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

Presented by: 鍾毓安 許詠翔 張君瑋
{b01902040, b01902029, b01902051}@ntu.edu.tw

2016.6.15

Outline

- Introduction to P2P network
- Introduction to BitTorrent
- A simple fluid model
- The incentive mechanism
- Experiments
- Conclusions

Outline

- **Introduction to P2P network**
- Introduction to BitTorrent
- A simple fluid model
- The incentive mechanism
- Experiments
- Conclusions

Peer-to-peer network

- Client-Server network vs. Peer-to-peer (P2P) network
- Peers in a P2P network operate as both servers and clients.
- P2P traffic is starting to dominate the bandwidth of the Internet.
- Among P2P applications, file sharing is the most popular one.
- P2P file sharing system is scalable.
- BitTorrent, one of the most popular P2P applications, will be studied in this paper.

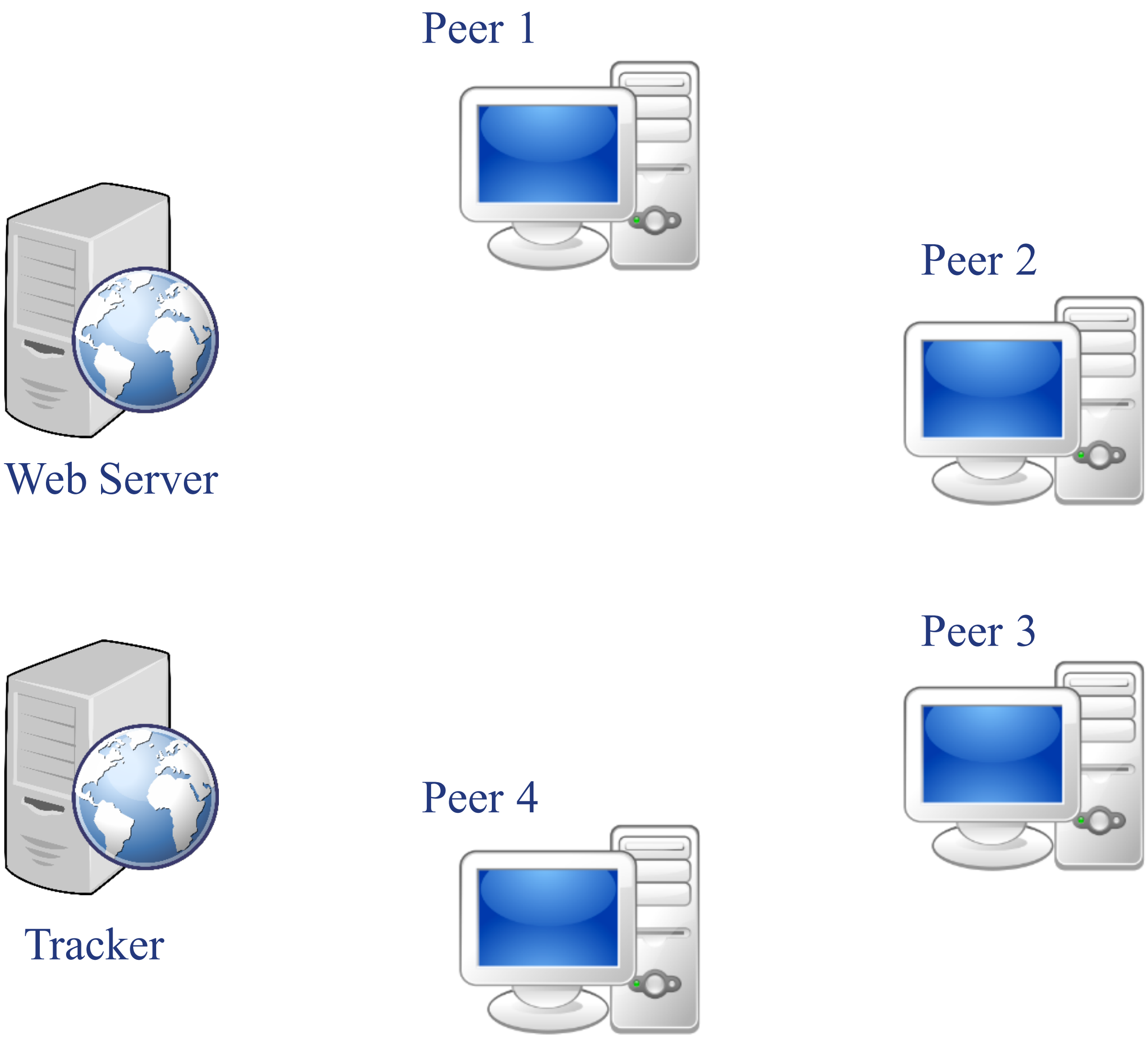
Outline

- Introduction to P2P network
- **Introduction to BitTorrent**
- A simple fluid model
- The incentive mechanism
- Experiments
- Conclusions

BitTorrent

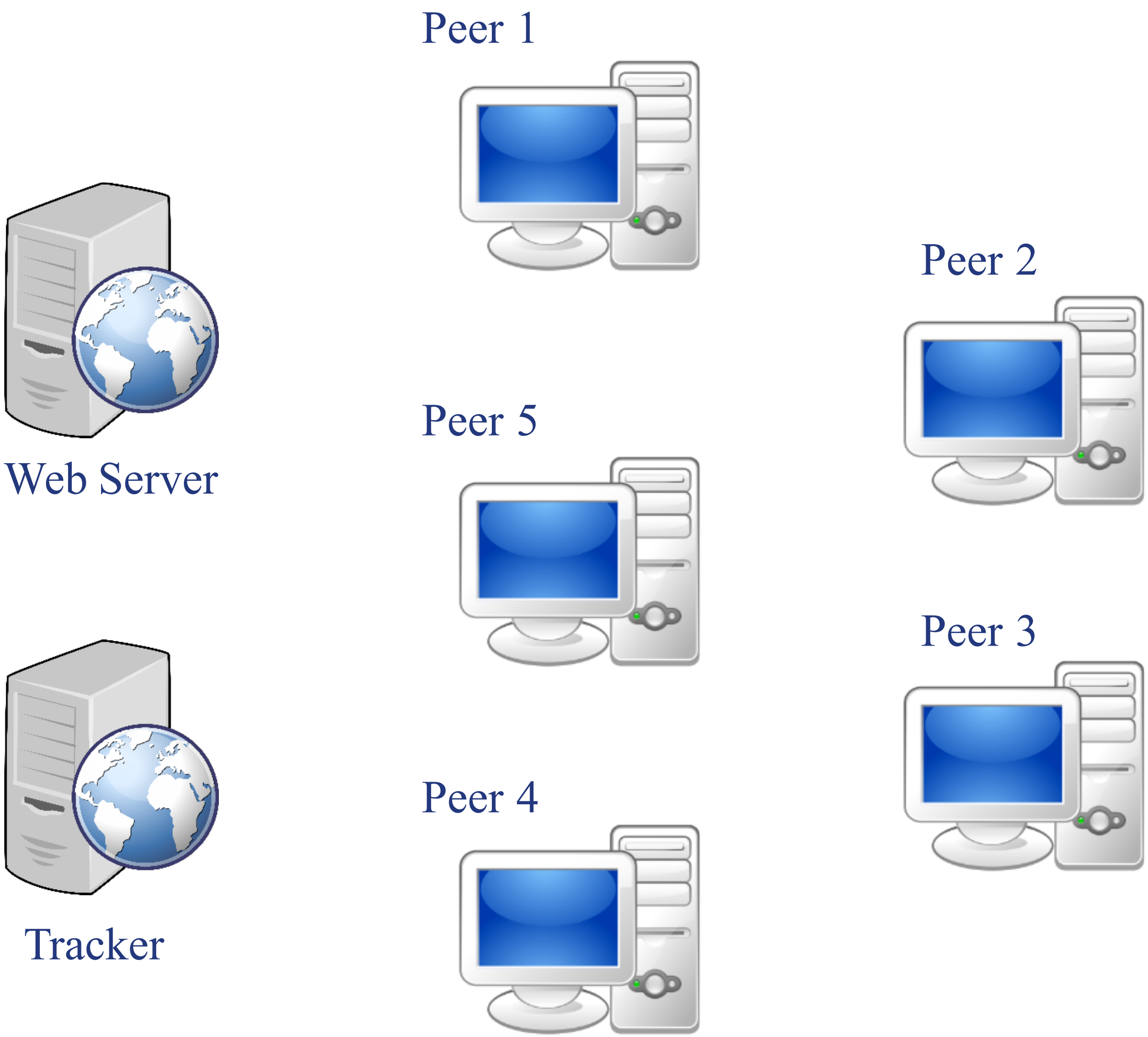
- Some concepts of BitTorrent:
  - A server, called the seed, has the entire file of interest.
  - The file is separated into many pieces, where each piece is of 256 KB.
  - As peers arrive, they download random pieces of the file from the seed.
  - Now there are many peers in the system with different or overlapping pieces of the file.
  - The peers can act as servers even they only have parts of the file.
  - Peers can then download from each other!
- In a traditional client-server system (such as FTP), clients can only download from a single server.

How do BitTorrent work?



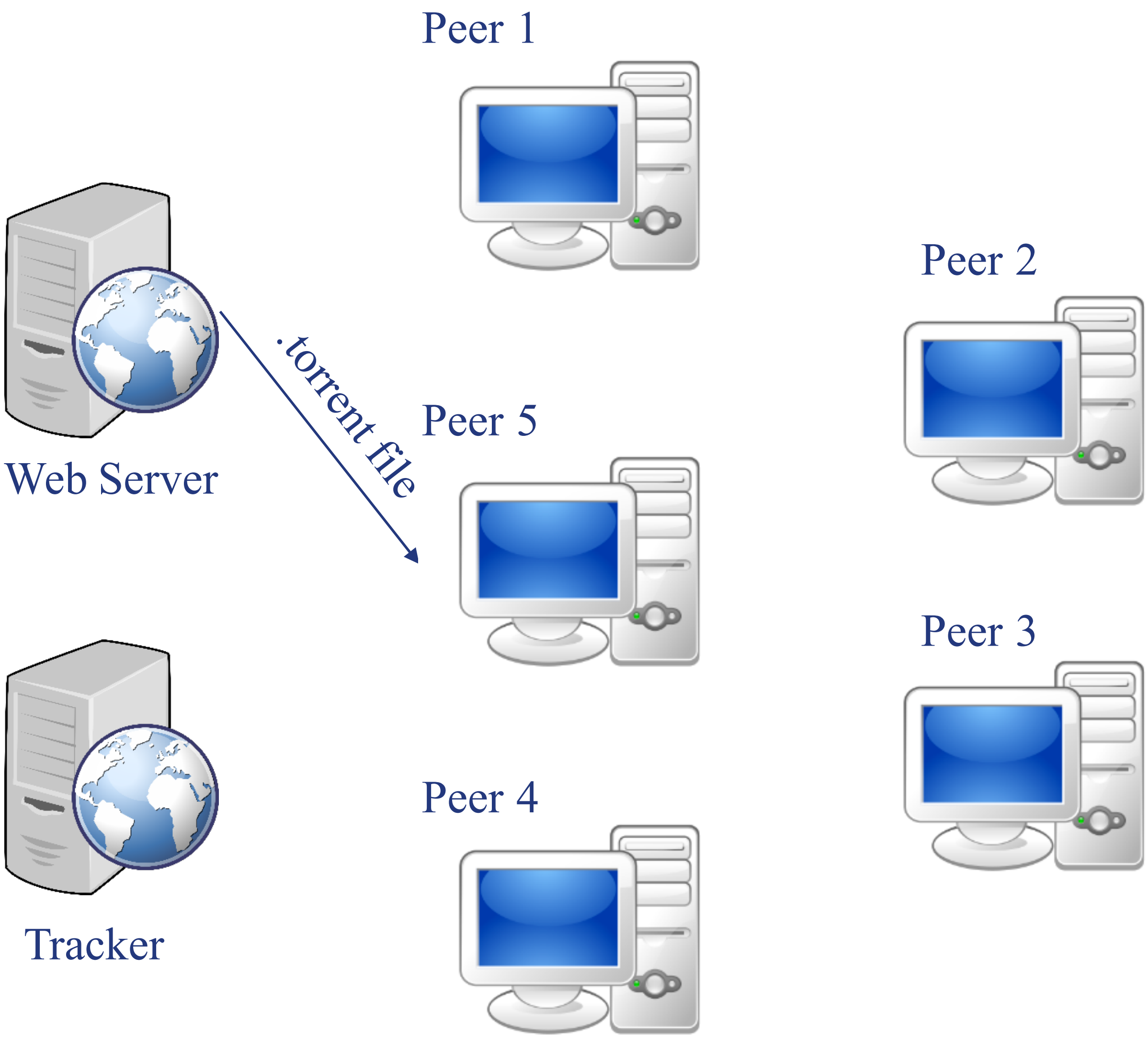
- In this network, there are 4 peers downloading pieces of the file of interest.

How do BitTorrent work?



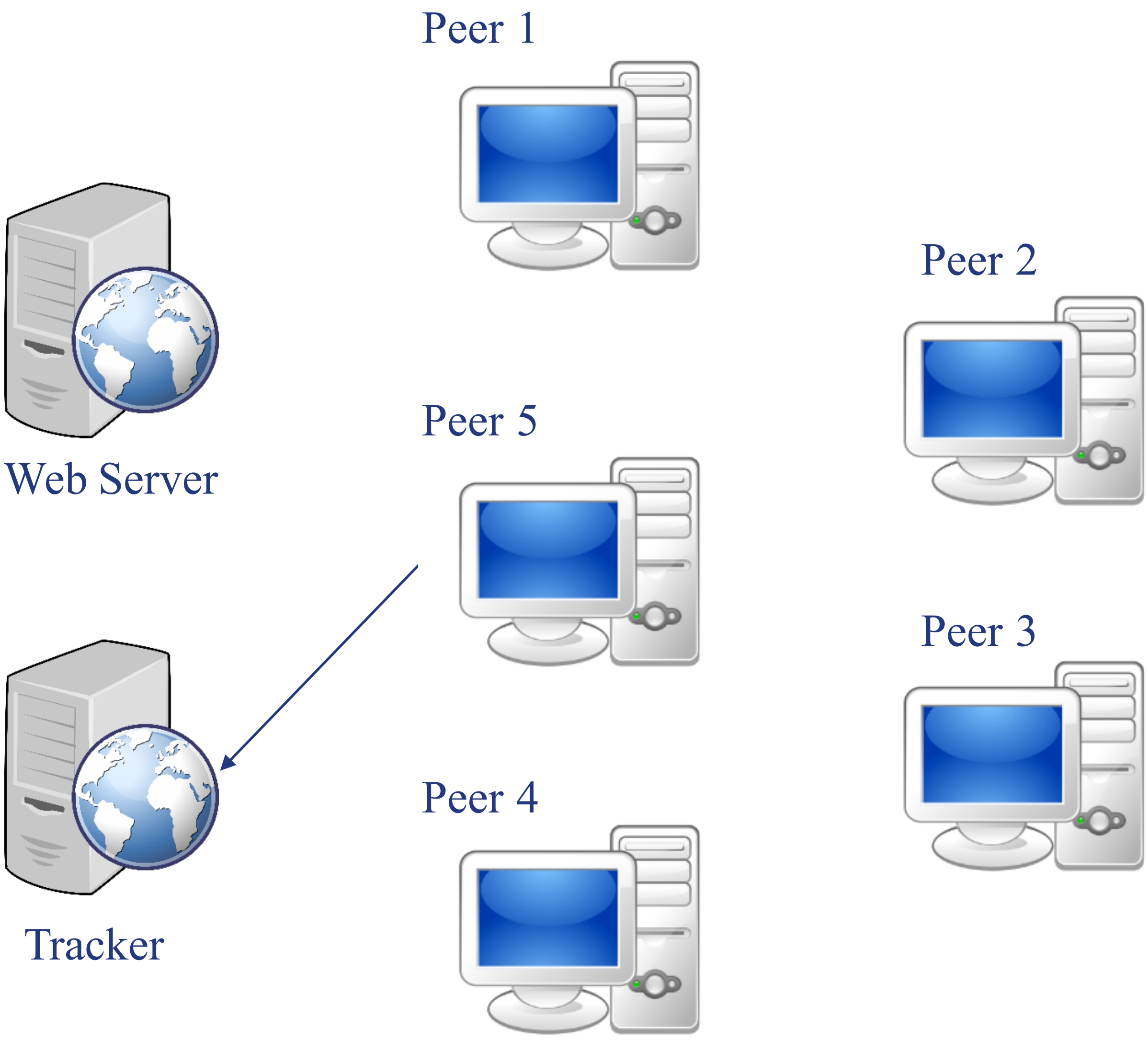
- In this network, there are 4 peers downloading pieces of the file of interest.
- Peer 5 joins the network to download the file.

How do BitTorrent work?



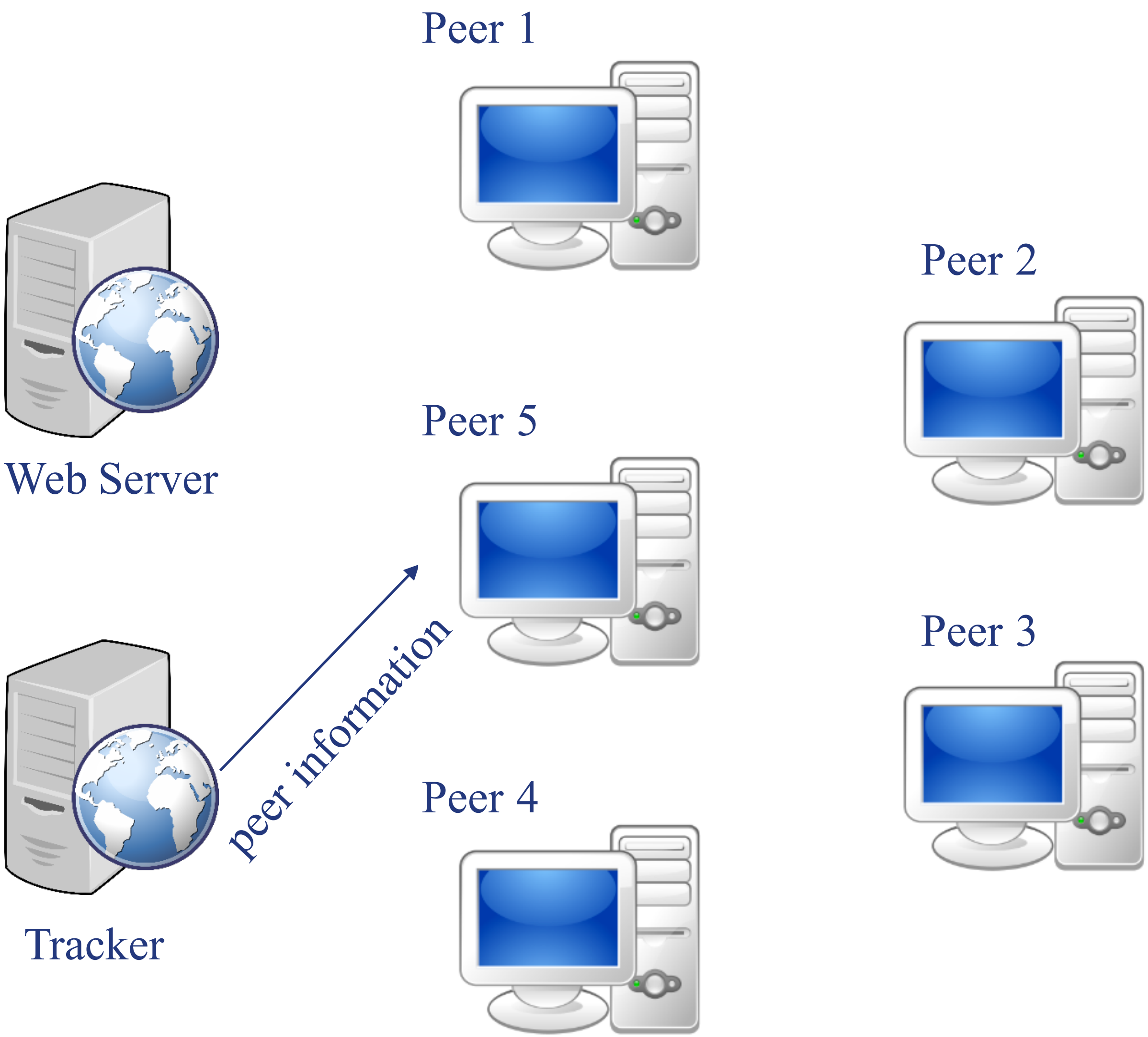
- In this network, there are 4 peers downloading pieces of the file of interest.
- Peer 5 joins the network to download the file.
- It first downloads a .torrent file from the web server.

How do BitTorrent work?



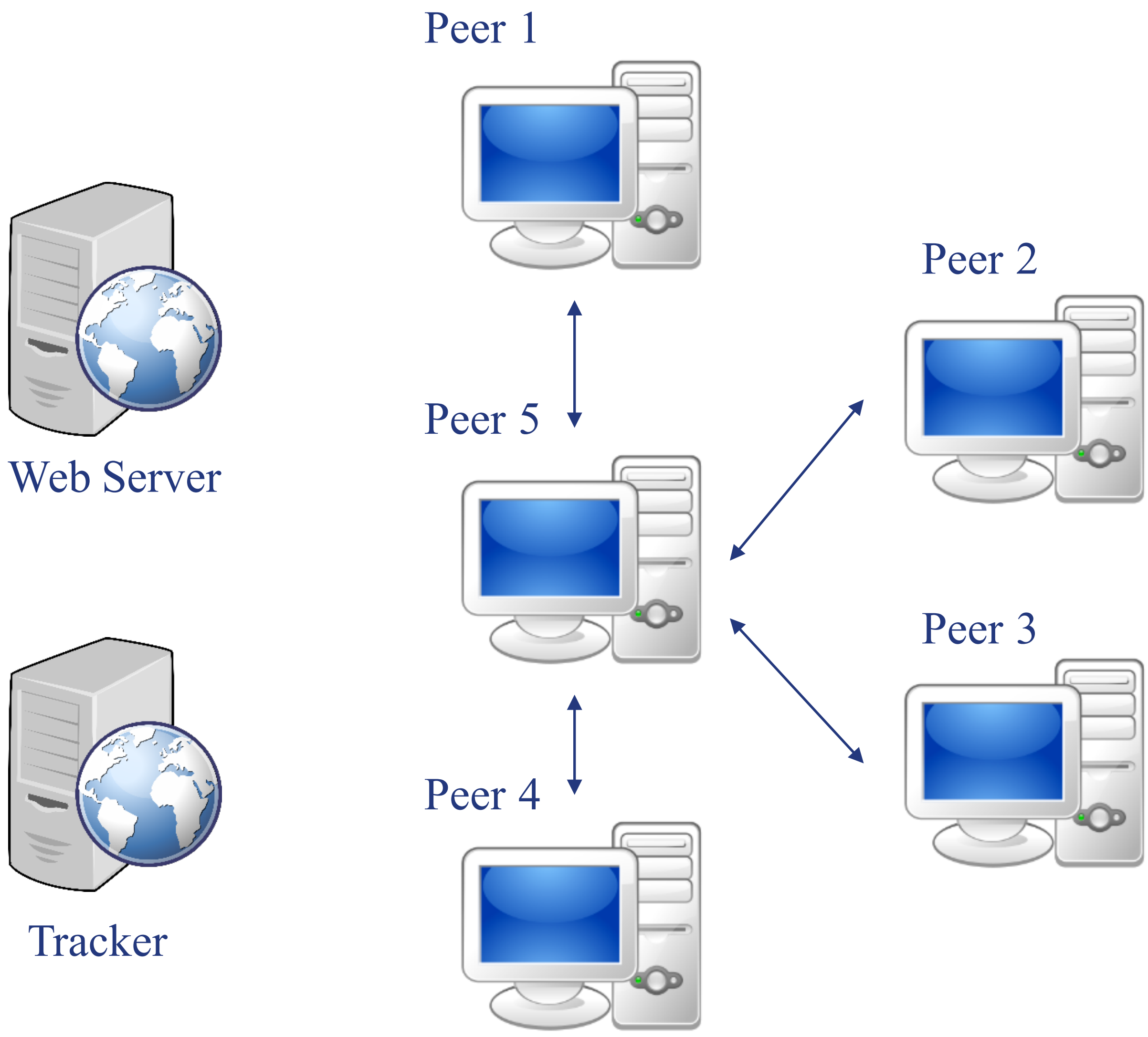
- In this network, there are 4 peers downloading pieces of the file of interest.
- Peer 5 joins the network to download the file.
- It first downloads a .torrent file from the web server.
- Next, it connects to the tracker.

How do BitTorrent work?



- In this network, there are 4 peers downloading pieces of the file of interest.
- Peer 5 joins the network to download the file.
- It first downloads a .torrent file from the web server.
- Next, it connects to the tracker.
- The tracker returns the peer information.

How do BitTorrent work?

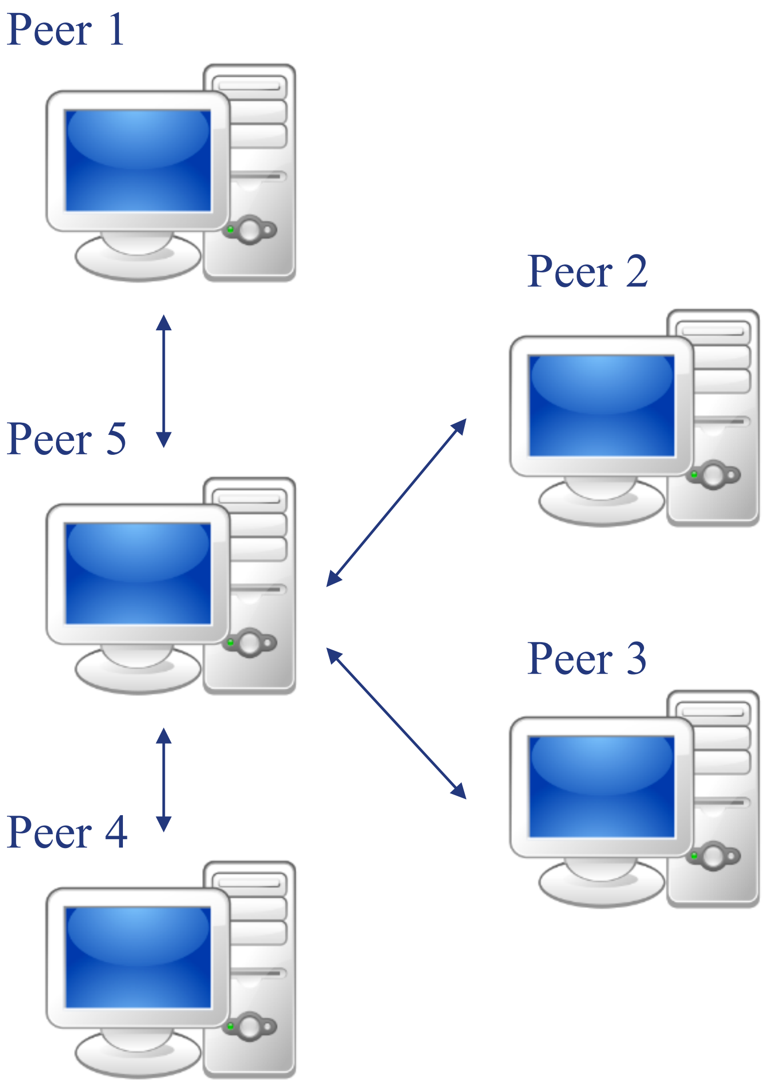


- In this network, there are 4 peers downloading pieces of the file of interest.
- Peer 5 joins the network to download the file.
- It first downloads a .torrent file from the web server.
- Next, it connects to the tracker.
- The tracker returns the peer information.
- Peer 5 connects to 4 other peers and begins downloading and exchanging.

Some notes behind the operation of BitTorrent

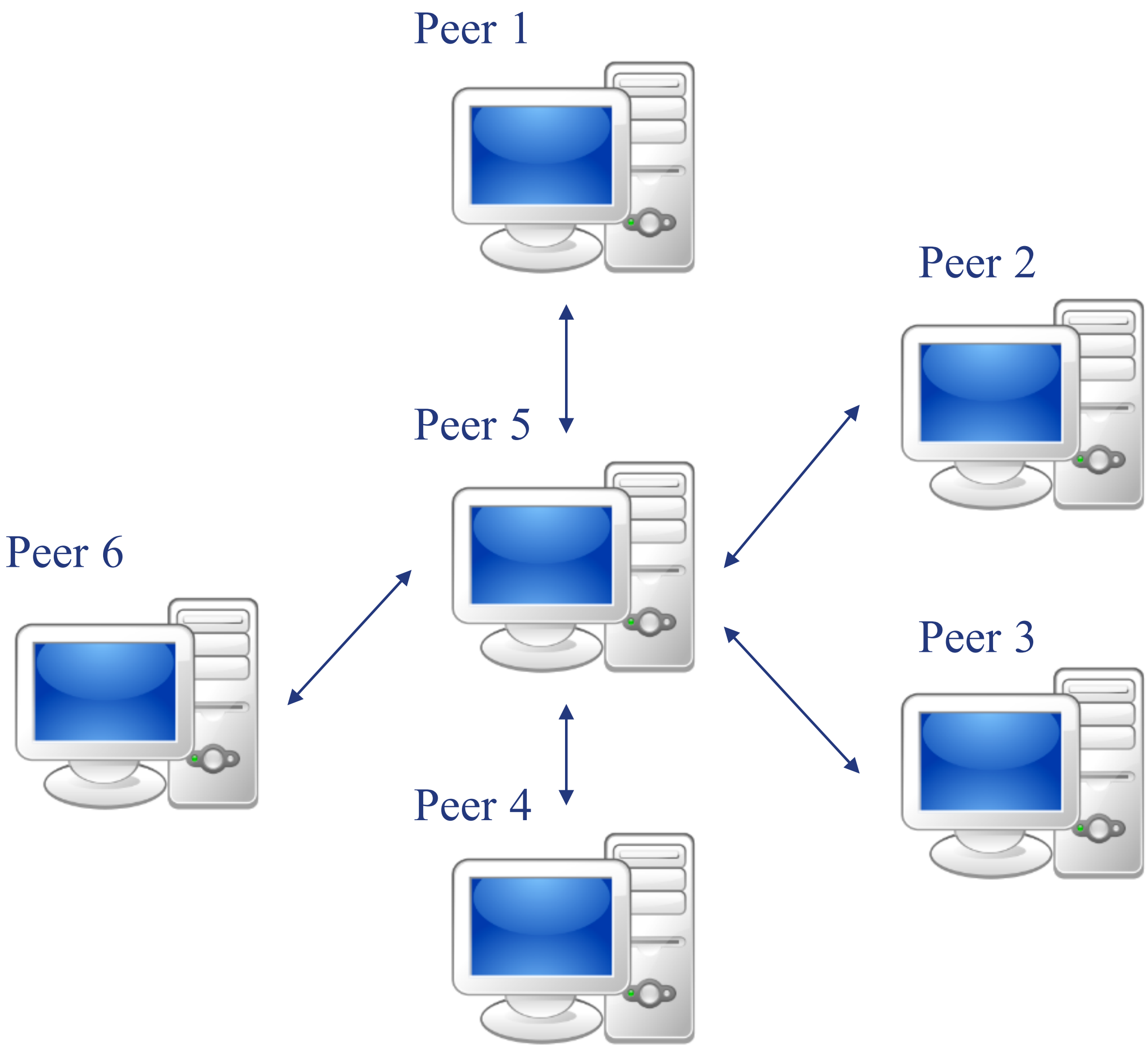
- Downloaders are peers who only have a part (or none) of the file while seeds are peers who have all the pieces (the complete file) but stay in the system to allow other peers to download from them.
- Each peer is allowed to upload only to a fixed number (by default 4) of other peers at a given time.
- What is and why unchocking? To prevent free-riding, the most important concern in BitTorrent.
- The mechanism of Optimistic Unchocking → to search for potential peer with high uploading/downloading speed.

How do Optimistic Unchocking work



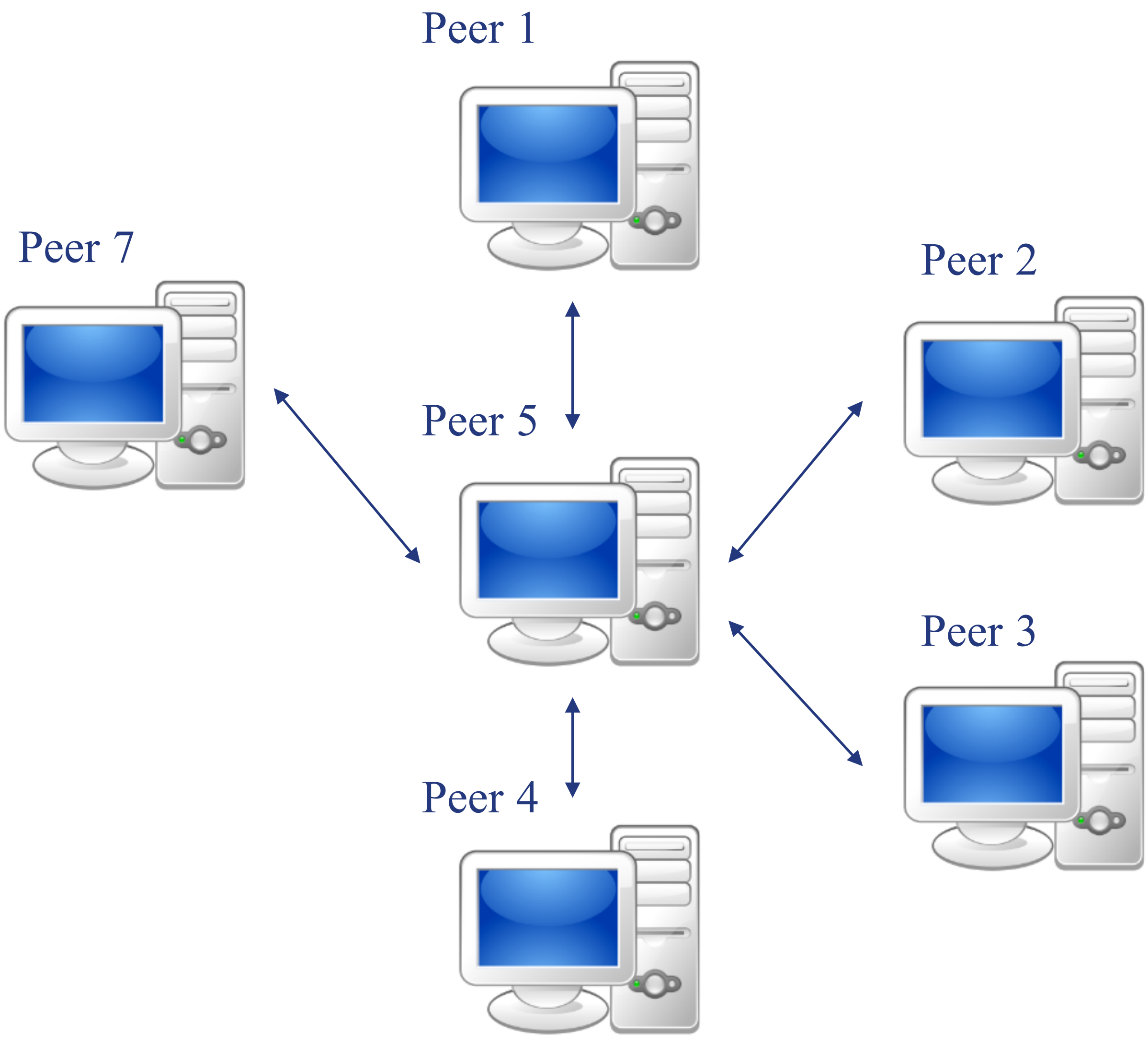
- Peer 5 is now connecting with peer 1 ~ 4 for downloading / uploading the file of interest.

How do Optimistic Unchocking work



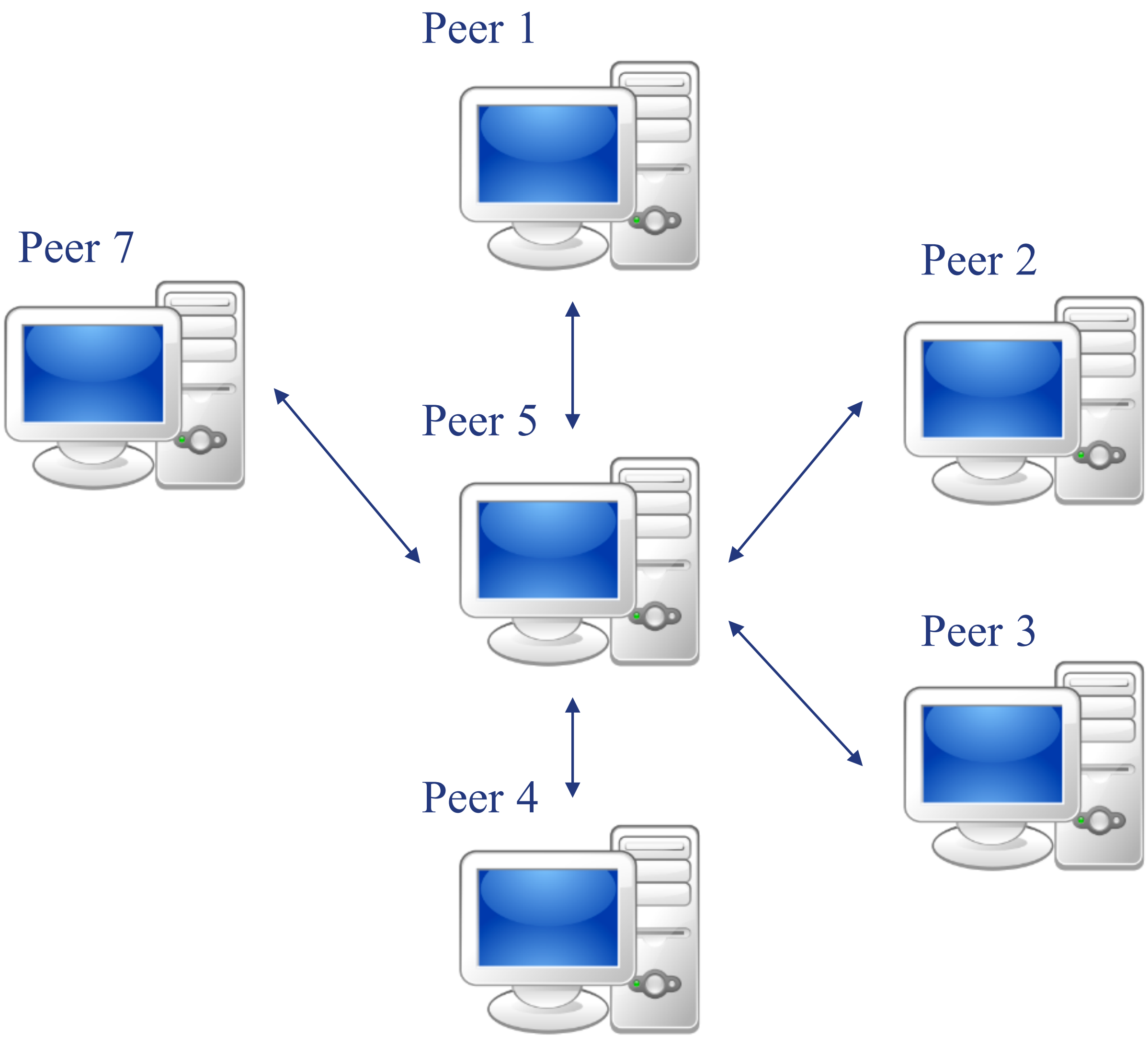
- Peer 5 is now connecting with peer 1 ~ 4 for downloading / uploading the file of interest.
- After 30 seconds, peer 5 randomly connects to another peer, say peer 6, and finds that the transmission speed of peer 6 does not exceed any of the peers that peer 5 has already connected to.

How do Optimistic Unchocking work



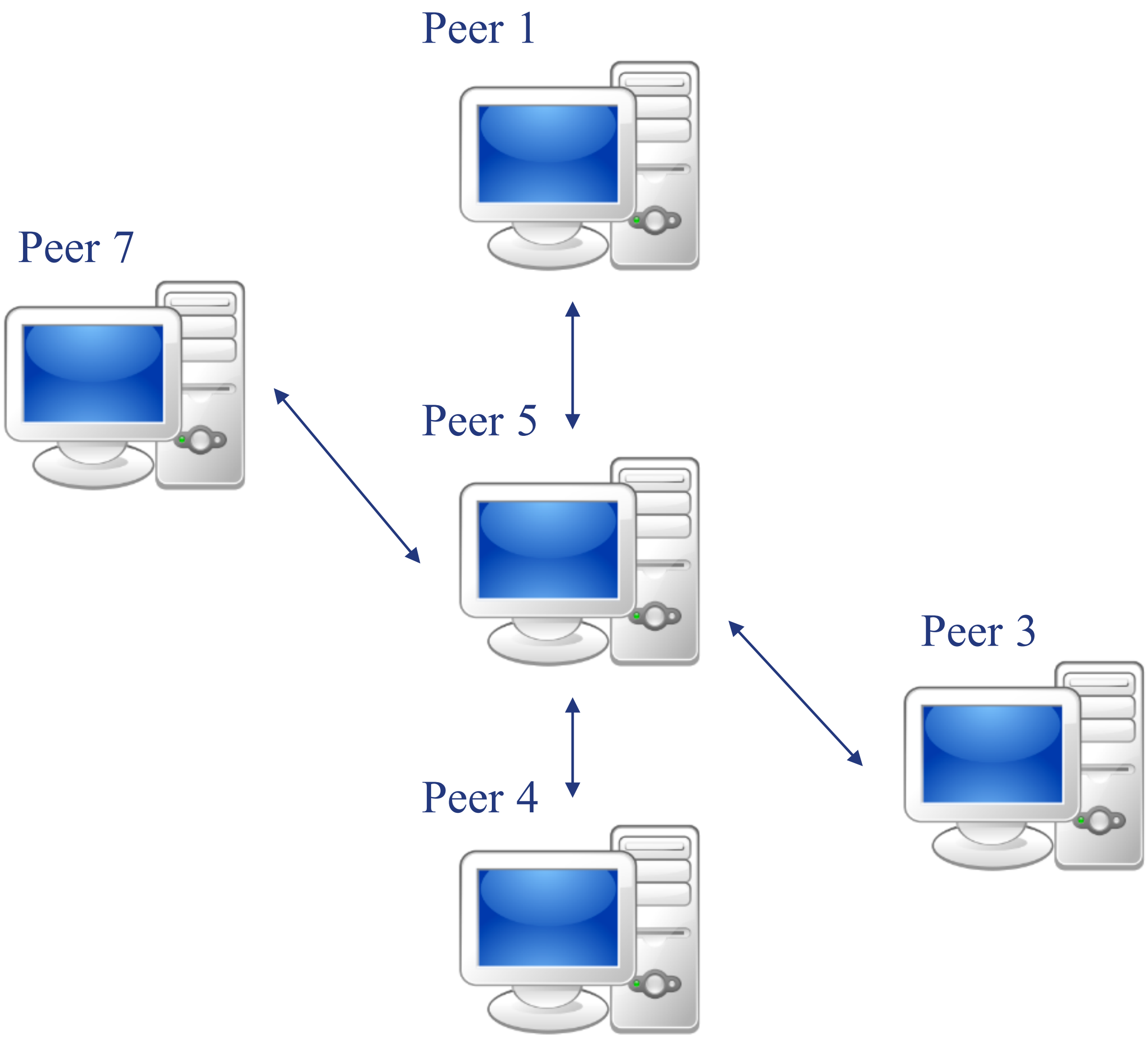
- Peer 5 is now connecting with peer 1 ~ 4 for downloading / uploading the file of interest.
- After 30 seconds, peer 5 randomly connects to another peer, say peer 6, and finds that the transmission speed of peer 6 does not exceed any of the peers that peer 5 has already connected to.
- Therefore, after another 30 seconds, peer 5 disconnects with peer 6, and connects with another peer, say peer 7.

How do Optimistic Unchocking work



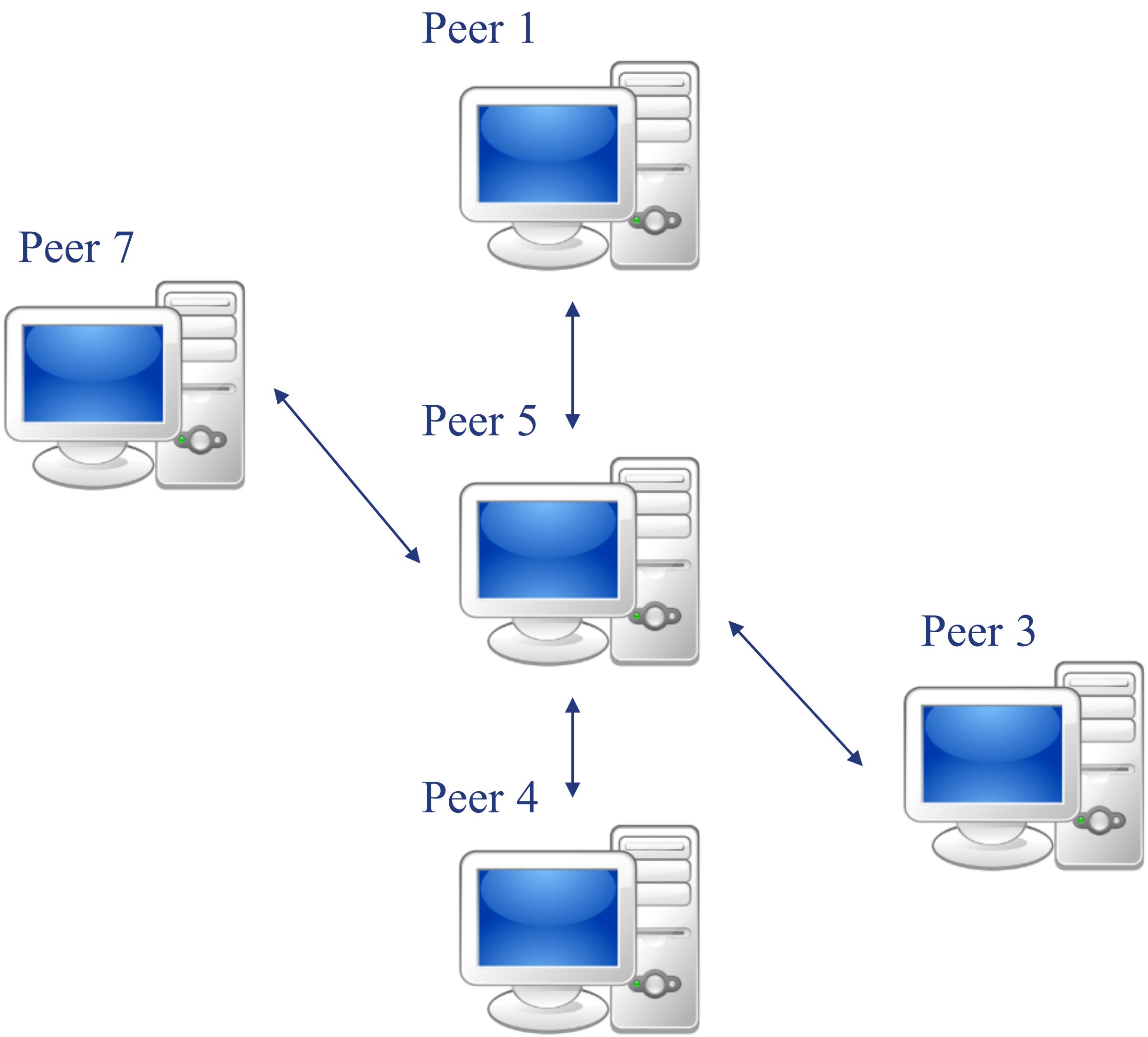
- Peer 5 finds that after it connects to peer 7, peer 2 becomes the one that has the slowest transmission speed.

How do Optimistic Unchocking work



- Peer 5 finds that after it connects to peer 7, peer 2 becomes the one that has the slowest transmission speed.
- Therefore, peer 5 disconnects with peer 2 after another 30 seconds.

How do Optimistic Unchocking work



- Peer 5 finds that after it connects to peer 7, peer 2 becomes the one that has the slowest transmission speed.
- Therefore, peer 5 disconnects with peer 2 after another 30 seconds.
- The same process is repeated again and again …

Some Issues of BitTorrent

- **Peer Evolution**: how the number of peers evolves as a function of the request arrival rate, the peer departure rate, the uploading/downloading bandwidth of each peer, etc.
- **Scalability**: as the size of the network (number of peers) grows, the network performance (average file downloading time) should preferably to actually improve.
- **File Sharing Efficiency**: as peers often have different uploading/downloading bandwidths, it is important to design a good file-sharing protocol such that the uploading/downloading bandwidths are fully utilized.
- **Incentives to prevent free-riding**: need to a mechanism to deter peers from free-riding (download from other peers while not uploading to others).

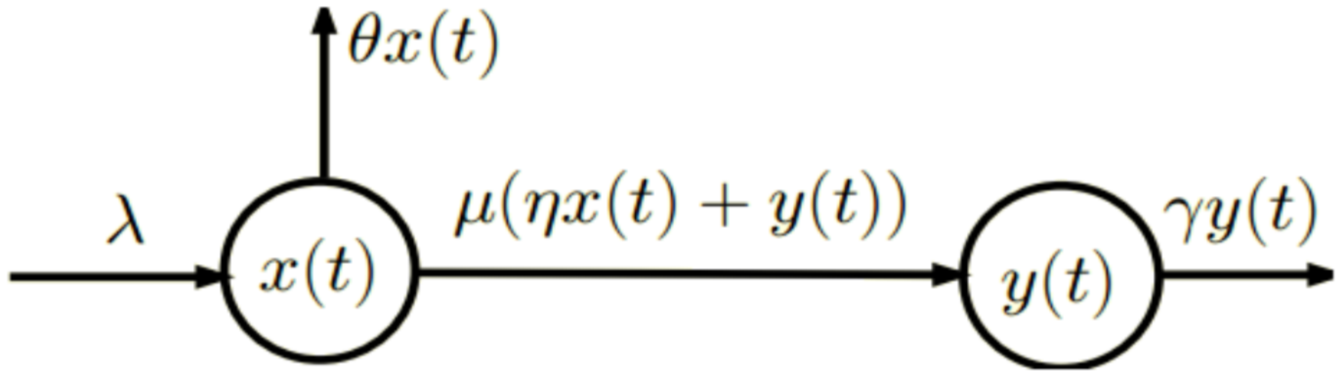
Outline

- Introduction to P2P network
- Introduction to BitTorrent
- **A simple fluid model**
- The incentive mechanism
- Experiments
- Conclusions

Fluid Model

- $x(t)$ - number of downloaders in the system at time t.
- $y(t)$ - number of seeds in the system at time t.
- λ - the arrival rate of new requests (Poisson process)
- μ - the uploading bandwidth of a given peer (normalized by file size)
- C - the downloading bandwidth of a given peer (normalized by file size).
- θ - the rate at which downloaders abort the download (Poisson process)
- γ- the rate at which seeds leave the system (Poisson process)
- η - indicates the effectiveness of the file sharing , takes values in [0, 1].

Fluid Model



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

Steady-State Performance

- To study the system in steady-state, we let

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = \frac{\mathrm{d}y(t)}{\mathrm{d}t} = 0 \qquad \begin{aligned} 0 &= \lambda - \theta\bar{x} - \min\{c\bar{x}, \mu(\eta\bar{x} + \bar{y})\}, \\ 0 &= \min\{c\bar{x}, \mu(\eta\bar{x} + \bar{y})\} - \gamma y(t), \end{aligned}$$

- X¯ , Y ¯ are equilibrium values

$$\begin{aligned} \bar{x} &= \frac{\lambda}{\beta(1 + \frac{\theta}{\beta})} \\ \bar{y} &= \frac{\lambda}{\gamma(1 + \frac{\theta}{\beta})}. \end{aligned}$$

- β is determined by the bottleneck between download rate and upload rate

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

Steady-State Performance(cont.)

- Little's law: Average download time

$$\frac{\lambda - \theta\bar{x}}{\lambda}\bar{x} = (\lambda - \theta\bar{x})T, \qquad T = \frac{1}{\theta + \beta}.$$

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

- 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

Effectiveness of File Sharing

- For a given downloader i, we assume that it is connected to k other downloaders.
  - $\eta = 1 - P$ (downloader i has no piece that the connected peers need) .
- We assume that the piece distributions between different peers are independent and identical. Then
  - $\eta = 1 - P$ (downloader j needs no piece from downloader i $)^k$,
  - 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(cont.)

- 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(cont.)

$$\eta \approx 1 - \left(\frac{\log N}{N} \right)^{k}.$$

- 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 $k = 0$, $\eta = 0$.

Local Stability

- eigenvalues of A1 and A2 have negative real parts – system is stable.

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

$$\mathbf{A}_1 = \begin{bmatrix} -(\mu\eta + \theta) & -\mu \\ \mu\eta & -(\gamma - \mu) \end{bmatrix}.$$

$$\begin{aligned}\frac{\mathrm{d}x(t)}{\mathrm{d}t} &= \lambda - \theta x(t) - cx(t) \\ \frac{\mathrm{d}y(t)}{\mathrm{d}t} &= cx(t) - \gamma y(t).\end{aligned}$$

$$\mathbf{A}_2 = \begin{bmatrix} -(\theta + c) & 0 \\ c & -\gamma \end{bmatrix}.$$

Outline

- Introduction to P2P network
- Introduction to BitTorrent
- A simple fluid model
- **The incentive mechanism**
- Experiments
- Conclusions

Peer Selection Algorithm

- Peer i selects the n_u(default 4) peers that give it the best upload to download to
- Assumptions : Global information , No optimistic unchocking ,No download limit
- Sort the peers according to their uploading bandwidth (physical or determined), first peer has the highest uploading bandwidth.
- peer i choosing peers to upload at step i.
- N total number of peers
- u_i: the uploading bandwidth of peer i.

Peer Selection Algorithm(cont.)

1. If peer i is selected by peer j $(j < i)$, then i selects j. For any peer k $(k \geq i)$, let n_k^i be the number of peers that have selected peer k prior to step i.

2. If $n_i^i < n_u$ and $n_u - n_i^i \leq N - i$, peer i selects $n_u - n_i^i$ peers from the set $\{k|k > i\}$ using the following set of rules to prioritize a peer, say $k1$, over another peer $k2$:

   (a) If $\mu_{k1} > \mu_{k2}$, select $k1$.

   (b) If $\mu_{k1} = \mu_{k2}$ and $n_{k1}^i < n_{k2}^i$, select $k1$.

   (c) If $\mu_{k1} = \mu_{k2}$, $n_{k1}^i = n_{k2}^i$, and $k1 < k2$, select $k1$.

3. If $n_i^i < n_u$ and $n_u - n_i^i > N - i$, peer i selects all peers in $\{k|k > i\}$ and also randomly selects $(n_u - n_i^i) - (N - i)$ peers from the peers that i has not selected yet.

Peer Selection Algorithm(cont.)

- Lemma 1. With the peer selection algorithm, when peer i selects uploading peers, $n_i^i \le n_u$ and for any k2 > k1 ≥ I , $n_{k2}^i \le n_{k1}^i \le n_u$
- Lemma 2. Suppose that peers i, i+1, · · · , j have the same uploading bandwidth μ.If j –i+1 > n_u ≥ 2, then for any k > j, we have

  1. d(i) ≥ d(i)+1 ≥ · · · ≥ d(j) ≥ d(k),
  2. d(i) > d(k),
  3. d(μ) > d(k)

where

$$d_i = \frac{1}{n_u} \sum_{k \in D_i} \mu_k . \quad d(\mu) = \frac{1}{j-i+1} \sum_{k=i}^{j} d_k$$

Peer Strategy

- First maximizing download bandwidth , then minimizing upload bandwidth
- Then peer i *choose* u_i s.t.

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

P_i denote the physical upload rate

- Or more realistic

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

- Where ε is the difference between two rates that a peer can differentiate.

Peer Strategy(ex.)

- consider a small BitTorrent network with 6 peers
- The number of uploads $n_u = 4$ for all peers
- peers have different physical uploading bandwidth
- If d the minimum uploading bandwidth $\min\{p_i\} > 2\varepsilon$, there is no Nash equilibrium point for the system

Nash Equilibrium Point

- Divide the network to sub-groups . In each group j, all peers have the same physical uploading bandwidth p_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 $\{\bar{\mu}_i\}$.*

Outline

- Introduction to P2P network
- Introduction to BitTorrent
- A simple fluid model
- The incentive mechanism
- **Experiments**
- Conclusions

Optimistic Unchoking Recap

- Each peer uploads to other n_u (by default 4) peers which provide it with the best downloading rate.
- Under optimistic unchoking, each peer randomly selects a fifth peer to download from exploring other download rates.
- Then peer with the least downloading rate is dropped.
- Optimistic unchoking happens once every 30 seconds.
- However, such mechanism gives chances to free-riders!

Free-riding Recap

- Free-riding means that a peer does not contribute anything to the system, but attempts to obtain service (or downloading) from other peers.

- Lack of global knowledge (full downloading rates information) requires the use of optimistic unchoking.

- This allows for peers with no uploading bandwidth to get download bandwidth.

Free-riding Formulation

- Consider a group of peers that have the same uploading bandwidth μ and number of peers in the group N.
- Assume that each peer has n_u uploads and 1 optimistic unchoking upload.
- By optimistic unchoking, peer *j* which is a free-rider get selected 1/(N- n_u) of the time by any other peer.
- So, total average downloading rate of peer *j* will be

$$N \frac{1}{N - n_u} \frac{\mu}{n_u + 1} \approx \frac{\mu}{n_u + 1}$$

when N is large.

- The free-riders problem is not yet solved in BitTorrent.

(in BitTorrent, n_u= 4 so free-rider gets 20% of the possible maximum downloading rate.)

Notations Recap

- $x(t)$: number of downloaders in the system at time t.
- $y(t)$: number of seeds in the system at time t.
- λ : the arrival rate of new request.
- μ : the uploading bandwidth of a given peer.
- c : the downloading bandwidth of a given peer.
- θ : the rate at which downloaders abort the system.
- γ : the rate at which seeds leaves the system.
- η : the effectiveness of the file sharing. η takes values in [0,1].

Experiment Setup

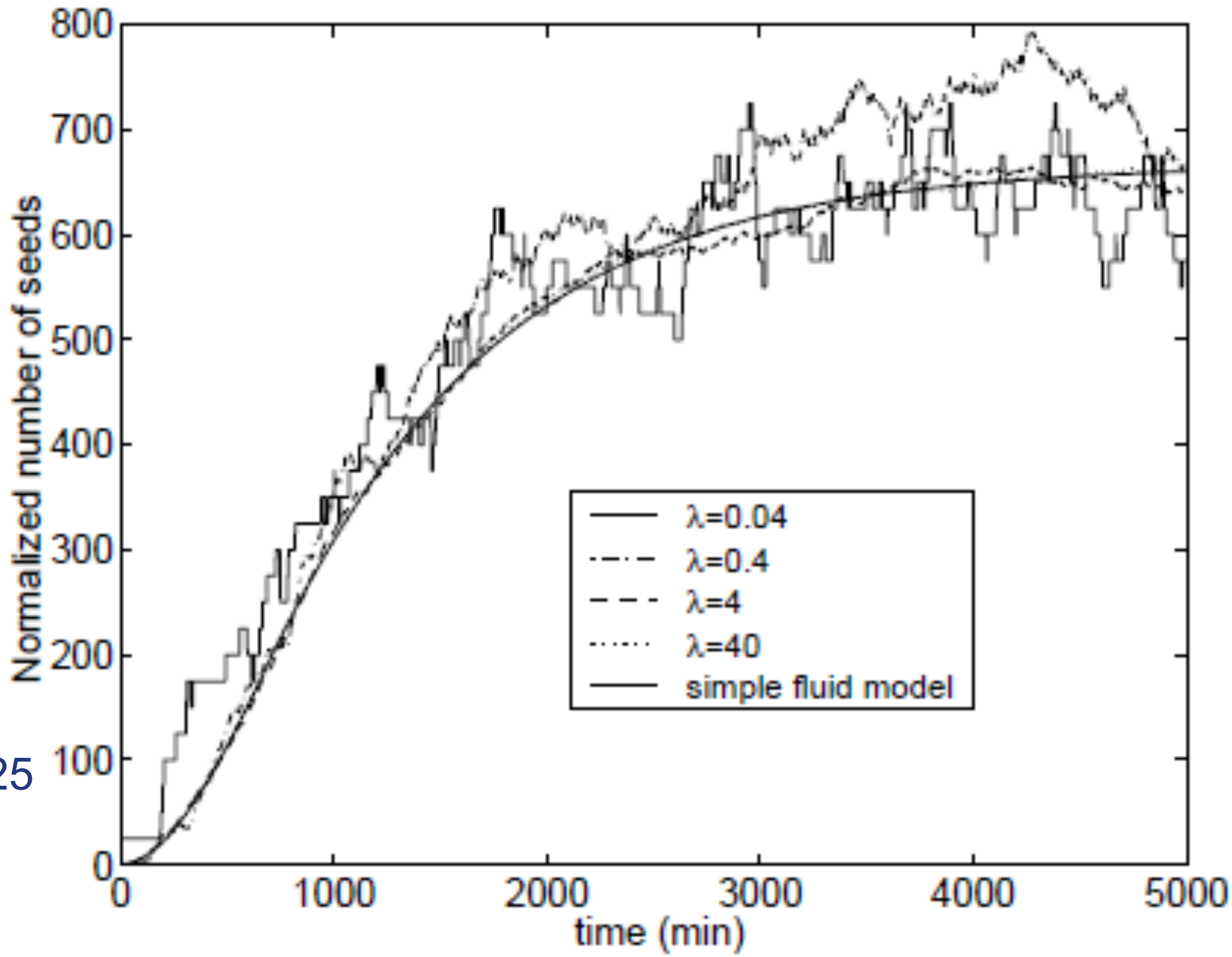
- Compare fluid model to:
  (1) Simulated BitTorrent network that is download-limited ($\gamma < \mu$)
    - Discrete-event simulation based on the Markov model
    - The arrival rate $\lambda \in \{0.04, 0.4, 4, 40\}$.
    - Normalize the number of seeds/downloaders by dividing by λ

  (2) Simulated BitTorrent network that is upload-limited (same setup with (1), but $\gamma > \mu$)

  (3) Real-world BitTorrent network
  → collect the log files of the BitTorrent tracker for a time period of around three days.

  Set $\eta = 1$ for all cases.

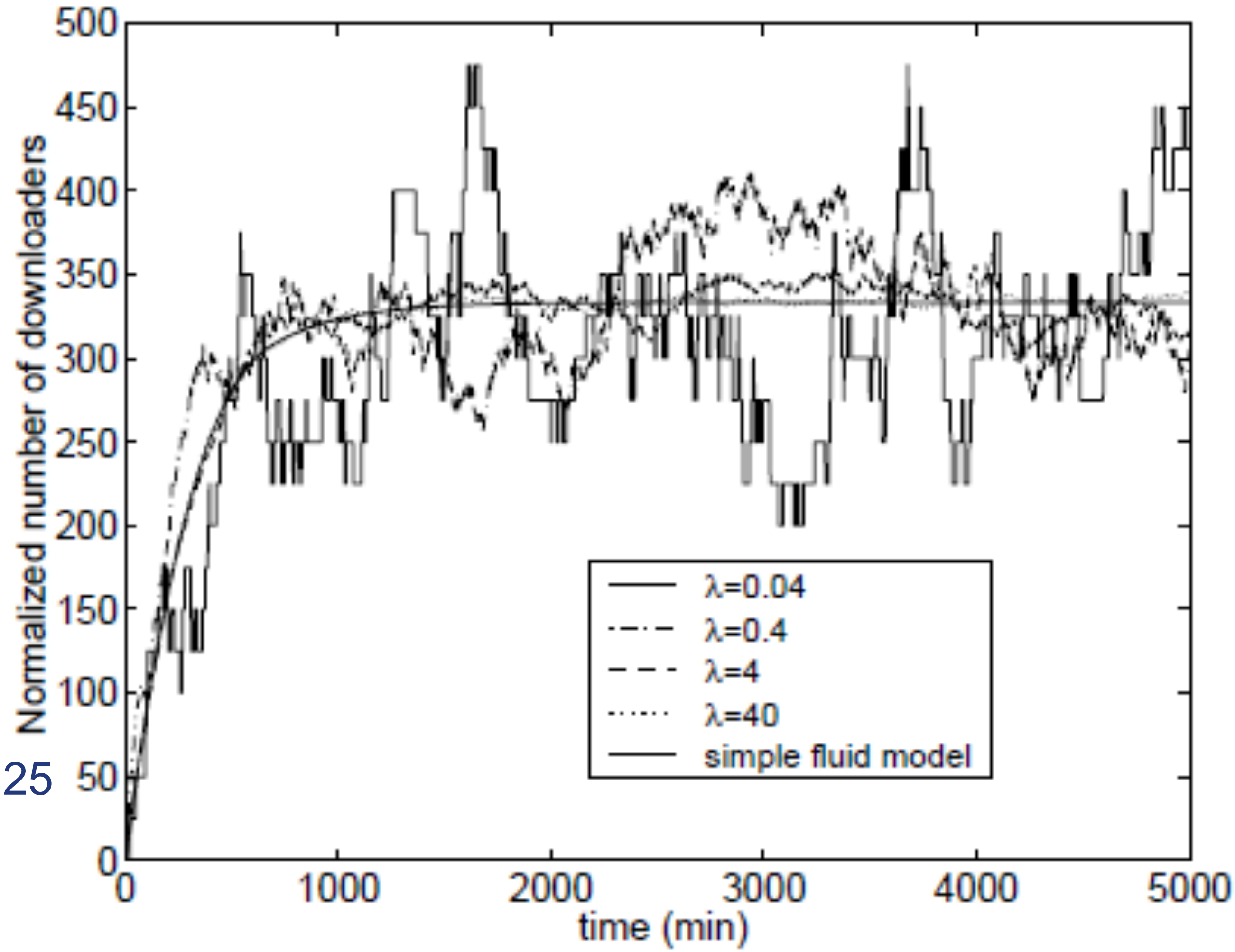
Experiment 1



μ = 0.00125
θ = 0.001
c = 0.002
γ = 0.001

λ : arrival rate of new request

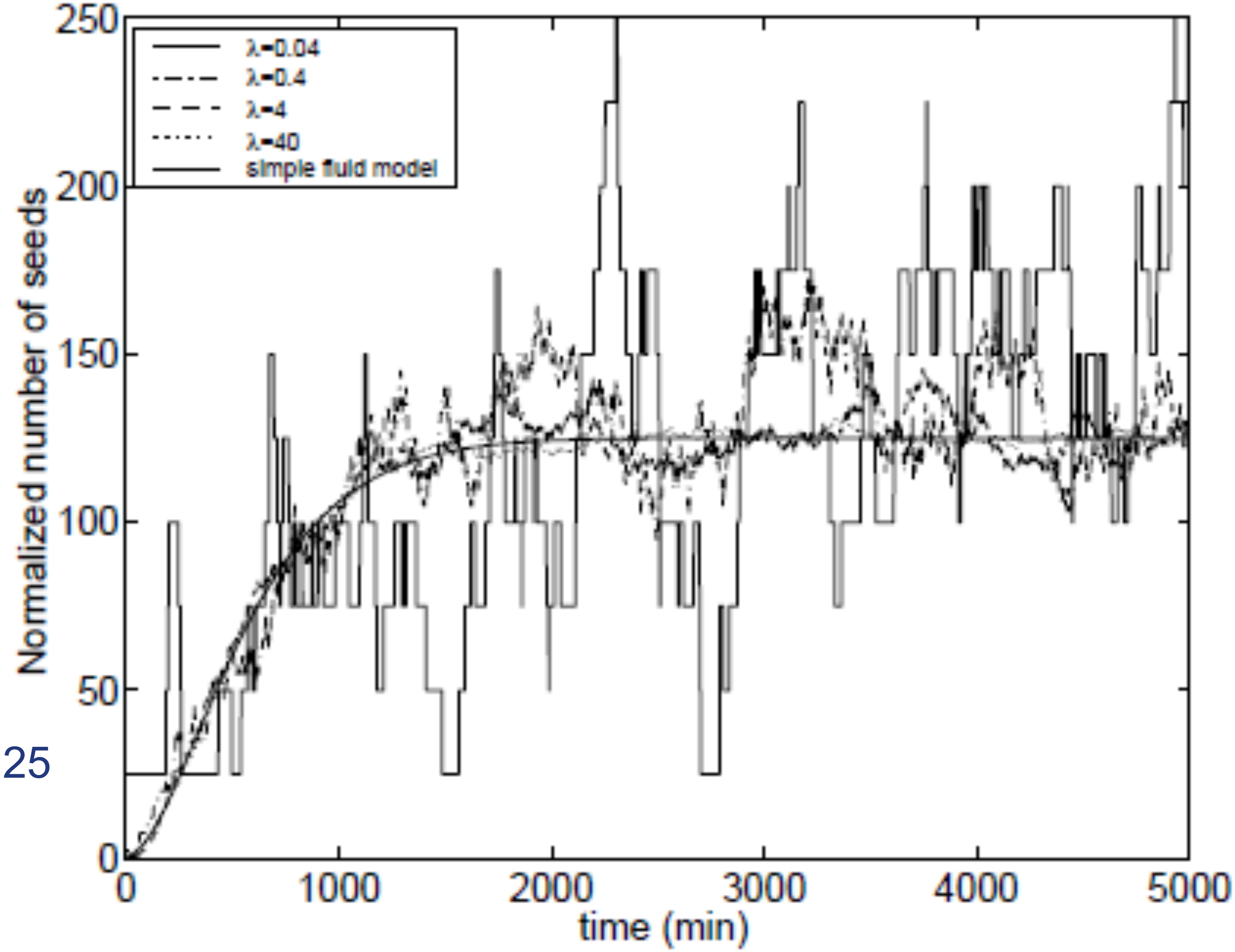
Experiment 1



$\mu = 0.00125$
$\theta = 0.001$
$c = 0.002$
$\gamma = 0.001$

λ : arrival rate of new request

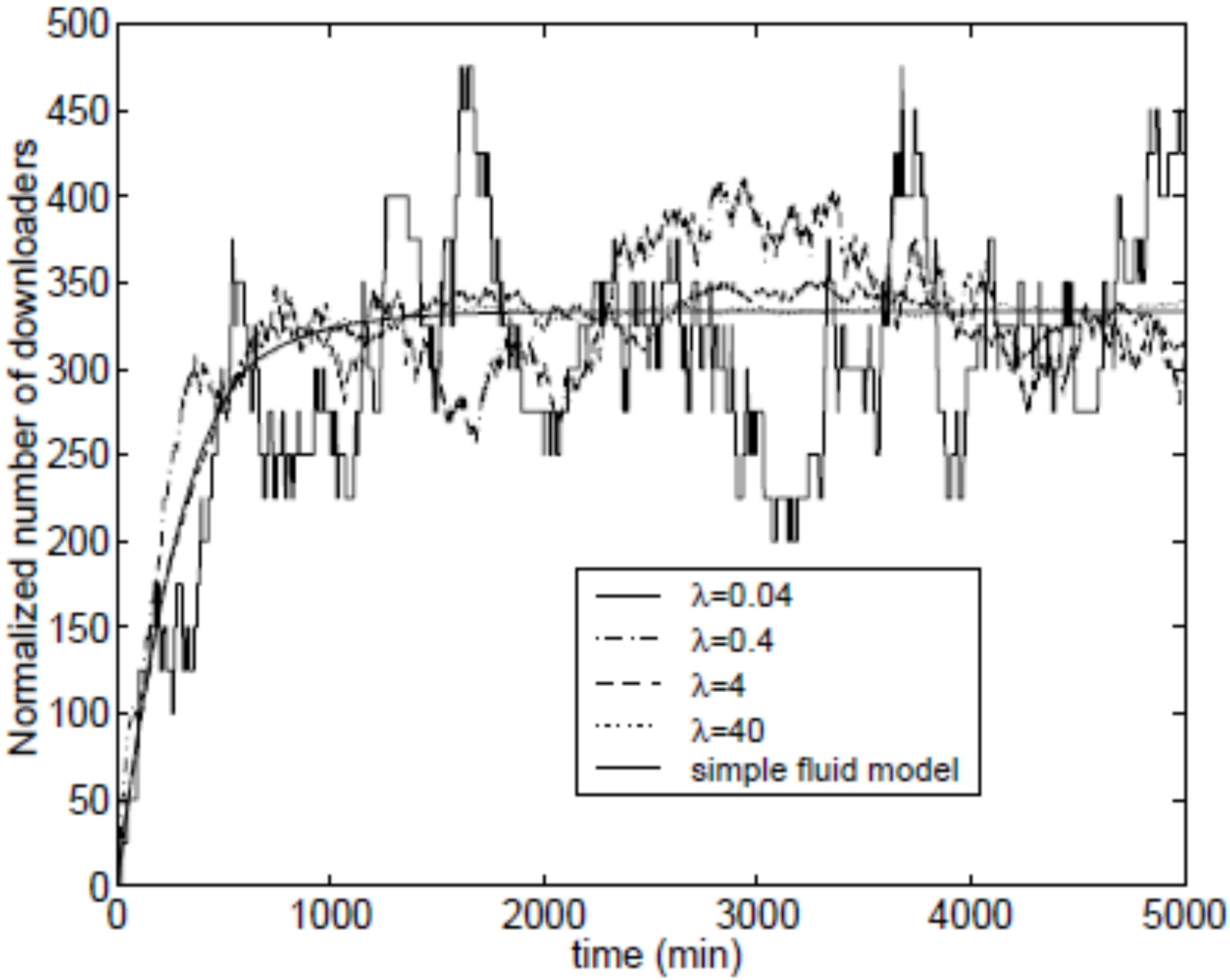
Experiment 2



μ = 0.00125
θ = 0.001
c = 0.002
γ = 0.005

λ : arrival rate of new request

Experiment 2



$\mu = 0.00125$
$\theta = 0.001$
$c = 0.002$
$\gamma = 0.005$

λ : arrival rate of new request

Analysis of (1) & (2)

- The fluid model is a good approximation when the arrival rate λ is large regardless of the relationship between μ and γ.

- The approximation is also good for small λ.

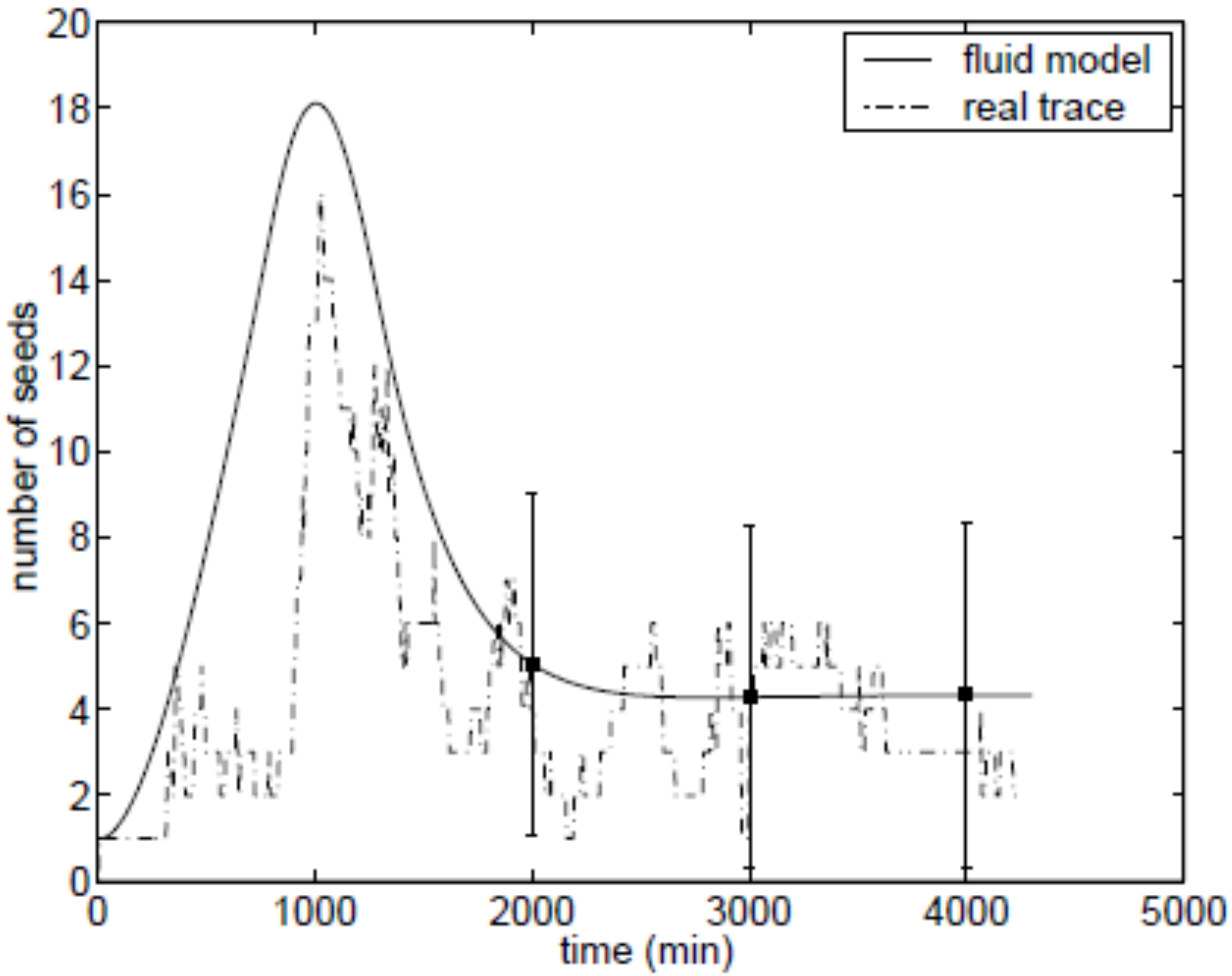
Experiment 3 – Real-world Data

- Assume the system is upload-limited ($\gamma > \mu$).

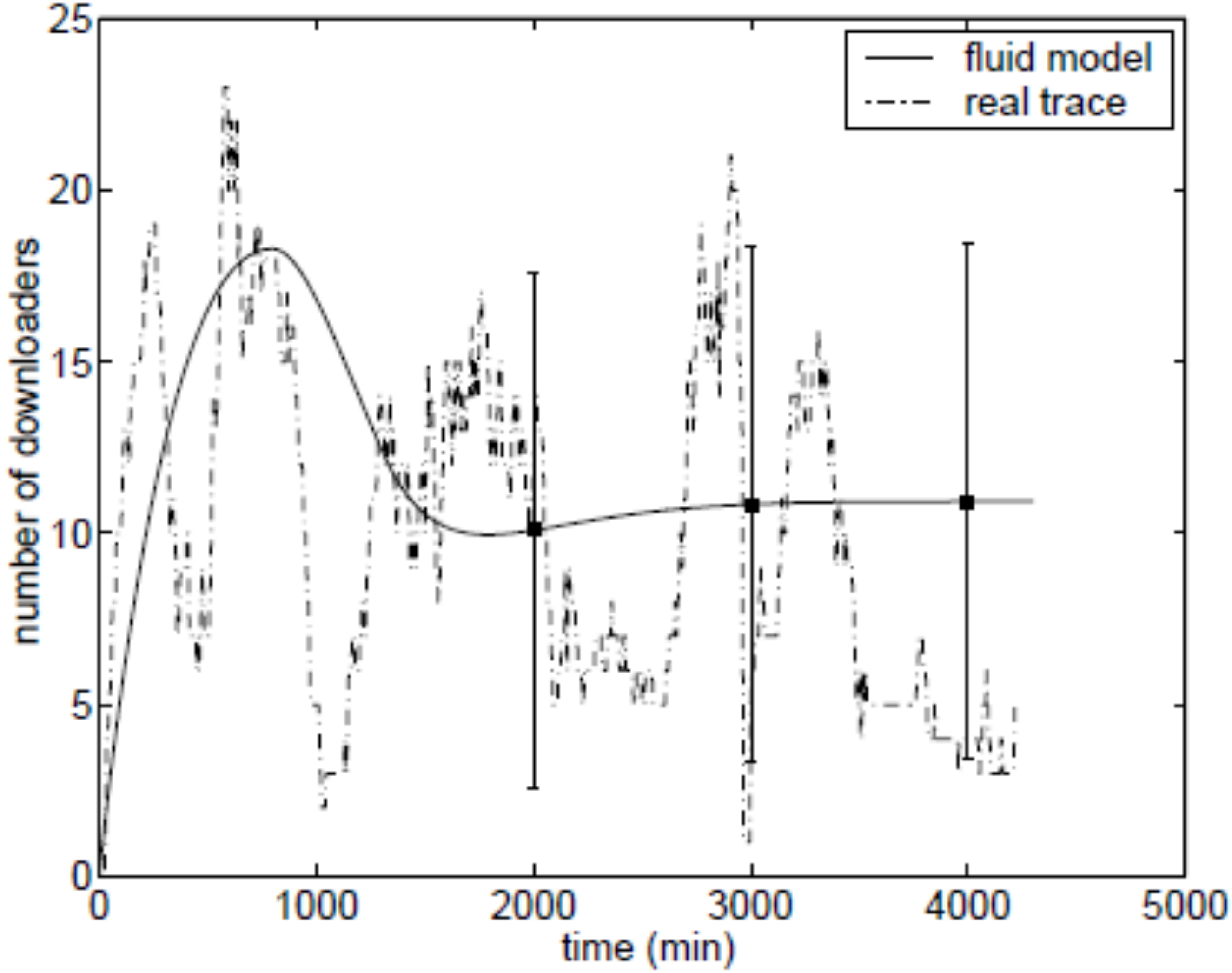
- Measured λ and γ are time-dependent
  - for t < 800 min, $\lambda = 0.06$ $\gamma = 0.001$
  - for t > 1300 min, $\lambda = 0.03$ $\gamma = 0.0044$
  - in between, they vary linearly

- Also numerically calculate the expected variation from the model as a 95% confidence interval.

Experiment 3



Experiment 3



Outline

- Introduction to P2P network
- Introduction to BitTorrent
- A simple fluid model
- The incentive mechanism
- Experiments
- **Conclusions**

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.
- The effect of optimistic unchoking on free-riding.
- The experiments show that the simple fluid model is able to capture the behavior of the system even when the arrival rate is small.