Inferring traffic burstiness by sampling the buffer occupancy

Michel Mandjes   Remco van de Meent

24th August 2004

University of Twente   CWI

Common practice to determine the required bandwidth capacity for a network link is to measure the 5 minute average link load and then add a safety margin to cater for traffic bursts on small time-scales. Because of the substantial measurement effort required to appropriately determine the effect of these bursts, network managers often rely on rules of thumb to find the safety margin, e.g. ‘the mean plus 50%’.

In this paper we propose a novel method to accurately determine the burstiness of traffic on small time-scales, *without* requiring measurements on such small time-scales. Instead, our method is based on coarse-grained polling of the occupancy of a buffer in front of the network link, and ‘inverting’ the resulting statistical distribution of the buffer contents to find the burstiness of the offered traffic on small time-scales.

We provide theoretical foundations of the inversion approach, relying on a large-deviations framework. Validation is done through both simulation using synthetic traffic, as well as extensive measurements in various operational networks, showing remarkably accurate estimates of the traffic’s burstiness. In fact, we are able to accurately determine the burstiness on small time-scales (for instance 5 ms) by sampling the buffer occupancy (for instance) every second.

1 Introduction

Provisioning of network resources addresses the interrelationship between: (i) offered traffic (in terms of both average load and burstiness), (ii) desired level of performance, and (iii) the required capacity. Generally, more capacity is needed when offered load and burstiness increase, or when the performance criterion becomes more stringent. To operate a network in a viable way, provisioning procedures balancing (i), (ii), and (iii) are required: scarce provisioning inevitably leads to performance degradation, whereas (too much) over-provisioning results in a waste of resources.

Provisioning procedures are commonly based on rules of thumb. Considering a time window in which network traffic can be assumed stationary, one often uses rules of the type 'the mean traffic rate, plus a margin of 50%'. Obviously, such a fixed margin is not universally applicable. This motivates the use of formulae of the type

$$C = M + \alpha\sqrt{V} \tag{1}$$

for mean load M, some factor α (reflecting the performance target, which is mostly determined by the applications involved), and a standard-error term $\sqrt{V}$ to account for traffic fluctuations, see for instance [1]. Hence, the size of the 'safety margin' is affected by the chosen performance target, and the burstiness of the offered traffic. From (1) we conclude that provisioning (for a given performance target) requires knowledge of both M and V.

The mean traffic rate M can be determined by standard coarse-grained traffic measurements. A common way is to poll Interfaces Group MIB counters via the Simple Network Management Protocol (SNMP) every 5 minutes; this yields the total amount of traffic sent through the network interface over this time-interval. Determining the burstiness V, on the other hand, is more involved, and is essentially the subject of this paper.

Let $A(t)$ denote the traffic offered over a window of length $t > 0$, and the function $V(t) := \mathbb{V}\mathrm{ar}A(t)$. Then our analysis indicates that there is some 'dominant time-scale' T, such that the (burstiness) V in formula (1) corresponds to $V(T)$; hence it is this $V(T)$ that needs to be estimated. Note that the time-scale T is usually relatively small (for instance in the order of 100 ms, and often even considerably less). Clearly, $V(T)$ can be estimated directly by doing measurements on time-scale T, and computing the sample variance. However, it is hard to do accurate measurements on these small time-scales; they are hardly feasible through SNMP. This motivates the need for alternative, 'cheap' measurement techniques for determining the traffic variance on small time-scales.

1.1 Contribution

As argued above, it is of crucial interest to develop methods for efficiently and accurately estimating $V(T)$ for small time-scales T. The main contribution of this paper is that we propose an alternative for 'direct estimation' (i.e., by doing measurements on the time-scale of interest). In our approach the (i) *buffer occupancy is polled* on a regular basis (for instance every 10 seconds), and (ii) subsequently we 'invert' the resulting (estimated) buffer content distribution to the variance function $V(\cdot)$.

Importantly, this approach eliminates the need for traffic measurements on small time-scales. In this sense, we remark that our proposed procedure is rather counterintuitive: without doing measurements on time-scale T, we are still able to accurately estimate $V(T)$. In fact, one of the attractive features of our 'inversion procedure' is that it yields the *entire* 'variance curve' $V(\cdot)$ (of course up to some finite horizon), rather than just $V(T)$ for some pre-specified T.

1.2 Approach and organization

The variance estimation technique proposed in this paper relies on the assumption that traffic is (fairly) Gaussian: for any $t \geq 0$, the amount of traffic offered in a time window of length t is accurately described by a normal distribution, parameterized by a mean Mt and variance $V(t) := \mathbb{V}\mathrm{ar}A(t)$. This Gaussianity was observed in various measurement studies. Kilpi and Norros [2] have statistically verified that the use of Gaussian traffic models is justified as long as the aggregation is sufficiently large (both in time and number of flows), due to Central Limit type of arguments. Importantly, Gaussian models cover long-range dependent processes, such as fractional Brownian motion (fBm); traffic measurements in the 1990s showed that in various situations this fBm model accurately models network traffic, see, e.g., [3].

Section II presents preliminaries on Gaussian queues. Particular attention is paid to provisioning formulae (for both buffer and bandwidth), in line with (1). The provisioning formula motivate the need for methods to estimate the variance function $V(\cdot)$, i.e., $V(t)$ as a function of the interval length t. The formulae for Gaussian queues lead to our efficient method for estimating the $V(\cdot)$, that is explained in Section III; importantly, we derive an 'inversion formula' that yields $V(\cdot)$ from the empirically determined buffer content distribution. In Section IV we describe the inversion procedure, and demonstrate it by using synthetic traffic (in this case fBm traffic).

In the inversion procedure, we have identified three sources of possible errors: (A) the (large-deviations) asymptotics give an *approximation* of the overflow probability (rather than an exact formula), (B) the buffer content distribution is *estimated* through the buffer polling procedure, and there can still be estimation errors, and (C) we *assume* that traffic is (accurately approximated by) Gaussian. In Section V we present a detailed, quantitative study of the impact of each of these errors on the resulting estimation of the variance.

To investigate the applicability of the procedure in practice, we have performed extensive numerical experiments with real data from various real-life settings. These networks have different access technologies and link speeds, and different applications and user populations. We present and discuss the results of this experimental validation in Section VI. It is noted that both the simulations of Sections IV and V, as well as the experiments of Section VI show that the inversion procedure yields remarkably accurate estimates of the traffic's burstiness.

Section VII presents a number of reflections of the feasibility of implementing our estimation procedure, and conclude that there are no conceptual impediments. Section VIII concludes, and and lists a number of possible directions for future work.

2 Gaussian queues – motivation

In this section we review some basic principles of Gaussian traffic, and recapitulate the main fundamental (large-deviations) theory for queues with Gaussian input. Then we derive, for this Gaussian setting, a number of dimensioning rules. These formulae motivate the need for estimating specific traffic characteristics, viz., the mean rate M and the variance function $V(\cdot)$, cf. provisioning rule (1).

2.1 Preliminaries on Gaussian queues – many-sources asymptotics

Consider n independent, statistically identical *Gaussian* sources. It is assumed that the traffic pattern generated by an individual source corresponds to a Gaussian process with stationary increments. This type of sources is characterized by (i) their mean traffic rate μ, and (ii) their variance function $v(t)$, for $t \geq 0$. With $A_i(t)$ denoting the amount of traffic generated by the ith source in an interval of length $t \geq 0$, then $\mathbb{E}A_i(t) = \mu t$, and $\mathbb{V}\mathrm{ar}A_i(t) = v(t)$.

Now suppose that the n sources feed into a queue with capacity C, and apply the scaling $C \equiv nc$. It is well-known that the stationary queue length, say Q_n has the same distribution as the maximum of the corresponding 'free-buffer process':

$$Q_n \stackrel{\mathrm{d}}{=} \sup_{t>0} \left(\sum_{i=1}^{n} A_i(-t,0) - nct \right).$$

The following fundamental result can be found in, e.g., [4].

Lemma 2.1 – *see, e.g., Addie, Mannersalo & Norros [4]. Suppose that there is an $\alpha < 2$ such that $v(t)/t^{\alpha} \to 0$ for $t \to \infty$. Then, for any $b > 0$, and $c > \mu$,*

$$\begin{aligned} I(b) &:= -\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}(Q_n \geq nb) \\ &= \inf_{t>0} \frac{(b+(c-\mu)t)^2}{2v(t)}. \end{aligned}$$

The above result holds for the system 'scaled by n', but gives rise to an approximation for the 'unscaled' situation, see also for instance [4], [5, Eq. 3]. With $B \equiv nb$, consider the probability that the buffer content Q exceeds B. Denote $M \equiv n\mu$ as the aggregate mean, and $V(t) \equiv nv(t)$ as the aggregate variance.

Approximation 2.2 *For any $B > 0$, and $C > M$,*

$$\mathbb{P}(Q > B) \approx \exp\left(-\inf_{t>0} \frac{(B+(C-M)t)^2}{2V(t)} \right). \tag{2}$$

2.2 Provisioning formulae

One of the major tasks in network management is the provisioning of resources: choose the link rate and/or buffer size such that some pre-specified performance criterion is met. The above approximation formula (2) provides us with a tool for developing such provisioning rules. For several performance criteria, we derive the corresponding rules.

2.2.1 Link provisioning of an unbuffered resource

Suppose the goal is to provision the link such that the probability of exceeding the capacity C for a period of length T is smaller than ϵ. Hence we have to find the smallest $C = C(T,\epsilon)$ such that

$$\exp\left(-\frac{((C-M)T)^2}{2V(T)} \right) \leq \epsilon,$$

cf. (2). It is readily checked that this yields, with $\delta := \sqrt{-2\log\epsilon}$,

$$C(T,\epsilon) = M + \frac{\delta}{T}\sqrt{V(T)}. \tag{3}$$

Notice that $C(T,\epsilon)$ decreases in ϵ, as expected: the less stringent the performance target, the less bandwidth is needed.

2.2.2 Link provisioning of a buffered resource

In the setting of provisioning rule (3) we considered an unbuffered resource. In practice, however, network elements are often equipped with a queue, to absorb traffic rate fluctuations. If the router has a queue of size B, and suppose we wish to provision the capacity, we have to find the minimal $C = C(\epsilon)$ such that (2) is below ϵ. Hence we are searching for

$$\min\left\{C \mid \forall t > 0 : \exp\left(-\frac{(B + (C-M)t)^2}{2V(t)}\right) \leq \epsilon\right\}.$$

After rearranging terms, we find that

$$C(\epsilon) = M + \inf_{t>0}\left(\frac{\delta}{t}\sqrt{V(t)} - \frac{B}{t}\right). \tag{4}$$

Again, the bandwidth required decreases in ϵ. Moreover, it also decreases in B: the larger the queue, the better traffic fluctuations can be absorbed by the buffer, and hence less link capacity is needed.

2.2.3 Buffer provisioning

Similarly, we can determine the minimum required buffer $B = B(\epsilon)$:

$$B(\epsilon) = \inf_{t>0}\left(\delta\sqrt{V(t)} - (C-M)t\right). \tag{5}$$

$B(\epsilon)$ decreases in ϵ and C, as expected. As an aside, we mention that it can be shown that for fixed performance target ϵ buffer and bandwidth trade off *in a convex way*; this also holds for non-Gaussian traffic, see [6].

Example 2.3 *fBm. Motivated by several measurements studies [3], we focus here on the highly relevant example of fractional Brownian motion input, i.e., Gaussian traffic with* $V(t) = \sigma^2 t^{2H}$ *– take for ease* $\sigma = 1$. $H \in (0,1)$ *is the so-called Hurst parameter; for network traffic a typical value is 0.7-0.8. Straightforward computations give for (3):*

$$C(T,\epsilon) = M + \frac{\delta}{T^{1-H}}.$$

When computing $C(\epsilon)$ *in (4), the optimizing* t *is given by* $B^{1/H}(1-H)^{-1/H}\delta^{-1/H}$*, yielding*

$$C(\epsilon) = M + \delta^{1/H}\left(\frac{1-H}{B}\right)^{1/H-1} H.$$

In buffer provisioning rule (5) the optimizing t is given by $(\delta H)^{1/(1-H)}(C-M)^{-1/(1-H)}$, such that

$$B(\epsilon) = \left(\frac{\delta H}{(C-M)^H}\right)^{1/(1-H)} \frac{1-H}{H}.$$

Notice that (for fixed ϵ) B and C trade off such that $B^{1-H}C^H$ is a constant, cf. [6]. ♢

The main conclusion from this section is that the above provisioning formulae (3), (4), and (5) indicate that it is of crucial importance to have accurate estimates of the average traffic rate M, as well as the variance curve $V(\cdot)$ (i.e., $V(t)$ as a function of $t \geq 0$); having these at our disposal, we can find the required bandwidth capacity or buffer size. As estimating M is straightforward, the next sections concentrate on efficient methods for estimating the variance curve $V(\cdot)$.

3 Derivation of the inversion formula

As mentioned in the introduction, the mean traffic rate M can be determined by standard coarse-grained traffic measurements (polling Interfaces Group MIB counters via SNMP every 5 minutes). It is clear that determining the variance curve $V(\cdot)$ is more involved. The standard way to estimate $V(T)$ (for some interval length T) is what we refer to as the 'direct approach'. This method is based on traffic measurements for disjoint intervals of length T, and just computes their sample variance. It is noted that the convergence of this estimator could be rather slow when traffic is long-range dependent [7, Ch. I], but the approach has three other significant drawbacks:

1. When measuring traffic using windows of size T, it is clearly possible to estimate $V(T)$, $V(2T)$, $V(3T)$, etc. However, these measurements obviously do not give any information on $V(\cdot)$ on time-scales *smaller* than T. Hence, to estimate $V(T)$ measurements should be done at granularity T or less. This evidently leads to a substantial measurement effort.
2. Another problem is that measurements on very small time-scales are not feasible. For a 'representative' setup, i.e., using only off-the-shelf equipment, [8, Section 3] argues that on time-scales smaller than, say, 10 ms, the reliability of the measurements cannot be guaranteed.
3. The provisioning formulae (4) and (5) require knowledge of the *entire* variance function $V(\cdot)$, whereas the direct approach described above just yields an estimate of $V(T)$ on a pre-specified time-scale T. Therefore, a method that estimates the entire curve $V(\cdot)$ is preferred.

This section presents a powerful alternative to the direct approach; we refer to it as the *inversion approach*, as it 'inverts' the buffer content distribution to the variance curve. This inversion approach overcomes the problems identified above. We rely on the many-sources framework of Section 2.1.

Define the 'most likely epoch of overflow' for a given buffer value $b > 0$:

$$t_b := \arg\inf_{t>0} \frac{(b+(c-\mu)t)^2}{2v(t)};$$

note that t_b is not necessarily unique. Define the set $\mathcal{T}$ as follows:

$$\mathcal{T} := \{t > 0 \mid \exists b > 0 : t = t_b\}.$$

The following theorem gives, for any $t > 0$, an upper bound on the variance $v(t)$, for given $I(b)$, and presents conditions under which this upper bound is tight.

Theorem 3.1 *(i) For any $t > 0$,*

$$v(t) \le \inf_{b>0} \frac{(b + (c-\mu)t)^2}{2I(b)}. \tag{6}$$

(ii) There is equality *in* (6) *for all $t \in \mathcal{T}$.*
(iii) If $2v(t)/v'(t) - t$ grows from 0 to ∞ when t grows from 0 to ∞, then $\mathcal{T} = (0, \infty)$.

Proof: Clearly, due to Lemma 2.1, for all $b > 0$ and $t > 0$, we have that

$$I(b) \le \frac{(b + (c-\mu)t)^2}{2v(t)},$$

which implies claim (i) immediately. Now consider a $t \in \mathcal{T}$. Then there is a $b = b_t > 0$ such that

$$I(b) = \frac{(b + (c-\mu)t)^2}{2v(t)}.$$

We thus obtain claim (ii).

Now consider claim (iii). We have to prove that for all $t > 0$ there is a $b > 0$ such that $t = t_b$. Evidently, t_b solves $2v(t)(c-\mu) = (b + (c-\mu)t)v'(t)$, or, equivalently,

$$b = b_t := \left(2\,\frac{v(t)}{v'(t)} - t\right)(c-\mu). \tag{7}$$

Hence, it is sufficient if b_t in the right hand side of (7) grows from 0 to ∞ when t grows from 0 to ∞.

Example 3.2 *Brownian bridge. In a Brownian bridge, time is restricted to the interval $[0, 1]$, and $v(t) = t(1-t)$. It is easily verified that*

$$\arg\inf_{t\in(0,1)} \frac{(b + (c-\mu)t)^2}{2t(1-t)} = \frac{b}{(c-\mu) + 2b};$$

$$I(b) = \frac{b(b + (c-\mu))}{2b + (c-\mu)}.$$

Conclude that $\mathcal{T} = (0, \frac{1}{2})$. We can now retrieve $v(t)$ from $I(b)$, using the inversion formula of Theorem 3.1. To this end, we first compute

$$\arg\inf_{b>0} \frac{(b + (c-\mu)t)^2}{2I(b)} = \frac{(c-\mu)t}{1-2t};$$

then a lengthy calculation indeed gives $v(t) \le t(1-t)$, with equality for $t \in \mathcal{T} = (0, \frac{1}{2}]$. Note that the condition of part (iii) is not met outside $\mathcal{T}$. ♢

Remarkably, Theorem 3.1 gives, loosely speaking, that for Gaussian sources the buffer content distribution uniquely determines the variance function. This property is exploited in the following heuristic.

Approximation 3.3 *The following estimate of the function $V(t)$ (for $t > 0$) can be made using the buffer content distribution:*

$$V(t) \approx \inf_{B>0} \frac{(B + (C - M)t)^2}{-2 \log \mathbb{P}(Q > B)}. \tag{8}$$

Hence, if we can estimate $\mathbb{P}(Q > B)$, then 'inversion formula' (8) of Approximation 3.3 can be used to retrieve the variance; notice that the infimum can be computed for any t, and consequently we get an approximation for the entire variance curve $V(\cdot)$ (of course up to some finite horizon). These ideas are exploited in the procedure described in the next section.

Remark 3.4 *The result of Theorem 3.1 shows an interesting similarity with the main result of [9]. There it was shown that the decay rate function $I(\cdot)$ is convex in buffer level b* if and only if *the variance function $v(\cdot)$ is concave in the optimizing timescale t_b.* ♢

4 Demonstration of the inversion procedure

In this section we show how the theoretical results of the previous section can be used to estimate $V(\cdot)$. In Section IV.A we propose an algorithm for estimating the (complementary) buffer content distribution (in the sequel abbreviated to BCD), such that, by applying Approximation 3.3, the variance curve $V(\cdot)$ can be estimated. In our demonstration (Section IV.B) we specialize to the case of synthetic input, i.e., traffic generated according to some stochastic process; we choose fBm input, but we emphasize that the procedure could be followed for any other process. Section IV.C compares, for fBm, our estimation for $V(\cdot)$ with the actual variance curve, yielding a first impression of the accuracy of our approach (a more detailed numerical evaluation follows in Sections V and VI).

4.1 Algorithm

The inversion procedure consists of two steps: (1) determining the BCD, and (2) 'inverting' the BCD to the variance curve $V(\cdot)$ by applying Approximation 3.3. We propose the following algorithm:

Algorithm 4.1 *Inversion approach.*
1. Collect 'snapshots' of the buffer contents: $q_1, \dots, q_N$; here q_i denotes the buffer content as measured at time $\tau_0 + i\tau$, for some $\tau > 0$. Estimate the BCD *by the empirical distribution function of the q_i, i.e., estimate $\mathbb{P}(Q > B)$ by*

$$\phi(B) = \frac{\#\{i : q_i > B\}}{N}.$$

2. Estimate $V(t)$, for any $t \geq 0$, by

$$\inf_{B>0} \frac{(B + (C - M)t)^2}{-2 \log \phi(B)}. \tag{9}$$

In the above algorithm, snapshots of the buffer content are taken at a constant frequency. To get an accurate estimate of the BCD, both τ and N should be chosen sufficiently large. We come back to this issue in Section V. Notice that we chose a fixed polling frequency (i.e., τ^{-1}) in our algorithm, but this is not strictly necessary; the BCD-estimation procedure obviously still works when the polling epochs are not equally spaced.

In the remainder of this section we demonstrate the inversion approach of Algorithm 4.1 through a simulation with synthetic (fBm) input. The simulation of the queue fed by fBm yields an estimate for the BCD (Section IV.B); this estimated BCD is inverted to obtain the estimated variance curve, which is compared with the actual variance curve (Section IV.C).

4.2 Simulation procedure

We now demonstrate the procedure in more detail. Concentrating on slotted time, we generate traffic according to some stochastic process. In the example below we focus on the (practically relevant) case of fBm input, but it is stressed that the procedure could be followed for any other stochastic process; we have simulated fBm by a *fractional Brownian motion simulator* [10] (based on Davis and Harte's circulant method). The traffic stream is fed into a queue with link rate C. The buffering dynamics are simulated follows:

Algorithm 4.2 *Simulation of the buffer dynamics.*

1. Using the fBm simulator we generate fBm, with a specific Hurst parameter $H \in (0,1)$. This yields a list $A_1, \ldots, A_Z$, for some $Z \in \mathbb{N}$, where A_j denotes the amount of traffic offered in the jth slot.

2. The list $A_1, \ldots, A_Z$ is used to simulate the buffer dynamics. This is done recursively:

$$Q_{j+1} := \max\left\{Q_j + A_j - C, 0\right\},$$

where Q_j denotes the amount of contents in the buffer at the beginning of slot j.

3. The buffer content Q_j is observed every τ slots, which results in $N = Z/\tau$ snapshots q_i of the buffer content. These snapshots are used to estimate $\mathbb{P}(Q > B)$, as described in Algorithm 4.1.

In (standard) fBm, the average traffic rate M equals 0. Trivially, fBm with non-zero drift M can be simulated by replacing the list $A_1, \ldots, A_Z$ by $A_1 + M, \ldots, A_Z + M$.

In this demonstration of the inversion procedure, we generate an fBm traffic trace with Hurst parameter $H = 0.7$ and length $Z = 2^{24}$ slots. The link capacity C is set to 0.8, and we take snapshots of the buffer content every $\tau = 2^7 = 128$ intervals.

4.3 Estimating the variance curve

In this subsection we discuss the output of the inversion procedure for our simulated example with fBm traffic.

First we estimate the BCD; a plot is given in Fig. 1. For presentation purposes, we plot the (natural) logarithm of the BCD, i.e., $\log \mathbb{P}(Q > B)$.

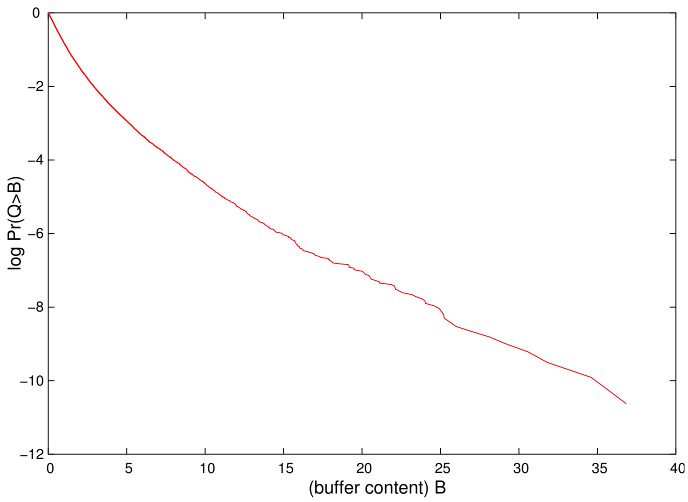

Figure 1: Sample buffer content distribution

The BCD in Fig. 1 is 'less smooth' for larger values of B. This is due to the fact that large buffer levels are rarely exceeded, leading to less accurate estimates.

Secondly, we estimate the variance $V(t)$ for t equal to the powers of 2 ranging from 2^0 to 2^7, using the BCD, i.e., by using (9). The resulting variance curve is shown in Fig. 2 ('inversion approach'). The minimization (over B) in (9) was done by straightforward numerical techniques.

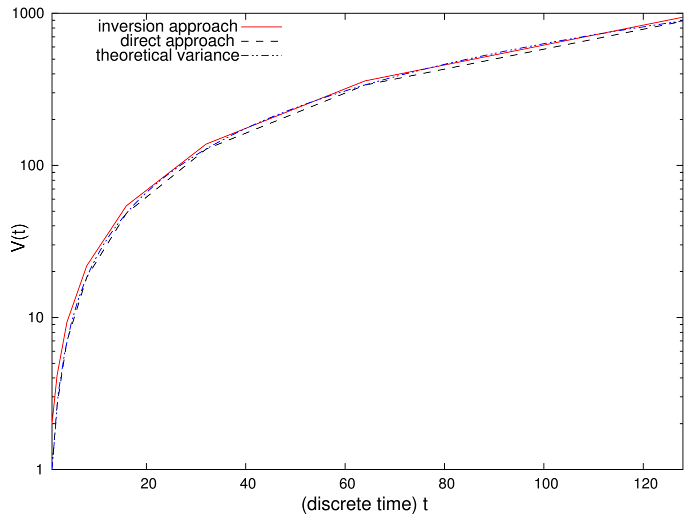

Figure 2: Sample variance curves

To get an impression of the accuracy of the inversion approach, we have also plotted in Fig. 2 the variance curve as can be estimated directly from the synthetic traffic trace (i.e., the 'direct approach' introduced in Section III), as well as the real variance function for fBm traffic, i.e., $V(t) = t^{2H}$.

Fig. 2 shows that the three variance curves are remarkably close to each other. This confirms that the inversion approach is an accurate way to estimate the burstiness. We note that the graph shows that the inversion approach slightly overestimates the variance; hence, when using this curve for provisioning purposes, for instance by applying rules (4) or (5), this would result in conservative numbers.

5 Error analysis of the inversion procedure

In the previous section the inversion approach was demonstrated. It was shown to perform well for fBm with $H = 0.7$, under a specific choice of N and τ. Evidently, the key question is whether the procedure still works under other circumstances. To this end, we first identify the three possible sources of errors:

A) The inversion approach is based on the *approximation* (2).

B) $\mathbb{P}(Q > B)$ is *estimated*; there could still be an estimation error involved. In particular, we wonder what the impact of the choice of N and τ is.

C) The procedure *assumes* perfectly Gaussian traffic, although real network traffic may not be (accurately described by) Gaussian.

In the remainder of this section, we will quantitatively investigate the impact of each of these errors on the inversion procedure. These investigations are performed through simulation, following the procedure outlined in Algorithm 4.2.

5.1 Approximation of the buffer content distribution

In Equation (2) an approximation of the BCD is given. As the inversion approach is based on this approximation, evidently, errors in (2) might induce errors in the inversion. This motivates the assessment of the error made in (2).

We first determine the infimum in the right-hand side of (2), which we consider as a function of B. In line with the previous section, we choose fBm input: $M = 0$ and $V(t) = t^{2H}$. Straightforward calculations now reveal that we can rewrite (2) as follows:

$$\log \mathbb{P}(Q > B) \approx -\frac{1}{2}\left(\frac{B}{1-H}\right)^{2-2H}\left(\frac{C}{H}\right)^{2H}. \tag{10}$$

We now verify how accurate this approximation is, for two values of H: the pure Brownian case $H = 0.5$, and situation with long-range dependence $H = 0.7$ (in line with earlier measurement studies of network traffic).

We generate several runs of fBm traffic (with different random seeds) to generate $N = 2^{24}$ slots of traffic per run, with $M = 0$. We then simulate the buffer dynamics. For $H = 0.5$ we choose link rate $C = 0.2$, for $H = 0.7$ we choose $C = 0.8$; these choices C are such that the queue is non-empty sufficiently often (in order to obtain a reliable estimate of the BCD).

Figs. 3 and 4 show for the various runs the approximation of the BCD, as well as their theoretical counterpart, for $H = 0.5$ and $H = 0.7$. It can be seen that, in particular for small B (i.e., buffer levels that are exceeded frequently), the empirically determined BCD almost perfectly fits the theoretical approximation.

Remark 5.1 *It is noted that the simulator uses slotted time, whereas (2) involves an optimization over continuous time. This actually means that we should take the minimum over $t \in \mathbb{N}$ in (2), instead of $t \in \mathbb{R}$. Computation of the minimum over $\mathbb{N}$ shows that this has hardly any significant impact.* $\diamondsuit$

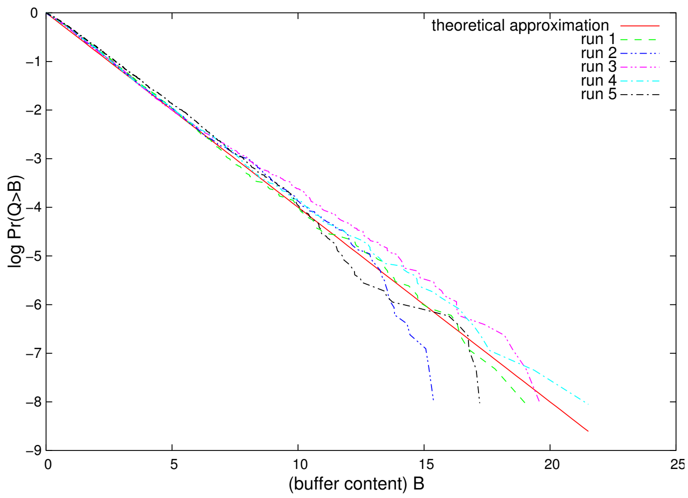

Figure 3: $\mathbb{P}(Q > B)$ plotted against the theoretical approximation (10), for $H = 0.5$

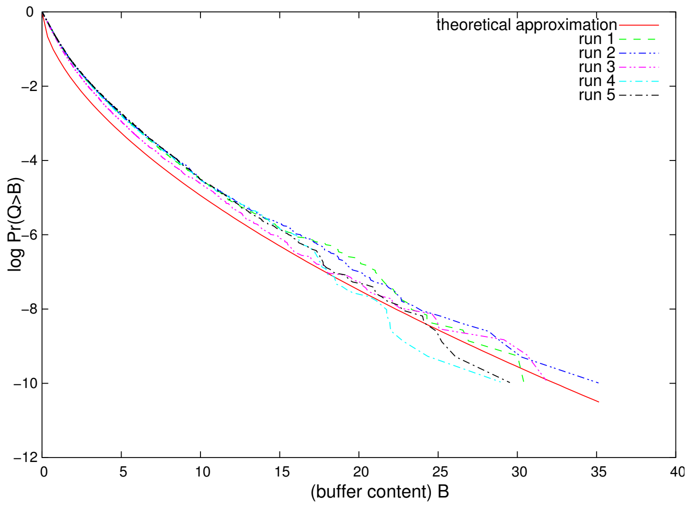

Figure 4: $\mathbb{P}(Q > B)$ plotted against the theoretical approximation (10), for $H = 0.7$

5.2 Estimation of the buffer content distribution

A second possible error source in our inversion approach, relates to the estimation of the BCD. As we estimate the BCD on the basis of the snapshots $q_1, \ldots, q_N$ of the buffer content, there will be some error involved. The impact of this error is the subject of this subsection. It could be expected that the larger N (more observations) and τ (less correlation between the observations), the better the estimate.

We first investigate the impact of N. The simulator is run as in previous cases (with $H = 0.7$), with the difference that we only use the first $x \times N = x \times 2^{24}$ samples to determine $\mathbb{P}(Q > B)$, for $x \in (0, 1]$; we study the impact of x. Fig. 5 shows the estimation of the buffer content distribution, for various x ranging from 0.001 to 1.

From Fig. 5 it can be seen that, in particular for relatively small B, a relatively small number of observations suffices to get an accurate estimate of the BCD.

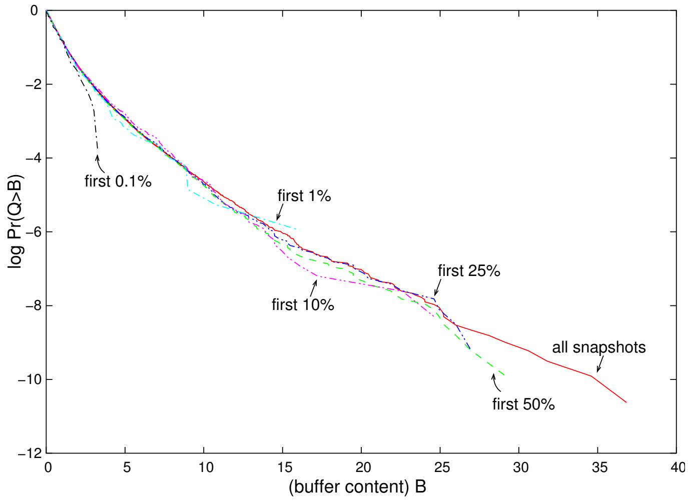

Figure 5: Comparing $\mathbb{P}(Q > B)$ for different trace lengths, for $H = 0.7$

Secondly, we investigate the impact of the interval length between two consecutive snapshots τ. One might expect that the more often the buffer occupancy is polled, the closer the resulting buffer content distribution would look like the theoretical approximation. Note, however, that when the snapshots are taken close together, the observations might be highly correlated due to the long-range dependence of the simulated fBm traffic, which might (negatively) affect the accuracy of the estimate.

Fig. 6 shows the determined buffer content distribution for τ ranging from observing every 32 to every 8192 slots. It can be seen that, quite remarkably, particularly for small B the fit is quite good, even when the buffer content is polled only relatively rarely.

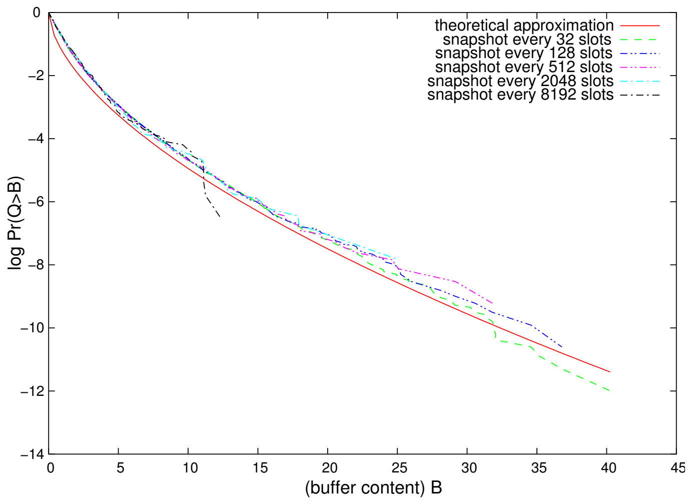

Figure 6: Comparing $\mathbb{P}(Q > B)$ for different observation intervals τ, for $H = 0.7$

5.3 The impact of the Gaussianity assumption

Approximation (2) explicitly assumes that the traffic process involved is Gaussian. Various measurement studies find that real network traffic on the Internet is (accurately described by) Gaussian, see, e.g., [3], but the fit is of course not perfect. In this subsection we investigate the impact of the Gaussianity assumption on the inversion, i.e., we test how the procedure works for traffic that is not (perfectly) Gaussian.

We study the impact of non-Gaussianity as follows. Mix the trace $A_{i_{[\mathrm{fBm}]}}$ generated by the fBm simulator with traffic $A_{i_{[\mathrm{alt}]}}$ from an alternative (non-Gaussian) input stream:

$$A_i := \alpha \cdot A_{i_{[\mathrm{fBm}]}} + (1-\alpha) \cdot A_{i_{[\mathrm{alt}]}},$$

where $\alpha \in [0,1]$ corresponds to the fraction of fBm traffic in the mixture. The resulting traffic stream is fed into the queue, cf. Algorithm 4.2. Clearly, the variance of the traffic mixture is

$$V(t) = \alpha^2 V_{[\mathrm{fBm}]}(t) + (1-\alpha)^2 V_{[\mathrm{alt}]}(t). \tag{11}$$

For $\alpha = 1$ we are in the pure-Gaussian case, of which we have seen that the inversion procedure performed well. We now vary α from 1 to 0, to see the impact of the non-Gaussianity.

The alternative input model that we choose here is an M/G/∞ input model, inspired by, e.g., [4]. In the M/G/∞ input model, jobs arrive according to a Poisson process. The job durations are i.i.d., and during their duration each job generates traffic at a constant rate r. In line with measurements studies, we choose Pareto(β) jobs, obeying the distribution function

$$F_D(x) = 1 - 1/(x+1)^{\beta}.$$

As the objective is to assess the impact of varying the parameter α, we have chosen to select the parameters of the M/G/∞ model such that the processes $A_{i_{[\mathrm{fBm}]}}$ and $A_{i_{[\mathrm{alt}]}}$ are 'compatible', in that their mean M and variance $V(\cdot)$ are similar. This has been done as follows.

- The mean of the described M/G/∞ input model is given by $M_{[\mathrm{alt}]} = \lambda r/(\beta - 1)$. We choose $\lambda = 10$ and $r = 1$ for ease; the other involved parameters will be used to achieve the desired 'compatibility'. The mean of the original fBm traffic model is $M_{[\mathrm{fBm}]} = 0$; however, as argued earlier, we may add a drift of $M_{[\mathrm{alt}]}$ to the $A_{i_{[\mathrm{fBm}]}}$-values to ensure that $M_{[\mathrm{alt}]} = M_{[\mathrm{fBm}]}$.
- In earlier work, see e.g. [11], an exact formula for the variance curve $V(t)_{[\mathrm{alt}]}$ has been derived. It is not possible to achieve the desired 'compatibility' of the variance on all time scales. As long-range dependence is mainly a property of long time-scales, we choose to focus on these. For larger time scales, the variance $V(t)$ from [11] roughly looks like, assuming $\beta \in (1,2)$,

  $$V_{[\mathrm{alt}]}(t) \approx r^2 \lambda \frac{2}{(3-\beta)(2-\beta)(\beta-1)} t^{3-\beta}.$$

  The variance of the original fBm traffic model is also known: $V_{[\mathrm{fBm}]}(t) = \sigma^2 t^{2H}$. Assuming $H = 0.7$ and sufficiently large t, we can now determine the remaining parameters: $\beta \approx 1.6$, $\sigma \approx 7.72$, and $M \approx 16.67$. The variance function of the traffic mixture is now also known, due to (11).

The next step is to run, for different values of α, the simulation, and to determine the theoretical variance curve of (11), and the execute inversion procedure of Algorithm 4.1.

In Fig. 7 we focus on the 'nearly-Gaussian' cases $\alpha = 0.8$ and $\alpha = 0.9$, which are plotted together with their theoretical counterparts. The figure shows that the presence of non-Gaussian traffic has some, but no crucial impact on our inversion procedure.

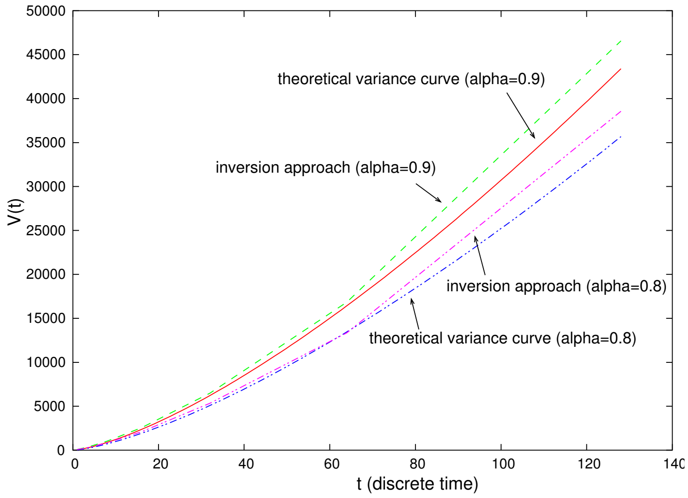

Figure 7: Variance curves for Gaussian/non-Gaussian traffic mixtures, $\alpha = 0.9$ and $\alpha = 0.8$

We also consider the (extreme) case of $\alpha = 0$, i.e., no Gaussian traffic at all, to see if our inversion procedure still works. In Fig. 8 the various variance curves are shown: the theoretical curve, the curve based on the 'direct approach', as well as the curve based on the inversion approach. Although not a perfect fit, the curves look similar and still relatively close to each other (but, of course, the fit is worse than for $\alpha = 0.8$ and 0.9). Note that the non-Gaussian traffic may 'have some Gaussian characteristics' if there is a large degree of aggregation, by virtue of central-limit type of arguments, which may explain that the fit is still reasonable.

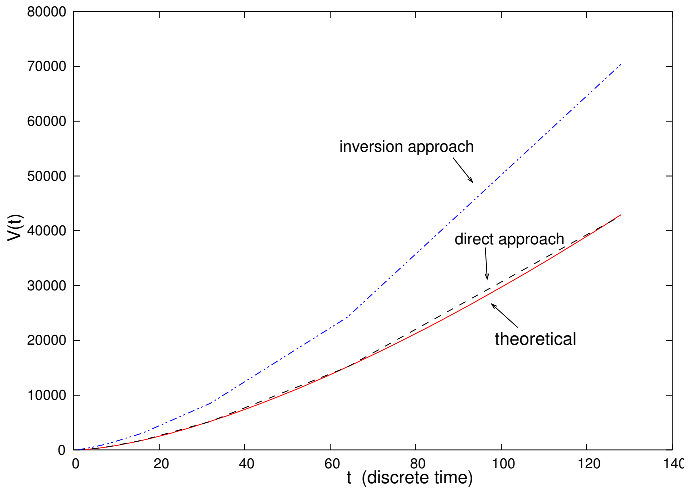

Figure 8: Variance curves for Gaussian/non-Gaussian traffic mixture, $\alpha = 0$

We conclude that our simulation experiments show the 'robustness' of the inversion procedure. Despite the approximations involved, with a relatively low measurement effort, the variance curve is estimated accurately, even for traffic that is not perfectly Gaussian. Given the evident advantages of the inversion approach over the 'direct approach' (minimal measurement effort required, retrieval of the entire variance curve $V(\cdot)$, etc. – see the discussion in Section III), the former method is to be preferred. In the next section we verify whether this conclusion also holds for real (i.e., not artificially generated) network traffic.

6 Empirical validation of the inversion procedure

Whereas the previous sections studied the performance of our inversion approach by executing simulation experiments with synthetic traffic, in this section we use traces of real network traffic. These traces are collected from different networking environments (i.e., link speeds, traffic aggregation level, type of users), detailed in Section VI.A. As we do not have the theoretical variance curve here, we evaluate the inversion approach by comparing with the 'direct approach'; this is done in Section VI.B.

6.1 Measurement and simulation setup

The traces used here are collected at the so-called (Ethernet) uplinks of various networks (i.e., links that connect these networks to their Internet service providers).

For each network, we have hooked up an off-the-shelf PC (for specifications, see Table 1) to a router/switch that copies all traffic from/to the uplink to the measurement PC. Using the standard tcpdump software [12], the headers (i.e., the first 64 octets of all Ethernet frames) of all packets are captured and subsequently made anonymous through the tcpdpriv tool [13] to protect the users' privacy. When time-stamping the packet headers a precision of at least 10 ms can be guaranteed, see [8]. In this way we have obtained a considerable number of traces, each of them containing 15 minutes of network traffic.

Table 1: Measurement PC Configuration

Component	Specification
CPU	Pentium-III 1 GHz
Mainboard	Asus CUR-DLS (64 bit 66 MHz PCI)
Hard disk	60 + 160 Gigabyte, UDMA/66
Operating system	Debian Linux, 2.4 kernel
Network interface	1 x Gbit/s Intel Pro/1000T
Main memory	512 MB reg. SDRAM

The collected packet traces are used in the following two ways:

1. to estimate the variance curve through the 'direct approach', see Section III; and
2. to 'replay' the traffic, submitting the packets to a buffer and output link, and take snapshots of the buffer content. Then the inversion approach is used to estimate the variance curve.

First we give an overview of the networks considered, see Table 2. We have deliberately selected networks that have different configurations and user bases, in order to validate the inversion approach for a broad range of situations.

Table 2: Overview of networks

Loc.	Access	Uplink	#Users	Type
I	FastEthernet	1 Gbps	200	researchers
II	ADSL	1 Gbps	400	mostly students
III	FastEthernet	30 Mbps	50	web-servers

6.2 Empirical validation

We have applied the procedure sketched in the previous subsection to three sample traces taken from the networks listed in Table 2.

We have set the 'sampling interval' τ (used to estimate the BCD) to 1 second, to ensure that a considerable number of snapshots (900) can be taken. A substantial fraction of these snapshots, also depending on the value of the output link's capacity, do not provide any information as the buffer turns out to be empty at the time the snapshot is taken.

We choose the smallest interval length for which we compare the variance estimated through our inversion approach with the actual variance found through the 'direct approach', to be 5 ms (which is, in other words, 200 times as small as the interval we poll the buffer occupancy).

Figures 9, 10 and 11 show the variance curves for networks I, II and III, respectively. Clearly, the three graphs, corresponding to highly diverse network environments, demonstrate that the inversion approach is capable of adequately estimating $V(\cdot)$. Hence even for real network traces, of which for instance the Gaussian character is far from evident, our coarse-grained buffer polling approach suffices to estimate $V(\cdot)$. We recall that this implies that this eliminates the need for doing detailed, small-time-scale, traffic measurements. In particular, our results indicate that we can estimate the variance on the time-scale of 5 ms, just by polling the buffer content at a low frequency, without ever doing a traffic measurement on the 5 ms time scale.

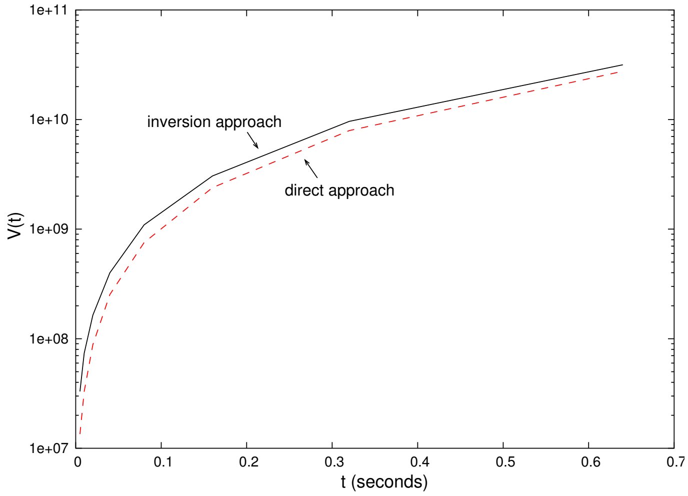

Figure 9: Variance curves, location I

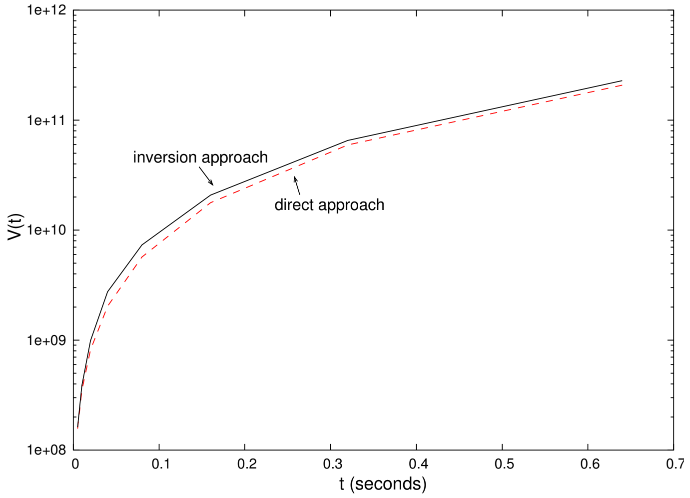

Figure 10: Variance curves, location II

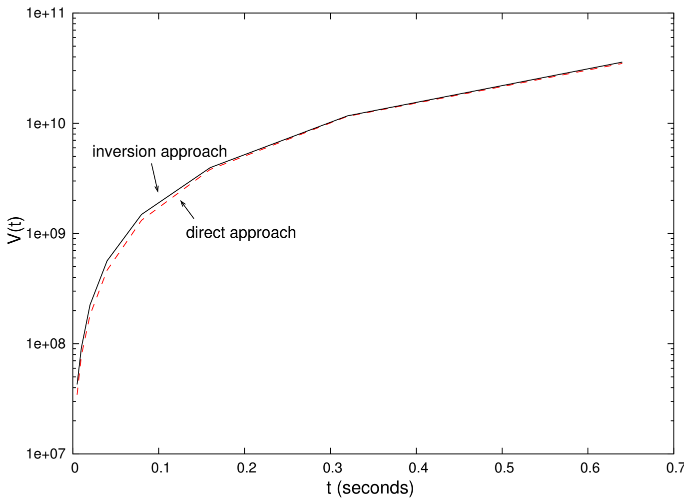

Figure 11: Variance curves, location III

7 Discussion

After the numerical tests of the previous sections, the next issue concerns the feasibility of implementing the inversion approach in operational environments; this is the subject of Section VII.A. We also comment on the interesting concept of *virtual queues*. In Section VII.B we compare our method with other 'cheap' measurement procedures that have been proposed in the literature.

7.1 Feasibility of the inversion approach

The above experiments (both with artificial traffic and real traces) have been done off-line, in that we have used scripts that 'parse' the synthetic and real traffic, mimicking the buffer content dynamics. An interesting question is whether this approach is feasible in run-time, in operational environments. From the proposed procedure, we can derive the following functional requirements for an implementation of the inversion procedure:

- a notion of the amount of data in a buffer;
- a way to regularly poll this information;
- software/hardware to determine the BCD, and then determine the resulting estimate of $V(\cdot)$.

To our best knowledge, these requirements do not lead to any fundamental or conceptual problems. The first requirement is already addressed: Random Early Detection (RED) queueing algorithms, which are widely implemented in modern routers, also keep track of the amount of queued data. In RED the buffer content (or, more precisely, a proxy of the buffer contents in the near past) is used to randomly discard packets (to which the TCP-users react by reducing their window size), see [14]. However, it is evident that information on the buffer occupancy can also be used for other purposes, such as the estimation of the BCD.

The second requirement may be fulfilled by, for instance, the use of SNMP (in case the entire procedure is not run on the router itself).

The last requirement has already been addressed – see our results in this paper. Hence, we conclude that there are no fundamental or conceptual problems preventing the actual application of our approach in practice.

Virtual queues. Interestingly, there is the possibility of *decoupling* the inversion procedure from the actual queue in the router. More precisely: in software, one could keep track of a 'virtual queue' that is drained at a link rate C' that might be different from the actual link rate C of the router. Particularly when the 'real queue' is empty during a substantial fraction of the time, which inevitably results in poor estimates of the BCD, one could better use a virtual queue that is drained at a lower rate C'. First experiments indicate that the choice of this C' has some impact in the performance of the inversion approach. A more detailed analysis is a subject for future research.

7.2 Alternative measurement procedures

The purpose of our inversion method is to retrieve the essential traffic characteristics with at a low measurement cost. We remark that several other 'cheap' (i.e., with low measurement effort) methods have been proposed. We now discuss some of these, and compare them with our approach.

The method described in Duffield *et al.* [15] aims to estimate the *asymptotic cumulant function*

$$\Lambda(\theta) := \lim_{t\to\infty} \frac{1}{t} \log \mathbb{E} e^{\theta A(t)}$$

from traffic measurements – this function is useful, because, under some assumptions on the traffic arrival process, it holds that $\log \mathbb{P}(Q > B) \approx -\theta^\star B$, for B large, where $\theta^\star$ solves $\Lambda(\theta) = C\theta$. The crucial assumption on the input traffic is that it should be short-range dependent – otherwise $\Lambda(\cdot)$ does not even exist (think of fBm, for which $\mathbb{E}\exp(\theta A(t))$ is of the form $\frac{1}{2}\theta^2 t^{2H}$). Of course, this requirement is quite restrictive. It is noted that the estimation of $\Lambda(\cdot)$ turns out to be far from straightforward (block-sizes need to be chosen, etc.). A crucial difference with our approach is that [15] measures *traffic*, whereas we propose to measure (or, better: to poll) the *buffer content*.

Another related study is by Kesidis *et al.* [16]. Like in our method, their approach relies on the estimation of the buffer content distribution $\log \mathbb{P}(Q > B)$. Under the assumption of short-range dependent input, $\log \mathbb{P}(Q > B)$ is linear for large B (with slope $-\theta^\star$). Having estimated $\theta^\star$ the probability of overflow over higher buffer levels can be estimated, by extrapolating $\log \mathbb{P}(Q > B)$ linearly. Also this method does not deal with long-range dependent input.

8 Concluding remarks

Summary: We have presented a novel method to determine the burstiness of network traffic; here burstiness is in terms of the variance $V(T)$ of the traffic generated in an arbitrary window of length T. Our approach estimates the entire variance curve $V(\cdot)$ (of course up to some horizon), also on small time-scales, *without performing detailed traffic measurements*. Instead, the buffer content is polled (at some coarse-grained frequency) to obtain an estimate of the buffer content distribution. Then this distribution is 'inverted' to find the variance curve of the traffic rates, which gives the 'burstiness' $V(T)$ of the network traffic *at any time scale* T (up to some horizon). Knowledge of the variance curve can immediately be used in provisioning formulae.

We have presented the mathematical foundations under this inversion method, and have investigated its accuracy by performing a thorough analysis of the possible sources of error. Furthermore, we have extensively validated our approach in various real-life settings. This has shown that our inversion method provides remarkably accurate estimates of the traffic's burstiness. In particular, we have observed that our approach yields reliable estimates of the variance for very small time-scales. A seemingly counterintuitive but representative example: by sampling from the buffer occupancy every second we have found an accurate estimate of the variance on the time-scale of 5 ms.

Future work: In Section VII.A, we have introduced the concept of virtual queues. In further work, we intend to investigate the impact of the precise choices of the virtual queue's link rate C' on our inversion approach. It would be interesting to optimize the C', i.e., to find the C' such that the fit is optimal (with respect to some optimality criterion). We also intend to extend the concept to a network setting.

Acknowledgments

Part of the research presented in this paper has been sponsored by the Dutch 'Telematica Instituut'. The authors are indebted to Aiko Pras (University of Twente) for valuable comments.

References

[1] Remco van de Meent, Aiko Pras, Michel Mandjes, Hans van den Berg, Frank Roijers, Pieter Venemans, and Lambert Nieuwenhuis, "Burstiness predictions based on rough network traffic measurements," To appear in *Proceedings WTC/ISC 2004*.

[2] J. Kilpi and I. Norros, "Testing the gaussian approximation of aggregate traffic," in *Proceedings Internet Measurement Workshop,* Marseille, France, 2002.

[3] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking*, vol. 1, no. 2, pp. 1–15, February 1994.

[4] R. Addie, P. Mannersalo, and I. Norros, "Most probable paths and performance formulae for buffers with gaussian input traffic.," *European Transactions on Telecommunications*, vol. 13, no. 3, pp. 183–196, 2002.

[5] Chuck Fraleigh, Fouad Tobagi, and Christophe Diot, "Provisioning IP Backbone Networks to Support Latency Sensitive Traffic," in *Proceedings of IEEE Infocom*, San Francisco, U.S.A., April 2003.

[6] K. Kumaran and M. Mandjes, "The buffer-bandwidth trade-off curve is convex," *Queueing Systems*, vol. 38, pp. 471–483, 2001.

[7] Jan Beran, *Statistics for Long-Memory Processes*, Chapman & Hall/CRC, 1994.

[8] Remco van de Meent, Aiko Pras, Michel Mandjes, Hans van den Berg, and Lambert Nieuwenhuis, "Traffic Measurements for Link Dimensioning: A Case Study," in *Proceedings of the 14th IFIP/IEEE Workshop on Distributed Systems: Operations and Management (DSOM2003)*, M. Brunner and A. Keller, Eds., October 2003, number 2867 in Lecture Notes in Computer Science (LNCS), pp. 106–117.

[9] M. Mandjes, "A note on the benefits of buffering," *Stochastic Models*, vol. 20, pp. 43–53, 2004.

[10] Ton Dieker, "Fractional Brownian motion simulator," `http://homepages.cwi.nl/~ton/fbm/index.html`.

[11] M. Mandjes, I. Saniee, and A. Stolyar, "Load characterization, overload prediction and load anomaly detection for voice over IP traffic," in *Proceedings 38th Allerton Conference*, 2000, pp. 567–576.

[12] Lawrence Berkeley National Laboratory Network Research, "TCPDump: the Protocol Packet Capture and Dumper Program," 2003, `http://www.tcpdump.org/`.

[13] Ipsilon Networks, "tcpdpriv," 1997, `http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html`.

[14] Sally Floyd and Van Jacobson, "Random Early Detection (RED) gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, August 1993.

[15] N.G. Duffield, J.T. Lewis, N. O'Connell, R. Russell, and F. Toomey, "Entropy of ATM traffic streams: a tool for estimating quality of service parameters," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 981–990, 1995.

[16] C. Courcoubetis, G. Kesidis, A. Ridder, J. Walrand, and R. Weber, "Admission control and routing in ATM networks using inferences from measured buffered occupancy," *IEEE Transactions on Communications*, vol. 43, pp. 1778–1784, 1995.