UNIVERSIDADE FEDERAL DE ITAJUBÁ
INSTITUTO DE MATEMÁTICA E COMPUTAÇÃO

DEPARTAMENTO DE MATEMÁTICA E COMPUTAÇÃO

SEMINÁRIO

Arquitetura Peer to Peer

REDES DE COMPUTADORES

Alunos:

David Mateus Batista
Gabriel Erzinger Dousseau
Gabriel Augusto Alves Taets
Maurício Leite Ferreira

Professor:

Bruno Guazzelli Batista

1 de maio de 2017

Resumo

Esta monografia tem o objetivo de estudar e analisar a arquitetura de redes Peer-to-peer, suas vantagens, desvantagens, limitações e aplicações.

Sumário

1. Introdução

2. Fundamentação Teórica

Peer-to-Peer significa, em tradução literal, "par a par". Em termos formais, os dispositivos da rede estão conectados por uma cadeia descentralizada, onde cada um possui funções equivalentes, sem hierarquias. Todos os usuários são clientes e servidores, funcionando de forma independente e livre da existência de um servidor central. A busca por um arquivo é enviada a todos os dispositivos da rede, os quais tomam conhecimentos dos outros usuários através de um hospedeiro (que apenas monitora pontos de presença na rede e não tem um banco de dados com nomes de usuários ou arquivos), que informa o IP dos usuários ativos na rede. O pedido é repassado para todos os usuários, que vão passando, usuário por usuário, até que o arquivo desejado seja encontrado. [9]

É preciso notar que P2P e arquiteturas centralizadas não são alternativas disjuntas. Existem muitos sistemas que podem ser classificados como uma mistura de um sistema P2P e um sistema centralizado. O problema é que não existe uma fronteira clara entre um paradigma P2P e outros paradigmas supostamente opostos como cliente-servidor. Nos extremos, algumas arquiteturas são claramente P2P enquanto outras são claramente cliente-servidor. No entanto, existem arquiteturas que podem ser consideradas uma ou outra ou ambos, dependendo da definição de P2P considerada. Consequentemente, é importante entender o que é comum a todas as definições de P2P e o que são traços não-comuns que alguns autores incluem em suas próprias definições. [2]

Considera-se um sistema P2P se os elementos que formam o sistema compartilham seus recursos com o propósito de prover o serviço que o sistema foi desenhado para prover. Os elementos no sistema tanto proveem os serviços aos outros elementos, como requisitam os serviços aos outros elementos. A princípio, todos os elementos no sistema devem satisfazer este critério para que o sistema seja considerado P2P. No entanto, na prática, o sistema pode ter algumas exceções (isto é, alguns nós que não satisfazem este critério) e ainda ser considerado P2P. Por exemplo, um sistema P2P pode ainda ser considerado P2P mesmo se tiver um servidor centralizado de registros. Por outro lado, alguns sistemas dividem *endpoints* entre pares e clientes. Pares requisitam e oferecem serviços, enquanto clientes geralmente apenas requisitam serviços. Um sistema onde a maioria dos *endpoints* se comportam como clientes não pode ser considerado estritamente P2P. [2]

Alguns autores adicionam ainda que os elementos que formam os sistemas P2P (que não surpreendentemente são denominados *pares*), devem ser capazes de se comunicarem diretamente, sem passar por intermediários [8]. Outros autores dizem que o sistema deve ser auto-organizável e ter controle decentralizado [7].

É notável que as definições acima são dadas no contexto de um único serviço individual. Um serviço complexo pode ser composto de vários serviços individuais. Alguns

desses serviços individuais pode consistir de serviços P2P e alguns deles podem consistir de serviços cliente-servidor. Por exemplo, um cliente de compartilhamento de arquivos pode incluir um cliente P2P para efetuar o compartilhamento em si, e um navegador *web* para acessar informações adicionais num servidor *web* centralizado. Adicionalmente, existem arquiteturas onde um sistema cliente-servidor pode servir como "estepe"para um serviço normalmente provido por um sistema P2P, ou vice-versa. [2]

Prover um serviço geralmente envolve processamento ou armazenamento de dados. De acordo com a definição de Camarillo, num sistema P2P, os pares compartilham suas capacidades de processamento e armazenamento (isto é, seus recursos de software e hardware), de forma que o sistema possa prover o serviço. Por exemplo, se o serviço a ser provido é um serivço de distribuição de arquivos, pares diferentes entre o sistema irão armazenar arquivos diferentes. Quando um certo par quer um arquivo em particular, primeiro ele deve descobrir qual par possui (ou quais pares possuem) o arquivo, e então obter o arquivo destes pares [2]. Esta definição de P2P nos fornece um critério para decidir se um sistema é ou não P2P.

Figura 1. Esquema Cliente-Servidor [9]

A imagem acima representa um sistema cliente-servidor, onde o nó central é o servidor, e ele responde as requisições de todos os clientes.

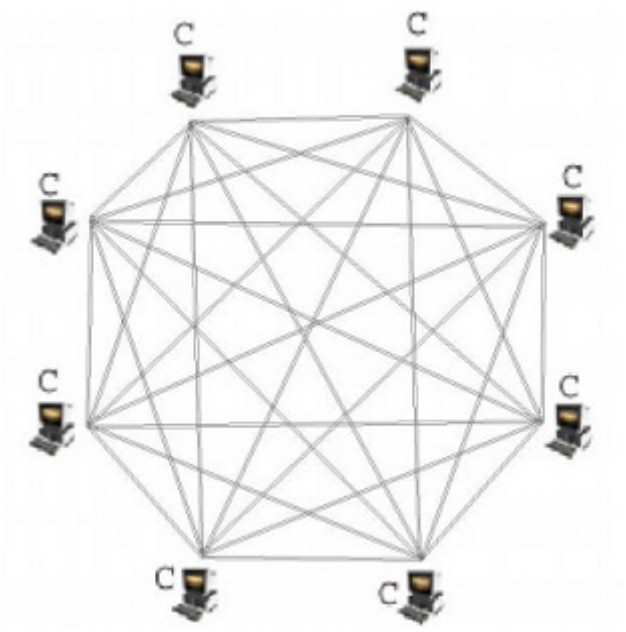

Figura 2. Esquema P2P [9]

A imagem acima representa um sistema P2P, onde todos os nós são ligados a todos os nós, e as requisições podem ser feitas e respondidas por qualquer nó.

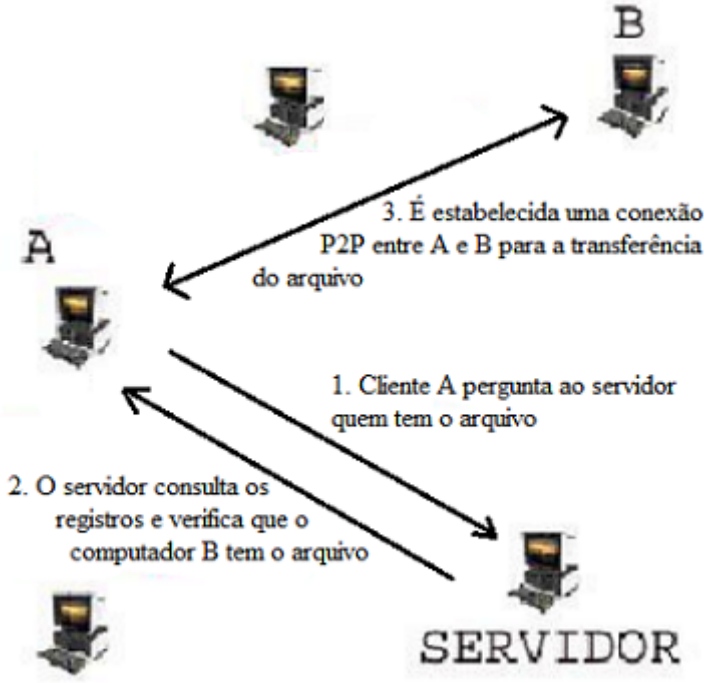

Figura 3. Esquema Misto [9]

A imagem acima representa um esquema misto, onde existe um servidor centralizado que recebe as requisições e direciona a requisição para um nó que possa respondê-la.

3. Aplicações

Como visto nas seções anteriores, a utilização da arquitetura **P2P** pode trazer diversas vantagens, desvantagens e outras características únicas para o aplicativo (ou serviço) que estará implementando-a.

Dessa forma, é importante analisar quais serviços utilizam essa arquitetura, como se beneficiam de suas vantagens e como lidam com suas desvantagens. Isto é, dissecar sobre como as características da arquitetura P2P são tratadas em cada caso de uso.

3.1. Napster

O **Napster** foi uma das primeiras aplicações de compartilhamento de arquivos, elaborada em Junho de 1999. O serviço permitia apenas o compartilhamento de arquivos **MP3** e é um grande responsável pela popularidade do termo "P2P".

A rede se baseia num servidor central de índices. Os nós da rede registram uma lista dos arquivos que desejam compartilhar. Dessa forma, a busca é processada baseando-se em **palavras chave** e retornará uma lista com os arquivos encontrados e informações como a banda disponível, tamanho e fonte.

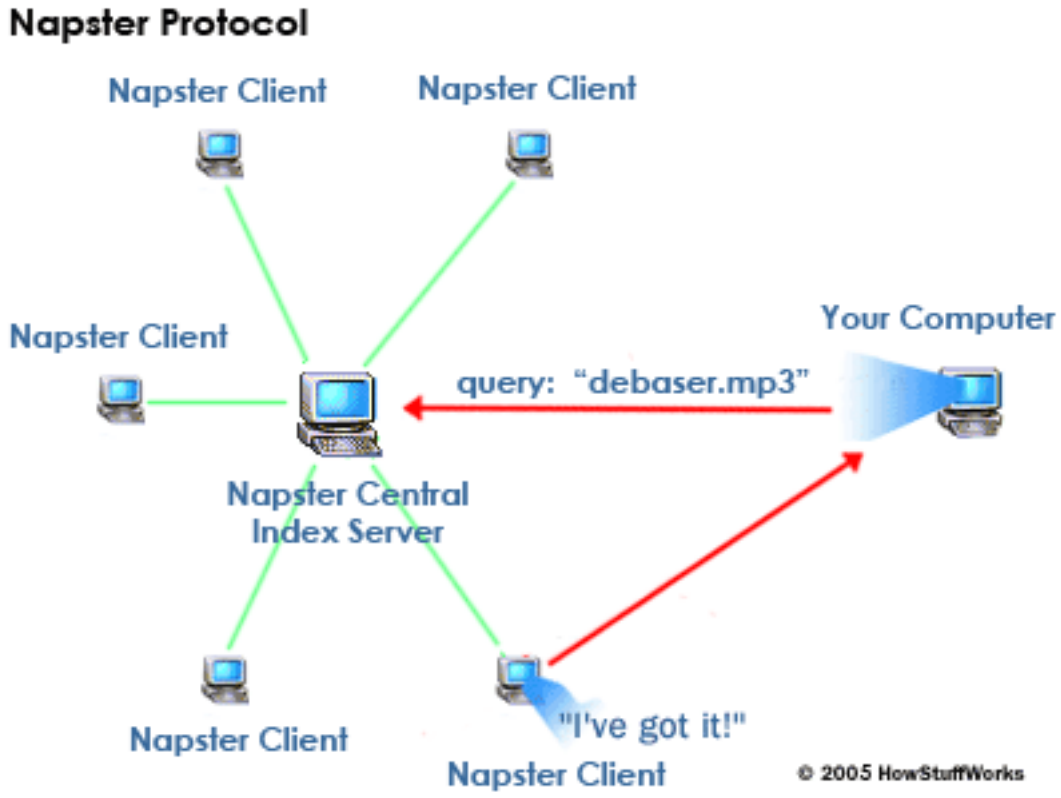

Figura 4. Protocolo Napster [11]

A imagem acima representa o funcionamento do protocolo Napster, indicando um cliente fazendo uma **consulta** ao servidor central, para que o servidor então busque

os outros clientes que possuem tal arquivo. Em seguida, tais clientes se conectam ao requisitante, para transferir o arquivo.

3.1.1. Vantagens e Desvantagens

Como um dos pioneiros na área, a principal vantagem do Napster se devia ao fato de ele possuir uma busca rápida e eficiente (por se tratar apenas de uma consulta ao servidor central). Além disso, oferecia uma visão simples e consistente da rede.

As desvantagens se devem exatamente a **presença de um servidor**, uma vez que, assim como em arquiteturas cliente-servidor, a central representa um ponto de falha único, que se fosse afetado, prejudicaria toda a rede. Por se tratar de um servidor que lida com diversas requisições, o servidor central também possuía um alto custo de manutenção.

3.2. Gnutella

Gnutella é um **protocolo de busca** aberto e descentralizado usado principalmente para o compartilhamento e a busca de arquivos. Foi criado com o objetivo de ser uma rede dinamica que permite que seus usuários entrem e saiam a qualquer momento, tenha uma boa escalabilidade, garanta a anonimidade e confiança em relação a ataques externos. [10]

O termo Gnutella se refere a todo o grupo de computadores que possuem aplicativos carregados com o protocolo Gnutella que formam uma espécie de rede virtual. Cada nó nessa rede pode funcionar tanto como um cliente como quanto um servidor.

Dessa forma, eles podem criar e receber requisições com outros nós utilizando os dados que estarão em seu disco rígido. Como é sabido, tais requisições não são enviados para nenhum servidor central, elas são exclusivamente tratadas entre os nós da rede.

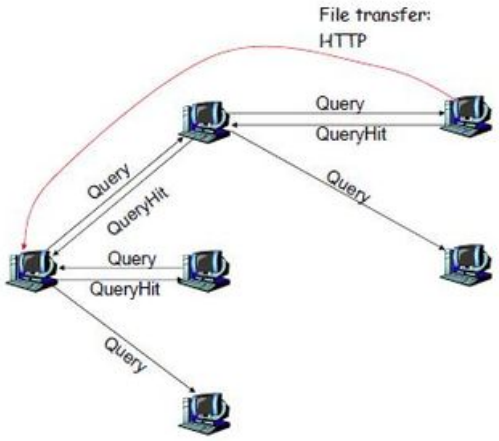

Figura 5. Protocolo Gnutella [3]

A imagem acima, representa um grupo de computadores utilizando aplicativos

com o protocolo Gnutella. Na mesma, um computador realiza uma **query** buscando um arquivo nos nós da rede. Os computadores que possuírem tal arquivo, retornam um **Query hit** para que, em seguida, seja estabelecida uma conexão **HTTP** entre eles para a transmissão do arquivo.

3.2.1. Mecanismo de Busca

O protocolo utiliza o algoritmo padrão conhecido como BFS(**B**readth **F**irst **S**earch) - Busca em Largura - portanto, tal qual o algoritmo é utilizado em grafos, ele será utilizado na rede de aplicativos que utilizam o protocolo: [10]

- O nó inicialmente busca um arquivo e envia a mensagem de busca a seus vizinhos
- Os vizinhos encaminham a mensagem para todos os seus vizinhos
- Os Nós que possuírem o arquivo que está sendo requisitado, começam uma mensagem de resposta item. Após a mensagem de resposta atingir o nó origem, o download do arquivo começa.

3.2.2. Vantagens e Desvantagens

O protocolo GNutella apresenta algumas modificações que **superam** as desvantagens do Napster, como por exemplo: não depender de um servidor para indexar os arquivos, dessa forma, não se cria um "único ponto de falha"na arquitetura. Entretanto, este mesmo fato causa uma das suas principais desvantagens: a busca em largura acaba por adicionar um certo período de tempo bem maior do que se fosse utilizado um servidor.

Além disso, outros pontos de falha do protocolo Gnutella são: o fato de que metade dos arquivos são servidos por apenas 1 porcento dos usuário , além da perca de banda larga e segurança em relação a transmissão de arquivos mascarados. [10]

3.3. BitTorrent

Apesar de muitos sistemas de compartilhamento de arquivos terem sido propostos e implementados, poucos persistirão ao teste intensivo de uma grande quantidade de usuários diários. O **BitTorrent** é um destes sistemas que, além de persistir, evoluiu como uma das mais populares redes da internet. De fato, a pesquisa **The True Picture of Peer-to-Peer File sharing** indica que, em Junho de 2004, o sistema era responsável por mais da metade de todo o tráfico P2P do mundo. [6]. Dessa forma, nesta monografia, citamos o sistema BitTorrent como representante das demais aplicações de torrents que funcionam de forma parecida, porém que não alcancaram o a mesma difusão e boa performance.

Diferente do que visto anteriormente, o BitTorrent é apenas um protocolo de

download de arquivos, de forma que dependa de outros serviços ou websites para fazer a busca destes arquivos [5, 6].

3.3.1. Funcionamento

O sistema funciona dividindo um arquivo em partes menores, dessa forma, ao se executar o download do arquivo, o usúario ao mesmo tempo realiza o upload de uma **parte** do arquivo que ele já possui, para outros usúarios que ainda não possuem esta parte. Todo este funcionamento é gerenciado de forma a otimizar a taxa de download e upload.

Um usuário com uma alta taxa de upload, tem uma chance maior de possuir também uma alta taxa de download, o que funciona como um incentivo para que os usúarios contribuam com a rede. Quando um usúario finaliza o download de um arquivo, ele permanece como uma semente, executando apenas o upload deste arquivo, para colaborar com a rede. [5]

Para se publicar um arquivo utilizando tal sistema, primeiro um arquivo estatico $.torrent$ é adicionado a qualquer servidor web para que os usuários o encontrem. Este arquivo contem informações como o tamanho, nome, hash e a url de um $tracker$. Estes trackers são responsáveis por fazer com que os usúarios da rede se "encontrem". Para dar inicio, basta então, que um usúario que contenha o arquivo em sua totalidade inicie o arquivo $.torrent$ utilizando o BitTorrent para que ele se torne uma $semente$. Dessa forma, outros usúarios que abrirem o arquivo .torrent irão receber partes do arquivo total, semeadas pelo primeiro usúario. A medida que forem executando o download, eles também irão semear partes para outros usuários. [5]

A maior parte dos problemas de sistemas de compartilhamento P2P reside na interação entre peers. Para ter conhecimento de qual usúario tem o que, o BitTorrent particiona o arquivo em pedaços de tamanho fixo, usualmente de $\frac{1}{4}$ de megabyte. Cada usuário então reporta para seus pares qual parte do arquivo ele possui. Para garantir a integridade do dado, o arquivo $.torrent$ possui a função criptográfica hash de todas as partes e um peer só reporta posuir uma parte após assegurar que possui o hash correto da mesma. [5]

Em seguida, é necessário então, entender o como o BitTorrent seleciona em qual ordem as partes do arquivo devem ser requisitadas para a rede, de forma a otimizar a performance do sistema. Para fazer tal escolha, existem diversos algoritmos utilizados no sistema. Os algoritmos podem por exemplo, buscar primeiro completar partes maiores do arquivo, buscar primeiro as partes mais raras do arquivo (esse método é útil pois estas partes possivelmente irão ter uma velocidade de download menor, portanto, começando antes, elas tendem a terminar junto com o arquivo como um todo) e, para decidir qual

parte do arquivo irá começar o download, o algoritmo realiza uma escolha aleatória, uma vez que a escolha da primeira parte não irá impactar tanto no processo como um todo. [5]

3.3.2. Sistema de Busca

A busca dos arquivos $.torrent$ é realizada de forma separada do sistema de download, não estando relacionada necessariamente ao BitTorrent, porém, como é a base para muitos outros sistemas P2P, é importante ser discutida.

Ela ocorre em sites que indexam os arquivos a partir de seus nomes, tamanho e número de pares conectados. Tais sites fornecem **links magnéticos** para a aplicação utilizada (BitTorrent por exemplo) se conectar com os pares que possuem as partes do arquivo.

3.3.3. Vantagens e Desvantagens

Como observado, uma das maiores vantagens do BitTorrent associado com um site de bsuca de arquivos .torrent é o alto nível da integridade do arquivo, devido ao fato de que todas as partes são averiguadas através do hash [5].

Além disso, a velocidade de download por sistemas como o BitTorrent costuma ser bem maior para um grande número de usúarios, devido a auto-escalabilidade padrão dos sistemas P2P: de acordo com o aumento do número de usúarios na rede, o número de sementes aumenta de forma proporcional, pois mais usúarios possuirão partes de arquivos diferentes para compartilhar. [6]

Surge também uma problemática relacionada aos sites que armazenam os arquivos $.torrent$, uma vez que grande parte destes sites são responsáveis pela pirataria de diversos programas, músicas, filmes e outras mídias, aproveitando-se da $semi - anonimidade$ oferecida pelos sistemas de torrent. Entretanto, não é uma desvantagem que concerne ao sistema BitTorrent em sí mas que o acaba afetando.

Os maiores desafios para aplicações como o BitTorrent continuam sendo os mesmos nos ultimos tempos: aumentar o incentivo para que usuários se comportem como sementes após a finalização do download do arquivo para que a rede se torne maior e mais consistente, e otimizar a largura de banda dos trackers, para que os pares se encontrem mais rapidamente.

4. Discussão

Na seção anterior foi explicado sobre alguns exemplos de aplicações que utilizam a arquitetura P2P, e suas vantagens e desvantagens particulares, faremos agora uma analise mais geral sobre a arquitetura em sí, com suas principais vantagens e desvantagens.

4.1. Vantagens

As vantagens obtidas ao se utilizar uma arquitetura P2P abrangem desde não existe a necessidade de um servidor dedicado, o custo total para se construir e fazer a manutenção necessária para este tipo de rede é bem menor, até a existência de uma dependência central, a arquitetura P2P é mais confiável, pois a queda de um par não afeta diretamente os outros pares.

4.1.1. Auto-Escalabilidade

A rede cresce de acordo com o aumento dos provedores e consumidores inseridos nela, um alto numero de somente provedores ou consumidores na rede não traz vantagem nenhuma a mesma, visto que com um alto numero somente de consumidores haverá um gargalo pelo qual os provedores não serão capazes de atender, já com um aumento somente nos provedores a rede haverá um aumento no custo para um esforço aparentemente sem utilidade. [4]

4.1.2. Auto-Organização

Com a inserção de um novo par na rede, requer uma mínima ou quase nenhuma configuração para ela começar a se comunicar com os outros pares já presentes na rede, [4]

4.1.3. Tolerância a falhas

Servidores centrais são alvos de ataques de hackers com uma alta frequência, como eles aderem a um modelo em que todos os dados estão centralizados em um centro de dados central. No caso de um evento (cyberataque ou desastre natural) que traga o servidor a ter um mal funcionamento, os usuários não teriam mais acesso aos dados dos provedores. Já, observando esses eventos em uma rede P2P, após o "provedor"presente na rede cair (sendo o provedor um dos pares presentes na rede), os dados buscados na rede continuariam sendo acessíveis a partir de outro par, visto que nenhum nó na rede é crítico para que ela continue operando. [4]

4.2. Desvantagens

Observando as vantagens apresentadas anteriormente, temos algumas consequências que podem afetar tanto a segurança do usuário, como a do dado presente na rede.

4.2.1. Integridade dos Dados

Através de um modelo P2P, existem inúmeras ocasiões em que os dados são filtrados, já que não existe um administrador central para essa arquitetura. Isso ocorre também com dados nocivos como modelos 3D de munições e armas. O mais comum é isso ocorrer com dados que violam os direitos autorias de multimídia. Esses exemplos abrangem os casos em que os dados realmente contém as informações que são marcados. Um outro problema é quando os dados são intencionalmente mal rotulados por um dos pares, por exemplo um dos pares substitui os dados que um consumidor estava buscando por um vírus, mas mantém o rótulo dos dados buscados, causando assim o descarregamento de um vírus no consumidor.

4.2.2. Desaparecimento da Rede

Como a transmissão de arquivos ficam somente para os pares na rede, quando um par é o único na rede a possuir um arquivo e ele não está mais presente na mesma, existe uma defasagem do arquivo necessitado, e caso o par não volte a rede, o arquivo em questão pode não ser mais encontrado quando solicitado futuramente.

4.3. Exploit no BitTorrent

Um dos mais populares protocolos P2P para compartilhamento de arquivos possui um exploit no qual se pode lançar um ataque DRDoS, visto que o BitTorrent faz uso do protocolo UDP, o qual não inclui mecanismos para evitar a falsificação do endereço de origem IP. [1]

5. Considerações finais

Referências

[1] Florian Adamsky, Syed Ali Khayam, Rudolf Jäger, and Muttukrishnan Rajarajan. P2p file-sharing in hell: Exploiting bittorrent vulnerabilities to launch distributed reflective dos attacks.

[2] Gonzalo Camarillo. Peer-to-Peer (P2P) Architecture: Definition, Taxonomies, Examples, and Applicability. *Internet Architecture Board*, 2009.

[3] Andrew T. Campbell. *Examples Peer-to-Peer Applications*. Departament of Computer Science - Darthmouth College.

[4] Adam Cohen. Is peer-to-peer inherently bad? 2015.

[5] Bram Cohen. Incentives build robustness in bittorrent. *Workshop on Economics of Peer-to-Peer Systems, Berkeley, USA*, 2003.

[6] J.A Powelse, P. Garbacki, and H.J. Sips D.H.J Epema. The bittorrent p2p file-sharing system: measurements and analysis.

[7] Mema Roussopoulos, Mary Baker, Giuli TJ, Petros Maniatis, David Rosenthal, and Jeff Mogul. Workshop on Peer-to-Peer Systems. 2004.

[8] Rüdiger Schollmeier. First International Conference on Peer-to-Peer Computing P2P '01. 2001.

[9] Alberto Scremin, Barbara Herrera, Bianca Paixão, and Daniel Tamaki. *Sistemas de redes peer to peer*. Universidade Federal Fluminense, 2007.

[10] Gayatri Tribhuvan. A brief introduction and analysis of the gnutella protocol.

[11] Jeff Tyson. *How the old Napster Worked*. How Stuff Works.