2nd POLISH-GERMAN TELETRAFFIC SYMPOSIUM PGTS 2002

9th Polish Teletraffic Symposium

TREND-BASED MODIFICATIONS OF THE EXPONENTIAL MOVING AVERAGE ALGORITHM FOR BANDWIDTH ESTIMATION

Lars Burgstahler, Martin Neubauer
Institute of Communication Networks and Computer Engineering,
University of Stuttgart, Pfaffenwaldring 47, 70569 Stuttgart, Germany
E-mail: burgstahler@ind.uni-stuttgart.de., neubauer@ind.uni-stuttgart.de

Abstract

In this paper, we describe different modifications of the Exponential Moving Average (EMA) algorithm that can be used for bandwidth estimation. Bandwidth estimation has to be deployed by nodes in connectionless networks that perform Quality of Service routing based on the available bandwidth. Since applications do not signal their bandwidth requirements in such networks, estimation is the nodes' only way to gain knowledge about the links' status. To use the estimation results for routing, we have identified several constraints that have to be fulfilled. We show how the modified algorithms behave with respect to these constraints and we will compare them with the basic EMA algorithm.

Keywords

Bandwidth estimation, Exponential Moving Average, QoS Routing

1. INTRODUCTION

Bandwidth estimation is essential when link occupancy should be used for QoS routing in connectionless networks without signalling capabilities. The nodes cannot know how much bandwidth is still available, and which link is therefore best suited for packet forwarding. To measure the traffic and to process these samples is the only way to get meaningful information. This is were bandwidth estimation algorithms come into action.

In connection-oriented networks, estimation algorithms are not so important for routing purposes. Applications running over ATM networks can easily reserve bandwidth using ATM's signalling capability. Thereby, the nodes always know the residual bandwidth on their attached links. Routing information is easily available and can be forwarded to other nodes and can be used for route computation.

Classical IP networks do not have any signalling mechanisms by which the applications can make reservations, since each packet is routed independently. RFC 2205 describes the Resource ReSerVation Protocol (RSVP) that can be used by applications to signal traffic characteristics to RSVP-capable IP routers [1]. In principle, RSVP could help the nodes to gain knowledge about link status. However, even if nodes knew about an application's bandwidth consumption via signalling they cannot predict which path a packet will take and they do not know the residual bandwidth of the attached links. As a solution to this problem, route pinning in conjunction with RSVP was proposed [2].

To get information on the link status in IP networks without the methods described above, estimation algorithms can be used. Each node measures the amount of traffic on its ports and knows the occupancy of the links. Using a routing protocol like OSPF [3] - possibly with the QoS extensions described in [4] - the information can be exchanged between the routers and be used for route calculations. The results of the estimation can be used to distinctively route certain flows, e.g. flows belonging to a higher prioritised DiffServ class, to keep adaptive routing tables to balance load equally in a network, or for connection admission control as described in [5].

In chapter 2 we give an overview on bandwidth estimation, including the exponential moving average algorithm. Chapter 3 explains our modifications on the algorithm and chapter 4 shows the impact of the modifications with the help of simulation results. In chapter 5 conclusions are made and future work is discussed.

2. BANDWIDTH ESTIMATION

In reality, links are always either completely occupied, i.e. a packet is currently being transmitted, or completely empty, i.e. no packet is currently being transmitted. However, this information does not help for routing. Instead, we need to calculate mean values over a certain interval.

Mean value calculation is not a single task but it consists of different subtasks. The first task is to measure the currently occupied bandwidth. Depending on the method, this leads either to an accurate but very volatile curve or a steadier, but not accurate one. Neither of them can be used well for routing. The second (optional) step is to smooth the measurement, i.e. levelling out short peaks. The third step is the real estimation, where we try to derive a bandwidth value which is a meaningful representation of the current occupied bandwidth. The next sections will contain a brief classification of methods that can be used for these three steps.

2.1 Measurement

We will distinguish between two different measurement methods:

•*Packet-based:* For each packet arrival, the time difference between the last packet arrival and the current time is calculated. The rate is defined as packet length over the respective time interval. This method is accurate. However, the main drawback is, that if no packet arrives for some time, the router does not get any new information on bandwidth occupancy and it is impossible to ever discover a completely empty link. To solve this problem, a timer can be used. If during the timer's interval no packet arrives, the calculation is triggered as described above.

•*Time-window-based:* For a fixed time window, packet lengths are summed up. The ratio of total data length and time gives the mean bandwidth over that interval. The time window will then be restarted. This method is stable and produces no direct problems, but the quality of the results depends heavily on the size of the time window.

2.2 Smoothing

Smoothing is a way to remove short peaks in a series of measurements. For the following, we have to remember, that if a series of measurement samples is given, a smoothing algorithm produces a new *series* of values which describes a smoother curve. The deviation from the original curve depends on the parameterisation of the algorithm.

2.3 Estimation

Estimation creates a *single* new value e_i which is based on a series of values. The simplest algorithm is the *arithmetic mean,* where n consecutive samples are summed up and divided by the number of samples. The main problem is to find the right size n for the set. If n is too big, real changes are levelled out, if n is too small, brief peaks are not suppressed.

For the *exponential moving average* in its simplest form (of order 1), only one old value has to be remembered. Still the results are satisfying. A new estimation e_i is calculated as follows:

$$e_i = (1-\alpha)e_{i-1} + \alpha b_i$$

The difficulty lies in the proper choice of α, the exponential weight. With a large α the estimation follows the measurement truly, but does not suppress short peaks, whereas with a small α peaks are suppressed but the estimation follows real changes too slowly.

In [6] an EMA algorithm with a dynamic weight is presented. The *distance adaptive EMA* algorithm uses the inter-arrival time between packets to modify the weight. However, we cannot use this approach, since in the following we will restrict ourselves to the time-window measurement where the time window and the distances are constant.

2.4 Influence of estimation on routing

Different bandwidth estimation algorithms have different characteristics, that make them better or worse suited to create input for a routing algorithm. In the following classification we show these characteristics and describe the desired behaviour of the algorithm.

•Reaction: The algorithm should quickly discover an increase or decrease in occupied bandwidth. It should only include a change into its result, when a long-term change is about to happen. Brief peaks should not influence the result. This leads often to the case, that the estimation lags behind the real trend, because current values are only moderately taken into account to avoid over-reaction.

•Stability: The algorithm should change the result of the estimation as rarely as possible to avoid updates for the routing. On the other hand, the results should still reflect reality as close as possible (which leads to inevitable changes).

•Symmetry: Some algorithms are fast in discovering an increase and slow in discovering a decrease in occupied bandwidth. Others behave vice-versa, they tend to underestimate the occupied bandwidth. The first group wastes resources, the second one might lead to an overloaded network.

•Convergence: No matter how fast the algorithm is, it should eventually reach the true current mean value of the occupied bandwidth. Some algorithms tend to overshoot or never reach the mean at all.

•Cost: Costs can be expressed in various terms: computational complexity, memory, and time for calculation. The optimum is a simple, fast algorithm that need not store much data.

Note that although the first two characteristics are the most important, each of them has to satisfy contradictory requirements. The goal is to find the most effective compromise, while achieving also good results for the three other characteristics.

3. MODIFICATIONS OF THE EMA ALGORITHM

The basic EMA algorithm shown above is statically configured, i.e. that α is possibly optimised for special traffic conditions only. The composition of traffic cannot always be predicted which complicates the selection of a suitable weight. Therefore, it is desirable that the estimation algorithm adapts to the traffic: reacting fast on real load changes, especially under heavy load conditions, but staying on an average level when brief oscillations happen.

3.1 Low pass EMA algorithm

Our basic assumption is, that sharp increases or decreases can first be treated as peaks. If the change persists, it should be taken into account but then very quickly. The goal is to consider these changes with lower weight. Smaller changes or stagnancy indicate a stable trend, these measurements can be considered with a higher weight. As a consequence the main problem is to distinguish different situations, and to identify the correct reaction. The problem is shown in Figure 1.

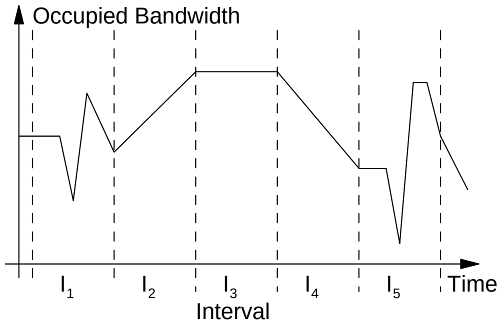

Figure 1 : Short peaks and long term changes

During intervals 1 and 5 the changes are brief and only temporary, caused by variations in single flows. In contrast, changes in interval 2 and 4 are more permanent, caused by flows beginning or ending respectively. The difference between the two situations is clear: although the changes are approx. equally large, the duration of them differs significantly. Based on these considerations, we use the gradient of occupied bandwidth for the generation of a dynamic weight.

Looking at times t_1 and t_2, in Figure 2, we see that the occupied bandwidth is identical, but the occupancy trend is different which can be seen from the gradient of the curve. A good estimation not only depends on the currently occupied bandwidth, but also on the change of the occupied bandwidth over time. To enable this, we can take into account the gradient m_i between two points t_{i-1} and t_i:

The *low pass EMA (LpEMA) algorithm* uses this gradient to modify the weight of a 1st order EMA algorithm. The weight is calculated with the help of a low pass filter of 1st order. Using the formula for a low pass filter we replace the frequency f by the gradient m_i and the limit f_g by a normalizing gradient m_{norm}. To control the maximum adaptation, we introduce a maximum weight α_{max}. This leads to the following equation for the weight α_i:

$$\alpha_i = \alpha_{\max} \frac{1}{1 + |m_i| / m_{norm}}$$

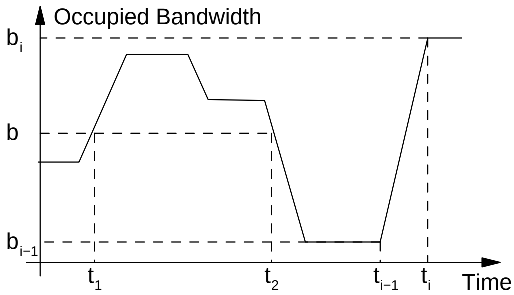

Figure 2 : Change of occupied bandwidth

The larger the gradient is, compared to m_{norm}, the smaller the fraction becomes, and the smaller α_i becomes. The result is, that short peaks are hardly noticed. In case that the absolute of the current gradient $|m_i|$ is equal to the normative gradient m_{norm}, we get

$$\alpha_i = 0.5\alpha_{max}$$

This means, that for comparable investigations between this modification and the classical EMA, the weight has to be doubled. Further thoughts on the selection of α_{max} can be found in [7], where we also show how to obtain values for m_{norm}. In this paper, we will restrict ourselves to the so-called traffic-dependent mean, that represents the mean gradient based on observation of the actual traffic.

3.3 Retrospective EMA algorithm

In section 2.2 we have shown that we can already smooth the curve before making the estimation. We will explain here what problems occur when the estimation is made with an EMA algorithm. For reasons of simplicity, the EMA algorithm used here does not have a dynamic component.

Figure 3 depicts how an estimation based on the smoothed curve (solid line) is calculated. t_i denotes the times at which samples are taken, m_i denotes the measured sample, s_i denotes the smoothed value of the sample, and e_i is the estimation. So in the first graph we get

$$e_3 = (1-\alpha)e_2 + \alpha s_3$$

In the next graph, at t_4 the same calculation is done again. However, we can see that due to smoothing, s_3 has changed to s_3'. From the current point of view, it seems that using s_3 as a base for the estimation of e_3 was not optimal. Thus using e_3 as a base for the calculation of e_4 is also not optimal.

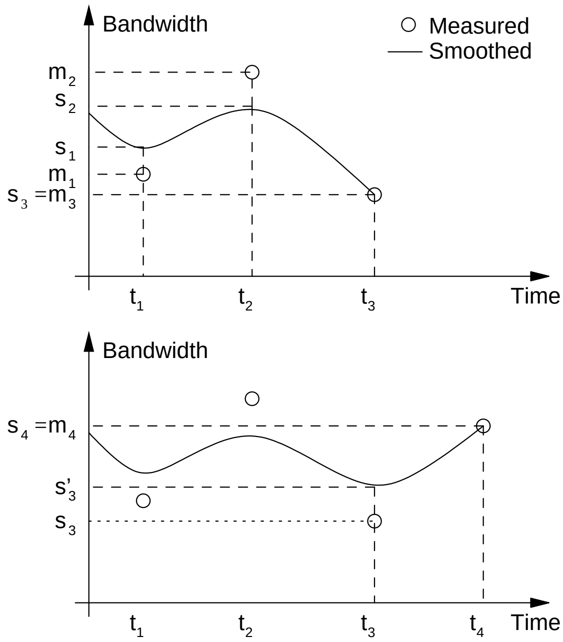

Figure 3 : Retrospective change of smoothed value

The solution is "to go back in time" and recalculate e_3. However, the smoothing algorithm might not only have changed s_3, but also older values of s. This means that e_2 and other estimations before are also not optimal. Actually recalculation has to start at the last value that is not affected by smoothing anymore, i.e. that has already reached a steady state.

In general terms, to get e_i at time t_i with a smoothing window size of n, we have to rely on value e_{i-n}, the last trustworthy estimation. Then - step by step - all estimations $e_{i-n+1}...e_i$ have to be recalculated by use of the EMA algorithm.

As a result, we get an estimation, that is based on values that are smoothed by future information (from the point of view of these values). Of course, this approach cannot predict the future for the current estimation.

4 SIMULATION RESULTS

In this section, we will present simulation results, that show the behaviour of the proposed variations. We have used Internet traffic to evaluate the behaviour of our modifications. Further simulations were made using IP telephony traffic, the results can be found in [7]

The traffic consists of Pareto distribution-sized blocks that were segmented into IP packets (*MaxSize*=1500Byte). The parameters for the Pareto are α=1.6 (*H*=0.7) and *MinBlockSize*=3750. The generators produce on-off traffic, offering 70% of load to the network, a 10Mbit/s link.

The parameters for the different algorithms are:

- Simple EMA: The base weight is set to α=0.3.
- Low pass EMA: The base weight is set to α_b=0.6.
- Retrospective EMA: The base weight is set to α=0.3. For the smoothed and the retrospective EMA algorithm, the penalized least squares algorithm was used before doing the estimation. The parameters are: *NumberOfMeasurements*=5, *Smoothness*=10000, *DataWeight*=1, and *DifferenceDegree*=1.

For the simulations, we have chosen the time-window measurement approach with a jumping window with a size of 1s. Routing protocols do generally not need any finer granularity. The result of the time-window measurement is always shown as reference. Further the results of a comparable simple EMA algorithm are always shown as a dashed line.

We will rate the result of the algorithms by visually comparing the estimation curve with the original measurement to see if the defined requirements are met (see 2.4). This is because so far, we have not identified a good method to make an analytical comparison. Methods like the *root-mean-square (RMS)* are not suitable since

1. if our estimation does not follow a short peak, we get a large deviation which is desired,
2. if our estimation does not follow a long-term trend, we get a large deviation which is not desired.

Hence, the difference between the measurement and the estimation does not directly express the quality of the estimation. Only in the context of the current trend, we can make a statement.

The low pass EMA shows good results (Figure 4). It recognizes the fully loaded link faster than the normal EMA. Not only does it react slightly faster, but it also converges much faster. The traffic-dependant alternative is a little bit smoother. The link-dependent alternative is rarely faster although it uses a much higher normalizing gradient. On the other hand, it shows a rougher curve, following the measurement much closer.

The results of the retrospective EMA are shown in Figure 5. It is a little bit slower than the simple EMA, but it converges earlier, when the bandwidth remains constant for some time. Besides, it provides a better envelope curve when the trend remains for some time. Whereas the smoothing generally overestimates the occupancy during decreases, the retrospective EMA is more optimistic without being overly optimistic. The overall behaviour is very similar to the other two scenarios.

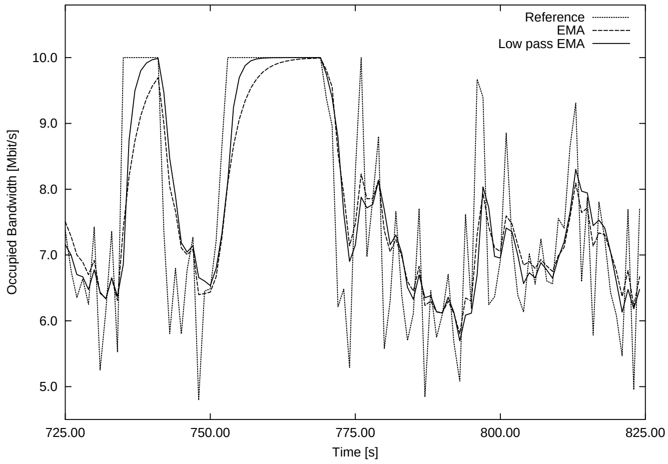

Figure 4 : Low pass EMA - Internet traffic

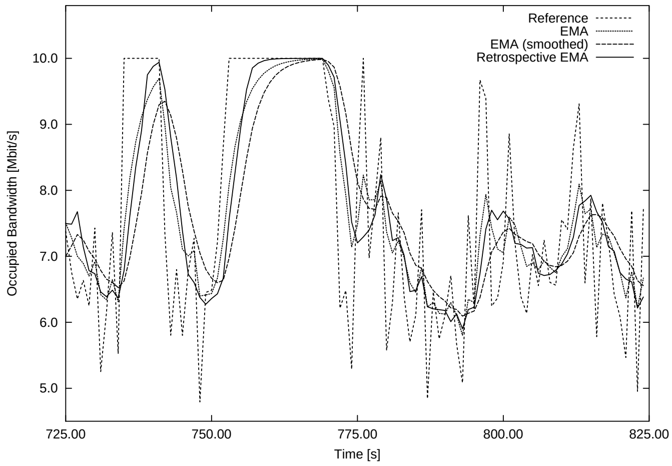

Figure 5 : Retrospective EMA - Internet traffic

5 CONCLUSION

In this paper, we have presented and evaluated by simulation two modifications of the exponential moving average algorithm. For the first, the weight was dynamically calculated to adapt to different load situations and to filter out short-term effects. For the second, a smoothing algorithm was and the simple

EMA algorithm was adapted to get the maximum out of the smoothed results, by modifying the history.

The simulations showed that the low pass EMA algorithm works a little bit better than the simple EMA algorithm. Whether this algorithm could perform even better with different parameters has yet to be investigated. The retrospective EMA algorithm showed a good performance. The curve runs very smooth, levelling out minor peaks. At the same time, it reacts at least as fast as the simple EMA algorithm but converges faster. Using a small series of samples to smooth the curve, the computational overhead is sufficiently small.

In our further work, we will examine if the combination of retrospective behaviour and dynamic adaptation of the weight could bring further enhancement. During the evaluation of our modifications, several other ideas for adaptive estimation algorithms were developed, that have to be analysed. Eventually, we like to investigate the effects of bandwidth estimation on routing decisions.

6. BIBLIOGRAPHY

[1] Zhang, L., Berson, S., Herzog, S., Jamin, S., Braden, R.: *Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification*, RFC 2205, IETF, September 1997.

[2] Kamat, S., Guerin, R., Herzog, S.: *QoS Path Management with RSVP*, Proceedings of 2nd Global Internet Miniconference, Phoenix, AZ, November 1989.

[3] Moy, J.: *OSPF Version 2*, RFC 2328, IETF, April 1998.

[4] Williams, D., Kamat, S., Guerin, R., Orda, A., Przygienda, T., Apostolopou-los, G.: *QoS Routing Mechanisms and OSPF Extensions*, RFC 2676, IETF, Aug. 1999.

[5] Shenker, S., Danzig, P., Jamin, S.: *Comparison of Measurement-based Admission Control Algorithms for Controlled-Load Service* Proceedings of IEEE INFOCOM '97, pp. 973-980, Kobe, Japan, April 1997.

[6] LeBoudec, J.-Y., Ferrari, T., Almesberger, W.: *Scalable resource reservation for the Internet*, Proceedings of IEEE Conference, Protocols for Multimedia Systems -- Multimedia Networking (PROMSMmNet), Santiago, Chile, November 1997.

[7] Burgstahler, L., Neubauer, M.: *New Modifications of the Exponential Moving Average for Bandwidth Estimation*, Proceedings of the 15th ITC Specialist Seminar on Internet Traffic Engineering and Traffic Managment, Würzburg, Germany, July 2002