Broadcasting in chains of rings

1st Hovhannes A. Harutyunyan
Department of Computer Science and Software Engineering
Concordia University
Montreal, QC, H3G 1M8, Canada
haruty@cs.concordia.ca

2nd Narek Hovhannisyan
Department of Computer Science and Software Engineering
Concordia University
Montreal, QC, H3G 1M8, Canada
narek.hovhannisyan@mail.concordia.ca

3rd Edward Maraachlian
Department of Computer Science and Software Engineering
Concordia University
Montreal, QC, H3G 1M8, Canada
emarashlian@yahoo.com

***Abstract*—*Broadcasting* is an information dissemination problem in an interconnection network where one node, called the *originator*, must distribute a message to all other nodes by placing a series of calls along the links of the network. Every time the informed nodes aid the originator in distributing the message. Finding the broadcast time of any node in an arbitrary network is NP-complete. Polynomial time solutions are known only for a few classes of networks. In this paper, we consider chains of rings. We present a linear algorithm to find the broadcast time of arbitrary chains of rings and closed chains of rings.**

***Index Terms*—interconnection networks, message dissemination, broadcasting**

I. Introduction

In order to find the best communication structure for parallel and distributed computing, a lot of work has been done in the study of the properties of interconnection networks. The ability to effectively disseminate information among the processors is an important feature of an interconnection network.

Broadcasting is one of the most important information dissemination processes in an interconnected network. Over the last four decades, a large number of research work has been published concerning broadcasting in networks under different models. These models can have different numbers of originators, numbers of receivers at each time unit, distances of each call, numbers of destinations, and other characteristics of the network such as the knowledge of the neighborhood available to each node. In the context of this paper, we are going to focus on the classical model of broadcasting. The network is modeled as an undirected connected graph $G = (V, E)$, where $V(G)$ and $E(G)$ denote the vertex set and the edge set of G, respectively. The classical model follows the below-mentioned basic assumptions.

1) The broadcasting process is split into discrete time units.
2) The only vertex that has the message at the first time unit is called *originator*.
3) In each time unit, an informed vertex (*sender*) can *call* at most one of its uninformed neighbors (*receiver*).
4) During one time unit, all calls are performed in parallel.
5) The process halts as soon as all the vertices in the graph are informed.

We can represent each call in this process as an ordered pair of two vertices (u, v), where u is the sender and v is the receiver. The *broadcast scheme* is the order of calls made by each vertex during a broadcasting process and can be represented as a sequence $(C_1, C_2, ..., C_t)$, where C_i is the set of calls performed in time unit i. An informed vertex v is *idle* in time unit t if v does not make any call in time t. Given that every vertex, other than the originator, can be informed by exactly one vertex, the broadcast scheme forms a directed spanning tree (*broadcast tree*) rooted at the originator. We are also free to omit the direction of each call in the broadcast tree.

Definition 1. *The* ***broadcast time*** *for a vertex v in a given graph G, denoted by $b(v, G)$, is the minimum number of time units required for broadcasting in G if v is the originator.*
The broadcast time for a given graph G, is the maximum broadcast time from any originator in G, formally $b(G) = max_{v \in V(G)}\{b(v, G)\}$.

A broadcast scheme for an originator v that uses $b(v, G)$ time units is called optimal broadcast scheme and is denoted by $\mathcal{S}(v, G)$. Obviously, from the constraints, the number of informed vertices after each time unit can be at most doubled. Meaning, in general, the number of informed vertices after time unit i is upper bounded by 2^i. Therefore, it is easy to see that $b(v, G) \geq \lceil \log n \rceil$, where n is the number of vertices in G, which implies that $b(G) \geq \lceil \log n \rceil$. Another obvious lower bound on the broadcast time $b(G)$ is the diameter of graph G $(b(G) \geq d(G))$.

Generally, the broadcast time decision problem in an arbitrary graph is NP-complete [5, 16]. Moreover, the broadcast time problem was proved to be NP-complete even for some specific graph classes, such as 3-regular planar graphs [14]. There is a very limited number of graph families, for which an exact algorithm with polynomial time complexity is known for the broadcast time problem. Exact linear time algorithms are available for the broadcast time problem in trees [15, 16], in connected graphs with only one cycle (unicyclic graphs) [7, 8], in k-restricted cactus graphs [3], in fully connected trees [6], and in Harary-like graphs [1, 2]. For a more detailed introduction to broadcasting, we refer the reader to [4, 10, 11, 12, 13].

A *Cactus graph* is a connected graph in which any two cycles have at most one vertex in common. Or, equivalently, each edge in a cactus graph meets at most one cycle. A *k-restricted cactus graph* is a cactus graph where no more than k cycles can have more than one vertex in common, or equivalently, a cactus graph in which every vertex is on at most k cycles. A *chain of rings* (also referred to as a necklace graph) is a collection of cycles, where each consecutive pair of cycles is connected by one vertex. Another interesting topology of networks is the *closed chain of rings*.

Chains of rings as a subfamily of cactus graphs are an interesting area of research. Moreover, solving the minimum broadcast problem in graph families with intersecting cycles can help to understand the problem in gerenal graphs.

The paper is organized as follows. After the introduction in Sect. I, in Sect. II, we introduce some previous results on this problem and describe a simple linear-time algorithm for finding an optimal broadcast scheme in a chain of rings. In Sect. III we present the complete list of possible broadcast times of a chain of rings. Next, Sect. IV presents a polynomial-time exact algorithm for closed chains of rings. Finally, Sect. V concludes the paper.

II. A BROADCASTING ALGORITHM IN A CHAIN OF RINGS

In [9], the authors introduce algorithms for finding an optimal broadcast scheme for graphs that are made up of a cycle and an attached tree such that the root of the tree has a degree of 2 in the tree. First, they consider a graph G made up of a cycle C_N and a graph G', such that both graphs have a vertex v_c in common. The vertex v_c is called the connecting vertex (Fig. 5.c). The following theorem is proved in [9].

Theorem 1. *[9] For any originator v_0 on C_N, there exists an optimal broadcast scheme that first sends the information along the shorter path towards vertex v_c and then along the longer path.*

Then, assume that a graph G is made up of a cycle C and a graph G' such that G' is attached to C at v_c and the degree of v_c in G' is equal to 2. Assume that T is a broadcast tree of G' for the originator v_c. Let r_1 and r_2 be the two children of v_c and T_1 and T_2 be the two subtrees of T rooted at r_1 and r_2 correspondingly. Several results introduced in [9] are summarized in the following lemma.

Lemma 1. *For any originator v_0 on C other than v_c the broadcast time $b(v_0, G) = b(v_0, G_T)$ where the graph G_T is made up of the cycle C and the broadcast tree T. If there are several broadcast trees then the tree that has $b(v_1, T_1) > b(v_2, T_2) + 2$ must be chosen.*

Note that all the broadcast trees of a chain of rings have roots of degree 2 since they are on a cycle where all vertices have degree 2. Using the previous results, the authors construct a spanning tree that yields the optimal broadcast time for the chain of rings, N_k, when the originator is on the end cycle and is not a connecting vertex. The broadcast tree is constructed using a bottom-up approach. In summary, the authors in [9], introduced an exact broadcasting algorithm for the chain of rings originating from an end cycle and posted an open problem of finding the minimum broadcast time in the chain of rings when the originator is on an internal cycle or is a connecting vertex. Further in this section, we will introduce simple broadcast tree construction algorithms for the cases when the originator is on an internal cycle or is a connecting vertex.

Later, in [3], the authors introduce an exact algorithm for the broadcast time problem on cactus graphs with superpolynomial time complexity. However, the algorithm runs in linear time for the broadcast time problem in k-restricted cactus graphs. Since chains of rings (necklace graphs) are a subfamily of 2-restricted cactus graphs, the algorithm in [3] can be applied to a chain of rings.

Further, we present simple algorithms for finding the broadcast scheme from any originator in a chain of rings.

A. Broadcasting from a non-connecting vertex on an internal cycle

In this section, we will study the case where the originator in the chain of rings is not on an end cycle but is a non-connecting vertex on an internal cycle (vertex u in Fig. 5.c). The approach we use here is dividing the problem into two instances where in each instance the originator is on an end cycle. After the problem is solved in these two instances the solutions will be combined to obtain the broadcast time and broadcast scheme for the original problem.

Assume that we are given a chain of rings with k cycles and an originator v_0 on cycle C_i, $1 < i < k$. We will split the necklace chain into 2 and solve the broadcast problem in each. In each problem, the originator is on an end cycle. The first chain of rings is made up of the cycles C_1 to C_{i-1} and the second chain of rings is made up of the cycles C_{i+1} to C_k. In the first graph, the originator will be considered to be the connecting vertex that connects the cycles C_{i-1} and C_i. For the second graph, the originator is considered to be the vertex connecting the cycles C_i and C_{i+1}. After solving the broadcast problem in the two chains of rings we get 2 broadcast trees which when attached back to the cycle C_i result in a unicyclic graph. The final step is to solve the broadcast problem in the unicyclic graph where the originator is v_0 [8].

Earlier we discussed that it is possible to split a chain of rings into two, solve the problem in one part and use its broadcast tree to solve the broadcast problem for the whole necklace graph [9]. The case that we face now is quite similar but with two chains of rings connected at two different vertices of a cycle. Let us denote the cycle by C and the first chain of rings by N_1 which is connected to C at vertex v_1. The second chain of rings, N_2, is connected to the cycle at vertex v_2. Let G denote the graph made up of two chains of rings and the cycle (i.e. the complete chain of rings). We proved that there exists a broadcast tree in G which has two subtrees which are

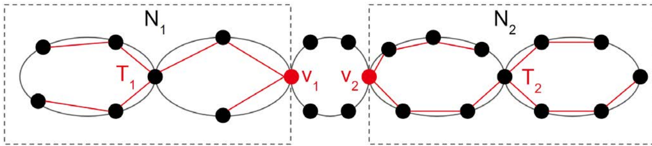


Fig. 1. Example of a cycle and two attached necklace graphs, where the subtrees T_1 and T_2 induced by an optimal broadcast scheme of the overall graph are also optimal broadcast trees for the chains of rings N_1 and N_2, respectively.

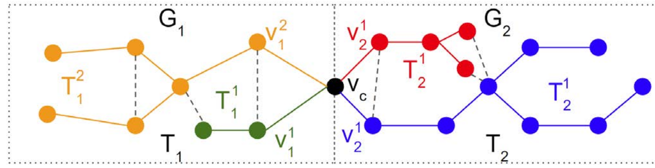


Fig. 2. Example of two graphs G_1 and G_2 with a common vertex v_c, where the subtrees T_1^1, T_1^2, T_2^1, and T_2^2 induced by an optimal broadcast scheme of the overall graph also result in optimal broadcast trees for each graph separately.

the broadcast trees of N_1 and N_2 (Fig. 1). The proof of the following was done by a contradiction and is omitted due to space limitation.

Theorem 2. *There exists an optimal broadcast tree in G such that the subtrees induced by the vertices of N_1 and N_2 are optimal broadcast trees for N_1 and N_2 respectively.*

From Theorem 2 we conclude that it is possible to build a broadcast tree of the graph G by first solving the broadcast problem in the two chains of rings and then using the broadcast trees of those two graphs. However, it is possible that chains of rings N_1 and N_2 can have more than one broadcast tree. If there are multiple trees, similar to the justification of the case where we had one tree attached to a cycle, we will choose a tree in which the difference in the broadcast times of the subtrees attached to the root is at least 2 (if such exists). Algorithm 1 presents the pseudocode for the above-described algorithm.

Algorithm 1: INTERNALCYCLEBROADCASTING

Input : A necklace graph N_k and a non-connecting vertex $v_0 \in C_i$, $1 < i < k$
Output: A broadcast scheme $S(v_0, N_k)$

```
begin
    N_1 = the chain formed by cycles C_1 to C_{i-1}
    N_2 = the chain formed by cycles C_{i+1} to C_k
    v_1 = the connecting vertex of cycles C_{i-1} and C_i
    v_2 = the connecting vertex of cycles C_i and C_{i+1}
    T_1 = S(v_1, N_1), with maximum imbalance [9]
    T_2 = S(v_2, N_2), with maximum imbalance [9]
    G = C_i ∪ T_1 ∪ T_2
    T = an optimal broadcast scheme S(v_0, G) [7]
    Return T
end
```

B. Broadcasting from a connecting vertex

A connecting vertex is a vertex that is common to two cycles (vertex v_c in Fig. 5.c). The correctness of the Algorithm 1 depends on Theorem 2 which assumes that when the connecting vertex is informed one of its neighbors is informed too. In this part, we study the case where the originator is a connecting vertex. The connecting vertex v_c is connecting two cycles and hence has a degree 2 in each cycle.

Consider a graph $G = (V, E)$ which can be divided into two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ such that $V_1 \cap V_2 = v_c$, $V_1 \cup V_2 = V$, $E_1 \cap E_2 = \emptyset$, and $E_1 \cup E_2 = E$. In other words, v_c is a cut vertex. Moreover, the graphs G_1 and G_2 are such that the degree of v_c is 2 in both graphs. Let T_1 rooted at v_c be a broadcast tree of G_1 and T_2 be a broadcast tree of G_2. Since v_c has degree 2 in each graph then T_1 has two subtrees connected to v_c, denoted by T_1^1 and T_1^2 rooted at v_1^1 and v_1^2, respectively. Similarly, T_2 has two subtrees T_2^1 and T_2^2 rooted at v_2^1 and v_2^2, respectively (Fig. 2). The following theorem can be easily proved by contradiction. The details of the proof are omitted due to space limitations.

Theorem 3. *Given a graph G with cut vertex v_c as defined above, if $b(v_c, G_1) \neq b(v_c, G_2)$ then an optimal broadcast tree T of the originator v_c in G can be constructed by adding edges from v_c to the vertices v_1^1, v_1^2, v_2^1, and v_2^2, the roots of the subtrees T_1^1, T_1^2, T_2^1, and T_2^2 respectively.*

Also, we proved that if $b(v_c, G_1) = b(v_c, G_2)$ then it is possible to achieve an optimal broadcast tree for G by using non-optimal broadcast trees from G_1 or G_2.

We can use this result to study the broadcast problem in chains of rings where the originator is a connecting vertex. A simple algorithm will be to first study the problem in the two parts of the chain of rings N_1 and N_2. If $b(v_0, N_1) \neq b(v_0, N_2)$ then the broadcast trees of N_1 and N_2 are used to construct a tree T and calculate the broadcast time of T using the well-known broadcast algorithm for trees [16]. However, if $b(v_0, N_1) = b(v_0, N_2)$ then the bottom-up algorithm should be modified such that at every iteration two spanning trees should be calculated. The first tree is the broadcast tree which has the maximum imbalance between its two subtrees and the second one is a spanning tree whose broadcast time is one more than the broadcast time of the optimal tree and again has the maximum imbalance between its two subtrees. Recall that a non-optimal broadcast tree of one of the chains of rings combined with an optimal broadcast tree of the other chain of rings can result in the optimal solution of the original chain of rings.

Algorithm 2 presents the pseudocode for the above-described algorithm. Thus, we completed algorithms for all possible originators of any chain of rings.

Finally, we show that the complexity of the algorithm is linear. First of all, finding the end cycles takes a linear time. Next, we build the broadcast tree in a bottom-up approach and calculate the broadcast time of the root. For the construction of a broadcast tree of a graph made up of a cycle with an attached tree to its vertices all the information about the tree has already been calculated. The only remaining task can be solved by traversing the cycle once. This will take a linear time. Hence, both Algorithms 1 and 2 can be implemented with linear-time complexity.

Algorithm 2: INTERNALCYCLEBROADCASTING

Input : A chain of rings N_k and a connecting vertex $v_0 \in C_i \cap C_{i+1}$
Output: A broadcast scheme $S(v_0, N_k)$

begin
- $N_1 =$ the chain formed by cycles C_1 to C_i
- $N_2 =$ the chain formed by cycles C_{i+1} to C_k
- $T_1 = S(v_0, N_1)$, with maximum imbalance [9]
- $T_1' =$ a spanning tree of N_1 rooted at v_0 with maximum subtree imbalance, such that $b(v_0, T_1') = b(v_0, T_1) + 1$
- $T_2 = S(v_0, N_2)$, with maximum imbalance [9]
- $T_2' =$ a spanning tree of N_2 rooted at v_0 with maximum subtree imbalance, such that $b(v_0, T_2') = b(v_0, T_2) + 1$
- **if** $b(v_0, T_1) \neq b(v_0, T_2)$ **then**
  - $T = S(v_0, T_1 \cup T_2)$ [7]
- **else**
  - $G = T_1 \cup T_2$
  - $G_1 = T_1' \cup T_2$
  - $G_2 = T_1 \cup T_2'$
  - $T = \min\{S(v_0, G), S(v_0, G_1), S(v_0, G_2)\}$
- **end**
- Return T

end

III. UPPER AND LOWER BOUNDS

Consider a broadcast algorithm that tries to inform the vertices of the next cycle whenever possible. Such a scheme is a greedy scheme. We will show that the greedy scheme cannot always generate the optimal solution. However, we will use this scheme to prove that the upper bound on the broadcast time of a chain of rings is $d + 2$, where d is the diameter of the graph.

Assume that we are given a chain of rings with k cycles and an originator v_0 which is on the end cycle C_0. The greedy algorithm, $A_{GreedyNecklace}$, is as follows:

1) v_0 first sends the message to the neighboring vertex that is closest to C_1. In the second time unit, v_0 informs its other neighbor.
2) If an informed vertex is not a connecting vertex, then it informs its neighbor at the next time unit after it gets informed.
3) If an informed vertex, v, belongs to 2 cycles C_j and C_{j+1}, it has 3 uninformed neighbors to inform. Assuming that the vertex informing v was in C_j, v first informs its neighbor on C_{j+1} that is closest to C_{j+2}, then it informs the other vertex on C_{j+1}, and finally, informs its uninformed neighbor on C_j.

Let v_0 be the originator on an end cycle and v_{c_i} be the vertex that connects cycles C_{i-1} and C_i.

Theorem 4. *In the greedy algorithm $A_{GreedyNecklace}$, the time it takes to inform vertex v_{C_i} is equal to the distance $d(v_0, v_{c_i})$.*

Proof. We can prove this theorem by induction. Consider the first connecting vertex of the cycles C_0 and C_1. According to the algorithm, the originator first informs its neighbor that is closest to v_{c_1}. Moreover, every newly informed vertex on the path between v_0 and v_{c_1} informs its uninformed neighbor in the next time unit. Therefore, v_{c_1} will be informed in $d(v_0, v_{c_1})$ time units. This proved the base case of the induction.

Assuming that the theorem is true for v_{c_i} we need to prove that the result will be correct for $v_{c_{i+1}}$. By the assumption, we know that v_{c_i} was informed in $d(v_0, v_{c_i})$ time units. Note that v_{c_i} belongs to the two cycles C_{i-1} and C_i. According to $A_{GreedyNecklace}$ algorithm when vertex v_{c_i} is informed, it first informs its two neighbors that are on the cycle C_i and then informs its uninformed neighbor on C_{i-1}. Therefore vertex $v_{c_{i+1}}$ will be informed $d(v_{c_i}, v_{c_{i+1}})$ time units after v_{c_i} gets informed. Since $d(v_0, v_{c_{i+1}}) = d(v_0, v_{c_i}) + d(v_{c_i}, v_{c_{i+1}})$, we conclude that $v_{c_{i+1}}$ gets informed at time $d(v_0, v_{c_{i+1}})$. This proves the inductive step and concludes the proof. □

Theorem 5. *The greedy algorithm $A_{GreedyNecklace}$ finishes broadcasting in at most $d + 2$ time units.*

Proof. First, note that the broadcast time of any chain of rings is equal to the broadcast time of a vertex $v_0 \in C_1$ (or $v_0 \in C_k$), $b(G) = b(v, G)$, such that $d(v_1, v_{c_1})$ (or $d(v_1, v_{c_k})$) is maximum. In order to prove this theorem we should note that every cycle C_j can have at most 2 vertices that are at distance d from the originator. There can be any number of cycles that contain at most 2 vertices that are at distance d from the originator. Consider any cycle C_j that contains two vertices v_1 and v_2 that are at distance d from the originator. Wlog assume that v_2 is closer to the connecting vertex, $v_{c_{j+1}}$ on cycles C_j and C_{j+1}. Let v_3 be the other neighbor of v_2 aside from v_1. According to the greedy scheme of $A_{GreedyNecklace}$, vertex v_1 will be informed in time unit $d + 1$ by a direct path from v_{c+j}. However, due to the delay of 2 time units at vertex $v_{c_{j+1}}$, vertex v_3 will be also informed in time unit $d + 1$. Finally, v_2 can be informed in time unit $d + 2$ either from v_1 or v_3. Regardless of how many cycles have vertices at distance d from the originator, all the vertices of those cycles will be informed by the time $d + 2$ with the greedy scheme. This concludes the proof. □

We showed that $d + 2$ is an upper bound on the broadcast time in chains of rings. Recall that for any graph G, $b(G) \geq d$. Thus, for a chain of rings G, $d \leq b(G) \leq d + 2$. In Fig. 5, we can see examples of three chains of rings with cycles of arbitrary length that can easily be shown to have broadcast times of d, $d+1$, and $d+2$. Hence, both upper and lower bounds are tight.

IV. Broadcasting in a closed chain of rings

A closed chain of rings is a chain of rings where the two end cycles of the chain of rings have a common vertex other than the two connecting vertices of the two endpoints (Fig. 3). In this section, we will use the results achieved in previous sections to introduce an exact algorithm for the broadcast time problem in a closed chain of rings.

A. Broadcasting from an internal vertex

Let G be a closed chain of rings, v be a non-connecting vertex in G, and T be a broadcast tree that results in a minimum broadcast time in the graph G from the originator v. Graph G consists of n cycles $C_1, C_2, ..., C_n$. Wlog assume that $v \in C_1$. Let T_1 and T_2 be two subtrees of the tree T rooted at two neighbors of v (v_1 and v_2, respectively).

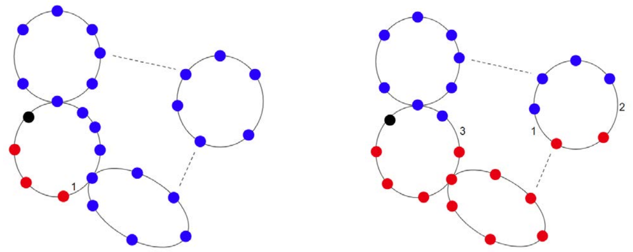
It is easy to see that there exist either one or three vertices in T_1, such that they have a neighbor in T_2. The case when there is only one such vertex is only possible if all vertices of T_1 are in C_1. We will consider each case separately.

1) There exists one vertex $u \in T_1$ that has a neighbor w in T_2. In this case, splitting the cycle C_1 by cutting the edge (u, w) will result in a graph G' which will have the same broadcast time as graph G. In G', two disjoint subgraphs G'_1 and G'_2 correspond to the trees T_1 and T_2 and are connected to the originator v. Graph G'_1 is a path graph, whereas, graph G'_2 is a chain of rings with a tree attached to the end cycle. Clearly, G' is a 2-restricted cactus, and we can find the broadcast time in linear time using the algorithm in [3].
2) There exist three vertices $u_1, u_2, u_3 \in T_1$ that have neighbors w_1, w_2, w_3 in T_2. This case is only possible when two of the vertices are in the same cycle C_i, and the other vertex is in the cycle C_1. Let u_1 and u_2 be in the same cycle C_i and u_3 be in C_1. In this case, splitting the cycle C_i by cutting the edges (u_1, w_1) and (u_2, w_2), and splitting the cycle C_1 by cutting the edge (u_3, w_3) will result in a graph G' which will have the same broadcast time as graph G. Similarly, G' is a 2-restricted cactus.

Hence, given the vertex or the pair of vertices that define an optimal broadcast tree, we can construct the broadcast scheme in a bottom-up approach similar to broadcasting in the open chain of rings. The set of possible split vertices is limited and can be considered exhaustively.

1) Recall that in the first case, the vertex u was in the cycle C_1. Hence, only $|C_1|$ vertices can be considered.
2) Whereas, in the second case, both vertices u_1 and u_2 belong to the same cycle C_i. Thus, there exist $\sum_{1<i\leq n} \binom{|C_i|}{2}$ possible pairs of vertices.

Fig. 3 shows examples of each possible case of two subtrees T_1 and T_2. Note that the examples do not represent an actual broadcast scheme and are aimed to visualize the possible cuts.



(a) (b)

Fig. 3. Examples of subtrees of the broadcast three rooted at an internal originator (black vertex). The vertices of each subtree are marked with different colors. Numbered edges are the edges that need to be cut.

B. Broadcasting from a connecting vertex

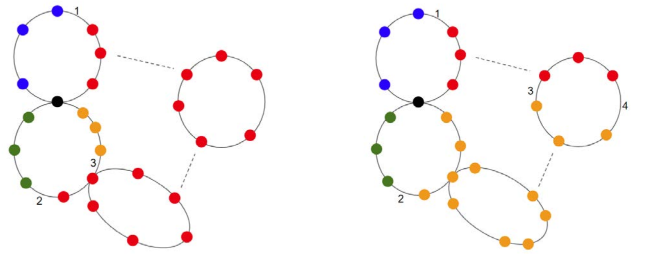
Let G be a closed chain of rings, v be a connecting vertex in G, and T be a broadcast tree that results in a minimum broadcast time in the graph G from the originator v. Graph G consists of n cycles $C_1, C_2, ..., C_n$. Wlog assume that v is the connecting vertex between cycles C_1 and C_2. Let T_1, T_2, T_3 and T_4 be four subtrees of the tree T rooted at four neighbors v_1, v_2, v_3, v_4 of v, such that $v_1, v_2 \in C_1$ and $v_3, v_4 \in C_2$.

Similar to the mechanism used in section IV-A, we are going to find the possible number of vertices that define the broadcast tree.

Obviously, other than v, the other connecting vertex in the cycle C_1 (as well as the one in C_2) can belong to only one of the subtrees, making the other subtree in that cycle a simple path. Wlog let T_1 and T_3 be simple paths from v to u_1 and u_3, respectively. Both u_1 and u_3 have adjacent vertices (w_1 and w_3) that can be either in T_2 or T_4. Whereas, the split between T_2 and T_4 can be defined analogously to the way described in section IV-A: there can exist either one or two vertices that have a neighbor in the other subtree. We will consider each case separately.

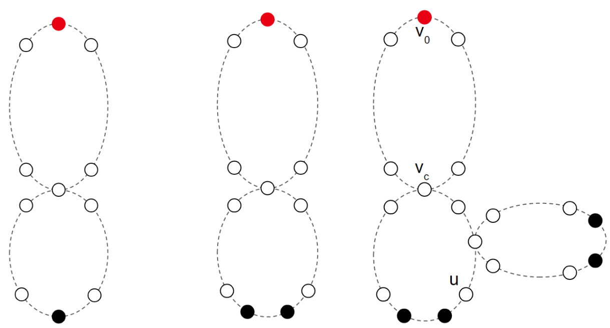
1) There exists one vertex $u \in T_2$ that has a neighbor w in T_4. This case is only possible if $u \in C_1$ or if $w \in C_2$. In this case, cutting the edges $(u, w), (u_1, w_1), (u_3, w_3)$ will result in a graph G' which will have the same broadcast time as graph G. In G', four disjoint subgraphs G'_1, G'_2, G'_3 and G'_4 correspond to the trees T_1, T_2, T_3 and T_4 and are connected to the originator v. Again, graph G' is a 2-restricted cactus graph.
2) There exist two vertices $u_2, u_4 \in T_2$ that have neighbors w_2, w_4 in T_4. Moreover, this case is only possible when both u_2 and u_4 are in the same cycle C_i. In this case, cutting the edges $(u_1, w_1), (u_2, w_2), (u_3, w_3)$ and (u_4, w_4) will result in a 2-restricted cactus graph G' which will have the same broadcast time as graph G.

Hence, given 3 or 4 vertices that define an optimal broadcast tree, we can construct the broadcast scheme in a bottom-up approach similar to broadcasting in the open chain of rings. The set of possible split vertices is limited and can be considered exhaustively. Vertices u_1 and u_3 have $|C_1|$ and $|C_3|$ possible cases respectively.



(a) (b)

Fig. 4. Examples of subtrees of the broadcast three rooted at an internal originator (black vertex). The vertices of each subtree are marked with different colors. Numbered edges are the edges that need to be cut.



(a) (b) (c)

Fig. 5. Examples of infinite chain of rings constructions with broadcast times d (a), $d+1$ (b), and $d+2$ (c).

1) Recall that in the first case, the vertex u was in the cycle C_1 or C_2. Hence, only $|C_1|+|C_2|$ vertices can be considered.
2) Whereas, in the second case, both vertices u_2 and u_4 belong to the same cycle C_i. Thus, there exist $\sum_{1<i\leq n}\binom{|C_i|}{2}$ possible pairs of vertices.

Hence, it is possible to construct an optimal broadcast scheme for broadcasting in closed chains of rings in polynomial time. Fig. 4 shows examples similar to the previous section.

V. Conclusion

In this paper, we present a simple linear algorithm to find an optimal broadcast scheme from any originator in an arbitrary chain of rings. A vertex that has the largest broadcast time belongs to an end cycle and is the furthest vertex from the connecting vertex of the end cycle. Therefore, the algorithm actually can find the broadcast time of a chain of rings. Additionally, we showed the complete list of possible broadcast times of a chain of rings. We also introduced an optimal algorithm for closed chains of rings.

References

[1] P. Bhabak, H. A. Harutyunyan, and S. Tanna. Broadcasting in harary-like graphs. In *17th International Conference on Computational Science and Engineering (CSE '14)*, pages 1269–1276. IEEE, 2014.

[2] P. Bhabak, H. A. Harutyunyan, and P. Kropf. Efficient broadcasting algorithm in harary-like networks. In *46th International Conference on Parallel Processing Workshops (ICPPW '17)*, pages 162–170. IEEE, 2017.

[3] M. Čevnik and J. Žerovnik. Broadcasting on cactus graphs. *Journal of Combinatorial Optimization*, 33:292–316, 2017.

[4] P. Fraigniaud and E. Lazard. Methods and problems of communication in usual networks. *Discrete Applied Mathematics*, 53(1-3):79–133, 1994.

[5] M. R. Garey and D. S. Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979.

[6] S. Gholami, H. A. Harutyunyan, and E. Maraachlian. Optimal broadcasting in fully connected trees. *Journal of Interconnection Networks*, 23(01):2150037:1–2150037:20, 2023.

[7] H. A. Harutyunyan and E. Maraachlian. Linear algorithm for broadcasting in unicyclic graphs. In *International Computing and Combinatorics Conference (COCOON '07)*, pages 372–382. Springer, 2007.

[8] H. A. Harutyunyan and E. Maraachlian. On broadcasting in unicyclic graphs. *Journal of combinatorial optimization*, 16(3):307–322, 2008.

[9] H. A. Harutyunyan, G. Laza, and E. Maraachlian. Broadcasting in necklace graphs. In *Proceedings of the 2nd Canadian Conference on Computer Science and Software Engineering (C3S2E '09)*, pages 253–256, 2009.

[10] H. A. Harutyunyan, A. L. Liestman, J. G. Peters, and Richards D. Broadcasting and gossiping. In *Handbook of Graph Theory*, pages 1477–1494. Chapman and Hall, 2013.

[11] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman. A survey of gossiping and broadcasting in communication networks. *Networks*, 18(4):319–349, 1988.

[12] J. Hromkovič, R. Klasing, B. Monien, and R. Peine. Dissemination of information in interconnection networks (broadcasting & gossiping). In *Combinatorial network theory*, pages 125–212. Springer, 1996.

[13] J. Hromkovič, R. Klasing, A. Pelc, P. Ruzicka, and W. Unger. *Dissemination of information in communication networks: broadcasting, gossiping, leader election, and fault-tolerance*. Springer Science & Business Media, 2005.

[14] M. Middendorf. Minimum broadcast time is np-complete for 3-regular planar graphs and deadline 2. *Information Processing Letters*, 46(6):281–287, 1993.

[15] A. Proskurowski. Minimum broadcast trees. *IEEE transactions on computers*, 30(05):363–366, 1981.

[16] P. J. Slater, E. J. Cockayne, and S. T. Hedetniemi. Information dissemination in trees. *SIAM Journal on Computing*, 10(4):692–701, 1981.