Modeling TCP in Small-Buffer Networks

Mark Shifrin and Isaac Keslassy

Department of Electrical Engineering
Technion - Israel Institute of Technology
Haifa 32000, Israel
{logout@tx,isaac@ee}.technion.ac.il

Abstract. Today, the large packet buffers present in backbone routers significantly increase their power consumption and design time. Recent models of networks with large buffers have suggested that these large buffers could be replaced with much smaller ones. Unfortunately, it turns out that these models are not valid anymore in networks with small buffers, and therefore cannot predict how these small-buffer networks will behave. In this paper, we introduce a new model that provides a complete statistical description of small-buffer Internet networks. First, we present novel models of the distributions of several network components, such as the line occupancies of each flow, the instantaneous arrival rates to the bottleneck queues, and the bottleneck queue sizes. Then, we combine all these models in a single fixed-point algorithm that forms the key to the global statistical small-buffer network model. In particular, given some QoS requirements, this new model can be used to precisely size small buffers in backbone router designs.

Keywords: Small-Buffer Network, Network Model, Backbone Routers.

1 Introduction

Current backbone routers typically contain extremely large buffers. These buffers take about half of their board space and a third of their power consumption [1]. They rely on massive amounts of SRAM and DRAM with fast access times, require complex scheduling algorithms to manage these SRAM and DRAM modules, and can take a significant amount of time to design [2].

These large buffer sizes typically result from a widely-followed rule of thumb, stating that router buffer sizes should be equal to the product of the typical (or worst-case) round-trip-time by the router capacity [3]. This rule of thumb is derived when considering synchronized TCP flows. For instance, given a standard linecard with 40 Gbps and a 250-ms round-trip time, the rule of thumb dictates a large linecard buffer of 10 Gb, which needs several DRAM modules and cannot be practically implemented in SRAM alone.

Recent papers in the literature suggest that this rule of thumb overprovisions buffers by several orders of magnitude [4,5,6,7,8]. In fact, as the number of TCP flows becomes extremely large, they are much less synchronized, and therefore their sum is smoother than for synchronized flows, hence incurring smaller buffer

needs. More precisely, these papers argue that the TCP flows can be modeled as independent, and therefore, by the law of large numbers, the total number of TCP packets in the network converges to a Gaussian distribution. As a consequence, in this model, also known as the *Stanford model*, it is hypothesized that the needed buffer size is smaller by a factor of about $\sqrt{n}$ than the rule of thumb for synchronized flows, where n is the number of TCP flows going through the buffer. For instance, given a million flows, the example above yields a buffer size of about 10 Mb in the Stanford model, which can be implemented in SRAM instead of DRAM. If true, such a result would obviously incur significant architectural changes in backbone routers: for the same power budget, it would be possible to pack more lines, thus increasing the router capacity. The memory architecture would be much simplified. And the input and output queues might be packed together with the switch fabric in a single chip, hence increasing its modularity as well.

Unfortunately, because it analyzes networks with large buffers, the Stanford model assumes that most of the traffic variability is in the buffers, and not on the lines. However, this assumption does not hold anymore in small-buffer networks, where the variability in the line occupancy cannot be neglected. Therefore, a new model is needed in small-buffer networks. This is the objective of this paper.

Our work is related to several studies in the literature. First, the Stanford model is developed in [4, 5, 6], but none of these papers provide any complete network models for small-buffer networks. In contrast, [7,8,9] do provide network models, but they all assume Poisson arrivals to the bottleneck queues, and we will later see that these assumptions do not match simulations in small-buffer networks. Other models also consider non-droptail queueing policies [10]. Finally, [6] shows that buffers can be made even smaller, but it assumes that TCP is altered or access lines are made slower.

In this paper, we introduce a new method that provides a complete statistical description of large Internet small-buffer networks with TCP traffic. To do so, we consider each bottleneck queue, and successively build models for the distributions of several network components around this queue. We then connect all these models together in a closed loop, and derive the final network model as a result of a fixed-point equation.

In particular, contributions of this paper include: (a) to our knowledge, the first-ever model for the traffic distribution on the links entering the bottleneck queues; (b) a new model for the instantaneous arrival rate to the bottleneck queues, including a decomposition along the input lines and a Gaussian-based model for the total rate; (c) a model for the occupancy distribution and packet loss rate of bottleneck queues that does not make assumptions on the incoming traffic load and does not assume that it is Poisson; and, most importantly, (d) a closed-loop model of small-buffer networks that enables us to determine their loss rate, as well as the distributions of the major network components. We also conclude this paper with a detailed discussion of assumptions and consequences of these results. In particular, these models can be used by router designers to

determine the necessary router buffer size given any loss probability target in any large network topology.

The rest of the work is organized as follows. Section 2 presents the notations and the closed-loop model used in this paper. Section 3 contains the models of the different network components. Then, Section 4 presents simulation results. Finally, Section 5 discusses the assumptions used and Section 6 discusses the generality of the presented results.

2 Notations and Closed-Loop Model

2.1 Notations

Our objective is to model a large network with small buffers. We will first formally reduce the problem to a simpler dumbbell topology problem, and then introduce the different notations used.

Assumption 1 (Dumbbell Topology). *The large modeled network can be decomposed into subnetworks with a single bottleneck buffer in each subnetwork, each subnetwork being modeled using a dumbbell topology around its bottleneck buffer.*

This is a classical assumption (see for instance [4, 6, 9, 11, 12]), which we further discuss in Section 5. The intuition behind it is that the behavior of TCP flows mainly derives from the congestion of the buffers on their path, and that in practice, a single buffer on their path typically causes most of the congestion.

As shown in Figure 2.2, the dumbbell topology includes n persistent TCP flows sharing the same bottleneck buffer of capacity C and buffer size B. For each flow i, we will denote the congestion window size as W_i, the access link occupancy as L_i, the access link propagation time as T_i, and the total round-trip propagation time, not including the queueing time, as RTT_i. We will also denote the bottleneck queue size as Q, and its loss rate as p.

The latencies of the forward and backward access links are assumed to follow some given positive distribution, and their capacities are assumed to be very high so that the bottleneck link is the only one experiencing congestion. Further, the TCP window sizes are assumed to be integer and have positive lower and upper bounds. The queueing policy is assumed to be drop-tail. Finally, all modeled distributions are assumed to be stationary. Note that the simulations in Section 4 relax these assumptions, e.g. by allowing for short TCP flows. Also, Section 5 further discusses the influence of these assumptions on the general results.

2.2 Closed-Loop Model

Our goal is to provide a closed-loop model for the dumbbell network. First, we will establish a general set of inter-related models. Then, by solving a fixed-point problem involving all these inter-related models, we will converge towards a final network model.
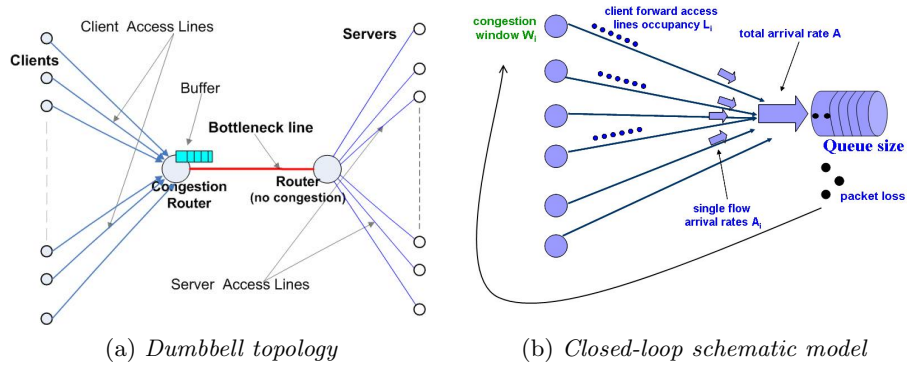
(a) *Dumbbell topology*   (b) *Closed-loop schematic model*

Fig. 1. Two topology views

As illustrated in Figure 2.2, the inter-related models will successively provide equations for the distributions of: (1) the access link occupancies $\{L_i\}$, (2) the instantaneous arrival rates $\{\Delta A_i\}$, (3) the total instantaneous arrival rate ΔA, (4) the bottleneck queue size Q, (5) the value of the loss rate p, and (6) the congestion window sizes $\{W_i\}$. The first five models are presented in this paper. The last model for the congestion window distribution can be taken from the many literature references (see for instance [13, 14]).

Once the inter-related models are obtained, we can put them together in a loop using the following schematic chain:

$$p \Rightarrow \{W_i\} \Rightarrow \{L_i\} \Rightarrow \{\Delta A_i\} \Rightarrow \Delta A \Rightarrow Q \Rightarrow p, \tag{1}$$

which can be rewritten as:

$$p = f(p). \tag{2}$$

We can then simply find p by solving this fixed-point equation, and the solution provides us as well with the distributions of all the network components mentioned above. To solve for p, it is possible to use the gradient descent algorithm for a few iterations till $|p - f(p)| < \epsilon$ for a desired ϵ.

In other words, we now have the ability to provide a model for all the main characteristics of this small-buffer network when given only the link latencies, bottleneck link capacity, and buffer size.

3   Closed Loop for the Packet Loss Rate Derivation

In this section we will successively go through the inter-related models presented above.

3.1   Model of L_i

We will now assume that we are given the distributions of the TCP congestion windows W_i, and want to provide a model for the distribution of L_i. To do so, we will first make two simplifying assumptions.

Assumption 2 (Independence). *The $\{W_i\}$ are independent and identically distributed.*

In the remainder, we will denote their common distribution function as f_W. Intuitively, this simplifying assumption relies on the fact that as the number of flows increases, their mutual synchronization decreases, and therefore the congestion windows can increasingly be modeled as independent. Note that this assumption, as well as the next one, are further discussed in Section 5.

TCP flows typically send packets and ACKs (acknowledgements) in a highly bursty manner. Further, we can use their congestion window size to approximate the size of this burst. The following assumption models this high burstiness.

Assumption 3 (Burstiness). *Flow i has a total of W_i packets (or ACKs on the reverse path), all present as a single burst on a given link.*

Note that this simplifying assumption directly contradicts the common fluid models of TCP, which assume that the window is spread out, and assumes instead that the window is concentrated at a single point. This assumption also uses the fact that we consider small-buffer networks: since the probability of having packets in the buffer is small enough, it can be neglected in this model. We can now derive the distribution of the access link occupancies L_i:

Theorem 1 (Access Link Distribution). *The number of packets on forward access link i is distributed as:*

$$Pr(L_i = k) = \begin{cases} 1 - \frac{T_i}{RTT_i} & \text{if } k = 0, \\ \frac{T_i}{RTT_i} \cdot f_W(k) & \text{otherwise.} \end{cases} \tag{3}$$

Proof. Using Assumption 3, the probability that the burst is not present on the access line is $1 - T_i/RTT_i$. The burst presence probability is independent of its size, since the propagation times are the same no matter what the burst size is. Therefore, the probability that $k > 0$ packets are present on the access line is the product of the probability that the burst size is k ($f_W(k)$) by the probability that the burst is present on the access line (T_i/RTT_i). ∎

The simulation results regarding this model are presented in Section 4.

3.2   Arrival Rates of Single Flows and Total Arrival Rate

Denote the number of packets of flow i arrived to the bottleneck queue in Δt seconds as ΔA_i. Intuitively, $\frac{\Delta A_i}{\Delta t}$ represents the (instantaneous) arrival rate on line i to the bottleneck queue. We are interested in studying the distribution of ΔA_i for some small $\Delta t < T_i$, and obtain the following model:

Theorem 2 (Flow Arrival Rate Distribution). *The number of packets ΔA_i of flow i arrived during time Δt is distributed as:*

$$Pr(\Delta A_i = k) = \begin{cases} 1 - \frac{\Delta t}{T_i} + Pr(L_i = 0) \cdot \frac{\Delta t}{T_i} & \text{if } k = 0, \\ Pr(L_i = k) \cdot \frac{\Delta t}{T_i} & \text{otherwise.} \end{cases} \tag{4}$$

Proof. By Assumption 3, packets of flow i move on each line in a single burst of size W_i and at a constant speed. Therefore, the probability that on some link i, $k > 0$ packets arrive within Δt, is $Pr(L_i = k) \cdot \frac{\Delta t}{T_i}$. This gives us the probability for the packet arrival of any size larger then zero. The complementary probability, therefore, stands for the no-arrival event.

Incidentally, note that Equation (4) can be rewritten as in Equation (3):

$$Pr(\Delta A_i = k) = \begin{cases} 1 - \frac{\Delta t}{RTT_i} & \text{if } k = 0, \\ \frac{\Delta t}{RTT_i} \cdot f_W(k) & \text{otherwise.} \end{cases} \tag{5}$$

This is Equation (3), replacing the access link propagation time T_i with the propagation time Δt. ∎

We now want to find the total instantaneous arrival rate $\Delta A = \sum_i (\Delta A_i)$ of all flows. Since the $\{W_i\}$ (and the $\{L_i\}$) are statistically independent by Assumption 3, the $\{\Delta A_i\}$ are statistically independent as well. Thus, we can use Lindeberg's Central Limit Theorem [15] to show that the total arrival rate ΔA will be Gaussian for a large number of flows. We refer interested readers to [14] for the detailed proof.

Theorem 3 (Total Arrival Rate Distribution). *When the number of flows $n \to \infty$, the normalized total arrival rate $\frac{\Delta A - \sum_i E(\Delta A_i)}{\sqrt{\sum_i Var(\Delta A_i)}}$ converges in distribution to the normalized Gaussian distribution $\mathcal{N}(0, 1)$.*

The simulation results comparing this Gaussian model with a typical Poisson model are presented in Section 4.

3.3   Queue Size Distribution and Packet Loss Rate

Our next objective in the fixed-point model is to find the distribution of the queue size Q and the packet loss rate p given the above model for ΔA. We will decompose time into frames of size Δt, and assume that packets arrive as bursts of size ΔA every Δt seconds. Thus, we clearly get the following queueing model:

Theorem 4 (Queue Size and Loss Rate). *The queue size distribution and the packet loss rate are obtained by using a $G_{\Delta t}^{[\Delta A]}/D/1/B$ queueing model, in which every Δt seconds, packets arrive in batches of size ΔA and immediately obtain service for up to $C\Delta t$ packets.*

We implemented this queueing model using both the algorithm developed in [16] and a simple Markov chain, and both yielded similar results.

We are now done with our set of network models, and are ready to analyze their correctness using simulation results.
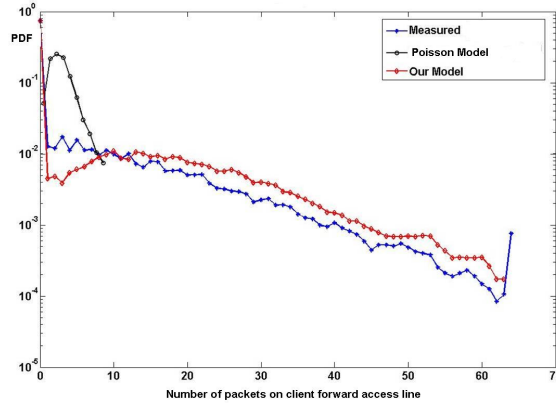
Fig. 2. Distribution of the access link occupancy L_i (logarithmic scale)

4   Simulation Results

We will now present the simulation results for the different parts of the closed-loop model. The simulations were done in ns2. Specific settings are detailed below for their respective simulations. In all the simulations below, we ran 2000 simultaneous persistent TCP NewReno flows, unless noted otherwise. The packet size was set to 1000 bytes. The maximum allowed window size was set to $W_{max} = 64$ packets. The propagation time of the bottleneck link was fixed to 20 milliseconds, and the propagation times for all the other links were chosen randomly according to a uniform distribution.

4.1   Access Link Occupancy Model vs. Fluid Model

We want to compare our bursty model for $\{L_i\}$ against a fluid model, in which packets are distributed uniformly on all the links (the queueing time is neglected). According to this fluid model, the number of packets present on access link i at time t is thus equal to $L_i(t) = \frac{T_i}{RTT_i} \cdot W_i(t)$. The maximum number of packets on the access link, therefore, is bounded by $\frac{T_i}{RTT_i} \cdot W_{max}$.

Figure 2 represents the probability distribution function (PDF) of the access link occupancy L_i of some random flow i using a logarithmic scale. It was obtained using a simulation involving 500 simultaneous TCP flows. It can be seen that our bursty model is fairly close to the measured results throughout the whole scale. It behaves especially better than the fluid model, for which the pdf is equal to 0 for any occupancy above $\frac{T_i}{RTT_i} \cdot W_{max} \approx 10$, and thus cannot even be represented on the logarithmic scale. This observation strengthens the intuition that Assumption 3 was reliable. Note that we obtained similar results on many simulations with different parameters.
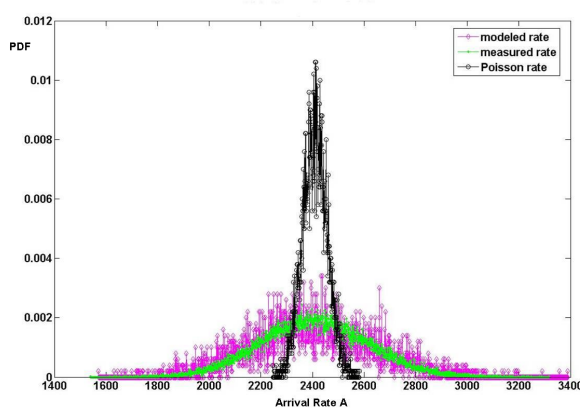
Fig. 3. Distribution of the total arrival rate ΔA in time Δt

4.2   Arrival Rate Model vs. Poisson Model

We want to compare our Gaussian-based model for the distribution of the total instantaneous arrival rate ΔA with a typical Poisson model [7, 8, 9]. Figure 3 plots the pdf of ΔA, using $\Delta t = 10$ ms, and compares it with simulated data obtained artificially using the two models. Our Gaussian-based model relies on the expected value and variance computed above. On the contrary, for the Poisson model, we force the expected value to equal the measured value. Of course, over this amount of time, the Poisson model yields an approximately Gaussian distribution as well. As shown in the figure, our model approximately yields the correct variance and is close to the simulated results, while the Poisson model yields too small a variance. This can be explained by the burstiness properties of the TCP flows.

4.3   Queue Size Distribution Results and Comparisons

Figure 4 compares the measured queue size distribution with our model. We used a buffer size of 580 packets. Our Markov-chain-based queue model was obtained after the convergence of the entire closed-loop model, and therefore uses our modeled arrivals as well.

It can be seen that our queue model is fairly close on most of the simulated range of Q, but cannot exactly reproduce the smooth continuous behavior of the queue at the edges ($Q = 0$ and $Q = B$) because of its discrete bursty nature. In fact, in this example, the true loss rate was 0.55%, while our model gives 0.76%.

4.4   Fixed-Point Solution

In the simulations, the fixed-point solution of our network model was found using the gradient descent algorithm, in typically less than 50 iterations. The exact number of needed iterations depends of course on the desired precision
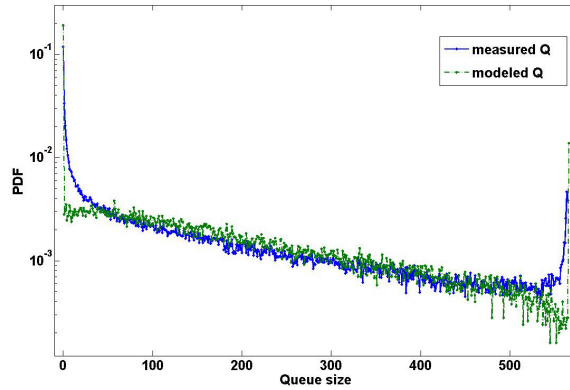
Fig. 4. Queue size distribution

and on the network parameters. Table 1 illustrates the average flow throughput, the average queueing delay (measured and modeled) and the average loss rate (measured and modeled) using the simulation results in 4 settings with quite different parameters:

- *Case 1:* 500 long-provisioned TCP flows with RTT_i distributed between 80 and 440 msec, $B = 232$ packets and $C = 232.5$ Mbps.
- *Case 2:* 500 long-provisioned TCP flows with RTT_i distributed between 80 and 440 msec, $B = 450$ packets and $C = 116.25$ Mbps.
- *Case 3:* 750 long-provisioned TCP flows and a constant number of 25 short TCP flows, with RTT_i distributed between 70 and 2040 msec, $B = 244$ packets and $C = 69.75$ Mbps. (The short flows were omitted in the model.)
- *Case 4:* 1500 long-provisioned TCP flows and a constant number of 5 short TCP flows, with RTT_i distributed between 220 and 240 msec, $B = 828$ packets and $C = 209.25$ Mbps. (The short flows were omitted in the model.)

In all these cases, the modeled queueing delay and packet loss rate were close to, but slightly above, the measured results. Incidentally, using the same simulations without short flows, we also verified that the influence of the short flows on the simulation results was negligible. Note also that all these cases use small buffers, with the last case following the Stanford model, and the three first cases having even smaller buffers.

5   Discussion of Assumptions

Let's now discuss the correctness and generality of the assumptions.

Dumbbell Topology — We assumed in Assumption 1 that any large network can be subdivided into dumbbell topologies. This assumption relies on the observation that in the Internet, few flows practically have more than one bottleneck, and that flows having more than one bottleneck actually mainly depend on the most congested one [4]. Thus, the assumption of a single point of congestion

Table 1. Measured versus modeled results

Case	Average flow throughput	Queueing delay		Loss rate p	
		Measured	Modeled	Measured	Modeled
Case 1	52 pkts/sec	1.79 msec	1.9 msec	0.79 %	0.90 %
Case 2	29 pkts/sec	18.27 msec	19.87 msec	1.81 %	2.10 %
Case 3	11 pkts/sec	7.84 msec	8.54 msec	1.30 %	1.30 %
Case 4	18 pkts/sec	22.16 msec	26.15 msec	3.30 %	3.50 %

seems realistic enough. However, we also assumed that the congestion only affects packets, not ACKs. This assumption might be too restrictive – even though we found that our Gaussian-based models still held in various simulations using reverse-path ACK congestion.

Statistical Independence of $\{W_i\}$ — It is obviously not correct that the congestion windows are completely independent, since they interact through the shared bottleneck queue. However, in order to study how far from reality the independence assumption is, we checked how correlated the congestion windows were over time. We obtained the following correlation matrix for the congestion windows of five arbitrary flows, using 70,000 consecutive time samples in a simulation with 500 persistent TCP flows.

$$C = \begin{pmatrix} 1 & 0.066 & 0.14 & 0.058 & -0.025 \\ 0.066 & 1 & 0.054 & 0.0005 & -0.081 \\ 0.14 & 0.054 & 1 & 0.063 & 0.051 \\ 0.058 & 0.0005 & 0.063 & 1 & 0.003 \\ -0.025 & -0.081 & 0.051 & 0.003 & 1 \end{pmatrix} \tag{6}$$

It can be seen that the correlation coefficients between different flows were indeed quite low and far from the maximum absolute value of 1. Of course, while independent random variables have zero correlation, the reverse is not necessarily true. Nevertheless, this low correlation would tend to indicate that the flows are indeed desynchronized, and therefore that the simplifying assumption of independence is not too far from reality. In fact, as the number of flows increases, we also found that a heuristic measure of the independence of their window sizes was decreasing (we took two arbitrary flows and used the symmetric form of the Kullback-Leibler distance between the joint distribution of their window sizes and the product of their respective distributions). For instance, it was 20.03 for 10 flows, 3.37 for 50 flows, and 2.65 for 200 flows.

Identical Distribution of $\{W_i\}$ — The distributions of the congestion window sizes mainly depend on the loss rate p in the shared bottleneck buffer [13,14]. In simulations, the loss rate was indeed found to be equal for all flows when there is no strong synchronization.

Burstiness — In Section 4, the comparison of the bursty model with the fluid model already strengthened the bursty assumption. More generally, this assumption needs to be used with care in networks without enough space on the links for a packet burst (extremely small link latencies or link capacities). In other cases, while not exactly reflecting reality, this assumption seems close enough [4].

RTT Distribution and Queuing Delay — We assumed that the queueing delay can be neglected in front of the link propagation times. In fact, in the Stanford model, the worst-case queueing delay is $\frac{B}{C} = \frac{RTT}{\sqrt{n}}$, where RTT is the average round-trip propagation time. Consequently, the assumption seems reasonable when n is large, as long as there is no flow with a round-trip propagation time significantly small in front of RTT.

Number of Packets — We assumed that the number of packets in the network is close to the total window size, as reflected by the definition of the window size. This assumption is especially justified when the loss rate is reasonably small, as seen in our simulations and in the literature [4].

6   Generality of Results

It is obviously impossible to consider every possible topology and every possible set of flows. We made hundreds of simulations for this paper, and still feel that there is much to research. Nevertheless, we can already discuss the scope of the results and their sensitivity to various topology parameters.

Buffer Size — Our paper is about models that are valid in networks with small buffers. In a network with large buffers, the flows might become synchronized, and our independence assumptions and the ensuing models might not be applicable. Likewise, we would not be able to neglect the queueing delay in our models.

Propagation Times — We chose the link propagation times using a uniform distribution with different parameters so as to reflect the diversity of real-life Internet flows. In simulations, our models were fairly insensitive to these propagation times, as long as there were many flows and there were no flows with near-zero round-trip-times.

Protocols — The closed-loop model fits a network with long-provisioned TCP flows. The cases above illustrate how a small portion of short TCP flows had no major influence on the results – in fact, the short flows contribute to the global desynchronization and make the independence-based models even closer to simulated results! Likewise, we believe that a small portion of UDP traffic would not have any influence on the results because of its open-loop nature.

Number of Flows — In simulations, we found that the desynchronization was already practically correct for several hundreds flows, as previously stated in [4]. We thus believe that with the hundreds of thousands of flows present in a congested backbone router, the desynchronization will be even more correct.

7   Conclusion

In this paper, we provided a complete statistical model for a large network with small buffers. We started with a model for the traffic on a single access line. Then, we modeled the arrival rates to the bottleneck queues. Later, we found

a model for the queue size distribution and the loss rate. Finally, using these inter-related models, we showed how to solve a single fixed-point equation to obtain the full network statistical model.

A router designer might directly use this network model for buffer sizing. Indeed, the designer might consider a set of possible benchmark parameters and model the behavior of the resulting network. Then, given target QoS requirements such as the required maximum packet loss rate or maximum expected packet delay, the router designer will be able to design the buffer size that satisfies these constraints.

References

1. McKeown, N.: Sizing router buffers. EE384Y Course Talk, Stanford (2006)
2. Shrimali, G., Keslassy, I., McKeown, N.: Designing packet buffers with statistical guarantees. In: IEEE Hot Interconnects XII, Stanford, CA (2004)
3. Villamizar, C., Song, C.: High performance tcp in ansnet. ACM Computer Communications Review (1994)
4. Appenzeller, G., Keslassy, I., McKeown, N.: Sizing router buffers. In: ACM SIGCOMM, Portland, OR (2004)
5. Appenzeller, G., McKeown, N., Sommers, J., Barford, P.: Recent results on sizing router buffers. In: Network Systems Design Conference, San Jose, CA (2004)
6. Enachescu, M., Ganjali, Y., Goel, A., McKeown, N., Roughgarden, T.: Routers with very small buffers. In: IEEE Infocom, Barcelona, Spain (2006)
7. Raina, G., Towsley, D., Wischik, D.: Part II: control theory for buffer sizing. ACM Computer Communications Review (2005)
8. Raina, G., Wischik, D.: Buffer sizes for large multiplexers: TCP queueing theory and instability analysis. In: EuroNGI (2005)
9. Avrachenkov, K.E., Ayesta, U., Altman, E., Nain, P., Barakat, C.: The effect of router buffer size on the TCP performance. In: LONIIS Workshop, St. Petersburg, Russia, pp. 116–121 (2002)
10. Bu, T., Towsley, D.F.: A fixed point approximation of TCP behavior in a network. In: ACM SIGMETRICS (2001)
11. Hespanha, J.P., Bohacek, S., Obraczka, K., Lee, J.: Hybrid modeling of TCP congestion control. In: Di Benedetto, M.D., Sangiovanni-Vincentelli, A.L. (eds.) HSCC 2001. LNCS, vol. 2034. Springer, Heidelberg (2001)
12. Shorten, R., Wirth, F., Leith, D.: "A positive systems model of TCP-like congestion control: Asympototic results," Tech. Rep. 2004-1, Hamilton Institute (2004)
13. Altman, E., Avrachenkov, K.E., Kherani, A.A., Prabhu, B.J.: Performance analysis and stochastic stability of congestion control protocols. In: IEEE Infocom, Miami, FL (2005)
14. Shifrin, M.: "The Gaussian nature of TCP in large networks," M.Sc. Research Thesis, Technion, Israel (August 2007),
   http://www.graduate.technion.ac.il/Theses/Abstracts.asp?Id=23526
15. Zabell, S.L.: Alan Turing and the central limit theorem. Amer. Math. Monthly 102, 483–494 (1995)
16. Tran-Gia, P., Ahmadi, H.: Analysis of a discrete-time $G^{[X]}/D/1 - S$ queueing system with applications in packet-switching systems. In: IEEE Infocom, New Orleans, LA (1988)