Routing in Unidirectional (n,k)-Star Graphs

Eddie Cheng and Serge Kruk
Department of Mathematics and Statistics,
Oakland University,
Rochester, Michigan USA 48309

ABSTRACT

The class of (n,k)-star graphs and their unidirectional version were introduced as generalizations of star graphs and unidirectional star graphs respectively. In this paper, we substantially improved previously known bound for the the diameter of unidirectional (n,k)-star graphs. The previous bound was $10k-5$ for small k and $5k+5\lfloor(n-1)/2\rfloor$ for large k; the new bound is $7(k-3)+18$. In addition, a distributing routing algorithm is presented, analyzed theoretically for worst-case behaviour and exercised experimentally for average case behaviour.

Keywords: Interconnection networks, unidirectional (n,k)-star graphs, distributed routing.

INTRODUCTION

Many applications require unidirectional interconnection networks. Some recent papers include [2–8, 11–13]. In particular [8, 11] provide specific proposals and applications in which unidirectional graph topologies are appropriate as architectural models.

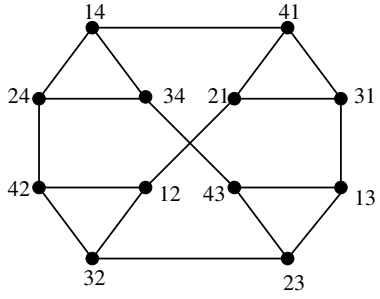
The star graph proposed by [1] has many advantages over the hypercube, such as lower degree and a smaller diameter. One of the main criteria of a good interconnection network topology is that it is maximally edge-connected. So the ideal situation is for an unidirectional graph topology to have the highest possible arc-connectivity. This has been verified for the unidirectional hypercube [11,13] and the unidirectional star graph [3, 12]. An orientation of the star graph was proposed in [12], and they gave an efficient near-optimal distributed routing algorithm for it.

The main drawback of the star graphs is related to its number of vertices: $n!$ for an n-dimensional star graph. For example, the smallest star graph with at least 6000 vertices is a graph with 40320 vertices. The (n,k)-star graphs, which include the star graphs, were introduced in [9, 10] to address this issue. In [7], it is shown that (n,k)-star graphs also have the desired property that they can be oriented to achieve the highest possible arc-connectivity with small diameter. However [7] concentrates on the connectivity issue and treats the routing issue (and the diameter issue) as a by-product. It seems that finding the diameter of unidirectional (n,k)-star graph is more difficult than the undirected version as even the diameter of of the unidirectional star graph remains an open question. In this paper, we

1. give a distributed heuristic routing algorithm for unidirectional (n,k)-star graphs. It is obtained by modifying and extending the algorithm given in [12] for the unidirectional star graphs and by utilizing some basic results given in [7],
2. obtain a theoretical bound for the diameter of unidirectional (n,k)-star graphs that is 30% better than the one in [7], and
3. fine tune the heuristic to obtain a better practical results.

(n,k)-STARS

Basic terminology in graph theory can be found in [14]. An (n,k)-star graph $S_{n,k}$ with $1 \le k < n$ is governed by the two parameters n and k. The vertex-set of $S_{n,k}$ consists of all the permutations of k elements chosen from the ground set $\{1,2,\ldots,n\}$. Two vertices $[a_1,a_2,\ldots,a_k]$ and $[b_1,b_2,\ldots,b_k]$ are adjacent if one of the two conditions holds: (1) There exists r, $2 \le r \le k$ such that $a_1 = b_r$, $a_r = b_1$ and $a_i = b_i$ for $i \in \{1,2,\ldots,k\} \setminus \{1,r\}$. (2) $a_i = b_i$ for $i \in \{2,\ldots,k\}$, $a_1 \ne b_1$. Every vertex has $k-1$ neighbours via adjacency rule (1) and $n-k$ neighbours via adjacency rule (2). Adjacency rule (1) is precisely the rule for the k-dimensional star graph; an edge produced this way is a *star-edge*, it will be called an *i-edge* if the exchange is between position 1 and position i where $i \in \{2,3,\ldots,k\}$. A *residual-edge* is an edge produced by adjacency rule (2). The figure below illustrates $S_{4,2}$. (We note that for convenience, we write the (n,k)-permutation $[i,j]$ as ij.) Note that given an edge in $S_{n,k}$ with the labelings of its two end-vertices, one can immediately determine whether it is a star-edge or a residual-edge.



$S_{4,2}$

The family of $S_{n,k}$ generalizes the star graph since $S_{n,n-1}$ is isomorphic to the star graph S_n (an $(n-1)$-permutation on an n-set is really a permutation on n elements). For $S_{n,n-1}$, that is, the star graph S_n, the unique residual-edge for each vertex is its *n-edge*. The (n,k)-star graph $S_{n,k}$ has $n!/(n-k)!$ vertices and is an $(n-1)$-regular graph. Other properties of (n,k)-star graphs can be found in [7, 9, 10]. $S_{n,k}$ has two important classes of subgraphs.

1. Let $\{x_1, x_2, \ldots, x_k\} \subseteq \{1, 2, \ldots, n\}$ with $k \geq 3$. Let G be the subgraph of $S_{n,k}$ induced by vertices whose labelings are permutations of $x_1, x_2, \ldots, x_k$. Then G is isomorphic to S_k. It is called a *fundamental star* and $S_{n,k}$ has $\binom{n}{k}$ fundamental stars.

2. Let $\{x_2, x_3, \ldots, x_k\} \subseteq \{1, 2, \ldots, n\}$. Let G be the subgraph of $S_{n,k}$ induced by vertices of the form $[y_1, x_2, x_3, \ldots, x_k]$ where $y_1 \in \{1, 2, \ldots, n\} \setminus \{x_2, x_3, \ldots, x_k\}$. Then G is isomorphic to K_{n-k+1}, the complete graph on $n-k+1$ vertices. It is called a *fundamental clique* and $S_{n,k}$ has $\binom{n}{k-1}(k-1)! = \frac{n!}{(n-k+1)!}$ fundamental cliques.

DIRECTED (n,k)-STARS

In this section, we give the orientation proposed by [7]. Given an (n,k)-star graph $S_{n,k}$, we map each vertex to a unique full permutation of $\{1, 2, \ldots, n\}$. Suppose a vertex in $S_{n,k}$ has the labeling $[a_1, a_2, \cdots, a_k]$. Then the unique permutation on $\{1, 2, \ldots, n\}$ associated with it is $[a_1, a_2, \cdots, a_k, x_1, x_2, \cdots, x_{n-k}]$ where $\{x_1, x_2, \ldots, x_{n-k}\} = \{1, 2, \ldots, n\} \setminus \{a_1, a_2 \ldots, a_k\}$ and $x_1 < x_2 < \cdots < x_{n-k}$. We call a vertex *odd* if its associated permutation is odd and *even* if its associated permutation is even. We note that under this definition, a a star-edge is still between an odd vertex and an even vertex. This definition is consistent with the basic properties and terminology of star graphs. The idea used in [7] was to orient the edges in the fundamental stars and fundamental cliques semi-independently but carefully combine the choices to maximize arc-connectivity. Algorithm 1 gives the orientation rule. (For the boolean functions even$(\cdot)$ and odd$(\cdot)$, they use the usual rule if the argument is an integer. If it is a vertex of the (n,k)-star graph, it uses the above definition. The parity$(\cdot)$ function, whose input are vertices of (n,k)-star graph, uses the above definition.)

We will denote this orientation of $S_{n,k}$ by $\overrightarrow{S_{n,k}}$. If $k = n-1$, this reduces to the unidirectional star graph given in [12]. One can check that $\overrightarrow{S_{n,k}}$ is a directed graph with odd (even) vertices having in-degree $\lceil \frac{n-1}{2} \rceil$ ($\lfloor \frac{n-1}{2} \rfloor$)and out-degree $\lfloor \frac{n-1}{2} \rfloor$ ($\lceil \frac{n-1}{2} \rceil$). Results on arc-connectivity are given in [7] (and [3] for the case $k = n-1$).

Algorithm 1 Orienting edge $e = \{\pi_a, \pi_b\}$

if e is a star-edge: i-edge even$(\pi_a) \wedge$ odd(π_b) **then**
  if (even(i)) **then**
    $\pi_a \rightarrow \pi_b$ **else** $\pi_b \rightarrow \pi_a$
  end if(*Day-Tripathi rule*)
else
  $\{e$ is a residual-edge, $\pi_a = [x, a_2, \ldots, a_k]\}$
  $\{\pi_b = [y, a_2, \ldots, a_k]$: Assume $x < y\}$
  $\{z = \min(\{1, 2, \ldots, n\} \setminus \{a_2, \ldots, a_k\})\}$
  $\{\pi_z = [z, a_2, \ldots, a_k]\}$
  if even$(n-k) \vee$ (odd$(n-k) \wedge$ (even$(n-1)$ $\wedge$ odd$(\pi_z)) \vee$ (odd$(n-1) \wedge$ even$(\pi_z)))$ **then**
    if parity$(\pi_a) \neq$ parity(π_b) **then**
      $\pi_a \rightarrow \pi_b$
    else
      $\pi_b \rightarrow \pi_a$
    end if
  else
    if parity$(\pi_a) =$ parity(π_b) **then**
      $\pi_a \rightarrow \pi_b$
    else
      $\pi_b \rightarrow \pi_a$
    end if
  end if
end if

If $k \neq n-1$, [7] showed that every residual-arc is on a small directed cycle (length 3 or 4) and such a cycle can be found easily. Suppose π_y and π_x are ends of a residual-arc and we want to route from π_y to π_x. If the arc is $\pi_y \rightarrow \pi_x$, then it is easy (one step); otherwise, we can route around this small cycle in two or three steps. This fact will be useful in the next section. We denote a path obtained this way by CycleMove(π_y, π_x). If $n-k$ is even, then CycleMove produces a path of length at most two; otherwise, at most three. Details of the CycleMove routine can be found in [7].

ROUTING

Since the unidirectional (n,k)-star graphs reduce to the undirectional star graphs if $k = n-1$, it is natural to develop an algorithm that coincides with the algorithm given by [12] if $k = n-1$. From now on we will assume $k \geq 3$ as certain structural properties will not hold otherwise. (See [7].)

The general structure of the Algorithm 2 is to sweep from position k down to position 3. The last 3 positions are dealt with separately at the end. A vertex is called *i-good* if the unique i-edge is directed away from this vertex where $i \geq 2$. If π is i-good, then we may move from $\pi = [a_1, a_2, \ldots, a_i \ldots, a_k]$ to $[a_i, a_2, \ldots, a_1, \ldots, a_k]$. This is a *StarMove(i, π).* It is easy to see that if a vertex is not i-good, then it is $(i-1)$-good with $i \geq 3$; moreover, the resulting vertex is i-good after an $(i-1)$-move. It is also easy to

see that if a vertex is i-good with $i \geq 4$, then it is $(i-2)$-good.

Algorithm 2 Mainline
Routing $\pi_a = [a_1, a_2, \ldots, a_k]$ to $\pi_b = [b_1, b_2, \ldots, b_k]$

if (π_a is not k-good) **then**
  (π_c:=StarMove$(k-1, \pi_a)$)
else
  $\pi_c := \pi_a$
end if
for $(i = k; i > 3; i = i-1)$ **do**
  {At stage i, current vertex is $\pi_c := [c_1, c_2, \ldots, c_i, b_{i+1}, \ldots, b_k]$}
  if $b_i \in \{c_1, c_2, \ldots, c_i\}$ **then**
    {Assume $b_i == c_j$}
    π_f:=StarManeuver(π_c, i, j)
  else
    π_f:=ResidualManeuver(π_c, b_i, i, j)
  end if
end for
Lastthree(π_f, π_b)

To ensure that Algorithm 2 is at a k-good vertex at the beginning of stage k, it applies StarMove$(k-1, \pi_a)$ if π_a is not k-good. The algorithm enters stage i at $\pi_c = [c_1, c_2, \ldots, c_i, b_{i+1}, \ldots, b_k]$. We consider two cases, $b_i \in \{c_1, c_2, \ldots, c_i\}$ and $b_i \notin \{c_1, c_2, \ldots, c_i\}$. They correspond, respectively to StarManeuver and ResidualManeuver.

Algorithm 3 A StarManeuver(π_c, i, j) [12]

if $i == j$ **then**
  π_c:=StarMove$(i-2, \pi_c)$
else
  if $j == 1$ **then**
    π_c:=StarMove(i, π_c)
  end if
  if $i + j$ is odd **then**
    π_c:=StarMove(i, π_c);
    π_c:=StarMove(j, π_c);π_c:=StarMove(i, π_c);
  else
    π_c:=StarMove(j, π_c);π_c:=StarMove$(i-1, \pi_c)$;
    π_c:=StarMove(i, π_c);
    π_c:=StarMove$(i-1, \pi_c)$;π_c:=StarMove(i, π_c);
  end if
end if
return π_c

Suppose $c_j = b_i$. Then all the moves at this stage are within a fundamental star (StarManeuver). Note that the StarMove$(i-2, \pi_c)$ for the case $j = i$ is to ensure that the vertex at the end of this stage is $(i-1)$-good. Hence a maximum of 5 steps in such a stage. In the second case, $b_i \notin \pi_b$ (ResidualManeuver), the first step is to route π_c to $[b_i, c_2, \ldots, c_i, b_{i+1}, \ldots, b_k]$. If this cannot be done in one step (the arc is oriented in the opposite way), then we route around a small cycle using CycleMove in a fundamental clique. Now a series of StarMoves are used to put b_i in the ith position. Hence a maximum of 6 steps if $n-k$ is even and 7 steps if $n-k$ is odd. Readers may wonder what happen if $k = n-1$. In this case, the ResidualManeuver will not work as the validity of CycleMove is based on $k \neq n-1$. However, one can view a residual-edge in this case as a star-edge (n-edge). So there will be no ResidualManeuvers. Of course, in this case, this reduces to the algorithm given in [12].

Algorithm 4 ResidualManeuver(π_y, x, i, j)

$\pi_x := [x, y_2, \ldots, y_k]$; CycleMove$(\pi_y, \pi_x)$
{Assume $\pi_y = [y_1, y_2, \ldots, y_k]$}
if (π_x is i-good) **then**
  π_c:=StarMove(i, π_x)
else
  π_c:=StarMove$(i-1, \pi_x)$;π_c:=StarMove(i, π_c);
  π_c:=StarMove$(i-1, \pi_c)$;π_c:=StarMove(i, π_c);
end if
return π_c

After stages $k, \ldots, 4$, the algorithm arrives at a vertex of the form $[-, -, -, b_4, \ldots, b_k]$. Now the algorithm will route in $\overrightarrow{S_{n-k+3,3}}$ or $\overrightarrow{S_3}$. This is called *Lastthree(π_f, π_b)* in Algorithm 2. It is easy to route in $\overrightarrow{S_3}$ and its diameter is 5. Proposition 1 gives a bound of the diameter of $\overrightarrow{S_{n-k+3,3}}$.

Proposition 1 *Let $q \geq 4$. Then the diameter of $\overrightarrow{S_{q,3}}$ is at most 14 if q is odd and is at most 17 if q is even.*

Proof: Suppose we want to route from π_c to π_d in $\overrightarrow{S_{q,3}}$. Let the three symbols of π_d be a_1, a_2, a_3. Then we only have to look at 8 cases to route π_c to a vertex with symbols a_1, a_2, a_3 which in turn can route to π_d in at most 5 steps. We will use the term *CycleMove* to indicate routing through a cycle if necessary in a fundamental clique. In the following list, we use + to mean the element belongs to $A = \{a_1, a_2, a_3\}$ and $-$ otherwise.

1. $\pi_c = [+, +, +]$. We are done.
2. $\pi_c = [-, +, +]$. Apply CycleMove to move the other symbol of A to the 1st position.
3. $\pi_c = [+, -, +]$. If it is 2-good, apply the sequence 2-move, a suitable CycleMove; otherwise (it is 3-good), apply the sequence 3-move, 2-move, a suitable CycleMove.
4. $\pi_c = [+, +, -]$. Symmetric to the above case.
5. $\pi_c = [-, -, +]$. Apply a suitable CycleMove and we are back to case 3.
6. $\pi_c = [-, +, -]$. Apply a suitable CycleMove and we are back to case 4.

7. $\pi_c = [+,-,-]$. It is either 2-good or 3-good. Hence in one step, we are back to one of the above two cases.

8. $\pi_c = [-,-,-]$. Apply a suitable CycleMove and we are back to the above case.

Since a CycleMove is at most 3 steps if $q-3$ is even and at most 2 steps if $q-3$ is odd, the result follows.

Lastthree(π_f, π_b) can easily be constructed from the proof of Proposition 1, so we omit the details. Hence we have a distributed routing algorithm and the following result.

Theorem 2 *Suppose $k \geq 3$. If the routing from π_a to π_b in $\overrightarrow{S_{n,k}}$ by the algorithm uses α residual-maneuvers then the number of steps is at most $5(k-3)+\alpha+15$ if $n-k$ is even and at most $5(k-3)+2\alpha+18$ if $n-k$ is odd. In particular, the diameter of $\overrightarrow{S_{n,k}}$ is at most $6(k-3)+15$ if $n-k$ is even and at most $7(k-3)+18$ if $n-k$ is odd.*

This result is much better than the one given in [7]: $10k-5$ if $1 \leq k \leq \lfloor n/2 \rfloor$ and $5k+5\lfloor (n-1)/2 \rfloor$ for $\lfloor n/2 \rfloor \leq k \leq n-1$. We also observe that α may not be the number of symbols in π_b that are not in π_a. During ResidualManeuvers, some of the symbols that are common to π_a and π_b may be moved out of the labelings of the current vertex. That does not affect the worst-case analysis given above, but in practice such cases could be recognized and handled more efficiently.

HEURISTICS

First, we note that the ordering of the stages in the algorithm given in above is unimportant. Next we note that if a vertex is not i-good, then it is j-good if i and j have different parity and that $1 \notin \{i,j\}$. Since we are not following the decreasing order, we no longer need to ensure a vertex is $(i-1)$-good at the end of stage i. Recall that our destination vertex is $[b_1, b_2, \ldots, b_k]$. Suppose the current vertex is $\pi = [c_1, c_2, \ldots, c_k]$. Let $B_\pi = \{i : i \geq 2, c_i \neq b_i\}$. Position i in B_π is called *1-eligible* if $c_1 = b_i$ and π is i-good. It is called *2s-eligible* if there is a $j \in \{2, \ldots, k\}$ such that $c_j = b_i$, i and j are of different parity and π is j-good. It is called *2r-eligible* if $b_i \notin \{c_1, c_2, \ldots, c_k\}$, the arc is directed from π to $[b_i, c_2, \ldots, c_k]$ and $[b_i, c_2, \ldots, c_k]$ is i-good. We note that we can move b_i into the ith position in 1 or 2 moves in these situations. If $i \in B_\pi$ and it does not belong to one of the above cases, then i is *regular-eligible*. We note that such a position can be corrected using the original algorithm with the convention that a StarMove($i-1, \pi_c$) is replaced by a StarMove(j, π_c) where $j \in B_\pi$ with i, j having different parity and π_c is the current vertex. These observations give the following heuristic: While B_π has at least two even elements or at least two odd elements, we look for a position with the preference order being 1-eligible, 2s-eligible, 2r-eligible and regular-eligible. When this fails, we note that B_π has at most one odd element and at most one even element. We can now route in $\overrightarrow{S_{q,3}}$ or $\overrightarrow{S_3}$. After some experimentations, we tweaked it to include the following: the algorithm performs 1 or 2 StarMoves moves within a fundamental star so that the 1st position is not an element of the destination before a residual-maneuver.

Table 1 gives the comparison between the path length given by the heuristics and the true diameter. To compute the latter, we computed the all-pairs shortest paths. We note that unlike the unidirectional star graphs, the unidirectional (n,k)-star graphs are far from symmetric making the computations rather tedious. For $k = n-1$, where the $\overrightarrow{S_{n,n-1}}$ graph is isomorphic to the $\overrightarrow{S_n}$ graph studied by Day and Tripathi [12], our algorithm is almost identical to theirs except for the preferred ordering of the main while loop. The results are therefore not surprisingly almost identical (a worst bound on the diameter but better average distance), as reported in Table 1 on the diagonal.

The code to reproduce all experiments is available at `personalwebs.oakland.edu/~kruk/research` for download.

CONCLUSIONS

Unidirectional interconnection networks are applicable in many settings. In the note, we use two known ingredients, namely, a heuristic for the unidirectional star graphs and the fact that a properly directed fundamental clique has a small cycle for every arc, to obtain a routing algorithm for the unidirectional (n,k)-star graphs that is roughly 30% better than the previously published algorithms.

In [7], the definition of $\overrightarrow{S_{n,k}}$ actually allows several variations of the orientation and their results are independent of the variations. Algorithm 1 gives one such variation. In our theoretical analysis of Algorithm 2, the only fact that we require from the directed fundamental clique is that it contains a small directed cycle (to allow CycleMove). Since this fact is common among all variations, Theorem 2 is true for every variation. It will be interesting to see whether one variation is more convenient than another under other properties.

References

[1] S.B. Akers, D. Harel, and B. Krishnamurthy. The star graph: An attractive alternative to the n-cube. *Proc. Int'l Conf. Parallel Processing*, pages 393–400, 1987.

[2] Y. Chen, E. Cheng, M.J. Lipman, and S.G. Kruk. A note on the diameter of unidirectional split-stars. *Congressus Numerantium*, 163:49–56, 2003.

[3] E. Cheng and M.J. Lipman. On the Day-Tripathi orientation of the star graphs: connectivity. *Inform. Proc. Lett.*, 73:5–10, 2000.

[4] E. Cheng and M.J. Lipman. Orienting split-stars and alternating group graphs. *Networks*, 35:139–144, 2000.

[5] E. Cheng and M.J. Lipman. Orienting the arrangement graphs. *Congressus Numerantium*, 142:97–112, 2000.

[6] E. Cheng and M.J. Lipman. Connectivity properties of unidirectional star graphs. *Congressus Numerantium*, 150:33–42, 2001.

[7] E. Cheng and M.J. Lipman. Unidirectional (n, k)-star graphs. *Journal of Interconnection Networks*, 3:19–34, 2002.

[8] S.C. Chern, J.S. Jwo, and T.C. Tuan. Unidirectional alternating group graphs. In *Computing and combinatorics (Xi'an, 1995), Lecture Notes in Comput. Sci., 959*, pages 490–495. Springer, Berlin, 1995.

[9] W.K. Chiang and R.J. Chen. The (n, k)-star graph: a generalized star graph. *Inform. Proc. Lett.*, 56:259–264, 1995.

[10] W.K. Chiang and R.J. Chen. Topological properties of the (n, k)-star graph. *Intern. J. Found. of Comp. Sci.*, 9:235–248, 1998.

[11] C.H. Chou and D.H.C. Du. Unidirectional hypercubes. *Proc. Supercomputing'90*, pages 254–263, 1990.

[12] K. Day and A. Tripathi. Unidirectional star graphs. *Inform. Proc. Lett.*, 45:123–129, 1993.

[13] J.S. Jwo and T.C. Tuan. On container length and connectivity in unidirectional hypercubes. *Networks*, 32:307–317, 1998.

[14] D.B. West. *Introduction to Graph Theory*. Prentice Hall, 1996.

	k =3	4	5	6	7	8
	4.79/4.46/5.33					
4	10/9/11					
	5.34/4.61	6.19/5.33/6.2				
5	12/8	15/10/12				
	6.06/4.97	7.90/6.33	8.27/7.14/8.66			
6	15/8	18/12	20/11/16			
	6.51/5.21	8.65/6.36	9.61/7.44	9.76/8.27/9.85		
7	14/9	20/11	24/13	25/14/19		
	6.92/5.38	9.12/6.88	10.70/7.88	11.72/8.97	11.68/9.76/11.97	
8	15/9	20/12	26/13	29/15	30/16/24	
	7.22/5.61	9.50/6.87	11.05/8.26	12.62/9.25	13.34/10.24	13.21/11.85/13.68
9	14/9	21/11	25/14	30/15	34/17	35/18/27

Table 1: Average algorithm routing distance / Average distance / Average case from [12]
Worst-case algorithm routing distance / Diameter/ Worst-case from [12]