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 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) 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

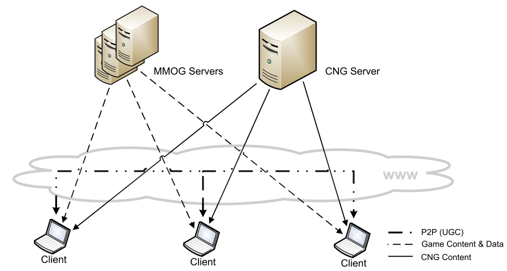

Figure 1. CNG architecture.

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.

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

Figure 2. Original MMOG screenshot.

Figure 3. IGIT-modified MMOG screenshot.

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.

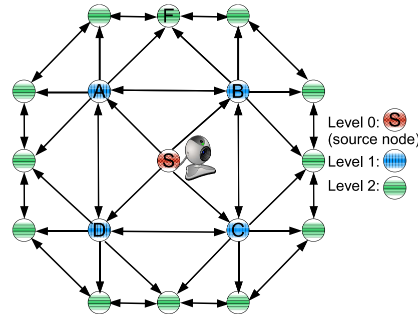

Figure 4. P2P topology. S is the source peer.

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.