Modeling, Simulation and Measurements of Queuing Delay under Long-tail Internet Traffic

Michele Garetto
Dipartimento di Elettronica
Politecnico di Torino
Torino, Italy 10129
garetto@polito.it

Don Towsley
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
towsley@cs.umass.edu

ABSTRACT

In this paper we describe an analytical approach for estimating the queuing delay distribution on an Internet link carrying realistic TCP traffic, such as that produced by a large number of finite-size connections transferring files whose sizes are taken from a long-tail distribution. The analytical predictions are validated against detailed simulation experiments and real network measurements. Despite its simplicity, our model proves to be accurate and robust under a variety of operating conditions, and offers novel insights into the impact on the network of long-tail flow length distributions. Our contribution is a performance evaluation methodology that could be usefully employed in network dimensioning and engineering.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Packet-switching networks*; I.6.5 [**Simulation and Modeling**]: Model Development

General Terms

Performance, Measurement, Theory

Keywords

Queueing Analysis, Markovian Models, TCP

1. INTRODUCTION

Modeling the behavior of a queue loaded by Internet traffic is a fundamental problem of network performance evaluation for which satisfactory solutions are not yet available. Several Internet research topics relate directly or indirectly to this problem: network design, capacity planning, Quality of Service guarantees, end-to-end measurements, congestion control, Active Queue Management.

SIGMETRICS'03, June 10–14, 2003, San Diego, California, USA.

In this paper we present a simple analytical technique to study the case of an uncongested link. This is the normal operating condition for backbone links, that are generally over-provisioned, so that utilization is below 50% even at peak hours. However, this is not a limitation of our approach, just a simpler case that allows us to assume an infinite buffer size. Our problem is that of predicting the behavior of a FIFO queue fed by the traffic produced by a large number of finite TCP flows, where the objects being transferred are characterized by a long-tail length distribution. We assume that the following parameters are known: (i) the traffic intensity; (ii) the flow size distribution; (iii) the packet loss probability suffered by the flows on their end-to-end path (it might be equal to zero). The solution presented in this paper is open-loop, in the sense that we require an estimate of the packet loss probability a priori. Again, this is not a limitation of our approach: our model can be incorporated into a closed-loop algorithm, such as a Fixed Point Approximation (see for example [18]), requiring no additional inputs other than physical system parameters. In fact we have successfully extended our model to consider the case of a congested link using the FPA technique, but we do not present this extension here due to lack of space (the interested reader is referred to [1]).

The main contributions of this paper can be summarized as follows: (i) we identify the transport protocol (TCP) as primarily responsible for the queue behavior; (ii) we describe in detail a simple stochastic model of TCP that can quantitatively characterize the burstiness produced by the traffic sources; (iii) we introduce a novel queuing system to obtain the queue length distribution; (iv) we offer original insights into the effect of long-tail file size distributions on the network; (v) we show how our approach can be successfully applied to study more general network scenarios than those considered in this paper, proving that it could be an attractive solution for network engineering.

Our model obtains surprisingly accurate results in terms of queue length distribution (or, equivalently, queueing delay distribution) according to *ns-2*simulations in a controlled network environment. We have also validated analytical predictions against actual delay measurements taken from a real network scenario, although only qualitatively. This achievement is very important, since it allows one not only to compute the mean and variance of the queuing delay suffered by the flows traversing the link (which is part of their round trip times), but also to answer common questions that arise in network design and operation, such as (i) what is

the probability that a packet experiences a queuing delay higher than a certain value (think of Service Level Agreements) ? (ii) what is the probability of buffer overflow given the maximum buffer size and the link utilization (think of buffer dimensioning) ? Many network operators and Internet Service Providers still dimension their networks through trial and error, or in a completely ad-hoc manner leading to considerably inefficient resource usage. The availability of simple and robust analytical models is thus essential.

The literature related to our work is quite vast, so that it is impossible to provide here a comprehensive overview of previous contributions. Since the failure of Poisson modeling [4] was pointed out [4], many efforts have been devoted to study characteristics of Internet traffic such as long range dependence, self-similarity, and multi-fractal scaling [6–8]. The complexity of the packet arrival process has appeared to be intractable using traditional approaches [3], and it has even been suggested that an entirely new theory is necessary to develop traffic engineering tools [5]. To our knowledge, no study has yet attempted to predict the entire queue length distribution on an Internet link carrying realistic traffic such as that produced by finite TCP flows transferring an amount of data characterized by a long-tail length distribution. The behavior of finite-size TCP connections has been studied in [14–16]. The impact of packet-level characteristics of Internet traffic on queue behavior has been considered in [10–13]. Multifractal behavior as a manifestation of the burstiness induced by TCP congestion control has been suggested in [9]. Measurements of one-hop delay on an operational backbone network have been reported in [21].

The paper is organized as follows: in Section 2 we clarify the problem that we face by means of a numerical example. We describe our model in detail in section 3. In Section 4 we report on an extensive study with *ns-2*intended to assess the robustness of the model under different parameter settings, focusing on the impact of long-tail flow length distributions and discussing the applicability of our model to general network topologies. In Section 5 we provide an example of how our modeling technique was successfully employed to study a real network scenario. Finally, we conclude in Section 6.

2. OVERVIEW

We start with a simple example that illustrates much of the problem complexity. Consider the topology depicted in Figure 1. This example will be revisited several times in the rest of the paper. Its physical parameters have been chosen to represent the real network scenario considered in Section 5, which is the access link of a campus network.

The bottleneck link capacity is $C_L = 28$ Mbps, with a propagation delay of 0.5 ms. There are $N_2 = N_1 = 100$ peripheral links at both sides, of capacity $C_1 = C_2 = C_L$. The propagation delay at the transmitters' side is uniformly distributed between $D_{min} = 10$ ms and $D_{max} = 50$ ms. The propagation delay at the receivers' side is uniformly distributed between 0.4 ms and 0.6 ms.

We assume that new connections open according to a time invariant Poisson process, so that the average link utilization is constant over time. This is not true over a long time scale in a real network (think of typical daily variations) but can be considered acceptable over small time scales (say for a few minutes). The connection establishment dynamics is as follows: when a new connection opens, it chooses randomly a transmitter node and a receiver node. A connection consists

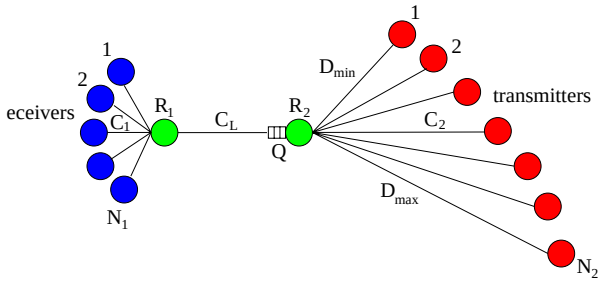

Figure 1: Simple topology for an access link

of a single data transfer whose size S, expressed in a number of full-size packets, is taken from a geometric distribution with an average $\bar{S} = 20$. We ignore the three-way-handshake and connection termination. The packet size P is assumed constant, equal to 8000 bits. The packet transmission time over the link is $\tau = P/C_L$. The maximum window size is 32 packets and the TCP version is New-Reno, implemented as in the *ns-2*simulator.

We are interested in the behavior of queue Q of router R_2. We obtained from simulation the distribution of the total number of packets in the queue (both queued packets and the packet being transmitted) in order to compare the results from what can be obtained from standard queuing theory. For now we consider an infinite capacity drop-tail queue, so that no packets are lost traversing the access link.

For this particular example, we artificially introduced iid losses on each of the N_2 links with probability $p = 0.01$. The arrival rate of new connections is denoted by Λ. We define as *link utilization* ρ the quantity:

$$\rho = \frac{\Lambda \bar{S} P}{C_L} \tag{1}$$

Because of the way losses are introduced in the network in our simulation experiments (this will be explained in Section 3-A.1) we are able to say that ρ is the *actual* utilization of the link, equal to what is referred to in queueing theory as *utilization factor* or *traffic intensity*. This comes from the property that, in our simulations, each packet traverses the link exactly once. In the case $\rho = 0.8$ the results of this experiment are shown in Figure 2. We compare the simulation results with 4 analytical curves: an $M/M/1$ queue, an $M/D/1$ queue, a $G/M/1$ queue and an $M^{[X]}/M/1$ queue, where arrivals occur in batches whose sizes are taken from a distribution computed by our model. The $M/M/1$ model is indeed the simplest, and yields the well known result for the probability of having i packets in the queue $\pi_i = (1-\rho)\rho^i$. It fails at predicting the actual queue length distribution because it assumes that the packet arrival process is Poisson, whereas this is not the case in the Internet [4].

If one wants to account for the fact that constant-size packets result in deterministic service times, an $M/D/1$ model should be used, but the distribution of packets in the queue would be even further from the simulation. One now may think that because the inter-arrival time between two consecutive packets is not exponential, a more general $G/M/1$ model should be adopted. Figure 2 includes the queue length distribution predicted by the $G/M/1$ model, where the inter-arrival time distribution is obtained directly from the simulation, observing the arrival process of packets

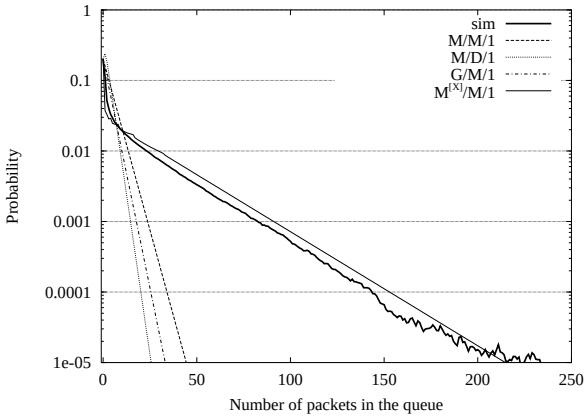

Figure 2: Queue length distributions obtained from simulation and different analytical models when $\rho = 0.8$

at the queue. After we solved the $G/M/1$ model numerically, we obtained a curve worse than that of the $M/M/1$ queue !

The errors that we encounter are due to the assumption that inter-arrival times are independent. TCP introduces strong correlations that have a deep effect on the the queue length. This is a well known characteristic of TCP traffic, generally referred to as its "burstiness". In order to capture quantitatively this important correlation, we resorted to a $M^{[X]}/M/1$, a queue with batch arrivals with batch size X. We discovered that approximating X by the number of packets sent by TCP sources every round trip time provides very accurate results. In particular, the geometric decay of the buffer occupancy is surprisingly good, as can be seen in Figure 2. The model also provides a conservative prediction, which is a desirable property for its possible applications. We will show in Section 3-B that it is possible to further refine the analysis resulting in a queue length distribution almost indistinguishable from that obtained from simulation. In Section 4 we will observe that our technique indeed captures most (unfortunately not all) of the traffic correlations under a variety of parameter settings, and we will explain when and why it instead fails at providing an accurate prediction.

3. MODELING

Our modeling approach consists of two components: the first one is a stochastic model of TCP that, given a quite generic flow length distribution, derives the distribution of the number of packets transmitted every RTT. This will be described in section 3-A, and it is one of the contributions of this paper; the second one is an analytical model of the queue capable of predicting all of the results of interest (average values as well as distributions). This will be described in section 3-B and is another important contribution of our work.

3-A TCP Model

In this section we provide a detailed description of our stochastic model of TCP including a discussion of the assumptions that we made. We will first consider the case in which the amount of data transferred by a TCP connection is a geometrically distributed random variable with mean

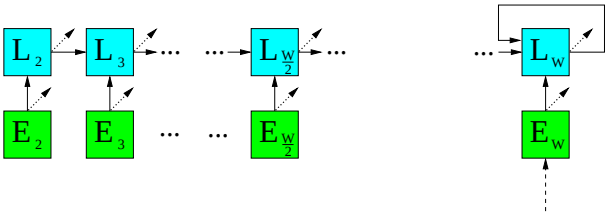

Figure 3: Stochastic Finite State Machine of TCP

$1/q$. In Section 3-A.3 we will describe how this model can be extended to account for a generic long-tail distribution of the flow length. We exploit the memoryless property of the geometric distribution to build a simple, yet accurate model of TCP dedicated only to the purpose of deriving the batch size distribution of the packets flowing through the network. Thus we neglect many details of TCP that have no impact on this distribution. The model describing the TCP dynamics of relevance to us is shown in Figure 3, and is pretty simple. Basically, it is a finite state model where each state corresponds to a state of the TCP protocol, connected with probabilistic transitions that depend on the packet loss probability. We call this object a *Stochastic Finite State Machine* (SFSM). Readers familiar with queuing networks will find it simpler to think of it as an open network of $\cdot/G/\infty$ queues where customers in a given queue stand for TCP connections in a given protocol state. The solution of the queueing network is straightforward, as it requires solving the flow balance equations just once.

In Figure 3 we have only shown transitions corresponding to the successful delivery of all of the packets sent in each state. Many other transitions corresponding to loss events are not shown because doing so would complicate the diagram. The entry point to this state machine is state E_W, as indicated by the dashed arrow, that represents a source that starts data transmission. The exit point can be any state in the diagram, as indicated by the oblique dotted arrows, representing sources that have exhausted the amount of data to send.

We assume the reader is familiar with the basic congestion control algorithms of TCP. States labelled L_i stand for the congestion avoidance phase with window size i (Linear growth), $i = 2, \cdots, W$. Here W is the maximum window size. States labelled E_i represent the slow-start phase with threshold i (Exponential growth), $i \in \{2, \cdots, \lfloor W/2 \rfloor, W\}$. The index range for i can be explained as follows: when a connection starts transmitting, we assume the threshold is set to the maximum window size. After the first loss event, the threshold cannot exceed the value $\lfloor W/2 \rfloor$ for the remainder of the connection lifetime. The total number of states is thus roughly $\frac{3}{2}W$.

Since we are not interested in *when* a source transmits its packets, but only *how*, we do not mind how long a connection remains in each state, and so we can disregard timeouts and the timeout back-off mechanism. The purpose of this model is not to derive performance figures of TCP (such as throughput or completion time) but to identify the number of packets sent every RTT, in order to capture the burstiness of TCP traffic and analyze its impact on the network.

In order to completely specify the model, we need to describe the underlying loss model (3-A.1), the transition probabilities and the batch size computation (3-A.2).

3-A.1 Loss Model

The loss model is an important component of any stochastic model of TCP. For our purposes, we have chosen a quite general model, that allows us to understand the impact of different degrees of loss correlation.

First, we introduce the notion of *loss event*. A *loss event* is a sequence of packets that starts with the first packet lost in a window (this packet is included in the sequence) and ends when the same packet is retransmitted by the source (this retransmission is not part of the sequence). We assume that loss correlation, if any, involves only the packets sent between these transmissions of the same packet. A packet not included in a loss event can generate a new loss event with a probability p that we call *loss event probability*. Note that the retransmission of a packet can generate a new loss event. We distinguish the *loss event probability* from the generic *packet loss probability*, indicated as $\bar{p}$, which is defined as the average loss probability of all transmitted packets [1].

A *loss event* can be regarded as the "congestion signal" sent by the network to the source, leading to a single window reduction [2]. We rely on an assumption commonly used in the literature, that the arrival of congestion signals at the source is well described by a Poisson stream, so that we can use a Bernoulli scheme for the first packet lost in a window. This assumption is verified on simulation (see Section 4), and has also been validated by real measurements in [19].

The number of packets lost within a loss event depends on the degree of loss correlation. We have chosen to model only two extreme cases: the case in which only the first packet of a loss event is lost (let P_s denote the probability of such an event), and the case where the complete sequence of packets corresponding to a loss event is lost (let P_w denote the probability of such an event, with the constraint that $P_s + P_w = 1$). The first case typically allows the source to fast retransmit the single lost packet and continue to send new segments at half of the window size (provided that three dup acks have been received and the retransmission has been successful). The second case results in a timeout and the sizing of the window to one. Probabilities P_s and P_w can be used to model the behavior of different TCP versions as well as to account for the effect of particular loss correlations. It is also possible to make these probabilities depend on the window size or on the state of the protocol, if desired, without increasing the complexity of the solution of the SFSM. We will observe in Section 4 that P_s and P_w do not play an important role in the burstiness of the aggregate traffic produced by TCP, so that we believe our approach is acceptable.

We introduced this loss model directly into *ns-2*, in order to relieve ourselves from discrepancies in the results due to different reactions to losses between analysis and simulations. After validating the model using this artificial loss process we ran experiments with real drop-tail losses, and observed that our our approach is suitable to analyzing a realistic loss process as well. The case of drop-tail losses will be considered in Section 4. Finally, we remark that our loss model eliminates unnecessary retransmissions from the sources[3] and, since losses occur only on the transmitters' side of the access link, we are guaranteed that each packet traverses the link just once, as anticipated in Section 2.

3-A.2 Transition probabilities and batch size computation

In this Section we describe the transition probabilities necessary to solve the SFSM shown in Figure 3, for the case in which the number of packets in the transfer is geometrically distributed. At the same time we explain how to compute the size of the groups of packets sent in each state. To simplify the exposition, we restrict ourselves to the groups of packets arriving at the queue of interest. In general it is possible to describe both the groups of packets flowing *before* the point in the network in which losses occur, and the groups of packets flowing *after* that point. Here we compute the size of the groups of packets arriving at the queue *after* traversing the lossy links (see Figure 1).

The parameters necessary to specify the model are:

- q, the parameter of the geometric distribution (the average size is $1/q$).
- p, the *loss event probability*.
- P_s, the probability of a single loss per *loss event*, or its complement P_w, as explained in Section 3-A.1.
- W, the maximum window size expressed in number of full-size segments.

By "solving the model" we mean finding the arrival rate to each state, given the arrival rate of new connections at the entry point, which is simply $\lambda_{E_W} = \Lambda$, where Λ is the arrival rate of new connections. This reduces to solving the system of linear equations defined by the matrix of transition probabilities $P(S_i, S_j)$ from state S_i to state S_j. The complexity of the numerical solution is thus very low. Once we have obtained the arrival rates to the states, we can compute the arrival rate of batches at the queue, using the probabilities that batches of any given size are generated in each state.

The specification of the transition probabilities in form of equations requires an excessively complex notation. Here we describe the model in an informal way to make it more understandable to the reader[4].

To simplify the notation, we define $s = 1 - p$ (where s stands for "successful"), the probability $M_q(i) = (1-q)^i$ that more than i packets remain to send, the probability $R_q(i) = q(1-q)^{i-1}$ that exactly i packets remain to send. We assume that TCP receivers generate acks for every packet, i.e. the delayed acknowledgement option is not used (a similar model can be built if this option is used).

3-A.2.1 Congestion avoidance without loss.

When the window size is i during congestion avoidance (state L_i), we assume that i packets are sent before the window grows to size $i+1$. If no losses occur, and more than i packets remain to be sent, a connection transits to state L_{i+1} and generates a batch of size i. This occurs with probability $s^i M_q(i)$. As a special case, the window stops

[1] The stronger and longer the correlation between successive losses in a sequence of packets, the smaller is p with respect to $\bar{p}$

[2] This is true for NewReno and SACK, but not for the Reno version of TCP

[3] It can be easily shown that if a single packet or all of the packets within a window are lost, a source does not send again a packet already delivered to the destination

[4] A piece of code implementing the TCP model described in this Section is available at [2]

growing when it reaches the maximum window size W; hence we add a self loop to the state L_W (Fig. 3). A flow can also exhaust the amount of data to send. For each $j\,, 1 \leq j \leq i$, a connection leaves the SFSM sending a batch of size j, with probability $s^j R_q(j)$.

3-A.2.2 Congestion avoidance with loss.

Suppose that a loss event occurs starting with the j^{th} packet sent in state $L_i\,, (1 \leq j \leq i)$. The window, before shrinking, grows to size $n_i = \min(i+1, W)$, and when the loss is detected the threshold is updated to the value $t_i = \max(2, \lfloor n_i/2 \rfloor)$. The maximum number of packets that can be sent before the window is reduced is $d_{i,j} = j - 1 + n_i$.

In case an entire window is dropped, a connection moves to state E_{t_i}, while a batch of size $j-1$ arrives at the queue. This occurs with probability $p\, s^{j-1}\, P_w\, M_q(j-1)$.

In the case of a single loss, a connection remains within the $SFSM$ only if more than $d_{i,j}$ packets remain to be sent while entering state L_i, with probability $p\, s^{j-1}\, P_s\, M_q(d_{i,j})$. To determine the next state, we first consider the case that not enough duplicated acks are received to trigger the fast retransmit mechanism. This happens when $n_i < 4$, leading to state E_2. If fast retransmit is triggered ($n_i \geq 4$), we must distinguish the case in which the retransmission is successful (with probability s), resulting in a transition to state L_{t_i}, from when it fails (with probability p), resulting in a transition to E_{t_i}. We still need to specify the groups of packets sent in the case of a single loss. If $m \leq i$ packets remain to be sent, a batch of size $m-1$ arrives at the queue and the connection leaves the SFSM. This occurs with probability $p\, s^{j-1}\, P_s\, R_q(m)$, where j is restricted to the range $(1 \leq j \leq m)$. If m packets remain to be sent, $i < m \leq d_{i,j}$, we must split the packets into two batches of size $i-1$ and $m-i$, because they are sent spaced by a RTT, with probability $p\, s^{j-1}\, P_s\, R_q(m)$. If more than $d_{i,j}$ packets are remaining, two batches of size $i-1$ and $d_{i,j}-i$ arrive at the queue, with probability $p\, s^{j-1}\, P_s\, M_q(d_{i,j})$. Finally, we account for the retransmission of the single lost packet as a batch of size 1, with probabilities $p\, s^{j-1}\, P_s\, M_q(j-1)\,, (1 \leq j \leq i)$.

3-A.2.3 Slow start without loss.

Now we consider the states E_i representing slow-start. We remind that here i represents the threshold. The maximum number of packets sent in state E_i while the window grows from one to i is given by $e_i = 2^{\lfloor \log_2 i \rfloor + 1} - 1$. If no losses occur, and more than e_i packets remain to be sent, a connection transits to state L_i shifting to congestion avoidance, with probability $s^e M_q(e_i)$, generating a *sequence* of size e_i. A *sequence* is the set of all packets sent during a slow start phase. A simple algorithm, that we omit here, is used to convert a *sequence* into the groups of packets sent every RTT, according to the geometric progression 1, 2, 4, 8, $\cdots$. For example, a sequence of size 20 must be split into the groups $1, 2, 4, 8, 5$. In the next paragraph we will specify only the *sequences* of packets, implying that they must be split into groups in order to obtain the batches arriving at the queue.

3-A.2.4 Slow start with loss.

Suppose that a loss event occurs starting with the j^{th} packet sent in state $E_i\,, (1 \leq j \leq e_i)$. Before shrinking, the window grows to size $n_i = \min(i, j)$, and when the loss is detected the threshold is updated to the value $t_i = \max(2, \lfloor n_i/2 \rfloor)$. The maximum number of packets that can be sent before the window is reduced is $d_{i,j} = j - 1 + n_i$. Using the same notation, the transition probabilities are exactly the same as those identified for states L_i. The number of packets sent in states E_i is also computed in the same way as described for states L_i. In the case that an entire window is dropped, a *sequence* of size $j-1$ arrives at the queue. In the case of a single loss, if m packets remain to be sent, $j < m \leq d_{i,j}$, a *sequence* of size $m-1$ is generated; if more than $d_{i,j}$ packets are remaining, a *sequence* of size $d_{i,j}-1$ is generated. Finally, we account for the retransmission of the single lost packet as a batch of size 1.

3-A.3 Extension to generic long-tail flow length distributions

Our model can be easily extended to the case in which the flow length distribution is not geometric. Traffic measurements have shown that the distribution of the amount of data transferred by a TCP connection exhibits a long tail, i.e. a tail that decays more slowly than exponentially. The key idea is to fit a finite mixture of exponentials to a given long-tail distribution. More precisely, since our flow size is expressed as a discrete number of full-size packets, we use a mixture of n geometric distributions to fit the actual distribution for the number l of full-size packets transferred by TCP connections:

$$H_n(l) = \sum_{i=1}^{n} \alpha_i\, q_i\, (1-q_i)^{l-1} \tag{2}$$

This decomposition allows us to apply our TCP model to study each geometric component separately, obtaining a partial batch size distribution as described in Section 3-A.2. Finally we add up the arrival rates of batches of all possible sizes derived for each geometric component. The aggregate arrival rate of batches of size i is denoted by λ_i. The resulting aggregate batch size distribution is denoted by β.

The reader is referred to [17] for the details regarding the fitting procedure, which applies to a large class of distributions, especially the long-tail distributions that characterize Internet traffic. The authors of [17] provide the theoretical basis for this technique, as well as a recursive algorithm for approximating a long-tail distribution with a given number of exponential components.

3-B Queuing Model

The simplest way to obtain the queue length distribution is to use the classic $M^{[X]}/M/1$ model, in which the batch size X is distributed according to the distribution β computed by the TCP model (Section 3-A). We find that this simple model alone accurately predicts the queue length distribution while providing a conservative estimate (see fig. 2). The assumption of a Poisson arrival of batches is justified by the presence of a large number of flows started at random times and with different RTTs.

In this Section we introduce a novel queuing system, indicated as ${M_S}^{[X]}/M/1$, that provides a better estimate of the queue length distribution in comparison with the $M^{[X]}/M/1$ model. We believe this model represents an important contribution of our work.

Basically we modify the batch size distribution β obtained from the Stochastic model of TCP to account for the fact that packets belonging to the same batch do not enter the queue exactly at the same time, but are 'spread' over time.

The modified batch size distribution is then used in a standard $M^{[X]}/M/1$ model.

The inter-arrival time of two successive packets within a batch must at least equal the minimum transmission time of a packet on the path followed from the source to the queue. For simplicity we consider the case of homogeneous link capacities, so that the inter-arrival time between two packets belonging to the same batch is taken equal to τ, the transmission time of a packet over the link that drains the queue. It is possible to relax this assumption, in order to consider the case of heterogeneous links capacities, but we do not report on this extension.

We can make an analogy with a fluid, considering a batch arrival as a fluid chunk arriving at the queue at a constant fluid rate, which in our case also equals the rate with which the link drains the queue. It is intuitive to understand that the effect on the queue of this "distributed" arrival can be rather different from what is modeled by a $M^{[X]}/M/1$, in which packets within a batch arrive simultaneously.

An approximate solution of the ${M_S}^{[X]}/M/1$ can be obtained in the following way. First we derive the distribution of the aggregate, fluid arrival rate of packets at the queue, indicated by R_i ($i = 0, 1, 2, ...$), where R_i is the probability that at any given time the aggregated fluid rate of packets arriving at the queue equals i. This is equivalent to saying that i batches arrive concurrently, so that i takes only integer values greater than or equal to zero. The assumption of a Poisson arrival of batches allows us to say that R_i has the same distribution as the number of customers in an $M/G/\infty$ queue, which is a Poisson distribution with parameter equal to the traffic intensity ρ regardless of the duration of batches:

$$R(i) = \frac{\rho^i}{i!}\ e^{-\rho} \tag{3}$$

The key idea of our solution is to "modulate" the amplitude of arriving batches using the distribution R_i. Thanks to the PASTA property, when the first packet of a batch arrives at the queue, it finds the queue currently loaded by an aggregated fluid rate distributed according to (3). If we assume that this 'background' rate remains constant during the arrival of the batch we can easily determine how many packets will be in the system when the last packet of the batch arrives. If the 'background' rate is equal to j, the arriving batch increases by one the total arrival rate of packets at the queue, but at the same time the link drains packets from the queue at the same rate, so that during the arrival of the batch the number of packets in the queue changes at a rate equal to j. Thus a transition occurs from the initial state to a state with $i * j$ more packets in the queue, if $j \geq 1$. In the special case in which $j = 0$, it turns out that the batch adds a single packet to the queue (the first packet increases by one the number of packets in the queue, and after that for each new packet that arrives another one is drained by the link, so nothing changes). Finally, transition rates have to be adjusted in order to respect the average arrival rate of packets at each state: if a batch of size i adds k packets to the system with probability p (depending on the fluid rate encountered), and batches of size i arrive with rate λ_i, that transition occurs with rate $\lambda_i\ p\ i/k$. A comparison of Markov chains corresponding to the $M^{[X]}/M/1$ and ${M_S}^{[X]}/M/1$ models in the case of batch arrivals of size three is shown in Fig. 5.

We observed that our solution of the ${M_S}^{[X]}/M/1$ model

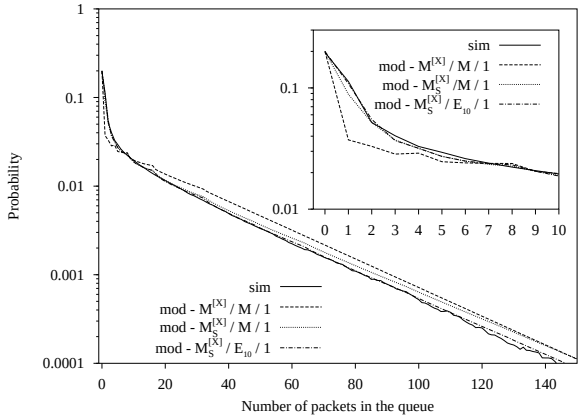

Figure 4: Comparison of queuing delay distributions obtained with a simple $M^{[X]}/M/1$ and more sophisticated models

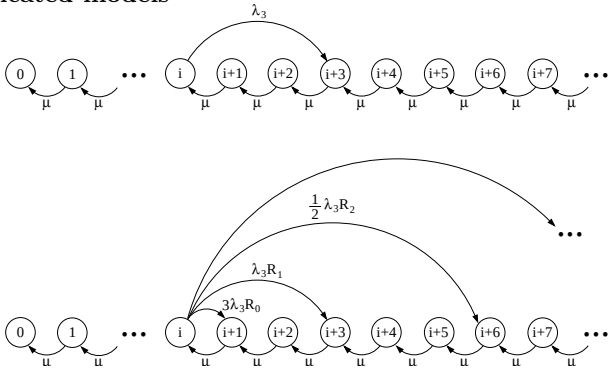

Figure 5: Comparison of Markov chains corresponding to the $M^{[X]}/M/1$ model (upper part) and the ${M_S}^{[X]}/M/1$ model (bottom part)

always provides more accurate results than the $M^{[X]}/M/1$ model. Results for the network setup described in Section 1 are shown in Fig. 4. Even better results can be obtained by considering that service times should be deterministic, rather than exponentially distributed. Unfortunately the exact solution of a ${M_S}^{[X]}/D/1$ is computationally very expensive in the case of a complex batch size distribution as computed by our model. We resorted to approximating the deterministic service time D with an Erlang distribution E_n, that still allows a simple Markovian analysis of the system, at the cost of higher computational complexity for increasing values of n. As shown in Fig. 4, a value of $n = 10$ is able to correct the curve, producing a distribution almost indistinguishable from that obtained by simulation.

Looking only at the queue length distribution, the improvements achieved with our more sophisticated queue models may appear to be marginal, but if we look at the average queue length, which is surely an important metric, we see that the ${M_S}^{[X]}/M/1$ model and even more the ${M_S}^{[X]}/E_{10}/1$ model provide results much better than the $M^{[X]}/M/1$ model. A comparison of results for the average queue length is reported in Table 1 for four different values of link utilization. The "sim" column contains average values and 95% confidence intervals obtained by simulations. The other columns report analytical results, with relative errors with respect to

ρ	sim	$M^{[X]}/M/1$	$M_S{}^{[X]}/M/1$	$M_S{}^{[X]}/E_{10}/1$
0.3	0.917 (± 0.015)	2.293 (150.0 %)	0.997 (8.7 %)	0.939 (2.4 %)
0.6	4.950 (± 0.128)	8.025 (62.1 %)	5.527 (11.6 %)	5.122 (3.5 %)
0.8	16.696 (± 0.698)	21.401 (28.2 %)	18.420 (10.3 %)	16.979 (1.7 %)
0.9	41.091 (± 2.887)	48.145 (17.2 %)	45.617 (11.0 %)	41.819 (1.8 %)

Table 1: Comparison of average queue lengths (in packets) predicted by different queue models

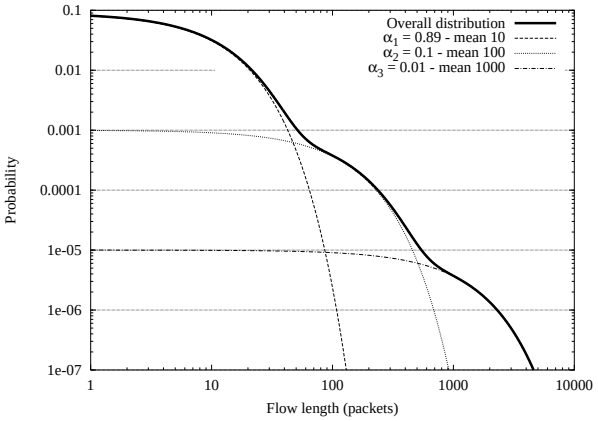

Figure 6: Flow length distribution resulting from a mixture of three geometric components

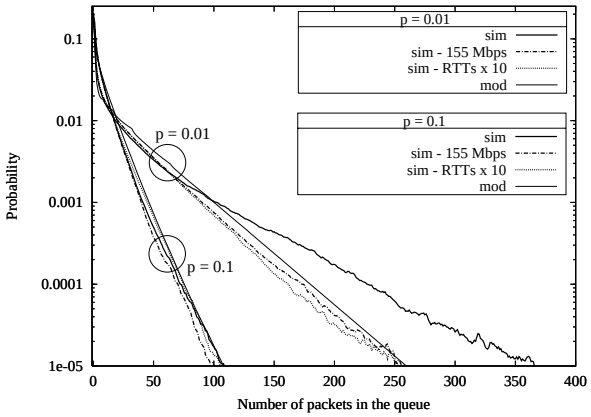

Figure 7: Queue length distributions when the flow size is taken from a mixture of three geometric components, p equal to 0.01 or 0.1

the central value obtained from simulation indicated within brackets. The significant improvement obtained by the more sophisticated models is due mainly to the fact that the probability of finding just a few packets in the queue is better estimated by 'spread' batch arrivals than by 'instantaneous' batch arrivals, as shown in the insert of Fig. 4. Moreover, if one wants to solve the case of a finite buffer size, the correct model to be adopted is a $M_S{}^{[X]}/M/1/B$, not a $M^{[X]}/M/1/B$, as we will point out later in Section 4.

4. SIMULATIONS

We assessed the robustness of our model running a number of simulation experiments in which we studied the sensitivity of different parameters on the resulting queue length distribution: link utilization, loss event probability, loss correlation, link capacity, round trip times, heterogeneous link capacities, maximum window size, delayed acknowledgement option, average flow length (geometrically distributed), long-tail flow length distributions, number of router interfaces (parameter N_2 on Figure 1), tandem network topology, connection inter-arrival time distributions (Weibull), simultaneous opening of connections. This extensive study allowed us to say that our model is accurate and robust under a variety of parameters settings. However, due to lack of space we are not able to show results for all of the experiments (see [1] for further results).

In particular we will not show results for those cases in which analytical results are very close to those obtained from simulation. Instead we point out the most important limitation of our approach that we have found.

Our model accounts precisely for the strong correlation of packet transmissions within the same window of data, but neglects the correlation between the size of batches sent by the same source in successive RTTs. As a consequence, under particular circumstances the analysis underestimates the queueing delay. For example, Figure 7 shows what occurs in our scenario when the flow size is taken from a mixture of three geometric distributions with means 10, 100 and 1000 packets and probabilities $\alpha_1 = 0.89$, $\alpha_2 = 0.1$ and $\alpha_3 = 0.01$, respectively, while the loss event probability is equal to either 0.01 or 0.1. The actual flow length distribution is shown on a log-log scale in Figure 6.

In the case of event loss probabilities equal to 0.01 the model underestimates the slope of the queue length distribution. However, if we increase the number of active flows while maintaining the same traffic intensity, the additional correlation of the traffic that is not captured by the model disappears, and simulation results agree again with the model prediction. To show this phenomenon we ran the same simulation either increasing the channel speed from 28 to 155 Mbps, or multiplying by ten the propagation delays at the transmitters' side. Both of these modifications produce the effect of altering the average number of active connections, establishing our claim. Interestingly, when the loss event probability equals 0.1 the model prediction is already accurate without changing the network parameters. This can be explained by observing that increasing the loss event probability is actually another way of increasing the number of active flows; moreover, loss events break the correlation between successive batches belonging to a flow. We point out that when the assumption of uncorrelated batch sizes holds, the queue length distribution is insensitive to both round trip times and channel speed (at the same ρ).

The discrepancy that we observe between simulation and analysis is an effect of long range dependence in the input traffic of the queue. However it does not seem to be due to the 'long' tail of the flow length distribution. We studied

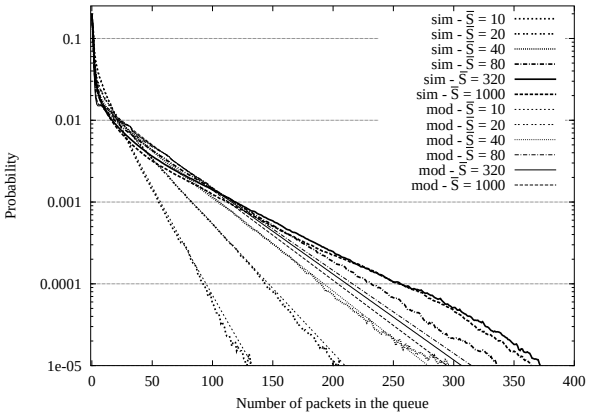

Figure 8: Queue length distribution resulting from geometric flow length distributions with increasing average

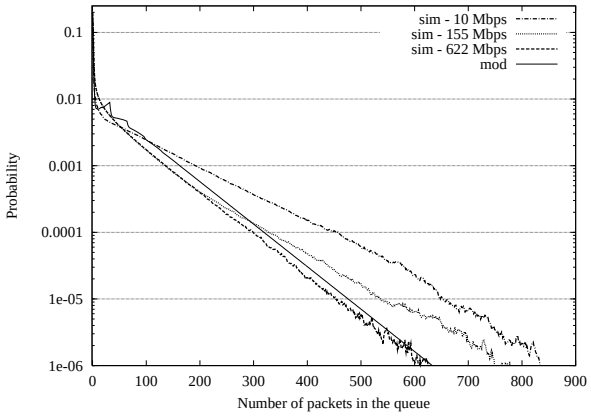

Figure 9: Queue length distributions with $\bar{S} = 320$, $p = 0$, and different link speeds

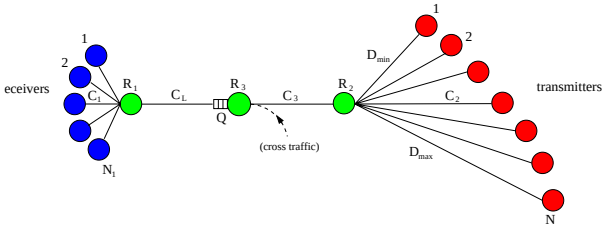

Figure 10: Simple example of tandem network

the impact of flow length distributions consisting of a single geometric component, and obtained the results shown in Figure 8, for the same value of $\rho = 0.8$ and channel speed equal to 28 Mbps. The model is accurate for flow sizes up to $\bar{S} = 40$ packets. A discrepancy shows up using $\bar{S} = 80$, which increases when $\bar{S} = 320$, but remains essentially the same when $\bar{S} = 1000$. Actually, the model predicts that the maximum burstiness is already achieved when $\bar{S}$ is about 80. This is due to the first slow start phase, during which the window rapidly grows up the maximum window size. Simulations confirm that increasing the mean size of the flow length distribution (and its variance) does not result in stronger and stronger correlations. This is because the batches produced by very long flows are in any case limited by the maximum window size. This fact suggests that, contrary to a common belief, the heavy tail of real flow length distributions does not play an important role on the queue behavior, if we use TCP as the transport protocol. An extreme case is shown in Figure 9. Here we assume a loss event probability equal to zero and a single geometric component with $\bar{S} = 320$. Since without loss the behavior of TCP is deterministic, the batch size distribution used in the queue model is definitely correct. However there are strong correlations in the size of batches arriving at the queue, due to flows that reach the maximum window size and stay there until completion. In fact it is necessary to increase the channel speed up to 622 Mbps to see these correlations disappear.

In order to discuss the applicability of the model to more complicated network scenarios than that depicted in Figure 1, we consider the tandem network shown in Figure 11, which is essentially the same topology of Figure 1 with the addition of link C_3. This provides a stress test to the naïve approach of solving each queue in isolation: in fact the presence of traffic arriving at router R_3 from a different interface (indicated in the Figure as 'cross traffic') would make the independence hypothesis more reliable. We obtained from simulation the queue length distribution at Q for different values of the capacity C_3 (the traffic intensity on link C_2 is always equal to 0.8), in order to understand what kind of error is produced solving the queues in isolation. Note that if $C_3 \leq C_L$ the queue does not even form. Results are shown on Figure 11 and reveal that the error becomes marginal as soon as the capacity C_3 is a little higher than capacity C_L. Moreover, the independence assumption always leads to a pessimistic prediction, which is a desirable property in the context of network design and management.

In the presence of a large amount of cross traffic on the paths of the flows, one may think that intermingling of packets belonging to different flows, with the consequent separation of packets belonging to the same batch, would mitigate the strong correlations that we have found in the traffic produced by the TCP sources. On the other hand, the *ack compression* phenomenon [22] is expected to exacerbate the burstiness of packet transmissions. Finally, heterogenous link capacities should be taken into account. Whatever happens, it is intuitive to understand that a queue model with instantaneous batch arrivals will always produce a conservative analysis, provided that there are enough active flows to assume uncorrelated batch sizes.

Finally we want to show that the loss model on which we built our stochastic model of TCP (see Section 3-A.1 is general enough to deal with the important case in which losses are introduced by a drop-tail queueing discipline, which is still widely used in the Internet. For this purpose we ran simulations in which losses are not artificially introduced on the peripheral links but result from buffer overflow at the access router R_2 (see Figure 1). We obtained from simulation the loss event probability required to compute the transition probabilities of our TCP model, and we varied the parameter P_s that allows us to consider all values of loss correlation within the same window of data. Note that now the batch size distribution required by the queue model is that of groups of packets flowing *before* the point in which

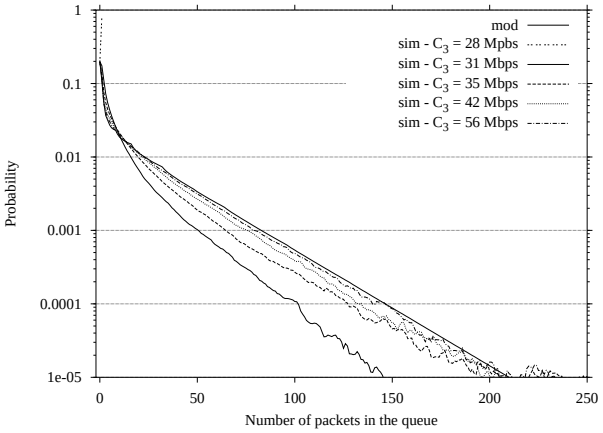

Figure 11: Queue length distributions obtained in the tandem network scenario changing the capacity of link C_3

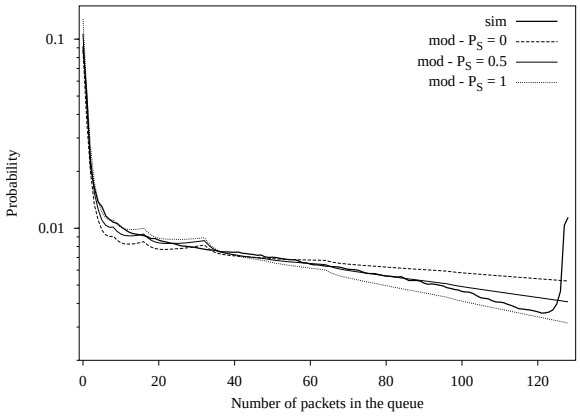

Figure 12: Queue length distributions according to simulation and the $M^{[X]}/M/1/B$ model for different values of P_s

losses occur (the ingress of the drop-tail queue). To solve the queue we simply found the numerical solution of the $M^{[X]}/M/1/B$ queue to which we reduce the analysis of the more precise ${M_S}^{[X]}/M/1$ (Section 3-B), although this is only an approximation, as we will show.

There are two methods for deriving the loss event probability p from the simulation. The first is based on the definition of *loss event* introduced in Section 3-A.1 and requires to simply count the number of loss events occurred during the simulation. We refer to this as estimate p_1. The second method is based on the following consideration. A flow that transfers i packets completes without loss with probability s^i, where $s = 1 - p$, if we assume that loss events occur independently according to a Bernoulli process (this is indeed an assumption of our model). From simulation it is easy to compute the fraction f of the flows that complete without losses. Assuming a geometrical distribution of flow size with parameter q, we have $f = \sum_{i=1}^{\infty} q(1-q)^{i-1}\ s^i$, and after some algebraic manipulations we obtain a different estimate p_2 of loss event probability as

$$p_2 = \frac{q - qf}{q - qf + f} \tag{4}$$

According to our simulations, p_1 and p_2 almost coincide, which validates our assumption that loss events are described by a Bernoulli process. In our scenario we set the buffer size B to 128 packets, and ran a simulation at 90% link utilization with a flow size geometrically distributed with an average of 60 packets. We obtained from simulation $p_1 = 0.0036$ and $p_2 = 0.0033$. Using p_1 into our TCP model and solving the ${M_S}^{[X]}/M/1/B$ queue resulted in the queue length distributions shown in Figure 12 for three different values of P_s. We observe that the agreement with simulation is quite good, and that P_s does not play an important role, the best choice of this parameters lying in between the two extreme values zero and one. Nevertheless the distribution obtained from simulation exhibits a puzzling peak at the maximum value of buffer size that cannot be obtained from a queue of type $M^{[X]}/M/1/B$. In fact the correct model of the queue should be the finite buffer version of the ${M_S}^{[X]}/M/1$ model introduced in Section 3-B. However we do not present this model here due to lack of space. We simply state that it is possible to obtain analytically also the final peak, which is again a consequence of the fact that packets within a batch do not enter the buffer simultaneously, but are spread over time (see [1] for details).

5. MEASUREMENTS

We compared analytical results also with real measurements taken on the access link of our campus network. This is only a qualitative validation, because the examined scenario was no more 'under control' as a simulation with *ns*.

We performed one-way delay measurements over the 28 Mbps link connecting the internal LAN of our university to the Internet, following the recommendations of [23] and [24]. The average link utilization produced by incoming traffic from the Internet was obtained with MRTG, a passive monitoring system widely used throughout the world [26]. Our measurements consisted of sending a poisson stream of probe packets, at a rate of 20 probes per second, for a time interval of ten minutes. This interval was considered to be a good compromise in terms of (i) assuming the traffic stationary, and (ii) obtaining enough points to estimate the delay distribution. We collected several traces at different link utilizations, each 'trace' containing about 12000 samples.

We adopted the algorithm presented in [25] to estimate and remove a clock skew between the sender and the receiver, that were not synchronized. After that, we have assumed that the variable part of the total delay accumulated by the probes over the link is completely due to queuing delay at the buffer of the router.

Examining off-line the trace of all of the packets traversing the link by means of a tool developed at our university [27], we obtained a number of statistics used to correctly parameterize the model and to run simulations of the same system. Among them the most important were the following: the fraction of the traffic volume carried by TCP (about 95%), the distribution of the most typical values of Maximum Segment Size, the distribution of the typical values of maximum window size, the flow length distribution of incoming flows, the packet size distribution, and a rough estimate of the loss event probability based on out-of-sequence bursts of data in the received packet stream. Due to lack of space we are not

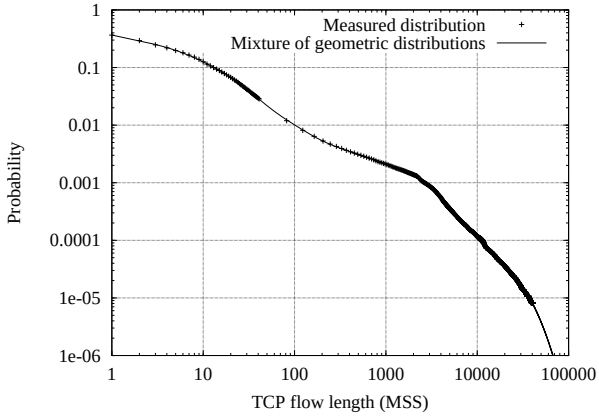

Figure 13: ccdf of the flow length of incoming TCP connections as a number of data segments

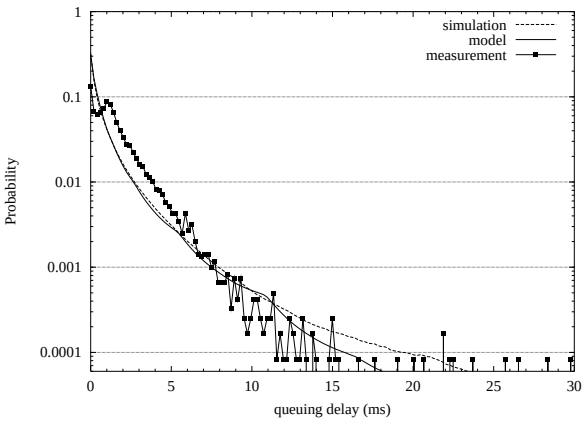

Figure 14: Comparison of queuing delay distributions at 69% link utilization

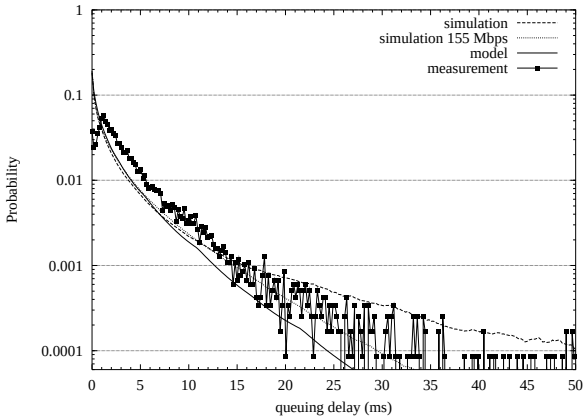

Figure 15: Comparison of queuing delay distributions at 82% link utilization

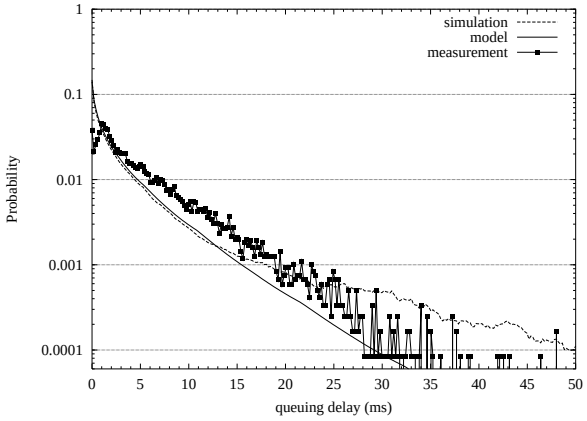

Figure 16: Comparison of queuing delay distributions at 86% link utilization

able to report here on all of these statistics[5]. In Figure 13 we show the complementary cumulative distribution function (ccdf) of the flow length of incoming flows expressed as a number of data segments, together with the fitting of a mixture of seven geometric distributions. The distribution is highly skewed, with 64% of the connections transferring less than one full-size MSS and a typical long tail clearly visible on the log-log scale. The average number of data packets per flow turns out to be 14.75.

Finally Figures 14, 15 and 16 show the queuing delay distributions obtained from analysis, measurements and simulation for different values of link utilization. A discrete distribution for the queuing delay in Q was derived using a constant bin size equal to the transmission time on the link of an average-size packet, which is 709 bytes. This corresponds to about 0.2 ms over a 28 Mbps link. Simulation results are obtained using the network setup of Figure 1. Losses are introduced randomly at the transmitters' side using the loss model described in section 3-A.1, with a loss event probability $p = 0.001$ and $P_s = 0.7$. For each connection we simulated also three-way handshake and connection termination. TCP receivers use the delayed acknowledgement option.

We briefly report also on the complexity of the model solution. We need seven geometric components to model the flow length distribution, and we consider four different maximum window sizes (8, 16, 32 and 64 kbytes), so that 28 realizations of the TCP model described in Section 3-A are required to account for all possible combinations. The model used to solve the queue is the ${M_S}^{[X]}/M/1$. On a 333 Mhz pentium II machine the execution time is less than one second.

Away from the origin the analytical distributions always exhibit a geometric decay. Since we have considered a 28 Mbps channel, we expected to see a discrepancy between model and simulation. On Figure 15 we have also reported the simulation curve obtained in the case of a 155 Mbps link (at the same traffic intensity).

Measurement traces (including other traces not reported here) can exhibit quite different behaviors for similar values of link utilization. For example we can see that the trace at 82% utilization deviates from a geometric decay much more than the trace collected at 86% utilization. We argue that this is due to the extreme variability of a traffic carried by TCP connections with a long-tail flow length distribution:

[5]Further details on the measurement setup, collection of traces and model parameterization can be found in [1]

while LRD effects show up regularly on simulations run for a sufficiently long time, they occasionally appear on traces collected for just ten minutes.

6. CONCLUSION

We conclude summarizing the main results presented in this paper. In order to study analytically the behavior of a queue loaded by Internet traffic, any packet-level model of type $GI/GI/1$ fails, because it neglects the strong correlations of packets sent by the same TCP source. We discovered that a model with batch arrivals that consider together the packets sent every RTT is able to capture most of the traffic correlations. Additional correlations tend to disappear increasing the number of active flows. We proposed a simple stochastic model to obtain the correct batch size distribution in the case of a fairly general long-tail distribution of the flow length. We also described a queue model in which packets within a batch do not enter the queue simultaneously, but are equally spaced over time. Comparing with simulation results, our model seems to be able to capture what really happens in the queue under a variety of parameter settings. We also compared the model predictions with actual measurements of queueing delay on an Internet link, showing that our technique can be successfully applied to study the behavior of a queue in a real network scenario.

7. REFERENCES

[1] M. Garetto, "Modelling, Simulation and Measurements of Queueing Delay under Long-tail Internet Traffic," extended version, `http://www1.tlc.polito.it/~garetto/pub/tcpreport.ps.gz`

[2] `http://www1.tlc.polito.it/~garetto/pub/batch.c`

[3] L. Kleinrock, "Queueing Systems, Vol I: Theory," John Wiley and Sons, New York, NY, 1975.

[4] V. Paxson, S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling," In *IEEE/ACM Transactions on Networking,* 3(3):226–244, June 1995.

[5] W. Willinger and V. Paxson, "Where Mathematics meets the Internet," *Notices of the American Mathematical Society*, 45(8):961–970, August 1998.

[6] M. E. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," In *IEEE/ACM Transactions on Networking,* 5(6):835–846, December 1997.

[7] M. Grossglauser and J. Bolot, "On the Relevance of Long Range Dependence in Network Traffic," In *IEEE/ACM Transactions on Networking,* 7(5):629–640 (1999).

[8] A. Feldman, A. Gilbert, P. Huang and W. Willinger, "Data networks as cascades: Explaining the multifractal nature of Internet Wan traffic," In *ACM SIGCOMM '98*, pp. 42–55, Vancouver, Canada, 1998.

[9] A. Feldman, A. Gilbert, P. Huang and W. Willinger, "Dynamics of IP traffic: A study of the role of variability and the impact of control" In *ACM SIGCOMM '99*, pp. 301–313, 1999.

[10] A. Erramilli, O. Narayan, and W. Willinger, "Experimental queuing analysis with long-range dependent packet traffic," In *IEEE/ACM Transactions on Networking,* 4(2):209–223, April 1996.

[11] A. Erramilli, O. Narayan, A. Neidhardt "Performance Impacts of Multi-Scaling in Wide Area TCP/IP Traffic" In *IEEE INFOCOM '00,* Tel Aviv, Israel, March 2000.

[12] V. Ribeiro, R. Riedi, M. Crouse, R. Baraniuk, "Multiscale Queuing Analysis of Long-Range-Dependent Network Traffic," In *IEEE INFOCOM '00,* Tel Aviv, Israel, March 2000.

[13] S. Vanichpun, A. Makowski, "Positive correlations and buffer occupancy: Lower bound via supermodular ordering," In *IEEE INFOCOM '02,* New York, NY, June 2002.

[14] N. Cardwell, S. Savage, T. Anderson, "Modeling TCP Latency," In *IEEE INFOCOM '00,* Tel Aviv, Israel, March 2000.

[15] S. B. Fredj, T. Bonald, A. Proutiere, G. Régnié, J. Roberts, "Statistical Bandwidth Sharing: A Study of Congestion at Flow Level," In *ACM SIGCOMM '01*, pp. 111-122, (San Diego, CA), September 2001.

[16] C. Barakat, P. Thiran, G. Iannaccone, C. Diot and P. Owezarski, "A flow-based model for Internet backbone traffic," In *ACM Internet Measurement Workshop,* Marseille, November 2002.

[17] A. Feldmann, W. Whitt, "Fitting Mixtures of Exponentials to Long-Tail Distributions to Analyze Network Performance Models," In *IEEE INFOCOM '97*, pp. 1096–1104, 1997.

[18] T. Bu and D. Towsley "Fixed Point Approximation for TCP behavior in an AQM Network," In *ACM SIGMETRICS '01,* pp. 216–225, 2001.

[19] V. Misra, W. Gong and D./, Towsley, "Stochastic Differential Equation Modeling and Analysis of TCP Windowsize Behavior," In *Performance '99*, Istanbul, Turkey, October 1999.

[20] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation," In *ACM SIGCOMM '98*, 28(4):303–314, September 1998.

[21] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, F. Tobagi, C. Diot, "Analysis of Measured Single-Hop Delay from an Operational Backbone Network," In *IEEE INFOCOM '02,* New York, NY, June 2002

[22] L. Zhang, S. Shenker and D. D. Clark, "Observations and Dynamics of a Congestion Control Algorithm: The Effects of Two-Way Traffic," In *ACM SIGCOMM '91*, pp. 133–147, 1991

[23] V. Paxson, G. Almes, J. Madhavi, M. Mathis, "Framework for IP performance Metrics", RFC 2330, May 1998

[24] G. Almes, S. Kalidindi, M. Zekauskas, "A one-way Delay Metric for IPPM", RFC 2679, September 1999

[25] S. Moon, P. Skelley, and D. Towsley, "Estimation and removal of clock skew from network delay measurements," In *IEEE INFOCOM '99*, pp. 227–234, New York, March 1999

[26] "MRTG - the Multi Router Traffic Grapher", `http://people.ee.ethz.ch/~oetiker/webtools/mrtg/`

[27] M. Mellia, A. Carpani, and R. Lo Cigno, "`Tstat` web page," `http://verza.polito.it/index.html`