Goals

Free Riding in BitTorrent is Cheap

Thomas Locher, **Patrick Moor**, Stefan Schmid and Roger Wattenhofer

Distributed Computing Group
ETH Zürich

HotNets-V, 2006

- Analyze free riding (not uploading any user data) in BitTorrent with a practical setup
- Try different tricks and evaluate their effectiveness
- Examine swarms with different characteristics

About BitTorrent

Popular P2P system with centralized coordinator (**tracker**), whose address can be found in the **torrent metafile**.

Nodes divided into **seeders** owning a complete download of the file and **leechers**, which are still in progress of downloading the file.

Upload decisions are made using a **choking** algorithm based on the download rates received from remote peers. At regular intervals an **optimistic unchoke** takes place which allows new peers to participate more quickly.

Choking algorithm is *similar* to **tit for tat**.

BitThief

We've written a BitTorrent client from scratch in Java, which we used to implement and test the different tricks. A lot of instrumentational code was added to measure many of the aspects of a torent download.
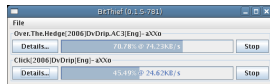
Many Connections

- Mainline (official) client imposes an incoming connection limit of 80. For an honest peer this is usually enough as it will use only a few connections to make up for most of its downstream.

Many Connections

- Mainline (official) client imposes an incoming connection limit of 80. For an honest peer this is usually enough as it will use only a few connections to make up for most of its downstream.
- Our client opens as many connections as possible. The data amounts transferred on these connections are usually more balanced compared to the honest peer's connections.

Many Connections - Implications

- We take advantage of being connected to more seeders, which yields higher download rates directly.
- We are connected to more leechers and thus profit more often from their optimistic unchoke slots.

Many Connections - Implications

- We take advantage of being connected to more seeders, which yields higher download rates directly.
- We are connected to more leechers and thus profit more often from their optimistic unchoke slots.
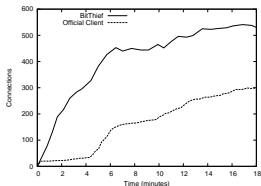
Uploading Garbage

- Trick remote peers into thinking that we upload lots of valid data in order to be unchoked more often
- Only large chunks of data (eg. 256KB) verifiable by peers, while exchange units are 16KB

Uploading Garbage

- Trick remote peers into thinking that we upload lots of valid data in order to be unchoked more often
- Only large chunks of data (eg. 256KB) verifiable by peers, while exchange units are 16KB
- Current implementations (Azureus) too clever
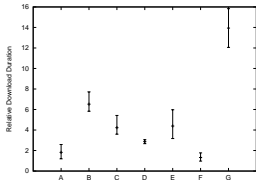- Download rates actually got worse (IP banned)

Results

Relative download rates ranging from 0.2 up to 6 times the ones achieved by the mainline client.
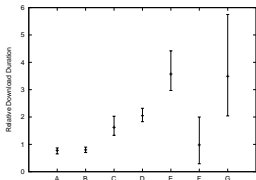
Results

When considering leechers only, the download rates obtained by BitThief range from 0.9 to 16 times the rates achieved by the mainline client (*not* ignoring seeders!).

Sharing Communities

- Torrent metafiles can be hard to find, spam et al.

Sharing Communities

- Torrent metafiles can be hard to find, spam et al.
- *Closed* communities have evolved with different rules for joining them

Sharing Communities

- Torrent metafiles can be hard to find, spam et al.
- *Closed* communities have evolved with different rules for joining them
- Communities punish peers not able to keep their *sharing ratio* > 1.0

Sharing Communities

- Torrent metafiles can be hard to find, spam et al.
- *Closed* communities have evolved with different rules for joining them
- Communities punish peers not able to keep their *sharing ratio* > 1.0
- Sharing ratio measurement methods are weak: Make use of
  information sent by the client to the tracker which can easily be faked

Tracker Protocol

All information available to the tracker comes from the periodic announce messages peers send to it:

Tracker HTTP Request

GET /announce?...&uploaded=**86016**&downloaded=22528&left=81920&...

Experiment

We downloaded a 12h old file (350MB), three times from within a community and three times from an open tracker. The community swarm consisted of 25 peers and the public one of 830, according to the trackers.
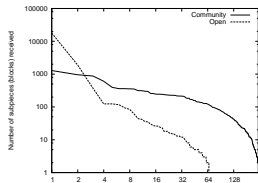
Experiment

We downloaded a 12h old file (350MB), three times from within a community and three times from an open tracker. The community swarm consisted of 25 peers and the public one of 830, according to the trackers.
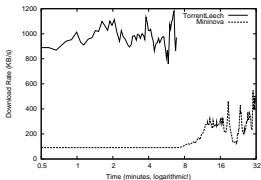
Another Experiment

Another 350MB file, much more popular this time (Heroes Episode 5).

Possible Explanations

- More seeders because of docile peers trying to push their sharing ratio in an honest way.
- More generous leechers donating a lot of upstream to push their sharing ratio.
- Community members more tech-savvy? (less firewalled peers, faster network connections)

Conclusion

- Possible to download entire files without ever contributing a single byte.
- Download speed not much worse compared to honest peers, if worse at all.
- Communities, while actively encouraging collaboration, actually render cheating a lot easier.

BitThief is available at http://dcg.ethz.ch/projects/bitthief/.

Where to go?

- How to prevent these exploits?
- Implications of large scale BitThief deployment?

Questions and Comments?

Thank you for your attention!

patrick@moor.ws

http://www.dcg.ethz.ch/