

# Optimal Sequencing and Arrangement in Distributed Single-Level Tree Networks with Communication Delays

V. Bharadwaj, D. Ghose, and V. Mani

**Abstract**—The problem of obtaining optimal processing time in a distributed computing system consisting of  $(N + 1)$  processors and  $N$  communication links, arranged in a single-level tree architecture, is considered. It is shown that optimality can be achieved through a hierarchy of steps involving optimal load distribution, load sequencing, and processor-link arrangement. Closed-form expressions for optimal processing time is derived for a general case of networks with different processor speeds and different communication link speeds. Using these closed-form expressions, this paper analytically proves a number of significant results that in earlier studies were only conjectured from computational results. In addition, it also extends these results to a more general framework. The above analysis is carried out for the cases in which the root processor may or may not be equipped with a front-end processor. Illustrative examples are given for all cases considered.

**Index Terms**—Communication delays, distributed processing, optimal arrangement, optimal load distribution, optimal load sequencing, optimal processing time, single-level tree networks

## I. INTRODUCTION

THE load allocation problem involving the distribution of processing loads to individual processors to achieve minimum processing time is an important problem. The solution to this problem must take into account the network architecture, speeds of the processors, speeds of the communication links/channels, the number of processors and links, and the origination point of the load. An example of such a situation is the distributed intelligent sensor networks [5], [11], [12]. In such a network, the sensors/processors are geographically distributed, and one of the main problems is to determine the fractions of the total computational load to be distributed to the individual sensors/processors. In general, the computational load can be either indivisible or arbitrarily divisible. In this paper, we consider the latter case, which finds application in the areas of processing of large data files, such as in signal processing, Kalman filtering, and experimental data processing. In [6], [7], communicating processors arranged in a linear and tree network are considered. In [1], [2], the problem of intelligent sensor network of communicating processors, connected through a common bus, is considered. In [1], [2], [6], [7], recursive equations for optimal load distribution are

developed and solved computationally. A closed-form solution for minimum finish time for a bus architecture is also derived in [1], [2]. In [10], the problem of a linear network of communicating processors is considered, and closed-form solutions and an easily implementable computational technique for determining optimal distribution of processing loads among individual processors is presented. In [8], the asymptotic performance analysis of linear and tree networks is presented.

In this paper, we give closed-form solutions for processing time in a single-level tree network, wherein the root processor may or may not be equipped with a front-end processor. The single-level tree network can also be viewed as a star structure, with a processor designated as a central processor and all other processors connected to it through communication links. However, throughout this paper, we refer to this architecture as a single-level tree network. We show that optimal processing time not only can be obtained through an optimal distribution of load but also can be improved further through a combination of optimal load distribution, optimal sequence of distribution, and optimal arrangement of processors and communication links (when such a rearrangement is possible). This combination constitutes an hierarchy of steps leading to the achievement of optimal processing time.

The organization of the paper is as follows. Section II formulates the problem, gives some necessary definitions, derives closed-form expressions, and presents some important previous results. Section III proves the main results of achieving optimal processing time. Section IV concludes the paper with some relevant discussions.

## II. DEFINITIONS AND PROBLEM FORMULATION

### A. Definitions and Previous Results

A single-level tree architecture with  $(N + 1)$  processors and  $N$  links is shown in Fig. 1(a). All the processors are connected to the root processor  $p_0$  via communication links. This tree configuration can be represented as an ordered set as follows:

$$T(p_0) = \{(l_1, p_1), \dots, (l_k, p_k), \dots, (l_N, p_N)\}, \quad (1)$$

where  $(l_k, p_k)$  represents the  $k$ th processor ( $p_k$ ) connected to the root ( $p_0$ ) via a link  $(l_k)$ . This ordered set  $T(p_0)$  gives the arrangement of  $(N + 1)$  processors and  $N$  links. The order represents the sequence in which the root processor distributes

Manuscript received May 18, 1992; revised April 10, 1993.

The authors are with the Department of Aerospace Engineering, Indian Institute of Science, Bangalore 560 012 India; e-mail: vbhwaj@aero.iisc.ernet.in, dghose@aero.iisc.ernet.in, mani@aero.iisc.ernet.in.

IEEE Log Number 9401221.

loads to other processors (i.e., from processors  $p_1$  to  $p_N$  through links  $l_1$  to  $l_N$ ). Note that this order need not represent any physical order in which the processors are arranged. However, for convenience, and without loss of generality, we also assume that the sequence of load distribution is from left to right in Fig. 1(a). Thus, a change in the sequence of load distribution is equivalent to a corresponding change in the order shown in the ordered set  $T(p_0)$  given in (1) or in Fig. 1(a). Processing load distribution is a set of the fractions of the total processing load that is received and processed by each individual processor. This is denoted by  $\alpha = \{\alpha_0, \alpha_1, \dots, \alpha_N\}$ . It is assumed that a processor starts computing only after it receives the whole of the processing load assigned to it. For a single-level tree network  $T(p_0)$ , we define the following terms.

- 1) *Processing Time*: This is denoted as  $\Gamma(T(p_0))$  and defined as follows:

$$\Gamma(T(p_0)) = \max(T_0, T_1, \dots, T_N), \quad (2)$$

where  $T_k$  is the time difference between the instant at which the  $k$ th processor stops processing and the instant at which the root processor initiates the process.

- 2) *Optimal Load Distribution*: This is defined as the load distribution for a given arrangement and a given sequence such that  $\Gamma(T(p_0))$  is minimum.
- 3) *Optimal Sequence*: This is defined as that sequence of optimal load distribution for a given arrangement such that  $\Gamma(T(p_0))$  is minimum.
- 4) *Optimal Arrangement*: This is defined as the arrangement of links and processors, such that  $\Gamma(T(p_0))$  is minimum, provided that optimal sequence and optimal load distribution is followed.

*Notation:*

$w_i$ : A constant inversely proportional to the speed of the processor  $p_i$ .

$z_i$ : A constant inversely proportional to the speed of the link  $l_i$ .

$T_{cm}$ : Time to communicate the entire processing load through a standard link.

$T_{cp}$ : Time to process the entire processing load by a standard processor.

$\sigma$ : Ratio of communication time to processing time for a given load in a standard link and processor (i.e.,  $T_{cm}/T_{cp}$ ).

For a standard processor and a standard link  $w = 1$  and  $z = 1$ , respectively.

Let us define a class  $\tilde{C}$  of single-level tree networks in which the following condition is satisfied.

$$z_k T_{cm} < z(k+1, \dots, k+r) T_{cm} + w(k+1, \dots, k+r) T_{cp}, \quad (3a)$$

for all  $k \in \{0, 1, \dots, N-1\}$  and  $r \in \{1, \dots, N-k\}$ ,

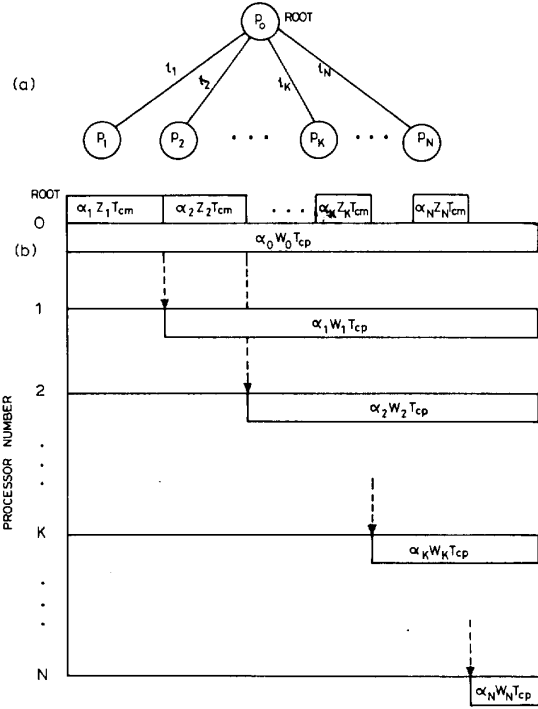


Fig. 1. (a) Single-level tree architecture. (b) Timing diagram for with front-end case.

where

$$z_{eq} = z(k+1, \dots, k+r) = \left[ z_{k+r} + \sum_{i=k+2}^{k+r} \left\{ \prod_{j=i}^{k+r} f_j \right\} z_{i-1} \right] / \left[ 1 + \sum_{i=k+2}^{k+r} \prod_{j=i}^{k+r} f_j \right], \quad (3b)$$

$$w_{eq} = w(k+1, \dots, k+r) = w_{k+r} / \left[ 1 + \sum_{i=k+2}^{k+r} \prod_{j=i}^{k+r} f_j \right], \quad (3c)$$

$$f_j = (w_j + z_j \sigma) / w_{j-1}. \quad (3d)$$

For this, consider (3a) is violated with strict inequality for some  $k$  and  $r = 1$ . Then we have the following equation:

$$z_k T_{cm} > (z_{k+1} T_{cm} + w_{k+1} T_{cp}). \quad (3e)$$

This means that the time taken by the front-end of the processor  $p_0$  to distribute a fraction of the load to the processor  $p_k$  via link  $l_k$  is more than the time taken to distribute the same load fraction through link  $l_{k+1}$  and process it at the processor  $p_{k+1}$ . Hence, it is logical to send the load fraction  $\alpha_k$  to the processor  $p_{k+1}$  rather than to the processor  $p_k$ . The following example clearly illustrate the significance of (3e).

*Example:* Consider a single-level tree network with  $N = 3$  and  $w_0 = 2, w_1 = 3, w_2 = 1, w_3 = 2, z_1 = 2, z_2 = 0.5, z_3 = 5$ , and  $T_{cm} = T_{cp} = 1$ . Assuming all the processors stop computing at the same time instant, the load distribution  $\alpha$  is given as  $\alpha \in \{0.4321, 0.1728, 0.3457, 0.0494\}$  and  $\Gamma(T(p_0)) = 0.8462$ . Now observe that the condition (3e) holds for  $k = 1$  and  $r = 1$ . Hence, omitting  $(l_1, p_1)$  from the network, we obtain the new load distribution,  $\alpha' = \{0.3962, 0, 0.5283, 0.0755\}$  and  $\Gamma(T(p_0))' = 0.7924$ , which is an improvement over the original load distribution.

In a similar way, extending the same logic, if (3a) is violated for some  $r \geq 2$ , then by redistributing the load fraction  $\alpha_k$  among the set of processors  $p_{k+1}, \dots, p_{k+r}$ , the processing time can be decreased. This motivates the condition (3a). Further details are available in [3], [4]. Now we state the optimal load distribution theorem for the single-level tree network of processors and links that belongs to the class  $\tilde{C}$ .

*Theorem 1 (Optimal Distribution):* In a single-level tree  $T(p_0) \in \tilde{C}$ , with a given arrangement and a given sequence, in order to achieve minimum processing time, the optimal load distribution should be such that all the processors must stop computing at the same time (i.e.,  $T_0 = T_1 = \dots = T_N$ ).

A rigorous proof for this theorem is given in [3], [4]. The single-level tree network assumed in the subsequent Sections III-A and III-B belong to the class  $\tilde{C}$ . However, we emphasize that the results proved for this class  $\tilde{C}$  of single-level tree networks can also be extended to a general single-level tree network. This will be shown in Section III-C.

### B. Single-Level Tree Network with Front End

In this architecture, the root processor  $p_0$  is equipped with a front-end processor. The root processor divides the total load into  $(N + 1)$  parts, namely,  $\alpha_0, \alpha_1, \dots, \alpha_N$ . The root processor keeps the fraction  $\alpha_0$  for itself for processing. It transmits the remaining fractions  $\alpha_1, \alpha_2, \dots, \alpha_N$  to the processors  $p_1, \dots, p_N$ , respectively. All the processors at the first level of this architecture perform only computation.

The timing diagram for optimal load distribution [7], using Theorem 1, is shown in Fig. 1(b). The following are the corresponding recursive load distribution equations:

$$\alpha_k w_k T_{cp} = \alpha_{k+1} z_{k+1} T_{cm} + \alpha_{k+1} w_{k+1} T_{cp}, \quad k = 0, 1, \dots, N-1, \quad (4)$$

and the following is the normalizing equation:

$$\sum_{j=0}^N \alpha_j = 1. \quad (5)$$

Rewriting (4) as follows:

$$\alpha_k = \alpha_{k+1} f_{k+1}, \quad k = 0, 1, \dots, N-1, \quad (6)$$

these recursive equations can be solved by expressing all the  $\alpha_k (k = 0, 1, \dots, N-1)$  in terms of  $\alpha_N$  as follows:

$$\alpha_k = \left\{ \prod_{j=k+1}^N f_j \right\} \alpha_N. \quad (7)$$

From (5), the value of  $\alpha_N$  is obtained as follows:

$$\alpha_N = 1 / \left( 1 + \sum_{i=1}^N \prod_{j=1}^N f_j \right). \quad (8)$$

Thus, the fraction of the processing load assigned to the  $k$ th processor is as follows:

$$\alpha_k = \left( \prod_{j=k+1}^N f_j \right) / \left( 1 + \sum_{i=1}^N \prod_{j=1}^N f_j \right), \quad k = 0, 1, \dots, N-1. \quad (9)$$

From Fig. 1(b), it can be seen that the processing time  $\Gamma(T(p_0))$  is the processing time of the root processor given by  $\alpha_0 w_0 T_{cp}$ . Thus,  $\Gamma(T(p_0))$  is as follows:

$$\Gamma(T(p_0)) = \left\{ \left( \prod_{j=1}^N f_j \right) w_0 T_{cp} \right\} / \left( 1 + \sum_{i=1}^N \prod_{j=1}^N f_j \right). \quad (10)$$

The above closed-form solution will be used to prove some of the important results on the minimization of processing time.

### C. Single-Level Tree Network Without Front End

In this architecture, the root processor  $p_0$  is not equipped with front-end. This means that the root processor cannot compute and communicate at the same time instant. Hence, the root processor first distributes the fraction of the processing loads  $\alpha_1, \alpha_2, \dots, \alpha_N$  to the processors  $p_1, p_2, \dots, p_N$ , respectively, and then starts processing its own fraction of the load. In this situation, it can be easily proved that only if the time taken by the root processor to process a given load is more than the time taken to communicate the same load through a communication link can the root processor shares the processing load with another processor through this channel. This leads to the following important criterion for load distribution:

$$w_0 > z_i \sigma, \quad i = 1, 2, \dots, N. \quad (11)$$

Otherwise, if the above criterion is not satisfied for any link ( $z_i$ ), then the root processor does not transmit any load through this link. The timing diagram for optimal load distribution [7], using Theorem 1, is shown in Fig. 2. The following are the corresponding recursive load distribution equations:

$$\begin{aligned} \alpha_0 w_0 T_{cp} &= \alpha_N w_N T_{cp}, \\ \alpha_k w_k T_{cp} &= \alpha_{k+1} z_{k+1} T_{cm} + \alpha_{k+1} z_{k+1} T_{cp}, \end{aligned} \quad k = 1, 2, \dots, N-1. \quad (12)$$

The normalizing equation is as follows:

$$\sum_{j=0}^N \alpha_j = 1. \quad (13)$$

Following the procedure adopted in the case with front-end and using (6), we obtain the fraction of the processing load

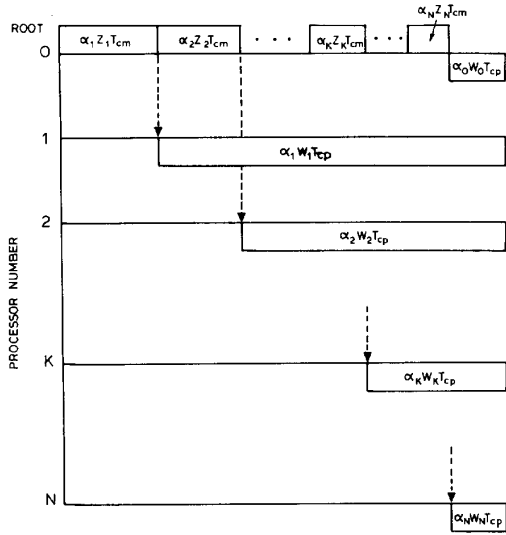


Fig. 2. Timing diagram for without front-end case.

assigned to the  $k$ th processor as follows:

$$\begin{aligned} \alpha_0 &= (w_N/w_0) / \left\{ 1 + (w_N/w_0) + \sum_{i=2}^N \prod_{j=1}^N f_j \right\}, \\ \alpha_k &= \left( \prod_{j=k+1}^N f_j \right) / \left\{ 1 + (w_N/w_0) + \sum_{i=2}^N \prod_{j=1}^N f_j \right\}, \\ &k = 1, 2, \dots, N-1, \\ \alpha_N &= 1 / \left\{ 1 + (w_N/w_0) + \sum_{i=2}^N \prod_{j=1}^N f_j \right\}. \end{aligned} \quad (14) \quad \text{where}$$

Here the processing time  $\Gamma(T(p_0))$  is obtained from  $T_1$  and

is given by the following equation:

$$\Gamma(T(p_0)) = \left\{ \left( \prod_{j=1}^N f_j \right) w_0 T_{cp} \right\} / \left\{ 1 + (w_N/w_0) + \sum_{i=2}^N \prod_{j=1}^N f_j \right\}. \quad (15)$$

In the next section, these closed-form expressions are used to prove some significant results on processing time minimization.

### III. ANALYSIS OF SINGLE-LEVEL TREE ARCHITECTURE

#### A. Minimization of Processing Time: With Front-End Case

In this section, we use the closed-form solutions given in Section II-B to prove the main results. For this, we use the load distribution pattern between two adjacent processors-link pairs ( $k$  and  $k+1$ ) and prove some intermediate results first. Thus, we rewrite the closed-form solutions in such a way that only the terms corresponding to the  $k$ th and  $(k+1)$ th processor and link are present explicitly in the expression for processing time  $\Gamma(T(p_0))$ . The other terms are absorbed in constants defined in (16)–(20) at the bottom of this page. The above expressions and constants are valid for  $k = 1, \dots, N-3$ . These have to be redefined for the right extreme end of the tree as shown below:

$$\begin{aligned} \Gamma(T(p_0)) &= \frac{D(k) f_k f_{k+1} \dots f_N w_0 T_{cp}}{1 + K_4(k) f_k \dots f_N + f_{k+1} \dots f_N + \dots + f_N}, \\ &k = N-2, N-1, \end{aligned} \quad (21)$$

$$D_k = \prod_{j=1}^{k-1} f_j, \quad k = N-2, N-1, \quad (22)$$

$$\begin{aligned} \Gamma(T(p_0)) &= \frac{C(k)(w_{k+2} + z_{k+2}\sigma)(w_{k+1} + z_{k+1}\sigma)(w_k + z_k\sigma)w_0 T_{cp} / (w_{k+1}w_k w_{k-1})}{K_1(k) + K_2(k)(w_{k+2} + z_{k+2}\sigma)\{1 + (w_{k+1} + z_{k+1}\sigma)/w_k\}/w_{k+1} \\ &+ K_3(k)(w_{k+2} + z_{k+2}\sigma)(w_{k+1} + z_{k+1}\sigma)(w_k + z_k\sigma) / (w_{k+1}w_k w_{k-1})} \\ &k = 1, \dots, N-3. \end{aligned} \quad (16)$$

where

$$C(k) = \prod_{\substack{j=1 \\ j \neq k, k+1, k+2}}^N f_j, \quad (17)$$

$$K_1(k) = 1 + \sum_{i=k+3}^N \prod_{j=1}^N f_j, \quad (18)$$

$$K_2(k) = \prod_{j=k+3}^N f_j, \quad (19)$$

$$K_3(k) = \sum_{i=1}^k \prod_{\substack{j=1 \\ j \neq k, k+1, k+2}}^N f_j, \quad (20)$$

$$K_4(k) = \left\{ 1 + \sum_{i=1}^{k-1} \prod_{j=i}^{k-1} f_j \right\}, \quad k = N-2, N-1. \quad (23)$$

**Lemma 1:** In a single-level tree  $T(p_0)$ , if  $z_{k+1} \leq z_k$  for any two adjacent processor-link pairs, then the processing time will decrease or remain the same when  $(l_k, p_k)$  and  $(l_{k+1}, p_{k+1})$  are interchanged.

*Proof:* When the pairs  $(l_k, p_k)$  and  $(l_{k+1}, p_{k+1})$  are interchanged, the resulting arrangement  $T_A(p_0)$  is as follows:

$$T_A(p_0) = \{(l_1, p_1), (l_2, p_2), \dots, (l_{k+1}, p_{k+1}), (l_k, p_k), \dots, (l_N, p_N)\}. \quad (24)$$

We have to prove that  $\Gamma(T_A(p_0)) \leq \Gamma(T(p_0))$  if  $z_{k+1} \leq z_k$ . Using the constants defined earlier, and assuming that  $z_{k+1} = \tau z_k$ ,  $\tau \leq 1$ , the processing times  $\Gamma(T(p_0))$  and  $\Gamma(T_A(p_0))$  are obtained from (16), as shown in (25) and (26) at the bottom of this page. Let us denote the numerators of (25) and (26) as  $N_1$  and  $N_2$ , and the respective denominators as  $D_1$  and  $D_2$ . Since  $N_1 = N_2$ , we calculate the value of  $D_1 - D_2$  as follows:

$$D_1 - D_2 = \{C(k)K_2(k)(w_{k+2} + z_{k+2}\sigma)^2 / (w_{k-1}w_k^2w_{k+1}^2) \times (w_k + z_k\sigma)(w_{k+1} + \tau z_k\sigma)(1 - \tau)z_k\sigma w_0 T_{cp}\}. \quad (27)$$

Thus, in (27), the RHS  $\geq 0$  when  $\tau \leq 1$ , which proves the lemma. Also note that when  $\tau < 1$ , RHS  $> 0$ , which implies a definite decrease in the processing time.

The lemma is proved here for  $k = 1, 2, \dots, N-3$ . For  $k = N-2$  and  $N-1$ , it can be similarly proved using (21)–(23).  $\square$

As mentioned earlier, interchanging adjacent processor-link pairs do not imply a physical rearrangement in the architecture,

but rather a change in the sequence of load distribution by the root processor. An immediate consequence of this result is the following theorem.

**Theorem 2 (Optimal Sequence):** In a single-level tree  $T(p_0)$ , in order to achieve minimum processing time, the sequence of load distribution by the root processor  $p_0$  should follow the order in which the link speeds decrease.

*Proof:* The proof directly follows from Lemma 1.  $\square$

**Lemma 2:** In a single level tree  $T(p_0)$ , the following conditions exist.

- 1) If  $w_{k+1} \leq w_k$  and  $z_{k+1} > z_k$  for any two adjacent processor-link pairs  $(l_k, p_k)$  and  $(l_{k+1}, p_{k+1})$ , then the processing time will decrease or remain the same when only the processors  $p_k$  and  $p_{k+1}$  are interchanged.
- 2) If  $z_{k+1} = z_k$  for any two adjacent links  $l_k$  and  $l_{k+1}$ , then the processing time is independent of the order in which the processors  $p_k$  and  $p_{k+1}$  are arranged.

*Proof:*

- 1) If the processors  $p_k$  and  $p_{k+1}$  are interchanged, then the resulting arrangement  $T_B(p_0)$  is as follows:

$$T_B(p_0) = \{(l_1, p_1), (l_2, p_2), \dots, (l_k, p_{k+1}), (l_{k+1}, p_k), \dots, (l_N, p_N)\}. \quad (28)$$

We have to prove that  $\Gamma(T_B(p_0)) \leq \Gamma(T(p_0))$  if  $w_{k+1} \leq w_k$ , given that  $z_{k+1} \geq z_k$ . Using the constants defined earlier, and assuming that  $w_{k+1} = \beta w_k$ ,  $\beta \leq 1$ , and  $z_{k+1} = z_k$ ,  $\tau > 1$ , the processing times for the above two cases are obtained from (16) as shown in (29) and (30) at the bottom of the page. Following the same procedure as in previous lemma, the value of

$$\Gamma(T(p_0)) = \frac{C(k)(w_{k+2} + z_{k+2}\sigma)(w_{k+1} + \tau z_k\sigma)(w_k + z_k\sigma)w_0 T_{cp} / (w_{k+1}w_kw_{k-1})}{K_1(k) + K_2(k)(w_{k+2} + z_{k+2}\sigma)\{1 + (w_{k+1} + \tau z_k\sigma)/w_k\}/w_{k+1} + K_3(k)(w_{k+2} + z_{k+2}\sigma)(w_{k+1} + \tau z_k\sigma)(w_k + z_k\sigma)/(w_{k+1}w_kw_{k-1})} \quad (25)$$

$k = 1, \dots, N-3.$

$$\Gamma(T_A(p_0)) = \frac{C(k)(w_{k+2} + z_{k+2}\sigma)(w_k + z_k\sigma)(w_{k+1} + \tau z_k\sigma)w_0 T_{cp} / (w_kw_{k+1}w_{k-1})}{K_1(k) + K_2(k)(w_{k+2} + z_{k+2}\sigma)\{1 + (w_k + z_k\sigma)/w_{k+1}\}/w_k + K_3(k)(w_{k+2} + z_{k+2}\sigma)(w_k + z_k\sigma)(w_{k+1} + \tau z_k\sigma)/(w_kw_{k+1}w_{k-1})} \quad (26)$$

$k = 1, \dots, N-3.$

$$\Gamma(T(p_0)) = \frac{C(k)(w_{k+2} + z_{k+2}\sigma)(\beta w_k + z_{k+1}\sigma)(w_k + z_k\sigma)w_0 T_{cp} / (\beta w_k^2 w_{k-1})}{K_1(k) + K_2(k)(w_{k+2} + z_{k+2}\sigma)\{1 + (\beta w_k + z_{k+1}\sigma)/w_k\}/\beta w_k + K_3(k)(w_{k+2} + z_{k+2}\sigma)(\beta w_k + z_{k+1}\sigma)(w_k + z_k\sigma)/(\beta w_k^2 w_{k-1})} \quad (29)$$

$k = 1, \dots, N-3.$

$$\Gamma(T_B(p_0)) = \frac{C(k)(w_{k+2} + z_{k+2}\sigma)(w_k + z_{k+1}\sigma)(\beta w_k + z_k\sigma)w_0 T_{cp} / (\beta w_k^2 w_{k-1})}{K_1(k) + K_2(k)(w_{k+2} + z_{k+2}\sigma)\{1 + (w_k + z_{k+1}\sigma)/\beta w_k\}/w_k + K_3(k)(w_{k+2} + z_{k+2}\sigma)(w_k + z_{k+1}\sigma)(\beta w_k + z_k\sigma)/(\beta w_k^2 w_{k-1})} \quad (30)$$

$k = 1, \dots, N-3.$

$N_1D_2 - N_2D_1$  is obtained as follows:

$$N_1D_2 - N_2D_1 = \left\{ \frac{C(k)K_1(k)(w_{k+2} + z_{k+2}\sigma)}{\beta w_k^2 w_{k-1}} + (\beta w_k + w_k + z_{k+1}\sigma) \right\} \times (\beta - 1)(1 - \tau)w_k z_k \sigma w_0 T_{cp}. \quad (31)$$

The RHS  $\geq 0$  when  $\beta \leq 1$ , given that  $\tau > 1$ , thus proving the first part of this lemma.

- 2) To prove the second part, we use (31). We see that when  $z_{k+1} = z_k$ , i.e.,  $\tau = 1$ , the value of  $(N_1D_2 - N_2D_1)$  reduces to 0, regardless of the value of  $\beta$ , implying that the processing time is independent of the order in which the processors  $p_k$  and  $p_{k+1}$  are arranged.

Here, too, the lemma was proved for  $k = 1, 2, \dots, N - 3$ . It can be similarly proved for  $k = N - 2$  and  $N - 1$  using (21)–(23).  $\square$

Note that interchanging two adjacent processors implies a physical rearrangement in the architecture. An immediate consequence of part 2) of the above lemma is the following theorem.

**Theorem 3:** In a single-level tree  $T(p_0)$ , if all the links have the same speed, i.e.,  $z_i = z$ ,  $i = 1, 2, \dots, N$ , then the processing time is independent of the order in which the processors are arranged.

*Proof:* The proof directly follows from Lemma 2.  $\square$

Note that the above theorem can also be proved using Lemma 1, assuming that  $\tau = 1$ . The result of this theorem was also observed in the numerical example given in [2]. When all the  $z$ 's are equal the network behaves like a bus architecture discussed in [2].

**Theorem 4:** In a single-level tree  $T(p_0)$ , in order to achieve minimum processing time, the processors and links should be arranged in such a way that  $w_{k+1} \geq w_k$  and  $z_{k+1} \geq z_k$ ,  $k = 1, 2, \dots, N - 1$ .

*Proof:* The proof directly follows from Lemma 1 and Lemma 2.  $\square$

This theorem proposes a method by which minimum processing time can be achieved, provided that an architectural rearrangement of links and processors, connected to the root processor, is possible. So far, the root processor was not taken into account during the rearrangement. The following results deal with this aspect of the problem. Here, though exchanging the root with a processor at the first level, the front-end is assumed to remain at the root.

**Lemma 3:** In a single-level tree  $T(p_0)$ , when  $p_0$  is equipped with a front-end, with  $w_{k+1} \geq w_k$  and  $z_{k+1} \geq z_k$ ,  $k = 1, 2, \dots, N - 1$ ; if  $w_1 \leq w_0$ , then the processing time will decrease or remain the same by interchanging the root processor ( $p_0$ ) with the first left-hand-side processor ( $p_1$ ).

*Proof:* Since the processors and links are arranged in such a way that  $w_{k+1} \geq w_k$  and  $z_{k+1} \geq z_k$ , the fastest link-processor pair will be in the first left position, i.e.,  $(l_1, p_1)$ . Suppose we interchange the root processor  $p_0$  with the processor  $p_1$ . The resulting arrangement  $T_C(p_1)$  will be

as follows:

$$T_C(p_1) = \{(l_1, p_0), (l_2, p_2), \dots, (l_N, p_N)\}. \quad (32)$$

We have to prove that  $\Gamma(T_C(p_1)) \leq \Gamma(T(p_0))$  if  $w_1 \leq w_0$ . For this, we use the constants  $K_1(0)$  and  $K_2(0)$ , defined in (18) and (19). Letting  $w_0 = \beta w_1$ ,  $\beta \geq 1$ , the processing times  $\Gamma(T(p_0))$  and  $\Gamma(T_C(p_1))$  are obtained as:

$$\Gamma(T(p_0)) = \frac{K_2(0)(w_2 + z_2\sigma)(w_1 + z_1\sigma)\beta w_1 T_{cp}/(\beta w_1^2)}{K_1(0) + K_2(0)(w_2 + z_2\sigma)\{1 + (w_1 + z_1\sigma)/\beta w_1\}/w_1} \quad (33)$$

$$\Gamma(T_C(p_0)) = \frac{K_2(0)(w_2 + z_2\sigma)(\beta w_1 + z_1\sigma)w_1 T_{cp}/(\beta w_1^2)}{K_1(0) + K_2(0)(w_2 + z_2\sigma)\{1 + (\beta w_1 + z_1\sigma)/w_1\}/\beta w_1}. \quad (34)$$

Denoting the respective numerators as  $N_1$  and  $N_2$ , and denominators as  $D_1$  and  $D_2$ , the value of  $N_1D_2 - N_2D_1$  is obtained as follows:

$$N_1D_2 - N_2D_1 = \{K_2(0)K_1(0)(w_2 + z_2\sigma)/\beta w_2^2\} \times \{z_1\sigma + (w_2 + z_2\sigma)(z_1\sigma + ((w_1 + z_1\sigma)(\beta w_1 + z_1\sigma)/\beta w_1))\} \times (\beta - 1)w_1 T_{cp} \quad (35)$$

In (35), the RHS  $\geq 0$  if  $\beta \geq 1$ , thus proving the lemma. Note that if  $\beta > 1$ , then the RHS  $> 0$ , implying a definite decrease in the processing time.  $\square$

Using all the above results, we state the following theorem.

**Theorem 5 (Optimal Arrangement for with Front-End Case):** Given a set of  $(N + 1)$  processors and  $N$  links to be arranged in a single-level tree architecture, the processing time will be minimum if the processors and links are arranged in such a way that,  $w_0 \leq w_1$ ,  $w_{k+1} \geq w_k$ , and  $z_{k+1} \geq z_k$ ,  $k = 1, 2, \dots, N - 1$ .

*Proof:* The theorem can be easily proved by a contradiction using Theorem 4 and Lemma 3.  $\square$

The above results are useful in improving the performance of a distributed computing network. The optimal distribution theorem (Theorem 1, proved in [4]) provides a basis for obtaining an optimal distribution of processing load to individual processors for a given sequence of distribution in an existing distributed computing network with a single-level tree architecture. The optimal sequence theorem (Theorem 2) prescribes a sequence of optimal load distribution that further enhances the performance of the network. In fact, Theorem 2 also shows that by adopting the sequence of load distribution according to the order relationship between links, an improvement in the processing time is possible, regardless of the speeds of the processors. This, in a way, stresses the ‘‘priority’’ of the links over the processors in minimizing the processing time. Theorem 3 shows that when the links are identical, no improvement can be obtained by changing the sequence of distribution. If it is possible to rearrange the processors and links in the first level, then Theorem 4 prescribes a simple way to improve the performance further.

In addition, if it is possible to rearrange the root also, then Theorem 5 provides a scheme by which best performance can be achieved. Theorem 5 also provides guidelines to design new distributed computing systems with communication delays. In the next section, we show that the results in Lemmas 1 and 2 and Theorems 2–4 are also valid when the root is not equipped with a front-end, though Lemma 3 and Theorem 5 need not hold true in a general sense.

### B. Minimization of Processing Time: Without Front-End Case

In this section, we use the closed-form solution given in Section II-C to prove the main results. Following the procedures adopted in the previous section, closed-form solution for  $\Gamma(T(p_0))$  is rewritten as shown in (36), (37), and (38) at the bottom of this page. These constants are all valid for  $k = 2, 3, \dots, N - 2$ . As in the with front-end case, the processing time for the left and right extremes of the tree are as follows. For  $k = 1$ , we get the following equation:

$$\Gamma(T(p_0)) = \frac{K_2(1)f_3f_2f_1w_0T_{cp}}{M_1(1) + K_2(1)f_2f_3 + K_2(1)f_3}, \quad (39)$$

and for  $k = N - 2, N - 1$ , we get the following:

$$\Gamma(T(p_0)) = \frac{D(k)f_k f_{k+1} \cdots f_N w_0 T_{cp}}{1 + (w_N/w_0) + K_5(k)f_k \cdots f_N + f_{k+1} \cdots f_N + \cdots + f_N}, \quad (40)$$

where

$$K_5(k) = 1 + \sum_{i=2}^{k-1} \prod_{j=1}^{k-1} f_j. \quad (41)$$

It may be noted that (16) and (36) have the same structure, except for some of the constants. Hence, in the case of without front-end, Lemmas 1 and 2 and Theorems 2, 3, and 4 are still valid.

These results imply that even when the root processor is not equipped with a front-end, the optimal sequence of load distribution and the optimal arrangement of links and processors at the first level remain the same as the case with front-end. However, when the root processor is also considered during rearrangement, the results are somewhat different.

When all the link speeds are equal, this architecture behaves like a bus architecture. In [1], it has been conjectured that in such architectures, the fastest processor should be at the root to achieve minimum processing time. In the following analysis, we prove the above conjecture.

*Lemma 4:* In a single-level tree  $T(p_0)$ , with  $w_{k+1} \geq w_k, k = 1, 2, \dots, N - 1$ , and  $z_k = z, k = 1, 2, \dots, N$ , if  $w_1 \leq w_0$ , then the minimum processing time will decrease or remain the same by interchanging the root processor ( $p_0$ ) with the first left-hand-side processor ( $p_1$ ).

*Proof:* Interchanging the processors  $p_1$  and  $p_0$ , we get the following configuration:

$$T_D(p_1) = \{(l_1, p_0), (l_2, p_2), \dots, (l_N, p_N)\}. \quad (42)$$

We have to prove that  $\Gamma(T_D(p_1)) \leq \Gamma(T(p_0))$  if  $w_1 \leq w_0$ . We define the following constants:

$$N = (w_2 + z_2\sigma)K_2(0) \quad (43)$$

$$E = 1 + \sum_{i=2}^{N-1} \prod_{j=i+1}^N f_j, \quad (44)$$

using which, the closed-form expression for  $\Gamma(T(p_0))$  can now be written as follows:

$$\Gamma(T(p_0)) = \{(N/w_1)((w_1 + z\sigma)/w_0)w_0T_{cp}\} / \{E + (w_N/w_0) + (N/w_1)\}. \quad (45)$$

As mentioned earlier, the root processor distributes the load only when (11) is satisfied. So far, we have assumed that (11) is satisfied for the initial arrangement  $T(p_0)$ . However, when  $p_1$  and  $p_0$  are interchanged, this may no longer be valid. Thus, we have two cases.

*Case 1*  $w_1 > z\sigma$ : Then the expression for  $\Gamma(T_D(p_1))$  is as follows:

$$\Gamma(T_D(p_1)) = \{(N/w_0)((w_0 + z\sigma)/w_1)w_1T_{cp}\} / \{E + (w_N/w_1) + (N/w_0)\}. \quad (46)$$

Let us denote the numerators of the above expressions as  $N_1$  and  $N_2$ , and the respective denominators as  $D_1$  and  $D_2$ . Then the value of  $(N_1D_2 - N_2D_1)$  is obtained as follows:

$$\begin{aligned} N_1D_2 - N_2D_1 &= (N/w_1w_0)\{z\sigma E + w_N(1 + z\sigma(w_0 + w_1)/w_1w_0) - N\} \\ &\quad \times (w_0 - w_1), \end{aligned} \quad (47)$$

$$\Gamma(T(p_0)) = \frac{C(k)(w_{k+2} + z_{k+2}\sigma)(w_{k+1} + z_{k+1}\sigma)(w_k + z_k\sigma)w_0T_{cp}/(w_{k+1}w_kw_{k-1})}{M_1(k) + K_2(k)(w_{k+2} + z_{k+2}\sigma)\{1 + (w_{k+1} + z_{k+1}\sigma)/w_k\}/w_{k+1} + M_2(k)(w_{k+2} + z_{k+2}\sigma)(w_{k+1} + z_{k+1}\sigma)(w_k + z_k\sigma)/(w_{k+1}w_kw_{k-1})} \quad (36)$$

$k = 2, \dots, N - 3.$

where

$$M_1(k) = K_1(k) + (w_N/w_0) \quad (37)$$

$$M_2(k) = \sum_{i=1}^{k-1} \prod_{\substack{j=i+1 \\ j \neq k, k+1, k+2}}^N f_j \quad (38)$$

which can be further reduced to the following expression:

$$\begin{aligned} N_1 D_2 - N_2 D_1 \\ = (N/w_0 w_1) \{z\sigma + (w_N z\sigma(w_0 + w_1)/w_0 w_1)\} (w_0 - w_1). \end{aligned} \quad (48)$$

In (48), the RHS  $\geq 0$ , if  $w_0 \geq w_1$ , which proves the lemma.

*Case 2*  $w_1 < z\sigma$ : The processing time is as follows:

$$\Gamma(T_D(p_1)) = w_1 T_{cp}. \quad (49)$$

This is because of the condition  $w_1 < z\sigma$  for which the root processor does not distribute any load to any other processors. We have to prove that  $\Gamma(T_D(p_1)) \leq \Gamma(T(p_0))$ , if  $w_1 \leq w_0$ . From (45) and (49), we get the following:

$$\begin{aligned} \Gamma(T(p_0)) - \Gamma(T_D(p_1)) \\ = (N/w_1)(1 - w_1/z\sigma)(w_1 + z\sigma) - w_1 w_N(1/w_0 - 1/z\sigma). \end{aligned} \quad (50)$$

In (50), the RHS  $\geq 0$  if  $w_1 \leq z\sigma$  and  $w_0 \geq z\sigma$ , thus proving the lemma.  $\square$

**Theorem 6 (Optimal Arrangement for Equal Link Speeds):** Given a set of  $(N+1)$  processors with arbitrary speeds and  $N$  links with equal speeds, to be arranged in a single-level tree architecture, the processing time will be the minimum if the fastest processor is at the root.

*Proof:* This can be proved by contradiction using Lemma 4.  $\square$

Note that Theorem 6 is valid even for the case with front-end. In fact, Theorem 6 is a special case of Theorem 5. Also, when the conditions of Theorem 6 are satisfied, the best performance is achieved, regardless of the arrangement of processors at the first level, as long as the root processor is the fastest.

Unlike the with front-end case, when the link speeds are different, the fastest processor need not be the root processor in order to achieve minimum processing time. This is shown with an example in Section IV-B. It is not possible to arrive at a simple condition (as in Theorem 5) to determine the root processor in the without front-end case. Therefore, we propose the following algorithm to achieve an optimal arrangement in this case.

**Algorithm:** Let there be  $(N+1)$  processors with speed parameters  $w_0, w_1, \dots, w_N$  and  $N$  links with speed parameters  $z_1, z_2, \dots, z_N$ , respectively. The algorithm takes these speed parameters as its input. We denote the root processor as  $p_r$  and its corresponding speed parameter as  $w_r$ .

*Step 0:* Arrange the processors and links such that  $w_0 \leq w_1, w_k \leq w_{k+1}, z_k \leq z_{k+1}$ , for  $k = 1, 2, \dots, N-1$ , where  $w_0$  is the speed parameter of the root processor in this initial arrangement. Thus, the processors and links are arranged in decreasing order of speeds.

*Step 1:* Set  $w_r = w_0$ . Delete all the pairs  $(l_i, p_i)$  for which  $w_r \leq z_i\sigma$ ,  $i = 1, 2, \dots, N$ . Compute the processing time  $\Gamma(T(p_0))$ . Restore the deleted pairs. Set  $k = 1$ .

TABLE I  
OPTIMAL ARRANGEMENT FOR WITHOUT FRONT-END CASE  
 $T_{cm} = 0.2, T_{cp} = 2.0, \sigma = 0.1$

$k$	$w_0$	$w_1$	$w_2$	$z_1$	$z_2$	$\Gamma(T(p_k))$
0	1.0	1.5	3.0	1.0	8.0	1.240816
1	1.5	1.0	3.0	1.0	8.0	1.229411
2	3.0	1.0	1.5	1.0	8.0	1.331578

*Step 2:* Interchange  $p_k$  and  $p_r$ . Set  $w_r = w_k$ . Delete all the pairs  $(l_i, p_i)$  for which  $w_r \leq z_i\sigma$ ,  $i = 0, 1, \dots, N$ ,  $i \neq k$ . Compute the processing time  $\Gamma(T(p_k))$ . Restore the deleted pairs.  $k = k + 1$ . If  $k \leq N$ , go to Step 2.

*Step 3:* Find  $j = \arg \min_{0 \leq k \leq N} \Gamma(T(p_k))$ . The configuration  $T(p_j)$  is the optimal arrangement.

Now we present a numerical example to demonstrate the various steps of the proposed algorithm. The results are shown in Table I. Here, initially, the processors and the links have been arranged in decreasing order of speeds, with the root as the fastest processor. For  $k = 1$  and  $k = 2$ , Step 2 of the algorithm is carried out. Finally, in Step 3, we choose the arrangement that gives the minimum processing time (given by  $k = 1$ ). Note that the processing time is minimum for a case in which the root processor is not the fastest.

### C. Extension to General Single-Level Tree Networks

In this section, we show that the results on optimal sequence and optimal arrangement proved earlier will also hold to a general single-level tree network  $T(p_0)$  that may not belong to the class  $\tilde{C}$ . For this, we shall prove the following theorem.

**Theorem 7:** Consider two single-level tree networks  $T(p_0) = \{(l_1, p_1), \dots, (l_N, p_N)\}$  and  $T^*(p_0) = \{(l_1, p_1), \dots, (l_N, p_N), (l_{N+1}, p_{N+1})\}$ . Then  $\Gamma(T^*(p_0)) < \Gamma(T(p_0))$ .

*Proof:* Let us consider a single-level tree network in which the root processor  $p_0$  is equipped with front-end. We denote the numerator and denominator of (10) as  $P$  and  $(1+Q)$ , respectively. Hence, we get the following equation:

$$\Gamma(T(p_0)) = P/(1+Q). \quad (51)$$

Similarly, the expression for  $T^*(p_0)$  can be written as follows:

$$\Gamma(T^*(p_0)) = P f_{N+1}/(1+Q f_{N+1} + f_{N+1}). \quad (52)$$

On comparing (51) and (52), we see that  $\Gamma(T^*(p_0)) < \Gamma(T(p_0))$ . Similar result can be proved for without front-end case.  $\square$

Now we apply the above theorem to any general single-level tree network  $T(p_0)$ . Let us define an ordered set of all processor-link pairs (ordered according to the decreasing link speeds) for which the condition (3a) is violated. We prune off this set of processor-link pairs and append each of these one-by-one at the tail end of the tree, maintaining the order in which the link speeds decrease. At each step of this process, according to Theorem 7, the processing time decreases. The entire process is repeated at each step. Since the number of elements in  $T(p_0)$  is finite, the process terminates resulting



in a network that belongs to class  $\tilde{C}$ . Further, in Lemmas 1–4, interchanging of processors and/or links does not violate condition (3a). Hence, a network in class  $\tilde{C}$  remains in class  $\tilde{C}$ , regardless of such interchanges. Thus, the optimal sequence and optimal arrangement theorems are valid for the general network, too.

#### IV. CONCLUSION

In this paper, we have proved some fundamental results concerning optimal load distribution, and optimal sequencing of the computational load in a single-level tree network consisting of  $(N + 1)$  processors and  $N$  links. We have also proposed a scheme to obtain an optimal arrangement of links and processors in the network, when such a rearrangement is possible. Unlike previous literature [1], [2], [6], [7], where some of these facts were conjectured from computational results, we present closed-form solutions and mathematically rigorous proofs. It should be pointed out that in a recent paper [9], results regarding optimal sequencing are presented for tree networks only when the root processor is equipped with front-end.

There is a definite scope for much further research in this area. For example, the mathematical model adopted in this paper, and in the previous literature [6], [7], assumes that the link speeds are independent of the processor speeds. However, this may not be true in practical situations. Moreover, the communication delay that is assumed to bear a linear relationship with the load may, in reality, have a more complicated relationship. These factors should be taken into account in analyzing a realistic situation. It would also be interesting to obtain a simpler scheme to determine the optimal arrangement for the without front-end case.

The results given in this paper can provide a basis for solving the problems arising as a result of the above-mentioned practical issues. It also seems possible to extend these ideas to other distributed computing architectures to obtain closed-form solutions for optimal load distribution, optimal load sequencing, and optimal arrangement.

#### ACKNOWLEDGMENT

We would like to thank T. G. Robertazzi for his keen interest in this work, and also one of the referees for bringing [9] to our attention.

#### REFERENCES

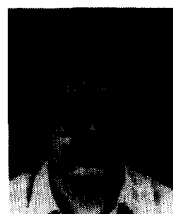
- [1] S. Bataineh and T.G. Robertazzi, "Bus oriented load sharing for a network of intelligent sensors," presented at the *Conf. Inform. Sci. Syst.*, Johns Hopkins Univ., Baltimore, MD, Mar. 1991.
- [2] ———, "Bus oriented load sharing for a network of sensor driven processors," *IEEE Trans. Syst., Man, Cybernetics*, vol. 21, pp. 1202–1205, Sept.–Oct. 1991.
- [3] V. Bharadwaj, D. Ghose, and V. Mani, "A study of optimality conditions for load distribution in tree networks with communication delays," Tech. Rep. 423/GI/02-92, Dept. of Aerospace Eng., Indian Inst. of Sci., Bangalore, India, Dec. 1992.
- [4] ———, "Optimality conditions for load distribution in tree networks with communication delays," private communication.

- [5] Z. Chair and P. K. Varshney, "Optimum data fusion in multiple sensor detection systems," *IEEE Trans. Aerospace Electron. Syst.*, vol. AES-22, no. 1, pp. 98–101, Jan. 1986.
- [6] Y. C. Cheng and T. G. Robertazzi, "Distributed computation with communication delays," *IEEE Trans. Aerospace Electron. Syst.*, vol. 24, pp. 700–712, Nov. 1988.
- [7] ———, "Distributed computation for a tree network with communication delay," *IEEE Trans. Aerospace Electron. Syst.*, vol. 26, pp. 511–516, July 1990.
- [8] D. Ghose and V. Mani, "Distributed computation with communication delays: Asymptotic performance analysis," *J. Parallel Distrib. Computing*, to appear.
- [9] H. J. Kim, G. I. Jee, and J. G. Lee, "Optimal load distribution for tree network processors," private communication.
- [10] V. Mani and D. Ghose, "Distributed computation in a linear network: Closed-form solutions," *IEEE Trans. Aerospace Electron. Syst.*, vol. 30, pp. 471–483, Apr. 1994.
- [11] A. R. Reibman and L. W. Nolte, "Optimal detection and performance of distributed sensor systems," *IEEE Trans. Aerospace Electron. Syst.*, vol. 23, no. 1, pp. 24–30, Jan. 1987.
- [12] R. R. Tenney and N. R. Sandell, "Detection with distributed sensors," *IEEE Trans. Aerospace Electron. Syst.*, vol. AES-17, pp. 501–510, July 1981.



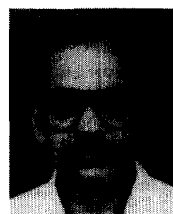
**V. Bharadwaj** received the B.Sc. degree in physics from Madura College, Madurai, India, in 1987, and the M.E. degree in electrical communication engineering from the Indian Institute of Science, Bangalore, India, in 1991.

Currently, he is working towards the Ph.D. degree in the Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India. His research interests include parallel and distributed computing, fiber optic communications, and electromagnetic analysis of integrated optics.



**D. Ghose** received the B.Sc. (Eng.) degree in electrical engineering from the Regional Engineering College, Rourkela, India, in 1982, and the M.E. and Ph.D. degrees in electrical engineering from the Indian Institute of Science, Bangalore, India, in 1984 and 1990, respectively.

From 1984 to 1987, he worked as a Scientific Officer in the Joint Advanced Technology Program at the Indian Institute of Science. Since 1990, he has been a Lecturer in the Department of Aerospace Engineering, Indian Institute of Science. His research interests include guidance and control of aerospace vehicles, game theory, and mathematical economics.



**V. Mani** received the B.E. degree in civil engineering from Madurai University, India, in 1974, the M.Tech. degree in aeronautical engineering from the Indian Institute of Technology, Madras, India, in 1976, and the Ph.D. degree in engineering from the Indian Institute of Science, Bangalore, India, in 1986.

From 1986 to 1988, he was a Research Associate in the School of Computer Science, University of Windsor, ON, Canada, and from 1989 to 1990 in the Department of Aerospace Engineering, Indian Institute of Science. Since 1990, he is an Assistant Professor at the Department of Aerospace Engineering, Indian Institute of Science. His research interests include queuing networks, reliability, neural computing, and mathematical modeling.