IsTorrent

< iskandar@arc.itb.ac.id | iskandarsetiadi@students.itb.ac.id >

@author : Iskandar Setiadi (freedomofkeima)

A lightweight and easy to use Torrent Applications

Recommended for ones who want to learn the basic concept of torrent

Distributed System Laboratory Assistant Selection Test

Sister - 妹 lab

Institute Technology of Bandung

Jl. Ganesha No. 10, Bandung 40132

Year 2013

Document Change Records

Version	Date	Author	Description
0.1	03-08-13	IS	Initial

Contents

I. Introduction

Nowadays, there are a lot of torrent applications such as Bit Torrent, uTorrent, etc. In brief description, torrent application is a peer-to-peer file sharing protocol which is being used for distributing large amount of data. Statistic states that approximately 27-55% of all Internet traffic is dominated with torrent transactions.

Torrent protocol allows user to distribute their data without putting the level of strain on their servers / computers. This protocol makes even device with low bandwidth capable to use these application.

Terminology

Torrent is a small metadata file which contains information about the data you want to download, not the data itself (as the name states that it's a meta). IsTorrent file extension is defined as **.keima**. Usually, these files are hundred times smaller its real data.

The other common terminology is **peer**. Simply said, peer is all connected device on the internet that is sharing the file you wish to download. Peer usually does not have the complete file, or they will be called seeder.

Seeder is a device that has a complete copy of the specific Torrent you are downloading. A person who creates new torrent will be automatically called as a seeder (as he/she will actually have the complete parts of its data).

Swarm is a terminology which is being used in referring all people who are connected to the same clients for a torrent file. Swarm is a combination of seeder and leeches.

Tracker serves as the central coordination of all clients. The detailed explanation regarding tracker will be explained in IsTracker.

II. IsTracker

As the central system of torrent application, tracker coordinates the action of torrent clients. IsTorrent is using IsTracker as its main tracker. Built in PHP, almost every Web server can accommodate to use IsTracker.

IsTracker is mainly divided into two files, **announce.php** and **db.php**. Main flow of this tracker is written below:

Get database configuration from db.php → resolve clients IP address → validate clients data → update tracker (MySQL) database → fetch peers data from database → return peers data to client in JSON format

To make this application easier to understand, db.php is storing five attributes:

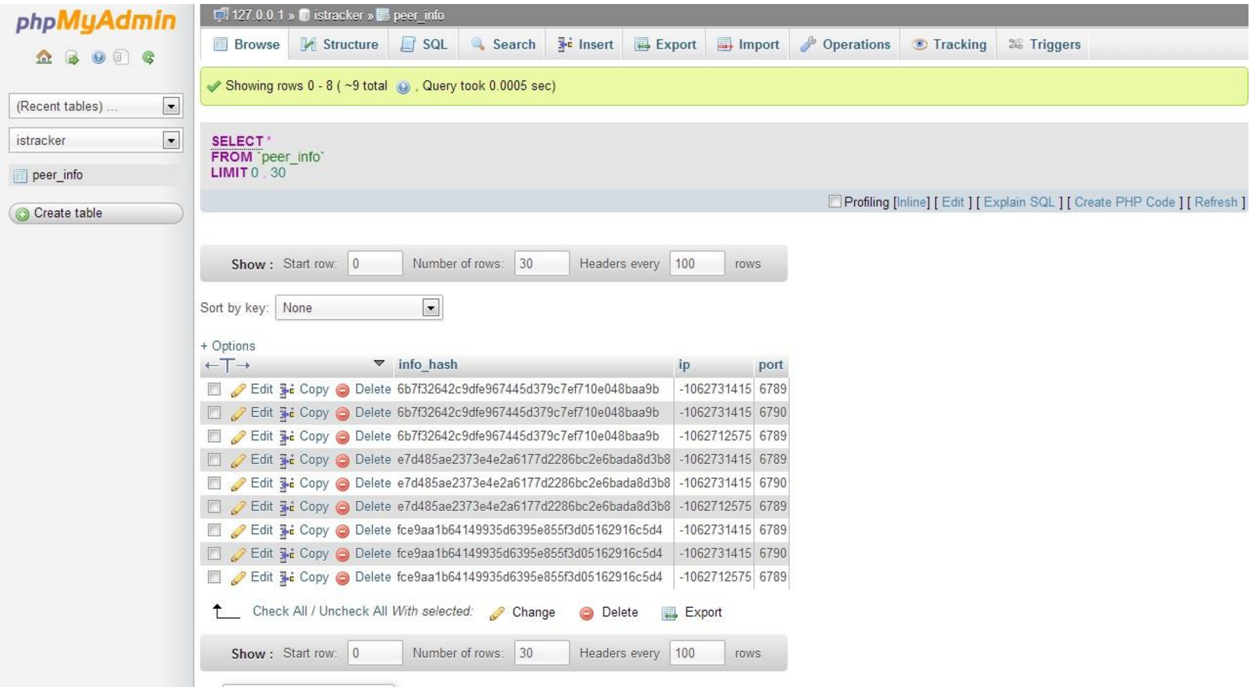
- $db_ip	:	IP address of database location (default : localhost)
- $db_user	:	username of database (default : root)
- $db_pass	:	password of database (default : -)
- $db_database	:	IsTracker database name (default : IsTracker)
- $db_table	:	table name inside main database (default : peer_info)

Clients IP address is resolved using *resolve_ip($host)* function.
In this torrent application, we only store three attributes from clients (which are needed to be validated):

- info_hash	:	hash from a specific torrent (VAR CHAR 40)
- ip	:	IP address of the client (INT 11)
- port	:	Incoming port of the client (INT 5)

In case these information hasn't been stored in IsTracker, this tracker will insert these information into its database.

IsTracker will fetch peers data from database, limited to 50 entries by random. Using PHP function (*json_encode*), IsTracker will return these peers information to client.


phpMyAdmin
(Recent tables) ...
istracker
peer_info
Create table
127.0.0.1 » istracker » peer_info
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Tracking
Triggers
□ Profiling [Inline] [Edit] [Explain SQL] [Create PHP Code] [Refresh]
+ Options
Check All / Uncheck All With selected: Change Delete Export

✓ Showing rows 0 - 8 (~9 total , Query took 0.0005 sec)

```
SELECT *
FROM `peer_info`
LIMIT 0 , 30
```

Show : Start row: 0 Number of rows: 30 Headers every 100 rows

Sort by key: None

	info_hash	ip	port
□ Edit Copy Delete	6b7f32642c9dfe967445d379c7ef710e048baa9b	-1062731415	6789
□ Edit Copy Delete	6b7f32642c9dfe967445d379c7ef710e048baa9b	-1062731415	6790
□ Edit Copy Delete	6b7f32642c9dfe967445d379c7ef710e048baa9b	-1062712575	6789
□ Edit Copy Delete	e7d485ae2373e4e2a6177d2286bc2e6bada8d3b8	-1062731415	6789
□ Edit Copy Delete	e7d485ae2373e4e2a6177d2286bc2e6bada8d3b8	-1062731415	6790
□ Edit Copy Delete	e7d485ae2373e4e2a6177d2286bc2e6bada8d3b8	-1062712575	6789
□ Edit Copy Delete	fce9aa1b64149935d6395e855f3d05162916c5d4	-1062731415	6789
□ Edit Copy Delete	fce9aa1b64149935d6395e855f3d05162916c5d4	-1062731415	6790
□ Edit Copy Delete	fce9aa1b64149935d6395e855f3d05162916c5d4	-1062712575	6789

Show : Start row: 0 Number of rows: 30 Headers every 100 rows

Picture 2-1 IsTracker as seen with phpMyAdmin

Each main application which wants to add / create a new torrent will access this tracker from http://[ip_address]:[port]/IsTracker/announce.php on default.

III. Main Application (IsTorrent)

The main application of our torrent is IsTorrent. Using **.keima** as its torrent, IsTorrent is built using Java as its backbone.

You may use "*java –jar res.jar*" to start this application.

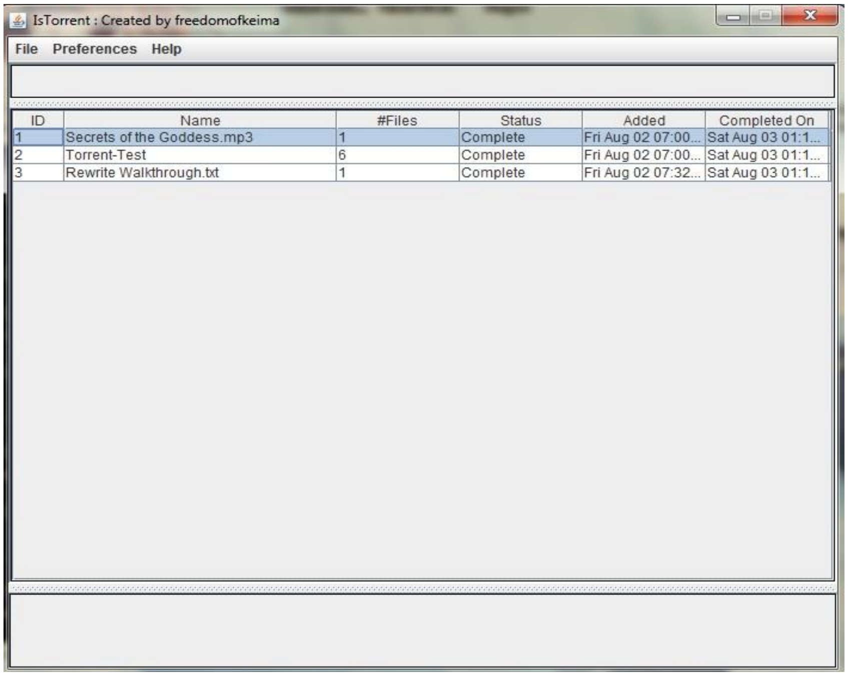
IsTorrent consists of **15 classes** and **json_simple.jar** library, which can be divided into this categorization:

Main	Main.java	Main class, activating protocol & interface
Protocol & Service	DownloadService.java	Handling seeding and leeching mechanism, especially handling the connection between TCPClient and TCPServer
	TCPClient.java	Used for receive file data from peers
	TCPServer.java	Used for send file data to peers
	TorrentPreferences.java	Default save directory, default tracker, block size, default port, update time, and number of clients thread per object
Interface	UserInterface.java	Handling UI in application
Functional	AddTorrent.java	Add **.keima** to application (if it hasn't been added previously)
	NewTorrent.java	Create a new **.keima** from file / directory, automatically seed those file / directory contents
Model	TorrentFile.java	Model for each file in a torrent
	TorrentModel.java	Main model for a torrent (may consists of multiple TorrentFile model)

	TableModel.java	Model for torrent table in UI
Helper	FileHelper.java	Helper for file
	FileWalker.java	Helper for recursively read directory contents
	FileWriterHelper.java	Helper for writing file such as **db.dat** or temporary torrent files information
	JSONHelper.java	Helper for parsing JSON

Basically, this application starts with its TCPServer activation, followed with its main interface within **UserInterface.java**.

In UserInterface class, you can see two main methods, that's it, *initSingleton()* and *initComponents()*. All information regarding previously added torrents is being loaded into application when *initSingleton()* is called. When *initComponents()* is called, you'll see the following interface:



ID	Name	#Files	Status	Added	Completed On
1	Secrets of the Goddess.mp3	1	Complete	Fri Aug 02 07:00...	Sat Aug 03 01:1...
2	Torrent-Test	6	Complete	Fri Aug 02 07:00...	Sat Aug 03 01:1...
3	Rewrite Walkthrough.txt	1	Complete	Fri Aug 02 07:32...	Sat Aug 03 01:1...

Picture 3-1 IsTorrent User Interface

There are two main components in this interface (MenuBar and Torrent table information). You may create a new torrent file and add **.keima** into this application via “File” in MenuBar. All information regarding your torrent will be added into torrent table information, which is consisting ID, Name, #Files, Status, Added On, and Completed On.

So, what is the main content of **.keima** file?

Single file

To make our explanation easier, see the following **.keima** contents:

```
Rewrite Walkthrough.txt

1

Rewrite Walkthrough.txt

1

1 : 2efedb2100fe8cca932e3b5c8c98e69e47ff7d74

afc0439720e02ff6bc983c56511a89f90b42171d

http://localhost/PrivateTorrent/IsTracker/announce.php

Created by freedomofkeima
```

The meta-info file of our torrent consists of this following information:

- Torrent name : Rewrite Walkthrough.txt
- Torrent number of files : 1
- Torrent name of file : Rewrite Walkthrough.txt (Single file = torrent name)
- Torrent number of pieces : 1
- Torrent pieces hash : 2efedb2100fe8cca932e3b5c8c98e69e47ff7d74 (piece #1)
- Torrent main hash : afc0439720e02ff6bc983c56511a89f90b42171d
- Torrent tracker : http://localhost/PrivateTorrent/IsTracker/announce.php
- Torrent credits / info : Created by freedomofkeima

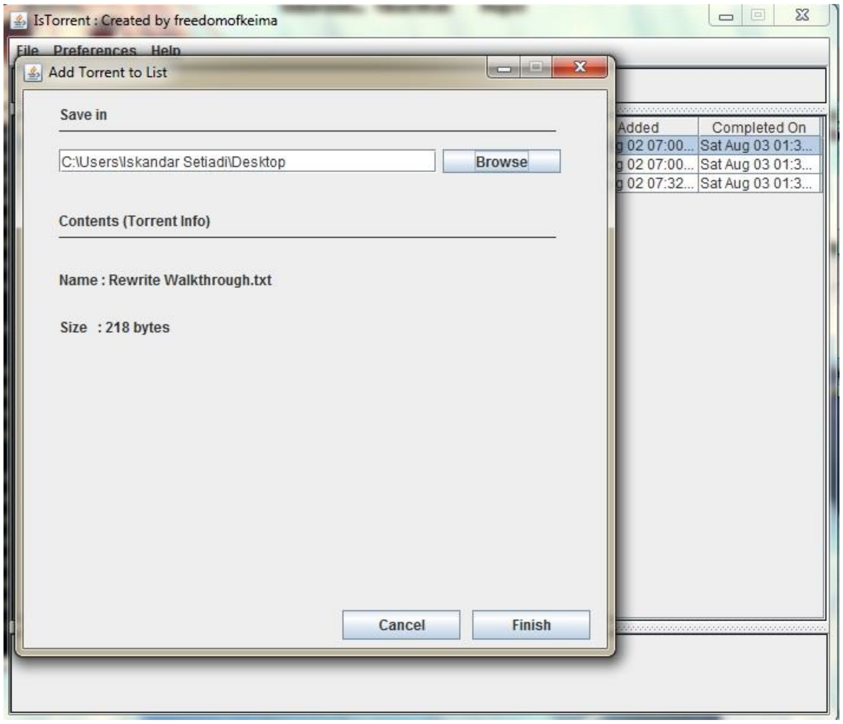
Directory / Multiple files

```
Torrent-Test
2
\Basic Skills & TPA TBS vol 1.pdf
2
1 : 5cdd05ffc4a6a234ac25f2c484d150eab63bf231
2 : 2537d7ac7482340893eb4e3eb20f966db351f73a
98d97fb6341262b97cca5eebb18d5c5c8cde7095
\Basic Skills & TPA TBS vol 2.pdf
3
1 : f53cb6c949816c6e7e5afa57537ec2771339eabd
2 : 50351518720a261c6f4a0a80d5b32f3e679cc04b
3 : 303f1b2caaa058b386d71bc8059f98ff8ca7cf4b
1f53ad52ca3ca78be84b800efcd21765ec7455e1
http://localhost/PrivateTorrent/IsTracker/announce.php
Created by freedomofkeima
```

The main difference between single and multiple files is that multiple files torrent consists of multiple torrent name of file, torrent number of pieces, torrent pieces hash, and torrent main hash information.

The information above is being stored as **ArrayList** of **TorrentModel**, which consists of multiple **TorrentFile** for directory / multiple files category. Each model in this list will have its own **DownloadService** to maintain download and upload activity.

Two main functions in this application is creating a new torrent and adding existing torrent to our download list. **AddTorrent.java** will handle all operation regarding existing torrent information (torrent → application database), while **CreateTorrent.java** will handle all operation regarding new torrent information (files → torrent).



Save in

C:\Users\Iskandar Setiadi\Desktop

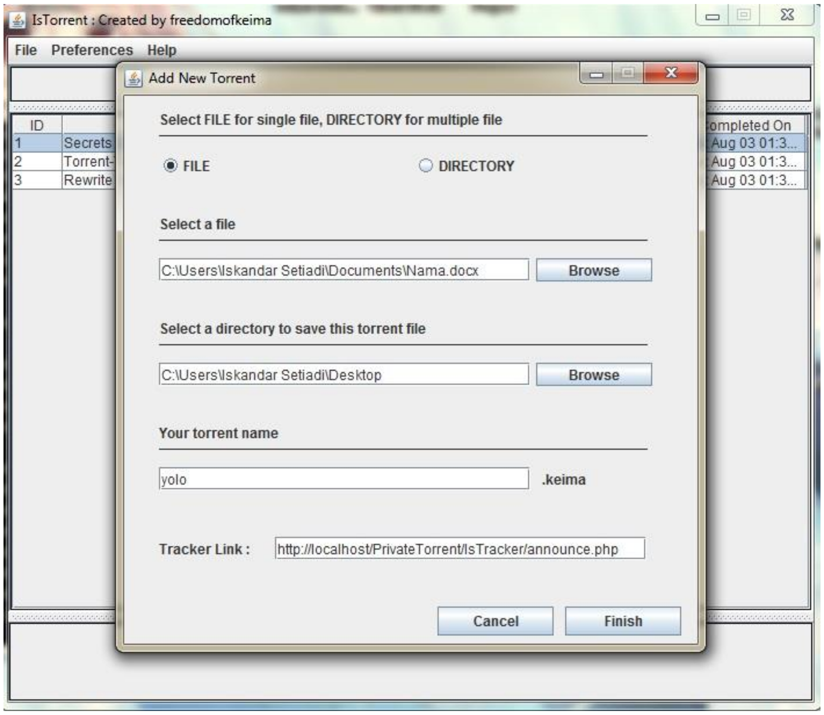
Contents (Torrent Info)

Name : Rewrite Walkthrough.txt

Size : 218 bytes

Added	Completed On
g 02 07:00...	Sat Aug 03 01:3...
g 02 07:00...	Sat Aug 03 01:3...
g 02 07:32...	Sat Aug 03 01:3...

Picture 3-2 IsTorrent Add Torrent Interface



ID		ompleted On
1	Secrets	Aug 03 01:3...
2	Torrent-	Aug 03 01:3...
3	Rewrite	Aug 03 01:3...

Select FILE for single file, DIRECTORY for multiple file

● FILE ○ DIRECTORY

Select a file

C:\Users\Iskandar Setiadi\Documents\Nama.docx

Select a directory to save this torrent file

C:\Users\Iskandar Setiadi\Desktop

Your torrent name

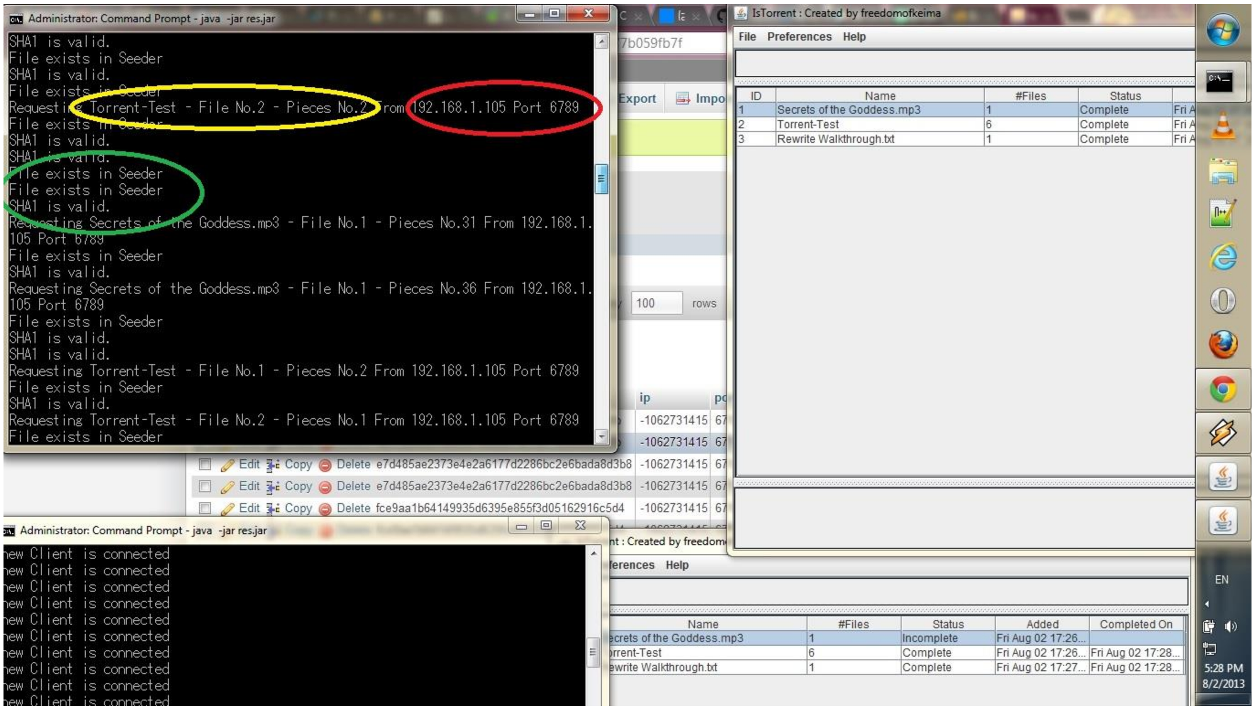
yolo .keima

Tracker Link : http://localhost/PrivateTorrent/IsTracker/announce.php

Picture 3-3 IsTorrent Create Torrent Interface

TableModel will handle all information regarding this application table interface. While helpers such as FileHelper, FileWalker, FileWriterHelper, and JSONHelper will help main application in performing each tasks.

Finally, the main functionality of torrent application is absolutely its peer-to-peer protocol for transferring files.


Administrator: Command Prompt - java -jar res.jar
IsTorrent : Created by freedomofkeima
File
Preferences
Help
Export
Impo
100
rows
EN
5:28 PM
8/2/2013

```
SHA1 is valid.
File exists in Seeder
SHA1 is valid.
File exists in Seeder
Requesting Torrent-Test - File No.2 - Pieces No.2 From 192.168.1.105 Port 6789
File exists in Seeder
SHA1 is valid.
SHA1 is valid.
File exists in Seeder
File exists in Seeder
SHA1 is valid.
Requesting Secrets of the Goddess.mp3 - File No.1 - Pieces No.31 From 192.168.1.
105 Port 6789
File exists in Seeder
SHA1 is valid.
Requesting Secrets of the Goddess.mp3 - File No.1 - Pieces No.36 From 192.168.1.
105 Port 6789
File exists in Seeder
SHA1 is valid.
SHA1 is valid.
Requesting Torrent-Test - File No.1 - Pieces No.2 From 192.168.1.105 Port 6789
File exists in Seeder
SHA1 is valid.
Requesting Torrent-Test - File No.2 - Pieces No.1 From 192.168.1.105 Port 6789
File exists in Seeder
```

```
□ Edit Copy Delete e7d485ae2373e4e2a6177d2286bc2e6bada8d3b8 -1062731415 67
□ Edit Copy Delete e7d485ae2373e4e2a6177d2286bc2e6bada8d3b8 -1062731415 67
□ Edit Copy Delete fce9aa1b64149935d6395e855f3d05162916c5d4 -1062731415 67
```

ID	Name	#Files	Status
1	Secrets of the Goddess.mp3	1	Complete
2	Torrent-Test	6	Complete
3	Rewrite Walkthrough.txt	1	Complete

```
new Client is connected
new Client is connected
new Client is connected
new Client is connected
new Client is connected
new Client is connected
new Client is connected
new Client is connected
new Client is connected
new Client is connected
```

Name	#Files	Status	Added	Completed On
ecrets of the Goddess.mp3	1	Incomplete	Fri Aug 02 17:26...	
rrent-Test	6	Complete	Fri Aug 02 17:26...	Fri Aug 02 17:28...
ewrite Walkthrough.txt	1	Complete	Fri Aug 02 17:27...	Fri Aug 02 17:28...

Picture 3-4 IsTorrent Transfer Protocol

Basically, **DownloadService** will ask for peers info in certain interval (defined in **TorrentPreferences** as *updateTime*). The red circle in Picture 3-4 shows a client (or peer) who also has this torrent (yellow circle) in their list. The yellow circle has three information regarding requested pieces (Name, #File, and #Piece). If the file exists, we'll see the following information in our prompt (green circle). Lastly, IsTorrent will validate the following piece with our SHA1 160-bit digest and return true if the downloaded piece is valid.

Each object in list will have certain number of downloader clients (**TCPClient**), which is defined in **TorrentPreferences** (default = 3). Finally, **DownloadService** will stop these downloader clients when all pieces have been downloaded successfully and switch into full seeding mode. Every certain time (default = 3 seconds), information table will refresh itself and update its value with newest data.

Our piece size (as set in **TorrentPreferences**) is 64 KB. It means that every 64 KB of data, we will have 160-bit SHA1 digest which is stored in our torrent file. Several study stated that pieces with sizes greater than 512 KB will reduce the size of a torrent file for a very large payload, but is claimed to reduce the efficiency of the protocol. Usually, this piece size varies between 32 KB and 4 MB each.

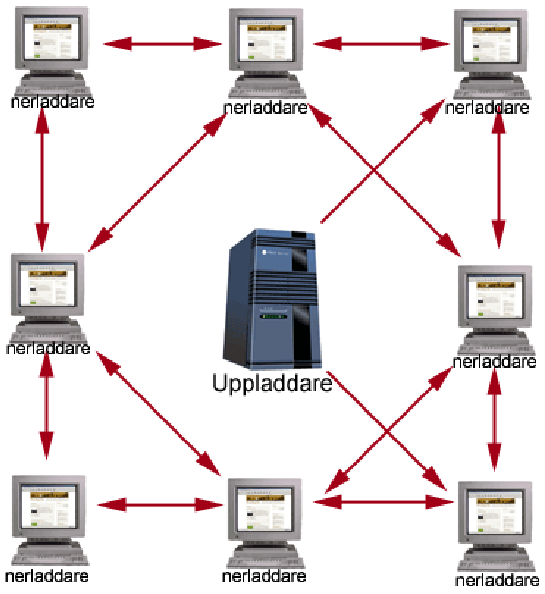
IV. How Torrent Works

Almost all torrent application is based on a protocol which supports the practice of Peer-to-peer (P2P) file sharing. IsTorrent is also working based on same concept. Simply said, rather than downloading a file from a single source server, torrent protocol allows users to join a swarm of hosts to download and upload from each file simultaneously.

Picture 4-1 Concept of Torrent

Earlier this morning, I've found this interesting picture from Internet. This is an analogy to how torrent actually works. In school, everyone needs to work on their homework, yet sometimes, you barely have enough time to finish all of them. For example, you've done with "A" homework, but you're only barely started with "B" and "C" homework. Now, you want to **leech** "B" and "C" homework from some of your friends who have already finished it. At the same time, they want to **leech** your "A" homework from you. This entire community is basically called as **swarm**.

In torrent concept, there's usually one person who acts as a first-time uploader and has a complete set of torrent files. After sending certain pieces of data to several users, other people who want the complete set of data may take these pieces from other users. This protocol brings one of the main advantages of torrent. You can download it faster and connection issue with server will not halt your downloading process.



Picture 4-2 Torrent Download and Upload Processes

Picture 4-2 represents how each client is connected and downloaded specific pieces from a torrent file.

All of these torrent specifications actually bring several drawbacks. It may reduce computer performance (by using multi-threading download processes) and require several additional ports to transfer data. A firewall may block some of these ports, which will lead to problem in transferring data. Another main drawback is every P2P software could not verify the legality of its contents. Sometimes torrent is being used to distribute illegal / pirated files throughout network. Yet, torrent protocol makes it more difficult for investigators to track down individual users who is actually created and distributed these files.

V. FAQ & Troubleshooting

This section is intended to be used for frequently asked question (FAQ) and troubleshooting.

Q : Is this application free?

A : Of course. This is an open source project. You can even modify it as long as my courtesy is still written there ☺

Q : Could I contribute in this project?

A: Yeah! You can access this project from git repository (http://github.com/freedomofkeima [still private project atm])

Q : How can I report error / bug / troubleshoot?

A : You may contact me via e-mail which are stated above (in cover page).

Q : Are you feeling fabulous?

A : Yes. I'm fabulous!

"I CAN SEE THE ENDING!" – KATSURAGI KEIMA