CNG

ICT-248175

Deliverable D7.2

Report on Project Scientific Dissemination

Authors:	Raouf Hamzaoui (DMU)
Version:	V1.0
Date:	12/11/12
Classification:	Public
Contract Start Date:	01/02/2010
Project Co-ordinator:	Exent
File Name:	CNG D7.2 Report on Project Scientific Dissemination

SEVENTH FRAMEWORK PROGRAMME

Project funded by the European Commission under the

"Information and Communication Technologies" Programme

The CNG Consortium consists of

Participant name	S.N	Country
Exent Technologies Ltd.	EXENT	IL
Research Academic Computer Technology Institute	CTI	GR
Institut Telecom	ITE	FR
De Montfort University	DMU	UK
Kaltura Ltd.	KAL	IL
RedBedlam LTD	RB	UK
European Games Developer Federation	EGDF	SE
i2 media research limited	i2MEDIA	UK

Executive Summary

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 common interests, shared adventures, and common objectives.

The Community Network Game (CNG) project is a small or medium-scale focused research project (STREP) that is focused on applying new network technologies to support community activities over highly interactive centrally managed MMOGs.

The CNG project provides tools to enhance collaborative activities between online gamers and develops new tools for the generation, distribution and insertion of user-generated content (UGC) into existing MMOGs. CNG allows the addition of new engaging community services without changing the game code and without adding new processing or network loads to the MMOG central servers.

CNG provides in-game community activities using an in-game graphical insertion technology (IGIT) and an architecture that combines the client-server infrastructure for the MMOG with a peer-to-peer (P2P) system for the delivery of User Generated Content (UGC). IGIT is an innovative technology of replacing or inserting content into the game in real time without the need to change the game's code in the client or server. The UGC considered by the CNG project focuses on live video to be shared using P2P technology. The video traffic represents a real challenge to the network already occupied by the MMOG client-server data. The project proposes new techniques for P2P video streaming that are “friendly” to the MMOG client-server traffic.

This report gives an overview of the scientific dissemination activities of the project. It presents all scientific publications in journals and conferences as well as demos and workshops.

Table of Contents

1 Introduction

Massively Multiplayer Online Games (MMOGs) allow a large number of online users (in some cases millions) to inhabit the same virtual world and interact with each other in a variety of collaborative and competing scenarios. MMOG gamers can build and become members of active communities with mutual interests, shared adventures, and common objectives. Players can play against other players (player versus player) or build groups (guilds) to compete against other groups (realm versus realm) or against computer-controlled enemies.

This Community Network Game (CNG) project aims at enhancing community activities for MMOG gamers. This is achieved by providing Web collaboration tools and developing new tools for the generation, distribution and insertion of User-Generated Content (UGC) into existing MMOGs. This UGC may include textures and 3D objects to be added to the game, live video captured from the game screen and streamed to other players, as well as videos showing walk-throughs, game tutorials, or changes in the virtual world to be watched on demand.

The main technologies proposed by the CNG project are the In-game Graphical Insertion Technology (IGIT) and a Peer-to-Peer (P2P) system for the distribution of live video. IGIT is an innovative technology of replacing or inserting content into a game in real time without the need to change the game code in the client or server. For example, billboards can be inserted, tattoos can be added to in-game characters, an area on the screen can be assigned to display user information, and any type of window (browser, chat, etc.) can be inserted floating on or outside the game area. The technology can be implemented on multiple games, making it possible to create a community that is not limited to a specific game or publisher.

Enabling thousands of users to communicate UGC represents a significant challenge to networks already dealing with the MMOG client-server data. The CNG project develops new techniques for UGC distribution that are ``friendly" (supportive and not disruptive) to the MMOG client-server traffic. The key innovation is a P2P system that allows MMOG gamers to stream live video of the game without interrupting the MMOG data flow and without the need to upload the video data to a central server.

This report describes the scientific dissemination activities related to the CNG project. Chapter 2 presents scientific papers published in conferences and journals. Chapter 3 is dedicated to demos and workshops. Chapter 4 concludes the report. The publications are appended to the report.

2 Scientific publications

The CNG project has resulted in eight conference papers and two journal papers. In the following, we list each publication, describe its target audience and provide a summary of its contributions.

- [1] S. Ahmad, C. Bouras, R. Hamzaoui, A. Papazois, E. Perelman, A. Shani, G. Simon, and G. Tsichritzis, The Community Network Game project: Enriching online gamers experience with user generated content, in Proc. 2nd Int. Conf. Creative Content Technologies (CONTENT 2010), Lisbon, Nov. 2010.

- [2] S. Ahmad, C. Bouras, R. Hamzaoui, A. Papazois, E. Perelman, A. Shani, and G. Simon, The Community Network Game project: Enhancing collaborative activities in online games, in Proc. Networked & Electronic Media Summit, Barcelona, Oct. 2010.

- [3] S. Ahmad, C. Bouras, R. Hamzaoui, J. Liu, A. Papazois, E. Perelman, A. Shani, G. Simon, and G. Tsichritzis, Community Tools for Massively Multiplayer Online Games, International Journal on Advances in Networks and Services, vol. 4, no 3&4, 2011.

Audience: The Second International Conference on Creative Content Technologies (CONTENT) (http://www.iaria.org/conferences2010/CfPCONTENT10.html) targets advanced concepts, solutions and applications in producing, transmitting and managing various forms of content and their combination. Multi-cast and unicast content distribution, content localization, on-demand or following customer profiles are common challenges for content producers and distributors. Special processing challenges occur when dealing with social, graphic content, animation, speech, voice, image, audio, data, or image contents. Advanced producing and managing mechanisms and methodologies are now embedded in current and soon-to-be solutions. CONTENT 2010 is aimed at technical papers presenting research and practical results, industrial small- and large-scale systems, challenging applications, position papers addressing the pros and cons of specific topics, such as those being discussed in the standard fora or in industry consortia, survey papers addressing the key problems and solutions on any of the topics, short papers on work in progress, and panel proposals.

The Networked and Electronic Media (NEM) (http://nem-summit.eu/past-events/2010-nem-summit/) Initiative is a European Technology Platform under the Seventh Framework Programme. As an industry-led initiative, NEM aims at fostering the convergence between consumer electronics, broadcasting and telecoms in order to develop the emerging business sector of networked and electronic media. The NEM Summit, organised every year since 2008 by the NEM Initiative in close cooperation with leading industrial and academic players worldwide, is an annual event for all those interested in Future Internet developments and in the fast paced evolution of the networked and electronic media industry. Over the years, the NEM Summit has grown to become the key conference and exhibition to meet and network with the most active stakeholders, access up-to-date information, discover the latest technology and market trends, identify research and business opportunities, and find partners for upcoming EU-funded calls. The event traditionally involves over 400 attendees, 20 sponsors, 50 exhibitors and 40 speakers and authors selected following a competitive call for papers.

The International Journal on Advances in Networks and Services journal (http://www.iariajournals.org/networks_and_services/index.html) covers a large spectrum of networking and services, web-based applications, wireless and wired networks, optical networks, high speed networks, sensor networks, mesh and other access networks, and personalized networks, services and applications.

Contribution: These papers give an overview of the state-of-the art in the areas of UGC, Web collaboration tools, P2P live video streaming, and game adaptation technologies. They present the project's concept, objectives and technologies. They also discuss the project's benefits and expected impact. Paper [1] received a best paper award at the CONTENT 2010 conference. Paper [3] is an extended version of papers [1] and [2].

- [4] E. Ferrari, J. Lessiter, and J. Freeman, Users and uses of multiplayer games and community activities, in Proc. Networked & Electronic Media Summit, Turin, Sept. 2011.

Contribution: This paper outlines user needs and initial stakeholder research conducted as part of the CNG project. Seventeen interviews were conducted with multiplayer gamers to identify why players communicate with others, what they share and how, and what they find frustrating about using current community related tools. Seventeen interviews were conducted with multiplayer gamers to identify why players communicate with others, what they share and how, and what they find frustrating about using current community related tools. Analysis of the interviews indicated different drivers and hurdles to communicating with other gamers. Whilst interviewees used a wide range of existing community related tools, they expressed various frustrations about their use of them including reduced immersion through over-use of UGC, communication difficulties and privacy issues, poor usability of tools, and poor quality control of UGC. In the second phase, an online survey (n=414) aimed to quantify these frustrations and interest in the functionalities to be offered by CNG. The results gave support to the development of a broad range of in-game functions, available without the need to minimise the game screen. Moreover, an initial gamer segmentation indicated that the key CNG functions scheduled for development in the project such as video sharing, chat services and in-game web-browsing, were particularly popular with dedicated online multiplayer gamers. Initial stakeholder research was broadly consistent with gamers' feedback and considered that the in-game live video streaming tool would be of most interest to game developers.

- [5] G. Adam, C. Bouras, V. Kapoulas, and A. Papazois, Providing community and collaboration services to MMOG players, Proc. IADIS Game and Entertainment Technologies 2012 (GET 2012), Lisbon, July 2012.

- [6] G. Adam, C. Bouras, V. Kapoulas, and A. Papazois, Building community and collaboration applications for MMOGs, International Journal of Computer Games Technology, Volume 2012, Article ID 969785, 13 pages.

Audience: The IADIS conference (http://www.gaming-conf.org/) aims to bring together research and practice from creative, social and business practitioners and researchers in this challenging field. The focus of this conference is on design, development and evaluation of games, entertainment technologies and the nature of play.

The International Journal of Computer Games Technology (http://www.hindawi.com/journals/ijcgt/) is a peer-reviewed, open access journal that publishes original research and review articles on both the research and development aspects of games technology covering the whole range of entertainment computing and interactive digital media.

Contribution: The papers describe the high-level design and the system architecture components developed within the CNG system for the support of community and collaboration applications. Furthermore, a detailed design of the system's architectural elements starting with details about the CNG Server's components and continuing with the Web 2.0 applications is given. The interfaces and protocols, as well as the system modules and the internal design specification are described. The focus is on features that distinguish CNG collaboration tools from a typical social networking website. The work also highlights any design issues that can potentially contribute to future standardization processes for WWW and open social networking. Paper [6] is an extended version of paper [5].

- [7] E. Buyukkaya, S. Ahmad, M. Dawood, J. Liu, F. Zhou, R. Hamzaoui, and G. Simon, Level-based peer-to-peer live streaming with rateless codes, to appear in Proc. IEEE International Symposium on Multimedia, Irvine, Cal., Dec. 2012.

Audience: The IEEE International Symposium on Multimedia (ISM2012) is an international forum for researchers to exchange information regarding advances in the state of the art and practice of multimedia computing, as well as to identify the emerging research topics and define the future of multimedia computing.

Contribution: The paper describes a static version of the CNG peer-to-peer system. Peers are arranged in levels so that video is delivered at about the same time to all peers in the same level, and peers in a higher level watch the video before those in a lower level. The video bitstream is encoded with rateless codes and trees are used to transmit the encoded symbols. Trees are constructed to minimize the transmission rate for the source while maximizing the number of served peers and guaranteeing on-time delivery and reliability at the peers. This objective is formulated as a height bounded spanning forest problem with nodal capacity constraint and a solution is computed using a heuristic polynomial-time algorithm. ns-2 simulations are provided to study the trade-off between used bandwidth and video quality for various packet loss rates and link latencies.

- [8] J. Liu, S. Ahmad, E. Buyukkaya, R. Hamzaoui, and G. Simon, Resource allocation in underprovisioned multioverlay live video sharing services, to appear in: Proc. ACM CoNEXT Capacity Sharing Workshop, Nice, Dec. 2012.

Audience: The 8th ACM International Conference on emerging Networking Experiments and Technologies (ACM CoNEXT) 2012 (http://conferences.sigcomm.org/co-next/2012/index.html) will be held in Nice, France from Dec. 10-13, 2012. The goal of the conference is to provide a selective and interdisciplinary forum for research in Networking and Communications. To meet the increasing bandwidth demand of in the Internet especially of mobile users, operators need to seek for new technologies that efficiently utilize their network resource and share the available capacity in a fair manner between all users. Both, data center and next generation access networks provide new requirements on capacity sharing. Different interests should be reconciled, as operators want to be able to exercise control on how the network is used, avoid service disruptions and account for network usage. Users and application providers are interested in a certain level of service quality and consistent experience. The Capacity Sharing Workshop brings together researchers in the area of transport protocols, system designers of fixed and mobile access networks, and their applications to advance the state of research on capacity sharing.

Contribution: To allow a user to simultaneously watch multiple live video streams, the CNG project has developed a multioverlay P2P system consisting of multiple P2P live systems. Some users are sources, which emit a user-generated live video, while the others are peers, which receive one or several videos and participate in diffusing these videos to other peers. Each P2P network contains one source and all peers that have subscribed to its live stream (channel). A major challenge for multioverlay P2P systems is the inter-overlay bandwidth competition problem, which is to find an upload bandwidth allocation between the overlays each peer has subscribed to. So far, no solution has been proposed in the literature for the important case where the overall system is underprovisioned, that is, when peers do not have enough upload bandwidth to ensure a distribution of videos at full quality. The paper shows that an allocation of upload resources that minimizes the wastage of resources (i.e., minimizes the upload bandwidth allocated to overprovisioned overlays) can be computed in polynomial time. It presents a generic model that allows the design of different strategies for the management of the resource deficit. Finally, it provided simulation results to demonstrate the gains in video quality resulting from the implementation of our solutions.

- [9] F. Zhou, S. Ahmad, E. Buyukkaya, R. Hamzaoui, and G. Simon, Minimizing server throughput for low-delay live streaming in content delivery networks, in Proc. ACM NOSSDAV, Toronto, June 2012.

Audience: The 22nd SIGMM Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV) (http://london.csl.toronto.edu/nossdav12/) focuses on both established and emerging research topics, high-risk high-return ideas and proposals, and future research directions in multimedia networking and systems, in a single-track format that encourages active participation and discussions among academic and industry researchers and practitioners.

Contribution: The paper extends the CNG P2P approach to Content Delivery Networks (CDNs). For live streaming, the infrastructure of a CDN comprises three distinct elements. The role of the origin server is to forward the original stream to edge servers. However the origin server has a limited fan-out, i.e., the number of streams it can concurrently emit is limited. The CDN thus uses some intermediate nodes, which are called reflectors or shield caches. While these intermediate nodes have to serve a subset of the thousands of edge-servers, they have a limited upload capacity. The bottleneck in the CDN infrastructure comes either from the origin server that cannot serve the intermediates nodes, or from the intermediate node that cannot serve the edge servers. The problem considered in the paper is how to deliver a live stream to a set of destination nodes with minimum throughput at the source and limited increase of the streaming delay. The proposed solution is based on rateless codes and cooperation among destination nodes. With rateless codes, a node is able to decode a video block of k information symbols after receiving slightly more than k encoded symbols. To deliver the encoded symbols, multiple trees are used where inner nodes forward all received symbols. The goal is to build a diffusion forest that minimizes the transmission rate at the source while guaranteeing on-time delivery and reliability at the nodes. When the network is assumed to be lossless and the constraint on delivery delay is relaxed, we give an algorithm that computes a diffusion forest resulting in the minimum source transmission rate. We also propose an effective heuristic algorithm for the general case where packet loss occurs and the delivery delay is bounded. Simulation results for realistic settings show that with our solution the source requires only slightly more than the video bit rate to reliably feed all nodes.

- [10] S. Ahmad, C. Bouras, E. Buyukkaya, R. Hamzaoui, V. Kapoulas, A. Papazois, A. Shani, and G. Simon, Evaluating P2P live streaming systems: the CNG Case, in Proc. 17th International Conference on Distributed Multimedia Systems (DMS'2011), Florence, Aug. 2011.

Audience: The main theme of the 17th International Conference on Distributed Multimedia Systems (DMS'2011) (http://www.ksi.edu/seke/dms11.html) is multimedia inspired computing. The conference organizers seek contributions of high quality papers, panels or tutorials, addressing any novel aspect of computing (e.g., programming language or environment, data analysis, scientific visualization, etc.) that significantly benefits from the incorporation/integration of multimedia data (e.g., visual, audio, pen, voice, image, etc.).

Contribution: The paper identifies the main challenges in the evaluation of P2P live streaming systems and uses the CNG project as an example to illustrate them. In particular, the paper presents the procedure that will be followed to evaluate the CNG P2P live streaming system. Two phases are described: a laboratory “offline” one using simulation software and an online one based on a real deployment. The online evaluation involves real gamers who provide feedback through questionnaires. The performance of the system is monitored by software, which will collect and provide useful information for further analysis.

3 Demos and Workshops

The CNG P2P system was presented at the Peer-to-Peer (P2P) Techniques for Media Delivery Workshop which took place on 23 February 2012, De Montfort University, Leicester, at the 2011 NEM Summit in Turin, and at the IEEE International Conference on Peer-to-Peer Computing, Tarragona, September 2012.

Peer-to-Peer (P2P) Techniques for Media Delivery Workshop

This workshop presented recent advances in P2P networking, focusing on video communication applications. In addition to talks, a number of demos showing implementations of real P2P systems were presented. The workshop attracted about 50 researchers and practitioners from academia and industry.

•From P2P to Social and Back Again, Prof Jon Crowcroft, University of Cambridge

The talk discussed the history and evolution of content distribution, from download, through multicast, on to packet swarms/torrents and P2P, and then lately to social content.

•P2P live streaming for massively Multiplayer Online Games, Prof Raouf Hamzaoui, De Montfort University

In Massively Multiplayer Online Games (MMOGs), a large number of online players 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. The talk presented a peer-to-peer live video system that allows 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. The system was developed as part of the FP7 CNG project (http://www.cng-project.eu/).

•Pull-based P2P content adaptation and optimised SVC layer selection, Prof Toufik Ahmed, CNRS LaBRI, France

Scalable Video Coding (SVC) content delivery over peer-to-peer (P2P) networks has drawn great interest from its ability to support large number of users, while simultaneously handling client heterogeneity, in terms of download bandwidth, terminal capabilities and user preferences, using content adaptation. In order to support such a system, three essential components need to be considered: overlay construction, layer selection and data scheduling and adaptation. The talk presented a pull-based P2P network for delivering optimised SVC layers, which was developed as part of the FP7 ENVISION project (http://www.envision-project.org/).

•Evaluating P2P as a platform for delivering next-generation TV services, Dr Nick Race, Lancaster University

The talk introduced the recent design and development work for the converged IPTV service that has been deployed in live test-bed (Living Lab) at Lancaster University for thousands of students. High quality audio-visual content is distributed over heterogeneous IP-based content networks, on both set-top box and web-based platforms. Peer-to-Peer (P2P) technologies are exploited to provide energy efficient and low-cost delivery for commercial and user-generated content. The infrastructure and functional components were first presented exploring a number of key designs that facilitate the entire eco-system of content ingest, transcoding, P2P tracking, distribution, statistics, end systems, as well as integration of social networking. Due to the dynamic nature of P2P distribution, a quality measurement service with respect to user experience is also essential for the service evaluation and diagnosis. A multimodal QoE measurement framework which evaluates the IPTV services by collaborating measurements with a variety of different aspects was presented. Results of a use case were also described to verify the effectiveness of the measurement framework in exploiting relevant metrics from service components. The work was done as part of the FP7 P2P-Next project (http://www.p2p-next.org/) and the "RuralConnect Living Lab" (http://www.infolab21.lancs.ac.uk/livinglab/).

•Distribution of Multi-view entertainment using content aware delivery systems, Dr Erhan Ekmekcioglu, University of Surrey

The FP7 DIOMEDES project (http://www.diomedes-project.eu/) focuses on new methods on compression and delivery of multi-view video accompanied by multi-channel audio for realistic and immersive 3D media experiences. Specifically, DIOMEDES project's target is to develop a hybrid Terrestrial DVB and Peer-to-Peer 3D media distribution system, where potential users are eligible to receive either conventional HD or stereoscopic HD service over DVB-T and accompanying camera views using a P2P overlay through broadband access simultaneously. This requires the on-the-fly and terminal synchronisation of multiple audio channels and video camera viewpoints, while allowing real-time immersive 3D media playback. Content aware distribution of the huge amount of multi-view content constitutes a major research challenge. Therefore, the P2P overlay formation and P2P client operation have been developed to take into account content priorities in streaming and also synchronisation with the on-going DVB broadcast. This presentation outlined the major research outcomes delivered in this project, including content aware coding and distribution, with a special emphasis on the developed P2P system.

Exhibition at the 2011 NEM Summit

The CNG system was presented at the 2011 NEM summit in Turin. The exhibition consisted of a poster that showed the results achieved after 20 months and a demo of the CNG platform. As the P2P system was not fully ready at that time, only an offline version was included.

Demo at IEEE International Conference on Peer-to-Peer Computing

The IEEE International Conference on Peer-to-Peer Computing provides a single-track forum for presenting new research results on all aspects of P2P computing. It is the oldest and largest conference dedicated to P2P computing, and has been held every year since 2001. IEEE P2P'12 marked the twelfth anniversary of the conference. The P2P'12 conference in Tarragona, September 2012 (http://www.p2p12.org/) solicited papers on all aspects of large-scale distributed computing. Of particular interest is research that furthers the state-of-the-art in the design and analysis of large-scale distributed applications and systems, or that investigates real, deployed, applications or systems.

The CNG P2P system was presented at the IEEE P2P'12 conference. The presentation consisted of a five-minute talk, a poster, a demo, and a paper. The demo used three laptops 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. A number of 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. The demo paper (2 pages) was published in the proceedings of the conference under:

- [11] S. Ahmad, C. Bouras, E. Buyukkaya, R. Hamzaoui, A. Papazois, A. Shani, G. Simon, and F. Zhou, Peer-to-peer live streaming for massively multiplayer online games, Proc. IEEE P2P'12, Tarragona, Sept. 2012.

4 Conclusion

The scientific activities of the CNG project were presented at major venues such as ACM NOSSDAV, IEEE International Symposium on Multimedia, ACM CoNEXT, IEEE P2P'12, and NEM Summit 2010 and 2011. In addition, three journal papers are in preparation. The first one is an extended version of [7]. It describes the dynamic version of the P2P system and provides extensive simulation results. The second paper is an extended version of [8]. It gives more details, especially about the implementation, adds a major contribution (the distributed algorithm for the fair allocation of bandwidth), and provides more experimental results. The third one is an extended version of [4].

5 Appendix

The appendix contains the following publications.

[1] S. Ahmad, C. Bouras, R. Hamzaoui, A. Papazois, E. Perelman, A. Shani, G. Simon, and G. Tsichritzis, The Community Network Game project: Enriching online gamers experience with user generated content, in Proc. 2nd Int. Conf. Creative Content Technologies (CONTENT 2010), Lisbon, Nov. 2010.

[2] S. Ahmad, C. Bouras, R. Hamzaoui, A. Papazois, E. Perelman, A. Shani, and G. Simon, The Community Network Game project: Enhancing collaborative activities in online games, in Proc. Networked & Electronic Media Summit, Barcelona, Oct. 2010.

[3] S. Ahmad, C. Bouras, R. Hamzaoui, J. Liu, A. Papazois, E. Perelman, A. Shani, G. Simon, and G. Tsichritzis, Community Tools for Massively Multiplayer Online Games, International Journal on Advances in Networks and Services, vol. 4, no 3&4, 2011.

[4] E. Ferrari, J. Lessiter, and J. Freeman, Users and uses of multiplayer games and community activities, in Proc. Networked & Electronic Media Summit, Turin, Sept. 2011.

[5] G. Adam, C. Bouras, V. Kapoulas, and A. Papazois, Providing community and collaboration services to MMOG players, in Proc. IADIS Game and Entertainment Technologies 2012 (GET 2012), Lisbon, July 2012.

[6] G. Adam, C. Bouras, V. Kapoulas, and A. Papazois, Building community and collaboration applications for MMOGs, International Journal of Computer Games Technology, Volume 2012, Article ID 969785, 13 pages.

[7] E. Buyukkaya, S. Ahmad, M. Dawood, J. Liu, F. Zhou, R. Hamzaoui, and G. Simon, Level-based peer-to-peer live streaming with rateless codes, to appear in Proc. IEEE International Symposium on Multimedia, Irvine, Cal., Dec. 2012.

[8] J. Liu, S. Ahmad, E. Buyukkaya, R. Hamzaoui, and G. Simon, Resource allocation in underprovisioned multioverlay live video sharing services, to appear in Proc. ACM CoNEXT Capacity Sharing Workshop, Nice, Dec. 2012.

[9] F. Zhou, S. Ahmad, E. Buyukkaya, R. Hamzaoui, and G. Simon, Minimizing server throughput for low-delay live streaming in content delivery networks, in Proc. ACM NOSSDAV, Toronto, June 2012.

[10] S. Ahmad, C. Bouras, E. Buyukkaya, R. Hamzaoui, V. Kapoulas, A. Papazois, A. Shani, and G. Simon, Evaluating P2P live streaming systems: the CNG Case, in Proc. 17th International Conference on Distributed Multimedia Systems (DMS'2011), Florence, Aug. 2011.

[11] S. Ahmad, C. Bouras, E. Buyukkaya, R. Hamzaoui, A. Papazois, A. Shani, G. Simon, and F. Zhou, Peer-to-peer live streaming for massively multiplayer online games, in Proc. IEEE P2P'12, Tarragona, Sept. 2012.

The Community Network Game Project: Enriching Online Gamers Experience with User Generated Content

Shakeel Ahmad*, Christos Bouras†, Raouf Hamzaoui*, Andreas Papazois†, Erez Perelman‡, Alex Shani ‡,
Gwendal Simon §, George Tsichritzis †
*Department of Engineering
De Montfort University, Leicester, UK
sahmad@dmu.ac.uk, rhamzaoui@dmu.ac.uk
†Research Academic Computer Technology Institute
N. Kazantzaki, GR26500 Patras, Greece
bouras@cti.gr, papazois@ceid.upatras.gr, tsixritzis@cti.gr
‡Exent Technologies
Bazel 25, 49125 Petach-Tikva, Israel
eperelman@exent.com, ashani@exent.com
§Institut TELECOM
TELECOM Bretagne, France
gwendal.simon@telecom-bretagne.eu

***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. We present the EU funded Community Network Game (CNG) project. The CNG project will provide tools to enhance collaborative activities between online gamers and will develop new tools for the generation, distribution and insertion of user generated content (UGC) into existing MMOGs. CNG will allow the addition of new engaging community services without changing the game code and without adding new processing or network loads to the MMOG central servers. The UGC considered by the CNG project includes 3D objects and graphics as well as video to be shared using peer-to-peer (P2P) technology. We describe the concept, innovations, and objectives of the project.**

***Keywords*-Massively Multiplayer Online Games; user generated content; P2P streaming; graphics insertion.**

I. Introduction

Massively Multiplayer Online Games (MMOGs) allow a large number of online users (in some cases millions) to inhabit the same virtual world and interact with each other in a variety of collaborative and competing scenarios. MMOGs are rapidly gaining in popularity. Data from [1] suggests that there were over 16 million active subscriptions to MMOGs by 2008, a figure that is growing fast and predicted to rise to at least 30 million by 2012.

MMOG gamers can build and become members of active communities with mutual interests, shared adventures, and common objectives. Players can play against other players (player versus player) or build groups (guilds) to compete against other groups (realm versus realm) or against computer-controlled enemies. This paper presents the Community Network Game (CNG) project [2], a recently EU funded project within the Seventh Framework Programme. The project, which started in February 2010 and has a duration of 30 months, aims at enhancing collaborative activities between MMOG gamers. This will be achieved by developing new tools for the generation, distribution and insertion of user-generated content (UGC) into existing MMOGs. This UGC may include items (textures, 3D objects) to be added to the game, live video captured from the game screen and streamed to other players, and videos showing walk-throughs, game tutorials, or changes in the virtual world to be watched on demand.

The main technologies proposed by the CNG project are the in-game graphical insertion technology (IGIT) and a peer-to-peer (P2P) system for the distribution of live video. IGIT is an innovative technology of replacing or inserting content into a game in real time without the need to change the game code in the client or server. For example, billboards can be inserted, tattoos can be added to in-game characters, an area on the screen can be assigned to display user information, and any type of window (browser, chat, etc.) can be inserted floating on or outside the game area. The technology can be implemented on multiple games, making it possible to create a community that is not limited to a

This paper was published in Proc. 2nd Int. Conf. Creative Content Technologies (CONTENT 2010), Lisbon, Nov. 2010.

specific game or publisher.

Enabling thousands of users to communicate UGC represents a significant challenge to networks already occupied by the MMOG client-server data. The CNG project intends to develop new techniques for UGC distribution that are friendly (supportive and not disruptive) to the MMOG client-server traffic. The key innovation will be a P2P system that will allow MMOG gamers to stream live video of the game without interrupting the MMOG data flow and the need to upload the video data to a central server.

The remainder of the paper is organized as follows. Section II gives an overview of the state-of-the art in the areas related to the CNG project. Section III presents CNG's proposed innovations and technologies. Section IV concludes the paper by discussing CNG's benefits and expected impacts.

II. Related Work

A. UGC and Web collaboration tools

UGC refers to various kinds of media content that are produced by end-users. In the context of a game, this may refer to screen captures and video capture from within the video game. Another example of UGC may be the various mods created by the users. Furthermore, sharing and remixing UGC is a widespread online activity that crosses borders of age and gender. Avid players go to great lengths in their efforts to create shared content in which they reveal their mastery. Additional data layers are always included: narration, animation and primarily soundtrack. UGC sharing and remixing within game platforms, one of the most important goals of the CNG project, is currently not supported. Most MMOG-based UGC content is confined to dedicated player/game company sites as in World of Warcraft [3]. Many MMOG games also have their own community pages in social networking sites such as Facebook [4]. In April 2010, Facebook released significant updates to its API by allowing external websites to uniformly represent objects in the graph (e.g., people, photos, events, and community pages) and the connections between them (e.g., friend relationships, shared content, and photo tags). As a result, the Facebook API [5] can provide an unprecedented bridge between gamespaces and the social web. Additionally, many MMOG players use sites such as YouTube in order to share their game-based UGC. In 2008, Maxis incorporated YouTube APIs within their game, Spore, by enabling a player to upload video of their creations to their YouTube account with only two clicks [6].

Web 2.0 is a trend in the use of World Wide Web (WWW) technology and Web design that aims to facilitate creativity, information sharing, and, most notably, collaboration among users. These concepts have led to the development and evolution of Web-based communities and hosted services, such as social networking sites, wikis, blogs, and folksonomies. The Web 2.0 technologies are standardized by the WWW Consortium (W3C) [7]. Although the Web 2.0 term suggests a new version of the WWW, it does not refer to an update to any technical specifications, but to changes in the ways software developers and end-users use the web. The Web 2.0 based collaboration applications may include instant messaging, audio and video chat, file sharing and online voting and polling. For audio/video capturing and playback the Flash software platform [8] is commonly deployed. Other solutions are the Java Applet technology or standalone applications which run on Web browser and offer interoperability over different platforms. For instant messaging, online polling/voting and file sharing, Asynchronous JavaScript and XML (AJAX) [9] are commonly used. AJAX allows Web applications to retrieve data from the server asynchronously in the background without interfering with the display and behavior of the existing page. The use of AJAX techniques has led to an increase in interactive or dynamic interfaces on webpages. Finally, for WWW client-server communication, most of the Web 2.0 applications are based on Simple Object Access Protocol (SOAP) [10]. SOAP relies on XML as its message format, and usually relies on other Application Layer protocols, most notably the Remote Procedure Call (RPC) and HTTP.

Table I: Game adaptation technologies in FreeRideGames (FRG), Massive Incorporated (MI), PLAYXPERT (PX), Xfire (XF), Double Fusion (DF).

Product	FRG	MI	PX	XF	DF
In-game overlay	Yes	No	Yes	Yes	No
Game resize	Yes	No	No	No	No
Texture replacement	No	Yes	No	No	Yes
Need for SDK	No	Yes	No	No	Yes

B. Game adaptation technologies

In-game technologies have been used in the gaming market for several years. The gaming industry has adopted these technologies to increase its revenue by finding more financial sources and by attracting more users. In-game overlay allows to view and interact with windows outside the game, but without "Alt-Tabbing". It does so by rendering the window inside the game. Texture replacement enables to replace an original game texture with a different texture. In this way, the newly placed textures are seen as part of the original game content. This method is commonly used for dynamic in-game advertisement. Game size modification technology adapts the original game by decreasing its original size and surrounding it with an external content. The existing game adaptation products can be divided into two groups: products that require for the game developer to integrate the products software development kit (SDK) and products that do not impose this constraint (see Table I).

C. P2P live video systems

Traditional client-server video streaming systems have critical issues of high cost and low scalability on the server.

P2P networking has been shown to be cost effective and easy to deploy. The main idea of P2P is to encourage users (peers) to act as both clients and servers. A peer in a P2P system not only downloads data, but also uploads it to serve other peers. The upload bandwidth, computing power and storage space on the end user are exploited to reduce the burden on the servers.

Viewers of a live event wish to watch the video as soon as possible. That is, the time lag between the video source and end users is expected to be small. In a live streaming system, the live video content is diffused to all users in real time and video playback for all users is synchronized. Users that are watching the same live video can help each other to alleviate the load on the server. P2P live streaming systems allow viewers to delete the historic data after the playback, and hence have no requirement for any data storage and backup.

Based on the overlay network structure, the current approaches for P2P live streaming systems can be broadly classified into two categories: tree-based and mesh-based. In tree-based systems, peers form an overlay tree, and video data are pushed from the parent node to its children. However, a mesh-based system has no static streaming topology. Peers pull video data from each other for content delivery. Over the years, many tree-based systems have been proposed and evaluated, however, never took of commercially. Mesh-based P2P streaming systems achieve a large-scale deployment successfully, such as PPLive [11], PPStream [12], etc.

Most P2P live video systems rely on the transmission control protocol (TCP) (as in e.g., CoolStreaming, PPStream). TCP guarantees reliable transmission of the data by automatic retransmission of lost packets. However, as TCP requires in order delivery of the data and keeps on retransmitting a packet until an acknowledgement is received, significant delays may be introduced. Further delays are caused by the congestion control algorithm used by TCP, which reacts to packet loss by reducing the transmission rate, leading occasionally to service interruption. This presents a serious drawback for real-time video communication where the data must be available to the receiver at its playback time. Lost and delayed packets that miss their playback deadline not only are useless, they also consume the available bandwidth unnecessarily. An alternative to TCP is to use UDP as the transport protocol and apply application-layer error control. This includes UDP without error control (PPLive, TVAnts), UDP with FEC [13], ARQ [14], and Multiple Description Coding (MDC) [15].

III. Technologies and Innovations

To achieve its objectives, CNG will rely on innovative software technologies and a P2P live video system. While the MMOG architecture is not modified (the game content and the game data are still transferred through the MMOG servers), the following components will be added (Fig. 1): (i)

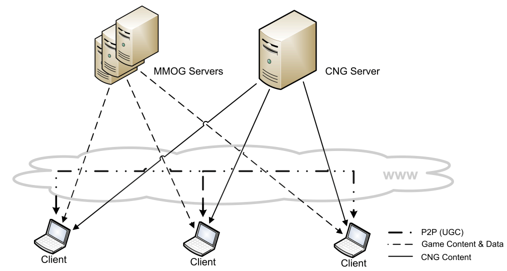


Figure 1. CNG architecture.

Sandbox on the client side that is responsible for modifying the game environment; (ii) CNG Server for monitoring the P2P UGC communication. The CNG server acts as a tracker for the system in the sense that it is in charge of introducing peers to other peers. It has persistent communication with the clients and manages the organization of the P2P exchanges.

A. UGC and Web collaboration tools

In CNG, the participation in community activities will not require closing or resizing the screen of the game and activating the tools' window. Instead, the CNG tools' window, will be integrated into the MMOG application environment. The CNG tools will use Web 2.0 technology to enable voice and video chat, instant messaging, polling, and file sharing. A flash-based collaborative video editor will be included in the CNG toolbox to allow users to edit videos and images. The system also includes tools to enable the upload of video files to social networking sites.

B. IGIT

The CNG project will enable to resize the game and surround it with external content, overlay the game, and replace an existing game texture with an external content. This will be done in a way that does not harm the game experience and without the need for SDK integration. Fig. 2 and Fig. 3 illustrate some of these features. Fig. 2 is a screenshot from the MMOG game "Roma Victor" [16] by RedBedlam. Fig. 3 shows the same game scene with a mock-up of CNG features. The modifications, which are numbered in Fig. refigit, are as follows: (1) The original resolution of the game was modified to enable an additional frame around the game to hold the in-frame objects. IGIT uses the GPU of the user's machine for changing the resolution of the game to avoid reduction in the image quality; (2) Instant messaging window as an example of active Web 2.0 application; (3) Web browser that presents online passive information (in this example, a leader board); (4) Another Web browser window that presents an updated advertisement; (5) MMOG specific chat to enable the users in a specific scene to cooperate; (6) In-game 3D UGC. In this example, a user added a note on a tree to publish an eBay auction; (7) Two windows of a video chat with casual friends or cooperative players.

Figure 2. Original MMOG screenshot.

Figure 3. IGIT-modified MMOG screenshot.

The choices of which application to use and the applications' screen location are under the control of the user (player).

C. P2P live video system

In existing MMOGs, a player can capture the video of the game and send it to a central server which broadcasts it live to other users [17]. However, this solution, which heavily relies on central servers has many drawbacks such as high costs for bandwidth, storage, and maintenance. Moreover, this solution is not easily scalable to increasing number of users. The CNG project intends to develop a P2P live video streaming system to address the limitations of server-based solutions. The CNG P2P live video system will allow every peer to become a source of a user-generated video stream for a potentially large set of receivers. While many P2P live video systems have been proposed, none of them has been specifically designed for the unique environment of MMOGs. In particular, none of the existing P2P live video systems addresses the following challenges:

- Any MMOG player should be able to multicast live video. The video can potentially be received by any other player in the P2P network.
- 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).
- Live video should be delivered at about the same time to all peers at the same "level". This optional requirement can be useful in some situations. For example, a level can be defined as the set of all peers that are in the same region of the virtual space.

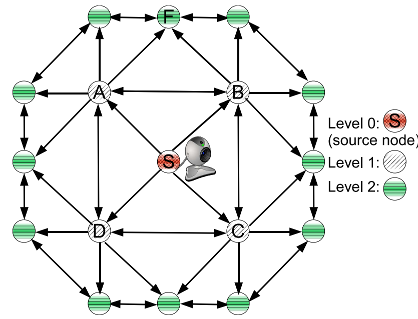


Figure 4. P2P topology. S is the source peer.

The CNG P2P live video system is designed as follows. A mesh-based topology is used for the P2P overlay. Peers are organized in different levels (Fig. 4). The source peer is placed at level 0. All peers connected directly to the source peer are at level. In general, a peer is considered to be at level j if its shortest route to the source peer consists of j intermediate links.

The video is captured in real time from the source screen, compressed, and partitioned into source blocks. Each block consists of one GOP (Group of Pictures) and is an independent unit of fixed playback duration (e.g., 1 s). The UDP protocol is used as the transport protocol. The source peer applies rateless coding on each source block and keeps on sending the resulting encoded symbols in encoded packets (packets of encoded symbols) to level-1 peers until it receives an acknowledgment. Level-1 peers forward the received packets to other level-1 peers immediately as instructed by the source peer. Level-1 peers also forward the received packets to the level-2 peers that are directly connected to them, etc. When a level-1 peer completes the decoding of a block, it sends an acknowledgment to all senders so that they stop sending it packets. Then it applies rateless coding on the decoded block to feed level-2 peers. Thus, each receiving peer has two phases: forwarding (before the decoding is successful) and encoding (after decoding the block). In the first phase, the receiving peer just forwards the received packets to the next level peers connected to it, while in the encoding phase, it generates encoded symbols from the decoded block and feeds the next-level peers.

The source peer computes a scheduling strategy for each source block. The strategy specifies the maximum number of encoded packets n that can be sent for this block, and the time t_i at which packet i is sent with a hierarchical forwarding scheme $F_i,\ i = 1, 2, \ldots, n$. If the source peer receives an acknowledgment from a level-1 peer j before n packets are sent, it can update its scheduling strategy by, e.g., removing peer j from the forwarding schemes of the remaining packets. An example of a scheduling strategy for $n = 4$ and four level-1 peers A to D is as follows. $1 : t_1 : A \rightarrow B(\rightarrow D) + C, 2 : t_2 : B \rightarrow A + D(\rightarrow C), 3 : t_3 : C \rightarrow A(\rightarrow B) + D, 4 : t_4 : D \rightarrow C(\rightarrow A) + B$. For packet 1, the strategy says: transmit at time t_1 to A. A forwards the packet to B and C. B forwards it to D.

The complexity of the scheduling strategy depends on the neighbourhood relationships. In a clustered topology (where neighbours of a given peer are also neighbours with high probability), the scheduling strategy can become complex to decide. One of the challenges of the project consists of determining topologies, which allow efficient and simple computation of scheduling strategies.

Since a peer can have multiple neighbours, it can receive the same packet from multiple senders. To avoid this, a parent should know the other parents of its children. For example, peer B should know that F is its common child with A and not forward a packet to F if this packet has previously visited A. In the encoding phase, receiving duplicate packets can be avoided with high probability by forcing peers to use different seed values for the rateless code.

By having multiple senders, lost packets on one link can be compensated for by receiving more packets on other links. Players that are neighbours in the virtual world can be placed at the same level in the mesh, so that they can watch the video with approximately the same playback lag with respect to the original source.

Our system extends previous ideas proposed in [18], [19]. However there are many important differences between these works and our scheme. For example, the systems of [18], [19] do not have the notion of scheduling strategy and use a different approach to minimise the number of received duplicated packets. Also in [18], [19], there is no notion of levels within the mesh.

As UDP does not have a built-in congestion control mechanism, a pure UDP-based application may overwhelm the network. To address this problem, we aim to adapt the UDP sending rate according to receiver feedback. The feedback may consist of the average packet loss rate and the forward trip time (FTT). If the average FTT and loss rate are higher than threshold values, this is a strong indication of congestion. As a result, the sender has to adapt the sending rate accordingly.

Many peers can become a source of live content. However, a peer cannot participate in all overlays, because some resources are used in every overlay a peer belongs to. In practice, a user can decide whether to receive the stream from a given source. But an automatic management would be more suitable. We propose to continuously adjust the set of peers who are targeted to receive data from a source. The goal is to obtain a fair nearly congested system, where the peers that receive the stream are "close" to the user who generates the content.

We use the concept of Area of Interest (AoI) [20] for that purpose. An AoI is defined as the part of the virtual world around a user that generates content. When a peer is within the AoI of a user generating high-quality content, or when it belongs to many AoIs, it may experience congestion. The challenge is to design a mechanism for determining the best size of these AoIs, that is, a size such that the maximum amount of UGC is practically delivered while no user experiences congestion. The management of the AoI must then take into account the popularity of the virtual place and the capacity of the devices of the players that are located there. Such a management has been shown to be hard in wireless sensor networks [21], but some heuristics can perform well. For a player, the decision of increasing or decreasing the size of the AoI should be based on feedbacks from other nearby players in a collective manner.

Two strategies can be implemented. In the first one, one peer is congested, and not all peers in an AoI can be served. However, the capacity of peers in the surroundings of the congested peers makes that a new computation of AoI for all sources is not necessary. Instead, it is possible to "pass" one peer from one P2P overlay to another P2P overlay, so that the capacity provided by this peer can tackle the congestion issue. This strategy, which avoids heavy computations can solve local small congestion problems. The second strategy can be implemented when this first one fails. A process similar to the one that ensures fair resource sharing in TCP can be used. Every source periodically tries to increase (in an additive manner) the size of its AoI until congestion is detected. Then, the radius of the AoI is decreased in a multiplicative manner (see [22] for a similar technique).

In addition to designing an efficient P2P live video streaming system for a game environment, the CNG project proposes to contribute to a better understanding of the general problem of the diffusion of multiple video streams in a constrained environment. The goal is to maximize the amount of peers receiving content in an environment where not all peers can be reached because too few resources are available. If we assume tree overlays and consider only one video stream, the problem is to build a tree that spans the maximum number of peers with the constraint that every peer can only serve a limited number of other peers. In the context of many concurrent video streams, the problem becomes even harder with a constrained forest.

The building of degree-constrained trees is an NP-hard problem [23]. We propose to contribute to the analysis of the computational properties of this problem. In particular, the formulation of the problem into several Integer Programming models, and comprehensive benchmarks of these models will enable the computation of optimal solutions on small instances of the problem. Besides, we aim at designing heuristic algorithms, which allow the computation of nearly optimal solutions for large problem instances, as well as approximate algorithms (algorithms that compute solutions that are proved to be never far to the optimal solutions).

IV. Conclusion

We presented the EU funded CNG project. CNG will support and enhance community activities between MMOG gamers by enabling them to create, share, and insert UGC. The UGC considered by the CNG project includes 3D objects, graphics, and video. CNG will develop in-game community activities using an in-game graphical insertion technology (IGIT). IGIT allows to replace or insert content in real time without the need to change the game's code in the client or server. CNG uses an architecture that efficiently combines the client-server infrastructure for the MMOG activities with a P2P overlay for the delivery of live video. The video traffic represents a real challenge to the network already occupied by the MMOG client-server data. The project will research and develop new techniques for P2P live video streaming that are friendly to the MMOG client-server traffic. Since video can be resource heavy, the network indirectly benefits from the increased locality of communication. CNG will also provide Web 2.0 tools for audio and video chat, instant messaging, in-game voting, reviewing, and polling. This will reduce the need for visiting forums outside the game and diluting the MMOG experience.

CNG has the potential to provide huge benefits to MMOG developers and operators. New community building tools will be offered cost-effectively and efficiently, without the need to redesign or recode the existing game offerings. The user experience will be enriched, and the needs of the end-users will be better addressed. The community will be brought into the content, and the game communities will become more engaged, reducing churn to other MMOGs. New income streams will be delivered with the help of in-game and around game advertising. Yet, MMOG developers and operators will be able to maintain control over how various commercial and UGC content is displayed, thus keeping editorial control of the look and feel of their MMOG.

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 at: http://www.mmogchart.com. Last accessed: 24/8/2010.

[2] [Online]. Available at: http://www.cng-project.eu/. Last accessed: 24/8/2010.

[3] [Online]. Available at:: http://www.worldofwarcraft.com/splash-sc2date.htm. Last accessed: 25/5/2010.

[4] [Online]. Available at: http://www.Facebook.com/group.php?gid=2210867052. Last accessed: 25/5/2010.

[5] Facebook API. [Online]. Available at: http://graph.Facebook.com. Last accessed: 25/5/2010.

[6] Youtube API. [Online]. Available at: http://code.google.com/apis/youtube/casestudies/ea.html. Last accessed: 30/6/2010.

[7] W3C. [Online]. Available at: http://www.w3.org/. Last accessed: 25/5/2010.

[8] Adobe Flash. [Online]. Available at: http://www.adobe.com/support/flash/downloads.html. Last accessed: 26/8/2010.

[9] XMLHttpRequest, W3C Working Draft, 2009. [Online]. Available at: http://www.w3.org/TR/2009/WD-XMLHttpRequest-20091119/. Last accessed: 26/8/2010.

[10] SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), W3C Recommendation, 2007. [Online]. Available at: http://www.w3.org/TR/soap12-part1. Last accessed: 26/8/2010.

[11] [Online]. Available at: http://www.PPlive.com. Last accessed: 26/8/2010.

[12] [Online]. Available at: http://www.PPstream.com. Last accessed: 26/8/2010.

[13] V.N. Padmanabhan, H.J. Wang, and P.A. Chou, "Resilient peer-to-peer streaming," Proc. IEEE ICNP, pp. 16–27, Atlanta, GA, Nov. 2003.

[14] E. Setton, J. Noh, and B. Girod, "Rate-distortion optimized video peer-to-peer multicast streaming," Proc. P2PMMS'05, pp. 39–48, Singapore, Nov. 2005.

[15] E. Akyol, A.M. Tekalp, and M.R. Civanlar, "A flexible multiple description coding framework for adaptive peer-to-peer video streaming," IEEE Journal of Selected Topics in Signal Processing, vol. 1, no. 2, pp. 231–245, Aug. 2007.

[16] [Online]. Available at: http://www.roma-victor.com. Last accessed 23/8/10.

[17] [Online]. Available at: http://xfire.com. Last accessed 25/8/10.

[18] C. Wu and B. Li, "rStream: resilient and optimal peer-to-peer streaming with rateless codes," IEEE Transactions on Parallel and Distributed Systems, vol. 19, pp. 77–92, Jan. 2008.

[19] M. Grangetto, R. Gaeta, and M. Sereno, "Reducing content distribution time in P2P-based multicast using rateless codes," Proc. Italian Networking Workshop, Cortina, pp. 1–12, Jan. 2009.

[20] J.R. Jiang, Y.-L. Huang, and S.-Y. Hu, "Scalable AOI-cast for peer-to-peer networked virtual environments," Journal of Internet Technology, vol. 10, no. 2, pp. 119–126, 2009.

[21] B. Han and J. Leblet, and G. Simon, "Query range problem in wireless sensor networks," IEEE Comm. Letters, vol. 13, no. 1, pp. 55–57, 2009.

[22] B. Han and G. Simon, "Fair capacity sharing among multiple sinks in wireless sensor networks," Proc. IEEE MASS Conference, Pisa, pp. 1–9, Oct. 2007.

[23] J. Könemann and R. Ravi, "Primal-dual meets local search: approximating MST's with nonuniform degree bounds," Proc. 35th Annual ACM Symposium on Theory of Computing, San Diego, CA, pp. 389–395, 2003.

The Community Network Game Project: Enhancing Collaborative Activities in Online Games

Shakeel Ahmad*, Christos Bouras†, Raouf Hamzaoui*, Andreas Papazois†, Erez Perelman‡, Alex Shani ‡, Gwendal Simon §

*Department of Engineering, De Montfort University, Leicester, UK
sahmad@dmu.ac.uk, rhamzaoui@dmu.ac.uk

†Research Academic Computer Technology Institute
N. Kazantzaki, GR26500 Patras, Greece
bouras@cti.gr, papazois@ceid.upatras.gr

‡Exent Technologies
Bazel 25, 49125 Petach-Tikva, Israel
eperelman@exent.com, ashani@exent.com

§Institut TELECOM
TELECOM Bretagne, France
gwendal.simon@telecom-bretagne.eu

***Abstract*: The EU-funded Community Network Game (CNG) project aims at enhancing collaborative activities between Massively Multiplayer Online Games (MMOG) players. This will be achieved by developing new tools for the generation, distribution and insertion of user-generated content (UGC) into existing MMOGs without changing the game code and without adding new processing or network loads to the MMOG central servers. This UGC may include items (e.g., textures, 3D objects) to be added to the game and live video captured from the game screen and streamed to other players. We present the objectives of the project, focusing on its main scientific and technological contributions.**

Keywords: Massively Multiplayer Online Games; user generated content; P2P streaming; graphics insertion.

1 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 collaborating and competing scenarios. MMOG gamers become members of active communities with mutual interests, shared adventures, and common objectives. Players can play against other players (player versus player) or build groups (guilds) to compete against other groups (realm versus realm) or against computer-controlled enemies.

MMOGs are rapidly gaining in popularity. Data from [1] suggests that there were over 16 million active subscriptions to MMOGs by 2008, a figure that is growing fast and predicted to rise to at least 30 million by 2012.

This paper presents on-going work within the Community Network Game (CNG) project [2], a recently EU funded project within the Seventh Framework Programme. The project, which started in February 2010 and has a duration of 30 months, aims at enhancing collaborative activities between MMOG gamers. This will be achieved by developing new tools for the generation, distribution and insertion of user-generated content (UGC). This UGC may include video captured from the game screen to be streamed live for cooperative play, coaching, and experience sharing, as well as 3D graphics and images to be inserted into the game.

The main innovations proposed by the CNG project are the in-game graphical insertion technology (IGIT) and a peer-to-peer (P2P) system for live video streaming. IGIT can insert content into a game in real time without the need to change the game code in the client or server. For example, billboards can be inserted, an area on the screen can be assigned to display user information, and any type of window (Web browser, instant messaging, etc) can be inserted floating on or outside the game area. The technology can be implemented on multiple games, making it possible to create a community that is not limited to a specific game or publisher.

Fig. 1 and 2 show some possibilities of the usage of IGIT. Fig. 1 is a screenshot from the MMOG game "Roma Victor" [3] by RedBedlam. Fig. 2 shows the same game scene with a mock-up of some of CNG capabilities. The modifications, which are numbered in Fig. 2, are as follows: (1) The original resolution of the game was modified to enable an additional frame around the game to hold the in-

Figure 1. Original MMOG screenshot.

Figure 2. IGIT-modified MMOG screenshot.

frame objects. IGIT uses the GPU of the user's machine for changing the resolution of the game to avoid reduction in the image quality; (2) Instant messaging window as an example of active Web 2.0 application; (3) Web browser that presents online passive information (a leader board here); (4) another web browser window that presents an updated advertisement; (5) MMOG specific chat to enable the users in a specific scene to cooperate; (6) In-game 3D UGC. In this example, a user added a note on a tree to publish an eBay auction; (7) two windows of a video chat with casual friends or cooperative players. The choices of which application to use and the applications' screen location are under the control of the user (player).

Enabling thousands of users to communicate live in-game video represents a significant challenge to networks already occupied by the MMOG client-server data. The CNG project intends to develop new techniques for live video distribution that are "friendly" (supportive and not disruptive) to the MMOG client-server traffic. The key innovation will be a P2P system that will allow MMOG gamers to share video without interrupting the MMOG data flow and the need to upload the data to a central server. While the generic MMOG architecture is not modified (the game content and the game data are still transferred through the MMOG servers), the following components will be added: (i) Sandbox on the client side that is responsible for modifying the game environment; (ii) CNG Server for monitoring the P2P communication (Fig. 3). The CNG server acts as a tracker for the system in the sense that it is in charge of introducing peers to other peers. It has persistent communication with the clients and manages the organization of the P2P exchanges. While the CNG server is not dedicated to supply users with content, it can in a few well-defined cases act as a backup storage for the UGC of users, e.g., when these players suffer from transient network troubles or when a player that holds a specific data leaves the scene.

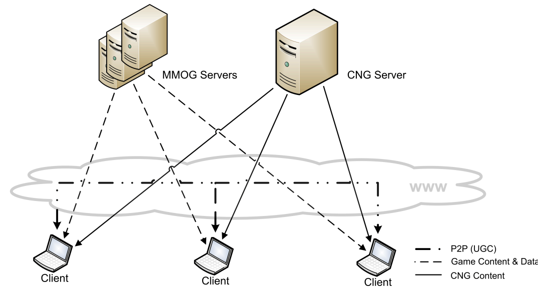


Figure 3. CNG architecture.

In addition to the IGIT technology and the P2P system, the CNG project will offer to MMOG gamers Web 2.0 online collaboration services and video capturing and editing tools.

The remainder of the paper is organized as follows. Section 2 presents the IGIT technology. Section 3 discusses the MMOG-friendly P2P live video system. Section 4 briefly presents the CNG Web 2.0 online collaboration tools. In Section 5, the expected impact of the project is discussed and conclusions are drawn.

2 IGIT

Game adaptation technologies have been used in the gaming market for several years. The gaming industry has adopted these technologies to increase its revenue by finding more financial sources and by attracting more users. In-game overlay allows to view and interact with windows outside the game, but without "Alt-Tabbing". It does so by actually rendering the window inside the game. Texture replacement enables to replace an original game texture with a different texture. In this way, the newly placed textures are seen as part of the original game content. This method is commonly used for a dynamic in-game advertisement. Game size modification technology adapts the original game by decreasing its original size and surrounding it with an external content. Some of the existing game adaptation products require for the game developer to integrate the

Product	FRG	MI	PX	XF	DF
In-game overlay	Yes	No	Yes	Yes	No
Game resize	Yes	No	No	No	No
Texture replacement	No	Yes	No	No	Yes
Need for SDK	No	Yes	No	No	Yes

Table I
GAME ADAPTATION TECHNOLOGIES IN FREERIDEGAMES (FRG), MASSIVE INCORPORATED (MI), PLAYXPERT (PX), XFIRE (XF), DOUBLE FUSION (DF).

product's software development kit (SDK). Table I lists the main game adaptation products.

The CNG project intends to combine all the existing methods for in-game adaption but with no need for SDK integration. In this way, we will be able to surround the game, overlay it or replace an existing game texture with an external content. We will use those techniques in a way that will not harm the game experience. The CNG project will combine the following game adaption technologies:

1) Texture replacement. This will consist of the three following techniques: Spots Identification - we will identify the relevant spots in the game which are suitable for replacement; we will enable the user to replace 2D textures in the identified spots; we will also enable to manipulate 3D objects in those spots.
2) Game size modification. We will provide the ability to decrease the game window size and surround it with new windows.
3) Game overlay. We will implement a windows system with a predefined fixed layouts in order to give the user the ability to manage his Web/Web 2.0 applications. The windows system will surround the game as an overlay.

3 MMOG-FRIENDLY P2P LIVE STREAMING

The existing solution to stream live video of an MMOG game is to capture the video of the game from the screen and send it to a central server which broadcasts it live [4]. However, this solution, which heavily relies on central servers has many drawbacks such as high costs for bandwidth, storage, and maintenance. Moreover, this solution is not easily scalable to increasing number of users. The CNG project intends to develop a P2P live streaming system for MMOGs that addresses the limitations of the server-based solution. While many P2P live video systems have been proposed, none of them has been specifically designed for the unique environment of MMOGs. Potential scenarios for live video streaming in MMOGs include:

- Scenario 1: A player broadcasts live screen-captured video of its game to any other player.
- Scenario 2: A player streams live screen-captured video of its game to a restricted group (guild).

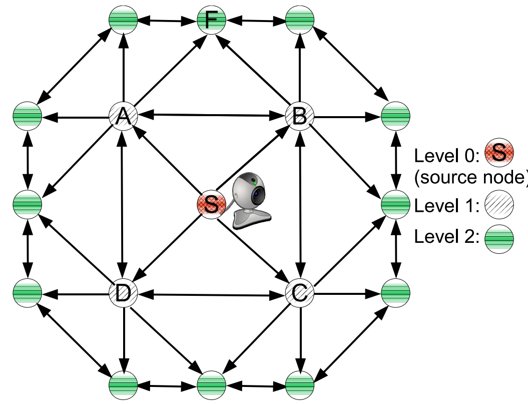


Figure 4. P2P topology. S is the source peer.

- Scenario 3: A player streams animated virtual 3D objects. The "clients" are players whose virtual position is close to the virtual position of the object.

To enable these scenarios, the P2P system needs to address the following challenges:

- Many MMOG players can simultaneously emit different live streams, so the P2P overlay consists of many concurrent P2P overlays. A peer cannot participate in all P2P overlays because some of its resources will be used in every overlay it belongs to. The challenge for a user is to adequately allocate its physical resources, including upload and download bandwidth. These resources are limited, so they have to be shared carefully. Moreover, a player has additional connections with the MMOG game server, which must be given highest priority.
- Live video should be delivered at about the same time for all peers at the same "level". For example, a level can be defined as the set of MMOG players that are in the same region of the virtual world.

The solution proposed by the CNG P2P live video system is as follows. The video is captured in real time from the source screen, compressed, and partitioned into source blocks. Each source block corresponds to one GOP (Group of Pictures) and is an independent unit of fixed playback duration (e.g., 1 s). A mesh-based topology is used for the P2P network. Peers are organized in different levels of the mesh (Fig. 4).

The source peer is placed at level 0. All peers directly connected to the source peer are at level 1 and the peers that are not connected to the source peer but are connected to level-1 peers are at level 2. In general, a peer is considered to be at level j if its shortest route to the source peer consists of j intermediate links.

The UDP protocol is used as the transport protocol. The source peer applies rateless coding on each source block and keeps on sending the resulting encoded symbols in encoded packets (packets of encoded symbols) to level-1 peers until it receives an acknowledgment from them or a timeout occurs. As the acknowledgment needs time to reach the sender, the sender may transmit redundant encoded symbols, which results in significant bandwidth wastage. Thus, a major challenge is to determine packet scheduling algorithms that minimise this overhead. This optimization problem was recently introduced for a client-server model [5]. An optimal strategy typically consists of a sequence of successive transmission bursts and waiting times. We propose to extend this work by devising solutions for the more challenging context of a P2P network.

The source peer computes a scheduling strategy for each source block. The strategy specifies the maximum number of encoded packets n that can be sent for this source block, and the time t_i at which packet i is sent with a hierarchical forwarding scheme $F_i,\ i = 1, 2, \ldots, n$.

If the source peer receives an acknowledgment from a level-1 peer j before n packets are sent, it can update its scheduling strategy by, e.g., removing peer j from the forwarding schemes of the remaining packets.

An example of a scheduling strategy for $n = 5$ and the four level-1 peers of Fig. 4 is as follows.

- $1 : t_1 : A \to B$
- $2 : t_2 : B \to A + C$
- $3 : t_3 : C \to B + D(\to A)$
- $4 : t_4 : A \to D(\to C)$
- $5 : t_5 : D \to A(\to B)$

The strategy says that packet 1 should be sent at time t_1 to peer A, which should forward it to peer B. Packet 2 should be sent at time t_2 to peer B, which should forward it to peer A and peer C. Packet 3 should be sent at time t_3 to peer C, which should forward it to peer B and peer D. Peer D should forward it further to peer A. Packet 4 should be sent at time t_4 to peer A, which should forward it to peer D. Peer D should forward it further to peer C, etc.

Note that the forwarding information should be included in the packet so that when a peer receives this packet it can forward it to the right peers. For example, when the source peer sends packet 4 to peer A, it includes in the packet the information $\to D(\to C)$. When peer A forwards this packet to peer D, it keeps only the information $\to C$.

Level-1 peers immediately forward the received packets to other level-1 peers as instructed by the source peer. In addition, a level-1 peer has its own scheduling strategy for level-2 peers. When it receives a packet from the source, it also tries to forward it to the level-2 peers that are directly connected to it according to its strategy.

When a level-1 peer completes the decoding of a source block, it sends an acknowledgment to all senders so that they stop sending it packets for this block. Then it applies rateless coding on the decoded block to feed level-2 peers. Thus, each receiving peer has two phases: forwarding (before the decoding is successful) and encoding (after decoding the source block). In the forwarding phase, the receiving peer just forwards the received packets to the next-level peers connected to it, while in the encoding phase, it generates encoded symbols from the decoded block and feeds the next-level peers.

Note that in the forwarding phase, a level-1 peer may not be able to comply with the transmission times of the scheduling strategy. Indeed, a level-1 peer may not have received a packet by its transmission time. However, in the encoding phase, the scheduling strategy can be obeyed fully.

A peer can receive the same packet from multiple senders. For example, peer F can receive the same packet from A and B in both the forwarding and encoding phase. To avoid this, a parent should know the other parents of its children, e.g., peer B should know that F is its common child with A. In this way, peer B will not forward a packet to F if this packet has previously visited peer A. This will avoid receiving duplicate packets from A and B while they are in the forwarding phase. In the encoding phase, receiving duplicate packets can be avoided with high probability by forcing peers to use different seed values for the rateless code.

The proposed scheme is resilient to packet loss, bandwidth fluctuation and peer churn. By having multiple senders, lost packets on one link can be compensated for by more packets on other links. Similarly, packet loss rate and bandwidth dynamics can be averaged out smoothly. If one of the senders churns out, it can be compensated by others before running the neighbor-discovery mechanism.

Players that are neighbours in the virtual world can be placed at the same level in the mesh, so that they can watch the video with approximately the same playback lag with respect to the original source.

Our system extends previous ideas proposed in [6] and [7]. However there are many important differences between these works and our scheme. For example, the system of [6], [7] do not have the notion of scheduling strategy and use a different approach to minimise the number of received duplicated packets.

One important requirement is that the live video stream should not consume the (upload and download) bandwidth that is necessary for the smooth operation of the MMOG. Usually, peers have much lower upload bandwidth than download bandwidth. Therefore, protecting the upload bandwidth for MMOG traffic is more crucial. One simple way to achieve this would be to reserve a portion of the upload bandwidth for the MMOG traffic. However, this would result in inefficient bandwidth utilization, especially when the MMOG traffic pattern is unknown and the upload bandwidth is scarce. We propose instead to use priority queues at the peer side. Each peer maintains two queues, one for the MMOG traffic and the other for the CNG traffic (including live video and any other user generated traffic). The scheduler ensures that (i) the MMOG traffic always gets higher priority over the CNG traffic by multiplexing both queues appropriately over the outgoing link (ii) the average transmission rate on the outgoing link does not exceed the upload bandwidth allocated by the ISP to avoid unnecessary queuing at the first router. Note that this technique does not need to know the pattern of MMOG traffic generation. Protecting enough download bandwidth for MMOG is equally important. However, since a peer can receive traffic from multiple senders, the technique of priority queues cannot be implemented. The only possibility is to reserve enough download bandwidth. It can be accomplished, e.g., by making each peer keep track of all its download activities actively so that there is enough download bandwidth available for the MMOG.

As UDP does not have a built-in congestion control mechanism, a pure UDP-based application may overwhelm the network. Therefore, we propose an application-layer congestion control mechanism for the P2P system. The idea is to adapt the UDP sending rate according to receiver feedback. For each block, the receiver sends a feedback to the sender reporting the average packet loss rate and the forward trip time (FTT). If the average FTT and loss rate are higher than some preset threshold values, this is a strong indication of congestion, and the sender decreases the sending rate accordingly.

Since many peers may become a source of a live stream, the P2P overlay consists of multiple P2P overlays rooted at distinct sources. A peer cannot obviously participate in all overlays because of its limited resources. To address this problem, we propose to continuously adjust the set of peers who are targeted to receive a stream from a source. The goal is to maximise the number of users receiving the stream from a given source, without causing network congestion and in a way such that peers near to the source get higher priority than distant ones. We use the concept of Area of Interest (AoI) [8] for that purpose. An AoI is defined as the part of the virtual world around a user that generates content. When a peer is within the AoI of a user generating high-quality content, or when it belongs to many AoIs, it may experience congestion. The challenge is to design a mechanism for determining the best size of these AoIs, that is, a size such that the maximum amount of UGC is delivered while no user experiences congestion. The management of the AoI must take into account the popularity of the virtual place and the capacity of the devices of the players that are located there. Such a management has been shown to be hard in wireless sensor networks [9], but some heuristics can perform well. For a player, the decision of increasing or decreasing the size of the AoI should be based on feedbacks from other nearby players in a collective manner.

Two strategies can be implemented. In the first one, one peer is congested, and not all peers in an AoI can be served. However, the capacity of peers in the surroundings of the congested peers makes that a new computation of AoI for all sources is not necessary. Instead, it is possible to “pass” one peer from one P2P overlay to another P2P overlay, so that the capacity provided by this peer can tackle the congestion issue. This strategy avoids heavy computations and can solve local small congestion problems. The second strategy can be implemented when the first one fails. A process similar to the one that ensures fair resource sharing in TCP can be used. Every source periodically tries to increase (in an additive manner) the size of its AoI until congestion is detected. Then, the radius of the AoI is decreased in a multiplicative manner (see [10] for a similar technique).

We expect that the combination of both strategies can guarantee that sources take profit from the capacity of peers to diffuse content as massively as possible, without provoking congestion in the system. Thus, a casual peer can automatically switch from one source to another one, depending on its distance to these sources and on the capacity of nearby peers.

In addition to designing a practical and efficient P2P live video streaming system for an MMOG environment, the CNG project proposes to contribute to a better understanding of the general problem of the diffusion of multiple video streams in a constrained environment. In most prior works, the system is assumed to be over-provisioned, that is, the resources are abundant, and so the content can be delivered on time. In the CNG context, the goal is to maximize the number of peers receiving content in an environment where not all peers can be served because not enough resources are available. If we assume tree overlays and consider only one video stream, the problem is to build a tree that spans the maximum number of peers with the constraint that every peer can only serve a limited number of other peers. In the context of many concurrent video streams, the problem becomes even harder with a constrained forest. The building of degree-constrained trees is an NP-hard problem [11]. We propose to contribute to the analysis of the computational properties of this problem. In particular, the formulation of the problem into several Integer Programming models and comprehensive benchmarks of these models will enable the computation of optimal solutions on small instances of the problem. Besides, we aim at designing heuristic algorithms, which allow the computation of nearly optimal solutions for large problem instances, as well as approximate algorithms (algorithms that compute solutions that are proved to be never far to the optimal solutions).

4 Web 2.0 collaboration tools

In addition to the IGIT technology and the P2P live video system, one of the major objectives of CNG is to offer a variety of Web 2.0 tools to the gamers. Web 2.0 tools are widely used for collaboration activities but, for most of the cases, the use of these tools involves closing or resizing the screen of the game and activating the tool’s window. This is the reason that some of the online collaboration tools have been already integrated in MMOGs as part of the game's screen [12].

CNG IGIT will create floating windows for browsing, chat and other Web 2.0 applications. These applications will enable users to interact with other players of the game, form communities and collaborate. Instead of having to connect to an online collaboration application or some other community tool outside of the game, the Web 2.0 tools offered will allow the users to collaborate and share information without ever leaving the game world. It is important that the tight integration of the collaboration tools within the game will be achieved without interfering with the development process of the game itself. Therefore, it can be easily applied to several different online games. The tools that are going to be offered to the users will allow them to interact directly. For example, when a user is playing an MMOG, she or he can instantly broadcast a personal message with a guide or a walkthrough using the available collaboration tool for instant messaging. The message will be sent to all community users as a personal message. This can be done by the player without any need to leave the game scene. The in-game experience will include mostly real-time communication tools like voice and video chat, messaging and file sending. The offered services, both synchronous and asynchronous, will create communication channels between the members of the game communities. The state of the art of the available frameworks for Web 2.0 applications will be used (e.g., Mia-Chat [13]). The tools integrated in the in-game environment will make use of current Asynchronous JavaScript and XML (AJAX) and Flash / Flex technologies. The CNG Server will be the entry point to all of these online features and will also be used for the storage of any necessary information over the users or the collaboration activities that should be permanently stored. The CNG Server design will be based on the state of the art of the principles for web design and architecture supporting modular growth and expansion of the offered features.

5 Conclusion

The CNG project will support and enhance MMOG community activities by enabling MMOG gamers to create, share, and insert UGC. CNG will provide an efficient, cost-effective way of enabling MMOGs to offer new community building tools without changing the game code and without adding new processing loads to the MMOG central servers. Such tools and functions may include instant messaging, in-game voting, reviewing and polling. This will reduce the need for visiting forums outside the game and diluting the MMOG experience. Moreover, MMOG gamers will be able to capture live video of their game and distribute it to other gamers using a P2P network.

CNG can provide huge benefits to MMOG developers and operators. New community building tools will be offered cost-effectively and efficiently, without the need to redesign or recode existing game offerings. The user experience will be enriched and the needs of the end-users will be better addressed, reducing churn to other MMOGs. Other benefits include lower network costs as well as new income streams through in-game and around game advertising. MMOG providers will maintain control over how various commercial and UGC content is displayed within the MMOG, thus keeping editorial control of the look and feel of the game.

Acknowledgment

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

Some of the materials used for this paper are taken from [14].

References

[1] MMOGCHART.COM website. [Online]. Available: http://mmogchart.com. Last accessed 30/09/2010.

[2] CNG website. [Online]. Available: http://www.cng-project.eu/. Last accessed 30/09/2010.

[3] Roma Victor website. [Online]. Available: www.roma-victor.com. Last accessed 30/09/2010.

[4] Xfire website. [Online]. Available: http://www.xfire.com. Last accessed 30/09/2010.

[5] S. Ahmad, R. Hamzaoui, and M. Al-Akaidi, "Adaptive unicast video streaming with rateless codes and feedback," IEEE Trans. Circuits Syst. Video Technol. vol. 20, pp. 275–285, Feb. 2010.

[6] C. Wu and B. Li, "rStream: resilient and optimal peer-to-peer streaming with rateless codes," IEEE Trans. Parallel Distrib. Syst. vol. 19, no. 1, pp. 77-92, Jan. 2008.

[7] M. Grangetto, R. Gaeta, and M. Sereno, "Reducing content distribution time in P2P-based multicast using rateless codes," in Proc. Italian Networking Workshop, Cortina, Jan. 2009.

[8] J.-R. Jiang, Y.-L. Huang, and S.-Y. Hu, "Scalable AOI-cast for peer-to-peer networked virtual environments," Journal of Internet Technology, vol. 10, no. 2, pp. 119–126, 2009.

[9] B. Han and J. Leblet, and G. Simon, "Query range problem in wireless sensor networks," IEEE Comm. Letters, vol. 13, no. 1, pp. 55–57, 2009.

[10] B. Han and G. Simon, "Fair capacity sharing among multiple sinks in wireless sensor networks," in Proc. IEEE MASS Conference, Pisa, Oct. 2007.

[11] J. Könemann and R. Ravi, "Primal-dual meets local search: approximating MST's with nonuniform degree bounds," in Proc. 35th Annual ACM Symposium on Theory of Computing, San Diego, 2003.

[12] E. Chan and P. Vorderer, "Massively Multiplayer Online Games," in Playing Video Games: Motives, Responses, and Consequences, P. Vorderer and J. Bryant (Eds.), Mahwah: Lawrence Erlbaum Associates, pp. 77–90, 2006.

[13] Mia-Chat website. [Online]. Available: http://code.google.com/p/mia-chat/. Last accessed 30/09/2010.

[14] S. Ahmad, C. Bouras, R. Hamzaoui, A. Papazois, E. Perelman, A. Shani, G. Simon, and G. Tsichritzis, "The Community Network Game project: enriching online gamers experience with user generated content," to appear in Proc. 2nd Int. Conf. Creative Content Technologies (CONTENT 2010), Lisbon, Nov. 2010.

Community Tools for Massively Multiplayer Online Games

Shakeel Ahmad*, Christos Bouras†, Raouf Hamzaoui*, Jiayi Liu§, Andreas Papazois†, Erez Perelman‡, Alex Shani ‡, Gwendal Simon §, George Tsichritzis †

*Department of Engineering
De Montfort University, Leicester, UK
sahmad@dmu.ac.uk, rhamzaoui@dmu.ac.uk

†Computer Technology Institute & Press "Diophantus"
N. Kazantzaki, GR26500 Patras, Greece
bouras@cti.gr, papazois@ceid.upatras.gr, tsixritzis@cti.gr

§Institut Telecom
Telecom Bretagne, France
jiayi.liu@telecom-bretagne.eu, gwendal.simon@telecom-bretagne.eu

‡Exent Technologies
Bazel 25, 49125 Petach-Tikva, Israel
eperelman@exent.com, 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. We present the EU funded Community Network Game (CNG) project. The CNG project provides tools to enhance collaborative activities between online gamers and offers new tools for the generation, distribution and insertion of user-generated content in MMOGs. CNG allows the addition of new engaging community services without changing the game code and without adding new processing or network loads to the MMOG central servers. The user-generated content considered by the CNG project includes 3D objects and graphics, as well as screen-captured live video of the game, which is shared using peer-to-peer technology. We survey the state of the art in all areas related to the project and present its concept, objectives, and innovations.**

***Keywords*-Massively Multiplayer Online Games; user generated content; P2P video streaming; graphics insertion.**

I. Introduction

Massively Multiplayer Online Games (MMOGs) allow a large number of online users (in some cases millions) to inhabit the same virtual world and interact with each other in a variety of collaborative and competing scenarios. MMOG gamers can build and become members of active communities with mutual interests, shared adventures, and common objectives. Players can play against other players (player versus player) or build groups (guilds) to compete against other groups (realm versus realm) or against computer-controlled enemies. MMOGs are rapidly gaining in popularity. Data from [1] suggests that there were over 16 million active subscriptions to MMOGs by 2008, a figure that is growing fast and predicted to rise to at least 30 million by 2012.

This paper presents the Community Network Game (CNG) project [2], an EU funded project within the Seventh Framework Programme. The project, which started in February 2010 and has a duration of 30 months, aims at enhancing community activities for MMOG gamers. This is achieved by providing Web collaboration tools and developing new tools for the generation, distribution and insertion of User-Generated Content (UGC) into existing MMOGs. This UGC may include textures and 3D objects to be added to the game, live video captured from the game screen and streamed to other players, as well as videos showing walk-throughs, game tutorials, or changes in the virtual world to be watched on demand.

The main technologies proposed by the CNG project are the In-game Graphical Insertion Technology (IGIT) and a Peer-to-Peer (P2P) system for the distribution of live video. IGIT is an innovative technology of replacing or inserting content into a game in real time without the need to change the game code in the client or server. For example, billboards can be inserted, tattoos can be added to in-game characters, an area on the screen can be assigned to display user information, and any type of window (browser, chat, etc.) can be inserted floating on or outside the game area. The technology can be implemented on multiple games, making it possible to create a community that is not limited to a specific game or publisher.

Enabling thousands of users to communicate UGC repre-

This paper appears in International Journal on Advances in Networks and Services, vol. 4, no 3&4, 2011.

sents a significant challenge to networks already dealing with the MMOG client-server data. The CNG project develops new techniques for UGC distribution that are "friendly" (supportive and not disruptive) to the MMOG client-server traffic. The key innovation is a P2P system that allows MMOG gamers to stream live video of the game without interrupting the MMOG data flow and without the need to upload the video data to a central server.

The remainder of the paper is organized as follows. Section II gives an overview of the state-of-the art in the areas of UGC, Web collaboration tools, P2P live video streaming, and game adaptation technologies. Section III presents the project's concept, objectives and technologies. Section IV concludes the paper by discussing the project's benefits and expected impact. The paper is an extension of the conference paper [3].

II. Related Work

In this section, we review the state-of-the art in the areas of UGC, Web collaboration tools, P2P live video streaming, and game adaptation technologies.

A. UGC

UGC includes various kinds of media content produced by end-users. In a game context, for example, this may be screen-captured video. Another example of UGC is the various mods created by the users. Sharing and remixing UGC is a widespread online activity that crosses borders of age and gender. Avid players go to great lengths in their efforts to create shared content in which they reveal their mastery. Additional data layers are always included: narration, animation and primarily soundtrack. Most MMOG-based UGC content is confined to dedicated game company sites as in World of Warcraft [4]. Many MMOG games also have their own community pages in social networking sites such as Facebook. In April 2010, Facebook released significant updates to its API by allowing external websites to uniformly represent objects in the graph (e.g., people, photos, events, and community pages) and the connections between them (e.g., friend relationships, shared content, and photo tags). As a result, the Facebook API [5] can provide an unprecedented bridge between gamespaces and the social web.

Current UGC tools can be classified into three categories: tools for capturing, for editing, and for uploading/broadcasting.

Capturing: Capturing videos can be done within the gamespace as in Spore [6]. This is not a common feature in MMOGs. More commonly, capturing is done with external video capture software such as Camtasia [7] or Fraps [8]. Fraps is the preferred software for users who want to capture high quality video. However, its free version has a 30 s recording limitation.

Editing: UGC sharing and remixing within game platforms is currently not supported. To edit the video and add effects, narration, soundtrack and text overlays, users tend to use readily available software such as Windows Movie Maker for Windows or iMovie for Mac that allow for the inclusion of additional content: audio, images and other videos. Annotations can only be added after the capture is done and cannot include other participants' comments.

Uploading/broadcasting: Once a user has captured and edited the video, a final step is needed to upload it for viewing. Many MMOG players use sites such as YouTube to share their game-based UGC. In 2008, Maxis incorporated YouTube APIs within their game, Spore, enabling players to upload videos of their creations to their YouTube account with only two clicks [9]. The collaboration between YouTube and a game creator (Electronic Arts), including revenue share from advertisements, is unique to date. Players of other games need to upload their video creation from their computer and cannot do it from within the game itself [10].

A user can capture the video of the game and broadcast it live to other users via a video server. This feature is offered by Xfire [11], which allows anyone to watch a live feed of a user's game screen. When a user begins a stream, a chat room is opened that anyone watching the live feed can join.

B. Web collaboration tools

Web 2.0 based collaboration applications may include instant messaging, audio and video chat, file sharing and online voting and polling. For audio/video capturing and playback the Flash software platform [12] is commonly deployed. Other solutions are the Java Applet technology or standalone applications which run on a Web browser and offer interoperability over different platforms. For instant messaging, online polling/voting and file sharing, Asynchronous JavaScript and XML (AJAX) [13] are commonly used. AJAX allows Web applications to retrieve data from the server asynchronously in the background without interfering with the display and behavior of the existing page. The use of AJAX techniques has led to an increase in interactive interfaces on webpages. Finally, for WWW client-server communication, most of the Web 2.0 applications are based on Simple Object Access Protocol (SOAP) [14]. SOAP relies on XML as its message format, and usually relies on other Application Layer protocols, most notably the Remote Procedure Call (RPC) and HTTP.

In the following, we give examples of Web 2.0 collaboration software.

1) Instant messaging: Instant messaging software is mainly based on AJAX technology. A typical AJAX chat application uses a database (MySQL) and AJAX to store and retrieve the users' messages and pass them between the client and the server. Examples of instant messaging software include AJAX Chat [15], Google Talk [16], ChatZilla [17], Mibbit [18], and Java/JavaScript Chat [19].

Table I: POPULAR CHAT TOOLS.

Tools	Instant Messaging	Audio and video chat	File sharing	Protocols
CGI:IRC	Perl/CGI	Not supported	Not supported	IRC
PJIRC	Java Applet	Not supported	Not supported	IRC
qwebirc	Ajax Applet	Not supported	Not supported	IRC
Parachat	Java Applet	Not supported	Not supported	Jabber/XMPP
Pichat	Ajax	Not supported	Not supported	Unknown
Facebookchat	Ajax	Not supported	Not supported	Jabber/XMPP
eBuddy	Ajax	Not supported	Not supported	Jabber/XMPP
Omegle	Ajax	Flash	Not supported	Jabber/XMPP
webcamnow	Ajax	Flash	Not supported	Jabber/XMPP
JatChat	Java Applet	Java Applet	Not supported	Jabber/XMPP
campfire	Ajax	Not supported	Ajax	Unknown
Single Operator Ajax chat	Ajax	Not supported	Ajax	Unknown

2) File sharing: Examples of popular Web2.0 file sharing systems include Meebo [20], iGoogle [21], Orkut [22], and FlashComs Community chat [23].

3) Audio and video chat: Audio and video chat applications are based on the Flash Platform. Some typical examples of Web-based audio and video chat tools are AVChat 3 [24], Red5Chat [25], MeBeam [26], Web Voice Chat [27], and 123 Live Help Chat Server Software [28]. Table I lists some widely used chat tools together with their underlying technology.

Table II: POPULAR VOTING AND POLLING APPLICATIONS.

Tools	Technology
Poll4Web	Flash
Flash Web Poll	Flash
ABPollMaster Polling	Java Applet
Fly06 Poll	Ajax

Table III: POPULAR BLOGWARES.

Tool	Technology
Kontain	Flash
Blogsmith	Ajax
TypePad	Ajax
Gawker bespoke software	Ajax

4) Online voting and polling: Examples of Web-based collaborative voting and polling tools are VotingPoll [29], DPolls [30], and XML Flash Voting Poll [31]. Table II lists other examples and the technology used for their implementation.

5) Blogging: Important tools used for the building of online blogging applications are WordPress [32], and Movable Type [33]. Table III lists popular blogwares.

C. P2P live video systems

Traditional client-server video streaming systems have critical issues of high cost and low scalability on the server. P2P networking has been shown to be cost effective and easy to deploy. The main idea of P2P is to encourage users (peers) to act as both clients and servers. A peer in a P2P system not only downloads data, but also uploads it to serve other peers. The upload bandwidth, computing power and storage space on the end user are exploited to reduce the burden on the servers.

Viewers of a live event wish to watch the video as soon as possible. That is, the time lag between the video source and end users is expected to be small. In a live streaming system, the live video content is diffused to all users in real time and video playback for all users is synchronized. Users that are watching the same live video can help each other to alleviate the load on the server. P2P live streaming systems allow viewers to delete the historic data after the playback, and hence have no requirement for any data storage and backup.

Based on the overlay network structure, the current approaches for P2P live streaming systems can be broadly classified into two categories: tree-based and mesh-based. In tree-based systems, peers form an overlay tree, and video data are pushed from the parent node to its children. However, a mesh-based system has no static streaming topology. Peers pull video data from each other for content delivery. Over the years, many tree-based systems have been proposed and evaluated, however, never took of commercially. Mesh-based P2P streaming systems achieve a large-scale deployment successfully, such as PPLive [34], PPStream [35], etc.

1) Tree-based systems: Many early P2P streaming systems use a tree-based approach that is typically based on application-level multicast architectures. Tree-based systems, such as ESM [36] and P2Cast [37], organize peers into a tree structure for delivering data. The data are diffused following this well-defined structure, typically pushed from a peer to its children. Tree-based solutions are perhaps the most natural and efficient approach, but they face several challenges. One major drawback of tree-based systems is the system fragility due to peer churn. A peer departure will disrupt data delivery to all its descendants, particularly for the peers in the higher level of the tree. The high dynamicity of peers in a P2P network potentially deteriorates transient performance. Another drawback is the under-utilized upload

Table IV: TRANSPORT PROTOCOLS IN P2P LIVE VIDEO STREAMING SYSTEMS.

System	Protocol
CoolStreaming [41]	TCP
PPStream [64]	TCP
PPLive [65]	Combination of TCP and UDP
TVAnts [64]	Combination of TCP and UDP
Joost [66]	UDP and TCP with UDP being the dominant traffic
SopCast [65]	Combination of TCP and UDP
[49]	UDP with FEC
[53]	UDP with ARQ
[60], [61], [67]	UDP
GridMedia [68]	UDP
iGridMedia [69]	UDP
PULSE [70]	UDP for control messages and TCP for data exchange
R2 [71]	UDP or TCP when UDP cannot be used due to firewall blocking

bandwidth of the peers. The leaf nodes in the tree cannot contribute any upload bandwidth resource to the system. Since a majority of nodes are leaves in the tree structure, this significantly reduces the overall efficiency. To address the issues of leaf nodes, multi-tree structures were introduced [38], [39]. In a multi-tree system, the source encodes the stream into several sub-streams and diffuses each sub-stream along one sub-tree. Each peer participates in many or all sub-trees to retrieve sub-streams. Hence, a peer might be deployed on an intermediate position in one sub-tree or a leaf position in another sub-tree. Compared with the single-tree approach, the multiple-tree solution has two advantages. First, the system's robustness is improved, as the failure of a high-level node would not completely disrupt all its descendants. Second, the upload bandwidth of all nodes could be well utilized, since each node stands a good chance to be both a leaf and an intermediate node. However, since the multiple-tree approach is still a tree-based solution, the drawbacks of tree-based systems remain basically unsolved. First, the construction and maintenance of the multiple-tree structure are costly because of frequent peer churn behaviours. Second, the upload bandwidth contribution of a node, which depends on the position in each sub-tree, is deficient. Furthermore, the design involves overhead.

Viewers in P2P live streaming systems only focus on the live video data that currently are output from the source. Hence, the video playbacks for all users are synchronized. In tree-based P2P live systems [36], all users participating in a video streaming session can form a tree at the application layer with the root at the video source. Each peer receives the live video data from its parent and immediately forwards the data to its children. Usually peers at lower levels receive the live data after peers at upper levels. The major consideration is to balance the depth of the tree and the out-degree of the intermediate nodes. Multi-Tree based approaches for P2P live systems are described in [38].

2) Mesh-based systems: In a mesh-based P2P streaming system, peer relationships are built and terminated according to data availability and bandwidth availability on peers. A video is typically divided into many chunks. Moreover, a tracker server maintains the relationship between peers and video data. Then, a peer can dynamically connect to a peer list that is chosen randomly from the tracker server according to which chunk the peer requests. After that, the peer maintains multiple neighbours and exchanges chunks with these neighbours simultaneously. A gossip protocol [40] is typically used for the topology management. Peers also periodically exchange information of the chunk availability using a buffer map. Usually, a chunk is pulled by a peer from its neighbours who have the requested chunk. The pull policy can avoid redundant chunk transmission.

If a neighbour leaves, the peer can continue retrieving chunks from other neighbours. The peer also explores some new neighbours to keep a certain number of neighbours. Due to the maintenance of multiple neighbours, mesh-based systems are highly robust against peer churns and fully utilize users upload bandwidth. However, transmission delay presents a challenge to mesh-based systems (for example, long start-up delay and channel switching problems for live streaming systems).

Many successful P2P live streaming systems [34], [35], [41] use the mesh-based streaming approach. The design of mesh-based P2P live streaming systems is relatively simple. All users are interested in the same live data. All chunks downloaded at a peer are always useful to other peers that have not retrieved these chunks. Some studies investigate the quality of peering connections. Several strategies are proposed to construct the peer relationship. The first consideration is the workload and resource availability on both peers, such as the current number of connections, upload and download bandwidth, and system resources. Other considerations are the network condition, which includes end-to-end delay and loss rate, and the network proximity, including geographical position, bandwidth, delay and network distance.

3) Server-assisted P2P systems: Most peer-to-peer systems rely on a server, either a bootstrapping server or a tracker server. A bootstrapping server is only used when a new peer joins an overlay. The bootstrapping server is expected to give to this newcomer a list of peers that are currently in the system. In this way, the new peer can quickly open connections. It has been shown, however, that a popular P2P streaming system like PPLive fails to provide accurate information to newcomers, resulting in a too long start-up delay [42]. Actually, a large proportion of peers that are given by the bootstrapping server do not answer the initial request of the newcomer, either because they are no longer in the system, or because they do not need any new connections. A tracker server extends the bootstrapping function. Every peer periodically sends a message to the tracker, which gives in return a list of peers (peerlist). That is, the participants to a peer-to-peer system can discover new peers on a periodic basis. The bit-torrent system has popularized this hybrid architecture, which guarantees, among other suitable properties, that new peers can quickly find matching peers.

In general, implementations of tracker-based peer-to-peer systems are simple. The tracker sends to a requesting peer a list of randomly chosen peers among the set of peers that are expectedly active in the system. Interestingly, the resulting topology is a random regular graph: every peer is connected to a given number of randomly chosen other peers. This random-like underlying topology is interesting on several aspects, especially random regular graphs are connected with high probability (so, any information is accessible from any peer), and the diameter of a random regular graph is small (therefore any information is close to any peer, if it is able to find it).

This topology links “acquaintances”. A peer can contact any subset of peers in the peerlist, but it is free to choose, among them, some privileged peers with which it will exchange data. This presents some problems. First, the peerlist contains peers exhibiting a broad scope of capacities, although the overlay tends to connect peers having similar characteristics [43]. The overlay would converge faster if the peerlist could contain preferentially the peers having the closest characteristics to the requester. However, it would require to authorize the tracker to determine as accurately as possible the capacity of peers, which appears to be impossible or costly in many cases. Second, the peerlist topology does not take into account the location of peers. Therefore the overlay wastes network resources.

4) Hybrid CDN-P2P systems: Peer-assisted (PAS) Content Delivery Networks (CDNs) have attracted a lot of attention in recent years. In this section, we present the architecture of a real-world CDN-P2P live video streaming system called LiveSky [44], which has been deployed in China. The system is designed to solve a set of problems in current CDN and P2P live video streaming systems such as scaling, fast startup and upload fairness.

Server side organization: The CDN overlay is constructed according to a tree-based structure, where leaves are edge servers, whose role is to serve end users. All other intermediate nodes are core servers, which are responsible for delivering content to edge servers. Because of their work load, edge servers are not allowed to transfer content between each other. To realize a P2P organization at the client side, an edge server has several roles: 1) a regular server for legacy clients; 2) a tracker for the P2P operation; 3) a seed in the P2P system.

Client side organization: There are two types of clients: legacy clients and P2P clients. Legacy clients are served in the traditional CDN manner and receive low quality streams. P2P clients are organized in a hybrid scheme proposed in [45], [46] that combines the multi-tree and mesh topologies. As usual a video is divided into several substreams. Each substream contains nonconsecutive frames. The peers that host the same substream compose a tree-based overlay. This ensures upload fairness of each peer. On the other hand, peers also use a mesh-style pull mechanism to retrieve missing frames for continuous playback. This enhances the robustness of the network. Moreover, P2P clients are allowed to access high quality videos.

Adaptive scaling and improvements: In the system, each edge server decides whether a new arrival client should be treated as a legacy client or a P2P client independently. A threshold is pre-configured in every edge server. When the number of clients is below the threshold, all clients retrieve the content directly from the edge server. If the number of clients exceeds the threshold, new arrival clients will be redirected to other clients to form a P2P organization. Both the threshold and the capacity of an edge server are calculated by some parameters, including the level of the P2P tree overlay, peer arrival rate, peer leaving rate and peer contribution rate. When an edge server reaches its capacity limitation, new clients will be redirected to a less loaded edge server.

Fast startup: Startup time is optimized in LiveSky in two ways. First, the buffer size is reduced to 15 seconds. Second, the first request of a client is always replied directly by an edge server, thus it is very quick to retrieve startup streams.

5) Video transmission protocols: In private, well-managed IP networks, the quality of service (QoS) is maintained by calibrating the end-to-end infrastructure. This is not possible in P2P overlays since they are built on open IP networks, which are best-effort in nature. Real-time video communication over P2P overlays on the public Internet mainly relies on the transmission control protocol (TCP). TCP guarantees reliable transmission of the data by automatic retransmission of lost packets. However, as TCP requires in order delivery of the data and keeps on re-transmitting a packet until an acknowledgement is received, significant delays may be introduced. Further delays are caused by the congestion control algorithm used by TCP, which reacts to packet loss by reducing the transmission rate, leading occasionally to service interruption. This presents a serious drawback for real-time video communication where the data must be available to the receiver at its playback time. Lost and delayed packets that miss their playback deadline not only are useless, they also consume the available bandwidth unnecessarily.

An alternative to TCP is to use UDP as the transport protocol and apply application-layer error control. For example, the Darwin Streaming Server, which is the open-source version of Apple's QuickTime Streaming Server, uses a simple timeout-based ARQ scheme [47]. The Helix DNA streaming system, which is the open-source version of RealNetworks Helix streaming suite, also uses timeout based ARQ [47]. Windows Media uses a selective retransmission scheme. If the client detects gaps in the packet sequence numbers, it sends a retransmission request to the server, which retransmits the missing packets. Packet retransmissions are limited to a certain percentage of the available bandwidth and packets to be retransmitted are prioritized according to their content. Audio packets are given the highest priority. Video packets close to their playout deadlines are given the lowest priority, on the presumption that retransmissions are most likely to miss the playout deadline [48]. VideoLAN, a popular open-source streaming system, uses either TCP or UDP without packet loss mechanisms [47]. These streaming techniques are suitable for well-managed networks. However, they face considerable problems in open IP networks where the packet loss rate may be significant, and the available bandwidth may be variable. In P2P overlays, in particular, packet loss is not only due to congestion at routers but also to the heterogeneity in node stay-time duration.

Most P2P streaming systems use TCP (CoolStreaming, PPStream), a combination of UDP without error control and TCP (PPLive, TVAnts), UDP with Forward Error Correction (FEC) [38], [49], [50], [51], [52], and ARQ [53]. Another approach is Multiple Description Coding (MDC) [54], [55]. However, MDC schemes are rarely used in practice because they rely on non-standard video coders.

Thomos and Frossard [56] use network coding with rateless codes for P2P video streaming. The technique exploits path diversity and lessens the burden of re-encoding on an intermediate forwarding peer.

Wu and Li [57] also use network coding based on rateless codes for P2P live video streaming. A peer can recover the original video source block by receiving enough encoded symbols from multiple receivers. As soon as a receiving peer successfully decodes the source block, it becomes a source and applies rateless coding on the decoded source block to generate encoded symbols for other peers. To avoid receiving redundant symbols, each peer uses a different seed for rateless encoding. The authors propose a distributed algorithm for best peer selection and optimize rate allocation to guarantee minimum delay.

Grangetto, Gaeta, and Sereno [58] propose an improvement to the method of Wu and Li [57]. In their method, called 'Relay and Encode' (RE), a receiving peer relays the received encoded symbols immediately. Once it decodes the source block, a rateless code is applied on the source block and newly produced encoded symbols are sent to its children. The authors show that RE has a lower delay than the method of [57]. However, the paper does not consider the effects of varying channel conditions and does not exploit feedback to minimize bandwidth usage. Moreover, the protocol is not robust against failures. For example, if one peer cannot decode the source block, all its descendent peers are affected. A similar system is proposed by the same authors in [59]. Here, receiver feedback is used to ask the sender to stop sending more symbols when the source block is decoded.

In [52], rateless coding is used to make a P2P VoD system resilient to peer churn. The source partitions the video into source blocks and applies rateless coding on these blocks. During a push phase, for each source block, distinct groups of encoded symbols are distributed among a number of volunteering peers, which may not be interested to watch the video. This pushing can be done during low network utilization time. In a pull phase, a peer who wants to watch the video needs to collect a minimum number of distinct encoded symbols from any subset of volunteering peers.

Setton, Noh, and Girod [60] propose a system for live video streaming over P2P networks aimed at low latencies and congestion avoidance. Video packets are sent using UDP/IP, and a scheduling algorithm is used to maximize the received video quality, while minimizing network congestion.

In [61], a P2P live video streaming system aimed at low delays is presented. The system uses the Stanford Peer to Peer Multicast Protocol to build multiple complementary multicast trees, all originating at the source. The source exploits path diversity and sends different packets over different multicast trees. The video is compressed with the Scalable Video Coding (SVC) and packets are sent over UDP/IP. Depending on the available bandwidth and packet loss rates, intermediate nodes decide how many layers should be sent to their children. A simple ARQ mechanism is used to deal with packet loss.

One limitation of the UDP protocol is the lack of a congestion control mechanism. Congestion control with UDP can be realized with the Datagram Congestion Control Protocol (DCCP) [62]. DCCP uses an Explicit Congestion Notification (ECN) bit, which is set on by a congested router. When a receiver receives a packet with an ECN bit set on, it asks the sender to react to the congestion accordingly. However, most routers disable ECN [63].

Table IV gives an overview of the transport protocols used in P2P systems.

D. *Game adaptation technologies*

In-game technologies have been used in the gaming market for several years. The gaming industry has adopted these

Table V: IN-GAME TOOLS. AN X INDICATES THAT THE TOOL IS OFFERED BY THE PRODUCT.

Product	XFIRE	PLAYXPERT	Massive	FreeRideGames	Double Fusion	Steam	Overwolf	Raptr
Texture replacement			x		x			
Game resize				x				
In-game overlay	x	x		x		x	x	x
Video capture	x	x					x	
Video edit								
Video upload	x						x	
Live video	x							
Instant messaging	x	x				x	x	x
Audio chat	x					x		x
File sharing								
Online blogging								
Need for SDK	No	No	Yes	No	Yes	Yes	No	No

technologies to increase its revenue by finding more financial sources and by attracting more users. In-game overlay allows to view and interact with windows outside the game, but without "Alt-Tabbing". It does so by rendering the window inside the game. Texture replacement enables to replace an original game texture with a different texture. In this way, the newly placed textures are seen as part of the original game content. This method is commonly used for dynamic in-game advertisement. Game size modification technology adapts the original game by decreasing its original size and surrounding it with an external content.

Some of these technologies are distributed as an external utility that can overlay a pack of games while others are part of MMOG service features which are provided to the users.

The main available in-game adaptation products on the market are Xfire [11], PLAYXPERT [72], FreeRidesGames [73], Massive [74], Double Fusion [75], Steam [76], Overwolf [77], and Raptr [78]. Some of the products require for the game developer to integrate the product's Software Development Kit (SDK) (for each game to be developed the game developer must use the products SDK). The use of those in-game adaptation products is not available for the existing games catalogue. Table V lists the in-game tools surveyed in this paper, showing those offered by existing products.

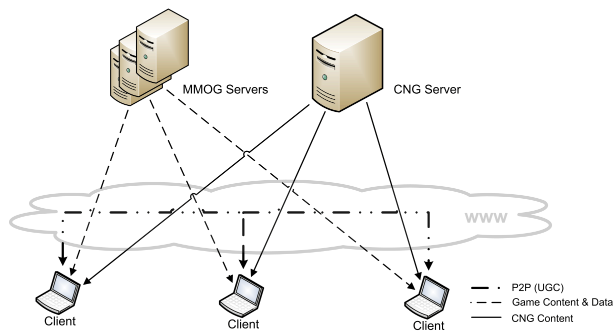


Figure 1. CNG architecture.

III. TECHNOLOGIES AND INNOVATIONS

Fig. 1 shows the CNG architecture. While the MMOG architecture is not modified (the game content and the game data are still transferred through the MMOG servers), the following components are added: (i) Sandbox on the client side that is responsible for modifying the game environment; (ii) CNG Server for monitoring the P2P UGC communication. The CNG server acts as a tracker for the system in the sense that it is in charge of introducing peers to other peers. It has persistent communication with the clients and manages the organization of the P2P exchanges.

A. IGIT

IGIT enables the user to resize the game and surround it with external content, overlay the game, and replace an existing game texture with an external content. This is done in a way that does not harm the game experience and without the need for SDK integration. Fig. 2 and Fig. 3 illustrate some of these features. Fig. 2 is a screenshot from the MMOG game "Roma Victor" [79] by RedBedlam. Fig. 3 shows the same game scene with a mock-up of CNG features. The modifications, which are numbered in Fig. refigit, are as follows:

- (1) The original resolution of the game was modified to enable an additional frame around the game to hold the in-frame objects. IGIT uses the GPU of the user's machine for changing the resolution of the game to avoid reduction in the image quality;
- (2) Instant messaging window as an example of active Web 2.0 application;
- (3) Web browser that presents online passive information (in this example, a leader board);
- (4) Another Web browser window that presents an updated advertisement;
- (5) MMOG specific chat to enable the users in a specific scene to cooperate;
- (6) In-game 3D UGC. In this example, a user added a note on a tree to publish an eBay auction;
- (7) Two windows of a video chat with casual friends or cooperative players.

The choice of application and the application's screen location are under the control of the user (player). The Web 2.0 applications are browser-based applications that are downloaded from the CNG Server and run in the web-browser instances within the CNG Client. The purpose of Web 2.0 Applications is to offer online collaboration services to the user. They are browser-based and downloaded from the CNG Server and run in the web-browser instances within the CNG Client.

Since they are web-based the CNG client retrieves all the necessary information from the CNG Server

In addition, the CNG toolbox includes video recording and editing tools that allow users to capture the video of the game and

1) trim the captured game video
2) split, duplicate and sequence the recorded video cuts
3) remix trimmed cuts of the recorded video
4) upload the edited video to YouTube

Figure 2. Original MMOG screenshot.

B. P2P live video system

In existing MMOGs, a player can capture the video of the game and send it to a central server which broadcasts it live to other users [11]. However, this solution, which heavily relies on central servers, has many drawbacks such as high costs for bandwidth, storage, and maintenance. Moreover, this solution is not easily scalable to increasing number of users. The CNG project proposes a P2P live video streaming system to address the limitations of server-based solutions. The CNG P2P live video system allows every peer to become a source of a user-generated video stream for a potentially large set of receivers. While many P2P live video systems have been proposed, none of them has been specifically designed for the unique environment of MMOGs. In particular, none of the existing P2P live video systems addresses the following challenges:

Figure 3. IGIT-modified MMOG screenshot.

- Any MMOG player should be able to multicast live video. The video can potentially be received by any other player in the P2P network.
- 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).
- Live video should be delivered at about the same time to all peers at the same "level". For example, a level can be a priority class in a multi-tiered premium service. Peers in a higher priority class should be able to watch the video before those in a lower priority class. Alternatively, a level can be defined as the set of MMOG players that are in the same region of the virtual world.

The CNG P2P live video system is designed as follows. Peers are organized in different levels (Fig. 4). The source peer is placed at level 0. All peers directly connected to the source peer are at level 1. In general, a peer is considered to be at level j if the shortest path from the source peer to it consists of j intermediate links.

The video is captured in real time from the source screen, compressed, and partitioned into source blocks. Each block consists of one GOP (Group of Pictures) and is an independent unit of fixed playback duration (e.g., 1 s). The UDP protocol is used as the transport protocol. The source peer applies rateless coding on each source block and sends the resulting encoded symbols in successive packets to level-1 peers. Packets are sent according to a scheduling strategy. The strategy specifies the maximum number n of encoded packets that can be sent by the source for this block, the time t_i at which packet i is sent, and a hierarchical forwarding scheme $F_i,\ i = 1, 2, \ldots, n$. An example of a scheduling strategy for $n = 4$ and the four level-1 peers of Fig. 4 is as follows.

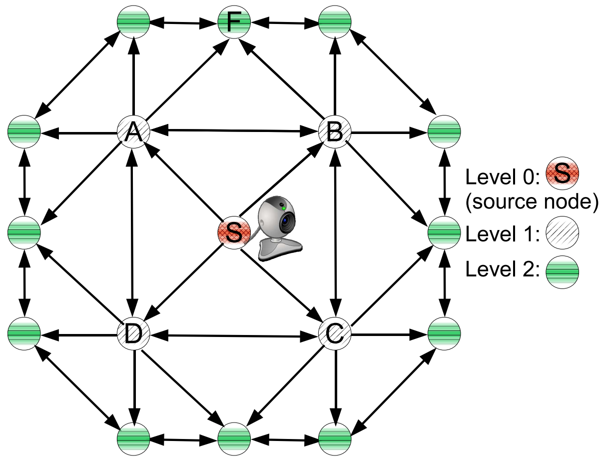


Figure 4. P2P topology.

- $1 : t_1 : A \rightarrow B + D(\rightarrow C)$,
- $2 : t_2 : B \rightarrow A + C(\rightarrow D)$,
- 3: $t_3 : C \rightarrow B(\rightarrow A) + D$,
- 4: $t_4 : D \rightarrow C(\rightarrow B) + A$

The strategy says: packet 1 should be transmitted at time t_1 to A. A forwards the packet to B and D. D forwards it to C. Packet 2 should be transmitted at time t_2 to B. B forwards it to A and C. C forwards it to D. Packet 3 should be transmitted at time t_3 to C. C forwards it to B and D. B forwards it to A. Packet 4 should be transmitted at time t_4 to D. D forwards it to C and A. C forwards it to B.

Level-1 peers forward packets that are directly received from the source to their adjacent level-2 peers. When a level-1 peer completes the decoding of a source block, it sends an acknowledgment to the source peer so that it stops sending it packets. Then it applies rateless coding on the decoded source block and starts acting as a source for those level-2 peers that are its direct successors. The same procedure applies to peers at the next levels.

Thus, each receiving peer has two phases: forwarding (before the decoding is successful) and encoding (after decoding the block). In the encoding phase, receiving duplicate packets can be avoided with high probability by forcing peers to use different seed values for the rateless code.

One of the main challenges consists of optimizing the scheduling strategy.

Our system extends previous ideas proposed in [57], [58]. However there are many important differences between these works and our scheme. For example, the systems of [57], [58] do not have the notion of scheduling strategy. Also in [57], [58], there is no notion of levels within the mesh.

As UDP does not have a built-in congestion control mechanism, a pure UDP-based application may overwhelm the network, leading to packet loss and degraded video quality. Therefore, the CNG project proposes an application-layer congestion control mechanism for the P2P system. The source peer adapts the video bit rate and UDP sending rate according to feedback received periodically from all peers. This feedback consists of the outage rate, i.e., the percentage of source blocks that were not decoded in time.

IV. Conclusion

We presented the EU funded CNG project. CNG supports and enhances community activities between MMOG gamers by enabling them to create, share, and insert UGC. The UGC considered by the CNG project includes 3D objects, graphics, and video. CNG develops in-game community activities using an in-game graphical insertion technology that replaces or inserts content in real time without the need to change the game's code in the client or server. CNG uses an architecture that efficiently combines the client-server infrastructure for the MMOG activities with a P2P overlay for the delivery of live video. The video traffic represents a real challenge to the network already occupied by the MMOG client-server data. The project proposes new techniques for P2P live video streaming that are "friendly to the MMOG client-server traffic. Since video can be resource heavy, the network indirectly benefits from the increased locality of communication. CNG also provides Web 2.0 tools for audio and video chat, instant messaging, in-game voting, reviewing, and polling. This will reduce the need for visiting forums outside the game and diluting the MMOG experience.

CNG has the potential to provide huge benefits to MMOG developers and operators. New community building tools will be offered cost-effectively and efficiently, without the need to redesign or recode the existing game offerings. The user experience will be enriched, and the needs of the end-users will be better addressed. The community will be brought into the content, and the game communities will become more engaged, reducing churn to other MMOGs. New income streams will be delivered with the help of in-game and around game advertising. Yet, MMOG developers and operators will be able to maintain control over how various commercial and UGC content is displayed, thus keeping editorial control of the look and feel of their MMOG.

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 at: http://www.mmogchart.com. Last accessed: 8/8/2011.

[2] [Online]. Available at: http://www.cng-project.eu/. Last accessed: 7/2/2012.

[3] S. Ahmad, C. Bouras, R. Hamzaoui, A. Papazois, E. Perelman, A. Shani, G. Simon, and G. Tsichritzis, “The Community Network Game project: Enriching online gamers experience with user generated content,” in Proc. 2nd Int. Conf. Creative Content Technologies (CONTENT 2010), Lisbon, Nov. 2010.

[4] [Online]. Available at:: http://eu.battle.net/wow/en/. Last accessed: 7/2/2012.

[5] Facebook API. [Online]. Available at: http://developers.facebook.com/docs/. Last accessed: 7/2/2012.

[6] [Online]. Available at: http://eu.spore.com/home.cfm?lang=en. Last accessed: 7/2/2012.

[7] [Online]. Available at: http://www.techsmith.com/camtasia/. Last accessed: 7/2/2012.

[8] [Online]. Available at: http://www.fraps.com. Last accessed: 7/2/2012.

[9] Youtube API. [Online]. Available at: http://code.google.com/apis/youtube/casestudies/ea.html. Last accessed: 7/2/2012.

[10] D. Takahashi, “YouTube game videos become a big channel for game marketers”, 2008. [Online]. Available at: http://games.venturebeat.com/2008/12/18/youtube-game-videos-become-a-big-channel-for-game-marketers/. Last accessed: 7/2/2012.

[11] [Online]. Available at: http://xfire.com. Last accessed 7/2/2012.

[12] Adobe Flash. [Online]. Available at: http://www.adobe.com/support/flash/downloads.html. Last accessed: 7/2/2012.

[13] XMLHttpRequest, W3C Working Draft, 2009. [Online]. Available at: http://www.w3.org/TR/2009/WD-XMLHttpRequest-20091119/. Last accessed: 7/2/2012.

[14] SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), W3C Recommendation, 2007. [Online]. Available at: http://www.w3.org/TR/soap12-part1. Last accessed: 7/2/2012.

[15] [Online] Available at: http://blueimp.net/ajax. Last accessed: 7/2/2012.

[16] [Online] Available at: http://www.gmail.com. Last accessed: 7/2/2012.

[17] [Online] Available at: https://addons.mozilla.org/en-US/firefox/addon/chatzilla/. Last accessed: 7/2/2012.

[18] [Online] Available at: http://www.mibbit.com. Last accessed: 7/2/2012.

[19] [Online] Available at: http://www.php-development.ru/software/chat-java.php. Last accessed: 7/2/2012.

[20] [Online] Available at: http://www.meebo.com/. Last accessed: 7/2/2012.

[21] [Online] Available at: http://www.google.com/ig. Last accessed: 7/2/2012.

[22] [Online] Available at: http://www.orkut.com/. Last accessed: 7/2/2012.

[23] [Online] Available at: http://www.flashcoms.com/products/community_video_chat/. Last accessed: 7/2/2012.

[24] [Online] Available at: http://avchat.net/. Last accessed: 7/2/2012.

[25] [Online] Available at: http://www.red5chat.com/. Last accessed: 7/2/2012.

[26] [Online] Available at: http://www.mebeam.com/. Last accessed: 7/2/2012.

[27] [Online] Available at: http://www.fileguru.com/Web-Voice-Chat/info. Last accessed: 7/2/2012.

[28] [Online] Available at: http://www.123flashchat.com. Last accessed: 7/2/2012.

[29] [Online] Available at: http://acidjs.wemakesites.net/voting-poll.html. Last accessed: 7/2/2012.

[30] [Online] Available at: http://ajaxian.com/archives/dpolls-an-ajax-pollster. Last accessed: 7/2/2012.

[31] [Online] Available at: http://www.flabell.com/flash/XML-Flash-Voting-Poll-39. Last accessed: 7/2/2012.

[32] [Online] Available at: http://wordpress.org/. Last accessed: 7/2/2012.

[33] [Online] Available at: http://www.movabletype.org/. Last accessed: 7/2/2012.

[34] [Online]. Available at: http://www.pptv.com/. Last accessed: 7/2/2012.

[35] [Online]. Available at: http://www.PPstream.com. Last accessed: 7/2/2012.

[36] Y.-H. C. Sanjay, S. G. Rao, S. Seshan, and H. Zhang, “A case for end system multicast,” in Proc. ACM Sigmetrics, pp. 1–12, 2002.

[37] G. Yang, S. Kyoungwon, K. Jim, and T. Don, “P2cast: peer-to-peer patching scheme for VoD service,” in Proc. 12th Int. Conf. World Wide Web, pp. 301–309, 2003.

[38] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, “SplitStream: High-bandwidth multicast in a cooperative environment,” in Proc. ACM SOSP, 2003.

[39] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai, “Distributing streaming media content using cooperative networking,” in Proc. NOSSDAV ’02, pp. 177–186, 2002.

[40] M. Jelasity, R. Guerraoui, A.-M. Kermarrec, and M. V. Steen, “The peer sampling service: experimental evaluation of unstructured gossip-based implementations,” in Proc. ACM/IFIP/USENIX Int. Conf on Middleware, pp. 79–98, 2004.

[41] B. Li, G. Y. Keung, C. Lin, J. Liu, and X. Zhang, “Inside the new coolstreaming: Principles, measurements and performance implications,” in Proc. INFOCOM’08, 2008.

[42] A.-M. Kermarrec, E. Le Merrer, Y. Liu, and G. Simon, “Surfing peer-to-peer IPTV system: distributed channel switching,” in Proc. EuroPar, 2009.

[43] A.-T. Gai, F. Mathieu, F. de Montgolfier, and J. Reynier, “Stratification in P2P networks: application to BitTorrent,” in Proc. 27th IEEE International Conference on Distributed Computing Systems (ICDCS), 2007.

[44] H. Yin, X. Liu, T. Zhan, V. Sekar, F. Qiu, C. Lin, H. Zhang, and B. Li, “Design and deployment of a hybrid CDN-P2P system for live video streaming: Experiences with livesky,” in Proc. ACM Multimedia, 2009.

[45] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. G. Liu, and A. Silberschatz, “P4p: Provider portal for applications,” ACM SIGCOMM Computer Communication Review, Vol. 38, No. 4, Oct. 2008.

[46] M. Zhang, J.-G. Luo, L. Zhao, and S.-Q. Yang, “A peer-to-peer network for live media streaming using a push-pull approach,” in Proc. 13th ACM International Conference on Multimedia, 2005.

[47] M. Röder, *Efficient Rate-Distortion Optimized Media Stream-*

ing, PhD Thesis, University of Konstanz, 2007.

[48] P.A. Chou, “Streaming media on demand,” in M. van der Schaar, P.A. Chou (editors), Multimedia over IP and Wireless Networks: Compression, Networking, and Systems, Academic Press, 2007.

[49] V. N. Padmanabhan, H. J. Wang, and P. A. Chou, “Resilient peer-to-peer streaming,” in Proc. IEEE ICNP, pp. 16–27, Atlanta, GA, Nov. 2003.

[50] M. Hefeeda, A. Habib, B. Botev, D. Xu, and B. Bhargava, “PROMISE: peer-to-peer media streaming using CollectCast,” in Proc. ACM Multimedia (MM’03), Berkeley, CA, Nov. 2003.

[51] X. Xu, Y. Wang, S.§. Panwar, and K. W. Ross, “A peer-to-peer video-on-demand system using multiple description coding and server diversity,” in Proc. IEEE Int. Conf. Image Processing, 2004.

[52] K. Suh, C. Diot, J. Kurosey, L. Massoulie, C. Neumann, D. Towsley, and M. Varvello, “Push-to-Peer video-on-demand system: design and evaluation,” IEEE Journal on Selected Areas in Communications, vol. 25, no. 9, 2007.

[53] E. Setton, J. Noh, and B. Girod, “Rate-distortion optimized video peer-to-peer multicast streaming,” in Proc. P2PMMS’05, pp. 39–48, Singapore, Nov. 2005.

[54] E. Akyol, A. M. Tekalp, and M. R. Civanlar, “A flexible multiple description coding framework for adaptive peer-to-peer video streaming,” IEEE Journal of Selected Topics in Signal Processing, vol. 1, no. 2, pp. 231–245, Aug. 2007.

[55] M.-T. Lu, J.-C. Wu, K.-J. Peng, P. Huang, J. J. Yao, and H. H. Chen, “Design and evaluation of a P2P IPTV system for heterogeneous networks,” IEEE Transactions on Multimedia, vol. 9, pp. 1568–1579, Dec. 2007.

[56] N. Thomos and P. Frossard, “Raptor network video coding,” in Proc. 1st ACM International Workshop on Mobile video (in conjunction with ACM Multimedia 2007), Augsburg, Germany, Sep. 2007.

[57] C. Wu and B. Li, “rStream: resilient and optimal peer-to-peer streaming with rateless codes,” IEEE Transactions on Parallel and Distributed Systems, vol. 19, pp. 77–92, Jan. 2008.

[58] M. Grangetto, R. Gaeta, and M. Sereno, “Rateless codes network coding for simple and efficient P2P video streaming,” in Proc. IEEE ICME 2009, Cancun, Mexico, 2009.

[59] M. Grangetto, R. Gaeta, and M. Sereno, “Reducing content distribution time in P2P based multicast using rateless codes,” in Proc. Italian Networking Workshop, pp. 1–12, Cortina, 2009.

[60] E. Setton, J. Noh, and B. Girod, “Congestion-distortion optimized peer-to-peer video streaming,” in Proc. IEEE ICIP-2006, Atlanta, GA, Oct. 2006.

[61] P. Baccichet, T. Schierl, T. Wiegand, and B. Girod, “Low-delay peer-to-peer streaming using scalable video coding,” in Proc. International Packet Video Workshop, PV2007, Lausanne, Switzerland, Nov. 2007.

[62] [Online]. Available at: http://tools.ietf.org/html/rfc4340. Last accessed 7/2/2012.

[63] R. Diana and E. Lochin, “ECN verbose mode: A statistical method for network path congestion estimation,” in Proc. INFOCOM, San Diego, CA, 2010.

[64] T. Silverston and O. Fourmaux, “Measuring P2P IPTV systems,” in Proc. ACM NOSSDAV’07, June 2007.

[65] S. Ali, A. Mathur, and H. Zhang, “Measurement of commercial peer-to-peer live video streaming,” in Proc. ICST Workshop Recent Adv. Peer-To-Peer Streaming, Waterloo, Aug. 2006.

[66] M. Alhaisoni and A. Liotta, “Characterization of signalling and traffic in Joost,” Journal of Peer-to-Peer Networking and Applications, vol. 2, issue 1, pp. 75–83, 2009.

[67] P. Baccichet, J. Noh, E. Setton, and B. Girod, “Content-aware P2P video streaming with low latency,” Proc. IEEE International Conference on Multimedia and Expo, ICME 2007, Beijing, China, Jul. 2007.

[68] M. Zhang, Q. Zhang, L. Sun, and S. Yang, “Understanding the power of pull-based streaming protocol: Can we do better?,” IEEE Journal on Selected Areas in Communications, Vol. 25, pp. 1678–1694, Dec. 2007.

[69] M. Zhang, L. Sun, and S. Yang, “iGridMedia: Providing delay-guaranteed peer-to-peer streaming service on Internet,” in Proc. IEEE GLOBECOM 2008, New Orleans, LO, Dec. 2008.

[70] F. Pianese, D. Perino, J. Keller, and E.W. Biersack, “PULSE: An adaptive, incentive-based, unstructured P2P Live streaming system,” IEEE Transactions on Multimedia, Volume 9, Number 8, pp. 1645–1660, Dec. 2007.

[71] M. Wang and B. Li, “R2: Random push with random network coding in live peer-to-peer streaming,” IEEE Journal on Selected Areas in Communications, Vol. 25, Number 9, pp. 1655–1666, Dec. 2007.

[72] [Online]. Available at: http://www.playxpert.com/. Last accessed 7/2/2012.

[73] [Online]. Available at: http://www.freeridegames.com/. Last accessed 7/2/2012.

[74] [Online]. Available at: http://en.wikipedia.org/wiki/Massive_Incorporated. Last accessed 7/2/2012.

[75] [Online]. Available at: http://www.doublefusion.com/. Last accessed 7/2/2012.

[76] [Online]. Available at: http://store.steampowered.com/. Last accessed 7/2/2012.

[77] [Online]. Available at: http://www.overwolf.com/. Last accessed 7/2/2012.

[78] [Online]. Available at: http://raptr.com/. Last accessed 7/2/2012.

[79] [Online]. Available at: http://www.roma-victor.com. Last accessed 7/2/2012.

Users and Uses of Multiplayer Games and Community Activities

Eva Ferrari[1], Jane Lessiter[1], Jonathan Freeman[1]

[1]i2 media research limited, Goldsmiths University of London, London, UK

E-mail: [1]{e.ferrari|j.lessiter|j.freeman}@gold.ac.uk

Abstract: **This paper outlines user needs and initial stakeholder research conducted as part of the Community Network Game (CNG) project. CNG aims to apply new network technologies to support community activities, particularly based on user generated content (UGC) within (massively) multiplayer online games. Seventeen interviews were conducted with multiplayer gamers to identify why players communicate with others, what they share and how, and what they find frustrating about using current community related tools. Analysis of the interviews indicated different drivers and hurdles to communicating with other gamers. Whilst interviewees used a wide range of existing community related tools, they expressed various frustrations about their use of them including reduced immersion through over-use of UGC, communication difficulties and privacy issues, poor usability of tools, and poor quality control of UGC. In the second phase, an online survey (n=414) aimed to quantify these frustrations and interest in the functionalities to be offered by CNG. The results gave support to the development of a broad range of in-game functions, available without the need to minimise the game screen. Moreover, an initial gamer segmentation indicated that the key CNG functions scheduled for development in the project such as video sharing, chat services and in-game web-browsing, were particularly popular with dedicated online multiplayer gamers. Initial stakeholder research was broadly consistent with gamers' feedback and considered that the in-game live video streaming tool would be of most interest to game developers.**

Keywords: User Generated Content, User needs, Massively Multiplayer Online Games.

1 INTRODUCTION

Massively multiplayer online games (MMOGs) represent a particular genre of the games market. Players log into and out of these persistent worlds, usually maintaining a character that develops abilities and skills and which is typically part of a long-term social group of other players [1]. MMOGs are defined by having a large number of players which can, from across the globe, cooperate, compete or simply co-exist in a persistent environment. Players can develop relationships with others whether or not they know them in the real world.

Research has found that MMOGs have wide demographic appeal [2]. MMOG players spend an average of 10.8 hours a week on any activity pertaining to the game which does not involve playing the game itself (on which they spend around 22 hours) [3]. Players engage in activities such as: looking for game-related information (3.5 hours on average), reading reviews or comments on forums (3.6 hours), posting videos, chatting with other players, looking up strategies, managing guilds and so on - all activities where user generated content (UGC) has a core role and that are performed, in most cases, outside the gaming world.

Video games are more and more becoming not just a place that people access to play, but a network of individuals, connected by interests and social identities. The sociability is transforming the industry, and its effects are demonstrated by the growth of communities and tools available for gamers that provide a combination of features to collaborate, create and share content. Some MMOGs have a modifiable interface that allows players to use tools that overlay the game and to extract information from interactions within the game. Power has been given to the user, who can now fashion new contents, be they text, audio, photos, videos, objects or a combination of them all.

Some tools aim to improve in-game communication, facilitate game distribution or guild coordination, improve the capture creation and distribution of customised UGC.

According to a recent industry report from Strategy Analytics, the global market for massively multiplayer online role-playing games (MMORPGs) is expected to be worth $8 billion in 2014 [4]. Nevertheless, despite being on the research agenda for many years and their high commercial relevance, MMOGs have been the subject of surprisingly little research [5].

1.1 The Community Network Game project

This paper describes research on (massively) multiplayer online gamers' requirements conducted as part of the Community Network Game (CNG) project. CNG is focused on applying new network technologies to support community activities over highly interactive centrally managed MMOGs. This will be achieved by developing new tools for the generation, distribution and insertion of user-generated content (UGC) that are friendly to the MMOG client-server traffic.

In-game community activities using In-Game Graphics Insertion Technology (IGIT) and a peer-to-peer (P2P) architecture for the distribution of live video (that is streamed without interrupting the MMOG data flow and the need to upload the video data to a central server) are being developed.

The IGIT is an innovative technology for replacing or inserting content to games in real time without the need to change the game's code in the client or server. For example, billboards can be inserted, areas on the screen can be assigned to display user information, and any type of window (browser, chat, etc.) can be inserted, floating on or outside the game area.

Corresponding author: Eva Ferrari, i2 media research, Psychology Department, Goldsmiths University of London, SE14 6NW, London, UK, 0044 (0)20 7919 7338, e.ferrari@gold.ac.uk

The tools proposed for development by the CNG project include those relating to:

- Video (live video streaming from one user to another user or from one user to many users; record video in-game; easy click and share; pause/rewind live action; edit and annotate video recordings either individually or collaboratively);
- Customisation (personalise characters in video editing; replace/share 3D objects and textures);
- Communication (text, voice and video chat; file sharing; easy in-game access to chat rooms, blogs and forums);
- Other functions (in-game web browsing, voting polls).

All tools will be available in-game, reducing the need for visiting sites or accessing applications outside the game and disrupting the MMOG experience

1.2 Research aims

Research was conducted to understand:

- What motivates gamers to interact with others as part of their MMOG play?
- What do gamers communicate or share with others, and how?
- What frustrations do gamers have with regard to communicating and sharing with others; what are their unmet needs?
- How interested are gamers in a range of community related tools planned for development within the CNG project?
- How are the CNG tools perceived from a stakeholder perspective?

2 METHOD

Both qualitative and quantitative research methods were deployed to explore the research aims described above.

2.1 Semi-structured interviews

Semi-structured interviews were used as a project scoping tool to address the research aims. The discussion covered a broad range of activities in which gamers currently engage (or would like to) during their online gaming experience. Interviewees were prompted towards the end of the interview to ensure that the key benefits that CNG intends to support were addressed. Interviews were conducted face-to-face and lasted between 45 and 90 minutes; each interviewee was given £10 for their time and participation.

2.1.1 Sample

Seventeen gamers were interviewed of which ten were male and seven were female. The sample was aged between 18 and 41 (mean age = 24 years). The sample reported using a range of gaming platforms (PC, console and handheld) and games/social gaming applications covering a range of genres such as RPGs (Role Playing Games), sports, first person shooters and casual.

The analysis of the interviews is presented across all user research sections of the results.

2.2 Online survey

The qualitative research informed the development of an online survey. The aim was to quantify (a) the use and popularity of game-related community activities; (b) the extent of interest in CNG community tools, and; (c) how gamers prioritise their unmet needs, to support the development of CNG tools in the project.

The CNG online questionnaire was piloted with six gamers and improved based on their feedback prior to its launch using the online survey software SurveyMonkey.

2.2.1 Sample

At the time of data extraction, the working dataset for this report totalled 414 respondents who completed at least some of the questionnaire. Sample data presented here and in the results section are reported as proportions of the valid base per question (i.e., excluding missing cases, which is variable across questions).

The sample gave reported ages of between 18 and 99 years (n=302). When the two cases reporting age 99 were removed, maximum reported age was 75 years. Including this isolated respondent, the mean age of the sample was 28.8 years (SD = 11 years; n = 300). The data (with 300 cases) were skewed towards younger ages: almost half of the sample (47.7%) was aged 18-24 years; 28% was 25-34 years, 14.6% was 35-44 years; 9% 45-64 years and only 2 cases (0.7%) reported being aged 65 or older.

The majority (79%) were male and 21% were female (n=314). Respondents were sampled from over 40 different nationalities (n=303) from countries across Europe, The Americas, Australia, Africa and Asia. Around 40% of the sample was from the UK (40%) and over 10% was from The Americas.

Over half of the sample (56.7%; n=312) was in employment, either full-time (38.8%), part-time (13.1%), or casual (4.8%). Students comprised more than a fifth of the sample (22.8%).

In order to explore whether there were differences between gamers in their preferences for CNG functionalities, the sample was segmented based on the item, "*How much of your ONLINE gaming time is Multiplayer/Massively Multiplayer)?"*, which measured gamers' time spent on MultiPlayer Online games (referred to as MPO) and Massively Multiplayer Online games (MMO).

Three groups of MPO/MMO of gamers were created:

- "Non MPO/MMO" (based on those responding "None of my gaming time"),
- "Medium MPO/MMO" ("Less than half/Half of my gaming time"),
- "High MPO/MMO" ("More than half/All of my gaming time").

The findings from the online survey are presented in the results sections 3.3 (Frustrations and unmet needs) and 3.4 (Gamer interest in CNG tools) below. All of the quantitative analyses reported in these sections will refer to this segmentation.

2.3 Stakeholder feedback

To obtain an early view from games industry representatives about CNG, the user research was presented at GameLab 2010, followed by a panel discussion with industry experts.

2.3.1 Sample

On the panel were four games industry experts from the following companies: Redbedlam, CPMStar, Enigma Software Productions, and GameForge. Feedback from the panel is presented in section 3.5 (Stakeholder views on CNG tools).

3 RESULTS

3.1 Motivation to game socially

Interviewees discussed a number of motivations and hurdles to sharing and communicating with others whilst playing (massively) multiplayer online games.

Some gamers either preferred the content in some genres more than others, or were limited in their choice of game by their PC specification. For either reason, genre preference had large implications for the extent to which, and type of, social interactions (communication) that gamers had in-game. For MMORPGs and some other MMOs, social interaction is an integral part of the game play. Guilds or clans in MMORPGs require members to collaborate and mutual support in game play is commonplace. They share various information including their statistics and strategy.

Game play was often prompted by boredom and time availability. The amount of time they had available clearly impacted on how deep they could enter into the game experience as social interaction with others can be a more time consuming element, particularly for MMORPGs. There is often pressure on Guild members for whom the game play requires a level of inter-dependency between players, sometimes to the detriment of participating in everyday real life. Some interviewees found this high social interaction requirement off-putting if they had less free time.

Playing with and against other gamers was perceived by some interviewees as more challenging and fun than playing against the computer as it enabled better skill matching, less predictability in game play, improved realism and it was more fun to work as a team. This was particularly so for highly skilled gamers. Conversely, interviewees with less confidence, experience or skill in the game, or those less attracted by the competitive nature of online social games, reported being slightly put off by the prospect of the shared online game experience. They believed they were unlikely to ‘survive’ for long in the game environment, and were less likely to enter the shared experience until their competence and skills had improved. Some interviewees reported playing against the computer to improve their skills to a level that would increase their skill before entering the shared world. Player skill influenced interviewees’ inclination to share UGC such as game play videos.

Tips and cheats were liked and exchanged by some, but not all, interviewees. For some interviewees, the sense of challenge was diminished when a player had to resort to ‘cheating’ or to ask for someone’s help to progress in the game. The journey towards the reward (the process) was more important for some interviewees than the reward itself.

The degree to which interviewees reported sharing their performance and know-how varied. Some enjoyed the status afforded by the visibility of their achievements in the game. Others who were more motivated to game by the personal challenge were less interested in ‘bragging’ to players or in accessing others’ clips about their performance.

Interviewees were relatively polarised in their preference for playing either with people they knew in the real world, or with online strangers. The size of the group or number of people they chose to interact with also varied in our sample.

Some had a preference to play with familiar others, such as friends and family because it complemented their real world social interactions, and they reported having better game rapport with them. Game play with friends of friends was viewed as a way of initiating future real-world interactions. The games they played were directly influenced by their peers.

Conversely, other interviewees preferred to play with strangers. One interviewee admitted that he felt stigmatised as a gamer, and preferred his real world friends to be unaware of his game play. This would restrict this type of gamer’s propensity to be willing to share in-game experiences with real world friends out-game. Some interviewees enjoyed the anonymity associated with playing with strangers in that it enabled them to fully exploit their character – to be who they wanted to be without shame or embarrassment. Playing with strangers could also be of benefit strategically in that their anonymity enhanced their confidence to interact in more ‘sneaky’ or manipulative ways with others during game play. Other more social gamers in our sample enjoyed developing online-only friendships with other gamers. They noted that trust was built over a period of time through the games they mutually played. This was particularly true for MMOGs, where players spend many hours with other members of their clan or Guild pursuing a common aim. Because of this, gamers were often able to bond with their online friends, often developing meaningful relationship.

Overall social interactions were perceived positively in the game environment and contributed at least towards a general sense of in-group inclusion and belonging. Sharing information with others was mostly reported by interviewees as a positive, satisfying experience that they hoped would be reciprocated at another point in time.

3.2 Communications between players

Interviewees reported communicating and interacting with others for different reasons and through a range of methods. Information sharing was direct (ask of/respond to another player) and indirect (e.g., inferring a player’s ability or potential usefulness from their character stats and qualities). Sharing was reported to occur in-game (whilst playing, without minimising the game window) and out-game, using other applications. Player skill relative to others' often influenced the degree to which our interviewees reported asking or giving information. Text-based chat was a common method of communication, often enabled in the game itself, but applications outside of the game were also used such as Windows Live Messenger and Facebook chat. In contrast voice was used less often but where it was, applications such as Skype, Steam voice chat and Ventrilo were used. Cost, ease of use, practicality, and anonymity were issues for users of voice chat. None of our interviewees used video chat.

Around half of our sample used or was aware of game tool set applications such as Xfire, PlayXpert and Steam that offer a range of community related functions. Guilds and (official and unofficial) game forums were commonly accessed spaces for players to exchange information. Gamers who created content for others used video recordings of their game play (e.g., using Fraps) or constructed image (e.g., screen shots) and text instructional guides which they disseminated in game related spaces such as forums, or wider community spaces such as YouTube and Facebook. Low awareness of such tools and poor ease of use were hurdles to interviewees' adoption of such services. Search engines such as Google were also used to find game related information.

Interviewees reported sharing different types of information to: (a) support their game progress; (b) enhance the game experience; and (c) enhance interpersonal and community relationships.

3.2.1 Support game progress

Examples of exchanges to support game progress included: direct in-game supportive information, such as character-to-character exchange of help; strategy, particularly for Guilds and raids; tips, cheats and hints, for instance via a secondary out-game source; achievement and stats denoting status and skill which can be useful to others in deciding whether or not another player may be of help to them.

3.2.2 Enhance the game experience

To 'spice up' their game experience, interviewees reported creating and/or accessing a range of customisation tools such as modifications (mods) or add-ons that are often installed to enhance or improve interaction with the game.

However, some interviewees disliked these types of customisation tools particularly those that reduced the challenge or disrupted the original game.

3.2.3 Enhance interpersonal and community relationships

Finally, interviewees reported communicating and sharing with others for a number of interpersonal and community reasons. These included: 'bragging' (showing off) about performance and skills; sharing funny clips for entertainment and humour that bore no practical use to improving one's game play; affect and emotion to enhance the communication between players in game using emoticons, keycodes and macros; insults, which were occasionally reported by our interviewees in their experiences of the somewhat anonymised game world; their identity as a gamer as well as out-game interests.

3.3 Frustrations and unmet needs

Whilst communicating and sharing among players was generally viewed positively, it was also associated with a number of frustrations. Insights were initially obtained from interviewees who were asked to indicate the reasons for any frustration that they experienced when using additional applications and tools with their MPO/MMO games.

3.3.1 Results from the interviews

Some interviewees noted how attempts to increase the level of social interaction in games by enabling UGC to be more easily shared in-game could easily destroy the sense of immersion which is fundamental to some people's game play. Minimising the game screen to do other things was a frequently reported frustration amongst our interviewees. Also noted were issues with 'noise' in connecting with others using voice chat, the impact of additional applications on speed (e.g., of displaying elements in the environment, action-responses), pop-ups, and advertising in general within game play.

A range of communication issues were raised and there was large variation across interviewees in their communication needs. Challenges were reported in some cases in arranging team and Guild meetings. Players are located worldwide, in different time zones, and may be online but not in the game.

For many interviewees, poor ease of use of applications to enhance and expand on their game experiences (e.g., in-game recording) explained their non-use of them. Many reported that if the process was made easier, they would be more likely to try it. This was particularly evident for recording game play.

Downloading game-related extras and click-throughs were treated with caution by some of our sample for fear of infecting their computer with trojans and viruses.

3.3.2 Results from the survey

To quantify which were the biggest frustrations, the relevance of these issues (outlined in 3.3.1) to gamers were explored in the online survey. Respondents were asked, "*When you use additional applications and tools with your online multiplayer/MMO games (e.g., Instant Messaging (IM), video/image capture programs, etc.) which, if any, of the following are frustrations or concerns for you?*".

Results revealed that, among the 152 Medium and High MPO/MMO players who selected at least one of the response options to this question, only 14.5% reported that they did not have any frustrations (see Figure 1).

More than a third, however, indicated that the need to minimise the game screen to access community tools was their biggest frustration, followed by "Reduced immersion" and "Technical instability", both selected by over a quarter of the respondents (26%). The quantitative insights also supported findings in relation to intrusive advertisements. Pop-ups, especially when out of context, were perceived as irritating by 24% of respondents.

The survey's results also reflected concerns over "Security against trojan and viruses" and "Usability of applications", both indicated as reasons for irritation by 23% of the sample.

Gamers showed slightly less concern for issues such "Privacy" (18%). Only a minority of respondents indicated that coordination of Guilds/teams was a reason for frustration, whilst "Too few chat channels" was the least frequently endorsed frustration (3%).

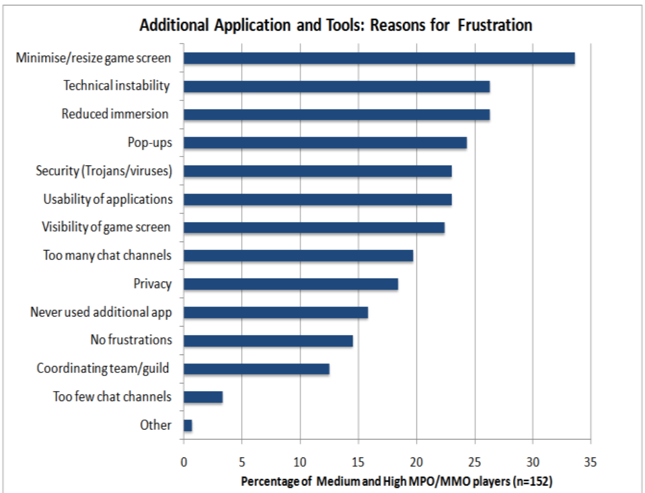


Figure 1: Reasons for frustration on use of additional tools

3.4 Gamer interest in CNG tools

Results from the qualitative research indicated that interviewees already used a number of tools similar to the ones planned for development in CNG. However, their access and use of existing tools was associated with a number of frustrations and unmet needs, which were supported by the quantitative findings from the online survey.

To identify how interesting the CNG functionalities were perceived by gamers, respondents to the online survey were asked, using a 3-point scale ('not at all', 'somewhat', 'very' interested), to indicate their level of interest in each CNG function. Functionalities that were mentioned during the interviews (beyond CNG's primary scope) were also included in the survey.

Before answering the questionnaire, respondents were briefed on the project's purpose (i.e., to develop a suite of functions that support community activities for gamers) and were reminded that the technology behind the project was able to develop functions that are available in-game (e.g., no need to minimise the screen to use them) and without affecting the game experience (e.g., no risk of slowing down the computer/device).

The vast majority of the functions listed were rated as of interest ('somewhat' or 'very') to over 50% of the High MPO/MMO gamer sample, with a third of functions rated as of interest to at least 70% of these respondents.

The functionalities or services that appealed to the largest proportion of High MPO/MMO gamers were: Complex game statistics (80%), Availability of players across games (77%) and Skill matching (71%); all functions that require some form of game integration.

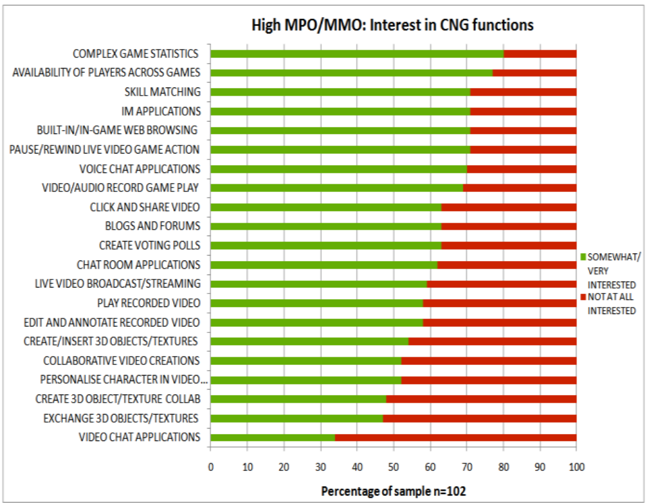


Figure 2: Interest in CNG functions by High MPO/MMO gamers

With regard to functions that do not need such integration, notably more respondents reported interest in Video and Communication-related functions than in Customisation functions.

All Video functions were described as of interest to over 50% of the most dedicated MPO/MMO gamers: Pause/rewind Live Video Action (71%), Video/Audio Record Game Play (69%) and Click and Share Video (63%) were the most popular functions, followed by Live Video Broadcast/Streaming (59%) and Play Recorded Video (58%). These results are consistent with the qualitative research which indicated that the most committed gamers have a need or desire, to enrich their game experience with videos in an easy/accessible way. However, Collaborative Creation of videos was a slightly less popular function, receiving interest from 52% of this High MPO/MMO sample.

The interviews revealed that communication is central for gamers, particularly so for multiplayer gamers. In line with this finding, the online survey results showed that tools such as IM (71%) and Voice Chat Applications (70%), Blogs and Forums (63%), and Chat Room Applications (62%) were all of interest to the vast majority of High MPO/MMO gamers. The only Communication function for which gamers had reservations was Video Chat, with only 33% showing interest in it. This was not surprising and was consistent with findings from the qualitative data, which showed that Video Chat could ruin the anonymity of the game which is particularly important when playing with strangers.

Despite that Customisation functions were not perceived as a priority, over half the sample showed interest in functions such as Edit/Annotate recorded Video (58%); Create/Insert 3D Object/Textures (54%); and Personalise Characters in Video Editing (52%).

Fewer respondents were interested in tools that allowed gamers to Exchange 3D Object/Textures or to create them collaboratively, (47% and 48% respectively).

Other functions proposed by CNG such as In-game Web Browsing (71%) and Creating Voting Polls (63%) were found to be popular amongst the most dedicated gamers.

3.5 Stakeholder views on CNG tools

Stakeholders acknowledged the importance of community building tools in games citing an influx of new gamers drawn to the opportunities to socialise. They also considered there to be gaps in the existing market for specific community, particularly communication, related tools. Information sharing has an integral role in the online experience as the social aspect of games is becoming ever more important.

One of the main issues identified by stakeholders concerned the impact of CNG tools on the illusion of the game; gamers enjoy the immersion into the fantasy environment, thus efforts to maintain this illusion are critical. The panel asserted that tools should be designed such that they do not harm the gaming experience and need to be carefully tested for stability and reliability across different games. Stakeholders noted that acceptance by gamers would also depend on the level of interaction that the tools provide. In contrast to end users of other services, stakeholders pointed out that gamers often expect to influence how the game service is run or how content is delivered to them. According to the experts, tools that support feedback from users to game developers are needed.

The industry experts noted that when players choose their tools, they are driven by the freedom and control over what they see on the screen. Different segments of the gamer market are likely to want different features. Therefore it was considered essential for CNG to offer a versatile service that users can easily adapt to their specific preferences, needs or skills. Personalisation was identified as one of the main key elements for the success of CNG; both in terms of the choice of tools available and the design of the interface that needs to suit the online game world.

The expert panel considered that the in-game video streaming tool would be of most interest to game developers. However to maximise adoption by users they indicated a need to establish whether or not the European games market is ready for in-game video streaming prior to its introduction.

4 CONCLUSIONS

CNG is conducting research to inform the development of in-game community activities that enable gamers to create, share, and insert UGC. The UGC considered by the CNG project includes 3D objects, graphics, and video. The objective of the research outlined in this paper was to understand gamers' current needs and unmet needs for in-game communication and community related tools; to quantify gamer interest in the range of tools proposed in the CNG project; and to identify initial views of the project by stakeholders.

The interviews with gamers identified both a wide range of motivations to engage in community related game activities and various difficulties associated with existing tools. These qualitative findings were supported by the results of the stakeholder research and the online survey which found that amongst High MPO/MMO players in particular there is interest in and a need for improved tools to support community activities around their game use, especially in relation to video and communication. Furthermore, whilst these needs are currently met through the use of standalone applications running in the background to the MMOG (e.g., Google talk for text chat; Skype or Ventrilo for VOIP (Voice over Internet Protocol), Fraps for video capture, the biggest frustrations reported by MMOG players were delays and instability caused by the need to minimise the game screen or to switch to an external application for community tools. This is encouraging for the CNG approach which intends to offer such tools without the need to minimise the game screen. Furthermore, as identified in the interviews and confirmed by the survey and stakeholder results, efforts to maintain the game illusion are critical to avoid harming the game experience. Finally, in relation to future game development and project partner interaction with game developers, a number of functions that were popular with gamers in this research, such as skill matching, complex statistics and availability of players across games, could be more elegantly implemented with access to the game.

Acknowledgment

The CNG project is funded under the ICT (Information and Communication Technologies) priority of the European Union's FP7 (Seventh Framework Programme) (ICT-248175). The project is a STREP for 30 months.

References

[1] Williams, D. (2006a). A (brief) social history of gaming. In P.Vorderer & J.Bryant (Eds.), Video Games: Motivations and Consequences of Use. Mahwah, New Jersey: Erlbaum.

[2] http://www.nickyee.com/daedalus

[3] Yee, N. (2009). *Changing the Rules*: *Social Architectures in Virtual Worlds.* In Bainbridge, W. (Ed.), *Online Worlds: Convergence of the Real and Virtual* (pp. 213-223). London: Springer.

[4] http://blogs.strategyanalytics.com/DMS/post/2010/08/10/Global-MMORPG-Market-to-hit-$8-billion-in-2014.aspx

[5] Caplan, S., Williams, D., & Yee, N. (2009). Problematic internet use and psychosocial well-being among MMO players. Computers in Human Behavior, 25,1312-1319.

PROVIDING COMMUNITY AND COLLABORATION SERVICES TO MMOG PLAYERS*

George Adam, Christos Bouras, Vaggelis Kapoulas and Andreas Papazois
Computer Technology Institute & Press "Diophantus"
N. Kazantzaki, Panepistimioupoli, Rion 26504, Greece

ABSTRACT

Most of Massively Multiplayer Online Games (MMOGs) aim to support and improve the community activities among their members. The Community Network Game (CNG) project has focused on services supporting players during their in-game community and collaboration activities. The tools we offer use an innovative technology, namely the In-game Graphical Insertion Technology (IGIT), which permits the addition of web-based applications without having to modify the game code. These tools follow a design adapted to the needs of MMOG players. They are provided by the so-called CNG Server and have been implemented based on recent advances in Web 2.0 technology. The in-game collaboration activities provided by the underlying framework are also enhanced using Peer-to-Peer (P2P) technology for the distribution of user-originated live video. In this paper, we concentrate our focus on the architecture of the server as well as on the design and implementation of the community and collaboration tools.

KEYWORDS

Massively multiplayer online gaming, gaming communities, user-generated content, game adaptation, system design

1. INTRODUCTION

The Community Network Game (CNG) [5] is a research and innovation project funded by the European Commission to research and develop in-game community applications for MMOGs. Currently, many Massively Multiplayer Online Games (MMOG) providers offer various tools for building online communities and for sharing User-Generated Content (UGC) as an integral part of their games. Having a common look-and-feel of out-of-game features for different games is a recent trend and several game-adaptation frameworks were born for this purpose. Steam and XFire are such applications, with many surveys, like [4] and [8], mentioning XFire [10] to be the most important one among them. XFire provides a set of tools that is displayed in an overlay on top of the screen. It is a free product and does not change the game code. Some of the key features supported by Xfire are: text and voice chat, screenshot and video sharing, live video streaming, interaction with external Instant Messaging (IM) networks, web browsing, etc.. All these functions can be started in the game-screen without any player's obligation to leave the game window.

The progress beyond the current game adaptation products offered by CNG can be summarized in the following points. It should be noted that all technologies mentioned below are developed in an MMOG "independent" and "friendly" manner:

- Cutting-edge network technologies for live video sharing over Peer-to-Peer (P2P) [2].
- Game-oriented social networking, collaboration and video-editing tools.
- Innovative web technologies suitable for in-game rendering of online community and collaboration tools.
- Advanced game adaptation technologies allowing the user to replace 2-D textures and inserting 3-D objects in selected game scenes.

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

CNG makes use of In-game Graphical Insertion Technology (IGIT), which is an innovative technology permitting the replacement of game objects and the insertion of UGC as an overlay of the game scene in real-time. Using IGIT external web-based applications can be rendered, as a new layer on top of the game screen. In this way, innovative community and collaboration activities can be easily introduced as in-game features. It is important that IGIT functions are offered, without changing the game application in the MMOG client or in the server [3]. The MMOG "independence" allows the same technology to be used in multiple games, and therefore CNG consists a generic application with a single installation being able to address directly to multiple MMOGs and MMOG operators. The provision of the new community and collaboration features will be provided by a deployment separate from the MMOG delivery network and thus, by moving the processing and network load away from the MMOG server or content delivery network, new demanding features and applications of can be offered to the users.

The CNG community and collaboration tools are provided by the CNG Server, which consists the core of the CNG system. The implementation of the online community and collaboration applications is based on Web 2.0 technology, which is widely used for social and UGC-sharing activities. In this paper, we describe the high-level architectural design of the system components developed to support CNG community and collaboration tools. In addition, we provide a detailed description of the architectural elements, starting with details on the CNG Server's components and continuing with the online web applications. The most important aspects of system's design and implementation are described, including the used interfaces and protocols, the interaction of the system modules and the adopted technologies. We concentrate our focus on the features that distinguish the CNG services from a typical social networking system and we issue any design aspects that can contribute to future standardization processes for WWW and open social networking.

2. DESIGN & ARCHITECTURE

2.1 User Interface

The graphical design for widgets implementing the various tools follows a simple and clear approach, so as not to dominate the game scene, thus avoiding to distract the users, and to affect their in-game experience. The tools-widgets are web pages that are rendered inside the game window, through the IGIT technology, which allows the in-game display of web content.

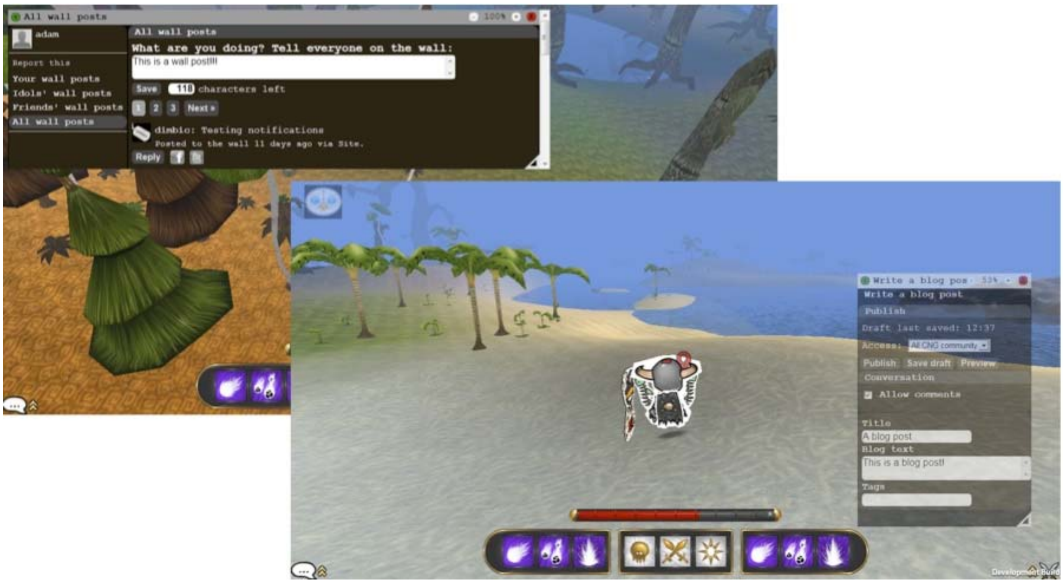


Figure 1. RedBedlams's "The Missing Ink" game with integrated CNG collaboration tools

The CNG tools use advanced web technologies to provide customizable layouts. This is required in order to allow both for different positioning options in the different games, since the CNG collaboration tools should not overlap the game's on screen tools and options, and for the users' various preferences and styles. Different layouts can be used for different scenes of the same game and/or different views of the same scene.

The CNG tools appear as floating windows within the game screen and can be considered as part of the game. Thus the tools enhance the in-game experience without requiring the players' to break their gameplay to access external tools. Two different positioning, transparency setting and sizing options are depicted in the two screenshots of Figure 1 using the CNG framework with RedBedlam's “The Missing Ink” [9]. One shows a vertical alignment of the panes of the blogging widget, and the other shows a horizontal alignment of the panes of the “Wall” widget.

Although positioning the CNG collaboration tools outside the game window, e.g., next to right side of the game window, was an option, gamers strong expressed their preference for tools rendered on top of the game window.

The support of multiple game resolutions and screen sizes is a big challenge for the design of the CNG User Interface. The CNG online community and collaboration tools have been designed independent of the game resolution selected by the user. The independence from the game screen configuration is not just limited to the size of the CNG widgets but extends to their web content. When the session is initialized, the CNG Server dynamically generates the appropriate dimensions for the CNG windows, as well as the proper Cascading Style Sheet (CSS), using information for the game screen configuration sent by the CNG Client, in order to achieve a consistent layout for the CNG tools' windows.

2.2 Web Server

The web server, as part of the CNG Server, provides the online community and collaboration services. The web server consists of two components: the presentation-tier, which is the UI component, and the business logic-tier, which is the backend component. The implementation of the web server is based on the Elgg social networking engine [6].

The UI Component is the interface of the overall web server with the CNG Client side and in particular with the web-browser instances in it (at this point it should be recalled that the client side of the CNG system is executed within a container that, using instances of a web-browser, displays web content on top of the game screen). The UI component serves the client side with the user interface of the CNG Web 2.0 online collaboration tools, as web content (HTML, JavaScript and Flash objects). The web browser instances in the CNG Client side communicate directly with the UI component requesting these services’ layout and content. Thus, the CNG user interface and the CNG Web 2.0 online collaboration tools can be updated and extended without the need to change the CNG Client. In addition the UI Component provides of the web-based interface used for the administration of the web server. This is only available to administrators of the CNG system (i.e. users with relevant privileges).

The Backend component is the mid-tier between the other parts of the system, i.e., the UI Component and the Database and it consists the core of the web server. The Backend component provides all the functionality that is related to the Web 2.0 community and collaboration applications of the CNG system. This includes the management of these web applications, and the management of users, access groups, and. all the other involved entities. Finally, the Backend component is located between all other components and the Database. This way it hides the schema and implementation details, decoupling the Database from the rest of the system.

The CNG Server component provides the ability to users to interact with their accounts in external social networks. Towards this direction, the system provides interconnection with Facebook and Twitter social networks with single sign-on functionality. An one-time setup phase has to be completed by the user in order to enable this feature. The user provides his credentials for his Facebook and Twitter account and authorize the CNG application to have access to his accounts. When completing this setup phase CNG server receives an access token from the external social networks and acquires permanent access. It is important to note that CNG Server does not store the user's credentials for obvious security reasons. Once the user authorizes the CNG application, then he is able to post the messages that are displayed in CNG Web tools to his his Twitter and Facebook walls. The server side does this posting, without prompting the user to enter his credentials. This procedure can be considered secure, as the access tokens are valid only for usage by the CNG application.

2.3 Chat Application

Chat application allows users to communicate in groups using instant messaging and voice while being logged in to the system. It also provides private chat between two users. In group chat, the users are able to enter virtual rooms that others have created and communicate with all the users of the room. It is obvious that the users of the room have to be online. However, in the case of private messaging, the messages sent to offline users will be delivered when the users log back in. At this time, a notification message for new chat messages is presented to the user. CNG Server is responsible for room management and also for access management.

The implementation of the web based Chat Application is based on Adobe Flash [1] technology. This technology was selected after examining the capabilities of other solutions like Java and HTML. Especially, the selection of HTML5 was discouraged, as it does not support voice capturing. In addition, Adobe Flash presented better animation quality when comparing with both HTML5 and Java applets.

Chat services require an infrastructure, which will enable them to operate. For this purpose, a media server is used as an application server. Red5 Media Server [7] was selected, as it is an open source multimedia server that supports media streaming features. While the user is connected to the system, the chat application maintains a permanent connection between the user and the media server. The user can retrieve data asynchronously without interfering with the display and behavior of the web page within the browser. Chat application can be considered both transmitter and receiver of data. It is able to capture voice and text messages and transmits them to the media server. In parallel, it can receive voice streams and messages and reproduce them in the client side.

2.4 Database

The database is an essential part of the CNG Server's architecture since it stores data for the CNG community and collaboration tools. The web server and the CNG tools are based on the Elgg social networking engine [6], and therefore CNG database schema is based on Elgg's schema. This schema models the entities and the general entity relationships, while it allows custom entities and relationships. In addition, it can store access rights to objects that can be used for authorization purposes.

The database schema provides two ways to store additional information to entities. The first way is the addition of metadata, which can be considered as adding attributes to an entity. This way is used in order to prevent altering the schema. The second way is the addition of annotations, which are generally information added by third parties like comments and ratings. Metadata and annotations are modeled by tables that store the strings of extra information, which are linked to the entities.

2.5 Administration & Monitoring

In online systems like CNG where many users are served at the same time, it is essential to monitor the system's performance and also the online tools usage using administrative and monitoring tools. There are many cases where administration and moderation must be applied when the users are free to interact with the online community. In these cases, the administrative tools are used in order to supervise the CNG community network and avoid the distribution of inappropriate content. Using the infrastructure of the CNG tools, the user can easily report to the administrator any user or content.

The administrative tools provide an interface for the administrators to manage the CNG system. The management can be done using network administration tools and a control panel for users' accounts administration, which actually provides full access to users of the system. This control panel supports operations for adding, removing, activating, deactivating a user and also resetting passwords and promoting users as system administrators.

Information about system usage can be considered valuable when analyzing complex online systems with many users. CNG monitoring tools log the users' activity within the CNG tools and intend to provide information on the users' behavior. These tools are able to provide detailed logs for each user or tool and aggregated visualized statistics. The monitoring procedure intends to analyze the traffic of the CNG tools usage at the highest possible level. The main functionality of monitoring tools relies on collecting real data and analyzing them in order to extract knowledge about the usage of CNG community tools. The system logs every HTTP request and then extracts the requested URL and the HTTP request method. With this information it can map each request to a specific action of the user, like writing a wall post, viewing a live-stream, upload a file, etc..

The implementation has been based on the Elgg [6] framework's infrastructure, so that the monitoring tools are used in combination with existing online verification tools in order to provide a complete view of the system's behavior. CNG Server provides a system diagnostics tool that can be used for inspection of all the configuration information of the system, like the paths to the files installed in the web server and all the installed collaboration and community tools accompanied with necessary related details.

3. CONCLUSION

In this paper, we presented the online system that was developed as a part of "The Community Network Game" (CNG) project. CNG aims to provide online community and collaboration services to MMOG players. We presented an introduction to its key features, like the IGIT technology and the live video streaming over the peer-to-peer network. We described the high-level architectural design of the CNG system and highlighted the system's design and implementation for the provision the online community and collaboration tools. We showed the key features of the CNG Server and also covered some important aspects of the CNG Client. Our analysis included details on the user and administration interfaces, the element's structure and the functionality for both the CNG Server and Client.

ACKNOWLEDGEMENT

The authors wish to thank Kerry Fraser-Robinson from RedBedlam for his ideas on the in-game social terminology, Jonathan Freeman and Eva Ferrari from i2media research for their recommendations on the users' needs, as well as all the partners belonging to the CNG project consortium for their collaboration.

REFERENCES

[1] Adobe flash multimedia platform software. http://blogs.adobe.com/flashplatform/

[2] S. Ahmad, C. Bouras, R. Hamzaoui, A. Papazois, E. Perelman, A. Shani, G. Simon. The community network game project: Enhancing collaborative activities in online games, in Proc. of Networked & Electronic Media Summit (NEM Summit 2010) (Barcelona, 2010)

[3] S. Ahmad, C. Bouras, R. Hamzaoui, A. Papazois, E. Perelman, A. Shani, G. Simon, G. Tsichritzis. The community network game project: Enriching online gamers experience with user generated content, in Proc. of 2nd Int. Conf. Creative Content Technologies (CONTENT 2010) (Lisbon, 2010)

[4] S. Bergstrer, T. Hildebrandt, C. Rensing, R. Steinmetz. Virtual context based services for multiplayer online games to facilitate community participation, Multimedia Tools and Applications 45, 347 (2009). URL http://dx.doi.org/10.1007/s11042-009-0293-9.

[5] CNG project - the community network game. http://www.cng-project.eu/

[6] Elgg - open source social networking engine. http://elgg.org/

[7] Red5 media server. http://www.red5.org/

[8] S. Shen, A. Iosup. The xfire online meta-gaming network: Observation and high-level analysis, in Proc. of International Symposium on Audio-Visual Environments and Games (HAVE 2011) (Qinhuangdao, Hebei, China, 2011)

[9] The missing ink. http://www.missing-ink.com/

[10] Xfire gaming simplified. http://www.xfire.com/

Hindawi Publishing Corporation
International Journal of Computer Games Technology
Volume 2012, Article ID 969785, 13 pages
doi:10.1155/2012/969785

Research Article

Building Community and Collaboration Applications for MMOGs

George Adam, Christos Bouras, Vaggelis Kapoulas, and Andreas Papazois

Computer Technology Institute & Press "Diophantus" and Computer Engineering & Informatics Department, University of Patras, N. Kazantzaki, Panepistimioupoli, 26504 Rion, Greece

Correspondence should be addressed to Christos Bouras, bouras@cti.gr

Received 20 March 2012; Accepted 13 July 2012

Academic Editor: Mark Green

Supporting collaborative activities among the online players are one of the major challenges in the area of Massively Multiplayer Online Games (MMOG), since they increase the richness of gaming experience and create more engaged communities. To this direction, our study has focused on the provision of services supporting and enhancing the players' in-game community and collaboration activities. We have designed and implemented innovative tools exploiting a game adaptation technology, namely, the In-game Graphical Insertion Technology (IGIT), which permits the addition of web-based applications without any need from the game developers to modify the game at all, nor from the game players to change their game installation. The developed tools follow a design adapted to the MMOG players' needs and are based on the latest advances on Web 2.0 technology. Their provision is performed through the core element of our system, which is the so-called Community Network Game (CNG) Server. One of the important features provided by the implemented system's underlying framework is the utilization of enhanced Peer-to-Peer (P2P) technology for the distribution of user-generated live video streams. In this paper, we focus on the architecture of the CNG Server as well as on the design and implementation of the online community and collaboration tools.

1. Introduction

One of the most interesting potentials that Massively Multiplayer Online Games (MMOGs) developers and operator have is that they can offer to the online players the possibility to interact with a large number of other players as well as to collaborate and compete in a large variety of gaming situations. The majority of these activities are made in the context of online communities that can be built around the game and where players of MMOGs tend to join to connect with people with common interests and passions and to share their in-game experience [1]. The Community Network Game (CNG) [2] is an EU-funded research and innovation project that researches and develops in-game activities using the In-game Graphical Insertion Technology (IGIT) and that proposes an architecture that combines efficiently the client-server infrastructure for the MMOG activities with a Peer-to-Peer (P2P) overlay for the delivery of user-generated live video.

IGIT is an innovative technology that permits replacing game's objects and inserting User-Generated Content (UGC) within the game in real time. Using IGIT, the in-game avatars can be customized with players' images, external 3D objects can be inserted within the game screen, and other types of windows, for example, external applications, can be inserted as a new layer on top of the game screen. It is important that IGIT's features are offered without the need for the game developers to change the game's code in the MMOG client or server [1]. This MMOG "independence" is achieved through a game adaptation process and an integrated browser's rendering engine, which permits that the same technology implementation can be used in the same way on multiple games, and thus making CNG a generic application able to address directly multiple MMOGs and MMOG operators.

In this paper, we describe the high-level design and the system architecture components developed within CNG system for the support of community and collaboration applications. Furthermore, we provide a detailed design of the system's architectural elements starting with details about the CNG Server's components and continuing with the Web 2.0 applications. The most important aspects of system's design and implementation are described. the used interfaces and protocols, as well as the system modules and the internal design specification. We focus on the features that distinguish CNG collaboration tools from a typical social networking website and we highlight any design issues that can potentially contribute in future standardization processes for WWW and open social networking.

The remainder of the paper is structured as follows. Section 2 reviews existing systems in the area and compares similar systems' achievements with those of CNG. Section 3 provides an overview of the CNG system's architecture and focuses on the supported community and collaboration tools. The design of the CNG Server architecture with regards to the online community and collaboration tools is presented in Section 4. The first results on our collaborating experience within CNG environment are presented in Section 5. In Section 6 we conclude, and, finally, in Section 7 we present our expectations as well as the planned next steps of this work.

2. Similar Systems and CNG Progress

Currently, many MMOG creators tend to build online communities around their games and to allow their players creating their own content and sharing it. In order to have a common look-and-feel around multiple games, a new trend towards the game adaptation frameworks and Online Meta-Gaming Networks (OMGNs) was arisen. There are several game adaptation and OMGN products that have been created so far, like Steam [3] and XFire [4], with the XFire being the most notable one among them, as mentioned in several studies including [5, 6]. XFire achieves game adaptation and provides a set of tools that appear in an overlay on top of the game screen. XFire is a free product and does not need any modification of the game code. Some of the most important features that XFire supports are the following:

(i) text and voice chat,

(ii) screenshot and video sharing,

(iii) live video streaming,

(iv) interconnection with external Instant Messaging (IM) networks.

(v) web browsing.

All of the above features can be launched within the game screen, eliminating the disruption of the player's in-game experience since quitting of the game application is not needed.

Using IGIT, CNG is able to introduce innovative community and collaboration activities between gamers. MMOG serving systems can provide only limited and low-volume out-of-game services via their central servers. By the reduction of the processing and network load, the new game features and applications offered by CNG are able to overcome the limits that the majority of the current MMOG providers face. CNG community and collaboration applications appear within the game environment and can be accessed and manipulated by the players without the need to interrupt their gaming experience. These services are offered by the CNG Server (or multiple CNG Servers) which is the core of the CNG system architecture. One of the most important services offered by CNG is the streaming of user-generated live video among the online players using P2P technology. In the context of this service, CNG has researched and developed an innovative architecture that permits sharing of live video streams between multiple players via P2P without interrupting the MMOG data flow [8]. This P2P technology includes techniques for the creation of a dynamically optimized scalable network for the distribution of live video streamed from one player to another or from one to many players. The generated live stream traffic constitutes a major challenge given that it creates flows within a network already used by the MMOG data flows. To this direction, the project has researched and developed new techniques for P2P live video streaming that respect the MMOG client-server traffic (i.e., show "MMOG-friendliness"). The corresponding P2P tracker constitutes part of the CNG Server and has persistent communication with the CNG Client applications (peers) being responsible for the coordination and organization of the peers in order to assure an efficient and "MMOG-friendly" P2P live video communication. It should be noted that the implementation of online community and collaboration applications is based on Web 2.0 technology which is widely used for social and UGC-sharing activities. Further enhancements have been added to support the in-game activities as well as to make use of the features offered by the IGIT technology.

Table 1 lists the CNG key features and shows whether they are offered by the current game adaptation products or not. The most significant in-game adaptation frameworks have been considered in this comparison, namely, XSteam [3], XFire [4], PLAYXPERT [9], Overwolf [10], and Raptr [11]. It is obvious that XFire is the most notable among the similar systems. The progress of CNG beyond the current similar systems can be summarized in the following directions:

(i) innovative network technologies for live video sharing over a P2P overlay network,

(ii) game-oriented social networking, collaboration and video-editing tools,

(iii) innovative web technologies appropriate for the in-game rendering of the online community and collaboration tools,

(iv) advanced game adaptation technologies for 2D texture replacement and 3D object insertion at game-spots indicated by the user.

It should be noted that all of the above technologies have been developed in an MMOG "independent" and "friendly" manner.

3. The CNG System

In this section we provide a high-level overview of the CNG system's architecture and present the online community and collaboration tools designed to support MMOG communities.

Table 1: CNG features support by the most significant similar systems (source: [7]).

Feature	XFire	PLAYXPERT	Steam	Overwolf	Raptr
Texture replacement					
In-game overlay	✓	✓	✓	✓	✓
Video capture	✓	✓		✓	
Video edit					
Video upload	✓			✓	
Live video	✓				
Instant messaging	✓	✓	✓	✓	✓
Audio chat	✓		✓		✓
File sharing					
Online blogging					
SDK free	✓	✓		✓	✓

3.1. Overview of CNG Architecture. Figure 1 provides an overview of the CNG system's architecture. The CNG system consists of two basic elements, namely, the CNG Client and the CNG Server. The CNG Client has been designed as a framework that runs together with the game itself on the user's machine. On the other hand, the CNG Server consists of a set of components providing the various services for community and collaboration activities as well as for live video sharing. Figure 1 also depicts the MMOG Server entities as well as the network flows related to the CNG and MMOG services provision. It is obvious that the generic MMOG architecture is not modified since the game content is still transferred through the MMOG Server(s).

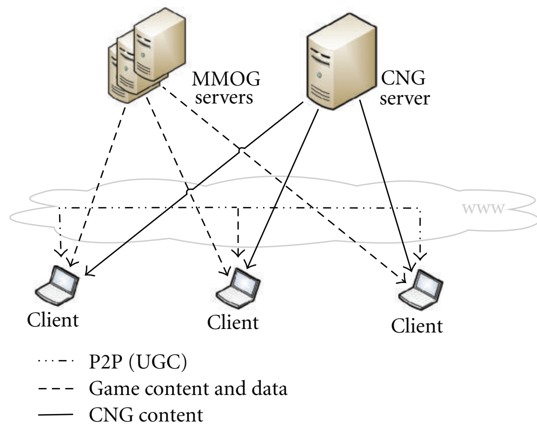


Figure 1: The CNG network architecture (source [1]).

The CNG Client is an application that integrates the game application and provides the framework enabling the user to access the MMOG as well as the CNG toolbox. This framework is able to embrace any online game application simply by a configuration process that can be executed easily by an inexperienced user. The CNG Client includes all the software that is necessary for the game adaptation, that is, the IGIT libraries, as well as the P2P module.

The CNG Client's P2P module is responsible for the live video sharing over P2P among the peers. It has a dual role based on whether it is located in a source or sink peer. In a source peer it is responsible for the captured video adaptation and streaming to the appropriate sink peers. Within a sink peer the P2P module is responsible for receiving the video streams, retransmitting them to other peers when necessary, as well as for the adaptation of the received video data in order to play back the video initially captured in the source peer. In order to achieve the above procedure in an efficient manner, the P2P module interacts with the P2P tracker in order for the P2P tracker to build and manipulate the P2P overlays. The P2P module also includes mechanisms that add reliability and improve the performance of the P2P system. These mechanisms include schemes for congestion control, forward error detection, and correction of packet losses. The P2P module has also been designed to create a UDP traffic that is "friendly" to the game's TCP traffic. "MMOG friendliness" has been implemented in the sense that the CNG P2P module is able to adapt the use of network resources by identifying the available network capacity and thus respecting the game traffic flows [8].

Within the CNG Client framework, the various CNG tools are offered to the players through the CNG toolbox. The CNG toolbox is the web application that is initially rendered within the game screen through the IGIT technology. It provides access to the various Web 2.0 tools for social networking, collaboration, video editing, and video sharing over P2P, which are all developed within CNG project. CNG toolbox also provides access to other services like access to external IM applications and web search engines. The IGIT module provides controls for the look-and-feel of all the out-of-game CNG tools on the game screen including their size, position, and transparency. A JavaScript API has been implemented within the IGIT module to provide the necessary IGIT functionality to the Web 2.0 tools [1].

The CNG Server is the core of the CNG system and has a twofold role within it. First, it provides the CNG online community and collaboration web services to the connected CNG Clients, and, second, it monitors the P2P communication as a P2P tracker. As a web server, CNG Server provides all the CNG community and collaboration services, the video and graphics tools that are provided in the CNG set of tools. As a fully integrated and standalone system, web server also hosts additional applications like media services, administration UIs, and applications for the interaction with external social web services like Facebook, Twitter, and YouTube. It should be noted that the CNG Server architectural design permits that all the above services can be hosted on the same or multiple machines. As a P2P tracker, CNG Server is responsible to interact with the various P2P modules within the CNG Clients, to organize of the P2P overlays for achieving an efficient live video diffusion. Finally, the P2P tracker hosts all the common information entities that are to be accessed by the peers [12].

3.2. Social Networking and Online Collaboration. CNG Server hosts various community tools that offer online collaboration services to users. The tools are accessible to the users through the GNG Toolbox. Since they are web based, the CNG Client retrieves all the necessary information from the CNG Server in order to provide these applications to the user.

The community tools are based on the Web 2.0 technology and are following the science-driven and interoperable design. The users are able to interact, communicate, and collaborate with each other in a social networking platform. In this architecture, the users are considered to be both consumers and producers of the generated content. As long as the CNG Server acts as a social networking system, it considers four different types of user-to-user relationship:

(1) idol,

(2) fan,

(3) friend,

(4) friend of a friend.

Assuming that player A adds player B to his (A's) friend list means that player A is now a fan of player B and player B is an idol of player A. If player B also adds player A to his (B's) friend list then player A will be friend of player B and player B will be friend of player A. Finally, if player C is a friend of player B, he has just become a friend of a friend of player A. The friend list can be organized by creating various groups of friends.

Community tools provide a wide variety of interactive applications. The members are able to interact and communicate to each other using many available tools, from simple private messaging to live video streaming. The tools are intended to be accessible within the game environment. Taking this into account, they are having minimal style.

The design of the web tools can support and present many forms of user-generated content. Firstly, the Wall application is a community tool in which the users can publish posts in a common thread. The users can view posts made by other users and can interact by replying or republish them to his Facebook and Twitter accounts.

The user can create his personal blog, create custom polls, or upload files. These features are widely used in social networks and are considered to be very essential tools for the members of gaming communities. The proposed design follows the principle that the user-generated content should be exposed only to authorized groups. When uploading a file, creating a poll, and so forth, the user specifies the access restrictions, thus the content may be publicly available or only available to a group of friends.

Voice chat applications are very popular to gamers as long as they support voice conversations besides text-based chat. CNG Server includes a voice chat tool that can be used for real-time communication among the connected members. Every user can create his own chat room providing information about who are authorized to join it.

IM belongs to the real-time text-based communication systems, between users that use any supported devices. As the majority of The Internet users are using IM services, the need of interconnection with such external services arises. Following the users' needs, CNG Server integrates browser-based IM systems that support multiple networks, like Windows Live Messenger [13] and gTalk [14].

4. Design and Architecture

In this section we present the architectural components developed within CNG to provide the online community and collaboration tools, namely, the user interface, the web server, the database, and the chat application. We describe the chat application separately since a large part of its functionality has been implemented to run within the CNG Client and because it is the sole system's module that interacts with the media services of the CNG Server. Figure 2 illustrates an overview of the system design for the provision of the community and collaboration services.

Before presenting the architectural components in more detail we summarize the user needs that led to the CNG system and the design presented in this paper.

4.1. User Needs for Community Activities within MMOGs. The CNG project has conducted a user needs' analysis and an initial research on stakeholders' requirements. The results are available in [15] and are outlined in [16]. This research identifies why players communicate with others, what they share and how, as well as what they find frustrating about using current community related tools. Some of the main points found in this work are summarized below.

(i) The interviewees were frustrated by the use existing community tools, because of reduced immersion, communication difficulties and privacy issues, and poor usability of tools.

(ii) Some of the interviewees have used or were aware of similar systems, low awareness and poor ease of use being the hurdles to adoption.

(iii) Many reported that if the process of using such tools was made easier, they would be more likely to try it.

(iv) The key social/community functions that were particularly popular include video sharing, chat services and in-game web browsing.

(v) The results support the development of a range of social/community function, in game, without the need to minimize the game screen.

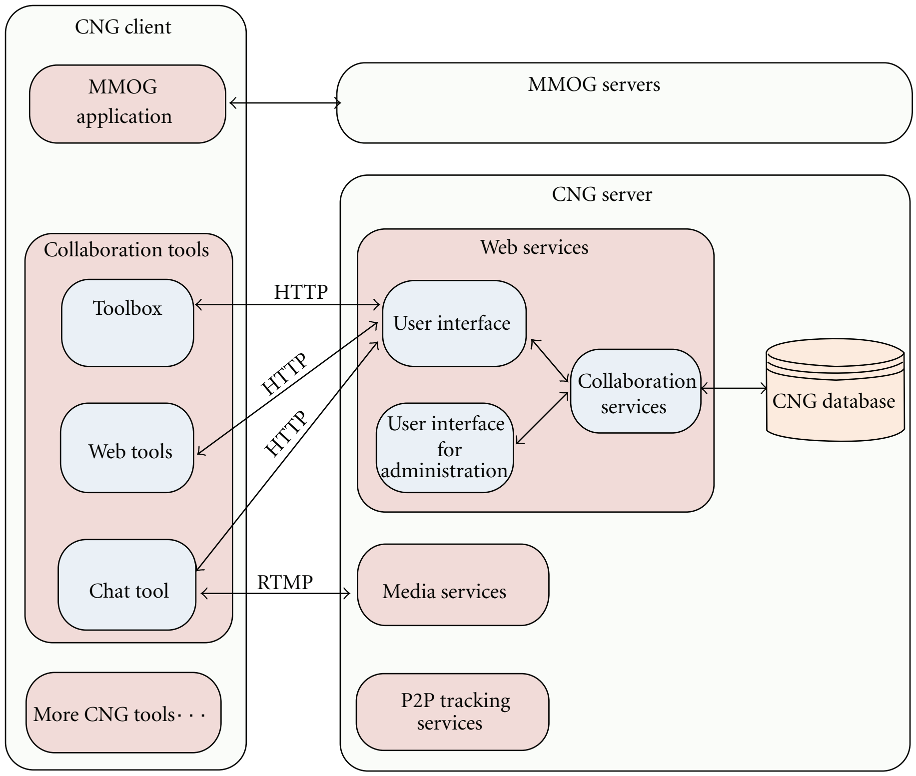


Figure 2: System architecture for the provision of community and collaboration services.

These findings formed the basis for the selection of the CNG tools to support social and community activities from within multiplayer games and the design of the CNG system, detailed in the following sections.

4.2. User Interface. As discussed previously, IGIT technology permits the in-game rendering of web content and thus the CNG online community and collaboration tools are implemented as web pages that are rendered inside the game window and corresponding to various independent widgets that are offered to the players. The graphical design for these widgets follows a simple and clear approach, so as not to dominate over the game scene itself onto the screen and therefore without affecting or distracting the user's in-game experience. Figure 3 depicts some examples of these widgets.

The CNG tools have customizable layout by making use of advanced web technologies. This is to accommodate both the various users' preferences and styles, as well as the different positioning options in the different games since the CNG collaboration tools should not overlap the game's on screen tools and options. The customization options can also be used to offer different layouts for different scenes of the game and/or different views.

Figure 3: Examples of CNG community and collaboration tools' widgets.

Using IGIT, the CNG tools behave as floating windows within the game and thus achieve to appear as part of the game. The importance of this functionality is that it enhances the game experience without any need for the players quitting the game and breaking their in-game experience. It should be noted that positioning the CNG collaboration tools outside the game window, for example, next to right side of the game window, has also been considered but gamers seem to prefer the tools rendered on top of the game window. IGIT permits that the tools' containers can be highly customizable in terms of size, position, and transparency. Two different positioning, transparency setting, and sizing options are depicted in the two screenshots of Figures 4 and 5 where RedBedlam's MMORPG, the so-called "The Missing Ink" [17], is being used within CNG framework. The first screenshot shows the layout of the blogging widget when the user has expanded the widget's height and therefore a vertical alignment of the panes in the widget is more suitable in terms of usability and visibility. The second one shows the "Wall" widget layout in case the user has expanded the widget's width and therefore a horizontal alignment of the panes in the widget is more suitable.

Finally, an important challenge during the design of the CNG User Interface is the support of multiple game resolutions and screen sizes. To this direction, the CNG online community and collaboration tools have been designed in a manner that is independent of the game resolution selected by the user. This adaptation to the game screen configuration is not limited to the size of the CNG tools' windows but also considers their web content. At the initiation of the session, the game screen configuration is read through the IGIT JavaScript library and communicated to the CNG Server. Based on this input, CNG Server generates dynamically the appropriate dimensions of the CNG windows as well as consistent Cascading Style Sheet (CSS) files for the various elements (including Flash objects) consisting the web content, thus, achieving a consistent layout of the CNG tools' windows.

Figure 4: RedBedlam's "The Missing Ink" game with integrated CNG collaboration tools, where widget panes are in vertical alignment.

Figure 5: The same game with integrated CNG collaboration tools, where widget panes are in horizontal alignment.

4.3. Web Server. The web server is the CNG Server's component responsible for the provision of the online community and collaboration services. The web server consists of two components: the UI component, which is the part of the web server that constitutes the presentation-tier, and the backend component, which is the part of the web server that constitutes the business logic-tier. The web server is based on the the Elgg social networking engine [18].

4.3.1. UI Component. The UI Component is the interface of the overall web server towards the web browser instances in the CNG Client side. At this point it should be noted that the client side of the CNG system is executed within a container that is able to display web content on top of the game, using web browser instances. The UI component provides the client side with the CNG user interface and the Web 2.0 online collaboration tools as web content, that is, HTML, JavaScript, and Flash objects. The web browser instances in the CNG Client side can communicate directly with the UI component in order to be provided with these services. With this arrangement it is possible to update and extend the CNG user interface and the CNG Web 2.0 online collaboration tools without changing the CNG Client.

An additional service offered by the UI Component is the provision of the web-based interface that is necessary for the administration of the web server. This interface is available to the users that have privileges as administrators on the CNG system.

4.3.2. Backend Component. The backend component is in the core of the overall web server and it is the mid-tier between the other parts of the system, that is, the UI Component and the Database. The purpose of the backend component is to provide all the functionality that is related to the Web 2.0 community and collaboration applications that the CNG system offers and integrates with the games played. This functionality includes also the management of these web applications, as well as the management of the involved entities, for example, users, access groups, and so forth.

In addition the backend component also offers authentication and authorization services, to the rest of the system that might require it. Although the authorization and authentication is designed as part of the backend component, since the user accounts and the access rights are a key part of the social applications, the pertinent functionality is provided also using a simple and clear interface that other servers and/or modules can use. The backend component stands between all other components and the database, hiding the database schema and database implementation details, from the rest of the system.

4.3.3. External Social Networks. CNG Server has the capability to interact with other existing external social networks. The system's design includes interconnection with Facebook [19] and Twitter [20] and single sign-on functionality. The user has to follow an one-time setup phase in which he enters his Facebook and Twitter credentials and an access token is generated. In this setup phase, the user has to authorize the corresponding CNG application to grant write permissions in his account. The credentials are used only to acquire a permanent token and are not stored in CNG Server.

After this initialization step, the messages that are posted using the community tools can be posted to his Twitter and Facebook walls. This procedure is done at server side, without prompting the user to provide his credentials for logging in to his social networks accounts. The design of the interconnection with the above networks can be considered secure, as long as no credentials are stored in the CNG Server but only the access tokens. These tokens are valid only for usage by the declared network address of CNG Server.

4.4. Chat Application. Chat application can be considered as a real-time communication channel which enables two or more online users to chat using text and voice. It can provide both private and group chat using chat rooms that the users can create. In group chat, all the users on the chat room must be connected to the CNG Server but in the case of private messaging, the messages sent to offline users will be delivered as soon as the users log back in. A notification for new messages is presented to the user when there are any new unread messages and the chat window is enabled but not visible.

We have chosen Adobe Flash [21] technology for the implementation of the CNG web-based chat application. This choice was done by observing a lot of benefits in comparison with other solutions like Java and HTML. The most important are the simplicity of the implementation that Flash offers and the client side compatibility. Flash is a multimedia platform that is used for graphic animations and is able to provide interactivity to web pages. Some of its features include the support of bidirectional multimedia streaming and capturing the users' input devices, including microphone and camera. The Adobe Flash Player is a very popular multimedia and application player supporting Shockwave-Flash (SWF) files. It is integrated as a plugin at most web browsers and is available on many platforms. Its efficiency relies on the fact that it uses vector graphics to minimize file size and create files that save bandwidth and loading time. Finally, the selection of HTML5 was discouraged in the current implementation while it is a new technology that supports audio (and video) playback but not capturing and therefore it cannot adequately cover the CNG chat application functionality.

Chat application was designed to be simple and easy to use in order to not require a lot of resources and degrade the gaming experience. An overview of the intefaces between the involved system modules is depicted in Figure 6. Like every CNG web tool it is applied as a module in CNG Server and it is located at server side. This architecture enables the maintenance of this tool independently from the system. A replacement of this tool with a tool that relies on different technology is also possible without introducing any overhead to CNG Client software.

A media server is used as an application server for the chat services. Flash is accompanied by open-source solutions like the Red5 Media Server [22], which is used for the support of the media streaming functionality. Red5 is an open-source RTMP server written in Java that can be used for text, audio, and video chat applications. It supports audio/video streaming (FLV and MP3), client stream recording (FLV only), and live stream publishing. This server handles the CNG chat application requests by implementing the Real-Time Messaging Protocol (RTMP) protocol for all multimedia transmissions of the chat application.

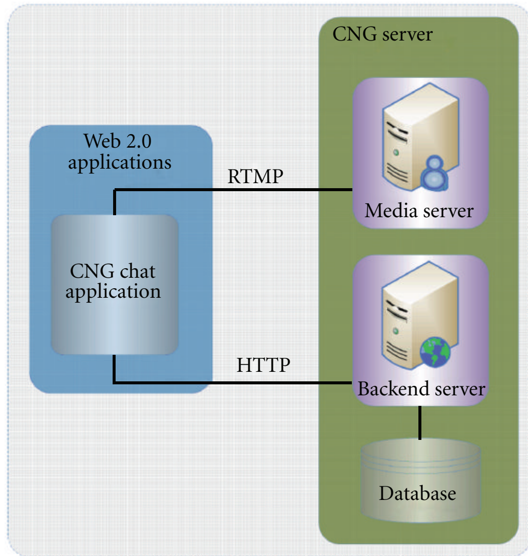


Figure 6: Overview of the interfaces for CNG chat application.

The Flash objects that are provided by the CNG Server are intended to provide web-based text and voice communication in an asynchronous mode in order to implement an interactive web application. The chat application maintains a permanent connection between the user and the media server as long as the user is connected. Thus, the user can retrieve data asynchronously without interfering with the display and behavior of the existing page.

The CNG chat application acts both as transmitter and as receiver at client side. As a transmitter, it is able to capture voice and text messages and transmits them to the media server, while as a receiver it can receive voice streams and messages and reproduce them in the client side. The backend component's role is to be responsible for room management and also for access management in cooperation with the media server. Users have to select from the list of the users joined in the room, which voice streams they wish to receive. They can also send text messages to public (in-room) chat or to individual (private) user. The communication within the chat service is not direct between two users, but the messages and streams are transferred through the media server. Figure 7 presents the operation of the CNG chat application with respect to its interaction with the CNG Server in the form of a sequence diagram.

4.5. Database. The database is part of the CNG Server and constitutes the data-tier of the CNG Server's 3-tier architecture. The main part of the data stored in the database are the data for the CNG community and collaboration tools. As mentioned before, the web server and the CNG tools are based on the Elgg social networking engine [18], and therefore the CNG database follows the schema required by Elgg.

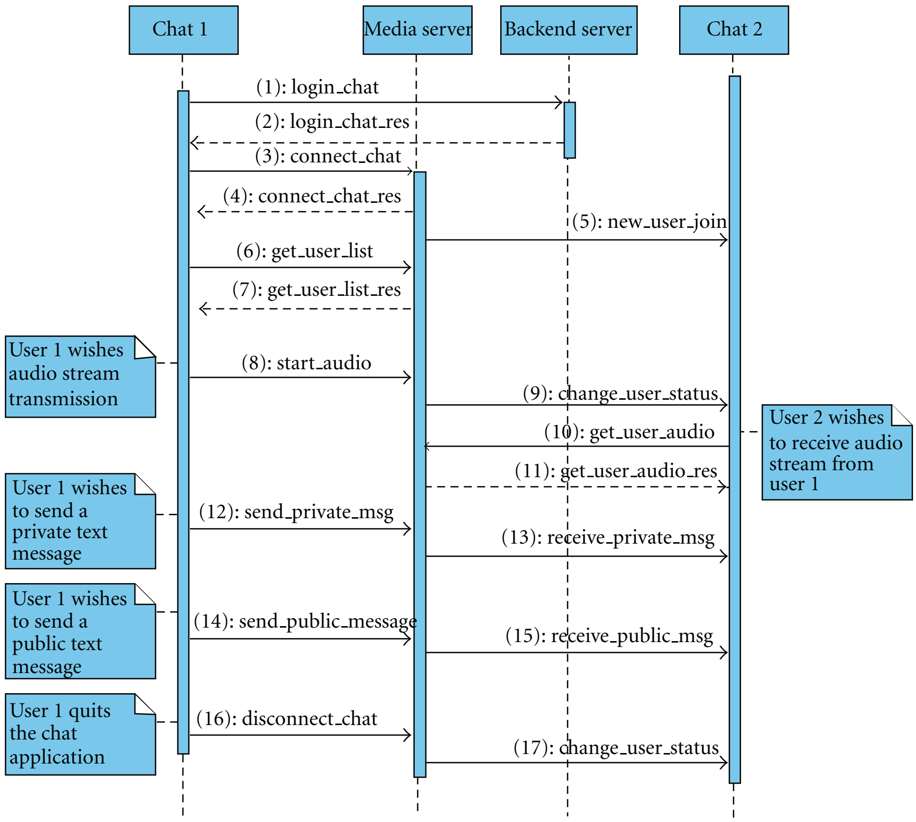


Figure 7: Sequence diagram presenting the operation of the CNG chat application with respect to its interaction with the CNG Server.

In order to be extensible, this schema models the entities and the entity relationships in general and allows specific entities and relationships that are inherited from the general entities and relationships. The schema for the "Entities" and the "Entities_Relationships" tables is shown in Figure 8. The main entities used are the User_Entities, The Group_Entities, and the Object_Entities. The "User_Entities" table and the "Object_Entities" table model the CNG system users and the user groups defined in the CNG system, respectively. The "Object_Entities" table models the social network objects like the video streams, the blog posts, the uploaded files, and the bookmarks. Figure 9 shows the part of the schema for these main entities.

Further to these entity types, the nature of an entity can be further specified through the definition of subtypes. For example, an entry in the Entities that is also related to an entry in the "Object_Entities" table and is related with the subtype entry "chatRoom" is interpreted as an entry for a chat-room object. Similar approach is followed for all the other community and collaboration objects. The subtypes are mainly used for the objects entities, for example, wall posts or video streams, but can also be used for all the other types of entities. The subtypes are modeled by the "Entity_Subtypes" table, which is related with the "Entities" table, as shown in Figure 10.

Moreover, additional information can be added to entities in two ways. The first way is the addition of metadata, which is information that can be added to an entity to describe it more precisely. For example, tags, an ISBN number, or a file would fall under metadata. The second way is the addition of annotations, which are generally information added by third parties. For example, comments and ratings are both annotations. Metadata and annotations are modeled by the "Metadata" and "Annotations" with an additional "Metastring" table to store the actual stings of extra information, which is linked to the entities through the "Metadata" and "Annotations" tables, as shown in Figure 11.

Finally for granular access control every entity, annotation, and piece of metadata is related to an entry in the "Access_Groups" table that models the access rights and is related through the "Access_Group_Memberships" table to the "Entities" table, thus signifying which entities, for example, users, have access to the entity at hand, for example, a live video stream. The relevant part of the schema is shown in Figure 12.

This flexible design of the database schema provides extensibility to the system, in terms of the possibility for having additional social objects, additional relationships between the objects, additional information to existing objects, and so forth.

4.6. Administrative and Monitoring Tools. As the CNG Server provides online tools to many users at the same time, it is important to monitor the system's performance and also the online tools usage using administrative and monitoring tools. The users are free to interact with the online community using the CNG tools and it is always possible to generate

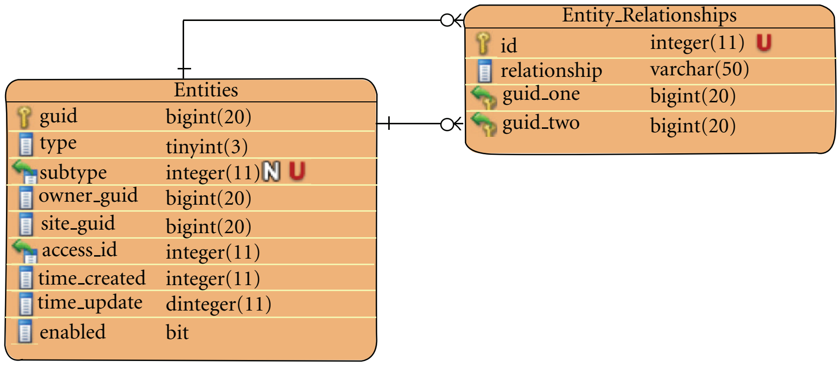


Figure 8: The tables "Entities" and "Entity_Relationships."

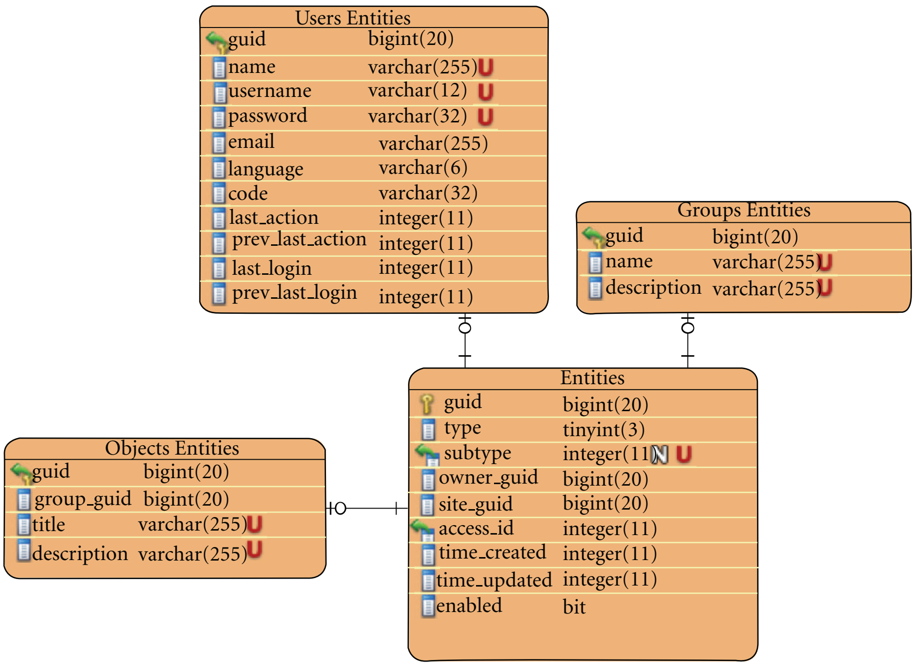


Figure 9: The tables "Users_Entities," "Groups_Entities," and "Object_Entities" as well as their relationship with the "Entities."

inappropriate content. In these cases, the administrative tools are used in order to supervise the CNG community network and avoid the distribution of content that violates the terms of service. The users can report any content that they feel that breaches standards on social, religious, cultural, or other grounds. Then, the administrators are finally able to moderate and review any reported user or content.

Administrative tools include an interface for the administrators of the CNG system to manage the CNG system using network administration tools and a control panel for users' accounts administration. The administrative procedures provide statistics like the number of registered users, groups, existing polls, issued blog posts, wall posts, uploaded files, shared streams, and information about currently online users. The administration panel provides full access to users of the system. It supports operations for adding, removing, activating, and deactivating a user and also resetting passwords and promoting users as system administrators. A common usage of the users' administration tool is when a user generates some appropriate content and should be banned from the administrator.

CNG Server can be configured by setting its basic properties like the URL of the website for the provision of the Web 2.0 collaboration tools, along with its name and description. The administrator can also set the sites email address, the sites default language, and whether secure access through HTTPS is used or not. Apart from global configuration settings, the administrator can activate or deactivate any tools and also configure their parameters. The CNG community network is designed to be modular and can be considered as a compilation of different online tools. That means that the community network can be totally changed by enabling or disabling some tools. With this functionality the collaboration tools could be easily extended and maintained without affecting the whole system. Furthermore, as the administrator is able to configure the tools' parameters it

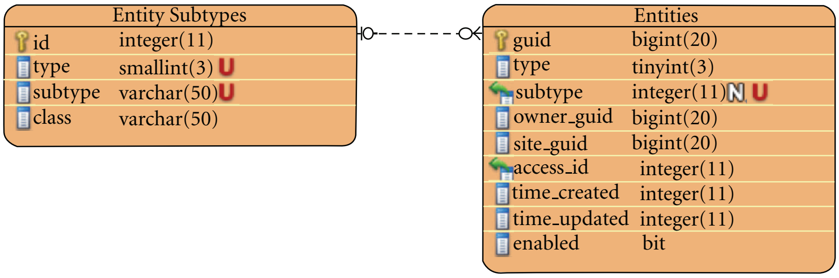


Figure 10: The "Entity_Subtypes" table.

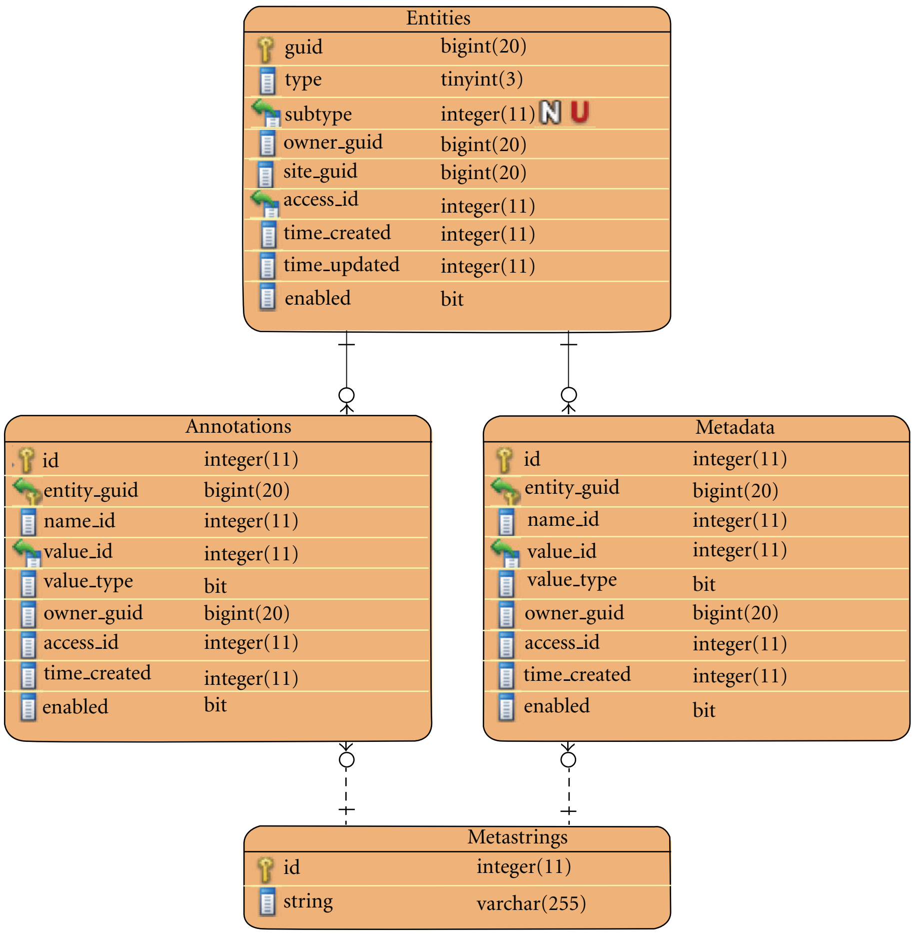


Figure 11: The tables "Annotations," "Metadata," and "Metastrings".

means that the configuration of each tool may be changed online at anytime by the administrators without interrupting the other tools.

The monitoring tools log the users' activity within the CNG tools and intend to provide information on the users' behavior in the form of detailed logs as well as through visual aggregated statistics. The implementation of the tools has been based on the Elgg [18] framework's infrastructure. These tools are used in combination with the other online verification tools in order to provide a complete view of the systems behavior. CNG Server provides a system diagnostics tool that generates a text file with all the configuration information of the system. This information includes all the paths to the files installed in the web server and all the installed collaboration and community tools and any necessary related details. Moreover, the diagnostics tool provides information on the global parameters of the system, such as the Server's environment, IP, URL, and connected database.

Apart from generic diagnostic tools, CNG Server includes monitoring tools that are related to specific collaboration tools. The tool for live video streaming is considered as one of the most important and most frequently used tool of CNG system. The monitoring of live video streams

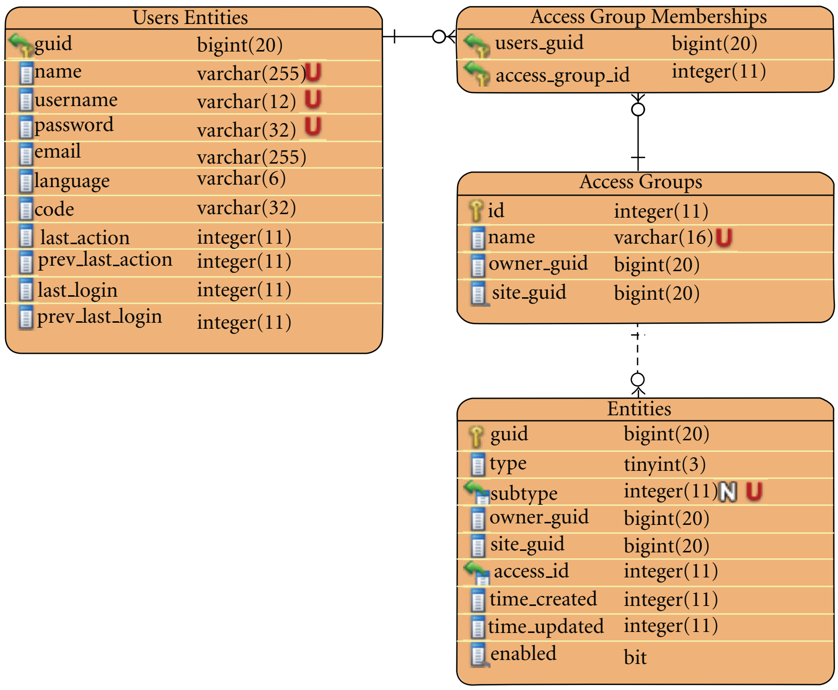


Figure 12: The "Access_Groups" and "Access_Group_Membership" tables for the application of access controls to entities.

requires a dedicated monitoring tool that will be able to provide detailed information. The implementation of this monitoring tool is illustrated in Figure 13.

The monitoring tools intend to analyze the traffic of the CNG tools usage at the highest possible level. The main functionality relies on collecting real data and analyzing them in order to extract knowledge about the usage of CNG community and collaboration tools. It is obvious that the usage of packet capturing (sniffing) software, for example, Wireshark, is discouraged in this case, since the desired information exists at the application layer. On the other hand, the existing common web analytics tools (like Google Analytics [23]) could be used for monitoring the CNG system usage but in a generalized way. They are not able to depict the CNG tools usage because they have no information about CNG tools or users. Thus, the existing common web analytics are not suitable for such in-depth analysis. The implemented web-monitoring tools provide this type of CNG tools analysis and of course can be used for extracting generalized results for the system usage.

These web-monitoring tools are integrated to the CNG Server and are fully accessible from the administration panel of the CNG Server (of course for the CNG Server's super-users). They exploit the existing technology for the Web 2.0 tools. The logged information is stored in the MySQL database of the system and the results can be viewed from the existing CNG web interface. Moreover, the "Google Chart Tools" [24], which is a state-of-art web-technology for data visualization, is used for rendering charts at client side using JavaScript.

The logging is executed at every user HTTP request towards the CNG Server. The system parses the requested URL and then finds the referring tool. Furthermore, it utilizes the same URL and the HTTP request method (GET or POST) in order to identify the action of the user within the CNG tool. The results of the web-monitoring process can lead to a variety of conclusions. The collected information reflects the activity of the user on the CNG tools. The results can be organized in various ways based on the choices of the system's verification engineer or administrator. A filtering procedure that organizes the results (a) per user, (b) per tool, and (c) per action has been implemented for this purpose. This filtering procedure is depicted at the following flowchart.

The results are real time presented to the administrator and they are available at three formats: tables, pie charts, and raw logs. Each table shows the results in two columns. The first column refers to the object that is monitored and the second column refers to the hits. At pie chart, the objects are coloured and labelled. This representation is used for visualizing the table content, thus the administrator is able to quickly identify which objects are popular.

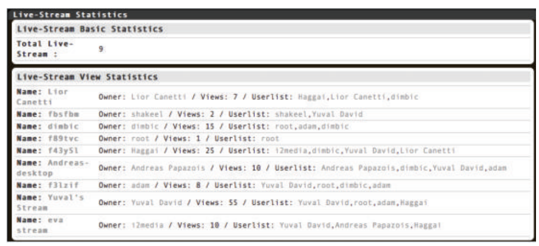
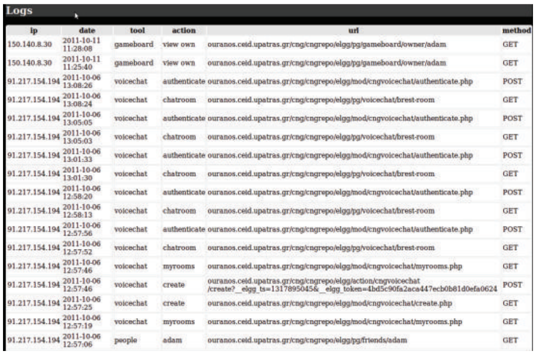


Figure 13: Live video streams' monitoring on the web server.



Logs

ip	date	tool	action	url	method
150.140.8.30	2011-10-11 11:28:08	gameboard	view own	ouranos.ceid.upatras.gr/cng/cngrepo/elgg/pg/gameboard/owner/adam	GET
150.140.8.30	2011-10-11 11:25:40	gameboard	view own	ouranos.ceid.upatras.gr/cng/cngrepo/elgg/pg/gameboard/owner/adam	GET
91.217.154.194	2011-10-06 13:08:26	voicechat	authenticate	ouranos.ceid.upatras.gr/cng/cngrepo/elgg/mod/cngvoicechat/authenticate.php	POST
91.217.154.194	2011-10-06 13:08:24	voicechat	chatroom	ouranos.ceid.upatras.gr/cng/cngrepo/elgg/pg/voicechat/brest-room	GET
91.217.154.194	2011-10-06 13:05:05	voicechat	authenticate	ouranos.ceid.upatras.gr/cng/cngrepo/elgg/mod/cngvoicechat/authenticate.php	POST
91.217.154.194	2011-10-06 13:05:03	voicechat	chatroom	ouranos.ceid.upatras.gr/cng/cngrepo/elgg/pg/voicechat/brest-room	GET
91.217.154.194	2011-10-06 13:01:33	voicechat	authenticate	ouranos.ceid.upatras.gr/cng/cngrepo/elgg/mod/cngvoicechat/authenticate.php	POST
91.217.154.194	2011-10-06 13:01:30	voicechat	chatroom	ouranos.ceid.upatras.gr/cng/cngrepo/elgg/pg/voicechat/brest-room	GET
91.217.154.194	2011-10-06 12:58:20	voicechat	authenticate	ouranos.ceid.upatras.gr/cng/cngrepo/elgg/mod/cngvoicechat/authenticate.php	POST
91.217.154.194	2011-10-06 12:58:13	voicechat	chatroom	ouranos.ceid.upatras.gr/cng/cngrepo/elgg/pg/voicechat/brest-room	GET
91.217.154.194	2011-10-06 12:57:56	voicechat	authenticate	ouranos.ceid.upatras.gr/cng/cngrepo/elgg/mod/cngvoicechat/authenticate.php	POST
91.217.154.194	2011-10-06 12:57:52	voicechat	chatroom	ouranos.ceid.upatras.gr/cng/cngrepo/elgg/pg/voicechat/brest-room	GET
91.217.154.194	2011-10-06 12:57:46	voicechat	myrooms	ouranos.ceid.upatras.gr/cng/cngrepo/elgg/mod/cngvoicechat/myrooms.php	GET
91.217.154.194	2011-10-06 12:57:46	voicechat	create	ouranos.ceid.upatras.gr/cng/cngrepo/elgg/action/cngvoicechat /create?__elgg_ts=1317895045&__elgg_token=4bd5c90fa2aca447ecb0b81d0efa0624	POST
91.217.154.194	2011-10-06 12:57:25	voicechat	create	ouranos.ceid.upatras.gr/cng/cngrepo/elgg/mod/cngvoicechat/create.php	GET
91.217.154.194	2011-10-06 12:57:19	voicechat	myrooms	ouranos.ceid.upatras.gr/cng/cngrepo/elgg/mod/cngvoicechat/myrooms.php	GET
91.217.154.194	2011-10-06 12:57:06	people	adam	ouranos.ceid.upatras.gr/cng/cngrepo/elgg/pg/friends/adam	GET

Figure 14: Raw logs for a specific user's activity.

Raw logs are presented in a table with the columns: IP, Date, Tool, Action, URL, Method. The rows represent the users' interaction on the CNG tools. This output format can be used for in-depth monitoring of activity within a specific tool or of a specific user. Next, Figure 14 displays the logs of a specific user.

When a tool filtering is applied, then an action analysis for the selected tool is presented. The verification engineer/administrator can observe which tools are popular and which actions are popular for a specific tool. For example, Figure 15 shows the action monitoring for the "Blog" tool. Finally, it is worth mentioning that special care has been given in order for all the above tools to be automatically extended in case additional tools/user activities are added in the CNG Server.

5. Internal Testing

The prototype of the CNG system including the CNG Server and the collaboration tools implemented, based on the presented design, has undergone a testing phase, called internal testing or closed alpha testing. During this phase the CNG system was used together with the TMI on game, by a small number of users, that are members of the CNG consortium and students. Although the purpose of the internal testing was to verify the correct operation of the CNG system and identify problems and bugs to be fixed, this testing gave some initial insights into the usability of the system and its expected acceptance.

The testers found that the CNG system did not affect the game's responsiveness and that the CNG tools work well. The testers appreciated the ability to change the size of the CNG widgets and rearrange them, as to minimize interference with the in-game experience. They also appreciated the ability to change the opacity of the CNG widgets and render them at a configurable level of transparency.

However, it should be mentioned that this internal testing procedure was not meant as a large-scale verification, and the tester demographics may not be representative of the target group's (the players of online games) demographics. Therefore a more complete picture on how the users experience CNG will be available after the completion of the large scale CNG verification.

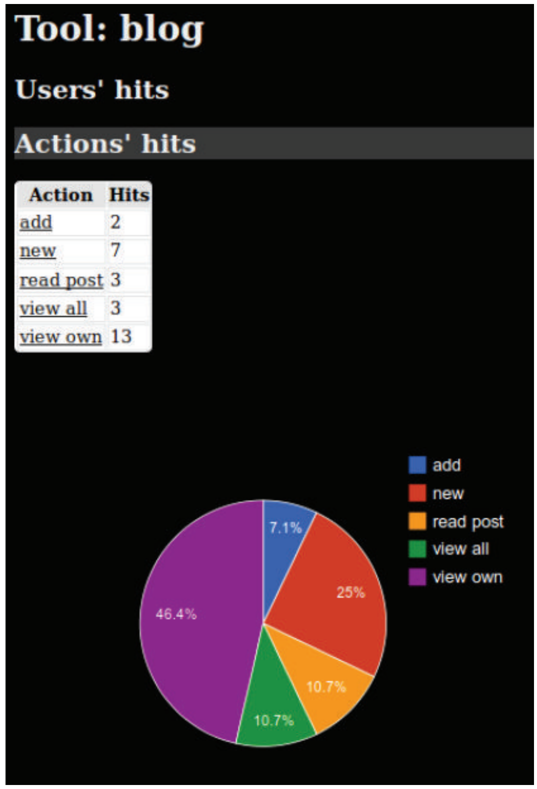


Figure 15: Reporting the actions usage for the "Groups" tool.

6. Conclusions

In this paper, we presented the system developed in the context of the CNG project in order to provide online community and collaboration services to MMOG players. After a brief introduction of the CNG system and its key features, which are, namely, the IGIT technology and the live video streaming over P2P, we described the CNG system's high-level architectural design. Then we focused on the system's design and implementation for the provision the online community and collaboration tools. We presented a detailed description of the CNG system's core element, that is, the CNG Server as well as the relevant aspects of the CNG Client. Our analysis included details on the user and administration interfaces, the element's structure, and the functionality for both the CNG Server and Client. All the relevant architectural components were described including the web server, the CNG database, as well as the client-side Web 2.0 components with respect mainly to the CNG chat. It should be noted that, during the presentation of the above aspects, we highlighted all the innovative design and implementation features that distinguish CNG collaboration tools from a typical social networking website.

7. Future Work

The next step of this work is the evaluation of the design and the CNG system, during the CNG verification phase. Tests will be conducted, with the engagement of online players, for an extended period of time (approximately 2 months). During this process, online users will be led to use the TMI online game both without and with the CNG system. The assessment of the CNG system will be performed through the collection of feedback from the users through online questionnaires and with the aid of the implemented verification and monitoring tools.

As the design of the CNG system is based on real user needs and is influenced by the user preferences, it is expected that the verification phase will confirm the system's usability, and the players' intention to use it in order to enhance their in-game experience. It is also expected that the verification process will produce several comments for enhancing and fine-tuning the CNG system, making it more appealing to the users.

In addition, since the CNG collaboration tools extensively use and integrate open or de facto WWW standards, several possible contributions to the standardization processes have been identified so far. The first possible contribution could be to the specification of the CSS. CNG system has been designed to support multiple MMOG resolutions and the collaboration tools have to be rendered consistently in all resolutions, therefore it is required to render the content in a resolution-independent manner. Anther possible contribution of CNG to WWW standards could be a set of specifications for audio and video capturing and visual effects support within HTML. Finally, a new set of social relations, typically used within games, as well as support for user-generated multimedia content over online communities, can be contributed to open social networking standards.

Acknowledgments

The authors wish to thank Kerry Fraser-Robinson from RedBedlam for his ideas on the in-game social terminology, Jonathan Freeman and Eva Ferrari from i2 media research for their recommendations on the users' needs, as well as all the partners belonging to the CNG project consortium for their collaboration. 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] S. Ahmad, C. Bouras, R. Hamzaoui et al., "The community network game project: enriching online gamers experience with user generated content," in *Proceedings of the 2nd International Conference Creative Content Technologies (CONTENT '10)*, Lisbon, Portugal, 2010.

[2] CNG project—the community network game, 2011, http://www.cng-project.eu/.

[3] Steam, the ultimate online game platform, http://store.steampowered.com/.

[4] Xfire gaming simplified, http://www.xfire.com/.

[5] S. Bergsträber, T. Hildebrandt, C. Rensing, and R. Steinmetz, "Virtual context based services for multiplayer online games to facilitate community participation," *Multimedia Tools and Applications*, vol. 45, p. 347, 2009.

[6] S. Shen and A. Iosup, "The xfire online meta-gaming network: observation and highlevel analysis," in *Proceedings of the International Symposium on Audio-Visual Environments and Games (HAVE '11)*, Qinhuangdao, China, 2011.

[7] The CNG Consortium, "D2.10: soa and guideline for technology, ugc, and collaboration tools," Tech. Rep., CNG Project, 2011, http://www.cng-project.eu/?page_id=59.

[8] S. Ahmad, C. Bouras, R. Hamzaoui et al., "The community network game project: enhancing collaborative activities in online games," in *Networked & Electronic Media Summit (NEM Summit '10)*, Barcelona, Spain, 2010.

[9] Playexpert, home of kochava, gamerdna, and trueoverlay technology, http://www.playxpert.com/.

[10] Overwolf adds facebook, skype, video recording, messenger and much more to you favorite games!, http://www.overwolf.com/.

[11] Raptr: What are you playing?, http://raptr.com/.

[12] The CNG Consortium, "D3.1: system architecture and detailed design of games adaptation, 3d and ugc," Tech. Rep., CNG Project, 2011.

[13] Windows live messenger im service, http://messenger.live.com/.

[14] Google talk, http://www.google.com/talk/start.html.

[15] The CNG Consortium, "D2.1: users and uses of games and community activities," Tech. Rep., CNG Project, 2010.

[16] E. Ferrari, J. Lessiter, and J. Freeman, "Users and uses of multiplayer games and community activities," in *Proceedings of the Networked & Electronic Media Summit (NEM Summit '11)*, Torino, Italy, 2011.

[17] The missing ink, http://www.missing-ink.com/.

[18] Elgg—open source social networking engine, http://elgg.org/.

[19] Facebook social networking service, http://www.facebook.com/.

[20] Twitter social networking service,http://www.twitter.com/.

[21] "Adobe ash multimedia platform software," http://blogs.adobe.com/flashplatform/.

[22] Red5 media server, http://www.red5.org/.

[23] Google analytics, http://www.google.com/analytics/.

[24] Google chart tools, http://code.google.com/apis/chart.

Level-based peer-to-peer live streaming with rateless codes

Eliya Buyukkaya* Shakeel Ahmad† Muneeb Dawood† Jiayi Liu* Fen Zhou‡ Raouf Hamzaoui† Gwendal Simon*
*Institut Mines-Télécom - Télécom Bretagne, France, {firstname.lastname}@telecom-bretagne.eu
†De Montfort University, United Kingdom, {sahmad, mdawood, rhamzaoui}@dmu.ac.uk
‡LIA / CERI, University of Avignon, France, {fen.zhou}@univ-avignon.fr

***Abstract*—We propose a peer-to-peer system for streaming user-generated live video. Peers are arranged in levels so that video is delivered at about the same time to all peers in the same level, and peers in a higher level watch the video before those in a lower level. We encode the video bitstream with rateless codes and use trees to transmit the encoded symbols. Trees are constructed to minimize the transmission rate for the source while maximizing the number of served peers and guaranteeing on-time delivery and reliability at the peers. We formulate this objective as a height bounded spanning forest problem with nodal capacity constraint and compute a solution using a heuristic polynomial-time algorithm. We conduct ns-2 simulations to study the trade-off between used bandwidth and video quality for various packet loss rates and link latencies.**

I. Introduction

The last five years have seen the emergence of online video platforms that enable users to broadcast their own live video [1]–[3]. This user-generated content typically consists of reality TV, shows, concerts, or, as in Xfire [4] and TwitchTV [5], live streams from online gamers.

A major problem faced by these platforms is that not all user-generated content is popular. Consequently, thousands of live video streams need to be simultaneously streamed to only a small number of clients. This poses a serious challenge to the existing content delivery infrastructures, which have not been designed for that purpose. An alternative solution is to use a peer-to-peer (P2P) system.

This paper describes a P2P system developed within the CNG project [6]. The goal is to enable Massively Multiplayer Online Game (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. In this context, the video source is any casual gamer, whose network and computer resources are used for the game. *It is therefore critical to minimize the transmission rate at the source.*

In addition, from discussions with stakeholders in the Internet TV and gaming communities, we have identified another requirement, which has not been addressed in previous P2P works. The idea is that peers are arranged in *levels* such that live video is delivered at the same time to all peers in the same level. Moreover, peers in a higher level should be able to watch the video before those in a lower level. This level-based model is a generic way to implement a number of practical features:

- in a tiered or freemium service (two popular business models where users are charged according to the quality of service received), a level is a priority class. Users in the same class have the same playback lag. Moreover, users in a higher class have a lower playback lag than those in a lower class.
- in MMOGs, a level can be defined as a virtual area within the game. Players that are located in one particular area watch a live stream of another player at the same time, which ensures fairness and consistency. Moreover, the playback lag for a player increases with increasing distance to the source of the video.
- in eSport systems (broadcast of live sport events), a level is associated with physical areas. Two people watching the same event and close enough to *physically* interfere have the same playback lag. This aims to prevent the annoying situation where viewers react to an event (e.g., shouting "goal" in a football match) before it has occurred

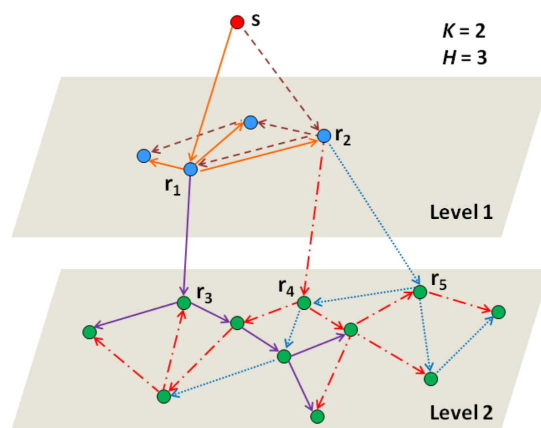


Fig. 1: Video diffusion within and across levels via multicast trees. Example where five trees are used: two in Level 1 (higher level) rooted at r_1 and r_2 and three in Level 2 (lower level) rooted at r_3, r_4, and r_5. A node must receive $K = 2$ packets to decode the video. A packet cannot be forwarded more than three times in a tree ($H = 3$).

on the screen of their neighbors.

We design a multi-level P2P system, which consists of peers and a P2P server. The P2P server has persistent communication with the peers and is responsible for building the P2P *overlay*. *Multicast trees* are used to diffuse video data within and across levels (Fig. 1). We use rateless codes [7] to provide resilience against packet loss. Rateless codes are probabilistic erasure codes that can recover k information symbols from any received $k(1+\epsilon)$ encoded symbols with high probability. Here ϵ is a small positive number that gives the trade-off between the error recovery property of the code and the amount of redundancy it introduces. 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.

The use of rateless codes for P2P video streaming was introduced by Wu and Li [8]. As soon as a receiving peer successfully decodes the source block, it becomes a source and applies rateless coding on the decoded source block to generate encoded symbols for other peers. Grangetto, Gaeta, and Sereno [9] improved this seminal work by introducing a method, called 'Relay and Encode' (RE), where a receiving peer *immediately* relays the received encoded symbols. The authors show that RE has a lower delay than the method of [8]. In [10], they introduced feedback so that peers can stop receiving symbols when a block is decoded.

We introduce two new contributions to P2P live video systems: we design a new multi-level architecture, which aims at guaranteeing that all peers within the same level play the video at the same time, and 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 for the source while maximizing the number of served peers and guaranteeing on-time delivery and reliability at the peers. We formulate this objective as a height bounded spanning forest problem with nodal capacity constraint and compute a solution in polynomial-time.

The remainder of the paper is as follows. In §II, we describe the video transmission scheme. In §III, we present our tree construction algorithm. In §IV, we use ns-2 simulations to test our P2P system. Finally, we give our conclusions in §V.

II. Transmission scheme

Initialization. A peer wishing to broadcast its video requests the P2P server to advertise it. This peer is called the *source*. If another peer is interested in the advertised video, it sends a request to the P2P server. Like a *tracker* (a server of *peerlists* in traditional P2P systems), the P2P server is in charge of updating the overlay information and informing the participating peers. The overlay information consists of peer assignments to levels and sets of multicast trees for each level (see §III). The overall overlay is denoted by $G(V,E)$ where V is the set of peers and E is the set of links. An arc between u and v is denoted by (u,v).

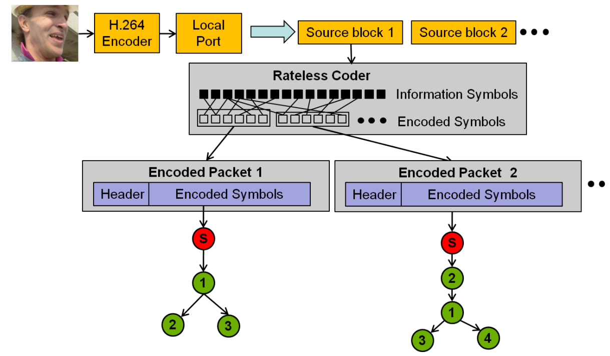


Fig. 2: Video coding and streaming. Node s denotes the source. Packet 1 is sent over a multicast tree to peers 1, 2, and 3. Packet 2 is sent to peers 2, 1, 3, and 4.

Video and channel coding. As soon as the source receives the overlay information from the P2P server, it captures the video and compresses it with the H.264 video coder [11]. 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 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 with a uniform inter-departure time, and one packet is sent on each multicast tree (Fig. 2).

Inter-level communication. A root of a Level 1 multicast tree (r_1 and r_2 in Fig. 1) 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. 1). 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). To reduce the probability that a Level 2 peer receives duplicate packets, Level 1 root peers use randomly chosen rateless code *seeds* when they encode a source block. The value of the seed is sent as part of the packet header. The number of packets sent 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 procedure described above for two levels is repeated for the next levels.

Level-Aware Timely Video Delivery. To ensure that all peers in the same level have the same playback lag, and peers in a higher level have a shorter playback lag than those in a lower one, the following procedure is followed.

All peers are synchronized in time. This can be achieved, for example, with the Network Time Protocol [12]. A time stamp is inserted in each UDP packet to indicate the start time of the current source block. All Level 1 peers play back the first GOP at time $2\Delta+D_{\max}$. Here $D_{\max}=(H+1)\times l_{\max}$, where H is the maximum height of a multicast tree and $l_{\max}$ is an estimation of the maximum latency between two nodes in the overlay. Thus $2\Delta+D_{\max}$ is the latest possible arrival time

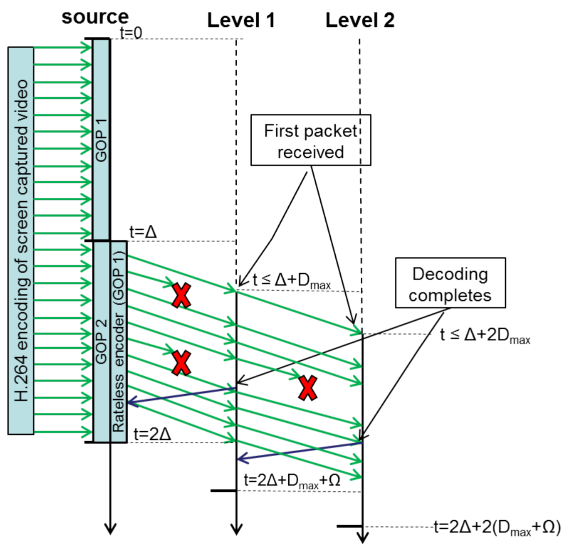


Fig. 3: Transmission strategy.

for any packet from the first source block (Fig. 3).

When a Level 1 root peer completes the decoding of the first source block, it enters the re-encoding phase up to time $2\Delta + D_{\max} + \Omega$. Here, Ω gives sufficient time to send re-encoded packets in case the decoding of the source block is delayed. As the packet loss rate increases, the decoding completion time shifts towards $2\Delta + D_{\max}$ and more time is needed for sending re-encoded packets. We used $\Omega = p_{\max}\Delta$ where $p_{\max}$ is an estimation of the maximum packet loss rate between two nodes in the overlay.

All Level 2 peers play back the first GOP at time $2\Delta + 2D_{\max} + \Omega$, which is the latest possible arrival time of any packet for the first source block. More generally, all peers at Level L play back the kth GOP at time $(k+1)\Delta + LD_{\max} + (L-1)\Omega$. Thus, the playback lag for level L is $2\Delta + LD_{\max} + (L-1)\Omega$.

III. Formulation of the video diffusion problem

In §III-A, we formulate the problem of constructing an optimal forest for the diffusion of source blocks in *one* level. In §III-B, we explain the inter-level interactions in a multi-level overlay. We detail our forest construction algorithm in §III-C.

A. Intra-Level Multicast Trees

For simplicity, we consider only one level, so our goal is to deliver GOPs from a source s to a set of peers. The main idea is to use a multiple-tree structure for video diffusion. Each multicast tree (or *tree*, in short) is used to transmit *one* packet of encoded symbols from the source to the peers. To recover the video block, a peer should receive at least K packets of encoded symbols. In other words, a peer should belong to at least K trees.

There are constraints on the trees. A critical one is that every peer $v \in V$ is associated with an *upload capacity*, denoted by c_v, which is the number of packets the node v can transmit. This capacity constraint limits the number of children a node can have. In addition, to guarantee on-time delivery and reliability at the peers, the end-to-end delay and the packet loss rate on the path from the source to each peer should be bounded. This requirement thus imposes a bound on the height of each tree.

Our objective is to minimize the transmission rate for the source while guaranteeing recovery of the video block. The more trees are used to ensure that all peers receive at least K packets, the higher is the source transmission rate. Thus, **minimizing the source transmission rate is equivalent to minimizing the number of trees**. In light of this, we formulate the video diffusion problem in the overlay as a *Height-Bounded Spanning Forest problem with Capacity constraint* (HBSFC). The goal of HBSFC is to find a forest F with minimum cardinality such that

- the number of trees is limited by source capacity c_s.
- each tree is rooted on a peer, which directly receives a packet from the source s. In other words, we distinguish the source from the *root* of each tree.
- for each node v, the sum of its out-degrees in all trees of the forest is not greater than its capacity c_v.
- each node is spanned in at least K trees. A node that cannot be spanned in K trees is rejected from the overlay.
- the height of each tree is limited by a bound H.

These constraints are imposed on every level. Before going into the details of the algorithm (see §III-C), we discuss inter-level capacity issues.

B. Multi-level overlay management

When the number of levels is greater than one, we have to make sure that the video can be relayed from one level to the next. We create *inter-level connections* for that purpose.

To simplify the management of the whole system, we propose that the root peers in a level are in charge of transmitting data to the next level. Simply put, a root peer immediately forwards a packet it received directly from the previous level to the next level. In other words, a root peer in level l acts as a source for some root peers in level $l+1$. This strategy has three advantages: (i) there is no delay between the reception of a packet and its transmission to the next level, (ii) the inter-level connections are well distributed over peers because root peers are well distributed over the population, and (iii) the management of inter-level connections is easy for the P2P server: it only has to inform the root peers of every level about the trees in the next level.

We denote by R_l the set of Level l root peers. If there is no capacity issue, each root peer in level l should become a source of $\left\lfloor \frac{|R_{l+1}|}{|R_l|} \right\rfloor$ trees in level $l+1$. The remaining trees are randomly allocated to the root peers.

Fig. 1 gives an example of a two-level overlay. Each peer in a level is spanned in two different trees, with height no more than three. Peers r_1 and r_2 are the root peers in level 1, while r_3, r_4 and r_5 are the root peers in level 2, i.e., $R_1 = \{r_1, r_2\}, R_2 = \{r_3, r_4, r_5\}$. Therefore, $|R_1|$ is equal to 2, while $|R_2|$ is equal to 3. Peers r_1 and r_2 are the sources

Algorithm 1 Resource Allocation and Overlay Algorithm

Input: Complete graph $G(V,E)$, a decomposition of V into L disjoints subsets $\{V_1,\ldots,V_L\}$, capacity c_v for all $v \in V$, and the number of packets K

Output: Forest F_l in each level l

1: $|F_1| \leftarrow c_s$
2: **for** l from 1 to $L-1$ **do**
3: $\quad C^{next} \leftarrow K \times |V_{l+1}|$
4: $\quad$ build forest F_l with C^{next} and V_l
5: $\quad$ **while** forest F_l does not cover all peers in V_l **do**
6: $\quad\quad$ **if** $C^{next} > K$ **then**
7: $\quad\quad\quad C^{next} \leftarrow C^{next} - 1$
8: $\quad\quad$ **else if** $\sum_{v \in V_l} c_v > K$ **then**
9: $\quad\quad\quad$ reject $v \in V_l$ with minimum capacity
10: $\quad\quad$ **else**
11: $\quad\quad\quad$ sacrifice F_l
12: $\quad\quad\quad$ re-build V_l such that $C^{next} = \sum_{v \in V_l} c_v \geq K$
13: $\quad\quad\quad$ break
14: $\quad\quad$ **end if**
15: $\quad\quad$ build forest F_l with C^{next} and V_l
16: $\quad$ **end while**
17: $\quad |F_{l+1}| \leftarrow C^{next}$
18: **end for**
19: build forest F_L with $C^{next} = 0$ and V_L

for trees in level 2. Peers r_3 and r_4 are connected to r_1 and r_2 respectively. To be the source of r_5, r_2 is then randomly selected between r_1 and r_2.

The management of peer capacity is a critical issue for the inter-level links. We explain how we solve this issue hereafter.

A root peer should reserve some upload capacity to serve some peers in the next level. For any peer v, we distinguish the upload capacity c_v^{cur} that can be used to serve peers in the same level from the upload capacity c_v^{next} that is secured to serve peers in the next level. Of course, $c_v^{cur} + c_v^{next} \leq c_v$ for any peer v, and $c_v^{next} = 0$ if v is not a root peer or is a root peer at the lowest level.

The number of trees that are constructed in a level l depends on the amount of resources that have been secured by the root peers in the previous level. Formally, if we denote by F_l the forest in level l, we have: $|F_l| = \sum_{v \in V_{l-1}} c_v^{next}$.

Once the number of trees for level l is given, we should both build the trees and determine the amount of capacities that must be secured for the next level $l+1$. On the one hand, we aim to maximize the number of peers that are covered in level l, i.e., the number of peers that are spanned in at least K trees of F_l. *This objective calls for a high* $\sum_{v \in V_l} c_v^{cur}$. On the other hand, we have to guarantee that we reserve enough capacities for the next level. *This objective calls for a high* $\sum_{v \in V_l} c_v^{next}$. The two objectives conflict.

We propose a heuristic (see Alg.1), which is based on the following observation. The number of trees that should be constructed in order to serve all peers in a level is bounded:

$$K \leq |F_l| \leq K \times |V_l|$$

We suggest the following for resource allocation at level l. We start with the most optimistic scenario where $\sum_{v \in V_l} c_v^{next} = K \times |V_l|$. We try to build a forest F_l with the remaining capacities. If all peers in V_l are covered at least K times in F_l, then both objectives are fulfilled and it ends the algorithm for level l. Otherwise, we have to revise the allocation of resources:

1) we reduce the amount of capacities secured for the next level $\sum_{v \in V_l} c_v^{next}$ as long as this allocation still enables the construction of more than K trees in F_{l+1} (lines 6-7),
2) when we reach the tipping point for which $\sum_{v \in V_l} c_v^{next} < K$, we have to reject peers from level l to save more capacities. We choose to reject the peer $v \in V_l$ that has the least capacity c_v (lines 8-9),
3) finally, when the set of remaining peers V_l cannot gather enough resources to serve the next level, a radical decision should be taken: either level $l+1$ cannot be served (and all subsequent levels as well), or level l is sacrificed. We opt for the latter choice (lines 10-13).

C. Resource-Aware Multicast Trees

We detail now the algorithm for the construction of *one* intra-level overlay forest. This algorithm is a key routine called by Alg. 1 in lines 4, 15 and 19. This algorithm should consider the constraints described in §III-A (about the multicast trees) and §III-B (about the sharing of physical resources). It returns either a failed forest, or a forest containing the given number of trees such that all nodes are spanned in at least K trees.

We give the pseudocode of our algorithm for level l in Alg. 2. Note that our implemented algorithm contains various slight improvements, compared to this simplified version. This algorithm is based on two distinct steps.

First, we create the trees and we associate the roots (lines 1-5). The number of trees to generate is based on the resources secured by peers in level $l-1$. The peers with maximum capacities are chosen as roots. Then, we secure the capacity for level $l+1$. If the roots do not have collectively enough capacities to secure the requested amount of capacity, then the algorithm ends with a failure notification.

Second, we build the trees in a breadth-first manner (lines 10-21). The choice of which unspanned peer to add to the trees (line 14) is done in two ways in our implementation: we choose the nodes with minimum spare capacity (or empty capacity) when the height is H, while we choose the nodes with maximum spare capacity otherwise. The algorithm successfully returns the forest F_l if all peers are spanned K times.

IV. Simulation Results

We used the ns-2 network simulator to test the performance of our system. We considered a scenario where the overlay consists of four levels with 10 peers in each level. The encoded symbols of the rateless code were sent in successive UDP packets of size S bytes each. The size of an encoded symbol was one byte. An accurate Raptor code model proposed in [13] was used to simulate Raptor coding. With this model, a redundancy of 5%, gives a high probability of successful decoding [13]. The diffusion forest construction algorithm was run with $H = 3$. The number of packets K used in the forest construction algorithm was determined by $K = \frac{b\Delta(1+\epsilon)}{8 \times S}$ where

Algorithm 2 Forest construction algorithm

Input: Complete graph $G(V,E)$, capacity c_v for all $v \in V$,
V_l set of peers in level l (initially *unspanned*),
$C_l = \sum_{v \in V_{l-1}} c_v^{next}$ number of trees to build in level l,
C_l^{next} amount of resources to secure for level $l+1$,
number of packets K, maximum height H

Output: Height-bounded forest F_l

for k **from** 1 **to** C_l **do**
  $u \leftarrow$ the unspanned node in V_l with max. capacity
  $p \leftarrow$ a root node with spare capacity in V_{l-1}
  create $T_l^k = (\{u\}, \emptyset)$ with p as source
end for
secure capacity c_u^{next} from all root nodes u
if impossible to secure overall capacity C_l^{next} **then**
  fail - **end of the algorithm**
end if
for k **from** 1 **to** C_l **do**
  for h **from** 0 **to** $H-1$ **do**
    while $\exists u$ at height h: $c_u > 0$ **and** $V_l \setminus T_l^k \neq \emptyset$ **do**
      $u \leftarrow$ a node in T_l^k at height h with $c_u > 0$
      $v \leftarrow$ an unspanned node $\notin T_l^k$
      add $\{v\}$ and edge (u,v) to T_l^k, update c_u
      if v is spanned in K trees **then**
        mark v as *spanned*
      end if
    end while
  end for
end for
if at least one node is still *unspanned* **then**
  fail - **end of the algorithm**
else
  $F_l \leftarrow \{T_l^k, \forall k \in \{1, \ldots, C_l\}\}$
end if

b is the source bit rate in bps, Δ is the transmission duration in s, and ϵ is the rateless code redundancy factor.

For the peer upload capacity, we used a log-normal distribution with a mean of 512 kbps. The second parameter of the distribution, σ, was 0.1. This corresponds to a homogeneous configuration where the upload capacity of all peers is close to the mean. The download capacity was 10 Mbps for all peers. For the link latencies, we followed [14] and used a log-normal distribution with mean 17.19 ms and variance 0.0029.

Fig. 4 shows the average received peak signal-to-noise ratio (PSNR) in the P2P network as a function of the average used upload bandwidth for two packet loss rates. When a source block was not recovered, the last successfully received frame was copied. Results are given for the CIF Foreman video sequence encoded at 30 frames per second and 320 kbps. The forest was constructed for a fixed number of multicast trees corresponding to a maximum of 60% redundancy by setting $\epsilon = 0.6$ and $S = 1000$. The code redundancy was then varied by changing the packet size S, which changed the number of transmitted encoded symbols leading to different used bandwidths. We used S equal to 656, 688, 750, 812, 875, 938 and 1000 bytes corresponding to code redundancy

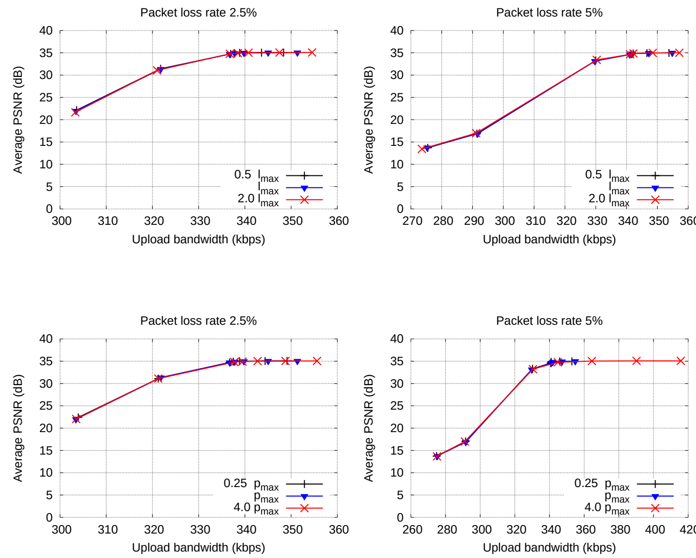


Fig. 4: Top: Average PSNR vs. average used upload bandwidth for the 320 kbps Foreman video when the value of $l_{\max}$ was the actual one, 100% higher, and 50% lower. Bottom: Average PSNR vs. average used upload bandwidth for the 320 kbps Foreman video when the value of $p_{\max}$ was the actual one, 300% higher, and 75% lower.

5%, 10%, 20%, 30%, 40%, 50% and 60%, respectively. The figures present results when the maximum network latency $l_{\max}$ (see §II) and the maximum packet loss rate $p_{\max}$ (see §II) were set to the actual values, as well as when they were estimated.

The results show that our system provided all peers with the highest achievable PSNR (35.07 dB) at low bandwidth cost. They also show that our system is robust against a mismatch between the used values and the actual ones. We obtained similar results for video bit rates 640 kbps and 960 kbps, as well as for video sequences Mother and Daughter and Mobile.

Fig. 5 shows the upload bandwidth used by the source to ensure that all 40 peers get maximum video quality. For a given packet loss rate, we ran simulations for different redundancies and used the one that maximized the PSNR (Table I).

The source transmission rate depends on the redundancy, loss rate, and the effectiveness of feedback. The source transmission rate increases with the loss rate because as the loss rate increases it delays the feedback. We also observe that when Δ decreased, the effectiveness of feedback also decreased. The source may have already sent all the data with the full selected redundancy before feedback arrives.

We now study the effects of peer leaving on the PSNR of the remaining peers. We simulated a pre-scheduled live broadcast event announced to all peers in advance. Each peer joined the system at the start of the event and remained online for an exponentially distributed time τ (with probability density function $\lambda e^{-\lambda\tau}$ for $\tau \geq 0$) after which it left the system and did not rejoin. We used $\lambda = 0.4$ to ensure that at least a few

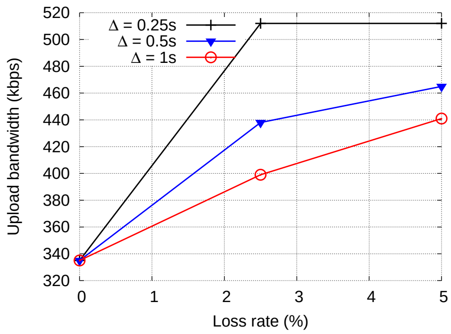


Fig. 5: Source transmission rate to provide maximum video quality in the P2P network for various packet loss rates and transmission durations.

TABLE I: Raptor code redundancy to provide maximum video quality in the P2P network for various packet loss rates and transmission durations.

	0%	2.5%	5%
$\Delta = 0.25$ s	5%	60%	60%
$\Delta = 0.5$ s	5%	60%	60%
$\Delta = 1$ s	5%	50%	60%

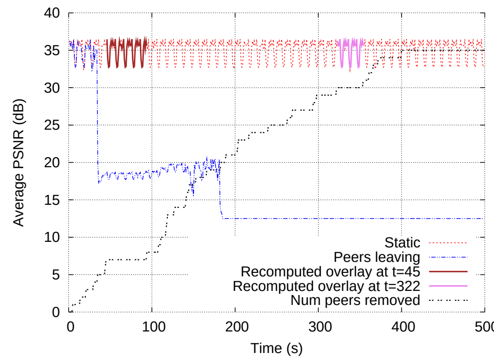


Fig. 6: Average PSNR in the P2P network vs. time. "Static": no peer leaves. "Peers leaving": peers leave according to curve "Num peers removed". "Recomputed overlay at t=45": P2P server reconstructs the overlay after 45 s. 'Recomputed overlay at t=322": P2P server reconstructs the overlay after 322 s.

peers stay in the system until the end of the session.

Fig. 6 shows the average PSNR of peers as a function of time. As more and more peers left the system, the average PSNR of the remaining peers dropped. The drop in PSNR is negligible after the first few departures but it increases significantly with further leaving of peers. However, the drop in PSNR can always be recovered by recomputing the overlay. This is shown by the average PSNR with recomputed overlay at times 45 s and 322 s when 7 and 30 peers had left respectively. The PSNR for the recomputed overlay is identical to the one where no peer leaves. The PSNR with recomputed overlay is shown only up to the time when the next peer leaves at which time the overlay can be recomputed.

V. Conclusion

We presented a P2P live video system that uses levels to define a hierarchy among peers. The system relies on rateless coding and multicast trees for video diffusion. Multicast trees are built with a low-complexity polynomial-time algorithm that tries to minimize the source transmission rate while guaranteeing on-time delivery and reliability at the peers. Minimizing the source transmission rate is critical for user-generated video where the source usually has limited upload speed. Experimental results with the ns-2 network simulator showed that the system can offer high video quality to all peers at low bandwidth cost. The results also highlighted the benefits of rateless coding for fast delivery and robustness against packet loss.

The system was designed for a static scenario where all peers join before the start of the video session and remain subscribed until the end. Applications include team mates participating in an MMOG quest while sharing screen-captured video of their gameplay for coordination purposes. Since even for such applications one cannot exclude the eventuality that peers decide to leave, we tested the performance of the system accordingly. Simulations showed that the system can sustain high performance as long as the number of dropped peers is moderate. When too many peers leave, the performance can deteriorate suddenly because the probability that a vital peer is dropped increases. While such cases are unlikely to happen for the intended applications, they can always be handled by the following procedure. Keep-alive messages are periodically sent from the peers to the server. The server regularly checks if all peers send keep-alive messages within a given time interval. If a peer stops sending keep-alive messages, the server removes it from the overlay and updates the multicast trees.

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. http://www.justin.tv/.
[2] Online. http://www.livestream.com/.
[3] Online. http://www.ustream.tv/.
[4] Online. http://www.xfire.com/.
[5] Online. http://www.twitch.tv/.
[6] Online. Available at: http://www.cng-project.eu/.
[7] A. Shokrollahi, "Raptor codes," IEEE Trans. Inf. Theory, vol. 52, pp. 2551–2567, June 2006.
[8] C. Wu and B. Li, "rstream: Resilient and optimal peer-to-peer streaming with rateless codes,"IEEE Trans. Parallel Distrib. Syst., vol. 19, no. 1, pp. 77-92, 2008.
[9] M. Grangetto, R. Gaeta, and M. Sereno, "Rateless codes network coding for simple and efficient P2P video streaming," in Proc. IEEE ICME, 2009.
[10] M. Grangetto, R. Gaeta, and M. Sereno, "Reducing content distribution time in P2P based multicast using rateless codes," in Proc. Italian Networking Workshop, pp. 1–12, Bologna, 2009.

[11] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," IEEE Trans. Circuits and Systems for Video Technology," vol. 13, no. 7, pp. 560–576, Jul. 2003.
[12] D. Mills, "Network Time Protocol (Version 3) Specification, Implementation and Analysis. Request for Comments 1305, Internet Engineering Task Force," 1992. http://www.rfc-editor.org/rfc/rfc1305.txt.
[13] M. Luby, T. Gasiba, T. Stockhammer, and M. Watson, "Reliable multimedia download delivery in cellular broadcast networks," IEEE Trans. Broadcasting, vol. 53, pp. 235-246, 2007.
[14] A. Hernandez and E. Magana, "One-way delay measurement and characterization," in Proc. ICNS '07, Athens, Greece, June 2007.

Resource Allocation in Underprovisioned Multioverlay Live Video Sharing Services

Jiayi Liu
Telecom Bretagne, France
jiayi.liu@telecom-bretagne.eu

Shakeel Ahmad
De Montfort University, UK
sahmad@dmu.ac.uk

Eliya Buyukkaya
Telecom Bretagne, France
eliya.buyukka@telecom-bretagne.eu

Raouf Hamzaoui
De Montfort University, UK
rhamzaoui@dmu.ac.uk

Gwendal Simon
Telecom Bretagne, France
gwendal.simon@telecom-bretagne.eu

ABSTRACT

In a multioverlay live video sharing service consisting of multiple independent peer-to-peer live video streaming systems, a user can simultaneously watch multiple live video streams. A major challenge for such services is the inter-overlay bandwidth competition problem, which is to find an upload bandwidth allocation between the overlays each peer has subscribed to. So far, no solution has been proposed in the literature for the important case where the overall system is underprovisioned, that is, when peers do not have enough upload bandwidth to ensure a distribution of videos at full quality. We show that an allocation of upload resources that minimizes the wastage of resources (i.e., minimizes the upload bandwidth allocated to overprovisioned overlays) can be computed in polynomial time. Then we present a generic model that allows the design of different strategies for the management of the resource deficit in underprovisioned systems. Finally, we provide relevant simulation results to demonstrate the gains in video quality resulting from the implementation of our solutions.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems; H.4.3 [**Information Systems Applications**]: Communications Applications

General Terms

Algorithms, Design, Performance

Keywords

Peer-to-Peer Live Streaming, Multioverlay, Resource Allocation

CSWS'12, December 10, 2012, Nice, France.

1. INTRODUCTION

The popularity of academic papers related to peer-to-peer (P2P) systems has followed a "bubble" evolution over the last decade. In their well-documented analysis [4], Li, Feng and Li show that after a surge of popularity from 2000 to 2008, P2P has become a "cold topic". On the one hand, this sharp decrease in scientific interest makes sense because P2P systems are now mature, well-understood, and can be fearlessly used by industry. Moreover, compared to cloud offers, the trade-off between operational costs and guarantee of performance is now less favorable to P2P systems. On the other hand, companies that implement P2P systems face new issues, which are not directly related to the P2P systems, but rather to their integration in a global system.

In this paper, we address a capacity management problem for P2P systems. This problem was brought to our attention by an independent producer of Massively Multiplayer Online Game (MMOG). MMOGs enable users to play against other players or to build groups to achieve missions. Within a group, synchronisation among teammates requires communication tools. Unfortunately, existing games offer only a few basic communications tools (if any). Surveys [1] have 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 for gamers [9, 14] rely on a centralized architecture. Even when the system is coupled with a Content Delivery Network (CDN), this solution is not cost-effective. Indeed, the popularity distribution of such service poses a major challenge: (*i*) a large proportion of players are likely to act as video sources, so there are many live streams to deal with, and (*ii*) each stream is typically watched by a small population consisting of a few friends and teammates.

MMOG producers foresee a *multioverlay* P2P system consisting of multiple P2P live video streaming systems [10]. Some users are sources, which emit a user-generated live video, while the others are peers, which receive one or several videos and participate in diffusing these videos to other peers. Each P2P network contains one source and all peers that have *subscribed* to its live stream (channel).

In the context of MMOGs, players in a team share their videos for coordination purposes, so a peer can watch several videos simultaneously. The fundamental problem faced by a peer that subscribes to several P2P networks is how to share its uplink bandwidth among these concurrent systems. Only a few papers [10–12] have addressed this resource allocation problem, and they focused on the case where the overall system is overprovisioned. Our tests based on realistic settings will reveal that, on the contrary, *multioverlay systems are underprovisioned*, i.e., the upload bandwidth allocated to this system is smaller than the streaming demand.

Some previous works have suggested to assist the P2P overlays by resources from servers in datacenters [7,13]. This solution however does not accommodate well a scenario with many small-size overlays, as it would require reserving and managing a large number of Virtual Machines, each generating a small amount of traffic.

The current paper makes the following original contributions to the problem of resource allocation for underprovisioned multioverlay systems.

- We show how to minimize the waste of resources resulting from overprovisioning some overlays although some others are underprovisioned. We show that an optimal solution can be found in a time that is polynomial in the number of users.
- We show how to share the bandwidth deficit among channels so that a pre-determined policy is satisfied. We consider two policies: minimizing the number of underprovisioned channels, and prioritizing the most popular channels. We tackle this new problem in a generic way. We present a polynomial-time algorithm that finds allocations that are optimal in terms of resource waste and correspond to the best allocations with respect to the policy defined by the service provider.

Note that our resource allocation strategies are compatible with any P2P video streaming protocol. Some previous works have proposed to solve the resource allocation competition and the P2P video streaming delivery all at once [12]. These approaches are not agnostic to the P2P video streaming system. We believe contrarily that a clear separation between *inter-overlay* resource management and *intra-overlay* video diffusion is crucial. Indeed, the multioverlay system should leverage the advances in P2P streaming systems.

We compare our strategies to the algorithms presented in [10, 11]. With simulation parameters based on previous observations, we show that the overall system is underprovisioned, which confirms our intuition that underprovisioning is a critical issue for multioverlay systems. We implement a mesh-based intra-overlay P2P simulator and compute the Peak Signal to Noise Ratio (PSNR) of the received stream for every peer. To our knowledge, this is the first time that the quality of experience of users in underprovisioned overlays is studied.

2. SYSTEM MODEL

We first give the notations used throughout the paper (see Table 1). Then, we present the bipartite graph that is the basis for our proposals. Finally, we show that the waste of resources can be minimized by determining the maximum flow in this bipartite graph.

The system includes multiple P2P overlays, one for each video stream, and a global server. The role of the server is to authorize a peer to watch a video emitted by a source and to compute the optimal bandwidth allocation. Peers transmit an estimation of their available upload capacity to the server on a regular basis. Then, based on these reports, the server computes and sends the optimal bandwidth allocation to each peer.

P, S	set of peers, set of sources
G_s, P_s	overlay of source s and set of peers in G_s
B_p	upload capacity of a peer p
b_p^s	upload capacity reserved by p for G_s
$G(p)$	set of sources to which peer p subscribed
d_s	video bit-rate of the video emitted by s
D_s, C_s	demand and capacity of G_s
Δ_s, Δ_s^r	provisioning and relative provisioning of G_s

Table 1: Notations

2.1 Notations

Sources. The set of sources is denoted by S. A source $s \in S$ is associated with an overlay G_s, which contains the set P_s of all peers that have subscribed to the video emitted by s. To avoid confusion, $s \notin P_s$.

Peer-to-Peer Live Streaming System. Our resource allocation strategies are independent of the intra-overlay structure. They can be used with any state-of-the-art P2P live video streaming system.

Peer Uplink Management. The set of all peers is denoted by P. We denote by $G(p)$ the set of sources from which the peer p receives a video. Every peer uses its uplink to transfer the chunks it received to other peers in the same overlay. The upload capacity of p is denoted by B_p while the upload capacity that p reserves to serve video chunks in the overlay G_s is denoted by b_p^s. Clearly, $\sum_{s \in G(p)} b_p^s \leq B_p$. We assume that s reserves all of its upload bandwidth to its overlay G_s.

Overlay Capacity. In an ideal system without control messages or network overhead, the capacity of an overlay G_s, which is denoted by C_s, would be equal to $\sum_{p \in P_s} b_p^s + B_s$, which is the aggregate upload bandwidth allocated from peers to s plus the capacity of s. In a real system, the control traffic cannot be neglected, so the overlay capacity should be reduced by a value proportional to the number of peers.

Overlay Demand. The demand of a source s corresponds to the smallest overlay capacity required to satisfy all peers in P_s. The bit-rate of the video emitted by s is denoted by d_s. Since overhead is already incorporated in the overlay capacity, we assume that the demand of s, denoted by D_s, is computed as $|P_s| \times d_s$.

Overlay Provisioning. The provisioning Δ_s of a given overlay G_s is the difference between its capacity C_s and its demand D_s, i.e., $\Delta_s = C_s - D_s$. An overlay is said to be underprovisioned when Δ_s is negative. The average upload capacity is smaller than the video bit-rate, so some peers in this overlay are unable to watch the video at full quality. On the other hand, the overlay is overprovisioned when Δ_s is positive.

Overlay Relative Provisioning. The relative provisioning Δ_s^r of a given overlay G_s is defined as the overlay provisioning divided by the number of peers in the overlay, that is, $\Delta_s^r = \frac{\Delta_s}{|P_s|}$. The more negative is the relative provisioning, the worse is the video quality experienced by the peers.

2.2 Bipartite Flow Network Model

We aim at minimizing the wastage of resources caused by allocating resources to overprovisioned overlays although they could be allocated to underprovisioned ones. We show that this problem can be solved by determining the maximum flow in a flow network.

We build an *abstract* structure, which is a flow network. A link in this flow network is an abstract representation of the existence of a relationship between two system elements. Our flow network $N = (V, E)$ is built according to source-peer relationships (Fig. 1). The set V contains a virtual fountain l, a virtual sink q, the set P of all peers in the system, and the set S of all sources. Thus $V = P \cup S \cup \{l, q\}$.

The set of edges E represents resource allocation. The capacity indicates a limitation in the amount of bandwidth resources that a node can reserve to another. The set E contains three subsets. First, $E_1 = \{(l \to p) : p \in P\}$, contains edges from the fountain to each peer p with a maximum capacity of B_p. Then, $E_2 = \{(p \to s) : p \in P, s \in G(p)\}$, contains edges from p to s if p subscribes to s with infinite maximum capacity. The third set, $E_3 = \{(s \to q) : s \in S\}$, contains edges from each source s to the sink with a maximum capacity equal to $D_s - B_s$, the overlay demand minus the source capacity.

Given demands and capacities, the resource allocation should ensure that no resource is allocated to an overprovisioned source when it could have been allocated to an underprovisioned one. Let S^+ (respectively S^-) be the set of sources with a positive (respectively negative) overlay provisioning. We look for an uplink sharing among the overlays such that the sum of all negative provisionings is minimum. Hence, our goal is to minimize $\sum_{s \in S^-} |\Delta_s|$.

Proposition 1. *The total underprovisioning is minimum iff the maximum flow is achieved in network N.*

Proof. We denote by $f_{s,q}$ the flow on the arc $(s \to q)$. The cut-set between $V \setminus \{q\}$ and $\{q\}$ bounds a flow: $|f| = \sum_{s \in S} f_{s,q}$.

For each source s, the absolute underprovisioning $|\Delta_s|$ is equal to $D_s - B_s - f_{s,q}$. Note that, for a source s in S^+, we have that $D_s - B_s - f_{s,q}$ is equal to zero because the flow $f_{s,q}$ cannot be greater than $D_s - B_s$. Thus,

$$\sum_{s \in S^-} |\Delta_s| = \sum_{s \in S^-} (D_s - B_s - f_{s,q})$$

$$= \sum_{s \in S} (D_s - B_s - f_{s,q}) = A - \sum_{s \in S} f_{s,q}$$

where A is a constant. Minimizing $\sum_{s \in S^-} |\Delta_s|$ is equivalent to maximizing $\sum_{s \in S} f_{s,q}$. Moreover, (i) edges from l to P fuel the system with all capacities $C = \sum_p B_p$, (ii) edges from P to S respect peer subscription. Hence, the overall underprovisioning $\sum_{s \in S^-} |\Delta_s|$ is minimized iff the flow is maximized. □

3. POLICY DRIVEN BANDWIDTH ALLOCATION STRATEGIES

There is often more than one resource allocation that minimize the total underprovisioning because the maximum flow is not unique. The service provider has here an opportunity to design its own *policy* for preserving some sources to be affected by the overall underprovisioning. In this section, we present a *generic* way to implement such a policy.

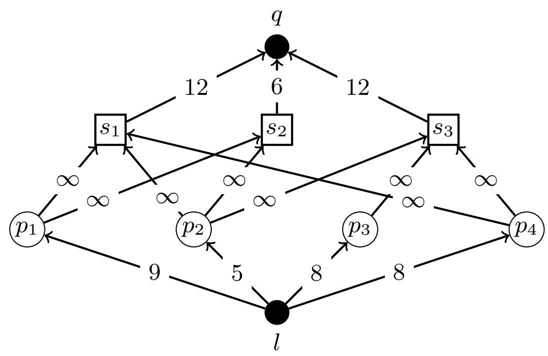


Figure 1: Example of the bipartite flow network model. Numbers in the arrows are capacities, *i.e.* limitations in the amount of bandwidth resources that a node can reserve to another.

3.1 Cost-Function Flow Network

In our flow network model, we propose to define a cost function for edges in E_3. As the minimum-cost maximum-flow problem aims at minimizing the sum of flow cost, the edges associated with a lower cost will be prioritized in the bandwidth allocation. Consequently, different bandwidth allocation strategies can be applied by using correspondingly defined cost functions. This approach is generic in the sense that various cost functions can be designed, which result in various resource allocations. We present later two distinct bandwidth allocation strategies.

Strategy I: Prioritize Channel Diversity The goal is to satisfy the maximum number of sources regardless of their popularity. A source is said to be satisfied when its relative provisioning is positive, or slightly negative. Thus, we define our first strategy as one that minimizes the total relative under-provisioning: $\sum_{s \in S^-} |\Delta_s^r|$. Let us define the cost function as:

$$cost_1(e) = \begin{cases} 1, \text{if } e \in E_1 \cup E_2 \\ 1 - \frac{1}{|P_s|}, \text{if } e \in E_3 \end{cases}$$

We now show that this policy can be obtained by the minimum-cost maximum flow corresponding to $cost_1(e)$.

Proposition 2. *The minimum-cost maximum flow corresponding to $cost_1(e)$ minimizes the total relative underprovisioning $\sum_{s \in S^-} |\Delta_s^r|$.*

Proof.

$$\sum_e cost(e) \times f_e = \sum_{e \in E_1} f_e + \sum_{e \in E_2} f_e + \sum_{e \in E_3} (1 - \frac{1}{|P_s|}) \times f_e$$

E_1 and E_2 are cut sets separating l and q, so $\sum_{e \in E_1} f_e = \sum_{e \in E_2} f_e = |f_{max}|$. Thus

$$\sum_e cost(e) \times f_e = 2|f_{max}| + \sum_{e \in E_3} f_e - \sum_{s \in S} \frac{D_s - B_s - |\Delta_s|}{|P_s|}$$

$$= 3|f_{max}| - \sum_{s \in S} \frac{D_s - B_s}{|P_s|} + \sum_{s \in S^-} |\Delta_s^r|$$

where the first two terms are constants. Thus, minimizing $\sum_e cost(e) \times f_e$ is equivalent to minimizing $\sum_{s \in S^-} |\Delta_s^r|$. □

Strategy II: Prioritize Channel Popularity The goal is to prioritize the most popular channels. This can be achieved by maximizing the number of *unsatisfied* sources, for example, maximizing the total relative underprovisioning $\sum_{s \in S^-} |\Delta_s^r|$.

$$cost_2(e) = \begin{cases} 1, \text{if } e \in E_1 \cup E_2 \\ \frac{1}{|P_s|}, \text{if } e \in E_3 \end{cases}$$

PROPOSITION 3. *The minimum-cost maximum flow corresponding to $cost_2(e)$ maximizes $\sum_{s \in S^-} |\Delta_s^r|$.*

PROOF. The proof is similar to that of Proposition 2. □

3.2 Practical Optimization

Previous strategies have a common drawback: they aim at ensuring a zero provisioning to the prioritized sources, although a slightly negative relative provisioning would have a small impact on the overall quality of experience. To address this problem, we introduce a tunable *tolerable video quality parameter* k and say that an overlay has tolerable video quality if its relative provisioning $|\Delta_s^r|$ is smaller than k. The demand D_s of a source s can be interpreted as the amount of upload bandwidth required by s to be provisioned as $\Delta_s = 0$. If the system is globally very underprovisioned, rather than requiring perfect video quality on each source, we only require tolerable video quality. This can be done by tuning the parameter k. Consequently, the actual D_s is equal to $(d_s - k) \times |P_s|$.

4. IMPLEMENTATION DETAILS

This section provides practical implementation details of the multioverlay system. The system is managed by a global server called *P2PServer*. Peers send reports to P2PServer on a regular basis, e.g., every minute. This report contains an estimation of the peer's available bandwidth. Based on these reports, P2PServer computes and sends the optimal bandwidth allocation to each peer.

4.1 Coping with Peer Dynamics

The system described above is easy to implement in a static environment; however peer-to-peer live video streaming applications face the problem of peer churn. Peers are ordinary users who are free to enter and leave the system, and also to switch from one channel to another. Moreover, the available bandwidth can vary between two measurements. Even if the computation of bandwidth allocation can be done in polynomial time, it is unrealistic to repeat it after every event.

To cope with peer churn, the strategy proposed in [10,11] can be used. The system time can be cut into *sessions*. The computation of upload bandwidth allocation in a session only involves peers and sources that exist in the system at the starting point of that session. Every event that occurs during a session will have an effect on the system until the end of the session and the start of the next session. So, the capacity of the system to handle the dynamic behavior of peers only depends on the choice of the length of a session. The shorter is a session, the better the system reacts to changes, but the more computation it requires.

nb. peers	1,000	5,000	10,000	50,000	100,000
time (sec)	0.005	0.086	0.311	7.455	31.887

Table 2: Computation time for minimum-cost maximum-flow based algorithms.

4.2 Peer-Server Communication Overhead

We are concerned about the extra-traffic generated by the system when peers send reports to P2PServer, and when P2PServer sends bandwidth allocation information to peers. However, this extra-traffic has to be seen in light of the huge amount of data needed for the live video streams. For example, if the average number of videos watched by a peer is three, two bytes are used to specify the upload bandwidth reserved to an overlay, and the bandwidth allocation is recomputed every minute, then P2PServer needs to transmit 0.8 bps per peer. If we consider in addition the 54 bytes for the TCP/IP/Ethernet packet overhead, then 0.8 Mbps server upload bandwidth would be needed for 100,000 users. Similarly, if the report sent by a peer to P2PServer includes four bytes to specify the estimated upload bandwidth and four bytes for the peer ID, then only 0.826 Mbps server download bandwidth would be needed for 100,000 users. This example shows that, from a network standpoint, the system can be implemented without much fear for scalability.

4.3 Algorithm Computation Time

Another concern is the scalability in terms of computation time. We measured the exact computation time for the centralized minimum-cost maximum-flow based algorithms. We used the *preflow* maximum flow algorithm and a scaling approximation minimum-cost flow algorithm [3]. The measurement was done on a typical server (2 × 4 cores Intel(R) Xeon(R) 2.67GHz CPUs). We computed the average running time for 5 different runs. Results are given in Table 2. We changed the instance size by increasing the number of peers from 1,000 to 100,000. For each instance, the number of sources was set to 10% of the population. The number of channels a peer watches was randomly chosen between 1 and 5.

Our measurement demonstrates that a practical implementation of the minimum-cost maximum-flow algorithm can compute the resource allocation for very large instances (with 100,000 peers and 10,000 sources) in reasonable time. It also demonstrates that refreshing the bandwidth allocation every minute is an appropriate choice. All in all, the small peer to server communication overhead and the fast resource allocation algorithm demonstrate the feasibility of managing a centralized server to recompute periodically an optimal resource allocation in a dynamic environment.

5. SIMULATION RESULTS

We evaluated the proposed inter-overlay bandwidth allocation algorithms through simulations based on a realistic model. The chunk diffusion of each overlay was simulated with *P2PTVSim* [5]. Using these results, we evaluated the video quality at each receiving peer by measuring the average luminance PSNR.

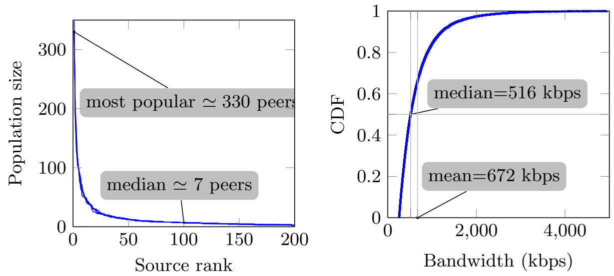
5.1 Simulation Setting

Our simulation is based on an MMOG video sharing application: players publish their own videos and share them with their friends. The Xfire measurements presented in [6] show that these platforms are social networks. We thus look for real traces of social networks that have the same characteristics as the one exhibited in [6]. We found that the Facebook social network of the Smith College (MA) [8] is very close to Xfire, especially the average social degree (around 65 connections per user). This social network contains 2,970 users.

We then selected the sources. We ensured that the Pareto rule is obeyed, that is, 80% of videos are published by 20% of the most active peers. It means that the most socially active peers have a higher probability to be sources. A peer decides to become a source based on a probability equal to $e^{-\frac{i}{\tau N}}$, where i denotes its rank in terms of social degree in decreasing order, N denotes the total number of peers, and τ is a parameter. We chose $\tau = 0.1$; on average, 214 channels are created.

We then selected the viewers. We model the peer watching decision such that the resulting video popularity follows the same Zipf's law as in [6]. Sources are ranked by their social degree in decreasing order. Let s_i be the ith source with n_{s_i} denoting its number of friends. We associate each source s_i with an *attractiveness index* α_i which is equal to the probability that its friends watch this video. The index α_i is calculated such that the channel population follows Zipf's law: $\frac{n_{s_i}}{n_{s_1}} \cdot \alpha_i \sim i^{-b}$. Fig. 2(a) shows the resulting channel popularity.

For the upload capacity of peers, we followed some recent works that model home upload capacity with a *log-normal* distribution ranging from 256 kbps to 5 Mbps [2]. Fig. 2(b) shows the cumulative distribution function (cdf) of the upload bandwidth distribution. Each video diffusion in an overlay was mesh-based pull-based, and evaluated with 400 chunks. The video bit rate varied from 240 kbps (when the system was overprovisioned) to 330 kbps (when it was underprovisioned). The P2P system overhead (control messages and redundant transmission) was set to 50 kbps.



(a) Channel popularity (b) Upload Bandwidth

Figure 2: Simulation settings

We used a 600-frame video by concatenating 300 frames of the Foreman sequence and 300 frames of the Mother and Daughter sequence. Both sequences are in CIF format and have a frame rate of 30 fps. The video was compressed with the H.264/AVC encoder at bit rates ranging from 240 to 330 kbps using the H.264 high profile. We used the GOP structure IBBPBBPBBP (10 frames per GOP). Each chunk corresponds to one GOP, so each GOP is played back independently of the other chunks. At the receiver side, we

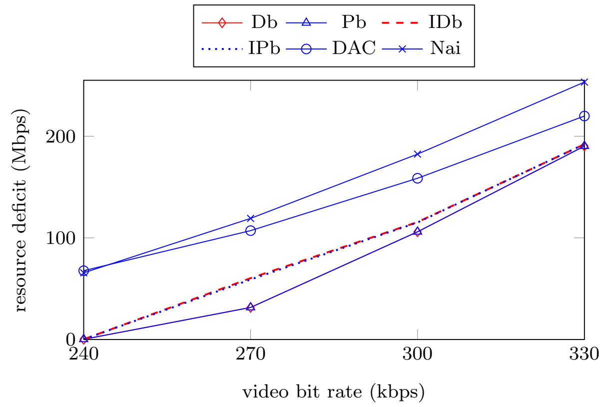


Figure 3: Amount of missing upload capacity.

used the standard frame copy error concealment technique to deal with lost frames. With this technique, the last frame of the last decoded GOP is used to represent all frames of a missing chunk.

We compared the following six algorithms:

DAC: This algorithm was proposed in [10, 11]. It fairly allocates upload bandwidth based on overlay demands, but it is blind to the provisioning of overlays.

Diversity-based **(Db)**: corresponds to the strategy that enhance channel diversity.

Popularity-based **(Pb)**: corresponds to the strategy that serves the most popular channels first.

Improved diversity-based **(IDb)** and *Improved popularity-based* **(IPb)**: include the tolerable video quality parameters to the policies of **Db** and **Pb**, respectively. The system is overprovisioned with video bit rate 240 kbps, then k was set to 0 kbps. For the other video bit rates, k was set to 50 kbps. *Naive* **(Nai)**: This algorithm equally divides the upload bandwidth among the overlays.

5.2 Provisioning Results

Fig. 3 shows the total underprovisioning $\sum_{s\in S^-} |\Delta_s|$ of all algorithms for all video bit-rates. The two maximum-flow based approaches, (**Db** and **Pb**), minimize the total underprovisioning, producing identical curves. For video bit rate 240 kbps, the system is overprovisioned, and **Db**, **Pb**, **IDb** and **IPb** have zero underprovisioning, which means that every overlay in the system is well provisioned. We observe that both **IDb** and **IPb** (which also have identical curves) can have a slightly larger underprovisioning than their non-improved counterparts. Indeed, the diminution of the overlay demand can prevent the computation of a maximum flow in the flow network. Finding the best tolerable video quality parameter according to the system state and the video bitrate is left as a future work. The **DAC** allocation according to the demand favors the most demanding overlays, with the cost that less demanding overlays are underprovisioned. It performs almost as badly as the **Nai** algorithm when the system is slightly underprovisioned, and then improves when all demands grow.

5.3 PSNR Results

We provide objective video quality results by measuring the average luminance PSNR at each receiving peer. To illustrate the distribution of the PSNR, we show the com-

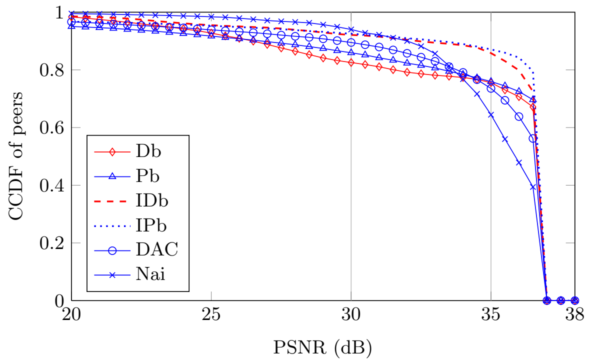


Figure 4: CCDF of PSNR for video bit rate 330 kbps.

plementary cumulative distribution function (ccdf) of the PSNR for video bit rate 330 kbps (Fig. 4). We distinguish three sets of allocations: the maximum-flow allocations (**Db**, and **Pb**), their improved counterparts (**IDb** and **IPb**), and **DAC** and **Nai**. For the first two groups, lines almost overlap between allocations. For the maximum achievable PSNR value (37 dB), we observe that a significant proportion of peers experienced high video quality: 70% (respectively 90%) of peers for the maximum-flow allocations (respectively the improved ones). For **DAC** and **Nai**, this percentage is 40% and 60%, respectively. With both **IDb** and **IPb**, only 15% of peers get video with PSNR below 35 dB, which demonstrates the potential of these policies. Note also that the maximum-flow allocations (**Db**, and **Pb**) have extreme bandwidth allocation strategies that result in around 20% of peers with PSNR below 30 dB.

6. CONCLUSION

Some (supposedly scalable and robust) technical solutions can meet unexpected capacity issues when implemented in a real system. In this paper, we consider the challenge of peers contributing to several overlays. This problem is almost ignored in the P2P streaming literature and can prevent an efficient deployment of P2P technologies.

First, *global underprovisioning can be minimized by a smart resource allocation algorithm*. Flow-based algorithms, such as the one presented in this paper, are traditionally used for finding efficient matching; they are appropriate for resource allocation problems as well. Fortunately, there exist very efficient implementations of flow algorithms. From our experience, a server can easily manage up to thousands of simultaneous peers. Moreover, distributed flow algorithms can be implemented in clusters of servers. In summary, the implementation of optimal resource allocation is possible, and it significantly reduces the impact of underprovisioning compared to a naive approach.

Second, *sharing the deficit in a smart way is not restricted to fair policies*. In this study, the resources are under the exclusive control of the service provider, and it is probable that business-oriented motivations prevail over fairness. We made a first step in the development of such pre-determined allocation policies with our original minimum-cost maximum flow algorithm. Note that this algorithm can apply to other policies as well. For example, it is straightforward to define a cost function to serve in priority a premium class of fee-paying sources.

Third, *tolerating a slight underprovisioning for everybody is a remarkably efficient way to reduce the impact of underprovisioning*. Multimedia services include multiple technologies to cope with packet loss, including data redundancy and channel coding techniques. As a matter of fact, a small amount of packet loss has imperceptible (or negligible) impact on the Quality of Experience. Our study shows that it is possible to leverage this characteristics to reduce the impact of underprovisioning in general. We take advantage of internal loss correction techniques. By purposely introducing a small underprovisioning, the overall system has far better performance. Our future works will include the development of algorithms to set this tolerable underprovisioning according to the context.

7. ACKNOWLEDGMENTS

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).

8. REFERENCES

[1] CNG. http://www.cng-project.eu/.

[2] M. Dischinger, A. Haeberlen, P. K. Gummadi, and S. Saroiu. Characterizing residential broadband networks. In *ACM IMC*, 2007.

[3] A. V. Goldberg. An efficient implementation of a scaling minimum-cost flow algorithm. *Journal of Algorithms*, 22:1–29, 1997.

[4] B. Li, Y. Feng, and B. Li. Rise and fall of the peer-to-peer empire. *Tsinghua Science and Technology*, 17(1):1–16, Feb. 2012.

[5] P2PTVSim. http://napa-wine.eu/cgi-bin/twiki/view/Public/P2PTVSim.

[6] S. Shen and A. Iosup. The xfire online meta-gaming network: Observation and high-level analysis. In *Workshop MMVE*, 2011.

[7] R. Sweha, V. Ishakian, and A. Bestavros. Angelcast: cloud-based peer-assisted live streaming using optimized multi-tree construction. In *ACM MMSys*, 2012.

[8] A. L. Traud, P. J. Mucha, and M. A. Porter. Social structure of facebook networks. *CoRR*, abs/1102.2166, 2011.

[9] TwitchTV. http://www.twitch.tv/.

[10] M. Wang, L. Xu, and B. Ramamurthy. A flexible divide-and-conquer protocol for multi-view peer-to-peer live streaming. In *IEEE P2P*, 2009.

[11] M. Wang, L. Xu, and B. Ramamurthy. Improving multi-view peer-to-peer live streaming systems with the divide-and-conquer strategy. *Computer Networks*, 55(18):4069–4085, 2011.

[12] C. Wu, B. Li, and Z. Li. Dynamic bandwidth auctions in multioverlay p2p streaming with network coding. *IEEE Trans. Parallel Distrib. Syst.*, 19(6):806–820, 2008.

[13] C. Wu, B. Li, and S. Zhao. On dynamic server provisioning in multichannel p2p live streaming. *IEEE/ACM Trans. Netw.*, 19(5):1317–1330, 2011.

[14] Xfire. http://www.xfire.com/.

Minimizing Server Throughput for Low-Delay Live Streaming in Content Delivery Networks

Fen Zhou
Telecom Bretagne
Brest, France
fen.zhou@telecom-bretagne.eu

Shakeel Ahmad
De Montfort University
Leicester, United Kingdom
sahmad@dmu.ac.uk

Eliya Buyukkaya
Telecom Bretagne
Brest, France
eliya.buyukka@telecom-bretagne.eu

Raouf Hamzaoui
De Montfort University
Leicester, United Kingdom
rhamzaoui@dmu.ac.uk

Gwendal Simon
Telecom Bretagne
Brest, France
gwendal.simon@telecom-bretagne.eu

ABSTRACT

Large-scale live streaming systems can experience bottlenecks within the infrastructure of the underlying Content Delivery Network. In particular, the "equipment bottleneck" occurs when the fan-out of a machine does not enable the concurrent transmission of a stream to multiple other equipments. In this paper, we aim to deliver a live stream to a set of destination nodes with minimum throughput at the source and limited increase of the streaming delay. We leverage on rateless codes and cooperation among destination nodes. With rateless codes, a node is able to decode a video block of k information symbols after receiving slightly more than k encoded symbols. To deliver the encoded symbols, we use multiple trees where inner nodes forward all received symbols. Our goal is to build a diffusion forest that minimizes the transmission rate at the source while guaranteeing on-time delivery and reliability at the nodes. When the network is assumed to be lossless and the constraint on delivery delay is relaxed, we give an algorithm that computes a diffusion forest resulting in the minimum source transmission rate. We also propose an effective heuristic algorithm for the general case where packet loss occurs and the delivery delay is bounded. Simulation results for realistic settings show that with our solution the source requires only slightly more than the video bit rate to reliably feed all nodes.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; G.2.2 [**Discrete Mathematics**]: Graph Theory—*network problems, trees*

General Terms

Algorithms, Design, Performance

Keywords

Rateless Codes, Live Streaming, Delivery Trees, CDN

1. INTRODUCTION

As illustrated by the recent decision from Korea Telecom to block Samsung's Smart TVs from accessing the Internet [7], *over-the-top live streaming* represents a major challenge for Internet Service Providers (ISPs) and Content Delivery Networks (CDNs). The bottleneck of large-scale live streaming delivery platforms is no longer in the *last-mile* since rate-adaptive streaming enables a match between video bit-rate and available bandwidth [8]. It is not in the *peering link* either because ISPs and CDNs develop more friendly relationships, including co-location of edge servers. Today, the bottleneck is likely located *within the CDN infrastructure*, which has to cope with the tremendous growth of video traffic and the multiplication of video encodings per TV channel. The (relative) failure of the latest SuperBowl streaming confirms this trend [13].

For live streaming, the infrastructure of a CDN comprises three distinct elements. The role of the *origin server* is to forward the original stream to *edge servers*. However the origin server has a limited fan-out, *i.e.,* the number of streams it can concurrently emit is limited. The CDN thus uses some *intermediate nodes*, which are called reflectors [3] or shield caches [5]. While these intermediate nodes have to serve a subset of the thousands of edge-servers, they have a limited upload capacity. The bottleneck in the CDN infrastructure comes either from the origin server that cannot serve the intermediates nodes, or from the intermediate node that cannot serve the edge servers.

In this paper, we introduce the generic problem of one source (either origin server or intermediate node) that has to serve a small set of nodes (respectively intermediate nodes or edge servers) in such a way that the throughput of the source is minimized and the streaming delay stays reasonable. We explore a solution based on rateless codes and data exchanges among the nodes. Since the latest rate-adaptive streaming protocols rely on a segmentation of streams into independent chunks, the use of *rateless codes* on every video chunk appears as an attractive idea. With rateless codes (*e.g.*, Raptor codes [12] and LT codes [9]), a video chunk of k information symbols can be recovered with high probability if slightly more than k encoded symbols are received. The idea is that the source applies rateless codes to every video chunk and delivers each created symbol to only one node, which then uses an application-level multicast tree to deliver this symbol to some other nodes.

1.1 Related Work

The first publications about live streaming in CDNs are ten years old (*e.g.*, [2, 4, 11]). Most of these publications emphasized the fan-out limitations of intermediate equipments. The focus of these papers was to build on top of large overlays a set of low-delay application-level multicast trees, subject to the constraints of the capacity of nodes. We revisit this objective through more recent delivery techniques, including rateless codes and video chunks. Moreover, we restrict our study to a specific local part of the whole overlay, where the bottleneck lies.

After five years without much academic activity, the delivery of a live stream in CDNs has become hot again with a series of recent publications [1,3,8]. These works consider the equipment bottleneck as well. They aim at creating delivery trees but their optimization purpose is generally to save bandwidth cost. Since we are within the CDN infrastructure, we neglect bandwidth cost.

Previous works related to rateless codes focus on peer-to-peer systems (*e.g.*, [6,14,15]). These works address problems related to scalability and management of churn. These issues are not major in CDN infrastructures, where the number of nodes is typically less than one hundred, and machines are rarely faulty.

1.2 Our Contributions

Our contributions are twofold:

- theoretical contribution: we aim at building a multi-tree structure that minimizes the source transmission rate. When we do not take into account the constraints of delay and packet loss, we give in §3.1 an exact algorithm (minimum source transmission rate). When we consider the joint height constraint of the delay and the reliability, we give in §3.2 an effective heuristic algorithm.

- practical contribution: our evaluation in §4 highlights that when the nodes can reserve an upload bandwidth that is larger than the video bit-rate, the source needs slightly more than the video bit-rate to transmit a video block to up to 25 nodes in a short delay and in a reliable way.

Our paper makes a first step toward addressing the problem of *equipment bottleneck* within CDN networks.

The rest of this paper is organized as follows. In §2, we present the system model and formulate the problem of minimizing the throughput for live streaming in CDNs. To this end, we propose two diffusion forest construction algorithms in §3. Then we evaluate the performance of our heuristic algorithm with simulations in §4. Finally, we conclude this paper and mention future work in §5.

2. SYSTEM MODEL

The source is the equipment that is the bottleneck of the CDN infrastructure. We suppose that the CDN is able to identify it. We denote it by s. This source is expected to deliver a flow of video chunks to some other equipments within the CDN networks. We denote by V this set of equipments ($|V| = n$), and we denote by $G = (V, E)$ the connected symmetric digraph modeling the small part of the whole CDN network that we consider in this paper. This is the network that is affected by the lack of upload capacity of the source. Our goal is to prevent the bottleneck to affect the whole CDN, thus we propose that these nodes cooperate in order to fix the bottleneck issue. We refer to this network as the *support network*. As usual, E is the set of links between nodes ($|E| = m$). The source s does not belong to V. An arc from u to v is denoted by (u, v). We assume that the support network is fully connected.

Every node $v \in V$ is associated with a *support capacity* (or capacity for short), denoted by c_v, which is the number of packets the node v can relay in the support network. It is important to understand that the equipments are not expected to re-transmit data to their sibling equipments in a hierarchical infrastructure like today's CDN networks. That is, intermediate nodes do usually not transmit data to intermediate nodes, and edge servers do not communicate with other edge servers. But there is not a lot of options for addressing the equipment bottleneck. We thus consider that the CDN has "reserved" a small fraction of the overall upload capacity in every equipment for the support network.

The *transmission delay* over the link (u, v) is d_{uv}. We assume that the transmission delay is a real value in $(0, 1]$, for example a fraction of the maximum Round-Trip Time (RTT) experienced in the CDN network.

The *transmission reliability* over the link (u, v), which corresponds to the probability that a packet sent by u is successfully received by v, is denoted by p_{uv}. It is also in $(0, 1]$. Note that the delay is additive while the reliability is multiplicative over arcs.

In the support network, we use a multi-tree structure for the diffusion of each video chunk. Each tree is the support for the transmission of *one* packet of encoded symbols from the source to nodes. To recover the video chunk, a node should receive at least K packets of encoded symbols. In other words, a node should belong to at least K trees. We illustrate our proposal in Figure 1. On the left, in the gray box, we represent the support network. On the right, we represent the four trees that allow each node to receive $K = 3$ packets of encoded symbols.

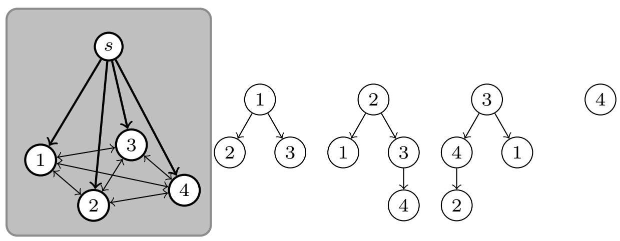


Figure 1: Multi-tree structure for the diffusion of each video chunk (four nodes and $K = 3$)

As the capacity of each node is limited, it may be impossible to use one tree to relay a packet to all nodes. In addition, to guarantee on-time delivery and reliability at the nodes, the end-to-end delay and transmission reliability on the path from the source to each node should be bounded to a reasonable value.

Our objective is to minimize the transmission rate for the source. The more trees are used in order to ensure that all nodes receive at least K packets, the higher is the source transmission rate. Thus, minimizing the source transmission rate is equivalent to minimizing the number of trees. In light of this, we formulate the video diffusion problem in the support network as a *Height-Bounded Spanning Forest problem with Capacity constraint* (HBSFC). The goal of HBSFC is to find a forest F with minimum cardinality such that

- each tree $T \subseteq G$ is rooted on a node, which directly receives a packet from s. For simplicity, we assume no packet loss between s and a root.
- for each node $v \in V$, the sum of its out-degrees in all trees of F is not greater than its capacity c_v. Let $deg_T^+(v)$ be the out-degree of v in $T \in F$,

$$\forall v \in V, \quad \sum_{T \in F} deg_T^+(v) \leq c_v \tag{1}$$

- each node $v \in V$ is spanned in at least K trees of F. Let $F(v)$ be the set of trees in which v is included. Then $|F(v)| \geq K, \forall v \in V$.
- for each tree $T \in F$, the transmission delay from the root to each node in T is no bigger than a given threshold $D_h > 0$. That is, the *delay height* of each tree is bounded by D_h.
- for each tree $T \in F$, each node $v \in T$ receives a packet from the root with a reliability no smaller than a reliability bound $P_h \in (0, 1]$. Let $SP_T(v)$ be the list of arcs in the path from the root to v in tree T. Then

$$\forall T \in F, \forall v \in T, \quad \prod_{e \in SP_T(v)} p_e \geq P_h \tag{2}$$

By applying logarithms on both sides, we obtain:

$$\forall T \in F, \forall v \in T, \quad \sum_{e \in SP_T(v)} \log(p_e) \geq \log(P_h) \tag{3}$$

which means that the *logarithmic reliability height* of each tree is bounded by $\log(P_h)$.

3. DIFFUSION FOREST CONSTRUCTION ALGORITHMS

We propose two *polynomial-time* algorithms to optimize the diffusion forest in two models: unbounded height model and bounded height model.

3.1 Unbounded Height Model

We first assume an unbounded height model: (i) the transmission reliability is $p_e = 1$ for all $e \in E$, and (ii) the delay threshold D_h satisfies $D_h \geq n - 1$, so that no tree will violate this constraint. Please refer to Algorithm 1 for the pseudocode of our algorithm. Note that notation $\overline{V_k}$ stands for $V \setminus V_k$ and (V_k, E_k) represents tree T_k. We use c'_v to denote the available capacity of a node v.

This algorithm consists of two phases. First, we construct K trees, and maximize the number of nodes that are spanned

Algorithm 1: Unbounded Height Model

Input : Complete graph $G = (V, E)$, capacity c_v of $v \in V$, number of packets K
Output: A forest F

$k \leftarrow 1$;
$F \leftarrow \emptyset$;
for $v \in V$ **do** $c'_v = c_v$;
$V_c \leftarrow \{v : c'_v > 0, v \in V\}$;
while $k \leq K$ **and** $V_c \neq \emptyset$ **do**
    find node $r \in V_c$ **s.t.** $c'_r = \max\{c'_v : v \in V_c\}$;
    $V_k \leftarrow \{r\}$;
    $E_k \leftarrow \emptyset$;
    $T_k \leftarrow (V_k, E_k)$;
    while $V_k \neq V$ **and** $V_c \neq \emptyset$ **do**
        find node $u \in \overline{V_k}$ **s.t.** $c'_u = \max\{c'_v : v \in \overline{V_k}\}$;
        pick any node $w \in V_k \cap V_c$;
        $V_k \leftarrow V_k \cup \{u\}$;
        $E_k \leftarrow E_k \cup \{(w, u)\}$;
        $T_k \leftarrow (V_k, E_k)$;
        $c'_w \leftarrow c'_w - 1$;
        if $c'_w = 0$ **then**
            $V_c \leftarrow V_c \setminus \{w\}$
    $F \leftarrow F \cup \{T_k\}$;
    $k \leftarrow k + 1$;
for $v \in V$ **do**
    $k_v \leftarrow$ number of trees v belongs to;
    if $k_v < K$ **then**
        add $(K - k_v)$ trees $\big(\{v\}, \emptyset\big)$ to F

in these trees. For each tree, the root r is selected as the node having the biggest available capacity. Then, we attach nodes to this tree starting from the nodes with biggest available capacity until either all nodes are spanned or no node has available capacity. Second, we take care of nodes that are not spanned in the K trees. As all nodes have exhausted their capacities, we span them by building *shallow trees* containing only one node. For example, the tree on the right in Fig. 1 is a shallow tree as it contains only the node 4.

THEOREM 1. *In the unbounded height model, the optimal forest has cardinality*

$$|F| = \begin{cases} K & \text{if } \sum_{v \in V} c_v \geq K(n-1) \\ K \times n - \sum_{v \in V} c_v & \text{otherwise} \end{cases} \tag{4}$$

PROOF. Let c_v^F be the sum of the out-degrees of node v in F. Then the number of nodes in F is equal to $\sum_{v \in V} c_v^F + |F|$. In an optimal solution, exactly $K \times n$ nodes belong to F. We have

$$|F| + \sum_{v \in V} c_v \geq |F| + \sum_{v \in V} c_v^F = K \times n \tag{5}$$

When $|F| = K$, the sum of capacities should thus verify:

$$\sum_{v \in V} c_v \geq K(n-1) \tag{6}$$

Otherwise, the total capacity is not enough to span $K \times n$ nodes in K trees. To minimize $|F|$, we should use all the nodal capacities, i.e., $\forall v \in V, c_v^F = c_v$. Thus, we obtain a forest cardinality of $|F| = K \times n - \sum_{v \in V} c_v$. $\square$

THEOREM 2. *Algorithm 1 computes an optimal solution*

PROOF. At step (5) of Algorithm 1, the first loop terminates when one of the following conditions is satisfied:

- All nodes are spanned K times in the K trees. Thus only K trees are required, $|F| = K$. Every tree in F contains n nodes, so exactly $K(n-1)$ out degree is needed. Thus we have $\sum_{v \in V} c_v \geq K(n-1)$.
- All nodes have exhausted their capacity ($V_c = \emptyset$) before finishing K trees. Let $\overline{k} \leq K$ be the number of constructed trees. These trees contain $\overline{k} + \sum_{v \in V} c_v$ nodes. As no node has available capacity, all remaining trees in the resulting forest are shallow trees, and $K \times n - (\overline{k} + \sum_{v \in V} c_v)$ shallow trees are needed. Thus,

$$|F| = \overline{k} + K \times n - (\overline{k} + \sum_{v \in V} c_v) = K \times n - \sum_{v \in V} c_v \tag{7}$$

In short, the forest built by Algorithm 1 has minimum cardinality and spans each node exactly K times. □

At each step of Algorithm 1, it takes $\mathcal{O}(n)$ time to find the node with the biggest capacity and there are at most $K \times n$ steps, thus the overall time complexity is $\mathcal{O}(Kn^2)$.

3.2 Bounded Height Model

We now take into account both the delay and the reliability constraints. We suppose that the delay and reliability are identical over all links, i.e., $\forall e \in E, d_e = d > 0, p_e = p < 1$. Let $h(T)$ be the height of T, the delay height constraint and the reliability constraint in equation (3) are thus simplified into one joint tree height constraint, i.e.

$$\forall T \in F, h(T) \leq H = \min\{\lfloor \frac{D_h}{d} \rfloor, \lfloor \frac{\log(P_h)}{\log(p)} \rfloor\} \tag{8}$$

This is to say that the number of hop counts from the root to each node is bounded by H in any tree. When $H = 0$, each tree contains only a node, and thus Kn trees are required. When H is bigger than $n-1$, the unbounded height algorithm that we previously presented can be used. For all other heights, we propose a time-efficient heuristic algorithm, detailed in Algorithm 2. We use $h(T_k)$ for the height of tree T_k and $h_{T_k}(w)$ for the depth of a node w in T_k. In the pseudocode, V_K stores the set of nodes not yet spanned in K trees, and V_k^+ stores the nodes via which a node can be added to T_k without violating the height and capacity constraint.

Following the same idea as in Algorithm 1, our algorithm consists of two phases. The first one exhausts node capacity and the second one creates additional shallow trees if necessary. For every tree, we select the root as the node having the biggest available capacity, then we attach other nodes to the tree in a breadth-first manner starting from the nodes having biggest capacity until either the height of the tree reaches $H-1$ or no more unattached node has capacity. Note that a node can be attached although it has already been spanned K times. At the end of each tree, we add the nodes that are still in V_K starting from the nodes with the smallest capacity until until either all nodes are spanned, or all capacities of nodes with height smaller than H are exhausted, i.e., $V_k^+ = \emptyset$. In the second phase, we build the remaining shallow trees if V_K is not empty. At most $K \times n$ trees may be computed and each tree contains at most n nodes. It also takes $O(n)$ time to find the node with the maximum or minimum capacity. Thus the time complexity is $O(Kn^3)$.

Algorithm 2: Bounded height model

Input : Complete graph $G = (V, E)$, capacity c_v of $v \in V$, number of packets K, and height H

Output: A height-bounded forest F

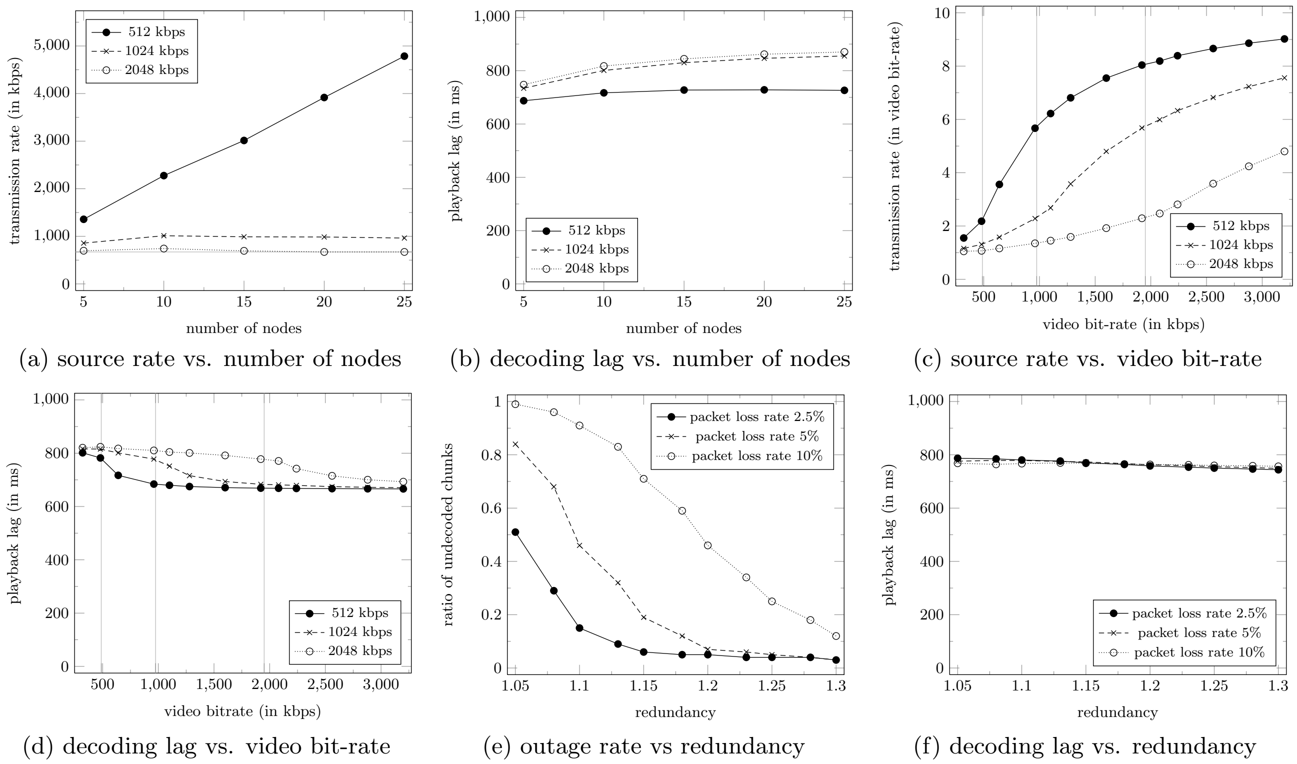
$k \leftarrow 1$;
$F \leftarrow \emptyset$;
$V_K \leftarrow V$;
for $v \in V$ **do** $c'_v = c_v$;
$V_c \leftarrow \{v : c'_v > 0,\ v \in V\}$;
while $V_c \neq \emptyset$ **and** $V_K \neq \emptyset$ **do**
  find node $r \in V_c$ **s.t.** $c'_r = \max\{c'_v : v \in V_c\}$;
  $V_k \leftarrow \{r\}$;
  $E_k \leftarrow \emptyset$;
  $T_k \leftarrow (V_k, E_k)$;
  if $r \in V_K$ **and** r *is spanned* K *times* **then**
    $V_K \leftarrow V_K \setminus \{r\}$;
  $V_k^+ \leftarrow \{v : v \in V_k \cap V_c$ **and** $h_{T_k}(v) < H\}$;
  while $\overline{V_k} \cap V_K \neq \emptyset$ **and** $V_k^+ \neq \emptyset$ **do**
    find $w \in V_k^+$ **s.t.** $h_{T_k}(w) = \min\{h_{T_k}(v) : v \in V_k^+\}$;
    if $h_{T_k}(w) < H - 1$ **and** $\overline{V_k} \cap V_c \neq \emptyset$ **then**
      find $u \in \overline{V_k}$ **s.t.** $c'_u = \max\{c'_v : v \in \overline{V_k}\}$;
    else
      find $u \in \overline{V_k} \cap V_K$ **s.t.** $c'_u = \min\{c'_v : v \in \overline{V_k} \cap V_K\}$;
    $V_k \leftarrow V_k \cup \{u\}$;
    $E_k \leftarrow E_k \cup \{(w, u)\}$;
    $T_k \leftarrow (V_k, E_k)$;
    $c'_w \leftarrow c'_w - 1$;
    if $c'_w = 0$ **then**
      $V_c \leftarrow V_c \setminus \{w\}$;
    $V_k^+ \leftarrow \{v : v \in V_k \cap V_c$ **and** $h_{T_k}(v) < H\}$;
    if $u \in V_K$ **and** u *is spanned* K *times* **then**
      $V_K \leftarrow V_K \setminus \{u\}$;
  $F \leftarrow F \cup \{T_k\}$;
  $k \leftarrow k + 1$;
for $v \in V_K$ **do**
  $k_v \leftarrow$ the number of trees v belongs to;
  add $K - k_v$ trees $(\{v\}, \emptyset)$ to F

4. SIMULATIONS

We use the ns-2 network simulator to evaluate the performance of our heuristic algorithm (Algorithm 2).

4.1 Video and Rateless Code Settings

The video was compressed with the H.264 coder at various bitrates ranging from 320 kbps to 3.2 Mbps. The resulting bitstream was partitioned into chunks where each chunk corresponded to one Group of Pictures (GOP). Each chunk had a playback duration of 0.5 s. The source applies rateless coding on each chunk and sends the encoded symbols in successive UDP packets of size 1,000 bytes. The size of



(a) source rate vs. number of nodes (b) decoding lag vs. number of nodes (c) source rate vs. video bit-rate

(d) decoding lag vs. video bit-rate (e) outage rate vs redundancy (f) decoding lag vs. redundancy

Figure 2: Results of our evaluation

an encoded symbol was one byte. For rateless coding, the Raptor code model proposed in [10] was used. With this model, a redundancy of 5%, gives a very high probability of successful decoding [10]. Thus, for a default video bitrate of 640 kbps, a node needs about 672 kbps to recover the original chunk.

4.2 Nodes Settings

The number of nodes varied from 5 to 25 (by default 10). Since there is no existing solution like the one we propose, it is hard to set the support capacity of the node. Our goal is to explore various capacity distributions and to observe the impact on the performance. Thus, our simulations can guide CDNs to set the support capacity. We used a log-normal distribution. The first parameter is the mean upload capacity, which was chosen in the set {512 kbps, 1,024 kbps, 2,048 kbps}. The second parameter is σ, which we abusively call *capacity heterogeneity*. We used three values for σ: $\{0.1, 0.8, 1.5\}$. The first value $\sigma = 0.1$ corresponds to a homogeneous configuration where the capacity of all nodes is close to the mean. At the other end, the value $\sigma = 1.5$ generates heterogeneous configurations where a few nodes have a high capacity while the remaining ones have a low capacity. As we have observed a low impact of the capacity heterogeneity, we do not present curves with distinct heterogeneities.

4.3 Network and Algorithm Settings

The core network of the CDN is a homogeneous network where most equipments and links are standard. Thus, we assume that both RTT and the packet loss probability are identical on every link (respectively set to 50 ms, and, for the packet loss probability to zero in Figs. 2(a) to 2(d) and 0.025, 0.05, or 0.1 in Fig. 2(e) and 2(f)). The Raptor code redundancy was set to 5% in Figs. 2(a) to 2(d). We used three different height bounds, $H \in \{1, 2, 3\}$. As for the capacity heterogeneity, we do not present results for each tree height bound separately because this parameter had a low impact on the overall performance. Therefore, every point on a curve corresponds to the average value of ten runs for three different tree height bounds and three different capacity heterogeneities.

4.4 Discussion

Figs. 2(a) and 2(c) show the required transmission rate at the source to guarantee that all nodes are able to decode the video chunk for various numbers of nodes and video bit-rates, respectively. Figs. 2(b) and 2(d) show the average lag between the time at which the source sends the first packet of a video chunk and the time at which a node is able to decode the video chunk again for various numbers of nodes and video bit-rates, respectively. Gray vertical lines in Figs. 2(c) and 2(d) indicate the video bit-rate for which the support network is just-provisioned for our different mean upload capacities. In the following, we use these four figures to highlight that our proposal has two distinct behaviors regarding the provisioning of the support network.

When the support network is under-provisioned, upload resources are exhausted before all nodes are spanned K times, so the source has to compensate for the lack of upload capacity by directly sending the last packets of the video chunk in shallow trees. On one hand, these shallow trees are costly in terms of source transmission rate because each one transmits one packet to only one node. On the other hand, as no relay is necessary, the decoding lag is lower. The more under-provisioned is the support network, the larger is the number of shallow trees in the forest. Therefore in the extreme case (video bit-rate 3.2 Mbps for mean capacity 512 kbps), the playback delay is stable around the minimum transmission delay, and the source transmission rate is roughly equal to the number of nodes times the video bit-rate.

When the support network is over-provisioned, the source does not need to build shallow trees. Our algorithm ensures that the forest contains compact and well-balanced trees. The decoding lag is less than 400 milliseconds, even when 25 nodes should be served. Our algorithm also succeeds in fully utilizing the resources of the nodes, as demonstrated by the source transmission rate, which is stable around the optimal lower bound in Fig. 2(a) for a mean capacity 2,048 kbps and never requires the transmission of more than 2.5 times the video bit-rate in every over-provisioned configuration. Our ability to serve a large number of nodes in a short delay makes the case for our proposal.

We then evaluate the performances of our algorithm when UDP packets can be lost during transmission. Rateless codes allow the diffusion of a potentially infinite number of distinct symbols until all nodes can decode the chunk. In our context, we fix a redundancy index, which indicates the bit-rate of the encoded stream including the redundancy in comparison to the video bit-rate (*e.g.*, a redundancy of 1.05 for a video bit-rate of 1,000 kbps means a stream bit-rate of 1,050 kbps). Fig. 2(e) shows the *outage rate*, which is the ratio of chunks that are not decoded to the overall number of transmitted chunks. Fig. 2(f) shows the decoding lag of the successfully decoded chunks.

Since the packets may be relayed many times before reaching a node, packet loss has a dramatic impact on successful chunk decoding. With low redundancy (less than 1.10), the outage rate is intolerable when the network is faulty (almost no chunk successfully decoded when 10% of transmissions are faulty). However, the benefits of using rateless codes become clearer for stream bit-rates that include a reasonable redundancy (between 1.15 and 1.20). Here, the outage rate falls below 0.2 in the most realistic scenarios. Finally, as shown in Fig. 2(f), neither the redundancy nor packet loss affect the decoding lag. When the redundancy increases, the number of shallow trees increases, and the decoding lag paradoxically decreases.

5. CONCLUSION AND FUTURE WORK

In this paper, we try to prevent one equipment (source) to throttle a whole CDN infrastructure in the context of live stream delivery. We propose the notion of support network where the equipments that receive the video chunks from the source cooperate in order to compensate for its missing capacity. We study one solution for this support network, which consists in leveraging on rateless codes. We modeled the problem as finding a diffusion forest with guaranteed end-to-end delay and bounded packet loss probability such that the total number of trees in the forest is minimized. We presented an optimal solution when both the delay and packet loss rate are not bounded. When the delay and the reliability are bounded and identical over all links, we proposed a heuristic algorithm.

The numerical results demonstrate the performance of the heuristic algorithm in diverse system configurations. Future work will include a real implementation in a real CDN infrastructure in order to validate the practical interest of our solution on the overall CDN.

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).

6. REFERENCES

[1] M. Adler, R. K. Sitaraman, and H. Venkataramani. Algorithms for optimizing the bandwidth cost of content delivery. *Computer Networks*, 55(18):4007–4020, 2011.

[2] J. M. Almeida, D. L. Eager, M. K. Vernon, and S. J. Wright. Minimizing delivery cost in scalable streaming content distribution systems. *IEEE Trans. Multimedia*, 6(2):356–365, 2004.

[3] K. Andreev, B. Maggs, A. Meyerson, J. Saks, and R. Sitaraman. Algorithms for constructing overlay networks for live streaming. *CoRR 1109.4114*, 2011.

[4] K. Andreev, B. M. Maggs, A. Meyerson, and R. K. Sitaraman. Designing overlay multicast networks for streaming. In *Proc. ACM SPAA*, pages 149–158, 2003.

[5] R. Gibbs. A new approach to publishing and caching video. Technical report, Alcatel-Lucent, Jan. 2012.

[6] M. Grangetto, R. Gaeta, and M. Sereno. Rateless codes network coding for simple and efficient P2P video streaming. In *Proc. of ICME*, pages 1500–1503, Jul. 2009.

[7] S. Higginbotham. Smart TVs cause a net neutrality debate in S. Korea. Giga OM, Feb. 2012.

[8] C. Liu, I. Bouazizi, M. M. Hannuksela, and M. Gabbouj. Rate adaptation for dynamic adaptive streaming over HTTP in content distribution network. *Signal Processing: Image Communication*, 27(4):288–311, 2012.

[9] M. Luby. LT Codes. In *Proc. of FOCS*, pages 271–280, 2002.

[10] M. Luby, T. Gasiba, T. Stockhammer, and M. Watson. Reliable multimedia download delivery in cellular broadcast networks. *IEEE Trans. Broadcast.*, 53(1):235–246, 2007.

[11] J. Ni and D. H. K. Tsang. Large-scale cooperative caching and application-level multicast in multimedia content delivery networks. *IEEE Commun. Mag.*, 43(5):98–105, May 2005.

[12] A. Shokrollahi. Raptor codes. *IEEE Trans. Inf. Theory*, 52(6):2551–2567, 2006.

[13] T. Siglin. Super Bowl Streaming Fail. Streaming Media, Feb. 2012.

[14] N. Thomos and P. Frossard. Network coding of rateless video in streaming overlays. *IEEE Trans. Circuits Syst. Video Technol.*, 20(12):1834–1847, 2010.

[15] C. Wu and B. Li. rstream: Resilient and optimal peer-to-peer streaming with rateless codes. *IEEE Trans. Parallel Distrib. Syst.*, 19(1):77–92, 2008.

Evaluating P2P Live Streaming Systems: the CNG Case*

Shakeel Ahmad[1] Christos Bouras[2] Eliya Buyukkaya[3] Raouf Hamzaoui[1]
Vaggelis Kapoulas[2] Andreas Papazois[2] Alex Shani[4] Gwendal Simon[3]

[1]Department of Engineering, De Montfort University, Leicester, UK
sahmad@dmu.ac.uk, rhamzaoui@dmu.ac.uk
[2]Computer Technology Institute & Press "Diophantus", Patras, Greece
bouras@cti.gr, kapoulas@cti.gr, papazois@ceid.upatras.gr
[3]Institut Telecom, Telecom Bretagne, France
eliya.buyukkaya@telecom-bretagne.eu, gwendal.simon@telecom-bretagne.eu
[4]Exent Technologies, Petach-Tikva, Israel
ashani@exent.com

Abstract

Many peer-to-peer (P2P) systems have been proposed for the provision of scalable live video streaming services over the Internet. While the literature contains surveys of the architectures of these systems, there is a lack of work on methodologies for their evaluation. We identify the main issues in the evaluation of P2P live streaming systems and use the Community Network Game (CNG) project as an example to illustrate them. The evaluation of the P2P system consists of two phases: a laboratory one using the ns-2 network simulator and an online field test with Massively Multiplayer Online Games (MMOG) players.

1. Introduction

Traditional client-server video streaming systems have critical issues of high cost and poor scalability. P2P networking exploits the upload bandwidth, computing power and storage space of the end users to reduce the burden on the servers and has been shown to be cost effective and easy to deploy. We identify the main issues in the evaluation of P2P live video streaming systems and use the CNG project as an example to illustrate them. The CNG project (`http://www.cng-project.eu/`) is an EU-funded research project that is focused on applying new network technologies to support community activities over highly interactive centrally managed MMOGs. CNG enhances collaborative activities between online gamers and develops new tools for the generation, distribution and insertion of User Generated Content (UGC) into existing MMOGs. It allows the addition of new engaging community services without changing the game code and without adding new processing or network loads to the MMOG servers. In particular, CNG proposes a P2P live video system to stream screen-captured video of MMOGs.

This paper presents the procedure that will be followed to evaluate the P2P live streaming system. Two phases are planned: a laboratory "offline" one using simulation software and an online one based on a real deployment. The online evaluation will be done by real gamers who will provide feedback through questionnaires. The performance of the system will be monitored by software, which will collect and provide useful information for further analysis.

The remainder of the paper is as follows. Section 2 provides an overview of the CNG P2P live streaming system. The plans for CNG laboratory experiments and online evaluation are presented in Section 3 and 4 respectively.

2. CNG P2P Live System

Allowing MMOG players to share their live game play with other players can have many useful applications. For example, skilled players can showcase their game to a large audience. Currently, the only platform that offers this service is Xfire (`http://www.xfire.com/`). Xfire captures the video of the game from the screen and sends it to a central server which broadcasts it live. However, this solution, which relies on central servers is expensive due to bandwidth and maintenance costs. To address these limitations, the CNG project proposes to use a P2P system. While many P2P live video systems have been developed, none of

*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).

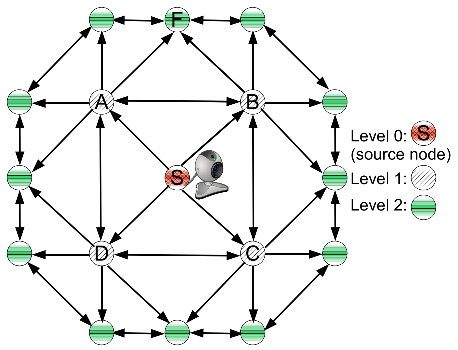


Figure 1. P2P topology.

them is suitable for the specific requirements of MMOGs:

- MMOG friendliness. The game experience should not be harmed by the P2P streaming. Thus, data communication with the MMOG game server must be given higher priority.

- Efficient management of multiple P2P overlays. Many MMOG players may simultaneously emit live streams, so the P2P overlay consists of many concurrent P2P overlays. A peer cannot participate in all P2P overlays because some of its resources will be used in every overlay it belongs to. The challenge for a user is to adequately allocate its physical resources, including upload and download bandwidth. These resources are limited, so they have to be shared carefully.

- Live video should be delivered at about the same time for all peers at the same "level". Moreover, peers in a higher level should in general be able to watch the video before those in a lower level. A level can be a priority class in a multi-tiered premium service. Alternatively, a level can be defined as the set of MMOG players that are in the same region of the virtual world.

In the following, we describe the CNG P2P system. The video of the game is captured in real time from the computer screen of the source and compressed. The resulting bitstream is partitioned into a sequence of source blocks, each of which corresponds to one Group of Pictures (GOP).

A mesh topology is used for the P2P network. This mesh is a directed graph $G = (V, E)$ where V is the set of peers, and (x, y) is in E if x may directly send packets to y. Peers are organized in levels. Level 0 contains the source. Level 1 consists of all peers that are direct successors of the source. In general, level k consists of all peers that are direct successors of level $k-1$ peers but are not in level $k-1$.

The UDP protocol is used as the transport protocol. The source applies rateless coding on each source block and sends the resulting encoded symbols in successive packets to level-1 peers until it receives an acknowledgment or a timeout occurs.

Packets are sent according to a scheduling strategy. The strategy specifies the maximum number of encoded packets n that can be sent by the source, the time at which a packet is sent, and a hierarchical forwarding scheme.

An example of a scheduling strategy for the P2P network of Fig. 1 is as follows. Packet 1 should be sent at time t_1 to A, which should forward it to B and D. Packet 2 should be sent at time t_2 to B, which should forward it to C. Packet 3 should be sent at time t_3 to C, which should forward it to B and D. Peer D should forward it further to A. Packet 4 should be sent at time t_4 to D, which should forward it to A and C. Peer C should forward it further to B.

Level-1 peers forward packets that are directly received from the source to their adjacent level-2 peers.

When a level-1 peer completes the decoding of a source block, it sends an acknowledgment to the source. Then it applies rateless coding on the decoded source block and starts acting as a source for level-2 peers that are its direct successors. The same procedure applies to peers at the next levels.

3. Laboratory Experiments

In the laboratory experiments, the CNG network solution with be evaluated with simulation software.

3.1. Metrics

To evaluate the performance of P2P video streaming systems, the following metrics were used in the literature: (1) Start-up delay: delay between the time a user joins a P2P system and the time it starts playing back the video [1] (2) Playback lag: time difference between the playback position of the source and that of the receiving peer [2] (3) Failure rate: probability that a user is rejected when it tries to join the system [1] (4) Continuity index: ratio of the number of video blocks that are available at their due playback time to the number of blocks that should have been played back by that time [1] (5) Peak Signal to Noise Ratio (PSNR) [3]. The PSNR is a standard video quality metric computed as: $PSNR(dB) = 10 \log_{10} \frac{255^2}{MSE}$ where MSE is the mean squared error between the original frame and the reconstructed frame (6) Percentage Degraded Video Duration (PDVD) [4]. The PDVD is the percentage of received frames whose PSNR is more than 2 dB worse than the PSNR of the corresponding encoded frames.

Both the PSNR and PDVD can easily be computed in the lab experiments as all required videos (original, encoded, received) are available.

In addition to measuring mean values of these metrics, we plan to divide CNG users into classes according to bandwidth, and measure minimum, maximum and variance values of the metrics for each class.

3.2. Examined Aspects

Scalability: Scalable systems are characterized by the property that the usage of resources is independent of the size of the system. Simulations for P2P live video streaming systems rarely consider more than several thousand simultaneous peers. Indeed, it is assumed that if the system scales at this stage, it is highly likely that it will scale to more peers. For the simulation of the CNG P2P video system, we plan to consider at least 2,000 peers.

Heterogeneity: Measurements show that P2P systems are characterized by a very high diversity of participating peers [5]. The variability of the *upload capacity* is considered as the major challenge in terms of heterogeneity, because it requires specific strategies in order to leverage the high capacity on some peers, and serve the peers with low capacity [1]. In [6], the average upload capacity of peers using BitTorrent has been found to be 180 kbps, while [7] shows an average upload capacity of 150 kbps. Two empirical distributions of peer upload capacities, one of broadband hosts (`http://www.dslreports.com/archive`) and one of BitTorrent hosts [8] are similar and are well modelled by a log-normal distribution [9]. Taking into account the above studies, the upload capacity in our experiments will follow a log-normal distribution with parameters μ in $\{150, 500, 1000\}$ and σ in $\{0.2, 0.9, 1.4\}$.

Churn: Based on the results from [10] and [7], we propose to use an exponential distribution of Time to Live (TTL) to model the churn rate. The parameter of this exponential distribution should result in approximately one tenth of the peer population refreshed at every simulation unit (which can be fixed at 1 hour). This setting, which is higher than what has been measured in [10], represents a system with a high churn. We will also evaluate our system under lower churn rates.

3.3. Simulation Tools

We considered several simulators including: ns-2, ns-3, P2PSim, Overlay Weaver, PeerSim, PlanetSim, Neurogrid, Query-Cycle, and Narses. We have assessed the available simulators based on several criteria, like simulator architecture, usability, scalability, statistics, underlying network simulation, and system limitations. We also considered the use of a testbed, like Planetlab. However, testbeds do not permit complete control of the system (e.g., for churn purposes) and, additionally, the scale of the experiments is severely limited. Taking into account the above, we decided to use the ns-2 simulator for CNG laboratory experiments.

3.4. Simulation Environment & Settings

A network game model for Counter-Strike is proposed in [11]. It is reported that 3–4% of all packets in a backbone could be associated with only 6 popular games [12]. One major concern is the upstream bandwidth. The peak percentage of traffic contributed by clients playing Counter-Strike to the total UDP traffic in the upstream direction can go up to 12% [13]. The work in [11] provides a simple traffic model for fast action multiplayer games. The game traffic model consists of only two independent modules, the client traffic model and the server traffic model with a burst size equal to the number of clients participating. In [14], the Extreme Value distribution has been identified to fit best for Quake traffic and other measurements have shown that newer MMOGs have bandwidth requirements that surpass those of older games [15].

It is important to identify user behavior with respect to network gaming. The study in [13] shows the percentages of subscribers playing Counter-Strike in various markets and identifies the trends in user behavior versus the day of the week. The common trend for all the markets is that this proportion increases as the weekend approaches, peaking on Friday, Saturday or Sunday. Also, there is a period of time in a day when there are very few to no users playing games. The exact hours vary depending on the market, indicating varying user behavior. The authors of [16] have filtered and analyzed MMOG traffic of the three-day long passive measurement, which contained about 200 World of Warcraft (WoW) flows and 100 other MMOG flows. Finally, [17] analyses a 1,356-million-packet trace from a sizable MMOG called ShenZhou Online.

3.5. Comparison to Similar Systems

Comparing the CNG system to state of the art commercial systems such as PPlive (`http://www.pptv.com/`) would present a major challenge as an implementation of these systems in ns-2 is not available. Moreover, the existing systems have not been designed for the CNG envisioned application. As an alternative, we propose to compare the CNG system to two related systems [18, 19] which although not designed for an MMOG environment can be implemented in reasonable time by making appropriate changes in the CNG system implementation.

4. Online Evaluation

In the online evaluation, MMOG players will be asked to answer an online questionnaire to describe their experience. In addition, the performance of the P2P network will be monitored using online tools.

Evaluation tools will be used for monitoring the activities and the traffic generated by the CNG system. Packet analyzers like Wireshark (`http://www.wireshark.org/`) are commonly used for traffic monitoring. In [20], Wireshark is used for analyzing and modeling the traffic generated by WoW. In [15], Wireshark was running on a computer along with Second Life, capturing game packets while filtering out irrelevant packets. On the other hand, built-in packet sniffers are useful for capturing the traffic and investigating protocols or applications in a non-intrusive way. When the source code of the application is available, traffic analysis can be done by adding proper logging and profiling functions to the source code [21].

In the online evaluation of the CNG network solution, Wireshark will be used to monitor the MMOG and other background traffic. On the other hand, the P2P network will be monitored through modules that will be developed in order to track user activities as well as the generated traffic.

For the online evaluation, a sample of 100-200 players will be recruited through forum posts and direct invitations to MMOG communities. An online user guide will be made available to help the players install and use the CNG tool. A number of evaluation measures will be used before, during and after the evaluation trial. The evaluations will be conducted online, allowing the project to obtain fast feedback. Participants will then be requested to continue their MMOG play and to augment it where they wish to with the CNG tools available to them. The usage period will be in the region of 4 weeks. Online players will be able to provide their feedback through a questionnaire where they will describe their experience with the CNG system, indicate any technical problem they encountered and suggest any further enhancement for the system.

5. Conclusion & Future Work

We have presented the plannning for the evaluation of the CNG P2P live streaming system. The process consists of two phases: laboratory experiments and online evaluation of the integrated system. The paper also includes a survey on the state of art for the evaluation of similar systems.

The next step of this work is the execution of the above plans. Our goal is that the execution of CNG evaluation will not only lead to significant results on the CNG system performance, but it can also introduce innovative ways of working for the evaluation of similar systems.

References

[1] B. Li, S. Xie, Y. Qu, G. Keung, C. Lin, J. Liu, and X. Zhang. Inside the new coolstreaming: Principles, measurements and performance implications. In *Proc. INFOCOM 2008*, pages 1031 –1039, Apr. 2008.

[2] D. Purandare and R. Guha. BEAM: An efficient peer to peer media streaming framework. In *Proc. Local Computer Networks '06*, pages 513–514, 2006.

[3] V. Padmanabhan, H. Wang, and P. Chou. Resilient peer-to-peer streaming. In *Proc. 11th IEEE International Conference on Network Protocols*, pages 16 – 27, Nov. 2003.

[4] 3GPP. Technical specification group services and system aspects; video codec performance (rel. 9), TR 26.902 v9.0.0. Technical report, 2009.

[5] J. Sacha. *Exploiting heterogeneity in peer-to-peer systems using gradient topologies*. PhD thesis, University of Dublin, Trinity College, 2009.

[6] O. Luzzatti. P2P live streaming for the masses - deployment and results. In *Proc. P2P-TV '07*, 2009.

[7] S. Khirman and V. K. Gurbani. Swarm analysis: 7 days of two popular swarms. In *Proc. P2PRG*, pages 26–31, Jul. 2009.

[8] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani. Do Incentives Build Robustness in BitTorrent. In *Proc. NSDI'07*, 2007.

[9] A. Bharambe, J. R. Douceur, J. R. Lorch, T. Moscibroda, J. Pang, S. Seshan, and X. Zhuang. Donnybrook: enabling large-scale, high-speed, peer-to-peer games. In *Proc. SIGCOMM '08*, pages 389–400, 2008.

[10] H. Liang, R. N. Silva, W. T. Ooi, and M. Motani. Avatar mobility in user-created networked virtual worlds: measurements, analysis, and implications. *Multimedia Tools Appl.*, 45:163–190, Oct. 2009.

[11] J. Färber. Network game traffic modelling. In *Proc. NetGames '02*, pages 53–57, 2002.

[12] S. McCreary and K. Claffy. Trends in wide area ip traffic patterns - a view from ames internet exchange. In *ITC Specialist Seminar on IP Traffic Modeling, Measurement and Management*, pages 1–11, 2000.

[13] A. Sinha, K. Mitchell, and D. Medhi. Network game traffic: A broadband access perspective. *Comput. Netw.*, 49:71–83, Sep. 2005.

[14] M. S. Borella. Source models of network game traffic. *Computer Communications*, 23(4):403 – 410, 2000.

[15] J. Kinicki and M. Claypool. Traffic analysis of avatars in second life. In *Proc. NOSSDAV '08*, pages 69–74, 2008.

[16] G. Szabó, A. Veres, and S. Molnár. Effects of user behavior on mmorpg traffic. In *Proc. ICC'09*, pages 1531–1536, 2009.

[17] K.-T. Chen, P. Huang, C.-Y. Huang, and C.-L. Lei. Game traffic analysis: an MMORPG perspective. In *Proc. NOSSDAV '05*, pages 19–24, 2005.

[18] M. Grangetto, R. Gaeta, and M. Sereno. Reducing content distribution time in p2p based multicast using rateless codes. In *Proc. Italian Networking Workshop*, pages 1–12, Jan. 2009.

[19] C. Wu and B. Li. rStream: Resilient and optimal peer-to-peer streaming with rateless codes. *IEEE Trans. Parallel Distrib. Syst.*, 19:77–92, Jan. 2008.

[20] M. Suznjevic, M. Matijasevic, and O. Dobrijevic. Action specific massive multiplayer online role playing games traffic analysis: case study of world of warcraft. In *Proc. NetGames '08*, pages 106–107, 2008.

[21] W. Acosta and S. Chandra. Trace driven analysis of the long term evolution of gnutella peer-to-peer traffic. In *Proc. PAM'07*, pages 42–51, 2007.

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 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 per-

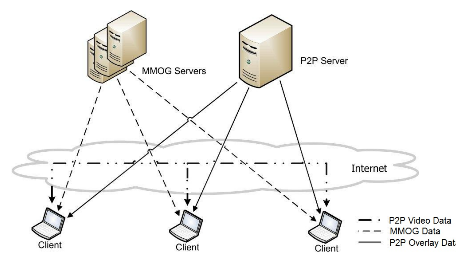


Fig. 1: System architecture.

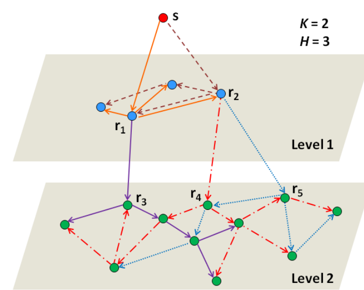


Fig. 2: Overlay example when $K = 2$ encoded packets are needed to successfully decode the video and the maximum height of any tree is constrained to be $H = 3$. There are two trees in Level 1 (higher level) rooted at r_1 and r_2 and three trees in Level 2 (lower level) rooted at r_3, r_4, and r_5. Every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

sistent 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 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.

Fig. 3: Snapshot of the CNG P2P system.

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/.