Balanced Spanning Trees in Complete and Incomplete Star Graphs

Tzung-Shi Chen, *Member, IEEE Computer Society*, Yu-Chee Tseng, *Member, IEEE Computer Society*, and Jang-Ping Sheu, *Member, IEEE Computer Society*

Abstract—Efficiently solving the personalized broadcast problem in an interconnection network typically relies on finding an appropriate spanning tree in the network. In this paper, we show how to construct in a complete star graph an asymptotically balanced spanning tree, and in an incomplete star graph a near-balanced spanning tree. In both cases, the tree is shown to have the minimum height. In the literature, this problem has only been considered for the complete star graph, and the constructed tree is about 4/3 times taller than the one proposed in this paper.

Index Terms—Balanced spanning tree, interconnection network, parallel architecture, personalized broadcast, star graph.

1 Introduction

The star graph interconnection network was first proposed in [1], and since then it has received increasing attention. The star graph has been considered as an attractive alternative to the widely accepted hypercube as the network architecture for parallel processing. Part of the reason is its lower node degrees as opposed to the hypercubes. Large references can be found in studying the star graph regarding such as its topological properties [3], embedding capability [5], [9], [10], communication capability [2], [7], [8], [11], [12], and even the use of incomplete star graphs [6].

There have been intensive studies on the *collective communication* in a star graph. In an n-star (or S_n), assuming that a node can send and receive at most one message at a time, the algorithms in [2], [7], [12] require $O(n \log n)$ time to perform one-to-all broadcasting. However, if a node can send and receive messages concurrently along all communication ports, the algorithm in [11] needs $2n - 3$ steps to broadcast a message, and needs $O(n + m)$ steps to broadcast a sequence of m messages in a pipelined fashion. A systematic way to find a greedy spanning tree in an n-star was proposed in [3], thus implying the realization of one-to-all broadcast in $\left\lfloor \frac{3(n-1)}{2} \right\rfloor$ steps. An all-to-all broadcasting algorithm was proposed in [7], which has a time complexity of $O(n^2)$. Note that all these results are only regarding to the sending of *nonpersonalized* messages. In the *personalized broadcast* problem, a source node has $N - 1$ *distinct* messages, each to be sent to one of the other nodes in the network, where N is the network size. This problem can be further classified as *one-to-all* and *all-to-all* personalized broadcast. Applications of such broadcast include the fast Fourier transformation (FFT) and matrix algorithms.

In this paper, we consider the problem of constructing a balanced spanning tree in a star graph. A balanced tree is one in which the degree of the root node is maximized and every subtree of the root node has approximately the same number of nodes. One important application of such a tree is to perform one-to-all personalized broadcast in a network. For instance, if a node is to perform a one-to-all personalized broadcast, then it can construct a spanning tree rooted at itself and deliver messages according to the structure of the tree. The communication bottleneck is on the outgoing links (edges) from the source node. If a balanced tree is used, then all outgoing links from the root will transmit about the same number of personalized messages, thereby incurring even load on these links and thus minimizing the transmission time. Constructing a balanced spanning tree has been considered for star graphs in [3] and for hypercubes in [4]. In this paper, we present results for both complete and incomplete star graphs. The incomplete star graph under consideration is the class C^{n-1} (defined later) proposed in [6]. The class C^{n-1} is more scalable in network size than the original class S_n, while at the same time still keeps a diameter equal to that of an S_n.

Given a complete S_n, we show how to construct from any root node a spanning tree that is asymptotically balanced and has a minimum height of $\left\lfloor \frac{3(n-1)}{2} \right\rfloor$. To evaluate how balanced the tree is, we use a measure called of *balance factor*, which is defined to be the ratio of the size of the largest subtree of the root to that of the smallest subtree. This factor indicates how much more transmission time the root node requires to send personalized messages on the link to the largest subtree to that on the link to the smallest subtree. A perfectly balanced tree will have a balance factor equal to, or close to, 1. The balance factor of our tree quickly

• *T.-S. Chen is with the Department of Information Management, Chang Jung University, Tainan, Taiwan, Republic of China. E-mail: chents@mail.cju.edu.tw.*
• *Y.-C. Tseng is with the Department of Computer Science, Chung-Hua Polytechnic Institute, Hsin-Chu, 30067, Taiwan, Republic of China. E-mail: yctseng@chpi.edu.tw.*
• *J.-P. Sheu is with the Department of Computer Science and Information Engineering, National Central University, Chung-Li 32054, Taiwan, Republic of China. E-mail: sheujp@mbox.ee.ncu.edu.tw.*

Manuscript received Oct. 3, 1994; revised Aug. 12, 1995.
For information on obtaining reprints of this article, please send e-mail to: transpds@computer.org, and reference IEEECS Log Number D95201.

1045-9219/96$05.00 ©1996 IEEE

converges to 1 as n increases. For incomplete stars in the class C^{n-1}, our spanning tree is near-balanced and also has a minimum height.

Our result has implied how to perform personalized broadcast efficiently in both S_n and C^{n-1}. In the literature, this problem has only been considered for the complete star graph in [3], where an asymptotically balanced tree is proposed. The height of the tree is $2n - 3$, which is about 4/3 taller than ours. The tree, when used for personalized broadcast, may incur more traffic than ours. Furthermore, the height, say h, of a tree sets a lower bound of $h\tau$ on the time complexity, if one is to use the tree to solve any kind of communication problem, where τ is the start-up time to initiate a message along a communication link. This factor is of particular importance in recent machines because in current technology the start-up overhead is typically higher than the data transmission overhead by an order or more.

The rest of this paper is organized as follows. Preliminaries about the complete and incomplete star graphs are given in Section 2. Our tree construction scheme is presented in Section 3. Conclusions are drawn in Section 4.

2 Preliminaries

An *n-dimensional star graph*, also referred to as *n-star* or S_n, is an undirected graph consisting of $n!$ nodes (or vertices) and $(n - 1)n!/2$ edges. Each node is uniquely assigned a label $x_1x_2 \cdots x_n$ which is the concatenation of any permutation of n distinct symbols $\{x_1, x_2, \ldots, x_n\}$. Two nodes are joined by an edge *along* dimension d iff the label of one node can be obtained from the other by swapping the first symbol and the dth symbol, $2 \le d \le n$. Without loss of generality, let these n symbols be $\{1, 2, \ldots, n\}$. With these edges, we define $n - 1$ functions g_d, $d = 2 \ldots n$, such that if u and v are two nodes joined by an edge along dimension d, then $g_d(u) = v$ and $g_d(v) = u$. For instance, $g_4(12345) = 42315$. The ith symbol of a node x's label may be denoted as $x[i]$. It is known that S_n is node- and edge-symmetric and has a diameter of $\left\lfloor \frac{3(n-1)}{2} \right\rfloor$. More topological properties of the star graph can be found in [1].

Consider an S_n. Let $s_1, s_2, \ldots, s_k$ be k distinct symbols. Let $p_1, p_2, \ldots, p_k$ be k distinct integers each indicating a position in a node label, $p_i \in \{1, 2, \ldots, n\}$, $1 \le i \le k$. We define $S_n^k(s_1^{p_1}, s_2^{p_2}, \ldots, s_k^{p_k})$ to be the subgraph of S_n with vertex set:

$$V\left(S_n^k\left(s_1^{p_1}, s_2^{p_2}, \ldots, s_k^{p_k}\right)\right) = \left\{x_1x_2 \cdots x_n \mid x_{p_i} = s_i, i = 1 \ldots k\right\}.$$

Intuitively, we fix the symbol at position p_i with s_i, $i = 1 \ldots k$, and arbitrarily change the rest symbols. For any two nodes u and v in the vertex set, an edge $\langle u, v\rangle$ is introduced in $S_n^k(s_1^{p_1}, s_2^{p_2}, \ldots, s_k^{p_k})$ iff it is an edge in S_n. For instance, $S_4^2(3^2, 1^4)$ consists of nodes 2341 and 4321 and an edge $\langle 2341, 4321\rangle$. In fact, it is not hard to verify that if $p_i \ge 2$ for all i, then $S_n^k(s_1^{p_1}, s_2^{p_2}, \ldots, s_k^{p_k})$ is an $(n - k)$-star. Such a structure is usually referred to as a *substar* or *(n − k)-substar* of S_n. Occasionally, we may denote $S_n^k(s_1^{p_1}, s_2^{p_2}, \ldots, s_k^{p_k})$ by a sequence $x_1x_2 \ldots x_n$ such that for all $i = 1 \ldots k$ symbol $x_{p_i} = s_i$, and for all $j \notin \{p_1, p_2, \ldots, p_k\}$ symbol $x_j = *$, where a * means a "don't care." For instance, $S_4^2(3^2, 1^4)$ may be written as *3*1.

One of the reasons making the star graph so attractive is its recursive nature. For example, an S_n can be partitioned into $n(n - 1)$-substars, $S_n^1(1^n), S_n^1(2^n), \ldots, S_n^1(n^n)$. Each of these substars can be further partitioned into $n - 1$ $(n - 2)$-substars. This is also the most common and straightforward recursive technique that has been used so far (e.g., [2], [3], [7], [11]). In the following lemma, we describe a very different way to partition an S_n. It turns out that such a partitioning is critical to construct a tree of a minimum height.

Lemma 1. *An S_n can be partitioned into (disjoint) substars A, $B_{i,j}$, $C_{i,j}$, and R as defined below:*

a) $A = S_n^1(1^n)$,

b) $B_{i,j} = S_n^{n-j+1}(i^j, j+1^{j+1}, j+2^{j+2}, \ldots, n^n)$ *for all* $1 \le i < j \le n-1$ *(that is,* $B_{i,j} = * \cdots *i(j+1)(j+2) \cdots n$*)*,

c) $C_{i,j} = S_n^2(i^j, j^n)$ *for all* $1 \le i \le n$ and $2 \le j \le n-1$ *such that* $j \ne i$, and

d) $R = 12 \ldots n$.

Proof. Recall that S_n can be partitioned substars $S_n^1(1^n), S_n^1(2^n), \ldots, S_n^1(n^n)$. A is the first substar. The lemma is obtained by further partitioning the rest substars.

Now we prove that the second, third, …, $(n - 1)$th substars are partitioned into $C_{i,j}$s. Observe that $C_{i,j}$ is the substar with symbol i fixed at position j and symbol j fixed at position n such that $i \ne j$. For a fixed j, the $n - 1$ substars $C_{i,j}$ for all possible values of i together form the substar $S_n^1(j^n)$. This proves our claim.

The last substar $S_n^1(n^n)$ will be partitioned recursively. We first partition it into $n - 1$ substars $S_n^2(1^{n-1}, n^n), S_n^2(2^{n-1}, n^n), \ldots, S_n^2(n-1^{n-1}, n^n)$ (of dimension $n - 2$). Clearly, the first $n - 2$ substars are $B_{i,n-1}$, $i = 1 \ldots n - 2$. The remaining substar is $S_n^2(n-1^{n-1}, n^n)$. We then partitioned it into $n - 2$ substars

$$S_n^3(1^{n-2}, n-1^{n-1}, n^n), S_n^3(2^{n-2}, n-1^{n-1}, n^n), \ldots, S_n^3(n-2^{n-2}, n-1^{n-1}, n^n).$$

Again, the first $n - 3$ substars are $B_{i,n-2}$, $i = 1 \ldots n - 3$. The remaining substar is $S_n^3(n-2^{n-2}, n-1^{n-1}, n^n)$, which can be recursively partitioned as above. Finally, a 1-substar $S_n^{n-1}(2^2, 3^3, \ldots, n^n)$ will be left unpartitioned, which is the substar R. Hence the lemma. □

Example 1. By Lemma 1, an S_6 can be partitioned into the following 32 substars, ordered from larger ones to smaller ones:

$A = *****1$

$C_{1,2} = *1***2$ $C_{3,2} = *3***2$ $C_{4,2} = *4***2$ $C_{5,2} = *5***2$ $C_{6,2} = *6***2$

$C_{1,3} = **1**3$ $C_{2,3} = **2**3$ $C_{4,3} = **4**3$ $C_{5,3} = **5**3$ $C_{6,3} = **6**3$

$C_{1,4} = ***1*4$ $C_{2,4} = ***2*4$ $C_{3,4} = ***3*4$ $C_{5,4} = ***5*4$ $C_{6,4} = ***6*4$

$C_{1,5} = ****15$ $C_{2,5} = ****25$ $C_{3,5} = ****35$ $C_{4,5} = ****45$ $C_{6,5} = ****65$

$B_{1,5} = ****16$ $B_{2,5} = ****26$ $B_{3,5} = ****36$ $B_{4,5} = ****46$

$B_{1,4} = ***156$ $B_{2,4} = ***256$ $B_{3,4} = ***356$

$B_{1,3} = **1456$ $B_{2,3} = **2456$

$B_{1,2} = *13456$

$R = 123456$

Next, we introduce a class of incomplete star graphs proposed in [6] called C^{n-1}. The class C^{n-1} consists of $n-1$ graphs. The kth member, $k = 2 \dots n$, is denoted as $C^{n-1}(k)$ and has the vertex set

$$V\left(C^{n-1}(k)\right) = \bigcup_{i=n-k+1}^{n} V\left(S_n^1\left(i^n\right)\right).$$

For any two nodes u and v in $C^{n-1}(k)$, edge $\langle u, v\rangle$ is introduced in $C^{n-1}(k)$ iff it is an edge in the original S_n. Thus, $C^{n-1}(k)$ is a subgraph of S_n and $C^{n-1}(n) = S_n$. Each $C^{n-1}(k)$ has $(n-1)!$ more nodes than $C^{n-1}(k-1)$. As an S_n has $(n-1)(n-1)!$ more nodes than S_{n-1}, the introduction of these incomplete stars makes the original star graphs incrementally more scalable. Several interesting topological properties of C^{n-1} have been proven in [6], one of which states that the diameter of $C^{n-1}(k)$ is the same as S_n.

3 Constructing Balanced Spanning Trees

3.1 Basic Idea

In the literature, the most common technique to find a spanning tree in a star graph is by recursion. For example, to construct a spanning tree rooted from node r in an S_n, we can

1) partition S_n into substars $S_n^1(1^n), S_n^1(2^n), \dots, S_n^1(n^n)$,
2) construct a path from r to a representative node in each of these substars except the one where r is resident,
3) recursively construct a tree in each of the above substars by regarding the representative node as the root.

Several works (e.g., the broadcast trees [2], [11] and the balanced tree [3]) essentially follow the above recursive rules. Observe that the dimension of the stars is reduced by 1 after each recursive step and the path constructed in step 2) would be as long as 2. Thus, the final tree will have a height $\approx 2n$. Unfortunately, this height is not optimal for the diameter of S_n is $\left\lfloor \frac{3(n-1)}{2} \right\rfloor$.

In our scheme, in each recursion, we will also construct from r a number of paths, each connected to a representative node of some substar. However, the substars are partitioned according to Lemma 1. Most importantly, these paths will have a length ≤ 3. A path of length 1 must connect to a substar of dimension $\le n-1$, and a path of length 2 or 3 must connect to a substar of dimension $\le n-2$. Recursively applying this rule, a tree of an optimal height can be found.

3.2 Complete Stars

Given an S_n and any node r, our purpose in this section is to construct from S_n a balanced spanning tree called $BT(r, S_n)$ rooted from r. As star graphs are node-symmetric, it suffices to show the case of $r = 12 \cdots n$. The balanced spanning tree rooted from any node $r \ne 12 \cdots n$ can be obtained from $BT(12 \cdots n, S_n)$ by substituting symbol $r[i]$ for each occurrence of symbol i in all node labels, where $i = 1 \dots n$.

The construction of $BT(r, S_n)$ is by induction on n. The induction base is $BT(1, S_1) = S_1$ and $BT(12, S_2) = S_2$. For the induction hypothesis, we assume that spanning tree $BT(12 \cdots i, S_i)$ has been constructed for any $i \le n-1$. The following is the outline of the induction step to establish $BT(r, S_n)$, where $r = 12 \cdots n$.

1) Build a tree Ψ rooted from r. The tree Ψ is a subgraph of S_n and has a height of 3.
2) Partition Ψ into a number of disjoint subtrees such that each subtree is associated with one of the substars of S_n as defined in Lemma 1 in a *one-to-one* manner.
3) Following step 2, consider each subtree, say, T of Ψ and its associated substar, say, S_k. We designate a node t as the root of T. As $k \le n-1$, by the induction hypothesis we can construct a spanning tree $BT(t, S_k)$ in S_k. Our final tree $BT(r, S_n)$ is then obtained by attaching each $BT(t, S_k)$ to Ψ through the node t. It is to be noted that our yet-to-be-presented scheme will guarantee that T "matches" $BT(t, S_k)$ in the following sense:

   Tree T is a subgraph of tree $BT(t, S_k)$.

   So the above attaching process will not create any redundant node or cycle, and $BT(r, S_n)$ is really a spanning tree of S_n.

To realize the above steps, we first define some nodes neighboring to r. For each $i = 2 \dots n$ and $j = 2 \dots n$ such that $i \ne j$, define

$$a_i = g_i(r) \tag{1}$$

$$b_{i,j} = g_j(a_i) \tag{2}$$

$$c_{ij} = \begin{cases} g_n(b_{i,j}) & \text{if } j \ne n \\ g_i(b_{i,j}) & \text{if } j = n \end{cases} \tag{3}$$

The tree Ψ is defined as follows:

$$V(\Psi) = \left\{ r, a_i, b_{i,j}, c_{i,j} \mid i = 2 \dots n, j = 2 \dots n, i \ne j \right\}$$

$$E(\Psi) = \left\{ \langle r, a_i\rangle, \langle a_i, b_{i,j}\rangle, \langle b_{i,j}, c_{i,j}\rangle \mid i = 2 \dots n, j = 2 \dots n, i \ne j \right\}$$

For instance, Fig. 1 shows the tree Ψ rooted from $r = 123456$ in an S_6.

Our next job is to partition Ψ into subtrees. As mentioned in earlier step 2), we will also associate each subtree with one of the substars defined in Lemma 1. This is done by the following four rules. (See Fig. 2 for an illustration.)

R1: Let node r be a trivial subtree consisting of only r and be associated with substar R.

R2: For each a_i, construct a tree $Sub(a_i)$ as follows.

- **a)** If $i < n$, let a_i be the root of $Sub(a_i)$. Then, include each $b_{i,j}$ such that $i > j$ into $Sub(a_i)$ and join a_i and $b_{i,j}$ with an edge $\langle a_i, b_{i,j}\rangle$. Finally, associate $Sub(a_i)$ with the substar $B_{1,i}$.
- **b)** If $i = n$, let $Sub(a_n)$ be a trivial tree consisting of only a_n and be associated with substar A.

R3: For each $b_{i,j}$ such that $i < j$, let $b_{i,j}$ be a trivial subtree which is associated with

- **a)** substar $B_{i,j}$ if $j < n$, and
- **b)** substar $C_{1,i}$ if $j = n$.

R4: For each $c_{i,j}$, let $c_{i,j}$ be a trivial tree which is associated with

- **a)** substar $C_{i,j}$ if $j < n$, and
- **b)** substar $C_{j,i}$ if $j = n$.

For instance, the tree Ψ shown in Fig. 1 is partitioned into 32 subtrees. The roots of these subtrees are highlighted by rectangles. Each non-root node (without a rectangle) is connected to its parent in Ψ to form a subtree. Thus, a_4 has two children $b_{4,2}$ and $b_{4,3}$, and this tree is $Sub(a_4)$. Below each rectangle, we indicate the substar associated with the root node as well as the rule used to obtain such association. For example, $Sub(a_4)$ is associated with substar $B_{1,4}$ by applying rule **R2a**.

Finally, we use the induction hypothesis to obtain a spanning tree from each substar and attach these subtrees to Ψ as described in step 3). This completes the construction. The following theorem states some properties of $BT(r, S_n)$.

THEOREM 1. *$BT(r, S_n)$, $n \geq 3$, is a spanning tree of S_n with a height of $\left\lfloor \frac{3(n-1)}{2} \right\rfloor$ and a balance factor of $(\sum_{i=1}^{n-1} i!)/(n-1)!$.*

PROOF. To prove that $BT(r, S_n)$ is a spanning tree, it suffices to show that

1) rules **R1-4** do partition Ψ into disjoint subtrees,
2) rules **R1-4** do associate each subtree with one of the substars defined in Lemma 1 in a one-to-one manner, and
3) the property stated in step 3) is satisfied (i.e., each subtree is a subgraph of the spanning tree in the corresponding substar).

It is trivial to prove 1) and 2). To prove 3), when rule **R2a** is used, the subtree $Sub(a_i)$ is associated with substar $B_{1,i}$. Observe that $a_i = i23 \cdots (i-1)1(i+1)(i+2) \cdots n$ and $B_{1,i} = S_n^{n-i+1}(1^i, (i+1)^{i+1}, (i+2)^{i+2}, \cdots, n^n)$. The root node a_i is indeed in $B_{1,i}$. Furthermore, in the recursive call to construct $BT(a_i, B_{1,i})$, a_i will be extended with $i-2$ children $g_2(a_i)$, $g_3(a_i)$, ..., $g_{i-1}(a_i)$ (observe this from (1), in a recursive sense). These nodes are $b_{i,2}$, $b_{i,3}$, ..., $b_{i,i-1}$, which are exactly the children of a_i in subtree $Sub(a_i)$ according to **R2a**. So 3) is proved when **R2a** is applied. When any rule other than **R2a** is applied, the subtree is a trivial tree. One can easily see that the (trivial) tree is inside the corresponding the substar. Hence $BT(r, S_n)$ is a spanning tree.

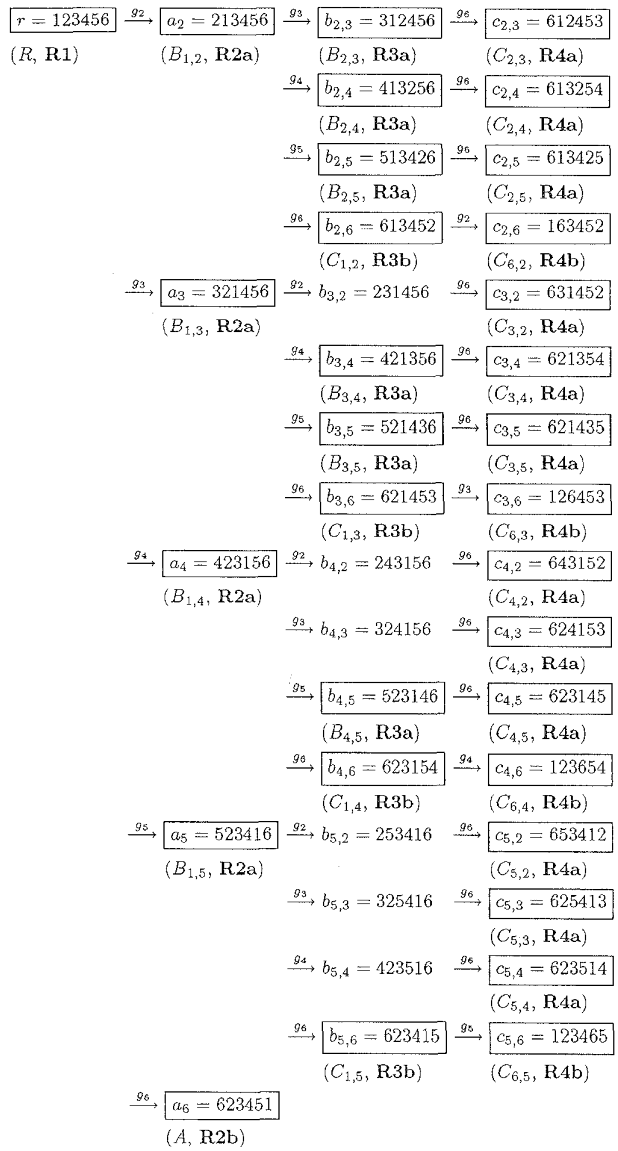


Fig. 1. The tree Ψ in an S_6. Each rectangle indicates a root of some subtree. Indicated inside the parentheses are the associated substars and the rules applied.

The height of $BT(r, S_n)$ can be derived as follows. Let $h(i)$ be the height of $BT(12 \cdots i, S_i)$. We have $h(1) = 0$ and $h(2) = 1$. Consider the outgoing paths leading from node r in the tree Ψ. A path of length 1 must arrive at a substar of dimension at most $n-1$, and a path of length 2 or 3 must arrive at a substar of di-

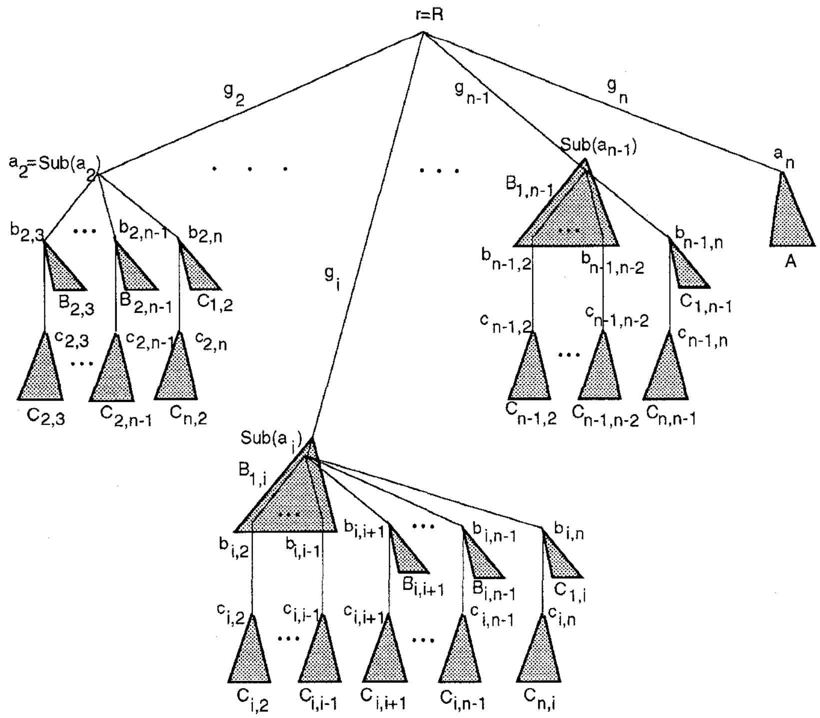


Fig. 2. The construction of a balanced tree $BT(r, S_n)$. Each shaded triangle means a recursive construction of the tree in the corresponding substar.

mension at most $n - 2$. Thus, $h(i) = \max\{1 + h(i-1), 3 + h(i-2)\}$, $i \geq 3$. Solving this recurrence relation, we have $h(n) = \left\lfloor \frac{3(n-1)}{2} \right\rfloor$.

The balance factor of $BT(r, S_n)$ can be derived as follows. Recall that the balance factor is the ratio of the number of nodes in the largest subtree to that in the smallest subtree. The subtree rooted from a_n is the smallest one, which has $(n-1)!$ nodes. The largest subtree is rooted at a_2, which contains a 1-substar, a 2-substar, ..., an $(n-3)$-substar, and $n(n-2)$-substars. The balance factor then follows by counting the numbers of nodes in these substars. □

Note that the above height is minimum because it is equal to the diameter of S_n. The balance factor converges to 1 as n increases.

3.3 Incomplete Stars

Consider an incomplete star $C^{n-1}(k)$, $2 \leq k \leq n-1$ (clearly there is no need to consider $C^{n-1}(n) = S_n$). Given any node r in this incomplete star, let $BT(r, C^{n-1}(k))$ be the graph obtained from $BT(r, S_n)$ by removing all nodes and edges that do not belong to $C^{n-1}(k)$. The following results state that such a graph $BT(r, C^{n-1}(k))$ is indeed a spanning tree in $C^{n-1}(k)$ and is nicely balanced. Note that as $C^{n-1}(k)$ is not node-symmetric any more, it may be necessary to consider different values of r.

LEMMA 2. *The $BT(r, C^{n-1}(k))$ is a spanning tree in $C^{n-1}(k)$.*

PROOF. Let Ψ' be the *graph* obtained from Ψ after removing those nodes and edges that do not belong to $C^{n-1}(k)$. To prove that $BT(r, C^{n-1}(k))$ is a spanning tree, it suffices to show that

1) Ψ' is still a spanning *tree*, and
2) for each subtree, say T, in the original tree Ψ and the substar, say S_m, associated with T, either both T and S_m exist in $C^{n-1}(k)$, or both do not exist.

$C^{n-1}(k)$ consists of those nodes with the last symbols ranging from $n-k+1$ to n. To prove 1), observe the following equalities (note that a_i, $b_{i,j}$, and $c_{i,j}$ are defined with respect to r as in (1)-(3)):

$$a_i[n] = b_{i,j}[n] = r[n], \text{ where } i < n \text{ and } j < n \tag{4}$$

$$b_{i,n}[n] = c_{i,n}[n] = r[i], \text{ where } i < n \tag{5}$$

Since $n-k+1 \leq r[n]$, (4) implies that a_i and $b_{i,j}$ must be in Ψ', where $i < n$ and $j < n$. Equation (5) implies that either both $b_{i,n}$ and $c_{i,n}$ exist in Ψ', or both do not exist, where $i < n$. We thus conclude that no internal node could be removed from Ψ without its children being removed, thus proving part 1). Part 2) can be proved similarly by observing the last symbol of each substar's label. So $BT(r, C^{n-1}(k))$ is a spanning tree. □

THEOREM 2. *Let r be any node in $C^{n-1}(k)$, $n \geq 4$, $2 \leq k \leq n-1$. Rooted at r, tree $BT(r, C^{n-1}(k))$ has a height of $\left\lfloor \frac{3(n-1)}{2} \right\rfloor$ and a balance factor b of*

$$b \leq \begin{cases} \frac{k+1}{k+s} + \frac{\sum_{j=1}^{n-3} j!}{(k+s)(n-2)!} & \text{if } r[1] \leq n-k \\ \frac{n-1}{k-1} + \frac{s\sum_{j=1}^{n-3} j!}{(k-1)(n-2)!} & \text{if } r[1] \geq n-k+1 \end{cases} \tag{6}$$

TABLE 1
THE OBTAINED BALANCE FACTORS
FOR SOME EXAMPLE COMPLETE AND INCOMPLETE STAR GRAPHS

n	4	5	8	10	13	16	20
S_n	1.5	1.37	1.17	1.12	1.091	1.071	1.055
$C^{n-1}(n-1)^\dagger$	1.12	1.10	1.02	1.01	1.007	1.004	1.002
$C^{n-1}(n-1)^\ddagger$	1.75	1.5	1.20	1.14	1.099	1.076	1.058
$C^{n-1}(\lfloor \frac{n}{2} \rfloor)^\dagger$	1.75	1.75	1.30	1.22	1.183	1.134	1.105
$C^{n-1}(\lfloor \frac{n}{2} \rfloor)^\ddagger$	3	4	2.33	2.25	2.4	2.142	2.111

Note that for incomplete stars the values are upper bounds.
The mark † indicates the case when $r[1] \le n-k$, and the mark ‡ indicates when $r[1] \ge n-k+1$.

where $s = 0$ if $k < n-1$, and $s = 1$ if $k = n-1$.

PROOF. It is obvious that the height of $BT(r, C^{n-1}(k))$ can not exceed the height, $\lfloor \frac{3(n-1)}{2} \rfloor$, of S_n. Since $\lfloor \frac{3(n-1)}{2} \rfloor$ is also the diameter of $C^{n-1}(k)$ [6], the height follows.

Now we prove the balance factor of $BT(r, C^{n-1}(k))$ for the case of $r[1] \le n-k$. Since $a_n[n] = r[1]$, node a_n does not exist in Ψ'. Recall that $C^{n-1}(k)$ consists of nodes from the k substars $S_n^1(i^n)$, $i = n-k+1 \dots n$ (each of dimension $n-1$). Consider the distribution of these nodes in Ψ'. By our association rules, nodes in the substar $S_n^1((r[n])^n)$ will be attached to the nodes r, a_i, and $b_{i,j}$ in Ψ', where $i < n$ and $j < n$. The subtree rooted from each a_i will receive $\sum_{j=i-1}^{n-2} j!$ nodes contributed from substar $S_n^1((r[n])^n)$, where $i = 2 \dots n-1$.

The remaining $k-1$ substars (of dimension $n-1$) will *each* be partitioned into $(n-1)$ substars (each of dimension $n-2$). By our association rules, these $(k-1)(n-1)$ substars (of dimension $n-2$) will be evenly distributed to the subtrees rooted at a_i, $i = 2 \dots n-1$. Thus, each subtree will receive either $\lceil \frac{(k-1)(n-1)}{n-2} \rceil$ or $\lfloor \frac{(k-1)(n-1)}{n-2} \rfloor$ number of $(n-2)$-substars. The former number is equal to k, while the latter is equal to $k-1+s$, where s is as defined in the theorem. Combining with the numbers discussed in the previous paragraph, the largest subtree can receive no more than $k(n-2)! + \sum_{j=1}^{n-2} j!$ nodes, and the smallest subtree no less than $(k-1+s)(n-2)! + \sum_{j=n-2}^{n-2} j!$ nodes. The balance factor then follows.

Finally, we consider the case of $r[1] \ge n-k+1$. Similar to the previous case, the subtree rooted from each a_i will receive $\sum_{j=i-1}^{n-2} j!$ nodes contributed from the substar $S_n^1((r[n])^n)$, where $i = 2 \dots n-1$. Node a_n exists in Ψ' and substar $S_n^1((r[1])^n)$ will be attached to a_n. The remaining $k-2$ substars (of dimension $n-1$) will each be partitioned into $(n-1)$ substars (of dimension $n-2$). These $(k-2)(n-1)$ substars (of dimension $n-2$) will be evenly distributed to the subtrees rooted at a_i, $i = 2 \dots n-1$. Hence each a_i will receive either $\lceil \frac{(k-2)(n-1)}{n-2} \rceil$ or $\lfloor \frac{(k-2)(n-1)}{n-2} \rfloor$ number of $(n-2)$-substars. The former number is $k-2+\lceil \frac{k-2}{n-2} \rceil$, while the latter is $k-2$. The largest subtree could be either the one rooted at a_n (which contains an $(n-1)$-substar), or one rooted at some a_i (which may contain at most

$$(k-2+\lceil \tfrac{k-2}{n-2} \rceil)(n-2)! + \sum_{j=1}^{n-2} j! \le k(n-2)! + \sum_{j=1}^{n-3} j!$$

nodes). As the latter will be greater than the former only when $k = n-1$ (i.e., $s = 1$), the number of nodes in the largest subtree cannot exceed $(n-1)! + s\sum_{j=1}^{n-3} j!$. The smallest subtree will have at least $(k-2)(n-2)! + \sum_{j=n-2}^{n-2} j! = (k-1)(n-2)!$ nodes. The balance factor then follows. □

Note that the above height is minimum because the same value has been proved to be the diameter of $C^{n-1}(k)$ in [6]. Also note that the upper term in (6) converges to $\frac{k+1}{k+s}$ as n increases, while the lower term converges to $\frac{n-1}{k-1}$.

4 CONCLUSIONS

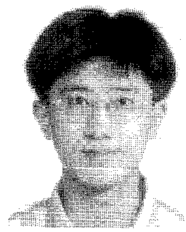
We have shown how to construct balanced spanning trees in complete star graphs and in the class of incomplete star graphs, C^{n-1}. These trees all have the minimum height. Table 1 summarizes the balance factors obtained by our scheme for some example graphs. Being nicely balanced, these trees, when applied to one-to-all personalized broadcast, will incur about equal load on each outgoing link from the root node, thus giving efficient algorithms.

ACKNOWLEDGMENTS

Dr. Jang-Ping Sheu's research is supported by the National Science Council of the Republic of China under Grant #NSC83-0408-E-008-048. Dr. Yu-Chee Tseng's research is supported by the National Science Council of the Republic of China under Grant #NSC85-2213-E-216-021.

REFERENCES

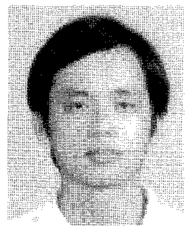
[1] S.B. Akers, D. Harel, and B. Krishnameurthy, "The Star Graph: An Attractive Alternative to the n-Cube," *Proc. Int'l Conf. Parallel Processing*, pp. 393-400, 1987.
[2] S.G. Akl, K. Qiu, and I. Stojmenovic, "Data Communication and Computational Geometry on the Star and Pancake Interconnection Networks," *Proc. Symp. Parallel and Distributed Processing*, pp. 415-422, 1991.
[3] K. Day and A. Tripathi, "A Comparative Study of Topological Properties of Hypercubes and Star Graphs," *IEEE Trans. Parallel and Distributed Systems*, vol. 5, no. 1, pp. 31-38, Jan. 1994.
[4] S.L. Johnsson and C.T. Ho, "Optimal Broadcasting and Personalized Communication in Hypercubes," *IEEE Trans. Computers*, vol. 38, no. 9, pp. 1,249-1,268, Sept. 1989.
[5] J.-S. Jwo, S. Lakshmivarahan, and S.K. Dhall, "Embeddings of Cycles and Grids in Star Graphs," *Proc. Symp. Parallel and Distributed Processing*, pp. 540-547, 1990.
[6] S. Latifi and N. Bagherzadeh, "Incomplete Star: An Incrementally Scalable Network Based on the Star Graph," *IEEE Trans. Parallel and Distributed Systems*, vol. 5, no. 1, pp. 97-102, Jan. 1994.
[7] V.E. Mendia and D. Sarkar, "Optimal Broadcasting on the Star Graph," *IEEE Trans. Parallel and Distributed Systems*, vol. 3, no. 4, pp. 389-396, July 1992.
[8] J. Misic and Z. Jovanovic, "Communication Aspects of the Star Graph Interconnection Network," *IEEE Trans. Parallel and Distributed Systems*, vol. 5, no. 7, pp. 678-687, July 1994.
[9] M. Nigam, S. Sahni, and B. Krishnamurthy, "Embedding Hamiltonians and Hypercubes in Star Interconnection Graphs," *Proc. Int'l Conf. Parallel Processing*, pp. III-340-343, 1990.
[10] Shen, Hu, Cong, Sudborough, Girou, and Bettayeb, "The 4-Star Graph Is Not a Subgraph of Any Hypercube," *Information Processing Letters*, vol. 45, 1993.
[11] J.-P. Sheu, W.-H. Liaw, and T.-S. Chen, "A Broadcasting Algorithm in Star Graph Interconnection Networks," *Information Processing Letters*, vol. 48, pp. 237-241, 1993.
[12] J.-P. Sheu, C.-T. Wu, and T.-S. Chen, "An Optimal Broadcasting Algorithm without Message Redundancy in Star Graphs," *IEEE Trans. Parallel and Distributed Systems*, vol. 6, no. 6, pp. 653-658, June 1995.



Tzung-Shi Chen received the BS degree in computer science from Tamkang University, Republic of China, in 1989, and the PhD degree from the Department of Computer Science and Information Engineering, National Central University, Taiwan, Republic of China, in 1994.

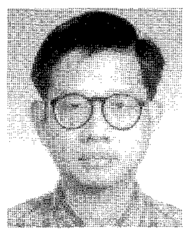
Since June 1996, he has been an associate professor in the Department of Information Management, Chang Jung University, Tainan, Taiwan, Republic of China. His current research interests include parallel and distributed computing, parallelizing compilers, and parallel algorithms.

Dr. Chen is a member of the IEEE Computer Society.



Yu-Chee Tseng received the BS degree in computer science and information engineering from the National Taiwan University, Taipei, in 1985, and the MS degree in computer science from the National Tsing-Hua University, Taiwan, in 1987. He worked for the Wang Laboratory and D-LINK Inc. as a software engineer from 1989 to 1990. He received his PhD in computer and information science from the Ohio State University in January 1994. In the same year, he joined the Department of Computer Science at Chung-Hua Polytechnic Institute, Taiwan, where he is currently an associate professor. Dr. Tseng has served on the Program Committee of the International Conference of Parallel and Distributed Systems, 1996. His research interests include parallel and distributed computing, fault-tolerant computing, parallel computer architecture, and interconnection networks.

Dr. Tseng is a member of the IEEE Computer Society and the ACM.



Jang-Ping Sheu (S'85-M'86) received the BS degree in computer science from Tamkang University, Taiwan, Republic of China, in 1981, and the MS and PhD degrees in computer science from the National Tsing Hua University, Taiwan, Republic of China, in 1983 and 1987, respectively.

Dr. Sheu joined the faculty of the Department of Electrical Engineering, National Central University, Taiwan, Republic of China, as an associate professor in 1987. He has been a full professor there since 1992 in the Department of Computer Science and Information Engineering. From March to June of 1995, he was a visiting researcher at the IBM T.J. Watson Research Center, Yorktown Height, New York. His current research interests include parallelizing compilers, fault-tolerant computing, and interconnection networks.

Dr. Sheu is a member of the IEEE Computer Society and Phi Tau Phi Society. He received the Distinguished Research Award of the National Science Council of the Republic of China in 1993 and 1995.