Aggregation of Heterogeneous Real-Time Flows with Statistical Guarantees

Krishna Pandit, Jens Schmitt, Ralf Steinmetz

Multimedia Communications
Department of Electrical Engineering and Information Technology
Darmstadt University of Technology
Merckstr. 25 • D-64283 Darmstadt • Germany
{Krishna.Pandit, Jens.Schmitt, Ralf.Steinmetz}@kom.tu-darmstadt.de

Abstract: Aggregation of data flows has two major advantages. One is the reduction of state complexity within the network, the other is the saving of resources by statistical multiplexing between the aggregated flows. In this paper, we show a simple, yet effective scheme to aggregate real-time flows which require a statistical guarantee on experienced loss and a deterministic guarantee on the maximum delay on an individual basis. The focus of our aggregation scheme is on the reduction of state complexity. Therefore, we try to maximize the number of flows to be aggregated by the consideration of heterogeneous flows at the cost of maximally saving resources which would require homogenous flows to be aggregated. Our approach is to first gain insight on the buffer occupancy distribution of a single flow. In practice, the buffer occupancy distribution function of a real-time flow can be considered as monotonic decreasing. We show that the uniform distribution, which is analytically very tractable, is always more *pessimistic* than a monotonic decreasing distribution. This allows us to aggregate heterogeneous flows by taking the uniform distribution as a worst-case bound for the individual flows' buffer distributions and exploiting its statistical properties to save buffer resources by statistical multiplexing between the individual flows of the aggregate. Finally, we discuss at which rate such an aggregate of heterogeneous flows has to be served while maintaining the statistical guarantees given to individual flows.

Keywords: QoS, aggregation, statistical multiplexing, resource allocation, buffer occupancy distribution.

1 Introduction

1.1 Motivation

Many new applications especially in the field of multimedia require *Quality of Service* (QoS) assurances in order to satisfy users' expectations. Examples of such applications are interactive streaming media applications, IP telephony, or networked games to name only a few. They can usually be considered as soft real-time applications with fairly stringent delay requirements, yet more relaxed loss requirements. From the perspective of these applications QoS guarantees provided on a per-flow basis are extremely desirable. On the other hand, from the network's perspective, the currently broadly accepted general wisdom is that per-flow traffic management does not scale up to the dimensions required in backbone networks of large-scale internetworks as the Internet. A straightforward solution to this dilemma is to *aggregate* flows within the network in a controlled fashion such that the individual guarantees can still be met, which allows to keep state information within the backbone only for the aggregates. In the case of statistically guaranteed QoS, which for many of the above mentioned applications and many usage contexts is sufficient, there is a further incentive to aggregate flows due to the potential to save resources by statistically multiplexing the individual flows.

It is important to note here the difference between *aggregation* and *multiplexing*. By aggregation we denote the grouping of flows over a sub-network in which only aggregates can be handled. By multiplexing the local aspect of sending several flows over one outgoing link is generally described. Thus, aggregation can be considered as multiplexing over a link, which represents an aggregate's path over the subnetwork, and which can be dimensioned dynamically. Seemingly insignificant, the difference between multiplexing and aggregation expresses itself in the different goals they induce: while multiplexing only targets the efficient use of resources on a link, aggregation has as an additional goal to reduce the number of aggregates in order to improve on the sub-networks' scalability. As we will see, these two goals of aggregation need to be traded off against each other. In particular, many good multiplexing schemes only consider homogeneous flows (with respect to delay, loss, and bandwidth requirements) since this restriction allows for more efficient resource usage. Yet, from an aggregation perspective this may be prohibitive due to restrictions on the number of aggregates that can be supported by a sub-network. Thus our first-order goal is to reduce the number of aggregates by allowing the aggregation of *heterogeneous* flows and only as a second-order goal do we try to save resources for the aggregate.

While there is a lot of basic work on controlled multiplexing to achieve statistical QoS guarantees for individual flows (which is discussed as related work in the next section), there is only little work on how to aggregate statistically guaranteed flows [1]. In particular, [1] focused on a class-based form of aggregation which still required the subnetwork to react to each individual flow request. In our work, we rather extend the topological aggregation scheme investigated for deterministic services in [2]. This scheme is based on aggregation between flows sharing the same ingress and egress nodes for traversal over the sub-network.

1.2 Related Work

Knightly and Shroff [3] give a very good and comprehensive overview and evaluation of different admission control methods for statistical QoS provision. In the following, we briefly review the most important schemes following their classification:

Average / peak rate combinatorics: In these models ([4], [5]) a source is described as an on/off source. In [5], the admission control is done by computing the distribution of the aggregate arrivals at a bufferless multiplexer, in [4] by computing the probability for a delay-bound violation for an EDF (Earliest Deadline First) scheduler.

Additive effective bandwidths: Another way to conduct statistical multiplexing ([6], [7], [8], [9]) is to assign each flow a bandwidth between its average rate and its peak rate. This is referred to as *effective bandwidth* and is a function of the required loss probability and the particular flow's statistical properties (e.g., autocorrelation function, or peak and average rate together with mean burst duration).

Engineering the loss curve: The *loss curve* is the relationship between loss probability and buffer size. In this approach finding a model for this relationship that best resembles experimental results is targeted. In [10], the loss curve is modelled as $P(\underline{S} > z) \approx e^{-Kz}$ for large buffer sizes and with the histogram model from [11] for small buffer sizes. This is referred to as *hybrid scheme*.

Maximum variance (MV) approaches: MV approaches [12] are based on the observation that aggregate arrivals at a node are Gaussian. This is a reasonable assumption if the number of flows to be aggregated is large.

[3] evaluates the most prominent schemes with a collection of MPEG traces from [13] and Markov modulated on/off sources. Only strictly homogenous sources are evaluated here, which is typical for such a study.

1.3 Outline

In the next section, we investigate the buffer occupancy distribution function for regulated real-time flows. We argue that their density is typically monotonic decreasing by showing empirically that for realistically shaped traffic sources with a stringent delay constraint the buffer histogram in a simple single-server queuing system exhibits the corresponding shape. Motivated by this observation, we then propose in Section 3 the uniform distribution as a worst-case buffer occupancy distribution and then show that it really can be considered as a worst-case distribution for monotonic decreasing density functions. In Section 4 this result is then exploited in order to dimension the shared buffer of an aggregate of heterogeneous flows via a Chernoff bound on the individual flow's buffer occupancy distribution. While this allows to reduce the buffer requirements for the aggregate flow (apart from our first-level goal of reducing the state complexity within the sub-network / backbone), we also briefly discuss in Section 5 how this can also lead to a reduction of the rate for the aggregate. Section 6 then concludes our work and outlines areas of future research.

2 The Buffer Occupancy Distribution for a Single Flow

Our method is to determine the buffer distribution function for a single flow and infer from it the behaviour of a shared buffer for an aggregate. The buffer occupancy of a single flow depends on the properties of the source and the server.

As mentioned above, our main goal is to provide a method which allows to aggregate as many flows as possible. Of course, the more different the flows are allowed to be, the more of them can be aggregated. A flow is characterized by its average rate, buffer space, delay and loss rate. We only require the maximum delay and the packet loss to be approximately similar. This is a fuzzy requirement which emerges since, after the aggregation, all flows are treated the same. In order to avoid uncontrolled QoS violations, we have to assign each flow the tightest delay and smallest loss rate that appear among all flows that are to be aggregated. Ultimately, it comes down to an optimization problem, how many aggregates are required given a set of flows with individual parameters. While this is an interesting issue, it is not subject of this paper, but we deal with the preceding step of how to dimension the resource allocation for an aggregate here.

In order to aggregate heterogeneous flows, we need a description of a single flow that behaves worse than each one of the flows being aggregated. We refer to this as worst-case distribution. It is our strong belief that for realistic delay constrained and regulated flows the buffer occupancy distribution is monotonic decreasing which makes the uniform distribution appropriate for this purpose as we will show in the next section. To make our point, we empirically show that the buffer occupancy distribution of a flow with the following properties is monotonic decreasing:

1. It is *(B,r)* constrained.
2. The server rate is constant and higher than the average arrival rate.
3. It has a fixed delay requirement.
4. A packet that is dropped is not re-sent.

A typical example of a flow with these properties is a real-time MPEG flow. We exemplarily show such a buffer occupancy. From the collection in [13] the movie trace *term_.IPB* is taken, which was encoded using the UC Berkeley MPEG-1 encoder [14]. The trace is given in the form of a list of *40.000* frame sizes a_i. The largest frame size is *9.9 Kbyte*. The next step is to fit a *(B,r)* token bucket on it so that no data is lost. The average frame size $E\{a_i\}$ is

1.36 Kbyte. The minimum service rate is *E{a_i}/*Δ*t*, where Δ*t* is 40 ms, the time between two frames. With *25* frames per second, the minimum average service rate is

$$R_{min} = \frac{25E\{A_i\}}{\text{sec}} = 34\frac{\text{Kbyte}}{\text{sec}} \qquad \text{(Eq. 1)}$$

Simulations show that using this service rate requires a buffer of size *1100 Kbyte* in a lossless system, which causes a worst-case delay of

$$d = \frac{B}{R} = \frac{1100\ \text{Kbyte}}{34\ \text{Kbyte/sec}} = 32\ \text{sec} \qquad \text{(Eq. 2)}$$

This, of course, is not acceptable and we will target a delay *d* of less than *100 ms*.

Next we fit a token bucket to the flow. There are many techniques to determine the parameters for a token bucket [15]. We do not go into detail here, but instead use a simple algorithm. In accordance with the form of the trace, we model the buffer as a discrete process where

$$\underline{b}(k) = b(k-1) + \underline{a}(k) - r(k) \qquad \text{(Eq. 3)}$$

b(k) is the current buffer occupancy, *a(k)* is the frame size and *c(k)* is the server rate per frame. Incorporating that it is bounded by *0* and B_{max} it becomes

$$\underline{b}(k) = max\{0, min\{B_{max}, b(k-1) + \underline{a}(k) - \underline{r}(k)\}\} \qquad \text{(Eq. 4)}$$

Now we have to determine feasible values for *B* and *r*. Recall the relationship between buffer size, server rate and maximum delay.

$$r = \frac{B}{d} \qquad \text{(Eq. 5)}$$

The buffer occupancy is calculated as follows. We set the buffer size to *B = max{a_i}* and fix the delay *d = 96 ms*, which is equivalent to *2.4* Δ*t*. This leads to a rate of *104 Kbyte/s*. The buffer variable is initially set to *0* and then iteratively calculated with Eq. 4. The resulting buffer is shown in Figure 1.

In Figure 2 the empirical buffer occupancy density function in form of a histogram with 25 bins is depicted. Note that the first bin is left out of the plot. This is because the probability that the buffer is empty is *0.95*, which would ruin the scale. The movie traces *movie2.IPB* and *simpsons.IPB* from the same library are included as well.

As can be seen, apart from a few statistical fluctuations the functions are rapidly decreasing. Note that the ordinate is scaled logarithmically. Figures 3 and 4 are analogous with *d = 80 ms*.

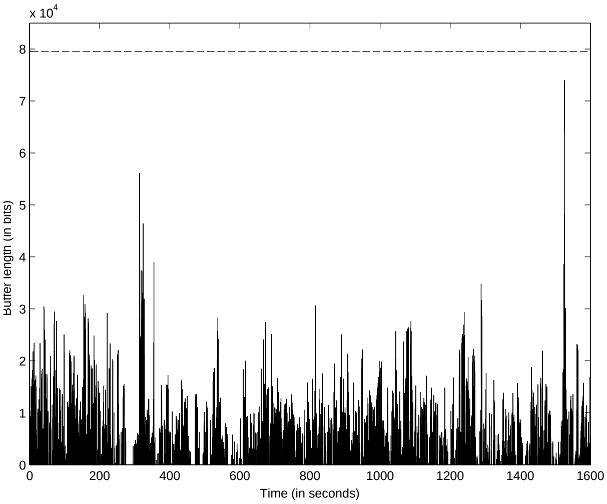

Figure 1: Buffer for *d* = 96 ms.

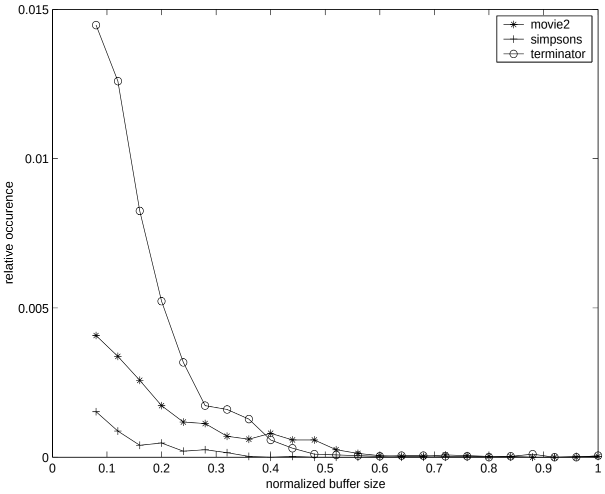

Figure 2: Buffer occupancy density function for *d* = 96ms.

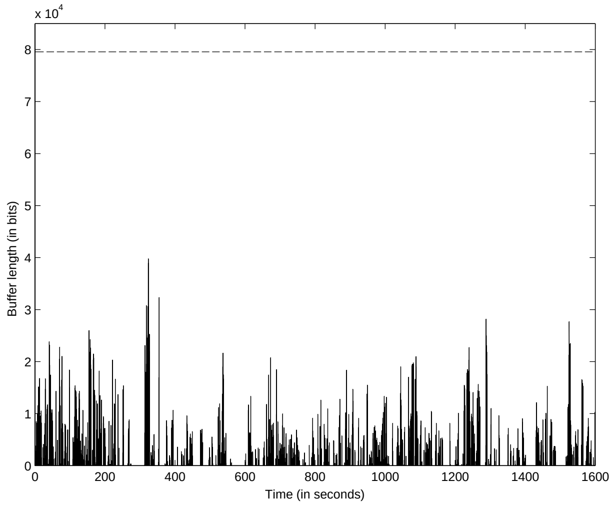

Figure 3: Buffer for *d* = 80 ms.

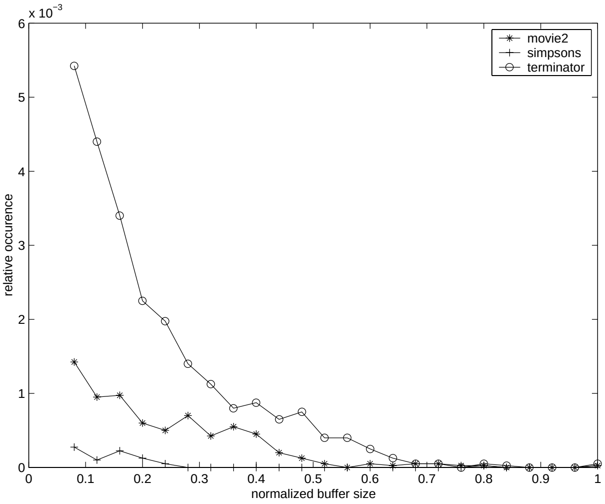

Figure 4: Buffer occupancy density function for *d* = 80 ms.

Here, it can also be seen that the increased rate causes the density function to decrease faster. In this particular trace, the buffer never fills beyond half. This implies that rate and buffer could be optimized but that is not subject of this paper.

Our objective of this section has been to show that for realistic real-time flows, the buffer occupancy density function is monotonic decreasing. We have shown it for a particular case of an MPEG trace. It seems intuitive that this is the case for any flow where the server rate is considerably higher than the arrival rate. If the flow is self-similar, we notice that there are long stretches which have a rate higher than the average rate. To ensure a decent real-time transmission, the server rate has to be much higher than the average rate, which results in the buffer being rather empty most of the time.

3 Worst-Case Buffer Occupancy Distribution

In this section we will prove that to assume uniform distribution is always more *pessimistic* than any one that is monotonic decreasing. A distribution that is more pessimistic than the actual one allows us to aggregate a large number of heterogeneous flows without worrying about the parameters of the individual distributions.

Let $f_{\underline{x},1}(x)$ and $f_{\underline{x},2}(x)$ be two different density functions for the buffer occupancy. The distribution function that the buffer is filled with B or less units, i.e. that $P(\underline{X} < B)$, is then given by

$$F_{\underline{X},\,i}(x) \;=\; \left\{ \begin{array}{ll} 0 & \textit{for } x \le 0 \\ \int_0^x f_{\underline{X},\,i}(y)dy & \textit{for } 0 < x < B_{max} \\ 1 & \textit{for } x \ge B_{max} \end{array} \right. \qquad \text{(Eq. 6)}$$

for $i = 1,2$; and with B_{max} being the maximum buffer size.

Definition 1: A buffer occupancy distribution $F_{\underline{x},1}(x)$ is more *pessimistic* than $F_{\underline{x},2}(x)$, if the probability that the buffer overflows given $F_{\underline{x},1}(x)$ is always greater or equal than given $F_{\underline{x},2}(x)$, i.e., when

$$1 - F_{\underline{X},\,1}(x) \ge 1 - F_{\underline{X},\,2}(x) \qquad \textit{for all } x \qquad \text{(Eq. 7)}$$

In the previous section, we argued that realistic buffer occupancies of real-time flows are monotonic decreasing.

Theorem 1: The uniform distribution is always more pessimistic than a distribution with a monotonic decreasing density function.

Proof: Let u be the sequence denoting the discrete uniform density function

$$u_k \;=\; \frac{1}{N} \qquad \textit{for } 0\ \le k \le N-1 \qquad \text{(Eq. 8)}$$

The discrete distribution is then given by

$$U_k \;=\; \left\{ \begin{array}{ll} 0 & \textit{for } i < 0 \\ \sum_{i=0}^{k} u_i \;=\; \frac{k+1}{N} & \textit{for } 0\ \le i \le N-1 \\ 1 & \textit{for } i > N-1 \end{array} \right\} \qquad \text{(Eq. 9)}$$

A discrete distribution with a monotonic decreasing density function is defined as follows. Let x_k, with $k = 0,1, \ldots ,N-1$ be a sequence where for all k it applies that $x_k \ge x_{k+1}$.

In order to be a valid density function, we have

$$\sum_{i=0}^{N-1} x_i = 1 \qquad \text{(Eq. 10)}$$

The distribution is then given by $X_k = \sum_{i=0}^{k} x_i$

Now we show that U_k is more pessimistic than X_k. In that case, according to Eq. 7,

$$1 - U_k \geq 1 - X_k \qquad \textit{for all } k \qquad \text{(Eq. 11)}$$

which becomes

$$\sum_{i=0}^{k} x_i \geq \frac{k+1}{N} \qquad \text{(Eq. 12)}$$

This we will prove by contradiction. Let us first look at the first points, X_0 and U_0.

$$U_0 = \frac{1}{N} \qquad \text{(Eq. 13)}$$

Now let us assume that

$$X_0 = \frac{1}{N} - \varepsilon < \frac{1}{N} \qquad \text{(Eq. 14)}$$

where ε > *0*. Since $x_k \geq x_{k+1}$, in this case the sum over all *x* is upper bounded by

$$\sum_{i=0}^{k} x_i \leq N x_0 = N\left(\frac{1}{N} - \varepsilon\right) = 1 - N\varepsilon \qquad \text{(Eq. 15)}$$

which clearly is a contradiction to Eq. 10. This line of argument can be extended to the first *M* points. Let

$$\sum_{i=0}^{M} x_i = \frac{M+1}{N} - \varepsilon < \frac{M+1}{N} \qquad \text{(Eq. 16)}$$

Analogous to the case above

$$\sum_{i=0}^{N-1} x_i < \frac{N}{M+1}\left(\frac{M+1}{N} - \varepsilon\right) = 1 - \frac{N\varepsilon}{M+1} \qquad \text{(Eq. 17)}$$

which again contradicts Eq. 10.

Thus it was shown that the uniform distribution is always more pessimistic than a distribution obtained from a monotonic decreasing density function.

4 Efficiently Dimensioning the Aggregate’s Buffer via a Chernoff Bound

In this section, we present how we exploit the single flow buffer occupancy distribution to dimension the aggregate’s buffer efficiently by applying a Chernoff bound on the sum of the individual flow’s buffer occupancy distribution. In our scheme, we substitute the actual buffer occupancy distribution by the uniform distribution as an upper bound which gives us two advantages. First, it allows to aggregate a large number of heterogeneous flows irrespective of the different shapes of their actual buffer occupancy distribution and second, the simple form of the uniform distribution makes its analytical as well as numerical treatment much more practical.

Let G be a set $\{g_1, g_2, ..., g_n\}$ of (B,r)-shaped flows. The flows are heterogeneous, i.e. each flow g_i has a different burst size B_i and rate r_i. Here, we discuss the aggregation of such flows with statistical guarantees. Aggregating deterministically, we would need a buffer of the size

$$B_D = \sum_{i=1}^{n} B_i \qquad \text{(Eq. 18)}$$

According to [16], a loss probability in the magnitude of 10^{-4} to 10^{-9} is targeted. If all flows act independently*, it becomes unlikely that they will burst at the same time. From the derivation of the Chernoff bound [17] we take the following equation. For any random variable $\underline{Y}$ with the density function $f_y(y)$

$$P(\underline{Y} \geq y) \leq e^{-vy} M_{\underline{y}}(v) \qquad \text{(Eq. 19)}$$

$M_y(v)$ is the characteristic function of $f_y(y)$.

$$M_y(v) = \int_{-\infty}^{\infty} e^{vw} f_{\underline{Y}}(w) dw \qquad \text{(Eq. 20)}$$

Recall that the density of a sum of random variables corresponds to the convolution of their density functions. Further, the convolution of two functions in the time domain corresponds to their multiplication in the frequency domain. We now define the random variable $\underline{Y}$ as the sum of the instantaneous buffer occupancies.

$$\underline{Y} = \sum_i^n \underline{b}_i \qquad \text{(Eq. 21)}$$

Therefore, the density function of $\underline{Y}$ is the convolution of all density functions $\underline{b}_i$. The characteristic function $M_{\underline{y}}(v)$ is then the product of the characteristic functions of the individual buffer occupancies.

$$M_{\underline{y}}(v) = \prod_i^n M_{\underline{b}_i}(v) \qquad \text{(Eq. 22)}$$

* Even though this assumption may be arguable, it is made in all relevant approaches to statistical multiplexing known to us.

Hence, the overflow probability of an aggregated buffer B_S is bounded by

$$P(\underline{Y} \geq B_s) \leq e^{-vB_s}\prod_i^n M_{\underline{b}_i}(v) \qquad \text{(Eq. 23)}$$

Since this bound holds for all $v \geq 0$, the tightest bound is

$$P(\underline{Y} \geq B_s) \;=\; \inf_v \left\{ e^{-vB_s}\prod_i^n M_{\underline{b}_i}(v) \right\} \qquad \text{(Eq. 24)}$$

For the individual flow we have to assume the worst case, which is that in the event of a loss the entire buffer of the flow is lost. Adding an additional layer of statistics to find a relationship between the aggregate's loss probability and individual loss probability is subject to further research.

In Figure 5, we show a numerical example for resource usage. The abscissa denotes the number of flows and the ordinate denotes the relative buffer size compared to a buffer of size *1*, which is the deterministic case. I.e., this graph shows, how large the buffer needs to be to ensure the given loss probability for a given number of flows. Eq. 24 is solved for B_S, which obviously can only be done numerically. By toggling the abscissa and ordinate, i.e., solving Eq. 24 for *n*, we obtain an admission control graph: how many flows can be allowed given the loss probability and the buffer size.

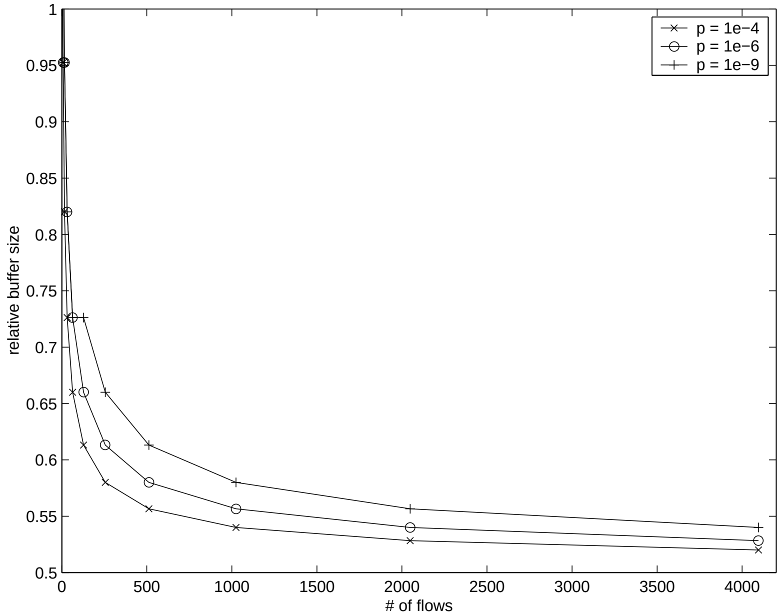

Figure 5: Numerical example.

5 Dimensioning the Aggregate's Rate

In this section, we briefly discuss the dimensioning of the rate with which the aggregate has to be served. Again we first consider deterministic aggregation. Recall that we allow heterogeneous flows, i.e., flows with different delays, to be aggregated. Since the flows are not handled individually after the aggregation, we must assume each flow to have the minimum delay of all flows and assign it the according rate. With Eqs. 5 and 18, this is

$$r_D = \frac{B_D}{\min_i \{d_i\}} \tag{Eq. 25}$$

This rate is larger than the sum of all rates.

$$r_D - \sum_{i=1}^{n} r_i = \sum_{i=1}^{n} B_i \left(\frac{1}{\min_j \{d_j\}} - \frac{1}{d_i} \right) \tag{Eq. 26}$$

This waste of bandwidth is unavoidable if the aggregate is strictly to be treated as one flow. But applying Eq. 5 after the aggregation, we find that the minimum delay is

$$d_S = \frac{B_S}{r_D} = \frac{B_S}{B_D} \min_i \{d_i\} \le \min_i \{d_i\} \tag{Eq. 27}$$

This result, which states that the delay is always better than required, implies an avoidable waste of resources. Intuitively, it seems that this can be solved by just reducing the rate so that the desired delay is obtained. But it has to be ensured that it is at least the sum of the initial rates, i.e., that not more is taken away than what was added.

$$r_S = \max \left\{ \frac{B_S}{\min_i \{d_i\}}, \sum_{i=1}^{n} r_i \right\} = \max \left\{ r_D - \left(\frac{B_D - B_S}{\min_i \{d_i\}} \right), \sum_{i=1}^{n} r_i \right\} \tag{Eq. 28}$$

It is not clear whether this step is allowed. It is an issue that so far has not come up as it only appears when aggregating flows with heterogeneous delays and requires further research.

6 Conclusion and Outlook

In this paper we showed a method to aggregate data flows with statistical guarantees while putting the emphasis on reducing the control complexity. We introduced the concept of a worst-case distribution which is a property that allows many heterogeneous flows to be aggregated. The uniform distribution can be used as a worst-case distribution if the buffer occupancy density function is monotonic decreasing. We believe that this is the case for all *(B,r)*-shaped real-time flows, but could not generally prove it due to the vast amount of degrees of freedom in the model. We then pointed out how some resources, namely buffer space, can be saved by applying large deviation statistics. This uncovered the question what the required server rate is, which is subject to further research.

Several more issues which are subject to further research were revealed during this project. It is not optimal to treat all flows like the one with the tightest delay bound when aggregating heterogeneous flows. The relationship between the aggregate and individual loss probability has not yet been addressed. Finally, the complexity and applicability of the method introduced in this paper have to be studied in more detail.

References

[1] J. Liebeherr, S. Patek, and E. Yilmaz. Tradeoffs in Designing Networks with End-To-End Statistical QoS Guarantees. In *Eigth International Workshop on Quality of Service (IWQOS)*, pages 221–230, 2000.

[2] J. Schmitt, M. Karsten, and R. Steinmetz. On the Aggregation of Deterministic Service Flows. *Computer Communications*, 24(1):2–18, January 2001.

[3] E. Knightly and N. Shroff. Admission Control for Statistical QoS: Theory and Practice. *IEEE Network*, 13:20–29, March/April 1999.

[4] D. Ferrari and D. Verma. A Scheme for Real-Time Channel Establishment Wide-Area Networks. *IEEE JSAC*, 8:368–379, April 1990.

[5] T. Lee, K. Lai, and S. Duann. Design of a Real-Time Call Admission Controller for ATM Networks. *IEEE/ACM Transactions on Networking*, 4:758–765, October 1996.

[6] C. Courcobetis and R. Weber. Effective Bandwidths for Stationary Sources. *Probability in Engineering and Information Science*, 9(2):285–294, 1995.

[7] A. Elwalid and D. Mitra. Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High Speed Networks. *IEEE/ACM Transactions on Networking*, 1:329–343, June 1993.

[8] R. Guerin, A. Ahmadi, and M. Nagshineh. Equivalent Capacity and its Applications to Bandwidth Allocation in High Speed Networks. *IEEE JSAC*, 9:961–981, September 1991.

[9] G. Kesidis, J. Walrand, and C. Chang. Effective Bandwidths for Multiclass Markov Fluids and other ATM Sources. *IEEE/ACM Transactions on Networking*, 1:424–428, August 1993.

[10] N. Shroff and M. Schwartz. Improved Loss Calculations at an ATM Multiplexer. *IEEE/ACM Transactions on Networking*, 6:411–422, August 1998.

[11] P. Skelly, M. Schwartz, and S. Dixit. A Histogram-Based Model for Video Traffic Behaviour in an ATM Multiplexer. *IEEE/ACM Transactions on Networking*, 1:446–459, August 1993.

[12] J. Choe and N. Shroff. A Central Limit Theorem Based Approach to Analyze Queue Behaviour in ATM Networks. *IEEE/ACM Transactions on Networking*, 6:659–671, October 1998.

[13] O. Rose. *Traffic Modelling of Variable Bit Rate MPEG Video and its Impacts on ATM Networks*. PhD thesis, University of Würzburg, Germany, Dept. of Computer Science, 1997.

[14] K. Gong. Berkeley MPEG-1 Encoder, User's Guide, 1994.

[15] O. Heckmann, F. Rohmer, and J. Schmitt. The Token Bucket Allocation and Reallocation Problems (MPRASE Token Bucket). Technical Report TR-KOM-2001-12, http://www.kom.e-technik.tu-darmstadt.de/publications/abstracts/HRS01-1.html, Darmstadt University of Technology, December 2001.

[16] I. Habib and T. Saadawi. Multimedia Traffic Characteristics in Broadband Networks. *IEEE Communications Magazine*, pages 48–54, July 1992.

[17] L. Kleinrock. *Queuing Systems – Theory*. Wiley-Interscience, New York, vol.1, 1975.