

# Blockchain, Criptomoedas & Tecnologias Descentralizadas

## Tecnologias descentralizadas: Compartilhamento de arquivos

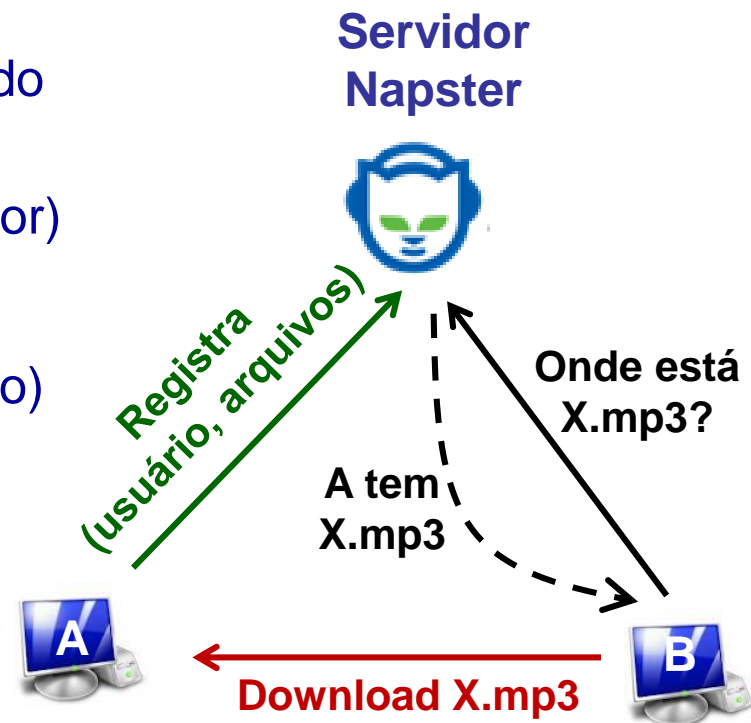
Prof. Dr. Marcos A. Simplicio Jr. – [mjunior@larc.usp.br](mailto:mjunior@larc.usp.br)  
Escola Politécnica, Universidade de São Paulo

# Objetivos

- Entender o processo de evolução das aplicações de compartilhamento de arquivos P2P
  - **Napster**: centralizado
  - **Gnutella**: descentralizado, não estruturado
  - **Kazaa**: super-nós
  - **Bittorrent**: descentralização, controle de integridade, DHTs

# Napster

- Distribuição de arquivos MP3 via P2P
  - Híbrido descentralizado, não estruturado
- Combinação de abordagens cliente/servidor e P2P
  - Rede de usuários registrados executando software cliente
  - Comunicação c/ diretório central (servidor)
- Servidor mantém 3 tabelas
  - (Indexador\_Arquivo, Metadados\_Arquivo)
  - (ID\_Usuário, Info\_Usuário)
  - (ID\_Usuário, Indexador\_Arquivo)



# Napster (cont.)



- Um pouco de história:
  - O Napster foi inventado em 1999 por Shawn Fanning, músico da Northeastern University, para **compartilhar músicas** no campus.
  - Primeiros processos por **infração de direitos autorais** de grandes gravadoras apareceram em Dezembro/1999
  - Sua **popularidade cresceu** imensamente até 2001, quando seu **fechamento** foi ordenado pela justiça americana.
  - Multa: U\$ 26 milhões por **danos** passados e **futuros**,
  - Declaração de **falência** da empresa em 2002
  - O nome “Napster” retornou em 2003 como uma **loja de músicas** online (nada a ver com P2P)...
- Principais fraquezas:
  - Servidor central: único ponto de falha (e alvo de ataques legais...)
  - Resultados pouco confiáveis (dados entrados por usuário)



# Gnutella

- Um pouco de história
  - Gnutella apareceu pouco depois do Napster
  - **Resolve** alguns dos **problemas do Napster** (mas **introduz outros**)
  - Protocolo de **especificação aberta**
  - Originalmente desenvolvida pela Nullsoft (posteriormente comprada pela AOL)
  - Versão 0.4 (= Gnutella original) será coberta aqui
    - Embora tenha sido substituída por **versão semelhante ao KaZaa**, discutido mais adiante

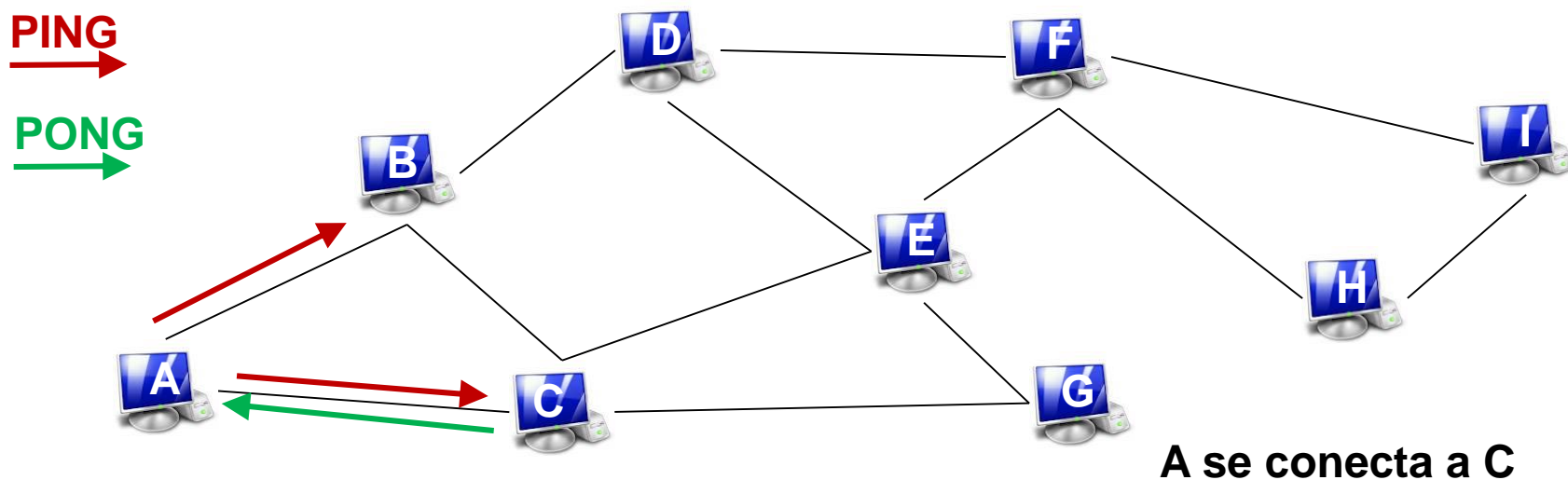
**gnutella**

# Gnutella (cont.)

- **Puramente descentralizado, não-estruturado**
- **Características:**
  - Poucos nós com alta conectividade
  - A maioria dos nós com conectividade esparsa
- **Cada instância da aplicação (nó):**
  - Armazena/distribui arquivos
  - Roteia mensagens de busca para seus vizinhos
  - Responde a requisições de arquivo

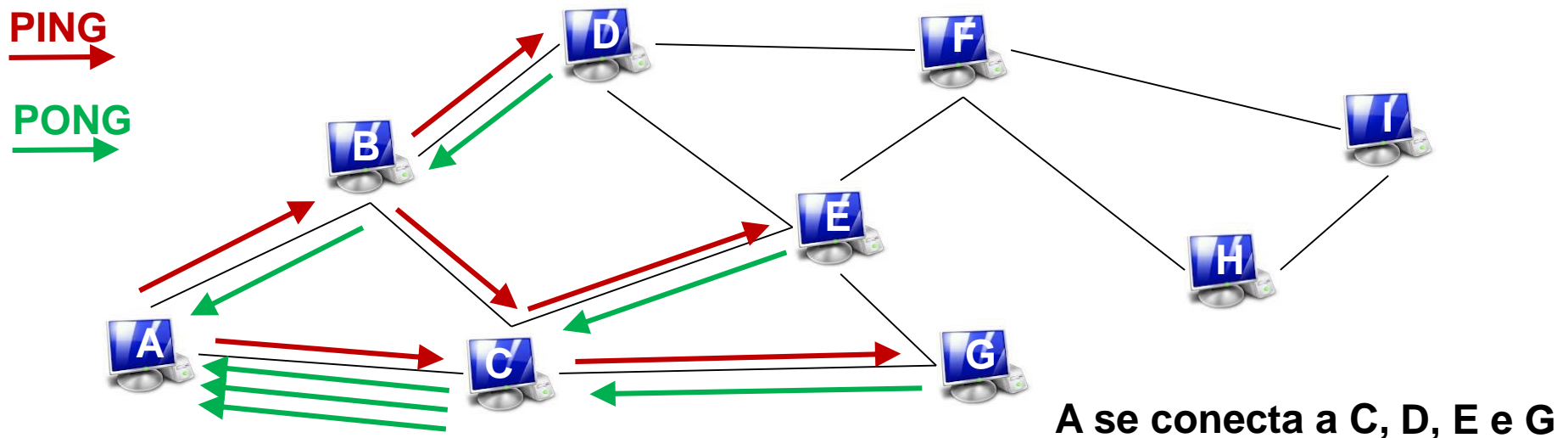
# Gnutella (cont.)

- Juntando-se à rede (formação de conexões lógicas):
  - Mensagem de **PING** enviada a nó que já está na rede: lista obtida “fora de banda”, por exemplo, via uma página web.
  - Mensagens de PING **encaminhadas** a vizinhos **via inundação**
  - Nós podem **responder** com mensagens de **PONG** contendo endereço do nó e outras informações relevantes



# Gnutella (cont.)

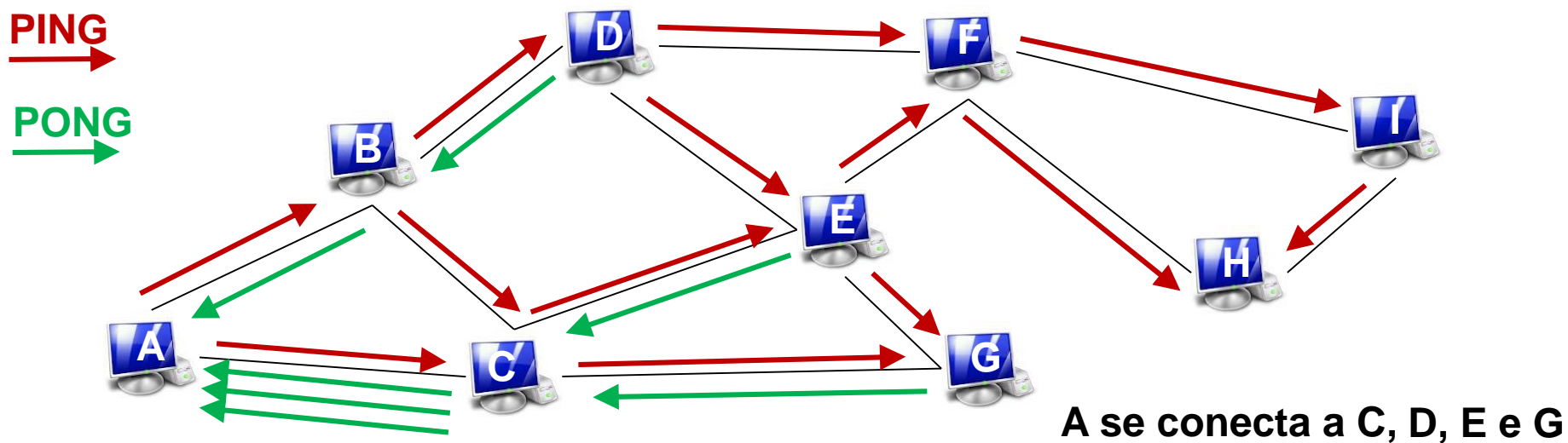
- Juntando-se à rede (formação de conexões lógicas):
  - Mensagem de **PING** enviada a nó que já está na rede: lista obtida “fora de banda”, por exemplo, via uma página web.
  - Mensagens de PING **encaminhadas** a vizinhos **via inundação**
  - Nós podem **responder** com mensagens de **PONG** contendo endereço do nó e outras informações relevantes





# Gnutella (cont.)

- Juntando-se à rede (formação de conexões lógicas):
  - Mensagem de **PING** enviada a nó que já está na rede: lista obtida “fora de banda”, por exemplo, via uma página web.
  - Mensagens de PING **encaminhadas** a vizinhos **via inundação**
  - Nós podem **responder** com mensagens de **PONG** contendo endereço do nó e outras informações relevantes

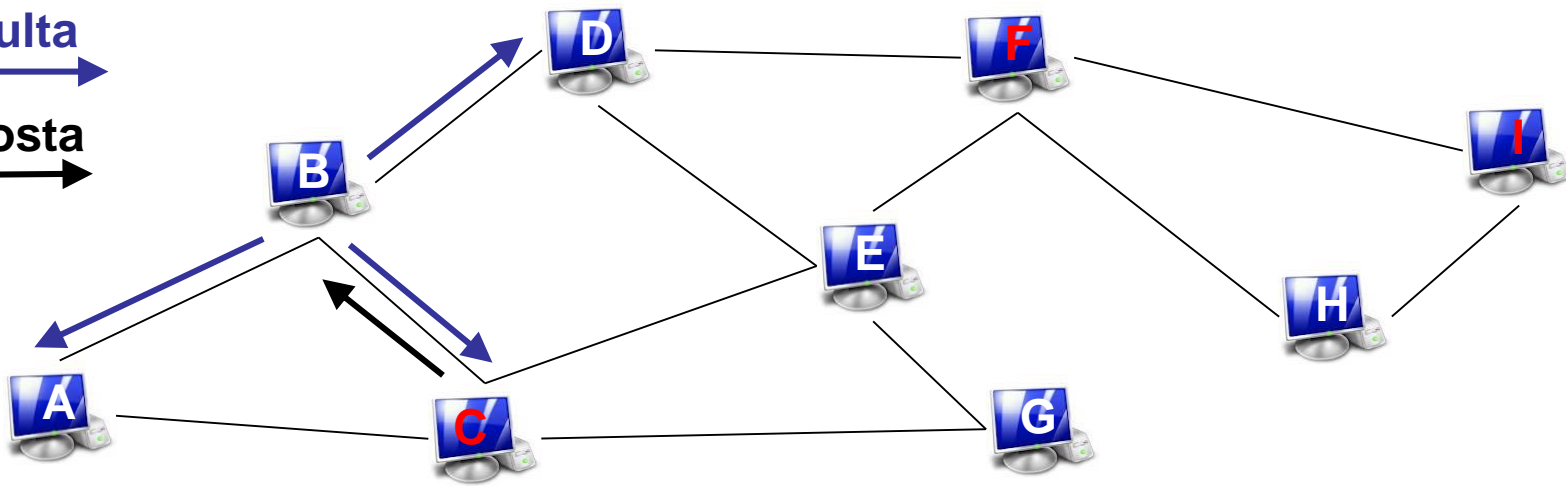


# Gnutella (cont.)

- **Busca:**

- Mensagens de consulta enviadas via **conexões TCP existentes**
- Peers encaminham mensagem (**inundação**) até que **TTL expira**
- Quando objeto é encontrado, **respostas** (IP + porta) são enviadas no **caminho reverso da consulta**

Consulta  
Resposta



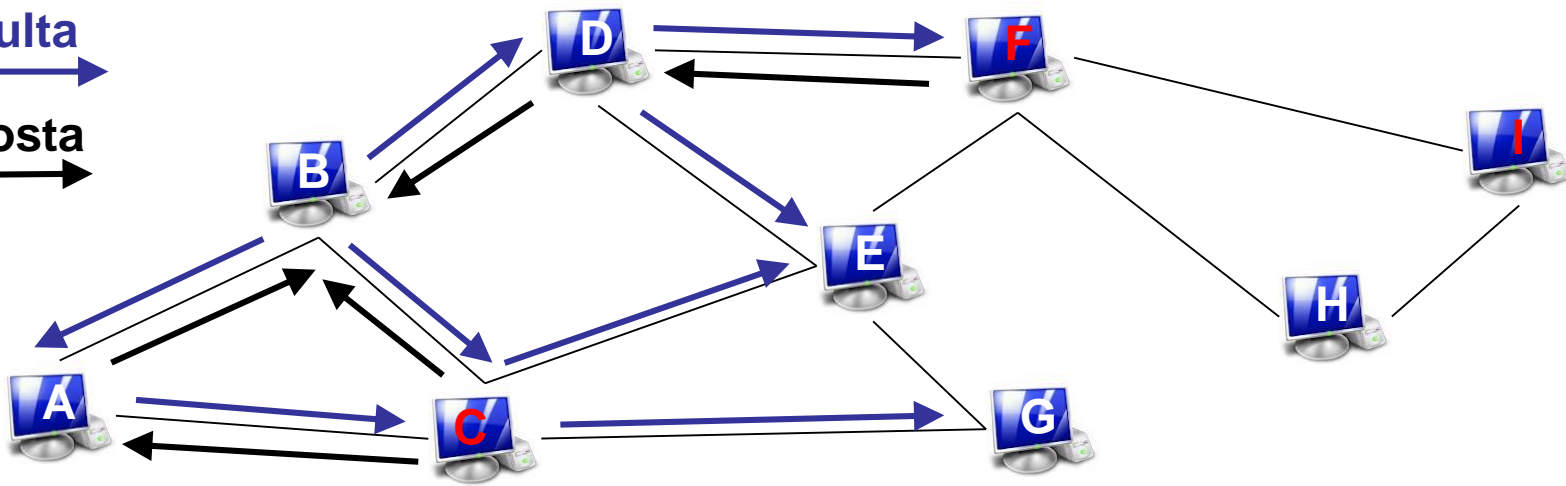
**C, F e I têm o conteúdo buscado por B**

# Gnutella (cont.)

- **Busca:**

- Mensagens de consulta enviadas via **conexões TCP existentes**
- Peers encaminham mensagem (**inundação**) até que **TTL expira**
- Quando objeto é encontrado, **respostas** (IP + porta) são enviadas no **caminho reverso da consulta**

Consulta  
Resposta



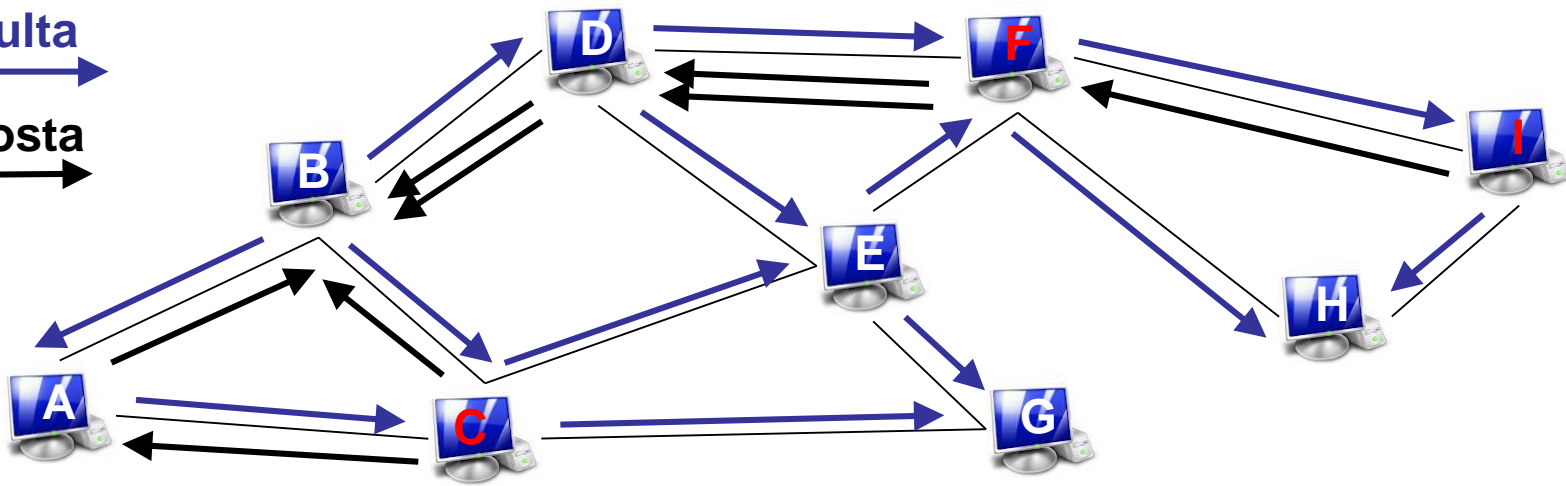
C, F e I têm o conteúdo buscado por B

# Gnutella (cont.)

- **Busca:**

- Mensagens de consulta enviadas via **conexões TCP existentes**
- Peers encaminham mensagem (**inundação**) até que **TTL expira**
- Quando objeto é encontrado, **respostas** (IP + porta) são enviadas no **caminho reverso da consulta**

Consulta  
Resposta



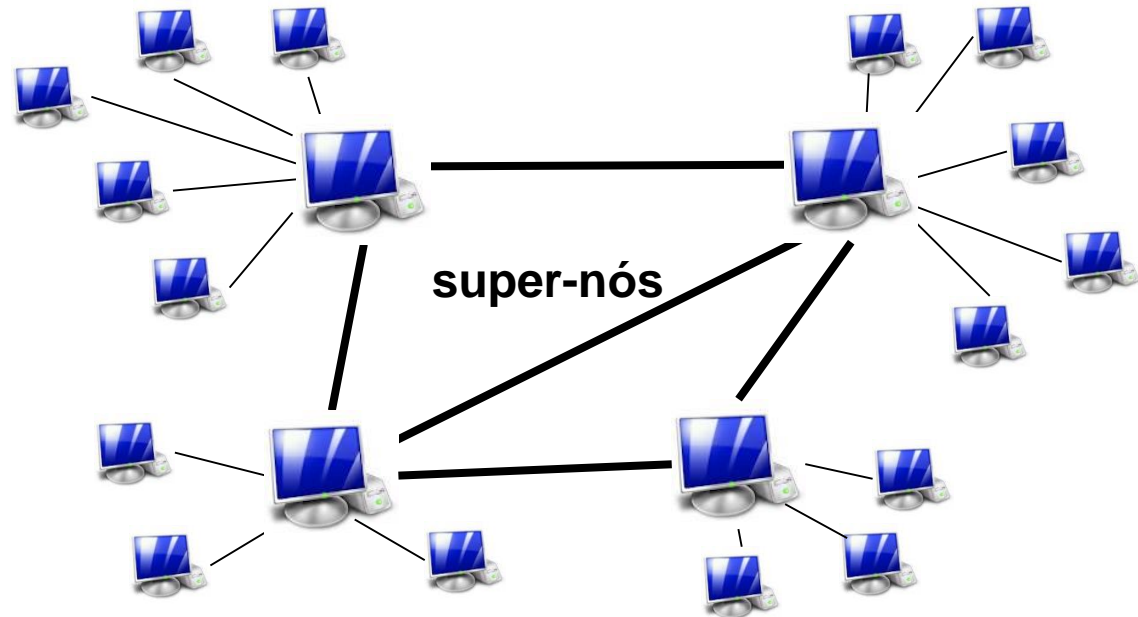
C, F e I têm o conteúdo buscado por B

# Gnutella (cont.)

- **Download:** diretamente entre nós envolvidos
  - Mensagens no estilo HTTP, na porta informada, para obter conteúdo
- **Vantagens:**
  - **Totalmente distribuída:** sem pontos centrais de falha
  - Prova-se que rede é bastante **robusta a falhas aleatórias** (nem tanto a falhas em nós “bem conectados”, comuns na rede)
- **Limitações:**
  - **PINGs/PONGs** periódicos consomem **muitos recursos**
  - **Inundação (limitada)** cria conflito entre **completude e eficiência:**
    - TTL baixo: buscas podem não encontrar dados
    - TTL alto: elevado consumo de recursos
  - Nenhum mecanismo contra **comportamento egoísta**

# KaZaA

- O KaZaA (ou Kazaa) surgiu em 2001 e rapidamente superou seus antecessores
  - Nova arquitetura, baseada em super-nós: parcialmente centralizada, não estruturada
  - Cada super-nó conhece diversos outros super-nós (malha quase completa)



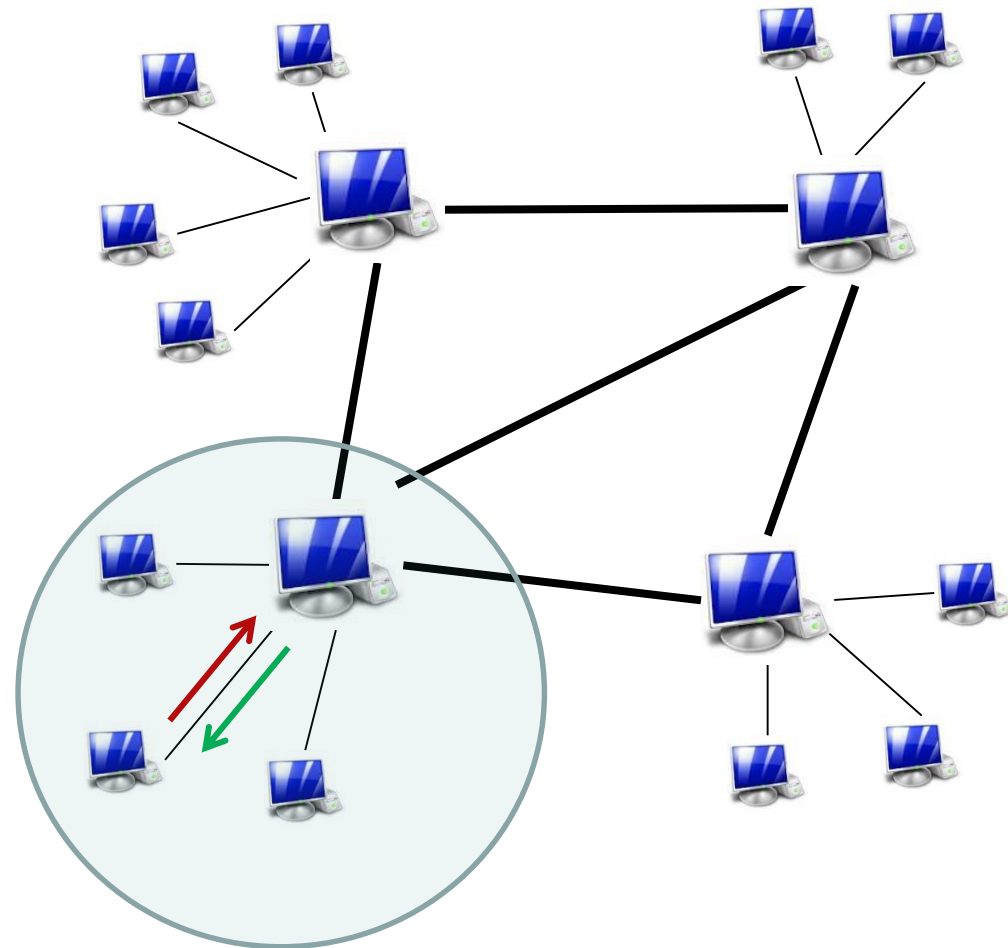
# KaZaA (cont.)

- **Super-nós:** peers normais, porém com mais recursos e responsabilidades
  - Usuários normalmente **podem se recusar** a tornar super-nós
  - Nó regular se **conecta a apenas um super-nó** em cada instante
  - Cada super-nó mantém **registro de todos os arquivos nos peers** aos quais está conectado (e apenas para esses peers)
  - **Super-nó atua como** uma espécie de “**hub**” **Napster** para todos os nós normais conectados a ele.
- **Conexões entre super-nós mudam** regularmente
  - Periodicidade de algumas dezenas de minutos

# KaZaA (cont.)

- **Montando a rede**

- Peer **obtêm endereço de super-nó** de alguma forma (e.g., site web ou incluso no software cliente)
- Peer enviam **requisição** a super-nó, informando **lista de arquivos** que deseja compartilhar
- Apenas aquele **super-nó mantém registro** desse novo peer

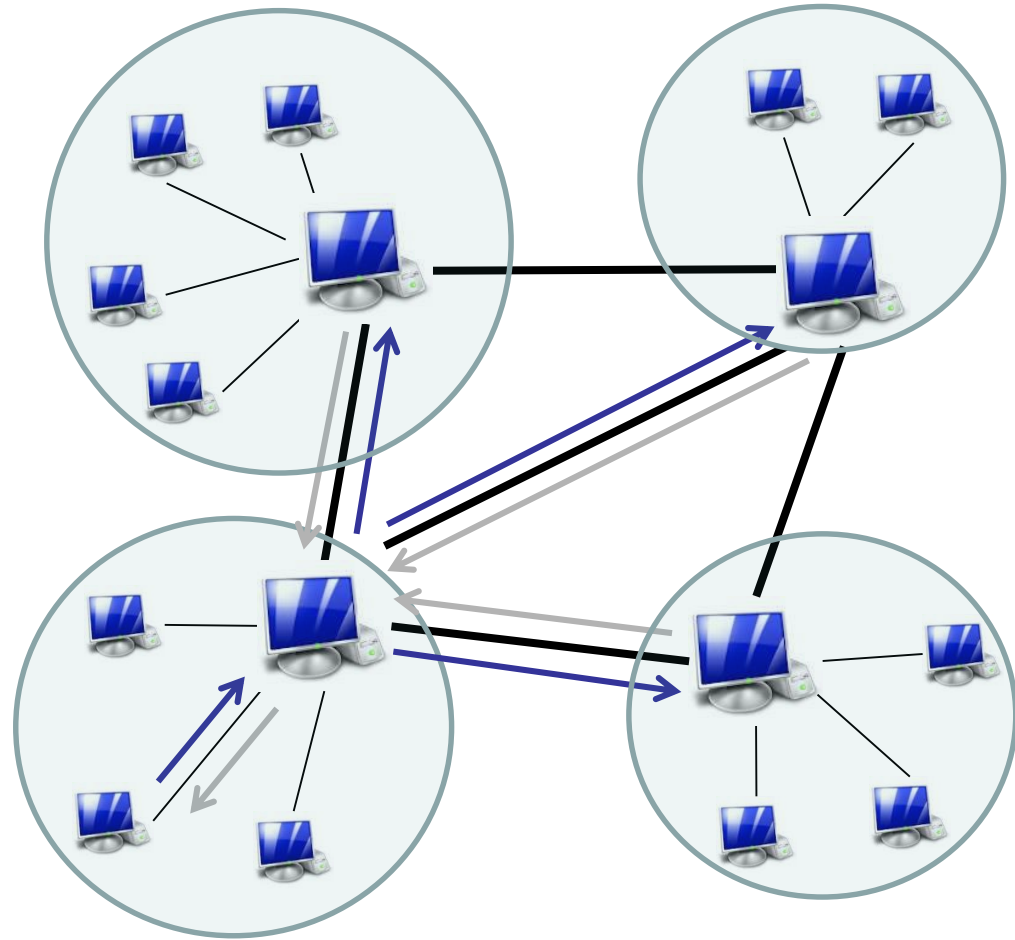




# KaZaA (cont.)

- **Buscando conteúdo**




- Peer envia **requisição** de conteúdo (e.g., palavra chave) a **seu próprio super-nó**
- Super-nó **responde** por todos os nós conectados a ele e **encaminha** pedido a outros **super-nós**
- Outros **super-nós respondem** pelos nós a ele conectados



# KaZaA (cont.)

- Vantagens
  - **Escalabilidade** (mais usuários = mais **super-nós**)
  - **Eficiência: inundação limitada** aos super-nós
  - Explora **heterogeneidade** dos nós
  - **Tolerância a falhas**
- Desvantagens
  - **Poluição** de conteúdos
  - Vulnerabilidade a ataques de **negação de serviço** (*Denial of Service* – DoS) contra super-nós

# BitTorrent

- Uso: distribuição de arquivos P2P
  - Ex.:  uTorrent,  qBittorrent,  Deluge
- Nova rede overlay criada para cada arquivo sendo distribuído
- Pode-se enviar “link” (arquivo .torrent) a um amigo
  - “Link” sempre se refere ao mesmo arquivo
    - Não é o caso de Napster, Gnutella, ou KaZaA: redes baseadas em buscas (difícil identificar arquivo específico)
  - Permite verificação de integridade (hash criptográfico)
  - Buscas não estão inclusas no protocolo, mas podem ser implementadas via sites web ou na interface de um aplicativo

# BitTorrent (cont.)

- Nomenclatura:



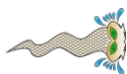
- **Tracker** (Rastreador): mantém lista de peers interessados em certo conteúdo



- **Piece** (Pedaço): Uma parte de um arquivo que está disponível na rede.



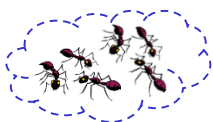
- **Seeders** (Semeadores): peers que têm o arquivo completo e continuam compartilhando-o (comportamento altruísta)



- **Leechers** (Sanguessugas): peers que têm apenas partes do arquivo e estão compartilhando e recebendo pedaços



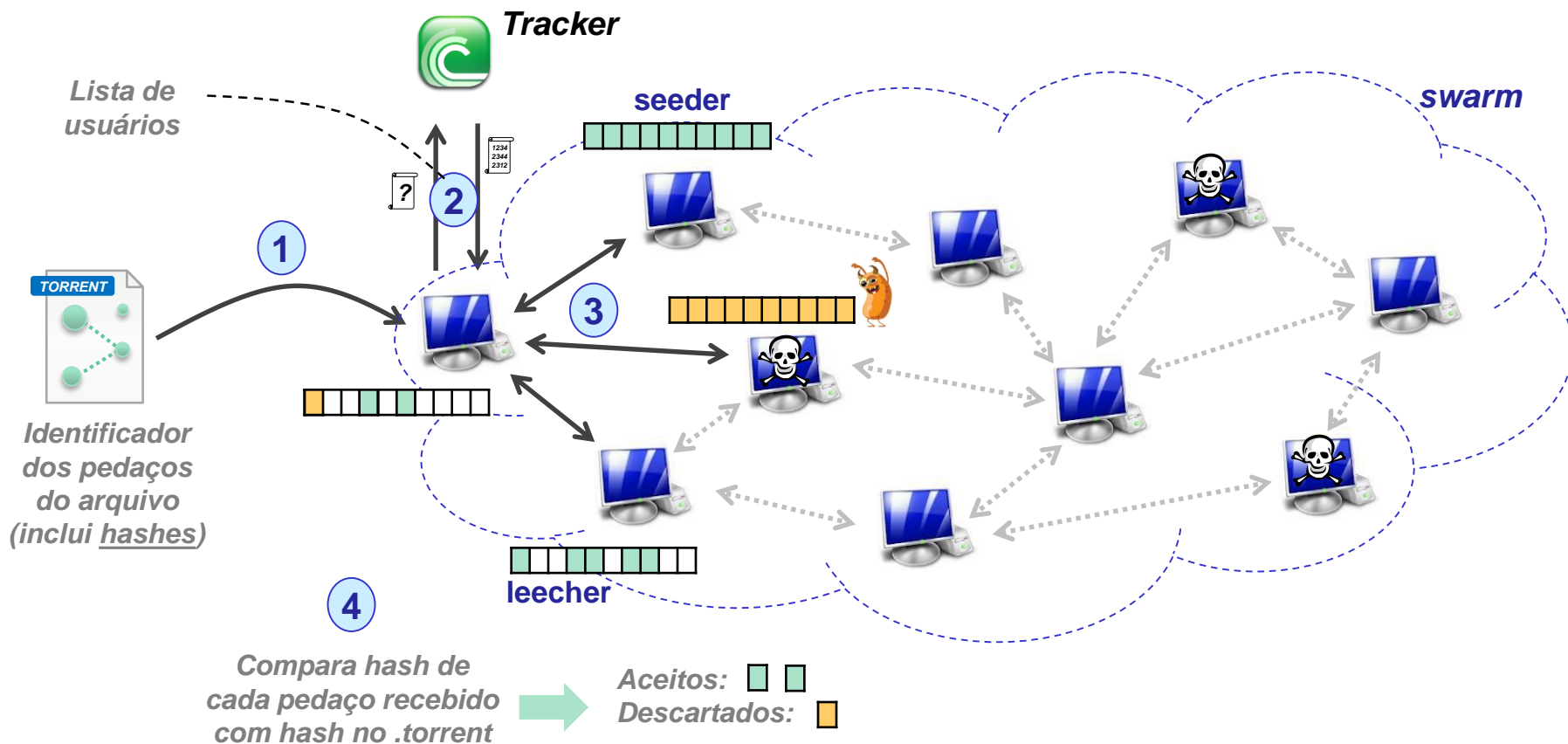
- **Arquivo .torrent**: metadados do arquivo



- **Swarm** (enxame): conjunto de peers que participam na distribuição de um determinado conteúdo.

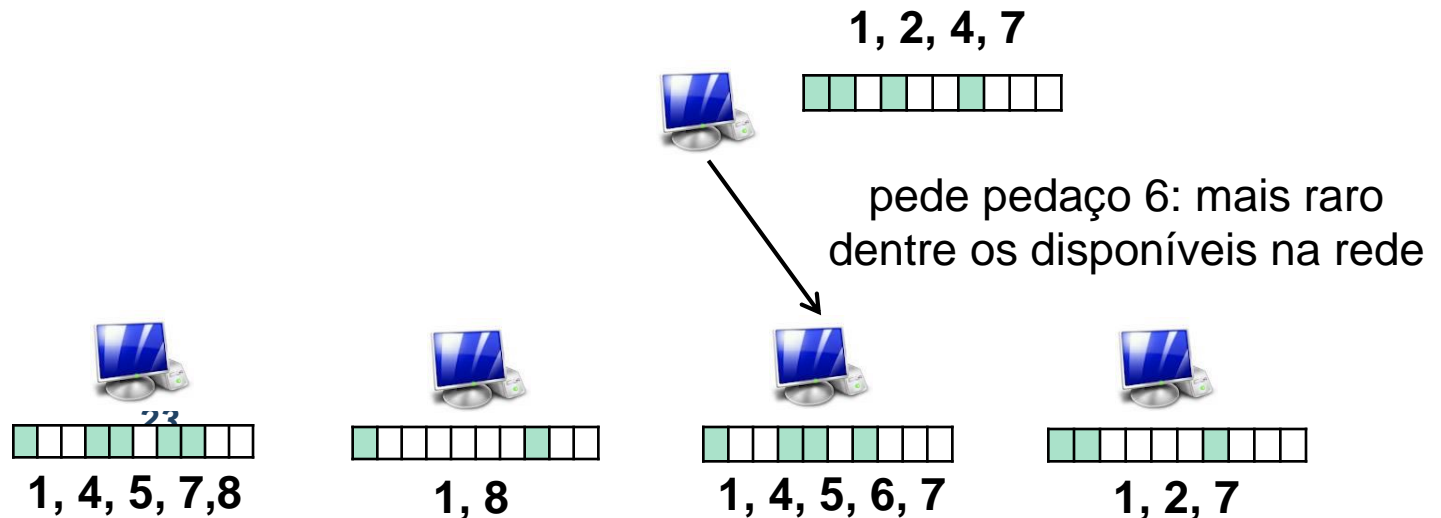
# BitTorrent (cont.)

- Funcionamento: com tracker



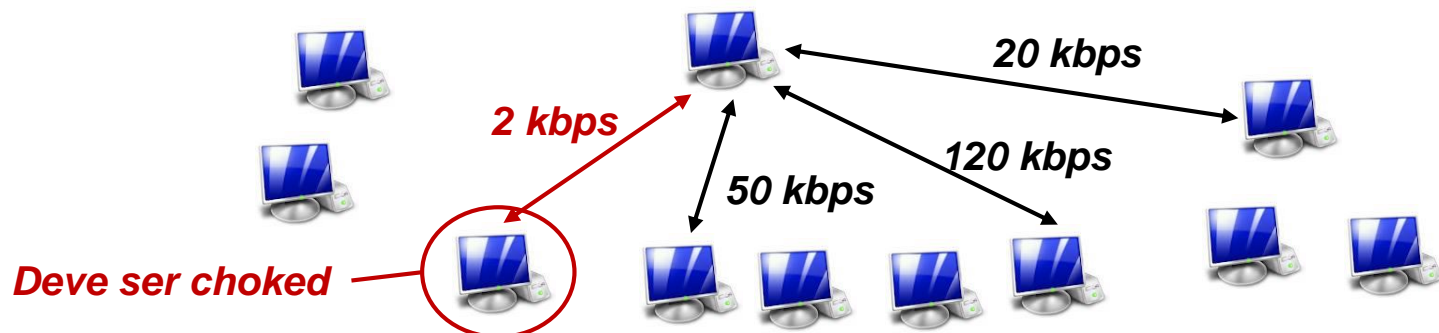
# BitTorrent: mecanismos

- Arquivo dividido em pedaços (comum: 256 KiB)
- Recebendo pedaços:
  - Periodicamente, cada peer pede a cada vizinho a lista de pedaços que eles têm;
  - Ele então envia requisições para os pedaços que faltam, dando prioridade àqueles com menor disponibilidade: **mais raros primeiro**



# BitTorrent: mecanismos (cont.)

- Enviando pedaços:
  - Cada nó envia pedaços a  $n$  (comum: 4) vizinhos atuais, dando preferência àqueles que estão fornecendo pedaços com maior velocidade: “**tit-for-tat**”, ou “**olho por olho**”
    - Diz-se que os peers neste grupo estão “unchoked” (não estrangulado)
  - Reavalia grupo unchoked a cada  $t$  (comum: 10) segundos
  - Seleciona um nó aleatoriamente a cada  $t_c$  (comum: 30) segundos e o coloca no grupo unchoked, substituindo nó com menor velocidade: **optimistic unchoke**
    - Diz-se que o nó removido foi “choked” (estrangulado).



# BitTorrent: mecanismos (cont.)

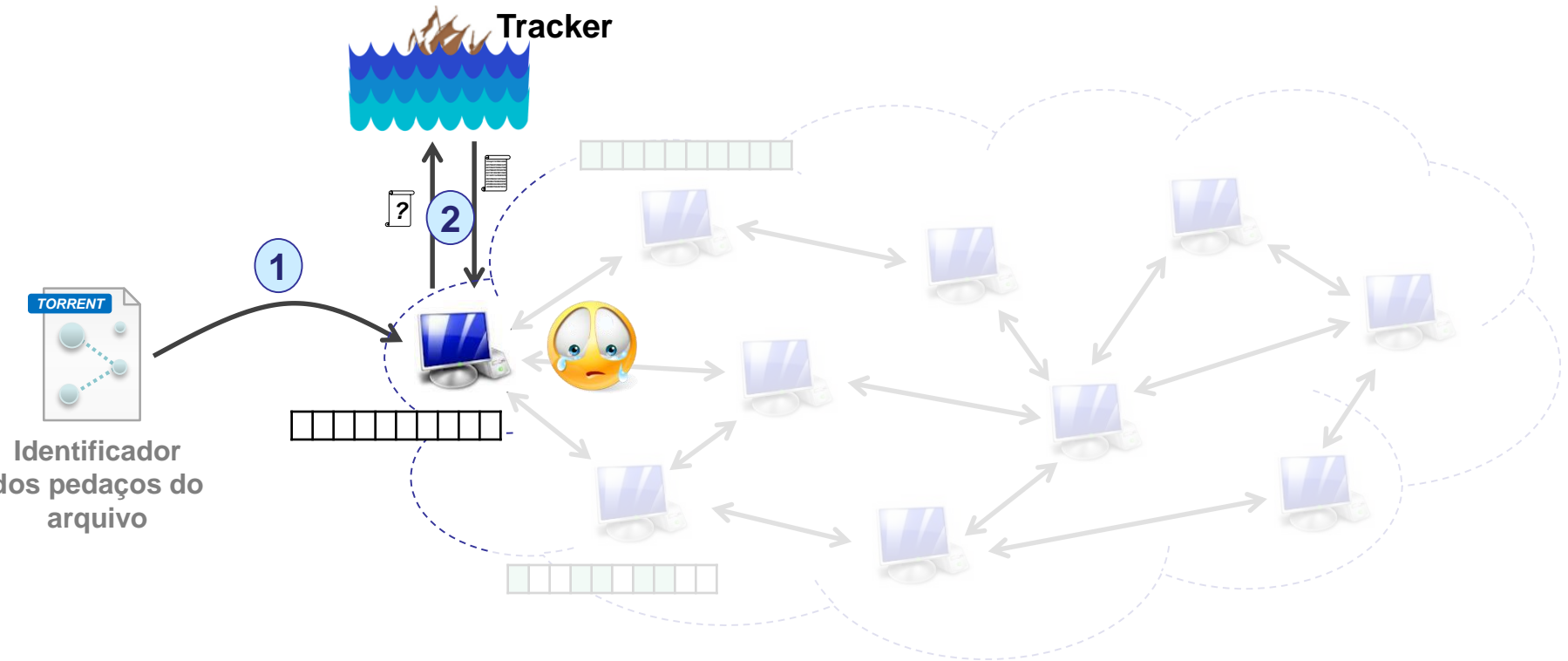
- Resultado das políticas do BitTorrent
  - Peers servem peers que os servem em retorno: peers com **capacidades semelhantes** (i.e., banda) tendem a interagir
  - Encoraja **cooperação**, desencoraja free-riding: free-riders são “choked” após algum tempo
  - “Rarest first” não apenas ajuda a **manter arquivo na rede**, mas também **incentiva colaboração** com novos peers: eles recebem primeiro os pedaços mais raros!
  - **Conexão a diversos peers** ao mesmo tempo pode levar a **“chokes” frequentes**: divisão da banda disponível entre os peers conectados
  - **Seeders** também têm política de “tit-for-tat”, mas observam a **taxa de download** do leecher ao invés da taxa de upload.
    - Preferência por leecher fazendo bastante download





# BitTorrent sem tracker: disponibilidade

- Tracker essencial para busca de peers...
  - E se tracker sair do ar...?



# BitTorrent sem tracker: disponibilidade

- BitTorrent sem tracker:
  - Distributed Hash Table (**DHT**): peers se organizam de forma que um auxilia o outro na busca por arquivos
    - Nota: 1º peer obtido via tracker, cache local, ou servidor web
  - **Links magnéticos**: usa DHT para obter arquivo .torrent, antes de iniciar download do arquivo em si
  - Peer Exchange (**PEX**): peers conectados a um nó qualquer fornecem listas de nós aos quais eles estejam conectados



- Maior disponibilidade

- Resistência a censura/ações legais
- Resistência a ataques de negação de serviço



# BitTorrent: anonimato?



- Apenas de **quem** gerou conteúdo
  - Nós trocando pedaços enxergam endereços IP uns dos outros: **sem anonimato dos nós!**



- Monitoramento possível:

- Entrar na rede e **anunciar que possui conteúdo**
- **Opcional:** Enviar pedaços falsos (descartados pelo receptor)
- **Coletar IPs** dos nós que tentarem obter o conteúdo
- **Mapear IP-usuário** com ajuda de operadora de Internet

- **Privacidade** possível se usado **com VPNs**

- Não use Tor: <https://blog.torproject.org/bittorrent-over-tor-isnt-good-idea>



# BitTorrent: anonimato?



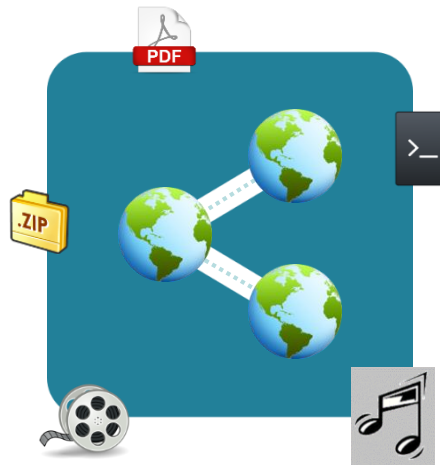
<https://canaltech.com.br/pirataria/usuarios-de-torrent-no-brasil-voltam-a-receber-notificacoes-extrajudiciais-182223/>



<https://olhardigital.com.br/2019/01/18/noticias/pf-derruba-site-de-torrents-brasileiro-como-parte-da-operacao-copyright/>



<https://www.diariodaregiao.com.br/cidades/policia/policia-investiga-600-suspeitos-de-pornografia-infantil-na-regi-o-de-rio-preto-1.817865>

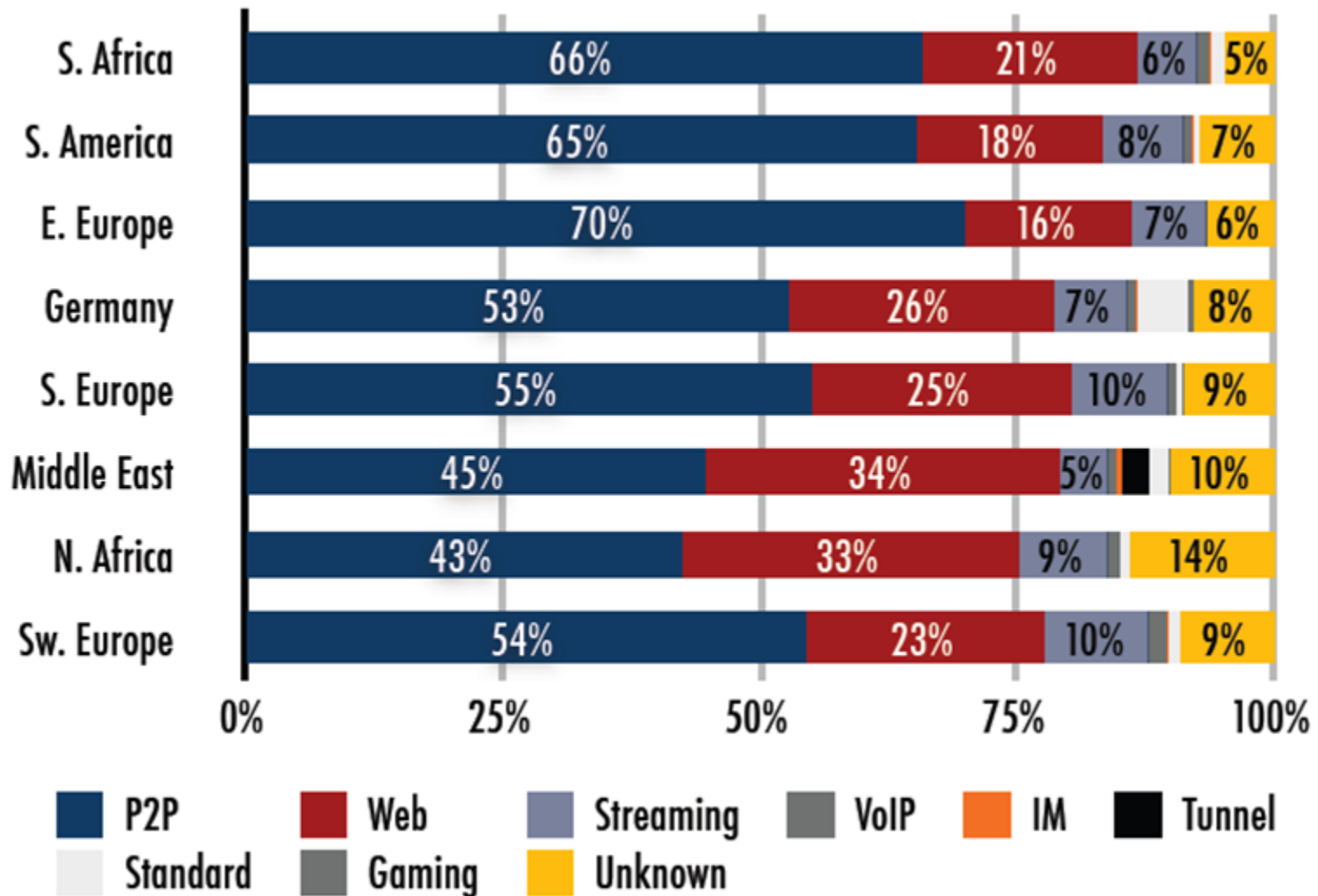


# Blockchain, Criptomoedas & Tecnologias Descentralizadas

## Tecnologias descentralizadas: Compartilhamento de arquivos

Prof. Dr. Marcos A. Simplicio Jr. – [mjunior@larc.usp.br](mailto:mjunior@larc.usp.br)  
Escola Politécnica, Universidade de São Paulo

# Internet: Distribuição de tráfego 2008/2009



\*Streaming: crescimento mais expressivo desde então

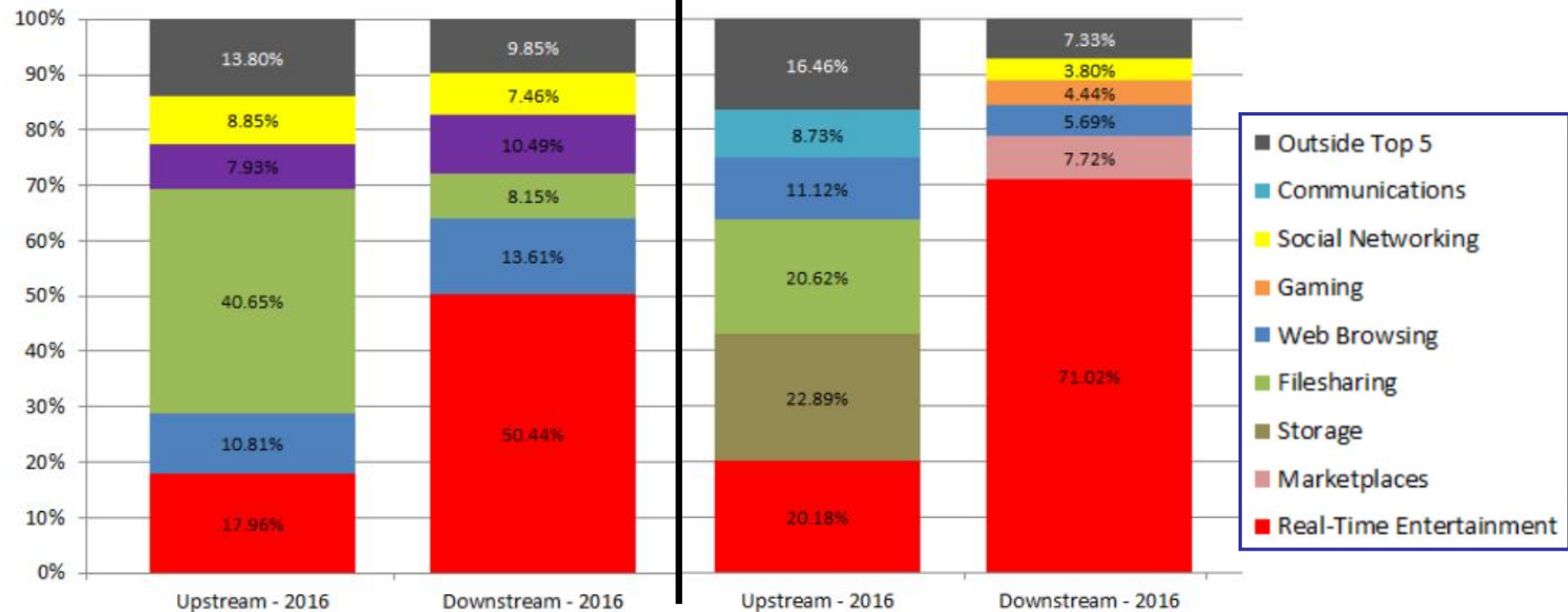
Fonte: IPOQUE

# Internet: Distribuição de tráfego 2016

- Tráfego em período de pico (redes fixas)
  - Fonte: Sandvine Global Internet Phenomena 2016

## América do Sul

## América do Norte



# Internet: Distribuição de tráfego 2016

- Tráfego em período de pico (redes fixas) – Principais aplicações
  - Fonte: Sandvine Global Internet Phenomena 2016

## América do Norte

P2P



Upstream		Downstream		Aggregate	
BitTorrent	18.37%	Netflix	35.15%	Netflix	32.72%
YouTube	13.13%	YouTube	17.53%	YouTube	17.31%
Netflix	10.33%	Amazon Video	4.26%	HTTP - OTHER	4.14%
SSL - OTHER	8.55%	HTTP - OTHER	4.19%	Amazon Video	3.96%
Google Cloud	6.98%	iTunes	2.91%	SSL - OTHER	3.12%
iCloud	5.98%	Hulu	2.68%	BitTorrent	2.85%
HTTP - OTHER	3.70%	SSL - OTHER	2.53%	iTunes	2.67%
Facebook	3.04%	Xbox One Games	2.18%	Hulu	2.47%
FaceTime	2.50%	Facebook	1.89%	Xbox One Games	2.15%
Skype	1.75%	BitTorrent	1.73%	Facebook	2.01%
	69.32%		74.33%		72.72%



# Internet: Distribuição de tráfego 2016

- Tráfego em período de pico (redes fixas) – Principais aplicações
  - Fonte: Sandvine Global Internet Phenomena 2016

## América do Sul

P2P	Upstream		Downstream		Aggregate	
	→	BitTorrent	30.03%	YouTube	28.48%	YouTube
	YouTube	9.30%	HTTP - OTHER	11.66%	HTTP - OTHER	11.12%
	HTTP - OTHER	7.59%	SSL - OTHER	9.76%	BitTorrent	10.06%
	Facebook	6.72%	Netflix	8.31%	SSL - OTHER	9.28%
	SSL - OTHER	6.19%	BitTorrent	6.96%	Netflix	7.45%
→	Ares	5.27%	Facebook	5.10%	Facebook	5.32%
	Skype	2.53%	MPEG - OTHER	2.28%	MPEG - OTHER	2.10%
	Netflix	1.97%	RTMP	1.79%	RTMP	1.66%
	Dropbox	1.16%	Google Market	1.69%	Google Market	1.52%
	MPEG - OTHER	0.92%	Flash Video	1.60%	Flash Video	1.46%
		71.69%		77.63%		75.87%

# Internet: Distribuição de tráfego 2018

Video is almost **58%** of the total downstream volume of traffic on the internet


**NETFLIX** is **15%** of the total downstream volume of traffic across the entire internet

More than **50%**



of internet traffic is encrypted, and TLS 1.3 adoption is growing

Alphabet / **Google** applications make up over **40%** of the total internet connections in APAC

 BITTORRENT is almost **22%** of total upstream volume of traffic, and over **31%** in EMEA alone

## GAMING



is becoming a significant force in traffic volume as gaming downloads, Twitch streaming, and professional gaming go mainstream

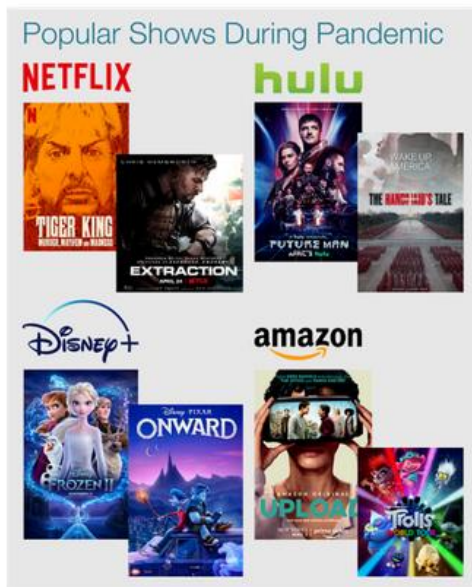
Plus spotlights on:

Traffic share leaders for video, social networking, messaging, audio streaming, and gaming



Fonte: <https://www.sandvine.com/hubfs/downloads/phenomena/2018-phenomena-report.pdf>

# Internet: Distribuição de tráfego 05/2020

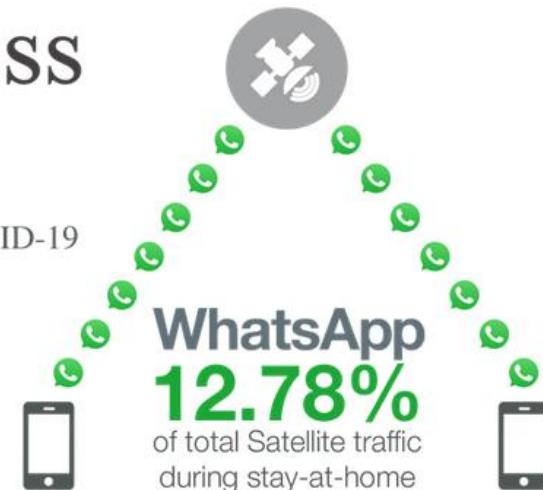


11% OF GLOBAL TRAFFIC IS  
**NETFLIX**



during the worldwide stay-at-home orders

WORDPRESS  
is **4.93%** of all traffic in EMEA during COVID-19



Fonte: <https://www.sandvine.com/phenomena>

# References

- Ding, C. H., Nutanong, S., & Buyya, R. (2004). P2P Networks for Content Sharing. *arXiv preprint cs/0402018*. URL: <https://arxiv.org/abs/cs/0402018>
- Liang, J., Kumar, R., & Ross, K. W. (2004). Understanding kazaa. URL: <http://pages.di.unipi.it/ricci/kazaa.pdf>
- Cohen, B. (2003). Incentives build robustness in BitTorrent. In Workshop on Economics of Peer-to-Peer systems (Vol. 6, pp. 68-72). URL: <https://www.cs.swarthmore.edu/~newhall/readings/bittorrentecon.pdf>
- Wolchok, S., & Halderman, J. A. (2010). Crawling BitTorrent DHTs for Fun and Profit. In 4th USENIX Workshop on Offensive Technologies (WOOT 10). URL: [https://www.usenix.org/legacy/event/woot10/tech/full\\_papers/Wolchok.pdf](https://www.usenix.org/legacy/event/woot10/tech/full_papers/Wolchok.pdf)