

 Open access • Proceedings Article • DOI:10.1109/P2P.2012.6335814

Peer-to-peer live streaming for Massively Multiplayer Online Games — [Source link](#)

Shakeel Ahmad, Christos Bouras, Eliya Buyukkaya, Raouf Hamzaoui ...+4 more authors

Institutions: De Montfort University, Research Academic Computer Technology Institute, École nationale supérieure des télécommunications de Bretagne

Published on: 22 Oct 2012 - International Conference on Peer-to-Peer Computing

Topics: Game mechanics, Video game development, Turns, rounds and time-keeping systems in games and Game theory

Related papers:

- [Peer-to-peer infrastructures for games](#)
- [A Peer to Peer Architecture Applied to Multiplayer Games](#)
- [The Community Network Game project: Enhancing collaborative activities in online games](#)
- [PartyPeer: a P2P massively multiplayer online game](#)
- [Future collaborative systems between peer-to-peer and massive multiplayer online games](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/peer-to-peer-live-streaming-for-massively-multiplayer-online-54im4125zt>

Peer-to-peer live streaming for massively multiplayer online games

Shakeel Ahmad* Christos Bouras† Eliya Buyukkaya‡ Raouf Hamzaoui* Andreas Papazois†
Alex Shani§ Gwendal Simon‡ Fen Zhou‡

*De Montfort University, United Kingdom, {sahmad, rhamzaoui}@dmu.ac.uk

†Computer Technology Institute & Press “Diophantus”, Greece, bouras@cti.gr, papazois@ceid.upatras.gr

‡Telecom Bretagne, France, {firstname.lastname}@telecom-bretagne.eu

§Exent Technologies, Israel, ashani@exent.com

Abstract—One of the most attractive features of Massively Multiplayer Online Games (MMOGs) is the possibility for users to interact with a large number of other users in a variety of collaborative and competitive situations. Gamers within an MMOG typically become members of active communities with mutual interests, shared adventures, and common objectives. This demonstration presents a peer-to-peer live video system that enables MMOG players to stream screen-captured video of their game. Players can use the system to show their skills, share experience with friends, or coordinate missions in strategy games.

I. INTRODUCTION

Massively Multiplayer Online Games (MMOGs) allow a large number of online users to inhabit the same virtual world and interact with each other in a variety of collaborative and competing scenarios. Players can play against other players or build groups to compete against other groups or against computer-controlled enemies. Within a group, synchronisation among teammates requires communication tools. Unfortunately, existing games offer only a few basic communications tools (if any). The survey we conducted in the context of the FP7-funded CNG project [1] has highlighted that streaming live screen-captured video of the game is one of the most desirable communication tools. Players can use it to show their skills, share experience with friends, or coordinate missions in strategy games.

Existing online video platforms such as Xfire [2] and TwitchTV [3] rely on a centralized architecture. Even when the system is coupled with a Content Delivery Network (CDN), this solution is not appropriate. Indeed, the popularity distribution of our service poses a major challenge to current large-scale delivery systems. On one hand, a large proportion of players are likely to act as sources, so there are many live streams to deal with. On the other hand, each stream is typically watched by a small population consisting of a few friends and teammates.

Within the CNG project, we have developed a P2P system for sharing live screen-captured video of MMOGs. While many P2P live video systems are available, none of them was specifically designed for MMOGs. In particular, no existing P2P live video system fulfills all the following requirements:

- The video source can be a casual gamer with limited

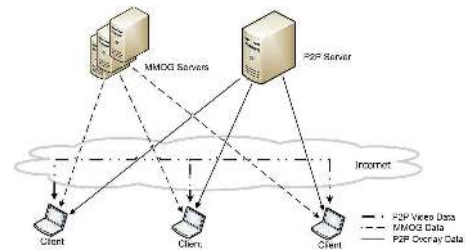


Fig. 1: System architecture.

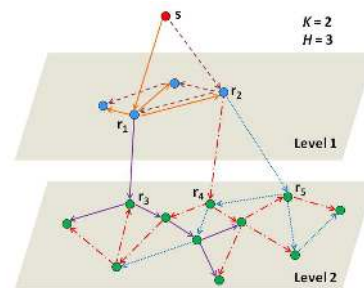


Fig. 2: Overlay example when $K = 2$ encoded packets are needed to successfully decode the video and the maximum height of any tree is constrained to be $H = 3$. There are two trees in Level 1 (higher level) rooted at r_1 and r_2 and three trees in Level 2 (lower level) rooted at r_3 , r_4 , and r_5 . Every peer is contained in two trees.

upload bandwidth. It is therefore critical to minimize the transmission rate at the source.

- Live video streaming should not consume the upload and download bandwidth that is necessary for the smooth operation of the MMOG (MMOG client-server traffic).
- Users should be able to watch several videos simultaneously for, for example, intra-group coordination.

We designed a multi-level multi-overlay P2P system consisting of peers and a P2P server. The P2P server has persistent communication with the peers and is responsible for building the P2P overlays (Fig. 1). *Multicast trees* are used to diffuse video data within and across levels (Fig. 2). We use rateless codes [4] to provide resilience against packet

loss. Rateless codes are ideally suited for our application as they (1) have very low computational cost, (2) minimize delivery redundancy when a peer receives data concurrently from multiple peers, (3) make the system adaptive to varying channel conditions since the encoder can generate on the fly as many encoded symbols as needed.

We propose a novel algorithm to efficiently schedule the rateless encoded packets within and across levels. Our algorithm tries to build multicast trees that minimize the transmission rate at the source while guaranteeing on-time delivery and reliability for the peers. We formulate this objective as a height bounded spanning forest problem with nodal capacity constraint and compute a solution in polynomial-time.

II. CNG P2P SYSTEM

A peer wishing to broadcast its video sends a request to the P2P Server. If the request is accepted, an overlay is created for this video session and the video is advertised. If another peer is interested in the advertised video, it sends a request to the P2P Server. If the request is accepted, the P2P Server updates the overlay information and informs participating peers. Social relationships between peers are exploited in the management of the available live videos. For example, peers can easily find all videos advertised by their friends. The overlay information consists of peer assignments to levels and sets of multicast trees for each level. When constructing an overlay, peers' upload capacity constraint is respected, that is, for each peer, the capacity needed for forwarding received packets to other peers does not exceed the peer's upload capacity. When a peer participates in several P2P overlays, its resource is allocated among the overlays according to a bandwidth allocation strategy. Keep-alive messages are periodically sent from the peers to the P2P server. If a peer stops sending keep-alive messages, the server removes it from the overlay and updates the multicast trees. Because of the lack of space, details cannot be included here and will be published elsewhere (check [1]).

As soon as the source peer receives the overlay information from the P2P Server, it captures the video and compresses it with the H.264 video coder. The resulting bitstream is partitioned into source blocks, where each source block corresponds to one GOP (Group of Pictures) and is an independent unit of fixed playback duration Δ . Then the source peer applies rateless coding on each source block and sends the resulting encoded symbols in successive UDP packets. Packets are transmitted in an interval of duration Δ , and one packet is sent on each multicast tree.

A root of a Level 1 multicast tree (r_1 and r_2 in Fig. 2) immediately forwards packets directly received from the source to the Level 2 multicast trees associated to it (one tree for r_1 and two trees for r_2 in Fig. 2). Moreover, as soon as it successfully decodes a source block, it sends an acknowledgment to the source, so that the source stops sending it packets, and it creates new encoded packets by applying rateless coding on the decoded source block. Then it sends these new encoded packets to Level 2 peers over the multicast trees associated to it (ignoring those already used). The number of packets sent



Fig. 3: Snapshot of the CNG P2P system.

by a Level 1 root peer to Level 2 is set not to exceed the number of Level 2 multicast trees associated to this root peer. The same procedure is repeated for the next levels.

For Network Address Translator (NAT) traversal, the Interactive Connectivity Establishment (ICE) protocol is used. For efficient frame capture, the GPU is used. GStreamer is used to encode the video and send it to the P2P application over a local port. Video playback uses the VLC ActiveX plug-in to play a video stream from a local port.

III. DEMONSTRATION

The scalability of the CNG P2P system and its robustness to peer churn were successfully tested using the ns-2 simulator with up to 1500 peers. An online version of the system was integrated in The Missing Ink [5] MMOG and is currently in the Alpha testing phase. A large scale Beta testing will be conducted with The Missing Ink gamers in the summer of 2012. Figure 3 shows a snapshot of the screen of an Alpha tester during an online live streaming session. The tester is using the P2P system to watch three live streams while playing the game. The demonstration at the P2P'12 conference will use four computers and the available Internet connection to illustrate how a user can advertise a live stream, capture and stream a video of the game, and watch up to three simultaneous live streams while playing The Missing Ink game. Ten live videos will also be streamed from CNG consortium members located in Leicester, London, Patras, Tel Aviv, and Brest. In addition, a number of live videos from regular The Missing Ink gamers will be available.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Commission's Seventh Framework Programme (FP7, 2007-2013) under the grant agreement no. ICT-248175 (CNG project).

REFERENCES

- [1] Online. Available: <http://www.cng-project.eu/>.
- [2] Online. Available: <http://www.xfire.com/>.
- [3] Online. Available: <http://www.twitch.tv/>.
- [4] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, pp. 2551–2567, June 2006.
- [5] Online. Available: <http://www.missing-ink.com/>.