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

Dongyu Qiu and R. Srikant

Coordinated Science Laboratory

University of Illinois at Urbana-Champaign

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.

Related Work

- P2P system design and traffic measurement [Ripeanu 2001, Ripeanu et al 2002, Eugene et al 2003]
- Stochastic fluid model for P2P web cache [Clevenot et al 2003]
- Simple Markovian model and service capacity for BitTorrent-Like P2P file sharing [Yang and de Veciana 2004]

Overview of BitTorrent

- Process starts with a server called the seed which has the file of interest

Overview of BitTorrent

- Process starts with a server called the seed which has the file of interest
- File stored in many pieces of 256 KB each

Overview of BitTorrent

- Process starts with a server called the seed which has the file of interest
- File stored in many pieces of 256 KB each
- As peers arrive, they download "random" pieces of the file from the seed

Overview of BitTorrent

- Process starts with a server called the seed which has the file of interest
- File stored in many pieces of 256 KB each
- As peers arrive, they download "random" pieces of the file from the seed
- Each peer may have different parts of the file

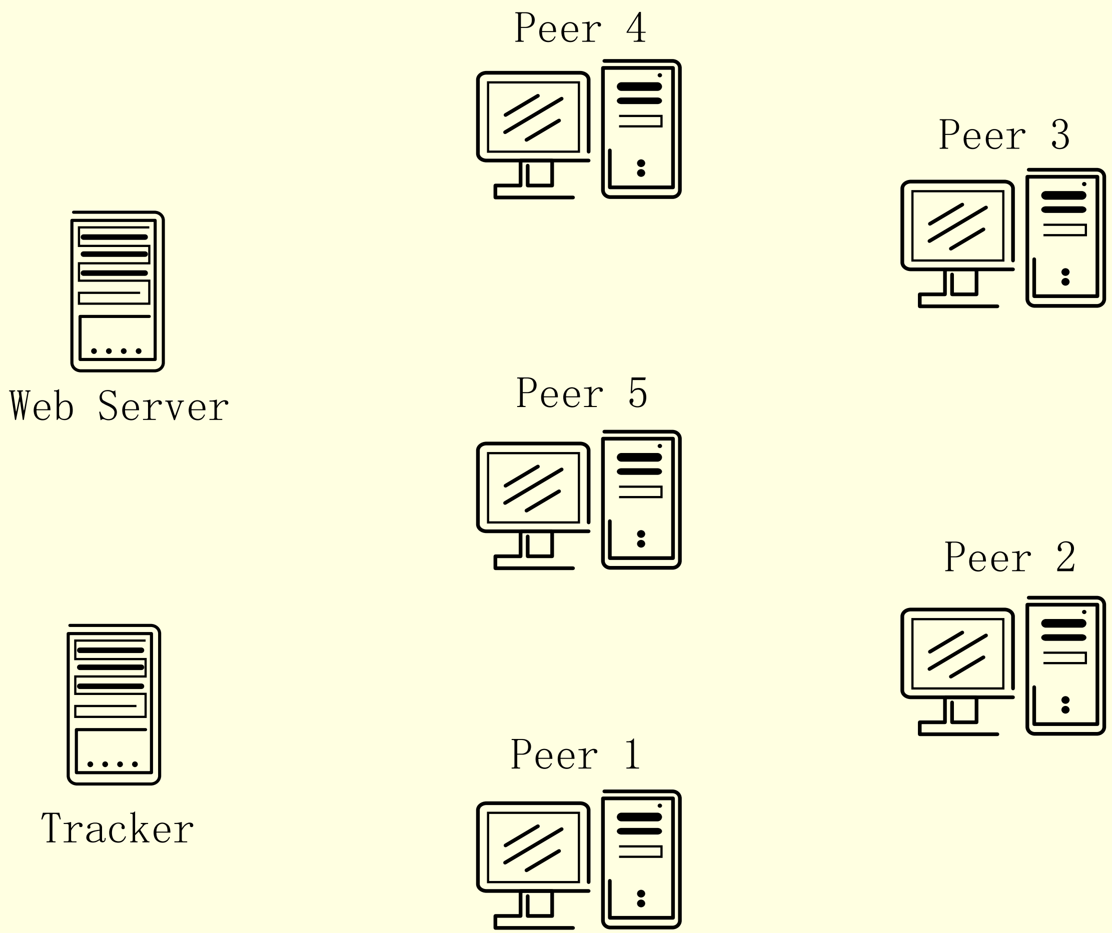
Overview of BitTorrent

- Process starts with a server called the seed which has the file of interest
- File stored in many pieces of 256 KB each
- As peers arrive, they download "random" pieces of the file from the seed
- Each peer may have different parts of the file
- Peers can act as servers even if they only have parts of the file.

Overview of BitTorrent

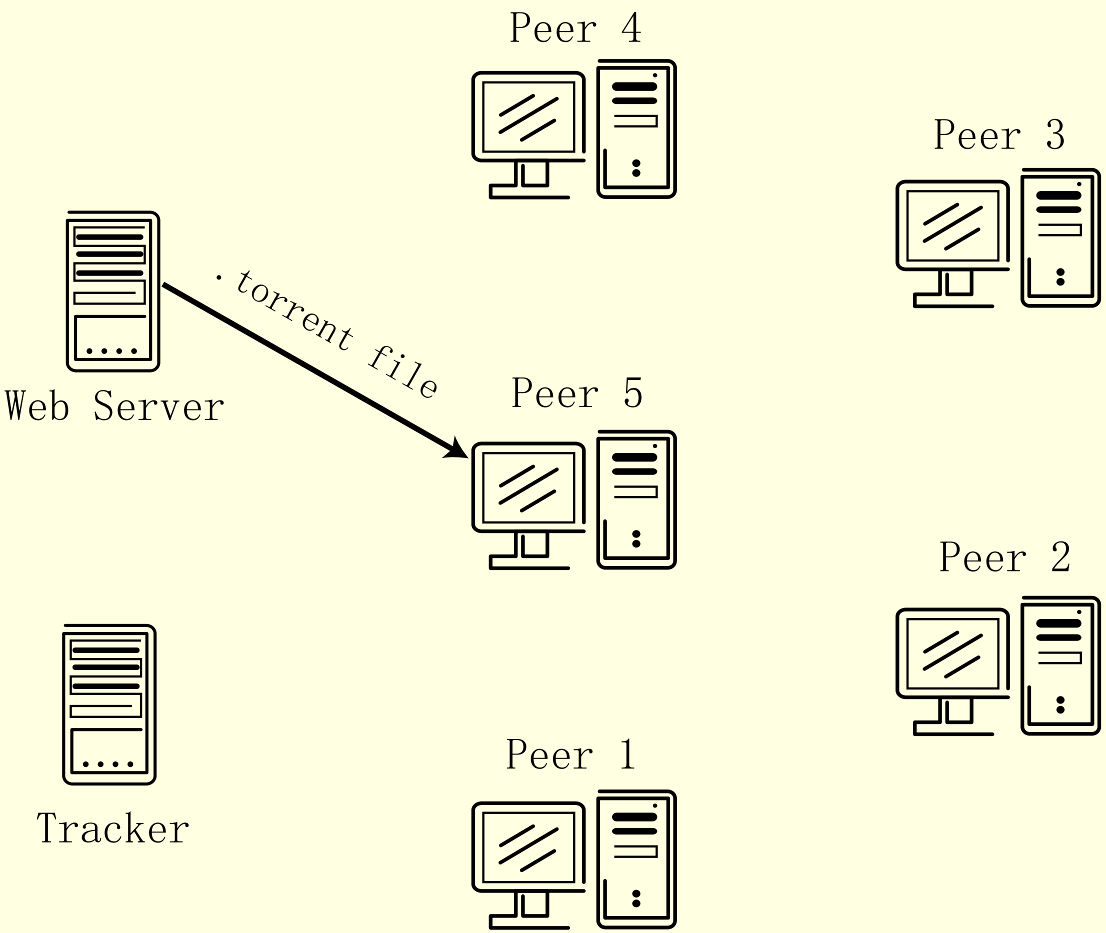
- Process starts with a server called the seed which has the file of interest
- File stored in many pieces of 256 KB each
- As peers arrive, they download "random" pieces of the file from the seed
- Each peer may have different parts of the file
- Peers can act as servers even if they only have parts of the file.
- Key point: Peers download from each other, while in traditional client/server system, clients only download from a single server

Overview of BitTorrent



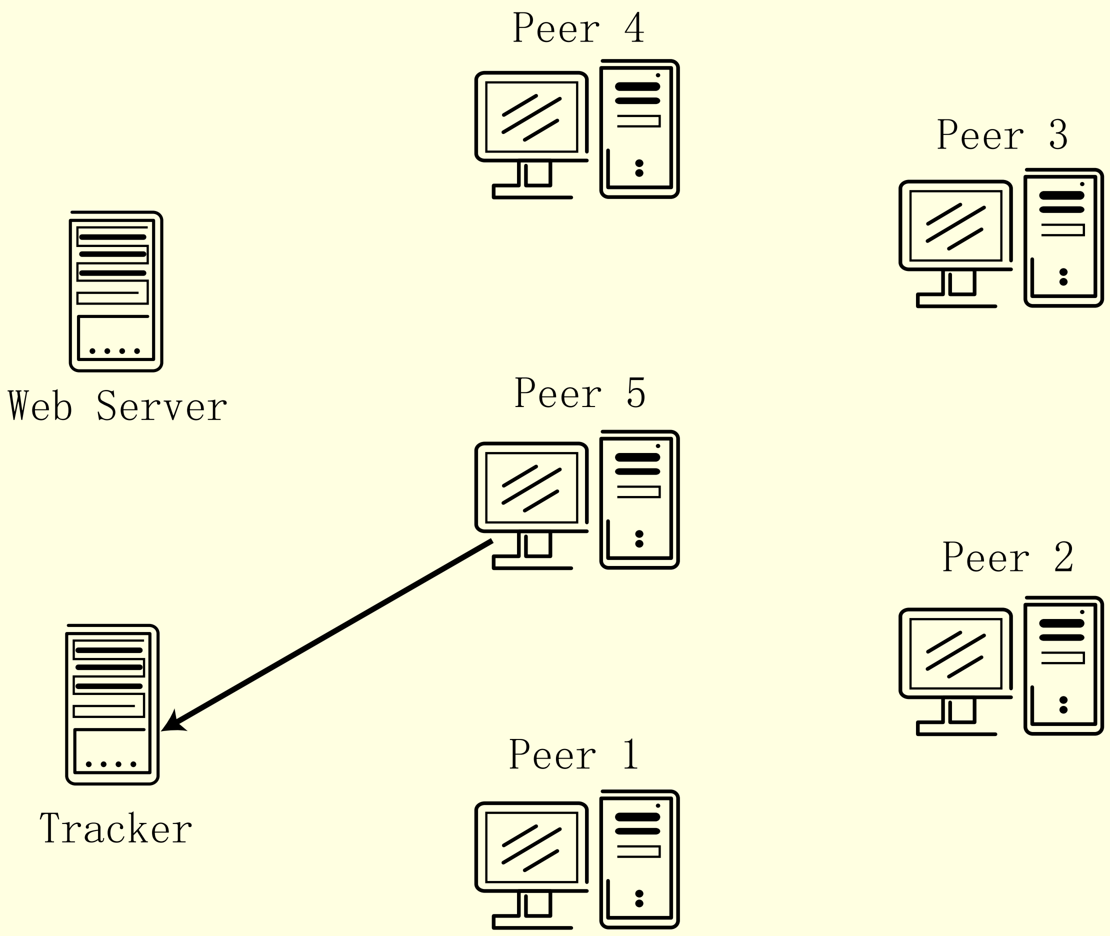
- Peer 5 joins the network

Overview of BitTorrent



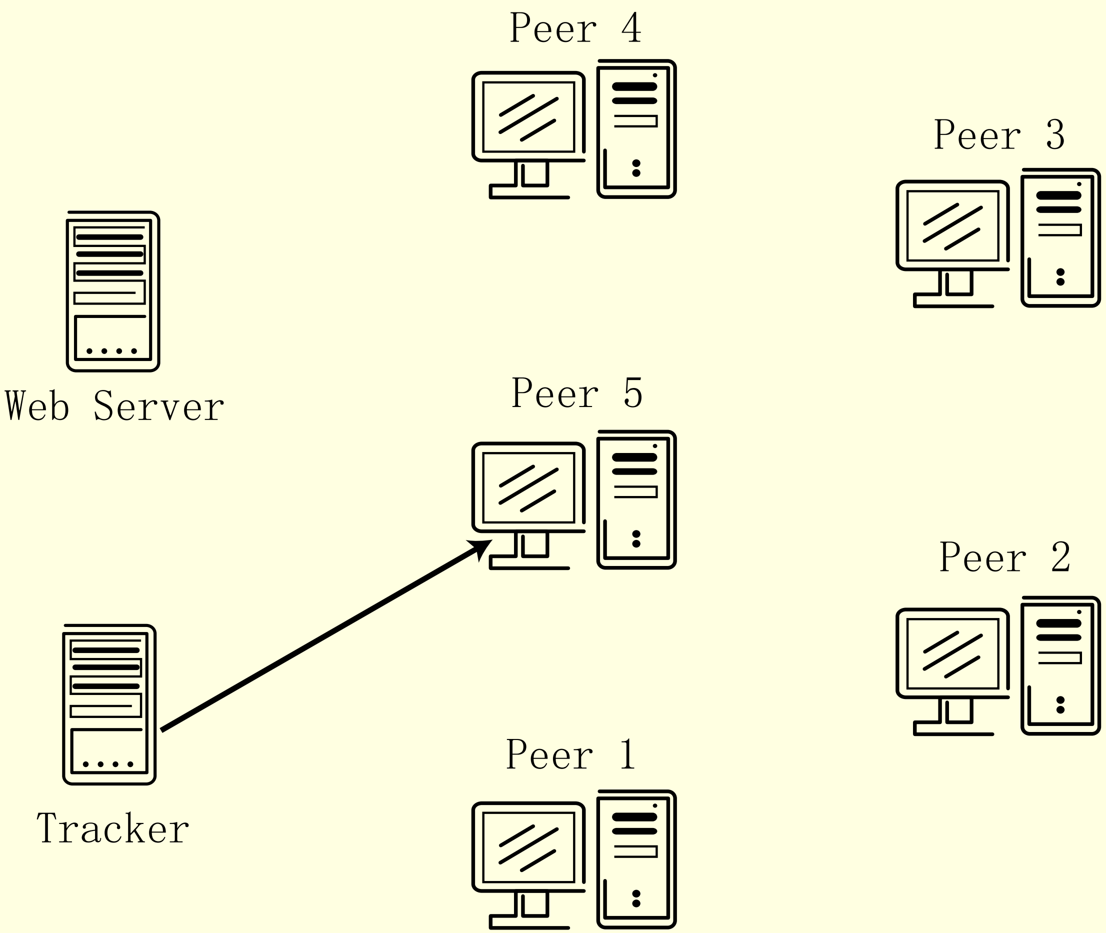
- Peer 5 joins the network
- downloads a .torrent file

Overview of BitTorrent



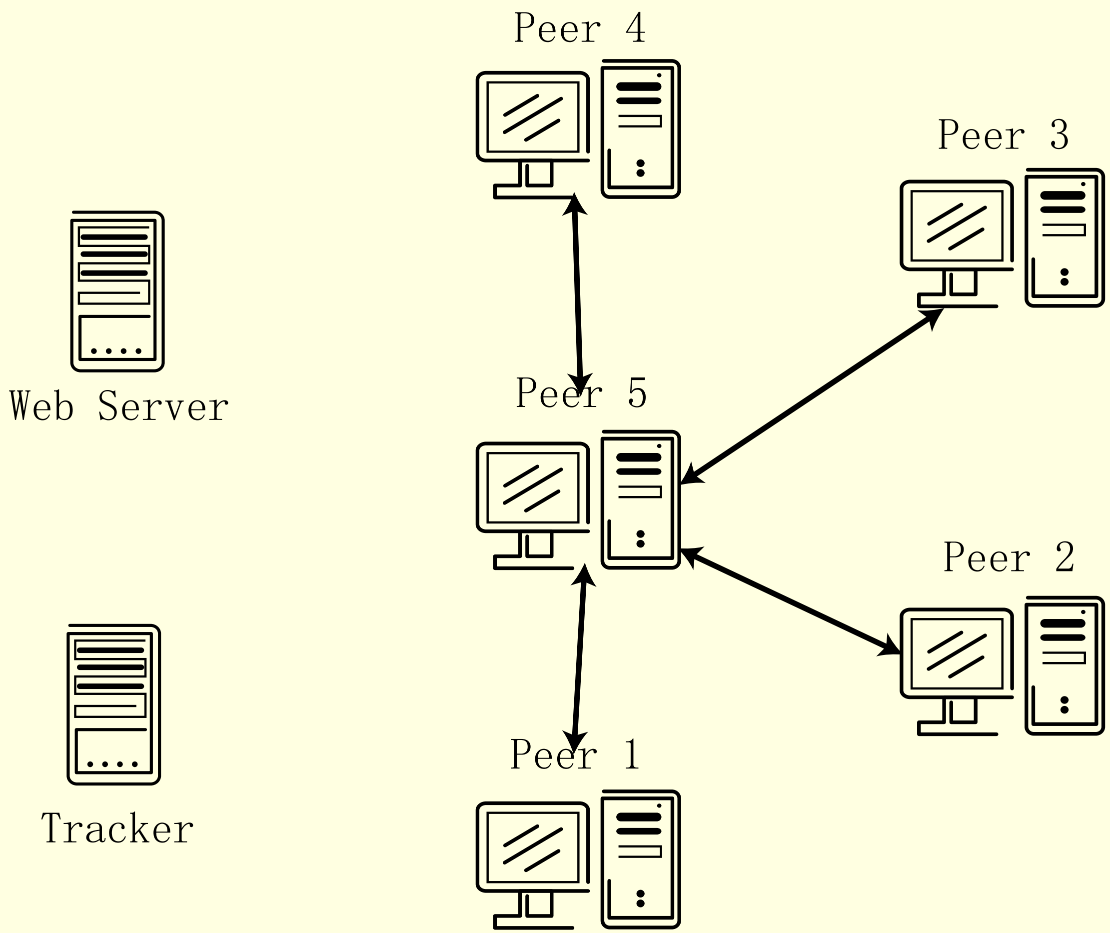
- Peer 5 joins the network
- downloads a .torrent file
- connects to the tracker

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

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
- Every 30 seconds, each peer drops its upload to the peer with the smallest download rate and picks a new one at random (Optimistic Unchoking)

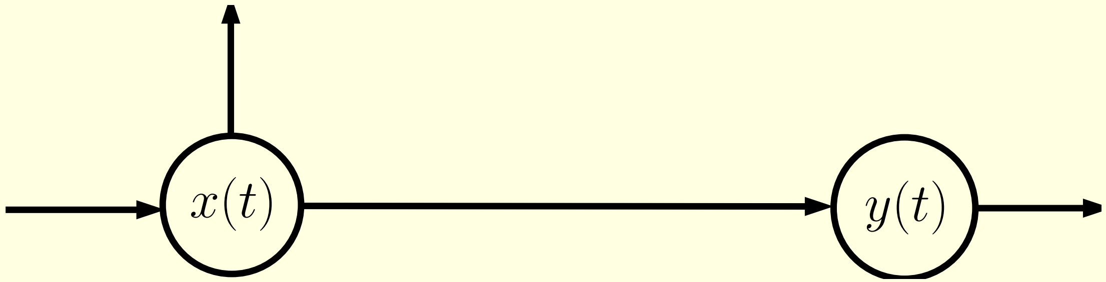
Terminology

- 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
- Every 30 seconds, each peer drops its upload to the peer with the smallest download rate and picks a new one at random (Optimistic Unchoking)
- Free-riding: Selfish peers tend to download at maximum rate while not uploading at all if they can get away with it

Issues to Be Addressed

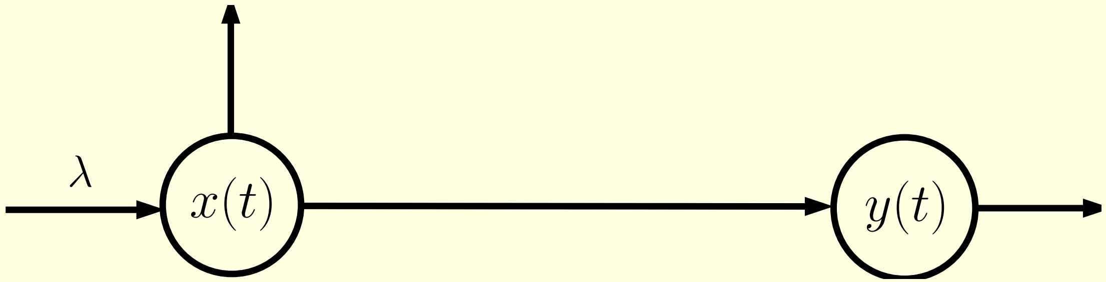
- Peer evolution
- Scalability
- Performance of the built-in incentive mechanism (Optimistic Unchoking) to combat free-riding

Model



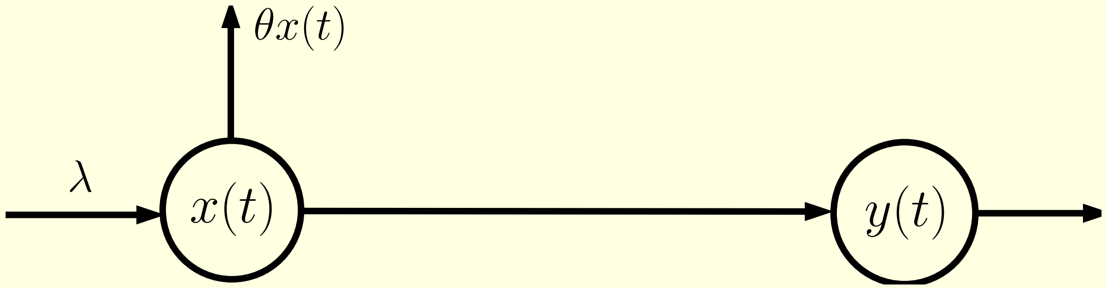
- $x(t)$: number of downloaders, $y(t)$: number of seeds

Model



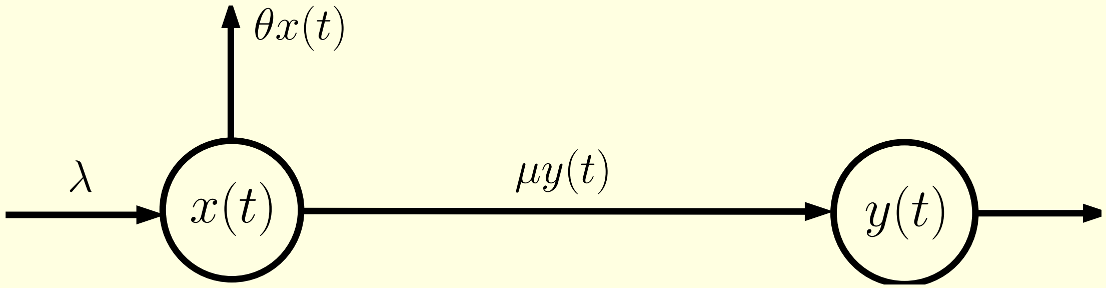
- $x(t)$: number of downloaders, $y(t)$: number of seeds
- λ: arrival rate of new requests

Model



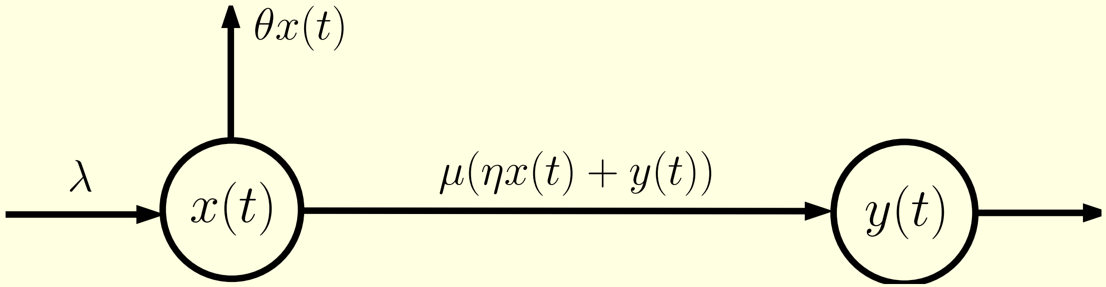
- $x(t)$: number of downloaders, $y(t)$: number of seeds
- λ: arrival rate of new requests
- θ: the rate at which a downloader aborts the download

Model



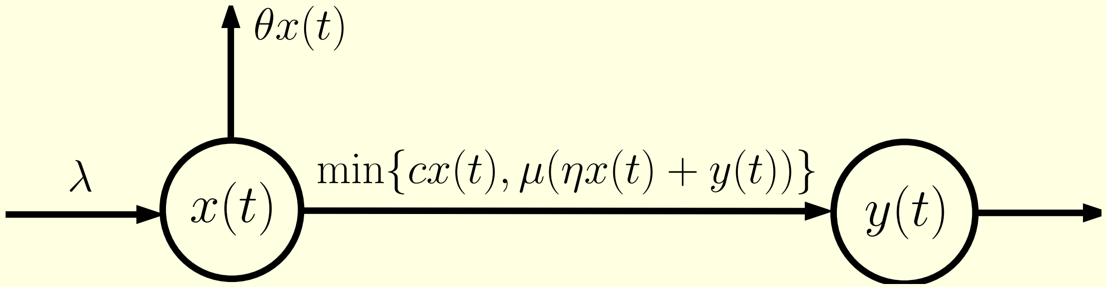
- $x(t)$: number of downloaders, $y(t)$: number of seeds
- λ: arrival rate of new requests
- θ: the rate at which a downloader aborts the download
- μ: uploading bandwidth of a peer

Model



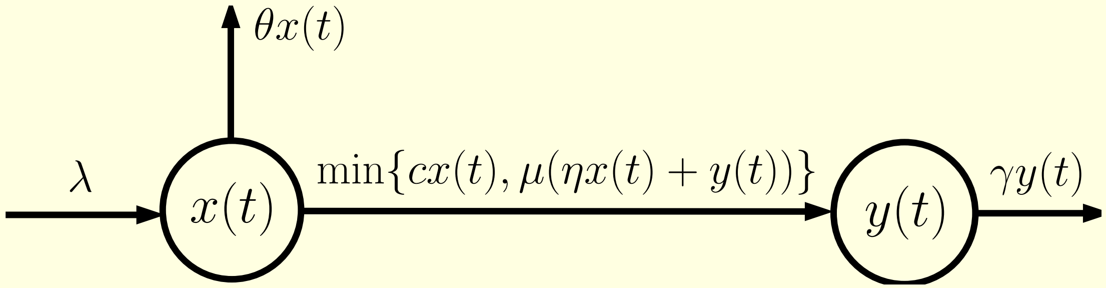
- $x(t)$: number of downloaders, $y(t)$: number of seeds
- λ: arrival rate of new requests
- θ: the rate at which a downloader aborts the download
- μ: uploading bandwidth of a peer
- η: effectiveness parameter (Yang and de Veciana)

Model



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

- Single server: service rate is fixed at μ, need $\lambda < \mu$ for stability
- P2P: service rate increases when number of peers increases
- 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 downloading time

$$T = \frac{1}{\theta + \beta}, \text{ where } \frac{1}{\beta} = \max\left\{\frac{1}{c}, \frac{1}{\eta}\left(\frac{1}{\mu} - \frac{1}{\gamma}\right)\right\}$$

Steady-State Performance

- Little's law: average downloading time

$$T = \frac{1}{\theta + \beta}, \text{ where } \frac{1}{\beta} = \max\left\{\frac{1}{c}, \frac{1}{\eta}\left(\frac{1}{\mu} - \frac{1}{\gamma}\right)\right\}$$

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

Steady-State Performance

- Little's law: average downloading time

$$T = \frac{1}{\theta + \beta}, \text{ where } \frac{1}{\beta} = \max\left\{\frac{1}{c}, \frac{1}{\eta}\left(\frac{1}{\mu} - \frac{1}{\gamma}\right)\right\}$$

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

Steady-State Performance

- Little's law: average downloading time

$$T = \frac{1}{\theta + \beta}, \text{ where } \frac{1}{\beta} = \max\left\{\frac{1}{c}, \frac{1}{\eta}\left(\frac{1}{\mu} - \frac{1}{\gamma}\right)\right\}$$

- Scalability: T is not a function of λ, the request arrival rate
- When the seed departure rate γ increases, T increases
- Even if $c \gg \mu$, the downloading bandwidth c may still be the bottleneck (e.g. if $\gamma < \mu$)

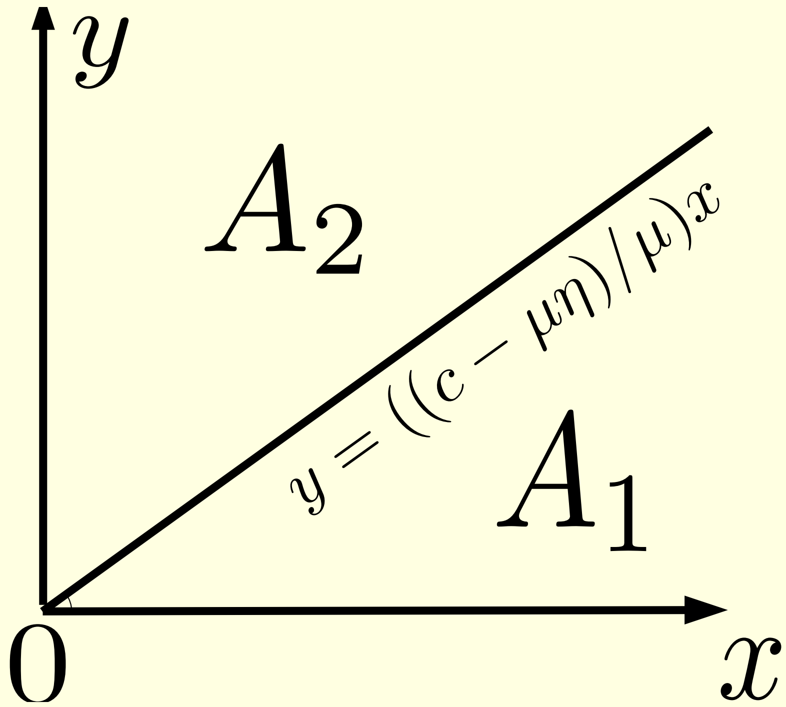
Steady-State Performance

- Little's law: average downloading time

$$T = \frac{1}{\theta + \beta}, \text{ where } \frac{1}{\beta} = \max\left\{\frac{1}{c}, \frac{1}{\eta}\left(\frac{1}{\mu} - \frac{1}{\gamma}\right)\right\}$$

- Scalability: T is not a function of λ, the request arrival rate
- When the seed departure rate γ increases, T increases
- Even if $c \gg \mu$, the downloading bandwidth c may still be the bottleneck (e.g. if $\gamma < \mu$)
- Prior work assumes $c = \infty$ (motivated by the asymmetry in cable modem and DSL rates): doesn't capture the above effect

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_2 is a stable matrix, but A_1 may not be a stable matrix
- However, the system is globally stable

Characterizing Variability

- How 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 $\hat{\mathbf{X}}(t) = (\hat{x}(t), \hat{y}(t))^T$ 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

- $$\mathbf{B} = \left[\begin{array}{cccc} 1 & -\sqrt{\rho} & -\sqrt{(1-\rho)} & 0 \\ 0 & 0 & \sqrt{(1-\rho)} & -\sqrt{(1-\rho)} \end{array} \right],$$
  where ρ is a constant depending on θ, c, μ, γ, and η.
- In steady-state, the number of seeds and downloaders is Gaussian with covariance $\mathbf{\Sigma}$:
  $$\mathbf{A\Sigma} + \mathbf{\Sigma A}^T + \mathbf{BB}^T = 0.$$

Peer Selection Algorithm

- Assumptions:
  - Each peer has the global information of uploading rates of other peers.
  - No downloading bandwidth constraints, all peers are fully connected and have demands from each other.
- Peer i selects n_u other peers to upload, which give peer i the best download rates
- With global information, the peer selection can be done in a systematic way

Peer Strategy

- In BitTorrent, a peer i can choose its uploading bandwidth up to a maximum of the physical uploading bandwidth p_i.
- $d_i(\mu_i, \mu_{-i})$: the download rate of peer i when its uploading bandwidth is μ_i and the uploading bandwidth of other peers is μ_{-i}
- Peer i try to choose μ_i such that

$$\mu_i = \min\{\tilde{\mu}_i | d_i(\tilde{\mu}_i, \mu_{-i}) = d_i(p_i, \mu_{-i})\}$$

Nash Equilibrium Point

- Given the peer selection algorithm (game rules), we can now study the system as a non-cooperative game. A Nash equilibrium for our problem is a set of uploading rates $\{\bar{\mu}_i\}$ such that

$$\bar{\mu}_i = \min\{\tilde{\mu}_i | d_i(\tilde{\mu}_i, \bar{\mu}_{-i}) = d_i(p_i, \bar{\mu}_{-i})\}$$

- For a general network setting, there may be no Nash equilibrium point exists.

Nash Equilibrium Point

- We consider a network with a finite number of groups of peers. In group j, all peers have the same physical uploading bandwidth p_j.
- Let g_j be the set of peers in group j and $||g_j||$ be the number of peers in group j.
- **Proposition 1** *If $n_u \geq 2$ and the number of peers in a group $||g_j|| > n_u + 1$ for all groups, there exists a Nash equilibrium point for the system, in which $\bar{\mu}_i = p_j$ if peer $i \in g_j$. Moreover, with any initial setting of $\{\mu_i^0\}$, the system converges to the Nash equilibrium point $\{\bar{\mu}_i\}$.*

Simulation Result

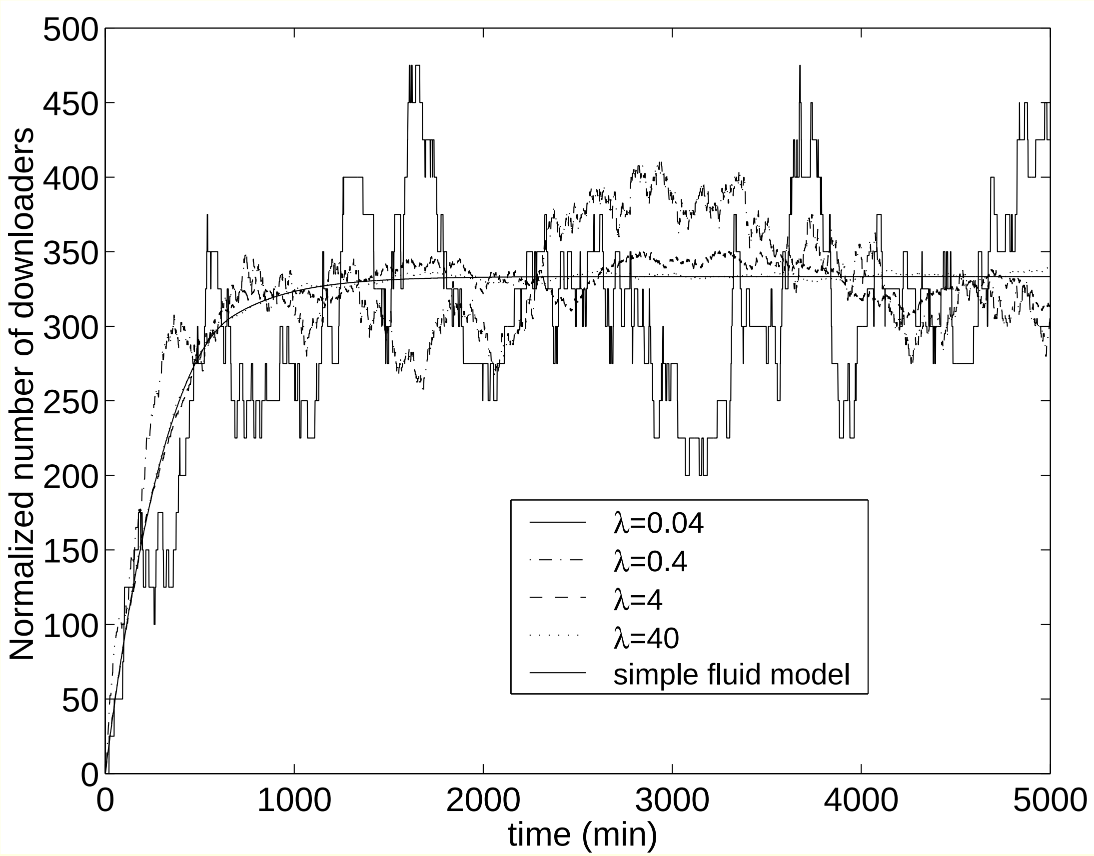


Figure 1: The evolution of the number of downloaders as a function of time ($\mu = 0.00125$, $c = 0.002$, $\theta = 0.001$, $\gamma = 0.005$)

Simulation Result

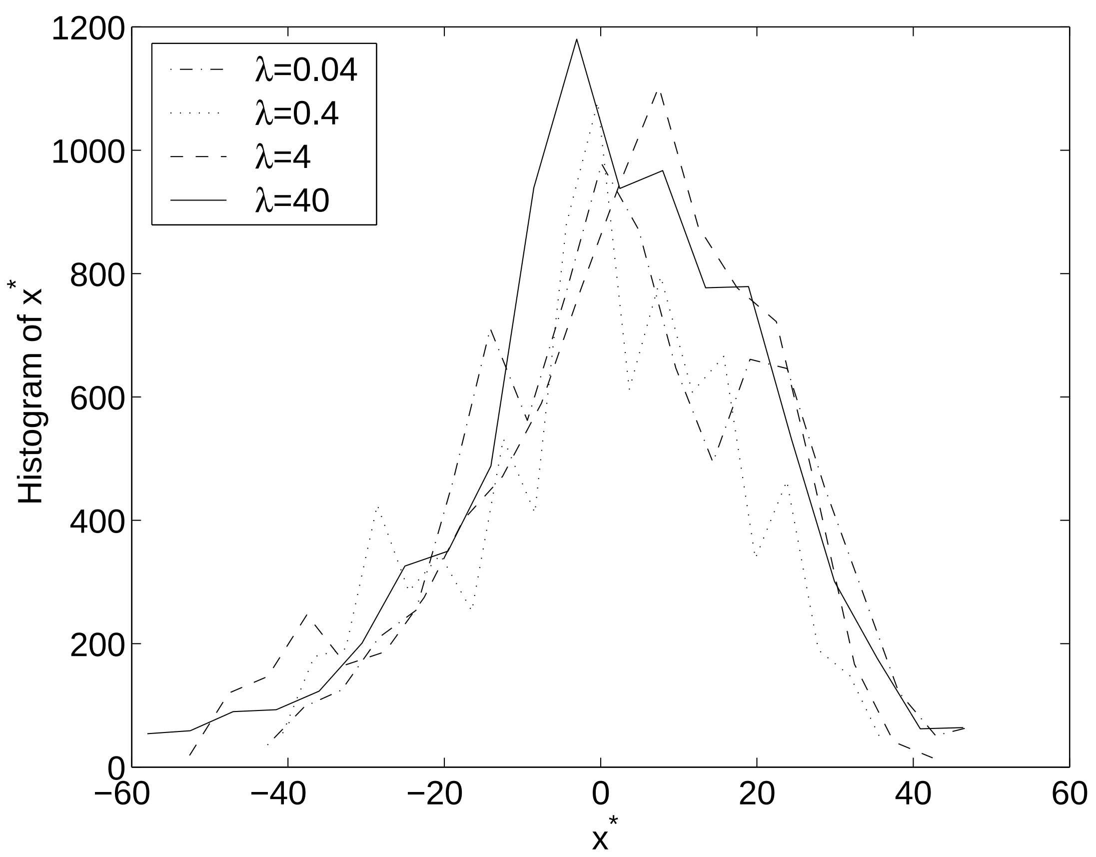


Figure 2: Histogram of the variation of the number of downloaders around the fluid model ($\mu = 0.00125$, $c = 0.002$, $\theta = 0.001$, $\gamma = 0.005$)

Experimental Result

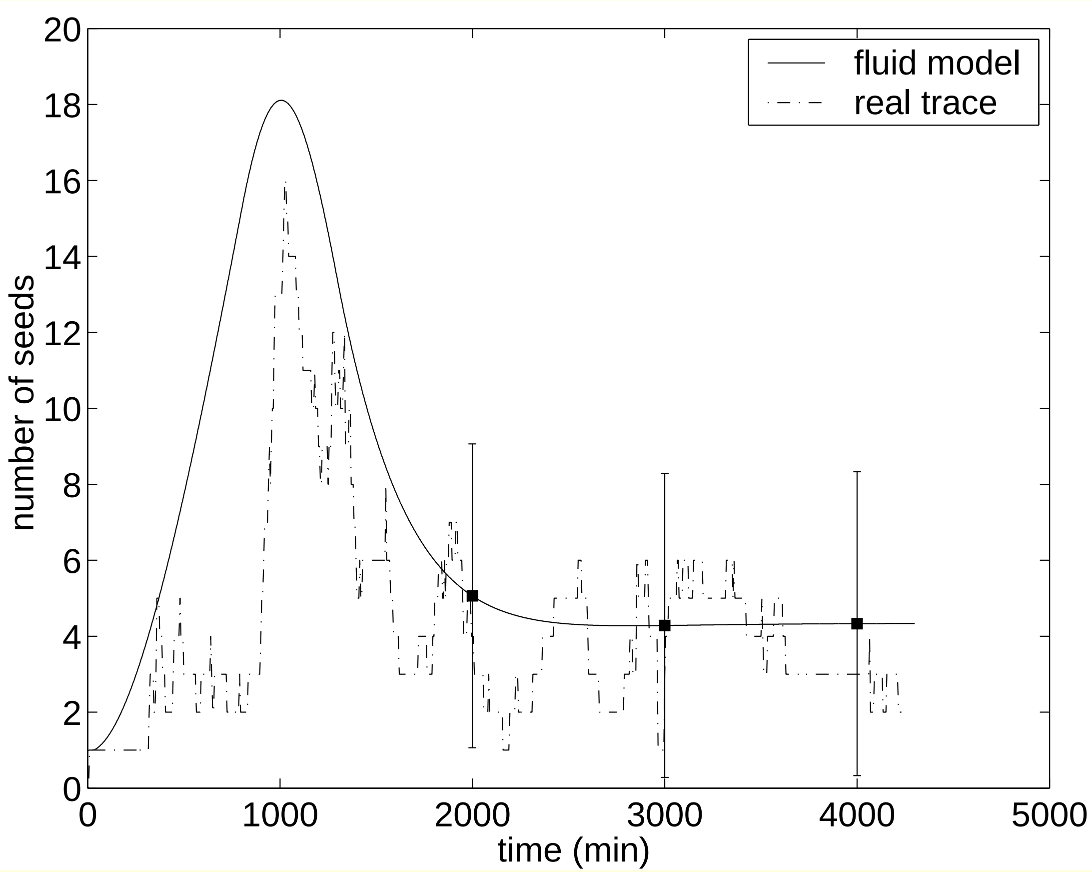


Figure 3: The evolution of the number of seeds as a function of time (real trace)

Conclusions

- Presented a simple fluid model and a game-theoretic model for BitTorrent-like networks
- Studied the steady-state network performance and stability
- Obtained insight into the effect of different parameters on network performance
- Studied the effect of the built-in incentive mechanism of BitTorrent on preventing free-riding