Traffic Splitting and Its Application to Network-Wise Performance Analysis

Huei-Wen Ferng and Cheng-Ching Peng

Department of Computer Science and Information Engineering,
National Taiwan University of Science and Technology, Taipei 106, Taiwan
E-mail: hwferng@mail.ntust.edu.tw; D9015001@mail.ntust.edu.tw

Keywords: Traffic splitting, Probabilistic routing, MMPP, Network-wise performance, Performance evaluation

Abstract

Traffic in a network may split into several streams when passing through an intermediate node, e.g., router or switch. Then one question may arise: what do the resultant traffic streams become? Trying to answer such question, we characterize the split traffic routed by a probabilistic routing algorithm when the input traffic is modelled by a Markov-modulated Poisson process (MMPP) in this paper. We find out that each split traffic under such splitting mechanism is also an MMPP with a proper parameter modification which is investigated in this paper. After having characterized the split traffic, we apply the result to enable us to get network-wise performance measures, i.e., end-to-end sojourn delay and delay variance in networks employing probabilistic routing algorithms via an approximate approach. With the evidence of numerical experiments, we show that the accuracy for those performance measures remains in an acceptable region. Therefore, such network-wise performance tool may have widespread applications in networks, such as the Internet and asynchronous transfer mode (ATM) network.

1 Introduction

In the past, Poisson processes are frequently employed to model voice traffic in telephony networks. However, such type of model is no more suitable to characterize the multimedia traffic, such as voice, video, and data in current and future networks, e.g., the Internet and ATM network. Therefore, more sophisticated stochastic models should be utilized to model traffic in current and future networks. In the literature, MMPP [4] and Markov-modulated Bernoulli process (MMBP) [15], [16] are some of popular models used to capture the characteristics of burstiness and correlation of traffic in current and future networks (see [5], [6], [17].). Of course, for other purposes, one may use other different models, such as fluid model [7] as traffic models. For the sake of tractability and simplicity, we shall focus on the results with input traffic streams of MMPP in this paper.

Using such flexible input traffic model, we shall investigate the split traffic model generated by a probabilistic routing algorithm in this paper (we shall call the issue as *traffic splitting* hereafter.). Such routing algorithm can be employed to efficiently utilize network resources due to that multiple paths/routes between a specific source-destination pair avoid overloading a particular path/route determined by a fixed routing algorithm, such as Dijkstra algorithm [10]. That is to say, during a certain time period, some sub-nets may fall into congestion, while other sub-nets are light loaded for a fixed routing algorithm. In particular, such overloading situation becomes more severe for heavy traffic load and bursty traffic. Therefore, the probabilistic routing can achieve such goal to allow data traffic to be dispersed on multiple paths at the same time to reduce the fluctuation of traffic distribution, i.e., to balance traffic load in a network. Moreover, the probabilistic routing is quite simple to implement. For the above reasons, we shall only pay attention to the probabilistic routing algorithm in this paper although some adaptive routing schemes have been proposed, e.g., those in [1], [9], [11], [12]. In the literature, there are several papers have discussed traffic splitting or traffic merging. For example, in [16], the split streams of a two-state MMBP under the probabilistic routing algorithm are approximated by other two-state MMBPs with modified parameters. In [19], Tran and Do proposed an analytical approach for an IP multi-path routing scenario using the MMCPP/GE/N/∞ queueing model (here MMCPP is the Markov-modulated composed Poisson process.). Among the past related papers, [16] is the closest one to our current paper. However, unlike [16], we characterize the split traffic streams of an MMPP with an arbitrary state number in an exact form rather than in an approximate form.

In the literature, performance evaluation issue has been migrated from node-level to network-level since network operation is indeed based on network-level quality-of-service (QoS). For example, Naser and Leon-Garcia [14] investigated the network-level performance using a simulation approach. In [13], Modiano and Wieselthier analyzed the sojourn delay at each node in a tree-type network using the simple discrete-time M/D/1 queueing model. Compared to [13], authors in [2], [3], [6], [16] employed more sophisticated traffic models, such as MMPP and MMBP to capture the characteristics of multimedia traffic rather than a Poisson model. Most of the above-mentioned papers except [16] just studied the tree-type network without considering the routing mechanism. Instead, we analyze the network-wise performance for networks incorporating probabilistic routing algorithms like [16] but we consider heterogeneous MMPP input streams inbound to general servers in this paper. Based on the results of traffic splitting and the results in [2] and [3], we propose a method of network-wise performance analysis for networks with probabilistic routing algorithms. The method serves as an extension to the method given in [2], which can be regarded as the special case when all routing probabilities are set to 1. As for the queueing models addressed here, they are more general in the following aspects. (i) The input traffic models are heterogeneous MMPPs. (ii) General service time distributions are applicable. (iii) Mesh topology with probabilistic routing is assumed. Hence, it provides a broader range of application than the results given in previous papers, e.g., [2], [3], [6], [16], [18]. In addition, the probabilistic routing in some sense can emulate the working mechanism of connectionless technology, such as Internet protocol (IP). It also suitable to model the connection-oriented network, such as ATM network using the special case when all routing probabilities are set to 1. Therefore, the method of performance evaluation provided in this paper can find many potential applications in TCP/IP and ATM networks.

The rest of the paper is organized as follows. In Section 2, we first describe the traffic model, i.e., the MMPP. Then, we derive results pertinent to traffic splitting under a probabilistic routing algorithm in Section 3. Section 4 briefly depicts the approximate systematic method evaluating the end-to-end performance for networks with probabilistic routing algorithms. As for Section 5, we arrange several numerical experiments to illustrate that the approximate method in Section 4 performs well. Finally, Section 6 concludes the paper and outlines possible future work.

2 Traffic Model

The input traffic model used in this paper is the MMPP, which has been thoroughly reviewed in [4], to capture the burstiness and correlation of the input traffic. For the completeness of the paper, we now briefly depict the MMPP and its notation. An m-state MMPP can be described by two $m \times m$ matrices: an infinitesimal generator matrix $\boldsymbol{Q}$ with negative diagonal entries and non-negative off-diagonal entries and a rate matrix $\boldsymbol{\Lambda}$ which is a non-negative diagonal matrix. Moreover, the matrix $\boldsymbol{Q}$ governs the m-state underlying Markov chain and the ith diagonal entry of rate matrix $\boldsymbol{\Lambda}$ represents a Poisson arrival rate when the underlying Markov chain is in state i.

Now define the *conditional counting density function* as follows.

$$[\boldsymbol{P}(n,t)]_{ij} = Pr\{N(t)=n, J(t)=j \mid N(0)=0, J(0)=i\}, \qquad (1)$$

where $N(t)$ is the random variable which stands for the number of arrivals during the time interval $(0,t]$ and $J(t)$ is the state of the underlying Markov chain at time t, respectively. It has been shown in [4] that the matrix generating function $\boldsymbol{P}^*(z,t)$ of $\boldsymbol{P}(n,t)$, i.e., $\boldsymbol{P}^*(z,t) = \sum_{n=0}^{\infty} \boldsymbol{P}(n,t) z^n$ has the following form

$$\boldsymbol{P}^*(z,t) = e^{[\boldsymbol{Q}-(1-z)\boldsymbol{\Lambda}]t}, \text{ for } \mid z \mid \leq 1,\ t>0. \qquad (2)$$

3 Traffic Splitting under a Probabilistic Routing Policy

Let us start to address the issue of traffic splitting under the probabilistic routing strategy which uses the probabilistic manner to determine a route between the current node and the down-stream nodes for a packet. Shown in Figs. 1(a) and 1(b) are a scenario of two routing paths and a scenario of multiple routing paths. As we mentioned in the previous section, the arrival process of packets is modelled by an MMPP using parameters $\boldsymbol{Q}$ and $\boldsymbol{\Lambda}$. In the following theorem, we shall derive the traffic descriptor on each routing path if the original input traffic is an MMPP.

Theorem 1 Consider an m-state MMPP with traffic descriptor $(\boldsymbol{Q}, \boldsymbol{\Lambda})$ as an input stream of packets. Now route the packet stream modelled by the MMPP according to a random path selection on two routing paths RP_1 and RP_2 with probabilities p_1 and p_2 $(p_1 + p_2 = 1)$, respectively (See Fig. 1(a)), then the traffic streams on both routing paths are also MMPPs and they can be described by traffic descriptors $(\boldsymbol{Q}, p_1\boldsymbol{\Lambda})$ and $(\boldsymbol{Q}, p_2\boldsymbol{\Lambda})$, respectively.

Proof. Let's first define $N_{RP_1}(t)$ and $J_{RP_1}(t)$ 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 of the packet stream at time instant t, respectively, on routing path RP_1. Similar to the definition of (1), we can define the *conditional counting density function* on routing path RP_1, i.e., $[\boldsymbol{P}_{RP_1}(n_1,t)]_{ij}$ as $Pr\{N_{RP_1}(t) = n_1, J_{RP_1}(t) = j | N_{RP_1}(0) = 0, J_{RP_1}(0) = i\}$. Then we can relate $[\boldsymbol{P}_{RP_1}(n_1,t)]_{ij}$ to $[\boldsymbol{P}(n,t)]_{ij}$ (for the incoming path) as follows:

$$\begin{aligned} [\boldsymbol{P}_{RP_1}(n_1,t)]_{ij} &= \sum_{n=n_1}^{\infty} [\boldsymbol{P}(n,t)]_{ij} \frac{n!}{n_1!(n-n_1)!} \\ &\quad \times p_1^{n_1}(1-p_1)^{n-n_1}. \end{aligned} \qquad (3)$$

In the above equation, the binomial distribution is applied due to the random path selection. We note that the above entry-wise relations lead to the following matrix form:

$$\boldsymbol{P}_{RP_1}(n_1,t) = \sum_{n=n_1}^{\infty} \boldsymbol{P}(n,t)\frac{n!}{n_1!(n-n_1)!} \times p_1^{n_1}(1-p_1)^{n-n_1}. \quad (4)$$

Applying the matrix z-transform to $\boldsymbol{P}_{RP_1}(n_1,t)$ enables us to write the following equation which will have a compact expression as demonstrated in the following.

$$\begin{aligned}
\boldsymbol{P}^*_{RP_1}(z,t) &= \sum_{n_1=0}^{\infty} \boldsymbol{P}_{RP_1}(n_1,t)z^{n_1} && (5)\\
&= \sum_{n_1=0}^{\infty}[\sum_{n=n_1}^{\infty} \boldsymbol{P}(n,t)\frac{n!}{n_1!(n-n_1)!} \\
&\quad p_1^{n_1}(1-p_1)^{n-n_1}]z^{n_1} && (6)\\
&= \sum_{n=0}^{\infty}[\sum_{n_1=0}^{n} \boldsymbol{P}(n,t)\frac{n!}{n_1!(n-n_1)!} \\
&\quad (p_1 z)^{n_1}(1-p_1)^{n-n_1}] && (7)\\
&= \sum_{n=0}^{\infty} \boldsymbol{P}(n,t)[1-p_1(1-z)]^n && (8)\\
&= \boldsymbol{P}^*(1-p_1(1-z),t) && (9)\\
&= e^{[\boldsymbol{Q}-(1-z)p_1\boldsymbol{\Lambda}]t}, && (10)
\end{aligned}$$

where (5) comes from the definition of the matrix z-transform; the skill of change of variables is employed in (7) when rewriting (6); the sum of binomial form in (7), i.e., $\sum_{n_1=0}^{n}\frac{n!}{n_1!(n-n_1)!}(p_1z)^{n_1}(1-p_1)^{n-n_1}$, leads to (8) after a suitable rearrangement. Finally, we obtain the z-transform of the *conditional counting density function* $\boldsymbol{P}_{RP_1}(n_1,t)$ in (10) by applying the definition of matrix z-transform of $\boldsymbol{P}(n,t)$ in (8) with z substituting by $1-p_1(1-z)$ and rewriting (9) using (2). Thus,

$$\boldsymbol{P}^*_{RP_1}(z,t) = e^{[\boldsymbol{Q}-(1-z)p_1\boldsymbol{\Lambda}]t}. \quad (11)$$

Referring to (2), we notice that (11) is the matrix z-transform of the *counting density function* of an MMPP with traffic descriptor $(\boldsymbol{Q}, p_1\boldsymbol{\Lambda})$. Therefore, the traffic stream on routing path RP_1 is an MMPP with traffic descriptor $(\boldsymbol{Q}, p_1\boldsymbol{\Lambda})$. Using an analogous manner, we can deduce the remaining part of the theorem. Hence, the proof of the theorem is completed. □

The above result can be easily extended to multiple routing paths (more than 2 paths) in the following corollary.

Corollary 1 Consider an m-state MMPP with traffic descriptor $(\boldsymbol{Q},\boldsymbol{\Lambda})$ as an input stream of packets and route the packet stream modelled by the MMPP according to a random path selection on multiple routing paths RP_i $(1 \le i \le r)$ with probabilities p_i $(\sum_{i=1}^{r} p_i = 1)$ (See Fig. 1(b)), then on the ith routing path, the traffic stream can be described by an MMPP with traffic descriptor $(\boldsymbol{Q}, p_i\boldsymbol{\Lambda})$.

Proof. We first choose arbitrarily a routing path, say the ith routing path RP_i. The routing probability of this path is p_i. Then, treat the other routing paths as one virtual routing path with routing probability $(1-p_i)$. From the result of Theorem 1, we reach that the traffic stream on the ith routing path can be described by an MMPP with traffic descriptor $(\boldsymbol{Q}, p_i\boldsymbol{\Lambda})$. This completes the proof of the corollary. □

In [4], the property that aggregation/merge of independent heterogeneous MMPPs results in an MMPP has been illustrated, while we investigate the property that split of an MMPP according to an probability manner also produces an MMPP for each path. With both of these two properties, one may get a simple method when dealing with network-wise performance analysis. In the later section, we shall demonstrate this by suitable examples.

4 Network-Wise Performance Analysis for Networks with Probabilistic Routing

In this section, we shall apply the results of traffic splitting to network-wise performance analysis in networks which employ probabilistic routing algorithms.

4.1 Calculation of End-to-End Performance Measures

Since the probabilistic routing algorithm is taken into consideration, packets may traverse networks through different paths between the source node (SN) and the destination node (DN). For example, there are four available/possible paths from SN to DN in Fig. 2(a), i.e., path 1: SN $\to I_{(0,1)} \to I_{(0,2)} \to$ DN; path 2: SN $\to I_{(0,1)} \to I_{(1,2)} \to$ DN; path 3: SN $\to I_{(1,1)} \to I_{(0,2)} \to$ DN; path 4: SN $\to I_{(1,1)} \to I_{(1,2)} \to$ DN. Without loss of generality, 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. 2(c), i.e., SN $\to I_{(0,1)} \to I_{(1,2)} \to$DN and there are five nodes along the path shown in Fig. 2(d). 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 the routing path RP_i, and the random variable of sojourn delay at the jth 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 link delay $E[T_{EE}^{RP_i}]$ and mean nodal delay $E[T_j^{RP_i}]$ as follows:

$$\begin{aligned}
E[T_{EE}] &= \sum_{i=1}^{r} E[T_{EE}^{RP_i}]p_i \\
&= \sum_{i=1}^{r}(\sum_{j=1}^{n_i} E[T_j^{RP_i}])p_i.
\end{aligned} \quad (12)$$

The first equality of (12) 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, we first note that:

$$\begin{aligned} Var[T_{EE}] &= \sum_{i=1}^{r} Var[T_{EE}^{RP_i}]p_i \\ &= \sum_{i=1}^{r}[\sum_{j=1}^{n_i} Var[T_j^{RP_i}] \\ &\quad +2\sum_{j=1}^{n_i-1}\sum_{k=j+1}^{n_i} \\ &\quad Cov(T_j^{RP_i}, T_k^{RP_i})]p_i. \end{aligned} \tag{13}$$

Although (13) 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's difficult to get such information in a real network. Therefore, we neglect the correlation of two sojourn delays at intermediate nodes in the following calculation. This results in the following approximation.

$$Var[T_{EE}] \approx \sum_{i=1}^{r}(\sum_{j=1}^{n_i} Var[T_j^{RP_i}])p_i. \tag{14}$$

4.2 The Recursive Scheme

Combining the results given in the previous subsection and the results given in [2] (or [3]), we now describe a recursive scheme which provides a systematic approach to get the network-wise performance in networks with probabilistic routing algorithms. As mentioned earlier, networks with mesh topology, such as those in Figs. 2(a) and 2(b) and heterogeneous traffic of MMPPs will be considered in this paper. Since the superposition of independent MMPPs yields another MMPP, we can category the traffic streams inbound to a node into two types, i.e., the tagged MMPP (denoted by $\mathrm{MMPP_t}$) and the aggregated MMPP (denoted by $\mathrm{MMPP_c}$) of other interfering/cross MMPPs. A simplified queueing model of mesh network is shown in Fig. 3. Now, the network-wise performance analysis scheme is described immediately in the following.

Step 1: Initialization phase: set parameters of traffic models and servers as well as the routing probability at each node.

Step 2: Assume that there are r available paths from SN to DN. Choosing a path, say RP_i, from these available paths, we can then calculate the link performance through step 4 to step 11.

Step 3: ***If*** the link performance of all possible routing paths has been completed, go directly to step 12.

Step 4: For convenience, we index the source node and the destination node with label $(0,0)$ and $(0,n+1)$, respectively. As for the other intermediate nodes, they are labelled by (i,j), where the intermediate nodes are configured into $m \times n$ mesh network ($1 \le i \le m$, $1 \le j \le n$).

Step 5: Set the node index $NI = (u,v) = (0,0)$.

Step 6: Perform the nodal performance analysis for node NI using the results given in [4] to get the tagged performance measure.

Step 7: ***If*** $NI = (0,n+1)$ (the destination node), go to step 11.

Step 8: First apply the decomposition scheme to collect the tagged departure statistics before splitting/routing using the method given in [2], and then characterize the split traffic using the results of traffic splitting in the previous section.

Step 9: Use the moment matching scheme to emulate the output process as a two-state MMPP using the method given in [2].

Step 10: Set the node index $NI = (u,v)$ with the u related to the next node on the chosen path and $v \leftarrow v+1$, then go to step 6.

Step 11: Gather nodal performance measures from SN to DN for the chosen path and output its link performance measure, then go to step 2.

Step 12: First multiply all obtained link performance measures, i.e., performance measures of paths by the corresponding path routing probabilities, then sum all the product terms to get the end-to-end performance from SN to DN, see (12) and (14).

5 Numerical Experiment and Verification

We arrange experiments to study the performance of the approximate approach proposed in the previous section for evaluating the network-wise performance. Before showing these experiment results, we first describe the experiment arrangement and the system model employed in these experiments.

5.1 Experiment Arrangement and System Model

For convenience, we first introduce some shorthands to be used in the following, that is, D: deterministic, Exp.: exponential, and E_k: k-stage Erlangian (see [8]). These notations are used to denote different service time distributions, for example, see Fig. 2 for different service arrangements. As for softwares employed in calculation and simulation

Matlab ver. 5.3 is applied to evaluate the analytical results and Microsoft Visual C++ is used to run simulation programs. All these softwares are executed on an IBM compatible PC.

We arrange two simple mesh networks shown in Figs. 2(a) and 2(b) in the experiment. For mesh network in Fig. 2(a), the service types are arranged as follows: E_4 at SN, Exp. at $I_{(0,1)}$, D at $I_{(0,2)}$, E_4 at $I_{(1,1)}$, E_4 at $I_{(1,2)}$, and E_4 at DN, while all E_4 servers are assumed for each node in mesh network of Fig. 2(b) for simplicity. Packets in both of networks are routed according to probabilistic routing. For example, routing probability matrices described in (15)–(17) are applicable for the network shown in Fig. 2(a), while routing probability matrix given in (18) is used for the network shown in Fig. 2(b). We note that the routing probability matrix $\boldsymbol{M}_{rprob_1}$ given in (15) routes almost all packets to path SN$\rightarrow I_{(0,1)} \rightarrow I_{(0,2)} \rightarrow$ DN. Hence, it is a connection-oriented-like mechanism; in (16), $\boldsymbol{M}_{rprob_2}$ represents a symmetric routing with respect to the sub-net topology; and $\boldsymbol{M}_{rprob_3}$ in (17) is a random routing. For each server, an infinite buffer is endowed and its mean service time is set to 1. As for input traffic models, we use heterogeneous two-state MMPPs which are parameterized by a four-tuple traffic descriptor $(\lambda_1, \lambda_2, \sigma_1, \sigma_2)$ for simplicity. 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 (0.06, 0.02, 0.01, 0.015) and aggregated cross traffic at each node according to Table 5 (for network in Fig. 2(a)) and Table 6 (for network in Fig. 2(b)). For parameters in both Tables 5 and 6, there are four different parameter sets with cross traffic load increasing as the set number increases.

$$\boldsymbol{M}_{rprob_1} = \begin{array}{c c} & \begin{array}{cccccc} SN & I_{(0,1)} & I_{(0,2)} & I_{(1,1)} & I_{(1,2)} & DN \end{array} \\ \begin{array}{c} SN \\ I_{(0,1)} \\ I_{(0,2)} \\ I_{(1,1)} \\ I_{(1,2)} \\ DN \end{array} & \begin{bmatrix} 0.00 & 0.95 & 0.00 & 0.05 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.95 & 0.00 & 0.05 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \\ 0.00 & 0.00 & 0.95 & 0.00 & 0.05 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{bmatrix} \end{array} \tag{15}$$

$$\boldsymbol{M}_{rprob_2} = \begin{array}{c c} & \begin{array}{cccccc} SN & I_{(0,1)} & I_{(0,2)} & I_{(1,1)} & I_{(1,2)} & DN \end{array} \\ \begin{array}{c} SN \\ I_{(0,1)} \\ I_{(0,2)} \\ I_{(1,1)} \\ I_{(1,2)} \\ DN \end{array} & \begin{bmatrix} 0.00 & 0.50 & 0.00 & 0.50 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.50 & 0.00 & 0.50 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \\ 0.00 & 0.00 & 0.50 & 0.00 & 0.50 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{bmatrix} \end{array} \tag{16}$$

$$\boldsymbol{M}_{rprob_3} = \begin{array}{c c} & \begin{array}{cccccc} SN & I_{(0,1)} & I_{(0,2)} & I_{(1,1)} & I_{(1,2)} & DN \end{array} \\ \begin{array}{c} SN \\ I_{(0,1)} \\ I_{(0,2)} \\ I_{(1,1)} \\ I_{(1,2)} \\ DN \end{array} & \begin{bmatrix} 0.00 & 0.65 & 0.00 & 0.35 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.75 & 0.00 & 0.25 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \\ 0.00 & 0.00 & 0.15 & 0.00 & 0.85 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{bmatrix} \end{array} \tag{17}$$

$$\boldsymbol{M}_{rprob_4} = \begin{array}{c c} & \begin{array}{cccccccc} SN & I_{(0,1)} & I_{(0,2)} & I_{(0,3)} & I_{(1,1)} & I_{(1,2)} & I_{(1,3)} & DN \end{array} \\ \begin{array}{c} SN \\ I_{(0,1)} \\ I_{(0,2)} \\ I_{(0,3)} \\ I_{(1,1)} \\ I_{(1,2)} \\ I_{(1,3)} \\ DN \end{array} & \begin{bmatrix} 0.00 & 0.65 & 0.00 & 0.00 & 0.35 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.75 & 0.00 & 0.00 & 0.25 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.82 & 0.00 & 0.00 & 0.18 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \\ 0.00 & 0.00 & 0.15 & 0.00 & 0.00 & 0.85 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.66 & 0.00 & 0.00 & 0.34 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{bmatrix} \end{array} \tag{18}$$

5.2 End-to-end Performance in a Mesh Network with Probabilistic Routing

Now, we are ready to have a look at the end-to-end performance of a mesh network under a probabilistic routing algorithm. We show performance measures of end-to-end delay and delay variance as well as nodal performance measures in Tables 1–3 for the network in Fig. 2(a) using parameters given in Table 5 under three different routing probability arrangements described by (15)–(17). The results given in Tables 1–3 suggest the following observations. (i) Effect of cross traffic load: observing the differences among results of different parameter sets within any specific table, one can easily know that both delay and delay variance increase as cross traffic load increases. (ii) Effect of traffic concentration: we notice that delay and delay variance of nodes $I_{(0,1)}$ and $I_{(0,2)}$ in Table 1 are the largest than the results corresponding to other two tables. This is mainly due to that most of traffic is routed to the path SN $\rightarrow I_{(0,1)} \rightarrow I_{(0,2)} \rightarrow$ DN for the connection-oriented-like routing given in (15). As for the end-to-end delay and delay variance, they posses the similar trends except for parameter set # 1. These suggest one to suitable balance traffic load to the entire network. For Table 4, it contains the performance measures for the network in Fig. 2(b) using parameters given in Table 6 under the routing probability matrix governed by (18). The result given in Table 4 is just used to demonstrate that our approach is still applicable for different size of a network since the worst error for delay or delay variance is lower than 9.40%. Therefore, the network-wise performance evaluation approach provided in this paper performs well in most situation with acceptable accuracy.

6 Conclusions

We have mathematically and exactly derived the split traffic under a probabilistic routing mechanism in a rigorous manner rather than an approximate approach as proposed in [16]. It turns out that the split traffic on each routing path from an original MMPP input is again another MMPP. The characterization on the split traffic is done via relating the parameters of the original MMPP and routing probabilities to those of the resultant split traffic. Based on the results of traffic splitting, we then propose a network-wise performance evaluation method to extend the scope of network topology used in [2] from a connection-oriented tree-type network to a mesh network with probabilistic routing. Through numerical experiments, we show that the network-wise performance evaluation method has good accuracy in estimating end-to-end delay and delay variance of a network because the observed worst case error in these experiments is at most 9.4%.

The current work can be further extended in the following directions which are currently under investigation: (i) to incorporate the finite-buffer queueing model to figure out the packet/cell loss behavior in a network; (ii) to enlarge the scope of traffic model from MMPP to batch Markovian arrival process (BMAP) and the discrete-time version of BMAP, i.e., D-BMAP etc.

References

[1] S. Bahk and M. E. Zarki. 1994. "Preventive Congestion Control Based Routing in ATM Networks." In Proceedings of IEEE ICC '94, (May 1–5). 1592–1599.

[2] H. W. Ferng and J. F. Chang, 2001, "Connection-wise End-to-end Performance Analysis of Queueing Networks with MMPP Inputs.", Performance Evaluation 43, 39–62.

[3] H. W. Ferng and J. F. Chang, 2000, "Connection-wise End-to-end Delay Analysis in ATM Networks.", IEICE Trans. Commun. E83–B, no.3, 659–671.

[4] W. Fischer and K. S. Meier-Hellstern, 1993, "The Markov-modulated Poisson Process (MMPP) Cookbook.", Performance Evoluation 18, 149–171.

[5] H. Heffes and D. M. Lucatoni, 1986, "A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance.", IEEE J. Select. Areas Commun. SAC–4, no.6, 856–868.

[6] A. Heindl, 2001, "Decomposition of General Tandem Queueing Networks with MMPP Input.", Performance Evaluation 44, 5–23.

[7] O. Kella, 2001, "Markov-Modulated Feedforward Fluid Networks.", Queueing Systems 37, 141–161.

[8] L. Kleinrock. 1975. *Queueing Systems: Theory, Vol. I.* Wiley, New York.

[9] R. Krishnan and J. A. Silvester. 1993. "An Approach to Path-splitting in Multipath Networks." In Proceedings of IEEE ICC '93, (May 23–26). 1353–1357.

[10] J. F. Kurose and K. W. Ross. 2002. *Computer Network – A Top-Down Approach Featuring the Internet.* 2nd edition, Addison Wesley, New York.

[11] K. C. Leung and V. O. K. Li. 1999. "Flow Assignment and Packet Scheduling for Multipath Networks." In Proceedings of IEEE GLOBECOM '99, 246–250.

[12] K. C. Leung and V. O. K. Li. 2000. "Generalized Load Sharing for Packet-switching Networks." In Proceedings of IEEE Network Protocol '00, 2000, 305–314.

[13] E. Modiano and J. E. Wieselthier, 1996, "A Simple Analysis of the Average Queueing Delay in Tree Networks.", IEEE Transactions on Information Theory, 42, no.2, 660–664.

[14] H. Naser and A. Leon-Garcia. 1996. "A Simulation Study of Delay and Delay Variation in ATM Networks." In Proceedings of IEEE INFOCOM '96, (March 24–28). 393–400.

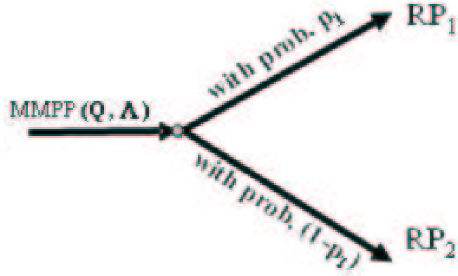
[15] D. Park, H. G. Perros, and H. Yamashita, 1994, "Approximate Analysis of Discrete-time Tandem Queueing Networks with Bursty and Correlated Input Traffic and Customer Loss.", Oper. Res. Lett. 15, 95–104.

[16] N. K. Singh, S. K. Bose, and Y. N. Singh. 1999. "An Approach for Performance Analysis of Discrete-time Finite Capacity Open Queueing Networks with Correlated Arrivals." In Proceedings of. ICICS '99, (Dec. 7-10). Singapore.

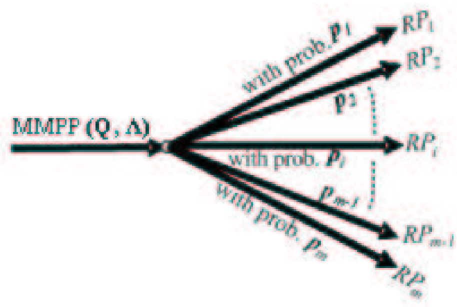
[17] P. Skelly, M. Schwartz, and S. Dixit, 1993, "A Histogram-based Model for Video Traffic Behaviour in An ATM Multiplexer.", IEEE/ACM Transaction on Networking 1, no.4, 446–459.

[18] W. Stallings. 2002. *High-Speed Networks and Internets – Performance and Quality of Service.* 2nd edition. Prentice Hall, New Jersey.

[19] H. T. Tran and T. V. Do. 2002. "An Analytical Model for Analysis of Multipath Routing Scheme." In Proceedings of the 2002 Interational Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS '02 (July 19–22).

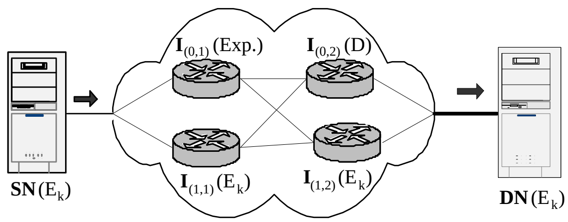


(a) Two routing paths.

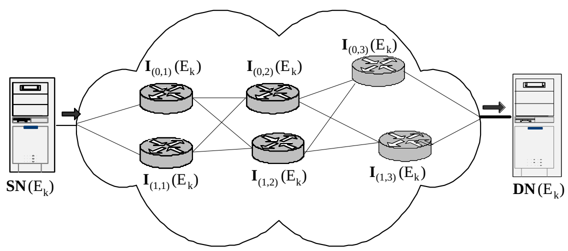


(b) m (≥ 2) routing paths.

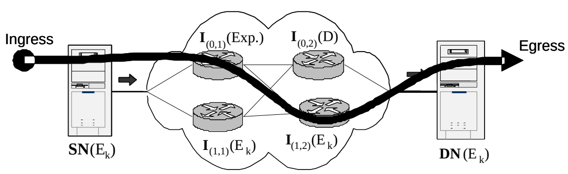
Figure 1: Scenarios of traffic splitting.



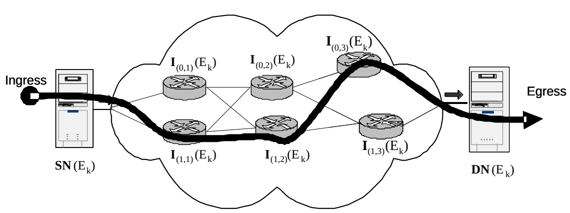
(a) A network with 2×2 mesh sub-net.



(b) A network with 2×3 mesh sub-net.



(c) A routing path in the 2×2 mesh sub-net.



(d) A routing path in the 2×3 mesh sub-net.

Figure 2: Network configurations and routing paths.

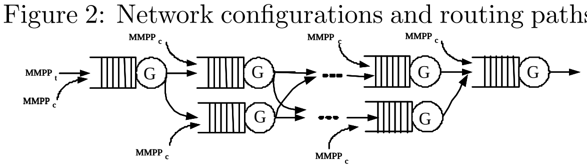


Figure 3: Mesh network configuration of MMPP/G/1 queues.

Table 1: End-to-end delay and delay variance accompanied with information on each intermediate node for the network in Fig. 2(a) governed by a connection-oriented-like routing using (15).

Para. Set		SN E_4	$I_{(0,1)}$ Exp.	$I_{(0,2)}$ D	$I_{(1,1)}$ E_4	$I_{(1,2)}$ E_4	DN E_4	End-to-end
	Mean delay							
para.#1	Simulation	1.2313	1.4898	1.2870	1.5148	1.5966	1.4069	5.4318
	Analysis	1.2315	1.4996	1.2900	1.5189	1.5931	1.4193	5.4622
	Error	0.02%	0.65%	0.23%	0.27%	0.22%	0.87%	0.56%
para.#2	Simulation	1.6137	2.1710	1.6662	1.8729	1.9650	1.9241	7.3787
	Analysis	1.6132	2.1786	1.6702	1.8677	1.9673	1.9309	7.3915
	Error	0.03%	0.35%	0.24%	0.28%	0.12%	0.35%	0.17%
para.#3	Simulation	2.5947	3.5897	2.4394	2.3875	2.4584	2.8412	11.4060
	Analysis	2.5966	3.6120	2.4414	2.3943	2.4453	2.8464	11.4110
	Error	0.07%	0.62%	0.08%	0.28%	0.54%	0.18%	0.04%
para.#4	Simulation	3.0958	4.4210	2.7776	2.7005	2.7388	3.2205	13.2552
	Analysis	3.0958	4.2575	2.7731	2.7088	2.7333	3.2131	13.2274
	Error	0.00%	0.39%	0.16%	0.31%	0.20%	0.23%	0.21%
	Delay variance							
para.#1	Simulation	0.5441	2.5449	0.2830	1.0531	1.2118	0.8457	3.9915
	Analysis	0.5434	2.2702	0.2861	1.0556	1.1735	0.8595	3.9423
	Error	0.13%	0.67%	1.08%	0.24%	3.26%	1.61%	1.23%
para.#2	Simulation	1.3197	4.9081	0.9737	2.0138	2.2957	2.1827	9.4649
	Analysis	1.3150	4.9108	0.9779	1.9826	2.2381	2.1704	8.9289
	Error	0.36%	0.05%	0.43%	1.57%	2.57%	0.57%	5.66%
para.#3	Simulation	4.8225	13.6631	3.4546	3.9308	4.1097	6.0802	27.0428
	Analysis	4.8613	13.7903	3.4597	3.9728	4.1448	6.0480	25.0383
	Error	0.80%	1.14%	0.15%	1.06%	0.85%	0.53%	7.41%
para.#4	Simulation	7.3628	18.9538	4.9389	5.2880	5.5785	8.2337	38.2751
	Analysis	7.4056	19.0920	4.8969	5.3961	5.4787	8.0632	34.9318
	Error	0.58%	0.72%	0.86%	2.00%	1.82%	2.11%	8.73%

Table 2: End-to-end delay and delay variance accompanied with information on each intermediate node for the network in Fig. 2(a) governed by a symmetric routing using (16).

Para. Set		SN E_4	$I_{(0,1)}$ Exp.	$I_{(0,2)}$ D	$I_{(1,1)}$ E_4	$I_{(1,2)}$ E_4	DN E_4	End-to-end
	Mean delay							
para.#1	Simulation	1.2314	1.4426	1.2587	1.5667	1.6495	1.4108	5.6010
	Analysis	1.2315	1.4486	1.2600	1.5705	1.6465	1.4115	5.5901
	Error	0.01%	0.41%	0.10%	0.24%	0.18%	0.05%	0.19%
para.#2	Simulation	1.6113	2.0701	1.6083	1.9579	2.0578	1.9269	7.3854
	Analysis	1.6132	2.0751	1.6070	1.9552	2.0584	1.9157	7.3628
	Error	0.12%	0.24%	0.08%	0.14%	0.03%	0.58%	0.31%
para.#3	Simulation	2.5966	3.3363	2.2920	2.5587	2.6030	2.8536	10.8459
	Analysis	2.5966	3.3533	2.2856	2.5497	2.5958	2.8178	10.8071
	Error	0.00%	0.51%	0.28%	0.35%	0.28%	1.27%	0.36%
para.#4	Simulation	3.0977	3.8796	2.5721	2.9229	2.9294	3.2287	12.4791
	Analysis	3.0958	3.9106	2.5689	2.9134	2.9268	3.1794	12.4363
	Error	0.06%	0.79%	0.12%	0.33%	0.09%	1.55%	0.34%
	Delay variance							
para.#1	Simulation	0.5445	2.0896	0.2451	1.1924	1.3444	0.8524	3.9282
	Analysis	0.5434	2.1173	0.2484	1.1659	1.3373	0.8444	3.9214
	Error	0.20%	1.31%	1.33%	2.27%	3.53%	0.95%	0.17%
para.#2	Simulation	1.3116	4.3984	0.8494	2.3073	2.5765	2.1746	8.7247
	Analysis	1.3150	4.4546	0.8440	2.2447	2.5721	2.1242	8.7593
	Error	0.26%	1.26%	0.64%	2.79%	0.17%	2.37%	0.40%
para.#3	Simulation	4.8916	11.7771	2.8990	4.7706	4.8580	6.1454	23.6841
	Analysis	4.8613	11.9178	2.8712	4.6400	4.437	5.9072	23.9176
	Error	0.62%	1.18%	0.97%	2.81%	0.30%	4.03%	0.99%
para.#4	Simulation	7.4042	15.9059	3.9976	6.4980	6.5092	8.2160	32.8713
	Analysis	7.4056	16.1734	3.9964	6.4164	6.4831	7.8750	33.1783
	Error	0.02%	1.65%	0.03%	1.27%	0.40%	4.33%	0.94%

Table 3: End-to-end delay and delay variance accompanied with information on each intermediate node for the network in Fig. 2(a) governed by a random routing using (17).

Para. Set		SN E_4	$I_{(0,1)}$ Exp.	$I_{(0,2)}$ D	$I_{(1,1)}$ E_4	$I_{(1,2)}$ E_4	DN E_4	End-to-end
	Mean delay							
para.#1	Simulation	1.2321	1.4587	1.2622	1.5514	1.6400	1.4112	5.5704
	Analysis	1.2315	1.4653	1.2639	1.5529	1.6419	1.4131	5.4973
	Error	0.05%	0.45%	0.13%	0.10%	0.12%	0.13%	1.31%
para.#3	Simulation	1.6137	2.0957	1.6161	1.9220	2.0550	1.9301	7.3968
	Analysis	1.6132	2.1086	1.6151	1.9250	2.0505	1.9191	7.3167
	Error	0.03%	0.61%	0.06%	0.16%	0.22%	0.57%	1.08%
para.#3	Simulation	2.5891	3.4190	2.2987	2.4913	2.5875	2.8587	10.9739
	Analysis	2.5966	3.4361	2.3045	2.4958	2.5826	2.8247	10.9615
	Error	0.29%	0.50%	0.25%	0.18%	0.19%	1.20%	0.11%
para.#4	Simulation	8.0255	10.3284	6.1501	6.5392	6.9978	9.1619	12.6599
	Analysis	8.0357	10.3885	6.1570	6.5876	7.0096	9.0026	12.7349
	Error	0.13%	0.58%	0.11%	0.73%	0.17%	1.77%	0.59%
	Delay variance							
para.#1	Simulation	0.5441	2.1392	0.2529	1.1309	1.3217	0.8477	4.0253
	Analysis	0.5434	2.1666	0.2533	1.1275	1.3266	0.8476	3.9409
	Error	0.13%	1.26%	0.16%	0.30%	0.37%	0.01%	2.09%
para.#2	Simulation	1.3135	4.5407	0.8642	2.1682	2.5724	2.1920	9.0406
	Analysis	1.3150	4.6000	0.8607	2.1525	2.5461	2.1346	8.8290
	Error	0.11%	1.29%	0.41%	0.73%	1.03%	2.69%	2.34%
para.#3	Simulation	4.8248	12.3103	2.9050	4.4072	4.8066	6.1820	24.8049
	Analysis	4.8613	12.5036	2.9391	4.4029	4.7831	5.9428	24.2598
	Error	0.75%	1.55%	1.61%	0.10%	0.49%	3.87%	2.20%
para.#4	Simulation	7.4353	16.7457	4.1577	6.1097	6.4968	8.1581	34.5703
	Analysis	7.4056	17.0779	4.0970	6.0514	6.3926	7.9242	33.7014
	Error	0.40%	1.95%	1.48%	0.96%	1.63%	2.87%	2.51%

Table 4: End-to-end delay and delay variance accompanied with information on each intermediate node for the network in Fig. 2(b) governed by the routing using (18).

Para. Set		SN E_4	$I_{(0,1)}$ E_4	$I_{(0,2)}$ E_4	$I_{(0,3)}$ E_4	$I_{(1,1)}$ E_4	$I_{(1,2)}$ E_4	$I_{(1,3)}$ E_4	DN E_4	End-to-end
	Mean delay									
para.#1	Sim	1.2312	1.2839	1.3230	1.3675	1.5486	1.6425	1.7442	1.4723	7.1287
	Ana	1.2315	1.2919	1.3293	1.3931	1.5529	1.6419	1.7031	1.4716	7.0043
	Error	0.02%	0.62%	0.47%	1.84%	0.28%	0.04%	2.41%	0.05%	1.78%
para.#2	Simulation	1.6116	1.6958	1.7567	1.8262	1.9238	2.0493	2.2044	2.0323	9.3480
	Analysis	1.6132	1.7040	1.7625	1.8758	1.9250	2.0505	2.1246	2.0210	8.9215
	Error	0.11%	0.48%	0.33%	2.64%	0.06%	0.06%	3.76%	0.56%	4.78%
para.#3	Simulation	2.5954	2.5753	2.5883	2.6200	2.4960	2.5878	2.6820	2.9299	13.3156
	Analysis	2.5966	2.5831	2.5955	2.7223	2.4958	2.5826	2.5650	2.8896	12.6324
	Error	0.05%	0.30%	0.28%	3.76%	0.01%	0.20%	4.56%	1.39%	5.41%
para.#4	Simulation	3.0913	2.9694	2.9375	2.9231	2.8410	2.9232	2.9902	3.2563	15.1635
	Analysis	3.0958	2.9740	2.9418	3.0559	2.8421	2.9096	2.8408	3.2118	14.3089
	Error	0.15%	0.15%	0.15%	4.35%	0.04%	0.47%	5.26%	1.39%	5.97%
	Delay variance									
para.#1	Simulation	0.5447	0.6253	0.6903	0.7682	1.1217	1.3300	1.5990	0.9665	4.5741
	Analysis	0.5434	0.6369	0.6986	0.8106	1.1275	1.3266	1.4723	0.9597	4.2776
	Error	0.24%	1.82%	1.19%	5.23%	0.51%	0.26%	8.61%	0.71%	6.48%
para.#2	Simulation	1.3098	1.5167	1.6837	1.8713	2.1672	2.5556	3.0983	2.5042	10.5496
	Analysis	1.3150	1.5372	1.6894	2.0057	2.1525	2.5460	2.8514	2.4490	9.8202
	Error	0.40%	1.33%	0.34%	6.70%	0.68%	0.38%	8.66%	2.25%	6.91%
para.#3	Simulation	4.8684	4.7752	4.8337	4.9855	4.3842	4.8406	5.2623	6.5066	26.4101
	Analysis	4.8613	4.7952	4.8464	5.4361	4.4029	4.7831	4.8354	6.2626	25.8828
	Error	0.15%	0.42%	0.26%	8.29%	0.42%	1.20%	8.83%	3.90%	2.00%
para.#4	Simulation	7.3354	6.7248	6.5577	6.5017	6.0550	6.4592	6.4592	8.3458	36.0944
	Analysis	7.4056	6.7415	6.5659	7.1764	6.0514	6.3925	6.0311	8.0511	35.3288
	Error	0.95%	0.25%	0.12%	9.40%	0.06%	1.04%	7.10%	3.66%	2.12%

Table 5: The parameters of cross MMPP traffic at each node for Fig. 2(a).

Para. Set	SN	$I_{(0,1)}$ / $I_{(1,1)}$	$I_{(0,2)}$ / $I_{(1,2)}$	DN
para.#1	(0.2667, 0.1333, 0.1130, 0.1695)	(0.3333, 0.2000, 0.1160, 0.1740) (0.5000, 0.3667, 0.1185, 0.1778)	(0.3667,0.2333, 0.1190, 0.1785) (0.5333, 0.4000, 0.1215, 0.1823)	(0.4000,0.2667, 0.1220, 0.1830)
para.#2	(0.5333, 0.2667, 0.1330, 0.1995)	(0.5867, 0.3200, 0.1360, 0.2040) (0.6700, 0.4033, 0.1385, 0.2078)	(0.6133,0.3467, 0.1390, 0.2085) (0.6967, 0.4300, 0.1415, 0.2123)	(0.6400,0.3733, 0.1420, 0.2130)
para.#3	(0.8000, 0.4000, 0.1530, 0.2295)	(0.8167, 0.4167, 0.1560, 0.2340) (0.8200, 0.4200, 0.1585, 0.2378)	(0.8250,0.4250, 0.1590, 0.2385) (0.8283, 0.4283, 0.1615, 0.2423)	(0.8333,0.4333, 0.1620, 0.2430)
para.#4	(0.8667, 0.4333, 0.1730, 0.2595)	(0.8733, 0.4400, 0.1760, 0.2640) (0.8750, 0.4417, 0.1785, 0.1685)	(0.8767,0.4433, 0.1790, 0.2685) (0.8783, 0.4450, 0.1815, 0.2723)	(0.8800,0.4467, 0.1820, 0.2730)

Table 6: The parameters of cross MMPP traffic at each node for Fig. 2(b).

Para. set	SN	$I_{(0,1)}$ / $I_{(1,1)}$	$I_{(0,2)}$ / $I_{(1,2)}$	N(0,3) / N(1,3)	DN
para.#1	(0.2667, 0.1333, 0.1130, 0.1695)	(0.3333, 0.2000, 0.1160, 0.1740) (0.5000, 0.3667, 0.1185, 0.1778)	(0.3667, 0.2333, 0.1190, 0.1785) (0.5333, 0.4000, 0.1215, 0.1823)	(0.4000, 0.2667, 0.1220, 0.1830) (0.5667, 0.4300, 0.1245, 0.1868)	(0.4333,0.3000, 0.1250, 0.1875)
para.#2	(0.5333, 0.2667, 0.1330, 0.1995)	(0.5867, 0.3200, 0.1360, 0.2040) (0.6700, 0.4033, 0.1385, 0.2078)	(0.6133, 0.3467, 0.1390, 0.2085) (0.6967, 0.4300, 0.1415, 0.2123)	(0.6400, 0.3733, 0.1420, 0.2130) (0.7233, 0.4567, 0.1445, 0.2168)	(0.6667,0.4000, 0.1450, 0.2175)
para.#3	(0.8000, 0.4000, 0.1530, 0.2295)	(0.8167, 0.4167, 0.1560, 0.2340) (0.8200, 0.4200, 0.1585, 0.2378)	(0.8250, 0.4250, 0.1590, 0.2385) (0.8283, 0.4283, 0.1615, 0.2423)	(0.8333, 0.4333, 0.1620, 0.2430) (0.8367, 0.4367, 0.1645, 0.2468)	(0.8417,0.4417, 0.1650, 0.2475)
para.#4	(0.8667, 0.4333, 0.1730, 0.2595)	(0.8733, 0.4400, 0.1760, 0.2640) (0.8750, 0.4417, 0.1785, 0.1685)	(0.8767,0.4433, 0.1790, 0.2685) (0.8783, 0.4450,0.1815, 0.2723)	(0.8800, 0.4467, 0.1820, 0.2730) (0.8817, 0.4483, 0.1845, 0.2768)	(0.8833,0.4500, 0.1850, 0.2775)