A Performance Modeling Scheme for Multistage Switch Networks With Phase-Type and Bursty Traffic

Ming Yu, *Senior Member, IEEE*, and Mengchu Zhou, *Fellow, IEEE*

***Abstract*—Existing analytical methods to model multistage switch networks cannot be applied to the performance modeling of switch networks with phase-type and bursty traffic because of the problem of state-space explosion and unrealistic assumptions, e.g., uniform traffic and independent destination (UTID). This paper presents an approximate scheme to model and analyze such networks. First, a traffic aggregation technique is proposed to deal with phase-type and bursty traffic, including splitting and merging. For the aggregation of two bursty traffic, a closed-form solution is obtained for buffer state probabilities. For the aggregation of more bursty traffic, a recursive algorithm is derived in terms of the buffer size and number of inputs of a switch. Second, a switch decomposition technique is developed, by which the crossbar of a switch is decomposed from its preceding and succeeding buffers. In this way, a switch network of N inputs and outputs is converted to N tandem queues, for which the performance can be easily evaluated. Our extensive numerical and simulation examples have shown that the proposed scheme achieves satisfied accuracy and computational efficiency.**

***Index Terms*—Finite buffer, performance modeling, switch network, traffic modeling.**

I. Introduction

With the advent of optical WDM technologies, we can use multistage switch architectures to build large switches for backbone networks to meet the increasing demand on traffic volume on order of terabits per second [1]. However, the performance modeling problem of multistage switches has not been adequately solved due to the diversity of the packet generation mechanism, inherently bursty nature of data traffic, and various switching stages the packets have to go through. This work aims to develop an approximate solution for the problem. It will help one gain analytical insight into the performance of multistage switches and assist simulation studies in practice.

For a packet-switch, there are three basic buffering strategies: input-, output-, and shared-buffering. Each has different advantages and drawbacks in terms of performance and feasibility [2]–[4]. Today's switches mainly use the input-buffering strategy with a nonblocking architecture due to its feasibility in building large size switches with high-speed links. Each input maintains a separate queue for each output, known as virtual output queue (VOQ), to avoid head-of-line (HOL) blocking. An example is shown in Fig. 1, where a buffer and crossbar (XB) are used to represent the buffering and switching procedures applied on the incoming packets, respectively. The scheduling is an algorithm for input-output matching, such as the parallel iterative matching (PIM) [4]–[6].

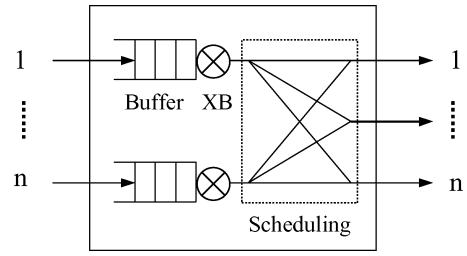

Fig. 1. An input-buffered switch with n inputs and n outputs.

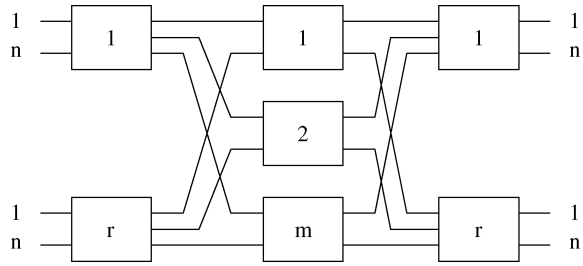

Fig. 2. A three-stage Clos network with r switches in the first and third stages while m in the second stage.

To construct a large-scale switch with high capacity, various numbers of switches at different stages are interconnected by using multistage interconnection networks. An example is shown in Fig. 2, where each of the switches can be an input-buffered one as shown in Fig. 1. Note that the switches in the intermediate stage can be unbuffered or buffered at both input and output [1], [7]. For simplicity, the input-buffered switch with a fast matching algorithm is referred to as a *switch*, while the constructed large-scale one is referred to as a *switch network*. Actually, the architecture shown in Fig. 2 is a three-stage Clos network. Other networks that can be similarly constructed are delta or banyan networks.

The performance modeling of switch networks has been a classical and difficult problem. The major difficulties are due to: 1) the lack of an appropriate description of the traffic aggregation processes; and 2) the *state-space explosion*. By the latter, we mean that the number of states needed to describe a switch network grows exponentially with the number of switches, switch size, and buffer size.

Manuscript received March 26, 2007; revised December 31, 2007 and July 31, 2008; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. G. Ajmone Marsan. First published December 11, 2009; current version published August 18, 2010.

M. Yu is with the Department of Electrical and Computer Engineering, Florida State University, Tallahassee, FL 32310 USA (e-mail: mingyu@eng.fsu.edu).

M. Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: zhou@njit.edu).

Digital Object Identifier 10.1109/TNET.2009.2036437

A. Related Work

Traditionally, many traffic models, in conjunction with constant service time, have been used in performance modeling of high-speed networks. The commonly used models can be categorized into three classes. The first one includes the simple Bernoulli processes and two-state Markov modulated Bernoulli process (MMBP), i.e., bursty traffic [8], [9]. The second one includes the classical Poisson processes and correlated Markov modulated Poisson process (MMPP) models [10], [11]. Since much of queueing theory revolves around exponential distribution that enables an analysis via Markov chains, it is a natural way to model a general distribution by a combination of exponential distributions, known as phase-type (PH) distribution [12]. Therefore, the third class is the traffic process described by PH distributions, i.e., PH traffic, which includes exponential distribution, Erlang distribution, mixture of generalized Erlang (MGE) distribution, i.e., Coxian phase-type distribution, and hyperexponential distribution.

For a switch, a simple method for performance modeling is to use a discrete time M/G/1 queue. The most common assumptions are: 1) uniform traffic (UT): packet arrivals to the switch are uniformly distributed over all the input links, and thus over all the outputs; 2) independent destination (ID): consecutive packets at each input link are independently assigned random destination addresses upon entry to the switch; and 3) buffers have infinite size. For simplicity, the first two are referred to as a UTID assumption. In this way, the multiple queues in a switch exhibit the same behavior. Thus, its performance can be analyzed by using a single queue. In [5] and [6], under the UTID assumption, for a switch with multiple input queues and finite buffer size, and i.i.d. Bernoulli or bursty traffic, a closed-form solution is found for the maximum throughput of the switch by using Markov chains. In [13], for an $N \times N$ switch with a shared buffer of finite size, the packets in the buffer are organized into N queues, each for an output. The N queues strongly depend on each other. Under the UT assumption, for i.i.d. Bernoulli and bursty traffic, the authors present an iterative aggregation method that combines two queues into one queue at a time until all the queues are combined into one block [13].

For a switch network, under the UTID assumption and Bernoulli traffic, the interstage traffic is often modeled by a Bernoulli process with traffic splitting and merging [9]. In this way, the states of the entire switch network are explicitly modeled, rather than the states of individual input or output queues in a stage. Thus, it can be referred to as a total network modeling (TNM) [2]. An iterative algorithm is developed to numerically solve for the state probabilities. The TNM method is extended to switch networks with buffers shared among all the inputs and outputs in a switch [14]–[16]. In [17], a network of M stages of $k \times k$ switches is modeled as a system of k queues working in parallel, with a deterministic service time for each queue. The derived steady-state queue length distribution is exact for the first stage and approximate for stage 2 and thereafter. In addition to the UTID assumption, another critical assumption made in [17] is that the interstage traffic is an independent Bernoulli process with a packet generation probability equal to the utilization of the stage.

Recently, it was found that the interstage traffic is less uniform than what is assumed by the existing models. Thus, the analysis based on the UTID assumption underestimates the packet loss performance [7]. For this reason, we assume that the heterogeneous traffic in switch networks, including the interstage traffic and aggregation processes, is of general distributions and described by PH distributions.

Cao and Towsley use the PH (or discrete Coxian) distributions to describe the batch traffic [18]. For a switch with infinite buffers, they develop a queueing model and find that such a switch can be well approximated by a queueing network of a closed product form. For switch networks with PH traffic, only those that have two and three stages and finite buffers with a single input can be modeled and solved analytically [19]. It is worth noting that a multistage buffered network with one input, i.e., a tandem queue model, is well investigated in production research by using the decomposition method (DM) [20]–[24]. In DM, the original network of machines and buffers is decomposed into a set of virtual two-machine tandem queues. The buffer states can be obtained by iteratively solving a set of equations, each describing a two-machine tandem queue.

In summary, the existing analytical methods can be applied to: 1) PH traffic, for only two- and three-stage switch networks with tandem queue models; 2) bursty traffic, for only single switches under the UTID or UT assumption. Without the assumptions of UTID and Bernoulli interstage traffic, switch networks cannot be solved analytically.

B. Contribution of This Work

First, for PH and bursty traffic, we propose to recursively aggregate (disaggregate) all the input (output) traffic to (from) a buffer into one input by using merging (splitting). In this way, the number of states in buffer modeling is significantly reduced, and a switch network is regrouped into multiple tandem queues. Second, for the aggregation of two bursty traffic inputs, we find an exact solution for the states of the aggregation buffer, which can be recursively used to aggregate more bursty traffic inputs. Third, we develop an approximate switch decomposition technique, by which the XB of a switch is decomposed from its preceding and succeeding buffers. Based on the conservation-of-flow principle, a set of nonlinear equations is formed and solved for the buffer states. In this way, the performance of switch networks can be modeled without the commonly used UTID assumption, and the state-space explosion problem is significantly reduced.

C. Organization of the Paper

Section II presents a traffic aggregation technique for PH traffic and an exact solution for the aggregation of two bursty traffic streams. Section III presents a new decomposition technique to model switch networks with finite buffers. The performance metrics such as throughput and delay are then evaluated in Section IV. Numerical and simulation examples are given to illustrate the proposed methods in Section V. Concluding remarks are presented in Section VI.

II. Traffic Aggregation

A typical switch network is shown in Fig. 2. It is a three-stage Clos network $C(n, m, r)$, where r is the number of switch modules in the first and third stages; m is the number of switch modules in the middle stage; and n is the number of inputs (outputs) of the switch in the first (third) stage [3]. The network size is $N = n \times r$. A typical input-buffered switch is shown in Fig. 1,

which has n input and n output ports, respectively. The size of a buffer B_i for input i in a switch is assumed to be finite and denoted as K_i. We assume that a packet has a unit size and is processed in a time slot. If the number of packets that need to be buffered is larger than K_i, then some of the packets have to be dropped. In this work, we assume that the buffer management algorithm drops the newly arrived packets when the buffer is full. Note that an XB can hold a packet being switched. It is assumed that the XB has a buffer size of one packet for modeling purpose.

A. Aggregation of PH Traffic

A PH distribution can be considered as the distribution of the packet service time until absorption in a Markov chain with a single absorbing state. More precisely, a PH distribution is the distribution of time X till absorbing state 0 in a Markov chain of the states $\{0, 1, \ldots, n\}$ with its initial probability vector

$$(\tau_0, \boldsymbol{\tau}) \tag{1}$$

where $\boldsymbol{\tau}$ is a row vector of size n, $\tau_0 = 1 - \boldsymbol{\tau}\mathbf{1}$; an infinitesimal generator

$$\begin{bmatrix} 0 & \mathbf{0} \\ \boldsymbol{t} & T \end{bmatrix} \tag{2}$$

where $\boldsymbol{t}$ is a column vector of size n; T is an $n \times n$ matrix, $\boldsymbol{t} = -T\mathbf{1}$; $\mathbf{0}$ is a row vector of 0's, and $\mathbf{1}$ is a column vector of 1's. The PH distribution is denoted by $PH(\boldsymbol{\tau}, T)$. The density function [25] is

$$f(x) = \boldsymbol{\tau} e^{Tx} \boldsymbol{t}, \quad \text{for } x > 0. \tag{3}$$

The moments [25] are

$$E[X^k] = k! \boldsymbol{\tau} (-T^{-1})^k \mathbf{1}, \quad \text{for } k \geq 1. \tag{4}$$

For the aggregation of the heterogeneous traffic streams in a switch network, PH distributions have been extensively used to approximate the distribution of the aggregated process through a moment-matching method [12], [26]. The structure and parameters of the approximate PH distribution depend on the *squared coefficient of variation*. For an r.v. X, the squared coefficient of variation, denoted by ξ, is defined as the ratio of its variance to its squared mean of X [27]

$$\xi = \frac{Var[X]}{E^2[X]} = \frac{E[X^2]}{E^2[X]} - 1. \tag{5}$$

It can be seen that $\xi \in [0, \infty)$; ξ is close to zero (low variability) if X assumes values that are close to each other.

For the merge of two PH traffic processes X_1 and X_2, denoted by $PH(\boldsymbol{\tau}_1, T_1)$ and $PH(\boldsymbol{\tau}_2, T_2)$, respectively, the aggregated traffic proces, X is also a PH traffic denoted by $PH(\boldsymbol{\tau}, T)$. Because $X = X_1 + X_2$, the distribution of X is the convolution of those of X_1 and X_2. Thus, we have [25]

$$\boldsymbol{\tau} = [\boldsymbol{\tau}_1, \tau_{1,0}\boldsymbol{\tau}_2] \tag{6}$$

and

$$T = \begin{bmatrix} T_1 & \boldsymbol{t}_1 \cdot \boldsymbol{\tau}_2 \\ 0 & T_2 \end{bmatrix} \tag{7}$$

where $\tau_{1,0} = 1 - \boldsymbol{\tau}_1\mathbf{1}$; $(\boldsymbol{t}_1 \cdot \boldsymbol{\tau}_2)_{ij} = t_1(i)\tau_2(j)$, while $t_1(i)$ and $\tau_2(j)$ are the ith and jth components of $\boldsymbol{t}_1$ and $\boldsymbol{\tau}_2$, respectively.

For the splitting of a PH traffic process X, denoted by $PH(\boldsymbol{\tau}, T)$, we assume that

$$T = \begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix}. \tag{8}$$

If X is split into X_1 and X_2, with probability q and $1 - q$, respectively, then we can find that X_1 and X_2 are also of PH distributions, that is, $PH(\boldsymbol{\tau}_1, T_1)$ and $PH(\boldsymbol{\tau}_2, T_2)$, respectively, where [25]

$$\left.\begin{aligned} \boldsymbol{\tau}_1 &= \boldsymbol{\tau}(1 : n_1)/q \\ \boldsymbol{\tau}_2 &= \boldsymbol{\tau}(n_1 + 1 : n)/(1 - q) \end{aligned}\right\} \tag{9}$$

where $\boldsymbol{\tau}(1 : n_1)$ and $\boldsymbol{\tau}(n_1 + 1 : n)$ stand for the elements of $1 \sim n_1$ and $n_1 + 1 \sim n$ of $\boldsymbol{\tau}$, respectively. The above splitting requires two assumptions. First, the Markov chain of the states $\{1, 2, \ldots, n\}$ has two disjoint subsets, that is, $\{1, 2, \ldots, n_1\}$ and $\{n_1 + 1, n_1 + 2, \ldots, n\}$. Otherwise, we need to separate the Markov chain by using balance equations before conducting the splitting [25]. Second, q is given by the scheduling mechanism of a switch [4], [5].

One common phase-type distribution is MGE, which is often called the Coxian distribution. It is used to model the mixtures of exponential, hyperexponential, and Erlang distributions. As an example, for the mixture of two exponential distributions, in terms of (3), the density function of the MGE distribution is

$$f(x) = c_1\mu_1 e^{-\mu_1 x} + c_2\mu_2 e^{-\mu_2 x}, \quad x \geq 0$$

where $c_1 = (\mu_1(1 - a_1) - \mu_2)/(\mu_1 - \mu_2)$ and $c_2 = 1 - c_1$ with $\mu_1 \neq \mu_2$, μ_1 and μ_2 being the two exponentially distributed mean service rates, and a_1 is a state transition probability that describes a two-state Markov chain. In terms of (4) and (5), we find

$$E[X] = \frac{1}{\mu_1} + \frac{a_1}{\mu_2} \tag{10}$$

$$\xi = 1 - \frac{2a_1\mu_1\left[\mu_2 - \mu_1(1 - a_1)\right]}{(\mu_2 + a_1\mu_1)^2}. \tag{11}$$

The above MGE distribution is a two-phase distribution and is denoted as $\mathrm{MGE}{-}2(\mu_1, \mu_2, a_1)$. Similar results can be obtained for the aggregation of two phase-type distributions.

Another common PH distribution is Erlang. The density function of an Erlang distribution with n phases with rate parameter λ, denoted by Erlang-$n(\lambda)$, is [25]

$$f(x) = \frac{\lambda^n x^{n-1} e^{-\lambda x}}{(n-1)!}, \quad \text{for } x > 0. \tag{12}$$

It has a mean of n/λ and variance n/λ^2, respectively.

In principle, the Laplace–Stieltjes transform (LST) of any distribution function can be approximated arbitrarily closely by a rational function. Therefore, PH distributions can be used to model any aggregation processes. In order to use PH distributions, many efforts have been devoted to determine the structure and parameters by various methods, including moment-

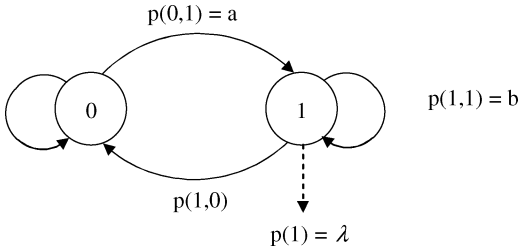

Fig. 3. A bursty traffic model.

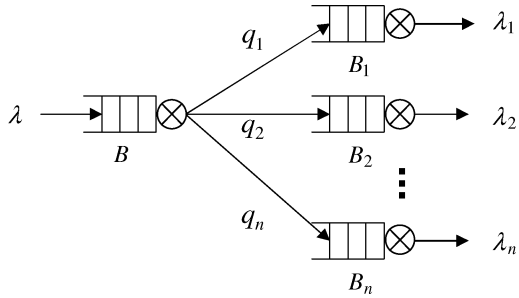

Fig. 4. Traffic splitting.

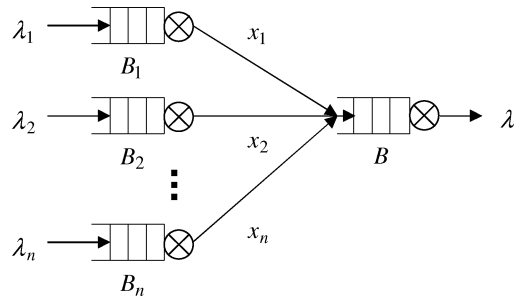

Fig. 5. Traffic merging.

matching methods [12]. In this work, we assume that heterogeneous traffic processes are described by phase-type distributions. Thus, we can find the parameters for traffic aggregation processes, similar to (10) and (11).

It is worth pointing out that in order to accurately model the correlation properties of a traffic aggregation process, we can fit the traffic into correlated traffic models such as MMPP [11]. In this work, we focus on switch networks with PH and bursty traffic. We leave the cases of correlated and mixed traffic as a future research topic.

B. Aggregation of Bursty Traffic

Denote by $B(\lambda,\gamma)$ the bursty traffic model, which is governed by a two-state Markov model shown in Fig. 3. At an input port, a packet is assumed to be generated when the underlying Markov chain is in state 1. Otherwise, in state 0, there will be no arriving packet at this port in the current time slot. In this model, γ is the burstiness [8] defined as

$$\gamma \triangleq p(1,1) - p(0,1) \tag{13}$$

where $p(i,j)$ is the transition probability from state i to j, $i,j \in \{0,1\}$. By defining $a = p(0,1)$ and $b = p(1,1)$, we have

$$\gamma = b - a.$$

The average length of the periods in state 1 is $1/(1-b)$, i.e., the burst length. The length in state 0 is $1/a$. Thus, the average offered load is

$$\lambda = a/(a+1-b)$$

which is the mean arrival rate of an incoming Bernoulli process. Combining the above two expressions, we find

$$\left.\begin{aligned} a &= \lambda(1-\gamma) \\ b &= \lambda+\gamma-\lambda\gamma \end{aligned}\right\}. \tag{14}$$

Therefore, $B(\lambda,\gamma)$ can be also described by a and b, as discussed in the Appendix.

For an independent splitting of $B(\lambda,\gamma)$, as shown in Fig. 4, each packet is routed along a tagged direction with a splitting probability. The splitting process is exactly described by an MMBP, denoted by $B(\lambda_d,\gamma_d,q_d)$, $d=1,2,\ldots,n$, where $\lambda_d = \lambda q_d$, $\gamma_d = \gamma$, and q_d is the splitting probability. The splitting process [9] can be approximated by $B(\lambda_d,\gamma_d)$ with

$$\left.\begin{aligned} \lambda_d &= q_d\lambda \\ \gamma_d &= q_d\gamma(1-\lambda)/(1-q_d\lambda) \end{aligned}\right\}. \tag{15}$$

Similar results are found for a correlated splitting of the bursty traffic $B(\lambda,\gamma)$ by using a three-state MMBP model [9].

The traffic merging process is more complicated. As shown in Fig. 5, n processes $B(\lambda_d,\gamma_d)$, $d=1,2,\ldots,n$, are merged in a buffer B with size of K. The merged traffic stream is described by $B(\lambda,\gamma)$. Denote by $x=(x_1,x_2,\ldots,x_n)$ the joint state of the n traffic arrivals, where $x_d=1$ if queue d transmits a packet, and otherwise, $x_d=0$, for $d=1,2,\ldots,n$. Denote by $p(j;x)$ the steady-state joint probability that there are j packets in B when it is in state x. Note that $j=K+1$ means that the buffer is full and the succeeding XB holds one packet currently being worked on. The proceeding XB is said to be in a blocking state. Then, the newly arrived packets to the buffer will be dropped, as analyzed in the Appendix.

Denote by p_0 the probability that the output buffer is empty. By applying the concept of the conservation-of-flow, we have

$$\lambda = 1 - p_0. \tag{16}$$

Upon substituting $a = p(0,1) = 1-p(0,0)$ and $b = \gamma + a = \gamma + 1 - p(0,0)$ into a local balance equation $ap_0 = (1-b)(1-p_0)$, we find

$$\gamma = \left(p(0,0) - p_0\right)/(1-p_0).$$

By using the boundary condition (e.g., (A.21) in the Appendix), we have $p(0;\overbrace{00\cdots0}^{n \text{ terms}}) = p_0\bar{a}_1\bar{a}_2\cdots\bar{a}_n$, where $\bar{a}_d = 1-a_d$, $d=1,2,\cdots,n$, which is $p_0p(0,0)$. Thus, $p(0,0) = \bar{a}_1\bar{a}_2\cdots\bar{a}_n$. We find the equivalent parameters

$$\left.\begin{aligned} \lambda &= 1-p_0 \\ \gamma &= (\bar{a}_1\bar{a}_2\cdots\bar{a}_n - p_0)/(1-p_0) \end{aligned}\right\}. \tag{17}$$

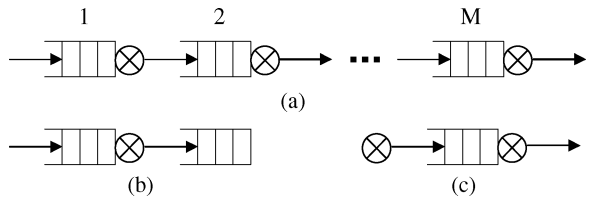

Fig. 6. (a) A switch network in tandem queue model, (b) a decomposed switch, and (c) a decomposed buffer.

In the Appendix, we present a recursive algorithm to find the exact solution of p_0. For the case of $d = 2$, the solution is summarized in the following theorem.

Theorem 1: The probability that the output buffer is empty is given by

$$p_0 = p(0;00)/(\bar{a}_1\bar{a}_2) \tag{18}$$

where

$$p(0;00) = \frac{1}{e\left[(I - R_{21})^{-1}(\Psi_1 + \Psi_2)\right]_{\text{col},1}} \tag{19}$$

where $[\cdot]_{\text{col},1}$ means the first column of the inside matrix; e is an all-1 row vector; R_{21} is given in (A.16); and Ψ_1 and Ψ_2 are given in (A.19). Note that R_{21}, Ψ_1, and Ψ_2 only depend on parameters of the input traffic, i.e., a_d and b_d, $d = 1, 2$. ■

For the cases of $d \geq 3$, we can use the exact solutions in the Appendix to find the parameters of the aggregated traffic, as in (17). This method is named the exact traffic aggregation (ETA) method. By using the results from the aggregation of two traffic streams, we can also iteratively aggregate two streams into one at a time until all the d streams are aggregated into an equivalent one, for which we can find the state of the output buffer. This method is named the approximate traffic aggregation (ATA) method. Mathematically, the latter is more tractable than the former, particularly when the buffer is finite but of relatively large size, which cannot be handled by any existing methods to the authors' knowledge.

III. Switch Decomposition Method

After traffic aggregation, a switch with n input and output ports can be regrouped into n queues. Accordingly, a switch network with M stages can be regrouped into n tandem queues, with each queue representing a concerned input–output route as shown in Fig. 6.

This work considers two flow control mechanisms: global and local [2]. With the former, the XB of a switch allows its predecessor to send it a packet if it has one empty buffer slot currently or if one of the packets in its buffer leaves in the current cycle. With the latter, the XB allows its predecessor to send a packet only if its buffer has an empty slot.

A. Switch Decomposition

Consider a tandem queue in Fig. 6(a), where the buffers and XBs are numbered as $B_1, XB_1, B_2, XB_2, \ldots, B_M$, and XB_M. Denote by $p_i(j;x)$ the steady-state joint probability that there are j packets in the buffer at stage i, where x is the state of the traffic arrivals.

For traffic merging, as shown in Fig. 5, the buffer for merged traffic has n traffic inputs from the XBs in its previous stage. To simplify the notation, let p_{ij} denote $p_i(j;x)$. Thus, p_{i0} and p_{i,K_i+1}, are the probability for $j = 0$ and $j = K_i + 1$, i.e., B_i is empty and full, respectively.

In order to find the relationship between the state of an XB and its buffers, we denote by α_i the probability that a packet is available to enter B_i, as in [2]. Here, we use an overline to indicate the complement of a probability, e.g., $\bar{\alpha}_i = 1 - \alpha_i$ and $\bar{p}_{ij} = 1 - p_{ij}$. By its definition, we can find

$$\begin{aligned}\alpha_i &= 1 - p_{i-1}(0,\ldots,0;x_1,x_2,\ldots,x_n)\\ &= 1 - p_{i-1}(0,\ldots,0)\\ &= \bar{p}_{i-1,0}.\end{aligned} \tag{20}$$

The reason is that if at least one of the preceding buffers is not empty, then a new packet is available to enter B_i.

The case that meets the UTID assumption (see [2]) can be treated as a special case of (20). Here, α_i can be calculated based on one buffer and thus (20) can be simplified to

$$\alpha_i = 1 - (1 - \bar{p}_{i-1,0}/n)^n. \tag{21}$$

The reason is that a packet is available to enter an input buffer of a switch at stage i if at least one of the n buffers of the predecessor is nonempty and has a first packet for the particular switch at stage i. Note that the states of the n buffers are assumed to be independent.

For traffic splitting, as shown in Fig. 4, the buffer in stage i has n successors in stage $i+1$, with their buffer size $K_1, K_2, \ldots$, and K_n. We also use $x = (x_1, x_2, \ldots, x_n)$ to represent the joint state of the n traffic outputs. Let β_i be the probability that a successor of the XB at stage i can accept a packet. Under local flow control, we can find

$$\begin{aligned}\beta_i &= 1 - p_{i+1}(K_1+1, K_2+1, \ldots, K_n+1; x_1, x_2, \ldots, x_n)\\ &= 1 - p_{i+1}(K_1+1, \ldots, K_n+1)\\ &= \bar{p}_{i+1,K_{i+1}+1}.\end{aligned} \tag{22}$$

The reason is that if at least one of the successors is not being blocked, then it can accept a packet.

Under global flow control, we need to consider one more stage. The probability that the first packet in a buffer in stage $i+1$ can leave during a given cycle equals the probability that its successor's buffer in stage $i+2$ is not full when the buffer in stage $i+1$ is full. Thus, we have

$$\beta_i = \bar{p}_{i+1,K_{i+1}+1} + \bar{p}_{i+2,K_{i+2}+1}p_{i+1,K_{i+1}+1}. \tag{23}$$

The reason is that even if the successor's buffer is full, it can still accept a packet if there is a packet leaving the successor for its next stage during the same cycle.

From the viewpoint of an XB, it is said to be in a starving state when its preceding buffer is empty. It is in a blocking state when its succeeding buffer is full. Only when its preceding buffer is not empty and succeeding one is not full can the XB work properly. We denote by μ_i the service rate of an XB at stage i, the actual rate at which the packets are leaving XB_i and entering B_{i+1} is

$$\beta_i\alpha_{i+1}\mu_i = \bar{p}_{i,0}\bar{p}_{i+1,K_{i+1}+1}\mu_i. \tag{24}$$

Now, it is clear that if the state of a buffer, i.e., p_{ij}, is known, then the state of its XBs can be obtained. This allows us to express the state of a buffer as a function of one parameter only as follows.

Consider a buffer B_i, $i = 1, 2, \ldots, M$. Denote by ω_i the actual arrival rate to B_i in the steady state. The offered load is defined as

$$\rho_i \triangleq \omega_i/\mu_i \tag{25}$$

or equivalently

$$\omega_i = \rho_i \mu_i \tag{26}$$

in which ρ_i is unknown and used as a to-be-determined parameter. If XB_i has both input and output buffer, an output buffer B_i is followed by an input buffer B_{i+1}, instead of XB_{i+1}. Then, μ_i in (25) is replaced by ω_{i+1}. We need to express the state of buffers in the form of $p_{ij}(\rho_i)$.

The throughput of an XB can be defined as a leaving rate, at which the packets are processed and output to a next stage. For an XB at stage i of the tandem queue, the throughput is

$$\vartheta_i \triangleq \beta_i \alpha_{i+1} \mu_i. \tag{27}$$

In terms of (24), we have

$$\vartheta_i = \bar{p}_{i,0} \bar{p}_{i+1,K_{i+1}+1} \mu_i, \quad i = 1, 2, \ldots, M-1 \tag{28}$$

where M is the number of stages in the switch network.

For a tandem queue with M stages, as shown in Fig. 6(a), it is assumed that B_1 is always nonempty, i.e., with saturated input, and the buffer after XB_M always has space to accept packets, i.e., $p_{10} = 0$ and $p_{M,K_M+1} = 0$. Thus, there are $M-1$ buffers whose states need to be determined. In the steady state, the number of packets processed by the XBs at each stage should observe the conservation of flow. Thus, we obtain the following set of equations:

$$\vartheta_{i+1} = \vartheta_i, \quad i = 1, 2, \ldots, M-1. \tag{29}$$

If we can express the buffer state in a form of $p_{ij}(\rho_i)$, then by substituting (28) into (29), we can obtain $M-1$ equations and thus solve them for the $M-1$ parameters, $\rho_i, i = 1, 2, \ldots, M-1$.

Otherwise, for example, for a bursty traffic with fixed service time, in terms of (A.23) in the Appendix, we can choose p_{io} as the buffer parameter at a stage i since all the p_{ij} can be obtained by using p_{i0}. For a tandem queue with M stages, in terms of (16), for stage $i+1$, we have

$$\lambda_{i+1} = 1 - p_{i0}, \quad i = 1, 2, \ldots, M-1. \tag{30}$$

Therefore, the traffic parameters for the next stage, i.e., λ_{i+1} and γ_{i+1}, can be obtained in terms of (17).

B. Buffer Modeling

Consider a buffer in stage i with its predecessor and successor XBs, as shown in Fig. 6(c). Note that each buffer has only one traffic input after the splitting and merging operations as discussed in Section II.

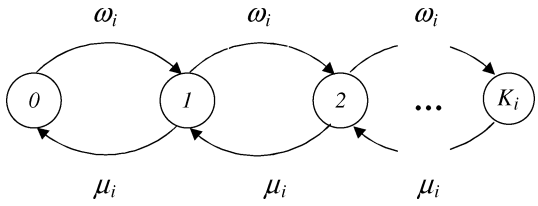

Fig. 7. A buffer state transition diagram.

It is well known that, for Poisson arrivals, the buffer can be exactly modeled as an M/M/1 queue if we choose ω_i as the average arrival rate, which is defined in (26). The buffer state transition diagram is shown in Fig. 7. In terms of flow balance equations, the following results can be easily obtained:

$$\left.\begin{aligned} p_{ij}(\rho_i) &= p_{i0}\rho_i^j \\ p_{i0}(\rho_i) &= (1-\rho_i)/\left(1-\rho_i^{K_i+2}\right) \\ p_{i,K_i+1}(\rho_i) &= \rho_i^{K_i+1}(1-\rho_i)/\left(1-\rho_i^{K_i+2}\right) \end{aligned}\right\} \tag{31}$$

to model a buffer with Poisson arrivals and exponential service times.

For PH traffic, including MGE-n and Erlang-n distribution, we can use (31) to approximately model the buffer states. The reason is that the rates of traffic that arrive at and departure from an M/M/1 buffer can be also interpreted as the mean rates of packet arrival and departure. Thus, for PH traffic process X, the traffic arrival rate is $\lambda_i = 1/E[X]$, which is given by (10) and (11). If the service time of a switch is of PH distribution, e.g., $X \sim$ PH distribution, then we can also use (10) and (11) to calculate the mean service time, that is, $\mu_i = 1/E[X]$. For traffic of Erlang-n distributions, similar results can be obtained by using (12), as shown in the examples.

Once the buffers have been modeled by (31), by substituting (31) into (29), we can numerically solve for the buffer parameters ρ_i, $i = 1, 2, \ldots, M-1$, for a tandem queue with M stages. Subsequently, all the states $p_{ij}(\rho_i)$, $j = 0, 1, 2, \ldots, K_i$, $i = 1, 2, \ldots, M-1$, can be obtained.

For bursty traffic, we propose to aggregate all the traffic inputs to a buffer into one traffic by using the ATA method. The buffer state can be easily obtained by simplifying (A.7) and (A.21) (we omit the stage number i for simplicity)

$$\left.\begin{aligned} P_0 &= (I-\pi_0)^{-1}\pi_0 P_0 \\ P_1 &= \left[(I-\pi_1) - \pi_1(I-\pi_0)^{-1}\pi_0\right]^{-1}\pi_0 P_2 \\ P_j &= (I-\pi_1)^{-1}\pi_0 P_{j+1}, \text{ for } j = 2, \ldots, K, \end{aligned}\right\} \tag{32}$$

where π_0 and π_1 have the first and second column of π, respectively, while the other column is zero

$$\pi = \begin{bmatrix} \bar{a} & \bar{b} \\ a & b \end{bmatrix}$$

where $a = \lambda(1-\gamma)$, $b = \lambda(1-\gamma) + \gamma$, and

$$P_j = [p(j;0), p(j;1)]^T$$

and P_{K+1} is the probability that the buffer is in the blocking state. Define a row vector of ones: $e \triangleq [1, 1]$, we have

$$p_j = eP_j. \tag{33}$$

By applying the normalization condition, $\sum_{j=0}^{K+1} eP_j = 1$, to (32), we have

$$[v_1, v_2]P_{K+1} = 1 \tag{34}$$

where $[v_1, v_2] = e[(I+D_0)D_1D^{K-1} + (I-D^K)(I-D)^{-1}]$, $D_0 = (I-\pi_0)^{-1}\pi_0$, $D_1 = [(I-\pi_1) - \pi_1(I-\pi_0)^{-1}\pi_0]^{-1}\pi_0$, and $D = (I-\pi_1)^{-1}\pi_0$. Thus

$$P_{K+1} = \begin{bmatrix} 0 \\ \frac{1}{v_2} \end{bmatrix} - \begin{bmatrix} 1 \\ \frac{v_1}{v_2} \end{bmatrix} p(K+1;0). \tag{35}$$

By applying the boundary condition in (A.21), i.e., $P_0 = p_0[\bar{a}, a]^T$, to $P_0 = D_0P_1 = D_0D_1D^{K-1}P_{K+1}$, we find

$$\begin{bmatrix} p_0 \\ p(K+1;0) \end{bmatrix} = \begin{bmatrix} \bar{a} & s_1 \\ a & s_2 \end{bmatrix}^{-1} D_0D_1D^{K-1} \begin{bmatrix} 0 \\ \frac{1}{v_2} \end{bmatrix} \tag{36}$$

where

$$\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = D_0D_1D^{K-1} \begin{bmatrix} 1 \\ \frac{v_1}{v_2} \end{bmatrix}.$$

Accordingly, we can find P_{K+1} in terms of (35), and p_j, $j = 1, 2, \ldots, K+1$, in terms of (33) and (32).

For a tandem queue with M stages, the traffic parameters are given only for the first stage. For the second stage and thereafter, by substituting (36) into (30), we can find the parameters λ_{i+1} and also γ_{i+1} in terms of (17).

It is worth noting that, for an infinite buffer size, the buffer states can be easily obtained by using the generating function method [8], [9]. The difficult case is the buffer with a finite but relatively large size because there needs a large dimension to describe the buffer state transition matrix.

IV. Performance Analysis

Based on the proposed switch decomposition technique, we can easily calculate the performance metrics as defined in [2].

The average number of packets in the network can be calculated straightforwardly as [2]

$$\kappa_{\text{pkt}} = \sum_{i=1}^{M} \sum_{j=1}^{K_i+1} jp_{ij}. \tag{37}$$

The average delay through the network can be calculated by summing the average delays at each stage, which can be obtained by using Little's Law. For a network using a global control strategy, the delay is [2]

$$\tau_i = \frac{1}{\alpha_i\,(1 - p_{i,K_i+1}\bar{q}_i)} \sum_{j=0}^{K_i+1} jp_{ij} \tag{38}$$

where the quantity in the denominator of the initial fraction is the average arrival rate at stage i and the summation is the average queue length. If the local control strategy is used, we just substitute $\alpha_i\bar{p}_{i,K_i+1}$ for the expression in the denominator of (38) to obtain the delay [2]

$$\tau_i = \frac{1}{\alpha_i\bar{p}_{i,K_i+1}} \sum_{j=0}^{K_i+1} jp_{ij}. \tag{39}$$

The throughput can be directly evaluated by using one of the stage

$$\vartheta_i = (1-p_{i0})(1-p_{i+1,K_{i+1}+1})\mu_i. \tag{40}$$

The packet loss ratio for each stage can be calculated by

$$\eta_i = \frac{\lambda_i - \vartheta_i}{\lambda_i}. \tag{41}$$

The entire algorithm can be outlined as follows:

1) Find the parameters of the traffic models. They are given in (1) and (2) for PH traffic and (14) for bursty traffic.
2) Aggregate the multiple traffic inputs into one stream. They are given in (6)–(9) for PH traffic and (15) and (17) for bursty traffic.
3) Solve the tandem queues for buffer parameters. For PH traffic, numerically solve (29) for ρ_i. For bursty traffic, the solution is given in (36).
4) Calculate buffer states. They are given in (31) for PH traffic and (32) for bursty traffic.
5) Evaluate network performance according to (38)–(41).

The computational advantage of the proposed technique can be analyzed in two cases. Here, we assume that a network has M stages of switches, each switch has N inputs and N outputs, and each input has a buffer of size K.

1) PH Traffic: The number of states needed for a Markov chain to describe a buffer is 2^{K+1}. The number of states to describe a switch is $\overbrace{2^{K+1}\cdots 2^{K+1}}^{N \text{ terms}} = 2^{(K+1)N}$. The number of states to describe the network is $\overbrace{2^{(K+1)N}\cdots 2^{(K+1)N}}^{M-1 \text{ terms}} = 2^{(K+1)N(M-1)}$, which equals 2^{704} for a typical network: $N = 16$, $M = 5$, and $K = 10$, leading to a state-space explosion problem.

Using the proposed method, a buffer only needs to be identified as empty, full, or not empty and not full. Only one parameter (e.g., ρ) is needed to describe the three states, which can be numerically solved from a set of $M-1$ equations, each is a $(K+1)$-th order algebraic equation, see (31), which describes the underlying Markov chain of $K+1$ states and can be solved with a time of $O((K+1)^3)$. Note that in a tandem queue model, after switch decomposition, an XB's state is only related to its neighboring buffers' states, which are governed by a set of algebraic equations (e.g., the principle of flow conservation). Therefore, a buffer needs only one state to be modeled. All other states of the buffer can be derived from the buffer parameter. Therefore, we only need to determine the $M-1$ buffer states for a tandem queue with a time of $O((M-1)(K+1)^3)$. The network is converted into N tandem queues after traffic aggregations. Therefore, the time complexity is $O(NMK^3)$ to analyze the network performance.

2) Bursty Traffic: To exactly describe the merge of N bursty traffic streams, we need 2^N states, which is the dimension of the transition matrix (see the Appendix). For a buffer size of K, we need to solve for P_j, $j = 0, \ldots, K$ to determine the buffer states, where P_j is a vector of dimension 2^N. Therefore, we need $(K+1)2^N$ states to describe a buffer. For a tandem queue of M stages, we need $M(K+1)2^N$ states. Since the network is converted into N tandem queues, we need $NM(K+1)2^N$ states to describe the network if the ETA method is chosen for

traffic aggregations. Clearly, the state-space explosion problem is scaled down significantly but not eliminated.

As we mentioned in the previous section, we can choose the ATA method in traffic aggregations. By aggregating two traffic streams at a time, which costs a fixed amount of time (see Theorem 1), we need $N-1$ times to aggregate N traffic streams into one. Each time only a buffer parameter needs to be solved. Therefore, we need a total time of $O(NM(K+1)(N-1)2^2)$, i.e., $O(MKN^2)$, to evaluate the network performance, which is a polynomial function of the network sizes.

Based on the above discussion, we summarize the computational complexity in the following theorem.

Theorem 2: The proposed algorithm has the complexity of $O(NMK^3)$ for PH traffic. For bursty traffic, it has the complexity of $O(NMK2^N)$ with the ETA method and $O(MKN^2)$ with the ATA method, respectively. ■

Typically, a nonblocking switch network with size $N = n \times r$ is constructed from switch blocks with size $n \times n$, where $n = 2 \sim 16$ and $r = 2 \sim 16$. In practice, if one parameter is large, then the other one is small, e.g., if $n = 16$, then $r = 2 \sim 4$. Thus, usually $N = 8 \sim 64$. For these typical sizes, the exact performance modeling becomes mathematically intractable. Consequently, the approximate performance modeling that is often assisted by simulation studies has become the major approach. In terms of the above theorem, the proposed scheme can be used with simulations to model the typical switch networks, such as Clos, banyan, and delta networks.

V. Numerical Examples and Simulation Results

In this section, we verify the proposed method by several examples. The analysis and simulation are conducted by using Matlab ver. 7.0 release 14 to find numerical results by following the algorithm outlined in Section IV. For PH traffic, including MGE and Erlang distributions, the first three examples demonstrate the accuracy of our method when applied to the two-, three-, and five-stage networks and with small, medium, or large buffers. The results are compared to those by using the decomposition method (DM) [20]–[24]. For bursty traffic, the fourth example is to demonstrate the effectiveness of the proposed method when applied to a typical Clos network without meeting the UTID assumption. The results are compared to those obtained by simulations on a Dell PowerEdge 2850 server with two Intel Xeon CPUs at 2.8 GHz and 4 GB memory.

Example 1: Consider a tandem queue model, as shown in Fig. 6, which consists of two switches and a buffer in between them. The packet arrivals have an MGE-2 distribution: $(\lambda_1, \lambda_2, a_1) = (2.0, 1.0, 0.25)$. The service times of the two switches have the following MGE-2 distributions: $(\mu_1, \mu_2, a_2) = (2.0, 1.0, 0.25)$ and (2.5,2.0,0.5), respectively. The buffer size is $K = 3$.

To model this simple tandem queue, we assume that the first switch is never starved and the second one is never blocked. By using (10) and (11) to find the traffic and service rate, and substituting (31) into (29), we obtain a set of equations with the buffer parameter as the unknown, which can be easily solved for the buffer states. The results are shown in Table I, which also lists the results by using DM method. Define an error, $e \triangleq p_j - \hat{p}_j$, and a relative error, $e_r \triangleq |p_j - \hat{p}_j|/p_j$, where p_j is the value obtained from DM, and $\hat{p}_j$ is the value from this work. The relative errors on the empty, blocking probability, and throughput are about 0.27%, 0.54%, and 0.09%, respectively. We also notice that one of the intermediate results, p_3, has a relative error 12.01%. This is due to the approximation of buffer modeling in our method, which emphases the accuracy of only those special buffer states that directly impact the performance results, i.e., p_0 and p_{K+1}.

TABLE I
MGE-2 Distribution With $K = 3$

	DM	This Work	Error (%)
p_0	0.2616	0.2609	0.2676
p_1	0.2159	0.2261	4.7244
p_2	0.1960	0.1960	0
p_3	0.1757	0.1968	12.0091
p_{K+1}	0.1480	0.1472	0.5405
ϑ	1.1360	1.1371	0.0968

TABLE II
MGE-2 Distribution With $K = 30$

	DM	This Work	Error (%)
p_0	0.13481	0.13472	0.0668
p_1	0.11180	0.11675	4.4275
p_2	0.09929	0.10119	1.9136
p_5	0.06592	0.06587	0.0758
p_{10}	0.03264	0.03221	1.3174
p_{15}	0.01649	0.01575	4.4876
p_{20}	0.00799	0.00770	3.6295
p_{25}	0.00396	0.00377	4.7980
p_{30}	0.00374	0.00184	50.8021
p_{K+1}	0.00171	0.00160	6.4328
ϑ	1.33106	1.33121	0.0113

For a larger buffer size of $K = 30$, similar results are shown in Table II. It can be seen that the errors are small. It is also found that the relative errors on p_0, p_{K+1}, and ϑ are 0.07%, 6.43%, and 0.01%, respectively. However, for p_{30}, the relative error is about 50.80%. Therefore, the proposed method can model the performances with high accuracy, but not for every state of the network.

For packet arrivals described by a two-phase representation with rates $(\lambda_1, \lambda_2, a_1) = (4.0, 2.0, 0.5)$, the service times of the two switches are exponentially distributed with rates 2 and 4. The size of the buffer between the two switches is $K = 10$. The results are shown in Table III, which are similar to those for MGE distributions. The errors are found to be in the range of $0.5 \sim 8.4 \times 10^{-6}$.

For packet arrivals with Erlang distribution, it is assumed that the traffic can be described by an Erlang-10 distribution with rate $\lambda = 10$. The first switch has a service time with an Erlang-10 distribution with mean service time $\mu_1^{-1} = 1$ and $\xi_1 = 0.1$. The second switch has a service time with an Erlang-8 distribution with a mean service time $\mu_2^{-1} = 0.8$ and $\xi_2 = 0.125$. The buffer has size of $K = 25$.

The results are shown in Table IV. The error for throughput is $e = 0.00060593 \sim 0.01509593$, or a relative error $0.06 \sim 1.49\%$. Note that the DM is considered as a relatively accurate method because it models a large number of the states of the underlying Markov chain, for example, 2018 states in this example [20], [23], while we use only 27 states. We also see a large

TABLE III
QUEUES OF PH/M/1 WITH $K = 10$

	DM	This Work
p_0	0.50012	0.50012
p_1	0.25006	0.25006
p_2	0.12503	0.12503
p_3	0.06252	0.06251
p_4	0.03126	0.03126
p_5	0.01563	0.01563
p_6	0.00781	0.00781
p_7	0.00391	0.00391
p_8	0.00195	0.00195
p_9	0.00098	0.00098
p_{10}	0.00048	0.00049
p_{K+1}	0.00024	0.00024
ϑ	1.99952	1.99951

TABLE IV
ERLANG-N DISTRIBUTIONS AND $K = 25$

	DM	This Work
p_0	0.18841	0.20049
p_{K+1}	2.6166e-10	7.5741e-4
K_{pkt}	1.24542	3.93456
ϑ	1∼1.01449	0.99939

TABLE V
THROUGHPUT FOR 3 AND 5 STAGES

Methods	3 Stages	5 Stages
This Work	0.2204	0.1987
DM	0.2254	0.1984
Simulation	0.2262	0.1964

relative error for p_{K+1}, which is caused by the numerical instabilities due to the procedures in finding a solution to the buffer states when the probability is very small.

Example 2: We consider a special case of a tandem queue model for which an exact solution exists. The network consists of three switches and two buffers in between them. The traffic arrived to the first switch is Poisson with rate $\lambda = 2.0$. The switches all have exponentially distributed service times, with rates $\mu_1 = \mu_2 = \mu_3 = 2.0$. Two buffers have size of $K_1 = K_2 = 1$. For the throughput, our result is $\vartheta = 1.13968$, with an error of $e = 0.01147$, or a relative error $e_r = 1.01665\%$, as compared to the exact result obtained by using the Hunt's formula $\vartheta = 1.12821$ [20]. For DM, $\vartheta = 1.11111$, the error is $e = -0.0171$, or a relative error $e_r = 1.51568\%$. It can be seen that our method results in slightly better accuracy than DM.

We consider a more general three-stage tandem queue with Poisson traffic with rate $\lambda = 0.5$. Three switches have mean service times $\mu^{-1} = 2.0$, 4.0, and 3.0, with squared coefficient of variation $\xi = 2.0$, 1.0, and 0.5, respectively. Two buffers have sizes: $K_1 = 3$ and $K_2 = 3$. The throughput values are shown in the second column of Table V.

Example 3: Consider a five-stage tandem queue network shown in Fig. 6. The traffic has a mean interarrival time of 2.5 with $\xi = 2.5$. The switches have mean service times $= 2.5$, 4.0, 3.0, 2.0, and 5.0, with the squared coefficient of variation $\xi = 2.5$, 0.25, 3.0, 0.30, and 0.75, respectively. The buffer sizes are $K = 4$, 4, 10, and 4, respectively. The results on network throughput are shown in the third column of Table V. The simulation results in the table are the average of 15 simulation runs using Matlab on the Dell server, with 100 000 packets processed during each run.

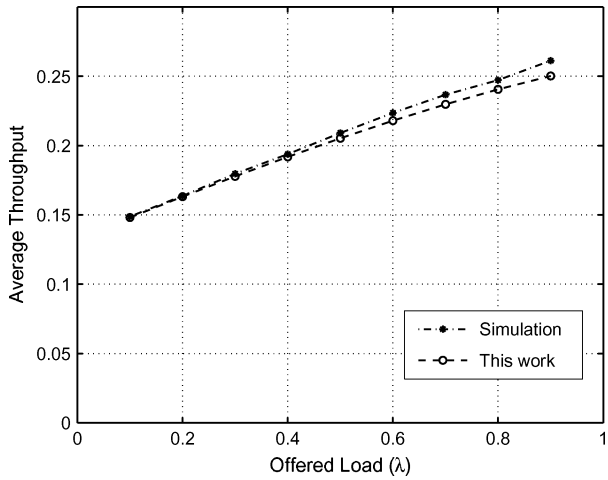

Fig. 8. Throughput for the three-stage Clos network.

Example 4: Consider a more complicated switch network in Fig. 2, where $n = 3$, $r = 2$, $m = 3$, and $M = 3$. We use subscript (k, j, i) to denote the kth input of the jth switch in the ith stage. In this example, we assume that all the buffers have the size of 4, i.e., $K_{(k,j,i)} = 4$, for all k, j, and i. The offered traffic load to the switches in the first stage is $\lambda_{(k,j,1)} = \lambda$, for $k = 1$, 2, 3, and $j = 1$, 2, where λ varies in the range of $0.1 \sim 0.9$, but is not uniformly distributed over all the output ports in a switch. We assume that for an input port s and output port d of a switch, a specific scheduling algorithm generates: $q_{s,d} = w + (1 - w)/n$, if $s = d$; and $q_{s,d} = (1 - w)/n$, otherwise. Note that w is a fraction of input traffic directed to an output [4]. For the switches at stages 1, 2, and 3, we choose $w = 0.5$, 0.6, and 0.5, respectively. For simplicity, we choose $\gamma_{(k,j,i)} = 0.5$, for all k, j, and i. We assume that all the switches have unit mean service rate, that is, $\mu_{(k,j,i)} = 1.0$, for all k, j, and i.

First, the traffic aggregations are conducted at each switch in each stage by recursively merging two inputs into one at a time until an approximate single input is obtained. Then, for the equivalent tandem queue model with only one traffic input, the buffer state can be found by applying the results given in (32), as well as the network performance, such as the average throughput and packet delay. For different values of λ, the results on throughput are plotted in Fig. 8 (labeled as "This work"). The results on the average delay are plotted in Fig. 9. Also plotted in the figures are the simulation results (labeled as "Simulation"), which are the average of 10 simulation runs. Each run continues until 1000 dropped packets are observed.

It can be seen that the error of the throughput e_ϑ, as compared to the simulation results, is relatively small when $\lambda < 0.50$. As λ increases, the error e_ϑ also increases, which is caused by the approximation error in buffer states due to high offered load. It is observed that $e_\vartheta \leq 0.012$, or a relative error less than 4.0%. As for the average delay, it has a constant error $e_\tau \approx 0.20$ (time units), or a relative error 2.0%, in the whole range of the offered load. The reason is that the delay calculation is impacted by all the buffer states instead of a few states like the throughput. Thus, the impact on the error does not change significantly as the

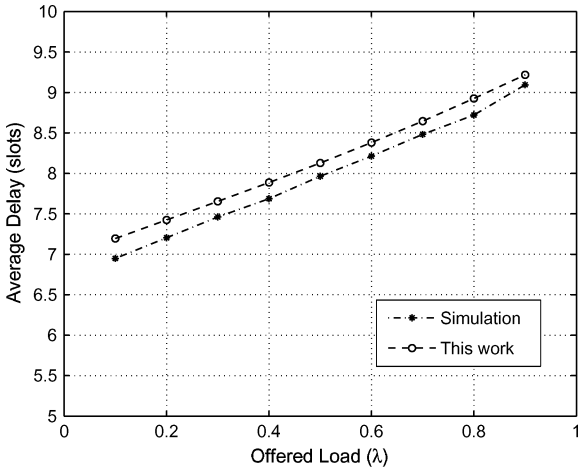

Fig. 9. Delay for the three-stage Clos network.

offered load increases. Similar results are observed in different simulation scenarios, e.g., a relative error less than 5.0%, which is considered as satisfactorily accurate in practice.

VI. Conclusion

This paper presents an approximate method for the performance modeling of complex switch networks with PH and bursty traffic inputs. For both traffic patterns, we propose to aggregate multiple traffic inputs into a single one, convert a network of N inputs and outputs into N tandem queues, and then evaluate the network performance. In this way, the commonly used UTID assumption is no longer needed. We derive a recursive solution for the bursty traffic aggregation in a finite buffer.

It is noticed that the proposed method is based on the assumption that a general traffic can be modeled by the powerful PH distributions. Theoretically, it is true, but in practice, fitting a traffic to a PH model, particularly the interstage traffic, may be difficult. Furthermore, even if an approximate PH representation is obtained, if the number of phases is large, it may be difficult and sometimes impossible to deal with the underlying Markov process due to the size of its state space. Therefore, our future work will investigate the methods to reduce the size of traffic described by PH distributions. Another work is to model switch networks with correlated and mixed traffic and thus further investigate the applicability of the proposed method to more general classes of switch networks.

Appendix

In this Appendix, we present a recursive algorithm to find an exact solution to the aggregation of multiple bursty traffic inputs.

Consider the traffic merging process, shown in Fig. 5, where the bursty traffic model is shown in Fig. 3. Denote by $x = (x_1, x_2, \ldots, x_n)$ the state of buffer B, which consists of the states of input buffers $B_1 \sim B_n$; where x_d, $d = 1, 2, \ldots, n$, equals 1 if buffer B_d transmits in a slot, and 0 otherwise. We assume that the size of B, i.e., $K \geq n$. Let $p^{(t)}(j; s)$ be the probability that buffer B contains j packets at the end of slot t and the buffer state is s. In the steady state, $p^{(t)}(j; s)$ is written as $p(j; s)$. Let $\pi_{x;s}$ be the transition probability that the buffer state goes from x at the end of slot $t-1$ to s at the end of slot t, that is

$$\begin{aligned}\pi_{x;s} &\triangleq Pr\{B\text{'s state goes from } x \text{ at the end of slot } t-1 \\ &\qquad \text{to } s \text{ at the end of slot } t\} \\ &= \prod_{d=1}^{n} Pr\{B_d\text{'s state goes from } x_d \text{ to } s_d\}.\end{aligned}$$

As an example, for $n = 2$, $x, s \in \{00\ 01\ 10\ 11\}$, we can find

$$\pi = A_1 \otimes A_2 \tag{A.1}$$

where $\otimes$ is the Kronecker product; the matrix that describes the traffic input is:

$$A_d = \begin{bmatrix} \bar{a}_d & \bar{b}_d \\ a_d & b_d \end{bmatrix}, \text{ for } d = 1, 2.$$

The first-order Markov model yields the following one-step transition equations [8]:

$$\begin{aligned} p^{(t)}(j; s) &= \sum_{x_1=0}^{1} \cdots \sum_{x_n=0}^{1} p^{(t-1)}\left(j + 1 - \sum_{d=1}^{n} x_d; x\right) \pi_{x;s}, \\ &\qquad \text{for } n+1 \leq j \leq K+1 \qquad \text{(A.2)} \\ &= \sum_{k=1}^{j+1} \sum_{x_1=0}^{1} \cdots \sum_{x_n=0}^{1} I_{(w_x = j+1-k)}(x) p^{(t-1)}(k; x) \pi_{x;s} \\ &\quad + \sum_{x_1=0}^{1} \cdots \sum_{x_n=0}^{1} I_{(w_x = j)}(x) p^{(t-1)}(0; x) \pi_{x;s}, \\ &\qquad \text{for } 0 \leq j \leq n \qquad \text{(A.3)} \end{aligned}$$

where

$$I_X(x) = \begin{cases} 1, & \text{if } x \in X \\ 0, & \text{otherwise} \end{cases}$$

and

$$w_x = \sum_{d=1}^{n} x_d.$$

It can be seen that $1 \leq w_x \leq n$, depends on the traffic arrivals. As we mentioned, if in the current time slot $j = K+1$, then the newly arrived packets will be dropped without admitting to the buffer. More precisely, since the XB will output one packet in a time slot, as seen in the above one-step transition equations, then only one packet will be admitted in the next time slot, and all other newly arrived packets will be dropped. The buffer is said to be overflowed. However, which packet is allowed to enter the buffer is up to the buffer management algorithm. In this work, we assume that the buffer just randomly chooses one of the newly arrived packets for admission.

Accordingly, when $j = K+1$, if $w_x = 2 \sim n$, then there will be $1 \sim (n-1)$ packets that have to be dropped, respectively. Similarly, when $j = K$, if $w_x = 3 \sim n$, then there will be $1 \sim (n-2)$ packets that have to be dropped, respectively. Up to $j = K+3-n$, only if $w_x = n$, only one packet needs to be dropped. For $j \leq K+2-n$, no packets will be dropped.

Therefore, the buffer overflow probability, denoted by p_{ov}, can be found

$$p_{\text{ov}}(K) = \sum_{j=K+3-n}^{K+1} \sum_{w_x=K+3-j}^{n} p(j;x) \tag{A.4}$$

where $p(j;x) = p^{(t)}(j;x)$ as $t \to \infty$, i.e., the steady-state probability of the buffer.

A. Aggregation of Two Traffic Inputs

Recursive Equation: For the case of $n = 2$, $x = (x_1, x_2)$; $x_d \in \{0,1\}$, $d = 1, 2$. In (A.1), the first column of π is $\pi_{00;s}$, $s \in \{00\ 01\ 10\ 11\}$, thus we can denote the first column as π_{00}. Similarly, the second, third, and fourth columns are denoted as π_{01}, π_{10}, and π_{11}, respectively.

In the steady state, (A.2) and (A.3) have the following solutions:

$$\begin{aligned}
p(0;s) =& p(0;00)\pi_{00;s} + p(1;00)\pi_{00;s} \\
p(1;s) =& p(0;10)\pi_{10;s} + p(0;01)\pi_{01;s} + p(2;00)\pi_{00;s} \\
&+ p(1;10)\pi_{10;s} + p(1;01)\pi_{01;s} \\
p(2;s) =& p(0;11)\pi_{11;s} + p(3;00)\pi_{00;s} + p(2;10)\pi_{10;s} \\
&+ p(2;01)\pi_{01;s} + p(1;11)\pi_{11;s} \\
p(j;s) =& p(j+1;00)\pi_{00;s} + p(j;01)\pi_{01;s} \\
&+ p(j;10)\pi_{10;s} + p(j-1;11)\pi_{11;s}, \\
&\text{for } j = 3, 4, \ldots, K \\
p(K+1;s) =& p(K+1;01)\pi_{01;s} + p(K+1;10)\pi_{10;s} \\
&+ p(K;11)\pi_{11;s}.
\end{aligned} \tag{A.5}$$

Denote by

$$\left.\begin{aligned}
s &\triangleq [00\ 01\ 10\ 11]^T \\
P_j &\triangleq [p(j;00)\ p(j;01)\ p(j;10)\ p(j;11)]^T, \\
&\quad j = 0, 1, 2, \ldots, K+1
\end{aligned}\right\}. \tag{A.6}$$

In order to express the individual $p(j;x)$ by its vector form P_j, we need to expand the vector π_x, $x \in \{00\ 01\ 10\ 11\}$ into a matrix. For example, in order to express $p(0;00)$ by P_0, we expand π_{00} into a matrix $[\pi_{00}\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}]$, where $\mathbf{0}$ is a vector of all zeros. Thus, $p(0;00)\pi_{00} = [\pi_{00}\ 0\ 0\ 0]P_0$.

Define

$$\begin{aligned}
\pi_0 &\triangleq [\pi_{00}\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}] \\
\pi_1 &\triangleq [\mathbf{0}\ \pi_{01}\ \pi_{10}\ \mathbf{0}] \\
\pi_2 &\triangleq [\mathbf{0}\ \mathbf{0}\ \mathbf{0}\ \pi_{11}].
\end{aligned}$$

The one-step transition equations in (A.5) become

$$\left.\begin{aligned}
P_0 &= \pi_0 P_0 + \pi_0 P_1 \\
P_1 &= \pi_1 P_0 + \pi_1 P_1 + \pi_0 P_2 \\
P_2 &= \pi_2 P_0 + \pi_2 P_1 + \pi_1 P_2 + \pi_0 P_3 \\
P_j &= \pi_2 P_{j-1} + \pi_1 P_j + \pi_0 P_{j+1} \quad \text{for } j = 3, 4, \ldots, K \\
P_{K+1} &= \pi_2 P_K + \pi_1 P_{K+1}
\end{aligned}\right\}. \tag{A.7}$$

In order to systematically divide the transition matrix into subblocks for general number of inputs, we regroup the state space of s according to the number of packets arrived: $s_0 = [00]$, $s_1 = [01\ 10]^T$, and $s_2 = [11]$. The state regrouping can be shown by dividing an identity matrix of the same size as π into subblocks

$$\left[\begin{array}{c|cc|c} 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right].$$

To further simplify the notation, we define the following matrices:

$$\begin{aligned}
C_0 &= [\mathbf{1}\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}] \\
C_1 &= [\mathbf{0}\ \mathbf{1}\ \mathbf{1}\ \mathbf{0}] \\
C_2 &= [\mathbf{0}\ \mathbf{0}\ \mathbf{0}\ \mathbf{1}]
\end{aligned}$$

where a $\mathbf{1}$ in ith column represents a column vector of all 0 components, except the ith component is 1, while $\mathbf{0}$ represents a column vector of all 0 components.

Now, we can represent the matrix π_0, π_1, and π_2 in terms of π

$$\pi_0 = \pi C_0,\ \pi_1 = \pi C_1,\ \pi_2 = \pi C_2.$$

Note that π_0 is singular. Thus, in (A.7), we cannot express P_{j+1} in terms of P_j, for $j = 0, 1, \ldots, K+1$. In order to do so, we need to rearrange the equations.

Define

$$Q_{00} \triangleq [p(0;00)\ 0\ 0\ 0]^T \tag{A.8}$$

$$\begin{aligned}
Q_{K+1} &\triangleq [0\ p(K+1;01)\ p(K+1;10)\ p(K+1;11)]^T \\
Q_j &\triangleq [p(j+1;00)\ p(j;01)\ p(j;10)\ p(j;11)]^T, \\
&\quad j = 0, 1, 2, 3, \ldots, K.
\end{aligned} \tag{A.9}$$

It can be seen that only the first component of P_j is changed. If we denote Q_{-1} by Q_{00}, then (A.9) can be written as

$$P_j = C_0 Q_{j-1} + (I - C_0)Q_j, \quad \text{for } j = 0, 1, 2, 3, \ldots, K+1 \tag{A.10}$$

where I is an identity matrix of $n \times n$.

Substituting (A.10) into (A.7), and note that $\pi_0 C_0 = \pi_0$, $\pi_1 C_0 = 0$, and $\pi_2 C_0 = 0$, we have

$$\left.\begin{aligned}
Q_0 &= \pi_0 Q_0 + \pi_0 Q_{00} + C_0(Q_0 - Q_{00}) \\
Q_1 &= \pi_1 Q_0 + \pi_1 Q_1 + \pi_0 Q_1 + C_0(Q_1 - Q_0) \\
Q_2 &= \pi_2 Q_0 + \pi_2 Q_1 + \pi_1 Q_2 + \pi_0 Q_2 + C_0(Q_2 - Q_1) \\
Q_j &= \pi_2 Q_{j-1} + \pi_1 Q_j + \pi_0 Q_j + C_0(Q_j - Q_{j-1}) \\
&\quad \text{for } j = 3, 4, \ldots, K \\
Q_{K+1} &= \pi_2 Q_K + \pi_1 Q_{K+1} - C_0 Q_K
\end{aligned}\right\}. \tag{A.11}$$

In the derivation of the last equation, we use the fact that $C_0 Q_{K+1} = 0$. Comparing (A.11) to (A.7), we can see the corresponding changes in the equations are straightforward. First, for each equation of P_j, $j = 0, 1, 2, 3, \ldots K+1$, change all the $P_0, P_1, \ldots, P_j$ to $Q_0, Q_1, \ldots, Q_j$, respectively, except that the last term, P_{j+1}, which is multiplied by π_0, needs to be changed to Q_j. Second, in the first equation, P_1 needs to be changed to Q_{00}. Third, each equation needs to be compensated by $C_0(Q_j - Q_{j-1})$ in the right-hand side of the equation as the results of the rearrangement.

Now, (A.11) can be easily solved as follows:

$$\left.\begin{array}{l} Q_0 = \Phi_0(\pi_0 - C_0)Q_{00} \\ Q_1 = \Phi_1(\pi_1 - C_0)Q_0 \\ Q_2 = \Phi_1\{\pi_2 Q_0 + (\pi_2 - C_0)Q_1\} \\ Q_j = \Phi_1(\pi_2 - C_0)Q_{j-1}, \text{ for } j = 3, \ldots, K, \\ Q_{K+1} = \Phi_2(\pi_2 - C_0)Q_K \end{array}\right\}. \quad \text{(A.12)}$$

where

$$\left.\begin{array}{l} \Phi_0 = (I - \pi_0 - C_0)^{-1} \\ \Phi_1 = (I - \pi_1 - \pi_0 - C_0)^{-1} \\ \Phi_2 = (I - \pi_1)^{-1} \end{array}\right\}. \quad \text{(A.13)}$$

It can be verified that Φ_0 does exist, so do Φ_1 and Φ_2. Therefore, $Q_j, j = 0, 1, 2, 3, \ldots K+1$, can be expressed in terms of Q_{00}, which contains only one unknown, i.e., $p(0;00)$.

Normalization Condition: Define a row vector of ones: $e \triangleq [1\ 1\ \ldots\ 1]$, we have $p_j = eP_j$. The normalization condition is

$$\sum_{j=0}^{K+1} eP_j = 1. \quad \text{(A.14)}$$

Substituting (A.10) into (A.14), we have

$$e(Q_{00} + \sum_{j=0}^{K+1} Q_j) = 1. \quad \text{(A.15)}$$

To simplify the notation, we define

$$\left.\begin{array}{l} R_0 = \Phi_0(\pi_0 - C_0) \\ R_1 = \Phi_1(\pi_1 - C_0) \\ R_{20} = \Phi_1\pi_2 \\ R_{21} = \Phi_1(\pi_2 - C_0) \\ R_2 = \Phi_2(\pi_2 - C_0) \end{array}\right\} \quad \text{(A.16)}$$

and also

$$S \stackrel{\text{def}}{=} Q_{00} + \sum_{j=0}^{K+1} Q_j. \quad \text{(A.17)}$$

By substituting (A.12) and (A.16) into (A.17), we have

$$S = \Psi_1 Q_{00} + R_{21}S + \Psi_2 Q_{00}$$

that is

$$S = (I - R_{21})^{-1}(\Psi_1 + \Psi_2)Q_{00} \quad \text{(A.18)}$$

where

$$\left.\begin{array}{l} \Psi_1 = (I + R_0 - R_{21}) + (R_1 + R_{20} - R_{21})R_{21} \\ \Psi_2 = (R_2 - R_{21} - R_{21}R_2)R_{21}^{K-2}(R_{20} + R_{21}R_1)R_0 \end{array}\right\}. \quad \text{(A.19)}$$

By substituting (A.18) into (A.15), we have

$$e(I - R_{21})^{-1}(\Psi_1 + \Psi_2)Q_{00} = 1.$$

Thus, we easily find

$$p(0;00) = \frac{1}{e\,[(I - R_{21})^{-1}(\Psi_1 + \Psi_2)]_{\text{col},1}} \quad \text{(A.20)}$$

where $[\cdot]_{col,1}$ means the first column of the inside matrix.

Boundary Condition: The boundary condition can be derived as follows:

$$\begin{aligned} p(0;x) &\triangleq Pr[B_d \text{ is empty}] \\ &\quad \cdot Pr[\text{state of } B_1, \ldots, B_n \text{ is } x | B_d \text{ is empty}] \\ &= p_0 Pr[\text{state of } B_1, \ldots, B_n \text{ goes from } 0 \to x] \\ &= p_0 \prod_{d=1}^{n} \delta_d \end{aligned}$$

where

$$\delta_d = \begin{cases} a_d, & \text{if } x_d = 1 \\ \bar{a}_d, & \text{if } x_d = 0 \end{cases}$$

and $\bar{a}_d = 1 - a_d$, where a_d can be obtained from the parameter of the individual traffic stream $B(\lambda_d, \gamma_d)$ by using (14).

Noting the definition in (A.6), we have

$$P_0 = p_0[\bar{a}_1, a_1]^T \otimes \ldots \otimes [\bar{a}_n, a_n]^T, \quad \text{(A.21)}$$

in which the first row is

$$p(0;00) = p_0 \bar{a}_1 \ldots \bar{a}_n.$$

Thus, we have

$$p_0 = p(0;00)/(\bar{a}_1 \cdots \bar{a}_n) \quad \text{(A.22)}$$

where $p(0;00)$ is given in (A.20).

In terms of (A.20) and (A.8), we immediately find Q_{00}. By substituting Q_{00} into (A.12), we find $Q_j, j = 0, 1, 2, \ldots, K+1$. In terms of (A.10), we find $P_j, j = 1, 2, \ldots, K+1$. Finally, we have

$$p_j = eP_j, \quad j = 1, 2, \ldots, K+1. \quad \text{(A.23)}$$

The parameters for the aggregated traffic and the buffer overflow probability can be obtained, as defined in (17) and (A.4), respectively.

B. Aggregation of Three Traffic Inputs

For $n = 3$, the transition matrix can be found

$$\pi_{\{n=3\}} = \pi_{\{n=2\}} \otimes A_3$$

which is an 8×8 matrix. Note that $2^3 = 8$.

The state vector is

$$x = [000\ 001\ 010\ 100\ 011\ 101\ 110\ 111]^T$$

which can be divided into $n + 1 = 4$ groups, corresponding to the number of packets arrived at a time

$$i = I_{w(x)} = 0, 1, 2, 3.$$

The number of states in each group is a binomial coefficient:

$$n_i = \binom{n}{i} = 1, 3, 3, 1, \quad \text{for } i = 0, 1, 2, 3.$$

The division can be shown by an 8×8 identity matrix: The four blocks are the first column, next three columns, three columns again, and the last column

$$\left[\begin{array}{c|ccc|ccc|c} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right].$$

Similar to the case of $n = 2$, we define $C_0 = [\mathbf{1}\,\mathbf{0}\,\ldots\,\ldots\,\ldots\,\mathbf{0}]$, $C_1 = [\mathbf{0}\,\mathbf{1}\,\mathbf{1}\,\mathbf{1}\,\mathbf{0}\,\ldots\,\mathbf{0}]$, $C_2 = [\mathbf{0}\,\ldots\,\mathbf{0}\,\mathbf{1}\,\mathbf{1}\,\mathbf{1}\,\mathbf{0}]$, and $C_3 = [\mathbf{0}\,\ldots\,\ldots\,\ldots\,\mathbf{0}\,\mathbf{1}]$, where a $\mathbf{1}$ in ith position represents a column vector of all 0 components, except the component at the ith position in the vector is 1, while a $\mathbf{0}$ represents a column vector of all 0's. Now, we can divide π into

$$\pi_i = \pi C_i, \quad \text{for } i = 0, 1, 2, 3.$$

The one-step transition equations for P_j, $j = 0, 1, \ldots, n$, are the same as those for $n = 2$. For P_j, $j = n+1, \ldots, K$

$$\left.\begin{array}{l} P_3 = \pi_3 P_0 + \pi_3 P_1 + \pi_2 P_2 + \pi_1 P_3 + \pi_0 P_4 \\ P_j = \pi_3 P_{j-2} + \pi_2 P_{j-1} + \pi_1 P_j + \pi_0 P_{j+1} \\ \qquad \text{for } j = 4, 5, \ldots, K \\ P_{K+1} = \pi_3 P_{K-1} + \pi_2 P_K + \pi_1 P_{K+1} \end{array}\right\}. \tag{A.24}$$

The above equations can be similarly solved as follows:

$$\left.\begin{array}{l} Q_3 = \Phi_1 \{\pi_3 Q_0 + \pi_3 Q_1 + (\pi_2 - C_0) Q_2\} \\ Q_j = \Phi_1 \{\pi_3 Q_{j-2} + (\pi_2 - C_0) Q_{j-1}\} \quad \text{for } j = 4, 5, \ldots, K \\ Q_{K+1} = \Phi_2 \{\pi_3 Q_{K-1} + (\pi_2 - C_0) Q_K\} \end{array}\right\} \tag{A.25}$$

where Φ_0, Φ_1, and Φ_2 are defined in (A.13). The normalization and boundary conditions are the same as (A.15) and (A.22), respectively. Therefore, we can similarly find P_j, $j = 0, 1, 2, \ldots, K+1$.

C. Aggregation of n Traffic Inputs

For a general case of n traffic inputs, the transition matrix can be found

$$\pi_{\{n\}} = \pi_{\{n-1\}} \otimes A_n.$$

The state x can be divided into $n+1$ groups, corresponding to the number of packets arrived at a time: $i = 0, 1, 2, \ldots, n$. The number of states in each group is the binomial coefficient

$$n_i = \binom{n}{i}, \quad \text{for } i = 0, 1, 2, \ldots, n.$$

The division can be shown by a $2^n \times 2^n$ identity matrix: every n_i number of columns form a block, $i = 0, 1, \ldots, n$. Correspondingly, C_i, $i = 0, 1, \ldots, n$, can be defined using vectors $\mathbf{1}$ and $\mathbf{0}$. In this way, we can divide $\pi_{\{n\}}$ into

$$\pi_i = \pi_{\{n\}} C_i, \quad \text{for } i = 0, 1, 2, \ldots, n.$$

The one-step transition equations for $P_j, j = 0, 1, \ldots, K+1$, are

$$P_j = \begin{cases} \pi_j P_0 + \sum_{k=1}^{j+1} \pi_{j+1-k} P_k, & \text{for } j = 0, 1, 2, \ldots, n \\ \sum_{k=j+1-n}^{j+1} \pi_{j+1-k} P_k, & \text{for } j = n+1, \ldots, K \\ \sum_{k=1}^{n} \pi_k P_{K-k+2}, & \text{for } j = K+1. \end{cases}$$

By using the transformation in (A.9) and (A.10), the above equations are reformatted as follows:

$$\begin{aligned} Q_0 &= \Phi_0(\pi_0 - C_0) Q_{00} \\ Q_1 &= \Phi_1(\pi_1 - C_0) Q_0 \\ Q_j &= \Phi_1 \left\{ \pi_j Q_0 + \sum_{k=1}^{j-2} \pi_{j+1-k} Q_k + (\pi_2 - C_0) Q_{j-1} \right\}, \\ &\qquad \text{for } j = 2, \ldots, n \\ &= \Phi_1 \left\{ \sum_{k=j+1-n}^{j-2} \pi_{j+1-k} Q_k + (\pi_2 - C_0) Q_{j-1} \right\}, \\ &\qquad \text{for } j = n+1, \ldots, K \\ &= \Phi_2 \left\{ \sum_{k=3}^{n} \pi_k Q_{K-k+2} + (\pi_2 - C_0) Q_K \right\}, \\ &\qquad \text{for } j = K+1 \end{aligned}$$

where Φ_0, Φ_1, and Φ_2 are defined in (A.13). Note that the above set of linear equations has only one unknown. By using the normalization and boundary conditions e.g., (A.15) and (A.22), P_j, $j = 0, 1, 2, \ldots, K+1$, can be similarly obtained.

In the above recursive algorithm, the traffic aggregation is conducted recursively in terms of the number of traffic inputs. To aggregating n inputs into one, the algorithm needs to run $n-1$ times. Within each time of the traffic aggregation, the buffer states are solved recursively in terms of the size of the buffer. For a buffer size K, the algorithm needs to run $K+1$ times recursively. Therefore, the recursion stops after $(n-1) \times (K+1)$ times of iteration.

ACKNOWLEDGMENT

The authors would like to thank the editors and reviewers for their constructional comments and suggestions that help improve this paper.

REFERENCES

[1] E. Oki, N. Yamanaka, K. Nakai, and N. Matsuura, "Multi-stage switching system using optical WDM grouped links based on dynamic bandwidth sharing," *IEEE Commun. Mag.*, vol. 41, no. 10, pp. 56–63, Oct. 2003.

[2] J. S. Turner, "Queueing analysis of buffered switching networks," *IEEE Trans. Commun.*, vol. 41, no. 2, pp. 412–420, Feb. 1993.

[3] J. S. Turner and R. Melen, "Multirate Clos networks," *IEEE Commun. Mag.*, vol. 41, no. 10, pp. 38–44, Oct. 2003.

[4] R. Rojas-Cessa, E. Oki, and H. J. Chao, "On the combined input-cross-point buffered packet switch with round-robin arbitration," *IEEE Trans. Comms.*, vol. 53, no. 11, pp. 1945–1951, Nov. 2005.

[5] G. Nong, J. K. Muppala, and M. Hamdi, "Analysis of nonblocking ATM switches with multiple input queues," *IEEE/ACM Trans. Netw.*, vol. 7, no. 1, pp. 60–74, Feb. 1999.

[6] G. Nong, M. Hamdi, and J. K. Muppala, "Performance evaluation of multiple input-queued ATM switches with PIM scheduling under bursty traffic," *IEEE Trans. Commun.*, vol. 49, no. 8, pp. 1329–1333, Aug. 2001.
[7] A. Pattavina and C. Catania, "Performance analysis of ATM replicated banyan networks with external input-output queueing," *Telecommun. Syst.*, vol. 23, no. 1/2, pp. 149–170, 2003.
[8] A. M. Viterbi, "Approximate analysis of time-synchronous packet networks," *IEEE J. Sel. Areas Commun.*, vol. SAC-4, no. 6, pp. 879–890, Sep. 1986.
[9] I. Stavrakakis, "Efficient modeling of merging and splitting processes in large networking structures," *IEEE J. Sel. Areas Commun.*, vol. 9, no. 8, pp. 1336–1347, Oct. 1991.
[10] D. P. Heyman and D. Lucantoni, "Modeling multiple IP traffic streams with rate limits," *IEEE/ACM Trans. Netw.*, vol. 11, no. 6, pp. 948–958, Dec. 2003.
[11] P. Salvador, R. Valadas, and A. Pacheco, "Multiscale fitting procedure using Markov modulated Poisson processes," *Telecommun. Syst.*, vol. 23, no. 1–2, pp. 123–148, Jun. 2003.
[12] T. Osogami, "Analysis of multi-server system via dimensionality reduction of Markov chains," Ph.D. dissertation, School of Computer Science, Carnegie Mellon Univ., Pittsburgh, PA, Jun. 2005.
[13] Z. Zhang and Y. Yang, "A novel analytical model for switches with shared buffer," *IEEE/ACM Trans. Netw.*, vol. 15, no. 5, pp. 1191–1203, Oct. 2007.
[14] G. Bianchi and J. S. Turner, "Improved queueing analysis of shared buffered switching networks," *IEEE/ACM Trans. Netw.*, vol. 1, no. 4, pp. 482–490, Aug. 1993.
[15] S. Gianatti and A. Pattavina, "Performance analysis of ATM banyan networks with shared queueing, I. Radom offred traffic," *IEEE/ACM Trans. Netw.*, vol. 2, no. 4, pp. 398–410, Aug. 1994.
[16] S. Gianatti and A. Pattavina, "Performance analysis of ATM banyan networks with shared queueing, II. Correlated/unbalanced offered traffic," *IEEE/ACM Trans. Netw.*, vol. 2, no. 4, pp. 411–424, Aug. 1994.
[17] C. Bouras, J. Garofalakis, P. Spirakis, and V. Triantafillou, "An analytical performance model for multistage interconnection networks with finite, infinite and zero length buffers," *Perform. Eval.*, vol. 34, pp. 169–182, 1998.
[18] X. R. Cao and D. Towsley, "A performance model for ATM switches with general packet length distribution," *IEEE/ACM Trans. Netw.*, vol. 3, no. 3, pp. 299–309, Jun. 1995.
[19] M. Moulki, A. L. Beylot, L. Truffet, and M. Becker, "An aggregation technique to evaluate the performance of a two-stage buffered ATM switch," *Ann. Oper. Res.*, vol. 79, pp. 373–392, 1998.
[20] T. Altiok, *Performance Analysis of Manufacturing Systems*. New York: Springer, 1997.
[21] T. Altiok, "Approximate analysis of queues in series with phase-type service times and blocking," *Oper. Res.*, vol. 37, pp. 601–610, 1989.
[22] L. Gun and A. Makowski, "An approximate method for general tandem queueing systems subject to blocking," in *Queueing Networks with Blocking*, H. G. Ferraros and T. Altiok, Eds. Amsterdam, The Netherlands: Elsevier, 1989.
[23] Y. Dallery and H. L. Bihan, "An improved decomposition method for the analysis of production lines with unreliable machines and finite buffers," *Int. J. Prod. Res.*, vol. 37, no. 5, pp. 1093–1117, 1999.
[24] S. Helber and H. Jusic, "A new decomposition approach for non-cyclic continuous material flow lines with a merging flow of materials," *Ann. Oper. Res.*, vol. 125, no. 1/4, pp. 117–139, Jan. 2004.
[25] G. Latouche and V. Ramaswami, *Introduction to Matrix Analysis Methods in Stochastic Modeling*. Philadelphia, PA: ASA-SIAM, 1999.
[26] M. A. Johnson and M. F. Taaffe, "An investigation of phase-distribution moment-matching algorithms for use in queueing models," *Queueing Syst., Theory Appl.*, vol. 8, no. 2, pp. 129–147, 1991.
[27] A. Panchenko and A. Thummler, "Efficient phase-type fitting with aggregated traffic traces," *Perform. Eval.*, vol. 64, no. 7/8, pp. 629–645, Aug. 2007.

Ming Yu (M'97–SM'03) received the Doctor of Engineering degree from Tsinghua University, Beijing, China, in 1994, and the Ph.D. degree from Rutgers University, New Brunswick, NJ, in 2002, both in electrical and computer engineering.

He joined the Operation Technology Center, AT&T, Middletown, NJ, in July 1997 as a Senior Technical Staff Member. In 1999, he was with the Department of ATM Network Services, AT&T Labs. From December 2002, he worked for the Department of IP/Data Network Management System Engineering, AT&T Labs. During September 2003 and August 2006, he was with the Department of Electrical and Computer Engineering, State University of New York at Binghamton, NY. As of September 2006, he joined the Department of Electrical and Computer Engineering, Florida State University, Tallahassee, and is currently an Associate Professor. His research interests are in the areas of routing protocols, MAC, QoS, security, energy efficiency, clustering, radio resource management, traffic engineering, and performance analysis for both wired and wireless networks.

Dr. Yu was awarded an IEEE Third Millennium Medal in May 2000.

Mengchu Zhou (S'88–M'90–SM'93–F'03) received the B.S. degree from Nanjing University of Science and Technology, Nanjing, China, in 1983; the M.S. degree from Beijing Institute of Technology, Beijing, China, in 1986; and the Ph.D. degree from Rensselaer Polytechnic Institute, Troy, NY, in 1990, all in electrical engineering.

He joined the New Jersey Institute of Technology (NJIT), Newark, in 1990 and is currently a Professor of electrical and computer engineering and Director of the Discrete-Event Systems Laboratory. He has over 300 publications, including six books and more than 120 journal papers. He has led or participated in over 30 research and education projects with total budget over $10 million, funded by the NSF, DOD, and industry. His interests are in computer integrated systems, Petri nets, networks, and automation.

Prof. Zhou is a Life Member of the Chinese Association for Science and Technology—USA and served as its President in 1999. He was the recipient of the CIM University LEAD Award from the Society of Manufacturing Engineers, the Perlis Research Award from NJIT, the Humboldt Research Award for US Senior Scientists, and a Distinguished Lecturer of the IEEE SMC Society. He is Managing Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, Associate Editor of the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and Editor-in-Chief of the *International Journal of Intelligent Control and Systems*. He served as General and Program Chair for many international conferences.