

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
Mauricio Ferreira Leite

Professor:

Bruno Guazzelli Batista

25 de abril 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	1
2	Fundamentação Teórica	2
3	Aplicações	5
3.1	Napster	5
3.1.1	Vantagens e Desvantagens	6
3.2	Gnutella	6
3.2.1	Mecanismo de Busca	7
3.2.2	Vantagens e Desvantagens	7
4	Discussão	8
5	Considerações finais	9
	Referências	10

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. [4]

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. [1]

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 provêm 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. [1]

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 [3]. Outros autores dizem que o sistema deve ser auto-organizável e ter controle descentralizado [2].

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. [1]

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 serviço 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 [1]. Esta definição de P2P nos fornece um critério para decidir se um sistema é ou não P2P.

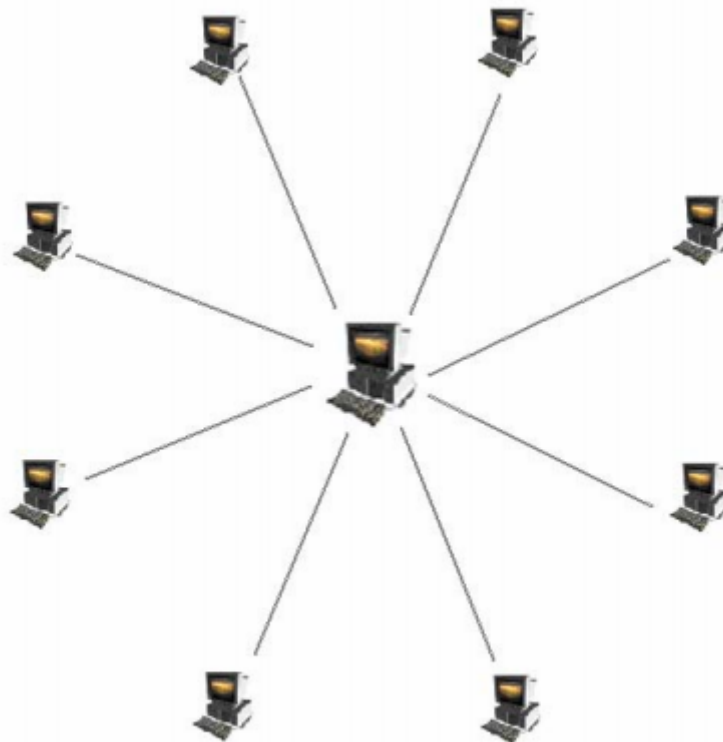


Figura 1. Esquema Cliente-Servidor

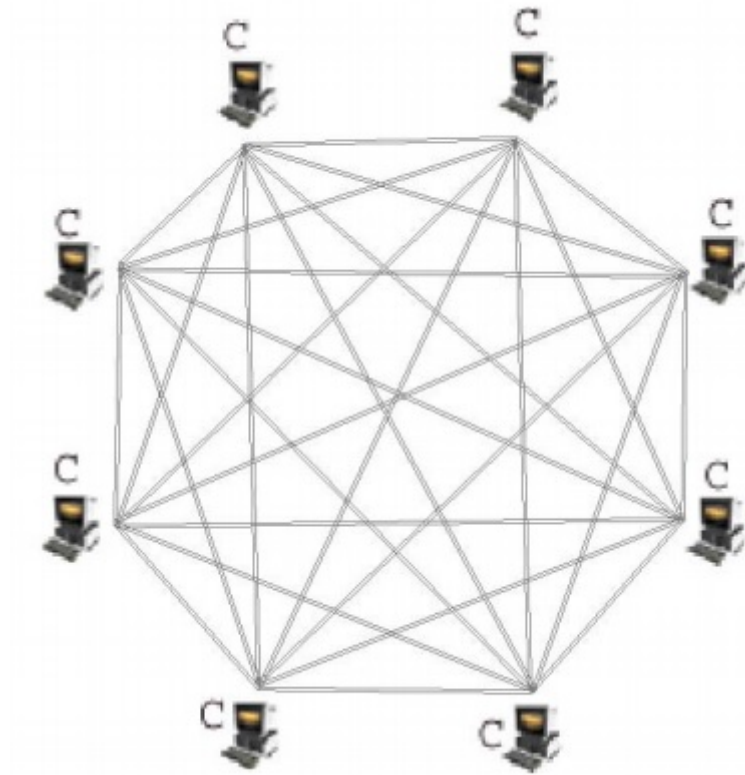


Figura 2. Esquema P2P

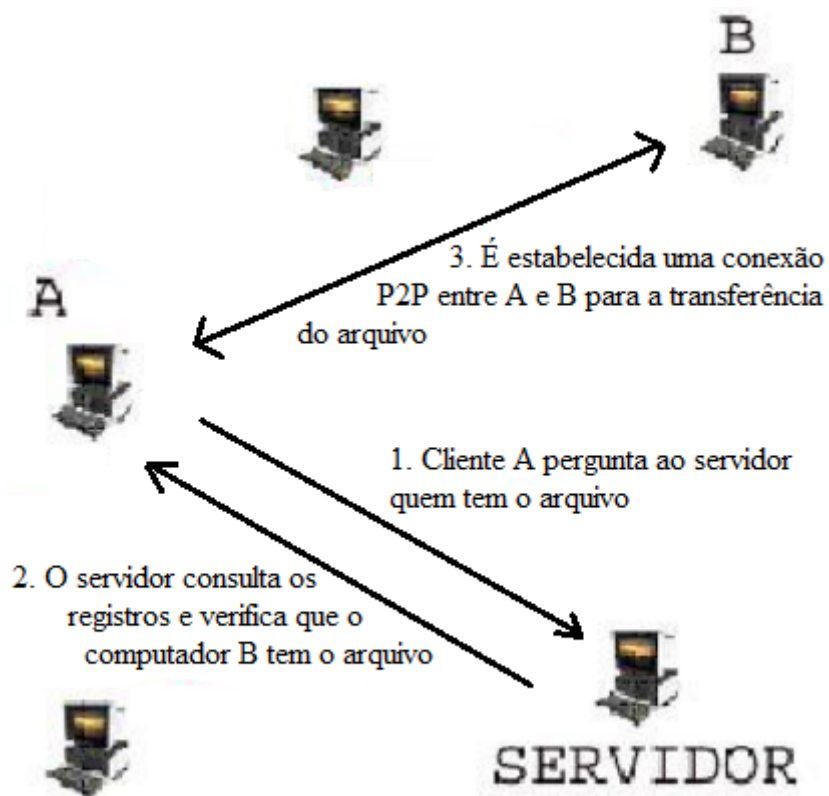


Figura 3. Esquema Misto

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.

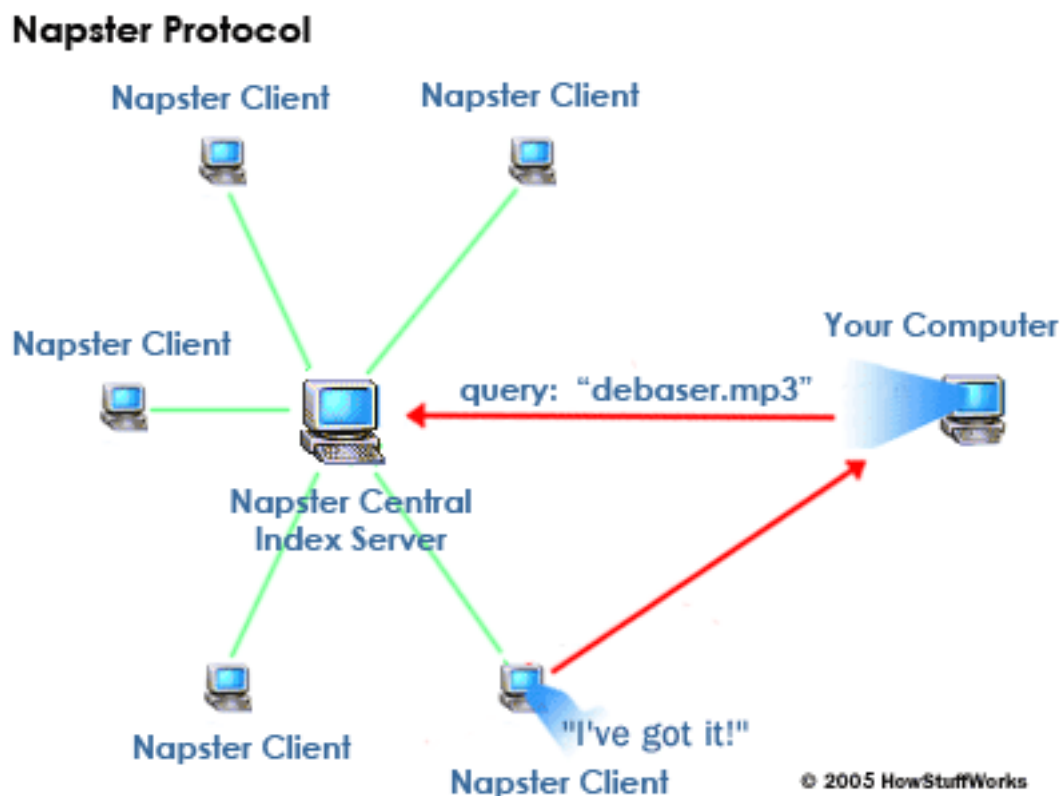


Figura 4. Protocolo Napster [?]

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. 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.

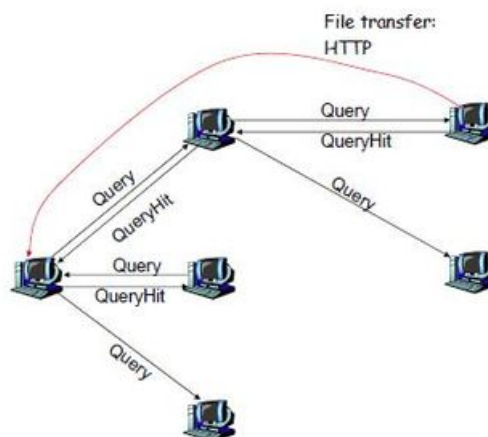


Figura 5. Protocolo Gnutella [?]

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:

- 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 periodo de tempo bem maior do que se fosse utilizado um servidor. As outras desvantagens do Gnutella possuem origem nas desvantagens da própria arquitetura P2P : segurança, poucos usuários servindo arquivos e perda de banda.

4. Discussão

5. Considerações finais

Referências

- [1] Gonzalo Camarillo. Peer-to-Peer (P2P) Architecture: Definition, Taxonomies, Examples, and Applicability. *Internet Architecture Board*, 2009.
- [2] Mema Roussopoulos, Mary Baker, Giuli TJ, Petros Maniatis, David Rosenthal, and Jeff Mogul. Workshop on Peer-to-Peer Systems. 2004.
- [3] Rüdiger Schollmeier. First International Conference on Peer-to-Peer Computing P2P '01. 2001.
- [4] Alberto Scremin, Barbara Herrera, Bianca Paixão, and Daniel Tamaki. *Sistemas de redes peer to peer*. Universidade Federal Fluminense, 2007.