A Simple Construction of Broadcast Graphs

Hovhannes A. Harutyunyan[1] and Zhiyuan Li[2](✉)

[1] Department of Computer Science and Software Engineering, Concordia University, Montreal, QC H3G 1M8, Canada
haruty@cs.concordia.ca

[2] Computer Science and Technology, BNU-HKBU United International College, Zhuhai 0086-519000, China
goliathli@uic.edu.hk

Abstract. Broadcasting is one of the basic primitives of communication in usual networks. It is a process of information dissemination in which one informed node of the network, called the originator, distributes the message to all other nodes of the network by placing a series of calls along the communication lines. The network is modeled as a graph. The broadcast time of a given vertex is the minimum time required to broadcast a message from the originator to all other vertices of the graph. The broadcast time of a graph is the maximum time required to broadcast from any vertex in the graph. Many papers have investigated the construction of minimum broadcast graphs, the cheapest possible broadcast network architecture (having the fewest communication lines) in which broadcasting can be accomplished as fast as theoretically possible from any vertex. Since this problem is very difficult, numerous papers give sparse networks in which broadcasting can be completed in minimum time from any originator. In this paper, we improve the existing upper bounds on the number of edges by constructing sparser graphs and by presenting a minimum time broadcast algorithm from any originator.

1 Introduction

In today's world, due to massively parallel processing communication between different processors in a large network is of main concern. One of the main problems of information dissemination in large networks is called *broadcasting*. Broadcasting is the message dissemination problem in a connected network in which initially one informed node, called the originator, must distribute the message to all other nodes of the network. In this paper, we investigate the problem of designing sparse networks in which fast broadcasting is possible. Our study focuses on the classical broadcast model under the following assumptions.

- The process is split into discrete time units.
- Initially, only one node, called the *originator* has the information.
- In each time unit, every informed node can only call at most one uninformed neighbor.
- The process terminates when every node in the network is informed.

D.-Z. Du et al. (Eds.): COCOON 2019, LNCS 11653, pp. 240–253, 2019.
https://doi.org/10.1007/978-3-030-26176-4_20

A network can be modeled as a graph. The *broadcast scheme* from the originator v in the graph G is a sequence of parallel calls from v. Each call, expressed by a directed edge, defines the sender and the receiver. The broadcast scheme also defines a directed spanning tree of G rooted at v, which is the *broadcast tree* of originator v. The minimum number of time units required to broadcast from v in the graph G is the *broadcast time* of vertex v and denoted by $b(G, v)$. And the maximum broadcast time from any vertex in G is the broadcast time of G, denoted by $b(G) = max\{b(G, u), u \in V\}$.

Note that in each time unit, the number of informed vertices is at most doubled because every vertex can only inform at most one vertex in a single time unit. Thus, $b(G) \geq \lceil \log n \rceil$ for an arbitrary graph G.

A graph G on n vertices is called a *broadcast graph* if $b(G) = \lceil \log n \rceil$. A broadcast graph with the minimum number of edges is called a *minimum broadcast graph* (mbg). The broadcast function $B(n)$ denotes the number of edges in mbg on n vertices. From the application perspective, mbgs are the "cheapest" graphs with the fastest broadcasting. In this research area, there are two major topics.

- Given a graph G and a vertex v, determine the optimal broadcast scheme originating from v in G and the value of $b(G, v)$. It is called the *broadcast time* problem.
- Given an natural number n, construct a minimum broadcast graph on n vertices and determine the value of $B(n)$, called the *minimum broadcast graph* problem.

The broadcast time problem in general is NP-hard [22]. The minimum broadcast graph problem is even more difficult. The exact value of $B(n)$ is known only for $n \leq 15$ [7], $n = 17$ [18], $n = 18, 19$ [3,23], $n = 20, 21, 22$ [17], $n = 26$ [20,24], $n = 27, 28, 29, 58, 61$ [20], $n = 30, 31$ [3], $n = 63$ [16], $n = 127$ [9], and $n = 1023, 4095$ [21]. Knödel graphs [5,15], hypercubes [7], and recursive circulant graphs [19] give the exact value for $n = 2^m$. Knödel graphs also give the exact value for $n = 2^m - 2$ [5,14].

Determining the exact values of $B(n)$ is very difficult. Thus, the line of research in this area studies the construction of broadcast graphs (not necessarily minimum), which implies upper bounds on $B(n)$. The broadcast graph constructions include compounding method [1,2,10,12], ad-hoc constructions [8,12], vertex addition methods [3,9,11,13], and vertex deletion methods [3,12].

In this paper, we follow the compounding construction and improve the general upper bound on $B(n)$ for even $2^m - 2^{\frac{m+3}{2}} < n \leq 2^m - 8$. The previously best bound for this interval is $B(n) \leq \frac{1}{2}(m-1)n$ [15]. Our improvement is no less than $\frac{n}{4}$. Section 2 reviews the important graphs being used in our construction. Section 3 gives a trivial compounding construction. Section 4 constructs our new graph. Section 5 proves that the newly constructed graph is a broadcast graph by demonstrating a broadcast algorithm from any originator. We also compare the new upper bound with the existing bounds.

2 Important Graphs

Definition 1. *A hypercube Q_k of dimension k, for any $k \in \mathbb{N}$ is a graph on 2^k vertices, where each vertex is represented by a binary string of length k and two vertices are adjacent if and only if the two strings have Hamming distance 1.*

In 1975, Knödel defined a class of broadcast graphs on even number of vertices [15]. We follow the equivalent definition given in [12,14].

Definition 2. *A Knödel graph $KG_n = (V, E)$ is defined for even values of n, where the vertex set is $V = \{v_0, v_1, v_2, ..., v_{n-1}\}$ and the edge set is $E = \{(v_x, v_y) | x + y \equiv 2^s - 1 \mod n, 1 \le s \le \lfloor \log n \rfloor\}$, where $0 \le x, y \le n - 1$.*

We recall the construction of a broadcast graph H from [12,14]. Let $s \ge 4$ and $1 \le t \le s-2$. A graph H on $2^s - 2^t$ vertices consists of 2^{t-1} copies of a Knödel graph $KG_{2^{s-t+1}-2}$. In each copy, the vertices are labeled from 0 to $2^{s-t+1} - 3$. All vertices with the same even label further form a hypercube Q_{t-1} of dimension $t-1$ on 2^{t-1} vertices. Thus, H has two types of vertices: half of the vertices are of degree $s-t$, and the other half of the vertices are of degree $s-1$. Thus, the number of edges of broadcast graph H is

$$\frac{1}{2}(\frac{n}{2}(s-t) + \frac{n}{2}(s-1)) = \frac{n}{2}(s - \frac{t+1}{2})$$

For more details of the construction and the broadcast algorithm, see [12].

3 Trivial Compounding

Before constructing the new broadcast graph, we introduce a compounding based on two existing broadcast graphs with similar number of vertices (in the same interval of two consecutive powers of 2). This compounding is trivial, straight forward, and well-known in this research area.

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two broadcast graphs on n_1 and n_2 vertices respectively, with $\lceil \log(n_1+n_2) \rceil = \lceil \log \max(n_1, n_2) \rceil + 1$, and $n_1 \ge n_2$. To construct a broadcast graph $T = (V, E)$, we connect every $v \in V_2$ with a distinct vertex in V_1. Since $n_1 \ge n_2$, there are some vertices in graph G_1 without any neighbor in graph G_2. We connect each of these vertices with an arbitrary vertex in graph G_2. This construction is similar to the broadcast graph construction in [6] and [4]. It is clear that the number of vertices in T is $n = n_1 + n_2$ and the number of edges is $e = |E_1| + |E_2| + \max(n_1, n_2) = |E_1| + |E_2| + n_1$. This construction adds n_1 edges between G_1 and G_2, such that every vertex in G_1 is adjacent to at least one vertex in G_2 and every vertex in G_2 is also adjacent to at least one vertex in G_1. Two vertices v and u connected by this construction are *trivially adjacent*. And v is u's *trivial neighbor* or vice versa. We introduce a new notation for the trivial compounding.

Definition 3. *Let $\mathfrak{T}$ be a binary operator on two graphs G_1 and G_2, constructing the trivially compounded graph T described above.*

Compounding of graphs G_1 and G_2 results in the graph T.

$$T = G_1 \mathfrak{T} G_2$$

We assume that A_1 and A_2 are two broadcast algorithms for two different arbitrarily selected originators in graphs G_1 and G_2, respectively. Algorithm 1 is the broadcast algorithm originating from any vertex in graph T.

Algorithm 1. Broadcast algorithm γ

Input : Graph $T = (V, E)$ constructed above and an arbitrary originator $v \in V$.
Output: The broadcast time, $b(T, v)$.

1 **begin**
2   $b(T, v) \longleftarrow 0$;
3   $Informed \longleftarrow \{v\}$;
4   $Uninformed \longleftarrow V \setminus \{v\}$; `/* Initialization */`
5   **if** $v \in G_1$ **then**
6     $u \longleftarrow v$'s neighbor in G_2;
7     $Informed \longleftarrow Informed \cup \{u\}$;
8     $Uninformed \longleftarrow Uninformed \setminus \{u\}$;
9     $b(T, v) \longleftarrow b(T, v) + 1$; `/* First time unit */`
10     $b(T, v) \longleftarrow b(T, v) + \max(A_1(G_1, v), A_2(G_2, u))$;
11   **else**
12     $u \longleftarrow v$'s neighbor in G_1;
13     $Informed \longleftarrow Informed \cup \{u\}$;
14     $Uninformed \longleftarrow Uninformed \setminus \{u\}$;
15     $b(T, v) \longleftarrow b(T, v) + 1$; `/* First time unit */`
16     $b(T, v) \longleftarrow b(T, v) + \max(A_1(G_1, u), A_2(G_2, v))$;
17   **end**
18 **end**

Observation 1. *The graph T constructed above is a broadcast graph.*

Proof. We proof the observation by verifying the broadcast time $b(T, v) = \lceil \log n \rceil$ from an arbitrary originator v of graph T. $b(T, v)$ is increased by one on line 9 or 15. Then, the algorithm calls A_1 and A_2. By the assumption, G_1 and G_2 are broadcast graphs, A_1 and A_2 are broadcast algorithms and return $\lceil \log n_1 \rceil$ and $\lceil \log n_2 \rceil$ broadcast times respectively. Thus, the return value is $b(T, v) = 1 + \max(\lceil \log n_1 \rceil, \lceil \log n_2 \rceil)$ given by line 10 or 16.

Again by the assumption, the number of vertices in T is n and $\lceil \log n \rceil = \lceil \log(n_1 + n_2) \rceil = \lceil \log \max(n_1, n_2) \rceil + 1$, which is $b(T, v)$. Thus, T is a broadcast graph. $\square$

4 A New Graph Construction

We are now ready to construct a new broadcast graph, IBC on n vertices for even $2^{m-1}+2 \leq n \leq 2^m-2$, where $m \geq 5$. IBC stands for *imbalanced compounding* because in the trivial compounding, the two broadcast graphs used in our construction are of different orders. The strategy of the construction is simple. First, we construct a sequence of the broadcast graphs H described in Sect. 2 (also in [12,14]). The total number of vertices in all graphs H is equal to n. And the broadcast time of any H is exactly one time unit more than its nearest successor in the sequence. Then, the trivial compounding combines all graphs H to form a connected graph. We give the details of IBC construction below.

Since $2^{m-1}+2 \leq n \leq 2^m-2$ is an even integer, n has the form

$$n = 2^m - 2^{k_1} - 2^{k_2} - \cdots - 2^{k_p}$$

where $1 \leq k_p < k_{p-1} < \cdots < k_2 < k_1 \leq m-2$ and $1 \leq p \leq m-2$.

We further decompose n in order to use the broadcast graph H under the assumption $k_1 \leq m-3$ and hence $2^m - 2^{m-2} + 2 \leq n \leq 2^m - 2$. Each H is on $n = 2^s - 2^t$ vertices, for some s and t, $s \geq 4$ and $1 \leq t \leq s-2$.

$$n = (2^{m-1} - 2^{k_1}) + (2^{m-2} - 2^{k_2}) + \cdots + (2^{m-p+1} - 2^{k_{p-1}}) + (2^{m-p+1} - 2^{k_p}) \tag{1}$$

Note that if $p = 1$, the value of $n = 2^m - 2^{k_1}$ and Eq. (1) has only the first term. The new graph IBC is the same as the graph H. No trivial compounding is required. This obvious case is also excluded from the following discussions. So, the range of k_i and p is slightly different. And we further assume that n is even, $n = 2^m - 2^{k_1} - 2^{k_2} - \cdots - 2^{k_p}$, $m \geq 5$, $1 \leq k_p < k_{p-1} < \cdots < k_2 < k_1 \leq m-3$, and $2 \leq p \leq k_1 \leq m-3$ without any further specification in the rest of the paper.
Under the assumption, each term in Eq. (1) is the number of vertices in H.

$$\begin{aligned} n &= \underbrace{2^{m-1} - 2^{k_1}}_{H_1} + \underbrace{2^{m-2} - 2^{k_2}}_{H_2} + \cdots + \underbrace{2^{m-p+1} - 2^{k_{p-1}}}_{H_{p-1}} + \underbrace{2^{m-p+1} - 2^{k_p}}_{H_p}, \\ &= \sum_{i=1}^{p-1} 2^{k_i-1}(2^{m-i-k_i+1} - 2) + 2^{k_p-1}(2^{m-p-k_p+2} - 2) \end{aligned} \tag{2}$$

where H_i is the graph H on $2^{m-i} - 2^{k_i}$ vertices for $1 \leq i \leq p$. Then, we recursively define the graph

$$T_i = \begin{cases} H_p & \text{if } i = p; \\ H_i \mathbin{\tau} T_{i+1} & \text{if } 1 \leq i \leq p-1. \end{cases} \tag{3}$$

The new graph

$$IBC = T_1$$

Counting the exact number of edges in IBC is not trivial (Fig. 1).

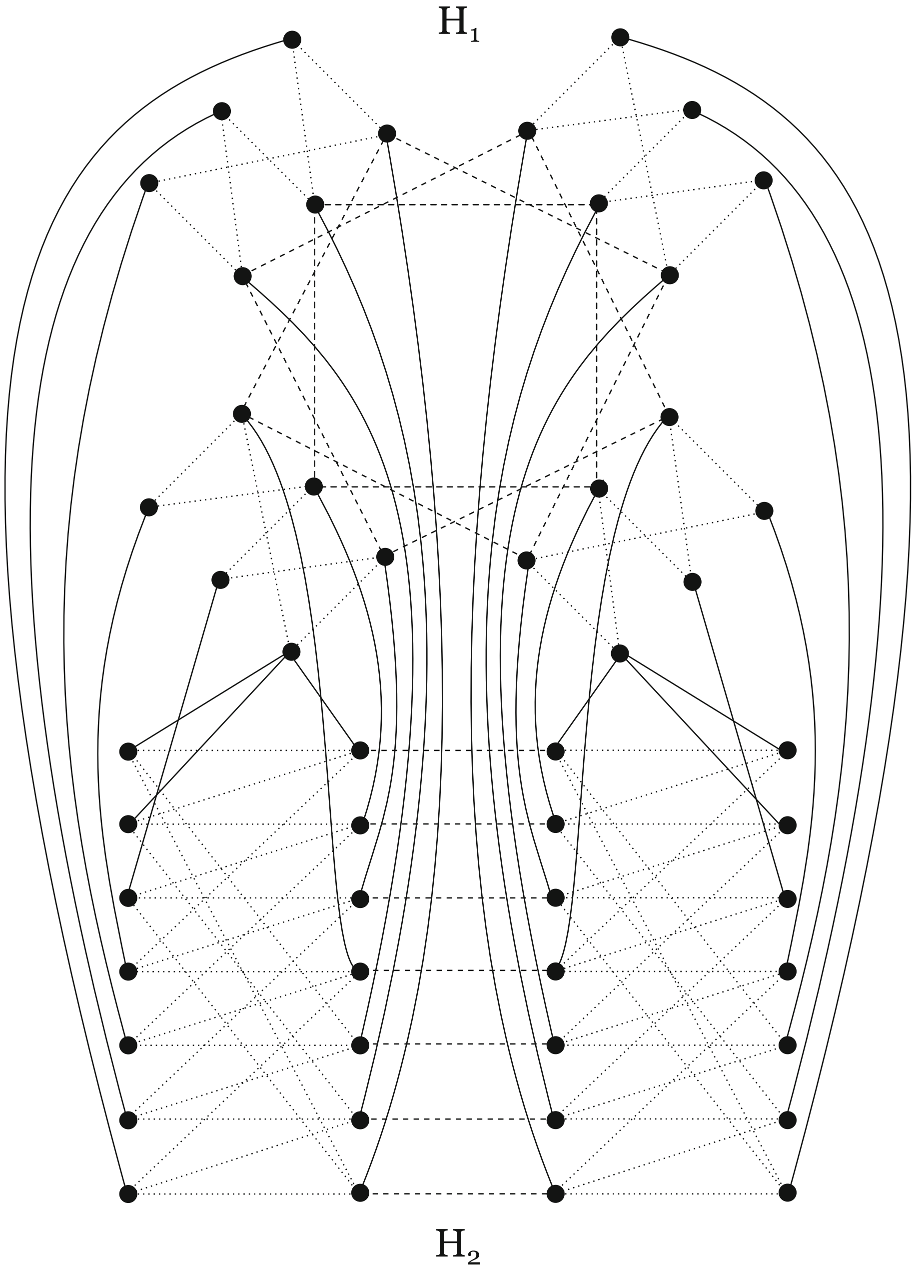

Fig. 1. An IBC on $n = 52$ vertices. $m = 6$, $p = 2$, $k_1 = 3$, and $k_2 = 2$. The 24 vertices on the top are in H_1, while the 28 vertices at the bottom belong to H_2. Dotted lines are the edges in Knödel graphs. Dashed lines are the compounded edges of H. And the solid lines are the trivial compounded edges.

Lemma 1. *Let IBC be a graph on n vertices constructed above, the number of edges in IBC is*

$$\begin{aligned} e = \frac{1}{2}(&\frac{1}{2}n(m-1) \\ &+ \frac{1}{2}\sum_{i=1}^{p-1}(2^{m-i} - 2^{k_i})(m-k_i) \\ &+ \frac{1}{2}(2^{m-p+1} - 2^{k_p})(m-k_p) \\ &+ \sum_{i=1}^{p-1}(2^{k_i} - \sum_{j=i+1}^{p} 2^j)) \end{aligned}$$

Proof. We prove the lemma by induction on p.

Base Case: When $p = 2$, IBC consists of H_1 on $2^{m-1} - 2^{k_1}$ vertices, H_2 on $2^{m-1} - 2^{k_2}$ vertices, and a trivial compounding between them. For H_1, $\frac{1}{2}(2^{m-1} - 2^{k_1})$ vertices are of degree $m-1-k_1$ and the same number of vertices are of degree $m-2$. Similarly for H_2, $\frac{1}{2}(2^{m-1} - 2^{k_2})$ vertices are of degree $m-1-k_2$ and the same number of vertices are of degree $m-2$. Thus, the total sum of all degrees of graphs H_1 and H_2 before counting the edges of trivial compounding is

$$\begin{aligned} &\frac{1}{2}(2^{m-1} - 2^{k_1})(m-k_1-1) + \frac{1}{2}(2^{m-1} - 2^{k_1})(m-2) \\ &\quad + \frac{1}{2}(2^{m-1} - 2^{k_2})(m-k_2-1) + \frac{1}{2}(2^{m-1} - 2^{k_2})(m-2) \\ &= \frac{1}{2}n(m-2) + \frac{1}{2}(2^{m-1} - 2^{k_1})(m-k_1-1) + \frac{1}{2}(2^{m-1} - 2^{k_2})(m-k_2-1) \end{aligned}$$

Then the trivial compounding adds one to the degree of each vertex in both graphs H_1 and H_2. Since H_2 has more vertices than H_1 and each vertex must have a trivial neighbor, then, this will add $(2^{m-1} - 2^{k_2}) - (2^{m-1} - 2^{k_1})$ to the sum of the degrees. Thus, the total number of edges is

$$\begin{aligned} e = \frac{1}{2}(&\frac{1}{2}n(m-2) \\ &+ \frac{1}{2}(2^{m-1} - 2^{k_1})(m-1-k_1) \\ &+ \frac{1}{2}(2^{m-1} - 2^{k_2})(m-1-k_2) \\ &+ n + 2^{k_1} - 2^{k_2}) \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2}(\frac{1}{2}n(m-2) \\
&\quad + \frac{1}{2}(2^{m-1} - 2^{k_1})(m-1-k_1) \\
&\quad + \frac{1}{2}(2^{m-1} - 2^{k_2})(m-1-k_2) \\
&\quad + \frac{n}{2} + \frac{1}{2}(2^{m-1} - 2^{k_1}) + \frac{1}{2}(2^{m-1} - 2^{k_2}) + 2^{k_1} - 2^{k_2}) \\
&= \frac{1}{2}(\frac{1}{2}n(m-1) \\
&\quad + \frac{1}{2}(2^{m-1} - 2^{k_1})(m-k_1) \\
&\quad + \frac{1}{2}(2^{m-1} - 2^{k_2})(m-k_2) \\
&\quad + 2^{k_1} - 2^{k_2})
\end{aligned}
$$

Inductive Hypothesis: Assume that Lemma 1 is true for $p = r$, where $2 < r \leq m-3$, which means

$$
\begin{aligned}
e = \frac{1}{2}(\frac{1}{2}n(m-1) \\
&\quad + \frac{1}{2}\sum_{i=1}^{r-1}(2^{m-i} - 2^{k_i})(m-k_i) \\
&\quad + \frac{1}{2}(2^{m-r+1} - 2^{k_r})(m-k_r) \qquad (4) \\
&\quad + \sum_{i=1}^{r-1}(2^{k_i} - \sum_{j=i+1}^{r} 2^j)) \qquad (5)
\end{aligned}
$$

Inductive Step: When $p = r+1$, the last term of Eq. (2) is further decomposed into $2^{m-r} - 2^{k_r} + 2^{m-r} - 2^{k_{r+1}}$, representing H_p and H_{p+1}. After constructing the whole graph on n vertices, half of the vertices in H_r and H_{r+1} are of degree $m-1$; the other half of the vertices in H_r are of degree $m - k_r$; and the other half of the vertices in H_{r+1} are of degree $m - k_{r+1}$. Thus, Eq. (4) is split into $\frac{1}{2}(2^{m-r} - 2^{k_r})(m-k_r) + \frac{1}{2}(2^{m-r} - 2^{k_{r+1}})(m - k_{r+1})$.

The trivial compounding further adds $\frac{1}{2}(2^{k_{r+1}} - 2^{k_r})$ edges to the compounding of H_r and H_{r+1}. And since each T_i contains H_{r+1}, each term in the summation on Eq. (5) has one additional term $-2^{k_{r+1}}$. Thus, when $p = r+1$,

$$\begin{aligned} e = \frac{1}{2}(\frac{1}{2}n(m-1) \\ &+ \frac{1}{2}\sum_{i=1}^{r}(2^{m-i} - 2^{k_i})(m-k_i) \\ &+ \frac{1}{2}(2^{m-r} - 2^{k_{r+1}})(m-k_{r+1}) \\ &+ \sum_{i=1}^{r}(2^{k_i} - \sum_{j=i+1}^{r+1} 2^j)) \end{aligned}$$

which completes the proof. □

5 Broadcast Algorithm

Assume the broadcast algorithm for the graph H from any originator v in [12] is $C(H, v)$. Algorithm 2 is a $\lceil \log n \rceil$ time broadcast algorithm for the graph IBC. The If statement on line 4 is the termination condition for the recursion. When this statement is executed, the recursion calls $C(H, v)$ and halts. The cases in step 7 and 12 are distinguished because the vertex v can be either in H_{t+1} or T_{t+2}. If v is in H_{t+1}, u is the trivial neighbor in T_{t+2}. In the next iteration, the algorithm broadcasts from v in H_{t+1} and from u in T_{t+2}. And it is vice versa if v is in T_{t+2}.

Theorem 1. *Algorithm γ' completes the broadcast from any originator in graph IBC on n vertices in $\lceil \log n \rceil$ iterations and returns $b(IBC, v) = \lceil \log n \rceil$.*

Proof. We prove the theorem by induction on p, which is the number of additions in Eq. (2) (the number of trivial compoundings) plus 1, or the number of 0's minus 1 in the binary representation of n.

Base Case: When $p = 2$, IBC consists of H_1 on $2^{m-1} - 2^{k_1}$ vertices and H_2 on $2^{m-1} - 2^{k_2}$ vertices. Assume the originator v is in H_1 without loss of generality, the condition on line 8 is true and algorithm γ' executes line 9 to line 12 in the first iteration. Thus, v informs its trivial neighbor u in H_2 in the first time unit. Then, algorithm γ' calls $C(H_1, v)$ and $C(H_2, u)$, the broadcast algorithm in [12] from v and u in H_1 and H_2 respectively in the second iteration. Both of the algorithms return $m - 1$ broadcast time. Thus, the total broadcast time is $m - 1 + 1 = m = \lceil \log n \rceil$.

Inductive Hypothesis: Assume when $p = r$, γ' returns $b(IBC, v) = \lceil \log n \rceil$ for $2 < r \leq m-3$. This implies that in the last $(r-1)$-th iteration, line 5 returns $b(IBC, v) \longleftarrow C(H_r, v)$. Since H_r is on $2^{m-r+1} - 2^{k_r}$ vertices,

$$C(H_r, v) = m - r + 1 \tag{6}$$

Algorithm 2. Broadcast algorithm γ'

Input : A graph $IBC = (V, E)$, an originator $v \in V$, and the broadcast time t from the previous iteration, equals to 0 initially.

Output: Broadcast time $b(IBC, v)$

1 **begin**
2 $\quad Informed \longleftarrow \{v\}$;
3 $\quad Uninformed \longleftarrow V \setminus \{v\}$;
4 $\quad$ **if** $v \in H_p$ **then**
5 $\quad\quad b(IBC, v) \longleftarrow C(H_p, v)$; `/* The termination condition. And` $C(H, v)$ `is the broadcast scheme on` H `from` v `given in [12]. */`
6 $\quad$ **end**
7 $\quad$ **if** $v \in H_{t+1}$ **then**
8 $\quad\quad u \longleftarrow v's$ trivial neighbor in T_{t+2};
9 $\quad\quad Informed \longleftarrow Informed \cup \{u\}$;
10 $\quad\quad Uninformed \longleftarrow Uninformed \setminus \{u\}$;
11 $\quad\quad b(IBC, v) \longleftarrow t + 1 + \max(\gamma'(T_{t+2}, u, t + 1), C(H_{t+1}, v))$;
12 $\quad$ **else**
13 $\quad\quad u \longleftarrow v's$ trivial neighbor in H_{t+1};
14 $\quad\quad Informed \longleftarrow Informed \cup \{u\}$;
15 $\quad\quad Uninformed \longleftarrow Uninformed \setminus \{u\}$;
16 $\quad\quad b(IBC, v) \longleftarrow t + 1 + \max(\gamma'(T_{t+2}, v, t + 1), C(H_{t+1}, u))$;
17 $\quad$ **end**
18 $\quad$ **return** $b(IBC, v)$
19 **end**

Inductive Step: When $p = r + 1$, the only difference is that the algorithm will run one extra iteration. In the second last $(r - 1)$-th iteration, if the increment on $b(IBC, v)$ is δ (assume $v \in T_{r+1}$ and $u \in H_r$ without loss of generality) line 16 will call the algorithm again and

$$\delta = 1 + max(\gamma'(T_{r+1}, u, r), C(H_r, v))$$

In the last iteration, the algorithm returns $C(H_{r+1}, u) = m - r$ because H_{r+1} is on $2^{m-r} - 2^{k_{r+1}}$ vertices. And we know $C(H_r, v) = m - r$ since H_r is on $2^{m-r} - 2^{k_r}$ vertices. Thus, $\delta = m - r + 1$ which is the same as the Eq. (6). Therefore, the total broadcast time $b(IBC, v) = m = \lceil \log n \rceil$. $\square$

Summarizing Theorem 1 and Lemma 1, we obtain our upper bound on $B(n)$.

Theorem 2.

$$\begin{aligned} B(n) \leq \frac{1}{2}(\frac{1}{2}n(m-1) & \\ &+ \frac{1}{2}\sum_{i=1}^{p-1}(2^{m-i} - 2^{k_i})(m - k_i) \\ &+ \frac{1}{2}(2^{m-p+1} - 2^{k_p})(m - k_p) \\ &+ 2^{k_{p-1}} - 2^{k_p} \\ &+ \sum_{i=1}^{p-1}(2^{k_i} - \sum_{j=i+1}^{p} 2^j)) \end{aligned}$$

Let NB be the new upper bound given by Theorem 2 and

$$OB = \frac{1}{2}(m-1)n$$

in [15]. The comparison shows that NB is a better upper bound and

$$\begin{aligned} OB - NB &\geq \frac{1}{2}(2^{m-1} - 2^{m-3} - 2^{m-3}) + \frac{1}{2}(2^{m-2} + 2^{m-3}) \\ &= 2^{m-2} + 2^{m-4} \end{aligned}$$

The tedious calculation is given in the Appendix.

6 Conclusion

In this paper, we constructed a new broadcast graph based on the broadcast graph H defined in [12]. The new construction applies the trivial compounding on a sequence of broadcast graph H and improves the upper bound on $B(n)$ for $2^m - 2^{\frac{m+3}{2}} < n \leq 2^m - 8$. In the future, we can try to compound the sequence of H in a better way instead of just using the trivial compounding. This will reduce the number of edges because the trivial compounding is too simple and adds to many redundant edges.

Another work can also be done in the future is the comparison between the new upper bound NB and the bound OB' given by [10], which is the best bound on $B(n)$ when $2^{m-1} + 1 \leq n \leq 2^m - 2^{\frac{m+3}{2}}$. Currently, we only know that NB is better than OB' in some of the intervals, but the intervals are not continuous. Therefore, we can further investigate the comparison in the future.

Appendix: Comparison

By our assumption $k_p < \cdots < k_1 \leq m-3$, k_i is strictly larger than k_{i+1}. Then in general $k_i \leq m-i-2$ for $1 \leq i \leq p$. Similarly, since $1 \leq k_p < \cdots < k_1$, $p-i+1 \leq k_i$ for $1 \leq i \leq p$. Thus, $p-i+1 \leq k_i \leq m-i-2$, where $1 \leq i \leq p$.

$$
\begin{aligned}
&OB - NB \\
&= \frac{1}{2}n(m-1) - \frac{1}{2}(\frac{1}{2}n(m-1) + \frac{1}{2}\sum_{i=1}^{p-1}(2^{m-i} - 2^{k_i})(m-k_i) \\
&\qquad + \frac{1}{2}(2^{m-p+1} - 2^{k_p})(m-k_p) + \sum_{i=1}^{p-1}(2^{k_i} - \sum_{j=i+1}^{p} 2^{k_j})) \\
&= \frac{1}{4}n(m-1) - \frac{1}{4}\sum_{i=1}^{p-1}(2^{m-i} - 2^{k_i})(m-k_i) \\
&\qquad - \frac{1}{4}(2^{m-p+1} - 2^{k_p})(m-k_p) - \frac{1}{2}\sum_{i=1}^{p-1}(2^{k_i} - \sum_{j=i+1}^{p} 2^{k_j})) \\
&= \frac{1}{4}n(m-1) - \frac{1}{4}\sum_{i=1}^{p-1}(2^{m-i} - 2^{k_i})(m-k_i) \\
&\qquad - \frac{1}{4}(2^{m-p+1} - 2^{k_p})(m-k_p) - \frac{1}{2}(2^{k_1} - \sum_{i=2}^{p-1}(i-2)2^{k_i} - (p-1)2^{k_p})
\end{aligned}
$$

By substituting $n = \sum_{i=1}^{p-1}(2^{m-i} - 2^{k_i}) + 2^{m-p+1} - 2^{k_p}$,

$$
\begin{aligned}
&= \frac{1}{4}\sum_{i=1}^{p-1}(2^{m-i} - 2^{k_i})(m-1) + \frac{1}{4}(2^{m-p+1} - 2^{k_p})(m-1) - \frac{1}{4}\sum_{i=1}^{p-1}(2^{m-i} - 2^{k_i})(m-k_i) \\
&\qquad - \frac{1}{4}(2^{m-p+1} - 2^{k_p})(m-k_p) - \frac{1}{2}(2^{k_1} - \sum_{i=2}^{p-1}(i-2)2^{k_i} - (p-1)2^{k_p}) \\
&= \frac{1}{4}\sum_{i=1}^{p-1}(2^{m-i} - 2^{k_i})(k_i+1) + \frac{1}{4}(2^{m-p+1} - 2^{k_p})(k_p+1) \\
&\qquad - \frac{1}{2}2^{k_1} + \frac{1}{2}\sum_{i=2}^{p-1}(i-2)2^{k_i} + \frac{1}{2}(p-1)2^{k_p}
\end{aligned}
$$

Since $k_i \geq 1$,

$$
\begin{aligned}
&\geq \frac{1}{2}\sum_{i=1}^{p-1}(2^{m-i} - 2^{k_i}) + \frac{1}{2}(2^{m-p+1} - 2^{k_p}) \\
&\quad - \frac{1}{2}2^{k_1} + \frac{1}{2}\sum_{i=2}^{p-1}(i-2)2^{k_i} + \frac{1}{2}(p-1)2^{k_p} \\
&= \frac{1}{2}(2^{m-1} - 2^{k_1} - 2^{k_1}) + \frac{1}{2}\sum_{i=2}^{p-1}(2^{m-i} - 2^{k_i}) + \frac{1}{2}(2^{m-p+1} - 2^{k_p}) \\
&\quad + \frac{1}{2}\sum_{i=2}^{p-1}(i-2)2^{k_i} + \frac{1}{2}(p-1)2^{k_p} \\
&= \frac{1}{2}(2^{m-1} - 2^{k_1} - 2^{k_1}) + \frac{1}{2}(2^{m-2} + 2^{m-3} + 2^{m-p-2}) \\
&\quad + \frac{1}{2}\sum_{i=2}^{p-1}(i-2)2^{k_i} + \frac{1}{2}(p-1)2^{k_p} \\
&\geq \frac{1}{2}(2^{m-1} - 2^{k_1} - 2^{k_1}) + \frac{1}{2}(2^{m-2} + 2^{m-3})
\end{aligned}
$$

As $k_1 \leq m-3$,

$$
\begin{aligned}
OB - NB &\geq \frac{1}{2}(2^{m-1} - 2^{m-3} - 2^{m-3}) + \frac{1}{2}(2^{m-2} + 2^{m-3}) \\
&= 2^{m-2} + 2^{m-4}
\end{aligned}
$$

Thus, our new upper bound is a better upper bound on $B(n)$ when $2^m - 2^{\frac{m+3}{2}} < n \leq 2^m - 8$.

References

1. Averbuch, A., Shabtai, R.H., Roditty, Y.: Efficient construction of broadcast graphs. Discrete Appl. Math. **171**, 9–14 (2014)
2. Bermond, J.-C., Fraigniaud, P., Peters, J.G.: Antepenultimate broadcasting. Networks **26**(3), 125–137 (1995)
3. Bermond, J.-C., Hell, P., Liestman, A.L., Peters, J.G.: Sparse broadcast graphs. Discrete Appl. Math. **36**(2), 97–130 (1992)
4. Chau, S.-C., Liestman, A.L.: Constructing minimal broadcast networks. J. Comb. Inf. Syst. Sci. **10**, 110–122 (1985)
5. Dinneen, M.J., Fellows, M.R., Faber, V.: Algebraic constructions of efficient broadcast networks. In: Mattson, H.F., Mora, T., Rao, T.R.N. (eds.) AAECC 1991. LNCS, vol. 539, pp. 152–158. Springer, Heidelberg (1991). https://doi.org/10.1007/3-540-54522-0_104
6. Farley, A.M.: Minimal broadcast networks. Networks **9**(4), 313–332 (1979)
7. Farley, A.M., Hedetniemi, S., Mitchell, S., Proskurowski, A.: Minimum broadcast graphs. Discrete Math. **25**(2), 189–193 (1979)

8. Grigni, M., Peleg, D.: Tight bounds on mimimum broadcast networks. SIAM J. Discrete Math. **4**(2), 207–222 (1991)
9. Harutyunyan, H.A.: An efficient vertex addition method for broadcast networks. Internet Math. **5**(3), 211–225 (2008)
10. Harutyunyan, H.A., Li, Z.: A new construction of broadcast graphs. In: Govindarajan, S., Maheshwari, A. (eds.) CALDAM 2016. LNCS, vol. 9602, pp. 201–211. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29221-2_17
11. Harutyunyan, H.A., Li, Z.: Broadcast graphs using new dimensional broadcast schemes for Knödel graphs. In: Gaur, D., Narayanaswamy, N.S. (eds.) CALDAM 2017. LNCS, vol. 10156, pp. 193–204. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-53007-9_18
12. Harutyunyan, H.A., Liestman, A.L.: More broadcast graphs. Discrete Appl. Math. **98**(1), 81–102 (1999)
13. Harutyunyan, H.A., Liestman, A.L.: Upper bounds on the broadcast function using minimum dominating sets. Discrete Math. **312**(20), 2992–2996 (2012)
14. Khachatrian, L.H., Harutounian, H.S.: Construction of new classes of minimal broadcast networks. In: International Conference on Coding Theory, Dilijan, Armenia, pp. 69–77 (1990)
15. Knödel, W.: New gossips and telephones. Discrete Math. **13**(1), 95 (1975)
16. Labahn, R.: A minimum broadcast graph on 63 vertices. Discrete Appl. Math. **53**(1–3), 247–250 (1994)
17. Maheo, M., Saclé, J.-F.: Some minimum broadcast graphs. Discrete Appl. Math. **53**(1–3), 275–285 (1994)
18. Mitchell, S., Hedetniemi, S.: A census of minimum broadcast graphs. J. Comb. Inf. Syst. Sci. **5**, 141–151 (1980)
19. Park, J.-H., Chwa, K.-Y.: Recursive circulant: a new topology for multicomputer networks. In: International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN 1994), pp. 73–80. IEEE (1994)
20. Saclé, J.-F.: Lower bounds for the size in four families of minimum broadcast graphs. Discrete Math. **150**(1–3), 359–369 (1996)
21. Shao, B.: On k-broadcasting in graphs. Ph.D. thesis, Concordia University (2006)
22. Slater, P.J., Cockayne, E.J., Hedetniemi, S.T.: Information dissemination in trees. SIAM J. Comput. **10**(4), 692–701 (1981)
23. Xiao, J., Wang, X.: A research on minimum broadcast graphs. Chin. J. Comput. **11**, 99–105 (1988)
24. Zhou, J., Zhang, K.: A minimum broadcast graph on 26 vertices. Appl. Math. Lett. **14**(8), 1023–1026 (2001)