



Traffic splitting in a network: split traffic models and applications

Huei-Wen Ferng*, Cheng-Ching Peng

*Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology,
43 Keelung Road, Section 4, Taipei 106, Taiwan, ROC*

Abstract

The contemporary high-speed networks, e.g. the Internet and asynchronous transfer mode (ATM) networks provide a convenient and cost-effective communication platform to carry the emerging multimedia applications. However, problems, such as network congestion caused by overloaded traffic get worse day by day. Thus, how to properly control the system while maintaining the quality of service (QoS) for users becomes an important and challenging issue. To reach the goal of QoS guarantee, load sharing or traffic balancing is one of important techniques to improve the whole network performance. In this paper, we focus on the probabilistic routing policy and analyze split traffic models under such a probabilistic routing policy with input traffic models of the batch Markovian arrival process (BMAP) and discrete-time BMAP (D-BMAP) which can be employed to characterize the multimedia traffic in current and future networks. We successfully show that the split traffic on each routing path from an original BMAP (D-BMAP) input is still a BMAP (D-BMAP). The detailed traffic structure is also acquired. Applying traffic splitting along with the output process characterization, we propose a network-wise performance evaluation method suitable for mesh networks with a probabilistic routing algorithm to get the end-to-end delay, delay variance, and cell loss probability. With numerical experiments, we show that (i) the probabilistic routing policy with a random routing matrix performs well for systems; (ii) the proposed method of network-wise performance analysis has good accuracy.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Traffic splitting; Probabilistic routing; Batch Markovian arrival process; Discrete-time batch Markovian arrival process; Mesh network; Network-wise performance

1. Introduction

The rapid growth and development of networks have raised problems, such as congestion etc. in current high-speed networks, e.g. the Internet and ATM network [8,18,22,23]. Therefore, corresponding issues, e.g. congestion control and admission control to solve the above problem have drawn much attention in the literature. Before considering these issues, system designers first need a powerful tool to help themselves understand the performance of a whole network. However, it is hard to develop such a tool due to (i) traffic modeling to properly capture the bursty nature exhibited by multimedia sources, such as voice, data, and video etc.; (ii) large scale of networks; and (iii) complicated network operations, e.g. routing algorithms. To address the issues mentioned above, more sophisticated models, such as BMAP [19,26]

and D-BMAP [4,30] are proposed and suggested in the literature. And systematic methods for performance evaluation, e.g. those in Refs. [8,10,12,15,16] are employed to analyze network-wise performance. As for network operations, they are mostly treated specifically. Likewise, we focus on the routing mechanism only in this paper. Even only routing mechanisms are considered, this makes one hard to represent or model the traffic sources inbound to intermediate nodes in a network. To solve the problem, we develop split traffic models and discuss an application in this paper. In fact, the traffic splitting related issues also received attention in the research field because of considerations of load balance, network efficiency, and reliability. For examples, Bertsekas [2] studied routing algorithms for computer networks and showed that a routing policy that uses only a single path is not necessarily optimal from the viewpoint of an entire network even though the path is the shortest one. Bertsekas thus pointed out that the traffic splitting mechanism plays an important role on traffic load balancing. In Ref. [25], Livny and Melman illustrated that

* Corresponding author. Tel.: +886-2-2730-1064; fax: +886-2-2730-1081.

E-mail address: hwferng@mail.ntust.edu.tw (H.-W. Ferng).

a network is always inefficient unless load balancing policies are applied since some nodes in the network are busy for packets waiting for service while others are idle.

In the literature, many traffic distributing/routing schemes, such as round robin policy (RRP) [14,27] probabilistic routing policy (PRP), and weighted fair routing policy (WFRP) [24] have been proposed. Among these schemes, the RRP assigning packets to a set of available routing paths in a round robin manner is the most common one [21,29]. In Ref. [1], Adishesu et al. further proposed a scheme modified from the traditional RRP capable of treating packets of variable-length and non-first-in-first-out data delivery. As for the probabilistic routing policy, it routes packets or cells to all paths according to a specific distribution. To reach the goal of probabilistic routing, Cao et al. [6] refined the hashing-based traffic splitting algorithms. Five direct hashing methods and one table-based hashing method were evaluated in Ref. [6] in which the hashing using a 16-bit CRC over the five-tuple performs most excellent in load balance. Saltouros et al. also applied the probabilistic routing policy to the hierarchical routing in an ATM network in Ref. [28], where a routing cost function achieving the minimum number of hops is used to assign a path selection probability. To consider the load sharing of packet-switched networks, a weighted fair routing policy was proposed by Leung and Li in Ref. [24].

Compared with hashing-based traffic splitting approaches [6] which can preserve per-flow packet ordering but need extra efforts to maintain hashing keys and an appropriate hashing function, the probabilistic routing policy is easy to implement and can be directly applied to existing protocols with few modifications. As for the RRP, it has the property of simplicity but it is only suitable for uniform traffic splitting over all routing paths (see Ref. [5]). For the heterogeneous networks, the RRP may not be a proper choice. However, the probabilistic routing policy is appropriate to both homogeneous and heterogeneous networks. For the above reasoning, we adopt the probabilistic routing algorithm in this paper since it is easy to implement. Based on such a routing algorithm with further consideration of the broadest traffic sources including BMAP and D-BMAP capable of capturing the characteristics of the bursty multimedia traffic in current or future high-speed networks, we try to study the model of split traffic passing through a node/element, e.g. router or switch in a network. Thus, the above work of this paper can be regarded as an extension of our previous work [12] in which only the Markov-modulated Poisson process (MMPP) fed into the traffic splitting element, i.e. router or switch is taken into account. Hence, we provide more general results in this paper to facilitate border applications.

With the knowledge of split traffic models of the probabilistic routing policy, we are able to have the input traffic models at intermediate nodes in a network. Thus, the split traffic models can be applied at least to the network-wise performance evaluation for mesh networks with

probabilistic routing. For illustrative purpose, we just feed MMPPs which are special cases of the BMAP to the network because of simplicity and tractability. As for this direction of applications, there are a lot of related papers in the literature. For example, Winston [31] investigated the performance of a homogeneous system using the Poisson input process and exponential distributed service times. In Refs. [8,10], Ferng and Chang studied the end-to-end performance for ATM networks and general queueing networks using discrete-time and continuous-time queueing systems of infinite capacity with heterogeneous MMPPs. To further enlarge the application scope, Chao [7] used MMPPs (and MAPs) and finite-buffer systems to analyze the path-level performance metrics of the resultant system, including end-to-end delay, delay variance, and cell loss probability. However, the above results are applicable to environments of connection-oriented networks only. In this paper, we propose a method of network-wise performance analysis with the aid of traffic splitting for networks with a probabilistic routing policy. The method is then definitely suitable for both connection-oriented in which a routing probability of 1 is set for a specific path and connectionless networks which is approximately emulated by the probabilistic routing algorithm.

The rest of the paper is organized as follows. In Section 2, the system and traffic models are described. Then, we derive split traffic models pertinent to traffic splitting under a probabilistic routing algorithm in Section 3 for both BMAP and D-BMAP input traffic models. In Section 4, we apply the results obtained in the previous section to an application of network-wise performance evaluation using an approximate systematic method to evaluate the end-to-end performance for networks with probabilistic routing algorithms. As for Section 5, we illustrate that the approximate method in Section 4 performs well through numerical experiments. Finally, Section 6 concludes the paper and outlines possible future works.

2. Model descriptions for the system and input traffic

2.1. System model

Since we investigate traffic splitting and an application of network-wise performance evaluation, the system to be addressed is a mesh network like that in Fig. 1(a) in which each node (either a router or a switch) is endowed with a probabilistic routing policy (Fig. 1(b) shows one possible path.). Let us now further elaborate the model for each node. Each node is modeled by a queue of either finite- or infinite-buffer with service times of general distributions. All packets in the buffer get their own service based on the first-come-first-serve (FCFS) discipline. For the finite-buffer queue, all arriving packets get dropped once the buffer is full, i.e. the queued packets plus the packet in service reach the capacity denoted by the symbol K if required.

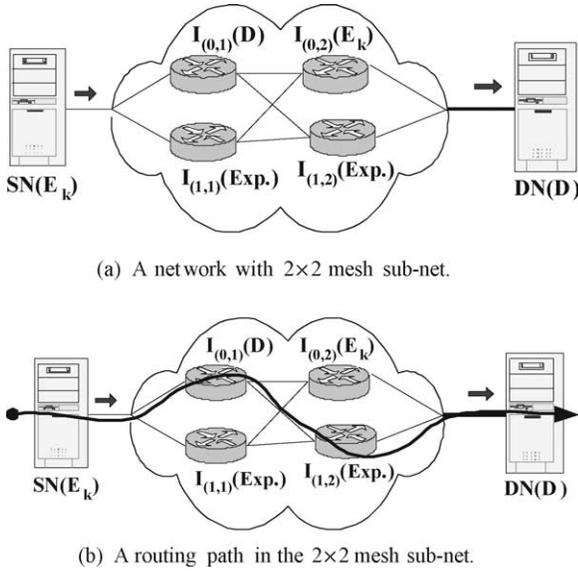


Fig. 1. Topology and a routing path of a mesh network.

In the following, both the BMAP and D-BMAP traffic models are allowed to input to each node. To be familiar with these two input processes, let us now briefly describe their definitions and notations mostly taken from Refs. [3,4,26] for readers' convenience as follows.

2.2. Batch Markovian arrival process (BMAP)

The BMAP was proposed by Lucantoni [26] and is the most sophisticated Markovian point process of continuous time reported in the literature. For an m -state BMAP, it is governed by two types of $m \times m$ rate matrices: non-singular matrix $\mathbf{D}_0^{[c]}$ governing transitions with no arrival in which non-negative off-diagonal elements and negative diagonal elements cause zero sums for all rows and non-negative matrices $\mathbf{D}_k^{[c]}$, $k \geq 1$, governing transitions corresponding to arrivals of batch size k . Thus, it constructs a two-dimensional Markov process $\{N^{[c]}(t), J^{[c]}(t)\}$ with the corresponding state space $\{(v, i) : v \geq 0, 1 \leq i \leq m\}$, where $N^{[c]}(t)$ denotes the number of arrivals in $(0, t]$ and $J^{[c]}(t)$ is the state of the process at time t . Therefore, a transition from state (v, i) to state $(v+k, j)$, $k \geq 0, 1 \leq i, j \leq m$ triggers an arrival of batch size k with state change from i to j and it is governed by the (i, j) th entry of the matrix $\mathbf{D}_k^{[c]}$. Let matrix $\mathbf{D}^{[c]} = \sum_{k=0}^{\infty} \mathbf{D}_k^{[c]}$, it then represents the transition matrix of the underlying Markov chain. First defining the matrix $\mathbf{D}^{[c]}(z)$ to be the matrix-form generating function of $\mathbf{D}_k^{[c]}$, i.e.

$$\mathbf{D}^{[c]}(z) = \sum_{k=0}^{\infty} \mathbf{D}_k^{[c]} z^k \quad (1)$$

as well as the *conditional counting density function*

$$[\mathbf{P}^{[c]}(n, t)]_{ij} = \Pr\{N^{[c]}(t) = n, J^{[c]}(t) = j | N^{[c]}(0) = 0, J^{[c]}(0) = i\} \quad (2)$$

which is the (i, j) th entry of the $m \times m$ matrix $\mathbf{P}^{[c]}(n, t)$. Lucantoni [26] showed that the matrix-form generating function $\mathbf{P}^{[c]*}(z, t)$ of $\mathbf{P}^{[c]}(n, t)$, i.e. $\mathbf{P}^{[c]*}(z, t) = \sum_{n=0}^{\infty} \mathbf{P}^{[c]}(n, t) z^n$ has the following form:

$$\mathbf{P}^{[c]*}(z, t) = e^{\mathbf{D}^{[c]}(z)t} \quad (3)$$

2.3. Discrete-time batch Markovian arrival process (D-BMAP)

Let us now consider the D-BMAP which is a discrete-time counterpart of the BMAP. Without loss of generality, the time axis is sliced into fixed intervals called slots which are enumerated successively and all transitions or arrivals for the D-BMAP occur at slot boundaries, e.g. the end of a slot. First defining $m \times m$ transition matrices $\mathbf{D}_k^{[d]}$, $k \geq 0$ from an arbitrary slot, say l th slot, to the next slot $((l+1)$ th slot) which are related to an arrival of batch size k ($k=0$ indicates no arrival), the m -state D-BMAP can be described using these matrices. Let matrix $\mathbf{D}^{[d]} = \sum_{k=0}^{\infty} \mathbf{D}_k^{[d]}$, it again represents the transition matrix of the underlying Markov chain for the D-BMAP. Let $N^{[d]}(0, l)$ denote the number of arrivals between 0th and l th slot and $J^{[d]}(l)$ be the state at slot l . Then $\{N^{[d]}(0, l), J^{[d]}(l)\}$ forms a two-dimensional discrete-time Markov chain on the state space $\{(v, s) : v \geq 0, 1 \leq s \leq m\}$. As for the *conditional counting density function* for the D-BMAP, it can be defined as follows:

$$[\mathbf{P}^{[d]}(w, l)]_{ij} = \Pr\{N^{[d]}(0, l) = w, J^{[d]}(l) = j | N^{[d]}(0, 0) = 0, J^{[d]}(0) = i\} \quad (4)$$

where $\mathbf{P}^{[d]}(w, l)$ is an $m \times m$ matrix. Shown in Ref. [17], the matrix-form generating function of $\mathbf{P}^{[d]}(w, l)$, i.e. $\mathbf{P}^{[d]*}(z, l) = \sum_{w=0}^{\infty} \mathbf{P}^{[d]}(w, l) z^w$ has the following form:

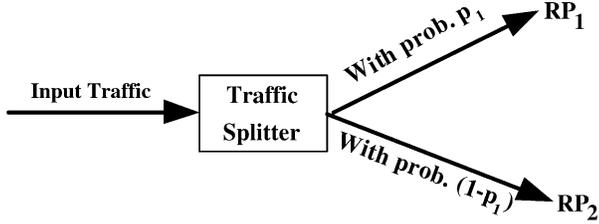
$$\mathbf{P}^{[d]*}(z, l) = [\mathbf{D}^{[d]}(z)]^l, \quad (5)$$

where

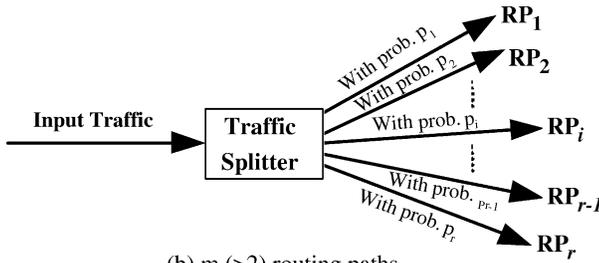
$$\mathbf{D}^{[d]}(z) = \sum_{k=0}^{\infty} \mathbf{D}_k^{[d]} z^k. \quad (6)$$

3. Split traffic models under a probabilistic routing policy

In this section, we investigate traffic splitting under a probabilistic routing policy to get split traffic models. When a node receives an incoming packet to be routed to downstream nodes, one may imagine that a traffic splitter (see Fig. 2) at each node is invoked to decide to which outgoing path the packet should be forwarded. In the following, we use p_i to denote the routing probability for routing path RP_i , which can be interpreted as the percentage of packets assigned to routing path RP_i with $\sum_{i=1}^r p_i = 1$, where r



(a) Two routing paths.



(b) m (≥2) routing paths.

Fig. 2. Scenarios of traffic splitting.

possible paths are assumed. Shown in Figs. 2(a) and 2(b) are a traffic splitter of two routing paths and a traffic splitter of multiple routing paths. As we mentioned in the previous section, the input processes of packets to be investigated include both BMAP and D-BMAP; thus, we derive split traffic models correspond to these two input processes in the following two subsections, respectively.

3.1. Split traffic models with the BMAP input

Let us first study the relationship among split (output) traffic, routing probabilities, and the original BMAP input traffic, which is illustrated in the following theorem.

Theorem 1. Suppose an input traffic stream of packets is modeled by an m -state BMAP. Now route these input packets to a random path chosen from two routing paths RP_1 and RP_2 with probabilities p_1 and p_2 , $p_1 + p_2 = 1$, respectively. Then the traffic streams on both routing paths are still BMAPs and the routed/split traffic on routing path RP_i can be described using rate matrices $\mathbf{D}_q^{[c],i}$, $q \geq 0$, where

$$\mathbf{D}_q^{[c],i} = \sum_{k=q}^{\infty} \mathbf{D}_k^{[c]} \binom{k}{q} (1-p_i)^{k-q} p_i^q, \quad i = 1, 2.$$

Proof. Let us first define $N_{RP_1}^{[c]}(t)$ and $J_{RP_1}^{[c]}(t)$ to be the random variable to represent the number of packets observed during the time period $(0, t]$ and the random variable to indicate the phase/state of the packet stream at time t , respectively, on routing path RP_1 . Similar to the definition of Eq. (2), we can define the conditional counting density function on routing path RP_1 , i.e. $[\mathbf{P}_{RP_1}^{[c]}(n_1, t)]_{ij}$ as $\Pr\{N_{RP_1}^{[c]}(t) = n_1, J_{RP_1}^{[c]}(t) = j | N_{RP_1}^{[c]}(0) = 0, J_{RP_1}^{[c]}(0) = i\}$.

Then we can relate $[\mathbf{P}_{RP_1}^{[c]}(n_1, t)]_{ij}$ to $[\mathbf{P}^{[c]}(n, t)]_{ij}$ (for the incoming path) as follows:

$$[\mathbf{P}_{RP_1}^{[c]}(n_1, t)]_{ij} = \sum_{n=n_1}^{\infty} [\mathbf{P}^{[c]}(n, t)]_{ij} \frac{n!}{n_1!(n-n_1)!} \times p_1^{n_1} (1-p_1)^{n-n_1}. \quad (7)$$

In Eq. (7), the binomial distribution is applied due to the random path selection and this entry-wise form leads to the following matrix form:

$$\mathbf{P}_{RP_1}^{[c]}(n_1, t) = \sum_{n=n_1}^{\infty} \mathbf{P}^{[c]}(n, t) \frac{n!}{n_1!(n-n_1)!} \times p_1^{n_1} (1-p_1)^{n-n_1}. \quad (8)$$

By figuring out the matrix-form generating function of $\mathbf{P}_{RP_1}^{[c]}(n_1, t)$, we have a compact expression as demonstrated in the following:

$$\mathbf{P}_{RP_1}^{[c]*}(z, t) = \sum_{n_1=0}^{\infty} \mathbf{P}_{RP_1}^{[c]}(n_1, t) z^{n_1} \quad (9)$$

$$= \sum_{n_1=0}^{\infty} \left[\sum_{n=n_1}^{\infty} \mathbf{P}^{[c]}(n, t) \frac{n!}{n_1!(n-n_1)!} \times p_1^{n_1} (1-p_1)^{n-n_1} \right] z^{n_1} \quad (10)$$

$$\mathbf{P}_{RP_1}^{[c]*}(z, t) = \sum_{n_1=0}^{\infty} \left[\sum_{n_1=0}^n \mathbf{P}^{[c]}(n, t) \frac{n!}{n_1!(n-n_1)!} \times (p_1 z)^{n_1} (1-p_1)^{n-n_1} \right] \quad (11)$$

$$\mathbf{P}_{RP_1}^{[c]*}(z, t) = \sum_{n=0}^{\infty} \mathbf{P}^{[c]}(n, t) [1 - p_1(1-z)]^n \quad (12)$$

$$\mathbf{P}_{RP_1}^{[c]*}(z, t) = \mathbf{P}^{[c]*}(1 - p_1(1-z), t) \quad (13)$$

$$\mathbf{P}_{RP_1}^{[c]*}(z, t) = e^{\mathbf{D}^{[c]}(1-p_1+p_1z)t}. \quad (14)$$

The part of the exponent in Eq. (14), i.e. $\mathbf{D}^{[c]}(1 - p_1 + p_1z)$ can be rearranged as follows:

$$\mathbf{D}^{[c]}(1 - p_1 + p_1z) = \sum_{k=0}^{\infty} \mathbf{D}_k^{[c]} (1 - p_1 + p_1z)^k \quad (15)$$

$$= \sum_{k=0}^{\infty} \mathbf{D}_k^{[c]} \sum_{q=0}^k \binom{k}{q} (1-p_1)^{k-q} (p_1z)^q \quad (16)$$

$$\mathbf{D}^{[c]}(1 - p_1 + p_1z) = \sum_{q=0}^{\infty} \left[\sum_{k=q}^{\infty} \mathbf{D}_k^{[c]} \binom{k}{q} (1-p_1)^{k-q} p_1^q \right] z^q \quad (17)$$

$$\mathbf{D}^{[c]}(1 - p_1 + p_1 z) = \sum_{q=0}^{\infty} \mathbf{D}_q^{[c],1} z^q, \quad (18)$$

where

$$\mathbf{D}_q^{[c],1} = \sum_{k=q}^{\infty} \mathbf{D}_k^{[c]} \binom{k}{q} (1 - p_1)^{k-q} p_1^q, \quad q \geq 0.$$

The above rearrangement is briefly explained as follows: Eq. (15) comes from the definition of the matrix z -transform; applying the skill of variable change to Eq. (16) results in Eq. (17). From Eq. (18), the traffic stream on routing path RP_1 is a BMAP with rate matrices

$$\mathbf{D}_q^{[c],1} = \sum_{k=q}^{\infty} \mathbf{D}_k^{[c]} \binom{k}{q} (1 - p_1)^{k-q} p_1^q, \quad q \geq 0$$

since it is the matrix z -transform of a BMAP with the above rate matrices. Similarly, the traffic stream on the other routing path can be shown to be a BMAP with rate matrices

$$\mathbf{D}_q^{[c],2} = \sum_{k=q}^{\infty} \mathbf{D}_k^{[c]} \binom{k}{q} (1 - p_2)^{k-q} p_2^q, \quad q \geq 0.$$

This completes the proof of the theorem. \square

We can easily extend the result in Theorem 1 to multiple routing paths as the following corollary.

Corollary 1. *Suppose an input traffic stream of packets is modeled by an m -state BMAP. Now route these input packets to a random routing path RP_i ($1 \leq i \leq r$) among r routing paths with probability p_i ($\sum_{i=1}^r p_i = 1$). Then traffic streams on routing path RP_i , $i = 1, 2, \dots, r$, are still BMAPs with rate matrices $\mathbf{D}_q^{[c],i}$, $q \geq 0$, where*

$$\mathbf{D}_q^{[c],i} = \sum_{k=q}^{\infty} \mathbf{D}_k^{[c]} \binom{k}{q} (1 - p_i)^{k-q} p_i^q.$$

Proof. Treating multiple routing paths as two routing paths: routing path RP_i with routing probability p_i and the other virtual routing path with routing probability $1 - p_i$ accounting for routing packets to paths other than path RP_i and applying Theorem 1, we can prove that the traffic stream on routing path RP_i is still a BMAP with rate matrices $\mathbf{D}_q^{[c],i}$, $q \geq 0$, where

$$\mathbf{D}_q^{[c],i} = \sum_{k=q}^{\infty} \mathbf{D}_k^{[c]} \binom{k}{q} (1 - p_i)^{k-q} p_i^q.$$

Thus the proof of the corollary is completed. \square

Corollary 2. *Consider an MAP which is a special case of BMAP with maximum batch size of 1 denoted by a pair of rate matrices $(\mathbf{D}_0^{[c]}, \mathbf{D}_1^{[c]})$ as the input traffic stream. Under the probabilistic routing policy, route packets of such a traffic stream to a random routing path RP_i ($1 \leq i \leq r$)*

among r routing paths with probability p_i ($\sum_{i=1}^r p_i = 1$). Then the traffic stream on routing path RP_i is again an MAP with the pair of rate matrices $(\mathbf{D}^{[c]} - p_i \mathbf{D}_1^{[c]}, p_i \mathbf{D}_1^{[c]})$, where $\mathbf{D}^{[c]} = \mathbf{D}_0^{[c]} + \mathbf{D}_1^{[c]}$.

Proof. Since $\mathbf{D}_k^{[c]}$, $k \geq 2$ vanish, i.e. they are zero matrices, we then have the following equations from Corollary 1.

$$\mathbf{D}_0^{[c],i} = \sum_{k=0}^{\infty} \mathbf{D}_k^{[c]} \binom{k}{0} (1 - p_i)^k = \mathbf{D}^{[c]} - p_i \mathbf{D}_1^{[c]}, \quad (19)$$

$$\mathbf{D}_1^{[c],i} = \sum_{k=1}^{\infty} \mathbf{D}_k^{[c]} \binom{k}{1} (1 - p_i)^{k-1} p_i = p_i \mathbf{D}_1^{[c]}, \quad (20)$$

$$\mathbf{D}_q^{[c],i} = \sum_{k=q}^{\infty} \mathbf{D}_k^{[c]} \binom{k}{q} (1 - p_i)^{k-q} p_i^q = \mathbf{0}, \quad q \geq 2. \quad (21)$$

where $\mathbf{0}$ is an $m \times m$ zero matrix. Eqs. (19) and (20) prove that the traffic stream on routing path RP_i is again an MAP with rate matrices $(\mathbf{D}^{[c]} - p_i \mathbf{D}_1^{[c]}, p_i \mathbf{D}_1^{[c]})$. This completes the proof of the corollary. \square

Finally, we discuss the correctness of the above results using the following remark.

Remark 1. If we narrow the result of Corollary 2 to an m -state MMPP input with traffic descriptor $(\mathbf{Q}, \mathbf{\Lambda})$ (see Refs. [12,13]), i.e. an MAP with rate matrices $(\mathbf{Q} - \mathbf{\Lambda}, \mathbf{\Lambda})$ since an MMPP is a special case of an MAP, then we can obtain the traffic stream on routing path RP_i is an MAP with rate matrices $(\mathbf{Q} - p_i \mathbf{\Lambda}, p_i \mathbf{\Lambda})$, i.e. an MMPP with traffic descriptor $(\mathbf{Q}, p_i \mathbf{\Lambda})$. The above result is the same as that obtained in our previous work [12]. This validates the correctness of the above theorem and corresponding corollaries.

3.2. Split traffic models with the D-BMAP input

Now, we turn our attention to traffic splitting under a probabilistic routing policy with the D-BMAP input. Let us begin with the case of two routing paths in the following theorem.

Theorem 2. *Consider a D-BMAP input stream of packets with transition matrices, $\mathbf{D}_k^{[d]}$, $k \geq 0$ and route these input packets modeled by the D-BMAP to routing paths RP_1 and RP_2 with probabilities p_1 and p_2 ($p_1 + p_2 = 1$), respectively. Then the traffic streams on both routing paths are still D-BMAPs expressed by transition matrices*

$$\mathbf{D}_q^{[d],i} = \sum_{k=q}^{\infty} \mathbf{D}_k^{[d]} \binom{k}{q} (1 - p_i)^{k-q} p_i^q$$

for routing path RP_i , $i = 1, 2$.

Proof. Similar to Theorem 1 for the BMAP input, we first define $N_{RP_1}^{[d]}(0, n)$ and $J_{RP_1}^{[d]}(n)$ to be the random variable to represent the number of packets observed between 0th and n th slot and the random variable to indicate the phase of the input process at slot n , respectively, for routing path RP_1 . We then define the *conditional counting density function* for routing path RP_1 , i.e. $[\mathbf{P}_{RP_1}^{[d]}(w_1, l)]_{ij}$ as $\Pr\{N_{RP_1}^{[d]}(0, l) = w_1, J_{RP_1}^{[d]}(l) = j | N_{RP_1}^{[d]}(0, 0) = 0, J_{RP_1}^{[d]}(0) = i\}$. Then we can relate $[\mathbf{P}_{RP_1}^{[d]}(w_1, l)]_{ij}$ to $[\mathbf{P}^{[d]}(w, l)]_{ij}$ of the incoming path as follows:

$$[\mathbf{P}_{RP_1}^{[d]}(w_1, l)]_{ij} = \sum_{w=w_1}^{\infty} [\mathbf{P}^{[d]}(w, l)]_{ij} \frac{w!}{w_1!(w-w_1)!} \times p_1^{w_1} (1-p_1)^{w-w_1}. \quad (22)$$

We can rewrite the above entry-wise form into the matrix form as follows:

$$\mathbf{P}_{RP_1}^{[d]}(w_1, l) = \sum_{w=w_1}^{\infty} \mathbf{P}^{[d]}(w, l) \frac{w!}{w_1!(w-w_1)!} \times p_1^{w_1} (1-p_1)^{w-w_1}. \quad (23)$$

Using the same approaches as those in Eqs. (7)–(14), we have the following matrix-form generating function for $\mathbf{P}_{RP_1}^{[d]}(w_1, l)$ as follows:

$$\mathbf{P}_{RP_1}^{[d]*}(z, l) = [\mathbf{D}^{[d]}(1-p_1+p_1z)]^l. \quad (24)$$

Following the same philosophy for Eqs. (15)–(18), we can rewrite $\mathbf{D}^{[d]}(1-p_1+p_1z)$ in Eq. (24) as follows:

$$\mathbf{D}^{[d]}(1-p_1+p_1z) = \sum_{q=0}^{\infty} \mathbf{D}_q^{[d],1} z^q, \quad (25)$$

where

$$\mathbf{D}_q^{[d],1} = \sum_{k=q}^{\infty} \mathbf{D}_k^{[d]} \binom{k}{q} (1-p_1)^{k-q} p_1^q.$$

This proves the theorem for routing path RP_1 . Using an analogous manner, we can prove the other part for routing path RP_2 . Hence, the proof of this theorem is completed. \square

For the multiple routing paths, the result is given in the following corollary.

Corollary 3. *Considering a D-BMAP with transition matrices $\mathbf{D}_k^{[d]}$, $k \geq 0$, the traffic stream on routing path RP_i is again a D-BMAP with transition matrices*

$$\mathbf{D}_q^{[d],i} = \sum_{k=q}^{\infty} \mathbf{D}_k^{[d]} \binom{k}{q} (1-p_i)^{k-q} p_i^q$$

when routing packets to routing path RP_i , $1 \leq i \leq r$ among r routing paths with probability p_i ($\sum_{i=1}^r p_i = 1$) under a probabilistic routing policy.

Proof. Since the proof can be proven analogous to Corollary 1, it is omitted here. \square

Corollary 4. *Considering a D-MAP which is a special case of D-BMAP with maximum batch size of 1 denoted by a pair of transition matrices $(\mathbf{D}_0^{[d]}, \mathbf{D}_1^{[d]})$, the traffic stream on routing path RP_i is also a D-MAP with the pair of transition matrices $(\mathbf{D}^{[d]} - p_i \mathbf{D}_1^{[d]}, p_i \mathbf{D}_1^{[d]})$, where $\mathbf{D}^{[d]} = \mathbf{D}_0^{[d]} + \mathbf{D}_1^{[d]}$ when routing packets to routing path RP_i , $1 \leq i \leq r$ among r routing paths with probability p_i ($\sum_{i=1}^r p_i = 1$) under a probabilistic routing policy.*

Proof. This corollary can be proven similar to Corollary 2. Hence, we omit the proof. \square

In Refs. [4,26], the property that aggregation/merge of independent heterogeneous D-BMAPs (BMAPs) results in again a D-BMAP (BMAP) has been explored. In this paper, we investigate the property that split of a BMAP (D-BMAP) following a probabilistic routing policy is also a BMAP (D-BMAP) for each routing path as mentioned above. Therefore, both of these two properties enable us to get an easier method when dealing with network-wise performance evaluation.

4. An application to network-wise performance analysis

In the following, we apply the results of traffic splitting under a probabilistic routing policy to an application of calculating the network-wise performance along with characterization of output processes given in Ref. [9] or Ref. [11], including performance measures of end-to-end sojourn delay, delay variance, and packet/cell loss probability.

4.1. Calculation of performance metrics

In networks with the probabilistic routing algorithm, packets may traverse networks through different paths between the source node (SN) and the destination node (DN). For example, there are four available/possible routing paths from SN to DN in Fig. 1(a), i.e. path 1: $SN \rightarrow I_{(0,1)} \rightarrow I_{(0,2)} \rightarrow DN$; path 2: $SN \rightarrow I_{(0,1)} \rightarrow I_{(1,2)} \rightarrow DN$; path 3: $SN \rightarrow I_{(1,1)} \rightarrow I_{(0,2)} \rightarrow DN$; path 4: $SN \rightarrow I_{(1,1)} \rightarrow I_{(1,2)} \rightarrow DN$. In the following derivations, we assume that there are r possible routing paths and n_i nodes along routing path RP_i with routing probability p_i , e.g. there are four nodes along the path shown in Fig. 1(b), i.e. $SN \rightarrow I_{(0,1)} \rightarrow I_{(1,2)} \rightarrow DN$. Let T_{EE} , $T_{EE}^{RP_i}$, and $T_j^{RP_i}$ denote, respectively, the random variable of sojourn delay incurred from SN to DN, the random variable of the end-to-end sojourn delay of routing path RP_i , and the random variable of sojourn delay at the j th node of routing path RP_i . Then, we can relate the mean end-to-end sojourn delay $E[T_{EE}]$ from SN to DN to mean path

delay $E[T_{EE}^{RP_i}]$ and mean nodal delay $E[T_j^{RP_i}]$ as follows:

$$E[T_{EE}] = \sum_{i=1}^r E[T_{EE}^{RP_i}] p_i = \sum_{i=1}^r \left(\sum_{j=1}^{n_i} E[T_j^{RP_i}] \right) p_i. \quad (26)$$

The first equality of Eq. (26) is just an ensemble of end-to-end sojourn delays along all possible routing paths and the second equality holds by using the relation

$$E[T_{EE}^{RP_i}] = \sum_{j=1}^{n_i} E[T_j^{RP_i}].$$

As for the variance of the end-to-end sojourn delay from SN to DN, it can be calculated as follows:

$$\begin{aligned} \text{Var}[T_{EE}] = & \sum_{i=1}^r \text{Var}[T_{EE}^{RP_i}] p_i = \sum_{i=1}^r \left[\sum_{j=1}^{n_i} \text{Var}[T_j^{RP_i}] \right. \\ & \left. + 2 \sum_{j=1}^{n_i-1} \sum_{k=j+1}^{n_i} \text{Cov}(T_j^{RP_i}, T_k^{RP_i}) \right] p_i. \end{aligned} \quad (27)$$

Although Eq. (27) exactly characterizes the variance of end-to-end sojourn delay from SN to DN, it requires more information on the correlation of two sojourn delays at intermediate nodes. However, it is difficult to get such information in a real network. Instead, we use an approximate method by neglecting the correlation of two sojourn delays at intermediate nodes to get the following approximate estimate for the variance of end-to-end sojourn delay.

$$\text{Var}[T_{EE}] \approx \sum_{i=1}^r \left(\sum_{j=1}^{n_i} \text{Var}[T_j^{RP_i}] \right) p_i. \quad (28)$$

In current or future networks, the packet/cell loss probability is another important performance metric other than delay or delay variance, especially the end-to-end loss probability from SN to DN. Thus, we discuss the calculation of the end-to-end cell loss probability incurred by a cell from SN to DN in the following. Let $P_{\text{loss,EE}}$, $P_{\text{loss,EE}}^{RP_i}$, and $P_{\text{loss},j}^{RP_i}$ denote, respectively, the end-to-end cell loss probability incurred by a cell from SN to DN, the end-to-end cell loss probability of routing path RP_i , and the cell loss probability occurring at the j th node of routing path RP_i . It is clear that $1 - P_{\text{loss},j}^{RP_i}$ is the non-blocking probability for a cell at j th node of routing path RP_i and the product term $\prod_{i=1}^{n_i} (1 - P_{\text{loss},j}^{RP_i})$ means the probability that a cell successfully gets through routing path RP_i . Therefore, the end-to-end cell loss probability of routing path RP_i can be represented as follows:

$$P_{\text{loss,EE}}^{RP_i} = 1 - \prod_{i=1}^{n_i} (1 - P_{\text{loss},j}^{RP_i}) \quad (29)$$

which leads to the following end-to-end cell loss probability incurred by a cell from SN to DN.

$$P_{\text{loss,EE}} = \sum_{i=1}^r p_i P_{\text{loss,EE}}^{RP_i}. \quad (30)$$

4.2. The procedure for the end-to-end performance evaluation

Using the methods of calculation mentioned in previous subsection together with nodal performance analysis, e.g. those given in Refs. [4,10,13,26], we propose an algorithm to calculate the network-wide performance for a network employing probabilistic routing policies. The detailed procedure written in a pseudo code is illustrated in Fig. 3.

```

01 Begin
02   Given relevant traffic, service parameters.
03   Construct network topology and index each node.
04   Given routing probabilities.
05   For  $i:=1$  to  $r$  do // Assume there are  $r$  available
        routing paths.
06     Begin
07       Choose-Path( $RP_1, RP_2, \dots, RP_i, \dots, RP_r$ )
        // Choose a path from available
        routing paths.
08       Link-Perf( $RP_i$ )
        // Calculate the link performance.
09     End
10     End-to-end-Perf( $RP_i$ )
        // Calculate the end-to-end performance.
11   End
12
13
14   Function Link-Perf( $RP_i$ )
15     Begin
16       Label SN(0, 0)
        // Label source node with index (0, 0).
17       Label DN(0,  $n_i-1$ )
        // Label destination node with index (0,  $n_i-1$ ).
18       Label the other intermediate nodes ( $i, j$ )
19       NI=( $u, v$ )=(0, 0)
20       Nodal-Perf( $u, v$ )
        // Calculate the nodal performance.
21     End
22
23
24   Function Nodal-Perf( $u, v$ )
25     Begin
26     Do
27       Begin
28         Decomposition( )
        // Collect the tagged departure statistics.
29         Traffic Splitting( )
        // Traffic splitting.
30         Moment Matching( )
        // Emulate the output process.
31         NI=( $u, v$ )
        //  $u$  is the next node of the chosen path;
         $v=v+1$ .
32       End
33     Until NI(  $u, v$  ) = (0,  $n_i-1$ )
34     End

```

Fig. 3. Pseudo code for the end-to-end network-wide performance evaluation.

5. Numerical results and discussions

In this section, some numerical experiments are done to show the accuracy of the network-wise performance analysis method proposed in Section 4. The phenomena to be observed include at least the effect of traffic characteristics and the effect of the probabilistic routing policy. In the following experiments, Matlab ver. 6.5 is used for analytical calculation and Microsoft Visual C++ is employed to run simulation programs on IBM compatible PCs. As for the network topology, a 2×2 mesh subnetwork for experiments (See Fig. 1(a)) is used for illustrative purpose. Each node which is modeled by a single server queue with a probabilistic traffic splitter routing packets probabilistically to successive nodes receives two/three input traffic streams with external streams modeled by MMPPs. Besides, different service types are allowed for each node, including deterministic, exponential, and k -stage Erlangian service distributions (see Ref. [20]), respectively. For convenience, we use short-hands D, Exp., and E_k , respectively, to denote the above three service types of distributions.

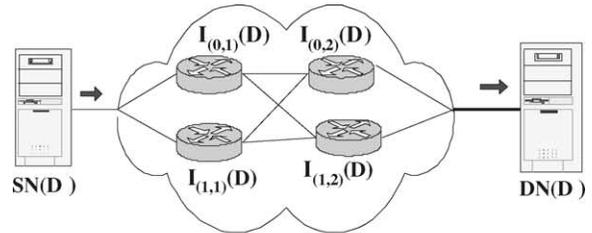
5.1. Experimental arrangement

$$\mathbf{M}_{\text{COLS}} = \begin{matrix} & \text{SN} & I_{(0,1)} & I_{(0,2)} & I_{(1,1)} & I_{(1,2)} & \text{DN} \\ \text{SN} & \begin{bmatrix} 0.00 & 0.97 & 0.00 & 0.03 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.97 & 0.00 & 0.03 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \\ 0.00 & 0.00 & 0.97 & 0.00 & 0.03 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{bmatrix} & , & (31) \end{matrix}$$

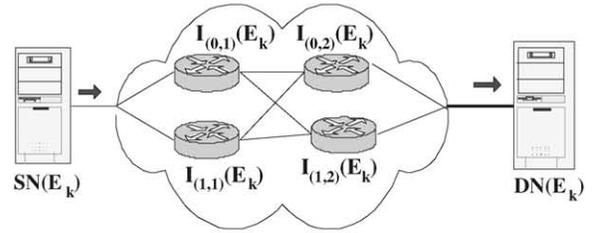
$$\mathbf{M}_{\text{FRS}} = \begin{matrix} & \text{SN} & I_{(0,1)} & I_{(0,2)} & I_{(1,1)} & I_{(1,2)} & \text{DN} \\ \text{SN} & \begin{bmatrix} 0.00 & 0.50 & 0.00 & 0.50 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.50 & 0.00 & 0.50 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \\ 0.00 & 0.00 & 0.50 & 0.00 & 0.50 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{bmatrix} & , & (32) \end{matrix}$$

$$\mathbf{M}_{\text{RRS}} = \begin{matrix} & \text{SN} & I_{(0,1)} & I_{(0,2)} & I_{(1,1)} & I_{(1,2)} & \text{DN} \\ \text{SN} & \begin{bmatrix} 0.00 & 0.22 & 0.00 & 0.78 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.68 & 0.00 & 0.32 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \\ 0.00 & 0.00 & 0.14 & 0.00 & 0.86 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{bmatrix} & . & (33) \end{matrix}$$

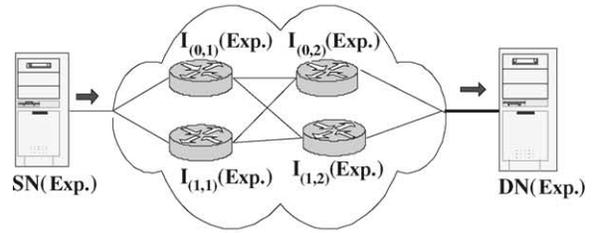
Two types of queueing models, i.e. finite-buffer queueing model and infinite-buffer queueing model are used in the following numerical experiments. Let us first focus on the finite-buffer queueing model. Shown in Fig. 4 are some of possible service arrangements in a 2×2 mesh subnetwork, including all D servers, all E_k servers, all Exp. servers,



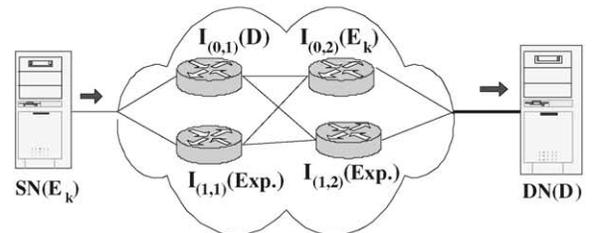
(a) A network with all D servers.



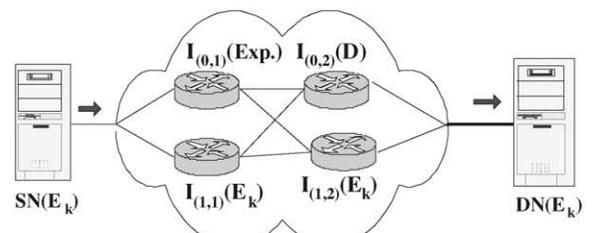
(b) A network with all E_k servers.



(c) A network with all Exp. servers.



(d) A network with mixed servers.



(e) A network with mixed servers.

Fig. 4. Mesh networks with different arrangements of servers.

and mixed types of servers with E_4 at SN, D at $I_{(0,1)}$, E_4 at $I_{(0,2)}$, Exp. at $I_{(1,1)}$, Exp. at $I_{(1,2)}$, and D at DN. Routing probability matrices given in Eqs. (31)–(33) are chosen to reflect different routing scenarios: the routing probability matrix \mathbf{M}_{COLS} given in Eq. (31) routing almost all packets to path $\text{SN} \rightarrow I_{(0,1)} \rightarrow I_{(0,2)} \rightarrow \text{DN}$ to approximate a connection-oriented-like scheme (COLS), \mathbf{M}_{FRS} in Eq. (32) to represent a symmetric and fair routing scheme (FRS) with respect to the topology of the subnetwork, and \mathbf{M}_{RRS} in Eq. (33) used to stand for a random routing scheme (RRS). In this paper, constant/fixed probabilistic routing models are considered because some simple dynamic probabilistic traffic splitting models, e.g. those with pre-defined or predictable pattern of time period during which a specific constant probabilistic traffic splitting is employed may be developed based on the constant probabilistic traffic splitting model. But more general dynamic probabilistic traffic splitting still requires non-trivial effort. Hence, this part of work deserves to be the future work. Due to the simplicity and tractability, we use heterogeneous two-state MMPPs which are parameterized by a four-tuple traffic descriptor $(\lambda_1, \lambda_2, \sigma_1, \sigma_2)$. Moreover, we feed two external MMPPs to node SN with one representing the tagged packet stream and the other denoting the cross/interfering packet stream and only an external MMPP to other nodes, which is used to model the cross/interfering packet stream to simplify the system model. As for traffic parameter arrangement, we set the original tagged MMPP input at SN to (0.070, 0.025, 0.120, 0.157) and the aggregated cross traffic at each node according to parameters given in Table 1 in which all nodes are indexed using a (row, column) pair, i.e. (SN, $I_{(0,1)}$, $I_{(0,2)}$, $I_{(1,1)}$, $I_{(1,2)}$, DN) = ((0, 0), (0, 1), (0, 2), (1, 1), (1, 2), (0, 3)) for the network in Fig. 1.

5.2. Network-wise performance for a mesh network with probabilistic routing

Fixing the capacity for each server to 10 (i.e. $K = 10$), we first observe the end-to-end performance of a mesh

network under a probability routing policy using the mesh network topology and the related information shown in Figs. 4(a)–4(d). Via Eqs. (26), (28), and (30), we get the end-to-end delay, delay variance, and cell loss probability under RRS of Eq. (33) as shown in Fig. 5. From Fig. 5, the following observations are drawn: (i) it is obvious that the end-to-end delay, delay variance, and cell loss probability increase as the cross traffic load increases; (ii) quantities of all metrics for an Exp. server are the largest while they are smallest for a D server because of the squared coefficient of variation for the Exp. server is the largest and it is the smallest for the D server [20]; (iii) the end-to-end delay is not sensitive to service types when the cross traffic load is extremely low or extremely high (see Fig. 5(a)) while the end-to-end delay variance (end-to-end cell loss probability) is not sensitive to service types only when the cross traffic load is extremely low (extremely high) from Fig. 5(b) (Fig. 5(c)). But they are sensitive to service types within the range of middle traffic load.

With similar parameters arrangements except $\rho_c = 0.6$ and varying the buffer size of each node, we obtain the above three performance measures in Fig. 6 and conclude that (i) a sharper increasing trend exists when the buffer size is small, i.e. 2–5, whereas the trend becomes smooth as the buffer size goes up further; (ii) from Fig. 6(c), we notice that the end-to-end cell loss probability decreases (approximately) linearly in the semi-log plot as the buffer size increases.

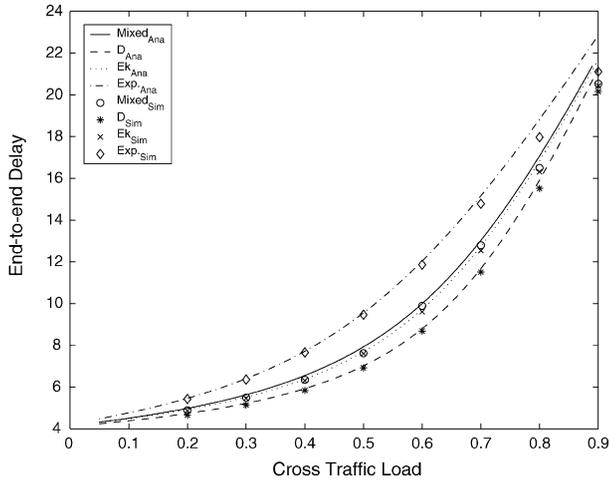
In Fig. 7, we show the effect of different routing probability arrangements upon the three performance measures. The parameter setting is the same as that of Fig. 5 except the routing probability matrices which may be varied according to COLS, FRS, and RRS. Although no significant difference among the three routing probability arrangements is observed in Figs. 7(a) and 7(b) for the end-to-end delay and delay variance, the end-to-end cell loss probability in Fig. 7(c) for COLS is larger than that for FRS or RRS. This phenomenon somewhat explains the need of load balance.

Table 1
Traffic parameters employed in the experiments

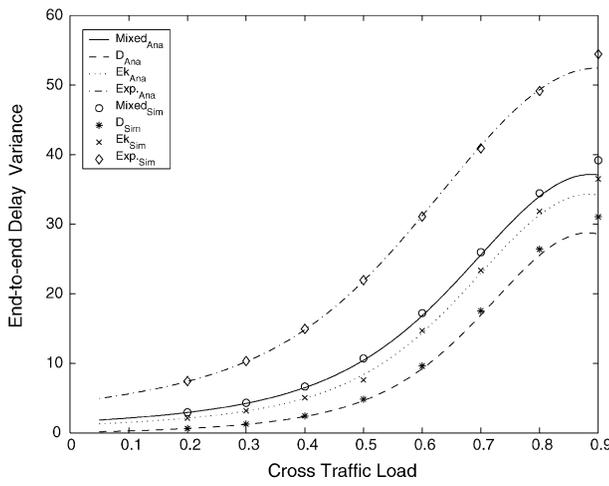
	SN	$I_{(0,1)}$	$I_{(0,2)}$	DN
Tagged traffic	(0.070, 0.025, 0.120, 0.157)	–	–	–
Cross traffic	$(\lambda_{1c}, \lambda_{2c}, \sigma_{1c}, \sigma_{2c})$	$(\lambda_{1c}, \lambda_{2c}, \sigma_{1c}, \sigma_{2c})$	$(\lambda_{1c}, \lambda_{2c}, \sigma_{1c}, \sigma_{2c})$	$(\lambda_{1c}, \lambda_{2c}, \sigma_{1c}, \sigma_{2c})$
Cross traffic	–	$I_{(1,1)}$	$I_{(1,2)}$	–
Cross traffic	–	$(\lambda_{1c}, \lambda_{2c}, \sigma_{1c}, \sigma_{2c})$	$(\lambda_{1c}, \lambda_{2c}, \sigma_{1c}, \sigma_{2c})$	–
1.	$(\lambda_{1c}, \lambda_{2c}, \sigma_{1c}, \sigma_{2c}) = (\frac{\lambda W + 4\rho_c}{3}, \frac{\lambda W + 2\rho_c}{3}, \sigma W + \rho_c, \sigma W + \rho_c)$			
2.	λW (λ weight) = (row _{index} \times $\lambda_{\text{row-weight}}$) + (column _{index} \times $\lambda_{\text{column-weight}}$)			
3.	σW (σ weight) = (row _{index} \times $\sigma_{\text{row-weight}}$) + (column _{index} \times $\sigma_{\text{column-weight}}$)			
4.	$(\lambda_{\text{row-weight}}, \lambda_{\text{column-weight}}, \sigma_{\text{row-weight}}, \sigma_{\text{column-weight}}) = (0.0027, 0.005, 0.254, 0.310)$			
5.	The index of each node: (SN, $I_{(0,1)}$, $I_{(0,2)}$, $I_{(1,1)}$, $I_{(1,2)}$, DN) = ((0, 0), (0, 1), (0, 2), (1, 1), (1, 2), (0, 3))			

Using Fig. 8, we illustrate the effect of different parameters of arrival processes to the end-to-end delay, delay variance, and cell loss probability. With the same parameter setting of Fig. 5 except parameters σ_1 and σ_2 for

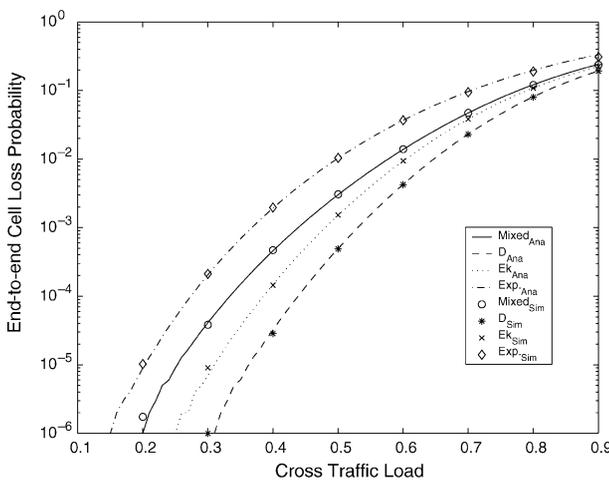
which σ_2 is set to 0.012, 0.0012, 0.00012, respectively, and the ratio of σ_1/σ_2 is kept to 3. From this figure, we see that the end-to-end performance measures increase as the sojourn time staying at each state (i.e. the inverses of σ_1



(a) End-to-end delay.

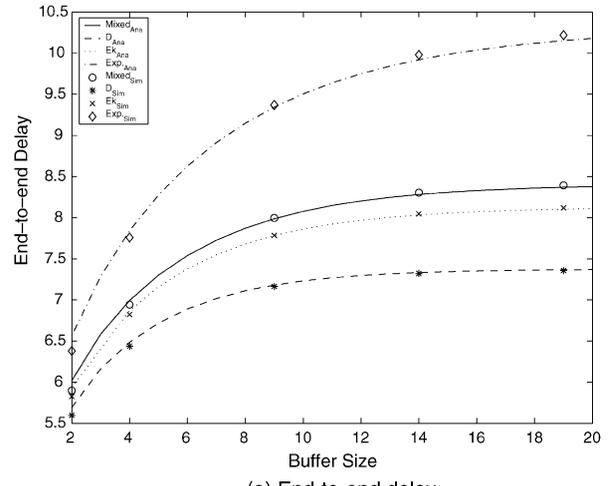


(b) End-to-end delay variance.

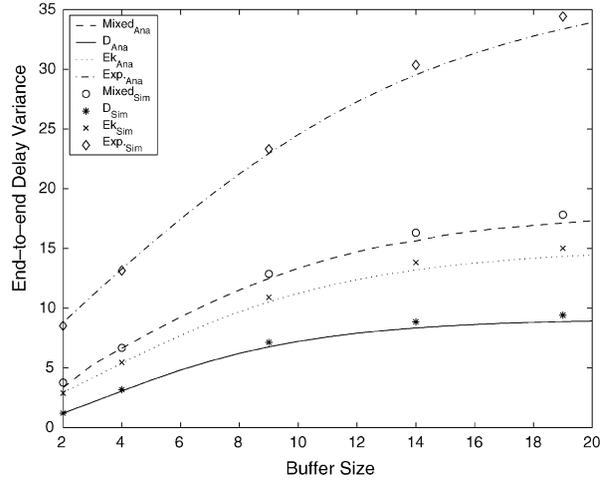


(c) End-to-end cell loss probability.

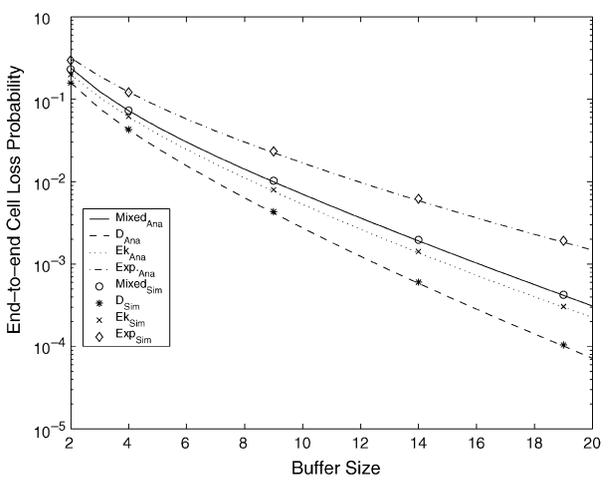
Fig. 5. Effects of the traffic load on performance measures.



(a) End-to-end delay.

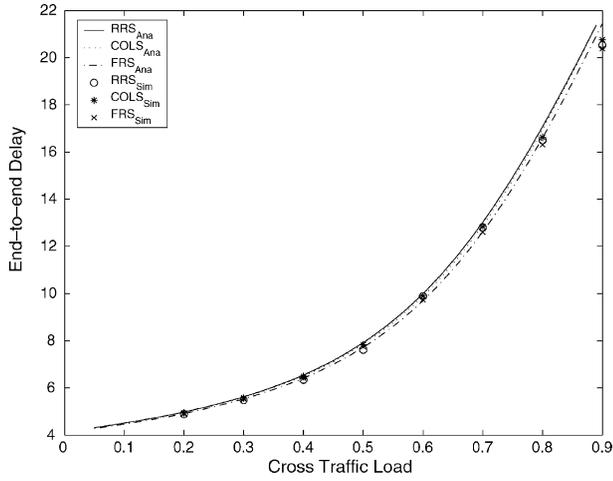


(b) End-to-end delay variance.

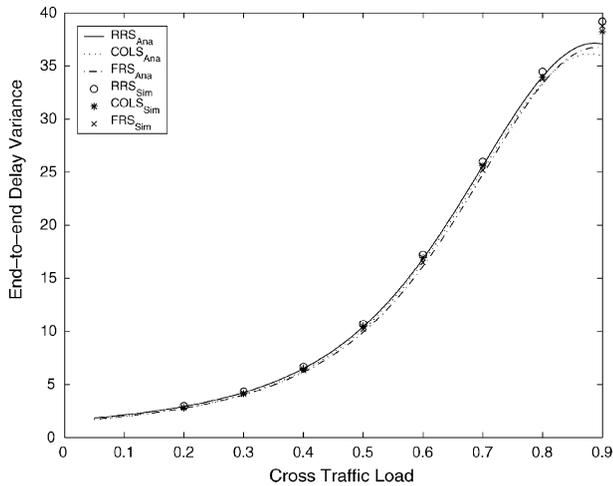


(c) End-to-end cell loss probability.

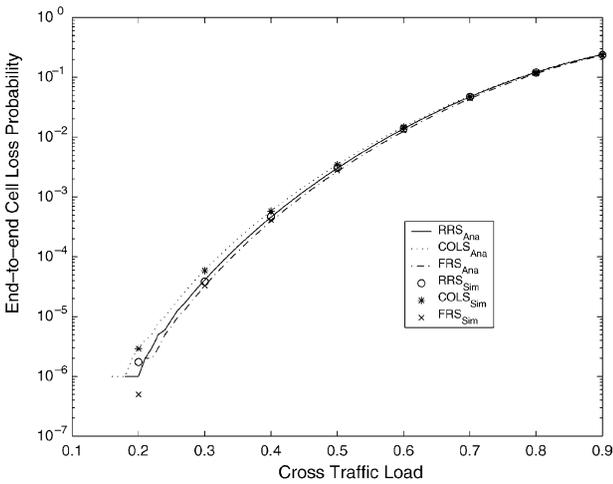
Fig. 6. Effects of the buffer size on performance measures.



(a) End-to-end delay.

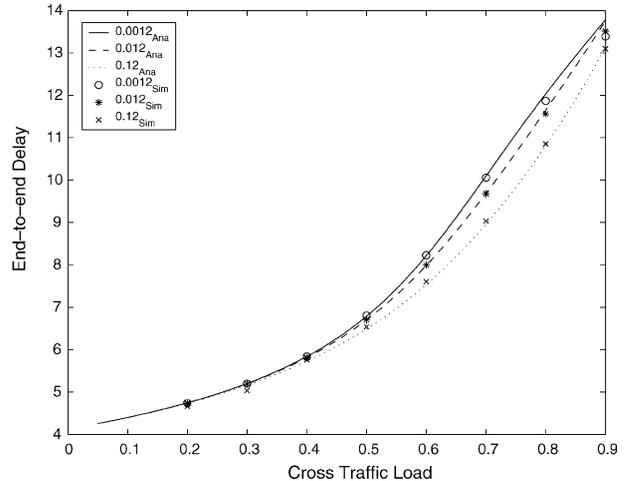


(b) End-to-end delay variance.

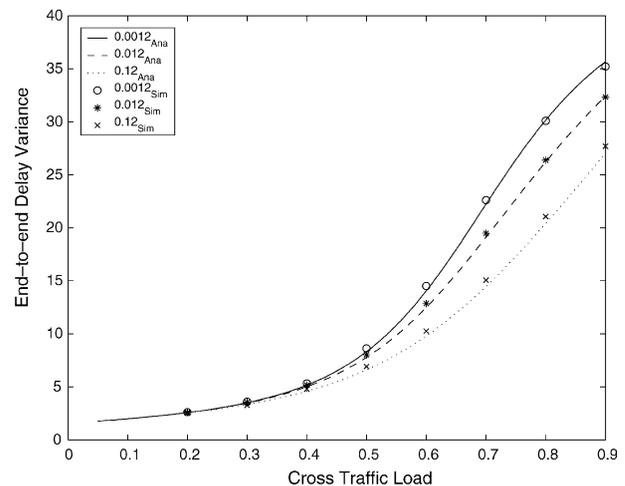


(c) End-to-end cell loss probability.

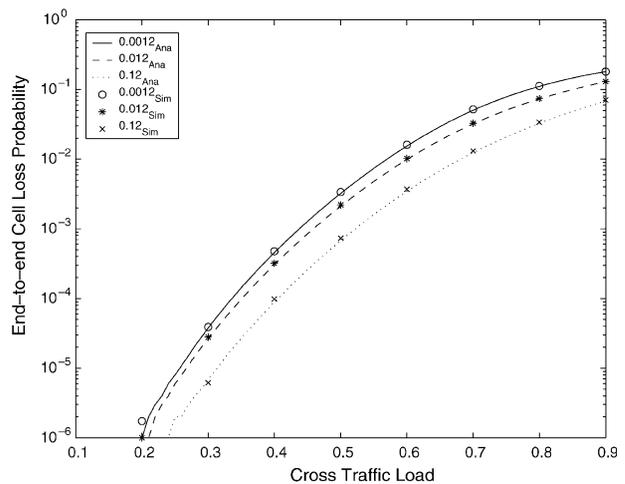
Fig. 7. Effects of different routing matrices on performance measures.



(a) End-to-end delay.



(b) End-to-end delay variance.



(c) End-to-end cell loss probability.

Fig. 8. Effects of the bursty duration on performance measures.

and σ_2) increases due to the fact that longer period staying at one state implies longer burst duration.

Shown in Fig. 9 is the end-to-end performance of a mesh network with a probability routing policy observed under

infinite-buffer queueing models with other parameters following those for Fig. 5. The mesh network topology and related information are shown in Fig. 4(e). From Fig. 9, we obtain similar results as those in Fig. 7, i.e. (i) both

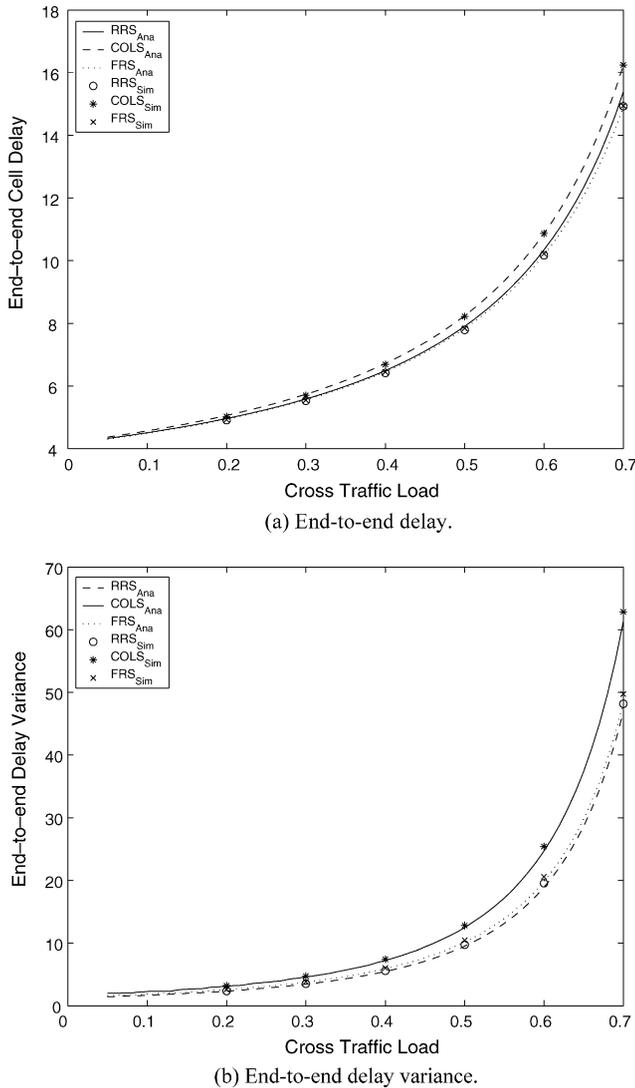


Fig. 9. Effects of different routing matrices on performance measures for the infinite-buffer queues.

Table 2

End-to-end delay, delay variance, and loss probability for Fig. 4(a) of all deterministic servers with capacity $K = 10$ using probability routing matrix of Eq. (33)

	SN	Intermediate Node (IN)				DN	End-to-end
		IN ₀₁	IN ₀₂	IN ₁₁	IN ₁₂		
<i>Delay</i>							
Simulation	1.7854	1.6352	1.6398	1.7165	1.7165	1.7069	6.9189
Analysis	1.7857	1.6395	1.6423	1.7326	1.7219	1.7549	7.0143
Error (%)	0.02	0.26	0.15	0.93	0.87	0.54	1.36
<i>Delay variance</i>							
Simulation	1.2874	0.9341	0.9410	1.1307	1.1055	1.1968	4.8300
Analysis	1.2891	1.7240	0.9409	1.1371	1.1117	1.1858	4.6354
Error (%)	0.13	0.23	0.01	0.56	0.56	0.93	4.20
<i>Cell loss probability</i>							
Simulation	0.0001810	0.0000550	0.0000540	0.0001050	0.0001020	0.0001260	0.000490
Analysis	0.0001816	0.0000543	0.0000551	0.0001097	0.0001008	0.0001262	0.000494
Error (%)	0.33	1.29	2.00	4.28	1.19	0.16	0.71

$\rho_c = 0.5$.

end-to-end delay and delay variance increase as cross traffic load goes up; (ii) performance with COLS is worse than FRS and RRS (This phenomenon for the infinite-buffer case is more obvious than the finite-buffer case.)

From the above observation, we noticed that the performance in Fig. 7 which considers the finite buffer queueing system seems to be independent of routing matrices except the cell loss probability when the cross traffic load is low. This is mainly due to the reason that finite buffers limit the size of a queue. Thus, little difference is observed for different routing matrices. However, the difference is magnified for the infinite buffer case as those shown in Fig. 9.

Finally, let us now examine the accuracy of the proposed method to evaluate the mesh network under a probabilistic routing policy. Using the parameters for Fig. 5, we show results in Tables 2–5 for cross traffic load $\rho_c = 0.5$. These tables show that (i) the accuracy for the end-to-end delay and delay variance is better than the accuracy for the cell loss probability; (ii) the worst case error shown in Table 2 for networks with all D servers is 4.28%. For networks with all E₄ servers, the worst case error shown in Table 3 is 9.15%. The worst case error for networks with all Exp. servers shown in Table 4 is 4.82%. Shown in Table 5, mixed servers has the worst case error of 8.72%. The above results demonstrate that the proposed method for evaluating network-wise performance performs good.

6. Conclusions

The split traffic models under a probabilistic routing policy are derived using a mathematical manner in this paper for both BMAP and D-BMAP input streams. It extends the results of Ref. [12] from MMPP input to BMAP and D-BMAP input. We demonstrate that the split traffic on each routing path from an original BMAP (D-BMAP) input

Table 3

End-to-end delay, delay variance, and loss probability for Fig. 4(b) of all E₄ servers with capacity $K = 10$ using probability routing matrix of Eq. (33)

	SN	Intermediate Node (IN)				DN	End-to-end
		IN ₀₁	IN ₀₂	IN ₁₁	IN ₁₂		
<i>Delay</i>							
Simulation	1.9589	1.7838	1.7899	1.8885	1.8772	1.9254	7.60410
Analysis	1.9599	1.7867	1.7900	1.8985	1.8859	1.9255	7.69328
Error (%)	0.05	0.16	0.01	0.53	0.46	0.01	1.16
<i>Delay variance</i>							
Simulation	2.2703	1.7614	1.7793	2.0546	2.0219	2.1604	7.6041
Analysis	2.2727	1.7672	1.7743	2.0642	2.0281	2.1363	8.3704
Error (%)	0.11	0.33	0.28	0.47	0.31	1.13	9.15
<i>Cell loss probability</i>							
Simulation	0.0005340	0.0001820	0.0001860	0.0003380	0.0003170	0.0004090	0.001529
Analysis	0.0005260	0.0001884	0.0001907	0.0003459	0.0003223	0.0003917	0.001516
Error (%)	1.52	3.40	2.46	2.28	1.64	4.42	0.85

 $\rho_c = 0.5$.

Table 4

End-to-end delay, delay variance, and loss probability for Fig. 4(c) of all exponential servers with capacity $K = 10$ using probability routing matrix of Eq. (33)

	SN	Intermediate Node (IN)				DN	End-to-end
		IN ₀₁	IN ₀₂	IN ₁₁	IN ₁₂		
<i>Delay</i>							
Simulation	2.4272	2.1982	2.2089	2.3533	2.3533	2.4109	9.46400
Analysis	2.4314	2.2009	2.2058	2.3563	2.3397	2.3936	9.55586
Error (%)	0.17	0.12	0.14	0.13	0.58	0.72	0.96
<i>Delay variance</i>							
Simulation	5.7509	4.8197	4.8655	5.6355	5.4131	5.3678	21.9458
Analysis	5.7587	4.8327	4.8490	5.4221	5.3566	5.5623	21.8387
Error (%)	0.14	0.27	0.34	3.94	1.05	3.50	0.49
<i>Cell loss probability</i>							
Simulation	0.0032110	0.0015410	0.0015890	0.0024270	0.0023680	0.0028300	0.010399
Analysis	0.0032206	0.0015682	0.0015851	0.0024572	0.0023396	0.0026998	0.010287
Error (%)	0.30	1.73	0.25	1.23	1.21	4.82	1.09

 $\rho_c = 0.5$.

Table 5

End-to-end delay, delay variance, and loss probability for Fig. 4(d) of mixed servers with capacity $K = 10$ using probability routing matrix of Eq. (33)

	SN	Intermediate Node (IN)				DN	End-to-end
		IN ₀₁	IN ₀₂	IN ₁₁	IN ₁₂		
<i>Delay</i>							
Simulation	1.9589	1.6374	2.2056	1.8879	2.3312	1.7614	7.85140
Analysis	1.9529	1.6385	2.2030	1.8934	2.3320	1.7478	7.92456
Error (%)	0.31	0.07	0.12	0.29	0.03	0.78	0.92
<i>Delay variance</i>							
Simulation	2.2699	0.9363	4.8571	2.0532	5.3560	1.2096	10.6912
Analysis	2.2547	0.9341	4.8383	2.0513	5.3285	1.1701	10.4320
Error (%)	0.67	0.24	0.39	0.09	0.52	3.38	2.48
<i>Cell loss probability</i>							
Simulation	0.0005270	0.0000490	0.0016130	0.0003310	0.0023220	0.0001300	0.0030625
Analysis	0.0005134	0.0000539	0.0015713	0.0003388	0.0022958	0.0001209	0.0030170
Error (%)	2.65	8.72	2.65	2.30	1.14	7.53	1.51

 $\rho_c = 0.5$.

is again another BMAP (D-BMAP) under a probabilistic routing policy with explicitly derived traffic structure.

Applying traffic splitting along with departure processes enables us to come up with a network-wise performance evaluation method in a systematic and easy manner. The proposed method is suitable for evaluating both connection-oriented networks and connectionless networks. Thus, it extends the application scope of the method proposed in Ref. [10] from a connection-oriented tree-type network to a mesh network with probabilistic routing, incorporating both finite-buffer and infinite-buffer queueing models to get performance measures including the end-to-end delay, delay variance, and cell loss probability. Through numerical experiments, we show that the network-wise performance evaluation method has good accuracy in estimating the end-to-end performance with the worst case error lower than 9.15%.

The probabilistic routing policy employed in this paper uses fixed routing probabilities because it is simple and easy to implement with minimal runtime overhead. However, it lacks flexibility to schedule packets based on the current system state which may make it fit for the system. Hence, the current work can be further extended in the direction to include the dynamic probabilistic routing policy in the future.

References

- [1] H. Adishesu, G. Varghese, G. Parulkar, An architecture for packet-stripping protocols, *ACM Trans. Comput. Syst.* 17 (4) (1999) 249–287.
- [2] D. Bertsekas, Dynamic behavior of shortest path routing algorithms for communication networks, *IEEE Trans. Autom. Contr.* AC-27 (1) (1982) 60–74.
- [3] C. Blondia, O. Casals, Performance analysis of statistical multiplexing of VBR sources, *Proc. IEEE INFOCOM'92* (1992) 828–838.
- [4] C. Blondia, A discrete time batch Markovian arrival process as B-ISDN traffic model, *Belg. J. Op. Res. Stat. Comput. Sci.* 32 (3/4) (1993) 3–23.
- [5] C. Breudan, S. Traw, J. Smith, Striping within the network subsystem, *IEEE Netw.* 9 (14) (1995) 22–32.
- [6] Z. Cao, Z. Wang, E. Zegura, Performance of hashing-based schemes for Internet load balancing, *Proc. IEEE INFOCOM'00* (2000) 332–341.
- [7] C.C. Chao, Link performance analysis in networks with blocking, Master Thesis, Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, 2002.
- [8] H.W. Ferng, J.F. Chang, Connection-wise end-to-end delay analysis in ATM networks, *IEICE Trans. Commun.* E83-B (3) (2000) 659–671.
- [9] H.W. Ferng, J.F. Chang, The departure process of discrete-time queueing systems with Markovian type inputs, *Queue. Syst.* 36 (2000) 201–220.
- [10] H.W. Ferng, J.F. Chang, Connection-wise end-to-end performance analysis of queueing networks with MMPP inputs, *Perform. Eval.* 43 (2001) 39–62.
- [11] H.W. Ferng, J.F. Chang, Departure processes of BMAP/G/1 queues, *Queue. Syst.* 39 (2001) 109–135.
- [12] H.W. Ferng, C.C. Peng, Traffic splitting and its application to network-wise performance analysis, *Proc. SCS SPECTS'03* (2003) 494–501.
- [13] W. Fischer, K.S. Meier-Hellstern, The Markov-modulated Poisson process (MMPP) cookbook, *Perform. Eval.* 18 (1993) 149–171.
- [14] J.J. Hasenbein, Stability of fluid networks with proportional routing, *Queue. Syst.* 38 (2001) 327–354.
- [15] A. Heindl, Decomposition of tandem queueing networks with MMPP input, *Perform. Eval.* 44 (2001) 5–23.
- [16] A. Heindl, Decomposition of general queueing networks with MMPP inputs and customer losses, *Perform. Eval.* 51 (2003) 117–136.
- [17] C. Herrmann, The complete analysis of the discrete time finite DBMAP/G/1/N queue, *Perform. Eval.* 43 (2001) 95–121.
- [18] K. Kang, C. Kim, Performance analysis of statistical multiplexing of heterogeneous discrete-time Markov arrival processes in ATM network, *Comput. Commun.* 20 (1997) 970–978.
- [19] A. Kel, C. Lin, M. Loh, Modeling IP traffic using the batch Markovian arrival process, *Perform. Eval.* 54 (2003) 149–173.
- [20] L. Kleinrock, *Queueing Systems: Theory*, vol. I, Wiley, New York, 1975.
- [21] J. van der Lande, Inverse multiplexing over ATM: broadband access for less, *Comput. Technol. Rev.* 18 (12) (1998) 24–26.
- [22] J.H. Lee, S.C. Liew, Q.L. Ding, String mode—a new concept for performance improvement of ATM networks, *IEEE J. Sel. Areas Commun.* 9 (9) (1991) 1452–1460.
- [23] T.T. Lee, S.C. Liew, Q.L. Ding, Parallel communications for ATM network control and management, *Perform. Eval.* 30 (4) (1997) 243–264.
- [24] K.C. Leu, V.O.K. Li, Generalized load sharing for packet-switching networks, *Proc. IEEE Netw. Protoc.* (2000) 305–314.
- [25] M. Livny, M. Melman, Load balancing in homogeneous broadcast distributed systems, *ACM Perform. Eval. Rev.* 11 (1) (1982) 47–55.
- [26] D.M. Lucantoni, New results on the single server queue with a batch Markovian arrival process, *Commun. Stat. Stochast. Models* 7 (1) (1991) 1–46.
- [27] S.C. Rhea, J. Kubiatowicz, Probabilistic location and routing, *Proc. IEEE INFOCOM'02* (2002) 1248–1257.
- [28] M.P. Saltouros, A.K. Taskaris, I.S. Venieris, A new route selection approach using scaling techniques: an application to hierarchical QoS-based routing, *Proc. IEEE LCN'00* (2000) 8–10.
- [29] S. Traw, J. Smith, Striping within the network subsystem, *IEEE Netw.* 9 (4) (1995) 22–32.
- [30] S. Wang, J.A. Silvester, A discrete-time performance model for integrated service ATM multiplexers, *Proc. IEEE GLOBECOM'93* (1993) 757–761.
- [31] W. Winston, Optimality of the shortest-line discipline, *SIAM J. Appl. Prob.* 14 (1977) 181–189.