Continuous Monitoring of Available Bandwidth over a Network Path

Svante Ekelin
Ericsson Research
and Mälardalen University

Martin Nilsson
Swedish Institute of Computer Science (SICS)
and Mälardalen University

Abstract—**This paper presents a new method for real-time monitoring of available bandwidth over a network path. The method uses active probing with trains of probe packets, and produces estimates in real-time using Kalman filtering. It improves the estimate for each new measurement of the *strain* of the inter-packet time intervals in a probe packet train. We have tested the method with good accuracy and agreement, both in simulation and in a physical test network. The method requires no communication from receiver to sender and only minor processing and memory space.**

Keywords: **available bandwidth, Kalman filter, active probing, packet train, estimate.**

I. INTRODUCTION

A. Overview

The study of available bandwidth is enabled by active probing of the network path. By observing and analysing effects of cross traffic on injected probe traffic, we may draw conclusions about the available bandwidth.

In this paper, we present a new active probing method for available bandwidth estimation: BART (Bandwidth Available in Real-Time).

We introduce the concept *inter-packet separation strain* for the relative increase in the time separation of two consecutive probe packets. This provides a convenient measure of congestion. When there is no congestion, this strain is zero on average. When the probe load becomes larger than the available bandwidth, the strain grows proportionally to the overload.

The main novelty is the technique for applying a Kalman filter for estimating the available bandwidth. We have results from simulations (not presented here) and laboratory testing, showing the viability of the method.

Some of the features of BART are: estimates are produced quickly; accuracy can be traded for agility; an estimate of the variance for the estimated available bandwidth is produced; tuning is largely automatic, i.e. few parameters need manual adjustment; no communication is required from the receiver of probe trains to the sender. Further, it has low requirements for processing power and memory use, so the algorithm could be implemented in a small microprocessor.

B. Related work

A number of papers have appeared in recent years in the field of available bandwidth measurements.

Jain and Dovrolis developed the method Pathload [3], and Melander et al. the method TOPP [1]. Both methods use probe-packet trains to estimate the point of congestion. TOPP fits data to a straight line, whereas Pathload looks for an increasing trend in the one-way delay. Both methods require a substantial time for probing and analysis before producing an estimate.

Ribeiro et al. developed the method pathChirp [4], which uses probe trains with internally varying inter-packet separations, in order to scan a range of probe rates with each train.

Keshav discussed the application of Kalman filters for traffic measurements in an early paper [7], but concluded that this was unlikely to be a suitable method. However, this conclusion was based on the assumption that routers operate with bandwidth reservation. This is typically not the case in today's Internet.

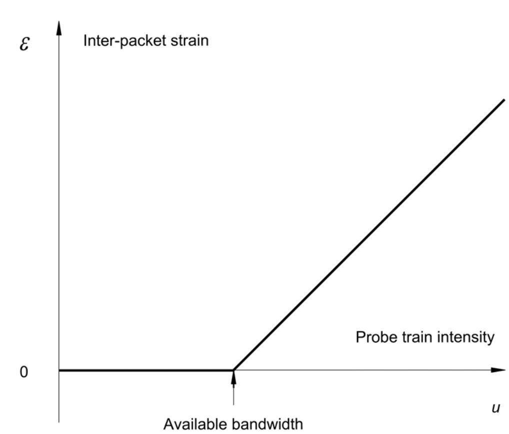

Fig. 1: The relation between available bandwidth, probe intensity and inter-packet strain.

A Kalman filter has previously been applied to some other types of traffic measurements, but not to estimation of available bandwidth. Jacobsson et al. used a Kalman filter for RTT estimation from the perspective of TCP [5]. Kim and Noble did this for estimating "bandwidth" over a path [6]. However, their definition of "bandwidth" bears no resemblance to the definition used by us and others in this field [1,2,3].

Similarly to the TOPP method, we use the fact that the strain is proportional to the overload. Thus, there is more information to be gained from each measurement than merely whether the value indicates overload.

Our original contribution is a technique for applying a Kalman filter, continuously updating the available bandwidth estimate for each new measurement (i.e. each new probe train). This leads to a simple, fast, and easy-to-implement method.

C. *Paper organization*

In the next section of this paper, we describe the problem of measuring available bandwidth. We introduce terminology, define the basic concepts, and describe our network model. The third section describes our proposed method for measuring bandwidth available in real-time. The fourth section presents results from experiments in a laboratory network. The fifth section discusses the results, as well as some limitations and possible variations of the algorithm, before the paper is concluded in section six.

II. Measuring Available Bandwidth

While several other network-path dependent traffic properties can be measured directly, such as packet delay and loss probability, measuring the available bandwidth is a greater challenge. If we don't have access to statistics from all network nodes along a path, the only way to measure available bandwidth is by active probing.

A. *Definition of Available Bandwidth*

In the literature, the term "available bandwidth" has been used in different ways. We want to make clear what we denote by this.

Each link j in a network path has a certain capacity, C_j, determined by the network interfaces in the nodes on each end of the link. This is simply the highest possible bit rate over the link at the packet level.

The capacity typically does not vary over short time scales. What varies is the load, or cross traffic, on the link, $S_j(t)$. The available bandwidth $B_j(t)$ of link j is

$$B_j(t) = C_j - S_j(t) \tag{2-1}$$

One of the links along the path has the smallest available bandwidth. This "bottleneck link" determines the available bandwidth of the path. For a network path, the available bandwidth is defined as the minimum of the available link bandwidths along the path:

$$B(t) = \min_j (C_j - S_j(t)) \tag{2-2}$$

This is in line with what is denoted by available bandwidth in [1,2,3].

An interpretation of the available bandwidth $B(t)$ is the theoretical maximum at time t of the increase in data rate from sender to receiver before causing a congestion. This interpretation is closely related to the way we measure the available bandwidth, namely by sending probe traffic, and determining the threshold rate when the probe traffic transiently experiences congestion. In fact, one might argue that the rate thus measured *defines* the available bandwidth.

B. *Network Model*

We model a network path as a series of hops, where each hop consists of an input FIFO queue and a transmission link. Each link has constant capacity and time-varying cross traffic.

Consider a fluid traffic model of a single FIFO hop with link capacity c and cross traffic s. The available hop bandwidth is then $B = c - s$. If the hop is also loaded by probe traffic u, and $u \le B$, there is no congestion, but if $u > B$, we have an overload situation. If r is the rate of the probe traffic exiting the hop, it was shown in [1] that

$$\frac{u}{r} = \begin{cases} 1 & (u \le B) \\ \frac{1}{c}u + \frac{s}{c} & (u > B) \end{cases} \tag{2-3}$$

By varying u and identifying the value where the demanded to received probe traffic rate u/r starts to deviate from unity, one can estimate the available bandwidth of the hop. It was also shown in [1] that by doing this over a path, the available bandwidth of the bottleneck link and thus for the whole path is obtained.

Next, we adapt this reasoning to take into account the fact that traffic is composed of discrete packets. We use the generic multiple-hop model presented in [2], and the notation therein.

Consider a sequence of packets arriving at a hop at times τ_i and arriving at the next hop at times τ_i^*. We are interested in the packet inter-arrival times :

$$t_i = \tau_i - \tau_{i-1} \quad , \quad t_i^* = \tau_i^* - \tau_{i-1}^* \ . \tag{2-4}$$

Now, we introduce the dimensionless quantity inter-packet *strain* ε_i, given by the ratio

$$\frac{t_i^*}{t_i} = 1 + \varepsilon_i \tag{2-5}$$

This strain provides a direct analogy in the discrete description to the fluid model ratio of demanded to received traffic rate discussed above. It is easy to see that at the time resolution of the separation of individual packets of size b we have

$$\frac{u}{r} = \frac{b / t_i}{b / t_i^*} = \frac{t_i^*}{t_i} = 1 + \varepsilon_i \tag{2-6}$$

We assume that, as in the fluid model, for the average inter-packet strain of consecutive pairs in a probe train of given rate u, any systematic deviation from zero is dominated by the effect of cross traffic interaction in the FIFO queue at the bottleneck link.

C. *The measurement problem*

Input data for our BART estimation algorithm is generated in a similar way as in the TOPP method [1]. Instead of just sending probe pairs, we send trains of probes. This way, we reduce variance and improve the statistical precision in our measurements.

Trains of $N+1$ probe packets (where $N \geq 2$) are sent from the sender to the receiver. Each probe packet is timestamped on sending, and the receiver calculates the inter-packet strain. When the train is received, the average inter-packet strain ε and its variance R are computed from the N values.

The problem is now how to compute an estimate of the available bandwidth B using ε and R. While TOPP uses linear regression and fits a curve to a large number of points offline, BART keeps an estimate, and updates it every time a new packet train is received, applying a Kalman filter using a trick to handle the non-linearity. BART is described in detail in the following section.

III. Estimation by Filtering

A. *Dealing with the non-linearity*

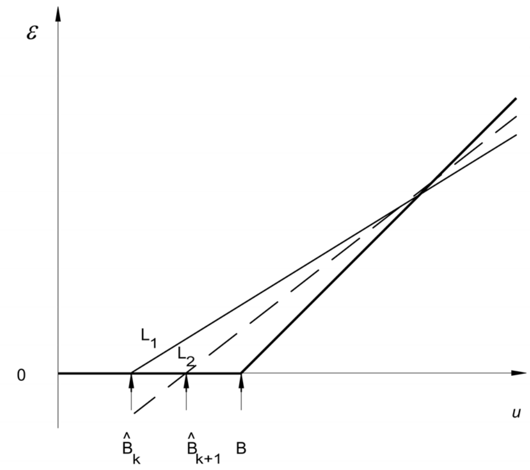

Fig. 2: Convergence of the BART method.

Our inter-packet strain measurements are described by a piecewise linear model (cf. fig. 1, eqs. (2-3), (2-6), and [1])

$$\varepsilon = v + \begin{cases} 0 & (u \leq B) \\ \alpha u + \beta & (u > B) \end{cases} \tag{3-1}$$

where u is the probe traffic rate, v is the measurement noise, and $\alpha = 1/c$ and $\beta = s/c - 1$ are state variables. An estimate of the available bandwidth $B = c - s$ can be computed from the Kalman estimates of α and β:

$$\hat{B} = -\frac{\hat{\beta}}{\hat{\alpha}} \tag{3-2}$$

However, the model above is only piecewise linear. The cusp at $u = B$ prevents direct application of a "standard" Kalman filter. The trick here is to only apply the filter to measurements for which $u > \hat{B}$, where $\hat{B}$ is the current estimate of the available bandwidth.

This can be understood as follows: Given only points for which $u > \hat{B}_k$, the Kalman filter attempts to find a straight-line approximation L_1 to the concave curve $\varepsilon(u)$ [fig. 2]. The intersection with the u-axis approximates the true available bandwidth B.

Suppose that we have an underestimate $\hat{B}_k < B$. The next time the Kalman filter is applied, we use only points for which $u > \hat{B}_k$, and the Kalman filter will attempt to find a new line L_2. Due to $\varepsilon(u)$ being concave, this line will tend to intersect the u-axis at a point $\hat{B}_{k+1}$, where $\hat{B}_k < \hat{B}_{k+1} < B$, showing that the new value in the mean will be an improved approximation.

B. The BART algorithm

The BART algorithm steps are:

1. The receiver initializes the estimates of the state vector $[\alpha \quad \beta]^T$ and the covariance matrix P for the state vector estimate.
2. The sender starts generating probe trains with traffic rate u. For each probe train, a new value of u is randomly generated. The sender passes the rate along inside the probe packets.
3. For each probe train, the receiver recovers u, and computes the average strain ε and its variance R.
4. The receiver inputs these values to the Kalman filter only if $u > \hat{B}$. It updates the state vector and the matrix P. The receiver also provides the filter with an estimate of the matrix Q (cf. below).
5. The receiver uses the updated state vector to compute a new available bandwidth estimate $\hat{B}$. The cycle repeats from step 3.

Here, Q and R are the covariances of the process noise and the measurement noise, respectively. Q is a 2×2 -matrix, and R is a scalar.

One difficulty when applying Kalman filters is the problem of finding a suitable Q. This matrix decribes fluctuations in the system, and can in some sense be considered to be part of the network model.

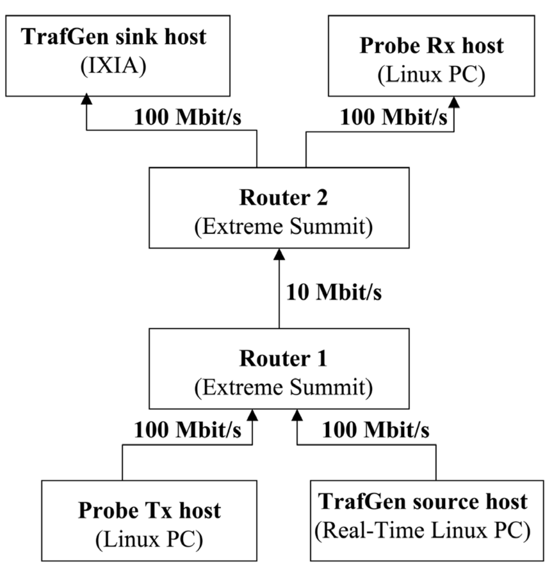

Fig. 3: Test network configuration.

One approach is to relate Q to the statistical properties of the cross traffic. Another approach is to regard Q as a tunable parameter. Large values of Q will result in low accuracy, but high agility of the filter. Smaller values of Q result in higher accuracy, but slower step response.

IV. RESULTS

A. Test network configuration

We carried out initial tests of a pilot implementation of the method in a laboratory network [Fig. 3], using controlled cross traffic in order to verify that the measurement method produces reasonable results.

Cross traffic was generated on a Real-time Linux PC, and sent to an IXIA traffic analyser. Probe traffic was generated on one Linux PC and sent to another one. Both traffic types have three hops along their paths, but it is only in the second hop there is interaction. All the hops have 100 Mbit/s capacity, except the router – router hop, which was configured by the router management interface to have a capacity of 10 Mbit/s.

The testbed is a slight modification of the one previously used for evaluating Pathload [9,10] and for analysing cross traffic effects on probe trains [11].

B. Test runs

We measured the actual capacity of the tight link as experienced by real traffic by generating cross traffic at a rate of 20 Mbit/s with no competing probe traffic, and then counting received bytes with the IXIA. The resulting received rate was measured to be 9.20 Mbit/s. This was used as the true value of the capacity of the bottleneck link.

We used a packet size of 1500 bytes both for the cross traffic and for the probe traffic. Each probe train consisted of $N + 1 = 17$ packets, and the measurement period (the inter-train separation) was 1 second.

The cross traffic in the test runs was generated according to a Poisson process. The observed strain for 5 Mbit/s cross traffic is shown in fig. 4. In fig. 5, the estimate of the available bandwidth is shown, when the cross traffic has a step increase of the average rate from 5 to 7.5 Mbit/s at the 1000th probe train.

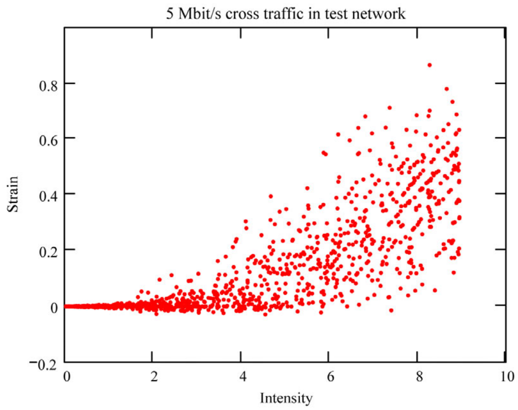

Fig. 4: Inter-packet strains for the test network.

V. Discussion

The current method appears to work well on the test cases we tried. One might ask what happens for traffic with less nice statistical characteristics, such as tail-heavy probability distributions with infinite variance [12]. Even if traffic is non-Gaussian, communication links have finite capacity, so all stochastic variables entering our filter calculations are bounded and consequently must have bounded variance, eliminating at least the problem of tail-heavy distributions.

Kalman filtering is not optimal for non-Gaussian distributions, but is known to behave well in general. In the future, we are going to study how the method handles traffic with more challenging distributions.

One way to improve the step response could be to extend BART to use adaptive filter techniques [5].

Since we use active probing, it is important to minimize the disturbance experienced by ordinary traffic due to probe traffic. For our method, it seems sufficient to inject probe traffic at an average level of less than 2% of the bottleneck link capacity.

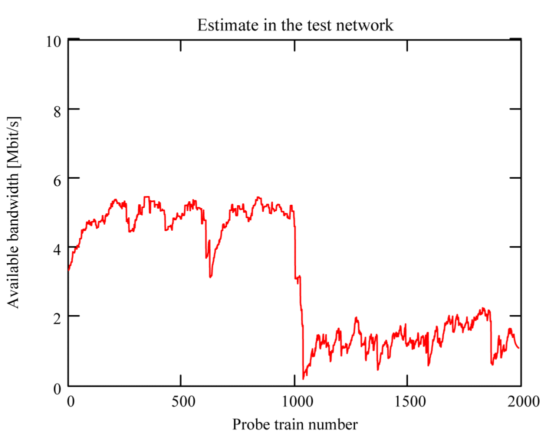

Fig. 5: Step response for 5 to 7.5 Mbit/s increase in cross traffic in the test network.

VI. Conclusions

We have demonstrated that Kalman filtering is applicable to measurement of available bandwidth, despite the system being strongly nonlinear. We have also shown that reasonable accuracy can be obtained with little computational efforts. We conclude that Kalman filters seem to be a good candidate for fast estimation of available bandwidth in the future.

VII. Acknowledgments

The authors are grateful to Andreas Johnsson, Bob Melander and Mats Björkman from the TOPP team for allowing us to use their probe generation and recording software, and for valuable discussions.

VIII. References

[1] B. Melander, M. Björkman, and P. Gunningberg, ”A new end-to-end probing and analysis method for estimating bandwidth bottleecks”, in *Proceedings of IEEE Globecomm '00*, San Francisco, USA, November 2000.

[2] A. Pásztor and D. Veitch, “The packet size dependence of packet-pair like methods”, in *Proceedings, Tenth International Workshop on Quality of Service (IWQoS 2002)*, Miami Beach, USA, May 2002.

[3] M. Jain and C. Dovrolis, “Pathload: a measurement tool for end-to-end available bandwidth”, in *Passive and Active Measurements (PAM) Workshop 2002 Proceedings*, pp 14-25, Ft Collins, USA, March 2002.

[4] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, L. Cottrell, “pathChirp: efficient available bandwidth estimation for network paths”, in *Passive and Active Measurements (PAM) Workshop 2003 Proceedings.*

[5] K. Jacobsson, H. Hjalmarsson, N. Möller and K H Johansson, “Estimation of RTT and bandwidth for congestion control applications in communication networks”, in *IEEE Conference on Decision and Control (CDC) 2004 Proceedings*, Paradise Island, Bahamas, 2004.

[6] M. Kim and B. Noble, “Mobile network estimation”, Mobile Computing and Networking (ACM MOBICOM), Rome, Italy, 2001.

[7] S. Keshav, A control-theoretic approach to flow control. In Proceedings of ACM SIGCOMM'91, pages 3--15, Zurich, Switzerland, September 1991.

[8] G. Bishop, and G. Welch, “An Introduction to the Kalman Filter,” TR 95-041, Department of Computer Science**,** University of North Carolina at Chapel Hill, NC, USA. 2004.

[9] A. Welin, S. Ekelin and R. Holm, “Study of active measurement methods using a testbed environment”, in *High Speed Networking 2003 Spring Proceedings*, Budapest, Hungary, May 2003.

[10] R. Holm, “Probe-based load measurement”, Master thesis, Lund Technical University, April 2003.

[11] A. Johnsson, B. Melander and M. Björkman, “On the analysis of packet train probing schemes*”, International Conference on Communications in Computing (CIC) 2004*, Las Vegas, USA, June 2004.

[12] V. Paxson and S. Floyd, ”*Wide-area traffic: The failure of Poisson modeling*,” IEEE/ACM Trans. on Networking, vol. 3, pp. 226--244, 1995.