Application of the many sources asymptotic in downscaling Internet-like Networks

Fragkiskos Papadopoulos, Konstantinos Psounis
University of Southern California
E-mail: fpapadop, kpsounis@usc.edu.

***Abstract*— In our earlier work [1], [2] we have presented two methods to *scale down* the *topology* of the Internet, while preserving important performance metrics. We have shown that the methods can be used to greatly simplify and expedite performance prediction. The key insight that we have leveraged is that only the congested links along the path of each flow introduce sizable queueing delays and dependencies among flows. Based on this, we have shown that it is possible to infer the performance of the larger Internet by creating and observing a suitably scaled-down replica, consisting of the congested links only. However, two main assumptions of our approach were that uncongested links are known in advance, and that the queueing delays imposed by such links are negligible.**

In this paper we provide rules that can be used to identify uncongested links when these are not known, and we theoretically establish the conditions under which the negligible queueing delay assumption is valid. In particular, we first identify scenarios under which one can easily deduce whether a link imposes negligible queueing by inspecting the network topology. Then, we identify scenarios in which this is not possible and use known results based on the large deviations theory to approximate the queue length distribution. Finally, we use this approximation to decide which links are uncongested, and show that in the many-sources limit the queueing delays of uncongested links are indeed negligible. Our results are verified using simulations with TCP traffic.

I. Introduction

Understanding the behavior of the Internet and predicting its performance are important research problems. These problems are made difficult because of the Internet's large size, heterogeneity and high speed of operation.

Researchers use various techniques to deal with these problems: modeling, *e.g.* [3], [4], [5], [6], measurement-based performance characterizations, *e.g.* [7], [8], [9], [10], [11], and simulation studies, *e.g.* [12], [13], [14], [15]. However, these techniques have their limitations.

First, the heterogeneity and complexity of the Internet makes it very difficult and time consuming to devise realistic traffic and network models. Second, due to the increasingly large bandwidths in the Internet core, it is very hard to obtain accurate and representative measurements. And further, even when such data are available it is very expensive and inefficient to run realistic simulations at meaningful scales.

To sidestep some of these problems, Psounis et al. [16], [17], [18] have introduced a method called SHRiNK, that predicts network performance by creating and observing a *slower* downscaled version of the original network.[1] In particular, SHRiNK downscales link capacities such that, when a sample of the original set of TCP flows is run on the downscaled network, a variety of performance metrics, *e.g.* the end-to-end flow delay distributions, are preserved.

This technique has two main benefits. First, by relying only on a sample of the original set of flows, it reduces the amount of data we need to work with. Second, by using actual traffic, it short-cuts the traffic characterization and model-building process. These in turn, expedite simulations and experiments with testbeds, while ensuring the relevance of the results. However, this technique did not solve the important problem of having to work with large and complex network topologies.

With the above problem in mind, we have proposed in [1], [2] two methods that can be used to scale down the topology of the Internet, while preserving the same performance metrics and having the same benefits with SHRiNK.[2] In particular, by defining a link to be congested if the link imposes packet drops or significant queueing delays, we have shown that it is possible to infer the performance of the larger Internet by creating and observing a suitably scaled-down replica, consisting of the congested links only. Further, based on the observation that the majority of backbone links are uncongested [19], [20], [21], [22] we have demonstrated that these methods can be used in practice, to dramatically simplify and expedite performance prediction. However, two main assumptions of our approach were that congested links are known in advance (*e.g.* by utilizing a performance measurement tool), and that the queueing delays imposed by uncongested links can be completely ignored.

This paper complements our earlier work. In particular, we keep the assumption that we know all the congested links that cause packet drops, but we relax the requirement that we know which of the other links can be considered as uncongested, i.e. of not imposing significant delays. Then, we provide rules to identify uncongested links by either inspecting the network topology, or whenever this is not possible, by using a known model from the large deviations theory (based on Fractional Brownian Motion (FBM)), to approximate the queue distribution. Our motivation for this is that while packet drops may be easily detected by a monitoring tool, accurately measuring queueing delays in high-speed backbone networks is quite difficult [8], [23], [19]. Further, we also study the

[1] SHRiNK: Small-scale Hi-fidelity Reproduction of Network Kinetics.

[2] We called the methods DSCALEd (Downscale using delays), and DSCALEs (Downscale using sampling).

conditions under which queueing delays become negligible, and use the aforementioned model to theoretically justify our arguments.

Using the large-deviations model in practice, requires knowledge of the average λ and of the variance σ^2 of the *packet* arrival process on every link of interest. However, measuring the traffic at the packet level to determine these parameters in high-speed backbone routers, has been proven to be a very difficult task, *e.g.* [19], [20]. One may argue that it may be easier to measure the exact queueing delays and hence that the model is of no practical use.

In this paper we show that both λ and σ^2 can be easily and accurately inferred from *flow-level* information. Given that it is much easier to monitor flows than to monitor packets in a router, *e.g.* [19], [20], we argue that the model can be of great practical use. This argument is further strengthened by the fact that information on flows can be either collected on the link we want to study or at the edges of a backbone network. Collecting flow information at the edge routers and combined with their routing information, will give us information on each link of the network. This alleviates the burden of having to monitor many links and makes the measuring procedure scalable. [3]

While an expression for λ is quite intuitive and has been derived in the past, *e.g.* in [19], as the product of the average flow arrival rate and of the expected flow size, estimating σ^2 is much more involved. In this paper we derive a new expression for σ^2. What distinguishes our expression from earlier ones [19], [20], [25] is that it requires less flow-level information and it has been derived without any assumptions, by explicitly taking into consideration the TCP feedback mechanism and long-range dependence.

The rest of the paper is organized as follows. In Section II we briefly review the main idea in scaling down the topology of the Internet. Further, we identify the scenarios under which one can easily deduce whether a link imposes negligible queueing, by just inspecting the topology. For the scenarios in which this is not possible, we review in Section III the large-deviations model we will be using to approximate the queue distribution. In Section IV we explicitly identify the conditions that should hold in the context of TCP networks for this model to be valid. In Section V we compute the packet-level information required to use the model, from flow-level information. In Section VI we validate the model and our theoretical arguments using simulations with TCP traffic. In the same section, we also demonstrate how to use the model to decide whether a link can be ignored, when performing downscaling. Comparison with earlier work follows in Section VII, and we conclude in Section VIII.

II. Scaling-down Network Topology

In this section we briefly review the main idea in scaling down the Internet topology. For more details, the interested reader is referred to our published work [2], [1]. Before proceeding, let's clearly define what we mean by an "uncongested" link in the context of downscaling. An uncongested link is a link which: (i) does not impose any packet drops, and (ii) its queueing delays are negligible compared to the total end-to-end delays of the packets that traverse it, *e.g.* one order of magnitude smaller. Most of the backbone links have both of these properties [21], [22], [23], [19], [20], [8], [7].

Now, as an illustrative example, let's consider the topology shown in Figure 1. In this topology we can see two congested links, and two groups of flows, Grp1 and Grp2. [4] Observe that Grp1 traverses only the one congested link, whereas Grp2 traverses both.

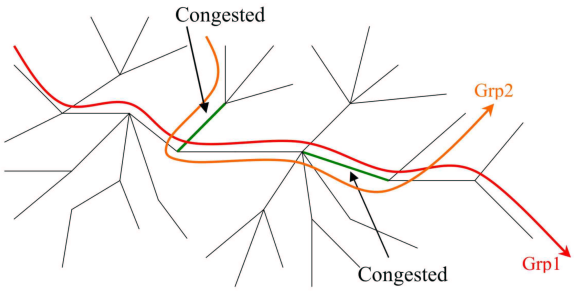

Fig. 1. Original network.

In [1], [2] we have presented two methods (DSCALEd and DSCALEs) that build a scaled replica consisting of the congested links only, along with the groups of flows that traverse them. For the example shown in Figure 1, the resulting scaled replica is shown in Figure 2. Then, the methods adjust the round-trip times in the scaled replica appropriately, such that the performance of the replica can be extrapolated to that of the original network.

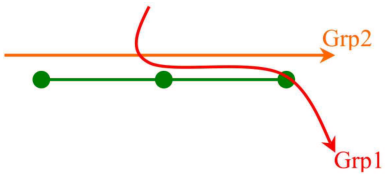

Fig. 2. Scaled replica.

Our main assumption was that we know in advance which links of the original network can be considered as uncongested. However, while links that cause packet drops can be easily identified by a monitoring tool, measuring the queueing delays on every other link to determine whether these are negligible, is clearly a not scalable procedure. Further, it becomes critical in high-speed backbone routers, *e.g.* see [8], [23], [19]. Hence, our motivation in this paper is to provide simple rules that

[3] Tools such as NetFlow already provide flow information in Cisco routers [24].

[4] A group of flows consists of those flows that follow the same network path.

can be used to identify links that impose negligible queueing, without having to explicitly measure their delays.

Our starting point is based on the observation that each link that belongs to the path of a group of flows (*e.g.* the path of Grp1 in Figure 1), can be considered as being part of sub-topologies similar to those shown in Figures 3(i) ... 3(iii). For example, as if it was link Q_2 in Figure 3(i), or link Q_2 in Figure 3(ii), or link Q_1 in Figure 3(iii). (The arrows correspond to groups of flows, the C's are capacities, Src1...SrcN correspond to sources, and Dst1...DstN to destinations.)

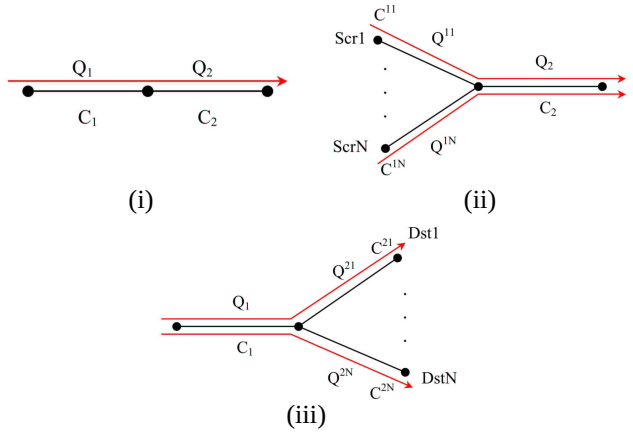

Fig. 3. Toy network topologies used to illustrate when a link can be considered as uncongested by topology inspection.

Now, let's study the conditions under which these links impose insignificant queueing. Let's first concentrate on the topology shown in Figure 3(i). Clearly if $C_1 \leq C_2$ there is not going to be any queueing at Q_2, whereas if $C_1 > C_2$ significant queueing at Q_2 is possible. Now, let's move to the topology shown in Figure 3(ii). If $\sum_{j=1}^{N} C^{1j} \leq C_2$ there is not going to be any queueing at Q_2, but if $\sum_{j=1}^{N} C^{1j} > C_2$ significant queueing at Q_2 is possible. Finally, let's study the topology shown in Figure 3(iii). If $C_1 \leq \sum_{j=1}^{N} C^{2j}$ we can have significant queueing at Q_1. Now, if $C_1 > \sum_{j=1}^{N} C^{2j}$, the C^{2j}'s will regulate the arrivals at Q_1 (through the TCP feedback mechanism) and queueing is expected to be quite low.

Hence, summarizing, the only case where one can decide by inspecting the network, that a link imposes negligible queueing, is the case where the link carries traffic from/to links for which the sum of their capacities is smaller than the capacity of the link.

For the rest of the cases, we will use a model from the theory of large deviations to approximate the queue distribution. In the next section we review this model. For ease of exposition we will be assuming a single link shared by N sources, similar to the one shown in Figure 4 (i.e. without any links attached to either of its edges). It will be clear that the procedure we will be using for deciding whether this link imposes significant queueing, will be directly applicable to the cases discussed earlier.

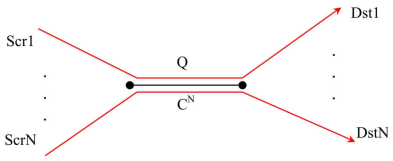

Fig. 4. Single link shared by N sources.

III. Preliminaries

In this section we review the large-deviations model that we will use.

Consider a single link shared by N sources (*e.g.* like the one shown in Figure 4). For simplicity, assume that these sources are homogeneous, i.e. they generate traffic according to the same process. Now, let $A_N(t) = A_N(0,t)$ denote the traffic generated by the superposition of these N sources in the interval $(0,t]$, with $t \in \mathbb{R}^+$ or $t \in \mathbb{Z}$. Further, let λ denote the mean input rate of a single source. Then $E[A_N(t)] = N\lambda t$. Now, let the queue's service rate be scaled with the number of sources, i.e. let the queue drain at rate $C^N \equiv NC$. To ensure stability, we assume that $\lambda < C$.

We are now interested in the steady-state probability $P(Q > \delta B^N)$ of the buffer content exceeding some prespecified level $\delta B^N > 0$, where $0 < \delta \leq 1$ and B^N is again scaled with the number of sources, i.e. $B^N \equiv NB$. Assuming an infinite buffer size, this probability can be expressed in terms of the aggregate cumulative arrival process $A_N(t)$, as follows (*e.g.* see [26] and references therein): [5]

$$P(Q > \delta NB) = P\left(\sup_{t \geq 0}[A_N(t) - NCt] > \delta NB\right). \quad (1)$$

Now, let's assume than $A_N(t)$ is a Gaussian process. Hence, its distribution can be completely characterized by its mean $E[A_N(t)] = N\lambda t$, and its variance $v(t) = \text{Var}[A_N(t)]$. Further, let's assume that the N sources are loosely correlated such that $v(t) \approx N\sigma^2 t^{2H}$, where $\sigma^2 t^{2H}$ is the variance of the traffic from a single source in the interval $(0,t]$, and $H \in [0.5, 1)$. [6] Finally, let's write:

$$I(H) = \frac{(C-\lambda)^{2H}(\delta B)^{2-2H}}{2\sigma^2 K^2(H)}, \quad (2)$$

where $K(H) = H^H(1-H)^{1-H}$.

Using large-deviations theory, it can be shown that the following relationship holds for $P(Q > \delta NB)$, when N is large (*e.g.* see [26], [28]) :

$$P(Q > \delta NB) \leq \exp(-NI(H)). \quad (3)$$

[5] This probability is often used to approximate the corresponding probability in a system with finite buffer equal to NB, when N is large [27].

[6] The exact equality corresponds to the Fractional Brownian Motion (FBM) process with Hurst parameter H. For $H = 0.5$ the process has independent increments, whereas for $H > 0.5$ the increments of the process are long-range dependent [26].

This relation is known in the literature as the *many-sources asymptotic* upper bound and the function $I(H)$ is called the *large deviations rate function*. If N is sufficiently large, Equation (3) is often used to approximate the queue distribution. [7]

The effectiveness of this model has been demonstrated in the context of open-loop networks, *e.g.* see [29], [30], [26], and has been used several times in arguments in the context of TCP networks, *e.g.* see [31], [32], [33]. Next, we review the reasons and clearly identify under which conditions the model is valid in this latter context.

IV. Application to TCP Networks

In this section we identify the reasons why the model discussed above is valid in the context of high-speed TCP networks. For this, let's consider a link/router with capacity NC, buffer size NB, and propagation delay T_{prop}, shared by N source-destination pairs, as shown in Figure 5.

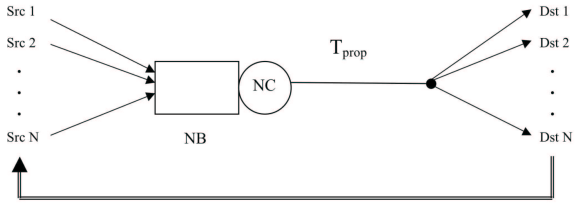

Fig. 5. Simplified link model.

Suppose that each source generates TCP flows according to some process, and that each flow consists of a number of packets that follows some distribution. Of course, packet arrivals at the router are dictated by the TCP feedback dynamics. Finally, assume that the drop-tail policy is adopted by the router.

We argue that the link shown in Figure 5 is a realistic representation of an Internet link since: (i) TCP flows in the Internet arrive at random times and have random sizes, (ii) to keep the utilization fixed, the capacity will usually grow with the number of source-destination pairs sharing the link, (iii) despite many proposals for sophisticated active queue management (AQM) schemes, drop-tail is still the most popular AQM [34], and (iv) routers today are sized according to the rule-of-thumb, where the buffer size equals the bandwidth-delay product [35], and hence the buffer will also grow with N.

If we also assume that N is sufficiently large and that each source generates a large number of flows, the above link will correspond to a high-speed backbone link. Such links usually multiplex hundreds of thousands of TCP flows [19], [36].

Now, it is well documented that if multiple TCP flows share a bottleneck link, they can be synchronized with each other [37], [38], [39]. They are coupled because they experience packet drops at roughly the same time and hence halve their window sizes at the same time. However, flows are not synchronized in a backbone router that carries a large number of them, with various round-trip, processing and startup times. These variations are sufficient to prevent synchronization, and this has been demonstrated in real networks [36], [21], [40].

Under the assumption of a large number of desynchronized TCP flows, it has been recently argued that the evolution of the flow window sizes becomes loosely correlated, and hence the distribution of their sum can be well approximated by a Gaussian distribution. This is justified by the Central Limit Theorem (CLT) as well as also by empirical measurement [36], [33]. The assumption of weak window correlations is strengthened further by the fact that backbone links are generally over-provisioned, (i.e., the network is designed so that a backbone link utilization stays below 50%, in the absence of link failure [22]), and thus drops on such links are rare. (However, this last condition is *not* a requirement for desynchronization to exist.)

Under the above observations, the applicability of the model presented in the previous section seems quite promising, since the model requires that: (i) The link is scaled by the number of sources N, (ii) the arrival process from each source is Gaussian, and (iii) there are no significant correlations between different sources. Further, the model also accounts for long-range dependence in the traffic originating from the same source, which is a well-known characteristic of traffic in the Internet [41], [10], [42].

However, as we observe, using the model requires knowledge of the average λ and of the variance σ^2 of the *packet* arrival process of a source. Further, it also requires knowledge of the parameter H. As mentioned earlier, it is difficult and not scalable to estimate these parameters by monitoring packets on every link of interest. As we have said, we prefer to monitor flows, which is much easier [19], [20].

Thus, in the next section we show how to compute these parameters from flow-level information. Before proceeding however, recall that we are interested in deciding of whether a link that does not impose packet drops, imposes significant delays. Therefore, we will be assuming links that do not impose any packet drops.

V. Parameter Estimation

In this section we use known expressions to compute λ and H, and we derive a new expression for σ^2.

We assume knowledge of some flow-level information on the link of interest. In particular, if there are N source-destination pairs sharing the link, we assume that we know: (i) the flow size distribution $F(s)$ of the flows traversing the link, (ii) the total average arrival rate of flows at the link, denoted by r_N, and (iii) the average of the total number of active flows on the link $E[A_N]$, and of its variance $\text{Var}(A_N)$. [8] This flow-level information can be easily extracted from a router, *e.g.* using NetFlow [24].

We express the average flow arrival rate for a source-destination pair as $r = \frac{r_N}{N}$ and the average number of

[7] Many-sources asymptotics have been derived for other input processes as well. For a nice review of the results we refer the interested reader to [26], [28].

[8] We say that a flow is "active" on a link, if the link belongs to the path of the flow, and the flow has more data packets to send.

active flows for a pair as $E[A] = \frac{E[A_N]}{N}$. Finally, we write $\text{Var}(A) = \frac{\text{Var}(A_N)}{N}$. This last equality assumes that there are no dependencies among different sources, which as mentioned before, is the case for backbone links. We start by giving the expression for λ.

A. Estimating λ

Let S be the random variable representing the size of a flow. Since we know $F(s)$ we can compute the average flow size $E[S]$. Since there are no drops, an intuitive and well-known expression for λ (*e.g.* see [19]) is:

$$\lambda = rE[S]. \tag{4}$$

The relation above states that the average packet arrival rate is equal to the average arrival rate of flows times the average amount of load brought by each flow. Note, that for a system to be stable (in the sense that the number of active flows never grows to infinity) it is required that $rE[S] < C$ [5]. Hence, for the system under study we assume that this holds, which yields $\lambda < C$. Recall, that this last condition is required in order to be able to invoke the model of Section III.

Next, we use another known result to show how one can estimate the parameter H.

B. Estimating H

The long-range dependence of Internet traffic has been shown to be the result of a heavy-tailed flow size distribution [42], [21]. A heavy-tailed distribution is one in which $P(S > s) \sim s^{-\alpha}, 1 < \alpha < 2$, as $s \to \infty$.

At large time-scales, *e.g.* greater than the round-trip time, the parameter H is directly related to the parameter α (called the shape parameter) of the size distribution. According to [42]:

$$H = \frac{3-\alpha}{2}. \tag{5}$$

Since we know $F(s)$, we can use the above Equation to approximate H. Next, we derive an expression for σ^2.

C. Estimating σ^2

The expression for σ^2 is given in the following Theorem:

Theorem 1:

$$\sigma^2 = \frac{E[A]\text{Var}(W) + (E[W])^2\text{Var}(A)}{(E[RTT])^{2H}}, \tag{6}$$

where $E[W]$ is the average congestion window size of a flow and $\text{Var}(W)$ its variance, $E[RTT]$ is the average round-trip time of a flow, and the rest of the parameters as defined earlier.

Proof: Assume that the time is slotted with the duration of slot i be equal to the current round-trip time. Now, denote by P the total number of packets that belong to a source-destination pair in some time-slot. Thus, $P = \sum_{j=1}^{A} W_j$, where A is the random variable representing the number of active flows of a pair in a time-slot, and W_j is the random variable representing the congestion window size of flow j, $j \in \{1...A\}$. By the conditional variance formula [43] we have:

$$\text{Var}(P) = E[\text{Var}(P|A)] + \text{Var}(E[P|A]). \tag{7}$$

Since there are no drops, the W_j's ($j \in 1...A$) are independent of the random variable A. It is then easy to see that:

$$E[\text{Var}(P|A)] = E[A]\text{Var}(W), \tag{8}$$

and:

$$\text{Var}(E[P|A]) = (E[W])^2\text{Var}(A). \tag{9}$$

Now, recall from Section III that $\sigma^2 t^{2H}$ is the variance of the amount of traffic that is injected into the network by a source in the interval $(0, t]$. Denote this amount of traffic by $A_1(t)$, and let $N(t)$ be the number of time-slots elapsed by time t. We can write $A_1(t) = \sum_{i=1}^{N(t)} P(i)$, where $P(i)$ is the random variable representing the number of packets of a pair in slot i.

In steady-state the $P(i)$'s are identically distributed. Accounting for long-range dependence in the sequence $\{P(i), i = 1, 2, ..., N(t)\}$, we can write $\text{Var}(A_1(t)) = (N(t))^{2H}\text{Var}(P) = \sigma^2 t^{2H}$. Now, for t large enough $N(t) = \frac{t}{E[RTT]}$, and hence:

$$\sigma^2 = \frac{\text{Var}(P)}{(E[RTT])^{2H}}. \tag{10}$$

From Equations (7)...(10) we get Equation (6).

■

Now, recall that $E[A]$ and $\text{Var}(A)$ in Equation (6) are known quantities. Hence, what remains to complete the calculation of σ^2 is to compute $E[W]$, $\text{Var}(W) = E[W^2] - (E[W])^2$, and $E[RTT]$.

We begin by $E[RTT]$. Let $E[D]$ be the average number of round-trips that a flow needs in order to complete. Using Little's Law we can write:

$$E[RTT] = \frac{E[A]}{rE[D]}. \tag{11}$$

Since $E[A]$ and r are known, we only need to compute $E[D]$. For this, we proceed as follows.

Suppose that the maximum window size of a flow is W_{max}. We divide flows into two categories: (i) short flows, whose size is less than or equal to $2W_{max}$, and (ii) long flows whose size is larger than $2W_{max}$. Given TCP's AIMD (Additive-Increase-Multiplicative-Decrease) mechanism, this separation implies that a short flow spends its lifetime in slow start, and may send W_{max} packets at most once during its lifetime. We can write:

$$E[D \mid \text{short flow}] = E[\lfloor \log_2 S \rfloor + 1_{[S - \sum_{i=0}^{\lfloor \log_2 S \rfloor - 1} 2^i > 0]} \mid S \leq 2W_{max}], \tag{12}$$

where $1_{[.]} = 1$ if the condition in the brackets is satisfied, and 0 otherwise. Now, long flows spend approximately $\log_2 2W_{max}$

round-trip times in slow-start and then send W_{max} packets per round-trip for the rest of their lifetime. Hence:

$$E[D \mid \text{long flow}] = E[\lfloor \log_2 2W_{max} \rfloor + \lfloor \frac{S - \sum_{i=0}^{\lfloor \log_2 2W_{max} \rfloor - 1} 2^i}{W_{max}} \rfloor + 1_{[R(S)>0]}],$$

where:
$R(S) = S - \left[\sum_{i=0}^{\lfloor \log_2 2W_{max} \rfloor - 1} 2^i + \lfloor \frac{S - \sum_{i=0}^{\lfloor \log_2 2W_{max} \rfloor - 1} 2^i}{W_{max}} \rfloor W_{max} \right]$.
Since we know $F(s)$, we can compute and uncondition the expectations above and find $E[D]$.

Since we have computed the expected flow size and the expected number of rounds a flow needs to complete, it is easy to see that the average window size of a flow is: [9]

$$E[W] = \frac{E[S]}{E[D]}. \quad (13)$$

What remains now is to compute the mean square window size of a flow $E[W^2]$. For this, we first need to find an expression for the expectation, of the sum of the squares of the window sizes that a flow reaches during its lifetime. We denote this expectation by $E[S^*]$. Considering TCP's AIMD mechanism as we did before, and distinguishing again short and long flows we can write:

$$E[S^* \mid \text{short flow}] = E[\sum_{i=0}^{\lfloor \log_2 S \rfloor - 1} (2^i)^2 + (S - \sum_{i=0}^{\lfloor \log_2 S \rfloor - 1} 2^i)^2 \mid S \leq 2W_{max}], \quad (14)$$

$$E[S^* \mid \text{long flow}] = E[\sum_{i=0}^{\lfloor \log_2 2W_{max} \rfloor - 1} (2^i)^2 + \lfloor \frac{S - \sum_{i=0}^{\lfloor \log_2 2W_{max} \rfloor - 1} 2^i}{W_{max}} \rfloor (W_{max})^2 + (R(S))^2 \mid S > 2W_{max}], \quad (15)$$

where $R(S)$ as defined above. As before, knowing $F(s)$, we can uncondition these expectations and find $E[S^*]$. The relation for $E[W^2]$ is now given in the following lemma:

Lemma 1:

$$E[W^2] = \frac{E[S^*]}{E[D]}, \quad (16)$$

where $E[S^*]$ and $E[D]$ as defined earlier.

Proof: Assume again that the time is slotted with the duration of the current slot be equal to the current round-trip time. Now, let Y be the sum of the squares of the window sizes, of all active flows that belong to a pair, i.e. $Y = \sum_{j=1}^{A} W_j^2$. As before, since there are no drops the W_j's ($j \in 1...A$) are independent of the random variable A. We can write:

$$E[Y] = E[W^2]E[A]. \quad (17)$$

Now, $E[Y]$ can be also written as:

$$E[Y] = \lim_{t \to \infty} \frac{\sum_{i=1}^{N(t)} \sum_{j=1}^{A(i)} (W_j^i)^2}{N(t)}, \quad (18)$$

where $N(t)$ is the number of time-slots elapsed by time t as before, $A(i)$ is the number of active flows in slot i, and W_j^i is the congestion window size of flow j ($j \in 1...A(i)$).

Let $F(t)$ be the total number of flows that have completed service within $N(t)$ slots. $E[D]$ can be also expressed as follows:

$$E[D] = \lim_{t \to \infty} \frac{\sum_{i=1}^{N(t)} A(i)}{F(t)}. \quad (19)$$

Further, the average number of active flows in a slot can be written as:

$$E[A] = \lim_{t \to \infty} \frac{\sum_{i=1}^{N(t)} A(i)}{N(t)}. \quad (20)$$

Now, equations (19) and (20) yield:

$$\lim_{t \to \infty} \frac{N(t)}{F(t)} = \frac{E[D]}{E[A]}. \quad (21)$$

Finally, since there are no drops it is easy to see that:

$$E[S^*] = \lim_{t \to \infty} \frac{\sum_{i=1}^{N(t)} \sum_{j=1}^{A(i)} (W_j^i)^2}{F(t)}. \quad (22)$$

Now, from Equations (18), (21) and (22) we can deduce that:

$$E[Y] = \frac{E[S^*]}{E[D]} E[A]. \quad (23)$$

From Equations (23) and (17) we get Equation (16). ■

We have now computed all the parameters required to estimate σ^2.

VI. Simulations

We now present simulation results using the ns-2 simulator [44] in order demonstrate how the model of Section III can be used in downscaling, and to verify our theoretical arguments. We use the setup shown in Figure 5. Each source-destination pair generates TCP flows according to a Poisson process at rate $r = 95$flows/sec. [10] The number of data packets S in each flow follows a bounded Pareto distribution with average $E[S] = 11.54$, maximum 10^6, and shape parameter 1.36. (Notice that $rE[S] < C$, and hence the system is stable.) The size of an IP data packet is 1040 bytes, the two-way propagation delay of the link is $2T_{prop} = 100$ms, $C = 10$Mbps $= 1200$packets/sec, and $B = 2T_{prop}C = 120$packets. Finally, $W_{max} = 20$packets and the simulation time of each experiment was 10000sec.

We first start by verifying that the aggregate arrival process can be approximated by a Gaussian distribution. Figures 6(i)

[9] A formal proof for this relation goes along the same lines with the proof of Lemma 1, which we will state shortly.

[10] The flow arrival process does *not* have to be Poisson. We have used this based on the argument in [5] according to which, since network sessions arrive as a Poisson process [45], [46], [10] network flows are *as if* they were Poisson. (In particular, the equilibrium distribution of the number of flows in progress is *as if* flows arrive as a Poisson process.)

and 6(ii) show that this is indeed the case, even for N's as small as 1 and 6 respectively. Note that for $N = 1$ the average number of active flows was approximately $E[A] = 40$, and the packet drop ratio was around 1.2%. This implies that the Gaussian approximation is accurate even when the number of multiplexed flows is relatively small and there are packet drops. This is in agreement with the observations in [36].

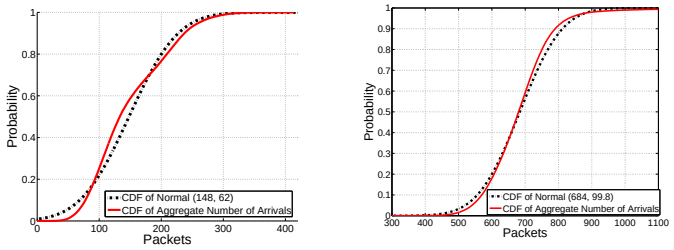

Fig. 6. The commulative distribution function (CDF) of the sum of the aggregate number of arrivals passing through the router during a round-trip time, and its approximation with a Gaussion CDF with the same parameters: (i) N=1, and (ii) N=6.

For $N = 6$, the total average number of active flows is 161.88, and the percentage of dropped packets 0.02%. In this case, because there are more flows active in the system, the Gaussian approximation is more accurate. This is evident from Figure 6(ii). Also, notice that the drop ratio is smaller than the case where $N = 1$. This is in agreement with Equation (3), which implies that for any level $\delta > 0$, as N increases, the probability that the buffer content exceeds δNB decreases. This also means that the queueing delays decrease. Hence, flows spend less time in the system when $N = 6$. This is also verified by observing that the number of active flows per source when $N = 6$ is $E[A] = \frac{161.88}{6} = 26.98$, which is less than the number of active flows when $N = 1$.

Now, let's test whether the model of Section III can be used to decide whether a link imposes negligible queueing delays. Recall that for the purposes of downscaling we are interested in cases where there are no packet drops. As we have observed from the simulator drops stop occurring for $N > 10$. Thus, we will demonstrate results for $N = 11, 16$, and 32. However, we will also show results for $N = 6$, where N is relatively small and there are some drops, just to check how accurate the model is in such scenarios.

We estimate λ, σ^2 and H, using the formulas of the previous section. [11] We compute that $\lambda = 1096$packets/sec and that $H = 0.82$. Now, recall that according to our procedure, in order to compute σ^2 we also need estimates for $E[A]$ and $\text{Var}(A)$. These are extracted from the simulator. The rest of the parameters required to compute σ^2 are: $E[D] = 2.6$rounds (which gives $E[W] = 4.44$packets), and $E[S^*] = 127.5$packets (which gives $E[W^2] = 49$packets). Table I gives the values for $E[A]$,Var(A), $\text{Var}(P)$, and the resulting σ^2, as we vary N.

[11]Recall that we ignore packet drops in our calculations, which occur when $N = 6$.

N	$E[A]$	$\text{Var}(A)$	$\text{Var}(P)$ (packets/RTT)	σ^2 (packets/sec)
6	26.98	46.07	1699	64166
11	25.34	30.00	1334	55838
16	24.83	27.47	1269	54919
32	24.72	25.92	1235	53838

TABLE I

FLOW- AND PACKET- LEVEL STATISTICS AT THE LINK.

Before proceeding, we make some comments regarding the values of σ^2. As we observe from Table I, σ^2 decreases as N increases. This is again in agreement with Equation (3): since queueing delays decrease, the variance of the arrivals (which are regulated by TCP) decreases. Further, we observe that the difference becomes less notable as N increases. This implies that the queueing delays become less and less significant. We verify this next. [12]

Figure 7 shows that the queueing delays indeed become negligible as N increases. Further, it shows that the model is quite accurate in approximating the queue distribution if N is sufficiently large, as expected.

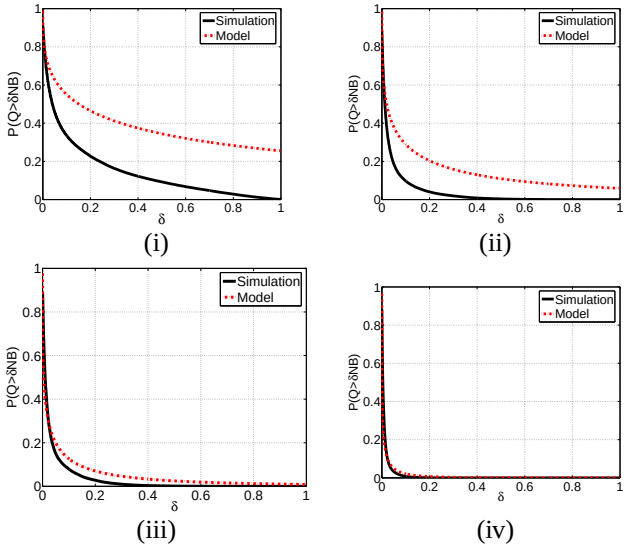

Fig. 7. Queue exceedance probability $P(Q > \delta NB)$ against the buffer level δ: (i) $N = 6$, (ii) $N = 11$, (iii) $N = 16$, and (iv) $N = 32$.

In particular, from the Figure we observe that for all N's the curve given by the model has (approximately) the correct slope for a wide range of values δ. Further, the model captures the speed by which the exceedance probability decays as N increases, and for $N \geq 16$ it accurately predicts the queue distribution. This verifies that our parameter estimation was correct. Further, it validates that the model can be used in the context of TCP networks, and in particular in the context of

[12]Notice that in this scenario, where flows arrive as a Poisson process, one can find the limiting σ^2, by taking the queueing delays to be zero and modeling the system at the flow level as an $M/G/\infty$ queue. The distribution of the number of active flows will be Poisson [5], with parameter given by Equation (11), taking $E[RTT] = 2T_{prop}$. Equation (7) then degenerates to $\text{Var}(P) = E[A]E[W^2] = 1210$, yielding $\sigma^2 = 52818$.

downscaling, to identify links with negligible queueing delays. Next, we summarize our procedure and give specific guidelines of how to use the model.

Application to network downscaling: Suppose that we have a backbone network and we wish to build a scaled-down replica using the downscaling techniques presented in [2]. Recall that according to the methods, the scaled system will consist of all the congested links that cause packet drops (identified by a monitoring tool), and those links that cause significant queueing delays. To identify links with negligible queueing delays and ignore them, we follow the steps below:

1) From the network topology and routing information, we identify and ignore every link for which the traffic it carries is being forwarded from/to links for which the sum of their capacities is smaller that the capacity of the link. (See Section II).
2) For all other links we use a flow-level measurement tool, *e.g.* such as NetFlow [24], to estimate: (i) The flow-size distribution, (ii) the flow arrival rate, and (iii) the average, and the variance of the number of active flows.
3) We use Equations (4)...(6) to compute λ, H, and σ^2 for each of these links, as described in Section V.
4) We use Equation (3) to approximate the queue distribution on each of these links.
5) From the network topology and traffic matrix we calculate for each of these links the average *two-way* end-to-end propagation delay among the groups of flows that traverse them.
6) As a rule-of-thumb, we ignore all those links for which their maximum queueing delay is one order of magnitude smaller than the corresponding two-way end-to-end propagation delay, with probability above 90%.

As an illustrative example, suppose that the average two-way end-to-end propagation delay among the groups of flows traversing the link shown in Figure 5 is 200ms. According to our rules we ignore this link, if the probability that its maximum queueing delay is below 20ms, is larger than 90%. It is easy to see that in our scenario a 20ms queueing delay corresponds to $\delta = 0.2$. As we can see from Figure 7, the model correctly predicts that we can ignore this link for the vast majority of cases of interest (*e.g.* for all $N \geq 16$, where there are no drops).

VII. Related Work

In this Section we review related work on the applicability of the model of Section III, and on estimating σ^2. For related work on network downscaling, we refer the interested reader to [2].

The model presented in Section III has been derived in several studies and its effectiveness has been verified in the context of open-loop networks, *e.g.* see [29], [30], [26]. Its applicability has been also demonstrated for Internet backbone traffic, *e.g.* [41]. And it has been used in this later context by authors, for their theoretical arguments, *e.g.* in [32], [33].

In this study we have shown that this model can be also effectively applied in the context of network downscaling. Further, we have clearly identified the necessary conditions for the model to be applicable, and we have used ns-2 simulations with TCP traffic to validate it.

In contrast to earlier studies that have utilized the model, by extracting its parameters from packet-level information, *e.g.* [41], in this study we have chosen to infer this information from flow-level statistics. In the process, we derived a formula that relates the variance σ^2 of the packet arrival process to some flow-level information. The most relevant to this studies are the ones in [19], [20], [25]. We now explain the main differences of our approach.

First, for their formula derivation, all of these studies have assumed flows that arrive to the system according to a Poisson process. In addition, in [25] the author has also assumed a bufferless link model and modeled the number of active flows as an $M/G/\infty$ queue. During our formula derivation, none of these assumptions have been made. Further, in [19], [20] the notion of "shots" was introduced to describe how flows transmit their packets. To accurately estimate the variance requires correct estimates for the shapes of the shots, which in general requires further measurements. Further in [25] it is assumed that packets of a flow are spread uniformly in time. In contrast, in our study we have not made any assumptions on how flows transmit their packets. We have explicitly taken into consideration TCP's AIMD mechanism and long-range dependence.

Finally, the study in [19], [20], which is the most related to our study, derives a variance formula that requires (in addition to the flow arrival rate), knowledge of the expectation $E[\frac{S^2}{D}]$, where S is the flow size and D the flow duration. This implies that one needs to keep track of flow sizes and their corresponding durations. In our study we still require knowledge of the flow sizes, but we do not need to keep track the corresponding durations. Instead, we only need estimates on the first two moments of the number of flows on a router, which can be easily measured, independently from the flow sizes.

VIII. Conclusion and Future Work

This paper complements our earlier work [1], [2] where two methods were presented to scale down the topology of the Internet, while preserving performance. In particular, we have provided guidelines that can be used to decide whether a link imposes negligible queueing delays, and hence can be ignored when building the scaled replica.

This study is also important independently from network downscaling. In this paper we have also demonstrated how a well-known model from the large-deviations theory can be utilized in practice, and in the process, we have derived a formula that relates the variance of the packet arrival process to flow-level statistics.

Our future work consists of verifying the model and our parameter estimation for larger network topologies under various loads.

REFERENCES

[1] Fragkiskos Papadopoulos, Konstantinos Psounis, and Ramesh Govindan, "Performance preserving network downscaling," in *Proc. of the 38th Annual Simulation Symposium*, 2005.

[2] Fragkiskos Papadopoulos, Konstantinos Psounis, and Ramesh Govindan, "Performance preserving topological downscaling of internet-like networks," *IEEE Journal on Selected Areas in Communications, Issue on Sampling the Internet: Techniques and Applications*, vol. 24, no. 12, 2006.

[3] Y. Liu, F. L. Presti, V. Misra, D. Towsley, and Y. Gu, "Fluid models and solutions for large-scale IP networks," in *Proc. of ACM SIGMETRICS*, June 2003.

[4] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, and P. Owezarski, "A flow-based model for Internet backbone traffic," in *Proc. of ACM SIGCOMM Internet Measurement Workshop*, November 2002.

[5] S. Ben Fredj, T. Bonalds, A. Prutiere, G. Gegnie, and J. Roberts, "Statistical bandwidth sharing: a study of congestion at flow level," in *Proc. of ACM SIGCOMM*, August 2001.

[6] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *Proc. of ACM SIGCOMM*, August 1998.

[7] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and C. Diot, "Packet-level traffic measurements from the Sprint IP backbone," *IEEE Network*, vol. 17, no. 6, November 2003.

[8] K. Papagianaki, S. Moon, C. Fraleigh, P. Thiran, F. Tobagi, and C.Diot, "Analysis of measured single-hop delay from an operational backbone network," in *Proc. of IEEE INFOCOM*, June 2002.

[9] J. Liu and M. Crovella, "Using loss pairs to discover network properties," in *Proc. of ACM SIGCOMM Internet Measurement Workshop*, November 2001.

[10] V. Paxson and S. Floyd, "Wide area traffic: the failure of Poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226–244, June 1995.

[11] J. C. Mogul, "Observing TCP dynamics in real networks," in *Proc. of ACM SIGCOMM*, August 1992.

[12] J. S. Ahn and P. B. Danzig, "Speedup and accuracy versus timing granularity," *IEEE/ACM Transactions On Networking*, vol. 4, no. 5, pp. 743–757, October 1996.

[13] D. Nicol and P. Heidelberger, "Parallel execution for serial simulators," *ACM Transactions On Modeling and Computer Simulation*, vol. 6, no. 3, pp. 210–242, July 1996.

[14] A. Yan and W. B. Gong, "Time-driven fluid simulation for high-speed networks," *IEEE Transactions On Information Theory*, vol. 45, no. 5, pp. 1588–1599, July 1999.

[15] B. Liu, D.R. Figueiredo, Y. Guo, J. Kurose, and D. Towsley, "A study of networks simulation efficiency: fluid simulation vs. packet-level simulation," in *Proc. of IEEE INFOCOM*, April 2001.

[16] K. Psounis, R. Pan, B. Prabhakar, and D. Wischik, "The scaling hypothesis: Simplifying the prediction of network performance using scaled-down simulations," in *Proc. of ACM HOTNETS*, October 2002.

[17] R. Pan, B. Prabhakar, K. Psounis, and D. Wischik, "SHRiNK: A method for scalable performance prediction and efficient network simulation," in *Proc. of IEEE INFOCOM*, March 2003.

[18] R. Pan, B. Prabhakar, K. Psounis, and D. Wischik, "SHRiNK: Enabling scaleable performance prediction and efficient simulation of networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, pp. 975–988, October 2005.

[19] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, and P. Owezarski, "A flow-based model for internet backbone traffic," in *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, 2002.

[20] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, and P. Owezarski, "Modeling internet backbone traffic at the flow level," *IEEE Transactions on Signal Processing, Special Issue on Networking*, vol. 51, no. 8, August 2003.

[21] C. Fraleigh, F. Tobagi, and C. Diot, "Provisioning IP backbone networks to support latency sensitive traffic," in *Proc. of IEEE INFOCOM*, March 2003.

[22] C. Fraleigh, *Provisioning Internet Backbone Networks to Support Latency Sensitive Applications*, Ph.D. thesis, Stanford University, June 2002.

[23] K. Papagiannaki, *Provisioning IP Backbone Networks Based on Measurements*, Ph.D. thesis, University College London, March 2003.

[24] "Cisco IOS netflow," `http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html`.

[25] D.P. Heyman, "Sizing backbone Internet links," *Operations Research*, vol. 53, no. 4, 2005.

[26] A. Ganesh, N. O' Connell, and D. Wischik, "Big queues," *Springer-Verlang*, Berlin, 2004.

[27] A. B. Dieker and M. Mandjes, "Fast simulation of overflow probabilities in a queue with gaussian input," *ACM Trans. Model. Comput. Simul.*, vol. 16, no. 2, pp. 119–151, 2006.

[28] P. Rabinovitch, "Statistical estimation of effective bandwidth," *Master's Thesis, Carleton University*, 2000.

[29] Z. Fan and P. Mars, "Accurate approximation of cell loss probability for self-similar traffic in ATM networks," *Electronic letters*, vol. 32, no. 19, pp. 1749–1751, September, 1996.

[30] I. Norros, "On the use of fractional Brownian motion in the theory of connectionless networks.," *IEEE Journal on selected areas in communications*, vol. 13, no. 6, 1995.

[31] A. Erramilli, O. Narayan, and W. Willinger, "Experimental queueing analysis with long-range dependent packet traffic," *IEEE/ACM Trans. Netw.*, vol. 4, no. 2, pp. 209–223, 1996.

[32] D. Wischik, "Buffer requirements for high-speed routers," in *Proceedings of ECOC*, 2005.

[33] D. Y. Eun and X. Wang, "Performance modeling of TCP/AQM with generalized AIMD under intermediate buffer sizes," in *IEEE International Performance Computing and Communications Conference*, April 2006.

[34] J. Sun, M. Zukerman, King-Tim Ko, G. Chen, and S. Chan, "Effect of large buffers on TCP queueing behavior," in *Proceedings of INFOCOM*, 2004.

[35] Y. Ganjali and N. McKeown, "Update on buffer sizing in internet routers," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 5, pp. 67–70, October 2006.

[36] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing router buffers," in *Proceedings of ACM SIGCOMM*, August 2004.

[37] L. Zhang, S. Shenker, , and D.D. Clark, "Observations on the dynamis of a congestion control algorithm: The effects of two-way traffic," in *Proc. of ACM SIGCOMM*, September 1991.

[38] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, 1993.

[39] L. Zhang and D. D. Clark., "Oscillating behavior of network traffic: A case study simulation," *Internetworking: Research and Experience*, pp. 101–112, 1990.

[40] G. Iannaccone, M. May, and C. Diot, "Aggregate traffic performance with active queue management and drop from tail.," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 3, pp. 4–13, 2001.

[41] L. Yao, M. Agapie, J. Ganbar, and M. Doroslovacki, "Long-range dependence in internet backbone traffic," in *Proc. IEEE International Conference on Communications*, 2003.

[42] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, "Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level," *IEEE/ACM Transactions on Networking*, vol. 5, no. 1, pp. 71–86, February 1997.

[43] S. M. Ross, "Introduction to probability models," *Academic Press, 8th edition*, 2002.

[44] "Network simulator," `http://www.isi.edu/nsnam/ns`.

[45] A. Feldmann, A. C. Gilbert, and W. Willinger, "Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic," in *Proceedings of ACM SIGCOMM*, August 1998.

[46] C. J. Nuzman, I. Saniee, W. Sweldens, and A. Weiss, "A compound model for TCP connection arrivals," in *Proceedings of ITC Seminar on IP Traffic Modeling*, September 2000.