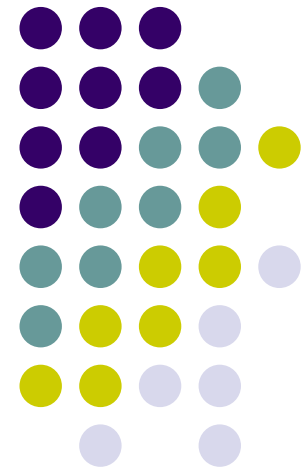


# Sistemas Distribuídos

---

Comunicação de Grupos

Peer to Peer



# Comunicação de Grupos

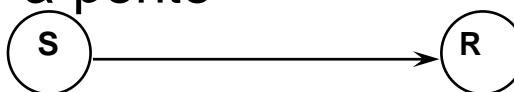


- Modelos Anteriores - Comunicação envolvia somente duas partes.
- RPC não permite comunicação de um processo com vários outros
- Grupo é uma coleção de processos que agem juntos em um sistema, de tal forma que quando uma mensagem é enviada para o grupo, todos os membros do grupo a recebem.

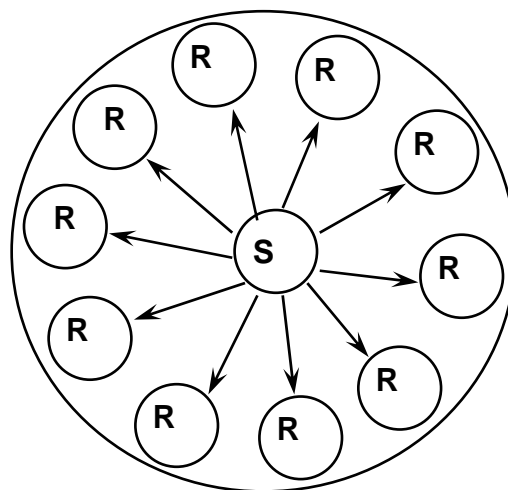


# Comunicação de Grupos

- Comunicação ponto-a-ponto



- Comunicação um-para-vários



# Comunicação de Grupos



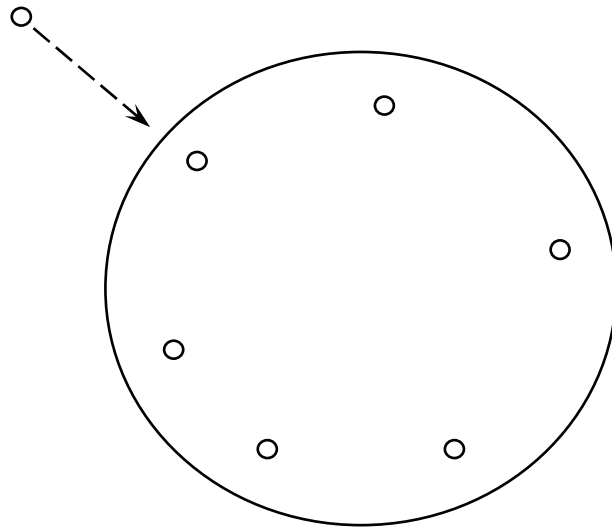
- Grupos são dinâmicos. Novos grupos podem ser criados e grupos existentes podem ser eliminados. Um processo pode entrar para o grupo ou o deixá-lo. Um processo pode ser membro de diversos grupo simultaneamente
- A implementação Depende do Hardware:
- Multicasting: Endereço especial que múltiplas máquinas podem receber (224.x.x.x – 239.x.x.x)
- Implementação é direta - basta atribuir a cada grupo um endereço multicasting diferente

# Comunicação de Grupos

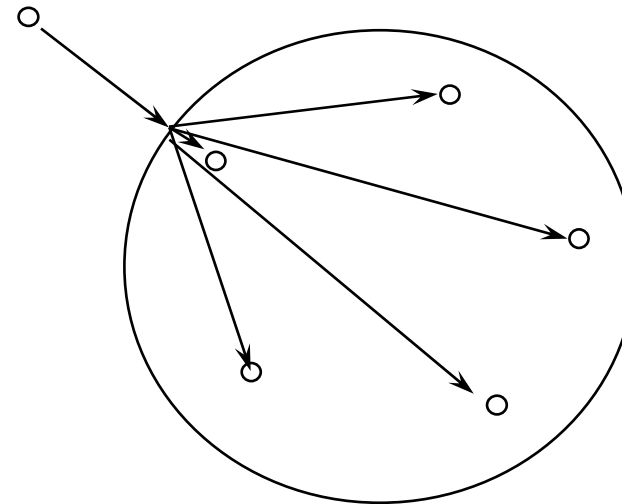


- Broadcasting: Pacotes contendo certos endereços são enviados para todas as máquinas. Menos eficiente que multicasting (todas as máquinas recebem as mensagens enviadas por broadcasting e o software precisa verificar se o pacote é para ele). Também necessita somente um pacote para atingir todos os membros do grupo.
- Unicasting: Transmissão separada de pacote para cada membro do grupo ( $n$  membros,  $n$  pacotes necessários)

# Grupos Fechados e Abertos

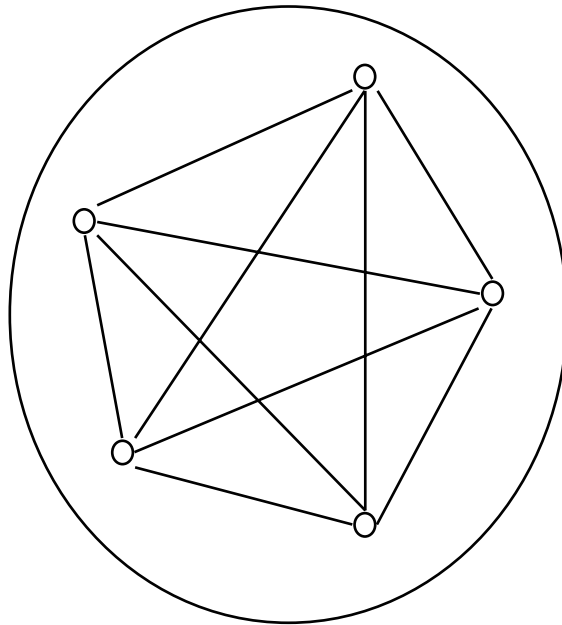
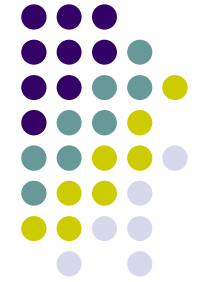


**Grupo Fechado**  
Ex: Processamento paralelo

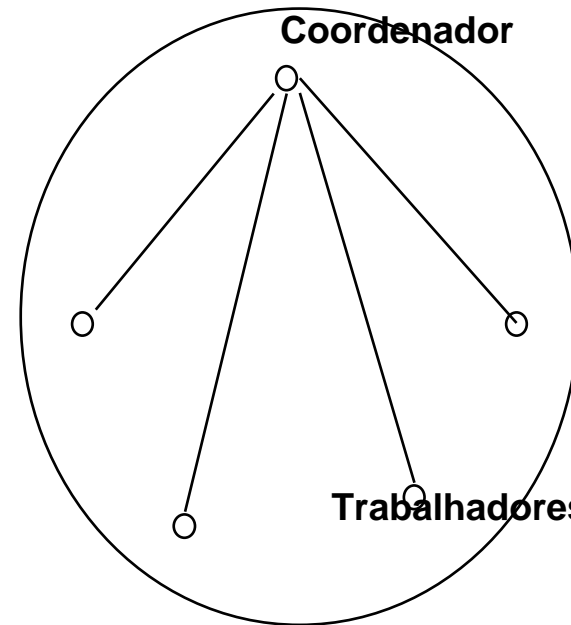


**Grupo Aberto**  
Ex: Servidores replicados

# Grupos Hierárquicos e “de iguais”



Grupo “de Iguais”



Grupo Hierarquico

# Grupos Hierárquicos e “de iguais”



- Hierarquico:
  - Vantagem- decisões mais rápidas
  - Desvantagem - a perda do coordenador para o grupo todo.
- Servidor de Grupo: criação e eliminação de grupos e permissão para processos aderir ou abandonar grupos.
- Monta uma base de dados com informações dos grupos (ponto crítico de falha)
- Outra forma é fazer o gerenciamento de forma distribuída.

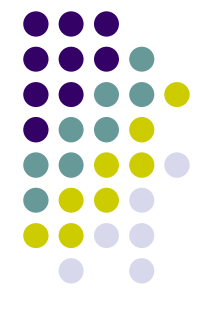


# Grupos

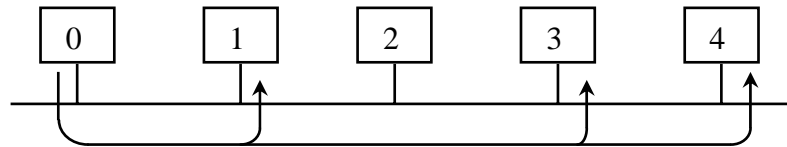


- Quando um processo deixa o grupo, envia uma mensagem de “goodbye” para todos os outros membros.
- Problema: Se um membro falha ele deixa o grupo sem o “goodbye”. Os outros membros tem que descobrir isso experimentalmente e removê-lo do grupo.

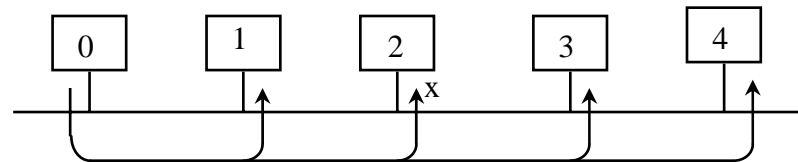
# Endereçamento



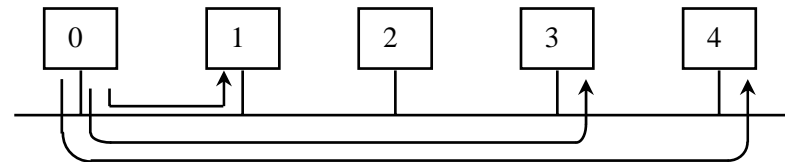
- Multicast



- Broadcast



- Unicast



# Atomicidade



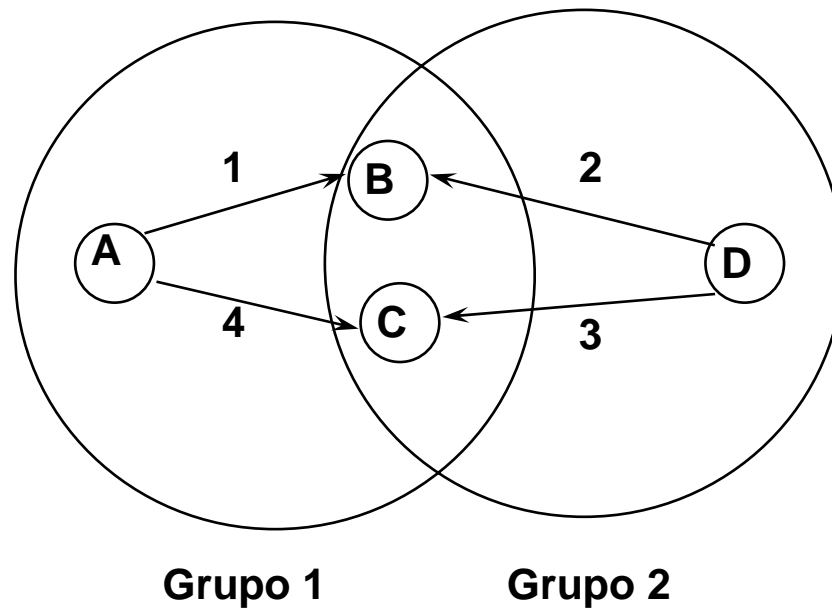
- Quando uma mensagem é enviada para um grupo, ela chega corretamente para todos os membros do grupo ou não chega para nenhum membro.
- Uma maneira de ter certeza que todos os destinatários receberam a mensagem é implementar o envio do ACK para cada mensagem recebida (Sem falhas esse método funciona)
- Com falhas - processo que envia *msg* inicia *timers* e envia retransmissões quando necessário. Quando um processo recebe uma *msg*, e não a havia visto ainda, envia a *msg* para todos os membros do grupo – problema: sobrecarga.



# Ordenando Mensagens

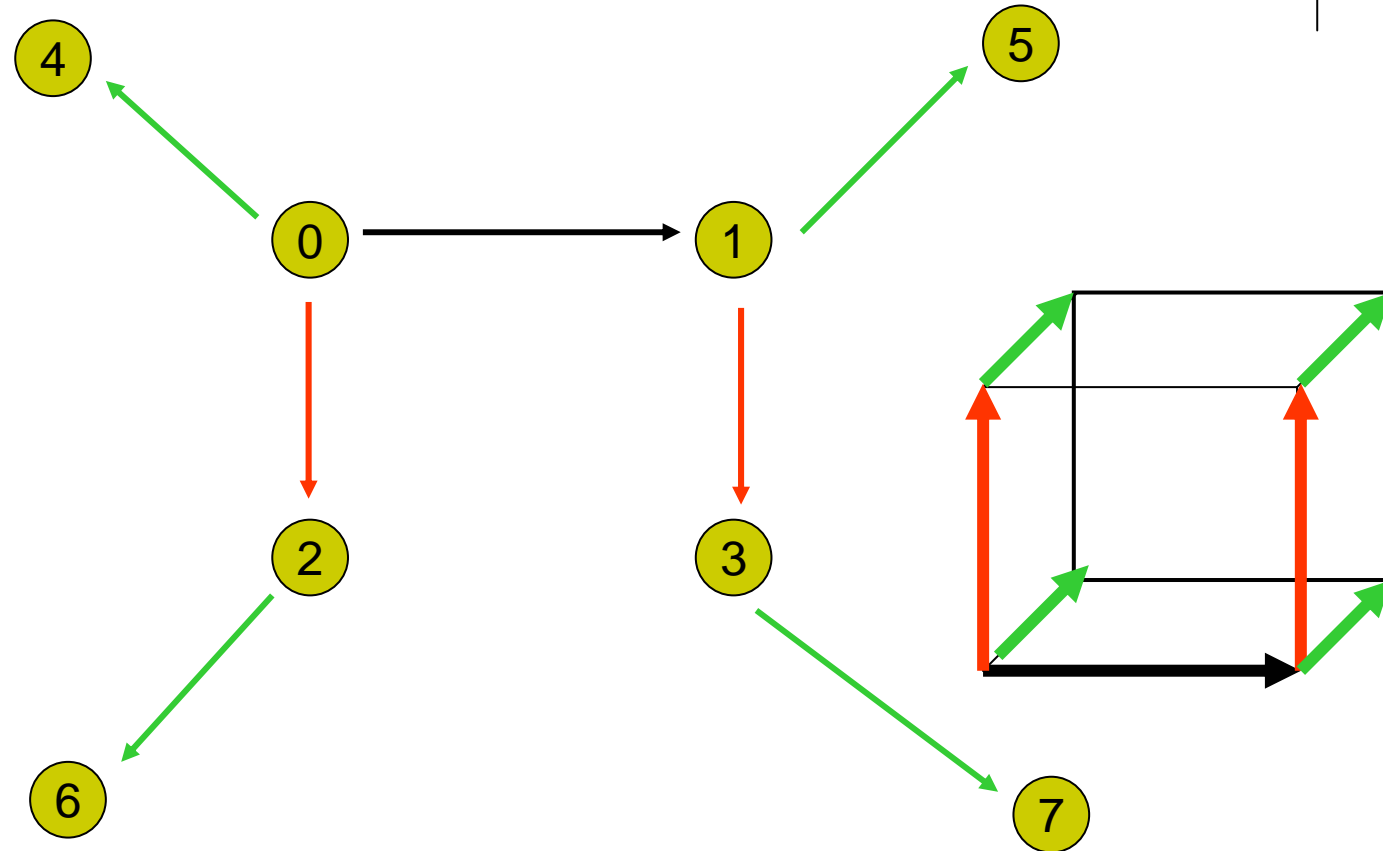
- Global Time Ordering - todas as mensagens chegam na ordem exata em que foram enviadas (não é fácil de ser implementado)
- Consistent Time Ordering - se duas mensagens são enviadas em tempos próximos, o sistema pega uma delas como sendo a “primeira” e envia a todos os membros do grupo segundo esta ordem (é garantido que as mensagens cheguem a todos os membros do grupo na mesma ordem; que podem não ser a ordem real)

# Overlapping Groups



Mesmo usando GTO, podem haver msgs chegando em ordem diferente

# Algoritmos Envolviendo Grupos: Hiper-cubo



# Desafio:



- Implementar este tipo de algoritmo em um ambiente distribuído

# Peer 2 Peer



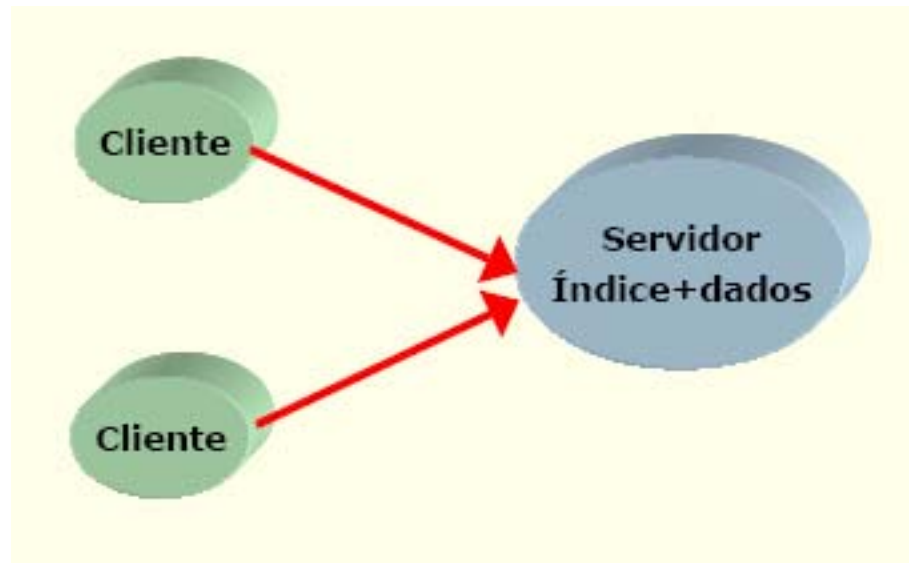
- Aplicações Peer-to-peer distribuem a informação entre seus nós membros em lugar de concentrar em um único servidor.
- Não há a necessidade de nenhum elemento coordenador ou centralizador de recursos ou políticas
- Existe um certo grau de anonimato para o proprietário do recurso
- Todos os nós membros possuem a mesma capacidade de compartilhar informação com os demais membros da rede (todos seriam ao mesmo tempo clientes e servidores de dados).
- Cada usuário torna seu repositório de informações disponível para distribuição e pode estabelecer conexão direta com outro usuário.





# Peer 2 Peer

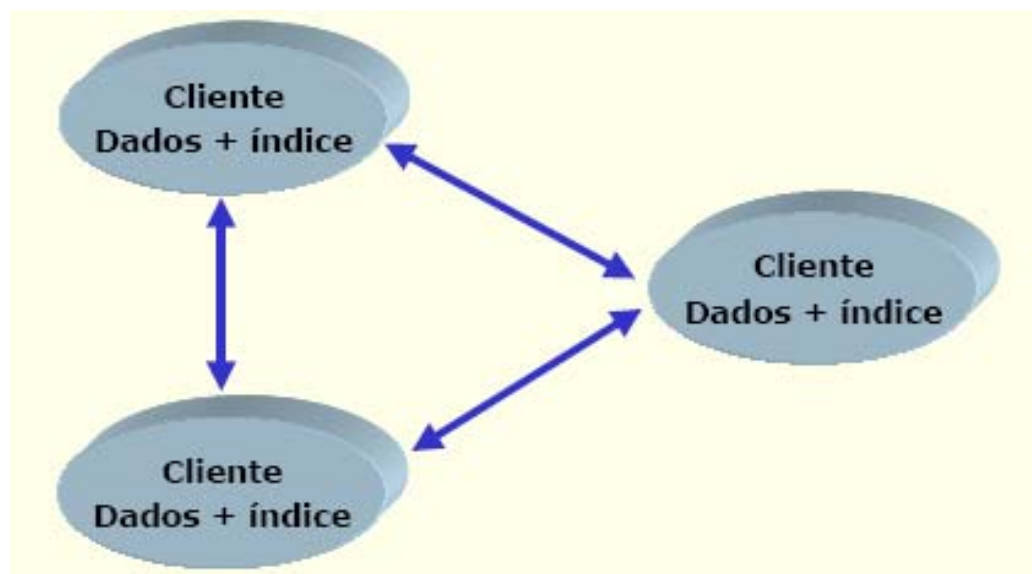
- No modo de operação cliente/servidor tem-se o acesso aos dados e ao índice centralizados no servidor.





# Peer 2 Peer puro

- No modo de operação puramente P2P tanto os dados quanto o índice são distribuídos.





# Requisitos de Peer to Peer

- Escalabilidade Global
  - Imensas quantidades de hosts conectados à rede
  - Milhares de objetos e dezenas de milhares de hosts
- Balanceamento de carga
  - Distribuição equalitária entre os peers
  - Possibilidade de download de diferentes peers, em função de sua carga
- Otimização das interações locais entre peers vizinhos
  - Idéia é buscar vizinhos mais “próximos”, evitando a latência da comunicação

# Requisitos de Peer to Peer



- Dinamicidade dos hosts
  - Peers podem entrar e sair do sistema a qualquer momento
  - Quando entram, devem ser integrados ao sistema global
  - Quando saem (voluntariamente ou não) o próprio sistema deve detectar e adequar a nova carga
- Segurança dos dados em um ambiente heterogêneo
  - Autenticação, criptografia, necessidade de membros da “Rede P2P”
- Anonimato, capacidade de Negação e resistência à censura
  - Capacidade de negar o compartilhamento de um arquivo
  - Possibilidade de não realizar download de conteúdo protegido

# Aplicações



- Aplicações de Computação Distribuída muitas vezes se enquadram na categoria P2P tal como SETI@home que utiliza milhões de clientes Internet para procura de vida extraterrestre. É implementado como um *screen saver* ([setiathome.ssl.berkeley.edu](http://setiathome.ssl.berkeley.edu))
- Outras aplicações nesta categoria incluem sistemas para modelagem financeira, bioinformática, teste de desempenho Web. Estes sistemas aproveitam o tempo ocioso da máquina dos clientes para realizar computações de forma distribuída.

# Aplicações



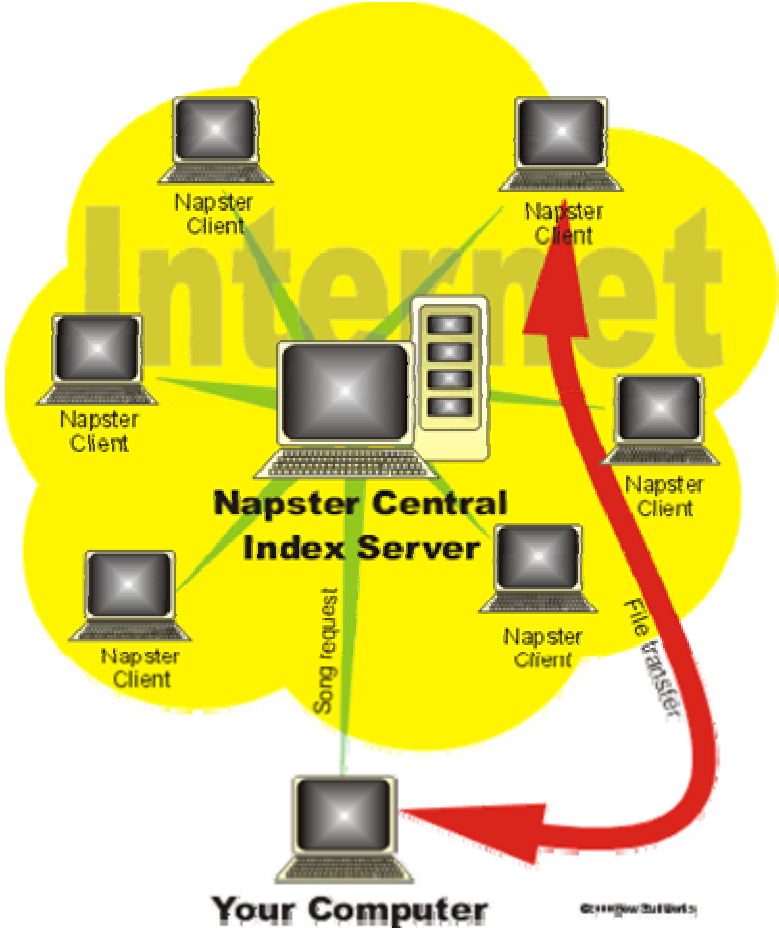
- Aplicações colaborativas também costumam ser consideradas na categoria P2P. Entre estas aplicações se incluem os “Instant Messenger” e salas de Chat ou “White Board”.
- Nas aplicações colaborativas existe interação entre clientes em torno de uma atividade comum que podem ser jogos ou simulações.
- Um exemplo é o “White Board” que é uma aplicação onde cada cliente pode alterar desenhos ou textos e todos os demais visualizam e podem também fazer alterações.



# Estudo de Caso: Napster

- 1o sistema Peer to Peer a ser altamente popularizado
- Troca exclusiva de músicas, principalmente em formato MP3
- Funciona usando uma arquitetura centralizada
  - Servidor de índice, que concentra todas as pesquisas
  - Cada peer, ao ser iniciado, torna-se um servidor de arquivos
    - Exporta seus índices ao servidor central do Napster
    - Mantém a lista de todos os Peers disponíveis

# Napster





# Napster



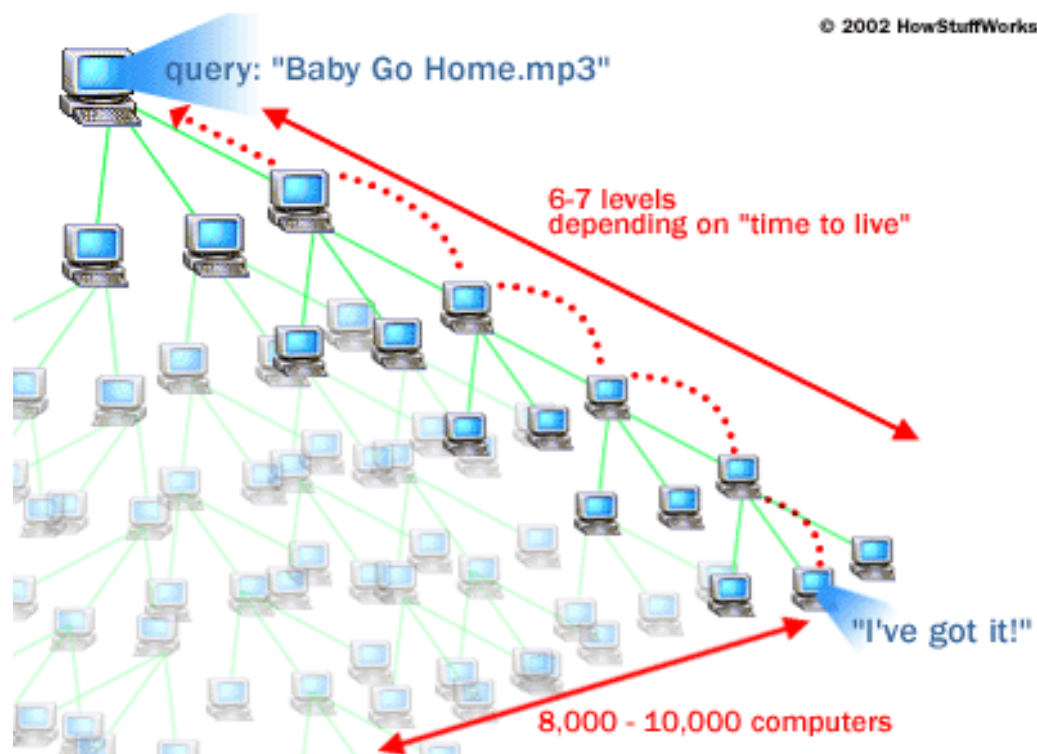
- Cliente que deseja realizar uma pesquisa, envia a query ao Servido Central do Napster
- Este identifica o peer que contém a música com as palavras-chaves da busca
- A troca é feita entre o cliente e o servidor
- Servidor do Napster funciona como um “Binder”

# Estudo de caso 2: Rede GNetella



- Funciona no padrão P2P puro
  - Alta disponibilidade
  - Alta dispersão
  - Alto nível de balanceamento de carga
  - Praticamente impossível de ser eliminada
- Não há garantia de que o arquivo exista em um dos N peers atingíveis a partir do peer de origem
- Cada peer funciona como:
  - Cliente
  - Servidor
  - Gateway – realizando “forward” das mensagens
- Busca?
  - TTL – Time To Live

# Rede GNUtella



# Rede GNetella



- Cada peer conhece, pelo menos, 1 peer vizinho
- A consulta é feita pelo peer de origem e a ela é atribuída um TTL (em geral até 6 ou 7)
- O peer vizinho realiza a consulta localmente e encaminha a consulta para seus vizinhos, incrementando o TTL
- A consulta é propagada até que o TTL atinja seu limite e, então, conforme os resultados são colhidos, a resposta é enviada.
- A partir do momento em que o peer de origem encontra um peer que possui o arquivo, a troca é realizada.



# Rede GNUtella

- Questionamentos
  - Legalidade do uso do software
  - Compartilhamento de conteúdo protegido por Direitos Autorais
- Comparação Napster X Gnutella
  - Napster – Centralizado
    - Ausência do servidor central – ausência do serviço
  - GNUtella – Distribuído
    - Critérios de busca distribuídos entre os diversos nós do sistema
    - Ausência de coordenação global
    - Altíssima disponibilidade