

Optimum Broadcasting Algorithms in (n, k) -Star Graphs Using Spanning Trees

Jinli Li¹, Manli Chen², Yonghong Xiang^{3,4}, and Shaowen Yao³

¹ Kunming Electric Power Supply Bureau, YNPG, Kunming, 650011, China

² National Computer network Emergency Response technical Team/Coordination Center-Yunnan Branch, Kunming, 650011, China

³ Department of Computer Science, University of Durham, Science Labs, South Road, Durham, EH1 3LE, UK

⁴ School of Software, Yunnan University, Kunming, 650091, China

{lonetimberwolf, chenmanliyn}@163.com, yonghong.xiang@durham.com

Abstract. In a multiprocessor network, sending a packet typically refers to start-up time and transmission time. To optimize these two times, as opposed to earlier solutions, a spanning tree and multiple spanning trees are constructed to solve four types of broadcasting problems in an (n, k) -star graph: one-to-all or all-to-all broadcasting with either one-port or all-port communication model, respectively. Since the proposed spanning tree has an optimal height, both one-to-all and all-to-all broadcasting algorithms achieve nearly optimal start-up time and transmission time under all-port model and one-port model, and optimal transmission time under one-port model. By using multiple spanning trees, both one-to-all and all-to-all algorithms achieve nearly optimal transmission time under all-port model and one-port model.

Keywords: (n, k) -star graph, a spanning tree, multiple spanning trees, one-to-all broadcasting, all-to-all broadcasting, all-port model, one-port model.

1 Introduction

As a new topology of interconnection networks, the (n, k) -star graph has attracted lots of attentions: it not only preserves many appealing properties of n -star: scalability, maximally fault tolerance, partitionability, node-symmetry, but also overcomes the practical drawback with the star graph in that only $\frac{n!}{(n-k)!}$ nodes are involved, compared with $n!$ nodes for an n -star. Some works have been done on this graph, such as basic properties[1],[2],[3], embeddability[3], broadcasting algorithms[4], and so on.

The tree structure has received much interest as a versatile architecture for a large class of parallel processing applications. Spanning trees in particular support communications in different networks: hypercube[5], star graph[6], and (n, k) -star graph[7], and (n, k) -arrangement graph[8].

In this paper, we study one-to-all and all-to-all broadcasting problems in an (n, k) -star graph using packet-switching technique, one-port and all-port communication capabilities are considered.

2 Preliminaries

For simplification, denote the identity node (12...k) as Id and set $\{1,2,\dots,n\}$ as $\langle n \rangle$.

Definition 1. An (n, k) -star graph, denoted by $S_{n,k}$, is an undirected graph. Any node p is denoted by $\{p_1 p_2 \dots p_k \mid p_i \in \langle n \rangle\}$ and $p_i \neq p_j$ for $i \neq j$. Two nodes p and q are adjacent if q is obtained by swapping p_1 with p_i , where $2 \leq i \leq n$, that is, $q = S_i(p_1 p_2 \dots p_i \dots p_k) = p_i p_2 \dots p_1 \dots p_k$.

$S_{n,k}$ is a regular graph of degree $n-1$ and $\frac{n!}{(n-k)!}$ nodes. The diameter is $2k-1$ for $1 \leq k \leq \lfloor \frac{n}{2} \rfloor$, and $k + \lfloor \frac{n-1}{2} \rfloor$ for $\lfloor \frac{n}{2} \rfloor < k < n$. $S_{4,3}$ is illustrated in Figure 1.

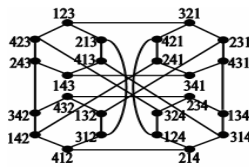


Fig. 1. $S_{4,3}$

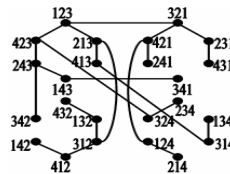


Fig. 2. $SP(S_{4,3})$

Definition 2. If node p has a label of $p_1 p_2 \dots p_k$, we may denote symbol p_i by $p[i]$. Given nodes p_1 and p_2 , $dif(p_1, p_2)$ is the smallest $i > 1$ such that $p_1[i] \neq p_2[i]$.

Steps to correct node $p = p_1 p_2 \dots p_k$ with α internal cycles and β external cycles are computed by the following formulas[1], [2].

For an internal cycle $C_i (1 \leq i \leq \alpha)$ of m_i symbols:

$$steps(C_i, m_i) = \begin{cases} m_i - 1, & p_1 \in C_i \\ m_i + 1, & p_1 \notin C_i \end{cases} \tag{1}$$

For β external cycles of M' symbols totally:

$$steps(\beta, M') = \begin{cases} M' + \beta - 1, & p_1 \in \text{some } C_j \\ M' + \beta + 1, & p_1 \notin \text{any } C_j \end{cases} \tag{2}$$

3 An Optimal Spanning Tree

Definition 3. Given any node p in $S_{n,k}$, the directed graph $SP(S_{n,k})$ is defined such that $V(SP) = V(S_{n,k})$; and for any node $s \in V(S_{n,k}) - \{v\}$, $E(SP)$ contains the directed edge $\langle s, v \rangle$, where s is the parent of node v , defined by $s = G_\alpha(v)$, where

$$\alpha = \begin{cases} dif(v, Id), & v[1] = Id[1] \\ i, & v[1] = Id[1] \wedge 2 \leq i \leq k \\ d[j], & v[1] = Id[j] \wedge k+1 \leq j \leq n \wedge e = 1 \\ \min\{D - d[j], v[1] = Id[j] \wedge k+1 \leq j \leq n \wedge 2 \leq e \leq n-k\} \end{cases}$$

Theorem 1 [7]. For an (n, k) -star graph, there must be a greedy spanning tree $SP(S_{n,k})$ of height $h(SP) = D(S_{n,k})$, and the result is optimal.

Corollary 1. Two nodes are adjacent in $S_{n,k}$, if they are at the same level or two consecutive levels of $SP(S_{n,k})$.

Proof. Suppose that two nodes p and q are adjacent in $S_{n,k}$, and they aren't neither at the same level nor two consecutive levels of $SP(S_{n,k})$. Let q be at the t , $0 \leq t < D(S_{n,k})$, level of $SP(S_{n,k})$ and $d(p, Id) \geq d(q, Id)$. If $d(p, Id) - d(q, Id) \geq 2$, i.e. $d(p, Id) \geq d(q, Id) + 2 = t + 2$. Since $d(p, q) = 1$, $d(q, Id) + d(p, q) = t + 1 < t + 2$, that is, $d(p, Id) \neq d(q, Id) + 1$, which is a contradiction.

So, $0 \leq d(p, Id) - d(q, Id) \leq 1$. $SP(S_{4,3})$ is given in Figure 2.

Corollary 2. The balance factor of $SP(S_{n,k})$ is at most 1, i.e. $0 \leq BF(SP) \leq 1$.

Proof. Suppose that $BF(SP) \geq 2$. Without generality, assume that p and q are two leaves in $SP(S_{n,k})$, since $h(SP) = D(S_{n,k})$, let $d(p, Id) = D(S_{n,k})$. Thus, node p has to satisfy one of following three conditions. We will prove that the proposition that p and q are two leaves and $d(p, Id) - d(q, Id) = 2$ is false under any circumstance. Let e be an external symbol, i be an internal symbol, inv be an invariable symbol and $|x|$ be the size of x .

(1) For $1 \leq k \leq \lfloor \frac{n}{2} \rfloor$, $C(p)$ satisfies one of the following two[2]:

- (1.1) $p_1 = 1$, and the other $k - 1$ symbols are external symbols;
- (1.2) $p_1 \neq 1$, k symbols are external symbols.

For (1.1), informally, $p = (1e_1e_2...e_{k-1})$. From formula (2), $steps(\beta, M') - steps(\beta - 1, M') = 2$. So if $d(p, Id) - d(q, Id) = 2$, q should satisfy that $q_1 \neq 1$, there are $k - 2$ external symbols and there is an invariant, that is, $q = (e_1...e_{x-1}e_xe_{x+1}...e_{k-1})$, where $\exists x \in \{1, \dots, k - 1\}$, $e_x = inv$ and $|e_x| = 1$; $\forall y \in \{1, \dots, k - 1\} - \{x\}$, $|e_y| = 1$. By definition 3, $q = G_\alpha(e_xe_2...e_{x-1}e_1e_{x+1}...e_{k-2})$, so q isn't a leaf.

For (1.2), informally, $p = (e_1e_2...e_k)$. If $d(p, Id) - d(q, Id) = 2$, q should satisfy that $q_1 = 1$, there are $k - 1$ external symbols and there is an invariant, i.e. $q = (1e_1e_2...e_{x-1}e_xe_{x+1}...e_{k-2})$, where $\exists x \in \{1, \dots, k - 1\}$, $e_x = inv$ and $|e_x| = 1$; $\forall y \in \{1, \dots, k - 1\} - \{x\}$, $|e_y| = 1$. By definition 3, $q = G_\alpha(e_xe_1e_2...e_{x-1}1e_{x+1}...e_{k-2})$ isn't a leaf.

(2) For odd n and $\lfloor \frac{n}{2} \rfloor < k < n$, $C(p)$ satisfies the following three[2]:

$p_1 = 1$, there are $n - k$ external symbols, and each of them forms a cycle of length 1, and the rest form internal cycles of length 2.

Same to (1.1), we have the result that $q = G_\alpha(e_xe_1e_2...e_{x-1}1e_{x+1}...e_{k-2})$ isn't a leaf.

(3) For even n and $\lfloor \frac{n}{2} \rfloor < k < n$, $C(p)$ satisfies one of the following two[2]:

(3.1) $p_1 = 1$, there are $n - k$ external symbols, and each of them forms a cycle of length 1; three symbols form one cycle of length 3, and the rest form internal cycles of length 2;

(3.2) $p_1 \neq 1$, there are $n - k$ external symbols, and each of them forms a cycle of length 1; and the rest form internal cycles of length 2.

Same to (1), we have the result that $q = G_\alpha(e_x e_2 \dots e_{x-1} i_1 e_{x+1} \dots e_{n-k-1})$ for (3.1) and $q = G_\alpha(e_x e_2 \dots e_{x-1} i_1 e_{x+1} \dots e_{n-k-1})$ for (3.2) are not leaves.

Thus, the conditions that $d(p, Id) - d(q, Id) \geq 2$ and p, q are two leaves can't be satisfied simultaneously. So, the corollary holds. Especially, $BF(SP(S_{4,2})) = 0$.

4 One-to-All Broadcasting Based on Optimal Spanning Tree

One-to-all broadcasting refers to the problem of sending a message from one source node to all other nodes in the network. In this section, we assume the packet-switching or store-and-forward model, thus, the latency to transmit a packet of b bytes along a link takes $T_s + bT_c$ time, where T_s is the time to start-up the communication link and T_c is the latency to transmit a byte. Under one-port model, a node can send and receive at most one packet at a time, while under all-port model, a node can send and receive packets along all $n - 1$ ports simultaneously.

Lemma 1. A lower bound for one-to-all broadcasting in a store-and-forward $S_{n,k}$ is $\max\{ D(S_{n,k})T_s, \frac{m}{n-1}T_c \}$ under all-port model, and $\max\{ \lceil \log(n!/(n-k)!) \rceil T_s, D(S_{n,k})T_s, mT_c \}$ under one-port model, where m is the size of message M .

4.1 All-Port Model

In the proposed algorithm, time will be slotted by fixed length and all nodes perform broadcast synchronously. In each time slot each node transmits a packet of size $\frac{m}{p(n-1)}$, and p is an integer to be determined later. So each time slot is of length $T_s + \frac{m}{p(n-1)}T_c$.

Algorithm-Broadcasting(one-to-all, all-port)

(1) Slice message M evenly into $p(n - 1)$ parts, each called a "message segment" and of size $\frac{m}{p(n-1)}$;

(2) In each time slot, node Id issues $n - 1$ message segments to the network. A message segment is then propagated along the tree. In each time slot, each node helps propagating all message segments it received in the previous time slot to the subsequent nodes in the corresponding tree.

By theorem 1, $h = D(S_{n,k})$, the total times of all-port broadcasting are: $T = h(T_s + \frac{m}{p(n-1)}T_c) + (p-1)(T_s + \frac{m}{p(n-1)}T_c)$, where the former term is the time for the first packet to arrive at the bottom of the tallest tree and the latter term is due to the pipelined effect. Let the derivative of T with respect to p equal to 0,

$$\frac{\partial T}{\partial p} = T_s - \frac{m(h-1)}{p(n-1)p}T_c \Rightarrow T_s - \frac{m(h-1)}{p(n-1)p}T_c = 0, \text{ therefore, } p = \sqrt{\frac{m(h-1)T_c}{(n-1)T_s}} = O(\sqrt{\frac{m(h-1)T_c}{(n-1)T_s}}).$$

Theorem 2. Under all-port model, one-to-all broadcasting can be performed in $S_{n,k}$ within time $O((D(S_{n,k}) - 1)T_s + \frac{m}{n-1}T_c + 2\sqrt{\frac{m(D(S_{n,k})-1)T_cT_s}{n-1}})$.

Table 1. Comparison of One-to-All Broadcasting Algorithms

Model	Algorithm	Start-up Comp.	Trans. Comp.	Overall Complexity
All-model	optimal	$O(D(S_{n,k})T_s)$	$O(\frac{m}{n-1}T_c)$	$\max\{D(S_{n,k})T_s, \frac{m}{n-1}T_c\}$
	Y.S. Chen	$O((2D(S_{n,k}) - 1)T_s)$	$O(\frac{4m}{n-1}T_c)$	$O(2D(S_{n,k}) - 1)T_s + \frac{4m}{n-1}T_c$
	a spanning tree	$O(D(S_{n,k})T_s)$	$O(\frac{m}{n-1}T_c)$	$O(D(S_{n,k})T_s + \frac{m}{n-1}T_c + 2\sqrt{\frac{m(D(S_{n,k})-1)T_cT_s}{n-1}})$
	multiple spanning trees	$O((2D(S_{n,k}) - 1)T_s)$	$O(\frac{m}{n-1}(\frac{n!}{(n-k)!} - 1)T_c)$	$O((2D(S_{n,k}) - 1)T_s + (\frac{n!}{(n-k)!} - 1)\frac{m}{n-1}T_c)$
One-model	optimal	$\max\{D(S_{n,k})T_s, \lceil \log(\frac{n!}{(n-k)!-1}) \rceil T_s\}$	$O(mT_c)$	$\max\{D(S_{n,k})T_s, mT_c, \lceil \log(\frac{n!}{(n-k)!-1}) \rceil T_s\}$
	Y.S. Chen	$O((2nk - n - k^2)T_s)$	$O(2mT_c + \frac{m(n-k-2)(n-k+1)!}{n!}T_c)$	$O(2nk - n - k^2)T_s + 2mT_c + \frac{m(n-k-2)(n-k+1)!}{n!}T_c$
	a spanning tree	$O(nD(S_{n,k})T_s)$	$O(mT_c)$	$O(n-1)(D(S_{n,k}) - 1)T_s + mT_c + 2\sqrt{m(n-1)(D(S_{n,k}) - 1)T_cT_s}$
	multiple spanning trees	$O(n-1)(2D(S_{n,k}) - 1)T_s)$	$O(mT_c)$	$O(n-1)(2D(S_{n,k}) - 1)T_s + mT_c + 2\sqrt{m(n-1)(2D(S_{n,k}) - 1)T_cT_s}$

4.2 One-Port Model

A node with one-port communication capability can simulate the communication activity of an all-port node in one time slot using $n - 1$ time slots. The simulation can be done as follows: in the first time slot, the one-port node simulates the all-port nodes activity along dimension 1; in the second time slot, the one-port node simulates the all-port nodes activity along dimension 2; etc. By simulation algorithm stated above at every one-port node in $S_{n,k}$, the following theorem is seen.

Algorithm-Broadcasting(all-to-all, one-port)

(1) Slice message M evenly into $p(n - 1)$ parts, each called a “message segment” and of size $\frac{m}{p(n-1)}$;

(2) In each time slot, node Id issues $n - 1$ message segments to the network. A message segment is then propagated along one of trees $SP(R_i(Id))$, $i = 1 \dots n - 1$. A message segment is then propagated along the tree it is issued. In each time slot, each node helps propagating all message segments it received in the previous time slot to the subsequent nodes in the corresponding tree;

(3) Repeatedly perform step 2 until all message segments have been broadcast.

Theorem 3. Under one-port model, one-to-all broadcasting can be performed in $S_{n,k}$ within time $O((n - 1)(D(S_{n,k}) - 1)T_s + mT_c + 2\sqrt{m(n - 1)(D(S_{n,k}) - 1)T_cT_s})$.

Table 2. Comparison of All-to-All Broadcasting Algorithms

Model	Algorithm	Start-up Comp.	Trans. Comp.	Overall Complexity
All-port	optimal	$O(D(S_{n,k})T_s)$	$O(\frac{m}{n-1}(\frac{n!}{(n-k)!}-1)T_c)$	$\max\{D(S_{n,k})T_s, \frac{m}{n-1}(\frac{n!}{(n-k)!}-1)T_c\}$
	a spanning tree	$O(D(S_{n,k})T_s)$	$O(\frac{m}{n-1}(\frac{n!}{(n-k)!}-1)T_c)$	$O(D(S_{n,k})T_s + (\frac{n!}{(n-k)!}-1)\frac{m}{n-1}T_c)$
	multiple spanning trees	$O(2D(S_{n,k})-1)T_s)$	$O(\frac{m}{n-1}(\frac{n!}{(n-k)!}-1)T_c)$	$O((2D(S_{n,k})-1)T_s + (\frac{n!}{(n-k)!}-1)\frac{m}{n-1}T_c)$
One-port	optimal	$\max\{D(S_{n,k})T_s, \lceil \log(n!/(n-k)!-1) \rceil T_s\}$	$O((\frac{n!}{(n-k)!}-1)mT_c)$	$\max\{(\frac{n!}{(n-k)!}-1)mT_c, D(S_{n,k})T_s, \lceil \log(n!/(n-k)!-1) \rceil T_s\}$
	a spanning tree	$O(n(n-1)D(S_{n,k})T_s)$	$O((\frac{n!}{(n-k)!}-1)mT_c)$	$O(n(n-1)D(S_{n,k})T_s + (\frac{n!}{(n-k)!}-1)mT_c)$
	multiple spanning trees	$O((n-1)(2D(S_{n,k})-1)T_s)$	$O((\frac{n!}{(n-k)!}-1)mT_c)$	$O((n-1)(2D(S_{n,k})-1)T_s + (\frac{n!}{(n-k)!}-1)mT_c)$

5 All-to-All Broadcasting Based on Optimal Spanning Tree

All-to-all broadcasting refers to the problem of sending a message from all source nodes to all other nodes in the network, which is $\frac{n!}{(n-k)!}-1$ copies of one-to-all broadcasting problem.

Lemma 2. A lower bound for all-to-all broadcasting in a store-and-forward $S_{n,k}$ is $\max\{(\frac{n!}{(n-k)!}-1)\frac{m}{n-1}T_c, D(S_{n,k})T_s\}$ under all-port model, and $\max\{D(S_{n,k})T_s, \lceil \log(n!/(n-k)!-1) \rceil T_s, (\frac{n!}{(n-k)!}-1)mT_c\}$ under one-port model.

5.1 All-Port Model

LC and EC schemes proposed by Tseng et al.[6] are used to design our all-to-all broadcasting such that each node in $S_{n,k}$ can use $n - 1$ spanning trees.

Theorem 4. Under all-port model, all-to-all broadcasting can be performed in $S_{n,k}$ within time $O(D(S_{n,k})T_s + \frac{m}{n-1}(\frac{n!}{(n-k)!} - 1)T_c)$.

Proof. Assume that in each iteration of time slot, all message segments can be combined into one packet and send at one time. Therefore, the start-up overhead times nearly $D(S_{n,k})T_s$. To propagate a message segment of size $\frac{m}{n-1}$ along a spanning tree of $\frac{n!}{(n-k)!} - 1$ links, network bandwidth of $\frac{m}{n-1}(\frac{n!}{(n-k)!} - 1)T_c$ is required. Totally, there are $(n - 1)(\frac{n!}{(n-k)!} - 1)$ message segments to be broadcast. Thus, the total network bandwidth required is $m(\frac{n!}{(n-k)!} - 1)^2 T_c$. Since the network is evenly loaded at every time step, the bandwidth is evenly distributed to all $(n - 1)(\frac{n!}{(n-k)!} - 1)$ links in the network, Therefore, the transmission time is obtained.

5.2 One-Port Model

Theorem 5. Under one-port model, all-to-all broadcasting can be performed in $S_{n,k}$ within time $O(n(n - 1)D(S_{n,k})T_s + (\frac{n!}{(n-k)!} - 1)mT_c)$.

6 Multiple Spanning Trees

Node p can be also described as $p = p_1 p_2 \dots p_i \dots p_k [p_{k+1} \dots p_n]$ [4]. The congestion of a directed tree is defined to be the maximum number of times the links of these trees overlapping on same edges.

Definition 4. Given any node p , $R_i(p)$ is the node obtained from p by cyclically shifting the label of p to the right by i positions.

Theorem 6. The $n - 1$ spanning trees $SP(R_1(Id))$, $SP(R_2(Id))$, ..., $SP(R_n(Id))$ totally have an edge congestion of i in $S_{n,k}$, when $n \in (i * k, (i + 1) * k]$, $i = 1, 2, \dots$.

Proof. The roots of multiple trees can be represented as $R_i(Id)$, $i \in \{1, 2, \dots, n - 1\}$, i.e. $R_1(Id) = n1 \dots i - 1 \dots k - 1 [k \dots n - 1]$, $R_2(Id) = n - 1n \dots i - 2 \dots k - 2 [k - 1 \dots n - 2]$, ..., $R_n(Id) = 23 \dots i \dots k - 1 [k \dots 1]$. For node $v = v_1 v_2 \dots v_i \dots v_k$, consider its in-degrees:

$$(1) v_1 = R_i(Id)[1]$$

Let $t = R_i(Id)[1]$, then, satisfying the first condition of definition 3. So there exists an edge $\langle s_i, v \rangle$ in $T(R_i(Id))$, where $s_i = G_\alpha(v)$ and $\alpha = dif(v, R_i(Id))$.

Since $i \in \{1, 2, \dots, n-1\}$, there are $n-1$ edges. For any $i, j \in \{1, \dots, n-1\}$, $R_i(Id) \neq R_j(Id)$ for $i \neq j$, so there is no congestion among $n-1$ edges.

(2) $v_1 \neq R_i(Id)[1]$

If $v_1 \neq R_i(Id)[1]$, where $2 \leq i \leq k$, then, satisfying the second condition of definition 3. So there exists an edge $\langle s_i, v \rangle \in T(R_i(Id))$, where $s_i = G_\alpha(v)$ and $\alpha = i$. Thus, there are $k-1$ edges and no congestion. When $k+1 \leq i \leq n$, discuss it by cases:

(2.1) $v_1 = R_i(Id)[n]$ and $n = k+1$

Since $n = k+1$ and $Id = 12\dots i\dots k[k+1]$, then, satisfying the third condition of definition 3. So there exists an edge $\langle s_i, v \rangle \in T(R_i(Id))$, where $s_i = G_\alpha(v)$ and $v = tv_2\dots v_i\dots v_k$, t is the desired symbol of $R_i(Id)[n]$, so $t = d(R_i(Id)[n]) = R_i(Id)[1]$. Thus, there exists an edge of congestion 1 inevitably.

(2.2) $v_1 = R_i(Id)[j]$ and $k < j \leq n$

Since the desired symbol of an external symbol corresponds to one symbol in $\langle k \rangle$, and the fourth condition of definition 3 is satisfied, thus, there are another $n-k$ number of edges. So the congestion is related to $n-k$, that is, Congestion = i , where $n \in (i * k, (i+1) * k]$, $i = 1, 2, \dots$.

Thus, the theorem holds. See figure 3 for example.

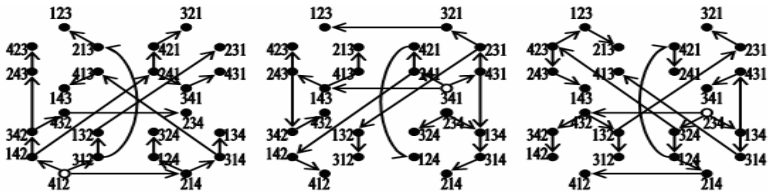


Fig. 3. Three Spanning Trees in $S_{4,3}$ Rooted as $(412), (341), (234)$

7 One-to-All Broadcasting Using Multiple Spanning Tree

Definition 5. Let $p_i : R_i(Id) \Rightarrow Id$ be the path in $T(R_i(Id))$ leading from $R_i(Id)$ to Id . Define $MT(R_i(Id))$ to be the directed graph obtained from $T(R_i(Id))$ by reversing the direction of all edges along the path p_i , where $i \in \{1, \dots, n-1\}$.

Theorem 7. $MT(R_i(Id))$, $i \in \{1, 2, \dots, n-1\}$, is a spanning tree of height h , where $k + D(S_{n,k}) \leq h \leq 2D(S_{n,k})$.

Proof. Only edges along p_i , $i \in \{1, 2, \dots, n-1\}$, in $T(R_i(Id))$ have reversed the direction, so $MT(R_i(Id))$ has $\frac{n!}{(n-k)!} - 1$ edges and all nodes are connected directly or indirectly, thus $MT(R_i(Id))$ is a spanning tree. The height is $D(S_{n,k})$ plus the length of p_i . Since p_i be the path leading from $R_i(Id)$ to Id , thus, $k \leq h(p_i) \leq D(S_{n,k})$, and

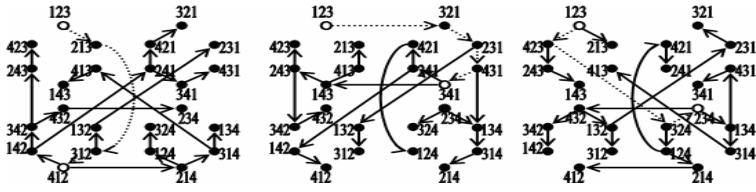


Fig. 4. Multiple Spanning Trees in $S_{4,3}$ Rooted as (123)

$T(R_i(Id))$ is of height $D(S_{n,k})$, so the height of $MT(R_i(Id))$ satisfies the condition that $k + D(S_{n,k}) \leq h \leq 2D(S_{n,k})$. See Figure 4 for example.

7.1 All-Port Model

As Algorithm-Broadcasting(one-to-all, all-port), similar algorithm under all-port for one-to-all broadcasting using multiple spanning trees is available.

Theorem 8. Under all-port model, one-to-all broadcasting can be performed in $S_{n,k}$ within time $O((2D(S_{n,k}) - 1)T_s + \frac{m}{n-1}T_c + 2\sqrt{\frac{m(2D(S_{n,k})-1)T_cT_s}{n-1}})$.

7.2 One-Port Model

As Algorithm-Broadcasting(one-to-all, one-port), similar algorithm under one-port for one-to-all broadcasting using multiple spanning trees is available.

Theorem 9. Under one-port model, one-to-all broadcasting can be performed in $S_{n,k}$ within time $O((n-1)(2D(S_{n,k}) - 1)T_s + mT_c + 2\sqrt{m(n-1)(2D(S_{n,k}) - 1)T_cT_s})$.

8 All-to-All Broadcasting Using Multiple Spanning Trees

Same to all-to-all broadcasting based on the optimal spanning tree, the following theorems are obtained.

Theorem 10. Under all-port model, all-to-all broadcasting can be performed in $S_{n,k}$ within time $O((2D(S_{n,k}) - 1)T_s + \frac{m}{n-1}(\frac{n!}{(n-k)!} - 1)T_c)$.

Theorem 11. Under one-port model, all-to-all broadcasting can be performed in $S_{n,k}$ within time $O((n-1)(2D(S_{n,k}) - 1)T_s + (\frac{n!}{(n-k)!} - 1)mT_c)$.

9 Conclusion

We have shown how to solve various versions of broadcast problems in an (n, k) -star graph by using a spanning tree and multiple spanning trees to simultaneously optimize both transmission time and start-up time.

Finally, we compare our results with other broadcasting algorithms. The numbers of start-up time, transmission time, and overall time complexity are presented in Table 1 and Table 2, respectively.

From Table 1, for one-to-all broadcasting, under all-port model, the algorithm based on an optimal spanning tree achieves nearly optimal start-up time $O(D(S_{n,k})T_s)$ and transmission time $O(\frac{m}{n-1}T_c)$, far lower than $O((2D(S_{n,k})-1)T_s)$ and $O(\frac{4m}{n-1}T_c)$ achieved by the Y.S. Chen's, also $O((2D(S_{n,k})-1)T_s)$ and $O(\frac{m}{n-1}(\frac{n!}{(n-k)!}-1)T_c)$ by using multiple spanning trees. Under one-port model, the algorithm based on an optimal spanning tree achieves start-up time $O(nD(S_{n,k})T_s)$ and nearly optimal transmission time $O(mT_c)$, Y.S. Chen's algorithm achieves start-up time $O((2nk-n-k^2)T_s)$ and transmission time $O(2mT_c + \frac{(n-k-2)(n-k+1)!}{n!}mT_c)$, the algorithm based on multiple spanning trees achieves start-up time $O((n-1)(D(S_{n,k})-1)T_s)$ and nearly optimal transmission time $O(mT_c)$.

From Table 2, for all-to-all broadcasting, under all-port model, the algorithm based on an optimal spanning tree achieves nearly optimal start-up time $O(D(S_{n,k})T_s)$, far lower than $O((2D(S_{n,k})-1)T_s)$ achieved by using multiple spanning trees, as for transmission time $O(\frac{m}{n-1}(\frac{n!}{(n-k)!}-1)T_c)$, both are nearly optimal. Under one-port model, the algorithm based on an optimal spanning tree achieves start-up time $O(n(n-1)D(S_{n,k})T_s)$, higher than $O((n-1)(2D(S_{n,k})-1)T_s)$ achieved by using multiple spanning trees, as for transmission time, both are nearly optimal.

From Table 1 and Table 2, we can see that it is a hard work to optimize both start-up time and transmission time simultaneously. So our algorithms based on an optimal spanning tree are asymptotically optimal. To the best of our knowledge, this is the first work reporting the possibility of embedding multiple $O(n)$ spanning trees in an (n, k) -star graph, while keeping the edge congestion variable. But no more comparative broadcasting algorithms are available in $S_{n,k}$.

Acknowledgments

This work is supported by Natural Science Foundation of Yunnan Province, China, under Grant NO.2004F0006Q.

References

1. Chiang, W.K., Chen, R.J.: The (n, k) -star graph: A generalized star graph. Information Proc. Let. 56, 259–264 (1995)
2. Chiang, W.K., Chen, R.J.: Topological Properties of the (n, k) -star graph. International Journal of Foundations of Computer Science 9(2), 235–248 (1998)
3. Chang, J.-H., Kim, J.: Ring Embedding in Faulty (n, k) -star graphs. In: Proceedings of the Eighth International Conference on Parallel and Distributed Systems (ICPADS'01), pp. 99–106 (2001)

4. Chen, Y.-S., Tai, K.-S.: A Near-Optimal Broadcasting in (n, k) -star graphs. In: ACIS Int'l Conf on Software Engineering Applied to Networking and Parallel/Distributed Computing (SNPD'00), pp. 217–224 (2000)
5. Johnsson, S.L., Ho, C.T.: Optimal Broadcasting and Personalized Communication in Hypercubes. *IEEE Trans. Computers* 38(9), 1249–1268 (1989)
6. Tseng, Y.-C., Sheu, J.-P.: Toward Optimal Broadcast in a star graph Using Multiple Spanning Tree. *IEEE Trans. Computers* 46(5), 593–599 (1997)
7. Li, J.L., Xiang, Y.H., Chen, M.L., Zhou, Y.H.: An Optimal Spanning Tree in (n, k) -star graph Network. *Microelectronics and Computer* 23(9), 168–170 (2006)
8. Chen, Y.S., Juang, T.Y.: Efficient Broadcasting in an Arrangement Graph Using Multiple Spanning Trees. *IEICE Tans. Fundamentals* E83-A(1), 139–149 (2000)