Anatomia do BitTorrent: a Ciência da Computação por trás do protocolo

Paulo Cheadi Haddad Filho — ~paulochf
Orientador: José Coelho de Pina — ~coelho

Objetivo

- **estudar** o protocolo do BitTorrent
- **identificar** seu caráter interdisciplinar na Computação pelo estudo de seus componentes internos
- **ilustrar** a implementação desses componentes utilizando o código-fonte do cliente Transmission

Introdução

- **redes peer-to-peer (P2P)**: redes de arquitetura descentralizada (sem um servidor central) e distribuída entre vários nós da rede (*peers*)
- **Napster** foi a primeira rede P2P, em 1999

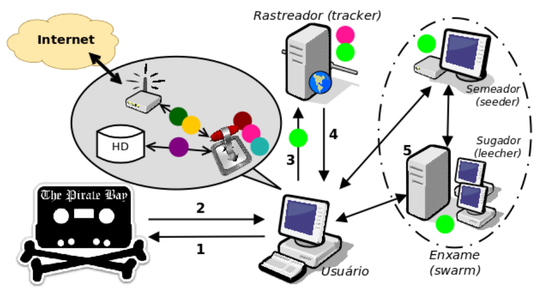
História do BitTorrent

- lançado por Bram Cohen em **2001**
- protocolo P2P **mais usado no mundo**, que gera ≈23% do tráfego de upload, ≈17% de download e ≈10% de todo o tráfego na América Latina [1]
- usado por **Twitter** [2] e **Facebook** [3] para distribuir os códigos dos seus sites para seus servidores
- baseado em **trocas justas** de arquivos e **comunicação eficiente** entre *peers*
- peers **sugadores** (*leechers*) e **semeadores** (*seeders*), pertencentes a um **enxame** (*swarm*), **trocam partes** de um torrent entre si
- **listas de peers** mantidas por **rastreadores** (*trackers*) e, geralmente, pelos próprios *peers*

Transmission

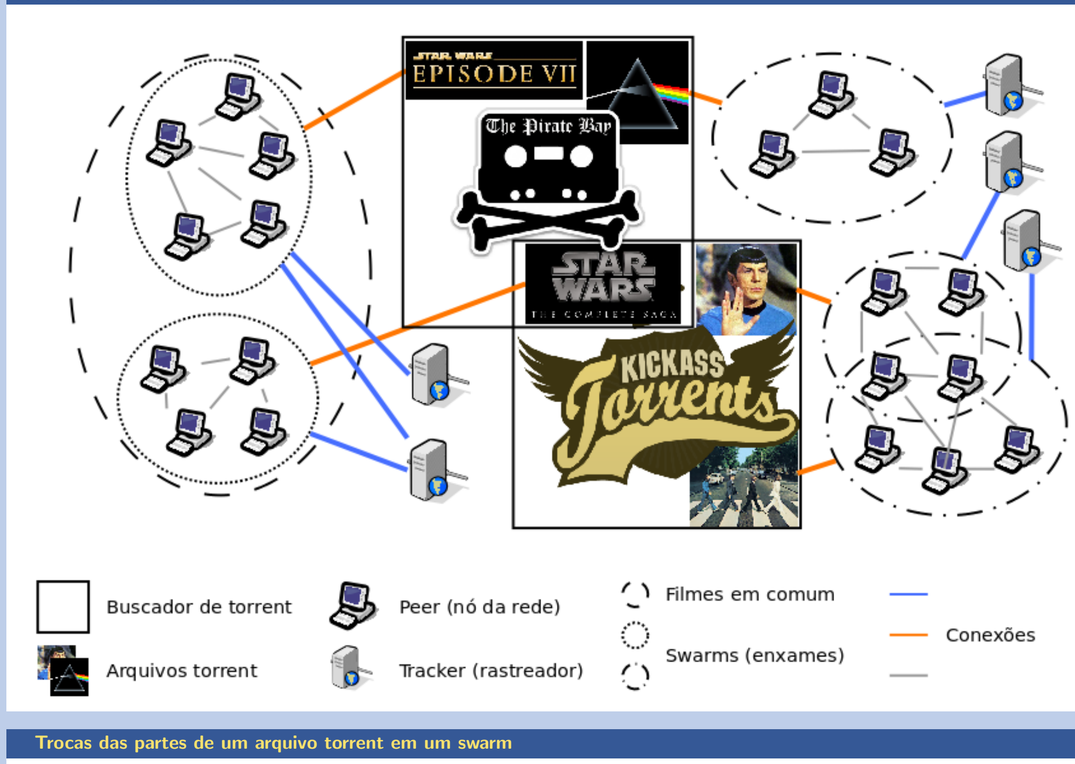
- **programa cliente** de **código aberto** para o protocolo BitTorrent
- escrito nas linguagens **C** e **C++**
- **várias plataformas**: *daemon* (serviço de segundo plano), roteadores, linha de comando e aplicação em janela

BitTorrent: visão do usuário e áreas da Computação

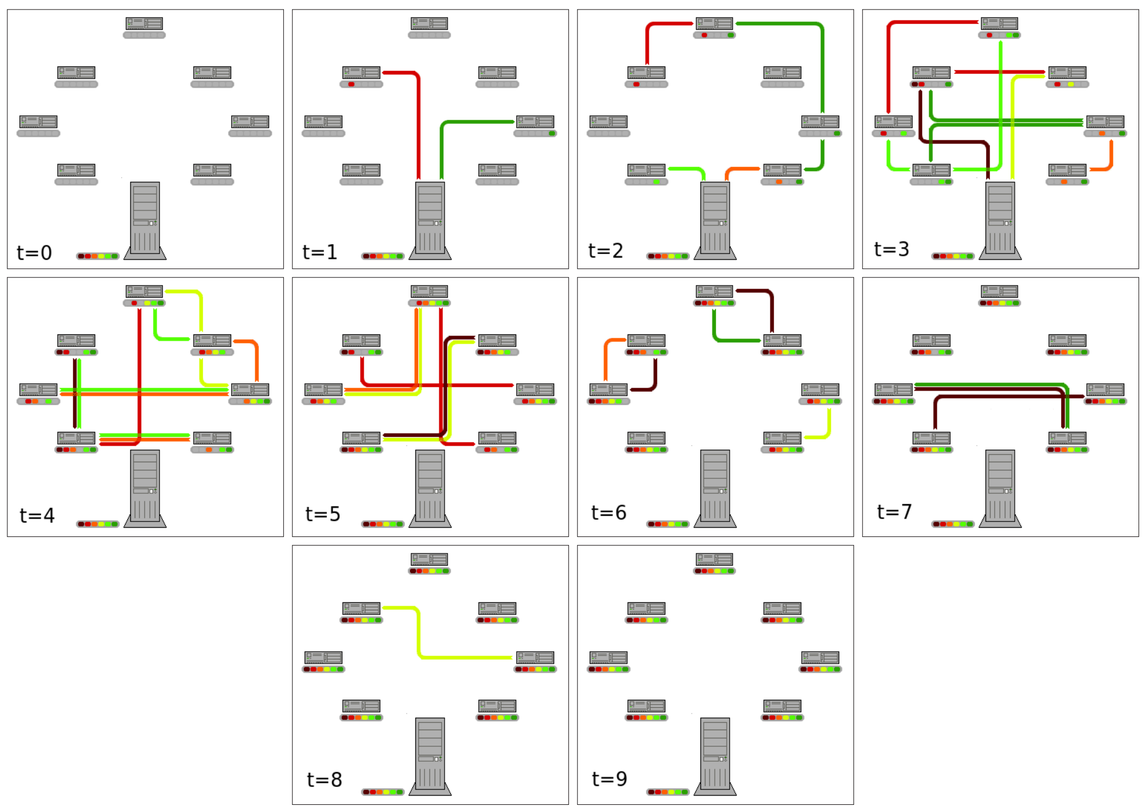


1. busca de conteúdo em sites buscadores de torrent
2. obtenção do arquivo .torrent desejado
3. computador do usuário se comunica com rastreadores (*trackers*), que mantêm listas dos *peers* que estão compartilhando os arquivos do torrent obtido
4. o *tracker* devolve uma lista de *peers* aleatória
5. computador do usuário inicia comunicação com os *peers* da lista, e começa a receber deles os arquivos pertencentes àquele torrent

BitTorrent: visão da Internet



Trocas das partes de um arquivo torrent em um swarm



Fonte: Wikipedia — `http://en.wikipedia.org/wiki/File:Torrentcomp_small.gif`

Funcionamento do BitTorrent

1. **busca** de conteúdo em sites buscadores de torrent

2. **obtenção** do arquivo .torrent, que é um **dicionário de dados** sobre os arquivos desejados, onde estão contidas informações como *announces* (endereços de *trackers*), número de partes e lista dos arquivos

3. **cliente BitTorrent** usa o *announce* para fazer um pedido de *peers*, usando uma **requisição HTTP GET** e passando dois identificadores: um seu e o do torrent

Ex: `http://tracker.publicbt.com:80/announce?info_hash=<hash-do-torrent>&peer_id=<hash-do-cliente>`

4. o **tracker adiciona** o *peer* requisitante à sua lista e **devolve** para ele um dicionário contendo: uma **lista aleatória de outros peers** para aquele identificador de torrent, quantidades de *seeders* e *leechers*, entre outros dados
Ex: {'complete': 1, 'downloaded': 11, 'incomplete': 6, 'interval': 1732, 'min interval': 866, 'peers': {lista-de-hash-ids-de-peers}}

5. o **cliente**, para entrar no *swarm*, **envia mensagens** para cada um dos *peers* recebidos, até obter uma resposta

5.1 quando **obtém resposta**, **adiciona o endereço** desta à sua "lista de contatos", que é acessível por outros *peers*, e implementada como uma tabela hash especial, do algoritmo Kademlia. Esse algoritmo faz com que uma busca por *novos peers* seja distribuída entre os que já são conhecidos, formando assim uma grande "lista telefônica". Essa estrutura de dados é chamada de **tabela hash distribuída (DHT)**
5.2 após o primeiro contato, o **cliente entra** no *swarm* e **recebe** uma parte aleatória do torrent, se tornando um *leecher*
5.3 como *leecher*, o **cliente entra** no "jogo de trocas" do *swarm*, onde **só recebe uma parte do torrent se fornecer outra**
5.4 ao **término** do download, o usuário **se torna um seeder**, passando a unicamente fornecer partes do torrent

ABC no BCC: disciplinas relacionadas

- **Desenv. de Algoritmos, Estrutura de Dados, Análise de Algoritmos**: tabela hash (DHT), listas ligadas, árvores e estruturas compostas
- **Prog. para Redes**: protocolos de rede HTTP e UDP, segurança SSL, conexões de rede IPv4 e IPv6, roteamento (NAT PMP), troca de mensagens entre *peers*, descoberta de *peers* locais (Multicast)
- **Intro. à Criptografia**: integridade e privacidade de dados (SHA-1, RC4)
- **Sistemas Operacionais, Prog. Concorrente, Intro. à Comp. Paralela e Distribuída**: *threads*, DHT, leitura e escrita de partes de arquivo, retomada de download (*download resume*)
- **Lab. de Programação 1–2**: organização do código, Automake, Autoconf, testes de código

Referências

[1] Sandvine Inc. **Global Internet Phenomena Report — 1H 2013**, 2013. `http://macaubas.com/wp-content/uploads/2013/05/Sandvine_Global_Internet_Phenomena_Report_1H_2013.pdf`

[2] Ernesto. **BitTorrent Makes Twitter's Server Deployment 75x Faster**, 16 de julho de 2013. *TorrentFreak* `http://torrentfreak.com/bittorrent-makes-twitters-server-deployment-75-faster-100716/`

[3] Ernesto. **Facebook Uses BitTorrent, and They Love It**, 25 de junho de 2013. *TorrentFreak* `http://torrentfreak.com/facebook-uses-bittorrent-and-they-love-it-100625/`

[4] Wikipedia. **Timeline of file sharing** — *Wikipedia, The Free Encyclopedia*, 2013. `http://en.wikipedia.org/wiki/Timeline_of_file_sharing`

[5] Matheus B. Lehmann, Rodrigo B. Mansilha, Marinho P. Barcellos e Flávio Roberto Santos. **"Swarming: como BitTorrent revolucionou a Internet"**. Em *Atualizações em Informática*. PUC-Rio. Vol. 1. Rio de Janeiro, 2011. Cap. 6, pp. 209–258