

By how far have recent churn-reduction techniques improved quality of experience in peer-to-peer video streaming environments?

Martin Symons

April 29, 2024

### **Abstract**

abstract !!!

## **1 Introduction**

Within the past ten years, video streaming traffic across the internet has exploded. Traditional client-server architectures places a large load on the small portion of nodes controlled by the streaming host, and lead to dramatic infrastructure costs. To combat this, peer-to-peer systems have been proposed that spread usage across the network's collective resources. Starting with Coolstreaming in 2004, these networks became defacto for IPTV streaming, especially pirated streams. With their utility proven, research accelerated, in pace and complexity. Industry, however, did not pick up. Since the shutdown of New Coolstreaming and PPLive in the late 2000's, these newer solutions have seen little real-world usage.

As such, the actual impact of current research on network performance is vague. Still, research since has suggested that churn has a substantial and exponential impact on client quality-of-service (QoS) beyond that which was considered in these original models. This paper investigates several marked improvements over the best-documented benchmark architecture, New Coolstreaming, with particular focus on methods to reduce churn throughout the network. wordsw words words

## 2 Literature Review

### 2.1 Peer-to-Peer Streaming Fundamentals

In the late 1990's the traditional client-server architecture began to crack, as computer capabilities began to outpace bandwidth growth across the internet. Small clusters of server nodes were expected to take the burden, whilst ample client resources waited idle. BitTorrent allows a swarm of peers to collaborate in file transmission, maximizing global utilization without overwhelming any single node. Peers instead connect to a tracker or query a global DHT, usually *Mainline DHT*, to gather a list of participatory nodes. Targetting the rarest blocks first, data chunks within the swarm are piped directly over UDP node-to-node. As more peers join the network, the overall bandwidth increases, with download speeds following suit.

This approach exploded onto the internet, claiming 35% of all upstream traffic (**archiveCacheLogicResearch**). It remained in the number one spot for total traffic until recently - finally dethroned in 2024 by video streaming.

In a file-share network QoS is taken mostly with upload speed, and by extension bandwidth utilization. Any intermediate action from loading a *.torrent* to completion is irrelevant. Video streaming, in contrast, introduces a large number of real-time requirements essential for a good user experience.

Peer-to-peer networks pipe data directly between nodes through a network, contrasted with the traditional client-server architecture seen most often today. Compared to file-sharing applications, peer-to-peer video networks face several new complexities. In the former, QoS is ruled almost entirely by bandwidth utilization. Users care only that a desired file eventually reaches them; the journey it takes to completion is unimportant. probably list some examples Video livestreaming, however, introduces new complexities. playoutdeadlines startupdelay end-to-enddelay one proposed solution was ip multicasting. it was . fucked. s owe don't use it and instead use the various better solutions

### 2.2 Performance Heuristics and Churn

explaining why we use/care about churn vs other characteristics

## 2.3 Overlay Structure

## 2.4 Peer Selection Strategies

## 2.5 Chunk Schedulers

## 2.6 Novelties

## 2.7 Historic Implementations

We now consider historic implementations of peer-to-peer streaming systems in relation to the above characteristics. Of most relevance to our paper is DONet, commercially branded Coolstreaming (**1498486**). Considered the first of its breed, Coolstreaming Xie et al. 2007 **asdasd 1498486**

# 3 Another Look at Coolstreaming

## 4 Introducing *Coolstreaming-Spiked*

## 5 Improvements Chosen

## 6 Methodology

## 7 Analysis

## 8 Discussion

## 9 Conclusion

## References

Xie, Susu, Bo Li, Gabriel Y. Keung, and Xinyan Zhang. 2007. Coolstreaming: design, theory, and practice. *IEEE Transactions on Multimedia* 9 (8): 1661–1671. <https://doi.org/10.1109/TMM.2007.907469>.