Editor: Munindar P. Singh • singh@ncsu.edu
Shengru Tu • shengru@cs.uno.edu

Free Riding in Peer-to-Peer Networks

Murat Karakaya, Ibrahim Korpeoglu, and Özgür Ulusoy • *Bilkent University*

Free riding in peer-to-peer (P2P) networks poses a serious threat to their proper operation. Here, the authors present a variety of approaches developed to overcome this problem. They introduce several unique aspects of P2P networks and discuss free riding's effects on P2P services. They categorize proposed solutions and describe each category's important features and implementation issues together with some sample solutions. They also discuss open issues, including common attacks and security considerations.

In a peer-to-peer (P2P) network, peers are expected to contribute to the system by sharing their resources in return for using the network and other peers' resources. However, in many P2P networks, a considerable portion of peers are reluctant to share resources. Thus, P2P networks' primary expectation — peers' implicit or explicit functional cooperation and resource contribution — might fail, leading to a situation called *free riding*. In a P2P context, a free rider is a peer that uses P2P network services but doesn't contribute to the network or other peers at an acceptable level.

Eytan Adar and Bernardo A. Huberman performed the first study specifically addressing the free-riding problem.[1] The authors reported that 70 percent of peers didn't share any files at all, and 25 percent provided 99 percent of all query hits in the network. Since then, many studies have verified that high degrees of free riding exist in P2P networks. As these works show, for some types of P2P networks (such as file sharing), a few altruistic peers can provide the requested services and might help the system survive. However, not all P2P networks (such as multimedia streaming and storage sharing networks) will have enough voluntary or altruistic peers to achieve the desired level of service. So, eliminating or reducing free riding's impact has become a topic of considerable research. Here, we examine the free-riding problem in P2P networks and elaborate on various proposed solutions.

P2P Network Types

Free riding's impact and the effectiveness of a solution to it depend on the P2P network in question, so we should first examine various types of networks that exist. We can classify such networks according to many criteria. One possible classification is based on two network features: the *degree of centralization* and the *degree of structure*. The former determines to what extent a P2P network relies on servers to assist interactions between peers, whereas the latter refers to the way in which content is indexed and located in the network. Using these two criteria, we can classify P2P networks into three types: *centralized*, *decentralized but structured* (hybrid), and *decentralized and unstructured* (pure). Centralized P2P networks maintain a central directory that is constantly updated; peers use this directory to locate resources (as with Napster). Hybrid P2P networks don't have a central directory but are structured — that is, hybrid P2P networks firmly control the P2P topology and systematically place file indices into peers, following a certain algorithm (as with Chord, the Content-Addressable Network, or Pastry). In this way, the network can efficiently resolve queries. Pure P2P networks have no centralized directory, and these networks have little control over the network topology (as with Gnutella or KaZaa).

Another possible classification regards the type of services a P2P network provides. *File sharing* is the most widespread P2P service and lets peers search and download files from oth-

1089-7801/09/$25.00 © 2009 IEEE

er peers connected to the network (as with Napster, Gnutella, Publius, Free Haven, or BitTorrent). *Distributed computing*, on the other hand, lets peers aggregate their computing power to solve a large and computationally intensive problem. SETI@home, Avaki, and Entropia are well-known examples in this category. P2P *storage services* provide virtual stable storage via redundancy and aim to allow peers to continuously access files while preserving author anonymity; OceanStore, PAST (a global, persistent storage utility), and FreeNet are examples of such systems. *Collaborative* P2P applications enable application-level collaboration among peers. These applications (such as NetMeeting, Magi, Groove, Jabber, and DOOM) include instant messaging, chats, and online games. P2P *platforms* (for example, JXTA) support common P2P services, such as naming, discovery, communication, security, and resource aggregation. Finally, P2P *multimedia streaming services* (for example, PPLive, UUSee, Peercast, and Freecast) let peers stream and multicast or broadcast audio and video to each other.

Free Riding's Impact

Free riding can adversely affect a P2P network's operation. In a free-riding environment, a small number of peers serve a large population, which can lead to scalability and single-point-of-failure problems. Free riders and their queries generate substantial network traffic, which might lead to service degradation. Furthermore, free riders occupy considerable underlying available network capacity and resources, causing delays and congestion for non-P2P traffic.

How serious free riding's effects are on a P2P network depends on many factors, including the network type and its requirements. In a file-sharing P2P network, for example, if most peers prefer not to share, renewal or presentation of interesting content might decrease with time; thus, the number of shared files can become limited or only increase slowly. The search process's quality might degrade due to an increasing number of free riders in the search horizon. With time, honest peers who have contributed many files to the network might begin to find fewer and fewer worthwhile files themselves and leave the system altogether, taking their files with them.

In a P2P CPU-sharing grid (a distributed computing system), free riding can easily decrease system utility or even make the system collapse due to insufficient CPU resources. Similarly, in media-streaming systems, peers gain utility not only from file availability but also from high-quality file streams.[2] Although a conventional file-sharing system might persist even with low-level cooperation, a P2P streaming system can't offer high streaming quality to its users if only a relatively few users cooperate. Even though the network isn't heavily congested, if cooperation is low, streaming quality will be poor.[2] Similarly, applications developed to run over P2P platforms or services that collaboration networks provide fail to execute properly if the network doesn't achieve the required cooperation level.

Structured P2P networks can be more vulnerable to some types of free riding than unstructured ones. In a structured P2P network that uses the CAN protocol, for example, peers must store key-value pairs for keys that fall into their zone. Within the CAN context, peers can free ride by not storing key-value pairs in their zones and by ignoring incoming queries. If most peers do this, CAN can't resolve most queries, and the network might easily fall apart.

Ultimately, free riding's effects can range from simply annoying users to crashing the whole system. So, P2P system developers should shape solutions to deal with free riding according to the expected impact it will have.

Methods to Combat Free Riding

Although cooperation is key to many P2P networks' existence and success, realizing it is difficult without effective mechanisms. To address this requirement, researchers have proposed several approaches to make P2P networks "contribution-aware" and thus combat free riding. We can categorize these approaches into three main groups: *monetary-*, *reciprocity-*, and *reputation-based approaches.*

Monetary-Based Approaches

Monetary-based approaches charge peers for the services they receive. Because these services are still very low cost, such approaches are also called micropayment-based solutions. Any monetary-based system requires two key mechanisms: an accounting module to securely store each peer's virtual currency and a settlement module to fairly exchange virtual currency for services. Most monetary-based systems implement these components by centralizing their functions within a single authority, which manages each peer's balance and transactions by tracking accounts and distributing and cashing virtual currency. Most of the proposed solutions depend on a public-key infrastructure (PKI) to provide security against fraud and errors. We discuss security concerns more in a later section.

When monetary-based solutions deal with only small payments, any incorporated security mechanisms need only be lightweight.[3] Most monetary-based solutions don't guarantee a totally fair exchange of goods and payment, however,[3] because tight security services can make transactions more expensive (in terms of complexity, computation, and communications) than the exchanged goods' value. Thus, effective monetary-based systems simply require "good enough" security, in which fraud is detectable, traceable, and unprofitable, while preserving high efficiency.

P2P networks can implement monetary-based approaches using two different payment methods: online and offline. In online payment methods, virtual currency exchange occurs at the same time as peers receive services. This solution can prevent most payment frauds. To apply this method, the central authority must be online at the moment of transaction.

Conversely, with offline payment methods, P2P networks can execute payment after services are exchanged whenever the central authority is available. This might require peers to use permanent identification. Furthermore, because payments occur offline, a P2P network might not discover coin fraud (using a counterfeit currency) until after the fact. Still, P2P network developers might prefer offline payment methods from a practical standpoint because use of offline payment methods involves lower communication and computational costs and lower latency. Researchers have proposed various monetary-based approaches in the context of P2P networks, such as PPAY,[3] and online and offline Karma.[4]

Monetary-based approaches have several implementation limitations when applied to P2P networks.

Centralization and communication overhead. All proposed solutions require some centralized authority to monitor each peer's balance and transactions. This can cause scalability and single-point-of-failure problems, as mentioned. Additionally, disseminating virtual currencies, managing transactions, and applying auditing mechanisms increase communication overhead in the network.

Persistent identifiers. To store peer balances and manage transactions, monetary-based approaches require persistent user identifiers. Providing such identifiers, however, is complicated by peers' anonymity, wide dispersion, and ease of identity modification in most unstructured and decentralized P2P networks. We discuss this issue more in a later section.

Mental transaction costs. Users dislike micropayments primarily because they must decide before each service request whether the service is worth a few cents, which leads to confusion and mental decision costs. Thus, monetary-based solutions involve users' mental effort in exchange for inexpensive resources.

Reciprocity-Based Approaches

In reciprocity-based approaches, a peer monitors other peers' behaviors and evaluates their contribution levels. We can define a peer's contribution level as a numerical assessment of that peer's contribution to the P2P network or to the assessing peer. These approaches usually involve a mutual action: the service quality a peer receives is determined by that peer's contribution level. Reciprocity-based approaches usually measure other peers' contributions only for the current session. These approaches maintain no long-term history about peers, which lets a peer that's judged as a free rider in one session be judged as a contributor in the next, provided the peer has changed its behavior. These approaches also let the system preserve peer anonymity because using persistent identifiers isn't mandatory.

Reciprocity-based approaches can be based on *mutual reciprocity* or *indirect reciprocity*. With mutual reciprocity, a peer decides how to serve another peer based solely on the direct service exchange it had with this peer. In contrast, with indirect reciprocity, this decision depends on the level of services a peer provides to the whole network.

Some existing P2P applications, such as BitTorrent, implement the mutual reciprocity-based approach by adjusting a peer's download speed according to its upload speed (requiring downloading and uploading peers to exchange file fragments). Other examples of mutual reciprocity-based approaches are FairNET,[5] the P2P Connection Management Protocol,[6] and eMule (www.emule-project.net), which uses a cross-credit system. When peer A uploads a file to peer B, peer A gets a credit from peer B, which will then privilege peer A in case it wants to download a file from peer B in the future.

Indirect reciprocity-based approaches consider peers' overall contribution to the network when differentiating service provisioning. For example, we suggest a distributed monitoring and punishment scheme elsewhere[7] in which each peer monitors its neighbors' contribution to the network. Depending on the contribution level, the peer applies an action to each neighbor. GNUnet also falls into this category: in GNUnet (http://gnunet.org), peers monitor each others' behavior with respect to resource usage. Peers that contribute to the network receive better service.

Reciprocity-based approaches face several implementation issues.

Fake services. To gain higher levels of contribution, peers can publish fake services.

Contribution-level credibility. Some proposed reciprocity-based methods depend on accurate information about peers, but peers themselves provide this information. A malicious user can cheat a P2P network that depends on such an approach by hacking the client program.

Peer identity management. Peers

are linked to their values through their identities. However, if newcomers are assigned a higher standard utility value than are free riders, a free rider can try to get rid of its degraded value by constantly getting a new identity.

Reputation-Based Approaches

Reputation-based approaches construct and maintain reputation information about peers, and peers with good reputations are offered better services. These approaches can construct reputation information about a peer on the basis of feedback from peers who have interacted with that peer. Such feedback can be positive, negative, or both. The system uses the feedback to build up a good reputation for contributing peers and a bad reputation for free riders.

A peer's reputation information corresponds to its long-term behavior, so reputation-based approaches store and manage long-term peer histories. This implies that it isn't easy to convert a bad reputation to a good one or vice versa.

We can categorize reputation-based methods into two main groups: *autonomous reputation* approaches and *global reputation* approaches.

In an autonomous reputation scheme, peers maintain reputation information about other peers they've interacted with. These local reputation values aren't disseminated or merged to create a global reputation database. Consequently, such approaches are relatively simple to implement because they don't call for a security infrastructure or centralized storage to protect local reputations' integrity. XRep is an example of an autonomous reputation system.[8]

Global reputation-based approaches aggregate the reputation information obtained from several peers or all peers. Such approaches store this consolidated reputation information either at a central location or with a set of peers in the network. Various methods exist for distributing and accessing reputation information in the network. One method is to piggyback reputation values onto P2P protocol messages. This is called a *gossip mechanism*: the receiving peer decides whether to use such gossip to create a consolidated reputation value for a given peer. Unlike gossip mechanisms, an *explicit mechanism* lets a peer retrieve others' reputations from the system through a P2P protocol.

Global reputation approaches speed up free rider identification, especially when the peer population is large and the chance of direct interaction with the same peer is low, because peers can learn from others' interactions. These approaches also provide more reliable and long-term reputation information about peers. EigenTrust[9] is an example of a global reputation system. In EigenTrust, in addition to local values stored at each peer, the global reputation derived from multiple local values is stored at random peers.

We must consider several important issues when implementing reputation-based solutions.

Reputation reliability. Reputation systems assume that peers report their interactions with other peers honestly and impartially. However, a peer can cheat the system and cost other peers by misreporting the services it receives from them.

Centralization and communication overhead. A global reputation system might rely on a centralized authority to store and manage reputation information. This is difficult to implement in pure P2P networks and can also cause scalability problems. Moreover, to exchange and consolidate reputation information, peers must communicate with each other, a centralized authority, or a special group of peers, which increases control traffic in the P2P network.

Persistent identifiers. Because peers' identities should be preserved across sessions to store long-term histories, implementing reputation-based approaches in anonymous systems is difficult. Similarly, preserving anonymity in systems implementing reputation-based approaches is difficult.

> **Reputation systems assume that peers report their interactions honestly. A peer can cheat the system and cost other peers by misreporting the services it receives from them.**

Open Issues

Many researchers have developed appropriate solutions to the free-riding problem. However, certain issues still remain.

Common Attacks or Cheats

Free riders might try to work around free-riding solutions, if doing so increases their benefits from the system. Researchers designing mechanisms to combat free riding should consider the nature of these attacks.[9–11] Let's look at some common attacks against free-riding solutions.

Collusion. We can define collusion as collaborative activity of a group of free riders that gives group members benefits they couldn't gain as individuals. A group of malicious peers can attempt to collectively challenge and fool free-riding mechanisms. For instance, a group of free riders can collude to promote one or more free-riding peers in the group or to damage a contributing peer's reputation. Thus, they can evade detection by exploiting feedback mechanisms.

Modifying virtual currency, contribution, or reputation value. A cheater might exaggerate its virtual currency, contribution, or reputation value by providing incorrect information about itself. Cheaters can do this by, for example, modifying client programs, cracking locally saved values, or modifying a transaction record. To prevent such attacks, systems can implement credit and transaction-record audit mechanisms via P2P protocols, as proposed elsewhere.[11] Such mechanisms can detect when a peer modifies its value and reverse the action. Other solutions are to use a voting scheme to collect opinions about a peer, implement heuristics to find groups of potentially malicious voters, or apply a distributed cryptographic infrastructure, such as a PKI.

Fake services and new content. Many solutions proposed for free riding ignore the issue of fake services, content cheating, or the lack of new content provisioning. For example, in a file-sharing P2P network, a free riding peer can share some files with fake filenames that resemble more popular ones. If other peers download these files, the free riding peer's contribution value might increase. To deal with this problem, some existing P2P clients, such as eMule and eDonkey, use peer interaction to verify service quality and identify fake services. However, these applications experience low peer participation in service evaluation. Similarly, new content is essential for any P2P file-sharing system's popularity and longevity. So, solutions should also aim to promote or enforce new content contribution from peers.

Whitewashing. In most current P2P networks, a peer's real-world identification isn't bound to its online identity, and joining the network obtaining an online identity is free. This lets the network grow rapidly because newcomers can easily join the system. Cheaters can use this fact to repeatedly change their online identities and thus have all the advantages and rights of a newcomer. This is called whitewashing. One technique researchers propose for combating whitewashing is to attach a high cost to acquiring new identities for all newcomers — for example, using proof-of-work protocols.[12] The main idea behind these protocols is that a *prover* demonstrates to a *verifier* that it has expended a certain level of computational effort in a specified time interval. However, as investigated elsewhere,[10] implementing such a solution should be adaptive with respect to newcomer turnover rates and the existing P2P community's contribution level. If the turnover rate is high, this solution imposes a high cost on the P2P community, but if the community's contribution level is high and the newcomer turnover rate is relatively low, the system can tolerate newcomers by imposing a small cost. Another measure to combat whitewashers uses free but irreplaceable pseudonyms for peers, via a trusted central authority to assign them persistent identities. An irreplaceable pseudonym for a peer could be, for example, the unique medium access control (MAC) address for the computer the peer is using.

Because newcomers can easily join the system, cheaters can use this fact to repeatedly change their online identities and thus have all the advantages and rights of a newcomer.

Replay and spoofing attacks. In a replay attack, free riders reuse a message another peer previously sent, which appears to be valid to other peers. Similarly, in spoofing attacks, cheaters construct messages that seem to originate from another peer. If the P2P protocol accepts them as valid, cheaters can benefit — for example, by gaining positive feedback or payments as if they contribute to the network. P2P protocols that assemble message elements simply by concatenating them are prone to such attacks. To prevent them, P2P protocols should validate P2P messages by performing proper checks before they're accepted as legitimate. For instance, peers can sign messages with the session ID and their private keys. However, implementing such a solution requires P2P networks to have a cryptographic infrastructure.

Securing Free-Riding Solutions

A robust and long-term free-riding solution must consider the possibility of the previously mentioned attacks and should incorporate security mechanisms that can successfully deal with them. However, deploying security mechanisms in P2P networks is difficult owing to the P2P paradigm's characteristics, such as anonymity, decentralization, self-organization, and frequent disconnections.

Most security solutions used in global-scale networks require using public keys for authentication, shared secret establishment, or integrity checking, and thus depend on a PKI. In the P2P context, directly implementing PKIs can be troublesome because they require considerable resources to plan, install, deploy, and maintain. Furthermore, the huge number of users and high turnover in P2P networks make key management a challenge in itself. Finally, pure P2P networks don't have any central management, which makes standard PKI implementation based on a certification authority (CA) hierarchy difficult.

One possible alternative to PKI's centralized CA approach is to decentralize the certification process, as with Pretty Good Privacy (PGP), a web-of-trust model in which no central CA exists. Peers can issue certificates and trust to each other in varying degrees. Hence, peers themselves decide the level of trust and the level of security. In PGP, peers can store certifications of other peers that they choose to trust. Then, they can exchange this collection of certifying signatures with each other. Gradually, a decentralized web of trust for public keys and peer identities emerges. However, identifying peers for services that require anonymity is still an issue. Another possible problem in implementing this mechanism in a real P2P environment can be collusion attacks, in which malicious peers inject false certifications.

Other Issues

In addition to the attacks and security issues described, solutions proposed to combat free riding should consider some other issues.

P2P network types and free riding. The type of P2P network can affect the network's ability to implement free-riding solutions. For example, the applicability of widespread authentication and encryption is questionable because security mechanisms require "trusted" endpoints, which don't exist in most unstructured, distributed P2P networks. Thus, P2P network developers should investigate which solutions are effective for which types of networks.

Evaluation methodology. Most mechanisms developed to combat free riding are verified through simulation models or game-theoretic approaches. These tools, however, are limited in analyzing peer behaviors and interactions in a P2P network and involve unrealistic assumptions. So, we must examine the proposed solutions in real environments using real P2P communities. In this way, we can better understand those solutions' impact on P2P networks.

Solutions' side effects. Dealing with free riding is a complex issue in P2P networks in which most peers are free riders. For example, a solution using traditional distributed systems techniques, such as detecting and disconnecting faulty peers from the network, might cause side effects, such as network vulnerability and partitioning.

Free riding in P2P networks creates problems that affect network operation on different levels. Even though existing P2P file-sharing systems might survive despite free riding, P2P network designers should take measures to improve P2P networks' performance and robustness. If this does not occur, performance and widespread use of some P2P networks could become seriously degraded, and these networks might not even survive. Setting up incentives, building reputations, and enforcing reciprocity are key research directions for enhancing contribution and preventing free riding in P2P networks.

Acknowledgments

The Scientific and Technical Research Council of Turkey (TUBITAK) partially supports this work under grants EEEAG-104E028 and EEEAG-105E065.

References

1. E. Adar and B.A. Huberman, "Free Riding on Gnutella," 2000, www.firstmonday.dk/issues/issue5 10/adar.
2. A. Habib and J. Chuang, "Service Differentiated Peer Selection: An Incentive Mechanism for Peer-to-Peer Media Streaming," *IEEE Trans. Multimedia*, vol. 8, no. 3, 2006, pp. 610–621.
3. B. Yang and H. Garcia-Molina, "PPay: Micropayments for Peer-to-Peer Systems," *Proc. 10th ACM Conf. Computer and Comm. Security* (CCS 03), V. Atluri and P. Liu, eds., ACM Press, 2003; http://ilpubs.stanford.edu:8090/757/.
4. F.D. Garcia and J.H. Hoepman, "Off-Line Karma: A Decentralized Currency for Static Peer-to-Peer and Grid Networks," *Proc. 5th Int'l Networking Conf.* (INC 05), S. Furnell, P. Dowland, and G. Kormentazas, eds., 2005, pp. 325–332.
5. E. Buchmann and K. Bohm, "FairNet: How to Counter Free Riding in Peer-to-Peer Data Structures," *Proc. Int'l Conf. Cooperative Information Systems*, Springer, 2004, pp. 337–354.
6. M. Karakaya, I. Körpeoglu, and Ö. Ulusoy, "A Connection Management Protocol for Promoting Cooperation in Peer-to-Peer Networks," *Computer Comm.*, vol. 31, no. 2, 2008, pp. 240–256.
7. M. Karakaya, I. Korpeoglu, and Ö. Ulusoy, "Counteracting Free Riding in Peer-to-Peer Networks," *Computer Networks*, vol. 52, no. 3, 2008, pp. 675–694.
8. E. Damiani et al., "A Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Networks," *Proc. 9th ACM Conf. Computer and Comm. Security*, ACM Press, 2002, pp. 207–216.
9. S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Net-