Universidade Federal de Pernambuco

Centro de Informática

Pós-graduação em Ciência da Computação

DETECTANDO CLIENTES DESONESTOS EM COMUNIDADES PRIVADAS BITTORRENT

Pedro Gustavo de Farias Paiva

DISSERTAÇÃO DE MESTRADO

Recife

27 de Agosto de 2012

Universidade Federal de Pernambuco

Centro de Informática

Pedro Gustavo de Farias Paiva

DETECTANDO CLIENTES DESONESTOS EM COMUNIDADES PRIVADAS BITTORRENT

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: *Docteur Paulo André da S. Gonçalves*

Recife

27 de Agosto de 2012

Este trabalho é dedicado a minha família e a todas as pessoas que contribuíram para que ele se tornasse realidade.

AGRADECIMENTOS

Primeiramente, gostaria de agradecer a Deus pela vida, por minha família e pela oportunidade de conviver com pessoas tão especiais, que através de suas atitudes, me ensinaram o que realmente importa nessa jornada. Agradeço em especial aos meus pais, por terem me ensinado desde cedo a importância da educação e do trabalho. Procuro sempre retribuir o esforço de vocês aplicando no dia a dia tudo o que me foi ensinado. Agradeço a minha mãe por suas orações, que me trouxeram mais paz e tranquilidade durante todos esses anos. Agradeço também aos meus irmãos, Bruno e Paulo, pelo companheirismo constante. Espero sempre servir de exemplo apesar dos meus defeitos.

À Ednilson Leite, Samara Maia e Stael Maia por terem me acolhido como membro da família e pela torcida. Agradeço especialmente à Juliana Maia, pela ajuda e por se fazer presente todos os dias desses dois anos e 5 meses que estamos juntos. Ter você comigo foi imprescindível para a realização deste trabalho.

À Moisés Rodrigues, por ter plantado na minha cabeça a ideia de cursar um mestrado. Se não fosse você, talvez eu não tivesse atentado para essa oportunidade. Obrigado por se portar como um irmão desde sempre!

O amadurecimento adquirido no decorrer dessa pós-graduação não seria possível se não fosse a contribuição de vários professores, colegas de curso e amigos de longas datas. Gostaria de agradecer ao professor *Docteur* Paulo Gonçalves pela orientação, por sua seriedade e incansável busca pela qualidade científica. Por sempre me indicar os melhores caminhos, mesmo quando discordei ou preferi seguir outros. Finalmente, por sua paciência e conselhos valiosos no decorrer de todo o curso.

Aos professores, Doutor Carlos Ferraz, Doutor Djamel Sadok, Doutor Paulo Maciel, Doutora Ivana Fechine e Doutor Paulo Gonçalves, pelo conhecimento passado durante

as disciplinas do mestrado, que certamente me ajudaram a ter opinião formada sobre diversas áreas de pesquisa e temas diferentes além do que eu estava pesquisando.

Aos professores convidados para compor a banca de avaliação, professor Doutor Carlos Ferraz (Universidade Federal de Pernambuco, UFPE) e a professora Doutora Michele Nogueira (Universidade Federal do Paraná, UFPR), por dedicarem um pouco do seu tempo livre para que, com suas sugestões e correções, eu possa melhorar este trabalho.

À Universidade Federal de Pernambuco por me fornecer um ambiente de pesquisa composto por profissionais altamente qualificados e uma excelente infraestrutura para o desenvolvimento de projetos científicos.

Aos colegas que entraram comigo no mestrado Gaúcho (Felipe Chaulet), Bahia (José Souza de Jesus) e Lucas (Luca Bezerra). Obrigado pela ajuda e união pessoal.

Aos colegas do grupo de pesquisa pelas sugestões durante todo o processo.

Aos amigos, Dannylo Xavier, Igor Fonseca, Italo'Ivo, Marcelo Aguiar, Thaís Castelo Branco, Ramide Dantas, Virgínia Campos, Gabriela Wanderley por terem sido tão solícitos quando precisei (até nos finais de semana =) !!).

Por fim, porém, não menos importante, à André Vieira, meu chefe, pelo apoio incondicional a realização desse mestrado. Sua contribuição foi muito importante.

“ E se faz guerreiro sob o sol. Mais um brasileiro sob o céu. Que transforma a luta num troféu. Busca nos sertões um mar de paz. ”

—HAZAMAT (Sob o sol, 2011)

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. Cada usuário deve manter sua Taxa de Compartilhamento (TC) acima da SRE da comunidade para não ser banido da mesma. Contudo, um usuário desonesto pode usar um cliente BitTorrent modificado que é capaz de manipular sua TC de modo que a mesma seja sempre maior do que a SRE da rede. Na Internet existem diversos clientes BitTorrent modificados, como o RatioMaster, que falsificam relatórios com o intuito de permitir a manipulação da TC. Nesse caso, o cliente envia relatórios com informações adulteradas sobre o total de dados enviados e recebidos a fim de obter uma TC maior do que o limiar imposto na comunidade.

Esta dissertação apresenta inicialmente um estudo sobre uma comunidade privada BitTorrent formada por um rastreador com a implementação Xbtit, pares BitTorrent honestos μTorrent e pares desonestos que usam a ferramenta *RatioMaster* para falsificar seus relatórios, inflando artificialmente a taxa de compartilhamento. A partir desse estudo, esta dissertação propõe dois classificadores que permitem ao rastreador analisar enxames e identificar automaticamente os pares desonestos. O primeiro classificador é denominado TbC (*Threshold-based Classifier*) e o segundo é denominado AbC (*Autocovariance-based Classifier*). Esses classificadores foram avaliados sob diversos cenários controlados. Os resultados obtidos mostram que o classificador TbC apresenta uma taxa de acerto de 100% na maioria dos cenários estudados, ao passo que o classificador AbC apresenta uma taxa de acerto de 100% em todos os cenários estudados.

Palavras-chave: SRE, Comunidades Privadas BitTorrent, Falsificação de Relatórios

ABSTRACT

The utilization of incentives mechanisms is one remarkably feature of P2P BitTorrent systems. Inside the BitTorrent Ecosystem there are 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 the private network. However, some users can use malicious BitTorrent client to manipulate their share ratio and keep it always above the network SRE.

This work introduces a brief study about one private BitTorrent community composed of a xbtit tracker, with μTorrent clients representing honests peers and dishonest peers using the malicious BitTorrent client, called Ratio Master, to pretend a high share ratio. Based on this study, this work also proposes two classifiers that allows the tracker to identify automaticaly dishonest peers. The first classifier is named TbC (Threshold-based Classifier) and the second AbC (*Autocovariance-based Classifier*). These classifiers were evaluated under different controlled scenarios. Results show that the TbC classifier has an accuracy rate of 100% in most studied scenarios, while the classifier AbC has a hit rate of 100% in all evaluated scenarios.

Keywords: Share Ratio Enforcement, BitTorrent Private Communities, Fake Reports

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

GLOSSÁRIO

AbC *Autocovariance-based Classifier*, Classificador baseado em Autocovariância. 44

BSD *Berkeley Software Distribution*. 25

DHT *Distributed Hash Table*, Tabela *Hash* Distribuída. 8

FTP *File Transfer Protocol*, Protocolo de Transferência de Arquivos. 13

HTTP *HyperText Transfer Protocol*. 3

IEEE *Institute of Electrical and Electronics Engineers*. 25

IP *Internet Protocol*. 11

P2P *Peer-to-Peer*, Par a Par. 1

PC *Personal Computer*, Computador Pessoal. 24

PEX *Peer Exchange List*. 8

PHP Personal Home Page , Linguagem de Programação PHP. 25

ROC *Reciever Operating Characteristics*. 47

SRE *Share Ratio Enforcement*, Taxa de Compartilhamento Imposta. 1, 17

TbC *Threshold-based Classifier*, Classificador baseado em Limiar. 34

TC Taxa de Compartilhamento. 17

TCP *Transmission Control Protocol*, Protocolo de Controle de Transmissão. 16

CAPÍTULO 1

INTRODUÇÃO

O BitTorrent é a aplicação P2P (*Peer-to-Peer*) mais utilizada para compartilhamento de arquivos pelos usuários da Internet em geral [Mansilha et al. 2010]. Estudos recentes mostram que um tipo dessas redes, as comunidades privadas BitTorrent, utilizam um mecanismo auxiliar de incentivo, denominado SRE (*Share Ratio Enforcement*) ou Taxa de Compartilhamento Imposta, para estimular o compartilhamento altruísta entre os usuários. A utilização do SRE traz como principal benefício uma melhora significativa no desempenho de *download* da rede visto que os pares passam a contribuir mais, fornecendo recursos de banda passante. Contudo, o SRE é vulnerável a ataques o que motiva pesquisadores e administradores dessas comunidades a buscarem formas de prevenção contra tais sinistros.

1.1 MOTIVAÇÃO

Os sistemas P2P surgiram apresentam sobrepostas para a distribuição de conteúdo multimídia como compartilhamento de arquivos, distribuição de *streaming* de vídeo, ao vivo ou sob demanda, jogos, dentre outras. 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 [Kurose and Ross 2008]. Esses sistemas aproveitam-se dos recursos de rede, dos recursos de processamento e do número de usuários participantes para entregar conteúdo de forma escalável, com custo reduzido e de forma robusta [Xia and Muppala 2010]. Então, uma aplicação P2P pode ser definida como sendo um sistema distribuído formado por uma população transiente de pares, que se organizam automaticamente em uma rede lógica sobreposta a uma rede física, geralmente a Internet,

para compartilhar recursos [Buford et al. 2009, Lua et al. 2005].

Essas aplicações atingiram grande popularidade principalmente por serem capazes de realizar, de forma eficiente, transferências de grandes arquivos diretamente entre os pares. Segundo [Lehmann et al. 2011], um reflexo dessa popularidade é o volume significativo de tráfego P2P gerado na Internet. Uma grande parcela desse tráfego advém de aplicações de compartilhamento de arquivos e dentre elas destaca-se o BitTorrent. BitTorrent é um exemplo de sistema P2P que pode se utilizar ou de uma tabela *hash* distribuída, ou da troca de listas de usuários, como também de alguns mecanismos de controle, para organizar a rede e fazer com que os arquivos sejam transferidos [Wojciechowski et al. 2010].

Estudos evidenciam o sistema BitTorrent ao afirmarem que ele é responsável pela maior fatia de tráfego trocado por aplicações par a par na Internet [Kaune et al. 2010, Zhang et al. 2011, Chen et al. 2010a, Dan and Carlsson 2012]. Além dos pontos já mencionados, pode ser destacado como mais um ponto de sucesso do BitTorrent, a introdução do conceito de enxames (*swarming*). Um enxame é formado por um conjunto de pares interessados em um mesmo arquivo. Nele, existe a possibilidade de um par contribuir com outros ao passo que ele ainda está obtendo partes do conteúdo compartilhado [Wang et al. 2010, Price 2011]. Embora existam muitos estudos sobre o BitTorrent, recentemente, foi criado o termo ecossistema BitTorrent (ou Universo BitTorrent) para tentar descrever em profundidade a complexidade desse sistema de acordo com a atuação de suas entidades principais [Zhang et al. 2010a, Zhang et al. 2010b, Mansilha et al. 2010, Kryczka et al. 2011, Chen et al. 2012].

O Universo BitTorrent é composto por pares, pelos mecanismos de organização da rede e pelos *sites* para obtenção de conteúdos. Os pares desempenham papéis distintos em diferentes momentos no sistema. Os mecanismos de controle atuam na organização da rede, além disso, fazem com que os conteúdos sejam trocados com justiça [1] [Fan et al. 2009]. Os *sites* para obtenção de conteúdo fornecem o acesso para suas respectivas comunidades, que podem ser públicas ou privadas [Zhang et al. 2010b].

[1] Os mecanismos de controle se utilizam de estratégias para fazer com que os pares de uma rede BitTorrent cooperam entre si de forma a manter uma boa escalabilidade. Geralmente, essas estratégias buscam premiar os pares que contribuem mais com o sistema.

As comunidades públicas aceitam qualquer pessoa que esteja conectada à Internet e tenha interesse em seus arquivos. Nenhum tipo de controle é feito sobre seus usuários e poucas informações sobre eles podem ser obtidas [Cohen 2003, Xia and Muppala 2010]. As comunidades privadas BitTorrent se caracterizam por dispor de um ambiente fechado onde seus usuários precisam se autenticar para acessar um conteúdo restrito. Essas redes são monitoradas pelo rastreador ou *tracker* de tal forma que suas informações podem ser utilizadas por mecanismos de incentivo para verificar o nível de colaboração de cada usuário cadastrado. Os dados monitorados informam sobre a quantidade de usuários cadastrados, quantidade de pares em um enxame, quantidade de dados enviados, quantidade de dados baixados, dentre outros [Zhang et al. 2010a].

Além da forma de acesso aos conteúdos, observam-se algumas diferenças entre as comunidades públicas e privadas com relação aos mecanismos de incentivo utilizados, às informações extraídas e ao desempenho geral de *download* da rede. Diferentemente das comunidades públicas, que confiam principalmente nas estratégias do *Tit-for-Tat* [2] como forma de incentivo e justiça, as comunidades privadas BitTorrent utilizam um mecanismo de incetivo auxiliar denominado Taxa de Compartilhamento Imposta ou SRE (*Share Ratio Enforcement*). Esse mecanismo busca garantir que os usuários mantenham um certo nível de contribuição. Os usuários que estiverem abaixo desse nível são retirados da comunidade e os que estão acima continuam tendo acesso aos conteúdos disponibilizados por ela [Zhang et al. 2010b].

Devido a importância do SRE para as comunidades privadas, existe o interesse de pares desonestos em burlar tal mecanismo. Como exemplo de ataques ao SRE podem ser mencionados os conluios e a falsificação de relatórios. Os conluios são ataques realizados por um grupo de pares que trabalham cooperativamente para obter vantagens sobre o mecanismo de incentivo da rede, prejudicando assim os pares que estão contribuindo de fato [Chen et al. 2010b].

Os relatórios são enviados pelos pares, de tempos em tempos, para que seus estados

[2] O Tit-for-Tat (ou TFT) é uma política de seleção de pares usada para manter a reciprocidade na rede. Assim, pares enviam dados, prioritariamente, para os nós que disponibilizam mais recurso de banda para o sistema.

sejam relatados para o rastreador da rede. Esses relatórios são repassados através de uma requisição HTTP onde são informadas a quantidade de dados enviados e a quantidade de dados baixados, além de outras informações referentes ao par anunciante. O ataque de falsificação de relatórios consiste no envio de relatórios forjados, contendo informações adulteradas sobre a quantidade de dados enviados e quantidade de dados baixados pelo usuário. Com essa adulteração um par desonesto espera aumentar seu nível de compartilhamento rapidamente de forma a permanecer sempre acima da SRE imposta pela comunidade sem estar contribuindo de fato com a rede [Liu et al. 2010].

Dentre os diversos clientes BitTorrent modificados existentes (*Ratio Faker*[3], *Tracker Pro*[4] e *Torrent Ratio Keeper*[5]), se encontra o *RatioMaster* [6]. Este cliente possui um nicho considerável de usuários de comunidades privadas na Internet, que através de sua utilização, adulteram suas taxas de compartilhamento para o SRE. O comportamento desses usuários pode prejudicar o bom funcionamento da rede, uma vez que suas trapaças enganam o mecanismo de incentivo implantado justamente para melhorar o desempenho de *download* geral da rede. Dada tal problemática se faz necessária uma abordagem que a trate através da identificação dos pares que estão manipulando indevidamente suas informações sem que o desempenho da rede seja prejudicado.

1.2 OBJETIVOS

Este trabalho tem como objetivo geral propor algoritmos para a detecção de pares desonestos que atuam em uma comunidade privada BitTorrent real se utilizando do cliente modificado *RatioMaster* para enganar o rastreador da rede através do ataque de falsificação de relatórios. Nesse contexto, os classificadores TbC e AbC permitem ao rastreador da rede identificar pares maliciosos que estão forjando suas informações relacionadas a quantidade de dados enviados e baixados através da utilização desses clientes. Para alcançar este objetivo geral, os seguintes objetivos específicos são definidos:

[3]http://ratiofaker.blogspot.com/ - Último acesso em 30/03/2012.
[4]http://www.esanu.name/software/?p=9 - Último acesso em 30/03/2012.
[5]http://www.torrentratiokeeper.com/tutorial/ - Último acesso em 30/03/2012.
[6]http://www.ratiomaster.net/ - Último acesso em 30/03/2012.

- Realizar estado da arte sobre Comunidades Privadas BitTorrent e o uso da falsificação de relatórios;
- Montar comunidade privada BitTorrent para realização de experimentos;
- Compreender o funcionamento do cliente malicioso *RatioMaster*;
- Avaliar algoritmos de classificação para detecção de pares desonestos.

1.3 CONTRIBUIÇÃO

Este trabalho se diferencia dos trabalhos relacionados por estudar uma forma de identificação automática de pares maliciosos que burlam o mecanismo SRE em comunidades privadas. Para isso, foram propostos dois classificadores que permitem ao rastreador analisar enxames e identificar automaticamente os pares maliciosos. Os pares são classificados sem a utilização de relatórios casados, sem adição de novas informações aos relatórios enviados pelos pares e nem alterações no protocolo BitTorrent original.

Os classificadores propostos foram avaliados em uma comunidade privada BitTorrent formada por um rastreador com a implementação Xbtit, pares honestos que utilizam o cliente BitTorrent μ*Torrent* e pares desonestos que usam o cliente BitTorrent malicioso *RatioMaster* para adulterar seus relatórios com o intuito de aumentar indevidamente sua taxa de compartilhamento.

1.4 ORGANIZAÇÃO

Esta dissertação está organizada como segue: O Capítulo 2 mostra uma visão geral sobre o protocolo BitTorrent. Além disso, apresenta as comunidades privadas e alguns aspectos de segurança relativos ao seu mecanismo de incentivo auxiliar. Por fim, apresenta os trabalhos relacionados e como esta dissertação se diferencia deles. Em seguida, o Capítulo 3 apresenta uma avaliação experimental sobre o comportamento da taxa de compartilhamento dos usuários de uma comunidade BitTorrent privada real. No Capítulo 4, são apresentados os dois algoritmos propostos para a detecção de pares desonestos que se uti-

lizam de um cliente desonesto BitTorrent (RatioMaster) para burlar o SRE. Por fim, no Capítulo 5 são apresentadas as considerações finais assim como propostas para trabalhos futuros.

CAPÍTULO 2

COMUNIDADES PRIVADAS BITTORRENT E ASPECTOS DE SEGURANÇA

Neste capítulo são apresentados a visão geral do BitTorrent, mais especificamente, seu funcionamento básico, o processo para obtenção de conteúdos e a comunicação entre pares e rastreadores. As comunidades privadas BitTorrent e tópicos relacionados a elas, como a importância do seu mecanismo de incentivo auxiliar conhecido por Taxa de Compartilhamento Imposta ou SRE, também são abordados neste capítulo. Além disso, são abordados ataques utilizados por pares desonestos para burlarem o SRE. Por fim, são apresentados os trabalhos relacionados mostrando sua relação com a problemática abordada neste trabalho e como o trabalho proposto se diferencia deles.

2.1 VISÃO GERAL DO BITTORRENT

O BitTorrent é considerado o principal sistema P2P para troca de arquivos na Internet [Buford et al. 2009]. Através da sua utilização, milhões de usuários se organizam em várias subredes lógicas com o intuito de adquirir conteúdos específicos. Esses arquivos são divididos em pequenos pedaços de modo que o desempenho de *download* da rede seja melhorado à medida que eles sejam transferidos [Xia and Muppala 2010].

Além da sua eficiência, alguns outros fatores justificam o sucesso desse protocolo, dentre eles, a publicação do código-fonte do seu principal cliente e da especificação do protocolo original, a utilização de mecanismos que buscam o compartilhamento justo e a vasta diversidade de conteúdos disponíveis na Internet [Fan et al. 2009]. Com a divulgação do código fonte, desenvolvedores de todo o mundo podem colaborar amplamente com o seu processo de desenvolvimento permitindo o surgimento de implementações personalizadas

para clientes, rastreadores (*trackers*) e extensões ao protocolo BitTorrent original.

Um cliente BitTorrent é um *software* utilizado para que um usuário seja capaz de adquirir um conteúdo específico [Mansilha et al. 2010]. Alguns clientes se destacam pela inovação que apresentaram ao serem lançados, como exemplo, podem ser citados o *Mainline* [Cohen 2005], o μTorrent [1] e o Vuze [2]. Os rastreadores são elementos centralizados na arquitetura BitTorrent e atuam como agentes organizadores da rede. Ao tentarem obter um conteúdo, os pares recém chegados, entram em contato com o rastreador para entender a situação atual da rede. Existe uma série de implementações de código aberto disponíveis. Uma delas é o *Xbtit Tracker* [3], que se destaca pela gama de funcionalidades que oferece para seus administradores. Dentre elas, um sistema de relatórios integrado, *scripts* de instalação e uma comunidade bastante ativa na Internet [Zhang et al. 2010a]. As extensões ao protocolo original foram sugeridas para corrigir pontos de falha não vislumbrados no decorrer do seu desenvolvimento. Algumas delas focam na melhoria do processo de descoberta de novos pares e consequentemente, como serão estabelecidas as conexões entre eles, como por exemplo, a extensão PEX [Wu et al. 2010] e a DHT [Junemann et al. 2011]. Já outras, como a *Fast Extension* [Harrison and Cohen 2008], são utilizadas para aumentar a eficiência do protocolo BitTorrent.

2.1.1 Funcionamento Básico

O comportamento do protocolo BitTorrent depende da dinâmica imposta pela necessidade dos usuários, pelo processo de aquisição de conteúdos e pela atuação dos mecanismos de organização e de incentivo da rede. Para baixar um arquivo, um usuário (ou par) necessita de uma série de informações referentes a ele, tais como, a descrição do conteúdo, quais os pares que o possuem, se o arquivo possui fornecedores [Cohen 2005]. Essas informações são inseridas em um arquivo metadados também conhecido por torrent (ou *.torrent*). Então, para cada conteúdo disponibilizado existe um arquivo torrent. No

[1] http://www.utorrent.com/ (Último acesso em 19/06/2012)
[2] http://www.vuze.com/ (Último acesso em 19/06/2012)
[3] http://www.xbtit.com/ (Último acesso em 20/07/2012)

entanto, podem haver casos onde o arquivo torrent existe mas seu respectivo conteúdo não é mais compartilhado. Como por exemplo, arquivos de filmes muito antigos. Qualquer usuário pode criar um arquivo torrent, mas, nem todos podem disponibilizá-lo na Internet visto que existem *sites* onde apenas um usuário administrador tem autorização para publicar os conteúdos da rede.

Os rastreadores, como comentado anteriormente, são nós especias que armazenam os arquivos torrent e gerenciam as informações enviadas pelos pares que estão participando de um enxame. Um enxame é formado por um conjunto de pares que estão compartilhando um mesmo arquivo. Os pares que estão fazendo parte de um enxame podem ser classificados como sugadores (*leechers*) e semeadores (*seeders*). Um semeador é um par que já possui uma cópia completa do arquivo e permanece na rede para compartilhá-lo com outros pares da rede. Para que um arquivo seja disponibilizado é necessário que exista pelo menos um semeador capaz de prover o conteúdo completo. Um par pode ser considerado sugador quando ainda estiver baixando um arquivo. Apesar de não possuir o arquivo completo ele pode contribuir com o sistema enviando as partes que já possui [Cohen 2003].

O funcionamento básico do BitTorrent é facilmente compreendido se observado o processo para obtenção de conteúdos representado na Figura 2.1. Nesse processo, os pares que desejam baixar um conteúdo devem, primeiramente, adquirir o torrent do referido arquivo. De posse desse arquivo, um par já é capaz de se anunciar para o rastreador através de uma requisição HTTP GET [Xia and Muppala 2010].

Quando recebe uma requisição, o rastreador responde ao par solicitante com uma lista de potenciais pares que poderão se tornar vizinhos dele. Esses pares serão selecionados de forma aleatória a partir do conjunto de usuários que possuem o arquivo desejado. Os passos para que um compartilhamento seja realizado estão descritos nos itens abaixo [Cohen 2003]:

1. Um par decide baixar um arquivo e acessa o *site* para fazer o *download* do arquivo *.torrent* correspondente;

2. O par, agora como sugador, entra em contato com o rastreador e solicita a lista de

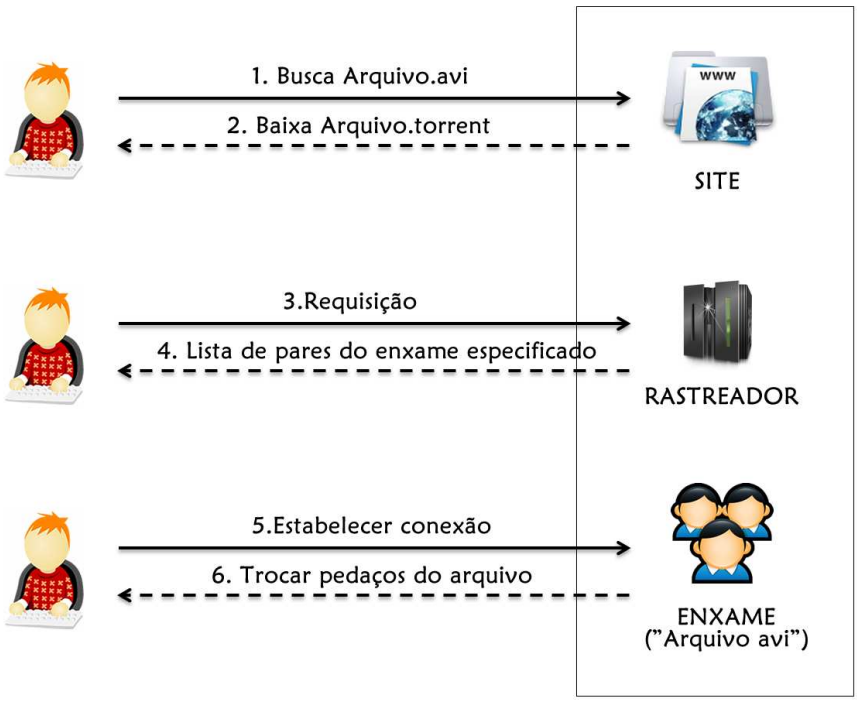


Figura 2.1: Processo para obtenção de conteúdo em uma rede BitTorrent.

pares ativos no enxame;

3. O rastreador responde a solicitação com uma lista de pares;

4. O sugador escolhe os vizinhos que possuem o arquivo almejado e estabelece conexão com eles;

5. O sugador agora pode trocar pedaços do arquivo com seus respectivos vizinhos.

O arquivo compartilhado entre vizinhos é dividido em pedaços (*pieces*) com tamanhos pré-estabelecidos, geralmente 256 KB, e quando são totalmente baixados, uma chave *hash* SHA1 é gerada e comparada com o valor *hash* contido no arquivo de metadados (.torrent) para verificar a integridade do pedaço transmitido. Se os valores forem iguais, o sugador anunciará que possui determinado pedaço e o disponibilizará para seus vizinhos [Quental and Gonçalves 2010]. Os pares trocam pedaços não só com os semeadores, mas também com outros sugadores vizinhos. Essa prática retira a sobrecarga de cima dos semeadores e a distribui entre os nós participantes do enxame.

2.1.2 Comunicação entre Pares e Rastreadores

Durante uma transferência, os pares de um enxame devem se reportar constantemente ao rastreador para atualizar sua situação na lista de pares de determinado enxame e obter endereços de outros pares que também estão no mesmo enxame. Para isso, se faz necessário o envio de relatórios ou anúncios (*Tracker Announces*).

Os relatórios são enviados pelos pares em forma de requisições HTTP. Elas são compostas por informações dos clientes e servem para auxiliar os rastreadores a manterem estatísticas gerais sobre os torrents da rede. Essa forma de interação entre os pares e rastreadores se caracteriza como um serviço HTTP capaz de gerenciar as solicitações dos usuários e as respostas emitidas pelos rastreadores. As informações relacionadas aos pares são enviadas através de mensagens *Announce* que possuem os seguintes parâmetros:

- ***Info-hash***: Identificador único do torrent;
- ***Peer-id***: Identificador do par solicitante;
- ***Port***: Fornece o número da porta pela qual o par está escutando;
- ***IP***: Parâmetro opcional que fornece o endereço IP do par;
- ***Numwant***: Parâmetro opcional que mostra o número de pares desejados;
- ***Event***: Parâmetro opcional que mostra a situação dedo par no enxame;
- ***Uploaded***: Quantidade de dados enviados pelo par até o momento do envio;
- ***Downloaded***: Quantidade de dados recebidos pelo par até o momento do envio;
- ***Left***: O número de *bytes* que restam ser baixados para o fim do *download*;
- ***NoPeerId***: Parâmetro opcional que pode ser usado caso o rastreador necessite omitir o id dos pares na resposta;
- ***Compact***: Parâmetro opcional que compreende a representação compacta de pares;

- ***Key***: Parâmetro opcional que pode armazenar um valor conhecido apenas para o rastreador e o par em questão;
- ***Trackerid***: Parâmetro opcional que identifica um par que está retornando ao enxame.

O rastreador trabalha para fazer com que os pares que desejam compartilhar um mesmo arquivo se encontrem e, além de responder ao par requisitante com uma lista de pares presentes no enxame, ele pode informar em suas respostas situações de erro, parâmetros de funcionamento e estados dos usuários do enxame. As respostas são dicionários codificados com *bencode* [4] e contêm alguns campos, conforme descrito abaixo:

- ***Peers****(Dicionário)*: Lista de pares com respectivos endereços IP (*Internet Protocol*) e portas;
- ***Peers****(Binário)*: *String* da lista de pares;
- ***Interval***: Intervalo entre requisições;
- ***Min-Interval***: Parâmetro opcional que representa o intervalo mínimo entre requisições;
- ***Tracker Id***: *String* que o par deve retornar em requisições futuras;
- ***Failure Reason***: Enviada em caso de falha;
- ***Warning Message***: Mensagem de aviso em caso de situação inesperada;
- ***Complete***: Número de semeadores no enxame;
- ***Incomplete***: Número de sugadores no enxame.

O intervalo entre envios de relatórios é delimitado pelos parâmetros *min-interval* e *interval flag* configurados no rastreador. O primeiro deles está relacionado com a quantidade de tempo, em segundos, que um par deve esperar para enviar seus relatórios.

[4] *Bencode* é uma codificação utilizada pelo BitTorrent para armazenar e transmitir estruturas de dados.

Caso utilizado, o segundo parâmetro delimita o intervalo mínimo para que todos pares presentes no enxame enviem seus anúncios.

2.2 COMUNIDADES PRIVADAS

Levando em consideração sua diversidade e sua parcela de contribuição com o tráfego total da Internet, alguns pesquisadores utilizam os termos Ecossistema BitTorrent [Zhang et al. 2010b] ou Universo BitTorrent [Mansilha et al. 2010] para descrever a complexidade desses sistemas de forma mais detalhada. 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*.

Os pares utilizam clientes ou agentes BitTorrent para ter acesso ao conteúdo desejado. Eles podem exercer papéis distintos e participar de vários enxames simultaneamente. Os possíveis papéis são definidos de acordo com a atuação de cada par no decorrer de uma transferência. Desse modo, ele pode ser considerado sugador ou semeador. Um enxame então, representa um conjunto formado por pares que possuem interesse no mesmo conteúdo. Já os rastreadores são nós que atuam gerenciando a entrada de nós e os estados de cada par em um ou múltiplos enxames.

Os mecanismos de organização da rede atuam fazendo com que os conteúdos sejam transferidos da forma mais eficiente e justa possível. Dentre eles, existem os mecanismos que tratam a escolha de pares vizinhos e os mecanismos que gerenciam a seleção de pedaços que serão trocados. Dentre os principais mecanismos podem ser citados o algoritmo de *choking* que é regido por quatro estratégias (*Tit-for-Tat*, *Optimistic Unchocking*, *Anti-Snubbing*, *Upload Only*) que visam melhorar o desempenho de *download* dos pares que contribuem mais [Fan et al. 2009].

Os torrents podem ser obtidos de diversas formas na Internet. Seja via email, seja via servidores FTP, bem como páginas WEB. Então, para facilitar o descobrimento de torrents surgiram as comunidades BitTorrent que disponibilizam seus conteúdos através de portais ou *sites* para descobrimento de torrents. Essas comunidades tipicamente disponibilizam localizadores que facilitam a descoberta de conteúdos, por exemplo, campos

para consultas por nome de arquivo, por tipo(dependendo de sua extensão) ou até mesmo por sua popularidade.

As comunidades BitTorrent podem ser classificadas em duas categorias, as públicas (ou abertas) [Cohen 2003] e as privadas (ou fechadas) [Zhang et al. 2010a]. A primeira é composta por comunidades abertas em que não existe qualquer tipo de restrição quanto a sua utilização, ou seja, qualquer pessoa que esteja conectada à Internet poderá obter conteúdos dessas comunidades. Como exemplos de comunidades públicas podem ser citadas *Mininova*[5], *The Pirate Bay*[6] e *Isohunt*[7]. Por outro lado, a segunda categoria abrange comunidades que restringem o acesso aos conteúdos compartilhados através da utilização de convites e registros de usuários. As comunidades *BitSoup*[8] e *Dimeadozen*[9] são exemplos dessa categoria. Este trabalho tem foco nas comunidades privadas BitTorrent.

Pesquisas recentes mostram o crescimento dessa categoria de comunidade BitTorrent e através de medições, afirmam que existem aproximadamente 900 redes privadas na Internet [Zhang et al. 2010a]. Nas redes privadas os usuários precisam se autenticar em uma comunidade restrita para ter acesso ao conteúdo almejado. O *site* ou portal da comunidade é o ponto central onde é feito o cadastro dos usuários que desejam participar delas. Tipicamente, convites de acesso são utilizados para que os registros de contas sejam limitados de acordo com a capacidade desejada pelo administrador. Dessa forma, caso prove ser "bom cidadão", um membro de uma comunidade fechada recebe convites que podem ser entregues para seus amigos ou até mesmo vendidos em *sites* como *Ebay* [10] ou *Mercado Livre* [11]. Esse procedimento é bem diferente do que ocorre em uma comunidade pública, que permite usuários arbitrários acessarem o *site* da comunidade para baixar arquivos torrent sem que sejam registrados [Chen et al. 2010b].

[5]http://www.mininova.com/- Último acesso em 20/07/2012
[6]http://www.thepiratebay.se/- Último acesso em 20/07/2012
[7]http://www.isohunt.com/- Último acesso em 20/07/2012
[8]http://www.bitsoup.org/- Último acesso em 20/07/2012
[9]http://www.dimeadozen.org/- Último acesso em 20/07/2012
[10]http://www.ebay.com/- Último acesso em 15/06/2012
[11]http://mercadolivre.com.br/- Último acesso em 30/03/2012

2.2.1 Processo para Obtenção de Conteúdos

O processo para obtenção de conteúdos em uma comunidade privada se diferencia em alguns aspectos, em relação às comunidades abertas. Na primeira, os usuários registrados são identificados unicamente por uma chave conhecida por *"pass key"*. Essa chave permite ao rastreador prover autorização para que eles sejam capazes de enviar(caso tenha permissão) ou baixar os conteúdos. A Figura 2.2 mostra os passos que um usuário deve percorrer antes de obter um determinado arquivo em uma comunidade fechada. Inicialmente, ele deve conseguir ou comprar um convite de acesso. De posse desse convite, ele pode se cadastrar no portal da comunidade e aguardar sua liberação. Quando liberado, o usuário já é considerado membro e agora pode acessar o portal para baixar os arquivos desejados.

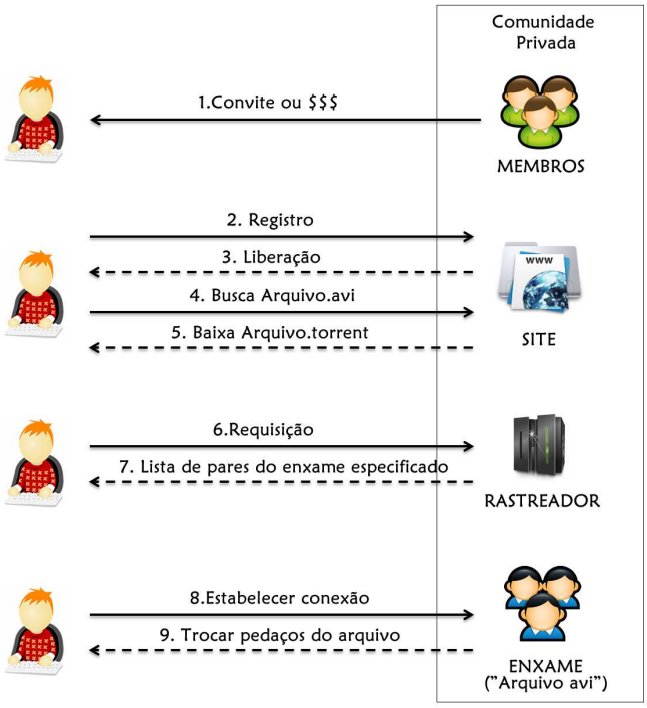


Figura 2.2: Obtenção de conteúdos em uma comunidade privada BitTorrent.

Para exemplificar o processo de obtenção, suponha que ***X*** é um membro registrado em uma comunidade fictícia *"comunidadeprivada.com"* e está interessado em baixar um arquivo de áudio específico. Primeiramente, ***X*** acessa *"comunidadeprivada.com"* e obtém

o arquivo metadados relativo ao áudio desejado. O arquivo retornado como resposta do *site* contém o endereço IP do rastreador e o arquivo torrent. Nesse instante a chave identificadora de ***X*** será inserida, pelo rastreador, no arquivo torrent e o campo *"private"* da requisição será passado para 1 (um). Isso indica que o conteúdo em questão pertence a uma comunidade privada.

Assim que executa o torrent baixado, o cliente utilizado por ***X*** se reporta ao rastreador nele especificado e passa sua chave. Feito isso, ele recebe autorização para conhecer os pares participantes do enxame relativo ao arquivo de áudio desejado. O rastreador valida essa requisição e retorna para ***X*** uma lista contendo um subconjunto de pares interessados no mesmo arquivo. Então, para iniciar o processo de compartilhamento, ***X*** estabelece conexões TCP com seus vizinhos e troca blocos de arquivos com eles.

Existem alguns jargões utilizados pelos usuários das comunidades privadas e o entendimento desses termos facilita a compreensão da dinâmica entre os administradores e os membros de uma comunidade privada. Por exemplo, o convite de acesso é conhecido por *"Invitation Code"*. Já a chave identificadora de um par é chamada de *"Passkey"*. Muitas comunidades empregam um sistema de perfis onde seus usuários são classificados de acordo com seus níveis de contribuição, tempo de registro, taxa de compartilhamento média ou até mesmo pela quantidade de arquivos enviados. Quanto mais participativo for o par, melhor a reputação dele na comunidade. O perfil estabelece diferentes privilégios e é chamado de *"User Class"*. O termo *"Snatched"* indica quantas vezes um arquivo foi baixado completamente. Já o termo HnR, acrônimo de *"High Prohibited"*, refere-se ao par que baixa um conteúdo e deixa a comunidade. Todo novo arquivo disponibilizado é conhecido por *"Scene Release"*.

2.2.2 Diferenças entre Comunidades Públicas e Privadas

Existem diferenças perceptíveis entre as comunidades abertas e fechadas. As abertas não identificam seus usuários, não registram suas contribuições (envio de dados), nem seu consumo (quantidade de dados baixados) nos enxames existentes, diferentemente do que acontece nas fechadas. Por esse motivo é mais difícil extrair dados sobre usuários de

enxames públicos.

Outra diferença importante está no fato de que os rastreadores de comunidades privadas utilizam os relatórios recebidos periodicamente de seus clientes para armazenar informações relativas à quantidade de dados baixados e enviados por cada um deles. Apesar de engessar um pouco mais a arquitetura da rede, esse nó centralizador se utiliza de algumas políticas de incentivo que usam os dados coletados para premiar os pares que estão colaborando mais com a comunidade. A partir das informações colhidas o rastreador obtém a fração do número de dados enviados pela de dados baixados para cada usuário cadastrado no sistema. Essa fração é denominada Taxa de Compartilhamento (TC) ou *Share Ratio* e é utilizada para medir o grau de compartilhamento de um usuário registrado em uma comunidade privada.

A diferença mais marcante das comunidades fechadas é o fato de que elas utilizam o mecanismo auxiliar de incentivo conhecido por Taxa de Compartilhamento Imposta ou *Share Ratio Enforcement* (SRE). O mecanismo SRE é importante, 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. Um valor mínimo para a razão de compartilhamento mínima imposta é determinada pelo administrador da comunidade, por exemplo 0,7. Por isso os pares devem fazer transferências para ficarem com suas TCs acima desse valor estipulado.

Nas comunidades privadas da Internet existem basicamente duas formas para se estabelecer o valor do SRE, a votação e a imposição [Xia and Muppala 2010]. Na primeira delas, os usuários participam de uma eleição para escolher um valor entre as opções dadas pelos organizadores da comunidade privada. Já na segunda opção, o administrador elege um valor para o SRE impondo-o perante os membros da comunidade.

2.2.3 Taxa de Compartilhamento Imposta ou SRE

As comunidades privadas BitTorrent aplicam políticas que visam incentivar os usuários a colaborarem mais com a rede. Seja através do envio de novos conteúdos ou pelo *upload* de

arquivos baixados anteriormente. Algumas comunidades exigem que seus usuários passem um tempo mínimo compartilhando após adquirirem um arquivo. Já outras oferecem benefícios para usuários que contribuem através de donativos (dinheiro). Porém, a forma mais tradicional de medir o grau compartilhamento de um usuário é através da TC de cada um. Caso sua TC esteja acima do SRE, significa que o usuário está compartilhando dentro do esperado.

A utilização desse mecanismo traz como grande benefício uma melhora significativa no desempenho geral da rede [Jia et al. 2011b] . Isso significa dizer que os pares participantes da comunidade são capazes de baixar mais conteúdos em um menor espaço de tempo caso estejam dentro do limiar esperado. Apesar disso, existem dois fatores negativos que precisam ser observados. O primeiro deles diz respeito a uma possível sub-utilização da banda de *upload* de um par que passa um grande período fornecendo um arquivo para obter maior TC enquanto nenhum outro par se interessa pelo conteúdo disponibilizado. Esse fato acarreta no desperdício da banda passante de *upload* do semeador. O outro fator tem ligação com a vulnerabilidade do SRE à ataques onde pares desonestos podem atuar sozinhos ou em conluios para obter maiores taxas de compartilhamento sem que estejam contribuindo de fato com a rede.

2.3 ATAQUES AO SRE

Segundo [Liu and Shi 2010], qualquer mecanismo de incentivo pode ser vulnerável a certos ataques ou até mesmo ser comprometido por tentativas desonestas de manipulação da reputação por parte de pares maliciosos. Dessa forma, esses pares podem lançar ataques que põem em risco o bom desempenho da rede. Os pares desonestos se utilizam do comportamento malicioso para tirar proveito dos usuários honestos da comunidade sem contribuir com ela.

Nas comunidades privadas BitTorrent existem pares desonestos que estão interessados em atacar o mecanismo SRE. Esses pares se utilizam de um cliente malicioso para manter, artificialmente, uma taxa de compartilhamento maior do que o limiar imposto para a comunidade. Exemplos de ataques com tal objetivo incluem os con-

luios [Lian et al. 2007], [Liu et al. 2010], [Ciccarelli and Cigno 2011] e a falsificação de relatórios [Liu et al. 2010].

2.3.1 Conluios

As redes fechadas tipicamente utilizam o SRE para incentivar a cooperação entre seus nós. Geralmente, tais mecanismos são vulneráveis à atuação de usuários que atuam em conjunto para obter vantagens sobre os outros. Então, um conluio pode ser definido como uma atividade de colaboração onde grupos de usuários atuam em busca de benefícios que não seriam capazes de obter caso eles estivessem agindo sozinhos.

Para enganar o rastreador, uma dupla de pares precisa cooperar entre si para ter a certeza de que suas ações sejam combinadas e dificultar sua identificação. Considerando que os membros de uma comunidade privada usam a quantidade de dados enviados para obter uma TC alta e dado que tais comunidades não possuem mecanismos de proteção contra conluios, pares desonestos, por exemplo, podem se beneficiar ao usarem um conjunto mínimo de arquivos para conseguirem manter sua TC acima do limiar imposto pela comunidade. Essa forma de complô pode ser investigada caso seja observada a quantidade de tráfego duplicada está passando na comunidade. Entende-se por tráfego duplicado a transferência recorrente de um mesmo conteúdo entre os mesmos pares vizinhos.

A Figura 2.3 mostra pares participantes de um conluio trocando conteúdo entre si para obter melhores taxas de compartilhamento enquanto que o par honesto não consegue evoluir mesmo tentando participar do enxame. Os pares desonestos compartilham seus recursos de banda entre si de forma que os dois, ou vários, participantes sejam capazes de obter uma TC significativamente acima do SRE da rede, enquanto o usuário honesto obtém um desempenho pífio por não participar das transferências.

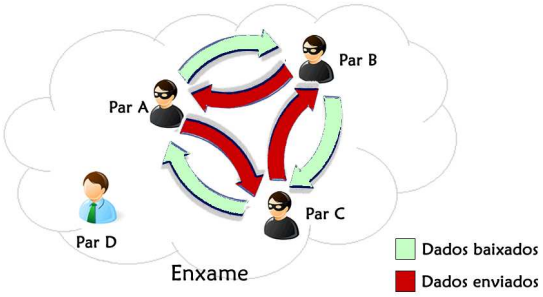


Figura 2.3: Exemplo de conluio.

2.3.2 Falsificação de Relatórios

Na falsificação de relatório para burlar o mecanismo SRE, pares desonestos falsificam seus relatórios com o intuito de permitir a manipulação da taxa de compartilhamento do usuário de uma comunidade privada. 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.

Esse tipo de falsificação é relativamente simples e pode ser feita através do uso de clientes BitTorrent modificados como o *RatioMaster*, *Ratio Faker*, *Tracker Pro* e *Torrent Ratio Keeper*. Esses clientes, tipicamente, informam ao rastreador uma quantidade de dados enviados e recebidos tal que a taxa de compartilhamento seja maior do que a imposta na comunidade.

A Figura 2.4 mostra uma falsificação de relatório onde um usuário desonesto **A** envia de fato 200MB para um par honesto **B**. Ao mandar seu relatório para o rastreador, o usuário **A** informa que enviou 1GB. Com isso ele terá uma TC de 5,0, ou seja, terá uma TC muito alta sem ter contribuído de fato com a comunidade. Esta dissertação tem como escopo a identificação de pares que atuam sozinhos se utilizando do ataque de falsificação de relatórios em uma comunidade privada BitTorrent.

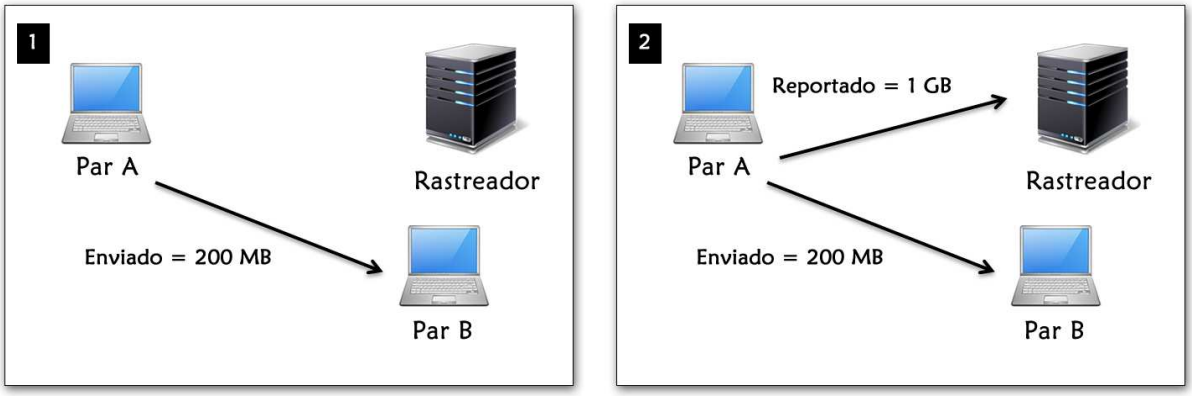


Figura 2.4: Exemplo de falsificação de relatório.

2.4 TRABALHOS RELACIONADOS

O estudo das comunidades privadas BitTorrent vem recebendo grande atenção na literatura [Meulpolder et al. 2010], [Zhang et al. 2010a], [Chen et al. 2010b], [Chen et al. 2011], [Paiva and Gonçalves 2012], [Kash et al. 2012]. 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], [Jia et al. 2011c].

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 proposto 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.

Para identificar pares utilizando essa forma de trapaça, em [Liu et al. 2010] é proposta a utilização de relatórios "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 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.

Esta dissertação 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 os classificadores de pares maliciosos, este trabalho foca em uma comunidade privada BitTorrent formada por um rastreador com a implementação *Xbtit*, pares BitTorrent honestos *µTorrent* e pares maliciosos que usam a ferramenta *RatioMaster* para falsificar seus relatórios, aumentando artificialmente a taxa de compartilhamento. Com essa comunidade, o comportamento dos clientes maliciosos é observado e, a partir disso, são propostos e avaliados dois classificadores que permitem ao rastreador analisar enxames e identificar automaticamente os pares maliciosos.

2.5 RESUMO

Neste capítulo foram apresentadas a visão geral e o funcionamento básico de uma rede BitTorrent para compartilhamento de arquivos. Além disso, também foram detalhados o funcionamento do processo para obtenção de conteúdos, bem como os parâmetros que compõem as interações entre os pares e os rastreadores.

Essas interações compõem uma espécie de serviço HTTP no qual as mensagens trocadas correspondem a requisições desse protocolo. Dessa forma, os nós de um enxame enviam seus relatórios ao rastreador para informar sobre seus *status*. Cabe a ele a responsabilidade de fazer com que os pares interessados em um mesmo conteúdo se encontrem.

As comunidades privadas BitTorrent utilizam uma política de incentivo baseada em uma Taxa de Compartilhamento Imposta ou SRE. A utilização do SRE tenta fazer com que seus usuários mantenham sua TC acima de um limiar para que eles não sejam expulsos da rede.

Estudos relacionados mostram que esse mecanismo auxiliar de fato motiva os usuários a compartilharem mais, no entanto, devido a sua importância, existe o interesse de usuários desonestos em enganá-lo. Alguns deles se utilizam de um cliente malicioso *RatioMaster* para fraudar os dados relacionados a quantidade de dados enviados e a quantidade de dados baixados que são reportados ao rastreador de uma comunidade privada, a cada envio de relatório.

Para que rastreadores privados sejam capazes de identificar automaticamente pares

desonestos, se faz necessária uma investigação mais apurada de como se comportam tais nós. A partir disso, formas de classificação podem ser desenvolvidas para a detecção de pares que estão se utilizando de tal ferramenta para obter TCs significativas em um menor intervalo de tempo.

CAPÍTULO 3

AVALIAÇÃO EXPERIMENTAL DA TAXA DE COMPARTILHAMENTO

Neste capítulo são apresentados o detalhamento do ambiente para realização de experimentos e os cenários de avaliação utilizados. O ambiente implantado serviu para a observação do comportamento de pares em um enxame privado. Primeiramente, foi realizada uma análise sobre a influência da variabilidade da TC no comportamento de um usuário. A partir disso, foram estudadas possíveis formas de detecção com a utilização de duas métricas auxiliares para a classificação de pares desonestos.

3.1 O μMUNDO

Para o estudo proposto neste trabalho, foi criada uma comunidade BitTorrent privada, a qual é doravante denominada *μMundo*. A utilização de um ambiente real reduzido se faz necessária por dois fatores principais: o primeiro é a praticidade do gerenciamento de nós no decorrer dos experimentos. Suponha que os algoritmos propostos fossem implantados em uma comunidade real na Internet. Seria inviável realizar testes sem que observações iniciais fossem realizadas. Por exemplo, para realizar um estudo com 1000 clientes, seriam necessários usuários espalhados pela Internet. Outro problema é a sincronização dos pares em um enxame real para a coleta de amostras, visto que os usuários da comunidade teriam que iniciar compartilhamentos em horários pré-definidos. O segundo fator diz respeito a forma de avaliação dos algoritmos propostos. Avaliá-los em um ambiente controlado permite saber quais são os pares desonestos e esse conhecimento prévio garante que os algoritmos propostos possam ser avaliados quanto a sua acurácia ao classificá-los.

A Figura 3.1 apresenta o *μMundo* utilizado para o estudo de uma rede privada Bit-

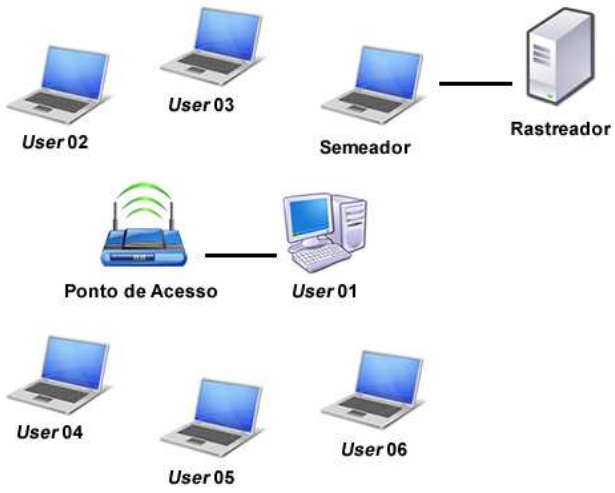


Figura 3.1: O $\mu Mundo$.

Torrent. Essa rede é composta por 6 *notebooks*, 1 PC e um Roteador sem fio. Cada *notebook* representa um único cliente da rede conectado através do padrão 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 [Zhang et al. 2010a], desenvolvido pela *BITteam* sob licença BSD. Essa implementação é desenvolvida em linguagem PHP e possui integração opcional com a linguagem C++ que pode ser utilizada caso seja necessária uma maior eficiência do servidor.

O banco MySQL é utilizado pelo rastreador para armazenar os arquivos *torrents* e as informações tratadas pelo rastreador, como por exemplo: o nome dos usuários, a quantidade de semeadores e sugadores por arquivo, as taxas de compartilhamento, a quantidade de arquivos existentes, dentre outras. Para o armazenamento dos experimentos foi criada uma tabela denominada *experiments* no banco de dados do Rastreador, *Xbtit*. Para cada

elemento de amostra foram armazenados o número identificador do experimento, em um campo *ID*, a sua hora de coleta, no campo *Time*, o número identificador do par (*UserId*), o nome de usuário do par (*UserName*), a taxa de compartilhamento do par (*ShareRatio*), a data(*Date*) e por fim o nome do arquivo(*fileName*). As coletas são feitas por enxame, ou seja, são armazenados os dados de todos os pares participantes de determinado enxame.

A realização da captura dos dados acima é feita por um *script* desenvolvido para coletar elementos de amostras e salvá-los na tabela *experiments*, mencionada anteriormente. Esse *script* efetua capturas de amostras em um dado intervalo de tempo, que pode variar, de acordo com a quantidade de amostras necessárias.

Os usuários honestos utilizam o cliente BitTorrent μTorrent no decorrer dos experimentos. Esse cliente foi escolhido por dois fatores principais: o primeiro deles é a praticidade quanto a sua instalação e o segundo por consumir pouco recurso computacional. O μTorrent se tornou popular após sua aquisição pelo criador do protocolo BitTorrent, Bram Cohen, passando, desta maneira, a ser a versão oficial do BitTorrent. Segundo informações divulgadas em [TorrentFreak 2009] e [TorrentFreak 2011], atualmente, ele é o cliente mais popular, com mais de 56% dos usuários do BitTorrent em 2009, chegando a 100 milhões de usuários em 2011.

Para caracterizar o comportamento de pares desonestos, foi utilizado o cliente malicioso *RatioMaster*. Esta ferramenta possui *sources* oficiais para *download*, enquanto que a maioria dos outros clientes maliciosos estão espalhadas em servidores de origem duvidosa na Internet. A praticidade na instalação e aprendizado desta ferramenta também foram pontos determinantes para a sua adoção.

3.2 DESCRIÇÃO DOS CENÁRIOS

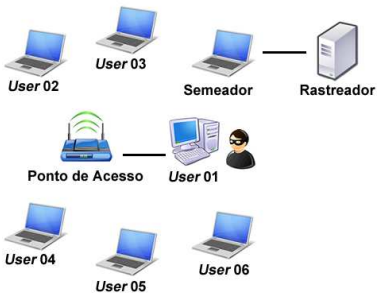
Para o estudo realizado neste trabalho, foram criados cenários que possibilitam a observação do comportamento dos pares através da métrica *share ratio* ou taxa de compartilhamento (TC), que é dada pela razão do total enviado sobre o total baixado para cada arquivo transferido. Essa métrica mede o nível 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. A Tabela 3.1 apresenta os tamanhos dos arquivos compartilhados em cada cenário. Nos cenários 01, 02 e 03 são compartilhados, respectivamente, um arquivo de 250 MB, 500 MB e 1024 MB.

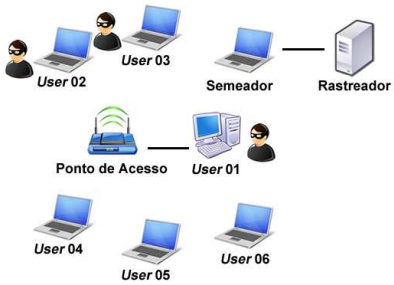
Tabela 3.1: Dados dos cenários estudados.

	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

Cada cenário foi subdivido em 3 casos distintos onde cada caso apresenta uma quantidade diferente de usuários desonestos no enxame. Os casos representados nas Figuras 3.2(a) e 3.2(b) mostram situações onde ocorrem a ação de pares desonestos. Já a Figura 3.2(c), representa o caso onde apenas usuários honestos participam do enxame. Dessa forma é possível caracterizar tanto o comportamento dos pares desonestos, quanto dos pares honestos.



(a) 1º Caso



(b) 2º Caso

(c) 3º Caso

Figura 3.2: Disposição dos pares nos três casos dos cenários estudados.

Os clientes desonestos são configurados previamente e nessa configuração são passados como parâmetros o arquivo .torrent (do respectivo enxame), o tamanho do arquivo e as taxas de *download* e *upload*. A Tabela 3.2 apresenta a configuração da largura de banda dos usuários *User01*, *User02* e *User03* quando os mesmos possuíam comportamento desonesto nos cenários avaliados.

Tabela 3.2: 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 DOS CENÁRIOS ESTUDADOS

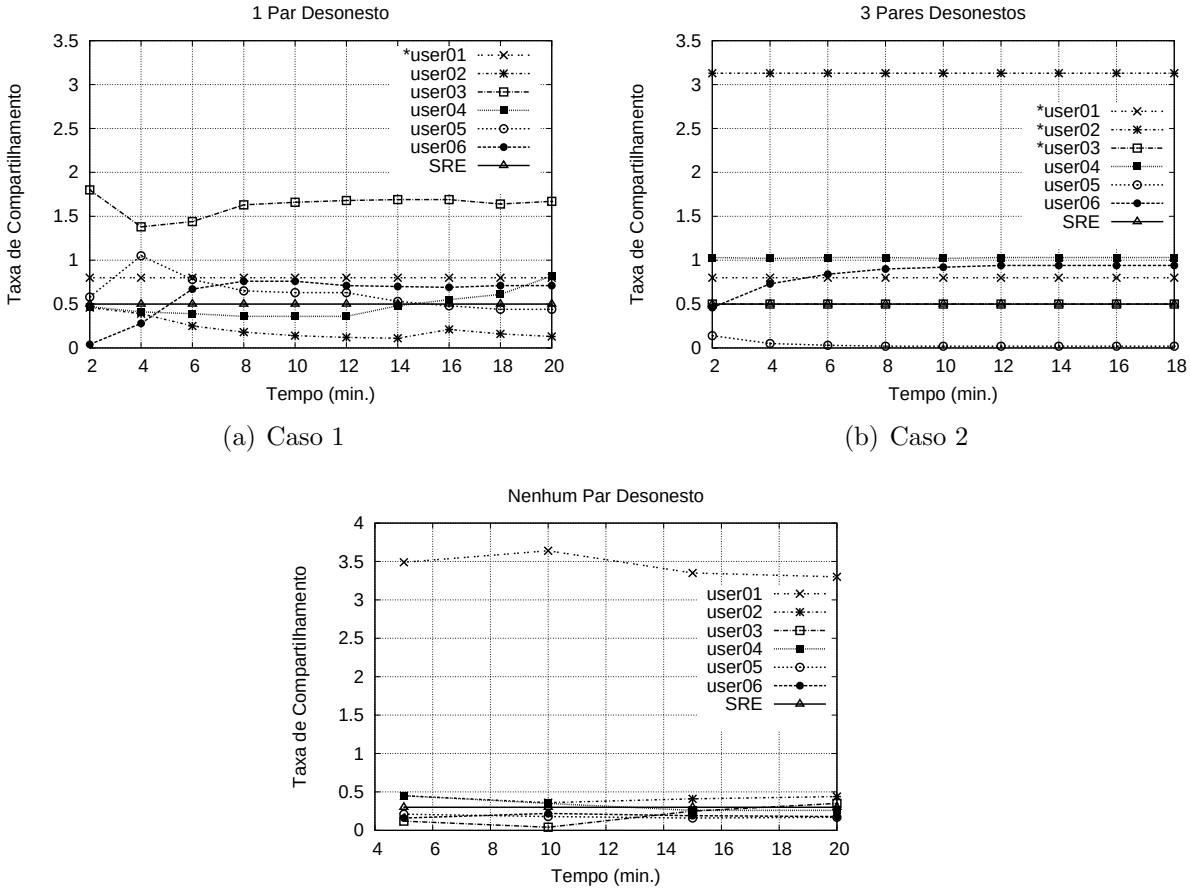
Esta seção estuda o comportamento da taxa de compartilhamento dos pares da rede privada BitTorrent definida. O rastreador coleta as informações necessárias em sua base de dados de tempos em tempos. A coleta de dados se inicia alguns minutos após o *Announce* do rastreador e termina quando todos os pares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

3.3.1 Análise Preliminar

A análise preliminar é composta da observação da TC dos usuários no tempo em um enxame do μ*Mundo*. A Figura 3.3 apresenta a taxa de compartilhamento de todos os usuários nos três casos estudados do cenário 01. Na Figura 3.3(a) é possível perceber 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. O semeador inicial é um par administrador da comunidade privada conhecido pelo rastreador. Desta forma, a evolução de sua TC, no tempo, não foi apresentada nos gráficos desta seção.

Os usuários honestos (*User02* e *User05*) terminaram suas transferências com suas res-

pectivas 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 3.3(b) mostra que os 3 usuários desonestos (*User01*, *User02* e *User03*) possuem configuração que lhes permitem 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 3.3(c) mostra que no cenário sem pares desonestos a TC de todos eles apresentou variabilidade perceptível.



(a) Caso 1

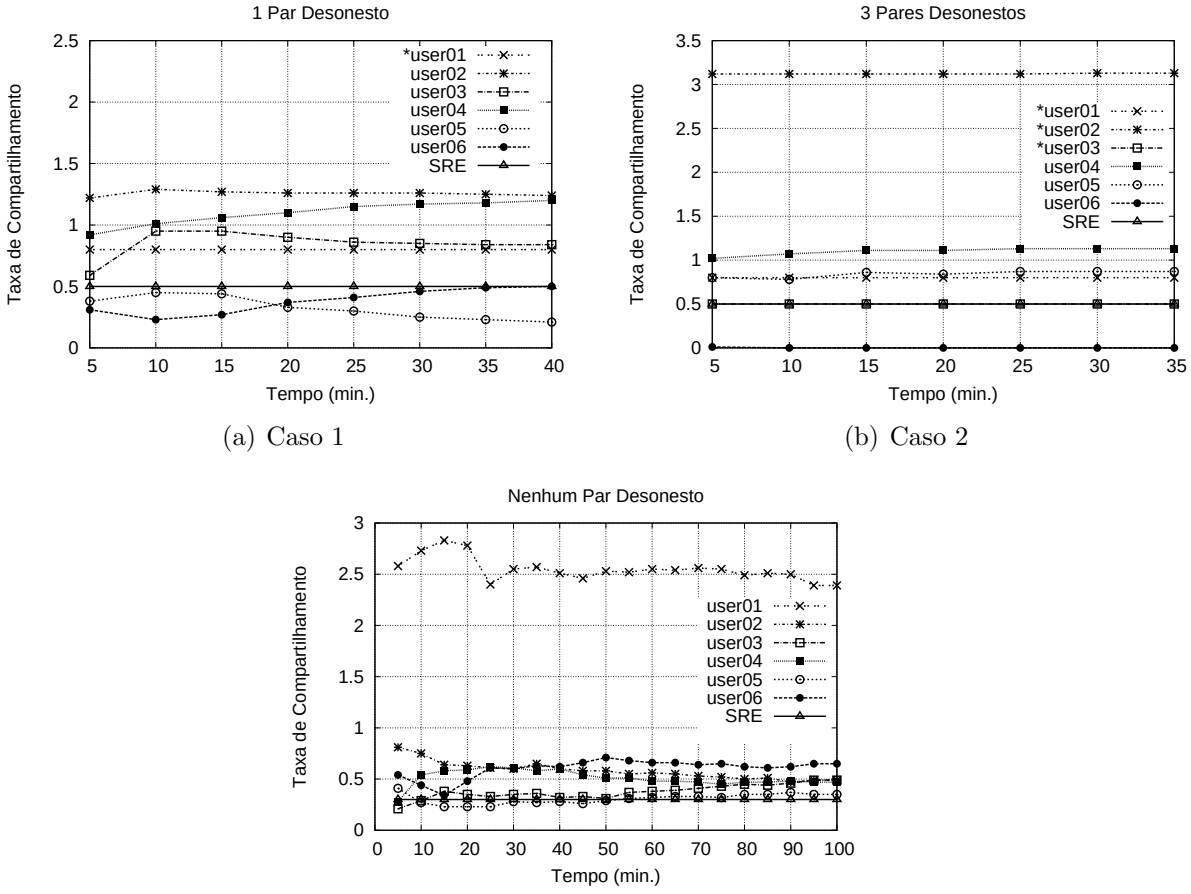
(b) Caso 2

(c) Caso 3

Figura 3.3: Comportamento da TC para os casos 1, 2 e 3 no Cenário 01.

A Figura 3.4 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 3.4(a) mostra que o usuário desonesto (*User01*) possui TC maior que a taxa SRE da rede. Apenas 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 3.4(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 3.4(c) mostra que no cenário sem pares desonestos a TC de todos eles apresentou variabilidade perceptível.



(a) Caso 1

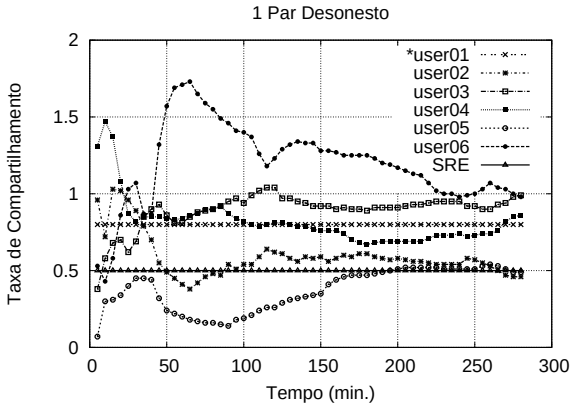
(b) Caso 2

(c) Caso 3

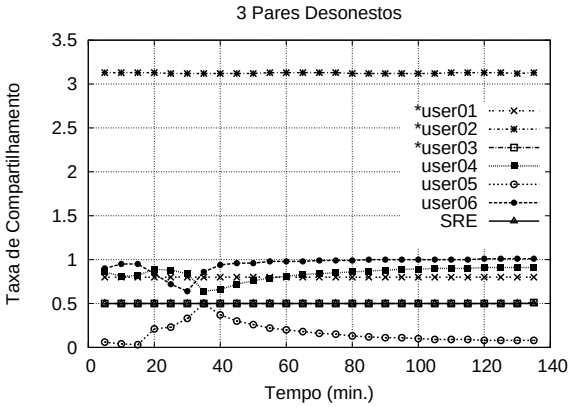
Figura 3.4: Comportamento da TC para os casos 1, 2 e 3 no Cenário 02.

A Figura 3.5 apresenta o comportamento da TC dos usuários para os casos 1,2 e

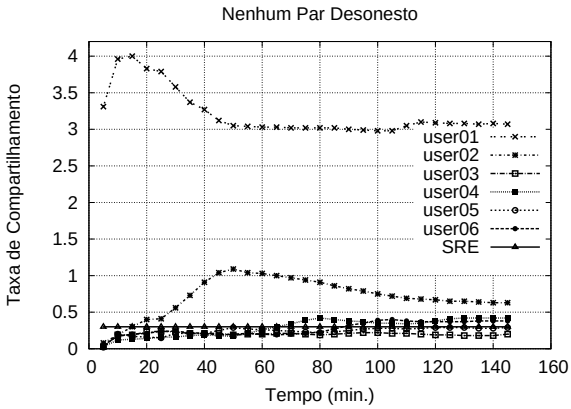
3 no cenário 03. Dessa vez, é feito o compartilhamento de um arquivo de 1024 MB para permitir aumentar ainda mais o número de amostras de TC ao longo do tempo. A Figura 3.5(a) mostra que apenas o usuário honesto *User02* termina suas transferências com TC abaixo da SRE estabelecida e todos os usuários honestos apresentam TC com variabilidade perceptível ao longo do tempo. Já a TC do usuário desonesto (*User01*) é aparentemente fixa. A Figura 3.5(b) mostra que os 3 usuários desonestos continuam com TC aparentemente fixa apesar do aumento na quantidade de amostras.



(a) Caso 1



(b) Caso 2



(c) Caso 3

Figura 3.5: Comportamento da TC para os casos 1, 2 e 3 no Cenário 03.

Observa-se, também, uma variabilidade na TC dos usuários honestos *User04*, *User05* e *User06*. A Figura 3.5(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 com usuários desonestos apresentando uma TC com variabilidade perceptível 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*.

3.3.2 Análise Numérica

O comportamento inesperado da TC de alguns pares desonestos motivou o monitoramento do tráfego gerado no decorrer de uma transferência. Com os dados obtidos através dele é possível comparar com os dados obtidos através do *script* de coleta com os dados exibidos pelo rastreador e assim, descobrir o que motivou o comportamento não esperado.

Foi configurado um ambiente para coleta de tráfego com o objetivo de obter os relatórios enviados para o rastreador no decorrer de uma transferência. Ao obter esses relatórios seria possível verificar a causa exata que estaria provocando a variabilidade perceptível inesperada na TC de alguns usuários desonestos. O ambiente configurado utilizou a ferramenta de captura de tráfego *Wireshark* e um analisador denominado *ReportDumpAnalizer*, desenvolvido especificamente para rastrear os arquivos coletados pelo *Wireshark* em busca dos relatórios emitidos no enxame.

O Wireshark [Munz and Carle 2008], inicialmente conhecido por *Ethereal*, é um popular analisador de tráfego bastante difundido na Internet. Por ser uma ferramenta de código aberto, seu desenvolvimento evoluiu através de contribuições de desenvolvedores de todo o mundo. Para realizar capturas em tempo real, esse analisador se utiliza da biblioteca pcap (libpcap) [Dabir and Matrawy 2007] e de funções de análise. Suas funções de análise são responsáveis pela compreensão dos protocolos, interpretação e remontagem dos quadros, coleta de tráfego e geração de estatísticas. Os dados coletados podem ser salvos com diversas extensões para consultas futuras.

Para realizar análises de tráfego em tempo real é importante que o analisador esteja instalado e seja executado no mesmo computador em que faz a captura, para que os dados sejam processados pela interface libpcap. Em caso de escassez de processamento o

Wireshark pode ser implantado em um local estratégico da rede que funcione como um ponto de observação por onde passe todo o tráfego que deve ser analisado. Por exemplo, o analisador pode ser instalado em um computador que está conectado a uma porta de monitoramento de um *switch* ou roteador [Munz and Carle 2008].

A Figura 3.6, mostra a localização do Wireshark no *µMundo* para que a coleta do tráfego seja realizada de forma eficiente. O computador utilizado pelo semeador é o hospedeiro do ambiente virtual onde se encontra instalado o rastreador da comunidade *µMundo*. Existe um *switch* virtual por onde o tráfego destinado ao rastreador passa. Os relatórios e a sinalização para organização da rede são os principais componentes dele. Assim, seu fluxo se inicia a partir do computador do semeador, segue para o *switch* virtual e, finalmente, alcança o seu destino final.

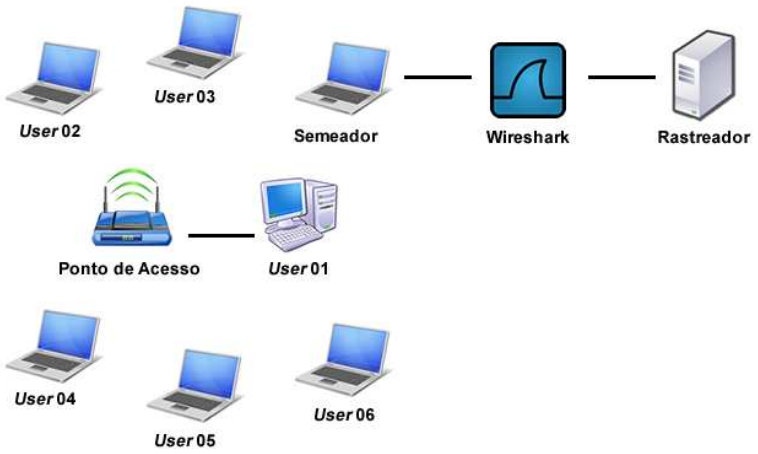


Figura 3.6: O *µMundo* com a presença do *Wireshark*.

A ferramenta *ReportDumpAnalizer* foi desenvolvida para rastrear os relatórios enviados pelos participantes de um enxame para o rastreador. Para isso, ela carrega um arquivo com o tráfego coletado (*dump*), por exemplo *coleta.csv*, e procura por requisições HTTP que contenham os campos *uploaded*, *downloaded*, *time*, e *Userid*. A partir desses dados é possível calcular a TC que foi enviada pelos pares, sem o tratamento do rastreador.

Assim, os experimentos do cenário 03 foram executados mais uma vez, porém usando como pares desonestos os usuários *User02*, *User03* e *User04*. A Tabela 3.3 mostra o conteúdo dos nove primeiros relatórios enviados pelo usuário desonesto (*User03*) ao ser-

vidor 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 3.3: 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 3.4 mostra a TC dos usuários desonestos *User02*, *User03* e *User04* calculada na implementação *Xbtit* do rastreador e a TC calculada truncada na terceira casa decimal. A primeira é representada pela nomenclatura **S** enquanto a segunda é representada pela nomenclatura **Tr**. 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 (**S**) considerando várias casas decimais possui pouca variabilidade (apenas a partir da terceira casa decimal). 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. Esta observação serve como base para a proposta do classificador denominado TbC (*Threshold-based Classifier*) que será apresentado próximo capítulo.

3.3.3 Análise Estatística

A observação da variabilidade da taxa de compartilhamento dos usuários em um intervalo de tempo constitui uma série temporal. Segundo [Hamilton 1994], uma série temporal pode ser definida como uma coleção de observações feitas sequencialmente ao longo do

Tabela 3.4: 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

tempo.

As séries temporais são usadas em vários campos do conhecimento, como Economia, Medicina, Meteorologia e Epidemiologia. Na economia, por exemplo, pode-se obter séries temporais a partir dos preços diários de ações e da taxa mensal de desemprego. Na medicina, através de eletrocardiogramas e eletroencefalogramas. No campo da epidemiologia, o número mensal de novos casos de uma doença e na meteorologia, a temperatura diária pode ser considerada um exemplo [Hamilton 1994].

Segundo [Stoffer and Shumway 2000], a análise dessas séries tem como objetivos principais, compreender o seu mecanismo gerador e predizer seu comportamento futuro. A compreensão do mecanismo gerador possibilita descrever efetivamente o comportamento, encontrar periodicidade e controlar a trajetória da série. Os resultados das análises preliminar e numéricas sugerem que a SRE dos pares desonestos formam processos estacionários. Neste trabalho, o estudo das séries temporais tem como objetivo principal compreender o comportamento da TC de um membro de uma comunidade privada em um intervalo de tempo, possivelmente através de variáveis auxiliares.

Uma série temporal é definida pelos valores Y_1, Y_2,...,Y_n de uma variável Y nos respectivos tempos t_1, t_2,...,t_n. Portanto, Y é uma função de t, ou seja, $Y=F(t)$. Neste estudo, Y corresponde à TC de um usuário em função de um intervalo de tempo t em minutos. Graficamente, essa série temporal pode ser representada por um gráfico de Y (TC) em função de t. A Figura 3.7 mostra uma série temporal descrita a partir dos valores da taxa de compartilhamento de dois usuários (*U1* e *U2*), de uma comunidade privada, no decorrer de uma transferência de arquivo [Stoffer and Shumway 2000].

Um processo estacionário é uma família $\{ Y(t), t \in \boldsymbol{T} \}$ tal que, $\forall\ t \in \boldsymbol{T}$, $Y(t)$ é uma

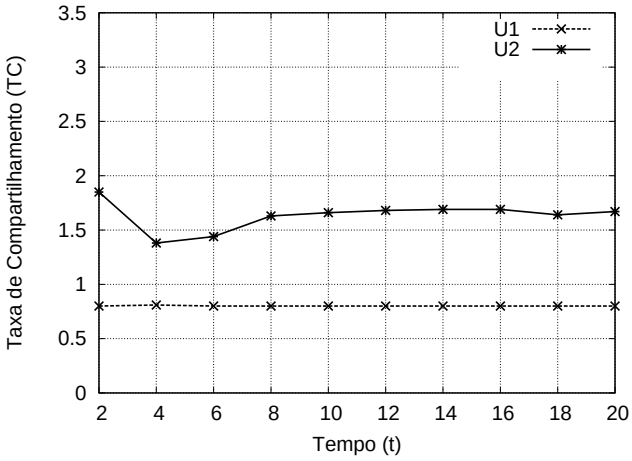


Figura 3.7: Representação gráfica de uma série temporal.

variável aleatória. Dessa forma, uma série temporal é um processo onde o conjunto de valores $\{ Y(t), t \in \boldsymbol{T} \}$ é chamado de espaço de estados e os valores de $Y(t)$ são chamados de estados. Cada valor $Y(t)$ apresenta uma distribuição de probabilidade que pode ser a mesma ou não [Stoffer and Shumway 2000].

Na prática, se torna difícil extrair informações de uma série a partir da definição formal apresentada acima pois é muito difícil especificar todas as distribuições finito-dimensionais nela presentes. Normalmente, o que se faz é concentrar nos primeiros momentos através da utilização da média $\mu(t)$ e da autocovariância $\gamma(t1,t2)$, pois essas funções definem um processo estacionário na prática [Stoffer and Shumway 2000].

A autocovariância descreve o quanto a variável avaliada varia em um intervalo de tempo [Priestley 1981]. Neste estudo, ela é utilizada para medir a variabilidade da (TC) de um usuário, como pode ser observado na Seção 3.3.1. Formalmente, ela pode se apresentar como, $\gamma(t1,t2) = E[(X_{(t)} - \mu_{(t)})(X_{(s)} - \mu_{(s)})]$. Onde E é o valor médio, esperança matemática ou expectativa da expressão. A variável aleatória correspondente a TC num instante t é representada por $X_{(t)}$, já a média $\mu_{(s)}$ representa as TCs de um usuário.

Com o intuito de calcular a autovariância para cada par participante de um enxame, foram executados experimentos com o objetivo de coletar elementos de amostras de pares através da taxa de compartilhamento (TC). Para a análise realizada nesta seção, foram executados mais 3 cenários de compartilhamento de arquivo. A Tabela 3.5 apresenta os

tamanhos dos arquivos compartilhados em cada cenário. Nos cenários 04, 05 e 06 são compartilhados, respectivamente, um arquivo de 1 GB, 3 GB e 4 GB. Os mesmos casos apresentados na Figura 3.2 foram utilizados nesta análise. Nela é possível perceber, em seu primeiro caso, a existência de um usuário desonesto, no segundo, a existência de 3 usuários desonestos e o último apenas pares honestos.

Tabela 3.5: Dados dos cenários estudados.

	Nº de Casos	**Nº de Pares**	**Tamanho do Arquivo (GB)**
Cenário 04	3	7	1
Cenário 05	3	7	3
Cenário 06	3	7	4

A Tabela 3.6 mostra os valores de autocovariância para todos os pares participantes de enxames executados em um cenário do *µMundo*. No primeiro caso *(Caso 1)*, o usuário desonesto apresentou um valor de autocovariância de zero. O restante dos usuários também apresentaram valores próximos de zero, porém, com uma autocovariância maior do que a apresentada pelo usuário desonesto *(User01)*. No segundo caso *(Caso 2)*, os três usuários desonestos (*User01*, *User02* e *User03*) apresentaram valores iguais ou muito próximos de zero. Já os usuários honestos mantiveram um nível de variabilidade perceptível, como no caso anterior. Observa-se ainda que um usuário honesto (*User04*) apresentou um valor negativo de autocovariância. Isso significa dizer que os valores da TC desse usuário obtiveram um decaimento em sua variabilidade. No terceiro caso *(Caso 3)* nenhum usuário honesto obteve autocovariância com valor zero.

Tabela 3.6: Autocovariância para os casos do Cenário 4.

	User01	**User02**	**User03**	**User04**	**User05**	**User06**
Caso 1	0,00000	0,00404	0,01955	0,00640	0,03400	0,01227
Caso 2	0,00000	7,16E-006	0,00000	0,00015	0,03588	0,05377
Caso 3	0,18987	0,01049	0,00071	0,00173	0,50896	0,02273

A Tabela 3.7 mostra os valores de autocovariância para todos os pares participantes do Cenário 5. O primeiro caso *(Caso 1)* apresentou um usuário desonesto com valor zero de autocovariância. Os usuários desonestos apresentaram uma autocovariância maior do que a apresentada pelo usuário desonesto presente *(User01)*. No segundo caso *(Caso 2)*, os três usuários desonestos (*User01*, *User02* e *User03*) continuaram apresentando valores

iguais ou muito próximos de zero. Os usuários honestos, como no caso anterior, obtiveram autocovariância maior. No terceiro caso *(Caso 3)* os usuários honestos apresentaram valores de autocovariância maiores que os apresentados pelos usuários desonestos nos casos anteriores.

Tabela 3.7: Autocovariância para os casos do Cenário 5.

	User01	**User02**	**User03**	**User04**	**User05**	**User06**
Caso 1	0,00000	0,00599	0,01096	0,00212	0,01169	0,03255
Caso 2	0,00000	1,316E-005	0,00000	0,04699	0,03850	0,03165
Caso 3	0,04114	0,00108	0,02968	0,00559	0,02286	0,18539

A Tabela 3.8 mostra os valores de autocovariância para os usuários envolvidos nos enxames do Cenário 5. O usuário desonesto *(User01)*, do *Caso 1*, apresentou mais uma vez o valor zero para autocovariância. Ainda neste caso, os usuários honestos apresentaram valores de autocovariância maior do que a apresentada pelo usuário desonesto. Os três usuários desonestos (*User01*, *User02* e *User03*) apresentaram valores iguais ou muito próximos de zero, já os usuários honestos mantiveram, mais uma vez, níveis perceptíveis de variabilidade. No terceiro caso *(Caso 3)*, nenhum dos usuários honestos obteve autocovariância com valor zero.

Tabela 3.8: Autocovariância para os casos do Cenário 6.

	User01	**User02**	**User03**	**User04**	**User05**	**User06**
Caso 1	0,00000	0,22016	0,03695	0,00431	0,06753	0,73104
Caso 2	0,00000	1,932E-005	0,00000	0,01039	0,00401	0,00116
Caso 3	0,33463	0,05096	0,00540	0,00147	0,66396	0,02255

O valor da autocovariância indica o grau de dependência entre as TCs sucessivas de cada usuário. Ou seja, esse pode ser utilizado para medir a influência que uma taxa de compartilhamento inicial terá em uma próxima TC.

Os resultados expostos nesta análise mostram que usuários desonestos apresentam valores de autocovariância significativamente menores do que os usuários honestos em um enxame privado. Na maioria dos casos esse valor chegou a ser zero ou bem próximo dele. Os resultados obtidos nesta seção também motivam o desenvolvimento de um classificador de pares desonestos baseado nos valores da autocovariância obtidos a partir das taxas de compartilhamento dos usuários de uma comunidade BitTorrent privada.

3.4 RESUMO

O *μMundo* foi o ambiente utilizado para a coleta de amostras da taxa de compartilhamento dos participantes de um enxame privado. Nele, foram implantados, um rastreador com implementação *Xbtit*, um banco de dados MySQL e um *script* para coleta de amostras desenvolvido em PHP. Os pares honestos se caracterizaram pela utilização do cliente μTorrent e os pares desonestos utilizaram o cliente malicioso *RatioMaster*. A criação desse ambiente possibilita a observação do comportamento dos pares honestos e desonestos. As observações realizadas no *μMundo* foram compostas por diversas transferências de arquivos em diversos cenários controlados. Os pares foram submetidos à transferências enquanto suas TCs eram coletas e armazenadas em uma tabela de experimentos denominada *experiments*.

Com essas informações é possível observar as características apresentadas por cada um dos pares participantes do enxame. Os resultados da análise inicial mostraram indicativos de que os usuários honestos apresentam uma variabilidade perceptível em suas TCs ao longo do tempo. Foi observado, também, que a variabilidade das TCs para os pares desonestos não foi perceptível ou quase não existiu. Contudo, alguns pares honestos também apresentaram falta de variabilidade em suas TCs e, somado a isso, alguns nós desonestos apresentaram pequena variações inesperadas em suas TCs. Os resultados obtidos na análise preliminar 3.3.1, mostram que a variabilidade da taxa de compartilhamento dos pares em uma comunidade privada BitTorrent, pode ser utilizada como critério de decisão para a classificação de pares de pares desonestos.

Os resultados apresentados na análise preliminar motivaram um segundo estudo, chamado de análise numérica (Seção 3.3.2). O seu objetivo principal é comparar a TC calculada a partir dos relatórios enviados pelos pares desonestos, com a TC obtida dos relatórios já processados e exibidos na comunidade privada pelo rastreador. Essa comparação mostrou que o que levou os pares desonestos a obterem variações inesperadas na TC de seus usuários foi um arredondamento *Round Half Up*, realizado pelo rastreador ao calcular a TC de casa usuário, além disso ele também considera apenas duas casas decimais para a exibição da TC dos usuários. Assim, foi introduzido um limiar máximo

para a variabilidade da TC dos pares desonestos ($\varepsilon = 0.01$).

A utilização do limiar ε é importante pois fornece uma medida concreta capaz de indicar o comportamento de um par desonesto. No entanto, o uso de métricas auxiliares, ou que indiquem claramente o comportamento malicioso, ajudam na criação de classificadores eficientes. A busca por métricas auxiliares motivou a análise estatística, apresentada na Seção 3.3.3. Essa análise se baseia em conceitos da teoria de séries temporais para introduzir a função de autocovariância. O valor obtido através dessa função fornece indicativos significativos que podem ser utilizados para classificar pares como desonestos. Foi observado que os pares que possuem valores de autocovariância muito próximos de zero podem ser considerados pares maliciosos.

As análises apresentadas neste capítulo fornecem importantes indicativos relacionados com o comportamento dos pares, que podem ser utilizados por classificadores para detectar, de fato, pares que estão burlando o mecanismo de incentivo de uma comunidade privada. Isso motivou a criação de classificadores para tal finalidade. No próximo capítulo serão apresentados os classificadores propostos e uma avaliação de desempenho baseada na taxa de acerto obtida por eles.

CAPÍTULO 4

CLASSIFICADORES PROPOSTOS

As análises apresentadas no capítulo anterior fornecem indicativos de que a variabilidade das TCs, dos pares participantes de enxames privados, pode ser utilizada para classificá-los em desonestos ou honestos. Este capítulo apresenta dois classificadores propostos com base nesta observação. O primeiro, chamando TbC (*Threshold-based Classifier*), baseia-se no limiar ε, para decidir se o par é honesto. O segundo, denominado AbC (*Autocovariance-based Classifier*), classifica os pares de uma comunidade privada baseando-se na autocovariância das TCs obtida por cada um deles. Ainda neste capítulo, são mostrados os desempenhos obtidos pelos dois classificadores de acordo com suas taxas de acerto, falsos positivos e falsos negativos.

4.1 CLASSIFICADOR TBC (THRESHOLD-BASED CLASSIFIER)

Os estudos das Seções 3.3.1 e 3.3.2 mostram que o comportamento da TC dos usuários desonestos apresenta falta de variabilidade ou uma variabilidade não perceptível graficamente. A partir de algumas coletas de tráfego foi introduzido o limiar máximo de variabilidade para pares desonestos, representado por ε. Na prática, os pares que porventura ultrapassarem este limiar devem ser considerados honestos, caso contrário podem ser ditos desonestos.

Com base nesta constatação, foi desenvolvido um algoritmo classificador automático capaz de analisar a taxa de compartilhamento de um usuário no decorrer do tempo e classificar se o par é desonesto caso a variabilidade de sua TC ultrapasse o limiar ε. Deste modo, o algoritmo proposto parte do pressuposto de que todos os usuários de um enxame são desonestos e a partir disso inicia a verificação da taxa de compartilhamento

de cada usuário conectado.

O Algoritmo 4.1 apresenta o pseudo código do classificador proposto. Ele 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:

$$y_c = \Delta tc \ , \tag{4.1}$$

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 (como exposto nas linhas 14, 15 e 16).

Algoritmo 4.1 Pseudo Código do Algoritmo Classificador TbC.

```
 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;
```

Com o intuito de exemplificar o funcionamento do algoritmo TbC proposto, a Figura 4.1 mostra a verificação das TCs dos usuários *User01* e *User02*. Primeiramente, o algoritmo inicia a variável TC_a (TC anterior) pegando o valor da primeira TC de cada

usuário presente no enxame. Em seguida, ele busca a próxima TC existente, a armazena na variável TC e calcula a diferença entre elas ($yc = TC - TC_a$). Com o valor de yc (ou Δtc), ele verifica se o valor absoluto da diferença calculada é maior do que ε. O exemplo mostra que o usuário *User01* possui um valor yc menor do que o limiar ε, ou seja, $\Delta tc < \varepsilon$. O exemplo mostra também, que o algoritmo, em um primeiro momento, considera todos os participantes do enxame como sendo desonestos.

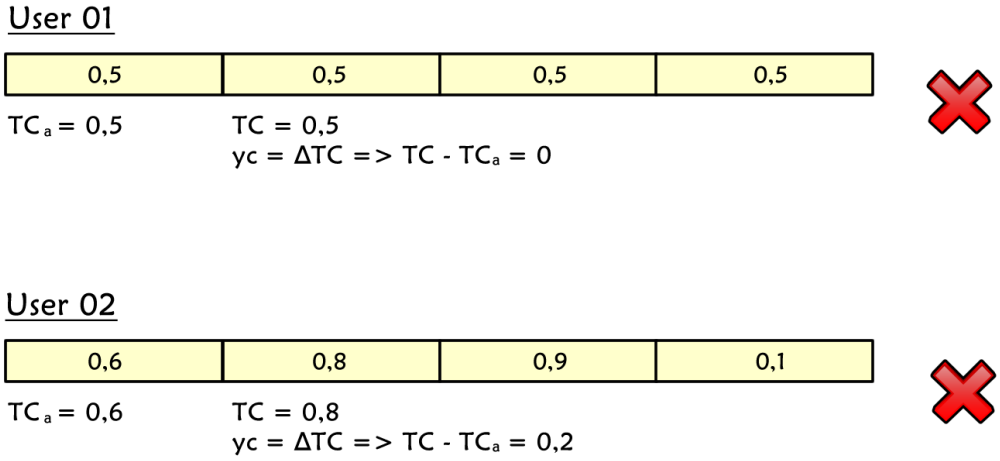


Figura 4.1: a) Exemplo de funcionamento do algoritmo TbC.

A medida que vai calculando as diferenças de TC, o algoritmo TbC passa a utilizar a variável $Diff$ para armazenar a nova diferença entre TCs sem que a diferença calculada anteriormente seja sobrescrita. Por obter uma variação de TC abaixo de ε, o usuário *User01* é classificado como desonesto, como mostra a Figura 4.2. Uma vez verificado que um determinado usuário apresentou uma diferença entre TCs, maior do que o limiar ε, ele passa a ser classificado como honesto, mesmo que posteriormente passe a apresentar uma variabilidade não perceptível em sua TC. E assim a variabilidade da TC de cada usuário presente vai sendo verificada.

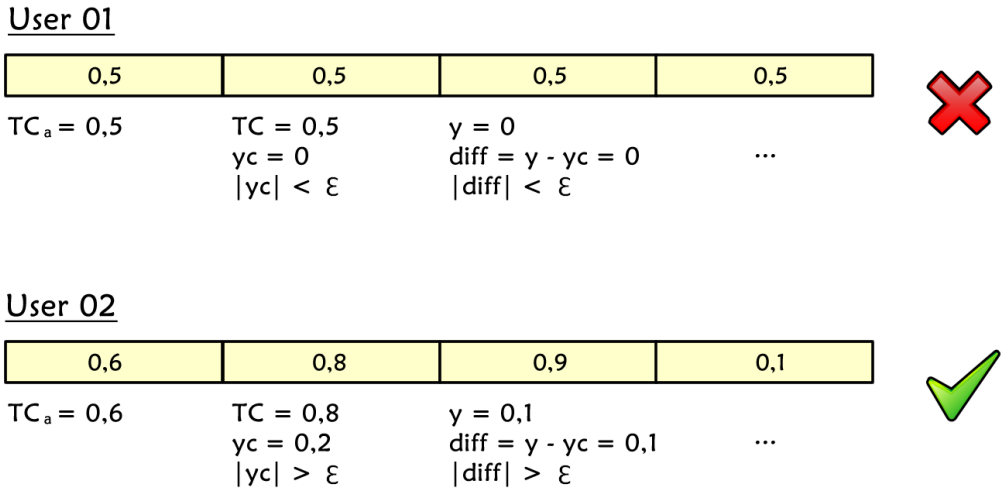


Figura 4.2: b) Exemplo de funcionamento do algoritmo TbC.

4.2 CLASSIFICADOR ABC (AUTOCOVARIANCE-BASED CLASSIFIER)

Embora o algoritmo TbC seja capaz de identificar pares desonestos que estão apresentando uma variabilidade não perceptível, compatível com o limiar ε, existem casos que usuários considerados honestos, apresentam baixo rendimento de *upload* e consequentemente uma TC com variabilidade não perceptível. Desta forma são classificados erroneamente como pares desonestos. Então, é importante considerar diferentes métricas além do limiar ε para que falsos positivos não sejam gerados. Com esse intuito, o algoritmo proposto nesta seção se baseia em valores de autocovariância da TC para classificar os pares de uma comunidade BitTorrent privada em honestos ou desonestos.

Através do estudo sobre séries temporais, apresentado na análise estatística da Seção 3.3.3, foi mostrado que os valores de autocovariância obtidos podem ser utilizados para representar o nível de variabilidade das TCs de um usuário de comunidade privada BitTorrent. Com base nessa informação, esta seção apresenta um segundo algoritmo classificador, denominado AbC (*Autocovariance-based Classifier*). Sua principal diferença para o TbC é a utilização da função de autocovariância que pode ser calculada para cada usuário de um enxame. Essa função define um usuário desonesto quando obtém

O Algoritmo AbC também considera todos os usuários de um enxame como desonestos

até que se infira o contrário. Em seu funcionamento básico, ele busca os usuários do enxame em questão com o objetivo de calcular o valor da autocovariância para cada um deles. Ao identificar todos os usuários, o algoritmo proposto AbC, processa a média amostral das TCs e a armazena na variável *average*. Depois disso, verifica a quantidade de amostras, contabiliza seu tamanho e o atribui à N. Em seguida, utiliza os valores obtidos através do produto entre as diferenças das TCs (inicial e final) e a média guardada na variável *average* para realizar o somatório que é sempre alocado em *sum*. Por fim, é calculado o valor da autocovariância através da razão entre *sum* e N. Esse cálculo é baseado na seguinte fórmula: $C = \frac{\sum_{i=1}^{N-1} (x_t - \overline{x}) * (x_{t+1} - \overline{x})}{N}$. Onde, a média amostral é representada por $\overline{x}$, a variável x_t corresponde aos elementos de amostra no tempo, ou seja, as TCs no tempo t, e N é o tamanho da amostra.

Os valores de autocovariância obtidos (armazenados em C) são processados em uma função de *Shift*, onde eles são multiplicados por 10^4 para que suas vírgulas sejam deslocadas 4 casas para a direita. Com isso, os pares que apresentarem autocovariância maior do que 1 devem ser classificados como honestos.

Algoritmo 4.2 Pseudo Código do Algoritmo Classificador AbC.

```
 1: users ← returnUsers();
 2: for i ← 0 to users.size() − 1 do
 3:     samples ← returnUserSamples(users[i]);
 4:     average ← returnAverage(users[i]);
 5:     N ← samples.size() − 1;
 6:     isColluder ← true;
 7:     C ← 0;
 8:     for j ← 0 to N do
 9:         shareRatio ← samples[j].getShareRatio();
10:         shareRatioAfter ← samples[j + 1].getShareRatio();
11:         sum ← sum + (shareRatio − average) ∗ (shareRatioAfter − average)
12:     C ← sum/N;
13:     acv ← shift(C, Right, 4);
14:     if acv > 1 then
15:         isColluder ← false;
```

A Tabela 4.1 mostra os valores de autocovariância obtidos nos experimentos da seção 3.3.3, já processados pela função de *Shift* nos casos do Cenário 4. É possível perceber que os pares desonestos obtêm valores zero ou muito próximos de zero, o que indica que quase não possuem variabilidade em suas TCs ao longo do tempo. No primeiro

caso (*Caso 01*), é possível perceber que apenas o usuário *User01* obteve valor abaixo de 1 e portanto é considerado desonesto. No segundo caso, são classificados como desonestos, os usuários *User01*, *User02* e *Use03*. Note que os três obtiveram autocovariância com o valor zero ou muito próximo dele. Isso significa que quase não apresentaram variabilidade ou apresentaram variabilidade pouco significativa. O último caso (*Caso 3*) mostra que todos os participantes do enxame privado obtiveram valores maiores que 1. Isso indica um cenário onde apenas pares honestos estavam presentes.

Tabela 4.1: Autocovariância com *shift* para os casos do Cenário 4.

	User01	**User02**	**User03**	**User04**	**User05**	**User06**
Caso 1	0,0	40,4	195,5	64,0	340,0	122,7
Caso 2	0,0	0,07	0,0	1,5	358,8	537,7
Caso 3	1898,7	104,9	7,1	17,3	5089,6	227,3

A Tabela 4.2 também mostra os valores da autocovariância já processados, ou seja, com a função de *shift* aplicada nos casos do cenário 5 (exposto na seção 3.3.3). É possível perceber que os valores de autocovariância obtidos apresentaram o comportamento esperado tanto para os pares honestos, quando para os pares desonestos. No *Caso 1*, o *User01* obteve valor igual a zero. No segundo, *Caso 2*, os usuários configurados com o *RatioMaster* (*User01*, *User02* e *User03*) obtiveram, novamente, valores abaixo de 1 e muito próximos de zero, ou mesmo o próprio valor de autocovariância zero. No último caso nenhum par obteve valor abaixo de 1.

Tabela 4.2: Autocovariância com *shift* para os casos do Cenário 5.

	User01	**User02**	**User03**	**User04**	**User05**	**User06**
Caso 1	0,0	59,9	109,6	21,2	116,9	325,5
Caso 2	0,0	0,131	0,0	469,9	385,0	316,5
Caso 3	411,4	10,8	296,8	55,9	228,6	1853,9

A Tabela 4.3 ratifica o comportamento já observado também para os casos do cenário 6 (exposto na seção 3.3.3). Todos os usuários que obtiveram autocovariância abaixo de 1, para qualquer um dos três casos apresentados, podem ser considerados desonestos. Já os pares que obtiveram valores acima são ditos honestos, por apresentar variabilidade perceptível de suas TCs.

Tabela 4.3: Autocovariância com *shift* para os casos do Cenário 6.

	User01	**User02**	**User03**	**User04**	**User05**	**User06**
Caso 1	0,0	1,6	369,5	43,1	675,3	7310,4
Caso 2	0,0	0,193	0,0	3,9	40,1	11,6
Caso 3	6,3	9,6	4,0	4,7	9,6	5,5

4.3 AVALIAÇÃO DOS CLASSIFICADORES PROPOSTOS

Com o intuito de avaliar o desempenho dos algoritmos classificadores TbC e AbC, foram utilizados alguns conceitos da análise ROC (*Reciever Operating Characteristics*). Através desta técnica é possível avaliar e eleger bons classificadores baseado em seus desempenhos. Para realizar estas análises, são utilizados gráficos ROC e métricas capazes de mostrar as taxas de acertos e falsos positivos dos classificadores avaliados [Prati et al. 2008].

Um classificador atribui um objeto a uma categoria ou classe pré-definidas. No caso deste trabalho, os algoritmos propostos classificam pares em duas categorias, gerando assim, resultados discretos que indicam apenas a classe do par. Ou ele será classificado como honesto ou como desonesto.

Considerando a teoria da análise ROC, temos: um conjunto de amostras, uma instância I, que poderá assumir os valores p, n, *positivo* e *negativo* e um classificador. Ao ser executado, um classificador de duas classes pode gerar quatro situações distintas. Se a instância considerada positiva for classificada como positiva, conta-se como um **verdadeiro positivo** (TP). Caso esta mesma instância seja classificada como negativa, conta-se como um **falso negativo** (FN). No caso de uma instância, considerada negativa, ser classificada como negativa, tem-se então um **verdadeiro negativo** (TN). Por fim, se a mesma instância (negativa) é classificada como positiva conta-se como um **falso positivo**.

Uma forma de apresentar estatísticas para avaliação de um classificador é por meio de uma matriz composta pelos dados da classe prevista e a classe real das instâncias avaliadas. Essa matriz é conhecida como tabela de contingência, também chamada de matriz de confusão e está representada na Figura 4.3 [Prati et al. 2008].

Dado um conjunto de pares onde um par pode assumir os valores *honesto* e *desonesto*. Os classificadores propostos (TbC e AbC) entendem que todo par é desonesto até que o

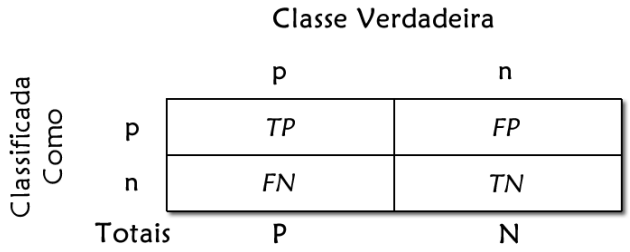


Figura 4.3: Matriz de contigência (ou confusão).

contrário seja inferido. Então, assumir que o par seja desonesto é uma instância positiva. Deste modo, ao classificar um par, podem ocorrer as seguintes situações:

1. Se o par é **Desonesto** (instância positiva) e classificado como **Desonesto** (instância positiva), conta-se como **verdadeiro positivo** (TP);

2. Se o par é **Honesto** (instância negativa) e classificado como **Honesto** (instância negativa), conta-se como **verdadeiro negativo** (TN);

3. Se o par é **Honesto** (instância negativa) e classificado como **Desonesto** (instância positiva), conta-se como **falso positivo** (FP);

4. Se o par é **Desonesto** (instância positiva) e classificado como **Honesto** (instância negativa), conta-se como **falso negativo** (FN).

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.

Existe uma série de métricas que podem ser calculadas a partir da matriz de contingência obtida. Podem ser destacadas a métrica *Precision*, que é a taxa de acerto do usuário, isto é, a quantidade de instâncias classificadas corretamente. As métricas TP_{rate} e FP_{rate} servem como base para a construção do espaço ROC. Por fim, a métrica Acurácia (do inglês, *Accuracy*) representa a taxa de acerto total do classificador. Ela é composta pela razão entre a soma dos acertos das duas classes e o número total de amostras existentes.

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 classificados corretamente e, além disso, verificar o quão eficiente são os classificadores propostos. 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.

O algoritmo TbC obteve uma taxa de acerto de 100% para todos os casos dos cenários (1 e 2) representados na Figuras 4.4(a) e 4.4(c). Isso indica que para tais cenários o algoritmo proposto identificou os pares desonestos corretamente sem que FP_s fossem levantados. Ainda na Figura 4.4(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.

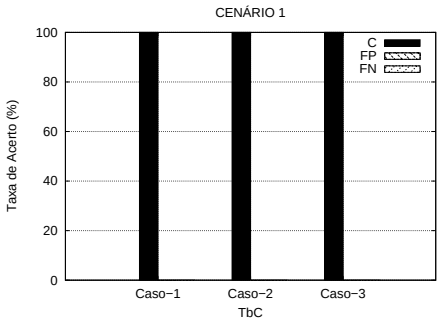
$$TA = \frac{(TP\ +\ TN)}{Total\ de\ Pares} \tag{4.2}$$

Os resultados obtidos foram dispostos em histogramas que representam a porcentagem da taxa de acerto, de falsos positivos e de falsos negativos. Nas Figuras 4.4(a), 4.4(b) e 4.4(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.

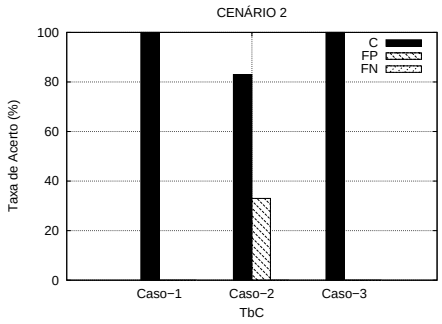
Apenas um caso do Cenário 2 apresentou falso positivo. Para esse caso específico, a taxa de acerto do algoritmo proposto foi de 83%, o que indica que os pares desonestos foram identificados, porém, um usuário (o par *User06*) foi indevidamente classificado como desonesto. Esse falso positivo pode ser 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 3.4(a) e 3.4(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 classificador TbC obteve uma taxa de acerto de 100%. Assim, todos os pares desonestos e honestos foram corretamente

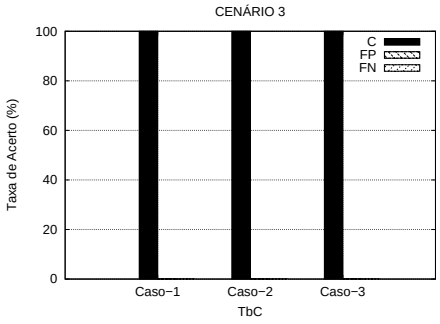
classificados. Analisando os gráficos das Figuras 4.4(a), 4.4(b) e 4.4(c), é possível observar que não ocorreram falsos negativos em nenhum dos casos estudados. Isso é importante, pois este resultado garante que, para os cenários estudados, pares desonestos não são classificados como honestos.



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



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



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

Figura 4.4: Taxas de acertos, falsos positivos e falsos negativos para o classificador TbC.

Além da taxa de acerto, exemplificada através das Figuras 4.4(a), 4.4(b) e 4.4(c), existem os valores de falsos positivos, de verdadeiros negativos, de verdadeiros positivos e de falsos negativos. Quando apenas um desses valores é observado, informações importantes podem ser descartadas, como por exemplo, o comportamento do erro ou quantidade de verdadeiros positivos. Essas métricas servem para avaliar o desempenho de um classificador no espaço ROC.

A curva ROC é um gráfico bidimensional, onde o eixo Y corresponde aos valores de verdadeiros positivos (TP) e o eixo X o valor de falsos positivos (FP). Um ponto (TP, FP) representa um classificador no espaço ROC. Neste trabalho, os algoritmos TbC e AbC são classificadores discretos, ou seja, são capazes de classificar apenas duas classes.

A Figura 4.5 apresenta o espaço ROC e a disposição de alguns classificadores.

Alguns pontos devem ser destacados no espaço ROC como mostra a Figura 4.5 . O ponto inferior esquerdo $(0, 0)$, representa uma estratégia que nunca gera uma classificação positiva. Um classificador que segue esse modelo não gera nenhuma falso positivo, como também, não é capaz de identificar nenhum verdadeiro positivo. A estratégia inversa é representada pelo ponto $(1, 1)$ que expressa um classificador que gera verdadeiros positivos incondicionalmente. O classificação ótima é representada pelo ponto $(0, 1)$. Esse caso pode ser representado pelo classificador D, ilustrado na Figura 4.5.

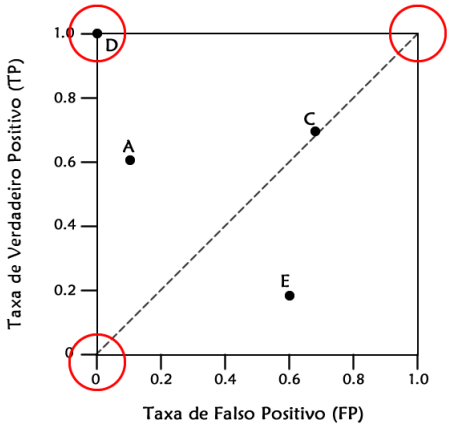
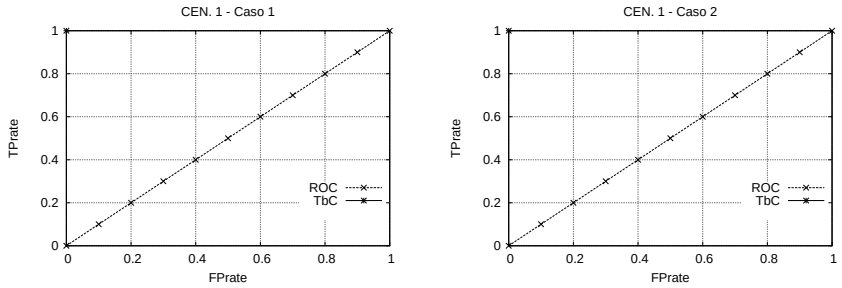


Figura 4.5: Espaço ROC.

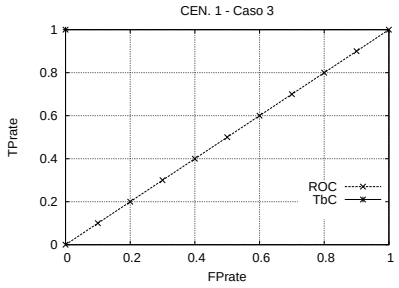
Os pontos que se encontram próximos ao lado esquerdo do gráfico ROC (próximos ao eixo Y) são ditos "conservadores" porque classificam positivamente apenas com fortes evidências, portanto cometem poucos falsos positivos FP. Já os classificadores localizados ao lado direito são ditos "liberais", pois fazem classificações gerando muitos falsos positivos. Na Figura 4.5, o classificador C é mais liberal que o A. De maneira geral, um ponto no espaço ROC é melhor do que outro se ele está à Noroeste do gráfico.

A Figura 4.6 apresenta a razão entre a taxa de falsos positivos e verdadeiros positivos obtida pelo classificador TbC em todos os casos do Cenário 1. É possível perceber que além da taxa de acerto de 100% demonstrada no gráfico da Figura 4.4(a), o classificador obtém bom desempenho sem gerar falsos positivos, apresentando um percentual de 100% de verdadeiros positivos. Esta observação demonstra uma classificação ótima, o que ratifica o bom desempenho do TbC para os casos do Cenário 1 estudados.



(a) Caso 1

(b) Caso 2



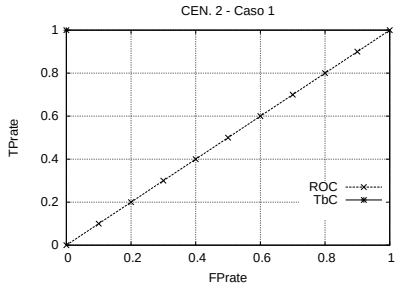
(c) Caso 3

Figura 4.6: Cenário 1 - Taxa de falsos positivos por verdadeiros postivos

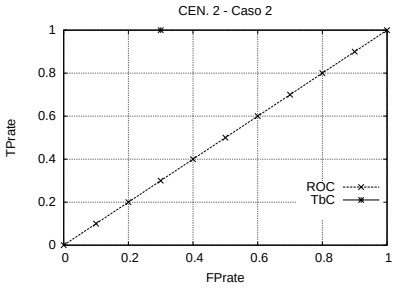
A Figura 4.7 apresenta o comportamento do TbC pela razão entre a taxa de falsos positivos e verdadeiros positivos nos casos do Cenário 2. O classificador obtém percentual de 100% de verdadeiros positivos nos casos 1 e 3, representados nas Figuras 4.6(a) e 4.6(c). Embora consiga boas taxas de verdadeiros positivos, o classificador gerou uma taxa de 33% de falsos de positivo, ratificando assim, o índice de FP demonstrado no gráfico 4.4(b). Esta observação demonstra que o TbC é capaz de identificar pares desonestos com uma boa taxa de acerto e verdadeiros positivos, embora apresente uma pequena taxa de falsos positivos.

A Figura 4.8 apresenta o desempenho ótimo obtido pelo classificador TbC em todos os casos do Cenário 3. Os gráficos indicam que ele foi capaz de classificar corretamente todos os pares desonestos sem que falsos positivos fossem gerados.

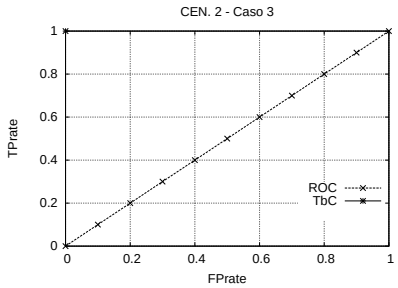
O algoritmo AbC obteve uma taxa de acerto de 100% para todos os casos dos cenários representados nas Figuras 4.9(a), 4.9(b) e 4.9(c). Isso indica que para tais cenários o algoritmo classificador AbC identificou os pares desonestos corretamente sem que FP_s fossem gerados. A taxa de acerto apresentada ratifica o que já foi mostrado nas Tabelas 4.1, 4.2



(a) Caso 1

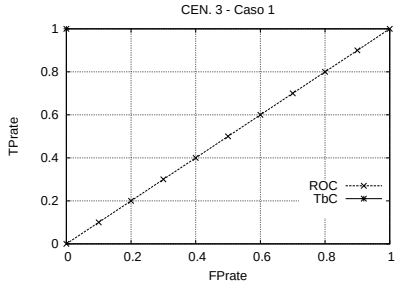


(b) Caso 2

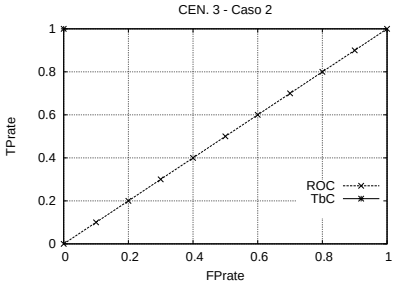


(c) Caso 3

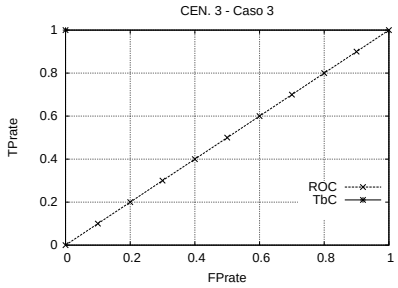
Figura 4.7: Cenário 2 - Taxa de falsos positivos por verdadeiros positivos



(a) Caso 1



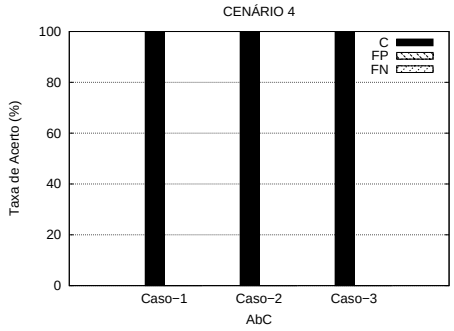
(b) Caso 2



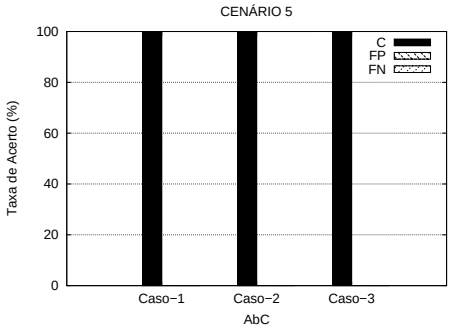
(c) Caso 3

Figura 4.8: Cenário 3 - Taxa de falsos positivos por verdadeiros positivos

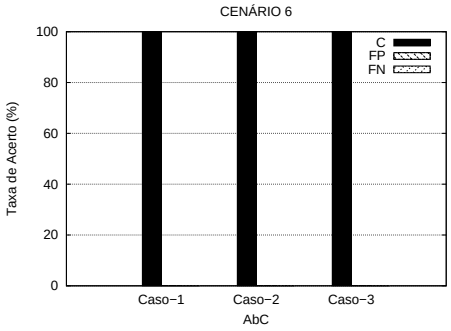
e 4.3.



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



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



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

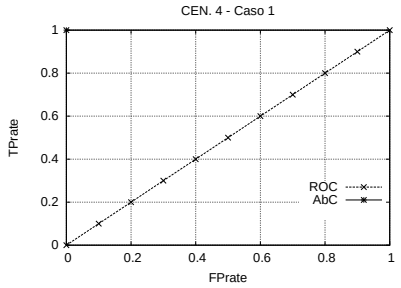
Figura 4.9: Taxas de acertos, falsos positivos e falsos negativos para o classificador AbC.

Os pares desonestos apresentaram autocovariância significativamente baixas e na maioria das vezes zero. Ainda analisando as Figuras 4.9(a), 4.9(b) e 4.9(c), é possível perceber que o algoritmo AbC obteve desempenho superior ao classificador TbC por não gerar falsos positivos em nenhum caso dos cenários estudados.

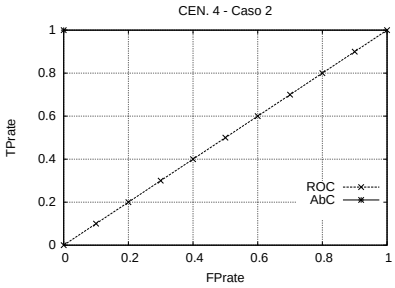
É possível perceber nas Figuras 4.10(a), 4.10(b) e 4.10(c) que o classificador AbC obteve o melhor desempenho possível nos casos do Cenário 1, de forma que nenhum falso positivo foi gerado.

Assim como nos casos do cenário anterior, o classificador AbC identificou corretamente todos os pares presentes nos enxames observados para o cenário 3, como ilustrado nas Figuras 4.11(a), 4.11(b) e 4.11(c). Ainda sobre o classificador AbC, as Figuras 4.12(a), 4.12(b) e 4.12(c) mostram que ele obteve uma taxa de verdadeiros positivos de 100%. Isso mostra que o melhor desempenho possível foi obtido.

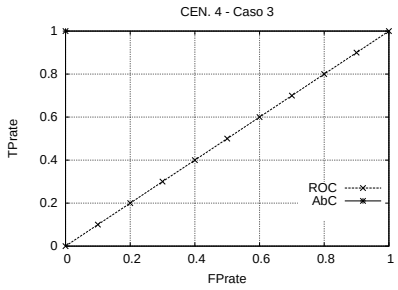
De maneira geral, as gráficos de curva ROC apresentaram que os dois classificadores



(a) Caso 1

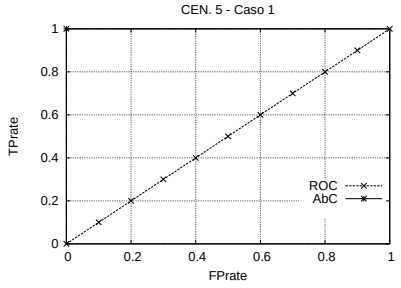


(b) Caso 2

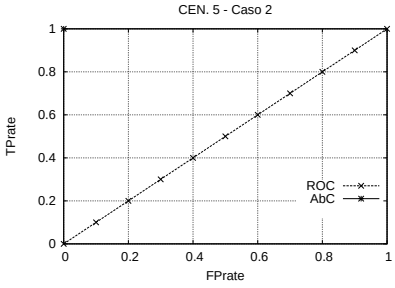


(c) Caso 3

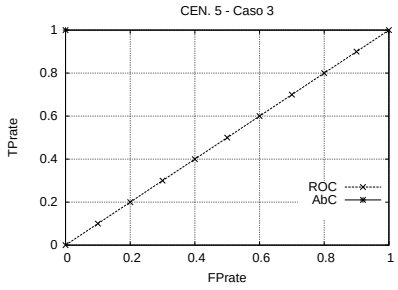
Figura 4.10: Cenário 4 - Taxa de falsos positivos por verdadeiros positivos



(a) Caso 1

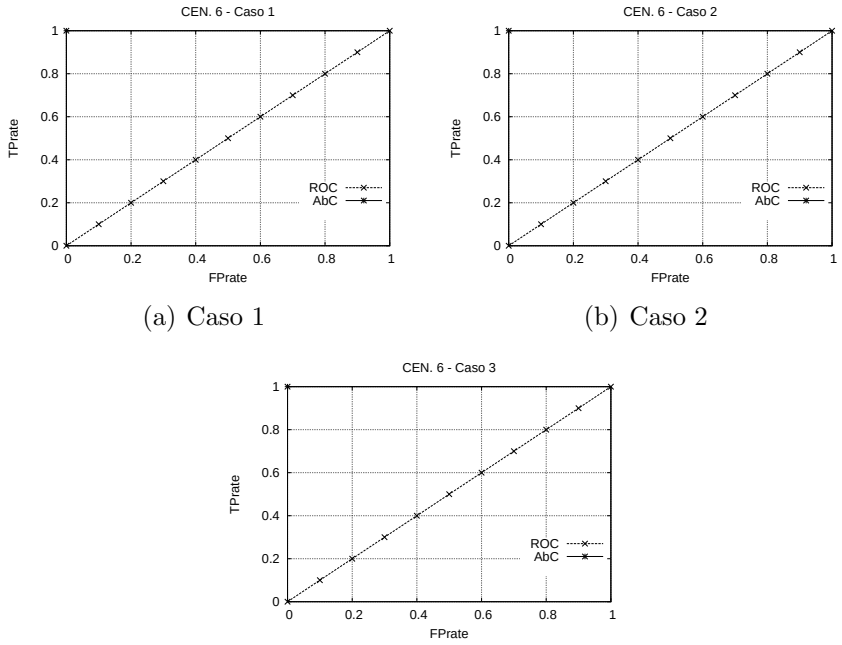


(b) Caso 2



(c) Caso 3

Figura 4.11: Cenário 5 - Taxa de falsos positivos por verdadeiros positivos



(a) Caso 1

(b) Caso 2

(c) Caso 3

Figura 4.12: Cenário 6 - Taxa de falsos positivos por verdadeiros positivos

propostos obtiveram boas taxa de acerto sem que falsos positivos fossem gerados. Indicando desta forma, que ambos são capazes de classificar como desonestos os pares que estão utilizando o *RatioMaster* para burlar o mecanismo de incentivo auxiliar de uma comunidade privada. Apesar de um deles, o classificador TbC, ter apresentado uma taxa de 33% de falsos positivos em um caso específico do segundo cenário (Cenário 2).

4.4 RESUMO

A detecção de pares desonestos que estão burlando o SRE de uma comunidade privada por através do uso do cliente BitTorrent malicioso *RatioMaster* motivou a criação de algoritmos classificadores. O primeiro deles, o TbC (*Threshold-based Classifier*), acompanha a variabilidade da TC de cada usuário de um enxame privado e a medida que ela ultrapassa um determinado limiar ε, o par é considerado honesto. Caso contrário, o nó é apontado como desonesto. Uma vez classificado como honesto, um par não será mais reclassificado posteriormente como desonesto caso não apresente variabilidade perceptível

de sua TC.

Com o intuito de minimizar a ocorrência de tais falsos positivos, foi proposto um novo classificador que baseia-se na função de autocovariância para decidir se o par é honesto ou não. Essa métrica aponta o nível de variação entre as TCs de um usuário. Análises apresentadas neste capítulo mostram que um par pode ser considerado malicioso caso apresente autocovariância zero ou muito próxima de zero.

A avaliação de desempenho dos dois classificadores propostos foi realizada seguindo alguns conceitos da análise ROC. Através desta técnica, é possível escolher classificadores de acordo com sua performance. Para isso, são utilizadas métricas que servem de indicadores para apontar as acurácia de um classificador. Neste trabalhado, foi utilizada a métrica *Accuracy*, que representa a taxa de acerto geral e é dada pela razão entre a soma de verdadeiros positivos TP com verdadeiros negativos TN, ou seja, a soma dos acertos das classes avaliadas, com o número total de amostras existentes.

Apesar de obter uma taxa de acerto relevante para a maioria dos cenários estudados, o algoritmo TbC, apresentou falso positivo para um caso onde um par honesto obteve uma variabilidade de TC próxima de zero. O algoritmo AbC (*Autocovariance-based Classifier*), obteve uma taxa de acerto de 100% sem que falsos positivos fossem levantados. Isso significa que o classificador AbC é capaz de identificar todos os pares maliciosos que estão utilizando o *RatioMaster* em uma comunidade privada.

CAPÍTULO 5

CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Pesquisas recentes mostram que as comunidades privadas BitTorrent oferecem um ambiente fechado, onde seus usuários conseguem melhores taxas de *download* devido a utilização de um mecanismo de incentivo auxilar (denominado SRE) e premiações para os pares que contribuem mais com a rede. Apesar disso, existe o interesse de alguns pares maliciosos em burlar tal mecanismo com o intuito de obter benefícios sem de fato estar contribuindo com a comunidade. Para isso, esses pares fazem uso de clientes maliciosos que adulteram seus relatórios e, consequentemente, aumentam sua taxa de compartilhamento em um curto espaço de tempo. Neste contexto, a detecção de pares que adulteram seus relatórios se faz necessária visto que tal ação degrada o bom desempenho não só do SRE, mas de toda a comunidade privada.

Este trabalho 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, foram propostos dois classificadores de pares. O primeiro deles, o TbC (*Threshold-based Classifier*), se baseou na variabilidade da taxa de compartilhamento (TC) dos pares da rede para classificá-los como honestos ou desonestos. Já o AbC (*Autocovariance-based Classifier*) utilizou a função de autocovariância para medir a variabilidade da TC dos usuários e assim definir se são ou não desonestos.

Os resultados apresentados sugerem que os classificadores propostos são eficientes na identificação de pares desonestos. O algoritmo classificador TbC apresentou, em dois dos três cenários estudados, uma taxa de acerto (TA) de 100%. Contudo, em um único cenário, a taxa de acerto foi de 83% devido à ocorrência de um falso positivo. O algoritmo

AbC mostrou que a medida estatística de autocovariância adotada atingiu uma taxa de acerto de 100% sem gerar falsos positivos.

Embora os resultados tenham apresentado que os classificadores TbC e AbC obtiveram bom desempenho nos cenários estudados há pontos de melhoria. Além do *RatioMaster* existem outros clientes BitTorrent maliciosos que precisam ser estudados. Com esse estudo é possível compreender melhor outros mecanismos geradores de falsos relatórios e consequentemente obter uma melhor caracterização do comportamento dos pares desonestos.

Em trabalhos futuros, a caracterização dos pares desonestos será expandida através do estudo de outras formas de manipulação da TC. Também serão estudadas formas de implantação dos algoritmos propostos em uma comunidade privada real na Internet visto que seus resultados indicam que é possível identificar os pares que estão utilizando o cliente BitTorrent malicioso *RatioMaster*.

REFERÊNCIAS BIBLIOGRÁFICAS

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

[Buford et al. 2009] Buford, J., Yu, H., and Lua, E. (2009). *P2P Networking and Applications*. Morgan Kaufmann Series in Networking. Elsevier/Morgan Kaufmann.

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

[Chen et al. 2010a] Chen, X., Chu, X., and Liu, J. (2010a). Unveiling Popularity of BitTorrent Darknets. Em *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1 –5.

[Chen et al. 2010b] Chen, X., Jiang, Y., and Chu, X. (2010b). Measurements, Analysis and Modeling of Private Trackers. Em *Proceedings of the Tenth IEEE International Conference on Peer-to-Peer Computing (P2P)*, pp. 1–10.

[Chen et al. 2012] Chen, X., Lin, K., Wang, B., and Yang, Z. (2012). Active Measurements on BitTorrent and eMule Ecosystem Over the Internet. Em *2nd International Conference on Consumer Electronics, Communications and Networks, (CECNet)*, pp. 126 –129.

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

[Cohen 2003] Cohen, B. (2003). Incentives build robustness in BitTorrent. Em *Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems (P2PeCon)*, volume 6, pp. 68–72.

[Cohen 2005] Cohen, B. (2005). BitTorrent Specification. Disponível em, *http://www.bittorrent.org./protocol.html* (Último acesso em 19/06/2012).

[Dabir and Matrawy 2007] Dabir, A. and Matrawy, A. (2007). Bottleneck Analysis of Traffic Monitoring Using Wireshark. Em *Proceedings of the 4th IEEE International Conference on Innovations in Information Technology (IIT'07)*, pp. 158–162.

[Dan and Carlsson 2012] Dan, G. and Carlsson, N. (2012). Centralized and Distributed Protocols for Tracker-Based Dynamic Swarm Management. *IEEE/ACM Transactions on Networking*, pp.(99):1.

[Fan et al. 2009] Fan, B., Lui, J., and Chiu, D.-M. (2009). The Design Trade-Offs of BitTorrent-Like File Sharing Protocols. *IEEE/ACM Transactions on Networking*, 17(2):365 –376.

[Hamilton 1994] Hamilton, J. (1994). *Time Series Analysis*. Cambridge Univ. Press.

[Harrison and Cohen 2008] Harrison, D. and Cohen, B. (2008). Fast Extension Draft - BitTorrent Enhancement Proposal BEP 6. Disponível em, *http://www.bittorrent.org/beps/bep-0006.html* (Último acesso em 23/06/2012).

[Jia et al. 2011a] Jia, A., D'Acunto, L., Meulpolder, M., and Pouwelse, J. (2011a). Modeling and Analysis of Sharing Ratio Enforcement in Private BitTorrent Communities. Em *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 1–5.

[Jia et al. 2011b] 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. Em *Proceedings of the IEEE International Conference on Peer-to-Peer Computing, (P2P)*, pp. 280–289.

[Jia et al. 2011c] Jia, A. L., Rahman, R., Vinko, T., Pouwelse, J., and Epema, D. (2011c). Sharing Ratio Enforcement: the Ultimate Solution for BitTorrent? Em *Proceedings of the First ICT ICT.Open*, pp. 14–15.

[Junemann et al. 2011] Junemann, K., Andelfinger, P., and Hartenstein, H. (2011). Towards a Basic DHT Service: Analyzing Network Characteristics of a Widely Deployed DHT. Em *Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, pp. 1 –7.

[Kash et al. 2012] Kash, I. A., Lai, J. K., Zhang, H., and Zohar, A. (2012). Economics of BitTorrent Communities. Em *Proceedings of the 21st ACM International Conference on World Wide Web (WWW)*, pp. 221–230, New York, NY, USA.

[Kaune et al. 2010] Kaune, S., Rumín, R. C., Tyson, G., Mauthe, A., Guerrero, C., and Steinmetz, R. (2010). Unraveling BitTorrent's File Unavailability: Measurements and Analysis. Em *Peer-to-Peer Computing*, pp. 1–9.

[Kryczka et al. 2011] Kryczka, M., Cuevas, R., Guerrero, C., Azcorra, A., and Cuevas, A. (2011). Measuring the Bittorrent Ecosystem: Techniques, Tips, and Tricks. *IEEE Communications Magazine*, 49(9):144 –152.

[Kurose and Ross 2008] Kurose, J. F. and Ross, K. W. (2008). *Computer Networking Third Edition a Top-Down Approach Featuring Internet*, volume 1. Pearson.

[Lehmann et al. 2011] Lehmann, M., Mansilha, R., Barcellos, M., and Santos, F. (2011). Swarming: como BitTorrent Revolucionou a Internet. Relatório Técnico disponível em, *http://www.inf.ufrgs.br/ rbmansilha/papers/minicursoBitTorrentJAI2011-CR.pdf* (Último acesso em 20/06/2012).

[Lian et al. 2007] 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. Em *Proceedings of 27th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp. 56-64.

[Liu and Shi 2010] Liu, L. and Shi, W. (2010). Trust and Reputation Management. *IEEE Internet Computing*, 14(5):10–13.

[Liu et al. 2010] 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 IEEE Conference on Distributed Computing Systems*, pp. 610–621.

[Lua et al. 2005] Lua, E. K., Crowcroft, J., Pias, M., Sharma, R., and Lim, S. (2005). A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. *IEEE Communications Surveys & Tutorials*, 7:72–93.

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

[Meulpolder et al. 2010] Meulpolder, M., D'Acunto, L., Capotă, M., Wojciechowski, M., Pouwelse, J., Epema, D., and Sips, H. (2010). Public and Private Bittorrent Communities: A Measurement Study. Em *Proceedings of the 9th International Conference on Peer-to-peer Systems*, pp. 10–14.

[Munz and Carle 2008] Munz, G. and Carle, G. (2008). Distributed Network Analysis Using TOPAS and Wireshark. Em *Proceedings of the IEEE Network Operations and Management Symposium Workshops (NOMS)*, pp. 161–164.

[Paiva and Gonçalves 2012] Paiva, P. G. F. and Gonçalves, P. A. S. (2012). Um Algoritmo de Classificação Automática de Pares Desonestos para Comunidades Privadas BitTorrent. Em *Anais do VIII Workshop de Redes Dinâmicas e Sistemas Peer-to-Peer (WP2P)*, pp. 31–43.

[Prati et al. 2008] Prati, R., Batista, G., and Monard, M. (2008). Curvas ROC para Avaliação de Classificadores. *Revista IEEE América Latina*, 6(2):215–222.

[Price 2011] Price, D. (2011). An Estimate of Infringing Use of the Internet. Disponível em, *http://documents.envisional.com/docs/Envisional-Internet-Usage-Jan2011.pdf* (Último acesso em 20/09/2012).

[Priestley 1981] Priestley, M. B. (1981). *Spectral Analysis and Time Series*, volume 1 of *Univariate Series*. Academic Press.

[Quental and Gonçalves 2010] Quental, N. C. and Gonçalves, P. A. S. (2010). CDS-BitTorrent: Um Sistema de Disseminação de Conteúdo para a Melhoria do Desempenho de Aplicações BitTorrent sobre MANETS. Em *Anais do VI Workshop de Redes Dinâmicas e Sistemas Peer-to-Peer (WP2P)*, pp. 85–89.

[Stoffer and Shumway 2000] Stoffer, D. and Shumway, R. (2000). *Time Series Analysis and Its Applications: With R Examples*. Springer Texts in Statistics. Springer.

[TorrentFreak 2009] TorrentFreak (2009). μTorrent Still on Top, Bitcomet's Market Share Plummets. Website TorrentFreak disponível em, *http://torrentfreak.com/utorrent-still-on-top-bitcomets-market-share-plummets-090814/* (Último acesso em 24/07/2012).

[TorrentFreak 2011] TorrentFreak (2011). μTorrent and BitTorrent Hit 100 Million Monthly Users. Website TorrentFreak disponível em, *http://torrentfreak.com/utorrent-bittorrent-hit-100-million-monthly-users-110103/* (Último acesso em 27/07/2012).

[Wang et al. 2010] Wang, H., Liu, J., and Xu, K. (2010). Exploring BitTorrent Peer Distribution via Hybrid PlanetLab-Internet Measurement. Em *Proceedings of the 18th IEEE International Workshop on Quality of Service - IWQoS*, pp. 1–2.

[Wojciechowski et al. 2010] Wojciechowski, M., Capotâ, M., Pouwelse, J., and Iosup, A. (2010). BTWorld: Towards Observing the Global BitTorrent File-Sharing Network. Em *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, pp. 581–588.

[Wu et al. 2010] Wu, D., Dhungel, P., Hei, X., Zhang, C., and Ross, K. (2010). Understanding Peer Exchange in Bittorrent Systems. Em *Proceedings of Tenth IEEE International Conference on Peer-to-Peer Computing (P2P)*, pp. 1–8.

[Xia and Muppala 2010] Xia, R. L. and Muppala, J. (2010). A Survey of BitTorrent Performance. *Communications Surveys & Tutorials*, 12:140–158.

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

[Zhang et al. 2010b] Zhang, C., Dhungel, P., Wu, D., and Ross, K. (2010b). Unraveling the BitTorrent Ecosystem. *IEEE Transactions on Parallel and Distributed Systems*, 22(99):1–1.

[Zhang et al. 2011] Zhang, J., Xing, W., and Lu, D. (2011). Tracker Algorithm Based on Upload Capacity in BitTorrent Network. Em *International Conference on Computer Science and Service System (CSSS)*, pp. 3792 –3795.