

Efficient Translation of Network Performance Parameters for Transport of IP Packets over Cell-Switched Subnetworks

Jens Schmitt¹, Martin Karsten¹, and Ralf Steinmetz^{1,2}

¹ Industrial Process and System Communications, Darmstadt University of Technology, Germany

² German National Research Center for Information Technology, GMD IPSI, Darmstadt, Germany

Email: {Jens.Schmitt,Martin.Karsten,Ralf.Steinmetz}@KOM.tu-darmstadt.de

<http://www.kom.e-technik.tu-darmstadt.de/>

<http://www.ipsi.gmd.de/>

Abstract – In this paper we deal with mapping performance-oriented IP services such as *Integrated Services (IntServ)* or *Differentiated Services (DiffServ)* onto network performance parameters for cell-switched transport networks, as for example ATM. The impact of translating IP performance parameters into ATM network services is analyzed and the detrimental effects of careless mappings are illustrated. Then, approaches to circumvent these detrimental effects are presented.

Keywords: Network QoS, IntServ, DiffServ, ATM, AAL.

I. INTRODUCTION

A. Motivation

Both the Internet's standardization organization, IETF, and the telecommunication standardization committees have developed Quality of Service (QoS) architectures. While the telecommunication people took a rather revolutionary step with the Asynchronous Transfer Mode (ATM) as the candidate for the next-generation integrated services network, the Internet community tries to follow an evolutionary path by integrating QoS-enabling components into the existing IP technology. Stemming from two very different, though today somewhat converging research and standardization communities this led to very different QoS architectures.

The "grand plan" of the telecommunication community with a global ATM network at the heart of a homogeneous integrated services network nowadays seems to fade away. Yet, while not being used as end-to-end solution the growth of ATM networks in the backbone of large-scale internetworks, as, e.g. the Internet, is a reality.

In general, it is agreed that heterogeneity is a fact for today's large-scale internetworks, with the global Internet as the most prominent example. Therefore it is also a fact for QoS architectures. Anyway, competition, different strengths and evolution are arguments for heterogeneity with regard to QoS as well.

QoS architectures can be viewed as a combination of QoS procedures, as, e.g. signalling protocols, and QoS models, which capture the declarative part of the architecture, as, e.g. the available service classes. We will concentrate on the mapping of QoS models between IP and ATM networks, and here in particular on the translation of the network performance parameters. With regard to the procedural aspects of mappings, see for example [1], or [2], yet there are many more.

B. Outline

First we give a brief overview of the different QoS models of IP and ATM, then some general thoughts on the mappings between the IP and ATM QoS models are presented, before we go into the details of one generic problem of any mapping: the translation of the performance parameters from packet or byte units into cell units. We identify two major problems of a straightforward translation and then present approaches to solve or at least alleviate each of them. At the end, we take a look at related work and draw some conclusions from our investigations.

II. QUALITY OF SERVICE MODELS

A. ATM (Forum) Model

The ATM service model is based on the traditional call paradigm with the following service classes as semantic interpretation framework for the network performance parameters of an ATM network [3]:

- Constant Bit Rate(CBR): offers a constant bit rate service suited for real-time applications with stringent requirements on delay and bandwidth.
- real-time Variable Bit Rate(rt-VBR): offers a similar service to CBR but allows for some controlled burstiness of the data stream.
- non-real-time Variable Bit Rate (nrt-VBR): a non-real-time service with a deterministic bound on loss as long as traffic adheres to its specified shape.
- Unspecified Bit Rate(UBR): plain best-effort service without any guarantees.
- Available Bit Rate (ABR): feedback-based service that allows for a minimum rate to be specified and ensures fair sharing within this class of traffic.
- Guaranteed Frame Rate(GFR): a frame-aware service that allows for a minimum rate to be specified, and, e.g. takes AAL5 frame boundaries into account when making cell discard or tagging decisions.

Most of these require a traffic specification which is based on the Generic Cell Rate Algorithm (GCRA). The unit of the parameters are cells resp. cells/s, even for the GFR service. For the exact definition of the parameters and their applicability to the service categories, see [3].

B. IETF Models

Much work inside the IETF has been devoted to the development of QoS models for the Internet. The outcome are two different models for achieving QoS: IntServ and DiffServ. These however deal with different needs and can also be seen as complementary and mutually assisting [4], and not necessarily competing.

1) IntServ

This model is more in the tradition of telecommunications business models, where an end-to-end service is offered to the customers at the end-systems. Therefore, the services offered are specified at the flow-level, i.e. very fine-grained. Two services have advanced to proposed standards: Guaranteed Service (GS) [5] and Controlled Load Service (CLS) [6].

GS offers deterministic guarantees on the maximum end-to-end delay and the available bandwidth as well as a zero loss assurance. It requires a traffic specification, called TSpec, which is essentially a double token bucket, with r as the token rate of the first bucket and b as its bucket depth, and for the second bucket the peak rate p and the maximum packet size M as the bucket depth. The service rate R as specified by the receiver(s) determines the experienced queuing delay and thus serves as control parameter to adjust the maximum delay tolerable for a GS user.

CLS has a much looser specification which is supposed to offer a service that is comparable to best-effort service in a "lightly loaded" network. It also requires the specification of a TSpec and ensures that under any load condition of the network a CLS user will at least have a throughput of the token rate r .

For both services the units in which parameters are specified are bytes resp. bytes/s.

2) DiffServ

This model [7] is a more pragmatic/less ambitious approach motivated by the reality of today's Internet service providers (ISP), which would like to offer higher value services to their customers, who are end-users as well as other ISPs. Hence, the services offered will be based on traffic aggregates and will thus be rather coarse-grained.

The approach taken for DiffServ is not to specify the services - these shall be part of bilateral Service Level Agreements (SLA) between providers or customers - but to specify the behaviour of the forwarding elements in so called Per-Hop Behaviours (PHB). Two PHBs have been advanced to proposed standards:

- Expedited Forwarding (EF) [8]
- Assured Forwarding (AF) [9]

Both of them require the configuration of a certain service rate to satisfy their specified behaviour. This rate will be given in bits/s or bytes/s, which are of course equivalent (for our purposes).

C. Selecting the ATM Service Categories

An important decision for the mapping of the QoS models is the assignment of IP-related services to ATM service categories. Although this is not the focus of our paper, we briefly want to discuss the issues for that fundamental selection here, for both IntServ and DiffServ.

1) IntServ

A mapping of the IntServ classes, GS and CLS, onto ATM service categories should try to preserve their respective semantics, while at the same time trying to minimize the resource usage inside the ATM subnetwork.

For GS, the most straightforward candidate is rt-VBR, although CBR is another, but presumably more costly alternative. The other service categories do not seem to be suited since they are for non-realtime applications. For CLS, either nrt-VBR, ABR, or GFR are good candidates, whereas CBR or rt-VBR are principally possible, but too costly as they offer more than is needed to satisfy the CLS specification.

2) DiffServ

The first question to be answered for DiffServ is whether ATM VCs correspond to SLAs or PHBs. While the ATM Forum takes the former position, the IETF favours the latter. For EF-based SLAs or for the EF PHB the hottest candidates are CBR and rt-VBR, whereas for AF, ABR or GFR seem the most reasonable choices.

The above considerations are led by technical and economical rationale and similar, yet much more detailed treatment of this can be found in ([10] for IntServ, and [11] for DiffServ). It is our future goal and our strong belief that it is necessary to verify and possibly modify these by trials and measurements for real traffic.

However, in this paper we want to concentrate on a common issue of any mapping. That is the conversion of the parameters from bytes to cells, which results from the fundamentally different characteristic of variable vs. fixed transport unit sizes. No matter whether it is IntServ, DiffServ or any other IP-performance-oriented service, they all have to deal with this issue of translating the packet-based nature of IP-performance related metrics into ATM's cell-based counterparts - supposing an ATM subnetwork is crossed.

III. TRANSLATING THE PERFORMANCE PARAMETERS

A. Straightforward Translations

Consider a flow of packets, for which an IP network service performance commitment exists, with each packet in isolation and assume that no more than one packet fits into a single cell (often more cells are required). Note here that an IP header already consumes 20 bytes, and a UDP header another 8 bytes so that for example an application using UDP/IP never produces packets of which more than one would fit into a single ATM cell, especially if possible further AAL-related encapsulation overhead is taken into account.

Let us look at that in a more formal and general way. First we define some terms:

Cell Overhead: o_c [in bytes].

Packet Overhead: o_p [in bytes].

Packet size: s_p [in bytes] with $s_p \in [m, M]$, i.e. m is the minimum and M the maximum packet size for the flow.

Cell size: s_c [in bytes].

Number of cells per packet: n_c [in cells/packet], where

$$n_c = \left\lceil \frac{s_p + o_p}{s_c - o_c} \right\rceil.$$

Since s_p may vary (while the other parameters are fixed, at least per flow), the number of cells per packet may also be regarded as a function of the packet size: $n_c(s_p)$. Given that $s_p \in [m, M]$, the following bounds on the number of cells per packet are given:

$$n_c^{\min} = n_c(m) = \left\lceil \frac{m + o_p}{s_c - o_c} \right\rceil \leq n_c \leq \left\lceil \frac{M + o_p}{s_c - o_c} \right\rceil = n_c(M) = n_c^{\max}$$

Given a certain IP performance-related rate r [in bytes/s], we get a packet rate r_p with

$$r_p = \frac{r}{s_p} \text{ [in packets/s],}$$

which again allows to compute the required cell rate r_c with

$$r_c = r_p n_c \text{ [in cells/s].}$$

Again the only variable parameter is s_p and we therefore realize that the cell rate is a function of the packet size(s), $r_c(s_p)$, as well as the packet rate, $r_p(s_p)$. Both n_c and r_p vary with s_p . While r_p weakly decreases with s_p , n_c weakly increases with s_p .

Noticing that r_c is a weakly decreasing function in s_p , i.e., r_c shows some spontaneous short-scale increases due to well-fitting packet sizes, but shows long-scale decreases due to the sharing of packet overhead, we obtain the following bounds on r_c :

$$\begin{aligned} r_c^{\min} &= \frac{r}{s_p^{\min}} \left\lceil \frac{s_p^{\min} + o_p}{s_c - o_c} \right\rceil = r_p(s_p^{\min}) n_c(s_p^{\min}) \leq r_c \\ &\leq r_p(s_p^{\max}) n_c(s_p^{\max}) = \frac{r}{s_p^{\max}} \left\lceil \frac{s_p^{\max} + o_p}{s_c - o_c} \right\rceil = r_c^{\max} \end{aligned} \quad (1)$$

where

$$\begin{aligned} s_p^{\max} &= \operatorname{argmax} r_c(s_p) | s_p \in [m, M] \\ &= \operatorname{argmax} r_c(s_p) | s_p \in \{m, \left\lceil \frac{m + o_p}{s_c - o_c} \right\rceil (s_c - o_c) - o_p + 1\} \end{aligned}$$

$$\begin{aligned} s_p^{\min} &= \operatorname{argmin} r_c(s_p) | s_p \in [m, M] \\ &= \operatorname{argmin} r_c(s_p) | s_p \in \{M, \left\lceil \frac{M + o_p}{s_c - o_c} \right\rceil (s_c - o_c) - o_p\} \end{aligned}$$

Of course for ATM $s_c = 48$, and for different AALs the

resulting numbers and formulae are given in Table 1, where it is assumed that LLC/SNAP encapsulation as defined in [12] is used in all cases. If instead of that VC-based multiplexing was used then all o_p values would need to be diminished by 8.

TABLE 1 Application of the Mathematical Framework.

AAL Type	o_c	o_p	n_c	r_c
AAL 1	1	8	$\left\lceil \frac{s_p + 8}{47} \right\rceil$	$\left\lceil \frac{r}{s_p} \left\lceil \frac{s_p + 8}{47} \right\rceil \right\rceil$
AAL 2	4	8	$\left\lceil \frac{s_p + 8}{44} \right\rceil$	$\left\lceil \frac{r}{s_p} \left\lceil \frac{s_p + 8}{44} \right\rceil \right\rceil$
AAL 3/4	4	16	$\left\lceil \frac{s_p + 16}{44} \right\rceil$	$\left\lceil \frac{r}{s_p} \left\lceil \frac{s_p + 16}{44} \right\rceil \right\rceil$
AAL 5	0	16	$\left\lceil \frac{s_p + 16}{48} \right\rceil$	$\left\lceil \frac{r}{s_p} \left\lceil \frac{s_p + 16}{48} \right\rceil \right\rceil$

This table is a slightly arguable since for AAL1 and AAL2 there are no standards or proposals as to how to encapsulate IP packets.

To assess how much the choice of the packet size affects the cell rate that is to be allocated, take a look at the cell rates for different packet sizes as depicted in Figure 1. Here we assumed the use of AAL5 and LLC/SNAP encapsulation and an IP performance-related rate r of 10000 bytes/s.

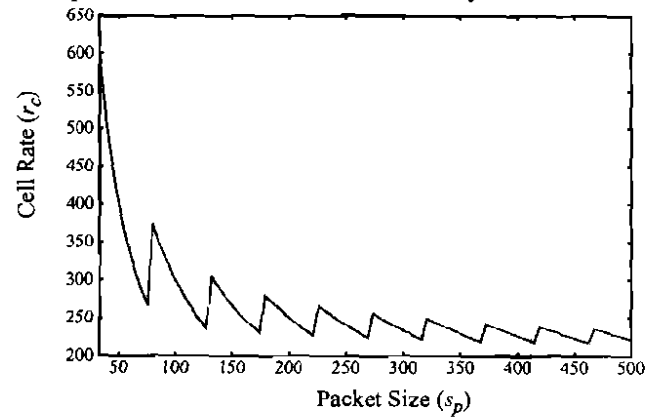


Figure 1: Cell Rates for different packet sizes.

Depending on the packet size we have to allocate cell rates differing by a factor of almost three. Furthermore, we notice that even for packet sizes closely together the difference in their corresponding cell rates may be huge. Let us look at that more rigorously.

B. Performance Analysis

In this section we first define and motivate some metrics, which then serve as criteria for discussing different schemes for translation of the packet-based performance parameters into their cell-based counterparts.

1) Metrics

Let us first define a metric called Cell Utilization Efficiency (*CUE*) as follows:

$$CUE = \frac{r}{r_c s_c} \in \left[\frac{r}{r_c^{max} s_c}, \frac{r}{r_c^{min} s_c} \right] \subset [0, 1] \quad (2)$$

The *CUE* is a measure of how well utilized allocated resources of the cell-switched network are if the expected packet size matches the actual packet size.

It may however be the case that the expected packet size when the allocation is made is not the packet size actually seen in the data flow. Therefore let us define a further metric to measure the cell utilization efficiency for this case. Assume \bar{r}_c is chosen as cell rate based on an expected packet size \bar{s}_p , yet s_p turns out to be the actual packet size. Then let us define the realized *CUE* (*rCUE*) as function of s_p :

$$rCUE(s_p) = \begin{cases} \frac{r}{r_c s_c} & s_p < \bar{s}_p \\ \frac{r}{r_c s_c} - \frac{\bar{r}_c - r_c}{\bar{r}_c} & s_p \geq \bar{s}_p \end{cases} \quad (3)$$

Certainly, the worst case with regard to efficiency is that the actual packet size is the packet size that minimizes the cell rate, i.e. s_p^{min} . We capture this case in a metric called worst-case *CUE* (*wCUE*), which is defined as:

$$wCUE = rCUE(M) = \frac{r}{r_c^{min} s_c} - \frac{\bar{r}_c - r_c^{min}}{\bar{r}_c} \quad (4)$$

In any case that means that it is favourable to base the cell rate on as large as possible packet sizes. But cell utilization is just one side of the "story", the other is how badly we may overload the cell rate allocation by overly "optimistic" packet size "expectations". That is captured in the following metrics.

The Cell Loss Rate (*CLR*) is defined as a function of s_p :

$$CLR(s_p) = \begin{cases} 1 - \frac{\bar{r}_c}{r_c} & s_p < \bar{s}_p \\ 0 & s_p \geq \bar{s}_p \end{cases} \quad (5)$$

Of course the highest rate of cell losses is incurred if the actual packet size maximizes the cell rate, i.e. it is s_p^{max} . Thus we define the worst-case Cell Loss Rate (*wCLR*) as:

$$wCLR = CLR(m) = 1 - \frac{\bar{r}_c}{r_c^{max}} \quad (6)$$

The *wCLR* measures how badly overloaded the cell-switched network may be due to an undersized cell rate allocation as the result of overestimating packet sizes.

2) Discussion

Let us now take a look at how the straightforward translation of IP performance parameters onto cell-switched network parameters behaves with regard to the introduced metrics.

In Figure 2 the *wCUE* is depicted, again for the case where AAL5 with LLC/SNAP encapsulation is used and the IP-related rate r is 10000 bytes/s.

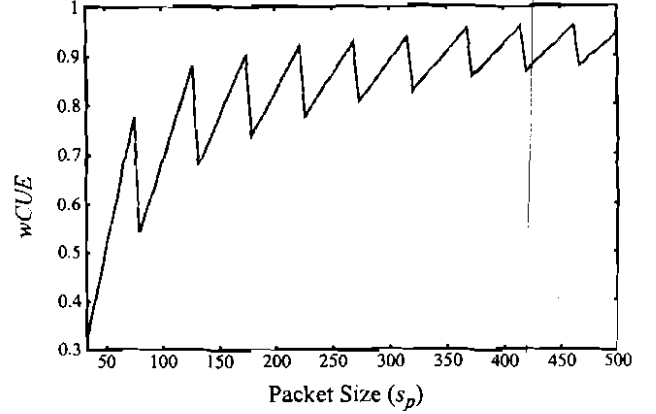


Figure 2: Worst-Case Cell Utilization Efficiency.

There are two basic and orthogonal problems that lead to inefficient use of cell-rate resources which are illustrated in the above graph:

1. Over-reservation due to the uncertainty about packet sizes, and therefore about the number of packets per unit of time, since this influences the overhead sharing of framing packets for transport over the cell-switched network. The weakening of this effect as the maximum packet size is approached is represented by the long-term increase of the *wCUE* curve.
2. Over-reservation due to unused capacity in partially filled cells resulting from "inconvenient" packet sizes. This effect is represented by the spontaneous short-term decreases of the *wCUE* curve, whenever a cell boundary is exceeded by the packet size on which the cell rate allocation is based.

Obviously, for efficiency reasons it would be advantageous to assume large packet sizes and to carefully choose the packet size (on one of the peaks if possible).

Yet, in Figure 3 the *wCLR* is depicted for different packet sizes.

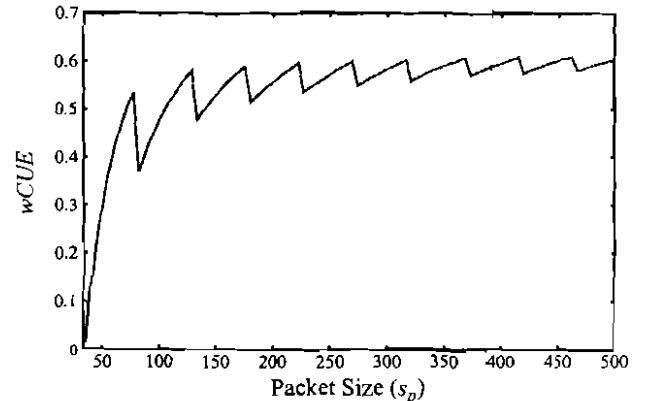


Figure 3: Worst-Case Cell Loss Rate.

Of course, the *wCLR*s rise as the packet size increases, on which the cell rate allocations are based. Furthermore, the

packet sizes that were convenient with respect to $wCUE$ are very bad for the $wCLR$ as they correspond to spontaneous peaks of it.

Obviously, the $wCUE$ and $wCLR$ are competing metrics because when trying to improve the cell utilization efficiency by lowering the cell rate, the risk is to incur a higher cell loss rate. Therefore a compromise for the assumed packet size of the IP data stream must be found according to its service semantics. A strict service as, e.g. IntServ's GS will not tolerate any cell loss, so that s_p^{max} must be assumed as packet size for the calculation of the cell rate corresponding to the service rate R . For services that do not provide for such strict guarantees a tradeoff between the risk of incurring cell loss and an improved efficiency is possible.

All of the above assumes that the packet size is not a controlled variable. Of course, one may argue that applications could generate IP packets of well-suited size that fit exactly into an integral number of cells and are as large as possible. Yet, in general this seems to be not feasible or at least not convenient due to the following problems:

- applications should not need to know about a (possibly "far away") cell-switched subnetwork,
- ATM is just one link, other links might have different needs with regard to packet size,
- applications would need link layer knowledge, which constitutes a gross layering violation.

Consequently, edge devices, mediating between IP and ATM, have to cope with uncertainty about packet sizes and with unluckily sized packet that do not fit the cell stream well. While solution approaches to the former problem will be dealt with in Section V, we will at first address the latter problem by a scheme we called cell-aligned framing.

IV. EFFICIENT TRANSLATION BASED ON CELL-ALIGNED FRAMING

A. Idea

The straightforward rate translation scheme presented and analysed in the previous section regarded each packet of an IP data stream in isolation and encapsulated it into a separate AAL frame. That leads to the problem of partially filled cells that have to be padded with bytes containing no information. The idea of cell-aligned framing is to fill AAL frames such that they fit exactly into the cell stream irrespective of the packet boundaries. Therefore a single AAL frame may contain two (partial) packets. However only the last cell of a frame should contain data from both packets: the end of the first packet and the beginning of the next packet. This scheme is illustrated in Figure 4.

This scheme requires that there is a way to mark the start of a new packet inside an AAL frame, which may result in some additional protocol overhead, which however as we will see in Section IV.D should not be inhibitive. Furthermore, note here that it is not necessarily required to circumvent padded cells but to use cell-alignment only in case it is necessary, i.e.,

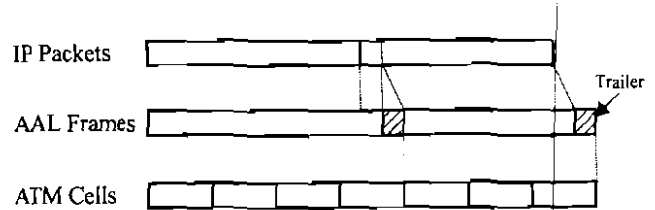


Figure 4: Cell-Aligned Framing.

if the worst case of a stream sending bursts at s_p^{max} sized packets is actually occurring, because on this case the rate calculations have to be based (at least for hard guarantees as, e.g. for IntServ's GS).

At this stage, one may argue that minimum packet sizes may be large enough to make the overhead incurred by partially filled cells negligible, yet that is not the case for many real-time applications where packetization delays still play a certain role, and furthermore not for IP traffic aggregates as they have to be dealt with when using DiffServ. Here packet sizes may vary highly (also to the lower end) and may be not known beforehand so that small packet sizes must be assumed to be on the safe side. To give a feeling for current IP traffic's packet size distribution, see Figure 5, which was produced by [CMT98] from a 24 hour traffic trace at an OC3 link of the MCI network backbone.

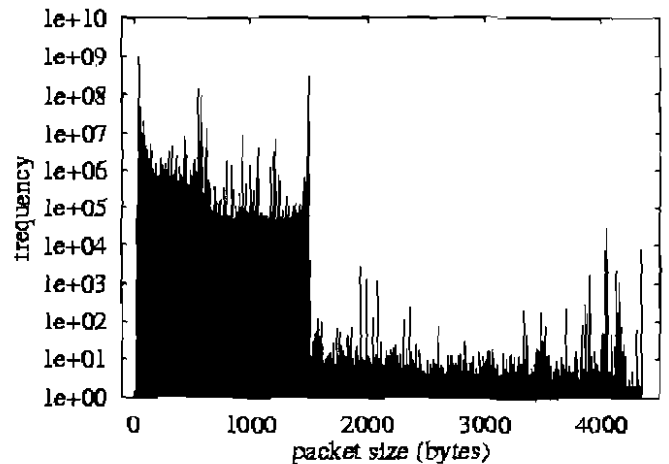


Figure 5: Typical Packet Size Distribution.

This clearly shows that small packet sizes are still predominant at least for today's IP traffic at the aggregate level. One should however be aware that new services as introduced by IntServ and DiffServ will certainly change traffic characteristics, as, e.g. the packet size distribution.

B. Analysis and Comparison

Using the notation and definitions of Section III lets us analyse the approach of cell-aligned framing and compare it with the straightforward approach:

Overhead for cell-alignment: o_{align} .

In this case the cell rate corresponding to a byte rate r is:

$$r_c(s_p) = \left\lceil \frac{r}{s_p} \times \frac{s_p + o_p + o_{align}}{s_c - o_c} \right\rceil \quad (7)$$

where we have the following bounds on r_c

$$r_c^{min} = \left\lceil \frac{r}{M} \times \frac{M + o_p + o_{align}}{s_c - o_c} \right\rceil \leq r_c \leq \left\lceil \frac{r}{m} \times \frac{m + o_p + o_{align}}{s_c - o_c} \right\rceil = r_c^{max} \quad (8)$$

In Figure 6 the $wCUE$ for the case of a straightforward translation and the approach based on cell-aligned framing are compared.

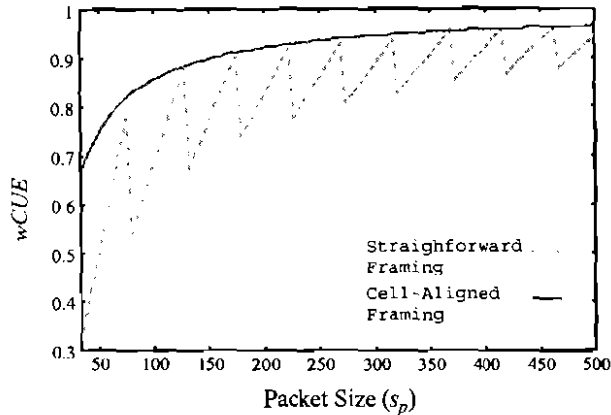


Figure 6: Worst-Case Cell Utilization Efficiency.

We used the same settings as in the examples before and assumed no overhead for the cell-alignment, which as will be shown in Section IV.D is possible for AAL5. It is obvious that cell-aligned framing can achieve quite a substantial efficiency gain, especially for very small packet sizes.

Let us now take a look at the $wCLR$ for both cases as it is depicted in Figure 7.

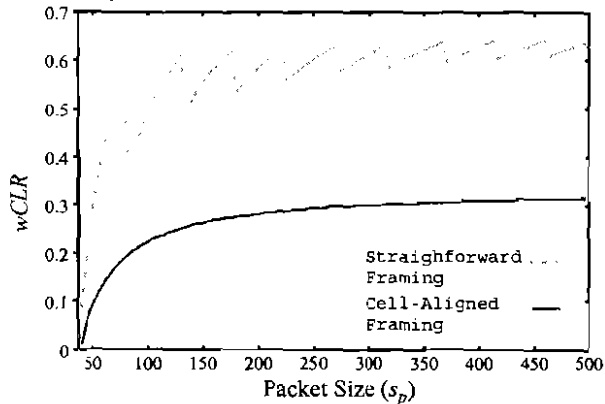


Figure 7: Worst-Case Cell Loss Rate.

Again it can be seen that cell-aligned framing is a considerable improvement over the straightforward approach where packets are treated in isolation. This is due to the fact that the space of possible cell rates, i.e. $[r_c^{min}, r_c^{max}]$, is considerably compressed and thus the risk of assuming large packet sizes for the cell rate allocation translates into much lower cell loss

rates if the actual packet size is less.

C. Potential Drawbacks

After having shown the benefits of cell-aligned framing over the straightforward rate translations, let us now look at some potential counter-arguments that may be raised against it:

- One question certainly is how expensive the regeneration of packet boundaries is. As mentioned above, a marking technique is needed, which may consume some PCI (Protocol Control Information), and we have some more computational effort in order to keep track of the fragmented packets. We will see below that this overhead can be kept reasonably small.
- When using cell-aligned framing not all the cells are equally important any more, because one lost cell may "kill" two packets, if it is the shared cell of two consecutive packets. However, it can be argued that either the packets are small and then there is not so much lost or they are large and then this should be an infrequent event.
- Frames may have to wait to be filled up. Yet, here the solution is to never wait for following packets to fill up the cell stream, but only fill it up if there are already packets waiting in the queue. The rationale here is that the rate computations are based on certain worst-case scenarios in which the approach would actually be applied, whereas if the rate is not fully used then the wastage of cell space is not such a big issue. The main point is that the rate translations which are based on this worst-case scenario can be kept low.

D. Implementation Using AAL5

After having shown the benefits and potential drawbacks of cell-aligned framing, we now present a very simple way of how the scheme could be implemented when AAL5 is used as adaptation layer for the transport of IP traffic over an ATM subnetwork. In the ATM terminology this could also be called a SSCS (Service-Specific Convergence Sublayer) of AAL5 for IP-performance oriented services such as IntServ or DiffServ. The task of that SSCS is to mark where a new packet starts within an AAL5 frame in order to be able to reassemble packets at the receiving side.

The AAL5 CPCS-PDU (Common Part Convergence Sublayer) is structured as depicted in Figure 8.

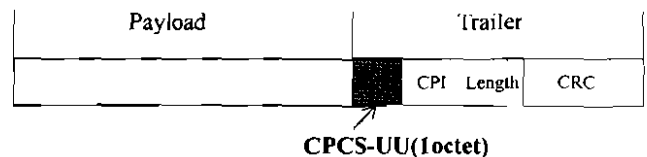


Figure 8: CPCS-PDU format for AAL5.

Fortunately, it possesses an unused field called UU (User-to-User Indication). The idea is now to use that field as a

pointer to the beginning of the next IP packet in an AAL5 frame. Thus, the semantics of the UU field is the number of bytes from the end of an AAL5 frame to the location where a new IP packet starts. This can of course be at most 255 bytes apart, yet it is sufficient if only the last cell is always with filled with the beginning of the next packet, as was proposed above. Note that UU=0 means that the encapsulated IP packet plus overhead fitted exactly an integral number of ATM cells.

In Figure 9 the required protocol processing for cell-aligned framing is illustrated in pseudocode for both, sending and receiving side. At the sending side, it has to be computed whether padding of the payload is necessary and if so, how many bytes of padding. If another packet is already waiting, then instead of padding the AAL5 frame, it is filled up with the first bytes of the waiting packet and the UU pointer is set to the beginning of that packet. At the receiving side, the packets are possibly reassembled by using the information delivered in the UU field of the AAL frame.

Using these algorithms results in no PCI overhead for cell-aligned framing, i.e. $o_{stream}=0$, but introduces a higher protocol processing cost due to the more complicated buffer management, which however from our perspective should be justified due to the considerable efficiency improvements presented above.

V. SOLUTION APPROACHES TO THE "UNKNOWN NUMBER OF PACKETS" PROBLEM

While cell-aligned framing avoids the segmentation overhead due to partially filled cells, a solution to the problem of the variability of packet sizes would save overhead that is accounted per packet, i.e. o_p . This overhead is proportional to o_p/s_p , and can of course not be totally circumvented but lowered by using some (heuristic) knowledge about the packet size distribution. This could be based upon statistics or past experience in general which might be available. The approach is mainly aimed at services that only provide for soft guarantees, as for example IntServ's CLS or DiffServ's AF.

The idea is to be able to make a quantitative statement about certain metrics given a certain packet size distribution. As an example, it should be possible to provide an assurance like: if packet sizes are uniformly distributed over $[m, M]$, then at a probability of 95% we obtain a CLR of 0.

Let us look at that in a more formal manner. Recall that s_p is a random variable which must be estimated well in order to be able to make rate allocations with favourable cell utilization and tolerable loss characteristics. Prominent example cases are:

1. s_p is uniformly distributed over $[m, M]$, i.e. its p.d.f. is

$$f(s_p) = \frac{1}{M - m + 1} \quad (9)$$

2. s_p is trapezoidally distributed over $[m, M]$ (with the slope a of the trapezoid representing the "optimism/pessimism" of the assumption on the packet sizes), i.e. its p.d.f. is:

$$f_a(s_p) = as_p - a\frac{M+m}{2} + \frac{1}{M-m} \quad (10)$$

$$a \in \left[\frac{2}{(M-m)^2}, \frac{2}{(M-m)^2} \right]$$

At first we define quantitized cell rates $r_{c,\alpha}$ as

$$p(CLR = 0 | r_{c,\alpha}) > 1 - \alpha \quad (11)$$

which means the probability to incur cell loss if we allocate $r_{c,\alpha}$ is less than α .

Let us look at the general case, where we assume that s_p has the distribution function $F(s_p)$. Yet, instead of the packet size distribution we introduce a transform of it, the packet rate distribution, where the packet rate is defined as:

$$r_p = \frac{r}{s_p} \quad (12)$$

From this the quantitized cell rates can be computed more easily (if cell-aligned framing is assumed), since the cell rate for the case of using cell-alignment can be rewritten as:

Sender-Algorithm

```

forever {
  wait for packet to be sent;
  compute #bytes required for padding;
  if (padding != 0 && another packet waiting {
    take padding bytes from waiting packet;
    fill it together with packet in AAL5 frame F;
    set F.cpcs-uu = padding;
  }
  else {
    send padded AAL5 frame;
    set F.cpcs-uu = 0;
  }
}

```

Receiver-Algorithm

```

NetworkBuffer outstanding_packet = empty;
forever {
  receive AAL5 frame F;
  append F.payload[0, F.length-F.cpcs_uu-1]
  to outstanding_packet;
  send outstanding_packet to upper layer;
  buffer F.payload[F.length-F.cpcs_uu,
  F.length] in outstanding_packet;
}

```

Figure 9: Cell-Aligned Framing Algorithm at Sender and Receiver.

$$r_c = \left\lceil \frac{r + r_p(o_p + o_{align})}{s_c - o_c} \right\rceil. \quad (13)$$

Since the packet rate has the mirrored distribution of the packet sizes (since r_p is a homomorphism of s_p), assumptions about packet sizes translate readily in the distribution of the packet rate.

To calculate those quantitized cell rates, note that

$$\begin{aligned} 1 - \alpha &< p(CLR = 0 | r_{c,\alpha}) = p(\bar{r}_c < r_{c,\alpha}) \\ &= p\left(\left\lceil \frac{r + \bar{r}_p(o_p + o_{align})}{s_c - o_c} \right\rceil < \left\lceil \frac{r + r_{p,\alpha}(o_p + o_{align})}{s_c - o_c} \right\rceil\right) \\ &< p\left(\frac{r + \bar{r}_p(o_p + o_{align})}{s_c - o_c} + 1 < \frac{r + r_{p,\alpha}(o_p + o_{align})}{s_c - o_c}\right) \\ &= p\left(\bar{r}_p < r_{p,\alpha} - \frac{s_c - o_c}{o_p + o_{align}}\right) \\ &= G\left(r_{p,\alpha} - \frac{s_c - o_c}{o_p + o_{align}}\right) \\ &= 1 - F\left(r / \left(r_{p,\alpha} - \frac{s_c - o_c}{o_p + o_{align}}\right)\right) \end{aligned} \quad (14)$$

Here $r_{p,\alpha}$ is the packet rate corresponding to $r_{c,\alpha}$ and G is the distribution of r_p . Due to the integrality constraints on cell rates it is not possible to calculate them exactly for every α , but only a (tight) lower bound can be computed, which gives a cell rate at which the $CLR = 0$ with a probability of at least $1 - \alpha$ (assuming a certain packet size distribution and therefore packet rate distribution).

To compute the cell rates from (14), note that from (13) it follows that

$$r_{p,\alpha} \leq \frac{r_{c,\alpha}(s_c - o_c) - r}{o_p + o_{align}} \quad (15)$$

Using that relation and after some algebra we obtain the relation:

$$r_{c,\alpha} > \frac{r}{s_c - o_c} \left(1 + \frac{o_p + o_{align}}{F^{-1}(\alpha)}\right) + 1 \quad (16)$$

which allows us to compute the quantitized cell rates as

$$r_{c,\alpha} = \left\lceil \frac{r}{s_c - o_c} \left(1 + \frac{o_p + o_{align}}{F^{-1}(\alpha)}\right) + 1 \right\rceil \quad (17)$$

In Table 2, are given some example values of quantitized cell rates for the sample packet size distributions (9) and (10). We used the same parameter settings as for the examples in preceding sections (in particular we used $m=33$ and $M=500$), and for the parameter a of the trapezoidal distribution we used the extreme values $\pm 2 / (M - m)^2$, which represent very optimistic respectively pessimistic assumptions on the packet

size distribution.

TABLE 2 Quantitized Cell Rates.

$r_{c,\alpha}$	$\alpha=0.01$	$\alpha=0.05$	$\alpha=0.1$	$\alpha=0.2$
uniform	298	269	252	236
optimistic trapezoidal	253	234	228	224
pessimistic trapezoidal	305	284	269	250

Alternatively and similarly, the CUE could be taken as a metric to define quantitized cell rates, or the CLR could be chosen less or equal to some $\beta > 0$. Yet, one must be aware that the latter would introduce another parameter that might be difficult to specify - parsimonious models are generally preferable.

VI. RELATED WORK

The issue of overlaying IP QoS services onto ATM subnetworks, has been and still is dealt with extensively, for an overview of that larger field of related work, see [2]. Yet, directly related to the issue of mapping the QoS models, the most important work has certainly been done in the IETF and ATM Forum. Here, for IntServ it exists a proposed IETF standard [10], that gives very detailed treatment on how to choose the ATM service categories for the GS and CLS classes. Similarly, there is work in progress [11] in the IETF and ATM Forum [13] on the mapping of PHB resp. SLAs to ATM service categories. Non-standardization work concerned with those issues can be found in [14], where it is shown that the IntServ to ATM mappings proposed in [10] are at least dubious, as they are shown to lead to excessive cell loss in simulations. The authors of [15] are especially concerned with how to map CLS to ATM and give some simulation results on their specific mapping scheme.

However, all of these do not consider the translation of the different parameter units in the detailed and rigorous manner we did in this paper. Furthermore, they restrict their investigations towards a certain IP QoS model or even only parts of it, whereas our work is generally applicable to performance-oriented IP network services, of which IntServ and DiffServ are just examples. Furthermore, most of our results are also generic for arbitrary cell-switched networks and not just for ATM. So, we see the major contribution of our work in the generality of the results on how to translate efficiently between IP and cell-switching network performance parameters.

VII. CONCLUSIONS

After thoroughly analysing previously proposed straightforward approaches we identified the two main obstacles to an efficient translation of IP to cell-switching performance

parameters as segmentation overhead and variability of packet sizes. We introduced and analysed the approach of cell-aligned framing in order to solve the issue of only partially filled cells due to the segmentation overhead. Furthermore, we presented a simple and efficient way to implement that approach for the case of AALS framing of IP packets. Based on cell-aligned framing we proposed a scheme to address the problem of variable packet sizes and therefore unknown overhead accounted per packet. The scheme is based on assumptions on packet sizes and allows for non-deterministic service guarantees to tradeoff resource allocation efficiency against cell loss probabilities.

VIII. FUTURE WORK

One obvious issue of future work is the actual implementation of the schemes presented and analysed above in order to verify experimentally the efficiency gains of cell-aligned framing and quantized cell rates. Since we have implemented a very flexible IP/ATM adaptation module for use at an edge device (described in [16]), which uses straightforward translations of the parameters, it should be easy to extend it to allow for the more sophisticated techniques proposed in this paper.

Further work items for the future could also be to find new definitions for quantized cell rates and to base them on other packet size distributions, which might be derived by observations for real traffic. Yet, note here that current IP traffic traces are only of limited value to predict statistics for IP-performance oriented traffic based on IntServ and Diffserv, since those are not commonly used in production networks at the time of writing.

IX. REFERENCES

- [1] L. Berger, E. Crawley, S. Berson, F. Baker, M. Borden, and J. Krawczyk. A Framework for Integrated Services with RSVP over ATM, August 1998. RFC 2382.
- [2] J. Schmitt and J. Antich. Issues in Overlaying RSVP and IP Multicast on ATM Networks. Technical Report TR-KOM-1998-03, University of Technology Darmstadt, August 1998.
- [3] ATM Forum Technical Committee: Traffic Management (TM) Specification 4.1, March 1999.
- [4] Y. Bernet, R. Yavatkar, P. Ford, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, and E. Felstaine. A Framework for Use of RSVP with Diff-Serv Networks, September 1999. Internet Draft, work in progress.
- [5] S. Shenker, C. Partridge, and R. Guerin. Specification of Guaranteed Quality of Service, September 1997. RFC 2212.
- [6] J. Wroclawski. Specification of the Controlled-Load Network Element Service, September 1997. RFC 2211.
- [7] D. Black, S. Blake, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services, December 1998. RFC 2474.
- [8] V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB, June 1999. RFC 2598.
- [9] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group, June 1999. RFC 2597.
- [10] M. Garrett and M. Borden. Interoperation of Controlled Load and Guaranteed Service with ATM, August 1998. RFC 2381.
- [11] S. Ayandeh, A. Krishnamurthy, and A. Malis. Mapping to ATM Classes of Service for Differentiated Services Architecture, November 1999. Internet Draft, work in progress.
- [12] J. Heinanen. Multiprotocol Encapsulation over ATM Adaptation Layer 5, July 1993. RFC 1483.
- [13] ATM Forum Technical Committee: Addendum to Traffic Management (TM) Specification 4.1 - Enhancements to Support IP Differentiated Services (Draft), December 1999. work in progress.
- [14] P. Francis-Cobley and N. Davies. Performance Implications of QoS Mapping in Heterogeneous Networks Involving ATM. In *Proc. of IEEE Conference on ATM '98 (ICATM'98)*. IEEE, June 1998.
- [15] P. Giacomazzi and L. Musumeci. Transport of IP Controlled-Load Service over ATM Networks. *IEEE Network*, 13(1), January 1999.
- [16] J. Schmitt. A Flexible, QoS-Aware IP/ATM Adaptation Module. Technical Report TR-KOM-1999-06, Darmstadt University of Technology, December 1999.