

A GENERAL PURPOSE SIMULATOR FOR UNSTRUCTURED P2P
NETWORKS AND A SAMPLE APPLICATION

Murat KARAKAYA¹
Özgür ULUSOY²
İbrahim KÖRPEOĞLU³

ABSTRACT

Peer-to-peer (P2P) networks are recently becoming very popular and successful alternative for resource sharing and content distribution. Thus, there have been many proposals to improve their performance. However, since a P2P network consists of many nodes, thousands or millions, it is hard to evaluate the performance of a P2P network and some related protocols analytically using only mathematical models. Most often we need to do simulations. Therefore it is important to have a simulation tool specifically designed for P2P network and protocol simulation. We developed such a tool, and in this paper we present the design and implementation of this simulation tool, which we call GnuSim. The paper includes both the simulation model details and its application in a sample research. This work will be useful for researchers conducting simulation study in P2P systems domain.

Keywords: Peer-to-peer Networks, Simulation, Performance Tests.

YAPISIZ EŞLERARASI AĞLAR İÇİN GENEL MAKSAT BİR SİMÜLATÖR
VE UYGULAMASI

ÖZET

Eşlerarası Bilgisayar Ağları (EBA), son zamanlarda kaynakların paylaşımı ve içerik dağıtımında başarılı ve yaygın bir seçenek haline gelmektedir. Bu nedenle onların başarımını artırmak maksatlı birçok çalışma yapılmaktadır. Ancak, milyonlarca katılımcıdan oluşmasından dolayı EBA'nın başarımının ölçülmesi ve ilgili sorunların analitik olarak matematik modellerle çözülmesi çok zordur. Çoğunlukla benzetim yapılmaktadır. Bu nedenle, EBA ve protokolleri için özellikle tasarlanmış benzetim araçlarına sahip olmak önemlidir. Bu makalede geliştirdiğimiz ve adına GnuSim dediğimiz böyle bir benzetim aracının tasarım ve uygulaması hakkında bilgi verilmektedir. Makalede hem benzetim modeli hem de kullanımı hakkında ayrıntılı bilgi sunulmuştur. Bu çalışmanın EBA sistemleri üzerinde çalışan araştırmacıların benzetim çalışmalarında çok faydalı olacağı değerlendirilmektedir.

Anahtar Kelimeler: Eşlerarası bilgisayar ağları, Simülasyon, Başarım testleri.

¹ Dr., Sistem Yönetim Bilimleri Bölümü, Kara Harp Okulu, Ankara, kmkarakaya@kho.edu.tr

² Prof. Dr., Bilgisayar Mühendisliği Bölümü, Bilkent Üni., Ankara, oulusoy@cs.bilkent.edu.tr

³ Yrd.Doç.Dr., Bilgisayar Mühendisliği Bölümü, Bilkent Üni., Ankara, korpe@cs.bilkent.edu.tr

1. INTRODUCTION

Peer-to-peer (P2P) networks have attracted a significant amount of interest both in the Internet community and in the academic world as a popular and successful alternative to traditional client-server networks for resource sharing and content distribution. There have been many system proposals and applications in the main functional areas of P2P paradigm such as data placement, file lookup, replication, etc. Most of these efforts aim to provide efficient, effective and fast exchange of files between peers by improving existing P2P network architectures and protocols. The effectiveness and the validity of them are mostly tested through simulation experiments. However, a standard simulator to compare different studies does not exist. Furthermore, each work has its own assumptions and level of details in simulating P2P networks, which makes very hard to compare different studies and their claimed success.

In this paper, we introduce a new simulation tool for modeling unstructured P2P networks implementing Gnutella protocol as an extendible, modular and easy-to-use simulator, which is called GnuSim. We simulated an unstructured P2P model with many parameters enabling researchers to observe the effect of different factors such as the number of peers, network topology, content distribution and replication, free riding, time-to-live value, query pattern and query generation rate, etc. The model can be extended to simulate other types of P2P networks as well. Therefore, the simulator can be used to measure and compare the performance metrics of different P2P network models and their functions such as searching and downloading. As a case study, we use GnuSim to simulate a new P2P protocol aiming reducing free riding and compare it with an existing protocol. Thanks to GnuSim, we have been able to realize our proposal and to prove the validity and effectiveness of the proposed protocol. In that study, we also confirmed that GnuSim is an extendible, modular and easy-to-use simulator.

The organization of the paper is as follows. Sections II is devoted to background information about P2P networks. Section III describes the simulation model in detail. We present the results of the sample application of GnuSim in Section IV. The conclusions are provided in the last section.

2. BACKGROUND

In this section, we describe some basic P2P systems concepts and protocols upon which our simulation model is built. Much more details about P2P networks and protocols can be found in (Ritter, 2001; Aberer and Hauswirth, 2001). The observations and findings about P2P network traffic measurement, modeling, and peer behaviors can be found (Markatos, 2002; Jovanovic, Annexstein and Berman, 2001; Saroiu, Gummadi, and Gribble, 2002; Gummadi, Dunn, Saroiu, Gribble, Levy, and Zahorjan, 2003; Ripeanu, Foster, and Iamnitchi, 2002).

We focus on unstructured P2P networks like Gnutella, because of their popularity and well-known protocols (Clip2, 2001). Unstructured P2P networks have the distinct properties that can be summarized as (Aberer and Hauswirth, 2001):

- no central coordination
- no central database
- no peer has a global view of the system
- global behavior emerges from local interactions
- all existing data and services should be accessible
- peers are autonomous and anonymous
- peers and connections are unreliable

These features enabled unstructured P2P networks to be very successful, but also brought some problems. Among the problems of such networks is the so-called reputation problem. In an unstructured P2P network such as Gnutella, peers interact with unknown peers and have no information about their reputations. Thus, peers do not have enough past experience to know if they can trust the other peers and their services.

2.1. A P2P Generic Scenario

The basic service that a P2P network provides is answering user queries and enabling file downloads. A requesting peer may send queries to the P2P network, wanting the network to answer the query. The query can be, for example, "list me the source peers that have the file x". The

outcome of this request can be one of the following: success (the P2P system could find the file and respond with a query hit); failure (the P2P system could not find a node sharing the file in the search process). If the peer could get a list of source peers that have the file, the requesting peer then can select one source peer and try to download the file from that source peer. Again, the outcome of this request can be one of the following: success (the source peer really has the file and provides the file, the network resources are enough for the download operation, and the download is successfully realized); failure (the source peer lied by sending a query hit and it really does not have the file, or the network resources are not enough to realize the download operation).

2.2. Phases in P2P communication

In an unstructured P2P network, a peer may go through four main phases:

- **Connection phase:** In this phase, the peer tries to find some other peers which have already been connected to the P2P network. On finding some peers, it announces its existence to these peers. These connections will be used to broadcast any search requests of the peer. Furthermore, connected peers may communicate with the peer for the search requests initiated by other peers.
- **Search Phase:** The peer needs a file and initiates its search operation by broadcasting a search message through its neighbors. It then waits for replies.
- **Downloading Phase:** If the peer has received a hit message, then it may begin to download the file from the source peer through a direct connection.
- **Local Search and Routing Phase:** The peer can have some search queries delivered to itself by neighboring peers. It first checks its local resources. If it has the file it returns a hit message to the neighboring node. Either it has the file or not, it decreases Time-To-Live (TTL) value of the search message, and if TTL value is greater than 1, it forwards it to all neighbors other than the one which has delivered the search. If any hit message arrives, the peer routes it back to the requesting peer.

KARAKAYA-ULUSOY-KÖRPEOĞLU

These phases are implemented with descriptors in the Gnutella Protocol (Clip2, 2001) (See Table 1).

Table 1 GNUTELLA protocol descriptors

Descriptor	Description	Content
Ping	Used to actively discover hosts on the network. A servent receiving a Ping descriptor is expected to respond with one or more Pong descriptors.	Nothing
Pong	The response to a Ping. Includes the address of a connected Gnutella servent and information regarding the amount of data it is making available to the network.	IP and port of responding host, number and size of files shared
Query	The primary mechanism for searching the distributed network. A servent receiving a Query descriptor will respond with a QueryHit if a match is found against its local data set.	Minimum speed requirement of the responding host;search string
QueryHit	The response to a Query. This descriptor provides the recipient with enough information to acquire the data matching the corresponding Query.	IP and port, speed of responding host; number of matching files and their indexed result set
Push	A mechanism that allows a firewalled servent to contribute file-based data to the network.	Responding host id; file index; IP and port of requesting peer

2.3. Free Riding Implemented in Scenario

In an ideal P2P network use case, peers share their content with other peers and a file that is downloaded by a peer is automatically opened for sharing with other peers. However, peers can, and frequently do, abstain from sharing any content to economize on their own resources such as bandwidth. Therefore, the primary property of P2P systems, the implicit or explicit functional cooperation and resource contribution of peers, may fail and lead to a situation called free riding. That is, free riding (FR) means exploiting P2P network resources (through searching, downloading

objects, or using services) without contributing to the P2P network at desirable levels.

User traffic on Gnutella network is extensively analyzed in (Adar and Huberman, 2000) and it is observed that 70% of peers do not share any file at all. Furthermore, 63% of the peers who share some files do not respond to any queries. That is, they are sharing some files but nobody is interested in them and therefore no queries are generated searching for these files. Another interesting observation is that 25% of the peers provide 99% of the whole content in the network. Similarly, in a more recent work, Saroui et al. confirm that there is a large amount of free riding in Gnutella network as well as in Napster (Saroui, Gummadi, and Gribble, 2002). An interesting observation in that study is the fact that 7% of the peers together provide more files than all of the other remaining peers.

Observing the existence of high degrees of free riding in P2P networks, researchers suggest that free riding may be an important threat against the existence and efficient operation of P2P networks. For example, Adar and Huberman argue that free riding leads to degradation of the system performance and adds vulnerability to the system (Adar and Huberman, 2000). If this trend continues copyright issues might become moot compared to the possible collapse of such systems. Therefore, considering the importance of free riding for P2P networks, in the simulation model, we provide several parameters to simulate FR phenomena and observe its effect on the system, which will be discussed in the following sections.

2.4. Performance Metrics

As stated before, this work aims to enable researchers to observe the performance of a P2P network under different scenarios and different mechanisms. Therefore, we set up three parameters for Quality of Service (QoS) and their metrics. For example, a work intending to implement a scheme reducing free riding in a P2P network may use these metrics. Researchers may want to observe that Quality of Service (QoS) for non-free riders is increased, while being diminished for free riders. Thus, we provided the following QoS parameters and related metrics to evaluate the performance of a P2P system that applies a proposed method. The researchers can easily extend these metrics. We classify QoS parameters under three topics. Summary information about parameters and metrics is given in Table 2.

Table 2 Summary of QoS Parameters and Metrics

QoS Parameter	Observed Metric(s)
Availability	# downloads
Availability	# unsuccessful downloads
Availability	Query Hit Ratio
Load Sharing	# uploads
Scalability	# P2P messages

Below, details of each parameter are presented. All these metrics are observed during simulation and output in the simulation report.

- **Availability:** The availability of content and services in P2P network can be an important issue. For example, if we consider upload capacity of peers, we may recognize that some peers may have upload bottleneck due to high level of downloading requests. When they reach the limit of upload capacity, they begin to refuse new requests.

The number of refused download requests (unsuccessful downloads) may be used as a metric for availability of content. Another metric for availability parameter may be the number of downloads. Furthermore, Query Hit Ratio can be calculated using number of files requested to download and number of queries submitted.

- **Load Sharing:** A large number of search and download operations may go towards few peers and this may lead a bottleneck. In ideal case in a P2P system, the load on peers can also be shared by peers. This will help the system to be more efficient so that larger number of search queries and download operations can be executed on the system successfully. Specifically, the number of uploads done by each peer type may be used as a sample metric for this reason.

- **Scalability:** One of the important scalability issues in P2P networks is the flooding of the messages. As the number of messages routed in the network increases, each peer needs to handle more messages. Network congestion can be occurred for large amounts of messages. Network congestion can affect several services such as login to network, querying for the content, and downloading files. Therefore, we

decide to observe the number of P2P network protocol messages as a metric related to network congestion.

3. ASSUMPTIONS AND PARAMETERS OF THE SIMULATION MODEL

We have implemented an event-driven P2P network and protocol simulator using CSIM 18 (Schwetman, 1991) simulation library and C++ programming language on the WINDOWS OS. The basic characteristics of the model are set to be similar to those of Gnutella network by implementing the protocol described in (Clip2, 2001). In the following subsections, details of the important parameters and related assumptions are provided as classified into those subsections: Network, Peers, Content, and Request. The parameter values used in simulation experiments are summarized in Tables 3, 4, 5 and 6.

3.1. Network

- **Network Topology:** The network topology defines the connectivity between peers. This configuration can be created by using a topology generator. The simulation program reads topology information from an input file.
- **Messaging:** There are two mailboxes within each peer. One mailbox is used for P2P network messages, and the other is used for download requests and downloads. In this way, each mailbox simulates a port in Gnutella and TCP/IP protocol stack running on a peer.
- **Connection Duration:** A peer is assumed to stay connected in the network during the whole simulation lifetime. Pinging Frequency: To check the validity of the connections with its neighbors, each peer submits a PING message at every PINGFREQ seconds.
- **Time-To-Live (TTL):** In unstructured P2P networks, messages are broadcasted into the network. The TTL parameter is a technique used to limit the broadcast horizon in the network. In the simulation, we assume that maximum TTL value for any P2P protocol message may be set to 7.

Table 3 Topology Parameters

Parameter	Definition	Default Value
MESHEDGE	Number of peers in one edge of mesh, if mesh structure is used for topology generation.	20
NUMPEERS	Total number of peers in the simulation.	MESHEDGE * MESHEDGE
maxNoOfCon	Maximum number of neighbors that a peer can be connected to.	Depends on topology

3.2. Peers

- **Peers and Peer Types:** We simulate a population of peers, NUMPEERS, constituting both free riders and contributors. Peers are grouped according to the given NUMBER OF PEER TYPES parameter and the corresponding properties that are given in Table V. We selected the default values in accordance with the observations done in [1].

- **Ratio of Free Riders:** At the beginning of the simulation, peers are grouped into different types according to the NUM OF PEER TYPES parameter. The number of peers in each type is determined according to the POPULATION RATIOS parameter considering the total number of peers (NUM PEERS). For each peer type, we can set FREE RIDING TYPE to determine FR type of peers in that group. The possible values of the FREE RIDING TYPE parameter are: NONE, NON CONTRIBUTOR, CONSUMER, DROPPER, and MIXED. MIXED means that the peers in that type are equally distributed to each of the three free riding types. Peers' free riding types will not change during a single run of a simulation (i.e. during the simulation lifetime).

- **Upload Bandwidth Capacity:** We assume that each peer has a limited bandwidth capacity to download and upload files. Download capacity is assumed to be 1. That means there is only one download operation that can be executed at a time. However, a peer can do more than one uploads at the same time which is limited by the value of NO OF MAXIMUM UPLOADS parameter.

- **Attempts:** If the number of a peer's downloads exceeds NO OF MAXIMUM UPLOADS, it can refuse more uploads. If a requesting peer

is refused by a resource peer, it can try another source peer if there is any in the queryHitList. MAX DL ATTEMPT NUMBER specifies how many times a peer should try to download the same file from different source peers if any peer refuses to upload the requested file.

Table 4 Peer Type Parameters

Parameter	Definition	Default Value
NUM OF PEER TYPES	Number of peer types in the simulation.	3
POPULATION RATIOS	Population ratios of each peer type.	{0.10, 0.20, 0.70}
FREE RIDING TYPE	Free riding types of each peer type.	{NONE, NONE, MIXED}
SHARED FILE RATIOS	Ratio of shared files of each peer type to total files in the simulation	{0.87, 0.12, 0.1}
NO OF MAXIMUM UPLOADS	Maximum number of uploads a peer can provide at the same time.	{3, 3, 3}
QUERY GENERATION MEAN	The mean value of the exponential distribution that defines the time between two consecutive generated queries.	{60, 60, 60}
CONSUMER QUERY GENERATION MEAN	The mean value of the exponential distribution that defines the time between two consecutive queries generated by Consumer peers.	60
REPLICATION	If peers' REPLICATION property is set "true" then the downloaded files are replicated and shared.	{true, true, false}

3.3. Content

- **Content Distribution:** We distribute the content to peers uniformly. First of all, peers are grouped into different types based on the NUM OF PEER TYPES parameter. Later, according to the given SHARED

FILE RATIOS parameter, the number of files to be distributed for each peer type is calculated. At last, for each type of peers, the determined number of files is distributed uniformly. However, if a free rider peer is specified as a dropper or a non-contributor, no files are distributed to it. The files that are not handled by this kind of peers are redistributed to consumer peers of the same peer type. Furthermore, at the beginning of the every simulation run, the distribution of the files to the peers is the same.

- **Content Replication during Simulation:** The settings given above are valid for the beginning of the each run of the simulation. During the simulation, the settings can be changed according to peers' property to replicate the downloaded files. If peers' REPLICATION property is set to true then the downloaded files are replicated and shared. Therefore, the content distribution in the system would be dynamic during the simulation time.

- **Size of files:** We assume that the sizes of all the files are the same and their download times are equal as defined by DOWNLOAD TIME parameter.

- **Uniqueness of the content:** At the beginning of the every simulation run, the distinct files (DISTINCT FILES) are copied according to given three parameters: COPY, RANGE and SKEWNESS. The first parameter specifies the number of copies of each distinct file. For example, if the number of distinct files (DISTINCT FILES) is 100 and the COPY parameter is 2 then, it means that the total number of files (TOTAL FILES) in the simulation would be 200 and there are 2 copies for each file.

Table 5 Content Parameters

Parameter	Definition	Default Value
DISTINCT FILES	Total number of distinct files in the simulation.	NUMPEERS * 10
COPY	Number of copies of each distinct files.	2
RANGE	Range of files to be replicated more than COPY.	0
SKEWNESS	Number of extra copies of the files in the RANGE.	0
TOTAL FILES	Total number of files in the system.	DISTINCT FILES * COPY + RANGE * SKEWNESS

3.4. Request

- **Request-File Matching:** We assume that the system replies the queries with exact matches only.
- **Request Pattern:** Peers randomly (uniform distribution) select a file to be requested from the P2P network. After selecting a file to be requested, it is checked if the peer itself has the file. If the peer does not have the file then it generates a Query message and submits the message to its neighbors. All files have equal probability for being requested.
- **Request Generation Rate:** The inter arrival time distribution of requests generated by a peer follows exponential distribution with a mean value QUERY GENERATION MEAN which is supplied to the simulation as a peer type parameter to observe the effect of the request pattern.

4. GNUSIM and a SAMPLE APPLICATION

We used GNUSIM in two different studies for simulating a free riding P2P environment (Karakaya, Korpeoglu and Ulusoy, 2008a and 2008b). Here we will summarize the first study (Karakaya, Korpeoglu and Ulusoy, 2008a) in which we propose to modify existing Gnutella protocol by creating a novel connection-based P2P protocol, the P2P Connection Management Protocol (PCMP), that will help to reduce the free riding effects on a P2P

network and discourage free riding. The solution includes a new connection type and an adaptive connection management that dynamically establishes and adapts a P2P network topology considering the contributions of peers. The aim of the protocol is to bring contributing peers closer to each other on the adapted topology and to push the free riders away from the contributors. It uses a new connection type to connect peers. The new connection allows the requests (queries) to be passed in only one direction. Our scheme manages those types of connections so that, eventually, contributors become more close to each other in the network, and free riders become isolated.

In order to implement our proposal, we slightly modified GNUSIM to realize the PCMP. We implemented two versions of PCMP. In the first version, the number of downloads or the number of hit messages provided by a neighboring peer is employed to decide how to manage its connection. For example, the connection with the least number of downloads or hit messages provided is selected to be pruned. We call this version Contribution-based PCMP (C-PCMP). In the second version, the time of the last download or the time of the last Query Hit provided from the neighboring peer is used as a parameter to manage the connections. For example, the connection with the oldest time of the last download or hit messages provided is selected to be cut off. We call this version the Time-based PCMP (T-PCMP).

We compare the performance results of the protocols under similar network and user characteristics. Thanks to GNUSIM we were able to show through simulation experiments that there is a significant improvement in performance for contributing peers in a network that applies our protocol. Here, we summarize the simulation settings and the results used in the work and further details can be found in (Karakaya, Korpeoglu and Ulusoy, 2008a).

For this work, we simulated a P2P network of 900 peer nodes. The peers were interconnected to form a mesh topology at the beginning of a simulation run. We assumed that there were two types of peers in the simulated network: contributors and free riders. The properties of each peer type are summarized in Table 6. There were 9000 distinct files, with four copies of each, distributed to the peer nodes at the beginning of each simulation run. These 36000 files were distributed among the peers and shared according to the file sharing ratios shown in Table 6.

Table 6 Properties of peer types

Property	Contributors	Free riders
Population ratios	30%	70%
Ratio of shared files of each peer type to total files	99%	1%
Peers replicate the files they have downloaded	True	False
Mean time between queries (exponentially distributed)	60 time units	60 time units
Maximum simultaneous uploads	10	10

We assumed that each file was of the same size and could be downloaded in 60 units of simulation time. During a simulation run, peers randomly selected files to search for download, and they submitted search queries for them. The inter-arrival time between search requests generated by a peer followed an exponential distribution with a mean of 60 time units. Each peer's upload capacity (the number of simultaneous uploads the peer could perform) was limited to 10. If a peer reached its upload capacity, any new upload requests were rejected. The querying peer could then try to download the file from another peer, selected from a list obtained from the Query Hit message. We assumed that the querying peer would repeat the same request a maximum of three times. After that, the peer would give up and could initiate a new search for another file. Simulation experiments were run for 4000 units of simulated time. Each simulation was repeated 10 times and plotted on a 95% confidence interval.

Using GNUSIM we evaluated the effectiveness of each protocol in terms of the performance metrics described in Section 2.4. Below we provide sample results which enabled us to claim the effectiveness of the proposed protocols over the Gnutella protocol.

- **Downloads of free riders.** As Figure 1 depicts, the number of downloads by free riders dropped when PCMP was applied. PCMP decreases free riders connections towards contributors, and this reduces the chance of getting a hit on the queries. In this way, the number of downloads by free riders is reduced. Both C-PCMP and T-PCMP reduce the number of downloads. C-PCMP caused a 14% reduction, whereas T-PCMP achieved a 16% reduction.

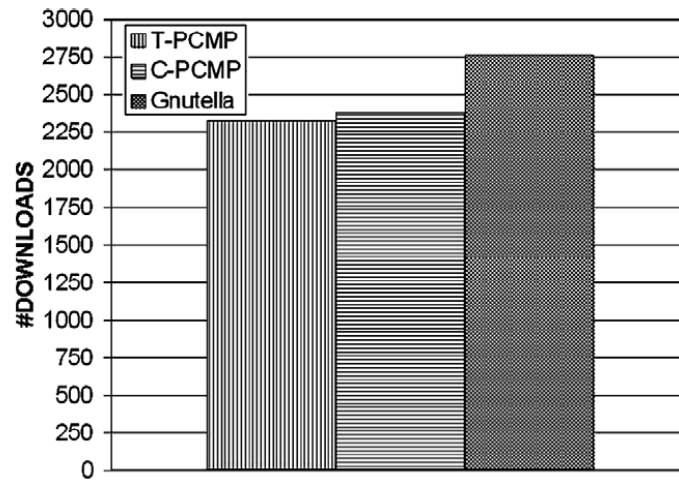


Figure 1 Decrease in free riding peers' downloads.

- **Downloads of contributors.** It is desirable to increase the number of downloads for contributors. As shown in Figure 2; applying the PCMP methods achieved an increase in downloads done by contributors by 51%. While TPCMP yielded an improvement of about 51%, the improvement when C-PCMP was used was about 46%.

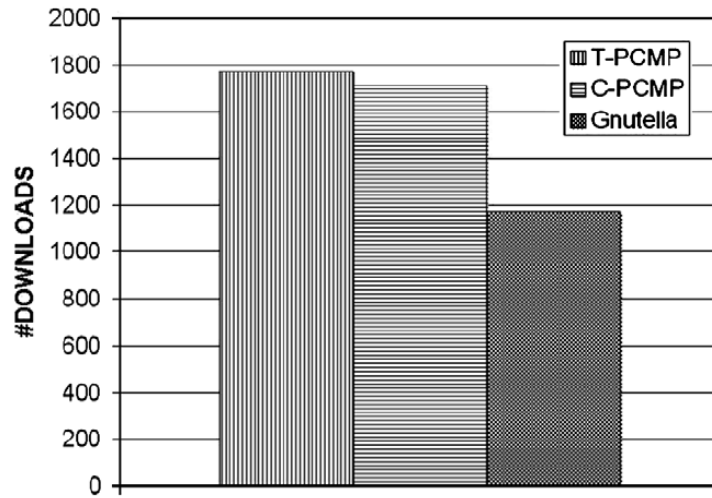


Figure 2 Increase in contributors' downloads.

- **Download cost.** The load on a contributor can also be defined as the ratio of its uploads to its downloads. The results of our experiments show that our PCMP methods also cause a reduction in the download cost of contributors. As shown in Figure 3, both T-PCMP and C-PCMP achieve a reduction of about 30% in the download cost for contributors.

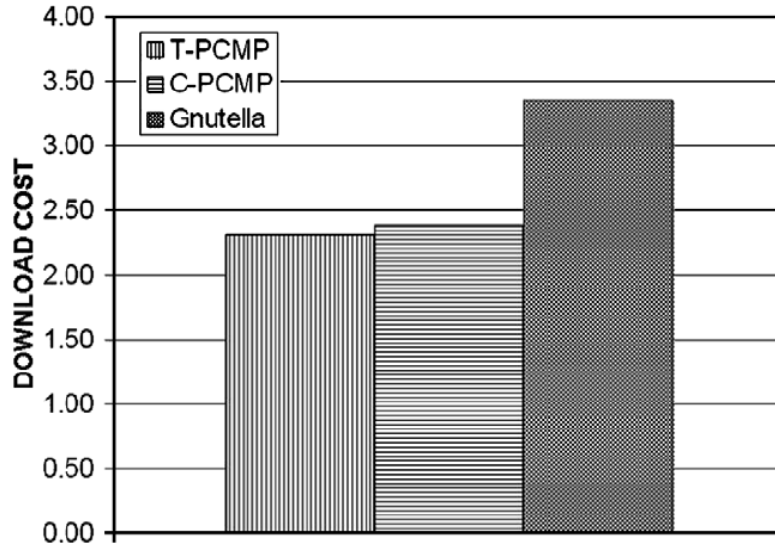


Figure 3 Decrease in contributors' download cost.

- Number of P2P protocol messages.** The number of P2P protocol messages transmitted in the network is an important factor affecting scalability and bandwidth efficiency. PCMP results in a reduction of up to 36% in the number of transmitted P2P protocol messages (Query and Query Hit messages) originating from and destined for the free riders (Figure 4). This result shows that applying the proposed PCMP helps a P2P network to handle more peers with less P2P messaging overhead and the system becomes more scalable with respect to the peer population. The reduction of control traffic in a P2P network also means a reduction in the overhead imposed on the underlying infrastructure. This reduction translates to a better utilization of available bandwidths and to a decreased processing load on each peer.

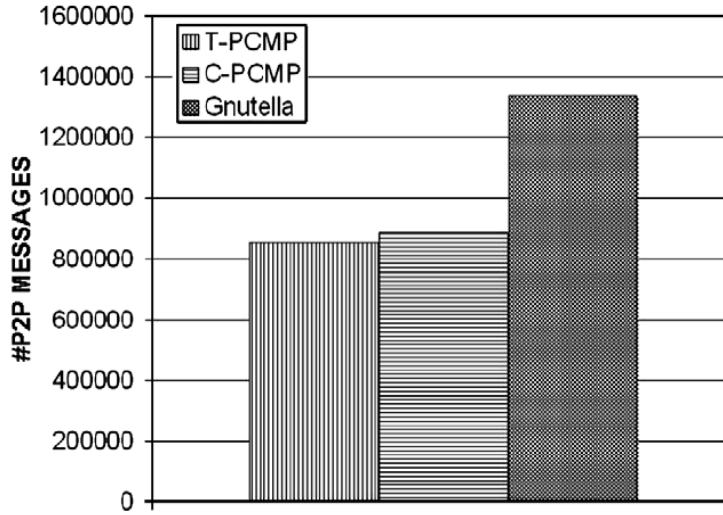


Figure 4 Decrease in P2P messages from free riders.

In addition to simulating new protocols and comparing their performance results with the existing Gnutella protocol, we also conducted several tests to observe the effects of possible attacks to them. For example, a malicious peer does not comply with the proposed PCMP rules. In order to observe the effects of this possible attack, we designed a new simulation setting. In the new simulation, we assumed that all free riders would reject obeying the connection management rules. As seen in Figure 5, this attack does not adversely affect the download performance of the contributors as compared to the results given in Figure 2. On the contrary, the contributors can download slightly more files, because they become more closely connected to each other.

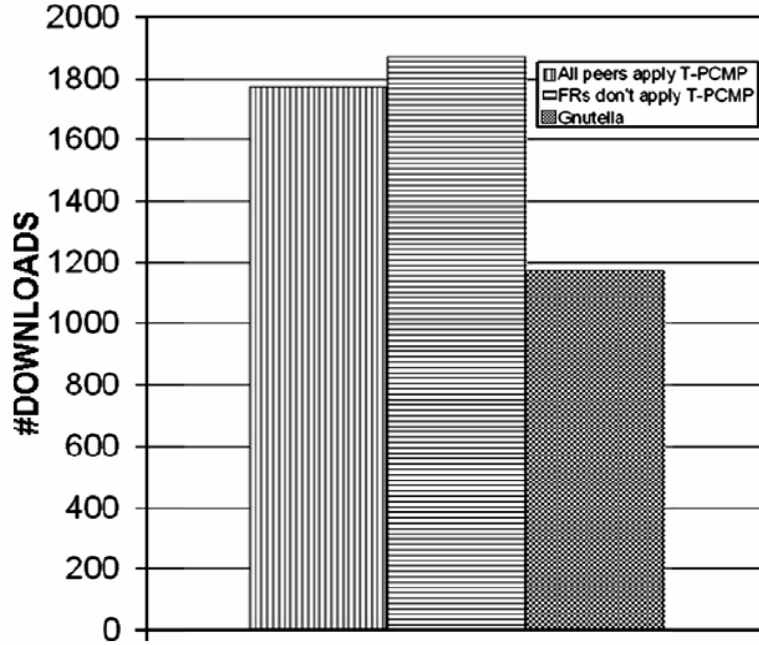


Figure 5 The number of contributors' downloads when free riders are non-cooperative

5. CONCLUSION

In this work, we have designed and implemented a simulation tool to model an unstructured P2P network such as Gnutella network. We have simulated the Gnutella protocol in details. Our model is sophisticated enough to observe and compare the performance of a P2P network under various parameters and their different values. To test the validity of the model and effectiveness of the simulator we used GnuSim in simulating a free riding environment and a proposal against free riders. In the study, we observed that GnuSim is successful in modeling and simulating different P2P protocols under various scenarios. We believe that researchers can benefit from the GnuSim simulator in many ways. GnuSim can be acquired freely by requesting from the corresponding author.

REFERENCES

- Adar, E., Huberman, B.A., 2000. Free Riding on Gnutella.
http://www.firstmonday.dk/issues/issue5_10/adar.
- Aberer, K. and Hauswirth, M. 2001. Peer-to-peer information systems: concepts and models, state-of-the-art, and future systems. In ***Proceedings of the 8th European Software Engineering Conference Held Jointly with 9th ACM SIGSOFT international Symposium on Foundations of Software Engineering***. ESEC/FSE-9. ACM, New York, NY, 326-327
- Clip2, 2001. The Gnutella Protocol Specification v0.4 (Document Revision 1.2), ***Limewire website***, <http://www9.limewire.com/developer/gnutella/protocol0.4.pdf>.
- Gummadi, K., Dunn, R., Saroiu, S., Gribble, S., Levy, H. and Zahorjan, J. 2003. Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload, ***Proceedings of the 19th ACM Symposium on Operating Systems Principles***, (SOSP-19).
- Jovanovic, M., Annexstein F.S. and K.A. Berman, 2001. Scalability Issues in Large Peer-to-Peer Networks - A Case Study of Gnutella, ***Technical Report***, University of Cincinnati.
- Karakaya, M., Korpeoglu, I., and Ulusoy, Ö., 2008a. A Connection Management Protocol for Promoting Cooperation in Peer-to-Peer Networks, ***Computer Comm.***, vol. 31, no. 2, pp. 240–256.
- Karakaya, M., Korpeoglu, I., and Ulusoy, Ö., 2008b. Counteracting Free Riding in Peer-to-Peer Networks, ***Computer Networks***, vol. 52, no. 3, pp. 675–694.
- Markatos, E. P. 2002. Tracing a Large-Scale Peer to Peer System: An Hour in the Life of Gnutella. In ***Proceedings of the 2nd IEEE/ACM international Symposium on Cluster Computing and the Grid CCGRID***. IEEE Computer Society, Washington, DC, 65.
- Saroiu, S., Gummadi, K., and Gribble, S., 2002. A Measurement Study of Peer-to-Peer File Sharing Systems, ***Proceedings of the Multimedia Computing and Networking***.
- Ripeanu, M., Foster, I., and Iamnitchi, A., 2002. Mapping the Gnutella Network: Properties of Large-Scale Peer-to-Peer Systems and Implications for System Design, ***IEEE Internet Computing***, Journal special issue on peer-to-peer networking, Volume.6.
- Ritter, J., 2001. Why Gnutella Can not Scale. No, Really, ***http://www.darkridge.com/jpr5/doc/gnutella.html***.
- Schwetman, H., 1991. CSIM: A C-based, process oriented simulation language, ***Proceedings of the 1991 Winter Simulation Conference***, pp. 387-396.