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

Dongyu Qiu, R. Srikant

Department of Electrical and Computer Engineering

and

Coordinated Science Lab

University of Illinois

Introduction

- Peer-to-peer networks:
  - Peers participate in an application level overlay network and operate as both servers and clients.
  - Scalable: the service burden is distributed to all participating peers.
- Applications: File sharing, distributed directory service, web cache, storage, and grid computation etc.
- P2P file sharing: Kazza, Gnuttella, eDonkey/Overnet, BitTorrent.
- In some segments of the Internet, P2P traffic accounts for 40% of the Internet traffic.

Overview of BitTorrent

- Seed is a server that has the entire file

Overview of BitTorrent

- Seed is a server that has the entire file
- File stored in many pieces of 256 KB each

Overview of BitTorrent

- Seed is a server that has the entire file
- File stored in many pieces of 256 KB each
- A new peer starts downloading some pieces from the seed

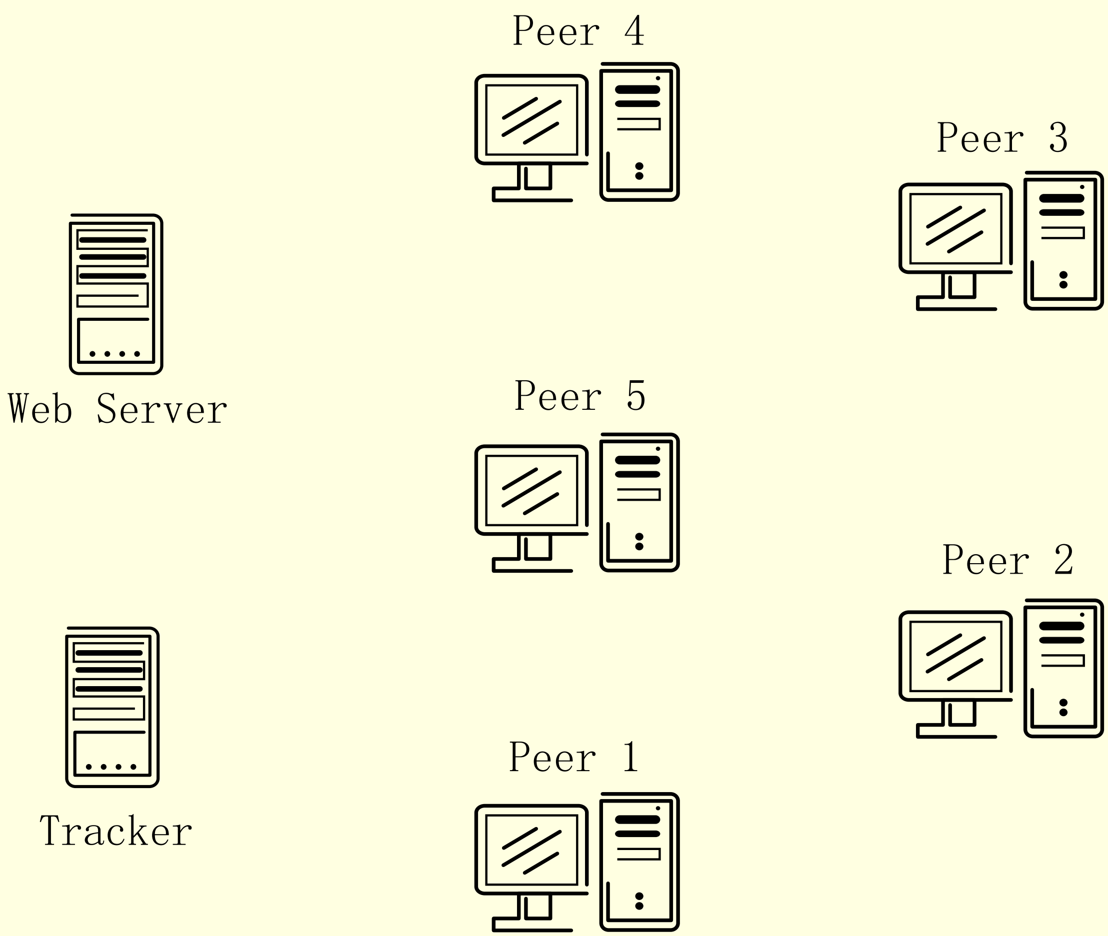
Overview of BitTorrent

- Seed is a server that has the entire file
- File stored in many pieces of 256 KB each
- A new peer starts downloading some pieces from the seed
- If there are many peers, each peer will have different parts of the file

Overview of BitTorrent

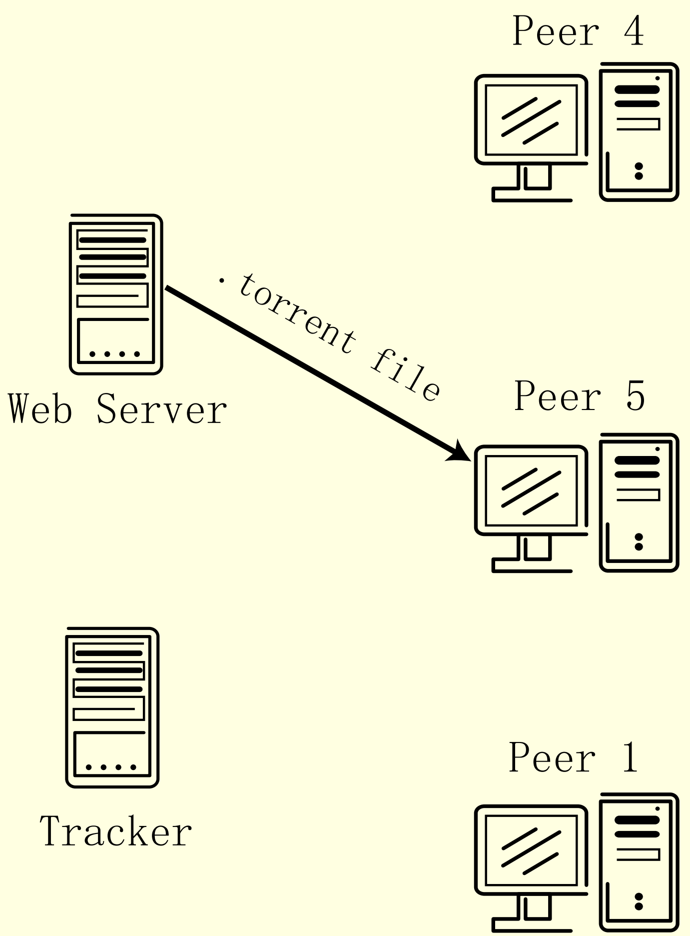
- Seed is a server that has the entire file
- File stored in many pieces of 256 KB each
- A new peer starts downloading some pieces from the seed
- If there are many peers, each peer will have different parts of the file
- Key point: Peers download from each other

Overview of BitTorrent



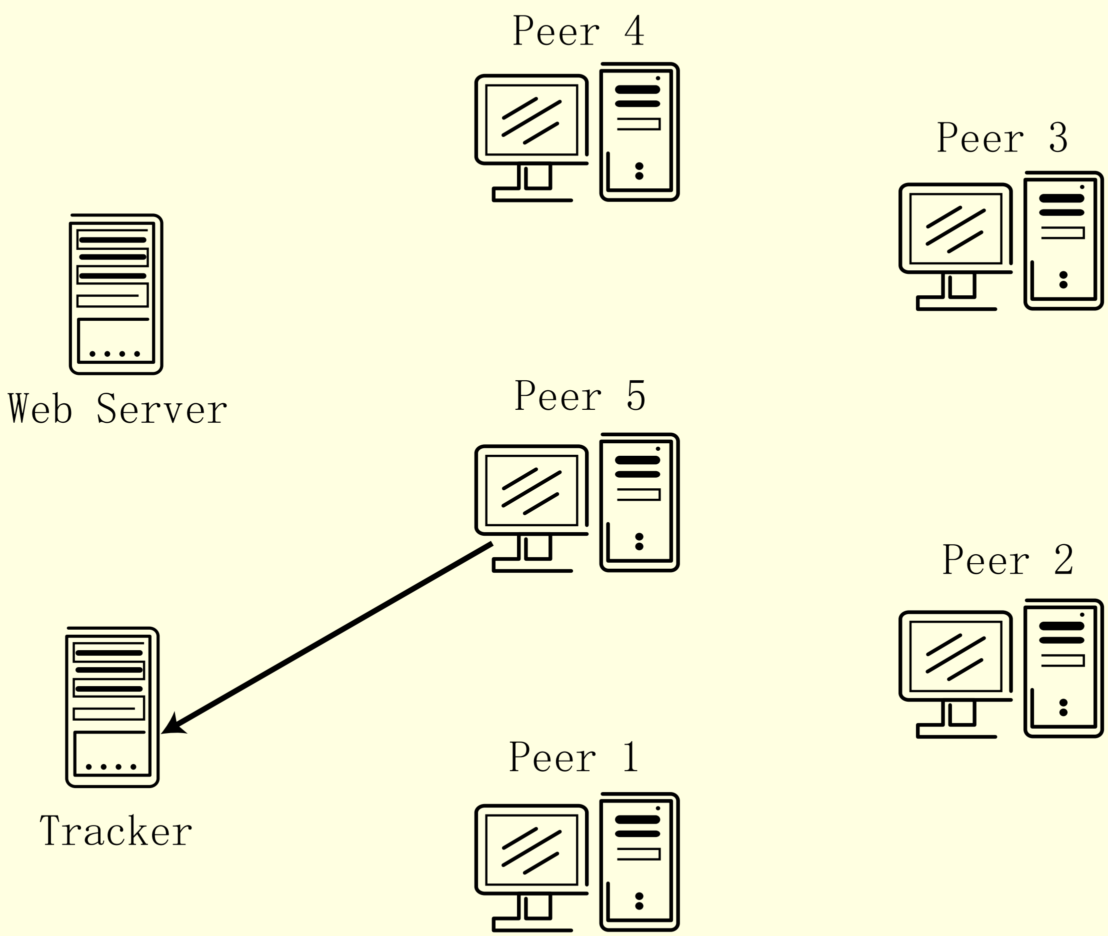
- Peer 5 joins the network

Overview of BitTorrent



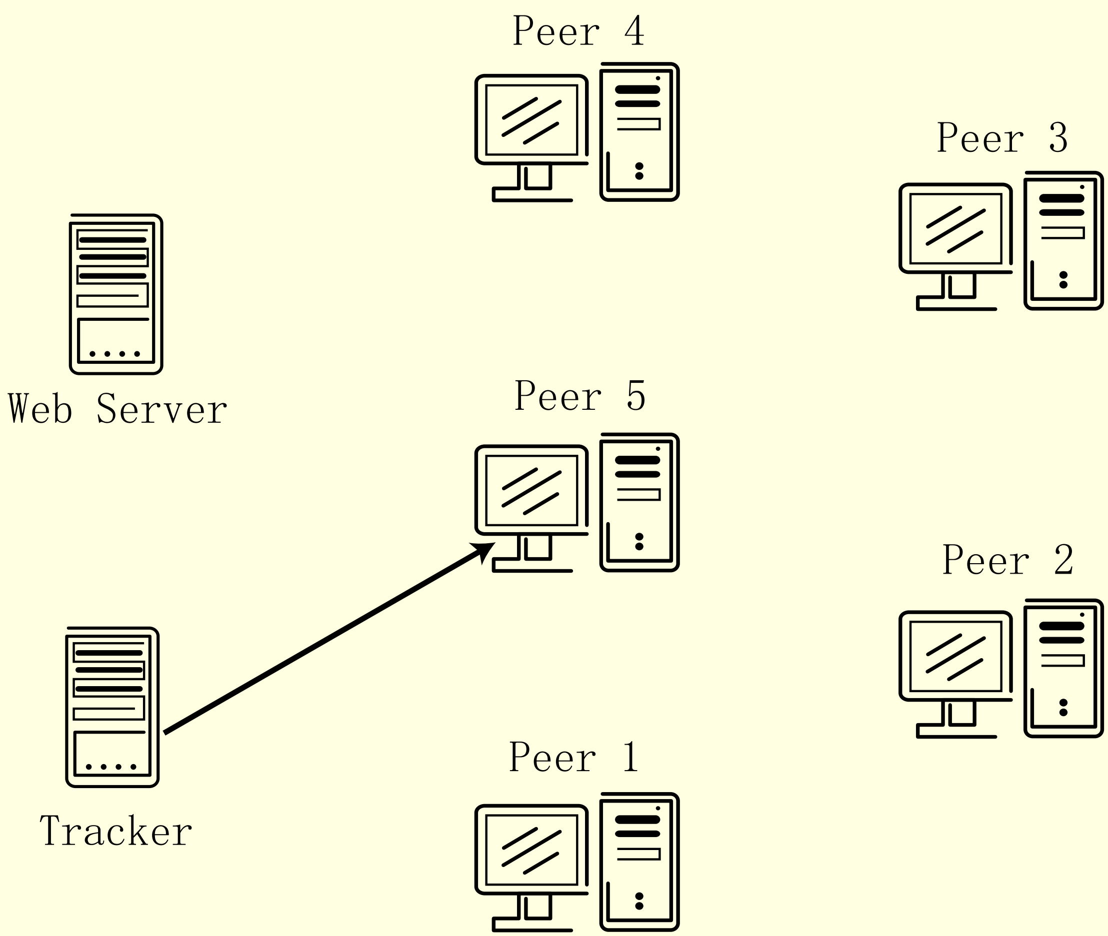
- Peer 5 joins the network
- downloads a .torrent file

Overview of BitTorrent



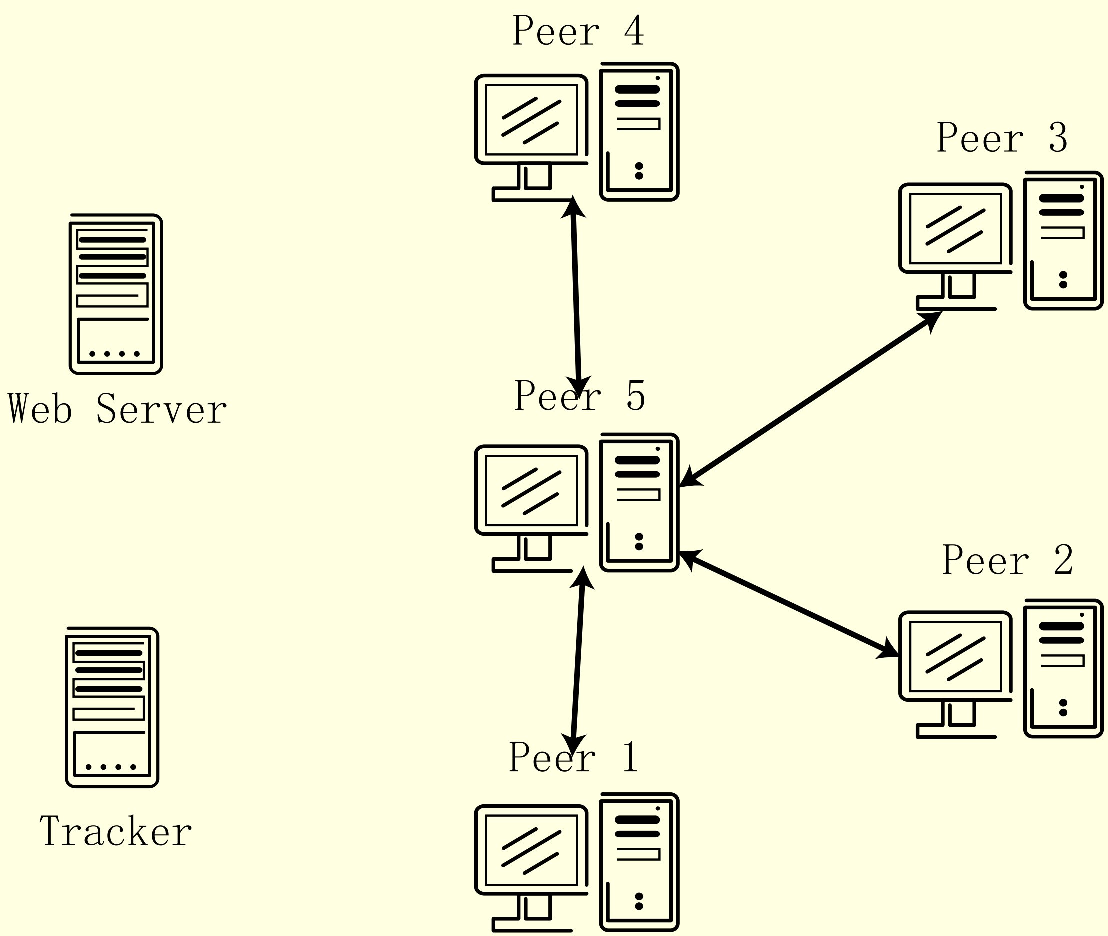
- Peer 5 joins the network
- downloads a .torrent file
- connects to the tracker

Overview of BitTorrent



- Peer 5 joins the network
- downloads a .torrent file
- connects to the tracker
- the tracker returns peer information

Overview of BitTorrent



- Peer 5 joins the network
- downloads a .torrent file
- connects to the tracker
- the tracker returns peer information
- connects to other peers and begins downloading

Terminology

- A peer which has the entire file is called a seed

Terminology

- A peer which has the entire file is called a seed
- A peer that is not a seed is called a downloader

Terminology

- A peer which has the entire file is called a seed
- A peer that is not a seed is called a downloader
- A downloader becomes a seed once it has the entire file

Terminology

- A peer which has the entire file is called a seed
- A peer that is not a seed is called a downloader
- A downloader becomes a seed once it has the entire file
- Each peer uploads to five other peers who give it the best download rates

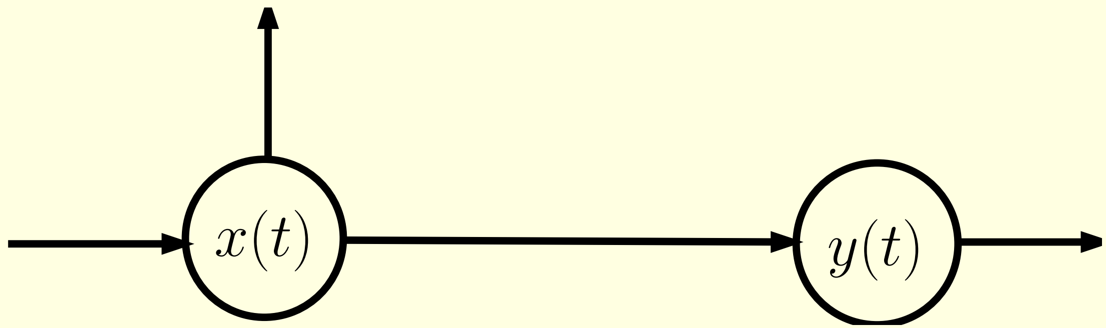
Terminology

- A peer which has the entire file is called a seed
- A peer that is not a seed is called a downloader
- A downloader becomes a seed once it has the entire file
- Each peer uploads to five other peers who give it the best download rates
- Every 30 seconds, drops its connection to peer with the smallest download rate and picks a new one at random

Issues to Be Addressed

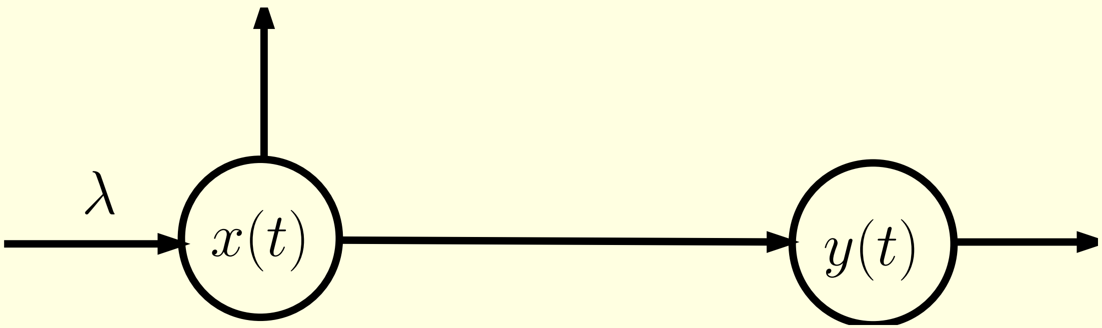
- Peer Evolution
- Scalability
- File Sharing Efficiency
- Incentive Mechanisms and its Effect.

Model



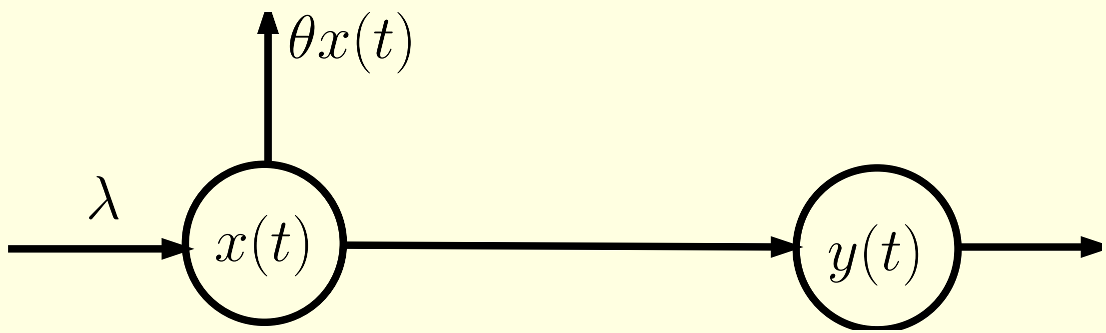
- $x(t)$: number of downloaders, $y(t)$: number of seeds

Model



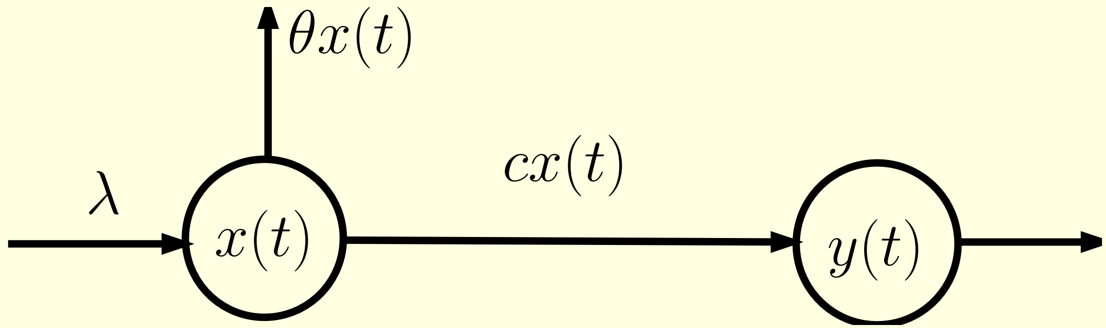
- $x(t)$: number of downloaders, $y(t)$: number of seeds
- λ: arrival rate of new requests

Model



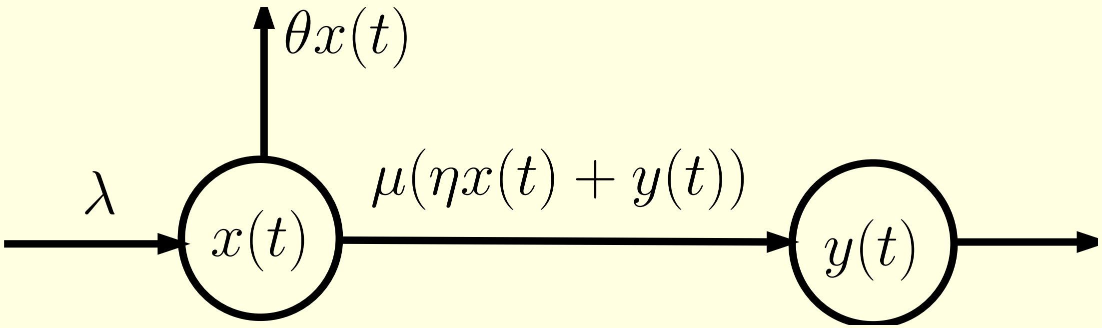
- $x(t)$: number of downloaders, $y(t)$: number of seeds
- λ: arrival rate of new requests
- θ: the rate at which downloaders abort the download

Model



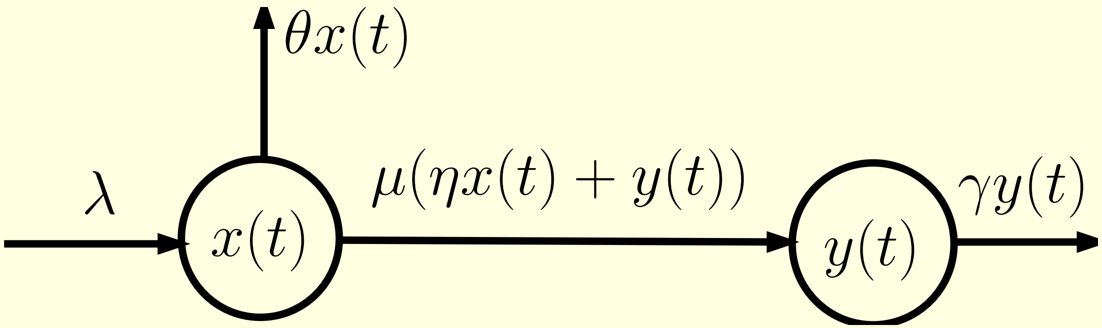
- $x(t)$: number of downloaders, $y(t)$: number of seeds
- λ: arrival rate of new requests
- θ: the rate at which downloaders abort the download
- c : downloading bandwidth, μ: uploading bandwidth of a peer

Model



- $x(t)$: number of downloaders, $y(t)$: number of seeds
- λ: arrival rate of new requests
- θ: the rate at which downloaders abort the download
- c : downloading bandwidth, μ: uploading bandwidth of a peer
- η: effectiveness parameter (Yang and de Veciana)

Model



- $x(t)$: number of downloaders, $y(t)$: number of seeds
- λ: arrival rate of new requests
- θ: the rate at which downloaders abort the download
- c : downloading bandwidth, μ: uploading bandwidth of a peer
- η: effectiveness parameter (Yang and de Veciana)
- γ: seed departure rate

A Simple Fluid Model

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

Comparison with download from a single server:

- Assume $\theta = 0, c = \infty$
- Departure rate of downloaders is μ, independent of x
- Need $\lambda < \mu$ for stability
- P2P is scalable, but single-server download is not

Steady-State Performance

- If $\frac{1}{c} \geq \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\gamma})$, the downloading bandwidth is the constraint:

$$\bar{x} = \frac{\lambda}{c(1 + \frac{\theta}{c})}, \qquad \bar{y} = \frac{\lambda}{\gamma(1 + \frac{\theta}{c})}$$

Steady-State Performance

- If $\frac{1}{c} \geq \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\gamma})$, the downloading bandwidth is the constraint:

$$\bar{x} = \frac{\lambda}{c(1 + \frac{\theta}{c})}, \qquad \bar{y} = \frac{\lambda}{\gamma(1 + \frac{\theta}{c})}$$

- If $\frac{1}{c} \leq \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\gamma})$, the uploading bandwidth is the constraint:

$$\bar{x} = \frac{\lambda}{\nu(1 + \frac{\theta}{\nu})}, \qquad \bar{y} = \frac{\lambda}{\gamma(1 + \frac{\theta}{\nu})},$$

where $\frac{1}{\nu} = \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\gamma})$.

Steady-State Performance

- Little's law: Average download time

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

Steady-State Performance

- Little's law: Average download time

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

- Scalability: T is not a function of λ, the request arrival rate

Steady-State Performance

- Little's law: Average download time

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

- Scalability: T is not a function of λ, the request arrival rate
- Need for incentives: When the seed departure rate γ increases, T increases

Steady-State Performance

- Little's law: Average download time

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

- Scalability: T is not a function of λ, the request arrival rate
- Need for incentives: When the seed departure rate γ increases, T increases
- Initially when the downloading rate c increases, T decreases. However, beyond a point the uploading rate becomes the bottleneck

Steady-State Performance

- Little's law: Average download time

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

- Scalability: T is not a function of λ, the request arrival rate
- Need for incentives: When the seed departure rate γ increases, T increases
- Initially when the downloading rate c increases, T decreases. However, beyond a point the uploading rate becomes the bottleneck
- Prior work assumes $c = \infty$ (motivated by the asymmetry in cable modem and DSL rates): doesn't capture the above effect

Effectiveness of File Sharing

- For a given downloader i, we assume that it is connected to k other downloaders.

$$\eta = 1 - \mathsf{P}\left\{ \begin{array}{c} \text{downloader } i \text{ has no piece that} \\ \text{the connected peers need} \end{array} \right\}.$$

- We assume that the piece distributions between different peers are independent and identical. Then

$$\eta = 1 - \mathsf{P}\left\{ \begin{array}{c} \text{downloader } j \text{ needs no} \\ \text{piece from downloader } i \end{array} \right\}^k,$$

  where j is a downloader connected to i.

- For each downloader, we assume that the number of pieces it has is uniformly distributed in $\{0, \cdots, N-1\}$, where N is the number of pieces of the served file.

Effectiveness of File Sharing

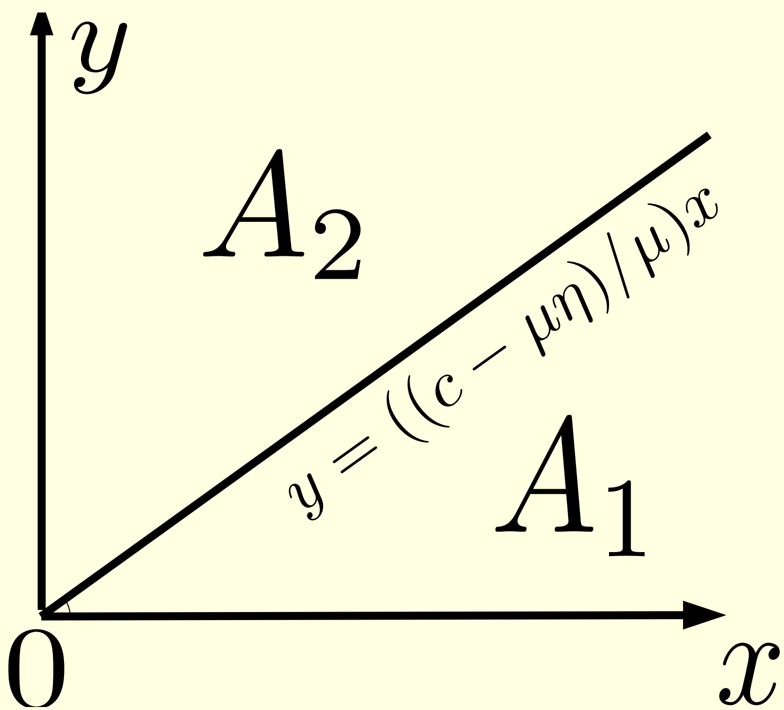
- Let n_i be number of pieces at downloader i.

$$
\begin{aligned}
& \mathsf{P}\left\{ \begin{array}{c} \text{downloader } j \text{ needs no} \\ \text{piece from downloader } i \end{array} \right\} \\
= \; & \mathsf{P}\{j \text{ has all pieces of downloader } i\} \\
= \; & \sum_{n_j=1}^{N-1} \sum_{n_i=0}^{n_j} \frac{1}{N^2} \mathsf{P}\{j \text{ has all pieces of } i | n_i, n_j\} \\
= \; & \sum_{n_j=1}^{N-1} \sum_{n_i=0}^{n_j} \frac{1}{N^2} \frac{\binom{N-n_i}{n_j-n_i}}{\binom{N}{n_j}} \\
= \; & \frac{N+1}{N^2} \sum_{m=2}^{N} \frac{1}{m} \approx \frac{\log N}{N}
\end{aligned}
$$

Effectiveness of File Sharing

- $$\eta \approx 1 - \left(\frac{\log N}{N}\right)^k$$
- In BitTorrent, each piece is typically $256KB$. For a file that is a few hundreds of megabytes in size, N is of the order of several hundreds.
- Even if $k = 1$, η is very close to one.
- When k increases, η also increases but very slowly and the network performance increases slowly. Hence, when λ increases, the network performance increases but very slowly.
- When $k = 0$, $\eta = 0$.

Stability



- $$\mathbf{A}_1 = \begin{bmatrix} -(\mu\eta + \theta) & -\mu \\ \mu\eta & -(\gamma - \mu) \end{bmatrix}.$$
- $$\mathbf{A}_2 = \begin{bmatrix} -(\theta + c) & 0 \\ c & -\gamma \end{bmatrix}$$
- A_1 is a stable matrix, but A_2 may not be a stable matrix
- However, the system is globally stable

Characterizing Variability

- How does the number of seeds and downloaders vary around the numbers predicted by the deterministic model?
- Peers arrive according to a Poisson process.
- Download times are exponentially distributed
- $$x(t) + \sqrt{\lambda}\hat{x}(t), \quad y(t) + \sqrt{\lambda}\hat{y}(t),$$

  respectively, where $\hat{\mathbf{X}} = (\hat{x}, \hat{y})^T$ are described by an Ornstein-Uhlenbeck process:

  $$\mathrm{d}\hat{\mathbf{X}}(t) = \mathbf{A}\hat{\mathbf{X}}(t)\mathrm{d}t + \mathbf{B}\mathrm{d}\mathbf{W}(t),$$
- $\mathbf{A} = \mathbf{A}_1$ or $\mathbf{A}_2$

Characterizing Variability

- $$\mathbf{B} = \begin{bmatrix} 1 & -\sqrt{\rho} & -\sqrt{(1-\rho)} & 0 \\ 0 & 0 & \sqrt{(1-\rho)} & -\sqrt{(1-\rho)} \end{bmatrix},$$
  where $\rho = T\theta/(T\theta + 1)$.
- In steady-state, the number of seeds and downloaders is Gaussian with covariance $\mathbf{\Sigma}$:
  $$\mathbf{A\Sigma} + \mathbf{\Sigma A}^T + \mathbf{BB}^T = 0.$$

Simulation Result

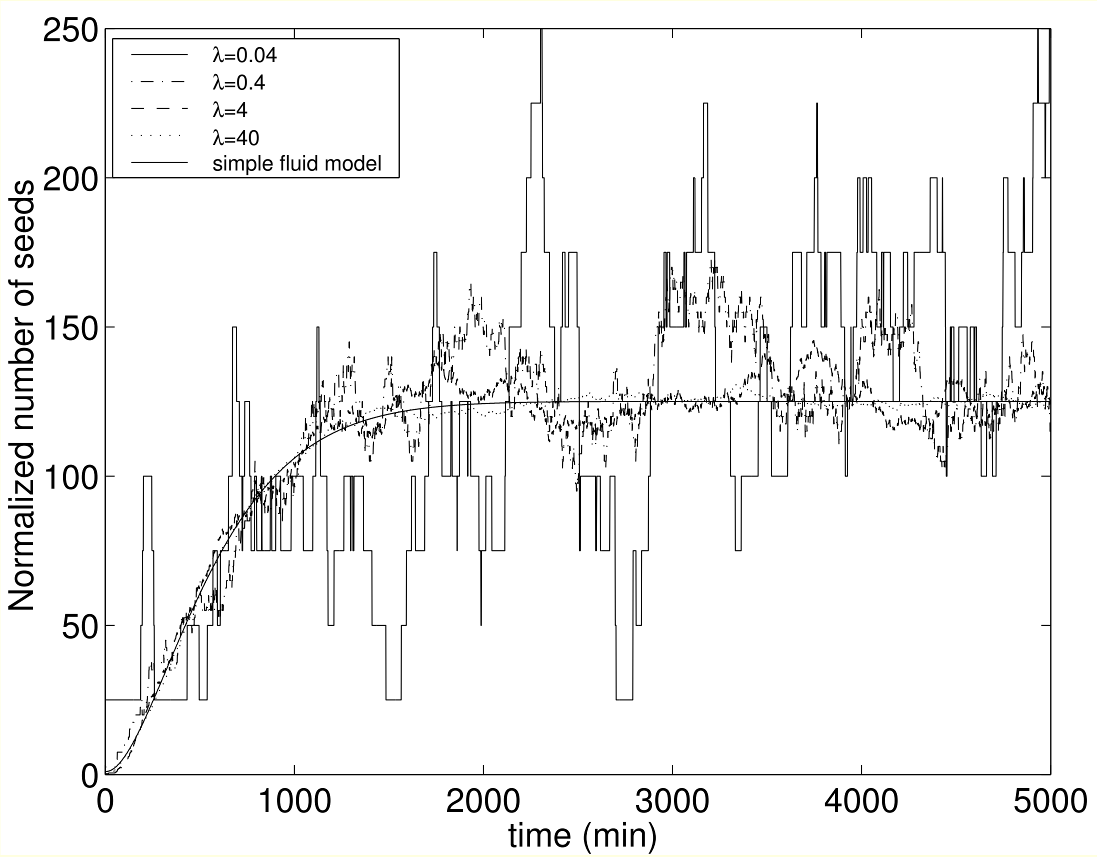


Figure 1: The evolution of the number of seeds as a function of time ($\mu = 0.00125$, $c = 0.002$, $\theta = 0.001$, $\gamma = 0.005$)

Simulation Result

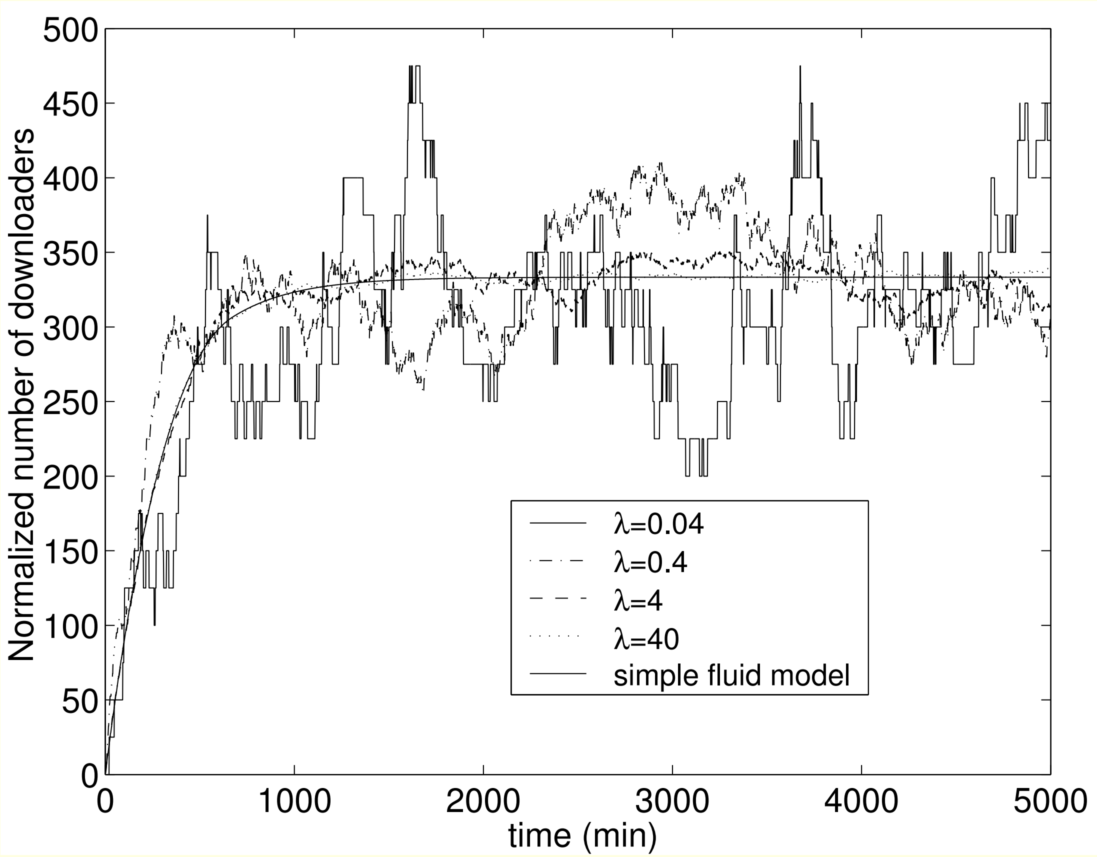


Figure 2: The evolution of the number of downloaders as a function of time ($\mu = 0.00125$, $c = 0.002$, $\theta = 0.001$, $\gamma = 0.005$)

Simulation Result

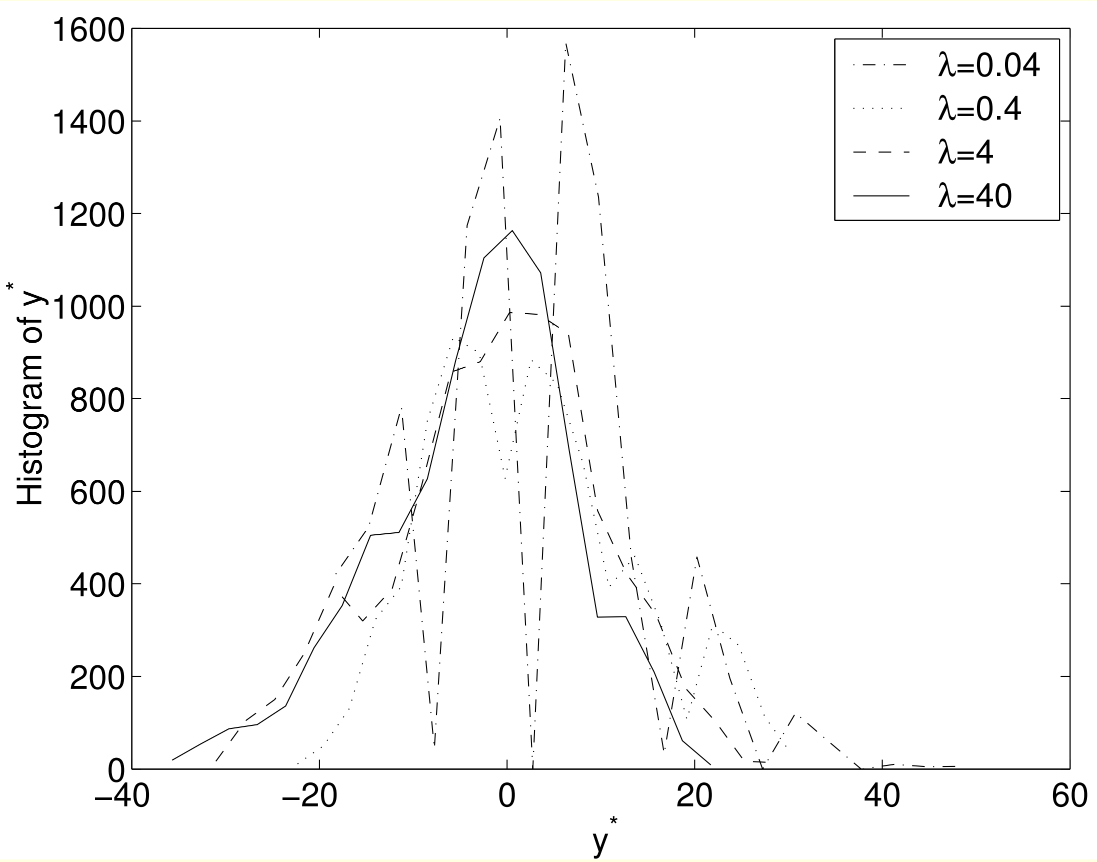


Figure 3: Histogram of the variation of the number of seeds around the fluid model ($\mu = 0.00125$, $c = 0.002$, $\theta = 0.001$, $\gamma = 0.005$)

Simulation Result

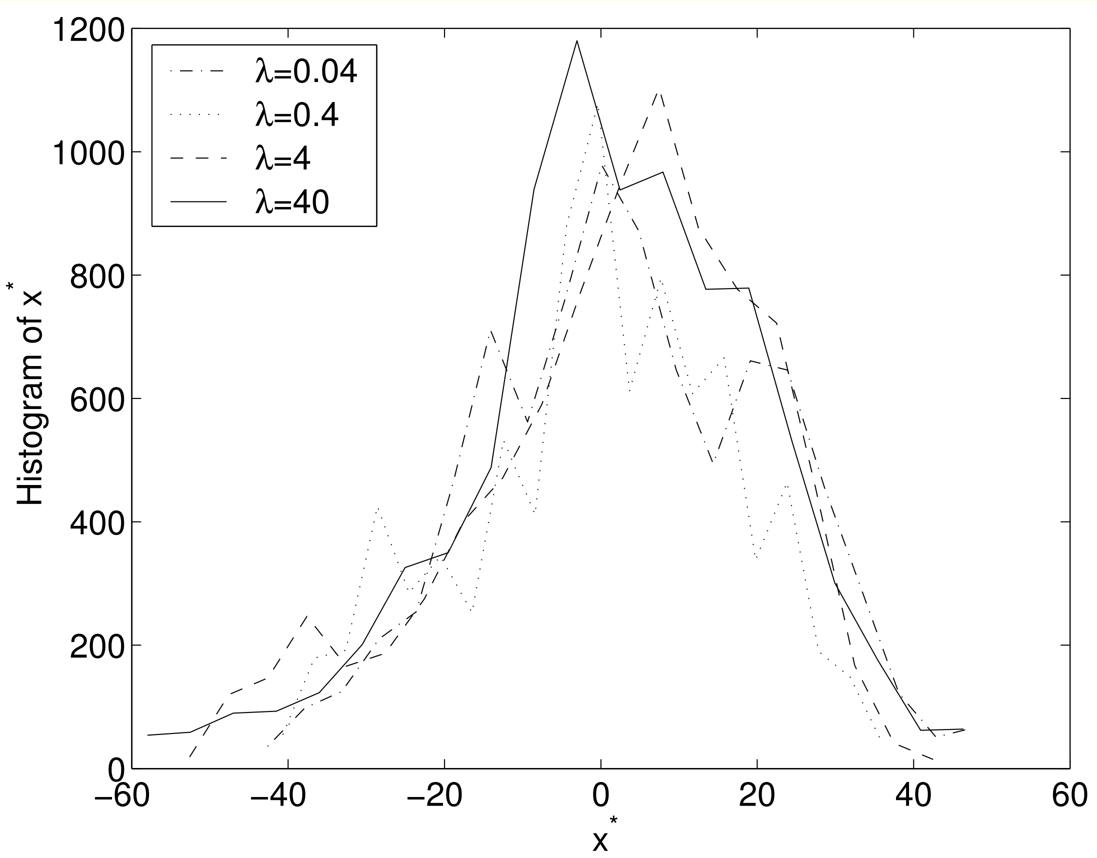


Figure 4: Histogram of the variation of the number of downloaders around the fluid model ($\mu = 0.00125$, $c = 0.002$, $\theta = 0.001$, $\gamma = 0.005$)

Experimental Result

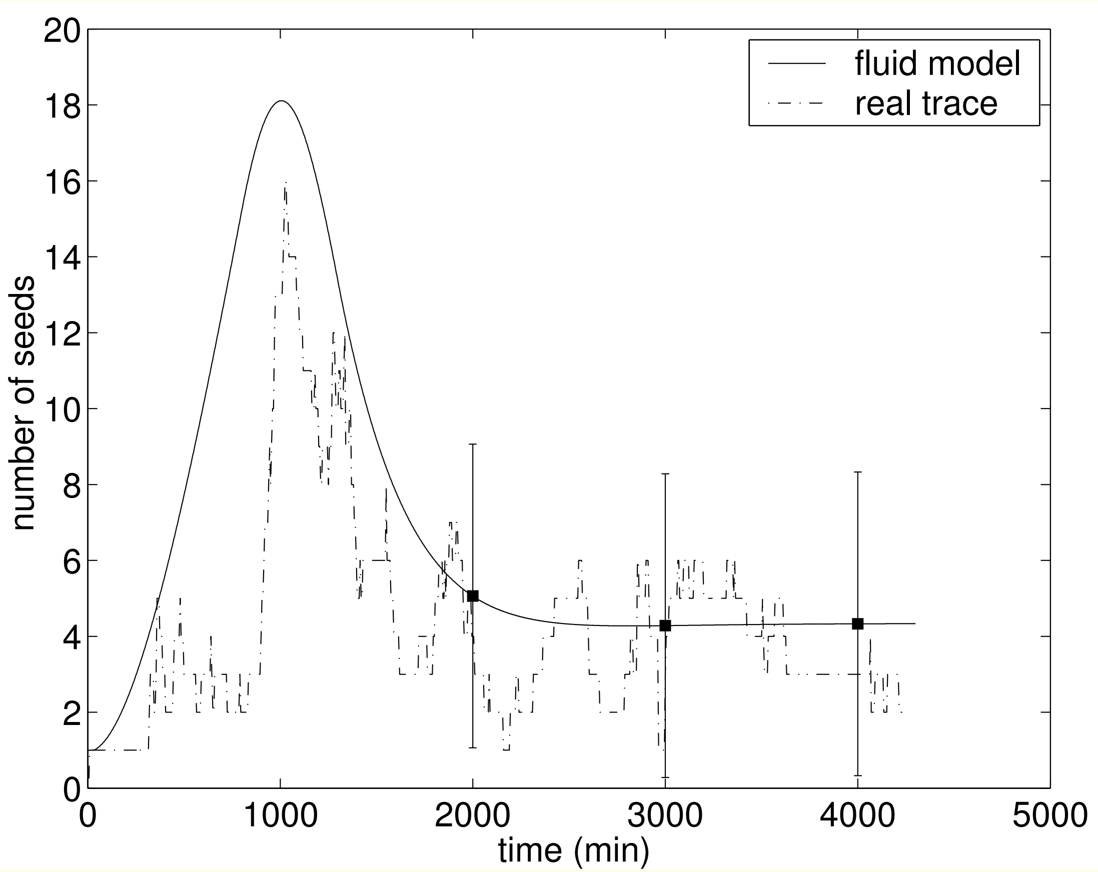


Figure 5: The evolution of the number of seeds as a function of time (real trace)

Experimental Result

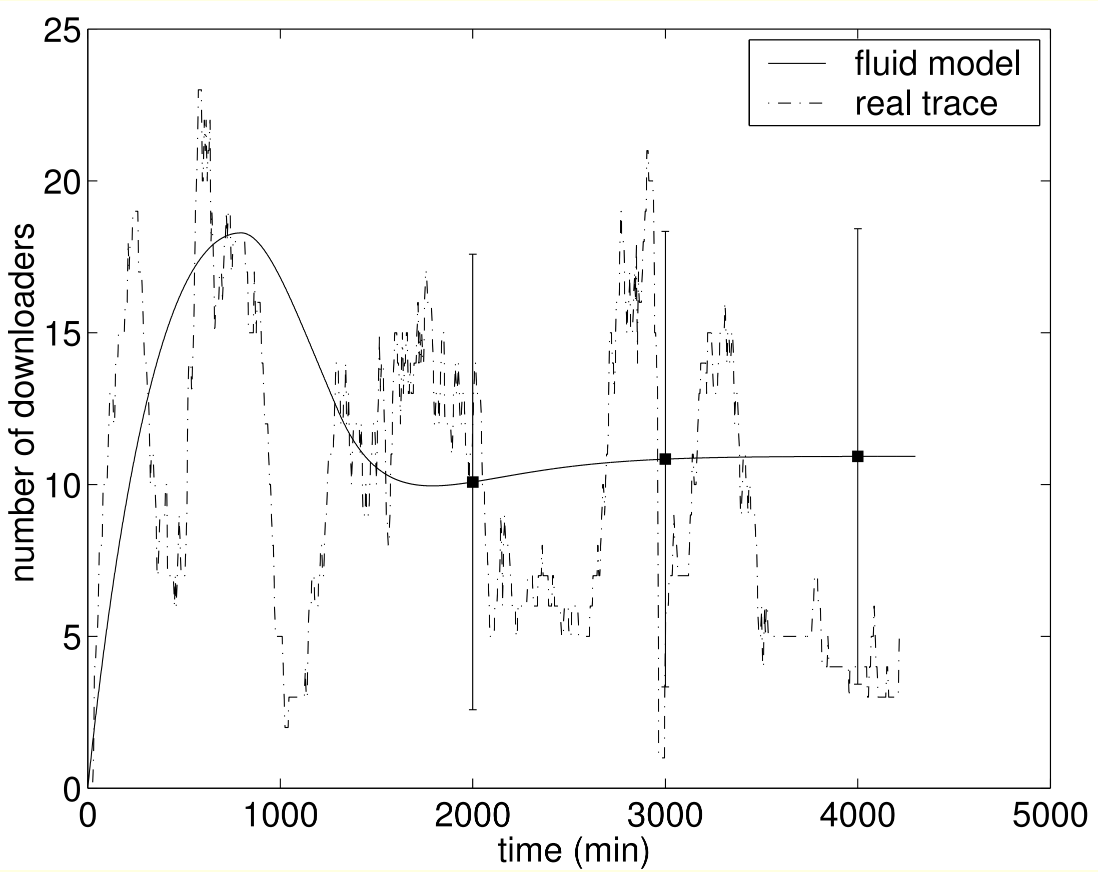


Figure 6: The evolution of the number of downloaders as a function of time (real trace)

Conclusions

- Presented a simple fluid model for BitTorrent-like networks
- Studied the steady-state network performance and stability
- Obtained insight into the effect of different parameters on network performance
- Not covered in this talk: the effect of the built-in incentive mechanism of BitTorrent on network performance and the effect of optimistic unchoking on free-riding