

TecnoLógicas

Tecno Lógicas

ISSN: 0123-7799

tecnologicas@itm.edu.co

Instituto Tecnológico Metropolitano  
Colombia

Gómez-Vargas, Ricardo C.; Espinosa-Buitrago, Mónica  
Generación y Análisis de Algoritmos para el Estudio de QBDs  
Tecno Lógicas, octubre, 2013, pp. 577-589  
Instituto Tecnológico Metropolitano  
Medellín, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=344234341044>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica  
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal  
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

## **Generación y Análisis de Algoritmos para el Estudio de QBDs**

### ***Generation and Analysis of Algorithms for the Study of QBDs***

Ricardo C. Gómez-Vargas<sup>1</sup>  
Mónica Espinosa-Buitrago<sup>2</sup>

- 
- 1 Facultad de Ingeniería de Telecomunicaciones,  
Universidad Santo Tomás, Bogotá-Colombia  
[ricardogomez@usantotomas.edu.co](mailto:ricardogomez@usantotomas.edu.co)
  - 2 Facultad de Ingeniería de Telecomunicaciones,  
Universidad Santo Tomás, Bogotá-Colombia  
[monica.espinosa@usantotomas.edu.co](mailto:monica.espinosa@usantotomas.edu.co)

**Resumen**

El análisis y caracterización de sistemas de telecomunicaciones generalmente se realiza utilizando modelos estocásticos, en este artículo se presentan dos algoritmos para el análisis de sistemas de cuasi nacimiento y muerte (QBDs), estos se implementan sobre dos tipos de colas PH/PH/1 y M/M/1, mostrando los resultados de los mismos y comparando sus tiempos de ejecución y error de resultados. Para el desarrollo del trabajo se utilizaron análisis matriciales de cadenas de Markov partiendo de la matriz generadora de los mismos.

**Palabras clave**

QBDs; estocásticos; colas; algoritmos; cadenas de Markov.

**Abstract**

The analysis and characterization of telecommunication systems generally is performed using stochastic models, this paper presents two algorithms for the analysis of quasi birth and death (QBDs), these are implemented on two types of queues PH/PH/1 and M/M/1, showing the results of these and comparing their execution times and error results. For development work using matrix analysis based on Markov chains begins from his generated matrix.

**Keywords**

QBDs; Markov chains; query; algorithms; stochastic.

## 1. INTRODUCCIÓN

Los procesos de cuasi nacimiento y muerte (QBDs) se han convertido en una herramienta recurrente en el análisis de sistemas de espera, estudios sobre el desempeño de modelos de radio cognitiva (Martinez-Bauset, 2013), modelamiento de retardos en redes Ad-hoc móviles (XJiang, 2013), distribución anchos de banda variables en sistemas inalámbricos (Fu Yu, 2011), métodos de expansión espectral (Chakka, 2013), entre otros son ejemplos de la importancia que este tipo de análisis posee. Análisis realizados muestran cargas computacionales asociadas a la solución de este tipo de sistemas que pueden llegar a ser considerables (Latouche, 2013), razón por la cual los estudios buscan evaluar diversas formas de encontrar las distribuciones de estado estable de las representaciones en forma de QBDs de sistemas de espera (Gómez, 2011).

Este estudio tiene como objetivo el construir y evaluar algoritmos, con diferentes estrategias de aproximación a la solución, para el análisis de sistemas de espera representados mediante QBDs. La primera parte del documento habla de la descripción de sistemas de espera, la generación de procesos de nacimiento y muerte y de cuasi nacimiento y muerte. Se describe la caracterización de los QBDs y se enuncian los problemas a nivel de cálculo que esta aplicación presenta. Finalmente se formula un experimento que pretende comparar dos estrategias de cálculo matricial para este tipo de sistemas. Finalmente se presenta el desempeño de las rutinas construidas en un sistema PH/PH/1

## 2. SISTEMAS DE ESPERA

Los sistemas de espera modelan el comportamiento de un proceso en lo referente a las políticas de atención que este posea y las medidas de rendimiento (calidad) del mismo, analizar e intervenir el comportamiento de un sistema de espera (denominados colas en gran parte de la literatura) es de importancia en la industria de las telecomunicaciones dado la afinidad de este proceso con la calidad del servicio prestado.

El análisis de sistemas de espera se fundamenta en la construcción de una matriz de probabilidad de transición ( $P$ ) y un vector de estado ( $S$ ) el cual depende del tiempo de observación, para encontrar estos descriptores se asume que el sistema se comporta como un proceso markoviano. Nos centraremos en sistemas cuyas variables de estado (Número de usuarios en el sistema, cantidad de paquetes en buffer, entre otras) incrementan o disminuyen en la unidad, es decir que únicamente existen conexiones entre un estado y sus vecinos.

### 2.1 Procesos de Nacimiento y Muerte

En los análisis de sistemas de espera, cobra importancia el cómo se abandona un estado y se ingresa a otro. Si en la representación observamos que los estados presentes se comunican únicamente con estados que estén a un nodo de distancia de ellos, podemos caracterizar este sistema como proceso de nacimiento y muerte (Fig. 1a), donde  $\mu$  representa los nacimientos o ingresos y  $\lambda$  las muertes o salidas del sistema.

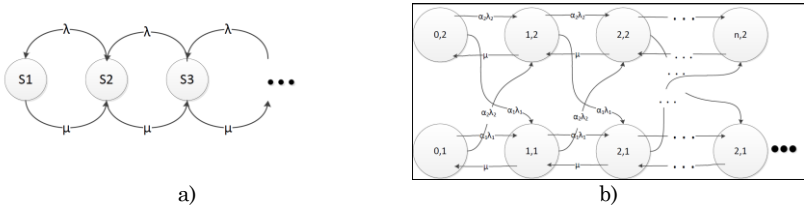


Fig. 1. a) Proceso de nacimiento y muerte, b) Proceso de cuasi nacimiento y muerte.  
Fuente: Autores

Matemáticamente la representación de estos sistemas se desarrolla mediante una matriz  $P$  tridiagonal (1), de dimensión  $N$ , donde  $N$  representa el número de estados del sistema.

### 2.2 Procesos de Cuasi Nacimiento y Muerte

Los procesos de cuasi nacimiento y muerte (QDB) poseen características similares a las vistas en la sección pasada, sin embargo en estos la atención de un servicio puede requerir de un número de fases, es decir que un usuario del sistema puede estar

en el mismo estado en diferentes procesos, en la Fig. 1b podemos observar un QBD, los estados ahora poseen un segundo índice el cual enumera la fase de servicio en la que se encuentra un estado, definiremos  $n$  como el número de estados y  $m$  como el número de fases por estado.

Es notable la diferencia en el comportamiento de este sistema y el anterior, si buscamos una representación matricial del sistema debemos, es necesario plantear un número de matrices que representan los arribos y salidas de los estados dependiendo de la fase en la que se encuentre. Luego de esto podemos definir la matriz de transición de estados (Latouch and V, 1999):

$$P = \begin{bmatrix} B & B_0 & 0 & 0 & 0 & \dots \\ B_2 & A_1 & A_0 & 0 & 0 & \dots \\ 0 & A_2 & A_1 & A_0 & 0 & \dots \\ 0 & 0 & A_1 & A_1 & A_0 & \dots \\ 0 & 0 & 0 & A_2 & A_1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (1)$$

En la anterior expresión cada uno de los elementos representa una matriz de  $n \times n$ . Podemos observar que la complejidad del esquema aumento de manera significativa, la dimensión de la matriz será en términos generales de  $N = m \times n$ , este cambio implica un aumento significativo en el tamaño de la matriz y un aumento importante de los ciclos de cómputo y tiempo de procesamiento de este tipo de sistemas. Lo anterior muestra la necesidad de implementar algoritmos de análisis que reduzcan de alguna manera la carga computacional requerida a la hora de procesar estos sistemas.

### 3. ALGORITMOS DE CÁLCULO

En esta sección nos concentraremos en el cómo realizar los algoritmos bajo los cuales analizar los QBDs, para ello haremos uso de otros descriptores del sistema y la relación de los mismos en el análisis de sistemas de espera. Primero crearemos la matriz generadora que permite realizar cálculos sobre los sistemas de espera, se calcula a partir de las tasas de arribo y de servicio de los esta-

dos existentes. La matriz generadora tiene dimensión  $N$  y sus elementos cumplen con las siguientes condiciones.

$$\sum_{j=1}^N q_{j,i} = 0 \quad 1 \leq i \leq N; \quad \sum_{j=1, j \neq i}^N q_{j,i} = -q_{j,i} \quad 1 \leq i \leq N \quad (2)$$

Dadas las políticas de transición de estado, los QBD poseen diferentes matrices generadora. Sin embargo, el orden de los estados, permite que encuentre una matriz como la mostrada en (3). Vale la pena resaltar que la matriz generadora de un QBD se puede llevar a una forma tridiagonal por bloques, hecho fundamental para el análisis posterior a realizar.

$$Q = \begin{bmatrix} \widehat{B} & \widehat{B}_0 & 0 & 0 & 0 & \dots \\ \widehat{B}_2 & \widehat{A}_1 & \widehat{A}_0 & 0 & 0 & \dots \\ 0 & \widehat{A}_2 & \widehat{A}_1 & \widehat{A}_0 & 0 & \dots \\ 0 & 0 & \widehat{A}_1 & \widehat{A}_1 & \widehat{A}_0 & \dots \\ 0 & 0 & 0 & \widehat{A}_2 & \widehat{A}_1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (3)$$

Podemos definir una matriz  $R$  que contenga los valores esperados del número de visitas a un estado  $n$  partiendo de un estado  $j$ , antes de regresar a un subconjunto de estados  $T$ , si la cadena es irreducible podemos afirmar que  $\pi_D = \pi_T R_{TD}$ , donde  $\pi_D$  representa la probabilidad límite y  $R_{TD}$  es el valor esperado de número de visitas a un estado de un subconjunto D entre dos visitas a un subconjunto T de estados (Latouch and V, 1999).

Definiremos ahora la matriz  $U$ , que representa la probabilidad de visitar un nivel  $n$ , dado que se arrancó en este mismo nivel, antes de visitar un nivel  $n-1$ , los elementos de esta matriz están dados por:

$$U_{ij} = P[\theta < \tau \mid X_\theta = (n, j) \mid X_0 = (n, i)] \quad (4)$$

Donde  $\tau$  define el tiempo de ocurrencia de la primera visita a un estado del nivel  $n-1$  y  $\theta$  es el tiempo de retorno a un estado del nivel  $n$ . podemos calcular el número de ocasiones en que se visita

un estado del nivel  $n$  antes de visitar un estado del nivel  $n-1$  como  $N_V = \sum_{i \geq 0} U^i = (I - U)^{-1}$ . Finalmente se definirá la matriz  $G$ , que representa la probabilidad de visitar un estado del nivel  $n-1$ , dado que se arrancó en un nivel  $n$ , antes en un tiempo finito.

$$G_{ij} = P[\tau < \infty \text{ y } X_T = (n-1, j) | X_0 = (n, i)] \quad (5)$$

Entre las matrices definidas se deben cumplir las siguientes relaciones

$$R = A_0(I - U)^{-1} \quad (6)$$

$$G = (I - U)^{-1}A_2 \quad (7)$$

$$U = A_1 + A_0G \quad (8)$$

$$U = A_1 + RA_2 \quad (9)$$

Podemos observar que si conocemos alguna de las matrices  $R$ ,  $G$  ó  $U$ , es posible encontrar las otras dos. Siendo este el fundamento de los algoritmos a revisar.

### 3.1 Algoritmo de Iteración Lineal

Para desarrollar este algoritmo se usaron las ecuaciones 4 a 7 y dando un valor inicial para  $G$  de cero ( $G(0)=0$ ), realizando sustituciones sucesivas en las expresiones calculamos los valores de  $G$  y de  $U$ , la ejecución del algoritmo termina cuando los cambios de la matriz  $G$  en la ejecución es menor a un valor estipulado e. Podemos realizar variaciones a las condiciones iniciales del anterior algoritmo y decir que  $G(0)=I$ . Si planteamos la misma estrategia de solución que en el caso anterior y obtenemos una variación del primer algoritmo diseñado.

Se debe notar que los algoritmos poseen diferente definición de error, esto se debe a que en el segundo programa la matriz  $Gi$  es una matriz estocástica y la definición de error en el primer algoritmo no sería útil.



### 3.2 Algoritmo de Reducción Logarítmica

En esta implementación se formulara la matriz  $G$  dentro de un sistema cuadrático, el cual se obtiene mezclando (8) y (9), el resultado será:

$$G = A_2 G^2 + A_1 G + A_0 \quad (10)$$

Lo que puede reescribir como:

$$G = L + H G^2 \quad (11)$$

Donde  $L = (I - A_1)^{-1} A_2$  y  $H = (I - A_1)^{-1} A_0$ . Las ecuaciones que hacen parte de este programa, provienen de la construcción de una ecuación de segundo orden en función de la matriz  $G$ , para lograr visualizar el problema se deben mezclar la ecuaciones utilizadas en el apartado anterior. La definición de error de este es similar a la de los anteriores algoritmos, lo que implica que se busca que la matriz  $G$  sea estocástica.

## 4. IMPLEMENTACIÓN

Para realizar pruebas de los algoritmos realizados, se modelará una cola PH/PH/1, la cual se puede representar como en la Fig. 2, los tiempos entre arribos obedece a una distribución de Erlang ( $F_{m,v}$ ) y los tiempos de servicio dependerán el número de fases que posean los estados y la probabilidad de visita a las mismas, para cada usuario los tiempos serán diferentes ya que al terminar el servicio uno se podrá optar por visitar el segundo servicio o salir (con probabilidades  $q$  y  $p$  respectivamente).

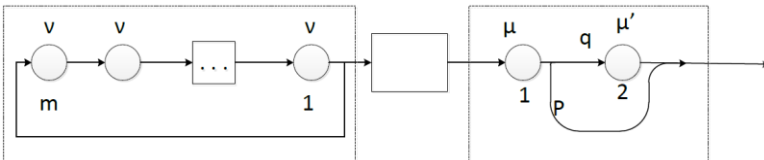


Fig. 2. Modelo de funcionamiento de una cola PH/PH/1. Fuente: Autores

Para modelar la cola se tomaron los siguientes valores:  $m = 3$ , Número de fases 2,  $v = 0.3\mu = 0.6$ ,  $\mu' = 0.4$ ,  $p = 0.6$  y  $q = 0.4$ .

#### 4.1 Implementación en Sistema M/M/1

En aras de revisar el funcionamiento de busque lineal y logarítmica, se propuso un sistema representado en una cola M/M/1, con dos y tres fases, el sistema implementado (Fig. 3), posee las siguientes constantes de control, un vector fila con las tasas de llegada para cada fase, así como un vector de tasas de atención, un número de niveles igual a 3.

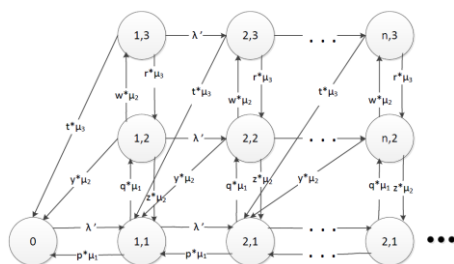


Fig. 3. Sistema M/M/1 propuesto. Fuente: Autores

Una vez se han calculado los bloques que conforman las matrices  $P$  y  $Q$ , podemos utilizar los algoritmos de iteración funcional (FI) y de reducción logarítmica (LR), con el fin de realizar los cálculos requeridos. Como ventaja del sistema encontramos la posibilidad de variar la tasa de ocupación del servidor  $\rho = \mu/\lambda$ , posibilitando el análisis de desempeño de los algoritmos generados.

#### 4.2 Implementación de un Sistema HE\_2/M/1

Se probaron los algoritmos utilizando un proceso con arribos y tiempos de servicio exponenciales. La cola incluirá dos fases de arribos representados por tasas  $\lambda_i$  con  $i = 1, 2$ , la relación lineal  $\alpha_1\lambda_1 + \alpha_2\lambda_2$  (tasa total de arribos) se mantendrá constante para las diferentes relaciones entre  $\lambda_1$  y  $\lambda_2$ .

El sistema que se analizará se presenta en la Fig. 1b. Las posibles transiciones y sus tasas se muestran en la Tabla 1. Los

bloques de la matriz generadora. Para la generación de los anteriores sistemas, se utilizaron algoritmos encargados de generar las matrices de transición y generadoras así como la evaluación de los algoritmos FI y LR.

Tabla 1. Resultados del análisis de la cola PH/PH/1, L es el valor esperado de usuarios en el sistema y W el tiempo esperado de espera. Fuente: Autores

| $\mu'$ | $\mu$ | V   | P   | Q   | L     | W     |
|--------|-------|-----|-----|-----|-------|-------|
| 0,4    | 0,6   | 0,3 | 0,6 | 0,4 | 3,27  | 11,46 |
| 0,4    | 0,6   | 0,8 | 0,6 | 0,4 | 8,36  | 10,97 |
| 0,4    | 0,6   | 1,5 | 0,6 | 0,4 | 17,09 | 11,99 |

5. RESULTADOS

5.1 Sistema PH/PH/1

Al analizar este sistema con un número relativamente alto de niveles se observó que en la distribución de estado estacionario la probabilidad de los últimos niveles tiende a cero (esto depende de la utilización que se haga del servidor), teniendo en cuenta esto se podría afirmar que los resultados son válidos para una cola con capacidad infinita, los resultados obtenidos son. Observamos que al aumentar parámetros como la tasa de arribos y la probabilidad de visitar la fase dos el número de usuarios en el sistema aumenta, mientras que aumentar la tasa de servicio disminuye este valor.

5.2 Algoritmos de Cálculo sobre una Cola M/M/1

Los tres algoritmos implementados mostraron respuestas iguales para valores de utilización del servidor baja, sin embargo difieren al aumentar esta, los tiempos de ejecución del sistema se presentan en la Fig. 4. Los criterios de evaluación que se utilizaron fueron el tiempo de ejecución del algoritmo y la máxima diferencia entre los resultados obtenidos con los algoritmos realizados (FI y LR) y el algoritmo basado en la matriz Q.

Los tiempos de respuesta de los algoritmos implementados varían según la tasa de utilización correspondiente Fig. 4. Aquí se

observa que el mejor tiempo de respuesta en todos los casos los presenta el algoritmo de iteración lineal modificado, evidentemente la matriz  $G$  converge menos rápido si la utilización tiende a uno. Una estrategia para solucionar este problema podría estar en aumentar el error permitido, sin embargo se detectó que el algoritmo lineal modificado no converge para definiciones de error baja.

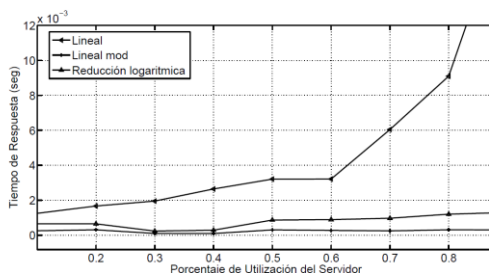


Fig. 4. Diferencias de tiempo de ejecución de los algoritmos

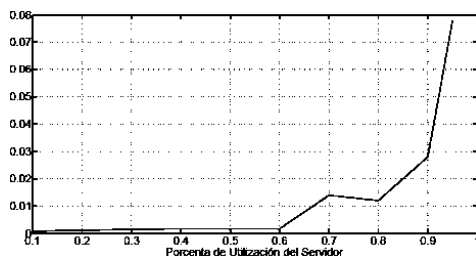


Fig. 5. Error en los algoritmos implementados

En cuanto a la diferencia entre los valores calculados a partir de la matriz generadora (en procesos finitos) y los bloques de a esta misma matriz (para QBDs), se observa que los algoritmos implementados presentan valores similares Fig. 5. También se observa que al aumentar la utilización del sistema el error aumenta. Al desarrollar los cálculos con diferentes valores de ocupación, notamos que el eigenvalor dominante de la matriz  $R$  tiende a la utilización del servidor, esto si las tasas de arribo  $\lambda_1$  y de servicio  $\mu_i$  son iguales para todo  $1 < i < m$ , lo cual implica que la utilización del servidor en este tipo de esquemas es similar a la tasa de crecimiento de los usuarios en el sistema (A, 1997).

### 5.3 Algoritmos de Cálculo sobre una Cola HE<sub>2</sub>/M/1

Al utilizar los algoritmos diseñados en este tipo de cola se vio que los algoritmos diseñados presentan un mayor error, también se observó que al aumentar la utilización del servidor el error se incrementa de manera importante, para este tipo de análisis al cambiar el factor  $\frac{\lambda_1}{\lambda_2}$  el tiempo de respuesta no se afectó significativamente.

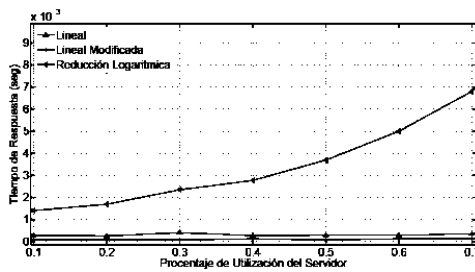


Fig. 6. Diferencias de tiempo de ejecución con tres fases.

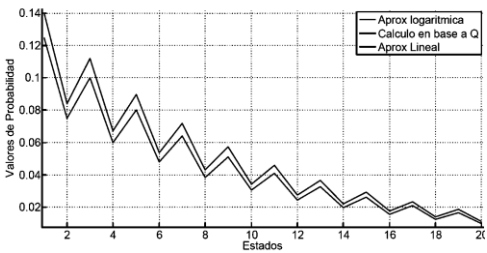


Fig. 7. Diferencias entre los valores de probabilidades estacionarias.

### 6. CONCLUSIONES

Los algoritmos diseñados funcionan de manera similar, únicamente el algoritmo de aproximación lineal modificado presenta diferencias dada su definición de error. Con los algoritmos diseñados no es necesario conocer la matriz generadora completamente, lo cual es importante en colas con capacidad infinita, en el trabajo

se compararon colas con un número de niveles relativamente altos (de 20 a 30) y los resultados se evidenciaron.

En una cola M/M/1 la relación entre tasas de arribo generó un aumento significativo en la diferencia de resultados entre los algoritmos, lo cual se puede explicar en la configuración de las matrices  $A_0$  y  $\widehat{A}_0$  de la matriz de transición y la matriz generadora respectivamente. Los sistemas lineales planteados no poseen solución para colas inestables, por lo cual el valor de la utilización del servidor evaluó la "dificultad" de solucionar el sistema, para bajos valores de utilización los algoritmos se comportaron de manera similar, esto debido al valor bajo del valor esperado de usuarios en el sistema.

## 7. REFERENCIAS

- Chakka, R. (2013). Spectral Expansion Method for QBD and QBD-M Processes in Performance Modeling of Computing and Communication Systems: A Review.
- Fu Yu, Z. Q. (2011). Variable Bandwidth Spectrum Access for Secondary User in Cognitive Radio Networks. National Natural Science Foundation of China, 1 - 5.
- Jiang, X. Gao, J. (2013). Delay Modeling for Broadcast-Based Two-Hop Relay MANETs. 2013 11th International Symposium and Workshops on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 1 - 7.
- Latouche, V. R. (2013). Matrix-Analytic Methods in Stochastic Models. New York: Springer.
- Gómez, J. C. (2011). Desarrollo de un módulo para modelar y resolver procesos de cuasi nacimiento y muerte como adición a la librería JMarkov.
- Martinez-Bauset, V. P. (2013). Approximate Analysis of Cognitive Radio Systems Using Time-Scale Separation and its Accuracy. IEEE COMMUNICATIONS LETTERS, 35-39.
- Suzán, H. (1997). Informe final del Proyecto G037 Análisis de viabilidad para poblaciones de la cactácea amenazada *Ariocarpus trigonus*.