ON THE GENERALITY OF BINARY TREE-LIKE MARKOV CHAINS

K. SPAEY , B. VAN HOUDT , AND C. BLONDIA *

Abstract. In this paper we show that an arbitrary tree-like Markov chain can be embedded in a binary tree-like Markov chain with a special structure. When combined with [7], this implies that any tree structured QBD Markov chain can be reduced to a binary tree-like process. Furthermore, a simple relationship between the V, R_s and G_s matrices of the original and the binary tree-like Markov chain is established. We also explore the effectiveness of computing the steady state probabilities from the reduced binary chain.

Key words. matrix analytic methods, (binary) tree-like Markov chains, embedded Markov chains

1. Introduction. Tree structured Quasi-Birth-Death (QBD) Markov chains were first introduced in 1995 by Takine et al [5] and later, in 1999, by Yeung et al [10]. More recently, Bini et al [1] have defined the class of tree-like processes as a specific sub-class of the tree structured QBD Markov chains. In [7] it was shown that any tree structured QBD Markov chain can be embedded in a tree-like process. Moreover, the natural fixed point iteration (FPI) to the nonlinear matrix equation $V = B + \sum_{s=1}^{d} U_s(I - V)^{-1}D_s$ that solves the tree-like process, was proven to be equivalent to the more complicated iterative algorithm presented by Yeung and Alfa [10]. In this paper, we demonstrate that any tree-like process can be reduced to a binary tree-like process (i.e., a tree-like process with $d = 2$). Thus, combined with [7], this implies that any tree structured QBD Markov chain can be embedded in a binary tree-like process. We also clarify the relationship between the V, R_s and G_s matrices of the original tree-like process and the binary one. The contribution made by this paper is mostly of theoretical interest, because a careful study on the effectiveness of computing the steady state probabilities from the reduced binary chain, seems to indicate that the reduction technique does not give rise to a speed-up of the iterative algorithms involved.

Typical applications of tree-like processes include preemptive and non-preemptive single server queues with a LCFS service discipline that serves customers of different types, where each type has a different service requirement [5, 10, 3, 2, 11]. Tree structured QBD Markov chains have also been used to evaluate conflict resolution algorithms of the Capetanakis-Tsybakov-Mikhailov-Vvedenskaya (CTMV) type [6, 9]. Some recent work also indicates that tree-like processes can be used to study FCFS priority queues with three service classes [8].

2. Tree-like quasi-birth-death processes - a review. The set of tree-like processes [1] was first introduced as a subclass of the set of tree structured Quasi-Birth-Death Markov chains and afterward shown to be equivalent [7]. This section provides some background information on this type of discrete time Markov chains (MCs). Consider a discrete time bivariate MC $\{(X_t, N_t), t \geq 0\}$ in which the values of X_t are represented by nodes of a d-ary tree, for $d \geq 2$, and where N_t takes integer values between 1 and m. We will refer to X_t as the *node* and to N_t as the *auxiliary* variable of the MC at time t. With some abuse of notation, we shall refer to this MC as (X_t, N_t). The root node of the d-ary tree is denoted as $\emptyset$ and the remaining

* University of Antwerp - IBBT, Department of Mathematics and Computer Science, Performance Analysis of Telecommunication Systems Research Group, Middelheimlaan, 1, B-2020 Antwerp - Belgium, {kathleen.spaey,benny.vanhoudt,chris.blondia}@ua.ac.be.

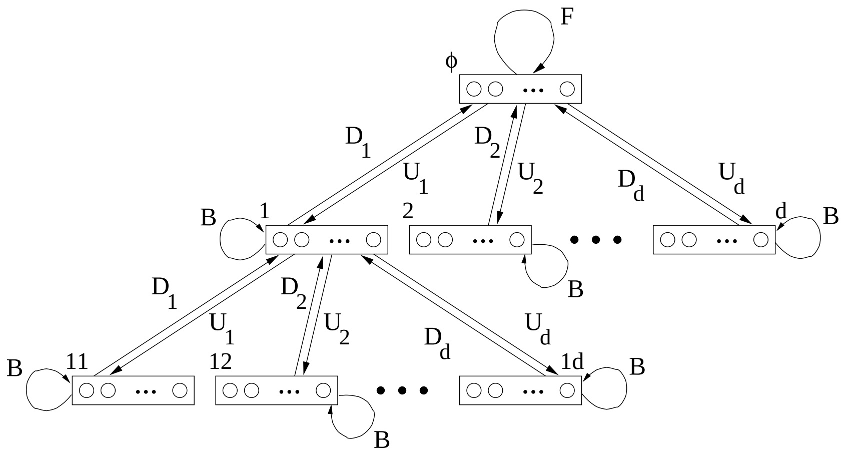

FIG. 2.1. *The structure of a tree-like Markov chain and the matrices characterizing its transitions.*

nodes are denoted as strings of integers, where each integer takes a value between 1 and d. For instance, the k-th child of the root node is represented by k, the l-th child of the node k by kl, and so on. Throughout this paper, we use the '+' to denote the concatenation on the right and '$-$' to represent the deletion from the right. For example, if $J = k_1k_2 \ldots k_n$, then $J + k = k_1k_2 \ldots k_nk$. Let $f(J, k)$, for $J \neq \emptyset$, denote the k rightmost elements of the string J, then $J - f(J, 1)$ represents the parent node of J.

The following restrictions need to apply for an MC (X_t, N_t) to be a tree-like process. At each step the chain can only make a transition to its parent (i.e., $X_{t+1} = X_t - f(X_t, 1)$, for $X_t \neq \emptyset$), to itself ($X_{t+1} = X_t$), or to one of its children ($X_{t+1} = X_t + s$ for some $1 \leq s \leq d$). Moreover, the state of the chain at time $t+1$ is determined as follows:

$$P[(X_{t+1}, N_{t+1}) = (J', j)|(X_t, N_t) = (J, i)] =$$
$$\begin{cases} f^{i,j} & J' = J = \emptyset, \\ b^{i,j} & J' = J \neq \emptyset, \\ d_k^{i,j} & J \neq \emptyset, f(J,1) = k, J' = J - f(J,1), \\ u_s^{i,j} & J' = J + s, s = 1, \ldots, d, \\ 0 & otherwise. \end{cases}$$

Notice, the transition probabilities between two nodes depend only on the spacial relationship between the two nodes and not on their specific values.

We can now define the $m \times m$ matrices D_k, B, F and U_s with respective $(i, j)^{th}$ elements given by $d_k^{i,j}$, $b^{i,j}$, $f^{i,j}$ and $u_s^{i,j}$. This completes the description of the tree-like process. Notice, a tree-like process is fully characterized by the matrices D_k, B, U_s and F (see Figure 2.1).

Next, we introduce a number of matrices that play a crucial role when studying the stability and stationary behavior of a tree-like process. The fundamental period of a tree-like process starting in state $(J + k, i)$ is defined as the first passage time from the state $(J + k, i)$ to one of the states (J, j), for $j = 1, \ldots, m$. Let G_k, for $1 \leq k \leq d$, denote the matrix whose $(i, v)^{th}$ element is the probability that the MC is in state (J, v) at the end of a fundamental period which started in state $(J + k, i)$. Let the $(i, v)^{th}$ element of the matrix R_k, for $1 \leq k \leq d$, denote the expected number of visits to state $(J+k,v)$ before visiting node J again, given that $(X_0,N_0)=(J,i)$. Finally, let V denote the matrix whose $(i,v)^{th}$ element is the taboo probability that starting from state $(J+k,i)$, the process eventually returns to node $J+k$ by visiting $(J+k,v)$, under the taboo of the node J. Notice, due to the restrictions on the transition probabilities, the matrix V does not depend on k. Yeung and Alfa [11] were able to show that the following expressions hold for these matrices:

$$G_k = (I-V)^{-1}D_k,$$

$$R_k = U_k(I-V)^{-1},$$

$$V = B + \sum_{s=1}^{d} U_s G_s.$$

Combining these equations, we have the following relation:

$$V = B + \sum_{s=1}^{d} U_s (I-V)^{-1} D_s.$$

Provided that the tree-like process $\{(X_t,N_t), t \geq 0\}$ is ergodic (which is the case if and only if all the G_k matrices are stochastic or likewise if and only if the spectral radius of $R = R_1 + \ldots + R_d$ is less than one [10]), define its steady state probabilities as

$$\pi_i(J) = \lim_{t\to\infty} P[X_t = J, N_t = i],$$
$$\pi(J) = (\pi_1(J), \pi_2(J), \ldots, \pi_m(J)).$$

The vectors $\pi(J)$ can be computed from $\pi(\emptyset)$ using the relation $\pi(J+k)=\pi(J)R_k$. $\pi(\emptyset)$ is found by solving the boundary condition $\pi(\emptyset)=\pi(\emptyset)(\sum_k R_k D_k + F)$ with the normalizing restriction that $\pi(\emptyset)(I-R)^{-1}e = 1$ (where e is a column vector with all its entries equal to one).

In this paper we will consider a tree-like process with a somewhat more general boundary condition as a starting point. We extend the state space of (X_t,N_t) with a single node $\emptyset_s$ consisting of m_s states. This node acts as a *super* root and transitions from and to this node can only occur via the node $\emptyset$. Transitions from, to and within node $\emptyset_s$ are characterized by the $m_s \times m$, $m \times m_s$ and $m_s \times m_s$ matrices $F_{s\rightarrow}, F_{\rightarrow s}$ and F_s, respectively. Apart from $\pi(\emptyset)$ and $\pi(\emptyset_s)$, the computation of the vectors $\pi(J)$ is not affected by adding the node $\emptyset_s$. We expand the MC with the node $\emptyset_s$, because the reduced tree-like process in [7] contains such a node, as do some applications (e.g., [8]).

3. Constructing the binary tree-like processes. The idea behind the construction used to reduce a tree-like process to a binary one exists in representing each integer part of a string J as a star followed by a series of zeros. For instance, $X_t = J = j_1 \ldots j_n$ is represented in its binary form as

$$\psi(J) \stackrel{def.}{=} * \overbrace{0\ldots0}^{j_1-1} * \overbrace{0\ldots0}^{j_2-1} * \ldots * \overbrace{0\ldots0}^{j_n-1}.$$

We will also denote $\psi(\emptyset) = \emptyset$ and $\psi(\emptyset_s) = \emptyset_s$. Obviously, simply representing all strings J by a binary string does not make $(\psi(X_t), N_t)$ a binary tree-like process, as a single transition may add/remove a series of zeros preceded by a single star.

To reduce $(\psi(X_t), N_t)$ into a binary tree-like process we construct an expanded MC $(\mathcal{X}_n, \mathcal{N}_n = (\mathcal{Q}_n, \mathcal{M}_n))$. The technique used to set-up this expanded MC has some similarity with Ramaswami's [4] to reduce a classic M/G/1-type MC to a QBD MC or to the approach taken in Van Houdt *et al* [9] to construct a tree structured QBD.

The MC $(\mathcal{X}_n, \mathcal{N}_n)$ is defined on the state space $\Omega = \{(\emptyset_s, (0, i))|i = 1, \ldots, m_s\} \cup \{(\emptyset, (0, i))|i = 1, \ldots, m\} \cup \{(J, (a, i))|J = *j_1j_2\ldots j_n; j_k = 0 \text{ or } *; k = 1, \ldots, n; n \geq 0; a = -(d-1), \ldots, d-1; i = 1, \ldots, m\}$[1]. We will establish a one-to-one correspondence between the state (J, i) of the original chain and the state $(\psi(J), (0, i))$ of the expanded chain (for all J and i). The key idea behind establishing this association is that whenever a transition occurs that adds a series of $k-1$ zeros preceded by a star to the node variable $\psi(X_t) = J$, we split this transition into k transitions that each add one symbol at a time. Similarly, the removal of a star followed by $k-1$ zeros from $\psi(X_t)$ will be split into k transitions that each remove one symbol. The role of the random variable $\mathcal{Q}_n$ is as follows. Having $\mathcal{Q}_n = a < 0$, implies that a series of zeros is being removed and so far $-a$ zeros have been removed. While $\mathcal{Q}_n = a > 0$ indicates that a more zeros need to be added to the string $\mathcal{X}_n$.

More formally, consider a realization $(X_t(w), N_t(w))$ of the Markov chain (X_t, N_t). The corresponding realization of the expanded chain $(\mathcal{X}_n, \mathcal{N}_n = (\mathcal{Q}_n, \mathcal{M}_n))$ is defined as follows.

Initial state: If $(X_0(w), N_0(w)) = (J, i)$, then set $(\mathcal{X}_0(w), \mathcal{N}_0(w)) = (\psi(J), (0, i))$. Also, set $t = 0$ and $n = 0$; t represents the steps of the original chain (X_t, N_t) and n represents the steps of the expanded chain.

Transition Rules: We distinguish between three possible cases: $\mathcal{Q}_n(w) = 0$, $\mathcal{Q}_n(w) > 0$ and $\mathcal{Q}_n(w) < 0$.

1. $\mathcal{Q}_n(w) = 0$, consider $(X_t(w), N_t(w))$, and do one of the following:
   a. Suppose $X_{t+1}(w) = X_t(w) + k = J + k$, for some $1 \leq k \leq d$ and string $J \neq \emptyset_s$. Let $\mathcal{X}_{n+1}(w) = \mathcal{X}_n(w) + * = \psi(J) + *$ and $\mathcal{N}_{n+1}(w) = (k-1, N_{t+1}(w))$.
   b. Given $X_{t+1}(w) = X_t(w) - k = J$, for some $1 \leq k \leq d$ and string $J \neq \emptyset_s$. Notice, if $k > 1$, then $\psi(J + k)$ ends on a zero (while for $k = 1$, it ends on a star) and we can define $\mathcal{X}_{n+1}(w) = \psi(J + k) - 0$ and $\mathcal{N}_{n+1}(w) = (-1, N_t(w))$ (notice, $\mathcal{M}_{n+1}(w) = N_t(w)$ and not $N_{t+1}(w)$). For $k = 1$, set $\mathcal{X}_{n+1}(w) = \psi(J + k) - * = \psi(J)$ and $\mathcal{N}_{n+1}(w) = (0, N_{t+1}(w))$.
   c. In all other cases (with $\mathcal{Q}_n(w) = 0$), set $\mathcal{X}_{n+1}(w) = \psi(X_{t+1}(w))$ and $\mathcal{N}_{n+1}(w) = (0, N_{t+1}(w))$. These cases include the transitions to and from $\emptyset_s$ and those for which $X_{t+1}(w) = X_t(w)$.

   Next, both n and t are incremented by one.
2. $\mathcal{Q}_n(w) > 0$, define $\mathcal{X}_{n+1}(w) = \mathcal{X}_n(w) + 0$ and $\mathcal{N}_{n+1}(w) = (\mathcal{Q}_n(w) - 1, \mathcal{M}_n(w))$. Remark, if $\mathcal{Q}_{n+1}(w)$ becomes zero, then $\mathcal{X}_{n+1}(w) = \psi(X_t(w))$. Next, increase n by one and do not alter the value of t.
3. $\mathcal{Q}_n(w) < 0$, consider $\mathcal{X}_n(w)$ and distinguish between the following two cases:
   a. Assume $f(\mathcal{X}_n(w), 1) = *$. Let $\mathcal{X}_{n+1}(w) = \mathcal{X}_n(w) - * = \psi(X_t(w))$ and $\mathcal{N}_{n+1}(w) = (0, N_t(w))$.

[1]Many of these states will be transient. However, their corresponding entries in the steady state probability vectors will automatically become zero, so there is no need to remove them.

b. Assume $f(\mathcal{X}_n(w), 1) = 0$. Set $\mathcal{X}_{n+1}(w) = \mathcal{X}_n(w) - 0$ and $\mathcal{N}_{n+1}(w) = (\mathcal{Q}_n(w) - 1, \mathcal{M}_n(w))$.

Next, increase n by one and do not alter the value of t.

Next, we show that $(\mathcal{X}_n, \mathcal{N}_n)$ is a (binary) tree-like process with a generalized boundary condition. Indeed, if we remove the nodes $\emptyset$ and $\emptyset_s$ from the state space Ω we end up with a state space of a standard tree-like process, where the star node figures as the root node. Moreover, the string $\mathcal{X}_n$ never grows/shrinks by more than one symbol at a time and it can be readily seen from its construction that the transition between different nodes only depends upon their spacial relationship as required. When describing the transition matrices that characterize $(\mathcal{X}_n, \mathcal{N}_n)$, we will add a hat to all matrices involved (if a conflict arises with earlier notations). Furthermore, the transition matrices Z between two nodes J and J' that both start with a star, are partitioned into nine submatrices as follows:

$$Z = \begin{bmatrix} Z_{-,-} & Z_{-,0} & Z_{-,+} \\ Z_{0,-} & Z_{0,0} & Z_{0,+} \\ Z_{+,-} & Z_{+,0} & Z_{+,+} \end{bmatrix},$$

where $Z_{x,y}$ with $x, y \neq 0$ are square matrices of dimension $(d-1)m$, while $Z_{0,0}$ is a square matrix of size m and all other matrices have an appropriate dimension such that Z is square. The subscripts of these matrices refer to the signs of $\mathcal{Q}_n$ and $\mathcal{Q}_{n+1}$. Let I_x denote the unit matrix of size x. In case $x = m$ (with m the number of values N_t and $\mathcal{M}_n$ can take), we drop the subscript.

A star is only added to $\mathcal{X}_n$ in case 1a, while the addition of a zero only occurs in case 2. Hence,

$$\begin{bmatrix} (U_*)_{0,0} & (U_*)_{0,+} \end{bmatrix} = \begin{bmatrix} U_1 & U_2 & U_3 & \dots & U_d \end{bmatrix},$$

$$\begin{bmatrix} (U_0)_{+,0} & (U_0)_{+,+} \end{bmatrix} = \begin{bmatrix} I & 0 & 0 & \dots & 0 \\ 0 & I & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & I & 0 \end{bmatrix},$$

where $U_1, \dots, U_d$ are the $m \times m$ matrices belonging to the MC (X_t, N_t). All other blocks of the matrices U_0 and U_* are zero. According to cases 1b, 3a and 3b, the non-zero blocks of the matrices D_0 and D_* equal

$$\begin{bmatrix} (D_*)_{-,0} \\ (D_*)_{0,0} \end{bmatrix} = \begin{bmatrix} D_2^T \Delta^{-1} & D_3^T \Delta^{-1} & \dots & D_d^T \Delta^{-1} & D_1^T \end{bmatrix}^T,$$

$$\begin{bmatrix} (D_0)_{-,-} & (D_0)_{-,0} \\ (D_0)_{0,-} & (D_0)_{0,0} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 & \dots & 0 \\ 0 & 0 & I & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & I \\ \Delta & 0 & \dots & 0 & 0 \end{bmatrix},$$

where Δ is a diagonal matrix with entries[2] $D_k e$ and ^{T} denotes the transpose of a matrix. In practice (e.g., [6, 9]) Δ need not be invertible as some of its diagonal

[2]Remark, the vectors $D_k e$, $1 \leq k \leq d$, are all identical.

entries might be zero, say those appearing on rows $\mathcal{S} \subset \{1, \ldots, m\}$. In this case all the states of the form $(J, (a, s))$ are transient for $a < 0$ and $s \in \mathcal{S}$ and can be removed at once. Hence, it suffices that Δ can be inverted after removing the rows and columns corresponding to these states. For ease of notation, we assume that $\mathcal{S}$ is empty. The identity matrices in D_0 are a consequence of case 3b. The appearance of the Δ matrix is caused by case 1b for $k > 1$, as we remove a zero and keep $\mathcal{M}_{n+1}(w)$ equal to $\mathcal{M}_n(w)$ irrespective of the value of k. The matrices of the form $\Delta^{-1} D_k$ make sure that $\mathcal{M}_{n+1}(w) = N_t(w)$ in case 3a. This construction is needed as we cannot determine the correct value of k until we have removed all the necessary zeros (which are counted by $\mathcal{Q}_n(w)$). Transitions of the MC $(\mathcal{X}_t, \mathcal{N}_t)$ from a node J, which differs from $\emptyset_s$ and $\emptyset$, to itself are captured by the matrix $\hat{B}$; where $\hat{B}_{0,0} = B$, while all other blocks of $\hat{B}$ are identical to zero. Finally, the transitions among the nodes $\emptyset$ and $\emptyset_s$ are still characterized by the matrices F, F_s, $F_{\to s}$ and $F_{s\to}$, while the transition matrix from node $\emptyset$ to node $*$ is identical to $[(U_*)_{0,-}\ (U_*)_{0,0}\ (U_*)_{0,+}]$ and from node $*$ to node $\emptyset$ is given by $[(D_*)^T_{-,0}\ (D_*)^T_{0,0}\ (D_*)^T_{+,0}]^T$.

4. Structural properties of the $\hat{V}$, R and G matrices. In this section, the structural properties of the matrices $\hat{V}$, R_0, R_*, G_0 and G_* of the MC $(\mathcal{X}_n, \mathcal{N}_n)$ will be discussed and their relationship with the matrices V, G_k and R_k, for $k = 1, \ldots, d$, of the original MC (X_t, N_t) will be identified. $A \otimes B$ denotes the Kronecker product of the matrices A and B.

Consider the matrix $\hat{V}$ whose elements labeled $((a, i), (a', i'))$ are the the taboo probabilities that starting from a state $(J + k, (a, i))$, for $k = 0$ or $*$, the process $(\mathcal{X}_n, \mathcal{N}_n)$ eventually returns to the node $J+k$ by visiting the state $(J+k, (a', i'))$, under the taboo of node J. By construction of $(\mathcal{X}_n, \mathcal{N}_n)$, every sample path in $(\mathcal{X}_n, \mathcal{N}_n)$ that starts and ends in a state with $\mathcal{Q}_n = 0$, and that does not visit any other such state, corresponds to a single transition in (X_t, N_t). So with every path in $(\mathcal{X}_n, \mathcal{N}_n)$ that starts in the state $(J + k, (0, i))$ and that eventually returns to the node $J + k$ under the taboo of node J by visiting the state $(J + k, (0, i'))$, there corresponds exactly one path in (X_t, N_t), namely a path starting in state $(\psi^{-1}(J + k), i)$ that eventually returns to node $\psi^{-1}(J + k)$ by visiting the state $(\psi^{-1}(J + k), i')$. By construction of $(\mathcal{X}_n, \mathcal{N}_n)$, both these sample paths occur with the same probability. As a consequence, $\hat{V}_{0,0} = V$. Any sample path of $(\mathcal{X}_n, \mathcal{N}_n)$ starting from a state $(J + k, (0, i))$ to the same node $J + k$, under taboo of its parent node J, is either of length one, or starts by adding a star to the string $J + k$. Due to the structure of $\hat{B}$ and D_*, this implies that $\hat{V}_{0,+} = 0$ and $\hat{V}_{0,-} = 0$.

If a path in $(\mathcal{X}_n, \mathcal{N}_n)$ starts in a node $(J+k, (a, i))$ with $a < 0$, this means that the process is in the course of removing symbols from the string $J + k$, so only transitions to node J are possible from such a state. So by definition of $\hat{V}$, $\hat{V}_{-,-} = \hat{V}_{-,+} = 0$ and $\hat{V}_{-,0} = 0$. In case a path in $(\mathcal{X}_n, \mathcal{N}_n)$ starts in a node $(J + k, (a, i))$ with $a > 0$, the process will, with probability one, reach the state $(J', (0, i))$ after making a transitions, where $J' = J + k + s_1 + ... + s_a$, with all $s_i = 0$, for $i = 1, \ldots, a$. Having reached $(J', (0, i))$, the process will follow a path starting from this state that passes l times through the node J' again, before eventually reaching node $J' - f(J', 1)$ by visiting some state $(J' - f(J', 1), (-1, i'))$. The probability of all these paths is given by the (i, i')-th element of the matrix $(\sum_{l=0}^{\infty} \hat{V}_{0,0}^l \Delta) = (\sum_{l=0}^{\infty} V^l)\Delta = (I - V)^{-1}\Delta$. In case $a = 1$, the process has now reached the node $J + k$ again, otherwise it will still make $a - 1$ transitions with probability one, after which it reaches the node $J + k$ again via the state $(J + k, (-a, i'))$. So $\hat{V}_{+,-} = I_{d-1} \otimes (I - V)^{-1}\Delta$, $\hat{V}_{+,0} = 0$, and $\hat{V}_{+,+} = 0$.

Due to the structure of $\hat{V}$, the only non-zero block of $\hat{V}^k$, $k \geq 2$, is $(\hat{V}^k)_{0,0} = V^k$. Hence,

$$(I - \hat{V})^{-1} = \sum_{k=0}^{\infty} \hat{V}^k = \begin{bmatrix} I_{d-1} \otimes I & 0 & 0 \\ 0 & (I-V)^{-1} & 0 \\ I_{d-1} \otimes (I-V)^{-1}\Delta & 0 & I_{d-1} \otimes I \end{bmatrix}.$$

Using the expressions $R_0 = U_0(I_{(2d-1)m} - \hat{V})^{-1}$ and $R_* = U_*(I_{(2d-1)m} - \hat{V})^{-1}$, we find that the non-zero entries of R_0 and R_* are equal to

$$\begin{gathered}[\ (R_0)_{+,-} \quad (R_0)_{+,0} \quad (R_0)_{+,+}\ ] = \\ \begin{bmatrix} 0 & 0 & (I-V)^{-1} & 0 & 0 \\ I_{d-2} \otimes (I-V)^{-1}\Delta & 0 & 0 & I_{d-2} \otimes I & 0 \end{bmatrix},\end{gathered}$$

and

$$[\ (R_*)_{0,-} \quad (R_*)_{0,0} \quad (R_*)_{0,+}\ ] = [\ R_2\Delta \quad \dots \quad R_d\Delta \quad R_1 \quad U_2 \quad \dots \quad U_d\ ].$$

Remark, $(R_0)^d = 0$, which is as expected as there can be at most $d-1$ consecutive zeros in a binary representation $\psi(J)$ of any string J. Analogously, the non-zero components of $G_0 = (I_{(2d-1)m} - \hat{V})^{-1} D_0$ and $G_* = (I_{(2d-1)m} - \hat{V})^{-1} D_*$ can be written as

$$\begin{bmatrix} (G_0)_{-,-} & (G_0)_{-,0} \\ (G_0)_{0,-} & (G_0)_{0,0} \\ (G_0)_{+,-} & (G_0)_{+,0} \end{bmatrix} = \begin{bmatrix} 0 & I_{d-1} \otimes I \\ (I-V)^{-1}\Delta & 0 \\ 0 & I_{d-1} \otimes (I-V)^{-1}\Delta \end{bmatrix},$$

and

$$[\ (G_*)^T_{-,0} \quad (G_*)^T_{0,0} \quad (G_*)^T_{+,0}\ ]^T = [\ D_2^T \Delta^{-1} \quad \dots \quad D_d^T \Delta^{-1} \quad G_1^T \quad \dots \quad G_d^T\ ]^T.$$

Remark that $\Delta e = D_k e$, meaning $\Delta^{-1} D_k e = e$, for $k \in \{1, \dots, d\}$. As a consequence, G_0 and G_* are stochastic if and only if all the matrices G_k, $k \in \{1, \dots, d\}$, of (X_t, N_t) are stochastic (as $(I-V)^{-1}\Delta e = (I-V)^{-1} D_k e = G_k e$). This means that the binary tree-like process $(\mathcal{X}_n, \mathcal{N}_n)$ is ergodic if and only if the tree-like process (X_t, N_t) is ergodic.

5. Computing steady state probabilities. From the previous section, it is clear that any algorithm that computes $\hat{V}$ produces the matrix V as a by-product and vice versa. The steady state probabilities of $(\mathcal{X}_n, \mathcal{N}_n)$ and (X_t, N_t), provided that the MC is stationary, can be easily computed from $\hat{V}$ and V, respectively, as explained in Section 2. In this section we will demonstrate that the use of some algorithms to compute $\hat{V}$, is equivalent to the computation of V via the same algorithm, provided that we make use of the structural properties of the matrices involved.

5.1. Fixed point iteration (FPI) [10, 1]. This algorithm computes $\hat{V}$ as follows. Set $\hat{V}[0] = \hat{B}$ and compute $\hat{V}[N+1]$ as

$$\hat{V}[N+1] = \hat{B} + U_*(I_{(2d-1)m} - \hat{V}[N])^{-1} D_* + U_0(I_{(2d-1)m} - \hat{V}[N])^{-1} D_0. \tag{5.1}$$

In this case $\hat{V}[N]$ monotonically converges to $\hat{V}$. It is easily seen that more iterations are required to compute $\hat{V}$, when compared to computing V via the FPI algorithm. We can improve the convergence by taking the specific structure of the $\hat{V}$ matrix into account. That is, it suffices to compute the component $\hat{V}_{0,0}[N+1]$ via (5.1) and to update the other entries such that $\hat{V}[N+1]$ has the same form as $\hat{V}$. Using the expressions for U_*, U_0, D_* and D_0 we have

$$\hat{V}_{0,0}[N+1] = B + \sum_{i=1}^{d} U_i(I - \hat{V}_{0,0}[N])^{-1} D_i,$$

which is identical to applying the FPI algorithm to V.

5.2. Reduction to quadratic equations (RQE) [1]. This algorithm allows us to compute G_0 and G_* iteratively, from which we can derive the matrix $\hat{V}$. We start with $G_0[0] = G_*[0] = 0$ and solve the following two quadratic equations to obtain $G_0[N+1]$ and $G_*[N+1]$ from $G_0[N]$ and $G_*[N]$:

$$0 = D_0 + \left(\hat{B} - I_{(2d-1)m} + U_* G_*[N]\right) G_0[N+1] + U_0 G_0^2[N+1], \tag{5.2}$$

$$0 = D_* + \left(\hat{B} - I_{(2d-1)m} + U_0 G_0[N+1]\right) G_*[N+1] + U_* G_*^2[N+1], \tag{5.3}$$

where $G_0[N]$ and $G_*[N]$ converge to G_0 and G_*, respectively. This iterative procedure converges more slowly when compared to applying the RQE algorithm to the MC (X_t, N_t). This can be seen by realizing that the $G_*[N]$ and $G_0[N]$ matrices hold the first-passage probabilities from the node $**$ and $*0$ to the node $*$ in the trees $\mathcal{T}_{N,*}$ and $\mathcal{T}_{N,0}$, respectively, with the tree $\mathcal{T}_{N,*} = \{\emptyset_s, \emptyset, *\} \cup \{*(*^{s_1}0^{s_2})^N i | i = 0 \text{ or } *; s_1, s_2 \geq 0\}$ and $\mathcal{T}_{N,0} = \{\emptyset_s, \emptyset, *\} \cup \{*0^{s_1}(*^{s_2}0^{s_3})^{N-1} i | i = 0 \text{ or } *; s_1, s_2, s_3 \geq 0\}$, for $N \geq 1$. Applying the RQE algorithm to the MC (X_t, N_t), however, generates a sequence of matrices $G_k[N]$, for $k = 1, \ldots, d$, where $G_k[N]$ holds the first-passage probabilities from the node k to the node $\emptyset$ in the tree $\mathcal{T}_{N,k} = \{\emptyset_s, \emptyset\} \cup \{(k)^{s_1} \ldots 1^{s_k}(d^{s_{k+1}} \ldots 1^{s_{d+k}})^{N-1} i | i = 1, \ldots, d; s_j \geq 0; j = 1, \ldots, d+k\}$. Thus, a sample path that visits a node J containing a series of identical integers $k > 1$ is more rapidly taken into account in the latter case.

We can improve upon (5.2)-(5.3) by taking the structure of the matrices involved into account. More specifically, the matrix equation $D_0 + (\hat{B} - I_{(2d-1)m} + U_* G_*) G_0 + U_0 G_0^2 = 0$ can be simplified to

$$0 = \Delta + \left(B - I + \sum_{\substack{1 \leq j \leq d \\ j \neq k}} U_j G_j \right) (I - V)^{-1} \Delta + U_k G_k (I - V)^{-1} \Delta.$$

If we post-multiply this equation by $\Delta^{-1} D_k$ for $k = 1, \ldots, d$, we end up with the d quadratic equations used by the RQE algorithm when applied to the MC (X_t, N_t).

5.3. Newton's iteration (NI) [1]. The NI algorithm can be used to compute the $m \times m$ matrices G_k of a tree-like process in a quadratically converging manner. However, each step requires the solution of a (large) linear system of equations of the form: $\sum_{k=1}^{d} H_k X K_k = X + L$ for some square matrices H_k, K_k and L of dimension m. Thus, after applying the reduction technique presented in this paper, it suffices to develop an efficient algorithm to solve a system of the form $H_1 X K_1 + H_2 X K_2 = X + L$, which is identical to a generalized Sylvester matrix equation, except for the X appearing on the right-hand side. Currently, it is unclear whether such a simplification can result in a computational gain.

Acknowledgements. The second author is a postdoctoral Fellow of the FWO-Flanders. This work was partly funded by the IWT project CHAMP "Cross-layer planning of Home and Access networks for Multiple Play".

REFERENCES

[1] D.A. Bini, G. Latouche, and B. Meini, *Solving nonlinear matrix equations arising in tree-like stochastic processes*, Linear Algebra Appl., 366 (2003), pp. 39–64.

[2] Q. He and A.S. Alfa, *The MMAP[K]/PH[K]/1 queues with a last-come-first-serve preemptive service discipline*, Queueing Systems, 28 (1998), pp. 269–291.

[3] ———, *The discrete time MMAP[K]/PH[K]/1/LCFS-GPR queue and its variants*, in Proc. of the 3rd Int. Conf. on Matrix Analytic Methods, Leuven (Belgium), 2000, pp. 167–190.

[4] V. Ramaswami, *The generality of QBD processes*, in Advances in Matrix Analytic Methods for Stochastic Models, Notable Publications Inc., Neshanic Station, NJ, 1998, pp. 93–113.

[5] T. Takine, B. Sengupta, and R.W. Yeung, *A generalization of the matrix M/G/1 paradigm for Markov chains with a tree structure*, Stochastic Models, 11 (1995), pp. 411–421.

[6] B. Van Houdt and C. Blondia, *Stability and performance of stack algorithms for random access communication modeled as a tree structured QBD Markov chain*, Stochastic Models, 17 (2001), pp. 247–270.

[7] ———, *Tree structured QBD Markov chains and tree-like QBD processes*, Stochastic Models, 19 (2003), pp. 467–482.

[8] ———, *Analyzing priority queues with 3 classes using tree-like processes*, Submitted for publication, (2005).

[9] ———, *Throughput of Q-ary splitting algorithms for contention resolution in communication networks*, Communications in Information and Systems, 4 (2005), pp. 135–164.

[10] R.W. Yeung and A.S. Alfa, *The quasi-birth-death type Markov chain with a tree structure*, Stochastic Models, 15 (1999), pp. 639–659.

[11] R.W. Yeung and B. Sengupta, *Matrix product-form solutions for Markov chains with a tree structure*, Adv. Appl. Prob., 26 (1994), pp. 965–987.