Fast Information Sharing in a Complete Network

V.S. Sunderam, Emory University

Peter Winkler*, Emory University and Bellcore

Abstract: We consider the problem of complete information dissemination among n autonomous processors in a fully connected distributed system. Initially, each processor possesses information not held by any other processor; it is required that all the processors obtain all the information in the shortest possible time. Messages are exchanged in discrete, synchronized rounds; message size is unlimited, but during a round, each processor may transmit messages to, or receive messages from, at most k other processors. We show that $\lceil \log_{\lambda(k)} n \rceil$ rounds are necessary for such an information exchange and that $\lceil \log_{\lambda(k)} n \rceil + 3$ are sufficient, where $\lambda(k) = (k+\sqrt{k^2+4})/2$. This settles in the affirmative a 10-year-old conjecture of Entringer and Slater; our lower bound is new even for the case $k = 1$.

1. Introduction

A fundamental requirement in distributed computing systems is the distribution of information known to each participant to every other participant. We consider distributed systems consisting of a set of n independent processors communicating by exchanging messages over a fully connected, point-to-point network. The network is synchronous and reliable, and is the only means of communication — there is no shared memory or central controller. Each processor is capable of transmitting messages to or receiving messages from a total of at most k other processors during one time unit or "round". In this paper, we present protocols that achieve complete information exchange and we prove that under the above conditions, these protocols operate within a few rounds of the minimum possible.

Information sharing plays a critical role in consensus protocols, where the global objective is the evaluation of a function or predicate whose arguments are distributed over the processing nodes. If the objective is to be achieved by replicating the computation at each node, then all the arguments must be made available to all the participants. Several classes of applications including distributed extrema finding, coordination of distributed checkpoints and some categories of election employ solutions of this nature. A particularly important application in current distributed memory

* research supported by ONR grant N00014 85-K-0769.

multiprocessor systems is barrier synchronization. In many parallel algorithms, all instances of the program arrive at a barrier almost synchronously; reducing the time required for barrier synchronization could contribute significantly to overall performance.

Information-exchange protocols have been analyzed in the literature from the points of view of time complexity, message complexity, and resilience to processor failures. For example, if simultaneous transmission and reception (from multiple senders) is permitted, consensus may be achieved in 1 round, with an exchange of $O(n^2)$ messages. In [9] Lakshman presents an algorithm based on finite projective planes which achieves consensus in two rounds and requires $O(n\sqrt{n})$ messages per round. This result has been generalized by Farrag [4] to the r rounds case which requires $O(n^{1+1/r})$ messages in each round.

From a different perspective, the impact of faults, failures, and asynchrony on information exchange has been studied extensively; representative work may be found in [1], [2], and [6].

When the objective is to distribute information held by everyone in as efficient a manner as possible, the process has most often been referred to in the mathematical literature as "gossiping." Gossiping has been studied extensively both with the objective of minimizing the number of messages and of minimizing the number of rounds; we refer the reader to the excellent survey article [7].

In Knödel [8], for example, the time complexity for consensus is considered under the constraint that each person may communicate (bi-directionally) with only one other person during a given round. In this case the minimum number of rounds is precisely $\lceil \log_2 n \rceil$ for even n and $\lceil \log_2 n \rceil + 1$ for odd n. Knödel's result is generalized by Schmitt [10] where k persons may be involved in a 'conference call'; the time required is then $\lceil \log_k n \rceil$ when k divides n, and $\lceil \log_k (n/(k-1)) \rceil + 1$ otherwise.

The object-model for early work in gossiping was humans making telephone calls, but as attention shifted to communication between processors it became natural to consider the case where communication is uni-directional. The natural case where the underlying network is a complete directed graph was addressed in Entringer and Slater [3] with the following results.

Suppose that in a given round, a given processor may either send messages to at most k other processors, or receive messages from at most k others. Then the minimum number of rounds required for complete dissemination among n processors lies between $\lceil \log_{k+1} n \rceil$ and $\lceil 2\log_{k+1} n \rceil$. In the k=1 and k=2 cases, Entringer and Slater were able to improve the upper bounds (but not the lower) to approximately $1.44\log_2 n$ and $1.25\log_3 n$ respectively. They then conjectured that for all k, the minimum number of rounds is $\beta_k \log_{k+1} n + O(1)$ where $\beta_k = \log_{(k+\sqrt{k^2+4})/2}(k+1)$.

In what follows, we show that Entringer and Slater were correct, and moreover, that a fairly simple set of protocols and a linear-algebraic lower bound argument can be made to agree within three rounds. In fact, the *lower* bound of $\lceil \beta_k \log_{k+1} n \rceil$ can be obtained even when conditions are relaxed so as to allow a processor simultaneously to transmit and receive, provided that it can deal with at most k messages at one time.

Our results permit us also to derive optimal protocols in the situation where information exchange is initiated by one processor and no other processor may transmit until it has been "awakened" by receiving a message.

NOTE: The authors have learned, since this paper was circulated in preprint form, that several of these results were independently obtained by Even and Monien and have appeared in their extended abstract [4].

2. The Protocols

Let the processors be $P_1, P_2, \ldots, P_n$ where the subscripts are taken modulo n, i.e., $P_0 = P_n$. At time $t = 0$, processor P_i has "knowledge set" $X_i = \{i\}$, that is, it begins only with its own initial information; after the last round of information exchange, each $X_i = \{1,2,\ldots,n\}$.

Let us now fix $k \geq 1$ and assume that n is even. For all k, our protocols begin with the same two not-very-efficient rounds, namely:

at time t=1, each even processor P_{2i} transmits to P_{2i+1}; and

at time t=2, each odd processor P_{2i+1} transmits back to P_{2i}.

Thus, at the end of the second round, we have both X_{2i} and X_{2i+1} equal to the set $\{2i, 2i+1\}$ which we think of also as the interval $[2i, 2i+1]$. Further, all of the following conditions hold and will continue to hold until some $X_i = \{1,\ldots,n\}$, that is, until some processor knows all:

(a) each X_i is an interval $[a_i, b_i]$, where a_i is even and b_i is odd. These intervals may go "around the corner"; for example, $[n-1, 2] = \{n-1, n, 1, 2\}$.

(b) for any i and j of the same parity, $a_i - a_j = b_i - b_j = i - j$. Thus, the knowledge sets are at all times circularly symmetric with respect to the even processors, and also with respect to the odd ones; let u be the amount of information known to any even processor (so that $u = b_i - a_i + 1$ for any even i) and v be the amount known to an odd processor.

(c) if $M(t) = \max(u, v)$ and $m(t) = \min(u, v)$ after round t, then

$$M(t) = \min\left(\frac{2+2\lambda}{\rho}\lambda^{t-2} - \frac{2+2\mu}{\rho}\mu^{t-2} \ , \ n \right)$$

and

$$m(t) = \min\left(\frac{2+2\lambda}{\rho}\lambda^{t-3} - \frac{2+2\mu}{\rho}\mu^{t-3} \ , \ n \right) = M(t-1) \ ,$$

where

$$\rho = \sqrt{k^2+4} \ , \ \lambda = \frac{k+\rho}{2} \ , \text{ and } \mu = \frac{k-\rho}{2} \ .$$

Let T be the least t for which $M(t-1) \geq n$; then after $T-1$ rounds either all the even processors or all the odd processors are fully informed. Thus at round T a repetition either of round 1 or of round 2 will complete the information exchange.

It remains, of course, to show that conditions (a), (b) and (c) can be preserved from round to round. Since $M(2) = m(2) = 2$ it is immediate that the conditions hold after round 2; assume that they hold after round $t-1$. We assume also that $M(t-1) = u \geq v$ at this point, that is, the even processors are at least as knowledgeable as the odds; the opposite case is the same.

In round t the even processors will transmit to the less-knowledgeable odd processors. Let P_i be an even processor, currently with knowledge $X_i = [a_i, b_i]$. From (b) we know that the set $\{b_j : j \text{ odd}\}$ is precisely the set of all odd numbers modulo n. It follows that for every integer s, the equation

$$1 + b_j + (s-1)u = a_i$$

has a unique odd solution $j = j_s$. (Explicit formulas for j_s in terms of i and s may be derived, but it seems no harder to generate these values recursively.) Processor P_i then sends its information to processors $P_{j_1}, P_{j_2}, \ldots, P_{j_k}$. (This could result in P_i sending more than one message to the same processor, wasting information. This happens just when $v + ku > n$ in which case the odd processors are about to learn everything.) In any case the recipient numbers $j_1, \ldots, j_k$ depend symmetrically on i; that is if $P_{i'}$ sends to $\{P_{j'_s}: s=1,\ldots,k\}$ then $j'_1 - j_1 = \cdots = j'_k - j_k = i' - i$. Thus each *odd* processor P_j plays the role of P_{j_s} precisely once for each s, and its new knowledge set will thus be

$$[a_j,\ b_j] \cup [b_j+1,\ b_j+u] \cup [b_j+u+1,\ b_j+2u] \cup \cdots \cup [b_j+(k-1)u+1,\ b_j+ku]$$
$$= [a_j,\ b_j+ku]\ .$$

We thus have that after round t, the odd processors will have increased the size of their knowledge sets from v to $\min(v+ku, n)$. Conditions (a) and (b) have been maintained, and to show (c) it suffices to demonstrate that $M(t)$ satisfies the recursion $M(t) = m(t-1) + kM(t-1) = M(t-2) + kM(t-1)$ as long as $M(t) \le n$; but this is immediate since λ and μ are roots of the equation $x^2 = 1 + kx$.

To determine the number of rounds required by the protocol notice that irrespective of the value of k, the root μ lies between -1 and 0 and the coefficient $(2+2\mu)/\rho$ lies between -1/2 and 0. Thus the second term in the expression for $M(t)$ is less than 1/2 in absolute value, and we may rewrite

$$M(t) = \min\left(\left\lfloor \frac{2+2\lambda}{\rho}\lambda^{t-2} + \frac{1}{2} \right\rfloor,\ n\right)\ .$$

It follows that T steps suffice iff $n \le M(T-1) = m(T) = (2+2\lambda)\lambda^{T-3}/\rho + 1/2$, i.e. iff

$$T \ge \left\lceil \log_\lambda\left(n - \frac{1}{2}\right) - \log_\lambda\left(\frac{2+2\lambda}{\rho}\right) + 3 \right\rceil\ .$$

In Fig. 1 below, the protocol for k=1 and n=16 is illustrated; in Fig. 2, the protocol for k=3 and n=26. In each case the box in column labeled "t" and row labeled "P_i" lists the processors which *transmit* to P_i during round t; the arrow points to the right for even-to-odd transmission, to the left for odd-to-even. (If P_i transmits rather than receives during round t, the box is left blank.) The unlabeled columns indicate knowledge sets between rounds.

Processor	Actions during each round and status of knowledge sets between rounds														
↓	1		2		3		4		5		6		7		
P_1	[1,1]	$P_{16}\rightarrow$	[16,1]		[16,1]	$P_{14}\rightarrow$	[14,1]		[14,1]	$P_8\rightarrow$	[8,1]		[8,1]	$P_2\rightarrow$	[8,7]
P_2	[2,2]		[2,2]	$\leftarrow P_3$	[2,3]		[2,3]	$\leftarrow P_7$	[2,7]		[2,7]	$\leftarrow P_1$	[8,7]		[8,7]
P_3	[3,3]	$P_2\rightarrow$	[2,3]		[2,3]	$P_{16}\rightarrow$	[16,3]		[16,3]	$P_{10}\rightarrow$	[10,3]		[10,3]	$P_4\rightarrow$	[10,9]
P_4	[4,4]		[4,4]	$\leftarrow P_5$	[4,5]		[4,5]	$\leftarrow P_9$	[4,9]		[4,9]	$\leftarrow P_3$	[10,9]		[10,9]
P_5	[5,5]	$P_4\rightarrow$	[4,5]		[4,5]	$P_2\rightarrow$	[2,5]		[2,5]	$P_{12}\rightarrow$	[12,5]		[12,5]	$P_6\rightarrow$	[12,11]
P_6	[6,6]		[6,6]	$\leftarrow P_7$	[6,7]		[6,7]	$\leftarrow P_{11}$	[6,11]		[6,11]	$\leftarrow P_5$	[12,11]		[12,11]
P_7	[7,7]	$P_6\rightarrow$	[6,7]		[6,7]	$P_4\rightarrow$	[4,7]		[4,7]	$P_{14}\rightarrow$	[14,7]		[14,7]	$P_8\rightarrow$	[14,13]
P_8	[8,8]		[8,8]	$\leftarrow P_9$	[8,9]		[8,9]	$\leftarrow P_{13}$	[8,13]		[8,13]	$\leftarrow P_7$	[14,13]		[14,13]
P_9	[9,9]	$P_8\rightarrow$	[8,9]		[8,9]	$P_6\rightarrow$	[6,9]		[6,9]	$P_{16}\rightarrow$	[16,9]		[16,9]	$P_{10}\rightarrow$	[16,15]
P_{10}	[10,10]		[10,10]	$\leftarrow P_{11}$	[10,11]		[10,11]	$\leftarrow P_{15}$	[10,15]		[10,15]	$\leftarrow P_9$	[16,15]		[16,15]
P_{11}	[11,11]	$P_{10}\rightarrow$	[10,11]		[10,11]	$P_8\rightarrow$	[8,11]		[8,11]	$P_2\rightarrow$	[2,11]		[2,11]	$P_{12}\rightarrow$	[2,1]
P_{12}	[12,12]		[12,12]	$\leftarrow P_{13}$	[12,13]		[12,13]	$\leftarrow P_1$	[12,1]		[12,1]	$\leftarrow P_{11}$	[2,1]		[2,1]
P_{13}	[13,13]	$P_{12}\rightarrow$	[12,13]		[12,13]	$P_{10}\rightarrow$	[10,13]		[10,13]	$P_4\rightarrow$	[4,13]		[4,13]	$P_{14}\rightarrow$	[4,3]
P_{14}	[14,14]		[14,14]	$\leftarrow P_{15}$	[14,15]		[14,15]	$\leftarrow P_3$	[14,3]		[14,3]	$\leftarrow P_{13}$	[4,3]		[4,3]
P_{15}	[15,15]	$P_{14}\rightarrow$	[14,15]		[14,15]	$P_{12}\rightarrow$	[12,15]		[12,15]	$P_6\rightarrow$	[6,15]		[6,15]	$P_{16}\rightarrow$	[6,5]
P_{16}	[16,16]		[16,16]	$\leftarrow P_1$	[16,1]		[16,1]	$\leftarrow P_5$	[16,5]		[16,5]	$\leftarrow P_{15}$	[6,5]		[6,5]

Fig. 1

Processor	*Actions during each round and status of knowledge sets between rounds*										
↓	*1*			*2*		*3*		*4*		*5*	
P_1	[1,1]	$P_{26}\rightarrow$	[26,1]		[26,1]	$P_{20}, P_{22}, P_{24}\rightarrow$	[20,1]		[20,1]	$P_2\rightarrow$	[2,1]
P_2	[2,2]		[2,2]	$\leftarrow P_3$	[2,3]		[2,3]	$\leftarrow P_{11}, P_{19}, P_1$	[2,1]		[2,1]
P_3	[3,3]	$P_2\rightarrow$	[2,3]		[2,3]	$P_{22}, P_{24}, P_{26}\rightarrow$	[22,3]		[22,3]	$P_4\rightarrow$	[4,3]
P_4	[4,4]		[4,4]	$\leftarrow P_5$	[4,5]		[4,5]	$\leftarrow P_{13}, P_{21}, P_3$	[4,3]		[4,3]
P_5	[5,5]	$P_4\rightarrow$	[4,5]		[4,5]	$P_{24}, P_{26}, P_2\rightarrow$	[24,5]		[24,5]	$P_6\rightarrow$	[6,5]
P_6	[6,6]		[6,6]	$\leftarrow P_7$	[6,7]		[6,7]	$\leftarrow P_{15}, P_{23}, P_5$	[6,5]		[6,5]
P_7	[7,7]	$P_6\rightarrow$	[6,7]		[6,7]	$P_{26}, P_2, P_4\rightarrow$	[26,7]		[26,7]	$P_8\rightarrow$	[8,7]
P_8	[8,8]		[8,8]	$\leftarrow P_9$	[8,9]		[8,9]	$\leftarrow P_{17}, P_{25}, P_7$	[8,7]		[8,7]
P_9	[9,9]	$P_8\rightarrow$	[8,9]		[8,9]	$P_2, P_4, P_6\rightarrow$	[2,9]		[2,9]	$P_{10}\rightarrow$	[10,9]
P_{10}	[10,10]		[10,10]	$\leftarrow P_{11}$	[10,11]		[10,11]	$\leftarrow P_{19}, P_1, P_9$	[10,9]		[10,9]
P_{11}	[11,11]	$P_{10}\rightarrow$	[10,11]		[10,11]	$P_4, P_6, P_8\rightarrow$	[4,11]		[4,11]	$P_{12}\rightarrow$	[12,11]
P_{12}	[12,12]		[12,12]	$\leftarrow P_{13}$	[12,13]		[12,13]	$\leftarrow P_{21}, P_3, P_{11}$	[12,11]		[12,11]
P_{13}	[13,13]	$P_{12}\rightarrow$	[12,13]		[12,13]	$P_6, P_8, P_{10}\rightarrow$	[6,13]		[6,13]	$P_{14}\rightarrow$	[14,13]
P_{14}	[14,14]		[14,14]	$\leftarrow P_{15}$	[14,15]		[14,15]	$\leftarrow P_{23}, P_5, P_{13}$	[14,13]		[14,13]
P_{15}	[15,15]	$P_{14}\rightarrow$	[14,15]		[14,15]	$P_8, P_{10}, P_{12}\rightarrow$	[8,15]		[8,15]	$P_{16}\rightarrow$	[16,15]
P_{16}	[16,16]		[16,16]	$\leftarrow P_{17}$	[16,17]		[16,17]	$\leftarrow P_{25}, P_7, P_{15}$	[16,15]		[16,15]
P_{17}	[17,17]	$P_{16}\rightarrow$	[16,17]		[16,17]	$P_{10}, P_{12}, P_{14}\rightarrow$	[10,17]		[10,17]	$P_{18}\rightarrow$	[18,17]
P_{18}	[18,18]		[18,18]	$\leftarrow P_{19}$	[18,19]		[18,19]	$\leftarrow P_1, P_9, P_{17}$	[18,17]		[18,17]
P_{19}	[19,19]	$P_{18}\rightarrow$	[18,19]		[18,19]	$P_{12}, P_{14}, P_{16}\rightarrow$	[12,19]		[12,19]	$P_{20}\rightarrow$	[20,19]
P_{20}	[20,20]		[20,20]	$\leftarrow P_{21}$	[20,21]		[20,21]	$\leftarrow P_3, P_{11}, P_{19}$	[20,19]		[20,19]
P_{21}	[21,21]	$P_{20}\rightarrow$	[20,21]		[20,21]	$P_{14}, P_{16}, P_{18}\rightarrow$	[14,21]		[14,21]	$P_{22}\rightarrow$	[22,21]
P_{22}	[22,22]		[22,22]	$\leftarrow P_{23}$	[22,23]		[22,23]	$\leftarrow P_5, P_{13}, P_{19}$	[22,21]		[22,21]
P_{23}	[23,23]	$P_{22}\rightarrow$	[22,23]		[22,23]	$P_{16}, P_{18}, P_{20}\rightarrow$	[16,23]		[16,23]	$P_{24}\rightarrow$	[24,23]
P_{24}	[24,24]		[24,24]	$\leftarrow P_{25}$	[24,25]		[24,25]	$\leftarrow P_7, P_{15}, P_{23}$	[24,23]		[24,23]
P_{25}	[25,25]	$P_{24}\rightarrow$	[24,25]		[24,25]	$P_{18}, P_{20}, P_{22}\rightarrow$	[18,25]		[18,25]	$P_{26}\rightarrow$	[26,25]
P_{26}	[26,26]		[26,26]	$\leftarrow P_1$	[26,1]		[26,1]	$\leftarrow P_9, P_{17}, P_{25}$	[26,25]		[26,25]

Fig. 2

We are now in a position to obtain an upper bound for the number of rounds require for complete exchange of information.

Theorem 1. Complete information exchange can be accomplished within $\lceil \log_{(k+\sqrt{k^2+4})/2} n \rceil + 3$ rounds, for any k and n.

Proof. Note first that since $k < \lambda = (k+\sqrt{k^2+4})/2 < k+2$, we have $(2+2\lambda)/\rho > 2$ and therefore $\log_\lambda((2+2\lambda)/\rho) > 0$. It therefore follows from (1) above that $\lceil \log_\lambda n \rceil + 3$ rounds suffice when n is even.

When $k > 1$ the odd case presents no problem. Processors P_1 through P_{n-1} follow the above protocol; processor P_n transmits to P_1 at time t=1 (when P_{n-1} is also doing so) and thus after the second round, P_n's initial information is found wherever P_1's is. P_n is idle until just prior to the the final round of the protocol; at that time half of the first $n-1$ processors know everything (including P_n's initial information), and one of them can inform P_n at the same time that its regular final-round message is sent.

In fact, an additional round will sometimes be saved by running the protocol only on processors $P_1, \ldots, P_q$ where q is the smallest even number for which $(n-q)/(k-1) \le q/2$, allowing the remaining processors to input during round 1 and receive during the final round.

When k=1 odd values of n are more troublesome, but fortunately the large value of the term $(2+2\lambda)/\rho = \log_{(1+\sqrt{5})/2}(3+\sqrt{5})/\sqrt{5} = 1.768\ldots$ provides additional play. We proceed as follows.

Let $q = 2\lceil n/4 \rceil$, so that q is the smallest even number above $n/2$. First, each P_{q+i}, $1 \le i \le n-q$, transmits to P_i during round 1. Next, processors P_1 through P_q proceed with our even-n protocol; lastly, each P_i, $1 \le i \le n-q$, transmits back to P_{q+i}. This requires $T+2$ rounds where T is the number of rounds used by our protocol for q processors. For example, if $n \equiv 3 \bmod 4$, we have

$$T = \left\lceil \log_\lambda\left(\frac{n+1}{2} - \frac{1}{2}\right) - \log_\lambda \frac{3+\sqrt{5}}{\sqrt{5}} + 3 + 2 \right\rceil$$
$$= \lceil \log_\lambda n + 1.79\ldots \rceil \ .$$

Formally, our protocol uses 0 rounds for n=1, 2 for n=2 and 4 for n=3; when n>3 the number of rounds used is precisely

$$\lceil \log_\lambda(n-(1/2)) - \log_\lambda \alpha \rceil + 3 \ , \text{ for } n \text{ even;}$$
$$\lceil \log_\lambda(n+1) + \log_\lambda 2\alpha \rceil + 5 \ , \text{ for } n \equiv 1 \bmod 4 \ ; \text{ and}$$
$$\lceil \log_\lambda n + \log_\lambda 2\alpha \rceil + 5 \ , \text{ for } n \equiv 3 \bmod 4$$

where $\lambda = (1+\sqrt{5})/2$ and $\alpha = (3+\sqrt{5})/\sqrt{5}$.

From this it follows that the number of rounds used in the k=1 case lies always between $\lceil \log_\lambda n \rceil$ and $\lceil \log_\lambda n \rceil + 2$. When k=2 the same statement applies (since then $\log_\lambda((2+2\lambda)/\rho) = 1$) so that in these two cases Theorem 1 is proved with a bit to spare. □

3. The Lower Bound

The purpose of this section is to prove the following theorem, which places our protocols within 3 rounds of best possible:

Theorem 2. Complete information exchange requires at least $\lceil \log_\lambda n \rceil$ rounds, where $\lambda = (k+\sqrt{k^2+4})/2$. This bound remains true even if a processor may both transmit and receive during a single round, provided the sum of the number of messages it sends and the number it receives is at most k.

Proof. The idea of the proof is, in principle, very simple. Fix n and k and let $X_1, \ldots, X_n$ be the knowledge sets at some point during an information exchange protocol.

Let x_i be the cardinality of X_i; the "knowledge vector" $\bar{x} = (x_1, \ldots, x_n)$ provides a crude (but, it turns out, surprisingly effective) measure of the progress of information exchange.

For, suppose we can show that during any round, the norm

$$|\bar{x}| = \left(\sum_{i=1}^{n} x_i^2 \right)^{1/2}$$

can increase by at most a factor of λ. Then since $\bar{x}$ begins at $(1,1,\ldots,1)$ which has norm $\sqrt{n}$, and finishes at $(n,n,\ldots,n)$ which has norm $n^{3/2}$, the number of rounds required is at least $\lceil \log_\lambda (n^{3/2}/\sqrt{n}) \rceil = \lceil \log_\lambda n \rceil$.

Thus, let us fix a number t, letting $\bar{x}$ be the knowledge vector just prior to round t, and $\bar{y}$ just after. For each i, $1 \le i \le n$, let $S_i = \{j: P_i \text{ sends to } P_j\}$ and let $R_i = \{j: P_i \text{ receives from } P_j\}$. Note that our conditions imply that for each i, $|S_i| + |R_i| \le k$, and

$$y_i \le x_i + \sum_{j \in R_i} x_j \ .$$

We are reduced to showing that under these circumstances, we can prove

Lemma 1. $|\bar{y}| \le \lambda |\bar{x}|$.

Let the $n \times n$ matrix $A = (a_{i,j})$ be defined by $a_{i,j} = 1$ if $j \in R_i$ and $a_{i,j} = 0$ otherwise. Then we have

$$\bar{y} \le \bar{x} + A\bar{x} = (I+A)\bar{x} \ .$$

Further, since for each i we have $\sum_j a_{i,j} = |R_i|$ and $\sum_j a_{j,i} = |S_i|$, A satisfies the condition

$$\sum_{j=1}^{n} (a_{i,j} + a_{j,i}) \le k \quad \text{for each } i \ , \ 1 \le i \le n \ .$$

Lemma 1 thus follows from the following more abstract statement:

Lemma 2. Let $B = (b_{i,j})$ be any non-negative-integer-valued $n \times n$ matrix satisfying $\sum_j (b_{i,j} + b_{j,i}) \le k$, and let $\bar{x}$ be any non-negative vector in $\mathbf{R}^n$ of norm 1. Then $|\bar{x} + B\bar{x}| \le \lambda$.

Lemma 2 is quite a strong statement if translated back into message-sending terms. In effect, it says that the bound of Theorem 2 holds even when

(a) duplicated information is multiply-counted---in effect, the protocol is deemed completed when each processor has accumulated n pieces of original information, even if several are the same;

(b) a processor may send several messages to the same destination---and have them all add to the recipients "knowledge";

(c) a processor may send messages to itself, although each such message counts both as sent and received.

Thus, in the remarkable world of Lemma 2, a processor may treble its own knowledge by sending two messages to itself (provided $k \geq 4$). Worse, it will be seen that for *general* $\bar{x}$, the matrices B resulting in largest $|\bar{x} + B\bar{x}|$ have some entries equal to k and thus represent nonsensical protocols (except in the k=1 case). The reason that our protocols nonetheless approach the bound is that the knowledge vectors which arise in them have many equal coordinates.

Lemma 2 is actually quite easy to prove in the case k=1, because then we may obtain, by suitably permuting the coordinates of the output vector $\bar{y} = (I+B)\bar{x}$, a version of B such that the matrix $C = I+B$ has a basis of eigenvectors and has largest eigenvalue λ; then $|C\bar{z}| \leq \lambda|\bar{z}|$ for any $\bar{z}$ (with equality only when $\bar{z}$ is an eigenvector for λ). For example, when n=5 we can have

$$C = \begin{bmatrix} k & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & k & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

which diagonalizes to

$$\begin{bmatrix} \lambda & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where $\mu = (k-\sqrt{k^2+4})/2$; we have left in the k's rather than replace them by 1's in order to help clarify what follows.

Unfortunately, some combinatorial manipulation seems to be necessary when $k>1$ in order to demonstrate that when $|C\bar{x}|$ is maximized, C can be assumed to be in something like the form illustrated above.

We begin by choosing $\bar{x}$ and B so as to satisfy the conditions of Lemma 2, and so as to maximize $\beta = |\bar{y}| = |C\bar{x}| = |(I+B)\bar{x}|$. That such a choice is possible follows from the following facts:

(1) $|C\bar{x}|$ is trivially bounded by $\sqrt{n(k+1)}$,

(2) there are only finitely many choices for B, and

(3) $|C\bar{x}|$ depends continuously on a variable ($\bar{x}$) whose domain is compact.

Further, we may assume without loss of generality that

(a) $x_1 \geq x_i$ for each $i>1$,

(b) $y_2 \geq y_i$ for each $i>2$, and

(c) the entry $b_{2,1}$ is maximized over all B for which all of the above hold.

We now assume $\beta > \lambda$ with the intent of deriving a contradiction.

Let us consider the matrix $M = C^tC$, where C^t is the transpose of C. Since M is real, symmetric and non-negative, it has a basis of eigenvectors and a full set of real eigenvalues of which the largest is some positive real γ. We claim that our vector $\bar{x}$ from above is an eigenvector for M corresponding to the eigenvalue γ, and moreover that $\beta = \sqrt{\gamma}$.

For, let $\bar{z}$ be an arbitrary unit vector; then $|C\bar{z}|^2 = (C\bar{z})^t(C\bar{z}) = \bar{z}^t(C^tC\bar{z}) \leq |\bar{z}^t||M\bar{z}| = |M\bar{z}| \leq \gamma|\bar{z}| = \bar{z}$, so that $|C\bar{z}| \leq \sqrt{\gamma}$ with equality just when $\bar{z}$ is an eigenvector of M corresponding to γ. Thus if there *is* such an eigenvector which is non-negative, $\bar{x}$ must be one in order to maximize $|C\bar{x}|$. But if $\bar{z}$ is an M-eigenvector for γ then so is the result of replacing each coordinate of $\bar{z}$ by its absolute value, since such an operation cannot decrease $|M\bar{z}|$.

Now let $\bar{w} = M\bar{x} = \gamma\bar{x}$ so that

$$w_1 = y_1 + \sum_i b_{i,1}y_i$$
$$= x_1 + \sum_j b_{1,j}x_j + \sum_i a_{i,j}(x_i + \sum_j a_{i,j}x_j)$$
$$= \gamma x_1 \ .$$

We have $\beta > \lambda$ (by assumption), and

$$\lambda^2 = \frac{1}{4}(k^2 + 2k\sqrt{k^2+4} + (k^2+4)) > \frac{1}{4}(k^2 + 2k^2 + k^2 + 4) = k^2 + 1 \ .$$

Thus $\gamma = \beta^2 > k^2 + 1$. But then $\sum_i b_{i,j}$ must be precisely k ; else, remembering that x_1 is the largest coordinate of $\bar{x}$, we would have

$$w_1 \geq x_1 + x_1 + (k-1)(k+1)x_1 = (k^2+1)x_1 < \gamma s_1 \ ,$$

a contradiction.

We must therefore have $\sum_i b_{1,i} = 0$. (In message terms, we have demonstrated that the best-informed processor must send a full complement of k messages and receive none.) Thus

$$w_1 = x_1 + \sum_i b_{i,1}y_i > (k^2+1)x_1$$

from which we may conclude that for some i , necessarily $\neq 1$, $y_i > kx_1$; in particular

$y_2 > kx_1$, and thus $\sum_j b_{2,j}$ must be equal to k (i.e. processor P_2 receives k messages). We now claim that the entry $b_{2,1}$ must be precisely k.

If not, then there must be a $j>2$ for which $b_{2,j} > 0$ and also an $i>2$ for which $b_{i,1} > 0$; that is, P_2 has received a message from some P_j other than P_i, and consequently one of the k messages from P_1 has gone to some P_j other than P_2. Not surprisingly we can switch these two messages without loss, as follows.

Let B' be a matrix identical to B except that

$$b'_{2,1} = b_{2,1} + 1 ,$$
$$b'_{2,j} = b_{2,j} - 1 ,$$
$$b'_{i,1} = b_{i,1} - 1 ,$$

and

$$b'_{i,j} = b_{i,j} + 1 .$$

If we set $\bar{y}' = B'\bar{x}$ then

$$|\bar{y}'|^2 - |\bar{y}|^2 = (y'_2)^2 + (y'_i)^2 - y_2^2 - y_i^2$$
$$= (y_2 + x_1 - x_j)^2 + (y_i - x_1 + x_j)^2 - y_2^2 - y_i^2$$
$$= 2(x_1 - x_j)^2 + 2(y_2 - y_i)(x_1 - x_j) \geq 0 .$$

Thus $|\bar{y}'| \geq |\bar{y}|$ and $y'_2 \geq y_2$ and $b'_{2,1} > b_{2,1}$; consequently criterion (c) above should have preferred the pair $(B',\bar{x})$ to the pair $(B,\bar{x})$, a contradiction.

We have thus shown that $b_{2,1} = k$, i.e. that P_1 has sent all k of his messages to P_2. In matrix terms,

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} ,$$

and therefore

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix}^t \begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} k^2+1 & k \\ k & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \gamma \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} .$$

However, the matrix

$$\begin{bmatrix} k^2+1 & k \\ k & 1 \end{bmatrix}$$

has eigenvalues $(k^2 + 2 + k\sqrt{k^2+4})/2 = \lambda^2$ and $(k^2 + 2 - k\sqrt{k^2+4})/2 = \mu^2$, of which λ^2 is the larger. Thus $\gamma \leq \lambda^2$ after all, therefore $\beta \leq \lambda$, a contradiction which proves Lemma 2. Lemma 1 and Theorem 2 have thus also been proved. □

4. Initiation

We now consider complete information exchange in the case where, in addition to the requirements previously studied, the protocol must be initiated by one of the processors. Then no other processor may transmit a message until it has been alerted by a reception.

In this case our protocol proceeds as follows. First, the initiating processor wakes up k others, then each of the k+1 alert processors wakes k others, etc. until at least half the processors are awake. This requires $r = \lceil \log_{k+1} n/2 \rceil$ rounds to accomplish.

Then we merely employ one of the protocols from §2 above; since none of them starts with more than half the processors transmitting, we need only arrange matters so that the transmitting processors are chosen from among those that are now awake.

The number of rounds used is thus bounded, in view of Theorem 1, by

$$r + \lceil \log_\lambda (n/2) \rceil + 3 \ .$$

To see that we cannot do much better than this, observe that no matter what takes place in the early stages, after r rounds the number m of processors which have not yet *transmitted* is at least $n/2$. Let us now consider only knowledge of the initial information from these m processors in computing our knowledge vector; then after round r the knowledge vector $\bar{x}$ consists of m 1's and $n-m$ 0's, and after the final round, $\bar{x}$ must be all m's.

It now follows from Lemma 1 of §3 that at least

$$\lceil \log_\lambda (m\sqrt{n}/\sqrt{m}) \rceil \geq \lceil \log_\lambda n - \log_\lambda \sqrt{2} \rceil$$

additional rounds are required to complete the information exchange.

Since $\log_\lambda 2 - \log_\lambda \sqrt{2} < 1$ for all k, we may use Theorems 1 and 2 together to bring our initiation protocols within 4 of best possible.

Theorem 3. The minimum number of rounds required for complete information exchange in the presence of the initiation requirement is at least T and at most T+4, where

$$T = \lceil \log_{k+1} (n/2) \rceil + \lceil \log_\lambda (n/\sqrt{2}) \rceil \ .$$

REFERENCES

[1] Dolev, D., C. Dwork and L. Stockmeyer, "On the Minimal Synchronism Needed for Distributed Consensus", *J. ACM*, Vol. 34, pp. 77-97, Jan. 1987.

[2] Dwork, C., N. Lynch and L. Stockmeyer, "Consensus in the Presence of Partial Synchrony", *J. ACM*, Vol. 35, pp. 288-323, April 1988.

[3] Entringer, R.C. and P.J. Slater, "Gossips and Telegraphs", *J. Franklin Inst.*, Vol 307, pp. 353-360, June 1979.

[4] Even, S. and B. Monien, "On the Number of Rounds Necessary to Disseminate Information (extended abstract)", in *Proc. ACM Symposium on Parallel Algorithms and Architectures*, Santa Fe NM, June 1989.

[5] Farrag, A.A. and R. J. Dawson, "On Designing Efficient Consensus Protocols", in *Proc. IFIP WG 10.3 Working Conf. on Dist. Proc.*, pp. 413-427, October 1987.

[6] Fischer, M.J., N. A. Lynch and M. S. Paterson, "Impossibility of Distributed Consensus with One Faulty Process", *J. ACM*, Vol. 32, pp. 374-382, Apr. 1985.

[7] Hedetniemi, S.M., S. T. Hedetniemi and A. L. Liestman, "A Survey of Gossiping and Broadcasting in Communication Networks", *Networks*, Vol. 18, pp. 319-349, 1988.

[8] Knödel, W., "New Gossips and Telephones", *Discr. Math.*, Vol. 13, pp. 95, 1975.

[9] Lakshman, T.V. and A. K. Agarwala, "Efficient Decentralized Consensus Protocols", *IEEE Trans. Software Eng.*, Vol. SE-12, pp. 600-607, May 1986.

[10] Schmitt, P., "Spreading Information by Conferences", *Discr. Math.*, Vol. 15, pp. 305-306, 1976.

Authors' Addresses:

V.S. Sunderam, Dept. of Mathematics and Computer Science, Emory University, Atlanta GA 30322, U.S.A.
telephone: 404-727-5926; e-mail: vss@mathcs.emory.edu

Peter Winkler, Bellcore MRE 2L335, 445 South ST., Morristown NJ 07960-1910, U.S.A.
telephone: 201-829-5086; e-mail: pw@bellcore.com