Anatomia do BitTorrent

a Ciência da Computação por trás do protocolo

Paulo Cheadi Haddad Filho
Orientador: José Coelho de Pina

Trabalho de Formatura Supervisionado

IME-USP

Universidade de São Paulo
São Paulo, 2013

Sumário

Lista de Códigos fonte

Lista de Figuras

Glossário

announce

endereço web (URL) do tracker . 14--17, 19

anycast

método de endereçamento e roteamento de rede onde os datagramas de um único remetente são roteados para um membro de um grupo de receptores potenciais que estão definidos pelo mesmo intervalo no endereço de destino. Geralmente é usado para serviços que demandem alta disponibilidade. 4

Audiogalaxy

Rede P2P de compartilhamento de músicas MP3 criado em 1998. 2, 3

bencode

codificação B, pronunciado *bê encode*; formato de codificação compacta de arquivos torrent para transmissão de metadado . 11

beta tester

usuários de uma versão beta de um software. 6

case insensitive

escrever

. ix

checksum

em português, soma de verificação; bloco de dados de tamanho fixo gerado por algum algoritmo de verificação de integridade usado no certificado de integridade contra problemas durante a transmissão ou de armazenamento (leitura ou escrita). ix

DHT

do inglês *distributed hash table*; tabela de hash distribuída, ou seja, é um serviço de busca similar a uma tabela de hash , mas descentralizada e na forma de sistema distribuído. vii, 5

eDonkey

lançado em 6 de setembro de 2000, o protocolo foi inaugurado juntamente com o software que o utilizava, o eDonkey2000, mas inúmeros softwares cliente para diferentes plataformas surgiram nos dias seguintes ao lançamento. 4, 5

função de hash

é uma função ou algoritmo matemático que mapeia um dado de comprimento variável em outro de comprimento fixo. ix, 13, 15, 18

gateway NAT

escrever

. 16

Gnutella

software de compartilhamento P2P desenvolvido por 3 programadores da empresa Nullsoft, recém adquirida da AOL Inc., lançado em 2000 sob a licença GPL. No dia seguinte ao lançamento, a AOL ordenou indisponibilizar o software, alegando problemas legais e proibindo a continuação do desenvolvimento. Alguns dias depois, o protocolo já tinha sido alvo de engenharia reversa e já havia softwares que o implementavam. 4, 5, 7

HTTP GET

escrever

. 15

ISP

do inglês *Internet Service Provider*; fornecedores de acesso a Internet, que são empresas que vendem serviço e equipamento que permitem o acesso de um computador pessoal à Internet. 2, 19

Kademlia

DHT usado em P2P que especifica a estrutura da rede e a troca de informações através de buscas de nós, guardando as localizações de recursos que estão na rede. 5

leecher

em português, sugador; nome dado ao peer que ainda não terminou um download de um torrent. iv, 8, 10, 17

magnet link

em português, link magnético; padrão aberto, definido por convenção, de esquema de URI utilizado para localizar recursos de rede BitTorrent para download. iii, 12--14

metadado

dados sobre outros dados; informação sobre outra informação. v, viii, 8, 12, 13, 15

MP3

do inglês *MPEG-1/2 Audio Layer 3*; formato patenteado de compressão de dados de áudio digital, que usa um método de compressão de dados com perdas. v, 2, 3

P2P

do inglês *peer-to-peer*; redes de arquitetura descentralizada e distribuída, onde cada nó (peer) fornece e consome recursos. v--vii, 2--4, 6, 8

peer

em português, significa par, colega; nome que se dá a cada nó da rede, ou seja, a um computador conectado. vii--ix, 4, 5, 7--9, 13, 15--17, 19

proxy

escrever

. 16

query string

escrever

. 13

RIAA

do inglês *Recording Industry Association of America*; Associação da Indústria de Gravação da América, organização que representa as gravadoras musicais e distribuidores, e tem sido autora de ações judiciais devido a quebra de direitos autorais causada por compartilhamento indevido de música. 3

scrape

endereço web (URL) do tracker . 17

seeder

em português, semeador; nome dado ao peer que já terminou um download de um torrent e que, por ainda estar conectado à rede, fornece partes a possíveis interessados. iv, 8, 10, 16, 17

string

sequência de caracteres. ix, 11, 13, 15, 16

swarm

em português, enxame; grupo de peer que estão compartilhando dados de um mesmo torrent num determinado momento. 8, 9, 13, 15, 17

swarming

também chamado de transmissão de arquivos por segmentação ou de múltiplas fontes, é a transmissão em paralelo de um arquivo, a partir de um ou vários locais onde estiver disponível, para um único destino. Cabe ao software do destinatário juntar as partes recebidas. 4, 5

tabela de hash

ou *mapa de hash*, é uma estrutura de dados que cria uma lista de correspondência chave-valor, onde os dados são guardados como valores e indexados por seus respectivos *valores hash*. vi, 5

TCP

escrever

. 15

torrent

arquivo de extensão .torrent que contém metadado , como a lista dos nomes dos arquivos

a serem baixados e seus tamanhos, checksum das partes do torrent, além de endereços de um ou mais tracker . v, viii, 8, 9, 11--14, 16, 17

tracker

em português, rastreador; servidor que funciona como um ponto de encontro de peer . v, viii, ix, 6, 8, 13, 15--17, 19

UDP

escrever

. 15

URI

do inglês *Uniform Resource Identifier*; Identificador Uniforme de Recursos, é uma string usada para identificar algum recurso, especificando algum protocolo e um caminho. Por exemplo, o URI `file:///arquivo.txt` indica um arquivo computador local (nome de esquema `file`), enquanto `http://pagina.com` se refere a uma página de Internet (nome de esquema `http`). vii, ix, 13

URL encode

escrever

. 15

URN

do inglês *Uniform Resource Name*; Nome Uniforme de Recursos, é o nome histórico dado a uma URI que usa o esquema "urn:". Sua sintaxe é ‘‘urn:<NID>:<NSS>’’, onde ‘‘urn:’’ é o prefixo case insensitive , ‘‘<NID>’’ é o identificador de espaço de nomes, que determina a interpretação sintática de ‘‘<NSS>’’, que é a string específica do espaço de nomes usado. 13

valor hash

ou *hash*; valores gerados por uma função de hash . 5, 13--15

Lista de tarefas pendentes

Capítulo 1

Introdução

Aqui vou explicar o objetivo do trabalho e o que será mostrado ao longo dele.

Capítulo 2

Napster, Gnutella, eDonkey e BitTorrent

Para entendermos como e por que o BitTorrent se tornou o que é hoje, devemos voltar um pouco no tempo e rever a história que precedeu à sua criação, que é o fim da década dos anos 1990.

2.1 Período pré-torrent

Entre o final dos anos 80 e o início dos 90 [38, 52], a Internet deixou de ser uma rede de computadores usada somente por entidades governamentais, laboratórios de pesquisa e universidades, passando a ter seu acesso comercializado para o público em geral pelos fornecedores de acesso a Internet (ISPs) [43]. Com o advento do formato de áudio MP3 (MP3) [47] no final de 1991 e do seu primeiro reprodutor de áudio MP3 Winamp, o tráfego da Internet aumentou devido ao aumento da troca direta desse tipo de arquivo.

Entre 1998 e 1999, dois sites de compartilhamento gratuito de músicas foram criados: o MP3.com [46], que era um site de divulgação de bandas independentes, e o Audiogalaxy.com [30, 34]. Mais popular que o primeiro, o Audiogalaxy era um site de busca de músicas, sendo que o download e upload eram feitos a partir de um software cliente. A lista de músicas procuradas ia da página para o computador onde usuário tinha instalado o cliente, que então conectava com o do outro usuário, que era indicado pelo servidor. A lista possuía todos os arquivos que um dia passaram pela sua rede. Se algum arquivo fosse requisitado mas o usuário que o possuísse não estivesse conectado, o servidor central do Audiogalaxy fazia a ponte, pegando o arquivo para si e enviando-o para o cliente do requisitante em seu próximo login.

O período dos 3 anos seguintes à criação desses dois sites foi muito produtivo ao mundo das redes peer-to-peer (P2P) de modo geral, onde surgiram alguns protocolos desse paradigma e

inúmeros softwares que os implementavam. Os mais relevantes foram o Napster, o Gnutella, o eDonkey e o BitTorrent.

2.1.1 Napster

Em maio de 1999 surgiu o Napster [48], um programa de compartilhamento de MP3 que inovou por desfigurar o usual modelo cliente-servidor, onde um servidor central localizava os arquivos nos usuários e fazia a conexão entre os usuários, onde ocorria a transferências. O Napster foi contemporâneo do Audiogalaxy e ambos fizeram muito sucesso por cerca de 2 anos, até que começaram as ações judiciais.

Não demorou muito tempo para a indústria da música entrar em ação contra a troca de arquivos protegidos por direitos autorais sem autorização pela Internet. Seu primeiro alvo foi o Napster, em dezembro de 1999, quando a RIAA (do inglês *Recording Industry Association of America*) entrou com processo representando várias gravadoras alegando quebra de direitos autorais [21]. Em abril de 2000, foi a vez da banda Metallica processar, como resposta à sua descoberto que uma música ainda não lançada oficialmente já circulava na rede [17, 20]. Um mês depois, outra ação, agora encabeçada pelo rapper Dr. Dre, que tinha feito pedido formal para a retirada de seu material de circular, também abriu processo [11]. Isso fez com que o Napster tivesse atenção da mídia, ganhando popularidade e atingindo o 20 milhões de usuários em meados de 2000 [25].

Em 2001, esses imbróglios judiciais resultaram numa liminar federal que ordenava a retirada de conteúdo protegido das entidades representadas pela RIAA. O Napster tentou, mas a juíza do caso não ficou satisfeita ordenando então, em julho, o desligamento da rede enquanto não conseguisse controlar o conteúdo que trafegava ali [48]. Em setembro, o Napster fez um acordo, onde pagou 26 milhões de dólares por danos já causados, uso indevido de música e também 10 milhões de dólares pelos danos futuros envolvendo royalties. Para pagar esse valor, o Napster tentou cobrar o serviço de seus usuários, que acabaram migrando de rede P2P, inclusive para o Audiogalaxy. Não conseguindo, em 2002, o Napster decreta falência e é forçado a liquidar seus ativos. De lá para cá, foi negociado algumas vezes e atualmente pertence ao site Rhapsody [28].

O sucesso do Napster, mesmo que por curto período tempo, mostrou o potencial das redes P2P poderiam ter, e com isso novos softwares e protocolos de redes foram sendo lançados, sempre tentando se diferenciar dos outros softwares a fim de não serem novos alvos de ações judiciais. A solução para isso foi tentar descentralizar os mecanismos de indexação e de busca, que foram os pontos fracos do Napster.

2.1.2 Gnutella

O tal sucessor foi o Gnutella, em março de 2000 [40], foi uma resposta de domínio público feita com ‘‘gambiarras’’ para os problemas que o Napster encontrou com relação ás acusações deviolação de direitos autorais. Enquanto o Napster possuía um servidor central como estrutura que, no julgamento, foi usado como prova de que o sistema encorajava a violação de direitos autorais, o Gnutella foi modelado como um sistema P2P puro, onde todos os peers (nós da rede) são completamente iguais, sendo responsáveis pelos seus próprios atos.

O Gnutella disponibiliza arquivos da mesma forma que o Napster [4], mas sem a limitação de ser de formato de música, ou seja, qualquer arquivo pode ser compartilhado. A diferença mais significativa entre os dois protocolos é o algoritmo de busca: a abordagem do Gnutella é baseada numa forma de anycast. Isso envolve duas partes: a primeira é como cada usuário é conectado a outros nós e mantém a lista dessas conexões atualizada. A segunda parte é como ele trata as buscas e trabalha inundando de pedidos para todos os nós que estão a uma certa distância do usuário (nó-cliente). Por exemplo, se a distância limite for de 4, então todos os nós que estiverem a 4 passos a partir do cliente serão verificados, começando a partir dos mais próximos. Eventualmente, algum nó possuirá o arquivo requisitado e responderá, e assim será feita a transferência desse arquivo. Muitos softwares que implementam o protocolo vão além dessa funcionalidade básica de download simples tentando transferir de forma paralela partes diferentes do arquivo desejado de nós diferentes, tentando amenizar eventuais problemas de velocidade de rede.

Experiências sugerem assim que o sistema escala para um tamanho maior, o mecanismo de anycast se torna extremamente caro e em algumas vezes até proibitivo. O problema ocorre nas buscas por arquivos menos populares, onde será necessário um maior número de nós perguntados.

O Gnutella ainda teve uma segunda versão [39], no final de 2002, onde utilizou o mesmo protocolo que o original, porém organizando a rede de peers em *leafs* (folhas, em inglês) e *hubs*. Um *hub* poderia ter centenas de conexões de folhas outras 7, em média, a outros *hubs*, enquanto uma folha se conectaria apenas a 2 *hubs*. Essa nova topologia, somada com uma nova tabela de índice de arquivos das folhas mantida pelos *hubs* onde estavam conectados, melhorou o desempenho das buscas, que era ruim na versão antiga.

2.1.3 eDonkey

O protocolo eDonkey inovou em muitos aspectos de seus precursores, tendo papel fundamental na história das redes P2P e sua consolidação como ferramenta de compartilhamento especializado em arquivos grandes.

O eDonkey implementou o primeiro método de download por swarming (enxame de peers,

que é chamado o método onde peers fazem downloads de diferentes partes de um arquivo e de peers diferentes, utilizando de forma efetiva a largura de banda de rede para todos os peers ao invés de ficar limitado somente à banda de um único peer.

Outra melhoria foi a busca: no seu lançamento, os servidores eram separados entre si, porém nas versões seguintes permitiu que eles formassem uma rede de buscas. Isso permitiu que os servidores repassassem buscas de seus clientes conectados localmente a outros servidores, facilitando a localização de peers conectados em qualquer servidor da rede de buscas, aumentando a capacidade de download do enxame.

Uma terceira diferença com o Napster foi o uso de valores hash de arquivos nos resultados de busca ao invés dos simples nomes dos arquivos. As buscas geradas pelos usuários eram baseadas em palavras-chave e comparadas com a lista de nomes de arquivos armazenada no servidor, mas o servidor retornava uma lista de pares de nomes de arquivos com seus repectivos valores valor hash. Enfim, quando o usuário selecionasse o arquivo desejado, o cliente iniciaria o download do arquivo usando o seu valor valor hash. Desse modo, um arquivo poderia ter muitos nomes entre os diferentes peer e servidores, mas seria considerado idêntico para download se possuísse o mesmo valor hash.

A arquitetura da rede em dois níveis usando cliente e servidor alcançou um meio termo entre as redes centralizadas, como o Napster, e as descentralizadas, como o Gnutella, já que o servidor central no primeiro era um alvo estável para ações legais, enquanto o segundo rapidamente mostrou-se inviável devido ao tráfego massivo de buscas entre peers.

Por fim, a inovação mais importante foi o uso de tabela de hash distribuídas (tabela de hash distribuídass), em específico o Kademlia, como algoritmo de indexação e busca nos servidores centrais dos arquivos através da rede eDonkey. Além de ser uma das causas da melhora no desempenho nas buscas, tabela de hash distribuídas possuem ainda outras características como tolerância a falhas e escalabilidade. O Kademlia ainda oferece outras vantagens como armazenamento de dados eficiente; anonimato; segurança de rede, conteúdo e usuário; e autenticação.

2.2 Nascimento do BitTorrent

Em meados dos anos 90, Bram Cohen era um programador que tinha largado a faculdade no segundo ano do curso de Ciência da Computação da Universidade de Buffalo, Nova Iorque, para trabalhar em empresas *pontocom*. A última delas foi a MojoNation, uma empresa que desenvolvia um software de distribuição de arquivos criptografados por swarming, que ele já tinha percebido ser uma vantagem com relação ao Kazaa, que fazia transferências de uma única origem.

Em abril de 2001, Bram saiu da MojoNation e começou a modelar o protocolo BitTorrent, lançando a primeira implementação em Python em julho de 2001. Em fevereiro de 2002, ele apresentou o seu trabalho na CodeCon [8] e na mesma época começou a testá-lo, usando uma coleção de material pornográfico para atrair beta testers [29]. O software começou a ser usado imediatamente pelos usuários.

Nesse meio tempo, Bram ainda passou pela Valve [36], empresa de desenvolvimento de jogos, para trabalhar no sistema de distribuição online do jogo Half Life 2. Em 2004, saiu da Valve e voltou o foco ao Torrent. Em setembro, fundou a BitTorrent Inc. com seu irmão Ross Cohen e o parceiro de negócios Ashwin Navin, sendo então responsável pelo desenvolvimento do protocolo. Nesse mesmo ano, o BitTorrent passou a ser bastante utilizado, quando surgiram os primeiros programas de televisão e filmes compartilhados pelo protocolo.

Já em maio de 2005, Cohen lança uma nova versão do BitTorrent que não precisa de rastreadores (*trackers*), juntamente com um site de buscas de conteúdo na Internet. Em setembro, a empresa recebe do investidor David Chao $8.5 milhões de dólares. No final desse ano, a BitTorrent Inc. e a MCAA [45] fizeram um acordo que a empresa de ajudar os usuário a encontrar conteúdos não autorizado dos representados da associação, que não ajudou no combate a pirataria pois já havia outros sites de busca nessa época, como o TorrentSpy, Mininova, NewNova e o The Pirate Bay.

2.3 Mundo pós-torrent

Desde o fechamento do site de buscas, a BitTorrent Inc. tem desenvolvido outros softwares baseados na tecnologia P2P [5], como transmissão de vídeos ao vivo (BitTorrent Live), sincronização de arquivos entre computadores ligados à Internet (BitTorrent Sync), publicação e distribuição de conteúdo de artistas a seus fãs (BitTorrent Bundles), entre outros serviços comerciais.

Como protocolo, o BitTorrent criou um novo paradigma de transmissão de informação pela Internet, sendo utilizado de inúmeras formas e motivos.

- alguns softwares de podcasting, como o Miro [23], passaram a usar o protocolo como forma de lidar com a grande quantidade de downloads de programas online;
- o site da gravadora DGM Live fornece o conteúdo via torrent após a venda [10];
- VODO [31] é um site de divulgação e distribuição de filmes sob a licença Creative Commons e que faz a publicação em outros sites de buscas de torrent;
- canais como a americana CBC [7] e a holandesa VPRO [9] já disponibilizaram programas de sua programação para download. A norueguesa NRK o faz para conteúdos em HD [26] e, apesar de algumas restrições de direitos, tem aumentado a oferta;

- o serviço da Amazon Web Services de armazenamento de conteúdo via web service, a Amazon S3, permite o uso de torrent para a transmissão de arquivos [1];
- as empresas de desenvolvimento de jogos CCP Games, produtora do jogo Eve Online, usa o protocolo para distribuir o instalador de seu jogo [2]. A Blizzard, dona de Diablo III, StarCraft II e World of Warcraft, distribui esses jogos e ainda suas eventuais atualizações [6];
- o governo britânico distribuiu os detalhes de seus gastos [15], enquanto a Universidade do Estado da Flórida usa para transmitir grandes conjuntos de dados científicos para seus pesquisadores [18];
- Facebook [12] e Twitter [14] usam para transmitir o código atualizado de seus sites para seus servidores de aplicação de forma eficiente [13].

Em 2013, o BitTorrent é um dos maiores geradores de tráfego de rede do mundo, ao lado do NetFlix, Youtube, Facebook e acessos HTTP [19], e segue em uma tendência de aumento.

2.3.1 Questões legais

Desde que surgiu, o BitTorrent, bem como os outros protocolos, chamou atenção dos defensores de direitos autorais e acabou sendo alvo de medidas judiciais. Porém, assim como o Gnutella e ao contrário do Napster, por possuir uma estrutura descentralizada e não armazenar dados sob direitos, dificulta o trabalho de se identificar usuários desses dados.

Ainda assim, não existe um consenso sobre os efeitos financeiros do compartilhamento de arquivos protegidos por direitos autorais, onde o principal argumento utilizado pelos reclamantes é que estes têm fortes prejuízos e, por isso, entram com ações indenizatórias de grandes valores. Existem alguns estudos que tentam medir esse prejuízo; um dos mais recentes [27] mostrou que não existem evidências de diminuição das receitas das empresas cujo conteúdo é pirateado e que o combate contra os usuários infringentes não tem o impacto esperado, que é de reduzir o compartilhamento.

2.3.2 Estudos acadêmicos

tá beeem resumido; desenvolve mais??

Academicamente, o protocolo é bastante estudado desde o seu surgimento, onde são focados efeitos do algoritmo original e alguns estudos de ajuste fino de seu funcionamento. Os pontos principais são o algoritmo do jogo da troca de pedaços, estudos das topologias das redes formadas pelos peers e melhorias de eficiência com alterações nessas topologias.

Capítulo 3

Anatomia do BitTorrent

O BitTorrent é uma rede P2P onde cada um de seus usuários assume o papel híbrido de servidor, que fornece os arquivos, e de cliente, que adquire os arquivos. Cada computador é chamado de peer.

Cada transferência por BitTorrent está associada a um arquivo de metadados chamado *torrent*. Esse arquivo contém informações sobre os arquivos que formam o pacote de dados daquele conjunto de dados e também um ou mais endereços de trackers, que mantém listas atualizadas dos peers que estão compartilhando os dados, atualizado em períodos de tempo curtos (usualmente 30 minutos).

Enquanto um peer estiver fazendo download de um torrent é chamado de leecher, pois ainda estará consumindo dados de outros peers; quando o download acabar, passará a ser um seeder, que somente envia dados para outros peers.

Os arquivos torrent ficam disponíveis em vários sites de índice (às vezes chamados de comunidades), como o ThePirateBay, o Kickass ou Torrentz, muitas vezes em mais de um deles ao mesmo tempo. Apesar de todo conteúdo compartilhado possuir um arquivo torrent, não necessariamente um arquivo torrent está sendo compartilhado, podendo inclusive estar extinto.

Peers que participam do compartilhamento de um arquivo torrent específico fazem parte do enxame (*swarm*), onde os dados contidos no pacote desse arquivo são compartilhados com outros por partes.

Essas partes variam de acordo com cada arquivo torrent. O tamanho total do conjunto de arquivos contidos nesse arquivo é dividido em partes de tamanho fixo (geralmente 256kB) e transmitido de forma independente das outras, seguindo a ordem estabelecida pelo algoritmo de troca de partes (explicado na seção 3.3), ordem essa que varia de acordo com o estado atual do swarm para o arquivo torrent dado.

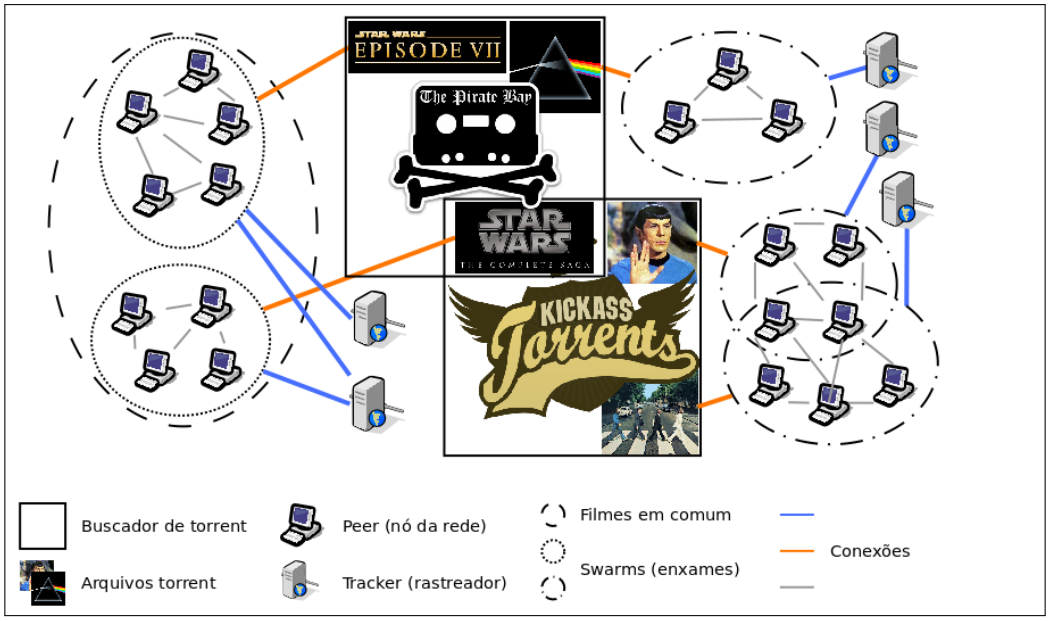

Figura 3.1: *amostra de uma rede de conexões BitTorrent*

Todos esses agentes possuem relações de múltiplas entre si. Por exemplo, um mesmo arquivo torrent pode estar indexado por vários sites indexadores. Como veremos nos capítulos seguintes, eles contém uma informação que os identificam unicamente entre si, mantendo a consistência de dados através desses vários sites de busca. Outra observação a ser feita é que um peer pode estar baixando mais de um torrent simultaneamente, ou seja, participando de dois swarms ao mesmo tempo. Por fim, em alguns casos um arquivo torrent possui grande quantidade de peers, havendo necessidade de se dividir o swarm em algumas partes para fins de escalabilidade da rede formada.

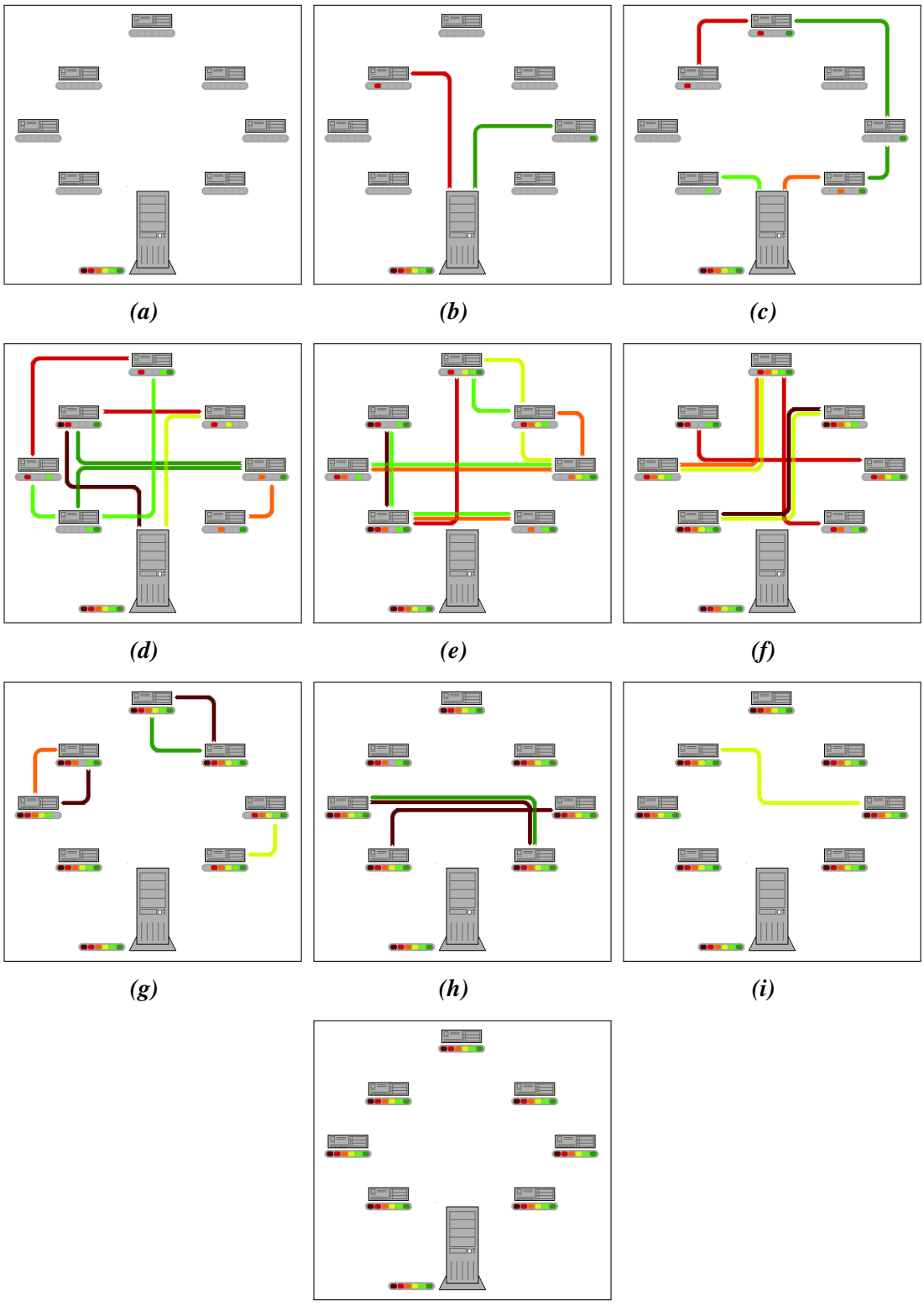

Figura 3.2: *simulação de uma transferência torrent: o seeder, na parte inferior das figuras, possui todas as 5 partes de um arquivo, que os outros computadores, os leechers, baixam de forma independente e paralela. Fonte: [53]*

Arquivo .torrent

Ao se adicionar um torrent em um programa cliente, ocorrem muitas transmissões de dados antes do download de fato. Para demonstrar isso, usaremos um arquivo torrent do filme ‘‘A Noite dos Mortos Vivos’’ de 1960 [24], que é de domínio público e livre de direitos autorais.

Se abrirmos esse arquivo, veremos uma grande sequência de caracteres (*string*), caracteres diferentes e incomuns, formando um conteúdo ilegível (na seção binária) e sob uma forma compacta, mostrado abaixo.

```
1  d8:announce36:http://bt1.archive.org:6969/announce13:announce-listll36:http://bt1.
2  archive.org:6969/announceel36:http://bt2.archive.org:6969/announceee7:comment13:crea
3  tiondatei1343715473e4:infod5:filesld5:crc328:030208fe6:lengthi4127671704e3:md532:627
4  f5a428f9e454ccfcb29d31b87169a5:mtime10:10794024804:pathl29:night_of_the_living_dead.
5  mpege4:sha140:5e44bb1b3f700240249a5287c64dc02dc56d034bee4:name24:night_of_the_living
6  _dead12:piecelengthi4194304e6:pieces23720:<binary>
7
8  (...)
9
10 e6:locale2:en5:title24:night_of_the_living_dead8:url-listl28:http://archive.org/
11 download/39:http://ia600301.us.archive.org/22/items39:http://ia700301.us.archive.
12 org/22/itemsee
```

Código-fonte 3.1: *trecho do conteúdo do arquivo .torrent do filme ‘‘A Noite dos Mortos Vivos’’, de 1960 [24], com a parte binária truncada*

Esse conteúdo está organizado usando o formato codificação B (bencode), que é um formato de codificação compacta de arquivos especial para arquivos torrent e não é legível ao ser humano. Com alguma formatação, podemos enxergar os componentes separadamente, como mostra o código 3.2.

Esse conteúdo tem significado, sendo utilizado da seguinte forma [54]:

- **strings** são prefixos de números na base 10 que representam comprimentos, seguidos por um caractere `:` e então o conteúdo. Por exemplo, na linha 2, `8:announce` corresponde à string `"announce"`.
- **números** são representados por um `i`, seguidos do valor na base 10 (sem qualquer limite, mas sem zeros precedentes como em `0003`, e podendo ser negativo), e terminados por um `e`. Por exemplo, na linha 11, `i1343715473e` corresponde ao número `1343715473`.
- **listas** são formadas por `l`, seguidos por seus elementos (também no formato bencode), e então terminados por `e`. Por exemplo, `l3:foo3:bare` corresponde a `["foo", "bar"]`. No código acima, é presente entre as linhas 43 e 47.
- **dicionários** são definidos por `d`, seguidos de uma lista alternada de chaves e seus valores correspondentes, terminando com `e`, onde as chaves devem estar ordenadas usando-se comparação binária ao invés da usual alfanumérica. Por exemplo, a string

`d3:foo3:bar6:foobar6:bazbare` corresponde ao dicionário puro `{"foo": "bar", "foobar": "bazbar"}`, e a estrutura mais complexa dada por `d3:fool6:foobar3:bazee` equivale a `{"foo": ["foobar", "baz"]}`.

```
d
    8:announce
    36:http://bt1.archive.org:6969/announce
    13:announce-list
    l
        l36:http://bt1.archive.org:6969/announcee
        l36:http://bt2.archive.org:6969/announcee
    e
    7:comment
    13:creation date
    i1343715473e
    4:info
    d
        5:files
        l
            d
                5:crc32
                8:030208fe
                6:length
                i4127671704e
                3:md5
                32:627f5a428f9e454ccfcb29d31b87169a
                5:mtime
                10:1079402480
                4:path
                l29:night_of_the_living_dead.mpege
                4:sha1
                40:5e44bb1b3f700240249a5287c64dc02dc56d034b
            e
        e
        4:name
        24:night_of_the_living_dead
        12:piece length
        i4194304e
        6:pieces
        23720:<binary>
    e
    6:locale
    2:en
    5:title
    24:night_of_the_living_dead
    8:url-list
    l
        28:http://archive.org/download/
        39:http://ia600301.us.archive.org/22/items
        39:http://ia700301.us.archive.org/22/items
    e
e
```

Código-fonte 3.2: *trechos formatados de forma legível do conteúdo do arquivo .torrent do filme ‘‘A Noite dos Mortos Vivos’’, de 1960 [24], com a parte binária truncada*

Magnet Link

Além do arquivo torrent, existe uma outra forma de se transmitir os metadados necessários para se iniciar a transmissão, que é utilizando magnet link.

Magnet links, ao contrário dos arquivos torrent, não estão gravados em algum dispositivo de armazenamento. Basicamente, é um esquema de identificador uniforme de recursos (URI) usado exclusivamente para o protocolo.

No caso citado, o site de origem do torrent que estamos usando não fornece um magnet link oficialmente. Porém, o Transmission consegue construir uma URI a partir do arquivo original, para fins de compartilhamento direto. O resultado, após decodificar o endereço para um formato legível (retirando a codificação de caracteres especiais [49]) foi o seguinte:

```
magnet:?xt=urn:btih:72d7a3179da3de7a76b98f3782c31843e3f818ee
&dn=night_of_the_living_dead
&tr=http://bt1.archive.org:6969/announce&tr=http://bt2.archive.org:6969/announce
&ws=http://archive.org/download/
&ws=http://ia600301.us.archive.org/22/items/&ws=http://ia700301.us.archive.org/22/items/
```

Código-fonte 3.3: *magnet link do arquivo .torrent do filme ‘‘A Noite dos Mortos Vivos’’ , de 1960 [24], com parâmetros divididos entre linhas para melhor visualização*

Esse endereço é composto por vários pares de nomes de parâmetros (sem qualquer ordem específica) e seus respectivos valores, formando uma *query string*. Podemos dividir esse endereço em partes, cada uma tendo o seu significado:

- **xt**: parâmetro para *exact topic*, ou tópico exato, contém a informação mais importante do magnet link: o identificador único de torrent. Serve para encontrar e verificar os arquivos especificados. No caso, `urn:btih:<hash>` corresponde ao nome uniforme de recursos (URN) `btih` (*BitTorrent Info Hash*), que é a string valor hash resultado da função de hash SHA-1, convertida para hexadecimal
- **dn**: parâmetro que contém o *display name*, ou nome de visualização, contém um nome que é mostrado para o usuário, por conveniência
- **tr**: o *address tracker*, ou endereço do tracker, onde o programa cliente vai procurar as informações de peers
- **ws**: endereço do arquivo para *webseed*, ou fornecimento web, que é o endereço do arquivo em um servidor HTTP ou FTP, sendo utilizado como alternativa a um swarm problemático [35]

3.1 Busca por informações

Quando adicionamos um torrent ao Transmission, o programa salva as informações em disco durante todo o período que estiverem sendo gerenciados por ele. Caso tenha sido por meio de um arquivo de metadados, uma cópia deste é salva em uma pasta pré-definida para seu controle interno; caso seja por magnet link, um novo arquivo é criado contendo as informações

obtidas por ele, por questões de praticidade, para que não necessite fazer essa aquisição dos dados novamente ao ser aberto ou quando alguma transferência for pausada e depois continuada.

Após esse processo de arquivamento, o programa processa essas informações salvas para deixá-las carregadas em memória a fim de obter o valor hash que identifica o torrent.

./libtransmission/metainfo.c:367

```
static const char*
tr_metainfoParseImpl(const tr_session * session, tr_info * inf,
bool * hasInfoDict, int * infoDictLength, const tr_variant * meta) {
  // variáveis temporárias
  int64_t i; size_t len; const char * str; const uint8_t * raw;
  tr_variant * infoDict = NULL;

  // flags
  bool b, isMagnet = false;

  /* info_hash: urlencoded 20-byte SHA1 hash of the value of the info key from the
   * Metainfo file. (...) */
  b = tr_variantDictFindDict(meta, TR_KEY_info, &infoDict);
  if (hasInfoDict != NULL) *hasInfoDict = b;
```

Se aquele arquivo para controle interno tiver sido criado por conta de um magnet links, não possuirá a chave `info` em seu conteúdo, mas deverá conter as chaves `urn:btih:<hash>` e `info_hash`.

./libtransmission/metainfo.c:367

```
  if (!b) { // Não possui a chave "info" no dicionário.
    // Será que não é um magnet link?
    if (tr_variantDictFindDict(meta, TR_KEY_magnet_info, &d)) {
        isMagnet = true;

        if (!tr_variantDictFindRaw(d, TR_KEY_info_hash, &raw, &len) ||
            len != SHA_DIGEST_LENGTH) // Se tiver a chave "info_hash" válida, a usa.
          return "info_hash";

        memcpy(inf->hash, raw, len);
        tr_sha1_to_hex(inf->hashString, inf->hash);
        (...)
    }
    else return "info"; // Não é magnet link e não possui a chave "info".
  }
```

Já se o arquivo para controle interno tiver sido gerado como cópia do arquivo torrent, possuirá o mesmo dicionário original, que contém a chave `info_hash`, que é utilizada para calcular o valor hash do arquivo.

./libtransmission/metainfo.c:367

```
  else {
    int len;
    char * bstr = tr_variantToStr(infoDict, TR_VARIANT_FMT_BENC, &len);
    tr_sha1(inf->hash, bstr, len, NULL);          // Calcula o hash SHA-1 do .torrent...
    tr_sha1_to_hex(inf->hashString, inf->hash);   // ...e converte de base2 para base16
    (...)
  }
```

Outras informações podem ser recuperadas dependendo do formato de origem do torrent, como a privacidade, lista de arquivos e seus respectivos tamanhos, valor hash de cada parte, entre outras. Por fim, termina coletando os endereços de announce.

./libtransmission/metainfo.c:367

```
  if (!tr_variantDictFindInt(infoDict, TR_KEY_private, &i)) // privacidade do torrent
    if (!tr_variantDictFindInt(meta, TR_KEY_private, &i)) i = 0;
  inf->isPrivate = i != 0;

  if (!isMagnet) { // quantidade de partes
    if (!tr_variantDictFindInt(infoDict, TR_KEY_piece_length, &i) || (i < 1))
      return "piece length";
    inf->pieceSize = i;
  }

  if (!isMagnet) { // hashes das partes
    if (!tr_variantDictFindRaw(infoDict, TR_KEY_pieces, &raw, &len) ||
        len % SHA_DIGEST_LENGTH) return "pieces";

    inf->pieceCount = len / SHA_DIGEST_LENGTH;
    inf->pieces = tr_new0(tr_piece, inf->pieceCount);
    for (i = 0; i < inf->pieceCount; i++)
      memcpy(inf->pieces[i].hash, &raw[i * SHA_DIGEST_LENGTH], SHA_DIGEST_LENGTH);
  }

  if (!isMagnet) { // lista de arquivos
    if ((str = parseFiles(inf, tr_variantDictFind(infoDict, TR_KEY_files),
                               tr_variantDictFind(infoDict, TR_KEY_length))))
      return str;

    if (!inf->fileCount || !inf->totalSize ||
        (uint64_t) inf->pieceCount != (inf->totalSize+inf->pieceSize-1)/inf->pieceSize)
      return "files";
  }

  if ((str = getannounce(inf, meta))) return str; // announce(s)
  (...)
  return NULL;
}
```

Announce

Para cada swarm gerenciado, o tracker possui uma lista dos peers que participam dele, que é enviada ao peer que a requer por meio de uma requisição *HTTP GET*. Quando essa requisição é recebida pelo tracker, este inclui ou atualiza um registro para o peer solicitante e devolve uma lista de 50 peers aleatórios, de forma uniforme, que fazem parte do swarm. Não havendo essa quantidade total, a lista toda é enviada ao requisitante; caso contrário, a aleatoriedade proporciona uma diversidade de listas enviadas, proporcionando robustez ao sistema [55].

Esse contato entre um peer e um tracker é chamado de announce, que pode ser feito usando-se tanto com o protocolo TCP bem como UDP, e é onde peers podem passam várias informações

- **info_hash**: valor hash de 20 bytes resultante da função de hash SHA-1 com *URL encode* do valor da chave `info` do arquivo de metadado
- **peer_id**: string com URL encode de 20 bytes usado como identificador único do programa cliente, gerado no ínício da sua execução. Para isso, provavelmente deve incorporar informações do computador, a fim de se gerar um valor único para o computador

- **uploaded**: a quantidade total de dados, em bytes, enviados desde quando o cliente enviou o primeiro aviso ao tracker
- **downloaded**: a quantidade total de dados, em bytes, recebidos desde quando o cliente enviou o primeiro aviso ao tracker
- **left**: a quantidade total de dados, em bytes, que faltam para o requisitante terminar o download do torrent e passe a ser um seeder
- **compact** (opcional): se o valor passado for 1, significa que o requisitante aceita respostas compactas. A lista de peers enviada é substituída por uma string de peers, cada um com 6 bytes, onde os 4 bytes iniciais são o host e os 2 bytes finais são a porta de transmissão. Por exemplo, o endereço IP 10.10.10.5:80 seria transmitido como `0A 0A 0A 05 00 80`. Deve-se observar que alguns trackers suportam somente conexões deste tipo para otimização da utilização da banda de rede e, pra isso, ou recusam requisições sem `compact=1` ou, caso não as recusem, enviam respostas compactas a menos que a requisição possua `compact=0`
- **no_peer_id** (opcional): sinaliza que o tracker por omitir o campo `peer id} no dicionário de \gls` É ignorado caso o modo compacto esteja habilitado
- **event** (opcional): pode possuir os valores `started` (iniciado), `completed` (terminado), `stopped` (parado), ou vazio para não especificar.
  - *started* : a primeira requisição para o tracker deve enviar este valor
  - *stopped* : avisa que o programa cliente está fechando
  - *completed* : quando o download que estava ocorrendo termina, ou seja, não é enviado quando o programa cliente inicia com o torrent em 100%
- **port** (opcional): o número da porta de conexão que o programa cliente está escutando por transmissões de dados. Em geral, portas reservadas para BitTorrent estão entre a 6881 e a 6889. Se esse for o caso, pode ser omitido
- **ip** (opcional): o endereço IP verdadeiro do requisitante no formato legível do IPv4 (4 conjuntos de número de 0 a 255 separados por `.`) ou do IPv6 (8 conjuntos de números hexadecimais de 4 dígitos separados por `:`). Não é sempre necessário, pois o endereço pode ser conhecido através da requisição. Assim, é usado quando o programa cliente está se comunicando com o tracker através de um *proxy* ou quando ambos cliente e tracker estão no mesmo lado local de um *gateway NAT*, pois nesse caso o endereço IP não é roteável
- **numwant** (opcional): quantidade de peers que o requisitante gostaria de receber do tracker. É permitido valor zero. Se omitido, assume valor padrão de 50
- **key** (opcional): mecanismo de idetificação adicional para o programa cliente provar sua identidade caso tenha ocorrido mudança no seu endereço IP
- **trackerid** (opcional): se a resposta de um announce anterior continha o endereço IP de um tracker, deve ser enviado neste campo

Essa comunicação ocorre nas seguintes situações:

- no primeiro contato do peer, para que ele tenha acesso a um swarm
- a cada período de tempo, estipulado pelo tracker, para que o peer continue mostrando que ainda está conectado, além de pode receber endereços de peers novos
- quando a quantidade de peers conhecidos que estão ativos é menor do 5
- quando terminar o download, notificando que passou a ser um seeder
- quando sai do swarm, seja por desconexão ou por encerramento do programa cliente

Além do announce, outra forma de troca de informação entre peers e trackers é pelo scrape. Geralmente usado pelos programas cliente para decidir quando realizar um announce, informa o número de peers, leechers e seeders de uma lista de um ou mais torrents. É dessa forma que os sites de indexação sabem dessas informações e as apresentam nas páginas.

explicar como o Transmission usa o torrent

3.2 Fontes de arquivos

Mostrarei o processamento dos dados adquiridos na seção anterior e como ele organiza a lista das fontes de arquivos usando a tabela hash DHT Kademlia.

3.3 Jogo da troca de arquivos

Explicarei o algoritmo tit-for-tat padrão do protocolo BitTorrent, que vem da Teoria dos Jogos, e como o Transmission o implementa.

Capítulo 4

Conceitos de Computação no BitTorrent

Fazer pequena introdução aqui.

Aqui mostrarei detalhes técnicos sobre as partes coadjuvantes do BitTorrent e do Transmission.

4.1 Estruturas de dados, listas ligadas e árvores

Aqui vou falar de tipos de estruturas de dados utilizadas no programa, como *structs* e a sua utilização na implementação de listas ligadas e árvores e em como estes são usados na implementação de filas.

4.2 Funções de hash

Aqui vou explicar como funciona o algoritmo da função de hash SHA-1 e mostrar como e para que é usado na identificação de torrents e na verificação de integridade de partes.

4.3 Criptografia

Aqui vou explicar como este algoritmo de chave simétrica funciona e como é utilizado pelo Transmission para criptografar pacotes de dados.

4.4 Bitfields

Apesar de ser um simples array de bits usado no gerenciamento de partes que o programa já baixou ou não, foi percebido que o seu uso de forma não-convencional, chamado de *lazy bitfield*, pode ajudar a evitar o controle de banda (chamado de modelagem de tráfego, ou *traffic shaping* em inglês), feito por ISPs.

4.5 Protocolos de redes

Aqui vou explicar o que são os protocolos de rede TCP e UDP, apontar suas diferenças e mostrar os motivos pelos quais o UDP é preferido ao TCP no uso de endereços de announce de trackers.

4.6 Multicast

Apesar de não ser utilizado pelo protocolo BitTorrent, o multicast, que é uma forma de entregar dados a um grupo de computadores simultaneamente numa só transmissão, é usado pelo Transmission para tentar descobrir peers que estão na mesma rede local, otimizando as conexões.

4.7 Roteamento de pacotes

Em alguns roteadores que são ponte entre a Internet e a rede que ele gerencia, existe uma função de se configurar portas de comunicação de rede automaticamente usando-se o *Network Address Translation Port Mapping Protocol*. Assim, não é necessário realizar uma configuração específica somente pare esse fim.

4.8 Retomada de downloads

No Transmission e em algums outros softwares que realizam downloads, existe a função de se pausar a transferência do arquivo para que seja retomado em outro momento. Nesta seção, mostrarei qual a idéiapor trás desse mecanismo e a forma como foi implementado no Transmission.

4.9 Conexão com a Internet

Aqui mostrarei a parte técnica da programação em linguagem C para utilização de transmissão de dados por rede.

4.10 IPv6

O IPv6 é a versão mais recente do protocolo de Internet (IP), que foi criado para substituir o IPv4, que atualmente é mais o usado porém sofre de exaustão de endereços. Nesta seção, falarei sobre o novo protocolo e quais as implicações na programação de softwares com comunicação de redes.

4.11 Threads

Dentro de um contexto de programas que utilizam a Internet e funcionalidades que chegam próximas ao tempo real, processamentos pesados devem ser tratados com cautela a fim de se manter a instantaneidade do processo. Aqui explicarei como o Transmission usa o conceito de *threads* para paralelizar esses processamentos e, com isso, conseguir utilizar as informações de rápida mudança antes que seja necessário obtê-las novamente.

4.12 Engenharia de Software

O Transmission é um programa extenso e complexo, desenvolvido por vários programadores que estão espalhados pelo globo. Nesta seção, abordarei alguns pontos utilizados pelos desenvolvedores na manutenção do código aberto de qualidade e em funcionamento.

Capítulo 5

Comentários Finais

Capítulo 6

Bibliografia

[1] *Amazon Simple Storage Service (Amazon S3) --- Amazon S3 Functionality*. [Online; acessado em 7-outubro-2013]. URL: http://aws.amazon.com/s3/#functionality.

[2] CCP Aporia. *All quiet on the EVE Launcher front?* [Online; acessado em 5-outubro-2013]. 11 de mar. de 2013. URL: http://community.eveonline.com/news/dev-blogs/74573.

[3] Kennon Ballou. *R.I.P. Audiogalaxy*. [Online; acessado em 30-setembro-2013]. 21 de jun. de 2002. URL: http://www.kuro5hin.org/story/2002/6/21/171321/675.

[4] Kenneth P. Birman. *Reliable Distributed Systems: Technologies, Web Services, and Applications*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005, pp. 532–534. ISBN: 0387215093.

[5] *BitTorrent*. URL: http://www.bittorrent.com/.

[6] *Blizzard Downloader --- Wowpedia*. [Online; acessado em 5-outubro-2013]. 2013. URL: http://wowpedia.org/index.php?title=Blizzard_Downloader&oldid=3198520.

[7] *CBC to BitTorrent Canada's Next Great Prime Minister*. URL: http://archive.is/VYmFD.

[8] *Codecon 2002 --- Schedule*. 2002. URL: http://web.archive.org/web/20021012072819/http://codecon.org/2002/program.html#bittorrent.

[9] M. Denters. *Download California Dreaming*. [Online; acessado em 7-outubro-2013]. 8 de nov. de 2010. URL: http://tegenlicht.vpro.nl/nieuws/2010/november/creative-commons.html.

[10] *DGM Live --- FAQ*. [Online; acessado em 7-outubro-2013]. URL: http://www.dgmlive.com/help.htm#whatisbittorrent.

[11] *Dr. Dre Raps Napster*. [Online; acessado em 30-setembro-2013]. 18 de abr. de 2000. URL: http://www.wired.com/techbiz/media/news/2000/04/35749.

[12] Ernesto. *BitTorrent And MPAA join forces*. 23 de nov. de 2005. URL: http://torrentfreak.com/BitTorrent-and-mpaa-join-forces/.

[13] Ernesto. *BitTorrent Makes Twitter's Server Deployment 75x Faster*. 16 de jul. de 2013. URL: http://torrentfreak.com/bittorrent-makes-twitters-server-deployment-75-faster-100716/.

[14] Ernesto. *Twitter Uses BitTorrent For Server Deployment*. 10 de fev. de 2013. URL: http://torrentfreak.com/twitter-uses-bittorrent-for-server-deployment-100210/.

[15] Ernesto. *UK Government Uses BitTorrent to Share Public Spending Data*. 4 de jun. de 2010. URL: http://torrentfreak.com/uk-government-uses-bittorrent-to-share-public-spending-data-100604/.

[16] Elle Cayabyab Gitlin. *BitTorrent gets US$8.75 million in VC money*. 29 de set. de 2005. URL: http://arstechnica.com/uncategorized/2005/09/5363-2/.

[17] Daniela Hernandez. *April 13, 2000: Seek and Destroy – Metallica Sues Napster*. [Online; acessado em 30-setembro-2013]. 13 de abr. de 2012. URL: http://www.wired.com/thisdayintech/2012/04/april-13-2000-seek-and-destroy-metallica-sues-napster/.

[18] *HPC Data Repository*. [Online; acessado em 7-outubro-2013]. URL: http://www.hpc.fsu.edu/index.php?option=com_wrapper&view=wrapper&Itemid=80.

[19] Sandvine Inc. *Global Internet Phenomena Report --- 1H 2013*. Online; retirado de http://macaubas.com/wp-content/uploads/2013/05/Sandvine_Global_Internet_Phenomena_Report_1H_2013.pdf. 2013. URL: https://www.sandvine.com/downloads/general/global-internet-phenomena/2013/sandvine-global-internet-phenomena-report-1h-2013.pdf.

[20] Christopher Jones. *Metallica Rips Napster*. [Online; acessado em 30-setembro-2013]. 13 de abr. de 2000. URL: http://www.wired.com/politics/law/news/2000/04/35670.

[21] David Kravets. *Dec. 7, 1999: RIAA Sues Napster*. [Online; acessado em 30-setembro-2013]. 7 de dez. de 2009. URL: http://www.wired.com/thisdayintech/2009/12/1207riaa-sues-napster/.

[22] M. Lehmann et al. "Swarming: como BitTorrent revolucionou a Internet". Em: *Atualizações em Informática*. Ed. por PUC-Rio. Vol. 1. [Online; accessed 23-outubro-2013]. Rio de Janeiro, 2011. Cap. 6, pp. 209–258. URL: http://www.lbd.dcc.ufmg.br/colecoes/jai/2012/006.pdf.

[23] *Miro*. [Online; acessado em 7-outubro-2013]. URL: http://getmiro.com.

[24] *Moving Image Archive - Night of the Living Dead (1968)*. [Online; acessado em 16-outubro-2013]. URL: http://archive.org/details/night_of_the_living_dead.

[25] *Napster: 20 million users*. [Online; acessado em 30-setembro-2013]. 19 de jul. de 2000. URL: http://cnnfn.cnn.com/2000/07/19/technology/napster/index.htm.

[26] *NRKbeta*. [Online; acessado em 7-outubro-2013]. URL: http://nrkbeta.no/bittorrent/.

[27] *Press Release: Global Napster usage plummets, but new file-sharing alternatives gaining ground, reports Jupiter Media Metrix*. [Online; acessado em 8-outubro-2013]. Out. de 2013. URL: http://www.lse.ac.uk/media@lse/documents/MPP/LSE-MPP-Policy-Brief-9-Copyright-and-Creation.pdf.

[28] *Rhapsody.com*. [Online; acessado em 30-setembro-2013]. URL: http://www.rhapsody.com.

[29] Clive Thompson. *The BitTorrent Effect*. Jan. de 2005. URL: http://www.wired.com/wired/archive/13.01/bittorrent.html.

[30] "Um pouco de história: redes P2P de compartilhamento de arquivos". Em: *Revista PnP* 10 (out. de 2008). [Online; retirado de http://www.thecnica.com/artigos/PnP_10_02.pdf], p. 12. URL: http://www.revistapnp.com.br/pnp_10.php.

[31] *VODO --- About*. [Online; acessado em 7-outubro-2013]. URL: http://www.dgmlive.com/help.htm#whatisbittorrent.

[32] Wikibooks. *The World of Peer-to-Peer (P2P) --- Wikibooks, The Free Textbook Project*. [Online; acessado em 3-outubro-2013]. 2012. URL: http://en.wikibooks.org/w/index.php?title=The_World_of_Peer-to-Peer_(P2P)&oldid=2316492.

[33] Wikipedia. *Anycast --- Wikipedia, The Free Encyclopedia*. [Online; acessado em 2-outubro-2013]. 2013. URL: http://en.wikipedia.org/w/index.php?title=Anycast&oldid=574680440.

[34] Wikipedia. *Audiogalaxy --- Wikipedia, The Free Encyclopedia*. [Online; acessado em 29-setembro-2013]. 2013. URL: http://en.wikipedia.org/w/index.php?title=Audiogalaxy&oldid=560950036.

[35] Wikipedia. *BitTorrent --- Wikipedia, The Free Encyclopedia*. [Online; accessed 28-October-2013]. 2013. URL: http://en.wikipedia.org/w/index.php?title=BitTorrent&oldid=578877679.

[36] Wikipedia. *Bram Cohen --- Wikipedia, The Free Encyclopedia*. [Online; acessado em 7-outubro-2013]. 2013. URL: http://en.wikipedia.org/w/index.php?title=Bram_Cohen&oldid=574084830.

[37] Wikipedia. *EDonkey network --- Wikipedia, The Free Encyclopedia*. [Online; acessado em 3-outubro-2013]. 2013. URL: http://en.wikipedia.org/w/index.php?title=EDonkey_network&oldid=568576016.

[38] Wikipedia. *File sharing --- Wikipedia, The Free Encyclopedia*. [Online; acessado em maio de 2013]. 2013. URL: http://en.wikipedia.org/w/index.php?title=File_sharing&oldid=556034682.

[39] Wikipedia. *Gnutella2 --- Wikipedia, The Free Encyclopedia*. [Online; acessado em 2-outubro-2013]. 2013. URL: http://en.wikipedia.org/w/index.php?title=Gnutella2&oldid=556794729.

[40] Wikipedia. *Gnutella --- Wikipedia, The Free Encyclopedia*. [Online; acessado em 2-outubro-2013]. 2013. URL: http://en.wikipedia.org/w/index.php?title=Gnutella&oldid=574304390.

[41] Wikipedia. *Hash function --- Wikipedia, The Free Encyclopedia*. [Online; acessado em 5-outubro-2013]. 2013. URL: http://en.wikipedia.org/w/index.php?title=Hash_function&oldid=574871670.

[42] Wikipedia. *Hash table --- Wikipedia, The Free Encyclopedia*. [Online; acessado em 5-outubro-2013]. 2013. URL: http://en.wikipedia.org/w/index.php?title=Hash_table&oldid=575499828.

[43] Wikipedia. *Internet service provider --- Wikipedia, The Free Encyclopedia*. [Online; acessado em 29-setembro-2013]. 2013. URL: http://en.wikipedia.org/w/index.php?title=Internet_service_provider&oldid=573549991.

[44] Wikipedia. *Kademlia --- Wikipedia, The Free Encyclopedia*. [Online; acessado em 3-outubro-2013]. 2013. URL: http://en.wikipedia.org/w/index.php?title=Kademlia&oldid=575742258.

[45] Wikipedia. *Motion Picture Association of America --- Wikipedia, The Free Encyclopedia*. [Online; acessado em 7-outubro-2013]. 2013. URL: http://en.wikipedia.org/w/index.php?title=Motion_Picture_Association_of_America&oldid=575124240.

[46] Wikipedia. *MP3.com --- Wikipedia, The Free Encyclopedia*. [Online; acessado em 29-setembro-2013]. 2013. URL: http://en.wikipedia.org/w/index.php?title=MP3.com&oldid=571025541.

[47] Wikipedia. *MP3 --- Wikipedia, The Free Encyclopedia*. [Online; acessado em 29-setembro-2013]. 2013. URL: http://en.wikipedia.org/w/index.php?title=MP3&oldid=574123988.

[48] Wikipedia. *Napster --- Wikipedia, The Free Encyclopedia*. [Online; acessado em 30-setembro-2013]. 2013. URL: http://en.wikipedia.org/w/index.php?title=Napster&oldid=573999770.

[49] Wikipedia. *Percent-encoding --- Wikipedia, The Free Encyclopedia*. [Online; accessed 27-October-2013]. 2013. URL: \url{http://en.wikipedia.org/w/index.php?title=Percent-encoding&oldid=573113056}.

[50] Wikipedia. *Recording Industry Association of America --- Wikipedia, The Free Encyclopedia*. [Online; acessado em 30-setembro-2013]. 2013. URL: http://en.wikipedia.org/w/index.php?title=Recording_Industry_Association_of_America&oldid=574192660.

[51] Wikipedia. *Segmented file transfer --- Wikipedia, The Free Encyclopedia*. [Online; acessado em 5-outubro-2013]. 2013. URL: http://en.wikipedia.org/w/index.php?title=Segmented_file_transfer&oldid=533100656.

[52] Wikipedia. *Timeline of file sharing --- Wikipedia, The Free Encyclopedia*. [Online; acessado em 28 de setembro de 2013]. 2013. URL: http://en.wikipedia.org/w/index.php?title=Timeline_of_file_sharing&oldid=571061187.

[53] Wikipédia. *BitTorrent --- Wikipédia, a enciclopédia livre*. [Online; accessed 22-outubro-2013]. 2013. URL: http://pt.wikipedia.org/w/index.php?title=BitTorrent&oldid=36953538.

[54] Theory.org Wiki. *BitTorrentSpecification --- Theory.org Wiki,* [Online; accessed 17-October-2013]. 2013. URL: https://wiki.theory.org/index.php?title=BitTorrentSpecification&oldid=2527#Bencoding.

[55] Theory.org Wiki. *BitTorrentSpecification --- Theory.org Wiki,* [Online; accessed 17-October-2013]. 2013. URL: https://wiki.theory.org/index.php?title=BitTorrentSpecification&oldid=2527#Tracker_Response.

Capítulo 7

Visão Pessoal