BROADCASTING IN CYCLES WITH A SINGLE CHORD

Applied Mathematics/Discrete Mathematics

Author: Dr. Lisa L. Kovalchick, Ph. D.
Associate Professor
Mathematics, Computer Science and Information Systems Department
California University of Pennsylvania
250 University Avenue
California, PA 15419

kovalchick@calu.edu

Abstract

This paper discusses the dissemination of information in a communication network. Specifically, we use a cycle to model the communication network. We discuss the potential benefit of inserting a chord into the cycle in order to increase the rate of dissemination and present an algorithm for disseminating information in such a graph, using a technique known as broadcasting.

Introduction

Broadcasting is the process of information dissemination in which one node, the originator, knows one or more pieces of information and using a series of calls must inform every other node in the network of this information. We assume that at any given time, a node can communicate one message by acting as either a sender or receiver, but not both. We also assume that the originator will only have a single message to disseminate.

The current push to obtain faster algorithms for many scientific problems often results in the use of either parallel or distributed computing, the running time of algorithms which employ such computing techniques is often determined by the communication time between processors. If it is possible to lower the amount of time taken for processor communication, it is often possible to decrease the running time of an algorithm. Therefore, broadcasting algorithms are an important part of current computing.

The problem of broadcasting was first formulated in 1977, by Slater, Cockayne and Hedetniemi who studied the amount of time needed for a single person to share a single piece of information with everyone else in a network. Today, many different broadcast models exist; [HHL88] provides a comprehensive investigation of current broadcast models.

We use a connected undirected graph $G = (V, E)$ to model the communication network, where V represents the set of n nodes and E represents the set of edges (i.e., communication links) between pairs of nodes. A broadcast algorithm is defined as a sequence of calls with the following constraints.

1. A message must arrive at a node before that node can pass the message on to another node.
2. At any given time unit, a node may act as either a sender or receiver of a message, but not both.
3. Each node receives the message once.
4. At the end of a broadcast algorithm, every node in the network has received the message.

Given a connected undirected graph $G = (V, E)$ and an originator, node u, we define the broadcast time of node u, ($b(u)$) to be the minimum number of time units required to complete broadcasting from node u. We also define the broadcast time of the graph G ($b(G)$) to be the maximum broadcast time of any node u in G.

Observe that when broadcasting begins, only a single node (u) has the message and according to the constraints, the number of informed nodes can at most double at each step (i.e., each informed node communicates with an uninformed node at each step), thus $b(u) \geq \lceil \log n \rceil$. Obviously, broadcasting can be completed in minimum time in a complete graph, although, when the number of nodes, n, is large the complete graph requires too many communication links (edges) in order to be practical. For this reason, researchers have investigated broadcasting in several other classes of graphs, including grids [FH78, K79, P80], cycles [S99, W99], trees [P81, SCH81], etc.

A cycle, C_n, is defined as an undirected connected graph containing n nodes and n edges, where the edges connect the nodes in a closed chain. We say that any two nodes of a graph are neighbors, if they are connected by an edge. The most obvious algorithm for broadcasting in a cycle is for every informed node to pass the message along to one of its uninformed neighbors at each time step, if such a neighbor exists; we refer to this algorithm as the Chordless Cycle Algorithm.

Using the Chordless Cycle Algorithm, the time to broadcast in a cycle with an even number of nodes is equal to the diameter of the graph (i.e., $\left\lfloor\frac{n}{2}\right\rfloor$) and the time to broadcast in a cycle with an odd number of nodes is equal to the diameter of the graph plus one (i.e., $\left\lfloor\frac{n}{2}\right\rfloor$ +1).

Recall that during broadcasting, at each time step all informed nodes may pass the message along to one of their uninformed neighbors; thus, the number of nodes informed at each successive time step can at most double. Table 1 shows the maximum total number of nodes informed at time steps 0 through 5 and the maximum number of new nodes informed at each time step.

Table 1: Maximum number of nodes informed at time steps 0 through 5.

Time	Total Number of Informed Nodes	Number of Newly Informed Nodes
0	1	
1	2	1
2	4	2
3	8	4
4	16	8
5	32	16

The slow broadcast time of cycles is due to the fact that after time 2, at most 2 new nodes can be informed at each time step. This occurs because each node of a cycle has exactly 2 neighbors and all nodes, except the originator, must be informed by one of their neighbors, leaving only a single neighbor for each node to inform.

To improve the broadcast time of cycles, chords must be added. In this paper, we discuss the placement of a single chord in a cycle. Then, we develop an algorithm for broadcasting in cycles containing a single chord.

Problem Definition

Formally, we define our problem as follows: *Let C_n be* a *cycle with n nodes and n edges and let there be* a *chord (an edge not in C_n whose endpoints lie in C_n) connecting* a *pair of nodes in the cycle, can we develop* a *broadcast algorithm for such* a *graph?*

Placement of the Chord

According to the definition of a cycle, each node of the cycle has exactly 2 neighbors. Observe that the addition of a chord connecting 2 nodes of a cycle will increase the number of neighbors of both the nodes forming the endpoint of the chord by one neighbor each. This allows for 2 nodes of the cycle to each inform an additional node (as long as that node was not already informed) during the broadcasting process. Thus, the addition of a chord has the potential to decrease the time to broadcast compared to the same size cycle without a chord.

Table 2 compares the number of nodes informed at each time step in a chordless cycle to the maximum number of nodes that can be informed at each time step. Observe that time 3 is the first place that the number of nodes informed by the Chordless Cycle Algorithm differs from the maximum number of nodes that can be informed at that time step. We want to increase the number of nodes informed as early as possible because this increase can increase the number of nodes informed at each additional time step. By adding a chord to a cycle, we are able to increase the number of nodes informed at time 3 from 2 to 4 nodes (see Figure 1). The 2 additional nodes informed at time 3 then have the ability to each inform an uninformed neighbor at time 4, which can increase the number of nodes informed at time 4 by at most 2. At all times after time 4, at most 4 new nodes can be informed, since after time 3, no remaining nodes have more than 2 neighbors.

Table 2: Comparison of nodes informed at time steps 0 through 5.

Time	Number of Newly Informed Nodes in a Chordless Cycle	Maximum Number of Newly Informed Nodes
0		
1	1	1
2	2	2
3	2	4
4	2	8
5	2	16

When broadcasting in a cycle with a single chord, in order to maximize the number of nodes informed at time 3, thus, increasing the total number of nodes informed at each time step after time 2, we must make the originator one of the endpoints of the chord (this can be seen by comparing Figure 1 with Figure 2).

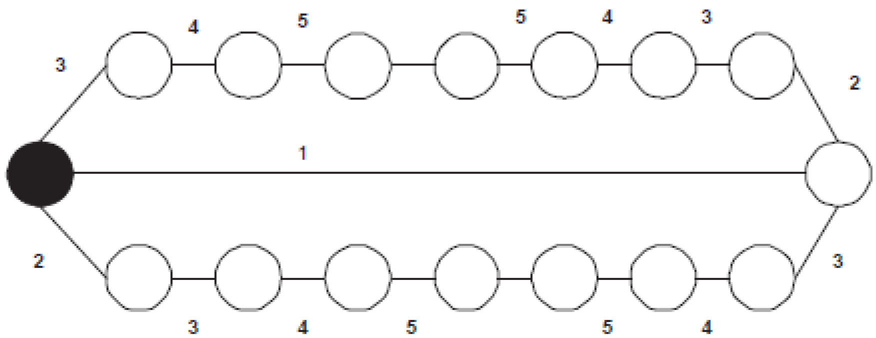

Figure 1: Sending the message across the chord as soon as possible.

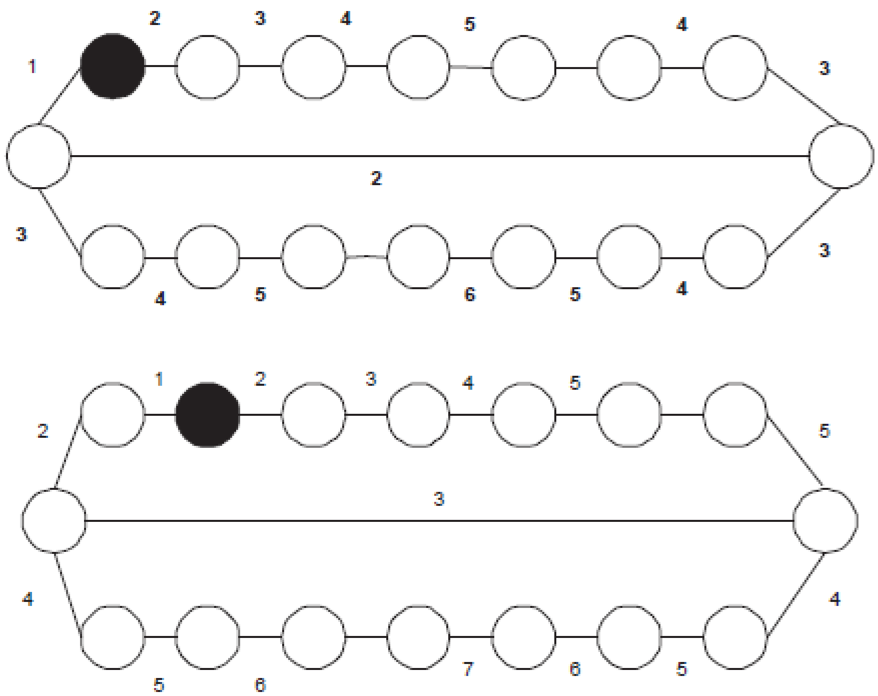

Figure 2: Delays in sending the message across the chord.

The next section presents an algorithm for broadcasting in a cycle containing a single chord with the originator as an endpoint of the chord.

The Equal Split Chord Originator Algorithm

Our algorithm considers the case in which the chord divides the cycle in half or, in the case of an odd number of nodes, into two cycles which differ in size by a single node. We assume that the originator is a node that forms one of the endpoints of the chord. We call this the Equal Split Chord Originator (ESCO) algorithm. Observe that when the chord is added to the cycle, it creates two cycles which share a single edge (call these cycles *A* and *B*). We label the cycles such that cycle *A* will always be either the same size as cycle *B* or, if the original cycle contains an odd number of nodes, cycle *A* will contain one more node than cycle *B*. ESCO proceeds by first sending the message across the chord and then one informed node sends the message to cycle *A* while the other informed node sends to cycle *B*. Algorithm 1 is a formal description of our technique. It is important to note that in our algorithm, a node only passes a message on if the node acting as the receiver has not yet been informed (i.e., any given node only receives the message once). See Figure 3 for an example of broadcasting using this Algorithm 1.

Function ESCO($\mathbf{G}, n$)

1: The originator sends across the chord.
2: The originator sends to its neighbor in cycle A while the node informed in Step 1 sends to its neighbor in cycle B.
3: The originator sends to its neighbor in cycle B while the node informed in Step 1 sends to its neighbor in cycle A. The nodes informed in Step 2 send to their uninformed neighbor.
4: **while** Not all nodes are informed. **do**
5:     Nodes informed in the previous step send to their uninformed neighbor.
6: **end while**

Algorithm 1: Equal Split Chord Originator Algorithm

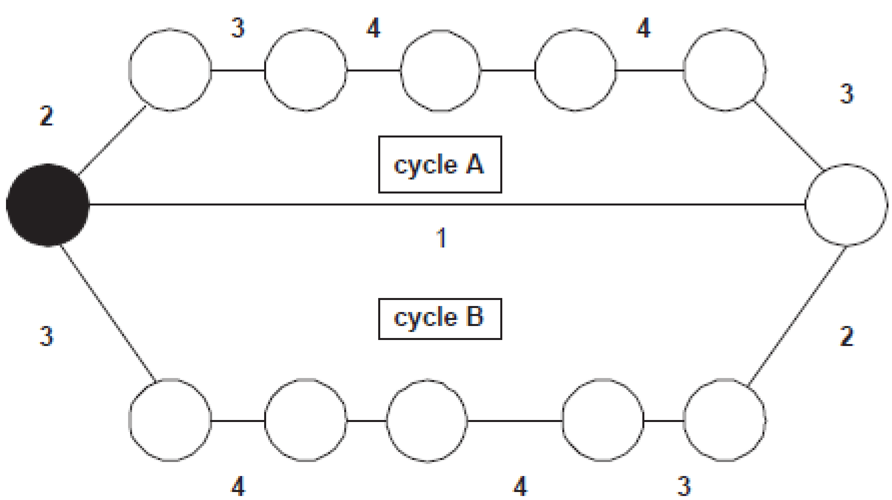

Figure 3: Broadcasting using Algorithm 1.

Lemma 1: *The time to broadcast using Algorithm 1 is equal to the diameter of the larger cycle formed by the chord (D_m) plus 1.*

Proof: We note that cycle *A* and cycle *B* differ in length by at most one node and according to the broadcast algorithm, after steps 1-3, both cycles contain the same number of informed nodes. Thus, the larger cycle (cycle *A*) must finish broadcasting at the same time as, or after, the smaller cycle (cycle *B*);

thus, we only need to consider cycle *A*, when analyzing the running time of the algorithm. Further analysis of cycle *A* requires us to consider the case when cycle *A* contains an even number of nodes and the case when cycle *A* contains an odd number of nodes.

Case 1: Cycle *A* contains an even number of nodes; we know that Farley's lower bound is realized by such cycles. This bound states that the minimum broadcast time is equal to $2 \cdot (M - 1) + D$, where M represents the number of messages being broadcast and D represents the diameter of the graph ([F80]). Observe that $D = \left\lfloor \frac{n}{2} \right\rfloor$, since we are dealing with a cycle, and since we are only interested in broadcasting one message, $M = 1$. Using this information, we calculate the minimum broadcast time for the cycle to be $\left\lfloor \frac{n}{2} \right\rfloor$. In Figure 4, the Chordless Cycle Algorithm is depicted on C_n where n is even, which obtains this bound. In Figure 5, Algorithm 1 is depicted on $C_{n+(n-2)}$ with a chord dividing the cycle in half (i.e., cycle *A* contains n nodes and cycle *B* contains n nodes). Comparing Figure 4 with cycle *A* of Figure 5, it can be seen that the last two nodes are informed at time $\left\lfloor \frac{n}{2} \right\rfloor$ in Figure 4; however, at that time, one node (x) remains to be informed in cycle *A* of Figure 5. The uninformed node can be informed by either one of its neighbors at the next time step. Thus, the time required to broadcast when cycle *A* contains an even number of nodes is $\left\lfloor \frac{n}{2} \right\rfloor$+1.

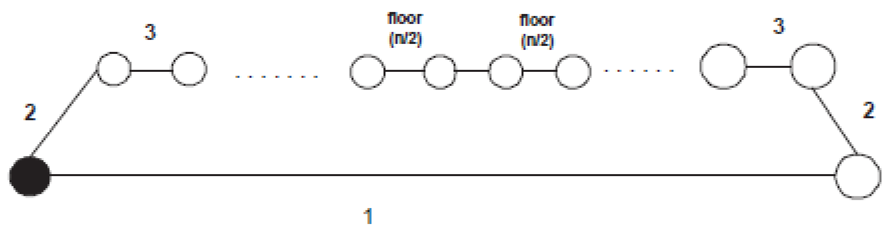

Figure 4: Case 1: Broadcasting in C_n, where n is even, using the Chordless Cycle Algorithm.

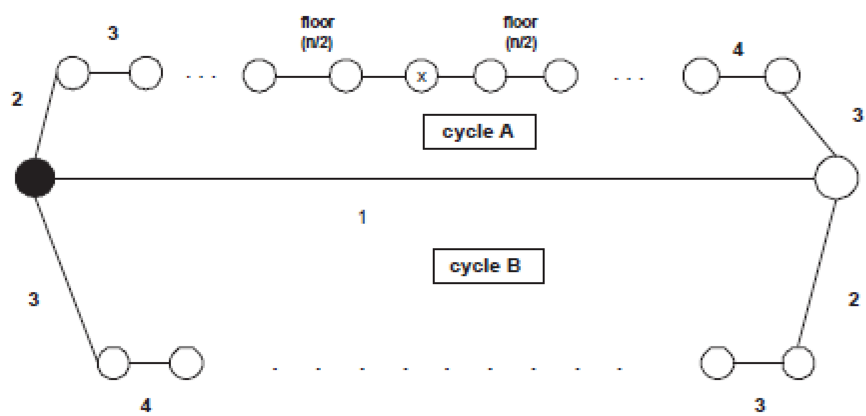

Figure 5: Case 1: Broadcasting in $C_{n+(n-2)}$, where n is even, using Algorithm 1.

Case 2: The larger cycle formed by the chord (cycle *A*) contains an odd number of nodes. We know that Farley's lower bound cannot be achieved by such cycles; since there are two nodes which are both the maximum distance from the originator. The time required to broadcast is $2 \cdot (M - 1) + D + 1$, where D =

$\left\lfloor \frac{n}{2} \right\rfloor$ and M = 1; thus, the time required to broadcast becomes $\left\lfloor \frac{n}{2} \right\rfloor + 1$. In Figure 6, the Chordless Cycle Algorithm is depicted on C_n where n is odd, which obtains this bound. In Figure 7, Algorithm 1 is depicted on $C_{n+(n-2)}$ with a chord dividing the cycle in half (i.e., cycle A contains n nodes and cycle B contains n nodes). From Figure 6, we note that, at the next to last time step, two nodes are informed; however, only one of these nodes is needed to pass the message on to the last node to be informed. We can increase the time at which the node adjacent to node x is informed by 1 without affecting the overall time needed to broadcast. This is precisely what happens in cycle A of Figure 7. Thus, the time required to broadcast when cycle A contains an odd number of nodes is $\left\lfloor \frac{n}{2} \right\rfloor+1$.

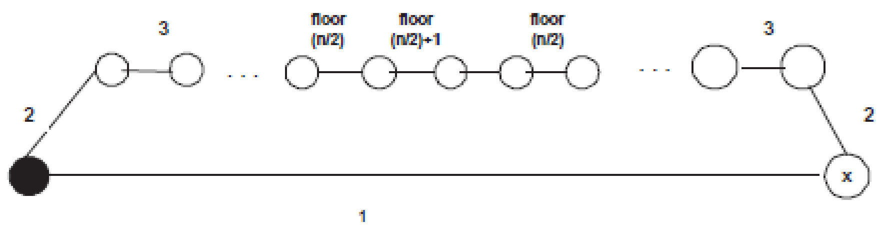

Figure 6: Case 2: Broadcasting in C_n, where n is odd, using the Chordless Cycle Algorithm.

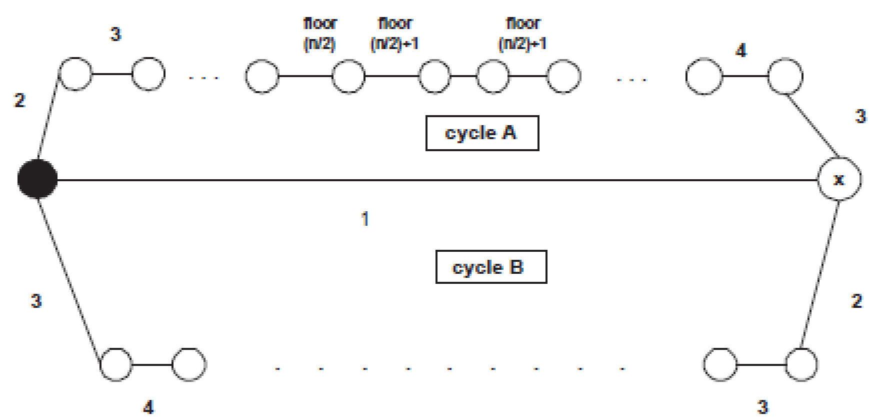

Figure 7: Case 2: Broadcasting in $C_{n+(n-2)}$, where n is odd, using Algorithm 1.

Results

Our algorithm used the originator as one endpoint of the chord and the node, p, which was located as far away from the originator as possible as the other endpoint of the chord. Using these nodes, the two smaller cycles which are created by the chord will be of equal size, if n is even or they will differ in size by a single node, if n is odd. Choosing any node closer to the originator than node p will create two smaller cycles that differ in size by at least 2 nodes. As shown in the previous section, due to the way broadcasting in a cycle progresses, increasing the size of either of the smaller cycles formed by the chord will increase the overall running time of the algorithm. Thus, the best running time that we can achieve when broadcasting in a cycle with a single chord occurs when the originator forms one endpoint of the chord and node *p* is chosen as the other endpoint of the chord. The chord placement technique and broadcasting algorithm presented in this paper assume that we know the location of the originator; however, if the location of the originator is unknown, then the best we can do is to arbitrarily choose a node as the originator and calculate its corresponding node *p*. In this case, since the originator is chosen arbitrarily, the actual originator could be located halfway between the chosen originator and node *p*. If this scenario occurs, the chord will produce no benefit to the broadcast time of the cycle (see Figure 8).

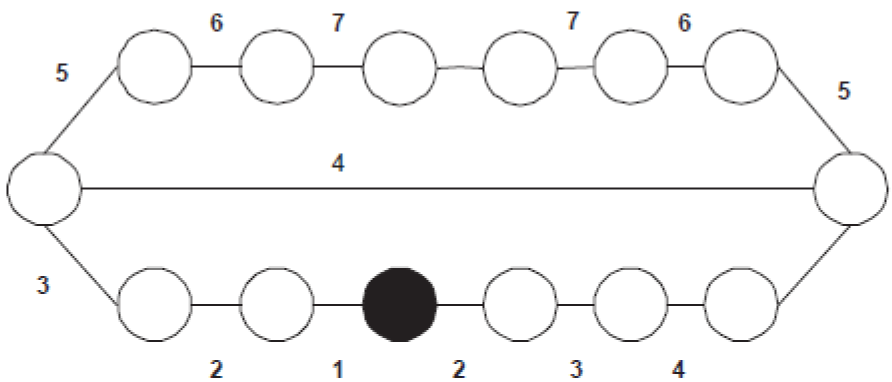

Figure 8: Worst case scenario for the location of the originator and node *p*.

Conclusion

This paper has considered the problem of broadcasting in a cycle containing a single chord. We have given a best case method for the placing the chord in the cycle, if the location of the originator is known, and have developed an algorithm for broadcasting in the resulting graph. Our algorithm achieves a running time of $\left\lceil \frac{n}{2} \right\rceil + 1$, when the location of the originator is known. Future research could consider broadcasting in cycles with a single chord using other communication models such as the postal broadcasting model, reliable broadcasting or multiple message broadcasting.

References

[FH78] A. Farley and S. Hedetniemi. Broadcasting in grid graphs. *In Proc. Ninth SE Conf. on Combinatorics, Graph Theory and Computing,* pages 275-288, Winnipeg, 1978. Utilitas Mathematica.

[F80] A. Farley. Broadcast time in communication networks. *SIAM Journal of Applied Mathematics,* 39:385-390, 1980.

[HHL88] S. M. Hedetniemi, S. T. Hedetniemi and A. Liestman. A survey of broadcasting and gossiping in communication networks. *Networks,* 18:319-349, 1988.

[K79] C. Ko. On a conjecture concerning broadcasting in grid graphs, preliminary report. *Notices Amer. Math. Soc.* 26:A-196, 1979.

[P80] G. W. Peck. Optimal spreading in an *n*-dimensional rectilinear grid. *Stud. Appl. Math.* 62:69-74, 1980.

[P81] Andrzej Proskurowski. Minimum broadcast trees. *IEEE Trans. On Comput.* 30:363-366, 1981.

[SCH81] P.J. Slater, E. Cockayne and S.T. Hedetniemi. Information dissemination in trees. *SIAM J. Comput.,* 10:692-701, 1981.

[S99] Matthew Suderman. Multiple message broadcasting. MS Thesis, *Simon Fraser University*, 1999.

[W99] I. Wojciechowska. Broadcasting in grid graphs. Dissertation, *West Virginia University*, 1999.