Stochastic Marked Graphs

Sergio Rajsbaum*
Instituto de Matemáticas
U.N.A.M.
D.F. 04510, México

Abstract

Stochastic marked graphs (SMG's) are marked graphs in which transmission delays of tokens, and firing durations are random variables. A technique is presented to study the performance of SMG's. The main performance measure is the rate of computation, i.e., the average number of firings of a vertex, per time unit.

The effect of the topology and the probability of the random variables on the rate is investigated. For deterministic random variables, the rate is maximized, while for exponential random variables the rate is minimized (among a natural class of distributions).

For random variables with exponential distribution several bounds on the rate are provided. The bounds depend on the degrees of the vertices and on the average number of tokens in a cycle, but not on the number of vertices itself. In particular, it is shown that the rate is at least the optimal (deterministic) rate, divided by a logarithmic factor of the vertex degrees. Thus, for some graphs the rate does not diminishes below a bound, regardless of the number of vertices.

1 Introduction

Diverse graph structure models for concurrent systems have been suggested and used. The structures differ in generality and scope according to the properties one wishes to model and analyze. In this paper the simple model of marked graphs (e.g. Commoner et. al. [6], Reisig [15]) is considered. Marked graphs are a special case of the more powerful model of Petri nets. They consist of a directed graph, with a marking which associates tokens to the edges.

Adding the time factor to marked graphs enables performance evaluation of concurrent systems. Time of operation can be expressed by a nonnegative real number (e.g. [8], [12], [17] [19]) or by a random variable (e.g. [9] , [10], [11]). In this paper the second method is used to study *stochastic* marked graphs (SMG's).

Two types of events exist in an SMG. The first type associates time with the processing performed at vertices (a vertex plays the role of a transition in a Petri net). The second type of events associates time with transmission delays of the edges. The results of this paper apply to SMG's with both types of events. For the sake of clarity this presentation is for the case of negligible transmission delays. However, it is easy to extend the results for the general case.

The paper is devoted to the performance analysis of strongly connected[1], directed stochastic marked graphs. The main performance measure is the *rate* of computation $R(v)$, i.e., the average number of computational steps (firings) of a vertex v, per time unit. A technique is presented to analyze and to compare the performance of different stochastic marked graphs. The technique clarifies the difference between processing times and transmission delays. This paper is a generalization of Rajsbaum and Sidi's paper [18] on the performance of synchronizers Awerbuch [1] – methods to adapt a synchronous distributed algorithm to run on an asynchronous network.

Section 2 contains the model, and the technique (a certain type of recurrence relations) used to analyze a SMG.

In Section 3 the case of random variables with general probability distributions is studied. First the effect of the topology on the rate is analyzed. Stochastic comparison techniques (e.g. [20]) are used to compare the rate of stochastic marked graphs with different topologies. Then systems with the same topology but different distributions of the random variables are analyzed. It is shown, using a partial order on the set of distributions, that determinism maximizes the rate of computation. On the other hand, exponentially distributed random variables minimize the rate, among a large class of distributions. Similar studies for acyclic networks and fork-join queues have been performed by several authors, e.g. Baccelli et. al. [3], and Pekergin et. al. [13].

The case of identical, exponential random variables is considered in Section 4. Several bounds on the rate are provided that 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, $R(v) = \Theta(\frac{a}{\log \delta})$.[2]

The main result is that, for the case of bounded degree graphs, $R(v) = \Theta(\hat{a})$, where $\hat{a}$ is the minimum

*E-mail: rajsbaum at unamvm1.bitnet

[1] a directed graph is strongly connected if there is a path in the graph between every pair of vertices.

[2] A function $f(n) = \Theta(g(n))$ if there exist constants c_1, $c_2 > 0$, and n_0, such that for every $n \geq n_0$, $c_1 g(n) \leq f(n) \leq c_2 g(n)$.

TH0386-3/91/0000/0095/$01.00 © 1991 IEEE

average number of tokens in a cycle.

The rate of a deterministic marked graph, i.e., one in which duration of events is a real number, is equal to $\hat{a}$ (e.g. [14], [17]). Since it is shown here that this is the case of maximum rate, Corollary 4.5 implies that random event durations reduce the rate by at most a factor of $1/\log\Delta$, where Δ is the maximum vertex degree.

The model of Rajsbaum and Sidi [18] is essentially that of a stochastic marked graph with one token per edge. In Berman and Simon [4], and Bertsekas and Tsitsiklis [5], a model similar to the one in [18] is considered, and some of the results for exponential distributions are obtained. The class of stochastic marked graphs with exponentially distributed random variables belongs to the more general case of stochastic petri nets (see Marsan [9] for a survey), where it is often assumed that the state space is given, which in our case is of exponential size in the number of vertices. Asymptotic performance of stochastic marked graphs as the number of tokens grows (but the graph remains fixed) is studied by Molloy [11].

2 The Model

Stochastic Marked Graphs

A *marked graph* $MG = (G, s_0)$, consists of a finite, directed and strongly connected graph $G = (V, E)$, and an *initial marking*, s_0. A *marking* s is a function from E to the non-negative integers $\mathcal{N}$ representing a state of the marked graph, where $s(e)$ is the number of *tokens* on edge e. A vertex v is *enabled* 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 incomming edge and adding one token to each outgoing edge. Thus, when a vertex v fires the marking changes to another marking s', such that for every edge $e = x \rightarrow y$,

$$s'(e) = \begin{cases} s(e) & \text{if } x = y = v, \text{ else} \\ s(e) - 1 & \text{if } y = v, \text{ else} \\ s(e) + 1 & \text{if } x = v, \text{ and} \\ s(e) & \text{otherwise.} \end{cases}$$

The operation of firing is atomic in that the tokens are removed simultaneously from the corresponding edges, and then are added simultaneously to the corresponding edges. But in a marked graph, it is not specified when the vertices fire, nor any other timing information.

In a stochastic marked graph there are two type of events that have durations: *delays* and *processing times.* Informaly, the delay is the time it takes a token to travel along an edge, and the processing time is the time it takes a vertex to fire (from the instant it removes tokens to the instant it adds tokens). In an SMG, a vertex fires as soon as it is enabled, provided it is has completed the previous firing. We say that v is *fireable* if v is enabled and has completed the previous firing. Observe that tokens may arrive on an edge $u \rightarrow v$ before v is ready to fire and consumme them, either because on another edge entering v there are no tokens, or because v is busy firing. Therefore edges are assumed to have storage buffers. The following simple proposition is well known.

Proposition 2.1 *The number of tokens in a cycle does not change by vertex firing.*

It follows from this proposition that the buffers needed are of bounded size (at most $|V|$). Note that for a marking s, $s(e)$ represents the total number of tokens on edge e: the tokens traveling along e plus the tokens stored in the buffer of e.

Assume that at time 0, the number of tokens in the buffers is given by s_0. Let us denote by $t_k(v)$, $k \geq 0$, the time on which v fires for the $k+1$-th time, and by $\tau_k(v)$ the corresponding processing time. Let us denote by M_k the tokens that are sent on time $t_k(v)$, and by $\delta_k(e)$ the delay of M_k on e. The processing times $\tau_k(v)$ and the delays $\delta_k(e)$, are positive, real-valued random variables defined over some probability space. Formally, a *stochastic marked graph*, $SMG = (MG, \tau, \delta)$, consists of a marked graph MG, together with the sequences of random variables $\tau_k(v)$ and $\delta_k(e)$, $k \geq 0$, $v \in V$, $e \in E$.

We say that SMG is *deadlock-free* if for every $v \in V$, the times $t_k(v)$, $k \geq 0$, are finite. We shall assume that SMG is deadlock-free. Another well known result is the following.

Proposition 2.2 *SMG is deadlock-free if and only if every cycle C has a positive number of tokens, $s_0(C) > 0$.*

Let us assume, for ease of notation, that there is a loop $v \rightarrow v$ on every vertex v, with one token, $s_0(v \rightarrow v) = 1$, and $\delta_k(v \rightarrow v) = 0$, $k \geq 0$. Also, for $k < 0$, let $\delta_k(e) = 0$, and $t_k(v) = 0$, and for $e \notin E$, let $\delta_k(e) = -\infty$. The behavior of SMG will not be affected by these assumptions.

Consider an edge $e = w \rightarrow v \in V$. Observe that v consumes the first token sent by w, only after having consumed all the $i = s_0(e)$ tokens initially in e. To fire for the $i+1$-th time, v has to wait for w to fire for the first time, and for the token to arrive to v. Thus, $t_{i+1}(v) \geq t_0(w) + \delta_0(e)$. In general, the token v consumes on e at time $t_k(v)$ is the one produced by w at time $t_{k-s_0(e)}$. It follows that the evolution of the system can be described by the following recursions:

$$t_k(v) = \max_{e=u\rightarrow v} \{t_{k-s_0(e)}(u) + \delta_{k-s_0(e)}(e)\} + \tau_k(v), \quad (1)$$

for $k \geq 0$, $v \in V$. Note that the assumption that there is a loop on every vertex ensures that v does not start firing for the k-th time before completing the previous firing. To simplify the presentation, we make the inessential assumption that the 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:

$$t_k(v) = \max_{e=u\rightarrow v} \{t_{k-s_0(e)}(u)\} + \tau_k(v), \quad (2)$$

for $k \geq 0$, $v \in V$.

It is interesting to note that the firing times $t_k(v)$ have a simple graph theoretic interpretation. For a vertex v, let $S_k(v)$ be the set of all maximal, directed paths of k tokens ending in v[3]. Note that, since SMG is deadlock-free, by Proposition 2.2, every maximal directed path is of finite length. 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, $S_k(v)$ 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 an initial part 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 random variable $T(P)$:

$$T(P) = \sum_{i=0}^{l} \tau_{s_0(P_i)}(v_i)$$

and the set of random variables $T(S_k(v)) = \{T(P) : P \in S_k(v)\}$. Thus, $T(P)$ is the sum of $l+1$ random variables, the first a $\tau_0(v_0)$ and the last a $\tau_k(v)$. Note that the random variables in $T(S_k(v))$ are in general not independent, even if the $\tau_i(v)$'s are independent. The graph theoretic interpretation of the firing times is given by the next theorem.

Theorem 2.3 *For every $v \in V$, $k \geq 0$, $t_k(v) = \max\{T(P) : P \in S_k(v)\}$.*

Proof: Define an order $<$ on the vertices of G as follows. If there is a token free path from u to v, then $u < v$. This defines a partial order on V, since the token free subgraph of MG is acyclic. Complete the partial order to a total order in an arbitrary way.

Let v_0 be the smallest element with respect to $<$. One can check that $t_0(v_0) = \tau_0(v_0)$.

Assume the theorem holds for every vertex up to $k-1$, and for vertices smaller than v, up to k. Then, by the recursions 2,

$$t_k(v) = \max_{e=u\to v} \{t_{k-s_0(e)}(u)\} + \tau_k(v).$$

By induction hypothesis,

$$\begin{aligned} t_k(v) &= \max_{e=u\to v} \{\max T(S_{k-s_0(e)}(u)) : k - s_0(e) \geq 0\} \\ &\quad + \tau_k(v), \end{aligned}$$

and leaving only one max,

$$t_k(v) = \max T(S_k(v)).$$

■

[3] A path P has k tokens if the sum of $s_0(e)$ over all edges e in P is equal to k. The path is also maximal, if there is no longer path ending in v, containing P and with the same number of tokens. The length of a path is the number of edges in the path.

The Performance Measures

The most important performance measures investigated in this paper are the firing times $t_k(v)$, $k \geq 0$, $v \in V$. A related performance measure of interest is the counting process $N_t(v)$, associated with v, defined by

$$N_t(v) = \sup\{k : t_k(v) \leq t\}.$$

That is, $N_t(v)$ is the number of firings (minus 1) completed by v up to time t. Similarly, $N_t = \sum_{v \in V} N_t(v)$ denotes the total number of firings (minus $|V|$) performed in the system up to time t. The following proposition is about synchronic distances. It indicates that no vertex can advance, in terms of firing times, too far ahead of any other vertex. Let $\hat{s}$ be equal to the maximum number of tokens in an edge in the initial marking, and let d be the diameter[4] of G.

Proposition 2.4 *For all $u, v \in V$, and $t \geq 0$, $|N_t(u) - N_t(v)| \leq d\hat{s}$.*

Proof: Denote by l the length of a simple path from u to v. A simple inductive argument on l, using the recursions 2 shows that if the last message sent by u up to time t is M_k, $k = N_t(u)$, then that $N_t(v) \leq k + l\hat{s}$. Thus, $N_t(v) - N_t(u) \leq l\hat{s} \leq d\hat{s}$. The same argument for a simple path from v to u proves that $N_t(u) - N_t(v) \leq d\hat{s}$. ■

Another important performance measure is the computation *rate of* v, $R(v)$, of a vertex v in G, defined by

$$R(v) = \lim_{t\to\infty} \frac{N_t(v)}{t},$$

whenever the limit exists. Note that if the limit exists, then $R(v)$ is a number, not a random variable. Similarly, the computation *rate of the network* is defined by

$$R = \lim_{t\to\infty} \frac{N_t}{t}.$$

Proposition 2.4 implies that for every $u, v \in V$, $R(u) = R(v)$, and therefore, $R = |V|R(v)$.

3 General Probability Distributions

In this section results about networks with general distributions of the processing times $\tau_k(v)$ are presented.

3.1 Topology

It is shown here that adding edges to an SMG with an arbitrary topology slows down the operation of each of the processors, regardless of the number of tokens in the edges added. The basic methodology used is the *sample path* comparision; that is, comparing the evolution of message transmissions in different SMG's for every instance, or realization, of the random variables $\tau_k(v)$. This yields a stochastic ordering (e.g. [16], [20]) between various SMG's.

[4] The diameter is the maximum distance between a pair of vertices in the graph.

Theorem 3.1 *Let SMG' be the stochastic marked graph obtained from SMG by adding to it a set of edges E' with an arbitrary number of tokens. For every realization of the random variables $\tau_k(v)$, $k \geq 0$, $v \in V$, the following inequalities hold*

$$t_k(v) \leq t'_k(v),$$

for all $k \geq 0$, $v \in V$, where $t'_k(v)$ are the firing times in SMG'.

Proof: The proof follows from theorem 2.3, since $S_k(v)$ is contained in $S'_k(v)$, the set of maximal paths ending in v, in SMG'. ■

Theorem 3.1 implies immediately the following theorem, where primed quantities are with respect to SMG'.

Corollary 3.2 *Under the conditions of Theorem 3.1, we have that $N_t(v) \geq N'_t(v)$ and $R(v) \geq R'(v)$ for all $v \in V$. Also $N_t \geq N'_t$ and $R \geq R'$. (When the limits exist).*

Notice that no assumption was made about the random variables $\tau_k(v)$. In particular, they need not be independent.

The sample path proof above implies that the random variable N_t is stochastically larger than the random variable N'_t, denoted $N_t \geq_d N'_t$, namely, $\Pr\{N_t \geq \alpha\} \geq \Pr\{N'_t \geq \alpha\}$, for all α. The corollary implies that if one starts with a simple directed cycle (a strongly connected graph with the least number of edges) and successively adds edges, a complete graph is obtained, without ever increasing the rate.

3.2 Probability Distributions

Now we compare networks SMG and SMG' having the same topology (V, E), but operate with different distributions of the processing times $\tau_k(v)$, $\tau'_k(v)$. To that end, we assume that in both networks the processing times $\tau_k(v)$, $k \geq 0$, $v \in V$ are independent and have finite mean $\mathrm{E}[\tau_k(v)] = \lambda_k^{-1}(v)$. In the case that $\lambda_k(v) = \lambda(v)$, we say that $\lambda(v)$ is the *potential rate* of v, as this would be the rate of v if it would not have to wait for tokens from its neighbors. The processing times in SMG' are distributed as in SMG, except for some of them, which may have another distribution.

Recall that a function h is *convex* if for all $0 < t < 1$, x_1, x_2, it holds $h(tx_1 + (1-t)x_2) \leq th(x_1) + (1-t)h(x_2)$. A random variable X with distribution F_X is said to be *more variable* than a random variable Y with distribution F_Y, denoted $X \geq_c Y$, or $F_X \geq_c F_Y$, if $\mathrm{E}[h(X)] \geq \mathrm{E}[h(Y)]$ for all increasing convex functions h. The partial order $\geq_c$ is called *convex order* (e.g. [16], [20]). Intuitively, X is more variable than Y if F_X gives more weight to the extreme values than F_Y; for instance, if $\mathrm{E}[X] = \mathrm{E}[Y]$, then $\mathrm{Var}(X) \geq \mathrm{Var}(Y)$, since $h(x) = x^2$ is an increasing convex function.

Assume that SMG and SMG' have the same arbitrary topology, but some of the processing times in SMG are more variable than the corresponding processing times in SMG', namely, for some v's, $\tau_k(v) \geq_c \tau'_k(v)$, while all other processing times have the same distributions in both graphs. When all processing times in SMG (SMG') are independent of each other, the following holds.

Theorem 3.3 *Under the above conditions the following holds for all processors v and $k \geq 0$*

$$t'_k(v) \leq_c t_k(v).$$

Proof: From Theorem 2.3 we have that $t_k(v) = \max\{T(P) : P \in S_k(v)\}$, where $P = v_0 \to v_1 \to \cdots \to v_l (= v)$ is a maximal directed path of k tokens ending in v. From the fact that the processing times are positive and max and $\sum$ are convex increasing functions, it follows that $t_k(v)$ is a convex increasing function of its arguments $\tau_i(u)$. Thus, we can use Proposition 8.5.4 in [16]:

Proposition 3.4 (Ross) *If $X_1, X_2, \ldots, X_n$ are independent random variables, $Y_1, Y_2, \ldots, Y_n$, are independent random variables, and $X_i \geq_c Y_i$, $i = 1, 2, \ldots, n$, then $g(X_1, X_2, \ldots, X_n) \geq_c g(Y_1, Y_2, \ldots, Y_n)$ for all increasing convex functions g which are convex in each of its arguments.*

The proof of the theorem now follows since by assumption the processing times in SMG are independent, the processing times in SMG' are independent, and $\tau'_k(v) \leq_c \tau_k(v)$. Note that the random variables $T(P)$ are not independent. ■

Corollary 3.5 *Under the above conditions, $N_t(v) \leq_c N'_t(v)$, $N_t \leq_c N'_t$, $R(v) \leq R'(v)$, and $R \leq R'$.*

Assume that the expected time until a processor finishes a processing step given that it has already been working on that step fop α time units is less or equal to the original expected processing time for that step. Namely, we assume that the distributions of the processing times are *new better than used in expectation* (NBUE) (e.g. [16] , [20]), so that if τ is a processing time, then for all $a \geq 0$

$$\mathrm{E}[\tau - a | \tau > a] \leq \mathrm{E}[\tau].$$

Consider three stochastic marked graphs with the same topology SMG, $SMG^{(d)}$ and $SMG^{(e)}$. The processing times of SMG are independent with any NBUE distribution. The processing times of $SMG^{(e)}$ have the same mean as in SMG, but are independent, exponentially distributed. The processing times of $SMG^{(d)}$ have the same mean as in SMG, but are deterministic[5]. The next theorem follows from Theorem 3.3, and from the fact that the deterministic is the minimum, while the exponential is the maximum among all NBUE distributions with respect to the convex ordering [16], [20] .

Theorem 3.6 *For every vertex v, and $k \geq 0$, $t_k^{(d)}(v) \leq_c t_k(v) \leq_c t_k^{(e)}(v)$.*

[5] A random variable is deterministic if it is always equal to some constant.

Some examples of distributions which are less variable than the exponential (with appropriate parameters) are the Gamma, Weibull, Uniform and truncated (in zero) Normal.

With respect to comparing the rate of a SMG with the rate of a SMG with deterministic processing times, we can prove a stronger result. Namely, the probability distributions of the processing times of SMG need not be NBUE; they can be of an arbitrary distribution, as long as they are independent and have finite mean. The proof of the next theorem is by induction on k, using Jensen's inequality.

Theorem 3.7 *Under the above conditions, $t_k^{(d)}(v) \le E[t_k(v)]$, for all vertices v, and $k \ge 0$. The expectation is taken over the respective distributions of processing times in SMG.*

When all processing times are deterministic, the computation of the rate is no longer a stochastic problem, but a combinatorial one. Thus, one conclusion of the last theorem is that in this case, the rate of $SMG^{(d)}$, obtained via combinatorial techniques ([7] ,[14], [17], [19]), yields an upper bound on the average rate of SMG. Furthermore, if the times $t_k^{(d)}(v)$ are computed, they give a lower bound on $E[t_k(v)]$. In the next section we concentrate on the other extreme case: exponentially distributed processing times. Therefore, the results will provide lower bounds on the rate of an SMG with arbitrary NBUE processing times.

4 Exponential Distributions

In this section we assume that the processing times $\tau_k(v)$, $v \in V$, $k \ge 0$, are independent and identically distributed exponential random variables with mean λ^{-1}. In this case, the SMG is a markov chain. A state of the markov chain at time t is a marking of the marked graph, and it is specified by the number of tokens stored in the buffer of each edge. Thus, a vertex with a positive number of tokens on each of its incomming edges in a state of the chain, is enabled and in a processing time. The number of states is finite, by Proposition 2.1. Moreover, the markov chain is irreducible, because the network is strongly connected (see Commoner et. al. [6]). Therefore, the limiting probabilities exist, they are all positive and their sum is equal to 1. Now, using the limiting probabilities, the percent of time that a vertex v is enabled, $P(v)$, is computed by summing the probabilities of all states in which the vertex is enabled. It follows that the rate $R(v) = \lambda P(v)$ exists and is positive [10]. However, the problem in using the markov chain to compute the rate is that the number of states is exponential. For example, the number of states of a complete graph with one token on each edge in s_0 is $2^{|V|} - 1$ [18].

We consider general topologies and derive upper and lower bounds on the rate of computation. These bounds depend on the degrees of the vertices and on the average number of tokens per edge in a cycle. But the bounds do not depend on the number of vertices itself. For bounded degree (either in-degree or out-degree) graphs, the bounds are tight (up to a small constant), and provide a characterization of which SMG's have a bounded rate independent of their number of vertices.

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

$$\Delta_{out} = \max_{v\in V} d_{out}(v), \quad \Delta_{in} = \max_{v\in V} d_{in}(v);$$

$$\delta_{out} = \min_{v\in V} d_{out}(v), \quad \delta_{out} = \min_{v\in V} d_{in}(v).$$

For a directed cycle C of length l and $s_0(C)$ tokens, let $A(C) = s_0(C)/l$. Let $\hat{A} = \max\{A(C) : C \text{ is a cycle}\}$, $\hat{a} = \min\{A(C) : C \text{ is a cycle}\}$ and $\hat{f} = \max\{s_0(P) : P \text{ is a simple path}\}$. Recall that $\hat{s} = \max\{s_0(e) : e \in E\}$.

The following *decomposition* procedure is used in the sequel. Let $P = v_0 \to v_1 \to \cdots \to v_n$ be a path of G. If P is simple, nothing is done. Otherwise, remove a simple cycle from P as follows. Let $j \le n$ be the least index such that $v_j = v_i$, $i < j$. 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 $C_2, \ldots, C_k$, and a simple (possibly empty) path P'. Observe that using the decomposition of P we get that $s_0(P) \le \hat{f} + n\hat{A}$.

Theorem 4.1 (Lower Bound)

(i) *For every $k \ge 0$ there exists a vertex v for which*

$$E[t_k(v)] \ge \frac{k - \hat{s} - \hat{f}}{\lambda \hat{A}} \log \delta_{out}.$$

(ii) *For every $k \ge 0$ and every vertex v,*

$$E[t_k(v)] \ge \frac{k - \hat{s} - \hat{f}}{\lambda \hat{A}} \log \delta_{in}.$$

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. Let v_1 be a vertex such that $v_0 \to v_1 \in E$, and

$$\tau_{s_0(v_0\to v_1)}(v_1) = \max_{e=v_0\to v} \tau_{s_0(v_0\to v)}(v).$$

In general, assume that the random walk has been defined up to v_i, $i \ge 0$, and call it P_i. Let $s_0(P_i) = f_i = f$, i.e., f is the number of tokens (in s_0) in the random walk defined so far. Then v_{i+1} is a vertex such that $v_i \to v_{i+1}$, (recall that we assume that every vertex has a loop) and

$$\tau_{f+s_0(v_i\to v_{i+1})}(v_{i+1}) = \max_{e=v_i\to v} \tau_{f+s_0(v_i\to v)}(v).$$

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 $t_{f_{i+1}}(v_{i+1}) - t_{f_i}(v_i) \geq \tau_{f_{i+1}}(v_{i+1})$. The quantity $\tau_{f_{i+1}}(v_{i+1})$ is equal to the maximum of at least δ_{out} independent and identically distributed random variables with mean λ^{-1}. It is well known that the mean of c such random variables is equal to $\sum_{j=1}^{c} 1/i \approx \lambda^{-1} \log c$[6]. It follows that

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

and thus

$$E[t_{f_{i+1}}(v_{i+1})] \geq E[t_0(v_0)] + \lambda^{-1}(i+1) \log \delta_{out},$$

where $E[t_0(v_0)] = \lambda^{-1}$. Using the decomposition procedure, one can see that the random walk P_i consists of an initial simple path (possibly empty), followed by a sequence of cycles. The number of tokens in the simple path is at most $\hat{f}$, and thus the number of tokens in P_i satisfies $f_i \leq i\hat{A} + \hat{f}$. Now, observe that $f_{i+1} \leq f_i + \hat{s}$, for $i \geq 0$. Hence, for $f_i < k \leq f_{i+1}$, we get

$$\begin{aligned} E[t_k(v_i)] &> E[t_{f_i}(v_i)] \\ &\geq \lambda^{-1} + \lambda^{-1} i \log \delta_{out}. \end{aligned}$$

Since $f_i \leq i\hat{A} + \hat{f}$, then

$$E[t_k(v_i)] \geq \lambda^{-1} \frac{f_i - \hat{f}}{\hat{A}} \log \delta_{out},$$

and since $k - \hat{s} \leq f_i$,

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

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, $E[t_k(v)] \geq \lambda^{-1} k/\hat{a}$. Therefore we have the following.

Corollary 4.2

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

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

Proposition 4.3 *Let $\{X_i\}$ be a sequence of independent exponential random variables with mean λ^{-1}. For every positive integer k and any $c > 4\log 2$,*

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

[6] **Natural logarithm.**

Theorem 4.4 (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,*

$$\begin{aligned} E[t_{k-1}(v)] \leq\ & \log |V| + \\ & \frac{4}{\lambda}(1 + |V| \log \Delta_{out} + \frac{k}{\hat{a}} \log \Delta_{out}). \end{aligned}$$

Proof: Again we restrict to the proof of part (i). Recall that Theorem 2.3 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, $l = \text{length}(P)$, independent and identically distributed random variables. By Proposition 4.3,

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

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

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

for every $c > 4$. Letting $t = \frac{cK}{\lambda} \log \Delta_{in}$, $dt = \frac{K}{\lambda} \log \Delta_{in} dc$. Hence,

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

Namely,

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

■

Corollary 4.5

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

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 4.6 *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 unfeasible to construct networks with vertex degrees that grow as $|V|$ grows. In this case, $R(v) = \Omega(\lambda \hat{a})$, by Corollary 4.5. Also, $R(v) = O(\lambda \hat{a})$, by Corollary 4.2. 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 4.7 (Main Result) *For bounded degree graphs,*

$$R(v) = \Theta(\lambda \hat{a}).$$

Acknowledgment

The author would like to thank two anonymous referees for their valuable comments.

References

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

[2] O.J. Boxma, "Sojourn Times in Cyclic Queues-The Influence of the Slowest Server," in *Computer Performance and Reliability,* G. Iazeolla, P.J. Courtois, O.J. Boxma (Ed.), Elsevier Science Publ. (North-Holland), 1988.

[3] F. Baccelli, W.A. Massey, A. Towsley, "Acyclic Fork-Join Queueing Networks," *Journal of the ACM*, vol. 36, no. 3, July 1989, pp. 615–642.

[4] P. Berman, J. Simon, "Investigations of Fault-Tolerant Networks of Computers," Proc. of the 20th ACM Symposium on the Theory of Computation (STOC), 1988.

[5] D.P. Bertsekas and J.N. Tsitsiklis, *Parallel and Distributed Computation,* Prentice-Hall, U.S.A., N.J. 1989.

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

[7] 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 (STOC) (May 1990), pp. 95–105.

[8] J. Magott, "Performance evaluation of concurrent systems using Petri nets," *Inform. Processing Letters*, vol. 18, pp. 7–13, Jan. 1984.

[9] M.A. Marsan, "Stochastic Petri Nets: an elementary introduction," in *Advances in Petri Nets 1989*, (Lecture Notes in Computer Science , vol. 424), Springer, 1989, pp. 1–29.

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

[11] M.K. Molloy, "Fast bounds for stochastic petri nets," *International Workshop on Timed Petri Nets*, Torino, Italy, July 1985, pp. 244–249.

[12] T. Murata, "Synthesis of decision-free concurrent systems for prescribed resources and performance," *IEEE Trans. Software Eng.*, vol. SE–6, no. 6, pp. 525–530, 1980.

[13] N. Pekergin and J-M. Vincent, "Stochastic Bounds on Execution Times of Task Graphs," Report EHEI, 1989.

[14] S. Rajsbaum, "Analysis of Distributed Protocols based on Recurrence Relations," to appear in 5th International Workshop on Distributed Algorithms, Greece, October 1991.

[15] W. Reisig, *Petri Nets*, Springer-Verlag, Berlin Heidelberg, 1985.

[16] S.M. Ross, "Stochastic Processes", J. Wiley, 1983.

[17] 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.

[18] S. Rajsbaum and M. Sidi, "On the Performance of Synchronized Programs in Distributed Networks with Random Processing Times and Transmission Delays," 4th International Workshop on Distributed Algorithms, Bari, Italy, September 1990 (Lecture Notes in Computer Science, Springer). Also in Department of Electrical Eng., Technion, EE PUB No. 770 (November 1990).

[19] J. Sifakis, "Performance evaluation of systems using nets," in *Net Theory and Applications* (Lecture Notes in Computer Science, Vol. 84), W. Brauer, Ed. Berlin, Germany: Springer, 1980, pp. 307–319.

[20] D. Stoyan, "Comparison Methods for Queues and Other Stochastic Models," English Translation (D.J. Daley, Ed.), J. Wiley & Sons, New York, 1984.