Journal of Universal Computer Science, vol. 16, no. 10 (2010), 1368-1387
submitted: 14/7/08, accepted: 9/3/10, appeared: 28/5/10 © *J.UCS*

Classifying and Tracking Free Riders in Multimedia-Based Systems

Farag Ahmed Azzedin
(Department of Information & Computer Science, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia
fazzedin@kfupm.edu.sa)

Abstract: The ever growing explosion in technological advancements is paving the way to the expansion of multimedia applications. Unfortunately, current multimedia applications use centralized architectures. Before decentralized architectures are utilized and used, some issues related to decentralization must be addressed. In this paper, we focus on the problem of free riding in decentralized collaborative environments. We propose a novel taxonomy of free riders in multimedia systems based on trustworthiness. To the best of our knowledge, no existing literature considers trustworthiness, which we believe is a vital dimension that should be considered when identifying free riders. We also propose a new mechanism to filter out and isolate free riders. Our extensive simulation experiments show that our proposed algorithm is reasonably successful in identifying free riders in multimedia-based systems.

Keywords: Multimedia systems, Free riders, P2P, Distributed computing
Categories: C.2.0, C.2.4, H.5.1

1 Introduction

Multimedia services and applications are experiencing a significant amount of attraction due to the current technological advancements in communication, computation, and storage [Mol et al. 2008, Azzedin and Ridha 2010]. Multimedia services and applications include the use of a wide range of multimedia within web sites such as video-streaming, audio-streaming, movies, videos, or audio files, and others. For example, some web sites [Live broadcasting, Canberra] transmit live webcasting of proceedings of the Senate debates, host virtual tours in the form of QuickTime movies, as well as provide interactive games to play.

The widespread and the success of collaborative applications such as multimedia applications and services are due mostly to the nature and benefits of collaboration. To name few of these benefits, collaboration nurtures cross-boundary sagacious solutions, increases the overall work done by the collection of shared resources, and creates an environment for mutual gains. Through collaboration, digital entertainment content such as image clips, video, and movie posts can be shared and viewed from remote nodes. Television broadcasts and the various video sharing websites such as YouTube are examples of the popularity and the explosive of multimedia-based applications attracting users from allover the globe.

Unfortunately, current multimedia applications use centralized architectures [Mol et al. 2008, Silverston et al. 2008, Chatzidrossos and Fodor 2008], which impose the use of high-end servers, high communication lines, as well as efficient database engines to deliver the contents to their huge domain of users. Additional undesirable features of centralized architectures including poor scalability, single point of failure, and communication bottleneck; have turned the eyes to solutions based on distributed architectures to boost the advancement of multimedia-based applications.

Employing decentralized systems have demonstrated to be efficient in distributing multimedia content over the Internet. In particular, *peer-to-peer* (P2P) architectures have proven [Silverston et al. 2008, Mol et al. 2008] the feasibility of global multimedia content. BitTorrent [Shah and Paris 2007], the large scale standard P2P technology for the delivery of rich media on the Internet, has recently joined forces with Orb Networks [OrbNet], the streaming media leader, to bundle BitTorrent software client applications and the Orb PC applications together. This joint effort will make BitTorrent downloads feasible and enjoyable on any connected device. Multimedia streaming such as PPStream [ppst], SOPCast [sopcast], PPLive [pplive], as well as live TV over P2P network (P2PIPTV) are emerging as successful applications of the growing explosion of the digital multimedia-based era. This era is moving towards a large scale multimedia content sharing.

The large scale multimedia content is a collaborative effort from users willing to share and contribute to achieve collectively and successfully a joint goal. Unfortunately, if users start to consume resources more than they contribute, then the burden of uploading will be on the altruistic users. These altruistic users might be very few and this will create bottlenecks resulting in degrading the *quality-of-service* (QoS). In both live streaming and *video-on-demand* (VoD), a minimal download speed is needed to sustain playback. As such, if users do not contribute, distributing multimedia content over the Internet may not survive. Also, burdening the collaborative users with huge requests might force them to be non-collaborative resulting in adding insult to injury. Furthermore, non-collaborative users negate the advantages of distribution and hence convert a distributed architecture to few scattered centralized nodes.

Non-collaborative users, who do not contribute to the community, are referred to as *free riders* in the literature [Chatzidrossos and Fodor 2008]. The phenomenon of free riding does not just create a tedious task that can be bothersome for the contributors, but also can create a potential performance bottleneck. As a consequence, responses to queries can experience a longer delay. In a collaborative file sharing system, 66% of users do not share files (i.e., free riders) and the top 25% users account for 99% of all downloads [Tang et al. 2004]. Free riding must be tackeled before P2P-based technology can be used to provide the necessary substrate to multimedia applications and services so that they can be redeployed over P2P networks.

This paper is organized as follows. In the remainder of the Introduction section, we outline and discuss the motivation behind writing this paper as well as its contributions. In section 2, we provide a taxonomy of free riders. We describe the related work in Section 3 and explain our free riders' detection algorithm in Section 4. In Section 5, we present the performance metrics and the simulation setup to evaluate our proposed detection algorithm. The results are presented and discussed in Section 6. Section 7 concludes the paper and envisions future directions to enhance our approach.

1.1 Motivation

We were motivated by the fact that existing free riders' detection algorithms classify a node as a free rider based on the node's contribution versus consumption as shown in Equation 1. Equation 1 will consider a node that contributes untrustworthy content or resources as not a free rider. In this paper, we argue that an untrustworthy node (i.e., a node providing corrupted resources) fits the classification of a free rider.

$$\Omega = \frac{consumption}{contribution} \tag{1}$$

The consumption-to-contribution ratio (i.e., Ω) is a reasonable measure to detect a free rider under the assumption that the node is trustworthy. A node is trustworthy if it does not contribute content or resources that inflect damage to other nodes. The intent of an untrustworthy node is to inject corrupted resources and hide its malicious behavior. To pursue this vicious intent, an untrustworthy node can exploit Ω. This can be done by injecting the corrupted resources such that the malicious node is considered to be contributing and hence its Ω is satisfactory.

In fact, when an untrustworthy node provides corrupted resources, it is actually consuming resources from other nodes without contributing anything. If node x is providing a corrupted file and node y is interested in downloading that file, then node y will allocate some of its resources to download the corrupted file. These wasted allocated resources as well as the time node y spent to download the corrupted file are all indirectly consumed by node x.

In addition, node x's behavior negates the advantages of distribution and sharing. Node x also contributes negatively to the active role of contributing nodes as they become overloaded and this might encourage selfishness. An untrustworthy contributor fits the classification of free riders because such a node is not contributing to the community, such a node is allowed to consume since its Ω is satisfactory, and such a node discourages trustworthy contribution. Therefore, an untrustworthy node does not only fit the classification of a free rider but also introduces new challenges to the success of multimedia services and applications.

1.2 Contributions

We propose a new mechanism to filter out and prevent free riders from degrading and hence bringing collaborative environments such as multimedia systems to its knees. In our mechanism, we use the notion of trustworthiness. Trustworthiness is a notion that describes the behavior of a node and it is a vital component in any Internet-based transaction [Azzedin and Ridha 2010]. Throughout this paper, we refer to the node that wants to consume resources as a *source node* whereas the node that provides resources as a *target node*.

The contributions of this paper is three folds: (a) draw the attention and raise the awareness to the fact that trustworthiness is an important attribute for identifying free riders, (b) provide a trust-based taxonomy for free rider, and (c) propose and evaluate a novel activity-based algorithm to identify free riders in multimedia-based systems. To the best of our knowledge, no existing literature tackles this issue, which we believe is a vital dimension that should be considered when identifying free riders.

2 Free Riders Taxonomy

Free riders attracted the attention of many research groups. Current research done on streaming systems [Silverston et al. 2008, Chatzidrossos and Fodor 2008] as well as research on P2P and VoD [Mol et al. 2008, Karakaya et al. 2008] have used a contribution-based taxonomy to classify free riders. Theses researchers have assumed that free riders are trustworthy and they are either selfish, consumers, or droppers. A selfish free rider is a node that does not share its own resources, does not replicate resources received from others, but it routes messages and generates normal traffic. A consumer free rider is a node that might share its own resources , does not replicate resources received from others, but it routes messages and generates high traffic. A dropper free rider is a node that does not share its own resources , does not replicate resources received from others, generates normal traffic, but does not route messages.

All these researchers use the ratio consumption over contribution as a performance metric to identify a free rider, where consumption might be number of files downloaded or query hit messages and contribution might be number of uploads or query messages sent to neighbors. We argue that a free rider might be untrustworthy (i.e., malicious) and hence contribute its malicious content or even contribute modified versions of replicas received from other nodes. An untrustworthy node will not be identified as a free rider using the ratio performance metric. As such, an untrustworthy node will be given the chance to damage a collaboration system and will not be labeled as a free rider.

We propose a taxonomy of free riders based on trustworthiness as shown in Figure 1. A free rider can be selfish meaning that the node is not contributing at all but only using or consuming resources. On the other hand, a free rider can contribute. Taking trustworthiness into consideration, a contributor fee rider can be either trustworthy or untrustworthy. A trustworthy or untrustworthy contributor free rider can easily detect if it is identified because its downloads, for example, are becoming slow since it is given smaller bandwidth than it used to get previously.

A contributor free rider can adapt a flip-flop phenomenon. Once it detects that it was identified as a free rider, it will start to contribute and then it will not contribute until it is again identified as a free rider. This cycle will repeat and the free rider will conditionally contribute and not for the sake of contribution. An untrustworthy free rider will negatively contribute to the collaborative system and will not be identified as a free rider. It can inject its malicious content based on conditions such as a time frame (i.e., temporal), when needed (i.e., on need), infecting just a specific regions or group of users (i.e., spacial), or worse infecting everyone any time (i.e., always).

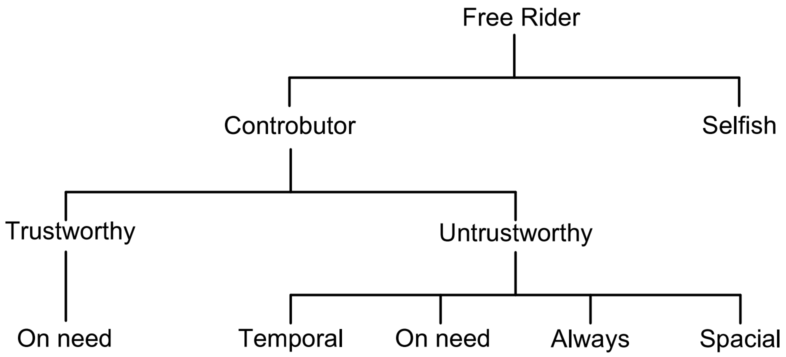


Figure 1: Trust-Based taxonomy for Free Riders.

3 Related Work

In the following two sections, we will shed some light on two free-riders-related issues. First, we will discuss the importance of free riders control and management in multimedia and ubiquitous applications. We will also emphasize that free riders identification is an essential component to the success of such multimedia-based applications. Second, we focus on free riders phenomenon in the P2P literature. Since the current multimedia applications use centralized architectures, we will discuss how free riders' filtering is done in P2P systems for the purpose of learning lessons and techniques to deploy distributed multimedia applications that are resilient to free riders.

3.1 Free Riders in Multimedia and Ubiquitous Applications

With the recent proliferation of multimedia applications and the deployment of ad-hoc networks, it became possible for peers to communicate and share resources using video collaboration and e-meeting services from almost any location at anytime [Kristiansson 2008]. In multimedia applications, trust is recognized by the research community as an important factor that contributes to the success such applications.

The *Ubiquitous Web Applications Working Group* [Ubiq 2006] is a working group seeking to simplify the creation of distributed Web applications, including multimedia applications. They identified trustworthiness as playing a vital role to enable widespread usage of distributed multimedia applications. Currently [Hwang and Yuan 2007, Burmester et al. 2006, Chen et al. 2007], researchers have started to resolve the trustworthiness in multimedia and e-services utilizing ad-hoc ubiquitous environments. Their aim is to facilitate and empower the potential benefits of various distributed multimedia applications and to makes it possible for nodes to safely collaborate in a multimedia-based distributed environment.

Studies on IPTV [Xiaojun et al. 2007, Magharei and Rejaie 2007] as well as live streaming [Pianese et al. 2007, Fodor and Dan 2007] have been recently attracted researchers to examine approaches and to compare different solutions to deploy multimedia-based applications in large-scale environments. A vital challenge for this deployment is the issue of free riding [Chatzidrossos and Fodor 2008, Mol et al. 2008, Silverston et al. 2008].

In [Chatzidrossos and Fodor 2008], researchers evaluated the effect of trustworthy free riders in a streaming system and introduced a policy-based approach to reduce the number of free riders in the system. In their scheme, the authors proposed a way to accommodate free riders that can not afford upload capacity with high bitrate and reliability. The idea is to integrate such free riders in the environment rather than isolating them since a collaborative environment's success is to involve as many nodes as possible as long as these nodes are not purposely selfish.

A contribute-to-consume treatment is proposed in [Mol et al. 2008] for a P2P VoD system, where a free rider is identified based on its ratio of consumption over contribution. Again, the authors assumed that free riders are trustworthy and therefore well-behaved. The proposed algorithm is inspired by BitTorrent [Shah and Paris 2007] and the performance evaluation shows that the algorithm provides good performance as long as the all peers are trustworthy and the shared videos are short, which the authors claim that short videos dominate the current VoD traffic on the Internet.

Incentives for free riders in streaming live multimedia systems are proposed in [Silverston et al. 2008]. The authors' aim is to deploy live multimedia streaming over P2P systems and hence uncovering distributed approaches to manage multimedia-based applications. In their approach, the authors emphasize on the temporal importance that exists in live multimedia streams. Therefore, block a in a stream, needs to be consumed before block $a+1$ and hence receiving block a is more important than receiving block $a+1$. This temporal property might make a certain node not transmitting, not because it is non-collaborative, but because the temporal constraints makes the transmission pointless.

3.2 Free Riders in Peer-to-Peer Applications

P2P systems have proven its success due to their attractive properties such as self-organizing, scalability, distributed resource sharing, self-autonomy, and more [Obele et al. 2009, Maille and Toka 2008, Karakaya et al. 2008]. Unfortunately, issues such as free riding is an important concern pertaining to resource sharing. Many researchers proposed solutions, to detect and discourage free riding, ranging from game theory [Buragohain et al. 2003] to providing incentives [Obele et al. 2009, Karakaya et al. 2008, Maille and Toka 2008].

Treating each peer as a rational player, a game theory scheme was proposed in [Buragohain et al. 2003] to give incentives to peers and hence discourage and isolate free riders. A rational peer wants always to maximize its profit by participating in the P2P network. This profit depends on the benefits (i.e., the systems resources he can use) and the cost (i.e., his contribution).

The authors in [Obele et al. 2009] used an algorithm to track free riders in file sharing applications. The authors used the popular chocking algorithm [Shah and Paris 2007] and how free riders affect the file sharing network. The authors excluded the concept of malicious nodes (i.e., they assumed that all nodes are trustworthy) and they used the ratio of download over upload to detect and isolate free riders.

In [Maille and Toka 2008], the authors addressed user incentives in a P2P storage system. In their paper, the authors proposed that each node will use resources based on its contribution. Using game theoretical analytical model to describe free riders reactions, the authors devised a scheme to over incentives for cooperation and model users's behavior in relation to resource availability.

Monitoring a peer's contribution to the network, a framework have been devised in [Karakaya et al. 2008] against free riders in unstructured P2P networks. This framework indirectly forces free riders to cooperate. To identify whether peer a is a free rider, the framework monitors the number of QueryHit messages originating from peer a and compares them to the number of Query messages sent by peer a to its neighbors.

4 The Proposed Approach

4.1 Free Riders' Filtering Algorithm

The aim of the proposed free riders' filtering algorithm is to identify and isolate free riders. In our algorithm, every peer maintains a black list. Each time a source peer interacts with a trustworthy target peer, two things can happen. If the target peer is trustworthy, the source peer globally posts an appraisal. This global posting can be easily established in structured P2P infrastructures such as P-Grid [Aberer et al. 2005] and the information is kept by a third-party peer, e.g., the storage peer. On the other hand, if the target peer is untrustworthy, the source peer will not post anything into the global list but it will mark the untrustworthy peer in its black list. In other words, a source peer can appraise but it can not complain. Using its black list, a source peer will not interact with a black listed target peer.

The algorithm uses an activity-set model to monitor the activeness of each peer. By activeness, we mean the trustworthy contribution of a peer. Let Δ be the activity-set window. The idea is to examine the number of times peer a got appraised in the current activity-set window Δ_c^a and the previous activity-set window Δ_p^a and hence the trustworthy contribution of peer a (TC_a) can be measured according to equation 2.

$$TC_a = \begin{cases} 1 & \text{if } \Delta_c^a > AC \times \Delta_p^a \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

In equation 2, AC is the activity constant, where $0 \leq AC \leq 1$ and is used to tune the comparison operation between Δ_c^a and Δ_p^a. If the TC_a is 0 , then peer a is considered to be a free rider. Otherwise, it is not a free rider.

Equation 2 will consider peer a with constant trustworthy contribution x as not a free rider even if $x = 1$. That is, a contributes only once in each Δ. To solve this problem, our algorithm uses a contribution factor of peer a (CF_a) that is calculated as follows:

$$CF_a = \frac{x}{N_{num}} \tag{3}$$

As stated in [Mihaela et al. 2008], when peers download and exchange conventional multimedia content, e.g., MP3 files for popular music, the content will likely be uniformly distributed. That is, if $\varDelta$ is determined based on the number of transactions (T_{num}), then the average number of transactions (N_{num}) can be easily calculated based on Equation 4, where P_{num} is the number of peers in the environment.

$$N_{num} = \frac{T_{num}}{P_{num}} \tag{4}$$

Now, if CF_a is greater than a threshold, then the contribution factor of peer a is acceptable and we can proceed to Equation 2. Otherwise, peer a is considered a free rider.

Algorithm 1 Appraisals posting algorithm. This algorithm is run by peer a

1: **procedure** ActivityBased
2:     $BL_a \leftarrow \{\}$
3:     **repeat**
4:         select a target peer t
5:         **if** $t \notin BL_a$ **then**
6:             perform the transaction with t
7:             **if** t is trustworthy **then**
8:                 peer a posts an appraisal for t
9:             **else**
10:                 $BL_a \leftarrow BL_a + t$
11:             **end if**
12:         **end if**
13:     **until** peer a is not interested in performing transactions
14: **end procedure**

For further illustration of how the proposed free riders' filtering algorithm works, Algorithm 1 shows how the appraisals are posted and Algorithm 2 shows how these appraisals are utilized to filter free riders.

Algorithm 1 shows the appraisals posting process performed by peer a. Line 2 initializes peer a's black list. In line 4, peer a choses a target peer t. Lines 5 and 6 state that if t is not black listed by a, then a will perform the transaction with t. Lines 7 to 12 states that if t is not in the black list of peer a, then a will interact with t. If a finds that t is trustworthy (i.e., shares trustworthy content), then a will post an appraisal with the storage peer of t. Otherwise, a will black list t.

For further illustration, the steps of the filtering algorithm are shown in Algorithm 2. This algorithm can be run by each storage peer in the P-Grid

Algorithm 2 The filtering algorithm. This algorithm is run by the storage peer of peer a

1: **procedure** The filtering algorithm
2: $\quad x = \Delta_c^a$
3: $\quad y = \Delta_p^a$
4: $\quad CF_a = \frac{x}{N_{num}}$
5: $\quad$**if** CF_a is satisfactory **then**
6: $\quad\quad$Compute TC_a based on x and y
7: $\quad\quad$**if** $TC_a == 1$ **then**
8: $\quad\quad\quad$this specific peer is not a free rider in this specific Δ_c
9: $\quad\quad$**else**
10: $\quad\quad\quad$this specific peer is a free rider in this specific Δ_c
11: $\quad\quad$**end if**
12: $\quad$**end if**
13: **end procedure**

[Aberer et al. 2005] infrastructure. In other words, if peer y is the storage peer for peer a, then y will determine whether a is a free rider or not. Algorithm 2 will start by computing the number of appraisals posted for a specific Δ. In lines 2 and 3, the storage peer gets the number of appraisals for the current and the previous activity-set windows. Then in lines 4 to 10, the contribution factor is calculated. If the contribution factor is satisfactory, peer a is identified whether it is a free rider by calculating TC_a as shown in equation 2.

5 Performance Evaluation

Filtering free riders aim to reduce the risk of loss, which is aggravated by untrustworthy contributors trying to pollute collaborative-oriented environments such as multimedia applications. The objective of our proposed free riders' filtering mechanism is to detect all types of free riders discussed in section 2. Existing mechanisms assume that free riders are trustworthy. However, this assumption is not realistic because an untrustworthy contributor can spoil solutions trying to create a seamless media experience harnessing all of the rich media downloads peers have on their computing systems, including movies, TV shows and music, and provides them with the ability to enjoy all of that content, anywhere, on any screen or audio device.

5.1 Performance Metrics

We use the following performance metrics to evaluate the effectiveness of the free riders' filtering algorithm:

1. **Detection Tolerance.** The free riders' filtering process ideally distinguishes free riders from positive contributors so that positive contributors can be recognized and rewarded for their effort in boosting and making multimedia applications a success. For the detection tolerance, we measure the following:
   (a) The percentage of free riders detected as free riders (true positives, i.e., T+).
   (b) The percentage of non-free riders misdetected as free riders (false positives, i.e., F+).
2. **Free riding Impact.** Free riders would affect how the source peer makes its decision. We measure the sensitivity of free riding towards the decision making by computing:
   (a) The percentage of lost trustworthy transactions. Lost trustworthy transactions are due to F+.
   (b) The percentage of committed untrustworthy transactions. Committed untrustworthy transactions are due to not identifying untrustworthy contributors.

5.2 Simulation Setup

We use 2048 peers and the simulation runs until in average each peer performs 40 transactions. The source and the target peers for each transaction are randomly generated from $[1, N]$, where $N = 2048$. The transaction will be conducted if the target peer is predicted as a positive contributor. If the transaction is committed, the source peer globally will provide an appraisal based on the trustworthiness of the target peer. In the simulation, we used P-Grid [Aberer et al. 2005] to post the appraisals. If the target peer is untrustworthy free rider, the target peer will be marked in the source peer's black list.

We experiment with three types of free riders, namely selfish (S), trustworthy contributors (TC), and untrustworthy temporal contributors (UTC). For each type, we use different proportions of free riders, i.e., 20%, 40%, 60%, and 80. All results shown are an average of 10 simulation runs. The size of 2048 nodes provides a confidence interval of 3.05% and 4.01% at confidence level of 95% and 99%, respectively, if the population size is $118,925$ nodes (the size of Gnutella network in February 2006 [Xie et al. 2008]. If the population size is assumed to be 1 million nodes, the confidence interval changes slightly to 3.07% and 4.03% at confidence level of 95% and 99%, respectively. As for the average number of transactions performed by peers, it indicates the length of the simulation. A free riders' filtering mechanism should be able to detect free riders quickly so that they do not have many opportunities to pollute the environment.The activity constant AC is set to 0.66.

The simulation starts with no transactions performed so there are no appraisals available. We use optimistic stranger policy i.e., we assume that new peers are not free riders. If a trustworthy source peer transacts with an untrustworthy contributor, it will detect the malicious content and will not offer it to the community. For the simulation, we assume that all peers are honest. That is, a source peer will not appraise an untrustworthy contributor and will appraise a trustworthy contributor.

6 Results and Discussion

6.1 Detection Tolerance

In this set of experiments, we examined the detection rate for S, TC, and UTC free riders as shown in Figures 2 to 5. Our proposed algorithm detects free riders by monitoring the number of appraisals a peer gets as explained in Section 4.

With 20% free riders as shown in Figure 2(a), the detection rate is between 96% and 100% for S, TC, and UTC free riders. The detection rate falls a little for the TCs because the peer's transactions may fall on either side of Δ boundary (i.e., a TC free rider a gives an appraisal in time slot x but receives an appraisal in time slot $x + 1$. In that case it will not be detected as a free rider. In Figure 2(b), the misdetection rate for the three types of free riders is very low and ranges from 1% to 2%. From these results, we see the our algorithm maximizes T+ and minimizes F+.

Figure 3(a) shows the detection rate for 40% free riders. The detection rate is 100% for S free riders since this type of free riders uses resources and does not share resources. So, the S free riders are always source peers and never target peers. Since an appraisal is made by the source peer for the target peer after the transaction is successful, this type of free riders will never get an appraisal and hence is easily detected. For the UTC free riders, the detection rate is also 100%. Sincethe UTC free riders contribute untrustworthy content, they will never get an appraisal. As such, this type of free riders is detected by our algorithm as well. On the other hand, the detection rate for the TC free riders ranges from 96.70% to 98.53% because the TC basically adapts a flip-flop phenomenon based on need. In spite of that, the detection rate of our algorithm of this type of free riders is more than 96%.

The misdetection rate when free riders are 40% is shown in Figure 3(b). The misdetection rate is very low and ranges from 0.1% to 0.7% because the non-free riders are only 60% compared to 80% in Figure 2(b). As it shows in Figures 2(b), 3(b), 4(b), 5(b) there is a trend in the misdetection rate. As the number of non-free riders gets smaller, the misdetection rate also gets smaller. These results validate our simulator and logically follow the definition of the misdetection rate which is the percentage of non-free riders misdetected as free riders.

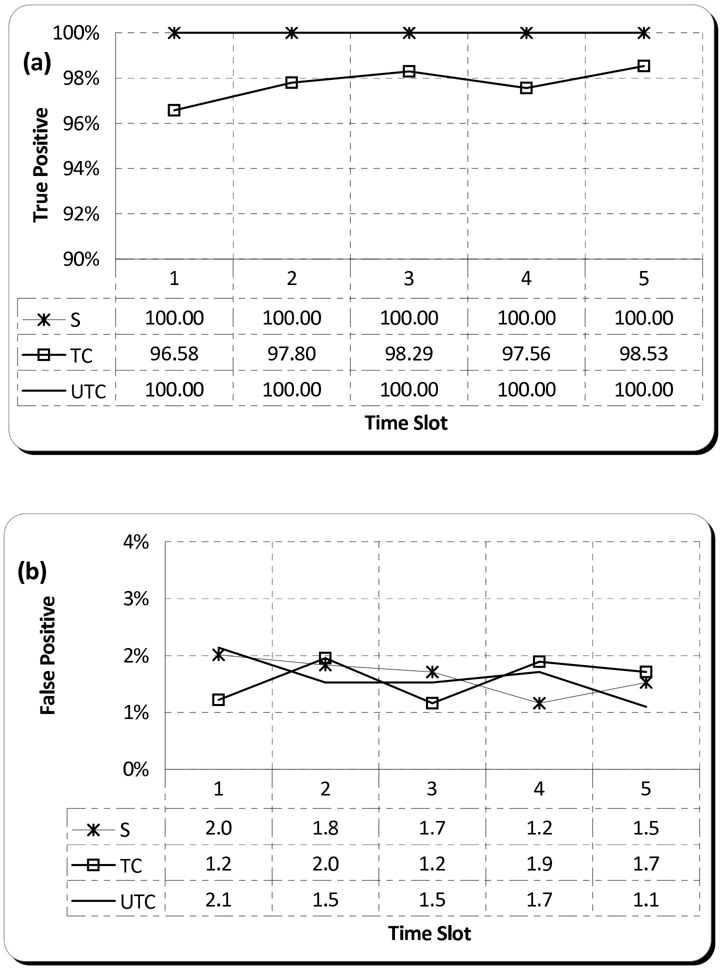


Figure 2: Detection tolerance using 20% free riders: (a) Free riders detected as free riders and (b) Non-free riders misdetected as free riders.

The detection rate for 60% and 80% free riders are shown in Figure 4(a) and 5(a), respectively. For both Figures, the detection rate for S as well as UTC is 100%. This for the same reason explained for Figures 2(a) and 3(a). On the other hand, detecting the TC free riders is more difficult since they change their behavior in a flip flop manner. This becomes apparent in the case of 80% free riders where we notice that their detection rate falls to about 91.03%.

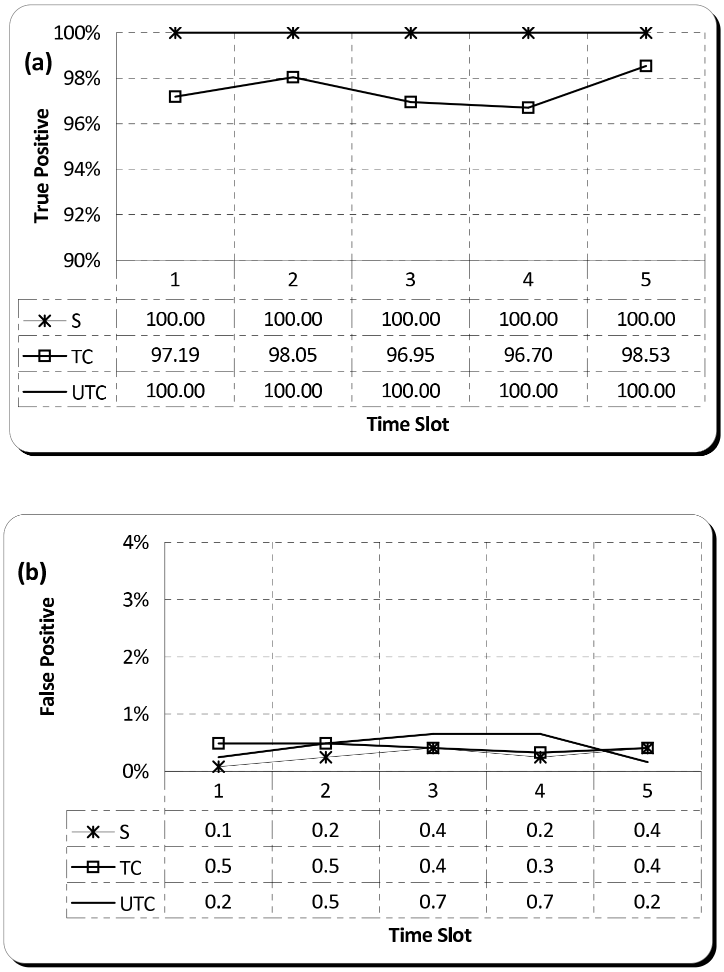
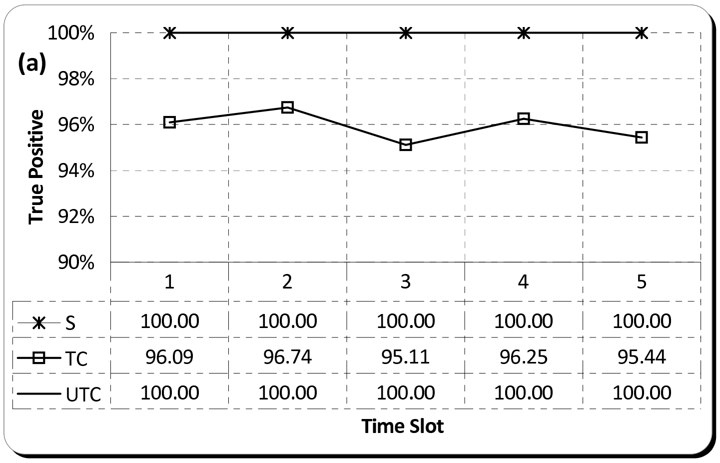


Figure 3: Detection tolerance using 40% free riders: (a) Free riders detected as free riders and (b) Non-free riders misdetected as free riders.

6.2 Free Riders Impact

In this set of experiments, we examine the effect of free riders on the decision making process made by a source peer. We focus on the UTC free riders because their impact is the worst since this type of free riders are injecting untrustworthy content and trying to pollute the network. As such, they infect the environment with malicious content for nefarious motives.

In Figure 6(b), the committed untrustworthy transactions are directly related



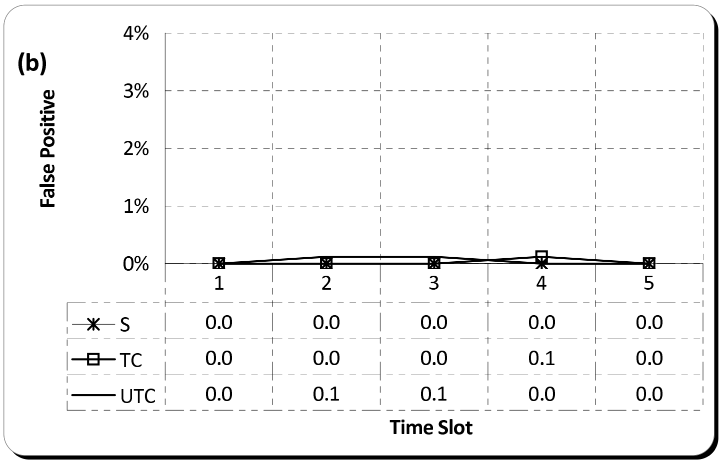


Figure 4: Detection tolerance using 60% free riders: (a) Free riders detected as free riders and (b) Non-free riders misdetected as free riders.

to the detection rate shown in the previous Figures. As shown in Figures 2(a), 3(a), 4(a), the T+ for UTC is 100%. This means that all free riders are detected and hence no untrustworthy transaction is committed. This fact is confirmed in Figure 6(b).

In Figure 6(a), the lost trustworthy transactions are also directly related to the misdetection rate The committed untrustworthy transactions are 0 as shown in Figure 6(b) because the untrustworthy transactions will only happen if a

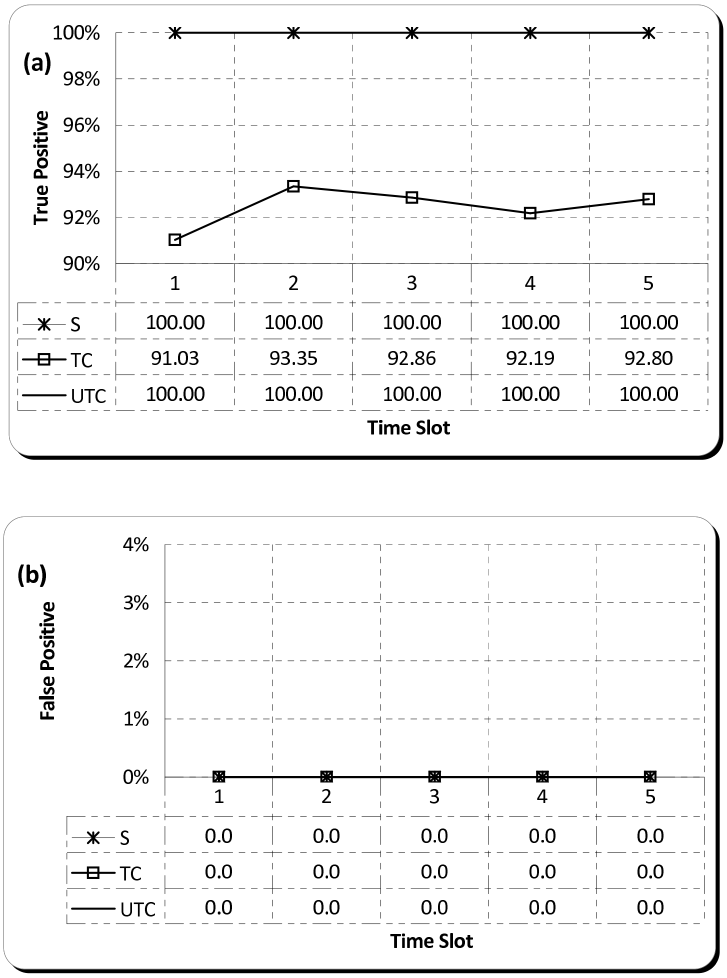


Figure 5: Detection tolerance using 80% free riders: (a) Free riders detected as free riders and (b) Non-free riders misdetected as free riders.

source peer transacts with a UTC free rider. A source peer will only transact with peers having appraisals greater than 0.66 of the previous Δ. The untrustworthy contributors' appraisals are never going to increase and no source peer will commit a transaction with them.

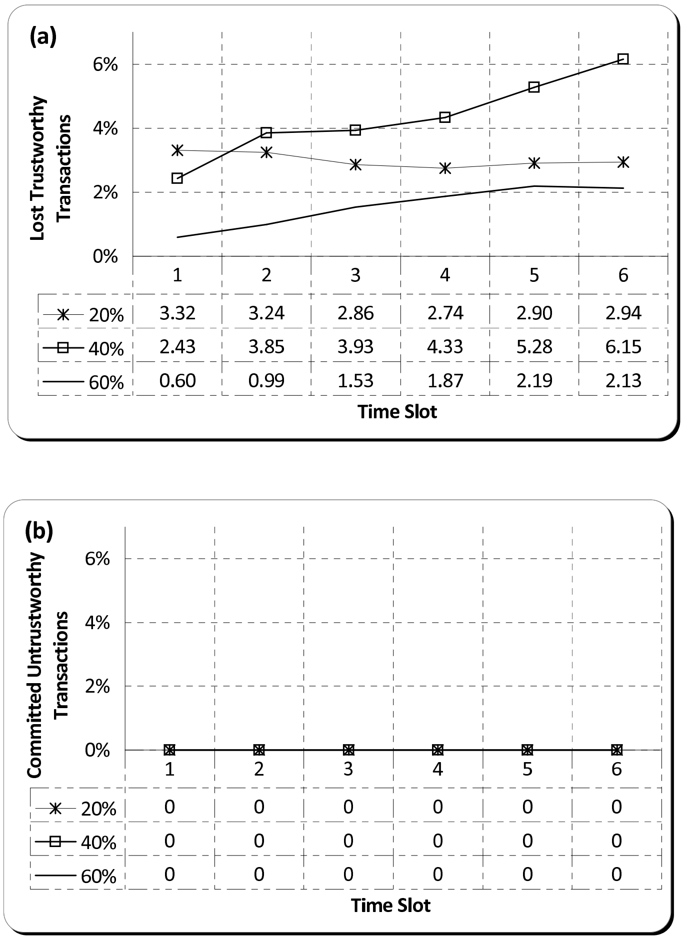


Figure 6: Free riders impact: (a) Lost trustworthy transactions and (b) Committed untrustworthy transactions.

7 Conclusions and Future Work

Current multimedia applications use centralized architectures. To deploy multimedia applications on distributed architectures such as P2P systems, many issues need to be resolved. In this paper, we investigated the free riding problem. Currently, free riding is tackled based on tit-for-tat strategies. With such strategies, an untrustworthy free rider, who consumes but injects malicious content, is undetected. This leads to the corruption and degradation of the distributed multimedia applications because peers will refrain from using the environment due to the lack of trust. In an ideal environment, leechers should be encouraged to become seeders so that a pure P2P environment will be established where almost every peer acts as a client and a server. Hence, bottleneck points due to centralization are reduced and the performance of multimedia applications improves as shown in our performance evaluation.

We argue that trustworthiness is a crucial attribute for free riders since an untrustworthy free rider can maintain its contribution factor by inflecting damage to the P2P community with its malicious content. Therefore, such untrustworthy contributors should be considered as free riders because they do not contribute but actually pollute and discourage contribution.

We proposed a novel free riders' filtering algorithm that detects selfish, trustworthy and untrustworthy free riders. Our simulation results show that the proposed algorithm identifies most of free riders even if the free riders are 80% of the whole pupolation. The results also show that our algorithm reduces the impact of free riders by minimizing the lost trustworthy transactions and the committed untrustworthy transactions. For future work, we are working on injecting dishonest free riders that lie. By considering honesty, we can evaluate our algorithm's resilience to free riders badmouthing or colluding.

Acknowledgment

The author would like to thank all anonymous reviewers for their helpful comments, resulting in a significant improvement in the quality of this manuscript. This research is supported by King Fahd University of Petroleum and Minerals under Research Grants IN070365 and IN080430.

References

[Aberer et al. 2005] Aberer, K., Datta, A., Hauswirth, M.: "P-Grid: Dynamics of self-organization processes in structured P2P systems"; Proc. Peer-to-Peer Systems and Applications, Lect. Notes in Comp. Sci. 3845, Springer, Berlin. 2005.

[Anderson and Reich 2000] Anderson, K., Reich, S. (eds.): "Proceedings of the Second Workshop on Structural Computing"; Lect. Notes Comp. Sci. 1903, Springer, Berlin.

[Azzedin and Ridha 2010] Azzedin, F., Ridha, A: "Feedback Behavior and Its Role in Trust Assessment for Peer-to-Peer Systems"; Telecommunication Systems (Jan 2010), DOI: 10.1007/s11235-009-9263-9

[Buragohain et al. 2003] Buragohain, C., Agrawal, D., Suri, S.: "A Game Theoretic Framework for Incentives in P2P Systems"; Proc. the 3rd International Conference on Peer-to-Peer Computing, 2003.

[Burmester et al. 2006] Burmester, M., Van Le, T., de Medeiros, B.: "Towards Provable Security for Ubiquitous Applications"; Information Security and Privacy. Springer Berlin, Heidelberg. (June 2006). 295-312.

[Canberra] National Museum of Australia: Canberra. `http://www.nma.gov.au/index.html`.

[Chatzidrossos and Fodor 2008] Chatzidrossos, I., Fodor, V.: "On the effect of free-riders in P2P streaming systems"; Telecommunication Networking Workshop on QoS in Multiservice IP Networks,(Feb 2008).

[Chen et al. 2007] Chen, H., Wu, H., Zhou, X., Gao, C.: "Reputation-based Trust in Wireless Sensor Networks"; the International Conference on Multimedia and Ubiquitous Engineering (MUE07), 2007.

[Christodoulakis et al. 1999] Christodoulakis, D., Vaitis, M., Papadopoulos, A., Tzagarakis, M.: "The Callimachus approach to distributed hypermedia"; Proc. 10^{th} ACM Conf. Hypert., ACM, New York (Feb 1999).

[Fodor and Dan 2007] Fodor, V., Dan, G: "Resilience in live peer-to-peer streaming"; Communications Magazine 45, 6. 2007. 116-123.

[Hwang and Yuan 2007] Hwang, Y., Yuan, S: "A Privacy-Aware Identity Design for Exploring Ubiquitous Collaborative Wisdom"; Computational Science. Springer Berlin, Heidelberg. 2007. 433-440.

[Karakaya et al. 2008] Karakaya, M., Korpeoglu, I., Ulusoy, O.: "Counteracting Free Riding in Peer-To-Peer Networks"; Computer Networks 52, 3 (Feb 2008), 675-694.

[Kristiansson 2008] Chatzidrossos, I., Fodor, V.: "On Technologies for Ubiquitous Multimedia Communication"; PhD Thesis, Lulea University of Technology, Sweden, (2006).

[Live broadcasting] Parliament of Australia: Live broadcasting. `http://webcast.aph.gov.au/livebroadcasting/`.

[Magharei and Rejaie 2007] Magharei, N., Rejaie, R.: "Mesh or multiple-tree: A comparative study of live p2p streaming approaches"; Proc. INFOCOM Conf., ACM, Anchorage, Alaska (Feb 2007). 142-1432.

[Maille and Toka 2008] Maille, P., Toka, L.: "Managing a Peer-to-Peer Data Storage System in a Selfish Society"; IEEE Journal on In Selected Areas in Communications 26, 7 2008, 1295-1301.

[Mihaela et al. 2008] Mihaela S., Deepak S. T., Ritesh S.: "Stochastic Optimization for Content Sharing in P2P Systems"; IEEE TRANSACTIONS ON MULTIMEDIA 1, 1 (Jan 2008).

[Mol et al. 2008] Mol, J., Pouwelse, J., Memulpolder, M., Epema, D., Sips, H.: "Give-to-Get: free-riding-resilient video-on-demand in P2P systems"; Proc. Multimedia Computing and Networking 2008, San Jose (Jan 2008).

[Obele et al. 2009] Obele, B., Ukaegbu, A., Kang, M.: "On Tracking Free-Riders in P2P Networks"; Proc. the 11th International Conference on Advanced Communication Technology, 2009.

[OrbNet] Orb Networks, Inc. `http://www.orb.com/`.

[Pianese et al. 2007] Pianese, F., Perino, D., Keller, J, Ernest, B.: "PULSE: An Adaptive, Incentive-Based, Unstructured P2P Live Streaming System"; Transactions on Multimedia 9, 8. 2007.

[pplive] PPLive. `http://www.pplive.com/`.

[ppst] ppstream TV. `http://www.ppstream.com/`.

[Shah and Paris 2007] Shah, P., Paris, F.: "Peer-to-Peer Multimedia Streaming Using BitTorrent"; the IEEE Performance, Computing, and Communications Conference, (April 2007) 340-347.

[Silverston et al. 2008] Silverston, T., Fourmaux, O., Crowcroft, J.: "Towards an incentive mechanism for peer-to-peer multimedia live streaming systems"; the Eighth International Conference on Peer-to-Peer Computing, (Feb 2008).

[sopcast] SopCast. `http://www.sopcast.com/`.

[Tang et al. 2004] Tang, Y., Wang, H., Dou, W.: "Trust based incentive in P2P network"; the IIEEE International Conference on E-Commerce Technology for Dynamic E-Business, (2004) 302-305.

[Ubiq 2006] Ubiquitous Web Applications Working Group. `http://www.w3.org/2006/10/uwa-charter.html`.

[Xiaojun et al. 2007] Xiaojun, H., Liang, C., Liang, J, Liu, Y., Ross, K.: "A Measurement Study of a Large-Scale P2P IPTV System"; Transactions on Multimedia 9, 8. 2007.

[Xie et al. 2008] Xie, C., Chen, G., Vandenberg, A., Pan, Y.: "Analysis of Hybrid P2P Overlay Network Topology"; Computer Communications 31, 2 (Feb. 2008), 190-200.