Estimating Link Capacity and Buffer Size Using Traffic Losses

Jun Liu
Computer Science Department
University of North Dakota
Grand Forks, ND 58202
jliu@cs.und.edu

Abstract

Accurate estimation of network characteristics based on endpoint measurements is a challenging and important problem. In this paper we consider a simple problem that has application to network discovery: given a FIFO queue with finite buffer size and constant service rate, estimate the buffer size and service rate given two sources of data: 1) traffic arriving at the queue; and 2) traffic lost due to buffer overflow. We describe an estimation method to solve this problem based on searching for particular events in the evolution of the queueing system. The events we look for are those in which the queue starts empty, and fills to overflow without ever decreasing. In this case queue properties can be inferred easily. Our central observation is that these events are most likely at a particular timescale, and our method is based on searching the traffic arrival and loss patterns at the critical timescale. Under the assumption that arriving traffic is fractional Gaussian noise, we show that the method is likely to work well. Using both open loop and closed (`ns`) simulations, we show that the method appears to be accurate and efficient.

Jun Liu
Computer Science Department
University of North Dakota
Grand Forks, ND 58202
jliu@cs.und.edu

1 Introduction

Methods for discovering network-internal characteristics using measurements taken at endpoints are increasingly valuable as applications and services seek to adapt to network properties. In this paper, we consider a basic problem that has application to the discovery of network properties. We are concerned with the following estimation problem concerning a FIFO queue with finite buffer size B and constant service rate C: assume that it is possible to observe the traffic arriving to the queue as well as the traffic lost due to buffer overflow; how can one then estimate B and C?

This simple model is interesting because in some networking settings it may be possible to observe (or estimate) traffic arriving at a link; furthermore, loss information may be available due to feedback mechanisms in a reliable transport like TCP. A particular motivating example is a very busy server (like a Web server) that is generating all or most of the traffic flowing over some network link; in this case (approximate) knowledge of both offered traffic and lost traffic may be available to the server.

The approach we take in this paper is to observe the traffic and loss streams over time, hoping to encounter a time interval in which B and C can be estimated accurately. We note that B and C can be estimated accurately when the following event occurs: at the beginning of some interval (s, t) the queue is empty, and some loss occurs during the interval; furthermore queue occupancy is non-decreasing in the interval. Assuming we can identify two such intervals (of different length), than it is straightforward to estimate B and C from the values of the arrival and loss streams over the intervals.

Of course, the difficulty is in knowing during which intervals (s, t) this queueing event has occurred. A starting point is to realize that for any interval of length $l = t - s$, a simple maximization procedure will identify the right interval if one exists (*i.e.*, of length l). The next step is the key observation in our method: we note that there is a particular interval of length $l = t^*$ (called the *critical timescale* of the queueing system) at which such an event is *most likely* to occur. The critical timescale is well-studied for the case in which arriving traffic is fractional Gaussian noise, and our theoretical results focus on that case as well.

Thus, our estimation method consists of looking for rare events in the evolution of the queueing system. Our central observation is that although these events are rare, they are most likely to occur at the critical timescale. We treat the question of how to estimate the critical timescale from arrival and loss data as a separate issue; we have developed solution methods for that problem and describe them in a companion paper [14]. In this paper we show how to use knowledge of critical timescale to construct an estimation procedure for C and B.

Our estimation procedure must necessarily examine the traffic and loss data at multiple timescales and at multiple locations (points in time). To make this efficient we sample the data on the dyadic grid, which is a exponentially-spaced set of points in the frequency-time domain. This is the same method used in the discrete wavelet transform to simultaneously analyze datasets in frequency and location.

This paper describes the theoretical foundation for our approach, as well as results of applying in both open-loop and closed-loop simulations. In summary, we find that:

- Our estimation procedure is computationally efficient;

- The actual estimation results are accurate; and
- Our estimation procedure is provably effective assuming the arrival traffic to the queue can be modeled as a *fractional Gaussian noise* process.

The reminder of this paper is organized as follows: Section **??** describes background and the notation; Section 2 more formally motivates our estimation procedure; Section 3 describes our main theoretical results on which our analysis and estimation procedure build; Section 4 describes the whole procedure; Section 5 describes the simulation used for testing our estimation procedure and results, suggesting certain limitations of our estimation procedure; Section 6 describes related work; and Section 7 concludes and summarizes our contribution.

2 Motivation

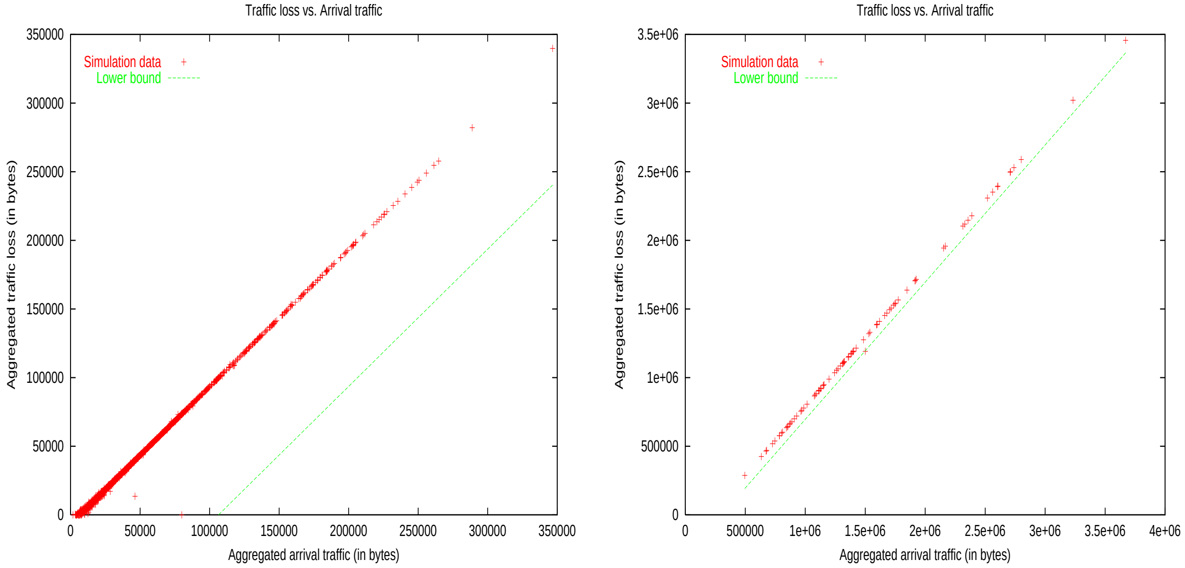
Our estimation approach relies on knowledge of traffic dropped due to limited buffer space. We assume that the only traffic dropped is the excess traffic that arrives when the buffer is full. The dropped traffic forms another stochastic process $\{L_t : t \geq 0\}$ where L_t denotes the *aggregated* traffic dropped by the queue in time interval $[0, t)$ and $L(s, t) = L_t - L_s$. We assume that $L_0 = 0$.
The starting point for our method is the observation that there is a lower bound on $L(s, t)$ as follows:

Proposition 2.1 For any interval $[s, t)$ $(0 \leq s < t)$, we have

$$L(s,t) \geq [A(s,t) - C \cdot (t - s) - B]^+$$

□

This can be seen as follows. Consider the case in which the queue is empty at time s, is full at time t, and arriving traffic exceeds the service rate throughout $(s, t]$ (*i.e.*, $X(u, u') > 0$ for $s \leq u < u' \leq t$)). Then the queue absorbs $C \cdot (t - s) + B$ traffic and the rest is lost; this is the equality situation. Now, if the queue is not empty at s, the amount of lost traffic increases; if the queue occupancy decreases at any time, the amount of lost traffic decreases; and if the queue does not reach B then both sides of the inquality must be zero.
As a concrete example to show this inequality, in Figure 1 we plot the aggregated amount of traffic loss (samples of $L(s, t)$) versus the aggregated amount of arrival traffic (samples of $A(s, t)$) for a simple queue simulation fed by a standard trace of WAN TCP traffic. On the left we show plots of L vs. A for samples of duration 1.28 s (*i.e.*, $t - s$ = 1.28 s); on the right we show plots for samples of duration 40.96 seconds. In addition to illustrating the inequality, the plots show that the lower bound is more frequently attained at the longer timescale; in other words, the likelihood of attaining the lower bound in practice is sensitive to the timescale of study.
We can make use of this lower bound to solve for B and C if we know that the lower bound on traffic loss is attained for two differently sized time intervals $[s_1, t_1)$ and $[s_2, t_2)$.



Time scale is 1.28 sec

Time scale is 40.96 seconds

Figure 1: Plots of aggregated traffic vs. aggregated loss for different time scales in a simulated queue with B = 100,000 bytes and C = 5000 bytes/sec. The trace used is `lbl-tcp-4` from the Internet Traffic Archives, `http://ita.ee.lbl.gov`.

We denote $a^{(k)}_{[s,t)}$ as one of the sample paths of process $A(s,t)$ where k is the index of this sample path; as well we denote $l^{(k)}_{[s,t)}$ as the amount of traffic loss when this sample path is fed to the queueing system. Suppose that the lower bound on traffic loss is attained on two differently sized time intervals $[s_1, t_1)$ and $[s_2, t_2)$ $(t_1 - s_1 \neq t_2 - s_2)$, *i.e.*,

$$l^{(k_1)}_{[s_1,t_1)} = a^{(k_1)}_{[s_1,t_1)} - C \cdot (t_1 - s_1) - B \qquad (1)$$

$$l^{(k_2)}_{[s_2,t_2)} = a^{(k_2)}_{[s_2,t_2)} - C \cdot (t_2 - s_2) - B. \qquad (2)$$

In this case we can solve for C and B using (1) and (2) as

$$\left\{ \begin{array}{lcl} C & = & \frac{[a^{(k_1)}_{[s_1,t_1)} - l^{(k_1)}_{[s_1,t_1)}] - [a^{(k_2)}_{[s_2,t_2)} - l^{(k_2)}_{[s_2,t_2)}]}{(t_1 - s_1) - (t_2 - s_2)} \\ B & = & \frac{(t_2 - s_2)\cdot[a^{(k_1)}_{[s_1,t_1)} - l^{(k_1)}_{[s_1,t_1)}] - (t_1 - s_1)\cdot[a^{(k_2)}_{[s_2,t_2)} - l^{(k_2)}_{[s_2,t_2)}]}{(t_1 - s_1) - (t_2 - s_2)} \end{array} \right. \qquad (3)$$

3 Theoretical Results

Equations (3) show that we can get an accurate estimation of B and C when it happens that the lower bound on traffic loss is attained (for two different intervals). This focuses the estimation problem on the question of how to identify intervals over which Equations (1) and (2) hold.

In order for (one of) Equations (1) or (2) to hold for a particular interval, the following conditions must be met:

1. The queue must be empty at the start of the interval;
2. Queue occupancy must be non-decreasing throughout the interval; and
3. Traffic loss must occur within the interval.

Note that conditions 2 and 3 together imply that traffic loss must be in progress at the end of the interval.

Our estimation method is intended to discover time intervals during which the queueing system meets these three conditions. To do this, we make the assumption that arriving traffic can be described as fractional Gaussian noise. Then, we address these conditions as follows: we show in this section that these three conditions are most likely to occur at the critical timescale, *i.e.*, the timescale that maximizes $\Pr\{A_t > B + C \cdot t\}$. As described in Section **??**, the critical timescale is a function of B, C, and traffic properties. The key observation is that time intervals meeting these three conditions are relatively rare events; but, they are most probable at the critical timescale. For this reason we use multiscale sampling on the dyadic grid (explained in the next section) to efficiently search for candidate intervals at the critical timescale.

In the remainder of this section we show why it is reasonable to expect that queue occupancy during a period of queue formation is non-decreasing when B is large, and why this means that the three conditions are most likely to be met at the critical timescale. In the next section we describe how we search at the critical timescale for intervals that meet the three conditions.

Our theoretical results are derived for the unbounded queue. In the unbounded queue, there is no traffic loss; instead we denote $U(t) = [Q(t) - B]^+$ as the "excess" queue occupancy exceeding the threshold B. Our main analytical result in this section is that the two events $\{U(t) = A_t - C \cdot t - B\}$ and $\{A_t - C \cdot t - B > 0\}$ are *almost surely* equivalent when B goes to infinity. To prove this result, we first note the following basic fact about queue occupancy in the unbounded queue:

$$Q(t) \quad = \quad \sup_{0 \le s \le t} [A(s,t) - C \cdot (t-s)] \tag{4}$$

This supremum formula for queue occupancy is used extensively and proved, *e.g.,* in [3, 6, 8].

To prove our main results, we first need to propose the following lemma:

Lemma 3.1 (Nondecreasing queue occupancy.) In an interval $[0, v)$, let A be a fractional Brownian traffic process, and let

$$u = \arg_w \sup_{0 \le w < v} [A(w,v) - C \cdot (v-w)]$$

Then $\forall s, t \ (0 < u \le s < t \le v)$, we have

$$\lim_{B \to \infty} \Pr\{X(s,t) > 0 | Q(v) > B\} = 1$$

□

(The proofs of the lemma and theorem appear in the extended version of this paper.)
Using Lemma 3.1, we can develop the following theorem (using the same u and v as before):

Theorem 3.2 If $U(v) > 0$, we have

$$\lim_{B \to \infty} \Pr\{U(v) = A(u,v) - C \cdot (v-u) - B | \\ A(u,v) - C \cdot (v-u) - B > 0\} = 1$$

□

Thus we see that when B is large, we can maximize the probability that $U(v) = A(u,v) - C \cdot (v-u) - B$ where $U(v) > 0$ by instead maximizing the probability that $A(u,v) - C \cdot (v-u) - B > 0$. Noting that $Q(u) = 0$, this is equivalent to maximizing $\Pr\{A_t - C \cdot t - B > 0\}$. As shown in [8], this probability is maximized at t^* when $\{A_t\}$ is a fractional Brownian traffic process.
Inspired by the fact that arrival traffic rate is consistently bigger than the service rate, we can get an corollary of 3.2 as follows, when the queue is a bounded one,

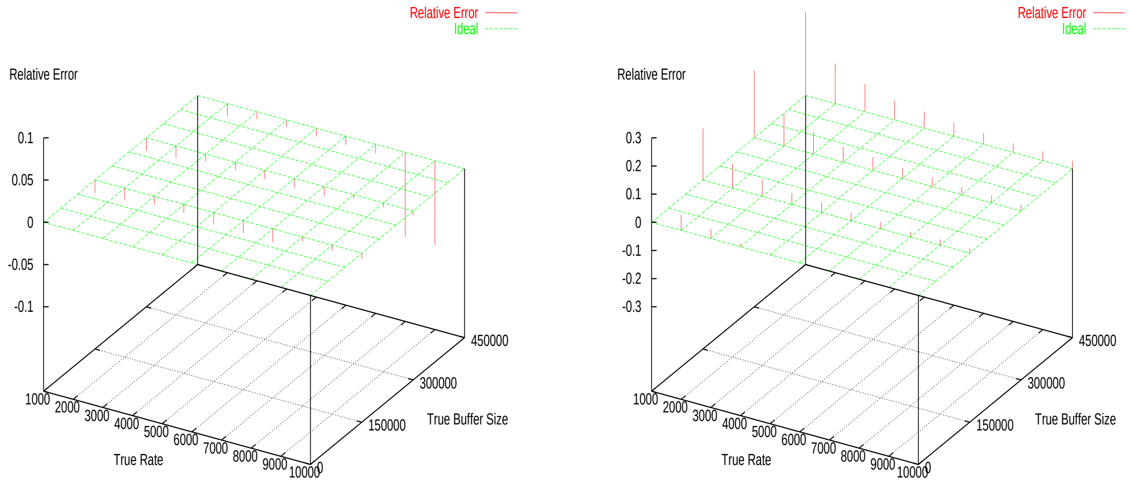
Corollary 3.3 In an interval $[u, v)$, if $\forall s, t : 0 \leq u \leq s < t \leq v, X(s,t) > 0$ and $L(u,v) > 0$, then

$$\frac{\Pr\{L(u,v) = A(u,v) - C \cdot (v-u) - B\}}{\Pr\{A(u,v) - C \cdot (v-u) - B > 0\}} = 1$$

□

4 Estimation Procedure

Our estimation procedure must process the traffic and loss measurements at multiple timescales; mainly this is because the procedure for estimating t^* (not part of this paper; described in [14]) works by inspecting multiple timescales. In addition, the measurements must be sampled at each timescale so as to search for intervals meeting the three criteria of Section 3. To do this efficiently, we analyze measurement data in the style of a *Multi-Resolution Analysis (MRA)* [15]. Like an MRA, our estimation procedure works on the dyadic grid. The dyadic grid is a set of time scales with sizes growing exponentially: $2^0 \cdot \tau, 2^1 \cdot \tau, \cdots, 2^i \cdot \tau, ...$ $(i \in \mathbb{N})$ where τ is the smallest time interval used in the procedure and the largest value of i is determined by the length of the trace. In all the results we present here, we set τ to be 10 millisecond.
The actual measurements of the aggregated traffic process $\{A_t\}$ and of the aggregated traffic loss process $\{L_t\}$ are discrete time series, rather than continuous time processes. These discrete time series are converted into counting series of different counting intervals in the dyadic grid. To obtain the counting series for a counting interval $\mu = 2^j \cdot \tau$, we divide the whole series into intervals of size μ in time, then in each interval, the values of all items are aggregated into one value. We denote $\{a_i(k) : k \geq 0\}$ and $\{l_i(k) : k \geq 0\}$ as the counting series of $\{A_t\}$ and $\{L_t\}$ with counting interval $2^i \cdot \tau$, respectively. For any



Buffer size estimation

Service rate estimation

Figure 2: Relative error in estimation results for different settings of true buffer sizes and service rates in the open system simulation using the `lbl-tcp-4` trace.

i, both $\{a_i(k) : k \geq 0\}$ and $\{l_i(k) : k \geq 0\}$ have exactly the same number of items and there is a 1-to-1 mapping between the two counting series based on k. Thus each item in the traffic counting series of interval μ (or traffic loss series) represents the total amount of arrival traffic in different time interval μ's (or total amount of traffic loss). The entire summary of $\{A_t\}$ and $\{L_t\}$ on these different counting intervals in the dyadic grid are fed as input to the estimation procedure. We denote $\hat{B}$ and $\hat{C}$ as the resulting estimations of B and C.

The advantage of sampling on the dyadic grid is that it can be conceptually implemented as a set of filter banks. If it is desired to search at m levels (where m is bounded by the binary logarithm of the trace length), then m filter banks can implement the sampling process. The banks are arranged in a linear array and each bank adds successive pairs of inputs and sends the sum to the next bank in line. This organization leads to an efficient algorithm that allows values at all levels to be computed in one pass over the trace (for any fixed m).

As described in Section 2, the accuracy of our estimation procedure depends on encountering an interval that obtains the lower bound on traffic loss. As described in the previous section, we are most likely to encounter such an event at the critical timescale. Supposing that the lower bound on traffic loss is most likely attained at the dyadic time scale j, then we form our estimates of B and C using the two pairs of counting series $\{a_{(j-1)}(k)\}$, $\{l_{(j-1)}(k)\}$ and $\{a_j(k)\}$, $\{l_j(k)\}$ by the following equations:

$$\begin{cases} C = \frac{M(j)-M(j-1)}{2^{(j-1)} \cdot \tau} \\ B = \frac{2^{(j-1)} \cdot M(j) - 2^j \cdot M(j-1)}{2^{(j-1)}} \end{cases} \qquad (5)$$

where

$$M(j-1) = \max_k \{a_{(j-1)}(k) - l_{(j-1)}(k)\}$$

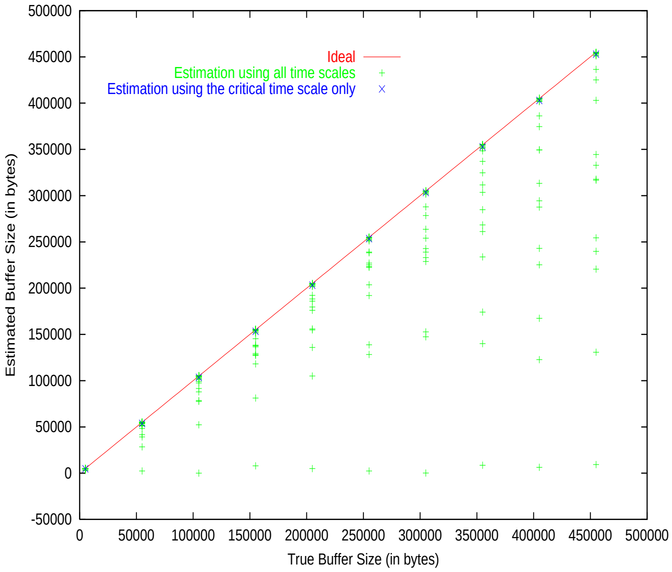


Figure 3: Estimation results for buffer size using all time scales. Estimation based on critical timescale t^* is marked with a '$\times$'.

and

$$M(j) = \max_k \{a_j(k) - l_j(k)\}.$$

Note the role of the $\max_k$ operation in this procedure. This operation has the effect of searching all intervals at the given level and selecting the interval with the largest difference between arriving traffic and lost traffic. By the arguments presented for Proposition 2.1, this method is certain to find an interval of interest (one that attains the lower bound) *if* such an interval exists among those at this level. If it happens that such an interval does not exist, it will find the interval in which the lower bound is closest to being attained.

So, to put our method in a nutshell, it is: choose the most likely timescale for the lower bound to be attained; and find the interval within that timescale that comes closest to attaining the lower bound. To illustrate this concept, in Figure 3 we have plotted the estimates produced by our method for various timescales. (This figure is based on our ns simulations described in the next section.) Along the x-axis we have varied the true buffer size and along the y-axis we plot the estimated buffer size obtained by applying our algorithm to different timescales. For each buffer size, there is a single timescale (or a small set of timescales) at which the estimation is accurate. This illustrates the importance of using the critical timescale as the principal basis for our method.

5 Evaluation

To test the accuracy of our estimation procedure, we tested it in two settings. First, we fed commonly-used traffic traces into a simulated queue; these were our *open* system simulations. These simulations allowed us to observe the effects of changing B and C for a fixed, known traffic input that reflects actual traffic captured from a operating network.

The trace we used was the `lbl-tcp-4` traces, available at the Internet Traffic Archive (`ita.ee.lbl.gov`). This is a trace of TCP connections flowing over a link to a wide area network. (We also tested our estimator on other traces from the same archive; results were essentially the same as for `lbl-tcp-4`.)

In the open system simulation, we fed the trace to a simple queue simulation to obtain the corresponding traffic loss trace. We then used the traffic trace and the corresponding traffic loss trace in our estimation procedure to form estimates for C and B. The relative errors in the estimation results are shown in Figure 2. In almost all cases, the resulting relative errors in buffer size estimation are less than 5%.

In the case of service rate estimation, relative error is also ususally small (less than 10%), but it is also clear from the Figure that the quality of the estimate declines as the ratio of buffer size to service rate increases. This effect can be understood as follows. As B/C increases, the critical timescale of the system (t^*) increases as well. This means that the aggregation level j used by the estimator increases; in any fixed length trace this means that there are fewer opportunities to search for intervals of interest. This is because the computational efficiency of the dyadic grid derives from its use of non-overlapping intervals at any particular level. This evidence suggests that the accuracy of the estimator could be improved at a certain performance price by searching the critical timescale in overlapping intervals.

Although the open system simulation controls the nature of the traffic arriving at the queue across different values of B and C, it may not be very representative of the behavior of a real network. This is because traffic loss in a network using TCP feeds back to the sources and affects the sources' sending rates. In effect, TCP sources always are attempting to fill without exceeding the capacity of their network paths. This means that an estimation procedure may have a very different sort of loss pattern to work with in the closed system as compared to the open system. For this second, closed system we used the `NS-2` simulator, available at `www.isi.edu/nsnam/ns`.

In this simulation, the network was configured in a dumb-bell configuration; there were 1000 pairs of TCP "clients" and "servers" which all used one bottleneck link as illustrated in Figure 4.

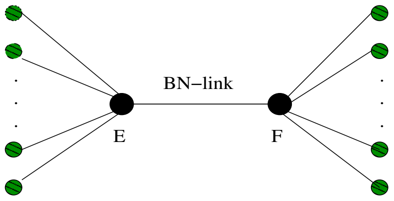
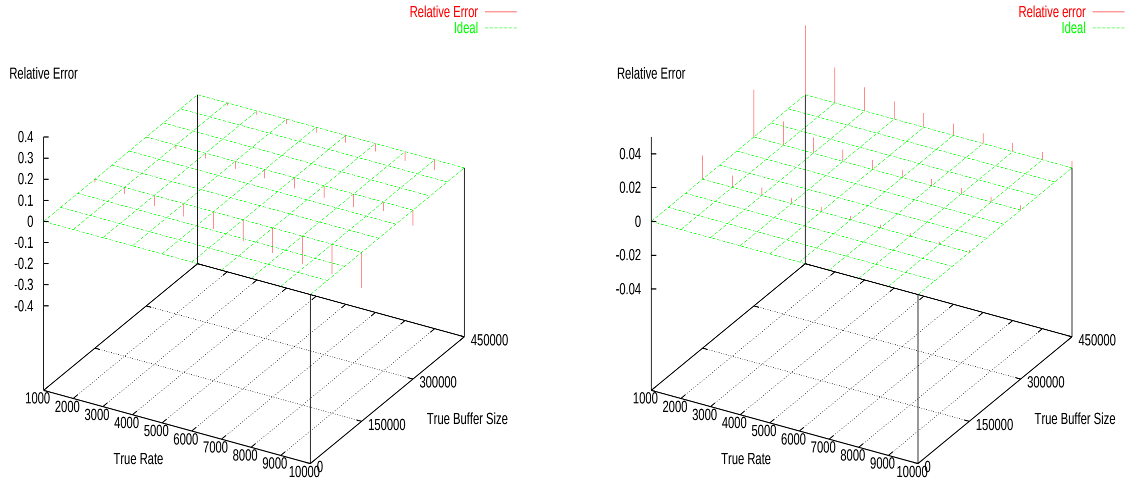


Figure 4: Dumb-bell configuration in NS simulator. E and F are routers, the end-points are `Ftp` server/client pairs that go through the bottle-neck link.

Each client alternates between making requests and lying idle for some period of time. Request sizes (in bytes) are drawn from a heavy-tailed distribution, as are the lengths of idle times. This method has been shown to be sufficient to generate self-similar traffic [16]. The queueing method is drop-tail, which agrees with our assumptions about how loss occurs. The bottleneck link rate and the outgoing buffer size of the bottleneck link are

varied to study the effectiveness of the estimation method.
Each simulation runs for 20,000 seconds. The output of the simulation is a trace file which records the queueing activities of one direction at the router of the bottleneck link, say the direction is from E to F at router E. From this trace file, we can extract the actual arrival traffic series to router E and the traffic loss series by selecting the dropped bytes by router E. We compute the counting series of the arrival traffic series and the traffic loss series for counting intervals $2^j \cdot \tau$ with $j = 0$ to 18. The smallest counting interval is $\tau = 0.01$ second. (Thus the range of j is governed by the simulation duration.) The actual estimation is then performed on the two sets of counting series.
Figure 5 shows the relative error in the estimations. For a wide range of B and C, we can see from Figure that the estimation results are reasonably accurate (less than 20% relative error). Buffer size estimation is less accurate than in the open system, but service rate estimation is more so. As in the open simulation, the quality of service rate estimation declines when t^* is large and therefore the number of samples available for estimation get smaller.



Buffer size estimation

Service rate estimation

Figure 5: Estimation results on buffer size and service rate for different settings of buffer sizes and service rates when $H = 0.75$ in the closed loop simulation by NS-2 simulator. The increments on buffer size is 150000 bytes and 1000 bytes/s on service rate.

A further effect is visible in Figure 5. As the ratio of B/C gets smaller (so that t^* get smaller) the accurate of buffer size estimation declines. This can be understood because there is a lower limit (τ) on the scale at which our method examines data. As t^* gets closer to the lower limit, the potential inaccuracy due to insufficient resolution increases.
Thus, our method has certain limitations:

- The estimation when t^* is large uses the low frequency part of the arrival traffic, so the duration of the collected arrival traffic should be long enough for accuracy.

- In our estimation, we make use of the traffic loss series with respect to the arrival traffic series. The traffic loss series should contain some items that are bigger than zero, *i.e.*, buffer overflow should happen and should not be too rare.
- The duration of the simulation is related to both buffer size B and C due to equation (**??**). That means that the duration of the simulation restricts the range of B and C that can be accurately discovered.

Nonetheless, the results in this section are encouraging, and the ns simulations in particular suggest that our method may be effective in situations where traffic is generated by multiple TCP flows travelling over a common bottleneck link.

6 Related Work

There are a number of papers are devoted to study the loss behavior in the Internet and that have proposed various methods to characterize features of Internet load or to measure statistics in the Internet.

Bolot [11] proposed a “packet-pair” method to study end-to-end packet delay and loss behavior in the Internet; this method allows discovery of bottleneck bandwidth but not buffer capacity.

R. Cáceres, et. al., [12, 13], worked out an estimator based on MLE method to estimate traffic loss probability in multicasting environment and further estimate the topology of the multicast tree based on traffic loss probability.

I. Norros [3] proposed the *fractional Brownian traffic* model to model the aggregated traffic generated by sufficiently large number of ON/OFF sources whose ON- and OFF- periods follow heavy-tailed distributions.

J. Choe and N.B. Shroff [6] studied the supremum distribution of a Gaussian process having stationary increments. They also studied the queueing impact with such Gaussian process as input to the queue. They extensively studied the property of the queue tail distribution.

Arnold L. Neidhardt and Jonathan L. Wang [8] studied the queueing performance on *fractional Brownian traffic* process. They explicitly claimed that the time scale is crucial to queueing performance study.

The concept of ”most relevant” time scale is discussed in both [6] and [8].

7 Conclusion

In this paper, we have described a method for estimating the link capacity C and buffer size B in a single server queueing system. We assume that the arriving traffic and lost traffic are observable; our approach is to observe the system over time watching for particular queueing events to occur. While these events are rare, our central observation is that they are most likely at a particular timescale. As a result we examine the traffic and loss processes at that timescale, using a method that is certain to find such an event should it occur at that timescale. We support our approach with analysis suggesting why the approach is likely to be effective. We evaluate the method in both open-system and closed-system settings and we characterize the situations in which it is likely to be less accurate, while finding that it is reasonably accurate in general.

References

[1] W. Feller, ”A Introduction to Probability Theory and Its Applications” third edition, Vol. I, 1968. *John Wiley & Sons, Inc*, 1976.

[2] L. Kleinrock, ”Queueing Systems” Vol. II: Computer Applications. *John Wiley & Sons, Inc*, 1976.

[3] I. Norros, ”On the use of Fractional Brownian Motion in the theory of connectionless networks”. *IEEE Journal on Selected Areas in Communications,* vol. 13, no. 6, August, 1995.

[4] N.G. Duffield and Neil O’Connell, ”Large deviations and overflow probabilities for the general single server queue, with applications”. *Math. Proc. Cam. Phil. Soc.*, 118, 363-374 (1995).

[5] J. Choe and N.B. Shroff, ”Use of supremum distribution of Gaussian processes in queueing analysis with long-range dependence and self-similarity”. submitted to *Stochastic Models*, 1997.

[6] J. Choe and N.B. Shroff, ”Queueing analysis of high-speed multiplexers including long-range dependent arrival processes” *proceedings of IEEE Infocom*, 1999.

[7] W. E. Leland, M. S. Taqqu, W. Willinger, D. V. Wilson, ”On the self-similar nature of Ethernet traffic (extended version)”. *IEEE/ACM Trans. Networking*, vol. 2, no. 1, pp. 1-15, Feb, 1994.

[8] Arnold L. Neidhardt, Jonathan L. Wang, ”Its application to queuing analysis of self-similar traffic” *proceedings of ACM SIGMETRICS*, 1998

[9] John Trevor Lewis, Raymond Russel, ”An introduction to large deviation” *Performance Tutorials, IFIP,* 1996

[10] A. Erramilli, O. Narayan and W. Willinger, “Experimental Studies on the Performance Impacts of Long Range Dependence” *IEEE/ACM Trans. on Networking* 4, 209, 1996

[11] Jean-Chrysostome Bolot, ”End-to-End Packet Delay and Loss Behavior in the Internet” *ACM SIGCOMM*, 1993

[12] R. Cáceres, N.G. Duffield, J. Horowitz and D. Towsley, ”Multicast-Based Inference of Network-Internal Loss Characterization” *IEEE Transactions on Information Theory, vol. 45, No. 7, pp. 2462 - 2480,* Nov. 1999.

[13] R. Cáceres, N.G. Duffield, J. Horowitz, F. Lo Presti, D. Towsley, ”Loss-based Inference of Multicast Network Topology”, *Proceedings of CDC’99*

[14] J. Liu and M. E. Crovella, “Estimating Critical Timescale from Traffic Loss”, *in preparation.*

[15] G. Kaiser, “A Friendly Guide to Wavelets,” Birkhäuser, 1994.

[16] Kihong Park, Gi Tae Kim and Mark E. Crovella, “On the Relationship Between File Sizes, Transport Protocols, and Self-Similar Network Traffic”, in *Proceedings of the Fourth International Conference on Network Protocols (ICNP’96)*, pp. 171-180, October 1996.