Efficient identification of uncongested links for topological downscaling of Internet-like networks.

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

ABSTRACT

In [33, 34] two methods have been presented to *scale down* the *topology* of the Internet, while preserving important *performance* metrics. In partcular, based on the observation that only the congested links along the path of each flow introduce sizable queueing delays and dependencies among flows two methods have been proposed that can infer the performance of the larger Internet by creating and observing a suitably scaled-down replica, consisting of the congested links only. It has been demonstrated that these techniques can be used in practice to greatly simplify and expedite performance prediction.

While a main requirement for topology downscaling is that uncongested links are known in advance, the question of whether one can identify them, in an efficient and scalable way, has not been addressed yet. However, this is quite important, as it is directly related to the practicability of topological downscaling.

In this paper we provide simple rules that can be used to identify uncongested links. In particular, we first identify scenarios under which one can easily deduce whether a link is uncongested by inspecting the network topology. Then, we identify scenarios in which this is not possible and show how one can *efficiently* use known results, based on the large deviations theory, to approximate the queue length distribution. While our main motivation in this paper is to complement the work on topology downscaling, our approach is quite general and can be used beyond this context, *e.g.* for traffic engineering and capacity planning.

Categories and Subject Descriptors

C.2.5 [**Local and Wide-Area Networks**]: Internet; C.4 [**Performance of Systems**]: Measurement techniques, Modeling techniques; G.3 [**Probability and Statistics**]: Queueing theory, stochastic processes; C.2.1 [**Network Architecture and Design**]: Network topology

General Terms

Performance, Measurement, Theory

Keywords

Topology downscaling, uncongested link identification

1. 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.* [23, 6, 16, 29], measurement-based performance characterizations, *e.g.* [14, 35, 22, 37, 24], and simulation studies, *e.g.* [2, 26, 43, 21]. 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. [38, 30, 31] 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 very important problem of having to deal with large and complex network topologies, like the Internet topology.

With the above problem in mind, two methods have proposed in [33, 34] 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, it has been 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 [4, 5, 15, 13] it has been demonstrated that these techniques can be used in practice, to dramatically

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

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

simplify and expedite performance prediction.

A main requirement of this approach is that uncongested links are known in advance. However, while links that cause packet drops can be easily detected 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 [35, 36, 4]. Hence, for this kind of topological downscaling to be practical, we need ways to identify uncongested links, without having to explicitly measure their delays. This is the main contribution of this paper.

In particular, in this paper we present an efficient and scalable procedure to identify which of the links of a network topology that do not impose packet drops are uncongested, i.e. they do not impose significant delays either. Our procedure consists of rules under which one can easily identify uncongested links by inspecting the network topology, and whenever this is not possible, by efficiently using a known model from the large deviations theory (based on Fractional Brownian Motion (FBM)), to approximate the queue length distribution.

The large-deviations model we use requires knowledge of *packet-level* statistics at the link of interest. In particular, it requires knowledge of the average packet arrival rate λ, of its variance σ^2, as well as of the Hurst parameter H, an index of long-range dependence in the arrival process [17]. However, as with the queueing delays, it is difficult and not scalable to estimate these parameters by monitoring packets on every link of interest. In our approach, we make *efficient* use of this model in the sense that we choose to *infer* these parameters from *flow-level* information at the link of interest. We have chosen to do this based on the observation that it is much easier to monitor flows on a router, instead of packets [4, 5]. 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, can give us information on each link of the network [4, 5]. This alleviates the burden of having to monitor many links and makes the measuring procedure scalable. [3]

However, while simple expressions connecting λ and H to flow-level statistics exist, *e.g.* [4, 41], inferring σ^2 from flow-level information is much more involved. Another contribution of this paper is that we derive a new expression for σ^2. What distinguishes our expression from earlier ones [4, 5, 19] 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.

While our main motivation in this paper is to complement the work on topology downscaling, by efficiently identifying the uncongested links that can be ignored, our approach is quite general and can be used beyond this context, *e.g.* by network operators and managers for traffic engineering and capacity planning.

The rest of the paper is organized as follows. In Section 2 we briefly review the main concept of performance-preserving topological downscaling. In the Section 3 we identify the scenarios under which one can easily deduce whether a link imposes negligible queueing, by inspecting the topology. Whenever this is not possible, we review in Section 4 the large-deviations model that we will be using to approximate the queue length distribution. In Section 5 we explicitly identify the conditions that should hold in the context of TCP networks for this model to be valid. In Section 6 we infer the packet-level information required to use the model, from flow-level information. In Section 7 we validate the model and our theoretical arguments using simulations with TCP traffic. In the same section we also present experiments using the CENIC backbone [7], to demonstrate how the model can be used in practice to identify uncongested links, and to decide of whether a link can be ignored when performing downscaling. Comparison with earlier work follows in Section 8, and we conclude in Section 9.

[3]Tools such as NetFlow can be easily used to provide flow-level information in Cisco routers [1].

2. PERFORMANCE-PRESERVING TOPOLOGICAL DOWNSCALING

In this section we briefly review the main concept of downscaling TCP networks. For more details, the interested reader is referred to [34, 33].

Before proceeding, let's first review the definition of 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. The *majority* of backbone links have both of these properties. In particular, it is well documented that the end-to-end delay inside a backbone network is dominated by the propagation delay, and that most of the backbone links never impose packet drops [15, 13, 36, 4, 5, 35, 14]. [4] The main idea in downscaling is to reduce the topology of the network by ignoring uncongested links.

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. [5] Observe that Grp1 traverses only the one congested link, whereas Grp2 traverses both.

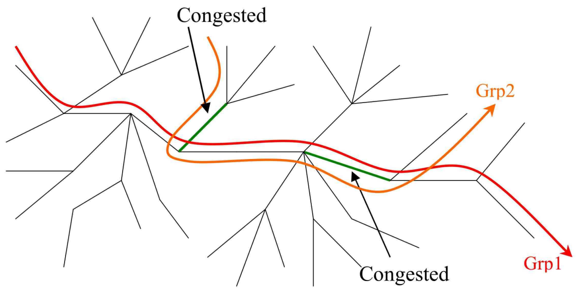


Figure 1: Original network.

In [33, 34] two methods have been proposed (DSCALEd and DSCALEs) that build scaled replicas consisting of the congested links only, along with the groups of flows that

[4]Congested links usually exist at access points and public exchange points.

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

traverse them. [6] 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.

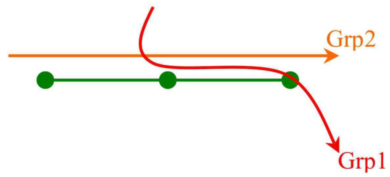


Figure 2: Scaled replica.

A main requirement of topology downscaling is that we know in advance which links of the original network are uncongested. However, as mentioned earlier, while links that cause packet drops can be easily detected by a monitoring tool, measuring queueing delays on every other link to determine whether these are negligible, is clearly a not scalable procedure, and becomes quite difficult in high-speed backbone routers [35, 36, 4]. Hence, for downscaling to be practical, we need efficient ways to identify links with negligible queueing, without having to explicitly measure their delays.

3. IDENTIFICATION OF UNCONGESTED LINKS BY TOPOLOGY INSPECTION

In this Section we identify the conditions under which one can decide whether a link is uncongested by just inspecting the network topology.

Our starting point is based on the observation that each link that belongs to the path of a group of flows of interest (*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 is 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 in these figures correspond to groups of flows, the C's are capacities, Src1...SrcN correspond to sources, and Dst1...DstN to destinations.)

Now, let's study the conditions under which the aforementioned 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. 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, for 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. But, 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, which is caused only by the first few packets of new unregulated flow arrivals, will be negligible.

Therefore, in summary, *the only case where one can decide by inspecting the network topology, 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 make *efficient* use of a model from the theory of large deviations to approximate the queue distribution. We review this model in the next section.

4. USING LARGE-DEVIATIONS THEORY TO APPROXIMATE THE QUEUE DISTRIBUTION

Consider a link/queue, and let $A(t) = A(0,t)$ denote the total traffic that has arrived at the queue (*e.g.* in units of packets or bits) in the interval $(0,t]$, with $t \in \mathbb{R}^+$ or $t \in \mathbb{Z}$. Further, let λ denote average input (arrival) rate, and C denote the queue's service rate (link capacity). To ensure stability, we assume that $\lambda < C$.

We are now interested in the steady-state probability $P(Q > \delta B)$ of the buffer content Q exceeding some prespecified level $\delta B > 0$, where $0 < \delta \leq 1$. Assuming an infinite buffer size, this probability can be expressed in terms of the arrival process $A(t)$, as follows (*e.g.* see [17]): $P(Q > \delta B) = P\left(\sup_{t\geq 0}[A(t) - Ct] > \delta B\right)$. [7]

Now, let's assume that the input process can be well described by a Fractional Brownian Motion (FBM) process. That is, let's assume that $A(t)$ is a Gaussian process with mean $E[A(t)] = \lambda t$ and variance $\mathrm{Var}[A(t)] = \sigma^2 t^{2H}$, where $H \in [0.5, 1)$. [8] Finally, let $I(H) = \frac{(C-\lambda)^{2H}(\delta B)^{2-2H}}{2\sigma^2 K^2(H)}$, where $K(H) = H^H(1-H)^{1-H}$. Then, using large-deviations theory, it can be shown that the following relationship holds for $P(Q > \delta B)$ [17, 39]:

$$P(Q > \delta B) \leq \exp\left(-I(H)\right). \tag{1}$$

The above relation is known in the literature as the *large-buffer asymptotic* upper bound and the function $I(H)$ is called the *large-deviations rate function.* If B is sufficiently large, Equation (1) is often used to approximate the queue distribution. When $\delta \geq \frac{1}{B}$ a better approximation is [25]:

$$P(Q > \delta B) \leq \frac{1}{(\delta B)^{\gamma}} \exp\left(-I(H)\right), \tag{2}$$

where $\gamma = \frac{(1-H)(2H-1)}{H}$. Hence, for better approximating the queue distribution for any δ, one can take the minimum of Equations (1) and (2).

The effectiveness of this model has been demonstrated in the context of open-loop networks, *e.g.* [11, 27, 17], and has been used several times in the context of TCP networks, *e.g.* [9, 42, 10, 44]. Next, we clearly identify the conditions under which the model is valid in this latter context. Then, we show how to use it efficiently, by inferring its parameters from flow-level information.

5. APPLICATION TO TCP NETWORKS

As we can see from the previous section there are two requirements for the model described there to be accurate: (i)

[6] The scaled replica may also include uncongested links of interest and the groups of flows that traverse them.

[7] Note that this probability is often used to approximate the corresponding probability in a system with finite buffer equal to B, when B is large [8].

[8] The constant H is the Hurst parameter. For $H = 0.5$ the process has independent increments, whereas for $H > 0.5$ the increments of the process are long-range dependent.

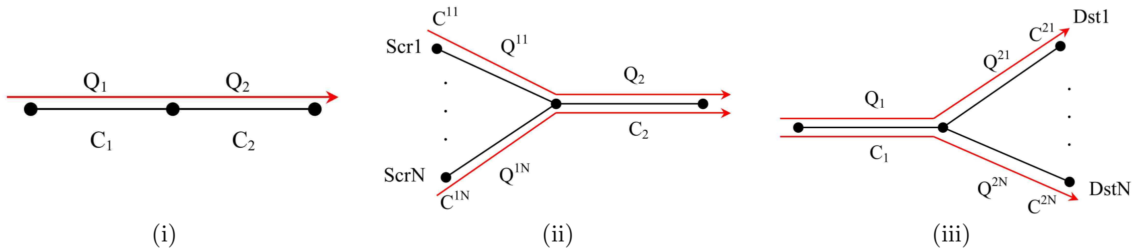


Figure 3: Toy network topologies used to illustrate when a link can be considered as uncongested by topology inspection.

the buffer size B should be large enough, and (ii) the input process should be well-described by a Gaussian process. We now briefly explain why both of these conditions hold true in the context of TCP backbone networks.

First, Internet routers today are still sized according to the rule-of-thumb, where the buffer size equals the bandwidth-delay product [18]. Since capacities in backbone links are quite large, to be able to support a large number of flows, the buffer size B is also large.

Further, while it is well known that if multiple TCP flows share a bottleneck link can get synchronized with each other [46, 12, 45], 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 [3, 15, 20].

Under the assumption of a large number of desynchronized TCP flows, the evolution of the flow window sizes becomes loosely correlated, and the distribution of their sum can be well approximated by a Gaussian distribution. This is justified by the Central Limit Theorem (CLT), it is supported by empirical measurements, and it has been argued in several recent studies [3, 10, 19].

Hence, requirements (i) and (ii) hold true in the case of Internet backbone networks. Finally, notice that the model of the previous section also accounts for long-range dependence in the increments of the aggregate Gaussian input traffic, which is another well-known characteristic of traffic in the Internet [44, 37, 41].

6. PARAMETER INFERENCE

Using the model of Section 4 requires knowledge of the *packet-level* statistics λ, σ^2, and of the parameter H. As mentioned earlier, it is difficult and not scalable to estimate these parameters by monitoring packets on every link that we want to study. As we have said, we prefer to monitor flows, which is much easier [4, 5]. Therefore, in this section we show how to infer these parameters from flow-level information. Before proceeding, recall that it is easy to detect links that impose packet drops, and thus we are interested in detecting which of the other links (that do not impose packet drops) impose significant queueing delays.

To be able to infer the *packet-level* statistics λ, σ^2 and the parameter H at the link we want to study, it is necessary (and sufficient) to have the following *flow-level* information: (i) the flow size distribution $F(s)$ of the flows traversing the link, (ii) the average flow arrival rate at the link, which we denote by r, and (iii) the average and the variance of the number of active flows on the link, which we denote by $E[A]$ and $\mathrm{Var}(A)$ respectively. [9] In practice, this flow-level information can be easily extracted from a router, *e.g.* using NetFlow [1].

6.1 Estimating λ

Let S be the random variable representing the size of a flow. Since we know $F(s)$ we can easily compute the average flow size $E[S]$. Assuming no drops at the link of interest, an intuitive and well-known expression for λ (*e.g.* see [4]) is: [10]

$$\lambda = rE[S]. \tag{3}$$

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 $\lambda = rE[S] < C$ [16]. We assume this to be the case here. (Recall that this condition is required in order to be able to invoke the model of Section 4.) Next, we use another known result to show how one can estimate the Hurst parameter H.

6.2 Estimating H

It is well-accepted that traffic in the Internet is long-range dependent, *e.g.* see [44, 37, 41]. This long-range dependence has been shown to be the result of a heavy-tailed flow size distribution [41, 15]. 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 Hurst parameter H is directly related to the parameter α (called the shape parameter) of the size distribution. According to [41]:

$$H = \frac{3 - \alpha}{2}. \tag{4}$$

Since we know the flow-size distribution $F(s)$ (and hence its shape parameter α), we can use Equation (4) to approximate

[9] 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.

[10] Of course, drops may occur on other links along the path of a group of flows that traverses the link under study. However, since we are interested in links that multiplex a large number of flows (backbone links), and given that the number of concurrent congested links in real networks is usually small, *e.g.* [34], we can make the assumption that none of the flows sharing the link under study experiences drops. Indeed, as we shall see in Section 7, when we study the CENIC backbone [7], this assumption does not affect the results.

H.

6.3 Estimating σ^2

To date, only few studies exist that relate σ^2 to flow-level information [4, 5, 19]. However, these studies either make unnecessary simplifying assumptions [19], or give fairly complicated expressions that require more information and measurements [4, 5], than we actually need.

Since we are interested in links with not drops, it turns out that we can derive new simple expression for σ^2, assuming knowledge of only the flow-level information mentioned earlier, and without making any other assumptions. (For a detailed comparison with prior work see Section 8.) The expression is given in the following Theorem:

THEOREM 1.

$$\sigma^2 = \frac{E[A]\,Var(W) + (E[W])^2\,Var(A)}{(E[RTT])^{2H}}, \tag{5}$$

where $E[W]$ is the average congestion window size of a flow that traverses the link and $Var(W)$ its variance, $E[RTT]$ is the average round-trip time of a flow, and $E[A]$, $Var(A)$ are respectively the average and variance of the number of active flows on the link.

PROOF. *Assume that time is slotted with the duration of slot i be equal to the current round-trip time. Further, for simplicity, let the current round-trip time be the same for all flows traversing the link. Now, denote by P the total number of packets that arrive to the link/queue within some time-slot. Then, $P = \sum_{j=1}^{A} W_j$, where A is the random variable representing the number of active flows 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 [40] we have:*

$$Var(P) = E[\,Var(P|A)] + Var(E[P|A]). \tag{6}$$

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:

$$E[\,Var(P|A)] = E[A]\,Var(W), \tag{7}$$

and:

$$Var(E[P|A]) = (E[W])^2\,Var(A). \tag{8}$$

Now, recall from Section 4 that $\sigma^2 t^{2H}$ is the variance of the amount of traffic that arrives at the queue in the interval $(0, t]$. As in Section 4 denote this amount of traffic by $A(t)$, and let $N(t)$ be the number of time-slots elapsed by time t. We can write $A(t) = \sum_{i=1}^{N(t)} P(i)$, where $P(i)$ is the random variable representing the number of packets arriving at the queue within 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 $Var(A(t)) = (N(t))^{2H}\,Var(P) = \sigma^2 t^{2H}$. Now, for t large enough $N(t) = \frac{t}{E[RTT]}$, and hence:

$$\sigma^2 = \frac{Var(P)}{(E[RTT])^{2H}}. \tag{9}$$

From Equations (6)...(9) we get Equation (5).

□

Note, that while in the proof of Theorem 1 we have assumed that flows have the same round-trip times, we will see in Section 7 that Equation (5) is remarkably accurate even if this is not the case.

Recall that $E[A]$ and $\mathrm{Var}(A)$ in Equation (5) are known quantities. Hence, what remains to complete the calculation of σ^2 is to compute $E[W]$, $\mathrm{Var}(W) = E[W^2] - (E[W])^2$, 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:

$$E[RTT] = \frac{E[A]}{rE[D]}. \tag{10}$$

Since $E[A]$ and r are known quantities, we only need to find $E[D]$.

Recall that S is the random variable that represents the size of a flow. Now, 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:

$$E[D \mid \text{short flow}] = E[\lfloor \log_2 S \rfloor + 1_{[S - \sum_{i=0}^{\lfloor \log_2 S \rfloor - 1} 2^i > 0]} \mid S \leq 2W_{max}], \tag{11}$$

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 $E[D]$. Thus, we can now compute $E[RTT]$ using Equation (10).

Since we know 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: [11]

$$E[W] = \frac{E[S]}{E[D]}. \tag{12}$$

What remains 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:

$$E[S^* \mid \text{short flow}] = E[\sum_{i=0}^{\lfloor \log_2 S \rfloor - 1} (2^i)^2 + (S - \sum_{i=0}^{\lfloor \log_2 S \rfloor - 1} 2^i)^2 \mid S \leq 2W_{max}], \tag{13}$$

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

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

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

LEMMA 1.

$$E[W^2] = \frac{E[S^*]}{E[D]}, \quad (15)$$

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

PROOF. *Assume again that the time is slotted with the duration of the current slot equal to the current round-trip time. Now, let Y be the sum of the squares of the window sizes of all active flows, 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:*

$$E[Y] = E[W^2]E[A]. \quad (16)$$

Let $N(t)$ be the number of time-slots elapsed by time t as before, and denote by $F(t)$ the total number of flows that have completed service within $N(t)$ slots. The average number of rounds for a flow to complete can be also expressed as $E[D] = \lim_{t\to\infty} \frac{\sum_{i=1}^{N(t)} A(i)}{F(t)}$, where $A(i)$ is the number of active flows in slot i. Also, the average number of active flows in a slot can be written as $E[A] = \lim_{t\to\infty} \frac{\sum_{i=1}^{N(t)} A(i)}{N(t)}$. From the last two equations we get that $\lim_{t\to\infty} \frac{N(t)}{F(t)} = \frac{E[D]}{E[A]}$. Further, it is easy to see that $E[S^*] = \lim_{t\to\infty} \frac{\sum_{i=1}^{N(t)} \sum_{j=1}^{A(i)} (W_j^i)^2}{F(t)}$, where W_j^i is the congestion window size of flow j ($j \in 1...A(i)$). Since $E[Y]$ can be also written as $E[Y] = \lim_{t\to\infty} \frac{\sum_{i=1}^{N(t)} \sum_{j=1}^{A(i)} (W_j^i)^2}{N(t)}$ we can conclude (from the last three relations) that:

$$E[Y] = \frac{E[S^*]}{E[D]} E[A]. \quad (17)$$

Combining Equations (17) and (16) we get the result. □

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

7. EXPERIMENTS

In this section we use the ns-2 simulator [28] to validate our theoretical arguments and to demonstrate the procedure for efficiently identifying uncongested links when performing topology downscaling. In particular, we present two sets of experiments. In the first set we consider a single link shared by TCP flows, in order to verify the accuracy of the model of Section 4 and of our parameter estimation (Section 6), as well as to give insights on the queueing behavior of Internet links that are shared by a large number of flows. In the second set, we use the topology of the CENIC backbone [7], to demonstrate the procedure for identifying uncogested links when performing topology downscaling on real networks.

7.1 Single Link Experiments

We consider a single link/queue like the one shown in Figure 4, having capacity NC, propagation delay $Tprop$, and buffer size $B = 2NCTprop$ (i.e. equal to the bandwidth-delay product). TCP flows arrive at the link at random times, according to a Poisson process, with rate $Nr = N95$flows/sec. [12] The number of data packets S in each flow follows a bounded Pareto distribution with average $E[S] = 11.5$packets, maximum 10^6packets, and shape parameter $\alpha = 1.34$. The size of an IP data packet is 1040 bytes, $T_{prop} = 50$ms, and $C = 10$Mbps. Finally, $W_{max} = 20$packets and the simulation time is 10000sec. We study the queueing dynamics of the link as N increases, i.e. as if this was a backbone link. (Notice that the offered load is $\rho = \frac{NrE[S]}{NC} = \frac{rE[S]}{C} = 0.91 < 1$, and does *not* change as we vary N.)

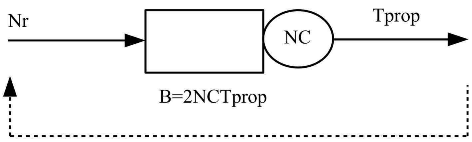


Figure 4: Single link topology.

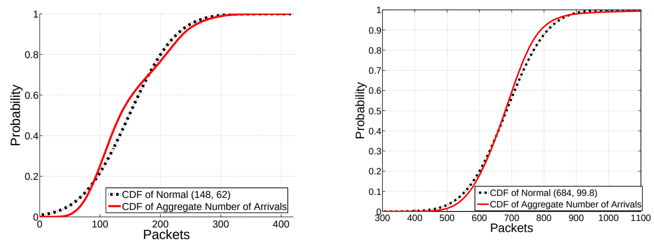
We start by verifying that the aggregate packet arrival process at the link can be approximated by a Gaussian distribution. Figures 5(i) and 5(ii) show that this is indeed the case, even for N's as small as 1 and 6 respectively. Note that for $N = 1$ the average number of active flows is approximately $E[A] = 40$, and the packet drop ratio is 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 [3]. [13] For $N = 6$, the average number of active flows is $E[A] = 162$, 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 5(ii). Also, notice that the drop ratio is smaller than the case where $N = 1$. This is in agreement with the model of Section 4, which implies that for any level $\delta > 0$, as N, and hence B, increases, the probability that the buffer content exceeds δB decreases.

We now test the accuracy of the model of Section 4 and of the expressions derived in Section 6. Recall, that for the purposes of downscaling we are interested in identifying which of the links that do not impose packet drops are uncongested, i.e. impose negligible queueing delays. As we have observed from the simulator, drops stop occurring for $N > 10$. Thus, we show results for $N = 11, 16,$ and 32.

We estimate λ, σ^2 and H, using the formulas of Section 6. Recall, that in order to compute σ^2 we also need estimates for $E[A]$ and Var(A). These are extracted from the

[12] Of course, while flow arrivals are Poisson, packet arrivals are dictated by the TCP dynamics. Further, similar results hold for any other flow arrival process.

[13] The study in [3] verified the Gaussian approximation assumption for the case of a single link that is shared by long-lived persistent TCP flows having unbounded window sizes. Here, we verify this for the more realistic case where TCP flows arrive at random times, have random sizes and bounded windows.



(i) (ii)

Figure 5: 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.

simulator. We compute the rest of the required parameters, and their values are: $E[D] = 2.65$rounds (which gives $E[W] = 4.3$packets), and $E[S^*] = 127.5$packets (which gives $E[W^2] = 48$packets). ($E[RTT]$ is computed by Equation (10) given the corresponding value for $E[A]$ and the flow arrival rate, which is $95N$flows/sec.)

Table 1 gives the values for $\lambda, E[A], \mathrm{Var}(A)$ and the resulting σ^2, as we vary N. In all cases $H = 0.83$ (as the shape parameter of the flow-size distribution remains the same).

N	λ(pkts/sec)	$E[A]$	Var(A)	σ^2 (pkts/sec)
11	12018	281	578	858148
16	17480	404	644	1093018
32	34960	807	929	1864839

Table 1: Flow- and packet- level statistics at the link.

Figure 6 shows that the model is quite accurate for approximating the queue length distribution, especially for large N, as expected, and also verifies that our parameter estimation is correct. (The latter has been also verified by comparing the derived theoretical values with the corresponding simulation values.) The plots also validate the argument that in backbone links, where N is sufficiently large, queueing delays can be ignored, and the model of Section 4 gives the theoretical justification. Indeed, for $N = 32$ the average queueing delay is approximately $\overline{T} = 1$ms, which is two orders of magnitude smaller than the two-way end-to-end propagation delay of a packet (which is 100ms). Interestingly enough, this is case even for links working at above 90% utilization, like the one in this example.

This last observation motivates us to study the amount of multiplexing (value for N) required at different offered loads ρ, such that for the majority of time, the queuing delay T remains below a sufficiently small fraction of the end-to-end propagation delay. This is important for topology downscaling, where we can ignore links with negligible queueing delay (compared to the end-to-end delay).

Figure 7 shows the value for N such that the queueing delay is one order of magnitude smaller than the end-to-end propagation delay for at least 90% of the time, for different values of ρ. From the figure we observe that at small offered loads, a small value for N is sufficient. In particular, for

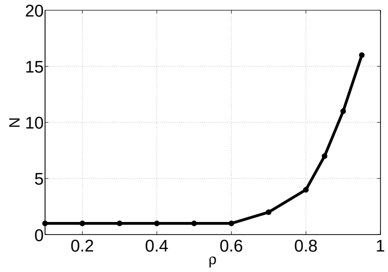
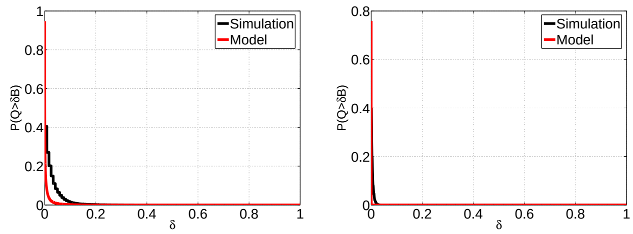


Figure 7: The value for N such that $P(T < 0.1 \times 2Tprop) > 0.9$, as a function of the offered load ρ.

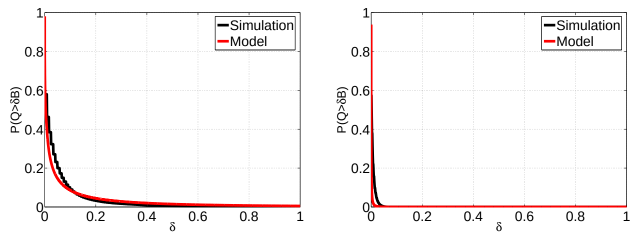
$\rho \leq 60\%$ $N = 1$ is sufficient, whereas as ρ increases, N also increases as expected, with the increase being faster than exponential as $\rho \to 1$. For example, for $\rho = 90\%$, we need $N = 15$. Even in this case however, this corresponds to a flow arrival rate of $95N = 1425$flows/sec, a capacity of $10N = 150$Mbps, and a buffer size of $120N = 1920$packets, all of which are not unrealistic for backbone links, *e.g.* see [3].

Figures 8...12 show how accurately the model can predict the queue distribution at other offered loads, for various values of N where there are no drops. First, from the plots we



(i) (ii)

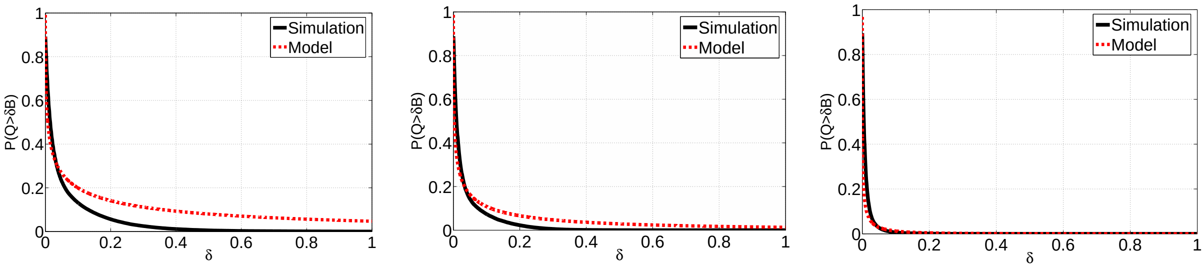
Figure 8: Queue exceedance probability $P(Q > \delta B)$ against the buffer level δ for $\rho = 0.4$: (i) $N = 1$, and (ii) $N = 6$.



(i) (ii)

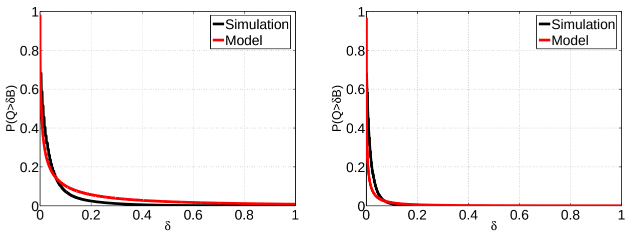
Figure 9: Queue exceedance probability $P(Q > \delta B)$ against the buffer level δ for $\rho = 0.6$: (i) $N = 1$, and (ii) $N = 6$.

observe that the approximation is not accurate when $\delta \to 0$. This is expected, since as we have said in Section 4 we require that $\delta > 0$. Further, we observe that the model is quite applicable for all link utilizations above 70%. At smaller



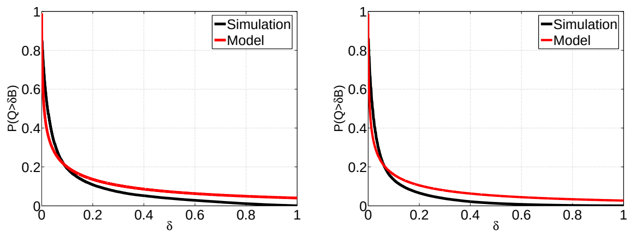
(i) (ii) (iii)

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



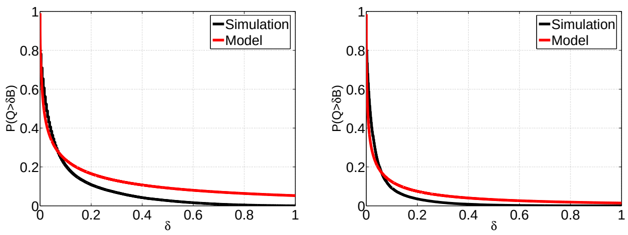
(i) (ii)

Figure 10: Queue exceedance probability $P(Q > \delta B)$ against the buffer level δ for $\rho = 0.7$: (i) $N = 2$, and (ii) $N = 4$.



(i) (ii)

Figure 12: Queue exceedance probability $P(Q > \delta B)$ against the buffer level δ for $\rho = 0.85$: (i) $N = 4$, and (ii) $N = 6$.



(i) (ii)

Figure 11: Queue exceedance probability $P(Q > \delta B)$ against the buffer level δ for $\rho = 0.8$: (i) $N = 2$, and (ii) $N = 4$.

utilizations we see that it underestimates the true queue occupancy. This agrees with the experimental observations in [36]. However, for the purposes of downscaling such discrepancies do not affect our decisions of whether to keep or ignore a link, since we are making *order-of-magnidute* comparisons between queueing delays and end-to-end delays. In addition, one can also argue that backbone links at utilizations below 50% impose insignificant queueing, and it is always safe to consider them as uncongested [13, 36], without the need of using the model to approximate their queue distribution.

7.2 Cenic Backbone Experiments

We now consider the topology of the CENIC backbone [7], which is shown along with link information in Figure 13. Note that the CENIC maps do not include information about the propagation delays of the links and the paths of the packets that traverse them. We estimate the propagation delay of a link by dividing the length of the link over the propagation velocity of the signal (taken as 133000 miles/sec). The propagation delay for all the links that belong to the same geographic area is taken as 0.1ms and for the rest of the links is shown in Figure 13 (appended next to each link). Further, the buffer size of each link equals the bandwidth-delay product, where the delay factor is taken equal to the maximum end-to-end propagation delay of a flow, which is 10ms.

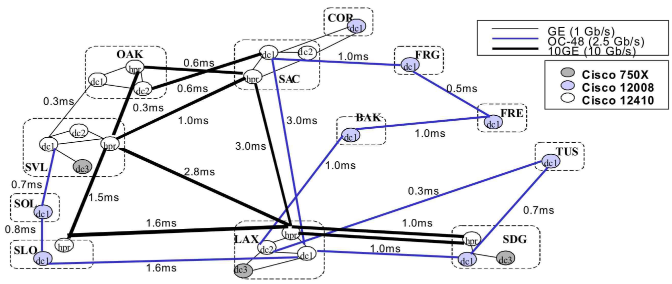


Figure 13: The CENIC Backbone.

We let each possible source-destination pair in the topology to correspond to a group of flows. (Notice that links are bidirectional.) Hence, in total there are 600 groups of flows. The flow arrival rate for all groups of flows is 100flows/sec, except from the group that enters *SVL(dc1)* and exits *SVL(hpr)*, whose rate is 5200flows/sec, and from the group that enters *SVL(hpr)* and exits *LAX(hpr)*, whose rate is 90000flows/sec.

Link *SVL(dc1)-SVL(hpr)* imposes packet drops, and hence is congested. No other link in the topology imposes packet drops. We are interested in studying the performance of the congested link and of the groups of flows that traverse it, which we call groups of interest. According to [33, 34] one can build a scaled replica consisting of this link along with the groups of interest and all other congested links in the topology that these groups traverse. Since we know that no other link imposes drops, our task is to identify if there are links traversed by groups of interest that have significant delays, and if so, include them in the scaled replica. Before proceeding, we summarize the general procedure that we follow.

Procedure for identification of uncongested links: (i) 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 2). (ii) For all other links we use a flow-level measurement tool, *e.g.* such as NetFlow [1], to estimate: (a) The flow-size distribution, (b) the flow arrival rate, and (c) the average, and the variance of the number of active flows. (iii) For each of these links, we use Equations (3)...(5) to compute λ, H, and σ^2. (iv) We use the model of Section 4 (Equations (1) and (2)) to approximate the queue distribution on each of these links. (v) 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, and (vi) As in [33, 34] we ignore all those links for which their average queueing delay is one order of magnitude smaller than the corresponding two-way end-to-end propagation delay.

Note, that a heuristic rule-of-thumb to expedite the above procedure is to measure, after step (i), the offered load ($\rho = \frac{rE[S]}{C}$) on all remaining backbone links and directly ignore all links for which this load is quite low, *e.g.* $\rho \leq 50\%$. As mentioned before, this is based on the observation that such links always impose negligible queueing [13, 36], and hence can be ignored.

In our simulation setup, the offered load on link *SVL(hpr)-LAX(hpr)*, which is traversed by a total of 102 groups of flows out of which 37 are groups of interest, is approximately 95%, whereas on all other links that are traversed by groups of interest is below 40%. Following our procedure, and using the aforementioned rule-of-thumb, the only link that we need to approximate the queue length distribution to decide whether it is congested, is link *SVL(hpr)-LAX(hpr)*. The average flow arrival rate at this link is $r = 95150$flows/sec, the flow characteristics are the same as before (except that $E[S] = 12$packets), the average number of active flows on the link is $E[A] = 1482$ and its variance is $\text{Var}(A) = 2464$. As before, we can compute $\lambda = 1141800$packets/sec, $\sigma^2 = 629752212$packets/sec, and $H = 0.83$. Figure 14 shows how accurately we can approximate the queue length distribution.

Our approximation yields an average queueing delay of $\overline{T} = 0.26$ms and the actual is $\overline{T} = 0.17$ms. In both cases this is one order of magnitude smaller than the average end-to-end propagation delay of flows that traverse the link under study, which is 6.33ms. Therefore we ignore this link.

To validate the whole procedure we use DSCALEd (which

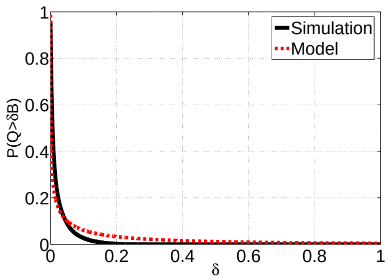


Figure 14: Queue exceedance probability $P(Q > \delta B)$ against the buffer level δ for link *SVL(hpr)-LAX(hpr)*.

accounts for the missing uncongested links by imposing appropriate delays at the sources of the packets) [33, 34], to build a scaled replica consisting of the congested link *SVL(dc1)-SVL(hpr)* only. In Figure 15 we present some of the most important performance metrics that we can predict using the scaled replica, and we compare them to that of the original system (Figure 13). In particular, we show the distribution of the number of active flows on link *SVL(dc1)-SVL(hpr)*, and the end-to-end flow delay histograms of two (out of the 71) groups of interest. (*grp2* also traverses link *SVL(hpr)-LAX(hpr)* that we previously decided to ignore.)

It is visually evident from the plots that performance prediction is quite accurate. (Similar results hold for all the metrics [34] that the replica can predict.) Further, in addition, if we use the same statistical measure to quantify differences between two distributions as in [34], i.e. the Histogram Similarity Measure (HSM), we find that the average HSM for these plots is 0.85, which is quite high given the reduction in the complexity of the network. (HSM=1, means that two distributions are identical). Therefore, the proposed procedure can be efficiently applied to identify and ignore uncongested links.

As another example, we further increase the offered load on link *SVL(hpr)-LAX(hpr)* to 98%. Figure 16 shows theoretical and simulation results for the queue exceedance probability as before. Our approximation yields an average queueing delay of $\overline{T} = 1.86$ms and the actuall is $\overline{T} = 1.42$ms. In both cases, this is comparable to the average end-to-end propagation delay, i.e. the one order of magnitude requirement is not satisfied, and hence we *do not* ignore this link.

Figure 17 demonstrates that performance prediction is indeed less accurate if we ignore the link, especially for the groups that were traversing this link in the original system, *e.g.* such as *grp2*, since these groups do not experience the same delays in the scaled replica. Also, inaccuracies are notable in the distribution of active flows. However, they are less notable for groups that traverse the congested link SVL(dc1)-SVL(hpr) and not the link *SVL(hpr)-LAX(hpr)*. This is because these groups experience similar delays in the two systems. The average HSM for these plots is 0.69, which is quite smaller than before.

8. RELATED WORK

We now review related work on the applicability of the model of Section 4, and on estimating σ^2. For related work

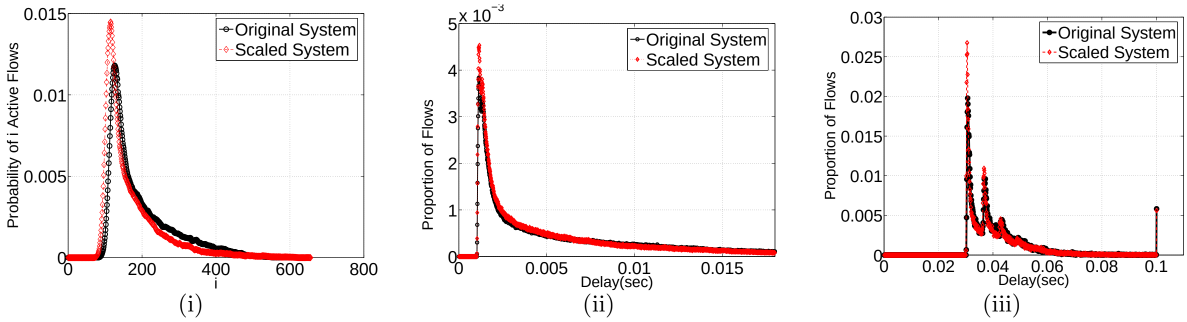


Figure 15: (i) Distribution of active flows on the congested link *SVL(dc1)-SVL(hpr)*, (ii) *grp1* end-to-end flow delay histogram, and (iii) *grp2* end-to-end flow delay histogram. (Scenario 1.)

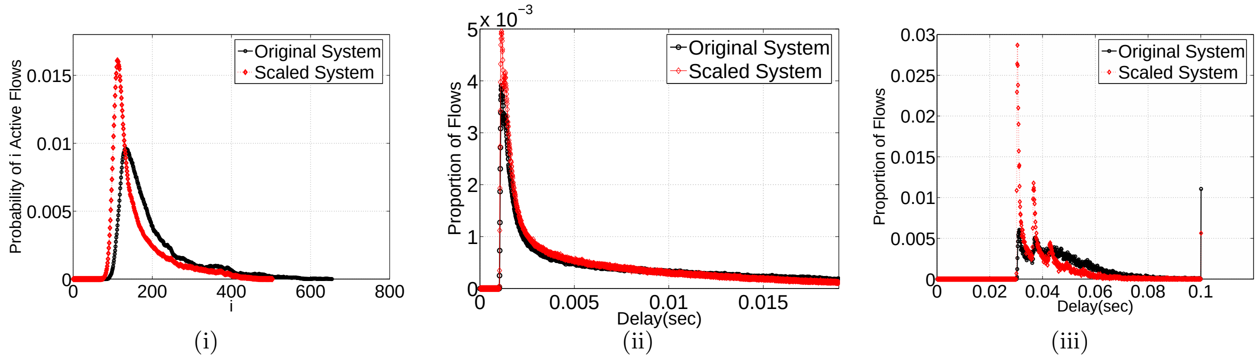


Figure 17: (i) Distribution of active flows on the congested link *SVL(dc1)-SVL(hpr)*, (ii) *grp1* end-to-end flow delay histogram, and (iii) *grp2* end-to-end flow delay histogram. (Scenario 2)

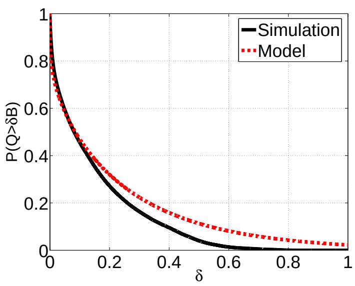


Figure 16: Queue exceedance probability $P(Q > \delta B)$ against the buffer level δ for link *SVL(hpr)-LAX(hpr)*.

on network downscaling see [34].

As mentioned earlier, the model presented in Section 4 has been derived in several studies and its effectiveness has been verified in the context of open-loop networks, *e.g.* see [11, 27, 17]. Its applicability has been also demonstrated for Internet backbone traffic, *e.g.* see [44]. And, it has been used in this later context by authors for their theoretical arguments, *e.g.* in [42, 10].

In this study we have shown that this model can be also effectively applied in the context of topology 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 further validate it.

In contrast to earlier studies that have utilized the model by extracting its parameters from packet-level traces, *e.g.* [44, 36], 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 are the studies in [4, 5, 19]. 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 [19] the author has also assumed a bufferless link model and modeled the number of active flows as an $M/G/\infty$ queue (which is only accurate when queueing delays are equal to zero). During our formula derivation, none of these simplifying assumptions have been made. Further, in [4, 5] 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. Also, in [19] it is assumed that the 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 [4, 5], which is the most relevant, 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 active flows on a router, which can be easily measured, independently from the flow sizes.

9. CONCLUSION AND FUTURE WORK

This paper complements recent work on topology downscaling of Internet-like networks [33, 34]. In particular, this paper proposes a procedure to identify links with negligible queueing delays that can be ignored when building scaled-down replicas.

Further, this study goes beyond the context of network downscaling. It demonstrates how a well-known model from the large-deviations theory can be efficiently utilized in practice, and it presents a new simple formula that relates the variance of the packet arrival process to flow-level statistics.

Future work consists of further validating the proposed procedure using other network topologies, and analytically quantifying the relationship between the number of uncongested links that are ignored by topology downscaling and the achieved accuracy in performance prediction. Relevant to this, another interesting direction is to *theoretically* establish the queueing delay threshold below which, the queueing dynamics of a link can be completely ignored when evaluating a network's performance.

10. REFERENCES

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

[2] J. S. Ahn and P. B. Danzig. Speedup and accuracy versus timing granularity. *IEEE/ACM Transactions On Networking*, 4(5):743–757, October 1996.

[3] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing router buffers. In *Proc of ACM SIGCOMM*, August 2004.

[4] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, and P. Owezarski. A flow-based model for internet backbone traffic. In *Proc. of the 2nd ACM SIGCOMM Workshop on Internet measurment*, 2002.

[5] 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*, 51(8), August 2003.

[6] 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.

[7] Intermapper web server. `https://intermapper.engineering.cenic.org`.

[8] A. B. Dieker and M. Mandjes. Fast simulation of overflow probabilities in a queue with gaussian input. *ACM Trans. Model. Comput. Simul.*, 16(2):119–151, 2006.

[9] A. Erramilli, O. Narayan, and W. Willinger. Experimental queueing analysis with long-range dependent packet traffic. *IEEE/ACM Trans. Netw.*, 4(2):209–223, 1996.

[10] 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.

[11] Z. Fan and P. Mars. Accurate approximation of cell loss probability for self-similar traffic in ATM networks. *Electronic letters*, 32(19):1749–1751, September, 1996.

[12] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, 1993.

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

[14] 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*, 17(6), November 2003.

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

[16] S. B. 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.

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

[18] Y. Ganjali and N. McKeown. Update on buffer sizing in internet routers. *ACM SIGCOMM Computer Communication Review*, 36(5):67–70, October 2006.

[19] D. Heyman. Sizing backbone Internet links. *Operations Research*, 53(4), 2005.

[20] G. Iannaccone, M. May, and C. Diot. Aggregate traffic performance with active queue management and drop from tail. *SIGCOMM Comput. Commun. Rev.*, 31(3):4–13, 2001.

[21] B. Liu, D. 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.

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

[23] 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.

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

[25] O. Narayan. Exact asymptotic queue length distribution for fractional brownian traffic. *Advances in Performance Analysis*, 1(1), 1998.

[26] D. Nicol and P. Heidelberger. Parallel execution for serial simulators. *ACM Transactions On Modeling and Computer Simulation*, 6(3):210–242, July 1996.

[27] I. Norros. On the use of fractional Brownian motion in the theory of connectionless networks. *IEEE Journal on selected areas in communications*, 13(6), 1995.

[28] Network simulator. `http://www.isi.edu/nsnam/ns`.

[29] 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.

[30] 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.

[31] R. Pan, B. Prabhakar, K. Psounis, and D. Wischik. SHRiNK: Enabling scaleable performance prediction and efficient simulation of networks. *IEEE/ACM Transactions on Networking*, 13(5):975–988, October 2005.

[32] F. Papadopoulos and K. Psounis. Efficient identification of uncongested links for topological downscaling of Internet-like networks. *Technical report, University of Southern California*, 2007.

[33] F. Papadopoulos, K. Psounis, and R. Govindan. Performance preserving network downscaling. In *Proc. of the 38th Annual Simulation Symposium*, 2005.

[34] F. Papadopoulos, K. Psounis, and R. Govindan. Performance preserving topological downscaling of internet-like networks. *IEEE Journal on Selected Areas in Communications, Issue on Sampling the Internet: Techniques and Applications*, 24(12), 2006.

[35] 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.

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

[37] V. Paxson and S. Floyd. Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, June 1995.

[38] 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.

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

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

[41] 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*, 5(1):71–86, February 1997.

[42] D. Wischik. Buffer requirements for high-speed routers. In *Proc of ECOC*, 2005.

[43] A. Yan and W. B. Gong. Time-driven fluid simulation for high-speed networks. *IEEE Transactions On Information Theory*, 45(5):1588–1599, July 1999.

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

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

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