APPROXIMATION ALGORITHMS FOR BROADCASTING IN SIMPLE GRAPHS WITH INTERSECTING CYCLES

Puspal Bhabak

A thesis
in
The Department
of
Computer Science

Presented in Partial Fulfillment of the Requirements
For the Degree of Doctor of Philosophy
Concordia University
Montréal, Québec, Canada

May 2015

Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Mr Puspal Bhabak**

Entitled: **APPROXIMATION ALGORITHMS FOR BROADCASTING IN SIMPLE GRAPHS WITH INTERSECTING CYCLES**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Computer Science)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________________ Chair
Dr. N. Bhuiyan

______________________________ External Examiner
Dr. P. Kropf

______________________________ External to Program
Dr. N. Hardy

______________________________ Examiner
Dr. L. Narayanan

______________________________ Examiner
Dr. J. Opatrny

______________________________ Thesis Supervisor
Dr. H. Harutyunyan

Approved ______________________________
Chair of Department or Graduate Program Director

__________ 20 _____ ______________________________

Amir Asif, Ph.D., Dean
Faculty of Engineering and Computer Science

Abstract

APPROXIMATION ALGORITHMS FOR BROADCASTING IN SIMPLE GRAPHS WITH INTERSECTING CYCLES

Puspal Bhabak, Ph.D.
Concordia University, 2015

Broadcasting is an information dissemination problem in a connected network in which one node, called the *originator*, must distribute a message to all other nodes by placing a series of calls along the communication lines of the network. Every time the informed nodes aid the originator in distributing the message. Finding the minimum broadcast time of any vertex in an arbitrary graph is NP-Complete. The problem remains NP-Complete even for planar graphs of degree 3 and for a graph whose vertex set can be partitioned into a clique and an independent set. The best theoretical upper bound gives logarithmic approximation. It has been shown that the broadcasting problem is NP-Hard to approximate within a factor of $3 - \epsilon$. The polynomial time solvability is shown only for tree-like graphs; trees, unicyclic graphs, tree of cycles, necklace graphs and some graphs where the underlying graph is a clique; such as fully connected trees and tree of cliques. In this thesis we study the broadcast problem in different classes of graphs where cycles intersect in at least one vertex. First we consider broadcasting in a simple graph where several cycles have common paths and two intersecting vertices, called a k-path graph. We present a constant approximation algorithm to find the broadcast time of an arbitrary k-path graph. We also study the broadcast problem in a simple cactus graph called k-cycle graph where several cycles of arbitrary lengths are connected by a central vertex on one end. We design a constant approximation algorithm to find the broadcast time of an arbitrary k-cycle graph.

Next we study the broadcast problem in a hypercube of trees for which we present a 2-approximation algorithm for any originator. We provide a linear algorithm to find the broadcast time in hypercube of trees with one tree. We extend the result for any arbitrary graph whose nodes contain trees and design a linear time constant

approximation algorithm where the broadcast scheme in the arbitrary graph is already known.

In Chapter 6 we study broadcasting in Harary graph for which we present an additive approximation which gives 2-approximation in the worst case to find the broadcast time in an arbitrary Harary graph. Next for even values of n, we introduce a new graph, called modified-Harary graph and present a 1-additive approximation algorithm to find the broadcast time. We also show that a modified-Harary graph is a broadcast graph when k is logarithmic of n.

Finally we consider a diameter broadcast problem where we obtain a lower bound on the broadcast time of the graph which has at least $\binom{d+k-1}{d} + 1$ vertices that are at a distance d from the originator, where $k \geq 1$.

Acknowledgments

I would like to take this opportunity to thank people who have been explicitly and implicitly involved in my life during my PhD.

First and foremost I would like to express my sincere gratitude to my supervisor Dr. Hovhannes Harutyunyan for being such a great guide. Dr. Harutyunyan is a wealth of knowledge. He has always helped, guided and motivated me with his intensive research capability since the first day that I arrived at Concordia. His constant direction has been significantly influential during my entire term of PhD. As an individual, he is a role model.

I would like to thank my family for being a pillar of support and always encouraged me to do higher education in the field of Computer Science. I run out of words while expressing gratitude to my parents Mr. Sunil Kr. Bhabak and Mrs. Kalyani Bhabak who are in Kolkata, India and have been praying every moment for my well-being and success. I am also greatly indebted to the blessings and support of my parents-in-law.

Last but not the least I acknowledge the loving support of my wife Debalina Bhowmick. She has always borne all my frustrations and has stayed by my side through all the ups and downs.

On a final note, I thank God for being such a positive influence throughout my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

When the computer was first designed, the main purpose was to perform the very tedious computations of everyday life and business in seconds. Since then many efforts have been made to transform the computing machines into intelligent ones that possess self-organizing skills like capable of massive parallel processing, support voice recognition and understand natural language. As a result the need of the hour has been to build an advanced technology as the single CPU systems take longer time to solve the problem serially.

Multi-computer and multi processor systems have been the solution to this problem. In the multi processor systems, different processors work in parallel. This is accomplished by breaking the problem into independent parts so that each processing element can execute its part of the algorithm simultaneously with the others. Sometimes the processors exchange data among themselves whenever it is needed either through shared memory (shared between all processing elements in a single address space), or distributed memory (in which each processing element has its own local address space).

Parallelism has several advantages. First of all it saves time and money as having more resources for a task will reduce the time to completion with potential cost savings. Besides it is more convenient to solve larger problems on multi-core due to increase in memory space. Now-a-days most of the computers being used by the common people have multi-core processors in their system. Along with the improvement on the physical level, one has to design an efficient algorithm that will distribute the information among the processors through the interconnection network so that we can get the most benefit out of the advances in the hardware domain. In recent years, a lot of work has been dedicated to studying properties of interconnection networks in order to find the best communication structures for parallel and distributed computing. The communication primitives can be defined as follows:

- **Routing** or one-to-one communication.
- **Broadcasting** or one-to-all communication.
- **Multicasting** or one-to-many communication.
- **Gossiping** or all-to-all communication.

One of the main problems of information dissemination investigated in this research area is broadcasting. The broadcast problem is one in which the knowledge of one processor must spread to all other processors in the network. For this problem we can view any interconnection network as a connected undirected graph $G = (V, E)$, where V is the set of vertices (or processors) and E is the set of edges (or communication lines) of the network. According to [113], the broadcast time problem was introduced in 1977 by Slater, Cockayne and Hedetniemi. Large sources of information about broadcasting and related problems are survey articles ([70], [113], [116]), book [117] and book chapter [97].

Formally, *broadcasting* is the message dissemination problem in a connected network in which one informed node, called the *originator*, must distribute a message to all other nodes by placing a series of calls along the communication lines of the network. Every time the informed nodes help the originator in distributing the message. This is assumed to take place in discrete time units. The broadcasting is to be completed as quickly as possible subject to the following constraints:

- Each call requires one unit of time.
- A vertex can participate in only one call per unit of time.
- Each call involves only two adjacent vertices, a sender and a receiver.

Given a connected graph G and a message originator, vertex u, the natural question is to find the minimum number of time units required to complete broadcasting in graph G from vertex u. This number is defined as the *broadcast time* of vertex

u, denoted $b(u, G)$ or $b(u)$. The broadcast time $b(G)$ of the graph G is defined as $\max\{b(u)|u \in V\}$. It is easy to see that for any vertex u in a connected graph G with n vertices, $b(u) \geq \lceil \log n \rceil$ (all log's in this thesis are base 2), since during each time unit the number of informed vertices can at most double. Also, in a connected graph there should be at least one new informed vertex at every new round which implies that $b(u) \leq n - 1$. G is called a *broadcast graph* if $b(G) = \lceil \log n \rceil$. For the complete graph K_n with $n \geq 2$ vertices, $b(K_n) = \lceil \log n \rceil$, yet K_n may not be minimal with respect to this property. That is, we may be able to remove some edges from K_n and still have a subgraph K'_n with n vertices such that $b(K'_n) = \lceil \log n \rceil$. In any connected graph G, a broadcast from a vertex u determines a spanning tree rooted at u. This spanning tree is called a broadcast tree. Figure 1 shows a broadcast scheme in 6 rounds, which is shown in the edge labelling. Vertex with the label 0 is the originator.

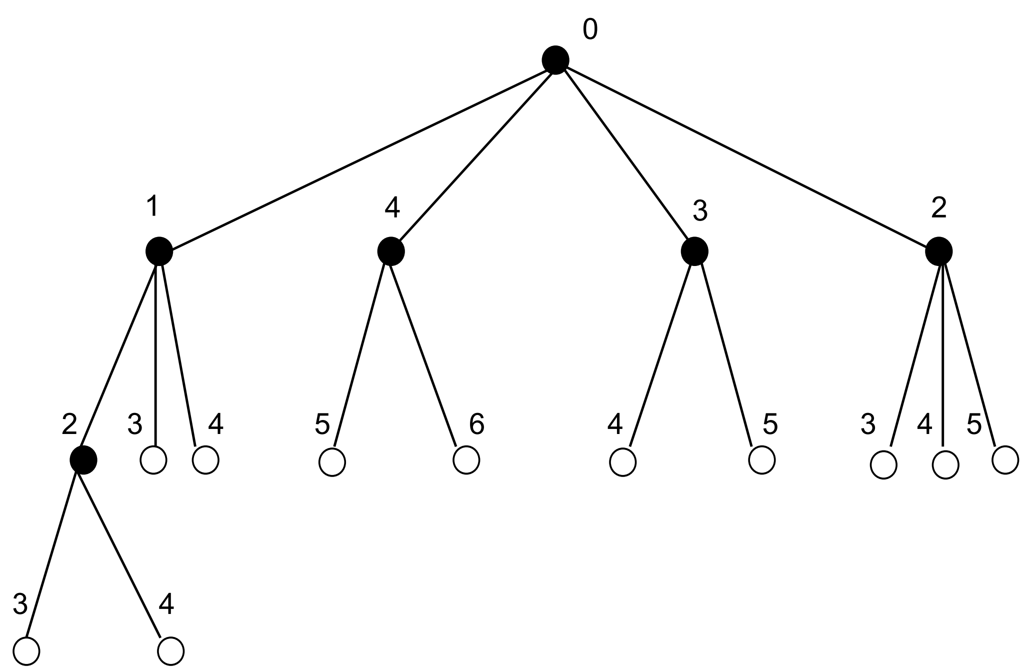

Figure 1: Broadcast Tree

Determining $b(u)$ for an arbitrary originator u in an arbitrary graph G has been proved to be NP-Complete in [162]. The problem remains NP-Complete even for 3-regular planar graphs [143] and for a graph whose vertex set can be partitioned into a clique and an independent set [121]. The best theoretical upper bound is obtained by the approximation algorithm in [49] which produces a broadcast scheme with $O(\frac{\log(|V|)}{\log\log(|V|)}b(G))$ rounds. Research in [159] has showed that it is NP-Hard to approximate the solution of the broadcast time problem within a factor $\frac{57}{56}-\epsilon$. However this result has been improved to within a factor of $3-\epsilon$ in [49]. As a result research has been made in the direction of finding approximation or heuristic algorithms to determine the broadcast time in arbitrary graphs (see [7], [12], [49], [50], [71], [72], [73], [110], [112], [131], [156], [158]).

Since the broadcast problem in general is NP-Hard, another direction is to design polynomial algorithms for some classes of graphs. The first result in this direction was a linear algorithm to determine the broadcast time of any tree [162]. The authors have introduced the term *broadcast centre*, which is the set of all vertices having minimum broadcast number, in order to determine the broadcast time for the tree in linear time. Recent research shows that there are polynomial time algorithms for the broadcast problem in tree-like graphs where two cycles do not intersect - unicyclic graphs, tree of cycles, or in graphs containing cliques, however with no intersecting cliques - fully connected trees and tree of cliques ([90], [100], [102], [103]). However, the problem remains NP-Hard for restricted classes of graphs.

A long standing open problem is to present a constant approximation for broadcasting in arbitrary graph or to prove that it is NP-Hard to approximate within a constant factor. One way of approaching this problem is to consider broadcasting in more complex graphs to the extent that we cannot provide a constant approximation for broadcasting in that graph. The thesis is a contribution to this longer research path. On the other hand polynomial time algorithms for the broadcast problems is known only for the class of graphs where two cycles do not intersect. Thus to bridge this gap, we consider broadcasting in simple graphs contain intersecting cycles. We first consider broadcasting in a simple graph where several cycles have two intersecting vertices, called a k-path graph. We next study the broadcast problem in a simple cactus graph called k-cycle graph where several cycles of arbitrary lengths are connected by a central vertex on one end. We next consider broadcasting in a graph where each vertex of the hypercube is the root of a tree, called hypercube of trees. In the literature there is a polynomial algorithm for the broadcast problem in fully-connected trees. However, the problem is much more difficult for hypercube of trees because in a hypercube any pair of vertices are not neighbors as in a clique. Finally we study the broadcast problem in Harary-like graphs which are regular k-connected graphs.

The rest of the thesis has been organized as follows. In the next chapter we present a literature review of some of the important works that have been done so far in the area of broadcasting in a network in general and the different network classes that have been considered. In Chapter 3 we will present a constant approximation algorithm to find the broadcast time of an arbitrary k-path graph. In Chapter 4 we study the broadcast problem in a simple cactus graph called k-cycle graph where we design a constant approximation algorithm to find the broadcast time of an arbitrary k-cycle graph. In Chapter 5 we study the broadcast problem in a hypercube of trees for which we present a 2-approximation algorithm for any originator. We provide a linear algorithm to find the broadcast time in hypercube of trees with one tree. We extend the result for any arbitrary graph whose nodes contain trees and design a linear time constant approximation algorithm. In the 6th Chapter we study broadcasting in Harary graph for which we present a $\log \frac{k-2}{2}$-additive approximation to find the broadcast time in an arbitrary Harary graph. For even values of n, we introduce a modified-Harary graph and present a 1-additive approximation algorithm to find the broadcast time. We show the optimality of our algorithm for a particular subclass of modified-Harary graph. Then we also show that modified-Harary graph is a broadcast graph when k is logarithmic of n. In Chapter 7 we consider a diameter broadcast problem where we obtain a lower bound on the broadcast time of the graph. Finally, Chapter 8 is the conclusion and a short note on future work.

Chapter 2

Related Work

This chapter reviews the important contributions made so far in the field of broadcasting problem. There are several communication modes being investigated in this literature. We first present the one-way mode and two-way mode which belong to the most extensively studied ones. The other modes will be discussed after this.

- One-way mode (also called telegraph communication mode)

  In this mode, flow of message in a single round can be in one direction only i.e each node in a single round is active through one of its adjacent edges either as a sender or as a receiver. In Figure 2, in the first round the node x_1 broadcasts all its message to node x_2 and x_7 sends message to x_6. In the second round, x_2 sends to x_3 and x_6 sends to x_5. In the 3rd round x_3 sends to x_4 and in the 4th round finally x_4 gets informed from x_5.

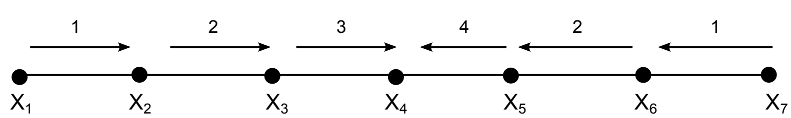

Figure 2: Telegraph Model

- Two-way mode (also called telephone communication mode)

  In this mode, in a single round, each node may be active through one of its adjacent edges. When it is active, it can simultaneously send message and receive message through this active edge. Sometimes this is also known as full-duplex communication mode. It can be easily observed that when one edge is used to transmit the message, the information flow is bi-directional. One can generalize this broadcast model. (i, j)-mode means that in any round one node can send message to i neighbors via i adjacent edges. At the same time it can receive messages from j neighbors through j adjacent edges. Thus, the two-way mode is a restricted $(1, 1)$-mode where any active node will use the same adjacent edge for both sending and receiving messages [55].

Based on the number of neighboring processors that can be communicated simultaneously, broadcast models are classified into:-

- *1-port communication model (or processor bound model)* where a node communicates with one neighbor at a time [70].

- *k-port communication model* where a processor can communicate with at most k of its neighbors at a time. A considerable amount of study [109], [108], [93], [94], [98], [130], [160] is dedicated to this model. This is useful in the area of DMA-bound systems [136] as well as in computing functions in networks [6], [22], [42].

- *link bound model* where all the neighbors can be informed at the same time [70].

Broadcast models can also be classified based on the time taken to send a message between the two nodes of the network.

1. *the constant model* where irrespective of the size of the message, time taken to broadcast it to another node is constant

2. *the linear model* where time needed to broadcast a message to a neighboring node is a linear function of the size of the message.
   There are some results with linear model [11], [150], [41], [65], [40] though the literature mainly deals with the constant model.

Again, sometimes in the literature broadcast model has been classified depending on how the communication has been setup with the neighbors.

1. *Vertex disjoint path mode broadcasting* where in every round information is being transmitted to the uninformed nodes via disjoint sets of vertices, which can be paths of length greater than one. There are two types of this model [21], [58], [61], [66], [67], [69], [118], [131] that have been studied in the literature. In one class of this model, one end-node broadcasts its whole piece of information to all other nodes along the path. In the second class of this model, the intermediate nodes in the path do not read the message being sent.

2. *Edge disjoint path mode broadcasting* where in every round information is being transmitted to the uninformed nodes via disjoint sets of edges, which can be paths of length greater than one. This model has been investigated in several papers [58], [61], [66], [68], [116], [119]. Note that both vertex disjoint and edge disjoint models are called *Line broadcast model*.

3. (i, j) *mode broadcasting* where in any round, a node can inform its i neighbors via i incident edges and it can receive messages from j neighbors via j incident edges. This model has been studied in [55].

4. *Radio Broadcasting* where the transmission of the message is assumed to take place in discrete pulses or rounds. In this model, on each communication round, each informed node can either inform all its neighbors or not send it at all i.e., it is not allowed to send to a subset of its neighbors at a time. Moreover, the node that receives the message from precisely one neighbor is considered to be informed at that round since in this model, it is assumed that if a node receives message from more than one neighbor at the same time, then the message is corrupted. There are several literature on the study of this model [3], [4], [10], [30], [28], [31], [29], [51], [52], [54], [53], [74], [76], [123], [132].

5. *Universal list broadcast model* where every vertex knows in prior the ordered list of neighbors that it is going to inform [43], [128], [111], [96]. This model differs from the classical broadcast model where every vertex can choose the ordered list of nodes it will inform depending on the source vertex.

6. *Messy broadcasting model* where each vertex sends the message randomly to its neighbors without any knowledge about the originator or the time at which

the message was sent. In other words, messy broadcasting model is looking for upper bounds in the broadcast time, following the constraints below:

- one node knows only its neighbors
- the originator is not known
- the time slot is not known
- there is no co-ordinating leader

This model is best suited in a topology which has insufficient memory to maintain a co-ordinated protocol. One of the major differences between the messy broadcasting and the previously mentioned topologies is that, in a messy broadcast scheme, the vertices at each round send the message to a randomly selected neighbor, without having the knowledge of the network topology [2], [35], [91], [89], [138], [85], [86]. There are 3 different types of messy models being studied in the literature:

- *M1 model* where at each time unit, every vertex knows the state of each of its neighbors i.e. informed or uninformed.
- *M2 model* where every informed vertex knows the source vertex from which it receives the message and also the neighbors to which it has sent the message.
- *M3 model* where every informed vertex knows the neighbors to which it has sent the message.

7. *Multiple message broadcasting* where large amounts of data are broken into smaller pieces of information which are then sent individually over the network [8], [9], [27], [33], [57], [87], [88], [140], [126], [127], [32].

8. *Fault tolerant broadcasting model* where it is assumed that some links in the network can be faulty. Thus in a k fault-tolerant broadcasting scheme, it is assured that any node in the network can receive the message from the originator in presence of at most k edge failures [1], [77], [79], [149].

9. *Broadcasting with randomly placed calls* originated from the spreading of rumour studies where each informed member of a population transmits the message to

other members of the population. Later a slightly different model is being introduced, in which an informed node transmits the information on average f times, where f is a function of time. Further work on this model can be found in [14], [24], [38], [39], [48], [62], [82], [81], [80], [75], [124], [134], [135], [146], [151], [152], [153], [154], [155].

Although there are several broadcast models, however in this thesis we will consider the classical broadcast model. Recall from the Introduction chapter, in the classical broadcast model, broadcasting is to be completed as quickly as possible subject to the following constraints: (1) Each call requires one unit of time. (2) A vertex can participate in only one call per unit of time. (3) Each call involves only two adjacent vertices, a sender and a receiver. In this model, the broadcast problem of determining $b(u)$ for an arbitrary originator u in an arbitrary graph is proved to be NP-Complete in [162]. This NP-Complete problem is as follows:
Given a graph $G = (V, E)$ with a specified set of vertices $V_0 \subseteq V$ and a positive integer k, does there exist a sequence $V_0, E_1, V_1, E_2, V_2, ..., E_k, V_k$, where $V_i \subseteq V$, $E_i \subseteq E (1 \leq i \leq k)$, $E_i = \{(u, v), u \in V_{i-1}, v \notin V_{i-1}\}$, $V_i = V_{i-1} \cup v$ and $V_k = V$. Here V_i is the set of informed vertices at round i, E_i is the set of active edges through which information is being sent at round i and k is the total broadcast time. When $|V_0| = 1$, then it is the case when broadcasting starts from an arbitrary single originator. The proof has been done by reducing the 3-dimensional matching problem (3DM) to the broadcast problem in polynomial time.

2.1 Approximation Algorithms and Heuristics

Since finding the minimum broadcast time of any originator in an arbitrary graph has been proved to be NP-Complete, many approximation algorithms and heuristics have been presented to determine the broadcast scheme with minimum time cost (see [7], [131], [49], [73], [72], [50], [71], [12], [156], [158], [60], [145]). The first work of this kind in [131] gives us a broadcast scheme whose performance is at most $b(u, G) + Diam(u) + 3\sqrt{|V|}$ rounds for a given graph $G = (V, E)$ and the originator u. Here $Diam(u)$ is the diameter of u and $b(u, G)$ is the optimal broadcast time. This gives us an $O(\sqrt{|V|})$ additive approximation algorithm. A randomized broadcast algorithm has been presented in [156] which is based on calculating the poise of a graph. The poise of a tree T is defined as the sum of the maximum degree of a vertex in the tree and the diameter of the tree. The poise of a graph G, denoted by $P(G)$, is defined as the minimum poise of any spanning trees. Calculating the poise of a graph is NP-Complete. Ravi in [156] presents a heuristic to compute a spanning tree of a graph on n vertices and m edges which runs in $O(nm \log n)$ time. The paper also shows there is a $O(\log(n)P(G) + \log^2 n)$ algorithm to calculate the poise of the tree and $b(G) = O(P(G)\frac{\log n}{\log \log n})$. The time complexity of the algorithm is $O(nm \log^2 n)$ and the upper bound of the broadcast time is $O(\frac{\log^2 n}{\log \log n} b(G))$. The best theoretical upper bound is obtained by the approximation algorithm in [49] which produces a broadcast scheme with $O(\frac{\log(|V|)}{\log \log(|V|)} b(G))$ rounds. Research in [159] has showed that the broadcast time cannot be approximated within a factor $\frac{57}{56} - \epsilon$. However this result has been improved within a factor of $3 - \epsilon$ in [49].

In the heuristic approach, researchers tried to match between the set of informed and the set of uninformed vertices in every round of calls. The Round-Heuristic described in [12] presents the simulation results which guarantees the performance of this algorithm is quite close or equal to the optimal value. The running time of Round-Heuristic is $O(Rnm \log n)$ where R is the number of rounds taken for broadcasting, n is the number of vertices and m is the number of edges in the graph. Another heuristic known as Tree-based Approach in [110] reduces the complexity of each round to $O(m)$. Both these approaches perform better in most of the commonly used interconnection networks and also produce better results in the graph models from the network simulator ns-2 [141], [7], [47], [23]. Recent heuristic approaches in [112] apply Random Heuristic and Semi-Random Heuristic algorithms which both reduce the total time complexity to $O(m)$ and the simulation results show that these new heuristics perform better than the previous approaches in the models representing real networks. Both these algorithms first generate a shortest path for every vertex to receive the message in the network. While Random Heuristic makes random decisions when matching children and parents, Semi-Random employs a strategy to distribute the children to the parents.

2.2 Commonly Used Topologies

In this section we present a family of commonly used graph topologies with well studied properties related to interconnection networks such as the diameter, the number of edges, the maximum degree, the broadcast time and others (see [70], [116], [137], [113]). For some of these graphs, the exact value of broadcast time is not known. In such cases the best known lower and upper bounds have been presented.

2.2.1 The Path P_n

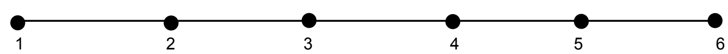

Figure 3: Path with n = 6

A path in a graph is a sequence of vertices such that from each of its vertices there is an edge to the next vertex in the sequence. In the path of length n denoted by P_n the nodes are all integers from 1 to n and the edges connect each integer i ($1 \leq i < n$) with $i + 1$. P_n has n vertices, diameter is equal to $n - 1$ and maximum degree 2. The broadcast time of P_n is equal to $n - 1$. This is because the end vertices have the maximum broadcast time in the path. In Figure 3, $b(P_6) = 5$.

2.2.2 The Cycle C_n

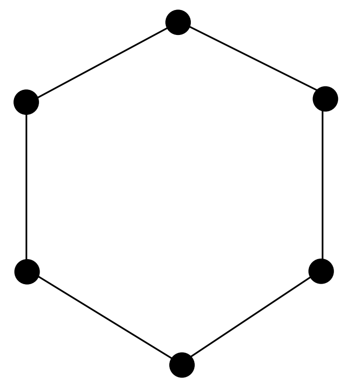

Figure 4: Cycle with n = 6

A cycle (ring) is a path such that the start vertex and the end vertex are also connected by an edge. C_n has n vertices, diameter is equal to $\lfloor \frac{n}{2} \rfloor$ and maximum degree 2. The broadcast time of C_n is equal to $\lceil \frac{n}{2} \rceil$. In Figure 4, $b(C_6) = 3$.

2.2.3 The Complete Graph K_n

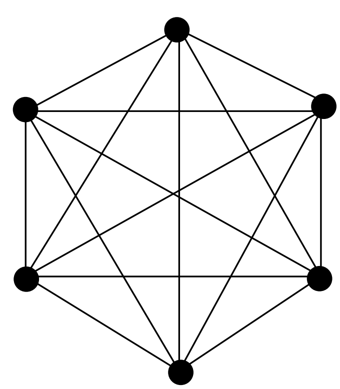

Figure 5: Complete Graph with n = 6

A complete graph is a simple undirected graph in which every pair of distinct vertices is connected by a unique edge. K_n has n vertices, diameter is 1 and degree $n-1$. The broadcast time of K_n is equal to $\lceil \log n \rceil$ as during each round every informed vertex can send message to an uninformed neighbor. In Figure 5, $b(K_6) = 3$.

2.2.4 The Hypercube H_m

The hypercube of dimension m, denoted by H_m is the graph whose vertices are all binary strings of length m and whose edges connect those binary strings which differ in exactly one position. H_m has 2^m vertices, $m2^{m-1}$ edges, diameter m and each vertex has exactly degree m. An $(m+1)$-dimensional hypercube is constructed from two m-dimensional hypercubes by connecting each pair of the corresponding vertices. The hypercube is one of the few family of graphs where the broadcast time is equal to $\log(|V|)$ where $|V|$ denotes the number of nodes in the hypercube, i.e. $b(H_m) = m$. In Figure 6, $b(H_3) = 3$.

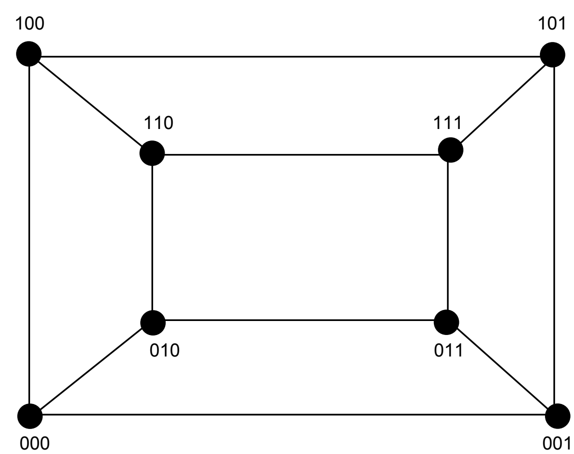

Figure 6: Hypercube H_3

2.2.5 The Cube-Connected Cycles CCC_m

The CCC_m is a modification of the hypercube H_m obtained by replacing each vertex of the hypercube with a cycle of m nodes. The i-th dimension edge incident to a node of the hypercube is then connected to the i-th node of the corresponding cycle of the CCC_m. This CCC_m has $m2^m$ vertices, diameter $\left\lfloor \frac{5m}{2} \right\rfloor - 1$ and maximum degree 3. From [139] we know that $b(CCC_m) = \left\lceil \frac{5m}{2} \right\rceil - 1$. Figure 7 shows a 3-dimensional cube-connected cycle.

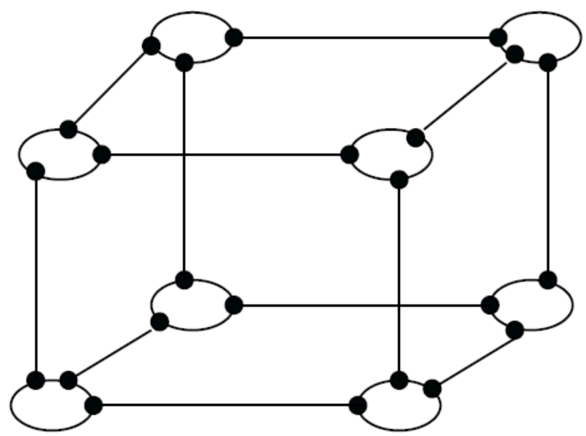

Figure 7: Cube-Connected Cycle CCC_3

2.2.6 The Butterfly BF_m

The m-dimensional butterfly network, BF_m is a graph with vertex-set $V_m = \{0, 1, ..., m-1\} \times \{0,1\}^m$, where $\{0,1\}^m$ denotes the set of length-m binary strings. For each vertex $v = \langle i, \alpha \rangle \in V_m, i \in \{0, 1, ..., m-1\}, \alpha \in \{0,1\}^m$, we call i the level and α the position-within-level of v. The edges of BF_m are of two types: For each $i \in \{0, 1, ..., m-1\}$ and each $\alpha = a_0 a_1 ... a_{m-1} \in \{0,1\}^m$, the vertex $\langle i, \alpha \rangle$ on level i of BF_m is connected

- by a straight-edge with vertex $\langle (i+1) \bmod m, \alpha \rangle$ and

- by a cross-edge with vertex $\langle (i+1) \bmod m, \alpha(i) \rangle$

on level $(i+1)$mod m. Here, $\alpha(i) = a_0 ... a_{i-1} c_i a_{i+1} ... a_{m-1}$, where c_i denotes the binary complement of a_i. The BF_m has $m2^m$ vertices, diameter $\left\lfloor \frac{3m}{2} \right\rfloor$ and maximum degree 4. From [148] we know that $1.7417m \leq b(BF_m) \leq 2m - 1$. Figure 8 shows a 3-dimensional butterfly network.

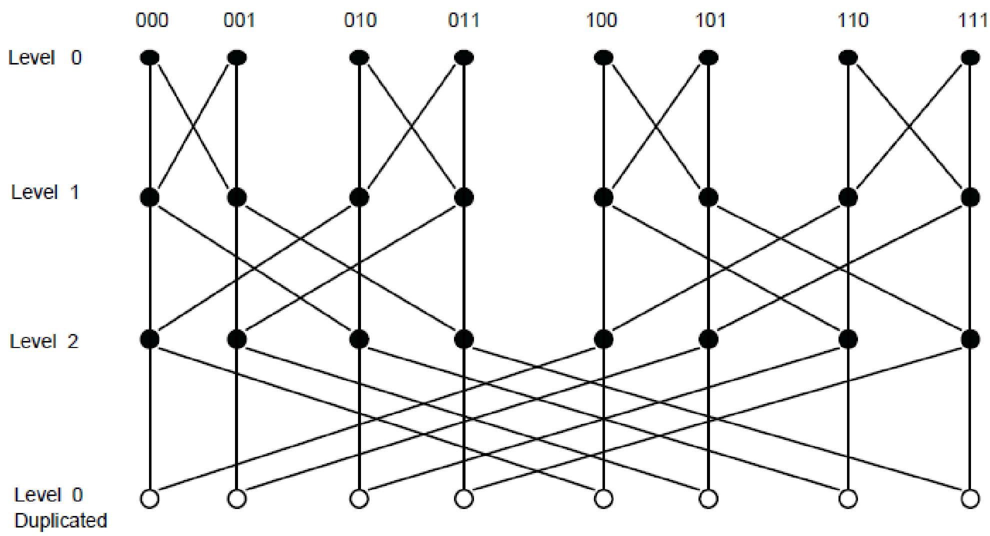

Figure 8: Butterfly graph, m = 3

2.2.7 The Shuffle-Exchange SE_m

The SE_m is the graph where the vertices are represented by binary strings of length m and whose edges connect each string αa, where α is a binary string of length $m-1$ and a is in $\{0, 1\}$, with the string αc and with the string αa, where c is the binary complement of a. The SE_m has 2^m vertices, diameter is $2m-1$ and maximum degree 3. From [122] we know that $b(E_m) \leq 2m - 1$. Figure 9 shows a 3-dimensional shuffle-exchange graph.

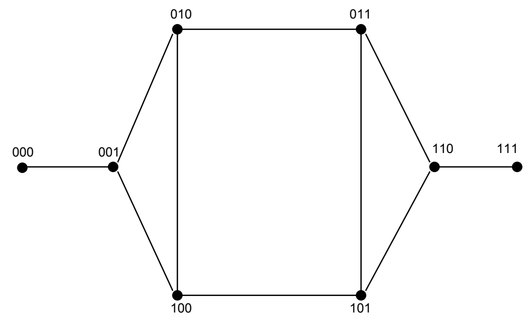

Figure 9: Shuffle-Exchange graph SE_3

2.2.8 The DeBruijn DB_m

The DB_m is the graph where the vertices are represented by binary strings of length m and whose edges connect each string αa, where α is a binary string of length $m-1$ and a is in $\{0,1\}$, with the strings αb, where b is a symbol in $\{0,1\}$. The DB_m has 2^m vertices, diameter is m and maximum degree 4. From [148] we know that $b(DB_m) \geq 1.3171m$ and from [19] we know that $b(DB_m) \leq 1.5m + 1.5$. Figure 10 shows a 3-dimensional DeBruijn graph.

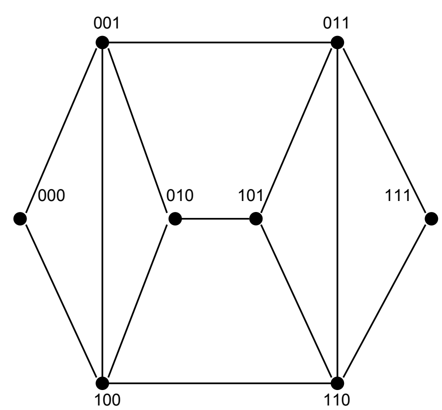

Figure 10: DeBruijn graph DB_3

2.2.9 2d Grid Network $G_{m,n}$

The 2 dimensional grid network $G_{m,n}$ (or mesh) is a network on mn vertices. A vertex having the tuple (i,j) is connected to a maximum of 4 vertices denoted by the tuples

$(i-1,j)$, $(i,j-1)$, $(i+1,j)$, $(i+1,j+1)$ for $1 < i < m$ and $1 < j < n$. The corner vertices are connected to 2 neighbors only, for example $(0,0)$ is connected to $(0,1),(1,0)$. Other than the corner vertices, all other vertices which are on the sides have 3 neighboring vertices, for example $(0,j)$ is connected to $(0,j-1)$, $(0,j+1)$, $(1,j)$. From [113] we know that $b(G_{m,n}) = m+n-2$. New heuristics in [110], [44] have found better results on the performance of various broadcast schemes in grids. Figure 11 shows a 2-grid graph $G[3\times 4]$.

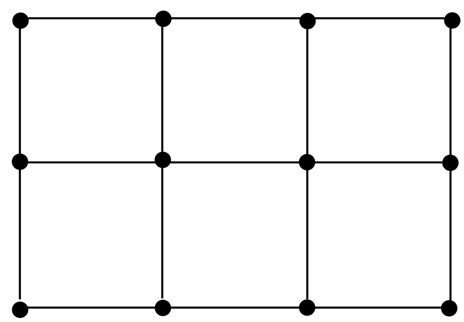

Figure 11: The Grid $[3 \times 4]$

2.2.10 The d-Torus $T[a_1 \times a_2 \times ... \times a_d]$

A d-Torus graph is a d-grid graph with both ends of rows and columns connected. The bounds on the broadcast time of the Torus are $D \leq b(T[a_1 \times a_2 \times ... \times a_d]) \leq D + max(0, m-1)$, where $D = \sum_{i=1}^{d} a_i - d$, and m is the number of odd a_i. Figure 12 shows a 2-torus graph $T[3 \times 4]$.

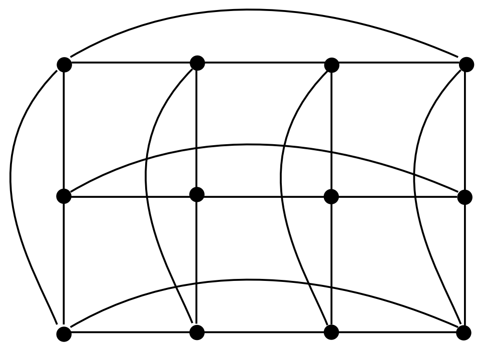

Figure 12: 2-Torus graph with 12 vertices

2.2.11 The k-ary Tree

The complete k-ary tree of height m, denoted by T_k^m , is the graph whose nodes are all k-ary strings of length at most m and whose edges connect each string α of length $i(0 \leq i \leq m)$ with the strings αa, $\alpha \in \{0, ..., k-1\}$, of length $i+1$. The nodes at level m are the leaves of the tree. For a node α at level $i, (0 \leq i < m)$, the nodes αa, $a \in 0, ..., k-1$, are called the children of α. α is called the parent of αa. T_k^m has $(k_{m+1} - 1)/(k-1)$ nodes, diameter $2m$ and maximum degree $k+1$. If v_0 is the root of T_k^m , then broadcast time of the tree from v_0 is equal to km. In Figure 13, $b(v_0, T_3^2) = 6$.

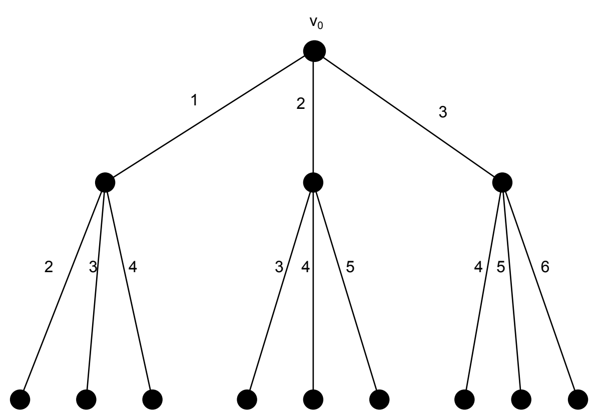

Figure 13: Complete Tree T_3^2

2.2.12 Unicyclic Graphs

A unicyclic graph is a connected graph with one cycle. It can be also represented as a cycle where every vertex on the cycle is a root of the tree. Figure 14 shows a unicyclic graph where the vertices of the cycle are denoted by $r_1, r_2, ..., r_k$ and the tree rooted at r_i by T_i, where $1 \leq i \leq k$.

From [100] we know that if $G = (V, E)$ is a unicyclic graph and T' is a spanning tree of G then $b_{min}(G) \leq b_{min}(T') \leq 2b_{min}(G) - 2$, where $b_{min}(G)$ is the minimum broadcast time of all the vertices in G. Similarly $b_{min}(T')$ is defined as the minimum of the broadcast times of all the k spanning trees that can be formed from the unicyclic graph having k trees. In [100], it has been also shown that $b(G) \leq b(T) \leq 2b(G) - 2$ where T is a spanning tree of G.

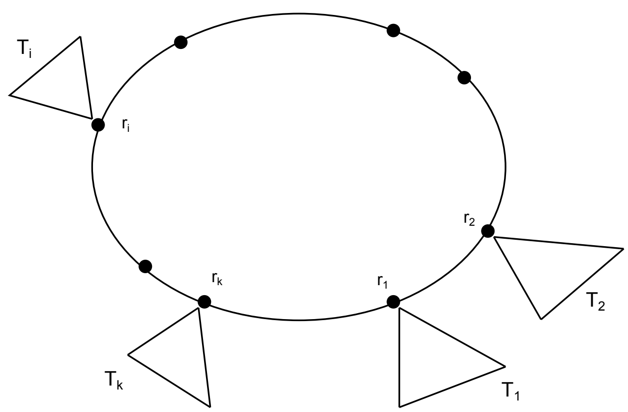

Figure 14: Unicyclic graph with trees T_i

2.2.13 Knödel graphs $W_{g,n}$

Knödel graphs $W_{g,n}$ are defined as undirected graph $G = (V, E)$ with $V = \{0, 1, ..., n-1\}$, n is even, and the set of edges $E = \{(i, j)|i + j = (2^k - 1) \text{mod } n\}$, where $0 \leq i, j \leq n - 1$, $1 \leq k \leq g$ and $1 \leq g \leq \lceil \log n \rceil$.
$b(W_{k,2^k}) = k$ [129] and $b(W_{k-1,2^k-1}) = k$ [125]

Several studies have been made on Knödel graphs in [16], [63], [64], [87], [106], [107], [129].

2.2.14 Other Topologies

There are several other graph topologies that have been considered in the literature besides the topologies mentioned above. Research in [34] has shown that an optimal broadcast algorithm is possible in directed graphs called the Manhattan street network. Broadcasting in generalized chordal rings has been studied in [36]. The work in [99] shows that the broadcast time of the optimal bipartite double loop graphs is $d + 2$ where d is the diameter of the graph. In [101] the optimal triple loop graphs have been considered and it has been proved that $d+2$ is the lower bound and $d+5$ is the upper bound for the broadcast problem in this kind of graph. In the same paper, a general upper bound of $d + 2k - 1$ has been given for multiple loop graphs where $2k$ is the degree of every vertex. In [104] a linear algorithm for broadcasting in networks with no intersecting cycles have been studied. In [103] it has been shown that a polynomial time solution is possible for broadcasting problems in fully connected trees. There is also a polynomial time solution for broadcasting in necklace graphs [90]. A constant approximation algorithm has been presented in [105] for hierarchical tree cluster networks. In [127] a constant factor approximation algorithm is given for network of workstations. The broadcast problem has also been studied for other topologies, such as Kautz graphs [115], pancake and star graphs [20], recursive circulants [147], banyan-hypercube [13], cycle-prefix digraphs [37]. Research has been made for networks under certain constraints, like bounded degree networks [17], [143] and planar graphs [114].

2.3 Minimum broadcast graphs

Sometimes, instead of finding the broadcast time of a specific graph, another approach is to find the graphs with minimum number of edges such that broadcast can be done within a certain amount of time. A broadcast graph is a connected graph G on n vertices such that $b(G) = \lceil \log n \rceil$. The broadcast function $B(n)$ is the minimum number of edges in any broadcast graph on n vertices. A *minimum broadcast graph* (mbg) is a broadcast graph on n vertices with $B(n)$ edges. Finding $B(n)$ is not easy even for small values of n. The exact value of $B(n)$ is known when $n = 2^k$ and $n = 2^k - 2$. Research has been done in constructing minimum broadcast graphs [84], [142], [133], [157]. Farley et al [59] showed that hypercubes are mbgs which results in $B(2^m) = m2^{m-1}$. This value is obtained by 3 non-isomorphic families of graphs: (1) the hypercube of dimension k [56], (2) the recursive circulant $G(2^k, 4)$ [147] and (3) the Knödel graph $W_{k,2^k}$ [129].

Khachatrian and Haroutunian [125] and Dinneen et al. [45] independently showed that $B(2^m - 2) = (m-1)(2^{m-1} - 1)$ for $m \geq 2$. $B(n)$ is known mostly for small values of n, mainly under 63.

- $1 \leq n \leq 16$ and $n = 32$ [59]
- $n = 17$ [144]
- $n = 18, 19$ [18], [164]
- $n = 20, 21, 22$ [142]
- $n = 26$ [157], [166]
- $n = 27, 28, 29, 58, 59, 60, 61$ [157]
- $n = 30, 31$ [18]
- $n = 62$ [56]
- $n = 63$ [133]
- $n = 127$ [165]
- $n = 1023, 4095$ [160]

It has been proved that it is very difficult to construct minimum broadcast graphs. So another direction of research has been to connect smaller broadcast graphs together to construct broadcast graphs on larger number of vertices [15], [25], [26], [45], [46], [78], [92], [125]. This approach is quite useful for designing graphs with even number of vertices. An upper bound on $B(n)$ for odd, positive n has been presented in [95]. Recently, a new improved upper bound on $B(n)$ appears in [5]. Figure 15 illustrates several examples of minimum broadcast graphs with authors name indicated in the parentheses.

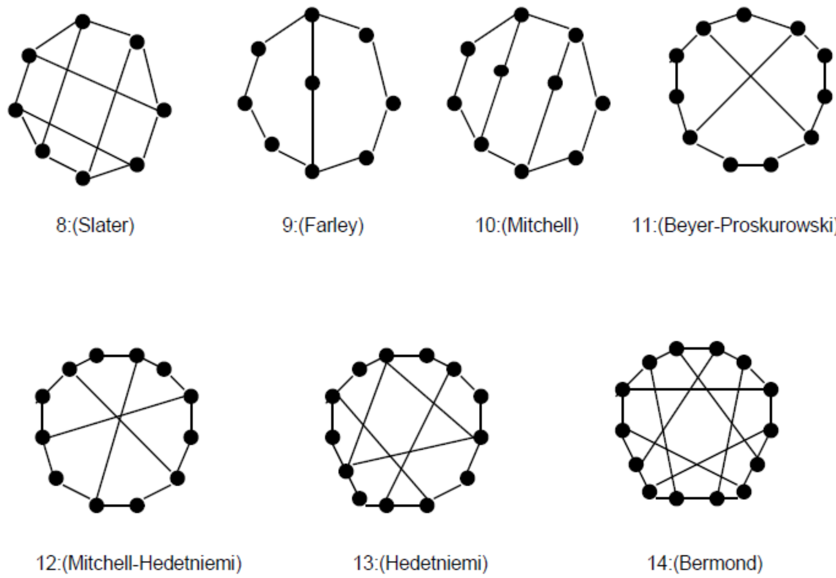

Figure 15: Minimum broadcast graphs

Chapter 3

Approximation Algorithm for the Broadcast Time in k-path Graph

As mentioned in Chapter 1, the broadcasting problem becomes very difficult when two cycles intersect. In this chapter we consider broadcasting in simple graphs where the intersection of two cycles is a path. The simplest such graph where several cycles have only two intersecting vertices is called a k-path graph. A k-path graph is a collection of k paths of arbitrary lengths connected by a vertex (or a junction) on both ends. We present a constant approximation algorithm to find the broadcast time of an arbitrary k-path graph. We also show the optimality of our algorithm for some subclasses of k-path graph.

3.1 Auxiliary Results

In this section we prove two auxiliary results which will be used later in designing of our approximation algorithm.

Definition 1. *Let $G_k = (V, E)$ be a connected graph consisting of k paths $P_1, P_2, P_3, ...,$ P_k and two vertices u and v connected to the end points of all paths. Vertices u and v are called junctions of G_k (see Figure 16).*

Let $l_1 \geq l_2 \geq ... \geq l_k \geq 1$, where l_i be the number of vertices in path P_i (excluding vertices u and v) for all $1 \leq i \leq k$.

First we assume that the originator is one of the junction vertices.

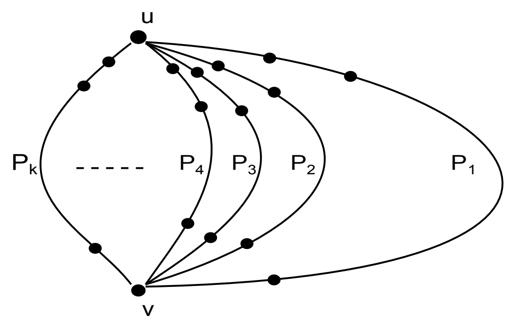

Figure 16: k-path graph

Lemma 1. *There exists a minimum time broadcast scheme from originator u in G_k in which the shortest path P_k is informed in the first time unit.*

Proof. We start with a minimum time broadcast scheme from originator u where v receives the message through some path P'_j and then construct another minimum time broadcast scheme where u informs P'_j at time unit one. Finally, we construct another broadcast scheme where we swap the order in which u informs along the shortest path P_k and path P'_j and prove that this is also a minimum time broadcast scheme. This will prove our claim.

Let S_{opt} be a minimum time broadcast scheme, $b_{S_{opt}}(u) = b(u, G_k)$ under which u informs its adjacent vertices of the k paths in some order P'_1, P'_2 ,...,P'_k with lengths l'_1,...,l'_k at time units $1, 2, ..., k$ respectively where $P'_1, P'_2, ..., P'_k$ is the permutation of the paths $P_1, ..., P_k$ and l'_1,...,l'_k is the permutation of l_1,...,l_k (see Figure 17). Let v receives the message through path P'_j at time unit $j - 1 + l'_j + 1 = l'_j + j$.

Step 1: Design a new broadcast scheme S_j where u informs the vertices of the k paths in the order P'_j, P'_1, P'_2,...,P'_{j-1}, P'_{j+1},...,P'_k at time units $1, 2, 3, ..., k$ shown in Figure 18. In this scheme, v gets informed at time unit $l'_j + 1$ instead of time unit $l'_j + j$ under scheme S_{opt}. We will show that S_j is also a minimum time broadcast scheme for originator u.

Note that paths $P'_{j+1}, ..., P'_k$ will get informed exactly at the same time unit under both schemes S_{opt} and S_j. However, under scheme S_j every vertex on the paths P'_1,

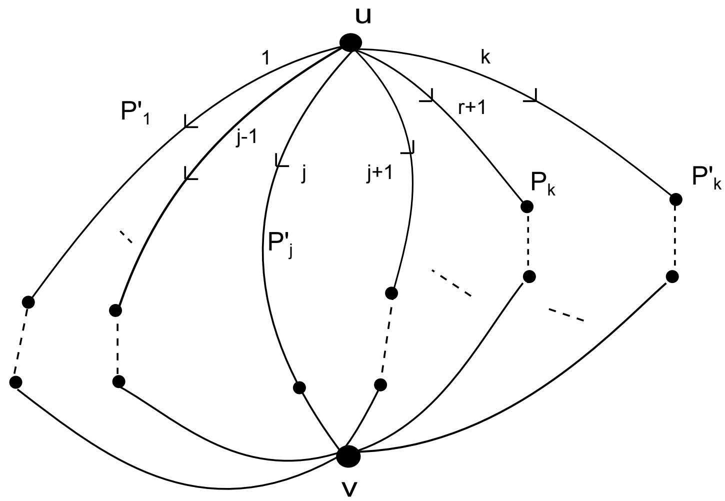

Figure 17: Scheme S_{opt}

P'_2,..., P'_{j-1} will receive the message exactly one time unit later. Recall that under S_{opt}, v is idle at time units $l'_j+2, l'_j+3, ..., l'_j+j$. However, under S_j, v receives the message $j-1$ time units earlier compared to scheme S_{opt} and can make $j-1$ extra calls, each one informing its adjacent vertices on paths $P'_1, ..., P'_{j-1}$ respectively. Thus, $b_{S_j}(u) \leq b_{S_{opt}}(u) = b(u, G_k)$. So, S_j is a minimum time broadcast scheme in which v gets informed through some path from u starting time unit 1.

Step 2: From scheme S_j, make a new scheme S_k where the times at which u sends the message along the paths P_k and P'_j are being swapped. Assume that under S_j, u informs the path P_k (a shortest path) at time $r+1$ for some $0 < r < k$. Then under S_k, u informs its adjacent vertices in the paths $P_k, P'_1, P'_2, ..., P'_j, ..., P'_k$ at time units $1, 2, 3, ..., r+1, ..., k$ respectively. The order in which u and v broadcast along the remaining $k-2$ paths is the same in both schemes. To prove that $b_{S_k}(u) = b_{S_j}(u) = b(u, G_k)$ we have to show that under S_k all vertices of path P'_j receive the message by time $b(u, G_k)$. There are two cases to consider:

Case 1: under S_j, v does not inform any vertex of P_k:
Under S_j, v is informed at time $l'_j + 1 \leq b(u, G_k)$ and u informs all the l_k vertices of P_k starting at time unit $r+1$. Similarly under S_k, v is informed at time l_k+1 and u informs at least l_k vertices on P'_j within $b(u, G_k)$. Since $l'_j + 1 \geq l_k + 2$ (otherwise

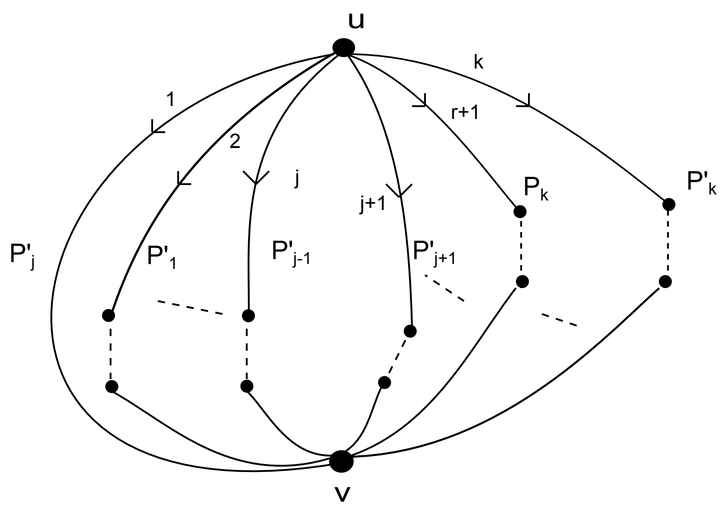

Figure 18: Scheme S_j

$l'_j \leq l_k$ means P'_j and P_k are identical paths), v has at least one free time unit immediately after $l_k + 1$ to inform along path P'_j at time unit $l_k + 2$. So, v can inform $b(u, G_k) - l_k - 1$ vertices on path P'_j. In total there are $b(u, G_k) - l_k - 1 + l_k = b(u, G_k) - 1$ informed vertices on P'_j under scheme S_k. Since $b(u, G_k) \geq l'_j + 1$ from scheme S_j, then $b(u, G_k) - 1 \geq l'_j$, and all vertices of P'_j will be informed within time $b(u, G_k)$ under scheme S_k. Since the broadcast time in the remaining $k - 2$ paths remains the same, $b_{S_k}(u) \leq b_{S_j}(u)$.

Case 2: Assume that under S_j, m vertices of P_k receive the message through vertex v starting at time $l'_j + 1 + c$ for some $c \geq 1$ (see Figure 19):
Under S_j, u informs $l_k - m$ vertices on P_k starting at time $r + 1$. Similarly under S_k, u informs at least $l_k - m$ vertices on P'_j within $b(u, G_k)$. As in Case 1, v informs along P'_j at time $l_k + 2$. Thus, v can inform $l'_j + c - (l_k + 1)$ vertices on P'_j before $l'_j + 1 + c$ time units in addition to another m vertices on P'_j before $b(u, G_k)$. Together there are $(l'_j + c - (l_k + 1)) + m + (l_k - m) = l'_j + c - 1$ informed vertices on P'_j. But $l'_j + c - 1 \geq l'_j$ since $c \geq 1$. This shows that all vertices on P'_j can be informed within the optimal broadcast time under S_k. Since the broadcast time in the remaining $k - 2$ paths remains the same, then $b_{S_k}(u) \leq b_{S_j}(u) = b(u, G_k)$. □

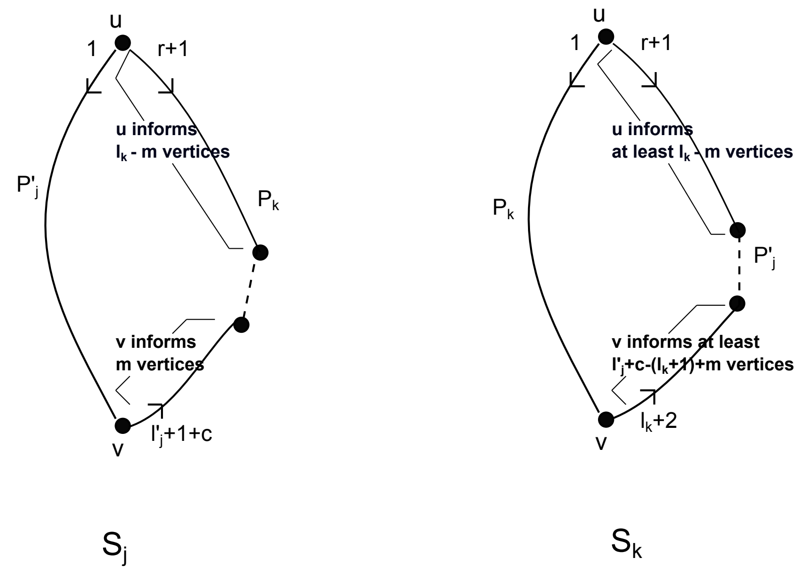

Figure 19: S_j and S_k where only P_k and P'_j are shown for the case where v informs P_k

Let us now consider the originator in G_k to be any vertex w on a path P_j, where $1 \leq j \leq k$ (see Figure 20). Let us assume that one of the junctions u is at a shorter distance from w and let the length of this shorter path $\overline{wu} = d$, where $d \geq 1$. Then the length of the path $\overline{wv} = l_j + 1 - d$ and $d \leq l_j + 1 - d$.

Lemma 2. *There is a minimum time broadcast scheme from originator w in G_k in which w first sends the information along the shorter path towards vertex u.*

Proof. Let S_v be a minimum time broadcast scheme, $b_{S_v}(w) = b(w, G_k)$ under which w first informs its adjacent vertex of the path $\overline{wv}$. We will construct a new broadcast scheme S_u under which w will first inform its adjacent vertex of the path $\overline{wu}$. We will show that $b_{S_u}(w) \leq b_{S_v}(w) = b(w, G_k)$.

According to scheme S_v, w informs its adjacent vertex of the path $\overline{wu}$ at time unit two, and u gets informed at time unit $d+1$. Now, we construct a new broadcast scheme S_u where w informs its adjacent vertex of the path $\overline{wu}$ at time unit one, and u is informed at time unit d. The order in which u and v broadcast along the remaining $k-1$ paths is the same in both schemes. However, under S_u, every vertex on sub-path $\overline{wv}$ of path P_j will receive the message exactly one time unit later compared to S_v.

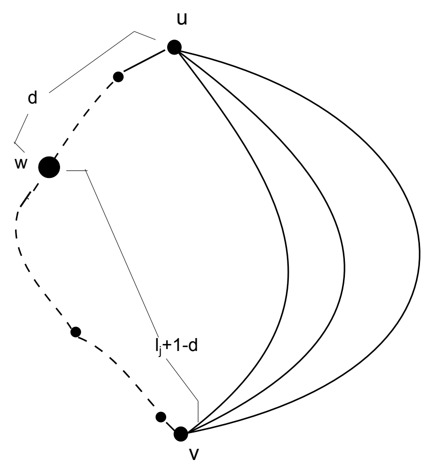

Figure 20: Originator w is any vertex other than junction

To prove that $b_{S_u}(w) = b(w, G_k)$ we consider three cases:

Case 1: under S_v, v is informed through some path other than P_j at time $b_1 \leq b(w, G_k)$:

Let this path be P_y. Since v is informed through P_y and $P_y \neq P_j$, then P_y must have been informed from u. Under S_u, u is informed exactly one time unit earlier. Subsequently every vertex on P_y will receive the message exactly one time unit earlier. So, v is informed at time unit $b_1 - 1$. v has exactly one free time unit immediately after $b_1 - 1$ to inform its adjacent vertex on P_j at time unit b_1. Since the broadcast time in the remaining $k-1$ paths remains the same, $b_{S_u}(w) \leq b_{S_v}(w)$.

Case 2: Assume that under S_v, v is informed through P_j and r vertices along the different paths in G_k receive the message through vertex v within $b(w, G_k)$ time units: Under S_u, v will receive the message exactly one time unit later compared to S_v. So, $r-1$ vertices along the different paths in G_k will receive the message through vertex v within $b(w, G_k)$. Let P_x be the path along which v informs one less vertex. Recall that u is informed exactly one time unit earlier. Thus, u can inform one extra vertex along P_x within $b(w, G_k)$. Since the broadcast time in the remaining $k-1$ paths remains the same, $b_{S_u}(w) \leq b_{S_v}(w)$.

Case 3: Assume that under S_v, v is informed through P_j and v does not inform a vertex in G_k:

Under S_v, v is informed at time unit $l_j + 1 - d$. Let the adjacent vertices of v in paths

P_1, P_2,..., P_j,..., P_k be v_1, v_2,..., v_j,..., v_k respectively. Since, under S_v, v does not inform any vertex in G_k, vertices v_1, v_2,..., v_{j-1}, v_{j+1},..., v_k must have been informed within l_j+1-d time units from u (see Figure 21). Recall that under S_u, u is informed exactly one time unit earlier and v_j will be informed at time unit l_j+1-d. Now vertices v_1, v_2,..., v_{j-1}, v_{j+1},..., v_k will be informed within l_j-d time units from u and one of them can inform v at time unit l_j+1-d. So, $b_{S_u}(w) \leq b_{S_v}(w) = b(w, G_k)$.

□

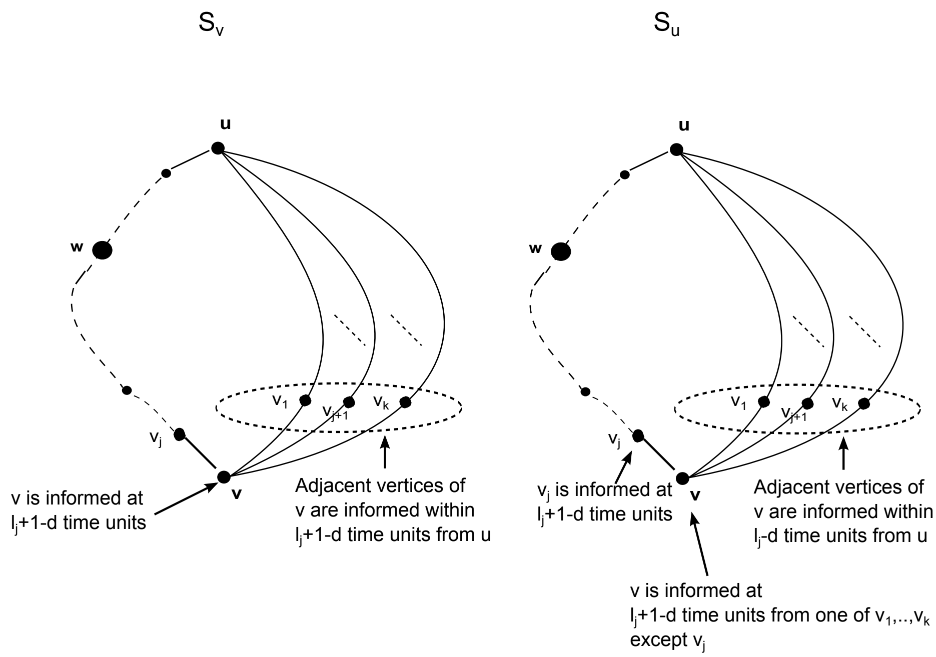

Figure 21: S_v and S_u where v does not inform a vertex in G_k

3.1.1 Lower bounds on broadcast time

In this section we will give lower bounds on the broadcast time of G_k from originators u and w. Recall that $l_1 \geq l_2 \geq ... \geq l_k \geq 1$ where l_i is the length of the path P_i (excluding vertices u and v) for all $1 \leq i \leq k$.

Lemma 3. *Let G_k be a k-path graph where the originator is a junction vertex u and $l_1 \geq l_2 \geq ... \geq l_k \geq 1$. Then*

(i) $b(u) \geq \left\lceil \frac{l_j+l_k+j+1}{2} \right\rceil$ *for any* j*,* $1 \leq j \leq k-1$*.*
(ii) $b(u) \geq \left\lceil \frac{k+l_k+1}{2} \right\rceil$ *if* $k > l_k + 1$*.*
(iii) $b(u) \geq \left\lceil \frac{2l_k+k+l_j+j+2}{4} \right\rceil$ *if* $k > l_k + 1$*.*

Proof. (i): By Lemma 1 there exists a minimum time broadcast scheme from originator u in G_k in which the shortest path P_k is informed in the first time unit. Considering such minimum time broadcast scheme, u informs along P_k at time unit one. It takes exactly $l_k + 1$ time units for vertex v to receive the message. Consider the cycle formed by the path P_k and any path P_j, where $1 \leq j \leq k-1$. Under any minimum time broadcast scheme all vertices in the cycle formed by these two paths must be informed. u informs the adjacent vertices of the remaining $k-1$ paths in some order and assume it informs along P_j at time unit $j+1$ or later. Then at time unit $l_k + 1$ there are at least $l_j - (l_k + 1 - j) = l_j - l_k + j - 1$ uninformed vertices in P_j. v sends the message along P_j no sooner than time unit $l_k + 2$. Since, starting at time $l_k + 2$ onwards, P_j receives the message from both u and v, then at each time unit 2 new vertices on P_j will get informed. So, $b(u) \geq l_k + 1 + \left\lceil \frac{l_j-l_k+j-1}{2} \right\rceil = \left\lceil \frac{l_j+l_k+j+1}{2} \right\rceil$. Suppose, by contradiction u calls path P_j before time $j+1$. Then by pigeonhole principle there exists m, $1 \leq m \leq j-1$ such that u calls P_m at time $j+1$. Similarly at time unit $l_k + 1$ there are at least $l_m - l_k + j - 1$ uninformed vertices in P_m. If, starting at time $l_k + 2$ onwards, P_m receives the message from both u and v, then $b(u) \geq l_k + 1 + \left\lceil \frac{l_m-l_k+j-1}{2} \right\rceil = \left\lceil \frac{l_m+l_k+j+1}{2} \right\rceil \geq \left\lceil \frac{l_j+l_k+j+1}{2} \right\rceil$ as $l_m \geq l_j$. Hence, $b(u) \geq \left\lceil \frac{l_j+l_k+j+1}{2} \right\rceil$.

Proof of (ii) goes as follows: v receives the message no sooner than $l_k + 1$ time units through P_k. After time $l_k + 1$, there are $k - (l_k + 1)$ paths with no informed vertices (see Figure 22). v will inform at least $\left\lceil \frac{k-l_k-1}{2} \right\rceil$ uninformed paths (u informs the remaining $\left\lfloor \frac{k-l_k-1}{2} \right\rfloor$ uninformed paths). So, $b(u) \geq l_k + 1 + \left\lceil \frac{k-l_k-1}{2} \right\rceil = \left\lceil \frac{k+l_k+1}{2} \right\rceil$.

For the proof of (iii), we combine the inequalities in (i) and (ii). We get $2b(u) \geq \left\lceil \frac{l_k+l_j+j+1}{2} \right\rceil + \left\lceil \frac{k+l_k+1}{2} \right\rceil \geq \left\lceil \frac{2l_k+k+l_j+j+2}{2} \right\rceil$. Hence, $b(u) \geq \left\lceil \frac{2l_k+k+l_j+j+2}{4} \right\rceil$ when $k > l_k + 1$. □

Lemma 4. *Let* G_k *be a* k*-path graph where the originator is a junction vertex* u *and* $l_1 \geq l_2 \geq ... \geq l_k \geq 1$*. Let* n *be the total number of vertices in* G_k*. Then*
(i) $b(u) \geq \left\lceil \frac{n-l_k-2}{2(k-1)} + \frac{k+l_k}{2} \right\rceil$ *if* $b(u) \geq k + l_k$*.*

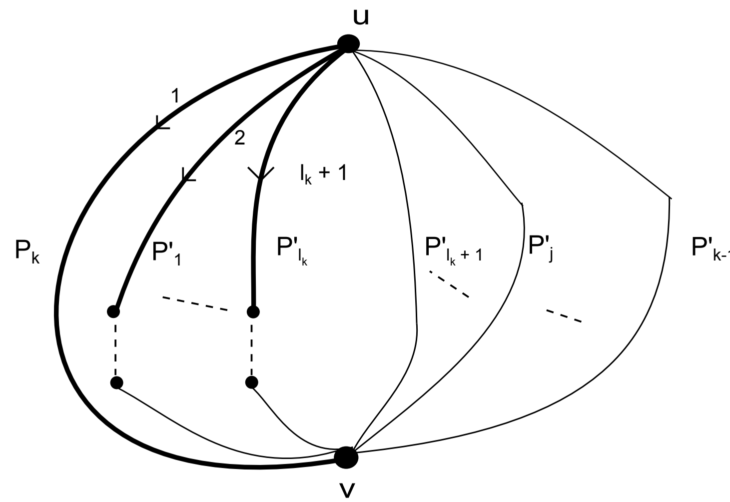

Figure 22: The paths marked in bold contain at least one informed vertex when v gets informed at time l_k+1. The rest $k-l_k-1$ paths $P'_{l_k+1},...,P'_{k-1}$ do not have any informed vertex at time l_k+1. $P'_1,...,P'_{k-1}$ is the combination of the paths $P_1,...,P_{k-1}$.

(ii) $b(u) \geq \left\lceil \frac{1}{3}\sqrt{(6n+4k^2-12k-6kl_k)-\frac{15}{4}}+\frac{6l_k-2k+9}{6} \right\rceil$ *if* $max\{k, l_k+1\} \leq b(u) < k+l_k$.

Proof. (i): By Lemma 1, u informs along the shortest path P_k at time unit one. It takes exactly l_k+1 time units for vertex v to receive the message. Since $b(u) \geq k+l_k$, then both u and v will be busy informing its adjacent vertices in the remaining $k-1$ different paths at time units $2, 3, ..., k$ and $l_k+2, ..., l_k+k$ respectively. By $b(u)$ time units, u can inform at most l_k+1, $b(u)-1$,..., $b(u)-(k-1)$ vertices in these k different paths. Similarly, by $b(u)$ time units, v can inform at most $b(u)-(l_k+1)$, $b(u)-(l_k+2)$,..., $b(u)-(l_k+k-1)$ vertices in the $k-1$ different paths. So, $n \leq l_k+1+\{b(u)-1 +...+ b(u)-(k-1)\}+\{b(u)-(l_k+1)+b(u)-(l_k+2) +...+ b(u)-(l_k+k-1)\}+1 \Rightarrow n \leq 2(k-1)b(u)-(k+l_k)(k-1)+(l_k+2)$. Hence, $b(u) \geq \left\lceil \frac{n-l_k-2}{2(k-1)}+\frac{k+l_k}{2} \right\rceil$.

Proof of (ii): Since $\max\{k, l_k+1\} \leq b(u)$, it guarantees that u can inform its adjacent vertices in k different paths at time units $1, 2, ..., k$ and v receives the message at time l_k+1. Similar to proof in (i), by $b(u)$ time units, u can inform at

most $l_k + 1$, $b(u) - 1$,..., $b(u) - (k-1)$ vertices in these k different paths. Similarly, by $b(u)$ time units, v can inform at most $b(u) - (l_k + 1)$, $(b(u) - (l_k + 1) - 1)$,..., $(b(u) - (l_k + 1) - (b(u) - (l_k + 2)))$ vertices as $b(u) < k + l_k$. So, $n \leq l_k + 1 + \{b(u) - 1 + ... + b(u) - (k-1)\} + \{b(u) - (l_k + 1) + (b(u) - (l_k + 1) - 1) + ... + (b(u) - (l_k + 1) - (b(u) - (l_k + 2)))\} + 1$
$= (k-1)b(u) - \frac{k(k-1)}{2} + (l_k + 2) + b(u)(b(u) - l_k - 1) - (l_k + 1)(b(u) - l_k - 1) - \frac{(b(u)-(l_k+2)(b(u)-(l_k+1))}{2}$.
$\Rightarrow 2n \leq 3b(u)^2 - b(u)(6l_k - 2k + 9) + (3l_k^2 + 9l_k - k^2 + k + 8)$
$\Rightarrow 3b(u)^2 - b(u)(6l_k - 2k + 9) - (2n + k^2 - 3l_k^2 - 9l_k - k - 8) \geq 0$.

Roots of $b(u)$ are $\frac{6l_k-2k+9 \pm \sqrt{(6l_k-2k+9)^2+12(2n+k^2-3l_k^2-9l_k-k-8)}}{6}$.

Considering the positive root of $b(u)$, we get $b(u) \geq \lceil \frac{6l_k-2k+9}{6} + \frac{\sqrt{(24n+16k^2-48k-24kl_k-15)}}{6}$
$\Rightarrow b(u) \geq \left\lceil \frac{1}{3}\sqrt{(6n + 4k^2 - 12k - 6kl_k) - \frac{15}{4}} + \frac{6l_k-2k+9}{6} \right\rceil$. □

Similarly we can obtain a lower bound on broadcast time when originator is any vertex w on a path P_j other than junction vertices u and v. Recall that $d \leq l_j + 1 - d$ where the lengths of the paths $\overline{wu}$ and $\overline{wv}$ respectively are d and $l_j + 1 - d$. Also $d \geq 1$ and l_j is the length of the path P_j .

Lemma 5. *Let G_k be a k-path graph where the originator w is any vertex on a path P_m and the lengths of the paths $\overline{wu}$ and $\overline{wv}$ respectively are d and $l_m + 1 - d$, where $d \geq 1$ and l_m is the length of the path P_m. Let us assume $l_m + 2 - 2d = \tau(m)$. If $l_1 \geq l_2 \geq ... \geq l_k \geq 1$ and $\tau(m) < l_k + 1$, then*
(i) $b(w) \geq d + \left\lceil \frac{l_j+\tau(m)+j-1}{2} \right\rceil$ for any j, $1 \leq j \leq k$ and $j \neq m$.
(ii) $b(w) \geq d + \left\lceil \frac{k+\tau(m)-1}{2} \right\rceil$ if $\tau(m) < k - 1$.
(iii) $b(w) \geq d + \left\lceil \frac{2\tau(m)+k+l_j+j-2}{4} \right\rceil$ if $\tau(m) < k - 1$ and $1 \leq j \leq k$ and $j \neq m$.

Proof. (i): By Lemma 2 there is a minimum time broadcast scheme from originator w in G_k in which w first sends the information along the shorter path towards vertex u. Considering this minimum broadcast scheme, u is informed no earlier than d time units and there are $l_m + 1 - d - (d-1) = l_m + 2 - 2d = \tau(m)$ uninformed vertices on path $\overline{wv}$ at time d. Since $\tau(m) < l_k + 1$, it takes at least $d + \tau(m)$ time units for v to receive the message. We consider the path along which v gets informed and any path P_j. Under any minimum broadcast scheme all the vertices in the cycle formed by these two paths must be informed. u informs the adjacent vertices of the $k-1$ paths in some order and let it informs P_j at time unit $d+j$. Then at time unit $d+\tau(m)$ there are at least $l_j-(d+\tau(m)-(d+j-1)) = l_j-\tau(m)+j-1$ uninformed vertices in P_j. v informs P_j no earlier than time unit $d+\tau(m)+1$. Similar to the argument given in the proof of Lemma 3(i), we can write $b(w) \geq d+\tau(m)+\left\lceil \frac{l_j-\tau(m)+j-1}{2} \right\rceil = d+\left\lceil \frac{l_j+\tau(m)+j-1}{2} \right\rceil$.

Proof of (ii) goes as follows: From the proof of Lemma 5(i) it follows that v receives the message no sooner than $d+\tau(m)$ time units. After time $d+\tau(m)$, there are $k-1-\tau(m)$ paths with no informed vertices. v will inform at least $\left\lceil \frac{k-1-\tau(m)}{2} \right\rceil$ uninformed paths (u informs the remaining $\left\lfloor \frac{k-1-\tau(m)}{2} \right\rfloor$ uninformed paths). So, $b(w) \geq d+\tau(m)+\left\lceil \frac{k-1-\tau(m)}{2} \right\rceil = d+\left\lceil \frac{k+\tau(m)-1}{2} \right\rceil$

For the proof of (iii), we combine the inequalities in (i) and (ii). We get $2b(w) \geq 2d+\left\lceil \frac{l_j+2\tau(m)+j+k-2}{2} \right\rceil$. Hence, $b(w) \geq d+\left\lceil \frac{l_j+2\tau(m)+j+k-2}{4} \right\rceil$ for the case when $k-1 > \tau(m)$ and $1 \leq j \leq k$ and $j \neq m$. □

Lemma 6. *Let G_k be a k-path graph where the originator w is any vertex on a path P_m and the lengths of the paths $\overline{wu}$ and $\overline{wv}$ respectively are d and l_m+1-d, where $d \geq 1$ and l_m is the length of the path P_m. Let us assume $l_m+2-2d = \tau(m)$. If $l_1 \geq l_2 \geq ... \geq l_k \geq 1$ and $\tau(m) \geq l_k+1$, then*
(i) $b(w) \geq d+\left\lceil \frac{l_j+l_k+j}{2} \right\rceil$ for any j, $1 \leq j \leq k-1$.
(ii) $b(w) \geq d+\lceil \frac{k+l_k}{2} \rceil$ if $l_k+1 < k-1$.
(iii) $b(w) \geq d+\left\lceil \frac{l_j+2l_k+j+k}{4} \right\rceil$ if $k-1 > l_k+1$ and for $1 \leq j \leq k-1$.

Proof. The proof of (i) is similar to the proof of Lemma 5(i) except that considering the minimum broadcast scheme as given in Lemma 2, v takes at least $d+l_k+1$ time units to get informed from u since $\tau(m) \geq l_k+1$. Similarly the number of uninformed vertices in P_j at time $d+l_k+1$ will be $l_j-(d+l_k+1-(d+j-1)) = l_j-l_k+j-2$. Here we consider $j=1$ for the path P_m as well as the path being informed from u at time $d+1$. Also $l_j = \tau(m)$ for the path P_m. v informs P_j no earlier than time unit $d+l_k+2$. Similar to the argument given in the proof of Lemma 3(i), we can get $b(w) \geq d+l_k+1+\left\lceil \frac{l_j-l_k+j-2}{2} \right\rceil = d+\left\lceil \frac{l_j+l_k+j}{2} \right\rceil$.

Proof of (ii): From the proof of Lemma 6(i) it follows that v receives the message no sooner than $d+l_k+1$ time units. After time $d+l_k+1$, there are $k-1-l_k-1 = k-l_k-2$ paths with no informed vertices. v will inform at least $\lceil \frac{k-l_k-2}{2} \rceil$ uninformed paths (u

informs the remaining $\left\lfloor \frac{k-l_k-2}{2} \right\rfloor$ uninformed paths). So, $b(w) \geq d + l_k + 1 + \left\lceil \frac{k-l_k-2}{2} \right\rceil$ $= d + \left\lceil \frac{k+l_k}{2} \right\rceil$.

For the proof of (iii), we combine the inequalities in (i) and (ii). We get $2b(w) \geq 2d + \left\lceil \frac{l_j+2l_k+j+k}{2} \right\rceil$. Hence, $b(w) \geq d + \left\lceil \frac{l_j+2l_k+j+k}{4} \right\rceil$ for the case when $k-1 > l_k+1$ and for $1 \leq j \leq k-1$. $\square$

3.2 Approximation Algorithm

In this section we present the broadcast algorithm S_{path} for graph G_k. We consider any vertex x to be the originator. When the originator is u then the algorithm S_{path} in G_k starts by informing the shortest path P_k in the first time unit. Starting at time two onwards u informs the path having the maximum number of vertices. When v gets informed at time $l_k + 1$, calculate the number of uninformed vertices in each of the paths $P_1, P_2, ..., P_{k-1}$. Starting at time $l_k + 2$ onwards v informs the path with the maximum number of uninformed vertices till v does not have any adjacent uninformed vertex.

When the originator is a non junction vertex w on path P_m then the algorithm S_{path} in G_k starts by informing the shorter path $\overline{wu}$ in the first time unit and then along the longer path $\overline{wv}$. u receives the message at time d. At time d there will be $d-1$ informed vertices on $\overline{wv}$ and $l_m + 1 - d - (d-1) = \tau(m)$ uninformed vertices. Depending on the relationship between $\tau(m)$ and the length of P_k, u decides its broadcast strategy. If $\tau(m) < l_k + 1$, u broadcasts along the path having the maximum number of vertices from the remaining paths starting at time $d+1$. If $\tau(m) \geq l_k+1$, u broadcasts along P_k at time $d+1$ and starting at time $d+2$ onwards, it informs the path having the maximum number of vertices. When v gets informed at time d+min$\{\tau(m), l_k+1\}$, calculate the number of uninformed vertices in each of the paths. Starting at time d+min$\{\tau(m), l_k+1\}$+1 onwards, v informs the path with the maximum number of uninformed vertices till v does not have any adjacent uninformed vertex.

Approximation Algorithm S_{path}:

INPUT: A k-path graph G_k where $l_1 \geq l_2 \geq ... \geq l_k \geq 1$ and any originator x

OUTPUT: Broadcast time $b_{S_{path}}(x)$ and scheme of G_k

BROADCAST-SCHEME-$S_{path}(G, l_1 \geq l_2 \geq ... \geq l_k \geq 1, x)$

1. If $x = u$
    - 1.1. u broadcasts to P_k in the first time unit.
    - 1.2. For each time unit $i = 2$ to k
        - 1.2.1. u broadcasts along P_{i-1} at time i
    - 1.3. v gets informed at time $l_k + 1$.
2. If $x = w$ and w is on path P_m
    - 2.1. w broadcasts along the shorter path $\overline{wu}$ in the first time unit and then along the longer path $\overline{wv}$.
      When u gets informed at time d, there are still $\tau(m)$ uninformed vertices along $\overline{wv}$.
    - 2.2. If $\tau(m) < l_k + 1$
        - 2.2.1. For each time unit $i = 1$ to k & $i \neq m$
            - 2.2.1.1. u broadcasts along P_i at time $d + i$
    - 2.3. Else-If $\tau(m) \geq l_k + 1$
        - 2.3.1. u broadcasts along P_k at time $d + 1$.
        - 2.3.2. For each time unit $i = 1$ to $k - 1$ & $i \neq m$
            - 2.3.2.1. u broadcasts along P_i at time $d + i + 1$.
    - 2.4. v gets informed at time d+min$\{\tau(m), l_k + 1\}$.
3. Calculate the number of uninformed vertices λ_j in P_j for $j = 1, 2, ..., k - 1$ when v gets informed
4. Arrange λ_j in decreasing order such that $\lambda'_1 \geq \lambda'_2 \geq ... \geq \lambda'_{k-1}$ where $\lambda'_1, \lambda'_2, ..., \lambda'_{k-1}$ is the permutation of $\lambda_1, ..., \lambda_{k-1}$.
   If P'_j contains λ'_j uninformed vertices then,
5. For each time unit $i = 1$ to k
    - 5.1. If v has an uninformed adjacent vertex in P'_i
        - 5.1.1. If $x = u$
            - 5.1.1.1. v broadcasts to P'_i at time $l_k + 1 + i$
        - 5.1.2. If $x = w$
            - 5.1.2.1. v broadcasts along P'_i at time d+min$\{\tau(m), l_k + 1\}$+i

Complexity Analysis:

Step 3 takes $O(k)$ time to calculate the number of uninformed vertices in $k-1$ paths.

Sorting them in decreasing order in step 4 takes $O(k \log k)$ time. Broadcasting done in steps 1, 2 and 5 can be accomplished in $O(|V|)$ time. Adding all the complexities we get that the complexity of the algorithm is $O(|V| + k \log k)$.

Recall that in algorithm S_{path}, when the originator is w on a path P_m, u receives the message from w at time d and there are $\tau(m)$ uninformed vertices on path $\overline{wv}$.

Theorem 1. *Algorithm S_{path} is a $(1.5 - \epsilon)$-approximation for any originator in the k-path graph G_k when $k \leq l_k + 1$, where $0 < \epsilon < 0.5$.*

Proof. 1) <u>when originator is u</u>:

Under algorithm S_{path}, u informs along P_j at time $j+1$ for $1 \leq j \leq k-1$. When v gets informed at time l_k+1, P_j has $l_j - (l_k+1-j)$ uninformed vertices. v can inform at most $k-1$ neighboring vertices along $k-1$ different paths $P_1, ..., P_{k-1}$. In the worst case, consider any path P_j that receives the message from v at time unit $l_k+1+k-1$. The number of uninformed vertices in path P_j before time unit $l_k+1+k-1$ will be $l_j - (l_k+1-j) - (k-2) = l_j - l_k + j - k + 1$. Since starting at time $l_k+1+k-1$ onwards, P_j receives the message from both u and v, then at each time unit 2 new vertices on P_j will get informed. So, $b_{S_{path}}(u) \leq l_k + 1 + k - 2 + \left\lceil \frac{l_j - l_k + j - k + 1}{2} \right\rceil = \left\lceil \frac{l_j + l_k + j + k - 1}{2} \right\rceil \leq \frac{l_j + l_k + j + k}{2}$.

Using Lemma 3(i), we can write $\frac{b_{S_{path}}(u)}{b(u)} \leq \frac{l_j + l_k + j + k}{l_j + l_k + j + 1} \leq \frac{l_j + l_k + j + l_k + 1}{l_j + l_k + j + 1}$ (as $k \leq l_k + 1$) $= 1 + \frac{l_k}{l_j + l_k + j + 1}$

Since $j \geq 1$ and $l_k \leq l_j$, $\frac{b_{S_{path}}(u)}{b(u)} \leq 1 + \frac{l_k}{2l_k+2} < 1.5$ since $l_k \geq 1$.

2) <u>when originator is w on a path P_m</u>:

<u>*Case 1:*</u> $k - 1 \leq \tau(m) < l_k + 1$:

Under algorithm S_{path}, u informs along P_j at time $d + j$ where $1 \leq j \leq k$ and $j \neq m$. When v gets informed at time $d + \tau(m)$, P_j has $l_j - (\tau(m) - j + 1)$ uninformed vertices. v can inform at most $k - 1$ neighboring vertices along $k - 1$ different paths $P_1, ..., P_k$ except P_m. In the worst case, consider any path P_j that receives the message from v at time unit $d + \tau(m) + k - 1$. The number of uninformed vertices in path P_j before time unit $d+\tau(m)+k-1$ will be $l_j - (\tau(m) - j + 1) - (k-2) = l_j - \tau(m) + j - k + 1$.

Since starting at time $d + \tau(m) + k - 1$ onwards, P_j receives the message from both u and v, then $b_{S_{path}}(w) \leq d + \tau(m) + k - 2 + \left\lceil \frac{l_j - \tau(m) + j - k + 1}{2} \right\rceil = \left\lceil \frac{2d + l_j + \tau(m) + j + k - 3}{2} \right\rceil$ $\leq \frac{2d + l_j + \tau(m) + j + k - 2}{2}$.

Using Lemma 5(i), we can write $\frac{b_{S_{path}}(w)}{b(w)} \leq \frac{2d + l_j + \tau(m) + j + k - 2}{2d + l_j + \tau(m) + j - 1} = 1 + \frac{k-1}{2d + l_j + \tau(m) + j - 1}$ $\leq 1 + \frac{k-1}{2d + l_j + \tau(m)}$ as $j \geq 1$.

Since $k \leq \tau(m) + 1 < l_k + 2 \leq l_j + 2$ and $d \geq 1$, $\frac{b_{S_{path}}(w)}{b(w)} \leq 1 + \frac{k-1}{(l_j+2) + (\tau(m)+1) - 1} <$ $1 + \frac{k-1}{2k-1} < 1.5$ for $k > 1$.

<u>*Case 2:*</u> $k - 1 < l_k + 1 \leq \tau(m)$:

Under algorithm S_{path}, u informs along P_j at time $d + j + 1$ where $1 \leq j \leq k - 1$ and $j \neq m$. v is informed at time $d + l_k + 1$ through P_k and P_j has $l_j - (l_k + 1 - j)$ uninformed vertices. Similarly, at time $d + l_k + 1$ path P_m will have $\tau(m) - (l_k + 1)$ uninformed vertices and $\tau(m) - (l_k + 1) < \tau(m) - (l_k + 1 - j)$ when $1 \leq j \leq k - 1$. So, for $1 \leq j \leq k - 1$ (including $j = m$) and $l_m = \tau(m)$, P_j has $l_j - (l_k + 1 - j)$ uninformed vertices at time $d + l_k + 1$. v can inform at most $k - 1$ neighboring vertices along $k - 1$ different paths $P_1, ..., P_{k-1}$. In the worst case, consider any path P_j that receives the message from v at time unit $d + l_k + 1 + k - 1$. The number of uninformed vertices in path P_j before time unit $d + l_k + 1 + k - 1$ will be $l_j - (l_k + 1 - j) - (k - 2)$ $= l_j - l_k + j - k + 1$. Since starting at time $d + l_k + 1 + k - 1$ onwards, P_j receives the message from both u and v, then $b_{S_{path}}(w) \leq d + l_k + 1 + k - 2 + \left\lceil \frac{l_j - l_k + j - k + 1}{2} \right\rceil$ $= \left\lceil \frac{2d + l_j + l_k + j + k - 1}{2} \right\rceil \leq \frac{2d + l_j + l_k + j + k}{2}$.

Using Lemma 6(i), we can write $\frac{b_{S_{path}}(w)}{b(w)} \leq \frac{2d + l_j + l_k + j + k}{l_j + 2d + l_k + j} = 1 + \frac{k}{l_j + 2d + l_k + j}$.

Note that the lower bound for Lemma 6(i) is true for all $1 \leq j \leq k - 1$. By picking $j = k - 1$ we get, $l_j + 2d + l_k + j = l_{k-1} + 2d + l_k + k - 1 > l_{k-1} + 2d + (k - 2) + (k - 1)$ (as $l_k > k - 2$) $\geq 2k$, since $d \geq 1$ and $l_{k-1} \geq 1$.

Hence, $\frac{b_{S_{path}}(w)}{b(w)} < 1 + \frac{k}{2k} = 1.5$. □

Theorem 2. *Algorithm S_{path} is a $(4 - \epsilon)$ approximation for any originator in the k-path graph G_k when $k > l_k + 1$, where $0 < \epsilon < 1$*

Proof. 1) <u>when originator is u</u>:

Under algorithm S_{path}, when v receives the message at time $l_k + 1$ there are no informed vertices in $k - l_k - 1$ paths. Recall that v can inform at most $k - 1$ neighboring vertices along $k-1$ different paths $P_1, ..., P_{k-1}$. In the worst case, consider any path P_j which has been informed from u at time unit k and assume that the same path has been informed from v at time unit $l_k + 1 + k - 1$. We calculate the time to inform all the vertices in path P_j. The number of uninformed vertices in P_j before time $l_k + 1 + k - 1$ will be $l_j - ((l_k + 1 + k - 1) - (k)) = l_j - l_k$. Since starting at time $l_k + 1 + k - 1$ onwards, it receives the message from both u and v, then $b_{S_{path}}(u) \leq l_k + 1 + k - 2 + \left\lceil \frac{l_j - l_k}{2} \right\rceil = \left\lceil \frac{l_j + l_k + 2k - 2}{2} \right\rceil \leq \frac{l_1 + l_k + 2k - 1}{2}$ as $l_1 \geq l_j$.

Using Lemma 3(iii), we can write $\frac{b_{S_{path}}(u)}{b(u)} \leq 2\frac{l_1 + l_k + 2k - 1}{2l_k + k + l_j + j + 2} = \frac{4k - 2 + 2l_1 + 2l_k}{2l_k + k + l_1 + 3}$ (as $j = 1$). So, $\frac{b_{S_{path}}(u)}{b(u)} \leq 3 + \frac{k - l_1 - 4l_k - 11}{2l_k + k + l_1 + 3}$.

It can be observed that, $(k - l_1 - 4l_k - 11) < (2l_k + k + l_1 + 3)$ and when $k > l_1 + 4l_k + 11$, $\frac{b_{S_{path}}(u)}{b(u)} < 4$.

2) <u>when originator is w on a path P_m</u>:

<u>*Case 1:*</u> $\tau(m) < l_k + 1$ and $\tau(m) < k - 1$:

When v receives the message at time $d + \tau(m)$, there are no informed vertices in $k - 1 - \tau(m)$ paths. Recall that v can inform at most $k - 1$ neighboring vertices along $k - 1$ different paths $P_1, ..., P_k$ except P_m. In the worst case, we consider any path P_j which has been informed from u at time unit $d + k - 1$ and assume that the same path has been informed from v at time unit $d + \tau(m) + k - 1$, where $1 \leq j \leq k$ and $j \neq m$. We calculate the time taken to inform all the vertices in path P_j. The number of uninformed vertices in P_j before time $d + \tau(m) + k - 1$ will be $l_j - ((d + \tau(m) + k - 1) - (d + k - 1)) = l_j - \tau(m)$. Since starting at time $d + \tau(m) + k - 1$ onwards, P_j receives the message from both u and v, then $b_{S_{path}}(w) \leq d + \tau(m) + k - 2 + \left\lceil \frac{l_j - \tau(m)}{2} \right\rceil = \left\lceil \frac{2d + l_j + \tau(m) + 2k - 4}{2} \right\rceil$. Now, if $P_m \neq P_1$ then $l_j \leq l_1$, else $l_j \leq l_2$. If we consider $l' \in \{l_1, l_2\}$ then $b_{S_{path}}(w) \leq \left\lceil \frac{2d + l' + \tau(m) + 2k - 4}{2} \right\rceil$ (as $l_j \leq l'$) $\leq \frac{2d + l' + \tau(m) + 2k - 3}{2}$.

Using Lemma 5(iii), we can write $\frac{b_{S_{path}}(w)}{b(w)} \leq 2\frac{2d + l' + \tau(m) + 2k - 3}{4d + 2\tau(m) + l_j + j + k - 2} \leq \frac{4k + 4d + 2l' + 2\tau(m) - 6}{4d + 2\tau(m) + l' + k - 1}$ (as $j \geq 1$ and $l_j = l'$) $= 3 + \frac{k - 8d - 4\tau(m) - l' - 3}{4d + 2\tau(m) + l' + k - 1}$

Again as $k - 8d - 4\tau(m) - l' - 3 < 4d + 2\tau(m) + l' + k - 1$ and when $k > 8d + 4\tau(m) + l' + 3$ we get, $\frac{b_{S_{path}}(w)}{b(w)} < 4$.

<u>*Case 2:*</u> $l_k + 1 \leq \tau(m)$ and $l_k + 1 < k$:

When v receives the message at time $d + l_k + 1$, there are no informed vertices in $k - l_k - 2$ paths. Recall that v can inform at most $k - 1$ neighboring vertices along $k - 1$ different paths $P_1, ..., P_{k-1}$. In the worst case, we consider any path P_j ($1 \le j \le k-1$ and $j \ne m$), which has been informed from u at time unit $d + k - 1$ and assume that the same path has been informed from v at time unit $d+l_k+1+k-1$. We calculate the time taken to inform all the vertices in path P_j. The number of uninformed vertices in P_j before time $d+l_k+1+k-1$ will be $l_j - ((d+l_k+1+k-1)-(d+k-1)) = l_j - l_k - 1$. Similarly the number of uninformed vertices in P_m before time $d + l_k + 1 + k - 1$ will be $\tau(m) - (k-2) - l_k - 1 < \tau(m) - l_k - 1$ as $k > 2$. So, for $1 \le j \le k-1$ (including $j = m$) and $l_m = \tau(m)$, P_j has at most $l_j - l_k - 1$ uninformed vertices. Since starting at time $d + l_k + 1 + k - 1$ onwards, P_j receives the message from both u and v, then $b_{S_{path}}(w) \le d + l_k + 1 + k - 2 + \left\lceil \frac{l_j - l_k - 1}{2} \right\rceil = \left\lceil \frac{2d + l_j + l_k + 2k - 3}{2} \right\rceil \le \frac{2d + l_j + l_k + 2k - 2}{2}$. Now, if $P_m \ne P_1$ then $l_j \le l_1$. If ($P_m = P_1$ and $\tau(m) \le l_2$) then $l_j \le l_2$ else if ($P_m = P_1$ and $\tau(m) > l_2$) then $l_j \le \tau(m)$. If we consider $l' \in \{l_1, l_2, \tau(m)\}$ then $b_{S_{path}}(w) \le \frac{2d + l' + l_k + 2k - 2}{2}$ as $l_j \le l'$.

Using Lemma 6(iii), we can write $\frac{b_{S_{path}}(w)}{b(w)} \le 2\frac{2d + l' + l_k + 2k - 2}{4d + 2l_k + l_j + j + k} \le \frac{4k + 4d + 2l_k + 2l' - 4}{4d + 2l_k + l' + k + 1}$ (as $j \ge 1$ and $l_j = l'$) $= 3 + \frac{k - 8d - 4l_k - l' - 7}{4d + 2l_k + l' + k + 1}$

Again as $k - 8d - 4l_k - l' - 7 < 4d + 2l_k + l' + k + 1$ and when $k > 8d + 4l_k + l' + 7$ we get, $\frac{b_{S_{path}}(w)}{b(w)} < 4$. □

From Theorem 1 and Theorem 2 we can conclude that in the worst case algorithm S_{path} gives $(4 - \epsilon)$ approximation for any originator and the worst case occurs for $k > l_1 + 4l_k + 8d + 7$. However, when $k \le l_1 + 4l_k + 8d + 7$, algorithm S_{path} gives 3-approximation for any originator. Moreover if k is smaller, then S_{path} generates even better approximation ratio. In particular, calculations similar to above show that if $k \le 6d + 3l_k + 2$, then S_{path} gives 2.75-approximation and when $k \le 2d + l_k + 3$, algorithm S_{path} generates 2-approximation for any originator.

3.2.1 Optimality of the approximation algorithm S_{path} for some subclasses of G_k when $k \le l_k + 1$

In this section we prove that our algorithm S_{path} generates the optimal broadcast time for some cases.

Theorem 3. *If $k \le l_k + 1$ and $l_j \ge l_{j+1} + 2$ for all $1 \le j \le k-1$, then algorithm*

S_{path} *generates the optimal broadcast time.*

Proof. 1) when originator is u:

Under scheme S_{path}, u informs along P_j at time $j+1$ for $1 \leq j \leq k-1$. Since $l_j \geq l_{j+1}+2$, when v gets informed at time l_k+1, the number of uninformed vertices, call it l'_j, in P_j are in the order $l'_j \geq l'_{j+1}+1$ for all $1 \leq j \leq k-2$ and $l'_{k-1} \geq k$. As a result all the vertices of path P_1 will be informed no sooner than the vertices of any other path in G_k. So in the worst case, we consider the time taken to inform all the vertices in P_1. When v receives the message at time unit l_k+1, P_1 has $l_1-(l_k+1-1)$ $= l_1 - l_k$ uninformed vertices. Since starting at time l_k+2 onwards P_1 receives the message from both u and v, so $b_{S_{path}}(u) \leq l_k + 1 + \lceil \frac{l_1-l_k}{2} \rceil = \lceil \frac{l_1+l_k+2}{2} \rceil$.

Using Lemma 3(i), $b(u) \geq \left\lceil \frac{l_j+l_k+j+1}{2} \right\rceil$ for any j. By picking $j=1$ we get $b(u) \geq \lceil \frac{l_1+l_k+2}{2} \rceil \geq b_{S_{path}}(u)$.

2) when originator is w on path P_m:

Case 1: $k-1 \leq \tau(m) < l_k+1$

Under scheme S_{path}, u informs along P_j at time $d+j$ for $1 \leq j \leq k$ and $j \neq m$. Since $l_j \geq l_{j+1}+2$, when v gets informed at time $d+\tau(m)$, the number of uninformed vertices, call it l'_j, in P'_j are in the order $l'_j \geq l'_{j+1}+1$ for all $1 \leq j \leq k-1$ and $j \neq m$. As a result all the vertices of path P'_j will be informed no sooner than the vertices of any other path in G_k. In the worst case, we consider the time taken to inform all the vertices in P'_j. If $P_m = P_1$, then $P'_j = P_2$ else $P'_j = P_1$. When v receives the message at time unit $d+\tau(m)$, P'_j has $l'_j - (d+\tau(m)-d) = l'_j - \tau(m)$ uninformed vertices. Starting at time $d+\tau(m)+1$ onwards, P'_j receives the message from both u and v, so $b_{S_{path}}(w) \leq d+\tau(m) + \left\lceil \frac{l'_j-\tau(m)}{2} \right\rceil = \left\lceil \frac{2d+l'_j+\tau(m)}{2} \right\rceil$. Depending on whether $P'_j = P_1$ or $P'_j = P_2$, either $l'_j \leq l_1$ or $l'_j \leq l_2$. If we consider $l' \in \{l_1, l_2\}$, then $b_{S_{path}}(w) \leq \left\lceil \frac{2d+l'+\tau(m)}{2} \right\rceil$ as $l'_j \leq l'$.

Using Lemma 5(i), we can write $b(w) \geq \left\lceil \frac{2d+l_j+\tau(m)+j-1}{2} \right\rceil \geq \left\lceil \frac{2d+l'+\tau(m)}{2} \right\rceil \geq b_{S_{path}}(w)$ as $l_j = l'$ and $j \geq 1$.

Case 2: $k-1 < l_k+1 \leq \tau(m)$

Under algorithm S_{path}, u informs along P_j at time $d+j+1$ for $1 \le j \le k-1$ and $j \ne m$. v is informed at time $d+l_k+1$ through P_k and P_j has $l_j-(l_k+1-j)$ uninformed vertices. Similarly at time $d+l_k+1$, path P_m will have $\tau(m)-(l_k+1)$ uninformed vertices and $\tau(m)-(l_k+1) < \tau(m)-(l_k+1-j)$ when $1 \le j \le k-1$. So, for $1 \le j \le k-1$ (including $j=m$) and $l_m=\tau(m)$, P_j has $l_j-(l_k+1-j)$ uninformed vertices at time $d+l_k+1$. Since $l_j \ge l_{j+1}+2$, at time $d+l_k+1$, the number of uninformed vertices, call it l'_j, in P'_j are in the order $l'_j \ge l'_{j+1}+1$ for all $1 \le j \le k-2$ (including $j=m$). As a result all the vertices of path P'_j will be informed no sooner than the vertices of any other path in G_k. In the worst case, we consider the time taken to inform all the vertices in P'_j. If $P_m \ne P_1$, then $P'_j = P_1$ else if ($P'_j = P_1$ and $\tau(m) \le l_2$) then $P'_j = P_2$ else if ($P'_j = P_1$ and $\tau(m) > l_2$) then $P'_j = P_m$. When v receives the message at time unit $d+l_k+1$, P'_j has $l'_j-(d+l_k+1-d-1) = l'_j - l_k$ uninformed vertices. Starting at time $d+l_k+2$ onwards, P'_j receives the message from both u and v, so $b_{S_{path}}(w) \le d+l_k+1+\left\lceil \frac{l'_j-l_k}{2} \right\rceil = \left\lceil \frac{2d+l'_j+l_k+2}{2} \right\rceil$. Depending on whether $P'_j = P_1$ or $P'_j = P_2$ or $P'_j = P_m$, either $l'_j \le l_1$ or $l'_j \le l_2$ or $l'_j \le \tau(m)$. If we consider $l' \in \{l_1, l_2, \tau(m)\}$, then $b_{S_{path}}(w) \le \left\lceil \frac{2d+l'+l_k+2}{2} \right\rceil$ as $l'_j \le l'$.

Using Lemma 6(i), we can write $b(w) \ge \left\lceil \frac{2d+l_j+l_k+j}{2} \right\rceil = \left\lceil \frac{2d+l'+l_k+2}{2} \right\rceil \ge b_{S_{path}}(w)$ as $l_j = l'$ and $j=2$. □

Theorem 4. *If $k \le l_k+1$ and $l_j = l_{j+1}$ for all $1 \le j \le k-1$, then algorithm S_{path} generates the optimal broadcast time.*

Proof. As a result, $l_j = l_k$.

1) <u>when originator is u</u>:

In scheme S_{path}, u informs along P_j at time $j+1$ for $1 \le j \le k-1$. When v gets informed at time l_k+1, number of uninformed vertices in P_j will be $l_j-(l_k+1-j) = l_k-(l_k+1-j) = j-1$. According to scheme S_{path}, v informs $P_{k-1}, P_{k-2}, ..., P_j, ..., P_1$ at times $l_k+2, l_k+3, ..., l_k+1+k-j, ..., l_k+k$ respectively. In general, P_j will have $j-1-(k-j-1) = 2j-k$ uninformed vertices before $l_k+1+k-j$ time units. Since starting at time $l_k+1+k-j$ onwards, P_j will now be informed from both u and v, then $b_{S_{path}}(u) \le l_k+k-j+\left\lceil \frac{2j-k}{2} \right\rceil \le \left\lceil \frac{2l_k+k}{2} \right\rceil$.

Now using Lemma 3(i), $b(u) \ge \left\lceil \frac{l_k+l_j+j+1}{2} \right\rceil \ge \left\lceil \frac{2l_k+k}{2} \right\rceil \ge b_{S_{path}}(u)$ (as $l_j \ge l_k$ and $j=k-1$).

2) when originator is w on path P_m:

In this case $k-1 < l_k+1 \leq \tau(m)$ is not possible because of the following reason: $l_j = l_k$ for all $1 \leq j \leq k-1$.
Recall that $\tau(m) = l_m + 2 - 2d = l_k + 2 - 2d \leq l_k$ as $d \geq 1$.
So $\tau(m) \geq l_k + 1$ is not possible. Thus, we assume that $\tau(m) < l_k + 1$.

In scheme S_{path}, u informs along P_j at time $d+j$ for $1 \leq j \leq k$ and $j \neq m$. When v gets informed at time $d+\tau(m)$, number of uninformed vertices in P_j will be $l_j-((d+\tau(m))-(d+j-1)) = l_k-(\tau(m)+1-j) = l_k-\tau(m)+j-1$. According to scheme S_{path}, v informs $P'_{k-1}, P'_{k-2}, ..., P'_j, ..., P'_1$ at times $d+\tau(m)+1, d+\tau(m)+2, ..., d+\tau(m)+k-j, ..., d+\tau(m)+k-1$ respectively where $P'_1, ..., P'_{k-1}$ is the combination of $P_1, ..., P_k$ except P_m. In general, P'_j will have $l_k-\tau(m)+j-1-(k-j-1) = l_k-\tau(m)+2j-k$ uninformed vertices before $d+\tau(m)+k-j$ time units. Since starting at time $d+\tau(m)+k-j$ onwards, P'_j will now be informed from both u and v, then $b_{S_{path}}(w) \leq d+\tau(m)+k-j-1+\left\lceil \frac{l_k-\tau(m)+2j-k}{2} \right\rceil = \left\lceil \frac{2d+l_k+\tau(m)+k-2}{2} \right\rceil$.

Using Lemma 5(i), we can write $b(w) \geq \left\lceil \frac{2d+l_j+\tau(m)+j-1}{2} \right\rceil = \left\lceil \frac{2d+l_k+\tau(m)+k-2}{2} \right\rceil \geq b_{S_{path}}(w)$ as $l_j = l_k$ and $j = k-1$. □

When $l_j = l_{j+1}+1$ for all $1 \leq j \leq k-1$, algorithm S_{path} does not generate the optimal broadcast time, however it gives better approximation than in general case.

Theorem 5. *If $k \leq l_k+1$ and $l_j = l_{j+1}+1$ for all $1 \leq j \leq k-1$, then algorithm S_{path} is a $(\frac{4}{3}-\epsilon)$-approximation for any originator in the k-path graph G_k, where $0 < \epsilon < 0.3$.*

Proof. As a result, $l_j = l_k + (k-j)$.
1) when originator is u:

In scheme S_{path}, u informs along P_j at time $j+1$ for $1 \leq j \leq k-1$. At time unit l_k+1, when v is informed, any path P_j will have $l_j-(l_k+1-j) = l_k+(k-j)-l_k-1+j = k-1$ uninformed vertices. Starting at time l_k+2 onwards v informs the path with maximum number of uninformed vertices. Thus, $b_{S_{path}}(u) = \max\ \{l_k+1+\lceil \frac{k-1}{2} \rceil$,

$l_k+1+1+\lceil\frac{k-1-1}{2}\rceil$,..., $l_k+1+(i-1)+\left\lceil\frac{k-1-(i-1)}{2}\right\rceil = l_k+\lceil\frac{k+i}{2}\rceil$,..., $l_k+1+k-2+\left\lceil\frac{k-1-(k-2)}{2}\right\rceil = l_k+k-1+\lceil\frac{1}{2}\rceil\} = l_k+k$.

Using Lemma 3(i), $\frac{b_{S_{path}}(u)}{b(u)} \leq \frac{2l_k+2k}{l_k+l_j+j+1} \leq \frac{2l_k+2k}{2l_k+k+1}$. (as $l_j = l_k+(k-j)$)

Therefore, $\frac{b_{S_{path}}(u)}{b(u)} \leq 1+\frac{k-1}{2l_k+k+1} \leq 1+\frac{k-1}{2(k-1)+k+1}$ (as $l_k \geq k-1$) $= 1+\frac{k-1}{3k-1} < 1+\frac{k-1}{3k-3} = 1+\frac{1}{3} = \frac{4}{3}$.

2) <u>when originator is w on path P_m</u>:

<u>Case 1:</u> $k-1 \leq \tau(m) < l_k+1$

When v gets informed at time $d+\tau(m)$, number of uninformed vertices in P_j will be $l_j-((d+\tau(m))-(d+j-1)) = (l_k+k-j)-(\tau(m)+1-j) = l_k-\tau(m)+k-1$. Starting at time $d+\tau(m)+1$ onwards v informs the path with maximum number of uninformed vertices. Thus, $b_{S_{path}}(w) = \max\{d+\tau(m)+\left\lceil\frac{l_k-\tau(m)+k-1}{2}\right\rceil, d+\tau(m)+1+\left\lceil\frac{l_k-\tau(m)+k-1-1}{2}\right\rceil,\ldots,d+\tau(m)+(i-1)+\left\lceil\frac{l_k-\tau(m)+k-1-(i-1)}{2}\right\rceil = \left\lceil\frac{2d+l_k+\tau(m)+k+i-2}{2}\right\rceil,\ldots,d+\tau(m)+k-2+\left\lceil\frac{l_k-\tau(m)+k-1-(k-2)}{2}\right\rceil = d+\tau(m)+k-2+\left\lceil\frac{l_k-\tau(m)+1}{2}\right\rceil\} = \left\lceil\frac{2d+l_k+\tau(m)+2k-3}{2}\right\rceil \leq \frac{2d+l_k+\tau(m)+2k-2}{2}$.

Using Lemma 5(i), $\frac{b_{S_{path}}(w)}{b(w)} \leq \frac{2d+l_k+\tau(m)+2k-2}{2d+l_j+\tau(m)+j-1} = \frac{2d+l_k+l_m+2-2d+2k-2}{2d+l_k+k-m+l_m+2-2d+m-1}$ (as $j=m$, $l_j = l_m = l_k+k-m$, $\tau(m) = l_m+2-2d$) $= \frac{l_k+l_m+2k}{l_k+l_m+k+1} = 1+\frac{k-1}{l_k+l_m+k+1} = 1+\frac{k-1}{2l_k+2k-m+1}$ (as $l_m = l_k+k-m$) $\leq 1+\frac{k-1}{4k-1-m}$ (since $l_k \geq k-1$) $\leq 1+\frac{k-1}{3k-1}$ (as $m \leq k$) $< 1+\frac{k-1}{3k-3} = 1+\frac{1}{3} = \frac{4}{3}$.

<u>Case 2:</u> $k-1 < l_k+1 \leq \tau(m)$

When v gets informed at time $d+l_k+1$ through P_k, P_j has $l_j-(l_k+1-j) = l_k+(k-j)$ - $(l_k+1-j) = k-1$ uninformed vertices for $1 \leq j \leq k-1$ and $j \neq m$. Similarly at time $d+l_k+1$, path P_m will have $\tau(m)-(l_k+1)$ uninformed vertices. Now, $\tau(m) = l_m+2-2d \leq l_m$ (as $d \geq 1$) $\leq l_1$ (as $l_m \leq l_1$) $= l_k+k-1$. Hence, $\tau(m)-(l_k+1) \leq l_k+k-1-l_k-1 = k-2$. Starting at time $d+l_k+2$ onwards v informs the path with maximum number of uninformed vertices. Thus, $b_{S_{path}}(w) = \max\{d+l_k+1+\lceil\frac{k-1}{2}\rceil, d+l_k+1+1+\lceil\frac{k-1-1}{2}\rceil,\ldots,d+l_k+1+(i-1)+\left\lceil\frac{k-1-(i-1)}{2}\right\rceil = \lceil\frac{2d+2l_k+k+i}{2}\rceil,\ldots,d+l_k+1+k-3+\left\lceil\frac{k-1-(k-3)}{2}\right\rceil = d+l_k+k-2+\lceil\frac{2}{2}\rceil, d+l_k+1+$

$k-2+\left\lceil\frac{k-2-(k-2)}{2}\right\rceil = d+l_k+k-1\} = d+l_k+k-1$.

Using Lemma 6(i), $\frac{b_{S_{path}}(w)}{b(w)} \leq \frac{2d+2l_k+2k-2}{2d+l_j+l_k+j} \leq \frac{2d+2l_k+2k-2}{2d+2l_k+k-1}$ (as $j=k-1$ and $l_j \geq l_k$) $= 1+\frac{k-1}{2d+2l_k+k-1} < 1+\frac{k-1}{3k-3}$ (as $l_k > k-2$ and $d \geq 1$) $= \frac{4}{3}$. □

Consider an instance of a k-path graph G_k where $k=10$ and the lengths of the paths are in the order $l_j = l_{j+1}+1$ for all $1 \leq j \leq 9$. Let us assume $l_{10} = 10$. Then the lengths of the paths $P_1, P_2, ..., P_9, P_{10}$ are respectively $19, 18, ..., 11, 10$. Similar to subcase 2, under algorithm S_{path}, $b_{S_{path}}(u) = k+l_k = k+l_{10} = 10+10 = 20$. However we can describe another algorithm A_{path} under which $b_{A_{path}}(u) = 18$.

Let us consider algorithm A_{path} for the same instance of G_k. Under scheme A, u informs $P_{10}, P_5, P_4, ..., P_1, P_9, ..., P_6$ at time units $1, 2, 3, ..., 6, 7, ..., 10$ respectively. When v gets informed at time 11, the number of uninformed vertices in the paths $P_1, ..., P_9$ forms an arithmetic series with difference 1 in some order starting from 5 upto 13. Starting at time 12 onwards v informs the path with maximum number of uninformed vertices till v does not have any adjacent uninformed vertex. Thus, $b_{A_{path}}(u) = \max\,\{11+\left\lceil\frac{13}{2}\right\rceil,\ 11+1+\left\lceil\frac{12-1}{2}\right\rceil,\ 11+2+\left\lceil\frac{11-2}{2}\right\rceil, ..., \ 11+5\} = 18$.

In general, let us assume the lengths of the paths $P_1, P_2, ..., P_{k-1}, P_k$ are respectively l_k+k-1, l_k+k-2,...,l_k+1, l_k. In order to find $b_{A_{path}}(u)$, we are going to consider two cases depending on whether k is even or odd.

i) When k is even:
Under scheme A_{path}, u informs the adjacent vertices in the paths P_k, $P_{\frac{k}{2}}$, $P_{\frac{k}{2}-1}$,...,P_1, P_{k-1},...,$P_{\frac{k}{2}+1}$ at time units $1, 2, 3, ..., \frac{k}{2}+1$, $\frac{k}{2}+2$,..., k respectively. When v gets informed at time l_k+1, the number of uninformed vertices in path P_j is $l_j-(l_k+1-(\frac{k}{2}-j+1))$ for $1 \leq j \leq \frac{k}{2}$. Similarly, when $\frac{k}{2}+1 \leq j \leq k-1$, the number of uninformed vertices in path P_j is $l_j-(l_k+1-(\frac{k}{2}+k-j))$. In other words, at time l_k+1, the number of uninformed vertices in the paths $P_1, ..., P_{k-1}$ forms an arithmetic series with difference 1 in some order starting from $\frac{k}{2}$ upto $\left\lceil\frac{3k}{2}\right\rceil-2$. Starting at time l_k+2 onwards v informs the path with maximum number of uninformed vertices. Thus, $b_{A_{path}}(u) = \max\,\{l_k+1+\left\lceil\frac{3k/2-2}{2}\right\rceil,\ l_k+1+1+\left\lceil\frac{(3k/2-3)-1}{2}\right\rceil, ..., \ l_k+1+i-1+\left\lceil\frac{(3k/2-(i+1))-(i-1)}{2}\right\rceil = l_k+i+\left\lceil\frac{3k-4i}{4}\right\rceil, ..., \ l_k+1+\frac{k}{2}\} = l_k+\left\lceil\frac{3k}{4}\right\rceil$.

ii) When k is odd:

Under scheme A_{path}, u informs along $P_k, P_{\frac{k-1}{2}}, P_{\frac{k-1}{2}-1}, ..., P_1, P_{k-1}, ..., P_{\frac{k-1}{2}+1}$ at time units $1, 2, 3, ..., \frac{k+1}{2}$, $\frac{k+1}{2}+1$,..., k respectively. When v gets informed at time l_k+1, the number of uninformed vertices in path P_j is $l_j - (l_k + 1 - (\frac{k+1}{2} - j))$ for $1 \leq j \leq \frac{k-1}{2}$. Similarly, when $\frac{k+1}{2} \leq j \leq k-1$, the number of uninformed vertices in path P_j is $l_j - (l_k + 1 - (\frac{k-1}{2} + k - j))$. In other words, at time $l_k + 1$, the number of uninformed vertices in the paths $P'_1, P'_2, P'_3, P'_4, ..., P'_{2i-1}, P'_{2i}, ..., P'_{k-2}, P'_{k-1}$ are $\frac{3k-5}{2}$, $\frac{3k-5}{2}$, $\frac{3k-9}{2}$, $\frac{3k-9}{2}$,..., $\frac{3k-(4i+1)}{2}$, $\frac{3k-(4i+1)}{2}$,..., $\frac{k+1}{2}$, $\frac{k+1}{2}$ respectively where $P'_1, P'_2, ..., P'_{k-1}$ is the permutation of the paths $P_1, ..., P_{k-1}$. Starting at time $l_k + 2$ onwards v informs the path with maximum number of uninformed vertices. Thus, $b_{A_{path}}(u) = \max\{l_k + 1 + \left\lceil \frac{(3k-5)/2}{2} \right\rceil, l_k + 1 + 1 + \left\lceil \frac{(3k-5)/2-1}{2} \right\rceil, ..., l_k + 1 + 2i - 2 + \left\lceil \frac{(3k-(4i+1))/2-(2i-2)}{2} \right\rceil = l_k + 2i - 1 + \left\lceil \frac{3k-8i+3}{4} \right\rceil, l_k + 1 + 2i - 1 + \left\lceil \frac{(3k-(4i+1))/2-(2i-1)}{2} \right\rceil = l_k + 2i + \left\lceil \frac{3k-8i+1}{4} \right\rceil, ..., l_k + 1 + \frac{k+1}{2}\} = l_k + \left\lceil \frac{3k+1}{4} \right\rceil = l_k + \left\lceil \frac{3k}{4} \right\rceil$ as k is odd.

Together, $b_{A_{path}}(u) = l_k + \left\lceil \frac{3k}{4} \right\rceil$

Using Lemma 3(i), $\frac{b_{A_{path}}(u)}{b(u)} \leq \frac{4l_k+3k}{2(l_k+l_j+j+1)} = \frac{4l_k+3k}{4l_k+2k+2}$. (as $l_j = l_k + (k-j)$)

Therefore, $\frac{b_{A_{path}}(u)}{b(u)} \leq 1 + \frac{k-2}{4l_k+2k+2} \leq 1 + \frac{k-2}{4(k-1)+2k+2}$ (as $l_k \geq k-1$) $= 1 + \frac{k-2}{6k-2} < 1 + \frac{k-2}{6k-12} = 1 + \frac{1}{6} = \frac{7}{6}$.

Observation: A_{path} algorithm gives better result as compared to S_{path} only for the subclass of the graph G_k where $l_j = l_{j+1} + 1$ and $k \leq l_k + 1$ since in this case, when v is informed, the number of uninformed vertices in the remaining $k-1$ paths forms an arithmetic series with difference either 1 or 2 depending on whether k is even or odd. However, for the broadcast problem in general k-path graph, algorithm A_{path} will not yield any better result. In any arbitrary k-path graph, if the longer path has at least 2 more vertices than its immediate shorter path, then under the scheme A_{path}, informing all the vertices in the longer path will be delayed since u always informs along the middle path first.

3.2.2 Summary of the Results:

Below is the summary of the results for algorithms S_{path} and A_{path}.

Table 1: Summary for k-path problem

Case	Algorithm	Result
General k-path	S_{path}	$(4-\epsilon)$-approximation
$l_j \geq l_{j+1}+2$ and $k \leq l_k+1$	S_{path}	optimal
$l_j = l_{j+1}$ and $k \leq l_k+1$	S_{path}	optimal
$l_j = l_{j+1}+1$ and $k \leq l_k+1$	S_{path}	$(\frac{4}{3}-\epsilon)$-approximation
$l_j = l_{j+1}+1$, $k \leq l_k+1$ and originator is u	A_{path}	$(\frac{7}{6}-\epsilon)$-approximation

Chapter 4

Constant Approximation for Broadcasting in k-cycle Graph

In this chapter we consider broadcasting in simple graphs where cycles intersect at single vertex. The simplest such graph where several cycles have only one intersecting vertex is called a k-cycle graph. A k-cycle graph is a collection of k cycles of arbitrary lengths connected by a central vertex on one end. Note that k-cycle graph is a cactus graph. Broadcasting in the k-cycle graph is different from broadcasting in the k-path graph in a way that in k-path graph, after a certain time, broadcasting depends on the strategy how the two intersecting vertices select the paths to send the message. However, in k-cycle graph, the entire broadcast scheme is dependent on the single central vertex. We present a constant approximation algorithm to find the broadcast time of an arbitrary k-cycle graph. Next we show the optimality of our algorithm for some subclasses of k-cycle graph. We also present another algorithm to generate the optimal broadcast time for a particular subclass of k-cycle graph.

4.1 Lower bounds on broadcast time

Definition 2. *Let $G_k = (V, E)$ be a connected graph consisting of k cycles C_1, C_2, C_3, ..., C_k and an intersecting vertex u connected on one end point of all cycles. Vertex u is called central vertex of G_k (see Figure 23).*

Let $l_1 \geq l_2 \geq ... \geq l_k \geq 2$, where l_i be the number of vertices in cycle C_i (excluding vertex u) for all $1 \leq i \leq k$.

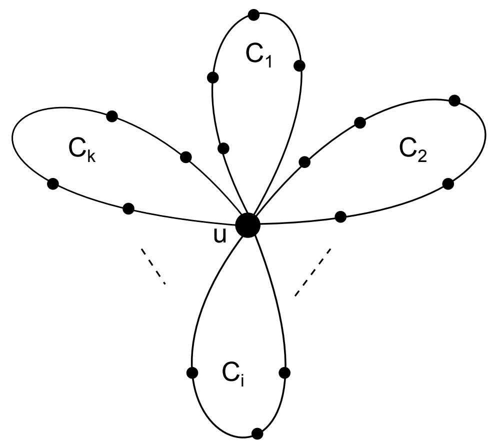

Figure 23: k-cycle graph

4.1.1 Lower bounds when originator is the central vertex

In this section we will give lower bounds on the broadcast time of G_k from originator u.

Lemma 7. *Let G_k be a k-cycle graph where the originator is the central vertex u and $l_1 \geq l_2 \geq ... \geq l_k \geq 2$. Then*
(i) $b(u) \geq k+1$. (ii) $b(u) \geq \left\lceil \frac{l_j+2j-1}{2} \right\rceil$ for any j, $1 \leq j \leq k$.
(iii) $b(u) \geq \left\lceil \frac{2k+l_j+2j+1}{4} \right\rceil$ for any j, $1 \leq j \leq k$.

Proof. (i): Under any minimum time broadcast scheme, k time units are necessary to inform at least one vertex in each of the k cycles from vetex u. Since $l_j \geq 2$ for any j, where $1 \leq j \leq k$, at least one more time unit is required to inform the second vertex on the cycle which initially receives the message from u at time unit k. So, $b(u) \geq k+1$.

(ii): We consider any cycle C_j where $1 \leq j \leq k$. Under any minimum time broadcast scheme all vertices in C_j must be informed. u informs the adjacent vertices of the k cycles in some order and assume it initially informs C_j at time unit j or later. Then u informs its second neighboring vertex in C_j no sooner than time unit $j+1$. At time unit j there are at least $l_j - 1$ uninformed vertices in C_j. Since, starting at time $j+1$ onwards, C_j receives the message from both directions from u, then at each time unit 2 new vertices on C_j will get informed. So, $b(u) \geq j + \left\lceil \frac{l_j-1}{2} \right\rceil =$

$\left\lceil \frac{l_j+2j-1}{2} \right\rceil$. Suppose, by contradiction u initially calls path C_j before time j. Then by pigeonhole principle there exists m, $1 \leq m \leq j-1$ such that u initially calls C_m at time j. Similarly at time unit j there are at least $l_m - 1$ uninformed vertices in C_m. If, starting at time $j+1$ onwards, C_m receives the message from both directions from u, then $b(u) \geq j + \lceil \frac{l_m-1}{2} \rceil = \lceil \frac{l_m+2j-1}{2} \rceil \geq \left\lceil \frac{l_j+2j-1}{2} \right\rceil$ as $l_m \geq l_j$. Hence, $b(u) \geq \left\lceil \frac{l_j+2j-1}{2} \right\rceil$.

For the proof of (iii), we combine the inequalities in (i) and (ii). We get $2b(u) \geq k+1 + \left\lceil \frac{l_j+2j-1}{2} \right\rceil \geq \left\lceil \frac{l_j+2j+2k+1}{2} \right\rceil$. Hence, $b(u) \geq \left\lceil \frac{l_j+2j+2k+1}{4} \right\rceil$ for any j, $1 \leq j \leq k$. □

Lemma 8. *Let G_k be a k-cycle graph where the originator is the central vertex u and n be the total number of vertices in G_k. Then*
(i) $b(u) \geq \lceil \frac{n-1}{2k} + k - \frac{1}{2} \rceil$ if $b(u) \geq 2k$.
(ii) $b(u) \geq \left\lceil \sqrt{(2n - \frac{7}{4})} - \frac{1}{2} \right\rceil$ if $k+1 \leq b(u) \leq 2k-1$.

Proof. (i): Since $b(u) \geq 2k$, then u will be busy informing its adjacent vertices in k different cycles at time units $1, 2, ..., 2k$. By $b(u)$ time units, u can inform at most $b(u)$, $b(u) - 1$,..., $b(u) - (2k-1)$ vertices in these k different cycles. So, $n \leq b(u) + b(u) - 1$ +...+ $b(u) - (2k-1) + 1 \Rightarrow n \leq 2kb(u) - k(2k-1) + 1$. Hence, $b(u) \geq \lceil \frac{n-1}{2k} + k - \frac{1}{2} \rceil$.

(ii): Since $k+1 \leq b(u) \leq 2k-1$, then u can inform its adjacent vertices in k different cycles at time units $1, 2, ..., b(u)$, where $b(u) \leq 2k-1$. By $b(u)$ time units, u can inform at most $b(u)$, $b(u) - 1$,..., 1 vertices in these k different cycles. So, $n \leq b(u) + b(u) - 1$ +...+ $1 + 1 \Rightarrow n \leq \frac{b(u)(b(u)+1)}{2} + 1 \Rightarrow b(u)^2 + b(u)$ - $(2n-2) \geq 0$.

Roots of $b(u)$ are $\frac{-1 \pm \sqrt{8n-7}}{2}$. Considering the positive root of $b(u)$, we get $b(u) \geq \left\lceil \sqrt{(2n - \frac{7}{4})} - \frac{1}{2} \right\rceil$. □

Let us now consider the originator in G_k to be any vertex w on a cycle C_j, for some $1 \leq j \leq k$. Let us assume the length of the shorter path from w to the central vertex u be d. Then the length of the longer path from w to $u = l_j + 1 - d$ and $d \leq l_j + 1 - d$ (see Figure 24).

Lemma 9. *There is a minimum time broadcast scheme from w in G_k in which w first sends the information along the shortest path towards vertex u.*

Proof. Let S_1 be a minimum broadcast scheme, $b_{S_1}(w) = b(w, G_k)$ under which w first informs its adjacent vertex along the longer path towards vertex u. We will

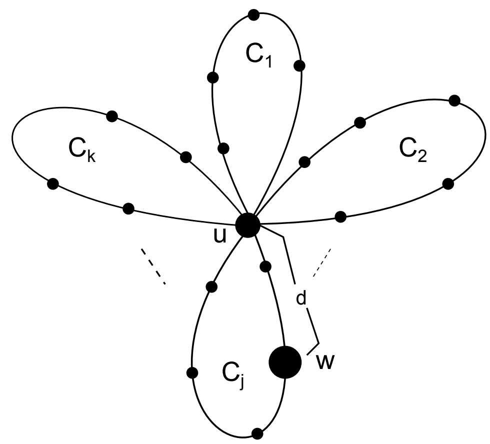

Figure 24: k-cycle graph with originator w

construct a new broadcast scheme S_2 under which w first sends information towards the shorter path. We will show that $b_{S_2}(w) \leq b_{S_1}(w) = b(w, G_k)$.

According to scheme S_1, w informs its adjacent vertex along the shorter path at time two. Now we construct a new broadcast scheme S_2 where w informs its adjacent vertex along the shorter path at time one. The order in which u broadcasts along the remaining $k-1$ cycles is the same in both schemes. However, under S_2, every vertex along the longer path towards vertex u from w will receive the message exactly one time unit later compared to S_1. To prove that $b_{S_2}(w) = b(w, G_k)$ we consider two cases:

Case 1: under S_1, u is informed along the shorter path at time $b_1 \leq b(w, G_k)$: Under S_2 all the vertices along the shorter path will be informed exactly one time unit earlier. So, u is informed at time $b_1 - 1$. u has exactly one free time unit immediately after $b_1 - 1$ to inform its adjacent vertex along the longer path towards w. Since the broadcast time in the remaining $k-1$ paths remains the same, $b_{S_2}(w) \leq b_{S_1}(w)$.

Case 2: under S_1, u is informed along the longer path from w:
Recall the length of the shorter path is d and the length of the longer path is $l_j + 1 - d$. Under S_1, u is informed along the longer path from w when either $d = l_j + 1 - d$ or $d + 1 = l_j + 1 - d$.

When $d = l_j + 1 - d$, it is quite trivial that $b_{S_2}(w) \leq b_{S_1}(w)$ since the broadcast time in the remaining $k-1$ paths remains the same.

When $d+1 = l_j + 1 - d$: Recall that under S_2 all the vertices along the shorter path will be informed exactly one time unit earlier. So u is informed at time unit d instead of time unit $l_j + 1 - d = d + 1$ under scheme S_1. u has exactly one free time unit immediately after d to inform its adjacent vertex along the longer path towards w. Since the broadcast time in the remaining $k-1$ paths remains the same, $b_{S_2}(w) \leq b_{S_1}(w)$. □

4.1.2 Lower bounds when originator is not the central vertex

In this section we will give lower bounds on the broadcast time of G_k from originator w.

Lemma 10. *Let G_k be a k-cycle graph where $l_1 \geq l_2 \geq ... \geq l_k \geq 2$ and the originator is any vertex w on a cycle C_m and the length of the shortest path from w to vertex u be d. Then*
(i) $b(w) \geq d + k$. (ii) $b(w) \geq d + \left\lceil \frac{l_j+2j-2}{2} \right\rceil$ for any j, $1 \leq j \leq k$.
(iii) $b(w) \geq d + \left\lceil \frac{2k+l_j+2j-2}{4} \right\rceil$ for any j, $1 \leq j \leq k$.

Proof. (i): By Lemma 9 there is a minimum time broadcast scheme from originator w in G_k in which w first sends the information along the shorter path towards vertex u. Considering this minimum broadcast scheme, u is informed no earlier than d time units. It takes another $k-1$ time units to inform at least one vertex in each of the remaining $k-1$ cycles from u. Recall that $l_j \geq 2$ for any j, where $1 \leq j \leq k$. So, at least one more time unit is required to inform the second vertex on the cycle which initially receives the message from u at time unit $d+k-1$. So, $b(w) \geq d+k$.

(ii): Similarly, at least d time units are necessary for u to receive the message from w. Now, we consider any cycle C_j where $1 \leq j \leq k$ and $j \neq m$. Under any minimum time broadcast scheme all vertices in C_j must be informed. u informs the remaining $k-1$ cycles in some order and assume it initially informs C_j at time unit $d+j$ or later. Then u informs C_j along the second branch no sooner than time unit $d+j+1$. At time unit $d+j$ there are at least $l_j - 1$ uninformed vertices in C_j. Similar to the argument given in Lemma 7(ii), we can write $b(w) \geq d + j + \left\lceil \frac{l_j-1}{2} \right\rceil = d + \left\lceil \frac{l_j+2j-1}{2} \right\rceil$ $\geq d + \left\lceil \frac{l_j+2j-2}{2} \right\rceil$.

When $j = m$, the number of uninformed vertices in C_m at time d, denoted as $\tau(m) = l_m - (2d-1)$. Considering $j = 1$ and $l_j = \tau(m)$ for the cycle C_m, we get

$b(w) \geq d + \left\lceil \frac{l_j+2j-2}{2} \right\rceil$ for any j, $1 \leq j \leq k$ included m.

For the proof of (iii), we combine the inequalities in (i) and (ii). We get $2b(w) \geq d + k + d + \left\lceil \frac{l_j+2j-2}{2} \right\rceil \geq 2d + \left\lceil \frac{l_j+2j+2k-2}{2} \right\rceil$. Hence, $b(w) \geq d + \left\lceil \frac{l_j+2j+2k-2}{4} \right\rceil$ for any j, $1 \leq j \leq k$. □

4.2 Approximation Algorithm

In this section we present broadcast algorithm S_{cycle} for graph G_k. We consider any vertex x to be the originator. When the originator is u then the algorithm S_{cycle} in G_k starts by informing the longest cycle C_1 in the first time unit.

When the originator is a non-central vertex w on cycle C_m then the scheme S_{cycle} in G_k starts by informing along the shorter path towards u. u is informed at time d. u informs the cycle with maximum number of uninformed vertices at time $d + 1$.

At time i, where $i \geq 1$ when $x = u$; else $i \geq d + 1$, consider the following 3 sets of cycles: a) The set X_0 consists of the cycles where there are no informed vertices. Let there be r such cycles arranged in non-increasing order of the number of uninformed vertices and let the cycle C_{10} has the maximum number of uninformed vertices of length l_{10}. b) The set X_1 consists of the cycles where at least one vertex has been informed along one branch from u. Let there be m such cycles arranged in non-increasing order of the number of uninformed vertices and let the cycle C_{11} has the maximum number of uninformed vertices of length l_{11}. c) The set X_2 consists of the cycles which have been informed from u along both directions. Depending on the lengths of l_{10} and l_{11}, u decides either to inform C_{10} or C_{11} at time $i + 1$. If there is no cycle in X_1 at time i, then u has no other choice but to inform C_{10} at time $i + 1$. If u informs C_{10}, then C_{10} becomes a member of the set X_1 from X_0. If u informs C_{11}, then C_{11} becomes a member of the set X_2 from X_1. Everytime a new cycle is being introduced in the set X_1, it is placed in non-increasing order of the number of uninformed vertices. Finally when there is no cycle in X_0 (at least one cycle will be present in X_1 at this moment), u broadcasts along the cycle having maximum number of uninformed vertices from the set X_1.

<u>Broadcast Algorithm S_{cycle}:</u>

INPUT: A k-cycle graph G_k where $l_1 \geq l_2 \geq ... \geq l_k \geq 2$ and any originator x.

OUTPUT: Broadcast time $b_{S_{cycle}}(x)$ and scheme of G_k.

BROADCAST-SCHEME-$S_{cycle}(G_k, l_1 \geq l_2 \geq ... \geq l_k \geq 2, x)$

0. Consider x as an originator.
1. when $x = u$, u broadcasts along C_1 at time unit 1.
2. when $x = w$, w first informs along the shorter path towards u.
   - 2.1. u is informed at time d.
   - 2.2. u informs the cycle with maximum number of uninformed vertices at time $d + 1$.
3. At time i, where $i \geq 1$ when $x = u$, else $i \geq d + 1$ consider the following 3 sets of cycles:
   - 3.1. X_0 : It consists of the cycles where there are no informed vertices. Let there are r cycles such that $l_{10} \geq l_{20} \geq ... \geq l_{r0}$, where l_{j0} is the length of the cycle C_{j0} in X_0 and $1 \leq j \leq r$. $C_{10}, C_{20}, ..., C_{r0}$ is the combination of r cycles from $C_1, ..., C_k$.
   - 3.2. X_1 : It consists of the cycles where at least one vertex has been informed along one branch from u. Let there are m cycles such that $l_{11} \geq l_{21} \geq ... \geq l_{m1}$, where l_{j1} is the number of uninformed vertices in the cycle C_{j1} in X_1 at time i and $1 \leq j \leq m$. $C_{11}, ...,$ C_{m1} is the combination of m cycles from $C_1, ..., C_k$ but not in set X_0.
   - 3.3. X_2 : It consists of the cycles which has been informed from u along both directions. Let there are p such cycles and $m + r + p = k$.
4. Starting at time $i + 1$ onwards until there is no cycle in X_0 do:
   - 4.1. If there is at least one cycle in X_1
     - 4.1.1. Select l_{10} and l_{11}
     - 4.1.2. If $l_{10} \geq l_{11} - 1$
       u broadcasts along C_{10} at time $i + 1$.
     - 4.1.3. Else-If $l_{10} < l_{11} - 1$
       u broadcasts along C_{11} at time $i + 1$.
   - 4.2. Else-If there is no cycle in X_1
     u broadcasts along C_{10} at time $i + 1$.
   - 4.3. If u informs C_{10}
     update $X_1 = X_1 + \{C_{10}\}$ and $X_0 = X_0 - \{C_{10}\}$.
   - 4.4. Else-If u informs C_{11}

update $X_2 = X_2 + \{C_{11}\}$ and $X_1 = X_1 - \{C_{11}\}$.

4.5. For every cycle in X_1 do

$l_{j1} = l_{j1} - 1$.

4.6. Arrange the cycles in X_1 in decreasing order of the number of uninformed vertices if u informs along C_{10}.

5. When there are cycles in X_1

u broadcasts along the cycle having maximum number of uninformed vertices.

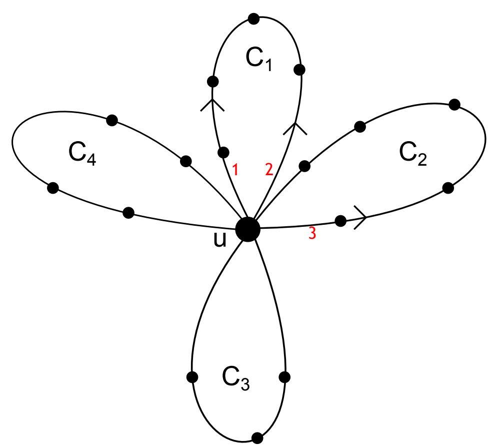

Figure 25: At time unit 3, under algorithm S_{cycle}: $X_0 = \{C_3, C_4\}$, $X_1 = \{C_2\}$, $X_2 = \{C_1\}$. Let at time unit 3, $l_{10} \geq l_{11} - 1$, where l_{10} and l_{11} are the number of uninformed vertices in C_3 and C_2 respectively. Then at time unit 4 in scheme S_{cycle}, u informs along C_3. Accordingly update $X_0 = \{C_4\}$, $X_1 = \{C_2, C_3\}$ and $X_2 = \{C_1\}$

Complexity Analysis:

Step 3.1 does not require any additional cost as the ordering of the cycles in the set X_0 remains the same from the beginning. The ordering of the cycles in step 3.2 is a direct consequence of the step 4.6. Since the elements in X_0 are already in the sorted order, inserting a new element in the decreasing order in step 4.6 will take $O(\log k)$ time. For k elements it can be accomplished in $O(k \log k)$ time. Steps 4.3 and 4.6 take $O(k)$ time to update the information in X_0 and X_1. Broadcasting done in steps 1, 2, 4.1, 4.2, 4.5 and 5 will take another $O(|V|)$ time to finish. Adding all the complexities we get that the complexity of the algorithm is $O(|V| + k \log k)$.

Theorem 6. *Algorithm S_{cycle} is a $(2-\epsilon)$-approximation for any originator in the k-cycle graph G_k.*

Proof. 1) when originator is u:

Under algorithm S_{cycle}, at any time unit u informs along the cycle either in X_0 or in X_1 depending on the lengths of the cycles C_{10} and C_{11} where C_{10} and C_{11} are the cycles from $C_1, ..., C_k$ having maximum number of uninformed vertices in X_0 and X_1 respectively. Assume that under scheme S_{cycle}, C_j is one of the cycles where broadcasting finishes at time unit $b_{S_{cycle}}(u)$. In scheme S_{cycle}, C_j has been informed from u at time $2j-1$ or sooner along its first branch. Let u informs its second adjacent vertex in C_j at time t_j, where $2j-1 < t_j \leq 2k$. At time $t_j - 1$ number of uninformed vertices in cycle C_j will be $l_j - (t_j - 2j + 1)$. Since starting at time t_j, C_j receives the message from both directions from u, then $b_{S_{cycle}}(u) = t_j - 1 + \left\lceil \frac{l_j - t_j + 2j - 1}{2} \right\rceil = \left\lceil \frac{l_j + t_j + 2j - 3}{2} \right\rceil \leq \left\lceil \frac{l_j + 2k + 2j - 3}{2} \right\rceil \leq \frac{l_j + 2k + 2j - 2}{2}$ as $t_j \leq 2k$.

Using Lemma 7(iii), we can write $\frac{b_{S_{cycle}}(u)}{b(u)} \leq 2\frac{l_j + 2k + 2j - 2}{l_j + 2j + 2k + 1} = 2 - \frac{6}{l_j + 2j + 2k + 1} < 2$.

2) when originator is w on cycle C_m:

Under algorithm S_{cycle}, w first sends the information along the shorter path towards vertex u. So u gets informed at time unit d. Consider the cycle C_j, where $1 \leq j \leq k$ and $j \neq m$. Similar to the proof when originator is u, we consider in the worst case any cycle C_j which finishes broadcasting in the last time unit in S_{cycle}. Similarly, u calls its first adjacent vertex in C_j at time $d + 2j - 1$ or sooner and informs its second adjacent vertex in C_j at time $d + t_j$, where $2j - 1 < t_j \leq 2k - 1$. The number of uninformed vertices in C_j before time $d + t_j$ will be $l_j - (t_j - 2j + 1)$. Now, let us consider the cycle C_m. Since starting at time two onwards w sends the information along the longer path towards vertex u, the number of informed vertices in C_m at time d will be $2d - 1$. Hence, the number of uninformed vertices in C_m before time $d + t_j$ will be $l_m - (2d - 1) - (t_j - 1) = l_m + 2 - 2d - t_j = \tau(m) - t_j + 1$ $< \tau(m) - (t_j - 2j + 1)$ since $j \geq 1$ and $\tau(m) = l_m + 1 - 2d$. So, for $1 \leq j \leq k$ (including $j = m$) and $l_m = \tau(m)$, C_j has at most $l_j - (t_j - 2j + 1)$ uninformed vertices. Since starting at time $d + t_j$ onwards, C_j receives the message from both directions, then $b_{S_{cycle}}(w) \leq d + t_j - 1 + \left\lceil \frac{l_j - t_j + 2j - 1}{2} \right\rceil = \left\lceil \frac{2d + l_j + t_j + 2j - 3}{2} \right\rceil \leq \left\lceil \frac{2d + l_j + 2k + 2j - 4}{2} \right\rceil$ $\leq \frac{2d + l_j + 2k + 2j - 3}{2}$ as $t_j \leq 2k - 1$.

Now, using Lemma 10(iii), we can write $\frac{b_{S_{cycle}}(w)}{b(w)} \leq 2\frac{2d + l_j + 2k + 2j - 3}{4d + l_j + 2j + 2k - 2} = 2 - \frac{4d + 2}{4d + l_j + 2j + 2k - 2}$ < 2. □

The above algorithm S_{cycle} is a $(2-\epsilon)$-approximation algorithm in general, but it generates the exact broadcast time for some subclasses of k-cycle graph.

4.3 Optimality of approximation algorithm S_{cycle} for some subclasses of G_k

In this section we consider several cases depending on the length of C_j and for some cases we will present an optimal algorithm.

Theorem 7. *If $l_j \geq l_{j+1} + 4$ for all $1 \leq j \leq k-1$, then algorithm S_{cycle} generates the optimal broadcast time in the k-cycle graph from originator u.*

Proof. In scheme S_{cycle}, u first informs one of its adjacent vertices along the cycle C_1 and C_1 is placed in the set X_1. Since $l_j \geq l_{j+1} + 4$, at time one, the number of uninformed vertices in C_1 is at least three more than the number of uninformed vertices in C_2 (C_2 is a cycle in X_0 having the maximum number of vertices among all cycles in X_0). So, according to scheme S_{cycle}, u informs along C_1 at time two. In other words, u informs the two adjacent vertices along cycle C_j at times $2j-1$ and $2j$ for $1 \leq j \leq k$. Since $l_j \geq l_{j+1} + 4$, when C_{j+1} gets informed from u at time $2(j+1)-1$, the number of uninformed vertices, call it l'_j in C_j are in the order $l'_j \geq l'_{j+1}$ for all $1 \leq j \leq k-1$. Starting at time $2(j+1)$ onwards, C_{j+1} also receives the message from both directions from u similar to C_j. As a result all the vertices in C_1 will be informed no sooner than the vertices of any other cycle in G_k. So in the worst case, we consider the time taken to inform all the vertices in C_1. In scheme S_{cycle}, starting at time 2 onwards, C_1 is informed from both directions from u. Thus, $b_{S_{cycle}}(u) \leq 1 + \lceil \frac{l_1-1}{2} \rceil = \lceil \frac{l_1+1}{2} \rceil$.

Using Lemma 7(ii), for $j = 1$ we get $b(u) \geq \lceil \frac{l_1+1}{2} \rceil \geq b_{S_{cycle}}(u)$. □

Theorem 8. *If $l_{j+1} + 4 \geq l_j \geq l_{j+1} + 3$ for all $1 \leq j \leq k-1$, then algorithm S_{cycle} is a 1.2-approximation in the k-cycle graph G_k from originator u, for $k \geq 3$.*

Proof. As a result $l_k + 4(k-j) \geq l_j \geq l_k + 3(k-j)$ and the total number of vertices in G_k, denoted as $n \geq l_k + (l_k+3) + \ldots + (l_k + 3(k-1)) + 1 = kl_k + \frac{3}{2}k(k-1) + 1$.

In scheme S_{cycle}, u first informs one of its adjacent vertices along the cycle C_1 and C_1 is placed in the set X_1. Since $l_1 \geq l_2 + 3$, at time one, the number of uninformed vertices in C_1 is at least two more than the number of uninformed vertices in C_2 (C_2 is a cycle in X_0 having the maximum number of vertices among all cycles in X_0). So, according to scheme S_{cycle}, u informs along C_1 at time two. In other words, u informs the two adjacent vertices along cycle C_j at times $2j-1$ and $2j$ for $1 \leq j \leq k$. At time $2j-1$, number of uninformed vertices in C_j will be $l_j - 1$. Now, $l_k + 4(k-j) \geq l_j \geq l_k + 3(k-j) \Rightarrow l_k + 4(k-j) - 1 \geq l_j - 1 \geq l_k + 3(k-j) - 1 > 0$ for $j \leq k$ and $l_k \geq 2$. As a result, all the cycles will receive the message twice from u. Since starting at time $2j$ onwards, C_j receives the message from both directions from u, then $b_{S_{cycle}}(u) \leq 2j - 1 + \left\lceil \frac{l_j - 1}{2} \right\rceil \leq 2j - 1 + \left\lceil \frac{l_k + 4k - 4j - 1}{2} \right\rceil = \left\lceil \frac{l_k + 4k - 3}{2} \right\rceil \leq \frac{l_k + 4k - 2}{2}$ as $l_k + 4(k-j) - 1 \geq l_j - 1$.

Now using Lemma 8(i), $b(u) \geq \left\lceil \frac{n-1}{2k} + k - \frac{1}{2} \right\rceil \geq \left\lceil \frac{kl_k + \frac{3}{2}k(k-1)}{2k} + k - \frac{1}{2} \right\rceil = \left\lceil \frac{2l_k + 7k - 5}{4} \right\rceil$.

Hence, $\frac{b_{S_{cycle}}(u)}{b(u)} \leq 2\frac{l_k + 4k - 2}{2l_k + 7k - 5} = \frac{2l_k + 8k - 4}{2l_k + 7k - 5} = 1 + \frac{k+1}{2l_k + 7k - 5} \leq 1 + \frac{k+1}{7k-1} = 1 + \frac{k+1}{5k+5+2k-6}$ $\leq 1 + \frac{k+1}{5k+5} = 1.2$, for $k \geq 3$ and $l_k \geq 2$. □

Observation: Note that algorithm S_{cycle} gives $\frac{7}{6}$-approximation for $k \geq 10$ for the case $l_{j+1} + 4 \geq l_j \geq l_{j+1} + 3$. Moreover if k is large enough then the approximation ratio of algorithm S_{cycle} approaches $\frac{8}{7}$.

Theorem 9. *If $l_j = l_{j+1} + 2$ for all $1 \leq j \leq k-1$, then algorithm S_{cycle} generates the optimal broadcast time in the k-cycle graph from originator u.*

Proof. As a result $l_j = l_k + 2(k-j)$ and total number of vertices in G_k, denoted as $n = l_k + (l_k + 2) + ... + (l_k + 2(k-1)) + 1 = kl_k + k(k-1) + 1$.

In scheme S_{cycle}, u first informs one of its adjacent vertices along the cycle C_1 and C_1 is placed in the set X_1. Since $l_1 = l_2 + 2$, at time one, the number of uninformed vertices in C_1 is one more than the number of uninformed vertices in C_2 (C_2 is a cycle in X_0 having the maximum number of vertices among all cycles in X_0). So, according to scheme S_{cycle}, u informs along C_2 at time two. In general, during the first k time units, u informs C_j at time j for $1 \leq j \leq k$. At time k, number of uninformed vertices in C_j will be $l_j - (k - (j-1)) = l_k + 2k - 2j - k + j - 1 = l_k + k - j - 1$. In other words, at time k, the number of uninformed vertices in the cycles $C_1, ..., C_k$ forms an arithmetic series with difference 1 starting from $l_k - 1$ up to $l_k + k - 2$. Starting at time $k+1$ onwards u informs the cycle with maximum number of uninformed vertices. Now we are going to consider two cases:

a) $k < l_k$: This ensures that all the cycles will get informed twice from u. So in general, u informs the second vertex in cycle C_i at time $k+i$ $(1 \leq i \leq k)$. Thus, $b_{S_{cycle}}(u) = \max\{k + \lceil \frac{l_k+k-2}{2} \rceil,\ k+1+\lceil \frac{l_k+k-3-1}{2} \rceil, ...,\ k+i-1+\left\lceil \frac{l_k+k-i-1-(i-1)}{2} \right\rceil = k + \lceil \frac{l_k+k-2}{2} \rceil, ...,\ k+k-1+\left\lceil \frac{l_k-1-(k-1)}{2} \right\rceil = k + \lceil \frac{l_k+k-2}{2} \rceil\} = \lceil \frac{l_k+3k-2}{2} \rceil$.

b) $k \geq l_k$: Since, at time k, the number of uninformed vertices in the cycles $C_1, ..., C_k$ forms an arithmetic series with difference 1 starting from $l_k - 1$ and $k \geq l_k$, some of the cycles will not receive the message from u twice. Assume there are p cycles $C'_1, ..., C'_p$ which will receive the information along its second branch from u starting at time $k+1$ onwards, where $C'_1, ..., C'_p$ is the combination of p cycles from $C_1, ..., C_k$. u finishes broadcasting all its adjacent vertices along these p cycles by time $k+p$. All the vertices in the remaining $k-p$ cycles must have been informed within $k+p$ time units. From the proof in part a) it is clear that the time taken to inform any of the p cycles will be $\lceil \frac{l_k+3k-2}{2} \rceil = k+p-1+\lceil \frac{l_k+k-2p}{2} \rceil$. Recall that at time k, number of uninformed vertices in C'_p is $l_k+k-p-1$. As a result, l_k+k-2p is the number of uninformed vertices in C'_p before time unit $k+p$. Since u informs along C'_p at time $k+p$, then $l_k+k-2p \geq 1$. Thus, $k+p-1+\lceil \frac{l_k+k-2p}{2} \rceil \geq k+p$. Hence, $b_{S_{cycle}}(u) \leq \lceil \frac{l_k+3k-2}{2} \rceil$ as in a).

Now using Lemma 8(i), $b(u) \geq \lceil \frac{n-1}{2k} + k - \frac{1}{2} \rceil = \left\lceil \frac{kl_k+k(k-1)}{2k} + k - \frac{1}{2} \right\rceil = \lceil \frac{l_k-2+3k}{2} \rceil$ $\geq b_{S_{cycle}}(u)$. □

Theorem 10. *If $l_j = l_{j+1}$ for all $1 \leq j \leq k-1$, then algorithm S_{cycle} generates the optimal broadcast time in the k-cycle graph G_k from originator u.*

Proof. Clearly $l_j = l_k$.

The order in which u initially informs the cycles is similar to the proof of Theorem 9. In scheme S_{cycle}, u first informs one of its adjacent vertices along the cycle C_1 and C_1 is placed in the set X_1. At time one, the number of uninformed vertices in C_1 is one less than the number of uninformed vertices in C_2 (C_2 is a cycle in X_0 having the maximum number of vertices among all cycles in X_0). So, u informs along C_2 at time two. In general, during the first k time units, u informs C_j at time j for $1 \leq j \leq k$. At time k, number of uninformed vertices in C_j will be $l_j - (k-(j-1))$ $= l_k - k + j - 1$. According to scheme S_{cycle}, starting from time $k+1$ onwards, u informs the adjacent uninformed vertices in the cycles C_k, C_{k-1},..., C_j,..., C_1 at times $k+1$, $k+2$,..., $2k+1-j$,..., $2k$ respectively. In general, C_j will have $l_k - k + j - 1 - (k-j) = l_k - 1 - 2(k-j) \geq l_k - (2k-1)$ uninformed vertices before $2k+1-j$ time units as $j \geq 1$. There are two cases to consider.

a) $l_k \geq 2k$: This guarantees that u has enough time to inform all its adjacent vertices in k cycles. Starting at time $2k+1-j$ onwards, C_j receives the message from both directions from u. Thus $b_{S_{cycle}}(u) \leq 2k - j + \lceil \frac{l_k-1-2k+2j}{2} \rceil = \lceil \frac{l_k-1+2k}{2} \rceil$.
b) $l_k < 2k$: As a result, some of the cycles will not receive the message from u twice. Let us assume there are $k-p+1$ such cycles $C'_k, ..., C'_p$ which will receive the information along its second branch from u starting at time $k+1$ onwards, where $C'_k, ..., C'_p$ is the combination of $k-p+1$ cycles from $C_1, ..., C_k$. u finishes broadcasting all its adjacent vertices along these $k-p+1$ cycles by time $2k+1-p$. All the vertices in the remaining $p-1$ cycles must have been informed within $2k+1-p$ time units. From the proof in part a) it is clear that the time taken to inform any of the $k-p+1$ cycles will be $\lceil \frac{l_k+2k-1}{2} \rceil = 2k - p + \lceil \frac{l_k-1-2k+2p}{2} \rceil$. Now, $l_k - 1 - 2k + 2p$ is the number of uninformed vertices in C'_p before time unit $2k+1-p$. Since u informs along C'_p at time $2k+1-p$, then $l_k - 1 - 2k + 2p \geq 1$. Thus, $2k - p + \lceil \frac{l_k-1-2k+2p}{2} \rceil \geq 2k+1-p$. Hence, $b_{S_{cycle}}(u) \leq \lceil \frac{l_k+2k-1}{2} \rceil$ as in a).

Now using Lemma 8(i), $b(u) \geq \lceil \frac{n-1}{2k} + k - \frac{1}{2} \rceil$ (n is the total number of vertices in G_k) $= \lceil \frac{kl_k}{2k} + k - \frac{1}{2} \rceil = \lceil \frac{l_k-1+2k}{2} \rceil \geq b_{S_{cycle}}(u)$ as $n - 1 = kl_k$. □

Theorem 11. *If $l_j \leq l_{j+1} + 1$ for all $1 \leq j \leq k-1$, then algorithm S_{cycle} is a $(1.5-\epsilon)$-approximation in the k-cycle graph G_k from originator u.*

Proof. As a result $l_j \leq l_k + k - j$ and total number of vertices in G_k, denoted as $n \leq l_k + (l_k+1) + ... + (l_k+k-1) + 1 = kl_k + \frac{k(k-1)}{2} + 1$.

In scheme S_{cycle}, u first informs one of its adjacent vertices along the cycle C_1 and C_1 is placed in the set X_1. Since $l_1 \leq l_2 + 1$, at time one, the number of uninformed vertices in C_1 is either exactly the same or one less than the number of uninformed vertices in C_2 (C_2 is a cycle in X_0 having the maximum number of vertices among all cycles in X_0). So, according to scheme S_{cycle}, u informs along C_2 at time two. In general, during the first k time units, u informs C_j at time j for $1 \leq j \leq k$. At time k, number of uninformed vertices in C_j will be $l_j - (k-(j-1)) \leq l_k + k - j - k + j - 1 = l_k - 1$. Starting from time $k+1$ onwards, u informs along the cycle having maximum number of uninformed vertices. Now, we are going to consider two cases:
a) $k < l_k$: This ensures that all the cycles receive the message twice from u. Thus, $b_{S_{cycle}}(u) \leq \max\{k + \lceil \frac{l_k-1}{2} \rceil, k+1+\lceil \frac{l_k-2}{2} \rceil, ..., k+(i-1)+\lceil \frac{l_k-1-i+1}{2} \rceil = k + \lceil \frac{l_k+i-2}{2} \rceil, ..., k+k-1+\left\lceil \frac{l_k-1-(k-1)}{2} \right\rceil\} = k + \lceil \frac{l_k+k-2}{2} \rceil \leq \frac{l_k+3k-1}{2}$
b) $k \geq l_k$: By time k, u has informed at least one vertex in each cycle and each cycle

has at most $l_k - 1$ uninformed vertices at time unit k. As a result, it will take at most another $l_k - 1$ time units to inform all the vertices in G_k. Thus, $b_{S_{cycle}}(u) \leq k + l_k - 1 = k + \frac{2l_k - 2}{2} < k + \frac{2l_k - 1}{2} < k + \frac{l_k + k - 1}{2} = \frac{l_k + 3k - 1}{2}$ as $k > l_k$.

Thus, for both cases, we get $b_{S_{cycle}}(u) \leq \frac{l_k + 3k - 1}{2}$.

Now using Lemma 7(ii) for $j = k$ we get, $\frac{b_{S_{cycle}}(u)}{b(u)} \leq \frac{l_k + 3k - 1}{l_k + 2k - 1} = 1 + \frac{k}{l_k + 2k - 1} \leq 1 + \frac{k}{2k+1} < 1.5$ as $l_k \geq 2$.

□

Note that 1.5-approximation ratio is achievable when $l_j = l_{j+1} + 1$ for $1 \leq j \leq k-1$. Next we present another algorithm which is optimal for $l_j = l_{j+1} + 1$, $1 \leq j \leq k - 1$

Broadcast Algorithm A_{cycle}:

1. u informs $C_{\lceil \frac{k}{2} \rceil}, C_{\lceil \frac{k}{2} \rceil - 1}, ..., C_1, C_k, ..., C_{\lceil \frac{k}{2} \rceil + 1}$ at time units $1, 2, ..., \lceil \frac{k}{2} \rceil$, $\lceil \frac{k}{2} \rceil + 1$,..., k respectively.
2. Calculate the number of uninformed vertices λ_j in C_j for $j = 1, 2, ..., k$ at time k.
3. Arrange λ_j in decreasing order such that $\lambda'_1 \geq \lambda'_2 \geq ... \geq \lambda'_{k-1}$ where $\lambda'_1, \lambda'_2, ..., \lambda'_{k-1}$ is the permutation of $\lambda_1, ..., \lambda_{k-1}$. If C'_j contains λ'_j uninformed vertices then,
4. For each time unit $i = 1$ to k
   - 4.1. If u has an uninformed adjacent vertex in C'_i
     - 4.1.1. u broadcasts along C'_i at time $k + i$

Complexity Analysis: Step 2 takes $O(k)$ time. Sorting in decreasing order in step 3 takes $O(k \log k)$ time. Broadcasting done in steps 1 and 4 can be accomplished in $O(|V|)$ time. Together, complexity is $O(|V| + k \log k)$.

Theorem 12. *If $l_j = l_{j+1} + 1$ for all $1 \leq j \leq k - 1$, then algorithm A_{cycle} generates the optimal broadcast time in the k-cycle graph from originator u.*

Proof. Similar to the proof in Theorem 11, total number of vertices in G_k, denoted as $n = kl_k + \frac{k(k-1)}{2} + 1$. We will consider two cases.
i) When k is odd: Under scheme A_{cycle}, u informs one of its two adjacent vertices of the cycles $C_{\frac{k+1}{2}}, C_{\frac{k+1}{2} - 1}, ..., C_1, C_k, ..., C_{\frac{k+1}{2} + 1}$ at time units $1, 2, ..., \frac{k+1}{2}$, $\frac{k+1}{2} + 1$,..., k respectively. At time k, the number of uninformed vertices in cycle C_j is $l_j -$

$(k - (\frac{k+1}{2} - j))$ for $1 \leq j \leq \frac{k+1}{2}$. Similarly, when $\frac{k+1}{2} + 1 \leq j \leq k$, the number of uninformed vertices in cycle C_j is $l_j - (j - \frac{k+1}{2})$. In other words, at time k, the number of uninformed vertices in the cycles $C_1, ..., C_k$ forms an arithmetic series with difference 1 in some order starting from $l_k - \frac{k+1}{2}$ up to $l_k + \frac{k+1}{2} - 2$. Starting at time $k+1$ onwards u informs the cycle with maximum number of uninformed vertices.

If $\frac{3k+1}{2} < l_k$, then all the cycles will get informed twice from u. Thus, $b_{A_{cycle}}(u) = \max\{k + \lceil \frac{2l_k+k-3}{4} \rceil, k+1+\lceil \frac{2l_k+k-7}{4} \rceil, ..., 2k-1+\left\lceil \frac{l_k-(k+1)/2-(k-1)}{2} \right\rceil\} = \lceil \frac{5k+2l_k-3}{4} \rceil$. If $\frac{3k+1}{2} \geq l_k$, then similar to Theorem 9.b), there are p cycles which receive the message along its second branch from u and it takes at most $\lceil \frac{5k+2l_k-3}{4} \rceil \geq k+p$ time units to complete broadcasting. Thus, for both cases we get, $b_{A_{cycle}}(u) = \lceil \frac{5k+2l_k-3}{4} \rceil$.

Now using Lemma 8(i), $b(u) \geq \lceil \frac{n-1}{2k} + k - \frac{1}{2} \rceil = \left\lceil \frac{kl_k+k(k-1)/2}{2k} + k - \frac{1}{2} \right\rceil = \lceil \frac{5k+2l_k-3}{4} \rceil \geq b_{A_{cycle}}(u)$.

ii) When k is even: Under scheme A_{cycle}, u informs along $C_{\frac{k}{2}}, C_{\frac{k}{2}-1}, ..., C_1, C_k, ..., C_{\frac{k}{2}+1}$ at time units $1, 2, ..., \frac{k}{2}$, $\frac{k}{2}+1$,..., k respectively. At time k, the number of uninformed vertices in cycle C_j is $l_j - (k - (\frac{k}{2} - j))$ for $1 \leq j \leq \frac{k}{2}$. Similarly, when $\frac{k}{2}+1 \leq j \leq k$, the number of uninformed vertices in cycle C_j is $l_j - (j - \frac{k}{2})$. In other words, at time k, the number of uninformed vertices in the cycles $C'_1, C'_2, C'_3, C'_4, ..., C'_{2i-1}, C'_{2i}, ..., C'_{k-1}, C'_k$ are $l_k + \frac{k}{2} - 2$, $l_k + \frac{k}{2} - 2$, $l_k + \frac{k}{2} - 4$, $l_k + \frac{k}{2} - 4$,..., $l_k + \frac{k}{2} - 2i$, $l_k + \frac{k}{2} - 2i$,..., $l_k - \frac{k}{2}$, $l_k - \frac{k}{2}$ respectively where $C'_1, C'_2, ..., C'_k$ is the permutation of the cycles $C_1, ..., C_k$. Starting at time $k+1$ onwards u informs the path with maximum number of uninformed vertices. Similar to case i), when $l_k \geq \frac{3k}{2}$, $b_{A_{cycle}}(u) = \max\{k + \left\lceil \frac{l_k+k/2-2}{2} \right\rceil$, $k+1+\left\lceil \frac{l_k+k/2-3}{2} \right\rceil$,..., $2k-1+\left\lceil \frac{l_k-k/2-(k-1)}{2} \right\rceil\} = \lceil \frac{5k+2l_k-2}{4} \rceil$. Similarly, when $l_k < \frac{3k}{2}$, for some p cycles which will be informed twice from u, $b_{A_{cycle}}(u) \leq \lceil \frac{5k+2l_k-2}{4} \rceil$.

Using Lemma 8(i) as in case i), $b(u) \geq \lceil \frac{n-1}{2k} + k - \frac{1}{2} \rceil = \lceil \frac{5k+2l_k-3}{4} \rceil = \lceil \frac{5k+2l_k-2}{4} \rceil \geq b_{A_{cycle}}(u)$ as k is even and so $5k + 2l_k - 3$ is always odd. □

4.3.1 Summary of the Results:

Below is the summary of the results for algorithms S_{cycle} and A_{cycle}.

Table 2: Summary for k-cycle problem

Case	Algorithm	Result
General k-cycle	S_{cycle}	$(2-\epsilon)$-approximation
$l_j \geq l_{j+1}+4$	S_{cycle}	optimal
$l_{j+1}+4 \geq l_j \geq l_{j+1}+3$	S_{cycle}	1.2-approximation for $k \geq 3$
		$\frac{7}{6}$-approximation for $k \geq 10$
$l_j = l_{j+1}+2$	S_{cycle}	optimal
$l_j = l_{j+1}$	S_{cycle}	optimal
$l_j \leq l_{j+1}+1$	S_{cycle}	$(1.5-\epsilon)$-approximation
$l_j = l_{j+1}+1$	A_{cycle}	optimal

Chapter 5

Broadcast Problem in Hypercube of Trees

In this chapter we continue the work in [103] and consider broadcasting in a hypercube graph where each vertex of the hypercube is the root of a tree, called hypercube of trees. Although there is a simple minimum time broadcast scheme for the hypercube, the problem is much more difficult for hypercube of trees because in a hypercube any pair of vertices are not neighbors as in a clique. However we were able to design a non-trivial algorithm to find the broadcast time of any originator for the hypercube of trees containing one tree. For the general case we present a linear time 2-approximation algorithm. We extend the result for any arbitrary graph whose nodes contain trees and design a linear time constant approximation algorithm.

5.1 Hypercube of Trees

Assume that we have a hypercube graph where every vertex is the root of a tree. We will call the resulting graph hypercube of trees.

Definition 3. *The hypercube of dimension k, denoted by H_k, is a simple graph with vertices representing 2^k binary strings of length k, $k \geq 1$ such that adjacent vertices have binary strings differing in exactly one bit position [113].*

Definition 4. *Consider 2^k trees $T_i = (V_i, E_i)$ rooted at r_i where $1 \leq i \leq 2^k$. We define the hypercube of trees, $HT_{k,n} = (V, E)$, to be a graph where $V = V_1 \cup V_2 \cup ... \cup V_{2^k}$ and $E = E_1 \cup E_2 \cup ... \cup E_{2^k} \cup E_{H_k}$ where $E_{H_k} = \{(r_i, r_j)|\ r_i,\ r_j$ are vertices of $H_k\}$.*

The roots of the trees, r_i, will be called root vertices and the rest of the vertices will be called tree vertices (see Figure 26).
$|V| = n \geq 2^k$ *and* $|E| = |V| - 2^k + k2^{k-1} = |V| + 2^{k-1}(k-2)$.

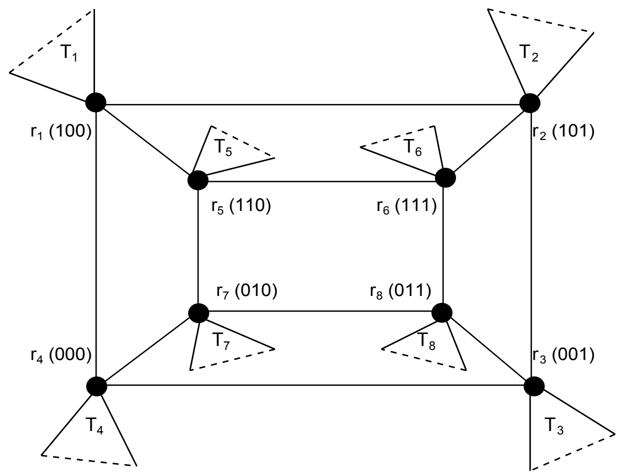

Figure 26: Hypercube of Trees $HT_{3,n}$ with 8 trees T_i rooted at r_i, $1 \leq i \leq 2^3$. Note that the roots r_i include a subgraph which is a hypercube H_3.

5.2 Broadcasting in Hypercube of Trees containing one tree

As mentioned above to find the broadcast time in hypercube of trees is difficult in general. In this section we design a linear algorithm to determine the broadcast time of $HT_{k,n}$ containing one tree.
Let G_1 be a $HT_{k,n}$ graph where r_0 is a root vertex and r_0 is the root of a tree T_0. The remaining $2^k - 1$ root vertices do not contain any tree. Let us also assume that r_0 has m neighbors in T_0, vertices $v_1, v_2, ..., v_m$. v_i is the root of the subtree T_i^0, $1 \leq i \leq m$. Let us consider $b(v_i, T_i^0) = t_i$ and without loss of generality we assume that $t_1 \geq t_2 \geq ... \geq t_m$. Then it follows from [162] that $b(r_0, T_0) = \max\{i + t_i\}$, where $1 \leq i \leq m$. Let $b(r_0, T_0) = \tau$ and $\tau \geq 1$ (see Figure 27).

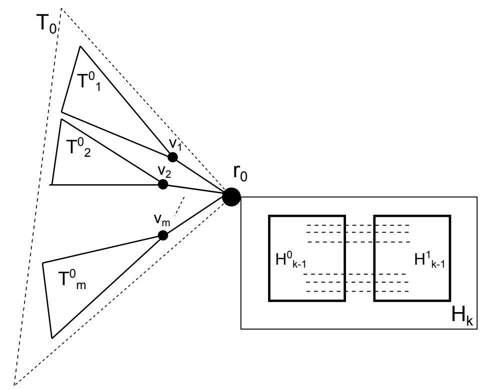

Figure 27: Hypercube of Trees $HT_{k,n}$ with only tree T_0 rooted at r_0

5.2.1 Broadcast Algorithm when originator is r_0

Consider two cases depending on the relationship between τ and the dimension of the hypercube in G_1. Let all the root vertices will be informed by $\tau(r)$ time units. The algorithm A calls another algorithm Broadcast-Hypercube which returns $\tau(r)$. When a tree vertex is informed there is not much it can do other than following the well known broadcast algorithm in trees [162], called A_T.

Tree Broadcast Algorithm A_T:

INPUT: originator r_i and tree rooted at r_i: T_i

OUTPUT: Broadcast time $b_{A_T}(r_i, T_i)$

TREE-BROADCAST(r_i, T_i)

1. r_i informs a child vertex in T_i that has the maximum broadcast time in the subtree rooted at it.
2. Let α_1,..., α_f be the broadcast times of the f subtrees rooted at r_i and $\alpha_1 \geq ... \geq \alpha_f$. Then, $b_{A_T}(r_i, T_i) = \max\{j + \alpha_j\}$ for $1 \leq j \leq f$.

Broadcast Algorithm A:

INPUT: $HT_{k,n} = (V, E)$, originator r_0, $b(r_0, T_0) = \tau$, m, $t_1 \geq t_2 \geq ... \geq t_m$

OUTPUT: Broadcast time $b_A(r_0)$ and broadcast scheme for $HT_{k,n}$

BROADCAST-SCHEME-A($HT_{k,n}$, r_0, τ, m, $t_1 \geq t_2 \geq ... \geq t_m$)

1. If $\tau \leq k$
   1.1. r_0 informs another root vertex r_1 in the first time unit.

1.2. $\tau(r)$ = BROADCAST-HYPERCUBE($HT_{k,n}$, r_1, 1).

1.3. For each time unit $i = 2$ to $m + 1$

1.3.1. r_0 informs tree vertex v_{i-1}.

2. If $\tau > k$

2.1. If $\tau \geq k + m$

2.1.1. For each time unit $i = 1$ to m

2.1.1.1. r_0 informs tree vertex v_i.

2.1.2. For each time unit $i = m + 1$ to $m + k$

2.1.2.1. an informed root vertex informs another uninformed root vertex using any shortest path.

2.2. If $k + m - 1 \geq \tau \geq k + 1$

Let j be the largest index such that $\tau = t_j + j$

2.2.1. For each time unit $i = 1$ to j

2.2.1.1. r_0 informs tree vertex v_i.

2.2.2. At time unit $j + 1$, r_0 informs another root vertex r_1.

2.2.3. $\tau(r)$ = BROADCAST-HYPERCUBE($HT_{k,n}$, r_1, $j + 1$).

2.2.4. For each time unit $i = j + 2$ to $m + 1$

2.2.4.1. r_0 informs tree vertex v_{i-1}.

2.2.5. If H_k is informed by time τ

then OUTPUT: $b_A(r_0)$

else FOLLOW steps 1.1 to 1.3

3. TREE-BROADCAST(v_i, T_i^0) for $1 \leq i \leq m$.

Broadcast-Hypercube:

INPUT: $HT_{k,n} = (V, E)$, originator r_1, time at which r_1 is informed: t_{r_1}

OUTPUT: $\tau(r)$

BROADCAST-HYPERCUBE($HT_{k,n}$, r_1, t_{r_1})

1. Assume r_1 is 10...0 (last $k - 1$ bits consist of zeroes)

2. For each time unit $i = t_{r_1} + 1$ to $t_{r_1} + k - 1$

2.1. For all $a_1, ..., a_{i-t_{r_1}-1} \in \{0, 1\}$ do in parallel

2.1.1. $1a_1...a_{i-t_{r_1}-1}00...0$ sends to $1a_1...a_{i-t_{r_1}-1}10...0$

3. For all $a_1, ..., a_{k-1} \in \{0, 1\}$ except r_1 do in parallel

3.1. $1a_1...a_{k-1}$ sends to $0a_1...a_{k-1}$

4. Return $t_{r_1} + k$

Complexity Analysis:

Broadcast-Hypercube takes $O(\log 2^k) = O(k)$ time to inform the root vertices. Algorithm A: Steps 1.1 and 1.3 take constant time to run. Step 2.1.2 can be completed in $O(k)$ time. Also steps 2.1.1 and 2.2 run in constant time. Again, the tree broadcast algorithm in step 3 takes $O(|V| - 2^k) = O(|V_T|)$ time to run, where $|V_T|$ is the number of tree vertices in G_1. Thus, complexity of algorithm is $O(|V_T| + k)$.

Proof of Correctness:

Theorem 13. *Algorithm A always generates the minimum broadcast time $b(r_0)$.*

Proof. Case 1: $m \leq \tau \leq k$
At least k time units are necessary to inform all the root vertices of G_1. Since r_0 is the root of the tree T_0, at least one more time unit is required to broadcast a tree vertex in T_0. So, $b(r_0) \geq k + 1$. Under algorithm A, the subroutine Broadcast-Hypercube informs the root vertices by time $k+1$. Since starting at time two onwards, r_0 informs the adjacent tree vertices in T_0, hence $b_A(r_0, T_0) = \tau + 1 \leq k + 1$ (as $\tau \leq k$). So, $b_A(r_0) = b(r_0) = k + 1$.
Case 2: $\tau \geq k + m$
At least τ time units are necessary to inform all the tree vertices of G_1. So, $b(r_0) \geq \tau$. Under algorithm A, r_0 first informs all the adjacent tree vertices. So all the vertices in T_0 will receive the message by time τ. Starting at time $m + 1$ onwards, r_0 informs the root vertices. Since it takes exactly k time units to inform all the root vertices, hence the root vertices will be informed by $m + k$ time units and $m + k \leq \tau$. So, $b_A(r_0) = b(r_0) = \tau$
Case 3: $k + m > \tau \geq k + 1$

It is always the case that $b(r_0) \geq \tau$. Let us assume that $b(r_0) = \tau$. In any minimum time broadcast scheme in hypercube H_k, every informed vertex cannot be idle during the time units $1, ..., k$ in order to complete broadcasting by time k. If originator u informs a vertex v at time one, and stays idle after time one, then v can finish broadcasting in H_k by $k + 1$ time units. Note that initially r_0 is the only informed vertex from which all other root vertices can receive the message. r_0 must make exactly m calls within T_0 sooner or later. So, it cannot make k calls within H_k, since $b(r_0) = \tau \leq k + m - 1$. Since $b(r_0) = \tau$ and $\tau = t_j + j = \max \{i + t_i\}$ for $1 \leq j \leq m$, then under any minimum time broadcast scheme, r_0 must call $v_1, ..., v_j$

at time units $1, ..., j$ within T. If r_0 cannot make k calls within H_k, then whether r_0 makes 1 call or $k-1$ calls within H_k, $b(r_0, H_k)$ will be the same and equal to $k+1$. The earliest time unit when r_0 can call one of its neighbors within H_k is the time unit $j+1$ or later. Thus, all the vertices in H_k can be informed no sooner than time $j+1+k$. Thus, $\tau = b(r_0) \geq b(r_0, H_k) \geq j+1+k$. We will show that algorithm A generates a $j+1+k$ time broadcast scheme in graph G_1.

Under algorithm A, r_1 receives the message at time $j+1$. The subroutine Broadcast-Hypercube informs the root vertices by time $j+1+k \leq \tau$. r_0 informs its adjacent tree vertices $v_1, ..., v_j, v_{j+1}, ..., v_m$ at time units $1, ..., j, j+2, ..., m+1$ respectively. As a result, $b_A(r_0, T_0) = \max\{t_1+1, ..., t_j+j, t_{j+1}+j+2, ..., t_m+m+1\} = \tau$ as $t_i + i < \tau \Rightarrow t_i + i + 1 \leq \tau$ for all $j+1 \leq i \leq m$. Thus $b_A(r_0) = b(r_0) = \tau$.

Note that $b(r_0) \leq \tau + 1$, since r_0 can call a neighbor in H_k at time 1 and then perform the minimum time broadcasting in T_0 starting time 2. Let us now assume that $b(r_0) = \tau + 1$. Under algorithm A, the subroutine Broadcast-Hypercube informs the root vertices by time $k+1 < \tau+1$ since $\tau > k$. Since starting at time two onwards, r_0 informs the adjacent tree vertices in T_0, hence $b_A(r_0, T_0) = \tau + 1$. So, $b_A(r_0) = b(r_0) = \tau + 1$. □

In the next section we will develop a broadcast algorithm for any originator in an arbitrary hypercube of trees, G_1. First we assume that the originator is any root vertex other than r_0. Finally we will discuss the broadcast algorithm in G_1 when the originator is any tree vertex.

5.2.2 Broadcasting from a root vertex other than r_0

In this section we present the broadcast algorithm A_r for graph G_1 when the originator is any root vertex (say r_j) other than r_0. Let us assume that r_j is at a distance d_r from vertex r_0, where $k \geq d_r \geq 1$. The algorithm A_r in G_1 starts by informing along the path $\overline{r_j r_0}$ (the shortest among all paths between r_j and r_0). r_0 receives the message at time d_r, and then it sends the message to the tree attached to it.

Broadcast Algorithm A_r:

INPUT: $HT_{k,n} = (V, E)$, originator r_j, $b(r_0, T_0) = \tau$

OUTPUT: Broadcast time $b_{A_r}(r_j)$ and broadcast scheme for $HT_{k,n}$

BROADCAST-SCHEME-A_r($HT_{k,n}$, r_j, τ)

1. r_j informs along the path $\overline{r_j r_0}$ (the shortest among all paths between r_j and r_0) in the first time unit.
2. r_j continues to inform the other root vertices using any shortest path.
r_0 receives the message at time d_r.
3. TREE-BROADCAST(r_0, T_0).

Complexity Analysis:
Steps 1 and 2 can be completed in $O(k)$ time. The tree broadcast algorithm in step 3 takes $O(|V_T|)$time to run. Complexity of algorithm is $O(|V_T| + k)$.

Proof of Correctness:

Theorem 14. *Algorithm A_r always generates the minimum broadcast time $b(r_j)$.*

Proof. Under algorithm A_r, r_0 receives the message at time d_r. Starting at time $d_r + 1$ onwards, r_0 informs the adjacent tree vertices. As a result all the vertices of T_0 will be informed by time $\tau + d_r$. Since r_0 does not play any role in informing a root vertex, it will take at most $k+1$ time units for all the root vertices in G_1 to receive the message.
Case 1: $\tau + d_r \leq k + 1$
Algorithm A_r in this case generates $b_{A_r}(r_j) = k + 1$.
Under any broadcast scheme, at least k time units are necessary to inform all the root vertices of G_1. Since r_0 is the root of the tree T_0, at least one more time unit is required to broadcast a tree vertex in T_0. So, $b(r_j) \geq k + 1$.
Case 2: $\tau + d_r > k + 1$
Algorithm A_r in this case generates $b_{A_r}(r_j) = \tau + d_r$.
Under any broadcast scheme, r_0 is informed no earlier than d_r time units. It takes another τ time units to inform all the tree vertices in T_0. So, $b(r_j) \geq \tau + d_r$. □

5.2.3 Broadcasting from a tree vertex

In this section we will develop a broadcast algorithm from any tree vertex in an arbitrary hypercube of trees G_1. Assume we are given a graph G_1 such that the originator v is in the subtree T_i^0 rooted at the root vertex r_0. There is a unique path P in T_i^0 connecting r_0 to the originator v. The vertex on the path P adjacent to r_0 is denoted by v_i. Let $u_1, u_2, ..., u_z$ be the z neighbors of v in the subtree. One of these vertices falls on the path P, call this vertex u_i. As shown in Figure 28, the graph G_1 can be restructured by drawing the tree T_i^0 rooted at the originator v and vertex v_i as one of its nodes. It can be observed that the remaining subgraph of G_1, denoted by G_1' is attached to T_i^0 by a bridge (v_i, r_0). Since the graph G_1' is connected to tree T_i^0 by

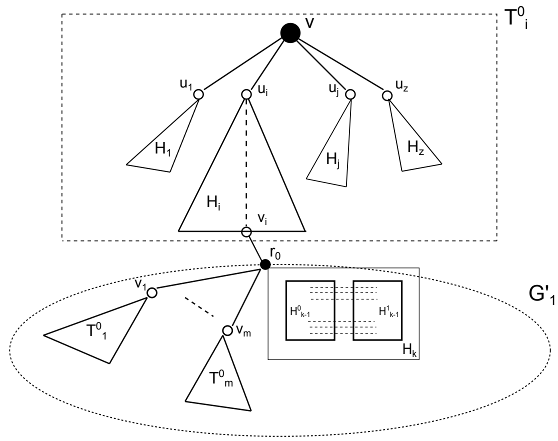

Figure 28: Hypercube of Trees G_1 with originator v. The subtree T_i^0 is separated from rest of the graph G_1'

a bridge, the broadcast algorithm in G_1' is independent of the broadcast algorithm in T_i^0. Once vertex r_0 is informed from v_i, it cannot inform any other vertex in T_i^0. It can only inform the vertices in G_1' in the minimum possible time unit. However, since r_0 is a root vertex and G_1' contains the hypercube H_k as its subgraph, broadcast in G_1' from r_0 can be considered as the broadcast problem in hypercube of trees with one tree rooted at r_0 where the originator is the root vertex r_0. We have a broadcast algorithm to solve this problem in G_1' from r_0. Let $T_{G_1'}$ be the broadcast tree of G_1' from r_0 obtained from algorithm A and τ_{m-1} be the broadcast time of r_0 in the remaining $m-1$ subtrees.

<u>Broadcast Algorithm A_v:</u>

INPUT: G_1', originator v in subtree T_i^0, r_0, τ_{m-1}, $m-1$, $t_1 \geq t_2 \geq ... \geq t_{m-1}$

OUTPUT: Broadcast time $b_{A_v}(v)$ and broadcast scheme for G_1

BROADCAST-SCHEME-A_v(G_1', v, T_i^0, τ_{m-1}, $m-1$, r_0, $t_1 \geq t_2 \geq ... \geq t_{m-1}$)

1. $T_{G_1'}$ = BROADCAST-SCHEME-A(G_1', r_0, τ_{m-1}, $m-1$, $t_1 \geq t_2 \geq ... \geq t_{m-1}$)
2. Attach $T_{G_1'}$ with T_i^0 by the bridge (r_0, v_i) and let the resulting tree be labelled as T_v.
3. TREE-BROADCAST(v, T_v).

Complexity Analysis:
Finding the broadcast time of a tree vertex in an arbitrary hypercube of trees with one tree is equivalent to solving two problems: (1) Finding the broadcast time of a root vertex in a hypercube of trees with one tree. As discussed before the complexity of this algorithm is linear. (2) Finding the broadcast time of a tree vertex in a tree. The complexity of this algorithm is also linear. Hence, the complexity is $O(|V|)$.

Proof of Correctness:

We can use the optimal broadcast tree of G'_1 obtained from algorithm A and attach it to the tree T_i^0 and solve the broadcast problem in the resulting tree. According to the broadcast algorithm in trees [162], v informs a child vertex that has the maximum broadcast time in the subtree rooted at it. The subtrees are labelled by H_j, where $1 \leq j \leq z$ (see Figure 28). The broadcast times $b(u_j, H_j)$ can be easily calculated except when $u_j = u_i$, since in this case G'_1 is attached to v_i. But we can solve the broadcast problem in G'_1 for the originator r_0 and obtain a broadcast tree $T_{G'_1}$. The weight of r_0 will then be initialized as the broadcast time in $T_{G'_1}$, call this $\tau_{G'}$. The optimal time required to inform all the vertices of H_i and G'_1 from u_i is equal to the broadcast time in the tree $H_i + (v_i, r_0)$ from u_i where weight$(r_0) = \tau_{G'}$.

5.3 Linear time 2-approximation algorithm in general hypercube of trees

In this section we will study the broadcast problem in general hypercube of trees.

5.3.1 Lower bound on the broadcast time

First we assume the originator is any root vertex.

Lemma 11. *Let G be a $HT_{k,n}$ where the originator r_0 is a root vertex. If $b(r_i, T_i)$ is the broadcast time of the root vertex r_i in the tree T_i where $0 \leq i \leq 2^k - 1$ then,*
i) $b(r_0) \geq max\{b(r_i, T_i)\}$ ii) $b(r_0) \geq k$.

Proof. For the proof of (i): Under any broadcast scheme, it takes at least maximum of $\{b(r_i, T_i)\}$ time units to inform all the vertices of G. Hence, $b(r_0) \geq max\{b(r_i, T_i)\}$.

Proof of (ii) goes as follows: At least k time units are necessary to inform all the root vertices of G. So, $b(r_0) \geq k$. □

Observation:

1. $b(r_0) = k$ when no trees are attached in $HT_{k,n}$.
2. Consider a graph $HT_{k,n}$ where only one tree is being attached at the originator r_0. The tree is a path P of length $l \geq k+1$. r_0 first informs along P and then informs the other root vertices. It is easy to see that the root vertices will be informed by $k+1$ time units. Similarly all the vertices in P will receive the message by $l \geq k+1$ time units. Thus, $b(r_0) = max\{b(r_i, T_i)\}$. Therefore, both lower bounds from Lemma 11 are achievable.

Let us now consider the originator is any tree vertex w in a tree T_i, where $0 \leq i \leq 2^k - 1$. Let us assume that w is at a distance d from the nearest root vertex r_0, where $d \geq 1$ (see Figure 29).

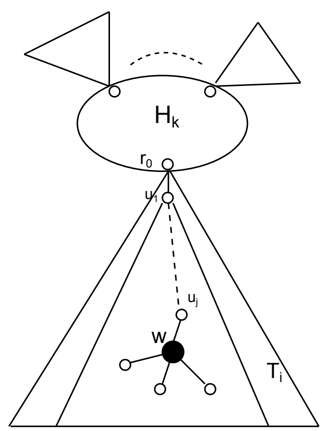

Figure 29: Hypercube of Trees G where the originator is a tree vertex w.

Lemma 12. *Let G be a $HT_{k,n}$ where the originator w is any tree vertex in a tree T_i and the length of the path $\overline{wr_0}$ is d, where $d \geq 1$. If $b(w, T_i)$ is the broadcast time of the tree T_i from w for $0 \leq i \leq 2^k - 1$ then,*
i) $b(w) \geq b(w, T_i)$ ii) $b(w) \geq max\{b(r_j, T_j)\}$ for all $j \neq i$ iii) $b(w) \geq k + d$.

Proof. For the proof of (i): w is any tree vertex in a tree T_i for $0 \leq i \leq 2^k - 1$ and it takes at least $b(w, T_i)$ time units to inform all the vertices of G. Hence, $b(w) \geq b(w, T_i)$.

Proof of (ii): For the remaining $2^k - 1$ trees T_j, where $0 \leq j \leq 2^k - 1$ & $j \neq i$, under any broadcast scheme, initially r_j is the only informed vertex from which the tree vertices can receive the message. Therefore at least $\max\{b(r_j, T_j)\}$ time units are necessary to broadcast in all the vertices. Hence, $b(w) \geq max\{b(r_j, T_j)\}$ for all $j \neq i$.

Proof of (iii): Under any broadcast scheme, r_0 is informed no earlier than d time units. It takes at least another k time units to inform all the root vertices of G. So, $b(w) \geq k + d$. □

5.3.2 Approximation Algorithm

In this section we present the broadcast algorithm S_{hyper} for graph G. We consider any vertex x to be the originator. When the originator is r_0 then the algorithm S_{hyper} in G starts by informing all the vertices of the hypercube. When all the root vertices are informed, each vertex informs the tree attached to it.

When the originator is w then the algorithm S_{hyper} in G starts by informing along the path $\overline{wr_0}$. r_0 receives the message at time d. During the next k time units all the vertices of the hypercube are being informed. Each root vertex will now send the message to the tree attached to it.

Approximation Algorithm S_{hyper}:

INPUT: $HT_{k,n} = (V, E)$ and any originator x

OUTPUT: Broadcast time $b_{S_{hyper}}(x)$ and broadcast scheme for $HT_{k,n}$

BROADCAST-SCHEME-$S_{hyper}(HT_{k,n}, x)$

1. If $x = w$
   - 1.1. w broadcasts along the shortest path $\overline{wr_0}$ in the first time unit. r_0 gets informed at time d.
   - 1.2. For each time unit $i = d + 1$ to $d + k$
     - 1.2.1. an informed root vertex informs another uninformed root vertex using any shortest path.
2. If $x = r_0$
   - 2.1. For each time unit $i = 1$ to k

2.1.1. an informed root vertex informs another uninformed root vertex using any shortest path.

3. TREE-BROADCAST(r_i, T_i) for $0 \leq i \leq 2^k - 1$.

Complexity Analysis:

Steps 1.2 and 2.1 take $O(k)$ time to inform the root vertices. In steps 1.1 and 3, the tree broadcast algorithm takes $O(|V_T|)$ time to run. Complexity of algorithm is $O(|V_T| + k)$.

Theorem 15. *Algorithm S_{hyper} is a 2-approximation for any originator in the graph $HT_{k,n}$*

Proof. When originator is r_0: Considering algorithm S_{hyper}, an upper bound on broadcast time can be obtained when the tree T_i with broadcast time max $\{b(r_i, T_i)\}$ is being attached at the root vertex r_i which is at distance $\log 2^k = k$ from originator r_0, where $0 \leq i \leq 2^k - 1$. By time k all the root vertices will be informed. Each root vertex r_i will take $b(r_i, T_i)$ time units to broadcast in T_i. As a result all the tree vertices will be informed by time $\max\{b(r_i, T_i)\}$. Thus, $b_{S_{hyper}}(r_0) \leq k + \max\{b(r_i, T_i)\}$, where $0 \leq i \leq 2^k - 1$. Combining Lemma 11(i) and Lemma 11(ii) we can write $b(r_0) \geq \frac{1}{2}(\max\{b(r_i, T_i)\} + k)$. Hence, $\frac{b_{S_{hyper}}(r_0)}{b(r_0)} \leq 2\frac{max\{b(r_i,T_i)\}+k}{max\{b(r_i,T_i)\}+k} = 2$.

When originator is w: w is any tree vertex in T_i. Let it is at a distance d from the nearest root vertex r_0. Considering algorithm S_{hyper}, an upper bound on broadcast time can be obtained when the tree T_j with broadcast time max $\{b(r_j, T_j)\}$ is being attached at the root vertex r_j which is at distance k from r_0, where $0 \leq j \leq 2^k - 1$ & $i \neq j$. r_0 receives the message at time d. By time $d + k$ all the root vertices will be informed. Each root vertex r_j will take $b(r_j, T_j)$ time units to broadcast in T_j.

Let $u_1, u_2, .., u_q$ be the q neighbors of w in T_i. One of these vertices falls on the path $\overline{wr_0}$, call this vertex u_r. u_j is the root of the subtree T_i^j, $1 \leq j \leq q$ and $b(u_j, T_i^j) = b_j$. If $b_1 \geq b_2 \geq ... \geq b_q$, then it follows from [162] that $b(w, T_i) = \max\{j + b_j\}$, where $1 \leq i \leq q$. Under algorithm S_{hyper}, w informs u_r,u_1,...,u_{r-1},u_{r+1},u_q at time units $1, 2, ..., r, r + 1, q$ respectively. If $b(w, T_i) = j + b_j$ for any $1 \leq j \leq r - 1$, then $b_{S_{hyper}}(w, T_i) \leq b(w, T_i) + 1$.

Let $u_{r_1}, u_{r_2}, .., u_{r_p}$ be the p neighbors of u_r in T_i^r and let u_{r_r} be the vertex that falls on the path $\overline{wr_0}$. Similarly if b_{r_j}, where $1 \leq j \leq p$ be the broadcast times of p subtrees rooted at u_{r_j} and $b_{r_1} \geq ... \geq b_{r_p}$, then $b(u_r, T_i^r) = \max\{j + b_{r_j}\}$. Under algorithm S_{hyper}, u_r informs u_{r_r}, u_{r_1},...,$u_{r_{r-1}}$,$u_{r_{r+1}}$,...,u_{r_p} at time units $1, 2, ..., r, r + 1, ..., p$

respectively (time units are considered after u_r is informed). If $b(u_r, T_i^r) = j + b_{r_j}$ for any $1 \leq j \leq r-1$, then $b_{S_{hyper}}(u_r, T_i^r) \leq b(u_r, T_i^r) + 1$. Thus, $b_{S_{hyper}}(w, T_i^r)$ $\leq 1 + b(u_r, T_i^r) + 1 = b(w, T_i^r) + 1$. Since the path $\overline{wr_0}$ has been given the priority in algorithm S_{hyper}, similarly in the worst case, at every level (upto d levels) the broadcast time of the subtrees will be delayed by one time unit. Therefore, $b_{S_{hyper}}(w, T_i) \leq b(w, T_i) + d$. As a result all the tree vertices in G will be informed by time $\max\{b(w, T_i) + d, k + d + max\{b(r_j, T_j)\}\}$.
If $b(w, T_i) + d \geq \max\{b(r_j, T_j)\} + k + d$, then $b_{S_{hyper}}(w) \leq b(w, T_i) + d$.
Combining Lemma 12(i) and Lemma 12(iii) we can write $b(w) \geq \frac{1}{2}(b(w, T_i) + k + d)$.
Hence, $\frac{b_{S_{hyper}}(w)}{b(w)} \leq 2\frac{b(w,T_i)+d}{b(w,T_i)+k+d} < 2$ as $k \geq 1$.
If $b(w, T_i) + d < \max\{b(r_j, T_j)\} + k + d$, then $b_{S_{hyper}}(w) \leq k + d +$ $\max\{b(r_j, T_j)\}$.
Combining Lemma 12(ii) and Lemma 12(iii) we can write $b(w) \geq \frac{1}{2}(\max\{b(r_j, T_j)\} + k + d)$.
Hence, $\frac{b_{S_{hyper}}(w)}{b(w)} \leq 2\frac{max\{b(r_j,T_j)\}+k+d}{max\{b(r_j,T_j)\}+k+d} = 2$. □

5.4 Linear time constant approximation algorithm in an arbitrary graph whose nodes contain trees

Assume that we have an arbitrary graph H where its vertices are the roots of the trees. We call the resulting graph arbitrary graph of trees G. In this section we design a linear time constant approximation algorithm to determine the broadcast time of G where the broadcast scheme in H is already known.

5.4.1 Lower bound on the broadcast time

Let H contains m vertices. We call the vertices of H as root vertices which contain trees; the rest are non-root vertices.

First we assume the originator is any root vertex.

Lemma 13. *Let G be an arbitrary graph of trees where the originator r_0 is a root vertex. If $b(r_i, T_i)$ is the broadcast time of the root vertex r_i in the tree T_i, where $1 \leq i \leq m$ then,*
i) $b(r_0) \geq max\ \{b(r_i, T_i), 1 \leq i \leq m\}$ *ii)* $b(r_0) \geq b(r_0, H)$

Proof. (i): Under any broadcast scheme, it takes at least maximum of $b(r_i, T_i)$ time units to inform all the vertices in G. Hence, $b(r_0) \geq \max \{b(r_i, T_i), 1 \leq i \leq m\}$.

(ii): At least $b(r_0, H)$ time units are necessary to inform all the vertices in H from r_0. So, $b(r_0) \geq b(r_0, H)$. □

Next we consider the originator is any non-root vertex r_n. Let us assume that r_n is at a distance d_1 from the nearest root vertex r_0, where $d_1 \geq 1$.

Lemma 14. *Let G be an arbitrary graph of trees where the originator r_n is a non-root vertex and the length of the path $\overline{r_n r_0}$ is d_1, where $d_1 \geq 1$. If $b(r_i, T_i)$ is the broadcast time of the root vertex r_i in the tree T_i, where $1 \leq i \leq m$ then,*
i) $b(r_n) \geq d_1 +$ max $\{b(r_i, T_i), 1 \leq i \leq m\}$ ii) $b(r_n) \geq b(r_n, H)$

Proof. (i): Under any broadcast scheme, r_0 is informed no earlier than d_1 time units. It takes another maximum of $b(r_i, T_i)$ time units to inform all the vertices in G. Hence, $b(r_n) \geq d_1 + \max \{b(r_i, T_i), 1 \leq i \leq m\}$.

(ii): At least $b(r_n, H)$ time units are necessary to inform all the vertices in H from r_n. So, $b(r_n) \geq b(r_n, H)$. □

Let us now consider the originator is any tree vertex w in a tree T_i, where $1 \leq i \leq m$. Let us assume that w is at a distance d_2 from the nearest root vertex r_0, where $d_2 \geq 1$.

Lemma 15. *Let G be an arbitrary graph of trees where the originator w is a tree vertex and the length of the path $\overline{w r_0}$ is d_2, where $d_2 \geq 1$. If $b(w, T_i)$ is the broadcast time of the tree T_i from w for $1 \leq i \leq m$ then,*
i) $b(w) \geq \{b(w, T_i), 1 \leq i \leq m\}$ ii) $b(w) \geq$ max $\{b(r_j, T_j)\}$ for all $j \neq i$ iii) $b(w) \geq d_2 + b(r_0, H)$

Proof. (i): w is any tree vertex in a tree T_i for $1 \leq i \leq m$. It takes at least $b(w, T_i)$ time units to inform all the vertices of G. Hence, $b(w) \geq \{b(w, T_i), 1 \leq i \leq m\}$.

(ii): For the remaining trees T_j, where $1 \leq j \leq m$ and $j \neq i$, under any broadcast scheme, initially the root vertex r_j is the only informed vertex from which the rest tree vertices can receive the message. Therefore at least maximum of $\{b(r_j, T_j)\}$ time units are necessary to broadcast in all the vertices. Hence, $b(w) \geq \max \{b(r_j, T_j)\}$.

(iii): Under any broadcast scheme, r_0 is informed no earlier than d_2 time units. It takes at least another $b(r_0, H)$ time units to inform all the vertices in H from r_0. So, $b(w) \geq d_2 + b(r_0, H)$. □

5.4.2 Approximation Algorithm

In this section we present the broadcast algorithm S_{Arbi} for graph G. We consider any vertex x to be the originator. When the originator is r_0 then the algorithm S_{Arbi} in G starts by informing all the vertices of H. When all the vertices in H are informed, each root vertex informs the tree attached to it.

When the originators are w and r_n then the algorithm S_{Arbi} in G starts by informing along the paths $\overline{wr_0}$ and $\overline{r_nr_0}$ respectively. When r_0 receives the message, the scheme informs all the vertices of H. Each root vertex will now send the message to the tree attached to it.

Approximation Algorithm S_{Arbi}:

INPUT: $G = (V, E)$ and any originator x

OUTPUT: Broadcast time $b_{S_{Arbi}}(x)$ and broadcast scheme for G

BROADCAST-SCHEME-$S_{Arbi}(G, x)$

1. If $x = w$
   - 1.1. w broadcasts along the shortest path $\overline{wr_0}$ in the first time unit. r_0 gets informed at time d_2.
   - 1.2. Starting at time $d_2 + 1$ onwards inform all the vertices in H.
2. If $x = r_n$
   - 2.1. r_n broadcasts along the shortest path $\overline{r_nr_0}$ in the first time unit. r_0 gets informed at time d_1.
   - 2.2. Starting at time $d_1 + 1$ onwards inform all the vertices in H.
3. If $x = r_0$
   - 3.1. Inform all the vertices in H.
4. TREE-BROADCAST(r_i, T_i) for $1 \leq i \leq m$.

Complexity Analysis:

Steps 1.2, 2 and 3 take $O(m)$ time to inform the vertices in H. In steps 1.1 and 4, the tree broadcast algorithm takes $O(|V| - m) = O(|V_T|)$ time to run. Complexity of algorithm is $O(|V_T| + m)$.

Theorem 16. *Let us assume there is a minimum time broadcast scheme in H from any originator. Then, algorithm S_{Arbi} is a 2-approximation for any originator in the graph G.*

Proof. When originator is r_0: Considering algorithm S_{Arbi}, all the vertices in H will be informed by $b(r_0, H)$ time units from r_0, since there is a minimum time broadcast scheme in H from any originator. Each root vertex r_i will take $b(r_i, T_i)$ time units to broadcast in T_i. As a result all the tree vertices will be informed by time $\max\{b(r_i, T_i)\}$. Thus, $b_{S_{Arbi}}(r_0) \leq b(r_0, H)+ \max\{b(r_i, T_i)\}$, where $1 \leq i \leq m$. Combining Lemma 13(i) and Lemma 13(ii) we can write $b(r_0) \geq \frac{1}{2}(b(r_0, H)+ \max\{b(r_i, T_i)\})$. Hence, $\frac{b_{S_{Arbi}}(r_0)}{b(r_0)} \leq 2\frac{b(r_0,H)+max\{b(r_i,T_i)\}}{b(r_0,H)+max\{b(r_i,T_i)\}} = 2$.

When originator is r_n: r_n is any non-root vertex in H and is at a distance d_1 from the nearest root vertex r_0. Considering algorithm S_{Arbi}, all the vertices in H will be informed by $d_1 + b(r_n, H)$ time units from r_n. Each root vertex r_i will take $b(r_i, T_i)$ time units to broadcast in T_i. Similarly, $b_{S_{Arbi}}(r_n) \leq d_1 + b(r_n, H)+ \max\{b(r_i, T_i)\}$, where $1 \leq i \leq m$. Combining Lemma 14(i) and Lemma 14(ii) we can write $b(r_n) \geq \frac{1}{2}(d_1 + b(r_n, H)+ \max\{b(r_i, T_i)\})$. Hence, $\frac{b_{S_{Arbi}}(r_n)}{b(r_n)} \leq 2\frac{d_1+b(r_n,H)+max\{b(r_i,T_i)\}}{d_1+b(r_n,H)+max\{b(r_i,T_i)\}} = 2$.

When originator is w: w is any tree vertex in T_i and is at a distance d_2 from the nearest root vertex r_0. The proof is exactly similar to the proof in Theorem 15 for the case when originator is w. As a result, under algorithm S_{Arbi}, all the tree vertices in G will be informed by time $\max\{b(w, T_i) + d_2,\ b(r_0, H) + d_2 + \max\{b(r_j, T_j)\}\}$.

If $b(w, T_i) + d_2 \geq b(r_0, H) + d_2 + \max\{b(r_j, T_j)\}$, then $b_{S_{Arbi}}(w) \leq b(w, T_i) + d_2$. Combining Lemma 15(i) and Lemma 15(iii) we can write $b(w) \geq \frac{1}{2}(b(r_0, H) + d_2 + b(w, T_i))$. Hence, $\frac{b_{S_{Arbi}}(w)}{b(w)} \leq 2\frac{d_2+b(w,T_i)}{b(r_0,H)+d_2+b(w,T_i)} < 2$.

If $b(w, T_i) + d_2 < b(r_0, H) + d_2 + \max\{b(r_j, T_j)\}$, then $b_{S_{Arbi}}(w) \leq b(r_0, H) + d_2 + \max\{b(r_j, T_j)\}$. Combining Lemma 15(ii) and Lemma 15(iii) we can write $b(w) \geq \frac{1}{2}(d_2 + b(r_0, H)+ \max\{b(r_j, T_j)\})$. Hence, $\frac{b_{S_{Arbi}}(w)}{b(w)} \leq 2\frac{d_2+b(r_0,H)+max\{b(r_j,T_j)\}}{d_2+b(r_0,H)+max\{b(r_j,T_j)\}} = 2$. □

Theorem 17. *Let us assume there is a c-approximation algorithm for the broadcast time problem in H from any originator, where c is a constant and $c > 1$. Then,*
(i) Algorithm S_{Arbi} is a ($2c - \epsilon$)-approximation for any originator in the graph G
(ii) Algorithm S_{Arbi} is a (1+c)-approximation for any originator x in the graph H when $b(x, H) \leq max\{b(r_i, T_i)\}$, for $1 \leq i \leq m$

Proof. I) When originator is r_0 : Considering algorithm S_{Arbi}, all the vertices in H will be informed by $cb(r_0, H)$ time units from r_0, since there is a c-approximation algorithm for the broadcast time problem in H from any originator. Similar to the proof of Theorem 16 for the case when originator is r_0, we can write $b_{S_{Arbi}}(r_0) \leq cb(r_0, H)+ \max\{b(r_i, T_i)\}$, where $1 \leq i \leq m$. Hence, $\frac{b_{S_{Arbi}}(r_0)}{b(r_0)} \leq 2\frac{cb(r_0,H)+max\{b(r_i,T_i)\}}{b(r_0,H)+max\{b(r_i,T_i)\}}$ $= 2c - \frac{(2c-2)max\{b(r_i,T_i)\}}{b(r_0,H)+max\{b(r_i,T_i)\}} < 2c$ for $c > 1$.

$b(r_0, H) \leq max\{b(r_i, T_i)\}$:

$\frac{b_{S_{Arbi}}(r_0)}{b(r_0)} \leq 2\frac{cb(r_0,H)+max\{b(r_i,T_i)\}}{b(r_0,H)+max\{b(r_i,T_i)\}} = 2 + \frac{2(c-1)b(r_0,H)}{b(r_0,H)+max\{b(r_i,T_i)\}} \leq 2 + \frac{2(c-1)b(r_0,H)}{2b(r_0,H)} = 1 + c$ since $b(r_0, H) \leq \max\{b(r_i, T_i)\}$.

II) When originator is r_n : Similar to the proof in Theorem 16 for the case when the originator is r_n and since all the vertices in H will be informed by $cb(r_n, H)$ time units from r_n under scheme S_{Arbi}, $b_{S_{Arbi}}(r_n) \leq d_1 + cb(r_n, H)+ \max\{b(r_i, T_i)\}$, where $1 \leq i \leq m$. Hence, $\frac{b_{S_{Arbi}}(r_n)}{b(r_n)} \leq 2\frac{d_1+cb(r_n,H)+max\{b(r_i,T_i)\}}{d_1+b(r_n,H)+max\{b(r_i,T_i)\}} = 2c - \frac{(2c-2)(d_1+max\{b(r_i,T_i)\})}{d_1+b(r_n,H)+max\{b(r_i,T_i)\}}$ $< 2c$ for $c > 1$.

$b(r_n, H) \leq max\{b(r_i, T_i)\}$:

$\frac{b_{S_{Arbi}}(r_n)}{b(r_n)} \leq 2\frac{d_1+cb(r_n,H)+max\{b(r_i,T_i)\}}{d_1+b(r_n,H)+max\{b(r_i,T_i)\}} = 2 + \frac{2(c-1)b(r_n,H)}{d_1+b(r_n,H)+max\{b(r_i,T_i)\}} \leq 2 + \frac{2(c-1)b(r_n,H)}{1+2b(r_n,H)}$ since $b(r_n, H) \leq \max\{b(r_i, T_i)\}$ and $d_1 \geq 1$. Thus, $\frac{b_{S_{Arbi}}(r_n)}{b(r_n)} < 2 + \frac{2(c-1)b(r_n,H)}{2b(r_n,H)} = 1 + c$.

III) When originator is w : Similar to the proof in Theorem 16 for the case when the originator is w and since under scheme S_{Arbi}, all the vertices in H will be informed by $cb(r_0, H)$ time units from r_0, all the tree vertices in G will be informed by time $\max\{b(w, T_i) + d_2,\ cb(r_0, H) + d_2+\max\{b(r_j, T_j)\}\}$.

Similarly, if $b(w, T_i) + d_2 \geq cb(r_0, H) + d_2+\max\{b(r_j, T_j)\}$, then $\frac{b_{S_{Arbi}}(w)}{b(w)} < 2$.

If $b(w, T_i) + d_2 < cb(r_0, H) + d_2+\max\{b(r_j, T_j)\}$, then $b_{S_{Arbi}}(w) \leq cb(r_0, H) + d_2+ \max\{b(r_j, T_j)\}$. Similarly, $\frac{b_{S_{Arbi}}(w)}{b(w)} \leq 2\frac{d_2+cb(r_0,H)+max\{b(r_j,T_j)\}}{d_2+b(r_0,H)+max\{b(r_j,T_j)\}} = 2c - \frac{(2c-2)(d_2+max\{b(r_j,T_j)\})}{d_2+b(r_0,H)+max\{b(r_j,T_j)\}}$ $< 2c$ for $c > 1$.

$b(r_0, H) \leq max\{b(r_j, T_j)\}$:

Similar to the proof above for the case when the originator is r_n and since $d_2 \geq 1$, $\frac{b_{S_{Arbi}}(w)}{b(w)} \leq 2 + \frac{2(c-1)b(r_0,H)}{d_2+b(r_0,H)+max\{b(r_j,T_j)\}} < 2 + \frac{2(c-1)b(r_0,H)}{2b(r_0,H)} = 1 + c$. □

Chapter 6

Broadcasting in Harary-like Graphs

The topology in distributed computing plays a central role in determining the performance of the system ([113], [120]). The two main constraints on designing a good topology are cost and reliability. We try to minimize the number of edges, which reduces the cost of network. At the same time, the connectivity of the topology should ensure the network is reliable. Frank Harary in [83] introduced the Harary Graph, $H_{k,n}$ which generates the minimal k-connected graph on n vertices. In this chapter we consider broadcasting in Harary graph. We present a $\log \frac{k-2}{2} + 1$-additive approximation to find the broadcast time in an arbitrary Harary graph. In the next section for even values of n, we introduce a modified-Harary graph and present a 1-additive approximation algorithm to find the broadcast time. We show the optimality of our algorithm for a particular subclass of modified-Harary graph. Then we also show that modified-Harary graph is a broadcast graph when k is logarithmic of n.

6.1 Diameter of Harary Graph and Lower bound on Broadcast Time

Frank Harary in [83] first defined the Harary graph.

Definition 1. The Harary graph $H_{k,n}$ is defined as follows:

Case 1: k is even:

Let $k = 2r$. $H_{2r,n}$ is constructed as follows: Given two positive integers n and $2r$

with $2r \leq n$, begin by drawing an n-gon and label its points $0, 1, ..., n-1$. Join two vertices i and j if and only if $|i-j| \equiv m \bmod n$, where $1 \leq m \leq 2r$ (see Figure 30).
Case 2: k is odd and n is even:

Let $k = 2r+1$. $H_{2r+1,n}$ is constructed by first drawing $H_{2r,n}$ and then adding edges by joining vertex i to vertex $i + \frac{n}{2}$ for $0 \leq i \leq \frac{n}{2} - 1$ (see Figure 31).
Case 3: both k and n are odd:

Let $k = 2r+1$. $H_{2r+1,n}$ is constructed by first drawing $H_{2r,n}$ and then adding edges by joining vertex 0 to vertices $\frac{n-1}{2}$ and $\frac{n+1}{2}$ and vertex i to vertex $i + \frac{n+1}{2}$ for $0 < i < \frac{n-1}{2}$ (see Figure 32). Here, only vertex 0 has degree $k+1$.

In all the 3 cases, n is sufficiently larger than k.

Definition 2. The diameter of a graph G, denoted as $D(G)$ is the greatest distance between any pair of vertices.

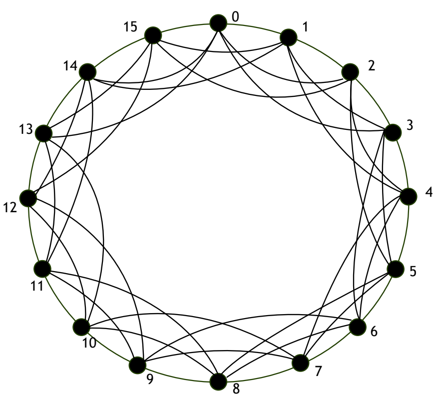

Figure 30: $H_{6,16}$ where k is even

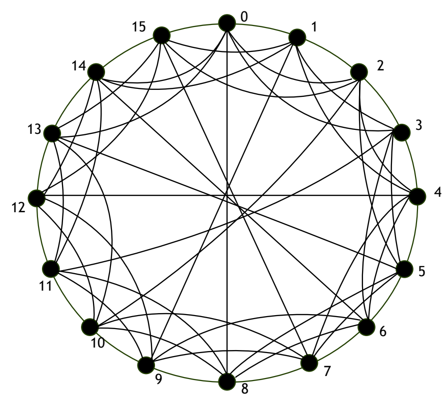

Figure 31: $H_{7,16}$ where k is odd and n is even

Properties of Harary Graph:

- When n or k is even, $H_{k,n}$ is a circulant graph [161].
- It is vertex transitive except for the case when both k and n are odd.
- For every vertex i, where $i = 0, ..., n-1$, there are two cliques in $H_{k,n}$. The first clique is formed with the set of vertices $V_1 = \{i, (i+1) \bmod n, ..., (i + \lfloor \frac{k}{2} \rfloor) \bmod n\}$ and the second clique contains the set of vertices $V_2 = \{i, (i-1) \bmod n, ..., (i - \lfloor \frac{k}{2} \rfloor) \bmod n\}$.
- $H_{n-1,n}$ is the complete graph K_n and $H_{2,n}$ is the cycle C_n.

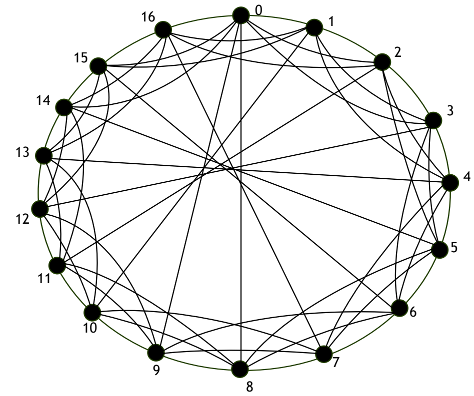

Figure 32: $H_{7,17}$ where both k and n are odd

$H_{k,n}$ is vertex transitive for the case when both k and n are not odd. When both k and n are odd, choosing any edges $\{0, \frac{n-1}{2}\}$ or $\{0, \frac{n+1}{2}\}$ from vertex 0 will be the same. In practice, $H_{k,n}$ is vertex transitive for all the cases and vertex 0 is considered as the originator. Since for every vertex i, where $i = 0, ..., n-1$, there are two cliques in $H_{k,n}$ formed by the set of vertices in V_1 and V_2, then from i, we can directly visit any of the vertices among $\{(i+1) \bmod n, ..., (i+\lfloor \frac{k}{2} \rfloor) \bmod n, (i-1) \bmod n, ..., (i-\lfloor \frac{k}{2} \rfloor) \bmod n\}$. From any node i, if we visit either node $(i+\lfloor \frac{k}{2} \rfloor) \bmod n$ or $(i-\lfloor \frac{k}{2} \rfloor) \bmod n$, then it is called a city-tour; otherwise it is called a village-tour.

Lemma 16. *There is always a shortest path between any pair of vertices in the Harary graph $H_{k,n}$, if we first take the city-tours as much as possible.*

Proof. Let us consider the pair of vertices to be 0 and X. Let us assume a path P_u where we have taken the maximum possible city-tours from 0 in order to reach vertex X. In one city-tour we can cover $\lfloor \frac{k}{2} \rfloor$ vertices. Thus, we can have at most $\lfloor \frac{2X}{k} \rfloor$ city-tours. From there, a maximum of one village-tour will take us to vertex X. Hence, $\text{dist}_{P_u}(0,X) = \lceil \frac{2X}{k} \rceil$. It can be seen easily that in another path P_v, if we take the village-tour in between the city-tours, then $\text{dist}_{P_v}(0,X)$ will still be $\lceil \frac{2X}{k} \rceil$.

Let us consider the third possible path P_w where we consider m village-tours along with some city-tours in order to reach vertex X, where $m > 1$. Let us assume that in m such village-tours, the vertices being covered are $c_1, c_2, ..., c_m$, where $c_i < \lfloor \frac{k}{2} \rfloor$ for $i = 1, ..., m$. Let us further assume that $\tau = c_1 + ... + c_m < m \lfloor \frac{k}{2} \rfloor$. After m passes, the vertices still need to be covered are $X - \tau$. If the remaining are all city-tours, then $\text{dist}_{P_w}(0,X) = m + \left\lceil \frac{X-\tau}{k/2} \right\rceil > m + \left\lceil \frac{X - m\lfloor \frac{k}{2} \rfloor}{k/2} \right\rceil \geq \lceil \frac{2X}{k} \rceil = \text{dist}_{P_v}(0,X)$ as $\tau < m \lfloor \frac{k}{2} \rfloor$. $\square$

In [163], the diameter of the Harary graph has been shown.

Lemma 17. *[163] Let $H_{k,n}$ be a Harary graph on n vertices where the degree of each vertex is at least k. Then,*
(i) $D(H_{k,n}) = \left\lceil \frac{n-1}{k} \right\rceil$ *when k is even.*
(ii) $D(H_{k,n}) = \left\lceil \frac{n+k-3}{2(k-1)} \right\rceil$ *when k is odd.*

Lemma 18. *Let $H_{k,n}$ be a Harary graph on n vertices where the degree of each vertex is at least k and $\frac{2n}{k} = p$ for some positive integer p. The broadcast time of $H_{k,n}$ from any originator, denoted as $b(H_{k,n})$ is*
(i) $b(H_{k,n}) \geq \left\lceil \frac{n-1}{k} \right\rceil$ *when k is even*
(ii) $b(H_{k,n}) \geq \left\lceil \frac{n+k-3}{2(k-1)} \right\rceil$ *when k is odd.*

Proof. Since $\frac{2n}{k} = p$, there is exactly one vertex in $H_{k,n}$ which is at a diametral distance from the original vertex. Thus, the proof is a direct consequence of the result in [116], where it has been shown that $b(G) \geq D(G)$ for any connected graph. □

Lemma 19. *Let $H_{k,n}$ be a Harary graph on n vertices where the degree of each vertex is at least k and $\frac{2n}{k} \neq p$ for some positive integer p. The broadcast time of $H_{k,n}$ from any originator, denoted as $b(H_{k,n})$ is*
(i) $b(H_{k,n}) \geq \left\lceil \frac{n-1}{k} \right\rceil + 1$ *when k is even*
(ii) $b(H_{k,n}) \geq \left\lceil \frac{n+k-3}{2(k-1)} \right\rceil + 1$ *when k is odd.*

Proof. Since $\frac{2n}{k} \neq p$, there are at least two vertices in $H_{k,n}$ which are at a diametral distance from the original vertex. It has been shown in [70], if there exists at least two vertices at a diametral distance D from vertex u in graph G, then $b(G) \geq D + 1$. This completes the proof. □

6.2 Approximation Algorithm for Broadcast time in Harary Graph

Under scheme S, we are going to consider two cases depending on whether k is even or odd. When k is even, the approximation algorithm S in $H_{k,n}$ starts by informing the vertices which can be reached through a city-tour from the originator 0 both in clockwise as well as in anti-clockwise directions. Every time a vertex receives the message, it first informs the vertex by making a city-tour. During the next sequence of time units, it informs the uninformed vertices in its clique.

When k is odd, the approximation algorithm S in $H_{k,n}$ starts by informing the vertex $\frac{n}{2}$ if n is even or vertex $\frac{n-1}{2}$ if n is odd at time 1. Similarly, the informed vertex first sends the message to the vertex along the city-tour. During the next sequence of time units, it informs the uninformed vertices in its clique.

<u>Broadcast Algorithm S:</u>

INPUT: A Harary Graph $H_{k,n}$ and originator vertex 0.

OUTPUT: Broadcast time $b_S(H_{k,n})$ and scheme of $H_{k,n}$.

BROADCAST-SCHEME-S$(H_{k,n}, 0)$

0. vertex 0 is the originator
1. When k is even:
   1.1. For $i = 1, ..., \lceil \frac{n}{k} \rceil$ do in clockwise direction
      1.1.1. vertex $(i-1)\frac{k}{2}$ informs vertex $i\frac{k}{2}$ at time i.
   1.2. For $j = 2, ..., \lceil \frac{n}{k} \rceil$
      1.2.1. vertex $(j-1)\frac{k}{2}$ informs the uninformed vertices in the clique formed by vertices $\{(j-1)\frac{k}{2}, (j-1)\frac{k}{2}+1, ..., j\frac{k}{2}-1\}$ starting at time $j+1$.
      1.2.2. Starting at time $\lceil \frac{n}{k} \rceil + 1$, vertex $\lceil \frac{n}{k} \rceil \frac{k}{2}$ informs the uninformed vertices in the clique $\{\lceil \frac{n}{k} \rceil \frac{k}{2}, \lceil \frac{n}{k} \rceil \frac{k}{2}+1, ..., (\lceil \frac{n}{k} \rceil + 1)\frac{k}{2}-1\}$.
   1.3. For $i = 2, ..., \lceil \frac{n}{k} \rceil$ do in anti-clockwise direction
      1.3.1. vertex $(n-(i-2)\frac{k}{2})$mod n informs vertex $(n-(i-1)\frac{k}{2})$mod n at time i.
   1.4. For $j = 2, ..., \lceil \frac{n}{k} \rceil$
      1.4.1. vertex $(n-(j-2)\frac{k}{2})$mod n informs the uninformed vertices in the clique formed by vertices $\{(n-(j-2)\frac{k}{2})$mod n, $(n-(j-2)\frac{k}{2})$mod $n+1$,..., $(n-(j-2)\frac{k}{2})$mod $n+\frac{k}{2}-1\}$ starting at time $j+1$.
      1.4.2. Starting at time $\lceil \frac{n}{k} \rceil + 1$, $(n-(\lceil \frac{n}{k} \rceil - 1)\frac{k}{2})$mod n informs the uninformed vertices in the clique $\{(n-(\lceil \frac{n}{k} \rceil - 1)\frac{k}{2})$mod n, $(n-(\lceil \frac{n}{k} \rceil - 1)\frac{k}{2})$ mod $n+1$,..., $(n-(\lceil \frac{n}{k} \rceil - 1)\frac{k}{2})$mod $n+\frac{k}{2}-1\}$.

2. When k is odd:
   2.1. vertex 0 informs vertex $\frac{n}{2}$ when n is even or vertex $\frac{n-1}{2}$ when n is odd at time 1. Let $v \in \{\frac{n}{2}, \frac{n-1}{2}\}$.

2.2. For $i = 2, ..., \lceil \frac{n}{2k} \rceil + 1$ do in clockwise direction

2.2.1. vertex $(i-2)\frac{k-1}{2}$ informs vertex $(i-1)\frac{k-1}{2}$ at time i.

2.2.2. vertex $v + (i-2)\frac{k-1}{2}$ informs vertex $v + (i-1)\frac{k-1}{2}$ at time i.

2.3. For $j = 3, ..., \lceil \frac{n}{2k} \rceil + 1$

2.3.1. vertex $(j-2)\frac{k-1}{2}$ informs the uninformed vertices in the clique formed by vertices $\{(j-2)\frac{k-1}{2}, (j-2)\frac{k-1}{2} + 1, ..., (j-1)\frac{k-1}{2} - 1\}$ starting at time $j+1$.

2.3.2. vertex $v + (i-2)\frac{k-1}{2}$ informs the uninformed vertices in the clique formed by vertices $\{v + (i-2)\frac{k-1}{2}, v + (i-2)\frac{k-1}{2} + 1, ..., v + (i-1)\frac{k-1}{2} - 1\}$ starting at time $j+1$.

2.3.3. Starting at time $\lceil \frac{n}{2k} \rceil + 2$ onwards:

(i)vertex $\lceil \frac{n}{2k} \rceil \frac{k-1}{2}$ informs the uninformed vertices in the clique $\{\lceil \frac{n}{2k} \rceil \frac{k-1}{2}, \lceil \frac{n}{2k} \rceil \frac{k-1}{2} + 1, ..., (\lceil \frac{n}{2k} \rceil + 1)\frac{k-1}{2} - 1\}$.

(ii)vertex $v + \lceil \frac{n}{2k} \rceil \frac{k-1}{2}$ informs the uninformed vertices in the clique $\{v + \lceil \frac{n}{2k} \rceil \frac{k-1}{2}, v + \lceil \frac{n}{2k} \rceil \frac{k-1}{2} + 1, ..., v + (\lceil \frac{n}{2k} \rceil + 1)\frac{k-1}{2} - 1\}$.

2.4. For $i = 3, ..., \lceil \frac{n}{2k} \rceil + 1$ do in anti-clockwise direction

2.4.1. vertex $(n - (i-3)\frac{k-1}{2}) \bmod n$ informs vertex $(n - (i-2)\frac{k-1}{2}) \bmod n$ at time i.

2.4.2. vertex $v - (i-3)\frac{k-1}{2}$ informs vertex $v - (i-2)\frac{k-1}{2}$ at time i.

2.5. For $j = 3, ..., \lceil \frac{n}{2k} \rceil + 1$

2.5.1. vertex $(n - (j-3)\frac{k-1}{2}) \bmod n$ informs the uninformed vertices in the clique formed by vertices $\{(n - (j-3)\frac{k-1}{2}) \bmod n, (n - (j-3)\frac{k-1}{2}) \bmod n + 1, ..., (n - (j-3)\frac{k-1}{2}) \bmod n + \frac{k-1}{2} - 1\}$ starting at time $j+1$.

2.5.2. vertex $v - (i-3)\frac{k-1}{2}$ informs the uninformed vertices in the clique formed by vertices $\{v - (i-3)\frac{k-1}{2}, v - (i-3)\frac{k-1}{2} + 1, ..., v - (i-3)\frac{k-1}{2} + \frac{k-1}{2} - 1\}$ starting at time $j+1$.

2.5.3. Starting at time $\lceil \frac{n}{2k} \rceil + 2$ onwards:

(i)vertex $v - (\lceil \frac{n}{2k} \rceil - 1)\frac{k-1}{2}$ informs the uninformed vertices in the clique $\{v - (\lceil \frac{n}{2k} \rceil - 1)\frac{k-1}{2}, v - (\lceil \frac{n}{2k} \rceil - 1)\frac{k-1}{2} + 1, ..., v - (\lceil \frac{n}{2k} \rceil - 1)\frac{k-1}{2} + \frac{k-1}{2} - 1\}$.

(ii)vertex $(n - (\lceil \frac{n}{2k} \rceil - 1)\frac{k-1}{2}) \bmod n$ informs the uninformed vertices in the clique $\{(n - (\lceil \frac{n}{2k} \rceil - 1)\frac{k-1}{2}) \bmod n, (n - (\lceil \frac{n}{2k} \rceil - 1)\frac{k-1}{2}) \bmod n + 1, ..., (n - (\lceil \frac{n}{2k} \rceil - 1)\frac{k-1}{2}) \bmod n + \frac{k-1}{2} - 1\}$.

Complexity: In every step of the algorithm, a set of informed vertices is informing another set of uninformed vertices and will be part of informed vertices in the next round. Thus the complexity of the algorithm S is $O(|V|)$.

Theorem 18. *Algorithm S gives $(\log \frac{k-2}{2} + 1)$-additive approximation when $\frac{2n}{k} = p$ for some positive integer p.*

Proof. 1. <u>When k is even</u>

SubCase 1.1: when $\frac{2n}{k} = 2q$, for some positive integer q.

In other words in either direction, starting from vertex 0, we can make q city-tours. Let us label the city-tours as $1, 2, ..., 2q$ from vertex 0 in the clockwise direction. Under algorithm S, starting at time 1 in a clockwise direction, vertex 0 makes $\frac{n}{k} = q$ city-tours to inform vertex $\frac{n}{k}\frac{k}{2} = \frac{n}{2}$ at time $\frac{n}{k}$. Similarly, starting at time 2 in an anti-clockwise direction, vertex 0 makes $q-1$ city-tours to inform vertex $(n-(\frac{n}{k}-1)\frac{k}{2})\text{mod}n$ $= \frac{n}{2} + \frac{k}{2}$ at time $\frac{n}{k}$. All the informed vertices will start informing the uninformed vertices in their respective cliques no later than $\frac{n}{k} + 1$ time units. Similarly, vertex $\frac{n}{2}$ will inform the vertices covered by the $(q+1)$th city-tour starting at time $\frac{n}{k}+1$. Since there are $\frac{k}{2} - 1$ uniformed vertices in the clique covered by the $(q+1)$th city-tour, it will take another $\log(\frac{k-2}{2})$ time units to complete broadcasting in the graph. Thus, $b_S(H_{k,n}) = \frac{n}{k} + \log(\frac{k-2}{2}) \leq b(H_{k,n}) + 1 + \log(\frac{k-2}{2})$ using Lemma 18(i) (since $\frac{n}{k} \leq \lceil \frac{n}{k} \rceil \leq \lceil \frac{n-1}{k} \rceil + 1 \leq b_S(H_{k,n}) + 1)$.

SubCase 1.2: when $\frac{2n}{k} = 2q - 1$.

Let us label the city-tours as $1, 2, ..., 2q - 1$ from vertex 0 in the clockwise direction. Under algorithm S, starting at time 1 in a clockwise direction, vertex 0 makes $\lceil \frac{n}{k} \rceil = q$ city-tours to inform vertex $\lceil \frac{n}{k} \rceil \frac{k}{2}$ at time $\frac{n}{k}$. Similarly, starting at time 2 in an anti-clockwise direction, vertex 0 makes $q - 1$ city-tours to inform vertex $(n - (\lceil \frac{n}{k} \rceil - 1)\frac{k}{2})\text{mod}n = n - \lceil \frac{n}{k} \rceil \frac{k}{2} + \frac{k}{2}$ at time $\frac{n}{k}$. All the informed vertices will start informing the uninformed vertices in their respective cliques no later than $\frac{n}{k} + 1$ time units. Similarly, vertex $\lceil \frac{n}{k} \rceil \frac{k}{2}$ will inform the vertices covered by the qth city-tour starting at time $\lceil \frac{n}{k} \rceil + 1$. Since there are $\frac{k}{2} - 1$ uniformed vertices in the clique covered by the qth city-tour, it will take another $\log(\frac{k-2}{2})$ time units to complete broadcasting in the graph. Thus, $b_S(H_{k,n}) = \lceil \frac{n}{k} \rceil + \log(\frac{k-2}{2}) \leq b(H_{k,n}) + 1 + \log(\frac{k-2}{2})$ using Lemma 18(i).

2. <u>When k is odd</u>

Under algorithm S, at time unit one, vertex 0 sends a message to vertex $\frac{n}{2}$ if n is even, otherwise the message is sent to vertex $\frac{n-1}{2}$. Let us assume $v \in \{\frac{n}{2}, \frac{n-1}{2}\}$.

SubCase 2.1. when $\frac{2n}{k-1} = 2q$

In scheme S, starting at time 2 in a clockwise direction, vertices 0 and v each makes $\lceil \frac{n}{2k} \rceil$ city-tours to inform vertices $\lceil \frac{n}{2k} \rceil \frac{k-1}{2}$ and $v + \lceil \frac{n}{2k} \rceil \frac{k-1}{2}$ respectively at time $\lceil \frac{n}{2k} \rceil + 1$. Similarly, starting at time 3 in an anti-clockwise direction, vertices 0 and v each makes $\lceil \frac{n}{2k} \rceil - 1$ city-tours to inform vertices $n - (\lceil \frac{n}{2k} \rceil - 1)\frac{k-1}{2}$ and $v - (\lceil \frac{n}{2k} \rceil - 1)\frac{k-1}{2}$ respectively at time $\lceil \frac{n}{2k} \rceil + 1$. Thus, in scheme S, there are in total $2\lceil \frac{n}{2k} \rceil - 1$ city-tours on either side of vertex 0. In general, one can make at most $\lceil \frac{n}{k-1} \rceil$ city-tours on either side of vertex 0. For the sake of clarity, if in a clique, there is an informed vertex u, we will term it as a clique of u.

Let $\lceil \frac{n}{k-1} \rceil = p$, p is any positive integer $\Rightarrow \frac{n}{k-1} = p - 1$ (taking only the integer value). Now, $\frac{n}{2k} \leq \frac{n}{2(k-1)} = \frac{p-1}{2} \Rightarrow \lceil \frac{n}{2k} \rceil \leq \frac{p+1}{2} \Rightarrow 2\lceil \frac{n}{2k} \rceil - 1 \leq p = \lceil \frac{n}{k-1} \rceil$ and $2\lceil \frac{n}{2k} \rceil - 1$ is an odd integer.

If $\lceil \frac{n}{k-1} \rceil$ is odd, in scheme S the uninformed vertices in the cliques of $\lceil \frac{n}{2k} \rceil \frac{k-1}{2}$ and $v - (\lceil \frac{n}{2k} \rceil - 1)\frac{k-1}{2}$ share a common vertex as $2\lceil \frac{n}{2k} \rceil - 1 \leq \lceil \frac{n}{k-1} \rceil$ and $\frac{2n}{k-1} = 2q$. Similarly, the cliques of $v + \lceil \frac{n}{2k} \rceil \frac{k-1}{2}$ and $n - (\lceil \frac{n}{2k} \rceil - 1)\frac{k-1}{2}$ also share a common vertex. If $\lceil \frac{n}{k-1} \rceil$ is even, cliques of $\lceil \frac{n}{2k} \rceil \frac{k-1}{2}$ and $v - (\lceil \frac{n}{2k} \rceil - 1)\frac{k-1}{2}$ do not share any common vertex and there are exactly $\frac{k-1}{2}$ vertices between the cliques. This is also true for the cliques of $v + \lceil \frac{n}{2k} \rceil \frac{k-1}{2}$ and $n - (\lceil \frac{n}{2k} \rceil - 1)\frac{k-1}{2}$ when $\lceil \frac{n}{k-1} \rceil$ is even. In both cases, vertex $\lceil \frac{n}{2k} \rceil \frac{k-1}{2}$ takes exactly another $\log \frac{k-2}{2}$ time units to inform the vertices in its clique. Thus, $b_S(H_{k,n}) = \lceil \frac{n}{2k} \rceil + 1 + \log(\frac{k-2}{2})$.

Using Lemma 18(ii), $b(H_{k,n}) \geq \left\lceil \frac{n+k-3}{2(k-1)} \right\rceil \geq \lceil \frac{n+k-3}{2k} \rceil \geq \lceil \frac{n}{2k} \rceil$ for $k \geq 3$.

Hence, $b_S(H_{k,n}) \leq \left\lceil \frac{n+k-3}{2(k-1)} \right\rceil + 1 + \log(\frac{k-2}{2}) \leq b(H_{k,n}) + 1 + \log(\frac{k-2}{2})$.

SubCase 2.2. when $\frac{2n}{k-1} = 2q + 1$

Similar to subcase 2.1., under scheme S, at time $\lceil \frac{n}{2k} \rceil + 1$, vertices $\lceil \frac{n}{2k} \rceil \frac{k-1}{2}$, $v + \lceil \frac{n}{2k} \rceil \frac{k-1}{2}$, $n - (\lceil \frac{n}{2k} \rceil - 1)\frac{k-1}{2}$ and $v - (\lceil \frac{n}{2k} \rceil - 1)\frac{k-1}{2}$ receive the message from vertex 0. Similarly, The number of city-tours can be at most $\lceil \frac{n}{k-1} \rceil = q + 1$ on either side of vertex 0. There are in total $2\lceil \frac{n}{2k} \rceil - 1$ city-tours on either side of vertex 0 in S and $2\lceil \frac{n}{2k} \rceil - 1 \leq \lceil \frac{n}{k-1} \rceil = q + 1$.

If q is even, then $\lceil \frac{n}{k-1} \rceil$ is odd. Thus, in scheme S the uninformed vertices in the cliques of $\lceil \frac{n}{2k} \rceil \frac{k-1}{2}$ and $v - (\lceil \frac{n}{2k} \rceil - 1)\frac{k-1}{2}$ share $\frac{k-1}{4}$ common vertices (as $\frac{2n}{k-1} = 2q + 1$ $\Rightarrow \frac{n}{2} = q\frac{k-1}{2} + \frac{k-1}{4}$). Similarly, the cliques of $v + \lceil \frac{n}{2k} \rceil \frac{k-1}{2}$ and $n - (\lceil \frac{n}{2k} \rceil - 1)\frac{k-1}{2}$ also share $\frac{k-1}{4}$ common vertices. If q is odd, then $\lceil \frac{n}{k-1} \rceil$ is even. As a result, cliques of $\lceil \frac{n}{2k} \rceil \frac{k-1}{2}$ and $v-(\lceil \frac{n}{2k} \rceil -1)\frac{k-1}{2}$ do not share any common vertex and there are exactly $\frac{k-1}{4}$ vertices between the cliques. This is also true for the cliques of $v+\lceil \frac{n}{2k} \rceil \frac{k-1}{2}$ and $n-(\lceil \frac{n}{2k} \rceil -1)\frac{k-1}{2}$. Thus, either of vertices $\lceil \frac{n}{2k} \rceil \frac{k-1}{2}$, $v+\lceil \frac{n}{2k} \rceil \frac{k-1}{2}$, $n-(\lceil \frac{n}{2k} \rceil -1)\frac{k-1}{2}$ or $v-(\lceil \frac{n}{2k} \rceil -1)\frac{k-1}{2}$ take less than $\log \frac{k-2}{2}$ time units to inform the uninformed vertices in their respective cliques. However, starting at time $\lceil \frac{n}{2k} \rceil +2$, vertex $(n-(\lceil \frac{n}{2k} \rceil -2)\frac{k-1}{2})$ takes exactly $\log \frac{k-2}{2}$ time units to inform the vertices in its clique. Similarly, $b_S(H_{k,n})$ $\leq b(H_{k,n}) + 1 + \log(\frac{k-2}{2})$ using Lemma 18(ii) for $k \geq 3$. □

Theorem 19. *Algorithm S gives $(\log \frac{k-2}{2})$-additive approximation when $\frac{2n}{k} \neq p$ for some positive integer p.*

Proof. 1. When k is even

Under algorithm S, similar to subcase 1.2 of Theorem 18, at time $\lceil \frac{n}{k} \rceil$, vertices $\lceil \frac{n}{k} \rceil \frac{k}{2}$ and $n - \lceil \frac{n}{k} \rceil \frac{k}{2} + \frac{k}{2}$ receive the message from vertex 0 in either direction. Since, $\frac{2n}{k} \neq r$, let us assume $\lceil \frac{n}{k} \rceil = q \Rightarrow \frac{n+c}{k} = q$ for some $1 \leq c < k$. When $\frac{k}{2} \leq c < k$ $\Rightarrow \frac{k}{2} \leq kq - n < k \Rightarrow -\frac{k}{2} \geq n - kq > -k$ we get, $n - \lceil \frac{n}{k} \rceil \frac{k}{2} + \frac{k}{2} = n - q\frac{k}{2} + \frac{k}{2} \leq q\frac{k}{2}$ (as $-\frac{k}{2} \geq n - kq) = \lceil \frac{n}{k} \rceil \frac{k}{2}$. So, in scheme S, the uninformed vertices in their respective cliques overlap each other. Similarly, when $1 \leq c < \frac{k}{2}$ we get, $\lceil \frac{n}{k} \rceil \frac{k}{2} < n - \lceil \frac{n}{k} \rceil \frac{k}{2} + \frac{k}{2}$. As a result, there are at least $c < \frac{k}{2}$ vertices that do not overlap. Thus, either $\lceil \frac{n}{k} \rceil \frac{k}{2}$ or $n - \lceil \frac{n}{k} \rceil \frac{k}{2} + \frac{k}{2}$ take less than $\log \frac{k-2}{2}$ time units to inform the uninformed vertices in their respective cliques. However starting at time $\lceil \frac{n}{k} \rceil + 1$, vertex $(n - (\lceil \frac{n}{k} \rceil - 2)\frac{k}{2})$ takes exactly $\log \frac{k-2}{2}$ time units to inform the vertices in its clique. Similarly, $b_S(H_{k,n})$ $\leq b(H_{k,n}) + \log(\frac{k-2}{2})$ using Lemma 19(i).

2. When k is odd

This is exactly similar to the subcase 2.2 of Theorem 18. Depending on whether $\lceil \frac{n}{k-1} \rceil$ is odd or even, in scheme S the uninformed vertices in the cliques of $\lceil \frac{n}{2k} \rceil \frac{k-1}{2}$ and $v - (\lceil \frac{n}{2k} \rceil - 1)\frac{k-1}{2}$ either share c_1 common vertices or do not share any common vertex. Instead there are exactly c_2 vertices between the cliques, where $1 \leq c_1, c_2 < \frac{k-1}{2}$. This is also true for the cliques of $v + \lceil \frac{n}{2k} \rceil \frac{k-1}{2}$ and $n - (\lceil \frac{n}{2k} \rceil - 1)\frac{k-1}{2}$. Similarly, starting at time $\lceil \frac{n}{2k} \rceil + 2$, vertex $(n - (\lceil \frac{n}{2k} \rceil - 2)\frac{k-1}{2})$ takes exactly $\log \frac{k-2}{2}$ time units to inform the vertices in its clique. Thus, $b_S(H_{k,n}) \leq b(H_{k,n}) + \log(\frac{k-2}{2})$ using Lemma 19(ii) for $k \geq 3$. □

Observations: $b_S(H_{k,n}) \leq b(H_{k,n}) + \log(\frac{k-2}{2})$.

(i) When $k = n-1$, $b_S(H_{n-1,n}) \leq b(H_{n-1,n}) + 1 + \log(\frac{n-3}{2}) < b(H_{n-1,n}) + \log n \leq 2b(H_{n-1,n})$ as $b(H_{n-1,n}) \geq \log n$ is always true. Thus, algorithm S is $(2-\epsilon)$-approximation.

(ii) When $k = \sqrt{n}$, similarly, $b_S(H_{k,n}) < b(H_{k,n}) + \frac{1}{2}\log n \leq 1.5b(H_{k,n})$. Thus, algorithm S is $(1.5-\epsilon)$-approximation.

(iii) It is also clear that when k is constant (even very large) then S gives a constant additive approximation.

6.3 Modified Harary Graph

In this section we will first introduce what we called a modified Harary graph. We will find the diameter of the new graph and present an approximation algorithm to broadcast in this graph.

Definition 3: Modified Harary graph, $MH_{k,n}$ on n vertices where the degree of each vertex is k, is constructed as follows: Let the vertices be labelled as $0, 1, ..., n-1$. The two vertices i and j are joined when vertices $j \in \{(i - 2^r + 1) \text{mod } n, (i + 2^r - 1) \text{mod } n\}$ for $i = 0, ..., n-1$ and $r = 1, ..., \frac{k}{2}$ (see Figure 33). Here n, k are even and n is sufficiently larger than k.

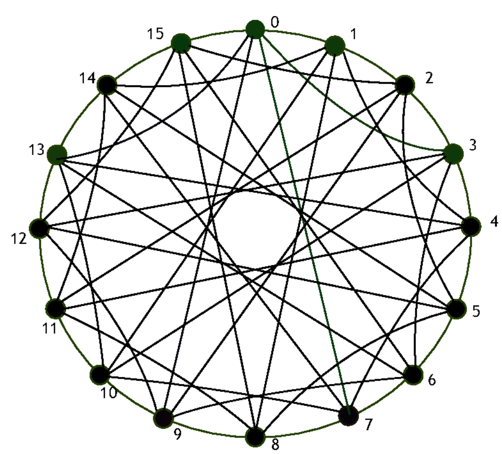

Figure 33: Modified Harary Graph $MH_{6,16}$

Properties of Modified Harary Graph:

- It is vertex transitive.

- Each vertex is connected to $\frac{k}{2}$ vertices in the clockwise direction and another $\frac{k}{2}$ vertices in the anti-clockwise direction.

- Every vertex has degree k.

The motivation behind introducing the modified Harary graph $MH_{k,n}$ is that in $MH_{k,n}$, a vertex which is at a farther distance from a particular vertex as compared to Harary graph, can be reached in one time unit. As a result the broadcasting in $MH_{k,n}$ on the same number of vertices and the same degree can be done in a more efficient manner as compared to $H_{k,n}$, where both k and n are even. In figures 30 and 33, it has been shown that in $MH_{6,16}$, the farthest vertices that can be reached directly from vertex 0 are vertices 7 and 9. However in $H_{6,16}$, vertices 3 and 13 are the farthest vertices that are connected with vertex 0. As a result broadcasting in $MH_{6,16}$ is faster than broadcasting in $H_{6,16}$.

6.3.1 Diameter of Modified Harary Graph and Lower bound on Broadcast Time

As the graph is vertex transitive, we will consider vertex 0 as the originator in $MH_{k,n}$. We will modify the definitions of city-tours and village-tours for $MH_{k,n}$. From any node i in $MH_{k,n}$, if we visit either node $(i + 2^{\frac{k}{2}} - 1) \text{mod } n$ or $(i - 2^{\frac{k}{2}} + 1) \text{mod } n$, then it is called a city-tour; otherwise it is termed as village-tour. For any vertex i, we will denote the set of vertices $\{i, i + 1, ..., i + 2^{\frac{k}{2}} - 2\}$ as the region of i.

Lemma 20. *Let $MH_{k,n}$ be a modified Harary graph on n vertices where the degree of each vertex is k. Then $D(MH_{k,n})$ is $\left\lceil \frac{n}{2(2^r-1)} \right\rceil + r - 1$, where $r = \frac{k}{2}$.*

Proof. The result in Lemma 16 is also applicable in the modified Harary graph $MH_{k,n}$. This is attributed to two reasons. Firstly, $MH_{k,n}$ is also a minimal k-connected graph on n vertices. Secondly, each vertex is connected to $\frac{k}{2}$ vertices in both clockwise and anti-clockwise directions. From vertex 0, at most $\frac{n}{2}$ vertices can be covered from either clockwise or anti-clockwise direction. To start, we will take the maximum possible city-tours from vertex 0. In one city-tour we can cover $2^r - 1$ vertices. Thus, initially we will traverse through $\left\lceil \frac{n}{2(2^r-1)} \right\rceil$ city-tours. There are $2^r - 1$ vertices in between nodes $\left\lceil \frac{n}{2(2^r-1)} \right\rceil (2^r - 1)$ and $(\left\lceil \frac{n}{2(2^r-1)} \right\rceil - 1)(2^r - 1)$. From either of these 2 nodes, we can make village-tours which will cover the 2^{r-1} vertices from either node. So in a recursive way, with each village-tour, the number of vertices to be covered reduces to half. Thus, at most we need $r-1$ village-tours in order to cover 2^{r-1} vertices. This makes the diameter of the graph to be $\left\lceil \frac{n}{2(2^r-1)} \right\rceil + r - 1$. □

Lemma 21. *Let $MH_{k,n}$ be a modified Harary graph on n vertices where the degree of each vertex is k. The broadcast time of $MH_{k,n}$ from any originator,*
(i) $b(MH_{k,n}) \geq \left\lceil \frac{n}{2(2^r-1)} \right\rceil + r - 1$, if $\frac{n}{(2^r-1)} = p$
(ii) $b(MH_{k,n}) \geq \left\lceil \frac{n}{2(2^r-1)} \right\rceil + r$, if $\frac{n}{(2^r-1)} \neq p$
where $r = \frac{k}{2}$ and p is any positive integer.

Proof. (i) Since $\frac{n}{(2^r-1)} = p$, there is exactly one vertex in $MH_{k,n}$ which is at a diametral distance from the original vertex. Thus, the proof is a direct consequence of the result in [116], where it has been shown that $b(G) \geq D(G)$ for any connected graph.
(ii) Since $\frac{n}{(2^r-1)} \neq p$, there are at least two vertices in $MH_{k,n}$ which are at a diametral distance from the original vertex. It has been shown in [70], if there exists at least two vertices at a diametral distance D from vertex u in graph G, then $b(G) \geq D+1$. Hence, $b(MH_{k,n}) \geq \left\lceil \frac{n}{2(2^r-1)} \right\rceil + r$. □

6.3.2 Approximation Algorithm for Broadcast time in the Modified Harary Graph

The approximation algorithm S_m in $MH_{k,n}$ starts by informing the vertices that can be reached through a city-tour from the originator 0 both in clockwise and in anti-clockwise directions. Every time an informed vertex first sends the message to an uninformed vertex along the city-tour. During the next sequence of time units it informs the uninformed vertices in its region following the REGION-BROADCAST scheme.

Broadcast Algorithm S_m:
INPUT: A Modified Harary Graph $MH_{k,n}$ and originator vertex 0.
OUTPUT: Broadcast time $b_{S_m}(MH_{k,n})$ and scheme of $MH_{k,n}$.
BROADCAST-SCHEME-$S_m(MH_{k,n}, 0)$

0. vertex 0 is the originator and let $r = \frac{k}{2}$.
1. For $i = 1, ..., \left\lceil \frac{n}{2(2^r-1)} \right\rceil$ do in clockwise direction

1.1. vertex $(i-1)(2^r-1)$ informs vertex $i(2^r-1)$ at time i.

2. For $j = 2, ..., \left\lceil \frac{n}{2(2^r-1)} \right\rceil$

   2.1. vertex $(j-1)(2^r-1)$ informs the uninformed vertices in its region starting at time $j+1$.

   REGION-BROADCAST-RB($(j-1)(2^r-1), r, j$).

   2.2. Starting at time $\left\lceil \frac{n}{2(2^r-1)} \right\rceil + 1$, vertex $\left\lceil \frac{n}{2(2^r-1)} \right\rceil (2^r-1)$ informs its region. REGION-BROADCAST-RB($\left\lceil \frac{n}{2(2^r-1)} \right\rceil (2^r-1), r, \left\lceil \frac{n}{2(2^r-1)} \right\rceil$).

3. For $i = 2, ..., \left\lceil \frac{n}{2(2^r-1)} \right\rceil$ do in anti-clockwise direction

   3.1. vertex $(n-(i-2)(2^r-1))$mod n informs vertex $(n-(i-1)(2^r-1))$ mod n at time i.

4. For $j = 2, ..., \left\lceil \frac{n}{2(2^r-1)} \right\rceil$

   4.1. vertex $(n-(j-2)(2^r-1))$mod n informs the uninformed vertices in its region starting at time $j+1$.

   REGION-BROADCAST-RB($(j-1)(2^r-1), r, j$).

   4.2. Starting at time $\left\lceil \frac{n}{2(2^r-1)} \right\rceil + 1$, vertex $(n - (\left\lceil \frac{n}{2(2^r-1)} \right\rceil - 1)(2^r-1))$mod n informs its region.

   REGION-BROADCAST-RB($(n - (\left\lceil \frac{n}{2(2^r-1)} \right\rceil - 1)(2^r-1))$mod $n, r, \left\lceil \frac{n}{2(2^r-1)} \right\rceil$).

<u>REGION-BROADCAST-RB(u, r, τ)</u>

1. If $r = 1$, then $MH_{k,n}$ is a cycle and there will be no uninformed vetex in the region of u.
2. If $r = 2$, $u \stackrel{\tau+1}{\rightarrow} (u+2^2-1)$ and $u \stackrel{\tau+2}{\rightarrow} (u+2^1-1)$, $(u+2^2-1) \stackrel{\tau+2}{\rightarrow} (u+2^2-2)$.
3. If $r \geq 3$

   3.1. Let $c = 1$

   3.2. While $r \geq 4$ do

   3.2.1. vertex u informs vertex $u+2^{r-1}-1$ at time unit $\tau+c$

   3.2.2. SUB-REGION-BROADCAST-SRB($u+2^{r-1}-1, \tau+c$)

   3.2.3. $r = r-1$ and $c = c+1$

   3.3. If $r = 3$

   3.3.1. The set of uninformed vertices within the region $\{u, ..., u+2^3-1\}$ are $u+1, ..., u+6$.

   3.3.2. $u \stackrel{\tau+c+1}{\rightarrow} (u+3)$

$u \overset{\tau+c+2}{\to} (u+1)$ $\qquad$ $(u+3) \overset{\tau+c+2}{\to} (u+6)$

$(u+3) \overset{\tau+c+3}{\to} (u+4)$ $\qquad$ $(u+1) \overset{\tau+c+3}{\to} (u+2)$ $\qquad$ $(u+6) \overset{\tau+c+3}{\to} (u+5)$

SUB-REGION-BROADCAST-SRB$(u + 2^{r-1} - 1, \tau + c)$

0. We will consider the vertices within the region R=$\{u + 2^{r-1} - 1$,...,u+ $2(2^{r-1} - 1)$ =$u + 2^r - 2\}$. Let the set of informed vertices in R be I. Initially $I = \{u + 2^{r-1} - 1\}$ and R is the region for $u + 2^{r-1} - 1$
1. For every vertex, $v_1 \in I$ do
   1.1. v_1 informs the farthest uninformed vertex v_2 within its own region.
   1.2. The set of vertices within $\{v_1, ..., v_2\}$ becomes the region for both v_1 and v_2.
   1.3. Update $I = I + v_2$

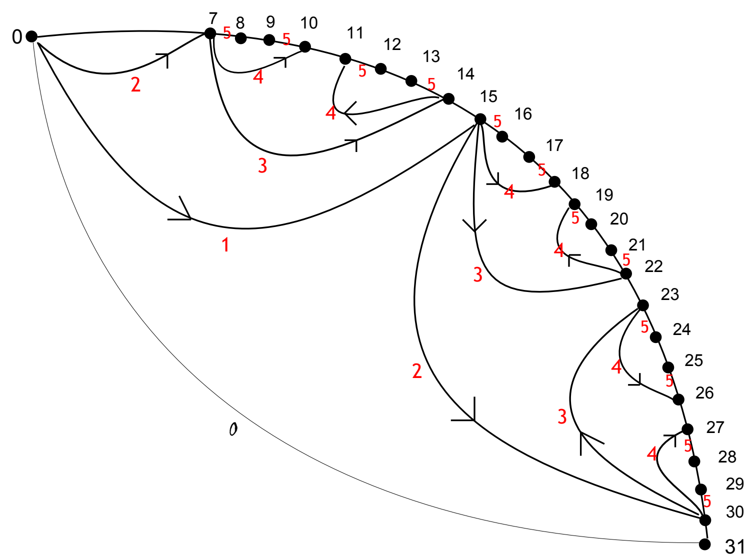

Figure 34: REGION-BROADCAST scheme for the region containing vertices $\{0, 1, ..., 2^5 - 1\}$ except when $r = 3$

The REGION-BROADCAST scheme for the region containing vertices $\{0, 1, ..., 2^5 - 1\}$ except when $r = 3$ has been illustrated in Figure 34. The figures in black are the labels of the vertices and the figures in red show the broadcast times. We assume in the figure that vertex 0 informs vertex 31 at time 0. The broadcast times for the regions $\{15, ..., 30\}$ and $\{7, ..., 14\}$ are based on the SUB-REGION-BROADCAST scheme. Initially $I = \{15\}$ for the region containing the vertices $\{15, ..., 30\}$ and region for vertex 15 is $\{15, ..., 30\}$. 15 informs the farthest uninformed vertex within its own region (in this case 30) at time 2. The region for vertex 30 is $\{15, ..., 30\}$ and $I = \{15, 30\}$. In the next time unit, both 15 and 30 respectively inform the farthest uninformed vertices in their regions. Thus, 15 informs vertex 22 and 30 informs vertex 23. The region for both vertices 15 and 22 now becomes $\{15, ..., 22\}$ and that for vertices 30 and 23 is $\{23, ..., 30\}$ and $I = \{15, 30, 22, 23\}$. Similarly in the next time unit, all the vertices in set I inform the farthest uninformed vertex in their own regions.

Under the Sub-Region-Broadcast scheme, when node, $u+2^{r-1}-1$ sends a message to its farthest uninformed vertex $u+2^r-2$, it divides the region with $u+2^r-2-(u+2^{r-1}-1)+1 = 2^{r-1}$ vertices. In the next time unit when both vertices $u+2^{r-1}-1$ and $u+2^r-2$ inform their respective farthest uninformed vertices within their region, the new regions formed will each have 2^{r-2} vertices. Thus, every time we send a message to the farthest uninformed vertex, we divide the region into two new regions with same number of vertices.

Complexity: In all the broadcast schemes S_m, Region-Broadcast and Sub-Region-Broadcast, at a given time, a set of informed vertices are informing another set of uninformed vertices and will be part of informed vertices in the next round. This makes the total complexity of the algorithms to be $O(|V|)$.

Lemma 22. *Let $MH_{k,n}$ be a modified Harary graph on n vertices where the degree of each vertex is k and let $r = \frac{k}{2}$. The Sub-Region-Broadcast scheme takes r time units to broadcast in a region of $MH_{k,n}$ with 2^r vertices.*

Proof. We will prove the result by method of induction.
Base Case: When $r = 1$: The region has 2 vertices, 0 and 1. Vertex 0 sends the message to vertex 1 at time unit one. So base case is true.
Inductive hypothesis: Assume that when $r = m$, broadcasting can be done by m time units.
Induction step: Assume $r = m + 1$. Vertex 0 sends message to vertex $2^{m+1} - 1$. According to the Sub-Region-Broadcast scheme, in the next time unit both vertices 0 and $2^{m+1} - 1$ will simultaneously inform the vertices in the regions each having 2^m vertices. We know from the inductive hypothesis that it will take m time units to inform in a region with 2^m vertices. However, 0 informs $2^{m+1} - 1$ at time one. Thus, it will take $m + 1$ time units to broadcast in a region with 2^{m+1} vertices. □

Lemma 23. *Let $MH_{k,n}$ be a modified Harary graph on n vertices where the degree of each vertex is k and let $r = \frac{k}{2}$. The Region-Broadcast scheme takes r time units to broadcast in a region of $MH_{k,n}$ with 2^r vertices.*

Proof. When $r = 2$, it is clear from step 2 of Region-Broadcast that it will take 2 time units to complete broadcasting. When $r = 3$, step 3.3 shows 3 time units are enough to broadcast. When $r \geq 4$, in step 3.2.1, u informs vertices $u + 2^{r-1} - 1$, $u + 2^{r-2} - 1$,...,$u + 2^{r-i} - 1$ at times $1, 2, ..., i$ respectively (we assume $\tau = 0$ here). In other words, after time i, we have a region containing 2^{r-i} vertices which will be operated upon by the Sub-Region-Broadcast scheme. From Lemma 22, we know the Sub-Region-Broadcast takes $r - i$ time units to finish broadcasting in this region. In total, $i + r - i = r$ time units are necessary. □

Theorem 20. *Algorithm S_m gives 1-additive approximation when $\frac{n}{(2^r-1)} = p$ for some positive integer p.*

Proof. We assume $\frac{k}{2} = r$.
Case 1: when $\frac{n}{2^r-1} = 2q$:

In other words in either direction, starting from vertex 0, we can make q city-tours. Let us label the city-tours as $1, 2, ..., 2q$ from vertex 0 in a clockwise direction. Under algorithm S_m, starting at time 1 in a clockwise direction, vertex 0 makes $\frac{n}{2(2^r-1)} = q$ city-tours to inform vertex $\frac{n}{2(2^r-1)}(2^r - 1) = \frac{n}{2}$ at time $\frac{n}{2(2^r-1)}$. Similarly, starting at time 2 in an anti-clockwise direction, vertex 0 makes $q - 1$ city-tours to inform vertex $(n - (\frac{n}{2(2^r-1)} - 1)(2^r - 1)) \bmod n = \frac{n}{2} + 2^r - 1$ at time $\frac{n}{2(2^r-1)}$. All the informed vertices will start informing the uninformed vertices in their respective regions no later than $\frac{n}{2(2^r-1)} + 1$ time units. Similarly, vertex $\frac{n}{2}$ will inform the vertices covered by the $(q + 1)$th city-tour. Since there are 2^r vertices in that region, we know from Lemma 23 that Region-Broadcast scheme will take r time units to finish broadcasting. Thus, $b_{S_m(MH_{k,n})} \leq \frac{n}{2(2^r-1)} + r \leq b(MH_{k,n}) + 1$ from Lemma 21(i).
Case 2: $\frac{n}{2^r-1} = 2q - 1$ is not possible.

Let us assume by contradiction that $\frac{n}{2^r-1} = 2q - 1$ is possible. Since $2^r - 1$ is odd, then n is also odd. This contradicts as in $MH_{k,n}$, n is even. □

Theorem 21. *Algorithm S_m is optimal when $\frac{n}{(2^r-1)} \neq p$ for some positive integer p.*

Proof. We assume $\frac{k}{2} = r$.

This is similar to the result we proved in Case 2 of Theorem 19. Similarly, depending on whether $\left\lceil \frac{n}{2^r-1} \right\rceil$ is odd or even in scheme S_m, the uninformed vertices in the regions of $\frac{n}{2(2^r-1)}(2^r-1)$ and $(n-(\frac{n}{2(2^r-1)}-1)(2^r-1))$ either share c_1 common vertices or do not share any common vertex. Instead, there are exactly c_2 vertices between the regions, where $1 \leq c_1, c_2 < 2^r$. Thus, these vertices will take less than r time units to inform the uninformed vertices in their regions using the Region-Broadcast scheme. However starting at time $\left\lceil \frac{n}{2(2^r-1)} \right\rceil + 1$, vertex $(n - (\frac{n}{2(2^r-1)} - 2)(2^r - 1))$ takes exactly r time units to inform the 2^r uninformed vertices using Region-Broadcast (from Lemma 23). Thus, $b_{S_m(MH_{k,n})} \leq \left\lceil \frac{n}{2(2^r-1)} \right\rceil + r \leq b(MH_{k,n})$ from Lemma 21(ii). □

Theorem 22. *$MH_{2\lceil \log n \rceil -2,n}$ is a broadcast graph.*

Proof. From Lemma 21(i), we know that $b(MH_{k,n}) \geq \left\lceil \frac{n}{2^{\frac{k}{2}+1}-2} \right\rceil + \frac{k}{2} - 1$.
When $k = 2\lceil \log n \rceil - 2$, $2^{\frac{k}{2}+1} = 2^{\lceil \log n \rceil} = n + c$ for some positive integer c.
Thus, $b(MH_{2\lceil \log n \rceil -2,n}) \geq \left\lceil \frac{n}{n-2} \right\rceil + \log n - 1 - 1 = 2 + \log n$ - $2 = \log n$. Hence, $MH_{2\lceil \log n \rceil -2,n}$ is a broadcast graph. □

Chapter 7

Diameter Broadcast Problem

In [101] a lower bound on the broadcast time of a general graph $G = (V, E)$ has been given where G has at least $d + 2$ vertices that are all at a distance d from a certain vertex v_0. The broadcast time of such a graph cannot be less than $d + 2$. In this section we have generalized the above result and obtained a lower bound on the broadcast time of G which has at least $\binom{d+k-1}{d} + 1$ vertices that are all at distance exactly d from v_0, where $k \geq 1$. First we consider the simple cases when $k = 2, 3$.

Lemma 24. *If a graph $G = (V, E)$ has more than $d + 1$ vertices at a distance d from another vertex v_0, then the broadcast time of G satisfies the following inequality: $b(G) \geq d + 2$.*

Proof. We start the proof by noting that at time d there can be only one informed vertex, v_d, at a distance d from the originator, call it v_0. Let $P = \{v_0, v_1, ..., v_d\}$ be the path from v_0 to v_d. At time i, vertex v_i receives the message and informs vertex v_{i+1} at time $i+1$, where $1 \leq i \leq d-1$. At time $d+1$, v_{d-1} informs a new vertex which is also at a distance d from v_0. Similarly, if all the vertices v_i along the path P except for v_d inform a new vertex at time $i + 2$ which through a chain of calls can inform a vertex at distance d from v_0 at time $d + 1$. Since there are d vertices on the path P (except v_d), at most d vertices can be informed at time $d + 1$. Finally including v_d, there can be at most $d + 1$ vertices at a distance d from v_0 that are informed at time $d + 1$. □

Lemma 25. *If a graph $G = (V, E)$ has more than $(d + 1)\frac{d+2}{2}$ vertices at a distance d from another vertex v_0, then the broadcast time of G satisfies the following inequality: $b(G) \geq d + 3$.*

Proof. Similar to the proof in Lemma 24, at time d there can be only one informed vertex, v_d, at a distance d from the originator, v_0 in the path $P = \{v_0, v_1, ..., v_d\}$. At time i, vertex v_i receives the message and informs vertex v_{i+1} at time $i+1$, where $1 \leq i \leq d-1$. Starting at time $d+1$ onwards, v_{d-1} can inform 2 new vertices at a distance d from v_0 by time $d+2$. Similarly, v_{d-2} informs a new vertex at time d which in turn informs 2 uninformed vertices at a distance d from v_0 by time $d+2$. Through another branch starting at time $d+1$, v_{d-2} sends the message to a new vertex which is also at a distance d from the originator by making a chain of 2 calls. In total, 3 new vertices can be informed through v_{d-2} by time $d+2$. Similarly there are exactly 4 new vertices which are at a distance d from v_0 that receive the message through v_{d-3} by time $d+2$. In general, $d+2-(i+1)$ new vertices at a distance d from v_0 can be informed through v_i by time $d+2$, where $0 \leq i \leq d-1$ (see Figure 35). Thus, maximum number of vertices at a distance d from v_0 that can be informed at time $d+2 = 1(v_d) + \sum_{i=0}^{d-1} d+2-(i+1) = 1+2+3+...+(d+1) = (d+1)\frac{d+2}{2}$. □

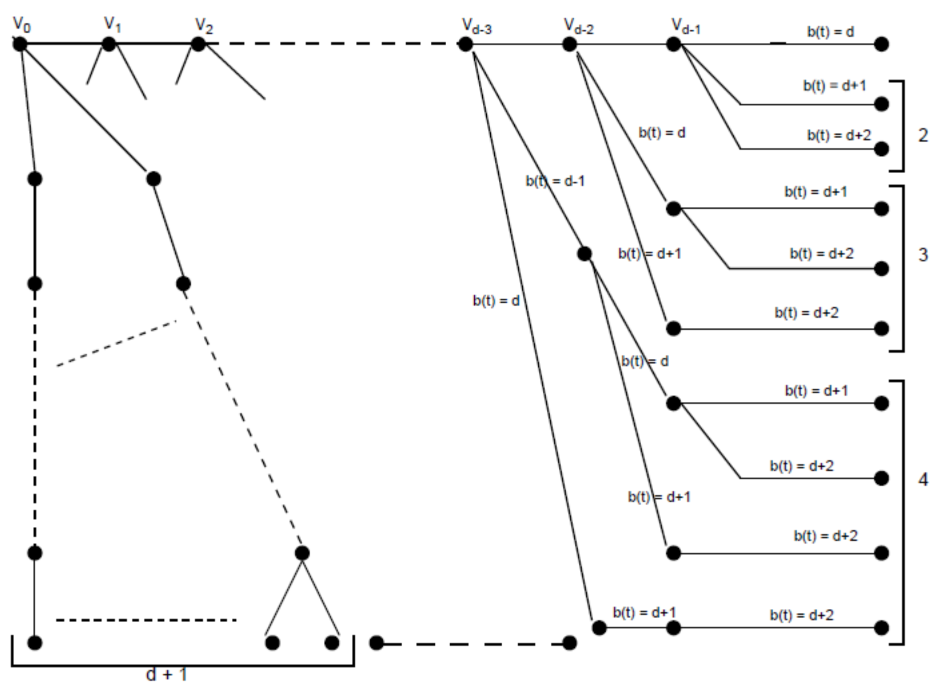

Figure 35: Maximum number of vertices that can be at distance d from the originator if the broadcast time is equal to $d+2$

Before we prove the general case, we define what is called a binomial tree and look at some of its properties.

Binomial Tree: A binomial tree is defined recursively. The binomial tree of order 0, denoted B_0 is a single vertex. A binomial tree of order k, denoted B_k is constructed from two copies of B_{k-1} by connecting their roots by an edge. Actually, B_k will have a root node whose children are roots of binomial trees of orders $k-1, k-2, ..., 2, 1, 0$ (in this order). See Figure 36.

Properties: (i) B_k has 2^k vertices.

(ii) B_k has $\binom{k}{d}$ vertices at depth d.

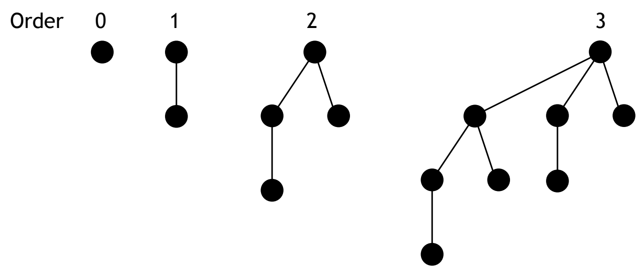

Figure 36: Binomial trees of order 0 to 3

The maximum number of vertices that can be informed in a k-broadcast graph (recall from Chapter 2, k-broadcasting is a variant of broadcasting in which an informed vertex can call up to k of its neighbors in each time unit) by time $t \geq 0$ along paths of length at most d has been shown in Lemma 2 of [94]. We use this result to generate our case when $k = 1$.

Lemma 26. *In any graph G, the maximum number of vertices which can be informed in a classical broadcast model by time $t \geq 0$ along paths of length at most d is at most $\sum_{i=0}^{d} \binom{t}{i}$.*

Proof. Following is a proof by induction.

Base case: When $t = 1$, there are only two vertices v_0 and v_1. v_0 sends the message to v_1 at time 1. In this case d is also 1. So base case is true.

Inductive hypothesis: Assume it is true when $t = k - 1$.

Induction step: Assume $t = k$. From the inductive hypothesis, the maximum number of vertices that can be informed by time $k-1$ along paths of length l is $\binom{k-1}{l}$, where $1 \leq l \leq d$. At time k, at most all the informed vertices which are at a distance $l-1$ from the originator will each inform a new uninformed vertex. Thus at time k, the maximum number of informed vertices which are at a distance l from the originator will be $\binom{k-1}{l} + \binom{k-1}{l-1} = \binom{k}{l}$, for $1 \leq l \leq d$. □

Theorem 23. *If a graph $G = (V, E)$ has more than $\binom{d+k-1}{d}$ vertices at a distance d from another vertex v_0, where $k \geq 1$, then the broadcast time of G satisfies the following inequality: $b(G) \geq d + k$.*

Proof. We will prove the theorem by contradiction. Let the graph G has at least $\binom{d+k-1}{d} + 1$ vertices at a distance d from originator vertex v_0, where $k \geq 1$. Let us assume that $b(G) \leq d + k - 1$.

In the graph G, during each time unit the number of informed vertices can at most double when none of the informed vertices remain idle. From Lemma 26, the maximum number of vertices that can be informed by time $d + k - 1$ at a distance d from v_0 is at most $\sum_{i=0}^{d} \binom{d+k-1}{i}$. Thus the broadcast tree of G is a subtree of the binomial tree B_{d+k-1}. Now, B_{d+k-1} has $\binom{d+k-1}{d}$ vertices at depth d. Thus the broadcast tree of G can have at most $\binom{d+k-1}{d}$ vertices at a distance d from v_0. This contradicts the fact that G has $\binom{d+k-1}{d} + 1$ vertices at a distance d from v_0.

□

Chapter 8

Conclusion and Future Work

The broadcast problem in general is an NP-Hard problem and it remains NP-Complete even for 3-regular planar graph and for a graph whose vertex set can be partitioned into a clique and an independent set. The broadcast problem is shown to be NP-Hard to approximate within a factor $3 - \epsilon$. The best known approximation for broadcasting in general graphs is $O(\frac{\log(|V|)}{\log\log(|V|)})$. Polynomial time algorithms for the broadcast problem are only known for some tree like graphs. In particular, there exist linear algorithms for trees, tree of cycles and necklace graphs. In all these graphs any two cycles intersect in at most one vertex. Tree of cliques is the only graph where two cycles intersect in many vertices but there is a $O(n \log\log n)$ algorithm. However, solving the broadcast problem for tree of cliques is relatively easy because in clique any pair of vertices are neighbors.

In the thesis our choice of graph classes is motivated by the longer research path: to increase the connectivity of the graphs to the extent that there is no constant approximation algorithm for the broadcast problem in that graph assuming that there is no constant approximation for broadcasting in general graphs. In this respect we first study the broadcast problem in the simple graphs where the cycles intersect in at least one vertex and present a constant approximation algorithm to broadcast in the graph. In Chapter 3 we consider the simplest graph where cycles intersect only at 2 vertices, namely k-path graph. As it turns out finding the exact broadcast time even in the k-path graph is not very simple. We give an approximation algorithm for the k-path graph, where the approximation ratio is less than 4 in the worst case. When k is bounded by some finite, large range of values, the approximation ratio can be at most 3. It is natural that the problem becomes difficult when k is large because the graph becomes denser. The main characteristic of this algorithm is its greedy approach: at each round the junction vertices always inform along the path having maximum number of uninformed vertices. Our approximation algorithm also takes advantage of the fact that a minimum time broadcast scheme first informs the shortest path. This leads to the possibility of generating the optimum broadcast time when the difference of path lengths between each pair is at least 2. Minimum time broadcasting in k-path graph is difficult when the number of paths is much larger than the lengths of the paths or the lengths of the paths form an arithmetic series with difference 1. The future work in this area of course will be to design a polynomial algorithm to find the exact broadcast time or to prove that the broadcast problem is NP-Hard for k-path graph.

In Chapter 4 we consider a simple graph where cycles intersect at a single vertex, namely k-cycle graph. Finding the exact broadcast time even in this graph is not trivial. We give an approximation algorithm for the k-cycle graph, with the approximation ratio 2. Our approximation algorithm follows the greedy method where at every round, the central vertex informs along the cycle having maximum number of uninformed vertices. This leads us in generating the optimum broadcast time when the difference of cycle lengths between each pair is at least 4. Minimum time broadcasting in k-cycle graph is difficult when the lengths of the paths form an arithmetic series with difference either 1 or 3. The future work in this area will be to design a polynomial algorithm to find the exact broadcast time or to prove that the broadcast problem in arbitrary k-cycle graph is NP-Hard.

In Chapter 5 we study broadcasting in hypercube of trees. The algorithm for fully connected trees and tree of cycles can not be applied to hypercube of trees because every pair of non-tree vertices in hypercube of trees are not connected. We present a linear time algorithm to find the broadcast time from any originator for hypercube of trees containing one tree. Depending on the broadcast times in hypercube and tree, the algorithm in an optimal way decides when to broadcast in tree or hypercube from the root vertex. For the general case we present a 2-approximation algorithm to find the broadcast time from any originator. The two main directions for future work are proving the NP-Completeness or designing a polynomial algorithm for the broadcast problem in hypercube of trees.

In Chapter 6 we consider the broadcast problem in Harary graph $H_{k,n}$, which is the minimal k-connected graph on n vertices. We present a linear $\log \frac{k-2}{2} + 1$-additive approximation to find the broadcast time in the graph. The approximation algorithm follows a natural way of initially informing the vertex which is at a farthest distance from the informed vertex and then informing all the vertices within its clique. The additive approximation is justified in a sense that the lower bound on broadcast time can be achieved when there are large numbers of $\frac{k}{2}$ regions in $H_{k,n}$ and all the vertices in a clique will receive the message from the vertices of the previously informed cliques at the same time. Moreover, when $\frac{2n}{k} = 4$, our broadcast algorithm will take exactly $b(H_{k,n}) + \log \frac{k-2}{2} + 1$ time units to complete broadcasting. Next we design a new modified Harary graph where both k, n are even. In modified Harary graph, a vertex which is at a farther distance from a particular vertex as compared to Harary graph, can be reached in one time unit. This leads to the possibility of generating a better approximation algorithm (in this case a linear 1-additive approximation instead of a linear $\log \frac{k-2}{2} + 1$-additive approximation in case of Harary graph) to find the broadcast time in the modified Harary graph. Since, the approximation is very close to optimal, the future work will be to provide a polynomial algorithm for the broadcast problem in modified Harary graph.

In the line of the graphs being studied in this thesis, there are other graph structures where cycles intersect in at least one vertex such as the k-cycle of trees and the cycle of k-path. It is natural to study the broadcast problems in these graphs too. k-cycle of trees is a k-cycle graph where some or all of its vertices are the roots of the trees. A natural way to solve the broadcast problem in k-cycle of trees is to first broadcast all the vertices in the k-cycle. The root vertices will then broadcast in the trees attached with them. A cycle of k-path is a graph formed by m k-path graphs such that every junction vertex of a k-path graph is connected to a junction vertex of its adjacent k-path graph. One way to broadcast in m k-path graphs will be to inform all the junction vertices as early as possible using the shortest path in every k-path graph and then broadcast along the longer paths.

Bibliography

[1] R. Ahlswede, L. Gargano, H. Haroutunian, and L. Khachatrian. Fault tolerant minimum broadcast networks. *Networks*, 27:293–307, 1996.

[2] R. Ahlswede, H. Haroutunian, and L. Khachatrian. Messy broadcasting in networks. *Communications and Cryptography, R. Blahut, D. Costello, T. Mittelholzer (Ed.), Kluwer Academic Pubishing*, pages 13–24, 1994.

[3] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg. A lower bound for radio broadcast. *Journal of Computer and System Sciences*, 43:290–298, 1991.

[4] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg. Single round simulation of radio networks. *Journal of Algorithms*, 13:188–210, 1992.

[5] A. Averbuch, R.H. Shabtai, and Y. Roditty. Efficient construction of broadcast graphs. *Discrete Appl. Math.*, 171:9–14, 2014.

[6] A. Bar-Noy, J. Bruck, C.T. Ho, S. Kipnis, and B.Scheiber. Computing global combine operations in the multi-port postal model. *IEEE Transactions on Parallel and Distributed Systems*, 6:896–900, 1995.

[7] A. Bar-Noy, S. Guha, J. Naor, and B. Schieber. Multicasting in heterogeneous networks. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing, STOC '98*, pages 448–453. ACM, 1998.

[8] A. Bar-Noy and S. Kipnis. Broadcasting multiple messages in simultaneous send/receive systems. *Discrete Appl. Math.*, 55(2):95–105, 1994.

[9] A. Bar-Noy, S. Kipnis, and B. Schieber. Optimal multiple message broadcasting in telephone-like communication systems. *Discrete Appl. Math.*, 100(1–2):1–15, 2000.

[10] R. Bar-Yehuda, O. Goldreich, and A. Itai. Efficient emulation of single hop radio network with collision detection on multi-hop radio network with no collision detection. *Distributed Computing*, 5:67–71, 1991.

[11] B. Beauquier, S. Perennes, and O. Delmas. Tight bounds for broadcasting in the linear cost model. *Journal of Interconnection Networks*, 2(2):175–188, 2001.

[12] R. Beier and J.F. Sibeyn. A powerful heuristic for telephone gossiping. In *Proc. of the 7th International Colloquium on Structural Information Communication Complexity, SIROCCO'00*, pages 17–36, 2000.

[13] A. Bellaachia and A. Youssef. Personalized broadcasting in banyan-hypercube networks. In *4th International Conference on Computer Communications and Networks (ICCCN '95).*, pages 470–474, 1995.

[14] S. Berg. Random contact process, snowball sampling and factorial series distribution. *J. Appl. Probl.*, 20:31–46, 1983.

[15] J.C. Bermond, P. Fraigniaud, and J. Peters. Antepenultimate broadcasting. *Networks*, 26:125–137, 1995.

[16] J.C. Bermond, H.A. Harutyunyan, A.L. Liestman, and S. Perennes. A note on the dimensionality of modified knödel graphs. In *IJFCS: International Journal of Foundations of Computer Science*, volume 8, pages 109–117, 1997.

[17] J.C. Bermond, P. Hell, A.L. Liestman, and J. Peters. Broadcasting in bounded degree graphs. *SIAM J. Discr. Math.*, 5(1):10–24, 1992.

[18] J.C. Bermond, P. Hell, A.L. Liestman, and J. Peters. Sparse broadcast graphs. *Discrete Appl. Math.*, 36:97–130, 1992.

[19] J.C. Bermond and C. Peyrat. Broadcasting in debruijn networks. In *19th South-Eastern Conf. on Combinatorics, Graph Theory and Computing*, 1988.

[20] P. Berthome, A. Ferreira, and S. Perennes. Optimal information dissemination in star and pancake networks. *IEEE Transaction on Parallel and Distributed Systems*, 7(12):1292–1300, 1996.

[21] B.D. Birchler, A. Esfahanian, and E. Torng. Information dissemination in restricted routing networks. In *Proceedings of the International Symposium on Combinatorics and Applications*, pages 33–44, 1996.

[22] J. Bruck and C.T. Ho. Efficient global combine operations in the multi-port message passing systems. *Parallel Processing Letters*, 3:335–346, 1993.

[23] K. Calvert, E.W. Zegura, and S. Bhattacharjee. How to model an internetwork. In *IEEE INFOCOM*, 1996.

[24] V.R. Cane. A note on the size of epidemics and the number of people hearing a rumour. *Journal of the Royal Statics Society, Series B*, 28(3):487–490, 1966.

[25] S.C. Chau and A.L. Liestman. Constructing minimal broadcasting networks. *J. Combin. Inform. System Sci.*, 10:110–122, 1985.

[26] X. Chen. An upper bound for the broadcast function b(n). *Chinese J. Comput.*, 13:605–611, 1990.

[27] P. Chinn, S. Hedetniemi, and S. Mitchell. Multiple message broadcasting in complete graphs. In *10th South-Eastern Conf. on Combinatorics, Graph Theory and Computing, Utilitas Mathematica*, pages 251–260, 1979.

[28] I. Chlamtac and S. Kutten. On broadcasting in radio networks - problem analysis and protocol design. *IEEE Transaction on Communications*, 33:1240–1246, 1985.

[29] I. Chlamtac and S. Kutten. A spatial-reuse tdma/fdma for mobile multi-hop radio networks. In *Proceedings IEEE INFOCOM.*, pages 389–394, 1985.

[30] I. Chlamtac and S. Kutten. Tree-based broadcasting in multihop radio networks. *IEEE Transaction on Communications*, 36(10):1209–1223, 1987.

[31] I. Chlamtac and O. Weinstein. The wave expansion approach to broadcasting in multi-hop radio networks. *IEEE Transaction on Communications*, 39:426–433, 1991.

[32] A.L.H. Chow, L. Golubchik, S. Khuller, and Y. Yao. On the tradeoff between playback delay and buffer space in streaming. In *Proceedings of the 2009 IEEE*

International Symposium on Parallel and Distributed Processings, pages 1–12, 2009.

[33] E. Cockayne and A. Thomason. Optimal multi-message broadcasting in complete graphs. In *11th South-Eastern Conf. on Combinatorics, Graph Theory and Computing*, pages 181–199, 1980.

[34] F. Comellas and C. Dalfo. Optimal broadcasting in 2-dimensional manhattan street networks. *J. Parallel and Distributed Computing and Networks*, 4:37–51, 2003.

[35] F. Comellas, H.A. Harutyunyan, and A.L. Liestman. Messy broadcasting in mesh and torus networks. *Journal of Interconnection Networks*, 4:37–51, 2003.

[36] F. Comellas and P. Hell. Broadcasting in generalized chordal rings. *Networks*, 42(3):123–134, 2003.

[37] F. Comellas and M. Mitjana. Broadcasting in cycle prefix digraphs. *Discrete Applied Mathematics*, 83(1–3):31–39, 1998.

[38] D.J. Daley and D.G. Kendall. Epidemics and rumors. *Nature*, 204:11–18, 1964.

[39] D.J. Daley and D.G. Kendall. Stochastic rumors. *J. Inst. Maths. Applics.*, 1:42–55, 1965.

[40] J. de Rumeur. *Communication dans les reseaux de processeurs.* Collection Etudes et Recherches en Informatique, Paris, 1994.

[41] O. Delmas. *Communications par commutation de circuits dan les reseaux d'interconnexion.* PhD thesis, Universite de Nice - Sophia Antipolis, U.F.R. de Sciences, France, 1997.

[42] M. Dietzfelbinger. *Gossiping and broadcasting versus computing functions in networks.* Technische Universitat Ilmenau, 1999.

[43] K. Diks and A. Pelc. Broadcasting with universal lists. *Networks*, 27(3):183–196, 1996.

[44] H. Dinh and Y.A. Kim. On the tree-based broadcast algorithm(tba) and an improved upper bound on the optimal broadcasting/gossiping time in d-dimensional torus graphs. *Networks*, 1997.

[45] M.J. Dinneen, M.R. Fellows, and V. Faber. Algebraic constructions of efficient broadcast networks. In *Proceedings of the 9th International Symposium, on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, AAECC-9*, pages 152–158. Springer-Verlag, 1991.

[46] M.J. Dinneen, J.A. Ventura, M.C. Wilson, and G. Zakeri. Compound constructions of broadcast networks. *Discrete Appl. Math.*, 93(2–3):205–232, 1999.

[47] M.B. Doar and A. Nexion. A better model for generating test networks. In *IEEE GLOBECOM '96*, pages 86–93, 1996.

[48] R. Dunstan. The rumor process. *J. Appl. Prob.*, 19:759–766, 1982.

[49] M. Elkin and G. Kortsarz. Combinatorial logarithmic approximation algorithm for directed telephone broadcast problem. In *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing, STOC '02*, pages 438–447. ACM, 2002.

[50] M. Elkin and G. Kortsarz. Sublogarithmic approximation for telephone multicast: path out of jungle (extended abstract). In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms, SODA '03*, pages 76–85. Society for Industrial and Applied Mathematics, 2003.

[51] M. Elkin and G. Kortsarz. Logarithmic inapproximability of the radio broadcast problem. *Journal of Algorithms*, 52(1):8–25, 2004.

[52] M. Elkin and G. Kortsarz. Polylogarithmic inapproximability of the radio broadcast problem. In *Proceedings of the 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems.*, pages 105–114, 2004.

[53] M. Elkin and G. Kortsarz. An improved algorithm for radio broadcast problem. In *Proceedings ACM-SIAM on Discrete Algorithms (SODA '05).*, 2005.

[54] M. Elkin and G. Kortsarz. Polylogarithmic additive hardness of approximating radio broadcast problem. *SIAM Journal of Discrete Mathematics*, 19(4):881–899, 2005.

[55] S. Even and B. Monien. On the number of rounds necessary to disseminate information (extended abstract). In *First ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 318–327, 1989.

[56] A. Farley. Minimal broadcast networks. *Networks*, 9:313–332, 1979.

[57] A.M. Farley. Broadcast time in communication networks. *SIAM J. Applied Math*, 39(2):385–390, 1980.

[58] A.M. Farley. Minimum-time line broadcast networks. *Networks*, 37:59–70, 1980.

[59] A.M. Farley, S.T. Hedetniemi, S. Mitchell, and A. Proskurowski. Minimum broadcast graphs. *Discrete Math*, 25:189–193, 1979.

[60] U. Feige, D. Peleg, P. Raghavan, and E. Upfal. Randomized broadcast in networks. In *Proceedings of the international symposium on Algorithms, SIGAL '90*, pages 128–137. Springer-Verlag New York, Inc., 1990.

[61] R. Feldmann, J. Hromkovic, S. Madhavapeddy, B. Monien, and P. Mysliwietz. Optimal algorithms for dissemination of information in generalized communication modes. *Discrete Appl. Math.*, 53(1–3):55–78, 1994.

[62] W. Feller. *An Introduction to Probability Theory and its Applications, 2nd ed.* Wiley, New York, 1971.

[63] G. Fertin and A. Raspaud. A survey on knödel graphs. *Discrete Appl. Math.*, 137(2):173–195, 2004.

[64] G. Fertin, A. Raspaud, H. Schroder, O. Sykora, and I. Vrto. Diameter of the knödel graph. In *Proceedings of the 26th International Workshop on Graph-Theoretic Concepts in Computer Science, WG '00*, pages 149–160, 2000.

[65] E. Fluery. *Communications, routage et architectures des machines a memoire distribuee - Autour du routage wormhole.* PhD thesis, Universite de Lyon, France, 1996.

[66] P. Fraigniaud. *Approximation Algorithms for Collective Communications with Limited Link and Node-Contention*. Tech. Rep. LRI-1264, Univ. Paris-Sud, France, 2000.

[67] P. Fraigniaud. Approximation algorithms for minimum-time broadcast under the vertex-disjoint path nodes. In *the 9th Annual Eur. Symp. on Alg., ESA '01*, volume 2161, pages 440–451. LNCS, 2001.

[68] P. Fraigniaud. Minimum-time broadcast under edge-disjoint paths modes. In *the 2nd International Conference on Fun with Algorithms (FUN '01)*, pages 133–148, 2001.

[69] P. Fraigniaud. A note on line broadcast in digraphs under the edge-disjoint paths mode. *Discrete Appl. Math.*, 144(3):320–323, 2004.

[70] P. Fraigniaud and E. Lazard. methods and problems of communication in usual networks. *Discrete Appl. Math*, 53:79–133, 1994.

[71] P. Fraigniaud and S. Vial. Approximation algorithms for broadcasting and gossiping. *J. Parallel and Distrib. Comput.*, 43(1):47–55, 1997.

[72] P. Fraigniaud and S. Vial. Heuristic algorithms for personalized communication problems in point-to-point networks. In *Proc. of the 4th Colloquium on Structural Information Communication Complexity, SIROCCO'97*, pages 240–252, 1997.

[73] P. Fraigniaud and S. Vial. Comparison of heuristics for one-to-all and all-to-all communication in partial meshes. In *Parallel Processing Letters*, volume 9, pages 9–20, 1999.

[74] H. Frank, I. Gitman, and R. van Slyke. Routing in packet swirching broadcast radio networks. *IEEE Transaction on Communications*, 24:926–930, 1976.

[75] A. Frieze and G. Grimmett. The shortest-path problem for graphs with random arc-lengths. *Discr. Appl. Mathematics*, 10:57–77, 1985.

[76] I. Gaber and Y. Mansour. Broadcast in radio networks. *ACM-SIAM Symposium on Discrete Algorithms*, pages 577–585, 1995.

[77] L. Gargano, A.L. Liestman, J.G. Peters, and D. Richards. Reliable broadcasting. *Discrete Applied Maths*, 53(1–3):135–148, 1994.

[78] L. Gargano and U. Vaccaro. On the construction of minimal broadcast networks. *Networks*, 19:673–689, 1989.

[79] L. Gargano and U. Vaccaro. Minimum time broadcast networks tolerating a logarithmic number of faults. *SIAM J. Discrete Maths*, 5(2):178–198, 1992.

[80] W. Goffman. Mathematical approach to the spread of scientific ideas-the history of mast cell research. *Nature*, 212(61):449–452, 1966.

[81] W. Goffman and V.A. Newill. Generalization of epidemic theory, an application to the transmission of ideas. *Nature*, 204:225–228, 1964.

[82] W. Goffman and V.A. Newill. Communication and epidemic processes. In *Proc. Royal Soc. A*, volume 298, pages 316–334, 1967.

[83] F. Harary. The maximum connectivity of a graph. In *National Academy of Sciences of the United States of America*, volume 48, pages 1142–1146, 1962.

[84] H. Haroutunian. Minimal broadcast networks. In *4th International Colloquium on Coding Theory, Dilijan, Armenia*, pages 36–40, 1991.

[85] T. Hart and H.A. Harutyunyan. Improved messy broadcasting in hypercube and complete bipartite graph. *Congressus Numerantium*, 156:124–140, 2002.

[86] T. Hart and H.A. Harutyunyan. Messy broadcasting in common communication networks. In *Proceedings of the 21st Biennial Symposium on Communications.*, pages 534–538, 2002.

[87] H.A. Harutyunyan. Multiple message broadcasting in modified knödel graph. In *SIROCCO '00*, pages 157–165, 2000.

[88] H.A. Harutyunyan. Minimum multiple message broadcast graphs. *Networks*, 47(4):218–224, 2006.

[89] H.A. Harutyunyan, P. Hell, and A.L. Liestman. Tight upper bounds on messy broadcast time. *Discrete Applied Mathematics*, 159(5):322–327, 2011.

[90] 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. ACM, 2009.

[91] H.A. Harutyunyan and A.L. Liestman. Messy broadcasting. *Parallel Processing Letters*, 4(2):149–159, 1998.

[92] H.A. Harutyunyan and A.L. Liestman. More broadcast graphs. *Discrete Appl. Math.*, 98:81–102, 1999.

[93] H.A. Harutyunyan and A.L. Liestman. Improved upper and lower bounds for k-broadcasting. *Networks*, 37:94–101, 2001.

[94] H.A. Harutyunyan and A.L. Liestman. k-broadcasting in trees. *Networks*, 38(3):163–168, 2001.

[95] H.A. Harutyunyan and A.L. Liestman. Upper bounds on the broadcast function using minimum dominating sets. *Discrete Math.*, 312:2992–2996, 2012.

[96] H.A. Harutyunyan, A.L. Liestman, K. Makino, and T.C. Shermer. Nonadaptive broadcasting in trees. *Networks*, 57(2):157–168, 2011.

[97] H.A. Harutyunyan, A.L. Liestman, J.G. Peters, and D. Richards. *Broadcasting and Gossiping*. Handbook of Graph Theory, Chapman and Hall/CRC, Second Edition, 2013.

[98] H.A. Harutyunyan, A.L. Liestman, and B. Shao. A linear algorithm for finding the k-broadcasting center of a tree. *Networks*, 53(3):287–292, 2009.

[99] H.A. Harutyunyan and E. Maraachlian. Broadcasting in optimal bipartite double loop graphs. In *Proceedings of the conference on Information Visualization, IV '06*, pages 521–528. IEEE Computer Society, 2006.

[100] H.A. Harutyunyan and E. Maraachlian. Linear algorithm for broadcasting in unicyclic graphs. In Guohui Lin, editor, *COCOON*, pages 372–382. Springer, 2007.

[101] H.A. Harutyunyan and E. Maraachlian. Near optimal broadcasting in optimal triple loop graphs. In *Proceedings of the 22nd International Conference on*

Advanced Information Networking and Applications, AINA '08, pages 227–233, 2008.

[102] H.A. Harutyunyan and E. Maraachlian. On broadcasting in unicyclic graphs. *J. Comb. Optim.*, 16(3):307–322, 2008.

[103] H.A. Harutyunyan and E. Maraachlian. Broadcasting in fully connected trees. In *Proceedings of the 2009 15th International Conference on Parallel and Distributed Systems*, ICPADS '09, pages 740–745, Washington, DC, USA, 2009. IEEE Computer Society.

[104] H.A. Harutyunyan and E. Maraachlian. Linear algorithm for broadcasting in networks with no intersecting cycles. In *PDPTA*, pages 296–301. CSREA Press, 2009.

[105] H.A. Harutyunyan and E. Maraachlian. Broadcasting in hierarchical tree cluster network. In *PDPTA*, pages 479–486, 2010.

[106] H.A. Harutyunyan and C.D. Morosan. The spectra of knödel graphs. *Informatica an International Journal of Computing and Informatics*, 30(3):295–299, 2006.

[107] H.A. Harutyunyan and C.D. Morosan. On the minimum path problem in knödel graphs. *Networks*, 50(1):86–91, 2007.

[108] H.A. Harutyunyan and B. Shao. A heuristic for k-broadcasting in arbitrary networks. In *IEEE Conference on Applications of Graph Theory, IV03*, pages 287–292, 2003.

[109] H.A. Harutyunyan and B. Shao. Optimal k-broadcast in trees. *Congressus Numerantium*, 164, 2003.

[110] H.A. Harutyunyan and B. Shao. An efficient heuristic for broadcasting in networks. *J. Parallel Distrib. Comput.*, 66(1):68–76, 2006.

[111] H.A. Harutyunyan and P. Taslakian. Orderly broadcasting in a 2d torus. In *Proceedings of the Information Visualisation, Eighth International Conference, IV '04*, pages 370–375, 2004.

[112] H.A. Harutyunyan and W. Wang. Broadcasting algorithm via shortest paths. In *Proceedings of the 2010 IEEE 16th International Conference on Parallel and Distributed Systems, ICPADS '10*, pages 299–305, 2010.

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

[114] P. Hell and K. Seyffarth. Broadcasting in planar graphs. *Australas. J. Combin.*, 17:309–318, 1998.

[115] M.C. Heydemann, J. Opatrny, and D. Sotteau. Broadcasting and spanning trees in debruijn and kautz networks. *Discrete Applied Math.*, 27–28:297–317, 1992.

[116] J. Hromkovic, R. Klasing, B. Monien, and R. Peine. Dissemination of information in interconnection networks. *Combinatorial Network Theory*, pages 125–212, 1996.

[117] J. Hromkovic, R. Klasing, A. Pelc, P. Ruzicka, and W. Unger. *Dissemination of information in communication networks: broadcasting, gossiping, leader election, and fault- tolerance*. Texts in Theoretical Computer Science, An EATCS Series. Springer, 2005.

[118] J. Hromkovic, R. Klasing, and E.A. Stohr. Dissemination of information in vertex-disjoint paths mode. *Computers and Artificial Intelligence*, 15(4):125–212, 1996.

[119] J. Hromkovic, R. Klasing, W. Unger, and H. Wagener. Optimal algorithms for broadcast and gossip in the edge-disjoint path modes (extended abstract). *INFORMATION AND COMPUTATION*, pages 219–230, 1994.

[120] M. Hwang and F. A. Briggs. *Computer Architecture and Parallel Processing*. 4th printing, McGraw-Hill International Editions, New York, 1988.

[121] K. Jansen and H. Muller. The minimum broadcast time problem for several processor networks. *Theoretical Computer Science*, 147:69–85, 1995.

[122] C.D. Jeschke, J. Hromkovic, and B. Monien. Optimal algorithms for dissemination of information in some interconnection networks. *Algorithmatica*, 10(1):24–40, 1993.

[123] R.E. Kahn, S.A. Gronemeyer, J. Burchfiel, and R.C. Kunzelman. Advances in packet radio technology. In *Proceedings of the IEEE.*, volume 66, 1978.

[124] W.O. Kermack and A.G. McKendrick. A contribution to the mathematical theory of epidemics. In *Proc. Royal Soc. A*, volume 115, pages 700–721, 1927.

[125] L. Khachatrian and H. Haroutunian. Construction of new classes of minimal broadcast networks. In *Proc. 3rd International Colloquium on Coding Theory*, pages 69–77, 1990.

[126] S. Khuller, Y.A. Kim, and Y.C. Wan. On generalized gossiping and broadcasting (extended abstract). In *European Symposium on Algorithms (ESA 03)*, pages 373–384, 2003.

[127] S. Khuller, Y.A. Kim, and Y.C. Wan. Broadcasting on networks of workstations. *Symposium on Parallel Algorithms*, pages 279–288, 2005.

[128] J.H. Kim and K.Y. Chwa. Optimal broadcasting with universal lists based on competitive analysis. *Networks*, 45(4):224–231, 2005.

[129] W. Knödel. New gossips and telephones. *Discrete Mathematics*, 13:95, 1975.

[130] J.C. Konig and E. Lazard. Minimum k-broadcast graph. *Discrete Appl. Math*, 53:199–209, 1994.

[131] G. Kortsarz and D. Peleg. Approximation algorithms for minimum time broadcast. *SIAM J. Discrete Math*, 8(3):401–427, 1995.

[132] D.R. Kowalski and A. Pelc. Deterministic broadcasting time in radio networks of unknown topology. In *Proceedings of the 43rd Symposium on Foundations of Computer Science.*, pages 63–72, 2002.

[133] R. Labahn. A minimum broadcast graph on 63 vertices. *Discrete Appl. Math.*, 53:247–250, 1994.

[134] H.D. Landahl. On the spread of information with time and distance. *Bull. Math. Biophys.*, 15:367–381, 1953.

[135] H.G. Landau and A. Rapoport. Contribution to the mathematical theory of contagion and spread of information, i. spread through a thoroughly mixed population. *Bull. Math. Biophys.*, 15:173–183, 1953.

[136] E. Lazard. Broadcasting in dma-bound bounded degree graphs. *Discrete Appl. Math*, 37–38:387–400, 1992.

[137] F. Thomson Leighton. *Introduction to parallel algorithms and architectures: array, trees, hypercubes.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992.

[138] C. Li, T.E. Hart, K.J. Henry, and I.A. Neufeld. Average-case messy broadcasting. *Journal of Interconnection Networks*, 9(4):487–505, 2008.

[139] A. Liestman and J. Peters. Broadcast networks of bounded degree. *SIAM J. Discrete Math*, 1998.

[140] A.L. Liestman, T.C. Shermer, and M.J. Suderman. Broadcasting multiple messages in hypercubes. In *International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN 00)*, pages 274–281, 2000.

[141] L. Lu, W. Aiello, and F. Chung. Random evolution in massive graphs. In *Proc. of 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 510–519, 2001.

[142] M. Maheo and J.F. Sacle. Some minimum broadcast graphs. *Discrete Appl. Math.*, 53:275–285, 1994.

[143] M. Middendorf. Minimum broadcast time is np-complete for 3-regular planar graphs and deadline 2. In *Inf. Proc. Lett.*, volume 46, pages 281–287, 1993.

[144] S. Mitchell and S. Hedetniemi. A census of minimum broadcast graphs. *J. Combin. Inform. System Sci.*, 5:141–151, 1980.

[145] C.D. Morosan. *Studies of interconnection networks with applications in broadcasting.* PhD thesis, Concordia University, 2007.

[146] G.K. Osei and J.W. Thomson. The suppression of one rumor by another. *J. Appl. Prob.*, 14:127–134, 1977.

[147] J.H. Park and K.Y. Chwa. Recursive circulant: A new topology for multicomputers networks. In *Proc. Int. Symp. Parallel Architectures, Algorithms and Networks ISPAN '94*, pages 73–80, 1994.

[148] R. Peine, E.A. Stohr, R. Klasing, and B. Monien. Broadcasting in butterfly and debruijn networks. *Discrete Appl. Math.*, 53:183–197, 1994.

[149] A. Pelc. Fault-tolerant broadcasting and gossiping in communication networks. *Networks*, 28(3):143–156, 1996.

[150] D. Peters. *Bounds of Communication Problems in Interconnection Networks under a Linear Cost Model.* PhD thesis, Simon Fraser University, 1995.

[151] B. Pittel. On spreading a rumor. *SIAM J. Appl. Math.*, 47:213–223, 1987.

[152] A. Rapoport. Spread of information through a population with socio-structural bias. i. assumption of transitivity. *Bull. Math. Biophys.*, 15:523–533, 1953.

[153] A. Rapoport. Spread of information through a population with socio-structural bias. ii. various models with partial transitivity. *Bull. Math. Biophys.*, 15:535–546, 1953.

[154] A. Rapoport. Spread of information through a population with socio-structural bias. iii. suggested experimental procedures. *Bull. Math. Biophys.*, 16:75–81, 1954.

[155] A. Rapoport and L.I. Rebhun. On the mathematical theory of rumor spread. *Bull. Math. Biophys.*, 14:375–383, 1952.

[156] R. Ravi. Rapid rumor ramification: approximating the minimum broadcast time. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science, SFCS '94*, pages 202–213. IEEE Computer Society, 1994.

[157] J.F. Sacle. Lower bounds for the size in four families of minimum broadcast graphs. *Discrete Math.*, 150:359–369, 1996.

[158] P. Scheuermann and G. Wu. Heuristic algorithms for broadcasting in point-to-point computer networks. *IEEE Trans. Comput.*, 33(9):804–811, 1984.

[159] C. Schindelhauer. On the inapproximability of broadcasting time. In *Proc. of the 3rd International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, pages 226–237, 2000.

[160] B. Shao. *On k-broadcasting in graphs.* PhD thesis, Concordia University, Montreal, Canada, 2006.

[161] S. Skiena. Harary graphs. In *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*, pages 179–180, 1990.

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

[163] G. E. Weichenberg. High-reliability architectures for networks under stress. Master's thesis, Massachusetts Institute of Technology, 2003.

[164] J. Xiao and X. Wang. A research on minimum broadcast graphs. *Chinese J. Computers.*, 11:99–105, 1988.

[165] X. Xu. Broadcast networks of odd size and minimum broadcast network on 127 nodes. Master's thesis, Concordia University, 2003.

[166] J.G. Zhou and K.M. Zhang. A minimum broadcast graph on 26 vertices. *Appl. Math. Lett.*, 14:1023–1026, 2001.