Foundations of Software Technology and Theoretical Computer Science,
13th Conference, Proceedings, R.K. Shyamasundar (Ed.),
Lecture Notes in Computer Science **761**, Springer-Verlag (1993) 388–399.

Keeping Track of the Latest Gossip: Bounded Time-Stamps Suffice

Madhavan Mukund and Milind Sohoni

School of Mathematics, SPIC Science Foundation
92 G.N. Chetty Road, Madras 600 017, INDIA
Email: {madhavan,sohoni}@ssf.ernet.in

Abstract. Consider a distributed system consisting of N independent communicating agents. Periodically, agents synchronize and exchange information, both about each other and about agents they have talked to earlier. As a result, an agent a_i may receive *indirect* information about another agent a_j which is more recent than the information exchanged in the last *direct* synchronization between a_i and a_j. The problem is to ensure that agents always come away from a synchronization with the latest possible information about all other agents. This requires that when a_i and a_j meet, they should decide which of them has more recent information about any other agent a_k. We propose an algorithm to solve this problem which is finite-state and local. Formally, this means our algorithm can be implemented by an asynchronous automaton.

Keywords: Distributed algorithms, synchronous communication, bounded time-stamping, asynchronous automata.

1 Introduction

Consider N agents $a_1, \ldots, a_N$ which synchronize with each other from time to time and exchange information about themselves and others. Whenever an agent a_i talks to another agent a_j the two of them must decide which of them has the latest information, direct or indirect, about every other agent a_k.

This is easily accomplished if the agents decide to locally time-stamp every call and pass these time-stamps along with each exchange of information. But as time progresses the time-stamp values increase without bound and most of the agents' time would be consumed in passing on large numbers, as opposed to actual gossip.

We propose a time-stamping algorithm in which the values are bounded. Despite this restriction, any pair of agents can always decide which of them has better information about every other agent. Thus, in essence, the agents may be finite state machines. Further, the algorithm itself does not induce any additional communications, and thus works for *all* communication sequences. The algorithm is implemented using asynchronous automata [Z]—a powerful and natural model for concurrent systems. The algorithm implies that we can extend the range of systems modelled by these automata to include those which require us to keep track of the latest information flow between agents.

We point out that "bounded time-stamps" have been studied, but in contexts very different from ours. Israeli and Li [IL] introduced them for creating "atomic registers" which are fundamental for algorithms on distributed systems. However, their work and that of others [CS, DS] is based on a shared-memory model, which is quite different in spirit from the asynchronous automaton model.

The paper is organized as follows. Section 2 formalizes the problem and describes the automata which run our algorithm. Sections 3 and 4 analyse the problem and present our algorithm. We conclude with a discussion of our result.

2 Preliminaries

Let $\mathcal{A} = \{a_1, a_2, \ldots, a_N\}$ be a set of *agents*. These agents communicate with each other synchronously. For simplicity, we assume that agents only synchronize in pairs. This restriction is not crucial: multi-way synchronizations are similarly handled. Details can be found in [MS].

Let a communication between agents a_i and a_j be denoted by $c_{\{i,j\}}$, abbreviated as c_{ij} or c_{ji}. Thus, if all pairs of agents can communicate with each other, we have a set of communication actions $\Sigma = \{c_{\{i,j\}} \mid i, j \in \{1, 2, \ldots, N\}$ and $i \neq j\}$. For $c_{ij} \in \Sigma$, let $dom(c_{ij})$ (the domain of c_{ij}) denote the set $\{a_i, a_j\}$. A word $\alpha \in \Sigma^*$ represents a finite sequence of communications between the agents.

2.1 Asynchronous automata

Associate a finite set (of *local states*) Q_i with each agent a_i, for $i \in \{1, 2, \ldots, N\}$. Each Q_i contains a distinguished initial state q^i_{in}. Let $Q_G = Q_1 \times Q_2 \times \cdots \times Q_N$. Q_G represents the set of possible *global states* of the system. For $\bar{q} = (q_1, q_2, \ldots, q_N) \in Q_G$, let $\bar{q}[i]$ denote the i^{th} component q_i of $\bar{q}$. With each pair of agents a_i and a_j, we associate a deterministic transition function $\delta_{ij} : (Q_i \times Q_j) \rightarrow (Q_i \times Q_j)$. Let $\delta = \{\delta_{ij} \mid i, j \in \{1, 2, \ldots, N\}, i < j\}$.

An asynchronous automaton [Z] over Σ is a structure $\mathcal{M} = (\Sigma, Q_1, Q_2, \ldots, Q_N, \delta, (q^1_{in}, q^2_{in}, \ldots, q^N_{in}))$.

We associate with $\mathcal{M}$ a global transition function $\Delta : Q_G \times \Sigma \rightarrow Q_G$ such that $\Delta((q_1, q_2, \ldots, q_N), c) = (q'_1, q'_2, \ldots, q'_N)$ iff for $a_i, a_j \in dom(c)$, $\delta_{ij}(q_i, q_j) = (q'_i, q'_j)$ and for all $a_k \notin dom(c)$, $q_k = q'_k$.

For $\alpha \in \Sigma^*$, let $|\alpha|$, the length of α, be M. We may then regard α as a function $\alpha : \{1, 2, \ldots, M\} \rightarrow \Sigma$. The n^{th} element of α, $1 \leq n \leq M$, is denoted by $\alpha(n)$.

A run of $\mathcal{M}$ on $\alpha : \{1, 2, \ldots, M\} \rightarrow \Sigma$ is a function $\rho : \{0, 1, \ldots, M\} \rightarrow Q_G$ such that $\rho(0) = (q^1_{in}, q^2_{in}, \ldots, q^N_{in})$ and $\rho(\ell) = \Delta(\rho(\ell-1), \alpha(\ell)), 1 \leq \ell \leq M$. Since $\mathcal{M}$ is deterministic, each word α gives rise to a unique run, which we denote ρ_α.

Next we define when a function is locally computable by such automata.

Definition 2.1. *Let Val be a set of values. A Σ-indexed family of functions is a set $\mathcal{F}_\Sigma = \{f_c : \Sigma^* \rightarrow Val \mid c \in \Sigma\}$.*

$\mathcal{F}_\Sigma$ *is* locally computable *if we can find an asynchronous automaton* $\mathcal{M}$ *(as above) and a family of* local functions $\mathcal{G}_\Sigma = \{g_c \mid c \in \Sigma\}$, *with each* g_c *of the form* $g_c : Q_{i_1} \times Q_{i_2} \to Val$, *where* $dom(c) = \{a_{i_1}, a_{i_2}\}$, *such that:*

$$\forall \alpha : \{1, 2, \ldots, M\} \to \Sigma.\ f_c(\alpha) = g_c(\rho_\alpha(M)[i_1], \rho_\alpha(M)[i_2])$$

In other words, for any $\alpha \in \Sigma^*$ and $c \in \Sigma$, the agents in $dom(c)$ can compute $f_c(\alpha)$ by synchronizing and checking their local states at the end of the run ρ_α.

2.2 The problem

Definition 2.2. *Let* $\alpha : \{1, 2, \ldots, M\} \to \Sigma$. *A* path π *from* a_i *to* a_j *in* α *is a sequence of pairs* $(k_1, a_{\ell_1})(k_2, a_{\ell_2}) \cdots (k_n, a_{\ell_n})$ *such that* $M \geq k_1 \geq k_2 \geq \cdots \geq k_n \geq 1$ *and* $\ell_m \in \{1, 2, \ldots, N\}$ *for* $m \in \{1, 2, \ldots, n\}$ *satisfying:*

(i) $dom(\alpha(k_1)) = \{a_i, a_{\ell_1}\}$.
(ii) $\forall m \in \{2, \ldots, n\}.\ dom(\alpha(k_m)) = \{a_{\ell_{m-1}}, a_{\ell_m}\}$.
(iii) $a_{l_n} = a_j$.

We say that $source(\pi) = k_1$, $target(\pi) = k_n$ *and that the length of* π *is* n.

Note that $k_i = k_{i+1}$ is expressly permitted. Let $\pi_1 = (k_1, a_{\ell_1})\,(k_2, a_{\ell_2}) \cdots (k_n, a_{\ell_n})$ be a path from a_i to a_j and $\pi_2 = (k'_1, a_{\ell'_1})(k'_2, a_{\ell'_2}) \cdots (k'_m, a_{\ell'_m})$ be a path from a_j to a_k, such that $k_n \geq k'_1$. Then we can concatenate the two paths to obtain a path $\pi_1\pi_2 = (k_1, a_{\ell_1})(k_2, a_{\ell_2}) \cdots (k_n, a_{\ell_n})(k'_1, a_{\ell'_1}) \cdots (k'_m, a_{\ell'_m})$ from a_i to a_k.

Definition 2.3. *Let* $\alpha : \{1, 2, \ldots, M\} \to \Sigma$. *A path* π *from* a_i *to* a_j *in* α *is* good *if for any other path* π' *from* a_i *to* a_j, $target(\pi') \leq target(\pi)$.

Thus, a good path from a_i to a_j terminates at the latest communication that a_j took part in which a_i has heard about. If π and π' are both good paths from a_i to a_j, then $target(\pi) = target(\pi')$.

With each $a_i \in \mathcal{A}$ we can associate a function $latest_{a_i} : \Sigma^* \times \mathcal{A} \to \mathbf{N}_0$ such that:

$$\forall \alpha \in \Sigma^*.\ \forall a_j \in \mathcal{A}.\ latest_{a_i}(\alpha, a_j) = \begin{cases} k \text{ if there is a good path } \pi \text{ from} \\ \quad a_j \text{ to } a_i \text{ in } \alpha \text{ and } target(\pi) = k \\ 0 \text{ otherwise} \end{cases}$$

So, $latest_{a_i}(\alpha, a_j)$ indicates the most recent information a_j has about a_i after α.

Example 2.4. *Let* $N = 5$ *and* α *be the communication sequence* $c_{12}c_{13}c_{14}c_{45}c_{25}c_{24}$. *So* $|\alpha| = 6$. *Some paths from* a_2 *to* a_1 *are the following:*

$$(1, a_1),\ (5, a_5)(4, a_4)(3, a_1),\ (6, a_4)(4, a_5)(4, a_4)(3, a_1),\ (6, a_4)(3, a_1).$$

Of these, all except the first are good paths. The paths $(4, a_4)(3, a_1)\,(2, a_3)$ *from* a_5 *to* a_3 *and* $(2, a_1)(1, a_2)$ *from* a_3 *to* a_2 *may be concatenated to get* $(4, a_4)(3, a_1)(2, a_3)$ $(2, a_1)(1, a_2)$ *from* a_5 *to* a_2. *Note that this path traverses* $\alpha(2)$ *in both directions.*

$latest_{a_1}(\alpha, a_2) = 3$, *whereas* $latest_{a_1}(\alpha, a_3) = 2$.

It is useful to draw the timing diagram of a communication sequence. Each agent's history is represented by a horizontal line, with time increasing from left to right. Vertical edges connecting pairs of agents correspond to communications. In the diagram, to trace a path from a_i to a_j, we begin on the horizontal line for a_i and move vertically and/or left along lines in the diagram, to reach the line a_j, A good path from a_i to a_j terminates as far right as possible on the line a_j.

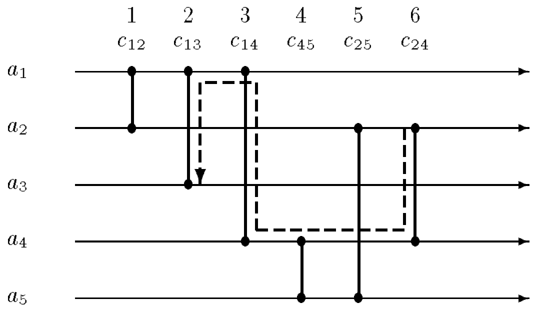

Fig. 1. Timing diagram for Example 2.4, showing a good path

Let $Val = (\mathcal{A} \cup \{*\})^N$. We define a family of functions $\mathcal{F}_\Sigma = \{f_c \mid c \in \Sigma\}$, such that $f_c = best_c : \Sigma^* \rightarrow Val$ as follows:

Let $c \in \Sigma$, *with* $dom(c) = \{a_i, a_j\}$.
Then $\forall \alpha \in \Sigma^*.\ best_c(\alpha) = (a_{k_1}, a_{k_2}, \ldots, a_{k_N})$ *where*

$$\forall m \in \{1, 2, \ldots, N\}.\ a_{k_m} = \begin{cases} a_i \text{ if } latest_{a_m}(\alpha, a_j) < latest_{a_m}(\alpha, a_i) \\ a_j \text{ if } latest_{a_m}(\alpha, a_i) < latest_{a_m}(\alpha, a_j) \\ * \ \ otherwise \end{cases}$$

So, if $dom(c) = \{a_i, a_j\}$ and $best_c(\alpha)[m] = a_i$ (respectively a_j), then a_i (respectively, a_j) has heard from a_m more recently than a_j (respectively a_i) in the communication sequence α. $best_c(\alpha)[m] = *$ signifies that both have exactly the same information about a_m after α. The main result of this paper is the following.

Theorem 2.5. *The* Σ*-indexed family of functions* $\mathcal{F}_\Sigma = \{best_c \mid c \in \Sigma\}$ *is locally computable.*

3 Global analysis

We will now analyse the "global" structure of good paths in the system. As we saw earlier, a communication sequence $\alpha \in \Sigma^*$ may have several good paths from a_i to a_j in α. Let us fix a canonical good path as follows.

Definition 3.1. *Let* $\alpha : \{1, 2, \ldots, M\} \to \Sigma$. *Path* $\pi = (k_1, a_{\ell_1})(k_2, a_{\ell_2}) \cdots (k_n, a_{\ell_n})$ *from* a_i *to* a_j *is an* ideal path *from* a_i *to* a_j *if the following conditions hold:*

(i) π *is a good path from* a_i *to* a_j.
(ii) For every $m \in \{1, 2, \ldots, n{-}1\}$, *the path* $\pi_m = (k_1, a_{\ell_1})(k_2, a_{\ell_2}) \ldots (k_m, a_{\ell_m})$ *is a good path from* a_i *to* a_{ℓ_m}.

We denote that π *is an ideal path from* a_i *to* a_j *by* $\pi : a_i \rightsquigarrow a_j$.

Example 3.2. *Consider the communication sequence of Example 2.4 (see Figure 1.) The path marked in it is the ideal path* $a_2 \rightsquigarrow a_3 = (6, a_4)(3, a_1)(2, a_3)$. *The prefixes* $(6, a_4)(3, a_1)$ *and* $(6, a_4)$ *of this path constitute the ideal paths* $a_2 \rightsquigarrow a_1$ *and* $a_2 \rightsquigarrow a_4$ *respectively. The other ideal path from* a_2 *is* $a_2 \rightsquigarrow a_5 = (5, a_5)$.

The following two observations are immediate.

Proposition 3.3. *Let* $\alpha : \{1, 2, \ldots, M\} \to \Sigma$ *and* $a_i, a_j \in \mathcal{A}$. *Let* $\pi = (k_1, a_{\ell_1})(k_2, a_{\ell_2}) \cdots (k_n, a_{\ell_n})$ *be an ideal path from* a_i *to* a_j *in* α. *Then, for each* $m \in \{1, 2, \ldots, n{-}1\}$, $\pi_m = (k_1, a_{\ell_1})\,(k_2, a_{\ell_2}) \cdots (k_m, a_{l_m})$ *is an ideal path from* a_i *to* a_{ℓ_m}.

Proposition 3.4. *Let* $\alpha : \{1, 2, \ldots, M\} \to \Sigma$ *and* $a_i, a_j \in \mathcal{A}$. *If* $latest_{a_j}(\alpha, a_i) \neq 0$ *then there is a unique ideal path* $\pi : a_i \rightsquigarrow a_j$ *in* α.

A vertical edge in the timing diagram for α which lies on some ideal path is said to be live. Formally, we have:

Definition 3.5. *Let* $\alpha : \{1, 2, \ldots, M\} \to \Sigma$ *and* $k \in \{1, 2, \ldots, M\}$. k *is* live in α for $a_i \in \mathcal{A}$ *iff there exists* $a_j \in \mathcal{A}$ *such* $\pi = (k_1, a_{\ell_1})(k_2, a_{\ell_2}) \cdots (k_m, a_{\ell_m})$ *is the ideal path* $a_i \rightsquigarrow a_j$ *and* $k = k_n$ *for some* $n \in \{1, 2, \ldots, m\}$.

Let $Live_i(\alpha) = \{k \mid k \textit{ is live in } \alpha \textit{ for } a_i\}$ and $Live(\alpha) = \bigcup_{i \in \{1,2,\ldots,N\}} Live_i(\alpha)$.

Proposition 3.6. *Let* $\alpha : \{1, 2, \ldots, M\} \to \Sigma$. $\forall i \in \{1, 2, \ldots, N\}$. $|Live_i(\alpha)| \leq N{-}1$. *So there are at most* $N(N{-}1)$ *live communications in* α.

Proof. Use Propositions 3.3 and 3.4. ☐

Lemma 3.7. *Let* $\alpha : \{1, 2, \ldots, M\} \to \Sigma$ *and* $\alpha' : \{1, 2, \ldots, M{+}1\} \to \Sigma$ *be communication sequences such that* $\alpha'(M{+}1) = c$ *for some* $c \in \Sigma$ *and* $\alpha'(m) = \alpha(m)$ *for* $m \in \{1, 2, \ldots, M\}$.

Let $dom(c) = \{a_i, a_j\}$. *Then the following statements hold:*

(i) For $a_k \notin \{a_i, a_j\}$, *for all* $a_\ell \in \mathcal{A} \setminus \{a_k\}$, *if* $\pi : a_k \rightsquigarrow a_\ell$ *is an ideal path in* α, *then* π *remains the ideal path* $a_k \rightsquigarrow a_\ell$ *in* α'.
(ii) For a_i *and* a_j, *the new ideal paths in* α' *are computed as follows. Let* $a_k \in \mathcal{A} \setminus \{a_i, a_j\}$ *and let* $\pi_{ik} : a_i \rightsquigarrow a_k$ *and* $\pi_{jk} : a_j \rightsquigarrow a_k$ *be ideal paths in* α. *Then the ideal paths* $\pi'_{ik} : a_i \rightsquigarrow a_k$ *and* $\pi'_{jk} : a_j \rightsquigarrow a_k$ *in* α' *are given as follows: Either* $\pi'_{ik} = \pi_{ik}$ *and* $\pi'_{jk} = (M{+}1, a_i)\pi_{ik}$, *or* $\pi'_{jk} = \pi_{jk}$ *and* $\pi'_{ik} = (M{+}1, a_j)\pi_{jk}$.

Proof. Part (i) is immediate, since α' has no new paths originating at a_k for $a_k \notin \{a_i, a_j\}$.

To show part (ii), let $a_k \in \mathcal{A} \setminus \{a_i, a_j\}$.

Case 1 Suppose that $target(\pi_{jk}) < target(\pi_{ik})$. Then clearly, π_{ik} remains the ideal path $a_i \rightsquigarrow a_k$ in α'. It is also obvious that $(M{+}1, a_i)\pi_{ik}$ is a good path from a_j to a_k. It is not difficult to show that this is in fact the ideal path $a_j \rightsquigarrow a_k$ in α'.

Case 2 The case $target(\pi_{ik}) < target(\pi_{jk})$ is symmetric to Case 1.

Case 3 The last case is when $target(\pi_{ik}) = target(\pi_{jk})$.
Let $\pi_{ik} = (k_1, a_{\ell_1}) \cdots (k_m, a_{\ell_m})$ and $\pi_{jk} = (k'_1, a_{\ell'_1}) \cdots (k'_{m'}, a_{\ell'_{m'}})$. We know that $dom(\alpha(k_m)) = \{a_{\ell_{m-1}}, a_{\ell_m}\}$ and $dom(\alpha(k'_{m'})) = \{a_{\ell'_{m'-1}}, a_{\ell'_{m'}}\}$. Since $target(\pi_{ik}) = target(\pi_{jk})$, we have $a_{\ell_m} = a_{\ell'_{m'}} = a_k$ and $k_m = k'_{m'}$. It follows that $a_{\ell'_{m'-1}} = a_{\ell_{m-1}}$.

We "step back" on both π_{ik} and π_{jk} and look at the paths $\pi_{ik}^{m-1} = (k_1, a_{\ell_1}) \cdots (k_{m-1}, a_{\ell_{m-1}})$ and $\pi_{jk}^{m'-1} = (k'_1, a_{\ell'_1}) \cdots (k'_{m'-1}, a_{\ell'_{m'-1}})$. Let $a_{\ell''} = a_{\ell_{m-1}} = a_{\ell'_{m'-1}}$. Then the two paths π_{ik}^{m-1} and $\pi_{jk}^{m'-1}$ are ideal paths $a_i \rightsquigarrow a_{\ell''}$ and $a_j \rightsquigarrow a_{\ell''}$ in α (by Proposition 3.3) and we induce on the length of the paths.

We argue for the base case, when we reach π_{ik}^0 or π_{jk}^0—i.e., either π_{ik} or π_{jk} becomes empty. Without loss of generality, assume that π_{ik} becomes empty. At this point we have $\pi_{jk}^{m'-m} : a_j \rightsquigarrow a_i$ in α. We replace the path $\pi_{jk}^{m'-m}$ with the unit path $(M{+}1, a_i)$ to get a new ideal path $(M{+}1, a_i)\pi_{ik} : a_j \rightsquigarrow a_k$ in α'.

□

So, agents a_i and a_j can update their ideal paths purely locally provided they know which of them has the better path to each agent $a_k \notin \{a_i, a_j\}$.

Corollary 3.8. *Let $\alpha : \{1, 2, \ldots, M\} \to \Sigma$ and $\alpha' : \{1, 2, \ldots, M{+}1\} \to \Sigma$ be communication sequences such that $\alpha'(M{+}1) = c$ for some $c \in \Sigma$ and $\alpha'(m) = \alpha(m)$ for $m \in \{1, 2, \ldots, M\}$.*

Then the following statements hold:

(i) $M{+}1$ is live in α'.
(ii) $Live(\alpha') \subseteq Live(\alpha) \cup \{M{+}1\}$.

Proof. Immediate from Lemma 3.7. □

4 Local Analysis

The analysis of the previous section immediately gives us an (unbounded) time-stamping algorithm by which agents may locally update ideal paths. Every communication c_{ij} is time-stamped by agents a_i and a_j with their local times. Each agent a_i maintains its ideal paths $a_i \rightsquigarrow a_k$ as a sequence of communications distinguished by the time-stamps given to them. When a_i and a_j synchronize after α, they first time-stamp the new synchronization action c_{ij}. Let $\pi_i : a_i \rightsquigarrow a_k$ and $\pi_j : a_j \rightsquigarrow a_k$ in α. Both π_i and π_j end with a communication involving a_k. So, the time-stamps given by a_k to the last communications in the sequences π_i and π_j will reveal which is later. Once this is known, the new ideal paths $\pi'_i : a_i \rightsquigarrow a_k$ and $\pi'_j : a_j \rightsquigarrow a_k$ in α' may be computed, as in the proof of Lemma 3.7.

Our algorithm is a modification of this procedure. When a_i and a_j synchronize, they label the current communication—this label is chosen from a sufficiently large, but finite, set $\mathcal{L}$. Agents maintain mildly enhanced versions of ideal paths called primary paths (Definition 4.1). To detect if a_i has a better ideal path to a_k than a_j, it suffices to check if some label from the ideal path of a_j is present on the primary paths of a_i (Lemma 4.3). In other words, a_i and a_j have only to check for *equality* of labels along their primary paths to compare their ideal paths.

The main complication is for a_i and a_j to decide which labels from $\mathcal{L}$ are currently in use (i.e., appear along primary paths for other agents) and which may be reused. This is decided by secondary paths maintained by each agent (Definition 4.4, Lemma 4.5). The update of the primary and secondary paths is similar to that of ideal paths and needs no additional information.

Henceforth, we assume we have a set of labels $\mathcal{L}$. Different occurrences of $c \in \Sigma$ in a communication sequence $\alpha : \{1, 2, \ldots, M\} \to \Sigma$ will be assigned distinct labels from $\mathcal{L}$. In other words, we regard α as an injective map from $\{1, 2, \ldots, M\}$ to $\Sigma \times \mathcal{L}$. We may regard the set $\Sigma \times \mathcal{L}$ as a set of *events* $\mathcal{E}$. Given an event $e = (c, l) \in \Sigma \times \mathcal{L}$, we shall say $a_i \in dom(e)$ to mean $a_i \in dom(c)$. Initially we assume that we have a infinite set of distinct labels $\mathcal{L} = \{\lambda_1, \lambda_2, \ldots\}$. Later we show that it suffices to use a finite set of labels which may be "recycled".

We now represent a communication sequence α as a sequence of events, i.e., $\alpha : \{1, 2, \ldots, M\} \to \mathcal{E}$. Given a path $\pi = (k_1, a_{\ell_1})(k_2, a_{\ell_2}) \cdots (k_n, a_{\ell_n})$, we let σ_π denote the sequence of events $\alpha(k_1)\alpha(k_2) \ldots \alpha(k_n)$.

Definition 4.1. *Let $\alpha : \{1, 2, \ldots, M\} \to \mathcal{E}$ be a communication sequence and $a_i \in \mathcal{A}$. A* primary path *for a_i is a path $\pi = (k_1, a_{\ell_1})(k_2, a_{\ell_2}) \cdots (k_n, a_{\ell_n})$ $(k_{n+1}, a_{\ell_{n+1}})$ such that for some $a_j, a_k \in \mathcal{A}$ we have:*

(i) $a_{\ell_n} = a_j$ and $\pi_n = (k_1, a_{\ell_1})(k_2, a_{\ell_2}) \cdots (k_n, a_{\ell_n})$ is the ideal path $a_i \rightsquigarrow a_j$ in α.

(ii) $a_{\ell_{n+1}} = a_k$ and for all $m \in \{k_{n+1}{+}1, k_{n+1}{+}2, \ldots, k_n\}$, $dom(\alpha(m)) \neq \{a_j, a_k\}$.

We say that π is a primary path from a_i to a_j to a_k and denote this by $a_i \rightsquigarrow a_j \to a_k$.

So, a primary path $a_i \rightsquigarrow a_j \to a_k$ is an ideal path $\pi : a_i \rightsquigarrow a_j$ extended with the most recent c_{jk} communication before and including the point *target*(π). Notice that the definition above does not rule out the case $k_{n+1} = k_n$.

Example 4.2. *In Example 2.4 (see Figure 1), $a_2 \rightsquigarrow a_1 = (6, a_4)(3, a_1)$. The primary path $a_2 \rightsquigarrow a_1 \to a_3$ is given by $(6, a_4)(3, a_1)(2, a_3)$. On the other hand, $a_2 \rightsquigarrow a_1 \to a_4 = (6, a_4)\ (3, a_1)(3, a_4)$; i.e., the communication $a_1 \to a_4$ in $a_2 \rightsquigarrow a_1 \to a_4$ is the same as the last communication in $a_2 \rightsquigarrow a_1$.*

Lemma 4.3. *Let $\alpha : \{1, 2, \ldots, M\} \to \mathcal{E}$, with $\pi_i : a_i \rightsquigarrow a_k$ and $\pi_j : a_j \rightsquigarrow a_k$ two ideal paths in α, for $a_i, a_j, a_k \in \mathcal{A}$. Let $\sigma_{\pi_i} = e'_1 e'_2 \ldots e'_m$ and $\sigma_{\pi_j} = e''_1 e''_2 \ldots e''_n$. Then:*

(i) $target(\pi_i) \leq target(\pi_j)$ iff for some $e'_\ell \in \sigma_{\pi_i}$, e'_ℓ also appears on some primary path for a_j.
(ii) $target(\pi_j) \leq target(\pi_i)$ iff for some $e'_\ell \in \sigma_{\pi_j}$, e'_ℓ also appears on some primary path for a_i.

Proof. Since the cases (i) and (ii) are symmetric, we prove (i).
($\Leftarrow$): Suppose that for some $e'_\ell \in \sigma_{\pi_i}$, e'_ℓ also appears on some primary path for a_j. Then, there is a path $\tilde{e}_1 \tilde{e}_2 \ldots \tilde{e}_{m'}$ originating from a_j such that $\tilde{e}_{m'} = e'_\ell$. So, the sequence $\tilde{e}_1 \tilde{e}_2 \ldots \tilde{e}_{m'-1} e'_\ell e'_{\ell+1} \ldots e'_m$ corresponds to a path $\tilde{\pi}$ from a_j to a_k such that $target(\tilde{\pi}) = target(\pi_i)$. $target(\pi_j)$ must be at least as large as $target(\tilde{\pi})$ since it is an ideal path $a_j \rightsquigarrow a_k$. So $target(\pi_i) \leq target(\pi_j)$ as required.
($\Rightarrow$): Suppose $target(\pi_i) \leq target(\pi_j)$. We then have to show that for some $e'_\ell \in \sigma_{\pi_i}$, e'_ℓ also appears on some primary path for a_j. We proceed by induction on m, the length of π_i.

$m = 1$: Then $\pi_i = (k_1, a_{\ell_1})$. Let $\pi_j = (k'_1, a_{\ell'_1})(k'_2, a_{\ell'_2}) \cdots (k'_n, a_{\ell'_n})$. We know that $a_{\ell_1} = a_{\ell'_n}$ and $k_1 \leq k'_n$. So, we have a path $\pi' = \pi_j(k_1, a_i)$ from a_j to a_i such that $k_1 = target(\pi')$.

This means that the ideal path $\tilde{\pi} : a_j \rightsquigarrow a_i$ is such that $k_1 = target(\pi') \leq target(\tilde{\pi})$. Then $\alpha(k_1)$ must be the most recent communication c_{ik} before $target(\tilde{\pi})$. So $e'_1 = \alpha(k_1)$ appears on the primary path $a_j \rightsquigarrow a_i \to a_k$.

$m > 1$: Let $\pi_i = (k_1, a_{\ell_1}) \cdots (k_m, a_{\ell_m})$ and $\pi_j = (k'_1, a_{\ell'_1}) \cdots (k'_n, a_{\ell'_n})$. Inductively assume that our claim holds for all ideal paths of length less than m which originate from a_i.

We know that $a_{\ell_m} = a_{\ell'_n} = a_k$ and $k_m \leq k'_n$. So, we have a path $\pi' = \pi_j(k_m, a_{\ell_{m-1}})$ from a_j to $a_{\ell_{m-1}}$.

The timing diagram we have is somewhat like the one in Figure 2.

Let $\tilde{\pi} : a_j \rightsquigarrow a_{\ell_{m-1}}$ in α. Then, we know that $k_m = target(\pi') \leq target(\tilde{\pi})$.

Suppose $target(\tilde{\pi}) \leq k_{m-1}$. Then, the communication $\alpha(k_m)$ must be the most recent $c_{\ell_{m-1}k}$ communication before $target(\tilde{\pi})$—if there were a more recent communication of this kind, it would appear on the ideal path π_i. So, the event $e_m = \alpha(k_m)$ appears on the primary path $a_j \rightsquigarrow a_{\ell_{m-1}} \to a_k$.

On the other hand, suppose $k_{m-1} < target(\tilde{\pi})$. We know that the path $\pi_i^{m-1} = (k_1, a_{\ell_1})(k_2, a_{\ell_2}) \cdots (k_{m-1}, a_{\ell_{m-1}})$ is the ideal path $a_i \rightsquigarrow a_{\ell_{m-1}}$. But $target(\pi_i^{m-1}) \leq target(\tilde{\pi})$ and the length of π_i^{m-1} is less than m. So, by the induction hypothesis, there is an event in the sequence $e'_1 e'_2 \ldots e'_{m-1}$ which occurs on some primary path for a_j and we are done.

□

So, to compare $\pi_i : a_i \rightsquigarrow a_k$ and $\pi_j : a_j \rightsquigarrow a_k$, we just have to look for events which are common to σ_{π_i} and $\sigma_{\pi'}$, for some primary path π' of a_j, or common to σ_{π_j} and $\sigma_{\pi''}$, for some primary path π'' of a_i.

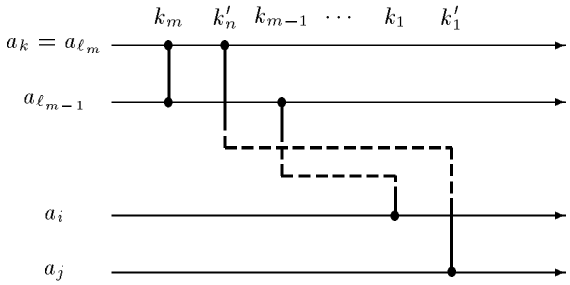

Fig. 2. Timing diagram for Lemma 4.3

In other words, to compare ideal paths, we need only test *equality* of events. The method works as long as communications in the primary paths are consistently labelled as distinct events. A slight extension of the arguments used in Lemma 3.7 and Corollary 3.8 establishes that communications which become dead (i.e., disappear from all primary paths in the system) do not become live again (i.e., reappear on some agent's primary paths).

Suppose agents a_i and a_j can decide that a label $\lambda \in \mathcal{L}$ assigned to a previous c_{ij} action is no longer being used—i.e., the event (c_{ij}, λ) is not on a primary path for any agent. Then, λ can be re-used for a later synchronization. Since no "old" copy of (c_{ij}, λ) is present on any primary path, this re-use of λ will not introduce any inconsistency into the procedure for comparing ideal paths.

Definition 4.4. *Let $\alpha : \{1, 2, \ldots, M\} \to \mathcal{E}$ be a communication sequence and $\pi = (k_1, a_{\ell_1})(k_2, a_{\ell_2}) \cdots (k_n, a_{\ell_n})$ a path in α. π is a* secondary path *for $a_i \in \mathcal{A}$ if we can find agents $a_j, a_k, a_m \in \mathcal{A}$ such that $a_i \neq a_j$, $a_j \neq a_k$ and $a_k \neq a_m$, π is a path from a_i to a_m and for some $n_j \in \{1, 2, \ldots, n{-}1\}$ the following two conditions are satisfied.*

(i) $\pi_{n_j} = (k_1, a_{\ell_1})(k_2, a_{\ell_2}) \cdots (k_{n_j}, a_{\ell_{n_j}})$ is the ideal path $a_i \rightsquigarrow a_j$ in α.
(ii) Let $\alpha_{n_j} : \{1, 2, \ldots, k_{n_j}\} \to \mathcal{E}$—i.e., the prefix of α upto k_{n_j}. Then, the path $(k_{n_j+1}, a_{\ell_{n_j+1}}) \cdots (k_n, a_{\ell_n})$ is the primary path $a_j \rightsquigarrow a_k \to a_m$ in α_{n_j}.

We say that π is a secondary path from a_i to a_j to a_k to a_m and denote this by $\pi : a_i \rightsquigarrow a_j \rightsquigarrow a_k \to a_m$.

Lemma 4.5. *Let $\alpha : \{1, 2, \ldots, M\} \to \mathcal{E}$. Suppose e appears on a primary path for a_j and $a_i \in dom(e)$. Then, e appears either on a primary path or a secondary path for a_i.*

Proof. Suppose e is on a primary path $\pi : a_j \rightsquigarrow a_k \to a_\ell$. Let $\pi = (k_1, a_{\ell_1})(k_2, a_{\ell_2}) \cdots (k_{n+1}, a_{\ell_{n+1}})$. Then, for some $m \in \{1, 2, \ldots, n{+}1\}$, $e = \alpha(k_m)$. So, either $a_{\ell_{m-1}} = a_i$ or $a_{\ell_m} = a_i$. Without loss of generality, we assume that $a_{\ell_m} = a_i$.

Case 1. ($m \leq n$): Then $(k_1, a_{\ell_1})(k_2, a_{\ell_2}) \cdots (k_n, a_{\ell_n})$, the ideal path $a_j \rightsquigarrow a_k$ in α, passes through a_i. This path is shown in Figure 3.

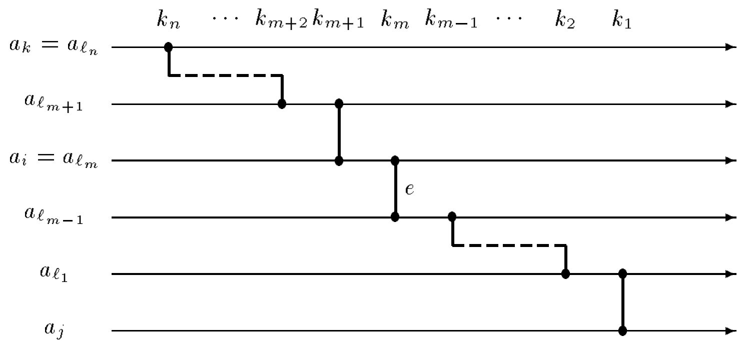

Fig. 3. Timing diagram for Lemma 4.5

We claim that for some $m' \in \{1, 2, \ldots, m-1\}$, e lies on the secondary path $a_i \rightsquigarrow a_{\ell_{m'}} \rightsquigarrow a_k \rightarrow a_\ell$. To see this, consider the ideal path $\pi_{ij} : a_i \rightsquigarrow a_j$ in α. If $k_1 \leq \mathit{target}(\pi_{ij})$, e lies on the secondary path $a_i \rightsquigarrow a_j \rightsquigarrow a_k \rightarrow a_\ell$. If $k_1 > \mathit{target}(\pi_{ij})$ then, look at $\pi_{i\ell_1} : a_i \rightsquigarrow a_{\ell_1}$. We know that $\mathit{target}(\pi_{i\ell_1}) < k_1$—otherwise we would be able to reach the point k_1 on a_j from a_i.

If $k_2 \leq \mathit{target}(\pi_{i\ell_1}) < k_1$, then e lies on the secondary path $a_i \rightsquigarrow a_{\ell_1} \rightsquigarrow a_k \rightarrow a_\ell$. On the other hand, if $\mathit{target}(\pi_{i\ell_1}) < k_2$, we look at $\pi_{i\ell_2} : a_i \rightsquigarrow a_{\ell_2}$ and repeat the analysis for $\pi_{i\ell_1}$.

In the worst case we come upto $\pi_{i\ell_{m-1}} : a_i \rightsquigarrow a_{\ell_{m-1}}$. We know that $\mathit{target}(\pi_{i\ell_{m-1}}) < k_{m-1}$ by the above analysis. On the other hand, $\alpha(k_m) = c_{i\ell_{m-1}}$. So we definitely have $k_m \leq \mathit{target}(\pi_{i\ell_{m-1}}) < k_{m-1}$. So we must have e lying on the secondary path $a_i \rightsquigarrow a_{\ell_{m-1}} \rightsquigarrow a_k \rightarrow a_\ell$.

Case 2. ($m = n+1$): So, $a_i = a_\ell$. Then, a slight modification of the argument put forward for Case 1 yields that e either lies on a secondary path $a_i \rightsquigarrow a_{\ell_{m'}} \rightsquigarrow a_k \rightarrow a_i$ for some $m' \in \{1, 2, \ldots, n-1\}$ or e lies on the primary path $a_i \rightsquigarrow a_k \rightarrow a_i$. We omit the details. □

So, the number of different copies of an action c_{ij} that can be part of other agents' primary paths is bounded—by the preceding lemma, each such copy must appear on the primary or secondary paths of a_i and a_j, which are bounded in length and number. In addition, once a particular c_{ij} event disappears from the primary and secondary paths of a_i and a_j, these two agents know that the event is not present on *any* primary path in the system. So, a_i and a_j can locally decide to recycle the label assigned to that event, knowing that this will not affect the outcome of any comparison in Lemma 4.3. This implies that a finite set of labels suffices.

The algorithm

Let $\mathcal{L} = \{\lambda_1, \lambda_2, \ldots, \lambda_K\}$ be a finite set of labels and let $\mathcal{E} = \Sigma \times \mathcal{L}$. We assume that each agent $a_i \in \mathcal{A}$ maintains all primary and secondary paths (starting at a_i) as sequences of events. In addition, each agent a_i also maintains an *immediate history* of events $H_i = \{last_{ij} \mid j \neq i\}$, such that for each $a_j \neq a_i$, $last_{ij} \in \mathcal{E}$ is the most recent event e in a_i's history such that $dom(e) = \{a_i, a_j\}$.

Suppose agents a_i and a_j synchronize after α. They then update this local information as follows.

(i) Let $\lambda \in \mathcal{L}$ be the first label in the sequence $\lambda_1\lambda_2\ldots\lambda_K$ which does not appear in any primary or secondary path for a_i or a_j. (We assume that $\mathcal{L}$ is large enough that we can always do this. See Lemma 4.6 below.) Label the new c_{ij} communication with λ—i.e., the new event is $e_\lambda = (c_{ij}, \lambda)$.
(ii) Update H_i to H_i' by setting $last_{ij} = e_\lambda$. Symmetrically update H_j to H_j' by setting $last_{ji} = e_\lambda$.
(iii) Define the new primary paths $\pi'_{ijk} : a_i \rightsquigarrow a_j \longrightarrow a_k$ by $\pi'_{ijk} = e_\lambda last_{jk}$, for each event $last_{jk} \in H_j'$. Symmetrically, define the new primary paths π'_{jik}.
(iv) For $a_k \notin \{a_i, a_j\}$, update primary paths $\pi_{ik\ell}$ and $\pi_{jk\ell}$ using lemmas 3.7 and 4.3. Secondary paths are updated in a similar fashion. Note that the update of the primary or secondary paths follows from the update of ideal path, i.e., Lemma 3.7.

It is not difficult to establish the following bounds on our local data structures.

Lemma 4.6. *Let $a_i \in \mathcal{A}$.*

(i) The primary paths of a_i can be stored as an $\mathcal{E}$-labelled tree with $O(N^2)$ nodes.
(ii) The secondary paths of a_i can be stored as an $\mathcal{E}$-labelled tree with at most $O(N^3)$ nodes.
(iii) It suffices to use $O(N)$ labels in $\mathcal{L}$.

To conclude this section, we formally relate the algorithm provided here to the Theorem in Section 2 that we set out to prove. A local state for a_i consists of a $\mathcal{E}$-labelled tree of primary paths and an $\mathcal{E}$-labelled tree of secondary paths for a_i. ¿From Lemma 4.6, it follows that the set of possible local states for a_i is bounded. Given $c_{ij} \in \Sigma$, the local function $g_{c_{ij}}$ which computes $best_{c_{ij}}$ is just the one which applies Lemma 4.3 to the local states of a_i and a_j and compares ideal paths $a_i \rightsquigarrow a_k$ and $a_j \rightsquigarrow a_k$.

5 Discussion

In this paper, we have demonstrated a finite-state algorithm for keeping track of the flow of information in a system where N agents communicate synchronously.

Our algorithm works on an asynchronous automaton [Z]. Asynchronous automata are closely related to trace theory [Maz], an important language-theoretic model of concurrent systems. Versions of these automata have been used to characterize ω-regular trace languages [GP, DM].

In addition to these connections to trace theory, versions of asynchronous automata are also used in model-checking, where one wants to verify whether a given system's behaviour corresponds to its specification [GW]. Recently, Thiagarajan [T] has developed an extension of propositional linear-time temporal logic which is interpreted over infinite traces, rather than linear sequences. This logic appears to be quite expressive, while remaining decidable. Our algorithm plays a crucial role in establishing the decidability of Thiagarajan's logic.

One interesting problem is to try and characterize the functions which are locally computable (Definition 2.1). Another line of work is to extend our approach to deal with (reliable) message-passing systems. We believe this is possible, subject to the assumption that there is an overall bound B on the number of *new* messages that a_i will send to a_j without an acknowledgment (direct or indirect) from a_j. A report of this extension is in preparation.

Acknowledgments We thank P.S. Thiagarajan for suggesting the problem and for numerous discussions which have helped clean up the presentation.

References

[CS] R. Cori, E. Sopena: Some combinatorial aspects of time-stamp systems, *Europ. J. Combinatorics*, **14** (1993) 95–102.

[DM] V. Diekert, A. Muscholl: Deterministic asynchronous automata for infinite traces, *Proc. STACS '93*, *LNCS* **665** (1993) 617–628.

[DS] D. Dolev, N. Shavit: Bounded concurrent time-stamps are constructible, *Proc. ACM STOC* (1989) 454–466.

[GP] P. Gastin, A. Petit: Asynchronous cellular automata for infinite traces, *Proc. ICALP '92*, *LNCS* **623** (1992) 583–594.

[GW] P. Godefroid, P. Wolper: A partial order approach to model checking, *Proc. 6th IEEE LICS*, Amsterdam (1991) 406–415.

[IL] A. Israeli, M. Li: Bounded time-stamps, *Proc. 28th IEEE FOCS* (1987) 371–382.

[Maz] A. Mazurkiewicz: Basic notions of trace theory, in: J.W. de Bakker, W.-P. de Roever, G. Rozenberg (eds.), *Linear time, branching time and partial order in logics and models for concurrency*, *LNCS* **354**, (1989) 285–363.

[MS] M. Mukund, M. Sohoni: Keeping track of the latest gossip: Bounded time-stamps suffice, *Report TCS-93-3*, School of Mathematics, SPIC Science Foundation, Madras, India (1993).

[T] P.S. Thiagarajan: A trace based extension of PTL, *Report TCS-93-4*, School of Mathematics, SPIC Science Foundation, Madras, India (1993).

[Z] W. Zielonka: Notes on finite asynchronous automata, *R.A.I.R.O.—Inf. Théor. et Appl.*, **21** (1987) 99–135.