Controlling Free Riding using Extended Point Based Incentive Mechanism in Peer-to-Peer Networks

Bikash Upadhyay
Computer Science and Engineering
Motilal Nehru National Institute of Technology
Allahabad, India
bikash4u06@gmail.com

Shraddha Dixit
Computer Science and Engineering
Motilal Nehru National Institute of Technology
Allahabad, India
bikash4u06@gmail.com

***Abstract*— Over the years of the emergence of peer-to-peer systems a phenomenon named 'Free Riding' has affected its performance and become a serious threat to be resolved. Free riding phenomenon was first felt in the study of Gnutella. As the number of free riders increases, the performance of the overall system decreases as free riders use resources without giving anything to the system. To solve the problem of free riding we have developed an Extended Point Based Incentive Mechanism (EPBIM) through which the users are given some incentives to share some files into the system. Our scheme attracts the users to share more and more as it gives more points to the users which provide responses for some queries of other users in the system. This scheme is different from other incentive based schemes in the way that it uses some new parameters for incentives like the time user spends in the network, upload speed of the user, how many simultaneous uploads a user is providing, how many query responses a user is giving and query forwarding of other users in the network. Our simulation results show that this scheme reduces the percentage of free riders to a great extent therefore increasing the performance of a peer-to-peer network.**

Keywords: Free riding; Peer-to-Peer networks; Incentive mechanism; Online time

1. INTRODUCTION

The Internet has become an integral part of world's infrastructure facilities. It is the biggest infrastructure on which different virtual networks reside. An overlay network is a virtual network which is built on top of one or more existing networks. Nodes in the overlay networks can be thought of as being connected by virtual or logical links, each of which corresponds to a path, perhaps through many physical links, in the underlying network. There is much interest in emerging P2P overlay networks because they provide good properties, such as self-organization, decentralization and scalability [1, 3, 7, 13]. Peer-to-peer overlay systems are different from the client-server systems by having symmetry in roles where a client may also be a server.

A peer-to-peer network is any distributed network architecture comprised of nodes which make a part of their resources available to other peers in the network without the need of the central coordination [1, 2, 4]. Peer to-peer networks follow the mechanism of sharing resources by direct exchange between peers in the network. The peers in this network can share information, disk storage, processing power and bandwidth. The peer-to-peer network takes advantage of computing power, disk space and connectivity of all the users allowing their collective power to benefit all of them only. Since peer-to-peer networks are dominating in the Internet traffic, there have been many attempts to implement it in different fields in different ways. Peer-to-peer networks follow the scheme in which there is no centralized server present but all the nodes in this network act as both client and server depending on the situation. This kind of network is dynamic in the way that the nodes can join and leave the system at any time with no restriction.

Basically the main functions of a peer-to-peer file sharing system are join, leave, publish and search for the object in the network. Join and leave are the terms for joining and leaving the system as their meaning implies. Publish is the function when a node joins the network and all its data gets published in the network, while search is whenever a node wants some file then he starts a query with the name of the file and this file is searched in the network. Peer-to-peer networks follow a request-response protocol.

The development of peer-to-peer networks started in the year 1999 and since then has been rapidly growing in popularity. Research, as of June 2006, over 60% of all consumer Internet traffic is peer-to-peer. With the development of the Napster there was a great revolution in the field of peer-to-peer. Napster was a centralized peer-to-peer network; it was not purely peer-to-peer [3]. For some legal reasons and scalability issues Napster had to be shutdown, and its application failed due to centralization as if the central server fails the whole system breaks. Then after that came the fully decentralized peer-to-peer network called Gnutella. Gnutella had no central server rather all the nodes in this network act as a client and server both [7]. A node which initiates a query is a client while the node satisfying this query acts as a server.

The main reason for nodes becoming free riders is that they only want to download from a network without bothering about sharing anything as they are not charged any cost for downloading. If every node thinks this way and free ride in the network to utilize resources shared by others then the whole system performance will degrade rapidly [5]. These nodes want their systems to be fully free from uploading load so that they can gain maximum speeds while downloading and their system speed would also be high. A large no. of free riders joins the network to download only one file and after that they leave the system.

The major problem with free riding present in the system is that it degrades the performance of the whole system as all the queries will be directed towards few contributors present. This will make the contributors heavily loaded. When these contributors reach their threshold of handling a no. of downloads then most queries from other peers will be roaming in the network. A large no. p2p messages will be present from free riders and it will increase rapidly making the whole system heavily loaded. This situation may result into the failure of the whole system. The other problem with free riding is that it creates vulnerabilities in the system like if there are only a few no. of contributors present in the system and they are responsible for all the uploads in the system, then these few nodes will act as a centralized server failing our purpose of a decentralized network.

These problems directed researchers to find some way to control these free riders [5, 8, 11, 15, 20]. The incentive based approach gives rewards to those peers who upload some file to other peers in the network [9, 12, 16, 21]. While rewards are given for uploading, downloading makes these rewards to be taken back. The main incentive used globally is point based scheme where some points are given for uploading and some points are deducted for downloading.

The above described point based scheme does not consider the main features of a contributor like upload speed of the contributor, online duration that this contributor has spent in the network, no. of simultaneous downloads from this contributor. These drawbacks resulted into the development of a new point based protocol for peer-to-peer networks. In the proposed approach for controlling free riding these features are included and it shows a great improvement in the results produced.

The proposed Extended Point Based Incentive Mechanism (EPBIM) focuses mainly on giving benefits to the users who contribute in the system. They can contribute either by uploading a file or by giving more responses or by forwarding the queries of other users in the system. But the users who upload more get more points than the users who only forward queries. The users forwarding queries will not get enough points to download something. The designed approach awards more points for uploading to the system than it cuts for downloading from the system. In this way this scheme will attract more users to give responses, to increase their uploading speed, to increase their online duration in the system and it will benefit the systems overall performance.

The organization of the paper is as follows. In section 2, we discuss the related work regarding free riding. The mechanism for controlling the behavior of free riding and its working are discussed in section 3. The simulation parameters used in evaluation of this mechanism and the results obtained are presented in section 4. Finally, in section 5, we conclude our work and give future directions of work in this field.

2. RELATED WORK

Before discussing about the incentive scheme regarding free riding in peer-to-peer network, the knowledge about basic solutions types is a must. Free riding is affecting the performance of all the peer-to-peer networks since the innovation of itself. So it is an issue that should be solved to focus mainly to improve searching a file, publishing the file and querying for a file.

As a peer-to-peer concept [17], *free riding* means exploiting peer-to-peer network resources (through searching, downloading objects, and using services) without contributing to the peer-to-peer network. This definition of free riding proves that this phenomenon is decrementing factor of performance of any peer-to-peer network. A peer that uses the services offered by the peer-to-peer network without bothering about contributing to the network at an acceptable level is called a *free rider.*

In the study of Gnutella [5, 14] it has been found that nearly 66% of Gnutella users share no files at all and nearly top 1% users share 37% of the total files while top 20% share a major 98% of the files in the network. This study also shows that out of nodes which have at least some files to share, approximately 63% never provide a query response because they do not have the files desired by the users. Regarding the responses this work shows that top 1% provides 47% of all the responses while top 25% providing a whopping 98% of all the responses . D. Hughes[14] extended the study of Adar[5] and showed more conclusive results as 85% of peers share no files, and 86% share 10 or fewer files (as opposed to Adar's original finding that 66% of peers share no files, while 73% of peers share 10 or fewer).

M. Karakaya [17] suggested that not all of free riders behave the same in the network. This study showed three different kinds of free riders with different properties. These three types of free riding can be defined as:

a) Non-contributor (The peers who do not share anything at all or share unpopular files).

b) Consumer (The peers who consume more services in the network than they contribute).

c) Droppers (The peers who drop queries of other peers and don not forward them).

The study above makes our mind to think about how to control the behavior of free riding on the network. There have been devised different detecting and controlling mechanisms for free riding. We will discuss here the controlling mechanisms only as detection mechanisms are out of the scope here. It starts with the conditional approach where the system or the network defines the conditions before having any nodes in it. These conditions can be of any type. The nodes who want to join in this network must first fulfill these conditions defined by the system after which only they are allowed to join the system.

The default directory approach [5] has a predefined condition for the users joining this network. The users in this network are required to have default upload directory which is shared with the system. All downloads by this user will go in this default upload directory. Napster [3] used this concept of a default upload directory. Following this scheme whenever a user downloads a file it is automatically shared with the system increasing a contributor for this file. This scheme has made an improvement as user will only keep good files (downloaded from the system only) in this folder. But this scheme also has a problem with it. The user can tamper this directory easily leaving the system in its previous state.

Under punishment mechanism [17], the users who download more than they upload or the users who either share some unwanted or bad files or do not share anything at all face a restriction in the system. These kinds of users face the consequences either for sharing some bad files which are not required by users in the system or for not contributing anything to the system. When any user of this kind starts a query for some file, its neighbors search for the file and then forward this query to its neighbors. While forwarding the query from its neighbor, the TTL value is normally decreased by one. But for these users the TTL is modified differently, i.e. the TTL value may be decremented by more than one. Following this way, the search area or search horizon of the free riding peer is shrunk. The decrementing TTL value can be either 2 or 4 or more than that. In this way the overhead on the network due to the messages form free riders is also reduced.

Some free riders act in different way than others, apart from not contributing to the network; they affect the working of the system by dropping the queries from their neighbors. These peers do not forward queries to save their connection bandwidth. These types of free riders are called Droppers [17]. To solve this problem, all the search requests from a neighbor identified as a dropper can be ignored at all. Dropping a query means neither searching for this query nor forwarding this query. This solution will also decrement the overhead of the messages from free riders. At the very harsh level of punishment the peers who neither contribute to the network nor behave normally like other free riders are disconnected from the network. Here the free riders directly get kicked after joining and getting caught as free riders.

In reward based systems [9, 12, 16, 21], the peers joining the network are given a fixed amount of points which they can spend to download the files from the network. There have been researches regarding how the point should be given to the nodes while uploading and how much points should be taken from the user while downloading any file. The incentive can also be defined in terms of utility function [6] which is based on several influencing factors like measuring the usefulness of every user in the system. This measurement of usefulness has many factors in it, the most influencing factor being the number of files shared by each user in the system. Along with the number of files shared utility function can also take into account the size of files shared. One of problem with some free riders was that they share some unpopular files and use the advantages of the system.

The exchange scheme [12] is more based on the technique ‘give and take’. Users in this scheme trade resources between themselves, so the overhead of an extra server to have an eye on the incentives of different users gets reduced and also less bookkeeping is required. In this scheme the requests from users who can provide symmetric service in return simultaneously are given preference. In this mechanism each node is having an incoming request queue (IRQ) where other peers in the network show their interest for a file owned by that node. The peers who are sharing more files than others in the network are more likely to get benefited by being able to participate in more exchanges with faster transfers. The basic idea is that peers give higher service priority to requests from a set of peers that can provide a simultaneous symmetric service in return.

The point based incentive scheme [16] follows the trend of cash system where the user earns money for his work and spends this money for fulfilling his utilities. The users in here gain points only for uploading a file to other users in the network. This is the only way these users will earn points to spend them for downloading something from the network. The rate at which the user earns points for uploading is fixed here and is not dependent on any factor. While the rate of decrementing the points for downloading is not static. The decrementing rate for downloading is dependent on the file size of user’s download. If the user is downloading fewer amounts then he will be benefited lesser than the user downloading a large file size. The users joining the network initially get fixed points which they can use wither to remain in the system for a long time or they can spend these points straight for the downloading a file of size that is allowed with these many points.

3. THE WORKING OF EPBIM

The proposed extended point based incentive mechanism (EPBIM) works on two major scenarios described as:

- Deduct the point value while downloading.
- Increase the point value while uploading.

In the scheme EPBIM the users are required to upload at least 1.5GB data for every 2GB of download at an upload speed of nearly 256Kbps. This scheme changes its point system according to the upload speed of the user. While for the same upload of 1.5GB data at an upload speed of greater than 512 Kbps the user will be able to download 5GB of data which is more than double of the data at 256Kbps. This thing shows that how much the upload speed of the user will affect the overall points gain in this system. The above scenario showed only one major which will attract the users to not limit their upload speeds to minimum. There are other benefiting points also for users to contribute in the system. Some users may enter the system only for the purpose of downloading a small file so they will not be banned by this mechanism. This mechanism will only help when the free riders want to download more and more without sharing anything. This situation is the major encouraging force of this mechanism as in this scenario after entering the system user will be restricted from downloading a file size greater than equating the initiation points given to him. After expending the initiation points this user will not be having any rights to download anything from the system.

Initially when a user joins the network he is initiated with 2048 points. These points are given to users only for joining the network. If we will not give any points at the time of joining they will have to earn points by uploading, only then they will be able to download. This scenario will fall in conditional approach under must sharing scheme where user can join the network only when they share a predefined amount of disk to the network. These starting points will attract the users to join the network as the users will be able to download some files without contributing anything. After a node joins the network, EPBIM comes into picture as only when a node joins the system he will be able to know the benefits of the system. From outside he will not be attracted so much as compared to the situation when the user is inside the system. When the user is in no need for any file then he will not be interested in knowing the attractive schemes of this mechanism. It is only when a user wants a file then only he will want to know the benefits.

3.1 Point Change While Downloading

As the user is provided with initiation points equal to 2048, the user can use these points to either download something and then upload the same thing and remain in the system for a long time or he can use these points only for downloading as much as he can, and lose the right of downloading anymore. It is up to the user to use the benefits of EPBIM scheme or just become a free rider. EPBIM does not force the user to share something as it is not a kind of punishment approach rather it attracts the users to share. If the user is not interested in any scheme then he can surely become a free rider, but this scheme avoids the free riding by limiting the point value after which user will not be able to free ride. EPBIM follows a scheme for changing the point value of every user while downloading which is defined as:

i. Decrement the point value by one for every MB (1/MB) of download by the user while the downloaded size of files is lesser than 1GB.

ii. Decrement the point value by one for every additional 2MB (0.5/MB) of download by the user when the downloaded size of files exceeds 1GB.

iii. Decrement the point value by one for starting a query.

iv. User cannot download after point value becomes zero.

Fig. 1 shows the above procedure of point change when the user is downloading from the system. Till the file size does not exceed 1GB the user is charged at 1/MB point but after 1GB he is charged at the rate of 0.5/MB points.

Now let a user wants to download a file of size x MB then we can deduce a function for decrement in point value for downloading (Pdd) this file of x MB as:

$$Pdd = 1 + \beta * x + \{\gamma * 1024 + \delta * (x - 1024)\}$$

Where *Pdd* is point decrement due to the downloads (it is negative quantity) while β, γ and δ are leveling constant for downloading which are defined by

$\beta =$ 1 if $x \leq 1024$MB
0 else,
$\gamma =$ 1 if $x > 1024$MB
0 else and
$\delta =$ ½ if $x > 1024$MB
0 else

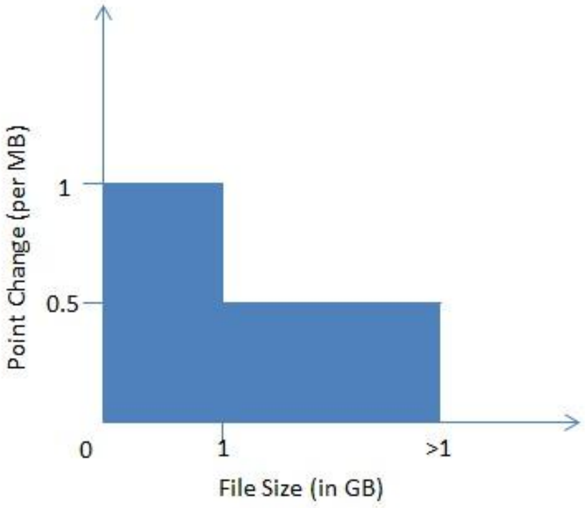

Fig. 1. Change in points while downloading.

In the equation above '1' shows this user has started the query. The leveling constant β is for a file size less than 1GB while other leveling constants are for a file size greater than 1GB. Following this scheme a user who has just joined the network will be able to download a total of 3GB of files without contributing anything to the system. If a user is getting 3GB of data for free then he will be attracted to enter the system. This situation leads more and more users to join the system and then they get to know about the attractive scheme of *'upload less download more'*. This scheme also follows a famous saying *'help others to help you'*. As this EPBIM scheme will attract the free riders in different ways to participate highly in the network, and to upload to others for their own benefit of downloading more.

3.2 Point Change While Uploading

Sharing is the major factor affecting the performance of any peer-to-peer network. The users who do not share anything at all and consume the resources of the network are called free riders. As shown in the study of Adar [4] that only top 20% of the Gnutella users were sharing a whopping 98% of the total files, rest of the 80% accounting for only 2%. This scenario provided the researchers a field to investigate an algorithm through which the free riders will themselves want to become contributors in the network. To attract the free riders this work is mainly focusing on three major scenarios where a user can earn points while uploading a file. First of all the restricting condition of zero points will fetch the interest of user to see the benefits of uploading. EPBIM uses the following scheme for change in the point value of every user when he is uploading.

3.2.1 Popularity Weight

The popularity of any file increases with increasing number of downloads. If one user downloads a new file then the peers in the neighbor of that user will get to know about that file. This will increase its popularity and these peers will also download this file. The popularity is the measurement constant to showcase if one file is popular among users or not. The file will be popular if it has got ten or more than that number of requests for downloading. This factor plays a major role in increasing the points of any uploading peer.

Popularity Weight (Qh) = 0.25 if $Qh \leq 0.1$,
0.5 if $0.1 < Qh \leq 0.5$,
0.75 if $0.5 < Qh \leq 1$,
0.5 if $Qh \geq 1$,

where Qh represents query hits in percentage of all users in a particular time period.

3.2.2 Give Response

Whenever a node gives response for a query in the system, it shows that he is contributing to the system. Hence this node should be given benefit for every query response. The responding node will earn 2 points for each query response he is giving in the system. In this way it would also be assured that the files shared by these nodes are somewhat of relevance that is why these are getting searched.

3.2.3 Upload Speed

Upload speed of the user plays an important role in EPBIM. EPBIM approach is dynamic in the way that point change is not fixed per MB of upload rather it is dependent on the upload speed of the user. The increment in the point value for uploading any file is directly proportional to the upload speed of the user uploading that file.

$$\text{Point Change} \propto \text{Upload Speed}$$

As this mechanism initiates every user's point value at 2048 so to remain inside the system and use its functionalities the users will have to upload before they use all their points for downloading and lose the right for downloading anymore. The scenario followed by EPBIM based upon which the point value is incremented for uploading to every user with upload transfer speed Tu is defined as:

i. +0.5/MB if $0 < Tu \leq 128$Kbps
ii. +1.0/MB if $128 < Tu \leq 256$Kbps
iii. +1.5/MB if $256 < Tu < 512$Kpbs
iv. +2.0/MB if $512 \leq Tu$

Where Tu is upload speed of the user. Here the user will not get benefits of the system if he is uploading at speed less than 128Kbps because he will earn one point for every 2MB of upload and he will spend one point for every 1MB of download till 1GB file size. In this situation the user will start losing his points more while downloading than he will earn while uploading. Fig. 2 shows the actual point change when the user is uploading graphically.

An example will show the actual scenario. Suppose a user downloads a file of size 768MB from the system. As the size of the file is less than 1GB hence one point will be deducted for every MB of download. Suppose just after joining the network the user download this file. So the user will be left with 1280 points after downloading the file. Now if the user wants to earn some points then he will upload this file to some other node in the network. Suppose the user restricts his upload speed to less than 128Kbps then he will earn one point for every 2MB of upload. After uploading this file of 768MB the user will get 384 points. Now after downloading and then uploading a file of 768MB the user is left with 1664 points which is less than the initiation points. Hence the user will lose points equal to half of the file size (< 1GB) after downloading and then uploading the same file at an upload speed less than 128Kbps. The situation above showed that if the user uploads a 768MB file at speed less than 128Kbps then he earn points to download only 384MB file. Contrast to this if the same user upload the same file at an upload speed of 512Kbps then he will earn enough points (1536) to download a file of size 2GB which is close to triple of uploaded file.

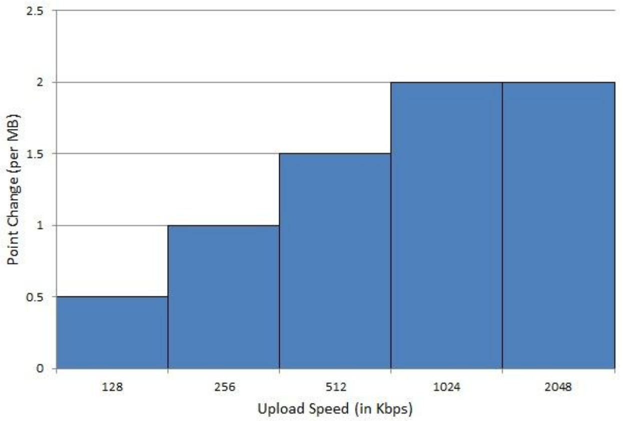

Fig. 2. Change in points while uploading

3.2.4 Online duration

The online duration in the network also plays an important role in the performance boost of the network as more users will be present in the system to fulfill any request. The more the users are in the system the more easy to get response for a query which will attract more users to join the system. For users to get benefited from the system they will have to spend at least one day in the network as point increment is directly proportional to number of days spent.

$$\text{Point Increment} \propto \text{Upload Time}$$

EPBIM follows the next explained point change scheme dependent on online duration of the user:

i. +50 points for spending one day in the network.
ii. +60 points for spending next day in the network.
iii. +40 points for spending every next day in the network.

This scheme helps the actual contributors as they spend more time in the network than others. Suppose a user spends a whole week in the network then he will earn 310 points (50+60+40*5) to download 310MB files.

3.2.5 *Simultaneous Uploads*

If the user is having more than one uploads then he is fulfilling the requests of more than one user in the network. The user must get awards for this kind of a helping nature. For uploading to every user he will earn points for uploading, but apart from this he will also earn points for number of simultaneous uploads by the same user. The increment in point value is also directly proportional to the number of simultaneous uploads.

$$\text{Point Increment} \propto \text{Simultaneous Uploads}$$

EPBIM follows these rules for incrementing points' depending on the number of simultaneous uploads:

i. +20 points for first user upload.
ii. +30 points for next one user upload.
iii. +10 points for every next user.

In a situation where a user is uploading a file to more than one user, suppose the number of users downloading from this peer is 10 then the uploading peer will earn upload points depending upon the upload speed. Apart from earning these points the user will also collect points for uploading to more than one user at a time. He will get 130 points (20+30+10*8) for simultaneous uploads.

From these major earning points we will be able to develop a function which will show overall increment for a particular user depending on all the factors. Suppose the size of uploading file is *x* MB, upload speed is *k* Kbps, the number of simultaneous uploads from the user is *n*, and the total online duration of this user is t_n then we can deduce a function for increment in point value for uploading to be defined as follows:

$$Pui = \tau *(n * 2 + \alpha * x + Psu + Put)$$

Where τ is popularity weight of the file that has been uploaded, α is upload speed constant, *Pui* is increment in the points during upload (it is a positive quantity), *Psu* is point increment by simultaneous uploads and *Put* is point increment by online duration of the user. These values can be defined as:

$\alpha = 0.5$ if $0 \leq k < 128$
1.0 if $128 \leq k < 256$
1.5 if $256 \leq k < 512$
2.0 if $512 \leq k$
$Psu = 20$ if $n \geq 1$
50 if $n=2$
$50+10*(n-2)$ if $n>2$
$Put = 50$ if $t_n=1$ day
110 if $t_n=2$ days
$110+40*(t_n-2)$ if $t_n>2$ days.

From the equations of *Pdd* and *Pui* we can have a formula for the current point value of any user as to be:

$$Pc = Pp + Pui - Pdd$$

Here *Pc* and *Pp* are user's current and previous points. Based on this current point value EPBIM decides whether the node is contributing in the network or not. If the value of Pc reaches to zero then that user cannot download anymore from the network. This mechanism attracts the users to participate in the network by sharing and uploading. This will increase the performance of the system and also the number of contributor will increase rapidly. Since free riders account for majority of traffic on the network, this mechanism will try to convert these free riders into contributors and reduce the traffic of free riders by a big margin. As the number of users in the network increases it will also increase the number of contributors and decrease the percentage of free riders in the network over the time.

4. Performance Evaluation

The performance of any peer-to-peer network is measured on the basis of how many contributors it is having and how much data each of them is sharing. We measure the performance based on different properties like the upload speed of the contributor, online duration of any peer. All the results have been generated considering the files being uploaded have popularity weight of one.

4.1 *Percentage of Free Riders*

As the number of peers in the network increases a major part of this increment is because of the free riders joining the network. In this situation these increased free riders will start more and more queries for the files in the system. This will increase the messages in the network to a very high extent. These messages from free riders can overload the whole system. The contributors present will receive a large number of queries which they may not be able to handle. If we are able to reduce the percentage of free riders to some extent then we can improve the performance of whole system.

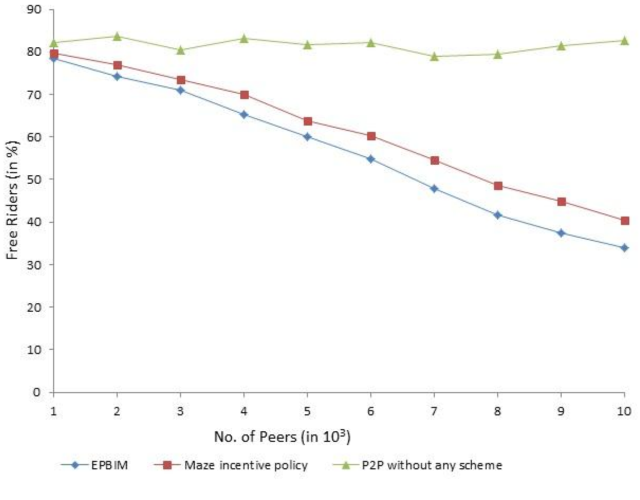

Fig. 3. Change in percentage of free riders with increasing peers.

The result in the Fig. 3 shows that by using EPBIM the percentage of free riders in the network to the overall nodes the network is less than both maze incentive policy and peer-to-peer without any policy. The result here does not guarantee that the number of free riders would also decrease with increasing number of nodes. Rather it is showing that the number of free riders will increase but increment in EPBIM will be lesser than the other two. The main reason behind this improvement is that a lot of free riders after joining the network see the benefits and then become contributors.

4.2 Download Time

The complexity of any peer-to-peer network can be defined in terms of its performance as the time it will take for a file to be downloaded by a user under this network. If the download time of any file in the system is very high then at any point of time there will be more users downloading the file. This will increase the no. of P2P messages in the system which will eventually increase the overhead.

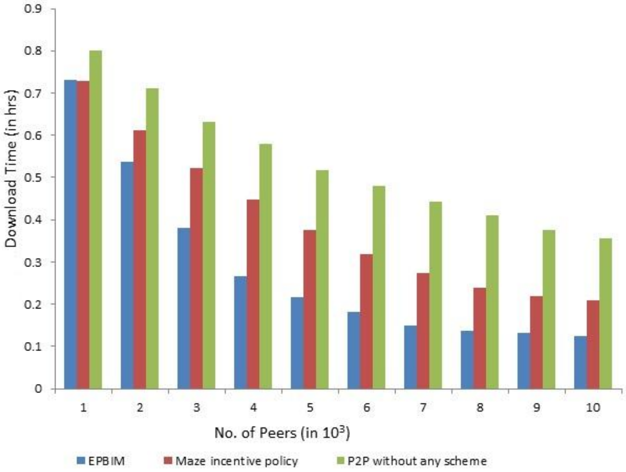

Fig. 4. Increments in download time.

The result in the Fig. 4 shows a great improvement in the download time of the system using EPBIM. At the start of the system the download time is same for all but after some time when the no. of nodes increases then the decrement in EPBIM is greater than the other two because this scheme gives high points to those users who have high upload bandwidth. As the no. of users increase the popularity of the system also increases making more users interested in joining the system as a contributor with high upload bandwidth.

4.3 Average Online Duration

A contributor serves the main purpose of query response in any peer-to-peer network. As long as the contributor is present it will help in reducing the load on the other contributors in the network. If most of the contributors join the system for a very short interval then most of the queries will be directed towards the contributors which are there for a long period, increasing the overhead on these contributors. Then this system will give a look of client-server model where some servers are processing the requests of the clients. Our aim here is to increase the average online duration of the system to increase its performance. Fig. 5 proves that EPBIM increases the average age of the system for every peer much more than maze policy and system without any scheme. The case with EPBIM is that it provides the points for high online duration in the system also which is not the case with the other two.

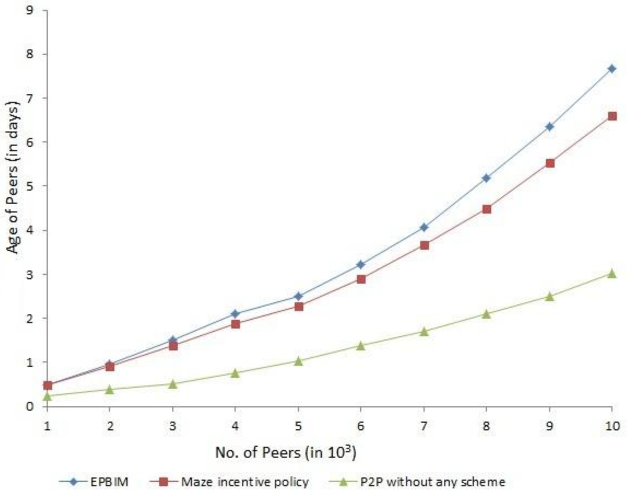

Fig. 5. Increments in online duration of peers.

4.4 Query Overhead

Average query overhead on the contributors defines the number of queries a contributor in the network gets for the data he is having. As the number of nodes in the network increases it also increases this value. Fig. 6 shows the comparison based on this. The result shows that the increment in average query overhead for EPBIM is less than the other two schemes. At the start, all the schemes have equivalent loads on the contributors. But as the number of nodes increases the increment in EPBIM is on the slower side than the other two. While the peer-to-peer without any scheme is heavily loaded after 3000 nodes, our mechanism has relatively much lower load at this point also.

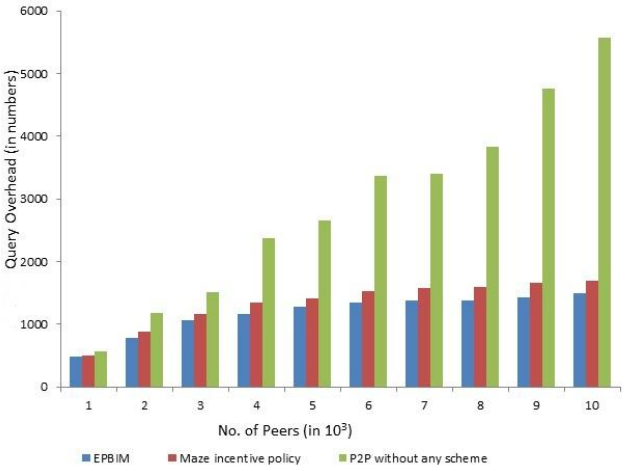

Fig. 6. Decrement in query overhead on contributors.

4.5 Download Time

The complexity of Maze policy is a function the _le size (*x*) and the current user points (*Pc*), while for EPBIM the complexity is dependent upon the popularity factor (*tp*) also. The complexity of Maze policy:

$$Tn = O(x * Pc\ )$$

The complexity of EPBIM:

$$Tn = O(tp * Pc * x)$$

So in terms of complexity the Maze policy is better than the EPBIM approach.

5. CONCLUSION AND FUTURE WORK

The problems with free riding were growing very fast. That's when the researchers found this field of study as an area for some research work. The incentive schemes have been effective in controlling the behavior of free riding. The extended point based incentive scheme proposed a solution for this problem with different approach. This work addresses the main focusing points of any contributor like his uploading speed, his online duration in the network, query responses and the simultaneous uploads.

This work first defines the constraint to be followed by the system to increase its performance. It then proposes the points to be followed by the users entering the system. Our work does not detect the free riders rather it attracts the free riders present in the network to become contributors. The results show a great improvement to its previous version of point based incentive scheme where only downloads and uploads were considered.

These results also prove that this mechanism can be used with millions of nodes and as the number of nodes in the network increases it also increases the performance of the system due to the increase in number of contributors. As the number of nodes increase, the percentage of free riders to all the nodes in the network decreases which is a big achievement.

This work proposes a solution for the free riding problem with point system. The management of the point changes, the conditions for these changes is done by a central server which is present there only for this work. The central server will be heavily loaded due to the changes from every node in the network. For every download there would changes in the point values of two nodes, one for downloading node and other for uploading node. This will make the work of this server even more complex. Because of this overhead this central server may get biased or may get crashed making the system vulnerable to the free riding attack. This thesis work can be modified to make it free from the central server problem. Also there can be some attacks on this system so some security can provided to counterattack these attackers.

REFERENCES

[1] "Peer-to-peer," http://en.wikipedia.org/wiki/Peer-to-peer

[2] Stephanos Androutsellis-Theotokis and Diomidis Spinellis, "A Survey of Peer-to-Peer Content Distribution Technologies," in ACM Computing Surveys, Vol. 36, No. 4, December 2004, pp. 335–371.

[3] "Napster home page," http://www.napster.com.

[4] "Basic Network types," http://www.pcc-services.com/network_types.html

[5] Eytan Adar and Bernardo A. Huberman, "Free Riding on Gnutella," October 2000. http://www.firstmonday.dk/issues/issue5_10/adar

[6] L.Ramaswamy and L.Liu, "Free riding: A new challenge to peer-to-peer file sharing systems," in Hawaii International Conference on System Science, 2003.

[7] "Gnutella home page," http://www.gnutella.wego.com.

[8] Ramayya Krishnan, Michael D. Smith, Zhulei Tang and Rahul Telang, "The Impact of Free Riding on Peer-to-Peer Networks," in 37th Hawaii International Conference on System Sciences, 2004.

[9] Michal Feldman, Kevin Lal, Ion Stoica and John Chuang, "Robust Incentive Techniques for Peer-to-Peer Networks," in Proceedings of EC, May 2004.

[10] Omnetpp, http://www.omnetpp.org

[11] Nazareno Andrade, Fransisco Brasileiro and Walfredo Cirne, "Discouraging Free Riding in Peer-to-Peer CPU-Sharing Grid," in Proceedings of HPDC, 2004.

[12] Kostas G. Anagnostakis and Michael B. Greenwald, "Exchange Based Incentive Mechanism for Peer-to-Peer File Sharing," in 24th International Conference on Distributed Computing Systems, 2004.

[13] "Kazaa home page," http://www.kazaa.com

[14] Daniel Hughes, Geoff Coulson and James Walkerdine, "Free Riding on Gnutella Revisited: the Bell Tolls?," in IEEE Distributed Systems Online, 2005.

[15] Maichal Feldman and John Chuang, "Overcoming Free Riding Behavior in Peer-to-Peer Systems," in ACM SiGecom Exchanges, Vol. 5, No. 4, July 2005, Pages 41-50.

[16] Mao Yang, Zheng Zhang, Xiaoming Li and Yafei Dai, "An Empirical Study of Free-Riding in the Maze P2P File-Sharing System," in Proceedings IPTPS, 2005.

[17] Murat Karakaya, Ibrahim Korpeoglu, Ozgur Ulusoy, "Counteracting free riding in Peer-to-Peer networks," in The International Journal of Computer and Telecommunications Networking, Vol. 52, Issue 3, February 2008, pp. 675-692.

[18] Tian Junfeng, Yang Lidan, Li Juan and Tian Rui, "A PEER-TO-PEER INCENTIVE SCHEME FOR OVERCOMING FREE RIDING," in Journal of Electronics , vol. 27, no. 1, pp. 60-67, 2010.