Reputation based probabilistic resource allocation for avoiding free riding and formation of common interest groups in unstructured P2P networks

Ruchir Gupta1 · Nitin Singha2 · Yatindra Nath Singh3

Received: 7 February 2014 / Accepted: 23 June 2015 / Published online: 16 July 2015
© Springer Science+Business Media New York 2015

Abstract Free riding is a major problem in peer-to-peer networks. Reputation management systems are generally employed to overcome this problem. In this paper, a new reputation based scheme called probabilistic resource allocation is proposed. This strategy probabilistically decide whether to provide the resource to requesting peer or not. Aforesaid method gives selection preference to higher reputation peers and at the same time provides some finite probability of interaction between those peers who don’t have good reputation about each other. This avoids disconnection between the aforesaid peers. The proposed scheme also introduces a new mechanism for resource distribution which not only allocates resources based on peers’ reputation but simultaneously maximizes network utility also. Algorithm for formation of interest groups based upon both similarity of interests and reputation between peers is also presented.

Keywords Freeriding · File sharing · Common interest group · Reputation based resource allocation

1 Introduction

Peer-to-peer systems are used to share resources like bandwidth, storage capacity and CPU cycles directly with its members. In this paper, we analyze bandwidth as the shared resource because it is the most commonly demanded resource in peer-to-peer networks. We are assuming that nodes share their downloaded files with other peers (e.g. in Bittorent), thus contributing back to network by sharing the uplink bandwidth. The benefit derived by a peer or node from the network will be more if greater number of peers exchange there resources. The term node and peer means same and are used interchangeably. Most of peers have to purchase their bandwidth for upstream as well as downstream link. As a peer has to pay for bandwidth, it will try to minimize its uploads to save in terms of upstream capacity. Therefore a rational peer will try to become free rider which downloads from network without sharing any thing back to it. This will overload other peers which are sharing resources. Consequently network performance will deteriorate because lower the amount of bandwidth a requesting peer gets, poorer will be the quality of service received by it. If every peer in system becomes free rider then network will have nothing to share and it will subsequently crash. Thus free riding is a major problem in the peer-to-peer networks [2, 11]. Different Reputation management systems have been proposed in past to overcome this problem [10, 21, 26]. The earlier work [16, 21] has assumed that nodes have single capacity link. A single capacity link implies that sum of download and uplink capacity is always constant.
Consequently this peer might be forced to leave the network. Without getting any chance to improve its reputation. Low reputation peers get completely disconnected from the network. 

Aforementioned measures lead to a situation where very low reputation peers get completely disconnected from the network. Prevailing reputation schemes distribute resources on the basis of node’s reputation. This forces the peers to contribute its resources to the network. To prevent downloading malicious material, nodes usually request resources from high reputation peers. Aforementioned measures lead to a situation where very low reputation peers get completely disconnected from the network without getting any chance to improve its reputation. Consequently this peer might be forced to leave the network.

More recently studies [1, 13] have confirmed that query efficiency of the network can be enhanced by grouping the nodes on the basis of similarity of their interests. Peers now quickly receive data at minimum cost because nodes, instead of sending query to any random set of peers will now send request to its group members who have a higher probability of possessing the required file as they share similar interests. Sometimes only interest based categorization and overlooking members reputation may lead to a situation where few high capacity peers may be grouped with large number of lower capacity peers. These high capacity peers will rarely get same quality of service which they offer to the network.

In this paper, we modify existing reputation system of [10], propose a new probabilistic resource allocation scheme and introduce an algorithm for common interest group formation to overcome all the aforementioned deficiencies discussed earlier in this paper. The important contributions of this paper can be summed up as follows:

1. To safeguard interests of peers demanding within the limits of their contribution, a reputation earned by a node is made proportional to ratio of resources requested by a node to the resources it has shared. Therefore now it is possible for low capacity peer to interact with high capacity peer when it demands as well as contributes less.
2. We have proposed a probabilistic resource allocation method in this paper. It first uses probabilistic selection scheme to decide whether to give service to requesting peers or not. The probability of requesting peer getting selected for service increases as its reputation increases. The low reputation peer have small finite probability of getting selected but they don’t get completely disconnected from high reputation peers. The aforementioned scheme then distributes resources among selected nodes in such a way that it maximizes utility of the competing nodes. The utility in this context represents the degree of satisfaction of requesting nodes [22]. Therefore overall allocation policy distributes shared capacity in both fair and efficient manner by considering both reputation and utility derived from received resources during the distribution process.

3. A new algorithm is introduced which constructs the neighbourhood by considering both interest and reputation of the members. Reputation of a node gives an idea about the quality of service offered by the node and it will discourage peers with large difference in reputation to be placed in same group.
4. Finally proposed model is evaluated through simulation and results substantiate our claim that the above mentioned proposals, considerably reduce the free riding in the peer-to-peer networks. An unstructured P2P system is considered for evaluation. Data rate signifies quality of service, higher the download data rate, better is the quality of the service received and vice versa. Therefore data rate available for the download is used as the metric for analyzing system performance.

Remainder of the paper is organized as follows. Section 2 gives a brief insight into the related work and Section 3 gives an overview about system model under consideration. In Section 4 an algorithm for optimizing the node’s shared capacity is discussed. Section 5 gives a method to compute the reputation and it further discusses probabilistic resource allocation based on reputation and utility derived by node. Thereafter server selection according to interests of node and reputation is discussed in Section 6. Numerical results are given to verify the hypotheses in Section 7. Finally Section 8, concludes the paper.

2 Related work

Various papers have suggested array of reputation systems with variety of techniques for preventing free riding in peer-to-peer network. Feldman et al. [8] estimated generosity of node as the ratio of the service provided by the node to the service received by the node. Nodes will be served as per their estimated generosity. Feldman et al. provided only the mathematical analysis for addressing the problem of free riding in peer-to-peer network. In the same context, Kung et al. [12] proposed selection of a peer for allocation of resource according to its contribution to the network and usage of resources. Nodes desirous to receive resources have
to contribute above a certain level to the network. Gupta et al. gave a reputation model [10] which considers uncertainties in the inputs to arrive at more accurate trust estimate for countering the free riding. Banerjee et al. [4] proposed that a node will calculate the expected utility function for the requesting node. On the basis of earlier calculated utility function, a peer will decide whether to provide or not to provide the service. Solutions proposed in [4, 8, 10, 12] do not discuss the mechanism for resource distribution in peer-to-peer network. Satsiou et al. [21] and Meo et al. [16] proposed resource allocation mechanism for single link capacity systems only. Most authors [4, 8, 10, 16, 21, 21] do not take into account the amount of resources offered in a given transaction. Rationally, the gain in the reputation for a node sharing more, should be greater than that for the node sharing less, because of the extra cost incurred in contributing more resources. Meo et al. [16] considered network as a market and proposed that the second price auction leads to optimality. The model proposed in [16] is limited to networks where at a time a node can perform only one upload and one download. Satsiou et al. [21] proposed a superior model for single link channel where multiple uploads and downloads are allowed. The resources are allocated based upon peers' reputation. Although the resources are efficiently utilized but degree of satisfaction received by competing nodes is completely ignored during allocation. Ma et al. [15] proposed progressive water filling algorithm based upon marginal utility for resource allocation among different requesting nodes. The base of bucket for water filling is determined by contribution of requesting node. Ma et al. [14] had also proposed to allocate the resource to requesting node on the basis of their contribution and requirement of bandwidth. Yan et al. [25] proposed a ranking based resource allocation scheme. Resource allocation is done according to utility and ranking of requesting peer to ensure max-min fairness.

All the above mentioned mentioned resource allocation schemes [14–16, 21, 25] may sometime leads to disconnection of lower reputation peers. New entrant and the non-cooperative peers in the P2P network comprise low reputation peers. Samuel et al. [20] put forward a system which allocates credit points based upon the peers' contribution to the network. The peer can download data by exchanging it with the credit points. They proposed a grace free method where new comers are provided some initial credit points to enable them to download file from the network and prevent their starvation. However lower reputation peers can misuse this method by leaving the network and re-joining again with new identity to gain extra credit points provided to the newcomers. Therefore grace free method encourages whitewashing [6] in the network. Samuel et al. also proposed auction method where low reputation peers can purchase credit points in exchange for money but this increases overall complexity of the system as it will require introduction of virtual banks in the P2P network. Shah et al. [5] put forward an optimal choking strategy where some share of the upload capacity of every peer is made available for the randomly selected peers independent of their reputation. However this optimal choking strategy can only be employed in system employing tit for tat strategy [19] for preventing free riding. In its current form, optimal choking strategy cannot be applied in the reputation based systems.

Social networks are formed on the basis of interests of users. This fact is being capitalised to improve query search as well as recommendation network in the peer-to-peer networks [7, 13, 23, 24]. Liu et al. studied BitTorrent traces [13] and concluded that interest based grouping of peers results in an efficient query system. Liu et al. also proposed a DHT based system for implementing this kind of group formation. In another system Gnutella [1] the connected nodes of network are called servant. Each servant maintains a list of nodes, which have recently replied to the query in the cache but it is limited to that only. Wang et al. [23] proposed interest based online social communities that are headed by the super nodes and the nodes join the communities according to their interests. These communities will have a trust relationship among its members. Farahbaksh et al. [7] established a friend network on the basis of similarity of interests. The interest groups existing in peer-to-peer networks till now are based on similarity of interest.

In order to ensure that the lower reputation nodes are not disconnected altogether, in this paper, probabilistic resource allocation is proposed which gives finite probability of interaction for low reputation peers and hence provides them a chance to interact and improve their reputation. It allocates resources efficiently while maximizing the utility derived by the receiving nodes. We have also considered reputation alongside similarity of interest for group formation to improve the quality of service received by the members of the group.

3 System model

We consider a P2P system of $N$ peers. In the system a peer acts as both server and client. In this paper, unstructured P2P system is studied. In this, the index entries (keyword and resource URI pairs) stored with a node are not related to node's address. Peers share bandwidth among each other for the content sharing. Higher the bandwidth a requesting peer receives, greater is the rate at which it can download data from the network and vice versa. A peer in the network is connected to other peers through an access link followed by the backbone network and then again by the access link. We also assume that the network is heavily loaded i.e. every peer has plenty of outstanding requests and
peers are always competing for the available upload capacity. The download and upload capacity in the system under consideration are independent of each other unlike the single capacity link proposed in [16, 21]. Although upload capacity is independent of the download capacity but as peer has to pay for the upload bandwidth therefore it will always try to minimize it, leading to the free riding in the network. It is assumed that download data is more valuable than the cost of download access link therefore a peer will never cut back its download capacity. Therefore a reputation management system is required to discourage peers from free riding. A download capacity sharing algorithm is also required which directs peer to share certain amount of bandwidth so that they can get a required quality of service from the P2P network.

4 Capacity sharing

As every node in the network is rational, therefore it will try to increase its pay-off by sharing minimum amount of resources until it starts impacting the amount of received resources. If there is a reputation management system implemented, the nodes are compelled to share a certain minimum resources for getting a desired level of service quality from the network. A node has a finite amount of bandwidth that it can share. It will share only some part of the bandwidth such that it gets required service quality. In this way, a rational node optimizes the maximize the gain of itself while minimizing the cost incurred in the upload bandwidth.

We propose an algorithm which helps nodes to share minimum amount of their upload capacity such that it gets the required quality of service from the network. The optimal operating point is dynamic as the download requirement of the node changes with time. Therefore the upload capacity of node needs to be adjusted regularly even if it has reached the optimal operating point in the previous round. The Algorithm 1 automatically keeps on adjusting the upload capacity of node so that it is always close to the optimal point. The Algorithm 1 is different from the algorithm given by Satsiou et al. In [21], the optimality corresponds to best possible partitioning of the upload and download capacity so as to maximize the utility rather than finding minimum upload capacity for getting requisite quality of the service. The optimality in [21] is defined in this manner because in the model proposed by Satsiou et al., sum of the upload and download capacities fixed i.e if we increase the upload capacity then the download capacity decreases and vice versa. While, in the current work, the upload capacity shared (Algorithm 1) is independent of the download capacity. Initially a node will share the capacity as per its perceived download requirement. By perceived download requirement, we mean a rough estimate of its download requirement such that it should neither be too low to ruin the reputation if it is a newcomer node nor be too high to cause a cost penalty. If no estimate is available with the node, half of total download capacity will be shared. Thereafter amount of shared capacity will be periodically reviewed and adjusted for getting requisite service quality. Node will tweak the value of its shared capacity by periodically increasing and decreasing it by some amount $\delta$ to get the optimal point where it will get required service at minimum cost. The above proposed strategy is described as follows.

Algorithm 1 Shared upload capacity adjustment of a node

\[
k = 0; \text{ and } A(k) = -1 \text{ (} k \text{ is the instant when node reviews its sharing capacity and } A(k) \text{ can take value +1, 0 or -1 indicating increase, no change or decrease respectively in shared bandwidth at time instant } k) \\
\text{repeat} \\
\quad D_k \leftarrow \text{average data download for } kT \text{ to } (k + 1)T \\
\quad D' \leftarrow \text{desired data download rate from network} \\
\quad U_x = U_x + \delta \cdot A(k) \{U_x \text{ is shared capacity of the node}\} \\
\quad \text{if } |D_k - D'| \leq \epsilon \text{ then} \\
\quad \quad A(k + 1) = 0 \\
\quad \text{else} \\
\quad \quad \text{if } D_k < D' \text{ then} \\
\quad \quad \quad A(k + 1) = 1 \\
\quad \quad \text{else if } D_k > D' \text{ then} \\
\quad \quad \quad A(k + 1) = -1 \\
\quad \text{end if} \\
\quad \text{end if} \\
\quad k \leftarrow \text{mod} \gamma (k + 1) \\
\text{until Node is in the network}
\]
5 Reputation based system

Network is only meaningful if nodes are interacting with each other and contributing to each others’ need. As mentioned earlier, in this paper we consider a network consisting of member nodes which are rational in nature i.e. they are only looking to maximize their own interest without caring about others. Rational nodes are selfish and contribute to the network only when they have some incentive for doing so, otherwise they become free riders. This leads to overloading of other nodes which are contributing to the network and in worse case if every body becomes free rider then the network will collapse after some time. Therefore free riding poses a grave threat to the overall system performance and needs a reputation management system. For avoiding free riding, a reputation based incentive system can be employed where every node keeps the record of cooperative behaviour of every other member node. Based upon the observed behaviour of a requesting node, serving nodes decide the quality of service which will be given to the requesting node. This kind of system forces rational nodes to contribute to the network so as to get desired level of service quality from the network. To implement above discussed reputation system, we need to formulate a way to estimate the reputation and thereafter allocating resources based upon the estimate.

5.1 Reputation management system

Ideally, reputation should be measure of cooperative behaviour of a node. However cooperative behaviour being an abstract quantity private to a node, its measurement is very difficult. It is possible to measure only implications of the cooperative behaviour and that too with certain degree of uncertainty. Consequently cooperative behaviour can only be estimated within certain accuracy level on the basis of observed behaviour.

There could be number of ways to observe the behaviour of a node. One method may be to use the ratio of received data rate to the requested data rate. The advantage of such a technique is that if some node is asking for less amount of data, the serving node will not earn a bad reputation which might be the case if only the received data rate is used as criterion. Moreover reputation estimation with this method remains between 0 and 1, as received data rate is always less than the data rate requested. The newcomers to the system should be assigned some initial reputation as in [21] so that they are eligible to receive services from the network. Later on based upon their cooperative behaviour, their reputation can increase or decrease. A basic trust metric [10] can be defined as

\[ t_{ij} = \frac{\hat{q}_{w,ji}}{q_{r,ij}} \times \left( \frac{q_{a,ji}}{\min(q_{i,ay}, q_{f,ij})} \right)^{1-\eta_i}. \] (1)

For easy reading, all the notations related to the reputation calculation are listed in Table 1.

| Notation | Description |
|----------|-------------|
| \( t_{ij} \) | Reputation of node \( i \) measured by node \( j \) |
| \( \hat{q}_{w,ji} \) | Data rate which node \( j \) is willing to provide to node \( i \) |
| \( q_{o,ji} \) | Data rate which node \( i \) is offering to node \( j \) |
| \( q_{r,ij} \) | Data rate requested by node \( i \) from node \( j \) |
| \( q_{a,ji} \) | Actual data rate received by node \( i \) from node \( j \) |
| \( q_{i,ay} \) | Data rate accepted by node \( i \) against whatever is offered by node \( j \) |
| \( q_{f,ij} \) | Feasible data rate through underlying TCP/UDP channel between the nodes \( i \) and \( j \) |
| \( \eta_i \) | Amount of penalty in case of cheating by serving node |
| \( q_{r,i,d} \) | Download capacity of node \( i \) |

Notation for the Reputation Metrics

This whole process is implemented by Algorithm 1. The user can set the desired download rate on its requirement, but higher the download rate desired more will be the bandwidth that is needed to be shared. The value of \( \epsilon \) used in Algorithm 1 is kept constant for this paper. The value of \( \epsilon \) can also be decided by variation in the demand for resources inside the network. If node observes high variation in the demand, value of \( \epsilon \) can be increased and vice versa. This will conserve energy by minimizing switching across optimal point when the demand variation is very high. In the Algorithm 1 the \( \Gamma \) is used for implementing wrap around so that value of \( k \) does not overflow the memory assigned to it.

\[ q_{f,ij} = \min(q_{i,ay}, q_{r,ij}) + \epsilon \]

This whole process is implemented by Algorithm 1. The user can set the desired download rate on its requirement, but higher the download rate desired more will be the bandwidth that is needed to be shared. The value of \( \epsilon \) used in Algorithm 1 is kept constant for this paper. The value of \( \epsilon \) can also be decided by variation in the demand for resources inside the network. If node observes high variation in the demand, value of \( \epsilon \) can be increased and vice versa. This will conserve energy by minimizing switching across optimal point when the demand variation is very high. In the Algorithm 1 the \( \Gamma \) is used for implementing wrap around so that value of \( k \) does not overflow the memory assigned to it.

\[ q_{f,ij} = \min(q_{i,ay}, q_{r,ij}) + \epsilon \]
The disadvantage of this kind of system is that it does not take into account the amount of resources requested by a peer during the calculation of trust value. It means that if a node is asking for less amount of resources from a node and more amount of resources from another node then both the serving nodes will get similar gain in reputation if both are fulfilling the requests of the node by same fraction. However, the node that was requested more resources, had to pay more in comparison to the other one in terms of upload bandwidth. Therefore logically reputation gained by this node should be more than the node which has served the lesser amount of resources.

This problem can be taken care of by assigning weights to the different transactions in proportion to the absolute amount of resources requested. Weights should range between 0 to 1 and larger service request should get a higher weight. In this paper, a transaction which is equal to download capacity of requesting node is given weight 1 because this is the maximum amount of resource, a node can ask in one single transaction.

After weight assignment node $i$ will calculate the reputation of node $j$ as,

$$ t_{ij} = \left( \frac{q_{u,ji}}{\min(q_{ij,ay}, q_{f,ij})} \right)^{1-\eta_j} \times \frac{q_{w,ji}}{q_{r,ji}} \times \frac{q_{r,ij}}{q_{r,i,d}} , \quad (2) $$

where $q_{r,i,d}$ is the download capacity of the node $i$.

The modified reputation model will work efficiently for homogeneous networks only. In homogeneous networks all the nodes have same upload and download capacity. For heterogeneous network, a high capacity node $B$ will always assign a low reputation to low capacity node $A$ irrespective of the fact that $A$ requests resources in correspondence with what it had earlier shared. Consequently chances of $A$ getting resources from $B$ will be very small. Therefore the proposed reputation model is unfair to low capacity nodes. To make system more fairer, resources are distribute on the basis of effective reputation of node which is discussed in the next section.

### 5.2 Probabilistic resource allocation

In the probabilistic resource allocation, a serving node first carries out probabilistic selection. Probabilistic selection is the process of selecting nodes from the set of requesting nodes on the basis of their effective reputation. The effective reputation is calculated later on in the Section 5.2.1. Afterwards the serving node distributes resources among the selected nodes such that it maximizes overall network utility. Altogether in the whole process of probabilistic resource allocation, the resources get distributed in such a way that the high reputation nodes are given preference in the allocation and at the same time, maximum network utility is achieved.

#### 5.2.1 Probabilistic selection

Suppose node $i$ had earlier requested resources from the node $j$, and based upon this transaction it had assigned reputation $t_{ij,\text{effective}}$ to the node $j$. This $t_{ij,\text{effective}}$ will be used by $i$, during probabilistic selection when node $j$ is requesting resources from the node $i$. In order to decide whether to select the requesting node $j$ for allocation, the serving node $i$ first checks its reputation table and converts $j$’s reputation into the effective reputation. Effective reputation is the reputation adjusted according to the amount of resource requested. Serving node calculates effective reputation by multiplying the reputation value of the requesting node with the ratio of its download capacity to the amount of resource requested, i.e.

$$ t_{ij,\text{effective}} = t_{ij} \times \frac{q_{r,ij}}{q_{r,i,d}}. \quad (3) $$

Note that as peers are rational therefore their download capacities are directly proportional to their upload capacities. Hence peer with high upload capacity will have high download capacity and vice versa. From now on for sake of brevity, peers with high upload and download capacities will be called high capacity peers whereas peers with low upload and download capacities will be called low capacity peers. If the node $j$ is requesting for the smaller amount of resource than the maximum download capacity of the serving node, then from Eq. 3 it is clear that $j$’s effective reputation will increase whereas if the opposite happens, $j$’s effective reputation decreases. Therefore the low capacity nodes can interact with the higher capacity nodes if they demand less from high capacity nodes. Same capacity nodes will always have a higher value of reputation between them. Also $\left( \frac{q_{r,id}}{q_{r,ij}} \right)$ will be close to 1 for them, therefore their effective reputation will still have a high value. Consequently the chances of nodes getting selected for resource allocation in case of the similar capacity nodes will remain unaffected by the proposed modification in reputation, when they demand in proportion to what they share.

Once node $i$ gets the effective reputation of the node $j$, it selects the node $j$ with probability proportional to its reputation. Node $i$ generates a random number $\text{rand}$ and if the number generated is smaller than $P_{\text{allocation},i,j}$, requesting node will be selected for resource allocation. $P_{\text{allocation},i,j}$ signifies chances of the node $j$ for being eligible for resource allocation. Higher the value of $P_{\text{allocation},i,j}$ higher
are the chances of a node getting selected for the resource allocation and vice versa. \( P_{allocation,i,j} \) is calculated as

\[
P_{allocation,i,j} = \begin{cases} \hat{P}_{allo} & \text{if } \hat{P}_{allo} < 1. \\ 1, & \text{otherwise} \end{cases}
\]

(4)

where \( \hat{P}_{allo} \) is given by

\[
\hat{P}_{allo} = (t_{ij,\text{effective}})^x \cdot v_i.
\]

(5)

Here value of \( x \) provides flexibility to the serving node in terms of the degree of control a reputation should have over the selection process. Analysis in [9] demonstrate that, cooperative node will be able to win over the free rider in reputation game if the value of \( x \) lie between 0 and 1. The optimal value of \( x \) for preventing the free riding in the p2p network was calculated by Gupta et al. [9] as 0.75.

In Eq. 5, \( v_i \) physically represents an adjustment parameter which modifies the value of \( \hat{P}_{allo} \) in such a way that the total bandwidth shared by the serving node \( i \) remains nearly equal to the bandwidth demanded by the nodes selected by \( i \) for the allocation. In this way the bandwidth requirements of all the requesting nodes are almost completely satisfied.

Serving node \( i \) will dynamically and periodically adjust the value of \( v_i \) to achieve the optimal utilization of the shared capacity. When node \( i \) joins the network \( v_i \) is initialized to 1 under assumption that the shared capacity is equal to the bandwidth demanded by the requesting nodes and then it is periodically adjusted on the basis of the load on the serving node. If the total shared capacity is more than the total demand of the selected nodes i.e some part of the shared capacity is remaining unused than the serving node will increase the value of \( v_i \) so that \( P_{allo} \) of every requesting node \( j \) increases. In this way, the chances of selection of every requesting node, for getting the service increase. Consequently the number of nodes getting selected using probabilistic selection increases. In this way newly selected nodes can use the untapped part of the shared capacity. If the total shared capacity is less than the demand by the selected nodes, it implies that the serving node is getting overloaded with the requests, therefore the value of \( v_i \) needs to be decreased which further decreases \( P_{allo} \) so that the load on the serving node can be reduced. The increase and decrease in the shared capacity can be performed in the steps of \( \theta \) where the value of \( \theta \) is decided by the serving node.

A node will be selected for resource allocation if,

\[
rand \leq P_{allocation,i,j}
\]

(6)

where \( rand \) is the random number generated by the node. The \( P_{allocation,i,j} \) given by Eqs. 4 and 5 is directly proportional to \( t_{ij,\text{effective}} \) hence nodes with higher reputation have greater chance of getting selected. This motivates peers in the network to contribute more so that they have higher chance of getting service from other nodes. Also sometimes random number generated may be small enough such that \( P_{allocation,i,j} \) corresponding to a low reputation nodes is greater, therefore low reputation nodes will have some finite probability of getting services from the network. The probabilistic selection neither gives any extra reputation to new comers nor does it requires virtual banks for its implementation. Therefore it doesn’t promotes whitewashing and is very simple to implement than the already existing methods to prevent starvation of the new comers and the low reputation peers For sake of brevity, from know on the nodes who finally get selected for the resource allocation, after probabilistic selection will be refereed as requesting node in the subsequent Section 5.2.2.

5.2.2 Resource allocation

After the selection of nodes, if the total amount of resources demanded by the requesting nodes from the serving node is less than or equal to the shared capacity of the serving node, then every requesting node is allocated resources as per their requirement. However if the total demand for the resources is greater than the shared capacity of the serving node then the serving node employs an allocation strategy for the distribution of the resources among the requesting nodes. The serving node distributes resources such that in the future, it can get better services from the current requesting nodes. As the serving node is rational therefore it will allocate the resources in such a way that its chance of getting selected for the resource allocation in future are maximized. From Eqs. 4 and 5 it is clear that higher the effective reputation of a node, greater are its chances of getting selected for the resource allocation. However based upon the future requirement a serving node can request for a resource from any random node in the network. Consequently providing resources to only one particular node or a set of the nodes may not be beneficial. Hence serving node \( i \) will distribute resources in such a manner that its overall gain in effective reputation i.e. \( \sum_j (\hat{t}_{ji,\text{effective}})^x \) gets maximized. Where \( j \) in summation signifies set of nodes which will be allocated resources by the serving node in the current round. To solve the problem of maximizing \( \sum_j (\hat{t}_{ji,\text{effective}})^x \), the serving node \( i \) will try to estimate the effective reputation \( \hat{t}_{ji,\text{effective}} \) obtained, if it allocate resources to a particular requesting node \( j \). This process is repeated for all the requesting nodes. The estimation of the effective reputation is performed before the resource allocation to any of the requesting nodes.

To simplify the estimation of \( \hat{t}_{ji,\text{effective}} \), the current serving node \( i \) assumes that the requesting node \( j \) demands
resources equal to their actual download requirement. This assumption will hold good for most of the time because a node demanding more than its actual requirement will be at loss during allocation. This is discussed in detail later in this section. The serving node also ignores underlying network congestion for estimation purpose. This is a valid assumption because earlier studies have shown that apparently the serving node’s capacity is most likely the bottleneck during the communication and not the underlying network [3]. Therefore based on these assumptions, \( q_{a,ij} = \min(q_{ji,ay}, q_f) \). Thus Eq. 3 reduces to

\[
\hat{r}_{ji,\text{effective}} = \frac{\hat{q}_{w,ij}}{q_{r,ij}}.
\] (7)

The value of \( \nu_j \) is continuously adjusted in Eq. 5 so that the serving node does not get overloaded by the requests. Therefore for most of time serving node’s willing rate \( \hat{q}_{w,ij} \) will be equal to its offered rate \( q_{o,ij} \). So \( \hat{q}_{w,ij} \) is replaced by \( q_{o,ij} \) in the Eq. 7 and we get

\[
\hat{r}_{ji,\text{effective}} = \frac{q_{o,ij}}{q_{r,ij}}.
\] (8)

\( \hat{r}_{ji,\text{effective}} \) is calculated by the serving node \( i \) when the node \( j \) requests for the resource. We can observe that the estimated reputation is directly proportional to rate of the service offered \( (q_{o,ij}) \) by the node \( i \). \( q_{r,ij} \) is the data rate demanded by the node \( i \) from the node \( j \) in the future transaction. Therefore there is no way of determining the actual value of \( q_{r,ij} \), its value can only be predicted statistically. During steady state operation of the network over a period of time if a node \( A \) is asking lesser amount of resource from the node \( B \) then later on during the future transaction, node \( B \) can demand only a lesser amount of resource from the node \( A \). If \( B \) demands for the bigger amount of resource, then its effective reputation will come down and its chances of getting selected for the service by \( A \) diminishes. Hence the amount of resource, node \( A \) demands in the future transactions from node \( B \) is approximately equal to the amount of resource \( B \) had earlier demanded from \( A \). Therefore statistically \( q_{r,ij} \) can be approximated to \( q_{r,ji} \). Hence \( \hat{r}_{ji,\text{effective}} \) estimated by node \( i \) is

\[
\hat{r}_{ji,\text{effective}} = \frac{q_{o,ij}}{q_{r,ji}}.
\] (9)

After serving node has estimated \( \hat{r}_{ji,\text{effective}} \) for all the requesting \( j \)’s than \( i \) will distribute its shared capacity in such a way that its utility or the net gain in its reputation from the network gets maximized. To simplifying the optimization problem of resource allocation, the value of \( v_j \) is taken as 1. Therefore the optimization problem, a node needs to solve, becomes

\[
\max \sum_j (\hat{r}_{ji,\text{effective}})^x \iff \max \sum_j \left( \frac{q_{o,ij}}{q_{r,ji}} \right)^x.
\] (10)

Such that

\[
\sum_j q_{o,ij} = U_{s,i} \;
\]

\[
q_{o,ij} \leq q_{r,ji} \quad \text{and} \quad \sum_j q_{r,ji} > U_{s,i} \forall j.
\] (11)

Here \( x \) is a constant whose value lies between 0 and 1. In this paper it is taken as 0.75.

In order to solve this problem, we can observe that in the objective function \( \sum_j (\frac{q_{o,ij}}{q_{r,ji}})^x \), both \( \frac{q_{o,ij}}{q_{r,ji}} \) and \( x \) vary between 0 and 1. Two facts are easy to observe about \( (\frac{q_{o,ij}}{q_{r,ji}})^x \); first it is a monotonically increasing function and second its rate of change is monotonically decreasing function. Therefore initially, objective function will get the maximum increment if some amount of resource is allocated to the node corresponding to smallest \( q_{r,ji} \). After some allocation, increase in the value of objective function will decrease and now this increase will be more for some other node that had initially requested more data than first one but less than all the remaining requesting nodes. Therefore it will be beneficial to allocate data to this second node. After certain finite allocations, some other node may result in more increment in objective function and this process continues till the resource allocation is complete.

Based upon the above discussed characteristics of utility function, we propose an algorithm for allocation, which maximizes the overall network utility. The total amount of resource shared by a node \( i \) i.e. \( (Us_i) \) is calculated using the Algorithm 1. Thereafter node \( i \) decides about \( \Delta_i \), the minimum unit of allocation. On the basis of \( \Delta_i \) and the amount of total resource shared \( (Us_i) \), \( i \) calculates the total number of allocation units \( (Us_{i,1}) \) such that \( U_{s,1} = \frac{Us_i}{N} \). Now, \( i \) constructs an allocation matrix \( U_{s,1} \) that has the dimension of \( U_{s,1} \times N \). Here \( N \) is the number of nodes selected for data allocation by Eq. 6. The elements of \( U_{s,1} \) are filled according to formula

\[
U_{s,1}(k, j) = ((k)^x - (k-1)^x) \cdot \left( \frac{\Delta_i}{q_{r,ji}} \right)^x;
\] (12)

where \( k \) and \( j \) are row and column indices respectively of \( U_{s,1} \). Basically \( k \) signifies the total number of \( \Delta_i \) units allocated to a node after current allocation and \( j \) corresponds to id of the requesting node, to which this unit gets allocated.
\((k)^x - (k-1)^x\) is the jth node’s marginal utility or the slope of the utility function after kth allocation. The network utility will be maximized if resources are distributed to that requesting node j which currently has largest marginal utility \(\left(\frac{dU(k,j)}{dt}\right)\). Therefore elements of \(U_{su,i}\) signifying gradient of utility function, are sorted and indices of top \(U_{su,i}\) elements are stored in a vector of dimension 1 \(\times U_{su,i}\). In this way aggregated utility after resource distribution gets maximized. Total number of units allocated to a requesting node is equal to the number of times, id of that node comes in the sorted array.

Generally in a peer-to-peer network, when a node asks for certain amount of resource then it is not guaranteed that it will get all the requested amount of resources. Therefore nodes generally requests more than their actual requirement and refuses the extra resource when they are allocated more than their requirement. This scenario leads to wastage of shared capacity which could have been avoided if nodes had requested their true resource requirement. This problem is easily solved when the resources are distributed to the requesting nodes such that it maximizes the serving node’s utility. In this kind of allocation, nodes asking for less amount of data will be given data first as marginal utility is inversely proportional to \(q_{r,ji}\). Hence if a node asks for more data than its requirement, it loses during the allocation part. Therefore requesting nodes will be compelled to demand resources equal to their actual download requirement.

As requests will be coming temporally in arbitrary fashion, it is necessary to define a policy followed by a node for provisioning the service. If node will serve the request as and when it comes, node will always remain busy in doing so. If node will service the requests periodically, there is a chance that a low reputation node will get the service while a high reputation node may keep on waiting. This happens when currently serving node is busy delivering resource to the low reputation nodes and a high reputation node places a request.

To overcome this issue, a node should have a dynamic policy about serving instants. It means that when sum of reputations of the requesting nodes crosses a certain threshold, node will serve the accumulated requests. If after a threshold period of time sum of reputation of requesting nodes does not cross the threshold, node will serve the requests accumulated by this time irrespective of accumulated sum of reputation. Due to use of sum of reputation as threshold, next serving instant will start with appreciable number of high capacity nodes which reduces chances of high reputation nodes having to wait for service because of low reputation node. Whenever, a node serves new requests, node will first do the selection process for new nodes and then it will redistribute the resources among newly selected as well as earlier pending nodes.

6 Server selection

6.1 Common interest groups

In peer-to-peer file sharing network, many users have common interests. For a user, it is beneficial to choose neighbours which share interests with him and are ready to serve him.

Interest is an abstract notion, thus the classification of nodes on the basis of interest is difficult. Even if it is done, each node may be part of many sets and each set may have number of nodes. This will be difficult to handle. Therefore, interest groups should be formed on the basis of files, users requested or provided. However, users with different interests may sometime request the same file. For example, a song may be liked for different reasons like music, singer or lyrics. But, if two users are requesting for more and more similar files, probably they may have some common interests. As the number of similar files grows, probability of two peers choosing a file due to same interest increases while choosing it for different interests decreases.

Therefore, we propose that a node will compute the similarity coefficient of the other nodes in the network. The similarity coefficient \(r_{ij}\) of node \(j\) will be calculated by node \(i\) using

\[
\phi_{ij} = \left\{ \begin{array}{ll}
\frac{\Omega_{ij}}{\Omega_{ji}} & \text{if } \Omega_{ij} < base_i; \\
\phi_{ji} & \text{otherwise.}
\end{array} \right.
\]

Here \(\Omega_{ij}\) is the total number of `queries exchanged between node \(i\) and \(j\). \(\phi_{ij}\) is the ratio of answered queries to total queries between node \(i\) and node \(j\). As calculation of \(\Omega_{ij}\) and \(\phi_{ij}\) includes queries originating from both participating nodes \(i\) and \(j\), therefore \(\Omega_{ij} = \Omega_{ji}\) and \(\phi_{ij} = \phi_{ji}\). The \(base_i\) will be dynamically and periodically adjusted as per the accuracy of similarity coefficient of the node. It means if the selected neighbours can not answer sufficient number of requests, value of \(base_i\) will be increased.

6.2 Inclusion of reputation in neighbourhood formation

The strategy in the earlier subsection is based on similarity of interest. But forming groups without considering reputation may sometime lead to a group formation where few high capacity peers are grouped with large number of low capacity peers. In this scenario high capacity peer may not be able to receive the high quality of service, which they actually deliver to the network. Therefore, a node should adopt a strategy to form its neighbourhood not only on the basis of similarity of interest but also on the basis of reputation. The novelty with this type of the group formation is that it promotes and aids in the communication between same capacity nodes and at the same time reduces...
the average searching overheads between the nodes of the network.

In earlier community detection algorithms different interest groups are identified such that neighbouring nodes i.e. affiliated to the same group have dense connection or close relationship between them, while nodes which belong to different group will have sparse connection or weak relationship among themselves. In the proposed group formation, nodes form groups such that neighbourhood of two neighbouring nodes are independent of each other i.e. two nodes sharing same interest can have different set of neighbours. Generally such set will not overlap because it is almost impossible for two nodes to share exactly same interests. Like academic interests of two nodes can match but this will not imply that their interest in music will also match.

As discussed earlier, for server selection, a node need to form its neighbourhood using interests and reputation. This can be done by combining reputation \(t_{ij}\) and similarity coefficient \(\chi_{ij}\) for node \(j\). The combined score

\[
score_{ij} = \alpha \cdot \chi_{ij} + (1 - \alpha) \cdot t_{ij},
\]

(14)
can be used to rank the other nodes in the network. This rank can be used to select the node to which queries needs to be sent. In the Eq. 14, \(\alpha\) is a combination coefficient with value between 0 and 1. Value of \(\alpha\) will depend upon the stability of common interest network. If a node has recently joined the network, it has to build the interest network hence \(\alpha\) will be taken high. Once it has a stable interest network, value of \(\alpha\) will be decreased to have more contribution from the reputation in the score.

7 Numerical results

We have done performance evaluation of reputation and resource allocation system for a network consisting of 200 nodes. The system put forward in this paper is simulated for unstructured P2P network. As the main difference between structured and unstructured network is the query routing mechanism employed and it is unrelated to the contribution level and the resources allocation of the node therefore the reputation and resource allocation system introduced in this paper can also be applied to structured or semi structured network. However author haven’t verified this claim through simulation results. The proposed system is being simulated for file transfer as it is the most popular application of peer to peer network. File can be of any type like data, video or audio file. Greater is the bandwidth available for requesting node more quickly it can download file. The high bandwidth becomes almost indispensable for downloading high definition video. Data rate implicitly signifies the bandwidth, therefore data rate available for the down-load is used as the metric for numeric evaluation. We have also evaluated interest based group formation algorithm for a network of 1000 nodes. For the purpose of measurement and estimation discrete time instances are considered in the simulations. Every slot is termed as an iteration. When network is in nascent stage all nodes are initialized and for first 50 iterations also called acquaintance period, nodes randomly allocate their bandwidth without considering the reputation of requester. During the acquaintance period nodes start evaluating the reputation of the nodes from which they have the taken the services. After the end of acquaintance period, nodes distribute resources based on the reputation of requesting peers. After end of acquaintance period a serving node assigns a reputation value of 0.07 to a new node entering the system or an existing node which has never interacted with the serving node. Some finite initial reputation make it possible for the aforementioned nodes to get resources from the network.

Figure 1 presents the average data rate received by nodes sharing different amount of resource to the network. Initially during the acquaintance period, there is no reputation system therefore all nodes receive equal amount of resources. After the end of the acquaintance period, reputation system is implemented and the nodes receive resources in proportion to their contribution to the network. From Fig. 1, it is evident that the free riders are not able to receive any resource and the node that is sharing more data, is getting better quality of service. Increase in the upload capacity from 30 to 60 as compared to increase from 60 to 90, result in much larger gain in the received resources. This implies

![Fig. 1](image-url)
that increase in the received resources is not in the same proportion with increase in the resources uploaded as the node sharing the resources is going towards its optimal point.

Figure 2 demonstrates the network performance for the different percentage of the free riders in the network with and without the reputation system. There is a significant improvement in the network performance with the reputation system. We can observe from Fig. 2 that for 5% to 10% free riders, the decay in network performance is almost negligible. After that, for network with 30% of free riders, performance decreases by small amount. Hence we can conclude that the proposed reputation system, is able to prevent performance deterioration, up to a significant percentage of the free riders.

Figure 3 shows the data rate received by the nodes employing different demand strategies for procuring the resources. BS represents the nodes that request for the amount of resource as per its requirement whereas GS1 and GS2 represents the nodes that requests the amount of resource multiple times their requirement. GS2 requests for larger multiple than the GS1. Here it can be seen that nodes making request as per their requirement are getting better quality of service whereas nodes that are trying to exploit network by making requests for multiple times than needed are not getting better quality of service than BS. This discourages the tendency of nodes to demand more than their actual requirement. It can be observed that quality of service decreases more rapidly form GS1 to GS2 than from BS to GS1 signifying more severe punishment with increase in greediness.

Figure 4 illustrates the advantage of forming common interest groups in peer-to-peer network. It can be clearly seen that, if node forms interest groups, their query gets resolved in much lesser number of hops than without the interest group formation. This happens because most of the nodes’ queries are getting resolved within the group.

7.1 Trust calculation overhead

The objective of this section is to compute time and space complexity of the model proposed in this paper and compare
it with the existing models. Seven variables are assigned values, and ten arithmetic operations which include multiplication, division, exponential and the calculation of the maximum between two quantities are performed in order to compute the reputation of a node by any other node. Clearly time complexity for the single reputation calculation will be $O(1)$. For computing the overall reputation table by a node, the time complexity will be $O(n)$, where $n$ is the total number of nodes in the network. The existing reputation models like the one proposed by Satsiou et al. [21] also has time complexity $O(n)$. Although Satsiou et al. model performs less calculations for computing the reputation but the asymptotic bound is same for both the models.

The reputation model introduced in this paper fares better in terms of the space requirement than Satsiou et al. model because Satsiou et al. model requires storage of last ten transaction whereas there is no such need in the proposed reputation model. The space complexity for the model proposed in this paper is of the order $O(n)$ same as Satsiou et al. model. Although proposed model may require more computation than the existing models for reputation calculation but it saves in terms of the space requirement.

As the model introduced in this paper doesn’t take recommendations, therefore there is no need of computing look out overhead. Clearly the model introduced doesn’t appreciably increase the computation and the lookout overhead as demonstrated by the asymptotic bounds calculated earlier. Therefore model proposed is an efficient model.

8 Conclusion

In this paper, we have discussed allocation of resources by a node on the basis of reputation. We have proposed an algorithm for probabilistic allocation of resource. In this algorithm, requesting node is offered resource with a probability proportional to its reputation. If total demand of selected nodes is more than offering node’s shared capacity, allocation will be done to optimize the gain in reputation of offering node. This algorithm also ensures that nodes do not request more than their actual demand. Another algorithm for formation of common interest group and shared capacity optimization has also been proposed. Numerical results show that proposed algorithms work as per the requirement.

Theoretical verification of efficiency of common interest group formation algorithm and game theoretic analysis of resource sharing algorithm needs further investigation.

References

1. Gnutella website (2007). http://www.gnutella.com/
2. Adar E, Huberman BA (2000) Free riding on gnutella. First Monday 5:2000
3. Andrew O (1999) Data networks are lightly utilized, and will stay that way. Technical report
4. Banerjee D, Saha S, Sen S, Dasgupta P (2005) Reciprocal resource sharing in p2p environments. In: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, AAMAS ’05, pp 853–859. ACM, New York
5. Boyd S, Ghosh A, Prabhakar B, Shah D (2006) Randomized gossip algorithms. IEEE/ACM Trans Netw 14(SI):2508–2530
6. Chen J, Huijuan L, Bruda SD (2009) A solution for whitewashing in p2p systems based on observation preorder. International Conference on Networks Security, Wireless Communications and Trusted Computing 2:547–550
7. Farahbakhsh R, Crespi N, Cuevas A, Adhikari S, Mani M, Sanguankotchakorn T (2012) socp2p: P2p content discovery enhancement by considering social networks characteristics. In: Computers and Communications (ISCC), 2012 IEEE Symposium on, pp 000530–000533
8. Feldman M, Lai K, Stoica I, Chuang I (2004) Robust incentive techniques for peer-to-peer networks. In: Proceedings of the 5th ACM conference on Electronic commerce, EC ’04, pp 102–111. ACM, New York
9. Gupta R, Singh YN (2013) Avoiding whitewashing in unstructured peer-to-peer resource sharing network. CoRR, arXiv:1307.5057
10. Gupta R, Singh YN (2013) Trust estimation in peer-to-peer network using blue. CoRR, arXiv:1304.1649
11. Karakaya M, Korpeoglu I, Ulusoy O (2009) Free riding in peer-to-peer networks. IEEE Internet Computing 13(2):92–98
12. Kung HT, Wu C-H (2003) Differentiated admission for peer-to-peer systems: Incentivizing peers to contribute their resources. In: Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems
13. Liu G, Shen H, Ward L (2012) An efficient and trustworthy p2p and social network integrated file sharing system. In: 2012 IEEE 12th International Conference on Peer-to-Peer Computing (P2P), pp 203–213
14. Ma H, Leung Hf (2006) A demand and contribution based bandwidth allocation mechanism in p2p networks: A game-theoretic analysis. In: 2006 20th International Conference on Advanced Information Networking and Applications, AINA 2006, vol 1, pp 1005–1010
15. Ma RTB, Lee SCM, Lui JCS, Yau DKY (2006) Incentive and service differentiation in p2p networks: a game theoretic approach. IEEE/ACM Trans Netw 14(5):978–991
16. Meo M, Milan F (2006) A rational model for service rate allocation in peer-to-peer networks. In: Proceedings of the 25th IEEE International Conference on Computer Communications, INFOCOM 2006, pp 1–5
17. Padhye J, Firoiu V, Towsley D, Kurose J (1998) Modeling tcp throughput: a simple model and its empirical validation. SIGCOMM Comput Commun Rev 28(4):303–314
18. Papaioannou TG, Stamoulis GD (2006) Reputation-based policies that provide the right incentives in peer-to-peer environments. Comput Netw 50(4):563–578
19. Peng D, Liu W, Lin C, Chen Z, Peng X (2008) Enhancing tit-for-tat strategy to cope with free-riding in unreliable p2p networks. In: Proceedings of the 2008 Third International Conference
Ruchir Gupta He was born in Jhansi, India. He has done his Ph.D. on ‘Reputation Management System in Peer-to-Peer Networks’ in December 2013 from IIT Kanpur. He is assistant professor in Computer Science & Engineering department at IIITDM Jabalpur. He has interests in Peer-to-Peer networks, Complex Networks and application of game theory in networks.

Nitin Singha He was born in Shimla, India. He is currently pursuing his Ph.D from department of Electrical Engineering at IIT Kanpur. His research interest includes Peer-to-Peer networks, Wireless Sensor Networks, Complex Networks and application of game theory in networks.

Ruchir Gupta  

Yatindra Nath Singh He was born in Delhi, India. He was awarded Ph.D for his work on optical amplifier placement problem in all-optical broadcast networks in 1997 by IIT Delhi. In July 1997, he joined EE Department, IIT Kanpur. He was given AICTE young teacher award in 2003. Currently, he is working as professor. He is fellow of IETE, senior member of IEEE and ICEITE, and member ISOC. He has interests in telecommunications’ networks especially optical networks, switching systems, mobile communications, distributed software system design. He has supervised 7 Ph.D and more than 97 M.Tech theses so far. He has filed three patents for switch architectures, and have published many journal and conference research publications. He has also written lecture notes on Digital Switching which are distributed as open access content through content repository of IIT Kanpur. He has also been involved in opensourse software development. He has started Brihaspati (brihaspati.sourceforge.net) initiative, an opensource learning management system, BrihaspatiSync – a live lecture delivery system over Internet, BGAS – general accounting systems for academic institutes.