The Expressiveness of Silence: Optimal Algorithms for Synchronous Communication of Information

Una-May O'Reilly and Nicola Santoro

SCS-TR-195, OCTOBER 1991

School of Computer Science, Carleton University
Ottawa, Canada, KIS 5B6

THE EXPRESSIVENESS OF SILENCE

Optimal Algorithms for Synchronous Communication of Information

(preliminary version)

Una-May O'Reilly and Nicola Santoro
School of Computer Science
Carleton University
Ottawa, K1S 5B6
Canada

Abstract: We establish a lower bound on the trade-off between time and bit complexity for two-party communication in synchronous networks. We prove that the bound is tight by presenting a protocol whose bit-time complexity matches the one expressed by the lower bound. The proposed algorithm is globally optimal (i.e., not only in the average and worst case). Similar results are derived when transmissions are subject to corruptions. Applications of the results to transforming an asynchronous distributed algorithm into a synchronous one are discussed.

1. INTRODUCTION

The process of efficiently and accurately communicating information along a link in a communication network is typically taken for granted (as a low level primitive) in the design of distributed algorithms. This process does, however, obey a set of rules (the *two-party communication protocol*) whose choice can greatly affect the overall performance of the distributed algorithms or protocols employed in the system. Most of the research activities on this subject have been carried out within coding theory and have focused on determining, for the information to be communicated, encoding and decoding schemes which satisfy some specific constraints (e.g., error-detection, error correction, etc.) or optimize some performance parameters (e.g., average number of bits, etc.). Related investigations include transmission of sequences over unreliable channels both in synchronous (e.g., [Aho, Ullman, Yannakakis-79], [Aho et al-82]) and asynchronous systems (e.g., [Lynch, Mansour, Fekete-88], [Wang, Zuck-89]), cryptographic protocols (e.g. [Godwasser et al -88]), and the communication complexity of evaluating specific functions (e.g. [Hajnal, Maass, Túran-88], [Yao-79], [Fleischer, Jung, Mehlhorn-90]).

An underlying assumption in all of this work is that transmission of bits is the only mechanism for communicating information; in other words, the*transmission* alphabet (i.e., {0,1}) and the *communication* alphabet coincide. This assumption is valid for *asynchronous* networks because transmission delays are finite but unpredictable. The situation, however, is different in *fully synchronous* networks where the transmission alphabet is still binary but the communication alphabet is ternary.

A fully synchronous network is defined by two basic properties: synchronized local clocks and bounded transmission delays. *Synchronized local clocks* refers to the fact that in a fully synchronous network all local clocks "tick" simultaneously (although they might not all sign the same absolute time); and *bounded transmission delays* refers to the fact that (in absence of failures) there exists a (known) bound on the time required for a bit to be transmitted on a link and processed at its destination. Because of these two properties, it is possible to redefine the *clock time* so that, in absence of failures, if a bit is sent on a link at local clock time t, it is received and processed at its destination at clock time t+1 (sender's time). As a consequence, in a fault-free system, if no bit is received at local clock time t+1, then none was transmitted at clock time t. Hence, absence of transmission (or *silence*) is detectable and can be used to convey information. In other words, in a fully synchronous network, the communication alphabet is ternary (i.e., C={0,1,"silence"}) while the transmission alphabet is still binary (i.e., T={0,1}).

An obvious consequence of this fact is that no more than $\log_3 x$ bits (instead of $\log_2 x$) need to be transmitted to communicate a positive integer x. (Note that delimiting the start and end of this communication becomes an issue.)

In general, silence may be exploited more effectively then just using it as another symbol in the communication alphabet. For example, it is not difficult to see that x can be communicated by transmitting at most $1+\lceil(\log_2 x)/2\rceil$ bits: choose the symbol (say "1") occurring least frequently in the bit string corresponding to x; transmit this symbol; communicate the string sequentially, using a bit transmission for each occurrence of "1" and silence for "0". Note that less than $\log_3 x$ bits will be transmitted (for x>280), but the communication will require more than $\log_3 x$ time units.

Associated with any synchronous protocol for two-party communication are two related cost measures: the total number of bits transmitted and the total number of clock ticks elapsed during the communication. The study of the two-party communication problem in synchronous networks is really the study of the trade-off between time and bits to answer the following question: how truly expressive is silence? The goal is to derive provably efficient (and possibly optimal) protocols.

Many encoding-decoding schemes and communication protocol for this problem can be devised, each offering a different trade-off between time and bits. For example, it is well known that any positive integer x can be communicated transmitting only two bits in time linear in x. On the other hand, using only one additional bit, the time complexity can be reduced to sublinear (e.g. [Santoro-90]). An encoding-decoding mechanism which uses k+3 bits and $kx^{1/k}$+l.o.t. time in the worst case to communicate an arbitrary positive integer x has been recently proposed [Schmeltz-90]. Some techniques will be briefly discussed in Section 2. It should be mentioned that any solution to the two-party communication problem can be directly applied to the *asynchronous-to-synchronous* conversion of distributed algorithms [Santoro-90]; applications to the *election* problem in synchronous rings have been studied in [Bodlaender, Tel-89] and [Schmeltz-90]. Unfortunately, no lower-bounds were known on this problem, making it difficult to evaluate the efficiency of a proposed solution.

In this paper, we investigate the two-party communication problem both in fault-free networks and in networks where transmissions are subject to *bit flips*. We establish tight bounds on the trade-off between time and bit complexity in both cases. In particular, for each type of network:

1. We establish a lower-bound on the trade-off between time and bit complexity. The lower-bound holds even when the universe (from which the information to be communicated is drawn) is *finite*; thus, it is also a lower-bound for the case of a countable but infinite universe.

2. We present a solution algorithm (i.e., an encoding-decoding scheme and a communication protocol) which is optimal at any point of the trade-off. The algorithm is optimal in a strong sense (*globally optimal*): it matches the lower-bound for every element of the universe (within a bijection) and at any point of the trade-off. Furthermore, the proposed algorithm solves the problem even when the universe is countable but *infinite*. The solution for networks where transmissions are subject to *bit flips* tolerates any number of corruptions.

Note that results 1. and 2. above imply that the existence and knowledge of a bound on the size of the universe does not affect the bit-time trade-off for the two-party communication problem.

The paper is organized as follows. In the next section, basic definitions are presented and some (known and new) protocols are described to provide the reader with an insight into the nature of the problem. In section 3, two distinct lower bounds are established on the trade-off between time and bits, depending on whether or not the value of transmitted bits can be used to convey information; these two cases correspond to networks where transmissions are subject to bit flips or are fault-free, respectively. In section 4, for both cases, we present protocols for the two-party communication problem which are globally optimal. We conclude with a discussion of applications which can use these protocols.

2. DEFINITIONS AND EXISTING TECHNIQUES

2.1 DEFINITIONS

Given a countable universe U, the *two-party communication problem* for U (denoted by TPC(U)), is the problem of communicating without ambiguity arbitrary elements of U using any combination of bit transmissions and silence. Without loss of generality, let U be a set of consecutive positive integers. A *quantum of silence* (or, simply, quantum) is the number of clock ticks between the transmission of two consecutive bits (the quantum is zero if bits are sent at two consecutive clock ticks).

We consider two sub-problems of TPC(U):

$TPC^1(U)$: the value of the bits can not be exploited by the protocol. Bits simply delimit quanta of silence. A solution protocol for $TPC^1(U)$ is therefore resilient to bit flips.

$TPC^2(U)$: the value of the bits can be used to convey information. A solution protocol for $TPC^2(U)$ therefore assumes a fault-free network.

Given a positive integer $b \geq 2$, let TPC(U,b) denote the problem of communicating without ambiguity any element of U transmitting exactly b bits; analogously, let $TPC^1(U,b)$ and $TPC^2(U,b)$ denote corresponding subproblems of TPC(U,b). In the following, b_i will denote a bit and q_i will denote a quantum; the subscript will specify the order in the communication.

Given a solution protocol A for $TPC^1(U)$ (alt. $TPC^2(U)$), let $A(x,b)$ denote the time required by A to communicate $x \in U$ transmitting b bits. Algorithm A is *worst-case optimal* for U if $\forall$ solution protocol B and $\forall$ $b \geq 2$ $\text{Max}\{A(x,b): x \in U\} \leq \text{Max}\{B(x,b): x \in U\}$. That is, a (worst-case) optimal algorithm is optimal (in the worst-case) *at any point* of the time-bit trade-off. Similarly, we can define average-case optimality.

A much stronger notion of optimality is expressed by the following definition. Algorithm A is *globally optimal* if $\forall$ solution protocol B , $\forall$ $b \geq 2$, $\forall$ U there exists a permutation π of of the elements of U such that $\forall$ x in U $A(x,b) \leq B(\pi(x),b)$. Therefore, a globally optimal algorithm is one which, within a bijection, requires no more time to communicate any element of any universe than any other algorithm using the same number of bit transmissions.

In the rest of this section, we will briefly discuss some solution techniques for the two-party communication problem.

2.2 TPC(U,b) FOR b=2, b=3

Consider first the case b=2. There exists a well known solution protocol for $TPC^1(U,2)$: to communicate any $x \in U$, the sender waits x time units between the transmission of the first bit b_0 and second bit b_1; the receiver decodes the quantum of silence as x. The time complexity is exactly x. If the bits can be used to carry information (problem $TPC^2(U,2)$) the quantum of silence can be reduced by a factor of four: the first bit, b_0, is used to indicate whether $w=\lfloor x/2 \rfloor$ is odd; the second bit, b_1, is used to indicate whether $z=\lfloor w/2 \rfloor$ is odd; the quantum waited is z. To obtain x the receiver simply computes $4z + 2b_1+b_0$ (here both bits are treated as integer values). It should be pointed out that this protocol has been successfully employed to obtain a bit-optimal election algorithm for ring networks of unknown size [Bodlaender, Tel-89].

For the case b=3, consider the following protocol for $TPC^1(U,3)$: to communicate $x \in U$, the sender transmits b_0, waits $q_1 = y = \lceil \sqrt{x} \rceil$, transmits b_1, waits $q_2 = z = y^2 - x$, transmits b_2, where the value of the b_i's is arbitrary. To obtain x the receiver simply computes $q_1^2 - q_2$. This protocol has time complexity at most $2\lceil \sqrt{x} \rceil+3$. In the case of $TPC^2(U,3)$, the bits can be used to indicate whether y and z are odd, reducing the time complexity to $\lceil \sqrt{x} \rceil+3$. For more details, see [Santoro-90].

2.3 GENERAL CASE

For the general case of b=k+1 bits (k quanta), a solution protocol for $TPC^2(U,b)$ has been recently proposed by Schmeltz, and it requires time $k\ x^{1/(k-2)}$+l.o.t. in the worst case [Schmeltz-90].

Consider the case when the value of the bits can be used to convey information, a solution protocol for $TPC^2(U,b)$ can be obtained by extending the protocol for $TPC^2(U,3)$ as follows. For simplicity, let k be a power of two. Given $x \in U$, the encoding E(x) of x is defined recursively as follows:

$$E(x) = \quad b_0 * E(X_1) * b_k$$

$$E(X_i) = \begin{array}{|ll} E(X_{2i})* b_i * E(X_{2i+1}) & \text{if } 1<i<k \\ & \\ \text{quantum of length } X_i & \text{if } k\leq i\leq 2k-1 \end{array}$$

where $X_{2i}=\lceil \sqrt{X_i} \rceil$, $X_{2i+1}=\lfloor (X_{2i}^2-X_i)/2 \rfloor$, $X_1=\lfloor \lfloor x/2 \rfloor /2 \rfloor$, $b_i=X_{2i+1} \bmod 2$, $b_k = \lfloor x/2 \rfloor \bmod 2$, and * denotes concatenation (see Figure 1 for an example). The encoded information to be communicated is then the sequence of bits and quanta defined by the recurrence relation defined above (see Figure 1). For the example of x=927 and k=4 the encoding is <*1*,4,*0*,0,*0*,4,*0*,2,*1*>. To obtain x, the receiver will recursively compute $X_i= X_{2i}^2 - (2X_{2i+1} + b_i)$ until X_1 is determined; then, $x=4X_1+2b_k+b_1$. The protocol is provably correct. Exactly k quanta will be used, and b=k+1 bits will be transmitted. It can be shown that $X_{2i+1}\leq\lceil\sqrt{X_i}\rceil$; since $X_{2i}=\lceil\sqrt{X_i}\rceil$ by definition, it follows that each quantum is at most $(x/4)^{1/k}$. Hence, the time complexity is at most $k\ (x/4)^{1/k}+b$.

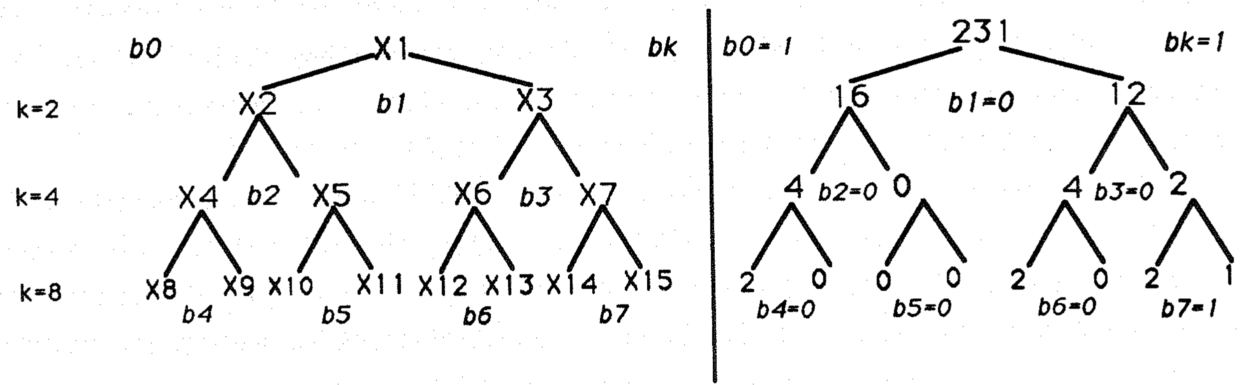

Figure 1. A Solution Protocol for TPC^2(U,b) and an example of E(x= 927)

3. LOWER BOUNDS

In this section we establish lower-bounds on the time-bit trade-off for the two-party communication problem, TPC(U), when the universe U is finite; as will be shown in section 4, these bounds are tight even when U is countable but infinite. We consider both versions of the problem, TPC^1 and TPC^2; that is, when bit transmissions are subject to corruption and when transmission are fault-free, respectively. The bounds apply to any solution protocol, regardless of the encoding/transmitting/decoding scheme employed.

3.1 TPC^1(U,b)

In TPC^1, the value of the transmitted bits cannot be relied upon; hence, the only meaningful events in the system are transmission and silence; furthermore, the time before the first transmission and after the last transmission cannot obviously be used to convey information.

Consider TPC^1(U,b); i.e., the two-party communication problem for U using only b bit transmissions. Observe that b time units will be required by any solution algorithm for

$TPC^1(U,b)$ to transmit the b bits; hence, the concern is on the amount of *additional* time required by the protocol. In the following, unless otherwise specified, "time" refers to "additional" time.

Given a finite universe U, let $T^1(U,b)$ denote the number of time units needed in the worst case to solve $TPC^1(U,b)$. To derive a bound on $T^1(U,b)$, we will consider the dual problem of determining the size $D^1(t,b)$ of the largest set U' for which $T^1(U',b) \leq t$; that is, U' is the largest set for which the two-party communication problem can always be solved using b transmissions and at most t additional time units. Notice that, with b bit transmissions, it is only possible to distinguish k=b-1 quanta; hence, the dual problem can be rephrased as follows:

> Determine the largest positive integer $n=D^1(t,b)$ such that every $x \in Z_n = \{0,1,\dots,n\}$ can be communicated using k=b-1 distinguished quanta whose total sum is at most t.

This problem has an exact solution which will enable us to establish the desired bounds.

Theorem 3.1: $D^1(t,b)=$ bin (t+k, k)

Proof: (Sketch) Let $n=D^1(t,b)$; by definition, it must be possible to communicate any element in $Z_n=\{0,1,\dots,n\}$ using k=b-1 distinguished quanta requiring at most time t. In other words, D(t,k+1) is equal to the number of distinct k-tuples $< t_1, t_2,\dots, t_k>$ of positive integers such that $\Sigma_i\, t_i \leq t$.

Given a positive integer x, let $T_k[x]$ denote the number of ordered k-decompositions of x whose sum is exactly x; i.e., $T_k[x] = |\{<x_1,x_2,\dots,x_k> : \Sigma x_j = x\, ,\, x_j \in Z^+\}|$. $T_k[x]$ is exactly the number of ways of throwing x undistinguished balls into k distinguished buckets; thus, $T_k[x]=$bin (x+k-1, k-1).

Therefore, $D^1(t,k+1) = \Sigma_i\, T_k[i] = \Sigma_i$ bin (i+k-1, k-1) = bin(t+k,k), which proves the theorem.

[]

Given two positive integers x and k, let h(x,k) be the smallest integer t such that $x \leq D^1(t,k+1)$.

Corollary 3.1 (Worst Case): Any protocol for $TPC^1(U,k+1)$ requires h(|U|,k) time units in the worst case; that is, $T^1(U,b)=h(|U|,k)$.

Proof: from Theorem 3.1.

Theorem 3.2: Let $|U| = D^1(t,k+1)$. For any solution protocol P for $TPC^1(U,k+1)$, there exists a partition of U into t+1 disjoint subsets $U_0, U_1, \ldots, U_t$ such that

1. $|U_i| = bin(i+k-1,k-1)$, and
2. the time $P(x)$ required by P to communicate $x \in U_i$ is $P(x) \geq i$.

Proof: Since $|U| = D^1(t,k+1)$, by Theorem 3.1, U is the largest set for which the two-party communication problem can always be solved using b=k+1 transmissions and at most t additional time units. Given a protocol P for $TPC^1(U,k+1)$, order the elements $x \in U$ according to the time $P(x)$ required by P to communicate them; let Ü be the corresponding ordered set. Define $\ddot{U}_i$ to be the subset composed of the elements of Ü whose ranking, with respect to the ordering defined above, is in the range $(\Sigma_{0 \leq j < i}\, bin(j+k-1,k-1),\ \Sigma_{0 \leq j \leq i}\, bin(j+k-1,k-1))$; hence, $|\ddot{U}_i| = bin(i+k-1,k-1)$ which proves part 1 of the Theorem.

We will now show that, for every $x \in \ddot{U}_i$, $P(x) \geq i$. By contradiction, let this not be the case. Let $j \leq t$ be the smallest index for which there exists an $x \in \ddot{U}_i$ such that $P(x) < i$. This implies that there exists a $j' < j$ such that $|\{x \in U: P(x) = j'\}| > bin(j'+k-1,k-1)$. In other words, in protocol P, the number of elements which are uniquely identified using k quanta for a total of j' time is greater than the number $T_k[j'] = bin(j'+k-1,k-1)$ of ordered k-decompositions of j' whose sum is exactly j'; a clear contradiction. Hence, for every $x \in \ddot{U}_i$, $P(x) \geq i$, proving part 2 of the theorem. []

Corollary 3.2 (Average Case): Any protocol for $TPC^1(U,k+1)$ requires on the average $(\Sigma_i\ i\ bin(i+k-1,k-1))\ /\ bin(t+k,k)$ time units.

Proof: from Theorem 3.2.

3.2 $TPC^2(U,b)$

In TPC^2, bit transmissions are error-free; hence, the value of a transmitted bit is meaningful. Consider $TPC^2(U,b)$; i.e., the two-party communication problem for U using only b bit transmissions. As in $TPC^2(U,b)$, the time before the first transmission and after the last transmission cannot be used to convey information, and the concern is on the amount of time in addition to the one needed for bit transmission.

Given a finite universe U, let $T^2(U,b)$ denote the number of time units needed in the worst case to solve $TPC^2(U,b)$. To derive a bound on $T^2(U,b)$, we will consider the dual problem of determining the size $D^2(t,b)$ of the largest set U" for which $T^2(U",b) \leq t$; that is, U" is the largest set for which the two-party communication problem can always be solved transmitting b bits and at most t additional time units. Since with b bit transmissions it is only possible to distinguish k=b-1 quanta, the dual problem can be rephrased as follows:

> Determine the largest positive integer $n=D^2(t,b)$ such that every $x \in Z_n = \{0,1,\dots,n\}$ can be communicated using b bits and k=b-1 distinguished quanta whose total sum is at most t.

This problem has an exact solution which will enable us to establish the desired bounds.

Theorem 3.3: $D^2(t,b) = 2^{k+1}$ bin (t + k, k)

Proof: (sketch). The number of distinct assignment of values to k+1 distinguished bits is 2^{k+1}. The number of distinct k-tuples $< t_1, t_2,\dots, t_k>$ of positive integers such that $\Sigma_j t_j \leq t$ is $D^1(t,b)$ (from Theorem 3.1).Therefore $D^2(t,b)= 2^{k+1} D^1(t,b)= 2^{k+1}$ bin(t+k,k). []

Given two positive integers x and k, let g(x,k) denote the smallest t such that $x \leq D^2(t,k+1)$.

Corollary 3.3 (Worst Case): Any protocol for $TPC^2(U,k+1)$ requires g(|U|,k) time units in the worst case; that is, $T^2(U,b)$=g(|U|,k).

Proof: By Theorem 3.3.

Theorem 3.4: Let $|U|= D^2(t,k+1)$. There exists a partition of U into t+1 disjoint subsets $U_0,U_1,\dots,U_t$, where $|U_i| = 2^{k+1}$ bin(i+k-1,k-1), such that any protocol for $TPC^2(U,k+1)$ requires at least time i to communicate an element $x \in U_i$.

Proof: Similar to Theorem 3.2

Corollary 3.4 (Average Case): Transmission of elements in Z_X where X = T[t,k] using k quanta requires (Σ_i bin(i+k-1,k-1)) / 2^{k+1} bin(t+k,k) time units on average.

Proof: By Theorem 3.4.

4. UPPER BOUNDS

In this section we present solution algorithms for the two versions of the two-party communication problem, TPC^1 and TPC^2. These algorithms solve the problem even when the the universe U of information is infinite. As will be shown later, their complexity matches the global lower-bounds for the problems.

4.1 *P1*: A GLOBALLY OPTIMAL SOLUTION FOR TPC[1]

For a given k, let V_t be an ordered set of size bin(t+k,k) where each element $V_t[i]$ consists of a distinct k-tuple $q= <q_1, q_2, \dots , q_k>$ where $q_i \in Z^+$ and $\Sigma_i \, q_i \le t$. The set is lexicographically ordered; that is, $V_t[i]<V_t[i+1]$ where $<q_1,q_2,\dots ,q_k> < <q'_1,q'_2,\dots,q'_k>$ if $q_j = q'_j$ for $1\le j \le l$, and $q_{l+1}< q'_{l+1}$. A solution algorithm, *P1*, is described below; it comprises of an encoding scheme, a decoding scheme, and a communication protocol.

Encoding Scheme: Given X and k,

1) Let t be the smallest integer such that $X \le$ bin(t+k,k); i.e., t =h(X,k).
2) Find $V_t[X] = <q_1, q_2, \dots , q_k>$
3) Set *encoding*(X) $=<p_0, p_1, \dots , p_{2k}>$, where $p_{2i}= b$ and $p_{2i+1}= q_i$ $(0\le i<k)$.

Decoding Scheme: Given $Z = <p_0, p_1, \dots , p_{2k}>$ and k,

1) Let $Y=<q_1, q_2, \dots , q_k>$ where $q_i=p_{2i+1}$ $(0\le i<k)$; let $t = \Sigma_i \, q_i$.
2) Find X such that $V_t[X] = Y$.
3) Set *decoding*(Z) = X.

Communication Protocol

```
SEND(X):
    Compute encoding(X) = $<p_0, p_1, \dots , p_{2k}>$
    for $0 \le i \le 2k$
        if even(i) then
            transmit $p_i$
        else
            wait $p_i$ time units
        endif
    endfor

RECEIVE(Z)
    i := 0
    receive(b)
    $p_0$= b
    Repeat
        wait q until receive(b)
        $p_{2i+1}$= q
        i := i + 1
        $p_{2i}$= b
    Until i = k
    Z = $<p_0, p_1, \dots , p_{2k}>$
    Compute decoding(Z)
```

For a fixed k, let *P1*(X) denote the amount of time required by algorithm *P1* to communicate integer X using k bit transmissions. Recall (from Section 3) that h(X,k) is the smallest integer t such that $x \leq D^1(t,k+1)$.

Lemma 4.1: For a fixed k, *P1*(X) = h(X,k) for every integer X.
Proof: By construction.

Corollary 4.1: For a fixed k, *P1* is worst-case optimal for every $Z_n=\{0,1,\dots,n\}$.
Proof: By Lemma 4.1 and Corollary 3.1.

Theorem 4.1: For a fixed k, *P1* is globally optimal for every $Z_n = \{0,1,\dots,n\}$.

Proof: Given Z_n, let t=h(n,k) be the smallest integer such that $n \leq D^1(t,k+1)$. Assume for simplicity that n=bin(t+k,k). Let $S_i=\{x \in Z_n : P1(x)=i\}$. By Lemma 1, for every $x \in Z_n$, $P1(x)=h(x,k) \leq t$; hence, $|S_i|=\text{bin}(i+k-1,k-1)$, $0 \leq i \leq t$. Recall that, by Theorem 3.2, for any solution algorithm *A*, there exists a partition of Z_n into t+1 disjoint subsets $A_0,A_1,\dots,A_t$ such that $|A_i|=\text{bin}(i+k-1,k-1)$ and $A(x) \geq i$ for every $x \in A_i$. Therefore, there exists a permutation π of Z_n such that $P1(x) \leq A(\pi(x))$ for all $x \in Z_n$, proving the theorem. []

4.2 *P2*: A GLOBALLY OPTIMAL SOLUTION FOR TPC^2

Given k, let W_t be an ordered set of size $2^{k+1}\text{bin}(t+k,k)$ where each element $W_t[i]$ is a distinct (2k+1)-tuple $p=\langle p_0,p_1, \dots , p_{2k}\rangle$ where $p_{2i} \in \{0,1\}$, $p_{2i+1} \in Z^+$, and $\sum_i p_{2i+1} \leq t$. The set is lexicographically ordered; that is, $W_t[i]<W_t[i+1]$ where $\langle p_0,p_1, \dots , p_{2k}\rangle < \langle p'_0,p'_1, \dots , p'_{2k}\rangle$ if $p_j = p'_j$ for $1 \leq j \leq l$ and $p_{l+1} < p'_{l+1}$.

A solution algorithm, *P2*, is described below; it comprises of an encoding scheme, a decoding scheme, and uses the same communication protocol as *P1*.

Encoding Scheme: Given X and k,

1) Let t be the smallest integer such that $X \leq 2^{k+1}\text{bin}(t+k,k)$; i.e., t =g(X,k).
2) Find $W_t[X] = \langle p_0, p_1, \dots , p_{2k}\rangle$
3) Set *encoding*(X) = $W_t[X]$.

Decoding Scheme: Given $Z = \langle p_0, p_1, \ldots, p_{2k} \rangle$ and k,

1) Let $Y=\langle q_1, q_2, \ldots, q_k \rangle$ where $q_i=p_{2i+1}$ $(0 \leq i < k)$; let $t = \Sigma_i q_i$.
2) Find X such that $W_t[X] = Y$.
3) Set *decoding*(Z) = X.

Communication Protocol: same as in *P1*

For a fixed k, let *P2*(X) denote the amount of time required by algorithm *P1* to communicate integer X using k bit transmissions. Recall (from Section 3) that g(X,k) is the smallest integer t such that $x \leq D^2(t,k+1)$.

Lemma 4.2: For a fixed k, *P2*(X) = g(X,k) for every integer X.

Corollary 4.2: For a fixed k, *P2* is worst-case optimal for every $Z_n=\{0,1,\ldots,n\}$

Theorem 4.2: For a fixed k, *P1* is globally optimal for every $Z_n = \{0,1,\ldots,n\}$.

The proofs of the above lemma and theorem are analogous to those presented in Section 4.1 and therefore are omitted here for brevity.

5. APPLICATIONS AND CONCLUSIONS

Applications of the results to transforming an asynchronous distributed algorithm into a synchronous one will be discussed in a forthcoming paper.

Acknowledgment

This work has been supported in part by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

[Aho, Ullman, Yannakakis-79] Aho, A.V., Ullman, J.D., Yannakakis, M., Modeling communication protocols by automata, Proceedings of 20th FOCS, 1979.

[Aho et al-82] Aho, A.V., Ullman, J.D., Wyner, A.D., Yannakakis, M., Bounds on the size and transmission rate of communication protocols, Computer and Mathematics with Applications, 8 (3) 1982, 205-214.

[Bodlaender, Tel-89] Bodlaender, H.L., Tel, G., Bit-Optimal Election in Synchronous Rings, Technical Report Utrecht University (RUU-CS-89-2), Jan, 1989.

[Fleischer, Jung, Mehlhorn-90] Fleischer, R., Jung, H., Mehlhorn, K., A Time-Randomness Tradeoff for Communcation Complexity, Proceedings of 4th International Workshop on Distributed Algorithms, 1990.

[Godwasser et al -88] Godwasser, S., Micali, S., Rivest, R., A digital signature scheme secure against adaptive chosen-message attack, SIAM Journal on Computing, 17 (2) 1988, 281-308 .

[Hajnal, Maass, Túran-88] Hajnal, A., Maass, W., Túran-, G., On the communication complexity of graph properties, Proceedings 20th STOC, 1988.

[Lynch, Mansour, Fekete-88] Lynch, N.A., Mansour, Y., Fekete, A., Data link layer: two impossibility results, Proceedings of 7th PODC, 1988.

[Santoro-90] Santoro, N., Computing With Time (Temporal Dimensions in Distributed Computing), Proceedings of 28th Annual Allerton Conference on Communication, Control and Computing, Oct 3-5, 1990.

[Schmeltz-90] Schmeltz, B., Optimal Tradeoffs Between Time and Bit Complexity in Distributed Synchronous Rings, Proceedings of STACS'90.

[Wang, Zuck-89] [Wang, D-W, Zuck, L.D., Tight bounds for the sequence transmission problem, Proceedings of 8th PODC, 1989.

[Yao-79] Yao, A.C., Some complexity questions related to distributive computing, Proceedings 11th STOC, 1979.

School of Computer Science, Carleton University
Recent Technical Reports

SCS-TR-155 **Hot-Spot Contention in Binary Hypercube Networks**
Sivarama P. Dandamudi and Derek L. Eager, April 89

SCS-TR-156 **Some Issues in Hierarchical Interconnection Network Design**
Sivarama P. Dandamudi and Derek L. Eager, April 1989

SCS-TR-157 **Discretized Pursuit Linear Reward-Inaction Automata**
B.J. Oommen and Joseph K. Lanctot, April 1989

SCS-TR-158 (revised) **Parallel Fractional Cascading on a Hypercube Multiprocessor**
Frank Dehne, Afonso Ferreira and Andrew Rau-Chaplin, May 1989 (Revised April 1990)

SCS-TR-159 **Epsilon-Optimal Stubborn Learning Mechanisms**
J.P.R. Christensen and B.J. Oommen, June 1989

SCS-TR-160 **Disassembling Two-Dimensional Composite Parts Via Translations**
Doron Nussbaum and Jörg-R. Sack, June 1989

SCS-TR-161 (revised) **Recognizing Sources of Random Strings**
R.S. Valiveti and B.J. Oommen, January 1990
Revised version of SCS-TR-161 "On the Data Analysis of Random Permutations and its Application to Source Recognition", published June 1989

SCS-TR-162 **An Adaptive Learning Solution to the Keyboard Optimization Problem**
B.J. Oommen, R.S. Valiveti and J. Zgierski, October 1989

SCS-TR-163 **Finding a Central Link Segment of a Simple Polygon in O(N Log N) Time**
L.G. Alexandrov, H.N. Djidjev, J.-R. Sack, October 1989

SCS-TR-164 **A Survey of Algorithms for Handling Permutation Groups**
M.D. Atkinson, January 1990

SCS-TR-165 **Key Exchange Using Chebychev Polynomials**
M.D. Atkinson and Vincenzo Acciaro, January 1990

SCS-TR-166 **Efficient Concurrency Control Protocols for B-tree Indexes**
Ekow J. Otoo, January 1990

SCS-TR-167 **A Hierarchical Stochastic Automaton Solution to the Object Partitioning Problem**
B.J. Oommen, January 1990

SCS-TR-168 **Adaptive List Organizing for Non-stationary Query Distributions. Part I: The Move-to-Front Rule**
R.S. Valiveti and B.J. Oommen, January 1990

SCS-TR-169 **Trade-Offs in Non-Reversing Diameter**
Hans L. Bodlaender, Gerard Tel and Nicola Santoro, February 1990

SCS-TR-170 **A Massively Parallel Knowledge-Base Server using a Hypercube Multiprocessor**
Frank Dehne, Afonso Ferreira and Andrew Rau-Chaplin, April 1990

SCS-TR-171 **Parallel Processing of Quad Trees on the Hypercube (and PRAM)**
Frank Dehne, Afonso Ferreira and Andrew Rau-Chaplin, April 1990

SCS-TR-172 **A Note on the Load Balancing Problem for Coarse Grained Hypercube Dictionary Machines**
Frank Dehne and Michel Gastaldo, May 1990

SCS-TR-173 **Self-Organizing Doubly-Linked Lists**
R.S. Valiveti and B.J. Oommen, May 1990

SCS-TR-174 **A Presortedness Metric for Ensembles of Data Sequences**
R.S. Valiveti and B.J. Oommen, May 1990

SCS-TR-175	**Separation of Graphs of Bounded Genus** Ljudmil G. Aleksandrov and Hristo N. Djidjev, May 1990
SCS-TR-176	**Edge Separators of Planar and Outerplanar Graphs with Applications** Krzystof Diks, Hristo N. Djidjev, Ondrej Sykora and Imrich Vrto, May 1990
SCS-TR-177	**Representing Partial Orders by Polygons and Circles in the Plane** Jeffrey B. Sidney and Stuart J. Sidney, July 1990
SCS-TR-178	**Determining Stochastic Dependence for Normally Distributed Vectors Using the Chi-squared Metric** R.S. Valiveti and B.J. Oommen, July 1990
SCS-TR-179	**Parallel Algorithms for Determining K-width-Connectivity in Binary Images** Frank Dehne and Susanne E. Hambrusch, September 1990
SCS-TR-180	**A Workbench for Computational Geometry (WOCG)** P. Epstein, A. Knight, J. May, T. Nguyen, and J.-R. Sack, September 1990
SCS-TR-181	**Adaptive Linear List Reorganization under a Generalized Query System** R.S. Valiveti, B.J. Oommen and J.R. Zgierski, October 1990
SCS-TR-182	**Breaking Substitution Cyphers using Stochastic Automata** B.J. Oommen and J.R. Zgierski, October 1990
SCS-TR-183	**A New Algorithm for Testing the Regularity of a Permutation Group** V. Acciaro and M.D. Atkinson, November 1990
SCS-TR-184	**Generating Binary Trees at Random** M.D. Atkinson and J.-R. Sack, December 1990
SCS-TR-185	**Uniform Generation of Combinatorial Objects in Parallel** M.D. Atkinson and J.-R. Sack, January 1991
SCS-TR-186	**Reduced Constants for Simple Cycle Graph Separation** Hristo N. Djidjev and Shankar M. Venkatesan, February 1991
SCS-TR-187	**Multisearch Techniques for Implementing Data Structures on a Mesh-Connected Computer** Mikhail J. Atallah, Frank Dehne, Russ Miller, Andrew Rau-Chaplin, and Jyh-Jong Tsay, February 1991
SCS-TR-188 Out of print	**Generating and Sorting Jordan Sequences** Alan Knight and Jörg-Rüdiger Sack, March 1991
SCS-TR-189	**Probabilistic Estimation of Damage from Fire Spread** Charles C. Colbourn, Louis D. Nel, T.B. Boffey and D.F. Yates, April 1991
SCS-TR-190	**Coordinators: A Mechanism for Monitoring and Controlling Interactions Between Groups of Objects** Wilf R. LaLonde, Paul White, and Kevin McGuire, April 1991
SCS-TR-191	**Towards Decomposable, Reusable Smalltalk Windows** Kevin McGuire, Paul White, and Wilf R. LaLonde, April 1991
SCS-TR-192	**PARASOL: A Simulator for Distributed and/or Parallel Systems** John E. Neilson, May 1991
SCS-TR-193	**Realizing a Spatial Topological Data Model in a Relational Database Management System** Ekow J. Otoo and M.M. Allam, August 1991
SCS-TR-194	**String Editing with Substitution, Insertion, Deletion, Squashing and Expansion Operations** B John Oommen, September 1991
SCS-TR-195	**The Expressiveness of Silence** Una-May O'Reily and Nicola Santoro, October 1991

School of Computer Science, Carleton University
Recent Technical Reports

SCS-TR-156	**Some Issues in Hierarchical Interconnection Network Design** Sivarama P. Dandamudi and Derek L. Eager, April 1989
SCS-TR-157	**Discretized Pursuit Linear Reward-Inaction Automata** B.J. Oommen and Joseph K. Lanctot, April 1989
SCS-TR-158 (revised)	**Parallel Fractional Cascading on a Hypercube Multiprocessor** Frank Dehne, Afonso Ferreira and Andrew Rau-Chaplin, May 1989 (Revised April 1990)
SCS-TR-159	**Epsilon-Optimal Stubborn Learning Mechanisms** J.P.R. Christensen and B.J. Oommen, June 1989
SCS-TR-160	**Disassembling Two-Dimensional Composite Parts Via Translations** Doron Nussbaum and Jörg-R. Sack, June 1989
SCS-TR-161 (revised)	**Recognizing Sources of Random Strings** R.S. Valiveti and B.J. Oommen, January 1990 Revised version of SCS-TR-161 "On the Data Analysis of Random Permutations and its Application to Source Recognition", published June 1989
SCS-TR-162	**An Adaptive Learning Solution to the Keyboard Optimization Problem** B.J. Oommen, R.S. Valiveti and J. Zgierski, October 1989
SCS-TR-163	**Finding a Central Link Segment of a Simple Polygon in O(N Log N) Time** L.G. Alexandrov, H.N. Djidjev, J.-R. Sack, October 1989
SCS-TR-164	**A Survey of Algorithms for Handling Permutation Groups** M.D. Atkinson, January 1990
SCS-TR-165	**Key Exchange Using Chebychev Polynomials** M.D. Atkinson and Vincenzo Acciaro, January 1990
SCS-TR-166	**Efficient Concurrency Control Protocols for B-tree Indexes** Ekow J. Otoo, January 1990
SCS-TR-167	**A Hierarchical Stochastic Automaton Solution to the Object Partitioning Problem** B.J. Oommen, January 1990
SCS-TR-168	**Adaptive List Organizing for Non-stationary Query Distributions. Part I: The Move-to-Front Rule** R.S. Valiveti and B.J. Oommen, January 1990
SCS-TR-169	**Trade-Offs in Non-Reversing Diameter** Hans L. Bodlaender, Gerard Tel and Nicola Santoro, February 1990
SCS-TR-170	**A Massively Parallel Knowledge-Base Server using a Hypercube Multiprocessor** Frank Dehne, Afonso Ferreira and Andrew Rau-Chaplin, April 1990
SCS-TR-171	**Parallel Processing of Quad Trees on the Hypercube (and PRAM)** Frank Dehne, Afonso Ferreira and Andrew Rau-Chaplin, April 1990
SCS-TR-172	**A Note on the Load Balancing Problem for Coarse Grained Hypercube Dictionary Machines** Frank Dehne and Michel Gastaldo, May 1990
SCS-TR-173	**Self-Organizing Doubly-Linked Lists** R.S. Valiveti and B.J. Oommen, May 1990
SCS-TR-174	**A Presortedness Metric for Ensembles of Data Sequences** R.S. Valiveti and B.J. Oommen, May 1990
SCS-TR-175	**Separation of Graphs of Bounded Genus** Ljudmil G. Aleksandrov and Hristo N. Djidjev, May 1990

SCS-TR-176 **Edge Separators of Planar and Outerplanar Graphs with Applications**
Krzystof Diks, Hristo N. Djidjev, Ondrej Sykora and Imrich Vrto, May 1990

SCS-TR-177 **Representing Partial Orders by Polygons and Circles in the Plane**
Jeffrey B. Sidney and Stuart J. Sidney, July 1990

SCS-TR-178 **Determining Stochastic Dependence for Normally Distributed Vectors Using the Chi-squared Metric**
R.S. Valiveti and B.J. Oommen, July 1990

SCS-TR-179 **Parallel Algorithms for Determining K-width-Connectivity in Binary Images**
Frank Dehne and Susanne E. Hambrusch, September 1990

SCS-TR-180 **A Workbench for Computational Geometry (WOCG)**
P. Epstein, A. Knight, J. May, T. Nguyen, and J.-R. Sack, September 1990

SCS-TR-181 **Adaptive Linear List Reorganization under a Generalized Query System**
R.S. Valiveti, B.J. Oommen and J.R. Zgierski, October 1990

SCS-TR-182 **Breaking Substitution Cyphers using Stochastic Automata**
B.J. Oommen and J.R. Zgierski, October 1990

SCS-TR-183 **A New Algorithm for Testing the Regularity of a Permutation Group**
V. Acciaro and M.D. Atkinson, November 1990

SCS-TR-184 **Generating Binary Trees at Random**
M.D. Atkinson and J.-R. Sack, December 1990

SCS-TR-185 **Uniform Generation of Combinatorial Objects in Parallel**
M.D. Atkinson and J.-R. Sack, January 1991

SCS-TR-186 **Reduced Constants for Simple Cycle Graph Separation**
Hristo N. Djidjev and Shankar M. Venkatesan, February 1991

SCS-TR-187 **Multisearch Techniques for Implementing Data Structures on a Mesh-Connected Computer**
Mikhail J. Atallah, Frank Dehne, Russ Miller, Andrew Rau-Chaplin, and Jyh-Jong Tsay, February 1991

SCS-TR-188 **Generating and Sorting Jordan Sequences**
Out of print
Alan Knight and Jörg-Rüdiger Sack, March 1991

SCS-TR-189 **Probabilistic Estimation of Damage from Fire Spread**
Charles C. Colbourn, Louis D. Nel, T.B. Boffey and D.F. Yates, April 1991

SCS-TR-190 **Coordinators: A Mechanism for Monitoring and Controlling Interactions Between Groups of Objects**
Wilf R. LaLonde, Paul White, and Kevin McGuire, April 1991

SCS-TR-191 **Towards Decomposable, Reusable Smalltalk Windows**
Kevin McGuire, Paul White, and Wilf R. LaLonde, April 1991

SCS-TR-192 **PARASOL: A Simulator for Distributed and/or Parallel Systems**
John E. Neilson, May 1991

SCS-TR-193 **Realizing a Spatial Topological Data Model in a Relational Database Management System**
Ekow J. Otoo and M.M. Allam, August 1991

SCS-TR-194 **String Editing with Substitution, Insertion, Deletion, Squashing and Expansion Operations**
B John Oommen, September 1991

SCS-TR-195 **The Expressiveness of Silence: Optimal Algorithms for Synchronous Communication of Information**
Una-May O'Reilly and Nicola Santoro, October 1991