Efficient Algorithms for Multi-sender Data Transmission in Swarm-based Peer-to-Peer Streaming Systems

Yuanbin Shen

Thesis Defense for Master of Science

School of Computing Science
Simon Fraser University

August 17, 2010

Outline

Outline

Peer-to-Peer (P2P) Streaming Systems

- A system to deliver streaming contents using P2P technology
- Example: PPLive, CoolStreaming, SopCast

- Design alternatives:
  - Live and On-demand Streaming
  - Tree-based and Swarm-based Structures
  - Push-based and Pull-based Protocols
  - Peer Matching: Random, ISP-aware, etc.
  - Segment Transmission Scheduling

Peer-to-Peer (P2P) Streaming Systems

- A system to deliver streaming contents using P2P technology
- Example: PPLive, CoolStreaming, SopCast
- Design alternatives:
  - Live and On-demand Streaming
  - Tree-based and Swarm-based Structures
  - Push-based and Pull-based Protocols
  - Peer Matching: Random, ISP-aware, etc.
  - Segment Transmission Scheduling

Outline

Problem Statement - Segment Transmission Scheduling

Transmit a video stream from multiple senders to a receiver

- The stream is divided into segments
- Segments have different sizes and deadlines
- Senders have different bandwidths
- Senders may or may not hold a copy of a segment
- Whole streaming time is divided into sliding windows

Goal: construct schedules for each sliding window to maximize the number of on-time segments

Schedule: specify from which senders to request which segments

Thesis Contributions

We study the Segment Transmission Scheduling (STS) problem:

- Show STS is NP-Complete
- Propose an Integer Linear Programming (ILP) formulation to optimally solve it
- Propose a 2-approximation algorithm to efficiently solve it:
  - Formally analyze its performance and time complexity
  - Evaluate it in both simulations and real experiments
  - Outperforms other algorithms used in current systems

Hardness

Theorem

The segment transmission scheduling problem defined above is NP-Complete.

Proof.

Sketch:
Reduce the NP-Complete parallel machine scheduling problem to the segment transmission scheduling problem □

Existing Algorithms

- Random [Pai et al., IPTPS'05]
- Weighted Round-Robin [Agarwal and Rejaie, MMCN'05]
- Rarest First [Zhang et al., INFOCOM'05]
- Min-Cost Flow Based [MZhang et al., TPDS'09]
- Weighted Segment Scheduling [Hsu and Hefeeda, MMSys'10]

Problem Formulation

$$
\begin{aligned}
z = \quad & \max \sum_{n=1}^{N} \sum_{m=1}^{M} x_{n,m} && \text{(1a)} \\
\text{s.t.} \quad & x_{\hat{n},\hat{m}} \leq a_{\hat{n},\hat{m}} && \text{(1b)} \\
& \sum_{n=1}^{\hat{n}} (s_n / b_{\hat{m}}) x_{n,\hat{m}} \leq d_{\hat{n}} && \text{(1c)} \\
& \sum_{m=1}^{M} x_{\hat{n},m} \leq 1 && \text{(1d)} \\
x_{\hat{n},\hat{m}} \in \{0,1\},\ & \forall\ \hat{n} = 1, 2, \ldots, N \text{ and } \hat{m} = 1, 2, \ldots, M. && \text{(1e)}
\end{aligned}
$$

- (1a): objective - maximize the number of on-time segments
- (1b): always schedule a segment to a sender that holds a copy of it
- (1c): ensure that all assigned segments meet their deadlines
- (1d): avoid assigning a segment to more than one sender

The Proposed SSTF Algorithm - Main Idea

- Main Idea
  - Sort segments increasingly on their sizes
  - Schedule on senders one by one
  - On a single sender:
    - Select a segment that i) has shortest transmission time (smallest size); ii) can arrive on time
    - Remove the scheduled segment, and repeat the above step until no more segments can be scheduled on that sender
  - Go on to schedule on the next sender in the same way
  - Stop when all segments are scheduled or when no more bandwidth left

SSTF - Pseudocode

SSTF: Serialized Shortest Transmission-time First Algorithm

INPUTS:
(i) Segment sizes and deadlines in current scheduling window
(ii) Sender bandwidths and availability information
OUTPUT:
A schedule for each sender: $\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_M$

let $\mathbf{Q}_m = \varnothing$, where $m = 1, 2, \ldots, M$
let $\bar{\mathbf{N}}$ consists of all remaining segments
sort segments increasingly in $\bar{\mathbf{N}}$ on segment size
for $m = 1$ to M // sequentially considers sender m
  let $t = 0$ // consumed transmission time
  foreach segment $n \in \bar{\mathbf{N}}$ // from small to large
    if $a_{n,m} = 1$ and $t + s_n/b_m \leq d_n$
      // segment n is available and can be transmitted on time
      add segment n to $\mathbf{Q}_m$
      remove segment n from $\bar{\mathbf{N}}$
      let $t = t + s_n/b_m$
return $\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_M$

Figure: The proposed approximation algorithm SSTF.

Approximation Factor

Theorem (Approximation Factor)

The SSTF algorithm returns a segment transmission schedule with a factor of at most 2 compared to the optimal solution.

Proof.

- On sender m:
  - Let $\mathbf{S}_m$ and $\mathbf{S}_m^*$ be the schedule produced by the SSTF algorithm and an optimal algorithm, respectively
- For all the senders:
  - Let $\mathbf{S} = \bigcup_{m=1}^{M} \mathbf{S}_m$ and $\mathbf{S}^* = \bigcup_{m=1}^{M} \mathbf{S}_m^*$
  - We can show that $|\mathbf{S}^*| \leq 2\,|\mathbf{S}|$

□

Approximation Factor

Theorem (Approximation Factor)

The SSTF algorithm returns a segment transmission schedule with a factor of at most 2 compared to the optimal solution.

Proof.

- On sender m:
  - Let $\mathbf{S}_m$ and $\mathbf{S}_m^*$ be the schedule produced by the SSTF algorithm and an optimal algorithm, respectively
- For all the senders:
  - Let $\mathbf{S} = \bigcup_{m=1}^{M} \mathbf{S}_m$ and $\mathbf{S}^* = \bigcup_{m=1}^{M} \mathbf{S}_m^*$
  - We can show that $|\mathbf{S}^*| \leq 2\,|\mathbf{S}|$

□

- Average performance is much better than the theoretical worst case
- 2: best approximation factor for this problem so far

Time Complexity

Theorem (Time Complexity)

The SSTF algorithm runs in time $O(MN + N\log N)$*, where* M *is the number of senders and* N *is the number of segments.*

Proof.

- Sorting segments takes $O(N\log N)$
- On each sender, the algorithm scans through the segment list in $O(N)$
- Number of senders: $O(M)$
- So, total time $= O(MN + N\log N)$

□

Outline

Outline

Simulation Setup

- An event-driven simulator written in Java
- Algorithms: RF, MC, SSTF, WSS, and OPT
- Two videos with different characteristics (Terminator 2, SonyDemo)
- Typical bandwidth distributions [Z. Liu et al., ICNP'08]
- 2000 peers (with 1% seeding peers)
- Peers join and leave the system dynamically

Performance Metrics

- Video quality: Average perceived video quality $\alpha = \sum_{n=1}^{N} w_n u_n / N$
- Smoothness: Continuity index $\beta = \sum_{n=1}^{N} u_n / N$
- Fairness: Load balancing factor γ: standard deviation of loads for all scheduling periods on senders

Simulation Results - Overall Comparison

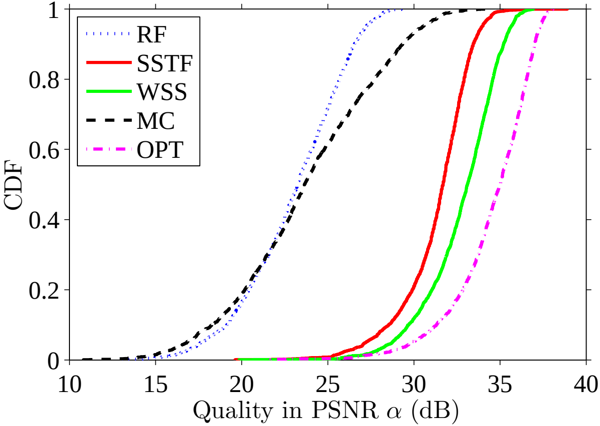

Figure: Overall comparison of the SSTF. RF, WSS, MC, and OPT algorithm. (Terminator 2 video)

Simulation Results - Overall Comparison

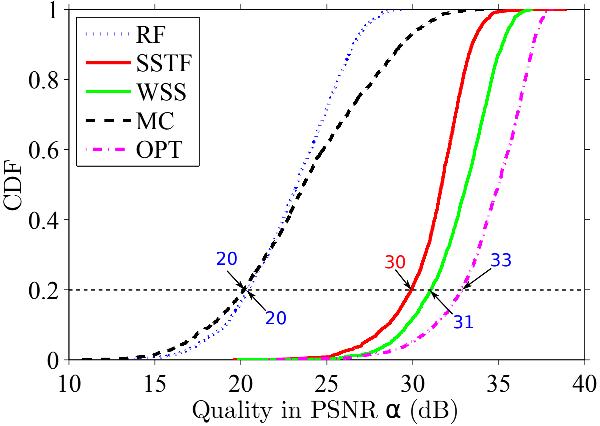

Figure: Overall comparison of the SSTF. RF, WSS, MC, and OPT algorithm. (Terminator 2 video)

Simulation Results - Overall Comparison

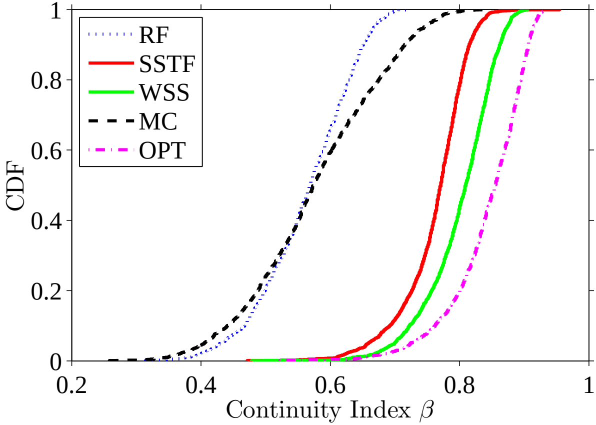

Figure: Overall comparison of the SSTF. RF, WSS, MC, and OPT algorithm. (Terminator 2 video)

Simulation Results - Overall Comparison

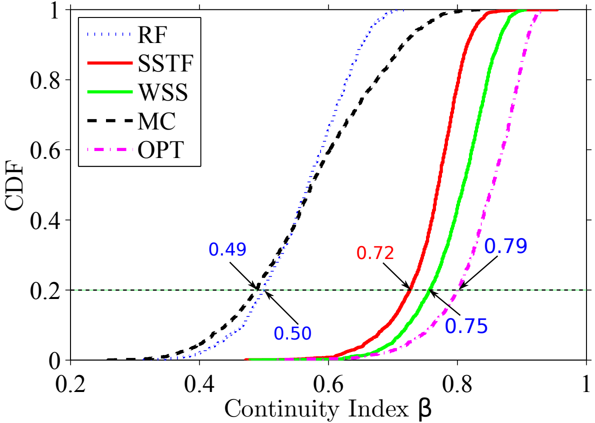

Figure: Overall comparison of the SSTF. RF, WSS, MC, and OPT algorithm. (Terminator 2 video)

Simulation Results - Overall Comparison

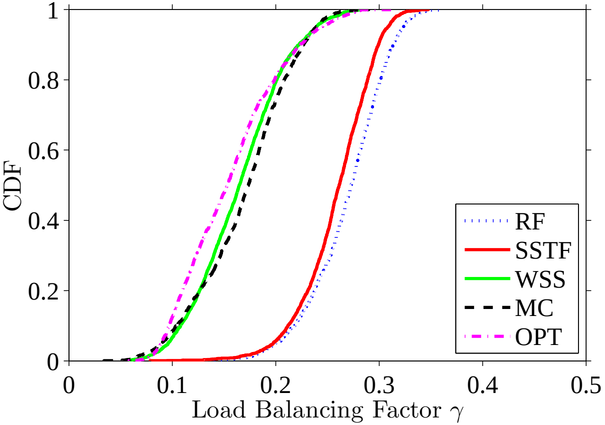

Figure: Overall comparison of the SSTF. RF, WSS, MC, and OPT algorithm. (Terminator 2 video)

Simulation Results - Overall Comparison

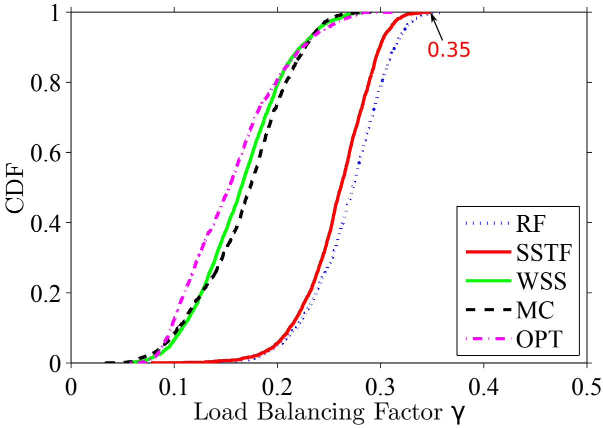

Figure: Overall comparison of the SSTF. RF, WSS, MC, and OPT algorithm. (Terminator 2 video)

Simulation Results - Overall Comparison

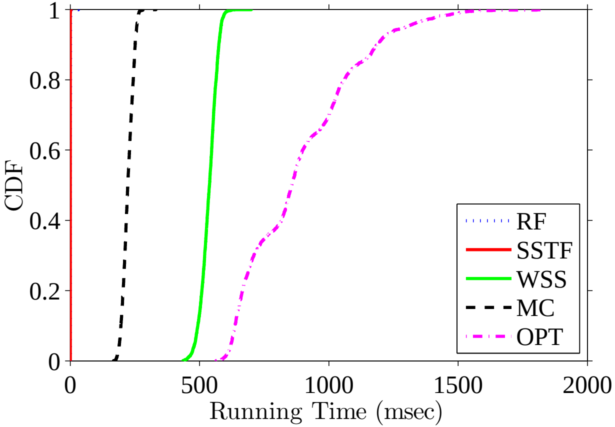

Figure: Overall comparison of the SSTF. RF, WSS, MC, and OPT algorithm. (Terminator 2 video)

Outline

Experiment Setup

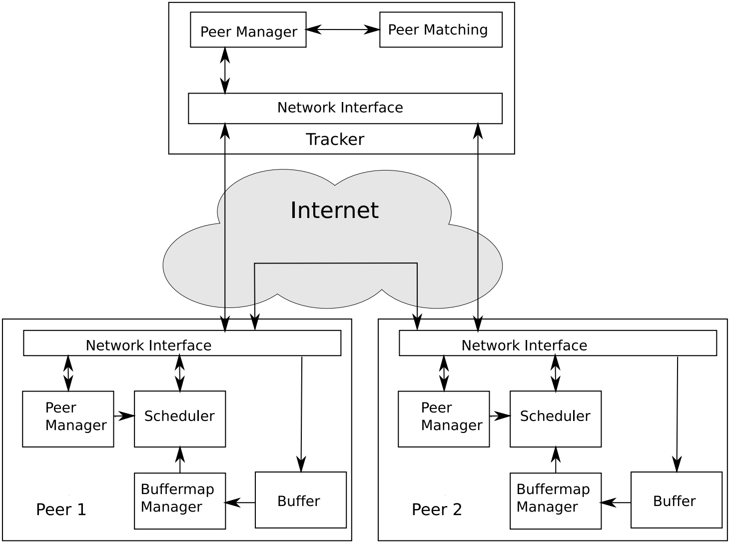

Figure: A high level diagram for the prototype system implementation

Experiment Setup contd.

- Implement the P2P prototype system in Java
- Algorithms: RF, MC, SSTF, WSS
- Use the same videos as in the simulation
- Deploy the prototype on 500 nodes in PlanetLab

Experiment Setup contd.

- Implement the P2P prototype system in Java
- Algorithms: RF, MC, SSTF, WSS
- Use the same videos as in the simulation
- Deploy the prototype on 500 nodes in PlanetLab

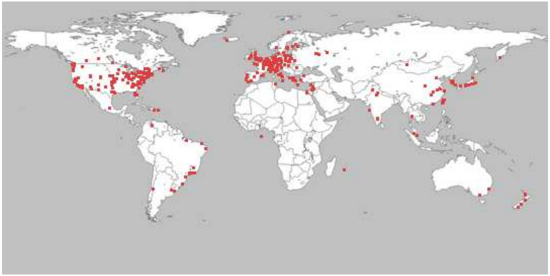

Figure: A snapshot of PlanetLab nodes distribution (http://www.planet-lab.org/generated/World50.png)

Experiment Results

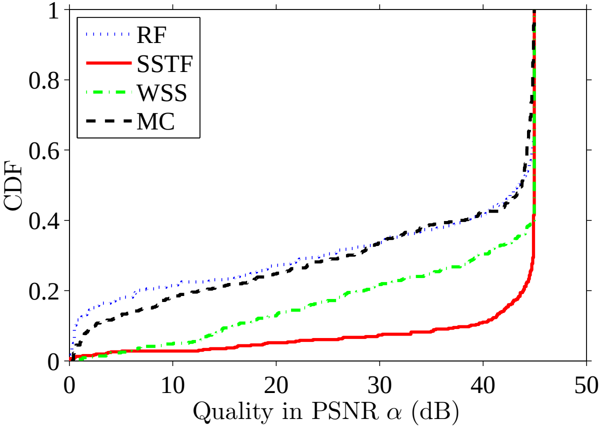

Figure: Overall comparison in PlanetLab-based experiments. (SonyDemo video)

Experiment Results

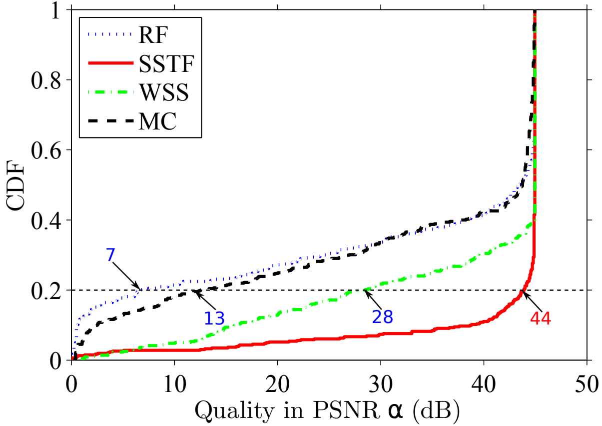

Figure: Overall comparison in PlanetLab-based experiments. (SonyDemo video)

Experiment Results

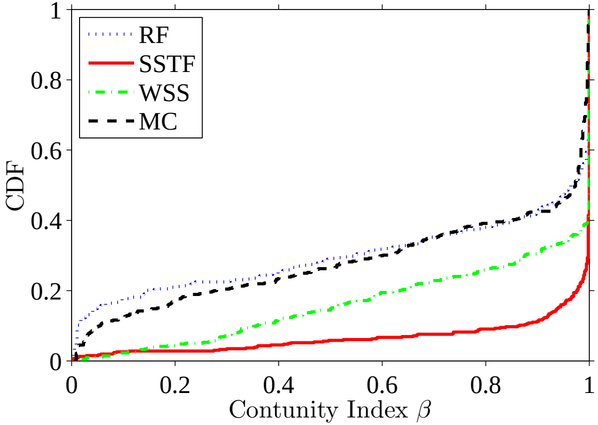

Figure: Overall comparison in PlanetLab-based experiments. (SonyDemo video)

Experiment Results

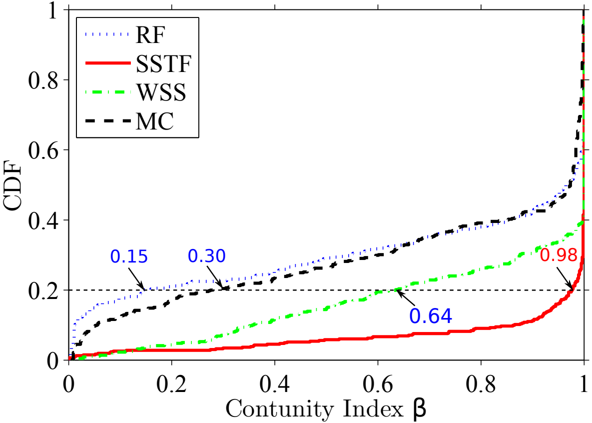

Figure: Overall comparison in PlanetLab-based experiments. (SonyDemo video)

Outline

Conclusions and Future Work

- Conclusions:
  - Studied the segment transmission scheduling problem in P2P video streaming systems
  - Hardness and optimal solution using Integer Linear Programming
  - A 2-approximation algorithm to efficiently solve it; simulation and experimental results show that it:
    - runs very fast
    - Outperforms other algorithms used in current systems
    - Is very close to the optimal solution

Conclusions and Future Work

- Conclusions:
  - Studied the segment transmission scheduling problem in P2P video streaming systems
  - Hardness and optimal solution using Integer Linear Programming
  - A 2-approximation algorithm to efficiently solve it; simulation and experimental results show that it:
    - runs very fast
    - Outperforms other algorithms used in current systems
    - Is very close to the optimal solution
- Future Work
  - Scheduling algorithms for scalable video streams with guaranteed performance
  - Interaction of the proposed algorithm with other parts of the system

Thank You

Questions?