



UNIVERSIDADE FEDERAL DO CEARÁ  
Campus Quixadá  
Disciplina: Sistemas Distribuídos

# Aula 4 – *Peer-to-Peer*

Prof. Carlos Bruno

# Agenda

---

- Características;
- Napster;
- Middlewares *peer-to-peer*;
- *Routing overlay*;
- Estruturação;
- Busca;
- BitTorrent.

# Características

---

- Compartilhamento de recursos sem a presença de servidor dedicado;
- Objetiva a entrega de serviços de forma **descentralizada** e **auto-organizável**;
- Balanceamento de cargas e armazenamento entre os *peers*.
- Não há garantias de que os nós se manterão ligados. Entretanto, há replicação de dados.

# Características

---

**“São aplicações que exploram recursos disponíveis na fronteira da internet”**

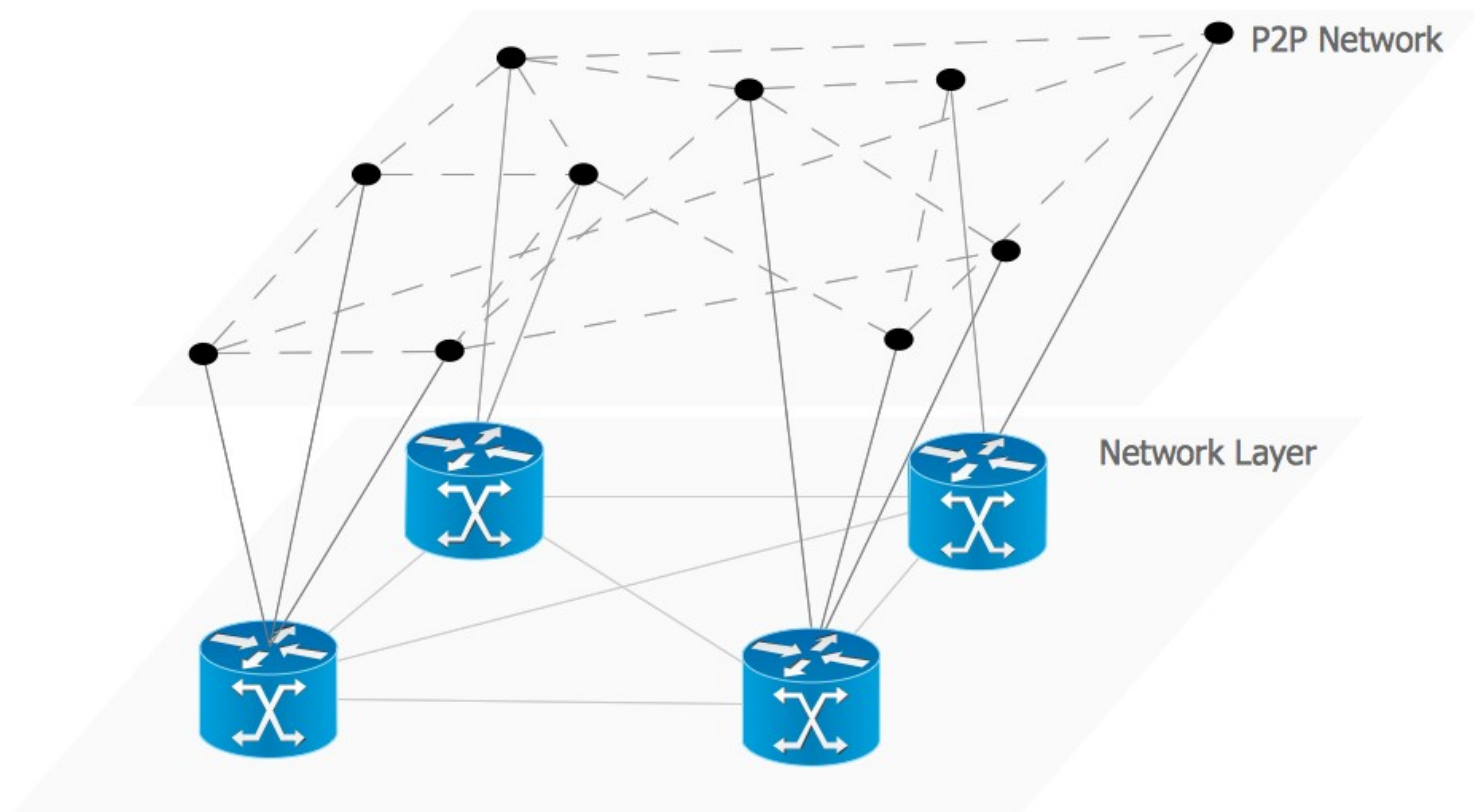
# Características

---

- Os recursos na rede são identificados por **GUID** (globally unique identifiers) que são derivados de um *hash*;
  - *Segurança*
  - *Integridade*
- Os recursos sempre estarão disponíveis em uma rede P2P?
- O sistema de comunicação em pares utiliza uma **rede sobreposta**.

# Características

---



# Características

	<i>IP</i>	<i>Application-level routing overlay</i>
<i>Scale</i>	IPv4 is limited to 232 addressable nodes. The IPv6 name space is much more generous (2128), but addresses in both versions are hierarchically structured and much of the space is pre-allocated according to administrative requirements.	Peer-to-peer systems can address more objects. The GUID name space is very large and flat (>2128), allowing it to be much more fully occupied.
<i>Load balancing</i>	Loads on routers are determined by network topology and associated traffic patterns.	Object locations can be randomized and hence traffic patterns are divorced from the network topology.
<i>Network dynamics (addition/deletion of objects/nodes)</i>	IP routing tables are updated asynchronously on a best-efforts basis with time constants on the order of 1 hour.	Routing tables can be updated synchronously or asynchronously with fractions of a second delays.
<i>Fault tolerance</i>	Redundancy is designed into the IP network by its managers, ensuring tolerance of a single router or network connectivity failure. $n$ -fold replication is costly.	Routes and object references can be replicated $n$ -fold, ensuring tolerance of $n$ failures of nodes or connections.
<i>Target identification</i>	Each IP address maps to exactly one target node.	Messages can be routed to the nearest replica of a target object.
<i>Security and anonymity</i>	Addressing is only secure when all nodes are trusted. Anonymity for the owners of addresses is not achievable.	Security can be achieved even in environments with limited trust. A limited degree of anonymity can be provided.

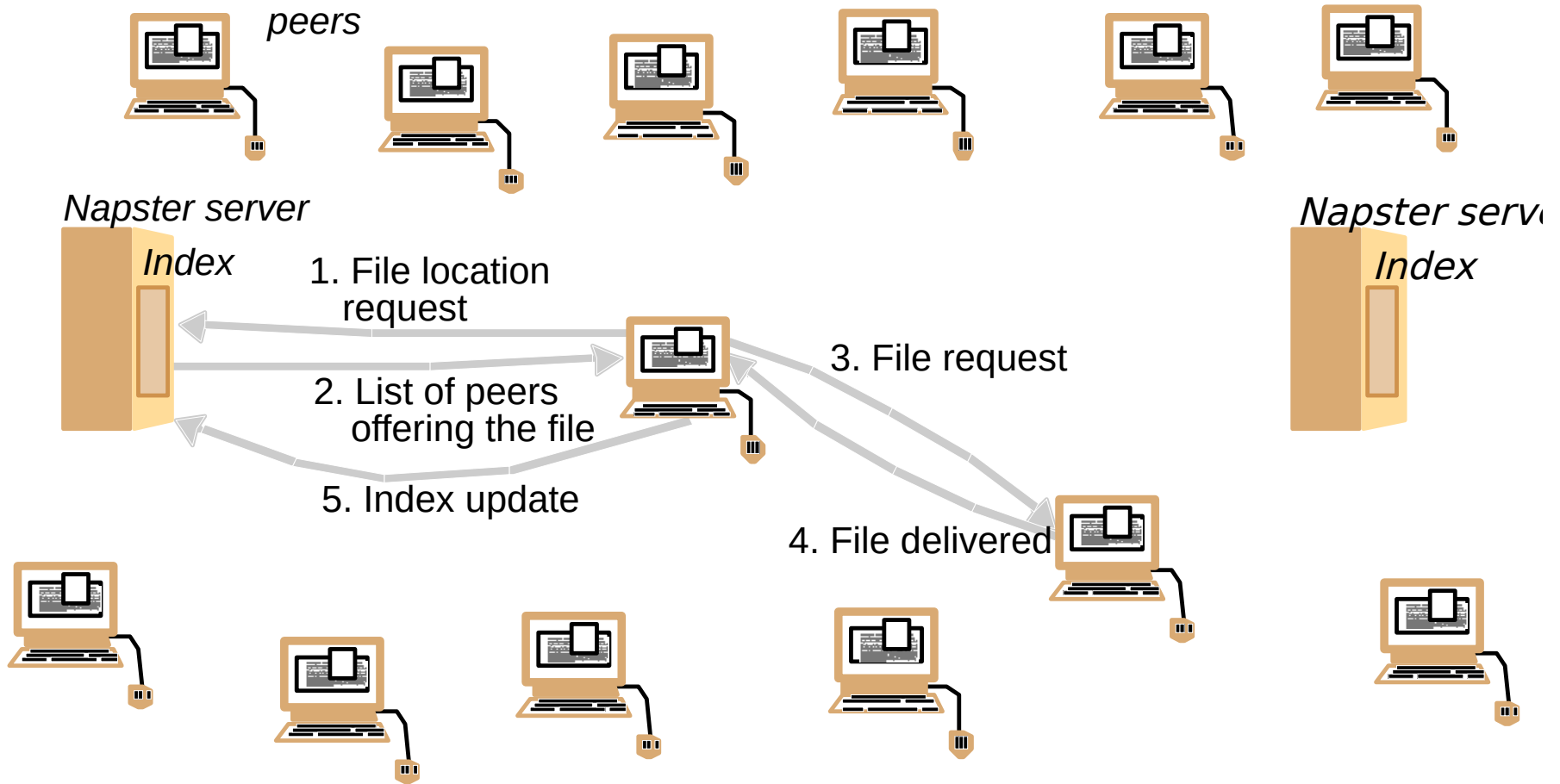
# Napster

---

- A primeira aplicação que fez uso do sistema P2P foi o compartilhamento de músicas;
- A empresa **Napster** foi a pioneira. Os usuários “*não anônimos*” compartilhavam seus recursos em índices centralizados num servidor;
- A empresa enfrentou processos legais e teve que desativar o serviço.



# Napster

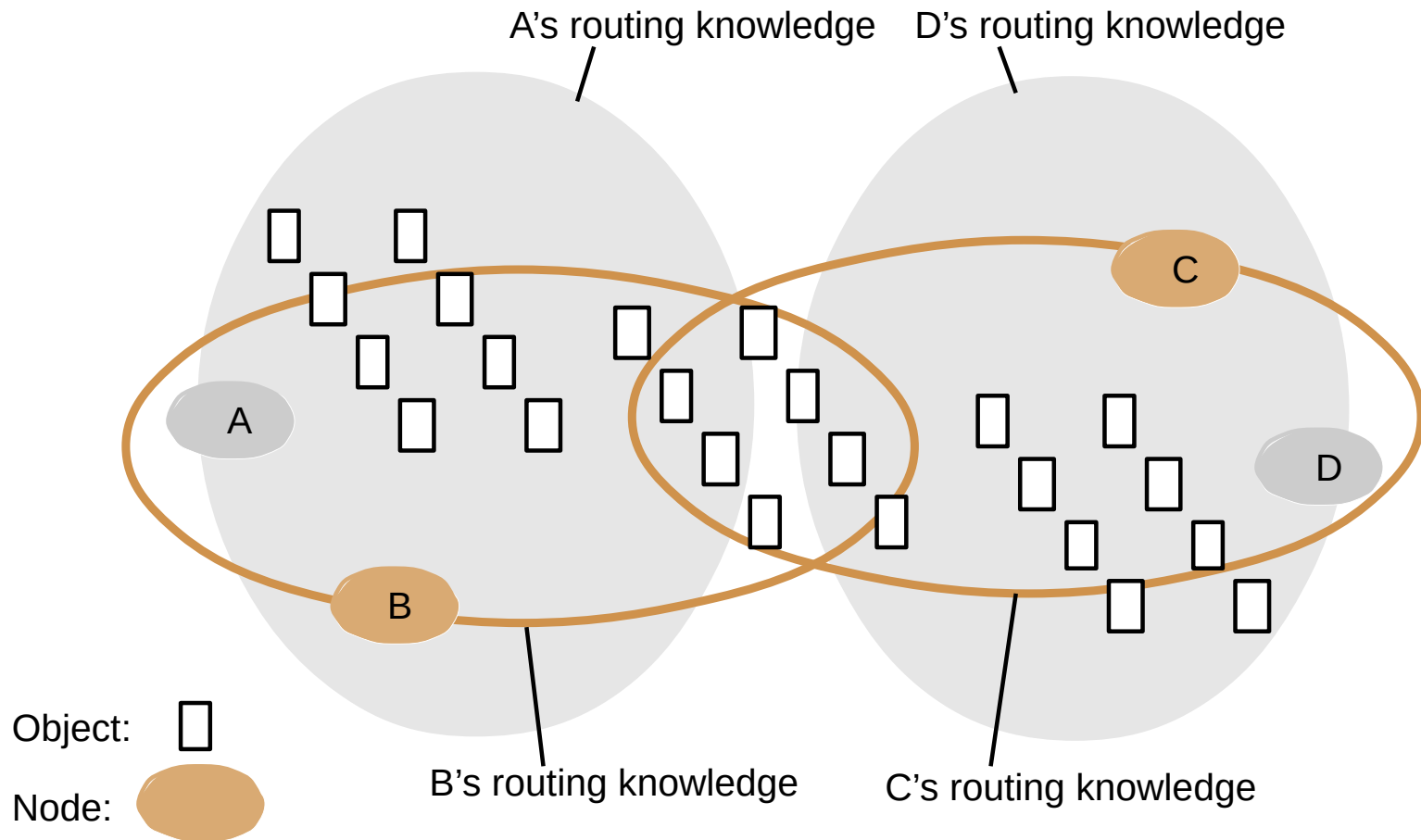


# Middlewares

---

- Peer-to-peer middlewares
  - Fornecer uma API simples de programação
  - Facilitar a localização, remoção e adição de objetos na rede
  - Balanceamento de carga
  - Aproximar aos nós os recursos mais acessados
  - Cada nó deve conhecer a localização dos objetos e nós e conhecer a topologia da rede.

# Middlewares



# Routing overlay

---

- Algoritmos de sobreposição de roteamento
  - Conhece toda a rede e roteia as requisições de objetos, baseado nos GUIDs
  - Inserção de objetos:
    - Gera um GUID e anuncia a chegada de um novo objeto à rede
    - Remoção de objetos
    - Adção e remoção de nós (redistribuindo as cargas)

# Estruturação

---

	<i>Structured peer-to-peer</i>	<i>Unstructured peer-to-peer</i>
<i>Advantages</i>	Guaranteed to locate objects (assuming they exist) and can offer time and complexity bounds on this operation; relatively low message overhead.	Self-organizing and naturally resilient to node failure.
<i>Disadvantages</i>	Need to maintain often complex overlay structures, which can be difficult and costly to achieve, especially in highly dynamic environments.	Probabilistic and hence cannot offer absolute guarantees on locating objects; prone to excessive messaging overhead which can affect scalability.

---

# Busca

---

- Estratégias de busca de arquivos numa rede não estruturada
  - **Busca em anel expandida**: os nós enviam pacotes com valores crescentes no campo TTL;
  - **Caminhadas aleatórias**: os nós enviam alguns pacotes que seguem caminhos aleatórios pela rede;
  - **Fofoca**: os pacotes são enviados a nós vizinhos com maior probabilidade de ter o recurso procurado. O processo continua recursivamente.

# BitTorrent

---

- Aplicação para o compartilhamento de **arquivos grandes**;
- Não se destina a streaming de vídeos em tempo real;
- Sua principal característica é dividir um arquivo em várias partes do mesmo tamanho;

# BitTorrent

---

- Quando um objeto é disponibilizado no BitTorrent, é criado um arquivo *.torrent* com:
  - 1) O nome e o tamanho do arquivo;
  - 2) A localização de um *tracker*;
  - 3) Um checksum associado a cada parte do arquivo
- O protocolo prioriza quem contribui mais para a rede.



# BitTorrent

---

<i>Term</i>	<i>Meaning</i>
<i>.torrent</i> file	A file that maintains metadata about an available file
tracker	A server containing information about the downloads in progress
chunk	A fixed size portion of a given file
seeder	A peer that holds a complete copy of a file (consisting of all its chunks)
leecher	A peer involved in downloading a file that currently holds only a portion of its chunks
torrent (or swarm)	A set of sites involved with downloading a file including the tracker, seeders and leechers
tit-for-tat	An incentive mechanism that governs the scheduling of downloads in BitTorrent
optimistic unchoking	A mechanism to allow new peers to establish their credentials
rarest first	A scheduling scheme whereby BitTorrent prioritizes frames that are rare within its set of connected peers