

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií

Analýza BitTorrent služby KickassTorrents

Projekt k predmetu ISA 2015

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
1.1	Cieľ . . . . .	2
<b>2</b>	<b>Bittorrent protokol</b>	<b>3</b>
2.1	Proces zdieľania súborov . . . . .	3
2.2	Tracker protokol . . . . .	3
2.2.1	Request . . . . .	3
2.2.2	Bencoding . . . . .	4
2.2.3	UrlEncoding . . . . .	4
<b>3</b>	<b>Návrh aplikácie</b>	<b>5</b>
3.1	Účel a technika . . . . .	5
3.2	Použíte technológie . . . . .	5
<b>4</b>	<b>Implementácia</b>	<b>6</b>
4.1	Spracovanie RSS súboru . . . . .	6
4.2	Spracovanie torrent súboru . . . . .	6
4.3	Získanie peerlistu . . . . .	6
4.3.1	Získanie zoznamu trackerov . . . . .	6
4.3.2	Príprava dát . . . . .	6
4.3.3	Komunikácia . . . . .	6
4.3.4	Spracovanie dát . . . . .	7
4.4	Problematika . . . . .	7
4.4.1	Spojenie s tretími stranami . . . . .	7
4.4.2	Nevalidné vstupy aplikácie . . . . .	7
4.5	Inštalácia . . . . .	7
4.6	Použitie . . . . .	8
<b>5</b>	<b>Záver</b>	<b>9</b>
<b>6</b>	<b>Použitá literatúra</b>	<b>10</b>

# 1 Úvod

Zadaním projektu je vytvoriť vhodný nástroj určený k monitorovaniu služby bittorrent na konkrétnom serveri KickassTorrent (ďalej len server) v sekcii Movie Torrents. Výsledná aplikácia je riešením akademického projektu, avšak ponúka plné praktické riešenie zadanej problematiky.

Táto dokumentácia má za úlohu poskytnúť ďalšie informácie o problematike zadania, základné informácie o programe a popis implementácie. Tieto časti sú spracované v jednotlivých sekciách. Návod na použitie programu je špecifikovaný v priloženom súbore README.

## 1.1 Cieľ

Úlohou aplikácie je pri spustení využiť RSS feed poskytovaný serverom k stiahnutiu informácií o danej sekcii. Alternatívnou možnosťou je spustenie programu nad už stiahnutým RSS Feed. Vďaka informáciám uloženým v tomto súbore je možné zistiť informácie o jednotlivých torrentoch a teda stiahnuť ich. Stiahnutím alebo otvorením existujúceho torrent súboru je možné zistiť všetky meta informácie o torrente a teda vyfabrikovať požiadavku alebo využiť podvrhnutú požiadavku na server, vďaka ktorej server poskytne všetky informácie o peeroch - teda užívateľoch zapojených do zdieľania obsahu torrentu.

## 2 Bittorrent protokol

Bittorrent protokol [4] je populárny protokol určený ku zdieľaniu dát formou peer-to-peer čo znamená, že užívateľ môže sťahovať dáta nielen zo serveru ale aj od ostatných užívateľov, ktorý tieto dáta majú.

Protokol sa často využíva k nelegálnej činnosti [1] preto sa teší veľkej popularite a vzniká množstvo stránok poskytujúcich tento obsah a veľké množstvo klientov zapojených do tejto činnosti. Toto konanie je porušovaním autorských práv a teda nežiadúce. Úlohou je teda monitorovať tieto aktivity a poskytnúť zoznam účastníkov na tejto činnosti.

### 2.1 Proces zdieľania súborov

Zdieľanie súborov [2] pozostáva z týchto častí:

**autor** Originálny autor dát, ktorý vytvorí torrent súbor metadát a ďalej ho šíri zdieľaním cez server.

**torrent súbor** Súbor metadát obsahujúci informácie o jednotlivých častiach zdieľaného obsahu a informáciach o ňom. Tento súbor sa uloží na server poskytujúci hosting, ktorý zabezpečí bežiaci tracker a následne stačí iba šíriť odkaz.

**Server** Typický server má dve úlohy: poskytovať hosting pre torrent súbory a zverejňovať odkazy na ne.

**Rss Feed** Rss feed obsahuje aktuálne informácie o dostupných torrentoch.

**BitTorrent tracker** Tracker obsahuje zoznam peerov s aktuálnym obsahom.

**užívatelia (peers)** Peers sú užívatelia, ktorý sťahujú a zdieľajú nelegálny obsah.

### 2.2 Tracker protokol

Tracker reaguje na požiadavky GET a ich parametre [3].

#### 2.2.1 Request

Požiadavka na tracker je zasielaná ako štandardný GET zasielaný na adresu trackeru s nasledujúcimi parametrami:

**info\_hash** Z hodnota kľúča info z torrent súboru zašifrovanej cez bencode sa vytvorí sha1 hash. Ten je následne zašifrovaný cez Url-encode.

**peer\_id** Náhodný 20-znakový reťazec špecifikujúci id užívateľa. Náhodne sa generuje pri začiatku a následne sa šifruje pomocou Url-Encode.

**uploaded** Celkové množstvo odoslaných dát v desiatkovej sústave.

**downloaded** Celkové množstvo stiahnutých dát v desiatkovej sústave.

**left** Celkové množstvo zvyšných dát v desiatkovej sústave.

**event** Reprezentuje akciu sťahujúceho, môže byť špecifikované ako *started*, *completed* alebo *stopped*, pre reprezentáciu začiatku, konca alebo zastavenia. V prípade nešpecifikovanej hodnoty to značí správu zasielanú v pravidelných intervaloch.

**numwant** Množstvo peer-ov, ktoré užívateľ požaduje od trackeru.

**compact** Príznak pre zasielanie dát v kompaktnej forme. V tomto prípade sa data zasielajú ako jeden reťazec reprezentujúci všetky adresy a porty peer-ov.

### 2.2.2 Bencoding

Bencoding sa využíva pri kódovaní metadát torrent súboru a odpovedi trackeru. Je to spôsob organizácie dát v úspornej forme využívaný pri komunikácii v bittorrent protokole. Podporuje tieto datové typy: bytový reťazec, integer, zoznam a slovník.

Integer sa ukladá ako celé číslo uložené v desiatkovej sústave ohraňované tagmi „i“ a „e“.

Reťazec sa ukladá vo forme <dĺžka reťazca>:<obsah reťazca>.

Zoznam obsahuje predošlé datové typy a je ohraňovaný tagmi „l“ a „e“.

Slovník pozostáva z bitových reťazcov zoradených podľa abecedy. Prvky sú uložené v slovníku ako kľúč nasledovaný jeho hodnotou.

### 2.2.3 UrlEncoding

UrlEncode je spôsob kovania dát pri posielaní požiadavku na server v tomto prípade konkrétne na tracker. Slúži ku zakódovaniu požiadavku aby ho bolo možné správne rozoznať od url.

UrlEncode zakóduje všetky ne-ASCII znaky pomocou % nasledovaného ich hexa-decimálnou hodnotou.

Keďže medzery sa nemôžu vyskytovať v url tieto sa zakódujú pomocou znaku + alebo %20.

## 3 Návrh aplikácie

### 3.1 Účel a technika

Zatiaľ, čo sa vedie legálny boj s poskytovateľmi služieb a majiteľmi serverov, je legislatívny boj veľmi náročný a neefektívny. Zatvorením jedného serveru sa nezabráni vzniku alternatív, alebo presunutiu obsahu inam.

Využitie verejnosti celého protokolu vedie k automatizovaniu prístupu k serveru a možnosti autonómne sťahovať a vyhodnocovať meta informácie o torrentoch. Vzhľadom k tomu, že väčšina populárnych trackerov je verejná a práve toto slabinou tejto služby. Podvrhnutím požiadavku pre tracker je možné získať zoznam užívateľov k nemu pripojených. Tejto technike nejde jednoducho zabrániť a jediným problém ostáva legislatívne riešenie prehrškov/ich delegácia autoritám.

Problematickou ostáva prístup k súkromným trackerom ,kde je prístup možný iba na pozvanie alebo inou formou zaručujúcou utajenie tohto trackeru. Pri jednotlivých požiadavkoch na tracker tento potom kontroluje ich autenticitu.

### 3.2 Použité technológie

Aplikácia je riešená v jazyku python, na základe jednoduchosti riešenia, minimálnej nadbytočnej réžií a možnosti ďalšieho širšieho rozširovania. Pri riešení boli využité ďalšie externé knižnice, ktoré sú súčasťou riešenia. Riešenie samotnej problematiky operácií popísaných v jednotlivých moduloch je poskytnuté samostatne na základe zadania. Existujúce alternatívy teda môžu nahradiť manuálne spracovanie RSS feedu s využitím knižnice pre operáciu s XML a posielanie požiadavku na tracker.

## 4 Implementácia

Riešenie úlohy je rozdelené na tri časti:

- Získanie a parsovanie RSS feedu
- Získanie a parsovanie torrent súboru
- Kontaktovanie trackeru a parsovanie odpovedi

V tejto implementácii som zvolil rozčlenenie do jednotlivých modulov na základe tohto riešenia. ďalej popíšem problematiku a implementačné detaily a jednotlivé moduly popisujúce implementáciu.

### 4.1 Spracovanie RSS súboru

Modul RSSparse je zodpovedný za prípravu obsahu RSS súboru jeho stiahnutím alebo otvorením existujúceho a vygenerovanie súboru *movies\_announce\_list.txt*.

Využíva sa knižnica *xml.dom.minidom* na prehľadávanie stromovej štruktúry na záznamy „item“. Vytvorenie zoznamu je rozdelené na dve časti: header, item. Pri generovaní zoznamu sa kontrolujú atribúty, ktoré je potrebné do zoznamu zaradiť a cyklom sa predpripravia dáta, ktoré sa následne zapíšu do súboru.

### 4.2 Spracovanie torrent súboru

Dvojica funkcií *openTorrent(location)* a *getTorrent(url)* zabezpečuje načítanie dát torrentu.

Celková ďalšia práca programu je založená na dátach z torrent súboru bez nutnosti xml súboru kvôli podpore rozšírenia -t.

### 4.3 Získanie peerlistu

#### 4.3.1 Získanie zoznamu trackerov

Zoznam trackerov sa prioritne získava z voliteľného parametru -a programu. Podľa špecifikácie sa následne prehľadáva kľúč announce-list, v prípade jeho prázdnoty kľúč announce. Pred ďalším spracovaním dochádza ku zamiešaniu zoznamu.

#### 4.3.2 Príprava dát

Pre každú položku zoznamu trackerov sa vytvoria dáta potrebné pre komunikáciu z dát torrentu.

Spracovaním dát torrentu sa získa query odosielaný ako parametre GET requestu.

Spracovaním url sa získa: protokol, adresa, port, cesta a ďalšie parametre. Kvôli rozličnosti trackerov je spracovávanie url dôležité kvôli voliteľným položkám ako je port, prípadne ďalšie parametre, ktoré je treba odoslať spolu s požiadavkom.

Vytvorenie kompletnej správy pozostáva z naformátovania položiek do potrebné tvaru a vrátenia ako celku. Správa pozostáva z týchto položiek: cesta, adresa, query a parametre. Výsledkom je GET požiadavok na tracker.

#### 4.3.3 Komunikácia

Na základe protokolu dochádza ku spojeniu s trackerom a odoslaniu požiadavku.

Prijímanie odpovedi trackeru je rozdelené na prijatie hlavičky a tela správy. Počas prijímania hlavičky sa určuje dĺžka tela a typ odpovedi.

Komunikácia s trackerom je náchylná na chyby, preto dochádza ku ošetrovaniu chyby spojenia ukončením timeout. Počas experimentovania s trackermi som došiel k záveru, že za normálnych podmienok nie je potrebné mať túto dobu dlhšiu ako 10 sekúnd.

V prípade chyby spojenia pri vyvolaní výnimky alebo detekovaní zlého stavu dochádza ku výpisu chybovej hlášky a preskočeniu trackeru v zozname.

#### 4.3.4 Spracovanie dát

V požiadavke na server je špecifikovaný compact parameter, ktorý požaduje zaslanie odpovedi v tzv. compact forme ktorá vracia jeden reťazec zložený z informácií o peeroch vo forme 4bity ip a 2bity port.

Počas testovania som však zistil, že tracker môže aj napriek tomu reagovať zaslaním odpovede ako zoznamu. V tomto prípade sa kontroluje formát odpovede a dáta sa spracovávajú prechodom zoznamom a prístupom do položiek ip a port.

### 4.4 Problematika

Ďalej sú popísané problémy s používaním aplikácie, jej vývojom alebo ďalšie problémy ktoré nastali počas návrhu aplikácie.

#### 4.4.1 Spojenie s tretími stranami

Jedným z problémov, ktoré nastali počas implementácie je závislosť na tretích stranách. Toto sa odráža na problémoch pripojenia na server, kedy tento server nereaguje a spojenie vyprší. Vďaka povahe aplikácie autor ani užívateľ aplikácie na toto nemajú vplyv a možno riešiť iba dôsledky tohto chovania.

Z rovnakého dôvodu nastáva teda problém pri spracovaní RSS súboru a to konkrétne v prípade, že medzi udalosťou zmazania torrentu zo serveru a aktualizáciou RSS feedu nastáva zpracovanie feedu a teda požiadavka na stiahnutie torrent súboru. Toto riešenie taktiež nemožno obmedziť je závislé na tretej strane a aktuálnom stave serveru.

Tretím problémom rovnakej povahy je nedostupnosť trackeru, ktorý nastáva najmä pri starších torrentoch, poprípade iný problém kedy nemožno štandardne reagovať na jeho odpoveď, alebo nemožno získať odpoveď.

Navrhované riešenie: Pri komunikácii s tretími stranami vhodne zvoliť štandardnú dobu ako timeout. Po uplynutí timeout-u, chybe spojenia sa o tejto udalosti informuje užívateľ a spracovanie tejto udalosti sa preskočí.

#### 4.4.2 Nevalidné vstupy aplikácie

Za istých podmienok nastáva situácia kedy nenastane problém v spojení s trackerom, ale vrátia sa nekorektné dáta.

Aplikácia na tento stav reaguje upozornením užívateľa a preskočením spracovávania dát alebo skončením aplikácie podľa závislosti na týchto dátach.

### 4.5 Inštalácia

Potrebné knižnice sú súčasťou archívu, teda inštalácia spočíva iba z nastavenia práv chmod +x.



## 4.6 Použitie

Výsledkom je konzolová aplikácia, ktorej chod je možno parametrizovať vstupom od užívateľa. Použitie programu s parametrami je nasledovné: `./antipirat ((-r url—i filename—t filename) [-a url] — -h)`

Povinné parametry (r, i, t) určujú vstup programu, parameter -a je doplnkom ku jednému z týchto parametrov a špecifikuje url používanú pri komunikácii s trackerom.

Parameter -r špecifikuje url, z ktorej sa sťahuje požadovaný rss feedom, ktorý je potom parsovaný a generuje sa výstup.

Parameter -i špecifikuje už stiahnutý rss feed.

Parameter -t špecifikuje torrent súbor už stiahnutý, teda nedochádza k stiahnutiu prvého záznamu z rss feedu

Parameter -a špecifikuje podvrhnutú url určenú ku komunikácii s trackerom.

Parameter -h zobrazí nápovedu aplikácie

## 5 Záver

Výsledkom tejto práce je aplikácia vyhovujúca požiadavkom a špecifikácií, táto dokumentácia a manuál na použitie. Prínosom je funkčné riešenie, ktoré bez znalosti BitTorrent protokolu prinesie očakávaný výsledok pre užívateľa. Súčasťou riešenia je aj testovacia sada pre použitie s funkčnými testami aplikácie.

Možné rozšírenia do budúcnosti zahŕňujú ďalšiu automatizáciu funkcionality v kombinácii so serverom riešením. Odpadla by potreba túto aplikáciu manuálne spúšťať a vznikla by možnosť neustáleho autonómneho monitorovania danej sekcie. V prípade potreby existuje možnosť jednoduchého pridania štatistík a zbierania dát.

Vďaka architektúre aplikácie je možné ju jednoducho ďalej rozširovať, či využívať v iných moduloch. Riešenie v jazyku python zaručuje čiastočnú prenositeľnosť nad zariadeniami s nainštalovaným interpretom jazyka.

## 6 Použitá literatura

### Reference

- [1] *BitTorrent* [online]. [cit. 2015-11-22]. Dostupné na: <[https://en.wikipedia.org/wiki/BitTorrent#Legal.i](https://en.wikipedia.org/wiki/BitTorrent#Legal_issues)>.
- [2] *The BitTorrent Protocol Specification - BEP003* [online]. [cit. 2015-11-22]. Dostupné na: <[http://www.bittorrent.org/beps/bep\\_0003.html](http://www.bittorrent.org/beps/bep_0003.html)>.
- [3] *BitTorrent Unofficial Specification* [online]. [cit. 2015-11-22]. Dostupné na: <[https://wiki.theory.org/BitTorrentSpecification#Tracker\\_Request\\_Parameters](https://wiki.theory.org/BitTorrentSpecification#Tracker_Request_Parameters)>.
- [4] COHEM, B. *Incentives Build Robustness in BitTorrent* [online]. 05/2003 [cit. 2015-11-22]. Dostupné na: <<http://www.bittorrent.org/bittorrentecon.pdf>>.