

# Modelagem de Aplicações Distribuídas

Prof. Marcelo Veiga Neves  
[marcelo.neves@pucrs.br](mailto:marcelo.neves@pucrs.br)

# Conteúdo

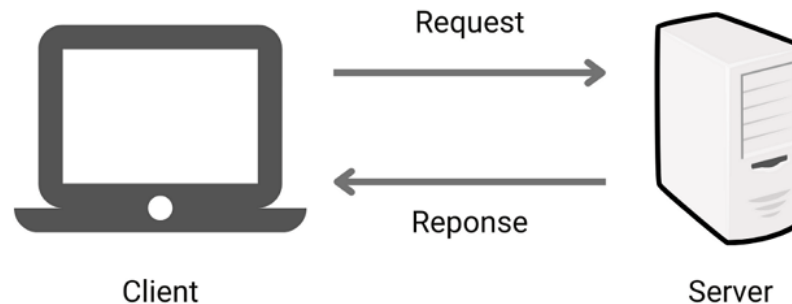
- Modelo cliente/servidor
- Modelo P2P (peer-to-peer)

# Modelo Cliente/Servidor

- Principal motivação: “Problema do encontro” (Rendezvous problem)
- Modelo: para qualquer par de aplicações que se comunicam, um dos lados deve iniciar a execução e esperar (indefinidamente) até ser contatado pelo outro lado
  - Cliente: lado que inicia a comunicação
  - Servidor: lado que espera por requisições de um cliente

# Modelo Cliente/Servidor

- Arquitetura Cliente/Servidor:
  - A camada do cliente invoca serviços ou procedimentos disponibilizados pela camada do servidor
  - Ex: Servidor Web, de arquivos, de impressoras, etc.



# Conteúdo

- Modelo cliente/servidor
- Modelo P2P (peer-to-peer)
  - Características
  - Arquiteturas
  - Redes de sobreposição
  - Indexação e busca
  - Integridade e proteção
  - Exemplos

# Introdução

- Modelo Peer-to-Peer (P2P)
  - peer = par (iguais)
  - São sistemas distribuídos nos quais os membros da rede são equivalentes em funcionalidade
  - Permitem que os pares compartilhem recursos diretamente, sem envolver intermediários

# Conceito de P2P

- Definição de sistema P2P:
  - “São sistemas distribuídos compostos de nós interconectados, aptos a se auto-organizar em topologias de rede, com o intuito de compartilhar recursos, como conteúdo, ciclos de CPU, largura de banda e armazenamento, com a capacidade de adaptação a falhas e acomodação a um número variável de nós, ao mesmo tempo que mantém a conectividade e o desempenho em níveis aceitáveis, sem a necessidade de suporte ou intermediação de um servidor centralizado.” (Adroutsellis-Theotokis & Spinellis, 2004)

# Conceito de P2P

- Outras definições:
  - P2P refers to a class of systems and applications that employ distributed resources to perform a critical function in a decentralized manner (Milojicic et al., HP)
  - P2P is a class of applications that take advantage of resources— storage, cycles, content, human presence – available at the edges of the Internet (Clay Shirky)



# Conceito de P2P

- Não há um consenso total na definição do conceito de P2P
- Sistemas P2P devem suportar os seguintes requisitos:
  - Comunicação direta entre os pares: se opõe ao tradicional modelo cliente-servidor, já que cada nó pode fornecer ou obter recursos
  - Nós com conectividade variável e temporária (endereços IP também -> não precisa ter IP fixo, nome, etc.)
  - Nós podem estar localizados nas “bordas da rede”
  - Nós com autonomia parcial ou total em relação a um servidor central (não há um coordenador do grupo; toda a coordenação pode ser distribuída)
  - Escalabilidade: rede cresce em escala facilmente; não há ponto de estrangulamento

# Classificação de Aplicações

- Comunicação e colaboração (ex: Skype);
- Distribuição de conteúdo (ex: Napster, Torrent).
- Computação distribuída (ex: SETI@Home);
- Sistemas de bancos de dados distribuídos (Blockchain);

# Classificação de Arquitetura

- Arquitetura centralizada: utiliza um servidor central para controle de acesso à rede e para publicação e pesquisa de conteúdo
- Arquitetura descentralizada: todos os peers possuem funcionalidade equivalente
- Arquitetura híbrida: alguns peers especiais, chamados supernós/supernodos, possuem um papel diferenciado na rede

# Arquiteturas P2P

- Arquitetura Centralizada
  - Um servidor central controla as entradas e saídas de peers da rede
  - Os peers registram no servidor central os recursos que compartilharão na rede
  - Pesquisas por recursos disponíveis nos peers são efetuadas pelo servidor central
  - **O acesso aos recursos é feito diretamente entre peers**
  - Problema: ponto único de falha = servidor central
  - Exemplos: Napster; eMule

# Arquiteturas P2P

- Arquitetura Híbrida
  - Supernós permitem o ingresso dos nós na rede, podendo também exercer atividades de coordenação do funcionamento da rede, indexar os recursos compartilhados pelos nós e permitir a busca por estes recursos
  - Após localizado, um recurso pode ser obtido a partir da interação direta entre nós
  - Falha de um supernó pode ser tolerada elegendo dinamicamente outro supernó
  - Exemplos: Kazzaa; Skype

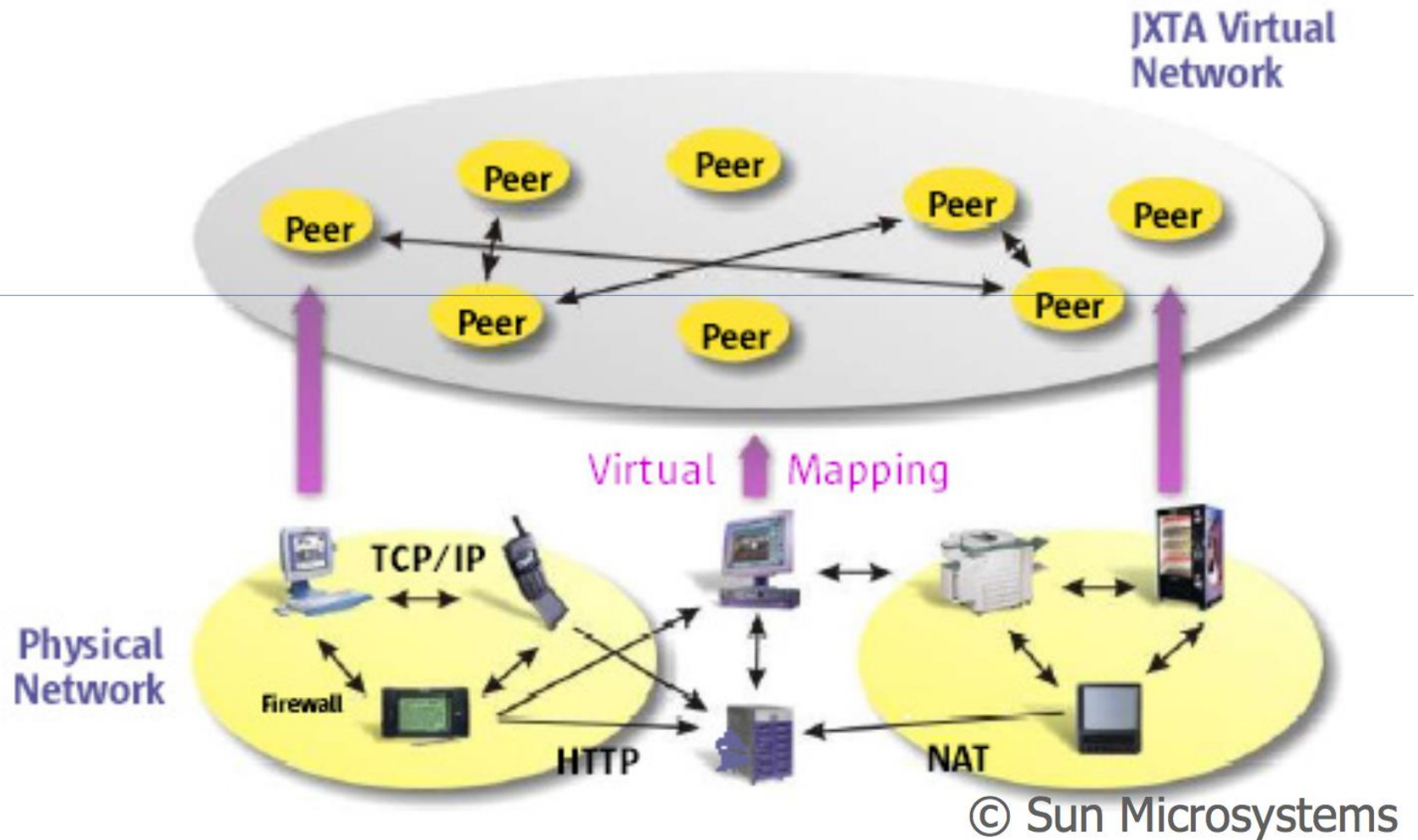
# Arquiteturas P2P

- Arquitetura Descentralizada
  - Não há um elemento central
  - Todos os nós possuem papel equivalente
  - As pesquisas por recursos compartilhados são feitas por inundação (flooding)
  - Gera um alto tráfego na rede
  - Desempenho das pesquisas é ruim devido à necessidade de contactar muitos nós e aguardar a resposta
  - Exemplos: Gnutella e JXTA

# Redes de Sobreposição (Overlay)

- Sistemas P2P normalmente empregam a criação de redes overlay
- Overlay: rede virtual (lógica) em cima de outra rede (normalmente IP)
- Abstrai a complexidade da rede real e pode oferecer serviços adicionais:
  - Descoberta de recursos
  - Nomeação/endereçamento
  - Roteamento, passar por firewalls, etc.

# Redes de Sobreposição (Overlay)





# Exemplos: Napster

- Criado em 1999, Shawn Fanning e Sean Parker
- Utilizava um servidor central (replicado) para efetuar a procura de arquivos na rede
- Transferência de arquivos era feita diretamente entre os peers
- Sem o servidor central, que foi fechado por ordem judicial (2001), a rede deixou de funcionar

# Exemplos: BitTorrent

- Protocolo P2P para download de arquivos mais utilizado atualmente
- O arquivo compartilhado é dividido em vários pedaços, que são baixados simultaneamente
- Pedaços recebidos já são compartilhados
- O protocolo engloba mecanismos para:
  - Controlar a integridade dos pedaços de arquivos baixados
  - Recompensar quem compartilha arquivos, e penalizar quem tira proveito e não colabora

# Exemplos: BitTorrent

- Arquivo .torrent
  - Criado por nó seed, que compartilha arquivo
  - Contém metadados que descrevem o arquivo e permitem verificar sua integridade
  - Indica servidores tracker
- Servidores tracker
  - Coordenam a distribuição de arquivos
  - Indicam os nós para download
  - Controlam a velocidade do download
  - Se o nó não colabora (leech), perde banda

# Referências

- Material baseado em slides do Prof. Prof. Frank Siqueira da UFSC e do Prof. Claudio Geyer da UFRGS.
- E no seguinte livro:
  - Sistemas Distribuídos: Conceitos e Projeto.  
George Coulouris, Tim Kindberg e Jean Dollimore,  
2013.