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

1. Introduction

2. Segment Transmission Scheduling

3. Evaluations
   - Simulation
   - Experiment

4. Conclusions and Future Work
Outline

1 Introduction

2 Segment Transmission Scheduling

3 Evaluations
   - Simulation
   - Experiment

4 Conclusions and Future Work
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

1 Introduction

2 Segment Transmission Scheduling

3 Evaluations
   - Simulation
   - Experiment

4 Conclusions and Future Work
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
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

\[ z = \max \sum_{n=1}^{N} \sum_{m=1}^{M} x_{n,m} \]  \hspace{1cm} (1a)

s.t. \[ x_{\hat{n},\hat{m}} \leq a_{\hat{n},\hat{m}} \]  \hspace{1cm} (1b)

\[ \sum_{n=1}^{\hat{n}} \left( s_{n}/b_{\hat{m}} \right)x_{n,\hat{m}} \leq d_{\hat{n}} \]  \hspace{1cm} (1c)

\[ \sum_{m=1}^{M} x_{\hat{n},m} \leq 1 \]  \hspace{1cm} (1d)

\[ x_{\hat{n},\hat{m}} \in \{0,1\}, \ \forall \ \hat{n} = 1,2,\ldots,N \ \text{and} \ \hat{m} = 1,2,\ldots,M. \]  \hspace{1cm} (1e)

- (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
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: 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: Q₁, Q₂, ..., Q₇

1. let Qₘ = ∅, where m = 1, 2, ..., M
2. let Ń consists of all remaining segments
3. sort segments increasingly in Ń on segment size
4. for m = 1 to M // sequentially considers sender m
5. let t = 0 // consumed transmission time
6. foreach segment n ∈ Ń // from small to large
7. if aₙₘ = 1 and t + sₙ/bₘ ≤ dₙ
8. // segment n is available and can be transmitted on time
9. add segment n to Qₘ
10. remove segment n from Ń
11. let t = t + sₙ/bₘ
12. return Q₁, Q₂, ..., Q₇

Figure: The proposed approximation algorithm SSTF.
**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 $S_m$ and $S^*_m$ be the schedule produced by the SSTF algorithm and an optimal algorithm, respectively.

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

- For all the senders:
  - Let \( S = \bigcup_{m=1}^{M} S_m \) and \( S^* = \bigcup_{m=1}^{M} S_m^* \).
  - We can show that \( |S^*| \leq 2 |S| \).

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

1. Introduction

2. Segment Transmission Scheduling

3. Evaluations
   - Simulation
   - Experiment

4. Conclusions and Future Work
Outline

1. Introduction

2. Segment Transmission Scheduling

3. Evaluations
   - Simulation
   - Experiment

4. Conclusions and Future Work
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 $\gamma$: 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)
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)
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)
Figure: Overall comparison of the SSTF. RF, WSS, MC, and OPT algorithm. (Terminator 2 video)
Outline

1. Introduction
2. Segment Transmission Scheduling
3. Evaluations
   - Simulation
   - Experiment
4. Conclusions and Future Work
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)
Figure: Overall comparison in PlanetLab-based experiments. (SonyDemo video)
Experiment Results

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

1. Introduction
2. Segment Transmission Scheduling
3. Evaluations
   - Simulation
   - Experiment
4. Conclusions and Future Work
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?