Translation of Specification Units
Between IP and ATM Quality of Service Declarations

Jens Schmitt
Multimedia Communications Lab (KOM)
Department of Electrical Engineering and Information Technology
Darmstadt University of Technology
Merckstr. 25 • 64283 Darmstadt • Germany
http://www.kom.e-technik.tu-darmstadt.de
Email: Jens.Schmitt@KOM.tu-darmstadt.de
Telephone: +49-6151-166163 • Fax: +49-6151-166152

Abstract An integral part of any network Quality of Service (QoS) system are its QoS declarations. QoS declarations consist of service classes, parameters, and specification units. QoS declarations are a component of the QoS architecture, as such they are a source of heterogeneity stemming from the fact that different QoS systems may be based on different QoS architectures and thus use different QoS declarations. A particular problem in that domain is the *translation* of specification units for QoS systems that are based on different forwarding technologies with respect to variable vs. fixed packet sizes, i.e., packet vs. cell switching. This is a problem that can be dealt with generically such that its solution can be applied to several situations of technically heterogeneous QoS systems like an RSVP/IntServ- or DiffServ- over an ATM-based system. While straightforward translations have been proposed, we investigate how more efficient translations can be achieved by using a slight but effective modification of existing AAL framing schemes as well as making use of statistical knowledge about packet size distributions.

Keywords IP over ATM, QoS, QoS mapping, ATM, IntServ, DiffServ.

1 Introduction

1.1 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 existence of ATM networks in the backbone of large-scale internetworks, as for example the Internet, is a reality (although it can be observed that some backbone providers move away from ATM technology again). 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 (for a detailed discussion of this see Chapter 1 in [1]). 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 signaling protocols for example, and QoS declarations as for instance the available service classes. We will concentrate on the mapping of QoS declarations between IP and ATM networks, and here in particular on the translation of specification units. With regard to the procedural aspects of mappings, see for example [2], [3], or [4], yet there are many more.

Different QoS declarations may be the most obvious hurdle to take in a heterogeneous QoS system. It is immediately clear that the different services classes, parameters, and specification units of QoS systems that are based on different QoS architectures need to be mediated at an edge device between such two QoS systems. In fact, however, QoS declarations are often not that different but rather similar on a conceptual level. For example, both, RSVP/IntServ and ATM, offer service classes for hard real-time traffic: RSVP/IntServ provides Guaranteed Service (GS) and ATM has Constant Bit Rate (CBR) and real-time Variable Bit Rate (rt-VBR) for that purpose. Furthermore, both use the concept of token buckets to regulate traffic that is injected into the network. The actual differences with respect to QoS declarations are in the very details of service classes and parameters.

The result of this discussion is that the interworking problems related to different QoS declarations in heterogeneous QoS systems are very much tied to the details of the respective service classes and parameters. Therefore, it is difficult to find problems in this area that have the potential to be treated in a generic way. In fact, it is our belief that the mapping of service classes and parameters from one set of QoS declarations to another must be done for each possible pair of QoS architectures individually. Although one can certainly "learn" from existing mappings, it is difficult to generalize without abstracting too much from the actual problem. There is, however, one problem that has some potential to be dealt with generically. That is the *translation* of specification units for packet-based into cell-based performance parameters. Since it is our high-order goal to investigate *generic* interworking problems for heterogeneous QoS systems that translation is the focus of this article. A further argument for concentrating on this issue is that there is existing and very valid work on the mapping of service classes and parameters for the major existing QoS architectures like RSVP/IntServ, DiffServ, and ATM, so that we can refer to this work whereas the issue of translating performance parameters has been given little attention so far.

1.2 Outline

In the next section, we present a brief review of the QoS declarations in the Internet QoS architectures RSVP/IntServ and DiffServ as well as for ATM. In order to give a comprehensive view of different QoS declarations in heterogeneous QoS systems, we review existing work in the area of mapping service classes and parameters for the most prominent configurations of heterogeneous QoS systems for IP/ATM networks in Section 3. Afterwards, we turn to the translation problem, and analyze straightforward approaches to translation as proposed in the literature. We show that these may have detrimental effects on the efficiency of a mapping between QoS declarations, and identify the two major problems which lead to these inefficiencies. Thereafter, translation approaches to solve or at least alleviate these problems are presented, analyzed, and compared to the straightforward approach. At the end of the article, we first give a concluding, illustrative example for the effectiveness of our translation schemes, before we review related work and summarize our main findings.

2 Review of Internet and ATM QoS Declarations

2.1 ATM (Forum) QoS Declarations

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 [5]:

- 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 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 respectively cells/s, even for the GFR service. For the exact definition of the parameters and their applicability to the service categories, see [5].

2.2 IETF Models

Much work inside the IETF has been devoted to the development of QoS architectures for the Internet. The outcome are two different models : IntServ (Integrated Services) and DiffServ (Differentiated Services). These however deal with different needs and can also be seen as complementary and mutually assisting [6], and not necessarily competing.

2.2.1 IntServ QoS declarations

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 new service classes have advanced to proposed standards: Guaranteed Service (GS) [7] and Controlled Load Service (CLS) [8]. GS offers deterministic guarantees on the maximum end-to-end delay and the available bandwith 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.

2.2.2 DiffServ QoS Declarations

This model [9] 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) [10]
- Assured Forwarding (AF) [11]

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).

3 Mapping of Services Classes and Parameters

We have argued in Section 1 that the best which can be done with respect to the mapping of service classes and parameters is to look at each pair of QoS architectures and see how classes should be matched and how parameters are assigned to each other. There is a considerable amount of work, especially within the IETF, that does just that for the most prominent combinations of QoS architectures. In the IP/ATM network context, these are RSVP/IntServ over ATM and DiffServ over ATM. We briefly review these examples and highlight the most important points of the respective mapping.

3.1 RSVP/IntServ over ATM

The discussions on the mapping of RSVP/IntServ's to ATM's classes and parameters are conducted along the RSVP/IntServ service classes GS and CLS. The best-effort service class is left out as it represents a rich research area of its own (see, for example, [12] and references therein). The discussion is based on the proposed standard RFC 2381 [13] which describes in much detail how IntServ service classes may be mapped to any of the ATM service categories.

Guaranteed Service For GS the only possible ATM service categories on which it can be mapped are CBR and rt-VBR. The other ATM service categories do not support real-time services as required for GS. The use of rt-VBR may allow to recover some unused bandwidth by a bursty source so that it is generally preferable over CBR. Furthermore, it is easier to support very bursty sources by rt-VBR without using a large amount of buffer space inside the edge device. In fact, the use of rt-VBR allows to "shift" some of the buffering necessary at the edge device inside the ATM network. The mapping of parameters depends on the selection of the service category for GS. For a mapping of GS to CBR, the following relationship needs to be taken into account:

$$p \geq PCR \geq R \tag{1}$$

If *PCR* is chosen equal to *R*, then the edge device is assumed to absorb bursts in its IP-level buffers. If *PCR* is chosen greater than *R*, then the buffer at the edge device can be dimensioned smaller. Obviously, the edge device has some degrees of freedom for balancing its local resource consumption with ATM network resource consumption. For a mapping of GS to rt-VBR, a larger set of parameters has to be set according to the following relationships:

$$\begin{aligned} &p \geq PCR \geq R \\ &PCR \geq SCR \geq r \\ &MBS \geq b \end{aligned} \tag{2}$$

The most straightforward assignment would be $PCR = R$, $SCR = r$ and $MBS = b$. However, as for CBR there are possible trade-offs with respect to which entity is absorbing bursts: the edge device or the ATM network. Note that the above equations cannot be taken literally since the quantities are specified in different units (cells vs. bytes) and link layer overheads are also not taken into account. This translation is what we deal with in the following sections.

Controlled Load Service The recommended mappings for CLS are CBR, nrt-VBR, ABR, or GFR. While a mapping based on CBR is probably wasteful in ATM network resources, it tends to be a simple solution, in particular if not all CLS flows are given a separate VC but they are sharing a single CBR VC. The nrt-VBR service category is considered the best match as it is very similar to CLS and the handling of excess traffic can be achieved by cell tagging, which is very much in the virtue of the CLS specification. Other good matches are ABR and GFR as they are based on the "best effort with floor" paradigm similar to CLS in that they are providing an *MCR* and sharing on top of this. The rt-VBR category may be used but is considered wasteful and does not offer the simplicity of a CBR VC. The assignment of parameters depends again on the service category on which CLS is mapped. For CBR the following relationship must be ensured:

$$PCR \geq r \quad . \tag{3}$$

If *PCR* is set equal to *r*, then the edge device must be prepared to buffer up to *b* bytes. For nrt-VBR the following assignment is recommended:

$$PCR = p, SCR = r, MBS = b. \tag{4}$$

This mapping would require no buffering at the edge device. Again, the edge device can do buffering to lower the values for *PCR* and *MBS*. For ABR as well as GFR, it is recommended to set *MCR* = *r* and provide a buffer of *b* bytes at the edge device.

Further issues for mapping RSVP/IntServ's QoS declarations are how to advertise the ATM network in an AdSpec object, how excess traffic is treated, and how to set the *maxCTD*, *CDV* and *CLR* parameters (as they do not have any counterparts in RSVP/IntServ). A very comprehensive discussion on these issues is contained in [13].

3.2 DiffServ over ATM

The first question to be answered for the mapping of DiffServ to ATM is whether ATM VCs correspond to SLAs or PHBs. Both, the ATM Forum [14] and the IETF [15] favor the latter . So, let us take their view, and discuss the service mapping along the PHBs.

Expedited Forwarding For EF-based SLAs or for the EF PHB, the only sensible candidates are CBR and rt-VBR. In [15], rt-VBR is favored since the use of CBR induces shaping delays at the edge device that might be unacceptable for EF. If EF is mapped to rt-VBR, then the following assignment of parameters is reasonable:

$$\begin{aligned} PCR &= \textit{incoming line rate} \\ SCR &= \textit{EF configured rate} \\ MBS &= \textit{max PDU size} \end{aligned} \tag{5}$$

A further advantage of using rt-VBR over CBR is that overprovisioned bandwidth can be recovered for other traffic to some extent.

Assured Forwarding ABR or GFR seem the most reasonable choices for AF as their service semantics match AF's "better than best-effort" semantics most closely. However, using a CBR with a very large *PCR* is certainly also possible but incurs a very inefficient ATM network resource usage. If ABR or GFR is used for

AF (ABR is recommended in [15]), then *MCR* should be set to the minimum allocated bandwidth for an AF class and each AF class should be provided a separate VC. The advantage of ABR is that congestion within the ATM network is pushed back to the ingress edge device which may then implement active queue management mechanisms for all of the three drop precedences of an AF class. For GFR, a possibility would be to use cell tagging in order to implement drop preferences but obviously only two could be provided. As it has been argued that three drop preferences would be very beneficial in order to be able to separate TCP from UDP out-of-profile traffic (and, of course, in-profile traffic) [16], this is a disadvantage of using GFR.

Note that both EF and AF require the setting of a certain configured rate to satisfy their specified behavior. This rate will most likely be given in bits/s or bytes/s (which are, of course, equivalent for our purposes), and needs to be translated into cells/s.

4 Translation of Specification Units

In the discussions of the preceding section, we have encountered several situations where performance parameters need to be translated from specification units defined for variable-size transport units into ones for fixed-size transport units. From now on, we concentrate on this generic problem of translation. At first, we present and analyze straightforward translations as they are proposed in [13] for IntServ over ATM or in [15] for DiffServ over ATM. Note that while it is our goal to keep the discussion as generic as possible, we often draw upon the specific instance of ATM as a cell-based QoS architecture.

4.1 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 of 48 bytes, 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), 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 can be derived:

$$n_c^{min} = n_c(m) = \left\lceil \frac{m + o_p}{s_c - o_c} \right\rceil \le n_c \le \left\lceil \frac{M + o_p}{s_c - o_c} \right\rceil = n_c(M) = n_c^{max}. \tag{6}$$

Given a certain IP performance-related rate r [in bytes/s] (this could be either the average rate r, the peak rate p for IntServ's TSpec or a configured rate for any of the DiffServ PHBs), we get a packet rate r_p with

$$r_p = \frac{r}{s_p} \text{ [in packets/s],} \tag{7}$$

which again allows to compute the required cell rate r_c with

$$r_c = r_p n_c \text{ [in cells/s].} \tag{8}$$

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:

$$r_c^{min} = \frac{r}{s_p^{min}} \left\lceil \frac{s_p^{min} + o_p}{s_c - o_c} \right\rceil = r_p(s_p^{min}) n_c(s_p^{min}) \le \boldsymbol{r_c} \le r_p(s_p^{max}) n_c(s_p^{max}) = \frac{r}{s_p^{max}} \left\lceil \frac{s_p^{max} + o_p}{s_c - o_c} \right\rceil = r_c^{max} \tag{9}$$

where

$$\begin{aligned} s_p^{min} &= \operatorname{argmax} r_c(s_p) \mid s_p \in [m,M] \\ &= \operatorname{argmax} r_c(s_p) \mid s_p \in \{m, \left\lceil \frac{m + o_p}{s_c - o_c} \right\rceil (s_c - o_c) - o_p + 1\} \end{aligned} \tag{10}$$

$$\begin{aligned} s_p^{max} &= \operatorname{argmin} r_c(s_p) \mid s_p \in [m,M] \\ &= \operatorname{argmin} r_c(s_p) \mid s_p \in \{M, \left\lfloor \frac{M + o_p}{s_c - o_c} \right\rfloor (s_c - o_c) - o_p\} \end{aligned} \tag{11}$$

The latter two equations can be explained by the fact that the maximum (repectively minimum) cell rate r_c can either be taken on at the minimum (respectively maximum) packet size or at the first (respectively last) “spike” of the cell rate curve.

Of course, for ATM $s_c = 48$ and for different AALs the resulting numbers and formulas are given in Table 1, where it is assumed that LLC/SNAP encapsulation as defined in [17] is used for all cases. If instead of that VC-based multiplexing was used, then all o_p values could be diminished by 8.

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	$\left\lceil \frac{s_p + 16}{48} \right\rceil$	$\left\lceil \frac{r}{s_p} \left\lceil \frac{s_p + 16}{48} \right\rceil \right\rceil$

Table 1: Application of the mathematical framework.

This table is slightly speculative as for AAL1 and AAL2, there are no standards or proposals regarding the encapsulation of 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.

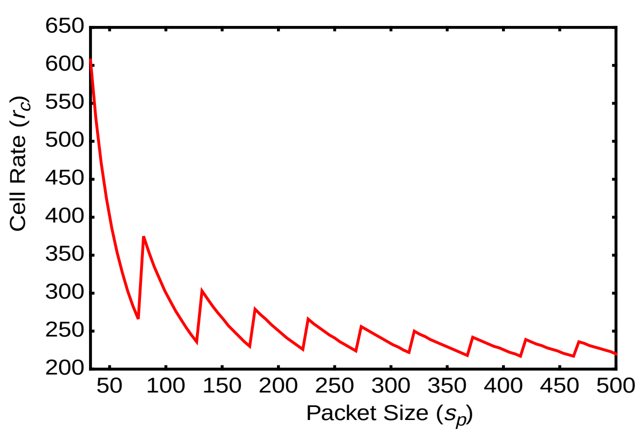

Figure 1: Cell rates for different packet sizes.

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.

4.2 Performance Analysis

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

4.2.1 Metrics

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

$$CUE = \frac{r}{r_c s_c} \in [\frac{r}{r_c^{max} s_c}, \frac{r}{r_c^{min} s_c}] \subset [0,1] \tag{12}$$

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

It may, however, be the case that, when the allocation is made, the expected packet size is not the packet size actually seen in the data flow. Therefore let us define another 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:

$$rCUE(s_p) = \begin{cases} \frac{r}{\bar{r}_c s_c} & s_p < \bar{s}_p \\ \frac{r}{r_c s_c} - \frac{\bar{r}_c - r_c}{\bar{r}_c} & s_p \geq \bar{s}_p \end{cases} \tag{13}$$

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:

$$wCUE = rCUE(s_p^{min}) = \frac{r}{r_c^{min} s_c} - \frac{\bar{r}_c - r_c^{min}}{\bar{r}_c} \tag{14}$$

In any case that means that it is favorable 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:

$$CLR(s_p) = \begin{cases} 1 - \frac{\bar{r}_c}{r_c} & s_p < \bar{s}_p \\ 0 & s_p \geq \bar{s}_p \end{cases} \tag{15}$$

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:

$$wCLR = CLR(s_p^{max}) = 1 - \frac{\bar{r}_c}{r_c^{max}} \tag{16}$$

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

4.2.2 Discussion

Let us now take a look at how the straightforward translation of IP performance parameters to 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. Furthermore, we assume m = 33 and M = 500.

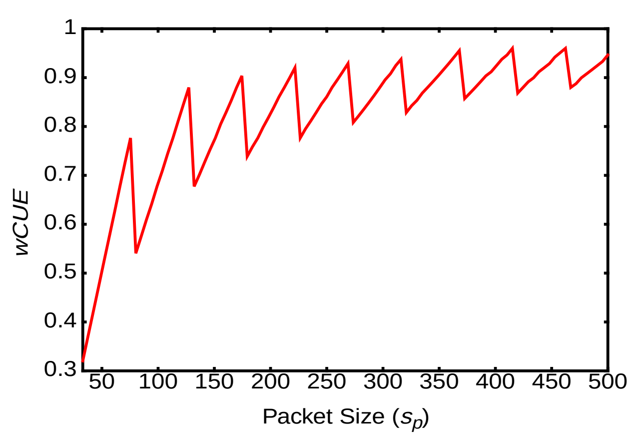

Figure 2: Worst-case cell utilization efficiency.

There are two basic and orthogonal problems that lead to inefficient use of cell rate resources as illustrated in Figure 2:

1. Over-reservation due to 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 “unfortunate” 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.

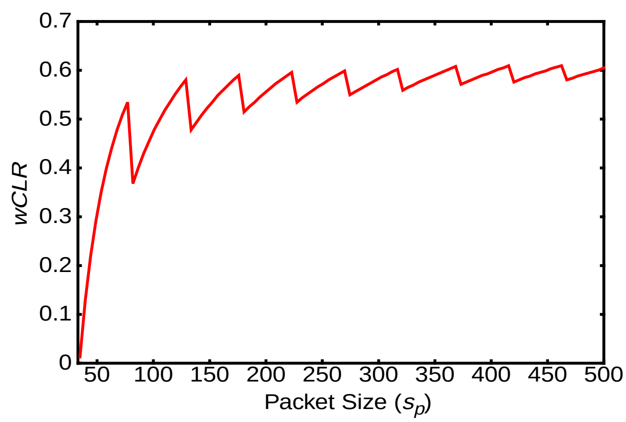

Figure 3: Worst-case cell loss rate.

Of course, the *wCLR* rises as the packet size on which the cell rate allocations are based increases. Furthermore, 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 for example 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 such strict guarantees, a trade-off between the risk of incurring cell loss and an improved efficiency is possible.

All of the above assumes that the packet size is an uncontrolled variable. Certainly, 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 infeasible or at least inconvenient 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 packet- and cell-based QoS architectures (like the IETF proposals and ATM) have to cope with uncertainty about packet sizes and with unluckily sized packets that do not suit the cell stream well. While solution approaches to the former problem, which we call the “unknown number of packets” problem, are dealt with in Section 6, we address at first the latter problem by a scheme we call cell-aligned framing.

5 Efficient Translation Based on Cell-Aligned Framing

5.1 Idea

The straightforward translation scheme presented and analyzed 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.

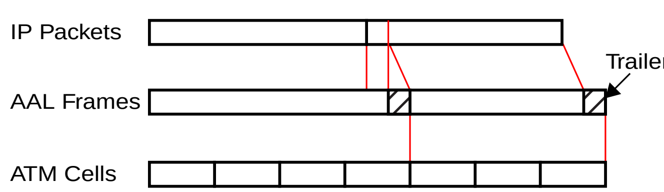

Figure 4: Cell-aligned framing.

This scheme requires that there is a way to mark the start of a new packet inside an AAL frame. This may result in some additional protocol overhead which, however, as we demonstrate in Section 5.4, should not be prohibitive. Besides, 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., if the worst case of a stream sending bursts at s_p^{max} sized packets is actually occurring because the rate calculations have to be based on this case (at least for hard guarantees as 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 the current packet size distribution of IP traffic, see Figure 5, which was produced by [18] from a 24 hour traffic trace at an OC3 link of the MCI network backbone.

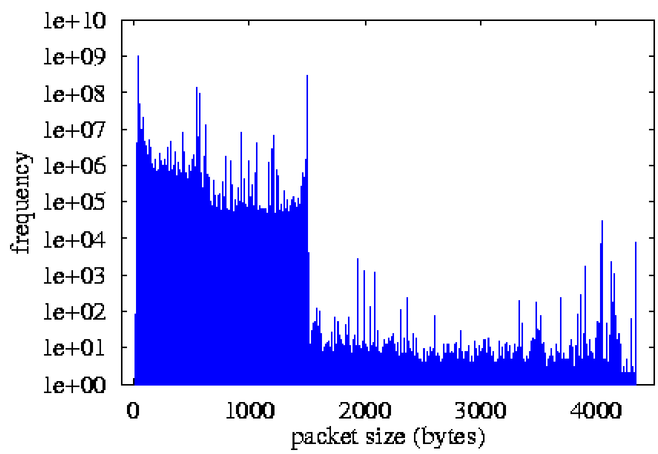

Figure 5: Typical packet size distribution.

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 for example the packet size distribution since the corresponding applications as for instance audio and video streaming. or IP telephony have their very own characteristics.

5.2 Analysis and Comparison

Using the notation and definitions of Section 4, let us analyze translation based on cell-aligned framing, and compare it with the straightforward translation approach:

Overhead for cell-alignment: o_{align} [in bytes].

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

$$r_c(s_p) = \left\lceil \frac{r}{s_p} \times \frac{s_p + o_p + o_{align}}{s_c - o_c} \right\rceil \quad (17)$$

where we have the following bounds on r_c

$$r_c^{min} = \left\lceil \frac{r}{M} \times \frac{M + o_p + o_{align}}{s_c - o_c} \right\rceil \le r_c \le \left\lceil \frac{r}{m} \times \frac{m + o_p + o_{align}}{s_c - o_c} \right\rceil = r_c^{max} \quad (18)$$

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

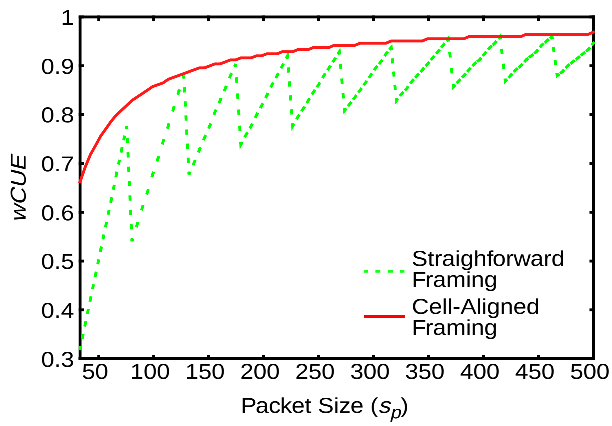

Figure 6: Worst-case cell utilization efficiency.

We used the same settings as in the examples before and assumed no overhead for the cell-alignment, which is possible for AAL5 as demonstrated in Section 5.4. 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.

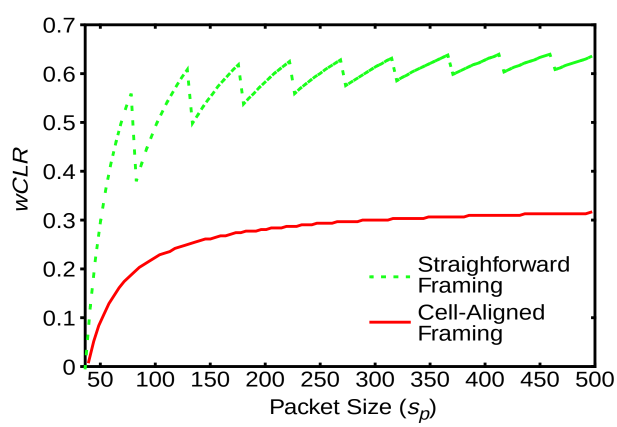

Figure 7: Worst-case cell loss rate.

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.

5.3 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 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 subsequent 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 need to be applied whereas if the rate is not fully used, then the wastage of cell space is not an issue as there is enough space anyway. The main point is that the a-priori translations which are based on worst-case scenarios can be kept low.

5.4 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.

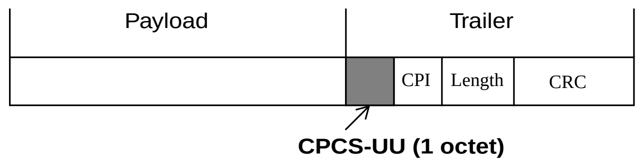

Figure 8: CPCS-PDU format for AAL5.

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 semantic 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 filled with the beginning of the next packet, as has been proposed above. Note that UU = 0 means that only one IP packet is contained in an AAL5 frame. That may be due to the fact that the encapsulated IP packet plus overhead fitted exactly into an integral number of ATM cells or because it has not been necessary to use cell-alignment at the sending side since the cell rate is over-dimensioned, anyway.

In Figure 9, the required protocol processing for cell-aligned framing is illustrated in pseudocode for both, sending and receiving side.

Sender-Algorithm

```
n = 0;
FOREVER
   wait for next packet p[n];
   compute #bytes2pad for p[n];
   IF #bytes2pad!=0 AND p[n+1]
            get #bytes2pad from p[n+1];
            fill them together with
            p[n] in AAL5 frame F;
            F.cpcs-uu = #bytes2pad;
   ELSE
            F.cpcs-uu = 0;
   send F to lower layer;
   n = n + 1;
```

Receiver-Algorithm

```
n = 0;
FOREVER
   receive AAL5 frame F;
   append F.payload[0,F.length-
            F.cpcs_uu-1] to p[n];
   send p[n] to upper layer;
   p[n+1] = F.payload[F.length-
            F.cpcs_uu,F.length];
   n = n + 1;
```

Figure 9: Cell-aligned framing algorithm at sender and receiver.

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 reassembled potentially 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., $o_{align} = 0$, but introduces a higher protocol processing cost due to the more complicated buffer management which, however, from our perspective, should be justified by the considerable efficiency improvements presented above.

6 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 knowledge 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 favorable 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

$$f(s_p) = \frac{1}{M - m + 1} \tag{19}$$

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:

$$f_a(s_p) = as_p - a\frac{M + m}{2} + \frac{1}{M - m} \text{ with } a \in [-\frac{2}{(M - m)^2}, \frac{2}{(M - m)^2}] \tag{20}$$

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

$$p(CLR = 0 | r_{c,\alpha}) \geq 1 - \alpha \tag{21}$$

which means the probability to incur cell loss if we allocate $r_{c,\alpha}$ is less or equal to 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 $G(r_p)$, where the packet rate is defined as in (7): $r_p = \frac{r}{s_p}$.

From this the quantilized 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:

$$r_c = \left\lceil \frac{r + r_p(o_p + o_{align})}{s_c - o_c} \right\rceil. \tag{22}$$

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 quantilized cell rates, note that

$$\begin{aligned}
p(CLR = 0 | r_{c,\alpha}) &= p(\bar{r}_c < r_{c,\alpha}) \\
&= p\left(\left\lceil \frac{r + r_p(o_p + o_{align})}{s_c - o_c} \right\rceil < \left\lceil \frac{r + r_{p,\alpha}(o_p + o_{align})}{s_c - o_c} \right\rceil\right) \\
&> p\left(\frac{r + r_p(o_p + o_{align})}{s_c - o_c} < \frac{r + r_{p,\alpha}(o_p + o_{align})}{s_c - o_c} + 1\right) \\
&= p\left(r_p < r_{p,\alpha} + \frac{s_c - o_c}{o_p + o_{align}}\right) \\
&= G\left(r_{p,\alpha} + \frac{s_c - o_c}{o_p + o_{align}}\right) \\
&= 1 - F\left(r \middle/ \left(r_{p,\alpha} + \frac{s_c - o_c}{o_p + o_{align}}\right)\right)
\end{aligned} \tag{23}$$

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

To compute the cell rates from (23), note that from (22) it follows that

$$r_{p,\alpha} \le \frac{r_{c,\alpha}(s_c - o_c) - r}{o_p + o_{align}} \tag{24}$$

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

$$r_{c,\alpha} \ge \frac{r}{s_c - o_c}\left(1 + \frac{o_p + o_{align}}{F^{-1}(\alpha)}\right) - 1 \tag{25}$$

which allows us to compute the quantilized cell rates as

$$r_{c,\alpha} = \left\lceil \frac{r}{s_c - o_c}\left(1 + \frac{o_p + o_{align}}{F^{-1}(\alpha)}\right) - 1 \right\rceil \tag{26}$$

In Table 2, some example values of quantilized cell rates are given for the sample packet size distributions (19) and (20). We use the same parameter settings as for the examples in preceding sections (in particular we use $m = 33$ and $M = 500$) and for the parameter a of the trapezoidal distribution, we take the extreme values $\pm 2/(M-m)^2$ which represent very optimistic respectively pessimistic assumptions on the packet size distribution.

$r_{c,\alpha}$	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.2$
uniform	*296*	*267*	*250*	*234*
optimistic trapezoidal	*251*	*232*	*226*	*222*
pessimistic trapezoidal	*303*	*282*	*267*	*248*

Table 2: Quantilized cell rates.

Alternatively and similarly, the *CUE* could be taken as a metric to define quantilized 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.

7 Concluding Example

At the end of this article, we want to present a comprehensive and illustrative example of how different translation schemes perform. We use again an IP-performance related rate r = 10000 bytes/s and the range of possible packet sizes for the regarded traffic stream as [33, 500]. The result of different translations for r is depicted in Figure 10.

As can be seen, the straightforward translations, as proposed in [13], lead to a cell rate of 607 cells/s whereas a translation based on cell-aligned framing yields a cell rate of 310 cells/s. If the service semantics permit, this cell rate can be further decreased to 232 cells/s by assuming an optimistic trapezoidal packet size distribution and setting $\alpha = 0.05$. Obviously, cell-alignment gives the major savings whereas assumptions on packet sizes have a comparably moderate effect. Furthermore, they are only applicable to some service classes. On the other hand, they do not involve any change or extension of existing protocols as described above for cell-aligned framing.

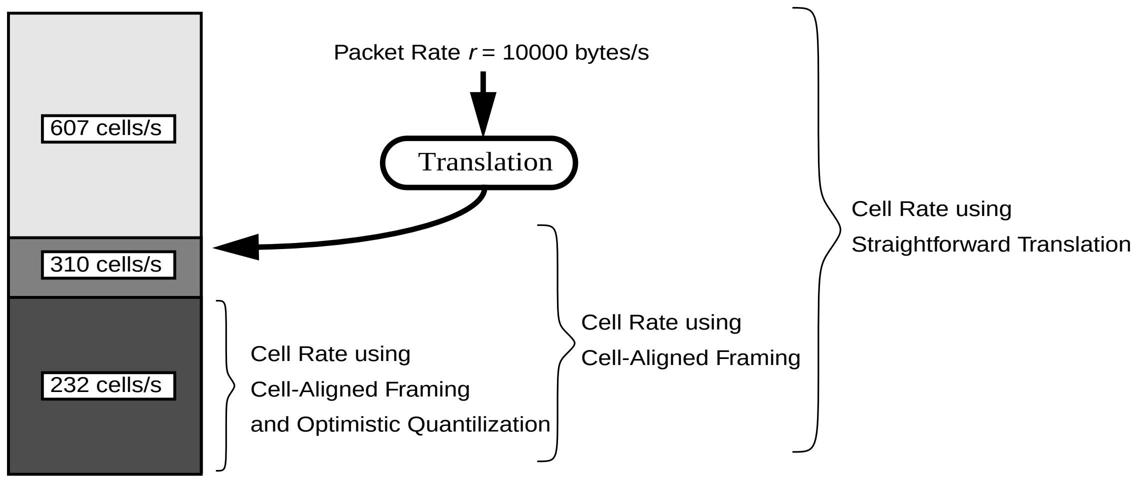

Figure 10: Example of different translations.

8 Related Work

Directly related to the issue of mapping QoS declarations, the most important work has been done in the IETF and ATM Forum. As mentioned above, for IntServ, a proposed IETF standard exists [13] which gives very detailed treatment on how to choose the ATM service categories for the GS and CLS classes. Similarly, there is work in the IETF [15] and ATM Forum [14] on the mapping of PHBs respectively SLAs to ATM service categories. Non-standardization work concerned with those issues can be found in [19], where it is shown that the IntServ to ATM mappings proposed in [13] are at least dubious, as they are shown to lead to excessive cell loss in simulations. However, it is not investigated in [19] what are the reasons for this inaccuracy nor is there any proposal how to circumvent the high cell loss rates. The authors of [20] are especially concerned with how to map CLS flows to ATM service categories, and give some simulation results on their specific mapping scheme. This scheme is based on grouping a number of CLS flows into a single CBR or nrt-VBR VC. In their simulations, the authors of [20] argue that nrt-VBR has little benefit over the simpler CBR service.

All of these do not consider the translation of different specification units for parameters in the detailed manner we have done in this article. In addition, they restrict their investigations to a certain IP QoS model or even only parts of it whereas our work is generally applicable to performance-oriented packet-based network services, of which IntServ and DiffServ are just examples. Furthermore, many 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 packet- and cell-based network performance parameters.

9 Summary

In this article, we have reviewed existing solutions for mapping service classes and parameters for the most prominent combinations of QoS architectures. In the course of this, we have identified a generic problem that has not been dealt with thoroughly by existing work: the translation of specification units for packet-based performance parameters into cell-based ones. After thoroughly analyzing previously proposed straightforward translation approaches, we pinpointed the two main obstacles to an efficient translation of packet- to cell-based performance parameters as segmentation overhead and variability of packet sizes. Consequently, we have introduced and analyzed the approach of cell-aligned framing in order to solve the issue of only partially filled cells due to the segmentation process. Additionally, we have presented a simple and efficient way to implement that approach for the case of ATM AAL5 framing of IP packets. Based on cell-aligned framing, we have proposed a scheme to address the problem of variable packet sizes which cause unknown overhead accounted per packet. The scheme is based on assumptions about packet sizes, and allows for non-deterministic service guarantees to trade-off resource allocation efficiency against cell loss probabilities.

References

1. J. Schmitt. *Heterogeneous Network Quality of Service Systems*. Kluwer Academic Publishers, June 2001. ISBN 0-793-7410-X.
2. L. Berger, E. Crawley, S. Berson, F. Baker, M. Borden, and J. Krawczyk. A Framework for Integrated Services with RSVP over ATM. Informational RFC 2382, August 1998.
3. J. Schmitt, M. Karsten, and R. Steinmetz. Layered Signaling - Motivation, Implementation & Measurements. In *Proceedings of the 25th Annual IEEE Conference on Local Computer Networks (LCN'2000)*, pages 48–58. IEEE, November 2000.
4. 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.
5. ATM Forum Technical Committee: Traffic Management (TM) Specification 4.1, March 1999.
6. Y. Bernet. The Complementary Roles of RSVP and Differentiated Services in the Full-Service QoS Network. *IEEE Communications Magazine*, 38(2):154–162, February 2000.
7. S. Shenker, C. Partridge, and R. Guerin. Specification of Guaranteed Quality of Service. Proposed Standard RFC 2212, September 1997.

8. J. Wroclawski. Specification of the Controlled Load Network Element Service. Proposed Standard RFC 2211, September 1997.

9. K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. Proposed Standard RFC 2474, December 1998.

10. B. Davie, A. Charny, J. Bennett, K. Benson, J.-Y. L. W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis. An Expedited Forwarding PHB (Per-Hop Behavior). Proposed Standard RFC 3246, March 2002.

11. J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group. Proposed Standard RFC 2597, June 1999.

12. R. Goyal. *Traffic Management for TCP/IP over Asynchronous Transfer Mode (ATM) Networks*. PhD thesis, Department of Computer and Information Science, The Ohio State University, 1999.

13. M. W. Garrett and M. Borden. Interoperation of Controlled Load and Guaranteed Service with ATM. Proposed Standard RFC 2381, August 1998.

14. ATM Forum Technical Committee: Addendum to Traffic Management 4.1: Differentiated UBR, July 2000.

15. S. Ayandeh, A. Krishnamurthy, and A. Malis. Mapping to ATM Classes of Service for Differentiated Services Architecture. Internet Draft, November 1999. Work in progress.

16. M. Goyal, A. Durresi, P. Misra, C. Liu, and R. Jain. Effect of Number of Drop Precedences in Assured Forwarding. In *Proceedings of GLOBECOMM'99, Rio de Janeiro, Brazil*, pages 90–97. IEEE, December 1999.

17. D. Grossmann and J. Heinanen. Multiprotocol Encapsulation over ATM Adaptation Layer 5. Proposed Standard RFC 2684, September 1999.

18. K. C. Claffy, G. Miller, and K. Thompson. The Nature of the Beast: Recent Traffic Measurements from an Internet Backbone. In *Proceedings of INET'98*, June 1998.

19. P. Francis-Cobley and N. Davies. Performance Implications of QoS Mapping in Heterogeneous Networks Involving ATM. In *Proceedings of IEEE Conference on ATM '98 (ICATM'98)*, pages 529–535. IEEE, June 1998.

20. P. Giacomazzi and L. Musumeci. Transport of IP Controlled-Load Service over ATM Networks. *IEEE Network*, 13(1):36–47, January 1999.

Jens Schmitt Jens Schmitt is research group leader of the Multimedia Distribution & Networking group at the Multimedia Communications Lab (KOM) of the Darmstadt University of Technology, where he works in the fields of Quality of Service (QoS) provisioning in distributed systems, in particular in heterogeneous network scenarios, QoS for mobile communications, and scalable distribution of multimedia content with an emphasis on high availability systems. Further research interests are in network traffic modelling, real-time scheduling, and evolutionary programming. He has taught several courses in these areas and produced over 50 refereed publications.

Jens Schmitt received his Master degree (Dipl.) from the University of Mannheim in Joint Business and Computer Sciences in 1996. In 1994, during a stay at the University of Wales, Swansea, he also did a European Master of Business Sciences degree. In 2000 he received his Ph.D. degree (Dr.-Ing.) from Darmstadt University of Technology, the title of his thesis has been “Heterogeneous Network Quality of Service Systems”.