Um Algoritmo de Classificação Automática de Pares Desonestos para Comunidades Privadas BitTorrent

Pedro Gustavo de Farias Paiva, Paulo André da S. Gonçalves

Centro de Informática (CIn)
Universidade Federal de Pernambuco (UFPE)
50.740-560 – Recife – PE – Brasil

{pgfp,pasg}@cin.ufpe.br

Abstract. *The utilization of incentives mechanisms is one remarkably feature of P2P BitTorrent systems. Inside the BitTorrent Ecosystem exists private communities which employ an additional incentive mechanism called SRE (Share Ratio Enforcement). Community users must maintain their share ratio above the SRE otherwise they could be banned from private network. However, some users can use malicious BitTorrent client to manipulate their share ratio to keep it always above network SRE. This paper proposes an algorithm to automatically identify malicious users that are spoofing their share ratios in a private BitTorrent community. The proposed algorithm is evaluated using a real private network controlled under various scenarios. The results shown that the proposed algorithm is able to identify and classify malicious users with significant accuracy, showing an hit rate of 100% in identifying rogue peers in most evaluated scenarios. In the worst case, the hit rate was 83%.*

Resumo. *Uma das características principais dos sistemas P2P BitTorent é a utilização de mecanismos de incentivo para estimular o compartilhamento de arquivos. Dentro de um ecossistema BitTorrent existem comunidades privadas que utilizam um mecanismo de incentivo adicional conhecido por SRE (Share Ratio Enforcement) ou Taxa de Compartilhamento Imposta. Todo usuário deve manter sua Taxa de Compartilhamento (TC) acima da SRE da rede para não ser banido da mesma. Contudo, os usuários da rede podem usar um cliente BitTorrent malicioso que é capaz de manipular a TC para que a mesma seja sempre maior do que a SRE da rede. Este artigo propõe um algoritmo para classificação automática de pares numa rede privada BitTorrent que permite ao rastreador da rede identificar pares desonestos manipulando sua TC. O algoritmo proposto é avaliado usando uma rede privada real sob diversos cenários controlados. Os resultados obtidos mostram uma taxa de acerto de 100% na identificação de pares desonestos na maioria dos cenários estudados. No pior caso encontrado, a taxa de acerto foi de 83%.*

1. Introdução

Uma aplicação P2P pode ser definida como um sistema distribuído composto por pares (*peers*) transientes que se organizam automaticamente em uma rede lógica sobreposta a uma rede física para compartilhar recursos. Esse modelo de distribuição é atraente porque todo o conteúdo é transferido diretamente entre os pares que compõem o sistema, sem

passar por servidores de terceiros. As redes BitTorrent são aplicações P2P que se caracterizam por sua popularidade, eficiência, diversidade de conteúdos compartilhados e por sua grande parcela de contribuição com o tráfego total da Internet [Mansilha et al. 2010].

Um ecossistema BitTorrent é formado basicamente por três componentes principais: os pares, os mecanismos de organização da rede e os *sites* para descoberta de *torrents*. Esses *sites* podem ser públicos [Cohen 2003] ou privados (*Darknets*) [Zhang et al. 2010]. As comunidades BitTorrent privadas provêem um ambiente fechado onde seus usuários precisam se autenticar em uma rede restrita para terem acesso ao conteúdo almejado. Além disso, essas redes utilizam um mecanismo de incentivo auxiliar denominado SRE (*Share Ratio Enforcement*) ou Taxa de Compartilhamento Imposta. Esse mecanismo motiva os usuários a manterem uma taxa de compartilhamento (*i.e.* razão *total de dados enviados/total de dados recebidos*) acima de um determinado limiar imposto, o qual é conhecido por SRE da rede [Chen et al. 2010].

O mecanismo SRE é importante nas comunidades privadas BitTorrent, pois é utilizado para banir da rede pares que possuam uma taxa de compartilhamento menor do que a SRE da rede. Dessa forma, o mecanismo procura garantir que os usuários mantenham um certo nível de contribuição na comunidade. Devido à importância do mecanismo SRE, existe o interesse em ataques contra ele que permitem a um cliente malicioso manter, artificialmente, uma taxa de compartilhamento maior do que o limiar imposto para a comunidade. Exemplos de ataques com tal objetivo incluem os conluios [Lian et al. 2007], [Liu et al. 2010], [Ciccarelli and Cigno 2011] e a falsificação de relatórios [Liu et al. 2010].

Na Internet, existem diversos clientes BitTorrent modificados, como o *RatioMaster*[1], que falsificam relatórios com o intuito de permitir a manipulação da taxa de compartilhamento. Nesse caso, o cliente envia relatórios com informações adulteradas sobre o total de dados enviados e recebidos a fim de obter uma taxa de compartilhamento maior do que o limiar imposto na comunidade. Este artigo apresenta um estudo sobre uma comunidade privada BitTorrent formada por um rastreador com a implementação *Xbtit*[2], pares BitTorrent honestos *uTorrent*[3] e pares maliciosos que usam a ferramenta *RatioMaster* para falsificar seus relatórios, inflando artificialmente a taxa de compartilhamento. A partir desse estudo, é proposto um algoritmo que permite ao rastreador analisar enxames e identificar automaticamente os pares maliciosos nesses enxames. O algoritmo proposto é avaliado sob diversos cenários controlados. Os resultados obtidos mostram uma taxa de acerto de 100% na identificação de pares desonestos na maioria dos cenários estudados. A principal contribuição deste artigo está na proposição do classificador automático e na avaliação de sua taxa de acerto, de falsos positivos e de falsos negativos nos cenários estudados.

O restante deste artigo está organizado como segue: a Seção 2 apresenta os trabalhos relacionados. A Seção 3 apresenta a comunidade BitTorrent privada utilizada para o estudo do comportamento de pares desonestos. A Seção 4 apresenta o classificador proposto e avalia o desempenho do mesmo. Finalmente, a Seção 5 apresenta as considerações finais e trabalhos futuros.

[1]http://ratiomaster.net - Último acesso em 29/03/2012

[2]http://www.xbtit.com - Último acesso em 29/03/2012

[3]http://www.utorrent.com - Último acesso em 29/03/2012

2. Trabalhos Relacionados

O estudo das comunidades privadas BitTorrent vem recebendo grande atenção na literatura [Zhang et al. 2010], [Chen et al. 2010], [Chen et al. 2011]. Diversos estudos relacionados a essas redes focam em seus mecanismos de incentivo [Andrade et al. 2005], [Liu et al. 2010], [Jia et al. 2011b], [Jia et al. 2011a].

Em [Andrade et al. 2005], é apresentado um estudo sobre o mecanismo de incentivo SRE em uma comunidade privada BitTorrent conhecida por *easytree.org*. Em tal estudo, são apresentadas evidências de que a utilização do mecanismo permitiu um aumento na colaboração entre os usuários dessa comunidade. Em [Jia et al. 2011a], é apresentado um modelo teórico que permite analisar como o mecanismo SRE provê incentivos para que os pares cooperem mais e como o mecanismo melhora o desempenho de *download* na rede.

Em [Jia et al. 2011b], é apresentada uma análise sobre como o tempo em que o cliente fica semeando na rede afeta a SRE. Para isso, é proposto e utilizado um modelo teórico que permite prever a velocidade média de *download*, o tempo médio de semeação (*seeding*) e a utilização média da capacidade de *upload* dos pares. O estudo mostra que o mecanismo SRE discrimina pares com pouca banda passante, forçando-os a semear por muito mais tempo do que pares com maior capacidade de banda passante. Além disso, o estudo mostra que apesar do SRE aumentar a taxa de *download* na rede, ele força os pares, indiretamente, a semearem por longos períodos de tempo com capacidade de *upload* subutilizada. A partir dessas observações, foram propostas estratégias que permitem à rede manter uma boa taxa de *download* enquanto liberam os pares dos longos períodos de semeação e, ao mesmo tempo, são justas com pares com diferentes capacidades de banda passante.

Os diversos estudos existentes sobre o mecanismo SRE mostram que ele é efetivo para melhorar a taxa de *download* na rede, incentivando os pares a contribuírem mais com a comunidade para não serem banidos. Contudo, existe o interesse em ataques contra o mecanismo SRE que permitam a um cliente malicioso usufruir da rede sem colaborar, mas mantendo, artificialmente, uma taxa de compartilhamento maior do que o limiar imposto para a comunidade. Exemplos de ataques com tal objetivo incluem os conluios [Lian et al. 2007], [Liu et al. 2010], [Ciccarelli and Cigno 2011] e a falsificação de relatórios [Liu et al. 2010].

Na falsificação de relatório para burlar o mecanismo SRE, o cliente malicioso informa ao rastreador uma quantidade falsa de dados enviados e recebidos. Esse tipo de falsificação é relativamente simples e pode ser feita através do uso de clientes BitTorrent modificados como o *RatioMaster*[4], *Ratio Faker*[5], *Tracker Pro*[6] e *Torrent Ratio Keeper*[7]. Esses clientes sempre informam ao rastreador uma quantidade de dados enviados e recebidos tal que a taxa de compartilhamento seja maior do que a imposta na comunidade. Para identificar pares utilizando essa forma de trapaça, em [Liu et al. 2010] é proposta a utilização de relarórios "casados" *(pair-wise reports)*. Nessa proposta, cada par reporta ao rastreador o quanto foi *recebido de* e *enviado para* cada outro par da rede em vez de

[4]http://ratiomaster.net/ - Último acesso em 30/03/2012

[5]http://ratiofaker.blogspot.com/- Último acesso em 30/03/2012

[6]http://www.esanu.name/software/?p=9 - Último acesso em 30/03/2012

[7]http://www.torrentratiokeeper.com/tutorial/ - Último acesso em 30/03/2012

reportar simplesmente seu agregado de envios e recebimentos. Suponha que um *par A* tenha enviado verdadeiramente 1 MB para um *par B*, mas reporta que o envio foi de 20 MB. Nesse caso, o rastreador pode detectar uma inconsistência se o *par B* reportar verdadeiramente que recebeu 1 MB do *par A*. Os autores propõem então que os usuários frequentemente envolvidos em inconsistências sejam bloqueados.

Embora seja uma forma de se detectar pares desonestos na rede, nenhum estudo sobre a eficácia da proposta é apresentado pelos autores. Além disso, tal proposta requer que os pares da rede passem a reportar mais informações para o rastreador. Como consequência, é necessário alterar todos os clientes e há um aumento no consumo de recursos da rede, já que mais dados necessitam ser transmitidos. Outro ponto importante é o fato de que clientes legítimos podem ser indevidamente bloqueados quando vítimas de conluios.

Este artigo se diferencia dos trabalhos relacionados por estudar uma forma de identificação automática de pares maliciosos que burlam o mecanismo SRE em comunidades privadas. A forma de identificação tem como princípio, não requerer a utilização de relatórios casados e nem a adição de novas informações aos relatórios enviados pelos pares. Para desenvolver o classificador automático de pares maliciosos, este artigo foca em uma comunidade privada BitTorrent formada por um rastreador com a implementação *Xbtit*, pares BitTorrent honestos *uTorrent* e pares maliciosos que usam a ferramenta *RatioMaster* para falsificar seus relatórios, inflando artificialmente a taxa de compartilhamento. Com essa comunidade, o comportamento dos clientes maliciosos é observado e, a partir disso, é proposto e avaliado um classificador que permite ao rastreador analisar enxames e identificar automaticamente os pares maliciosos.

3. Avaliação Experimental da Taxa de Compartilhamento

Para o estudo proposto neste artigo, foi criada uma comunidade BitTorrent privada, a qual é doravante denominada *μMundo*. A seguir, são apresentados: o detalhamento do *μMundo*, os cenários de avaliação do comportamento de pares e os resultados obtidos.

3.1. O *μMundo*

A Figura 1 apresenta o *μMundo* utilizado para o estudo de uma rede privada BitTorrent. Essa rede é composta por 6 *notebooks*, 1 PC e um Roteador sem fio. Cada *notebook* representa um único cliente da rede conectado via IEEE 802.11g. O PC representa um único cliente conectado ao roteador via Ethernet 10 Mbps. Um dos *notebooks* executa uma máquina virtual que representa o rastreador da rede. Os pares honestos não foram configurados com limites para suas larguras de banda, assim, cada nó dispôs da banda oferecida pelo ponto de acesso da rede compatível com o padrão de rede sem fio 802.11g.

O servidor utiliza o sistema operacional *Linux* CentOS 5.3 *Server*, o banco de dados MySQL e o Apache 2.3 como servidor de aplicação WEB. A implementação do rastreador é feita com o Xbtit, desenvolvido pela *Btiteam* sob licensa BSD. Essa implementação é desenvolvida em PHP e possui integração opcional com C++ que pode ser utilizada caso seja necessária uma maior eficiência do servidor. O MySQL é utilizado pelo rastreador para armazenar os arquivos *.torrent* e informações como por exemplo: o nome dos usuários, a quantidade de "sementes" (*seeders*) e "sugadores" (*leechers*) por arquivo, as taxas de compartilhamento, a quantidade de arquivos existentes, dentre outras.

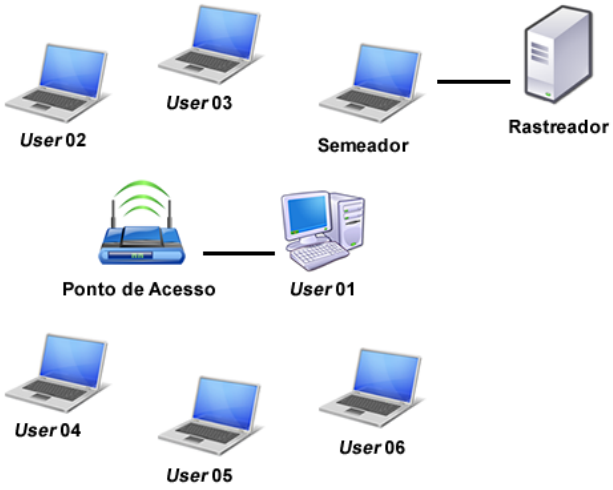


Figura 1. O *µMundo*.

A comunidade privada formada possui um usuário semeador. A função desse usuário é criar os arquivos *.torrent* e servir como primeira semente dos arquivos compartilhados. Esse usuário é o único que pode disponibilizar arquivos na rede. Os pares honestos foram implementados utilizando-se o cliente *uTorrent*. Tal cliente foi escolhido por ser bem difundido na Internet, por possuir fácil instalação, por não consumir muitos recursos computacionais e por não possuir custos de aquisição.

Um par desonesto ou malicioso é implementado com a utilização da ferramenta de domínio público *RatioMaster*. A *RatioMaster* permite executar clientes BitTorrent modificados que forjam, em seus relatórios, a quantidade de dados enviados e recebidos. Um cliente *RatioMaster* se conecta ao rastreador e age como um cliente BitTorrent normal porém, sem efetivamente trocar dados com outros pares que estão participando no enxame. Ele reporta ao rastreador que está enviando (ou recebendo ou ambos) dados a uma certa taxa, tornando-o útil para aumentar artificialmente a taxa de compartilhamento de clientes em redes que utilizam o mecanismo SRE.

3.2. Cenários Estudados

Para o estudo neste artigo, foram criados cenários que possibilitam a observação do comportamento dos pares através da métrica *share ratio* ou taxa de compartilhamento (TC), que é dada pela razão do total enviado sobre o total baixado para cada arquivo transferido. Essa métrica mede o nível de compartilhamento de cada participante de uma comunidade privada.

Com o intuito de compreender o comportamento de pares desonestos na rede privada definida para estudo, foram construídos 3 cenários de compartilhamento de arquivo. Em cada cenário, foram estudados 3 casos distintos referentes à quantidade de pares desonestos presentes na rede. A Tabela 1 apresenta os dados de cada cenário. Em todos os cenários há 7 pares e são estudados 3 casos:

- 1º Caso - 1 Usuário desonesto (*user01*);
- 2º Caso - 3 Usuários desonestos (*user01, user02, user03*);
- 3º Caso - Nenhum Usuário desonesto.

Nos cenários 01, 02 e 03 são compartilhados, respectivamente, um arquivo de 250 MB, 500 MB e 1024 MB.

Tabela 1. Dados dos cenários estudados.

	Nº de Casos	Nº de Pares	Tamanho do Arquivo (MB)
Cenário 01	3	7	250
Cenário 02	3	7	500
Cenário 03	3	7	1024

A ferramenta *RatioMaster* requer que cada cliente desonesto tenha suas taxas de *download* e *upload* preconfiguradas. A Tabela 2 apresenta a configuração da largura de banda dos usuários *User01*, *User02* e *User03* quando os mesmos possuíam comportamento desonesto nos cenários avaliados.

Tabela 2. Largura de banda de usuários quando desonestos.

	Nome do usuário	*Upload* (kbps)	*Download* (kbps)
Usuário 01	*user01*	800	1000
Usuário 02	*user02*	800	256
Usuário 03	*user03*	128	256

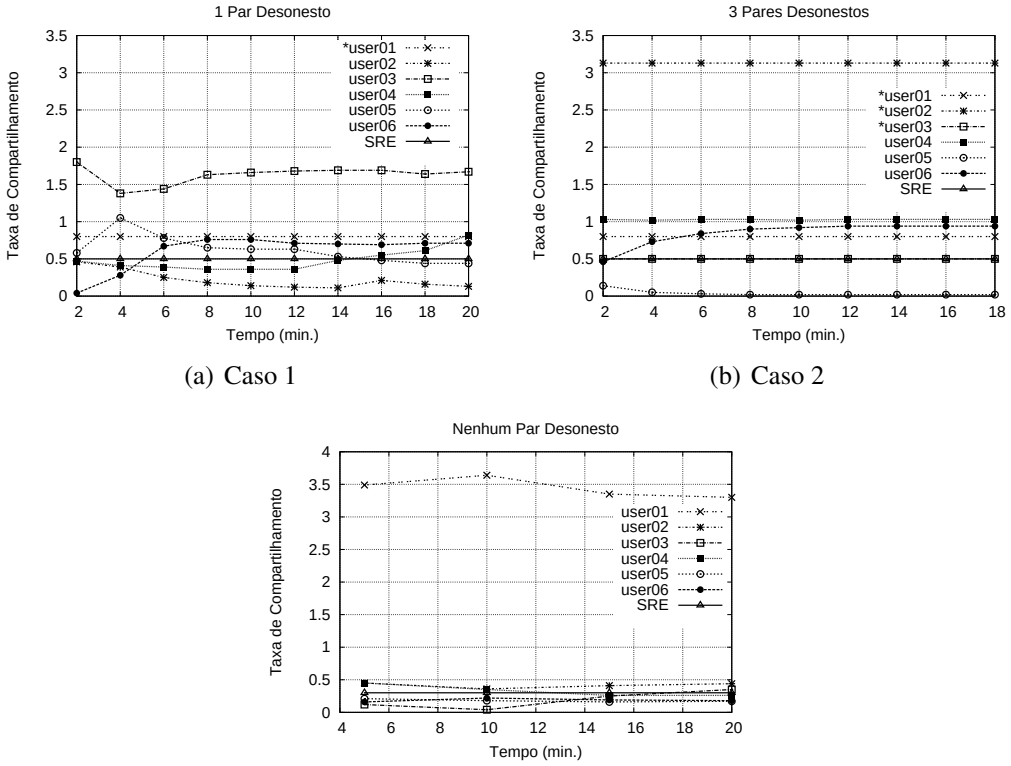
3.3. Avaliação do Comportamento dos Pares

Esta seção estuda o comportamento da taxa de compartilhamento dos pares da rede privada BitTorrent definida. O rastreador coleta as informações necessárias em sua base de dados de tempos em tempos. A coleta de dados se inicia alguns minutos após o *Announce* do rastreador e termina quando todos os pares no enxame se tornam semeadores. A SRE da rede é preestabelecida e seu valor é informado na descrição dos resultados.

A Figura 2 apresenta a taxa de compartilhamento de todos os usuários nos 3 casos estudados pra o cenário 01. A Figura 2(a) mostra que o usuário desonesto (*User01*) está configurado com uma taxa de compartilhamento de 0,8 enquanto a taxa de compartilhamento imposta é de 0,5. Esse exemplo mostra que o usuário consegue burlar o mecanismo SRE mesmo sem estar de fato compartilhando na rede. Os usuários honestos (*User02* e *User05*) terminaram suas transferências com suas respectivas TCs abaixo da SRE estabelecida. Note que a TC de todos os usuários honestos apresenta variabilidade perceptível ao longo do tempo enquanto a TC do usuário desonesto é aparentemente fixa.

A Figura 2(b) mostra que os 3 usuários desonestos (*User01*, *User02* e *User03*) possuem configuração que lhes permite obter uma TC maior ou igual a SRE da rede que é de 0,5. Note que o comportamento da TC desses usuários é aparentemente fixo. Observa-se ainda uma variabilidade na TC dos usuários honestos *User05* e *User06* enquanto a variabilidade da TC do usuário honesto *User04* é praticamente imperceptível graficamente. A Figura 2(c) mostra que no cenário sem pares desonestos, a TC de todos eles apresentou variabilidade perceptível.

A Figura 3 apresenta o comportamento da TC dos usuários para os casos 1,2 e 3 no cenário 02. A diferença para o cenário anterior está no aumento do tamanho do aquivo compartilhado para obtenção de uma maior quantidade de amostras. A Figura 3(a) mostra que o usuário desonesto (*User01*) possui TC maior que a taxa SRE da rede. Apenas



(a) Caso 1

(b) Caso 2

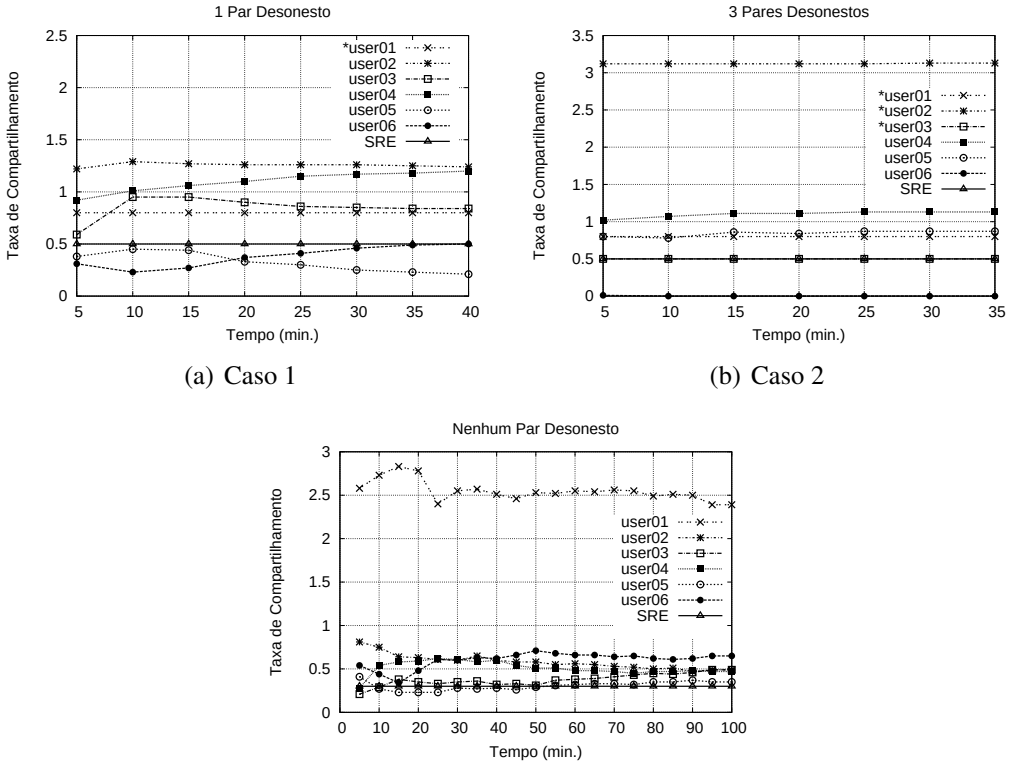
(c) Caso 3

Figura 2. Comportamento da TC para os casos 1, 2 e 3 no Cenário 01.

o usuário honesto *User05* termina suas transferências com TC abaixo da SRE estabelecida. Contudo, note que todos os usuários honestos apresentam TC com variabilidade perceptível ao longo do tempo enquanto a TC do usuário desonesto é aparentemente fixa. A Figura 3(b) mostra que os 3 usuários desonestos continuam com TC aparentemente fixa apesar do aumento na quantidade de amostras. Observa-se ainda a variabilidade na TC dos usuários honestos *User04* e *User05* enquanto a variabilidade da TC do usuário honesto *User06* é praticamente imperceptível graficamente. A Figura 3(c) mostra que no cenário sem pares desonestos, a TC de todos eles apresentou variabilidade perceptível.

A Figura 4 apresenta o comportamento da TC dos usuários para os casos 1,2 e 3 no cenário 03. Dessa vez, é feito o compartilhamento de um arquivo de 1024 MB para permitir aumentar ainda mais o número de amostras de TC ao longo do tempo. A Figura 4(a) mostra que apenas o usuário honesto *User02* termina suas transferências com TC abaixo da SRE estabelecida e todos os usuários honestos apresentam TC com variabilidade perceptível ao longo do tempo. Já a TC do usuário desonesto (*User01*) é aparentemente fixa. A Figura 4(b) mostra que os 3 usuários desonestos continuam com TC aparentemente fixa apesar do aumento na quantidade de amostras. Observa-se ainda uma variabilidade na TC dos usuários honestos *User04*, *User05* e *User06*. A Figura 4(c) mostra que no cenário sem pares desonestos, a TC de todos eles apresenta variabilidade perceptível.

Os resultados expostos indicam que a falta de variabilidade da TC de um usuário em um enxame pode ser um indicativo de que ele é desonesto. Contudo, houve cenários



(a) Caso 1

(b) Caso 2

(c) Caso 3

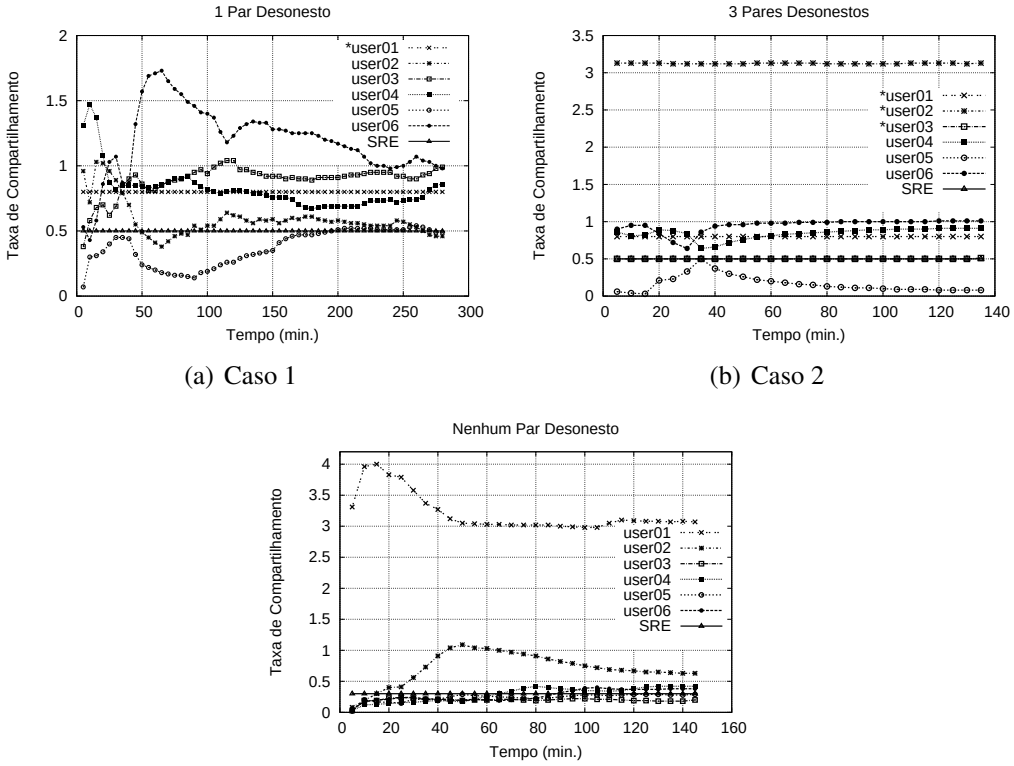
Figura 3. Comportamento da TC para os casos 1, 2 e 3 no Cenário 02.

com usuários honestos apresentando uma TC com variabilidade imperceptível graficamente. Isso motiva uma investigação mais detalhada sobre as informações que são enviadas pelo cliente malicioso (*RatioMaster*) e posteriormente tratadas pela implementação do rastreador *Xbtit*. Para isso, os experimentos do cenário 03 foram executados mais uma vez, porém usando como pares desonestos, os usuários *User02*, *User03* e *User04*. A Tabela 3 mostra o conteúdo dos nove primeiros relatórios enviados pelo *User03* ao servidor de compartilhamento e qual seria a sua TC representada com várias casas decimais. Note que há uma pequena variabilidade numérica na TC a partir da terceira casa decimal somente.

Tabela 3. Dados informados pelo usuário desonesto *user03* e sua TC.

	Enviados	**Baixados**	**TC**
Relatório 01	9,1717632E7	2,9341424E7	3,1258752813087733
Relatório 02	1,83484416E8	5,870008E7	3,12579499039865
Relatório 03	3,21110016E8	1,02740816E8	3,1254376644234556
Relatório 04	4,12844032E8	1,32098112E8	3,125283365139995
Relatório 05	5,04610816E8	1,61459696E8	3,1253051287796305
Relatório 06	5,96361216E8	1,90813808E8	3,1253567142268865
Relatório 07	6,88128E8	2,20195376E	3,125079247804005
Relatório 08	7,79862016E8	2,495688E8	3,124837784210206
Relatório 09	2,495688E8	2,20195376E	3,124837784210206

A Tabela 4 mostra a TC dos usuários desonestos *User02*, *User03* e *User04* calculada na implementação *Xbtit* do rastreador e a TC calculada truncada na terceira casa



(a) Caso 1

(b) Caso 2

(c) Caso 3

Figura 4. Comportamento da TC para os casos 1, 2 e 3 no Cenário 03.

decimal. A primeira é representada pela nomenclatura **Tr** enquanto a segunda é representada pela nomenclatura **S**. O valor da TC calculada pelo rastreador é representada com duas casas decimais e apresenta uma variabilidade máxima de apenas 0,01. Isso acontece, pelo seguinte: um cliente *RatioMaster* informa em seus relatórios sucessivos quantidades variadas de recebimentos e envios. Mesmo assim, a TC calculada considerando várias casas decimais possui pouca variabilidade (apenas a partir da terceira casa). Por outro lado, a implementação *Xbit* calcula a TC usando a regra *Round Half Up* com apenas duas casas decimais. Esses dois fatores, fazem com que a variabilidade máxima registrada na TC de um par desonesto seja de apenas 0,01.

Tabela 4. Comportamento da TC de 3 usuários desonestos.

	User02 (Tr)	**User02 (S)**	**User03 (Tr)**	**User03 (S)**	**User04 (Tr)**	**User04 (S)**
Relatório 01	3,125	3,13	0,498	0,50	0,999	1,00
Relatório 02	3,125	3,13	0,499	0,50	0,999	1,00
Relatório 03	3,125	3,13	0,499	0,50	0,999	1,00
Relatório 04	3,125	3,13	0,499	0,50	0,999	1,00
Relatório 05	3,125	3,13	0,499	0,50	0,999	1,00
Relatório 06	3,125	3,13	0,499	0,50	0,999	1,00
Relatório 07	3,125	3,13	0,499	0,50	0,999	1,00
Relatório 08	3,124	3,12	0,499	0,50	1,000	1,00

4. O Classificador Proposto

Os estudos apresentados na seção anterior fornecem indicativos sobre a viabilidade de se classificar pares em honestos e desonestos através da análise da variabilidade de suas TCs

ao longo do tempo enquanto participam de enxames. Esta seção propõe um classificador de pares com base nessa observação. O Algoritmo 4.1 apresenta o pseudo código do classificador proposto. O algoritmo parte do pressuposto de que todos os usuários do enxame analisado são desonestos e então inicia a verificação da taxa de compartilhamento de cada usuário conectado. A partir do momento em que a TC ultrapassa uma certa variabilidade, o algoritmo aceita o par como honesto.

O algoritmo calcula para cada duas amostras sucessivas de TC, no tempo, o módulo da diferença entre as subrretas formadas por elas. Assim, y_c é dado por:

$$y_c = \Delta tc \,, \tag{1}$$

onde Δtc é a variação da TC para cada usuário coletada em um determinado intervalo de tempo entre as duas amostras sucessivas. É esperado que y_c tenha uma variação muito próxima de 0 (zero) para todo usuário desonesto, o que representa o comportamento esperado para os pares que se utilizam de um cliente malicioso. Assim, essa variação é calculada para cada amostra do usuário e armazenada em y. Por fim, é verificado se a diferença entre y_c e y (nos pontos analisados) é maior que um limiar ε. Caso a diferença seja maior do que o limiar, o par analisado será classificado com honesto (linhas 15 e 16).

Algoritmo 4.1 Pseudo Código do Algoritmo Classificador.

```
1:  users ← returnUsers();
2:  epsilon ← 0.01;
3:  for i ← 0 to users.size() − 1 do
4:      samples ← returnUserSamples(users[i]);
5:      shareRatioBefore ← null; y_c ← null;
6:      isColluder ← true;
7:      for j ← 0 to samples.size() − 1 do
8:          shareRatio ← samples[j].getShareRatio();
9:          if shareRatioBefore¬null then
10:             if y_c == null then
11:                 y_c ← shareRatio − shareRatioBefore;
12:             else
13:                 y ← shareRatio − shareRatioBefore;
14:                 diff = |y − y_c|;
15:                 if diff > epsilon then
16:                     isColluder ← false;
17:                 shareRatioBefore ← shareRatio;
18:                 y_c ← y;
19:         else
20:             shareRatioBefore ← shareRatio;
```

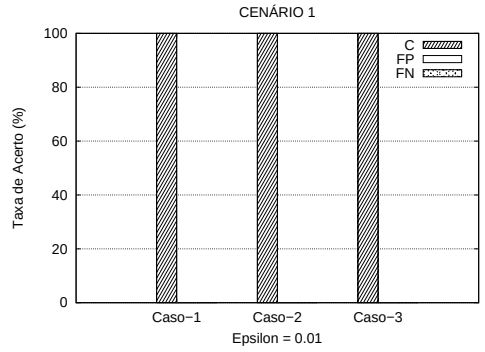
4.1. Taxa de Acertos, Falsos Positivos e Falsos Negativos

Para medir a acurácia de um classificador é importante que seja feita uma avaliação sobre a taxa de acerto, falsos positivos e falsos negativos. Dessa forma, é possível observar qual a porcentagem de usuários desonestos que foram classificados corretamente e, além disso, verificar o quão eficiente é o classificador proposto. No estudo desta seção, um falso positivo ocorre quando um usuário honesto é classificado como desonesto. Já um falso negativo ocorre quando um usuário desonesto é classificado como honesto. A taxa de acerto (TA) utilizada para a avaliação do classificador proposto é dada pela razão abaixo, onde: TP é a quantidade de verdadeiros positivos e TN, o total de verdadeiros negativos.

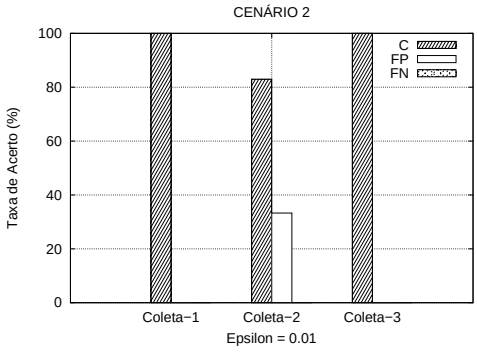
$$TA = \frac{(TP + TN)}{Total\ de\ Pares} \tag{2}$$

Os resultados obtidos foram dispostos em histogramas que representam a porcentagem da taxa de acerto, de falsos positivos e de falsos negativos. Nas Figuras 5(a), 5(b) e 5(c), a porcentagem de pares classificados corretamente é representada pela letra C. A porcentagem de falsos positivos é representada através da sigla FP e a porcentagem de falsos negativos é representada pela sigla FN.

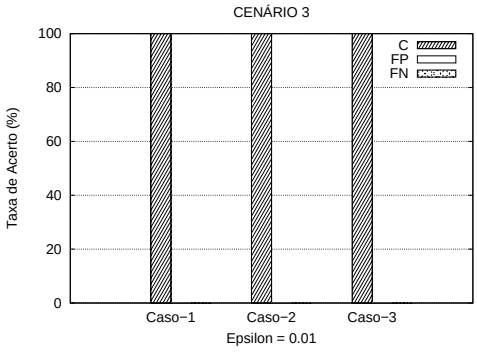
O algoritmo proposto obteve uma taxa de acerto de 100% para todos os casos dos cenários (1 e 2) representados na Figuras 5(a) e 5(c). Isso indica que para tais cenários o algoritmo proposto identificou os pares desonestos corretamente sem que FP_s fossem levantados. Ainda na Figura 5(a), é importante observar que mesmo com o aumento da quantidade de usuários desonestos, o algoritmo proposto continuou sem apresentar falsos positivos e os pares desonestos continuaram sendo identificados com sucesso.



(a) TA=100% para todos os casos.



(b) TA=100% para os casos 1 e 3.



(c) TA=100% para todos os casos.

Figura 5. Taxas de acertos, Falsos Positivos e Falsos Negativos para ε = 0,01.

Apenas um caso do Cenário 2 apresentou falso positivo. Para esse caso específico, a taxa de acerto do algoritmo proposto foi de 83%, o que indica que os pares desonestos foram identificados, porém, um usuário (o par *User06*) foi indevidamente classificado como desonesto. Esse falso positivo pode ser explicado através de uma pequena quantidade de dados enviados (*upload*) pelo par no decorrer da transferência. Esse fato fez com que sua TC, na prática, fosse zero. Observando as Figuras 3(a) e 3(b) nota-se que

todos os pares classificados indevidamente como desonestos apresentaram TC com baixa variabilidade.

Para todas as coletas do Cenário 3, o algoritmo proposto obteve uma taxa de acerto de 100%. Assim, todos os pares desonestos e honestos foram corretamente classificados. Analisando os gráficos das Figuras 5(a), 5(b) e 5(c), é possível observar que não ocorreram falsos negativos em nenhum dos casos estudados. Isso é importante, pois este resultado garante que, para os cenários estudados, pares desonestos não são classificados como honestos.

5. Conclusões

Este artigo focou numa comunidade privada BitTorrent formada por um rastreador com a implementação *Xbtit*, pares BitTorrent honestos *uTorrent* e pares desonestos que usam a ferramenta *RatioMaster* para falsificar seus relatórios, inflando artificialmente a taxa de compartilhamento. O comportamento dos pares dessa comunidade foi analisado e, a partir disso, foi proposto um classificador automático de pares. O classificador proposto se baseou na variabilidade da taxa de compartilhamento (TC) dos pares da rede para classificá-los como honestos ou desonestos.

Os resultados apresentados sugerem que o classificador proposto é eficiente na identificação de pares desonestos. Em dois dos três cenários estudados, a taxa de acerto (TA) foi de 100%. E em um único cenário, a taxa de acerto foi de 83% devido à ocorrência de um falso positivo. Em trabalhos futuros, serão efetuadas modificações no classificador proposto a fim de que os falsos positivos sejam reduzidos. Além disso, outras formas de manipulação da TC serão estudadas e levadas em consideração pelo classificador. Também serão estudados novos cenários com mais pares e arquivos sendo compartilhados.

Referências

Andrade, N., Mowbray, M., Lima, A., Wagner, G., and Ripeanu, M. (2005). Influences on Cooperation in Bittorrent Communities. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems (SIGCOMM)*, pages 111–115. ACM.

Chen, X., Chu, X., and Li, Z. (2011). Improving Sustainability of Private P2P Communities. In *Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, pages 1–6. IEEE.

Chen, X., Jiang, Y., and Chu, X. (2010). Measurements, Analysis and Modeling of Private Trackers. In *Proceedings of the Tenth International Conference on Peer-to-Peer Computing (P2P)*, pages 1–10. IEEE.

Ciccarelli, G. and Cigno, R. (2011). Collusion in Peer-to-Peer Systems. *Computer Networks*, 55(15):3517–3532.

Cohen, B. (2003). Incentives Build Robustness in BitTorrent. In *Proceedings of the 1st Workshop on the Economics of Peer-to-Peer Systems*.

Jia, A., D'Acunto, L., Meulpolder, M., and Pouwelse, J. (2011a). Modeling and Analysis of Sharing Ratio Enforcement in Private BitTorrent Communities. In *International Conference on Communications (ICC)*, pages 1–5. IEEE.

Jia, A., Rahman, R., Vinkó, T., Pouwelse, J., and Epema, D. (2011b). Fast Download But Eternal Seeding: The Reward and Punishment of Sharing Ratio Enforcement. In *International Conference on Peer-to-Peer Computing, (P2P)*, pages 280–289. IEEE.

Lian, Q., Zhang, Z., Yang, M., Zhao, B., Dai, Y., and Li, X. (2007). An Empirical Study of Collusion Behavior in the Maze P2P File-Sharing System. In *Proceedings of 27th International Conference on Distributed Computing Systems (ICDCS'07)*, page 56. IEEE.

Liu, Z., Dhungel, P., Wu, D., Zhang, C., and Ross, K. W. (2010). Understanding and Improving Ratio Incentives in Private Communities. *In Proceedings of the International Conference on Distributed Computing Systems*, pages 610–621.

Mansilha, R. B., Mezzomo, A., Facchini, G., Gaspary, L. P., and Barcellos, M. P. (2010). Observando o Universo BitTorrent Através de Telescópios. *Anais do XXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, 2010:233–246.

Zhang, C., Dhungel, P., Wu, D., Liu, Z., and Ross, K. W. (2010). BitTorrent Darknets. *Proceedings of 29th Annual Joint Conference on Computer Communications (IEEE/INFOCOM)*, pages 1–9.