Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks

Team 4

Shivani Chepuri
Gowri Lekshmy

Objectives

- Understanding P2P Networks and BitTorrents
- Develop a fluid model to study the performance of BitTorrent
- Propose strategies to prevent Free Riding
- Study Incentive Mechanism in BitTorrents
- Analyse experimental results
- Applications

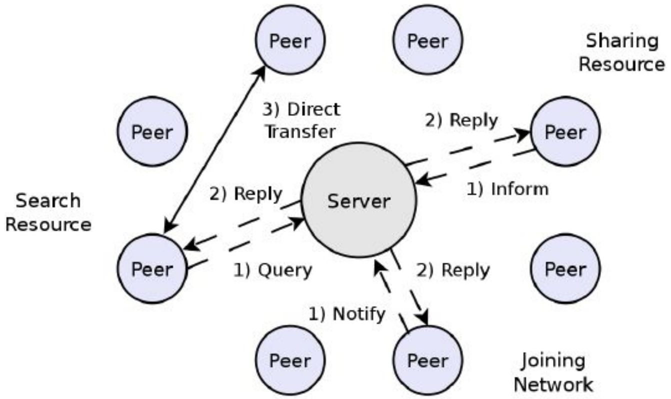
Approach/Workflow

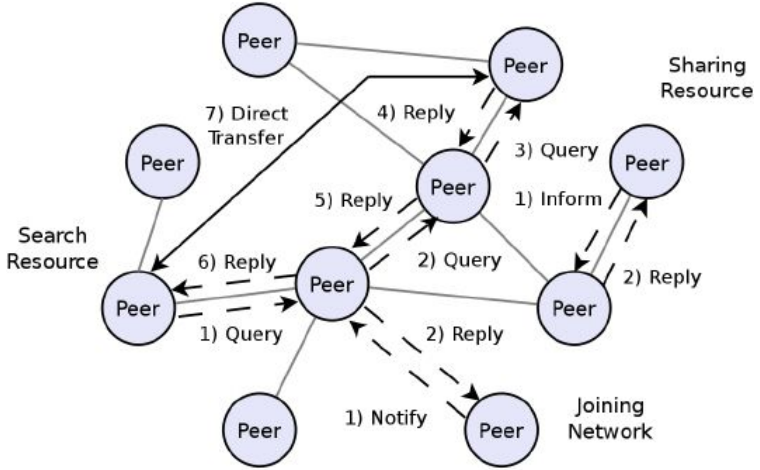
1. P2P File Sharing and Architecture
2. BitTorrent Protocol, Peer and Piece Selection
3. Focus on Optimistic Unchoking and Free Riding - Related Experimental analysis
4. Fluid Model and respectives Inferences
5. Real-Time Video Application - BitTorrent is a common protocols for transferring large files.

Overview of P2P networks

- P2P Networks account for approximately 43% to 70% of all Internet traffic.
- A distributed application that partitions tasks or workloads between peers.
- Peers can request for the files from other peers by establishing TCP or UDP connections.
- Eg. BitTorrent, Gnutella, Freenet, BearShare, Soulseek, etc
- **Peers:** Equally privileged, equipotent participants in the application

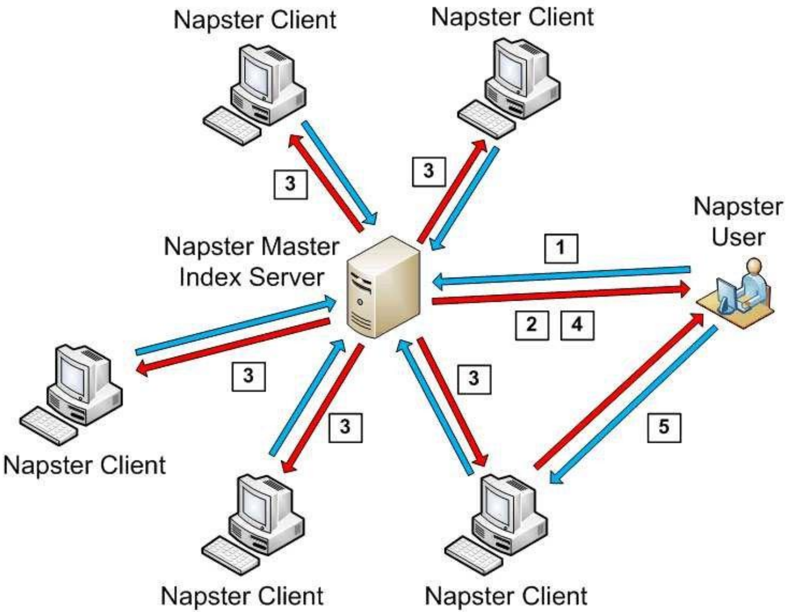
P2P Architecture





Napster's P2P file-sharing

Oldest P2P file-sharing protocol, used for downloading .mp3 files. It is based on a **Centralized/Hybrid architecture.**

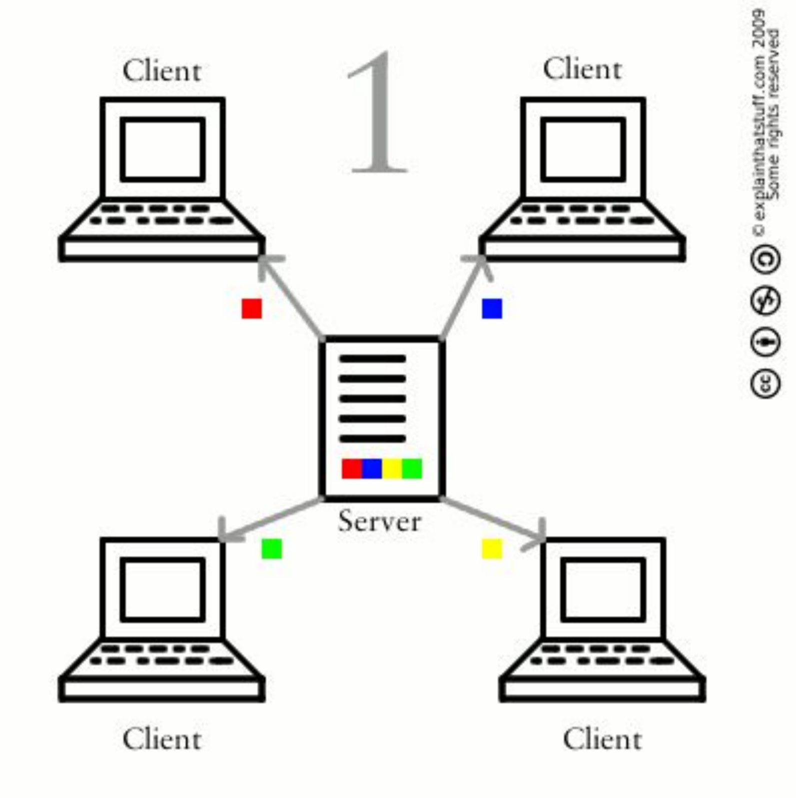


Decentralised P2P Networks

- Depending on how the nodes in the overlay network are linked, a decentralized P2P network is classified further
- **Unstructured** - Nodes are linked randomly.
  - Eg. Gnutella, Freenet, etc
- **Structured** - Predefined set of rules to link nodes, Uses Distributed Hash Table.
  - Eg. BitTorrent

BitTorrents

- A P2P protocol for sharing large files within a swarm.
- BitTorrent was responsible for 3.35% of all worldwide bandwidth, more than half of the 6% of total bandwidth dedicated to file sharing.
- **Swarm:** The set of all peers that takes part in a torrent (Swarm = Seeds + Leeches)
- **Torrent:** In BitTorrent, a file is referred to as a torrent.
- BitTorrent Protocol applies a set of policies:
  - To provide Fairness
  - Encourage Peers to Upload
  - Prevent Overloading a single peer
  - To allow a peer to find peers that provide better service.


Client
1
Client
Server
Client
Client
© explainthatstuff.com 2009
Some rights reserved

BitTorrents

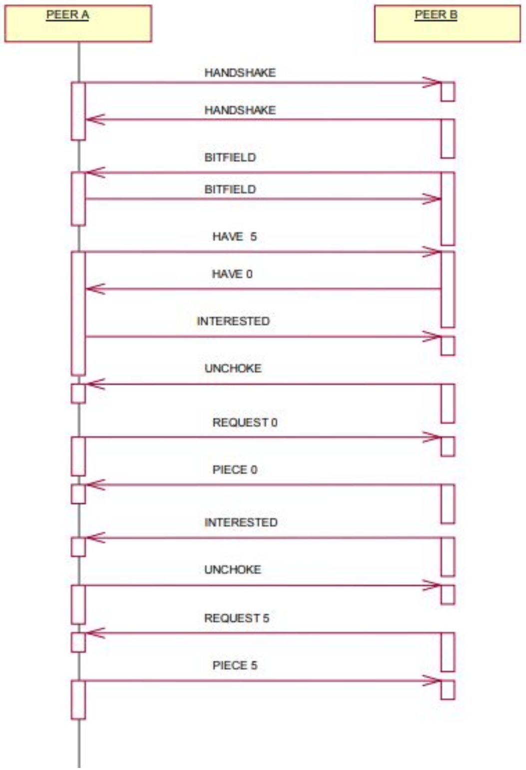
- **Tracker:**
  - Tracker: A central node that tracks the operation of the swarm
  - Provide a list of files and list of seeds available for transfer
  - No knowledge of the contents of the files being distributed

- Tracker functions - Centralised and Decentralised.
- Decentralised - Tracker's functions are distributed among the nodes (trackerless torrents or distributed torrents)

BitTorrents - Protocol Dependencies

- The well known TCP port for BitTorrent traffic is 6881-6889 and the port is 6969.
- A recent extension to BitTorrent is the DHT ("distributed sloppy hash table" or simply called UDP tracker) protocol.A UDP based peer to peer tracker protocol.
- The uTorrent imports another UDP based Micro Transport Protocol, called uTP.
- The DHT extension (peer2peer tracker) uses various UDP ports negotiated by the peers. DHTs are used in Bittorrent for peers to send a list of other seeds/peers in the swarm for a particular torrent directly to a client without the need for a tracker.

BitTorrent Message Flow (A Fair One)



Performance Metrics

- Peer Evolution
  - Number of peers in the network
  - Study the evolution of peers as a function
- Scalability
  - Network performance increases with size
  - Characterized by no. of peers, arrival rate of peers etc.
- File-Sharing Efficiency
  - Peers have different uploading/downloading bandwidths.
  - To efficiently download the file, design the file-sharing protocol such that each peer is matched with others who have the other pieces of the file
  - Ensure that the downloading bandwidth of each peer is fully utilized.

Peer Selection Mechanism

- To provide fairness
- Improve the downloading performance of peers
- Punish free-riders
- Tit-for-Tat - when a peer's upload bandwidth is saturated, it will use a tit-for-tat strategy. Cooperation is achieved when upload bandwidth is exchanged for download bandwidth. **Least Co-operative Peer is Choked**(Regular UnChoking)
- Upload Only - Reward the seeds
- Anti-Snubbing - When a peer received no data in 60s, we assume it is choked by all other peers, and refuse to upload to it except for the optimistic unchoking.
- Optimistic Unchoking

Optimistic Unchoking

- With TFT alone, No opportunity for discovering other peers that can provide higher uploading rates.
- A peer randomly promotes a single interested peer, regardless of its uploading rate, from the choked group and flags it as unchoked.
- Ineffective rewarding and punishing of downloaders can increase Free-Riding.

Peer Selection Algorithm

- Robust against free riding
- Peer Selection Algorithm
  - Find best k neighbours
  - Unchoke those k neighbors and choke all other neighbors
  - Choking algorithm (every 30 seconds)
    - Filter out uninterested neighbors
    - Sort neighbors in decreasing order of download rates
    - Pick first k neighbors
- Optimistic unchoking: 30 seconds of unconditional uploading

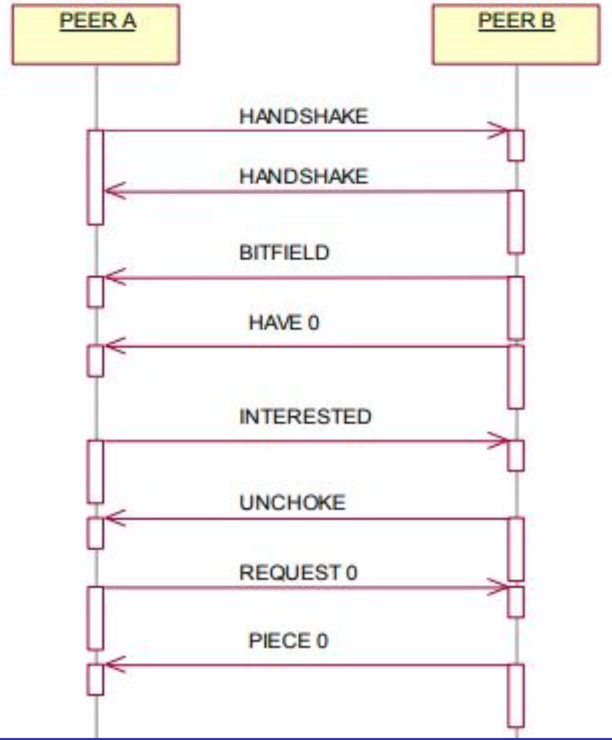
Piece Selection Strategies

- Strict Priority
  - Before requesting for another piece, a whole piece of a file should be downloaded
- Endgame Mode
  - Algorithm for downloading the last few pieces of a torrent.
  - Users can’t wait long for last few pieces
  - Requests **all peers** for last pieces
  - After receiving them, cancels request to other peers.
- Rarest first Strategy
  - First downloads pieces with the fewest copies among the neighbors to circulate faster
- Random First Piece
  - An exception to rarest first policy when a peer first joins a torrent.
  - Tries to download whole piece, to follow TFT quickly
  - Download the first piece randomly to get a whole piece quickly

Free Riding

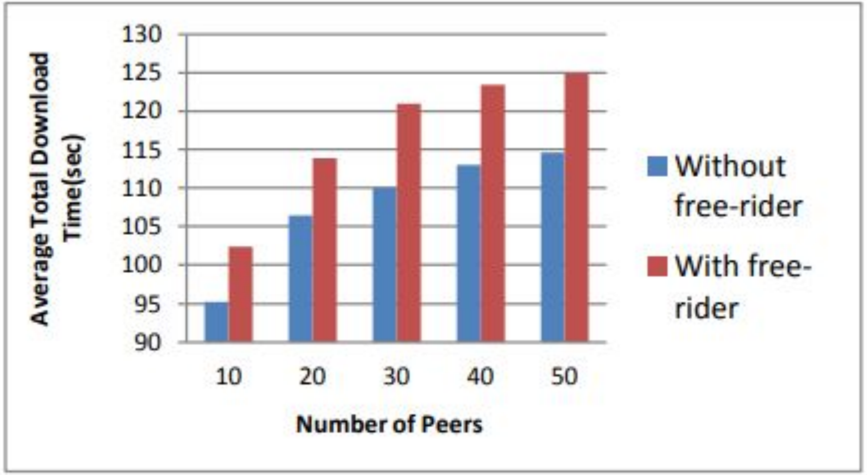
- This Peer does not contribute anything to the system, while it attempts to obtain serPice (or downloading) from other peers, to minimize bandwidth utilization at its own end.
- **OU:** If No.of uploading peers = 4, a free-rider gets 20% of the possible maximum downloading rate.
- **How to solve this?**
- Global Information on Band Width - Not Practical

Message Flow in Free Riding between a free-rider (A) and non-free-rider (B)



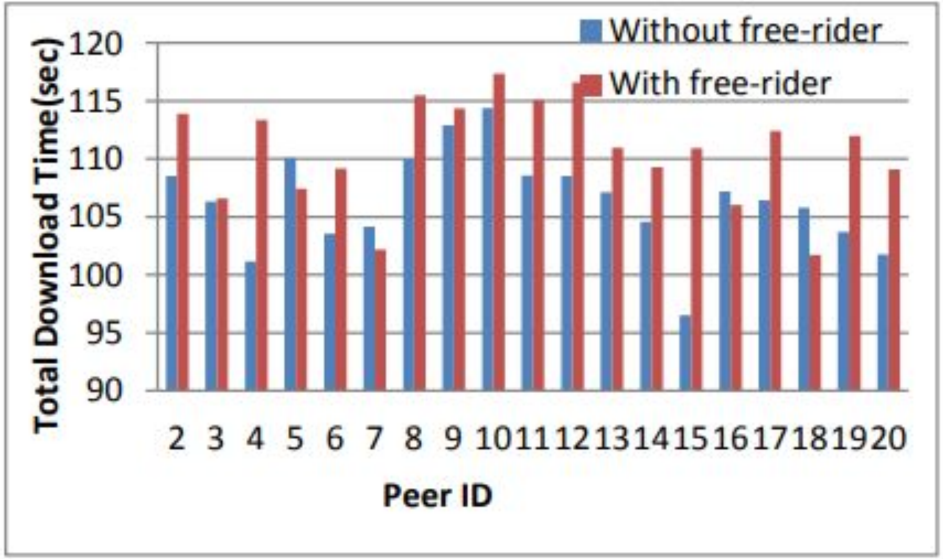
Preventing Free Riding
Average total download time (TDT) of peers

This shows free-riders in a P2P network increase the average total download time for all peers.



Average total download time (TDT) of peers

Free-riders are not effectively punished by standard mechanisms in the P2P network. Also, the presence of free-riders degrades the download performance of contributing peers i.e. non-free-riders.



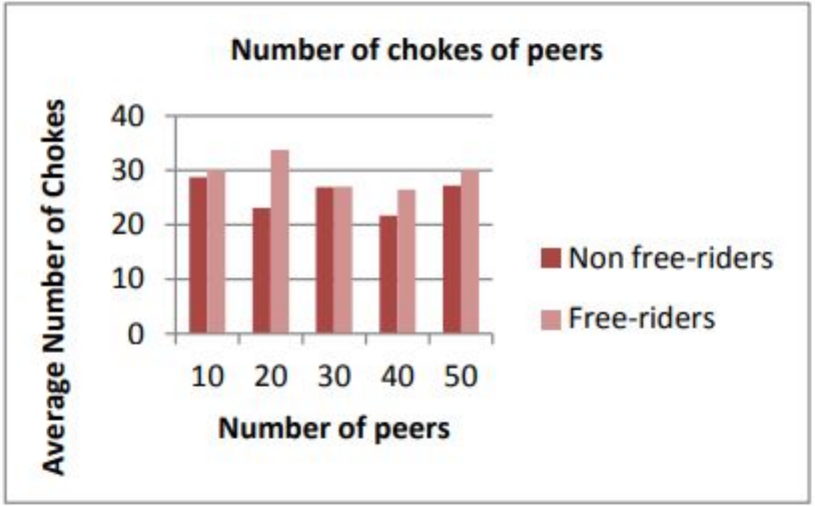
Detect and Punish Free-Riders

- A peer is declared as a free-rider under the following conditions:
  - It must download the minimum threshold, of the file size (here 20 MB)
  - It must not upload any file piece.
- How to Punish the free riders?
  - By decreasing the download times of non-free-rider peers of the network and punishing free-riders by increasing their download times.

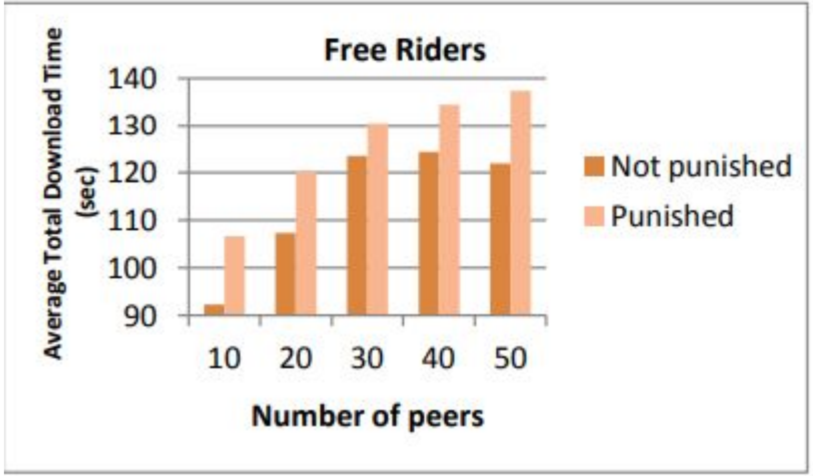
Principle: “The more a peer gets choked, the more will be its download time”.

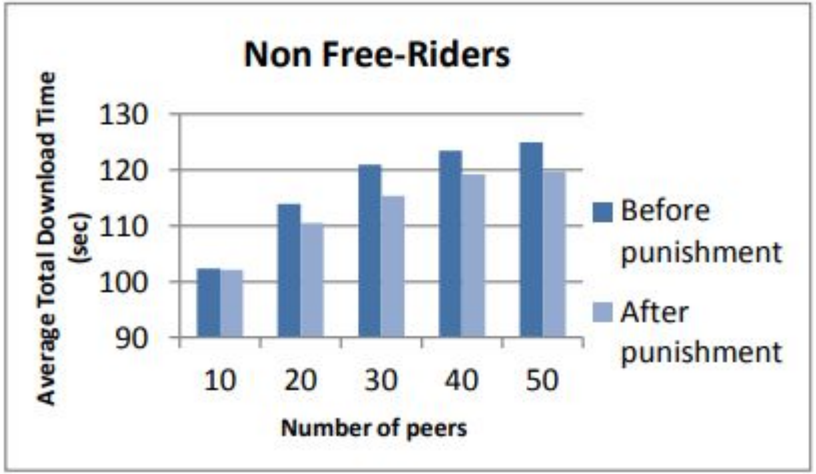
Results

- There is a 3 % decrease in the total download time of non-free-riders.
- An increase of an average of 10% in total download time is observed on the free-riders.
- Free-riders were continuously choked every time the decision of unchoking is made by an optimistic unchoking mechanism.



Results


Free Riders
Average Total Download Time (sec)
140
130
120
110
100
90
10
20
30
40
50
Number of peers
Not punished
Punished


Non Free-Riders
Average Total Download Time (sec)
130
120
110
100
90
10
20
30
40
50
Number of peers
Before punishment
After punishment

Simple Fluid Model

- A mathematical model used to describe the fluid level in a reservoir subject to randomly filling and emptying.
- They are used to predict queue lengths and waiting times.
- **Purpose** : Obtains the average file-transfer time and characterizes the variability of the number of peers around the equilibrium values predicted by the deterministic fluid model.
- Models evolution of no. of peers

$$
\begin{aligned}
\frac{\mathrm{d}x}{\mathrm{d}t} &= \lambda - \theta x(t) - \min\{cx(t), \mu(\eta x(t) + y(t))\}, \\
\frac{\mathrm{d}y}{\mathrm{d}t} &= \min\{cx(t), \mu(\eta x(t) + y(t))\} - \gamma y(t),
\end{aligned}
$$

Model Parameters

- $x(t)$ number of downloaders (leechers) in the system at time t
- $y(t)$ number of seeds in the system at time t
- λ the arrival rate of new requests
  - Assumption: Poisson Process
- μ the uploading bandwidth of a given peer
  - Assumption: All peers have the same uploading bandwidth
- c the downloading bandwidth of a given peer
  - Assumption: All peers have the same downloading bandwidth, $c \geq \mu$
- θ the rate at which downloaders abort the download.
    - Assumption: Each downloader independently aborts its download after a certain time, which is exponentially distributed with mean $1/\theta$.
- γ the rate at which seeds leave the system
- η indicates the effectiveness of the file sharing
  - η takes values in $[0, 1]$

The system in Steady-State:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = \frac{\mathrm{d}y(t)}{\mathrm{d}t} = 0$$

Little's Law:

- L = λW
- L – the average number of items in a queuing system
- λ – the average number of items arriving at the system per unit of time
- W – the average waiting time an item spends in a queuing system

Using Little's Law, we have

$$\frac{\lambda - \theta\bar{x}}{\lambda}\bar{x} = (\lambda - \theta\bar{x})T,$$

$$T = \frac{1}{\theta + \beta}.$$

Where, $\frac{1}{\beta} = \max\{\frac{1}{c}, \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\gamma})\}$

Inferences from The Fluid Model

- T is independent of λ (the request arrival rate). Hence, the BitTorrent P2P system scales very well.
- When η increases, T decreases.
- When γ increases, T increases

Inferences from The Fluid Model

- from deterministic fluid model equations

$$\frac{\mathrm{d}y(t)}{\mathrm{d}t} \leq (\mu - \gamma)y(t).$$

- This tells us that y(t) decreases at least exponentially.
- The number of seeds will exponentially decrease to zero and the system dies.
- So, it is very important for the downloaders to upload data to each other.

Effectiveness of File Sharing

- A Study on η, the parameter that captures the effectiveness of file sharing:
- Obtained Expression:

$$\eta = 1 - \mathbb{P}\left\{ \begin{array}{c} \text{downloader } j \text{ needs no} \\ \text{piece from downloader } i \end{array} \right\}^k, \qquad \eta \approx 1 - \left(\frac{\log N}{N} \right)^k.$$

- For a given downloader i, we assume that it is connected to $k = \min\{x - 1, K\}$ other downloaders
- Here, x is the number of downloaders in the system and K is the maximum number of downloaders to which a peer can connect.

Analysis of Results

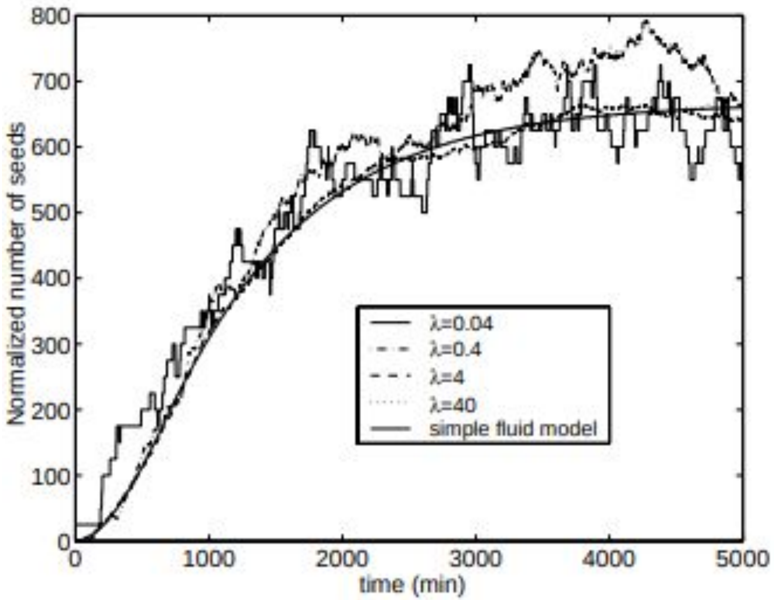
- N is of the order of several hundreds (In BitTorrent, each piece is typically 256KB). Hence, even if $k = 1$, η is very close to one. Typically $K = 40$ in BitTorrents.
- This tells us that **BitTorrent is very efficient** in sharing files.
- When k increases, η also increases but very slowly and the network performance increases slowly.
- Note that, since k depends on the number of other peers in the system, it may be related to the arrival rate λ

Experiments

- Comparison of the proposed simple deterministic fluid model with the results from a discrete-event simulation of a BitTorrent-like network.
- For a file which had a total of less than 100 completed downloads.
- γ is the rate at which seeds leave the system.
- Experiment 1 - with $\gamma = 0.001$
- Experiment 2 - with $\gamma = 0.005$
- The uploading bandwidth becomes the bottleneck.
- Simple fluid model is accurate when λ is large, but performs well even for smaller λ.
- λ the arrival rate of new requests

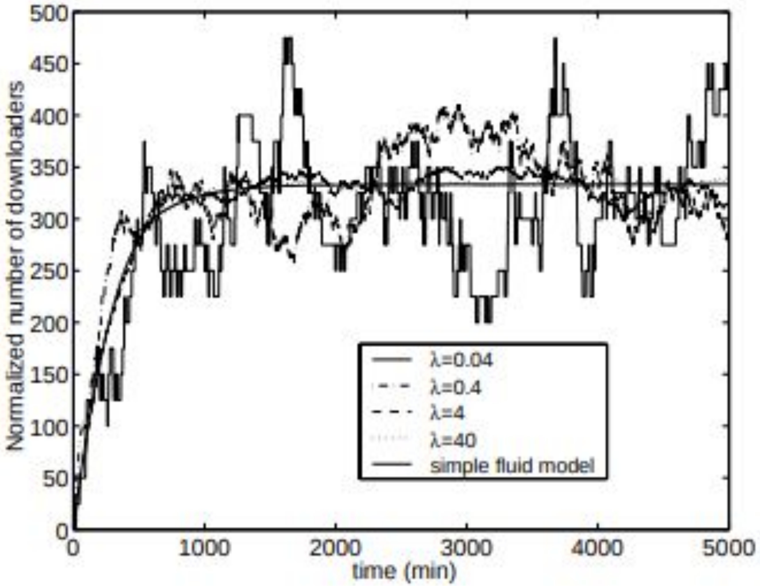
Experiment 1

The evolution of the number of seeds as a function of time



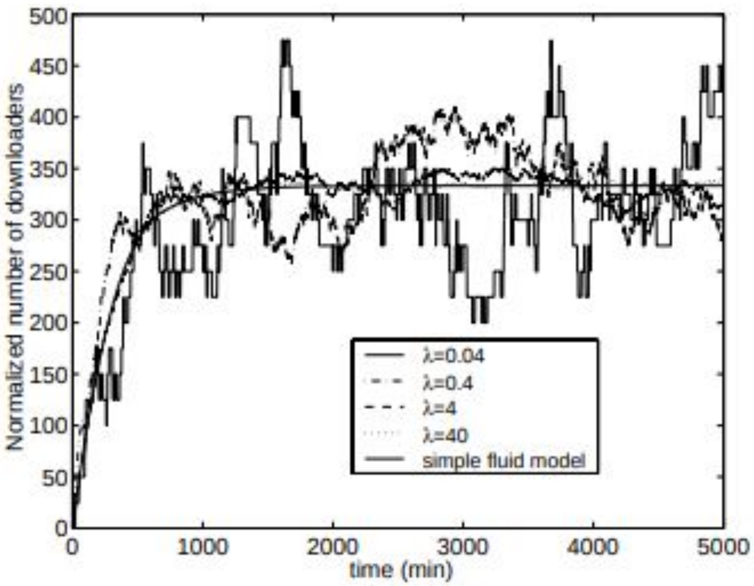
Experiment 1

The evolution of the number of downloaders as a function of time



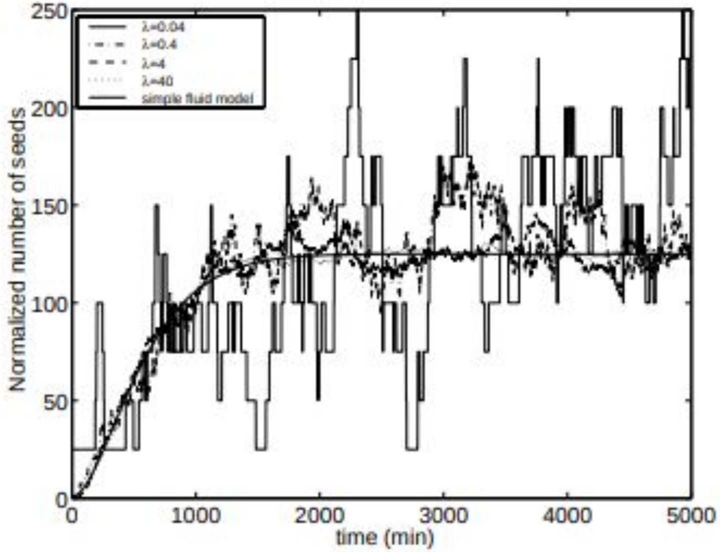
Experiment 2

- The uploading bandwidth becomes the bottleneck. Simple fluid model is accurate when λ is large, but performs well even for smaller λ.
- The evolution of the number of downloaders:



Experiment 2

The evolution of the number of seeds:



Experiment 2

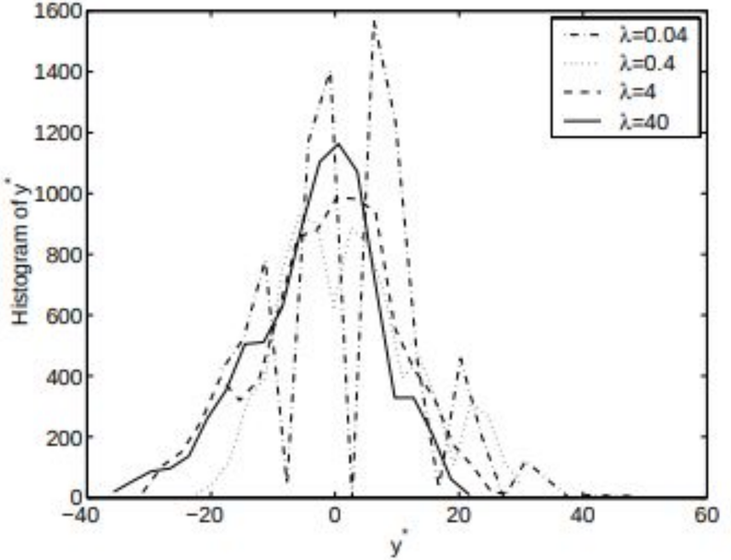
- Histogram of the variation of the number of seeds around the fluid model:
- Actual simulation values/parameters:
  - x_sim(t) is the #downloaders
  - y_sim(t) is the #seeds
- Deterministic fluid model values/parameters:
  - x(t) is the #downloaders
  - y(t) is the #seeds

$$\hat{x}(t) = \frac{x_{sim}(t) - x(t)}{\sqrt{\lambda}}$$

$$\hat{y}(t) = \frac{y_{sim}(t) - y(t)}{\sqrt{\lambda}},$$

Experiment 2

- Histograms look roughly Gaussian (figures for sufficiently large λ).
- The variances of ˆx and ˆy do not change much when λ changes from 0.04 to 40.



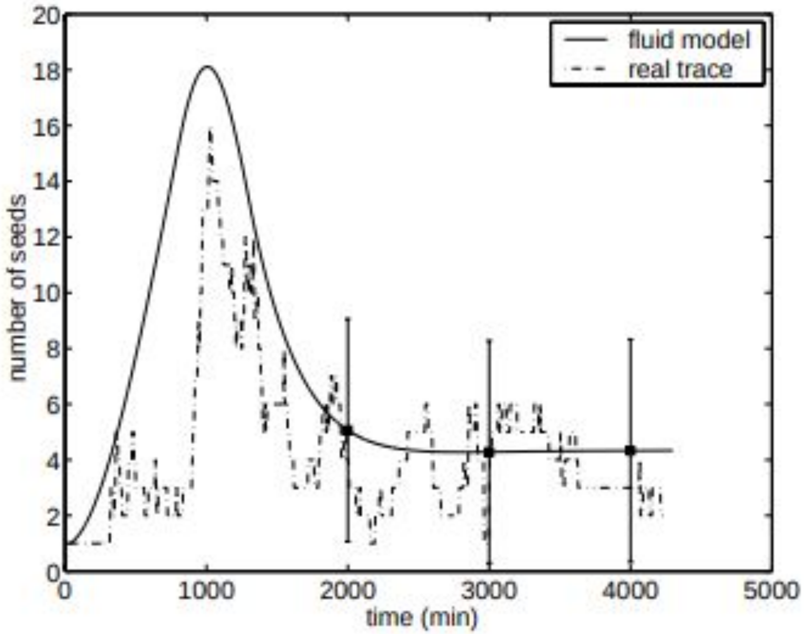
Experiment 3

Setup:

- A file is introduced into the BitTorrent network and the log files of the BitTorrent tracker are collected for a time period of around three days.
- Not possible to determine whether the uploading bandwidth or the downloading bandwidth is the bottleneck from the tracker log files (assume the uploading bandwidth is the bottleneck).

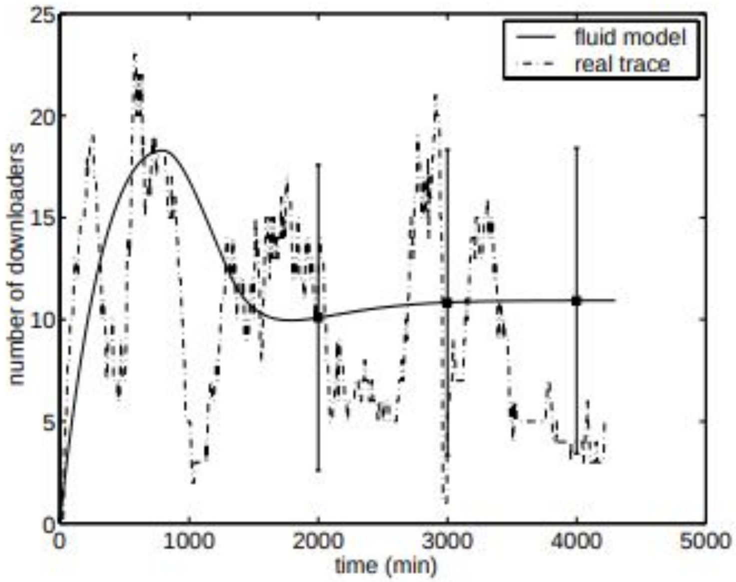
Experiment 3

Evolution of the number of seeds:



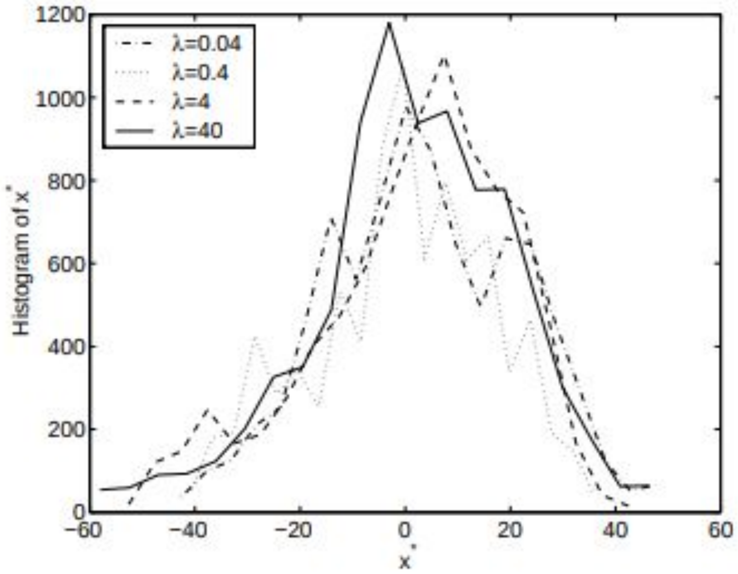
Experiment 3

Evolution of the number of downloaders:



Experiment 3

Histogram of the variation of the number of downloaders around the fluid model:

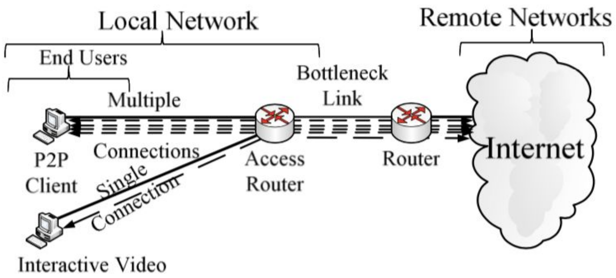


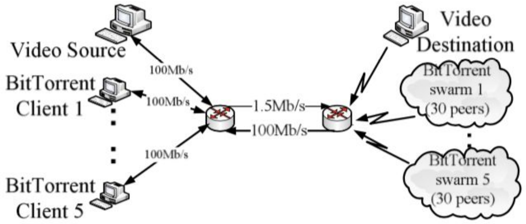
Real-Time Video Applications

- Analyze performance of BitTorrent on interactive video traffic
- Understand the impact of video traffic in an ISP network
- Minimize delay and jitter
- Tolerating small number of packet drops
- Bottleneck

Bottleneck and Performance Issues

- Performance problems for end-users
- Multiple TCP connections - unfair bandwidth allocation
- 2 types of bottleneck
  - Controllable
  - Uncontrollable



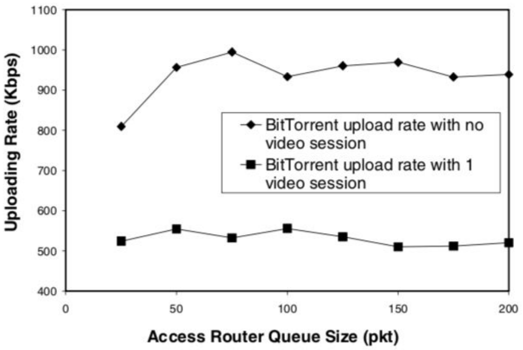


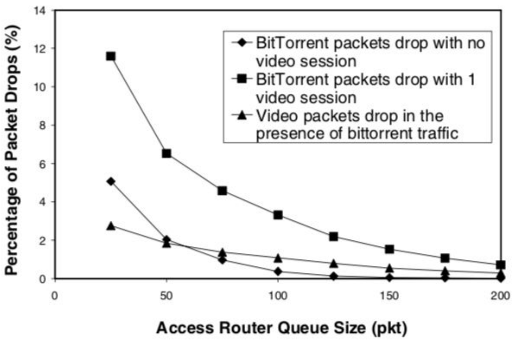
Drop-tail queue discipline

- A queue management algorithm
- Used by network schedulers
- Algorithm for deciding dropping of packets
- If queue is full , packets are dropped

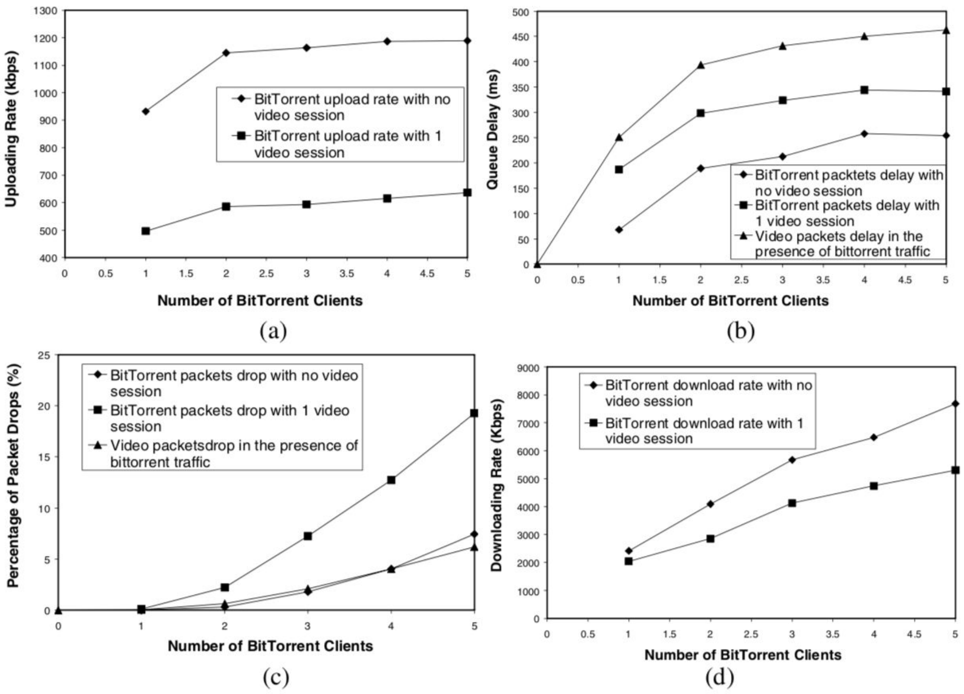
Analysis

- Two different simulation configuration
  - BitTorrent traffic with no video session
  - BitTorrent traffic with one video session
- Impact of varying Access router queue size
  - Percentage of Packet drops
  - Uploading rate





- Impact of varying the number of local BitTorrent clients from 0 up to 5



- Uploading Rate
- Queue Delay
- Percentage of Packet drops
- Downloading rate

Advantages of BitTorrents/P2P Networks

- Eliminates the extra cost to set up and maintain a server
- Helps the Internet as a whole by spreading traffic more evenly and helping to reduce congestion.
- Easy to set up and maintain, as each computer on the network is equipped to maintain itself.
- Distributing popular files which have high traffic for relatively short periods
- Unlike traditional server/client downloads, high traffic leads to more efficient file sharing via BitTorrent.

Disadvantages of BitTorrents/P2P Networks

- The absence of a centralized server makes it difficult to backup data as data is located on different workstations.
- There is no central point of data storage for file archiving.
- Security is weak as each system manages itself only.
- Lack of Collaboration from Peers
- An easy distribution method for pirated/illegal content
- Cannot modify/update the file to newer versions once the torrent has been distributed
- The IP of all peers and info of files they are downloading are publicly available on trackers

THANK YOU