Using Torrent Inflation to Efficiently Serve the Long Tail in Peer-assisted Content Delivery Systems

Niklas Carlsson
University of Calgary
Canada

Derek Eager
University of Saskatchewan
Canada

Anirban Mahanti
NICTA
Australia

IFIP Networking, Chennai, India, May 11, 2010

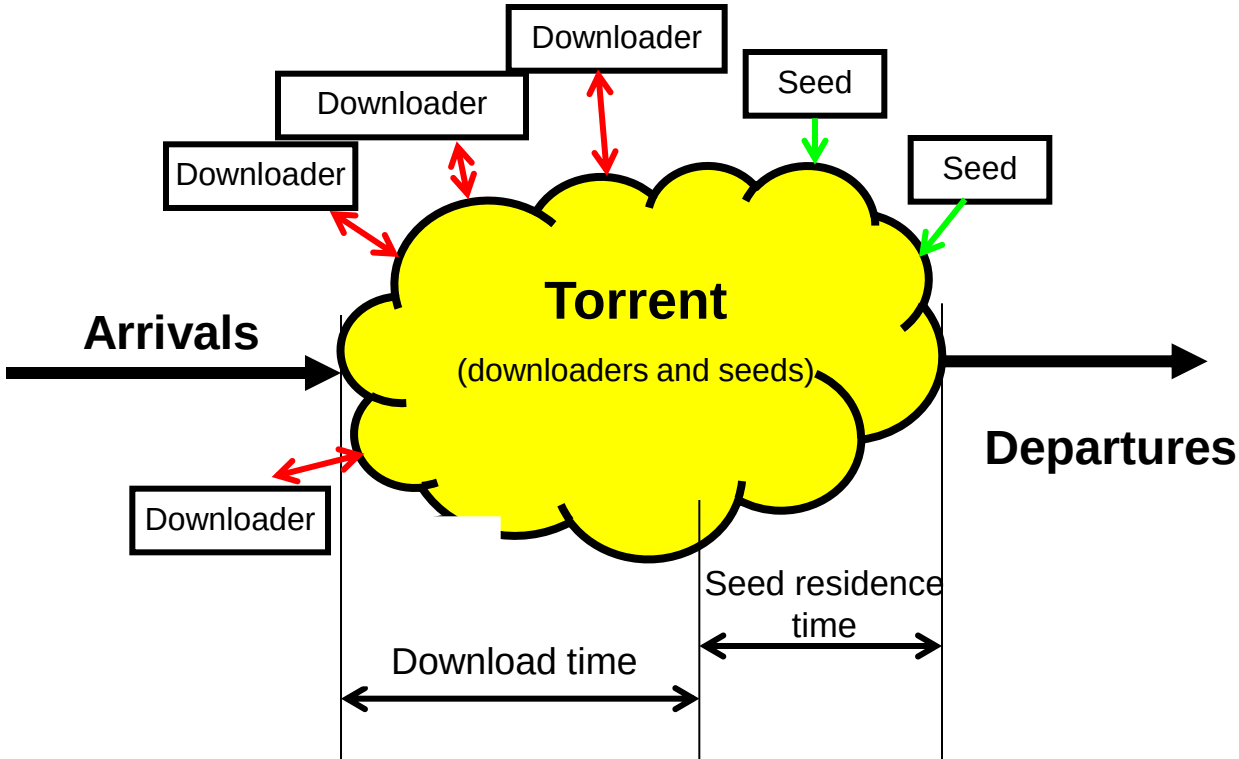
Scalable Content Delivery
Motivation

- Use of Internet for content delivery is massive … and becoming more so (e.g., recent projection that by 2013, 90% of all IP traffic will be video content)
- How to make scalable and efficient?
- Variety of approaches: broadcast/multicast, batching, replication/caching (e.g. CDNs), P2P, peer-assisted, …
- In this talk:
  - Serving the "long tail" in peer-assisted systems

Download using BitTorrent

Background: BitTorrent-like systems

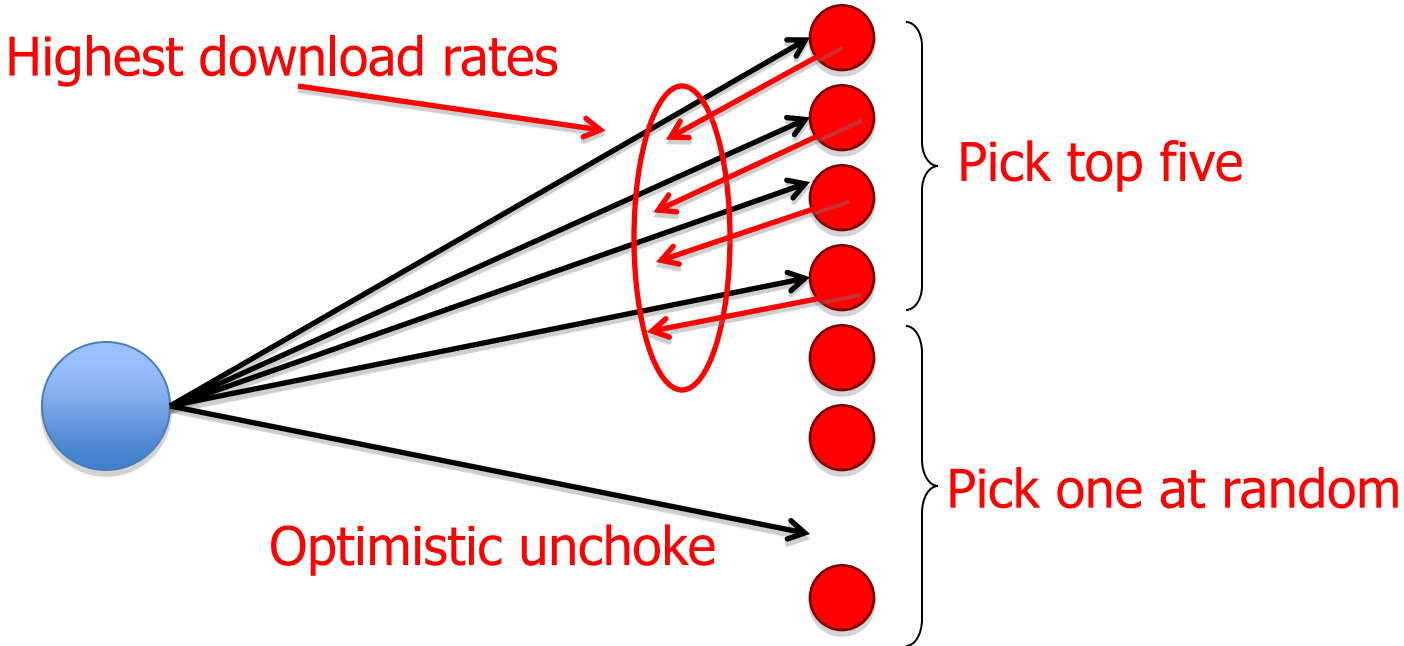
- File split into many smaller pieces
- Pieces are downloaded from both seeds and downloaders
- Distribution paths are dynamically determined
  - Based on data availability



Download using BitTorrent

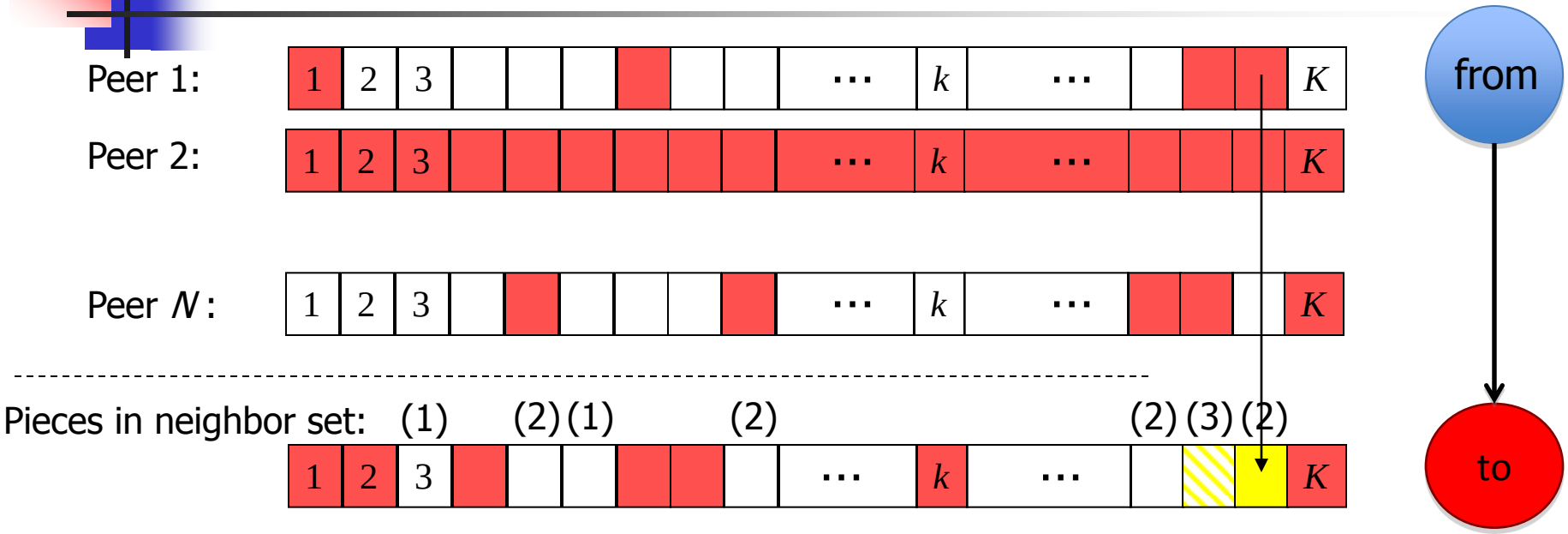
Background: Incentive mechanism

- Establish connections to large set of peers
  - At each time, only upload to a small (changing) set of peers
- Rate-based tit-for-tat policy
  - Downloaders give upload preference to the downloaders that provide the highest download rates



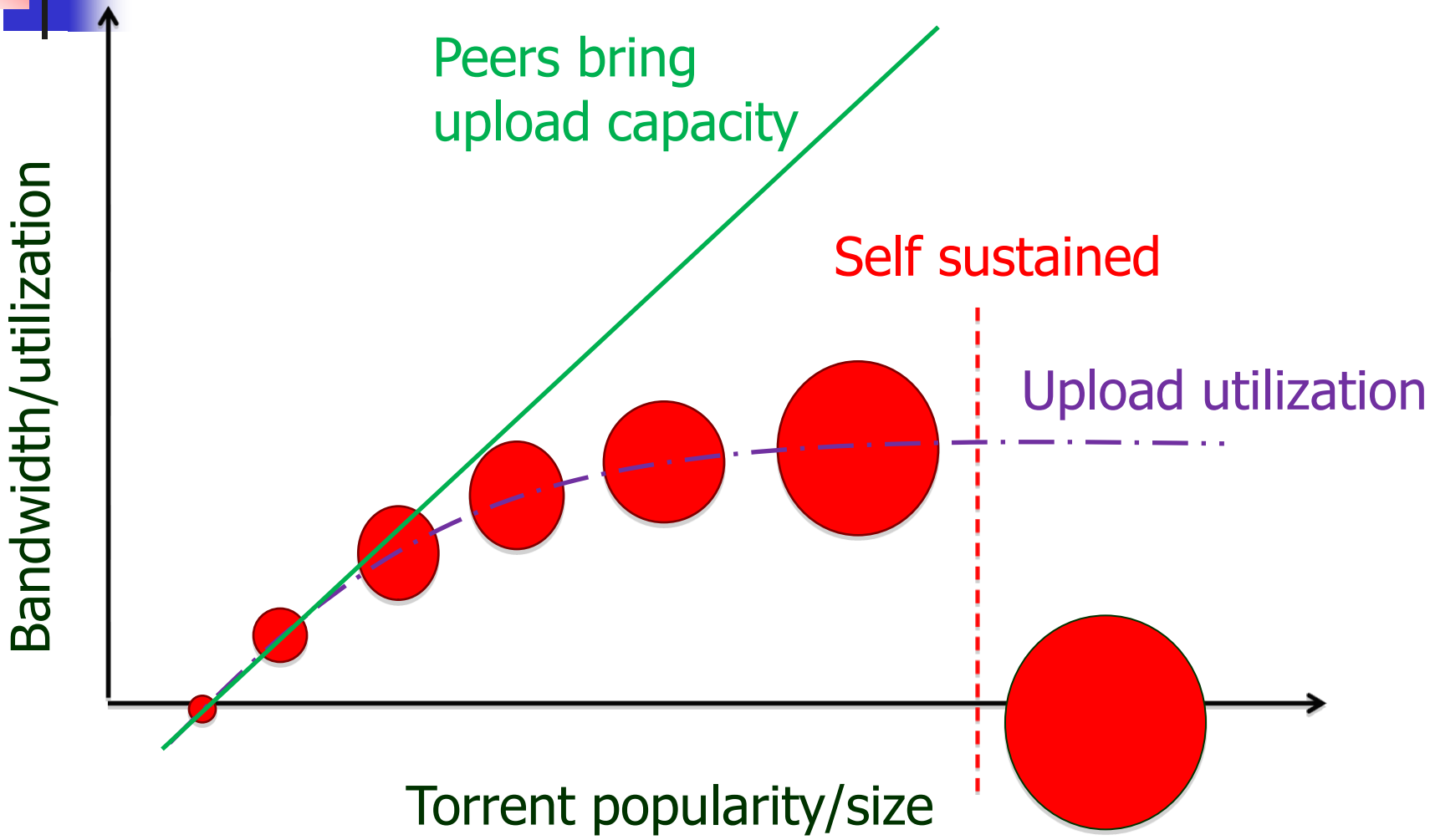
Download using BitTorrent

Background: Piece selection



- Rarest first piece selection policy
  - Achieves high piece diversity
- Request pieces that
  - the uploader has;
  - the downloader is interested (wants); and
  - is the rarest among this set of pieces

Single Torrent Bandwidth Scale



Scalability and the "long tail"

Consider a peer-assisted system

Clients contribute their upload bandwidth when downloading a file from the server (files seeded only by server)

Solves scalability problem, right?

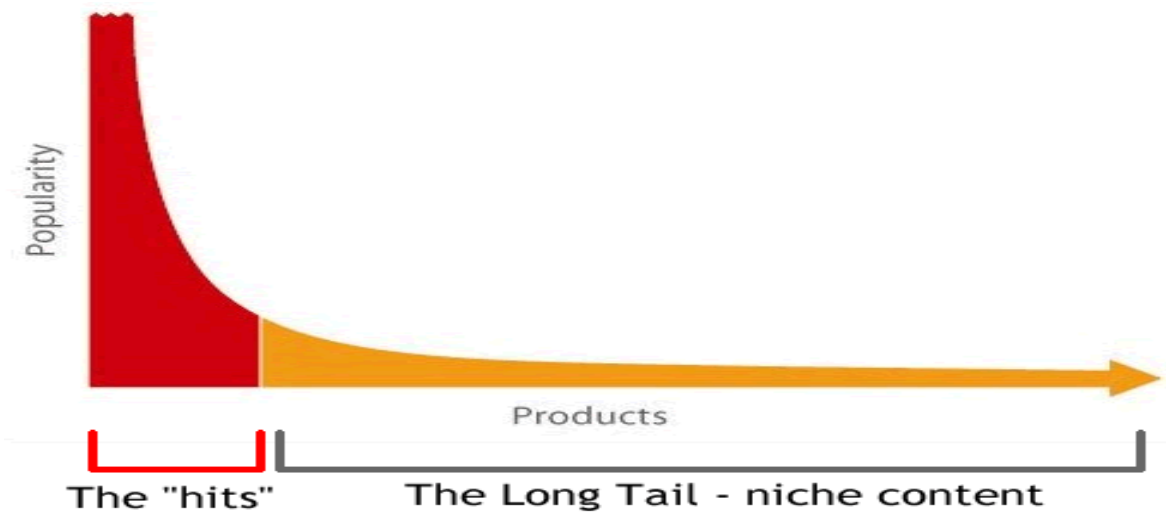
Scalability and the "long tail"

Consider a peer-assisted system

Clients contribute their upload bandwidth when downloading a file from the server (files seeded only by server)

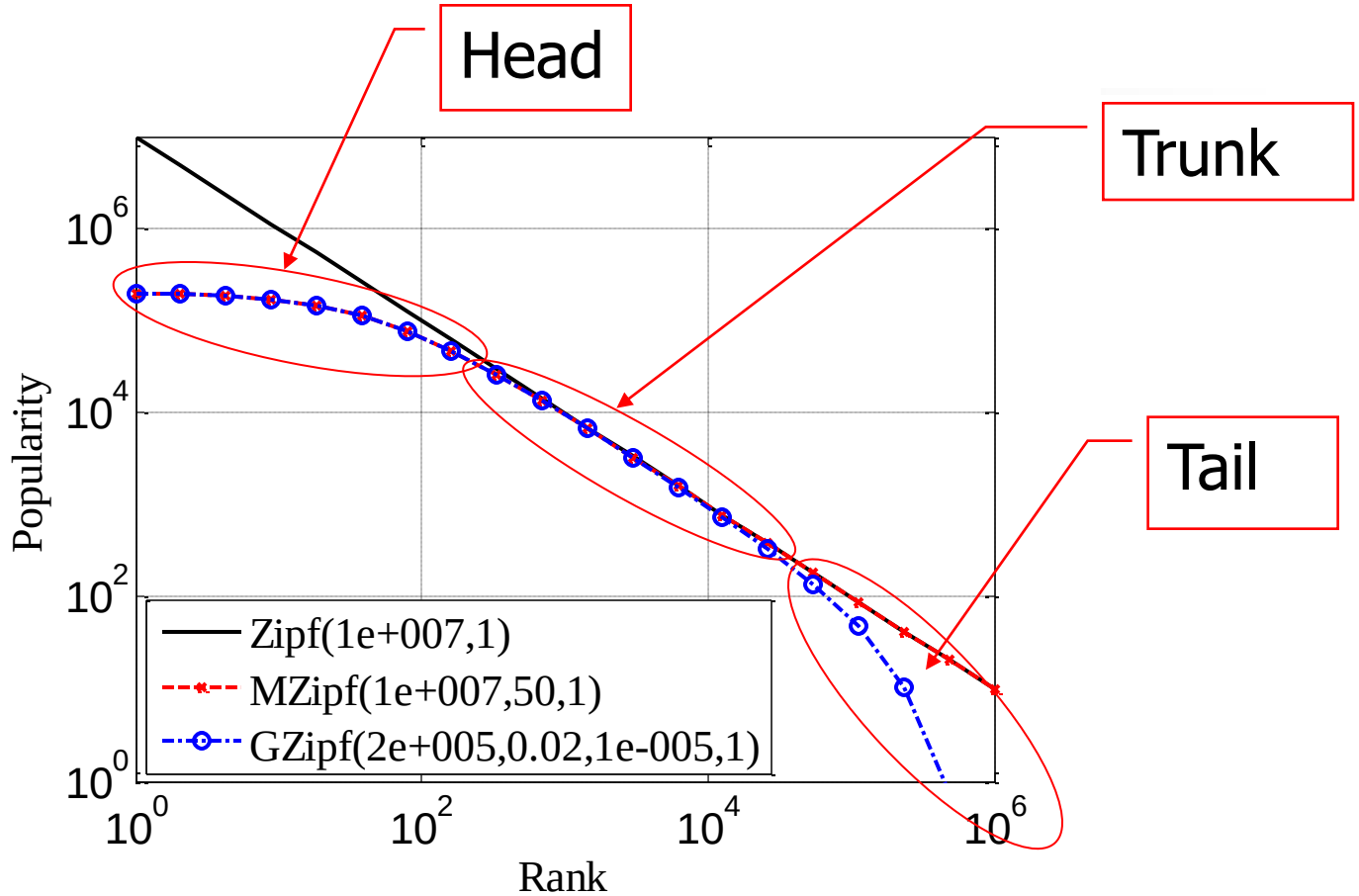
Solves scalability problem, right?

Perhaps not …



Zipf's Law and Beyond

Measurements
E.g., IPTPS '10



Power law file popularity distributions with "long tail"

- Substantial aggregate request rate
- But that individually have too few requestors to form active torrents

Dynamic server scheduling

First question … how to allocate server resources among multiple peers requesting different files?

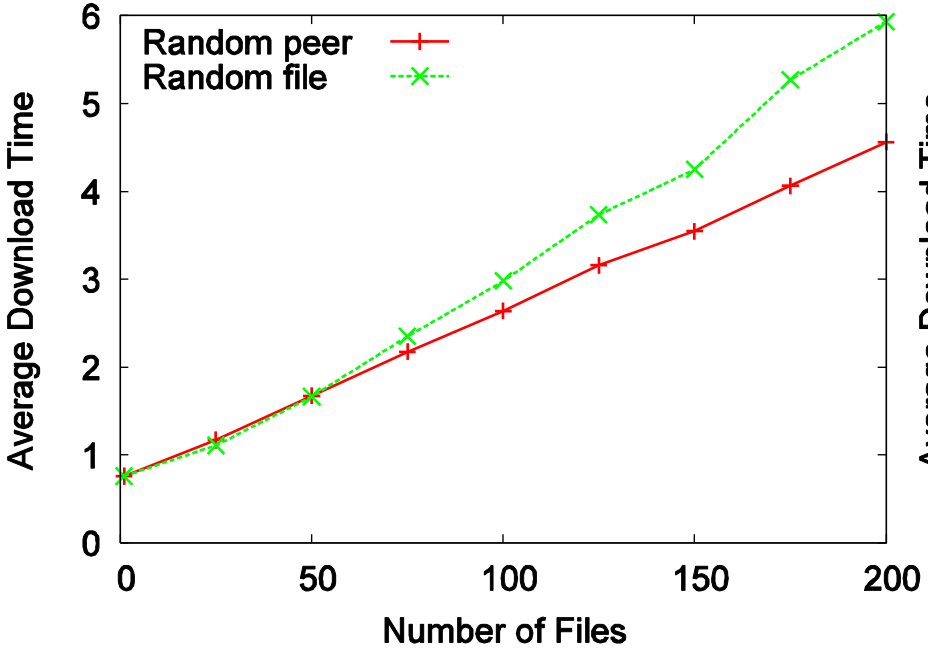
Baseline server scheduling policies:

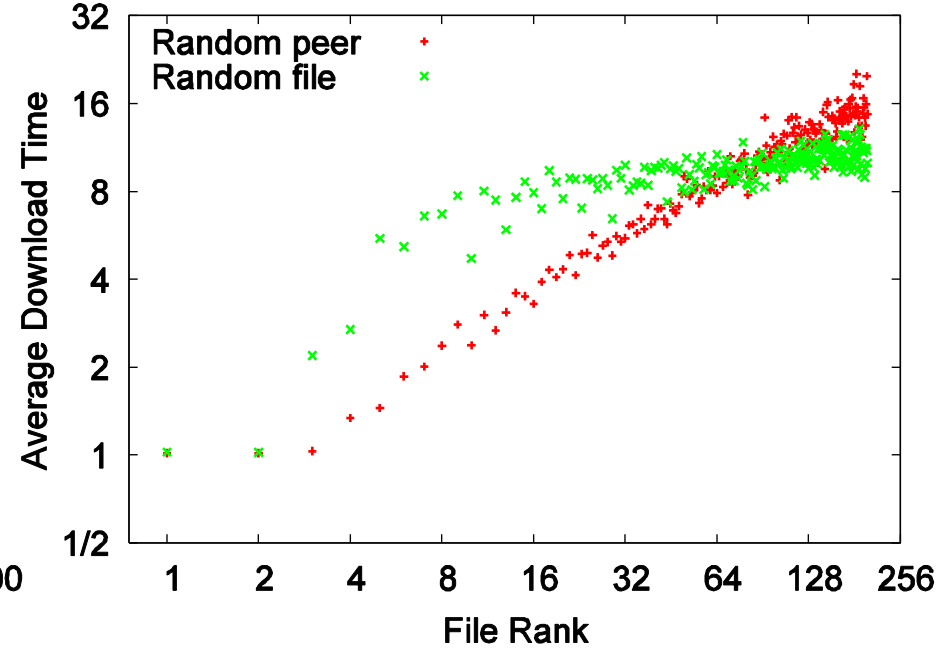
- random peer
- random file

New policies: select file randomly using non-uniform weights

- NEWP (weight by "Number of Excess Wait Peers")
- EW (weight by "Maximum Excess Wait")
- EW x NEWP

Dynamic server scheduling (baseline)

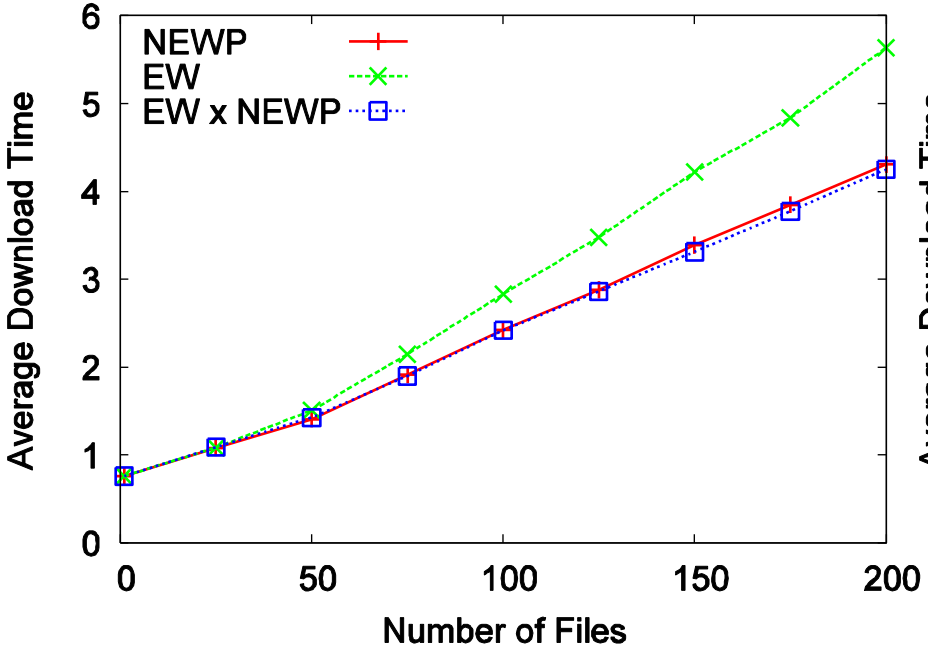


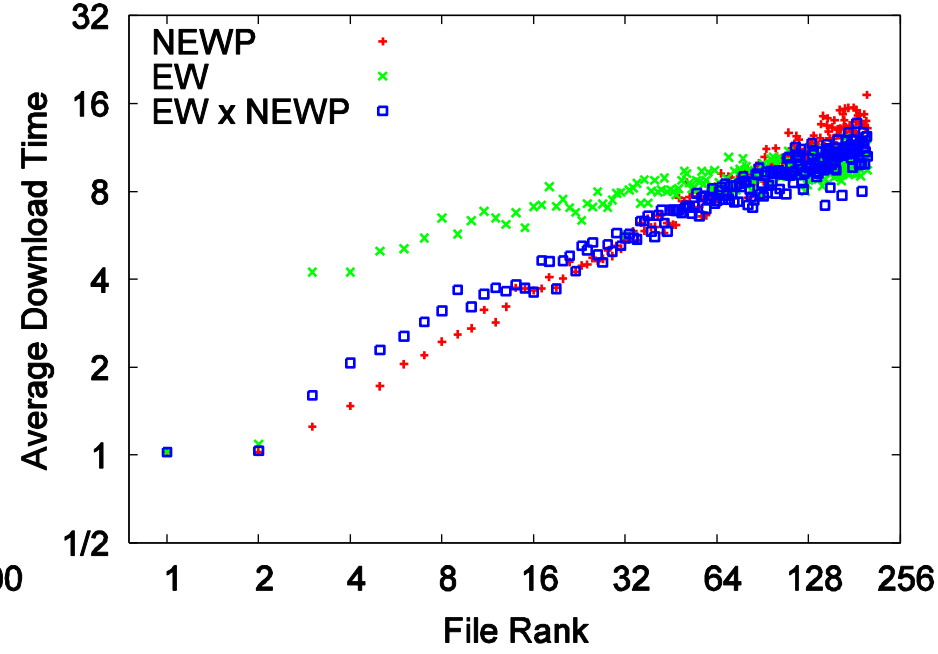


$B = 10$, $\lambda_i = 40/i$, $L = 1$, $U = 1$, $D/U = 3$, $K = 256$

"Random peer" better average delay, "random file" better fairness

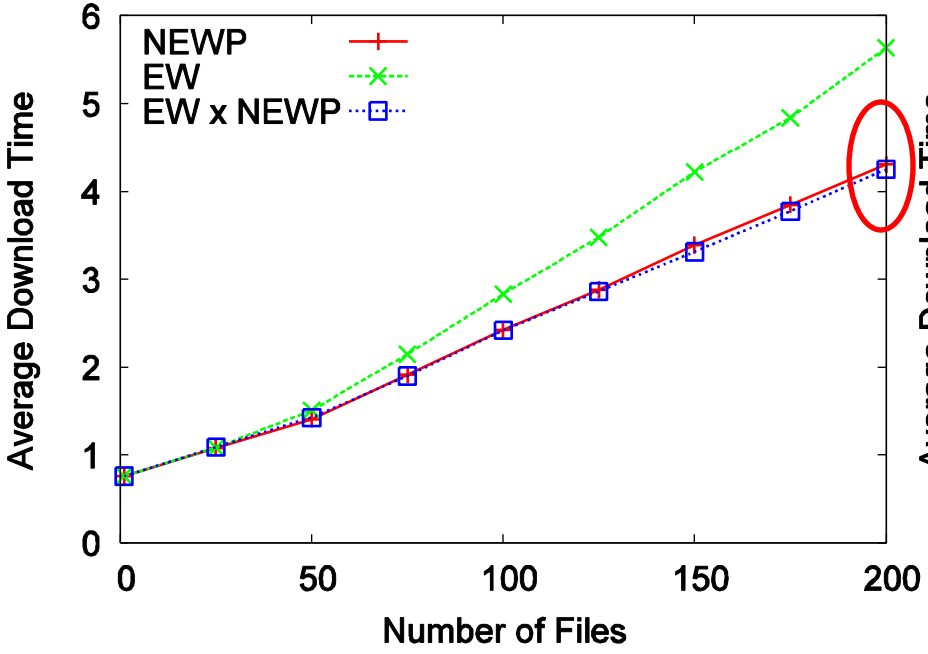
Dynamic server scheduling (priority)

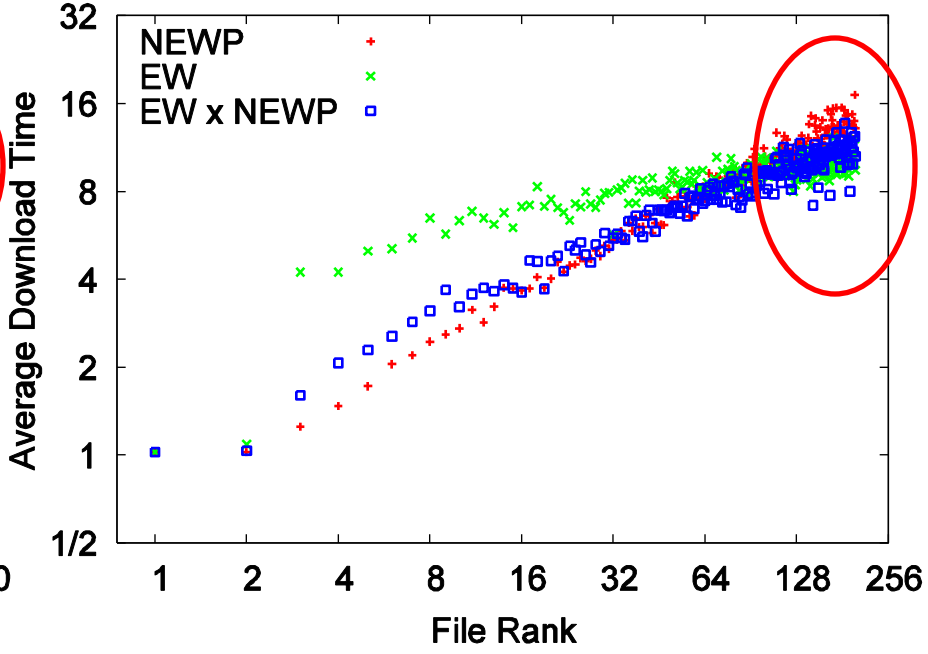




B = 10, λ_i = 40/i, L = 1, U = 1, D/U = 3, K = 256

Dynamic server scheduling (priority)





B = 10, λ_i = 40/i, L = 1, U = 1, D/U = 3, K = 256

Some improvements, but minor …

Torrent inflation

Torrent inflation … Using some of the available upload bandwidth from currently downloading peers to "inflate" torrents for files that would otherwise require substantial server b/w

Basic approach … Have each active downloader help out (potentially) in the torrent for a second file ("inflation file")

Wish list …

- Minimal overhead, as measured by number of blocks a peer downloads from its inflation file
- Harvest only peer upload bandwidth that could otherwise go unused within the peer's own torrent
- Apply harvested bandwidth "effectively"

Torrent inflation: File selection

Inflation file selection policies:

- AT ("Random Active Torrent")
- EW x NEWP
- CNP ("Conditional weight by Number of Peers")

Upload priority levels

Uploader file	File type at downloader		
	Requested	Inflation (case 1)	Inflation (case 2)
Requested	1st	3rd	5th
Inflation	2nd	4th	6th

Torrent inflation: Upload priority levels

Upload priority levels

Uploader file	File type at downloader		
	Requested	Inflation (case 1)	Inflation (case 2)
Requested	1st	3rd	5th
Inflation	2nd	4th	6th

Uploader

A

B

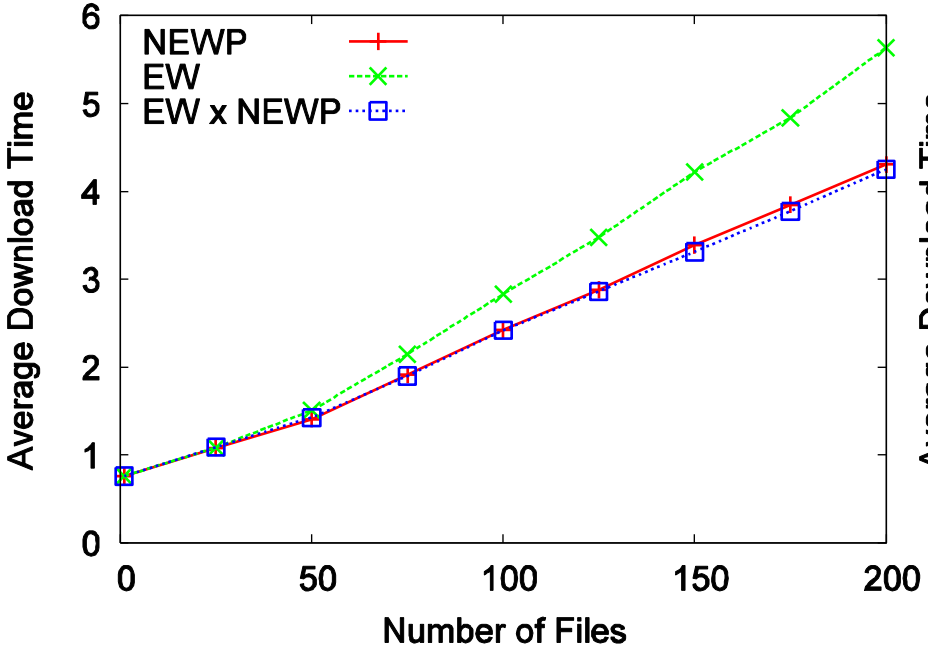
"green" (requested)

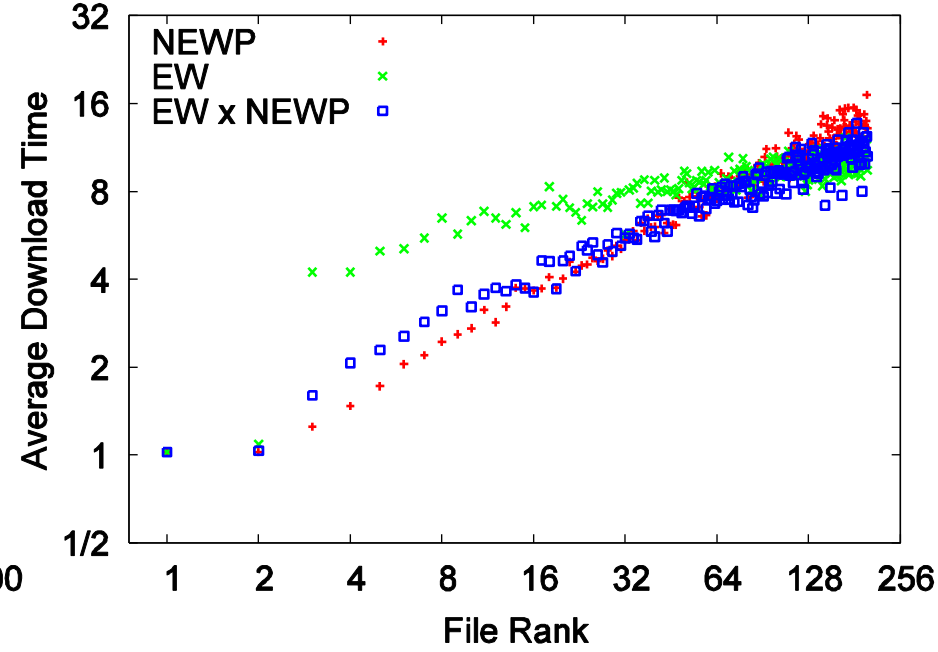
"red" (inflation)

Ranked downloaders

1	2	3	4	5	6
A	A	A		A	
	B B		B		B
C	C	C	C	C	C
		u<1	u<1		

Dynamic server scheduling (priority)

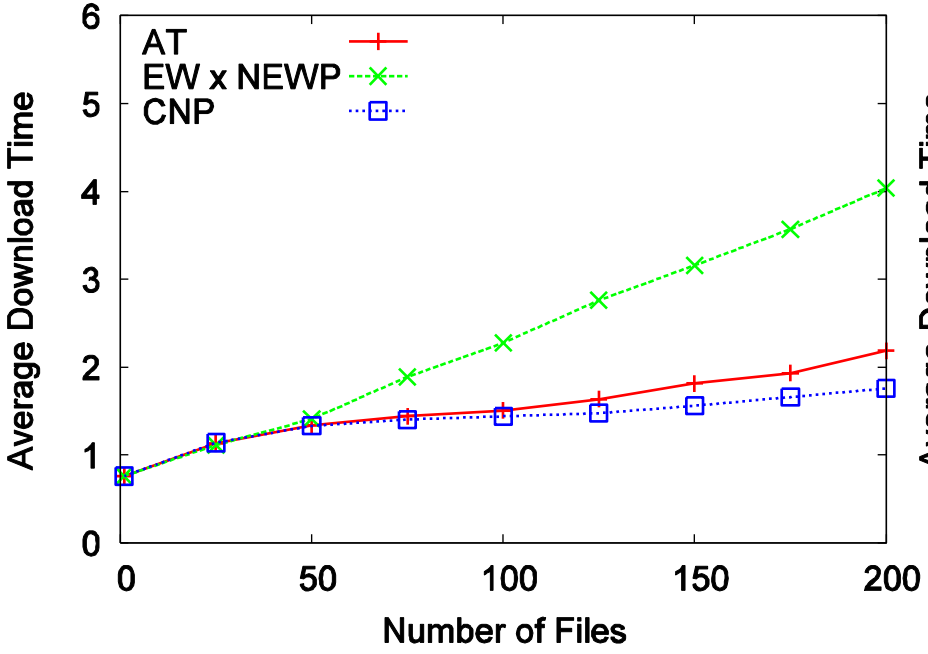


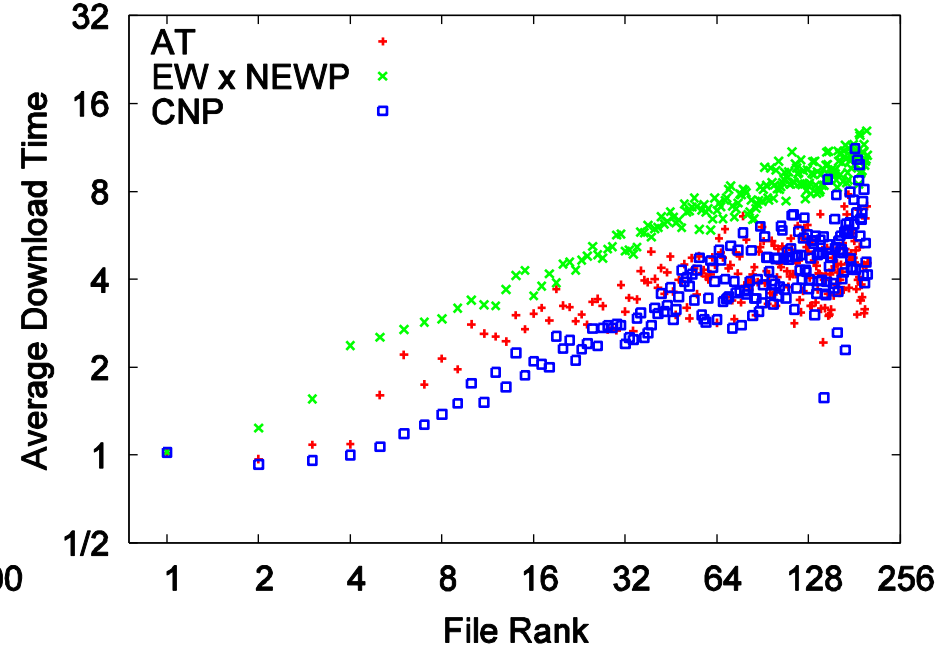


B = 10, λ_i = 40/i, L = 1, U = 1, D/U = 3, K = 256

(Note: Results using ONLY server policies)

Torrent inflation





B = 10, λ_i = 40/i, L = 1, U = 1, D/U = 3, K = 256

Substantial improvements with CNP (best) and AT

Scalability and the "long tail"
Summary and Conclusions

- Peer-assisted system
  - Seeding only at server
- How to efficiently server the "long tail"
  - Much lukewarm/cold niche content
- 1) Dynamic server scheduling
- 2) Torrent inflation
  - Use unused peer bandwidth to "inflate" torrents
  - Making torrents more self-sustaining
- Ongoing work … analytic modeling (for scale)

Questions?

Email: niklas.carlsson@ucalgary.ca