A Fast and Trend-Sensitive Function for the Estimation of Near-Future Data Network Traffic Characteristics

Manuel Günter and Torsten Braun
Institute of Computer Science and Applied Mathematics
Neubrückstrasse 10, CH-3012 Bern, Switzerland
http://www.iam.unibe.ch/~rvs/
mguenter@iam.unibe.ch

Keywords: computer networks, extrapolation, estimation, exponential average, differentiated services reservations

Abstract

Computer networking applications often have to adapt to different conditions of the network to guarantee smooth operation. Since events such as network congestion happen at remote locations there is a significant delay until applications notice condition changes. In case the network behaviour is hard to model mathematically, applications often use simple estimation methods such as exponential averaging to react to changing conditions. However, this function is not appropriate for extrapolation because it reacts too slow to trends. We propose an alternative that is also simple and assumes little knowledge aboout the event sources, but which extrapolates better.

1 INTRODUCTION

Many applications use the exponential average function to estimate the future values of unknown generator functions. Examples are the estimation of the round-trip time and its standard deviation in TCP [6] (which is crucial for the TCP timer management [11]), estimation of the mean allowed cell rate in ATM (ATM-Forum) and estimation of CPU burst times [10]. Another application area is the dynamic resource (e.g. bandwidth) reservation in the Internet. Recently, the Differentiated Service (DiffServ) [2] architecture has been standardised to allow prioritization of classes of Internet traffic. In order for this scheme to provide end-to-end quality of service for connections that go along several provider networks, DiffServ requires resource reservations for bulk traffic. For economic reasons network providers should not reserve too much or too little resources. So the amount of resources should reflect the upcoming usage. The upcoming demand can be signalled by a separate protocol. However, this negatively affects the scalability of the DiffServ approach. Another approach is to *estimate* the upcoming demand based on local usage measurements and to use the estimate for in-advance local reservations. Exponential average estimation seems suitable here because of the lack of a mathematical model for DiffServ traffic demand. Note that the demand depends on the (not-yet existing) users, the DiffServ pricing models and the applications that will use DiffServ. For our DiffServ research [5, 3] we compared signalling based reservation schemes with estimation based ones. However, the performance of the estimation based approach heavily depends on the quality of the estimation. With an unreliable estimation, too much capacity is reserved, leading to an under-usage of the network, or even worse, too little resources are reserved, possibly leading to loss of priority traffic. Our simulation results indicated that the exponential average estimation based reservation did not perform sufficiently. Especially for global traffic trends the result was disappointing. This is not only due to the estimation approach itself but to the nature of the exponential average estimation. In this paper we will analyse the problem with exponential average estimation (section 2), propose an alternative (section 3) and evaluate the alternative (section 4). Section 5 concludes the paper.

2 ANALYSIS OF THE EXPONENTIAL AVERAGE

The exponential average τ of the values $t_0, ..., t_n$ is defined as:

$$\tau(n+1) = \alpha t_n + (1-\alpha)\tau(n) \qquad (1)$$

where $\alpha \in [0..1]$ and $\tau(0)$ is an initial value, usually 0. The usage of the exponential average as an estimation function is simple. Given measured values $t_0, ..., t_n$, we use $\tau(n+1)$ as an estimate for the next (currently unknown) value t_{n+1}. Note that later in this paper we will sometimes attach to τ a letter in raised brackets in order to indicate what value sequence is averaged (here, this would be $\tau^{<t>}(n+1)$).

The factor α tailors the exponential average. A large α value puts emphasis on the most recent measured values while a small α value increases the influence of older measurements. In general we can say that the more recent the measurement value is, the more weight it has for the estimation.

The exponential average function as stated in (1) is obviously very easy to calculate. Initialise the exponential average with some value (usually 0) and choose an α. After the first measurement, you are able to calculate the next prediction using only the old prediction and the current measurement value.

Thus, the properties of the exponential average are: 1) It uses only the assumption that the more recently measured values are more significant than older values. 2) Averaging smoothes the estimations thus making it more robust (stable) against 'runaway' values. 3) It is very fast to calculate and needs almost no memory.

For notational convenience let $\beta \equiv 1-\alpha$. Here is then the non-recursive form of (1):

$$\tau(n+1) = \alpha t_n + \beta\alpha t_{n-1} + ... + \beta^n \alpha t_0 + \beta^{n+1}\tau(0)$$

For $\tau(0) = 0$ this boils down to:

$$\tau^{<t>}(n+1) = \alpha \sum_{i=0}^{n} \beta^i t_{n-i} \tag{2}$$

Let's assume that the measurement values are constant ($c_i = C$). The estimation for the $n+1$ th value is thus:

$$\tau^{<c>}(n+1) = \alpha \sum_{i=0}^{n} \beta^i C = \alpha C \frac{1-\beta^{n+1}}{1-\beta} = C(1-\beta^{n+1})$$

The estimation error $|c_n - \tau^{<c>}(n)| = \beta^n$ is shrinking rapidly with an increasing number of measurements n. β is the weight of the older measurements thus also reflecting the initial estimation 0 which is causing the estimation error, but which influence fades away.

This little reflection shows that the exponential average quickly adapts to a series of constant measurement values. However, the exponential average function does not behave so well when we are facing measurements which constantly increase/decrease in value. We demonstrate this for the case of a monotonically increasing[1] set of measurement value t_n. Thus $t_i \geq t_{i-1} (\forall i : i > 0)$. We will prove that if the initial estimate $\tau(0) < t_0$ and $\alpha < 1$ then $\tau(i+1) < t_i$. This means that in such a case the estimation is not only always smaller than the measured value, but it's even smaller than the last *already measured* value. Thus, if the measurements are generated by a monotonically increasing/decreasing function, then the estimation is always at least as far from the new value as the new value is away from the last value. We can prove this property using induction:

1. For $i = 0$: $\tau(1) = \alpha t_0 + \beta\tau(0) \underbrace{<}_{\tau(0)<t_0} \alpha t_0 + \beta t_0 = t_0$
2. Induction assumption: $\tau(i+1) < t_i$.
3. Induction:
$\tau(i+2) = \alpha t_{i+1} + \beta\tau(i+1) \underbrace{<}_{\text{I-assumption, } \alpha < 1} \alpha t_{i+1} + \beta t_i$
$\underbrace{\leq}_{\text{Monotony}} \alpha t_{i+1} + \beta t_{i+1} = t_{i+1}$ q.e.d.

[1]The statement is valid as well (in an analogous way) for monotonically decreasing values.

Note, that for $\alpha = 1$ the estimation error is exactly the difference between the new measurement value and its predecessor. Therefore, for a monotonically increasing function $\alpha = 1$ produces the best estimations. Unfortunately estimating the future value to be exactly the same as the last measured value is a very unstable method for a general purpose estimation.

An example: a simple linear generator function. Consider $l_i = m * i, m > 0$ which is a monotonically increasing function. Using formula 2 provides the exponential average estimation: $\tau^{<l>}(n+1) =$
$\alpha \sum_{i=0}^{n} \beta^i (n-i)m = \alpha m(n \sum_{i=0}^{n} \beta^i - \sum_{i=0}^{n} i\beta^i)$

If $\alpha = 0$ then $\tau_m(n+1) = 0$ which is constant and thus not suitable to estimate the linearly increasing t_n. Otherwise, we can use the following equations to get rid of the sums:

$$\sum_{i=0}^{n} a^i b^{n-i} = \frac{a^{n+1} - b^{n+1}}{a-b} \tag{3}$$

$$\sum_{i=m}^{n} p^i = \frac{p^m - p^{n+1}}{1-p} \tag{4}$$

$$\sum_{i=0|1}^{n} ip^i = \frac{p - p^{n+1}(n(1-p)+1)}{(p-1)^2} \tag{5}$$

This leads to the following simplified estimation formula:

$$\tau^{<l>}(n+1) = m(n - \frac{\beta(1-\beta^n)}{\alpha}) \tag{6}$$

Not surprisingly, since $t_n = m * n, m > 0$ is monotonically increasing, we see that the previous findings are true. According to (6) the estimate for value $n+1$ is a little bit smaller than mn. The correct estimate would be $m(n+1)$. The error is thus $m(1 + \frac{\beta(1-\beta^n)}{\alpha})$ and directly proportional to m. This can be intolerably large. In case n is large, the error is approximately $m(1 + \frac{\beta}{\alpha}) = m/\alpha$. Figure 1 shows the exponential average estimation for $m = 4, \alpha = 0.5$.

2.1 Motivation

Measurement values in IP networks are not random but expose some global trends. We speak of 'global trends' when the average over consecutive intervals of measurement values increases (decrease) steadily for several intervals. The linear functions discussed in the previous section are a special case

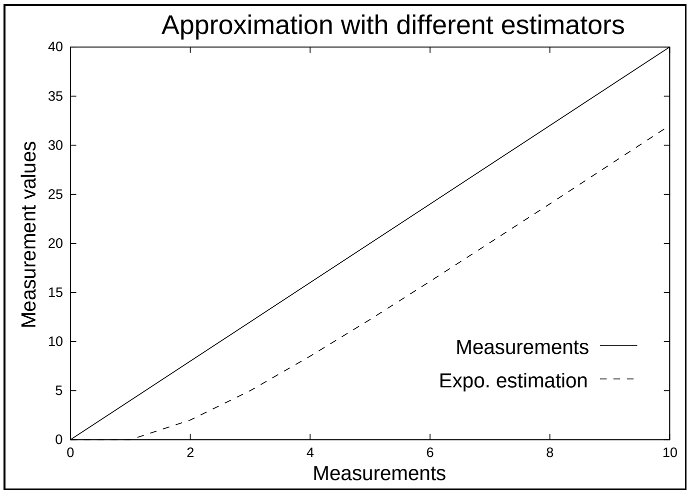

Figure 1: The exponential average estimation for a linear function.

of values with an inherent global trend. The exponential average estimation fails to detect and exploit these trends. Global trends are often observable in data networks such as the Internet. E.g. if a site becomes popular, the traffic to- and from that site increases steadily [1]. Another example is the traffic rate at different daytimes (see figure 2). E.g. on the beginning of the office hours, the global traffic volume increases steadily and rapidly. For our research in Differentiated Services we also assume reservation demand that follows global trends. The INDEX study on user behaviour (given a service differentiation) is motivating this assumption [4]. One way to calculate predictions based on given measurements is a thorough statistical analysis of the available data. Mathematical tools for such analysis are available [9]. However, the analysis usually works with an underlying model for the generation of the data. As discussed in the section 1, on one hand we are not even near of having a useful mathematical model for DiffServ traffic demands. This favours the use of the exponential average estimation. On the other hand DiffServ demand will probably also follow global trends e.g. at different daytimes. In our simulation a special global trend emerges at the beginning of the simulation. Then, no reservations are set up and DiffServ traffic starts flowing through the network. Since no reservation is there, traffic loss occurs. Local measurements are then used to estimate the demand. Based on these estimates the simulation sets up local reservations. The more reservations are set up the deeper the DiffServ traffic penetrates the mesh of provider networks until sufficient reservations are set up throughout the networks. Thus, in the first few simulation rounds there is a global increasing trend of DiffServ demand. The better the estimation function adapts to this trend, the shorter this initial phase of heavy loss is. We have showed in theory why the exponential average estimation performs insufficient in this situation and our simulations confirmed the finding (see also figure 4.2). We therefore started to look for an estimation function which is as stable, fast, easy to implement as the exponential average estimation, but which can cope with global trends.

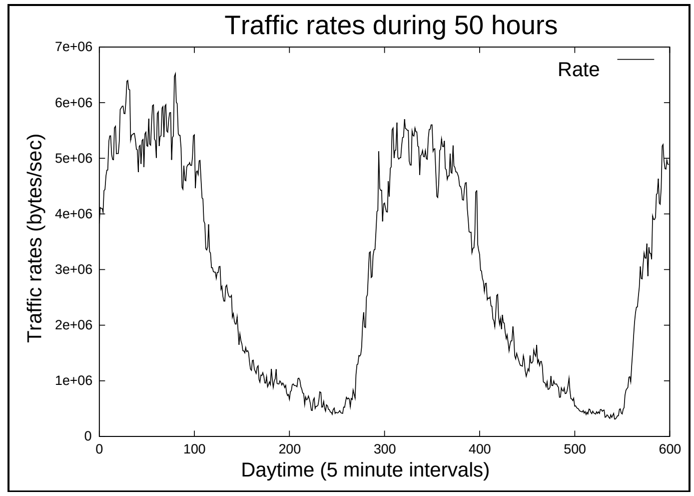

Figure 2: Traffic volume at different daytimes.

3 DELTA ESTIMATION

The problem of estimation can be addressed using extrapolation. A well-known extrapolation method is the polynomial extrapolation. The polynomial extrapolation uses a polynom of order $N-1$ to extrapolate from N measurement values. The N measured values uniquely define the polynom. For our purpose a linear extrapolation will do, so we use the uniquely defined line trough the last two measurements to extrapolate the next (expected) measurement. Figure 3 depicts the extrapolation. The extrapolation $L(i+1)$ of the next measurement (at x_{i+1}) is thus: $L(i+1) = y_i + \frac{(x_{i+1}-x_i)(y_i-y_{i-1})}{(x_i-x_{i-1})}$. For constant measurement intervals:

$$L(i+1) = y_i + \underbrace{(y_i - y_{i-1})}_{\Delta} = 2y_i - y_{i-1}$$

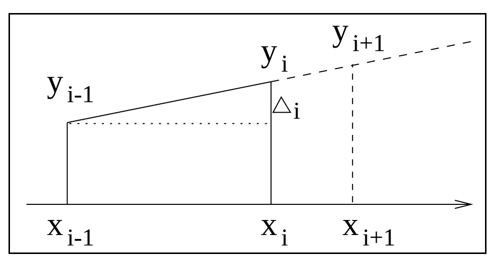

Figure 3: Linear extrapolation (polynom of order 1).

Using $L(i)$ as an estimation of the next expected measurement value is not a good idea. Such an estimation only bases on the last two measured values. We propose to *estimate the difference* using the exponential average. We denote t_i as the measured values and $d_i \equiv t_i - t_{i-1}$. We introduce the *delta estimation* function τ_Δ:

$$\tau_\Delta(n+1) = \begin{cases} 0 & n = 0 \\ t_n + \tau^{<d>}(n+1) & \text{otherwise.} \end{cases} \quad (7)$$

$\tau_\Delta^{<t>}$ extrapolates the estimate using an estimate of the next delta. Unlike simple extrapolation, all previously measured values influence the estimation (as long as $1 > \alpha > 0$). Note, that a possible extension to this schema is to estimate higher order differentials for the extrapolation with polynoms of higher orders. Fast extrapolation algorithms for such extrapolations exist [8].

3.1 Evaluation of the Delta Estimation

Considering its design it is obvious that the delta estimation will deliver good predictions for functions which are approximatively linear. Figure 4 shows an example of such a function. The graph shows the values of 30 measurement samples and their prediction using the exponential average estimation and the delta estimation. Since the measurement generating function consists of three linear functions, the delta estimation is doing very well. To measure the quality of the estimations we simply sum up the absolute value of the difference between each generated value and its prediction. Excluding the first prediction, the errors of the delta estimation estimator sum up to 6. The exponential average estimations produce an error sum of 38.

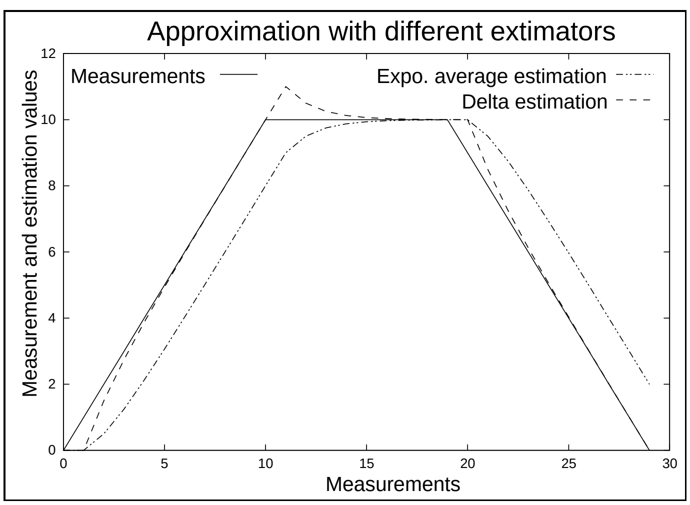

Figure 4: Estimation accuracy.

However, there are situations where the delta estimation works badly. If e.g the measurement values oscillate in the same period as the measurement intervals, the deltas change their signum for every measurement. Figure 5 shows the estimations ($\alpha = 1/8$) for measurements oscillating between the values 0 and 10. Since the exponential average puts more weight to the most recently measured value, the delta estimation predicts that the next value is larger than the current one if the previous value is smaller that the current value and vice versa. For this example the delta estimation prediction is always wrong the exponential average does better.

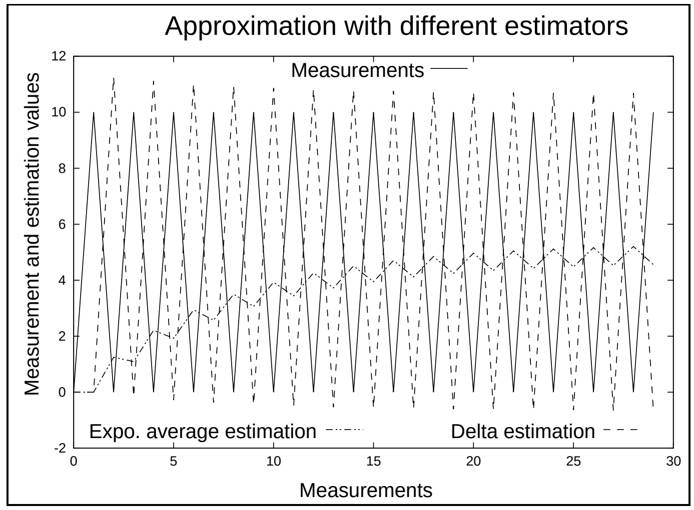

Figure 5: Estimation accuracy.

3.2 Improvement: Dynamic Hybrid Estimation

In general we don't know the function which generates the measurement values so it is not possible to predict if the exponential average or the delta estimation delivers a more accurate estimation. The delta estimation can exploit trends but the exponential average estimation is more stable in oscillating situations. However we can measure the accuracy of both estimations during the estimation process. Be $\epsilon_\tau(i) = |t_i - \tau(i)|$ and $\epsilon_{\tau_\Delta}(i) = |t_i - \tau_\Delta(i)|$ the absolute errors of the estimators for the i-th estimation. We define a_i to weight the accuracy of the delta estimation compared to the accuracy of the exponential average estimation:

$$a_i = \begin{cases} 0.5 & i = 0 \\ a_{i-1} & \epsilon_\tau(i) = \epsilon_{\tau_\Delta}(i) = 0 \\ \frac{\epsilon_\tau(i)}{\epsilon_{\tau_\Delta}(i) + \epsilon_\tau(i)} & \text{otherwise} \end{cases}$$

Note, that a_i is always between 0 and 1. The better the delta estimator predicts the i-th value (compared to the exponential average estimator) the closer a_i will be to 1. The a_i value is symmetric; if we swap the predictors in the previous formula, a_i will become $1 - a_i$.

We propose the dynamic hybrid estimation $\tau_\star^{<t>}(n+1)$:

$$\begin{aligned} \tau_\star^{<t>}(n) &= \gamma_n \tau_\Delta^{<t>}(n) + (1-\gamma_n)\tau^{<t>}(n) \\ \gamma_n &= \tau^{<a>}(n) \end{aligned} \quad (8)$$

$\tau_\star^{<t>}(n)$ uses a weighted average of the delta estimation and the exponential average estimation. The weight γ is dynamically determined by the previous performance of these two estimations. The γ is an exponential average estimation of the accuracy function a_i. Thus, if the measurements follow a global trend, the delta estimation will get more weight and so the dynamic hybrid estimation can exploit this trend. If the measurements fluctuate fast, the exponential average will get more weight and the dynamic hybrid estimation prediction will profit from the stability of the exponential average.

4 EVALUATION

4.1 Computational Complexity

The calculation effort of all three presented estimation functions is very small. The exponential average estimation uses 3 arithmetical operations per estimation, the delta estimation 4 and the dynamic hybrid estimation 16. This is a constant effort per estimation (independent of the number of measurements already taken).

4.2 Usage for Simulation Purposes

We replaced the exponential average estimation in the previously mentioned simulator for Differentiated Services with the delta estimation and the dynamic hybrid estimation respectively. We then compared the estimation performance. The simulation uses the estimations to predict upcoming traffic load for short term resource allocations. If the prediction is too low, there is a chance of traffic loss. If the prediction is too high there is an over-provisioning situation (not all available capacity is used). Good performance means little loss with little over-provisioning[2]. The results showed, that for instable traffic patterns, the delta estimation suffered from the oscillation problem shown in section 3.1. The dynamic hybrid estimation was immune to this problem. It performed similar to the exponential average estimation. This is obvious given the design of the dynamic hybrid estimation and the fact that the delta estimation performs badly. Nevertheless, in the start phase of the simulation, where the traffic builds up and reservations are set up based on traffic estimations both delta estimation and dynamic hybrid estimation perform better, since they are able to cope with the global trend (increasing traffic volume; starting from zero). Figure 4.2 illustrates these findings.

4.3 Usage on Real Measurements

We tested the estimators on real world data in order to compare their usefulness. The data set consists of 600 samples of

[2]Since loss is worse than over-provisioning, these simulations reserve for 20% more traffic than expected.

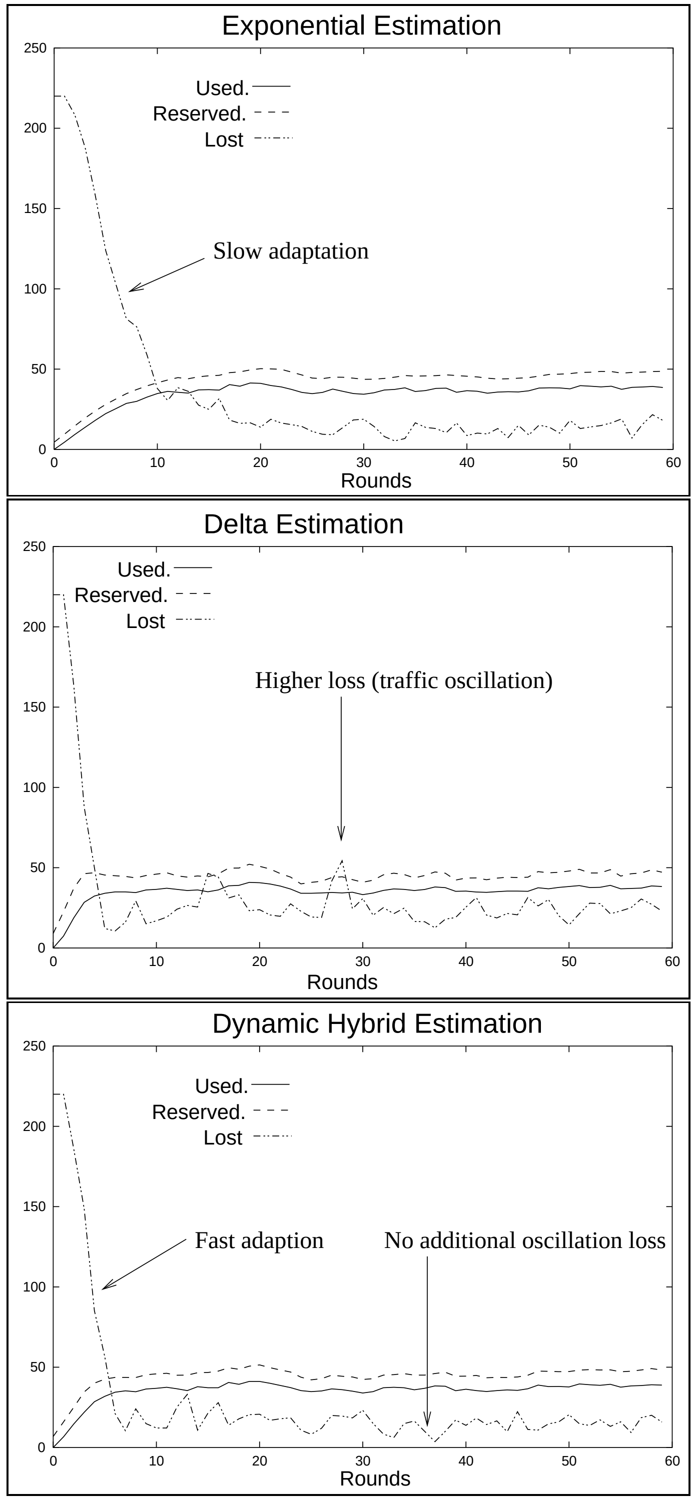

Figure 6: Adaptive DiffServ reservations and loss.

Table 1: Averaged errors on real world data.

$\alpha = 0.5$	Avrg. error per sample	Average relative error
exponential average estimation	171 KBps	6.2 %
delta estimation	184 KBps	6.7 %
dynamic hybrid estimation	145 KBps	5.2 %
$\alpha = 0.125$		
exponential average estimation	396 KBps	14.3 %
delta estimation	153 KBps	5.6 %
dynamic hybrid estimation	158 KBps	5.7 %

5-minute averages of Internet traffic entering the Swiss academic research network (SWITCH) over its trans-atlantic link [7]. The samples were already depicted in figure 2. Table 1 shows the errors averaged over the whole data set. The dynamic hybrid estimation has the best overall performance in these two test sets. For $\alpha = 0.5$ the delta estimation suffers from the oscillation problem shown in section 3.1. For $\alpha = 0.125$ the exponential average suffers from the monotony problem shown in section 2.

5 CONCLUSIONS

The classical exponential average estimation adapts only slowly to global trends. We therefore introduced the delta estimation which uses the exponential average estimation to estimate the difference between the current measurement value and the next value (to be estimated). However, the delta estimation is weak when the measurement traffic fluctuates heavily. The dynamic hybrid estimation uses a weighted average of both estimators. The weight is dynamically determined by estimating the accuracy of both internal estimators. While the dynamic hybrid estimation has the same (low) computational complexity as the exponential average estimation, it performs better in our simulations and on real network measurement data. The dynamic hybrid estimation has three parameters which could be optimised for data sets of a given type. The parameters are the α values of the exponential average estimators that are used within dynamic hybrid estimation. The delta estimation could also work with higher order polynomials (e.g. quadratics or cubics) to extrapolate the next value.

Acknowledgements. This work has been performed in the framework of the project Advanced Network and Agent Infrastructure for the Support of Federations of Workflow Trading Systems - ANAISOFT (SPP ICS ElCom Project 5003-057753) which is funded by the Swiss National Science Foundation (SNF). The authors also thank the people of the ApriQoS project of the national telecom provider Swisscom for their review of this work.

About the Authors. *Torsten Braun* got his Diploma Degree and Ph.D. degree from the University of Karlsruhe (Germany) in 1990 and 1993, respectively. From 1994 to 1995 he has been a guest scientist at INRIA Sophia-Antipolis (France). From 1995 to 1997 he has been working at the IBM European Networking Center Heidelberg (Germany), at the end as a project leader and senior consultant. Since 1998, he has been a full professor of Computer Science at the Institute of Computer Science and Applied Mathematics at the University of Bern (Switzerland), where he is heading the "Computer Networks and Distributed Systems" research group. He has been elected as a member of the SWITCH committee in 2000.

Manuel Günter got his Diploma Degree from the University of Berne (Switzerland) in 1998. He is now a research assistant and PhD student in the group of prof. T. Braun.

References

[1] M. Arlitt and T. Jin. “A Workload Characterization Study of the 1998 World Cup Web Site”. *IEEE Network*, 14(3):p.30–37, May/June 2000.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. “An Architecture for Differentiated Services”, 1998. RFC 2475.

[3] G. Dermler, M. Günter, T. Braun, and B. Stiller. “Towards a Scalable System for Per-Flow Charging in the Internet”. In B. Bodnar and A. Sharon, editors, *Proc. of the Applied Telecommunication Symposium (ATS 2000)*, April 2000.

[4] R. Edell and P. Varaiya. “Providing Internet Access: What We Learn from INDEX”. *IEEE Network*, September/October 1999.

[5] M. Günter and T. Braun. “Evaluation of Bandwidth Broker Signaling”. In *Proceedings of the International Conference on Network Protocols ICNP’99*, November 1999.

[6] V. Jacobsen. “Congestion Avoidance and Control”. In *Proc. SIGCOMM ’88*, pages 314–329. ACM, 1988.

[7] S. Leinen. “Switchlan Traffic Statistics”. http://www.switch.ch/lan/stat/, 2000.

[8] W. H. Press. *“Numerical Recipes in C : The Art of Scientific Computing”*. Cambridge University Press, 1988.

[9] A. Schlögl. “Time Series Analysis - A Toolbox for the Use with Matlab”. http://www.dpmi.tu-graz.ac.at/∼schloegl/matlab/tsa/, 1998.

[10] A. Silberschatz and P. Galvin. *“Operating System Concepts”*. Addison-Wesley, 5 edition, 1998.

[11] A. S. Tanenbaum. *“Computer Networks - Third Edition”*. Prentice-Hall International, Inc., 1996.