Network Decomposition: Theory and Practice *

Do Young Eun and Ness B. Shroff†
Purdue University
West Lafayette, IN 47907-1285, U.S.A.
{eun, shroff}@ecn.purdue.edu

Abstract

We show that significant simplicities can be obtained for the analysis of a network when link capacities are large enough to carry many flows. We develop a network decomposition approach in which network analysis can be greatly simplified. We prove that the queue-length at the downstream queue converges to that of a single queue obtained by removing the upstream queue, as the capacity and the number of flows at the upstream queue increase. The precise modes of convergence vary depending on the type of input traffic, i.e., from regulated traffic arrivals to point process inputs. Our results thus help simplify network analysis by decomposing the original network into a simplified network in which all the nodes with large capacity have been eliminated. By means of extensive numerical investigation under various network scenarios, we demonstrate different aspects and implications of our network decomposition approach. Some of our findings are that our techniques perform well especially for the cases when (i) many flows are multiplexed as they enter the queue and/or (ii) departing flows are routed to different downstream nodes, i.e., no single flow dominates at any node.

Index Terms: Network decomposition, many-sources-asymptotic, performance analysis, aggregation, overflow probability.

1 Introduction

The internet has undergone a tremendous increase both in network capacity and in the number of end-users, and this trend is expected to continue for the next several years. Further, these end-users are becoming increasingly sophisticated and demand high-bandwidth, low-delay network services at affordable prices. The conflicting requirements of maintaining a high level of network utilization (for affordable prices or high revenue), while at the same time keeping network congestion under check

*An earlier version of this paper appeared in the Proceedings of IEEE INFOCOM 2003 [1].

†This work has been partially supported by the National Science Foundation through the Special Projects Award ANI-0099137 and the Indiana 21st Century Research and Technology Award 1220000634.

(for ensuring a good level of quality of service), make it imperative to understand at a fundamental level how to design and control the next generation of networks.

The issues outlined above appear to be daunting within the confines of traditional stochastic and queueing techniques. However, in this paper, we will show that the fact that a large number of traffic flows will be supported on the network can actually be exploited to help obtain results for predicting performance and allocating network resources.

The analysis of queueing networks is a well-known difficult problem except in special cases, such as Markovian queueing networks, for which product-form solutions are available [2, 3]. The difficulty in the analysis is primarily because the traffic arrival processes lose their original statistical characteristics as they traverse through the network due to the interaction with other flows and the competition for limited resources. Given a myriad of sophisticated techniques developed for analyzing a single queue, there has been some recent work in the literature to introduce the notion of decomposition or decoupling in a network to make it possible to analyze a network using those techniques [4, 5, 6]. de Veciana *et al.* [4] showed that under certain constraints, the effective bandwidth of a traffic flow is not altered and devised a notion of a decoupling bandwidth by which the queueing network can be decoupled or decomposed in an appropriate way. More recently, under many sources assumptions, Wischik [5, 6] showed that the arrival traffic and the departure traffic after being averaged over its *i.i.d.* copies (or normalized by the number of sources) satisfies the same large deviations principle, as the number of traffic sources increases. However, due to the large deviations framework used in these works, the queue dynamics in the network are described only in a log-asymptotic sense via the rate function of each flow.

In this paper we set out to see if we can *decompose* a queueing network into a single queue. A typical scenario is to analyze a queue, which is located deep inside a network or at egress routers so that the traffic flows feeding that queue are essentially unreachable and statistically different from their exogenous counterparts (original traffic arrivals to the network). Our decomposition approach, from a queueing analysis perspective, will allow us to virtually "ignore" the preceding nodes that are capable of serving a large number of traffic flows. This property enables us to develop a simple decomposition method to simplify the analysis of a network with multiple queues.

This paper is largely comprised of two parts: a collection of theoretical results supporting the notion of network decomposition for various traffic models, and extensive simulation results supporting the utility of our decomposition approach under different network configurations.

For our theoretical analysis, we consider a two-stage queueing network where the upstream queue serves many flows. First, we consider the case when the input traffic to the network is regulated (e.g., with a leaky-bucket, etc.), i.e., there exists a maximal arrival pattern in the worst case. Under this assumption, we prove that, as the number of flows (or the capacity) at the upstream queue increases, the queue-length at the downstream queue converges almost surely to the queue-length of a single queue that is obtained by removing the upstream queue from (or decomposing) the original system. We then provide results for general (including non-regulated) "fluid-like" traffic models, which state that the overflow probability at the downstream queue converges uniformly to that of a single queue by removing the upstream queue. In both cases, the convergence of the overflow probability of the original downstream queue to that of the decomposed system, happens uniformly and at least exponentially fast. Our last set of theoretical results deal with stationary point process inputs. As will be discussed later, the point process models result in a different behavior of the upstream queue compared to the fluid traffic models. Nevertheless, it turns out that we still have the convergence of the overflow probability (with one packet 'offset').

Our analytical results indicate that, if internal nodes in a network are capable of serving many flows, we can remove these nodes from consideration and the queueing behavior of other network nodes remains largely the same. In this way, we can, in many cases, simplify the analysis of a queueing network into that of a single queue, for which many analytical techniques are available in the literature. Next, to verify our decomposition approach, we provide a number of numerical simulations in which we apply our method for different network configurations using different traffic models as well as actual input traces. Our numerical results confirm that removing nodes with large capacity will not change the performance of the network, e.g., the overflow probability at the downstream node or the end-to-end delay distribution for a particular traffic flow.

The rest of the paper is structured as follows. In Section 2, we present some background material and formulate our problem tailored to the theoretical results in Section 3. In Section 3, we provide theoretical results on the two-stage queueing network under various input models ranging from regulated traffic arrivals to 'fluid' arrivals (e.g., on-off sources with heavy-tailed on-time distributions) and to stationary point process inputs. Section 4 contains numerical results to verify the theory under more realistic and complicated configurations including multiple-stage queueing networks with many interfering (cross) traffic flows. In Section 5 we discuss issues concerning several ways of scaling the system with large capacity. We then conclude in Section 6.

2 Preliminary Results and Problem Description

In this section we first summarize the many-sources-asymptotic results from the literature, particularly an upper bound, which will serve as a starting point toward the development of our main results later in this paper. We then present our model and formulate the problem in detail.

2.1 Many-sources-asymptotic upper bound

Consider the queue shown in Figure 1. In this figure, for flow i, $A_i(s,t)$ and $D_i^N(s,t)$ represent the amount of traffic arriving to and departing from the queue, respectively, during a time interval $[s,t)$. Here, N is the number of flows in the queue and also used as a scale parameter. The server capacity of the queue is NC and $q^N(t)$ denotes the queue-length at time t.

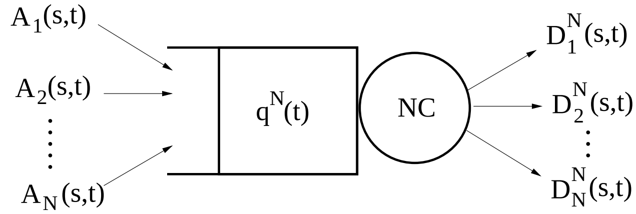


Figure 1: Example of a queue fed by many sources

Throughout the paper, for simplicity of exposition, we assume that $A_i(s,t)$, $i = 1, 2, \ldots, N$, are *i.i.d.* with stationary increments and the time indices s and t are taken from the discrete time domain, i.e., $s, t \in \mathbb{Z}$, unless otherwise noted. We consider a continuous time model ($T = \mathbb{R}$) in Section 3.3 and Section 3.6, and non-*i.i.d.* arrivals in Section 3.4. For stability, we require that $\mathbb{E}\{A_i(-t,0)\}/t := \lambda < C$ where $t > 0$. Then, the steady-state queue-length at time t can be expressed as [7]

$$q^N(t) := \sup_{s \le t} \ \big[\sum_{i=1}^{N} A_i(s,t) - CN(t-s)\big]. \tag{1}$$

Note that from the stationary increments property of $A_i(s,t)$, the distribution of $q^N(t)$ does not depend on t. We define

$$J_t(b) := \sup_{\theta} \ \big[\theta(Ct+b) - \log \mathbb{E}\{e^{\theta A_1(0,t)}\}\big]. \tag{2}$$

In this context, we recall the following many-sources-asymptotic upper bound from the work in [8], based on large deviations theory [9, 10, 11].

Proposition 1 (many-sources-asymptotic upper bound) *Suppose that*

$$\liminf_{t \to \infty} J_t(0)/\log t > 0. \tag{3}$$

Then, we have, for any t,

$$\limsup_{N \to \infty} \frac{1}{N} \log \mathbb{P}\left\{q^N(t) > Nb\right\} \le -I(b), \quad b \ge 0, \tag{4}$$

where $I(b)$ is given by

$$I(b) := \inf_{t>0} \sup_{\theta} \ \big[\theta(Ct+b) - \log \mathbb{E}\{e^{\theta A_1(0,t)}\}\big]. \tag{5}$$

□

In [8], the assumption (3) is shown to be more general than the one used in [12] in that (3) holds even for on-off sources with heavy-tailed on-time distributions. For the classical treatment of the many-sources-asymptotic results, we refer to the papers [12, 13, 14].

2.2 Problem description and model assumptions

We consider a two-stage queueing system shown in Figure 2. In this figure, the upstream queue (with queue-length $q^N(t)$) represents a node that is capable of serving a large number of traffic flows in a network, while each downstream queue at the second stage could be a node with only a small capacity (for example, an output port of an edge router).

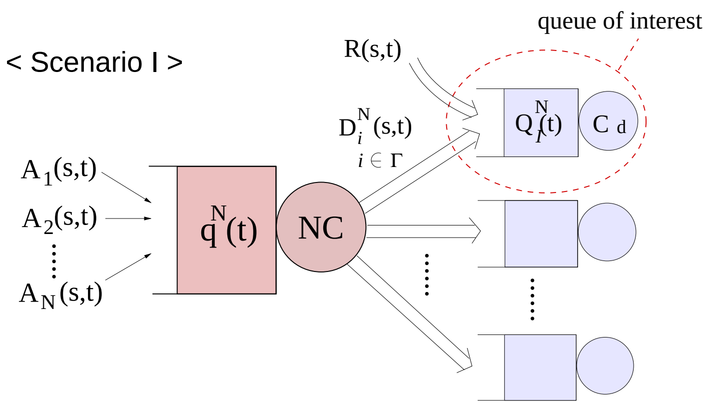


Figure 2: Queueing network: Scenario I

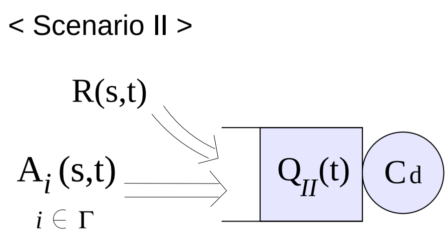


Figure 3: Scenario II; a simplified version of Scenario I

After being served at the upstream queue, the flows are routed to many different nodes, each of which usually serves only a fraction of flows. In particular, among the N flows, a *fixed subset** (not dependent on N) of the flows i ($i \in \Gamma$) after being served at the first (upstream) queue arrives to one of the downstream queues (with queue-length $Q_I^N(t)$ and capacity C_d) with an arbitrary interfering traffic flow $R(s,t)$, while the rest of flows are routed to other nodes. We can thus write the steady-state queue-length at that downstream node as

$$Q_I^N(0) := \sup_{t \geq 0} \left[\sum_{i \in \Gamma} D_i^N(-t,0) + R(-t,0) - C_d t \right].$$

We are then interested in estimating the steady-state overflow probability $\mathbb{P}\{Q_I^N(0) > x\}$ for a given buffer level x. In order to do that, we consider a simple single-stage queueing system shown

*In general, Γ does not have to be a fixed subset for the results to hold for "fluid" arrivals. However, the results are more meaningful in the case when Γ constitutes a fixed set of flows (see Remark 1 in Section 3.1). For point process models in Section 3.6, we strictly require that Γ be fixed.

in Figure 3, a simplified version of the original two-stage queueing system in Figure 2. In Scenario II, the queue has the same interfering traffic $R(s,t)$ and the same service capacity C_d as that of Scenario I, except that the traffic arrival of interest to the queue is now $A_i(s,t)$ instead of $D_i^N(s,t)$. Specifically, we write the steady-state queue-length in Scenario II as

$$Q_{II}(0) := \sup_{t\geq 0} \Big[\sum_{i\in\Gamma} A_i(-t,0) + R(-t,0) - C_d t\Big].$$

Thus, we obtain Scenario II if we remove the upstream queue in Scenario I (the queue with large capacity or bandwidth). Note that $Q_{II}(0)$ does not depend on N, while $Q_I^N(0)$ does. In the next section, we will relate aspects of the queue-length ($Q_{II}(0)$) in the simplified system to that of the downstream queue in the original system ($Q_I^N(0)$), when N is large.

3 Network Decomposition: Theory

In this section we show how $Q_I^N(0)$ converges to $Q_{II}(0)$ for various cases as the number of flows (N) increases. The precise mode of the convergence for different inputs will be revealed in each subsection. For the sake of smooth generalization, the subsections are ordered in an increasing order of complexity, i.e., from the simplest case where traffic arrivals are regulated in the discrete time domain, to the most general case where general (non-regulated) traffic arrivals or point-process arrivals are considered in the continuous time domain. We only provide complete proofs for regulated traffic arrivals in the discrete time domain. Proofs for other cases are beyond the scope of this paper and can be found in [15, 16].

3.1 Regulated traffic

In the first scenario, we assume that the input traffic arrivals $A_i(s,t)$ and the interfering traffic $R(s,t)$ are *regulated*; i.e., there exists a function $A^*(t)$ and $R^*(t)$ such that $A_i(s,s+t) \leq A^*(t)$ and $R(s,s+t) \leq R^*(t)$ for all $s,t > 0$. In this case, we can assume that, without loss of generality, the functions $A^*(t)$ and $R^*(t)$ are non-decreasing and subadditive [17]. Thus, $A^*(t)/t$ converges to its minimum value, and so does $R^*(t)/t$ (see Lemma 6.1.11 in [9]). Let a^* be this minimum value, i.e., we define

$$a^* := \lim_{t\to\infty} \frac{A^*(t)}{t} = \inf_{t>0} \frac{A^*(t)}{t}. \tag{6}$$

We also define

$$r^* := \lim_{t\to\infty} \frac{R^*(t)}{t} = \inf_{t>0} \frac{R^*(t)}{t}.$$

We will need the following assumption:

(A1): Let a^* and r^* be defined as above, and $|\Gamma|$ be the cardinality of the set Γ. Then, $a^* < C$ and $a^*|\Gamma| + r^* < C_d$.

The inequality conditions in (A1) were posed in order to exclude trivial cases. To see this, if $a^* \geq C$ and the traffic arrives according to the worst case $A^*(t)$, then we have from (6) that $A_i(-t, 0) \geq Ct$ for all t, and thus $q^N(0)$ in (1) grows without bound for any fixed N. The most common example of the bounding function $A^*(t)$ in the literature is the *dual leaky-bucket* type of regulator (see [18, 19]) with $A^*(t) = \min\{Pt, \gamma t + \sigma\}$, where $\gamma < C$.

We now state our main result.

Theorem 1 (network decomposition for regulated traffic) *Under Assumption (A1), we have*

$$\lim_{N\to\infty} Q_I^N(0) = Q_{II}(0), \quad \textit{almost surely.} \tag{7}$$

Furthermore, the speed of convergence of $\mathbb{P}\{Q_I^N(0) > x\}$ *to* $\mathbb{P}\{Q_{II}(0) > x\}$ *is uniformly at least exponentially fast in the sense that*

$$\limsup_{N\to\infty} \frac{1}{N} \log \left(\sup_{x\geq 0} \left| \mathbb{P}\{Q_I^N(0) > x\} - \mathbb{P}\{Q_{II}(0) > x\} \right| \right) \quad \leq \quad -I(0), \tag{8}$$

where x *is the buffer level and* $I(0)$ *is from (5).* □

Proof of Theorem 1: We first observe from assumption (A1) that there exists a $t_1 > 0$ such that, for all i,

$$A_i(s, s+t) < Ct \quad \text{for all } s \text{ and } t \geq t_1. \tag{9}$$

Similarly, there exists a t_2 such that $\sum_{i\in\Gamma} A_i(-t, 0) + R(-t, 0) < C_d t$ for $t \geq t_2$, since $\sum_{i\in\Gamma} A_i(-t, 0) + R(-t, 0) \leq |\Gamma| A^*(t) + R^*(t)$. Thus, we have

$$Q_{II}(0) := \sup_{t\geq 0} \Big[\sum_{i\in\Gamma} A_i(-t, 0) + R(-t, 0) - C_d t \Big] = \sup_{0\leq t\leq t^*} \Big[\sum_{i\in\Gamma} A_i(-t, 0) + R(-t, 0) - C_d t \Big] \tag{10}$$

for any $t^* \geq t_2$, since the inside of the bracket becomes negative for $t > t_2$.

We let $q_i^N(t)$ denote the workload in $q^N(t)$ due to flow i. Then, the following claim is straightforward to show from the regularity assumption.

Claim: $q_i^N(t) \leq A^*(t_1)$ for all i and t, where t_1 is given by the relation (9).

Proof of Claim: We note that the maximum busy period of the queue $q^N(t)$ is bounded by t_1. To see this, suppose that there exists a busy period longer than t_1. Then, at the end of this busy period, the total amount of traffic arrival should be larger than the capacity NCt_1, otherwise, the busy period ends earlier. So we have $\sum_{i=1}^N A_i(s, s+t_1) \geq NCt_1$ for some s, but this contradicts (9). Now, note that the workload due to the flow $i \in \Gamma$ at time t should be bounded by the case that all the traffic from flow i has been accumulated during the maximum busy period t_1. Hence, we have $q_i^N(t) \le A_i(t-t_1,t)$ and from the definition of $A^*(t)$, $A_i(t-t_1,t) \le A^*(t_1)$. Thus, the claim follows. $\square$

Note that from the above claim, for all t, we have

$$D_i^N(-t,0) = A_i(-t,0) + q_i^N(-t) - q_i^N(0) \le A^*(t) + A^*(t_1).$$

This implies that the departure traffic $D_i^N(-t,0)$ is also bounded by some function $D^*(t) = A^*(t) + A^*(t_1)$ and $\lim_{t\to\infty} D^*(t)/t = a^*$. Hence, as in (10), there exists a $t_3 > 0$ such that

$$Q_I^N(0) := \sup_{t\ge 0} \Big[\sum_{i\in\Gamma} D_i(-t,0) + R(-t,0) - C_d t\Big] = \sup_{0\le t\le t^*} \Big[\sum_{i\in\Gamma} D_i(-t,0) + R(-t,0) - C_d t\Big] \quad (11)$$

for any $t^* \ge t_3$.

Now choose $t_0 = \max\{t_2, t_3\}$. Then, from (10) and (11), we have, for any $N > 0$,

$$\begin{aligned}
&\Big|Q_I^N(0) - Q_{II}(0)\Big| \\
&= \Big|\sup_{t\ge 0}\Big[\sum_{i\in\Gamma} D_i^N(-t,0) + R(-t,0) - C_d t\Big] - \sup_{t\ge 0}\Big[\sum_{i\in\Gamma} A_i(-t,0) + R(-t,0) - C_d t\Big]\Big| \\
&= \Big|\sup_{0\le t\le t_0}\Big[\sum_{i\in\Gamma} D_i^N(-t,0) + R(-t,0) - C_d t\Big] - \sup_{0\le t\le t_0}\Big[\sum_{i\in\Gamma} A_i(-t,0) + R(-t,0) - C_d t\Big]\Big| \\
&\le \sup_{0\le t\le t_0}\Big|\sum_{i\in\Gamma} D_i^N(-t,0) - \sum_{i\in\Gamma} A_i(-t,0)\Big| \\
&= \sup_{0\le t\le t_0}\Big|\sum_{i\in\Gamma} q_i^N(-t) - \sum_{i\in\Gamma} q_i^N(0)\Big| \\
&\le \sup_{0\le t\le t_0}\sum_{i\in\Gamma} q_i^N(-t) \\
&\le \sup_{0\le t\le t_0} q^N(-t). \qquad (12)
\end{aligned}$$

Since the distribution of $q^N(t)$ is independent of t, we have

$$\begin{aligned}
\mathbb{P}\Big\{\big|Q_I^N(0) - Q_{II}(0)\big| > 0\Big\} &\le \mathbb{P}\Big\{\sup_{0\le t\le t_0} q^N(-t) > 0\Big\} \\
&\le \sum_{t=0}^{t_0}\mathbb{P}\{q^N(-t) > 0\} = (t_0+1)\mathbb{P}\{q^N(0) > 0\}. \qquad (13)
\end{aligned}$$

Since assumption (A1) clearly ensures that (3) holds, taking logs, dividing by N, and then taking the lim sup on both sides of (13) yields

$$\begin{aligned}
\limsup_{N\to\infty}\frac{1}{N}\log\mathbb{P}\Big\{\big|Q_I^N(0) - Q_{II}(0)\big| > 0\Big\} &\le \limsup_{N\to\infty}\frac{1}{N}\log(t_0+1)\mathbb{P}\{q^N(0) > 0\} \\
&= \limsup_{N\to\infty}\frac{1}{N}\log\mathbb{P}\{q^N(0) > 0\} \\
&\le -I(0), \qquad (14)
\end{aligned}$$

where the last inequality comes from (4).

Finally, for $t \in \mathbb{Z}_+$, it has been shown that [12, 13]

$$I(0) := \inf_{t>0} \sup_{\theta} \left[\theta C t - \log \mathbb{E}\{e^{\theta A_1(0,t)}\}\right] = \sup_{\theta} \left[\theta C - \log \mathbb{E}\{e^{\theta A_1(0,1)}\}\right]. \quad (15)$$

From the stability condition, i.e., $C > \mathbb{E}\{A_1(0,1)\} = \lambda$, and the convexity of the function $\log \mathbb{E}\{e^{\theta A(0,1)}\}$ in θ, $I(0)$ in (15) is always positive. Thus, from (14), we can write

$$\sum_{N=1}^{\infty} \mathbb{P}\{|Q_I^N(0) - Q_{II}(0)| > 0\} \leq \sum_{N=1}^{\infty} \exp(-NI(0) + o(N)) < \infty.$$

Hence, (7) follows from the Borel-Cantelli lemma† [20]. Also, (8) follows from (14) and by noting that, for any $x > 0$,

$$\left|\mathbb{P}\{Q_I^N(0) > x\} - \mathbb{P}\{Q_{II}(0) > x\}\right| \leq \mathbb{P}\{|Q_I^N(0) - Q_{II}(0)| > 0\}.$$

This completes the proof of Theorem 1. ■

Remark 1 In Theorem 1, our primary emphasis is on the case when Γ is fixed, for which $\mathbb{P}\{Q_I^N(0) > x\}$ does not go to zero as N increases. However, our results remain unchanged if $|\Gamma|$, the number of flows feeding the downstream queue, also increases as N increases provided that the capacity of the downstream queue C_d is also scaled to ensure stability. For example, if the interfering traffic $R(s,t) = 0$ and $\Gamma = \{1, 2, \ldots, N\}$, we note that

$$\mathbb{P}\{Q_I^N(0) > 0\} \leq \mathbb{P}\{Q_{II}(0) > 0\} + \left|\mathbb{P}\{Q_I^N(0) > 0\} - \mathbb{P}\{Q_{II}(0) > 0\}\right|. \quad (16)$$

The first term of the RHS of (16) decrease to zero exponentially fast with rate $I(0) > 0$, and so does the second term from (8). Following the same approach as in the proof of Theorem 1, we see that the queue-length at the downstream queue also decreases to zero almost surely. However, this case is uninteresting because both $Q_I^N(0)$ and $Q_{II}(0)$ go to zero almost surely.

In the proof of Theorem 1, the fact that $I(0)$ is positive plays a crucial role in establishing the convergence of $Q_I^N(t)$ to $Q_{II}(t)$. If we set $b = 0$ in (4), we see that $q^N(t)$, where the aggregation takes place, also decreases to zero almost surely by the Borel-Cantelli lemma. Hence, the departure traffic flows are more likely to be identical to arrival traffic flows, as the system size increases. An interesting interpretation of these observations is the following: *If congestion occurs in a network, the node under congestion will either be unstable or carry a small number of traffic flows at high utilization.*

†Borel-Cantelli Lemma: For a sequence of random variables X_n, if $\sum_{n=1}^{\infty} \mathbb{P}\{|X_n| > \epsilon\} < \infty$ for any $\epsilon > 0$, then X_n converges to zero almost surely.

Although convergence of $\mathbb{P}\{q^N(0) > 0\}$ to zero was used in order to prove Theorem 1, this is in general not necessary for the decomposition to hold, as will be seen for point process inputs in Section 3.6. Further, we will show in Section 4 through simulations that removing nodes and thus simplifying network analysis still remains in effect even if the actual value of $\mathbb{P}\{q^N(0) > 0\}$ is not very small.

3.1.1 Estimating the constant $I(0)$

In this section, we briefly examine how we can estimate the error $e^{-NI(0)}$, and how small it can be. As we see in (15), the constant $I(0)$ depends on the distribution of the traffic flow. Consequently, it is not obvious how one would be able to estimate it. Recall that this error term $e^{-NI(0)}$ is an upper bound on the *maximum* difference in the overflow probability between two scenarios. Thus, the actual error for a specific buffer level could be significantly smaller than the maximum error. Nevertheless, it turns out that we are able to find a lower bound on the constant $I(0)$ (hence, an upper bound on the error term $e^{-NI(0)}$ as a first-order approximation) in terms of the mean and the peak rate of the traffic that are relatively easy to estimate.

Let P be the peak rate of a traffic flow $A_i(s,t)$. Observe that

$$\begin{aligned} I(0) &:= \inf_{t>0} \sup_{\theta} \left[\theta C t - \log \mathbb{E}\{e^{\theta A_1(0,t)}\}\right] \\ &= \inf_{t>0} \sup_{\theta} \left[\theta C - \log \mathbb{E}\{e^{\theta \frac{A_1(0,t)}{t}}\}\right]. \end{aligned}$$

Since $0 \leq \frac{A_1(0,t)}{t} \leq P$ with $\mathbb{E}\{\frac{A_1(0,t)}{t}\} = \lambda$, using Corollary 2.4.5 in [9], we have

$$\mathbb{E}\{e^{\theta \frac{A_1(0,t)}{t}}\} \leq \frac{\lambda}{P} e^{\theta P} + (1 - \frac{\lambda}{P}). \tag{17}$$

Thus, we get

$$\sup_{\theta} \left[\theta C - \log \mathbb{E}\{e^{\theta \frac{A_1(0,t)}{t}}\}\right] \geq \sup_{\theta} \left[\theta C - \log(\frac{\lambda}{P} e^{\theta P} + (1 - \frac{\lambda}{P}))\right] = \sup_{\theta} h(\theta),$$

where

$$h(\theta) := \theta C - \log(\frac{\lambda}{P} e^{\theta P} + (1 - \frac{\lambda}{P})).$$

It is easy to see that $h(\cdot)$ is a concave function and $h'(0) = C - \lambda > 0$. Thus, by direct calculation, we have

$$I(0) \geq \frac{C}{P} \log(\frac{C}{\lambda}) + (1 - \frac{C}{P}) \log(\frac{P - C}{P - \lambda}), \tag{18}$$

where $\lambda < C < P$.

Figure 4 shows the right hand side of (18) as a function of the ratio between the peak rate and the mean rate (P/λ) for two different utilizations. For example, if $P = 5\lambda$ and $\rho = \lambda/C =$

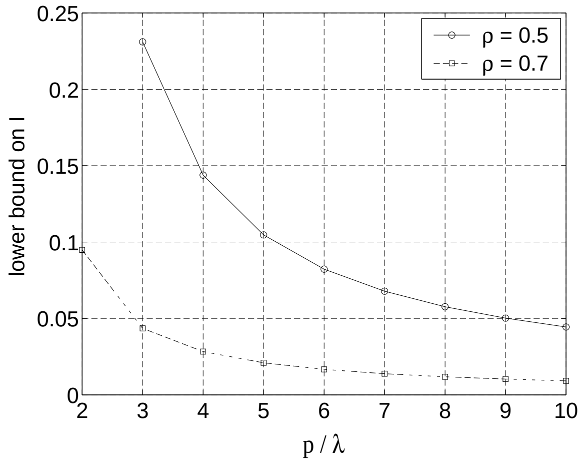


Figure 4: RHS of (18) as a function of P/λ for two different utilizations

$(N\lambda)/(NC) = 0.5$, then Figure 4 shows that $I \geq 0.1$. Hence, if there are 200 such sources being multiplexed, then

$$\exp(-NI+o(N)) \leq \exp(-200 \times 0.1+o(200)) \approx 2\times 10^{-9}.$$

Thus, the maximum error on the overflow probability at the downstream queue incurred by traversing the queue with 200 such flows is at most 2×10^{-9}. Note that the equality in (17) corresponds to the special case that the traffic rate is either the peak rate with probability λ/P or zero with probability $1 - \lambda/P$. If we have more information on the marginal distribution of the traffic, e.g., the variance, we can then find a sharper bound on the constant $I(0)$.

3.2 General (non-regulated) traffic

In the previous section, we have assumed that the traffic flows of interest are regulated. For non-regulated traffic arrivals, i.e., there is no bounding function $A^*(t)$, things are more complicated. In this case, the difference between $Q_I^N(0)$ and $Q_{II}(0)$ now depends on the entire past history of $q^N(t)$, i.e., $\sup_{t\geq 0} q^N(-t)$, rather than $\sup_{0\leq t\leq t_0} q^N(-t)$ as we see in (12) for regulated traffic arrivals.

Mathematically speaking, for regulated traffic arrivals, the point-wise convergence of $D_i^N(-t,0)$ to $A_i(-t,0)$ is sufficient to prove that of $Q_I^N(0)$ to $Q_{II}(0)$, since the busy period of the upstream queue ($q^N(t)$) is strictly bounded by a constant. However, in general, the point-wise convergence of the input traffic does not guarantee the convergence of the corresponding queue-length [6]. To see this, consider an input $X^N(-t,0)$ to a queue with capacity C as[‡]

$$X^N(-t,0) = \begin{cases} Ct & \text{if } 0 \leq t \leq N, \\ Ct+1 & \text{if } t > N. \end{cases}$$

[‡]This example is taken from [6] to clarify the point.

Clearly, as N increases, $X^N(-t,0)$ converges to a constant rate process Ct for any *fixed* $t > 0$. However, the queue process with input $X^N(-t,0)$ does not converge to that with the constant rate process Ct. Hence, the simple fact that the upstream queue decreases to zero over any finite period of time, does not immediately apply for general (non-regulated) traffic arrivals. Nevertheless, we are still able to show that the distribution of $Q_I^N(t)$ converges uniformly to that of $Q_{II}(t)$, and that the convergence happens at least exponentially fast.

We define $q_i(t)$ as the stationary workload with capacity C fed by single input $A_i(s,t)$, i.e.,

$$q_i(t) = \sup_{s \le t} \big[A_i(s,t) - C(t-s)\big]. \tag{19}$$

Since $A_i(s,t)$ $(i = 1,2,\ldots,N)$ are *i.i.d.* with stationary increments, $q_i(t)$ $(i = 1,2,\ldots,N)$ are also *i.i.d.* and stationary. We then assume the following:

(A2): There exists $\epsilon > 0$ such that $\mathbb{E}\{|q_i(0)|^{1+\epsilon}|\} < \infty$.

Assumption (A2) is quite general in that it includes almost all traffic models typically considered in the literature. For example, any long-range dependent traffic model with $\log \mathbb{P}\{q_i(0) > x\} \sim -\alpha x^\beta$, where $\alpha > 0$ and $0 < \beta \le 1$, satisfies assumption (A2). In fact, in this case, all the moments of $q_i(0)$ exist. Also, note that even if the workload is Pareto-distributed (i.e., having infinite variance) with parameter $1 < p \le 2$, assumption (A2) still holds with $\epsilon = (p-1)/2 > 0$.

Theorem 2 (network decomposition for general arrivals in the discrete time) *Suppose that (3) and Assumption (A2) are satisfied. Then, we have*

$$\lim_{N\to\infty} \big|\mathbb{P}\{Q_I^N(0) > x\} - \mathbb{P}\{Q_{II}(0) > x\}\big| = 0,$$

uniformly in $x > 0$. □

To address the speed of convergence in Theorem 2, we need the following assumptions.

(A3) There exist $H_1, H_2 \in (0,1)$ such that $v_A(t) := \mathrm{Var}\{A_i(-t,0)\} \in O(t^{2H_1})$ and $v_R(t) := \mathrm{Var}\{R(-t,0)\} \in O(t^{2H_2})$.§

(A4) $\mathbb{E}\{Q_{II}(0)\} < \infty$, if the queue is stable, i.e., $C_d > \lambda|\Gamma| + \bar{r}$.

The parameters H_1 and H_2 in assumption (A3) correspond to the "Hurst parameter" in the literature. This has been widely used to model long-range dependent traffic or self-similar traffic [21, 22, 23], for which $H \in (0.5, 1)$. Note that assumption (A4) is almost trivial in that it only requires finiteness of the expected workload when the service capacity is larger than the mean arrival rate.

§ $f(t) \in O(g(t))$ means $\limsup_{t\to\infty} f(t)/g(t) < \infty$.

Corollary 1 *Suppose that the assumptions in Theorem 2 hold and also Assumptions (A3) and (A4) hold. Then, there exists a positive constant* J^* $(0 < J^* < I(0))$ *such that*[¶].

$$\limsup_{N\to\infty} \frac{1}{N} \log \left(\sup_{x\geq 0} \left| \mathbb{P}\{Q_I^N(0) > x\} - \mathbb{P}\{Q_{II}(0) > x\} \right| \right) \leq -J^*. \tag{20}$$

□

Since the proofs of the above results are quite technical and beyond the scope of this paper, we omit them here. See [15] for complete proofs.

Note that, since Theorem 2 also holds for non-regulated traffic arrivals, we are unable to prove almost sure convergence as in (7) of Theorem 1. However, we can show convergence in distribution and show that the speed of convergence is still exponential, albeit with a rate slower than $I(0)$. We note that in this case, our decomposition approach still remains intact. The overflow probability at a downstream queue can be replaced with that of a single queue, in a manner similar to the regulated traffic arrivals.

3.3 General traffic in the continuous time domain

We now consider general traffic arrivals $A_i(s,t)$ in the continuous time domain, i.e., $s, t \in \mathbb{R}$. In this case, the following assumption is also required in order for the many-sources-asymptotic upper bound (4) to hold.

(A5): $\limsup_{t\to 0} \mathbb{E}\{ \exp(\theta \sup_{0\leq u\leq t} |A_i(0,u)|)\} = 1, \ \forall \theta > 0.$

In [12], assumption (A5) has been shown to be sufficient to carry the proof of the many-sources-asymptotic in the discrete time case, over to the continuous time case (see also [24]). This assumption is merely a technical one, and will be satisfied for most cases of practical interest.

Having equipped with the many-sources-asymptotic (see (4)) in the continuous time domain, we still need to ensure that $I(0)$ is positive in this case. While $I(0)$ is always positive in the discrete time cases, it may not be true for the continuous time model. We have already observed from (18) that $I(0) > 0$ if the peak rate of a traffic arrival is bounded. However, it turns out that $I(0)$ is in fact positive for a much larger class of traffic arrivals, as shown below.

(A6): We assume that $A_i(0,t)$ satisfies one of the following:

(a) For all $t > 0$, $0 \leq A_i(0,t) \leq Pt$ for some $P < \infty$.

(b) $A_i(0,t)$ can be represented as the integral of a stationary process $r_i(t)$, i.e., for all $t \geq 0$, $A_i(0,t) = \int_0^t r_i(u)du$ with $\mathbb{E}\{ \sup_{0\leq u\leq \epsilon} |r_i(u)|\} < \infty$ for sufficiently small $\epsilon > 0$.

[¶]The constant J^* is a function of the constant ϵ in (A2), Hurst parameters in (A3), and the rate function $I(0)$ in (5). In any case, it is always positive. See [15] for details.

Proposition 2 (Proposition 2 in [15]) *Suppose (A6) holds for the continuous time case. Then* $I(0) > 0$.

Proposition 2 guarantees that $I(0)$ is positive and hence $q^N(t)$ decreases to zero if (i) there exists a peak rate for each traffic arrival or (ii) the traffic arrival $A_i(s,t)$ is smooth enough such that we can define a well-behaved "instantaneous traffic rate" ($r_i(t)$). We now state the following result [15].

Theorem 3 (network decomposition for general arrivals in the continuous time) *Suppose that (3) and assumptions (A2) – (A6) are satisfied. Then, Theorem 2 and Corollary 1 remain unchanged for the continuous time model.* □

3.4 Extension to non-*i.i.d.* traffic arrivals

We have assumed that the arrival processes to the upstream queue in Figure 2 are *i.i.d.* In this section, we extend the result to the case of non-*i.i.d.* traffic arrivals.

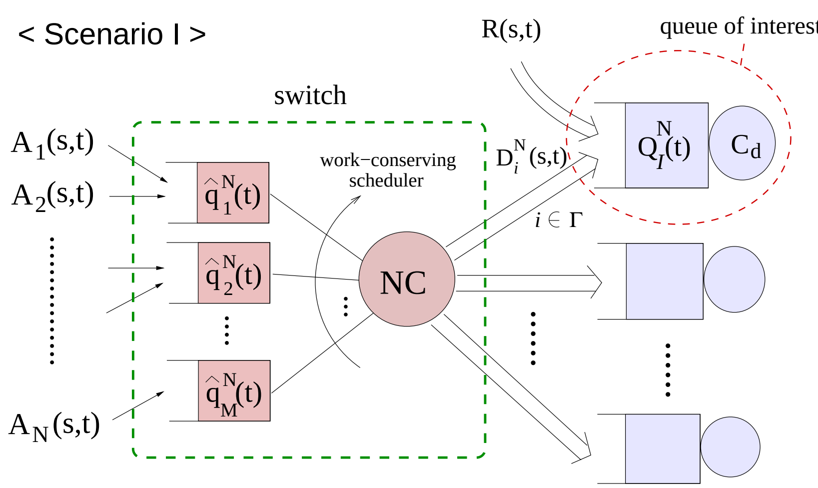


Figure 5: A multi-buffered queueing system

Consider a multi-buffered queueing system shown in Figure 5. In this figure, we assume that there are M (M is finite) different traffic classes, each of which is associated with a queue with queue-length $\hat{q}_j^N(t)$ at time t, ($j = 1, 2, \ldots, M$) and that the total service capacity is again NC. This capacity is being distributed to each class in a work-conserving manner. Then, note that for any work-conserving scheduling policy, we can write

$$q^N(t) = \sum_{j=1}^{M} \hat{q}_j^N(t),$$

where $q^N(t)$ is defined in (1).

Let $S_j(N)$ be the set of flows of class j ($j = 1, 2, \ldots, M$) that arrive to the queue $\hat{q}_j^N(t)$. As before, we scale the upstream queue such that the total number of flows and total service capacity increase proportionally. Note, however, that the number of flows for each class can be arbitrary as long as

$$\sum_{j=1}^{M} |S_j(N)| = N,$$

where $|S_j(N)|$ represents the number of flows for class j. We assume that $|S_j(N)|/N$ converges to a number $\gamma_k \in [0, 1]$ as N increases. We also assume that within a class, the flows are *i.i.d.*, i.e., $A_i(s,t), i \in S_j(N)$ are *i.i.d.* for any j. We however allow the flows for different classes to be possibly *heterogeneous* and *dependent*. We then have the following result (for the proof, see [15]).

Proposition 3 (network decomposition for non-*i.i.d.* traffic arrivals) *Suppose that we require the corresponding assumptions for each traffic arrival only within a class, (we allow any heterogeneity and dependency among different classes) and that $A_i(s,t)$, the amount of traffic arrival in $[s,t)$ is nonnegative. Then, Theorem 3 holds.*

Proposition 3 implies that when the system is large, the scheduling mechanism has little effect on the queueing performance at the next stage, and our simplifying approach is still valid (by removing the multi-buffered switch at the first stage in Figure 5). The proposition holds under general work-conserving scheduling policies such as static priority queueing or GPS [25], etc. For instance, if a user session consists of different classes of traffic, such as voice, video and data, we can allow any dependency and heterogeneity among these traffic and also can assign different priorities to each class.

3.5 Simplifying network analysis

Note that relation (8) in Theorem 1 and (20) in Corollary 1 allow us to write

$$\sup_{x \geq 0} \left| \mathbb{P}\{Q_I^N(0) > x\} - \mathbb{P}\{Q_{II}(0) > x\} \right| \quad \leq \quad \exp(-NJ^* + o(N)),$$

for some positive constant J^*. Thus, for any target QoS ($\mathbb{P}\{Q_I^N(0) > x\}$) in the original two-stage queueing system, we can replace it by a simpler system in which the first node has been removed and the maximum error is less than $\exp(-NJ^* + o(N))$, which decreases exponentially fast. If we extend this idea to an entire network, we can say that it is safe to remove nodes that have the capacity to carry a large number of flows from consideration. This simplification greatly reduces the burden of analyzing a queueing network, and helps simplify the difficult end-to-end QoS estimation problem into potentially a simpler single-stage queueing problem.

Although our model assumes that the upstream node has capacity proportional to N all the time, we can easily strengthen our results by noting that the network decomposition results apply as long as the capacity of the upstream queue increases in proportion to N *eventually*. This is a reasonable assumption in that a large system will, eventually, operate at a utilization level which is bounded away from 1 in the future, despite regular upgrades of the system.

Such a decomposition result would have important implications for network design and resource allocation/capacity planning problems. In designing a network where a number of flows are served, we need to understand where to deploy and how to assign the network resources, i.e., the link capacity and/or the buffer, to meet the QoS requirements. As our decomposition approach provides a simple, yet efficient tool for simplifying network analysis, we could fine-tune the network parameters to obtain desired network performances in terms of QoS.

From a measurement point of view this type of decomposition could also be useful. At one extreme, if one is able to access any point inside a network to obtain a sample path of a traffic flow, we can then simply measure the traffic and compute the QoS at the node in hand (for example, we could employ the "measurement-analytic" approach in [26, 27]). However, measurements in the interior of a network could be costly or even practically infeasible. To avoid this problem, we could use the decomposition idea (e.g., via Theorem 1) and measure the traffic only at the network periphery to make QoS predictions within the network.

Another feature of our decomposing approach is that we do not have to maintain any per-flow information at the interior nodes with large capacity, even if each flow requires stringent end-to-end QoS. Once we remove these nodes to simplify network analysis, they become invisible to us. In other words, inside a network, we do not really care where traffic flows are coming from and heading to, as long as all the nodes are kept stable. In Section 4, we will verify via simulations that our decomposition method works well under a variety of different network configurations and different traffic types.

3.6 Point process arrivals

So far, we have considered "fluid-like" traffic arrivals. The fluid models in this paper include regulated traffic arrivals, general arrivals with fixed time slots as in the discrete time model, or general arrivals satisfying (A6) in the continuous time domain. As we have noted, the rate function $I(0)$ is positive for any of these fluid models, which result in the workload at the queue $q^N(t)$ to decrease to zero almost surely as N increases. This fact plays a vital role in establishing the convergence of $Q_I^N(0)$ to $Q_{II}(0)$ under the various cases, although the point-wise convergence itself is not sufficient for non-regulated traffic arrivals.

In this section, we consider the case of point process inputs. As an illustrative example, suppose that each traffic arrival is a Poisson process with mean rate λ. It is then straightforward to see that the distribution of $q^N(t)$ does not vary with N [28]. More generally, it has recently been shown [29] that for general stationary point process inputs, $q^N(t)$ converges in distribution to a queue-length random variable assuming that the input is Poisson. Hence, for point process inputs, the queue $q^N(t)$ does not converge to zero in any sense, thereby implying that $I(0) = 0$. Hence, our previous analysis does not immediately apply to the case of point process inputs.

Let $A_i(s,t)$ here represent the number of packets that arrive during a time interval $(s,t]$ for flow i. We model each traffic arrival $A_i(s,t)$ as a *simple stationary point process* [30], i.e., we require that at most one packet can arrive at any instant and that $A_i(s, s+t)$ be stationary in s. Then, the next result shows that we can still decompose the network in the following sense.

Theorem 4 (network decomposition for point process arrivals) *Suppose that (3), (A2), and (A5) hold. Then, for any $x \geq 1$ and any given $\delta_1, \delta_2 > 0$, we have*

$$\begin{aligned}\mathbb{P}\{Q_{II}(0) > x + \delta_1\} &\leq \liminf_{N\to\infty} \mathbb{P}\{Q_I^N(0) > x\} \\ &\leq \limsup_{N\to\infty} \mathbb{P}\{Q_I^N(0) > x\} \leq \mathbb{P}\{Q_{II}(0) > x - 1 - \delta_2\}.\end{aligned}$$

□

We here provide some basis of the proof for Theorem 4. Since Γ is finite, the key is to notice that each individual departure flow still converges to the arrival in some sense even when $q^N(t)$ does not go to zero. Let $q_i^N(t)$ be the amount of backlog for flow i in the queue. Then, clearly, we have $\sum_{i=1}^N q_i^N(t) = q^N(t)$ and $\mathbb{E}\{q_i^N(t)\} = \mathbb{E}\{q^N(t)/N\}$ by symmetry. Further, since $I(x)$ is always positive for any positive x, we see from (4) that $q^N(t)/N$ still converges to zero almost surely. Hence, with assumption (A2), we can show that $q_i^N(t)$ also converges to zero in the mean, i.e., $\lim_{N\to\infty} \mathbb{E}\{q_i^N(t)\} = 0$. This means that each departure flow $(D_i^N(-t,0))$ still converges in the mean to the corresponding arrival $(A_i(-t,0))$ for any *fixed* t. Still, as before, the biggest step in the proof is to deal with the convergence over an infinite horizon, i.e., for *all* $t > 0$. This is beyond the scope of this paper, and we refer to our technical report [16] for complete proofs.

Theorem 4 tells us that the original overflow probability at the downstream queue ($\mathbb{P}\{Q_I^N(0) > x\}$) can be approximated by that of a single queue ($\mathbb{P}\{Q_{II}(0) > x\}$) with one packet offset when N is large. Hence, all the qualitative observations in Section 3.5 still hold for point process arrivals.

As shown earlier, for fluid models in which $I(0) > 0$, $\mathbb{P}\{Q_I^N(0) > x\}$ converges uniformly at least exponentially fast to $\mathbb{P}\{Q_{II}(0) > x\}$. In contrast, for point process inputs, obtaining the speed of convergence in Theorem 4 turns out to be much more challenging. As a special case, we have obtained an upper bound on the speed of convergence for Poisson inputs and found that it is slow with a speed of $O(1/N^{\frac{1}{6}})$. It could be that this is because the bound that we have obtained is conservative, or this is in fact the price we have to pay for $q^N(t)$ itself not converging to zero in the case of point processes, as compared to the "fluid-like" processes. However, in practice, it turns out that the actual speed of convergence appears to be fast enough and the network decomposition remains unaffected even for a moderate degree of flow aggregation (see Section 4.1.2).

Intuitively, the one packet "offset" for the upper bound in Theorem 4 can be explained as follows: For a simple point process, on its sample path basis, a packet can arrive at any time instant. Putting it in a different way, this means that no matter how smaller an interval we choose, the "amount" of traffic arrival during this interval does not always decrease to zero due to the discrete nature of point processes. Thus, the sample path of $q_i^N(t)$ jumps up and down like a staircase, implying that the departure $D_i^N(s,t)$ and the arrival $A_i(s,t)$ can differ by one packet at any time instant, and so can $Q_I^N(0)$ and $Q_{II}(0)$.

4 Numerical results

In this section, we report on an extensive set of simulations conducted to numerically test the performance of our network decomposition approach under different network scenarios. To represent a realistic system as closely as possible, we have developed a packet-based (versus fluid), event-driven simulator that consists of a number of queues with arbitrary input traffic arrivals we designate. This packet-based simulator is desirable since it is able to capture the case of $I(0) = 0$, e.g., each flow is modeled by a point process, as well as the case when $I(0) > 0$ (See (5) and (20)). In our simulator, we are able to monitor every packet anywhere in the network, each of which is associated with its packet size and time stamp. The QoS metrics we consider are the overflow probability at a particular node we are monitoring, and the end-to-end delay for a particular traffic flow. To be specific, we record queue sizes of a node just before each packet enters the node, and then take the average (over the number of packets) to estimate the overflow probability at that node. Similarly, we measure the end-to-end delay per packet for the traffic flow we are monitoring, and then take the average. For calculating the end-to-end delay of each packet, we only consider the queueing delay at each node and ignore the propagation delay, since the propagation delay is a constant value.

4.1 Two-stage queueing network

4.1.1 Voice traffic sources

We first apply our decomposition approach to a simple two-stage queueing network, as shown in Figure 2. The arrival traffic to the first queue consists of N multiplexed voice traffic sources. For a single voice traffic, we use an on-off Markov Modulated Deterministic (MMD) model. This is a continuous-time two-state Markov chain, which generates traffic at a constant rate (64 Kbps) while in the "on" state, and no traffic while in the "off" state. The sojourn time in the "on" and "off" states are exponentially distributed with mean 352 msec and 650 msec, respectively [31]. To simulate the constant rate during the on state using our simulator, we generate a series of packets with constant inter-arrival time where each packet has a fixed size of 53 bytes[‖].

We fix the number of voice traffic flows feeding the downstream queue as 10, i.e., $|\Gamma| = 10$. However, we vary the number of voice traffic flows at the upstream queue (N) while keeping the utilization there the same, and then, we vary the utilization at the upstream queue while keeping N the same. These experiments allow us to investigate the effect of different levels of aggregation and utilization at the upstream queue on the behavior of the downstream queue. For interfering traffic at the downstream queue, $R(s,t)$, we generate fractional Gaussian noise sequence with Hurst parameter $H = 0.8$, such that its mean is chosen to be equal to that of 10 multiplexed voice traffic flows. We choose the capacity of the downstream queue such that its utilization is 0.8.

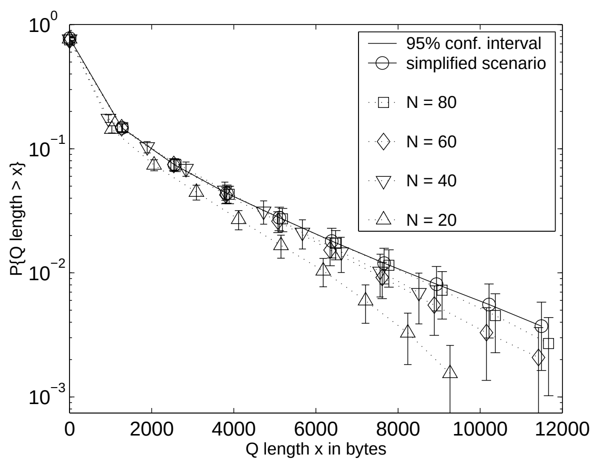


Figure 6: Buffer overflow probability at the downstream queue for different N at the upstream queue. ρ is fixed to 0.8.

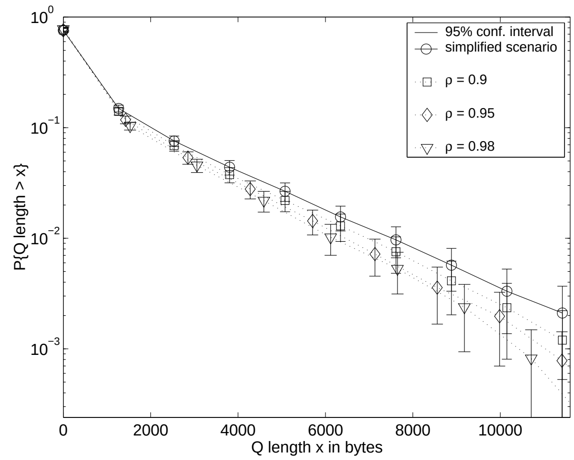
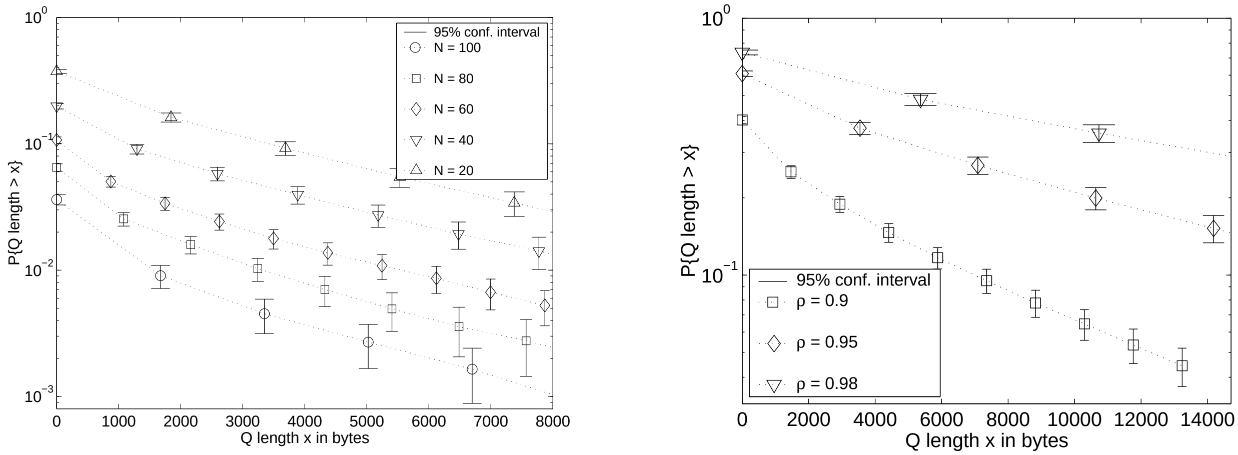


Figure 7: Buffer overflow probability at the downstream queue for different utilization ρ at the upstream queue. N is fixed to 60.

Figure 6 shows the different overflow probabilities at the downstream queue for different values of N at the upstream queue. The utilization, ρ, at the upstream queue remains at $\rho = NC/N\lambda = 0.8$, where $\lambda = 22.48$Kbps is the mean traffic rate of a single voice traffic flow. We ran many independent experiments, and here report their average with 95% confidence interval for each point in the figure. As N, the number of aggregated traffic flows, increases, we see that the buffer overflow probability at the downstream queue approaches to that of a single queue obtained by removing the upstream queue (simplified scenario). In particular, when N is equal to 80, the curve is nearly indistinguishable from the simplified scenario case. We now fix the number of multiplexed voice traffic flows N at $N = 60$, and then vary the utilization (i.e., the capacity) at the upstream queue. As Figure 7 clearly shows, we observe that removing the upstream nodes barely affects the queueing

[‖]The packet size has been chosen arbitrarily and any other packet size would suffice. Later in Section 4.1.2, for Ethernet traffic, we use traces with variable packet sizes.

performance at the downstream node, even for high utilizations.



(a) different N (b) different utilization ρ

Figure 8: Buffer overflow probability at the upstream queue

From the proof of Theorem 1 and Proposition 2, we have seen that for fluid arrivals, the queue-length at the upstream queue decreases exponentially fast as N grows. Figure 8 shows different overflow probability curves at the upstream node for (a) different N and (b) different utilizations ρ. Clearly, as N increases, the overflow probability decreases over all buffer levels. However, if we carefully compare Figure 8 to Figure 6 or Figure 7 for the same buffer levels, it turns out that the overflow probability at the upstream queue is not so small, and sometimes even larger than that of the downstream queue. Thus, although we used the fact that $q^N(t)$ decreases to zero in order to prove Theorems 1, 2, and 3, it does not appear to be necessary for the network decomposition technique to work well. This is also further demonstrated by Theorem 4, in which case the convergence holds without $q^N(t)$ decreasing to zero.

4.1.2 Ethernet (LAN) traffic trace

In this section, we repeat the same set of simulations as in Section 4.1.1, except that we now use an actual Ethernet (LAN) trace** instead of a traffic model. The Ethernet trace consists of a sequence of arrival times and packet sizes. To obtain each multiplexed Ethernet traffic, we generate different traffic traces from the original Ethernet trace with random offset (random starting points), and then superpose these traces. Since Ethernet traffic is known to be much burstier than the voice traffic [21], the question is whether we can still remove the upstream node in this case. Figures 9 and 10 show the overflow probability at the downstream queue under different scenarios. As N increases and/or ρ decreases, the curve becomes closer to the case of the simplified scenario, but the "convergence" happens slower than in the voice traffic case, due to the high level of burstiness

**Trace file can be obtained at "ftp://ita.ee.lbl.gov/traces/BC-Oct89Ext.TL.Z" of the Ethernet traffic.

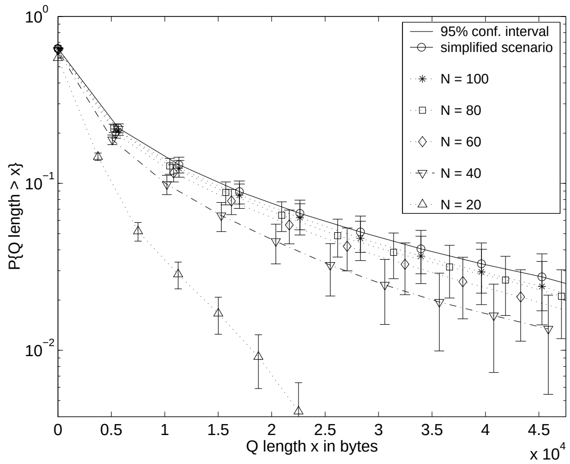


Figure 9: Buffer overflow probability at the downstream queue for different N. Utilization at the upstream queue is fixed to $\rho = 0.7$

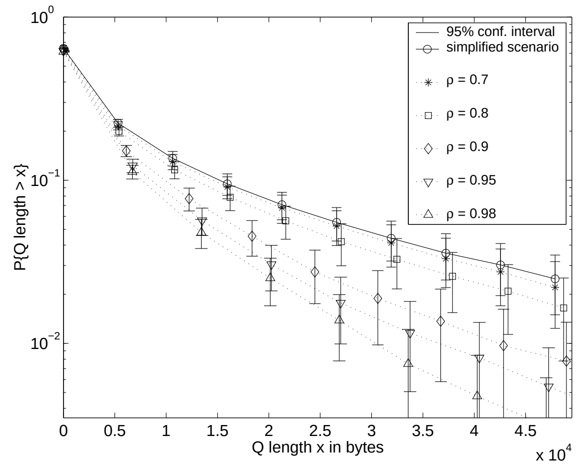


Figure 10: Buffer overflow probability at the downstream queue for different utilization ρ. N is fixed to 60.

Figure 11 presents the overflow probability at the upstream queue with N multiplexed Ethernet traffic arrivals. As in the voice traffic case, the overflow probability decreases as N increases. However, in contrast to Figure 8 (a), the probability that the buffer is non-empty stays around a fixed point as N varies, while for the voice traffic case it also decreases as N increases (see Figure 8 (a)). Remember that $\mathbb{P}\{q^N(t) > 0\}$ decreases to zero if $I(0) > 0$. Also, Proposition 2 tells us that $I(0) > 0$ whenever there exists a peak rate (even for continuous time traffic models). Since there is a peak rate ("on" state rate) for a *single* voice traffic, Figure 8 (a) is to be expected. However, for the Ethernet traffic trace, there is no pre-determined peak rate and thus $\mathbb{P}\{q^N(t) > 0\}$ does not go to zero in the range of N shown in Figure 11. This implies that a point process model would be a more suitable choice for the Ethernet trace. Nonetheless, as expected from Theorem 4, we point out that removing the upstream node still works well for most cases as shown in Figures 9 and 10. We have also observed similar results for other traffic arrivals, e.g, MPEG encoded video trace, Auto-Regressive models, etc, which we do not present here for space considerations.

We next investigate how different utilizations and levels of aggregation quantitatively affect the actual error of the overflow probability between two scenarios. For a given two-stage queueing network, we test if the following relation

$$\left|\mathbb{P}\{Q_I^N(0) > x\} - \mathbb{P}\{Q_{II}(0) > x\}\right| \leq K\mathbb{P}\{Q_{II}(0) > x\} \tag{21}$$

holds for all buffer levels x. Here, the constant K corresponds to some error margin around the target QoS. If (21) holds for $K = 0.1$, it means that the error between the two scenarios differs by only a 10% margin. For each traffic arrival and each N, we gradually decrease the utilization ρ,

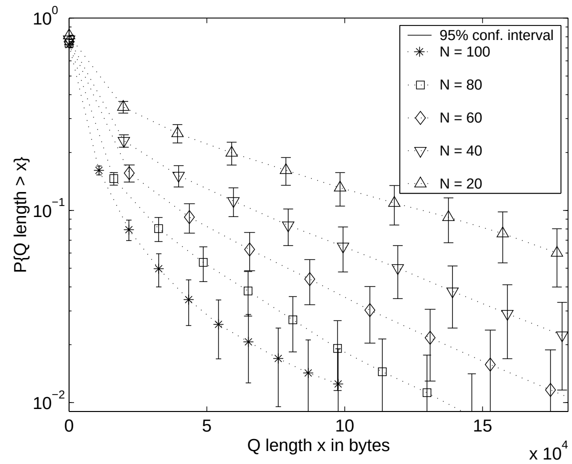


Figure 11: Buffer overflow probability at the upstream queue for different N

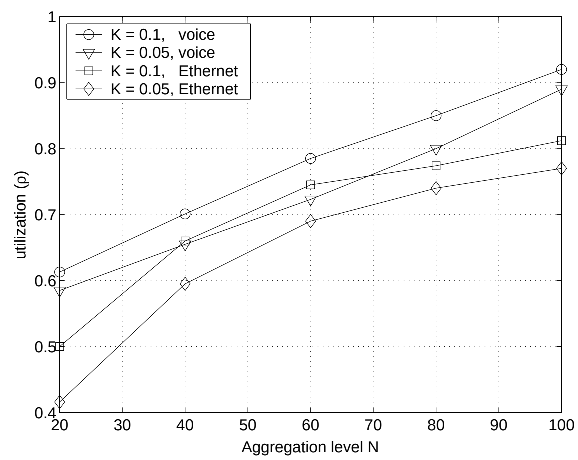


Figure 12: Relation between N and ρ for given error constraint (21) for voice and Ethernet traffic

i.e., increases the capacity at the upstream queue, until (21) holds for *all* buffer levels used in the simulation. We record the largest utilization value that satisfies (21) for all x. Figure 12 depicts the results of these tests for voice and Ethernet traffic. As the aggregation level (N) increases, the maximum possible utilization for (21) to hold also increases. For example, in Figure 12, for the Ethernet traffic case when $N = 60$ (recall that only 10 flows feed the downstream queue), we see that about 70% (or less) utilization at the upstream queue guarantees that the error is less than 5% of the overflow probability of the simplified scenario, regardless of the buffer level x.

4.2 Effect of the cross traffic

In this section we investigate how the amount of the cross traffic at the intermediate nodes affects the performance of the network decomposition approach. Consider Figure 13. In this figure, the original scenario consists of three-stage tandem queues where each queue is conveniently represented as a node. In the original scenario, 10 multiplexed Ethernet traffic flows (with aggregate mean rate λ_s) traverse nodes 1–3. At each node, different cross traffic flows are multiplexed with the Ethernet traffic flows. For each cross traffic, we vary the number of aggregated flows, which results in different mean rates (λ_c). Suppose that we want to estimate the overflow probability at node 3 that typically resides deep inside a network. As before, we obtain the simplified scenario by removing every node but node 3. We choose the capacity at each node such that its utilization is maintained at 0.7 all the time.

Figure 14 shows the overflow probability at node 3 for both scenarios with different loads of cross traffic. As the load of the cross traffic increases, the performance of the original scenario becomes closer to that of the simplified scenario. In particular, if the load of the background (cross) traffic is about 5 times or larger than that of the Ethernet traffic, removing all the other

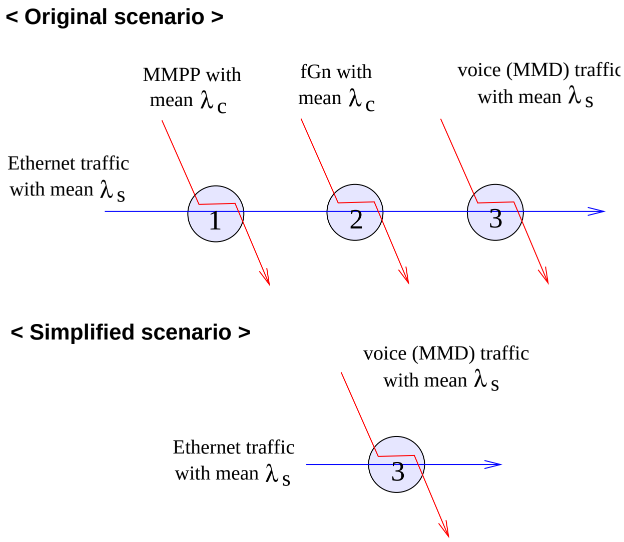


Figure 13: Three-stage queueing network with cross traffic flows

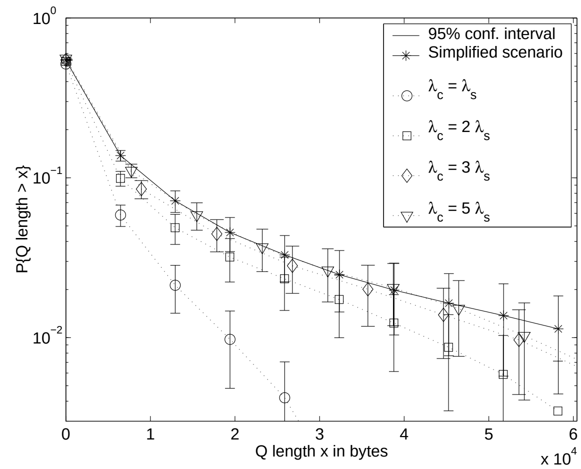


Figure 14: Buffer overflow probability at node 3 in Figure 13 with different λ_c. The utilization of each node is 0.7 at any time.

nodes does not incur any change in the overflow probability over most buffer levels. In other words, the Ethernet traffic does not seem to be significantly altered as it goes through nodes 1 and 2 due to the large amount of background traffic. Hence, with the previous examples of the two-stage queueing network, we can conclude that our decomposition approach works well when (i) the set of traffic flows of interest (here the 10 multiplexed Ethernet traffic) does not dominate at any node (i.e., it coexists with cross traffic), and/or (ii) the upstream queues serve not only the aggregate of *i.i.d.* traffic flows but with other types of traffic flows.

4.3 Multi-stage queueing network

In this section, we apply our simplifying techniques to a multi-stage queueing network. Consider the queueing network shown in Figure 15. In this figure, we assume that each node represents a queue with a certain capacity, where the numbers in parentheses stand for the capacity or the bandwidth of each node in Mbps. Nodes 5, 6, and 7 correspond to high-speed links (e.g., core routers), while the other nodes correspond to lower capacity links (e.g., edge routers) serving only a small fraction of flows. Traffic types and their routes are summarized in Table 1.

Let λ_i be the mean rate for flow Si. We choose the number of multiplexed traffic flows such that $2\lambda_1 = 2\lambda_4 = \lambda_3 = \lambda_5$, while the mean rate of S2 is about 20 times larger than the flow S1, i.e., $\lambda_2 = 20\lambda_1$. The resulting utilization of each node varies between 0.7 and 0.8 for each case reported here. We then remove node 5 from the original scenario shown in Figure 15 and record the overflow probability at node 7. Figure 16 shows the resulting simplified scenario. Contrary to the previous two-stage queueing network, in which the capacity of the downstream queue is much smaller than

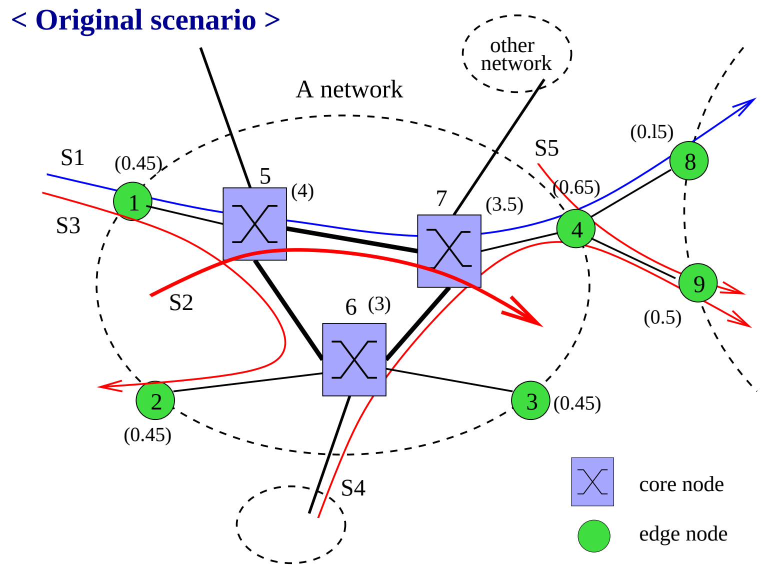


Figure 15: Multi-stage queueing network. The numbers in parentheses represent capacity at the nodes in Mbps.

Table 1: Description of traffic flows in Figure 15

aggregate flow	traffic type	routes
S1	10 Ethernet	1 - 5 - 7 - 4 - 8
S2	100 voice (MMF) or 6 MPEG video traffic	5 - 7
S3	20 fGn	1 - 5 - 6 - 2
S4	5 MMPP	6 - 7 - 4 - 9
S5	10 voices (MMF)	4 - 9

the upstream queue, note here that the capacity of node 7 is also large and comparable to that of node 5. Figure 17 shows the overflow probability at node 7 for the original scenario and the simplified scenario 1, while the interfering traffic S2 constitutes 100 multiplexed voice traffic flows for Figure 17(a) and 6 multiplexed MPEG video traffic flows for Figure 17(b). Since the capacity of node 5 is large in either case, the figures show that removing node 5 does not greatly affect the overflow probability at the downstream node.

Figures 17(a)-(b) also have several points worth mentioning. Since most traffic flows at node 5 arrive to node 7, we expect from Theorem 1 (or Theorem 2 or 3) that the overflow probability at node 7 itself goes to zero, as the number of multiplexed flows increases at the previous node. Therefore, when S2 constitutes 100 multiplexed voice traffic flows which correspond to 'fluid' traffic arrivals, the overflow probability at node 7 is very small. In this regard, Figure 17(a) implies that removing

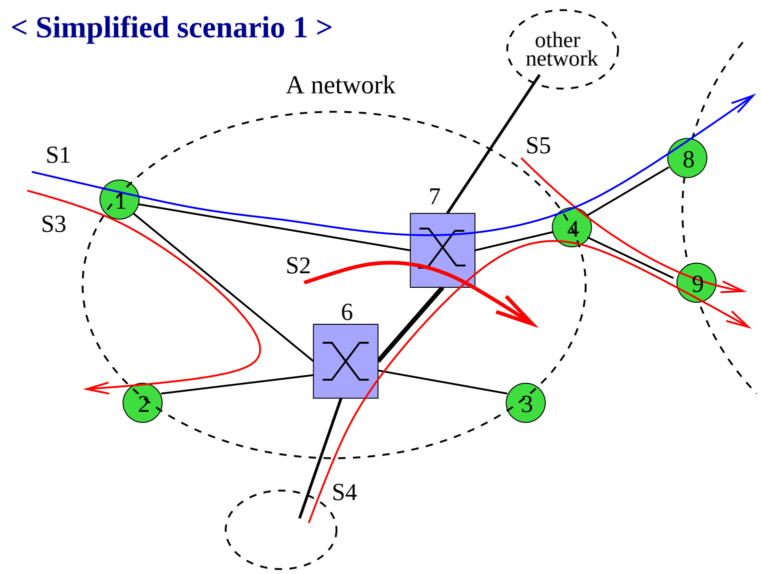
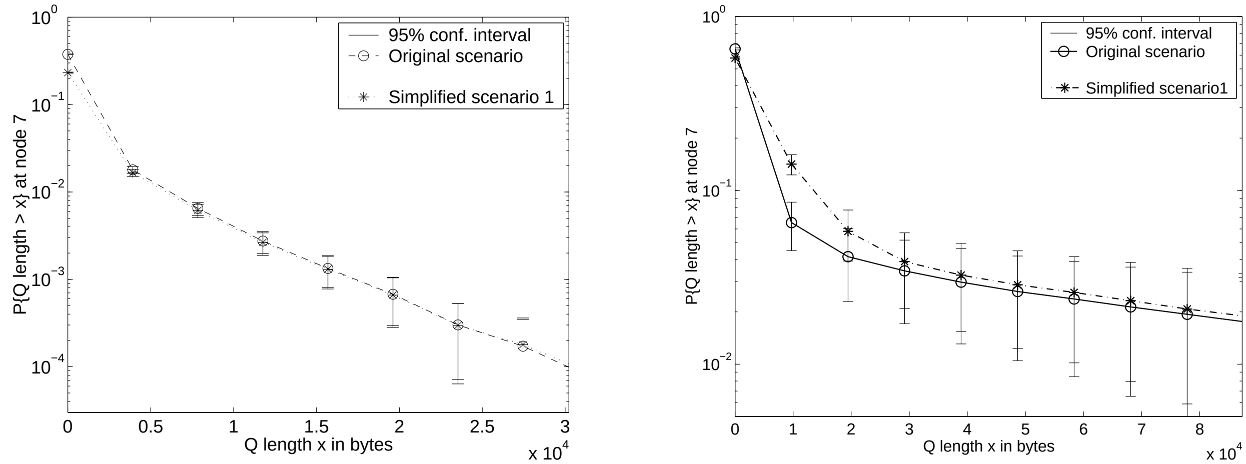


Figure 16: Simplified version of Figure 15 after removing node 5. We are interested in the overflow probability at node 7

the upstream queue with large capacity works well even for estimating small overflow probability at the downstream queue. These observations can also be supported by the recent theoretical result [32, 33], in which an overflow asymptotic was obtained using fluid models in a network of small buffers with buffer size $o(N)$, whereas the number of flows and the capacity linearly increases with N. Specifically, when all the sources require the same QoS, the authors in [32, 33] showed that asymptotically the admissible region corresponds to that which is obtained by assuming that flows pass through each node unchanged. Now, when S2 corresponds to 6 multiplexed MPEG video traffic flows, the overflow probability at node 7 is quite large, as we see in Figure 17(b), due to the burstiness of the video traffic. In this case, even though the capacity at node 5 is large, the number of multiplexed flows at the node is small, and the load at node 5 is quite high (approximately 0.76, from the simulation), we can still remove node 5 to estimate the overflow probability at the downstream node without incurring much error.

We will next remove multiple nodes from the original network and then monitor the queueing behavior at node 8 and the end-to-end delay for flow S1 (Ethernet traffic). Figure 18 shows two different simplified scenarios we now consider. First, we remove nodes 5, 6 and 7 that have a much larger capacity than the others, to obtain the simplified scenario 2. We further remove every node except the last one for flow S1 to obtain the simplified scenario 3. Since we are interested in flow S1, flow S2 acts as a multitude of interfering traffic flows inside the network at nodes 5 and 7, and flow S3, S4 and S5 as the cross traffic (or interfering traffic) at nodes 1 and 4.

Figures 19 and 20 show the overflow probability at node 8 and the end-to-end delay distribution for flow S1 when the cross traffic S2 corresponds to 6 MPEG video traffic flows. In both figures, the



(a) S2 is 100 voice traffic

(b) S2 is 6 MPEG video traffic

Figure 17: Buffer overflow probability at node 7 for different scenarios

curves for simplified scenario 2 are almost identical to the original scenario cases. In other words, removing nodes with large capacity has no effect on the performance of the queue at node 8 and the end-to-end delay for flow S1. This is because flow S1 comprises only a very small portion (about 5%) of total traffic flows feeding nodes 5 and 7. If we further remove nodes 1 and 4 (edge nodes) and thus only node 8 remains, the queueing behavior at node 8 still remains largely unaffected, as can be seen in Figure 19. Obviously, since the capacity of nodes 1 or 4 is within the same order of magnitude as compared to that of node 8, we could say that removing these nodes might affect the traffic flow S1. However, due to the multiplexing at node 1 and 4 with other cross traffic flows, the resulting QoS measures for flow S1 and at node 8 are almost the same as those of the original scenario.

4.4 Overall observations

Through the simulation results, we have observed the following: (1) For those nodes with capacity large enough to serve a large number of flows (without a few flows dominating), decomposing the network by removing these nodes will not affect the overflow probability at other nodes in the network (nor the end-to-end delay distribution) regardless of the traffic characteristics. (2) Now consider the case when we decompose the network by removing nodes that can multiplex only a small number of flows (e.g., less than 50 flows) and/or have very high utilization. In this case, other factors, such as burstiness of the traffic flows and the interfering traffic characteristics, come into play in determining the size of the error caused by the decomposition. (3) When even a small to moderate number of flows are multiplexed at a node, the flows routed to the different downstream nodes are not significantly altered, as long as each routed set of flows does not constitute a large fraction of the multiplexed flows. This idea enables us to analyze the performance of any particular

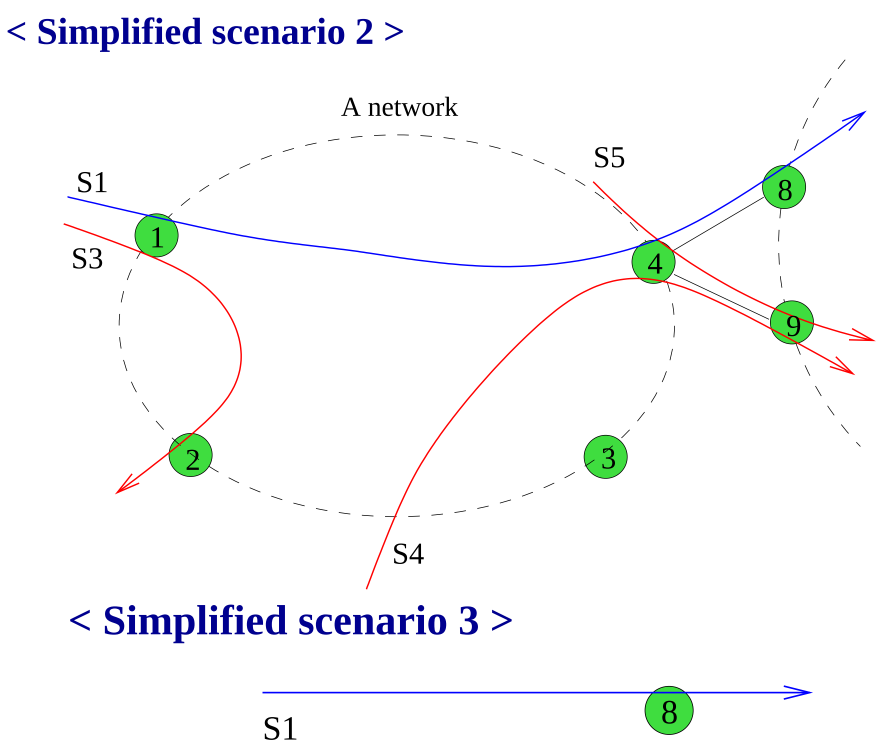


Figure 18: Simplified scenario 2 and Simplified scenario 3 of Figure 15 after removing multiple nodes

queue in the network by decomposing the entire network into a single queue, as was done for node 3 in the simplified scenario shown in Figure 13 or node 8 in the simplified scenario 3 in Figure 18.

5 Discussion

We have developed the network decomposition framework and witnessed how well it performs in different network scenarios. In this section, we investigate the behavior of the upstream queue itself where a large number of flows are multiplexed, and discuss our decomposing approach in a broader context.

As noted earlier, the fact that $I(0) > 0$ ensures the almost sure convergence of $q^N(t)$ to zero when the number of sources and the capacity are increased proportionally. We have found that $I(0)$ is always positive for discrete time models, and also for continuous time models, if there exists a peak rate or a well-behaved instantaneous traffic rate[††]. Although the positiveness of $I(0)$ means that $q^N(t)$ decreases to zero as N increases, we also observed from our simulations that network decomposition works well even if $q^N(t)$ is not small. For example, Figure 8 implies that the "empirical" speed of convergence of $q^N(t)$ to zero could be much slower than the convergence

[††]This can be visualized by a "fluid" type of traffic process.

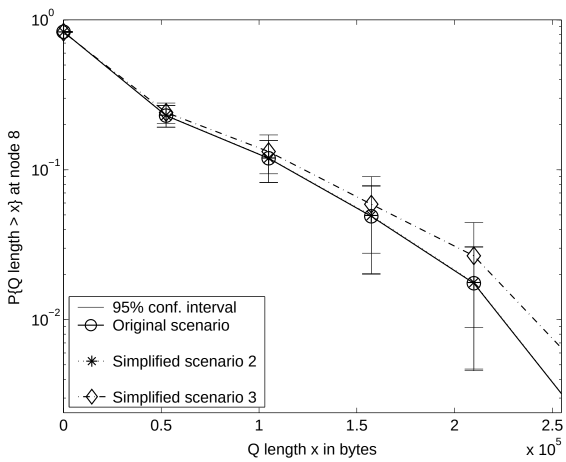


Figure 19: Buffer overflow probability at node 8 for different scenarios. S2 is 6 multiplexed MPEG video traffic.

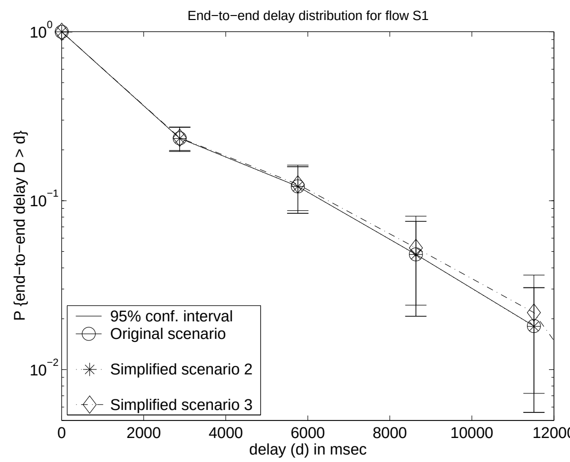


Figure 20: End-to-end delay distribution of flow S1 for different scenarios. S2 is 6 multiplexed MPEG video traffic.

of the queue-length at the downstream queue to that of the simplified scenario. Further, the decomposition method is still valid even if $\mathbb{P}\{q^N(t) > 0\}$ does not decrease at all, i.e., $I(0) = 0$, as we have seen for point process inputs in Section 3.6 and in Figure 11.

Apart from the many-sources-asymptotic scaling used so far, there have also been other ways of scaling large systems in the literature. For example, we can assume that the aggregate input process is well approximated by a Gaussian process based on the Central Limit Theorem type of arguments. This approach has been successful in estimating the overflow probability where a moderate to large number of traffic flows are multiplexed (see [34]). Likewise, we may also employ a moderate deviation scale to model a large system, which basically falls between the large deviation scale and the central-limit scale [35]. From a theoretical point of view, all of these different ways of scalings and different models (e.g., fluid or point processes) produce distinct limiting results. Hence, when applied to a given large system in practice, it is not clear which way of modeling and scaling the system would be the best (a heuristic explanation on this problem has been made in [35]).

However, there is in fact one thing in common. By aggregating independent traffic flows, we usually expect a certain level of statistical multiplexing gain, i.e., the resulting traffic becomes relatively smoother (or not burstier). Putting this in another way, we note that the aggregate queue-length ($q^N(t)$) will not blow up as the number of flows (N) increases for every occasion we have encountered. Instead, the queue-length becomes smaller as we showed in fluid models, or converges to a steady-state random variable for point process inputs as in the Poisson-limit case [36]. Similarly, under the central-limit scaling, the aggregate queue-length will be a well-defined stable one since for any given large capacity and number of flows, the resulting model consists of a Gaussian process input and a service capacity greater than the mean arrival rate.

Hence, from the above observations, we are led to believe that our decomposition approach will be good especially for the case when each set of flows is routed to different nodes in the network, i.e., no single bursty flow dominates at any node. To illustrate, recall that $q_i^N(t)$ denotes the amount of backlog in the queue ($q^N(t)$) due to the flow i. Then, it is unlikely that $q_i^N(t)$ is much larger than other $q_j^N(t)$ $(j \neq i)$ over a long period of time, unless the flow i dominates the traffic in the queue. Hence, the amount of backlog for each flow (or for a small set of flows) will become smaller and smaller as N increases, making the traffic departures behave more like the arrivals. This is in fact the key ingredient in proving Theorem 4 and is also in accordance with our findings in simulations via real Ethernet traffic traces, MPEG video traces, etc.

6 Conclusion

In this paper we have described the network decomposition approach, in which network analysis is made simple by eliminating other nodes. We have proved that the queue-length at the downstream queue converges to that of a single queue obtained by removing the upstream queue, as the capacity and the number of flows at the upstream queue become large. The modes of convergence vary with different input assumptions, from regulated arrivals to point processes. In all cases, however, the results say that the overflow probability at the downstream queue can be approximated by that of the decomposed system. Our results thus help simplify or analyze a large network by decomposing the original network into a simplified network in which all the nodes with large capacity have been eliminated. Through an extensive numerical investigation, we demonstrate several aspects and implications of the network decomposition approach. Our techniques perform well especially for the cases when (i) many flows are multiplexed and/or (ii) flows are routed to different nodes, i.e., no single flow dominates at any node. We believe our analysis and numerical results shed insight into the understanding of the queueing behavior in a network, where a large number of flows are aggregated at various nodes.

References

[1] D. Y. Eun and N. B. Shroff, "Simplification of Network Analysis in Large-Bandwidth Systems," in *Proceedings of IEEE INFOCOM*, San Francisco, CA, 2003.

[2] J. R. Jackson, "Networks of Waiting Lines," *Operations Research*, vol. 5, pp. 518–521, 1957.

[3] J. F. C. Kingman, "Markov Population Processes," *Journal of Applied Probability*, vol. 6, pp. 1–18, 1969.

[4] G. De Veciana, C. Courcoubetis, and J. Walrand, "Decoupling bandwidths: A decomposition approach to resource management in networks," in *Proceedings of IEEE INFOCOM*, 1994.

[5] Damon Wischik, "The output of a switch, or, effective bandwidths for networks," *Queueing Systems*, vol. 32, pp. 383–396, 1999.

[6] Damon Wischik, "Sample path large deviations for queues with many inputs," *Annals of Applied Probability*, vol. 11, pp. 379–404, 2000.

[7] R. M. Loynes, "The Stability of a Queue with Non-independent Inter-arrival and Service Times," *Proc. Cambridge Philos. Soc.*, vol. 58, pp. 497–520, 1962.

[8] N. Likhanov and R. Mazumdar, "Cell loss asymptotics for buffers fed with a large number of independent stationary sources," *Journal of Applied Probability*, vol. 36, no. 1, pp. 86–96, 1999.

[9] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications*, Springer-Verlag, 1998, 2nd Edition.

[10] J. A. Bucklew, *Large Deviation Techniques in Decision, Simulation, and Estimation*, John Wiley & Son, New York, 1990.

[11] A. Shwartz and A. Weiss, *Large Deviations for Performance Analysis: Queues, Communication and Computing*, CRC Press, 1995.

[12] D. D. Botvich and N. Duffield, "Large deviations, the shape of the loss curve, and economies of scale in large multiplexers," *Queueing Systems*, vol. 20, pp. 293–320, 1995.

[13] C. Courcoubetis and R. Weber, "Buffer overflow asymptotics for a buffer handling many traffic sources," *Journal of Applied Probability*, vol. 33, pp. 886–903, 1996.

[14] C. Courcoubetis, V. A. Siris, and G. D. Stamoulis, "Application of the many sources asymptotic and effective bandwidths to traffic engineering," *Telecommunication Systems*, vol. 12, pp. 167–191, 1999.

[15] D. Y. Eun and N. B. Shroff, "Network Decomposition in the Many-Sources Regime," Tech. Rep., Purdue University, West Lafayette, IN, 2002, (available at "http://yara.ecn.purdue.edu/~eun/doc/treport02-fluid.pdf").

[16] D. Y. Eun and N. B. Shroff, "Analyzing a two-stage queueing system with many Point Process arrivals at upstream queue," Tech. Rep., Purdue University, West Lafayette, IN, 2002, (available at "http://yara.ecn.purdue.edu/~eun/doc/treport02-point.pdf").

[17] C-S. Chang, "Stability, Queue Length, and Delay of Deterministic and Stochastic Queueing Networks," *IEEE Transactions on Automatic Control*, vol. 39, pp. 913–931, May 1994.

[18] R. L. Cruz, "A Calculus for Network Delay, Part I : Network Elements in Isolation," *IEEE Transactions on Information Theory*, vol. 37, pp. 114–131, Jan. 1991.

[19] R. L. Cruz, "A Calculus for Network Delay, Part II : Network Analysis," *IEEE Transactions on Information Theory*, vol. 37, pp. 132–142, Jan. 1991.

[20] R. Durrett, *Probability : Theory and Examples*, Duxbury Press, Belmont, CA, second edition, 1996.

[21] W. E. Leland, M.S. Taqqu, W. Willinger, and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1–15, Feb. 1994.

[22] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-range dependence in variable-bit-rate video traffic," *IEEE Transactions on Communications*, vol. 43, no. 2–4, pp. 1566–1579, Feb.–April 1995.

[23] J. Choe and N. B. Shroff, "Use of Supremum Distribution of Gaussian Processes in Queueing Analysis with Long-Range Dependence and Self-Similarity," *Stochastic Models*, vol. 16, no. 2, Feb. 2000.

[24] M. Mandjes and J. H. Kim, "Large deviations for small buffers: an insensitivity result," *Queueing Systems*, vol. 37, pp. 349–362, 2001.

[25] A. K. Parekh and R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 344–357, June 1993.

[26] D. Y. Eun and N. B. Shroff, "A Measurement-Analytic Framework for QoS Estimation Based on the Dominant Time Scale," in *Proceedings of IEEE INFOCOM*, Anchorage, AK, 2001.

[27] D. Y. Eun and N. B. Shroff, "A Measurement-Analytic Approach for QoS Estimation in a Network based on the Dominant Time Scale," *IEEE/ACM Transactions on Networking*, 2003, to appear.

[28] J. Roberts, U. Mocci, and J. Virtamo, *Broadband Network Teletraffic, Final Report of Action COST 242*, Springer-Verlag, New York, 1996.

[29] J. Cao and K. Ramanan, "A Poisson Limit for the Unfinished Work of Superposed Point Processes," *Bell Labs Tech. Report*, 2001.

[30] D. J. Daley and D. Vere-Jones, *An Introduction to the Theory of Point Processes*, Springer-Verlag, 1988.

[31] I. W. Habib and T. N. Saadawi, "Multimedia Traffic Characteristics in Broadband Networks," *IEEE Commun. Mag.*, pp. 48–54, July 1992.

[32] O. Ozturk, R. Mazumdar, and N. Likhanov, "Many sources asymptotics for a feedforward network with small buffers," in *Proceedings of the Allerton Conference*, Monticello, IL, 2002.

[33] O. Ozturk, R. Mazumdar, and N. Likhanov, "Many sources asymptotics in networks with small buffers," *Queueing Systems*, to appear.

[34] J. Choe and N. B. Shroff, "A Central Limit Theorem Based Approach for Analyzing Queue Behavior in High-Speed Networks," *IEEE/ACM Transactions on Networking*, vol. 6, no. 5, pp. 659–671, Oct. 1998.

[35] Damon Wischik, "Moderate deviations in queueing theory," Feb. 2001, Draft.

[36] J. Cao and K. Ramanan, "A Poisson Limit for Buffer Overflow Probabilities," in *Proceedings of IEEE INFOCOM*, 2002.