

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

Thiago Amaral Guarnieri

Impacto de Redes Sociais na Difusão de Conteúdo em Swarms Bittorrent

Juiz de Fora

2014

Thiago Amaral Guarnieri

Impacto de Redes Sociais na Difusão de Conteúdo em Swarms Bittorrent

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Juiz de Fora, na área de concentração em Redes de Computadores, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Alex Borges Vieira

Coorientador: Ana Paula Couto da Silva

Juiz de Fora

2014

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Amaral Guarnieri, Thiago.

Impacto de Redes Sociais na Difusão de Conteúdo em Swarms Bittorrent / Thiago Amaral Guarnieri. – 2014.

72 f. : il.

Orientador: Alex Borges Vieira

Coorientador: Ana Paula Couto da Silva

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação, 2014.

1. Redes Sociais. 2. Bittorrent. 3. Freeriding. I. Borges Vieira, Alex, orient. II. Couto da Silva, Ana Paula, coorient. III. Título.

Thiago Amaral Guarnieri

Impacto de Redes Sociais na Difusão de Conteúdo em Swarms Bittorrent

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Juiz de Fora, na área de concentração em Redes de Computadores, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Aprovada em: 22 de agosto de 2014.

BANCA EXAMINADORA

Prof. D.Sc. Alex Borges Vieira - Orientador
Universidade Federal de Juiz de Fora

Profa. D.Sc. Ana Paula Couto da Silva - Coorientador
Universidade Federal de Minas Gerais

Profa. D.Sc. Jussara M. Almeida
Universidade Federal de Minas Gerais

Prof. D.Sc.. Victor S. A. Menezes
Universidade Federal de Juiz de Fora

Ao Senhor Jesus em primeiro lugar. À minha namorada, família e amigos pelo suporte e incentivo.

AGRADECIMENTOS

A conclusão de um mestrado, em um país onde a educação de qualidade é um privilégio restrito, sem dúvidas é um motivo de orgulho e felicidade para mim. Entretanto esta é uma vitória coletiva. Não a teria alcançado caso não tivesse me cercado de pessoas que, para mim, serviram de apoiadores e exemplos a serem seguidos.

Em primeiro lugar gostaria de agradecer muito ao Senhor Deus pela sua misericórdia e livramento durante esses dois anos cheios de percalços. Sem Ele, já teria de fato sucumbido há muito tempo.

Agradeço aos meus familiares, particularmente meus pais, Jorge e Leda, minha namorada Tânia, meu irmão Wilder e minha cunhada Raquel, que, sem dúvida, também é como uma irmã. Acima de tudo vocês foram e são exemplos de competência a serem seguidos por quem deseja alcançar vitórias ao longo desta vida.

Aos meus professores do Departamento de Ciência da Computação pelo suporte, compreensão e amizade. Em especial, aos meus orientadores, Alex e Ana, que ao longo desses dois anos foram capazes de me moldar em uma nova pessoa. Os conhecimentos passados por eles é, sem dúvida, inestimável.

Aos meus amigos do grupo de redes que compartilharam comigo todas as dificuldades e vitórias alcançadas.

"Podemos facilmente perdoar uma criança que tem medo do escuro;
a real tragédia da vida é quando os homens têm medo da luz."

Platão

RESUMO

As aplicações peer-to-peer (P2P) estão presentes na *Internet* há mais de uma década. Seu sucesso, reconhecido em aplicações de compartilhamento de arquivo, tem se expandido para redes sociais, transmissão de vídeo, entre outras. Entretanto, o custo do tráfego das aplicações P2P é maior que o de sistemas cliente/servidor, pelo fato de os *peers* estarem dispersos geograficamente. Assim, muitos provedores reduzem a prioridade de pacotes P2P em seus roteadores, diminuindo o desempenho dessas aplicações. Observa-se ainda, a ocorrência de *freeriders*, que são *peers* que não destinam satisfatoriamente recursos para as redes de disseminação em que participam (*swarm*).

Muitas abordagens foram propostas para reduzir tráfego entre sistemas autônomos e *freeriding*, especialmente através do estabelecimento de relações de maior duração entre *peers*. Nesse sentido, pesquisas demonstram que o contexto social pode auxiliar no aumento de parcerias de longa duração e de regionalização de tráfego.

Com base nesse contexto, este trabalho avalia a influência de uma rede social, o *Facebook*, na disseminação de dados em redes *Bittorrent*, tanto em suas características estáticas quanto na evolução de sua topologia ao longo do tempo. Para isso foram coletados mais de 16.600 *swarms*, divididos em tradicionais - disseminados via sites de indexação - e sociais, que são disseminados via *Facebook*.

A análise estática tem como foco os seguintes fatores: (I) características gerais dos *swarms* (II) melhorias em disseminação de dados (i.e. saúde de *swarms*), (III) a localidade dos *peers* da rede P2P. Além disso é apresentado um estudo da evolução temporal de *swarms* segundo a duração das relações entre *peers* e métricas de centralidade.

Os resultados demonstram que a disseminação social provoca uma sensível redução de *freeriding* e maior localização de *peers*, indicando novos caminhos que apontam para o equilíbrio entre vazão de dados e custo de transmissão.

Palavras-chave: Redes Sociais. Bittorrent. Freeriding.

ABSTRACT

Peer-to-peer applications are very important on Internet since last decade. Its success as a file-sharing application is unquestionable and has been repeated on applications such cloud computing, social network and video broadcasting.

However, it is notorious that the traffic generated by peer-to-peer is not AS friendly: peers on a P2P system are usually geographically sparse, making more expensive the data traffic. Because of it, some content providers perform traffic shaping on their network, reducing P2P packet priority and, by consequence, its performance. Yet, other important problem on P2P systems is the freeriding behavior, where a peer doesn't contribute to the system with its resources.

Today, there are many proposed approaches to reduce inter-AS traffic and freeriding by finding a way to provide long term relationship between peers. With the present success of social networks, some proposals intend to investigate if social contexts can alleviate the impact of freeriding and inter-AS traffic.

Through measurements performed in a Bittorrent real-world swarm data-set, we present a characterization that intends to investigate the influences of swarm social dissemination on Facebook in peer relationship and also performance metrics, location, and freeriding reduction.

Results achieved by us shows that using social dissemination provides a notable improvement on solving the freeriding problem and peer geographic dispersion. So, a protocol that uses a geographic or social aware peer selection can have better performance on social disseminated swarms.

Key-words: Social Networks. Bittorrent. Freeriding.

LISTA DE ILUSTRAÇÕES

Figura 1 – Rede sobreposta e camada de rede	16
Figura 2 – Estrutura de uma rede <i>Bittorrent</i>	20
Figura 3 – Exemplo de mensagem PEX	24
Figura 4 – Representação dinâmica da topologia de rede para $t = 1, 2$ e 3	34
Figura 5 – Exemplo de identificação de comunidades em um <i>snapshot</i>	35
Figura 6 – Melhoria da disseminação na presença de <i>peers multi swarm</i>	39
Figura 7 – Representação gráfica da similaridade espacial e temporal	43
Figura 8 – Distribuição do tamanho dos <i>swarms</i> de <i>torrents</i>	44
Figura 9 – Distribuição da quantidade de arquivos compartilhados.	44
Figura 10 – Top 20 extensões compartilhadas.	46
Figura 11 – Quantidade de <i>peers</i> obtidos via PEX e DHT por <i>swarm</i>	47
Figura 12 – Distribuição do número de <i>seeders</i> em um <i>swarm</i>	47
Figura 13 – Distribuição da proporção <i>seeders/peers</i> em um <i>swarm</i>	48
Figura 14 – Distribuição de <i>peers</i> que participam de múltiplos <i>swarms</i>	49
Figura 15 – Distribuição do coeficiente de similaridade de <i>Jaccard</i>	49
Figura 16 – Matriz de similaridade.	50
Figura 17 – Número de países distintos em um <i>swarm</i>	50
Figura 18 – Top 15 países mais frequentemente encontrados nos <i>swarms</i>	51
Figura 19 – Número de ASes por <i>swarm</i>	52
Figura 20 – Localização dos <i>peers</i> de um <i>swarm torrent</i>	52
Figura 21 – Tamanho de comunidade	54
Figura 22 – Número de comunidades por <i>swarm</i>	54
Figura 23 – Coeficiente de <i>Jaccard</i> para comunidades.	55
Figura 24 – Coeficiente de <i>Jaccard</i> médio durante período de coleta.	55
Figura 25 – Tempo de permanência agregada individual de <i>peers</i>	56
Figura 26 – Taxa de duração de comunidades.	56
Figura 27 – Grau médio de <i>peers</i> em <i>swarms</i>	57
Figura 28 – Exemplo da organização de nós em comunidades tradicionais e sociais.	58
Figura 29 – <i>Betweenness</i> em comunidades.	59
Figura 30 – Transitividade média em comunidades.	59
Figura 31 – <i>Closeness</i> em comunidades.	60

LISTA DE TABELAS

Tabela 1 – Estrutura das listas de uma mensagem PEX	24
Tabela 2 – Descrição da coleta para análise de localidade e saúde	32
Tabela 3 – Tamanho das amostras social e tradicional	33
Tabela 4 – Aspectos gerais de saúde dos <i>swarms</i>	46
Tabela 5 – Regionalização de relações entre <i>peers</i>	53

SUMÁRIO

1	INTRODUÇÃO	13
1.1	CONTRIBUIÇÕES DO TRABALHO	14
1.2	ORGANIZAÇÃO DA DISSERTAÇÃO	15
2	CONCEITOS TEÓRICOS	16
2.1	ARQUITETURA P2P	16
2.1.1	Definição	16
2.1.2	classificação	17
2.1.2.1	<i>Estruturadas</i>	17
2.1.2.2	<i>Não-Estruturadas</i>	18
2.2	PROTOCOLO <i>BITTORRENT</i>	19
2.2.1	Definição	19
2.2.2	Funcionamento	20
2.2.3	<i>Tit-for-tat</i>	21
2.2.4	<i>Freeriding</i>	21
2.3	EXTENSÕES <i>BITTORRENT</i>	23
2.3.1	<i>Peer Exchange (PEX)</i>	23
2.3.2	Estrutura das mensagens PEX	23
2.3.3	<i>Kademlia</i>	24
2.4	REDES <i>TORRENT</i>	25
2.4.1	Redes públicas	25
2.4.2	Redes privadas	25
3	TRABALHOS RELACIONADOS	27
3.1	LOCALIZAÇÃO GEOGRÁFICA DE <i>PEERS</i>	27
3.2	<i>BITTORRENT</i> COMO APLICAÇÃO SOCIAL	28
4	METODOLOGIA E COLETA DE DADOS	30
4.1	ANÁLISE ESTÁTICA	30
4.2	ANÁLISE DINÂMICA	32
4.2.1	Identificação de Comunidades	35
5	MÉTRICAS DE INTERESSE	37
5.1	ANÁLISE ESTÁTICA	37
5.1.1	Tamanho de <i>swarm</i>	37
5.1.2	Quantidade de arquivos	37
5.1.3	Tipos de extensão	37

5.1.4	Número de <i>peers extra tracker</i>	38
5.1.5	Número <i>swarms multitracker</i>	38
5.1.6	Número de <i>swarms</i> mortos	38
5.1.7	Número de <i>seeders</i>	38
5.1.8	Proporção de <i>seeders</i> por <i>peers</i>	38
5.1.9	Número de <i>peers multi swarm</i>	38
5.1.10	Número de <i>peers</i> por AS e País	39
5.2	ANÁLISE DINÂMICA	39
5.2.1	Métricas de topologia	39
5.2.1.1	<i>Tamanho de comunidades</i>	39
5.2.1.2	<i>Número de comunidades</i>	40
5.2.2	Métricas de <i>peer</i>	40
5.2.2.1	<i>Grau</i>	40
5.2.2.2	<i>Betweenness</i>	40
5.2.2.3	<i>Closeness</i>	41
5.2.2.4	<i>Transitividade</i>	41
5.2.2.5	<i>Taxa de permanência</i>	41
5.2.2.6	<i>Número de reencontros entre peers</i>	42
5.2.3	Coefficiente de similaridade de <i>Jaccard</i>	42
5.2.3.1	<i>Duração das comunidades</i>	42
6	AVALIAÇÃO DO IMPACTO DE DISSEMINAÇÃO SOCIAL	44
6.1	ANÁLISE ESTÁTICA	44
6.1.1	Propriedades gerais	44
6.1.2	Análise de saúde	46
6.1.3	Análise de localidade	50
6.2	ANÁLISE DINÂMICA	52
6.2.1	Caracterização de comunidades	53
6.2.2	Centralidade de <i>peers</i>	57
7	CONCLUSÃO E TRABALHOS FUTUROS	61
	REFERÊNCIAS	63
	APÊNDICE A – INTERVALOS DE CONFIANÇA	67
A.1	<i>Peers</i> por <i>Swarm</i>	67
A.2	Arquivos por <i>swarm</i>	67
A.3	Número de <i>seeders</i>	67
A.4	Número de <i>seeders</i> por <i>peer</i>	68

A.5	<i>Multi swarm peers</i>	68
A.6	<i>Peers</i> DHT e PEX	68
A.7	Coefficiente de <i>Jaccard</i>	69
A.8	Países por <i>swarm</i>	69
A.9	<i>Peers</i> por país	69
A.10	<i>Peer</i> por AS	69
A.11	Coefficiente de <i>Jaccard</i> (topologia)	70
A.12	Coefficiente de <i>Jaccard</i> médio por snapshot	70
A.13	Tamanho das comunidades	70
A.14	Quantidade de comunidades	70
A.15	Permanência de <i>peers</i>	71
A.16	Grau médio	71
A.17	<i>Betweenness</i>	71
A.18	<i>Closeness</i>	72
A.19	Transitividade	72

1 INTRODUÇÃO

Desde a década passada, observa-se a massiva adoção do paradigma *peer-to-peer* (P2P) na *Internet*. De fato, o tráfego gerado por suas aplicações é de grande representatividade nas redes de computadores. Estimativas mostram que 36% de todo tráfego de *upload* da América do Norte tem como origem o *Bittorrent*, atualmente a principal aplicação para compartilhamento de arquivos em redes P2P. Além disso o *Bittorrent* apresenta picos de tráfego agregado nesta região que ultrapassam 10%¹.

O sucesso de sistemas P2P depende fortemente do nível de colaboração de seus participantes (*peers*). A falta de cooperação impõe um grande desafio à garantia de qualidade de serviço e por isso existem vários mecanismos que procuram incentivar a cooperação nesses sistemas. O *Bittorrent* implementa um esquema de reciprocidade descentralizado conhecido como *tit-for-tat* [10]. Por esse esquema, *peers* monitoram-se mutuamente e assim, recursos são oferecidos a um *peer* de acordo com sua contribuição.

Apesar da larga adoção e uso do *tit-for-tat*, observa-se que, em muitos casos, os *swarms*, que são os *peers* que compartilham o mesmo arquivo, não se beneficiam dele [22]: o histórico de contribuição de um *peer* não é passado de um *swarm* para outro. Desta forma os *seeders*, que são *peers* que possuem cópias completas dos arquivos disseminados, não têm incentivos para permanecer em um *swarm* contribuindo para o aumento da disponibilidade de conteúdo. Mesmo assim, [28] demonstra que o bom desempenho de redes *Bittorrent* se deve muito mais ao altruísmo de uma pequena parcela de *peers* do que ao *tit-for-tat*. Adicionalmente, [45] evidencia que mecanismos auxiliares como *share-ratio*² também são eficientes para prolongar a vida de um *swarm*.

O caráter aleatório de escolhas de parcerias baseado em reciprocidade do *tit-for-tat* também não favorece relações de longa duração entre *peers*. [41] mostram que a taxa de reencontros entre eles é menor que 5% e sugerem que um aumento desse fator pode ser benéfico para o *swarm*: *peers* altruístas poderiam contribuir com uma parcela maior de parceiros e por um tempo maior, levando a um aumento de desempenho de disseminação e maior disponibilidade de conteúdo.

Geralmente, uma rede sobreposta formada pelos *peers* engloba uma grande quantidade de sistemas autônomos (*autonomous system* - AS). Um sistema autônomo é definido como uma rede de roteadores, administrada por um ou mais provedores de conteúdo, com protocolo de roteamento único e bem definido³. O custo da transmissão de pacotes entre dois ASes é proporcional à sua distância e ao número de sistemas autônomos intermediários. Desta forma, ao possuir alta dispersão geográfica de *peers*, o protocolo *Bittorrent* tem

¹ Global Internet Phenomena Report: 2H 2012, www.sandvine.com

² Fração entre quantidade agregada de *upload* e *download* de um *peer*

³ RFC 1930, seção 3: <http://tools.ietf.org/html/rfc1930>

um alto custo agregado, impondo aos provedores de conteúdo a necessidade de reduzir a prioridade de pacotes P2P em seus roteadores, o que leva à uma perda da eficiência das soluções baseadas em *Bittorrent*.

A localidade de um *peer*, mesmo em um único *swarm*, também pode ser importante para redução global do tráfego entre sistemas autônomos. Por exemplo, [39] mostram que 85% dos *peers* participam de múltiplos *swarms*. Essa característica pode ser usada para reunir *peers* geograficamente próximos em comunidades que compartilhem o mesmo conjunto de arquivos e também incentivar a permanência de um *seeder* colaborando por mais tempo, premiando-o com maior prioridade na obtenção de conteúdo em outros *swarms* da comunidade.

Atualmente o *Bittorrent* tem sua aplicabilidade expandida para serviços além do compartilhamento de arquivos: tem-se redes sociais, mecanismos de *chat*, computação em nuvem e muitos outros exemplos que corroboram o fato de uma tendência no aumento da descentralização geográfica do tráfego na *Internet*. Por outro lado, a baixa cooperação observada entre *peers* reduz sensivelmente disponibilidade do conteúdo de um *swarm* e seu tempo de vida. Por esses motivos, estudar formas de melhorar a localidade e as relações de longa duração entre *peers* é de extrema importância pois, mesmo que isso não signifique melhoras de desempenho para usuários finais, auxilia na redução de custo e aumento da aplicabilidade de mecanismos baseados em P2P.

Nesse sentido observa-se, através de estudos recentes, a tendência de agregar contextos sociais à disseminação de arquivos em redes *Bittorrent*. Contextos sociais descrevem um conjunto de nós segundo suas similaridades, o que poderia interferir na organização de *peers* em um *swarm*, fazendo com que eles se agrupem segundo atributos semelhantes, como mesma região geográfica, língua, interesses, entre outros.

A escolha orientada a contextos sociais de *peers* implica diretamente em mudanças no protocolo *Bittorrent*, e por isso, sua eficiência depende do nível de adoção por parte dos usuários. Por outro lado a pura divulgação de *swarms* em redes sociais já existentes, embora não permita correlação direta entre os *peers* e os nós sociais, constitui-se numa oportunidade de agrupamento social sem modificações no protocolo existente, o que facilita sua adoção. Baseado nessa premissa, este trabalho avalia a influência de uma rede social, o *Facebook*, na disseminação *swarms* e seu impacto na localidade e na duração de relações entre *peers* de um *swarm Bittorrent*.

1.1 CONTRIBUIÇÕES DO TRABALHO

As principais contribuições dessa dissertação são:

- Estudo das relações de longa duração entre *peers* no contexto da disseminação social de *swarms*;

- Análise do impacto da disseminação social na incidência de *freeriding*;
- Análise do agrupamento geográfico e *inter-AS* de *peers* em *swarms Bittorrent*;
- Comparação qualitativa e quantitativa do conteúdo disseminado em *swarms* sociais e tradicionais.
- Demonstrar as melhorias proporcionadas por uma nova abordagem de disseminação, que já conta com adoção crescente no ambiente real de redes *Bittorrent* e que não necessita de modificações em seu protocolo.

1.2 ORGANIZAÇÃO DA DISSERTAÇÃO

O presente trabalho está organizado da seguinte forma: no capítulo 2 estão descritos os conceitos teóricos a respeito de sistemas P2P e *Bittorrent* concernentes ao desenvolvimento deste trabalho. No capítulo 3 são apresentados os trabalhos relacionados às pesquisas na área de localização e aplicação de contexto social na disseminação de *swarms* P2P. No capítulo 4 são descritos os conjuntos de dados, a metodologia de coleta e análise e o processo utilizado para investigar a formação de comunidades em *swarms* por influência da disseminação social. No capítulo 5 são descritas as métricas utilizadas para a análise do conjunto de dados do trabalho. No capítulo 6 é apresentada a análise estática dos *swarms* em termos de natureza, saúde e localidade. No capítulo 7 tem-se a análise dinâmica dos *swarms*. Nela é explorada a relação entre *peers* em termos temporais e quantitativos, bem como as influências da disseminação social nas métricas topológicas de centralidade. As conclusões e trabalhos futuros são apresentados no capítulo 8. Adicionalmente, no apêndice, apresenta-se o cálculo dos intervalos de confiança das métricas do trabalho como suporte às conclusões efetuadas.

2 CONCEITOS TEÓRICOS

Este capítulo tem como objetivo descrever os conceitos relacionados ao paradigma P2P. A seção 2.1 descreve os conceitos genéricos de redes *peer-to-peer* e a seção 2.2 apresenta uma descrição detalhada do protocolo *Bittorrent*, bem como suas principais extensões.

2.1 ARQUITETURA P2P

2.1.1 Definição

Define-se por redes P2P aquelas em que cada nó desempenha o papel de receptor e disseminador de informações. Em outras palavras, um nó é capaz de agir como cliente, recebendo dados de seus coparticipantes, e servidor, disponibilizando dados para outros nós [33]. Uma rede P2P é um conjunto auto-organizável de nós que formam entre si circuitos lógicos controlados pela camada de aplicação. Essa rede é conhecida como rede sobreposta e tem topologia independente da rede física (figura 1).

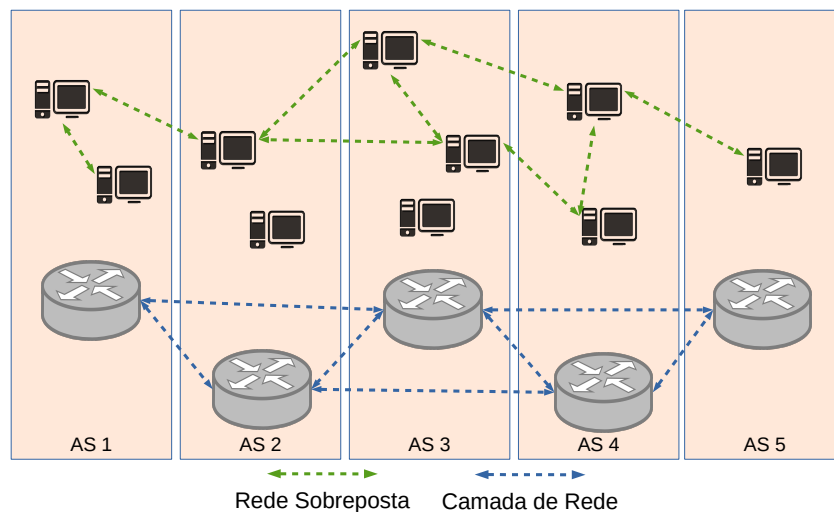


Figura 1 – Rede sobreposta e camada de rede

A abordagem P2P é diferente da arquitetura cliente-servidor, onde tem-se bem definidos e diferenciados os papéis de cada entidade do sistema. Na abordagem cliente-servidor observa-se uma entidade central, denominada servidor, que é responsável por armazenar e distribuir conteúdo para os clientes requisitantes. O papel da entidade servidor pode ser desempenhado por um conjunto interligado de computadores, sendo cada um responsável por determinado serviço disponibilizado pelo provedor de conteúdo.

A grande vantagem de sistemas cliente-servidor está na maior facilidade de garantir qualidade de serviço (*quality of service* - QoS) para seus usuários, pois ao possuírem um

ponto central de coordenação, torna-se mais fácil gerenciar suas funções. Para obtenção de QoS um servidor também deve ser escalável, ou seja, deve garantir que as requisições concomitantes sejam respondidas satisfatoriamente, independente de sua quantidade.

Em redes P2P, ao contrário, cada participante pode assumir o papel de servidor caso algum nó da rede falhe, bastando que haja um caminho entre ele e o requisitante. Desta forma a rede se torna mais resiliente. Além disso sistemas P2P tendem a ser mais escaláveis, visto que a banda agregada aumenta à medida em que novos nós ingressam na rede, ou seja, a qualidade de serviço é diretamente proporcional ao nível de colaboração entre os coparticipantes.

Embora a arquitetura P2P tenha surgido há décadas, ela atingiu notável popularidade no fim dos anos 90 com o surgimento do *Napster*¹. Esta rede possuía um coordenador central, que registrava a associação entre um arquivo e sua máquina de origem. Desta forma, quando um *peer* efetuava uma busca por conteúdo, esta era direcionada ao coordenador, que retornava os resultados ao requisitante. A existência de uma entidade central de coordenação, embora facilite a administração, faz com que a rede possua um ponto único de falha, o que reduz sua resiliência. No caso particular do *Napster*, que disseminava conteúdo protegido por direitos autorais, a existência de coordenação central foi um fator preponderante que culminou na sua desativação, fruto de processos movidos por infração de propriedade intelectual.

A partir desta aplicação, outras mais robustas surgiram com objetivo de eliminar a necessidade de coordenação central para efetuar buscas. No *Gnutella* [31], essa pesquisa era efetuada por *query flooding*, onde um *peer* inundava a rede até uma certa profundidade com o objetivo de obter parceiros com o conteúdo desejado. Esta abordagem é pouco escalável, sendo por isso pouco usada.

2.1.2 classificação

Existem várias formas de se classificar redes P2P, sendo que uma delas é através de seu nível de organização, ou, em outras palavras, a forma pela qual seus nós se conectam. Nesse contexto uma rede P2P pode ser classificada como estruturada ou não-estruturada.

2.1.2.1 Estruturadas

Em redes estruturadas a rede sobreposta é construída de forma determinista, ou seja, um nó constrói uma relação com outro nó baseado em critérios formais ou regras específicas. Um dos processos mais utilizados é o das tabelas hash distribuídas (*Distributed Hash Tables* – DHT) [7], no qual para uma chave k , um nó X é dono da chave ou tem uma ligação para um nó perto de k . Nesse mecanismo, a responsabilidade por manter a tabela

¹ www.napster.com

de relacionamento chave/valor é compartilhada entre os nós de forma que uma mudança topológica cause o mínimo de interferência.

O grande desafio nesse tipo de arquitetura é implementar um esquema determinístico capaz de evitar colisões entre chaves, além de lidar de modo eficiente com a atualização das tabelas quando há sucessivas saídas e entradas de nós do sistema. Exemplos de aplicações P2P estruturadas são o *Chord*[35], o *NodeWiz* [4]. Recentemente o *Bittorrent* tem implementado uma variação do mecanismo de DHT conhecido como *Kademlia*² para reduzir a dependência de coordenação central. Esse mecanismo é detalhado na subseção 2.3.3.

2.1.2.2 Não-Estruturadas

As redes P2P não-estruturadas não seguem critérios específicos para a construção da rede sobreposta. O estabelecimento de parcerias se dá de maneira aleatória. Desta forma, um nó que deseje um conteúdo específico deve inundar a rede com uma busca [32] para encontrar o maior número de nós com o requerido recurso. Embora simples, essa abordagem é pouco escalável [3], além de possuir profundidade de busca limitada pelo fato de um pacote não poder trafegar indefinidamente na rede. Em razão disso, surgiram outros mecanismos de roteamento como o de caminhada aleatória [17], onde a escolha do nó para o próximo salto é aleatória e condicionada a uma distribuição de probabilidade, e de índices de roteamento [37], onde os arquivos são classificados em categorias e a busca é direcionada aos nós que possuem mais arquivos cuja categoria é a mesma do conteúdo buscado.

Os sistemas P2P não-estruturados são adequados a sistemas onde há uma grande entrada e saída de nós, ou seja, quando há alta dinamicidade topológica. De acordo com [33], as redes não-estruturadas podem ser classificadas como puras, centralizadas ou híbridas.

- Pura ou descentralizada: Todos os nós estão num mesmo nível de hierarquia. Não existe ponto central de coordenação, ou seja, cada nó é responsável por manter sua lista de recursos. Por essa razão a busca deve ser efetuada por *flooding*, ou seja, a requisição é repassada para todos os nós da rede. Geralmente limita-se o número de saltos com objetivo de evitar sobrecarga. Exemplos de aplicação desse conceito são o *Gnutella*, *FreeNet*³ e *Kazaa*⁴.
- Centralizadas: Nesse modelo existe um dispositivo responsável por gerenciar e regular a entrada de nós na rede. É também atribuído a esse dispositivo a responsabilidade de

² xlattice.sourceforge.net/components/protocol/kademlia/specs.html

³ freenetproject.org

⁴ kazaa.com

mapear os recursos da rede e encaminhamento de buscas. De mais fácil gerenciamento, tem como desvantagem um ponto de falha único. Como exemplos dessa abordagem temos o *Napster* e o *Audiogalaxy*.

- **Híbrida:** Possui características de redes centralizadas e descentralizadas. Neste modelo existe um conjunto de nós com atribuições especiais como, por exemplo, encaminhar buscas e manter um registro parcial da rede. Estes nós formam entre si uma rede sobreposta particular que provê os recursos necessários para a interligação entre os nós dito “comuns”. Podemos destacar a rede *eMule*⁵, *Skype*⁶ e *Bittorrent* como aplicações deste paradigma.

2.2 PROTOCOLO BITTORRENT

O protocolo *Bittorrent* é um dos mais bem-sucedidos exemplos de aplicação do paradigma *peer-to-peer* na atualidade, sendo utilizado não somente em compartilhamento de arquivos, mas também em gerenciamento de servidores [41] e distribuição de vídeo⁷. Esta seção tem o objetivo de detalhar o funcionamento desse protocolo.

2.2.1 Definição

As redes de nós no protocolo *Bittorrent* são chamadas de *swarms*. Cada integrante de um *swarm* é denominado *peer* e, todos eles, compartilham entre si o mesmo arquivo. Um *peer* pode participar de múltiplos *swarms*, possuindo em cada um deles uma identificação particular. Em outras palavras, um *peer* não possui um identificador universal que permita a ele ser detectado univocamente em diferentes *swarms*.

Os *peers* em um *swarm* são classificados como *leechers*, que são *peers* que não possuem o arquivo completo disseminado no *swarm*, e os *seeders*, que são os *peers* que já possuem o arquivo completo, mas que permanecem no *swarm* com o objetivo de aumentar a disponibilidade de cópias completas para os *leechers*. Em algumas comunidades, conhecidas como *darknets* [45], o nível de colaboração de um *seeder* é tomado como critério para que o *peer* obtenha novos arquivos.

Um *swarm* geralmente possui uma unidade coordenadora central conhecida como *tracker*. O *tracker* é responsável por gerenciar os *participantes* dos *swarms* armazenados em sua base de dados. Seus serviços incluem registrar a entrada de novos *peers*, retirar aqueles que não participam mais da rede e responder a requisições em busca de novos parceiros efetuadas por *peers*. Em alguns casos o *tracker* também armazena o histórico de cooperação de um *peer*, que serve como critério para quantificar sua contribuição.

⁵ emule-project.net

⁶ skype.com

⁷ labs.bittorrent.com/experiments/bittorrent-live.html

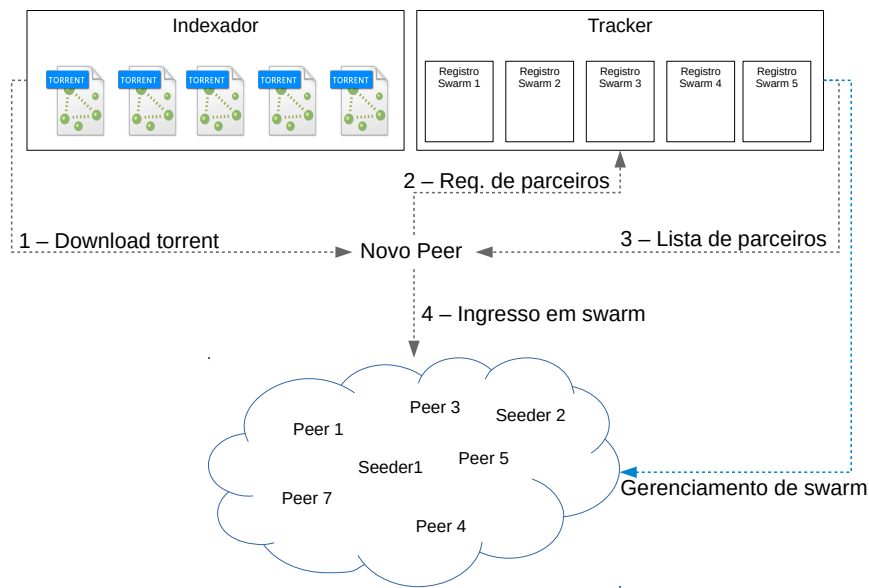


Figura 2 – Estrutura de uma rede *Bittorrent*

2.2.2 Funcionamento

O primeiro passo para um usuário ingressar em um *swarm* é obter informações sobre sua estrutura, que incluem a lista de arquivos disseminados, a lista de trackers, data de criação, entre outras. Para isso ele efetua o *download* de um arquivo de metainformações conhecido como *torrent*. Usualmente os arquivos *torrent* estão catalogados em sites de indexação, responsáveis por categorizá-los de acordo com a natureza de seu conteúdo.

Assim que um nó se conecta, ele informa ao *tracker* sua presença no *swarm* e requisita uma lista aleatória de *peers*, de tamanho padrão de 50 *peers*[18]. A partir desse instante a troca de blocos pode ser iniciada. Esta requisição é repetida usualmente a cada 15 minutos.

Nos casos em que o *swarm* não depende diretamente de um *tracker*, o usuário deve ingressar através de *magnet links*. Nos *magnet links*, os arquivos são encontrados através de seu conteúdo, em detrimento da sua localização. Para isso, uma função *hash* (SHA1) é aplicada ao conteúdo gerando um identificador único para cada arquivo disseminado no ambiente *Bittorrent*. Este identificador é posteriormente utilizado para buscar *peers* através de tabelas *hash* distribuídas (*Distributed Hash Tables – DHT*). Os nós com os quais um *peer* está conectado constituem sua vizinhança local e a troca de mensagens de controle é sucinta: cada nó reporta os blocos (denominados *chunks*) que possui e o interesse naqueles que ainda não tem. O conjunto de nós com os quais um *peer* está enviando dados é denominado seu conjunto ativo.

2.2.3 *Tit-for-tat*

O *tit-for-tat* é o mecanismo utilizado pelo *Bittorrent* para a escolha de nós do conjunto ativo de um *peer*. Essa abordagem é baseada em reciprocidade local, ou seja, um nó *A* só contribui com um nó *B* se tiver recebido contribuição de *B*. Formalmente o mecanismo de escolha do conjunto ativo é conhecido como *unchoking* [10]. Por padrão 4 nós com as maiores contribuições são escolhidos a cada rodada. Este processo se repete a cada 20 segundos, sendo 10 para escolha e 10 para estabilização de taxa de transferência. Ao final desse período o processo de envio de pacotes é iniciado. Para os nós que acabaram de entrar no sistema e ainda não obtiveram status de elegíveis pelo mecanismo de *unchoking* (ainda não tiveram oportunidade de contribuir), o *peer* transmissor efetua, a cada 30 segundos, o *unchoke* sem considerar contribuição. Esse processo é denominado *optimistic unchoking* e serve para que novos *peers* recebam seus primeiros pacotes bem como para descoberta de *peers* com melhores taxas de contribuição. *Peers* que não contribuem por um período maior que 1 minuto, são excluídos da vizinhança local.

Quando um *peer* se torna um *seeder*, ele não precisa mais receber dados, pois já tem a cópia completa do arquivo. Desta forma, ele deixa de utilizar o *tit-for-tat* e passa a contribuir com todos os *peers* usando o a abordagem *round robin* [24] para intercalação.

2.2.4 *Freeriding*

Em uma rede P2P é esperado que *peers* contribuam com o sistema compartilhando seus recursos enquanto usufruem dos recursos disponibilizados pela rede e por outros *peers*. Entretanto, em muitas redes P2P, uma porção considerável de *peers* relutam em adotar esta prática. Desta forma, o resultado esperado, que é de que o aumento de banda é proporcional ao número de *peers*, pode não ser alcançado, implicando em uma situação conhecida como *freeriding*. No contexto de sistemas P2P um *freerider* é um *peer* que usa os serviços de uma rede P2P mas não contribui com a rede e com os outros *peers* em um nível aceitável.

[1] mostraram que no *Gnutella* 70% dos *peers* não contribuem com nenhum arquivo e que 25% dos *peers* são responsáveis por 99% de todas as buscas com resultados na rede. A partir disso, muitos estudos têm confirmado alto grau de *freeriding* nas redes P2P existentes. Nesses estudos observa-se que, em redes P2P de compartilhamento de arquivo, uma pequena parcela de *peers* altruístas são responsáveis por manter o serviço em funcionamento.

Embora cooperação seja a chave para o bom funcionamento de redes P2P, desenvolver mecanismos de incentivo à cooperação é um dos maiores desafios no desenvolvimento de sistemas P2P. A partir dessa demanda, muitas pesquisas foram realizadas no sentido de tornar redes P2P sensíveis à contribuição e reduzir o problema do *freeriding*. De acordo

com [20], as técnicas *anti-freeriding* podem ser divididas em 3 grupos: abordagens baseadas em reciprocidade, em monetização e em reputação.

Nas abordagens baseadas em monetização, os usuários são pagos pela sua contribuição no formato de microtransações. Os ganhos de cada *peer* são gerenciados por uma entidade central responsável por distribuir unidades do valor monetário de forma justa e implementar mecanismos de criptografia para evitar que hajam fraudes. Esse tipo de abordagem tem como desvantagem a existência de um único ponto de falha, bem como o requisito de identificadores persistentes para usuários, o que reduz a privacidade. Além disso, o gerenciamento de microtransações e mecanismos de auditoria podem gerar grande *overhead* de comunicação de rede. Como exemplos de propostas baseadas nessa abordagem temos o *PPAY* [16] e o *Karma* [44].

Em abordagens baseadas em reciprocidade, um *peer* monitora o comportamento de outros *peers* e avalia seus níveis de contribuição, que pode ser definida como, por exemplo, o montante de *upload* disponibilizado para o sistema ou *peer* em particular. Geralmente essa avaliação persiste somente por um intervalo de tempo curto, permitindo que um *freerider* possa melhorar sua classificação caso seu comportamento se modifique e evitando que seja necessário armazenar históricos de colaboração. Como cada *peer* é responsável por prover essas informações, um *peer* malicioso poderia modificar o cliente para enviar informações de contribuição falsas. A reciprocidade pode ser dividida em duas classes: mútua e indireta. Na reciprocidade mútua, um *peer* A decide como vai contribuir com outro *peer* B baseado na contribuição de B para com A. Esta é a abordagem adotada pelo *Bittorrent*. Na reciprocidade indireta a decisão vai depender da contribuição do *peer* para a rede como um todo. Desta forma, os *peers* monitoram a contribuição de outros *peers* da sua vizinhança para definir o nível de recursos a que eles têm acesso. A eficiência dessa abordagem depende da acurácia das informações sobre os *peers*. Como exemplo de aplicação dessa abordagem tem-se o *GNUnet*⁸.

Já em mecanismos baseados em reputação, um *peer* tem sua credibilidade baseada no *feedback* dos *peers* que interagiram com ele. De posse dessa informação o sistema armazena o histórico de um *peer* para avaliar o montante de recursos que será disponibilizado para ele. Ao contrário de reciprocidade, a reputação é avaliada tomando como base um histórico de informações de longa duração, o que implica que não é fácil converter uma reputação ruim em boa reputação. Os níveis de reputação podem ser gerados localmente, ou seja, um *peer* A interage com um *peer* B e a reputação de B é disseminada por A para a rede, ou a rede pode coletar os dados de reputação de um conjunto de *peers* para gerar uma base central global de reputação que será acessada por todos os *peers*. Exemplos dessa abordagem são *XRep* [11] e o *EigenTrust* [19].

⁸ gnunet.org

2.3 EXTENSÕES *BITTORRENT*

Ao longo da existência do protocolo *Bittorrent*, muitas extensões foram desenvolvidas, particularmente com o objetivo de reduzir a dependência de coordenação central (i.e. *trackers*) por parte dos *swarms*. Exemplos de extensões com esse propósito são o PEX (*Peer Exchange Protocol*) e uma variação de DHT (*Distributed Hash Table*) conhecida como *Kademlia* [25]. A menor dependência de *trackers* evita a existência de um ponto central de falha na rede, o que aumenta a resiliência. Além disso, operadores de *trackers* são alvo frequente de processos legais relacionados à pirataria, o que torna mais atraente a busca por essa descentralização.

2.3.1 *Peer Exchange* (PEX)

Uma mensagem PEX contém uma lista de *peers* com os quais um *peer* remetente está conectado, ou seja, sua vizinhança local. Essas mensagens são trocadas com o objetivo de obter maior conhecimento do *swarm* e torná-lo mais resiliente à divisões [21]. De acordo com [13], 93% dos *peers* coletados em suas medidas suportam PEX.

Existem atualmente dois protocolos de troca de *peers*, sendo um desenvolvido pela equipe responsável pelo *utorrent*⁹, chamado *ut_pex*, e outro pela equipe responsável pelo cliente *Azureus/Vuze*¹⁰, chamado *az_pex*. De acordo com [43], estima-se que o suporte ao *ut_pex* alcance 70% dos clientes enquanto que o *az_pex* seja suportado por até 20% deles. No presente trabalho, o protocolo *ut_pex* foi utilizado para obter uma visão parcial da rede, com o objetivo de analisar as métricas topológicas discutidas na seção 5.2.

2.3.2 Estrutura das mensagens PEX

Uma mensagem *ut_pex* é composta de 6 listas contendo informações sobre quais *peers* estão conectados e quais *peers* foram recentemente desconectados. A mensagem pode ser dividida em duas partes principais: uma lista de *peers* chamada *added*, composta dos *peers* conectados e uma lista *dropped*, composta dos *peers* desconectados. Há um par de listas para ipv4 e uma para ipv6. Existe também uma lista de *flags*, ainda não totalmente aprovada, que indica o *status* de cada *peer*.

A informação de cada *peer* é codificada em um vetor de caracteres composto de 6 *bytes* em IP's versão 4, sendo 4 para endereço IP e 2 para porta. Em IP's versão 6, precisa-se de 16 *bytes* para endereço e 2 para porta.

Uma mensagem PEX é dividida em três partes, contendo 1 *byte* indicador de mensagem *Bittorrent*, 1 *byte* indicador do protocolo de extensões *Bittorrent* (*Libtorrent Extension Protocol* – LTEP), seguido da mensagem PEX propriamente dita. Analisando a

⁹ utorrent.com

¹⁰ vuze.com

Lista	Tamanho
added	<i>Peer Ipv4 – 6 bytes/peer (4 endereço + 2 porta)</i>
added.f	<i>flag Ipv6 – 1 byte/peer</i>
added6	<i>Peer Ipv6 – 18 bytes/peer (16 endereço + 2 porta)</i>
Added6.f	<i>flag Ipv4 – 1 byte/peer</i>
dropped	<i>Peer Ipv4 – 6 bytes/peer (4 endereço + 2 porta)</i>
dropped6	<i>Peer Ipv6 – 18 bytes/peer (16 endereço + 2 porta)</i>

Tabela 1 – Estrutura das listas de uma mensagem PEX

mensagem a partir da seção LTEP tem-se o *byte = 1*, que define a mensagem como sendo a do protocolo *ut_pex*. O resto da mensagem refere-se ao vetor de caracteres descritor de *peers*. Na figura 3 tem-se um exemplo de mensagem PEX. Nela informa-se que o *peer* remetente está conectado com *aaaaaa* e *bbbbbb* e se desconectou de *cccccc*.

Bittorrent Header Byte	LTEP Header Byte = 1	Mensagem PEX: 1:d5:added12:aaaaaabbabbb7:added.f2:☉◀16:added60:8:added6.f0:7dropped6:cccccc:8:dropped60:e
------------------------	----------------------	--------------------------------------------------------------------------------------------------------------

Figura 3 – Exemplo de mensagem PEX

Para converter o vetor de caracteres em IP's toma-se os valores decimais na tabela ASCII do caractere “a” para os quatro primeiros caracteres e os valores hexadecimais dos dois últimos. Convertendo o primeiro IP como exemplo tem-se: $a = 97_{10}$ e $a = 61_{16}$ e portanto $IP = 97.97.97.97 : 6161_{16}$. Convertendo a porta para seu valor decimal tem-se que o IP é 97.97.97.97:24929.

2.3.3 *Kademlia*

Kademlia é uma implementação do conceito de tabelas *hash* distribuídas. No contexto do *Bittorrent*, consiste de um conjunto de pares chave-valor, onde cada nó tem um identificador global único conhecido como *node ID*. No *Kademlia*, um *node ID* é escolhido aleatoriamente num espaço único com tamanho de 160 *bits* e servirá como *hash* de um *peer* ou *swarm*. Uma métrica de distância é utilizada para comparar dois códigos *hash* para determinar sua proximidade. Esta métrica de distância é definida como: dado dois nós *A* e *B* a distância entre *A* e *B* é dada por¹¹: $dist(A, B) = |A \oplus B|$.

Os nós devem manter uma tabela de roteamento contendo as informações de contato de um pequeno número de outros nós. A tabela de roteamento se torna mais detalhada a medida em que os ID's se aproximam do ID do próprio nó. Um nó tem maior

¹¹ bittorrent.org/beps/bep_0005.html

conhecimento a respeito de outros nós que estão próximos a ele, mas também armazena alguns contatos distantes no espaço de chaves global.

Quando um nó deseja achar *peers* de um *torrent* qualquer, ele aplica a métrica de distância à sua própria tabela de roteamento, contactando aqueles que são mais próximos à si e requisitando *peers* pertencentes ao *swarm* desejado. Se o nó possuir *peers* que atendam à requisição eles são retornados, senão ele retorna os dados dos nós que têm o *infohash* mais próximo ao do arquivo *torrent*. O nó requisitor iterativamente consulta os nós cujo *infohash* é próximo ao desejado até que não hajam mais nós próximos. Assim que a busca é exaurida, o cliente então efetua o anúncio de si próprio como um *peer* para os nós com os ID's mais próximos do *infohash* do arquivo *torrent*.

No *Kademlia*, os nós comunicam entre si utilizando chamadas RPC, que incluem: *STORE*, que armazena um conjunto de pares chave-valor no nó receptor, *FIND_NODE*, que requisita os k contatos mais próximos de um dado ID e *FIND_VALUE*, que faz com que o receptor retorne o valor referente a uma chave requisitada qualquer. Outra chamada importante é a *GET_PEERS*, que retorna um conjunto de *peers* próximos a um *infohash* *torrent* qualquer, caso o nó receptor os possua. Caso contrário o nó retorna os k nós em sua tabela que possuem *infohash* mais próximo ao que foi requisitado.

2.4 REDES TORRENT

2.4.1 Redes públicas

Uma rede pública *Bittorrent* geralmente é formada por um *site* indexador de *torrents* e um *tracker*. O *site* indexador é responsável por organizar os arquivos *torrent* em um banco de dados de acordo com critérios de classificação específicos como gênero, tamanho, número de *seeders*, entre outros. Redes públicas geralmente não exigem que seus usuários se cadastrem, desta forma qualquer um pode efetuar *download* dos seus arquivos *torrent* ou de seus *magnet links*.

O *tracker* público, quando presente, não armazena nenhum tipo de histórico a respeito dos participantes de um *swarm*, limitando-se ao gerenciamento padrão previsto no *Bittorrent*. Além disso, ao contrário de comunidades privadas, um *swarm* público permite o uso de extensões como *DHT* e *PEX* para obtenção de mais *peers*.

2.4.2 Redes privadas

De acordo com [45], as redes privadas ou *darknets*, possuem estrutura similar à de redes públicas, mas, ao contrário delas, possuem restrições quanto a quem usa seus recursos, tipicamente requerendo que os usuários se cadastrem em seus *sites*. Muitos deles utilizam sistemas de convite para limitar o número de cadastros. De maneira geral, as *darknets* registram a taxa de contribuição de seus usuários, obrigando que cada

usuário atenda a uma razão entre *download* e *upload* mínima para que ele obtenha acesso a novos *torrents*, podendo inclusive ser banido em caso de baixa contribuição. Também [45] elucidam que tal mecanismo de incentivo pode ser eficiente para manter uma maior taxa de compartilhamento em *swarms*.

A principal diferença entre um *torrent* público e um privado é que os *peers* em um *torrent* privado não efetuam busca por parceiros nas redes *PEX* e *DHT*, obtendo nós apenas do *tracker*. Para isso o arquivo *torrent* dispõe de uma *flag* que indica se ele é privado ou não. É importante ressaltar que essa *flag* nem sempre é respeitada pelos clientes *Bittorrent*. Além disso, o gerenciador de usuários precisa armazenar o IP de cada usuário em seu banco de dados, que por sua vez precisam logar no site de sua *darknet* sempre que seu endereço IP muda.

3 TRABALHOS RELACIONADOS

Apesar do sucesso do protocolo *Bittorrent*, muito ainda há a ser explorado. O fenômeno de *freeriding* ainda é proeminente em sistemas P2P, bem como o alto custo associado ao tráfego P2P entre sistemas autônomos (*autonomous system* - AS). O estudo de técnicas que permitam ao *Bittorrent* ser sensível tanto à localidade quanto ao contexto social podem apontar uma nova luz na direção da resolução destes problemas.

3.1 LOCALIZAÇÃO GEOGRÁFICA DE *PEERS*

O problema do tráfego inter-AS é bem conhecido na literatura, originando uma série de estudos com o objetivo de entender e melhorar a aglutinação de *peers* em sistemas autônomos geograficamente próximos. [23] demonstra que, mesmo em sistemas autônomos populares a quantidade de *peers* de um *swarm* qualquer não ultrapassa o montante de 5%. Este trabalho também revela que boa parte dos *peers* participam de vários *swarms*, sugerindo que considerar essa característica poderia melhorar a disponibilidade de conteúdo dentro de um AS em particular. [6], por sua vez demonstram quais seriam as vantagens de se ter alta localidade em *swarms*, dentre as quais citam-se redução em duas ordens de magnitude do tráfego inter-AS, e menor redução de desempenho em *links* Inter-AS congestionados.

De maneira geral as mudanças sugeridas para atingir sensibilidade à localidade recaem em três grupos:

- Modificação do cliente para que o critério de escolha leve em consideração localização geográfica;
- Modificação do *tracker* para que ele retorne *peers* geograficamente próximos;
- Modificação da comunicação entre clientes ou das respostas dos *trackers* pelos AS.

A respeito da primeira abordagem, [30] mede a distância entre *peers* através do RTT (*round-trip time*), desta forma escolhe-se preferencialmente *peer* próximos. Por sua vez [13] exploram um método de escolha de *peers* locais capaz de aumentar a localidade global de *swarms* em 6%, além de economizar 8TWh de energia elétrica por ano na transmissão de pacotes. Já na segunda abordagem [5] exploram um método onde o *peer* informa sua localização para o *tracker* que retorna uma lista de *peers* que atende a uma razão fixa entre *peers* locais e não locais. Por fim [36] propõe que os sistemas autônomos devem monitorar os *swarms* e substituir as listas de *peers* enviadas pelos *trackers* por listas com *peers* próximos ou implementar nós especiais para servirem de *cache* local para seus *peers*.

3.2 BITTORRENT COMO APLICAÇÃO SOCIAL

Há um grande número de trabalhos que foca na análise e melhoria da disseminação de conteúdo no *Bittorrent*. Notadamente, um dos mecanismos mais importantes nessa linha é o *tit-for-tat* [10]. Esse mecanismo incentiva a cooperação mútua nos *swarms torrent* a partir de monitorações distribuídas, realizadas localmente pelos *peers*. De certa forma, a eficiência do *Bittorrent* é associada a esse mecanismo. Porém, há um grande número de *swarms* que não se beneficia dele.

De fato, cerca de 40% dos *swarms* apresentam um desempenho baixo, com um grande número de usuários incapazes de obter o conteúdo compartilhado por completo [22]. Esse baixo desempenho é explicado pela falta de incentivo a um *peer* para que ele permaneça contribuindo com o *swarm* após ele ter conseguido capturar o conteúdo completo. O *tit-for-tat*, que é o principal mecanismo de incentivo no *Bittorrent*, não utiliza informações de longa duração, não explora as interações sociais dos *peers* e não extrapola as avaliações entre *swarms* diferentes. Assim, tão logo um *peer* consegue baixar os arquivos compartilhados, ele costuma abandonar o *swarm*.

Há um número ainda pequeno de trabalhos que extrapolam os incentivos providos pelo *tit-for-tat* e tentam utilizar o contexto social dos *peers* para melhorar a eficiência de sistemas P2P. Por exemplo, [8] exploram o contexto social do *Youtube*¹ e mostra a forte correlação existente entre os vídeos *Youtube* e os que são sugeridos na página de exibição. Assim, eles propõem mecanismos para melhorar a difusão desses vídeos em redes P2P. Os *peers* fazem *cache* dos vídeos e redistribuem entre seus parceiros, inclusive aqueles que estão assistindo os vídeos sugeridos.

Wang et al. apresentam três trabalhos que tratam o contexto social na difusão de *torrents* [41, 40, 39]. Nesses trabalhos, os autores avaliam contatos de longa duração entre *peers* de *swarms torrents* que são divulgados pelo *Twitter*. Segundo os autores, em *torrents* divulgados pelo *Twitter*, há uma grande localidade temporal entre os *peers*. Enquanto nos *swarms* de *torrents* tradicionais, menos de 5% dos *peers* se reencontraram durante os 80 dias de experimento, esse número chega a 35% nos *torrents* divulgados pelo *Twitter*. Essa localidade temporal oferece uma grande oportunidade para melhorar o grau de compartilhamento. Por exemplo, em seus experimentos, os autores mostram que a latência inicial de *download* dos arquivos e o tempo para completar o *download* são melhorados. [15] obtêm constatações semelhantes ao modificar o mecanismo de escolha de nós levando em consideração relações sociais entre *peers*. Essas relações são construídas através da implementação de um *framework* social que funciona em conjunto com o protocolo de escolha de *peers*.

A investigação da disseminação social de *swarms* é também o principal foco da

¹ youtube.com

presente dissertação mas, diferentemente dos trabalhos citados, o desempenho é analisado não do ponto de vista do usuário, mas da topologia dos *swarms* e da distribuição dos sistemas autônomos. Esta ótica permite que se descubram novas formas de aumentar a escalabilidade das redes *Bittorrent* e reduzir o custo do tráfego originado delas.

Outra abordagem para aumentar a contribuição de um *peer* é oferecer incentivos para sua permanência em um *swarm*. [45] estuda o mecanismo do *share-ratio*, amplamente usado em comunidades privadas e demonstra que ele é capaz de aumentar o nível de cooperação de *peers*. Em algumas comunidades foi observado um *share-ratio* acima de 1 para 90% dos usuários.

De maneira geral, os trabalhos apresentados desenvolvem um ambiente controlado de teste e modificações no protocolo para a escolha priorizada de nós sociais. Entretanto, o sucesso dessas abordagens depende de sua ampla adoção por parte dos usuários de redes *Bittorrent*, ou seja, a grande questão é como convencer os milhares de provedores de *internet* e milhões de *peers* a adotarem qualquer tipo de solução que implique em modificações do protocolo ou no gerenciamento dos sistemas autônomos. Desta forma sua aplicação prática se torna limitada. Além disso, soluções baseadas em AS geralmente ferem a noção de neutralidade, que diz que não deve haver acepção de tráfego dentro da rede.

Por outro lado é atraente estudar a eficiência da disseminação social em redes reais como *Twitter* e *Facebook*, devido ao fato de que já estão implantadas e em intenso uso. Além disso, com o uso dessa abordagem, modificações no protocolo ou AS podem ser evitadas.

Neste trabalho, diferente dos anteriores, são analisados três aspectos diferentes do impacto da divulgação de *torrents* em redes sociais. Inicialmente, é quantificada a diferença na saúde dos *swarms* sociais e dos tradicionais. Nesse caso, são verificadas métricas que indicam que um *torrent* social tem melhor desempenho que um tradicional. Mesmo sem nenhuma modificação nos mecanismos existente do *Bittorrent*, as métricas de saúde indicam que os *torrents* sociais podem ter melhores tempos de disseminação dos arquivos, melhores proporções de (*seeders*) e menos *swarms* mortos (*dead torrents*). Em segundo lugar, é abordado a localidade dos *peers* da rede P2P, principalmente quanto ao seu agrupamento por países e ASes e demonstra-se que existe uma melhoria na localidade geográfica e *inter-AS* de *peers*. Por fim, avalia-se a evolução dos *swarms* ao longo do tempo segundo as relações entre seus *peers*, bem como suas métricas de centralidade. Desta forma observou-se que há uma sensível melhora na permanência de *peers* e no agrupamento entre eles, bem como o aumento da duração de suas relações e redução de *freeriding*.

4 METODOLOGIA E COLETA DE DADOS

O presente trabalho possui duas etapas de caracterização distintas, cada qual com sua metodologia, métricas e base de dados. Em ambas as etapas são efetuadas análises comparativas entre *swarms* disseminados socialmente e *swarms* disseminados de forma tradicional, em *sites* de indexação.

A primeira etapa de caracterização é estática e tem como objetivo apresentar um panorama geral de *swarms Bittorrent*. Nela foram coletados *swarms* de conteúdos e tamanhos diversos. Está subdividida em três partes: na primeira, é feita uma descrição detalhada de todo conteúdo disseminado nos *swarms* coletados. São descritas características como tamanho de *swarm*, número de arquivos por *torrent*, distribuição das extensões mais populares, entre outras. Na segunda parte há uma análise comparativa das métricas de saúde de um *swarm*, ou seja, sua proporção de *seeders*, *trackers*, entre outros. Na terceira parte é estudado o agrupamento de *peers* geograficamente.

A segunda etapa tem o objetivo de acompanhar a formação de *swarms Bittorrent* ao longo do tempo. Nesse contexto suas topologias são analisadas segundo métricas de redes complexas, com foco na investigação da duração da interação entre *peers* e características topológicas herdadas por *swarms Bittorrent*, fruto da disseminação social. Nesta etapa, diferentemente da primeira, cada par de *swarm* tem o mesmo conteúdo e tamanhos similares.

4.1 ANÁLISE ESTÁTICA

A análise estática tem por objetivo apresentar uma visão da organização dos *peers* em um *swarm* em termos de sua natureza e sua distribuição geográfica. Entender como os *peers* se agrupam geograficamente pode ajudar no desenvolvimento de mecanismos que tornem o *Bittorrent* um protocolo mais amigável a sistemas autônomos [29].

Nesta etapa foram coletados 16.698 arquivos *torrent* subdivididos em *swarms* tradicionais e sociais. Um *swarm* tradicional é aquele em que seu arquivo *torrent* é publicado em sites indexadores. Um exemplo de *site* de indexação é o *piratebay*¹. Os *swarms* sociais são aqueles em que seus arquivos *torrent*, além de serem publicados de forma tradicional, são também disseminados em redes sociais. Existem diversas comunidades no *Facebook* e *Twitter* dedicadas à disseminação de arquivos *torrent*.

O tempo de coleta foi de 60 dias, entre julho e agosto de 2013, utilizando a rede da Universidade Federal de Juiz de Fora. A coleta dos *swarms* tradicionais foi efetuada através da técnica de *parsing*: a partir da página principal, um algoritmo varre um site de indexação de *torrents* em busca de *links* para arquivos *torrent* e em seguida gera uma

¹ <http://thepiratebay.se/>

lista que é submetida a um gerenciador de *downloads*. O *site* utilizado para a coleta foi o Btmon². Este *site* está organizado da seguinte forma: tem-se uma página principal com uma lista de *links* para arquivos *torrent*. Cada *link* leva à uma página que detalha o conteúdo do *swarm* em particular. Além das informações relacionadas ao *torrent*, existe uma lista de *links* para outros conteúdos relacionados. Com base nessas informações, o coletor de *torrents* limitou-se ao primeiro nível de recursividade de *links*.

Os *torrents* sociais foram obtidos em comunidades do *Facebook*, a partir de um coletor desenvolvido com a API pública do *Facebook*. O coletor varre uma lista de comunidades inicialmente usada como semente. Essa lista foi definida a partir de buscas no *Facebook* por termos como “*Bittorrent*”, “*torrent*” e “*P2P sharing*”. Adicionalmente, são varridas as comunidades relacionadas às encontradas na lista inicial.

Desta coleta obtiveram-se 15.086 *torrents* tradicionais e 1.612 sociais, sendo 38 deles encontrados em ambos os conjuntos que, por representarem uma parcela pequena no total dos dados, foram excluídos das análises apresentadas.

Para efetuar a coleta das informações sobre os *peers*, foi desenvolvido um cliente com o auxílio da biblioteca *libtorrent*³. Este cliente recebe uma lista inicial de *peers* do tracker, bem como informações sobre número total de *seeders* e *peers*. A seguir, ao longo do tempo ele passa a requisitar mais *peers* ao tracker e aos nós DHT. A coleta é efetuada até que o número de *peers* coletado se iguale ao número de *peers* relatado inicialmente pelo tracker ou quando, num período de 1 minuto, não há a descoberta de novos *peers*. Este valor foi estabelecido por observações empíricas feitas em coletas de teste durante o desenvolvimento do cliente. Quando algum desses critérios é atendido, o cliente gera o relatório acumulado das informações e passa para o próximo *torrent*. O relatório também contém a lista de arquivos e de trackers obtida diretamente dos arquivos *torrent*. Para obter as informações sobre localidade geográfica foi utilizada a biblioteca *geoip*⁴. A média de coleta de *peers* por *swarm* foi de $\approx 87\%$. Um sumário dos dados coletados pode ser visto na tabela 2.

A partir dos dados obtidos na etapa de coleta dos dados estáticos dos *swarms*, é possível realizar 3 tipos de análises definidas a seguir:

- Propriedades gerais: corresponde às meta-informações a respeito dos *swarms*, como seu tamanho, tamanho e quantidade de arquivos disseminados e lista de extensões (tipos de arquivo).
- Análise de saúde: neste trabalho, a saúde de um *swarm* está relacionada a dois fatores: redundância de recursos e às características de origem dos *peers*. Em termos

² btmon.com

³ rasterbar.com/products/libtorrent/

⁴ maxmind.com

Quantidade	Amostra tradicional	Amostra social
Endereços IP's	1.280.688	358.463
Participantes	1.434.393	410.858
Países	230	222
Sistemas autônomos	11.366	7.567
<i>Trackers</i>	3.602	821
Arquivos	962.997	23.398
Extensões	1.747	457

Tabela 2 – Descrição da coleta para análise de localidade e saúde

de redundância, um *swarm* com mais *trackers* e maior proporção de *seeders* será considerado mais saudável. Em termos de natureza, um *swarm* com mais *peers* obtidos de forma descentralizada (via extensões como DHT) e que participam de múltiplos *swarms* será considerado mais saudável, devido ao fato de que podem subsistir sem a dependência de um coordenador central.

- Análise de localidade: Diz respeito ao nível de agrupamento local de parceiros, ou seja, para cada país em que o *swarm* tem representantes, deve-se garantir que seus sistemas autônomos possuam grande número de *peers*. Desta forma há maior possibilidade de haver tráfego dentro de uma mesma região geográfica ou ISP, reduzindo o custo energético e monetário. Para isso as medidas de *peers* por AS e por país são apresentadas.

Para descrever as propriedades gerais dos *swarms* (e.g., o tamanho e a quantidade de arquivos compartilhadas) foram utilizadas todas as amostras coletadas. Entretanto, para comparar a localização dos *peers* dos *swarms* e a saúde dos *swarms*, foi utilizado um subconjunto aleatório dos *torrents* tradicionais, de tamanho similar à amostra social. O tamanho dos *swarms* também foi utilizado como critério de escolha. Assim, as comparações são realizadas em conjuntos de mesma ordem de grandeza. A taxa de superposição entre o subconjunto tradicional e o conjunto social é de 1,7%.

4.2 ANÁLISE DINÂMICA

A análise dinâmica tem por meta acompanhar a evolução de *swarms Bittorrent* ao longo do tempo. Devido à dinamicidade das trocas de parceiros do *tit-for-tat*, a topologia de um *swarm* é altamente variável. Espera-se, como mostrado por [15], que a associar contextos sociais à disseminação de arquivos pode prolongar a relação entre *peers* e reduzir *freeriding*, devido à natureza inerentemente duradoura e cooperativa das relações entre amigos em uma rede social.

Nesta etapa, a coleta foi efetuada no período de maio a junho de 2014, novamente através da rede da UFJF. Foram coletados 30 *torrents* de diferentes conteúdos, sendo 15 para disseminação social e 15 para disseminação tradicional. Cada par de *swarms* tradicional/social tinha o mesmo conteúdo e quantidade similar de *peers*. O número inicial de *peers*, como relatado pelos indexadores pode ser visto na tabela 3.

O acompanhamento dos *swarms* foi realizado a partir do seu lançamento, com período de observação de 5 dias. Para garantir que a análise englobasse a maior parte do tempo de vida útil, foram considerados conteúdos cujo interesse possui curta duração como, por exemplo, episódios de seriados. Desta forma, ao contrário da análise descrita em 4.1, esta tem caráter dinâmico e não estático. Por fim, a superposição entre o conjunto tradicional e social foi de 0,7%. Portanto, considera-se que a amostra social e tradicional representam conjuntos disjuntos.

Swarm	Qtd. <i>peers</i> tradicional	Qtd. <i>peers</i> social
1	135	617
2	304	102
3	427	213
4	349	225
5	296	226
6	1800	1500
7	197	113
8	880	800
9	640	630
10	2300	2500
11	2300	2100
12	2000	2800
13	1300	1600
14	2300	2800
15	2000	2000
Total	17228	18226
Endereços IP's	253.962	227.506
Conexões entre <i>peers</i>	17.857.966	25.914.519

Tabela 3 – Tamanho das amostras social e tradicional

Para a coleta dos dados, o cliente foi modificado para trocar pacotes *PEX* entre os seus parceiros. De acordo com [2], o intervalo entre mensagens *PEX* é de 1 minuto, sendo que a cada rodada um *peer* interage com 8 parceiros para trocar *peers*. O número de *peers* enviados por mensagem varia de 50 a 200. O *sniffer wireshark*⁵ foi utilizado para coletar os pacotes durante as 120 horas de coleta. Após esse tempo, outro *torrent* era inicializado. Em seguida os logs gerados pelo *wireshark* eram submetidos a um *parser* que extraía os

⁵ <http://www.wireshark.org/>

endereços IP das mensagens PEX. A estrutura de uma mensagem PEX é detalhada em na subseção 2.3.1.

Tomando como base a frequência de mensagens PEX, temos que um *peer* pode receber sugestões de até 1600 parceiros a cada minuto. Desta forma, em um intervalo de 10 minutos tem-se o recebimento de até 16000 *peers* ($8 \times 200 \times 10 = 16000$), o que é o suficiente para se obter uma visão parcial da rede que possa ser usada como instrumento para as medidas desse trabalho (o maior *swarm* da coleta possui menos de 3000 *peers*). Desta forma o tempo de coleta foi dividido em *snapshots* de 10 minutos, sendo que a rede sobreposta formada pelos *peers* em um *swarm* segue o seguinte modelo:

Seja o grafo não-direcionado e não-ponderado $G(V, E)$ a representação da rede, onde V é o conjunto de nós e E o conjunto de arestas. Um *peer* em um *swarm* é representado por um nó e as arestas descrevem a interação entre os nós. Uma ligação é formada entre dois nós em um snapshot quando há troca de *chunks* entre eles.

Seja o período total de observação igual a T_n , iniciado em $T_1 = 0$. Sem perda de generalidade, considera-se que a visão agregada da rede (modelagem considerando o tempo total de observação T_n) é feita através da construção do grafo estático G .

Seja $\mathcal{G}_t(\mathcal{V}_t, \mathcal{E}_t)$ a representação dinâmica do grafo agregado G , sendo \mathcal{V}_t o conjunto de nós e \mathcal{E}_t o conjunto de arestas. O grafo \mathcal{G}_t e os conjuntos \mathcal{V}_t e \mathcal{E}_t estão indexados no tempo t , onde $T_1 \leq t \leq T_n$. A duração de cada snapshot é definida por $\Delta = T_i - T_{i-1}$ com $1 \leq i \leq N$, onde $N = 720$. Para os 30 *swarms* o total é de 21.600 *snapshots*. Como já dito, o valor utilizado para Δ é de 10 minutos. Um representação da evolução de um *swarm* é mostrada na figura 4.

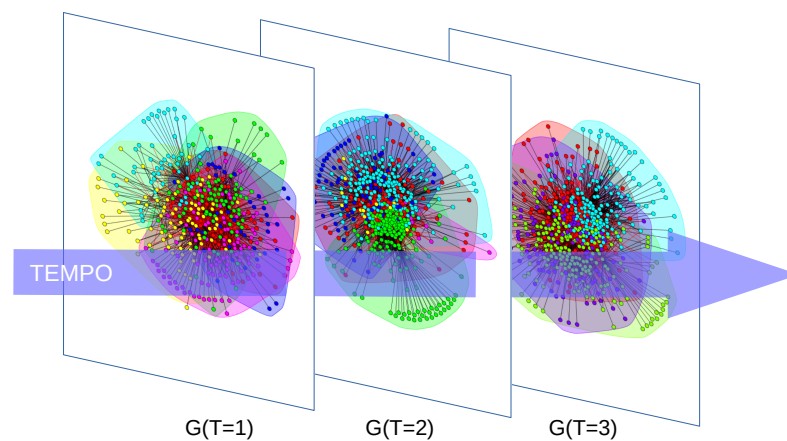


Figura 4 – Representação dinâmica da topologia de rede para $t = 1, 2$ e 3

4.2.1 Identificação de Comunidades

Ao longo do tempo de vida de um *swarm*, os *peers* tendem a se organizar em grupos (comunidades). Uma comunidade tem por característica grande densidade de arestas intra-grafo, com algumas ligações inter-grafo, ou seja, tendem a disseminar mais informação dentro do próprio grupo. O grau de cada nó dentro da comunidade corresponde à sua importância para a comunidade e indica seu nível de colaboração para com os outros *peers*.

Para detectar as comunidades formadas durante a disseminação, foi utilizado o algoritmo *FastGreedy* [9]. Este algoritmo aglomera os *peers* em comunidades segundo a densidade de suas ligações. Mais precisamente, a identificação das comunidades é determinada por uma função que identifica subgrafos densos utilizando uma otimização da medida de modularidade original de Newman [26] através de uma busca gulosa.

Modularidade é uma métrica que permite a avaliação de agrupamentos em uma rede. Ela mede a relevância de uma possível divisão da rede em grupos. A função de modularidade é dada por $Q = \sum_i (e_{ii} - a_i^2)$, onde e_{ii} é a fração das arestas do grafo que estão na comunidade i e a_i^2 é esta mesma fração, mas considerando que as arestas são inseridas de forma aleatória.

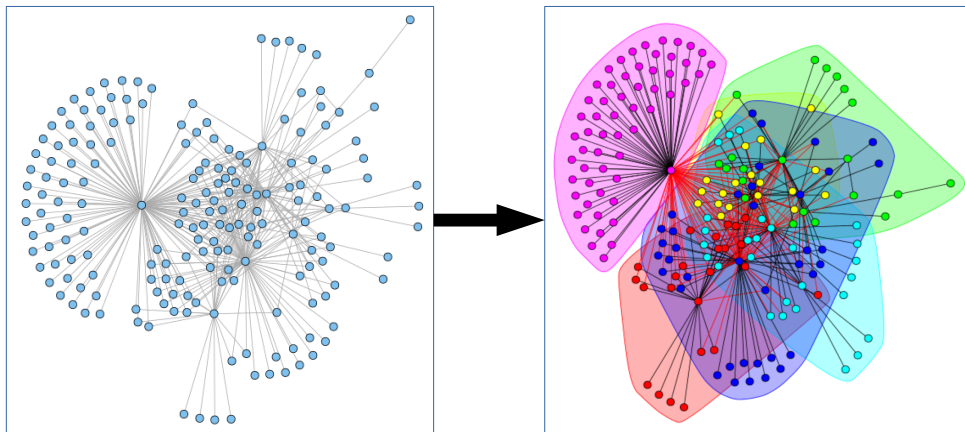


Figura 5 – Exemplo de identificação de comunidades em um *snapshot*

Dentre os algoritmos existentes, o *Fastgreedy* foi escolhido por não ser necessário ter conhecimento estrutural prévio dos grafos analisados. Algoritmos como o *k-means* requerem que se conheça (ou se assuma) o número k de partições totais. Essa abordagem não é adequada para o mecanismo automatizado de detecção e o número de *snapshots* considerado. Nos experimentos apresentados, para cada *swarm*, *FastGreedy* foi executado em cada um dos *snapshots* (ver subseção 4.2).

Para investigar a evolução de uma comunidade ao longo do tempo é necessário observá-la em *snapshots* sucessivos. Entretanto, através da metodologia utilizada, não é

possível identificar uma comunidade de um *snapshot* S em um *snapshot* $S + 1$ subsequente. Para isso, uma solução heurística foi implementada através do uso do coeficiente de *jaccard*: uma comunidade c_i no *snapshot* S tem seu coeficiente medido para as comunidades do *snapshot* $S + 1$. O maior valor indicará a comunidade de $S + 1$ que corresponde a comunidade c_i em S . Na figura 5 é mostrado um *snapshot* com suas comunidades. Cada região possui uma cor diferente, identificando as comunidades detectadas.

5 MÉTRICAS DE INTERESSE

As análises efetuadas nos grafos deste trabalho fundamentam-se em métricas utilizadas em redes complexas. Sem perda de generalidade, as definições utilizadas para os grafos podem ser estendidas para todo *snapshot* \mathcal{G}_t e para suas comunidades $C_i(t)$ onde $1 \leq i \leq c$ e $T_0 \leq t \leq T_N$, que representa a comunidade i no tempo t . O parâmetro c indica o número total de comunidades em cada *snapshot*.

5.1 ANÁLISE ESTÁTICA

Para avaliar a rede de forma estática e comparar as diferenças entre *swarms* tradicionais e sociais, foram utilizadas métricas quantitativas, como número de arquivos, número de *peers*, e também métricas de proporção como o número de *seeders* por *peer*, número de *peers* por País e sistemas autônomos. A seguir essas métricas serão formalizadas.

5.1.1 Tamanho de *swarm*

Esta medida tem por objetivo quantificar e comparar as dimensões dos *swarms* tradicionais e sociais. A dimensão de um *swarm* diz respeito ao número de *peers* que o compõe. Desconsiderando outros fatores, como transitividade e centralidade, o melhor *swarm* é aquele que tem o maior número de *peers*.

5.1.2 Quantidade de arquivos

Esta medida tem por objetivo quantificar e comparar a quantidade de arquivos disseminados em *swarms* tradicionais e sociais. O número de arquivos tem relação com a natureza do conteúdo disseminado: arquivos menores tendem a vir em conjunto, formando coleções (MP3 em uma discografia ou álbum, por exemplo), enquanto arquivos grandes tendem a vir individualmente (imagens de DVD's ou filmes).

Um *swarm* com número de arquivos muito alto não é desejável, pois cada arquivo precisará ter um *infohash* particular, o que pode aumentar a troca de pacotes de controle entre *peers*. Desta forma, ao se comparar os dois tipos de disseminação, um *swarm* com poucos arquivos é melhor do ponto de vista da redução do *overhead* de tráfego. É possível reduzir o número de arquivos em um *swarm* agrupando coleções em arquivos compactados.

5.1.3 Tipos de extensão

Em arquivos de computador, uma extensão é um sufixo que serve para definir a natureza de seu conteúdo. Geralmente é composto de um separador e um identificador de, em média, três letras. Um exemplo de extensão é o MP3. Em um *torrent* pode haver uma quantidade arbitrária de arquivos de diferentes tipos.

Esta medida tem por objetivo demonstrar a natureza do conteúdo disseminado em *swarms* tradicionais e sociais. Espera-se que a variedade de conteúdo social seja menor, devido às comunidades se organizarem baseadas em interesse específicos.

5.1.4 Número de *peers extra tracker*

Quantifica o número de *peers* obtidos pelas redes DHT e PEX. É uma medida de resiliência e demonstra a probabilidade de um *swarm* permanecer estruturado no caso de um *tracker* ficar indisponível.

5.1.5 Número *swarms multitracker*

Como explicado em 2.1, um dos grandes problemas de algumas implementações P2P é o ponto de falha único decorrente da existência de um único coordenador central. Em *Bittorrent*, existem *swarms* que possuem mais de um *tracker*, o que aumenta a resiliência da rede. Portanto essa métrica visa apresentar a proporção de *swarms* que apresentam mais de um *tracker*.

5.1.6 Número de *swarms* mortos

Apresenta a porcentagem de *swarms* que não possuem *seeders* ou não possuem nenhum *peer*. Um *swarm* sem *seeders* tem mais dificuldades ou mesmo impossibilidade de disseminar o arquivo completo.

5.1.7 Número de *seeders*

Essa métrica indica o número de cópias completas disponíveis em um conjunto de *swarms* Bittorrent. É uma medida de redundância e o número maior de *seeders* indica maior resiliência e que o nível de colaboração é mais alto.

5.1.8 Proporção de *seeders* por *peers*

Essa métrica apresenta a proporção entre o número de *seeder* e *peers* de um *swarm*. É uma métrica que ajuda a explicar o nível de cooperação de um *peer* em um *swarm*: um *swarm* com uma proporção alta, próxima de 1, é uma forte evidência de que um *peer* permanece mais tempo cooperando. Em outras palavras, pode indicar um menor número de *freeriders*.

5.1.9 Número de *peers multi swarm*

Essa métrica quantifica o número de *peers* que participam de 2 ou mais *swarms*. Um *peer*, ao participar de múltiplos *swarms*, pode aumentar a quantidade de troca de dados entre parceiros de um mesmo sistema autônomo. A figura 6 demonstra como *peers*

multi swarm podem ajudar na disseminação. Os nós verdes (*multi swarm*) servem de repositório de conteúdo para diferentes *swarms* dentro do AS, reduzindo a probabilidade de seus nós terem de buscar parceiros fora do sistema autônomo.

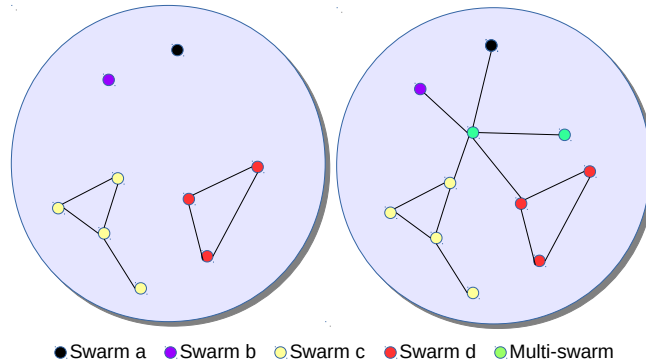


Figura 6 – Melhoria da disseminação na presença de *peers multi swarm*

5.1.10 Número de *peers* por AS e País

Essa métrica enuncia o nível de agrupamento de *peers*. Em outras palavras, para um dado *swarm*, apresenta sua distribuição de *peers* ao longo de sistemas autônomos e países. Se um sistema autônomo tem abundância de *peers* e se isso se repete para cada AS de um país, reduz-se a probabilidade de haver tráfego extra-AS. Um *swarm* rico em *peers* por AS e por país pode priorizar a escolha de nós locais com menor probabilidade de que haja perda de vazão.

5.2 ANÁLISE DINÂMICA

Para avaliar a rede de forma dinâmica e comparar as diferenças entre *swarms* tradicionais e sociais, foram utilizadas métricas de topologia, como o tamanho das comunidades, métricas de *peer*, como grau e centralidade e, por fim, métricas temporais como duração de comunidades, permanência de *peers*, entre outras. A seguir essas métricas serão formalizadas.

5.2.1 Métricas de topologia

5.2.1.1 Tamanho de comunidades

O tamanho de uma comunidade diz respeito ao número de *peers* que ela possui. Esta medida tem por objetivo apresentar as diferenças existentes nas dimensões das comunidades sociais e tradicionais.

5.2.1.2 Número de comunidades

A medida do número de comunidades tem por objetivo apresentar a variação na quantidade de comunidades sociais e tradicionais ao longo do tempo.

5.2.2 Métricas de *peer*

No *tit-for-tat*, como apresentado na subseção 2.2.3, nós que contribuem pouco tendem a ser retirado das vizinhanças locais de seus parceiros. Topologicamente isto significa que um *peer* estabelecerá menos conexões, o que influenciará em suas métricas de centralidade, colocando-o à margem da topologia formada pelos *peers* em um *swarm*. Desta forma a centralidade é necessária para obter informações sobre o nível de contribuição de um *peer*.

5.2.2.1 Grau

O grau de um nó denota a quantidade de arestas adjacentes à ele. É uma medida que indica a quantidade de parcerias estabelecidas por um *peer*. Cada mensagem PEX de B que inclui um nó A qualquer, significa o estabelecimento de uma parceria entre A e B. O grau de um nó também é indicativo do seu nível de contribuição.

A distribuição dos graus dos nós de um *swarm* tendem à seguir a lei da potência [27], onde observa-se uma grande quantidade de nós com grau baixo e uma pequena quantidade de nós com grau alto. Por outro lado, os nós de uma rede social tendem a ter uma distribuição mais igualitária de graus. Desta forma espera-se que essa característica seja herdada pelos *swarms* sociais, reduzindo os nós periféricos cujo grau é muito baixo.

5.2.2.2 Betweenness

Betweenness é uma medida de centralidade do nó definida pelo número de menores caminhos que passam por um vértice. Pode ser considerada uma medida de resiliência e define quantos caminhos serão mais longos caso haja a retirada de um vértice da rede. Segundo [14], dado um vértice v , o seu *betweenness* é definido formalmente como a fração dos menores caminhos entre dois vértices quaisquer que passam por v :

$$C_b(v) = \sum_{i \neq j, i \neq v, j \neq v} g_{ivj} / g_{ij}$$

Onde g_{ivj} é um menor caminho, em termos de número de arestas, de i até j que passa por v e g_{ij} é um menor caminho, em termos de quantidade de arestas, de i até j . Valores maiores representam maiores *betweenness*. No contexto de redes Bittorrent, quanto mais *chunks* um *peer* obtiver, mais importante para a rede ele se tornará, e, se ele for um bom colaborador, estabelecerá mais conexões, aumentando sua centralidade no

swarm. Nós com poucos *chunks* ou que contribuem pouco naturalmente serão encontrados na borda da rede sobreposta.

5.2.2.3 Closeness

O *Closeness* de um nó v é o valor médio de todos os menores caminhos a partir de v . Em outras palavras, ela demonstra o quanto um nó v está distante de todos os outros nós. É uma medida importante para mensurar o tempo de difusão de uma informação a partir de um nó qualquer para todos os outros em um grafo.

Seja v um vértice. Segundo [14], o *closeness* de v é definido formalmente como o inverso da soma de todos os menores caminhos médios a partir de v . Portanto, resultados maiores indicam maior *closeness*:

$$C_c(v) = \frac{1}{\sum_{i \neq v} d_{vi}}$$

5.2.2.4 Transitividade

A transitividade ou coeficiente de agrupamento é uma medida que quantifica a densidade de interconexão entre um nó e seus vizinhos. Ela expressa a probabilidade de existir uma conexão entre dois nós A e C, que por sua vez estão conectados a um nó B. Do ponto de vista das redes complexas, uma alta transitividade significa um alto número de triângulos na rede. Por [42], o coeficiente global de agrupamento \mathcal{C} de um nó é expresso por:

$$\mathcal{C}(\mathcal{G}) = \frac{\text{num. de triângulos}}{\text{num. de triplas conectadas}}$$

Um rede *Bittorrent* utiliza o critério de reciprocidade para estabelecer ligações entre *peers*. Desta forma, seu coeficiente de agrupamento é baixo em relação a um grafo de redes sociais. Espera-se que a disseminação social aumente a transitividade de *swarms Bittorrent*.

5.2.2.5 Taxa de permanência

Esta métrica tem por objetivo medir a duração da permanência de um *peer* em um *swarm*. Seja $T_p\text{inicio}$ e $T_p\text{final}$ os tempos de entrada e de saída respectivamente de um *peer p* em um *swarm*, ou seja, o intervalo de tempo entre a primeira e a última mensagens PEX onde p é relacionado. A permanência de um *peer* em um *swarm s* será dada por:

$$\text{perm}_p(s) = T_p\text{final} - T_p\text{inicio}$$

Onde dado um conjunto de n *swarms*, de mesmo tempo de duração de coleta, em que o *peer* participa, sua taxa de permanência global é definida por:

$$\sum_{i=1}^n perm_p(s_i)$$

Peers em um *swarm* que possuem $T_{pfinal} = T_{pinicio}$ foram considerados com taxa de permanência zero. A ocorrência de taxa de permanência zero pode significar saída precoce de um *peer* ou que ele não é um bom disseminador (*freeriding*), sendo excluído da vizinhança local de seus parceiros. Um nó que não participa da vizinhança local de outros *peers* não tem como ser relacionado em suas mensagens PEX. Os problemas decorrentes de uma alta taxa de *freeriders* em um *swarm* são discutidos em 2.2.4.

5.2.2.6 Número de reencontros entre *peers*

Dado que o tempo de parcerias entre *peers* é de aproximadamente 20 segundos [10] e que o intervalo de mensagens PEX é de 1 minuto [2], toma-se, por simplificação, o número de reencontros de um *peer* B com um *peer* A como a quantidade de mensagens PEX de A que relacionam B e seu conteúdo.

5.2.3 Coeficiente de similaridade de *Jaccard*

O coeficiente de *Jaccard* é uma medida que quantifica o nível de similaridade entre dois conjuntos A e B e é dado por:

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

No contexto desse trabalho, a similaridade pode ser definida a partir de duas óticas distintas: similaridade espacial e temporal (figura 7).

- similaridade espacial: a maior interseção entre *swarms* indica que os *peers* compartilham interesse coincidentes. Em outras palavras, mostra uma evidência de que o contexto social de interesse mútuo pode estar formando proto-comunidades nos *swarms*.
- similaridade temporal: Esse tipo de similaridade indica a duração das relações entre *peers*. Quanto mais um grupo de *peers* se preserva ao longo do tempo, mais saudável e duradoura é esta relação.

5.2.3.1 Duração das comunidades

Durante todo tempo de coleta, os *peers* se organizaram em comunidades, sendo que não existe garantias de que elas irão se preservar integralmente de um *snapshot* para outro. Desta forma, o tempo de vida de uma comunidade é variável.

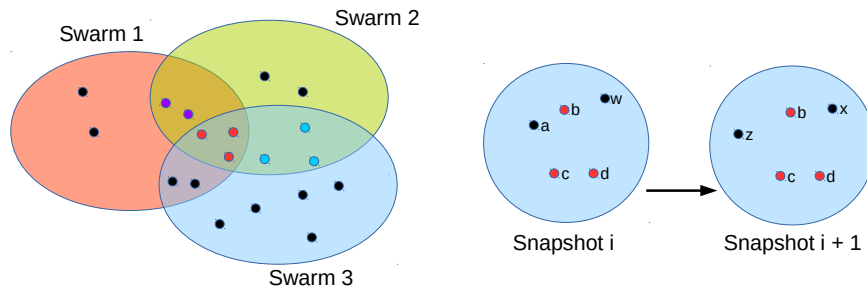


Figura 7 – Representação gráfica da similaridade espacial e temporal

Uma comunidade tende a se dividir em outras em *snapshots* subsequentes. As comunidades frutos desta divisão são chamadas comunidades secundárias. As comunidades que não são frutos da divisão de outras são chamadas primárias.

Para esta métrica foram consideradas apenas as comunidades primárias, que tem início bem definido. Em outras palavras: seja $S(a) = \{s_1, s_2, \dots, s_n\}$ o conjunto de *snapshots* de uma comunidade a . Uma comunidade é dita primária se seu *snapshot* $s_1 \in S(a)$ não pertencer ao conjunto de *snapshots* de outra comunidade.

Intui-se que a duração de uma comunidade é relacionada ao contexto social intrínseco. Desta forma, quanto mais estreita a relação entre *peers*, mais tempo eles permanecerão conectados.

6 AVALIAÇÃO DO IMPACTO DE DISSEMINAÇÃO SOCIAL

6.1 ANÁLISE ESTÁTICA

6.1.1 Propriedades gerais

Os *swarms* analisados apresentam uma população que pode variar de apenas alguns participantes até dezenas de milhares. A Figura 8 apresenta a distribuição dos tamanhos dos *swarms*. A linha azul, marcada com quadrado, apresenta a distribuição de todos os *swarms* avaliados (*torrents* sociais e tradicionais); a linha em preto, marcada com círculo, apresenta a distribuição somente dos *swarms* de *torrents* tradicionais; e a linha vermelha, marcada com asterisco, apresenta a distribuição para os *torrents* sociais.

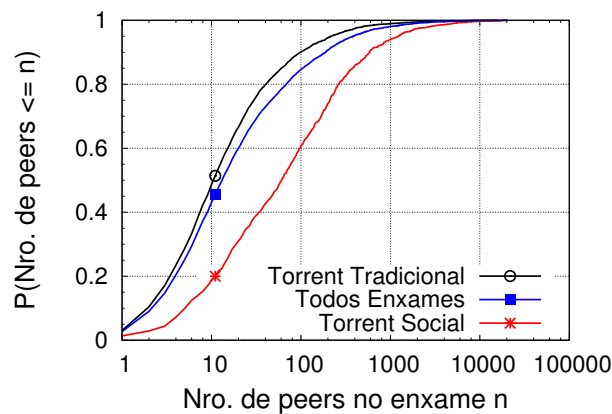


Figura 8 – Distribuição do tamanho dos *swarms* de *torrents*.

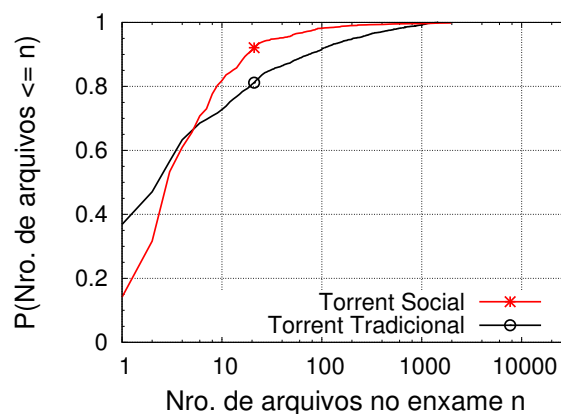


Figura 9 – Distribuição da quantidade de arquivos compartilhados.

Quando se avalia todos *swarms* (linha azul), observa-se que, apesar de existirem *swarms* com milhares de *peers*, a grande maioria apresenta na ordem de até 100 participantes. De fato, mais de 40% apresentam menos de 10 participantes. Há também uma fração não desprezível de *swarms* com apenas 1 participante (por volta de 2,7%).

Os *swarms* sociais e tradicionais apresentam propriedades gerais diferentes. Por exemplo, os *swarms* sociais avaliados tendem a ser maiores que os tradicionais. De acordo com a Figura 8 enquanto 50% dos *swarms* tradicionais apresentam menos de 10 *peers* durante a observação, a mesma fração de *swarms* de *torrents* sociais apresentam até 60 *peers*. Na média, durante o período de observação, um *torrent* social tem três vezes mais *peers* que um *torrent* tradicional. Além disso, cerca de 3% dos *torrents* tradicionais apresentam somente 1 *peer*. Para *torrents* sociais esse número cai para cerca de 1%.

O número de *swarms* com baixo número de participantes possivelmente reflete a distribuição de popularidade entre os *torrents*. Geralmente, arquivos compartilhados atraem a atenção apenas nos períodos de recente divulgação. Com o passar do tempo, esses arquivos são disseminados e perdem popularidade. Seus *swarms* perdem participantes e podem chegar ao caso extremo de se tornarem *torrents* mortos (*dead torrents*), isto é, *swarms* sem cópias completas dos arquivos compartilhados.

A Figura 9 apresenta as distribuições do número de arquivos compartilhados em um único *torrent*. Para avaliar estas distribuições, os *torrents* (sociais e tradicionais) são divididos em 2 grupos: *torrents* que compartilham até 7 arquivos e *torrents* que compartilham mais de 7 arquivos. Considerando somente os *torrents* que compartilham até 7 arquivos, nota-se que os *torrents* sociais tendem a ser um pouco maiores que os tradicionais. Por exemplo, 36% dos *torrents* tradicionais neste grupo compartilham apenas 1 arquivo versus somente 15% dos *torrents* sociais. Já para os *torrents* que compartilham mais de 7 arquivos, nota-se uma inversão de padrões: os *torrents* tradicionais tendem a ser maiores. Por exemplo, cerca de 9% dos *torrents* tradicionais têm mais de 100 arquivos, mas menos de 2% dos sociais têm essa mesma quantidade. Na média geral, os *torrents* tradicionais têm 63,8 arquivos, enquanto os sociais somente 14,51.

Entre os *torrents* avaliados, os dois *torrents* tradicionais com maior número de arquivos compartilhados apresentam 29.780 e 9.845 arquivos. Ambos têm mais de 50 GB de tamanho e compartilham jogos e músicas respectivamente. Em contrapartida, os *torrents* sociais, os dois *swarms* com maior número de arquivos compartilhados têm somente 1.973 e 1.421 arquivos. Os dois conjuntos têm 6 GB e 3 GB respectivamente e ambos compartilham jogos e programas (com extensões dll, cab, exe, .dat, msi, etc).

Finalmente, a Figura 10 apresenta as 20 extensões de arquivo mais comuns em cada tipo de *torrent*. Note que, os arquivos mais comuns dos *swarms* de *torrents* tradicionais são diferentes dos encontrados nos *torrents* sociais (i.e. os eixos “x” dos gráficos são diferentes). Especificamente, as comunidades sociais tendem a compartilhar temas específicos de arquivos (e.g. jogos e filmes). Por esse motivo, as 20 extensões nesse tipo de *torrent* são praticamente todas relativas a vídeo e áudio (mp3, wmv, avi, etc.) e jogos (cab, zip, BNDL, etc.). Enquanto isso, *torrents* tradicionais não são enviesados por tema, tendo uma diversidade maior nas extensões dos arquivos compartilhados. Desta forma, arquivos

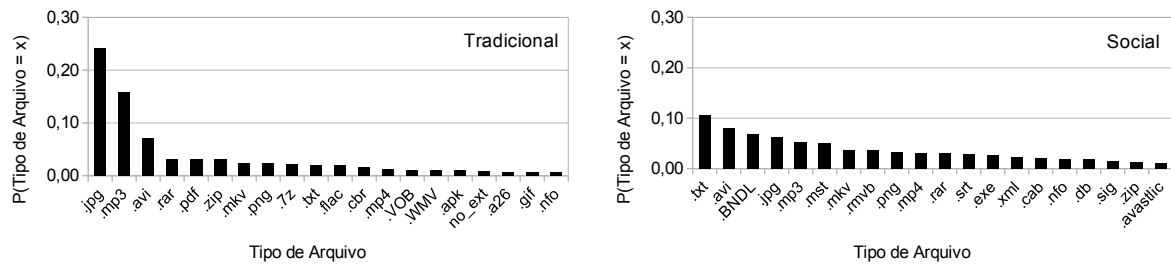


Figura 10 – Top 20 extensões compartilhadas.

de interesse restrito tendem a ser encontrados mais facilmente via mecanismos tradicionais de disseminação.

Tanto para os *torrents* sociais quanto para os tradicionais, é comum existir arquivos de texto descrevendo o conteúdo compartilhado. Assim, observa-se que em ambos os conjuntos há um número expressivo de arquivos “txt” e “nfo”.

6.1.2 Análise de saúde

Neste trabalho, a saúde de um *torrent* indica o quanto esse *torrent* pode ter sucesso em ser compartilhado. Por exemplo, quanto maior a quantidade de cópias completas, ou maior o número de *seeders*, maiores as chances de se difundir os dados de um *torrent*.

	Tradicional	Social
% de <i>swarms multi tracker</i>	81%	92%
% de <i>torrents mortos (dead torrents)</i>	47,45%	28,16%

Tabela 4 – Aspectos gerais de saúde dos *swarms*.

A Tabela 4 resume aspectos gerais relacionados à saúde das duas classes de *torrents*. Nela é possível observar um número significativamente maior de *torrents* sociais que são auxiliados por mais de um *tracker*. Em outras palavras, por terem redundância de assistência, os *torrents* sociais podem apresentar melhor resiliência que os *torrents* tradicionais. Finalmente, há um número maior de *torrents* tradicionais mortos, comparados aos *torrents* sociais. Quase metade dos *torrents* tradicionais analisados já haviam perdido interesse do público e não tinham nenhum *seeder* no *swarm*. Esse número cai para 28% entre os *torrents* sociais.

A figura 11 apresenta a distribuição da quantidade dos *peers* obtidos de forma descentralizada. Em outras palavras, a porcentagem de *peers* obtidos através das extensões PEX e DHT. Nela é possível observar que, em 80% dos *swarms* a quantidade de *peers* sem *tracker* é de 207 contra 27 da disseminação tradicional. Como discutido anteriormente (Capítulo 2), um *tracker* serve para gerenciar um *swarm*. Porém, por ser um ponto de

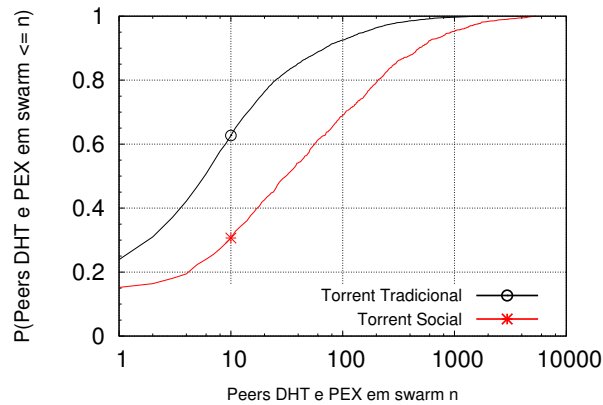


Figura 11 – Quantidade de peers obtidos via PEX e DHT por *swarm*.

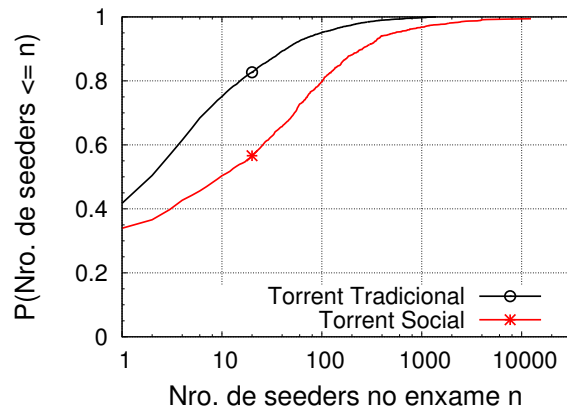


Figura 12 – Distribuição do número de *seeders* em um *swarm*.

centralização, ele pode ser alvo de ataques ou impedimentos judiciais¹. Atualmente, como corroborado pela figura 11, os *torrents* tendem a ser disseminados sem ajuda de *trackers*, o que pode aumentar sua resiliência.

Em números absolutos, os *torrents* sociais tendem a ter mais *seeders* que os tradicionais. A Figura 12 apresenta as distribuições acumuladas dos números de *seeders* de um *swarm*. Note que 80% dos *torrents* tradicionais têm menos de 16 *seeders*. Em comparação, a mesma fração de *torrents* sociais chega a ter 102 *seeders*. Em média, *torrents* sociais têm 137,3 *seeders* enquanto os tradicionais têm apenas 39,06.

Um número maior de *seeders* pode ser reflexo de um maior número de participantes no *swarm*. Assim, uma outra métrica que reflete a saúde de um *torrent* é a razão entre o número de *seeders* e o número total de *peers* no *swarm*. A Figura 13 apresenta as distribuições dessa proporção para *torrents* sociais e tradicionais. Os *torrents* sociais também tendem a ser melhores que os tradicionais quanto a esta métrica: 60% dos *torrents* sociais têm mais de 28% de seus participantes atuando como *seeders*, enquanto que somente 40% dos *torrents* tradicionais têm uma proporção de *seeders* nesta faixa. A maior

¹ No caso de compartilhamento de conteúdo ilegal ou com restrições de *copyright*.

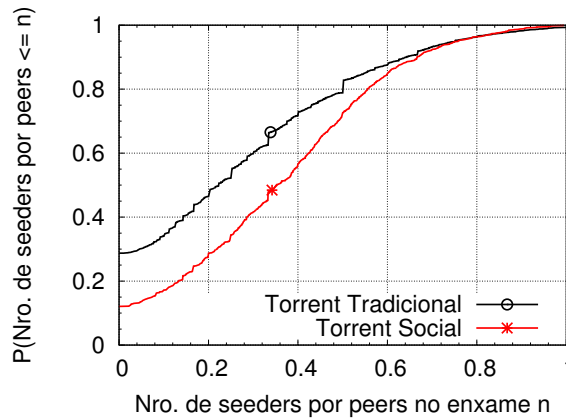


Figura 13 – Distribuição da proporção *seeders/peers* em um *swarm*.

acumulação de *seeders* pode ser uma evidência de sua tendência a permanecer por mais tempo em *swarms*. Isto pode ser decorrente do fator colaborativo inerente às comunidades de distribuição de conteúdo no *Facebook*: amigos formam comunidades para contribuir entre si em um objetivo comum, que no caso é disseminar conteúdo de determinado gênero. O tempo de vida de uma comunidade é reduzido se a colaboração entre seus membros for pequena.

A incidência de grupos de *peers* que participam de mais de um *swarm* também pode ser um indicador de saúde de um *torrent* uma vez que, neste caso, realocação e agrupamento dinâmico de *swarms* podem melhorar o desempenho do compartilhamento dos arquivos pertencentes aos *swarms* com *peers* em comum [12].

A Figura 14 mostra a distribuição acumulada do número de *peers* que participam de mais de um *swarm* em cada tipo *torrent* analisado. Nitidamente, observa-se que os *torrents* sociais apresentam um número maior de *peers* com esse tipo de característica. Na média, os *torrents* sociais têm cerca de 78 *peers* que participam de múltiplos *swarms*, e os tradicionais apenas 16. Em termos percentuais, cerca de 20% dos *peers* de *torrents* tradicionais participam de apenas 1 *swarm*, enquanto nos *torrents* sociais, esta proporção é de 5%.

Intuitivamente, *peers* com interesses em comum tendem a compartilhar também a participação em múltiplos *swarms* e assim, o protocolo Bittorrent poderia usar isso para estabelecer parcerias de longa duração. Além da melhora das parcerias formadas, a participação em múltiplos *swarms* pode ser explorada para balancear a carga (contribuição) entre os *peers* dos vários *swarms* envolvidos.

Para avaliar a coparticipação de *peers* em múltiplos *swarms* foi analisado o coeficiente de *Jaccard* espacial par a par, entre os conjuntos de participantes, para cada tipo de *torrent*.

A Figura 15 apresenta a distribuição dos coeficientes de *Jaccard* encontrada entre

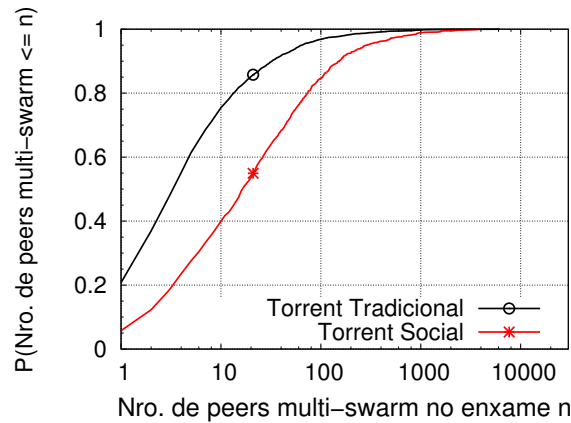


Figura 14 – Distribuição de *peers* que participam de múltiplos *swarms*.

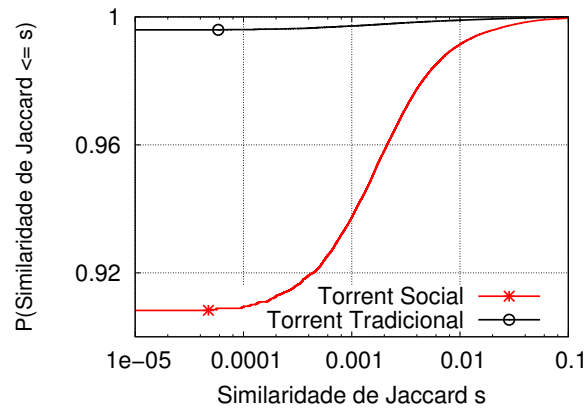


Figura 15 – Distribuição do coeficiente de similaridade de *Jaccard*.

os *swarms* sociais e tradicionais. Claramente, observa-se o efeito da disseminação social no surgimento de grupos de *peers* em comum entre *swarms*: enquanto essa característica é praticamente ausente em *swarms* tradicionais, a mesma se torna evidente na difusão social, mesmo levando-se em conta o fato de que o *Facebook* não tem foco em distribuição de conteúdo e que os *swarms* sociais também podem ser disseminados em sites de indexação, o que poderia reduzir suas interseções.

Para melhor visualização de como a disseminação social faz surgir grupos de *peers* comuns, a Figura 16 apresenta as matrizes de similaridade para os *swarms* tradicionais e sociais. Cada cruzamento de linha e coluna dessa matriz representa a semelhança, segundo o coeficiente de *Jaccard*, entre 2 *swarms*. O coeficiente de similaridade entre 2 *swarms* varia de baixo (branco) a alto (azul escuro). Enquanto a matriz dos *torrents* tradicionais indica baixa similaridade entre todos os pares de *swarm* desse tipo, a matriz de *torrents* sociais indica o contrário: uma grande proporção de pares de *swarm* de *torrent* social tem evidente similaridade. Observa-se então uma possível superposição entre o grafo de amigos do *Facebook* e os *peers* do *swarm*, o que poderia estar induzindo a formação de proto-comunidades de *peers* em decorrência da disseminação social.

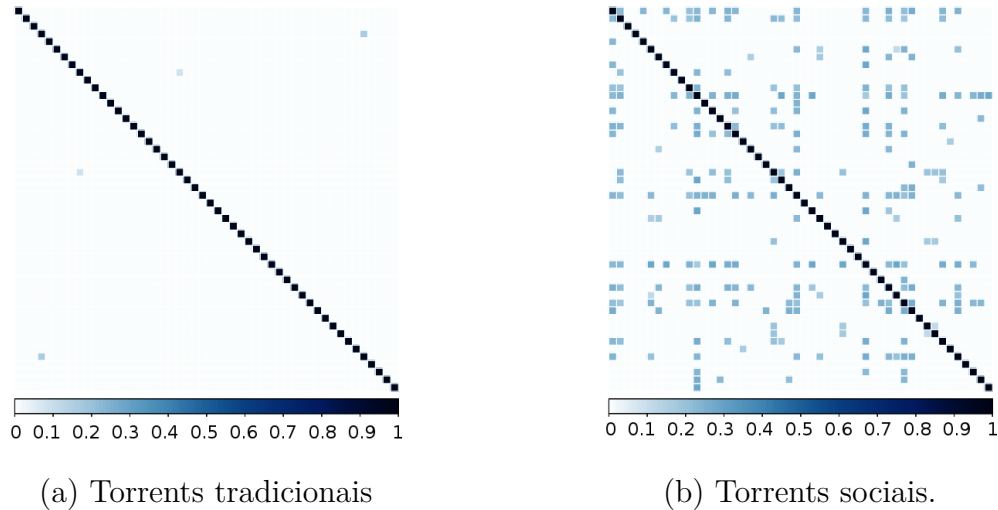


Figura 16 – Matriz de similaridade.

6.1.3 Análise de localidade

De forma geral, aplicações P2P não consideram localização na rede física para criar as parcerias da rede sobreposta [34]. O mesmo ocorre com o Bittorrent. Seu protocolo tenta maximizar o volume de dados trocados entre os *peers*, sem tomar conhecimento de suas localizações geográficas ou outras propriedades da rede físicas [38]. Isso pode causar um grande volume de tráfego entre ASes ou ISPs, e conseqüentemente, alto custo.

Há várias propostas para lidar com os problemas que o desconhecimento da rede física pode causar na construção da rede P2P. Uma abordagem comum é enviesar a seleção de parcerias e favorecer aquelas entre *peers* que são localizados próximos uns aos outros (i.e. *peers* localizados em um mesmo ISP, AS ou país) [34, 38]. Assim, uma comparação da localização dos *peers* nos *swarms torrent* permite avaliar o potencial de eficácia desta estratégia.

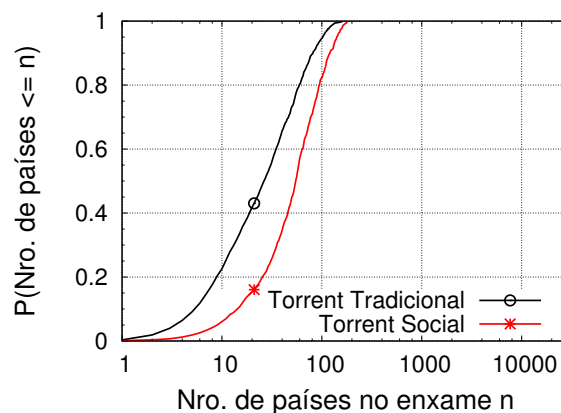


Figura 17 – Número de países distintos em um *swarm*.

Os *torrents* sociais e tradicionais apresentam diferenças substanciais com relação à localização de seus *peers*. Por exemplo, observa-se um número maior de países entre os *peers*

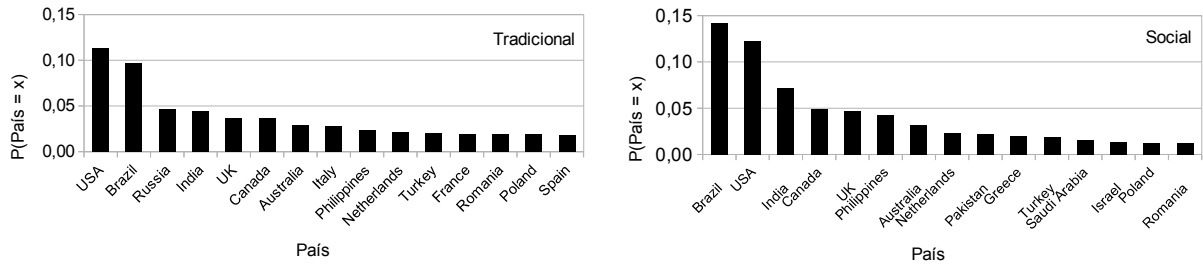


Figura 18 – Top 15 países mais frequentemente encontrados nos *swarms*.

de um *torrent* social. A Figura 17 mostra a distribuição do número de países identificados nos *swarms* analisados. Em média, os *peers* nos *torrents* sociais estão localizados em até 30 países distintos, contra apenas 14 países nos *torrents* tradicionais.

A Figura 18 apresenta os top-15 países mais frequentemente encontrados em cada tipo de *torrent*. Observar que, embora as distribuições sejam razoavelmente semelhantes, os países são diferentes. A distribuição para *torrents* sociais é influenciada pela distribuição dos países mais comuns no *Facebook*². Países como Índia, Filipinas e Brasil tiveram suas posições melhoradas no *ranking* para *torrents* sociais, o que indica uma sobreposição entre as redes social e P2P.

Como mencionado, a localização topológica dos *peers* na rede física também pode ser explorada para favorecer estabelecimento de parcerias entre pares localizados no mesmo AS, por exemplo, reduzindo assim o tráfego entre ASes diferentes. Logo, a localização dos *peers* nos *swarms* em termos de ASes também é analisada.

A Figura 19 apresenta o número de ASes encontrados em cada *swarm* e seu respectivo tamanho (em número de *peers*). Não há grandes diferenças entre os *swarms* sociais e os tradicionais. Para ambos os tipos de *torrents*, pode se observar uma correlação não linear entre o tamanho dos *swarms* e a quantidade de ASes encontrados entre seus participantes. Além disto, em ambos os casos, os *swarms* apresentam tipicamente até 800 ASes diferentes entre seus participantes. Entretanto, há *swarms* com *peers* distribuídos por até 1800 ASes diferentes.

De acordo com a Figura 20, *torrents* sociais apresentam maiores proporções de *peers* por país e por AS. Em outras palavras, a localidade (por país e por AS) é maior nos *torrents* sociais. Praticamente 50% dos *torrents* tradicionais tem menos de 29 *peers* por país. Esse número é significativamente maior nos *torrents* sociais: 50% dos *torrents* chega a ter até 89 *peers* por país. A diferença na relação de *peers* por AS é mais estreita, mas ainda assim impactante. Enquanto 60% dos *swarms* tradicionais têm menos de 10 *peers* por AS, a mesma fração de *swarms* de *torrents* sociais apresenta até 22 *peers* por AS.

Esses números deixam claro que o contexto social de *peers* podem influenciar seu

² <http://www.quintly.com/blog/2013/03/facebook-country-statistics-march-2013/>

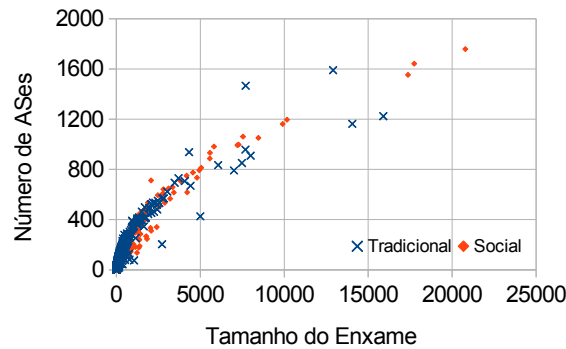
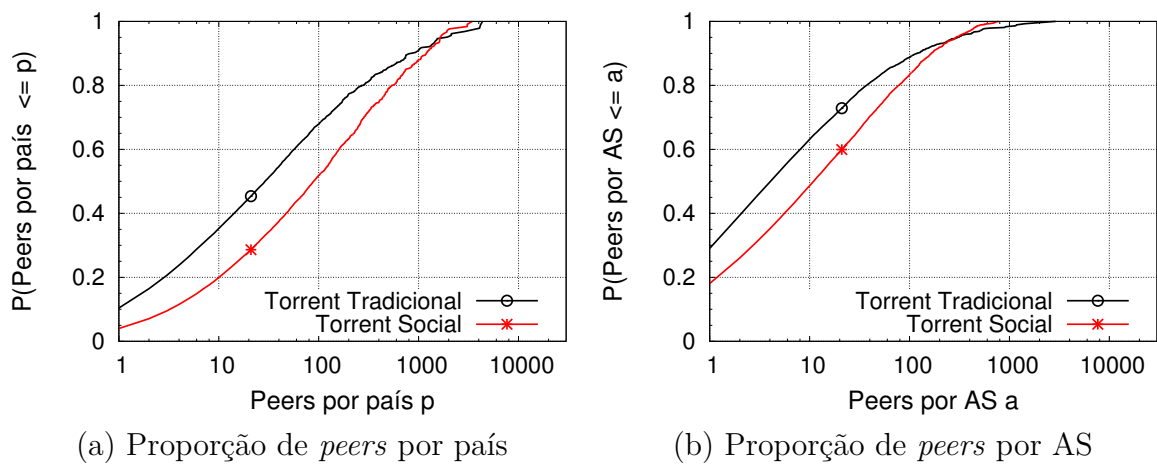


Figura 19 – Número de ASes por *swarm*.



(a) Proporção de *peers* por país

(b) Proporção de *peers* por AS

Figura 20 – Localização dos *peers* de um *swarm torrent*.

agrupamento geográfico: propriedades como idioma e interesses mútuos têm forte relação com a localização espacial de um *peer*. Desta forma, caso o protocolo *Bittorrent* utilize conhecimento das redes físicas na construção da rede P2P, *torrents* sociais apresentarão melhores resultados no agrupamento de *peers*. Conseqüentemente, *torrents* sociais poderão ter melhor taxa de transferência, difusão de dados mais rápida e menor tráfego de dados entre ASes. Mesmo em casos onde não há escolha de *peers* orientada a localização geográfica o *swarm* tem menor probabilidade de trocar dados entre *peers* distantes.

6.2 ANÁLISE DINÂMICA

Como apresentado na seção 4.2, a análise dinâmica acompanha a evolução da topologia de *swarms Bittorrent* ao longo do tempo. O intervalo de observação engloba desde o anúncio do *torrent* pelos disseminadores até o estado de estabilização da topologia, onde o *swarm* já atingiu sua plena formação. Neste período, a superposição acumulada dos conjuntos social e tradicional não ultrapassou os 0,7%. Dessa maneira pode-se considerar que são conjuntos que não possuem relação entre si. Além disso é uma evidência de que o conjunto tradicional tem pouca representatividade de *peers* que possam ter ingressado via

redes sociais.

Durante o intervalo de coleta, os resultados obtidos corroboraram as tendências observadas na análise estática de *swarms*: como observado na tabela 5, os conjuntos ativos dos *peers*, representados por suas mensagens *PEX*, apresentaram um ganho significativo na regionalidade, com um aumento de mais de 2 vezes na proporção de relacionamentos entre *peers* num mesmo país e de mais de 3 vezes em *peers* de um mesmo sistema autônomo.

Tipo	Num. PEX intra-País	Num. PEX total	Proporção
Tradicional	1.251.577	17.857.966	7%
Social	3.996.369	25.914.519	15,4%
	Num. PEX intra-AS	Num. PEX total	Proporção
Tradicional	199.766	17.857.966	1,1%
Social	931.187	25.914.519	4%

Tabela 5 – Regionalização de relações entre *peers*.

Muitas outras conclusões da análise estática foram confirmadas através das métricas apresentadas na análise dinâmica, como a maior permanência e taxa de contribuição de *peers* em um *swarm* e a herança das características topológicas de redes sociais em *swarms* disseminados socialmente.

6.2.1 Caracterização de comunidades

Como discutido na subseção 4.2.1, a identificação das comunidades se deu através do algoritmo *FastGreedy*, que aglomera *peers* em subgrafos segundo a densidade de suas ligações. É importante observar que nem todos os *peers* serão incluídos nas comunidades. Desta forma o montante de *peers* das comunidades é geralmente muito menor do que o total de *peers* de um *swarm*.

Uma rede social tem por característica a longa duração das relações entre seus constituintes, em outras palavras, as suas conexões tendem a não se modificar profundamente com o passar do tempo. Especula-se que a disseminação social possa transmitir traços dessa característica para *swarms Bittorrent*. Nesse contexto, são apresentadas as métricas que analisam a influência do contexto social na duração de comunidades e permanência de *peers*.

As figuras 21 e 22 mostram o tamanho e o número de comunidades dos dois tipos de disseminação, considerando-se todos os *snapshots*. Nelas é possível ver que o tamanho das comunidades é similar para ambas (75,8 *peers* para *swarms* sociais e 69,7 *peers* para *swarms* tradicionais), mas que, para disseminação social, prevalecem os *swarms* com até 5 comunidades e para disseminação tradicional prevalecem *swarms* com mais de 5 comunidades, sendo a média de aproximadamente 5,6 comunidades para os dois tipos de disseminação. O desvio padrão para a quantidade de comunidades é de 2,97 para

a disseminação tradicional e de 2,40 para disseminação social (para mais detalhes, ver apêndice A). Com base nessas informações, é possível afirmar que, mesmo que a quantidade média de comunidades seja a mesma, a variação dessa quantidade com o tempo é maior em *swarms* tradicionais. Isto pode significar uma evidência de que as comunidades em *swarms* tradicionais são menos estáveis ao longo do tempo, tendo seus *peers* migrando de uma comunidade para outra ou mesmo saindo precocemente do sistema.

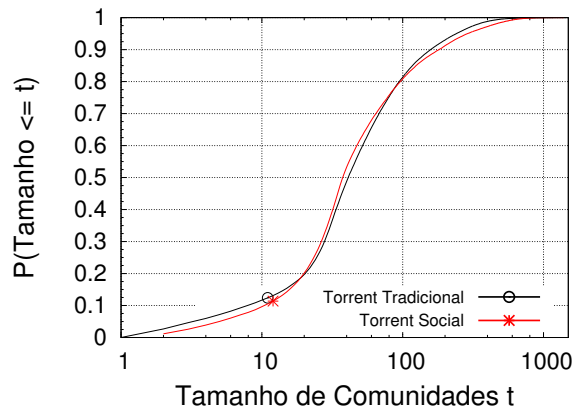


Figura 21 – Tamanho de comunidade

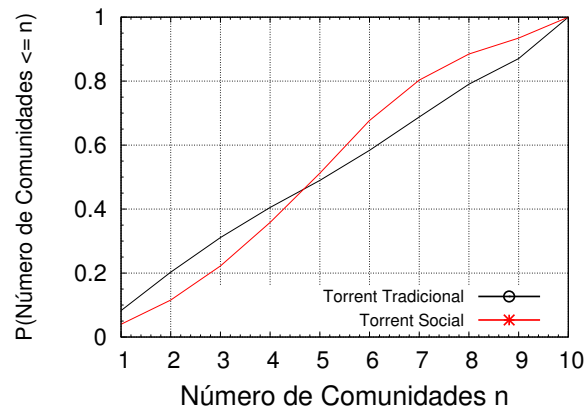


Figura 22 – Número de comunidades por *swarm*.

Essa evidência ganha mais força pela análise do comportamento temporal das comunidades. A figura 23 apresenta o coeficiente de *jaccard* para as comunidades de *swarms* tradicionais e sociais. O coeficiente de *jaccard* mede a porcentagem de uma comunidade que é preservada em *snapshots* consecutivos. Observa-se no gráfico que o corpo da curva social está deslocado para direita em relação a curva de *swarms* tradicionais, significando que uma comunidade de *swarm* social é mais consistente ao longo do tempo. Em disseminação social, 80% dos *swarms* preservam até 32% da integridade de uma comunidade. Em contrapartida, *swarms* tradicionais preservam até 26%. A figura 24

mostra a evolução da média da integridade dos *swarms* ao longo do 720 *snapshots*. É possível observar que, na maior parte do tempo, a curva social apresenta valores maiores dos que o da disseminação tradicional.

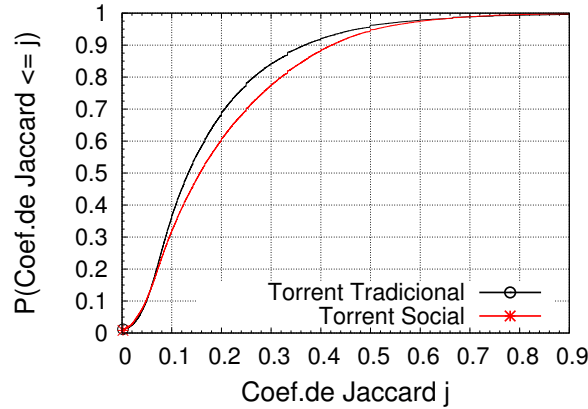


Figura 23 – Coeficiente de *Jaccard* para comunidades.

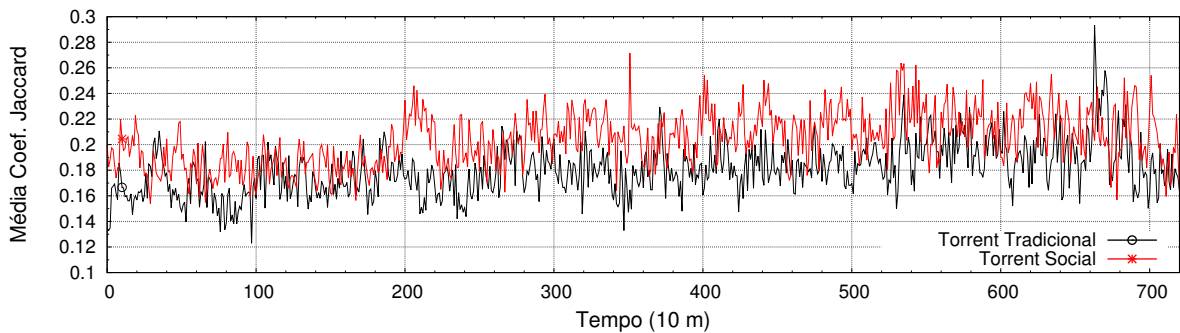


Figura 24 – Coeficiente de *Jaccard* médio durante período de coleta.

Uma mensagem *PEX* é composta por uma parcela da vizinhança local de um *peer*, que por sua vez, consiste dos parceiros com os quais ele está conectado. A referência a um *peer* em uma mensagem *PEX* é indicativo de que ele se encontra ativo em um *swarm*. Portanto o intervalo entre a primeira e a última referência pode ser tomado como seu tempo de permanência em um *swarm*. Um maior tempo de permanência de *peers* aumenta a probabilidade de que eles se reencontrem ao longo do tempo para a troca de informações.

Como mostra a figura 25, o tempo permanência individual de *peers* sociais é maior em *swarms* sociais: em 60% dos *peers* sociais a permanência é de até 128,4 minutos contra 75,05 minutos em *swarms* tradicionais. Além disso a fração de *peers* com permanência zero é 7% maior em *swarms* tradicionais, totalizando 35% contra 28% de *swarms* sociais.

Outra medida que leva em consideração as mensagens *PEX* é a taxa de reencontro entre *peers*. Como explicado na subseção 5.2.2.6, o intervalo entre mensagens *PEX* é de,

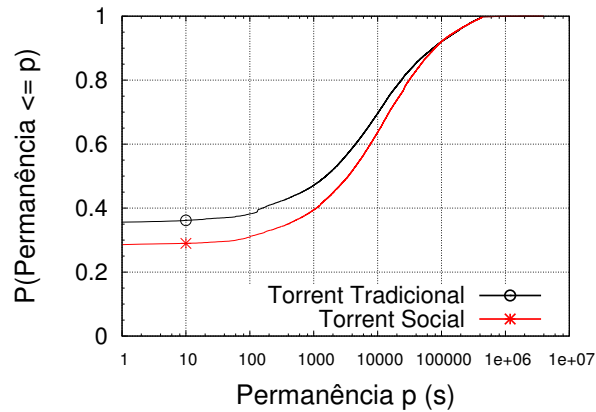


Figura 25 – Tempo de permanência agregada individual de *peers*.

no mínimo, 1 minuto e a duração de um conjunto ativo é de 20 segundos. Desta forma, por simplificação, considerou-se neste trabalho que a referência a um *peer* em uma mensagem PEX pode ser entendida como um reencontro (reingresso em conjunto ativo). Nesse sentido a disseminação social obteve um ganho de 29% na quantidade de reencontro de *peers*, o que significa um aumento de mais de 707 mil nessa métrica. Esta é uma evidência de que os *peers* tendem a colaborar mais em *swarms* sociais.

Com relação à duração das comunidades primárias (subseção 5.2.3.1), observa-se que em *swarms* sociais, seu tempo de vida é superior. De acordo com a figura 26, A média de duração de uma comunidade social é 111 horas contra 105 horas da disseminação tradicional. Este é um indicativo da possível herança em termos de estabilidade de ligações recebida pelos *swarms* sociais.

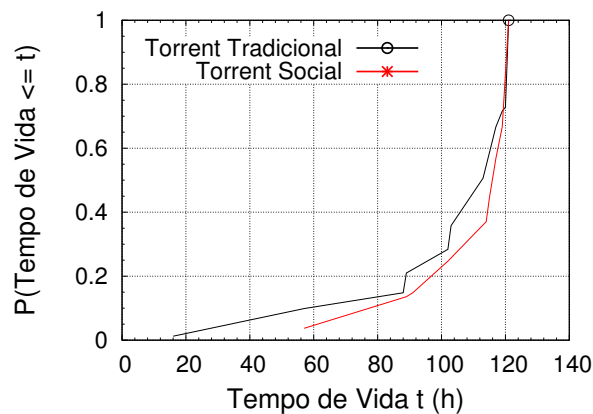


Figura 26 – Taxa de duração de comunidades.

Os fatores apresentados, em conjunto, são indício da sensível melhoria apresentada pela disseminação social na preservação de comunidades ao longo do tempo. A permanência de *peers* por mais tempo pode significar a redução de *freeriders* em *swarms*. Desta forma,

adotar contextos sociais para prolongar relações se mostra factível, mesmo em situações onde os *swarms* têm uma fração variável de *peers* sociais, o que é o caso deste trabalho.

6.2.2 Centralidade de *peers*

No contexto das redes Bittorrent, as medidas de centralidade têm por objetivo investigar o nível de importância de um *peer* dentro de um *swarm*. A importância de um *peer* para um *swarm* está relacionada a fatores de colaboração com o sistema. Quanto mais *chunks* do conteúdo disseminado um *peer* tiver, mais importante ele se torna e, por consequência, mais parceiros tenderão a conectar-se com ele para troca de informações. Por consequência disso, seu grau tenderá a aumentar. Baseado nas melhorias apresentadas na seção 6.2.1, intui-se que a centralidade média dos *peers* tende a sofrer um aumento por consequência dos maiores níveis de colaboração e permanência.

No decorrer das métricas apresentadas, optou-se por medir a centralidade em comunidades, e não no *swarm*, devido ao fato de que as páginas de *torrents* no *Facebook* também estão organizadas em comunidades. Além disso, o critério de detecção do *FastGreedy* leva em consideração a densidade das ligações. Por esses fatores, acredita-se que a acurácia da influência social será maior ao se medir centralidade das comunidades em detrimento de *swarms*.

A primeira medida de centralidade apresentada é a distribuição dos graus. A figura 27 corrobora o aumento de nós com graus altos em um *swarm*. Nela, tem-se que o grau médio de um nó social é 4, contra 3 da disseminação tradicional. Além disso, 85% dos nós sociais têm até grau 4 contra 2 dos *peers* tradicionais e há uma redução de 16% nos *peers* periféricos, que possuem grau igual a 1. Estes resultados demonstram a influência da rede social na distribuição dos graus de *swarms Bittorrent*. Além disso, um nó com alto grau é indicativo de que ele, ou tem uma vizinhança maior, ou faz parte de mais vizinhanças locais, o que evidencia uma tendência de maior colaboração.

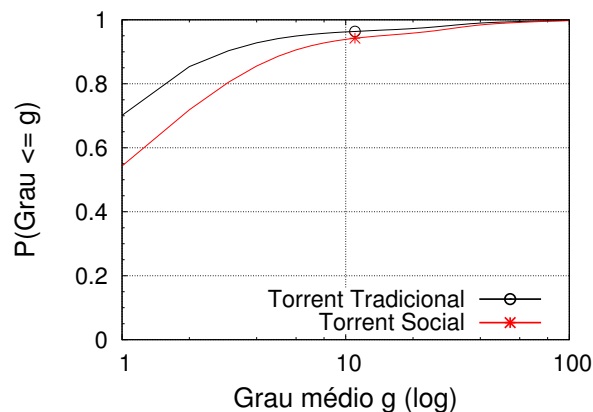


Figura 27 – Grau médio de *peers* em *swarms*.

A contribuição da disseminação social no aumento da centralidade média pode ser observada também no *betweenness*. Esta métrica quantifica o número de menores caminhos que passam por um *peer*. Desta forma, *peers* que estabelecem poucas conexões tendem a permanecer na borda do grafo que representa os *swarms Bittorrent*. A figura 28 mostra um exemplo de uma comunidade social e tradicional. Nela é possível observar que a centralidade é mais bem distribuída entre os nós da comunidade social. Já na comunidade tradicional, observa-se um nó com alta centralidade e o restante dos nós na periferia da rede, indicando que poucos deles são responsáveis por prover à rede os *chunks* mais raros do conteúdo disseminado. No exemplo da figura 28, retirar o nó central afetaria de forma significativa a disseminação de conteúdo.

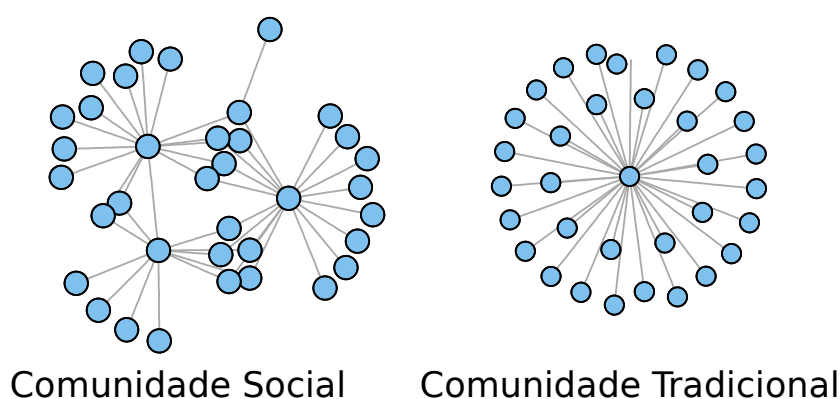


Figura 28 – Exemplo da organização de nós em comunidades tradicionais e sociais.

A tendência observada na figura 28 é confirmada pelo gráfico da figura 29. É possível observar uma melhora de centralidade para *swarms* sociais em até 85% dos seus *peers*. Além disso é importante observar que, na disseminação social houve uma menor prevalência de *betweenness* zero (58% contra 73% dos *swarms* tradicionais). Desta forma, observa-se que uma eventual saída de um *peer* impacta menos na disponibilidade de conteúdo em um *swarm* social.

Um *swarm Bittorrent*, como dito, conduz suas parcerias através de fatores de reciprocidade, desta forma, muitas conexões são formadas entre *peers* distantes. Uma rede social, por outro lado, possui muitas conexões locais e algumas poucas ligações entre comunidades. Por isso a transitividade de um grafo social tende a ser maior que a de um *swarm Bittorrent*.

O maior agrupamento de *peers* é outra característica social herdada pelos *swarms*. Essa influência pode ser observada na distribuição de transitividade global dos *peers* dos *swarms* apresentada na figura 30. Nas medidas efetuadas, a disseminação social propiciou um aumento da transitividade em *swarms Bittorrent*: Em 80% dos nós, a transitividade é

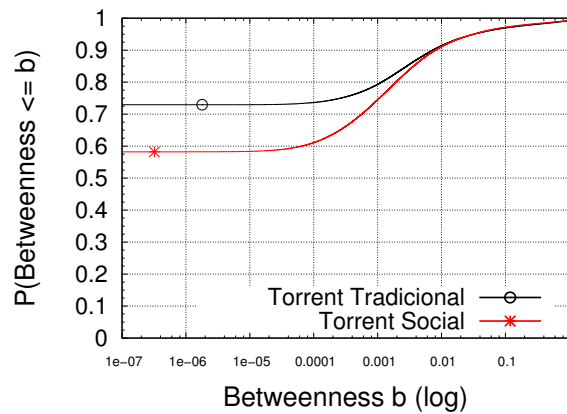


Figura 29 – *Betweenness* em comunidades.

até 10 vezes maior que a da disseminação tradicional. Além disso houve uma redução de 6% na transitividade zero, indicando uma possível melhora global de relacionamento entre *peers*.

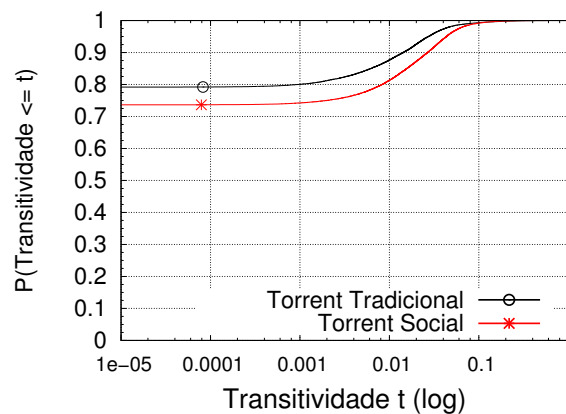


Figura 30 – Transitividade média em comunidades.

A figura 31 mostra outro indicador de centralidade: o *closeness* indica o nível de proximidade de um *peer* em relação a todos os outros. Uma informação disseminada a partir de um *peer* com *closeness* baixo pode levar mais tempo para ser disseminada para todos os outros nós do que a partir de um nó com *closeness* alto. Como demonstrado pela fórmula descrita na subseção 5.2.2.3, maiores valores indicam maiores *closeness*.

De acordo com a figura 31 a proximidade em *swarms* tradicionais é maior: 70% dos nós tradicionais tem *closeness* de pelo menos 0,38 contra 0,35 em *swarms* sociais. Isto se deve à característica de agrupamento mais intenso demonstrada nas medidas de transitividade. Como há mais parcerias entre *peers* próximos e menos entre *peers* distantes, a média de saltos para se atingir toda rede é maior. Essa característica pode ser observada de forma clara na figura 28. Nesse contexto, esta medida de centralidade é mais uma

evidência da maior clusterização herdada pelos *peers* de *swarms* sociais.

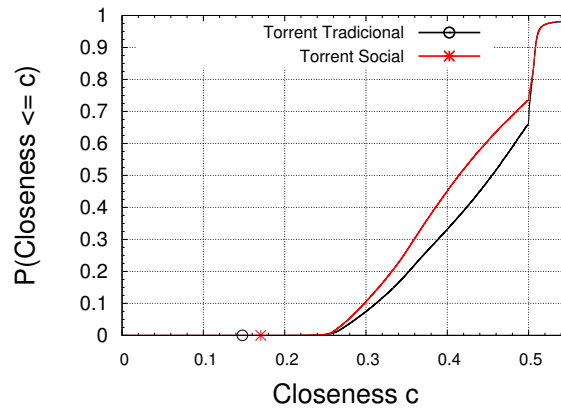


Figura 31 – *Closeness* em comunidades.

Os resultados apresentados nesse capítulo demonstram que associar contexto social à disseminação de arquivos *Bittorrent* melhora as relações entre *peers* ajudando a reduzir *freeriding*. Essa melhora é condicionada ao nível de disseminação social de um *swarm*: como não existem mecanismos para mapear um *peer* em seu correspondente no *Facebook*, não foi possível deduzir a proporção de *peers* que obtiveram o arquivo da rede social em detrimento de sites de indexação. Entretanto existe uma forte intuição de que, quanto maior a proporção de *peers* sociais, maior será a melhoria, devido ao fato de o *swarm* ter absorvido, em parte, as características topológicas de rede sociais, como, por exemplo, a maior transitividade e relacionamento mais duradouro.

7 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho apresentou uma caracterização do impacto da disseminação social de arquivos *Bittorrent*, dividida em três aspectos distintos: a aglutinação geográfica de *peers*, fatores de resiliência e qualidade das relações entre *peers*. Entender como essas características são influenciadas por contexto social pode ajudar no desenvolvimento de melhorias no protocolo *Bittorrent*, que visam torná-lo mais amigável a sistemas autônomos e aumentar a taxa de contribuição dos participantes de um *swarm*, reduzindo o fenômeno de *freeriding*.

Os dados utilizados nesse trabalho são fruto de uma coleta efetuada no ambiente real de disseminação *Bittorrent*: um *crawler* foi desenvolvido para ingressar nos *swarms* e coletar suas informações estáticas, como dados geográficos de *peers*, arquivos e *trackers*, bem como a evolução temporal da topologia da rede sobreposta. Em seguida esses dados foram analisados sob diversos aspectos, tais como agrupamento geográfico, resiliência, relações entre pares e centralidade.

Do ponto de vista estático, o *swarm* foi analisado segundo suas métricas de saúde, como proporção de *seeders*, número de *trackers*; métricas de localidade, como a distribuição de *peers* por sistema autônomo e países; e características gerais, como número e tipos de arquivo disseminados.

Para avaliar a dinamicidade temporal da rede, a topologia foi reconstituída segundo o modelo de representação por grafos, onde cada vértice representa um *peer* e cada aresta uma ligação entre eles. Os grafos foram particionados em comunidades pelo algoritmo *FastGreedy*. Em seguida elas foram analisadas segundo métricas de centralidade como *betweenness* e *closeness*, bem como grau e transitividade, com o objetivo de observar as características herdadas do grafo social onde o *swarm* foi disseminado. A qualidade das relações entre os *peers* foi analisada através da taxa permanência de *peers* e de comunidades. A integridade de uma comunidade ao longo do tempo foi calculada através do coeficiente de *Jaccard*.

Os resultados encontrados na caracterização evidenciam melhoramentos originados da adoção da disseminação social: 60% dos *swarms* sociais apresentam até 42% de *seeders* contra 30% dos *swarms* tradicionais. A proporção média de *peers multi swarm* também é maior, sendo 78 na disseminação social e 16 na tradicional. Este indicador evidenciou na disseminação social um fenômeno de interseção de *swarms*, indicando evidências de formação de proto comunidades.

O agrupamento de *peers* por sistema autônomo e país também é maior: 50% dos *swarms* sociais tem até 89 *peers* por país e 22 *peers* por AS. Em disseminação tradicional essa proporção é de 29 *peers* por país e 10 por AS.

Em contrapartida foi observado que o conteúdo disseminado em redes sociais tem uma natureza muito mais restrita, indicando que as comunidades tendem a disseminar conteúdos de interesse específico. Desta forma, um indexador tradicional de arquivos *torrent* tem muito mais variedade de conteúdo do que uma comunidade social de *torrents*.

No que diz respeito a evolução das comunidades ao longo do tempo, foi observado que a disseminação social aumentou a duração das relações entre *peers*. Em 60% dos *swarms* sociais a permanência individual de *peers* quase dobrou: 128,4 minutos em *swarms* sociais contra 75,5 minutos em *swarms* tradicionais. A integridade de comunidades também sofreu uma melhora: 80% dos *swarms* sociais têm uma taxa de até 32% de preservação entre *snapshots* sucessivos contra 26% na disseminação tradicional. A duração média de comunidades também é 6% maior: 111 horas contra 105 horas. Por fim, o número de reencontros entre *peers* teve um aumento de mais de 700 mil ocorrências. Essas métricas em conjunto sugerem que a incidência de *freeriders* é atenuada e que, quanto maior for o número de *peers* sociais em um *swarm*, maiores serão os melhoramentos.

Em termos de redes complexas, foi observada uma significativa modificação nas métricas, indicando evidências de uma herança das características topológicas de grafos sociais. Houve um maior taxa de agrupamento entre nós advinda do aumento de uma ordem de grandeza na transitividade, bem como uma disseminação mais bem distribuída entre os nós das comunidades, devido a uma redução significativa de nós com centralidade zero e média de graus maior.

Como trabalho futuro, propõe-se a avaliação de uma massa de dados maior para a análise topológica com o objetivo de verificar se há melhoria nas métricas apresentadas e redução de *bias*. Também é importante que seja proposto um mecanismo que vise integrar um nó de uma rede social com o *swarms* de que ele participa. Desta forma testar algoritmos de escolha de nós orientados por relações sociais se torna uma tarefa mais fácil. O *Facebook* já suporta a criação de miniaplicativos em seu *site* e isso pode ser explorado para desenvolver uma interface entre o *Bittorrent* e a rede social.

REFERÊNCIAS

- [1] ADAR, E.; HUBERMAN, B. A. Free riding on gnutella. **First Monday**, v. 5, n. 10, 2000.
- [2] AKAY, A. C. **Distributed Algorithms for Improving BitTorrent Performance**. Dissertação (Mestrado) — KTH School of Electrical Engineering. Stockholm, Sweden, 2010.
- [3] ANDROUTSELLIS-THEOTOKIS, S.; SPINELLIS, D. A survey of peer-to-peer content distribution technologies. In: **Proc. of ACM Computing Surveys**, 2004. p. 335–371.
- [4] BASU, S.; BANERJEE, S.; SHARMA, P.; LEE, S. Nodewiz: peer-to-peer resource discovery for grids. In: **the Fifth IEEE international Symposium on Cluster Computing and the Grid**, 2005. p. 213–220.
- [5] BINDAL, R.; CAO, P.; CHAN, W.; MEDVED, J.; SUWALA, G.; BATES, T.; ZHANG, A. Improving traffic locality in bittorrent via biased neighbor selection. In: **Proc. of IEEE International Conference on Distributed Computing Systems**, 2006.
- [6] BLOND, S. L.; LEGOUT, A.; DABBOUS, W. Pushing bittorrent locality to the limit. In: **Proc. of Computer Networks: The International Journal of Computer and Telecommunications**, 2011. p. 541–557.
- [7] CARRIERO, N.; GELERNTER, D.; LEICHTER, J. Distributed data structures in linda. In: **Proc. of the 13th ACM SIGACT-SIGPLAN symposium on Principles of programming languages**, 1986. p. 236–242.
- [8] CHENG, X.; LIU, J. Netteube: Exploring social networks for peer-to-peer short video sharing. In: **Proc. IEEE INFOCOM**, 2009.
- [9] CLAUSET, A.; NEWMAN, M. E. J.; MOORE, C. Finding community structure in very large networks. In: **Phys. Rev. E** 70, 2004.
- [10] COHEN, B. Incentives build robustness in bittorrent. In: **Proc. of 1st Workshop on Economics of Peer-to-Peer Systems (P2PECON)**, 2003.
- [11] DAMIANI, E.; VIMERCATI, D. C. di; PARABOSCHI, S.; SAMARATI, P.; VIO-LANTE, F. A reputation-based approach for choosing reliable resources in peer-to-peer networks. In: **Proc. of 9th ACM conference on Computer and communications security**, 2002. p. 207–216.
- [12] DÁN, G.; CARLSSON, N. Dynamic swarm management for improved bittorrent performance. In: **Proc. IPTPS**, 2009.
- [13] DECKER, C.; EIDENBENZ, R.; WATTENHOFER, R. Exploring and improving bittorrent topologies. In: **Proc. of IEEE P2P**, 2013. p. 1–10.
- [14] FREEMAN, L. C. Centrality in social networks: Conceptual clarification. In: **Social Networks**, 1979. v. 1, p. 215–239.

- [15] GALUBA, W.; ABERER, K.; DESPOTOVIC, Z.; KELLERER, W. Leveraging social networks for increased bittorrent robustness. In: **Proc. of 7th IEEE conference on Consumer communications and networking conference**, 2010. p. 159–163.
- [16] GARCIA, F. D.; HOEPMAN, J. H. Off-line karma: A decentralized currency for static peer-to-peer and grid networks. In: **Proc. of 5th Int. Networking Conference**, 2005. p. 325–332.
- [17] GKANTSIDIS, C.; MIHAIL, M.; SABERI, A. Random walks in peer-to-peer networks: algorithms and evaluation. In: **Proc. of Performance Evaluation - P2P computing systems**, 2004. p. 335–371.
- [18] IZAL, M.; URVOY-KELLER, G.; BIRSACK, E. W.; FELBER, P. A.; AL-HAMRA, A.; ERICE, L. G. Dissecting bittorrent: Five months in a torrent’s lifetime. In: **Proc. of Passive and Active Network Measurement**, 2004. p. 1–11.
- [19] KAMVAR, S. D.; SCHLOSSER, M. T.; MOLINA, H. G. The eigentrust algorithm for reputation management in p2p networks. In: **Proc. of 12th international conference on World Wide Web**, 2003. p. 640–651.
- [20] KARAKAYA, M.; KORPEOGLU, I.; ULUSOY, O. Free riding in peer-to-peer networks. In: **Proc. of IEEE Internet Computing**, 2009. p. 92–98.
- [21] KARLSSON, M. **To Be or Not to Be a BitThief**. Dissertação (Mestrado) — Swiss Federal Institute of Technology (ETH) Zurich, 2010.
- [22] KAUNE, S.; TYSON, G.; PUSSEP, K.; MAUTHE, A.; STEINMETZ, R. The seeder promotion problem: Measurements, analysis and solution space. In: **Proc. IEEE ICCCN**, 2010.
- [23] LIU, J.; WANG, H.; XU, K. Understanding peer distribution in the global internet. In: **Proc. of IEEE Network**, 2010. p. 40–44.
- [24] LOCHER, T.; MOOR, P.; SCHMID, S.; WATTENHOFER, R. Free riding in bittorrent is cheap. In: **Proc. of HotNets**, 2006.
- [25] MAYMOUNKOV, P.; MAZIERES, D. Kademia: A peer-to-peer information system based on the xor metric. In: **Proc. of Revised Papers from the First International Workshop on Peer-to-Peer Systems**, 2002. p. 53–65.
- [26] NEWMAN, M. E. J. Fast algorithm for detecting community structure in networks. In: **Phys. Rev. E** **69**, 2004.
- [27] NEWMAN, M. E. J. Power laws, pareto distributions and zipf’s law. In: **Proc. of Contemporary Physics**, 2005. p. 323–351.
- [28] PIATEK, M.; ISDAL, T.; ANDERSON, T.; KRISHNAMURTHY, A.; VENKATARAMANI, A. Do incentives build robustness in bit torrent. In: **Proc. of the 4th USENIX conference on Networked systems design and implementation**, 2007.
- [29] PIATEK, M.; MADHYASTHA, H. V.; JOHN, J. P.; KRISHNAMURTHY, A.; ANDERSON, T. Pitfalls for isp-friendly p2p design. In: **Proc. HotNets**, 2009.

- [30] REN, S.; TAN, E.; LUO, T.; CHEN, S.; GUO, L.; ZHANG, X. Topbt: A topology-aware and infrastructure-independent bittorrent client. In: **Proc. of IEEE International Conference on Computer Communications**, 2010.
- [31] RIPEANU, M.; FOSTER, I. Mapping the gnutella network: macroscopic properties of large scale peer-to-peer systems. In: **Proc. of 1st international workshop on peer-to-peer systems**, 2002.
- [32] RISSON, J.; MOORS, T. Survey of research towards robust peer-to-peer networks: Search methods. In: **Proc. of Computer Networks: The International Journal of Computer and Telecommunications Networking**, 2006. p. 3485–3521.
- [33] SCHOLLMEIER, R. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In: **Proc. of the IEEE First International Conference on Peer-to-Peer Computing**, 2002.
- [34] SEEDORF, J.; KIESEL, S.; STIEMERLING, M. Traffic localization for p2p-applications: The alto approach. In: **Proc. IEEE Peer-to-Peer Computing**, 2009.
- [35] STOICA, I.; MORRIS, R.; LIBEN-NOWELL, D.; KARGER, D. R.; KAASHOEK, M. F.; DABEK, F.; BALAKRISHNAN, H. Chord: A scalable peer-to-peer lookup protocol for internet applications. In: **Proc. of IEEE/ACM Trans. Netw**, 2003. p. 17–32.
- [36] TIAN, C.; LIU, X.; JIANG, H.; LIU, W.; WANG, Y. Improving bittorrent traffic performance by exploiting geographic locality. In: **Proc. of IEEE Global Telecommunications Conference**, 2008.
- [37] TSOUMAKOS, D.; ROUSSOPOULOS, N. A comparison of peerto-peer search methods. In: **Proc. of WebDB**, 2003.
- [38] VARVELLO, M.; STEINER, M. Traffic localization for dht-based bittorrent networks. In: **Proc. IFIP NETWORKING**, 2011.
- [39] WANG, F.; LIU, J.; XU, K.; ; WU, D. Torrents on twitter: Explore long-term social relationships in peer-to-peer systems. In: **Proc. of IEEE P2P**, 2013. p. 1–10.
- [40] WANG, H.; LIU, J. Exploring peer-to-peer locality in multiple torrent environment. In: **Proc. of IEEE TPDS**, 2012. p. 1216–1226.
- [41] WANG, H.; WANG, F.; ; LIU, J. On long-term social relationships in peer-to-peer systems. In: **Proc. IEEE 19th International Workshop on Quality of Service**, 2011.
- [42] WASSERMAN, S.; FAUST, K. **Social Network Analysis: Methods and Applications**, 1994. ISBN 0521387078.
- [43] WU, D.; SEN, S. Y.; DHUNGEL, P.; HEI, X.; ZHANG, C.; ROSS, K. W. Understanding peer exchange in bittorrent systems. In: **Proc. of IEEE Tenth International Conference on Peer-to-Peer Computing**, 2006. p. 1–8.
- [44] YANG, B.; MOLINA, H. G. Ppay: Micropayments for peer-to-peer systems. In: **Proc. of 10th ACM Conf. Computer and Comm. Security**, 2003.

- [45] ZHANG, C.; DHUNGEL, P.; WU, D.; LIU, Z.; ROSS, K. W. Bittorrent darknets. In: **Proc. of the 29th conference on Information communications**, 2010. p. 1460–1468.

APÊNDICE A – INTERVALOS DE CONFIANÇA

Os cálculos de intervalo de confiança para as métricas desse trabalho são apresentadas a seguir. Para cada métrica foi calculado o limite inferior e superior segunda a fórmula:

$$\bar{x} \pm z_{1-\frac{\alpha}{2}} \left(\frac{s}{\sqrt{n}} \right)$$

Em casos em que os intervalos são disjuntos, o maior intervalo é considerado aquele cuja média é maior. Em casos onde os intervalos se sobrepõem e a média de cada um está contida no limite oposto, as amostras são consideradas iguais. De acordo com a tabela z o valor $Z_{1-\alpha/2}$ para um nível de confiança de 95% é 1,96.

Os cálculos são apresentados da seguinte forma: existe uma seção para cada métrica, onde são calculados os intervalos de confiança e média para cada disseminação.

A.1 *Peers por Swarm*

Tradicional

$$103,55883150945102 - 1,96 * (940,4637964753286/105,152270541344) = 86,028929096$$

$$103,55883150945102 + 1,96 * (940,4637964753286/105,152270541344) = 121,088733922$$

$$\bar{x} = 103,558831509$$

Social

$$321,98902821316614 - 1,96 * (1204,2189464709031/35,7211421989835) = 255,914177585$$

$$321,98902821316614 + 1,96 * (1204,2189464709031/35,7211421989835) = 388,063878842$$

$$\bar{x} = 321,989028214$$

A.2 *Arquivos por swarm*

Tradicional

$$63,84625074587284 - 1,96 * (421,13026443817483/122,81286577553672) = 57,125331784$$

$$63,84625074587284 + 1,96 * (421,13026443817483/122,81286577553672) = 70,567169707$$

$$\bar{x} = 63,846250746$$

Social

$$14,514888337468983 - 1,96 * (81,61446559414762/40,149719799769464) = 10,530692349$$

$$14,514888337468983 + 1,96 * (81,61446559414762/40,149719799769464) = 18,499084326$$

$$\bar{x} = 14,514888338$$

A.3 *Número de seeders*

Tradicional

$$39,0694 - 1,96 * (367,3387/74,65922581972036) = 29,4257981$$

$$39,0694 + 1,96 * (367,3387/74,65922581972036) = 48,7130019$$

$$\bar{x} = 39,0694$$

Social

$$137,3313 - 1,96 * (587,1977/40,149719799769464) = 108,665907122$$

$$137,3313 + 1,96 * (587,1977/40,149719799769464) = 165,996692878$$

$$\bar{x} = 137,3313$$

A.4 Número de *seeders* por *peer*

Tradicional

$$0,29251765158641607 - 1,96 * (0,28056459916343546/74,65922581972036) = 0,285152097$$

$$0,29251765158641607 + 1,96 * (0,28056459916343546/74,65922581972036) = 0,299883206$$

$$\bar{x} = 0,292517652$$

Social

$$0,39300330731941846 - 1,96 * (0,2720713613635628/35,7211421989835) = 0,378074897$$

$$0,39300330731941846 + 1,96 * (0,2720713613635628/35,7211421989835) = 0,407931718$$

$$\bar{x} = 0,393003308$$

A.5 *Multi swarm peers*

Tradicional

$$16,4654 - 1,96 * (70,0251/65,74952471311104) = 14,377944667$$

$$16,4654 + 1,96 * (70,0251/65,74952471311104) = 18,552855333$$

$$\bar{x} = 16,4654$$

Social

$$78,2316 - 1,96 * (250,9946/35,38361202590826) = 64,328287482$$

$$78,2316 + 1,96 * (250,9946/35,38361202590826) = 92,134912518$$

$$\bar{x} = 78,2316$$

A.6 *Peers DHT e PEX*

Tradicional

$$55,4478 - 1,96 * (440,0220/74,65922581972036) = 43,896071322$$

$$55,4478 + 1,96 * (440,0220/74,65922581972036) = 66,999528678$$

$$\bar{x} = 55,4478$$

Social

$$269,5901 - 1,96 * (1065,1774/35,7211421989835) = 211,144384788$$

$$269,5901 + 1,96 * (1065,1774/35,7211421989835) = 328,035815212$$

$$\bar{x} = 269,5901$$

A.7 Coeficiente de *Jaccard*

Tradicional

$$0,00041456384396667596 - 1,96 * (0,019085112288182774/3941,766735868575) = 0,000405074$$

$$0,00041456384396667596 + 1,96 * (0,019085112288182774/3941,766735868575) = 0,000424054$$

$$\bar{x} = 0,000414564$$

Social

$$0,002095524793857525 - 1,96 * (0,04017785500820464/902,6217369418931) = 0,002008281$$

$$0,002095524793857525 + 1,96 * (0,04017785500820464/902,6217369418931) = 0,002182769$$

$$\bar{x} = 0,002095525$$

A.8 Países por *swarm*

Tradicional

$$13,509687836383208 - 1,96 * (17,511106613694466/74,65922581972036) = 13,049975476$$

$$13,509687836383208 + 1,96 * (17,511106613694466/74,65922581972036) = 13,969400197$$

$$\bar{x} = 13,509687837$$

Social

$$30,144200626959247 - 1,96 * (31,09281115046927/35,7211421989835) = 28,43815468$$

$$30,144200626959247 + 1,96 * (31,09281115046927/35,7211421989835) = 31,850246574$$

$$\bar{x} = 30,144200627$$

A.9 *Peers* por país

Tradicional

$$5,365085056372256 - 1,96 * (41,550689327167476/274,4139209296788) = 5,068309474$$

$$5,365085056372256 + 1,96 * (41,550689327167476/274,4139209296788) = 5,661860638$$

$$\bar{x} = 5,365085056$$

Social

$$10,681624376039933 - 1,96 * (61,66188027035963/274,4139209296788) = 10,241204718$$

$$10,681624376039933 + 1,96 * (61,66188027035963/274,4139209296788) = 11,122044034$$

$$\bar{x} = 10,681624376$$

A.10 *Peer* por AS

Tradicional

$$2,4704921942360256 - 1,96 * (13,023856109768476/404,3921364220625) = 2,40736842$$

$$2,4704921942360256 + 1,96 * (13,023856109768476/404,3921364220625) = 2,533615969$$

$$\bar{x} = 2,470492195$$

Social

$$3,5431621794097863 - 1,96 * (13,81978682236809/340,5260636133452) = 3,463618247$$

$$3,5431621794097863 + 1,96 * (13,81978682236809/340,5260636133452) = 3,622706112$$

$$\bar{x} = 3,54316218$$

A.11 Coeficiente de *Jaccard* (topologia)

Tradicional

$$0,178530746124198 - 1,96 * (0,14781684286817284/227,02422778197044) = 0,177254578$$

$$0,178530746124198 + 1,96 * (0,14781684286817284/227,02422778197044) = 0,179806914$$

$$\bar{x} = 0,179806914$$

Social

$$0,20192861708887402 - 1,96 * (0,15757975315982287/230,00869548780108) = 0,200585814$$

$$0,20192861708887402 + 1,96 * (0,15757975315982287/230,00869548780108) = 0,20327142$$

$$\bar{x} = 0,201928617$$

A.12 Coeficiente de *Jaccard* médio por snapshot

Tradicional

$$0,22211189930479192 - 1,96 * (0,14518216546768767/95,04735661763561) = 0,219118054$$

$$0,22211189930479192 + 1,96 * (0,14518216546768767/95,04735661763561) = 0,225105744$$

$$\bar{x} = 0,222111899$$

Social

$$0,23317451613950965 - 1,96 * (0,12235152243018545/97,25739046468397) = 0,230708801$$

$$0,23317451613950965 + 1,96 * (0,12235152243018545/97,25739046468397) = 0,235640231$$

$$\bar{x} = 0,233174516$$

A.13 Tamanho das comunidades

Tradicional

$$69,6899 - 1,96 * (86,1462/227,99122790142607) = 68,949316454$$

$$69,6899 + 1,96 * (86,1462/227,99122790142607) = 70,430483546$$

$$\bar{x} = 69,6899$$

Social

$$75,8374 - 1,96 * (106,8150/230,3714392019983) = 74,928618075$$

$$75,8374 + 1,96 * (106,8150/230,3714392019983) = 76,746181925$$

$$\bar{x} = 75,8374$$

A.14 Quantidade de comunidades

Tradicional

$$5,576177033057099 - 1,96 * (2,924505082938589/104,7902667235846) = 5,52147701$$

$$5,576177033057099 + 1,96 * (2,924505082938589/104,7902667235846) = 5,630877056$$

$$\bar{x} = 5,576177033$$

Social

$$5,452467119781093 - 1,96 * (2,3477875568749105/106,4377752492037) = 5,409233751$$

$$5,452467119781093 + 1,96 * (2,3477875568749105/106,4377752492037) = 5,495700489$$

$$\bar{x} = 5,45246712$$

A.15 **Permanência de *peers***

Tradicional

$$26576,7906 - 1,96 * (65573,4731/509,3711809672785) = 26324,471638743$$

$$26576,7906 + 1,96 * (65573,4731/509,3711809672785) = 27444,758511927$$

$$\bar{x} = 26884,615075335$$

Social

$$28598,1045 - 1,96 * (65713,6300/483,69411821935563) = 28331,823166511$$

$$28598,1045 + 1,96 * (65713,6300/483,69411821935563) = 28864,385833489$$

$$\bar{x} = 28598,1045$$

A.16 **Grau médio**

Tradicional

$$2,972309593256778 - 1,96 * (11,310488083263094/1903,2819023991165) = 2,960662051$$

$$2,972309593256778 + 1,96 * (11,310488083263094/1903,2819023991165) = 2,983957136$$

$$\bar{x} = 2,972309594$$

Social

$$4,126138482053744 - 1,96 * (11,602893496381967/2006,1821951158872) = 4,114802686$$

$$4,126138482053744 + 1,96 * (11,602893496381967/2006,1821951158872) = 4,137474278$$

$$\bar{x} = 4,126138482$$

A.17 ***Betweenness***

Tradicional

$$0,01883259053118077 - 1,96 * (0,11141069445481144/1902,5369904419729) = 0,018717815$$

$$0,01883259053118077 + 1,96 * (0,11141069445481144/1902,5369904419729) = 0,018947366$$

$$\bar{x} = 0,018832591$$

Social

$$0,016941212782451413 - 1,96 * (0,10316051014461482/2005,8658978107185) = 0,016840411$$

$$0,016941212782451413 + 1,96 * (0,10316051014461482/2005,8658978107185) = 0,017042014$$

$$\bar{x} = 0,016941213$$

Tradicional levemente maior devido a alguns poucos nós de alta centralidade

A.18 *Closeness*

Tradicional

$$0,4376725917007871 - 1,96 * (0,09517354990545734/1903,2819023991165) = 0,437574582$$

$$0,4376725917007871 + 1,96 * (0,09517354990545734/1903,2819023991165) = 0,437770601$$

$$\bar{x} = 0,437672592$$

Social

$$0,4172666459265785 - 1,96 * (0,09687818199721727/2006,1821951158872) = 0,417171998$$

$$0,4172666459265785 + 1,96 * (0,09687818199721727/2006,1821951158872) = 0,417361294$$

$$\bar{x} = 0,417266646$$

Tradicional levemente maior devido a alguns poucos nós de alta centralidade

A.19 **Transitividade**

Tradicional

$$0,004920588900904477 - 1,96 * (0,021825802193855354/229,7389823255949) = 0,004734384$$

$$0,004920588900904477 + 1,96 * (0,021825802193855354/229,7389823255949) = 0,005106794$$

$$\bar{x} = 0,004920589$$

Social

$$0,007395828550055586 - 1,96 * (0,020735185337534295/230,57536728800847) = 0,00721957$$

$$0,007395828550055586 + 1,96 * (0,020735185337534295/230,57536728800847) = 0,007572087$$

$$\bar{x} = 0,007395829$$