
| **D** | 0 | 0.25 | 0.44 | 0.61 | 0.77 | 0.89 | 0.95 |
| **E** | 0 | 0.25 | 0.57 | 0.83 | 0.96 | 1 | 1 |
| **F** | 0 | 0 | 0.17 | 0.42 | 0.67 | 0.84 | 0.93 |
| **G** | 0 | 0 | 0.05 | 0.16 | 0.30 | 0.43 | 0.55 |

Table 3.1: Probabilities of being infected

We show that epidemic will spread to entire network, in other words the system is stable at $\vec{P} = \vec{1}$, irrespective of the size of the initial number of infected nodes, where $\vec{P}$ is the vector of entries $p_i$, $i = 1, ..., n$. It is convenient to work with the probability of being susceptible rather than being infected. Let $q_{i,t} = 1 - p_{i,t}$. From (3.2), it is given by

$$q_{i,t} = q_{i,t-1} \prod_{j:\text{ neighbor of } i} \left[ \left(1 - \frac{1}{n_j}\right) + \left(\frac{q_{j,t-1}}{n_j}\right) \right] .$$

The probability that node $i$ is still susceptible at time $t$ can be represented with the following discrete non-linear dynamical system: $\vec{Q}_t = \vec{f}(\vec{Q}_{t-1})$ with $f = (f_1, \ldots\ f_n)$ where

$$f_i(\vec{Q}) = q_i \prod_{j:\text{ neighbor of } i} \left[ \left(1 - \frac{1}{n_j}\right) + \left(\frac{q_j}{n_j}\right) \right]$$

and $\vec{Q}$ is the vector of entries $q_i$, $i = 1, ..., n$ after suppressing the time for simplicity. The system's being stable at $\vec{Q} = \vec{0}$ means that the data will certainly diffuse, that is, $P_t$ will converge to $\vec{1}$, starting with any initial number of infectious nodes. Due to [33], pg. 280, the system is stable at $\vec{Q} = \vec{0}$ if the eigenvalues of $\nabla f(\vec{0})$ are less than 1 in absolute value. The gradient matrix is given by the entries $[\nabla f(\vec{Q})]_{ik} = \partial f_i(\vec{Q})/\partial q_k$, $i, k = 1, \ldots, N$. Taking the partial derivatives, we get

$$\frac{\partial f_i(\vec{Q})}{\partial q_i} = \prod_{j\text{: neighbor of } i} \left[\left(1 - \frac{1}{n_j}\right) + \left(\frac{q_j}{n_j}\right)\right]$$

since $j \neq i$ when $j$ neighbor of $i$. On the other hand, $\partial f_i(\vec{Q})/\partial q_k = 0$ if $k \neq i$ and $k$ is not a neighbor of $i$ since $f_i(\vec{Q})$ does not depend on $q_k$. Finally,

$$\begin{aligned} \frac{\partial f_i(\vec{Q})}{\partial q_k} &= q_i \frac{\partial}{\partial q_k} \left[\left(1 - \frac{1}{n_k}\right) + \left(\frac{q_k}{n_k}\right)\right] \\ &\quad \cdot \prod_{j\text{: neighbor of } i, j \neq k} \left[\left(1 - \frac{1}{n_j}\right) + \left(\frac{q_j}{n_j}\right)\right] \\ &= \frac{q_i}{n_k} \prod_{j\text{: neighbor of } i, j \neq k} \left[\left(1 - \frac{1}{n_j}\right) + \left(\frac{q_j}{n_j}\right)\right] \end{aligned}$$

as $k \neq i$ when $k$ is a neighbor of $i$. Therefore,

$$\frac{\partial f_i(\vec{0})}{\partial q_k} = \begin{cases} \displaystyle\prod_{j\text{: neighbor of } i} \left(1 - \frac{1}{n_j}\right) & \text{if} \quad k = i \\ 0 & \text{if} \quad k \neq i \end{cases}$$

In matrix notation, we find

$$\nabla f(\vec{0}) = \operatorname{diag}(\lambda_1, \ldots, \lambda_N)$$

with

$$\lambda_i = \prod_{j\text{: neighbor of } i} (1 - 1/n_j) \qquad i = 1, \ldots, N.$$

Clearly, $\lambda_i$ are simply eigenvalues of $\nabla f(\vec{0})$ and $0 \leq \lambda_i < 1$. Therefore, the data will certainly diffuse as expected.

The analysis above does not only confirm the applicability of the discrete model (3.2) for epidemic diffusion, but also provides the tools for evaluating the rate of dissemination in connection with the adjacency matrix. Scrutinizing the stability proof of [33] which states that there exists a constant $\mu < 1$ such that

$$\|\vec{Q}_t\| \leq \mu^t \, \|\vec{Q}_0\| \tag{3.3}$$

we see that $\mu$ can be chosen as a perturbation $|\lambda| + \epsilon$ of the maximum eigenvalue $\lambda$ (in absolute value) of $\nabla f(\vec{0})$ where $\epsilon > 0$ can be chosen arbitrarily small. The largest eigenvalue would be binding in the worst case, especially for large $t$. Therefore, Equation (3.3) reflects that the dissemination occurs exponentially with a rate depending in general on all the eigenvalues $\lambda_1, \ldots, \lambda_N$ which are found above in terms of the row sums (3.1) of the adjacency matrix. Since (3.1) corresponds to the number of degrees of each node $j$, we explore the effect of the degree distribution as well as the eigenvalues on the diffusion rate for different random topologies next.

## *3.4 Numerical Results*

We consider power-law and Erdös-Rényi graphs as overlay topologies. Power law graphs have attracted great interest since the Internet topology exhibits a power law degree distribution [32]. A power law graph is one where the number of nodes with degree $k$ is proportional to $k^{-\beta}$ for some $\beta > 1$. For the mean degree to be finite, we need $\beta > 2$. On the other hand, Erdös-Rényi graph is of interest as a bench-mark random graph. Erdös-Rényi is characterized by parameters $n$ and $p$ where $n$ is the number of nodes, and there exists an edge between each pair of nodes with probability $p$ independently from the other edges. It follows that the average degree is $(n-1)p$, as stated in [32].

We evaluate epidemic spreading in various power-law graphs using Barabási power-law graph generator [34]. The nodes have an average degree which is twice of a free parameter in the generator. The algorithm creates networks with a distribution following $k^{-2.9\pm0.1}$. For Erdös-Rényi graphs, we vary the parameter $p$ to obtain different mean degrees. The network size is 1024 and we evaluate 10 graphs of each topology by varying the mean degrees. The expected number of infected nodes is found by adding the entries of the vector $P_t$ and we report the percentage of infected nodes in our numerical evaluations. We know that mean degree is one of the basic parameters providing information about the network topology. Also, in the previous sub-section, in our probability calculations of being infected, we have come up with the concept of eigenvalues of gradient matrix, as a candidate parameter providing information about the network topology. In order to come up with a topological relation on dissemination rate, both mean degree and eigenvalues of the gradient matrix are investigated with respect to the rate of diffusion. We examine the percentage of infected nodes at 15$^{\text{th}}$ and 20$^{\text{th}}$ rounds of dissemination. At the 15$^{\text{th}}$ round, infection disseminates significantly on the graph and at the 20$^{\text{th}}$ round the dissemination is almost complete.

As observed in Fig. 3.3, the diffusion rate increases quickly with the mean degree up to a certain threshold, in this case 10, then only slightly for larger degrees. Erdös-Rényi graphs show faster dissemination when compared with power-law graphs with the same mean values. In order to say that mean degree is a discriminating graph invariant, we would expect to see the correlation between rate of dissemination and mean degree of topology. Since we observe different rate of dissemination for the same mean values, we conclude that mean degree is not a discriminating graph invariant across different topologies.

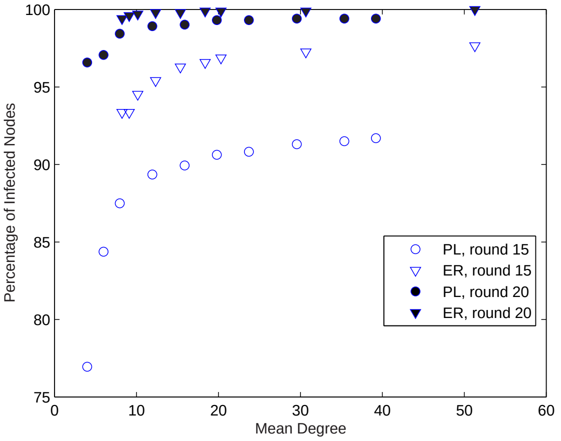

Figure 3.3: Impact of mean degree on diffusion

We observe that the mean of eigenvalues of the gradient matrix classifies the groups of different topologies at both 15$^{\text{th}}$ and 20$^{\text{th}}$ rounds of dissemination. Erdös-Rényi graphs all have a mean about 0.37 while power-law graphs have mean eigenvalue of 0.43 and larger as shown in Fig. 3.4.

We report the standard deviation of the eigenvalues in Fig. 3.5 which distinguishes clearly both between groups and within a specific group. Erdös-Rényi graphs all have smaller deviation of eigenvalues compared to power-law. In general, dissemination rate is inversely proportional to mean and standard deviation of the eigenvalues. We have also investigated the effect of standard deviation of the degree distribution. Similar to mean degree, standard deviation of the degree distribution alone is not a discriminating graph invariant across different topologies. However, Fig. 3.6 shows that dissemination rate is

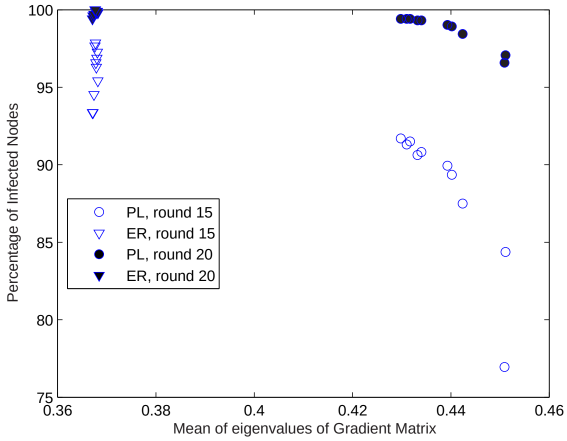

Figure 3.4: Impact of mean of eigenvalues of gradient matrix

inversely proportional to standard deviation of the degree distribution. We conclude that Erdös-Rényi graphs show faster dissemination when compared with power-law graphs since they have smaller standard deviation for both eigenvalue and degree distributions.

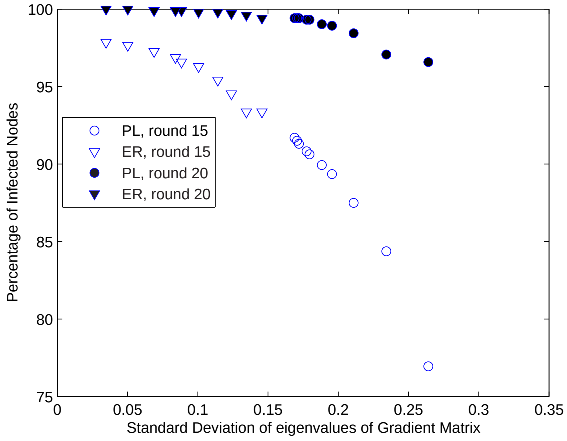

Figure 3.5: Impact of standard deviation of eigenvalues of gradient matrix

The maximum eigenvalue of the gradient matrix depicted in Fig. 3.7 shows a similar behavior to mean degree given in Fig. 3.3. Therefore, maximum eigenvalue alone is not a discriminating factor for different random graphs. Indeed, Erdös-Rényi graphs show faster

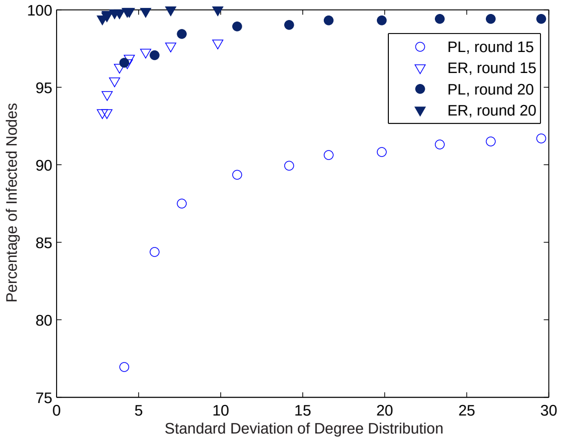

Figure 3.6: Impact of standard deviation of degree distribution

dissemination when compared with power-law graphs with the same maximum eigenvalues.

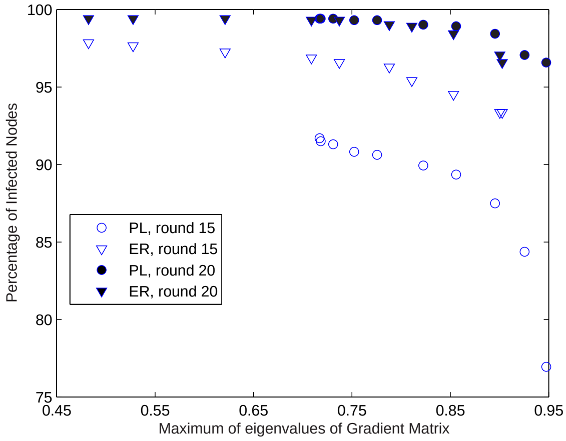

Figure 3.7: Impact of maximum of eigenvalues of gradient matrix

Chapter 4

# BUFFER SELECTION ALGORITHMS FOR PEER-TO-PEER EPIDEMIC DATA DISSEMINATION

Distributed systems have the advantage of using system resources cooperatively and several applications on the Internet are based on distributed principles, especially on P2P networking. Dynamic nature of P2P systems requires the participating elements to be up-to-date and peers should regularly be informed about the current situation of the system. Newly generated data should be disseminated throughout the network in order to maintain stability of the system. However, reliable dissemination of data is not easy and in many cases buffering is required in order to maintain reliability in case of network failures. Fairness and keeping the delay in acceptable levels are also important for the performance of the system [21].

Our contribution is to combine different approaches for data dissemination and buffer management and comparatively analyze these approaches focusing on topological properties. Dissemination of multiple data messages is our subject of interest. We consider two basic Internet modeling topologies, hierarchical [21] and most recently compromised power-law [32] Reliable data dissemination, buffer space reduction and fair distribution of bufferers are considered.

In our comparisons, we use epidemic anti-entropy with full membership knowledge and with partial membership knowledge for data dissemination; and stepwise, hash based and random schemes for buffer selection. The ideas for buffer selection and epidemics are similar but they mainly differ on topology view perspective. On data dissemination, epidemic anti-entropy with partial membership knowledge requires partial topology view while epidemic anti-entropy with full membership knowledge requires view of entire topology. On buffer selection, we have stepwise fair-share approach requiring partial view of the topology, hash based buffer selection requiring full topology view and random buffer selection algorithm requiring both partial and full topology views.

Throughout this chapter, details of each method are given with the algorithms and in order to get a general understanding, the models are described in detail, using some numerical examples. The following sections cover detailed algorithms and descriptive examples for each model and the final part covers basic events, variables, data structures and message formats being used in the models.

## *4.1 Stepwise Fair-share Buffering with Partial Membership Knowledge*

In Stepwise Fair-share Buffering [21], the bufferers are selected during data dissemination through an adaptive scheme considering locality information of the topology that distributes the buffering load uniformly. The advantage of the method is that dissemination takes place with only partial neighborhood knowledge which is the motivation of this algorithm. This approach provides a scalable and reliable data dissemination with fair buffering load imposed to the system.

In the method, every peer stores the number of messages that its neighbors have ever buffered. This is called the neighbor history information (NH). This information is used for determination of the bufferers. At specific time intervals, the peers update their neighbor history information. The bufferer determination phase is initiated by the source to one of its neighbors through a selection mechanism. Time-to-live (TTL) value attached to a buffering request message indicates the maximum number of times that request message can be forwarded among peers. When a peer receives a buffering request it decreases the TTL value attached to a buffering request message. If the TTL value becomes zero, then the peer accepts the buffering request. If TTL value is greater than zero, the peer multicasts neighbor history request messages to its neighbors. As soon as the peer receives all the responses from the neighbors, it updates its neighbor history information. Then, it detects the peers with the minimum number of messages buffered. If the corresponding peer is the peer itself it accepts the buffering request, otherwise if it is one of the neighboring peers it sends the buffering request to that neighbor. If there is more than one peer with the minimum number of buffered messages, the peer chooses randomly one of them. Similarly, if the peer is one of these candidate peers and it chooses itself then it accepts the request.

After sending the generated messages to the bufferers, data dissemination by epidemic anti-entropy starts. Messages are disseminated to the network using anti-entropy with partial membership knowledge.

Illustration of bufferer selection for Stepwise Fair-share Buffering on a simple network is given in Fig. 4.1. The figure describes the bufferer selection process. In this example, the partial view of the source node is composed of nodes 1, 2 and 3. Node 1 has neighborhood with node 6; and node 3 has neighborhood with nodes 4 and 5. Neighbor history ($NH$) information of the peers are also indicated on the figure. Following message generation, the source node searches for the neighbor with the minimum $NH$ value to forward the buffering request. Source peer discovers that, node 2 has previously buffered 3 messages and node 2 has the minimum $NH$ value compared to other neighbors (nodes 1 and 3). Source peer sends the buffering request to node 2 and node 2 starts searching for the best possible alternative among its neighbors. Node 2 discovers that node 4 has the minimum $NH$ value and node 4 is simply chosen to be the bufferer for the generated message.

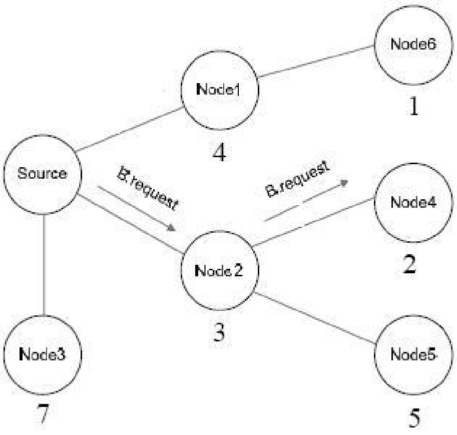

Figure 4.1: Selection of Bufferers

### 4.2 Hash-based Buffer Selection with Full Membership Knowledge

In hash-based approach [20], there is a hash function used for the bufferer selection process, following the generation of each message. If a member does not have the message buffered locally, it calculates the set of bufferers for the message using the hash function and picks one at random. The member then sends a retransmission request directly to the bufferer, specifying the message identifier and the destination address. A bufferer, on receipt of such a request, determines if it has the message buffered. If so, it satisfies the request. If not, it ignores the request. In our simulation environment, calculation of bufferers after the generation of each message is a time and memory consuming method, so we modified the algorithm such that bufferer selection process is completed prior to the beginning of dissemination. In our approach, a bufferer is assigned for each message using the hash function, before the source peer starts generating messages.

The hash function $H$ given in Alg.2 uses a table of 256 randomly chosen integers, called the shuffle table. The input to $H$ is a string of bytes, $b$ and the output is a number between 0 and 1. Byte $b$ is *messageID + peerID*. $C$ is the expected number of bufferers for a message. The output 0 means that peer with *peerID* isn't selected as the bufferer for the message with *messageID* and the output 1 means that peer with *peerID* is selected as the bufferer for the message with *messageID*.

**Algorithm 2** Hash Function

```
unsigned integer $hash = 0$;
for each byte b do
    $hash = hash$ XOR $shuffle[b$ XOR LSB$(hash)]$);
end for
double $a =$ (double) $hash$/MAX INTEGER;
return $a * numberOfPeers < C$;
```

However, for single buffer assignment case, when buffers are calculated using the hash function, some peers are assigned to buffer multiple messages, while some peers buffer none, because of randomness in buffer selection. This is an undesired situation, since we want the buffering load to be distributed uniformly among the peers. In order to overcome this deficiency, we calculate bufferers, assuming multiple bufferers for each message, i.e. 3

bufferers for each message. This way, we have observed that each peer has at least one message to buffer, but most of the messages have many bufferers, actually more than 3. Since we want single bufferer for each message, we randomly select a single peer from the multiple bufferer list.

Table 4.1 shows sample entries for messages and corresponding bufferers calculated using the hash function prior to data dissemination. As observed in the table, each message at least has a single bufferer, while some messages have multiple bufferers. As explained above, a single bufferer is randomly selected from multiple bufferer list, since we want the buffering load to be distributed fairly among the peers.

| Message | | | | |
|---|---|---|---|---|
| 1 | 5 | | | |
| 2 | 100 | 1503 | | |
| 3 | 187 | 1798 | 1820 | 1900 |
| ... | .. | | | |
| ... | .. | .. | .. | |
| 20,000 | 1 | 8 | 398 | 1500 |
| 20,001 | 89 | 120 | | |
| ... | .. | .. | .. | |
| ... | .. | .. | | |
| 100,000 | 34 | 876 | | |

Table 4.1: Table of Bufferers

After the completion of overall buffer selection, source starts generating messages and upon message generation, each message is sent to appropriate peers for buffering purposes. In this approach, entire topology knowledge is used to send each message to the appropriate buffer.

After sending the generated messages to the bufferers, data dissemination throughout the network starts. Messages are disseminated to the network using epidemic anti-entropy with full membership knowledge. Each peer randomly selects *fan-out* peers from the entire topology and exchanges data with the selected peer. Similar to buffer selection, dissemination of messages also requires the full topology knowledge of the network.

## 4.3 Random Buffer Selection

In this approach, buffers are selected randomly throughout the entire topology and messages are sent to bufferers first. After sending the generated messages to the bufferers, data dissemination throughout the network starts. Messages are disseminated to the network using two different methods: Epidemic anti-entropy with full membership knowledge and epidemic anti-entropy with partial membership knowledge. We call the former as random-full and the latter as random-partial.

## 4.4 Simulation Models

### *4.4.1 Events, Parameters and Message Formats*

In this section we give descriptions for the events, variables and data structures used in the models. Basic data structures are described in Table 4.2, parameters are described in Table 4.3 and events are described in Table 4.4.

| **Data Structure** | **Description** |
|---|---|
| History Information | List of messages that current peer has already received |
| Data Message | Data message received |
| Buffer | Buffer of the current node |
| Neighbor List | List of neighboring nodes in the partial view of current node |

Table 4.2: Table of Data structures

### *4.4.2 Algorithms*

In this section we give descriptions for the algorithms for our previously described simulation models. Alg.3 describes fair-share approach, Alg. 4 describes hash approach and Alg. 5 describes random approach.

| **Variable** | **Description** |
|---|---|
| Message ID | Unique id of each data message |
| Bufferer ID | The id of one of the bufferers corresponding to the message |
| Fan out | Number of nodes chosen in each epidemic round |
| Number of Bufferers | Number of bufferer nodes for the data messages |
| Generation Interval | Time interval of data generation determined by the source node |
| Digest size | Number of entries in the digest message |
| Buffer Capacity | Number of messages that can be stored in the buffer |
| Source ID | Unique id of the source of the message |
| Size of Message | The size of the payload |
| Time to Live | Max number of hops a buffering request can travel |
| TTL counter | Remaining lifetime of buffering request as number of hops |
| Neighbor History | Number of Messages that current neighbor has already buffered |

Table 4.3: Table of Parameters

| **Event** | **Description** |
|---|---|
| Data Generation | Generation of data by the source node |
| Bufferer Selection | Selection of bufferer peers of a data message |
| Epidemics | Exchange of data through networking peers |

Table 4.4: Table of Events

**Algorithm 3** Stepwise Fair-share Buffering with Partial Membership Knowledge

```
for $i = 0$ to $NumberofMessages$ do
  Source generates message $i$
  Buffer Selection
  The neighbor having minimum $NH$ is selected, $TTL$ value is decreased and bufferer discovery process is passed to the selected neighbor with the newly decreased $TTL$
  $TTL$ check
  if $TTL$ value is zero then
    The peer is selected as the bufferer of the message
  else if $TTL$ value is greater than zero then
    The peer continues bufferer selection process. The peer searches for the neighbor history information ($NH$) of its neighbors
    if the neighbor with minimum $NH$ value is the neighbor itself then
      the peer is selected as the bufferer of the message
    else if the neighbor with minimum NH value is a different neighbor then
      then the $TTL$ value is decreased. Go to '$TTL$ check' stage
    end if
  end if
  Piggyback the bufferer id to the message $i$
  Source sends message $i$ to the bufferer through the shortest path between the source and buffering peer (multi-hopping)
end for
```

**Algorithm 4** Hash-based Buffer Selection with Full Membership Knowledge

```
Bufferer Selection
for $i = 0$ to $NumberofMessages$ do
    Map message $i$ over 3 bufferer peers from the network peers, using the hash function, $h_1(i)$, $h_2(i)$, $h_3(i)$
    if Message $i$ has multiple bufferers then
        Equally likely choose one of them
    end if
end for
Data Dissemination
for $i = 0$ to $NumberofMessages$ do
    Source generates message $i$
    Piggyback the bufferer id to the message $i$
    Source sends message $i$ to the bufferer through the shortest path between the source and buffering peer (multi-hopping)
end for
```

**Algorithm 5** Random Buffer Selection

```
Data Dissemination
for $i = 0$ to $NumberofMessages$ do
    Source generates message $i$
    Randomly choose a single bufferer from the entire topology
    Piggyback the bufferer id to the message $i$
    Source sends message $i$ to the bufferer through the shortest path between the source and buffering peer (multi-hopping)
end for
```

Chapter 5

# PERFORMANCE RESULTS

In this chapter, the performance of stepwise fair-share model in different topologies is examined together with reliability and scalability through simulations. Scalability and reliability are necessary issues to be achieved in a network, while uniformity in buffering load and tolerable dissemination delays are also expected. We first define simulation settings and basic simulation parameters. The second and third parts of the chapter cover performance and scalability results for bufferer selection and data dissemination, respectively.

## *5.1 Simulation Settings*

Our models are implemented using JAVA programming language. We develop further the simulation tools generated by [21]. The models consist of a single source peer and the source continuously generates data messages at 20 msg/sec rate and a total of $100,000$ messages are generated. During data dissemination, each peer exchanges its digest with 5 randomly selected peers as fan-out. Size of each message is set to 1000 K-bytes. Each peer has a digest capacity of 500 K-bytes in each dissemination cycle, i.e. it takes two cycles to exchange a single 1000 K-bytes size message between peers. A single bufferer is assigned for each data message unless stated otherwise. Each peer has a buffer capacity of $10,000$ K-bytes, it can store at most 10 messages in its buffer and when new messages arrive to be buffered, the oldest messages are discarded using FIFO scheme.

We perform simulations with hierarchical and power-law topologies, from 1000 peers to $10,000$ peers, using four previously described bufferer selection models; namely, fair-share, hash, random full and random partial approaches. In all our simulations, data is disseminated to the entire network with 100% reliability, meaning that all $100,000$ messages completely reach to every peer in the network.

## 5.2 Topology Properties

Hierarchical model is considered as a good approximation of the Internet topology. The Internet can be viewed as a set of interconnected routing domains where each domain can be classified as either a stub or a transit domain[21]. Stub domains correspond to interconnected local area networks and the transit domains model wide or metropolitan area networks. A transit domain is composed of backbone nodes which are well connected to each other with high bandwidth links. Every transit node is connected to one or more stub domains.

Power law graphs have attracted great interest since the Internet topology exhibits a power law degree distribution [32]. A power law graph is one where the number of nodes with degree $k$ is proportional to $k^{-\beta}$ for some $\beta > 1$. For the mean degree to be finite, we need $\beta > 2$. The main charactheristic in power-law topology is the power-law degree distribution of peers. There are limited number of peers in the network with very high connectivity, while big percentage of the peers in the network have few number of neighbors.

Hierarchical and power-law topologies with various sizes (1000, 2000, 4000, 6000, 8000, 10,000 nodes) are considered with each buffer management approach. Hierarchical topologies are generated using $GT - ITM$ [35] topology generator and power-law topologies are generated using $BRITE$ [36] topology generator, with default link delay values created by the topology generators. The average degree for both hierarchical and power-law topologies is set to 10 manually, in order to make them comparable. The histograms of degrees for size 2000 hierarchical topology and size 2000 power-law topology are given in Fig. 5.1 and Fig. 5.2. Each topology has 2.5 msec. link delay on the average, as illustrated in Fig. 5.3 for hierarchical topology and and in Fig. 5.4 for power-law topology. Dijkstra's shortest path algorithm is used to find the optimum path for routing a message through multiple peers.

## 5.3 Bufferer Selection

In this section, we investigate the stepwise fair-share algorithm [21] with power-law topology. Previously, the algorithm was only investigated with hierarchical topology. In this algorithm, the bufferers are selected through an adaptive scheme considering locality information of the topology, providing a scalable and reliable data dissemination with a fair

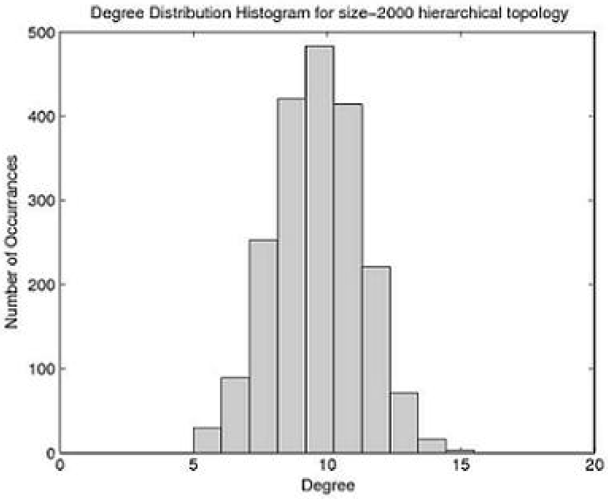

Figure 5.1: Degree distribution for hierarchical topology

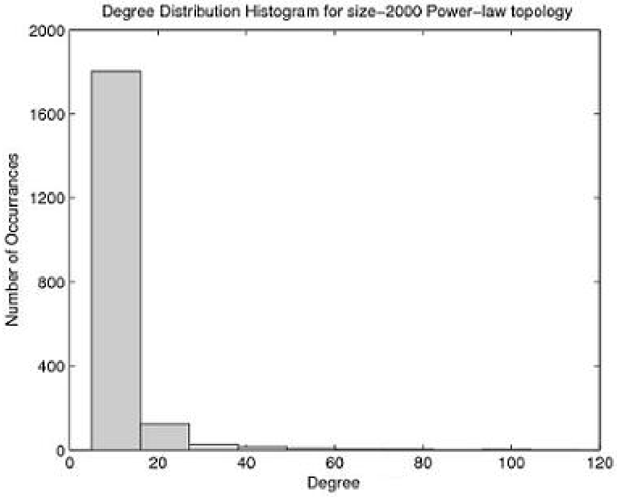

Figure 5.2: Degree distribution for power-law topology

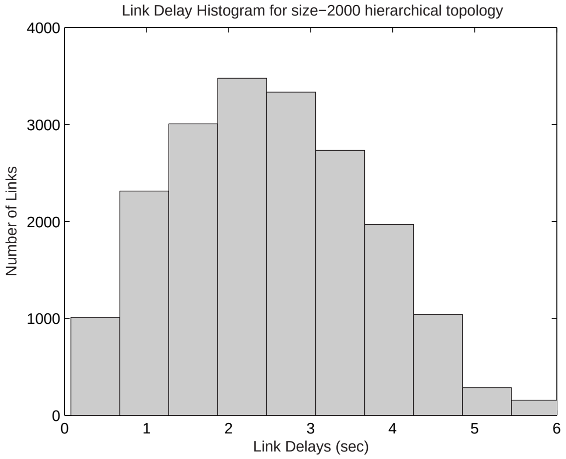

Figure 5.3: Link delay (in msec.) histogram for hierarchical topology

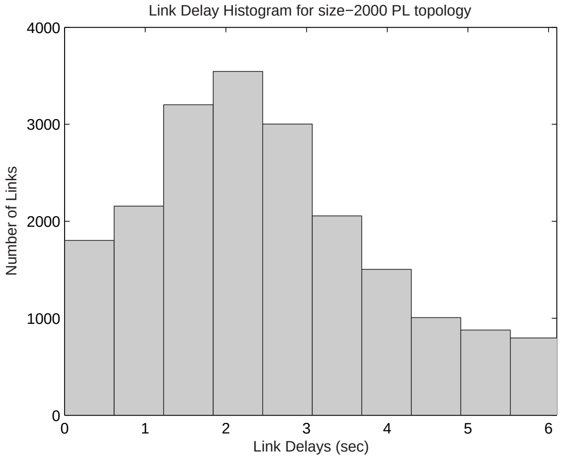

Figure 5.4: Link delay (in msec.) histogram for power-law topology

buffering load over the system. The advantage of the method is that dissemination takes place with only partial neighborhood knowledge which is the motivation behind this algorithm. Our main contribution is analyzing the performance of the algorithm with power-law topology. In this section, we study the quality of uniform buffering in detail. The simulations are performed over a network of 2000 peers with hierarchical and power-law topologies.

We plot the number of messages buffered for each peer, in the network of 2000 peers, with hierarchical and power-law topologies, using four previously described bufferer selection models; fair-share, hash, random full and random partial approaches. Average buffering load per peer is defined by

$$BL_{av} = \frac{Total\ Number\ of\ Messages}{Total\ Number\ of\ Peers}$$

Since total number messages generated is $100,000$ messages and total number of peers in the network is 2000 peers, the optimal buffering load is 50 messages per peer. Simulation results for buffering load performance are given in Fig. 5.5 for hierarchical topology and in Fig. 5.6 for power-law topology. The results for hash and random approaches are very close to each other due to the similarity in buffer selection process and for the ease of visuality, only fair-share and hash approaches are included in the graphs. The results show that fair-share approach performs better in uniformity of buffering load compared with the other approaches. In hierarchical topology, the standard deviation with hash approach is 8.59, while fair-share approach performs a standard deviation of only 0.63. In power-law topology, the standard deviation with hash approach is 8.04, while fair-share approach performs a standard deviation of only 0.17. The reason is that fair-share approach selects bufferers after searching for the best choice, in contrast to the randomness of the selection processes in the other approaches. These results confirm that buffering load performance is similar when hierarchical and power-law topologies are considered.

We scrutinize the performance of stepwise fair-share buffering with different topologies in terms of distributing the buffering load. For this, 1000-node power-law and hierarchical topologies are used. Buffer capacity of the nodes is 10 messages, and $50,000$ messages are disseminated from a single source with rate of 20 msgs/s. The TTL value is set to 20. In Fig. 5.7, the buffering load of the nodes is given for various dissemination percentages (20 - 100%) in power-law and hierarchical topologies. It is observed that stepwise fair-share buffering

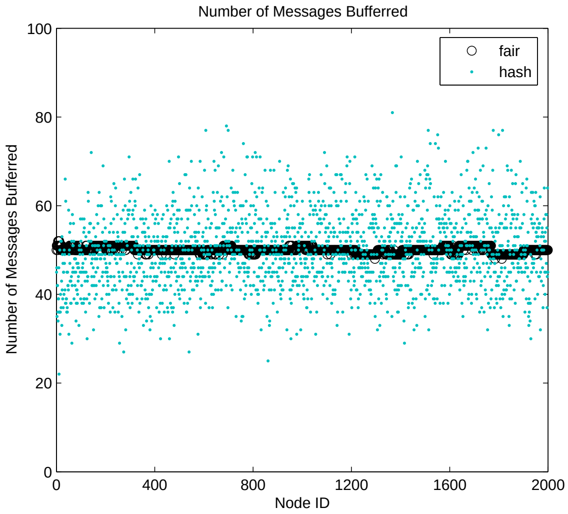

Figure 5.5: Number of messages buffered for hierarchical topology

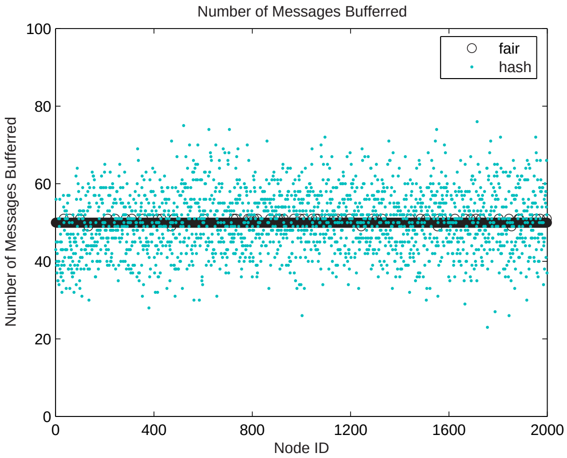

Figure 5.6: Number of messages buffered for power-law topology

provides uniformity over time which would be helpful for reliable data dissemination. In particular, it achieves a more uniform distribution with power-law topology in comparison to hierarchical, for all dissemination percentages. Likewise, the comparison for reliable data dissemination to the entire system for both topologies is given in Fig. 5.8.

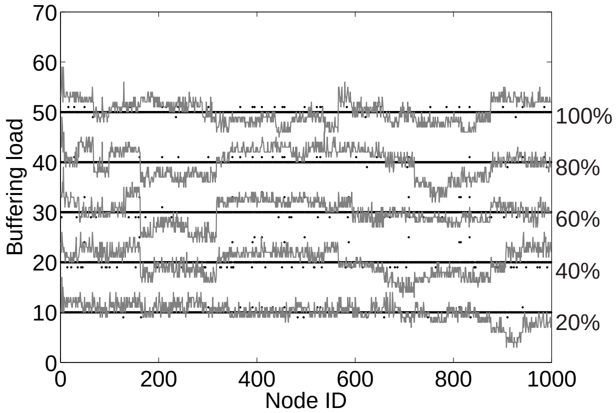

Figure 5.7: Uniformity of the fair-share scheme in time for power-law (black dots) and hierarchical (gray dots) topologies.

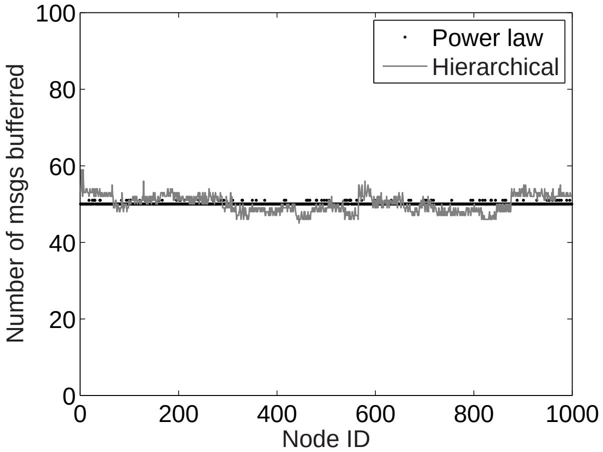

Figure 5.8: Comparison of buffering load distribution: hierarchical and power-law topologies.

For observing scalability, we plot buffering load for each peer, with fair-share approach, for different total number of peers in the network, up to 10, 000 peers. The results are given in Fig. 5.9 for hierarchical topologies and in Fig. 5.10 for power-law topologies. When the network size grows while keeping total number of messages at constant, buffering load per peer decreases inversely proportional with the network size and we observe uniform buffering load in all cases, so the system is scalable.

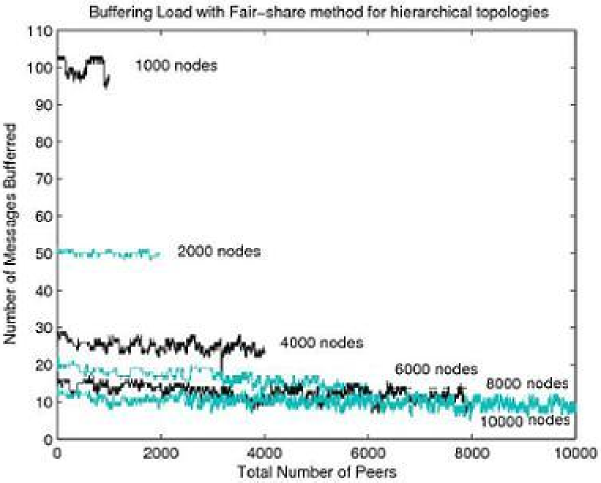

Figure 5.9: Buffering load for hierarchical topology

Due to large quantity of data for all approaches, we present the results statistically, simply using the mean and the standard deviation. We plot the number of messages buffered per peer vs. total number of peers in the network with hierarchical and power-law topologies, up to 10, 000 peers with fair-share and hash approaches, in Fig. 5.11 for hierarchical topologies and in Fig. 5.12 for power-law topologies. The error-bars denote two standard deviations, hence approximately a 95% confidence interval around the mean. The results show that fair-share approach provides a better buffering load compared to the other approaches, since the error-bars are narrower. Moreover, in all approaches, buffering with power-law topology performs better in uniformity of buffering load compared with hierarchical topology. Also, uniformity of buffering load increases when the network size grows.

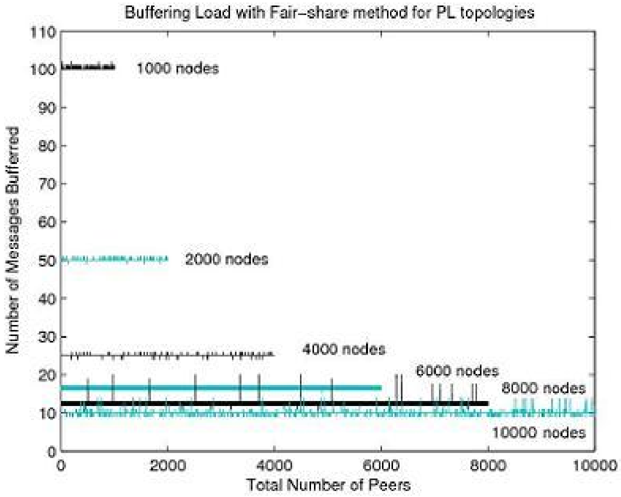

Figure 5.10: Buffering load for power-law topology

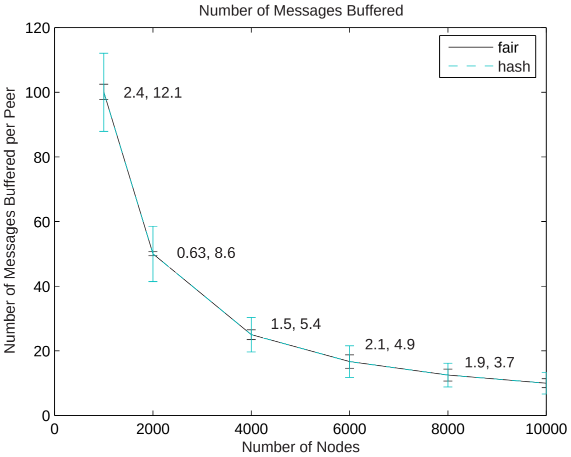

Figure 5.11: Mean number of messages buffered versus group size in hierarchical topology. The pair of numbers denote the width of the error bars for fair-share and hash approaches, respectively.

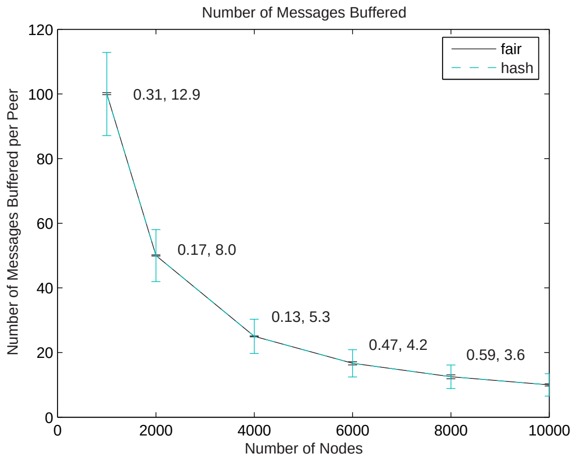

Figure 5.12: Mean number of messages buffered versus group size in power-law topology. The pair of numbers denote the width of the error bars for fair-share and hash approaches, respectively.

## 5.4 Data Dissemination

In this section, for the performance analysis of the system, we analyze delays for data dissemination with different topologies in detail. Dissemination time for each peer is plotted in the network of 2000 peers, with hierarchical and power-law topologies, using our four different methods; fair-share, hash, random full and random partial approaches. Dissemination time is simply the time it takes for the delivery of all the generated messages to each peer. We define average dissemination time per peer as

$$DT_{av} = \frac{Total\ Number\ of\ Messages}{Message\ Generation\ Rate}$$

Total number messages generated is $100,000$ messages and message generation is 20 messages per second, resulting in a dissemination time of 5000 seconds on the average as observed in Fig. 5.13 for hierarchical topologies and in Fig. 5.14 for power-law topologies. There is also delay introduced from buffer selection process for fair-share algorithm. However, this is negligible since it is relatively small compared to data dissemination. It results in about 0.02% more delay on the average, compared to other approaches, as observed in Fig. 5.13 for hierarchical topologies. The results for hash and random approaches are very close to each other due to the similarity in buffer selection process.

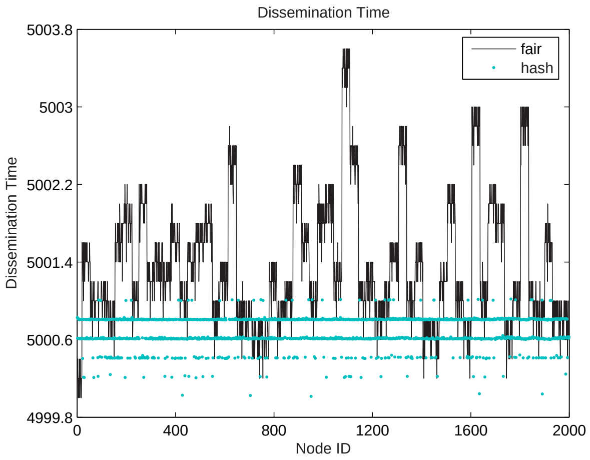

Figure 5.13: Dissemination times (in sec.) for hierarchical topologies

On the other hand, when we consider dissemination times for power-law topology, dissemination performance of fair-share approach is better compared with hierarchical topology. For power-law topologies, both fair-share and hash methods perform similar dissemination times. Moreover, as observed in Fig. 5.14, two different sets of data are overlapping and they are not distinguishable from each other. The reason for better performance in power-law topology is its structure. In power-law topology, there are very few number of highly connected peers, despite most of the peers having only a few number of connections. Better dissemination follows due to these highly connected members. Once the message reaches to highly connected peers, there are sufficiently many routes for the message to spread over the network.

Scalability results for dissemination time performance are given in Fig. 5.15 for hierarchical topology. For power-law topology, dissemination time, for all network sizes is close to each other and almost equal to 5000.8, so there is no need to plot the data. Due to large quantity of data, we present the results statistically, simply using the mean. Scalability is observed as the network size grows. In general, we observe quite equal dissemination times in all cases, with a small standard deviation. Power-law topologies have faster dissemination times compared with hierarchical topologies, due to their structures as explained above. However, the results are very close to each other and differ only about 0.02%.

We also analyze the average message delays in the network for each peer, in the network

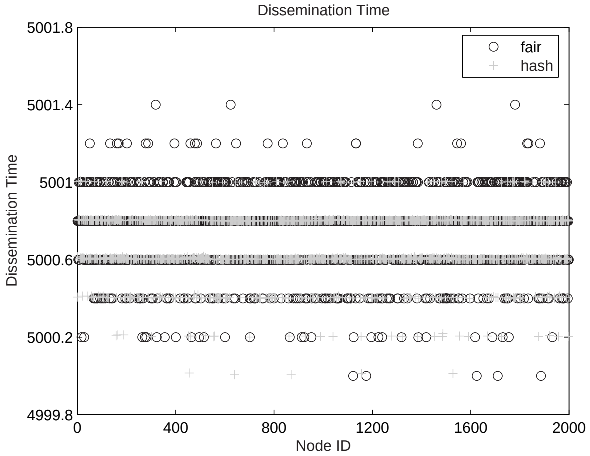

Figure 5.14: Dissemination times (in sec.) for power-law topology

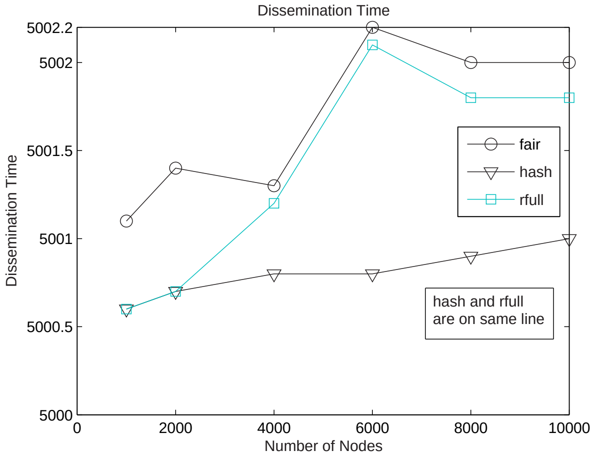

Figure 5.15: Dissemination times (in sec.) for hierarchical topology

of 2000 peers, with hierarchical and power-law topologies, using our four different methods; fair-share, hash, random full and random partial approaches. Message delay is defined as the time it takes from the generation of the message at the source to the delivery of the message at the receiving peer. Average message delay per peer is defined as

$$MD_{av} = \frac{Total\ time\ for\ delivery\ of\ all\ messages}{Total\ Number\ of\ Messages}$$

Simulation results for average message delay performance are given in Fig.s 5.16 and 5.17 for hierarchical topologies and for power-law topologies, respectively. The results for hash and random approaches are very close to each other, and again, only fair-share and hash approaches are included in the graphs. In fair-share approach, the decision process for selection of bufferers results in higher average message delay compared to other approaches. This is more prominent for hierarchical topology as given in Fig. 5.16. However, when considering average message delays for power-law topology structure, message delay of fair-share approach is slightly higher than the other approaches. This is due to the structures of topologies as explained above.

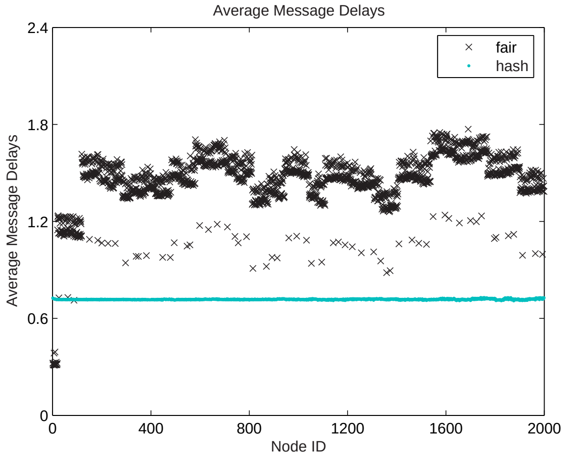

Figure 5.16: Average message delays (in msec.) for hierarchical topology

Scalability results for average message delay performance are given in Fig.s 5.18 and 5.19 for hierarchical topologies and for power-law topologies, respectively. The results for hash and random approaches are very close to each other due to the similarity in buffer selection process. We observe that average message delay increases when the size of the network

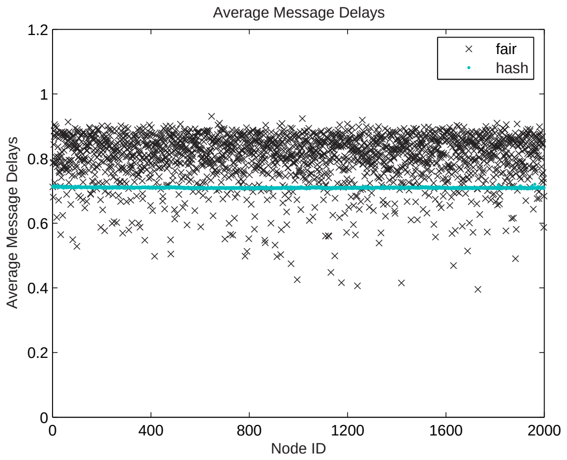

Figure 5.17: Average message delays (in msec.) for power-law topology

increases. This is only a logarithmic increase as expected from epidemic dissemination. The peak in 6000 for the hierarchical topology is only due to randomness in the topology generation process. Finally, for average message delay performance, we observe scalability and faster dissemination in power-law topology compared with hierarchical topology, consistently with previous results.

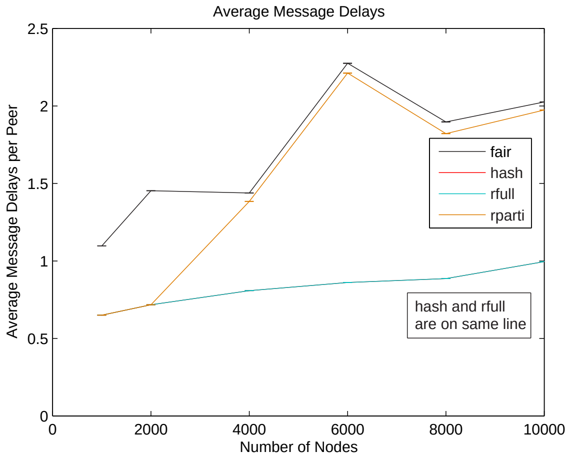

Figure 5.18: Average message delays (in msec.) for hierarchical topology

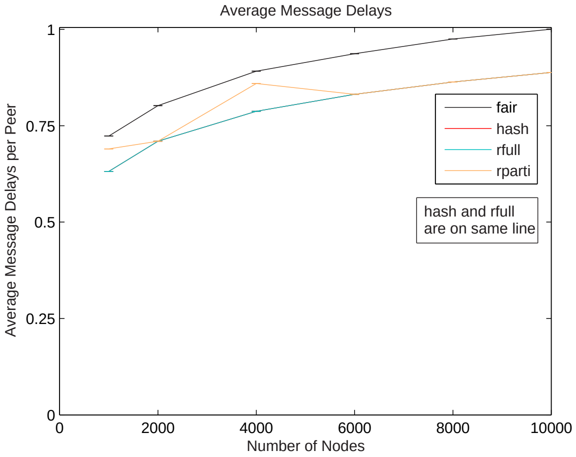

Figure 5.19: Average message delays (in msec.) for power-law topology

# Chapter 6

# CONCLUSIONS AND FUTURE WORK

In this thesis, we have investigated the effect of topology on data dissemination and buffer management, which are two major topics in P2P networking. Hierarchical and power-law topology models are accepted as good approximations of the Internet. Power-law topology is known to model the Internet better compared to hierarchical topology. For data dissemination, we find that topological properties are effective in predicting the rate of diffusion in a P2P network, using our numerical evaluations. For buffer management, we have shown that power-law topologies facilitate better buffer management performance compared to hierarchical topologies, in view of our simulations.

For data dissemination, we have derived an analytical model for pull type anti-entropy approach for SI epidemic data dissemination. We have assumed neighborhood knowledge among peers and data exchange based on proximity. Our model explicitly involves overlay topology through the inclusion of its adjacency matrix. The rate of dissemination is found to be related to the adjacency matrix in a nonlinear way. However, we can explicitly compute the gradient matrix of the function that governs the dynamics of diffusion. In our numerical evaluations, we have investigated the topological properties such as degree distribution and eigenvalues of the gradient matrix over Erdös-Rényi and power-law random graphs. Rather than the maximum eigenvalue, the mean and the standard deviation of all eigenvalues are found to be effective in predicting the rate of diffusion. In practical use, the operator of a P2P network may use our methodology and decide on modifiying network topology by encouring networking peers on a way that will increase the dissemination rate.

For buffer management, the performance of different buffering approaches have been evaluated through simulations for hierarchical and power-law topologies. Reliability and scalability are achieved in all methods. Uniformity in buffering load, dissemination times and average message delays are the basic performance metrics. Stepwise Fair-share Buffering method facilitate better uniformity in distribution of buffering load, in view of our simulations. We expect to have higher delays due to decision process performed for bufferer selection, however, it is also shown that dissemination delay performance drawback is eliminated when power-law topologies are considered. The advantage of the method is that dissemination takes place with only partial neighborhood knowledge which is the motivation of this algorithm. We conclude that Stepwise Fair-share method improves the efficiency of content dissemination, especially in power-law topology structure. In practical use, our methodology can be used for fair and reliable data dissemination in a P2P network.

As the contribution to this thesis for buffer management, performance evaluation of various models with hierarchical and power-law topologies are conducted. Scalability, reliability, dissemination delays and uniformity are considered as basic performance parameters. We have shown that Stepwise Fair-share Buffering method facilitate better uniformity in distribution of buffering load, in view of our simulations. We expect to have higher delays due to decision process performed for bufferer selection, however, it is also shown that dissemination delay performance drawback is eliminated when power-law topologies are considered.

As future work, we aim to include dissemination and bufferer selection for variable size messages and for content with different popularity. Since popular data are shared among many peers and hence they are less likely to be lost in the network, we can propose an algorithm to decrease the buffering of popular messages so that we can reduce the usage of memory resources. Another future direction would be to work on real-time video streaming by dividing real-time data into chunks. Each chunk would be marked with a unique number and tried to be disseminated in FIFO order, so that disseminated data will be ready for real-time streaming. In order to measure the accuracy of our results and our Internet topology models, an application can be developed and deployed on a set of testbed nodes. Then, the simulation results for the buffering algorithms can be compared with those obtained from the testbed.

## BIBLIOGRAPHY

[1] A-M. Kermarrec, L. Massoulie and A. J. Ganesh “Probabilistic Reliable Dissemination in Large-Scale Systems,” IEEE Transactions on Parallel and Distributed Systems, 14(2), February 2003

[2] Y. Drougas, T. Repantis and V. Kalogeraki “Load Balancing Techniques for Distributed Stream Processing Applications in Overlay Environments,” IEEE Proceedings of the 9th International Symposium on Object and Component-Oriented Real-Time Distributed Computing, April 2006, pp. 33-42

[3] Y. Li, Z. Li, M. Chiang and A. R. Calderbank “Video transmission scheduling for peer-to-peer live streaming systems,” IEEE International Conference on Multimedia and Expo, pp. 653-656, 2008

[4] http://www.pplive.com

[5] http://www.ppstream.com

[6] X. Zhang, J. Liu, B. Li and T. P. Yum “CoolStreaming/DONet: A Data-driven Overlay Network for Peer-to-Peer Live Media Streaming,” IEEE Infocom 05, Miami, FL, USA, March 2005

[7] J.A. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D.H.J. Epema, M. Reinders, M. van Steen and H.J. Sips “Tribler: A social-based Peer-to-Peer system,” International Workshop on Peer-to-Peer Systems (IPTPS 2006), Santa Barbara, CA, USA, February 2006

[8] D. Kempe, J. Kleinberg and A. Demers “Spatial Gossip and Resource Location Protocols,” ACM Proceedings, 33rd annual symposium on Theory of computing, pages 163-172, 2001

[9] E. İskender, M. Çağlar and Ö. Özkasap "Analytical Model for Topology Dependence in Peer-to-Peer Anti-Entropy Spreading," International Symposium on Computer Networks (ISCN 08), İstanbul, Turkey, July 2008

[10] A-M. Kermarrec and M. V. Steen "Gossiping in Distributed Systems," ACM SIGOPS Operating Systems Review, 41(5), October 2007

[11] I. Gupta, A-M. Kermarrec and A. J. Ganesh "Efficient and Adaptive Epidemic-style Protocols for Reliable and Scalable Multicast," IEEE Proceedings, 21st Symposium on Reliable Distributed Systems, pages 180-189, 2002

[12] P. T. Eugster, R. Guerraoui, A-M. Kermarrec and L. Massouli "Epidemic Information Dissemination in Distributed Systems," IEEE Publications on Computer, 37(5), pages 60- 67, May 2004

[13] E. Ahi, M. Çağlar and Ö. Özkasap "Stepwise Probabilistic Buffering for Epidemic Information Dissemination," Bio-inspired Models of Network, Information and Computing Systems (Bionetics06), Cavalese, Italy, 2006

[14] L. Rodrigues, S. Handurukande, J. Orlando, R. Guerraoui and A.-M. Kermarrec "Adaptive gossip-based broadcast," IEEE International Conference on Dependable Systems and Networks (DSN03), San Francisco, CA, USA, 2003

[15] J. F. Paris and J. Baek "A Heuristic Buffer Management and Retransmission Control Scheme for Tree-Based Reliable Multicast," ETRI Journal, 27(1), February 2005

[16] K. Guo and I. Rhee "Message Stability Detection for Reliable Multicast," Proc. of the 19th IEEE Conf. on Computer Comm. (INFOCOM00), New York, USA, 2000, pp. 814-823

[17] M. Costello and S. McCanne "Search Party: Using Randomcast for Reliable Multicast with Local Recovery," Proc. of the 18th IEEE Conf. on Computer Comm. (INFOCOM99), New York, USA, 1999, pp. 1256-1264

[18] C. Lindemann and O. Waldhorst "Modeling Epidemic Information Dissemination on Mobile Devices with Finite Buffers," Proc. of the ACM. Int. Conf. on Measurement and Modeling of Computer Systems (SIGMETRICS05), Banff, Canada, 2005, pp. 121-132

[19] B. Koldehofe "Buffer Management in Probabilistic Peer to Peer Communication," Proceedings of the 22nd International Symposium on Reliable Distributed Systems (SRDS03), IEEE, Florence, Italy, 2003

[20] Ö. Özkasap, R. van Renesse, K.P. Birman, and Z. Xiao "Efficient Buffering in Reliable Multicast Protocols," Proc. of the First International Workshop on Networked Group Communication (NGC99), Pisa, Italy, 1999, pp. 188-203

[21] E. Ahi "Design and Analysis of a Novel Buffer Management Model for Reliable Content Dissemination," Koc University, M.S. Thesis, May 2007

[22] Z. Xiao, K.P. Birman, and R. Renesse "Optimizing Buffer Management for Reliable Multicast," Proc. of the International Conf. on Dependable Systems and Networks (DSN02), Washington, D.C. USA, 2002

[23] J.C. Lin and S. Paul "RMTP: A Reliable Multicast Transport Protocol," Proc. of the 15th IEEE Conf. on Computer Comm. (INFOCOM96), San Francisco, USA, 1996, pp. 1414-1424

[24] Ö. Özkasap, E. Ş. Yazıcı, S. Küçükçiftçi and M. Çağlar "Exact Performance Measures for Peer-to-Peer Epidemic Information Diffusion ," ACM SIGMETRICS Performance Evaluation Review, 34(3), pages 6-8, December 2006

[25] Y. Wang, D. Chakrabarti, C. Wang and C. Faloutsos "Epidemic Spreading in Real Networks: An Eigenvalue Viewpoint," Proc. IEEE SRDS, 2003

[26] Y. Drougas and V. Kalogeraki "A fair resource allocation algorithm for peer-to-peer overlays," IEEE Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'05), 4:2853-2858, March 2005

[27] R. Jain, A. Durresi, and G. Babic "Throughput fairness index: an explanation," ATM Forum/990045, Feb. 1999

[28] B. Cohen "Incentives build robustness in BitTorrent," P2P Economics Workshop, Berkeley, CA, 2003

[29] Ö. Özkasap, M. Çağlar and A. Alagöz "Principles and Performance Analysis of SeCond: A System for Epidemic Peer-to-Peer Content Distribution," Journal of Network and Computer Applications, Elsevier Science, 32: 666-683, 2009

[30] M. Draief "Spread of Epidemics and Rumours in Networks," UK Social Network Conference, 2007

[31] D. Chakrabarti, Y. Wang, C. Wang, J. Leskovec and C. Faloutsos "Epidemic Thresholds in Real Networks," ACM Transactions on Information and System Security, 10: 1-26, 2008

[32] A. Ganesh, L. Massouli and D. Towsley "The Effect of Network Topology on the Spread of Epidemics," Proc. of IEEE INFOCOM, 2005

[33] M. W. Hirsch and S. Smale "Differential Equations, Dynamical Systems, and Linear Algebra," Academic Press, 1974

[34] http://www.cs.ucr.edu/%7Eddreier/barabasi.html

[35] http://www.cc.gatech.edu/projects/gtitm

[36] A. Medina, A. Lakhina, I. Matta, and J. Byers "BRITE: An approach to universal toplogy generation," MASCOTS, Aug. 2001

## VITA

EMRE İSKENDER was born in Tokat, Turkey on October 18, 1982. He received his B.Sc. degree in Electrical and Electronics Engineering from Bilkent University, Ankara, in 2004. After graduating from Bilkent University, he worked as a software engineer in different companies. From September 2006 to July 2009, he worked as a teaching and research assistant at Koç University, Turkey and had studied for his research *"Effect of Overlay Topology on Peer-to-Peer Data Dissemination and Buffer Management"* which was sponsored by TÜBİTAK, since January 2008. He has published two papers about *Topology Dependence in Peer-to-Peer Anti-Entropy Spreading* for the following conferences: SİU2008 (Didim, Turkey) and IEEE ISCN2008 (İstanbul, Turkey). He also contributed to journal article *"Stepwise Fair-Share Buffering for Gossip-Based Peer-to-Peer Data Dissemination"* published in Computer Networks, Elsevier Science. He is currently with Digiturk and working as a product development engineer, İstanbul, Turkey.