

Quantitative QoS-Mapping: A Unifying Approach

H. Knoche and H. de Meer

Dept. of Computer Science – Telecom. and Comp. Networks
University of Hamburg, Vogt-Kölln-Str. 30, D-22527 Hamburg
Tel: ++49 (40) 5494 2347 Fax: ++49 (40) 5494 2345
e-mail: [knoche,demeer]@ro2.informatik.uni-hamburg.de

Abstract

Distributed multimedia (MM) systems have to provide users with the ability to specify their performance requirements. Quality of service (QoS) parameters represent an adequate measure for the specification of time-dependent MM-data like audio or video streams. In order to guarantee the fulfillment of application requirements, a mapping onto the involved network and operating system resources has to be performed. This paper shows how QoS translation can be performed in distributed MM-systems. Parameter translations between abstraction layers including a terminology and the interdependencies between the parameters are presented. Furthermore, mapping stimuli that imply a modification of QoS parameters are identified and their respective effects are described.

Keywords

QoS mapping, QoS translation, QoS parameters, QoS mapping stimuli

1 INTRODUCTION

MM-applications are characterized by the capability of handling time-dependent data like video streams and time-independent data like traditional text. Streams consist of consecutive data units and the presentation of the stream must maintain the temporal relations between the consecutive units.

The qualitative and quantitative properties and requirements of MM-data can be expressed by means of QoS parameters. Imagine an application like a video conference. Users are geographically separated, but can audibly and visually communicate via their MM-workstations, which are interconnected by some sort of network. Before data can be sent over the network, it has to undergo various procedures like compression, segmentation, etc. These procedures consume a given QoS budget as they, for example, induce additional delay. Since the effects implied by the procedures have to be captured by the mapping algorithm, the respective procedures will be referred to as *mapping*

stimuli. All mapping stimuli have to be taken into account in order to guarantee a demanded QoS profile.

State of the art

[Ferrari, 1990] has presented translations of QoS parameters like delay, jitter, and reliability referring to constant size messages. Furthermore, he proposed translations for statistical bounds and fragmentation. [Moran *et al.*, 1992] investigated buffer dimensioning required for jitter smoothing. Throughput calculation is performed as a function of SDU size and time and includes variable bit-rate (VBR) effects. [Jung, 1996] contributed translations onto ATM network Network Performance parameters and additional knowledge on the mapping of reliability, when relations between the parameters are not one-to-one. [Damaskos *et al.*, 1994] translated SDU size, SDU inter-request interval, and SDU transmission delay onto ATM performance parameters. [Nahrstedt *et al.*, 1995] translated QoS parameters size, rate, delay, and loss rate from media quality to connection quality for constant size media samples.

The above mentioned approaches provide partial solutions to the mapping problem and concentrate mainly on the lower abstraction layers.

Although future multimedia applications are most likely to operate on data that entails VBR traffic, translations of QoS parameters for VBR services have hardly been addressed but rather been limited to constant bit-rate (CBR) services.

It is the main goal of this document to discuss stimuli for QoS-parameter modifications when mapping is performed across layers and to provide guidelines as to how QoS-parameters are affected by which stimulus. We provide a unifying approach that generalizes the mapping process, so it can be applied to every abstraction layer. This comprehensive study includes the dependencies and corresponding trade-offs between the QoS parameters.

In Sec. 2 we briefly summarize basic definitions of QoS parameters. Sec. 3 shows how mapping is performed across the abstraction layer hierarchy. Important mapping stimuli are identified that have an, usually adverse, effect on QoS parameters. A mapping function is defined that captures the implied effects accurately. Conclusions are given in Sec. 4.

2 QOS PARAMETERS

QoS mapping is regarded as the process of translating QoS-parameter bounds from layer to layer and, finally, to resources, e.g., buffers. In this section we will briefly present the definitions of the QoS-parameters that are subject of translation, a more thorough discussion is given by [De Meer *et al.*, 1997].

2.1 Delay

According to the ISO/OSI Reference Model, the delay of an SDU is defined as the time interval between the occurrence of a data.request at a layer's service

access point (SAP) at the sender and the occurrence of the corresponding data.indication at the peer SAP at the receiver [Danthine *et al.*, 1992]. It is assumed that data.request and data.indication occur instantaneously.

The required performance is characterized by the maximum delay D_{max} that should not be exceeded by the delay D_i of all SDUs i of a given set (or stream) on a given layer [Ferrari, 1990]:

$$D_i \leq D_{max}, \quad \forall i. \quad (1)$$

2.2 Jitter

The many existing definitions of jitter which are compared in more detail by [De Meer *et al.*, 1997], have in common to use jitter as a measure for temporal fluctuations of SDU delays. In this paper jitter is defined according to [Moran *et al.*, 1992]. Given the delay D_i of the i th SDU, the jitter J_i of that SDU is calculated by subtracting D_i from the maximum delay bound D_{max} that has been negotiated on:

$$J_i = D_{max} - D_i. \quad (2)$$

according to [Ferrari, 1990]: The bound for jitter can be given analogously to Eq. 1. With the chosen jitter Definition (2), J_{max} automatically implies the existence of a minimum delay bound D_{min} , since J_i can only be at its maximum when D_i is at its minimum:

$$D_{min} = D_{max} - J_{max}. \quad (3)$$

Deterministic jitter

Typically, networks dynamically impose delays on SDUs, which are not known in advance since they depend on load variations and queuing effects in intermediate nodes. In contrast, some procedures impose certain additional variable delays resulting in jitter effects, which can be accurately quantified from the beginning. To capture these effects, we would like to introduce the notion of *deterministic jitter*.

Given bandwidth W [$\frac{bits}{sec}$] and the size S_i [bits] of the i th video-frame, its transmission time t_i is computed by: $t_i = \frac{S_i}{W}$. If the maximum size S_{max} of all video-frames is known we can compute the jitter $j(trans)_i$ for the i th frame induced by the transmission. $j(trans)_i$ is the difference between the maximum transmission time t_{max} of the biggest frame and t_i .

2.3 Reliability

From a user point of view, reliability is a term which covers all kind of undesired effects. In a video conference reliability could express the probability, or some

bound hereof, that gaps in audio streams or distorted video frames do not occur. Effects related to excess data are not considered in this paper.

The required reliability can be expressed by the minimum probability P_{min} of a correct delivery of an SDU, i.e., neither loss nor bit error(s) do occur:

$$Prob(\text{correct delivery}) \geq P_{min}. \quad (4)$$

Sometimes it is more convenient to introduce an upper bound E_{max} for the probability of an erroneous data delivery:

$$E_{max} = 1 - P_{min}. \quad (5)$$

2.4 Throughput

When negotiating throughput, layer N and layer $N - 1$ agree on the (maximum) size of an SDU and on the minimum time that must pass between SDU-requests at layer $N - 1$. Therefore, throughput is defined via SDU size S , and the inter-request time T of the SDUs.

Bandwidth is defined as the number of SDU requests a service providing layer is able to accept with a negotiated SDU size in a given interval. The requested throughput of a service using layer is defined as the number of SDU requests with a negotiated SDU size in a given time interval.

CBR services usually imply constant SDU time intervals and sizes. In case of VBR services maximum and minimum sizes of SDUs are important for mapping purposes and should therefore be part of the negotiation.

More details of the different throughput parameter definitions are presented by [De Meer *et al.*, 1997].

3 THE MAPPING OF QOS-PARAMETERS

The mapping of QoS-parameters covered in this paper is limited to quantitative translation of the bounds of the QoS parameters. However, some parameters will have to be qualitatively translated across abstraction layers. An example for this more qualitative translation is given in Sec. 3.1 (c).

3.1 Prerequisites

The mappings are performed “top-down” from layer $N + 1$ to layer N . The reverse translations can in many cases be easily calculated. Ambiguities may occur when interlayer relations between QoS-parameters are not one-to-one.

(a) Notational Remarks

During the mapping of QoS-parameter bounds, we will stick to the following notation in order to avoid confusion with the different bounds involved.

On each layer N , the QoS-parameter bound Q^{N-1} , which is demanded from the service providing layer $N-1$, is calculated from the QoS-parameter bound Q^N . It is known how layer N affects the QoS-parameter bound Q^N . The QoS parameter budget spent or earned on layer N is accounted for through a pessimistic *local* bound \underline{Q}^N . Correspondingly, the bound that was originally demanded by the user, will be called the global bound.

To refer to delay d or to jitter j , which is caused as a side-effect by some procedure X , we use notation $d(X)$ or $j(X)$, correspondingly.

(b) Statistical Bounds

The translation of statistical bounds is omitted due to space limitations and can be found in detail in [Ferrari, 1990].

(c) Reliability

Assume layer N represents reliability with a single parameter (the upper bound of probability E_{max}^N for an erroneous delivery). If the probability of an erroneous delivery is represented by two bounds at layer $N-1$, a loss ratio bound L_{max}^{N-1} and an SDU error ratio bound H_{max}^{N-1} , E_{max}^N has to be qualitatively and quantitatively translated.

Depending on how a lost $(N-1)$ -SDU affects the reliability of layer N , different translations are needed. If a lost $(N-1)$ -SDU has the same adverse effect on QoS as an $(N-1)$ -SDU with single or multiple bit errors, i.e., they are both considered useless, then the following mapping can be applied:

$$E_{max}^N = L_{max}^{N-1} + H_{max}^{N-1}. \quad (6)$$

Consider a given bit error ratio B_{max}^{N-1} on layer $N-1$ (for the sake of simplicity, bit errors are assumed to occur independently) and M as the number of bits in an $(N-1)$ -SDU. An SDU loss may be indicated by layer $N-1$ and compensated for through layer N by means of dummy SDUs that are used to maintain bit count integrity [Jung, 1996]. An $(N-1)$ -SDU loss is thus considered as severe as K bit errors ($1 \leq K \leq M$). Under these assumptions the following mapping can be applied:

$$E_{max}^N = K \times L_{max}^{N-1} + \underbrace{1 - (1 - B_{max}^{N-1})^M}_{\cong M \times B_{max}^{N-1}}, \quad (7)$$

$$\cong M \times B_{max}^{N-1}, \quad \text{if } B_{max}^{N-1} \ll 1. \quad (8)$$

3.2 QoS Mapping Stimuli

(a) Segmentation and Reassembly

Video frames can be very large in size such that a network SDU can not accommodate a complete frame. Therefore frames have to be segmented at the

sending side. At the receiving side the reverse function is performed and frames are reconstructed.

The number M of $(N-1)$ -SDUs that are needed to transport constant sized (N) -SDUs can be calculated from the size S^N of the (N) -SDU and the size S^{N-1} of the $(N-1)$ -SDU:

$$M = \left\lceil \frac{S^N}{S^{N-1}} \right\rceil. \quad (9)$$

In the case of variable sized (N) -SDUs M is calculated with the maximum (N) -SDU size S_{max}^N instead of S^N . The effects of segmentation and reassembly are considered together and the procedure is hereafter referred to merely as segmentation.

Delay

The fundamental effect of segmentation on delay has been discussed by [Ferrari, 1990]. The time $d(S/R)_{max}$ it takes to hand over all of the segments to layer $N-1$, has to be deducted from the delay bound D_{max}^N . The time $d(S/R)_{max}$ is determined by the number of segments M and the time T^{N-1} that must pass before layer $N-1$ accepts the next $(N-1)$ -SDU-request. The “inter-request”-time has to be negotiated with the underlying layer such that D_{max} can be fulfilled.

$$D_{max}^{N-1} = D_{max}^N - \underbrace{T^{N-1} \times (M-1)}_{d(S/R)_{max}}. \quad (10)$$

Formula (10) provides an intermediate step to proceed with translation to the next layer, $(N-1)$, which is then only aware of the timely delivery of the segments, i.e., the $(N-1)$ -SDUs.

Jitter

Segmentation induces constant delay so that no jitter results, $J_{max}^{N-1} = J_{max}^N$, as long as the product $T^{N-1} \times (M-1)$ remains constant. As assumed, T^{N-1} is constant in all cases.

In the context of variable sized (N) -SDUs M , and consequently the delays induced by segmentation, will differ. The maximum (deterministic) jitter $j(S/R)_{max}$ introduced by the segmentation can be determined with the knowledge of the maximum and minimum negotiated size of (N) -SDUs, S_{max}^N and S_{min}^N :

$$J_{max}^{N-1} = J_{max}^N - \underbrace{T^{N-1} \times \left(\left\lceil \frac{S_{max}^N}{S^{N-1}} \right\rceil - \left\lceil \frac{S_{min}^N}{S^{N-1}} \right\rceil \right)}_{j(S/R)_{max}}. \quad (11)$$

Throughput

Segmentation may result in unused space in $(N-1)$ -SDUs. With fixed size

$(N - 1)$ -SDUs and (N) -SDU sizes not being a multiple of $(N - 1)$ -SDU sizes, segmentation will result in $M - 1$ full $(N - 1)$ -SDUs and one $(N - 1)$ -SDU that is only partly filled.

Although a higher bandwidth is requested from layer $N - 1$ by $T^{N-1} \times S^{N-1}$ layer N will only request a number of bits per second given by $\frac{T^N}{M} \times S^{N-1}$.

Reliability

Segmentation of SDUs entails appropriate adaptation of reliability requirements. When layer N breaks down (N) -SDUs into M $(N - 1)$ -SDUs then the reliability bound E_{max}^{N-1} demanded from layer $N-1$ depends on the reliability bound E_{max}^N demanded from layer N . The question is, how an erroneous $(N - 1)$ -SDU affects the (N) -SDU. The translation according to Eq. (12) assumes that an (N) -SDU is only correct if all of its parts (i.e., $(N - 1)$ -SDUs) are correct. $(N - 1)$ -SDU errors are assumed to occur independently [Jung, 1996]:

$$E_{max}^N = 1 - (1 - E_{max}^{N-1})^M, \quad (12)$$

$$\cong M * E_{max}^{N-1}, \quad \text{if } E_{max}^{N-1} \ll 1. \quad (13)$$

This represents a pessimistic approach, because it is assumed that a single erroneous part may corrupt the whole entity.

The translation of reliability by [Nahrstedt *et al.*, 1995] is based on a layer's SDU-loss-rate Lr . Although the rate itself being unaffected, the requested reliability is increased since there are M times more $(N - 1)$ -SDUs than (N) -SDUs:

$$Lr^{N-1} = Lr^N. \quad (14)$$

(b) Blocking

The procedure of mapping several (N) -SDUs into a single $(N - 1)$ -SDU is called blocking (or concatenation). This implies additional waiting delays. Assuming an ATM-network is used, several 16 bit samples of an audio stream have to be collected in order to fill a single cell.

The proposed translations are applicable if two conditions are satisfied: The sizes of the (N) -SDUs are constant and the (N) -SDU requests arrive at layer N according to the inter-request time T^N . The effects of blocking are comparable to those of segmentation with opposite implications. M (N) -SDUs are concatenated to form a single $(N - 1)$ -SDU:

$$M = \left\lceil \frac{S^{N-1}}{S^N} \right\rceil. \quad (15)$$

Delay

When blocking is used, delay occurs due to waiting for data from the above layer at the sender. The inter-request time T^N is specified by the upper layer $N + 1$. Each first (N) -SDU starting a new block experiences the most delay

$d(B)_{max}$ since it has to wait for the following $M-1$ (N)-SDUs to fill the $(N-1)$ -SDU:

$$D_{max}^{N-1} = D_{max}^N - \underbrace{T^N \times (M-1)}_{d(B)_{max}}. \quad (16)$$

Jitter

The maximum amount of (deterministic) jitter $j(B)_{max}$ induced by blocking is equal to the maximum delay $d(B)_{max}$. Each first (N)-SDU, which initiates a new block, has to be delayed for $d(B)_{max}$ until the last (N)-SDU completes the block. The minimum delay $d(B)_{min}$ for an (N)-SDU is zero since this last (N)-SDU experiences not delay:

$$j(B)_{max} = d(B)_{max} - d(B)_{min} = d(B)_{max}. \quad (17)$$

Throughput

Blocking may result in unused space when the (N)-SDUs do not neatly fit in the $(N-1)$ -SDU. The required bandwidth has to be increased, which has been done by the calculation of M . The inter-request time T^{N-1} of $(N-1)$ -SDUs is given by:

$$T^{N-1} = T^N \times M. \quad (18)$$

Reliability

After the (N)-SDUs have been assembled into an $(N-1)$ -SDU bursty errors may occur. When one $(N-1)$ -SDU is lost, M consecutive (N)-SDUs are lost.

If the user is resilient to particular errors, the reliability measures can be passed through the layer, since the probability for an erroneous SDU remains unchanged:

$$E_{max}^{N-1} = E_{max}^N. \quad (19)$$

(c) Interleaving

Interleaving is a measure to increase **reliability** and is often used in combination with forward error correction (FEC) mechanisms. Interleaving can be performed with a matrix which is column-wise filled with (N)-SDUs. When the matrix is full, each row is sent as a single $(N-1)$ -SDU to the receiver, who, in turn, waits for all matrix-SDUs to reconstruct the original (N)-SDUs.

The translations for reliability are not presented as they are highly dependent on the error profile of the underlying layer and on the amount of FEC information used*.

*Reliability is traded off against additional bandwidth requirements.

Interleaving causes additional **delay** and **jitter** similar to that introduced by blocking if all conditions, as mentioned in Sec. 3.2 (b), are satisfied.

Throughput is not altered by interleaving but by FEC. The translations for FEC are similarly performed as those for overhead (see Sec. 3.2 (e)).

(d) Playout Buffer

Statistical jitter can be compensated for by a playout buffer, which only works for SDUs that are requested periodically within a period T . Loss is not considered here. It is assumed that the clock of sender and receiver proceed at the same rate. The removed jitter is traded-off for additional delay due to buffering effects.

For example, if 2 msec of jitter shall be compensated for, the first SDU is delayed for the maximum playout buffer delay $d(PB)_{max} = 2$ msec. After this time, after each time interval T an SDU is removed from the buffer. If the buffer is empty the next SDU will be passed directly through the buffer. Should an SDU arrive at a full buffer, the next scheduled SDU is immediately forced out of the buffer.

Layer N-1 can be conceded an increased jitter bound but has to enforce a stricter (reduced) delay bound:

$$D_{max}^{N-1} = D_{max}^N - d(PB)_{max}, \quad (20)$$

$$J_{max}^{N-1} = J_{max}^N + d(PB)_{max}. \quad (21)$$

The size S_B of the playout buffer depends on the size S of the SDUs, the jitter J that shall be compensated, and the inter-request time T of the SDUs [Moran *et al.*, 1992]. When SDU sizes vary, the maximum SDU-size S_{max} has to be taken into account instead of S :

$$S_B = \left\lceil \frac{2 \times J}{T} \right\rceil * S. \quad (22)$$

The drawback of using a playout buffer is that additional delay might be imposed. Most important, though, is the handling of the first SDU. The first SDU should be delayed by $d(PB)_{max}$ only if it had suffered the minimum delay. In case the first SDU suffered maximum delay playout could commence at once. As we do not have information about the delay of the first SDU we always have to delay it by $d(PB)_{max}$.

In case of deterministic jitter, the delay of the first SDU is known. Consequently, deterministic jitter can be removed with less effort than stochastic jitter. *

For the removal of deterministic jitter induced by blocking, for example, the following measure is taken: Each SDUs is delayed by the amount of a

*Every SDU is then delayed by D_{max} ; so the particular SDU suffers more delay but the delay bounds remain unchanged.

jitter compensation delay $d(JC)_i$, which depends on the delay $d(B)_i$ that was imposed on the i th SDU due to blocking, and the maximum delay $d(B)_{max}$, which any SDU experiences due to blocking:

$$d(JC)_i = d(B)_{max} - d(B)_i. \quad (23)$$

Throughput and **reliability** requirements remain unchanged by the playout buffer if buffer overflow does not result in SDU-loss.

(e) Overhead

Overhead can be due to operating systems functionalities, like scheduling, or due communication protocol data.

Delay

It is assumed that at is known at every layer N how much time is needed for these tasks and this portion should be deducted from the given delay bound D_{max}^N .

Throughput translation is necessary on each layer N , when overhead (protocol data) is attached to (N) -SDUs. The SDU-size S^{N-1} , which is demanded from the underlying layer $N - 1$, is adjusted by adding the number of overhead bits O^N to the size S^N of the (N) -SDU. The translation for **reliability** depends on how sensitive the protocol data is compared to the (N) -SDU. The reliability measures at layer $N - 1$ are chosen according to whatever being more demanding, overhead or (N) -SDUs.

(f) Coding

Coding can have spatial and temporal implications. In an MPEG coded video sequence, for example, each frame is coded according to one of three encoding modes: I, P, and B. While the I-frames (intra-coded) are coded without references to any other frame, P-frames (predictive) are coded as difference pictures from the last I- or P-frame. B-frames (bi-directional predictive) are coded as differences from an interpolation of the preceding and the succeeding I- or P-frame. So when a B-frame has to be coded, future I- or P-frames have to be waited for.

The maximum coding delay for MPEG $d(MPEG)_{max}$ depends on the maximum computing time $d(C)_{max}$ for encoding a bidirectional frame, on the maximum distance R of a B-frame to the following reference frame, and on the frame rate f at which the frames are generated (this is similar to blocking and peak smoothing, since it has to be waited for future data):

$$d(MPEG)_{max} = R \times \frac{1}{f} + d(C)_{max}. \quad (24)$$

The minimum delay for an MPEG-coded video frame is equal to the delay of the fastest P-frame (en/de-coding), which does not have to wait for future

reference frames:

$$d(MPEG)_{min} = d(C)_{min}. \quad (25)$$

The **jitter** caused by MPEG coding is once again calculated by subtracting the minimum from the maximum delay.

The reductions in average bandwidth requirements depend on the content of the video and the IPB pattern. The required bandwidth is reduced at the cost of additional delay and jitter. Moreover, reliability has to be increased since the MPEG-coded video is more sensitive to errors than the original frames.

(g) Compression

Compressing SDUs is a measure to reduce the size of SDUs. The translation of the **delay** and **jitter** parameter bounds are simple. The new bounds are computed by deducting the maximum delay/jitter that can occur during compression tasks from the given delay and jitter parameter bounds.

Throughput

If there is no guaranteed compression minimum that holds for all SDUs, the reduction of some of the SDUs cannot be used for a deterministic bandwidth requirement. With the minimum compression ratio c_{min} from Eq. 26 the $(N - 1)$ -SDU-size S^{N-1} is given by Eq. 27 :

$$c_{min} = \max \left(\frac{\text{size after compression}}{\text{size before compression}} \right), \forall \text{ SDUs}, \quad (26)$$

$$S^{N-1} = S^N \times c_{min}. \quad (27)$$

It has to be noted that compression increases the vulnerability to distortions of the SDUs. So reliability requirements might need adjustments.

(h) Peak Smoothing

VBR services can have large bandwidth requirements W_{max} which can be reduced by utilizing peak smoothing. When peak smoothing, also known as frame spreading (cf. [Ismail *et al.*, 1995]), is used, an (N) -SDU, e.g., each video-frame, is not transferred completely in its request period T^N but in multiple periods.

Depending on the burstiness of the SDU-size and on the budget of time for smoothing, the maximum SDU size S_{max}^{N-1} for layer N-1 can be reduced to a value close to the average SDU-size S_{avg} .

The inter-request time of the SDUs remains unchanged by peak smoothing ($T^{N-1} = T^N$).

The peak smoothing **delay** is easy to calculate, since it has similar effects as blocking. The time interval of interest is given by the inter-request time T^N of the (N) -SDUs. The formula is the same as for blocking (Sec. b). Under our assumptions peak smoothing has no effect on **jitter**.

4 CONCLUSIONS

An overview was given how the translation of QoS parameters across layers may be performed. We identified some of the important features of a distributed multimedia system which stimulate effects of typical mapping issues on QoS parameters. Existing partial solutions of the mapping problem were related to each other. Additional mappings were given in order to complement the existing approaches from the literature for a more comprehensive view. In particular, effects related to VBR services were high-lightened and extended approaches were derived. Further research is necessary in order to relate the mapping issues closer to the cognitive capabilities of the human user.

REFERENCES

- [Anagnostou *et al.*, 1991] Anagnostou, M., Theologou, M., Vlakos, K., Tournis, D., and Protonotarios E. (1991) Quality of Service requirements in ATM-based B-ISDNs. *Computer Communications*, Vol. 14, No 4.
- [Casaca *et al.*, 1992] Casaca, A. and Nunes, M. (1992) Some aspects of the adaptation of services to an ATM network, in *Information Network and Data Communication* (eds. M. Tienari and D. Khakhar), IFIP, North Holland.
- [Danthine *et al.*, 1992] Danthine, A., Baguette, Y., Leduc, G., and Leonard, L. (1992) The OSI 95 Connection-Mode Transport Service - The Enhanced QoS. *4th IFIP Conf. on High Performance Networking*.
- [De Meer *et al.*, 1997] De Meer, H. and Knoche, H. (1997) Quality of Service Parameters: A Comparative Study. *Technical Report*, Univ. of Hamburg.
- [Damaskos *et al.*, 1994] Damaskos, S. and Gavras, A. (1994) A simplified QoS Model for Multimedia Protocols over ATM. in *High Performance Networking* (ed. S. Fdida), Proc. IFIP Transactions, C-26, North Holland.
- [Ferrari, 1990] Ferrari, D. (1990) Client requirements for real-time communication services. *IEEE Communications Magazine*.
- [Ismail *et al.*, 1995] Ismail, M., Lambadaris, M., Devetsikiotis, M., and Kaye, A. (1995) Modelling Prioritized MPEG Video Using TES and a Frame Spreading Strategy for Transmission in ATM Networks. *IEEE INFOCOM*.
- [Jung, 1996] Jung, J. (1996) Quality of Service in Telecommunications Part II: Translation of QoS Parameters into ATM Performance Parameters in B-ISDN. *IEEE Communications Magazine*.
- [Moran *et al.*, 1992] Moran, M. and Wolfinger, B. (1992) Design of a Continuous Media Data Transport Service and Protocol. *TR-92-019*, ICSI Berkeley.
- [Nahrstedt *et al.*, 1995] Nahrstedt, K. and Smith, J. (1995) The QoS Broker *IEEE Multimedia Journal*.

5 BIOGRAPHY

Hendrik Knoche is a graduate student at the CS-Dept. of the University of Hamburg. Since 1995 he is member of the QoS research group. He specializes in QoS parameter mappings in his MS thesis. In addition he has been involved in the design of commercial multimedia products.

Hermann de Meer received his Ph.D. from the University of Erlangen-Nuernberg, Germany, in 1992. He spent his postdoctoral research at Duke University and at the University of Texas at Austin, USA. Since 1993 he is affiliated as an Assistant Professor with the CS-Dept. of the University of Hamburg.

Interests of research include Distributed Systems, Quality of Service, User Perception and Cognition in Multimedia, and Performance and Performability modeling. He is director of the QoS research group at his Department and serves as the principle investigator in several funded international research projects on modeling and QoS related topics.

Besides having published at many reviewed conferences and in journals, he is co-authoring a textbook on queuing networks and Markov models. H. de Meer is an IEEE, ACM, and GI member and serves as program chair of the WCSS/ANMT QoS-Modelling Conf. in Singapore, Sept 1997.