



# Peer-to-peer systems: Modelling and performance

Samuli Aalto  
Aalto University

## Contents

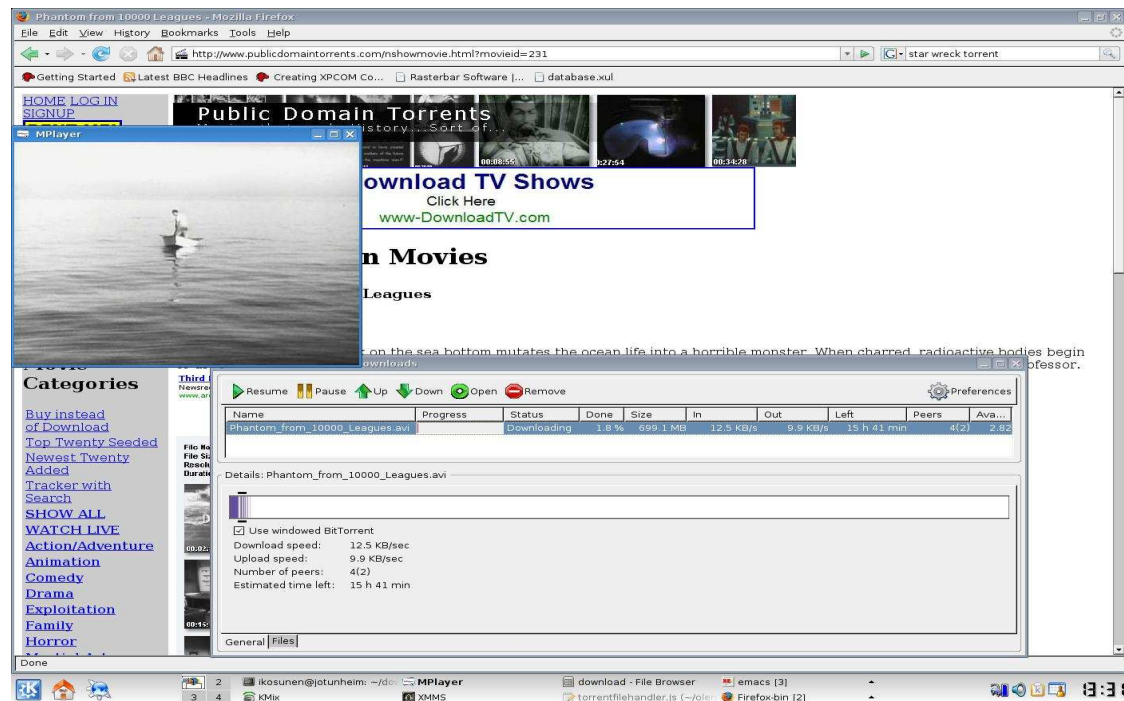
- Peer-to-peer systems
- File sharing: fluid model
- File sharing: steady-state analysis
- File sharing: conclusions
- Video-on-demand: fluid model
- Video-on-demand: steady-state analysis
- Video-on-demand: steady-state synthesis
- Video-on-demand: conclusions

## Fundamental principle

- **Client/Server (CS) paradigm**
  - **Clients** download content from **servers**
  - Clear distinction between the two roles
  - Service capacity remains the same, while load increases
  - When too many clients, transfer times explode
  - Offered load bounded by this stability limit (for sure!)
- **Peer-to-peer (P2P) systems**
  - **Peers** download pieces of content from other peers/seeds and **simultaneously** upload downloaded pieces to other peers
  - Blurring of roles: peers not only act as clients (when downloading) but also serve other peers (when uploading)
  - Service capacity scales with the offered load
  - No stability limit (for sure?)

# Applications

- P2P used commonly for **file sharing** (e.g. BitTorrent) and **live streaming**
- P2P **video-on-demand (VoD)**:
  - Alternative to client-server approaches (YouTube)?
  - Under what conditions?



## Quality of Service

- **P2P file sharing**
  - Retrieve the whole file as soon as possible
  - Retrieve pieces in any order
  - Minimize the file transfer time
- **P2P streaming**
  - Retrieve pieces at least at playback rate and in almost sequential order
  - Minimize the startup delay (needed to fill the playout buffer)
- **P2P video-on-demand**
  - Retrieve the whole file
  - Retrieve pieces at least at playback rate and in almost sequential order
  - Minimize the startup delay (needed to fill the playout buffer)

## Why performance modelling?

- Scalability
  - Is the system really scalable?
- Stability
  - If not, where is the stability limit for the load?
- Performance
  - When stable, is the performance sufficient?

## Modelling aspects

- **Dynamic population model**
  - describing the evolution of the peer population in the P2P system
- **Peer arrival process**
  - steady arrival rate, smoothly attenuating arrival rate, or flash crowd?
- **Efficiency of resource sharing**
  - utilization of a peer's upload capacity
  - effect of the piece/peer selection policy
  - number of parallel connections
- **Selfishness / altruism**
  - part of peers are free-riders that do not want to share upload capacity
- **Download and upload rates**
  - homogeneous or heterogeneous peer population?
- **Number of permanent seeds**
  - correspond to servers in the client-server architecture

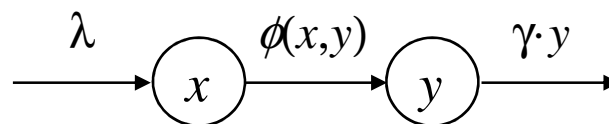
## Contents

- Peer-to-peer systems
- File sharing: fluid model
- File sharing: steady-state analysis
- File sharing: conclusions
- Video-on-demand: fluid model
- Video-on-demand: steady-state analysis
- Video-on-demand: steady-state synthesis
- Video-on-demand: conclusions



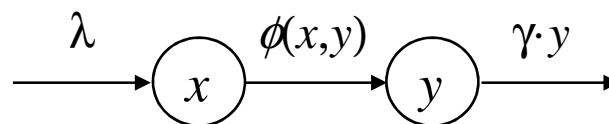
## Model for P2P file sharing

- Life span of a peer consists of two **sequential** phases:
  - **file transfer phase**, during which the peers are called **leechers**
  - **sharing phase**, during which the peers are called **seeds**
- **Altruistic peers** have a longer sharing phase than selfish peers
- Model by **Qiu and Srikant (2004)**:
  - deterministic **fluid model** (= system of differential equations)
  - describing the system dynamics related to sharing of a **single file**
  - $x(t)$  = (average) **number of leechers** at time  $t$
  - $y(t)$  = (average) **number of non-permanent seeds** at time  $t$



## Assumptions

- Steady arrival process described by
  - arrival rate  $\lambda$  to transfer phase (arrivals per time unit)
- Efficiency described by
  - upload utilization ratio  $\eta$  (belonging to  $(0,1]$ )
- Selfishness described by
  - departure rate  $\gamma$  from service phase (departures per time unit)
- Homogeneous peer population with
  - download rate  $c$  (file transfers per time unit) and
  - upload rate  $\mu$  (file transfers per time unit)
- No permanent seeds



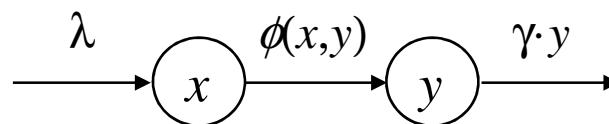
## Fluid model

- Switched nonlinear system:

$$\begin{cases} x'(t) = \lambda - \phi(t), \\ y'(t) = \phi(t) - \gamma y(t), \end{cases} \quad (1)$$

- Aggregate service rate:

$$\phi(t) = \min\{cx(t), \mu(\eta x(t) + y(t))\}. \quad (2)$$



## Contents

- Peer-to-peer systems
- File sharing: fluid model
- File sharing: steady-state analysis
- File sharing: conclusions
- Video-on-demand: fluid model
- Video-on-demand: steady-state analysis
- Video-on-demand: steady-state synthesis
- Video-on-demand: conclusions

## Steady-state analysis

- Solve the equilibrium of the system by setting  $x'(t) = y'(t) = 0$  in (1):

$$\begin{cases} \bar{\phi} = \lambda, \\ \bar{y} = \frac{\lambda}{\gamma}, \end{cases} \quad (3)$$

- Two cases considered separately:
  - **download-constrained** system in equilibrium
  - **upload-constrained** system in equilibrium
- Parameter space divided nicely in two complementary parts each of which has a unique equilibrium solution
  - that are even globally stable by [Qiu and Sang \(2008\)](#)

## Download-constrained system

If

$$c\bar{x} \leq \mu(\eta\bar{x} + \bar{y}), \quad (4)$$

then the system is *download-constrained*, and

$$c\bar{x} = \bar{\phi} = \lambda, \quad (5)$$

implying that

$$\begin{cases} \bar{x}_d = \frac{\lambda}{c}, \\ \bar{y}_d = \frac{\lambda}{\gamma}. \end{cases} \quad (6)$$

The resulting condition for a download-constrained system:

$$\frac{1}{\mu} \leq \frac{\eta}{c} + \frac{1}{\gamma}. \quad (7)$$

## Upload-constrained system

If

$$c\bar{x} > \mu(\eta\bar{x} + \bar{y}), \quad (8)$$

then the system is *upload-constrained*, and

$$\mu(\eta\bar{x} + \bar{y}) = \bar{\phi} = \lambda, \quad (9)$$

implying that

$$\begin{cases} \bar{x}_u = \frac{\lambda}{\eta} \left( \frac{1}{\mu} - \frac{1}{\gamma} \right), \\ \bar{y}_u = \frac{\lambda}{\gamma}. \end{cases} \quad (10)$$

The resulting condition for a upload-constrained system:

$$\frac{1}{\mu} > \frac{\eta}{c} + \frac{1}{\gamma}. \quad (11)$$

## Deterministic model vs. stochastic simulations

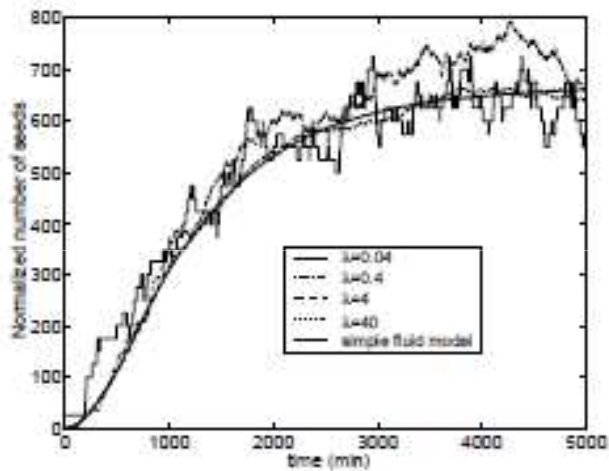


Figure 1: Experiment 1 : The evolution of the number of seeds as a function of time

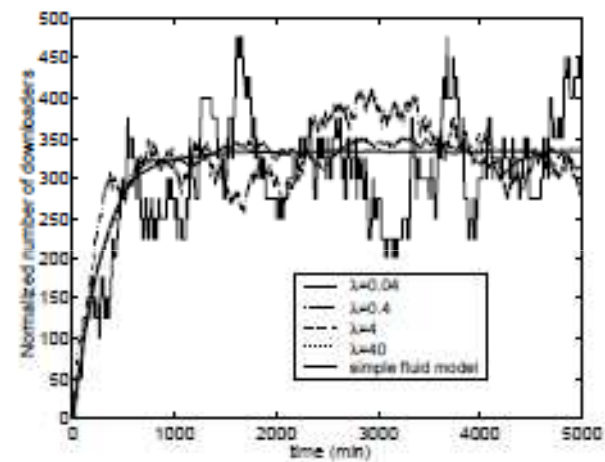


Figure 2: Experiment 1 : The evolution of the number of downloaders as a function of time

Source: [Qiu and Srikant \(2004\)](#)



## Contents

- Peer-to-peer systems
- File sharing: fluid model
- File sharing: steady-state analysis
- File sharing: conclusions
- Video-on-demand: fluid model
- Video-on-demand: steady-state analysis
- Video-on-demand: steady-state synthesis
- Video-on-demand: conclusions

## Conclusions from the P2P file sharing model

- **Scalability**
  - System scalable in the whole parameter space by (6) and (10), in particular for any  $\eta > 0$
- **Stability**
  - Consequently, system stable for any  $\lambda > 0$
- **Performance**
  - By Little's formula, the mean file transfer time is

$$T = \frac{\bar{x}}{\lambda} = \max\left\{\frac{1}{c}, \frac{1}{\eta} \left(\frac{1}{\mu} - \frac{1}{\gamma}\right)\right\} \leq \max\left\{\frac{1}{c}, \frac{1}{\eta\mu}\right\} \approx \max\left\{\frac{1}{c}, \frac{1}{\mu}\right\}. \quad (12)$$

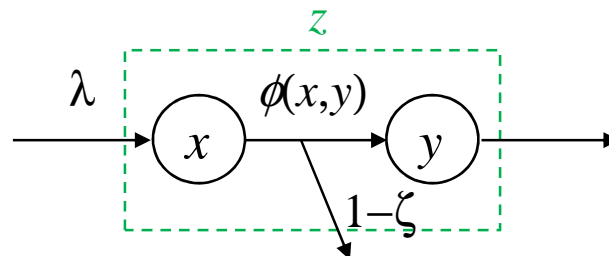
- Thus, no real problems in performance if reasonable download and upload rates with respect to the mean file size
- The last approximation justified for the file sharing application (mainly due to the free retrieving order of pieces)

## Contents

- Peer-to-peer systems
- File sharing: fluid model
- File sharing: steady-state analysis
- File sharing: conclusions
- Video-on-demand: fluid model
- Video-on-demand: steady-state analysis
- Video-on-demand: steady-state synthesis
- Video-on-demand: conclusions

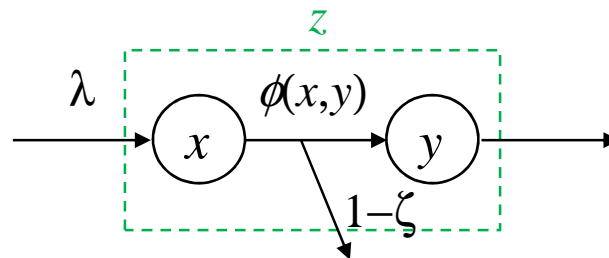
## Model for P2P VoD

- Life span of a peer consists of two **overlapping** phases:
  - **file transfer phase**, during which the peers are called **leechers**
  - **watching phase**, starting immediately after the initial buffering delay
- **Altruistic peers** become **seeds** after the file transfer phase if the watching phase still continues
- Model by **Aalto et al. (2009)**:
  - deterministic **fluid model** (= system of differential equations)
  - describing the system dynamics related to sharing of a **single video file**
  - $x(t)$  = (average) **number of leechers** at time  $t$
  - $y(t)$  = (average) **number of non-permanent seeds** at time  $t$



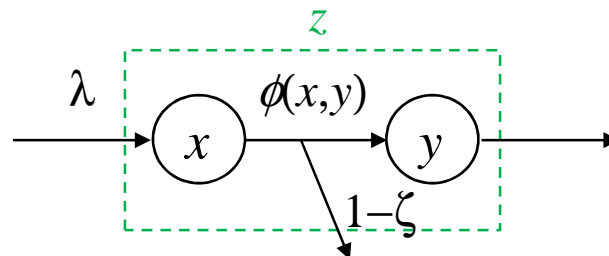
## Assumptions (1)

- Steady arrival process described by
  - arrival rate  $\lambda$  (arrivals per time unit)
- Efficiency described by
  - upload utilization ratio  $\eta$  (belonging to  $(0,1]$ )
- Altruism described by
  - probability  $\zeta$  (for a peer to become a seed)
- Homogeneous peer population with
  - download rate  $c$  (file transfers per time unit) and
  - upload rate  $\mu$  (file transfers per time unit)
- Number of permanent seeds =  $k$  (belonging to  $\{0,1,2,\dots\}$ )



## Assumptions (2)

- Startup delay negligible (if video sufficiently long)
  - Thus, the transfer phase and the playback phase start essentially at the same time
- Video watched at (fixed) playback rate
  - Total **watching time** denoted by  $z$
  - Natural requirement:  $z > 1/c$  (since transfer rate always bounded by  $c$ )
- **Playback quality problems** if the transfer phase takes longer than  $z$ 
  - In this case, the playback phase ends as soon as the transfer is completed
- **Selfish peers** stay in the system until the end of the transfer phase while **altruist peers** stay until the end of the playback phase
  - but no longer, which is a **worst case scenario**



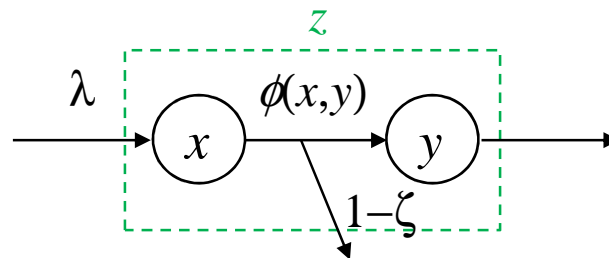
## Fluid model

- Switched nonlinear system:

$$\begin{cases} x'(t) = \lambda - \phi(t), \\ y'(t) = \zeta\phi(t) - \frac{y(t)}{z - x(t)/\lambda}, \end{cases} \quad (13)$$

- Aggregate service rate:

$$\phi(t) = \min\{cx(t), \mu(\eta x(t) + y(t) + k)\}. \quad (14)$$



## Contents

- Peer-to-peer systems
- File sharing: fluid model
- File sharing: steady-state analysis
- File sharing: conclusions
- Video-on-demand: fluid model
- Video-on-demand: steady-state analysis
- Video-on-demand: steady-state synthesis
- Video-on-demand: conclusions



## Steady-state analysis

- Solve the equilibrium of the system by setting  $x'(t) = y'(t) = 0$  in (13):

$$\begin{cases} \bar{\phi} = \lambda, \\ \bar{y} = \zeta \lambda (z - \bar{x} / \lambda), \end{cases} \quad (15)$$

- Two cases considered separately:
  - download-constrained system in equilibrium
  - upload-constrained system in equilibrium
- Multiple solutions found
- Local stability analysis used to rule out some of them

## Download-constrained system

If

$$c\bar{x} \leq \mu(\eta\bar{x} + \bar{y} + k), \quad (16)$$

then the system is *download-constrained*, and

$$c\bar{x} = \bar{\phi} = \lambda, \quad (17)$$

implying that

$$\begin{cases} \bar{x}_d = \frac{\lambda}{c}, \\ \bar{y}_d = \zeta\lambda \left( z - \frac{1}{c} \right). \end{cases} \quad (18)$$

The resulting condition for a download-constrained system:

$$\frac{1}{\mu} \leq \frac{\eta}{c} + \zeta\lambda \left( z - \frac{1}{c} \right) + \frac{k}{\lambda}. \quad (19)$$

## Upload-constrained system (1)

If

$$c\bar{x} > \mu(\eta\bar{x} + \bar{y} + k), \quad (20)$$

then the system is *upload-constrained*, and

$$\bar{\phi} = \mu(\eta\bar{x} + \bar{y} + k), \quad (21)$$

implying that

$$\begin{cases} \bar{x}_u = \frac{\lambda}{\eta - \zeta} \left( \frac{1}{\mu} - \zeta z - \frac{k}{\lambda} \right), \\ \bar{y}_u = \frac{\zeta \lambda}{\eta - \zeta} \left( -\frac{1}{\mu} + \eta z + \frac{k}{\lambda} \right). \end{cases} \quad (22)$$

The resulting conditions for a upload-constrained system:  $\eta \neq \zeta$  and

$$\begin{cases} \frac{1}{\mu} > \frac{\eta}{c} + \zeta \left( z - \frac{1}{c} \right) + \frac{k}{\lambda}, & \text{if } \eta > \zeta, \\ \frac{1}{\mu} < \frac{\eta}{c} + \zeta \left( z - \frac{1}{c} \right) + \frac{k}{\lambda}, & \text{if } \eta < \zeta. \end{cases} \quad (23)$$

## Upload-constrained system (2)

Additionally, for the solution to be meaningful, we require that  $\bar{x}_u > 0$  and  $\bar{y}_u > 0$ . The former one follows from (23), but the latter one leads to the following additional constraints:

$$\begin{cases} \zeta < \frac{1}{z} \left( \frac{1}{\mu} - \frac{k}{\lambda} \right) < \eta, & \text{if } \eta > \zeta, \\ \eta < \frac{1}{z} \left( \frac{1}{\mu} - \frac{k}{\lambda} \right) < \zeta, & \text{if } \eta < \zeta, \end{cases} \quad (24)$$

which implies that  $0 < \frac{1}{\mu} - \frac{k}{\lambda} < z$  is a necessary condition for the existence of a non-negative upload constrained solution.

## Summary of the steady-state analysis (1)

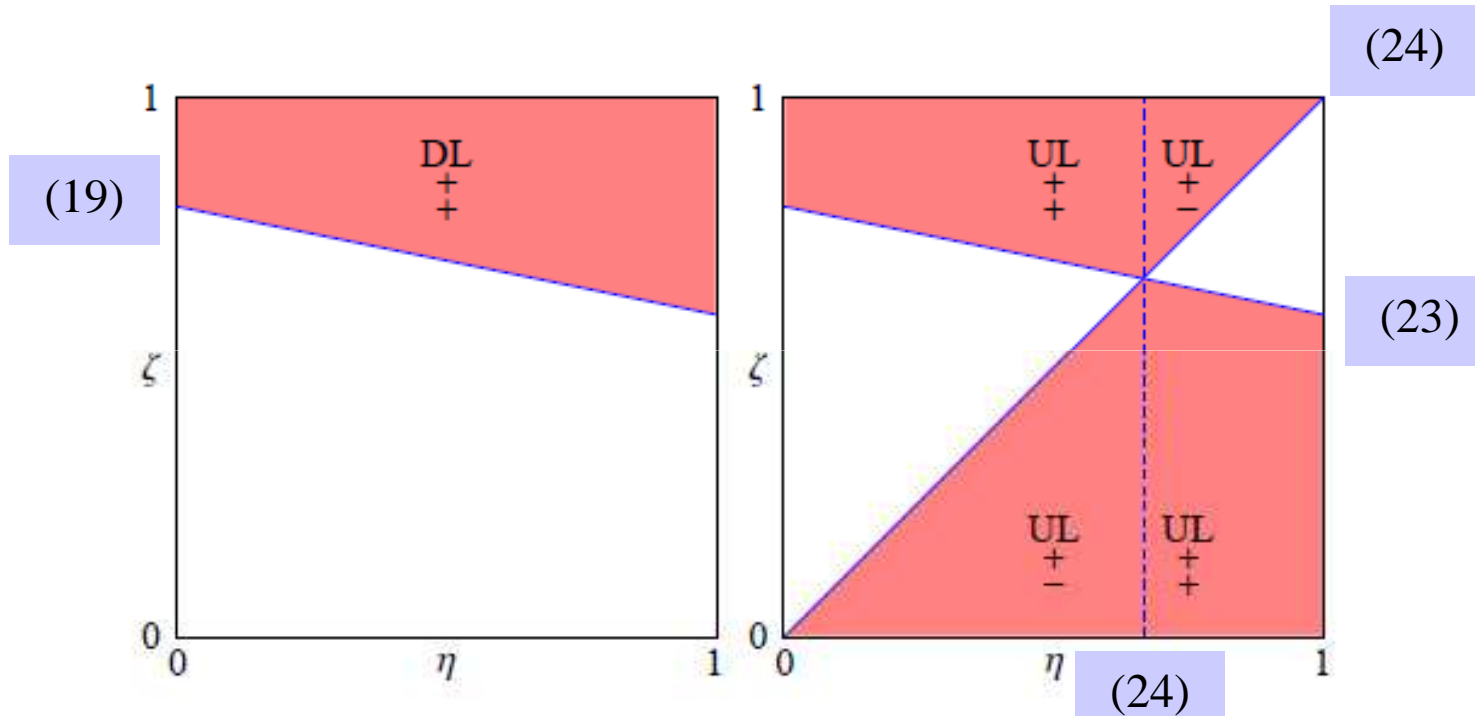


Fig. 1. *Left panel:* Download-constrained solution with  $+/+$  expressing that  $\bar{x}_d > 0$  and  $\bar{y}_d > 0$  in this area. *Right panel:* Upload-constrained solution with  $+/+ [+/-]$  expressing that  $\bar{x}_u > 0$  and  $\bar{y}_u > 0$  [ $\bar{y}_u < 0$ ] in this area.

## Summary of the steady-state analysis (2)

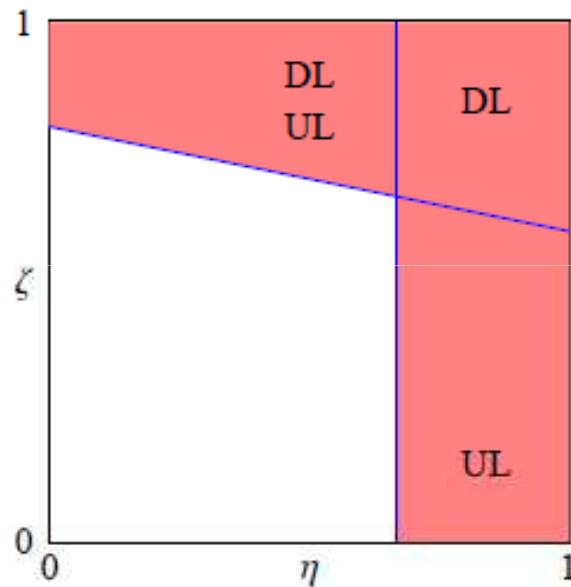


Fig. 2. Solution areas, where DL [UL] refers to a positive download [upload] constrained solution. The horizontal bordering line satisfies  $\frac{1}{\mu} = \frac{\eta}{c} + \zeta(z - \frac{1}{c}) + \frac{k}{\lambda}$  and the vertical bordering line satisfies  $\eta = \frac{1}{z}(\frac{1}{\mu} - \frac{k}{\lambda})$ .

## Contents

- Peer-to-peer systems
- File sharing: fluid model
- File sharing: steady-state analysis
- File sharing: conclusions
- Video-on-demand: fluid model
- Video-on-demand: steady-state analysis
- Video-on-demand: steady-state synthesis
- Video-on-demand: conclusions

## Steady-state synthesis (1)

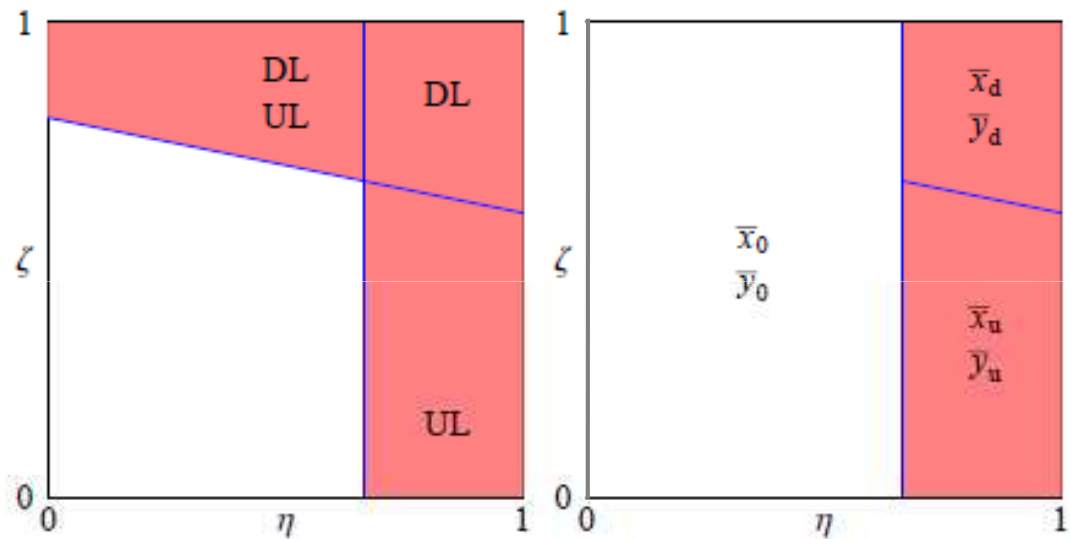


Fig. 1. *Left panel:* Solution areas, where DL [UL] refers to a positive download [upload] constrained solution. The horizontal bordering line satisfies  $\frac{1}{\mu} = \frac{\eta}{c} + \zeta(z - \frac{1}{c}) + \frac{k}{\lambda}$  and the vertical bordering line satisfies  $\eta = \frac{1}{z}(\frac{1}{\mu} - \frac{k}{\lambda})$ . *Right panel:* Steady state synthesis.



## Steady-state synthesis (2)

- If

$$\eta < \frac{1}{z} \left( \frac{1}{\mu} - \frac{k}{\lambda} \right)$$

transfer rate < playback rate,  
i.e. **playback quality problems**

- Number of leechers and seeds well estimated by  $(x_0, y_0)$ :

$$\begin{cases} \bar{x}_0 &= x_u|_{\zeta=0} = \frac{\lambda}{\eta} \left( \frac{1}{\mu} - \frac{k}{\lambda} \right), \\ \bar{y}_0 &= y_u|_{\zeta=0} = 0. \end{cases}$$

- If

$$\eta > \frac{1}{z} \left( \frac{1}{\mu} - \frac{k}{\lambda} \right)$$

transfer rate > playback rate,  
i.e. **sufficient playback quality**

- If further

$$\frac{1}{\mu} \leq \frac{\eta}{c} + \zeta \lambda \left( z - \frac{1}{c} \right) + \frac{k}{\lambda}.$$

DL constrained system  $(x_d, y_d)$

- Otherwise  
UL constrained system  $(x_u, y_u)$

## Deterministic model vs. stochastic and BitTorrent simulations

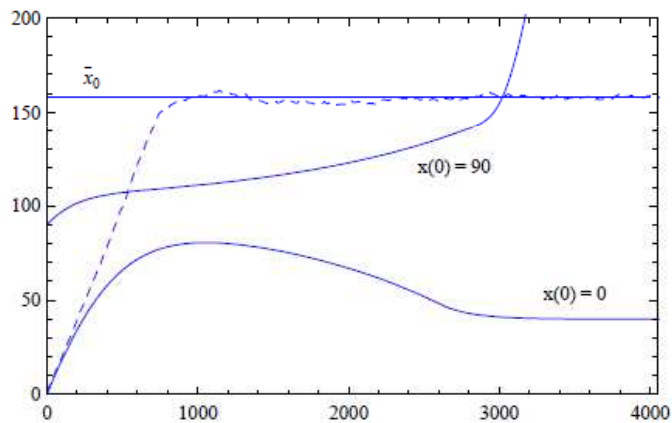


Fig. 5. Comparison of the fluid model (solid smooth lines) against the stochastic model (dashed line) with  $\eta = 0.5$  and  $\zeta = 0.8$ .

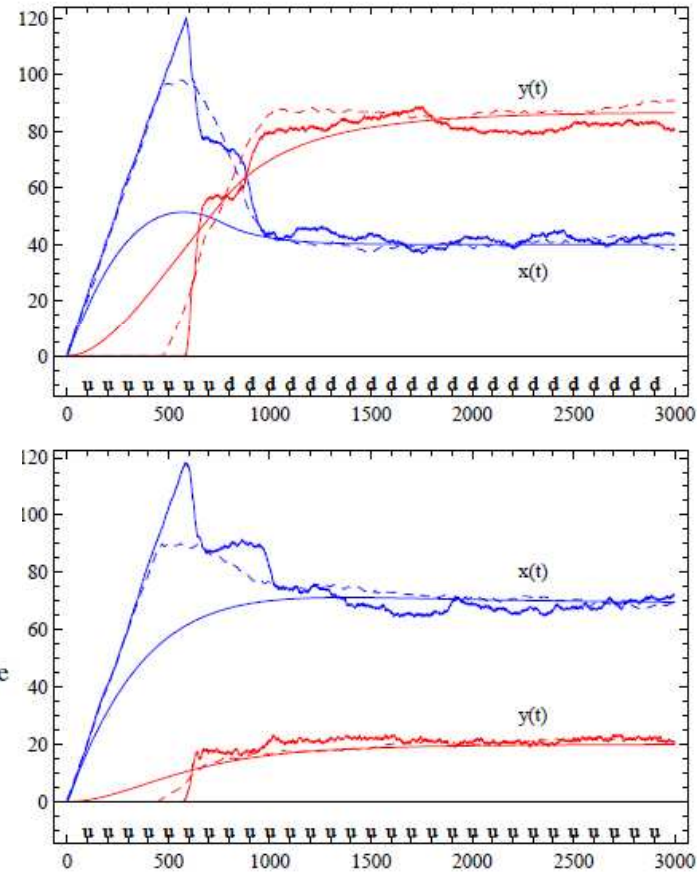


Fig. 4. Comparison of the fluid model (solid smooth line) against the stochastic model (dashed line) and the BitTorrent simulation (solid jagged line) with  $\zeta = 0.9$  (upper panel) and  $\zeta = 0.3$  (lower panel).

## Contents

- Peer-to-peer systems
- File sharing: fluid model
- File sharing: steady-state analysis
- File sharing: conclusions
- Video-on-demand: fluid model
- Video-on-demand: steady-state analysis
- Video-on-demand: steady-state synthesis
- Video-on-demand: conclusions

## Conclusions from the P2P VoD model

- **Scalability**
  - System scalable in the whole parameter space by the synthesis, in particular for any  $\eta > 0$
- **Stability**
  - Consequently, system stable for any  $\lambda > 0$
- **Performance**
  - Playback quality problems if the efficiency parameter  $\eta$  is too small
  - On the other hand, performance even "scales" (= good quality for all  $\lambda$ ) if the efficiency parameter  $\eta$  is sufficiently large
  - Transfer rates for DL and UL constrained cases:

$$\begin{cases} R_d = c \\ 1/z < R_u < c \end{cases}$$

## References

- [1] D. Qiu and R. Srikant, Modeling and performance analysis of BitTorrent like peer-to-peer networks, in *ACM SIGCOMM*, pp. 367-378, 2004.
- [2] D. Qiu and W. Sang, Global stability of peer-to-peer file sharing systems, *Computer Communications*, 31, 2, 212-219, 2008.
- [3] S. Aalto, P. Lassila, N. Raatikainen, P. Savolainen, and S. Tarkoma, P2P Video-on-Demand: Steady State and Scalability, submitted, 2009.