Sistemas Distribuídos

Profª Vanice Canuto Cunha

Instituto de Computação - IC

Sistemas P2P

- Os sistemas *peer-to-peer –p2p* representam um paradigma para a construção de sistemas e de aplicativos distribuídos em que dados e recursos computacionais são provenientes da colaboração de **muitas máquinas** na Internet de maneira uniforme.

Objetivo

- Permitir o compartilhamento de dados e recursos em uma escala muito grande, eliminando qualquer exigência de servidores gerenciados separadamente e sua infraestrutura associada.
- Suportar serviços e aplicativos distribuídos, usando dados e recursos computacionais disponíveis nos computadores pessoais e estações de trabalho que estão presentes em números cada vez maiores na Internet e em outras redes. Isso é cada vez mais atraente, à medida que diminui a diferença de desempenho entre as máquinas desktop e servidores e que as conexões a redes de banda larga proliferam.

Características

- Garante que cada usuário contribua com recursos para o sistema.
- Todos os nós em um sistema *peer-to-peer* têm as mesmas capacidades e **responsabilidades funcionais**.
- Seu correto funcionamento não depende da existência de quaisquer sistemas administrados de forma centralizada.
- Podem ser projetados de modo a oferecer um grau limitado de anonimato para os provedores e usuários dos recursos.
- Um problema importante para seu funcionamento eficiente é a escolha de um **algoritmo** para a distribuição dos dados em muitas máquinas e o subsequente acesso a eles, de uma maneira que equilibre a carga de trabalho e garanta a disponibilidade sem adicionar sobrecargas indevidas.

Gerações de Sistemas P2P

- Primeira geração foi lançada pelo serviço de troca de músicas Napster [OpenNap 2001].
- Segunda geração de aplicativos de compartilhamento de arquivos, oferecendo maior escalabilidade, anonimato e tolerância a falhas surgiu logo depois, incluindo Freenet [Clarke et al. 2000, freenetproject.org], **Gnutella, Kazaa** [Leibowitz et al. 2003] e **BitTorrent** [Cohen 2003].
- A terceira geração é caracterizada pelo aparecimento de camadas de middleware para o gerenciamento de recursos distribuídos em uma escala global independente de aplicativos.

Geração atual

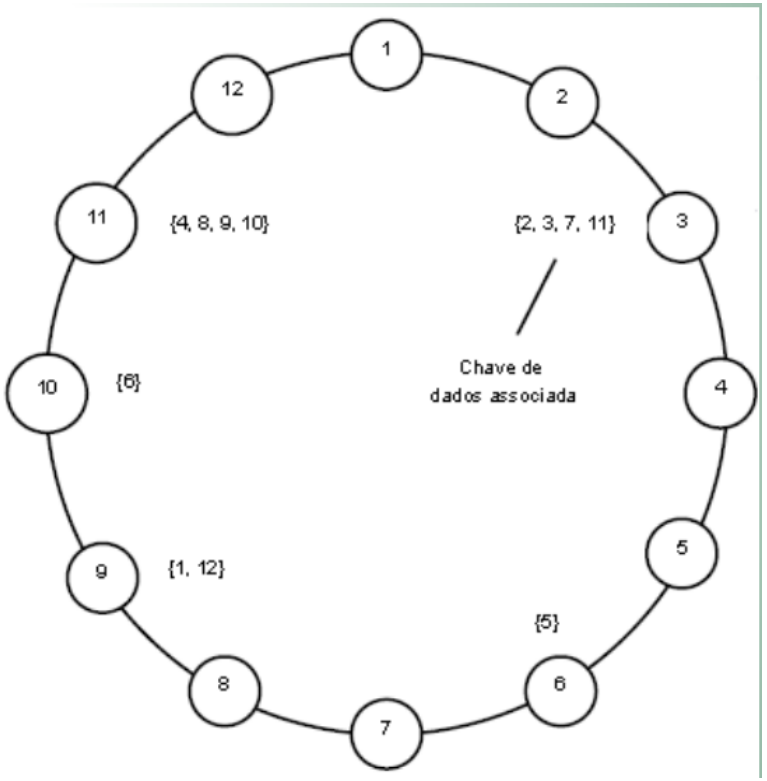
- Os exemplos mais conhecidos e totalmente desenvolvidos incluem o **Pastry** [Rowstron e Druschel 2001], o Tapestry [Zhao et al. 2004], o **CAN** [Ratnasamy et al. 2001], o **Chord** [Stoica et al. 2001] e o **Kademlia** [Maymounkov e Mazieres 2002].
- Vantagens:
  - Garantias do envio de pedidos em um número limitado de passos intermediário de rede.
  - Colocam réplicas dos recursos de maneira estruturada nos computadores disponíveis, levando em conta sua disponibilidade volátil, sua confiabilidade e seus requisitos de equilíbrio de carga variáveis, localização do armazenamento e uso das informações.

P2P estruturada

- Nas estratégias estruturadas, existe uma política global generalizada governando a topologia da rede, o posicionamento de objetos na rede e as funções de roteamento ou pesquisa usadas para localizar objetos na rede.
- Isso pode ser visto claramente nos exemplos do Pastry e do Tapestry, baseados na tabela de resumo distribuída (*hash table*) subjacente e nas estruturas em anel associadas.

P2P Estruturadas – Sistema Chord

- Sistema Chord (Soica et al,2003)
- Nós estão logicamente organizados em um anel
- Item de dado com chave k é mapeado para o nó com LOOKUP(k)



P2P não estruturada

- Nas estratégias não estruturadas, não há controle global sobre a topologia ou sobre o posicionamento de objetos dentro da rede.
- A sobreposição (a rede) é criada de maneira ad hoc, com cada nó que ingressa na rede seguindo algumas regras locais simples para estabelecer a conectividade.
- Ao ingressar, um nó estabelece contato com um conjunto de vizinhos, sabendo que os vizinhos também vão se conectar a outros e assim por diante, formando uma rede fundamentalmente descentralizada e que organiza a si própria;

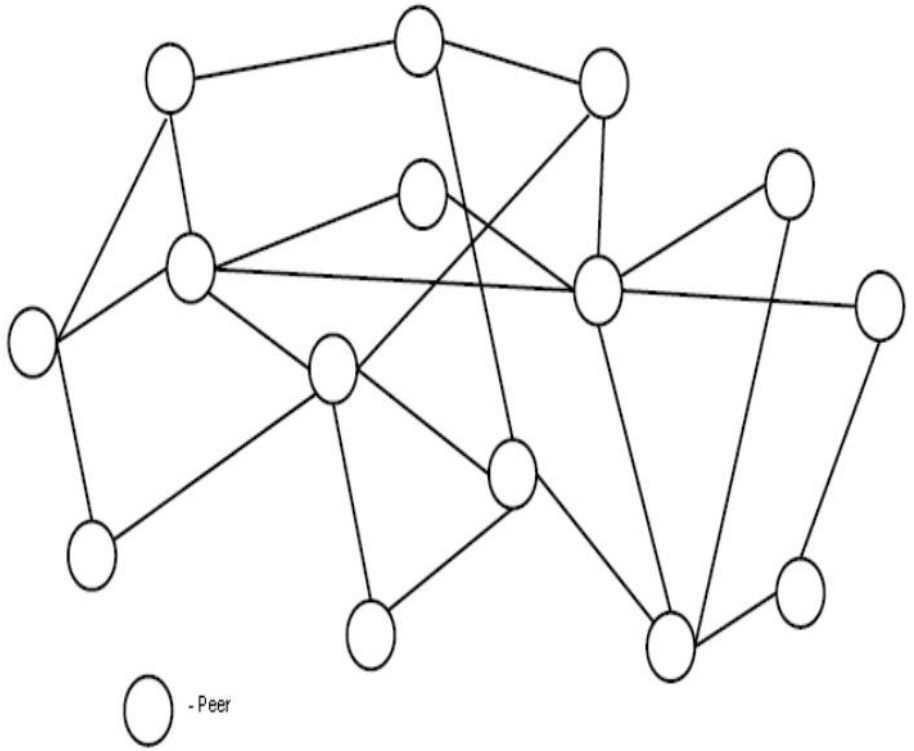
P2P não estruturada

- Portanto, resiliente às falhas de nó. Para localizar determinado objeto é, então, necessário fazer uma busca na topologia de rede resultante. Uma estratégia de inundação

- Exemplos: sistemas como Gnutella, FreeNet e BitTorrent adotando todos estratégias não estruturadas).

P2P Não-Estruturadas

- Algoritmos aleatórios
- Cada peer possui uma lista de vizinhos (visão parcial)
- Para encontrar dados, inundar a rede (no pior caso)
- Importante atualizar a lista de vizinhos. Mas como?



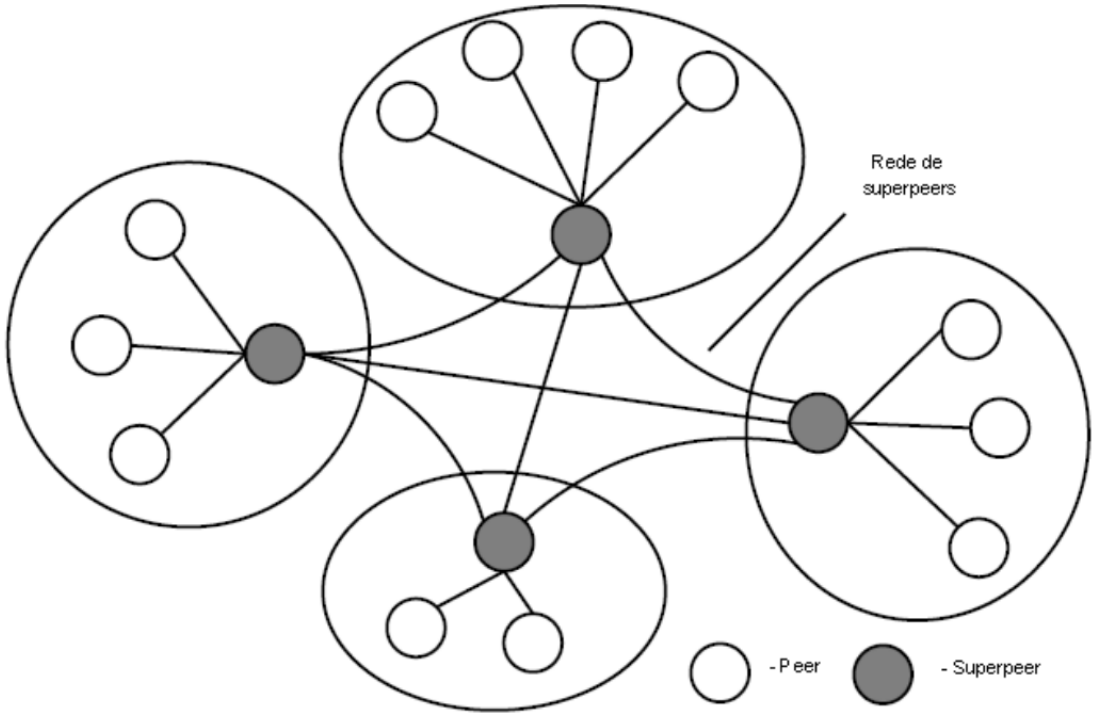
Estratégias para pesquisa de busca em redes não estruturadas

- **Pesquisa em anel expandida**: nesta estratégia, o nó iniciador realiza uma série de buscas com valores cada vez maiores do campo de tempo de vida (time-to-live), reconhecendo que um número significativo de requisições será satisfeito localmente.
- **Caminhadas aleatórias** (*random walks*): com caminhadas aleatórias, o nó iniciador define vários caminhantes que seguem seus próprios trajetos aleatórios no grafo interconectado oferecido pela sobreposição não estruturada.
- **Fococa**: nas estratégias de fofoca, um nó envia uma requisição para determinado vizinho com certa probabilidade e, assim, as requisições se propagam pela rede de maneira semelhante a um vírus em uma população (por isso, os protocolos de fofoca também são referidos como protocolos epidêmicos).

Problema P2P Não-Estruturadas

- Um dos problemas: como encontrar os dados de maneira eficiente?

Solução: Muitos sistemas utilizam nós especiais, que possuem um índice de itens de dados -> **Superpeers**



Sistemas *peer-to-peer* estruturados versus não estruturados

	Peer-to-peer estruturados	*Peer-to-peer não estruturados*
Vantagens	Há garantia na localização de objetos (supondo que eles existam) e podem oferecer limites de tempo e complexidade nessa operação; sobrecarga de mensagens relativamente baixa.	Organizam a si próprios e são naturalmente resilientes às falhas de nó.
Desvantagens	Necessidade de manter as estruturas da sobreposição que são frequentemente complexas e que podem ser difíceis e dispendiosas de obter, especialmente em ambientes altamente dinâmicos.	Probabilísticos e, assim, não podem oferecer garantias absolutas sobre a localização de objetos; propensos à sobrecarga de troca de mensagens excessiva, o que pode afetar a escalabilidade.

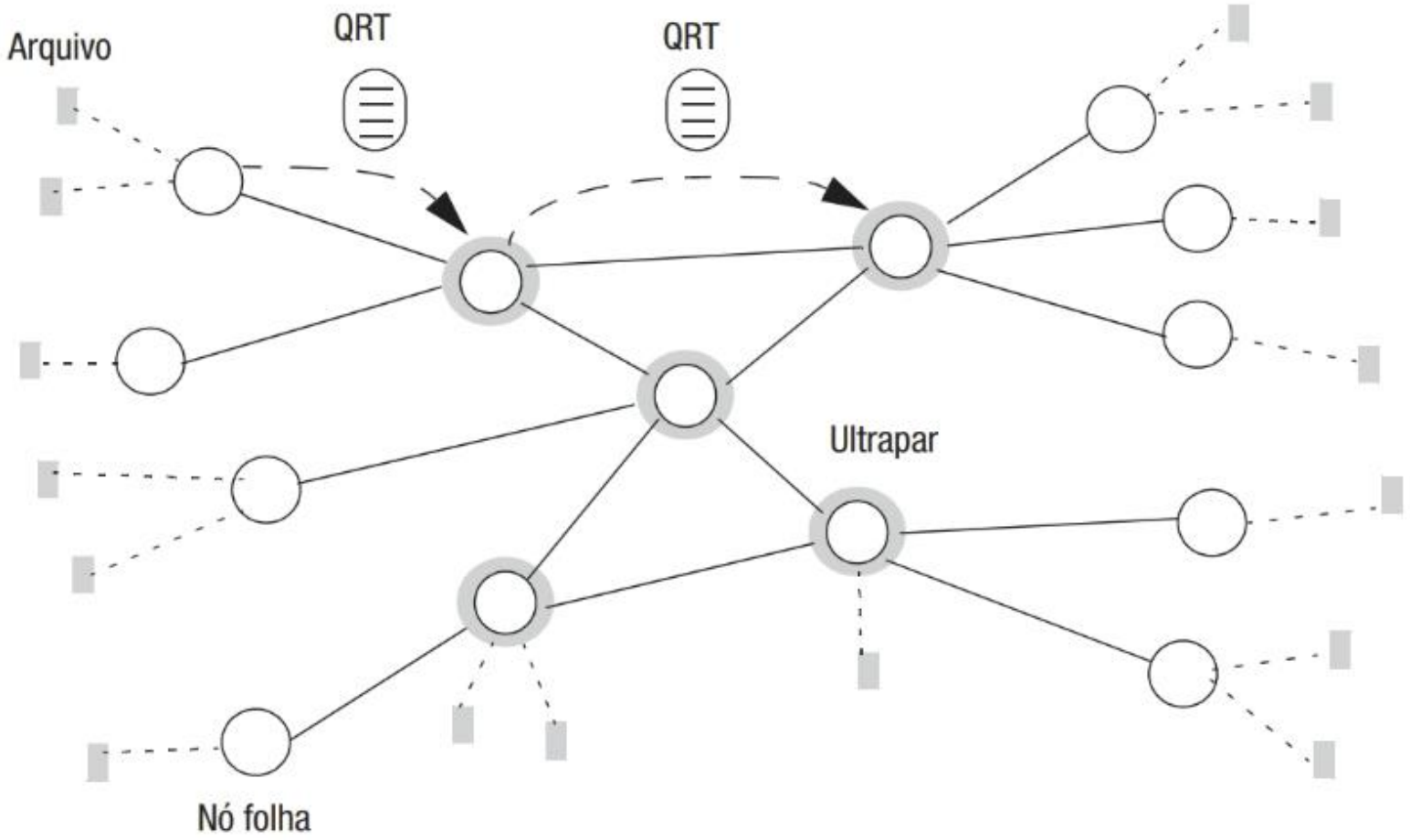
Estudo de caso: Principais elementos do protocolo Gnutella.

- Ultrapares - designados para ter recursos adicionais, são eleitos e formam a parte principal da rede.
- Folhas - se conectam com um pequeno número de ultrapares, os quais se conectam em grande quantidade com outros ultrapares (com mais de 32 conexões cada um)
- QRP (*Query Routing Protocol*) - protocolo projetado para reduzir o número de consultas feitas pelos nós. O protocolo é baseado na troca de informações sobre os arquivos contidos nos nós e, assim, só encaminham as consultas para os caminhos nos quais o sistema acredita que vai haver um resultado positivo.
- Em vez de compartilhar informações sobre os arquivos diretamente, o protocolo produz um conjunto de números a partir do resumo feito nas palavras individuais presentes em um nome de arquivo. Exemplo: um nome de arquivo como “Aula de SD” vai ser representado por quatro números, digamos <65, 47, 09, 76>.

Estudo de caso: Principais elementos do protocolo Gnutella.

- Um nó produz uma tabela QRT (*Query Routing Table*) contendo os valores de resumo representando cada um dos arquivos contidos nesse nó, a qual ele envia para todos os ultrapares associados.
- Os ultrapares produzem suas próprias *Query Routing Tables* com base em uma união de todas as entradas de todas as folhas conectadas, junto às entradas dos arquivos contidos nesse nó, e trocam isso com outros ultrapares conectados.
- os ultrapares podem determinar quais caminhos oferecem uma rota válida para determinada consulta, reduzindo significativamente o volume de tráfego desnecessário

Estudo de caso: Gnutella



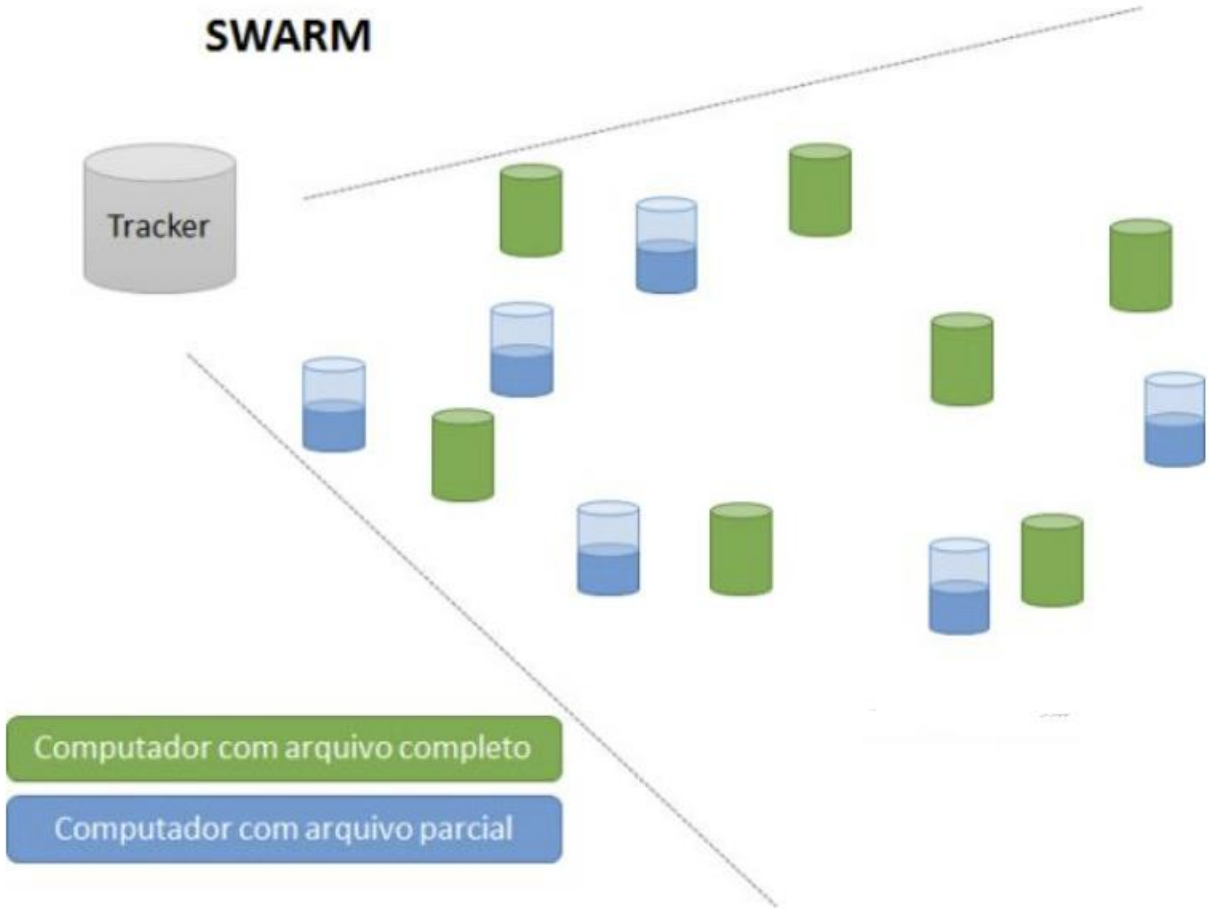
Estudo de caso: BitTorrent

- O BitTorrent em si, na verdade, é um protocolo de compartilhamento de dados e não um sistema centralizado. Não há um servidor provendo os dados, mas sim um padrão de comunicação entre vários computadores que permite que arquivos sejam localizados, distribuídos e obtidos por todos.
- Para compartilhar um arquivo usando o BitTorrent, um utilizador cria um arquivo **.torrent**, um pequeno "apontador" que contém: nome do arquivo, tamanho do arquivo, hash, a matrícula de cada bloco de arquivo.

Elementos bitTorrent

- **Seed** (semeador): é o nome dado a cada máquina que possui o arquivo *completo* que está sendo compartilhado. É necessário que haja pelo menos um seed para que o compartilhamento ocorra integralmente;
- **Peer** (ponto, nó): termo que indica cada computador que compartilha arquivos. Quando você está baixando algo pelo BitTorrent, seu computador assume o papel de um peer, ou seja, de um ponto ou nó na rede;
- **Leecher** ("sugador"): o termo faz referência aos computadores que ainda estão baixando arquivos ou que já o baixaram completamente, mas por alguma razão não o estão compartilhando;
- **Tracker** (rastreador): o tracker é um servidor que mantém o controle de comunicação entre todos os seeds e peers, de forma que os computadores envolvidos no processo possam saber a quais máquinas se conectar. Note, no entanto, que o tracker não tem cópia do arquivo, muito menos interfere diretamente no compartilhamento.
- **Swarm** (enxame): nome dado ao conjunto de computadores que está compartilhando o mesmo arquivo. Se, por exemplo, o arquivo *infowester.avi* estiver sendo compartilhado por oito seeds e 34 peers, o swarm do arquivo contém 42 computadores (oito seeds + 34 peers).

Arquiteturas P2P Híbridas - BitTorrent



Arquiteturas P2P Híbridas - BitTorrent