

# A Secure Hybrid Reputation Management System for Super-Peer Networks

Ghassan Karame

Department of Computer Science  
ETH Zürich  
8092 Zürich, Switzerland  
karameg@inf.ethz.ch

Ioannis T. Christou

Athens Information Technology  
19.5km Markopoulo Ave., Peania  
19002, Athens, Greece.  
ichr@ait.edu.gr

Tassos Dimitriou

Athens Information Technology  
19.5km Markopoulo Ave., Peania  
19002, Athens, Greece.  
tdim@ait.edu.gr

**Abstract**—In this paper, we propose a novel hybrid system for handling reputation in Super-Peer-based networks by combining the personal history of each user’s interactions with other users, the opinions of peer-friends together with global ratings of peers as they emerge from all of their interactions with other users of the network. We introduce the notion of *peer friends* in a P2P network and use it to prevent malicious collectives from reducing the reputation of a peer in the network. We also present a secure distributed framework that ensures that trust reports remain *encrypted* and are never opened during the submission or aggregation process. Computational results from our distributed prototype simulation show that our solution compares favorably with all other proposed methods for handling reputation when subject to various malicious strategies.

**Index Terms**—Social Networks, Peer-to-Peer, Super-peer Networks, Trust, Reputation, Security.

## I. INTRODUCTION

Peer-to-peer (P2P) systems are gaining increased popularity due to their ability to harness large amounts of resources (e.g. Gnutella[2]) and to scale and self-organize in the presence of a highly transient population of nodes. Since P2P networks enable files to be hosted by unknown users, there is an omnipresent risk that the downloaded files might carry various kinds of malicious software like viruses and worms. This suggests that a number of trust/reputation issues need to be addressed in P2P networks as means of providing popularity based ranking in order to help users discover the desired authentic data ([4], [5], [8], [9], [10]). Most of the proposed approaches rely on distributed polling protocols, thus incurring significant messaging overhead and load imbalance in the network. Furthermore, they fail to handle *spurious* rumors spread by independent and colluding peers.

This paper proposes a novel hybrid mechanism for handling reputation in P2P file sharing networks. Although our model can be easily integrated in most P2P topologies, we *only* target in this work super-peer architectures mainly due to their widespread use (e.g: Gnutella (v0.6), KaZaA [1]) and performance superiority when compared to other P2P architectures. Our model combines *i)* the personal history of each user’s interactions with other users, *ii)* the opinions of personal acquaintances, and *iii)* the global ratings of peers as they emerge from all of their interactions with other users of the network. We also introduce the notion of trusted

*friends*, consisting of a subset of peers that share previous social experience and constantly provide each other with their feedback on all the interactions they established with other peers. While it is true that “friends” may not always behave correctly and introduce spurious opinions, we devise some methods in Section III-D to defend against such malicious friends. The proposed combination of local and global metrics is highly resistant to malicious rumors spread by dishonest peers and mobs of malicious nodes. All the aforementioned parameters are securely pooled together in a general trust metric for evaluating an opinion about the trustworthiness of a peer in the P2P community. In addition, we make sure that *anonymity*, *privacy* and *integrity* of the votes, *fairness*, etc., are preserved when aggregating the global ratings.

The remainder of the paper is organized as follows. Section 2 highlights the related work in the area. Section 3 introduces our proposed solution of combining local and global metrics in order to evaluate the trustworthiness of peers. Section 4 describes how to ensure the security, anonymity and authenticity of the submitted ratings. Finally, in Section 5, we evaluate our simulation results derived from the integration of our approach with realistic P2P file sharing settings.

## II. RELATED WORK

Although the literature includes several proposals for reputation management in P2P systems, to the best of our knowledge, no previous work proposed the use of peer inter-personal relationships to strengthen trust handling in P2P networks.

A survey of the various research proposals in the area of trust and reputation is presented by Marti *et al.* [3] in the context of a taxonomy of trust and reputation systems for P2P Networks. Cornelli *et al.* [4] propose to base reputation sharing on distributed polling whereby a requestor *u* can assess the reliability of resource providers by polling its peers. In [5], this work has been extended to cover Super peer networks. However, in both works, there is no indication how this approach eliminates malicious votes if dishonest peers badly rate good peers and reconfirm their ratings when contacted. Yu and Singh [11] developed a model of reputation management based on the Dempster-Shafer theory of evidence. Each peer has a set of acquaintances, a subset of which are identified as its neighbors, that a peer would refer to in order to

investigate the ratings of another peer. However, as with other reputation approaches, this approach does not fully protect against malicious ratings generated by malicious agents.

EigenTrust [9] attempts to handle spurious ratings by assigning to every peers a unique global reputation value computed using an algorithm similar to PageRank by relying on the presence of *a-priori good nodes* whose reputation never goes down. However, this solution is not feasible in large scale P2P systems where some nodes might be unreachable [7].

### III. HYBRID REPUTATION MODEL

#### A. Design Goals

In this work, we focus on developing an efficient reputation management system resistant to rumors and colluding malicious peers attacking the reputation of loners. Our approach reduces several problems encountered in most reputation-based approaches, namely: whitewashing<sup>1</sup> and pseudo-spoofing<sup>2</sup> attacks, while maintaining load balance in the network.

Furthermore, we argue that nothing must affect the submission process: peers must not be able to affect the system if they submit wrong values, colluding peers must not be able to alter the resulting ratings and the contents of the votes must not be visible even by trust handling peers. In addition, our proposed model should incur minimal messaging/computational overhead.

#### B. P2P Communication Model

Since we are *not* concerned with proposing a new communication/peer-discovery protocol, we assume throughout this paper a variant of the traditional Super-peer model as implemented in KaZaA [1] and Gnutella v0.6. In the latter model, each peer is either a super node (*SN*) or an ordinary node (*ON*). TCP connections are established between *ON* and its *SN* and between pairs of *SN*s. Each *SN* tracks the content of its *ON*s. Similarly, in *K*-redundant Super-peer networks, *K*-super peers work in a round robin fashion for each cluster to decrease the load on super peers [12].

#### C. Algorithm Description

In this section, we present the details of our approach. We assume that each peer is associated with a number of peer “friends”. Peers can adaptively choose their friends based on some previous social experience, personal history, etc.. Such interpersonal relationships are more likely to coexist with the various P2P trends, owing to the heterogeneity present in these systems (e.g. friendship web-based applications: FaceBook [13] and Friendster [14]). In this paper, friendship connotes a relationship between two entities that share mutual knowledge and trust (e.g. web of trust of PGP [15]).

As depicted in Figure 1, upon reception of *v*’s request for download, peer *u* issues a reply confirming his possession of the requested resource. In addition, one of the superpeers

<sup>1</sup>Whitewashing describes the act in which malicious peers rejoin the network using a new ID to clean their previous bad record.

<sup>2</sup>Pseudo-spoofing refer to the creation of multiple IDs on different hosts by the *same* malicious user as a mean to affect the reputation building process.

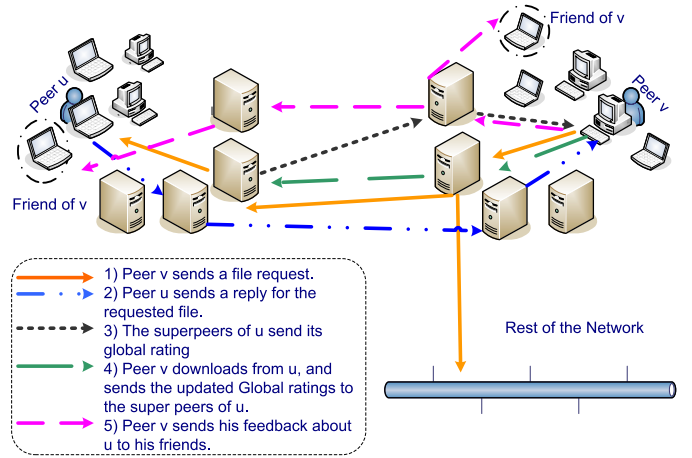


Fig. 1. Protocol Specification

of *u* responds to *v* with the global rating of *u*. The latter metric refers to the accumulation of all the peers’ opinions about *u*. The superpeers of *u* securely compute, aggregate and communicate its global rating (Section IV).

#### Algorithm 1: Distributed Reputation Building

1. Suppose peer *u* is requesting for a resource *r*. *u* first sends its request query in the network.
2. *u* awaits for peer replies. Suppose *V* is the finite set of all peers possessing *r*.
3. *u* receives *V*’s various global ratings (Section IV-A)
4. **for each** replier *v* in *V* **do**:
5.     *u* fetches Personal History  $P_v$  with *v* from local database,
6.     *u* fetches friends’ opinions  $F_v$  about *v* from local database,
7.     *u* locally computes *v*’s aggregate rating by combining *v*’s global rating with  $P_v$  and  $F_v$  (Section III-D)
8.     **end**
9. *u* picks offerer  $v_r$  that has a high reputation metric.
10. *u* downloads *r* from  $v_r$ .
11. *u* updates its interaction history with  $v_r$  and save it locally.
12. *u* sends its rating for the last interaction with  $v_r$  to be processed by the *super peers* of  $v_r$ .
13. *u* updates its friends with its new opinion about  $v_r$ .

After receiving the various providers’ replies, peer *v* calculates its own perception of the replying peers’ reputations based on their global rating, the advertised opinion of *v*’s friends about them and *v*’s interaction history with them (Section III-D). We refer to this perception as the aggregate rating of peer *v* for peer *u*. Once *v* computes the aggregate ratings of the offerers, it chooses a suitable peer to interact with. Then, peer *v* issues an *encrypted* rating of the interaction it had with peer *u*. Upon reception of the vote, the global trust value of *u* is securely updated by *u*’ superpeers. Finally, *v* updates its *friends* with its aggregate rating for *u*. A summary of these various steps in shown in Algorithm 1.

#### D. Ratings Aggregation

In our model, a peer’s trustworthiness is defined by an evaluation of its behavior towards other peers in the network. We identify three factors for such an evaluation: the feedback a peer obtains from its friends, the global rating of the offerer, reflecting the feedback from all the peers in the system and the personal interaction history with the offerer.

The aggregate rating of peer  $u$  is particular to every peer as it combines both global and local metrics. Upon completion of every interaction,  $v$ 's friends update  $v$  with their *latest* feedback, which will be stored locally in  $v$  for later use. This allows for "on-the-fly" computation of the value of aggregate rating, without the use of expensive polling protocols. The resulting trust metric is computed by means of weighted averaging, as follows:

$$R_{vu} = w_{Gvu} \cdot G_u + w_{Fv} \cdot F_{vu} + w_{Pvu} \cdot P_{vu}$$

- $R_{vu}$  : the *aggregate* metric for peer  $u$  as seen by peer  $v$ .
- $w_{Gvu}$  : the weight assigned by  $v$  for the aggregation of the global reputation of  $u$ .
- $G_u$  : the global reputation of  $u$  in the system, advertised and manipulated by  $u$ 's superpeers (Section IV-A).
- $w_{Fv}$  : the weight assigned by  $v$  for the aggregation of  $v$ 's friends opinions about  $u$ .
- $F_{vu}$  refers to  $v$ 's aggregate friends' opinions about  $u$ .  $F_{vu}$  is computed using simple averaging of the friends' opinions.  $F_{vu}$  excludes the opinions of suspicious friends by taking a "fault tolerant" average. That is, a subset of friends' opinions is chosen that do not differ from one another by more than a specified amount, and  $F_{vu}$  could be calculated using *only* these opinions.
- $w_{Pvu}$  : the weight assigned by  $v$  for the aggregation of the personal history of peer  $v$  with peer  $u$ .
- $P_{vu}$  refers to the personal history of peer  $v$  with  $u$ . It denotes the ratio of the number of "authentic" interactions to the total number of interactions between  $v$  and  $u$ . Note that all weights  $w_{Gvu}$ ,  $w_{Fv}$  and  $w_{Pvu}$  are *local* variables and thus are particular to peer  $v$ .

The magnitude of the resulting local reputation metric  $R_{vu}$  reveals peer  $u$ 's credibility as interpreted by peer  $v$ . In case a peer recently joined the system, its local reputation would be zero. This results in *cold start* problems since the new joining peer would gain limited exposure due to its low reputation. We propose to resolve this problem by setting  $R_{vu}$  to  $w_{Pvu} \times 5$  if  $u$  did not interact<sup>3</sup> with any peer in the network.

#### E. Selection of the Averaging Weights

The choice of the local averaging weights  $w_{Gvu}$ ,  $w_{Fv}$  and  $w_{Pvu}$  is particular to every peer and thus may differ between nodes present in the network. In order to cope with dynamics of P2P networks, we assume that peer  $v$ 's client *dynamically* adjusts these weights according to peer  $v$ 's view about its friends, other malicious peers, etc.. Nevertheless, we suggest some heuristics derived from extensive simulations for choosing the various weights:

- The sum of any two averaging weights *should* be larger than their third counterpart to ensure that any single malicious party is not able to impact peer  $v$ 's decision.
- $P_{vu}$  should be assigned with a significant weight ( $> \frac{1}{3}$ ) since no malicious party can affect its value.

<sup>3</sup>The number of interactions of peer  $u$  is *securely* provided by its Super peer(s) when peer  $v$  queries  $u$ 's global reputation (refer to Section IV).

After extensive simulations, we identified the triplet  $(w_{Gvu}, w_{Fv}, w_{Pvu}) = (0.2, 0.35, 0.45)$  to best cope with the multitude of malicious strategies studied in Section V.

#### F. Incentives for "Honest" Peers

Nowadays, most P2P systems provide various incentive mechanisms in order to encourage peers to actively participate in P2P file sharing networks. These existing schemes are in some sense *orthogonal* to reputation-based systems; as peers accumulate higher reputation metrics, they are likely to serve more requests for download. Therefore, highly reputable peers are more likely to experience additional privileges (e.g: increased download bandwidth).

### IV. SECURE RATINGS AGGREGATION

In this section, we describe a framework that ensures the anonymity, integrity, soundness and fairness of the ratings aggregation process.

#### A. Securing the aggregation of Global Ratings

We propose the following approach in order to secure global ratings aggregation (see [16] for more details). Our protocol makes use of two essential cryptographic tools: *i*) *homomorphic* encryption functions, and *ii*) a  $(t, n)$  threshold secret sharing scheme to protect against malicious aggregators submitting erroneous decryptions.

When a peer  $v$  joins the network, a trusted certificate authority assigns a random set of the Super peers pertaining to its cluster to be the  $n$  aggregators for  $v$ . These will be responsible for decrypting the aggregated trust value of  $v$  that remains encrypted in the system. This group has an associated public key  $PK_{vA}$ , which is used by other peers  $u$  to rate their interactions with  $v$  and submit their encrypted reports into the system. The corresponding decryption key  $SK_{vA}$  is *shared* amongst the aggregators using the threshold cryptosystem. In addition, the *CA* assigns a storage repository for each cluster chosen from the most reputable and trusted Super peers. When a peer  $u$  has interacted with peer  $v$ , it issues a report indicating the global rating of this interaction. Here, we assume that  $u$  has acquired the public key  $PK_{vA}$  through a previous reply received from  $v$ . Given the public key,  $u$  constructs the following message  $m$ :

$$m = E_v^A(Value) | T_i, u, v, Sig_v(T_i, u, v) | TimeStamp,$$

where  $E_v^A(Value)$  is the vote submitted by  $u$  encrypted with the public key of  $v$ 's aggregators and  $T_i, u, v, Sig_v(T_i, u, v)$  is a proof of interaction between  $u$  and  $v$ .

Then it transmits message  $m$  (along with its digital signature) and its public key/certificate pairs. Upon receiving the trust report for  $v$ , the aggregators update  $v$ 's reputation using the homomorphic property, and submit the aggregation result back to the storage Super peer in order to guarantee the durability of the ratings in the system. In turn, the storage Super peer *only* stores the encrypted value that was advertised by the majority of the aggregators. The invariant that is maintained at any moment is that the current aggregate value resides

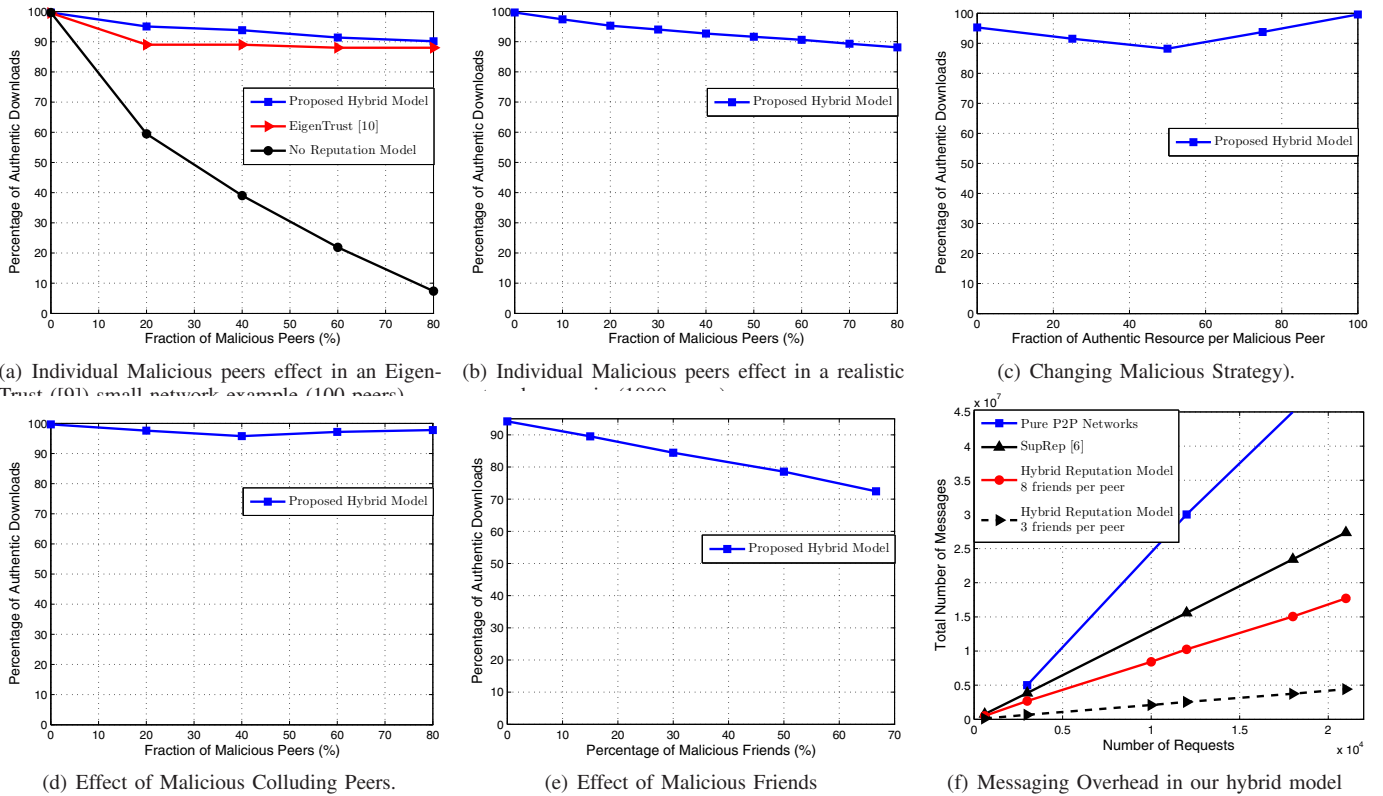


Fig. 2. Experimental Results of our Hybrid Model. Confidence intervals were omitted from the figures due to their negligible size.

encrypted in the system. Furthermore, only *eligible* peers with authentic proof of interaction  $\langle T_i, u, v, Sig_v(T_i, u, v) \rangle$  are able to submit their ratings.

### B. Securing the Friends Interactions

Since our protocol relies on existing trust relationships amongst friends, securing the friends interactions in our system reduces to guaranteeing the confidentiality and authenticity of the various advertised reports. For that purpose, we require that peer friends exchange symmetric keys, to be used for encrypting prospective reports. We argue that this scheme does not affect the soundness of the overall system since any vote can be later denied by both friend parties (symmetric encryption lacks the non-repudiation property).

## V. EXPERIMENTAL RESULTS

We have implemented our approach using JADE solution [17] as the main infrastructure for exchanging messages between peers. Our experiments aimed at analyzing three main performance metrics; namely, the effect of malicious peers, messaging costs and load balancing properties of the system. Table I depicts our implementation setup.

### A. Effect of Malicious Peers

1) *Individual Malicious attacks*: In this scenario, malicious peers always provide high ratings for other malicious peers and low ratings for honest peers. Figures 2(a) and 2(b) illustrate our results. Our model performs well even if the majority of the peers in the network are malicious, when compared

Time to live for query messages	5
Total Number of Peers	1000
Average number of files per Peer	8
File Replication Factor	2.45
Average # of friends per Peer (Min. = 0, Max. = 20)	10
Prob. that a good peer returns inauthentic resource	0.02
Prob. of a malicious peer returning authentic resource	0.03
Number of popular files attacked	4
Number of requests to attacked files	1000
Number of Attack Rounds	4
# of experiments over which results are averaged	6
Malicious Collectives Size	50-200 Peers
Number of Malicious Collectives	4
$(w_{Gvu}, w_{Fv}, w_{Pvu})$	(0.2, 0.35, 0.45)

to the EigenTrust approach [9]. In fact, malicious peers can not significantly affect peers' relative aggregate ratings as perceived by other honest peers and their acquaintances in the network. Note that our model is expected to even perform better as the network stabilizes (i.e: all peers/friends have interacted with other peers), nevertheless, Figure 2(a) (100 peers, and 50 total requests) suggests that our approach is very robust even at the first stages of trust establishment.

2) *Changing Malicious Strategies*: Here, we simulate a more challenging scenario where malicious peers respond up to  $f\%$  with authentic content (refer to [9] for a similar scenario). In this manner, they attempt to increase their ratings by providing authentic downloads to some peers, while offering malicious content to other peers. The simulation results for this scenario, featuring 20 % malicious peers in the network, are illustrated in Figure 2(c). Our results show that our approach

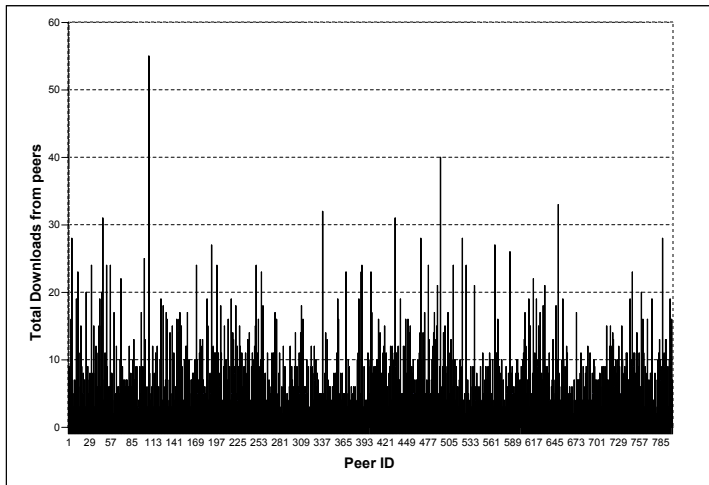


Fig. 3. Peer Load Share in the Hybrid Reputation model. We show the number of uploads per “honest” peer corresponding to 10000 requests.

limits the number of authentic downloads to 87 % when malicious peers behave randomly (i.e:  $f = 50$  %), resulting in 16% improvement over EigenTrust [9].

3) *Colluding Malicious Peers*: This scenario comprised of three stages. In the first stage, all peers interact normally, requesting for random files. In the second stage, a resource  $r$  becomes very popular and malicious peers post a bad version of  $r$ , form a collective in order to attack *all* the offerers of  $r$ , while in the same time, increasing the reputation of the malicious peer responsible for the file. The third and final stage of the model consists of a large number of requests for the popular file in question, by random peers.

As shown in Figure 2(d), *honest* peer’s reputations are not affected by the colluding malicious peers, since the aggregate trust metric depends, to a large extent, on reliable parameters that are *not vulnerable to malicious rumors*.

4) *Effect of Malicious Friends*: We simulate a scenario in which up to  $m$  % suspicious friends submit *spurious* opinions to promote the reputation of other malicious peers present in the network. At all times, these malicious friends provide *authentic* content for the various requestors in order to maintain a high credibility metric in the P2P network.

Figure 2(e) shows that these deceiving opinions have limited impact on the total percentage of authentic downloads even when up to 40 % of the peer friends provide inauthentic opinions. The use of the fault tolerant approach alleviates this attack by only using those friends’ opinions that do not differ from one another by some specified amount. As the percentage of malicious friends increase beyond 50 %, the performance of the system significantly degrades. We argue, however, that this is not likely to occur in reality since friendships often emerge from pre-existing interpersonal relationships.

### B. Messaging Costs

In order to evaluate the messaging overhead of our model, we have evaluated the performance of our hybrid reputation model in a 7-redundant Super peer network, featuring 5 aggregators per cluster. As shown in Figure 2(f), our scheme achieves much lower overhead when compared to other proposed protocols such as [4], [5] and [9]. In fact, these protocols

exhibit high stress in the network, since they require the requestor peer to poll all other peers.

### C. Load Balancing

In our hybrid model, every peer shares a unique and particular aggregate rating with respect to other participants as the reputation value includes *peer-local* metrics. This leads to the safeguarding of the load balancing properties initially pledged by the P2P model, when compared to other proposed schemes. As seen from Figure 3, our scheme results in a relatively balanced load distribution across network peers.

## VI. CONCLUSION

In this paper, we have presented a novel hybrid mechanism handling reputation in a Super Peer-based P2P network. Our model is based on social theory implications for evaluating the credibility of anonymous members, where personal and friends’ experience is highly valued when compared to the rumors spread in the community. This combination of local and global metrics reduces the effect of omnipresent *rumors* spread by dishonest peers. We have also presented a framework that secures the process of ratings aggregation and calculation.

Our simulation results demonstrate that our approach severely limits inauthentic file downloads, when subject to individual/colluding malicious peers and features low messaging overhead and increased load balance when compared to other reputation management models.

## REFERENCES

- [1] KaZaA: Available from <http://www.kazaa.com/us/index.htm>.
- [2] The Gnutella Protocol Specifications v0.4. Available from <http://www.clip2.com>.
- [3] S. Marti and H. Garcia-Molina, “Taxonomy of trust: Categorizing P2P Reputation Systems”, In *Computer Networks*, 2006.
- [4] F.Cornelli, E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P. Samarati, “Choosing Reputable Servents in a P2P Network,” In *Proceedings of WWW*, 2002.
- [5] S. Chhabra, E. Damiani, S. De Capitani di Vimercati, S. Paraboschi and P. Samarati, “A Protocol for Reputation Management in Super-Peer Networks”, In *Proceedings of DEXA*, 2004.
- [6] L. Mekouar, Y. Iraqi and R. Boutaba, “Peer to Peer’s most wanted: Malicious Peers”, In *The International Journal of Computer and Telecommunications Networking archive* Volume 50, Issue 4, (March 2006).
- [7] B. Yu, M. P. Singh, and K. Sycara, “Developing Trust in Large-Scale Peer-to-Peer Systems”, In *Proceedings of MAS&S*, 2004.
- [8] L. Xiong, L. Liu, “PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities,” In *IEEE Transactions on Knowledge and Data Engineering*, 2004.
- [9] S.D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, “The EigenTrust Algorithm for Reputation Management in P2P Networks,” In *Proceedings of WWW*, 2003.
- [10] M. Haque, S. Ahamed, “An Omnipresent Formal Trust Model for Pervasive Computing Environment”, In *COMPASAC*, 2007.
- [11] B. Yu and M. P. Singh, “Detecting deception in reputation management”, In *Proceedings of AAMAS*, 2003.
- [12] B. Yang and H. Garcia Molina, “Designing a Super-Peer Network”, In *Proceedings of the ICDE*, 2003.
- [13] Facebook: <http://www.facebook.com/>.
- [14] Friendster: <http://www.friendster.com/>.
- [15] PGP: [www.pgp.org](http://www.pgp.org).
- [16] T. Dimitriou, G. Karame and I. Christou, “SuperTrust: A Secure and Efficient Framework for Handling Trust in Super Peer Networks”, To Appear In *Proceedings of ICDCN*, 2007.
- [17] Jade: Java Agent DEvelopment Framework, Available from <http://jade.tilab.com/>.