

A Frequent-Sharer Program for Peer-to-Peer Systems

Minaxi Gupta, Paul Judge, Mostafa Ammar
College of Computing
Georgia Institute of Technology
 {minaxi, judge, ammar}@cc.gatech.edu

Abstract—

Airline frequent-flyer programs reward loyal customers with benefits like free tickets, seat upgrades, and priority check-in and boarding. Inspired by the concept, this position paper proposes an optional *frequent-sharer*¹ program for p2p systems. The frequent-sharer program is based on a point system with goals of providing incentives to peers to contribute to the system in order to receive better service from it. The point system is comprised of an *accounting component* that keeps track of each enrolled peer’s contribution to the system, and an *award component* that provides a level of service to each peer based on her level of contribution. We also propose a mechanism based on the concept of *reputation systems* to track the contributions of the enrolled peers securely.

I. INTRODUCTION

Peer-to-peer (p2p) systems are founded on the fundamental principle of cooperation among the peers. Peers mutually benefit from peers’ willingness to provide service to other peers. Users are drawn to these systems due to the ability to locate a wide variety of content. Sharing content and maintaining connectivity are important factors in the success of such systems. This position paper focuses on unstructured decentralized p2p systems like Gnutella [1], where locating and retrieving the existing close-by copy of the content requires peers to be mutually cooperative.

Inspired by the airline *frequent-flyer* programs, we propose an optional *frequent-sharer* program to provide an incentive for the peers in a p2p system to share and serve the content they download (or generate), stay online longer and hence contribute to the system in order to receive better service from it. The program is based on a point system that serves to identify the contribution of each peer who enrolls in the program and provides guidelines for various levels of service to peers based on the level of contribution. We also propose a security mechanism based on the concept of *reputation systems* to prevent malicious and colluding peers from thwarting the tracking of peer contributions.

In the absence of any incentive mechanism to share content, peers in p2p systems have sufficient motivation to be free-loaders. Free-loaders (aka free-riders) are peers who

only download content but never serve it to other peers. This is evidenced by several measurement studies of existing p2p systems like Napster and Gnutella. Study [2] concluded that at the time, nearly 70% of Gnutella users shared no files, and nearly 50% of all responses were returned by the top 1% of sharing hosts. Study in [3] quotes the free-loaders to be about 25% in Gnutella and far lesser in Napster. It is such free-loaders that the frequent-sharer program attempts to motivate to contribute.

The roadmap to the rest of the position paper is as follows. Section II provides an overview of the frequent-sharer program. Section III describes the point system in detail and section IV describes the reputation system based security mechanism. Finally, section V concludes the paper.

II. OVERVIEW OF THE FREQUENT-SHARER PROGRAM

The basis of the frequent-sharer program, the point system consists of two primary components. The first component is the *accounting component* that keeps track of the contribution of each peer who is enrolled in the program. It periodically updates points for the peers that enroll in the frequent-sharer program, as they serve content. The accounting component takes into consideration factors like file popularity, file sizes, and peer bandwidth. Special emphasis is given to encourage peers to serve hard-to-find or unpopular content. The accounting component also gives an incentive to the peers for staying online longer. The second component of the point system is the *award component*. A peer is provided a level of service (LoS) based on the number of points she earns in the system. Peers who choose not to enroll in the frequent-sharer program are provided a basic LoS throughout. The idea is to continue to provide an acceptable LoS to uncooperative peers or peers who chose not to enroll in the program, but differentiate among various peers by providing *enhanced* and *premium* LoS to peers who are sharing and serving more content than others. The prevalent unstructured decentralized p2p systems like Gnutella have a content search phase and a retrieval phase. A combination of parameters that impact each of these phases are used to map to the various LoSs.

Because enrollment of peers in the frequent-sharer program is optional, the accounting component runs only on

¹to mean peers who share frequently

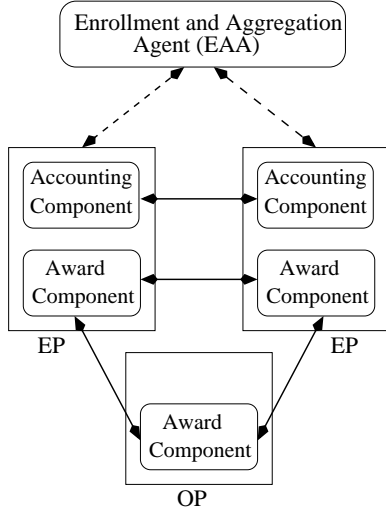


Fig. 1. Components of the Point System. Communication along the dashed lines is periodic and that along the solid lines is per transaction.

peers that choose to enroll in the program, but the award component requires each peer in the p2p system to run enhanced functionality of providing appropriate LoS to other peers. This is important to encourage peers to enroll in the program.

Peers can choose to enroll in the frequent-sharer program any time by contacting an enrollment and aggregation agent (EAA). The EAA could be replicated for efficiency purposes. They can also decide to un-enroll at any time. The first time an enrolled peer joins a p2p network, she is eligible to receive the basic LoS. Through their participation in the system, peers enrolled in the frequent-sharer program can earn points and upgrade to a better LoS, while peers who do not enroll in the program continue to receive the basic LoS. The point system allows peers to save their points across sessions. Thus, a cooperative peer can maintain benefits of its participation in the system in spite of being offline for a while.

The accounting component involves keeping a counter of points indicative of each enrolled peer's contribution to the system. This counter is sent along with each query and the retrieval request. Based on the value of the counter, peers who process the query and the peer who serves the content decide on the LoS that the peer requesting the document is eligible to receive. In a perfect world, each peer's local software can update the counter that keeps track of her points. However, this simple mechanism could be thwarted by the peers altering the point computation to their benefit or by tempering with the value of the stored counter. Hence, there is a need for security mechanisms to enforce secure point computation. We propose a *reputation-based* security mechanism that uses (public, private) key pair and the EAA to ensure fair periodic updates to each enrolled peer's points in the system, still ensuring points for each peer are kept locally for fast retrieval during content search and download.

There are two ways that malicious peers can hinder the proper functioning of the award component. First, they may not give better LoS to peers who are eligible for enhanced or premium LoS. Second, they may collude with other peers and give each other a better LoS than the points justify. Though some peers could be malicious but the success of p2p stems in part due to mutual goodwill. Peers may choose not to enroll in the frequent-sharer program for various reasons. One reason could be if they are not frequent users of the system. Moreover, some peers may not want to get their points tracked. The award component works on the belief that whether or not a peer enrolls in the program, if she trusts the points of other enrolled peers, she would be willing to provide them with the appropriate LoS. This is for three reasons. First, it is difficult for peers to know if they have indeed received the LoS they were eligible for. Second, a light-weight secure enforcement of the award component does not seem possible because of the possibility of a man-in-the-middle attack. Third, rewards from malicious behavior in the reward component are limited, unlike those possible by altering the point computation of the accounting component. As a result, section IV only describes the security measures for the accounting component.

Figure 1 shows the various components of the point system to be run on enrolled peers (EPs) and other peers (OPs) in the system. The communication between peers shown in the figure takes place only during search and retrieval phases and that with the EAA is periodic.

III. DETAILS OF POINT SYSTEM

As mentioned before, there are two main components of the point system:

- An *accounting component* that credits points in an enrolled peer's account based on her level of contribution in the p2p network.
- An *award component* that chooses appropriate level of service (LoS) for each peer based on the number of points she has earned in the system.

A. Accounting Component

The basic idea behind accounting is very simple, enrolled peers can receive better LoS for their downloads if they serve more content and stay online longer. The first time a peer joins a p2p network, she starts out with zero points and is only eligible for the basic LoS. Through her contribution to the system, she can earn more points and become eligible for better LoS (enhanced or premium). In order to ensure that peers continue to cooperate even after becoming eligible for enhanced or premium LoS, points earned by a peer expire periodically. Hence, peers have to keep contributing to the system in order to retain or upgrade their LoS. Peers who are not enrolled in the frequent-sharer program

always receive a basic LoS. Peers retain their points across sessions, so that a peer can continue to receive better LoS if she was a cooperative participant of the system in the past. For now, assume that the points for each enrolled peer are updated and maintained in the local software. Section IV discusses the details of how the points are periodically updated by the enrollment and aggregation agent (EAA) that handles the enrollment in the program because of security implications of this naive approach. Points are kept locally for fast retrieval but encrypted by the central authority. Sections III-A.1 and III-A.2 detail the algorithms that are used to credit an enrolled peer's points.

1) *Basic Accounting*: In the most simple case, a peer enrolled in the program *earns* a point for every request it serves. We call this scheme the *one-for-one accounting*. This scheme has some obvious shortcomings. It offers peers an incentive to serve requests for very short files in order to keep their upstream bandwidth consumption to a minimum. This can be avoided by earning points in terms of content size (i.e., every megabyte of content downloaded pays a point and every megabyte of content served earns a point). We call this scheme the *per-MB accounting*. Further, a peer with high bandwidth connection will be able to serve content faster than a peer with a slower connection. The accounting should account for a peer's serving capacity as well. We assume that methods exist to find out peers' connectivity since it was found in study [3] that peers often state their bandwidth availability incorrectly.

Another issue is that some files are popular while others are not. Since unpopular files may not be accessed frequently, the peers who keep them in the shared directory need to be given credit for being online and ready to serve the hard to find content. Figure III-A.1, describes a basic accounting algorithm that updates the points for a peer with bandwidth bw , based on the file sizes served, its bandwidth, and gives credit to the peer for being online.

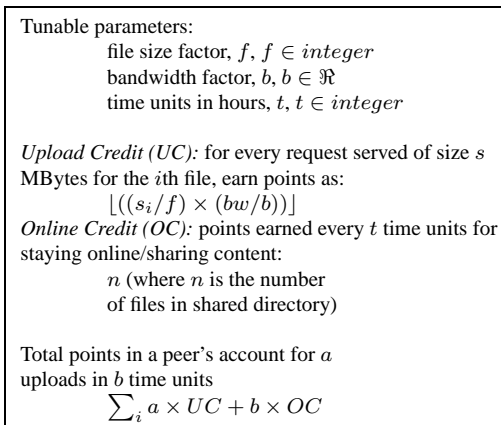


Fig. 2. Basic Accounting Algorithm

Some existing Gnutella like p2p protocols facilitate parallel downloads. Above formula for computing the number of points a peer has earned can be used without any change

for such p2p systems. It will give credit equivalent to sharing one file to each serving peer, but only according to the size of the download it facilitated.

2) *Enhanced Accounting with Individual File Popularities*: Measurement study of Gnutella queries ([4]) found that the popularity of search strings follows a Zipf-like distribution. Study [5] of availability and locality measurements of p2p file systems noted that most popular 10% of the transferred files account for 60% of the total files transferred; 10% of most popular stored files account for 50% of the total files stored. Further, the authors observe that 4MB was the most popular file size. This implies that on while some files are rarely accessed, most popular files are accessed very heavily. Although the basic point system described in III-A.1 recognizes the presence of *hard-to-find content*, it does not take into account the exact file popularity distribution. Hence, it may not produce the best results in terms of capturing the participation in the system. If a p2p system provides dynamic file popularity updates, this enhanced point system can be used. Secure functioning of the accounting component involves periodic computations by the EAA. As described in section IV, the EAA can facilitate such dynamic popularity updates. The basic formula for total point computation remains the same as before, only the upload credit (UC) component changes. Figure III-A.2 shows the changed component to be used in the formula in figure III-A.1. The file popularities are shown as a function $Func(pop)$ in the figure. It has been shown in [6] that for replication purposes, square root popularity produces the optimal results. Hence, we conjecture that incorporating the square root of file popularity as a divisor in the computation of points for a peer would produce optimal results. However, the exact function remains an area of future investigation.

Upload Credit (UC): for every request served of size s MBytes for the i th file, earn points as:

$$\lfloor (s_i / (f \times Func(pop_i)) \times (bw/b)) \rfloor$$

Fig. 3. Enhanced Accounting with Individual File Popularities

B. Award Component

While the accounting component ensures every enrolled peer's points are updated according to its level of contribution, the award component which runs on each peer's software uses those points to decide on the LoS other peers are eligible for. When a peer performs a search for content or requests retrieval of content, her points are sent along with the request. In the case of peers new to the system or the peers who are not enrolled in the program, these points will be zero, corresponding to the basic LoS. For others peers the points will be positive. The peers that process the query and the peer that serves the content provide a LoS to the requester based on the points it has earned in the system. An

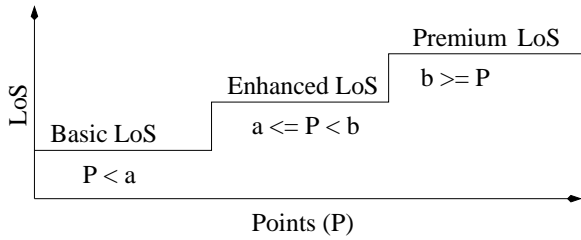


Fig. 4. Levels of Service in a P2P Network

example of the parameters that can be mapped to various LoSs are: *number of hops* a peer's query is allowed to go in search of the content (H), *type of scheduling* the serving peer uses in serving the content (S), and *rate* at which the peer serving the content is willing to transfer the document (R). Parameter H is specific to p2p networks similar to Gnutella and may be different for other p2p networks. The mapping of number of hops H to a LoS could be independent of the number of peers enrolled in the program, however the mappings of rate R , and scheduling S might have to be changed for the system as the enrollment in the frequent-sharer program changes. This can be done by monitoring the change in enrollment observed by individual peers.

In its basic form, the LoS can have three types of services, based on particular combination of values for H , S , and R . These levels are referred to as *basic* LoS, *enhanced* LoS, and *premium* LoS. Basic LoS is the service rendered by the system to the new, or uncooperative peers and to peers who do not enroll in the program. It corresponds to limiting the H , S , and R parameters by the peers when they have enhanced or premium customers in the system. We define the enhanced LoS to be for peers that maintain points P defined by $a \leq P < b$. Point more than that earn a peer premium service from the system and less than that only qualify a peer for basic service. a and b are positive valued tunable system parameters. Figure III-B shows the various LoSs.

IV. SECURITY ISSUES IN ACCOUNTING COMPONENT

When we described the accounting component, we assumed that the counter maintained by the accounting component that keeps the most recent points for each peer is kept and updated in the enrolled peer's software itself. P2p server software may be easily modified by malicious peers and as a result they can run a version of the software that thwarts the point computation algorithm.

The proposed security mechanism is based on the concept of *reputation systems*. Points maintained by the accounting component in essence are credentials that comprise each peer's reputation, upon which the award component makes decisions about the LoS. *eBay* and *Slashdot* are examples of existing centralized reputation systems. Freehaven project [7] and NICE project [8] use a similar concept

for their systems.

We assume that each peer interested in joining the frequent-sharer program obtains a (public, private) key pair from the enrollment and aggregation agent (EAA). The EAA could be replicated for efficiency and availability reasons. To avoid security issues arising out of peers obtaining multiple identities for the sake of earning more points in the system, we propose using human intervention of the type proposed in [9] while obtaining the (public, private) key pair. Further, we assume that each peer also has access to EAA's public key. The assumption is that the EAA is not malicious but peers can be malicious and can collude with other peers.

Let us denote the public and private keys of the requester peers by PK_r and SK_r , and those of the senders by PK_s and SK_s . The following exchange takes place between the requester peer and the sender peer at the time of file download:

- After the sender sends the file, the requester sends a *requester portion of the receipt (RPR)* $\{requester_identity, file_name, file_size, time_stamp, other_info\}_{SK_r}$ to the sender peer. *other_info* might be the file popularity, if the enhanced accounting with individual file popularities is used.
- The sender peer verifies the information using the requester's public key and stores $\{\{requester_identity, file_name, file_size, time_stamp, other_info\}_{SK_r}, \{sender_identity, sender_bandwidth\}\}_{SK_s}$ as a *receipt* of the transaction. The receipt is shown in figure 5.

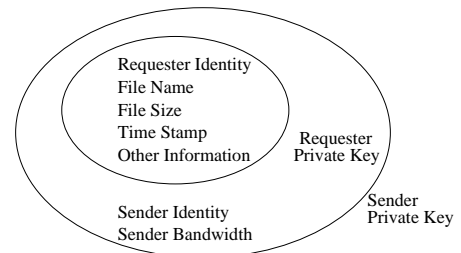


Fig. 5. Receipt of the Transaction

If any one of the peers involved in the transaction is not enrolled in the frequent-sharer program, above receipt exchange does not take place. This is to ensure that peers who do not enroll in the program do not incur the overhead. In that case, if the sender is enrolled in the program, it would not be able to get credit for serving the file. This could potentially serve as a motivation for the senders to prefer transactions only with other enrolled participants.

Above receipt generation ensures that the senders do not get credit without serving the file. This is because the requester peer does not give the receipt before getting the file. Since the sender gets credits for serving the file, it has no reason to drop the receipt. If the receiver does not send the

RPR, the sender can report to the EAA. Beyond a threshold number of such complaints the requester can be blacklisted.

If enrolled in the program, periodically the sender peers send these receipts to the EAA and they are translated into credits to each of their points by the EAA. The credit computation is in accordance with either of the point computation algorithms described in sections III-A.1 and III-A.2, taking into account factors like the file size, popularity, and sender's bandwidth. The frequency of these updates would depend on the balance between the overhead for the EAA, the sender peer, the network, and that of keeping sender accounts up-to-date.

After processing the receipts, the EAA sends an encrypted *credential* $\{EAA_identity, time_stamp, sender_points, sender_identity\}_{SK_{EAA}}$ to the sender peer. We expect the credential to contain points in fixed denominations. Left over odd denominations of points can be stored with the EAA to be incorporated in the future credentials. The time stamp in the credentials serves to expire the points. The expiration duration is expected to be a system parameter. When the requesters send their points during search and retrieval phases, the expired points do not carry any value. The motivation behind expiration of points is to prevent the peers who have once earned good credit in the system to never contribute after that and still receive better LoS. The EAA does not store the accounts for any peer. Thus, peers keep their latest points for transaction purposes and do not have to contact the EAA often. Since the credentials are encrypted, tempering with the points is not a possibility. Another important security concern addressed by EAA's processing of the receipts is that the EAA can run some simple algorithms to detect collusion among peers to increase their points in the system. This can be done by giving fewer credits if the same set of peers have many transactions with each other.

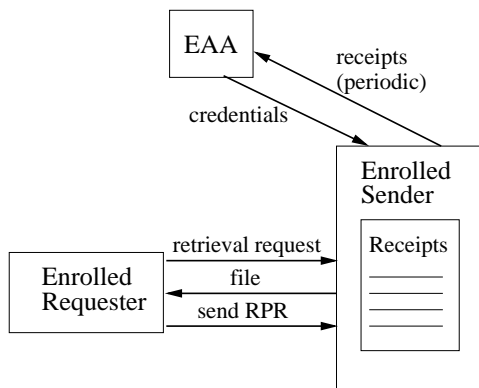


Fig. 6. Communication Induced by the Accounting Component.

Figure 6 shows the communication between the accounting component of two communicating peers and that between the sender peer and the EAA. The enrolled requester peer sends the receiver portion of the receipt (RPR) after

receiving the file. The enrolled sender peer verifies the RPR and stores the receipt. At some later time, the enrolled sender obtains credentials in fixed denominations, from the EAA, as points.

The accounting component also gives credit to peers for staying online, based on the number of files they are sharing. This will be done by periodic monitoring by the EAA. The exact frequency of such monitoring remains to be investigated. Just as the EAA sends encrypted credentials to relevant peers after processing receipts, the credits earned by each peer after the monitoring can be sent by the EAA to the peers.

The enhanced point computation algorithm described in III-A.2 requires an information about file popularities. One possible way to get such information would be for the EAA to do such computations since it has information about which files were accessed from processing the receipts. Local views of peers about file popularities may not give accurate information. The EAA can send such updates to peers when it sends them the processed credentials.

V. CONCLUSION

We have proposed a frequent-sharer program, based on a point system to increase participation in decentralized unstructured p2p systems. It securely tracks each peer's contribution to the system and provides a level of service commensurate with the participation. Currently, we are working on the performance evaluation of the point system.

REFERENCES

- [1] "Gnutella home page," <http://www.gnutella.com/>.
- [2] Eytan Adar and Bernardo A. Huberman, "Free riding on Gnutella," Tech. Rep., Xerox PARC, 2000.
- [3] Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble, "A measurement study of peer-to-peer file sharing systems," in *Proceedings of Multimedia Computing and Networking*, Jan. 2002.
- [4] Kunwadee Sripanidkulchai, "The popularity of Gnutella queries and its implications on scalability," White Paper Featured on O'Reilly's website <http://www.openp2p.com/>, Feb. 2001.
- [5] Jacky Chu, Kevin Labonte, and Brian Neil Levine, "Availability and locality measurements of peer-to-peer file systems," in *ITCom: Scalability and Traffic Control in IP Networks*, July 2002, vol. 4868 of *Proceedings of SPIE*, Proceedings of SPIE.
- [6] Mostafa H. Ammar and Jia W. Wang, "On the optimality of cyclic transmission in teletext systems," *IEEE Transactions on Communications*, vol. 35, no. 1, pp. 68–73, Jan. 1987.
- [7] Roger Dingledine, Michael J. Freedman, and David Molnar, *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, chapter 12: Free Haven, O'Reilly, Mar. 2001.
- [8] Seungjoon Lee, Rob Sherwood, and Bobby Bhattacharjee, "Cooperative peer groups in NICE," To Appear in INFOCOM, Apr. 2003.
- [9] Jun Xu, Richard Lipton, Irfan Essa, and Min-Ho Sung, "Mandatory human participation: A new scheme for building secure systems," To Appear in IEEE Network, 2002.