Advanced Mechanisms for Available Rate Usage in ATM and Differentiated Services Networks

Roland Bless, Dirk Holzhausen, Hartmut Ritter, Klaus Wehrle
Institute of Telematics, University of Karlsruhe
Zirkel 2, 76128 Karlsruhe, Germany
Tel.: +49 721 608 6411, Fax: +49 721 388097
{bless,holzhaus,ritter,wehrle}@telematik.informatik.uni-karlsruhe.de

Abstract

The Available Bit Rate (ABR) service category was defined for ATM networks in order to provide a minimal guaranteed rate while letting a user exploit additional unused capacity from other service categories. In this paper experiences with an implementation of a driver enabling ABR support and the steps needed to integrate it in an end system are described. Furthermore, the deployment of ABR in an xDSL environment is discussed. Additionally, an approach is presented that transfers the idea of using the available rate from ATM to IP networks based on Differentiated Services. A new forwarding behavior is proposed and evaluated by simulations.

1 Introduction

In ATM networks many different service categories are integrated [5]. Commonly used categories are UBR (unspecified bit rate), CBR (constant bit rate) or VBR (variable bit rate). Later in the development of ATM the service category ABR (available bit rate) was introduced; by using ABR the bandwidth not used by other categories can be exploited. The ABR in general is shortly described in section 2, while its integration into an end system is presented in section 2.1. Aspects of using ABR over xDSL techniques are discussed in section 3.

The situation in Differentiated Services IP networks is different: up to now, especially the services Best Effort and Premium Service are well defined, which can be compared with UBR respectively CBR. Nevertheless a service exploiting the available bandwidth, but providing some stronger guarantees than best effort is still missing. A new forwarding behavior providing strong guarantees while allowing to use residual unused bandwidth of other services is proposed in section 4 and evaluated by simulations in section 4.2.

2 Available Bit Rate Service Category

In order to obtain information about the currently available bandwidth, the ABR service inserts resource management (RM) cells into the cell stream from source to destination. ATM network nodes on the path between source and destination monitor the local congestion conditions and use RM cells for providing feedback to the sender. There are at least two ways how this can be done: network nodes may return a RM cell with the congestion indication bit set if there is local overload (binary feedback). RM cells reaching the receiver are passed back to the sender (backward RM cells). So the second way for a network node to inform a sender of congestion is to put information about the currently local available cell rate into these backward RM cells (explicit rate feedback). When a connection is set up, the source starts with an initial cell rate. If no congestion feedback is given, the cell rate is increased, using a predefined rate increase factor. If overload is indicated, the cell rate is again decreased according to the rate decrease factor. The result of the ABR mechanism is a fair share of the residual bandwidth not reserved by other service categories like CBR etc.: if a connection using ABR is idle for a moment another ABR source can increase its bandwidth; if the idle connection starts to transmit cells again, the second source will decrease its rate. The main advantages of introducing dedicated resource management cells are that the time scales for notifications are very short and that the coupling between end system and network is therefore very tight, if compared for example with congestion control mechanisms realized in TCP, which adapt in the time scales of many round trip times.

2.1 Integration of ABR Resource Management in the End System

In order to evaluate the defined ABR mechanisms the Linux driver for ATM NICs presented in [8] was modified to support generation and analysis of RM cells. The source code of the driver will be made available soon at [1], based partly on an existing driver [2]. The driver supports generation of the RM cell format and its insertion into an ABR cell flow. The socket API was patched in order to provide a way to specify ABR-related parameters such as increase and decrease factors. This was done by extending the setsockopt call. Received RM cells are parsed and the congestion indication or explicit rate information is passed to an RM analysis module.

The analysis module interacts with a traffic shaper working at cell level, which adapts the cell rates of the ABR connections to the currently available and admitted rate. Figure 1 shows this interaction. The traffic shaper uses fine grained clocks and timers as described in [8].

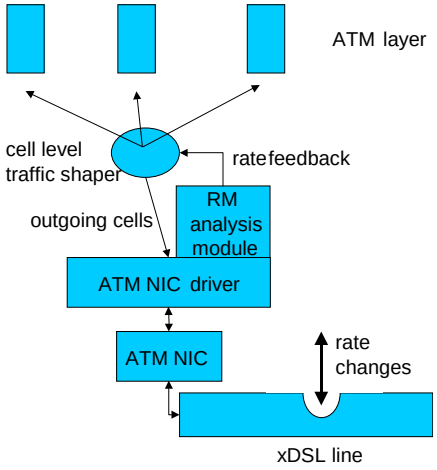

Figure 1 Integration of the RM analysis module and the cell level traffic shaper.

The traffic shaper is available at [1], as well as measurements about throughput and CPU load of the shaper and its interaction with the ABR mechanisms.

3 ATM over xDSL

The discussed ABR mechanisms can be deployed very well in the interaction with the xDSL physical transmission technology. The xDSL transmission technology provides a highrate data path targeted especially at private customers and small businesses. Due to the fact, that the xDSL technology exploits the simple copper wire using complex coding schemes, it is very sensitive to changes of the physical characteristics of the wire. Especially electromagnetic interferences on the large cable trunks between the home and the access concentrator at the provider result in varying transmission rates, ranging from 1.5 Mbit/s to 8 Mbit/s, for example. In the ATM Forum, for example in [6], different ways of using ABR mechanisms in connection with an ADSL link were discussed. The use of RM cells informs the server about congestion situations on the ADSL link from the provider to the client at the private customer. This client/server scenario is very typical for private users browsing in the web and requesting file downloads. Nevertheless, care must be taken that in this 'TCP/IP over ATM'-scenario the TCP sources at the sender are throttled back when a reduced rate was negotiated. In order to fully exploit the bandwidth of the connection, a fast adaptation to increasing rates as well as decreasing rates is needed. So in addition to the interaction between ABR and the traffic shaper described in section 2.1, TCP should be modified to fit into the ABR service category with its highly varying rates. Therefore, we propose a modified TCP called 'Stop-and-Go TCP', presented in the following section.

3.1 Stop-and-Go TCP

The adaptation mechanisms of TCP such as flow control and congestion control are not suited for an environment with guaranteed rates, as shown in [4]. As said before, the adaptation speed of TCP is slow compared to the rate changes on the ADSL link. Thus, a situation might result in which a TCP connection using the ABR service category does not exploit the suddenly increased rate or might decrease the rate too much just due to a temporary fall back of the available rate. One solution could be to pass the available rate information from the RM analysis module directly up to the TCP layer. This would require many different timers running in the system, and nevertheless a coordination function would be needed. This coordination function is already realized in the traffic shaper, which is informed about every rate change indicated by a RM cell, and which holds a context for every connection anyway.

The traffic shaper controls TCP by sending simple signals to change the state of this Stop-and-Go TCP rather than updating some QoS parameters in TCP. Since TCP does not know anything about QoS parameters but is controlled by the traffic shaper only very few changes to the TCP code are necessary. The standard TCP congestion control mechanisms – slow start, congestion avoidance, fast retransmit and fast recovery – are replaced by a simple Stop-and-Go mechanism controlled by the traffic shaper. The Stop-and-Go mechanism is realized by two states defined within Stop-and-Go TCP (SG-TCP). Every connection is started in state SG-Go which means that TCP can send packets. As long as the advertised window is open TCP will send out packets as fast as possible (the flow control mechanisms remain unmodified). All packets of a single connection are queued in a per connection output queue within the traffic shaper. The traffic shaper generates an output stream in conformance with the given QoS parameters. This mechanism allows TCP to use the complete data rate right after connection setup.

As soon as the traffic shaper's output queue length exceeds a predefined upper limit the traffic shaper sends a stop signal to TCP. This causes the TCP connection to switch to the SG-Stop state. In this state TCP will not send out any more packets but queue them in its own output queue. After some time the traffic shaper's output queue length will drop below a predefined lower limit which causes the traffic shaper to send a go signal to TCP. This signal resets the TCP state to SG-Go. Modifications of the source code are also available at [1].

4 Usage of Available Rate in Differentiated Services IP Networks

Integration of new and better network services is one of the most discussed topics in the Internet community. The Differentiated Services architecture [3] is one promising approach to provide guaranteed services. This architecture allows to use various per-hop forwarding behaviors (PHB) in routers to achieve different packet treatment of particular (mostly aggregated) flows. Nevertheless, currently predominantly the Expedited Forwarding (EF) forwarding behavior [7] seems to be widely accepted. It constitutes a basis for a service comparable to the CBR service category in ATM networks with minimal end-to-end delay. Similar to the CBR service category, the rate reserved for flows using Expedited Forwarding will usually not be completely exploited. As shown in [4], TCP of-

ten does not utilize the total amount of reserved rate. In ATM networks, the UBR and ABR service categories will profit from this situation, in Differentiated Services IP networks the available rate will be normally used by Best Effort flows. The Best Effort service is similar to the UBR service category, but a service using the available rate with some minimum guarantees, as the ABR service category provides it, is still missing in the Differentiated Services architecture. Thus, the introduction of an ABR-like service exploiting unused rates is proposed.

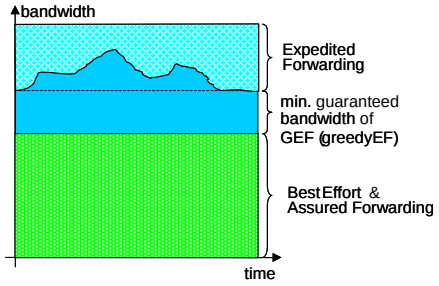

Figure 2 Rate Usage of the proposed GEF Forwarding Behavior.

4.1 Exploiting Reserved Rate not used by Expedited Forwarding Flows

Due to its simple realization, the EF PHB is widely accepted within the Differentiated Services community as one basic building block for provision of guaranteed services in Differentiated Services networks. In order to realize the EF behavior, a traffic shaper using a leaky bucket is needed at the ingress to the Differentiated Services network in order to ensure that the considered flow is in conformance with the service level agreement (similar to the traffic contract in ATM networks). Routers inside the network (so-called Interior Routers) treat all packets belonging to the EF behavior as one aggregated flow. Admission control ensures that EF packets can be forwarded always fast enough, so that the EF queue is almost empty at anytime. Therefore, the rate guarantee and minimal delay property can be realized by a simple priority scheduler, which serves the best effort queue only if no EF packets are to be sent. As a result, the remaining rate will be used by Best Effort packets if the aggregated EF flows use less than the reserved rate. Nevertheless, no service guarantees can be given for the Best Effort traffic. In order to increase the exploitation of the reserved rate and to provide at least some statistical guarantees, we propose a new forwarding behavior called *GEF (Greedy EF)*, which exploits the rate remaining from the aggregated EF flows for sending its own traffic more immediately (cf. Fig. 2).

GEF is short for Greedy Expedited Forwarding and is a new behavior realized in the Differentiated Services context. GEF keeps the minimal delay characteristics of EF, and additionally allows the end system to send more data at once, but it does not touch the remaining rate of Best Effort or of other low priority traffic. Therefore, GEF is very useful for exchanging short bulks of data, and it offers a better service for transactions or client/server communications. Such applications need support for a low delay, quick and reliable data transfer. EF is not appropriate for supporting such services, because it does not allow exchange short bulks of bursty data. The Assured Forwarding (AF) PHB as defined in RFC 2597 supports bursts, but assigns a higher drop precedence to excess packets which is unfavorable for transaction oriented applications.

GEF can be described by the following properties:

- **Own Resources:** GEF uses own resources, i. e., packet queues.
- **Guaranteed Mean Bitrate:** GEF offers a guaranteed mean rate controlled by a token bucket algorithm. Non-conforming packets are discarded. Moreover, a maximum short burst size will be accepted depending on the specified bucket size.
- **Bursts:** Client/Server applications often have a bursty nature of traffic. Therefore, GEF should allow passing bursty traffic as well. The agreed burst size should be similar to the transmitted size of client/server requests and should not be too big. Accepting bursty traffic at a guaranteed rate results in a guaranteed mean bit rate.
- **Greedy Behavior:** GEF assumes that there may be unused EF bandwidth, which is otherwise used by the BE traffic class. This unused bandwidth is now exclusively utilized by the GEF per-hop behavior to realize a faster forwarding for GEF packets. These packets will experience a shorter end-to-end delay than AF or BE packets.

Scheduling for GEF

The scheduling mechanisms for GEF have to consider some constraints concerning the impact on other per-hop behaviors such as EF or AF.

1. EF should always have the highest priority, because it is very sensitive to jitter.
2. Lower priority per-hop behaviors, such as AF or BE, should not suffer from the greedy behavior of GEF. There should be mechanisms which limit the effects on these per-hop behaviors.
3. GEF should utilize unused EF bandwidth before other per-hop behaviors do, e. g., before BE.

Priority Queuing (PQ) and Weighted Fair Queuing (WFQ) alone are not appropriate for GEF. Table 1 lists some advantages and disadvantages. This overview also introduces a new scheduling mechanism called *Weighted Fair Queuing with Priority for EF* which combines the advantages of PQ and WFQ.

EF and GEF are considered to be one single class which is sharing the overall bandwidth with the other classes. The EF queue is served with the highest priority. Every EF packet sent decreases the current weight of the EF/GEF class, so there is a fair share of bandwidth between the classes EF/GEF, AF and BE. Furthermore, GEF can exploit reserved but unused EF bandwidth.

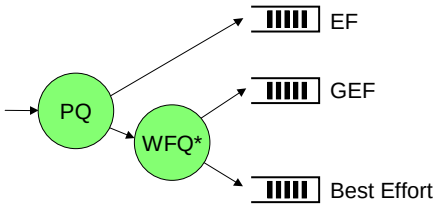

Figure 3 Realization of GEF in the router.

	Advantage (+)/Disadvantage (−)
Priority Queuing	+ EF can be forwarded at highest priority. + GEF packets are forwarded quickly. − AF and BE may suffer from a higher delay during transmitting short bulks of GEF data.
Weighted Fair Queuing	+ Fair share of bandwidth. − The EF queue cannot always be served at highest priority. Thus, EF packets suffer from a higher end-to-end delay and jitter.
Weighted Fair Queuing with Priority for EF	+ Fair share between AF, BE and EF/GEF. − GEF packets have to wait longer if there is no time quantum left to send packets.

Table 1 Comparison of scheduling mechanisms for GEF

This new scheduler can be implemented as shown in Fig. 3. A second queue is inserted for buffering GEF packets. A priority queuing element (PQ) takes care, that EF packets will always be sent before GEF packets. Additionally, a modified weighted fair queuing element (WFQ*) is positioned behind the priority queuing element. The WFQ scheduler splits up the total bandwidth between EF/GEF and other forwarding behaviors, including Best Effort. EF packets have the highest assigned priority due to the PQ element, and only if EF packets do not exploit their reserved bandwidth, GEF uses more than its minimum guaranteed rate. Therefore the GEF bandwidth in the WFQ* scheduler is configured with the sum of the EF aggregate and the minimal GEF bandwidth. If a GEF packet is sent, the consumed bandwidth will be considered in the WFQ*-scheduler. The difference between WFQ and WFQ* is now, that also EF packets will be considered in the WFQ*-scheduler. The resulting behavior is, that GEF can use up to the whole EF bandwidth, if that is not consumed by EF. But when an EF packet arrives this will limit the bandwidth of GEF up to the minimal guaranteed bandwidth of GEF, if the full EF aggregate is exploited.

4.2 Simulation Results

GEF was evaluated by using a modular QoS simulation suite called simulatedKIDS [10]. It is a construction set of basic QoS simulation models which can be combined to build arbitrary QoS behavior, like the GEF PHB. SimulatedKIDS is based on the discrete event simulation system OMNeT++ developed at the Technical University of Budapest [9]. This section presents the simulation results for GEF.

Fig. 4 shows the network that was simulated in order to evaluate GEF. It contains 3 data sources and 1 data sink. Every packet created by senders S_1, S_2 and S_3 is forwarded in direction to R_1.

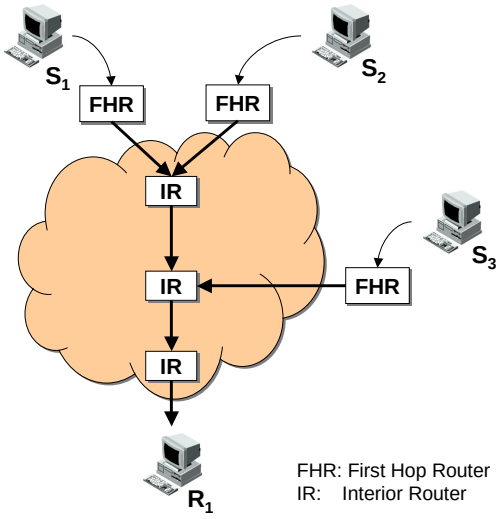

Figure 4 Simulated network used to evaluate GEF.

The further configuration was as follows:

- Simple traffic generators used by every data source create bursty traffic at a mean rate of 10 Mbps. The maximum burst size is 40,960 byte. The packet size is constant at 1,250 byte. Every generated packet is duplicated and forwarded as EF and as GEF packet.
- The bucket size of the token bucket algorithm used by both EF and GEF is configured at 40,960 byte. One token corresponds to one bit. The token generation rate is set to 10 Mbps. Since the token generation rate is equal to the guaranteed rate, no packets will be discarded.
- All router queues can hold a complete burst. No packets will be lost because of overfull queues.
- Only EF packets are passed through a traffic shaper.
- All routers use *Weighted Fair Queuing with Priority for EF* as scheduling algorithm.
- BE traffic is generated as noise traffic at the maximum link rate to use up all available residual bandwidth. Lost BE packets can be ignored.
- The link rate between first-hop routers (FHR) and interior routers (IR) is at 100 Mbps. The connections between the IR are working at the speed of 150 Mbps.

4.2.1 End-to-End Delay

Figures 5, 6 and 7 show the end-to-end delay for EF (blue curve) and GEF (red curve) packets. Each packet contains the sending time S_i as a parameter. S_i is set by the sending station. The arrival time A_i for packet i is taken at the

receiver R_1. The end-to-end delay d_i for each packet i can be calculated by the receiver as follows: $d_i = A_i - S_i$.

The effects of shaping EF packets can be seen in all figures. The traffic shapers release EF packets at the configured rate. This results in continuous data flow without bursts after the first router. But due to the artifical delay inserted by the traffic shaper, an increasing end-to-end delay for adjacent EF packets during a burst can be observed. Remember that the traffic generators are sending a burst of packets (40,960 bytes) at link rate and then pause for a while to achieve the guaranteed mean rate. This results in the depicted sawtooth curves. The end-to-end delay course of EF packets is identical for all simulation runs shown.

The reserved link bandwidth for EF and GEF is 30 Mbps, whereby token buckets control the guaranteed rate of 10 Mbps for each class. The remaining 10 Mbps allow GEF to be greedy. This situation is shown in Fig. 5. The GEF curve shows a lower packet delay than EF. GEF can make use of additional bandwidth that is not used by EF. GEF serves packets from the queue faster than EF resulting in a better delay performance.

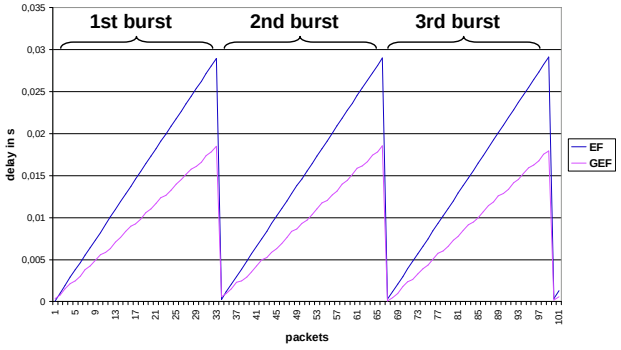

Figure 5 End-to-end delay comparison for EF and GEF.

Additionally, the delay for GEF packets is even much shorter if there is no BE traffic at all as shown in Fig. 6. GEF can be more greedy. In contrast, EF packets show the same delay as mentioned above due to traffic shaping.

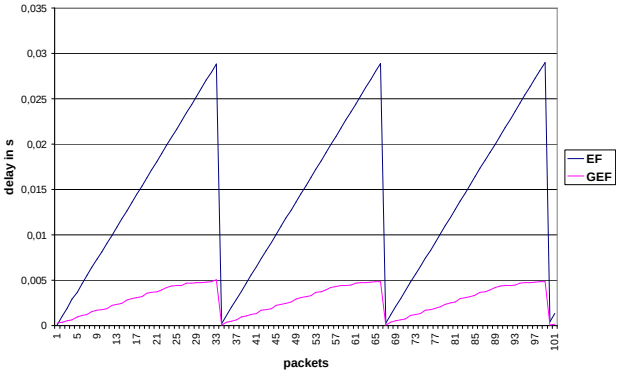

Figure 6 End-to-end delay without BE traffic on the link.

Fig. 7 shows a worst-case situation for GEF if there is no unused bandwidth at all. GEF cannot exploit any additional bandwidth. In this case, the end-to-end delay is similar to EF. In this simulation run the reserved bandwidth for EF and GEF was configured at 20 Mbps.

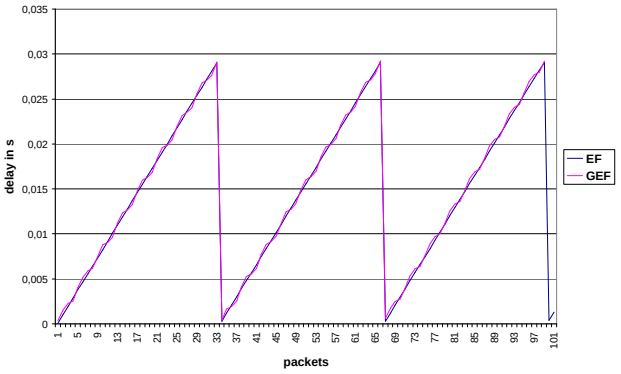

Figure 7 End-to-end delay. GEF cannot be greedy.

4.2.2 Data rate

Measuring the data rate is appropriate to show the greedy property of GEF. The data rate was measured in two ways:

1. An average data rate was calculated over time intervals of 10 ms for all arriving packets.

$$\text{Avg Rate} = \frac{\sum \text{Packet Lengths}}{10\,\text{ms}}$$

2. The current data rate was measured for each packet by the following formula:

$$\text{Current Rate} = \frac{\text{Length}_{\text{Packet}_i}}{\text{Arrival}_{\text{Packet}_i} - \text{Arrival}_{\text{Packet}_{i-1}}}$$

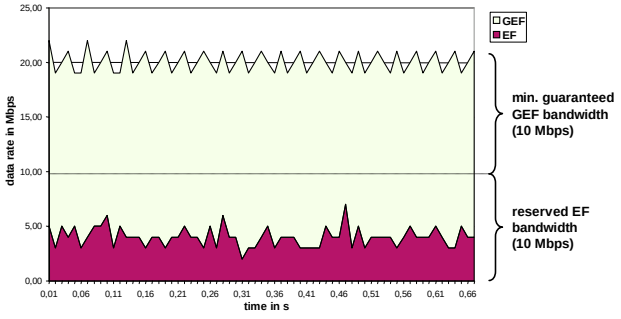

Figure 8 Average data rate measured over 10 ms time intervals.

The simulation run measuring the average data rate is shown in Fig. 8. Both EF and GEF have a reserved bandwidth of 10 Mbps policed by a token bucket. It was necessary to send a continuous GEF data flow for calculating the average data rate. Therefore, different traffic generators were used for EF and GEF. GEF packets are generated at maximum link rate to keep the queue full. In addition, the EF traffic generator is generating packets at a variable data rate between 25% and 100% of the guaranteed bandwidth. This allows GEF to be greedy because EF does not use all reserved bandwidth.

Fig. 8 shows that GEF exploits bandwidth that is not used by EF. Furthermore, EF and GEF together keep the reserved bandwidth of 20 Mbps, so the other per-hop behaviors do not suffer from a higher delay.

The greedy property of GEF can also be seen in Fig. 9 illustrating the current date rate of each packet. The minimum arrival rate of GEF packets is close to the guaranteed

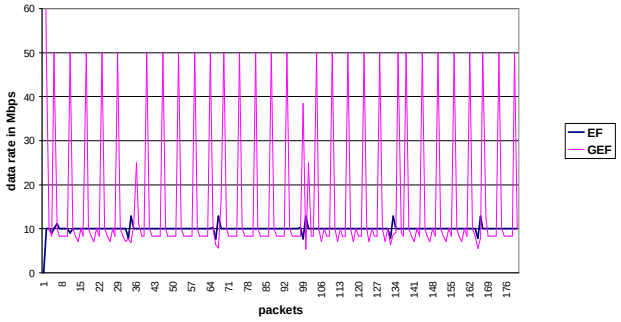

Figure 9 Current data rate measured by each packet.

rate of 10 Mbps. Moreover, the arrival rate can be higher because GEF packets can be sent earlier utilizing unused EF bandwidth. This results in maximum current data rate of 50 Mbps. Due to the traffic shaper, EF packets arrive exactly at the configured rate of 10 Mbps.

4.3 Simulation Summary

The simulation runs described above show that GEF can achieve a forwarding behavior resulting in a lower end-to-end delay compared to EF. Due to the compulsory traffic shaping, EF packets experience a higher delay. GEF is appropriate for applications generating bursty traffic patterns of a limited burst size. Examples for such applications are transaction oriented services or typical client/server applications.

It is important to implement a traffic usage control (e. g. by using token buckets) in order to keep a defined mean data rate. This is necessary to limit the damage to other per-hop behaviors, like AF or BE. Additionally, admission control has to be carried out for GEF to guarantee a mean rate by preventing an overload of resources. Furthermore, an appropriate scheduling mechanism has to be used. A new scheduler called *Weighted Fair Queuing with Priority for EF* has been introduced. This scheduler assigns the highest priority to EF packets, but keeps track of fair bandwidth allocation for all per-hop behaviors, too.

5 Conclusion

In both ATM and IP networks, along with the mechanisms providing hard rate guarantees there is a need for mechanisms using available rate. The main motivation for these mechanisms is to increase the overall exploitation of the link rate. By introducing the ABR service category, such a mechanism exists at least in ATM networks. The deployment of ABR is promising in new access technologies like xDSL. Therefore, we presented a way to integrate the resource management of ABR in an end system, and discussed problems of using ABR in an xDSL environment. A service comparable to ABR currently does not exist in IP networks. We presented a new way to adopt the idea of exploiting reserved, but unused rate by implementing a new forwarding behavior in the Differentiated Services architecture, called GEF. GEF guarantees a minimal bandwidth and it exploits unused EF bandwidth to provide a low-delay transmission for short bulks of data.

References

[1] ITM-Research Pages about ATM-PC-Lab. http://www.telematik.informatik.uni-karlsruhe.de/forschung/atm-pcs/.

[2] W. Almesberger. ATM on Linux. http://icawww1.epfl.ch/linux-atm/.

[3] F. Baker, J. Heinanen, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. RFC 2475, December 1998.

[4] R. Bless and K. Wehrle. Evaluation of Differentiated Services using an Implementation und Linux. Proceedings of the International Workshop on Quality of Service (IWQOS'99), London, June 1999.

[5] The ATM Forum. Traffic Management Specification Version 4.0 af-tm-0056.000, April 1996.

[6] The ATM Forum. Handling Physical Rate Changes in ADSL and Other Technologies. ATM Forum/98-0825, December 1998.

[7] V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB. RFC 2598, June 1999.

[8] H. Ritter and K. Wehrle. Traffic Shaping in ATM and IP networks using standard end systems. Conference on High Performance Switching & Routing, Joint IEEE/ATM Workshop 2000 and 3rd International Conference on ATM, Heidelberg, June 2000.

[9] A. Varga. Omnet++ discrete event simulation system 2.0. http://www.hit.bme.hu/phd/vargaa/omnetpp.htm, February 2000.

[10] K. Wehrle, V. Kahmann, and J.Reber. A simulation suite for the Internet Protocol with the ability to integrate arbitrary QoS behavior. Proceedings of the Computer Networks and Distributed Systems Modelling and Simulation Conference (CNDS'01), Phoenix, January 2001.