Upper and Lower Bounds for Stochastic Marked Graphs*

Sergio Rajsbaum†

Instituto de Matemáticas– U.N.A.M.

D.F. 04510, México

Keywords. Stochastic marked graphs, petri nets, distributed computing, performance evaluation.

1 Introduction

Marked graphs (e.g. [7]) are a subclass of Petri nets, called also Decision Free Petri Nets or event graphs. They consist of a directed graph with a marking that associates tokens to the edges. Two types of events exist in a marked graph (MG): processing performed at vertices, and transmission delays of tokens along the edges. Duration of events can be expressed by a nonnegative real number or by a random variable (r.v.). In this paper the second method is used to study *stochastic* marked graphs (SMG's). It is assumed that the random variables are exponential, independent, and identically distributed (i.i.d.).

Adding the time factor to MG's enables performance evaluation of concurrent systems. For example, parallel computation models [16], distributed computing systems [15], manufacturing systems [6], tandem queueing networks [3], and distributed algorithms [8, 9, 11].

*A preliminary version of this paper, including monotonicity properties for general distributions, appears in [13].

†E-mail: rajsbaum at theory.lcs.mit.edu. On leave at the Laboratory for Computer Science, MIT, Cambridge, MA 02139.

The main performance measure considered is the *rate* of computation $R(v)$, i.e., the average number of computational steps (firings) of a vertex v, per time unit. It is known that for strongly connected graphs, $R(v)$ is the same for every vertex v (e.g. [2]). It was shown by Molloy [12] that the the rate can be computed by analyzing a Markov chain. However, this method of computing the rate is prohibitively inefficient because in general, the size of the Markov chain is very big. For example, the number of states of the chain that corresponds to a complete MG with one token on each edge is $2^{|V|} - 1$ [14].

In this paper a form of recurrence relations (e.g. [2], and [11]) is used to derive efficiently computable bounds on the rate of strongly connected, SMG's. These bounds depend on the degrees of the vertices and on the average number of tokens per edge in a cycle, but do not depend on the number of vertices itself. For example, for the case of regular δ degree graphs (either in-degree or out-degree), such that the average number of tokens on every cycle is a, $R(v) = \Theta(\frac{a}{\log \delta})$. The main result is that, for the case of bounded degree graphs, $R(v) = \Theta(\hat{a})$, where $\hat{a}$ is the minimum average number of tokens in a cycle.

The rate of a deterministic MG, i.e., one in which duration of events is equal to one, is $\hat{a}$ (e.g. [11]). Since it is known that this is the case of maximum rate (e.g. [13], [14]), Corollary 3.5 implies that the random event durations reduce asymptotically the rate by at most a factor of $1/\log \Delta$, where Δ is the maximum vertex degree, independently of the number of vertices of the MG.

This paper is a generalization of the results for exponential distributions of Rajsbaum and Sidi's paper [14] (see also [4] and [5]) on the performance of synchronizers [1] – methods to adapt a synchronous distributed algorithm to run on an asynchronous network. Baccelli and Konstantopoulos [3] independently derived upper bounds on the rate for MG's with arbitrary integrable i.i.d. processing times. Their results, although more general, use subadditive ergodic theory and multitype branching processes. Our proofs are simpler and make explicit the role of the parameters of the graph on the rate.

2 The Model

Let $G = (V, E)$ be a finite, directed and strongly connected graph $G = (V, E)$. A *marking* s is a function from E to the non-negative integers representing a state of the graph, where $s(e)$ is the number of *tokens* on edge e. A *marked graph* $MG = (G, s_0)$, consists of a graph G and an *initial marking*, s_0. A vertex v is *enabled* in s if $s(e) > 0$ for every edge $e = u \rightarrow v$ going into v. An enabled vertex v *fires* by consuming one token from each incoming edge and adding one token to each outgoing edge. We assume that MG is *deadlock-free,* i.e., that every vertex fires infinitely many times. This is equivalent to assuming that every cycle has at least one token in s_0 (e.g. [7]).

In a stochastic marked graph $s(e)$ represents the total number of tokens on edge e: the tokens traveling along e plus the tokens stored in a FIFO buffer at the end of e. The operation of firing of a vertex v starts at the first moment in which there is at least one token in the buffer of each ingoing edge to v. After some *processing time,* instantaneously, the first token from the buffer of each incoming edge of v is removed, and a token is sent along each outgoing edge of v. At this moment the firing terminates.

Assume that at time 0, the marking is given by s_0 and no tokens are in transit (all tokens are in buffers). Let us denote by $t_k(v)$, $k \geq 0$, the time on which v completes the k+1-st firing, and by $\tau_k(v)$ the corresponding processing time. Let $\delta_k(e)$ be the *transmission delay* of the token sent by v on e at $t_k(v)$. The processing times $\tau_k(v)$ and the delays $\delta_k(e)$, are i.i.d. exponential r.v.'s with mean λ^{-1}. Formally, a *stochastic marked graph*, $SMG = (MG, \tau, \delta)$, consists of a marked graph MG, together with the sequences of r.v.'s $\tau_k(v)$ and $\delta_k(e)$, $k \geq 0$, $v \in V$, $e \in E$.

Consider an edge $e = w \rightarrow v \in V$. Observe that v consumes the $s_0(e) + 1$-st first token from e only after having consumed all the $s_0(e)$ tokens initially in e. To fire for the next time, v has to wait for w to fire for the first time, and for the token to arrive to v. Thus, $t_{s_0(e)}(v) \geq t_0(w) + \delta_0(e)$. In general, to fire at $t_k(v)$, v waits for the token produced by u at time $t_{k-s_0(u \rightarrow v)}(u)$, for every u, s.t. $u \rightarrow v \in E$. When these tokens arrive, v starts the firing that will take $\tau_k(v)$ time. Namely, the evolution of the system can be described by the following recursions:

$$t_k(v) = \max_{e=u\to v\in E} \{t_{k-s_0(e)}(u) + \delta_{k-s_0(e)}(e)\} + \tau_k(v), \quad k \geq 0, v \in V. \tag{1}$$

To simplify the presentation, we make the inessential assumption that the transmission delays are negligible; it is not difficult to extend the results of this paper to the case of non-negligible delays. The recursions (1) become:

$$t_k(v) = \max_{e=u\to v\in E} \{t_{k-s_0(e)}(u)\} + \tau_k(v), \quad k \geq 0, v \in V. \tag{2}$$

To ensure that v does not start firing for the k-th time before completing the previous firing, we assume, for ease of notation, that there is an edge $v \to v$ for every vertex v, with one token, $s_0(v \to v) = 1$. Initially no tokens are in transit, hence, for $k < 0$, let $t_k(v) = 0$.

The explicit form of the recursions (2) has a simple graph theoretic representation. A path starting in u is *maximal*, if for every $w \to u \in E$, $s_0(w \to u) > 0$. For a vertex v, let $S_k(v)$ be the set of all maximal, directed paths ending in v, and having k tokens (in s_0). Note that, since every cycle has a positive number of tokens, every maximal directed path of k tokens is of finite length (number of edges). For instance, for $k = 0$, if every edge entering v has a positive number of tokens in the initial marking, then $S_0(v)$ includes only v itself. And for every k, $S_k(v)$ is not empty, since there is always a path of length k which uses only the loop $v \to v$.

For a path $P \in S_k(v)$, $P = v_0 \to v_1 \to \cdots \to v_l\ (= v)$, define a *prefix* of P as $P_i = v_0 \to v_1 \to \cdots \to v_i$, $i \leq l$, and let $s_0(P_i)$ be equal to the number of tokens in P_i. Now we define the r.v. $T(P) = \sum_{i=0}^{l} \tau_{s_0(P_i)}(v_i)$. Thus, $T(P)$ is the sum of $l+1$ r.v.'s, the first a $\tau_0(v_0)$ and the last a $\tau_k(v)$. Consider the set of r.v.'s $\{T(P) : P \in S_k(v)\}$. Note that the r.v.'s in this set are in general not independent. The graph theoretic interpretation of the firing times is given by the next theorem, which can be proved using the recursions (2).

Theorem 2.1 *For every $v \in V$, $k \geq 0$, $t_k(v) = \max\{T(P) : P \in S_k(v)\}$.*

The performance measures investigated in this paper are the firing times $t_k(v)$, $k \geq 0$, $v \in V$, and the related computation *rate of* v, $R(v)$, of a vertex v in G, defined by $R(v) = \lim_{k\to\infty} \frac{k}{t_k(v)}$.

3 Upper and Lower Bounds

The bounds presented here are a function of the following quantities. Denote by $d_{out}(v)$ ($d_{in}(v)$) the number of edges going out of (into) v (the original number of edges plus 1, for the loop added), and let $\Delta_{out} = \max_{v \in V} d_{out}(v)$, $\Delta_{in} = \max_{v \in V} d_{in}(v)$; $\delta_{out} = \min_{v \in V} d_{out}(v)$, $\delta_{in} = \min_{v \in V} d_{in}(v)$. For a directed cycle C of length l and $s_0(C)$ tokens, let $A(C) = s_0(C)/l$. Let $\hat{A} = \max\{A(C) : C \text{ is a cycle}\}$, $\hat{a} = \min\{A(C) : C \text{ is a cycle}\}$, $\hat{f} = \max\{s_0(P) : P \text{ is a simple path}\}$, and $\hat{s} = \max\{s_0(e) : e \in E\}$. The quantities $\hat{a}$ and $\hat{A}$, can be computed in time $O(|V| \cdot |E|)$ using an algorithm of Karp [10].

The following *decomposition* procedure is used in the sequel. Let $P = v_0 \to v_1 \to \cdots \to v_l$ be a path of G. If P is simple, nothing is done. Otherwise, remove a simple cycle from P as follows. Let $j \leq l$ be the least index such that for some $i < j$, $v_j = v_i$. Clearly, $C_1 = v_i \to v_{i+1} \to \cdots \to v_j$ is a simple cycle. Remove from P all the edges of C_1 to obtain a shorter path. Repeat this procedure until the path is simple, obtaining simple cycles $C_1, \ldots, C_k$, and a simple (possibly empty) path P'. Observe that using the decomposition of P we get that $s_0(P) \leq \hat{f} + l\hat{A}$.

Theorem 3.1 (Lower Bound)

(i) *For every $k \geq 0$ there exists a vertex v for which $E[t_k(v)] \geq \frac{k-\hat{s}-\hat{f}}{\lambda \hat{A}} \log \delta_{out}$.*

(ii) *For every $k \geq 0$ and every vertex v, $E[t_k(v)] \geq \frac{k-\hat{s}-\hat{f}}{\lambda \hat{A}} \log \delta_{in}$.*

Proof: We present a detailed proof of part (i); the proof of part (ii) is discussed at the end. Define a random walk $v_0 \to v_1 \to v_2 \to \cdots$ on G as follows. Let v_0 be any enabled vertex in the initial marking and consider any execution of the MG. Let v_1 be a vertex such that $v_0 \to v_1 \in E$, and $\tau_{s_0(v_0 \to v_1)}(v_1)$ in the execution was the largest processing time of a token sent by v_0 for the first time; call this r.v. $\bar{\tau}_1$, e.g., $\bar{\tau}_1 = \max_{e = v_0 \to v} \tau_{s_0(v_0 \to v)}(v)$. In general, assume that the random walk has been defined up to v_i, $i \geq 0$, and call it P_i. Let $s_0(P_i) = f_i$, i.e., the number of tokens in the random walk defined so far. Then v_{i+1} is a vertex such that $v_i \to v_{i+1}$, (possibly $v_{i+1} = v_i$) and on the given execution $\tau_{f_i + s_0(v_i \to v_{i+1})}(v_{i+1})$ was the largest processing time of a token sent by v_i at its step f_i:

$$\bar{\tau}_{i+1} = \max_{e=v_i \to v} \tau_{f_i+s_0(v_i \to v)}(v).$$

Hence, $f_{i+1} = s_0(P_{i+1}) = f_i + s_0(v_i \to v_{i+1})$. Since v_{i+1} will not start the f_{i+1}–th firing before v_i finishes the f_i–th firing, it follows that $t_{f_{i+1}}(v_{i+1}) \geq t_{f_i}(v_i) + \bar{\tau}_{i+1}$. The quantity $\bar{\tau}_{i+1}$ is equal to the maximum of at least δ_{out} independent and identically distributed r.v.'s with mean λ^{-1}. It is well known that the mean of c such r.v.'s is equal to $\sum_{j=1}^{c} 1/i \approx \lambda^{-1} \log c$ (natural logarithm). It follows that

$$\mathrm{E}[t_{f_{i+1}}(v_{i+1})] \geq \mathrm{E}[t_{f_i}(v_i)] + \lambda^{-1} \log \delta_{out},$$

and thus $\mathrm{E}[t_{f_{i+1}}(v_{i+1})] \geq \mathrm{E}[t_0(v_0)] + \lambda^{-1}(i+1) \log \delta_{out}$, where by Eq (2), $\mathrm{E}[t_0(v_0)] = \lambda^{-1}$. Hence, for $f_i < k \leq f_{i+1}$, we get

$$\mathrm{E}[t_k(v_i)] > \mathrm{E}[t_{f_i}(v_i)] \geq \lambda^{-1} + \lambda^{-1} i \log \delta_{out}. \tag{3}$$

Using the decomposition procedure, one can see that $f_i \leq i\hat{A} + \hat{f}$. Thus $\mathrm{E}[t_k(v_i)] \geq \lambda^{-1} + \lambda^{-1} \frac{f_i - \hat{f}}{\hat{A}} \log \delta_{out}$. Since $k - \hat{s} \leq f_i$ and $\lambda^{-1} > 0$,

$$\mathrm{E}[t_k(v_{i+1})] \geq \lambda^{-1} \frac{k - \hat{s} - \hat{f}}{\hat{A}} \log \delta_{out}.$$

This completes the proof of part (i). The proof of part (ii) evolves along similar lines, except that we start from v_i and move backward along the path. ■

The inequalities of the previous theorem can sometimes be improved for the case in which $\hat{A}$ is large enough, by considering a cycle C for which $A(C) = \hat{a}$, and a walk which goes around C; namely, $\mathrm{E}[t_k(v)] \geq \lambda^{-1} k/\hat{a}$. Therefore we have the following.

Corollary 3.2

$$R(v) \leq \lambda \min\{\frac{\hat{A}}{\log\, \max(\delta_{out}, \delta_{in})}, \hat{a}\}.$$

The following proposition (similar to p. 672 in Bertsekas and Tsitsiklis [5]), is used in the proofs of the upper bounds on the firing times.

Proposition 3.3 *Let $\{X_i\}$ be a sequence of independent exponential r.v.'s with mean λ^{-1}. For every positive integer k and any $c > 4\log 2$,*

$$\Pr\left(\sum_{i=1}^{k} X_i \geq \frac{ck}{\lambda}\right) \leq e^{-\frac{ck}{4}}.$$

Theorem 3.4 (Upper Bound)

(i) *For every $k > 0$, for every vertex v, $E[t_{k-1}(v)] \leq \frac{4}{\lambda}(1 + |V|\log\Delta_{in} + \frac{k}{\hat{a}}\log\Delta_{in})$.*

(ii) *For every $k > 0$ and every vertex v, $E[t_{k-1}(v)] \leq \log|V| + \frac{4}{\lambda}(1 + |V|\log\Delta_{out} + \frac{k}{\hat{a}}\log\Delta_{out})$.*

Proof: Again we restrict to the proof of part (i). Recall that Theorem 2.1 states that for every $v \in V$, $k > 0$, $t_{k-1}(v) = \max\{T(P) : P \in S_{k-1}(v)\}$. Also, for a path $P \in S_{k-1}(v)$, $T(P)$ is equal to the sum of l, $l = \text{length}(P)$, i.i.d. random variables. By Proposition 3.3,

$$\Pr\left(T(P) \geq \frac{cl}{\lambda}\log\Delta_{in}\right) \leq e^{-\frac{cl}{4}\log\Delta_{in}},$$

for every $c > 4$, since $\log 2/\log\Delta_{in} \leq 1$. Using the decomposition procedure, we have that l is equal to the length of a simple path plus the length of some simple cycles. By the definition of $\hat{a}$, and since a simple path has length at most $|V| - 1$, then $l \leq k/\hat{a} + |V| - 1 = K$. Now, there are at most Δ_{in}^K paths of length K ending in v. It follows that

$$\begin{aligned} \Pr\left(t_{k-1}(v) \geq \frac{cK}{\lambda}\log\Delta_{in}\right) &\leq \Delta_{in}^K e^{-\frac{cK}{4}\log\Delta_{in}} \\ &= e^{-K(\frac{c}{4}-1)\log\Delta_{in}}, \ c > 4. \end{aligned}$$

Letting $t = \frac{cK}{\lambda}\log\Delta_{in}$, $dt = \frac{K}{\lambda}\log\Delta_{in}dc$. To compute a bound on the expectation we use the previous inequality for $c > 4$; for $c \leq 4$, only the fact that the probability is at most 1:

$$\begin{aligned} \mathrm{E}[t_{k-1}(v)] &\leq \int_0^{\frac{4K}{\lambda}\log\Delta_{in}} 1dt + \int_4^{\infty} e^{-K(\frac{c}{4}-1)\log\Delta_{in}}\frac{K}{\lambda}\log\Delta_{in}dc \\ &= \frac{4K}{\lambda}\log\Delta_{in} + \frac{4}{\lambda}. \end{aligned}$$

Namely,

$$\mathrm{E}[t_{k-1}(v)] \leq \frac{4k}{\lambda\hat{a}}\log\Delta_{in} + \frac{4(|V|-1)}{\lambda}\log\Delta_{in} + \frac{4}{\lambda}.$$

■

Corollary 3.5

$$R(v) \geq \frac{\lambda \hat{a}}{4 \log \min(\Delta_{out}, \Delta_{in})}.$$

Consider the meaning of the previous results. For regular in– or out–degree δ graphs, for which $\hat{a} = \hat{A}$, the bounds are tight up to a constant factor of 1/4:

Corollary 3.6 *For regular in–degree or out–degree δ graphs, for which $\hat{a} = \hat{A}$, $R(v) = \Theta\left(\frac{\lambda \hat{a}}{\log \delta}\right)$.*

The case of bounded degree graphs is of particular interest, because it is practically infeasible to construct networks with vertex degrees that grow as $|V|$ grows. In this case, $R(v) = \Omega(\lambda \hat{a})$, by Corollary 3.5. Also, $R(v) = O(\lambda \hat{a})$, by Corollary 3.2. Therefore, even if $\hat{A} > \hat{a}$, for bounded degree graphs the bounds are asymptotically tight (up to a constant factor of 1/4 of the logarithm of the bound on the degrees):

Corollary 3.7 *For bounded degree graphs, $R(v) = \Theta(\lambda \hat{a})$.*

References

[1] B. Awerbuch, "Complexity of Network Synchronization," *Journal of the ACM,* vol. 32, no. 4, pp. 804–823, Oct. 1985.

[2] F. Baccelli, "Ergodic Theory of Stochastic Petri Networks," Rapport INRIA No. 1037, 1989.

[3] F. Baccelli, P. Konstantopoulos, "Estimates of Cycle Times in Stochastic Petri Nets," in I. Karatzas, D. Ocone (Eds.), *Proc. of a US-French Workshop on Applied Stochastic Analysis,* Rutgers University, vol. 177 of *Lecture Notes in Control and Information Theory,* Springer Verlag, 1991.

[4] P. Berman, J. Simon, "Investigations of Fault–Tolerant Networks of Computers," *Proc. of the 20th ACM Symposium on the Theory of Computation,* pp. 66–77, 1988.

[5] D.P. Bertsekas and J.N. Tsitsiklis, *Parallel and Distributed Computation,* Prentice–Hall, N.J. 1989.

[6] G. Cohen, D. Dubois, J.P. Quadrat, M. Viot, "A Linear-System-Theoretic View of Discrete-Event Processes and Its Use for Performance Evaluation in Manufacturing," *IEEE Trans. on Automatic Control,* vol. AC-30, no. 3, pp. 210–220, March 1985.

[7] F. Commoner, A.W. Holt, S. Even, A. Pnueli, "Marked Directed Graphs," *Journal of Computer and Systems Sciences*, vol. 5, pp. 511-523, 1971.

[8] S. Even and S. Rajsbaum, "The Use of a Synchronizer Yields Maximum Rate in Distributed Networks," *Proc. of the 22nd ACM Symposium on the Theory of Computation,* pp. 95–105, May 1990.

[9] S. Even and S. Rajsbaum, "Unison, Canon, and Sluggish Clocks in Networks Controlled by a Synchronizer." To appear in *Mathematical Systems Theory.*

[10] R.M. Karp, "A characterization of the minimum cycle mean in a digraph," *Disc. Math.,* vol. 23, pp. 309–311, 1978.

[11] Y. Malka, S. Rajsbaum, "Analysis of Distributed Algorithms based on Recurrence Relations," in S. Toueg, P. Spirakis, Kirousis (Eds.), *Proc. 5th Int. Workshop on Distributed Algorithms,* Greece, vol. 579 of *Lecture Notes in Computer Science,* pp. 242–253, Springer Verlag, Oct. 1991.

[12] M.K. Molloy, "Performance analysis using stochastic petri nets," *IEEE Trans. on Computers*, vol. c–31, no. 9, pp. 913–917, 1982.

[13] S. Rajsbaum, "Stochastic Marked Graphs," *Proc. 5th Int. Workshop on Petri Nets and Performance Modeling,* Melbourne, pp. 95–101, IEEE Computer Society, 1991.

[14] S. Rajsbaum and M. Sidi, "On the Performance of Synchronized Programs in Distributed Networks with Random Processing Times and Transmission Delays." To appear in *IEEE Trans. on Parallel and Distributed Systems.*

[15] C.V. Ramamoorthy and G.S. Ho, "Performance evaluation of asynchronous concurrent systems using Petri nets," *IEEE Trans. Software Eng.*, vol. SE–6, no. 5, pp. 440–449, 1980.

[16] R. Reiter, "Scheduling Parallel Computations," *Journal of the ACM,* vol. 15, pp. 590–599, 1968.