Understanding of BitTorrent Protocol

kevinkoo001@gmail.com

Overview

1. Introduction
2. Terminology
3. Protocol specification
4. Operation
5. Example
6. References

Introduction

Introduction

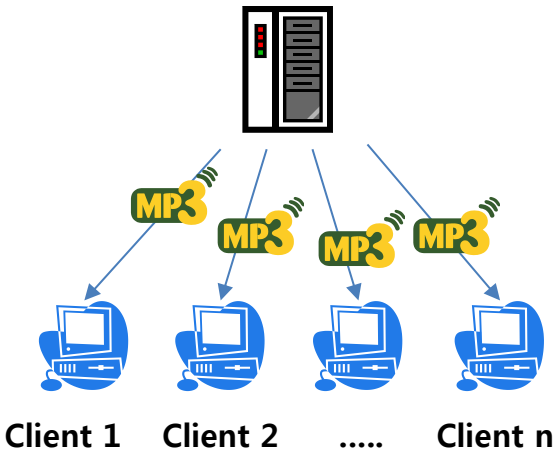
- History
  - Created by Bram Cohen in 2001
    - At the time p2p protocols only connected 2 peers to each other.
    - Speed limited due to one person's connection

- Brief Introduction
  - Estimated 150 million active users, approximately 250 million users (as of Jan. 2012)
  - Estimated that BitTorrent traffic accounts for roughly 35% of all traffic on the Internet. (http://www.zdnet.com/blog/itfacts/cachelogic-says-35-of-all-internet-traffic-is-now-bittorrent/6431)
  - Since 2010 200,000+ users have been sued for using the protocol to share copyrighted material

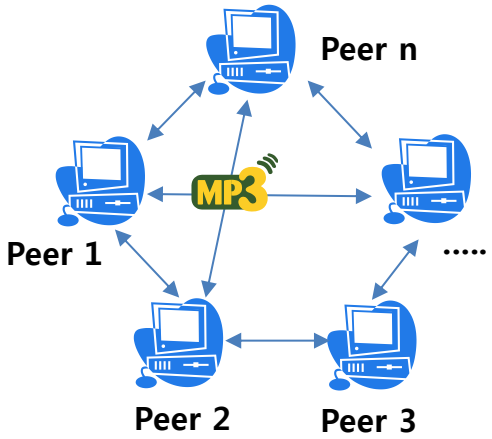
Introduction

- Description
  - Allows users to join a "swarm" of hosts to download and upload from each other simultaneously
  - Shares contents(files) efficiently using "file swarming"
  - Needs many concurrent sessions
  - Adopts Hybrid P2P instead of centralized P2P

Centralized Server in the past (Gnutella, Napster, Soribada)



Distributed structure (BitTorrent)



Terminology

- Essential Terms in BitTorrent Protocol (1)

block
A block is a piece of a file. When a file is distributed via BitTorrent, it is broken into smaller pieces, or blocks. Typically the block is 250kb in size, but it can vary with the size of the file being distributed. Breaking the file into pieces allows it to be distributed as efficiently as possible. Users get their files faster using less bandwidth.

client
the BitTorrent software used to download and upload files. The BitTorrent client can be downloaded here.

leech or leecher
usually refers to a peer that is downloading while uploading very little, or nothing at all. Sometimes this is unintentional and due to firewall issues. The term leech is also sometimes used to simply refer to a peer that is not seeding yet.

peer
one of a group of clients downloading the same file.

re-seed
Re-seeding is the act of putting up a new complete copy of a file after no more seeds are available to download from. This is done to allow clients with only partial downloads to complete the download process and increases availability

(Reference) http://www.bittorrent.com/intl/ko/help/faq/concepts

- Essential Terms in BitTorrent Protocol (2)

scrape
This is when a client sends a request to the tracker for information about the statistics of the torrent, like who to share the file with and how well those other users are sharing.

seed
a complete copy of the file being made available for download.

seeder/seeding
a peer that is done downloading a file and is now just making it available to others.

swarm
a group of seeds and peers sharing the same torrent.

torrent
generally, the instance of a file or group of files being distributed via BitTorrent.

torrent file
a file which describes what file or files are being distributed, where to find parts, and other info needed for the distribution of the file.

tracker
a server that keeps track of the peers and seeds in a swarm. A tracker does not have a copy of the file itself, but it helps manage the file transfer process.

(Reference) http://www.bittorrent.com/intl/ko/help/faq/concepts

Protocol Specification

- Bencoding (Binary encoding)

A way to specify the data in a terse format

Type	Description	Format	Example
Strings	Normal Strings [series of continuous characters]	**<length>:<data>**	7:network
Integers	Normal integers	**i<integer>e**	i**3**e
Lists	They are lists of types [strings, integers, lists, dictionaries].	**l<contents>e** Contents are bencoded.	l8:advanced7:networke
Dictionaries	They are a mapping of keys to values	**d<keys><values>e** Contents are bencoded with no separators.	d3:onei1e3:twoi2e5:threei3e4:four i4ee

- Structure of Torrent with a single file (MetaInfo)

  ✓ The **piece length** specifies the nominal piece size, and is usually a power of 2.
  ✓ The most common sizes are 256 kB, 512 kB, and 1 MB

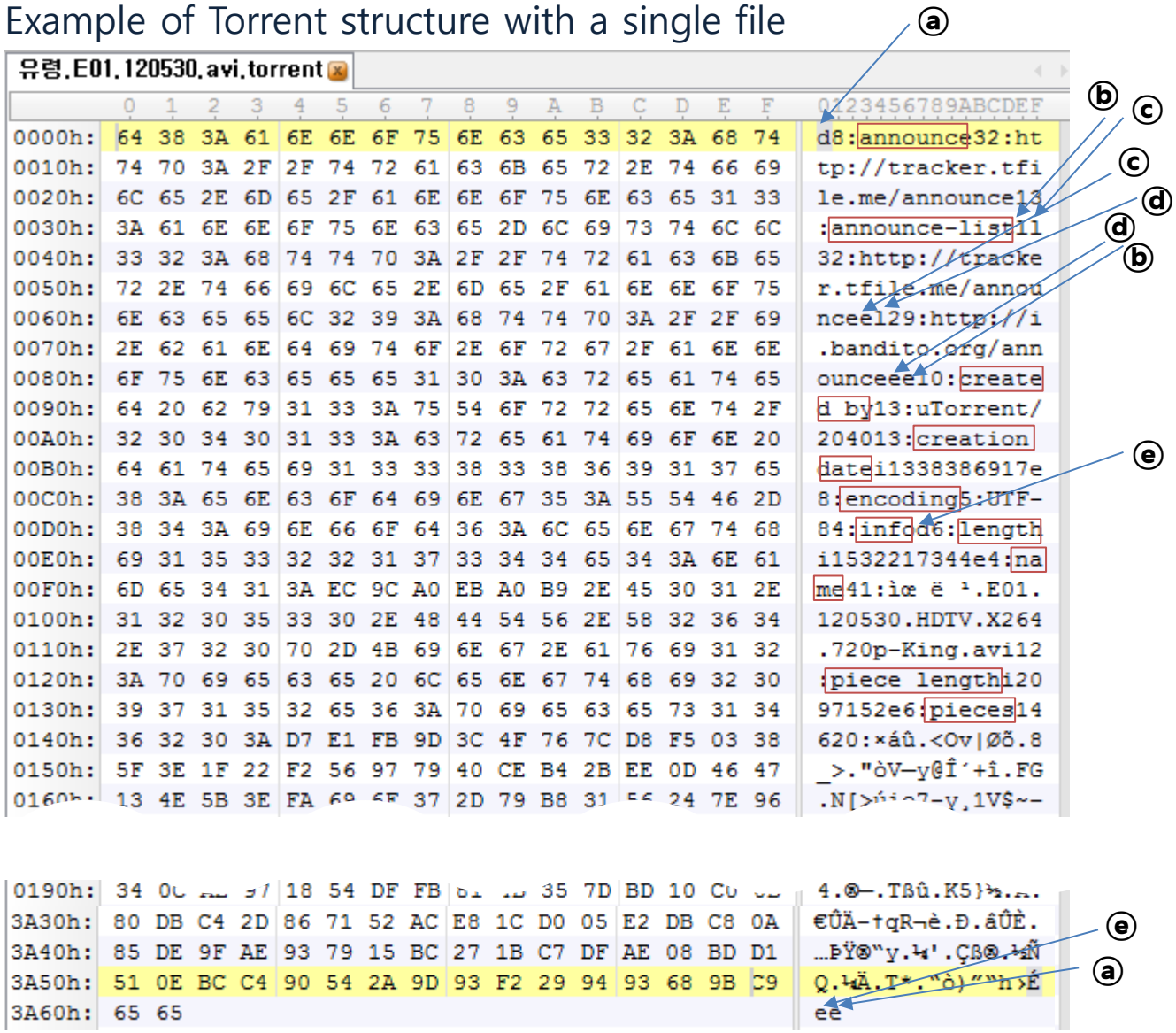
Key	Description
Info	A dictionary that describes the files
-length	Length of file in bytes (integer)
-md5sum(optional)	A 32 character hexadecimal string corresponding to the MD5 sum of the file.
-name	The filename of a string(string)
-piece length	Number of bytes in each piece (integer), **commonly 2^{18} = 256KB**
-pieces	String consisting of the concatenation of all **20-byte SHA1 hash values**, one per piece.(raw binary encoded)
Announce	**The announce URL of the tracker**
Announce-list (optional)	This is an extension to the official specification, which is also backwards comp atible. This key is used to implement lists of backup trackers.
Creation date (optional)	The creation time of the torrent, in standard Unix epoch format (integer seconds since 1-Jan-1970 00:00:00 UTC)
Comment (optional)	Free form text comments.(string)
Created by (optional)	Name and version of the program used to create.

- Structure of Torrent with multiple files (MetaInfo)

Key	Description
Info	A dictionary that describes the files
ofiles	**a list of dictionaries, one for each file.**
- length	Length of file in bytes. (integer)
-md5sum(optional)	A 32 character hexadecimal string corresponding to the MD5 sum of the file.
- path	a list containing one or more string elements that together represent the path and filename. Each element in the list corresponds to either a directory name o r the filename. (e.g) a the file "dir1/dir2/file.ext" would consist of three string el ements: "dir1", "dir2", and "file.ext". ***l4:dir14:dir28:file.exte***
oname	the name of the top-most directory in the structure -- the directory which cont ains all of the files listed in the above files list. (string)
opiece length	Number of bytes in each piece (integer)
opieces	String consisting of the concatenation of all 20-byte SHA1 hash values, one per piece. (raw binary encoded)
Announce	The announce URL of the tracker
Announce-list(optional)	This is an extension to the official specification, which is also backwards compat ible. This key is used to implement lists of backup trackers.
Creation date (optional)	The creation time of the torrent, in standard Unix epoch format (integer seconds since 1-Jan-1970 00:00:00 UTC)
Comment(optional)	Free form text comments. (string)
Created by(optional)	Name and version of the program used to create.

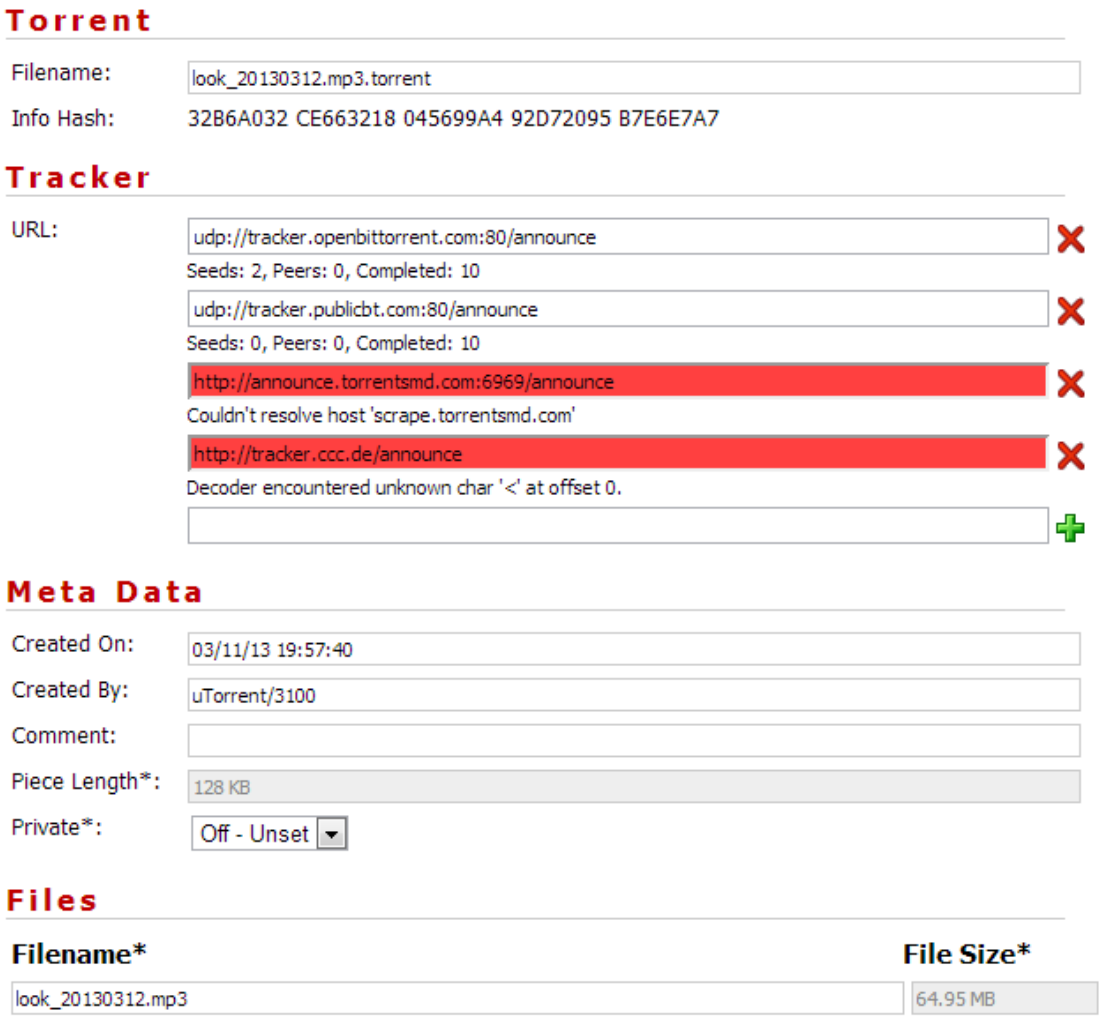
Protocol Specification Example

- Example of Torrent structure with a single file



Protocol Specification Example

- Torrent Editing Website (http://torrenteditor.com)



Protocol Specification

- Peer Wire Protocol

  ✓ It facilitates the exchange of pieces as described in the **meta-info** file.
  ✓ The response includes a peer list that helps the client participate in the torrent.
  ✓ A client must maintain state information for each connection with a remote peer.

• Choked: Whether or not the remote peer has choked this client. When a peer chokes the client, it is a notification that no requests will be answered until the client is "unchoked". The client should not attempt to send requests for blocks, and it should consider all pending (unanswered) requests to be discarded by the remote peer. Simply saying "when you are choked by the peer you can not download pieces from the peer until you are unchoked".

• Interested: Whether or not the remote peer is interested in something this client has to offer. This is a notification that the remote peer will begin requesting blocks when the client unchokes it. Typically the peer will send this message after it had received a Bit-Field message from the client telling the peer the list of pieces it has.

• am_choking=1: this client is choking the peer
• am_interested=0: this client is interested in the peer
• peer_choking=1: peer is choking this client
• peer_interested=0: peer is interested in this client

Protocol Specification

- Tracker HTTP Protocol (Request)

  ✓ HTTP Service which responds to HTTP GET requests.
  ✓ The response includes a peer list that helps the client participate in the torrent.

Parameter	Description
info_hash	20-byte SHA1 hash of the value of the info key from the Metainfo file.
peer_id	20-byte string used as a unique ID for the client, generated by the client at startup
port	The port number that the client is listening on. Ports reserved for BitTorrent are typically 6881-6889.
uploaded	The total amount uploaded so far, encoded in base ten ascii.
downloaded	The total amount downloaded so far, encoded in base ten ascii.
left	The number of bytes this client still has to download, encoded in base ten ascii.
event	If specified, must be one of started, completed, or stopped. If not specified, then this request is one performed at regular intervals.
-started	The first request to the tracker must include the event key with the started value.
-stopped	Must be sent to the tracker if the client is shutting down gracefully.
-completed	Must be sent to the tracker when the download completes. However, must not be sent if the download was already 100% complete when the client started.
ip	Optional. The true IP address of the client machine, in dotted quad format or rfc3513 defined hexed IPv6 address.
numwant	Optional. Number of peers that the client would like to receive from the tracker. This value is permitted to be zero. If omitted, typically defaults to 50 peers.

Protocol Specification

- Tracker HTTP Protocol (Response)

  ✓ HTTP Service which responds to HTTP GET requests.
  ✓ The response includes a peer list that helps the client participate in the torrent.
  ✓ Returns a random list of peers (50 by default)

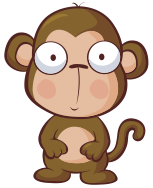
Key	Description
failure reason	If present, then no other keys may be present. The value is a human-readable error message as to why the request failed (string).
interval	Interval in seconds that the client should wait between sending regular requests to the tracker
Tracker id	String that the client should send back on its next announcements. If absent and a previous announce sent a tracker id, do not discard the old value; keep using it.
complete	number of peers with the entire file, i.e. **seeders** (integer)
incomplete	number of non-seeder peers, aka **leechers** (integer)
peers	The value is a list of dictionaries, each with the following keys
-peer id	peer's self-selected ID, as described above for the tracker request (string)
-ip	peer's IP address (either IPv6 or IPv4) or DNS name (string)
-port	peer's port number (integer)

Protocol Specification

- Tracker Request/Response

  - ✓ BitTorrent Handshaking
  - ✓ Have one?
  - ✓ Request
  - ✓ Piece (Data in Piece)
  - ✓ Port

No.	Time	Source	Destination	Dport	Protocol	Length	Info
434	2013-03-12 14	192.168.80.131	115.137.137.200	26068	BitTorrent	122	Handshake
454	2013-03-12 14	115.137.137.200	192.168.80.131	4176	BitTorrent	141	Handshake Extended [Unreassembled Packet]
455	2013-03-12 14	192.168.80.131	115.137.137.200	26068	BitTorrent	157	Extended [Unreassembled Packet]
457	2013-03-12 14	192.168.80.131	115.137.137.200	26068	BitTorrent	206	Continuation data
465	2013-03-12 14	115.137.137.200	192.168.80.131	4176	BitTorrent	538	Bitfield, Len:0x41 Have, Piece (Idx:0xb4) Have, Piece (Idx:0x1f9)
467	2013-03-12 14	192.168.80.131	115.137.137.200	26068	BitTorrent	61	Port
469	2013-03-12 14	192.168.80.131	115.137.137.200	26068	BitTorrent	66	Extended Interested



192.168.80.131

(1) Handshake: info_hash, peer_id →

← **(2) Handshake: info_hash, peer_id**

← **(3) Have: piece_index**

(4) Request: piece_index, begin_offset_of_piece, piece_length →

← **(5) Response: piece_index, begin_offset_of_piece, data_in_piece**

115.137.137.200

- Tracker Message Definition

✓ Handshaking

Client Handshake	<pstrlen><pstr><reserved><info_hash><peer_id>		
	O **pstrlen**: string length of <pstr>, as a single raw byte . O **pstr**: string identifier of the protocol. O **reserved**: eight (8) reserved bytes. O **info_hash**: 20-byte SHA1 hash of the info key in the metainfo file. O **peer_id**: 20-byte string used as a unique ID for the client.		
Tracker Handshake	<length prefix>	<message ID>	<payload>
Keep-alive	0000	0	none
Choke	**0001**	**0**	**none**
Unchoke	**0001**	**1**	**none**
Interested	**0001**	**2**	**none**
Not-interested	**0001**	**3**	**none**
Have	**0005**	**4**	**Piece index**
Bitfield	0001+X	5	Bitfield
Request	**0013**	**6**	**<index><begin><length>**
Piece	**0009+X**	**7**	**<index><begin><block>**
Cancel	0013	8	<index><begin><length>
port	**0003**	**9**	**<listen-port>**

- Tracker Message Definition

✓ General Response Code

Code	Description
100	Invalid request type: client request was not a HTTP GET.
101	Missing info_hash.
102	Missing peer_id.
103	Missing port.
150	Invalid infohash: infohash is not 20 bytes long.
151	Invalid peerid: peerid is not 20 bytes long.
152	Invalid numwant. Client requested more peers than allowed by tracker.
200	info_hash not found in the database. Sent only by trackers that do not automatically include new hashes into the database.
500	Client sent an eventless request before the specified time.
900	Generic error

- Piece Selection Algorithms

✓ **Super seeding(Initial Seeding Mode) → Special case**
A peer has nothing to trade initially
Important to get a complete piece ASAP
Select a random piece of the file and download it

✓ **Strict Priority: First Priority**
Keep the initial bitfield from each peer
Update it with every "have" message
Download the pieces that appear least frequently in these peer bitfields

✓ **Rarest First → General rule**
Determine the pieces that are most rare among your peers, and download those first.
Ensures that the most commonly available pieces are left till the end to download.

✓ **Endgame mode**
완전히 다운받는 시점이 가까워질수록 모든 peer에게 한꺼번에 요청
pending되고 있는 요청은 다운 완료 후 즉시 연결 취소
이를 통해 특정 연결 지연으로 인해 다운로드 완료가 늦어짐을 방지함
Bandwidth 낭비가 어느 정도 있으나 크지 않음

- Built-in Incentive Mechanism

  ✓ **Choking Algorithm**
  Cloaking은 업로드 일시적 거부 (temporary refusal)
  Each peer use a tit-for-tat-ish algorithm
  이유: Free Rider 방지 (게임 이론에 의거), 네트워크 혼잡도 낮춤

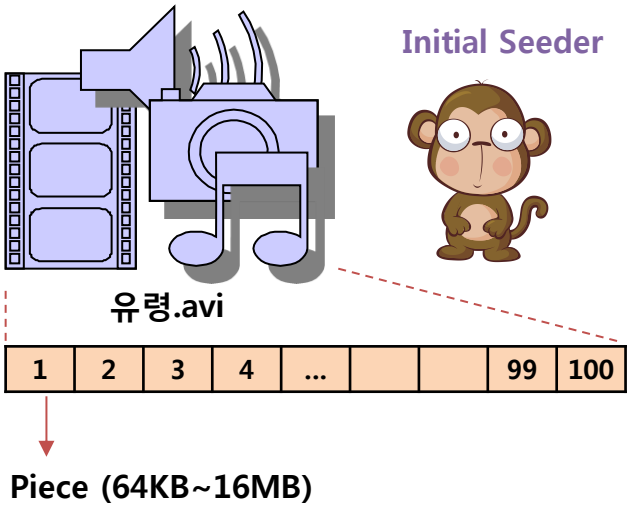
  ✓ **Optimistic Unchoking Algorithm**
  30초에 한 번씩 peer rotation
  이유: 현재 미사용 연결이 더 나은 속도를 보장할 수 있음, 새로운 peer에 service 제공

Operation

Operation

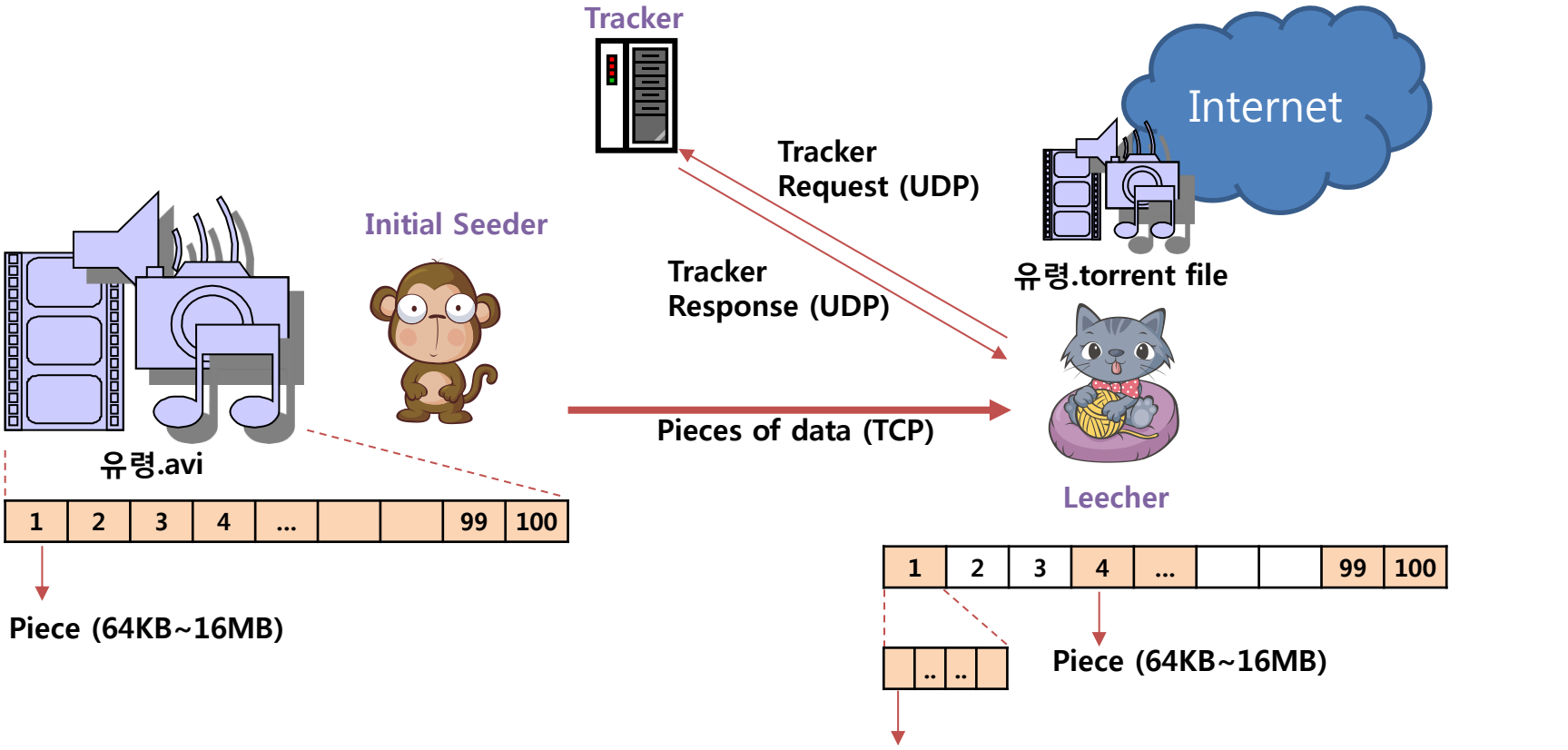
- Three aspects
  - ✓ Torrent files (Bencoded)
  - ✓ Trackers
  - ✓ Peers: Initial seeder, Seeders, Leechers

- File Sharing Mechanism
  - ✓ Initial seeder → to split file into many pieces



- File Sharing Mechanism

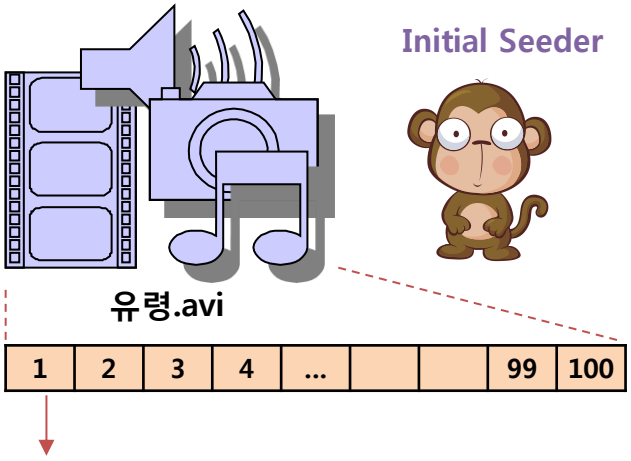
✓ Leecher would:
(1) locate the .torrent file that directs it to a tracker
(2) download and have a complete piece,

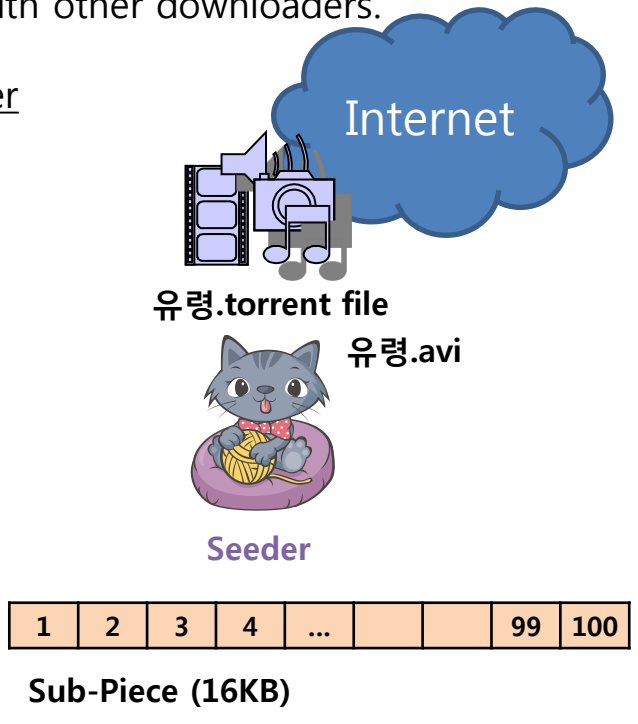


- File Sharing Mechanism

✓ Leecher would:
(1) locate the .torrent file that directs it to a tracker
(2) download and have a complete piece,
(3) start to share potentially, automatically the file with other downloaders.
(4) become another seeder

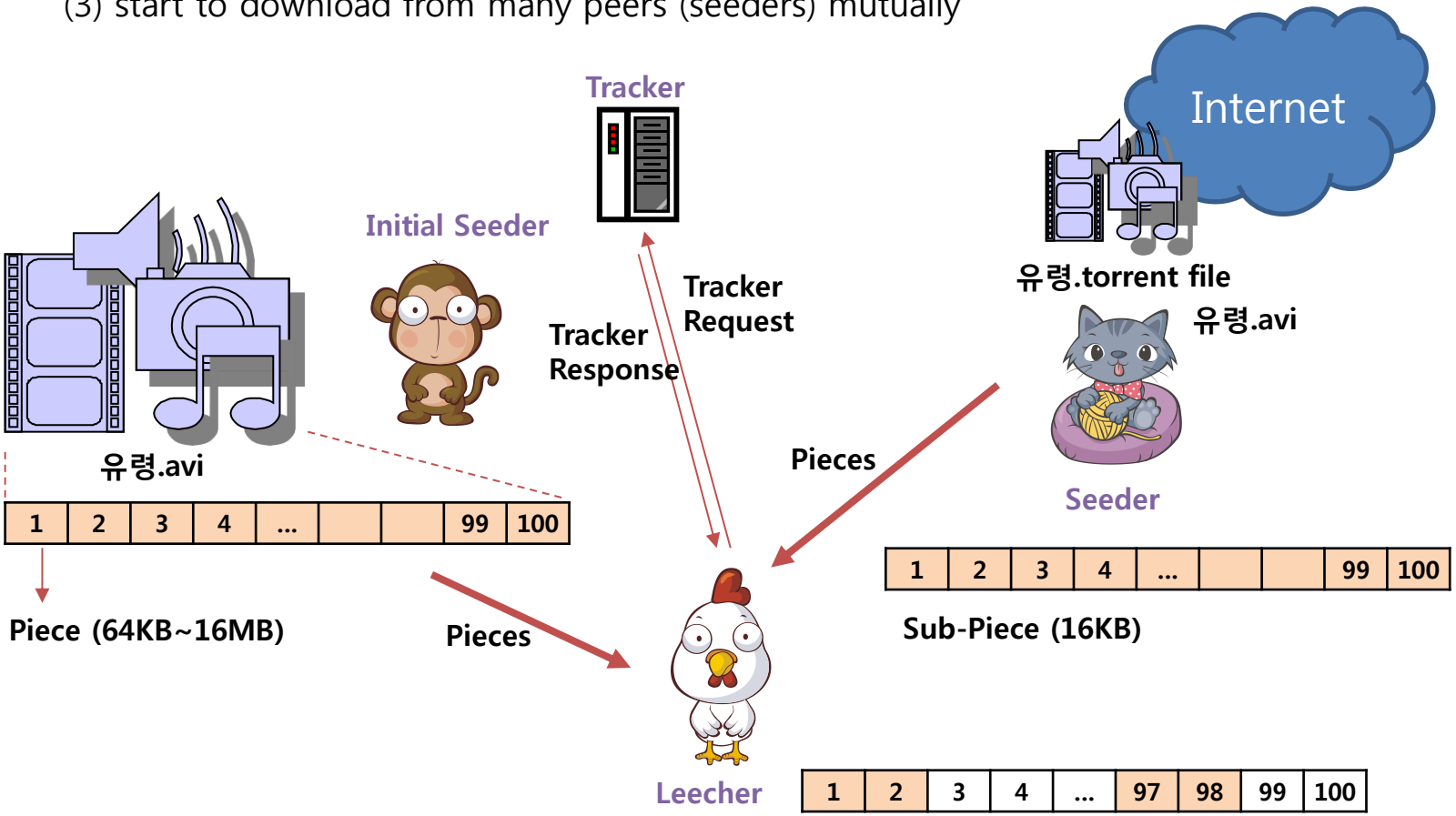
✓ The more seed, the more replicas available, the faster



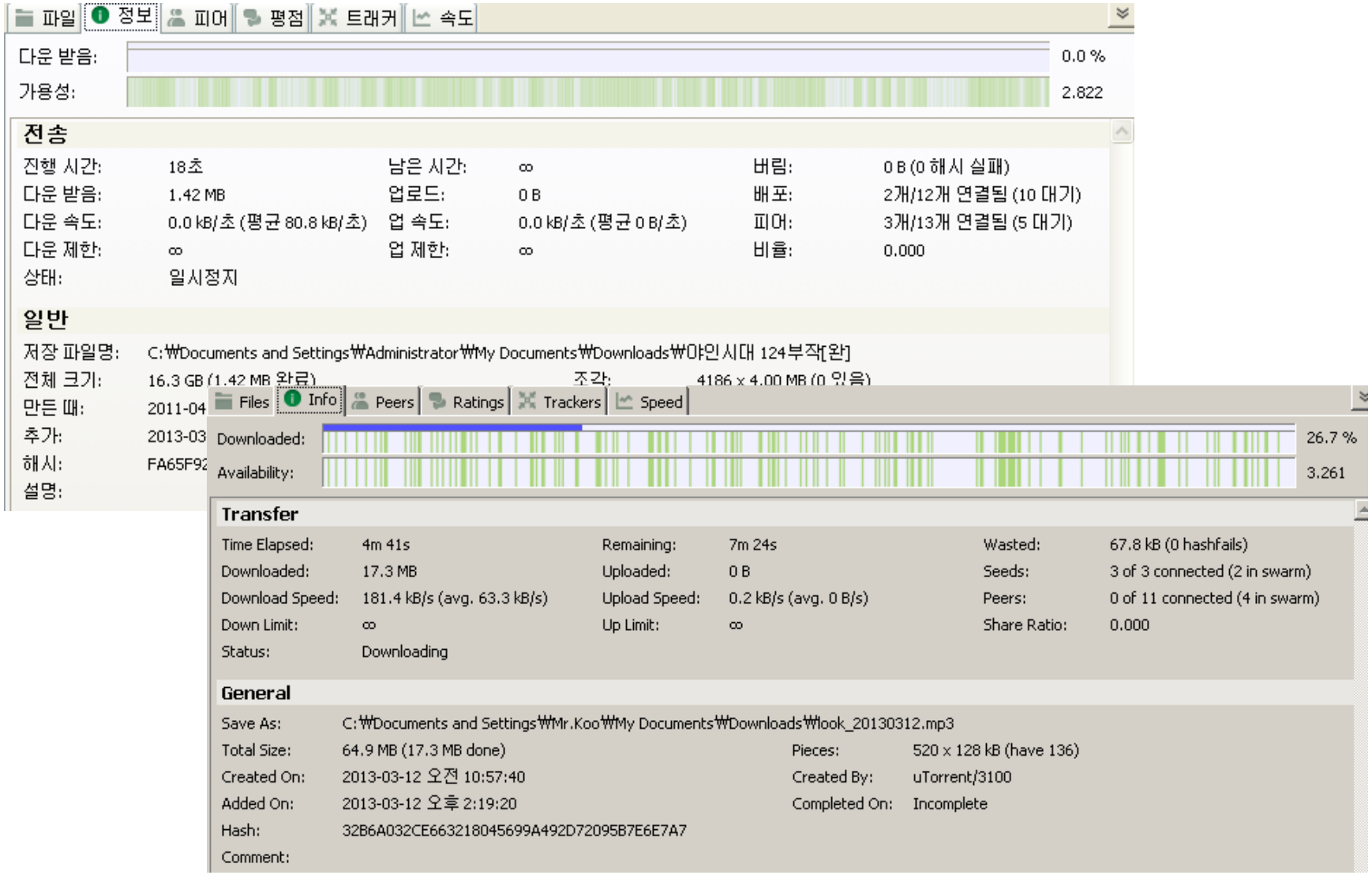


- File Sharing Mechanism

✓ Another Leecher would:
(1) request the same file to the initial seeder
(2) be returned a random list of peers (50 by default)
(3) start to download from many peers (seeders) mutually

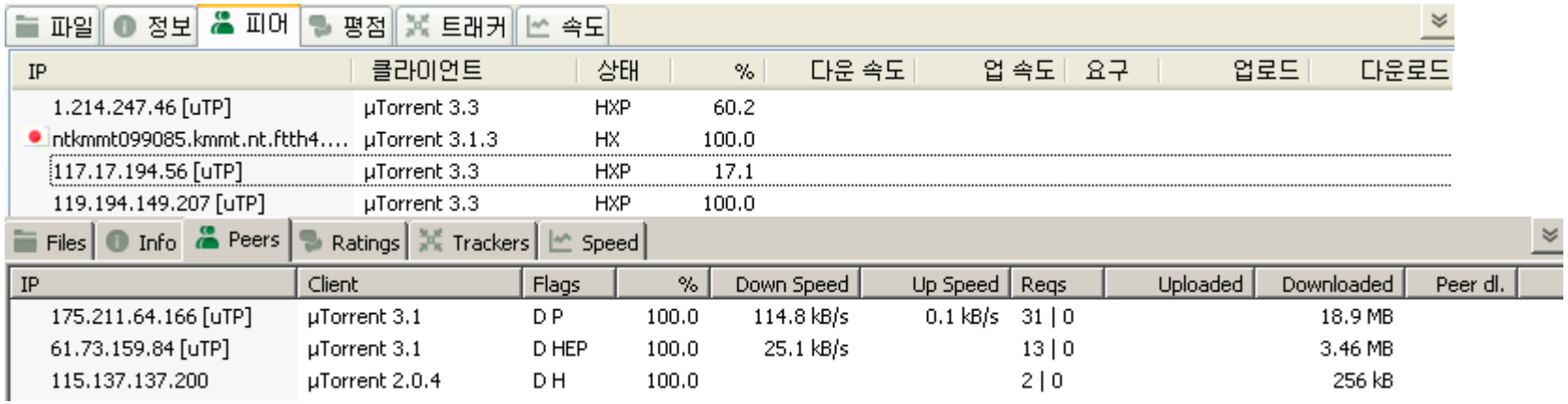


uTorrent 3.1.3 FileInfo



Example

uTorrent 3.1.3 Peers

- 

uTorrent 3.1.3 Trackers

파일 | 정보 | 피어 | 평점 | 트래커 | 속도

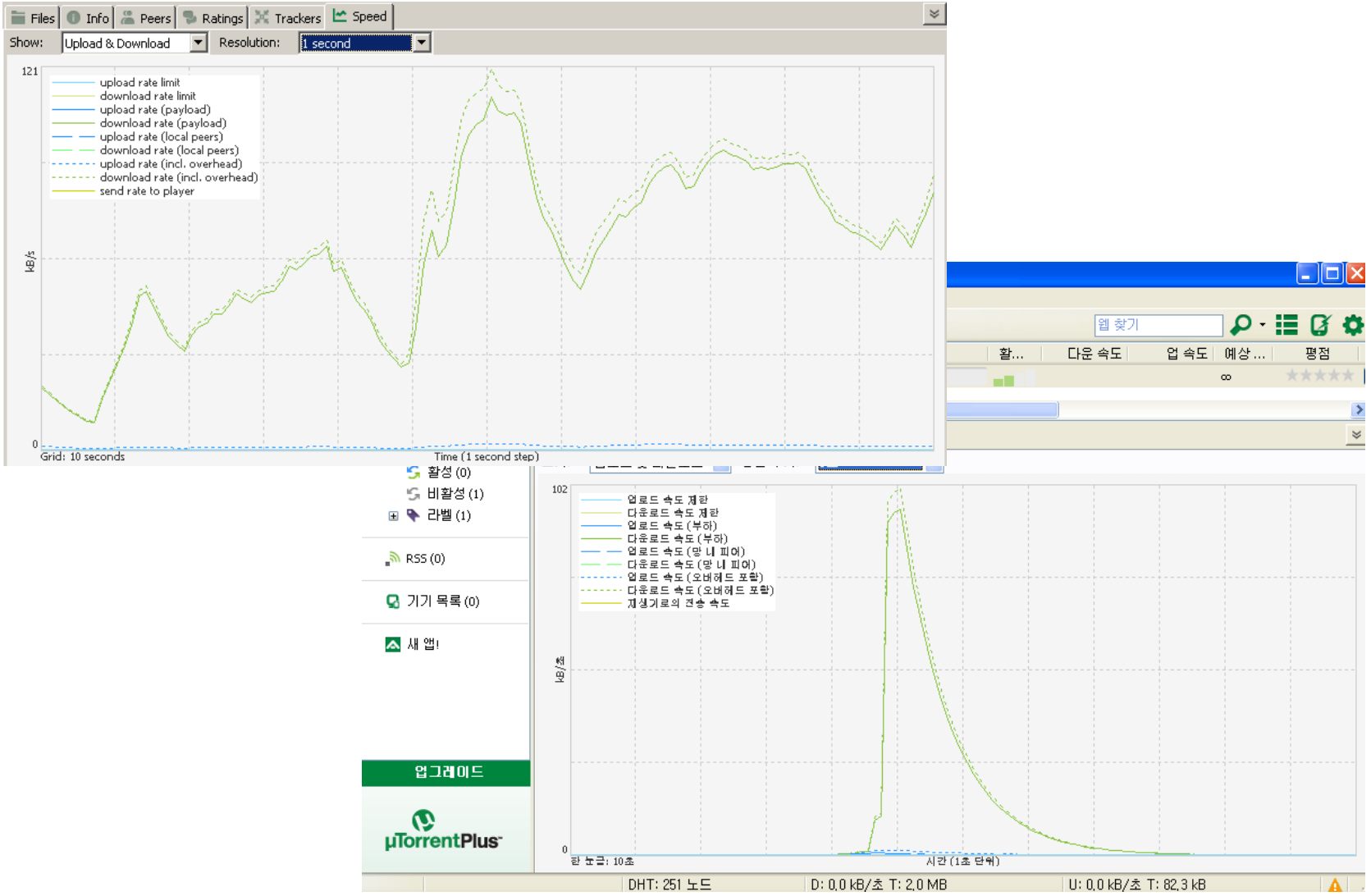
이름	상태	갱신 예정	배포	피어	다운 완료
[DHT]	작동	23분 33초	11	5	0
[망 내 피어 찾기]	동작		0	1	0
[피어 교환]	동작		15	6	0
http://10.rarbg.com:80/announce	오프라인 (시간 초과)		0	0	0
http://11.rarbg.com:80/announce	오프라인 (시간 초과)		0	0	0
http://9.rarbg.com:2710/announce	대상 컴퓨터에서 연결...		0	0	0
http://bt1.the9.com:6969/announce	오프라인 (시간 초과)		0	0	0
http://denis.stalker.h3q.com/announce	호스트명 찾을 수 없음		0	0	0
http://denis.stalker.h3q.com:6969/announce	호스트명 찾을 수 없음		0	0	0
http://denis.stalker.h3q.com:6969/announce.php	호스트명 찾을 수 없음		0	0	0
http://denis.stalker.h3q.com:80/announce	호스트명 찾을 수 없음		0	0	0
http://exodus.desync.com:6969/announce	긁어모으기 완료		10	5	1124
http://genesis.1337x.org:1337/announce			0	0	0
http://nemesis.1337x.org/announce			0	0	0
http://torrent-download.to:5869/announce	오프라인 (시간 초과)		0	0	0
http://tpb.tracker.prq.to/announce			0	0	0
http://tpb.tracker.prq.to/announce.php			0	0	0
http://tracker.bittorrent.am/announce			0	0	0
http://tracker.bittorrent.am:80/announce			0	0	0
http://tracker.ilibr.org:6969/announce			0	0	0
http://tracker.openbittorrent.com/announce			0	0	0
http://tracker.openbittorrent.com:80/announce			0	0	0
http://tracker.prq.to/announce			0	0	0
http://tracker.prq.to/announce.php			0	0	0
http://tracker.prq.to:80/announce			0	0	0
http://tracker.publicbits.com/announce			0	0	0

Files | Info | Peers | Ratings | Trackers | Speed

Name	Status	Update In	Seeds	Peers	Downloaded
[DHT]	working	15m 58s	2	8	0
[Local Peer Discovery]	working		0	1	0
[Peer Exchange]	working		0	3	0
http://announce.torrentsmd.com:6969/announce	working	56m 26s	1	1	2
http://tracker.ccc.de/announce	HTTP Error 404	22m 33s	0	0	0
udp://tracker.openbittorrent.com:80/announce	working	21m 30s	2	3	6
udp://tracker.publicbt.com:80/announce	working	25m 56s	1	4	5

Example

uTorrent 3.1.3 Speed



References

관련 기술

http://www.bittorrent.org/beps/bep_0003.html (Official BitTorrent Specification)
http://www.bittorrent.com/intl/ko/help/faq/concepts
http://en.wikipedia.org/wiki/BitTorrent (Wikipedia)
https://wiki.theory.org/BitTorrentSpecification (Wikipedia)
https://wiki.theory.org/BitTorrent_Tracker_Protocol (Wikipedia)
http://www.netmanias.com/bbs/view.php?id=techdocs&no=62 (BitTorrent Protocol의 동작원리)
https://github.com/rakshasa/libtorrent/blob/master/doc/multitracker-spec.txt (Multitracker-Spec)

참고 사이트

http://torrenteditor.com/ (Torrent file Editor 제공)
http://www.etorrent.co.kr (Torrent file Download 제공)

관련 논문

http://www.comp.brad.ac.uk/het-net/HET-NETs05/ReadCamera05/P30.pdf
https://web.cs.umass.edu/publication/docs/2012/UM-CS-2012-016.pdf
BitTorrent를 이용한 저작물 불법 공유 조사 방법에 관한 연구/박수영, 정현지, 이상진

Question and Answer