[ZM04] *Michael Zink, Andreas Mauthe:* **P2P Streaming using Multiple Description Coded Video;** Euromicro 2004, Rennes, September 2004, S. 240-247.

P2P Streaming using Multiple Description Coded Video

Michael Zink
Computer Science Department
University of Massachusetts – Amherst
Amherst, MA 01003-4610
USA
zink@kom.tu-darmstadt.de

Andreas Mauthe
KOM – TU Darmstadt
Merckstr. 25, D 64283 Darmstadt
Germany
mauthe@kom.tu-darmstadt.de

Abstract

Today's peer-to-peer applications benefit from the fact that many users offer their resources (mostly in form of files). Those resources are mainly connected via relatively low-bandwidth, asymmetric access networks (such as ADSL or cable modems), which make it hard to realize the streaming of video data. Thus, audio visual content is usually downloaded and not streamed in today's Peer-to-Peer (P2P) systems. In order to provide streaming support it is necessary to take into account the asymmetric character of the up-load and download links. In this paper, we show that by making use of Multiple Description Coded (MDC) video and the fact that single descriptions can be sent from different pees,r, streaming in peer-to-peer applications is feasible. The paper discusses the different issues related to this topic. It explains MDC and compares it to Hierarchically Layered Encoded Video (HLEV). Further, the conditions under which MDC can be used for P2P streaming are discussed and it is shown how it can be deployed in a P2P environment.

1. Introduction

In recent years peer-to-peer (P2P) applications have become a very popular tool in the Internet, mostly used for file sharing applications. The idea was first more widely deployed within Napster [10] followed by many other systems such as Gnutella [5] E-Donkey [4], but also Chord [16], CAN [13], etc. Mostly files in popular formats (such as MP3) are being exchanged. P2P file sharing applications are suitable for file transfer since they are not negatively affected by the elastic traffic characteristics of the underlying infrastructure. For these applications no constant bandwidth that is equivalent to the data rate is required. In contrast, this is required for on-demand video streaming where data has to arrive at a certain point in time to guarantee the continuously good quality of the streamed content.

Realizing video streaming in environments where senders and receivers are mainly connected via asymmetric links (e.g. ADSL or cable modems) becomes even harder due to the bandwidth constraints of the up-link. Another limiting factor can be the restricted amount of resources at individual peers and the quite large storage space requirements of video objects.

In this paper, we propose a new technique for P2P networks that allows video streaming making use of Multiple Description Coding (MDC) [14] to overcome the constraints of asymmetric access networks while also considering the available resources at individual peers. MDC codes a video stream into two or more complementary descriptions, which can be streamed and decoded separately from each other. In the paper it is shown that the proposed technique will also increase the fault-tolerance of such a P2P streaming system.

2. Requirements and Application Environment

The environment most P2P applications operate in is characterized by the asymmetric properties of the access networks, i.e. the bandwidth available for the downlink is significantly higher than for the uplink. For instance a typical ADSL service (such as the one offered by Deutsche Telekom) offers a downlink capacity of 768 kBit/s and an uplink bandwidth of 128 kBit/s.

Therefore, a participant is able to receive at 6-times the rate s/he can send. Thus, in a P2P environment the uplink capacity is the limiting factor for sending data to others within the peer group. While in the case of file transfer this only affects the duration of retrieving content, in the case of streaming this also has impact on the quality and format of the video respectively audio stream.

Given a scenario as shown in Figure 1 a video could only be streamed from one sender (Peer 1) to the receiver (Peer 7) if the bandwidth requirements of the video stream are less or equal to 128 kBit/s. Although it is actually possible to stream video with such a low bit rate it will be only at a comparatively poor quality.

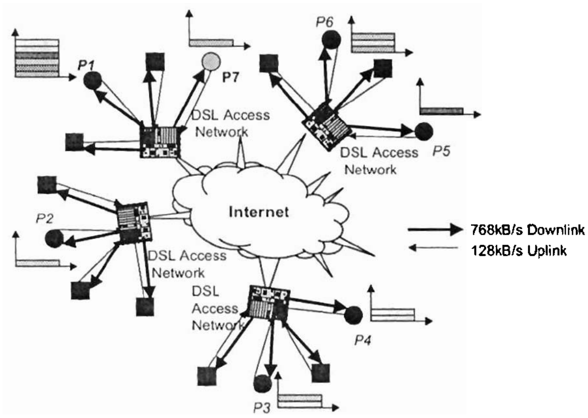

Figure 1. P2P Streaming restricted to uplink capacity

However, since the downlink of Peer 7 allows receiving the video at 768 kBit/s, it could be received in a much better quality. In order to exploit the full download bandwidth the stream should ideally be combined out of a number of sub-streams of which non should exceed the uplink capacity of a peer. This can be achieved by either using Hierarchically Layer Encoded Video (HLEV) or Multiple Description Coding Video (MDCV).

Figure 2 shows an example where six sub-streams are stored at different peers (Peer 1 up to Peer 6). To optimize the distributed streaming approach there would be ideally six sub-streams with a rate of 128 kBit/s each. A specific description can then be streamed from a particular sender; while all six streams can be transmitted in parallel on the down-link to the receiver (Peer 7) at a rate of768 kBit/s. This method obviously achieves a much better quality at the receiver than in the case when only one peer is used as sender.

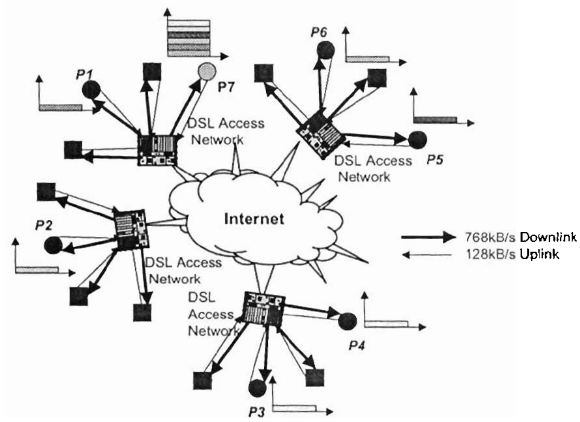

Figure 2. P2P Streaming using MDC video via asymmetric access links

However, in order to design such as system adequately the specific characteristics of P2P systems have to be taken into account. For instance compared to the approach presented in [1] where MDC streaming in CDN is investigated, the higher unreliability of the peers has to be considered. This might result in a dynamically changing set of sending instances. For instance, the system has to be able to cope with the inherent dynamicity of P2P systems (e.g. the unannounced leaving peers). A replacement strategy is required to ensure that in case a peer leaves the session it can be replace by another providing the equivalent data (if such a peer is available).

Another approach presented by Nguyen et al. [11] requires that the video object is always completely stored at the sender and assumes that the aggregate rate of each path between the senders and the receiver is greater than or equal to the rate of the complete video object. A P2P network as described above cannot meet these requirements. Here the different peers are dynamically becoming active and also may not have the complete video (i.e. all the different sub-streams).

3. Encoding Schemes: HLEV versus MDC

Sometime ago it has been recognized that video encoding schemes should not offer a static quality level but should much rather scale according to the user and environmental requirements. Current encoding schemes provide the necessary features and flexibility to deliver video at different rates. Within MPEG-2 for instance scalability is provided in four different modes (Spatial Scalability Mode, Temporal Scalability Mode, Data Partitioning Mode, SNR Mode) [9]. MPEG-4 allows video encoding rates from 5 kBit/s to 1 GBit/s [12]. The video formats can be progressive or interlaced and the resolution can vary from QCIF to 4K x 4K studio resolution. Further, with the *Fine Granularity Scalability* (FGS) Video Profile MPEG-4 also provides the possibility of fine grain layered video encoding. Hence, different (enhancement) layers can be stored and transmitted independently to the receivers where they are combined to form a video according to the capabilities of the receiver. Apart from FGS video the MPEG-4 standard also specifies a scalable Audio Profile.

The use of Hierarchically Layer Encoded Video requires that for decoding the base layer and all layers up-to the highest layer that provides a certain quality level are available. In contrast, with Multiple Description Coded Video any sub-set of the streams that make up the full quality stream are sufficient, i.e. it does not rely on a layered scheme where a lower layer is always required to allow an uninterrupted presentation of the video to decode the video stream. Both methods are viable alternatives for building a P2P based system for streaming video.

3.1. Hierarchically Layer Encoded Video

An encoding scheme that makes use of scalability is Hierarchically Layered Encoding (HLE). With HLE the video is split into one base layer and one or more enhancement layers. The base layer contains fundamental information and can be decoded without any additional information. Enhancement layers contain additional information, which increases the quality of the reconstructed video signal. In contrast to the base layer, enhancement layers are not independent from other layers. To reconstruct the information included in layer n all of the information of the lower layers (0, ..., n-1) are needed. If the base layer is missing, no video signal can be reconstructed at all. An introduction and overview about hierarchically layered encoding is given in more detail in [8].

The concept of scalable coding was first introduced in the MPEG-2 and H.263 [7] standards, which allow a two layer encoding (base layer plus one enhancement layer). This encoding scheme was extended with the H.263+ [7] standard, which allows several layers. In MPEG-4, the layer scheme of fine granularity scalability (FGS) [12] (which is basically a two layer scheme but allows a variable rate enhancement layer) is introduced. With FGS the bit stream can be truncated thus, it adapts to the available rate of the transmission channel.

3.2. Multiple Description Coding

MDC was originally developed at Bell Laboratories, having specifically circuit switched networks in mind. The idea was to transmit data over multiple (telephone) lines where in case of a line failure it should still be possible to decode the remaining data, though this would result in a reduced quality. This method was called channel splitting. The original bit stream is partitioned into different so-called descriptions of the one source. Receiving one or more of the source descriptions allows the source image to be reconstructed to a prescribed quality level [6]. MDC builds on Forward Error Concealment methods; i.e. the mechanisms to deal with errors (respectively reduced quality) are already implemented in the coding process. Therefore redundant information is encoded with each descriptor so that it is possible to decode each of the descriptors separately. This is called fractional repetition of core data. The protection of the core data can be higher (Unequal Error Protection) to ensure that a descriptor can be decoded. Any additional descriptor then enhances the presentation quality. Thus, descriptors carried in different streams do not build on each other and therefore do not need to be prioritized.

The challenge is how to divide the information in such a way that each descriptor contains a largely disjunctive set of information, while still maintaining the possibility to decode individual descriptor without reference to any other external information. In order to do this a considerable amount of basic information needs to be encoded redundantly with each descriptor. To reduce the required redundancy the coding is tailored towards a certain interval within which the different parameters can vary. The extent the interval covers depends on the chosen encoding scheme, kind and quantity of the source data and the amount of redundancy encoded into a descriptor. The interval can be dynamically adapted if a feedback scheme between the receiver and the source is implemented.

There are a number of different approaches being proposed for MDC, e.g. Polyphase Transformation and Correlating Transformation. All the work in this area is motivated by the requirement to decode descriptors independently of any other information transmitted in independent streams.

3.3. HLEV and MDC in P2P Systems

P2P Systems are characterized by the dynamic and autonomous behavior of the individual peers. The different components might often be running on computers with unpredictable uptimes. Not all the peers might have the storage capacity to hold a full-scale video and therefore chose to store only parts. In our scenario each of the peers can only contribute according to their uplink capability where it should be potentially possible to receive a stream exploiting the full bandwidth of the downlink.

MDC and HLEV have in common that the video object must not necessarily be stored completely at one peer but single descriptions (MDC) or layers (HLEV) can be stored at different peers. This feature makes both encoding techniques well suited for the requirement of P2P streaming. However, using a HLEV scheme as proposed in [19] may cause problems in such an environment. During the transmission a peer might leave unexpectedly. The worst-case scenario here would be if the peer sending the base layer goes off-line. This would lead to an interruption of the video playout at the receiver and, thus, be very annoying for the watching user. All the information received at the receiver at this point is useless until a new peer is found that can stream the base layer. However, each peer that holds the video should at least be able to provide the base layer. If any other layer fails at least the quality of the base layer and the lower layers can be provided. Moreover, if there would be dedicated peers holding the base layer they would very easily become hotspots in the system since they would always be required whereas the other layers are optional and may not be needed by users with lower quality requirements.

Since the MDC (in contrast to HLEV) does not rely on a layered scheme where a lower layer is always required to fully decode the stream, an uninterrupted presentation of the video can be ensured even if one or more peers fail during the transmission. Though, in this case the quality would be impaired due to the missing data of one or more descriptions.

In addition, MDC has the advantage that any subset of peers that store different descriptions of one video object can be used as senders while in the case of HLEV one member of the subset has to hold the base layer of the object to allow the presentation of the video at the receiver. Due to the fact that not all senders might stay on-line during the complete duration of the streaming session a mechanism should be provided that allows the receiver to choose a new sender for the description that is not delivered anymore in order to maximize the quality of the perception. As long as a different description to any currently received description is found, this will enhance the quality.

Although FGS has been specified bearing such requirements in mind in the case of P2P streaming FGS is not even as well suited as a HLE scheme with equal sized (in terms of bit rate) layers. To fully support FGS in a P2P streaming environment a new mechanism would be needed which would allow a combined streaming of the enhancement layer from several peers. This is the case because the rate of the enhancement layer might exceed the capacity of the uplink and, thus, could not be used within the above scenario.

4. Using MDC for P2P Streaming

From the above discussion it becomes clear that MDC should be better suited to stream video in a P2P system with asymmetric access links than HLEV. However, peer-to-peer streaming using MDC is not straightforward. There are a number of challenges related to the selection and transmission of descriptors of an MDC scheme in a P2P environment. The most crucial aspects are the assignment and selection of the right sender-to-description relationship and how to handle transport and signaling issues.

4.1. Sender-to-Description Assignment

One of the challenges of using MDC in a P2P system is for the receiver to decide which sender should stream which description. Note, there is no global instance that can assign the right set of senders according to the most optimal system state. Also, only the receiver knows about its capabilities (e.g. bandwidth and processing capacity) from which the number of description can be derived. Thus, each receiver has to choose the senders that satisfy its requirements best. A mechanism that solves this problem has to be designed in a way that increases the fault-tolerance of the system with respect to the unreliable characteristics of the peers.

Let us again assume the example from Figure 2. In this case there are 2 possibilities for the availability of descriptions in the P2P network:

More than 6 different descriptions:

If a location algorithm has found more than 6 different descriptions of a video object in the P2P network the receiver has to choose between senders that store identical descriptions. This decision should be made based on the path characteristics between the sender and receiver and the local characteristics of the sender itself (e.g. available information about the mean uptime of the sender but also if this sender already serves other peers and about its load condition). Assuming that peer ***A*** and peer ***B*** store the same descriptions and the path characteristics from peer ***A*** to the receiver are better (e.g. less hops and smaller delay) then peer ***A*** should be chosen as the sender. A conflict arises if the path characteristics are better of one peer (e.g. peer ***A***) but its local characteristics are worse than those of peer ***B***. For the maximum sender load a threshold should be defined. Once this threshold is reached no more streams should be requested from this sender. If the uptime characteristics are worse than this sender (i.e. ***A***) should be chosen since the path characteristics are better. In case ***A*** goes away the transmission can still be switched to ***B***. Hence, information about peers that are not chosen as a sender (e.g. peer ***B*** in the aforementioned example) should not be discarded, since this information can be used to increase the fault-tolerance of the system. If peer ***A*** then goes suddenly off-line the receiver can immediately decide to get this specific description from peer ***B***. Thus, the interval in which the quality of the video is reduced is kept as short as possible.

Less than or exactly 6 distinct descriptions:

This case is quite simple, since the receiver has not much choice. However, if any description is available more than once the receiver will apply the same selection criterion than in the above case. In order to increase the fault-tolerance the receiver should in any case either scan the P2P network or be informed by other peers about new different descriptions that become available (e.g. a peer becomes online during the streaming session and may distribute information about its capabilities, descriptors and status). This would help to increase the fault-tolerance and the quality of the delivered stream.

By considering the load situation of a peer in the local characteristics it is ensured that a specific peer is not being overloaded, i.e. it becomes a hot spot to a limited extend only. If the threshold load is reached other peers will be chosen to serve a request. However, this does not ensure a global optimum for the system since a peer receiver usually does not have full knowledge of all the peers that can provide a certain descriptor. In case a peer's threshold load is reached the peer should not be even considered as a sender even if no other alternative for this descriptor is available. Otherwise this peer can become overloaded and might drop out completely. The thresholds are not globally defined but depend on the capabilities of individual peers.

4.2. Transport and Signaling Issues

Video data is usually streamed via RTP/UDP where application-level framing (ALF) is defined by RTP profiles. This implies that profiles define how video data is packetized into RTP packets. All existing profiles assume that a video is always transported completely (e.g. all descriptions of an MDC video) from one sender to a receiver. This is obviously not the case in the described scenario for P2P streaming. In this case, every description will be transported as a single RTP/UDP stream. Thus, the receiver needs additional information in order to identify the description to which the data transported in the arriving RTP packet belongs to. In [18] we propose such a protocol extension for the transport of layer-encoded video using RTP. This protocol extension can also be applied for MDC. RTCP can be used for the signaling between each sender and the receiver without any modifications.

However, in any case a synchronization mechanism for streams with different traffic characteristics as perceived at the receiver is required. This includes the possibility of buffering data from different streams representing certain descriptors to be able to play them out together. The issue here is for how long the information has to be buffered since it does not make sense to fill all buffers equally for streams with different delay and jitter characteristics. Ideally these should be known so that the receiver can calculate the amount of buffer required for each stream. Here the buffer (B_i) for the stream (i) with the longest delay is just the maximum jitter (j_i_max) multiplied by the maximum rate (r_i_max) of this stream, i.e.:

$$B_i = j_i\_max * r_i\_max$$

For any other stream (k) the required buffer is taking into account the jitter (j_{k_max}) and rate (r_{k_max}) but also the difference of the delay ($d\__{max}$) between stream (i) and stream (k) multiplied by the maximum rate of stream (k) Thus, the buffer for stream (k) is calculated as follows

$$B_k = (j_{k\_max} * r_{k\_max})+((d_{i\_max}-d_{k\_max}) * r_{k\_max})$$

This only holds if all the streams are started at the same time. Buffer requirements can be reduced if the stream with the longest delay is started first, otherwise they might even have to be larger. A stream is considered failing if it exceeds its maximum jitter (j_k_max). In this case its buffer is empty at presentation time (t), respectively in the following period when any buffered information due to not being the stream with the longest delay, will run empty. In this case the receiver has to decide if the sender of this particular description is failing permanently and has to be replaced by another sender.

4.3. Implementation Issues

In such as system different concepts and components are coming together for which various platforms and frameworks already exist. Thus the system does not have to be implemented from scratch but can be based on existing modules. For the implementation of the proposed system different platforms and frameworks are chosen depending on the functionality required. The P2P functionality will be built on top of the JXTA middleware platform. Project JXTA [17] is an open platform designed for peer-to-peer (P2P) computing. Its goal is to develop basic building blocks and services to enable innovative applications for peer groups. JXTA provides a common set of open protocols and an open source reference implementation for developing peer-to-peer applications. The JXTA protocols standardize the manner in which peers:

- Discover each other
- Self-organize into peer groups
- Advertise and discover network services
- Communicate with each other
- Monitor each other

The JXTA protocols are designed to be independent of programming languages, and independent of transport protocols. A reference implementation is available in Java and additional implementation efforts are building compatible platforms in C/C++, Perl, and numerous other languages. They offer communication mechanisms implemented on top of TCP/IP, HTTP, and other transport protocols.

One common characteristic of peers in a P2P network is that they often exist on the edge of the regular network with unpredictable connectivity and, in many cases, variable network addresses. JXTA accommodates peers on the edge of the network by providing a system to uniquely address peers in a manner that is independent of traditional name services. Using JXTA IDs, peer addressing is independent of transport mechanisms and network addresses.

In order to consider the specific requirements of MDC video streaming places onto the P2P system a more complex overlay infrastructure is required. In contrast to other approaches multiple peers have to be found that are able to provide different descriptors. The work in this area builds on the work carried out within the Omicron framework that allows having structured overlays and peers that adopt different roles within the overlay according to their capabilities [2].

The actual streaming application will be build on top of the KOMSSYS streaming platform [3]. This platform was developed to allow researchers and developers to create new streaming mechanisms and make experiences with these newly created mechanisms on the basis of an actual implementation. Recently the KOMSYS was extended by a mechanism that allows the streaming of HLEV [15]. In the near future we plan to extend KOMSSYS by a mechanism that allows streaming of MDC video from several senders to one client. Due to the lack of MDC decoders that are available as open source we plan to create an abstract MDC format which has similar characteristics to MDC but will not transport real video data. We have already created such an abstracted format for HLEV and used it to perform measurements of HLEV streaming in the Internet [18].

5. Summary

P2P systems have become very popular for the sharing of content. Current systems are mainly based on file exchange. There are many issues, not only technical, related to this. However, within the research community it is currently being investigated if and how the peer-to-peer paradigm can be used in a much wider and different context. Compared to the client server approach P2P tends to be more robust and better extendable. Though, P2P mechanisms cannot be used for all purposes. At present it is being explored what the areas are in which P2P can be of advantage. The work presented in this paper is part of a number of research projects in which the different use cases for P2P are being studied. Therefore, we are as much interested in the way P2P streaming can work, as we are in the principles behind it. Understanding these principles is crucial since they enable us to deploy the results of this work in different areas. For instance P2P streaming mechanisms can be used within Content Distribution Networks (CDN) but also as part of professional content management and production. The different components of a video can for example also be the encoded video itself stored at one place and various audio tracks for different language versions stored at other locations. The principles investigated above can be applied to such a scenario as well as to the introduced example of Multiple Description Coded (MDC) video.

In this paper it has been demonstrated that using MDC video in P2P applications where receivers are mostly connected via asymmetric access networks is a viable solution to allow streaming of video as an alternative to file transfer. The major challenge here is to coordinate the streams from different senders in order to obtain the best possible quality of the video at the receiver. This issue of distributed streaming has been investigated by [1] and [11] but not within the scope of Multiple Description Coded Video and P2P networks that are based on asymmetric access technologies.

In addition, streaming of scalable video formats (e.g. MDC video) place new requirements onto P2P mechanisms such as data discovery. This is caused by the fact that a video object is not completely stored at one peer. In fact, the different parts of the video have to be retrieved from a number of peers that hold the respective information in order to make optimal us of asymmetric links. These peers might be geographically distributed over a wide area. There can be also multiple alternatives for the different descriptions. Since a central, coordinating instance is missing, the receiver have to select the most appropriate set of senders. This is in contrast to most P2P based file sharing application.

This paper presents an architecture that solves the problems mentioned above and, thus, provides a viable solution for video streaming in P2P overlay networks.

References

[1] J. Apostolopoulos, T. Wong, S. Wee, and D. Tan: "On multiple description streaming with content delivery networks", In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies* (INFOCOM'02), 2002.

[2] V. Darlagiannis, A. Mauthe, R. Steinmetz: "Overlay Design Mechanisms", *Journal of Network and Systems Management (Special Issue on Distributed Management)*, Kluwer Academic Publishers, accepted for publication in the 3rd quarter of 2004.

[3] C. Griwodz, M. Zink: „Dynamic Data Path Reconfiguration" In *International Workshop on Multimedia Middleware 2001*, Ottawa, Canada, October 2001

[4] eDonkey2000". http://www.edonkey2000.com, 2003.

[5] "Gnutella Protocol". http://www.clip2.com/GnutellaProtocol04.pdf, 2004.

[6] V. Goyal. Multiple Description Coding: "Compression Meets the Network". *IEEE Signal Processing Magazine*, 2001.

[7] ITU-T: "Video Coding for Low Bit Rate Communication", *International Standard, ITU-T Recommendation H.263*, 1995.

[8] J.-Y. Lee: "A Novel Approach to Multi-Layer Coding of Video for Playback Scalability", *PhD Thesis*, Graduate School Korea University, Seoul, Korea, 1999.

[9] A. Mauthe, P. Thomas: "Professional Content Management Systems – Handling Digital Media Assets", John Wiley & Sons *Ltd*, ISBN 0-479-85542-8, 2004.

[10] "Napster". http://www.napster.com, 2002.

[11] T. Nguyen and A. Zakhor: "Distributed video streaming with forward error correction", in *11th International Packet Video Workshop (PV2002)*, Pittsburgh, USA, 2002.

[12] F. Pereira, T. Ebrahimi (Editors): "The MPEG-4 Book", *IMSC Press Multimedia Series*, Prentice Hall, 2002.
[13] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker: "A scalable content-addressable network". In Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications, 2001.
[14] A. R. Reibman, H. Jafarkhani, Y. Wang, M. T. Orchard, and R. Puri: "Multiple Description Video Coding using Motion-compensated Temporal Prediction". *IEEE Transactions on Circuits and Systems for Video Technology*, 12(3), 2002.
[15] T. Spengler, Michael Zink, "Integration des SPEG Video Codec in die KOMSSYS Streaming Umgebung" (in German), *Technical Report TR-KOM-2004-2*, Darmstadt University of Technology, 2004
[16] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan. Chord: "A scalable peer-to-peer lookup service for internet applications". In *Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications*, 2001.
[17] Sun Microsystems: "JXTA v2.0 Protocol Specifications", *http://www.jxta.org*, 2003.
[18] M. Zink, C. Griwodz, J. Schmitt, and R. Steinmetz: "Scalable TCP-friendly Video Distribution for Heterogeneous Clients", in *Proceedings of SPIE/ACM Conference on Multimedia Computing and Networking (MMCN)*, Santa Clara, USA. 2003.
[19] M. Zink: "P2P Streaming using Hierarchically Encoded Layered Video", *Technical Report TR-KOM-2003-01*, Darmstadt University of Technology, 2003.