GIT-ICS-81/13

A LOWER BOUND FOR THE TIME TO ASSURE INTERACTIVE CONSISTENCY†

MICHAEL J. FISCHER*
NANCY A. LYNCH**

SEPTEMBER 1981

* Department of Computer Science
Yale University
New Haven, CT 06520

** School of Information and Computer Science
Georgia Institute of Technology
Atlanta, GA 30332

†This research was supported in part by the National Science Foundation under grants MCS77-02474, MCS77-15628, MCS80-03337, U.S. Army Research Office Contract DAAG29-79-C-0155 and Office of Naval Research Contracts N00014-79-C-0873 and N00014-80-C-0221.

A Lower Bound for the Time to Assure Interactive Consistency

Michael J. Fischer
Nancy A. Lynch

1. Introduction

The problem of "assuring interactive consistency" is defined in [PSL]. It is assumed that there are n isolated processors, of which at most m are faulty. The processors can communicate by means of two-party messages, using a medium which is reliable and of negligible delay. The sender of a message is always identifiable by the receiver. Each processor p has a private value $\sigma(p)$. The problem is to devise an algorithm that will allow each processor p to compute a value for each processor r, such that (a) if p and r are nonfaulty, then p computes r's private value $\sigma(r)$, and (b) all the nonfaulty processors compute the same value for each processor r.

It is shown in [PSL] that if $n < 3m + 1$, then there is no algorithm which assures interactive consistency. On the other hand, if $n \geq 3m + 1$, then an algorithm does exist. The algorithm presented in [PSL] uses $m + 1$ rounds of communication, and thus can be said to require "time" $m + 1$. An obvious question is whether fewer rounds of communication suffice to solve the problem.

In this paper, we answer this question in the negative. That is, we show that any algorithm which assures interactive consistency in the presence of m faulty processors requires at least $m + 1$ rounds of communication.

The remainder of the paper is organized as follows. Section 2 contains motivation for our formal model and problem statement, Section 3 contains the notation and definitions. Section 4 contains a reduction of our set of allowable algorithms to a more restrictive set of "uniform" algorithms. Section 5 contains a restatement of the relevant results of [PSL]. Section 6 contains our main lower bound result. Section 7 contains an important open question.

The reader is urged to read [LSP] and [L] for discussion of the practical importance of assuring interactive consistency, and [PSL] for additional results not immediately relevant to this paper. Other related papers are [DW] and [D].

2. Motivation for the Definitions

A general model for solving the interactive consistency problem might consist of n processors (automata) communicating by means of n^2 one-way "communication channels". Each channel can be formalized as a shared variable which can be modified by exactly one processor and read by exactly one processor. (Such a formalization can be carried out, for example, within the model of [LF].) The variables which each processor can modify are called its "out-channels", while the variables it can read are called its "in-channels".

Each processor p starts with an arbitrary private value $\sigma(p)$. Execution of the system proceeds in synchronous "rounds"; at each round, the following two steps occur: (1) First, each nonfaulty processor writes values ("sends messages") derived from its state into all of its out-channels, while each faulty processor writes arbitrary values into all of its out-channels. (2) Second, each processor

reads the values from all of its in-channels. After some specified number, k, of rounds, each processor p outputs a vector of values, one for each processor r. These outputs are required to satisfy conditions (a) and (b) stated in the Introduction.

If the only complexity measure of interest is the number of rounds, then we can assume without loss of generality that the messages sent by each nonfaulty processor p on the first round are all exactly equal to its private value $\sigma(p)$, and that the messages sent by each nonfaulty processor on subsequent rounds are all exactly the set of messages received from all processors on the previous round. That is, if there is any correct k-round algorithm, then there is a correct k-round algorithm in which exactly the messages described above are sent. This is so because (i) it is clear that the given information is the maximum nontrivial information which could be sent, (ii) it does not hurt to send nonfaulty processors the maximum information, since they can derive any needed information from the given maximum information, and (iii) it does not hurt to send faulty processors the maximum information, since it is assumed that the faulty processors can send arbitrary messages in any case - i.e. they could "guess" any missing information.

In such a maximum-information algorithm, the output vector of each processor p is simply a function of the set of all values received by p at all rounds of the computation. (So far, this reduction is as in [PSL] .) In addition, if p is nonfaulty, then the set of messages received by p at all rounds of the computation is determined by the set of messages received by p at the last round (since p sends messages to itself at each round containing the information p received at earlier rounds). Since the correctness conditions involve only the outputs of nonfaulty processors, it suffices to formalize the output of p as a function of the set of messages received by p at the last round of communication only.

3. Notation and Definitions

If A is any alphabet, $i, j \in N \cup \{0\}$, $i \geq j$, we use $A^{i:j}$ to denote the set of strings of symbols in A, of length at least i and at most j.

Let P be the set of *processors*, $|P| = n$, and let m be an upper bound on the number of faulty processors. Fix V to be the domain of values on which the processors wish to reach agreement. Assume $\{0,1\} \subseteq V$.

For any $k \in N \cup \{0\}$, let $\mathfrak{U}^k$ denote the set of mappings from P^k into V. (An element of $\mathfrak{U}^k$ is intended to represent a set of messages which a processor could receive at the last round of a computation.)

A k-round *algorithm* A (for P) is a set $\{F_p : p \in P\}$ of functions, where $F_p: \mathfrak{U}^k \times P \rightarrow V$. A is *uniform* if $F_p = F_q$ for all $p, q \in P$.

A k-round *scenario* (for P with m faults) is a mapping $\sigma: P^{1:k+1} \rightarrow V$, such that $|T_\sigma| \geq n - m$, where T_σ (the set of *truthtellers*) $= \{q \in P : \sigma(wqp) = \sigma(wq)$ for all $p \in P$ and all $w \in P^{0:k-1}\}$. Intuitively, $\sigma(p_1p_2...p_i)$ is intended to represent the value in V which p_{i-1} told p_i that p_{i-2} told p_{i-1} that ... that p_1 told p_2 was p_1's private value; as a special case, $\sigma(p)$ represents p's private value. (Note that this definition reverses the direction of the string arguments in the [PSL] definition.)

Let L_σ (the set of *liars*) denote $P\text{-}T_\sigma$. If $\sigma : P^{1:k+1} \rightarrow V$ and $p \in P$, then p's *view* of σ is the map $\sigma_p \in \mathfrak{U}^k$

given by $\sigma_p(w) = \sigma(wp)$. Let $\mathcal{V}_p^k = \{\sigma_p : \sigma$ is a k-round scenario (for P with m faults) and $p \in T_\sigma\}$. That is, $\mathcal{V}_p^k$ is the set of possible views for p when p is a truthteller in a k-round scenario.

Let $A = \{F_p : p \in P\}$ be a k-round algorithm. Then A *assures interactive consistency* (for P with m faults) provided for each k-round scenario σ (for P with m faults), the following two conditions hold.

(a) (*Validity*) $F_p(\sigma_p, r) = \sigma(r)$ for all $p, r \in T_\sigma$,

(b) (*Agreement*) $F_p(\sigma_p, r) = F_q(\sigma_q, r)$ for all $p, q \in T_\sigma$ and all $r \in P$.

4. Reduction to Uniform Algorithms

In this section, we show that it suffices to restrict attention to uniform algorithms.

Lemma 1: Assume $n \geq 2m+1$. Let $A = \{F_p : p \in P\}$ be a k-round algorithm which assures interactive consistency. Then $F_p(\alpha,r) = F_q(\alpha,r)$ for all p, q, $r \in P$ and all $\alpha \in \mathcal{V}_p^k \cap \mathcal{V}_q^k$.

Proof: Let $p, q, r \in P$, $\alpha \in \mathcal{V}_p^k \cap \mathcal{V}_q^k$. Then there are k-round scenarios σ and τ such that $p \in T_\sigma$, $q \in T_\tau$, and $\alpha = \sigma_p = \tau_q$. Let $s \in T_\sigma \cap T_\tau$. (Such an s is guaranteed to exist because $n \geq 2m + 1$.)

Modify only the last round of σ and τ to obtain new k-round scenarios σ' and τ', as follows. Let $\sigma'(ws) = \sigma(wp)$ for all $w \in P^k$, and let $\sigma'(x) = \sigma(x)$ otherwise. Similarly, let $\tau'(ws) = \tau(wq)$ for all $w \in P^k$, and let $\tau'(x) = \tau(x)$ otherwise. It is easy to check that σ' and τ' are scenarios, that $\{p,s\} \subseteq T_{\sigma'}$, $\{q,s\} \subseteq T_{\tau'}$, and that $\alpha = \sigma'_p = \sigma'_s = \tau'_q = \tau'_s$. Thus, $F_p(\alpha,r) = F_p(\sigma'_p,r)$, $= F_s(\sigma'_s,r)$ by the agreement property, $= F_s(\tau'_s,r)$, $= F_q(\tau'_q,r)$ by the agreement property, $= F_q(\alpha,r)$.

□

Theorem 1: Assume $n \geq 2m + 1$. If there is a k-round algorithm which assures interactive consistency, then there is a k-round uniform algorithm which assures interactive consistency.

Proof: Let $A = \{F_p : p \in P\}$ be a k-round algorithm which assures interactive consistency. Define $F: \mathcal{U}^k \times P \rightarrow V$ as follows. Let $F(\alpha,r) =$ $F_p(\alpha,r)$ if $\alpha \in \mathcal{V}_p^k$, 0 otherwise.

Lemma 1 shows that this definition is consistent. Then the algorithm which uses F for all processors is a k-round algorithm which assures interactive consistency.

□

5. Earlier Results

In this section, we state the two relevant results from [PSL].

Theorem 2: Assume $n < 3m + 1$. Then there is no algorithm which assures interactive

consistency.

Proof: [PSL]. □

Theorem 3: Assume $n \geq 3m + 1$. Then there is an $m+1$-round uniform algorithm which assures interactive consistency.

Proof: [PSL]. □

6. Lower Bound

In this section, we present our main result.

Theorem 4: If $k \leq m$, then there is no k-round algorithm which assures interactive consistency.

Proof: The theorem is easily seen to be true if $m = 0$, so assume that $m \geq 1$. Assume that the theorem is false: that $k \leq m$ and there is a k-round algorithm $A = \{F_p : p \in P\}$ which assures interactive consistency. By Theorem 1, we can assume that A is uniform, i.e. that $F_p = F$ for all $p \in P$. By Theorem 2, we know that $n \geq 3m + 1$.

Define a relation $\sim$ on $\mathcal{U}^k$ as follows. Let $\alpha \sim \beta$ provided there exist a k-round scenario σ and $p, q \in T_\sigma$ for which $\alpha = \sigma_p$ and $\beta = \sigma_q$. Let $\equiv$ be the smallest equivalence relation containing $\sim$. By the agreement property, we have:

Fact 1: $F(\alpha,r) = F(\beta,r)$ for all $\alpha, \beta \in \mathcal{U}^k$ with $\alpha \equiv \beta$, and all $r \in P$.

For each $v \in V$, $w \in P^k$, let $\gamma_v(w) = v$. By the validity property, we have:

Fact 2: $F(\gamma_v,r) = v$ for all $r \in P$ and all $v \in V$.

Define an arbitrary total order on P, let $N = n^k$, and let $I: P^k \rightarrow \{1,...,N\}$ be a bijection corresponding to lexicographic order on the strings in P^k. That is, if $v, w \in P^k$, $0 \leq i \leq k-1$, $p, q \in P$, $v = r_r...r_i px$, $w = r_1...r_i qy$ and $p < q$, then $I(v) < I(w)$.

For $1 \leq a \leq N + 1$, define $\alpha_a : P^k \rightarrow \{0, 1\}$ by

$\alpha_a(w) =$ 0 if $I(w) < a$,
1 otherwise.

Note that $\alpha_1 = \gamma_1$ and $\alpha_{N+1} = \gamma_0$.

We claim that $\alpha_a \sim \alpha_{a+1}$ for all $a, 1 \leq a \leq N$. If so, then $\gamma_1 = \alpha_1 \sim \alpha_2 \sim ... \sim \alpha_{N+1} = \gamma_0$, so that $\gamma_1 \equiv \gamma_0$. Fix any $r \in P$. By Fact 1, $F(\gamma_1,r) = F(\gamma_0,r)$. However, by Fact 2, $F(\gamma_1,r) = 1$ and $F(\gamma_0,r) = 0$. This provides the needed contradiction.

It remains to prove the claim. Fix a, $1 \leq a \leq N$, and choose $r_1,...,r_k$ so that $I(r_1...r_k) = a$. By

assumption, $n-k \geq n-m \geq 3m+1-m = 2m+1 > 2$, so that there exist two distinct participants, r_{k+1} and r_{k+2}, in $P - \{r_1,...,r_k\}$. Assume without loss of generality that $r_{k+1} > r_{k+2}$ in the total order on P. We construct a k-round scenario σ with $L_\sigma \subseteq \{r_1,...,r_k\}$, in which $\sigma_{r_{k+1}} = \alpha_a$ and $\sigma_{r_{k+2}} = \alpha_{a+1}$.

Let $\sigma(w) =$ 0 if $w = r_1...r_i px$, where $0 \leq i \leq k$, $p \in P$, $x \in P^{0:k-i}$, and $p < r_{i+1}$,
1 otherwise.

We show that σ is a k-round scenario, with $L_\sigma \subseteq \{r_1,...,r_k\}$. Let $q \in P - \{r_1,...,r_k\}$, $p \in P$ and $w \in P^{0:k-1}$. We must show that $\sigma(wq) = \sigma(wqp)$. Now, $|wq| \leq k$, so thatwq is of the form $r_1...r_i s x$, where $i \leq k-1$, $s \in P$, and $s \neq r_{i+1}$. Then $\sigma(wq) =$ 0 if $s < r_{i+1}$,
1 otherwise,

$= \sigma(r_1...r_i sxp) = \sigma(wqp)$.

Next, we show that $\sigma_{r_{k+1}} = \alpha_a$. Let $w \in P^k$. Then $\alpha_a(w) =$ 0 if $l(w) < a$,
1 otherwise,

$= \sigma(wr_{k+1}) = \sigma_{r_{k+1}}(w)$.

Finally, we show that $\sigma_{r_{k+2}} = \alpha_{a+1}$. Let $w \in P^k$.Then $\alpha_{a+1}(w) =$ 0 if $l(w) \leq a$,
1 otherwise,

$= \sigma(wr_{k+2})$ since $r_{k+2} < r_{k+1}$,

$= \sigma_{r_{k+2}}(w)$.

□

Note that Theorem 3 (of the previous section) provides an upper bound on both the number of processors and the number of rounds. Thus, it demonstrates that both the lower bounds of Theorems 2 and 4 are tight.

7. Open Question

The most important question remaining involves the amount of communication and storage needed to assure interactive consistency. The algorithm in [PSL], which uses the minimum possible number of rounds, involves sending enormous amounts of information - approximately n^{m+2} values in v. We would like to know if this amount can be reduced, say to an amount polynomial in n and m (using either the minimum number, m + 1, of rounds, or perhaps a larger number of rounds). An algorithm using such a reduced amount of communication might be of considerably more practical value than the current algorithm.

Acknowledgements:

The authors thank Leslie Lamport, Michael Merritt and Eugene Stark for helpful discussions and suggestions about this manuscript.

References

[D] Dolev, D. The Byzantine Generals Strike Again. To appear.

[DW] Davies, D., and Wakerly, J. Synchronization and matching in redundant systems. *IEEE Trans. on Comptrs. C-27*, 6(June 1978), 531-539.

[L] Lamport, L. Using Time Instead of Timeout For Fault-Tolerant Distributed Systems. June 1981.

[LF] Lynch, Nancy A., and Fischer, Michael J. On Describing The Behavior and Implementation of Distributed Systems. *Theoretical Computer Science 13* 1981, pp. 17-43.

[LSP] Lamport, L., Shostak, R. and Pease, M. The Byzantine Generals Problem. Manuscript.

[PSL] Pease, M., Shostak, R. and Lamport, L. Reaching Agreement in the Presence of Faults. *JACM*, Vol. 27, No. 2, April 1980, pp. 228-234.