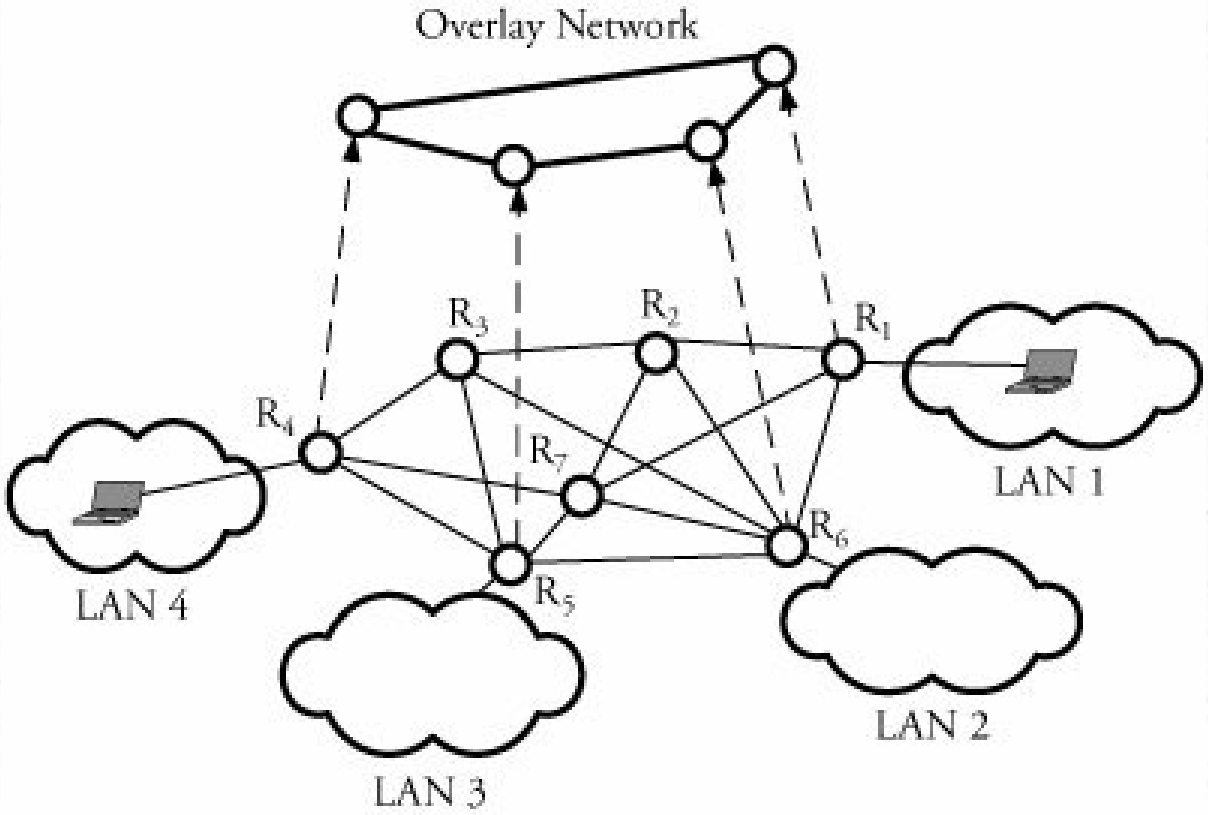
Aula 4
Redes Peer to Peer

Prof. Rafael Guimarães
FAESA

O que são as rede peer to peer (P2P)?

- Todos os nós são iguais (não há cliente e servidor)
- Nós se autodescobrem
- Recursos nas redes de acesso e não em ambientes controlados (como os servidores)
- Algoritmos eficientes para disponibilização de dados em diversos nós e posterior acesso a eles
- Conectividade pode ser intermitente!

P2P são redes overlay

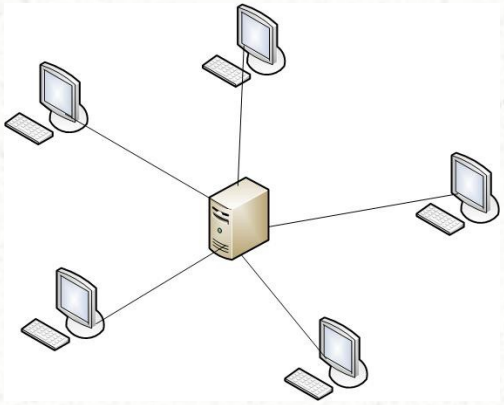


P2P são redes overlay

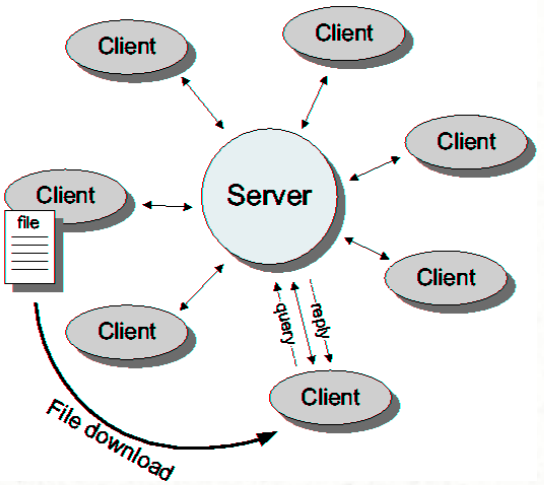
- Enlace: conexão TCP ou IP com “vizinho”
- “Vizinhos” podem ser escolhidos com base na topologia (atraso mínimo, por exemplo)
- Identificadores globais únicos (GUID) para nós e objetos armazenados (hash de 128 bits – SHA-1 – para valores de objetos, por exemplo)
- Localização do objeto pode ser aleatório e independente da topologia da rede
- Objetos e rotas podem ser replicados

Estruturas P2P

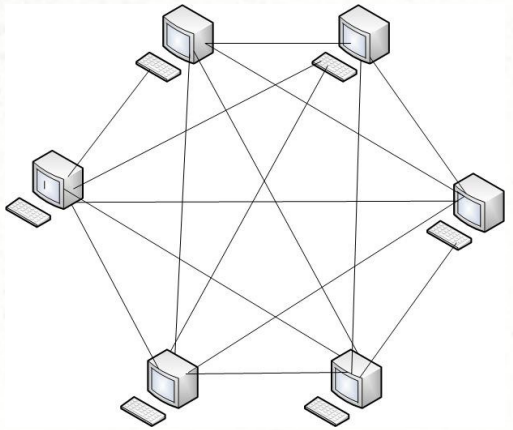
- Cliente-servidor (não é P2P)
- Servidor armazena dados + diretório



- Diretório centralizado, dados distribuídos (ex: Napster)
- Servidor é ponto de vulnerabilidade



- Diretório e dados distribuídos (ex: Gnutella)
- Onde está diretório? Difícil encontrar informação...



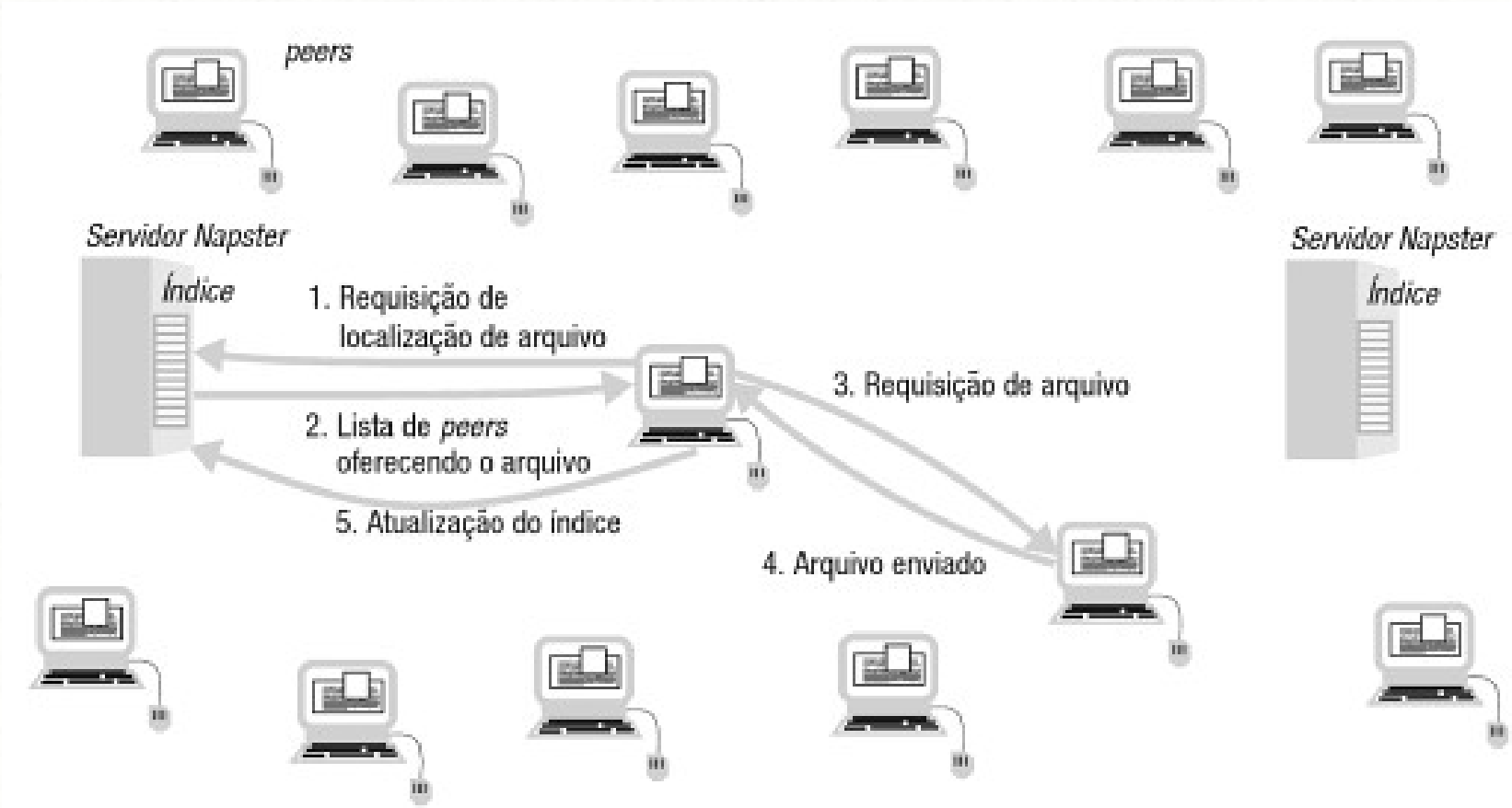
Problemas de compartilhamento

- Entrar na rede
  - Enviar requisição por broadcast
  - Servidor conhecido
- Publicar arquivo
  - Servidor centralizado / replicado
  - Lista local de conteúdo
- Procurar por arquivo
  - Diretório centralizado / replicado
  - Requisição em broadcast
  - Hash tables distribuídas
- Obter arquivo
  - Transferência ponto a ponto
  - Transferência distribuída de segmentos

Napster

- Compartilhamento de músicas
  - Em realidade uma aplicação para encontrar e baixar MP3 livres
- Centralizado (diretório replicado)
- Napster provia diretório com o que usuários estavam provendo + informação de conectividade
- Usuários proviam armazenamento e largura de banda (mais de 60 milhões de usuários)

Napster


peers
Servidor Napster
Índice
1. Requisição de localização de arquivo
2. Lista de peers oferecendo o arquivo
3. Requisição de arquivo
4. Arquivo enviado
5. Atualização do índice
Servidor Napster
Índice

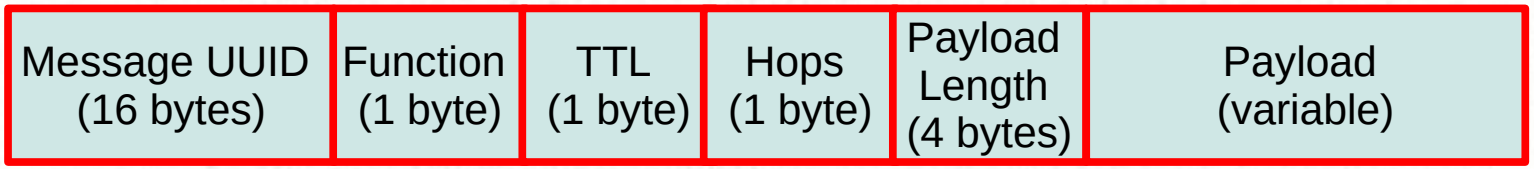
Gnutella

- Compartilhamento de arquivos com índice descentralizado
- Para entrar na rede, basta localizar um nó de alguma forma (amigos, publicação em servidores etc)
- Cada nó mantém lista de outros nós que ele conhece (vizinhos para os quais possui conexões TCP abertas)
- Ao receber mensagem, reencaminha a todos os vizinhos (exceto de quem recebeu a mensagem)

Gnutella

- 2 tipos de mensagens: ping e query
  - Ping: saber sobre vizinhos
  - Query: obter informações sobre arquivos
- Respostas a ping e query retorna pelo caminho que percorreram
- Cada mensagem possui ID único de 16 bytes
  - Mensagens duplicadas são descartadas
- Mensagens possuem TTL máximo de 7 hops
- Pode receber múltiplas resposta a query e elege de quem baixar o arquivo via requisição HTTP

Gnutella



- Ping
  - Usado para entrar na rede e descobrir vizinhos. Sem payload. Todos que recebem ping reencaminham aos vizinhos
- Pong
  - Resposta ao Ping. Payload contém endereço IP, porta, número de arquivo, tamanho dos arquivos. Pong pode ser enviado por todos os nós que recebam Ping multicast

Gnutella

- Query
  - Usado para localizar recursos. Payload contém a taxa de transferência mínima do enlace (2 bytes) e uma string com o critério de busca (pode incluir wildcards)
- Query_Hit
  - Resposta a uma Query. Gerada por nós onde a query teve sucesso. Payload contém número de hits (1 byte), IP/porta (6 bytes), taxa de transferência do enlace (4 bytes), identificador do nó (16 bytes), lista de hits com índices (4 bytes), tamanho de arquivo (4 bytes). Usa caminho reverso da Query.

Avaliação do Gnutella

- Algoritmo ineficiente (muito tráfego)
- Nós com altas taxas de transferência tendem a concentrar conexões e virarem “backbones”
- TTL limita visibilidade de nós (provê escalabilidade)
- Alto tráfego gerado por pings e queries
- Ping e query não contém endereço de origem (difícil saber quem está buscando arquivos)

Pastry

- Atribui-se GUID de 128 para nós e objetos aplicando um algoritmo de hash seguro na chave pública do nó ou conteúdo do objeto
- Colisões são muito pouco prováveis
- Se o GUID identificar um nó ativo, mensagem é entregue a ele, senão é entregue ao nó com GUID numericamente mais próximo
- Entregue em O(log N) passos
- Vizinhos são selecionados segundo número de hops ou atraso da camada inferior de transporte

Pastry

- publish(GUID)
  - GUID pode ser computado pelo objeto (ou parte dele, por exemplo, o nome)
  - Essa função permite que um nó declare que possui o objeto que corresponde ao GUID
- unpublish(GUID)
  - Torna o objeto correspondente ao GUID inacessível

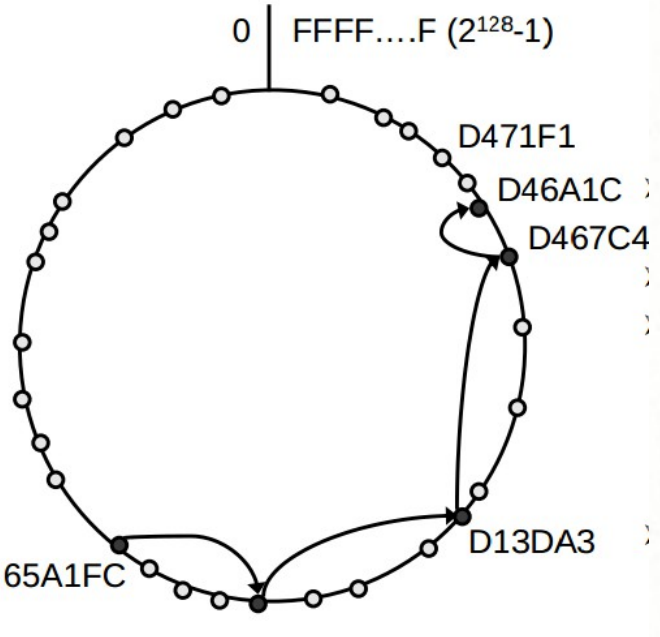
Pastry

- sendToObj(msg,GUID,[n])
  - Mensagem de invocação enviada a um objeto para acessá-lo. Pode ser uma requisição para abrir uma conexão TCP para transferir todo seus dados ou parte deles.
  - O parâmetro opcional final ([n]) solicita que a mensagem seja enviada a n réplicas do objeto

Pastry

- put(GUID, data)
  - O dado é armazenado em réplicas em todos os nós responsáveis pelo objeto identificado por GUID
- remove(GUID)
  - Remove todas as referências ao GUID e os dados associados
- value = get(GUID)
  - Os dados associados ao GUID são obtidos de um dos nós responsáveis por ele

Roteamento simples no Pastry



- Cada nó mantém lista de I nós para frente e para trás
- Os nós escuros são os que existem
- O diagrama ilustra o roteamento de uma mensagem do nó 65A1FC para D46A1C assumindo que I=4 (total de 8 vizinhos)
- Bastante ineficiente!

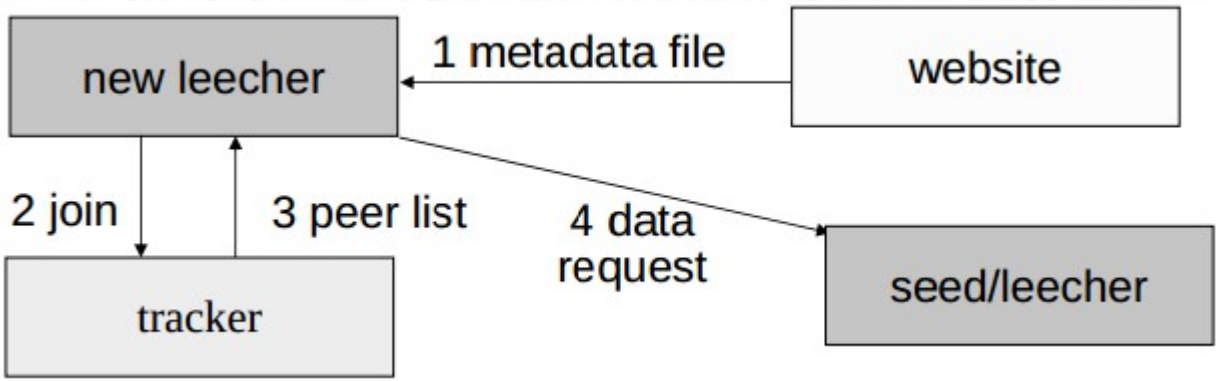
Pastry

- Um novo nó deve apenas conhecer qualquer nó da rede Pastry. Ao contactá-lo, o novo nó é roteado até aquele numericamente mais próximo. Durante o roteamento, a tabela de rotas do nó mais próximo é enviado ao novo nó e ele pode gerar uma tabela inicial baseado nessa informação.
- Se um nó falha, seus vizinhos detectam e solicitam a algum vizinho sua tabela de rotas a fim de selecionar novo vizinho substituto

BitTorrent

- Distribuição de conteúdo multimídia
- Focado em download eficiente, não em busca
- Distribui mesmo arquivo a todos os pares
- Um publicador apenas e diversos downloads
- Quem realiza downloads, compartilha segmentos do dado

BitTorrent



1. Localiza arquivo com metadados (arq.torrent)
    1. Tamanho, nome, hash e URL do tracker
2. Contacta tracker
3. Tracker fornece lista aleatória de pares com arquivo
    1. Seeds: possuem o arquivo completo
    2. Leechers: ainda estão realizando download
4. Contacta pares da lista para obter dados

BitTorrent

- Arquivo é quebrado em pedaços
  - Geralmente de 256KB (pode ter sub-pedaços de 16KB)
- Download de pedaços em paralelo
- Avisa a lista de pares sobre pedaços recebidos
- Upload de pedaços enquanto faz download
- Inicia download por pedaços mais raros para aumentar a oferta

BitTorrent

- Upload de seleção
  - Periodicamente calcula a taxa de downloads
  - Seleciona até 4 pares com as maiores taxas para fazer upload do que foi baixado
  - Periodicamente (a cada 30s) seleciona um par aleatoriamente e faz upload para ele

JXTA

- Framework Java para implementação de aplicações P2P (www.jxta.org)