International Journal of Computer Theory and Engineering, Vol. 1, No.2,June2009
1793-8201

# Intelligent Frame work for Increasing Life Time of Peer to Peer networks

Ganesh Kumar. M and Arun Ram. K

*Abstract*— **In a peer-to-peer network each computer acts as both a server and a client—supplying and receiving files—with bandwidth and processing distributed among all members of the network. Such a decentralized network uses resources more efficiently than a traditional network and is less vulnerable to systemic failure. Peer to Peer networks are used by Bluetooth electronics and Internet based communication services, but development has largely been driven by online file sharing. The existence of high degree of malicious peers is a serious threat to the P2P network life time. This paper we propose an intelligent framework which deals with the issue of identifying free riders and other kind of peers who reduces the networks life time. Additionally, this paper also proposes a basic data warehouse structure for a peer to peer system and analyzes peer behavior using data mining functionalities. Results prove that the free riders and malicious peers get automatically eliminated from the system thereby increasing the networks life time.**

*Index Terms*- **Data mining; Free Riders; peer to peer network; Malicious peers; Reputation**

## I. Introduction

The advent of peer-to-peer (P2P) file sharing systems heralds a new era in the field of Internet technology. While these systems alleviate the scalability problem that has dogged the client-server model, they present new data management problems. It is widely believed that the success of P2P file sharing systems depends upon the quality of service offered by such systems. Accordingly most of the present research in P2P systems has been concentrated on issues such as efficient data placement, fast file lookup, data replication etc.

We argue that, in addition to the quality of service, there is another key aspect that impacts the success and continued sustenance of P2P systems. It is the quality of the data present in the system. For a file sharing system, no matter how excellent the lookup capabilities of a system are, or what file download speeds it offers, if the system does not have a large and growing number of interesting files, it will eventually fail to attract or retain users. Unfortunately, research on developing mechanisms to maintain or enhance the quality of data is yet to receive much attention from the P2P research community.

This problem is exemplified by the phenomenon of free riding in many P2P file sharing systems. A recent study on Gnutella file sharing system shows that as many as 70% of its users don't share any files at all. This means that these users use the system for free. This behavior of an individual user who uses the system resources without contributing anything to the system is the first form of the Free Riding problem. Such users are referred to as free riders. The study further indicates that not all file sharers share popular and desirable files. It shows that as many as 63% of the peers, who shared some files, never answered a single query. This implies that these file sharers did not share any desirable files. This is a second form of the Free Riding problem, wherein users share some files that are not useful.

The free riding problem affects the system in two significant ways. First, the number of files in the system becomes limited or grows very slowly. The number of popular files may become even smaller as the time goes by. This adversely affects user's interest in the system and they eventually pull out of the system. When users who share popular files pull out of the system, the system becomes poorer in terms of the amount of files shared. This is a unproductive cycle and it may eventually lead to the collapse of the system. Second, if only a few peers share popular files, all the downloading requests are directed towards those peers. This causes those peers to become hot spots, overloading their machines and causing congestion on their network. Peers frequently experiencing CPU overloads or network congestion due to the P2P system may exit the system if it affects their other routine activities.

In order to maintain the productivity and ensure the healthiness of a P2P file sharing system, there is a need for mechanisms that can help in securing cooperation from its users by encouraging them to share popular files. Surprisingly, none of the existing P2P files sharing systems, to our knowledge, offer or incorporate mechanisms that effectively encourage their users to share files of interest with other users in the system.

Data mining is used for a variety of purposes in both the private and public sectors. Industries such as banking, insurance, medicine, and retailing commonly use data mining to reduce costs, enhance research, and increase sales. For example, the insurance and banking industries use data mining applications to detect fraud and assist in risk assessment (e.g., credit scoring). Using customer data collected over several years, companies can develop models that predict whether a customer is a good credit risk, or whether an accident claim may be fraudulent and should be investigated more closely. The medical community sometimes uses data mining to help predict the effectiveness of a procedure or medicine. Pharmaceutical firms use data mining of chemical compounds and genetic material to help guide research on new treatments for diseases. Retailers can use information collected through affinity programs (e.g., shoppers' club cards, frequent flyer points, contests) to assess the effectiveness of product selection and placement decisions, coupon offers, and which products are often purchased together. Companies such as telephone service providers and music clubs can use data mining to create a "churn analysis," to assess which customers are likely to remain as subscribers and which ones are likely to switch to a competitor.

In the public sector, data mining applications were initially used as a means to detect fraud and waste, but they have grown also to be used for purposes such as measuring and improving program performance. It has been reported that data mining has helped the federal government recover millions of dollars in fraudulent Medicare payments. The Justice Department has been able to use data mining to assess crime patterns and adjust resource allotments accordingly. Similarly, the Department of Veterans Affairs has used data mining to help predict demographic changes in the constituency it serves so that it can better estimate its budgetary needs.

Another example is the Federal Aviation Administration, which uses data mining to review plane crash data to recognize common defects and recommend precautionary measures. Recently, data mining has been increasingly cited as an important tool for homeland security efforts. Some observers suggest that data mining should be used as a means to identify terrorist activities, such as money transfers and communications, and to identify and track individual terrorists themselves, such as through travel and immigration records.

In this paper, we explore a new approach that integrates Data Mining with Peer to peer system which attempts to discover and extract new knowledge from the recorded data and information. This data is normally stored in databases, and can be of different nature such as peer id and reputation of the peer. The knowledge learned is represented in forms of rules, such as classification rules, prediction rules, association rules or clusters of rules. These results can be often used for identifying the peer behavior. There are also several counters and parameters that are explained in this paper, which will be used as dimensions for the data warehouse.

## II. Related Work

[1] explains the details about various problems in peer to peer networks and how they can be solved using reputation concepts. [4] Mainly concentrates on the various issues concerning free riding and gives us the formulas to identify the free riders based on the popularity, size and number of files shared by a peer. [2] and [3] give a broad outlook of a distributed way of identifying and isolating the free riders in the peer to peer system.

## III. The Peer to peer Architecture

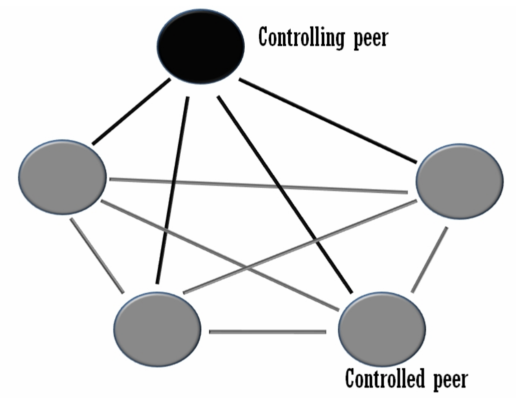


Fig. 1. Architecture of P2P network

Peer to peer architectural model considers two types of peer in the system. The peers which monitors other peers are called controlling peers and the peers who are monitored are called as controlled peers. The controlling peers are like super peers in Kazaa network, they act one layer above the controlled peers. The controlling peers take care of the isolation and elimination process within their horizon.

### *A. The Statistical Information*

The statistical information that a controlling Peer maintains about a controlled peer P consists of a set of counters. These counters are maintained and updated by the controlling peer regularly. The counters are listed below

**RQP**: The number of Query messages routed by peer P, is incremented whenever the controlling peer receives a Query message from peer P in which the TTL value is less than the fixed max TTL. The Queries originating from peer P are not counted; only the Queries originated at somewhere else and routed by peer P are counted. The controlling peer decides if the Query was originated by the controlled peer or not by looking at the TTL value. If the peer P has originated the Query, then the Query message would have a TTL value equal to the fixed max TTL.

**TQP**: the number of Query messages routed towards peer P, is incremented whenever the controlling peer sends a Query message to the controlled peer P. Both the Query messages originated at the controlling peer and the Query messages just forwarded by the controlled peer are counted.

**QHP**: the number of QueryHit messages submitted by peer P, is incremented whenever the controlling peer receives a QueryHit message from peer P. The message must be originated (not forwarded) by peer P. The controlling peer can decide this by looking at the IP address field of the message, which stores the IP address of the originator of the message.

**RQHP**: the number of QueryHit messages routed by peer P, is incremented whenever the controlling peer receives a QueryHit message from peer P in which the IP Address field in the message contains an IP address different than that of the peer P. QueryHit messages originating at peer P are not counted.

**SQHP**: the number of QueryHit messages satisfying queries of peer P, is incremented whenever a Query message formerly submitted by peer P receives a QueryHit through or from the controlling peer. To observe this, whenever the controlling peer receives a Query message whose TTL is the fixed max TTL, it records in its internal table (using the message ID of the Query message) that the Query originated from the neighbor P. Then, after receiving a Query-Hit message with the same message ID, the controlling peer decides that the QueryHit message is for that controlled neighbor and increments the counter QHSP. The controlling peer counts only once for all the QueryHit messages received for the same query.

**SSQHP**: Whenever a peer is satisfied with the service provided by the peer P, SSQHP will be incremented. To find whether the peer has been satisfied with the transaction, the peer who received the service returns a feedback packet to the controlling peer which controls the peer P or broadcasting can also be done about the satisfaction it received from the peer if any problem in the network.The values of these counters indicate both whether the neighbor is a free rider and the type of free riding. A different set of counters is maintained for each controlled peer. The details of how we employ these counters are explained in the following sections.

*A. Types of peers*

Below, we identify some possible types that a peer may exhibit. We also formulize how the identified malicious types can be detected by using the statistical information gathered about a free riding peer.

Type 1 - Non-contributor: Peer does not share anything at all or shares uninteresting files: It may be observed that a controlled peer does not return any QueryHit messages to the queries that it receives. Whenever the controlling peer initiates a search or routes a search on behalf of other peers by sending a Query message to its neighbors, the controlling peer also increases the value of the respective TQ counters (maintained in a log table) for its neighbors. The controlling peer also observes and counts the QueryHit messages received from the neighboring peers. If the controlling peer receives a QueryHit message that has the IP address of one of its neighbors in it, the controlling peer increases the value of the QH counter maintained for that peer in the log table. The controlling peer then compares the values of TQ and QH counters maintained for a neighboring peer to decide if that peer is a free rider that is not sharing any files (a non-contributor). More specifically, for this decision to be made, the controlling peer may compare the QH/TQ ratio against a threshold value and decide that the neighbor is a free rider of type non-contributor if the ratio is smaller than the threshold.

Below, we formulize the condition that is required to judge if a neighboring peer is a free rider or not. Furthermore, to remove the warm-up period and to obtain valid statistical information we propose to use a threshold value, αTQ, for the number of forwarded Query messages (TQP) to the peer that is observed. Only if that threshold is exceeded, the counter values are used to infer free riding.

***if*** *($TQ_P > \alpha TQ$) ^ ($QH_P/TQ_P$ ) < αnon contributor)*
***then***
*peer P is considered as a* ***non-contributor***
***endif***

Type 2 - Consumer: A peer consumes more resources than that it shares: A controlling peer counts the QueryHit responses (QH) originated from its neighbors and successful QueryHit messages (SQH) destined to and received by its neighbors. The comparison of these two numbers reveals if any of the neighboring peers consumes more than it shares. More specifically, a threshold value, αconsumer, can be compared against the ratio of these two numbers to decide if the neighboring peer is a free rider of type consumer or not.

***if*** *($TQ_P > \alpha TQ$) ^ ($QH_P/SQH_P < \alpha$consumer)*
***then***
*peer P is considered as a* ***consumer***
***endif***

Type 3 - Dropper: A peer drops other peers Queries: A controlling peer counts Query and QueryHit messages forwarded by each of its neighbors. If these two values (RQ and RQH) are very low for a neighboring peer, it can be assumed that the neighboring peer does not have enough connections or it drops queries and/or query hits. We call this type of free rider as a dropper. The ratio of sum of RQ and RQH counters to the value of TQ counter is compared against a threshold value, denoted with αdropper, to decide if a neighboring peer is a dropper or not.

***if*** *($TQ_P>\alpha TQ$) ^ ($RQ_P+RQH_P$)/$TQ_P$)) < αdropper)* ***then***
*peer P is considered as a* ***dropper***
***endif***

Type 4 - Malicious peer: A peer Who spreads Unauthentic files in the system: Malicious peers are the next level of free riders. Malicious peer may contribute to the peer to peer system, but their main motive is to subvert the whole system by providing unauthentic files or malicious files (Virus). A good model must not only eliminate free riders but also the malicious peer. These peers can be easily identified by the satisfaction they provide to the other peers and SSQH is used to measure the quantitative satisfaction, whenever a peer provides good files (non malicious files) then SSQH will be incremented. There is no ratio comparison for the malicious peers. They are compared with the αmalicious threshold for the identification process. Fake SSQHP can be provided by another malicious peer to eliminate good peers. For avoiding these kind of situation we weight the satisfaction by multiplying the reputation Ri value of the peer the satisfaction itself.

***if*** *(TQ$_P$ > αTQ) ^ (SSQH$_P$ *Ri < αmalicious)*
***then***
*peer P is considered as a* ***malicious***
***endif***

## IV. DATA WAREHOUSE FOR PEER TO PEER SYSTEMS

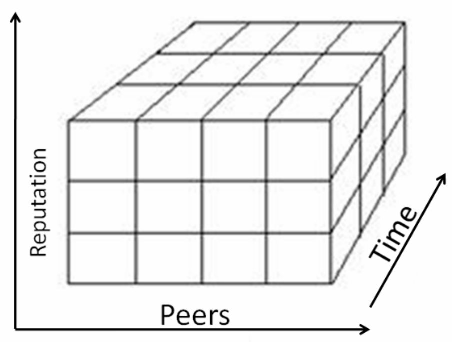


Fig.2 Data cube for a Peer to peer network

A data warehouse is a repository of information collected from multiple sources, stored under a unified schema, and which usually resides at a single site. Data warehouses are constructed via a process of data cleansing, data transformation, data integration, data loading, and periodic data refreshing. A data warehouse is usually modeled by a multidimensional database structure, where each dimension corresponds to an attribute or a set of attributes in the schema, and each cell stores the value of some aggregate measure, such as count or sales amount. The actual physical structure of a data warehouse may be a relational data store or a multidimensional data cube. It provides a multidimensional view of data and allows the pre-computation and fast accessing of summarized data. Fig.2 gives a basic description of how a data cube for a peer in a P2P network look like. The peers with whom the peer i has interacted is given in X-axis, the reputation gained is given in Y-axis, time is given in Z-axis.

### *A. Data warehousing Operations*

*1. Roll-up:* The roll-up operation (drill-up) performs aggregation on a data cube, either by climbing-up a concept hierarchy for a dimension or by dimension reduction
*2. drill-down:* Drill-down is the reverse of roll-up. It navigates from less detailed data to more detailed data. Drill-down can be realized by either stepping-down a concept hierarchy for a dimension or introducing additional dimensions.
*3. Slice and Dice:* The slice operation performs a selection on one or more dimension of the given cube, resulting in a sub cube.
*4. Pivot (rotate):* Pivot is a visualization operation which rotates the data axes in view in order to provide an alternative presentation of the data.

### *B. Super Marts*

A data mart is a subset of an organizational data store, usually oriented to a specific purpose or major data subject that may be distributed to support business needs. Data marts are analytical data stores designed to focus on specific business functions for a specific community within an organization. Data marts are often derived from subsets of data in a data warehouse, though in the bottom-up data warehouse design methodology the data warehouse is created from the union of organizational data marts. A data warehouse is a central aggregation of data (which can be distributed physically); a data mart is a data repository that may or may not derive from a data warehouse and that emphasizes ease of access and usability for a particular designed purpose. In general, a data warehouse tends to be a strategic but somewhat unfinished concept; a data mart tends to be tactical and aimed at meeting an immediate need.

The main Highlight of Peer to Peer Computing is its distributiveness. The properties of data marts and distributives can be easily cross produced to evolve a new concept of Super Marts. In Kaaza, all peers will be clustered into groups and a Super peer will be assigned to every group. The main role of the super peer is to monitor the activities taking place in that group. We propose a new concept of using data marts for those super peers so that the activities of the group can be effectively managed. All the super marts are directly controlled by a central Data Warehouse called Super Warehouse. The main job of Super Warehouse is to periodically update the information's about the peer's activities. The interaction between the super peer (with respect to KAZAA network), Super Mart and Super Warehouse the malicious behavior of the peers can be easily identified and eliminated. [7]At the same time good peers who effectively increase the life time of the system can be awarded with some incentives.

## V. DATA MINING IN P2P

The kinds of patterns that can be discovered depend upon the data mining tasks employed. By and large, there are two types of data mining tasks: descriptive data mining tasks that describe the general properties of the existing data, and predictive data mining tasks that attempt to do predictions based on inference on available data. The data mining functionalities and the variety of knowledge they discover are briefly presented in the following list.

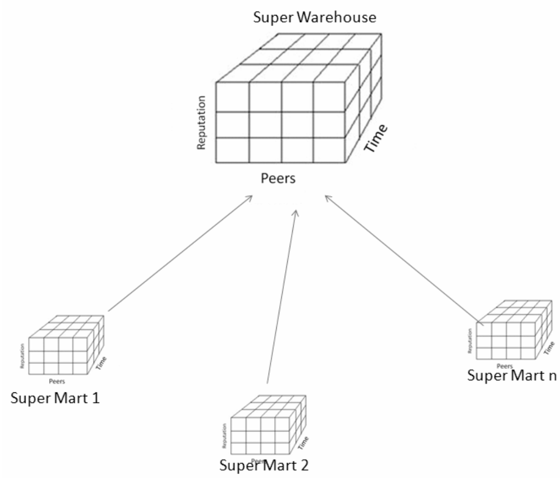


Fig. 3. Super mart and Super warehouse

### A. *Characterization*

Data characterization is a summarization of general features of objects in a target class, and produces what is called characteristic rules. The data relevant to a user-specified class are normally retrieved by a database query and run through a summarization module to extract the essence of the data at different levels of abstractions. For example, one may want to characterize the peers who play with him or the strategies that are used by a peer regularly. Free riders and malicious peer can be easily found using characterization. With concept hierarchies on the attributes describing the target class, the attribute-oriented induction method can be used, for example, to carry out data summarization. Note that with a data cube containing summarization of data, simple OLAP operations fit the purpose of data characterization.

***use*** *Gnutella_DB*

***mine characteristics as*** *"Free Riders"*

***in relevance*** *peer id, reputation, files shared*

***From*** *peerdetails*

***where*** *status in "peerstatus"*

*Describe general characteristics of Free rider in the Gnutella database*

## B. *Discrimination*

Data discrimination produces what are called discriminant rules and is basically the comparison of the general features of objects between two classes referred to as the target class and the contrasting class. For example, one may want to compare the general characteristics of the peers who have cheated more than 30 times with those whose cheated lesser than 5 times. The techniques used for data discrimination are very similar to the techniques used for data characterization with the exception that data discrimination results include comparative measures.

**Mine Comparison as** uploading_peers for goodpeers **where avg**( file.intersetingness()>80%) **Versus** freeriders **where avg**(file.intersetingness()>10) **analyze count**

*General Discrimination rules- to compare a good peer and a malicious peer*

## C. *Association* analysis

Association analysis is the discovery of what are commonly called association rules. It studies the frequency of items occurring together in transactional databases, and based on a threshold called support, identifies the frequent item sets. Another threshold, confidence, which is the conditional probability than an item appears in a transaction when another item appears, is used to pinpoint association rules. For example, a peer is interested to know what move the peer is going to take if he is going to download a file. For this the peer need to analyze the opponent peers history. With all these information the peer can easily find the next move. The discovered association rules are of the form: P -> Q [s,c], where P and Q are conjunctions of attribute value-pairs, and s (for support) is the probability that P and Q appear together in a transaction and c (for confidence) is the conditional probability that Q appears in a transaction when P is present.

## D. *Classification*

Classification analysis is the organization of data in given classes. Also known as supervised classification, the classification uses given class labels to order the objects in the data collection. Classification approaches normally use a training set where all objects are already associated with known class labels. The classification algorithm learns from the training set and builds a model. The model is used to classify new objects. For example, we might want to classify our agents into 'Good' or 'Malicious' or 'Free riding' categories with regard to their reputation. The category or 'class' into which each peer is placed is the 'outcome' of classification process.

Case based reasoning is an apt classifier for peer to peer network, which uses the previous history for the process of classification. This method is more efficient because of the possibility of unsupervised classification. To solve a current classification problem (to find a free rider), the problem is matched against the cases in the case base, and similar cases are retrieved. The retrieved cases are used to suggest a solution which is reused and tested for success. If necessary, the solution is then revised. Finally the current problem and the final solution are retained as part of a new case.

## E. *Prediction*

Prediction has attracted considerable attention given the potential implications of successful forecasting in a business context. There are two major types of predictions: one can either try to predict some unavailable data values or pending trends, or predict a class label for some data. The latter is tied to classification. Once a classification model is built based on a training set, the class label of an object can be foreseen based on the attribute values of the object and the attribute values of the classes. Prediction is however more often referred to the forecast of missing numerical values, or increase/ decrease trends in time related data. The major idea is to use a large number of past values to consider probable future values. For example if the peer is interested in predicting the chance of downloading without any interruption, he might simulate the whole transaction exactly with the history of the opponent. [5] Has effectively used the prediction technique to identify a winning strategy using data mining. We can use the same [6] Game theoretic technique to identify the malicious peer using data mining and Nash equilibrium concept.

## F. *Clustering*

Similar to classification, clustering is the organization of data in classes. However, unlike classification, in clustering, class labels are unknown and it is up to the clustering algorithm to discover acceptable classes. Clustering is also called unsupervised classification, because the classification is not dictated by given class labels. There are many clustering approaches all based on the principle of maximizing the similarity between objects in a same class (intra-class similarity) and minimizing the similarity between objects of different classes (inter-class similarity). For example if system is interested in finding the type of peer available into malicious and free rider, the system can use the history of each peer to form a cluster. Then any clustering algorithm can be used to cluster the peer types fig .4 shows an example cluster for peer type.

G. Outlier analysis

Outliers are data elements that cannot be grouped in a given class or cluster. Also known as exceptions or surprises, they are often very important to identify. While outliers can be considered noise and discarded in some applications, they can reveal important knowledge in other domains, and thus can be very significant and their analysis valuable. By analyzing the outliers peer can easily identify the outliers and isolate them from the system.

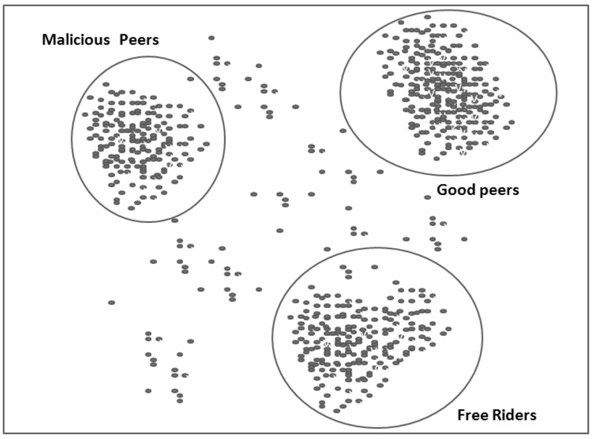


Fig.4 Peer Cluster

H. Evolution *and* deviation analysis

Evolution and deviation analysis pertain to the study of time related data that changes in time. Evolution analysis models evolutionary trends in data, which consent to characterizing, comparing, classifying or clustering of time related data. For example a peer can find the change in the strategy of each peer with respect to time and how the reputation gets changed. Deviation analysis, on the other hand, considers differences between measured values and expected values, and attempts to find the cause of the deviations from the anticipated values.

It is common that users do not have a clear idea of the kind of patterns they can discover or need to discover from the data at hand. It is therefore important to have a versatile and inclusive data mining system that allows the discovery of different kinds of knowledge and at different levels of abstraction. This also makes interactivity an important attribute of a data mining system.

*Actions against Free Riders*

If a peer identifies another peer as a free rider, it can take some counter-actions against it. We specify three levels of counter-actions. Level 1 action is the least restrictive one for the free rider, whereas level 5 action is the most restrictive one.

Level 1 Action: Decrementing TTL value more than one: To act against a suspected free rider, the controlling peer can play with the TTL value for Query messages that are received from the suspected peer, i.e. it can decrement the TTL value by more than one before forwarding. In this way, the search horizon of the free riding peer is narrowed down. This also reduces the overhead that Query messages may impose on the network. This counter-action is applied to a peer that exhibits only one type of free riding, i.e. the peer is either a non-contributor, or a dropper, or a consumer.

Level 2 Action: Ignoring requests: A free rider peer can be punished by the controlling peer by ignoring the searches (i.e. the Query messages) originating from that free riding peer. This counter-action is applied to a peer that is exactly exhibiting two types of free riding (for example, to a peer that is both a consumer and a dropper).

Level 3 Action: Reduce the horizon of file popularity: The files in the network are classified as popular, moderate and unpopular files. If a peer is found to be a free rider, then the controlling peer can reduce the horizon of popularity of the files. As a first step, the popular files can be hidden from the free rider. If the peer continues to be a free rider, the controlling peer can reduce the horizon further more to show only unpopular files.(for non contributors).

Level 4 Action: Final Warning: This is the final warning given to free riders. The warning asks the peer to upload files to certain threshold as compensation to join back the system. The threshold level is set high so that the peer’s motive towards system may change. If the peer continues to be a free rider then the level 5 Punishment can be deployed.

Level 5 Action: Disconnecting from network: If a peer is sure that a neighboring peer is a free rider or a malicious peer that is exhibiting all types of free riding, the peer may drop the connection with that peer. When disconnection is executed, the disconnected peer should reconnect to the system through a new peer if it wants to benefit from the network as a legitimate peer.

## VI. RESULT

Experiment on an existing peer to peer network was analyzed. Java environment was used to develop the peer to peer structure. Initially the experiment was done with 20 nodes, and then the nodes were scaled up to 100 nodes. Existence of behavior analyzer increased the interaction with good peer, indirectly isolating malicious and free riders in a peer to peer system. Fig 5 and 6 proves the above mentioned issue. At the starting point each peer will hold an INR initial reputation value (we took INR = 30) then with respect to the peers interaction the reputation value changes. The query given by a good peer will be given higher preference, which indirectly suppresses the free riders action. And the query replies from a good peer will be given more weight, which indirectly reduces the malicious peer’s action. So our concept of integrating data mining with peer to peer network greatly increases the lifetime of the p2p system by direct and indirect action of eliminating free riders.



IACSIT
International Association of Computer Science and Information Technology
WWW.IACSIT.ORG

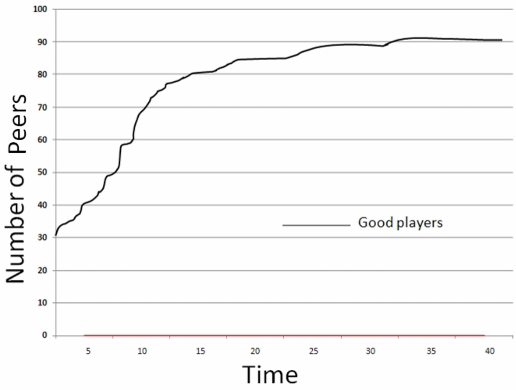


Fig.5. Interaction graph for good peer

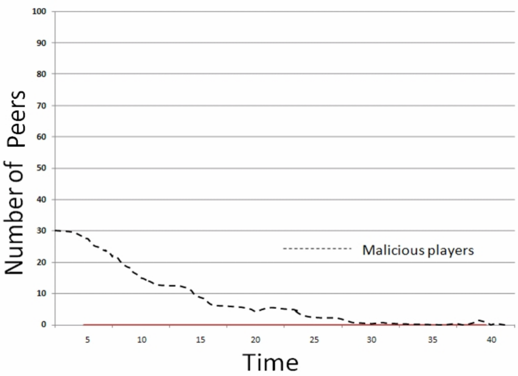


Fig.6. Interaction graph for Free rider or malicious peer

*Contribution Vs Payoff*

we consider a peer to peer system consisting of independent peers, the payoff of each peer is calculated with respect to their contribution and reputation. Now in the fig. 7 we change the policy of trust evaluation to check for the change in payoff level. We can infer from the graph that, as the policy changes from higher level to lower level the payoff level decreases, which means that the peers must have to maintain a good record so as to get a good payoff. With this result we can very easily prove that the architecture works well for any kind of policy the peers follow.

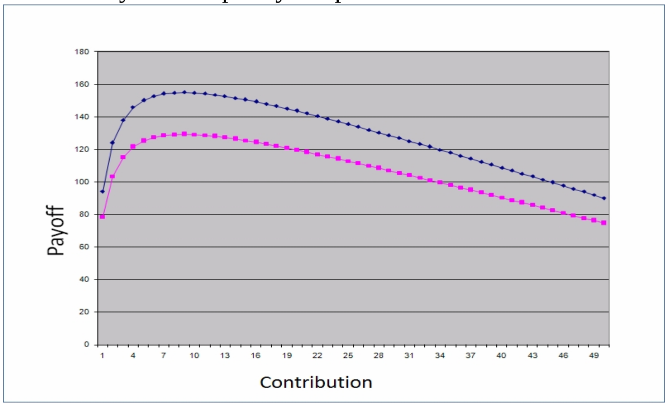


Fig.7. Payoff graph

## V CONCLUSION

While there are several ongoing research projects on improving the quality of service in P2P file sharing systems, there hasn't been much research to counter the problem of free riding effectively, which is essentially a data quality issue. To address the free riding problem in P2P systems, we have introduced this concept to measure the usefulness of every user to the system. We have proposed a free riding control scheme based on the general data mining functionalities. We expect that this paper to trigger further research in this area of P2P systems.

## ACKNOWLEDGEMENT

We would like to thank our department staffs for teaching us the latest trends in Peer to peer networks and our Head of Department for permitting us to experiment our peer to peer software in college research cell.

## REFERENCES

[1] H. Garcia Molina, S.D. Kamvar, A. Schossler “The EigenTrust Algorithm for Reputation Management in P2P networks”, Technical report, Stanford University, 2003.
[2] Karakaya, M., Korpeoglu, I., Ulusoy, O. “A Distributed and Measurement-based Framework Against FreeRiding in Peer-to-Peer Networks”. proceedings of the 4th IEEE International Conference on Peer-to-Peer computing (P2P’03). Zurich, Switzerland, September, 2003.
[3] S.D. Kamvar et al. “Incentives for Combating Free riding on P2P Networks”. Proceedings of EURO-PAR 2003
[4] Lakshmish Ramaswamy et al. “Free Riding: A New Challenge to Peer-to-Peer File Sharing Systems”. Proceedings of Hawaii International Conference on Systems Science 36 2003
[5] Yi Wang. “Integration of Data Mining With Game Theory”. International Federation for Information Processing (IFIP), Volume 207, 2006.
[6] Rohit Gupta and Arun K. Somani. “Game Theory As A Tool To Strategize As Well As Predict Nodes’ Behavior In Peer-to-Peer Networks”. Proceedings of the 11th International Conference on Parallel and Distributed Systems (ICPADS'05).
[7] S.D. Kamvar et al. ”Incentives for Combating Free riding on P2P Networks”. Euro-Par 2003 Parallel Processing.
[8] J. Han, M. Kamber, “Data Mining: Concepts and Techniques”, Harcourt India / Morgan Kauffman, 2001.
[9] Margaret H.Dunham, “Data Mining: Introductory and Advanced Topics”, Pearson Education 2004.
[10] Michal Feldman et al. “Overcoming Free-Riding Behavior in Peer-to-Peer systems”.
[11] Murat Karakaya et al. “A Distributed and Measurement-based Framework against FreeRiding in Peer-to-Peer Networks”.
[12] Ramayya Krishnan et al. “The Impact of Free-Riding on Peer to Peer Networks”.
[13] www.gnutella.com
[14] www.kazaa.com
[15] www.wikipedia.org

**GaneshKumar. M** Born on June 18, 1988, in Chennai, India. Is a under graduate student in the Department of Computer Science and Engineering Department at the Sri Venkateswara College of Engineering. His research interests are in Networking, Peer to Peer Networks, Ad Hoc Mobile Networks and Swarm intelligence. He has published many research papers and journals. He is a member if the IACSIT an International scientific association

**Arun Ram K**, Born on the 5th of July 1988, in Coimbatore, INDIA, is an Undergraduate Student in the Department of Computer Science and Engineering at Sri Venkateswara College of Engineering, Chennai, INDIA. His research interests are in Networking, User Interface Design, Peer to Peer Networks, Usability Testing and Human Computer Interaction. He is the founder of a designing and publishing group

called "eDen groups" with members from all over India and a few people from China and Singaore. He is a member of the International Association of Computer Science and Information Technology (IACSIT).