Athens University of Economics and Business
Master in Computer Science

Master Thesis

Evaluation of video streaming extensions to BitTorrent

Author: Nikolaos Mplatsas
Supervisor Prof.: George Xylomenos
Co-Supervisor Prof.: George Polyzos

June 2011

Outline

A. Introduction

B. Video Streaming

C. Using BitTorrent for video streaming

D. Related work

E. Thesis contribution

F. BiToS extensions

G. Playback

H. Simulation parameters

I. Experimental results

J. Overall evaluation

K. Future work

L. References

Introduction

- BitTorrent is a peer-to-peer file sharing protocol used for distributing large amounts of data.
- BitTorrent is one of the most common protocols for transferring large files
  - responsible for more than 45-78% of all P2P traffic.
  - accounts for roughly 27-55% of all Internet traffic as of February 2009.
- Resist free-riding
  - Tit-for-Tat policy

Video Streaming

- Video streaming has become increasingly popular in the last few years
  - bandwidth availability
  - lower cost
- Media content proliferation in many different application areas
  - education, entertainment, medical treatment

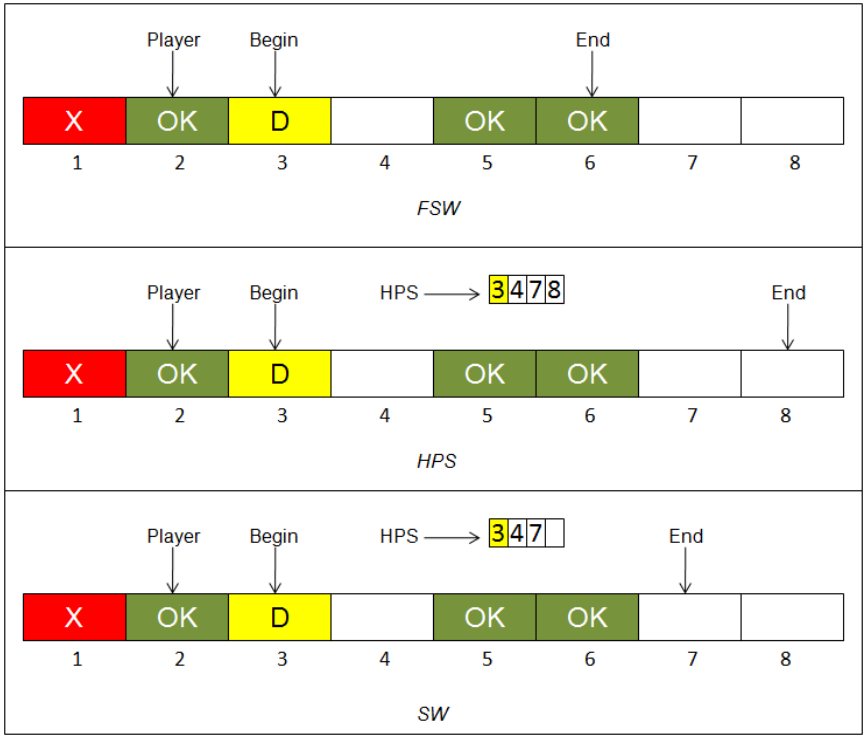
Using BitTorrent for video streaming

- BitTorrent with minimal changes can support streaming.
- BitTorrent vs Centralized solutions
  - we can achieve bandwidth scalability (overcome the bandwidth limitations of keeping the file in a server)
- Streaming-enhanced vs Standard BitTorrent
  - the peer may have the ability of watching the video before the complete download of the file.
  - the peer can evaluate the quality of the video content early and decide if this particular video worth spending time and resources.

Related Work

- P2P Multimedia Streaming using BitTorrent (P. Shah et al., 2007)
  - Sliding window (fixed size)
  - Not requesting pieces outside the window
- BiToS: Enhancing BitTorrent for supporting Streaming Application (A. Vlavianos et al., 2006)
  - High Priority Set
  - Requests pieces from the H.P.S with probability **p** (p=0.8 recommended)
- Windowing BitTorrent for VoD – Not all is lost with Tit-for-Tat (P. Savolainen et al., 2008)
  - Stretching window (adaptive size)
  - Not requesting pieces outside the window

Algorithm	Maximum Window Size	Probability of Requesting from within the Window
Fixed-size Window	w pieces	1
BiToS	w non-arrived pieces	p (typically 0.8)
Stretching Window	w non-arrived pieces or b pieces	1

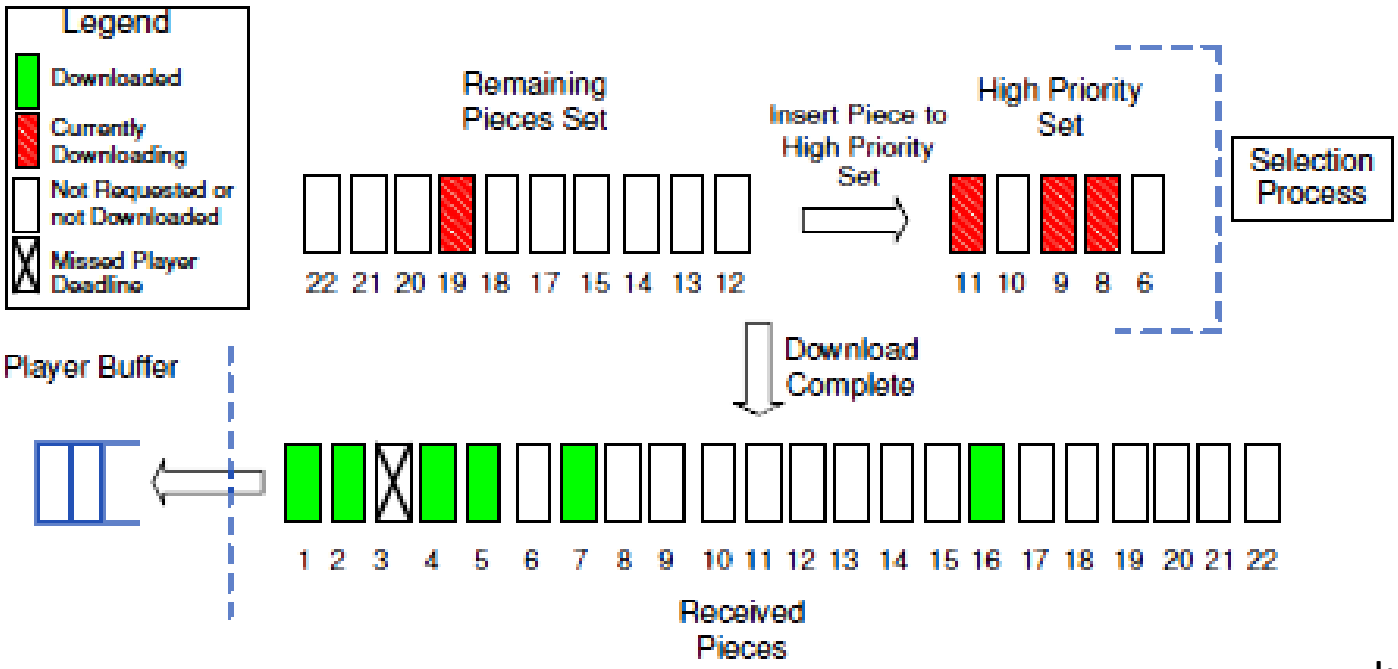


Thesis contribution

- Extended BiToS
  - Probability in each piece inside window
  - Probability in group of pieces inside window
- Evaluate Bitos, Bitos extensions and compare with FSW and SW through two techniques
  - Stall player
  - Player (missed blocks)
- Video Streaming implementation and simulation scenarios were made on a full featured and extensible implementation of BitTorrent for the OMNeT++ simulation environment.

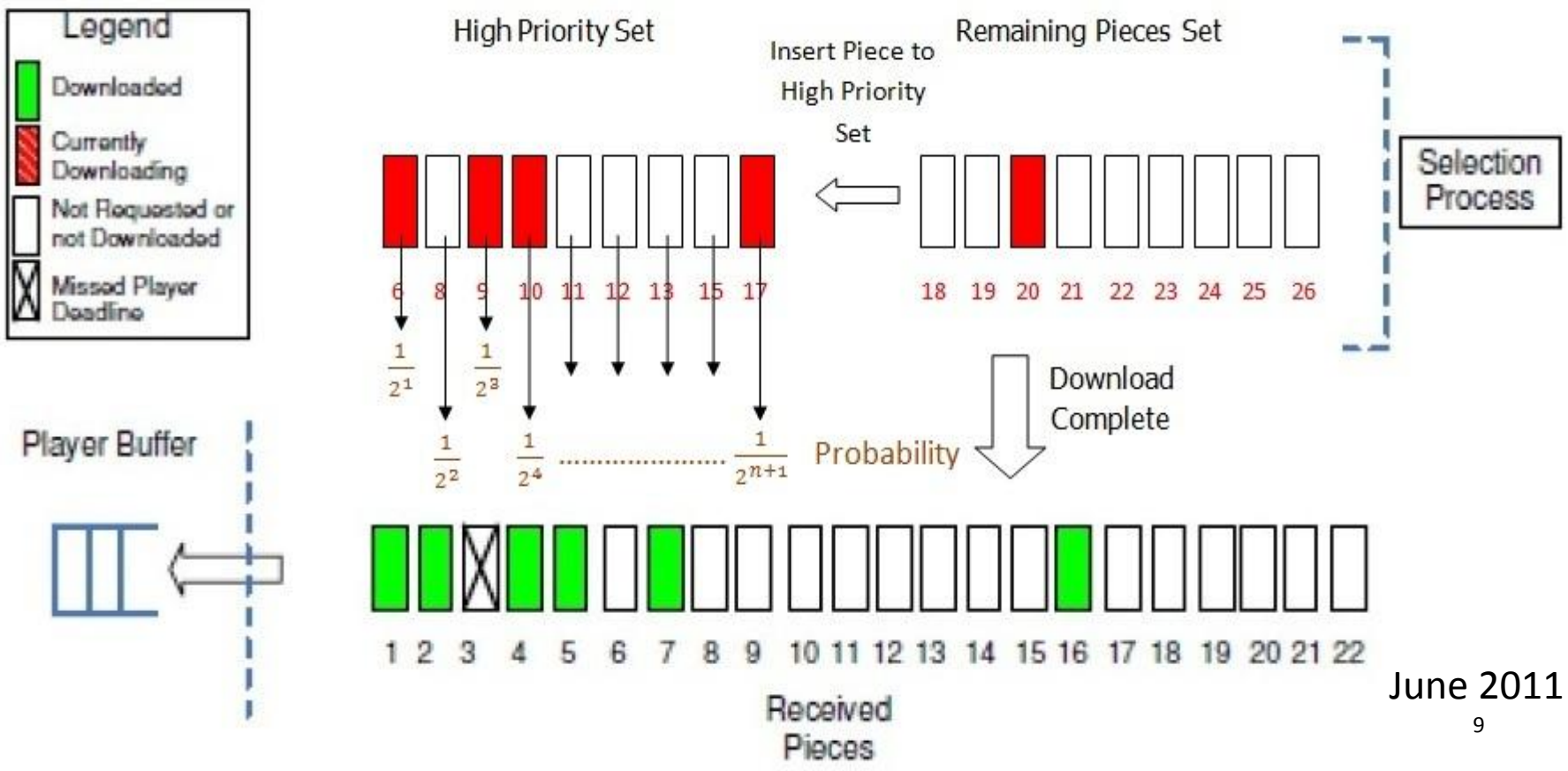
BiToS analysis

- Piece Selection Strategy for Video Streaming
  - Higher download priority to pieces which are close to deadline
  - Rarest first selection, among these *high priority* pieces



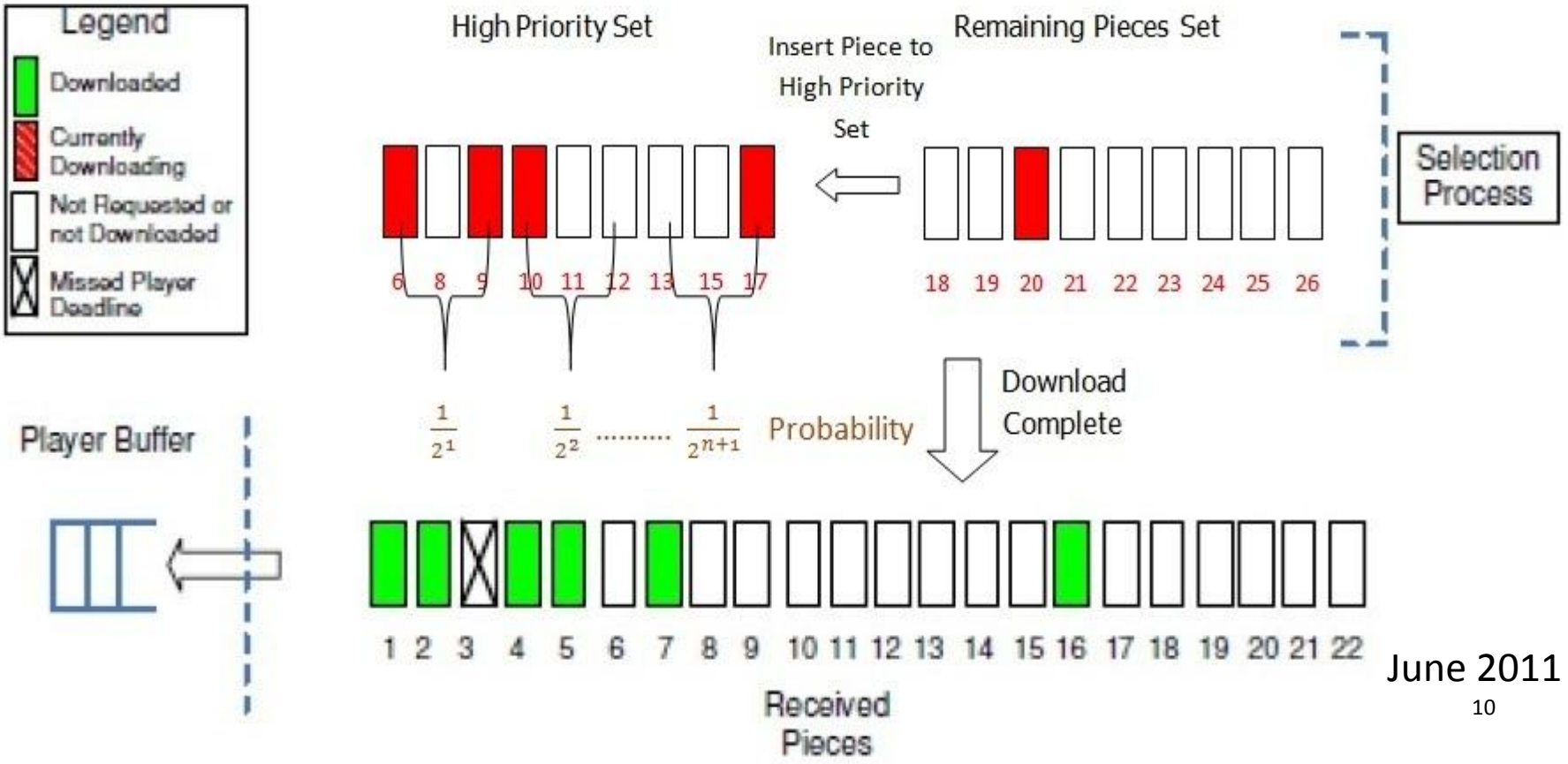
BiToS extension (1/2)

- BiToS with probability in each piece inside the High Priority Set.



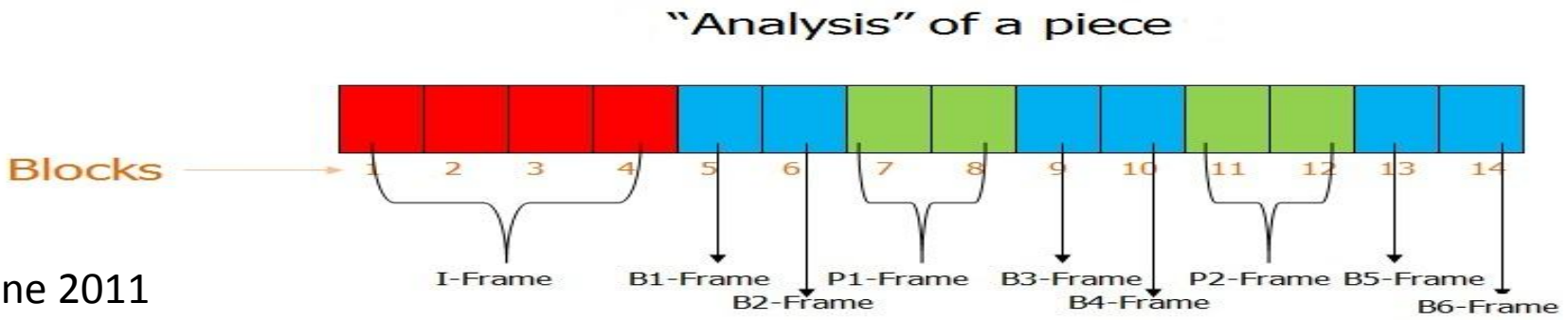
BiToS extension (2/2)

- BiToS with probability in a group of pieces inside the High Priority Set



Playback

- Player
  - Plays the available pieces in sequential order
    - Checks if the next piece in order is available
      - ➢ If not, we work through two techniques
        - 1st technique
          - Player waits until the piece is downloaded
        - 2st technique
          - Player now checks and plays if exists each block of the piece that missed and if it found a missed block it places it in the Missed Block Set
          - Each piece is divided in frames as it appears below



Simulation Parameters (1/2)

- Swarm size
    - Increasing the swarm size may increase the degree of parallelization. This may have positive effects (a peer may download a block faster) or negative effects (there may be more duplicates-redundant data).
- Window size
    - Larger Window size
        - Peers download pieces based on their rareness without considering their deadline
        - The more time it will take on average before the first piece in the window has arrived and the window can move forward
    - Small Window size
        - Peers do not increase the diversity of the pieces (Choked, suffer of overall system efficiency)
    - Tradeoff
        - Window-file size dependent
- Piece size
    - Smaller piece size
        + Shorter initial buffering time
        + Smaller (in KB) and more flexible window size
        - Overhead increase
        - Larger .torrent file

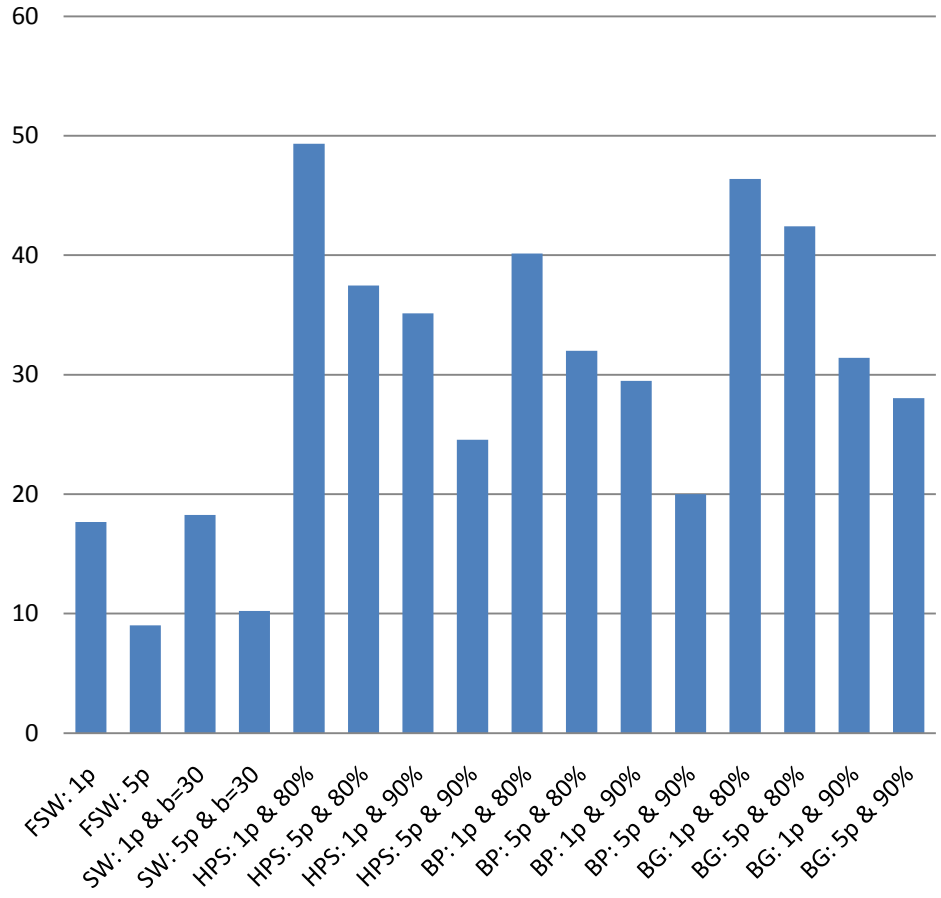
Simulation Parameters (2/2)

- HPS Probability p
  - Large values of p
    - \+ Pieces close to reproduction will be requested earlier than the rest
    - – Peers not choosing rare pieces → choked
- Video Bit Rate
  - Video Quality vs. Piece Loss
  - Player
    - 128KB piece played in 7sec (about 150Kbps)

Parameter	***Value***
Video size (in MB)	200
Video Bit rate (in Kbps)	128
Piece size (in KB)	224
Block size (in KB)	16
Num of pieces for prefetch buffering	1 or 5
Window size (% of total num of pieces)	2% or 8%
Probability p (only for HPS)	80% or 90%
Bound (only for SW)	30 or 100
Player Mode	1,2 or 3
Num of Pieces in group (only for HPS)	25%

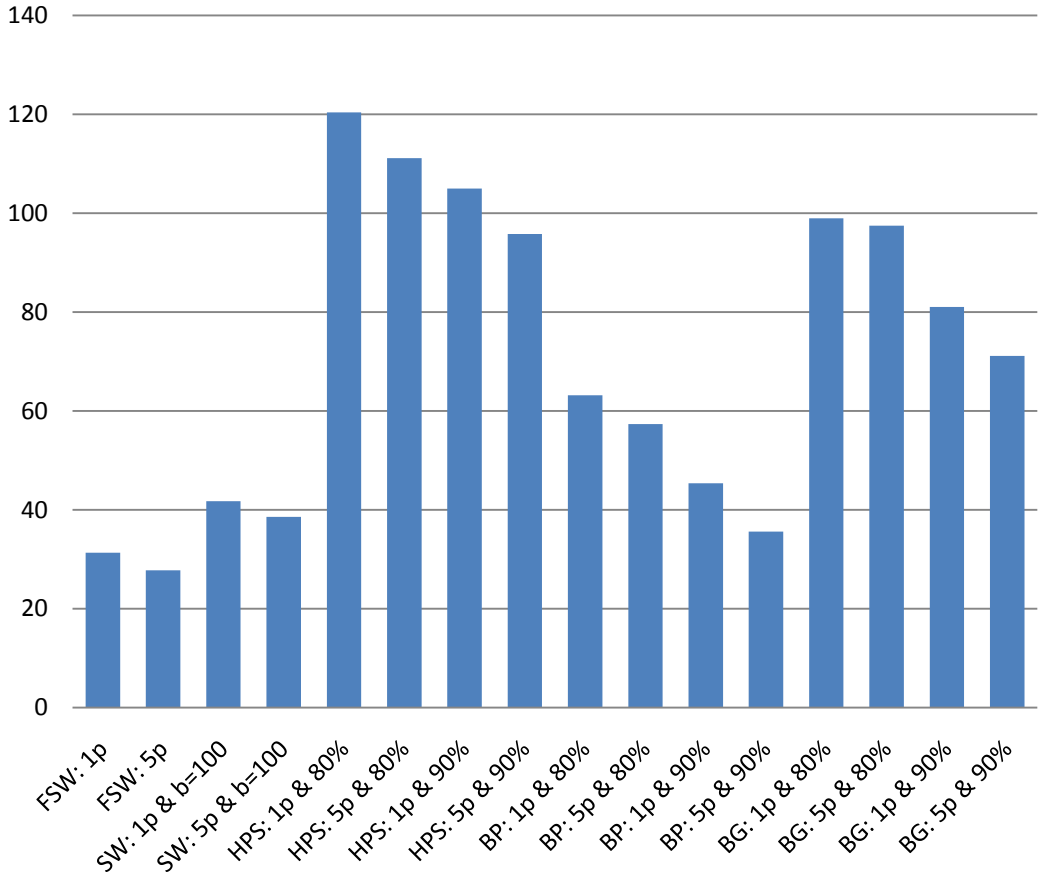
Experimental Results (1/8)

- Wait time for 2% window size (Player mode 1)



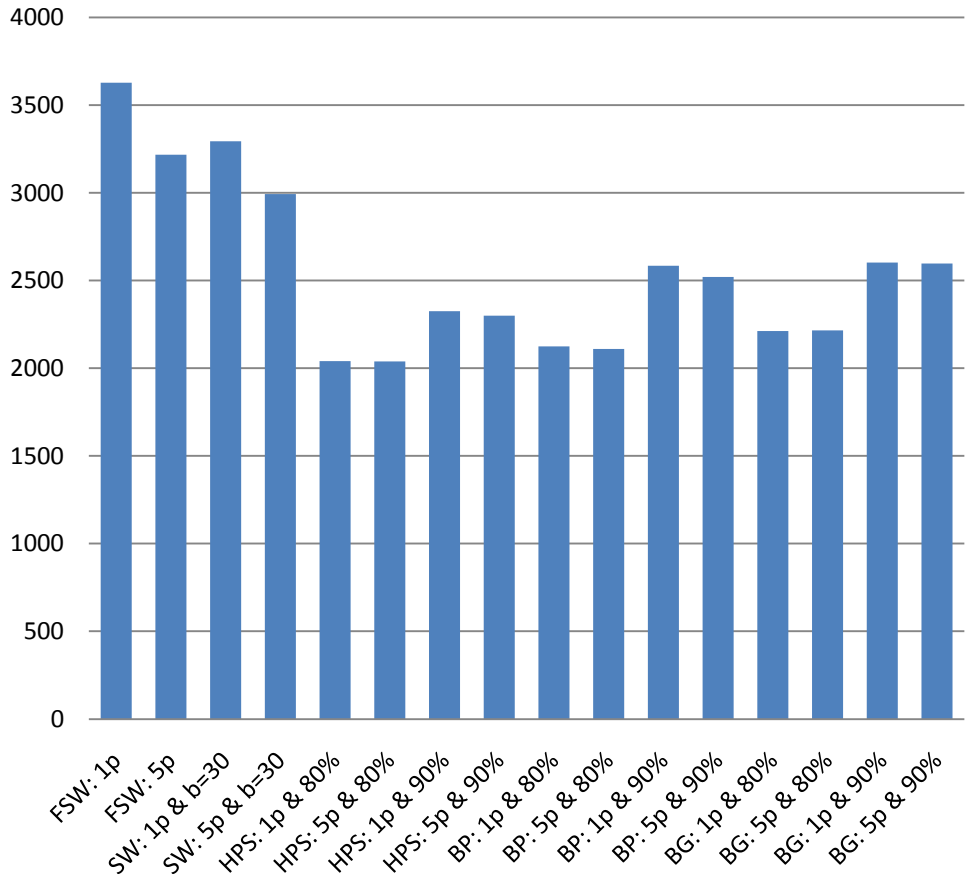
Experimental Results (2/8)

- Wait time for 8% window size (Player mode 1)



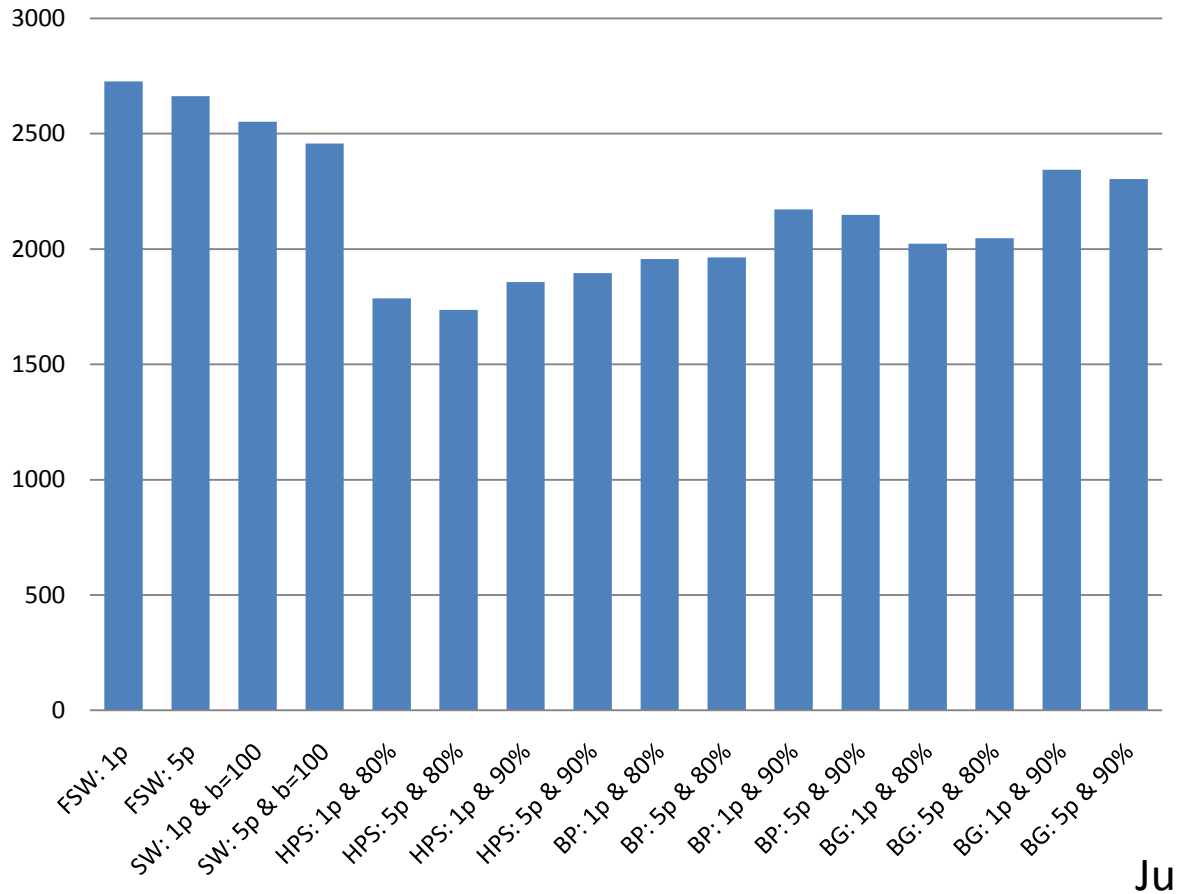
Experimental Results (3/8)

- Download time for 2% window size (Player mode 1)



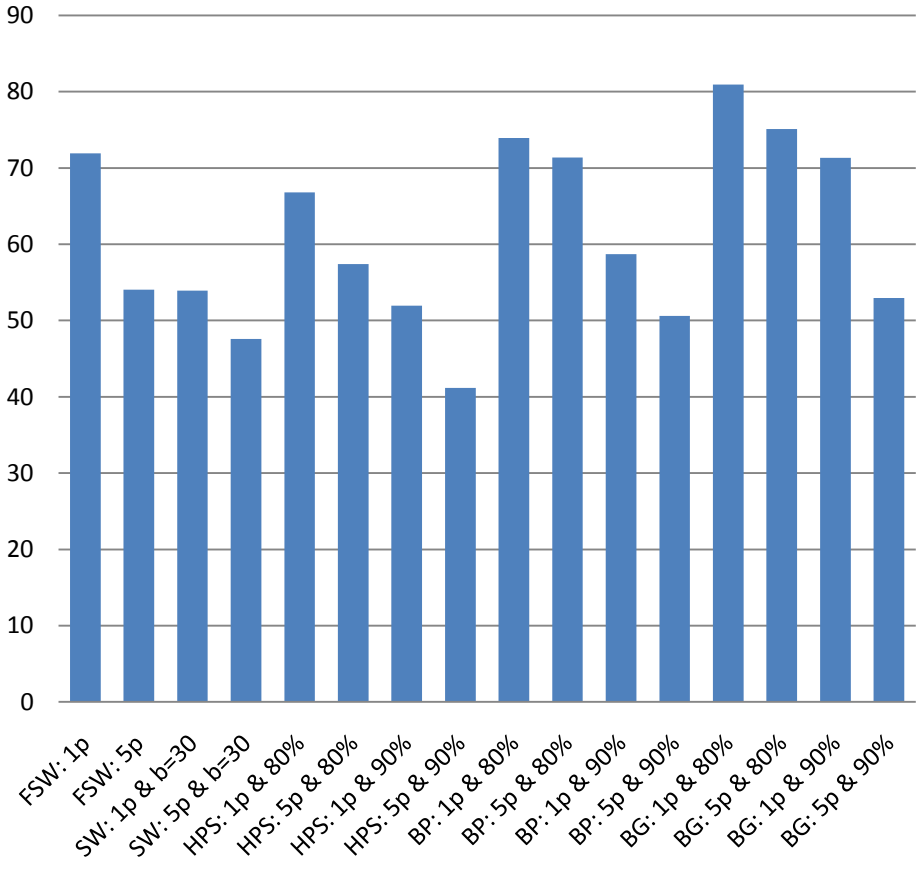
Experimental Results (4/8)

- Download time for 8% window size (Player mode 1)



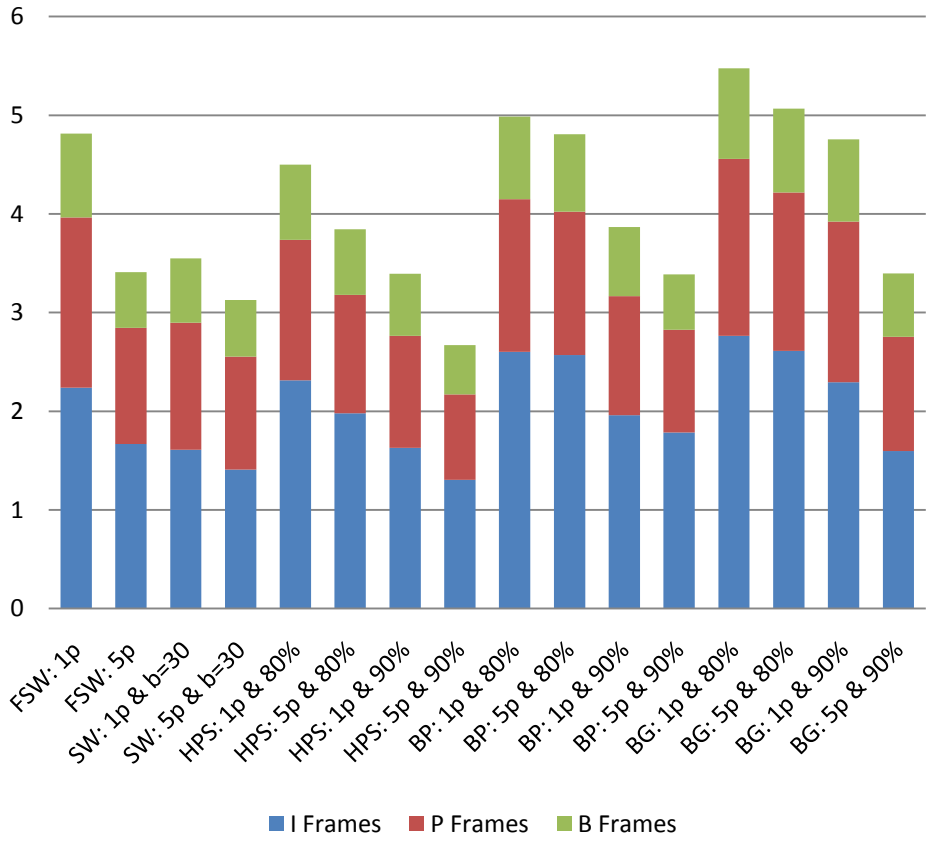
Experimental Results (5/8)

- Block loss for 2% window size (Player mode 2)



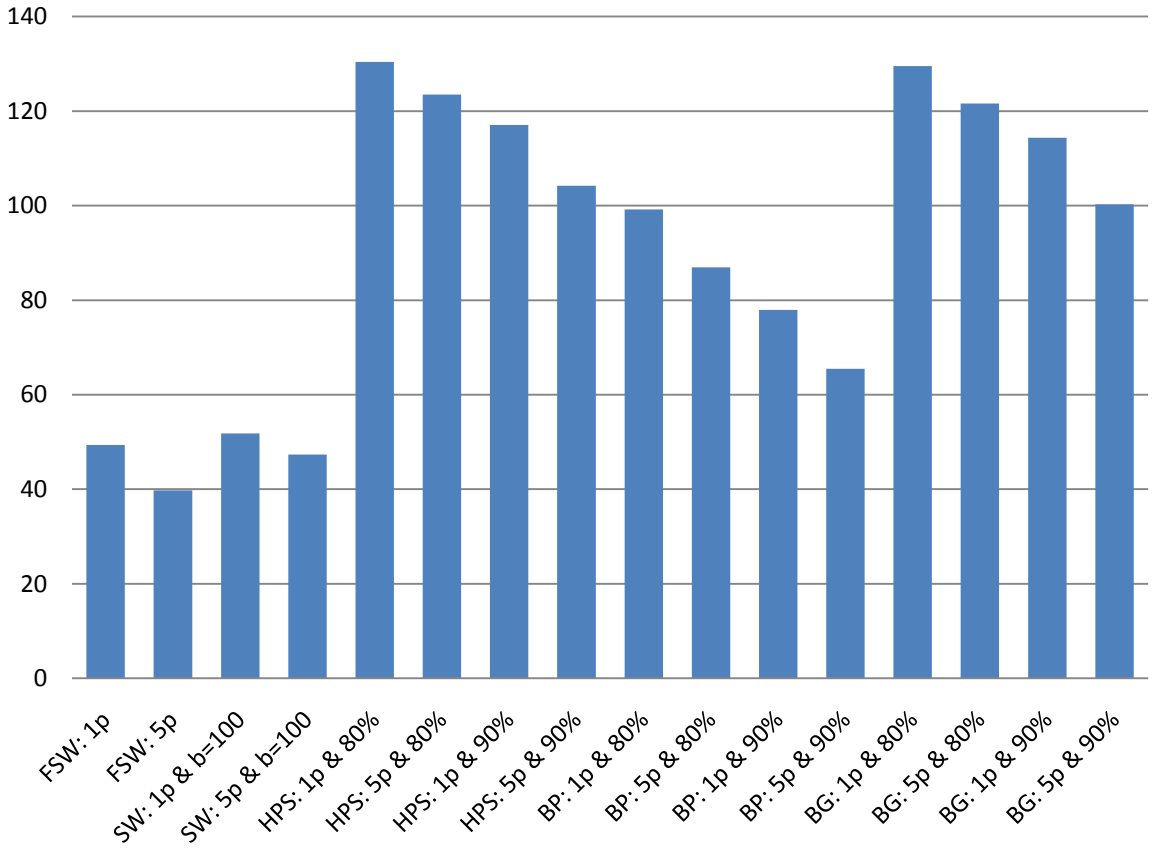
Experimental Results (6/8)

- Frame loss for 2% window size (Player mode 2)



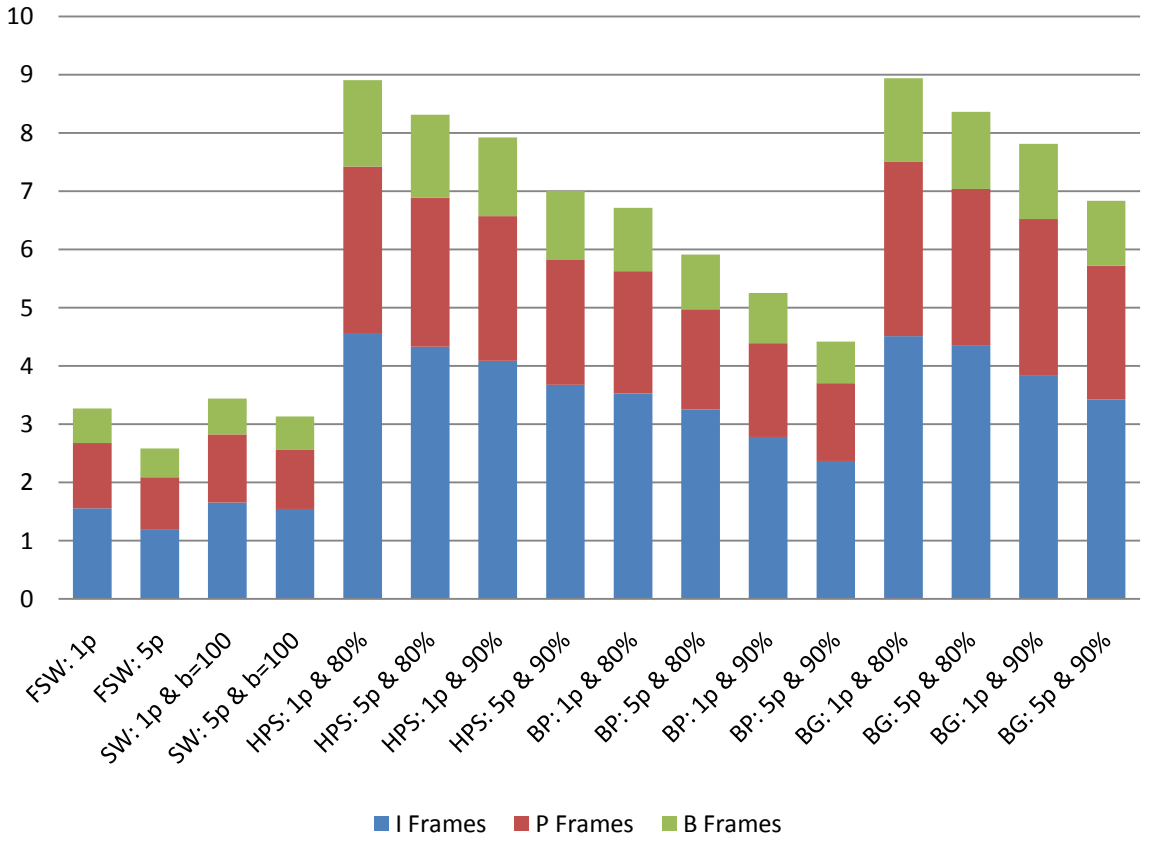
Experimental Results (7/8)

- Block loss for 8% window size (Player mode 2)



Experimental Results (8/8)

- Frame loss for 2% window size (Player mode 2)



Overall Evaluation

→ FSW exhibits lower wait time than SW, HPS and his modifications, both with a 2%, as well as an 8% window size

- FSW and SW perform nearly identically, therefore the extra complexity of SW does not seem to be worthwhile

→ FSW exhibits lower block loss rates than SW, HPS and his modifications for larger windows, but with smaller windows HPS works better

- Prefetching does lower the block loss rates and the wait time slightly, but at the cost of adding nearly 10 extra seconds of startup delay until the pieces are downloaded

→ Regarding download times, HPS is clearly the winner, but since all protocols complete the download well before the end of playback, this is not as important as a reduced loss rate or wait time.

Future Work

- Adaption of HPS probability p
  - Can be triggered by events like missing a deadline
    - Increase probability p in order to give high priority to pieces with shorter deadlines
    - Decrease p when missing many deadlines while not having any received pieces
      - Indicates that the peer is choked by most of its peers, because it doesn't have pieces to exchange
- In case of player mode two
    - try to download the missed blocks and seeding to other peers until the player move to the end of playback

References

- C.Stais, G.Xylomenos, A.Archodovassilis, “A comparison of streaming extensions to BitTorrent”, Proc. Of IEEE ISCC 2011, Corfu,Greece 2011 (to appear)
- K. Katsaros, V. P. Kemerlis, C. Stais and G. Xylomenos, "A BitTorrent Module for the OMNeT++ Simulator," Proc. 17th Annual Meeting of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), London, Great Britain, September 2009 [PDF]
- A. Vlavianos, M. Iliofotou, and M. Faloutsos, “BiToS: Enhancing BitTorrent for Supporting Streaming Applications”, in Global Internet Workshop in conjunction with IEEE INFOCOM 2006, April 2006 [PDF]
- P. Shah and J.-F. Pâris, “Peer-to-peer multimedia streaming using BitTorrent,” in *Proceedings of the 26th IEEE International Performance, Computing, and Communications Conference (IPCCC '07)*, pp. 340–347, New Orleans, La, USA, April 2007. [PDF]
- P. Savolainen, N. Raatikainen, and S. Tarkoma, “Windowing BitTorrent for video-on-demand: Not all is lost with tit-for-tat,” in *Proc. of IEEE GLOBECOM, 2008.* [PDF]
- C. Dana, D. Li, D. Harrison, and C. N. Chuah, "BASS: BitTorrent Assisted Streaming System for Video-on-Demand," in IEEE International Workshop on Multimedia Signal Processing (MMSP), October 2005. [PDF]
- R. LaFortune, C. D. Carothers, W. D. Simth, J. Czechowski and X. Wang, ”Simulating Large-Scale P2P Assisted Video Streaming”, In Proceedings of the Hawaii International Conference on System Sciences (HICSS-42), Waikoloa, Big Island, Hawaii, January 2009. [PDF]
- BitTorrent Development Community. BitTorrent Protocol Specification v1.0 http://wiki.theory.org/BitTorrentSpecification
- András Varga OMNeT++ network simulator homepage http://www.omnetpp.org

Thank you

Q & A