ARTICLE IN PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT

Performance Evaluation ▮ (▮▮▮▮) ▮▮▮–▮▮▮

PERFORMANCE EVALUATION
An International Journal

www.elsevier.com/locate/peva

Path-wise performance in a tree-type network: Per-stream loss probability, delay, and delay variance analyses☆

Huei-Wen Ferng*, Chi-Chao Chao, Cheng-Ching Peng

Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 106, Taiwan

Received 4 November 2002; received in revised form 16 January 2005

Abstract

This paper deals with path-wise performance analysis rather than a nodal one to enrich results previously obtained in the literature under simple but unsatisfactory assumptions, e.g., Poisson processes. First deriving the per-stream loss probability, delay, and delay variance of an isolated queue with multi-class input streams modeled by heterogeneous two-state Markov-modulated Poisson processes (MMPPs), we then propose simple and novel decomposition schemes working together with an input parameter modification scheme to (approximately) extract the per-stream output process for a lossy queue receiving MMPPs under a general service time distribution. The novelty of the decompositions is that they can be easily implemented based on a lossless queueing model. Through numerical experiments, we show that the accuracy in estimating the per-stream output process using such schemes is good. These decomposition schemes together with the input parameter modification scheme and a moment-based fitting algorithm used to fit the per-stream output as a two-state MMPP make analysis of path-wise performance viable by virtually treating each node in isolation along a path to get performance measures sequentially from the source node en route to the destination node.

Keywords: Performance analysis; Decomposition scheme; Markov-modulated Poisson process

1. Introduction

Current networks proliferate quickly as compared to those a few decades ago. This can be reflected from the following aspects. Firstly, network size: it grows explosively as time goes by. Secondly, service type: diverse types of services are offered to satisfy the needs of different users, e.g., constant bit rate (CBR), variable bit rate (VBR), available bit rate (ABR), and unspecified bit rate (UBR) services provided by the asynchronous transfer mode (ATM) network [27]. Thirdly, traffic pattern: traffic spectra range from voice, data, to image or video etc. Hence, bursty and correlative characteristics have been reported among these types of traffic. All the aforementioned changes make queueing networks employing simple traffic assumptions (e.g., Poisson arrival) and service mechanisms (e.g.,

☆ This work was supported in part by the National Science Council, Taiwan, R.O.C. under Contracts NSC 90-2213-E-011-096, NSC 93-2219-E-011-007, and NSC 94-2219-E-011-006.

* Corresponding author. Fax: +886 2 2730 1081.
E-mail address: hwferng@mail.ntust.edu.tw (H.-W. Ferng).

0166-5316/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.peva.2006.01.003

exponential server) in the past, such as Jackson network [5,27] unsatisfactory when applying these models to current and future networks. Therefore, the following modifications/enhancements for models are required at least: (i) input model: the renewal process, such as Poisson should be migrated to the non-renewal process to properly capture the bursty and correlative nature of multimedia traffic. In the literature, many point processes have been proposed and investigated, for example, Markov-modulated Poisson process (MMPP) [9], Markovian arrival process (MAP) [19], and batch Markovian arrival process (BMAP) [20] etc. Among these models, MMPPs are the most popular and frequently used to model multimedia traffic because of its simplicity and mathematical tractability [6,9,12]. (ii) Service time distribution: more general service time distributions should be incorporated to accommodate different service mechanisms. (iii) System capacity: finite capacity is a realistic assumption for real networks. For the above reasons, we adopt two-state MMPPs to model the external traffic, general service time distributions, and finite buffers in the considered queueing networks. In addition, heterogeneous multi-class input streams are taken into consideration to appropriately reflect per-path behavior in a network.

The queueing models to be addressed in this paper fall into the scope of finite-capacity tree-type networks. In the literature, tandem networks with blocking, which are a subset of the finite-capacity tree-type networks, have been studied since the 1960s, for example, [3] and [29]. These two papers employed a Poisson process as the input to the first node. Recently, Gomez-Corral [10] analyzed such queueing networks but employed the non-renewal MAP as the input to the first node. Klimenok et al. [17] also analyzed the two-stage $\mathrm{BMAP/G/1/\tilde{N}} \rightarrow \cdot/\mathrm{PH}/1/\mathrm{M}$ tandem queue. This clearly manifests one of the current tendencies: to employ non-renewal processes in queueing networks. Note that [10] and [17] both provided the exact mathematical results. In [10], no intermediate buffers are assumed except the first node with an infinite buffer, while finite buffers are assumed in [17]. Such models have many applications in many fields, but they do not satisfy the needs of current computer and communication networks. Many papers have addressed the end-to-end performance issue in the literature for the environment of computer and communication networks. In [2], Addie and Zukerman studied the loss probability of a tree-type ATM network using discrete-time Gaussian processes as input processes. Heindl [12] analyzed the mean queue length for the tandem queue with an MMPP to the first node using a decomposition method. Kim and Shroff [15] developed a new notation called the end-to-end capacity in terms of input and output processes of an end-to-end path to estimate end-to-end network delays. Kroner et al. [18] investigated the end-to-end delay/jitter for the ATM network using a tandem queue with burst silence sources. In [21], Mitchell and van de Liefvoort used the linear algebra queueing theory (LAQT) to create reduced space representations for queue departure processes and got approximation models for feed-forward G/G/1/N queueing networks employing suitable moment matching. In [23], Naser and Leon-Garcia used the simulation approach to study the delay and delay variation of CBR traffic in ATM networks. Nyberg et al. [24] proposed an algorithm based on the concept of convolution to calculate the end-to-end probability for multicast networks. Ren et al. [26] also employed a tandem queue with on-off tagged and cross traffic sources to study the mean queue length for each node in ATM networks. In [28], Sohraby and Privalov studied the end-to-end jitter of individual flows for the connection-oriented feed-forward networks using periodic flows.

Queueing networks studied in the literature can be categorized into continuous-time models [10] and discrete-time models [22,25]. In [22], Modiano and Wieselthier analyzed the queueing delay of a tree-type network using discrete-time $\cdot/\mathrm{D}/1$ to model each node. Park et al. [25] studied the loss probability of a discrete-time tandem queue with a Markov-modulated Bernoulli process (MMBP) at the first node. As for input models, fluid models are also frequently employed to study network-wise performance besides Markovian input streams, e.g., [1] and [14]. Aalto and Scheinhardt [1] studied tandem fluid queues fed by homogeneous on-off sources. In [14], Kella studied the Markov-modulated feed-forward fluid networks. In this paper, we investigate continuous-time tree-type queueing networks with Markovian external input models, i.e., MMPPs. This paper can be treated as the extension of [6] in which infinite buffers are assumed for the sake of simplicity and only the mean delay at each node and mean delay for each virtual/reference connection are acquired. Unlike [6], a finite buffer is endowed with each queue receiving external two-state MMPPs under general service time distributions in this paper. With the above assumption, we obtain per-stream loss probability, delay, and delay variance at each node and these measures for an individual path. Hence, this paper employs a more realistic queueing model and provides more performance measures than [6]. After deriving per-stream performance measures for an isolated queue, we propose two new decomposition schemes based on a lossless queue and an input parameter modification scheme which works together with the decomposition scheme to extract the per-stream output process of the original lossy queue. In the literature, [12] is also close to our paper. However, our paper differs from [12] at least in the inclusion of heterogeneous MMPPs rather than a single MMPP only.

The rest of this paper is organized as follows. Section 2 presents traffic and system models. In Section 3, the per-stream performance for an isolated queue is derived. Two decomposition schemes and an input parameter modification scheme are given in Section 4. Applying the results in Sections 3 and 4 enables us to get the performance of an individual path in a tree-type network using a systematic approach described in Section 5. In Section 6, the accuracy and limitation of our approach are investigated through numerical experiments. Finally, Section 7 concludes the paper.

2. Model description

2.1. Traffic model

In this paper, each external input stream is modeled by a two-state MMPP [9], which is governed by an infinitesimal generator matrix $\boldsymbol{Q}_{\mathrm{i}}$ and a rate matrix $\boldsymbol{\Lambda}_{\mathrm{i}}$ with the following forms:

$$\boldsymbol{Q}_{\mathrm{i}} = \begin{bmatrix} -\sigma_{1i} & \sigma_{1i} \\ \sigma_{2i} & -\sigma_{2i} \end{bmatrix} \tag{1}$$

and

$$\boldsymbol{\Lambda}_{\mathrm{i}} = \begin{bmatrix} \lambda_{1i} & 0 \\ 0 & \lambda_{2i} \end{bmatrix}, \tag{2}$$

where i serves as an index, say the ith MMPP, σ_{1i} (σ_{2i}) represents the transition rate from state 1 (state 2) to state 2 (state 1), and λ_{1i} (λ_{2i}) is the input rate of a Poisson process when the current state of the MMPP is state 1 (state 2). The reason why we adopt such a traffic model to describe each input stream is mainly due to the fact that MMPPs lead to tractable analytic models and have been popularly employed to model multimedia traffic, especially the data traffic in high-speed networks [6,9,12]. Note that MMPPs are suitable to model streaming traffic under open-loop flow control [30], such as leaky bucket (e.g., CBR or VBR connections in ATM networks) or without any flow control (e.g., UDP flows in IP networks). However, such models cannot be used for elastic traffic under closed-loop flow control [30] (e.g., ABR connections in ATM networks or TCP flows in IP networks). MMPPs also have a nice property: r independent MMPPs result in another MMPP with the infinitesimal generator matrix $\boldsymbol{Q}$ and rate matrix $\boldsymbol{\Lambda}$ described by the following two relations (see [9] for details):

$$\boldsymbol{Q} = \boldsymbol{Q}_1 \oplus \boldsymbol{Q}_2 \oplus \cdots \oplus \boldsymbol{Q}_r \tag{3}$$

and

$$\boldsymbol{\Lambda} = \boldsymbol{\Lambda}_1 \oplus \boldsymbol{\Lambda}_2 \oplus \cdots \oplus \boldsymbol{\Lambda}_r, \tag{4}$$

where the operator $\oplus$ is the so-called Kronecker sum [9]. Such an aggregated MMPP has the number of states $m = \prod_{i=1}^{r} m_i$ with m_i representing the number of states for the ith MMPP.

2.2. System model

The queueing systems to be addressed in the following can be categorized into two types: (i) an isolated finite-capacity (K is used to denote the capacity in the following) queue receiving r heterogeneous two-state MMPPs which form an aggregated MMPP with $m = 2^r$ states and (ii) a tree-type network in which each node is endowed with a finite buffer and external two-state MMPPs. For both systems, the service times for a specific server are independent and identically distributed (i.i.d.) and have the cumulative distribution function (CDF) $\tilde{H}(x)$ with finite mean h and the corresponding Laplace–Stieltjes transform (LST) $H(s)$. The service discipline employed by each server is the first-in-first-out (FIFO) discipline. In the tree-type networks, heterogeneous servers may be employed and only fixed/permanent routing protocols are taken into consideration. Hence, the results provided in this paper are more suitable for connection-oriented networks [27].

3. Performance analysis for an isolated node

Based on the traffic and system assumptions mentioned above, it is hard to get an exact solution for the per-path performance in a tree-type network. Therefore, we seek an approximate solution rather than pursuing an exact one by using the so-called Kleinrock independence approximation (KIA) [6,16] in this paper. Applying KIA enables one to regard the whole network as an ensemble of isolated nodes with proper decomposition. The approach employed in this paper is sketched as follows: first analyze the performance of the isolated node; then get the per-stream output process and approximate the per-stream output to a two-state MMPP via a moment-based fitting algorithm; finally perform the above procedure successively to get the performance of an individual path. Therefore, performance analysis for an isolated queue becomes the first step when analyzing the path-wise performance in the tree-type network. In the following, we derive loss probability and delay-related performance for overall and per-stream traffic.

3.1. Preliminaries

Let us consider the isolated queue described in Section 2.2. First denoting $\{\tau_n : n \geq 0\}$ to be the successive departure epochs with $\tau_0 \equiv 0$ and defining X_n and J_n to be the queue length and the state of the arrival process at time τ_n^+, it can be easily seen that $\{(X_n, J_n, \tau_{n+1} - \tau_n) : n \geq 0\}$ forms a semi-Markov sequence at departure epochs with state-space $\{0, 1, \ldots, K-1\} \times \{1, \ldots, m\}$ and the state transition probability matrix

$$\tilde{\boldsymbol{Q}}^*(x) = \begin{bmatrix} \tilde{\boldsymbol{B}}_0(x) & \tilde{\boldsymbol{B}}_1(x) & \ldots & \tilde{\boldsymbol{B}}_{K-2}(x) & \sum_{n=K-1}^{\infty} \tilde{\boldsymbol{B}}_n(x) \\ \tilde{\boldsymbol{A}}_0(x) & \tilde{\boldsymbol{A}}_1(x) & \ldots & \tilde{\boldsymbol{A}}_{K-2}(x) & \sum_{n=K-1}^{\infty} \tilde{\boldsymbol{A}}_n(x) \\ \boldsymbol{0} & \tilde{\boldsymbol{A}}_0(x) & \ldots & \tilde{\boldsymbol{A}}_{K-3}(x) & \sum_{n=K-2}^{\infty} \tilde{\boldsymbol{A}}_n(x) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \ldots & \tilde{\boldsymbol{A}}_0(x) & \sum_{n=1}^{\infty} \tilde{\boldsymbol{A}}_n(x) \end{bmatrix}, \quad K \geq 1, \tag{5}$$

where $\tilde{\boldsymbol{A}}_n(x)$ and $\tilde{\boldsymbol{B}}_n(x)$, $n \geq 0$, $x \geq 0$, are defined as follows [9]:

$$\tilde{\boldsymbol{A}}_n(x) = \int_0^x \boldsymbol{P}(n, u)\, \mathrm{d}\tilde{H}(u), \quad n \geq 0, x \geq 0, \tag{6}$$

$$\tilde{\boldsymbol{B}}_n(x) = \tilde{\boldsymbol{U}}(x) * \tilde{\boldsymbol{A}}_n(x), \quad n \geq 0, x \geq 0, \tag{7}$$

and

$$\tilde{\boldsymbol{U}}(x) = \int_0^x \mathrm{e}^{(\boldsymbol{Q}-\boldsymbol{\Lambda})t} \boldsymbol{\Lambda}\, \mathrm{d}t, \tag{8}$$

where $\boldsymbol{P}(n, u)$ is an $m \times m$ matrix with (i, j)th entry representing the probability that there are n customers arriving during a period of length u and the state of the arrival process is j at time u given the state of the arrival process is i at time 0 and the operator $*$ denotes matrix convolution. We shall use $\boldsymbol{Q}^*(s)$, $\boldsymbol{A}_n(s)$, $\boldsymbol{B}_n(s)$, and $\boldsymbol{U}(s)$ to denote LSTs of $\tilde{\boldsymbol{Q}}^*(x)$, $\tilde{\boldsymbol{A}}_n(x)$, $\tilde{\boldsymbol{B}}_n(x)$, and $\tilde{\boldsymbol{U}}(x)$, respectively. For brevity, we also use the shorthand $\boldsymbol{Q}^* = \boldsymbol{Q}^*(0)$, $\boldsymbol{A}_n = \boldsymbol{A}_n(0)$, $\boldsymbol{B}_n = \boldsymbol{B}_n(0)$, $\boldsymbol{U} = \boldsymbol{U}(0)$, and shorthand for derivatives of the above four quantities.

The queue length distribution at departure epochs can be obtained from the stationary vector $\boldsymbol{x}$ which can be solved via $\boldsymbol{x}\boldsymbol{Q}^* = \boldsymbol{x}$ and $\boldsymbol{x}\hat{\boldsymbol{e}} = 1$, where $\hat{\boldsymbol{e}}$ is an $mK \times 1$ column vector of all ones and can be partitioned into the form $\boldsymbol{x} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{K-1})$, where $\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,j}, \ldots, x_{i,m})$ with $x_{i,j} = \Pr\{\text{there are } i \text{ customers in the system right behind a departure epoch and the arrival process is in state } j\}$.

3.2. Overall and per-stream loss probabilities

The queue length distribution at an arbitrary instant for an MMPP/G/1/K queue, i.e., $\boldsymbol{y}_i$, $0 \le i \le K$, of a $1 \times m$ row vector has been acquired in [4], where Blondia conducted a more general N/G/1/K queue analysis, and the results are given as follows:

$$\begin{cases} \boldsymbol{y}_0 = \dfrac{-\boldsymbol{x}_0 \boldsymbol{R}^{-1}}{h - \boldsymbol{x}_0 \boldsymbol{R}^{-1} \boldsymbol{e}}, \\ \boldsymbol{y}_n = \dfrac{\boldsymbol{x}_0 \left[\boldsymbol{R}_{n-1} - \boldsymbol{U}\boldsymbol{R}_{n-1}\right] + \sum_{k=1}^{n-1} \boldsymbol{x}_k [\boldsymbol{R}_{n-k-1} - \boldsymbol{R}_{n-k}] - \boldsymbol{x}_n \boldsymbol{R}_0}{h - \boldsymbol{x}_0 \boldsymbol{R}^{-1} \boldsymbol{e}}, \quad 1 \le n \le K-1, \\ \boldsymbol{y}_K = \boldsymbol{\theta} - \sum_{n=0}^{K-1} \boldsymbol{y}_n, \end{cases} \tag{9}$$

where $\boldsymbol{e}$ is an $m \times 1$ column vector of all ones, $\boldsymbol{R} = \boldsymbol{Q} - \boldsymbol{\Lambda}$, $\boldsymbol{U} = (\boldsymbol{\Lambda} - \boldsymbol{Q})^{-1}\boldsymbol{\Lambda}$, $\boldsymbol{R}_k = \boldsymbol{R}_k(0)$ with

$$\boldsymbol{R}_k(s) = (s\boldsymbol{I} + \boldsymbol{R})^{-1}[-\boldsymbol{\Lambda}(s\boldsymbol{I} + \boldsymbol{R})^{-1}]^k, \tag{10}$$

and $\boldsymbol{\theta}$ is the stationary vector of the underlying Markov chain and it can be obtained via solving the following equations:

$$\boldsymbol{\theta}\boldsymbol{Q} = \boldsymbol{0}, \quad \boldsymbol{\theta}\boldsymbol{e} = 1, \tag{11}$$

where $\boldsymbol{0}$ is a $1 \times m$ zero vector. Utilizing the queue length distribution at an arbitrary instant enables us to derive the overall loss probability P_L and per-stream loss probability $P_{L,i}$ for an MMPP/G/1/K queue with r heterogeneous two-state MMPPs. These results are shown as follows:

$$P_L = \frac{\boldsymbol{y}_K \boldsymbol{\Lambda} \boldsymbol{e}}{\sum_{k=0}^{K} \boldsymbol{y}_k \boldsymbol{\Lambda} \boldsymbol{e}}, \tag{12}$$

$$P_{L,i} = \frac{\boldsymbol{y}_K \boldsymbol{\Lambda}(i) \boldsymbol{e}}{\sum_{k=0}^{K} \boldsymbol{y}_k \boldsymbol{\Lambda}(i) \boldsymbol{e}}, \tag{13}$$

where

$$\boldsymbol{\Lambda}(i) = \overbrace{\boldsymbol{0} \oplus \boldsymbol{0} \oplus \cdots \oplus \boldsymbol{0}}^{i-1\ \boldsymbol{0}'s} \oplus \boldsymbol{\Lambda}_i \oplus \overbrace{\boldsymbol{0} \oplus \cdots \oplus \boldsymbol{0} \oplus \boldsymbol{0}}^{r-i\ \boldsymbol{0}'s} \tag{14}$$

and $\boldsymbol{0}$ is a 2×2 zero matrix.

3.3. Overall and per-stream delay moments

To derive moments of overall and per-stream delays, we first need to figure out the $1 \times m$ row vector $\boldsymbol{W}(s)$ with the jth component representing the LST of the stationary virtual waiting (delay) time when the state of the arrival process is j. The virtual waiting time at time t is the amount of time that a "virtual" customer arriving at time t needs to wait before getting his service in the server (see [4] for more details). For the MMPP/G/1/K queue, $\boldsymbol{W}(s)$ has the following form [4]:

$$\boldsymbol{W}(s) = \frac{1}{(1 - \boldsymbol{y}_K \boldsymbol{e})(h - \boldsymbol{x}_0 \boldsymbol{R}^{-1} \boldsymbol{e})} \left\{ -\boldsymbol{x}_0 \boldsymbol{R}^{-1} + \sum_{n=1}^{K-1} \left[\sum_{k=0}^{n-1} \boldsymbol{x}_k \boldsymbol{R}_{n-k-1}(s) H^{n-1}(s) \right.\right.$$
$$\left.\left. - \boldsymbol{x}_0 \boldsymbol{U} \boldsymbol{R}_{n-1}(s) H^n(s) - \sum_{k=1}^{n} \boldsymbol{x}_k \boldsymbol{R}_{n-k}(s) H^n(s) \right] \right\}. \tag{15}$$

Using $\boldsymbol{W}(s)$ enables us to derive the LSTs of the overall and per-stream waiting (delay) time at arrival instants as follows:

$$W_a(s) = \frac{\boldsymbol{W}(s)\boldsymbol{\Lambda}\boldsymbol{e}}{\sum_{n=0}^{K-1} \boldsymbol{y}_n \boldsymbol{\Lambda}\boldsymbol{e}}, \tag{16}$$

$$W_{a,i}(s) = \frac{\boldsymbol{W}(s)\boldsymbol{\Lambda}(i)\boldsymbol{e}}{\sum_{n=0}^{K-1} \boldsymbol{y}_n \boldsymbol{\Lambda}(i)\boldsymbol{e}}. \tag{17}$$

The above two equations lead to the kth moments of the overall and per-stream waiting (delay) time shown below:

$$W_a^{(k)} = (-1)^k \frac{\mathrm{d}^k}{\mathrm{d}s^k} W_a(s)|_{s=0} = (-1)^k \frac{\boldsymbol{W}^{(k)}\boldsymbol{\Lambda}\boldsymbol{e}}{\sum_{n=0}^{K-1} \boldsymbol{y}_n \boldsymbol{\Lambda}\boldsymbol{e}}, \tag{18}$$

$$W_{a,i}^{(k)} = (-1)^k \frac{\mathrm{d}^k}{\mathrm{d}s^k} W_{a,i}(s)|_{s=0} = (-1)^k \frac{\boldsymbol{W}^{(k)}\boldsymbol{\Lambda}(i)\boldsymbol{e}}{\sum_{n=0}^{K-1} \boldsymbol{y}_n \boldsymbol{\Lambda}(i)\boldsymbol{e}}, \tag{19}$$

where $\boldsymbol{W}^{(k)} = \frac{\mathrm{d}^k}{\mathrm{d}s^k}\boldsymbol{W}(s)|_{s=0}$. In Appendix A, we give the formulas for $\boldsymbol{W}^{(k)}$ when $k = 1$ and $k = 2$. Thus, the mean and variance of delays can be obtained from (18) and (19).

4. Decomposition schemes and the input parameter modification scheme

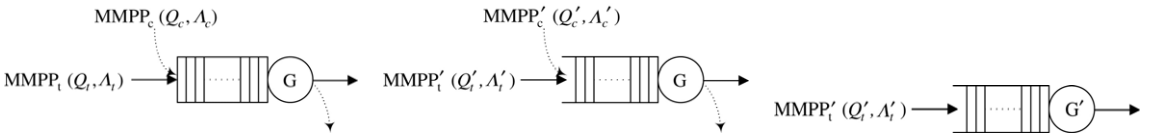
To apply the nodal performance obtained in Section 3 to path-wise performance, one viable way is to first find the per-stream output process; then one can treat the downstream node as an isolated node and sequentially obtain performance measures for each node along the path. Although He and Stanford [11] have studied the inter-departure time distributions of the MMAP/G/1 queues (here MMAP represents a Markov arrival process with marked transitions), the results obtained in [11] are confined to moments of inter-departure times. As for the covariance of two cross inter-departure times, it was not acquired there. Hence, [11] cannot satisfy the need of this paper. Instead, we propose simple decomposition schemes based on a lossless queue as well as an input parameter modification scheme to get per-stream output processes. These schemes are described as follows.

4.1. Decomposition schemes

For simplicity, decomposition schemes are used in this paper to extract the per-stream output process from the overall output process. In Appendix B, we give the overall output process of the MMPP/G/1 queue rendered from [7] in which the departure process characteristics of BMAP/G/1(/K) queues have been provided.

These decomposition schemes are developed based on queues with buffers of infinite size. The idea of the decomposition schemes is carried out by replacing the original queueing system receiving multiple input streams with a queue in which only the tagged stream is allowed to pass through the queue and the service time for this input stream should reflect the crossing/interfering effect coming from other streams in the original system (see Fig. 1(b)–(c)). In the following, the resultant queue and its server are called the *decomposed queue* and *effective server*, respectively.

Let us first characterize the service time distribution for the effective server using an approximate approach. Without loss of generality, we may regard the original queue as a queue with only two input streams, one tagged stream with traffic load ρ_t and the other aggregated cross stream with traffic load ρ_c. For the head of line (HOL) of the decomposed queue, it may directly go into the server with probability $1 - \rho_c$ because of no packets/customers of the cross stream ahead of it. However, it should first wait $i - 1$ ($i \geq 1$) full and one residual service time(s) for the cross stream and then gets its service with probability $\rho_c^i(1 - \rho_c)$. The above description leads to the following LST of the service time

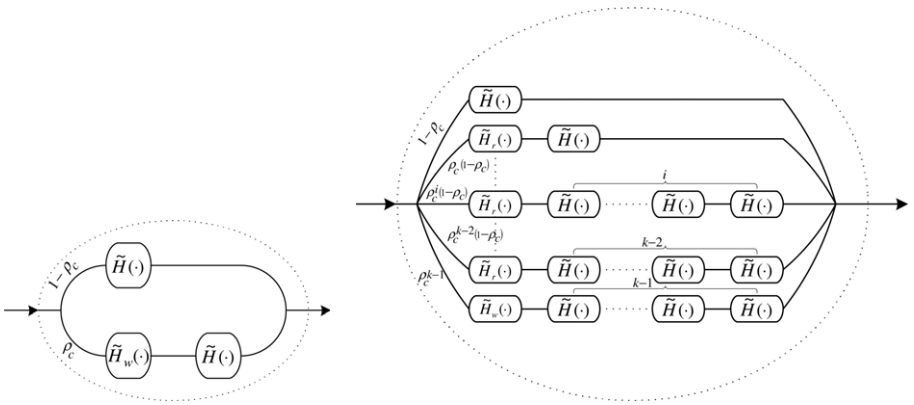


(a) Original finite-buffer queue.

(b) Effective infinite-buffer queue with input parameter modification.

(c) The decomposed infinite-buffer queue for the queue in (b).

Fig. 1. Input parameter modification scheme and decomposition scheme for the finite-buffer queue.



(a) 2-branch model.

(b) k-branch model.

Fig. 2. Effective servers.

distribution for the effective server

$$H_{\text{eff}}(s) = (1-\rho_c)H(s) + \sum_{i=1}^{\infty}(1-\rho_c)\rho_c^i H_r(s)H^{i-1}(s)H(s) \tag{20}$$

$$= (1-\rho_c)H(s) + \rho_c H(s)H_w(s), \tag{21}$$

where $H_r(s)$ is the LST of the residual service time and has the following form from renewal theory [16], while $H_w(s)$ represents the LST of the extra waiting time in the 2-branch effective server model (see Fig. 2(a)) before the HOL receives its service time with the form given below.

$$H_r(s) = \frac{1-H(s)}{sh_1}, \tag{22}$$

$$H_w(s) = (1-\rho_c)H_r(s)\frac{1}{1-\rho_c H(s)}. \tag{23}$$

In the above equation, we use the notation h_n to denote the nth moment of the service time (we note that $h_1 = h$), i.e.,

$$h_n = (-1)^n H^{(n)}(0). \tag{24}$$

For convenience, we also use r_n to denote the nth moment of the residual service time, that is

$$r_n = (-1)^n H_r^{(n)}(0) = \frac{h_{n+1}}{(n+1)h_1}. \tag{25}$$

The above relation for r_n has been acquired in [16]. Let $w_n = (-1)^n H_w^{(n)}(0)$ be the nth moment of the random variable of the extra waiting time when the system is busy for the cross traffic stream. From (22) and (23), we then

have

$$w_1 = \frac{h_2}{2h_1} + \frac{\rho_c h_1}{1-\rho_c}, \tag{26}$$

$$w_2 = \frac{h_3}{3h_1} + \frac{2\rho_c h_2}{1-\rho_c} + \frac{2\rho_c^2 h_1^2}{(1-\rho_c)^2}. \tag{27}$$

Therefore, we can obtain the variance $\sigma_w^2 = w_2 - w_1^2$ from the above two equations.

Instead of directly applying (21), we now further approximate it for ease of computation using results in Appendices C and D. Fitting $H_w(s)$ using the LST of a gamma distribution $\Gamma(x;\alpha_w,\mu_w)$ [13], we then have $\sigma_w^2 = \alpha_w/\mu_w^2$ and $w_1 = \alpha_w/\mu_w$ which yield

$$\mu_w = \frac{w_1}{\sigma_w^2} \quad \text{and} \quad \alpha_w = \frac{w_1^2}{\sigma_w^2}. \tag{28}$$

Using (21) and (28), we then have an approximate 2-branch effective server. We can further extend the 2-branch effective server model (see Fig. 2(a)) to a k-branch ($k \geq 3$) one (see Fig. 2(b)) as follows. Rewrite $H_{\text{eff}}(s)$ in the following form:

$$\begin{aligned} H_{\text{eff}}(s) &= (1-\rho_c)H(s) + \sum_{i=1}^{k-2}(1-\rho_c)\rho_c^i H_r(s)H^{i-1}(s)H(s) + \sum_{i=k-1}^{\infty}(1-\rho_c)\rho_c^i H_r(s)H^i(s) \\ &= (1-\rho_c)H(s) + \left[\sum_{i=1}^{k-2}(1-\rho_c)\rho_c^i H^{i-1}(s)\right] H_r(s)H(s) \\ &\quad + \rho_c^{k-1}(1-\rho_c)H_r(s)\frac{1}{1-\rho_c H(s)}H^{k-2}(s)H(s) \\ &= (1-\rho_c)H(s) + \left[\sum_{i=1}^{k-2}(1-\rho_c)\rho_c^i H^{i-1}(s)\right] H_r(s)H(s) + \rho_c^{k-1}H_w(s)H^{k-2}(s)H(s). \end{aligned} \tag{29}$$

Again, (28) can be utilized to approximate $H_w(s)$ like the 2-branch model. Similarly, $H_r(s)$ can be approximated using the LST of a gamma distribution $\Gamma(x;\alpha_r,\mu_r)$ with $\mu_r = r_1/\sigma_r^2$ and $\alpha_r = r_1^2/\sigma_r^2$, where $\sigma_r^2 = r_2 - r_1^2$. Fig. 2(b) shows the block diagram to implement the effective server with k branches using (29).

4.2. Input parameter modification scheme

The purpose of the input parameter modification scheme is to derive the equivalent input traffic parameters ($\boldsymbol{\Lambda}_i'$, $\boldsymbol{Q}_i'$) inbound to an infinite-buffer queue for the input traffic with parameters ($\boldsymbol{\Lambda}_i$, $\boldsymbol{Q}_i$) inbound to a finite-buffer queue (see Fig. 1(a)–(b)). Note that the salient difference between an infinite-buffer queue and a finite-buffer queue is the loss phenomenon of the finite-buffer queue. Hence, the loss effect of the finite-buffer queue should be considered when deriving ($\boldsymbol{\Lambda}_i'$, $\boldsymbol{Q}_i'$) from ($\boldsymbol{\Lambda}_i$, $\boldsymbol{Q}_i$). Since the ratio of the net traffic rate allowed to pass through a finite-buffer queue to its original traffic rate before passing through the queue for the ith MMPP is $1 - P_{L,i}$, we should modify the rate matrix $\boldsymbol{\Lambda}_i$ for the ith MMPP to $\boldsymbol{\Lambda}_i' = \boldsymbol{\Lambda}_i(1 - P_{L,i})$ to reflect the loss effect on rate. However, no modification is required for the infinitesimal generator $\boldsymbol{Q}_i$ to get $\boldsymbol{Q}_i'$, i.e., $\boldsymbol{Q}_i' = \boldsymbol{Q}_i$ because the mean duration staying in a specific state after passing through a finite-buffer queue for the ith MMPP remains unchanged regardless of clipping caused by the loss effect. The above results are summarized into the following two equations

$$\boldsymbol{\Lambda}_i' = \boldsymbol{\Lambda}_i(1 - P_{L,i}), \tag{30}$$

$$\boldsymbol{Q}_i' = \boldsymbol{Q}_i. \tag{31}$$

Utilizing the input parameter modification scheme prior to the decomposition scheme enables us to easily decompose a finite-buffer queue based on an infinite-buffer queue. The procedure is shown diagrammatically in Fig. 1. In the following, three resultant decomposition schemes for the lossy queue are listed:

Scheme I: The scheme employs the effective server constructed using (21) or (29) (see Fig. 2) with gamma distribution fitting for residual and extra waiting times together with the input parameter modification scheme.

Scheme II: The scheme is similar to Scheme I except directly setting $H_r(s) = H(s)$, i.e., the residual service time is replaced by a full service time.

Scheme III: The scheme employs one of the decomposition schemes provided in [6], i.e., Scheme I in that paper together with the input parameter modification scheme. This scheme is similar to Scheme II, while an exponential distribution is employed to fit the extra waiting time. In this paper, the scheme is used for comparison purposes.

Remark 1. Although decomposition schemes in Section 4.1 cannot reflect the burstiness of the cross traffic, the input parameter modification scheme does incorporate burstiness into the whole scheme.

5. Performance of an individual path in a tree-type network

Using the results obtained in previous sections as well as a moment-based fitting algorithm provided in Appendix E enables one to get the per-path performance for the tree-type network. Before describing the procedure of getting per-path performance, let us first relate nodal performance measures along a reference path to the performance of the reference path for the loss probability, delay, and delay variance. Assume that $P_{L,i}^{rp}$, D_i^{rp}, and DV_i^{rp} $(1 \le i \le n_t)$ represent the loss probability, delay, and delay variance at node i along a reference path with total nodes of n_t. Then, the corresponding quantities P_L^{rp}, D^{rp}, and DV^{rp} of the reference path are given as follows:

$$P_L^{rp} = 1 - \prod_{i=1}^{n_t}(1 - P_{L,i}^{rp}), \tag{32}$$

$$D^{rp} = \sum_{i=1}^{n_t} D_i^{rp}, \tag{33}$$

$$DV^{rp} \approx \sum_{i=1}^{n_t} DV_i^{rp}. \tag{34}$$

(32) calculates the per-path loss probability via first figuring out the probability that a packet/customer is not lost at any stage from the source node to the destination node, which is equal to the product term $\prod_{i=1}^{n_t}(1 - P_{L,i}^{rp})$. Therefore, the probability of the complementary event, i.e., $1 - \prod_{i=1}^{n_t}(1 - P_{L,i}^{rp})$ is the per-path loss probability. Using the linearity of (mathematical) expectation, one can easily obtain (33). Finally, (34) is an approximate relation due to the negligence of cross-term covariance of delay times. In Section 6, we may see that the approximate relation has good accuracy through extensive numerical examples.

Based on the above results, we are now ready to depict the procedure of getting per-path performance as follows.

Procedure of per-path performance evaluation:

Step 1: Label all external input traffic streams as "external".

Step 2: Identify a reference path in the tree-type network. For convenience, we number nodes from the source node to the destination node along the reference path from 1 to n_t.

Step 3: Initialize the node counter (NC), i.e., $NC = 1$.

Step 4: ***If*** $NC > n_t$ (the stopping criterion since the destination node has been reached one node ahead), go to Step 7.

Step 5: ***If*** all input traffic streams to node NC are labelled as "external", take the following actions:

- Analyze the nodal performance using results given in Sections 3.2 and 3.2.
- Get the per-stream output process characteristics (if they should be directed to the downstream nodes) using the decomposition scheme along with the input parameter modification scheme given in Sections 4.1 and 4.2.

- Apply the moment-based fitting algorithm in Appendix E to get the traffic descriptor of a two-state MMPP for the acquired per-stream output process.
- After that, label the per-stream output processes modeled by two-state MMPPs as "external" and set $NC \leftarrow NC + 1$.
- Finally, go to Step 4.

Else, go to the next step.

Step 6:
- First, trace upstream for all traffic streams not labelled as "external" until finding "external" for them. Then, perform the decomposition scheme along with the input parameter modification scheme given in Sections 4.1 and 4.2 to get per-stream output processes to be fitted by a two-state MMPP using the fitting algorithm in Appendix E.
- Again, label these per-stream output processes as "external".
- Perform the process successively and downstream until all traffic streams to node *NC* are all labelled as "external".
- Finally, go to Step 5.

Step 7:
- Gather all nodal performance measures and use (32)–(34) to obtain the per-path performance measures.
- Finally, output all performance measures (nodal and per-path ones) and stop the procedure.

6. Numerical examples and discussions

Doing extensive numerical experiments, we now examine the accuracy of the proposed approach and discuss its limitations via investigating (i) per-stream output statistics through a finite-buffer queue and (ii) per-path performance in tree-type networks. For convenience, we use the short-hand D (deterministic), M (exponential), and E_k (k-stage Erlangian, see [16]) to represent different types of service distributions and set all mean service times to 1, i.e., $h = 1$. Also, we use the four tuple $(\sigma_1, \sigma_2, \lambda_1, \lambda_2)$ to describe the input parameters of two-state MMPPs and the squared coefficient of variation (see [16]) c_v^2 of the inter-arrival time to represent the burstiness of MMPPs. In the following numerical examples (some specific parameter settings are described therein), the analytic results and simulation results are obtained using Matlab ver. 6.5 and C/C++ codes, respectively, run on IBM compatible PCs. Note that 95% confidence intervals are used for simulation results in figures based on 10 observations.

6.1. Per-stream output statistics through a finite-buffer queue

The accuracy exhibited by the decomposition scheme together with the input parameter modification scheme to a finite-buffer queue fed by two heterogeneous two-state MMPPs is examined by observing four per-stream output statistics: mean inter-departure time $E[T_{D,i}]$, squared coefficient of variation of the inter-departure time $c^2(T_{D,i}) = \mathrm{Var}(T_{D,i})/E^2[T_{D,i}]$, the third moment of the inter-departure time $E[T_{D,i}^3]$, and lag-1 covariance of two consecutive inter-departure times $\mathrm{Cov}(T_{D,i}, T_{D,i+1})$, where $T_{D,i}$ represents the random variable of the ith inter-departure time. In the following, 2-branch models are employed for Schemes I and II because of little improvements contributed by models with more than two branches (these results are omitted for brevity).

Fixing the tagged input at (0.01, 0.01, 0.12, 0.04), where $\rho_t = 0.08$ and $c_v^2 = 1.5$ and the cross input at (0.1, 0.1, 0.7, 0.3) with $\rho_c = 0.5$ and $c_v^2 = 1.258$ (denoted by subscript LB in the figures) and (0.0005, 0.0015, 0.635, 0.095) with $\rho_c = 0.5$ and $c_v^2 = 2.783$ (denoted by subscript HB in the figures), respectively, we observe per-stream output statistics through a finite-buffer queue. Varying the buffer size, we get results for D and M servers as shown in Figs. 3 and 4, respectively. Fig. 3 shows the following observations. (i) The mean inter-departure time is accurately (actually, it is exactly) analyzed by the three schemes and it falls down when the buffer size becomes large since a higher tagged rate is allowed to pass through the queue. Obviously, more bursty cross input causes longer mean inter-departure times as shown in Fig. 3 because a lower tagged rate is allowed to pass through the queue. (ii) Note that there is visual inaccuracy for $c^2(T_{D,i})$ when the buffer size is small since the scale is focused on [1.40, 1.52]. In fact, the maximum errors when buffer sizes are 1, 4, and 7 are 4.32%, 1.58%, and 0.97%, respectively. In general, Scheme III provides a bit better estimates than other schemes when the buffer size is larger than 3, while Scheme II performs more accurately when the buffer size is smaller than 3 and cross input with low burstiness is considered. (iii) All schemes have good estimates for $E[T_{D,i}^3]$. After further examining, Scheme III performs best. (iv) As for $\mathrm{Cov}(T_{D,i}, T_{D,i+1})$, Scheme III

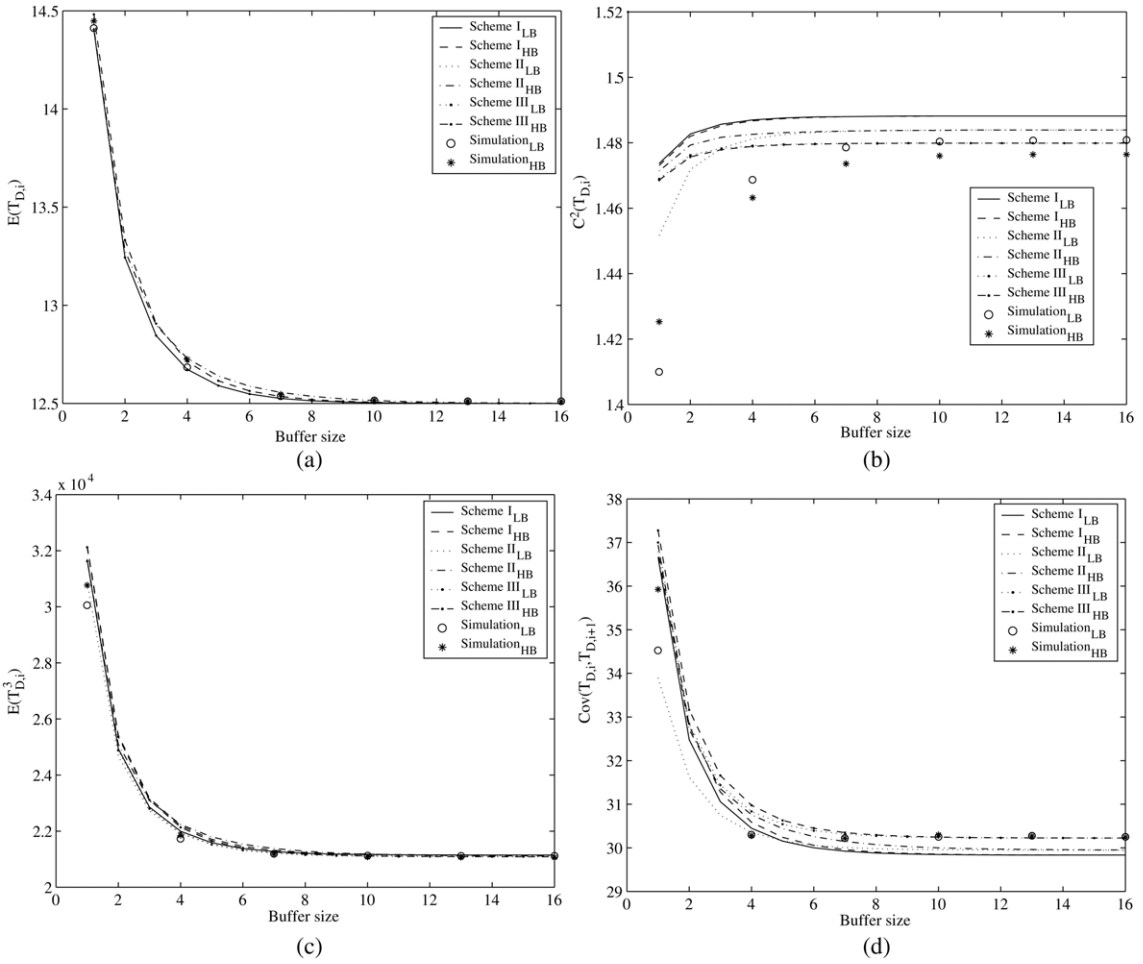


Fig. 3. Per-stream output statistics vs. buffer size for the finite D server.

has better estimates than other schemes when the buffer size is larger than 7. When the buffer size falls within [1, 6], Scheme I (II) performs best when cross input with high (low) burstiness is considered. Fig. 4 demonstrates that all schemes perform well to estimate per-stream output statistics for the M server (note that Schemes I and II behave identically because of the memoryless property of the exponential distribution). Of course, visual inaccuracy occurs for $c^2(T_{D,i})$ when the buffer size is small since the scale is focused on [1.40, 1.54]. From the above discussions, we reach the following remark.

Remark 2. Schemes I and II are mostly suggested for the case when small buffers in networks are considered. Of course, Scheme III suits the case with larger buffers.

In the following, we leave out the results of Scheme III (they have been discussed in [6]) and just show the results of Schemes I and II. We denote the analytic results corresponding to Scheme I and Scheme II by Analysis I and Analysis II, respectively.

6.2. Per-path performance in tree-type networks

Now, let us take a look at the per-path performance in a network in which two network configurations are considered, i.e., tandem as shown in Fig. 5(a) called Scenario I and tree-type as shown in Fig. 5(b) called Scenario II.

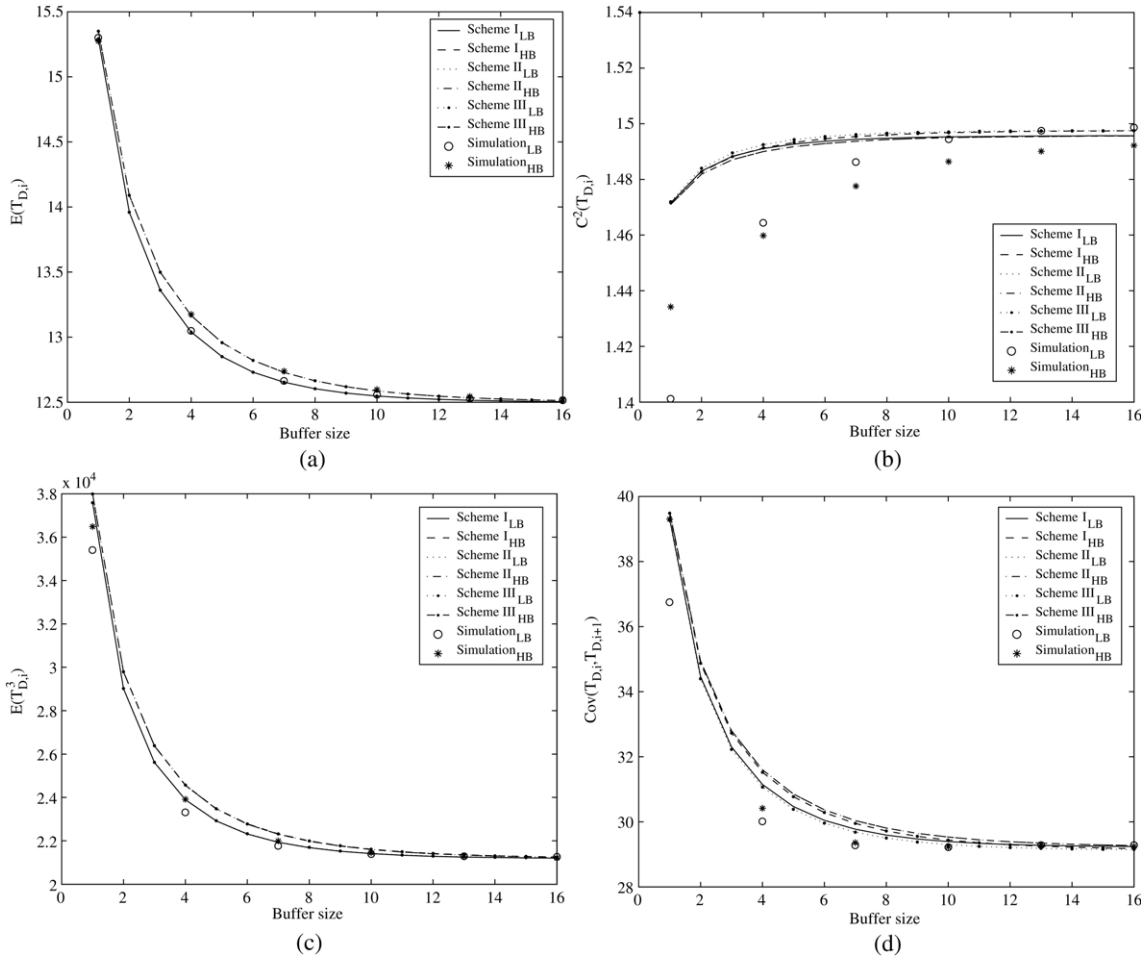
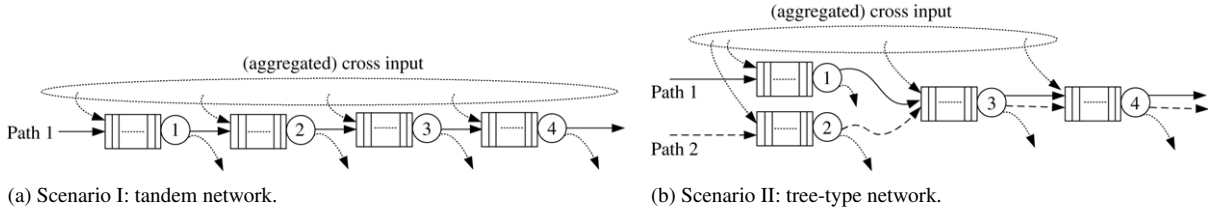


Fig. 4. Per-stream output statistics vs. buffer size for the finite M server.



(a) Scenario I: tandem network.

(b) Scenario II: tree-type network.

Fig. 5. Network configurations.

Fig. 5(a) considers a tagged path passing through all nodes with cross paths merely passing through one node of the tagged path. In real networks, routes of different paths may overlap. Therefore, Fig. 5(b) considers two tagged paths with partially overlapped routes. In the following, we set the tagged traffic load around 0.1, the cross traffic load over the range of [0.4, 0.6] (median load), and buffer size over the range of [4, 9].

For Scenario I, a two-state MMPP (0.01, 0.01, 0.12, 0.04) is injected and routed along the tagged path, while homogeneous (aggregated) cross input streams described by (0.1, 0.1, 0.7, 0.3) with low burstiness (LB) $c_v^2 = 1.258$ and (0.0005, 0.0015, 0.635, 0.095) with high burstiness (HB) $c_v^2 = 2.783$ are considered. Under service (buffer size) assignments for four nodes of (D(4), D(4), D(4), D(4)), (M(4), M(4), M(4), M(4)), and

Table 1
Loss probability, delay, and delay variance for the tagged path of Scenario I with (D(4), D(4), D(4), D(4)) service (buffer size) assignment

		Node 1	Node 2	Node 3	Node 4	End-to-end
LB	*Loss probability*					
	Simulation	1.3698e−02	1.2338e−02	1.1932e−02	1.1619e−02	4.9587e−02
	Analysis I	1.3699e−02	1.3433e−02	1.3191e−02	1.2970e−02	5.3293e−02
	Error	0.01%	8.15%	9.54%	10.41%	6.95%
	Analysis II	1.3699e−02	1.3388e−02	1.3112e−02	1.2865e−02	5.3065e−02
	Error	0.01%	7.84%	9.00%	9.69%	6.56%
	Mean delay					
	Simulation	0.7357	0.6968	0.6844	0.6757	2.7926
	Analysis I	0.7443	0.7387	0.7336	0.7288	2.9454
	Error	1.16%	5.67%	6.71%	7.29%	5.19%
	Analysis II	0.7443	0.7378	0.7320	0.7267	2.9408
	Error	1.16%	5.56%	6.50%	7.02%	5.04%
	Delay variance					
	Simulation	0.9052	0.8872	0.8718	0.8604	3.5246
	Analysis I	0.9099	0.9032	0.8970	0.8913	3.6014
	Error	0.52%	1.77%	2.81%	3.47%	2.13%
	Analysis II	0.9099	0.9021	0.8951	0.8887	3.5958
	Error	0.52%	1.65%	2.60%	3.18%	1.98%
HB	*Loss probability*					
	Simulation	1.6733e−02	1.5031e−02	1.4582e−02	1.4023e−02	6.0369e−02
	Analysis I	1.6761e−02	1.6378e−02	1.6031e−02	1.5714e−02	6.4884e−02
	Error	0.17%	8.22%	9.04%	10.76%	6.96%
	Analysis II	1.6761e−02	1.6324e−02	1.5938e−02	1.5592e−02	6.4615e−02
	Error	0.17%	7.92%	8.51%	10.06%	6.57%
	Mean delay					
	Simulation	0.7749	0.7360	0.7253	0.7139	2.9501
	Analysis I	0.8005	0.7939	0.7879	0.7822	3.1645
	Error	3.20%	7.29%	7.95%	8.73%	6.78%
	Analysis II	0.7749	0.7930	0.7863	0.7802	3.1600
	Error	3.20%	7.19%	7.76%	8.50%	6.64%
	Delay variance					
	Simulation	1.0203	1.0078	0.9900	0.9742	3.9923
	Analysis I	1.0173	1.0091	1.0014	0.9943	4.0221
	Error	0.29%	0.13%	1.14%	2.02%	0.74%
	Analysis II	1.0173	1.0080	0.9995	0.9917	4.0165
	Error	0.29%	0.02%	0.95%	1.76%	0.60%

(D(5), M(9), E_4(8), D(6)), we show results in Tables 1–3. From these tables, we notice that (i) Scheme II gives a bit better estimates than Scheme I; (ii) the maximum errors observed for these tables are 10.41% (10.76%), 4.33% (7.90%), and 9.09% (9.13%), respectively, when the LB (HB) cross input is considered. Of course, the above results suggest that the proposed approach performs better for the exponential server (the maximum error is 7.90%) than the deterministic one (the maximum error is 10.76%). That explains why the maximum error for the mixed servers in Table 3 falls between the case of all exponential servers and that of all deterministic ones.

As for Scenario II, we employ (D(5), E_4(9), M(8), D(6)) service (buffer size) assignment and traffic parameters described in Table 4. The results are shown in Table 5 in which a 2.65% worst case error for delay and delay variance and 6.50% worst case error for loss probabilities are reported and Table 6 in which a 5.55% worst case error for delay and delay variance and 4.14% worst case error for loss probabilities are observed. The above results reveal that our approach is applicable to tree-type networks with overlapped routes.

Finally, we comment on the suitability of the proposed approach regarding the scale of networks using the following remark.

Table 2
Loss probability, delay, and delay variance for the tagged path of Scenario I with (M(4), M(4), M(4), M(4)) service (buffer size) assignment

		Node 1	Node 2	Node 3	Node 4	End-to-end
LB	*Loss probability*					
	Simulation	4.1391e−02	3.9956e−02	3.8862e−02	3.7966e−02	1.5818e−01
	Analysis I	4.1357e−02	4.0431e−02	3.9580e−02	3.8796e−02	1.6016e−01
	Error	0.08%	1.17%	1.81%	2.14%	1.24%
	Analysis II	4.1357e−02	4.0431e−02	3.9580e−02	3.8796e−02	1.6016e−01
	Error	0.08%	1.17%	1.81%	2.14%	1.24%
	Mean delay					
	Simulation	1.0754	1.0612	1.0497	1.0391	4.2254
	Analysis I	1.1171	1.1061	1.0958	1.0861	4.4051
	Error	3.73%	4.06%	4.21%	4.33%	4.08%
	Analysis II	1.1171	1.1061	1.0958	1.0861	4.4051
	Error	3.73%	4.06%	4.21%	4.33%	4.08%
	Delay variance					
	Simulation	2.6068	2.5760	2.5518	2.5293	10.2639
	Analysis I	2.6626	2.6408	2.6204	2.6012	10.5250
	Error	2.10%	2.45%	2.62%	2.76%	2.48%
	Analysis II	2.6626	2.6408	2.6204	2.6012	10.5250
	Error	2.10%	2.45%	2.62%	2.76%	2.48%
HB	*Loss probability*					
	Simulation	5.0623e−02	4.8805e−02	4.7359e−02	4.5900e−02	1.9269e−01
	Analysis I	5.0525e−02	4.9277e−02	4.8141e−02	4.7103e−02	1.9505e−01
	Error	0.19%	0.96%	1.62%	2.55%	1.21%
	Analysis II	5.0525e−02	4.9277e−02	4.8141e−02	4.7103e−02	1.9505e−01
	Error	0.19%	0.96%	1.62%	2.55%	1.21%
	Mean delay					
	Simulation	1.0709	1.0548	1.0430	1.0276	4.1963
	Analysis I	1.1510	1.1383	1.1266	1.1158	4.5317
	Error	6.96%	7.34%	7.42%	7.90%	7.40%
	Analysis II	1.1510	1.1383	1.1266	1.1158	4.5317
	Error	6.96%	7.34%	7.42%	7.90%	7.40%
	Delay variance					
	Simulation	3.2225	3.1820	3.1517	3.1199	12.6761
	Analysis I	3.3294	3.2975	3.2687	3.2426	13.1382
	Error	3.21%	3.50%	3.58%	3.78%	3.52%
	Analysis II	3.3294	3.2975	3.2687	3.2426	13.1382
	Error	3.21%	3.50%	3.58%	3.78%	3.52%

Remark 3. We do not recommend applying the proposed approach to large-scale networks because errors in output statistics may propagate downstream, thus causing unacceptable estimates for per-path performance measures.

7. Conclusions

Two decomposition schemes accompanied with an input parameter modification scheme are proposed in this paper to facilitate extraction of per-stream output processes for the finite-buffer queue based on an infinite-buffer queue. Through numerical experiments, we show that these schemes perform well. With the aid of nodal performance analysis, decomposition scheme, input parameter modification scheme, and moment-based fitting algorithm, a systematic method for per-path performance evaluation of a tree-type network is then proposed. Via extensive numerical experiments, the accuracy of the method is examined and its limitations are given. This paper extends the scope of application for [6] by getting more path-wise performance measures, including loss probability, delay, and delay variance, which are critical quality-of-service (QoS) measures in high-speed networks and has potential

Table 3
Loss probability, delay, and delay variance for the tagged path of Scenario I with (D(5), M(9), $E_4(8)$, D(6)) service (buffer size) assignment

		Node 1	Node 2	Node 3	Node 4	End-to-end
LB	*Loss probability*					
	Simulation	7.1522e−03	5.2664e−03	2.1448e−03	3.3162e−03	1.7880e−02
	Analysis I	7.1818e−03	5.4619e−03	2.1988e−03	3.6479e−03	1.8490e−02
	Error	0.41%	3.58%	2.46%	9.09%	3.30%
	Analysis II	7.1818e−03	5.4420e−03	2.1882e−03	3.6216e−03	1.8434e−02
	Error	0.41%	3.23%	1.98%	8.43%	3.01%
	Mean delay					
	Simulation	0.8096	1.5406	1.0586	0.8190	4.2278
	Analysis I	0.8146	1.5705	1.0722	0.8489	4.3062
	Error	0.61%	1.90%	1.27%	3.52%	1.82%
	Analysis II	0.8146	1.5688	1.0708	0.8471	4.3013
	Error	0.61%	1.80%	1.14%	3.32%	1.71%
	Delay variance					
	Simulation	1.1673	5.4435	2.4001	1.3449	10.3558
	Analysis I	1.1720	5.4751	2.4155	1.3663	10.4289
	Error	0.40%	0.58%	0.64%	1.57%	0.70%
	Analysis II	1.1720	5.4685	2.4116	1.3628	10.4149
	Error	0.40%	0.46%	0.48%	1.31%	0.57%
HB	*Loss probability*					
	Simulation	9.1392e−03	8.8423e−03	3.2421e−03	4.3086e−03	2.5532e−02
	Analysis I	9.1445e−03	9.2255e−03	3.3835e−03	4.7786e−03	2.6532e−02
	Error	0.06%	4.15%	3.99%	8.54%	3.77%
	Analysis II	9.1445e−03	9.1954e−03	3.3670e−03	4.7414e−03	2.6448e−02
	Error	0.06%	3.84%	3.71%	9.13%	3.46%
	Mean delay					
	Simulation	0.8707	1.6974	1.1743	0.8894	4.6318
	Analysis I	0.8908	1.7629	1.2085	0.9374	4.7996
	Error	2.26%	3.72%	2.83%	5.12%	3.50%
	Analysis II	0.8908	1.7612	1.2070	0.9354	4.7944
	Error	2.26%	3.62%	2.71%	4.92%	3.39%
	Delay variance					
	Simulation	1.3530	6.5990	2.9386	1.5847	12.4753
	Analysis I	1.3452	6.6161	2.9468	1.5981	12.5062
	Error	0.58%	0.26%	0.28%	0.84%	0.25%
	Analysis II	1.3452	6.6094	2.9421	1.5938	12.4905
	Error	0.58%	0.16%	0.12%	0.57%	0.12%

applications in such networks to help assess the path-wise performance in the designing/dimensioning stage. With the help of [8], one is able to further extend the results of this paper to tree-type networks with probabilistic routing.

Appendix A. Formulas for $W^{(1)}$ and $W^{(2)}$

In the following, we explicitly express $\boldsymbol{W}^{(k)}$ for $k = 1$ and $k = 2$. As for the results when $k \geq 3$, they can still be obtained but may involve lengthy algebraic manipulations. By differentiating (10) and (15), we have

$$\boldsymbol{W}^{(1)} = \frac{1}{(1-\boldsymbol{y}_K\boldsymbol{e})(h-\boldsymbol{x}_0\boldsymbol{R}^{-1}\boldsymbol{e})}\sum_{n=1}^{K-1}\left\{\sum_{k=0}^{n-1}\boldsymbol{x}_k\boldsymbol{R}_{n-k-1}^{(1)} - \boldsymbol{x}_0\boldsymbol{U}\boldsymbol{R}_{n-1}^{(1)} - \sum_{k=1}^{n}\boldsymbol{x}_k\boldsymbol{R}_{n-k}^{(1)} + (n-1)H^{(1)}\sum_{k=0}^{n-1}\boldsymbol{x}_k\boldsymbol{R}_{n-k-1} - nH^{(1)}\boldsymbol{x}_0\boldsymbol{U}\boldsymbol{R}_{n-1} - nH^{(1)}\sum_{k=1}^{n}\boldsymbol{x}_k\boldsymbol{R}_{n-k}\right\}, \tag{A.1}$$

Table 4
Input traffic parameters for Scenario II

Parameter set I				
	Node 1	Node 2	Node 3	Node 4
Path 1	(0.01, 0.01, 0.08, 0.04)	–	–	–
C_v^2	$C_v^2 = 1.1818$	–	–	–
Path 2	–	(0.01, 0.04, 0.15 0)	–	–
C_v^2	–	$C_v^2 = 2.20$	–	–
Cross path	(0.02, 0.02, 1.6/3, 0.8/3)	(0.01, 0.02, 8.1/14, 2.7/14)	(0.01, 0.01, 1.6/3, 0.8/3)	(0.01, 0.04, 0.5, 0.25)
C_v^2	$C_v^2 = 1.2247$	$C_v^2 = 1.5286$	$C_v^2 = 1.2366$	$C_v^2 = 1.1356$
Parameter set II				
	Node 1	Node 2	Node 3	Node 4
Path 1	(0.001, 0.001, 0.1000, 0.0200)	–	–	–
C_v^2	$C_v^2 = 2.5094$	–	–	–
Path 2	–	(0.001, 0.002, 0.0480, 0.2640)	–	–
C_v^2	–	$C_v^2 = 2.5912$	–	–
Cross path	(0.0004, 0.0006, 0.1500, 0.7750)	(0.005, 0.0075, 0.15, 0.7500)	(0.0001, 0.0001, 0.1500, 0.7750)	(0.0005, 0.00075, 0.1655, 0.87675)
C_v^2	$C_v^2 = 2.6074$	$C_v^2 = 2.5936$	$C_v^2 = 2.5464$	$C_v^2 = 2.6670$

$$
\begin{aligned}
\boldsymbol{W}^{(2)} = {} & \frac{1}{(1-\boldsymbol{y}_K\boldsymbol{e})(h-\boldsymbol{x}_0\boldsymbol{R}^{-1}\boldsymbol{e})}\sum_{n=1}^{K-1}\left\{\sum_{k=0}^{n-1}\boldsymbol{x}_k\boldsymbol{R}_{n-k-1}^{(2)} - \boldsymbol{x}_0\boldsymbol{U}\boldsymbol{R}_{n-1}^{(2)}\right. \\
& - \sum_{k=1}^{n}\boldsymbol{x}_k\boldsymbol{R}_{n-k}^{(2)} + 2(n-1)H^{(1)}\sum_{k=0}^{n-1}\boldsymbol{x}_k\boldsymbol{R}_{n-k-1}^{(1)} - 2nH^{(1)}\boldsymbol{x}_0\boldsymbol{U}\boldsymbol{R}_{n-1}^{(1)} \\
& - 2nH^{(1)}\sum_{k=1}^{n}\boldsymbol{x}_k\boldsymbol{R}_{n-k}^{(1)} + [(n-1)H^{(2)} + (n-1)(n-2)(H^{(1)})^2]\sum_{k=0}^{n-1}\boldsymbol{x}_k\boldsymbol{R}_{n-k-1} \\
& \left. - [nH^{(2)} + n(n-1)(H^{(1)})^2]\left[\boldsymbol{x}_0\boldsymbol{U}\boldsymbol{R}_{n-1} + \sum_{k=1}^{n}\boldsymbol{x}_k\boldsymbol{R}_{n-k}\right]\right\}, \qquad \text{(A.2)}
\end{aligned}
$$

where $H^{(i)} = \frac{\mathrm{d}^i}{\mathrm{d}s^i}H(s)|_{s=0} = H^{(i)}(0)$ and

$$\boldsymbol{R}_0^{(k)} = (-1)^k k!\boldsymbol{R}^{-(k+1)}, \quad k \geq 1, \qquad \text{(A.3)}$$

$$\boldsymbol{R}_n^{(1)} = -\boldsymbol{R}^{-2}[-\boldsymbol{\Lambda}\boldsymbol{R}^{-1}]^n + \boldsymbol{R}^{-1}\left\{\sum_{j=0}^{n-1}[-\boldsymbol{\Lambda}\boldsymbol{R}^{-1}]^j\boldsymbol{\Lambda}\boldsymbol{R}^{-2}[-\boldsymbol{\Lambda}\boldsymbol{R}^{-1}]^{n-1-j}\right\}, \quad n \geq 1, \qquad \text{(A.4)}$$

$$
\begin{aligned}
\boldsymbol{R}_n^{(2)} = {} & 2\boldsymbol{R}^{-3}[-\boldsymbol{\Lambda}\boldsymbol{R}^{-1}]^n - 2\boldsymbol{R}^{-2}\left\{\sum_{j=0}^{n-1}[-\boldsymbol{\Lambda}\boldsymbol{R}^{-1}]^j\boldsymbol{\Lambda}\boldsymbol{R}^{-2}[-\boldsymbol{\Lambda}\boldsymbol{R}^{-1}]^{n-1-j}\right\} \\
& + \boldsymbol{R}^{-1}\left\{\sum_{j=0}^{n-1}\sum_{l=0}^{j-1}[-\boldsymbol{\Lambda}\boldsymbol{R}^{-1}]^l\boldsymbol{\Lambda}\boldsymbol{R}^{-2}[-\boldsymbol{\Lambda}\boldsymbol{R}^{-1}]^{j-1-l}\boldsymbol{\Lambda}\boldsymbol{R}^{-2}[-\boldsymbol{\Lambda}\boldsymbol{R}^{-1}]^{n-1-j}\right. \\
& - 2\sum_{j=0}^{n-1}[-\boldsymbol{\Lambda}\boldsymbol{R}^{-1}]^j\boldsymbol{\Lambda}\boldsymbol{R}^{-3}[-\boldsymbol{\Lambda}\boldsymbol{R}^{-1}]^{n-1-j} + \sum_{j=0}^{n-1}[-\boldsymbol{\Lambda}\boldsymbol{R}^{-1}]^j\boldsymbol{\Lambda}\boldsymbol{R}^{-2} \\
& \left.\times\sum_{m=0}^{n-2-j}[-\boldsymbol{\Lambda}\boldsymbol{R}^{-1}]^m\boldsymbol{\Lambda}\boldsymbol{R}^{-2}[-\boldsymbol{\Lambda}\boldsymbol{R}^{-1}]^{n-2-j-m}\right\}, \quad n \geq 1. \qquad \text{(A.5)}
\end{aligned}
$$

Table 5
Loss probability, delay, and delay variance for individual paths in Scenario II using parameter set I specified in Table 4 with (D(5), E_4(9), M(8), D(6)) service (buffer size) assignment

		Node 1	Node 2	Node 3	Node 4	End-to-end
	Loss probability					
Path 1	Simulation	1.2161e−03	–	7.5290e−03	2.8709e−03	1.1616e−02
	Analysis I	1.2151e−03	–	7.7102e−03	3.0166e−03	1.1942e−02
	Error	0.08%	–	2.35%	4.83%	2.73%
	Analysis II	1.2151e−03	–	7.7975e−05	3.9343e−03	4.0366e−03
	Error	0.08%	–	2.17%	4.53%	2.53%
Path 2	Simulation	–	1.3113e−03	8.0326e−03	3.0950e−03	1.2439e−02
	Analysis I	–	1.3119e−03	8.3467e−03	3.3103e−03	1.2969e−02
	Error	–	0.05%	3.76%	6.50%	4.09%
	Analysis II	–	1.3119e−03	8.3247e−03	3.2971e−03	1.2934e−02
	Error	–	0.05%	3.51%	6.13%	3.83%
	Mean delay					
Path 1	Simulation	0.4791	–	1.4257	0.8857	2.7950
	Analysis I	0.4797	–	1.4540	0.8974	2.8311
	Error	0.13%	–	1.95%	1.30%	1.43%
	Analysis II	0.4797	–	1.4534	0.8969	2.8300
	Error	0.13%	–	1.91%	1.25%	1.40%
Path 2	Simulation	–	1.1117	1.4807	0.9251	3.5175
	Analysis I	–	1.1131	1.5210	0.9454	3.5795
	Error	–	0.13%	2.65%	2.15%	1.73%
	Analysis II	–	1.1131	1.5196	0.9443	3.5770
	Error	–	0.13%	2.56%	2.03%	1.66%
	Delay variance					
Path 1	Simulation	0.5972	–	4.7685	1.3250	6.6907
	Analysis I	0.5978	–	4.8163	1.3408	6.7549
	Error	0.10%	–	0.99%	1.18%	0.95%
	Analysis II	0.5978	–	4.8137	1.3396	6.7511
	Error	0.10%	–	0.94%	1.09%	0.89%
Path 2	Simulation	–	2.6554	4.9475	1.3790	8.9819
	Analysis I	–	2.6569	5.0026	1.4027	9.0622
	Error	–	0.06%	1.10%	1.69%	0.89%
	Analysis II	–	2.6569	4.9980	1.4008	9.0557
	Error	–	0.06%	1.01%	1.56%	0.81%

Appendix B. Overall output process of an MMPP/G/1 queue

For the readers' convenience, we reproduce some results pertinent to the overall output process of an MMPP/G/1 queue from [7] as follows:

$$E[T_{D,i}] = h - \boldsymbol{x}_0\boldsymbol{R}^{-1}\boldsymbol{e}, \tag{B.1}$$

$$E[T_{D,i}^2] = H^{(2)} - 2h\boldsymbol{x}_0\boldsymbol{R}^{-1}\boldsymbol{e} + 2\boldsymbol{x}_0\boldsymbol{R}^{-2}\boldsymbol{e}, \tag{B.2}$$

$$E[T_{D,i}^3] = -H^{(3)} - 3H^{(2)}\boldsymbol{x}_0\boldsymbol{R}^{-1}\boldsymbol{e} + 6h\boldsymbol{x}_0\boldsymbol{R}^{-2}\boldsymbol{e} - 6\boldsymbol{x}_0\boldsymbol{R}^{-3}\boldsymbol{e}, \tag{B.3}$$

$$\begin{aligned}\mathrm{Cov}(T_{D,i}, T_{D,i+1}) \;=\; & h\boldsymbol{x}_0\boldsymbol{R}^{-1}\boldsymbol{e} - [\boldsymbol{x}_0\boldsymbol{R}^{-1}\boldsymbol{e}]^2 + \boldsymbol{x}_1\boldsymbol{A}_0^{(1)}\boldsymbol{R}^{-1}\boldsymbol{e} \\ & + \boldsymbol{x}_0\boldsymbol{R}^{-1}\boldsymbol{U}\boldsymbol{A}_0\boldsymbol{R}^{-1}\boldsymbol{e} + \boldsymbol{x}_0\boldsymbol{U}\boldsymbol{A}_0^{(1)}\boldsymbol{R}^{-1}\boldsymbol{e},\end{aligned} \tag{B.4}$$

where $T_{D,i}$ stands for the random variable of the ith inter-departure time. In Appendix C, we give the explicit formulas of $\boldsymbol{A}_0$ and $\boldsymbol{A}_0^{(1)}$ for several specific service distributions. The calculation of $\boldsymbol{x}_0$ is given in [9] and is related to the computation of parameter γ_n $(n \geq 0)$ given in Appendix D.

Table 6
Loss probability, delay, and delay variance for individual paths in Scenario II using parameter set II specified in Table 4 with (D(5), $E_4(9)$, M(8), D(6)) service (buffer size) assignment

		Node 1	Node 2	Node 3	Node 4	End-to-end
	Loss probability					
Path 1	Simulation	1.4536e−02	–	3.3977e−02	4.3046e−02	9.1559e−02
	Analysis I	1.4495e−02	–	3.4534e−02	4.4094e−02	9.3123e−02
	Error	0.28%	–	1.61%	2.38%	1.68%
	Analysis II	1.4495e−02	–	3.4514e−02	4.4088e−02	9.3097e−02
	Error	0.28%	–	1.56%	2.36%	1.65%
Path 2	Simulation	–	2.5041e−02	4.1881e−02	5.2856e−02	1.1978e−02
	Analysis I	–	2.4987e−02	4.3691e−02	5.5117e−02	1.2380e−02
	Error	–	0.22%	4.14%	4.10%	3.24%
	Analysis II	–	2.4987e−02	4.3570e−02	5.4999e−02	1.2356e−02
	Error	–	0.22%	3.88%	3.90%	3.06%
	Mean delay					
Path 1	Simulation	0.7547	–	1.7531	1.3672	3.8750
	Analysis I	0.7605	–	1.8295	1.4374	4.0274
	Error	0.76%	–	4.18%	4.88%	3.78%
	Analysis II	0.7605	–	1.8292	1.4374	4.0271
	Error	0.76%	–	4.16%	4.88%	3.78%
Path 2	Simulation	–	2.0481	1.9452	1.4884	5.4817
	Analysis I	–	2.0719	2.0442	1.5759	5.6920
	Error	–	1.15%	4.84%	5.55%	3.69%
	Analysis II	–	2.0719	2.0417	1.5747	5.6883
	Error	–	1.15%	4.73%	5.48%	3.63%
	Delay variance					
Path 1	Simulation	1.4459	–	7.2154	3.4115	12.0728
	Analysis I	1.4526	–	7.3408	3.4784	12.2720
	Error	0.46%	–	1.71%	1.92%	1.62%
	Analysis II	1.4526	–	7.3396	3.4786	12.2710
	Error	0.46%	–	1.69%	1.93%	1.61%
Path 2	Simulation	–	7.4915	7.9184	3.6854	19.0953
	Analysis I	–	7.5269	8.0314	3.7547	19.3130
	Error	–	0.47%	1.41%	1.85%	1.13%
	Analysis II	–	7.5269	8.0241	3.7533	19.4040
	Error	–	0.47%	1.32%	1.81%	1.08%

Appendix C. Formulas for A_0 and $A_0^{(1)}$

A_0 and $A_0^{(1)}$ are defined as follows:

$$A_0 = \int_0^\infty \mathrm{d}\tilde{A}_0(x), \tag{C.1}$$

$$A_0^{(1)} = -\int_0^\infty x\,\mathrm{d}\tilde{A}_0(x), \tag{C.2}$$

where $\tilde{A}_0(x) = \int_0^x \mathrm{e}^{Ru}\,\mathrm{d}\tilde{H}(u)$. In the following, we give formulas for some distributions.

1. Deterministic service distribution (also see [6]), i.e., $\frac{\mathrm{d}\tilde{H}(x)}{\mathrm{d}x} = \delta(x-h)$:

$$A_0 = \exp[Rh], \tag{C.3}$$

$$A_0^{(1)} = -hA_0. \tag{C.4}$$

2. k-stage Erlangian service distribution (also see [6]), i.e., $\frac{d\tilde{H}(x)}{dx} = \Gamma(x; k, \mu)$ [13]:

$$\boldsymbol{A}_0 = \mu^k[\mu\boldsymbol{I} - \boldsymbol{R}]^{-k}, \tag{C.5}$$

$$\boldsymbol{A}_0^{(1)} = -k\boldsymbol{A}_0[\mu\boldsymbol{I} - \boldsymbol{R}]^{-1}. \tag{C.6}$$

3. Serial connection of a deterministic server and a gamma server, i.e., $\frac{d\tilde{H}(x)}{dx} = \Gamma(x - h; \alpha, \mu)$ (where the parameter $\alpha > 0$):

$$\boldsymbol{A}_0 = \mu^\alpha \exp[\boldsymbol{R}h][\mu\boldsymbol{I} - \boldsymbol{R}]^{-\alpha}, \tag{C.7}$$

$$\boldsymbol{A}_0^{(1)} = -h\boldsymbol{A}_0 - \alpha\boldsymbol{A}_0[\mu\boldsymbol{I} - \boldsymbol{R}]^{-1}. \tag{C.8}$$

4. Serial connection of $\Gamma(x; \alpha_1, \mu_1)$ and $\Gamma(x; \alpha_2, \mu_2)$, i.e., $\frac{d\tilde{H}(x)}{dx} = \Gamma(x; \alpha_1, \mu_1) \star \Gamma(x; \alpha_2, \mu_2)$ where $\star$ denotes the functional convolution:

$$\boldsymbol{A}_0 = \mu_1^{\alpha_1}\mu_2^{\alpha_2}[\mu_1\boldsymbol{I} - \boldsymbol{R}]^{-\alpha_1}[\mu_2\boldsymbol{I} - \boldsymbol{R}]^{-\alpha_2}, \tag{C.9}$$

$$\boldsymbol{A}_0^{(1)} = -\alpha_1\boldsymbol{A}_0[\mu_1\boldsymbol{I} - \boldsymbol{R}]^{-1} - \alpha_2\boldsymbol{A}_0[\mu_2\boldsymbol{I} - \boldsymbol{R}]^{-1}. \tag{C.10}$$

5. Compound model: the resultant $\boldsymbol{A}_0$ and $\boldsymbol{A}_0^{(1)}$ are weighted sums of the individual $\boldsymbol{A}_0$ and $\boldsymbol{A}_0^{(1)}$.

Appendix D. Computation of γ_n

The computation γ_n for deterministic, exponential, and Erlang servers have been provided in [9]. In the following, we derive γ_n for serial connection servers. The formal definition of γ_n is given as follows:

$$\gamma_n = \int_0^\infty \mathrm{e}^{-\Theta x}\frac{(\Theta x)^n}{n!}\,\mathrm{d}\tilde{H}(x), \quad n \geq 0,$$

where the definition of Θ is given in [9].

1. Serial connection of a deterministic server and a gamma server, i.e., $\frac{d\tilde{H}(x)}{dx} = \Gamma(x - h; \alpha, \mu)$:

$$\gamma_0 = \left(\frac{\mu}{\mu + \Theta}\right)^\alpha \mathrm{e}^{-\Theta h}, \tag{D.1}$$

$$\gamma_n = \frac{\gamma_0\Theta^n}{\Gamma(\alpha)}\sum_{i=0}^{n}\frac{\Gamma(\alpha + i)h^{n-i}}{\Gamma(n - i + 1)\Gamma(i + 1)}(\mu + \Theta)^{-i}, \quad n \geq 1, \tag{D.2}$$

   where $\Gamma(\alpha)$ is the gamma function [13].

2. Serial connection of $\Gamma(x; \alpha_1, \mu_1)$ and $\Gamma(x; \alpha_2, \mu_2)$, i.e., $\frac{d\tilde{H}(x)}{dx} = \Gamma(x; \alpha_1, \mu_1) \star \Gamma(x; \alpha_2, \mu_2)$:

$$\gamma_0 = \left(\frac{\mu_1}{\mu_1 + \Theta}\right)^{\alpha_1}\left(\frac{\mu_2}{\mu_2 + \Theta}\right)^{\alpha_2}, \tag{D.3}$$

$$\gamma_n = \frac{\gamma_0\Theta^n}{\Gamma(\alpha_1)\Gamma(\alpha_2)}\sum_{i=0}^{n}\frac{\Gamma(\alpha_1 + i)\Gamma(\alpha_2 + n - i)}{\Gamma(n - i + 1)\Gamma(i + 1)}(\mu_1 + \Theta)^{-i}(\mu_2 + \Theta)^{i-n}, \quad n \geq 1. \tag{D.4}$$

3. Compound model: the resultant γ_n is a weighted sum of the individual γ_n.

Appendix E. The moment-based fitting algorithm

The following moment-based fitting algorithm rendered from [6] is used to match the departure process of the heterogeneous MMPPs/G/1 queue for the tagged input MMPP specified by $(\sigma_{11}, \sigma_{21}, \lambda_{11}, \lambda_{21})$ to a two-state MMPP with target parameters $(\sigma_1^{(m)}, \sigma_2^{(m)}, \lambda_1^{(m)}, \lambda_2^{(m)})$ using four departure statistics: the first moment of the inter-departure time $A_d = E[T_{D,i}]$, the squared coefficient of variation of the inter-departure time $B_d = \mathrm{Var}[T_{D,i}]/E^2[T_{D,i}]$, the third moment of the inter-departure time $C_d = E[T_{D,i}^3]$, and the lag-1 covariance of two adjacent inter-departure

times $D_d = \mathrm{Cov}(T_{D,i}, T_{D,i+1})$. For convenience, we employ the following notation: $C_{d,1} = (B_d + 1)(A_d)^2$, $D_{d,1} = (B_d - 1)(A_d)^2/2$, $\zeta = 1/A_d$, $\eta = D_d/[A_d(D_{d,1} - D_d)]$, and $\xi = [(B_d - 1)(\zeta + \eta) + 2\eta]/(2\zeta)$. The algorithm is now given as follows:

Step 1: Input departure statistics of the tagged traffic with parameters $(\sigma_{11}, \sigma_{21}, \lambda_{11}, \lambda_{21})$.

Step 2: If $B_d < 1$, then use a Poisson process to emulate the departure process by setting $\lambda_1^{(m)} = \lambda_2^{(m)} = \zeta$, $\sigma_1^{(m)} = \sigma_{11}, \sigma_2^{(m)} = \sigma_{21}$ and go to Step 5; else go to the next step.

Step 3: If $\eta < 0$, then set $\lambda_1^{(m)} = \lambda_{11}, \lambda_2^{(m)} = \lambda_{21}$, use (E.1) and (E.2) to get $\sigma_1^{(m)}$ and $\sigma_2^{(m)}$, and go to Step 5; else go to the next step.

Step 4: If $\alpha > 0$, directly use (E.3) and (E.4); else set $\alpha = \sigma_{11} + \sigma_{21}$ then use (E.3) and (E.4) to get $(\sigma_1^{(m)}, \sigma_2^{(m)}, \lambda_1^{(m)}, \lambda_2^{(m)})$. Go to Step 5.

Step 5: Output the descriptor $(\sigma_1^{(m)}, \sigma_2^{(m)}, \lambda_1^{(m)}, \lambda_2^{(m)})$ as the parameters of the departure process.

$$\alpha = \frac{\beta}{\zeta}, \quad \beta = \frac{2(\lambda_1^{(m)} - \zeta)(\zeta - \lambda_2^{(m)})}{B_d - 1} - \lambda_1^{(m)}\lambda_2^{(m)}. \tag{E.1}$$

$$\sigma_1^{(m)} = \frac{\alpha\lambda_1^{(m)} - \beta}{\lambda_1^{(m)} - \lambda_2^{(m)}}, \quad \sigma_2^{(m)} = \frac{\beta - \alpha\lambda_2^{(m)}}{\lambda_1^{(m)} - \lambda_2^{(m)}}. \tag{E.2}$$

$$\alpha = \frac{6(\zeta\xi - \eta)}{C_d\zeta(\zeta + \eta)^2 - 3C_{d,1}\zeta(\zeta + \eta) - 6(\xi + \xi^2)},$$

$$\beta = \zeta\alpha, \quad \gamma = \eta\alpha, \quad \delta = \zeta + \xi\alpha \tag{E.3}$$

$$\lambda_1^{(m)} = \frac{\delta + \sqrt{\delta^2 - 4\gamma}}{2}, \quad \lambda_2^{(m)} = \frac{\delta - \sqrt{\delta^2 - 4\gamma}}{2},$$

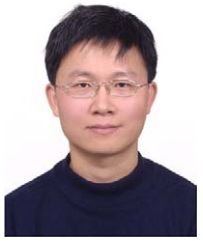
$$\sigma_1^{(m)} = \frac{\alpha\lambda_1^{(m)} - \beta}{\lambda_1^{(m)} - \lambda_2^{(m)}}, \quad \sigma_2^{(m)} = \frac{\beta - \alpha\lambda_2^{(m)}}{\lambda_1^{(m)} - \lambda_2^{(m)}}. \tag{E.4}$$

In (E.4), $\lambda_1^{(m)} \geq \lambda_2^{(m)}$ is assumed. For $\lambda_1^{(m)} < \lambda_2^{(m)}$, just switch the above solutions for $\lambda_1^{(m)}$ and $\lambda_2^{(m)}$.

References

[1] S. Aalto, W.R.W. Scheinhardt, Tandem fluid queues fed by homogeneous on-off sources, Operations Research Letters 27 (2000) 73–82.
[2] R.G. Addie, M. Zukerman, Queueing performance of a tree type ATM network, in: Proc. IEEE INFOCOM'94, June 1994, pp. 48–55.
[3] B. Avi-Itzhak, M. Yadin, A sequence of two servers with no intermediate queue, Management Science 11 (1965) 553–564.
[4] C. Blondia, The N/G/1 finite capacity queue, Communications in Statistics – Stochastic Models 5 (2) (1989) 273–294.
[5] J.A. Buzacott, J.G. Shanthikumar, Stochastic Models of Manufacturing Systems, Prentice Hall, New Jersey, 1993.
[6] H.W. Ferng, J.F. Chang, Connection-wise end-to-end performance analysis of queueing networks with MMPP inputs, Performance Evaluation 43 (2001) 39–62.
[7] H.W. Ferng, J.F. Chang, Departure processes of BMAP/G/1 queues, Queueing Systems 39 (2001) 109–135.
[8] H.W. Ferng, Modeling of split traffic under probabilistic routing, IEEE Communications Letters 8 (7) (2004) 470–472.
[9] W. Fischer, K.S. Meier-Hellstern, The Markov-modulated Poisson process (MMPP) cookbook, Performance Evaluation 18 (1993) 149–171.
[10] A. Gomez-Corral, A tandem queue with blocking and Markovian arrival, Queueing Systems 41 (2002) 343–370.
[11] Q.M. He, D.A. Stanford, Distributions of the interdeparture times in FCFS and nonpreemptive priority MMAP[2]/G[2]/1 queues, Performance Evaluation 38 (1999) 85–103.
[12] A. Heindl, Decomposition of general tandem queueing networks with MMPP input, Performance Evaluation 44 (2001) 5–23.
[13] P.G. Hoel, S.C. Port, C.J. Stone, Introduction to Probability Theory, Houghton Mifflin, Boston, 1971.
[14] O. Kella, Markov-modulated feedforward fluid networks, Queueing Systems 37 (2001) 141–161.
[15] H.S. Kim, N.B. Shroff, The notion of end-to-end capacity and its application to the estimation of end-to-end network delays, Computer Networks 48 (2005) 475–488.
[16] L. Kleinrock, Queueing Systems: Theory, Vol. I, Wiley, New York, 1975.
[17] V. Klimenok, L. Breuer, G. Tsarenkov, A. Dudin, The $BMAP/G/1/\bar{N} \to \cdot/PH/1/M$ tandem queue with losses, Performance Evaluation 61 (2005) 17–40.
[18] H. Kroner, M. Eberspacher, T.H. Theimer, P.J. Kühn, U. Briem, Approximate analysis of the end-to-end delay in ATM networks, in: Proc. IEEE INFOCOM'92, May 1992, pp. 978–986.

[19] D.M. Lucantoni, K.S. Meier-Hellstern, M.F. Neuts, A single-server queue with server vacations and a class of non-renewal arrival processes, Advances in Applied Probability 22 (1990) 676–705.
[20] D.M. Lucantoni, New results on the single server queue with a batch Markovian arrival process, Communications in Statistics – Stochastic Models 7 (1) (1991) 1–46.
[21] K. Mitchell, A. van de Liefvoort, Approximation models of feed-forward G/G/1/N queueing networks with correlated arrivals, Performance Evaluation 51 (2003) 137–152.
[22] E. Modiano, J.E. Wieselthier, A simple analysis of the average queueing delay in tree networks, IEEE Transactions on Information Theory 42 (2) (1996) 660–664.
[23] H. Naser, A. Leon-Garcia, A simulation study of delay and delay variation in ATM networks, in: Proc. IEEE INFOCOM'96, March 1996, pp. 393–400.
[24] E. Nyberg, J. Virtamo, S. Aalto, An exact end-to-end blocking probability algorithm for multicast networks, Performance Evaluation 54 (2003) 311–330.
[25] D. Park, H.G. Perros, H. Yamashita, Approximate analysis of discrete-time tandem queueing networks with bursty and correlated input traffic and customer loss, Operations Research Letters 15 (1994) 95–104.
[26] J.F. Ren, J.W. Mark, J.W. Wong, End-to-End performance in ATM networks, in: Proc. IEEE ICC'94, May 1994, pp. 996–1002.
[27] W. Stallings, High-Speed Networks and Internets — Performance and Quality of Service, 2nd edition, Prentice Hall, New Jersey, 2002.
[28] K. Sohraby, A. Privalov, End-to-end jitter analysis in networks of periodic flows, in: Proc. IEEE INFOCOM'99, March 1999, pp. 575–583.
[29] T. Suzuki, On a tandem queue with blocking, Journal of the Operations Research Society of Japan 6 (1964) 137–157.
[30] A.S. Tanenbaum, Computer Networks, 4th edition, Prentice Hall, 2003.



Huei-Wen Ferng was born in Taiwan in 1970. He received his B.S.E.E. degree from the National Tsing Hwa University, Hsinchu, Taiwan, in 1993 and his Ph.D. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 2000. Since August 2001, he has been with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, where he was an Assistant Professor from August 2001 to January 2005 and is currently an Associate Professor. Funded by the Pan Wen-Yuan Foundation, Taiwan, he spent the summer of 2003 visiting the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA. His research interests include teletraffic modeling, queueing theory, performance analysis, wireless networks, mobile computing, and high-speed networks.

Dr. Ferng is a member of the IEEE. He was a recipient of the research award for young researchers, Pan Wen-Yuan Foundation, Taiwan, in 2003.

Chi-Chao Chao received his B.S. degree in mathematics from the National Central University, Taoyuan, Taiwan, in 1998 and his M.S. degree in Computer Science and Information Engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 2001. His research interests include performance analysis and queueing theory.

Cheng-Ching Peng received his B.S. and M.S. degrees in Computer Science and Information Engineering from the Tatung University, Taipei, Taiwan, in 1997 and 1999, respectively. He is currently a Ph.D. candidate in the National Taiwan University of Science and Technology, Taipei, Taiwan. His research interests include network-wise performance analysis and performance analysis for cellular communication systems.