



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

► To cite this version:

Shakeel Ahmad, Christos Bouras, Raouf Hamzaoui, Andreas Papazois, Erez Perelman, et al.. The community network game project: enhancing collaborative activities in online games. NEM Summit, Oct 2010, Barcelone, Spain. hal-02367724

HAL Id: hal-02367724

<https://hal.science/hal-02367724>

Submitted on 18 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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 co-operative 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-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



Figure 1. Original MMOG screenshot.



Figure 2. IGIT-modified MMOG screenshot.

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

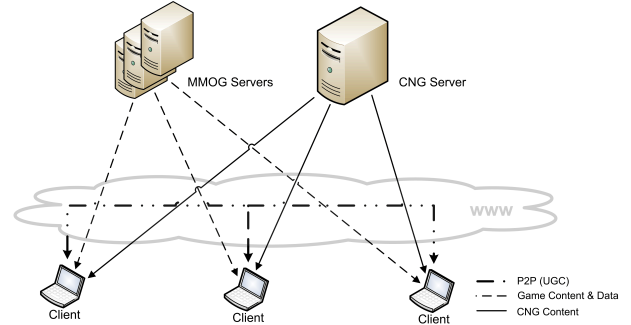


Figure 3. CNG architecture.

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.

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'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

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

Table I

GAME ADAPTATION TECHNOLOGIES IN FREERIDE GAMES (FRG), MASSIVE INCORPORATED (MI), PLAYXPERT (PX), XFIRE (XF), DOUBLE FUSION (DF). AN X MEANS THAT THE TECHNOLOGY IS AVAILABLE IN THE PRODUCT.

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).
- 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.

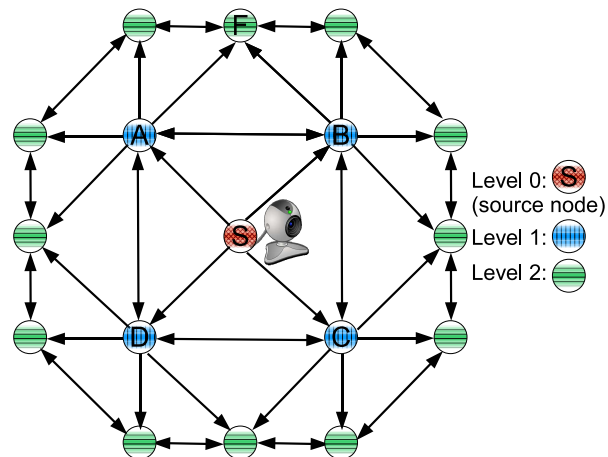


Figure 4. P2P topology. S is the source peer.

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 , $i = 1, 2, \dots, n$.

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

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 . Packet 5 should be sent at time t_5 to peer D , which should forward it to peer A . Peer A should forward it further to peer B .

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 $\rightarrow D(\rightarrow C)$. When peer A forwards this packet to peer D , it keeps only the information $\rightarrow C$.

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 IGT 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 Commission's Seventh Framework Programme (FP7, 2007-2013) under the grant agreement no. ICT-248175 (CNG project).

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.