ACM/Springer Multimedia Systems Journal – Special Issue on QoS Systems, Vol. 6, No. 3, May 1998, pp. 152-166

HeiRAT — Quality-of-Service Management for Distributed Multimedia Systems

Carsten Vogt[*], *Lars C. Wolf*[**], *Ralf Guido Herrtwich*[***], *Hartmut Wittig*[****]

IBM European Networking Center, Vangerowstr. 18, D-69115 Heidelberg, Germany

Abstract: Multimedia systems must be able to support a certain quality of service (QoS) to satisfy the stringent real-time performance requirements of their applications. HeiRAT, the Heidelberg Resource Administration Technique, is a comprehensive QoS management system that was designed and implemented in connection with a distributed multimedia platform for networked PCs and workstations. HeiRAT includes techniques for QoS negotiation, QoS calculation, resource reservation, and resource scheduling for local and network resources.

Key words: Quality of Service, Resource management, Distributed multimedia, Scheduling, Admission Control

1 Introduction

The processing of digital audio and video streams has to obey timing requirements which are typically not considered in traditional computer systems. Multimedia operating and communication systems have to take into account these timing criteria when managing system resources in order to provide a certain *quality of service* (QoS) to multimedia applications. This QoS typically includes specifications for throughput, delay, and reliability.

The *HeiProjects* (Herrtwich 1994) at IBM's European Networking Center in Heidelberg were aimed at providing a distributed multimedia platform for PCs and workstations in an internetwork of LANs such as Token Ring and Ethernet. Among other things, they included the development of *HeiTS* (the Heidelberg Transport System) for transporting multimedia streams across the network (Wolf and Herrtwich 1994) and *HeiRAT* (the Heidelberg Resource Administration Technique) for providing a well-defined QoS for this transport (Vogt, Herrtwich and Nagarajan 1993).

In this paper, we describe the features of HeiRAT in retrospective. The following section provides an overview of the HeiRAT system. We then devote one section each to the definition of QoS values in HeiRAT, to the enforcement of QoS, and to the calculation of QoS parameters.

2 HeiRAT Overview

All system resources through which a multimedia stream passes may affect the QoS of this stream. HeiRAT, therefore, considers all resources on a path from source to sink(s), both in the local systems and the network (see Figure 1). Resources can be classified as active and passive. Active resources process data, they include CPUs, busses, I/O systems, network adapters and transmission links. Passive resources store data, they include memory space in end nodes and network routers. In this section, we look at the basic features of HeiRAT to manage all these resources.

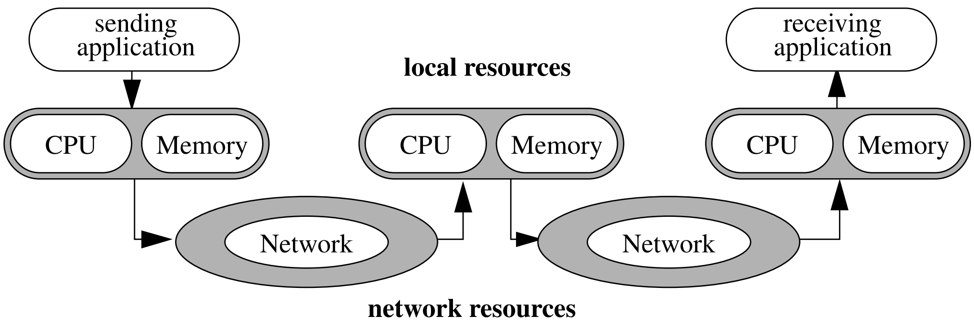

Figure 1: Resources Managed by HeiRAT

e-mail: vogt@fh-koeln.de, Lars.Wolf@kom.th-darmstadt.de, Ralf.Herrtwich@RWE-Telliance.de, wittig@mms-dresden.telekom.de
Correspondence to: Carsten Vogt

[*] Carsten Vogt is now with FH Köln, Fb. Nachrichtentechnik, Betzdorfer Str. 2, D-50679 Köln, Germany.
[**] Lars C. Wolf is now with TH Darmstadt, Fb. Elektrotechnik & Datentechnik, Merckstr. 25, D-64283 Darmstadt, Germany.
[***] Ralf Guido Herrtwich is now with RWE Telliance AG, Gildehofstr. 1, D-45127 Essen, Germany.
[****] Hartmut Wittig is now with Multimedia Software GmbH, Riesaer Str. 5, D-01129 Dresden, Germany

2.1 Functions of HeiRAT

HeiRAT provides the following functions for both active and passive resources:

- *Throughput test*: When a new multimedia stream shall be established, it is checked whether enough free resource capacity is available to handle it. This decision is influenced by the QoS guarantees already given to other streams; these must not be violated by the new stream.
- *QoS calculation:* Every resource computes the QoS it can provide for the new stream.
- *Resource reservation*: The resource capacity is reserved that is required to provide the QoS guarantee.
- *Resource scheduling*: Resource access is coordinated so that the respective QoS guarantees of all streams are satisfied.

In the *set-up* or *QoS negotiation phase* of a multimedia stream, applications specify their QoS requirements. These parameters are used for the throughput test and the QoS calculation which result either in a resource reservation or in the rejection of the stream establishment, the latter if the QoS cannot be met. In the *transmission* or *QOS enforcement phase*, after the successful establishment of a stream, the resources are scheduled with respect to the given QoS guarantees.

In the set-up phase, HeiRAT offers several options by which applications can specify their QoS requirements. QoS values are given in terms of maximum end-to-end delay, minimum throughput needed, and reliability class defining how the loss of data shall be treated. An application can select one of the QoS parameters for optimization by specifying an interval from *desired* to *worst-acceptable* values. For example, a video application might request a throughput between 15 and 30 video frames per second, indicating that video quality would not be acceptable with less than 15 frames, but that more than 30 frames are never needed. HeiRAT will then return the best QoS it can guarantee within this interval and make the corresponding reservation (or reject the request if even the lower bound cannot be supported).

In the transmission phase, data are processed and transmitted according to their urgency. Schedulers handle time-critical multimedia streams prior to time-independent data. They exploit properties of the underlying resources, for example, they are based on the operating system priority scheme for CPU scheduling or the MAC priority scheme of the network.

HeiRAT offers two types of QoS: *guaranteed* and *statistical*. For guaranteed QoS, the resource capacities reserved are for the maximum demand a stream may have during its lifetime. Reserving extensive amounts of capacities for such peak requirements can be rather costly and leads to the under-utilization of resources if there is a significant difference between peak and average data rate of a stream. A cheaper alternative is statistical QoS where resources are slightly overbooked. This implies that while QoS requirements will be met most of the time, occasional QoS violations may occur (and applications need to be ready to cope with them).

2.2 The Role of Resource Reservation Protocols

Multimedia streams that are transmitted across a multi-hop network are handled by multiple system resources. The guarantees of the individual resources must be aggregated to obtain an end-to-end QoS guarantee. This requires a *resource reservation protocol* to exchange and negotiate QoS requirements across system boundaries. The fact that the network is one of the resources to be managed makes it necessary to integrate the resource reservation protocol with the network layer of the transport system; higher layers have no information about the different resources in the network.

Examples of such reservation protocols include ST-II (Topolcic 1990), its more recent version ST-II+ (Delgrossi and Berger 1995), and RSVP (Zhang et al. 1993). While these protocols differ in their underlying design philosophy (modularity vs. completeness, connection-orientation vs. soft-state, etc.) (Mitzel et al. 1994, Delgrossi et al. 1993), they are all appropriate means for exchanging reservation information. When HeiTS and HeiRAT where conceived, ST-II was the only apparent reservation protocol under discussion in the IETF. Today, RSVP (in connection with IP-NG) attracts more attention. Most HeiRAT mechanisms can work with RSVP just as they work with ST-II.

As illustrated in Figure 2, origin and target applications, agents executing the resource reservation protocol, and local resource managers participate in the QoS negotiation. The origin supplies the initial QoS requirement. This QoS request is possibly mapped by the transport layer on a QoS request in terms of network layer units due to packet segmentation and then becomes part of a connection establishment message. Each local resource manager on the path receiving the message computes the QoS its resources can provide and reserves the corresponding resource capacities.

If the reservation fails (either due to resource overload or insufficient resources for a given QoS requirement), a corresponding message is sent back to the origin releasing all reservations made so far. Otherwise, the protocol agent updates the QoS specification (for example, keeping track of the accumulated delay or adjusting throughput) and passes the stream establishment message downstream towards the targets. Targets work in the same way and communicate a QoS specification or a refusal message back to the origin.

To make sure that all resources work with the same QoS parameters, the origin may then send out a request defining the final throughput and reliability values of the stream as the respective minimum values of all resources. *Excess delay* (any positive difference between achievable and desired delay) is distributed among the resources to relax guarantees.

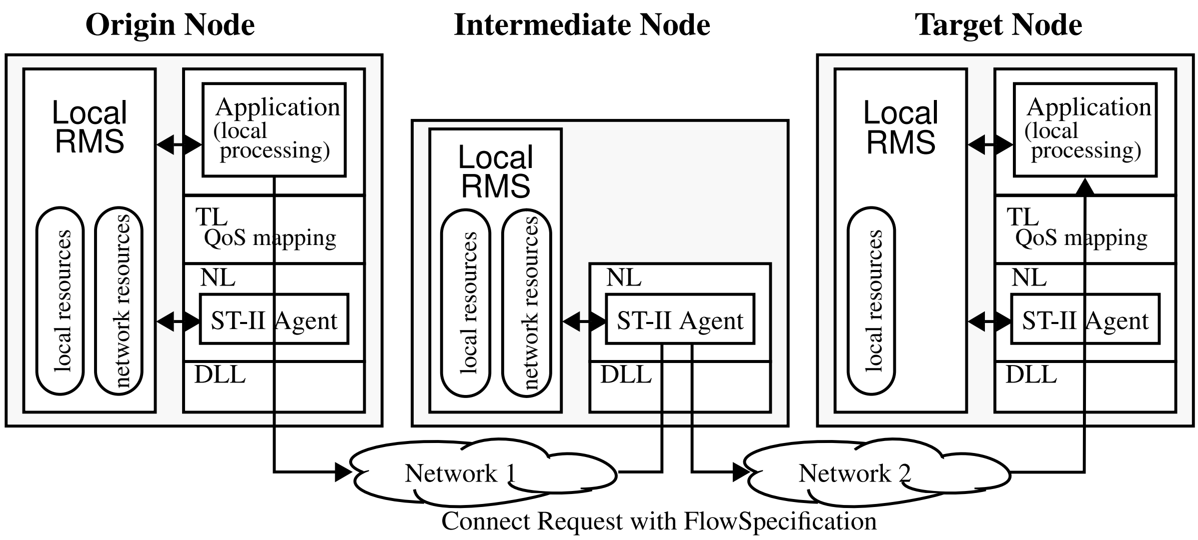

Figure 2: Distributed QoS Computation.

Resource reservation protocols should be supported by QoS-driven routing algorithms that find optimal paths for a given set of QoS requirements in a mashed network of resources. The lack of routing mechanisms with QoS support led us to develop QoSFinder. QoSFinder uses path vector routing with QoS metrics for the evaluation of routes. A detailed specification of the routing protocol, its metrics and a common algorithm for the comparison of two sets of QoS parameters can be found in (Vogel et al. 1995).

2.3 HeiRAT Usage Scenario

For the time being, our HeiRAT implementation is used in the HeiTS communication system and the HeiTS-based Ultimedia Server/6000, a multimedia client-server system for audio and video retrieval. Guaranteed and statistical QoS calculation and reservation are provided for CPU and main memory space as well as for Token Ring and Ethernet networks.

For the IBM AIX Version 3 operating system, schedulers for the CPU and the Token Ring adapter have been implemented which handle requests according to their urgency. The resource reservation protocol ST-II and the transport layer protocol HeiTP (Delgrossi et al. 1992) use the functionality provided by HeiRAT, the latter for mapping QoS specifications between the transport and the network layer.

The applicability of HeiRAT is not confined to the HeiTS environment. Indeed, the HeiRAT approach for distributed QoS calculation can be extended to arbitrary chains of software modules (defined as *stream handlers* in (Wolf and Herrtwich 1994)) and networks connecting sources to sinks. Here, for each individual stream handler or for sequences of adjacent stream handlers, the HeiRAT functions can be called to reserve appropriate resource capacities and to return QoS guarantees for the execution of these modules. In this scenario, the protocol stack of HeiTS could be one of the stream handlers. The QoS guarantees given for the individual stream handlers can then be accumulated in a similar fashion as done in a network with its routers and transmission links to yield end-to-end QoS guarantees. See (Wolf 1996) for more information.

2.4 Management of HeiRAT

To set and update HeiRAT parameters, the system can be managed using the Simple Network Management Protocol (SNMP) (Case et al. 1990)].

The HeiRAT Management Information Base (MIB) consists of three parts: *resources*, *reservation*, and *monitoring*.

The resource part of the MIB contains information related to all resources managed by HeiRAT. For example, information about the total capacity of the network can be obtained, or the maximum amount of bandwidth reservable by multimedia traffic can be set. The reservation part of the HeiRAT MIB contains actual information about the reservations for multimedia connections, for example, the set of QoS parameters for connections, or the scheduling priority of a reservation. From the variables in the monitoring part, resource consumption information (for example, used network bandwidth of a connection, CPU usage of a process) can be retrieved. A detailed description of the HeiRAT MIB can be found in (Kätker et al. 1993).

3 QoS Description

The purpose of a QoS description is manifold. QoS parameters (at least some of them) serve as a source description; they specify properties of the data stream an application will feed into the system. Additionally, QoS parameters are needed to describe the performance requirements of an application and to define the corresponding performance guarantees returned by the system. In this section, we look at how QoS is described in HeiRAT and compare this description with other ways to specify QoS.

3.1 QoS Parameters in HeiRAT

Three parameters are of main interest when it comes to transporting multimedia streams: *throughput*, *delay* and *reliability*. All three QoS parameters are closely related: The smaller the overall bandwidth of a resource is compared to its load, the more messages will accumulate in front of it and the larger the buffers need to be to avoid loss. The larger the buffers become, the more likely it gets that messages need to wait to be serviced, that is, the larger the delay will get. Hence, only a full description of the entire parameter set provides a clear understanding of the QoS provided.

3.1.1 Throughput

The HeiRAT throughput model is based on the *linear bounded arrival process (LBAP)* model as introduced by (Cruz 1991) and used by (Anderson 1993). The LBAP model assumes data to be sent as a stream of discrete units (*packets*) characterized by three parameters:

- S = *maximum packet size*,
- R = *maximum packet rate* (i.e., maximum number of packets per time unit), and
- W = *maximum workahead*.

The workahead parameter W allows for short-term violations of the rate R by defining that in any time interval of duration t at most $W + t*R$ packets may arrive on a stream. This is necessary to model input devices that generate short bursts of packets, for example disk blocks that contain multiple multimedia data frames, and also to account for any clustering of packets as they proceed towards their destination (for work conserving systems). Although it may be somewhat counter-intuitive, it is possible to use LBAPs for the management of variable bit-rate streams with varying bandwidth requirements as shown in (Vogt 1995).

A useful concept with regard to the LBAP is that of *logical arrival time*. The logical arrival time l of a message m_i is defined as: $l(m_0) = a_0$, the actual arrival time of the first packet, and $l(m_{i+1}) = \max\{a_{i+1}, l(m_i) + 1/R\}$. The concept of logical arrival time essentially acts as a smoothing filter for the traffic streams. It ensures that no particular stream hogs a resource at the expense of other streams given their declared workload characteristics. We will refer to the entity that computes these logical arrival times and schedules packets accordingly as the *regulator*. A packet whose logical arrival time has passed is called *critical*, otherwise it is referred to as *workahead*.

The output stream of a resource serving an input LBAP is itself an LBAP. Its parameters depend on the parameters of the input LBAP and the maximum and minimum delay within the resource. Their computation is described in (Anderson, Herrtwich and Schaefer 1990). This enables one to "push" the LBAP workload model from the origin through to the destination nodes for each stream.

In addition to the three LBAP parameters defined above, the user must specify the maximum processing time per packet for each resource such that resource capacities can be correspondingly reserved. In (Wittig, Wolf and Vogt 1994) the problem of processing time measurement is analyzed and a measurement tool for CPU processing times of multimedia stream handling modules is presented.

3.1.2 Delay

Delay in HeiRAT is specified in terms of *minimum actual delay* D_{min}, which is a lower bound to the actual packet transfer time on the connection, and *maximum regular delay* D_{max}, which is the time at which a packet leaves the transport system at the latest with respect to its logical arrival time. Only the maximum regular transit time is communicated between the application and HeiRAT, the minimum actual transit time is used only internally to determine the end-to-end *jitter*, the variance in the end-to-end delay.

HeiRAT does not specify jitter separately as it assumes that multimedia packets can be buffered before they are made available to the receiving application. To ensure that there is always a packet available when the application requires it, the buffering time is up to $D_{max} - D_{min}$. However, this approach can be expensive in terms of buffer space and additional transmission delay incurred by the buffering on the target node.

3.1.3 Reliability

The HeiRAT specification of reliability distinguishes between bit errors and packet losses. This distinction is motivated by the observation that one would not necessarily discard a whole multimedia message (for example a video frame) when only a small number of bits is corrupted. Reliability classes define how these two types of error shall be handled by the transport system (Table 1).

	Class 0	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Bit errors	ignore	ignore	indicate	ignore	correct
Packet errors	ignore	indicate	indicate	correct	correct

Table 1: Reliability Classes.

The reliability parameter specifies the best error treatment the network layer can provide without increasing the straightforward throughput or delay. For example, the IBM Token Ring Busmaster adapter can immediately indicate the successful or unsuccessful delivery of packets. An implicit error correction by retransmission would be possible, but would increase throughput and delay. Networks incorporating forward-error-correction may very well be capable of such correction procedures without significant delay increases. Hence, delay and throughput in HeiRAT does not account for error handling operations such as retransmissions – the overhead of these functions has to be considered by the transport layer when transforming a higher layer QoS specification into a network layer specification, or vice versa.

Note that the HeiRAT reliability model provides only the few reliability classes mentioned but nothing further (as, e.g., loss rate guarantees). This stems from the fact that the HeiRAT QoS management approach is primarily focused on the optimization of throughput and delay. Losses are assumed to happen, especially for streams with a statistical QoS, but it is left to higher layers to prepare for their occurrence and to handle them appropriately. (Delgrossi et al. 1994) and (Wolf, Herrtwich and Delgrossi 1995) include a detailed discussion of possible approaches.

3.2 HeiRAT and Other QoS Parameter Schemes

The HeiRAT QoS parameter set was chosen because we felt it described the most important properties of multimedia streams in a natural and simple way. However, there exist many other schemes for the specification of QoS. It is therefore important to find out whether mappings between such different parameter sets exist and how the various approaches can coexist in one system.

3.2.1 QoS Model Used in plaNET

plaNET is a high-speed packet-switched network for the integrated communication of voice, video, and data (Cidon, Gopal and Guérin 1991). A key feature of plaNET is its transparency, i.e. its ability to transmit information in various formats such as packets of variable sizes or fixed-size ATM cells.

QoS management in plaNET is based on a stochastic traffic model. This model assumes a traffic source to be in one of two states, either the *idle state* emitting no traffic at all or in the *burst state* sending traffic at a certain *peak rate*. The time during which the source is in its burst state is called a *burst phase.* Based on this model, traffic sources are characterized by three parameters:

- R = *peak rate* = traffic rate during burst phases
- m = *mean traffic rate* over the total time
- b = *average duration of a burst phase*

The plaNET scheme reserves bandwidth for individual connections on links in an internetwork. By this reservation, a certain throughput is guaranteed with some specified loss probability due to buffer overflow at the links' entrances. The amount of bandwidth reserved for a connection j, the so-called *equivalent capacity* c_j, lies between its mean rate m_j and its peak rate R_j. It depends on the above traffic parameters, the amount of buffer space available to store waiting messages and the desired buffer overflow probability.

In general, several connections are multiplexed over one link. If link bandwidth is high, and so is the number of connections that can be multiplexed, the aggregate bit rate of the connections can be approximated by a Gaussian distribution. If not, the aggregate bit rate is approximated by the sum of the equivalent capacities, i.e., for the purpose of QoS management a connection is treated as if it sent its bits at a constant rate of c_j. By comparing the aggregate bit rate already allocated and the equivalent capacity requested by a new connection with the total link bandwidth it can be decided whether the new connection can be admitted.

Although the traffic models and the approaches for QoS calculation used in HeiRAT and plaNET differ, an integration of their QoS management is possible. Consider the scenario illustrated by Figure 3 where an end-to-end connection between two stations across two lower-speed LANs and a high-speed WAN shall be established. Assume that QoS demands of the connection are specified by plaNET parameters, QoS management on the WAN is done by the plaNET scheme, and QoS management on the LANs by HeiRAT. Hence, on the LANs the plaNET parameters have to be mapped on HeiRAT parameters such that a HeiRAT resource reservation can take place.

This mapping can be done as follows: A plaNET connection does not require a 100% service guarantee but accepts a limited number of message losses. This corresponds to a statistical HeiRAT connection. Moreover, on LANs with a comparatively low capacity the number of connections is small and hence their aggregate bit rate has to be characterized by the sums of their equivalent capacities of all connections (see above). Thus, for the purpose of bandwidth allocation only the equivalent capacities play a role; the values of the three plaNET QoS parameters are needed only for calculating these. The equivalent capacity can be directly used as throughput parameter for the HeiRAT reservation function. The message size parameter is the message size of the plaNET connection, which must be specified in addition to the three plaNET parameters, and the workahead parameter can be zero.

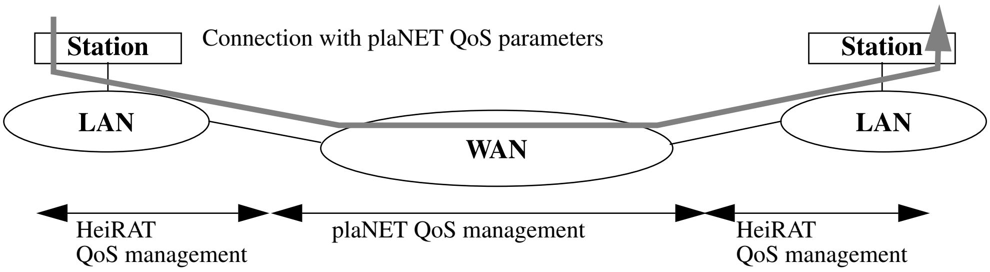

Figure 3: An Internet Integrating HeiRAT and plaNET QoS Management.

Now consider the scenario that the management of the LANs and the intermediate WAN is still done by plaNET and HeiRAT, respectively, but the end-to-end connection now defines its QoS in terms of HeiRAT parameters. Note that only statistical HeiRAT connections can be supported here as plaNET provides no

100% guarantees. In this scenario, the HeiRAT parameters have to be mapped on plaNET parameters such that bandwidth on the WAN can be reserved. This mapping is only possible for the throughput parameters of HeiRAT as plaNET manages only bandwidth, but gives no delay guarantees. The plaNET mean rate *m* is calculated as the product of the HeiRAT packet size and rate. A burst is, according to the HeiRAT scheme, a sequence of at most *W* incoming packets, *W* being the workahead parameter. The burst rate and hence the burst duration required by plaNET depend on how fast the gateway or the application can physically feed packets into the network; they can be directly derived from this information.

3.2.2 QoS Model Used in Q.933

A QoS standard that has emerged in the ISDN environment is Q.933 (ANSI 1991). Q.933 describes the traffic source and its throughput requirements in terms of an average throughput rate over any interval of a given length and a possible burst in such an interval, both given in terms of bits. As Q.933 assumes the packetization of traffic it also specifies a maximum packet size. Q.933 includes a delay parameter signifying that 95% of a stream of packets of maximum size experience a delay not longer than this value. Q.933 allows for specifying an acceptable interval for the QoS instead of fixed values when defining QoS requirements.

In the following, we consider again the second scenario of the previous section. QoS requirements and guarantees for an end-to-end connection are given in terms of HeiRAT parameters but the reservation of some intermediate resource is based on Q.933. When issuing a reservation request to a network whose management is based on Q.933, the HeiRAT QoS request has to be transformed into a reservation request in terms of Q.933 parameters.

A HeiRAT QoS description is based on the specification of a maximum workload. The number of packets $N(t)$ in an interval of length t is bounded by $N(t) \leq W + t*R$ for some workahead parameter $W \geq 1$ and some packet rate parameter R. This upper bound can be further reduced to $N(\mathrm{t}) \leq W + \lfloor t*R \rfloor$, i.e., to a discontinuous function, because $N(t)$ counts only complete packets. The graph of the function $b_{Hei}(t)$ in Figure 4 illustrates this upper bound. $b_{Hei}(t)$ defines the maximum number of packets that may have arrived after t time steps.

Q.933 defines the maximum workload in a different way. Here, the workload is described by a *committed burst size* (*CBS*) and a *throughput* (*TPT*). *CBS* is the maximum number of bits (not packets!) that may arrive in any interval of length *T*. *TPT* is the average number of bits per second. The *interval length T* is defined as the ratio of *CBS* and *TPT*, i.e., $T = CBS\ /\ TPT$. In Figure 4, The graph of the function $b_{Q.933}(t) = CBS + \lfloor t/T \rfloor * CBS$ illustrates this upper bound of the workload. $b_{Q.933}(t)$ defines the maximum number of bits that may have arrived after t time steps. Note that also this function is discontinuous because data may arrive in bursts.

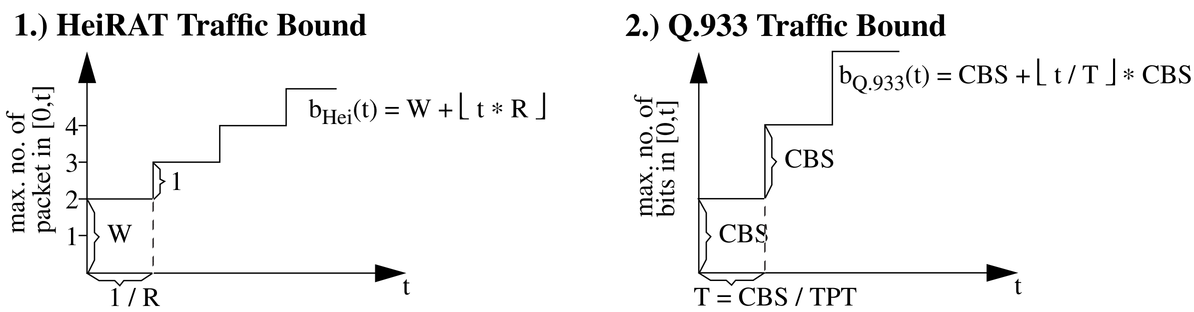

Figure 4: Traffic Bounds for the HeiRAT and the Q.933 QoS Model.

When submitting a reservation or a change request from HeiRAT to a network managed by Q.933, it must be ensured that enough network bandwidth for the calling HeiRAT function is reserved. To reserve “enough” bandwidth means that even the worst possible behavior of the requesting process must be covered by the reservation. Hence, the goal is to find for given *W* and *R* appropriate values for *CBS* and *TPT* such that for all t $b_{Hei}(\mathrm{t}) \leq b_{Q.933}(t)$ (abstracting for a moment from the fact that Q.933 counts bits rather than packets).

For given *W* and *R*, some properties of *CBS* and *TPT* can be derived:

(1) $CBS \geq W$, because the initial burst must be covered by CBS. Hence, there is some real number $r \geq 1$ with $CBS = r*W$.

(2) There should be at least one t' such that $b_{Hei}(t') = b_{Q.933}(t')$, because otherwise Q.933 would make an unnecessary over-reservation. As $b_{Hei}(t)$ is growing not faster than $b_{Q.933}(t)$, such a t' can be found in the environment of the first jump of $b_{Q.933}(t)$, i.e., for $b_{Q.933}(T)$. Hence, for some $\varepsilon>0$: $b_{Hei}(T\text{-}\varepsilon) = b_{Q.933}(T\text{-}\varepsilon) = CBS$. Also, the first jump of $b_{Q.933}(T)$ should coincide with a jump of $b_{Hei}(t)$ (also to avoid an over-reservation) and thus $CBS = W+T*R\text{-}1$.

(3) From (1) and (2), we have $W+T*R\text{-}1 = r*W$.

The question is now how to choose r. One possible approach is to attempt to minimize the overall throughput $TPT = CBS / T$ in order to claim as little bandwidth as possible. However:

$$TPT = \frac{CBS}{T} = \frac{rW}{T} = \frac{rW}{\frac{(r-1)W+1}{R}} = \frac{rWR}{(r-1)W+1}$$

It can be seen from the above expression that TPT has no minimum because TPT is continuously decreasing with growing r. Note also that CBS is growing linearly in r, i.e., there is a trade-off between burst size and overall throughput. Hence, one should select an r with an appropriate trade-off, yielding

$$CBS = rW \qquad TPT = \frac{rWR}{(r-1)W+1} \qquad T = \frac{(r-1)W+1}{R}$$

There are two special cases in which the calculation of CBS and TPT is much simpler. For guaranteed connections with a regulator, there is a workahead of $W = 1$. Hence, one can select $CBS = 1$ and $TPT = R$, i.e., $T = 1/R$. This implies $b_{Hei}(t) = b_{Q.933}(t)$ for all t, i.e., no overreservation is incurred by the mapping. The same values can also be chosen for statistical connections, which means that no reservations for bursts on such connections are made. The Q.933 QoS specification provides an extra parameter called *excess burst size* which can be used instead of CBS to indicate this burst.

Note that the above definitions of CBS and TPT are given in terms of packets. To get the corresponding values in terms of bits (as required by Q.933) the above expressions have to be multiplied by the maximum packet size (which is provided by HeiRAT) and by 8 (= number of bits per byte). The mapping of the other QoS parameters is straightforward (see (Taber 1993) for details).

The backward direction, i.e., the mapping of Q.933 QoS parameters to HeiRAT parameters is needed when Q.933 returns a QoS guarantee which must be forwarded in terms of HeiRAT QoS values. If Q.933 makes only a yes/no decision on the request, the returned QoS values are the same as the input values and hence the returned guarantee is simply the requested QoS. Otherwise, Q.933 returns a TPT and a CBS value for which W and R values must be found such that $b_{Hei}(t) \leq b_{Q.933}(t)$. This is easy and also described in (Taber 1993) in more detail.

4 QoS Enforcement

To enforce a given set of QoS assertions, it needs to be controlled which work item a resource processes at a given time. This may be achieved by admission control that locks out applications once a certain workload is reached. A more general and flexible way to enforce QoS is to take into account QoS requirements when scheduling a resource. In this section we look at the scheduling of both CPU and network access.

4.1 CPU Scheduling

The HeiRAT algorithms for CPU scheduling are based on classical approaches for real-time processing, namely *earliest-deadline-first* (EDF) and *rate-monotonic* (RM) scheduling (Liu and Layland 1973). In this context, HeiRAT assumes the packet streams on the network layer to be periodical. The scheduling and QoS management of aperiodical transport layer streams as, e.g., found in video applications with variable bit rates can be adapted to this model as described in (Vogt 1995).

EDF scheduling assumes each process to have a deadline at which its processing must be finished. For periodical packet streams, the deadline for the processing of a packet can be defined as the end of its period. Within EDF, the process with the earliest deadline among the waiting processes is executed first. In

RM scheduling, the process with the highest rate (i.e. the smallest period) is given the highest priority. RM scheduling is a special variant of *fixed-priority* (FP) scheduling that is also frequently used to approximate real-time behavior.

In HeiRAT, these approaches have been extended to account for the two classes of guaranteed and statistical connections as well as for workahead packets. This extension is based on the method of *deadline-workahead* scheduling (Anderson 1993) which dynamically classifies packets with respect to whether they are currently critical or workahead. Within this scheme, one can also easily account for guaranteed and statistical QoS streams. Hence, packets (or rather the processes handling them) are scheduled according to the following multi-level priority scheme:

(1) Critical guaranteed processes
(2) Critical statistical processes
(3) Non-multimedia processes
(4) Workahead processes (both guaranteed and statistical)

Scheduling within these priority classes is (preemptive) RM and EDF (except for Priority 3 where any strategy can be used), the deadline of a packet being its logical arrival time plus delay bound computed for this stream. The priority of a process is switched from 4 to 1 or 2, respectively, as soon as it becomes critical, which possibly entails the preemption of the currently executing process.

With the above scheme the situation may occur that some process is delayed further than expected by the execution of another process that takes more time than specified in its workload description. To avoid this problem, a variant of the algorithm with the following priority scheme can be considered:

(1) Critical processes (guaranteed and statistical)
(2) Critical processes that have used up their processing times as specified by their workload descriptions, but require further processing
(3) Non-multimedia processes
(4) Workahead processes

As soon as a statistical process executed with Priority 1 exceeds its specified processing time it is moved to Priority 2 and possibly preempted. Thus, misbehaving statistical processes cannot violate the QoS assurances given to guaranteed processes. This approach requires the supervision of processing times and increases the complexity of the implementation. Additionally, special care must be taken for a proper sequencing of packets on statistical connections, as a later arriving packet processed with Priority 1 might overtake an earlier packet waiting with Priority 2.

The cost of priority-driven scheduling is determined by several factors. Besides the scheduling decision itself (i.e., the selection of the process to be executed) the scheduling overhead includes the assignment of priorities to the processes, context switching and the use of timers, as needed in the above scheduling schemes. Hence, it has to be carefully decided whether the process priorities are assigned statically or dynamically and at which instants process contexts may be switched. The decision about preemptive or non-preemptive scheduling has also an immediate effect on the overhead (Mercer and Tokuda 1991).

The CPU scheduler currently implemented in HeiRAT is based on the first priority scheme presented above with preemptive fixed-priority RM scheduling. To avoid the overhead incurred by the processing of workahead packets, the scheduler can leave these packets in a wait state until their logical arrival time. Hence, Priority 4 may not be used. A description and evaluation of the scheduler is given in (Wolf, Burke and Vogt 1996).

4.2 Network Access Scheduling

Changing the resource scheduler for network access is not as easy as for the CPU: The order in which packets are sent on the network is determined by an *internal scheduler* on the adapter that can often not be modified. To deal with the problem, one typically has to implement an *external scheduler* that submits packets to the adapter according to their urgency.

4.2.1 External Schedulers

As the internal scheduler is determined by hardware and microcode of the adapter, it generally implements only a simple non-real-time scheduling strategy (for example, FIFO). Hence, it is desirable to reduce its impact and leave most of the task to the external scheduler. This is done by bounding the number of packets that can be queued on the adapter. Such a bound implies a trade-off between the delay that can be guaranteed to a stream and the overall performance of the network. On the one hand, it is desirable to have only a small number of packets on the adapter (one packet in the ideal case) to be able to send newly arrived urgent messages faster. On the other hand, the caching of packets on the adapter increases throughput.

When a packet arrives from the regulator it is inserted into the external scheduler queue according to its priority. If the number of packets waiting on the adapter falls below the threshold, packets are copied from the external scheduler queue to the adapter queue until the threshold value is reached. This is done either when a new packet arrives and there is still room in the adapter queue, or when the adapter indicates the successful transmission of one of its packets.

The external scheduler can work similar to the CPU scheduler: Packets from guaranteed streams have the highest priority followed by packets from statistical streams. Normal data packets rank last. Within the first and second classes, a real-time scheme such as EDF or RM can be used. In contrast to the CPU, scheduling here is non-preemptive – once a packet has been submitted to the adapter, its transmission is not aborted.

4.2.2 Token Ring Access Scheduling

The Token Ring was the first network to which HeiRAT QoS management was applied. It was chosen for its deterministic behavior which lends itself well to the calculation and enforcement of QoS guarantees. Let us use this as an example to discuss the work of the various schedulers.

In order to provide real-time services on the Token Ring, it is necessary to bound the transmission time of a node on receipt of the token. Although the Token Ring MAC protocol provides a limit of 10 ms on the transmission time per node, this value is obviously too high to support multimedia services with tight delay bounds. Hence, the choice of a smaller value is proposed. The limit on the transmission time could, for example, be chosen such that the calculated delay bounds of multimedia streams stay below the most stringent application delay requirements. We denote this value of the transmission time as *T*. *T* can easily be imposed by restricting the packet sizes submitted to the adapter, assuming that only a limited number of packets is transmitted per token visit.

An important consequence of this approach is that small values of *T* result in small packet sizes and hence introduce delays in disassembly and assembly of, for example, large video frames. Also, there is an increase in the number of interrupts generated at the system interface to the adapter on completion of transmission. Hence, it should be ensured that *T* is not too small. For purposes of further discussion, we assume that a suitable value can be chosen – a message size of 4 KByte, for example, will result on a 16 MBit/s Token Ring in a *T* value of approximately 2 ms.

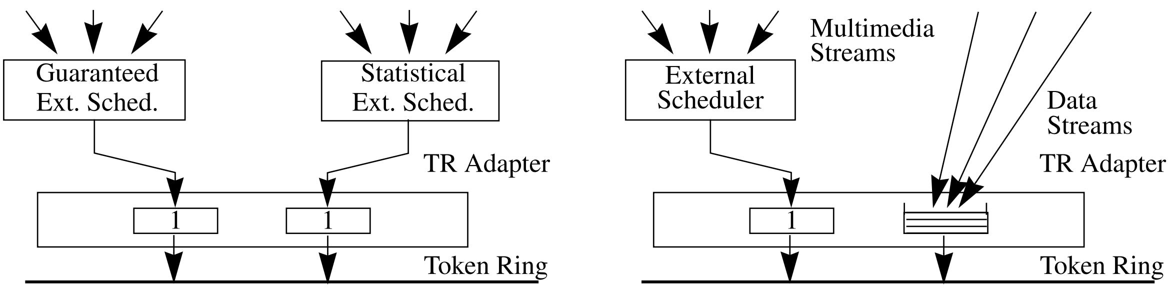

Figure 5: Token Ring Adapter and External Scheduler.

The basic Token Ring access scheme includes a priority mechanism at the MAC layer. The token is associated with a certain priority and stations that receive the token can only transmit packets that are of higher or equal priority than the current token priority. Furthermore, reservations can be made in the token or in passing packets for a pending packet of a certain priority. Reservations are useful in that they prevent stations from transmitting lower-priority packets when there is a higher-priority packet pending for transmission at one station.

The MAC priority is reflected in the internal scheduler of a Token Ring adapter and can be used by the external adapter in one of the following ways:

- *Scheme 1* (left of Figure 5): We assume a sender can actually set MAC priorities for outgoing packets or at least can distinguish between two FIFO queues serving packets with two different MAC priorities. One possibility here is to have two separate external schedulers for guaranteed and statistical connections. A *statistical scheduler* submits packets to the adapter at a lower MAC priority than the *guaranteed scheduler*. Time-independent packets also go through the statistical queue, but are put back by the external scheduler in favor of multimedia packets.
- *Scheme 2* (right of Figure 5): As an alternative, one could have a single external scheduler for both guaranteed and statistical streams. This scheduler prefers guaranteed to statistical connections, similar to the first CPU scheduling scheme described. The scheduler submits both guaranteed and statistical packets with the same high MAC priority to the adapter. Time-independent packets are transmitted through the second queue with a low MAC priority.
- *Scheme 3* (Figure 6): Adapters with only one queue needed to be used in the current implementation of HeiRAT. They bear the closest resemblance to the CPU scheduler: The external scheduler needs to determine the total order of packets

The MAC priority scheme of the Token Ring can also be used to give *all* traffic from one station a higher priority than traffic from other stations. This could be used for video servers where only one (or few) senders actually generate multimedia streams.

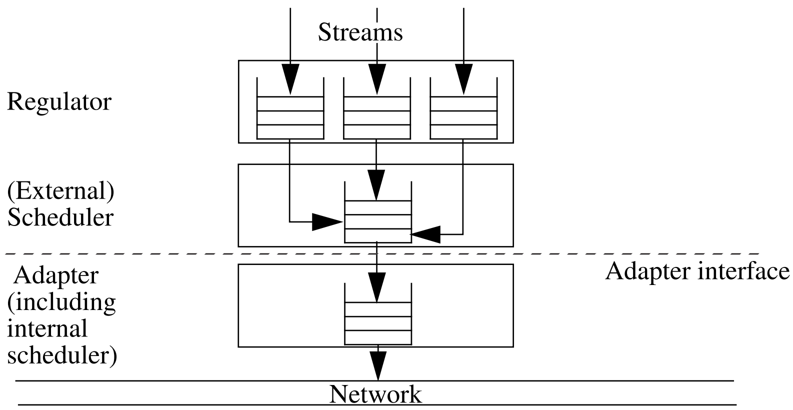

Figure 6: The Way of Packets from the DLS Interface to the Network.

5 Throughput Test and QoS Calculation

Knowing how resources are scheduled, we are now able to address the issues of QoS calculation and resource reservation for both local resources and networks.

5.1 Local Resource Management

The local resources in the network nodes managed by HeiRAT are CPU processing bandwidth and buffer space. Whereas the reservation of buffer space requires only the calculation of the amount of storage needed and a corresponding static buffer allocation at connection establishment time, CPU management must include the full set of functions defined in Section 2, i.e. throughput test, QoS calculation and resource reservation.

5.1.1 CPU Throughput Test

All multimedia processes to be handled by the CPU are characterized by the LBAP model, i.e. they are basically periodic. Moreover, these processes are scheduled by either an RM or an EDF strategy with a preference to processes with a guaranteed QoS. Hence, the throughput test needed can be directly taken from classical scheduling theory (Liu and Layland 1973). A new stream can be accepted by the CPU if

$$\sum_i R_i \cdot P_i \leq U$$

In this inequality, the index i runs through all existing streams and also the new stream. R_i denotes the maximum packet rate of connection i, P_i its processing time per packet, and U a non-negative real number of at most 1 for EDF scheduling and ln(2) for RM scheduling. In practice, it is advisable to restrict the utilization U to values smaller than 1 in order to provide some CPU capacity to non-multimedia processes.

5.1.2 CPU QoS Calculation

QoS calculation in HeiRAT means to optimize one of the three QoS parameters while the other two are fixed. For the CPU, this optimization proceeds as follows:

- *Throughput*: Given the current CPU utilization and the maximum CPU utilization as defined in the throughput test, the maximum additional throughput rate that can be supported by the CPU is computed easily from the above formula. The maximum packet size is obtained by table look-up, and the minimum actual transit time is the processing time P_i. A local delay bound has to be calculated from the this throughput value (see below). At the target host, the sum of these local delay bounds is used to check whether the user's end-to-end delay requirements can be met.
- *Delay*: Under both EDF and RM scheduling, the regular delay of a packet will never exceed $1/R$ when the throughput test holds (Liu and Layland 1973).[1] Hence, $1/R$ is a suitable delay value. For deadline-based scheduling, worst-case simulation is an alternative way to calculate delay bounds (cf. (ANSI 1991)), but currently not implemented in HeiRAT. Under rate-based scheduling, worst-case simulation is not useful since the subsequent establishment of other streams with higher rates causes the calculated delay to increase for lower-rate streams.
- *Reliability*: If sufficient buffer space to store all generated or incoming packets is reserved, no loss due to buffer overflow can occur for guaranteed streams. Hence, even the most restrictive reliability requirement (Class 4) will trivially be met. The local delay bound is calculated as described above.

5.2 Network Resource Management

Managing networks is special in that the various entities accessing the network may have a different view on its utilization and remaining bandwidth. *Local bandwidth allocation techniques* use only knowledge available on their system. We distinguish between two local techniques:

- The *bandwidth counter approach* allocates network bandwidth as long as the sum of the bandwidth requirements of the individual streams does not exceed a given bound.
- The *calculation-based approach* takes additional information into account, especially delay characteristics, thus managing not only the throughput but the full range of QoS parameters.

The advantage of a local approach is that the QoS can be calculated without requiring to collect knowledge about the streams sent by the other stations. However, the obtainable results can be unsatisfactory, when the number of stations connected to the ring is large. The assumption for the Token Ring, for example, is that *all* other stations always hold the token for the maximum time, although some of them might temporarily transmit no data at all. Hence, the bandwidth allocatable will be small and the delay will be high.

The problem can be solved by *global bandwidth allocation techniques* that look at the requirements of all stations. Such global allocation could be implemented in a distributed fashion. HeiRAT has chosen a simpler albeit less fault-tolerant approach and uses a centralized technique.

While the global and the calculation-based approach allow for a guaranteed QoS, the bandwidth counter technique yields only a statistical QoS (reasons for this are discussed below). Thus, it is especially appropriate for networks such as the Ethernet where due to the non-determinism of network access no guarantees can be given anyway.

1. This holds for guaranteed connections under both variants of the priority scheduling scheme described in Section 4.1. For statistical connections that stick to their specified workload bounds, this delay bound is valid under the second priority scheme.

In the following, we again use the Token Ring as an example for a network resource to be managed. We concentrate on local allocation techniques first and at the end move to a global scheme.

5.2.1 Processing Times on the Token Ring

To derive a throughput test and QoS calculation for the Token Ring, one needs to know the streams to be transmitted from the station under consideration and the maximum time it takes to transmit a packet once it is on the adapter. The time interval from packet submission to the adapter and its transmission completion is composed of four components:

- the time to copy the packet to the adapter C,
- the time to access the token A,
- the time for transmission of the packet T, and
- the packet propagation delay τ across the ring.

We will refer to this total time as the *processing time P* of the packet. This term is chosen to highlight the similarities to the CPU scheme discussed previously: In this analogy, a packet is "processed" by the Token Ring adapter from the instant when it is copied to the adapter until the instant when its transmission is completed. τ includes the signal propagation delay around the ring and the bit delays introduced at each station. C, T and τ are fixed given the particular ring configuration and the maximum packet size. A, which reflects the network bandwidth used by other stations, is variable. Its worst case needs to be computed.

For a scheduler using separate queues for guaranteed and statistical streams or, alternatively, for multimedia and other data traffic (see Figure 5), the worst-case access delay occurs when the guaranteed packet is copied to the adapter only to find another packet (from a statistical or a normal data stream) whose transmission has just started. Since a station can transmit only one packet per token visit, the transmission of the guaranteed packet must wait for the return of the token. However, prior to its return as many as N-1 stations could transmit, N being the number of stations on the ring. Hence, the worst-case access delay is given as $A = N*T + \tau$. Using this value for A, the processing times for all packets are constant and identical.

For a scheduling scheme using one adapter queue for all traffic (see Figure 6) it is easily shown that the worst-case access delay is $A = (N-1)*T+\tau$. This is due to the fact that there is no other external scheduler on the same station that interferes with the guaranteed streams and hence the only access delay incurred is due to the other $N-1$ stations.

If multimedia traffic is transmitted at a higher MAC priority than data traffic the impact of stations with only data traffic (called *data stations* in the following) can be greatly reduced. For such a mixed scenario, it can be shown for both scheduling schemes that the worst case value of $A = (N'+1)*T+\tau$, N' being the number of stations with multimedia traffic.The formula assumes that the transmission time of the other stations is bounded by T. If not, the access delay is somewhat higher. The additional delay incurred by data stations is due to the fact that in the worst case two data stations can transmit before any multimedia station can get hold of the token. This can be shown as follows: Consider the scenario when a data station is transmitting its information and the multimedia station frame arrives just too late on the adapter to make a reservation at a higher priority, i.e., the header of the data frame has just passed by the multimedia station. Now, a second data station may transmit its frame before the multimedia station makes a successful reservation. Afterwards, only multimedia stations can transmit and there are N'-1 of these stations excluding the one under consideration.

In contrast to guaranteed connections, the processing times for statistical packets are not computed on a worst-case basis. The goal here is to obtain estimates or to compute lower processing times to support large traffic loads and offer an inexpensive, but reasonable QoS. To achieve this goal, for statistical packets a fixed estimate of the *token rotation time* (TRT) is used that gives the length of the time interval between two successive transmissions of statistical frames from this station. This estimate is affected by the number of higher-priority frames of the guaranteed streams transmitted between the statistical frames – hence the TRT estimate for statistical streams is not necessarily smaller than the TRT for guaranteed streams.

In a more elaborate scheme, the estimate would not be fixed, but updated with each transmission completion in the following manner. Let TRT_{New} be a new estimate of the TRT, TRT_{Old} the old estimate and TRT_{Meas} the newly measured value of the TRT. Then the estimate can be updated as $TRT_{New} = a* TRT_{Old} + (1-a)* TRT_{Meas}$, where $0 \le a \le 1$ determines the sensitivity of the estimate to new observations.

5.2.2 QoS Calculation for the Token Ring

A main difference of the network resource compared to the CPU is its non-preemptive scheduling. In (Nagarajan and Vogt 1992) we derived a throughput test and algorithms for QoS computation under non-preemptive scheduling. The most important results of this work are presented in the Appendix. They allow for the calculation of throughput and delay guarantees under the FP, RM and EDF scheduling schemes.

Based on these findings, QoS can be calculated depending on the parameter to be optimized as follows:

- *Delay optimization*: For FP and RM scheduling, delay and throughput are computed through the expressions in A.3. If the throughput obtained through these expressions is larger than the desired value R, the throughput test has succeeded. However, only R rather than the maximum supportable value is guaranteed. For EDF and RM scheduling, the expressions in A.2 can be used provided the processing times for all streams are identical (as it would be the case using the above calculations). Again, if the maximum supportable throughput is larger than the pre-specified throughput R, then a throughput of R and a delay of $1/R + P$ is guaranteed where P is the common packet processing time for all streams.
- *Throughput optimization*: Again, the expressions in A.3 apply to the FP and RM schemes. If the maximum supportable throughput value is larger than the given desired value R, then again only R is guaranteed. If, however, the throughput value that can be supported is smaller than the desired value but larger than the worst-acceptable value, then this throughput value is guaranteed. Finally, if it is smaller than the required value, the stream will have to be rejected. Throughput optimization for the EDF scheme is done as described for EDF on the CPU, using the formulae in A.2.
- *Reliability optimization*: The IBM Token Ring Busmaster adapter can indicate the successful or unsuccessful delivery of packets to the transmitting adapter. Error correction, however, cannot be provided by the Token Ring without retransmissions. Thus, the best reliability class supported by the Token Ring is Class 2.

The delay values as calculated above may be relatively small. We suggest that these values be relaxed slightly as guaranteeing a small delay bound to a low-priority stream might prevent the admission of higher priority streams in the future.

5.2.3 Guaranteed vs. Statistical QoS on Token Ring

The above QoS calculations are based on a number of worst-case assumptions. Together they determine the maximum possible length of time between the arrival of a high priority multimedia packet at the driver interface and the end of its transmission across the ring. These assumptions were:

(1) When the copying of a packet to the adapter is completed, a free token has just passed.
(2) All multimedia stations on the ring always hold the token for the maximum possible time.
(3) It takes the maximum time until a token reservation for a packet with a high MAC priority becomes effective, i.e., two data stations send for the maximum possible time before a high-priority multimedia packet gets the token.

Each of these assumptions is indispensable for guaranteed QoS. However, in configurations with many multimedia stations, they result in long processing times and delays. They may cause that only a relatively small fraction of the actually available Token Ring bandwidth can be allocated to guaranteed connections.[2] Hence, a supplementary scheme for statistical QoS is required that can allocate the remaining bandwidth. Such schemes could relax the assumptions in the following way:

- If calculations are made on a global, rather then local basis, better estimates for the maximum TRT can be achieved. We discuss the central bandwidth allocator for this below.
- If there are no data stations on the ring the processing time can be reduced as indicated in Section 5.2.1. This still yields a guaranteed QoS. If there are data stations that put only a low load on the ring it can be assumed that they do not interfere much with the multimedia traffic. In this case, working with a reduced processing time yields a reasonable statistical QoS.

2. We want to emphasize that the scheme for guaranteeing QoS on Token Ring can be easily transferred to FDDI. FDDI synchronous mode is able to guarantee low upper bounds for the TRT thus greatly reducing the worst-case processing time for a packet and increasing the bandwidth available for guaranteed connections.

- As described in Section 5.2.2, statistical QoS calculation can be based on a TRT estimate. Although HeiRAT currently provides no monitoring function that keeps statistics on the TRT, a function is available which calculates the processing time based on a user-provided estimate for the token access time. The most optimistic assumption is that a token is immediately available after data is copied to the adapter. For a video server transmitting data on an otherwise unused ring, this assumption is justified. Here the access delay can be completely dropped from the sum defining the processing time. If there can be more than one frame on the adapter, copying and transmitting would proceed in parallel and one could even drop the copying time, thus reducing the processing time to the pure transmission time.

The above approaches reduce processing times, get a better estimate of the behavior of the other stations and the ring itself and relax assumptions (1)–(3). They can, however, not get around the problem of a temporary priority inversion: When a packet arrives at the adapter driver interface, another packet with a lower scheduling priority is just being copied to the adapter. This problem cannot be avoided due to the non-preemptiveness of the sending process.

5.2.4 Bandwidth Counting for Token Ring

A radically different (and simplifying) approach to manage statistical connections does not use the above formulae at all, but considers only the raw throughput required by the connections (Baugher et al. 1993). In this bandwidth counter approach, a new stream is admitted only if the total throughput needed by all streams lies below the total capacity of the ring (for example, 16 MBit/s or a lower value, if some residual bandwidth for other traffic is desired) or – alternatively – some capacity allocated to this station.

The advantage of this scheme is that all the raw Token Ring bandwidth available can be allocated to multimedia connections. It also permits to relax the requirement that the Token Ring driver never holds more than one frame on the adapter, provided that all connections are only statistical. The drawback of the bandwidth counter approach is that no tight delay bounds can be given.

In the current implementation of HeiRAT, the user can select whether statistical connections are managed by the calculation-based or the bandwidth counter approach. Hence, it is possible to combine the original guaranteed approach based on the formulae and the statistical approach based on a bandwidth counter. This way, one can establish a small number of guaranteed connections with tight delay guarantees and also to utilize the full ring bandwidth by admitting additional statistical connections.

5.2.5 Central Bandwidth Allocation

When the above calculations are not just made on a single station, but by an entity that knows about all existing requirements throughout the system, potentially better QoS can be achieved. The HeiRAT *central bandwidth allocator* (CBA) (Jordaan, Paterok and Vogt 1993) is such an entity that is located somewhere in the network and keeps track of stations that currently submit multimedia data to the ring. It can currently be used with bridged LANs consisting of Token Ring, Ethernet, or FDDI.

The central piece of CBA is the *QoS allocator*. It includes a Management Information Base (MIB) storing network topology information, the bandwidth currently available, other QoS parameters, and the current status of the resource reservations for individual streams. The allocator is able to receive requests for a certain QoS, to decide whether these requests can be granted or have to be rejected, to calculate QoS guarantees, to reserve corresponding resource capacities and to return QoS guarantees. On each network (single-segment or bridged) there can run only one allocator.

The allocator communicates with QoS requestors on the individual stations. This communication supports the registration of requestors with the allocator, the allocation, deallocation and change of QoS, and the refreshing of QoS reservations. QoS requestors act on behalf of QoS clients (for example, HeiRAT agents). A requestor offers functions such as initialization of the requestor, establishment, change and release of reservations. It thus shields the clients from the direct communication with the allocator by providing the protocols and the frame formats required for this communication.

The QoS allocator includes a graphical user interface which allows to feed in all needed configuration information (such as LAN topology and capacity, IP addresses of the QoS requestors). It can be used to monitor the current resource allocation status. Configuration and monitoring can be done via SNMP.

Figure 7: QoS Requestor and Allocator.

6 Closing Remark

In this paper, we have shown the full range of HeiRAT functions from QoS modelling, to QoS enforcement to QoS calculation as they have been implemented in the HeiProjects over the past years. While our initial work was mainly concerned with providing guaranteed QoS, we have gradually shifted our attention to also support statistical QoS as is visible in the text above. We also looked at alternative approaches to provide QoS such as scaling (Delgrossi et al. 1994) and filtering (Wolf, Herrtwich and Delgrossi 1995) and have implemented them in our system.

We have found that these various methods are no substitutes, but rather complement each other depending on the requirements of the applications that need to be supported. We therefore believe that even with growing system and network bandwidth, there is a role for careful QoS calculation and reservation techniques: As bandwidth goes up, so does demand. The modular HeiRAT framework is open to be extended by QoS mechanisms for further, more modern networks. The results we have presented for Token Ring management can be easily transferred to other token-based networks like FDDI. Switched networks like ATM require the design of own methods. However, where our design had to graft real-time mechanisms on the underlying system, future platforms may benefit from the inclusion of HeiRAT-like mechanisms in the system from the ground up.

7 Acknowledgments

We would like to thank our colleagues at IBM for their contributions to HeiRAT. Luca Delgrossi, Frank Hoffmann and Sibylle Schaller worked on the ST-II protocol implementation. Wolfgang Burke implemented the first version of the CPU scheduler under AIX and Andreas Mauthe did the same for OS/2. Ramesh Nagarajan devised the method for calculating and enforcing QoS guarantees for the Token Ring. Marcel Graf and Barbara Twachtmann implemented the Token Ring access scheduler under AIX; Ralph Demuth developed the OS/2 version. Sandhya Nagarajan programmed functions for the transformation of QoS values between the different layers. Derick Jordaan worked on bandwidth allocation issues. Last, but not least, Martin Paterok provided many valuable comments on our work.

8 References

Anderson DP, Herrtwich RG, Schaefer C (1990) *SRP: A Resource Reservation Protocol for Guaranteed-Performance Communication in the Internet*. TR-90-006, ICSI, Berkeley

Anderson DP (1993) *Metascheduling for Continuous Media*. ACM Transactions on Computer Systems, Vol. 11, No. 3

American National Standards Institute (1991) *Integrated Services Digital Network (ISDN) - Digital Subscriber Signaling System No.1 (DSS1) - Signaling Specification for Frame Relay Bearer Service,* ANSI T1.617-1991

Baugher M, French S, Stephens A, Van Horn I (1993) *A Multimedia Client to the IBM LAN Server.* ACM Multimedia '93, Anaheim

Case JD, Fedor MS, Schoffstall ML, Davin JR (1990) *A Simple Network Management Protocol*. Internet RFC 1157

Cidon I, Gopal I, Guérin R (1991) *Bandwidth Management and Congestion Control in plaNET.* IEEE Communications Magazine, Vol. 28, No. 10

Cruz RL (1991) *A Calculus for Network Delay, PART I: Network Elements in Isolation*. IEEE Transactions on Information Theory, Vol. 37, No. 1

Delgrossi L, Halstrick C, Hehmann D, Herrtwich RG, Krone O, Sandvoss J, Vogt C (1994) *Media Scaling in a Multimedia Communication System*. ACM Multimedia Systems, Vol. 2, No. 4

Delgrossi L, Halstrick C, Herrtwich RG, Stuettgen H (1992) *HeiTP: A Transport Protocol for ST-II.* GLOBECOM'92, Orlando

Delgrossi L, Herrtwich RG, Vogt C, Wolf L (1993) *Reservation Protocols for Internetworks: A Comparison of ST-II and RSVP.* Fourth International Workshop on Network and Operating System Support for Digital Audio and Video, Lancaster

Delgrossi L, Berger L (1995) *Internet STream Protocol Version 2 (ST2) – Protocol Specification – Version ST2+*. Internet RFC 1819

Herrtwich RG (1994) *Distributed Multimedia Solutions from the HeiProjects*. In: J.L. Encarnacao, J.D. Foley (Eds): *Multimedia – System Architectures and Applications*, Springer

Jordaan D, Paterok M, Vogt C (1993) *Layered Quality of Service Management in Heterogeneous Networks*. IBM European Networking Center, TR 43.9304, Heidelberg

Kätker S, Paterok M, Vogt C, Wittig H, Delgrossi L (1993) *An SNMP MIB for the ST-II Protocol and the Heidelberg Resource Administration Technique.* IBM European Networking Center, TR 43.9314, Heidelberg

Liu CL, Layland JW (1973) *Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment*. Journal of ACM, Vol. 20, No. 1

Mercer CW, Tokuda H (1991) *Priority Consistency in Protocol Architectures*. Second International Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg

Mitzel DJ, Estrin D, Shenker S, Zhang L (1994) *An Architectural Comparison of ST-II and RSVP*, IEEE Infocom '94

Nagarajan R, Vogt C (1992) *Guaranteed-Performance Transport of Multimedia Traffic over the Token Ring*. IBM European Networking Center, TR 43.9201, Heidelberg

Taber D (1993) *Multi-Media Network Resource Reservation (NRR), Functional Specification, Level 0.1*. IBM Networking Systems, TR NRR-FS-001, Raleigh

Topolcic C (Ed.) (1990) *Experimental Internet Stream Protocol, Version 2 (ST-II)*. Internet RFC 1190

Vogt C (1995) *Quality-of-service management for multimedia streams with fixed arrival periods and variable frame sizes*. ACM Multimedia Systems, Vol. 3, No. 2

Vogt C, Herrtwich RG, Nagarajan R (1993) *HeiRAT: The Heidelberg Resource Administration Technique - Design Philosophy and Goals*. Kommunikation in Verteilten Systemen, Munich

Vogel R, Herrtwich RG, Kalfa W, Wittig H, Wolf LC (1995) *QoS-Based Routing of Multimedia Streams in Computer Networks*, accepted for publication Special Issue of IEEE JSAC on Distributed Multimedia Systems and Technology

Wolf LC, Burke W, Vogt C (1996) *Evaluation of a CPU Scheduling Mechanism for Multimedia Systems*, Software – Practice and Experience, Vol. 26, No. 4

Wolf LC, Herrtwich RG, Delgrossi L (1995) *Filtering Multimedia Data in Reservation-Based Internetworks*. Kommunikation in Verteilten Systemen, Chemnitz

Wolf LC, Herrtwich RG (1994) *The System Architecture of the Heidelberg Transport System*. ACM Operating Systems Review, Vol. 28, No. 2

Wolf LC (1996) *Resource Management for Distributed Multimedia Systems.* Kluwer, Boston

Wittig H, Wolf LC, Vogt C (1994) *CPU Utilization of Multimedia Processes: The Heidelberg Predictor of Execution Tool*. Second International Workshop on Advanced Teleservices and High Speed Communication Architectures, Heidelberg

Zhang L, Deering S, Estrin D, Shenker S, Zappala D (1993) *RSVP: A New Resource ReSerVation Protocol*, IEEE Network

Appendix A QoS Calculation Under Non-Preemptive Scheduling

This Appendix contains theorems for QoS calculation under non-preemptive scheduling, which can be used for QoS management of the Token Ring (see Section 5.2.2). At this place, the proofs for the theorems are omitted due to space limitations. They can be found in (Nagarajan and Vogt 1992).

In general, we address the problem of the non-preemptive scheduling of N periodic streams with periods $T_1, T_2 \ldots T_N$ (i.e., rates $R_i = 1 / T_i$) and deadlines d_i smaller than or equal to the period.

A.1 Fixed Priority Non-Preemptive Scheduling of Periodic Streams

First, we consider the *fixed-priority* (FP) scheduling scheme, i.e., each stream is assigned a unique priority value and streams are scheduled according to this priority. One special case of this scheme is the *rate-monotonic* (RM) scheduling strategy. Our aim, here, is to formulate a throughput test in order to ensure that all deadlines are met for all packets of the streams.

Theorem 1: Given N periodic streams with periods $T_1, T_2 \ldots T_N$ and deadlines $d_i \le T_i, 1 \le i \le N$. Assume that the streams are numbered by increasing priorities, i.e., that stream N has the highest priority and stream 1 has the lowest. Also, assume that processing times of the stream packets are unity (i.e., identical). Then, the streams are schedulable within their deadlines with the non-preemptive fixed-priority scheme if

$$d_N \ge 2$$

$$d_i \ge 2 + C(d_i, T_{i+1}) + \ldots + C(d_i, T_N) \qquad 1 \le i < N$$

$$\text{where} \qquad C(x, y) = ceil\left(\frac{x-1}{y}\right) + 1$$

(ceil(z) is the smallest integer k with $z \le k$.)

For processing times that are not identical we have

Theorem 2: Given N periodic streams with periods $T_1, T_2, \ldots T_N$ and deadlines $d_1, d_2, \ldots, d_N$ with $d_i \le T_i, 1 \le i \le N$. Assume that the streams are numbered by increasing priorities, i.e., that stream N has the highest priority and stream 1 has the lowest. *Also,* assume that the processing time of the packets from stream i is P_i. Then, the streams are schedulable within their deadlines with the non-preemptive fixed-priority scheme if

$$d_N \ge P_N + max_{1 \le i \le N} P_i$$

$$d_i \ge P_i + max_{1 \le j \le N} P_j + \sum_{j=i+1}^{N} P_j \cdot F(d_i - R_j, T_j) \qquad 1 \le i < N$$

$$\text{where} \qquad F(x, y) = ceil\left(\frac{x}{y}\right) + 1$$

The discontinuities induced by the ceiling function make the calculation of optimum QoS values somewhat tedious (see A.3). Alternative expressions without these discontinuities are given by

Theorem 3: Given N periodic streams with periods $T_1, T_2, \ldots, T_N$ and deadlines $d_1, d_2, \ldots, d_N$ with $d_i \le T_i$, $1 \le i \le N$. Assume that the streams are numbered by increasing priorities, i.e., that stream N has the highest priority and stream 1 has the lowest. Also, assume that the processing time of the packets from stream i is P_i. Then, the streams are schedulable within their deadlines with the non-preemptive fixed-priority scheme if

$$d_N \ge P_N + max_{1 \le i \le N} P_i$$

$$d_i \ge P_i + max_{1 \le j \le N} P_j + \sum_{j=i+1}^{N} P_j \cdot G(d_i - P_j, T_j) \qquad 1 \le i \le N$$

$$\text{where} \qquad G(x, y) = \frac{x}{y} + 2$$

A.2 EDF and RM Non-Preemptive Scheduling of Periodic Streams

We now consider throughput tests for non-preemptive EDF and RM scheduling. These tests are extensions of the expressions in (Liu and Layland 1973) concerning the corresponding preemptive scheduling schemes.

Theorem 4: Given N periodic streams with periods $T_1, T_2, \ldots, T_N$ (i.e., rates $R_i = 1 / T_i$ and unit processing times P per packet. Let $d_i = T_i + P$ be the deadline for stream i. Then, the streams are schedulable within their deadlines with the non-preemptive RM scheme if

$$\sum_{i=1}^{N} \frac{1}{T_i} \cdot P \leq ln(2)$$

For EDF scheduling, the same holds if $\sum_{i=1}^{N} \frac{1}{T_i} \cdot P \leq 1$

A.3 Computation of Optimal Throughput and Delays

Finally, we consider the computation of the optimal throughput and delay values. This optimization is carried out for a single newly established stream given that the guarantees and stream characteristics for all existing streams are known. Such an optimization is needed for example for the computation of QoS values as in Section 5.2.2.

First, we consider the FP scheduling scheme and subsequently the RM and EDF schemes respectively. It can be easily derived from Theorem 3 that the smallest delay bound that can be guaranteed to stream i ($1 \leq i \leq N$) is:

$$d_i = \frac{P_i + (max_{1 \leq j \leq N} P_j) + \sum_{j=i+1}^{N} P_j\left(2 - \frac{P_j}{T_j}\right)}{1 - \sum_{j=i+1}^{N} \frac{P_j}{T_j}}$$

N being the total number of streams. Note that the guarantees for all streams with a higher priority than that of a newly established stream i are not affected by this new stream. We need only ensure that the new stream has no impact on the guarantees given to lower priority streams. The choice of the throughput value of the new stream is crucial in this respect. It can be easily shown that if the period T_i is chosen such that

$$T_i = \max(\max_{1 \leq j < i} L_j, d_i)$$

where

$$L_j = \frac{P_i(d_j - P_i)}{d_j - P_j - 2P_i - \max_{1 \leq k \leq N} P_k - \sum_{k=j+1, k \neq i}^{N} P_k\left(2 + \frac{d_j - P_k}{T_k}\right)}$$

then none of the guarantees for the lower streams will be violated and hence $1 / T_i$ is the highest value of the throughput that can be provided to the new stream i.

Next, we consider the RM scheme specifically. Note that the RM scheme is a special case of the FP scheme and hence the results above are applicable as well. As an alternative, an optimization based on the throughput tests in Theorem 4 is possible. For the RM scheme, we consider two potential scenarios. First, when it is desired to admit the connection at a fixed priority and later the case where the optimization is also over the choice of priority.

RM scheme - fixed priority: Here we assume that the new stream is to be admitted at priority i where $1 \leq i \leq N+1$ and N is the number of existing streams (i.e., streams $1,...,i-1,i+1,...,N+1$ exist already). First, we compute the residual capacity available at this station:

$$U_r = \ln(2) - \sum_{j=1(j \neq i)}^{N+1} \frac{P}{T_j}$$

where P is the common processing time for all streams and T_j are the respective periods. Note that the scheduling test in Theorem 4 does not allow for the possibility of variable processing times (see Section for the motivation of identical processing times). Given the residual processing capacity available, the maximum allowable throughput rate results from the minimum allowable period:

$$T_i = \max\left(\frac{P}{U_r}, T_{i+1}\right) \qquad 1 \le i \le N+1$$

(Take $T_{N+2} = 0$ while computing the above expressions).However, if $T_i > T_{i-1}$ then the new stream cannot be admitted at the given priority because otherwise the given priority ordering would be violated.

Authors' Biographies

CARSTEN VOGT received his diploma in Computer Science from the University of Bonn in 1986 and his doctorate in Natural Sciences from the University of Hamburg in 1990. From 1985 to 1990, he worked at the "Forschungsinstitut fuer Funk und Mathematik", Wachtberg/Bonn, in a project aimed at the development of an experimental object-oriented computer architecture. In 1991, Dr. Vogt joined the IBM European Networking Center, Heidelberg, where he was primarily active on resource management and operating system issues for distributed multimedia systems. Since 1994, he has been a Professor for operating systems at the Fachhochschule of Cologne.

LARS C. WOLF received the diploma degree from the University of Erlangen-Nuernberg in 1991 and the doctoral degree from the Technical University of Chemnitz in 1995, both in computer science. From 1991 to 1996 he worked at IBM's European Networking Center in Heidelberg, Germany, as visiting scientist and research staff member on multimedia transport, resource management and distributed multimedia systems. In 1996 he joined the Technical University of Darmstadt where he builds up a research group working on multimedia communication and mobility support.

RALF GUIDO HERRTWICH studied computer science at the Technical University of Berlin where he received his Ph.D. in 1987. He started work on multimedia communication systems during a sabbatical at the International Computer Science Institute at Berkeley and continued his multimedia projects at the IBM European Networking Center in Heidelberg, Germany, from 1990 till 1995. During this time he was also appointed multimedia business manager for IBM Germany and multimedia technology manager for IBM Europe. Dr. Herrtwich is one of the main editors of the ACM Multimedia Systems Journal and served as program committee member and chairman of various international multimedia conferences. Since 1995 he is director of product management at RWE Telliance in Essen, Germany, a new telecommunications venture operating in the German market.

HARTMUT WITTIG received the diploma degree from the Technical University of Dresden in 1993. In 1993 and 1994 he worked as visiting scientist at the IBM European Networking Center within the Heidelberg Transport System (HeiTS) and Globally Accessible Services (GLASS) projects. Since 1995 he is research staff member at the ENC and Editor in Chief of the IEEE Computer Society Multimedia Newsletter. Hartmut Wittig represents IBM to the Digital Audio-Visual Council (DAVIC) which defines a system standard for digital and interactive television systems.