

Online Estimation of Available Bandwidth and Fair Share Using Kalman Filtering

Zdravko Bozakov and Michael Bredel

Institute for Communications Systems, Leibniz Universität Hannover, Germany
{zdravko.bozakov,michael.bredel}@ikt.uni-hannover.de

Abstract. In this paper we address the problem of online bandwidth estimation in wired and wireless LANs. To this end, we employ active probing, i.e. we continuously inject packet probes into the network. We present the key challenges and analyze the trade-offs between fast change detection and estimate smoothness. We show the benefit of using Kalman filtering to obtain optimal estimates under certain conditions and provide a procedure for parameterizing the filter with respect to specific use cases. Furthermore, we evaluate the influence of probing train length on the results. Based on our findings we developed a tool implementing the presented methodology. We support our theoretical results by a number of real-world measurements as well as simulations.

Keywords: measurements, online bandwidth estimation, fair share, kalman filter.

1 Introduction

The idea of using end-to-end measurements to infer the capacity left over by cross traffic, also called available bandwidth, dates back to TCP congestion control [1] and packet pair probing [2]. In the field of wired networks a number of estimation methods, e.g. [3–7] exist that are well understood. The task of bandwidth estimation in wireless LANs, poses additional challenges [8, 9]. Nevertheless, some tools originating from the wired domain have been suggested for available bandwidth estimation in wireless networks [7, 9, 10]. Empirical evaluations for both domains can be found e.g. in [8, 11].

Active probing methods inject synthetic traffic into the network and attempt to infer the unused resources of the network path by analysing the packet dispersion, i.e. the changes in inter-packet gaps. Almost all tools calculate the average of several measurements and report a single bandwidth estimate. Under constant channel conditions, increasing probing intervals or the probing traffic intensity can improve the accuracy of the results. On the other hand, it is often necessary to probe bandwidth continuously in order to accurately detect changes over time. For minimal intrusiveness it is desirable to minimize probing traffic, which might negatively impact the sample accuracy. At the same time, the sampling rate must be high enough to guarantee that changes are detected sufficiently fast.

In this paper we analyze the constraints related to continuous probing and show how Kalman filtering can be used to improve estimates. We model the available bandwidth and the fair bandwidth share in wireless networks respectively as a time-varying process that is at least piecewise stationary. In wired networks this process is cross traffic dependent while in wireless domains it is also related to the number of nodes and the physical channel conditions. Using trains of packet probes we sample this process continuously to adapt to changes in the channel state, e.g. changing number of transmitting stations, rate adapting stations or cross traffic variations. Additionally, we assume that the maximum probing rate is limited. We show that the estimate variance, which we interpret as *measurement noise*, is related to the train length. Furthermore, we present a procedure for tuning the *process noise* which influences smoothness and agility of the Kalman estimate. We substantiate our analytical findings by an extensive set of measurements and simulations. Furthermore, we address the problem of finding a suitable number of packets per probing stream.

2 Related Work

In this section we discuss related work on bandwidth estimation in wireless and wired networks. We focus on active probing techniques rather than passive approaches which analyze cross traffic directly. In wireless networks for instance, the broadcast medium can be used to capture packets of other stations and derive bandwidth information. However, these methods cannot obtain end-to-end information.

Available bandwidth estimation techniques attempt to infer the capacity left over by cross-traffic in a network path. Hence, the available bandwidth AB_i of a single link i in a time interval $[\tau, t]$ is defined as [12]

$$AB_i(\tau, t) = C_i(1 - u_i(\tau, t)) \quad (1)$$

where C_i is the channel capacity and $u_i \in [0, 1]$ the utilization. The available bandwidth of the network path is given by the minimum available bandwidth over all single links $AB = \min_i(AB_i)$.

In recent years a number of tools, which use active probing for available bandwidth estimation have been proposed, e.g. [3, 4, 10]. Probes consisting of packet pairs or packet trains are injected into the network at specified rate. The inter packet gaps at the path egress are then used to infer the available bandwidth. So-called packet chirps [4] represent a special class where packets are sent with a geometrically decreasing gap.

Most tools assume First In First Out (FIFO) multiplexing which is the typical scheduling policy in today's Internet. Thus, network flows obtain a share of the capacity which is proportional to their sending rate. Under the assumption of constant bit rate (CBR) probing rates this yields

$$\frac{r_{in}}{r_{out}} = \max \left(1, \frac{r_{in} + \lambda}{C} \right) = \begin{cases} 1 & , \text{ if } r_{in} \leq C - \lambda \\ \frac{r_{in} + \lambda}{C} & , \text{ if } r_{in} > C - \lambda \end{cases} \quad (2)$$

also referred to as the rate response curve [6, 13], where r_{in} and r_{out} are the input and output rates of the probing traffic, λ is the cross traffic rate and C is the channel capacity. Therefore, estimating the available bandwidth is equivalent to finding the break in the rate response curve. *Iterative* probing tools [14] aim to detect the break using a rate scan, i.e. by iteratively increasing their sending rate. While (2) is typically applied to packet trains, it can easily be mapped to a gap response curve, used for packet pairs [13] by using the expressions $g_{in} = L/r_{in}$ and $g_{out} = L/r_{out}$ where L is a given packet size and g_{in} and g_{out} are the input and output gaps between successive packets.

The BART tool [7], a successor of DietTOPP [10], is an iterative probing method which uses a Kalman filter to obtain both, the end-to-end available bandwidth and the bottleneck capacity. However, the approach employs iterative probing while our work focuses on direct probing and an analytical derivation of the Kalman filter noise parameters.

If the link capacity C is known in advance (2) can be solved for the cross traffic rate λ if the probing rate is larger than the available bandwidth. In contrast to iterative probing this yields one estimate for every probe. Hence, rather than testing whether the probing rate is above or below the available bandwidth, the cross traffic itself is sampled. This approach is referred to as *direct* probing [14]. Let $r_{in} = C$ it follows from (2) that the available bandwidth can be calculated directly by [5, 9]

$$AB = C \left(2 - \frac{C}{r_{out}} \right) = C \left(1 - \frac{g_{out} - g_{in}}{g_{in}} \right). \quad (3)$$

In wireless networks the DCF aims to achieve per packet fairness for the medium access. Studies have related this fairness to the fair share achieved by Generalized Processor Sharing, e.g [15, 16]. Hence, the assumption of FCFS scheduling does not hold in wireless LANs. The fair share fs can be computed recursively as a solution of

$$fs : \sum_{i=1}^M \min\{r_i, fs\} = C \quad (4)$$

where C is the capacity, r_i is the sending rate of station i , and M is the number of competing stations. Unlike in FCFS systems in fair queueing systems iterative probing reports the fair share, which represents an upper bound for the available bandwidth. However, using (3) in fair queueing systems yields neither the available bandwidth nor the fair share [8]. Thus, direct probing tools using this equation tend to fail in wireless networks. Nevertheless, direct probing using a rate above the fair share easily reports the fair share, since it is equal to the output rate.

A partially implicit assumption of many bandwidth estimation tools is a simplified network model: the network path is abstracted by a single tight link, cross traffic is viewed as constant rate fluid and channel conditions are invariant. At least the last point is not necessarily true in wireless networks due to interference of other stations, other external radio sources or rate adaption.

3 Filtering of Active Probing Measurements

In order to detect changes in the cross traffic process, we probe continuously over time. Therefore, we require an accurate and fast probing approach with low probing overhead. To this end, we use direct probing as it provides fast and fairly reliable samples in FIFO as well as DCF networks. We measure the dispersion of packets in a packet train of length $l + 1$ with packet size L and derive the averaged gap of the departures, i.e. packets at the receiver side by

$$g_d = \frac{d(l+1) - d(1)}{l} \quad (5)$$

where $d(n)$ represents the receiving time-stamp of the n th packet. In FIFO networks available bandwidth is derived by (3) whereas the fair share is computed by $fs = L/g_d$.

In a perfect constant fluid system, every single sample would lead to a correct bandwidth estimate. Unfortunately, estimates derived through injected packet probes are strongly influenced not only by packetized cross-traffic but also by issues such as interface buffers or timer inaccuracies. We regard the resulting probing train gap variations as *measurement noise*. We found that these effects are diminished when using larger packet trains. Most existing tools use averaging over several probe estimates to deal with measurement noise and obtain more accurate bandwidth estimates [13, 17]. In essence, all approaches utilize some form of filtering. For instance, the use of packet trains itself constitutes a low pass filter. A naive approach for on-line bandwidth estimation is to employ a moving average over the past N samples. In all cases an open question is the suitable filter parametrization. Parameters include train lengths, probing intensity and filter lengths. In the following, we employ the Kalman filter, which known to produce optimal estimates for processes perturbed by Gaussian noise. In section 4.1 we argue, that active probing methods which use long trains of packet probes, produce estimates which are normally distributed around the true available bandwidth. Consequently, Kalman filtering is ideal for eliminating this type of measurement error. Additionally, we relate the Kalman filter to an exponentially weighted moving average filter (EWMA).

In order to correctly parametrize a probing technique, it is vital to consider its use-case. It is evident that in a scenario where the user is only interested in the average available bandwidth over several minutes different settings are required than if changes must be detected within seconds. To this end, we require the user to specify the minimum bandwidth change B to be identified within a given time T_s . Furthermore, we assume that the maximum probing rate is constrained to a specific value r_p (to e.g. 5% of the maximum link capacity), resulting in a maximum inter-train sending time $t_\Delta = L(l+1)/r_p$. Given these constraints, our goal is to derive optimal settings for the Kalman filter.

4 Bandwidth Estimation Using Kalman Filtering

As shown in [18] the channel access procedure used in the 802.11 DCF transforms the cross traffic process into an uncorrelated, approximately Gaussian

distributed random process. Furthermore, in Sect. 4.1 we demonstrate that Poisson cross traffic observed over long timescales has the same properties. Therefore, we model perturbations within packet train samples as a Gaussian noise process. This makes the Kalman filter an ideal candidate for estimating the true channel bandwidth.

In the following, we will outline the working of the Kalman filter and present simplifications, which are applicable when the filter noise parameters are time invariant. Moreover, we will show how suitable noise values should be derived in the context of available bandwidth and fair share measurements. Furthermore, we will demonstrate the effect of the parameters on the filter performance and convergence speed.

It is well known that the Kalman filter is the *minimum mean squared error* (MMSE) estimator when the process and measurement noises are Gaussian. If the Gaussian assumption is dropped, the Kalman filter is still the *best linear unbiased estimator*. The filter can estimate the state x_k at time k of any system which can be represented in a state space framework

$$x_k = Ax_{k-1} + B_k u_k + \omega_k \tag{6}$$

$$z_k = Hx_k + \nu_k \tag{7}$$

where A is a state transition matrix, B is a control-input matrix associated with an external input u_k , and ω_k is a normally distributed process noise with $\omega_k \approx N(0, Q)$. The measurements z_k , linked to the system state through an observation matrix H are perturbed by a normally distributed measurement noise $\nu_k \approx N(0, R)$.

The optimal state estimate is obtained by iteratively applying a set of equations [19] known as the time and measurement updates as samples z_k , e.g. packet train estimates in our specific use-case, become available. In our probing framework, the state estimate, i.e. the available bandwidth, is scalar, $A = 1$, $B = 0$ and $H = 1$. The reduced Kalman equations are then given by

$$G_k = \frac{P_{k-1} + Q}{P_{k-1} + Q + R} \tag{8}$$

$$\hat{x}_k = \hat{x}_{k-1}(1 - G_k) + G_k z_k \tag{9}$$

$$P_k = (1 - G_k)(P_{k-1} + Q) \tag{10}$$

Above G_k is the Kalman gain and P_k is the filter estimate error variance. Typically, the filter is initialized with a guess for the state estimate x_0 and large value P_0 representing the uncertainty associated with the guess.

It is evident from equations (8, 9, 10) that the error variance P_k and the Kalman gain G_k are independent of the current state estimate \hat{x}_k and the measurement z_k . Moreover, for stationary noises the parameters P_k and G_k quickly converge to constant values P_∞ and G_∞ respectively [19]. The values of these steady-state parameters can be calculated offline analytically

$$P_\infty = \frac{Q}{2} \pm \frac{Q}{2} \sqrt{1 + 4R/Q} = P_\pm \tag{11}$$

$$G_\infty = \frac{P_+}{P_+ + R} = -\frac{P_-}{R} \tag{12}$$

where P_∞ is positive. Evidently, the steady-state Kalman filter equations are equivalent to the recursive formulation of the exponentially weighted moving average (EWMA) filter with smoothing factor G_∞ . Using $z_k = x_k$ we get

$$\hat{x}_k = (1 - G_\infty)\hat{x}_{k-1} + G_\infty x_k \tag{13}$$

In contrast to classic EWMA filtering, where the smoothing factor is generally selected in an ad-hoc manner, the Kalman framework allows us to optimally parametrize the filter. If the Gaussian noise variances Q and R are known, the Kalman filter is optimal, i.e. no other filter can achieve a smaller MSE. For non-Gaussian noises, the filter is still the best linear unbiased estimator. In the following sections, we will outline a procedure for determining suitable process and measurement noises.

4.1 Measurement Noise Parametrization

We now derive the measurement noise associated with packet train sampling of a Poisson cross-traffic process. To calculate the dispersion and variance of g_d caused by cross traffic in the wireless case we use [18]

$$d(l + 1) - d(1) = (l + K) \left(\frac{L}{C} + \Delta \right) + \sum_{j=2}^{l+1} b(j) \tag{14}$$

where K describes the random number of cross traffic packets transmitted between packet 1 and l of the probing stream, Δ accounts for the protocol overhead and $b(j)$ for the DCF back-off procedure. For a wired FIFO system we set Δ and $b(j)$ to zero. The dominant source of randomness is given by the number of inter-transmitted packets K . Thus, the variation and the measurement noise mainly depend on K and can be derived from its distribution.

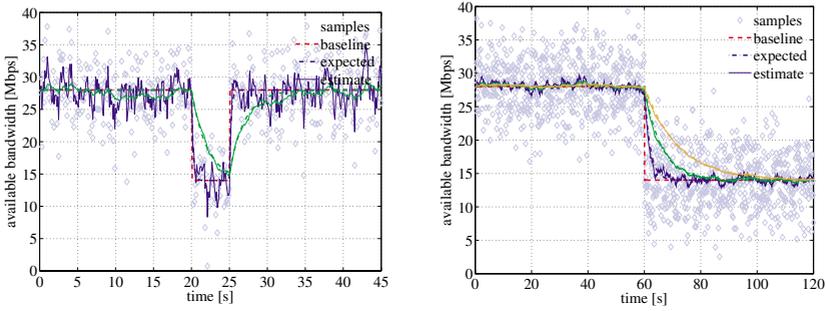
To derive the conditional distribution of $P[K = k|l]$, i.e. that a cross traffic source transmits k packets given a tagged station transmits l packets in a DCF system, [18] uses probability theory and findings presented in [20] and [21]. The conditional distribution of k under l can be expressed as follows

$$P[K = k|l] = P \left[\sum_{j=1}^k b_1(j) \leq b_2 l \text{ and } \sum_{j=1}^{k+1} b_1(j) > b_2 l \right] \tag{15}$$

where b_1 are i.i.d. random variables representing the inter-arrival times of cross traffic whereas b_2 is the gap between two successive packets of a CBR probing stream with length l . Based on this model, the authors relate measurement noise to probing train length. Equipped with these findings, we derive a similar expression for cross traffic in FIFO networks.

Lemma 1 (Poisson approximation). *Let λ be the average packet rate of Poisson cross traffic arrivals. Furthermore, let l be the length and r_p the rate of a probing stream. For $E[K] = l\lambda/r_p \gg 1$ (15) is approximately Gaussian where*

$$P[K \leq k|l] \approx P \left[N(0, 1) \leq \frac{k - l\lambda/r_p}{\sqrt{l\lambda/r_p}} \right].$$



(a) Change tracking using different process noise values: $Q=3.8$, $Q=0.38$. (b) Influence of process noise on convergence speed: $Q=0.04$ ($T=10s$), $Q=0.0044$ ($T=30s$), $Q=0.0011$ ($T=60s$).

Fig. 1. Simulation of estimates corrupted by Gaussian noise ($R=16$) with $t_{\Delta} = 0.1$ s

Proof. Using the central limit theorem, the Poisson distribution can be approximated by a normal distribution for $E[K] \gg 1$ with mean $\mu = E[K]$ and variance $\sigma^2 = E[K]$, i.e. it becomes $N(E[K], E[K])$ -distributed

$$P[K = k] \approx \frac{1}{\sqrt{2\pi E[K]}} e^{-\frac{(k - E[K])^2}{2 E[K]}}$$

To calculate $E[K]$, we assume the cross traffic arrivals as a Poisson process with an average packet rate λ . For a probing stream of length l and probing rate r_p , we get

$$E[K] = \frac{l}{r_p} \lambda$$

that describes the expected average packet arrivals during a sample interval related to the probing stream. We can now use the normal distribution to calculate the conditional probability

$$P[K \leq k|l] \approx P\left[N\left(0, \frac{l}{r_p} \lambda\right) \leq k - \frac{l}{r_p} \lambda\right]$$

Finally, we use the fact, that if X is $N(a\mu, a^2\sigma^2)$ then $Y = X/a$ is $N(\mu, \sigma^2)$ with $a^2 = l\lambda/r_p$ to standardize the result.

Combining (5) and (14) we find that in FIFO systems the variation of g_d is given by the distribution K/l . Using Lemma 1 and $Var(aX+b) = a^2Var(X)$ we calculate the standard deviation for Poisson cross traffic and derive the measurement noise

$$\sigma_{g_d} = \sqrt{\frac{\lambda}{r_p} \frac{L}{C}} \tag{16}$$

4.2 Process Noise Parametrization

For optimal operation, the Kalman filter must also be supplied with the true variance Q of the measured process. Essentially, the parameter Q determines how quickly the filter considers new measurements to be reliable. However, generally the variance of the measured process is not known in advance. Nevertheless, we may choose a value based on knowledge of the type of changes we are interested in capturing. Even if the measured process is not normally distributed, the Kalman filter provides the lowest MSE achievable by a linear filter.

Consider the simulation scenario depicted in figure 1(a). After 20 s the fair share abruptly drops from 28 to 14 Mbps. Expressed as a gap length, we denote this change of $B = 14$ Mbps as $g_j = L/B$. The fair share remains in this state for $T_s = 5$ s. Samples corrupted by a Gaussian noise process with variance $R = 16$ are collected every $t_\Delta = 0.1$ s. To ensure that the filter follows cross-traffic variations, we must set $Q > 0$. It is evident that if Q is too large, the estimate will be unnecessarily noisy. If, on the other hand, Q is too small, the discontinuity will be over-smoothed. This might be acceptable if one is not interested in tracking such short variations. Let us however assume that the application relying on our bandwidth estimate can benefit from accurately identifying this short fair share jump.

We can optimally identify the discontinuity by calculating the variance Q for the segment T_s after the fair share change. As $T_s/t_\Delta = n_p$ samples are generated during T_s , Q is calculated as follows

$$Q = \frac{g_j^2}{n_p} = \frac{g_j^2 L}{T_s r_p} (l + 1) \tag{17}$$

Evidently Q decreases linearly as the number of samples in the segment is increased, i.e. the number of packets per train is reduced. Using Eq. 17 the Kalman filter will yield the minimal MSE within the considered segment. Naturally the detection of shorter discontinuities will be sub-optimal. As a result, the parametrization can be viewed as a lower bound for the filter tracking ability.

4.3 Convergence Speed and Estimate Variance

We now consider the effects of the filter parametrization on the estimate. Firstly, we calculate the time needed for the filter to converge to a new value after an abrupt change. To this end, we examine the impulse response of the scalar steady-state Kalman filter where y_k is the system output and u_k is the step function

$$y_k = (1 - G_\infty)^k u_k = e^{-\alpha k} u_k \tag{18}$$

Evidently, the convergence speed to a new bandwidth is exponential. To calculate the convergence time, we make use of the fact that $e^{-\alpha k}$ decays to less than 1% of its initial value after $k = 5/\alpha$ time-steps.

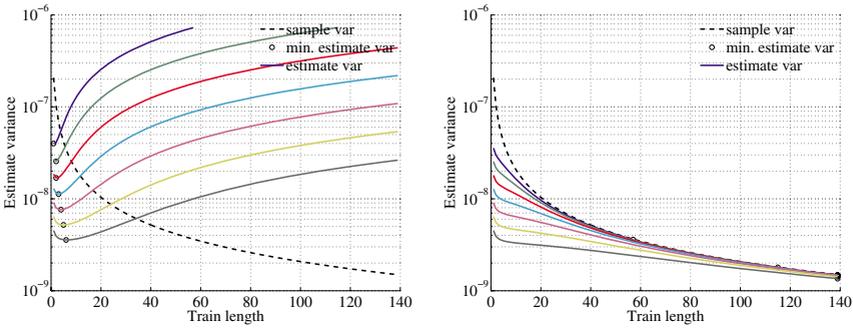


Fig. 2. Estimate variance for a 14 Mbit bandwidth change within $T_s = 0.5, 1, 2, 4 \dots 32s$ (solid curves; top to bottom) and dependency on train length and measurement noise variance. Within the segment T_s in (a) and for $t \gg T_s$ in (b).

The relationship¹ between the noise process variances Q and R to α , and consequently G_∞ is derived using

$$e^{-\alpha} = (1 - G_\infty) = 1 + \frac{P_-}{R} = 1 + \frac{Q}{2R} - \frac{Q}{2R} \sqrt{1 + 4R/Q} \quad (19)$$

$$e^\alpha = \frac{1}{(1 - G_\infty)} = \frac{R + P_+}{R} = 1 + \frac{Q}{2R} + \frac{Q}{2R} \sqrt{1 + 4R/Q} \quad (20)$$

$$\cosh(\alpha) = 1 + \frac{Q}{2R} \quad (21)$$

Moreover, taking into account that packet train probes are sent every t_Δ seconds, we can calculate the filter convergence time for a set of values Q, R

$$T = 5t_\Delta / \operatorname{arccosh}(1 + \frac{Q}{2R}) \quad (22)$$

Next, we derive the overall variance of the filtered estimate during the discontinuity period T_s . The EWMA filter is in essence an auto-regressive $AR(1)$ process with $a_1 = 1 - G$, driven by the process $G_\infty z$. The variance of the sample process z is the sum of the process and measurement variances $\operatorname{Var}(z) = Q + R$, which are independent by definition. Therefore, $\sigma_z^2 = \operatorname{Var}(G_\infty z) = G_\infty^2 (R + Q)$. Thus, the overall variance of the filtered estimate process is given by

$$\operatorname{Var}(\hat{x}_k) = \frac{\sigma_z^2}{1 - a_1^2} = \frac{G_\infty^2 (Q + R)}{1 - (1 - G_\infty)^2} \quad (23)$$

Figure 2(a) illustrates the relationship between the estimate variance and train length for discontinuity periods of different lengths.

Assuming no further cross-traffic jumps, for times significantly longer than T_s the effects of the jump on the variance become negligible, i.e. Q tends towards

¹ For $R \gg Q$ the approximation $\alpha \approx \sqrt{Q/R}$ may be used.

zero resulting in $\text{Var}(\hat{x}_k) = G_\infty^2 R / (1 - (1 - G_\infty)^2)$. The train length dependencies for this case are depicted in Figure 2(b).

For cases in which the variance of the cross-traffic is known in advance, we can readily employ equation 23 to determine the probing train length which produces a minimal MSE.

4.4 Packet Train Length Considerations

For the remainder of the paper, we focus on the wireless case, i.e. fair share estimation. The measurement noise variance derived in [18] is used.

As we showed above, increasing the packet train length l reduces the variance of the bandwidth samples proportionally to $1/l$. Additionally, because the ratio between the bandwidth used per train and number of train gaps $(l + 1)L/l$ is non-linear, probing bandwidth is wasted when using short packet trains. It is also clear that long train lengths, i.e. low sampling rates, will have a negative impact on the bandwidth change tracking accuracy.

Next, we analyze the optimal train length for a given T_s and B . Figure 2(a) depicts the variance of the filtered samples for a process noise optimized to track changes of $B = 14$ Mbps within $T_s = 0.5, 1, 2, 4 \dots 32s$. This variance is related to the MSE between the estimate and the actual bandwidth during the time T_s . It is comprised of the deviation during the convergence period T and the measurement variance after convergence.

It is evident that for the depicted scenario, it is desirable to use short train lengths even if the resulting sample variance R is increased: the overall segment MSE is minimized. For $t \gg T_s$, the filter has the largest variance improvement for short train lengths, as depicted in Figure 2(b). However, as the cross-traffic is assumed to be constant, we can use long packet trains sent at a low frequency to further improve the estimate.

Naturally, increasing the probing traffic intensity r_p will yield even better results, as longer trains can be used, yielding a lower probe variance.

5 Experimental Evaluation of Bandwidth Estimation

Based on the findings in Sect. 4 we developed a modular, portable measurement framework called *WiProbe*. It implements direct probing for estimating the available bandwidth in the wired domain and the fair share in wireless networks. Kalman filtering is used to continuously remove measurement noise from the probes.

In order to evaluate our method and the implementation of our tool, we performed experiments in a controlled testbed environment containing both wired and wireless links. We investigate the performance of the Kalman filter and the effects of parametrization to provide an underpinning of our theoretical findings in section 3 and section 4.

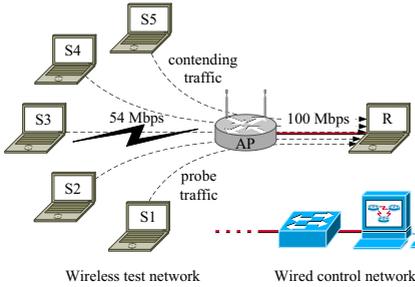


Fig. 3. Wireless testbed setup

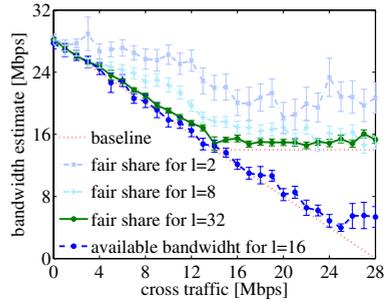


Fig. 4. Accuracy of WiProbe using the probe gap model for wired (FIFO) networks and the fair share model for wireless (DCF) networks at different train lengths

The wireless testbed² setup is depicted in Fig. 3. The distance between the wireless stations and the access point was between 0.5 m and 1.5 m. We switched off RTS/CTS, automatic rate adaptation as well as packet fragmentation. The DCF is used for medium access. All nodes were connected to a separate switched Ethernet control network. We employed *SSHLauncher* [22] to automate the experiment execution.

First, we focus on the accuracy of unfiltered packet train probes. We estimate the fair share by sending probing traffic from S1 to R and a single contending flow from S2 to R with an increasing rate λ from 0 Mbps to 28 Mbps. The cross traffic is generated using the D-ITG [23] traffic generator. Fig. 4 shows the average of 25 fair share and bandwidth estimates with different probing train lengths gathered in approximately 1 second per rate for all rates of cross traffic. Furthermore, we show the corresponding confidence intervals at a confidence level of 0.95. As a reference, the true fair share of a new flow $fs = \max\{C - \lambda, C/2\}$ and the available bandwidth are plotted. Due to large protocol overhead [8, 24], we use a nominal capacity $C = 28$ Mbps for a packet size of 1500 Bytes. It is obvious that at least for long probing trains, WiProbe accurately estimates the available bandwidth and the fair share in FIFO and DCF networks respectively. As stated in section 3, using shorter train lengths results in samples with a higher variance, as indicated by the larger confidence intervals in Fig. 4. Furthermore, the direct probing samples for FIFO networks and high cross traffic rates exhibit some inaccuracies due to packet loss caused by congestion.

² We used Lenovo ThinkPad R61i notebooks with 2.0 GHz, 2 GB RAM running Ubuntu Linux 8.04 with kernel version 2.6.24. We employed the internal Intel PRO/Wireless 4965 AGN IEEE 802.11g WLAN adapters. The access point is a Buffalo Wireless-G 125 series running DD-WRT version 24 RC-4. The switch used in the wired test network is a Netgear FS-108.

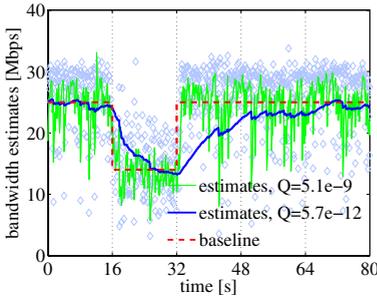


Fig. 5. $B = 14$ Mbit, $T_s = 16$ s: Optimal $Q = 5.1 \times 10^{-9}$ and non-optimal $Q = 5.7 \times 10^{-12}$ process noise

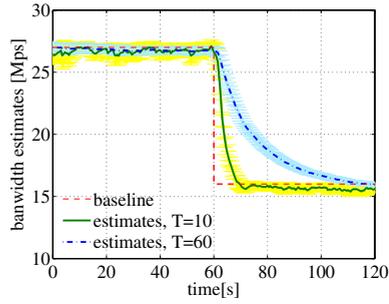


Fig. 6. Effect of process noise on estimate convergence speed using WiProbe

Fig. 5 depicts the effect of train length and process noise on the ability to track a fair share discontinuity. Based on the results plotted in Fig. 2(a) we probe using a train length of $l = 8$ to detect the change of 14 Mbits for $T_s = 16$ s. Using the calculated, optimal process noise $Q = 7.1 \times 10^{-5}$, the jump is accurately tracked at the expense of a higher estimate noise. Nevertheless, the measurement noise is significantly lower when compared to the unfiltered case. Using a sub-optimal value for Q , the filtered estimate variance is reduced for longer timescales, however the estimate of the discontinuity is highly distorted.

The measurement results depicted in Fig. 6 confirm our theoretical findings from section 4.3. We were able to achieve a predetermined convergence time by selecting the process noise Q according to Eq. 23. As expected, fast convergence times are associated with a larger estimate variance as indicated by the larger confidence intervals in Fig. 6.

6 Conclusion

In this paper we showed the benefit of employing Kalman filtering for improving estimates derived from continuous active probing of fair share or available bandwidth using a constant probing rate. Specifically, we showed how filter parameters should be chosen to fit a specific use case and calculated the effects of filtering on the estimate variance. Additionally, we showed the relationship between filter convergence time and the process and measurement noise parameters. Furthermore, we evaluated the influence of train lengths on the estimate’s variance for wired and wireless setups. We conclude that depending on the time-scale and intensity of the cross traffic variations of interest, it can be beneficial to sample cross-traffic frequently using short packet trains. When tracking short-term changes is not a primary concern, long packet trains sent less frequently should be employed, as these provide estimates with the lowest variance.

References

1. Jacobson, V.: Congestion avoidance and control. In: Proc. ACM SIGCOMM, pp. 314–329 (1988)
2. Keshav, S.: A control-theoretic approach to flow control. In: Proc. ACM SIGCOMM, pp. 3–15 (1991)
3. Jain, M., Dovrolis, C.: Pathload: A measurement tool for end-to-end available bandwidth. In: Proc. of PAM (2002)
4. Ribeiro, V., Riedi, R., Baraniuk, R., Navratil, J., Cottrell, L.: PathChirp: Efficient available bandwidth estimation for network paths. In: Proc. of PAM (2003)
5. Strauss, J., Katabi, D., Kaashoek, F.: A measurement study of available bandwidth estimation tools. In: Proc. ACM IMC, pp. 39–44 (2003)
6. Melander, B., Björkman, M., Gunningberg, P.: Regression-based available bandwidth measurements. In: Proc. of SPECTS (2002)
7. Ekelin, S., Nilsson, M., Hartikainen, E., Johnsson, A., Mangs, J.E., Melander, B., Björkman, M.: Real-time measurement of end-to-end available bandwidth using kalman filtering. In: Proc. IEEE/IFIP NOMS, pp. 5–17 (2006)
8. Bredel, M., Fidler, M.: A measurement study of bandwidth estimation in IEEE 802.11g wireless LANs using the DCF. In: Das, A., Pung, H.K., Lee, F.B.S., Wong, L.W.C. (eds.) NETWORKING 2008. LNCS, vol. 4982, pp. 314–325. Springer, Heidelberg (2008)
9. Li, M., Claypool, M., Kinicki, R.: WBest: A bandwidth estimation tool for multimedia streaming application over IEEE 802.11 wireless networks. Technical Report WPI-CS-TR-06-14, Department of Computer Science - Worcester Polytechnic Institute (2006)
10. Johnsson, A., Melander, B., Björkman, M.: Diettopp: A first implementation and evaluation of a simplified bandwidth measurement method. In: Proc. of SNCNW, p. 5 (2004)
11. Shriram, A., Kaur, J.: Empirical evaluation of techniques for measuring available bandwidth. In: Proc. IEEE INFOCOM (2007)
12. Jain, M., Dovrolis, C.: End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput. In: Proc. ACM SIGCOMM, pp. 295–308 (2002)
13. Liu, X., Ravindran, K., Loguinov, D.: A queuing-theoretic foundation of available bandwidth estimation: Single-hop analysis. IEEE/ACM Trans. Networking 15(4), 918–931 (2007)
14. Jain, M., Dovrolis, C.: Ten fallacies and pitfalls on end-to-end available bandwidth estimation. In: Proc. ACM IMC, pp. 272–277 (2004)
15. Vaidya, N.H., Bahl, P., Gupta, S.: Distributed fair scheduling in a wireless lan. In: Proc. of ACM MOBICOM, pp. 167–178 (2000)
16. Bansh, A., Perez, X.: Distributed weighted fair queuing in 802.11 wireless lan. In: IEEE ICC 2002, vol. 5, pp. 3121–3127 (2002)
17. Liu, X., Ravindran, K., Liu, B., Loguinov, D.: Single-hop probing asymptotics in available bandwidth estimation: sample-path analysis. In: Proc. of ACM SIGCOMM, ACM, pp. 300–313. ACM, New York (2004)
18. Bredel, M., Fidler, M.: Understanding fairness and its impact on quality of service in IEEE 802.11. In: IEEE Infocom (2009)
19. Anderson, B.D.O., Moore, J.B.: Optimal Filtering. Prentice-Hall, Englewood Cliffs (1979)

20. Berger-Sabbatel, G., Duda, A., Gaudoin, O., Heusse, M., Rousseau, F.: Fairness and its impact on delay in 802.11 networks. In: Proc. of IEEE GLOBECOM, pp. 2967–2973 (2004)
21. Berger-Sabbatel, G., Duda, A., Heusse, M., Rousseau, F.: Short-term fairness of 802.11 networks with several hosts. In: Proc. of IFIP MWCN, pp. 263–274 (2004)
22. Bozakov, Z., Bredel, M.: Sshlauncher.com - a tool for experiment automation in distributed environments. Technical Report KOM-TR-2008-11, TU-Darmstadt, Merckstr. 25, 64283 Darmstadt (2008)
23. D-ITG: Internet Traffic Generator, <http://www.grid.unina.it/software/ITG>
24. IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Std. 802.11-2007 (2007)