



## A dynamic game theory approach to solve the free riding problem in the peer-to-peer networks

J-S Hua, S-M Huang, D C Yen & C-W Chena

To cite this article: J-S Hua, S-M Huang, D C Yen & C-W Chena (2012) A dynamic game theory approach to solve the free riding problem in the peer-to-peer networks, Journal of Simulation, 6:1, 43-55, DOI: [10.1057/jos.2011.11](https://doi.org/10.1057/jos.2011.11)

To link to this article: <https://doi.org/10.1057/jos.2011.11>



Published online: 19 Dec 2017.



Submit your article to this journal [↗](#)



Article views: 53



View related articles [↗](#)



# A dynamic game theory approach to solve the free riding problem in the peer-to-peer networks

J-S Hua<sup>1</sup>, S-M Huang<sup>1</sup>, DC Yen<sup>2\*</sup> and C-W Chena<sup>1</sup>

<sup>1</sup>National Chung-Cheng University, Chia-Yi, Taiwan, ROC; and <sup>2</sup>Miami University, Oxford, OH, USA

The peer-to-peer (P2P) system has prevailed on the Internet. The old client-server architecture is gradually becoming unable to satisfy users' various requests and deal with the heavy loading of networks. Many applications adopt P2P architecture to provide more flexible and efficient service. However, P2P systems are not flawless. The anarchic property of P2P systems keeps participant behaviour uncontrolled. Thus, the behaviour of free riding occurs. This paper uses game theory to explore what results from free riding, designs a co-opetition framework to solve the free riding problem, and also proposes a fairness index to monitor the efficiency of the whole system. Simulations are used to test co-opetition under different environments.

*Journal of Simulation* (2012) 6, 43–55. doi:10.1057/jos.2011.11; published online 12 August 2011

**Keywords:** game theory; simulation; co-opetition; peer-to-peer (P2P) system

## 1. Introduction

Peer-to-peer (P2P) systems are comprised of voluntary peers that contribute their own resources to the resources of others. P2P systems are flexibly distributed systems, and users can play roles of server or client depending on whether they provide or request resources (Singh, 2001). P2P systems can form the P2P network and provide a scalable, efficient, and robust platform to exchange distributed resources, such as computing cycles, storages, and data (Bo and Li, 2003). Currently, P2P architecture is applied to many domains, including file sharing, distributed object location, and information retrieval. The myriad features of P2P systems have attracted many network users. The popularity of P2P systems helps explain why the majority of network bandwidth is used by such systems (Betts, 2005). Participants in the P2P network cooperate with each other and are benefited from this cooperation that requires computing resources and degrades computing performance (Feldman *et al.*, 2004). Participants have disincentives to cooperate because they are rational and strategic. The result is that they will only download resources from others, but share none. This is the phenomena of 'free riding' (Krishnan *et al.*, 2004). Previous research shows that almost 60% of nodes in the Gnutella system, a P2P file sharing system, never share their resources (Adar and Huberman, 2000). There are many studies exploring how to solve free riding problem. Recently, computer scientists have used game theory to solve free riding in a behavioural way (Zhu *et al.*, 2009; Matsuda *et al.*, 2010). They investigated how the interaction of participants

influences their payoffs. Previous studies used the dynamic and complete information game model to solve the free riding issue in the P2P network (Gupta and Somani, 2005). However, it is impossible for all participants to get information about what others do or obtain information about the P2P network. The P2P network is more like a dynamic and incomplete information game than a complete one. Therefore, it is necessary to develop a dynamic and incomplete game to analyse P2P systems.

This paper introduces a dynamic and incomplete game model to explain the interaction of peers in the P2P network. According to the analysis of the game, this study explored some of the causes of free riding and proposed a co-opetition framework to control free riding. The goal of this study is to ease free riding and to improve efficiency of the P2P system as a whole. The authors use simulation to test the co-opetition framework under different environments and make comparisons with other mechanisms.

## 2. Literature review

Since the emergence of Napster, P2P networking has prevailed. P2P networking is more like a concept, hence there are only a few general definitions of a P2P network (Stoica *et al.*, 2001). According to the system architecture, the P2P network can be categorized into centralized, decentralized and structured, and decentralized and unstructured P2P (Bo and Li, 2003). Until now, the P2P network has applied to many domains, including distributed computing, file sharing, and collaborative application, etc (Singh, 2001; Wang, 2011).

Game theory is used to study the strategies selection that occurs between multiple people (Rasmusen, 2006). Before

\*Correspondence: DC Yen, Department of DSC & MIS, Miami University, Oxford, Ohio 45056, USA.  
E-mail: yendc@muohio.edu

each player selects their strategy, they take their own payoff functions and others' strategies into consideration. Players select their best strategies to achieve the Nash equilibrium. If every player has selected a strategy and no one can benefit by only changing his strategy, then the current strategy set is said to have achieved Nash equilibrium (Owen, 1995).

The P2P network is formed by peers offering their resources. Each peer has access to the resources in the P2P network (Non-Exclusive). The benefit of downloading resources from other peers is not reduced when other peers download the same resources (Non-Rival). The resources can be considered public goods in the P2P network, because they have the properties of non-exclusive and non-rival (Krishnan *et al*, 2004). The public goods theory dictates that every network peer concerns himself with his profit potential exclusively, implying that everyone would take the option of free riding. However, there are many institutions offering resources to charities voluntarily in the real world. This is because of the concept of fairness. Free riders aside, there are people who pay much more attention to fairness—even more so than profit. If they get more profit because of unfairness, they would not be satisfied (Fehr and Schmidt, 1999). Research indicates that there is reciprocity between people. This means that if someone is friendly towards you, you will be friendly to him and vice versa. Fehr and Schmidt (1999) also indicate that when exchanging public goods, if there is no punishment mechanism, few free riders will cause all people to free ride; if there is a punishment mechanism, most people would rather take some cost to punish free riders in order to sustain the fairness.

There are many studies focusing on free riding (Xu *et al*, 2009; Wang *et al*, 2010). Most of these studies adopt one of the two incentives mechanisms to work—reputation and the virtual currency mechanism (Golle *et al*, 2001; Yang *et al*, 2010; Tseng and Chen, 2011). Some studies use game theory as the analysis tool to solve free riding. Golle *et al* (2001), Krishnan *et al* (2004), and Wang *et al* (2005) adopted static game theory to analyse and ease free riding. Feldman *et al* (2004) used the Generic Prisoner Dilemma (GPD) to model the interactions between peers, and provided some possible solutions to solve free riding and cheating in the P2P network. Gupta and Somani (2005) built a dynamic and complete game model and used the reputation level to decide the probability of receiving service for lessening free riding.

### 3. System architecture

The concepts of P2P systems are the same even though there are a number of different types of P2P networks. This paper focuses the research domain on decentralized and unstructured P2P networks because this type of P2P network is the most difficult one to control. The authors chose file sharing

systems as the research subject because it is one of the important applications of P2P networks (Betts, 2005). Most previous studies assume that users in the P2P network are homogeneous. However, according to the responses to resource requests, users can be categorized into three types: altruist, free rider, and somewhere between altruist and free rider. Altruists are glad to offer resources when facing resource requests; free riders do not offer any resources when facing resource requests; the third type of user sometimes offers resources, and sometimes does not. This paper operates under the assumption that there are only altruists and free riders in P2P networks, because the equilibrium in this condition would be the lower bound of the equilibrium in the real world.

#### 3.1. Game model of P2P file sharing system

According to the assumptions of game theory, this study first assumes users in P2P network are rational. This means that every user would pursue their maximum profit. Since game theory is used to explore the interaction of users in every period, the infinite lifespan in P2P network would be cut into  $t$  discrete periods that  $t = \{0, 1, 2, \dots, \infty\}$ . This study secondly assumes peers would receive resource requests from other peers or send resource requests to other peers in every period and then pursue the best strategy depending on their individual profit goal. Third, this study also assumes every peer owns some resources other peers wanted, and therefore any peer could get the resources they wanted from one of other peers. All of altruistic and free-riding peers have the same strategy set {Upload Resources, Not Upload Resources} when they face resource requests; however, they have different payoff functions.

Provided that when peers receive resources (the quantity is  $Q_d$ ), they would get profit,  $f_1(Q_d)$ .  $f_1(\cdot)$  is a monotonously increasing function. Therefore, the more resources peers receive, the more profit peers get. When peers upload resources (the quantity is  $Q_u$ ), it costs them  $g_1(Q_u)$ .  $g_1(\cdot)$  is a monotonously increasing function. Hence, the more resources peers upload, the more cost in resources peers incur. However, altruism on the Internet exists because helpers want to earn additional profit from social interaction, such as personal reputation, self-esteem, anticipated reciprocity, etc (Wallace, 2001). Therefore, when altruists upload resources to others, they will also get an extra profit,  $f_2(Q_u)$ .  $f_2(\cdot)$  also represents a monotonously increasing function. Besides, altruists would like to upload resources others request only when  $f_2'(\cdot) - g_1'(\cdot) > 0$ .

According to the second assumption above, peers would upload resources after receiving resource requests from other peers or download resources after sending resource requests to other peers in every period. Furthermore, peers would earn profit or incur cost when downloading resources they want or uploading resources others have requested. Therefore, the payoff for peers in every period depends on whether

they have chosen to upload or download. The payoff function for altruists is  $f_1(Q_d) + f_2(Q_u) - g_1(Q_u)$ ; the payoff function for free riders is  $f_1(Q_d) - g_1(Q_u)$ . According to user type and payoff function, the game models and Nash equilibrium can be represented as follows. Superscripts are used to present the parameter belonging to each peer. For instance,  $Q_d^1$  is the quantity of resources downloaded by peer 1. By Folk Theorem, if there is a strategy set such as the Nash equilibrium strategy profile of some period of a game, this strategy set is also the Nash equilibrium strategy profile when the game continues to the infinite. Therefore, finding the Nash equilibrium of an infinite game can be simplified to finding the Nash equilibrium of some period of the game (Rasmusen, 2006). The game models of the P2P network are incomplete information games because each peer does not know the payoff functions of others. Consequently, each peer will choose their strategies depending on their own profits. According to these assumptions, the game models can be built as Tables 1–3.

(1) Altruist to altruist

When both users are altruists, it is the best choice for each to select {Upload Resources}. Therefore, Nash equilibrium of this game model is {Upload Resources, Upload Resources}.

(2) Altruist to free rider

When one peer is an altruist, and the other is a free rider, {Upload Resources} is still the best choice for the altruist

to get the maximum profit, and selecting {Not Upload Resources} is better than selecting {Upload Resources} for the free rider.

(3) Free rider to free rider

The last situation is that both peers are free riders. In this situation, it is better to select {Not Upload Resources} than to select {Upload Resources} for both peers. Consequently, the Nash equilibrium of this game model is {Not Upload Resources, Not Upload Resources}.

In other words, whomever the peer faces, it is the best choice to select {Upload Resources} for altruists, and it is the best choice to select {Not Upload Resources} for free riders. For the P2P network as a whole, altruists can increase efficiency, but free riders cannot. Moreover, free riders request resources continuously, but do not offer any resources to others. This will impede the altruists. In conclusion, there should be some restrictions on free riders to offset the problems they cause in P2P networks.

### 3.2. Co-opetition framework

In real life, when people interact with other people over time, the history of past interactions informs them about other's abilities and dispositions. Besides, an expectation that people will consider other's pasts in future interactions constrains behaviour in the present (Resnick *et al.*, 2000). Therefore, it is common to take people's reputations to differentiate the

**Table 1** Game model: altruist to altruist

		Altruist 2	
		Upload resources	Not upload resources
Altruist 1	Upload resources	$(f_1(Q_d^1) + f_2(Q_d^2) - g_1(Q_d^2), f_1(Q_d^2) + f_2(Q_d^1) - g_1(Q_d^1))$	$(f_2(Q_d^2) - g_1(Q_d^2), f_1(Q_d^2))$
	Not upload resources	$(f_1(Q_d^1), f_2(Q_d^1) - g_1(Q_d^1))$	$(0, 0)$

**Table 2** Game model: altruist to free rider

		Free rider 1	
		Upload resources	Not upload resources
Altruist 1	Upload resources	$(f_1(Q_d^1) + f_2(Q_d^2) - g_1(Q_d^2), f_1(Q_d^2) - g_1(Q_d^1))$	$(f_2(Q_d^2) - g_1(Q_d^2), f_1(Q_d^2))$
	Not upload resources	$(f_1(Q_d^1), -g_1(Q_d^1))$	$(0, 0)$

**Table 3** Game model: free rider to free rider

		Free rider 2	
		Upload resources	Not upload resources
Free rider 1	Upload resources	$(f_1(Q_d^1) - g_1(Q_d^2), f_1(Q_d^2) - g_1(Q_d^1))$	$(-g_1(Q_d^2), f_1(Q_d^2))$
	Not upload resources	$(f_1(Q_d^1), -g_1(Q_d^1))$	$(0, 0)$

quality of services they could get. For example, most lines of credit are issued on the reputation rating of the borrowers. A borrower who has a better reputation of paying off debts could apply for a larger loan. This paper used a similar concept to build a co-opetition framework which establish a relationship between upload and download. A peer who would like to usually upload resources others have requested can acquire a better reputation which determines the ratio of resources it can download. By applying the same logic, peers who want to download more resources would frequently upload resources others have requested. The profit from cooperation of peers can therefore be increased. In addition, the co-opetition framework provides a threshold as the basis to eliminate free riders. If the ratio of resources which a peer can get is less than the threshold means that the requesting peer was free riding in the past and then cannot download any resources in the present. Consequently, the efficiency of the whole system can also be improved. According to the concept above, some simple rules are made in the co-opetition framework for solving free riding problem without expending too much computing capacity. Table 4 lists the parameters which are used in the rules of the co-opetition framework.

$P(R_i^{t-1})$  represents the ratio of resources which peer  $i$  can get in period  $t$  and is between 0 and 1. When peers enter a P2P file sharing system, they get initial values of  $P(R)$  from the co-opetition framework.  $P(R)$  is connected to their previous sharing behaviour and the resources uploaded this period to ensure that users upload resources for the opportunity to download resources. Therefore,  $P(R_i^t)$  expressed as Equation (1) is the ratio of resources which peer  $i$  can get in next period  $t+1$  and the weighted average of  $P(R_i^{t-1})$  and  $r_i^t$ .  $P(R_i^{t-1})$  is calculated at the end of last period  $t-1$  according to the previous sharing behaviour of peer  $i$ . The reputation  $r_i^t$  is calculated from resources uploaded by peer  $i$  in this period  $t$ .  $\alpha$  represents the relative

importance between past and present behaviour and is between 0 and 1. The result of this framework is influenced by  $\alpha$  considerably, hence the value of  $\alpha$  should be chosen carefully.  $r_i^t$  stands for the reputation of peer  $i$  in period  $t$  and is estimated by  $L_i^t$  which is the amount of resources uploaded by peer  $i$  in period  $t$ , shown as Equation (2). The value of  $r_i^t$  is between 0 and 1 and can be almost equal to 1 when peer  $i$  uploads more resources. Therefore, users can improve their reputation by uploading resources other users have requested.  $O_i^t$  is the amount of resources requested by peer  $i$  in period  $t$ . Both units of  $L_i^t$  and  $O_i^t$  are megabyte (MB). The amount of resources uploaded by one peer ( $L$ ) is decided according to the amount of resources other peers have requested ( $O$ ). The  $L$  of one peer can further determine its  $P(R)$ . This means that the amount of resources one peer would upload can influence the ratio of its requested resources ( $O$ ) which could be downloaded by this peer in the next period.

$$P(R_i^t) = (1 - \alpha)p(R_i^{t-1}) + \alpha r_i^t \quad (1)$$

$$r_i^t = \frac{L_i^t}{1 + L_i^t} \quad (2)$$

The threshold  $\delta (0 < \delta < 1)$  is the basis to eliminate free riders. When peer  $i$  requests resources from peer  $j$  in period  $t$ , peer  $j$  will get  $O_i^t$  and  $P(R_i^{t-1})$  from peer  $i$ . If  $P(R_i^{t-1})$  is smaller than  $\delta$ , there is no transferring of resources between peer  $i$  and peer  $j$ ; otherwise peer  $j$  will upload  $O_i^t P(R_i^{t-1})$  of resources to peer  $i$ . If peers cannot download resources they requested completely in one period, they will continue to download the rest in next period automatically until requested resources downloading is complete or aborted owing to  $P(R) < \delta$ . When the period  $t$  is over, peer  $i$  and peer  $j$  recalculate their  $P(R)$  and receive updated  $P(R)$  which are  $P(R_i^t)$  and  $P(R_j^t)$ , respectively. In order to avoid reducing the value of  $P(R)$  excessively fast,  $P(R)$  of peer  $i$  will be recalculated only when it interacts with other peers (uploads or downloads resources). It is not necessary to calculate  $r$  of peer  $i$  if its  $P(R)$  is not recalculated. Therefore, peer  $i$  will not acquire poor reputation due to a lack of requests for resources from other peers.

If peer  $i$  only downloads resources it required without uploading resources other peers requested, its  $P(R)$  will continue to reduce as periods go by and then the resources it could download will also keep on decreasing. Users whose  $P(R)$  is smaller than  $\delta$  have to upload resources to raise his  $P(R)$ . If  $\delta$  is too small, it is not effective in eliminating free riders. If  $\delta$  is too large, some users who would like to upload resources sometimes but not every time may no longer have the willingness to cooperate since they must upload more resources for keeping their  $P(R) > \delta$ . Therefore, the value of the threshold  $\delta$  is very important to this framework and should be carefully controlled.

The concepts of reputation and threshold were introduced into the co-opetition framework; hence the original game

**Table 4** The parameters in the co-opetition framework

Parameter	Definition
$P(R_i^{t-1})$	The ratio of resources which peer $i$ can get in period $t$ . $P(R_i) = (1 - \alpha)p(R_i^{t-1}) + \alpha r_i^t$
$\alpha$	Weight, $0 < \alpha < 1$ .
$r_i^t$	The reputation of peer $i$ in period $t$ . $r_i^t = (L_i^t) / (1 + L_i^t)$
$O_i^t$	The amount of resources requested by peer $i$ in period $t$ . (MB)
$L_i^t$	The amount of resources uploaded by peer $i$ in period $t$ . (MB)
$\delta$	Threshold. If $P(R_i^{t-1}) < \delta$ , peer $i$ is not allowed to download resources in period $t$ . If $P(R_i^{t-1}) \geq \delta$ , peer $i$ is allowed to download $O_i P(R_i^{t-1})$ of resources in period $t$ .

models proposed in Section 3.1 should be modified. First of all,  $Q_d$ , the quantity of resources downloaded by every peer, should be changed to  $Q_d P(R)$  for adjusting to achieving the ratio of resources which they can get in one period. Second, before peers decide their best strategies to interact with others, it should be determined whether peers who requested resources were free riding by comparing the values of  $P(R)$  and  $\delta$  from requesting peers. Since  $P(R) < \delta$  means requesting peers were free riding in the past, they will not get the chance to download the resources they have requested. Conversely,  $P(R) > \delta$  means requesting peers have uploaded resources for sharing with others in the past, hence they are allowed the opportunity to download the resources they requested. However, the resource-owning peers still have right to finally decide whether they want to upload the resources requested by other peers. The resource-owning peers will choose best strategies according to the modified game models which introduced the co-opetition framework. Nash equilibriums of the modified game models shown as in Tables 5-7 are the same as the original game models.

When peers join P2P network, the co-opetition framework gives them an initial value of  $P(R)$ ,  $P(R^0)$ . If a peer never uploads resources other peers requested and continuously downloads resources from the first period to the  $n$ th period,

the ratio of resources it can get in  $n+1$  period will be  $P(R_i^n) = (1-\alpha)^n P(R_i^0)$  according to Equation (1). It means that the ratio of resources this peer (a free rider) can get will continue to reduce as periods go by. When  $P(R_i^n) = (1-\alpha)^n P(R_i^0) < \delta$ , a free rider will not be allowed downloading any more resources after the  $n$ th period. Therefore, the number of periods that a free rider could download resources can be derived and expressed as  $n < \log(\delta / P(R_i^0)) / \log(1-\alpha)$ , and depends on the values of  $\delta$ ,  $\alpha$ , and  $P(R^0)$ . Using the ideas of the threshold ( $\delta$ ) and the ratio of resources which peers can get in one period ( $P(R)$ ), the resources free riders can download are limited, and altruists would not bear higher computing cost in uploading resources free riders request. Consequently, the co-opetition framework can ease the free riding problem and improve the efficiency of the P2P file sharing system.

### 3.3. Evaluation index

According to the literature, users expect free riders pay more attention to fairness of the whole P2P network system than their own profit. Therefore, this paper defines the efficiency of a P2P file sharing system as the fairness to the whole system. In order to evaluate whether the co-opetition framework can ease free riding and improve the fairness of

**Table 5** Modified game model: altruist to altruist

		<i>Altruist 2</i>	
		<i>Upload resources</i>	<i>Not upload resources</i>
Altruist 1	Upload resources	$\left( f_1(P(R_1^{t-1})Q_1) + f_2(P(R_2^{t-1})Q_2) - g_1(P(R_2^{t-1})Q_2), \right.$	$\left( f_2(P(R_2^{t-1})Q_2) - g_1(P(R_2^{t-1})Q_2), \right.$
	Not upload resources	$\left. f_1(P(R_2^{t-1})Q_2) + f_2(P(R_1^{t-1})Q_1) - g_1(P(R_1^{t-1})Q_1) \right)$	$\left. f_1(P(R_2^{t-1})Q_2) \right)$
		$(f_1(P(R_1^{t-1})Q_1), f_2(P(R_1^{t-1})Q_1) - g_1(P(R_1^{t-1})Q_1))$	$(0, 0)$

**Table 6** Modified game model: altruist to free rider

		<i>Free Rider 1</i>	
		<i>Upload resources</i>	<i>Not upload resources</i>
Altruist 1	Upload resources	$\left( f_1(P(R_1^{t-1})Q_1) + f_2(P(R_2^{t-1})Q_2) - g_1(P(R_2^{t-1})Q_2), \right.$	$\left( f_2(P(R_2^{t-1})Q_2) - g_1(P(R_2^{t-1})Q_2), \right.$
	Not upload resources	$\left. f_1(P(R_2^{t-1})Q_2) - g_1(P(R_1^{t-1})Q_1) \right)$	$\left. f_1(P(R_2^{t-1})Q_2) \right)$
		$(f_1(P(R_1^{t-1})Q_1), -g_1(P(R_1^{t-1})Q_1))$	$(0, 0)$

**Table 7** Modified game model: free rider to free rider

		<i>Free Rider 2</i>	
		<i>Upload resources</i>	<i>Not upload resources</i>
Free rider 1	Upload resources	$\left( f_1(P(R_1^{t-1})Q_1) - g_1(P(R_2^{t-1})Q_2), \right.$	$\left( -g_1(P(R_2^{t-1})Q_2), f_1(P(R_2^{t-1})Q_2) \right)$
	Not upload resources	$\left. f_1(P(R_2^{t-1})Q_2) - g_1(P(R_1^{t-1})Q_1) \right)$	$(0, 0)$
		$(f_1(P(R_1^{t-1})Q_1), -g_1(P(R_1^{t-1})Q_1))$	

the system, this paper uses (1) the number of requests fulfilled and (2) the ratio of uploading to downloading to evaluate the simulation results.

#### (1) The number of requests fulfilled

For free riders, the authors used the number of requests fulfilled to evaluate the influence of different mechanisms on the free riders. By using this index, this study can show whether the co-opetition framework could ease the free riding problem efficiently.

#### (2) The ratio of uploading to downloading

To test the fairness of the whole P2P file sharing system, this paper uses the ratio of uploading to downloading as the index. Since free riders never upload resources, the values of this index of free riders will be always zero. Therefore, this study applied this index only on non-free riders to explore whether they have equal ratio of uploading to downloading.

These two indexes can evaluate the impact the co-opetition framework makes on free riders and altruists separately. However, this study still does not know when these two indexes are representative. Therefore, the third index, Fairness Index, was introduced to monitor the fairness of the whole P2P network system continuously (Jain *et al*, 1984). Fairness index was originally used in computer networks and can be used to evaluate the equitable distribution of resources, the throughput of every peer, and the fairness of load balancing. Therefore, the fairness of the P2P file sharing system can be monitored by means of the fairness index. Equation (3) is the calculation of the fairness index:

$$\text{Fairness Index} = \frac{\left( \sum_{i=1}^n X_i \right)^2}{n \sum_{i=1}^n X_i^2} \quad (3)$$

$X$  was defined as the ratio of the total quantity of uploading to the total quantity of downloading of every peer. This ratio can remove the variation caused by network bandwidth or peers themselves. By this definition, the fairness index can judge whether the distribution of the ratio of uploading to downloading of every peer is fair. If the total quantity of uploading equals the total quantity of downloading in a P2P system, the fairness index will equal 1 ( $(\sum_{i=1}^n X_i)^2 = n \sum_{i=1}^n X_i^2$ ). Therefore, it can be inferred that every  $X_i$  is the same value which equals 1. It means that, for every peer, the total quantity of uploading is equal to the total quantity of downloading if the fairness index is larger to equal 1. Accordingly, it can be deduced that fairness index is bounded between 0 (unfair) and 1 (fair). The larger the value of fairness index is, the fairer the P2P system is.

However, the fairness index equals 1 does not exist since there are always free riders in a real P2P system. Assume that there are  $N_p$  peers in the P2P network and the ratio of free riders is  $F$ , there are  $F \cdot N_p$  free riders and  $(1-F) \cdot N_p$  non-free

riders. The values of  $X_i$  of free riders are equal to zero, because they would not like to upload resources for sharing. On the contrary, non-free riders are glad to upload resources for sharing; hence the value of their  $X_i$  would be larger than zero. The fairness index will achieve the maximum if  $X_i$  of non-free riders are the same value which is supposed as a constant  $C$ . According to Equation (3), it can be inferred that the maximum of the fairness index is equal to  $1-F$ .

$$\begin{aligned} \text{Fairness Index} &= \frac{(F \cdot N_p \cdot 0 + (1-F) \cdot N_p \cdot C)^2}{N_p(F \cdot N_p \cdot 0^2 + (1-F) \cdot N_p \cdot C^2)} \\ &= 1-F \end{aligned}$$

It shows that the maximum of the fairness index would be limited by the ratio of free riders in a P2P network. Consequently, the fairness index approaching the maximum represents that the values of  $X_i$  of peers is approaching a fixed value and the interactions among peers in a P2P network are achieving a stable status. The evaluation indexes, the number of requests fulfilled and the ratio of uploading to downloading, are representative in a P2P network with stable interaction and therefore can be used to analyse the simulation results.

## 4. Simulation model

Interaction among peers in a real P2P file sharing system would be more complex and hard to predict due to an increasing number of peers. Therefore, it is expensive to interrupt and alter day-to-day operations of a real P2P file sharing system in order to try out the co-opetition framework. Besides, the alterations may cause the operational performance of a real P2P file sharing system to worsen (Robinson, 2004). Consequently, this study verified the feasibility and effectiveness of the co-opetition framework by using simulation without any interruption to the operation of the real-world system. The complete experimental results regarding the performance of the co-opetition framework also can be explored by performing an experiment repeatedly under different experimental conditions. In order to validate the simulation model by comparing with other models, this simulation model will be run under different mechanisms including the co-opetition framework. The simulation model as shown in Figure 1 was built based on the co-opetition framework and the research of Gummadi *et al* (2003). Each step is described below.

#### (1) Initialization

*Input:* the number of peers and resources, the arrival rate of new peers, new resources and requests, the ratio of free riders, the simulation time, and the mechanism parameters. *Output:* the relation between peers and resources, and type of peers.

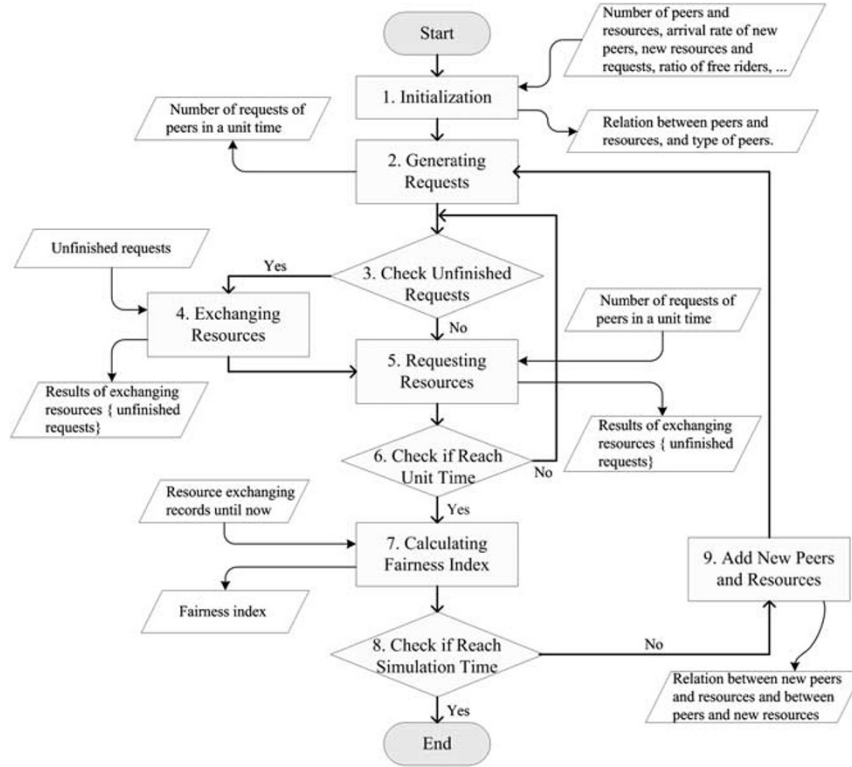


Figure 1 Simulation model.

*Operation:* the simulation parameters above should be inputted before the simulation starts. According to the inputted parameters, the simulation model generates peers and resources, decides the types of generated peers, assigns resources evenly to peers, and builds the relation between peers and resources (Figure 2).

#### (2) Generating Requests

*Output:* the number of requests submitted by peers in a unit time.

*Operation:* according to the arrival rate of requests inputted in step 1, the simulation model decides the number of requests submitted by peers in a unit time and distributes the decided number of resource requests to periods before peers generate resource requests (Figure 3).

#### (3) Checking Unfinished Requests

*Operation:* the simulation model checks if there are unfinished requests. If there are unfinished requests, the simulation goes to step 4 and continues exchanging the rest of resources, otherwise goes to step 5 (Figure 4).

#### (4) Exchanging Resources

*Input:* the unfinished requests.

*Output:* the results of exchanging resources (including the unfinished requests).

*Operation:* peers can decide whether to exchange resources according to the rules of the mechanism they are involved in.

#### -The Algorithm of Step 1-

```

Get simulation parameters;
Add new peers and resources according to simulation parameters;
For every peer {
    decide whether peers are free riders based on the ratio of free riders;
    assign resources to peers evenly;
}
  
```

Figure 2 Pseudo code of step 1.

#### -The Algorithm of Step 2-

```

For every peer {
    decide the number of requests in a unit time;
    distribute the decided number of requests to periods;
}
  
```

Figure 3 Pseudo code of step 2.

#### -The Algorithm of Step 3-

```

If there are unfinished requests
    go to step 4;
else
    go to step 5;
  
```

Figure 4 Pseudo code of step 3.

If there are exchanges of resources, the results of exchanging resources will be recorded. Moreover, if requests are still unfinished, the information regarding the unfinished requests will be updated (Figure 5).



**-The Algorithm of Step 4-**

Get the unfinished requests;  
 Decide whether to cooperate according to the different mechanisms;  
 Record the results of exchanging resources and update the information of the unfinished requests, if needed;

**Figure 5** Pseudo code of step 4.**(5) Requesting Resources**

*Input:* the number of requests submitted by peers in a unit time.

*Output:* the results of exchanging resources (including the unfinished requests).

*Operation:* according to the number of requests submitted by peers in a unit time, peers generate and send resource requests to other peers. The probability that a particular resource will be downloaded is proportional to the rank of resources according to Zipf Law based on the research of Gummadi *et al* (2003). After receiving these requests, peers can decide whether to interact according to the rules of the mechanism they involved in. If there are exchanges of resources, the results of exchanging resources will be recorded. Moreover, if requests are still unfinished, the information about the unfinished requests will be updated (Figure 6).

**(6) Checking if Reach the Unit Time**

*Operation:* the simulation model checks whether running time reaches unit time. If it does, the simulation goes to step 7, otherwise goes back to step 3 and continues the procedure (Figure 7).

**(7) Calculating Fairness Index**

*Input:* the resource exchanging records until now

*Output:* the fairness index

*Operation:* according to the resource exchanging records until now, the simulation model calculates the value of  $X$  of every peer and the fairness index (Figure 8).

**(8) Checking if Reach the Simulation Time**

*Operation:* the simulation model checks whether running time reaches the simulation time. If it does, the simulation is terminated, otherwise goes to step 9 (Figure 9).

**(9) Adding New Peers and Resources**

*Output:* the relation between new peers and resources and between peers and new resources.

*Operation:* when the simulation proceeds to the next unit time, there will be new peers and resources entering the P2P file sharing system. New peers will be assigned some resources randomly, and new resources will also be assigned to peers randomly. Therefore, the relation between new peers and resources and the relation between peers and new resources will be built (Figure 10).

Table 8 lists all parameters of the simulation model and values. Some parameters have more than one value, because this study wanted to explore how the co-opetition framework influences the efficiency of the P2P network under different conditions.

**-The Algorithm of Step 5-**

```
For every peer {
  generate resource requests according to the number of requests;
  For every request {
    select the wanted resource based on Zipf Law;
    select the peer owning the resource;
    request to the peer for the wanted resource;
    decide whether to cooperate according to the different mechanisms;
    record the results of exchanging resources and update the information of the unfinished requests, if needed;
  }
}
```

**Figure 6** Pseudo code of step 5.**-The Algorithm of Step 6-**

Case of time  
 Case: time < unit time  
     go to step 3;  
 Case: time = unit time  
     go to step 7;

**Figure 7** Pseudo code of step 6.**-The Algorithm of Step 7-**

Get the resource exchanging records until now;  
 For every peer {  
     compute the total upload quantity;  
     compute the total download quantity;  
     compute the value of  $X$ ;  
 }  
 Compute and record fairness index;

**Figure 8** Pseudo code of step 7.**-The Algorithm of Step 8-**

Case of time  
 Case: time < simulation time  
     go to step 9;  
 Case: time = simulation time  
     finish the simulation;

**Figure 9** Pseudo code of step 8.**-The Algorithm of Step 9-**

Generate the number of new peers according to the arrival rate of new peers;  
 Add new peers;  
 Assign resources to these new peers randomly;  
 Generate the number of new resources according to the arrival rate of new resources;  
 Add new resources;  
 Assign these new resources to peers randomly;

**Figure 10** Pseudo code of step 9.**5. Performance evaluation**

Many previous studies use simulations to test a proposed mechanism. However, all of them simulated the P2P network in a static environment. This means that there are a fixed number of peers and resources, and no more new peers and resources. This does not conform to the real P2P network. Therefore, this study first tested the co-opetition framework in a static environment, and then expanded the

environment to a dynamic one. In the dynamic environment, there are new peers and resources entering the P2P network continuously. Lastly, this study explored how different co-opetition framework parameters influence the P2P network. In order to check if the overall model is a sufficiently accurate representation of a real P2P file sharing system for easing free-riding problems, this study validates the simulation model by using one of the broad approaches of black-box validation, a comparison with other models (Robinson, 2004; Cheng, 2006; Wang and Brooks, 2007). Therefore, KaZaA and the mechanism proposed by Gupta and Somani (2005) were chosen to compare with the co-opetition framework. KaZaA is one of the more famous P2P file sharing systems, and it is categorized as a decentralized and unstructured P2P network. Gupta's mechanism is based on dynamic game theory, and it also uses the concept of reputation which affects the probability of receiving service to ease free-riding problems.

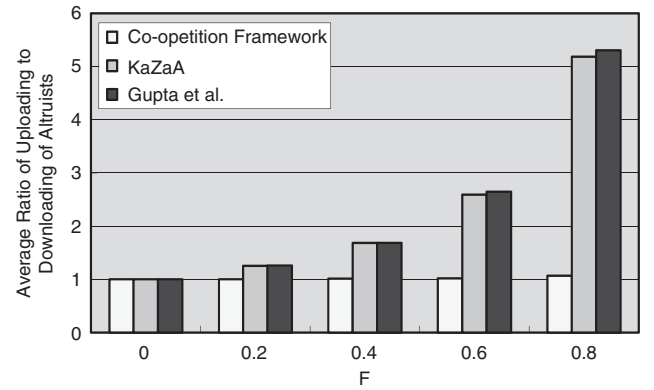
### 5.1. Simulation in static environment

At first, this study compared co-opetition framework, KaZaA, and Gupta's mechanism in a static environment. According to the parameters defined in Table 8, each mechanism should run the simulation combination as shown in Table 9.

According to the inference in Section 3.3, the ratio of free riders determines the maximum value of the fairness index. When the fairness index approaches the maximum, the

**Table 9** Simulation combination in static environment

Combination	$F$	$T$
1	0	6
2	0	12
3	0.2	6
4	0.2	12
5	0.4	6
6	0.4	12
7	0.6	6
8	0.6	12
9	0.8	6
10	0.8	12



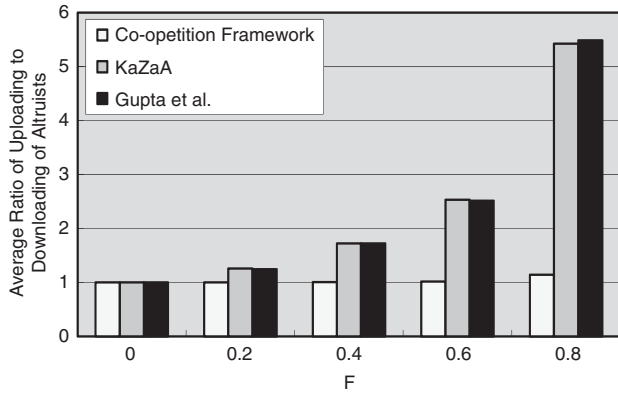
**Figure 11**  $T = 6$ , average ratio of uploading to downloading of altruists.

interaction among peers in the P2P system would become stable. This study compared the number of requests fulfilled and the ratio of uploading to downloading of the three mechanisms when the P2P network was in stable status. This paper argues that when the P2P network comes stable, it must satisfy two conditions: (1) current fairness index is larger than  $(1-F-0.01)$ , and (2) the standard deviation of fairness indexes in forwarding 10 days is less than 0.0005. When condition (1) is satisfied, it means the fairness index is approaching the maximum. Condition (2) represents only small variation in fairness indexes looking forward for a period of 10 days.

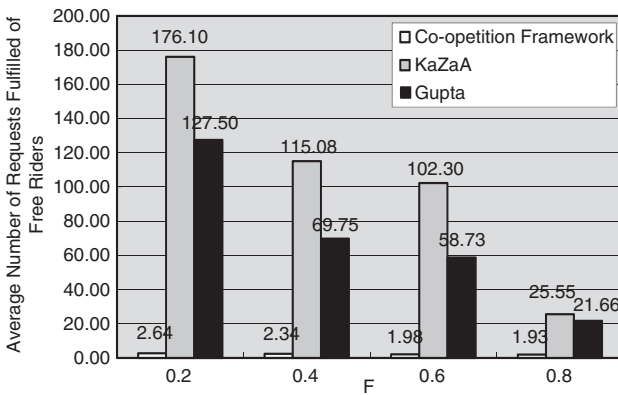
Figures 11 and 12 show the simulation results are similar and do not be influenced by the value of  $T$ . Whatever the value of  $F$  is, the average ratio of uploading to downloading of altruists in a co-opetition framework is better than in KaZaA and Gupta's mechanism. If there are no or few free riders, the average ratio only differs a little among the three mechanisms. However, if there are lots of free riders, altruists in KaZaA and Gupta's mechanism have to bear numerous requests from free riders; their average ratio of uploading to downloading would be larger. Therefore, to achieve the stable status of the P2P network, altruists in KaZaA and Gupta's mechanism have to upload more resources. Although altruists are glad to offer resources, it is

**Table 8** The parameters of the simulation model

Parameter	Description	Value
$T_s$	The number of days which simulation should run	100
$N_p$	The number of peers in initialization	1000
$N_y$	The number of resources in initialization	40 000
$F$	The ratio of free riders	0/0.2/0.4/ 0.6/0.8
$T$	The number of periods in one day (a unit time)	6/12
$\lambda_{\text{request}}$	The mean of the number of requests submitted by peers in one day which is conform to Poisson distribution	5
$\lambda_{\text{resource}}$	The mean of the number of new resources entering the P2P file sharing system in one day which is conform to Poisson distribution	5/50
$\lambda_{\text{peer}}$	The mean of the number of new peers entering the P2P file sharing system in one day which is conform to Poisson distribution	5/50
$\delta$	Threshold in the co-opetition framework	0.2
$\alpha$	Weight of $r_i'$	0.4
$P(R^0)$	Initial value of $P(R)$	0.6



**Figure 12**  $T=12$ , average ratio of uploading to downloading of altruists.



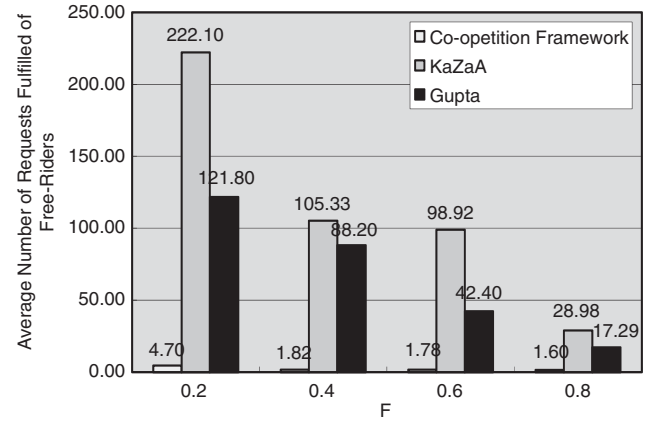
**Figure 13**  $T=6$ , average number of requests fulfilled of free riders.

unfair for altruists to upload more resources to achieve the same stable status. This might irritate altruists, prompt them to leave the P2P network, and cause the P2P network to collapse.

Figures 13 and 14 show that the co-opetition framework can mitigate the free riding problem. One of the limitations to Gupta's mechanism is that it can reduce only the average number of requests fulfilled of free riders by half. To summarize these results, the co-opetition framework can efficiently ease the free riding problem and provide better fairness than other mechanisms.

### 5.2. Simulation in dynamic environments

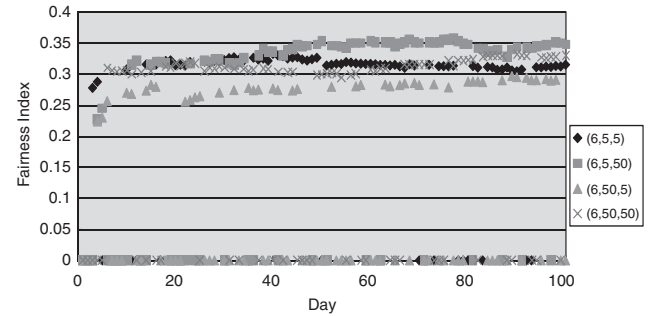
In this section, the study wanted to test whether the co-opetition framework can limit the free riding problem and guarantee the fairness of the P2P network in dynamic environments in which new peers and resources would continue entering. Since there is no previous research focusing on dynamic environments, this study has taken the co-opetition framework as an example to test its effects in a dynamic environment.  $F$  was set as 0.6 to run the simulation. The parameters  $\lambda_{\text{peer}}$  and  $\lambda_{\text{resource}}$  should be



**Figure 14**  $T=12$ , average number of requests fulfilled of free riders.

**Table 10** Simulation combination in dynamic environment

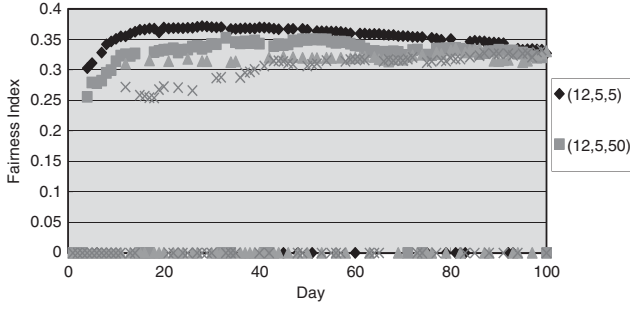
Combination	$F$	$T$	$\lambda_{\text{peer}}$	$\lambda_{\text{resource}}$
1	0.6	6	5	5
2	0.6	6	50	5
3	0.6	6	5	50
4	0.6	6	50	50
5	0.6	12	5	5
6	0.6	12	50	5
7	0.6	12	5	50
8	0.6	12	50	50



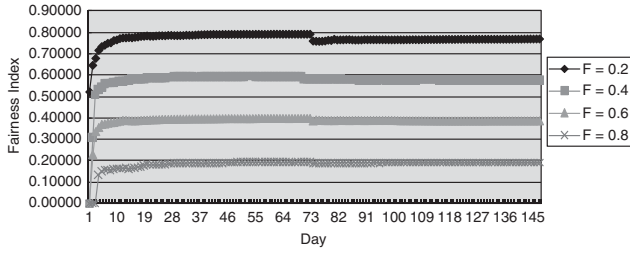
**Figure 15**  $T=6$ , simulation result in dynamic environment by co-opetition framework.

considered in the simulation combination as shown in Table 10, because new peers and resources would continue entering the P2P network in dynamic environments.

Figures 15 and 16 indicate that the fairness index cannot converge to a maximum value in a dynamic environment. There are some days during which the value of the fairness index is zero, because the new entering peers did not download any resources in these days and then the values of  $X$  of new peers approached infinite. Through the use of this simulation, this study can deduce that the fairness index cannot approach the maximum in dynamic environments. Since the fairness index determines whether the ratios of



**Figure 16**  $T=12$ , simulation result in dynamic environment by co-opetition framework.



**Figure 17** Simulation result after adding new peers in different  $F$ .

uploading to downloading of peers are distributed evenly, the entrance of new resources will not affect the fairness index. Therefore, this study discussed only how the new peers influence the value of fairness index and designed a new simulation environment to observe the effect of new peers. The approximate time of the fairness index coming stable is 75 days based on simulation results in a static environment. When the interaction of the original 1000 peers become stable, this study let 50 new peers enter the P2P network and observe whether fairness index converge to the maximum. This simulation was run in different  $F$  to see if the value of  $F$  influences the result.

Figure 17 indicates the simulation result of 150 days. The P2P network eventually became stable only when  $F$  is 0.8, because the fairness index satisfied two conditions. When  $F$  is 0.2, 0.4, or 0.6, the fairness index can only reach stability (satisfying condition (2)), but is not higher than  $(1-F-0.1)$  (not satisfying condition (1)). According to the simulation data, this study concluded that the reason why the fairness index only satisfies condition (2) is because less peers request new peers to upload resources. Hence, the values of  $X$  of new peers were below the average of original peers. As a result, the fairness index cannot reach the maximum and only satisfies condition (2). When  $F$  equals 0.8, there are fewer altruists among new entering peers; hence the interaction among peers of the P2P network is easier to reach stability. This finding explains why the fairness index cannot reach stability when new peers continue entering the P2P network in dynamic environments.

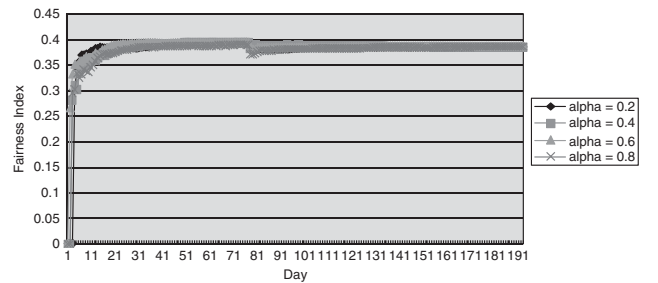
### 5.3. Modify the co-opetition framework

The entrance of new peers influences the stability of the fairness index which represents whether the ratio of uploading resources to downloading resources from every peer is distributed equally. This study wanted to lessen the impact of new peers and shorten the time needed to reach stability by modifying the parameters of the co-opetition framework. The simulation results in last section indicated the P2P network is hard to reach stability in dynamic environments. Therefore, this study referred to the research of Adar and Huberman (2000) and took  $F=0.6$  for an instance to run following simulation.

First, this study modified the weight,  $\alpha$ . If  $\alpha$  is 0, the value of  $P(R)$  is always the same. Therefore,  $\alpha$  was set as 0.2, 0.4, 0.6, and 0.8 to run a simulation of 200 days. Since the time of this simulation is longer than the previous, this study modified the day which new peers enter the P2P network to 80 days. By doing so, original peers have more time to interact with others, and the fairness index can reach a higher value. As a result, the variance between different environments is smaller.

Figure 18 and simulation data show that different values of  $\alpha$  will not allow the fairness index approach maximum and still only satisfies condition (2). The entrance of new peers still makes the fairness index unstable. However, the simulation data show the fairness index is the highest and rises fastest when  $\alpha$  equals 0.6. When  $\alpha$  is 0.8, the high weight of  $r'_i$  creates severe variation of  $P(R)$  which influences fairness index.

At this point, this study modified the threshold,  $\delta$ . If  $\delta$  is 0, it means there is no restriction against free riding. Therefore, the authors set  $\delta$  as 0.2, 0.4, 0.6, and 0.8 and observe simulation results of different  $\delta$ . When  $\delta$  is 0.8, no interaction occurred since the initial value of  $P(R)$  is 0.6. Therefore, this study set the initial value of  $P(R)$  as 1. Figure 19 and simulation data show that altruists can only download resources during a few periods since 0.8 is too high for the threshold. Therefore, altruists have to upload more resources to get the chance to download resources. This situation would have a great impact on the operation of the P2P network. Consequently, the fairness index cannot satisfy any condition of 'stability'. Since some new peers did



**Figure 18** Simulation results of different  $\alpha$ .

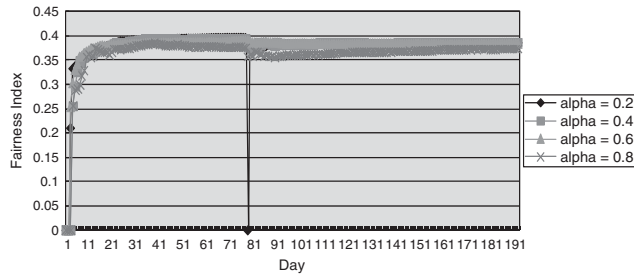


Figure 19 Simulation results of different  $\delta$ .

not download any resources, the fairness index might become 0 when new peers enter the P2P network (Day 81). When  $\delta$  is less than 0.8, the fairness index can only fulfil condition (2). Although the value of  $\delta$  cannot reduce the influence of new peers, a lower value of  $\delta$  can make the fairness index rise more quickly and maintain a higher value of the fairness index. The P2P network therefore can be fairer to all peers, including altruists. According to the simulation results above, the appropriate values of  $\alpha$  and  $\delta$  of the co-opetition framework make the fairness index rise faster and satisfy condition (2) in a shorter period. Therefore, the appropriate values of  $\alpha$  and  $\delta$  can ease the free riding problem and guarantee fairness to altruists. Lastly, this study made a comparison between the modified co-opetition framework, KaZaA, and Gupta's mechanism as shown in Figure 20.

According to the simulation results of 200 days, the three mechanisms cannot make the fairness index stable, as expected. However, the modified co-opetition framework can make the fairness index rise faster, thereby allowing for a higher fairness index after the simulation. Figures 21 and 22 signify the average ratio of uploading to downloading by altruists and the average number of requests fulfilled of free riders after simulations of 200 days. Under nearly the same efficiency, altruists in the co-opetition framework only have to upload the quantity of resources that they downloaded. In conclusion, these simulations show that the co-opetition framework can ease the free riding problem efficiently in a P2P network and improve the efficiency of the P2P file sharing system.

## 6. Conclusion and future work

This study proposed the use of practical co-opetition framework to solve the issue of free riding in P2P networks. The authors identified different peers in P2P networks, used dynamic and incomplete game theory to model the interaction between peers, and explored what causes the free riding problem. By implementing some simple rules of the co-opetition framework, along with the weight and the threshold, this study can identify peers who were free riding in the past and limit their chance to download resources

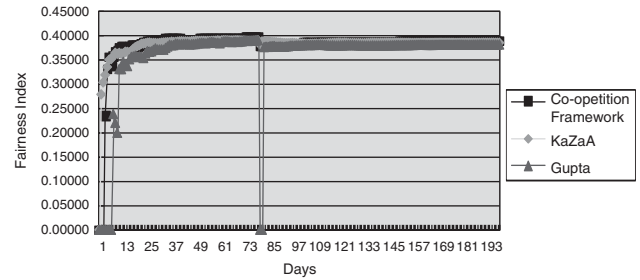


Figure 20 Modified co-opetition framework *versus* KaZaA *versus* Gupta's Mechanism.

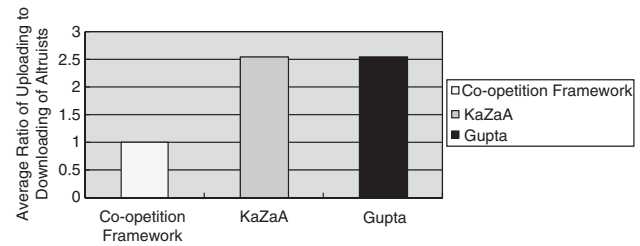


Figure 21 Average ratio of uploading to downloading of altruists.

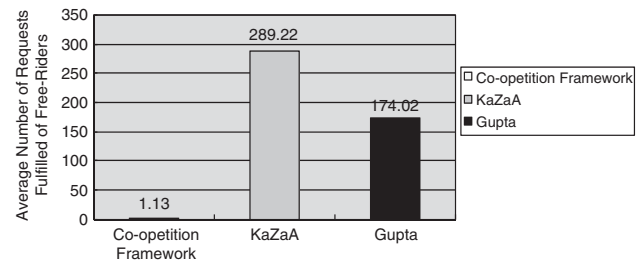


Figure 22 Average number of requests fulfilled of free riders.

from other peers. If free riders want to download additional resources, they have to upload resources to raise their reputation. This study also used simulation to test the co-opetition framework and make comparisons with KaZaA and Gupta's mechanism under different environments. Three stages of simulation were run, including static environment, dynamic environment, and modifying the co-opetition framework. Simulation results not only shown the reasons which caused the fairness index to be unstable in dynamic environments, but also demonstrated how the co-opetition framework can actually ease free riding problem and improve the efficiency of the P2P file sharing system—KaZaA and Gupta's mechanism cannot.

This paper divides peers into different types according to their responses when facing requests for resources. Through this natural division, one could explore the interaction of peers more deeply. This paper also builds the co-opetition framework to ease free riding and to improve the efficiency of the P2P network. The fairness index is proposed to

measure the efficiency of the P2P network. These could be launching points for future studies. All of the above analysis could be also used once new P2P software is developed that is widely used.

This paper focused on the issue of free riding and assumed there is no illegal behaviour in a P2P network, such as cheating. However, cheating is possible in real P2P network environments. If one can consider this issue and take some appropriate measures, the co-opetition framework would be more robust. This paper only discussed the co-opetition framework in decentralized and unstructured P2P networks. If one can apply the co-opetition framework to centralized or decentralized and structured P2P networks, the feasibility and availability of the co-opetition framework would be improved. This study used simulations to test the effects of the co-opetition framework and find the appropriate parameter combination. In the future, the co-opetition framework might be applied to P2P software and the parameter combination could be used as a case study. According to the experimental results, further parameter modification might be needed.

*Acknowledgements*—The work presented in this paper has been supported by National Science Council, Taiwan, R.O.C, under Grant No. NSC97-2410-H-194-074-MY3. We deeply appreciate their financial support and encouragement.

## References

- Adar E and Huberman BA (2000). Free riding on Gnutella. *First Monday* 5(10), <http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/issue/view/124>.
- Betts B (2005). P2P goes to work, <http://www.techworld.com/features/index.cfm?featureID=1722>, accessed 30 January 2011.
- Bo CW and Li B (2003). *Peer-to-peer overlay networks: A survey* Technical Report, Department of Computer Science, Hong Kong University of Science and Technology.
- Cheng RCH (2006). Validating and comparing simulation models using resampling. *J Simulation* 1: 53–63.
- Fehr E and Schmidt KM (1999). A theory of fairness, competition and cooperation. *Quart J Econ* 114(3): 817–868.
- Feldman M, Lai K, Stoica I and Chuang J (2004). Robust Incentive Techniques for Peer-to-Peer Networks. In: *Proceedings of the 5th ACM Conference on Electronic Commerce in New York, NY, USA*. ACM Press: New York, USA, pp 102–111.
- Golle P, Leyton-Brown K and Mironov I (2001). Incentives for sharing in peer-to-peer networks. In: *Proceedings of the 3rd ACM Conference on Electronic Commerce in Tampa, FL, USA*. ACM Press: New York, USA, pp 264–267.
- Gummadi PK et al (2003). Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In: *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles in Bolton Landing, NY, USA*. ACM Press, New York, USA, pp 314–329.
- Gupta R and Somani AK (2005). Game theory as a tool to strategize as well as predict nodes behavior in peer-to-peer networks. In: *Proceedings of the 11th International Conference on Parallel and Distributed Systems—Volume 01 in Fukuoka, Japan*. IEEE Computer Society: Washington, DC, USA, pp 244–249.
- Jain R, Chiu D and Hawe W (1984). *A quantitative measure of fairness and discrimination for resource allocation in shared systems*. DEC Research Report TR-301, Digital Equipment Corp.
- Krishnan R, Smith MD, Tang Z and Telang R (2004). The impact of free riding on peer-to-peer networks. In: *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04)—Track 7—Volume 7 in Hawaii, USA*. IEEE Computer Society: Washington, DC, USA.
- Matsuda Y, Sasabe M and Takine T (2010). Evolutionary game theory-based evaluation of P2P file-sharing systems in heterogeneous environments. *Int J Digital Multimedia Broadcast*: 1–12. Article ID 369814.
- Owen G (1995). *Game Theory*. 3rd edn, Academic Press: New York.
- Rasmusen E (2006). *Games and Information: An Introduction to Game Theory*. 4th edn, Wiley-Blackwell: Cambridge, MA.
- Resnick P, Kuwabara K, Zeckhauser R and Friedman E (2000). Reputation systems. *Commun ACM* 43(12): 45–48.
- Robinson S (2004). *Simulation: The Practice of Model Development and Use*. John Wiley & Sons: Hoboken, NJ.
- Singh MP (2001). Peering at peer-to-peer computing. *IEEE Internet Comput* 5(1): 4–5.
- Stoica I et al (2001). Chord: A scalable peer-to-peer lookup service for internet applications. In: *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications in San Diego, California, USA*. ACM Press: New York, USA, pp 149–160.
- Tseng YM and Chen FG (2011). A free-rider aware reputation system for peer-to-peer file-sharing networks. *Expert Syst Appl* 38(3): 2432–2440.
- Wallace PM (2001). *The Psychology of the Internet*. Cambridge University Press: New York.
- Wang J et al (2010). Resisting free-riding behavior in BitTorrent. *Future Generation Comput Syst* 26(8): 1285–1299.
- Wang L (2011). SoFA: An expert-driven, self-organization peer-to-peer semantic communities for network resource management. *Expert Syst Appl* 38(1): 94–105.
- Wang T, Lu X and Hou M (2005). A novel incentive mechanism for P2P systems. In: *Proceedings of the Sixth International Conference on Parallel and Distributed Computing Applications and Technologies in Dalian, China*. IEEE Computer Society, Washington, DC, USA, pp 801–803.
- Wang W and Brooks RJ (2007). Improving the understanding of conceptual modelling. *J Simulation* 1: 153–158.
- Xu HM, Lu XL and Li L (2009). A novel incentive mechanism based on debt theory for P2P file-sharing networks. *J Elect Electron Eng* 9(2): 1047–1056.
- Yang L et al (2010). A reputation-based method for controlling free-ride in P2P networks. In: *Proceedings of the 2010 International Conference on Communications, Circuits and Systems in Chengdu, China*. IEEE Computer Society: Washington, DC, USA, pp 249–253.
- Zhu Q, Fung C, Boutaba R and Basar T (2009). A game-theoretical approach to incentive design in collaborative intrusion detection networks. In: *Proceedings of the International Conference on Game Theory for Networks 2009 in Istanbul, Turkey*. IEEE Computer Society, Washington, DC, USA, pp 384–392.

Received 20 June 2009;  
accepted 25 March 2011