Athens University of Economics and Business
Master in Computer Science

Master Thesis

Implementation and Performance Evaluation of Video Streaming in the BitTorrent Protocol

Author: Aggelos Archodovassilis
Supervisor Prof.: George Xylomenos
Co-Supervisor Prof.: George Polyzos

June 2010

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, holding the lion's share among other P2P file sharing systems
  - BT is responsible for more than 45-78% of all P2P traffic. [src]

- BitTorrent has been estimated that it accounts for roughly 27-55% of all Internet traffic as of February 2009. [src]

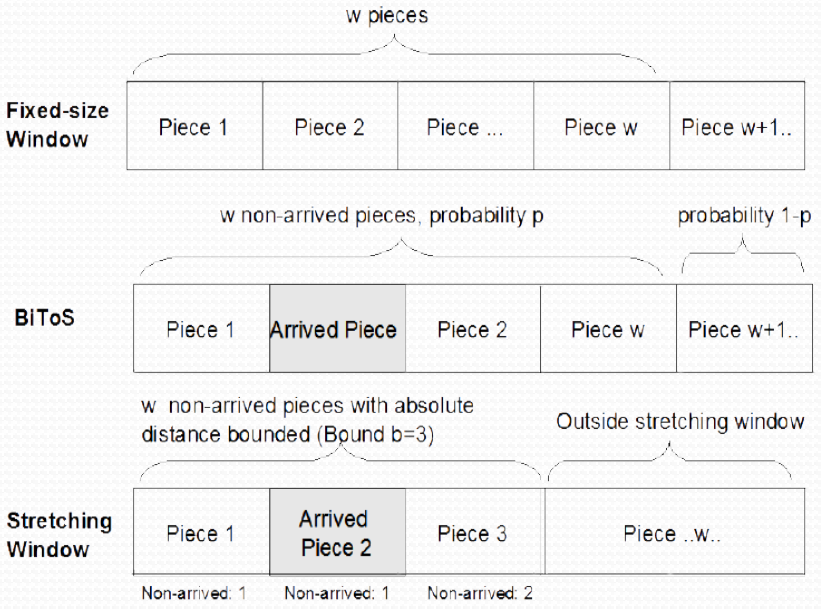
Video Streaming

- Video streaming has become increasingly popular in the last few years
  - media content proliferation in many different application areas
    - education, entertainment, medical treatment
- BitTorrent with minimal changes can support streaming.
- Importance of streaming capability
  - 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.
- Video Streaming implementation and simulation scenarios were made on a full featured and extensible implementation of BitTorrent for the OMNeT++ simulation environment.[PDF]

Related Work

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

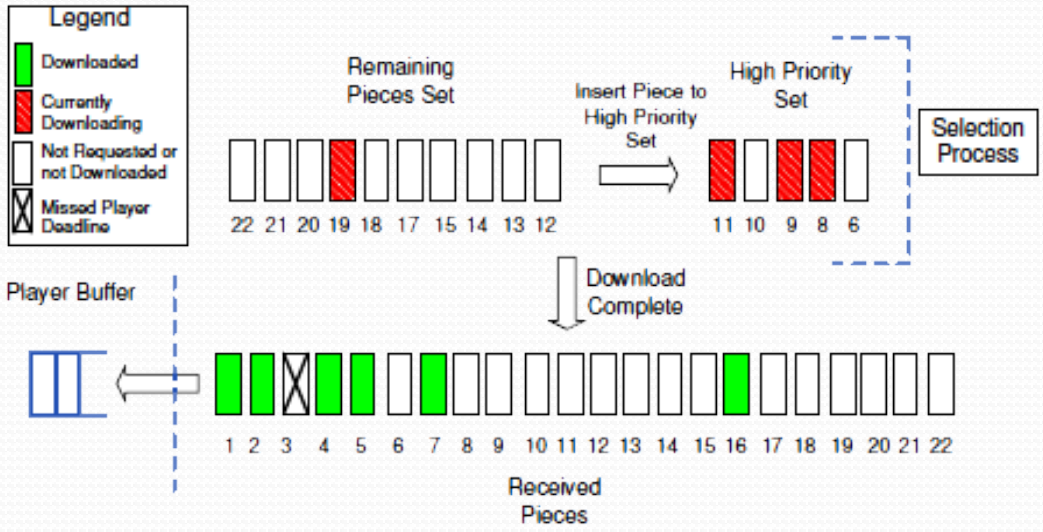
Algorithm	Maximum Window Size	Probability of Requesting from within the Window
Fixed-size Window	*w* pieces	1
BiToS	*w* non-arrived pieces	*p* (typically 0.8)
Stretching Window	*w* non-arrived pieces or *b* pieces	1



BitTorrent Module Modifications (1/2)

- Piece Selection Strategy
  - Rarest first – peer chooses the rarest among the *wanted* pieces
  - Random first – avoiding multiple peers converging on the same piece

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



BitTorrent Module Modifications (2/2)

- Buffering Time
  - Download the first five pieces of the file and then start playing the video
  - Buffering time usually between 7 to 25 seconds
  - Increasing the buffering time can dramatically decrease the missed pieces
- High Priority Set (HPS)
  - Fixed size depending on the file size
  - Coin throw to decide whether to choose a piece from the HPS or not
- Player
  - Plays the available pieces in sequential order
  - Checks if the next piece in order is available
    - If not, stops downloading it and places it in the *Missed Pieces Set*

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).
- Piece size
  - Smaller piece size
    - \+ Shorter initial buffering time
    - \+ Smaller (in KB) and more flexible window size
    - – Overhead increase
    - – Larger .torrent file
- HPS size
  - Large HPS size
    - Peers download pieces based on their rareness without considering their deadline
  - Small HPS size
    - Peers do not increase the diversity of the pieces
  - Tradeoff
    - HPS - file size dependent

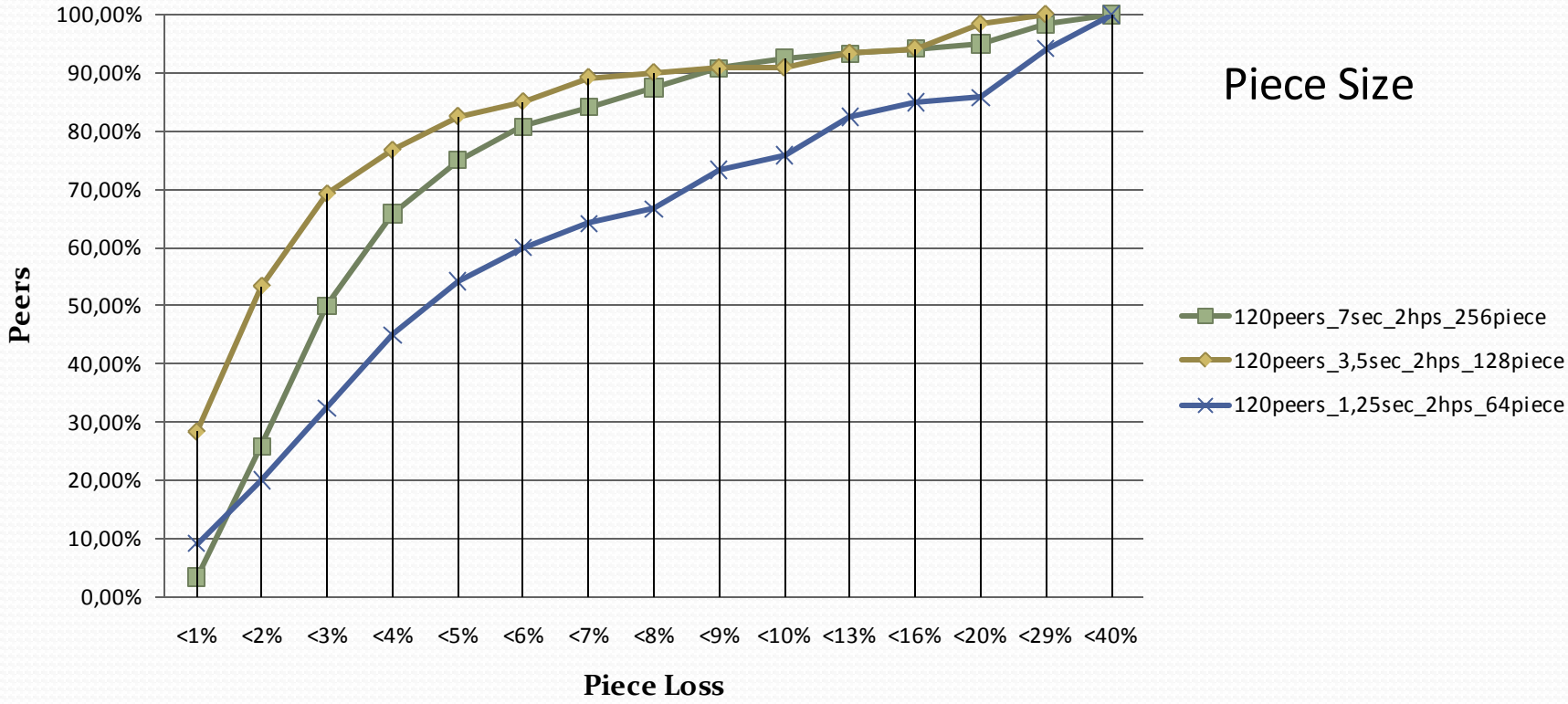
Simulations parameters (2/2)

- 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
    - 256KB piece played in 7sec (about 300Kbps)
    - 128KB piece played in 7sec (about 150Kbps)

Parameter	Value
file size (MB)	200
swarm size	60, 120, 180
piece size (KB)	64, 128, 256
block size (KB)	8, 16
HPS size (%)	2, 3, 8, 20
HPS probability	0.8, 0.9
video bit rate (kbps)	150, 300
timeToSeed (sec)	360

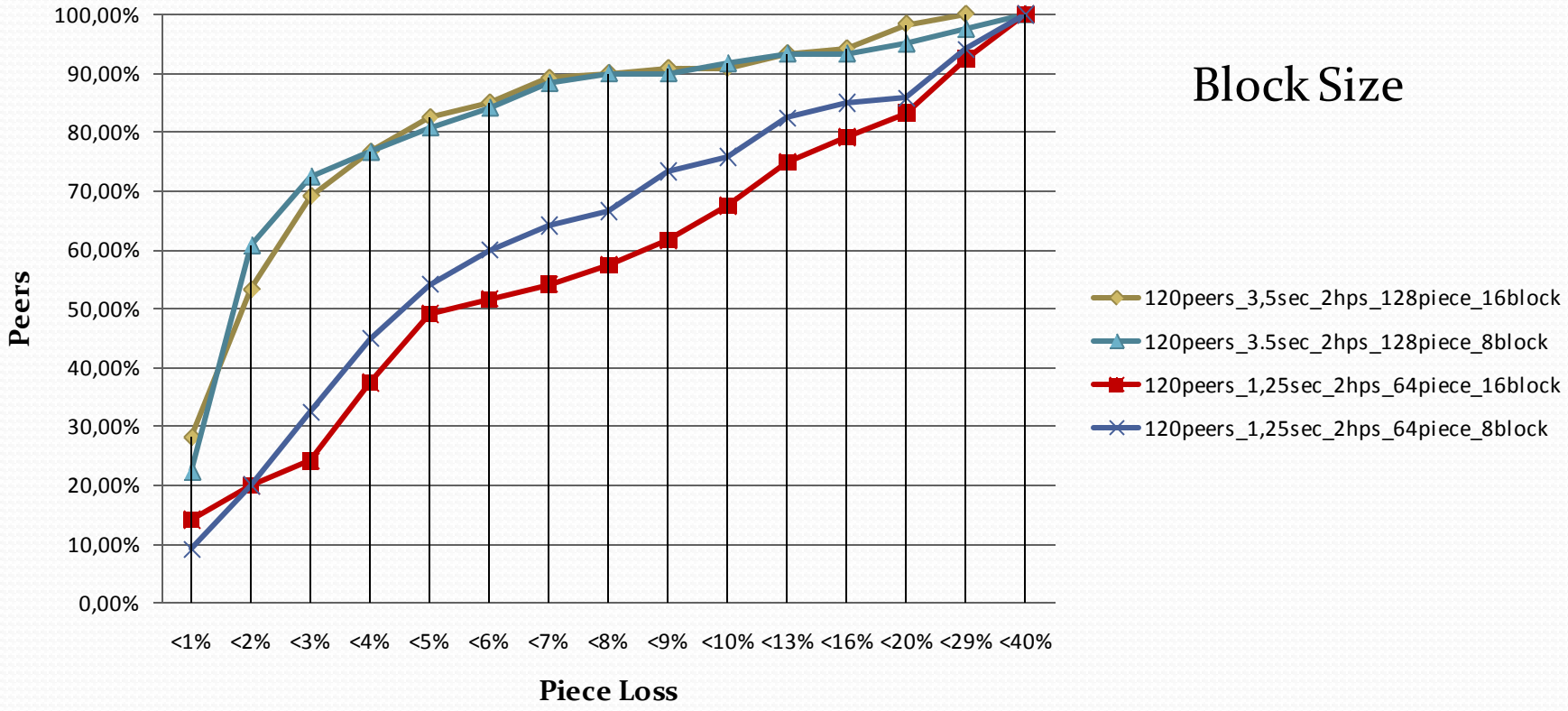
Observations (1/6)

- Piece Size
  - For large values (256KB) the window is not flexible enough, missed pieces occur more often
  - For very small values (64KB) the increased overhead reduces the number of *successful* pieces



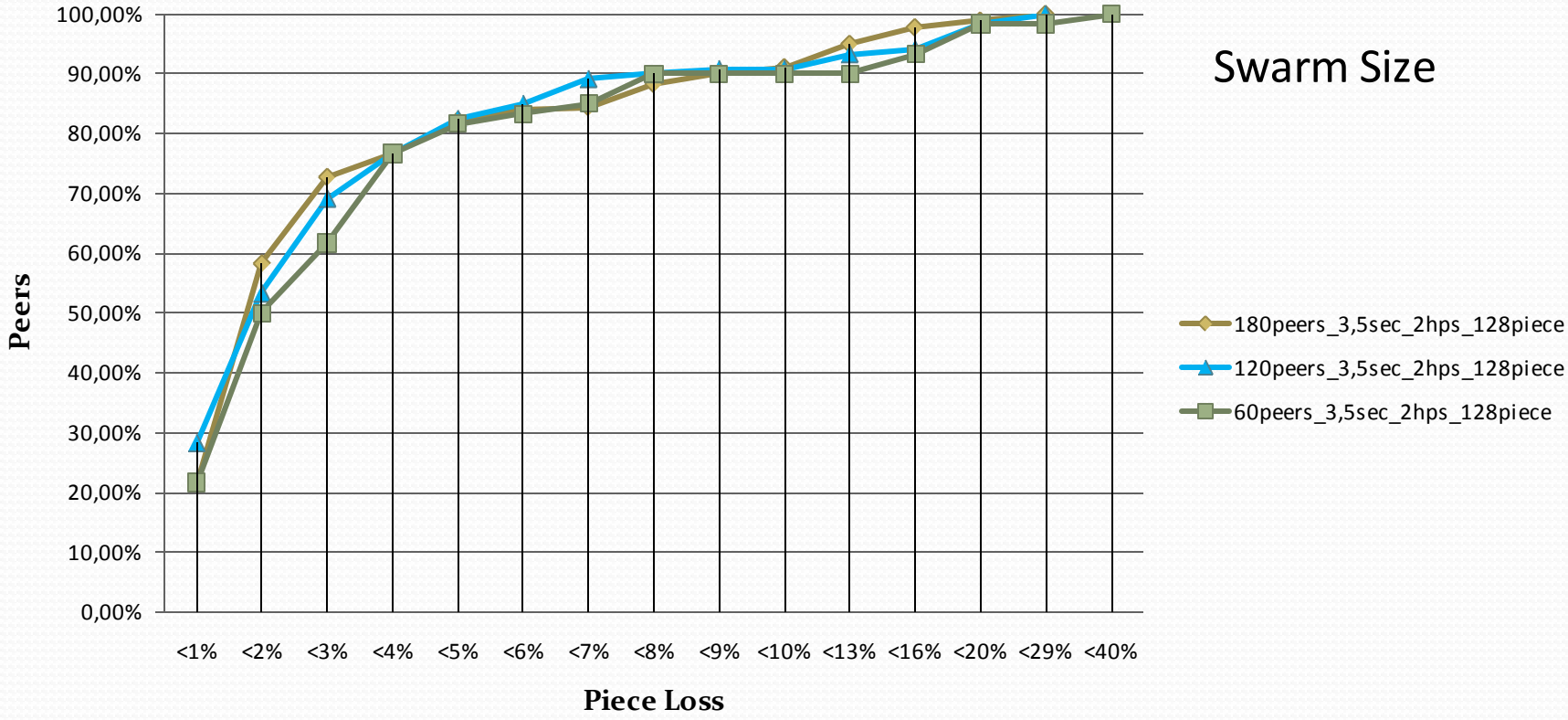
Observations (2/6)

- Block Size
  - Not much difference, when the block size is 8KB, the missed pieces are slightly decreased



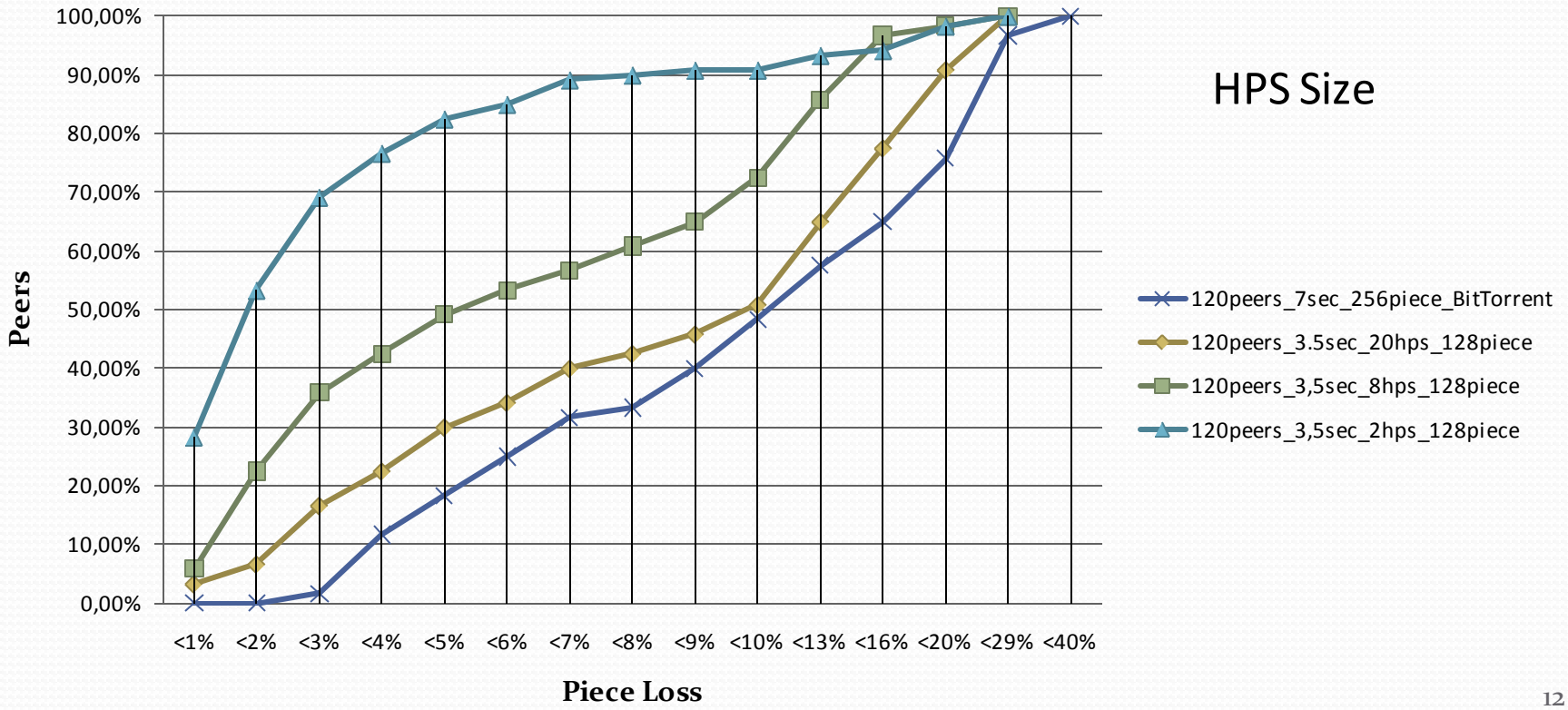
Observations (3/6)

- Swarm Size
  - Does not seem to affect much the piece loss rate
  - Larger swarm size, less missed pieces



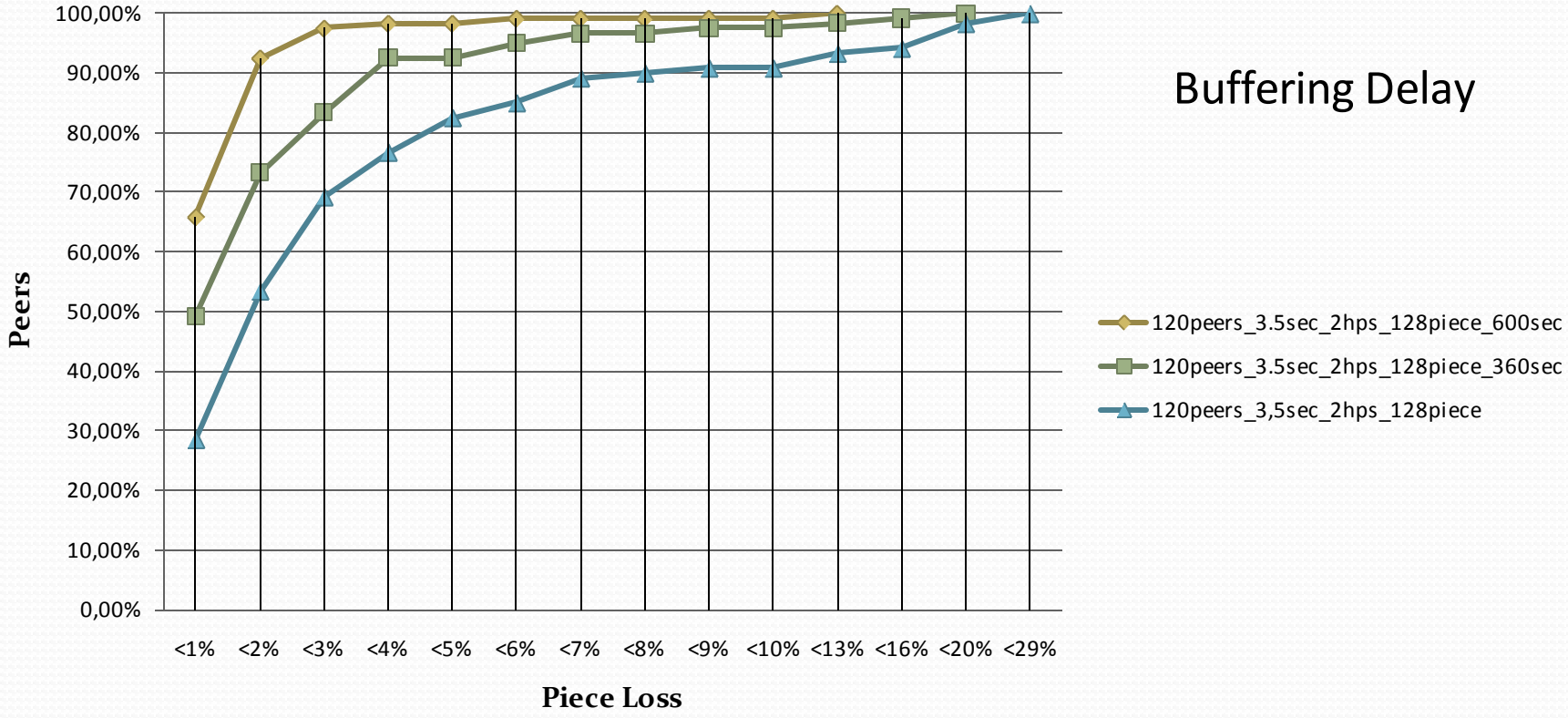
Observations (4/6)

- HPS Size
  - Original BitTorrent is unacceptable for video streaming
  - Large HPS size (20% of the file size) is the worst case scenario, close to original BitTorrent
  - BiToS recommended HPS size (8% of the file size) is far away from the optimal
  - Small HPS size (2%) is obviously the best choice while dealing with large files



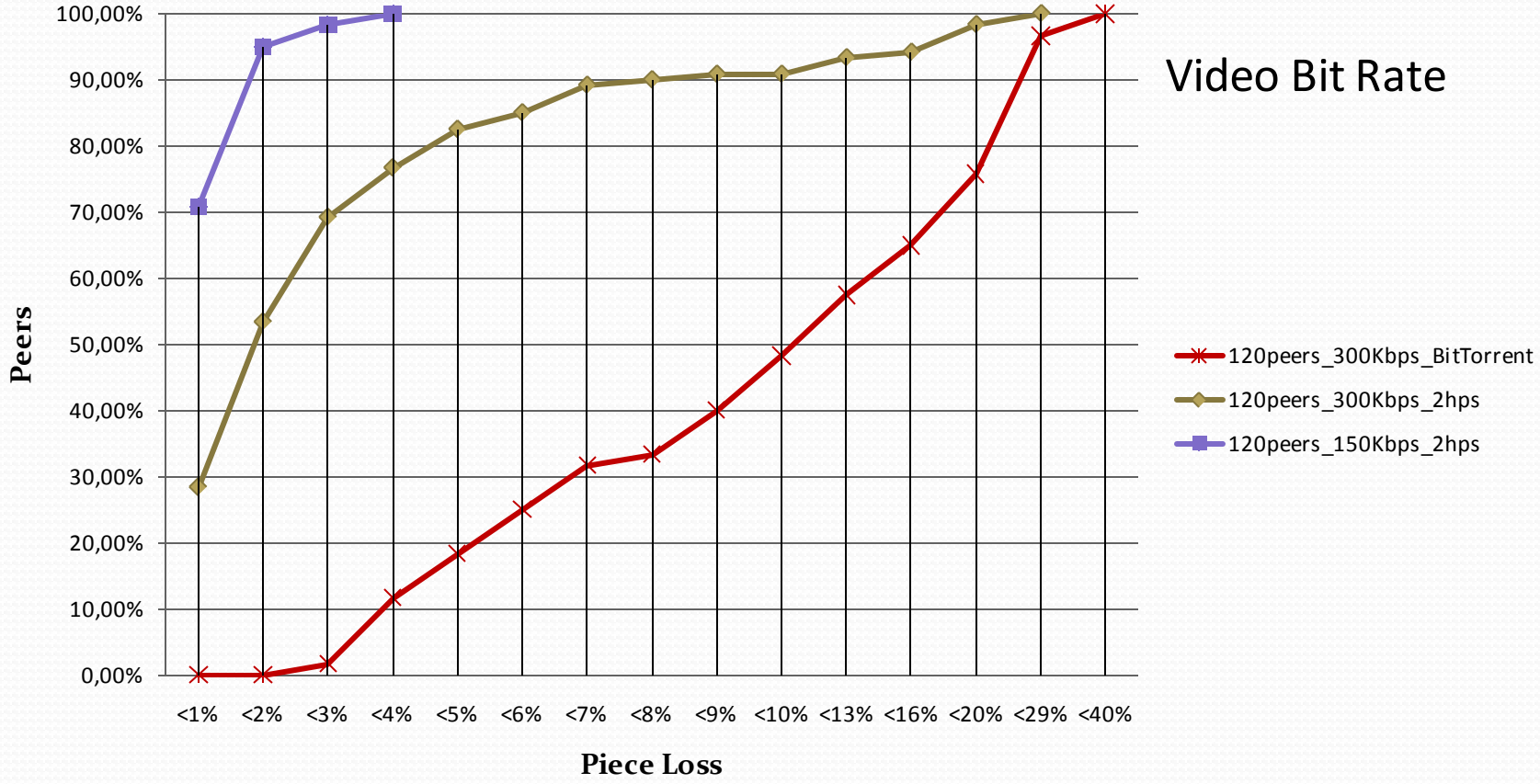
Observations (5/6)

- Buffering Delay
  - Adding an initiative buffering delay causes an enormous decrease on piece losses
  - The bigger the initiative buffering delay, the better for proper streaming



Observations (6/6)

- Video Bit Rate
  - Choosing lower streaming bit rate, the piece losses are minimal
  - The quality of the video though gets significantly low → limited applications

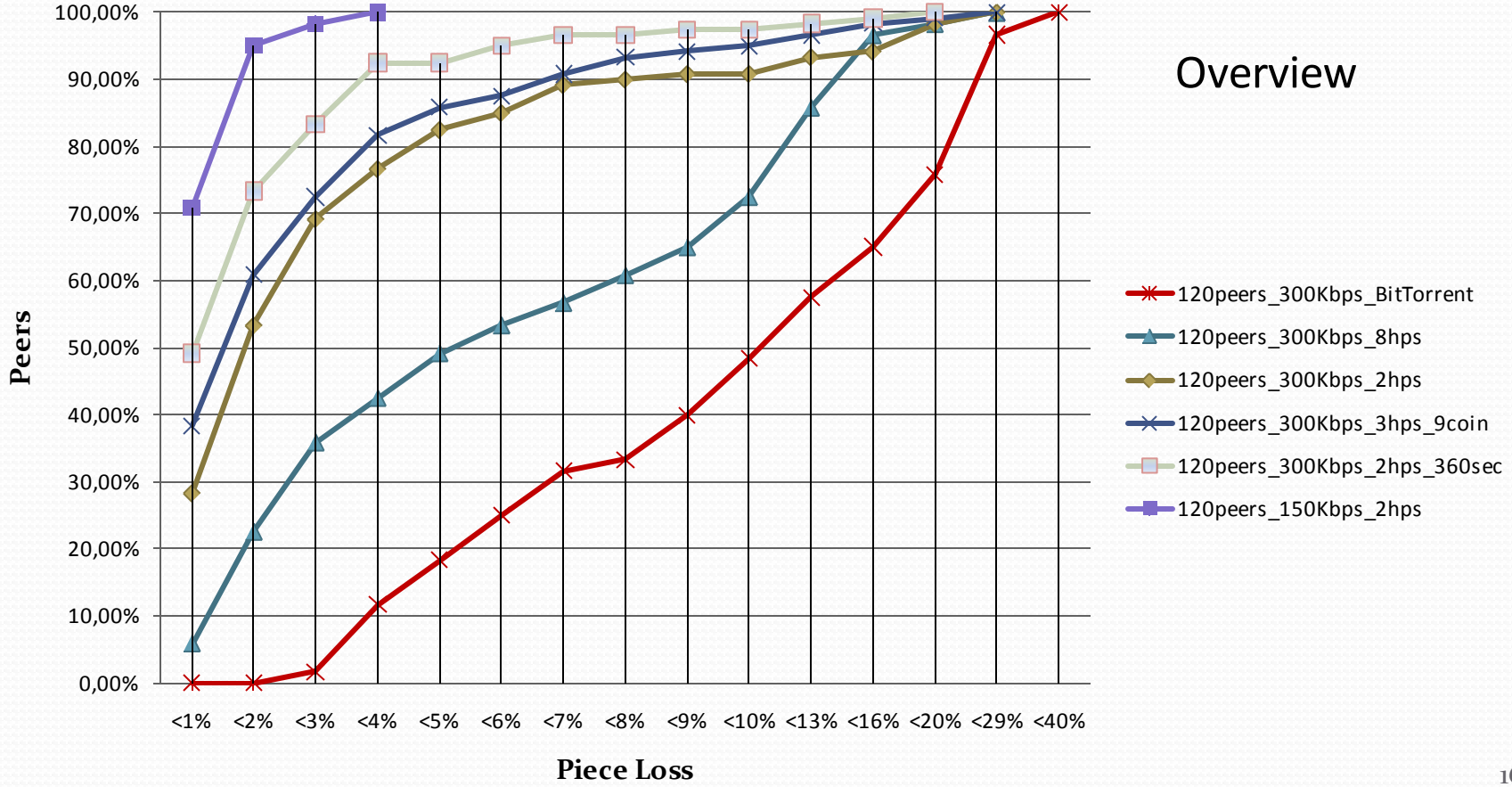


300

150

Overview

- Observations
  - The "probability p = 0.09 with 3% HPS size" scenario works quite well
  - Depending on the file/swarm size modifications can increase streaming quality



Comparison

- Peer-to-Peer Multimedia Streaming Using BitTorrent
  - Success Ratio: the number of pieces that arrive before a scheduled playback deadline over the total number of pieces in the video
  - Best case scenario success ratio (average): 83.3%
- BitTorrent Streaming Module for Omnet++
  - Best case scenario success ratio (average): 95.64%
  - Could reach 99% for 150Kbps streaming, or 97.84% by adding 360sec buffering delay

Peer-to-Peer Multimedia Streaming Using BitTorrent	
150MB file size, 256KB piece size, 100 swarm size, 60sec delay	
Piece Selection Policy	**Success Ratio**
Rarest-first policy (original BitTorrent)	23.6%
Sequential policy	8.1%
Sliding window and rarest-first policy	83.3%

BitTorrent Multimedia Streaming Module for Omnet++	
200MB file size, 120 swarm size, 7-27sec delay	
Piece Selection Policy	**Success Ratio**
HPS (8% of the file size) – 256KB piece size	91.59%
HPS (2% of the file size) – 256KB piece size	93.88%
HPS (2% of the file size) – 128KB piece size	95.64%

Final Results

- Video Streaming operates more efficiently with **128KB** piece size and **8KB** block size
- For streaming large files, above 100MB, the recommended HPS size is **2%** of the file size
- The possibility p to choose a piece from the HPS, should be near **0.8**
- We could significantly increase streaming performance by adding an initiative **buffering delay** or by decreasing the **streaming bit rate**
- Larger **swarm size** increases streaming performance

Future Work

- Adaptation of HPS probability p
  - Can be triggered by events like missing a deadline
    - **Increase** probability p in order to give higher 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
- Not missing a piece when most of its blocks are downloaded
  - When the piece meets its deadline and its not fully downloaded we could check how many blocks we already have and decide if the piece can be played or not

Based on:

- 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]

- BitTorrent Development Community. BitTorrent Protocol Specification v1.0 http://wiki.theory.org/BitTorrentSpecification

- András Varga OMNeT++ network simulator homepage http://www.omnetpp.org

References

- 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