Peer-assisted On-demand Video Streaming with Selfish Peers

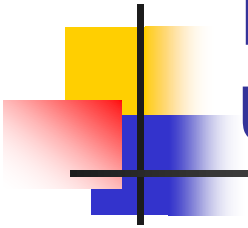
Niklas Carlsson[1] Derek Eager[2] Anirban Mahanti[3]

[1] University of Calgary, Canada

[2] University of Saskatchewan, Canada

[3] NICTA, Australia

IFIP Networking 2009, Aachen, Germany, 2009



Peer-assisted Content Delivery
Using BitTorrent-like Systems

- Second generation file-sharing protocol
  - Effectively serve many concurrent clients
- Files split into many small pieces
- Pieces are downloaded from
  - Leechers (partial) and seeds (full copy)
  - Server(s)
- Distribution paths are dynamically determined
  - Based on data availability
- On-demand Streaming
  - Allow playback to begin well before the entire file is retrieved

Download using BitTorrent
Incentive and Piece Selection

- Incentive: Rate-based tit-for-tat policy
  - Establish connections to large set of peers
  - Leechers: Upload preference to the leechers that provide the highest download rates
    - (n - 1) unchoked based on download rate
    - 1 optimistically unchoked (random peer)
- Piece selection: Rarest first
  - Request the piece that is the rarest among the set of pieces that the peer's neighbors have (and the peer itself needs)
  - Achieves high piece diversity

On-demand Streaming
Using BitTorrent-like Systems (2)

- Greedy peers
  - Require incentive : Peers are motivated to upload data to others owing to the likely beneficial impact on their own performance
- Mediates the conflict between the goals of
  - Low start-up delay and consistently on-time piece delivery
    - Motivates piece delivery that is more "in-order"
  - The requirements of effective tit-for-tat
    - Motivates delivery that is more "rarest first"

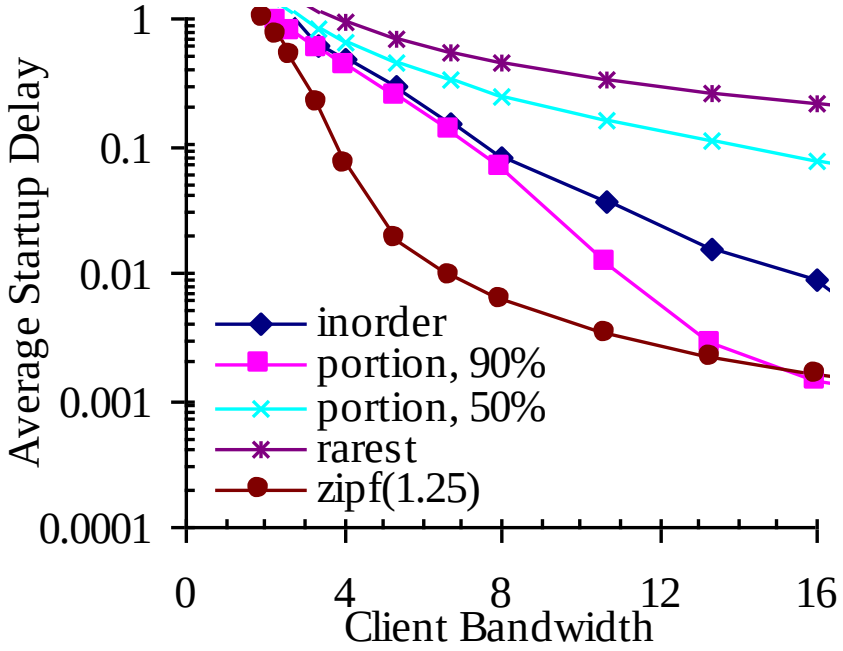
On-demand Streaming
using BitTorrent-like Systems (3)

- (basic) Protocols has three components
  - 1) Piece selection policy
    - Determines which piece to download
  - 2) Start-up rule
    - Determines when playback can safely commence
  - 3) Peer selection policy
    - Determines which peer(s) to upload to

Baseline Protocol
Piece Selection (1)

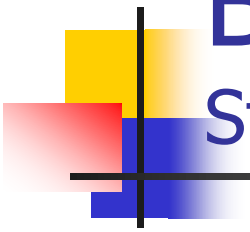
- Which piece to upload?
- Basic tradeoff
  - Piece diversity
    - Tit-for-tat is most effective with "rarest-first"
  - In-order requirements
    - Streaming is most natural using "in-order"
- Baseline policy (from '07 paper)
  - Simple probabilistic policy
    - Bias towards earlier pieces
  - Zipf(θ)

Piece Selection Policy
Example Results



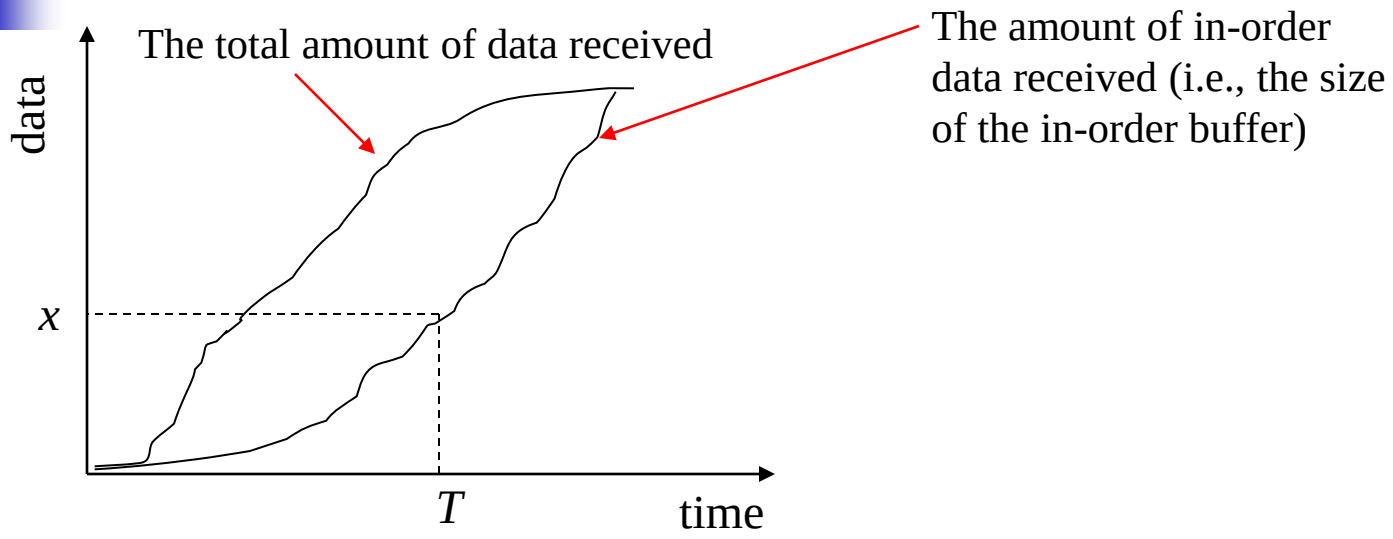
- Inorder, Rarest
- Portion, x%
  - x% inorder
  - (100-x)% rarest
- Zipf(θ)
  - Random with bias towards earlier pieces
  - Bias follow Zipf distribution

Baseline protocol
Start-up Rule



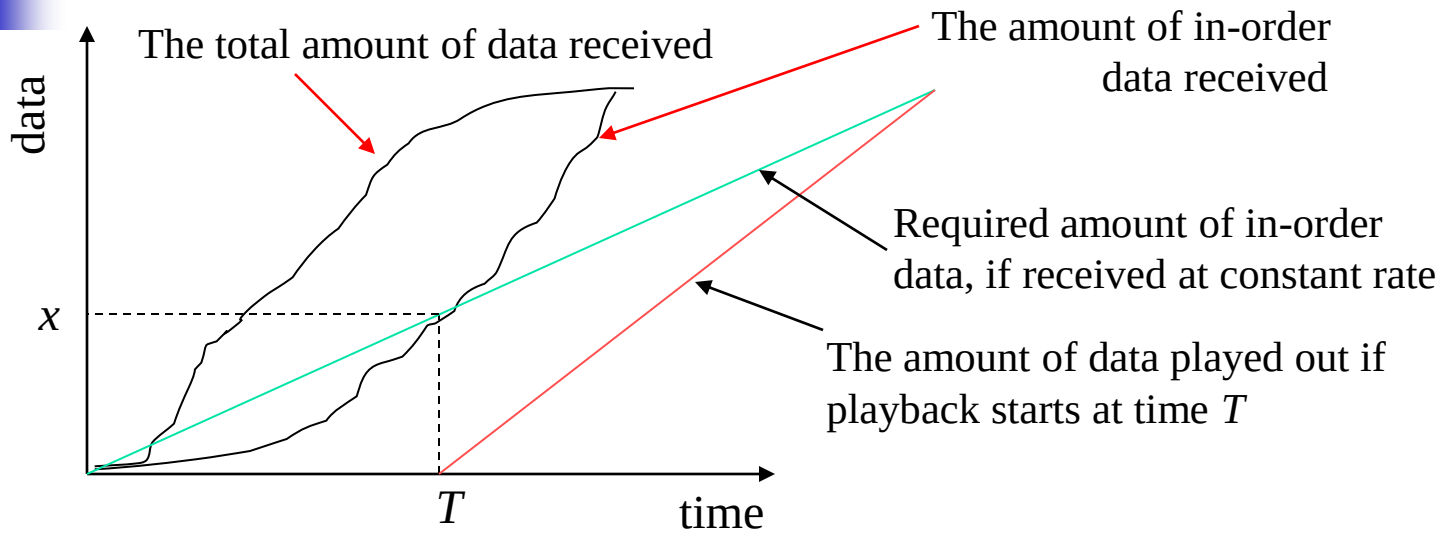
- When to commence playback?
  - Without significant chance of playback interruption
- Simple rule based on
  - At least retrieved the first two segments
    - Initial buffering: less likely falling behind
    - Get reasonable rate estimate
  - Rate (estimate) in-order pieces are retrieved
    - Sufficient to allow playback to begin (if that rate was to be maintained)

Start-up Rule
Intuition



- In-order buffer
  - Contains all pieces up to the first missing piece
- The rate the size of the in-order buffer increases is expected to increase with time (as holes are filled)

Start-up Rule
Intuition



- Estimate the rate using a "long-term" average (LTA)
  - Adjusts start-up delay based on network conditions, allowing it to maintain a small number of late pieces
  - Initial buffer
  - Enough (in-order) pieces to get a reasonable rate estimate

Baseline protocol
Peer Selection

- Which peer to upload to?
- Baseline policy (basic BitTorrent policy)
  - Server unchoke rule
    - Random peer
  - Peer unchoke rule
    - Rate based tit-for-tat
    - Optimistic unchoke is done using random peer
- Design Goals
  - Do not want to alter tit-for-tat (used by peers)
  - High piece sharing efficiency (efficient tit-for-tat)
  - Low start-up delay
  - No/few late pieces

Acquiring Rare Pieces
Peer Selection

- Previous protocols
  - Rely on older peers uploading to "new" peers
    - Including baseline protocol
  - "New" peers typically do not have anything to offer in exchange
  - Relatively long time until "new" peers acquire rare pieces
- Want to allow for quicker dissemination of "rare" pieces
- Proposed policy
  - Rare Piece delivery to New Peers (RPNP)

Acquiring Rare Pieces
Rare Piece delivery to New Peers (RPNP)

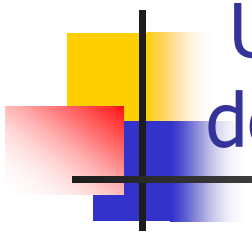
- 1) When server unchokes a "new" peer (that has not yet begun playback)
  - Upload the rarest piece that is not currently being uploaded
  - Ties are broken randomly **except** when
    - only the server has these pieces, **or**
    - the server can serve every active peer at the play rate
  - In these cases Zipf is used to break ties
- 2) Server gives upload priority to "new" peers
  - Among such peers, the server gives priority to peers that it has uploaded less data
  - Server only uploads to the *n* peers with the highest priority, with ties broken at random

Prioritize Urgent Piece Downloads
Peer Selection

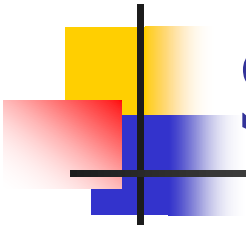
- RPNP is oblivious to if clients have begun playback or not
  - Would like to increase the likelihood that each piece is received by its scheduled playback point
- Proposed policy
  - Urgent piece Prioritization with Rare Piece delivery to New Peers (UP/RPNP)
- Define "Low-buffer" state:
  - have started playback, **and**
  - the next required piece is within either the segment being played back, or the next segment

Prioritize Urgent Piece Downloads

Urgent piece Prioritization with Rare Piece delivery to New Peers (UP/RPNP)



- 1) Peers in the low-buffer state use in-order piece selection (rather than Zipf)
- 2) The server gives the highest upload priority to peers in the low-buffer state
  - Among these peers, priority is given to peers that the server has uploaded less data
  - The remaining peers are prioritized as in RPNP
  - As with RPNP, the server uploads to the n highest priority peers, with ties broken randomly

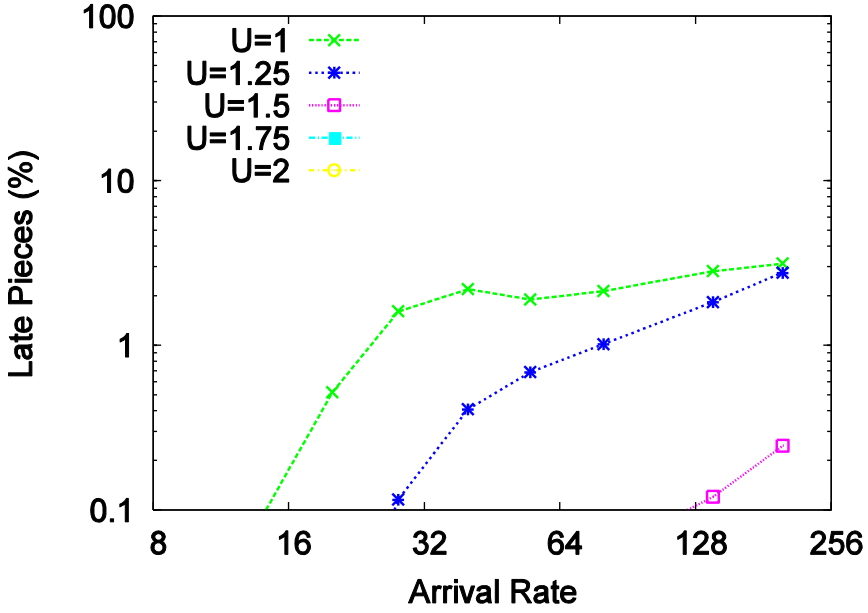


Simulations

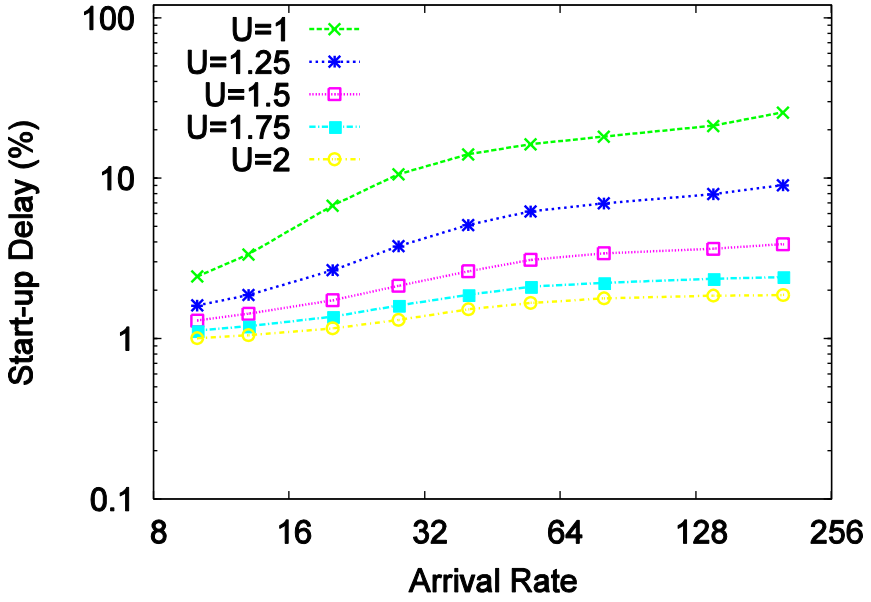
- Assumptions
  - Connection bottlenecks are located at end points
    - Max-min fair bandwidth sharing (e.g., TCP)
  - Single seed; all leechers leave as soon as fully downloaded
- Example Scenarios …
  - Steady state
  - Flash crowd (range: "all at once" $\rightarrow$ steady state)
  - Early departures (churn)
  - Client heterogeneity
  - Free loading scenarios
- Various parameters considered
  - E.g., client bandwidth, peer arrival rate, download/upload bandwidth ratio, server/client bandwidth ratio

Performance Comparison

Example Results: Steady state scenario



(a) Late pieces

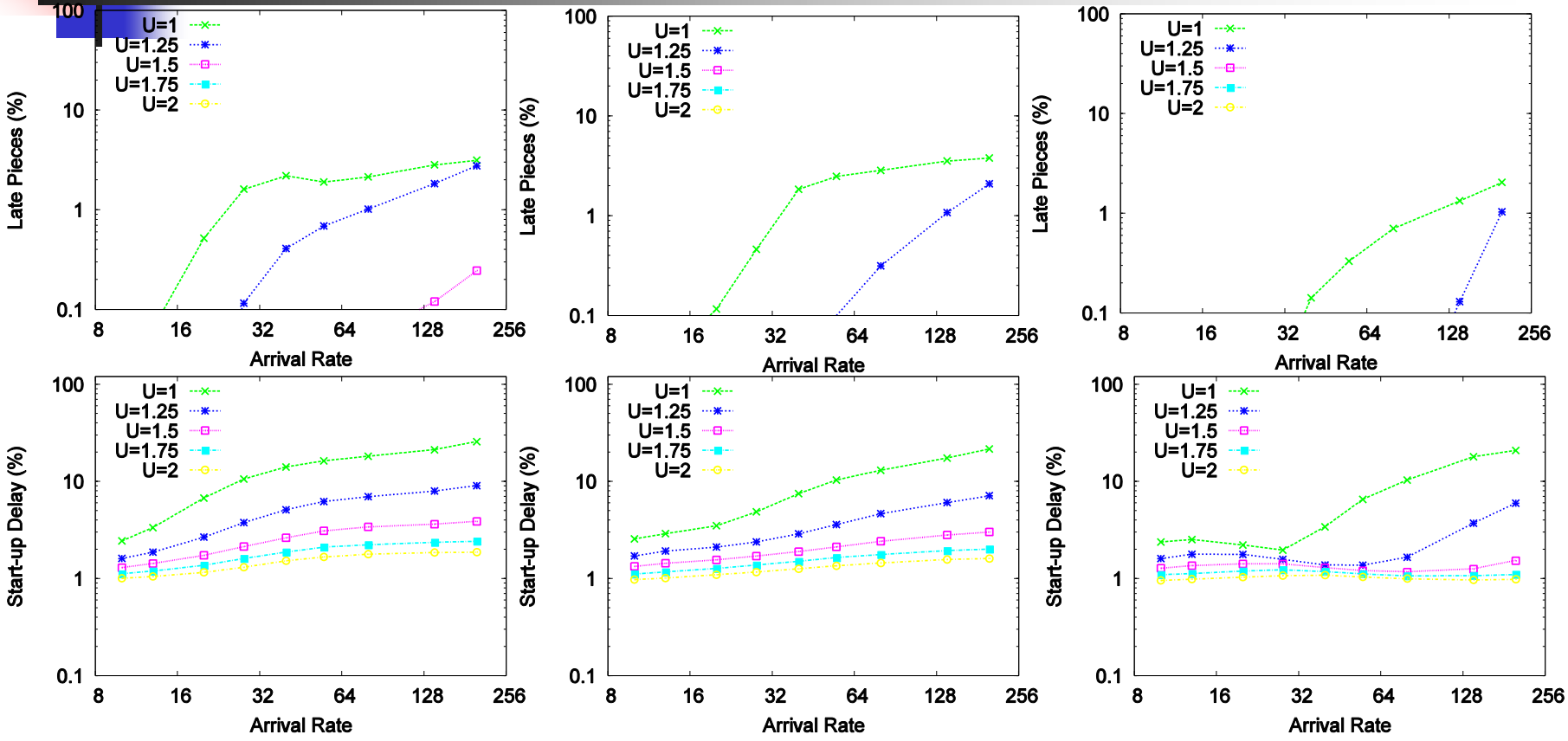


(b) Start-up delay

- Bandwidth requirements: 2-40 times server capacity
- Client upload b/w: 1-2 times the play rate

Performance Comparison

Example Results: Steady state scenario

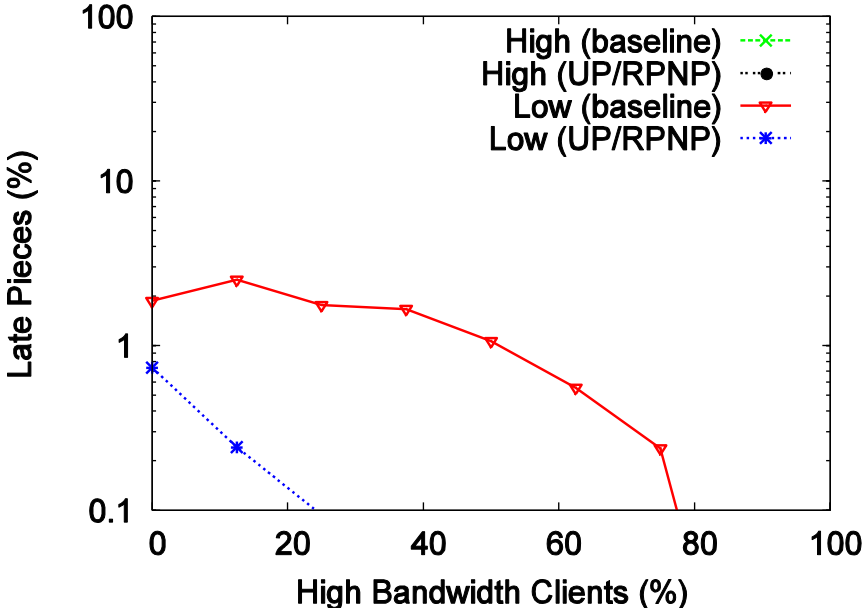


Baseline RPNP UP/RPNP

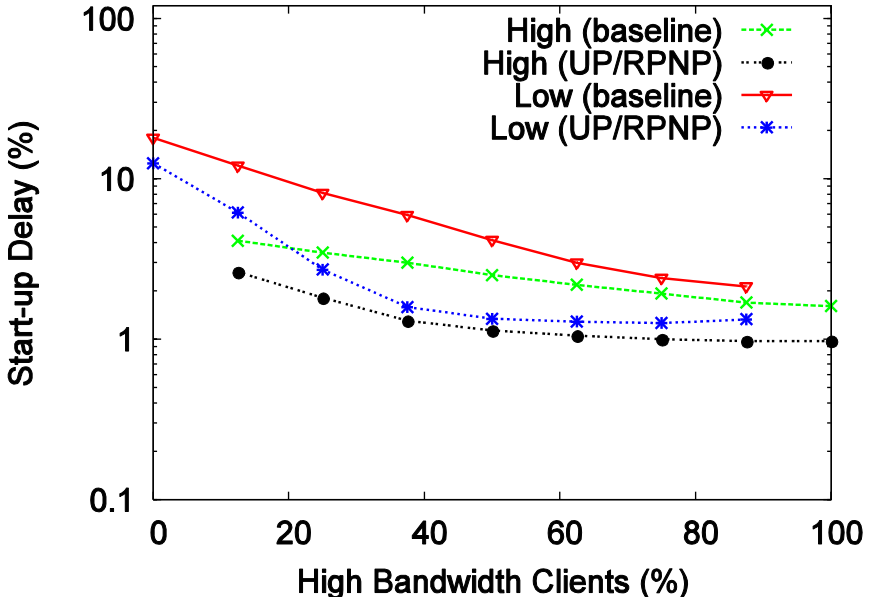
- RPNP: much fever later pieces
- UP/RPNP: additional improvements in both late pieces and delay

Performance Comparison

Example Results: Heterogeneous scenario



(a) Late pieces

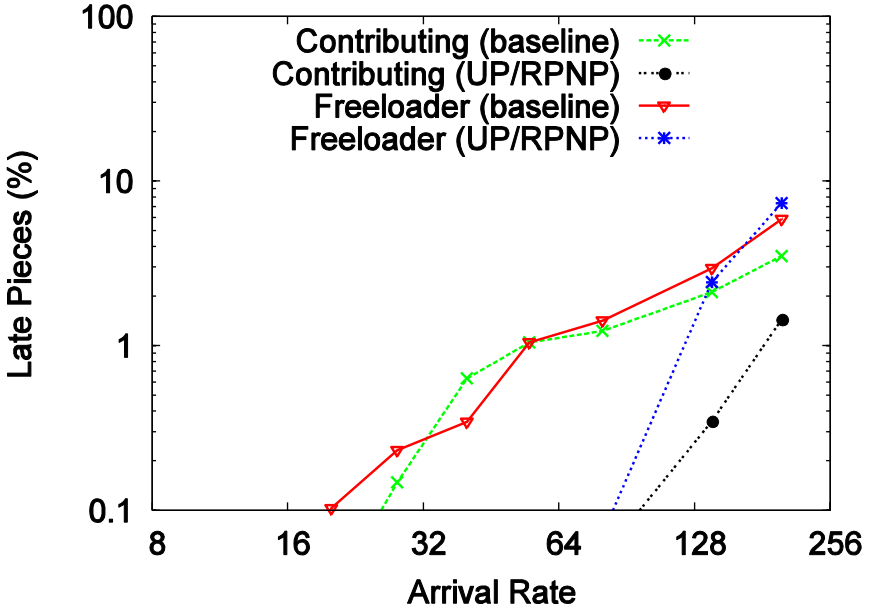


(b) Start-up delay

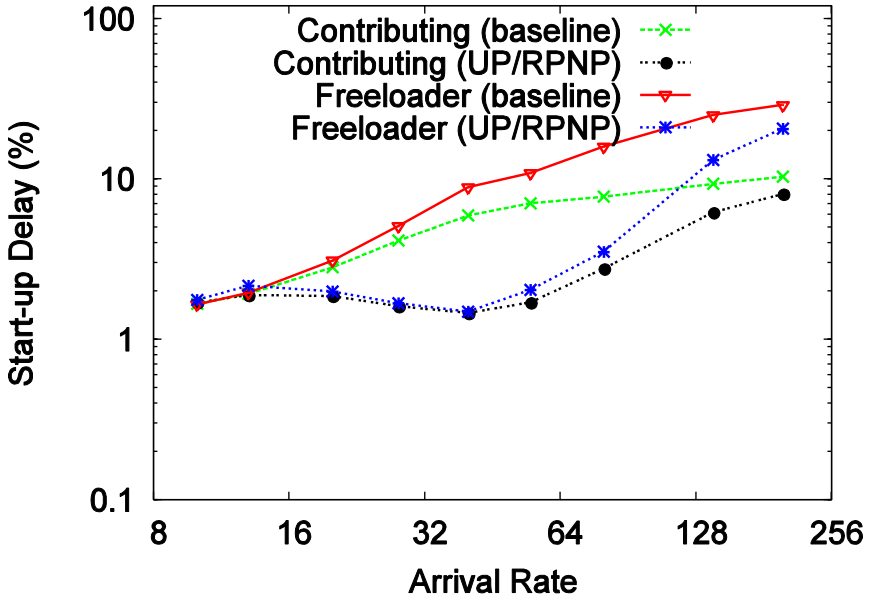
- For both Baseline and UP/RPNP, high bandwidth peers achieve both:
  - Fever late pieces
  - Smaller start-up delay

Performance Comparison

Example Results: Freeloader scenario



(a) Late pieces

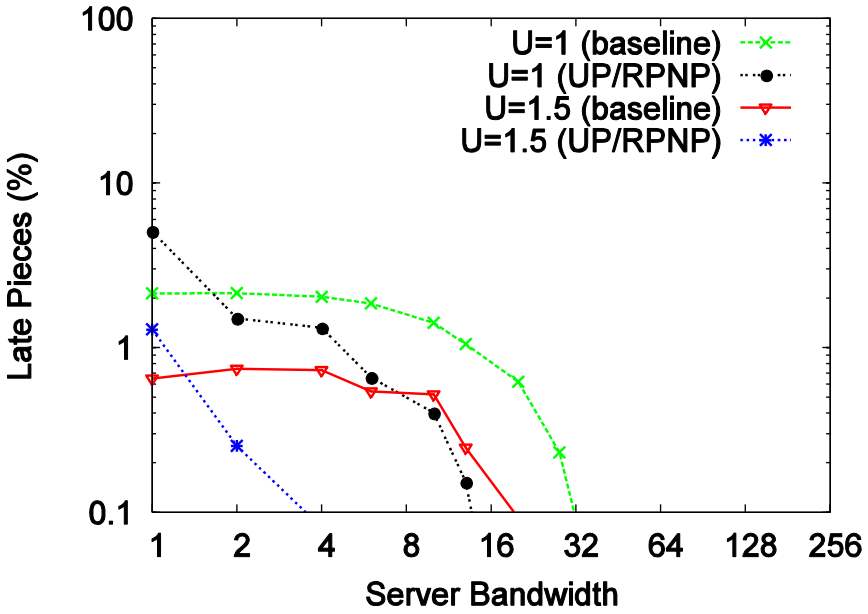


(b) Start-up delay

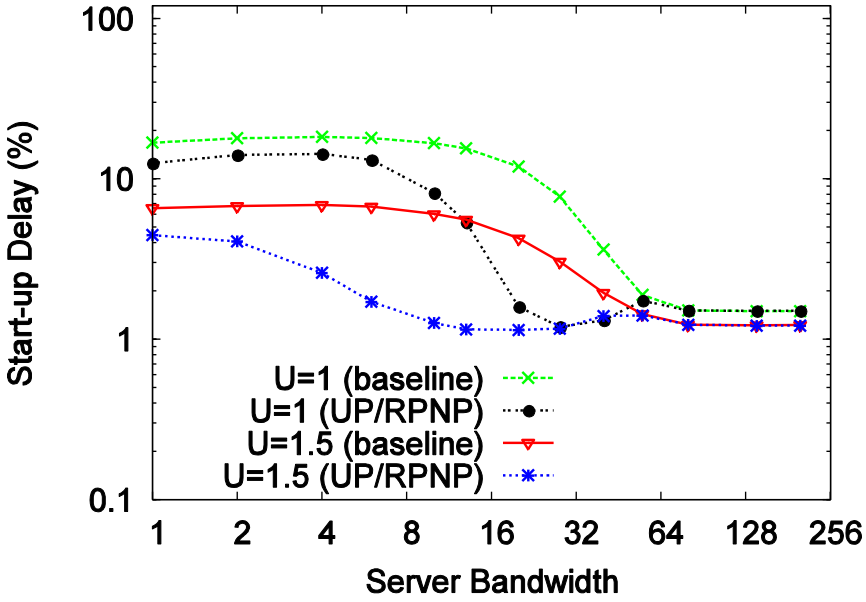
- UP/RPNP achieve significant improvements in both late pieces and start-up delay
- Both protocols achieve significant discrimination against freeloaders

Performance Comparison

Example Results: Impact of upload capacity

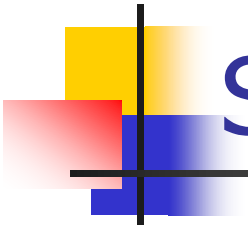


(a) Late pieces



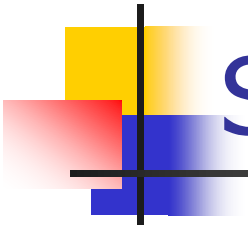
(b) Start-up delay

- Improvements as upload resources increases



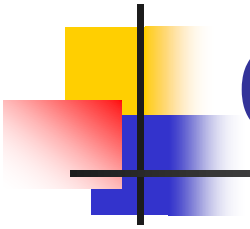
Summary and Conclusions (1)

- Devised BitTorrent-like VoD streaming protocol
- Tit-for-tat compatible policies
  - Peers are motivated to upload data to others owing to the likely beneficial impact on their own performance
- Mediates the conflict between the goals of
  - 1) Low start-up delay and consistently on-time piece delivery (motivates "in-order"), and
  - 2) Effective tit-for-tat (motivates "rarest first")



Summary and Conclusions (2)

- Server (for which tit-for-tat is not an issue), gives upload preference to
  - 1) Peers at imminent risk of receiving data too late for playback
  - 2) Upload of rare pieces to newly arrived peers
- Performance evaluation shows that
  - Our policies are able to provide substantial improvements in quality of service while ensuring that the piece diversity is sufficient for peers to effectively employ tit-for-tat



Questions…

?

Contact information:
Niklas Carlsson; carlsson@cs.usask.ca
Derek L. Eager; eager@cs.usask.ca
Anirban Mahanti; anirban.mahanti@nicta.au