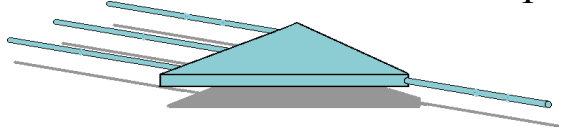
Scalability of the BitTorrent P2P Application

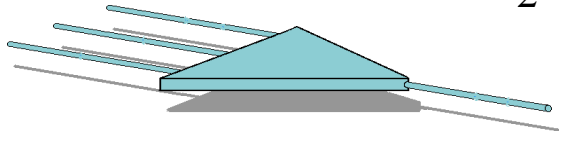
Kolja Eger, Ulrich Killat
Hamburg University of Technology

5.Würzburger Workshop
18.-19. July 2005



Overview

- File dissemination in peer-to-peer (p2p) networks
- Analytical model
- *BitTorrent* architecture and protocol
- Simulation set-up
- Simulation results
- Conclusion and future work



File Dissemination in P2P Networks

- Objective:

  Dissemination of a file to a number of users in minimal time

- Key concept:
  - Segmentation of a file into smaller pieces or chunks which can be shared before download of the whole file is completed

- Efficiency: All available resources of the peers should be used
- Fairness: Peers which contribute much should also gain much → incentive to contribute

Analytical Model

- What is the overall download time for N_P-1 peers to download a file of size S_F from one peer (or seed)?

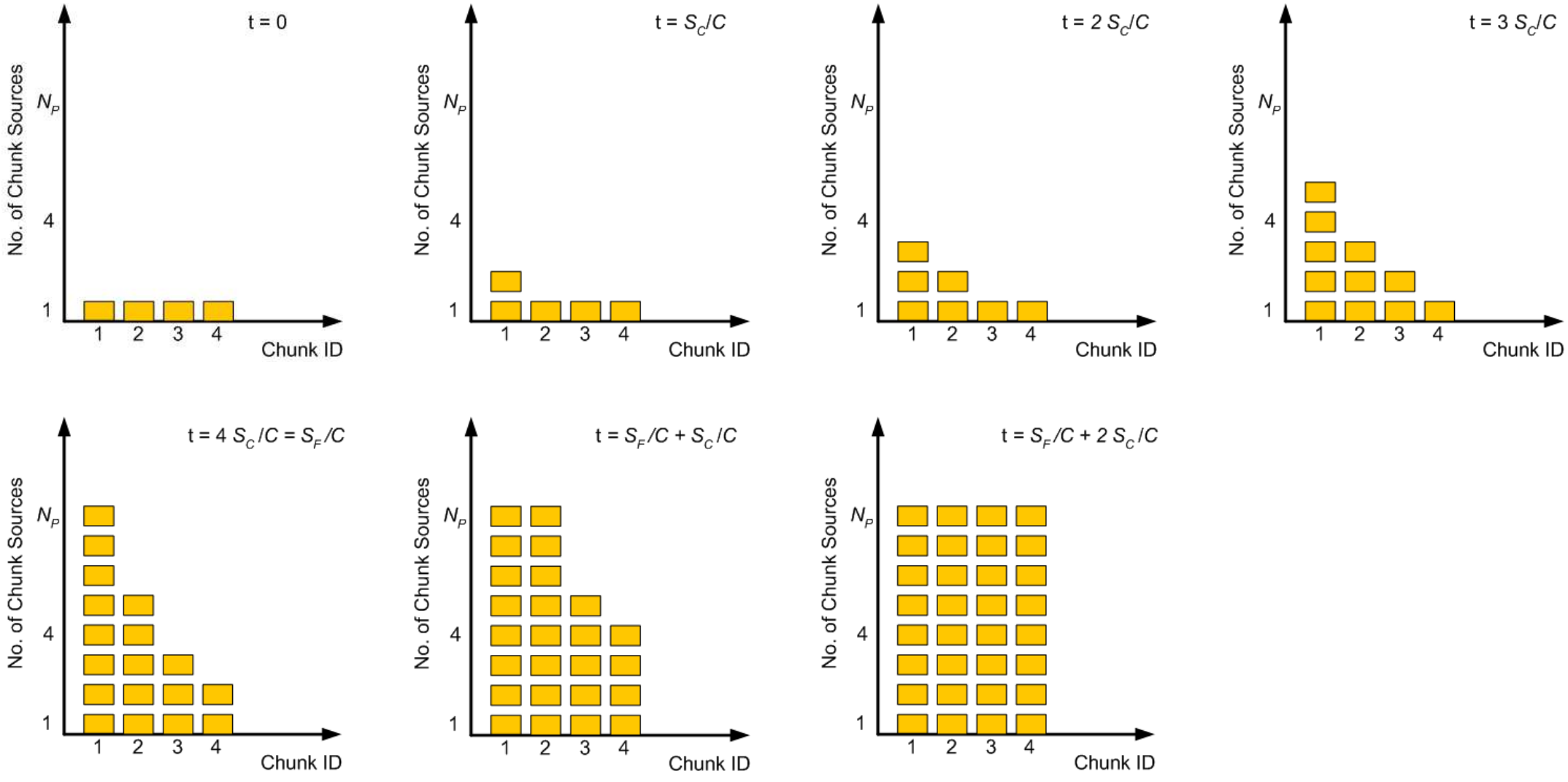
- Assumptions:
  - Flash crowd: peers enter the network at the same time
  - Peer's uplink and downlink capacities are the bottlenecks in the whole network and are equal to C
  - Capacity C is the same for all peers

File Size	S_F
Chunk Size	S_C
# of peers	N_P
Capacity	C

- For $N_P = 2^n$

$$t_{\mathrm{p2p}} = \frac{S_F}{C} + \frac{S_C}{C} \cdot \left(\log_2\left(N_P\right) - 1\right)$$

Analytical Model – Example for N_P=8



Analytical Model (cont.)

- In general

$$t_{\mathrm{p2p}} = \frac{S_F}{C} + \frac{S_C}{C} \cdot \left(\lfloor \log_2 (N_P) - 1 \rfloor + \min \left(1, \frac{N_P}{2^{\lfloor \log_2 (N_P) - 1 \rfloor}} - 2 \right) \right)$$

File Size	$S_F = 4$
Chunk Size	$S_C = 1$
# of peers	$N_P = 5$
# of seeds	$N_S = 1$
Capacity	$C = 1$

PEER ID	CHUNK ID 0	1	2	3
0	*0.0*	*0.0*	*0.0*	*0.0*
1	*1.0*	*3.0*	*4.0*	
2	*3.0*	*2.0*		
3	*2.0*	*4.0*		*4.0*
4	*3.0*	*4.0*	*3.0*	

- Model concentrates on symmetric access line capacities
- But access line capacities are often asymmetric (e.g. ADSL)

BitTorrent - Unchoking

- Popular application for p2p file dissemination
- Each peer uploads to a number of other peers (default = 4)
- This is called unchoking in BitTorrent
- Unchoking algorithm:
  - Every 10s unchoke the 4 peers which have the best upload rate and are interested in a data piece
  - Optimistic Unchoke:
    - Every 30s unchoke one peer regardless of its rate
    - Newly connected peers are three times as likely as others
- Seed:
  - Unchoke the 4 peers which have the best download rate and are interested

BitTorrent - Chunk Selection

- After peer is unchoked by another one, it requests pieces of a chunk according the following rules:
  - Strict Priority
    - After receiving some bytes of a chunk, request remaining parts instead of another chunk
  - Rarest First
    - Request the chunk which fewest of the known peers has
  - Random First Piece
    - At the time the peer has nothing, request a random chunk

Changes to BitTorrent

- Seeds upload to the peers which have the fewest number of chunks

  → New peers complete first chunk quickly and can contribute their resources to the network

- Chunk selection
  - Rarest-first instead of strict-priority, if
    - Uploading peer is a seed
    - Another peer is uploading this chunk currently

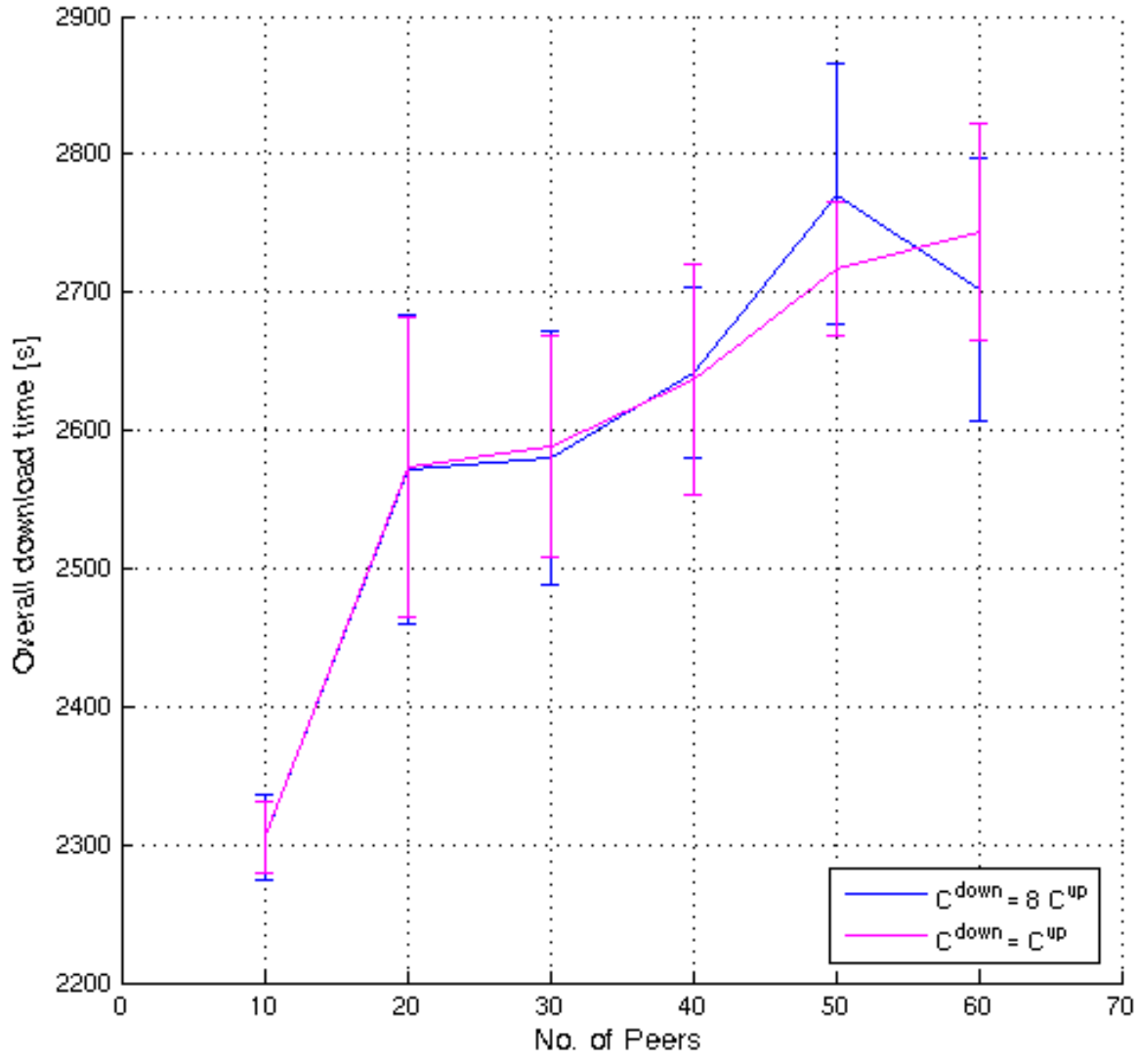
P2P Simulations

- Simulation of p2p network with the packet simulator ns2
- Assumptions:
  - No network losses
  - E2e propagation delay between peers is determined randomly between 25-100ms
  - No limit of TCP connections for each peer, tracker returns max. 50 addresses
- One seed at the beginning of the simulation
- Download of a 10MB file with chunks of size 256KB is investigated

Simulation Results (I)

BitTorrent with different download capacities

- $C^{up} = 10$ KB/s



Simulation Results (II)

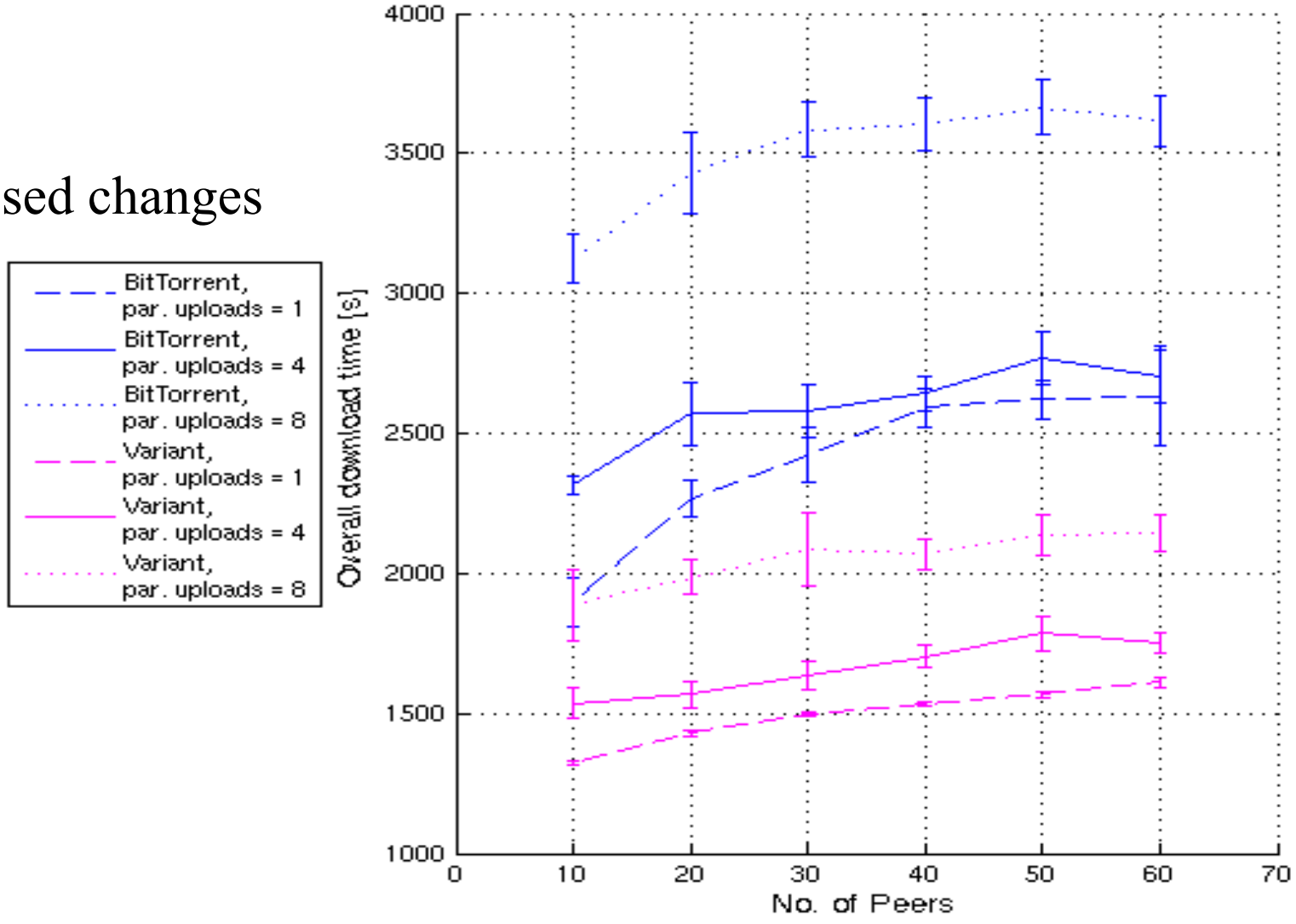
BitTorrent

vs.

BitTorrent with proposed changes

- Different number of parallel uploads (unchokes)

- $C^{up} = 10$ KB/s
- $C^{down} = 8\ C^{up}$



Simulation Results (III)

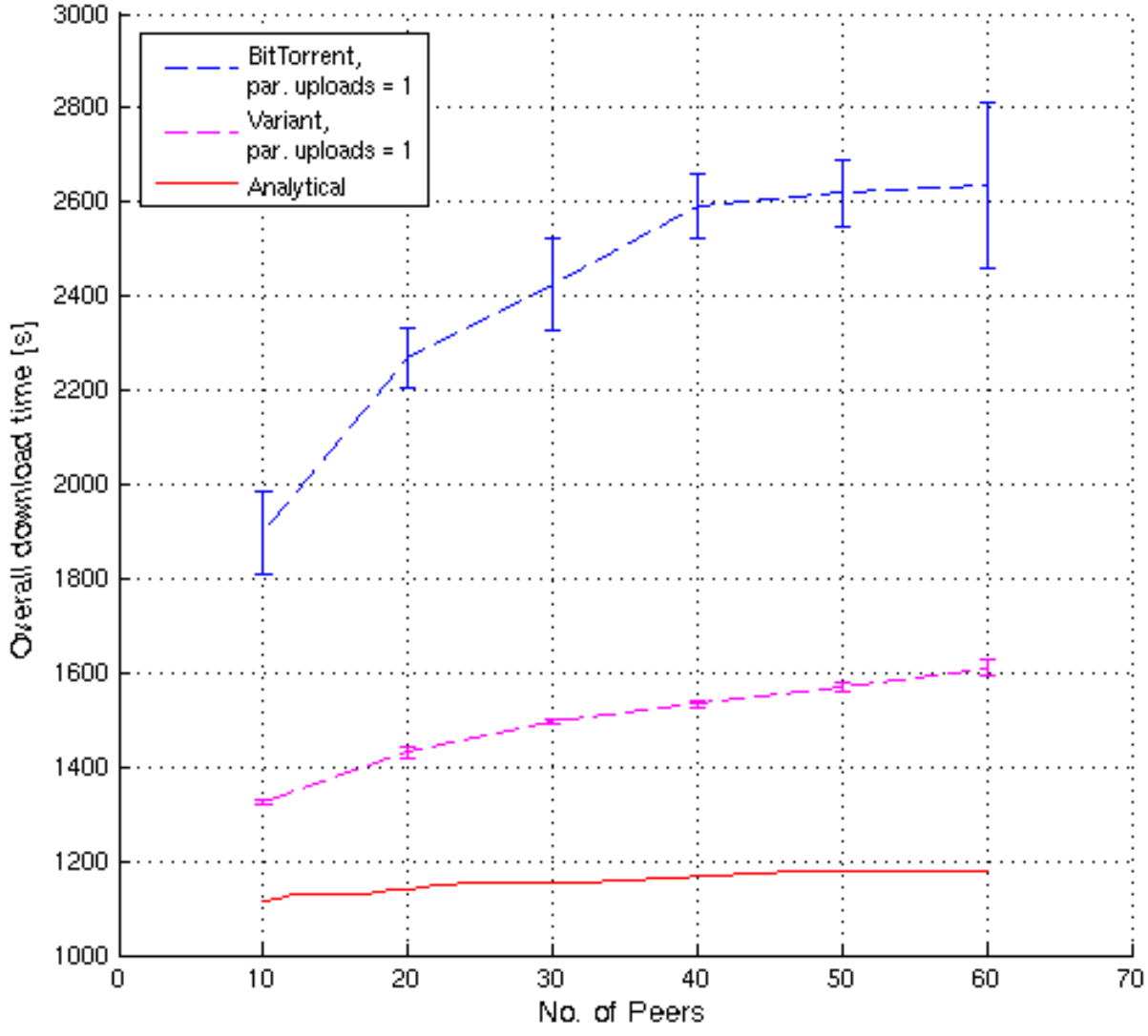
BitTorrent*

vs.

BitTorrent with proposed changes*

vs.

Analytical model



* results are the same as on Slide 12

Conclusion and Future Work

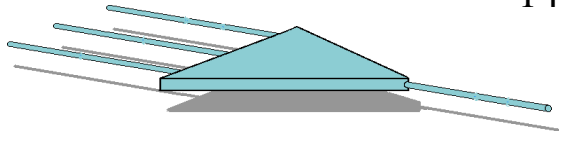
- Slow increase of download time for increasing number of peers, especially with proposed changes

  → scalable

- Performance of p2p application is sensitive to small changes

  → More work is needed

- Future work:
  - Other simulation models (e.g. Poisson arrival process of peers)
  - Defining file dissemination as optimisation problem



Thank you for your attention!