Exact Solution Using Matrix Geometric Techniques For a Per-Flow Queueing Node With Threshold Routing

Essia H. Elhafsi
Dept. of Computer Science and Engineering
University of California
Riverside, CA. 92521
essia@cs.ucr.edu

Mart Molle
Dept. of Computer Science and Engineering
University of California
Riverside, CA. 92521
mart@cs.ucr.edu

***Abstract*— We consider a router that uses a shared input queue and distributes traffic over multiple output ports. Though such an architecture is subject to the Head-of-the-Line (HOL) blocking problem, it is very attractive when the router supports both load balancing over alternate paths and QoS. Therefore, in this work, we consider per-flow (or destination) queueing which (1) eliminates the need to further consider per-flow queueing at the individual destination ports therefore avoids the HOL blocking problem and (2) achieves QoS.**

We model a single traffic class at a routing node as it passes through a single-shared-input and then assigned to two-output ports that lead to alternate paths to the same destination. We assume that the remainder of the network transit delay beyond the router are different for both paths. Therefore our routing policy reduces the end-to-end delay by favoring the faster path by assigning packets to the higher-delay path only if the input queue length exceeds a given threshold b. We generalize the model by permitting the output port that leads to the slower path to represent a multichannel link group (e.g., multiple parallel wavelengths on the same optical fiber). We model our system as a two-dimensional Markov chain and use matrix geometric techniques to solve for its stationary probabilities.

Keywords: **Markov chain; per-flow queueing; forking node; load balancing; alternate paths.**

I. INTRODUCTION

Alternate path routing can provide load balancing [17], [2], [13] and route failure protection [16] by distributing traffic among a set of diverse paths. These benefits make alternate path routing an ideal candidate for the bandwidth limited and mobile ad-hoc networks [21], [1]. Therefore, we consider a router that distributes traffic over two available paths to the same destination.

Routing in optical networks with WDM links [23], [18], [12] is another application domain where dynamic alternate routing thus load balancing may provide a significant performance improvement. In this case, each physical cable attached to a router's output port can carry many independent data streams simultaneously over different wavelengths. More importantly, the physical layer may support optical cross-connection of specific wavelengths between adjacent cables, to create a direct, all-optical, multi-hop *Virtual Link* between two physically non-adjacent routers. The Virtual Link can be established in one of two ways: (1) either the same wavelengths are available on every link in the physical path, or (2) all (or some) of the intermediate nodes in the path have wavelength converters.

Consider node A in Figure 1(a) and assume that all the traffic from A is routed to node D. In this example all AD traffic follows the 3-hop physical path (l_1, l_2, l_3) through intermediate nodes B and C. However, we might have the choice between two alternative paths at the logical level, as shown in Figure 1(c). Path 1 uses Virtual Link v_4 to reach D in a single "router-hop", after we have configured the optical interconnects appropriately at intermediate nodes B and C (Figure 1(b)). Path 2 requires three "router-hops" to reach D through the logical path $(v_1,\ v_2,\ v_3)$. Since path 1 avoids the store-and-forward delay at intermediate routers B and C, the remainder of network transit delay — between a packet's departure from A and its arrival at D — will be much shorter if we route the packet along path 1 than path 2.

At the logical level, we refer to node A as a forking node since it consists of a single input feeding two outputs. We assume that arrivals join a shared queue and are routed to one of two input ports when they reach the head of the queue.

Generally, router architecture do not use shared queues feeding multiple output ports because of the Head-of-Line (HOL) blocking problem. However, in this work, we assume that the router supports QoS and load balancing. Moreover, we consider a single class traffic thus there is no need to consider per-flow queueing at the individual output ports and eliminates the HOL blocking problem.

We adopt a routing policy that can reduce the end-to-end delay for traffic going from A to D that is easy to implement. Random routing is the simplest routing policy to implement and analyze. It is known to have tractable analytical model. However, our findings [4] show that the use of this policy in asymmetric networks may not provide satisfactory results.

In [8], we introduced the forking node scheduling problem with paths that have different transit delays. We also compared the performance of a variety of random, deterministic and state-dependent routing algorithms in this application, using

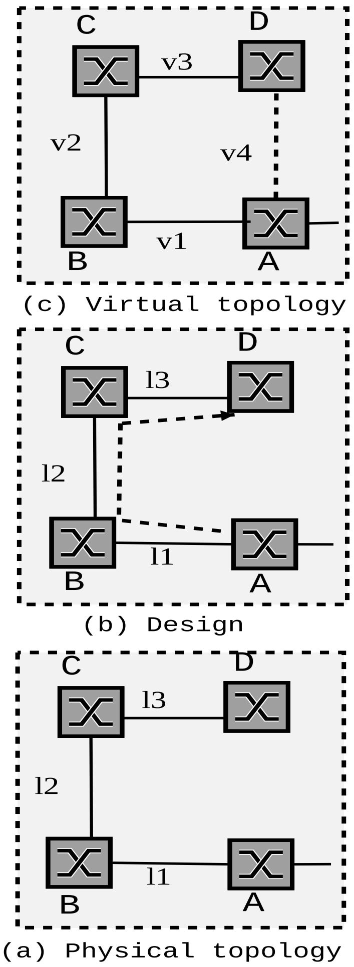

Fig. 1. The physical, design and virtual topologies. Virtual channel v_4 uses optical links $l_1,\ l_2$ and l_3. There is a single hop path $p_1 = (v_4)$ between A and D, as well as a multihop path $p_2 = (v_1,\ v_2,\ v_3)$

simulations. From all the algorithms we tested, a simple generalization of the well-known *Join-the-Shortest-Queue (JSQ)* consistently gave us the best performance according to a variety of metrics. The generalized JSQ is simply to add a bias b to favor the path with the lower downstream delay. Thus, it sends the packet on the slower path only when the queue length is larger than a given threshold b. Thus the routing policy is referred to as $JSQ + b$.

In this work we provide a further generalization of the *JSQ+b* routing policy by permitting one of the output ports to represent a multichannel link group (e.g., multiple parallel wavelengths on the same optical fiber). For the purpose of our analysis, we assume that the faster path is a Virtual Link representing a single channel/wavelength. In this way, we retain the flexibility to carry traffic over shorter distances on the remaining channels. The routing policy then has a choice of sending the packet on the slower path (or path 1) that is served by a group of $m_1 \geq 1$ parallel servers that we call link group 1, or on the faster path (or path 2) that is served by a single server link ($m_2 = 1$) or link group 2. Our objective is to model the forking node and obtain an exact solution to its stationary probabilities without resorting to the "brute force" approach of modeling the node as a two-dimensional Markov chain and computing its stationary probabilities by numerically solving its global balance equations (i., e. solving $\pi P = \pi$). The simple case of single-server link groups has been solved analytically [11]. However, the results obtained are complex expressions, so by extrapolation, the corresponding results for the case of multiserver link groups would be even more complicated, if obtained at all.

Another approach is to solve for bounds and/or approximations. In [7], we derived bounds for the stationary probabilities for the multiserver link group problem. In this paper, however, we show how to exploit the distinctive structure of the state space to find exact solutions to the stationary probabilities using the well-known matrix geometric method. We model the forking node as a QBD process with finite state space when the queue length is less than b. On the other hand, when the queue length exceeds b the chain is represented as a simple birth-death process with death rate equal to $(m_1 + 1)\mu$.

The rest of this paper is organized as follows. In Section II, we present our model and the state space. In Section III, we present an exact solution to the single server link groups ($m_1 = m_2 = 1$). In Section IV, we provide a general and a simplified solution to the multiserver link group 1 forking node using matrix geometric techniques. In Section V, we compare the methods computational complexity. In Section VI, we compute several performance metrics of the proposed method. Finally, we conclude in Section VII.

II. Model and State Space

We consider a node where per-flow arrivals join a single infinite buffer queue. The analysis can be easily extended to a finite buffer queue by truncating the state space at some specified maximum queue size. We assume that arrivals are Poisson and the queue is serviced by the $JSQ + b$ routing policy for a specified bias b. We further assume that the links run at the same speed of μ.

Let Q_t denote the system state at time t with $Q_t := [i, j]$, where $i \in \{0, 1, 2, \ldots, m_1\}$ denotes the number of packets currently being transmitted on link group 1 and $j \in \{0, 1, 2, \ldots\}$ denotes the number of other packets currently in the system. $\nu_2(j)$ of these remaining packets are currently being transmitted on link group 2, where we have defined $\nu_\ell(x) \equiv \min\{x, m_\ell\}$, and $j - \nu_2(j)$ are waiting in the queue. $\{Q_t\}_{t \geq 0}$ can be represented as a continuous time Markov chain (CTMC). Let $\mathcal{S}$ denote the state space of this chain. The CTMC for the system and the transition rates are shown in Figure 2. Clearly, for $\rho := \frac{\lambda}{(m_1+1)\mu} < 1$, the Markov chain is ergodic.

III. Single Server Link groups

The case of single server link groups ($m_1 = 1, m_2 = 1$) has been studied by Lin and Kumar [11] and exact solutions have been provided. For completeness, we provide a summary of their results. Using our notation, let $\pi(i, j)$, $[i, j] \in \mathcal{S}$, denote the equilibrium distribution of $\{Q_t\}$. $\pi(i, j)$ can be expressed in terms of $\pi_1 := \pi(1, 0)$ as follows:

$$\pi(i,j) = \begin{cases} \pi_1 \left(c_1 \eta_1^j + c_2 \eta_2^j \right) & 1 \le j \le b+1,\ i=1 \\ \pi(1,b+1)\ \rho^{j-b-1} & j \ge b+2,\ \ i=1 \\ \pi_1 \left(\frac{da^j}{a} - c_1 \eta_1^j - c_2 \eta_2^j \right) & 1 \le j \le b+1,\ i=0 \\ \pi_1 \left(\frac{d}{a} - 1 \right) & (i,j) = (0,0) \end{cases}$$

where $\eta_1, \eta_2, c_1, c_2, d$ and a are defined as follows

$$\begin{array}{ll} \eta_1 = \frac{1-\sqrt{1-4\mu\lambda}}{2\mu} & \eta_2 = \frac{1+\sqrt{1-4\mu\lambda}}{2\mu} \\ c_1 = \frac{1-\eta_1}{\eta_2-\eta_1} & c_2 = \frac{\eta_2-1}{\eta_2-\eta_1} \\ a = \frac{\lambda}{\mu} & d = \frac{c_1 \eta_2^{b+2} + c_2 \eta_2^{b+2}}{a^{b+1}} \end{array}$$

$$\pi_1^{-1} = \begin{cases} \text{for } \lambda \neq \mu \\ \qquad \sum_{i=1}^{2} c_i \eta_i^{b+1} \left(\frac{\eta_i a^{-b-2}}{(1-a)} + \frac{\rho}{1-\rho} - \frac{\eta_i}{1-a} \right) \\ \text{for } \lambda = \mu \\ \qquad \sum_{i=1}^{2} c_i \eta_i^{b+1} \left(2\eta_i + \frac{\rho}{1-\rho} \right) \end{cases}$$

The expressions for $\pi(i,j)$ are quite complex thus, a closed-form expression for the mean number in the system and other performance metrics using the above solution seems elusive. Therefore, numerical solution or tight bounds with simpler expressions may provide better insight and allow easier computation. In [7] we derive tight lower and upper bounds for the stationary probabilities for the more general case of multi-server link groups ($m_1 \ge 1$ and $m_2 \ge 1$) and consequently obtain bounds for the number in the system, the departure rate from the slower path and the mean average delay. In the following analysis, we are interested in simple exact solutions for the stationary probabilities.

IV. Multi-Server Link groups

We model the forking node as a two-dimensional Markov chain when the queue size is less than the routing threshold b and as a simple birth-death process when the queue length exceeds b.

The CTMC of the system (see Figure 2) shows that the chain can be partitioned into a lower subsystem $\underline{\mathcal{S}} \equiv \{(i,j)\ :\ 0 \le i \le m_1,\ 0 \le j \le b+1\}$ and a higher subsystem $\overline{\mathcal{S}} \equiv \{(i,j)\ :\ i = m_1,\ j > b+1\}$. We model both subsystems independently and solve for their stationary probabilities. Then the global solution is obtained by combining both solutions using a normalizing constant.

1. *Solution to the Markov chain defined over $\underline{\mathcal{S}}$*

In the following analysis, we explore the structure of the system defined over $\underline{\mathcal{S}}$ to use matrix geometric techniques and obtain a closed-form expression for the stationary probabilities. We first present a brief summary of the technique that we use in our methodology.

1) Matrix geometric solution to infinite state-space QBD processes: Key to using matrix geometric techniques is for the model to have a repetitive structure which leads to Markov models that fit within the matrix geometric framework. In this work, we are interested in QBD processes [19], [15], [14].

A QBD process is a Markov process with an infinitesimal generator matrix of the following form:

$$Q = \begin{bmatrix} \widehat{L} & \widehat{F} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & B & L & F & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & B & L & F & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & B & L & F & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & B & L & F & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

The block entries $B,\ L,\ F$ and $\widehat{L}$ are square sub-matrices, which satisfy the equilibrium conditions $\widehat{L}e + \widehat{F}e = \widehat{B}e + Le + Fe = (B + L + F)e = 0$; e is a column vector of ones of suitable dimension.

The state space is generally partitioned into a block of boundary states S_0 and the remaining blocks of states $S_\xi, \xi \ge 0$ that represent the repetitive portion of the Markov chain. We use the letters "L", "F" and "B" to describe "local", "forward" and "backward" transition rates respectively with relation to a block of states $S_\xi, \xi \ge 1$ and "$\widehat{\ }$" for matrices related to states in S_0.

QBD processes have interesting structural properties which can be used to simplify the computation of the stationary probabilities. Two matrices usually denoted by R and G play a major role in the general theory. These matrices have important probabilistic interpretations. An entry (k,j) in G expresses the conditional probability of the process first entering $S_{\xi-1}$ through state j given that it starts from state i of S_ξ [20]. An entry (k,j) in R is the expected time spent in state j of S_ξ before the first visit to $S_{\xi-1}$ given the starting state k in $S_{\xi-1}$ [14].

Both matrices R and G are minimal nonnegative solutions of two nonlinear matrix equations (i.e., $B + LG + FG^2 = 0$ and $F + RL + R^2B = 0$). Moreover for a QBD process, R can be expressed as $R = -F(L + FG)^{-1}$. Although QBD processes can be solved by either computing R or G, the first one is usually used.

Let $\underline{\pi} = (\underline{\pi}_0, \underline{\pi}_1, ...)$ be the stationary probability vector of a QBD process with infinite state space where $\underline{\pi}_\xi$ is the sub-vector of stationary probabilities for states in block ξ. Under general assumptions, the elements of this vector have the following matrix geometric property [14],

$$\underline{\pi}_{\xi+1} = \underline{\pi}_\xi \cdot R, \quad \xi \ge 0 \qquad (1)$$

2) Matrix geometric solution to finite state-space QBD processes: $\underline{\mathcal{S}}$ can be viewed as finite state space Markov chain that can be modeled as a QBD process. For finite state-space QBD processes however, the situation is quite different from the infinite state space due to the presence of additional boundary states. Thus, it is not possible to guarantee in general that the stationary distribution has a matrix geometric structure of the form of Equation (1). Several methods have

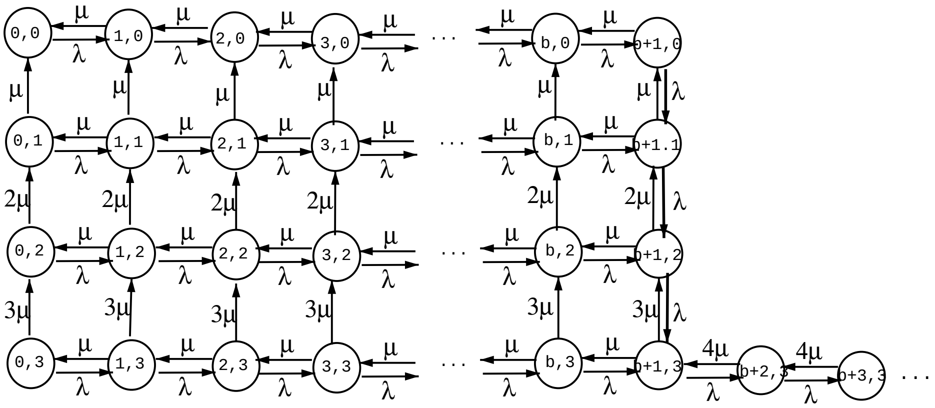

Fig. 2. The CTMC for the single shared queue model with threshold routing when $m_1 = 3,\ m_2 = 1$ and $b > 0$.

been proposed in the literature to solve for the stationary probabilities of QBD processes with a finite state space [10], [9], [3], [22]. In the following we present a simple method that has a lower computational complexity than the one provided in the literature [5].

The state space $\underline{S}$ can be further divided into blocks ξ where $0 \leq \xi \leq b+1$ of size m_1+1. Each block is composed of states $(i, \xi), 0 \leq i \leq m_1+1$ for a given ξ. Coordinate ξ denotes the 'level' and i the 'phase' of state (i, ξ). In this case we have $b+1$ levels.

The generator matrix Q of this finite state space process has the following block-tridiagonal structure:

$$Q = \begin{bmatrix} \widehat{L} & \widehat{F} & 0 & 0 & 0 & 0 & 0 & \cdots \\ \widehat{B} & L & F & 0 & 0 & 0 & 0 & \cdots \\ 0 & B & L & F & 0 & 0 & 0 & \cdots \\ 0 & 0 & B & L & F & 0 & 0 & \cdots \\ 0 & 0 & 0 & B & L & F & 0 & \cdots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & L & F_b \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & B_{b+1} & L_{b+1} \end{bmatrix} \tag{2}$$

The upper boundary block entries $\widehat{L}$, $\widehat{F}$ and $\widehat{B}$ and the non-boundary block entries $B,\ L$ and F and the lower boundary block entries $B_{b+1},\ F_b$ and L_{b+1} are square matrices of dimensions $(m_1+1) \times (m_1+1)$.

If the underlying Markov chain with generator matrix Q is irreducible, then the matrices $L,\ \widehat{L}$ and L_{b+1} along the diagonal can be shown to be nonsingular [14].

- *Global balance Equations.* The stationary probability vector $\underline{\pi}$ for Q is generally partitioned as $\underline{\pi} = [\underline{\pi}_0,\ \underline{\pi}_1,\ \underline{\pi}_2, ..., \underline{\pi}_{b+1}]$, where the sub-vectors $\underline{\pi}_\xi, (0 \leq \xi \leq b+1)$ are of dimension m_1+1. Solving $\underline{\pi}Q = 0$ along with the normalizing equation $\underline{\pi}e = 1$, yields the following set of equations in matrix form:

$$\underline{\pi}_0\widehat{L} + \underline{\pi}_1\widehat{B} = 0 \tag{3}$$

$$\underline{\pi}_0\widehat{F} + \underline{\pi}_1 L + \underline{\pi}_2 B = 0 \tag{4}$$

$$\underline{\pi}_{\xi-1}F + \underline{\pi}_\xi L + \underline{\pi}_{\xi+1}B = 0 \quad 2 \leq \xi < b \tag{5}$$

$$\underline{\pi}_{b-1}F + \underline{\pi}_b L + \underline{\pi}_{b+1}B_{b+1} = 0 \tag{6}$$

$$\underline{\pi}_b F_b + \underline{\pi}_{b+1}L_{b+1} = 0 \tag{7}$$

- *Computation of the rate matrices.* Here, we assume that Equation (8) holds among the stationary probability vectors $\underline{\pi}_\xi$ for states in set S_ξ, and R_ξ is a square matrix of order (m_1+1),

$$\underline{\pi}_\xi = \underline{\pi}_{\xi-1}R_\xi, \quad \xi \geq 1 \tag{8}$$

By simple algebraic manipulation of the global balance equations we obtain R_ξ's as follows.

  - From Equation (3) and assuming that $\widehat{L}$ is nonsingular, we get,

$$\underline{\pi}_0 = -\underline{\pi}_1\widehat{B}\widehat{L}^{-1} \equiv \underline{\pi}_1 R_0 \tag{9}$$

  - Equation (7) leads to the following expression for $\underline{\pi}_{b+1}$ and R_{b+1} where L_{b+1} is required to be non-singular,

$$\underline{\pi}_{b+1} = -\underline{\pi}_b\ F_b\ L_{b+1}^{-1} \equiv \pi_b\ R_{b+1}$$

  - Equation (6) leads to the following expression of $\underline{\pi}_b$ and R_b,

$$\begin{aligned} \underline{\pi}_b &= -\underline{\pi}_{b-1}F(L - FL_{b+1}^{-1}B_{b+1})^{-1} \\ &= -\underline{\pi}_{b-1}F(L + R_{b+1}B_{b+1})^{-1} \\ &\equiv \underline{\pi}_{b-1}R_b \end{aligned}$$

  - Finally, from Equation (5) we obtain a general relation between $\underline{\pi}_{\xi-1}$, $\underline{\pi}_\xi$ and R_{b+1},

$$\begin{aligned} \underline{\pi}_\xi &= -\underline{\pi}_{\xi-1}F(L + R_{\xi+1}B)^{-1} \\ &\equiv \underline{\pi}_{\xi-1}R_\xi, \qquad 2 \leq \xi < b \end{aligned}$$

  R_ξ can be computed using Algorithm 1.

In Algorithm 1, I represents the identity matrix of dimension $(m_1+1) \times (m_1+1)$. We also assume that $-\widehat{L}^{-1}$ and $-L_{b+1}^{-1}$ are nonnegative matrices. Moreover, if $(L + R_\xi B)$ is stable then it is nonsingular. Note that the rate matrices $R_\xi,\ \xi = 0, ..., b+1$ are obtained through purely algebraic manipulations starting from the global balance conditions; they have no probabilistic interpretation and therefore do not coincide in general with the rate matrix introduced by Neuts. Moreover, the rate matrices R_ξ introduced here are not always positive and this could lead to some numerical instabilities.

Algorithm 1 Compute$R(\xi)$

```
$R_{b+1} \leftarrow -FL_{b+1}^{-1}$
if $\xi > 1$ then
    for $j = b \rightarrow \xi + 1$ do
        $R_j \leftarrow -F(L + R_{j+1}B)^{-1}$
    end for
    $return\ \ R_\xi \leftarrow -F(L + R_{\xi+1}B)^{-1}$
end if
if $\xi == 0$ then
    return $R_0 \leftarrow -\widehat{B}\widehat{L}^{-1}$
end if
if $\xi == 1$ then
    return $R_1 \leftarrow I$
end if
```

It is worth noting that the solution to the rate matrices can be generalized to the infinite state space solution given in [14]. For infinitely large number of blocks ($b \to \infty$), the rate matrices R_ξ will converge to R which is the minimal nonnegative solution to

$$R = -F(L + RB)^{-1} \tag{10}$$

This condition is equivalent to the following nonlinear equation introduced in [14]

$$F + RL + R^2B = 0 \tag{11}$$

- *Stationary Probabilities*

  Theorem: for any QBD process with finite state space, having an infinitesimal generator matrix given by Equation (2), the stationary probabilities are given in matrix-geometric form by

  $$\underline{\pi}_\xi = \underline{\pi}_1 R_\xi^* \tag{12}$$

  where $R_\xi^* = \prod_{j=1}^{\xi} R_j$ and R_j is computed using Algorithm 1.

  Solving Equation (3) and (4) for $\underline{\pi}_1$ leads to,

  $$\underline{\pi}_1 \left(R_0\widehat{F} + L + R_2B \right) = 0 \tag{13}$$

  Thus, after substitution and mathematical manipulation, Equation (14) follows from the normalizing condition $\sum\limits_{\xi=0}^{b+1} \underline{\pi}_\xi \; e = 1$ and Equation (9).

  $$\underline{\pi}_1 \left(R_0 e + \sum_{\xi=1}^{b+1} R_\xi^* e \right) = 1 \tag{14}$$

  Solving the system of linear equations given by Equations (13) and (14), we solve for $\underline{\pi}_1$, and from Equation (8), we obtain $\underline{\pi}_\xi$ as,

  $$\begin{aligned} \underline{\pi}_\xi &= \underline{\pi}_{\xi-1} R_\xi \\ &= \underline{\pi}_{\xi-2} R_{\xi-1} R_\xi \\ &= \underline{\pi}_{\xi-2} R_{\xi-2} R_{\xi-1} R_\xi \\ &\vdots \\ &= \underline{\pi}_1 R_1 \ldots R_{\xi-2} R_{\xi-1} R_\xi \\ &= \underline{\pi}_1 \prod_{j=1}^{\xi} R_j = \underline{\pi}_1 R_\xi^* \end{aligned} \tag{15}$$

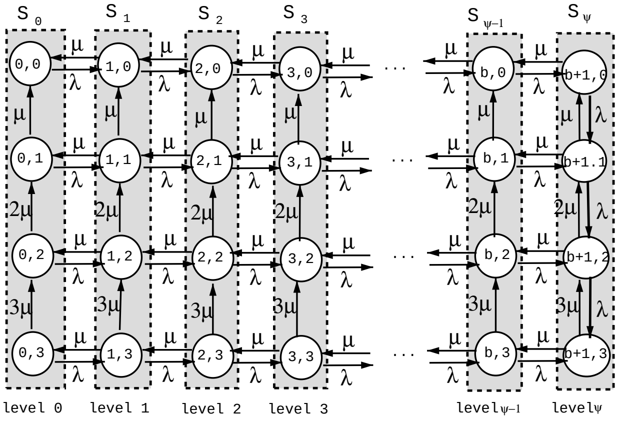

Fig. 3. CTMC of the shared queue forking node defined over $\underline{\mathcal{S}}$ when $m_1 = 3$, $m_2 = 1$ and $b > 0$.

Algorithm 2 is used to compute the stationary probabilities $\underline{\pi}$.

Algorithm 2 Compute $\underline{\pi}$

```
for $j = 1 \rightarrow \xi$ do
    $R_j \leftarrow computeR(j)$
    $\underline{\pi}_j \leftarrow \underline{\pi}_1 R_j$
end for
```

3) Simplified Solution Methodology: Now we customize the general solution procedure to a simple case problem. The CTMC for the system and the transition rates when $m_1 = 3$ and $m_2 = 1$ are shown in Figure 3 and the corresponding block matrices are given as:

$$B = \begin{bmatrix} \mu & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 \\ 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & \mu \end{bmatrix}, \qquad F = \begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda \end{bmatrix},$$

$$L = \begin{bmatrix} -(\lambda+\mu) & 0 & 0 & 0 \\ \mu & -(\lambda+2\mu) & 0 & 0 \\ 0 & \mu & -(\lambda+2\mu) & 0 \\ 0 & 0 & \mu & -(\lambda+2\mu) \end{bmatrix},$$

$$\widehat{L} = \begin{bmatrix} -\lambda & 0 & 0 & 0 \\ \mu & -(\lambda+\mu) & 0 & 0 \\ 0 & \mu & -(\lambda+\mu) & 0 \\ 0 & 0 & \mu & -(\lambda+\mu) \end{bmatrix},$$

$$\widehat{L}_{b+1} = \begin{bmatrix} -(\lambda+\mu) & \lambda & 0 & 0 \\ \mu & -(\lambda+2\mu) & \lambda & 0 \\ 0 & \mu & -(\lambda+2\mu) & \lambda \\ 0 & 0 & \mu & -2\mu \end{bmatrix},$$

$$\widehat{B} = B_{b+1} = B, \qquad \widehat{F} = F_b = F.$$

Note that matrices B, B_{b+1}, $\widehat{B}$, F, F_{b+1} and $\widehat{F}$ are diagonal and can be expressed as

$$B = B_{b+1} = \widehat{B} = \mu I$$
$$F = F_b = \widehat{F} = \lambda I.$$

This is very fortunate, since having B and F be diagonal matrices means that their inverse can be trivially computed by forming the scalar inverse of their diagonal elements. For the rest of the analysis, we assume that $m_2 = 1$ and $m_1 \geq 1$. The block matrices corresponding to the generator matrix Q are therefore, dependent on m_1. The larger m_1, the larger the matrices sizes are.

Moreover, note that B, B_{b+1}, $\widehat{B}$, F, F_{b+1} and $\widehat{F}$ are nonsingular diagonal matrices.

We solve the global balance equations given by Equations (16-18) and obtain a unique solution to the stationary probabilities $\{\underline{\pi}_\xi\}_{0 \leq \xi \leq b+1}$.

- The global balance equations are given as follows,

$$\begin{aligned} \underline{\pi}_0 \widehat{L} + \mu \underline{\pi}_1 I &= 0 && (16) \\ \lambda \underline{\pi}_{\xi-1} I + \underline{\pi}_\xi L + \mu \underline{\pi}_{\xi+1} I &= 0, \quad \xi = 1, ..., b && (17) \\ \lambda \underline{\pi}_b I + \underline{\pi}_{b+1} L_{b+1} &= 0 && (18) \end{aligned}$$

- The stationary probabilities are,

$$\begin{aligned} \underline{\pi}_1 &= & -\tfrac{1}{\mu} \underline{\pi}_0 \widehat{L} & \equiv \underline{\pi}_0 R_1 \\ \underline{\pi}_2 &= & -\tfrac{1}{\mu} \underline{\pi}_0 (\lambda I + R_1 L) & \equiv \underline{\pi}_0 R_2 \\ \underline{\pi}_3 &= & -\tfrac{1}{\mu} \underline{\pi}_0 (\lambda R_1 + R_2 L) & \equiv \underline{\pi}_0 R_3 \\ & & \vdots & \\ \underline{\pi}_b &= - & \tfrac{1}{\mu} \underline{\pi}_0 (\lambda R_{b-2} + R_{b-1} L) & \equiv \underline{\pi}_0 R_b \\ \underline{\pi}_{b+1} &= & -\tfrac{1}{\mu} \underline{\pi}_0 (\lambda R_{b-1} + R_b L) & \equiv \underline{\pi}_0 R_{b+1} \end{aligned}$$

which can be generalized to the following form:

$$\underline{\pi}_\xi = \underline{\pi}_0 R_\xi, \qquad \xi = 0, \ldots, b+1 \tag{19}$$

where R_ξ's are computed using the following procedure:

$$\begin{cases} R_0 = & I \\ R_1 = & -\frac{1}{\mu} \widehat{L} \\ R_\xi = & -\frac{1}{\mu} (\lambda R_{\xi-2} + R_{\xi-1} L), \quad \xi = 2, ..., b+1 \end{cases}$$

Note that Equation (18) can be expressed as,

$$\underline{\pi}_0 (\lambda R_b + R_{b+1} L_{b+1}) \quad = 0 \tag{20}$$

Therefore, $\underline{\pi}_0$ can be obtained by solving Equation (20) along with the following normalizing condition:

$$\underline{\pi}_0 \sum_{i=0}^{b+1} R_i e \quad = 1$$

2. *Solution to Markov chain defined over $\overline{\mathcal{S}}$*

The Markov chain defined over $\overline{\mathcal{S}}$ is a simple birth-death process with arrival rate λ and service completion rate $(1 + m_1)\mu$. Therefore, the solution to its stationary probabilities $\{\overline{\pi}_n\}_{\{n > m_1 + b + 1\}}$ can be easily computed and is given by:

$$\overline{\pi}_n = \overline{\pi}_\gamma \rho^{n-\gamma} \quad n > \gamma \tag{21}$$

where n is the number of packets in the system including the ones in service (if any) and $\gamma = m_1 + 1 + b$.

Finally, to obtain the general solution of the Markov chain defined over $\underline{\mathcal{S}}$, a normalizing constant can be easily computed to combine the solutions to the stationary probabilities given by Equations (19), or (15) and (21).

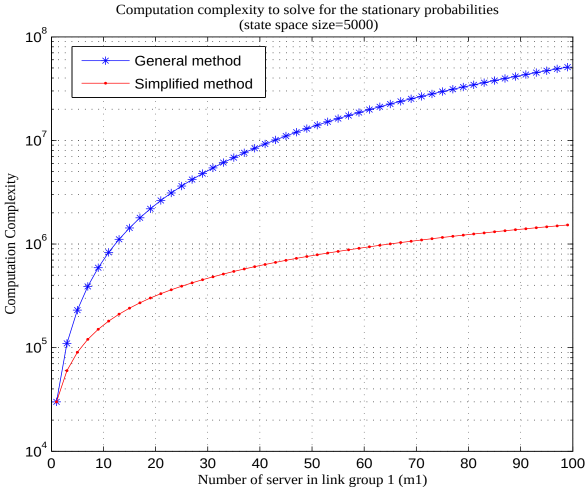

Fig. 4. Comparison of the running time to compute the stationary probabilities using the general and the simplified method for variable number of servers at link group 1 (or variable block size)

V. Comparative Analysis

The methods proposed in this paper provide an exact solution to the stationary probabilities. To verify our results, we computed the probabilities for various values of m_1 and b using our methods and using brute-force method (i. e., $\pi = \pi P$). The solutions were exactly the same.

The general method proposed in this paper assumes that $\widehat{L}$ and L_Ψ are nonsingular matrices. Moreover, if $\widehat{L}$ and L_Ψ are non-negative the system will be stable (R_ξ positive). One of the advantages of this method over several methods in the literature is that it does not need to solve any quadratic non-linear equation to obtain the stationary probabilities. The simplified method is derived from the general method when $B, \widehat{B}$, B_Ψ, $F, \widehat{F}$ and $F_{\Psi-1}$ are nonsingular matrices.

We compare the methods based on their computational complexity that we express in terms of the number of matrix operations performed by each method: number of matrix multiplications, additions and inversions.

Since the block size (or number of states in a given level) is equal to $(m_1 + 1)$, we assume that the addition and the multiplication of two $(m_1 + 1) \times (m_1 + 1)$ matrices requires respectively, $(m_1 + 1)$ and $(m_1 + 1)^2$ operations. The inversion of an $(m_1 + 1) \times (m_1 + 1)$ matrix, on the other hand, requires $(m_1 + 1)^3$ operations.

With regards to the general method proposed in this paper, to compute the stationary probabilities, we have to compute $R_\xi, \xi = 0, 1, 2, ..., b+1$ (see Algorithm 1) which requires $(b + 2)$ matrix inversions, $(4b+7)$ matrix multiplications and $(b+1)$ matrix additions. Therefore, the computation complexity of the general method can be easily computed to be $(b + 2)(m_1 + 1)^3 + (4b + 7)(m_1 + 1)^2 + (m_1 + 1)(b + 1)$. Since link group 2 is served by a single server, we expect the optimal routing bias to have a small value, therefore we can safely claim that the asymptotic complexity of the general method is $O(m_1^3)$.

Likewise, the simplified method requires $3(b + 2)$ matrix multiplications and $3(b+1)$ matrix additions. No matrix inversion is required. This leads to a total computation complexity equal to $3(b+2)(m_1+2)^2+3(b+1)(m_1+1)$ or $O(m_1^2)$. This result represents a huge computation improvement over the general method.

In Figure 4, we plot the computation complexity of the methods for various values of m_1 and for a given state space size $S=(b+2)(m_1+1)$. The general method has the worst performance. Moreover, we notice that the performance gap between both methods increases as m_1 increases.

VI. Forking node performance evaluation

The objective of this section is to use the obtained closed form expression for the stationary probabilities, and compute several performance measures at the forking node. We compute the delay at the node and its behavior as a function of the node's throughput. We are also interested in the proportion of traffic that joins the queue and uses the faster path we also study its sensitivity to the routing bias.

1. Average delay.

We assume that packets entering the forking node will have an average delay T. The delay represents the average time packets spend at the forking node (that includes the queueing, the processing and the service delays). We can compute the average number of packets at the forking node and then apply Little's theorem. Thus for a given arrival rate λ we have,

$$T=\frac{N}{\lambda} \tag{22}$$

where N is the average number of packets at the forking node including the ones in service (if any). Using first principles, we compute N which is composed of two terms *(1)* N_0, the number of packets at the lower states defined over P_0, and *(2)* N_1, the number of packets at the higher states defined over P_1. N_0 and N_1 combined using a normalizing constant lead to N. N_0 can be obtained from the following expression,

$$\begin{aligned} N_0 &= \underline{\pi}_0 n_0 + \sum_{\xi=1}^{b+1} \underline{\pi}_1 \left(\prod_{j=1}^{\xi} R_j \right) n_\xi \\ &= \underline{\pi}_1 \left((R_0 n_0 + n_1) + \sum_{\xi=2}^{\Psi} R_\xi^* n_\xi \right) \end{aligned}$$

where n_ξ is a column vector of length (m_1+1) and its i^{th} entry represents the number of packets at the forking node when its current state is at level ξ, $0 \le \xi \le b+1$ with phase i.

N_1 on the other hand can be obtained from Equation (23),

$$\begin{aligned} N_1 &= \sum_{n=m_1+b+2}^{\infty} \overline{\pi}_\gamma n \rho^{n-\gamma} \\ &= \tfrac{(m_1+b+2)-(m_1+b+1)\rho}{(1-\rho)^2} \rho \overline{\pi}_\gamma \end{aligned} \tag{23}$$

The delay-throughput for various link group 1 sizes (m_1) and a given routing bias b is presented in Figure 5. Note that as the number of servers on link group 1 increases, the average delay at the forking node decreases drastically. This, obviously represents a performance improvement at the forking node.

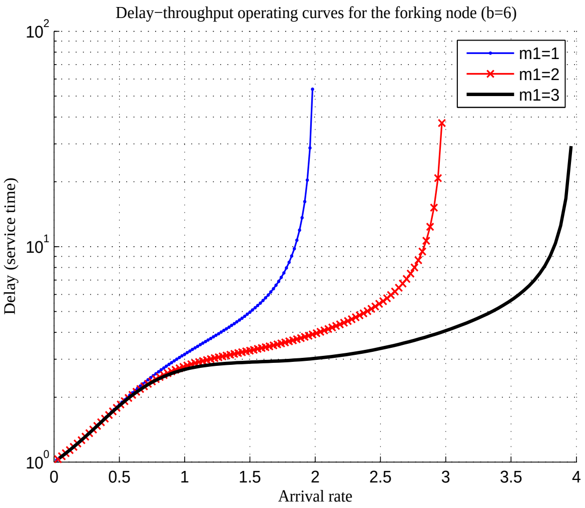

Fig. 5. Delay-throughput (in log-scale) operating curves for the forking node for different values of m_1.

However as m_1 increases, more packets are being transmitted on the slower path which means that either the queueing delay (time waiting to be served by the fast path server) is much higher than the slower path downstream delay or the routing bias has to be adjusted (increased or decreased) to account for and balance between the delay at the forking node and the slow path delay. Such a balance guarantees an overall network performance improvement [6].

2. Proportion of departures from the fast path.

Our assumption in this work is that packets that are transmitted on link group 1 (or the fast path) are subject to a lower network downstream delay. Figure 6 shows that for various number of channels on link group 1, for low network load, 100% of the traffic uses the fast path. As the network load increases, the queue builds up and exceeds b. Thus the routing policy sends more and more packets over the slower path. At high network loads, the routing policy does not seem to differentiate between the fast and the slow path, i. e., all servers are busy.

Figure 6 shows that the proportion of departure from the fast path is not sensitive to the number of servers in the slow path. Actually, as we see in the next subsection it is only sensitive to the routing bias. The proportion of departure from the fast path τ_0, computed over state space P_0 is given by Equation (24) and the one computed over state space P_1, τ_1, is given by Equation (24). τ_0 and τ_1 combined using a normalizing constant lead of the overall proportion of departure from the fast path.

$$\tau_i = \begin{cases} \frac{\mu}{\lambda} \underline{\pi}_1 \sum_{\xi=1}^{b+1} R_\xi^* e & i=0 \\ \frac{\mu}{\lambda} \sum_{n=m_1+b+2}^{\infty} \overline{\pi}_\gamma \rho^{n-\gamma} e \equiv \frac{\mu\rho}{1-\rho} \overline{\pi}_\gamma e & i=1 \end{cases} \tag{24}$$

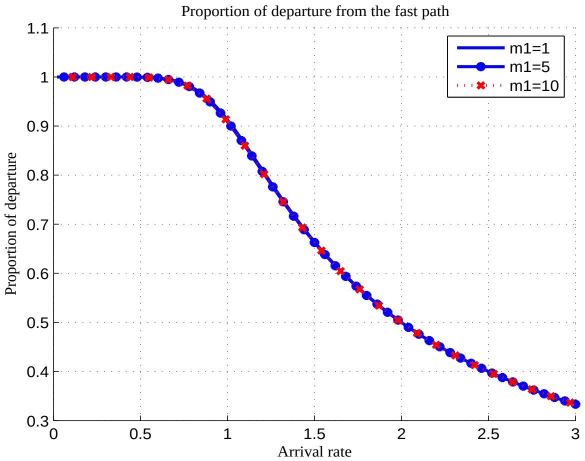

Fig. 6. Proportion of packet departure from the fast path for different values of m_1 and when $b = 6$.

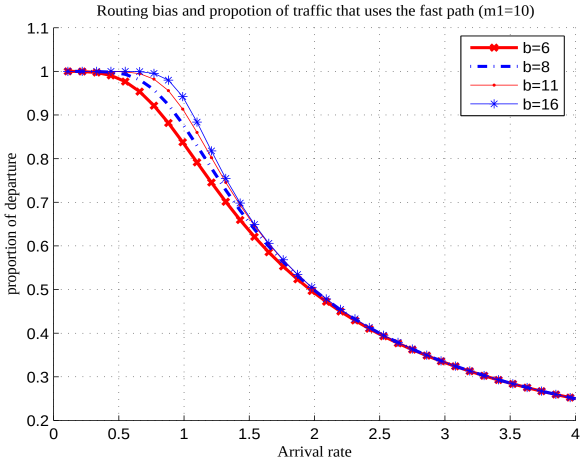

Fig. 7. Effect of the routing bias on the proportion of packet departure from the fast path for $m_1 = 10$.

3. *Effect of the routing bias on the use of the fast path.*

When link group 2 is served by a much smaller number of servers than link group 1, the routing policy can take advantage of the routing bias b to improve the forking node performance. Figure 7 shows that the proportion of traffic that uses the fast path is very sensitive to the routing bias at low network load. As b increases, more packets are allowed to wait in the queue for the fast server (as dictated by the $JSQ + b$ policy).

At high network load, an increase in b has no clear effect on the proportion of departures from the fast link. All servers are constantly busy and the queue length is always larger than the routing bias b.

VII. Conclusion

In this paper we provide a simple procedure based on matrix geometric techniques to compute exact solution to the stationary probabilities of a per-flow routing node with a single input and two outputs. Packets are transmitted on one of two link groups when they reach the head of the queue based on a threshold-type routing policy. The main contribution of this paper is that it is a generalization of the solution obtained in [11] to the single server output links to a single server link group 2 and a multiserver link group 1. We show that when simple solutions do not exit by solving the system Markov chain by brute force i. e, $\pi P = \pi$, exploring the structure of the system may lead to simpler solution procedures using matrix geometric techniques.

References

[1] Mike Burmester and Tri Van Le. Secure multipath communication in mobile ad hoc networks. *International Conference on Information Technology: Coding and Computing (ITCC'04)*, 2, 2004.

[2] Israel Cidon, Raphael Rom, and Yuval Shavitt. Analysis of multi-path routing. *IEEE/ACM transactions on Networking*, 7(6), December 1999.

[3] Vittoria de Nitto Persone and V. Grassi. Solution of finite QBD processes. *Journal of Applied Probability*, 33(4):1003–1010, December 1996.

[4] E. H. Elhafsi. *Modeling forking nodes with state dependent routing: Application to Asymmetric networks*. Ph. D. Dissertation, University of California Riverside, Riverside, CA, USA, 2005.

[5] E. H. Elhafsi and M. Molle. On the solution to QBD processes with finite state space (*submitted to the stochastic analysis and applications journal*).

[6] E. H. Elhafsi and M. Molle. Optimal routing between alternate paths with different network transit delays (*submitted to globecom*). 2006.

[7] E. H. Elhafsi, M. Molle, and D. Manjunath. Simple performance bounds for a shared-queue forking node with threshold routing and multiple links per output path (*submitted to mascots*).

[8] E. H. Elhafsi, M. Molle, and D. Manjunath. Can we use product-form solution techniques in networks with asymmetric paths? In *Proceedings of the International symposium on Performance Evaluation of computer and telecommuniaction systems (SPECTS 2005)*, pages 348–359, 2005.

[9] L. Gun and A.M. Makowski. Matrix-geometric solution for finite capacity queues with phase-type distributions. *Proceedings of Performance 87, Brussels, Belgium*, pages 269–282, December 1987.

[10] B. Hajek. Optimal control of two interacting service stations. *IEEE transactions on automatic control*, AC-29(6).

[11] W. Lin and P. R. Kumar. Optimal control of a queueing system with two heterogeneous servers. *IEEE Transactions on Automatic Control*, 29, 1984.

[12] Ahmed Mokhtar and Murat Azizolu. Adaptive wavelength routing in all-optical networks. *IEEE/ACM Trans. Netw.*, 6(2):197–206, 1998.

[13] Asis Nasipuri, Robert Castaeda, and Samir R. Das. Performance of multipath routing for on-demand protocols in mobile ad hoc networks. *Mobile Networks and Applications, Kluwer Academic Publishers. Manufactured in The Netherlands*, 6:339–349, 2001.

[14] M. F. Neuts. *Matrix Geometric Solutions in Stochastic Models*. John Hopkins University Press, Baltimore, 1980.

[15] M. F. Neuts and D. M. Lacantoni. A Markovian queue with N servers subject to breakdowns and repairs. *Management science*, 25:849–861, 1979.

[16] P. Papadimitratos and Z. J. Haas. Secure message transmission in mobile ad hoc networkstochastic dominance and expected utility: Survey and analysis. *Ah Hoc Networks*, 1:193–209, 2003.

[17] M. Pearlman, Z. Haas, P. Sholander, and S. Tabrizi. the impact of alternate path routing for load balancing in mobile ad hoc networks. *In Proceedings of the ACM MobiHoc*, pages 3–10, 2000.

[18] B. Ramamurthy and B. Mukherjee. Wavelength conversion in wdm networking. *IEEE Journal on Selected Areas in Communications*, 16(7):1061–1073, September 1998.

[19] B. M. Rao and M. J. M. Posner. Algorithmic and approximation analyses of the shorter queue model. *Naval Research Logistics*, 34:381–398, 1977.

[20] A. Riska and E. Smirni. M/G/1-type markov processes: A tutorial. http://www.cs.wm.edu/ riska/paper-MG1-tutorial.pdf.

[21] A. Tsirigos and Z.J. Haas. Multipath routing in the presence of frequent topological changes. *IEEE Comm. Mag.*, pages 132–138, November 2001.

[22] Le Boudec J. Y. An efficient solution method for Markov models of ATM links with loss priorities, April 1991.

[23] J. M. Yates, M. P. Rumsewicz, and J. P. R. Lacey. Wavelength converters in dynamically-reconfigurable wdm networks. *IEEE Communications Surveys & Tutorials*, 2(2):2–15, Second Quarter 1999.