New Upper Bound on *m*-Time-Relaxed *k*-Broadcast Graphs

Amir Averbuch, Irete Peeri, and Yehuda Roditty
School of Computer Sciences, Tel Aviv University, Ramat Aviv, Tel Aviv 69978, Israel

Broadcasting is a process in which an individual has an item of information which needs to be transmitted to all of the members in a network (which is viewed as a connected graph). *k-broadcasting* is a variant of broadcasting in which each processor can transmit the message to up to *k* of its neighbors. Another variant of broadcasting is the concept of *m-time-relaxed broadcasting*, where we allow *m* additional time units, enabling one to decrease the number of edges in the communication network. The combination of *m*-time-relaxed and *k*-broadcasting is studied in this article. The results shown here are an improvement on the upper bound obtained by Harutyunyan and Liestman, Discrete Math 262 (2003), 149–157, by constructing a connected graph, which admits *m*-time-relaxed *k*-broadcasting for all *n*, $n > (k+1)^{m+2}, m > 1$ and has fewer edges than the graph presented in Harutyunyan and Liestman, Discrete Math 262 (2003), 149–157. **NETWORKS, Vol. 000(00), 000–000 2017**

Keywords: broadcasting; *k*-broadcasting; *m*-time-relaxed broadcasting; *m*-time-relaxed k-broadcast graphs; *k*-nomial tree; Q^s-cube

1. INTRODUCTION

Broadcasting is a one-to-all information dissemination process. In fact, simple broadcasting is a process in which one individual, or originator, has an item of information, or message, which needs to be communicated, or transmitted, to all members (processors) of a network. Broadcasting is subject to the following rules:

1. A processor may send a message to only an adjacent processor.
2. At a given time each processor will perform in one of the following ways:
   (a) receive a message,
   (b) send a message to some neighbors,
   (c) be idle.

Received May 2017; accepted May 2017
Correspondence to: Y. Roditty; e-mail: jr@post.tau.ac.il
DOI 10.1002/net.21749
Published online 00 00 2017 in Wiley Online Library (wileyonlinelibrary.com).

More formally, we can view the communication network as a finite, connected, undirected graph $G = (V, E)$ on n vertices, where the vertices are considered as processors and each edge which connects two vertices is assumed to be a direct communication link between these vertices. Then, we define broadcasting from a vertex v (*the originator*) as transmitting a message from v to every vertex in $V \setminus \{v\}$ using the above rules. This problem, which appeared in 1970 (see [18]), is a variation of the gossiping problem (see [18]).

For basic graph theoretical definitions, one may see [9] or [30].

We define the *broadcast number* of $v \in V$, denoted by $b(v)$, as the minimum time required to broadcast one message from v. The *broadcast time* of G is defined as $b(G) = \max\{b(v) | v \in V(G)\}$. Let $b(n)$ denote the minimal message broadcast time $b(G)$ over all graphs G with n vertices. A graph G is said to be *a broadcast graph* if $b(G) = b(n)$.

The problem of broadcasting in a general graph, namely, the problem of determining $b(v)$ for an arbitrary vertex in an arbitrary graph, was proved by Johnson (see [18]) to be NP-complete. Conversely, in a tree with equal weights it happens to be solvable in linear time [28]. Recently a generalization of broadcasting in trees was obtained [2], where positive weights were assigned to the edges or the vertices of the tree. Other results concerning broadcasting in various topologies are available for complete graphs [4, 6, 7, 12], planar graphs [19], grids in several dimensions and multiple messages [10, 11, 22, 24, 26, 29], trees [15, 25, 28], minimal broadcast graphs [1, 8, 16, 17], weighted trees [2, 13], and various problems [20, 21].

The *broadcast number* $B(n)$ 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. The notion of *m-Relaxed Broadcast Graphs (m-RBG)*, appearing in [27], is a generalization of *1-RBG* [8], and was motivated by exploring the sparsest possible graphs in which broadcasting can be accomplished in slightly more than the optimal time of $\lceil \log_2 n \rceil$.

Denote by $B^m(n)$ the number of edges in the sparsest possible graph on n vertices in which broadcasting can be accomplished in $\lceil \log_2 n \rceil + m$ steps. Such a graph is called an *m-Relaxed Minimum Broadcast Graph (m-RMBG)*.

We follow the definition of k-broadcasting presented in [14] (which was defined earlier, see [18] and [23]), which is a generalization of the above notion of broadcasting, in which a processor can send the message to at most k neighbors in one time unit.

The *k-broadcast time of vertex v* is defined in a similar way as the broadcast time $b(v)$, enabling k-broadcasting, and is denoted $b_k(v)$. A lower bound on the k-broadcast time is $b_k(v) \geq \lceil \log_{k+1} n \rceil$ since at each time unit the number of informed vertices is multiplied by at most $k+1$. A *k-broadcast graph* is a graph G on n vertices where $b_k(G) = \lceil \log_{k+1} n \rceil$, that is a graph where k-broadcasting from any vertex is accomplished within at most $\lceil \log_{k+1} n \rceil$ time units.

The *k-broadcast function* $B_k(n)$ is the minimum number of edges in any k-broadcast graph on n vertices. A *minimum k-broadcast graph* is a k-broadcast graph on n vertices having $B_k(n)$ edges.

The next broadcasting concept, introduced in [8] (but only for $m = 1$), combines both concepts defined above. Namely m-time-relaxed k-broadcasting, which is k-broadcasting within $\lceil \log_{k+1} n \rceil + m$, $m \geq 0$, time units. The minimum number of edges in an m-relaxed k-broadcast graph on n vertices is denoted $B_k^m(n)$. A *minimum m-relaxed k-broadcast graph* (*m*-R*k*BG) is one with $B_k^m(n)$ edges. Recently, new results have been obtained for (0-R1BG) in [1] and [17].

Harutyunyan and Liestman [16] described two methods to construct m-time-relaxed k-broadcast graphs and provided upper bounds on $B_k^m(n)$:

Theorem 1.1.

$$B_k^m(n) \leq \min \left\{ n + \frac{1}{2}(k+1)^{\lceil \log_{k+1} n \rceil - m} (k \lceil \log_{k+1} n \rceil - km - 2), (n-1) + \lfloor n/m \rfloor \right\}, \quad 1 \leq m \leq \lceil \log_{k+1} n \rceil$$

The case $k = 1$, $m \geq 1$, was dealt with by Shastri [27], and a significantly better bound was obtained later by Averbuch et al. [3]. The bound obtained in [3] might be observed as almost "best possible" in the sense that for $m = 1, 2, 3$ and $16 \leq n \leq 65$ the values obtained from their bound almost meet the ones obtained by Shastri [27] in which he used some direct methods (ad hoc) different from those used for his bounds.

In the next theorem, we provide an improvement of the bound obtained in Theorem 1.1, by using some ideas from both papers [1] and [3].

Theorem 1.2.

(i) $B_k^m(n) \leq n\left(1 + \frac{1}{(k+1)^{m-1}}\right) - \Theta\left(\left(\frac{n}{(k+1)^{m-1}}\right)^{\frac{2}{3}}\right)$, $n > (k+1)^{m+1}, n = (k+1)^t$

(ii) $B_k^m(n) \leq A_{m,n} - \left\lfloor \frac{(k+1)^t - n}{(k+1)^{t-c} - 1} \right\rfloor ((k+1)^{t-c-m+1} - k(t-c) - k(m-1) + 1) - ((k+1)^t - n)$, $n > (k+1)^{m+1}$, $(k+1)^{t-1} < n < (k+1)^t$ *$A_{m,n}$ is the bound in (i) with $n = (k+1)^t$ and $c = \left\lfloor \frac{2}{3}(t-m+1) \right\rfloor$.*

(iii) $B_k^m(n) = n - 1, \quad n \leq (k+1)^{m+1}$

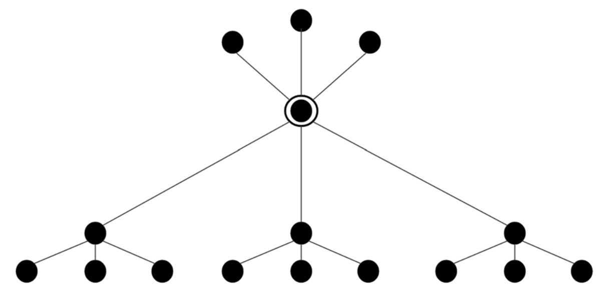

FIG. 1. The tree T_{k+1}^s where $k = 3$, $s = 2$.

This article is organized as follows: Section 2 presents some necessary definitions and preliminary results. In Section 3, we present the proof of Theorem 1.2.

2. *m*-TIME-RELAXED *k*-BROADCASTING GRAPHS

2.1. *Construction of* $(k+1)$*-Nomial Trees*

The basic underlying structure used in this article is the $(k+1)$-nomial tree. This is a generalization of the *Binomial-Tree* introduced in [25]. This tree enables k-broadcasting from the root in $\lceil \log_{k+1} n \rceil$ time units where $n = (k+1)^t$.

The $(k+1)$-nomial tree is constructed as follows:

Definition 2.1. *$(k+1)^s$ -nomial tree on $(k+1)^s$ vertices, denoted by T_{k+1}^s, is constructed recursively. The tree T_{k+1}^0 is a single vertex which is the root. T_{k+1}^s is constructed by connecting the roots of k copies of T_{k+1}^{s-1} to the root of another copy of T_{k+1}^{s-1}. This vertex is the root of T_{k+1}^s.*

In Figure 1, we give an example of such a tree with $k = 3, s = 2$.

Each vertex v_y^x is denoted by two labels: x, y.

x—a string $(x_1 x_2 \ldots x_i)$ where $1 \leq i \leq s$ and $1 \leq x_j \leq s$, $1 \leq j \leq i$.
- *$|x|$ determines the distance of the vertex from the root. This label is not unique.*

y—a $(k+1)$-ary string of length s.
- *That is, a numeric string representing an integer in base $k+1$. This label is unique.*

The root is denoted $v_{(0\ldots0)}$.

It has ks children which are denoted

$$v_{(0\ldots01)}^{(1)}, \ldots, v_{(0\ldots0k)}^{(1)}, v_{(0\ldots010)}^{(2)}, \ldots, v_{(0\ldots0k0)}^{(2)}, \ldots, v_{(10\ldots0)}^{(s)}, \ldots, v_{(k,0\ldots0)}^{(s)}.$$

Regarding the labelling of their children, only the first $k(s-1)$ *vertices, labeled* $v^{(1)}_{(0...01)}, \ldots, v^{(1)}_{(0...0k)}, v^{(2)}_{(0...010)}, \ldots, v^{(2)}_{(0...0k0)}, \ldots, v^{(s-1)}_{(01...0)}, \ldots, v^{(s-1)}_{(0k...0)}$, *have children vertices.*

Their labelling is as follows:

For vertices labeled $v^{(1)}_{(0...01)}, \ldots, v^{(1)}_{(0...0k)}$,

their children are labeled:

$$v^{(1,1)}_{(0...011)}, \ldots, v^{(1,1)}_{(0...0k1)}, \ldots, v^{(1,s-1)}_{(10...01)}, \ldots, v^{(1,s-1)}_{(k0...01)}, \ldots,$$
$$v^{(1,1)}_{(0...01k)}, \ldots, v^{(1,1)}_{(0...0kk)}, \ldots, v^{(1,s-1)}_{(10...0k)}, \ldots, v^{(1,s-1)}_{(k0...0k)}$$

For vertices labeled $v^{(2)}_{(0...010)}, \ldots, v^{(2)}_{(0...0k0)}$,

their children are labeled:

$$v^{(2,1)}_{(0...0110)}, \ldots, v^{(2,1)}_{(0...0k10)}, \ldots, v^{(2,s-2)}_{(10...010)}, \ldots, v^{(2,s-2)}_{(k0...010)}, \ldots,$$
$$v^{(2,1)}_{(0...01k0)}, \ldots, v^{(2,1)}_{(0...0kk0)}, \ldots, v^{(2,s-2)}_{(10...0k0)}, \ldots, v^{(2,s-2)}_{(k0...0k0)}$$

and so on, so that the leaves are labeled:

$$v^{(s)}_{(10...0)}, \ldots, v^{(s)}_{(k0...0)},$$
$$v^{(1,s-1)}_{(10...0k)}, \ldots, v^{(1,s-1)}_{(k0...0k)}, \ldots, v^{(s-1,1)}_{(1k0...0)}, \ldots, v^{(s-1,1)}_{(kk0...0)}, \ldots,$$
$$v^{(1,..,1)}_{(1...1)}, \ldots, v^{(1,..,1)}_{(1...1k)}, \ldots, v^{(1,..,1)}_{(k...k1)}, \ldots, v^{(1,..,1)}_{(k...k)}$$

whereas the superscript label $(1 \ldots 1)$ *is of length s.*

In order to simplify the notation, we shall denote the tree T^s_{k+1} by T^s.

2.2. *k-Broadcasting in* (k + 1)*-Nomial Trees*

The following algorithm, which is the k-broadcast scheme in the optimal $(k+1)^s$-nomial tree, shows that $b(v_{(0...0)}) \le \lceil \log_{k+1} n \rceil$, and as $b(v_{(0...0)}) \ge \lceil \log_{k+1} n \rceil$, it follows that $b(v_{(0...0)}) = \lceil \log_{k+1} n \rceil$.

1. At $t=1$, the root, $v_{(0...0)}$, broadcasts to the vertices $v^{(1)}_{(0...01)},\ldots,v^{(1)}_{(0...0k)}$.
2. At $t=2$, the root broadcasts to vertices $v^{(2)}_{(0...010)},\ldots,v^{(2)}_{(0...0k0)}$ and the vertices $v^{(1)}_{(0...01)},\ldots,v^{(1)}_{(0...0k)}$ broadcast to their children $v^{(1,1)}_{(0...011)},\ldots,v^{(1,1)}_{(0...0k1)},\ldots,v^{(1,1)}_{(0...01k)},\ldots,v^{(1,1)}_{(0...0kk)}$, respectively.
3. Each vertex $v^{(y)}_{(x_1x_2...x_{i-1})}$, $1 \le i \le s$, $0 \le y \le (k+1)^s - 1$ that receives the message at time t', will broadcast to k uninformed neighbors, labeled $v^{(x_1x_2...x_{i-1}1)}_{(y)}$ at time $t'+1$, to neighbors labeled $v^{(y)}_{(x_1x_2...x_{i-1}2)}$ at time $t'+2$ and so on.
   Larger subtrees are proceeded before smaller one. This is ensured by the labelling of the vertices.

When broadcasting from any vertex (not necessarily the root), the *k-broadcast time* is at most $\log_{k+1} n + s$. That is, the maximum time needed to broadcast from any vertex to the root, which is at most s, and then from the root to the whole tree which is $\log_{k+1} n$.

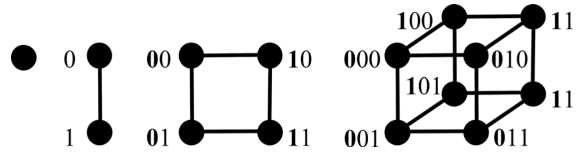

FIG. 2. $Q_{(2)^s}$-cubes, $k=1$, $0 \le s \le 3$.

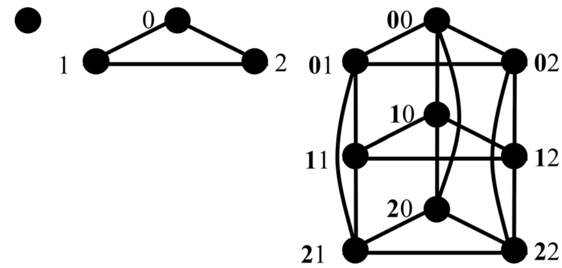

FIG. 3. $Q_{(3)^s}$-cubes, $k=2$, $0 \le s \le 2$.

In order to prove Theorem 1.2, the following basic underlying structure is essential, namely, the $Q_{(k+1)^s}$-cube. This is a generalization of the hypercube topology where $k=1$. The $Q_{(k+1)^s}$-cube enables k-broadcasting from any vertex in $\log_{k+1} n$ time units where $n = (k+1)^s$, that is in s time units. This section shows how to construct these cubes.

Definition 2.2. *The* $Q_{(k+1)^s}$ *-cube with* $(k+1)^s$ *vertices is constructed recursively. The cube* $Q_{(k+1)^0}$ *has a single vertex.* $Q_{(k+1)^s}$ *is constructed by connecting vertices with the same label (defined below) in k + 1 copies of* $Q_{(k+1)^{s-1}}$ *-cubes.*

Each vertex is denoted by a (k + 1)-ary string of length s by the following method:

The k + 1 vertices in $Q_{(k+1)^1}$ *are labeled:* $v_{(0)}, v_{(1)}, \ldots, v_{(k)}$.

The $(k+1)^2$ *vertices in* $Q_{(k+1)^2}$ *are labeled:*

$$v_{(00)}, v_{(01)}, \ldots, v_{(0k)}, v_{(10)}, v_{(11)}, \ldots, v_{(1k)}, \ldots, v_{(k0)},$$
$$v_{(k1)}, \ldots, v_{(kk)}.$$

The $(k+1)^s$ *vertices in* $Q_{(k+1)^s}$ *are labeled by prefixing the labels of the* $(s-1)(k+1)$ *vertices in each* $Q_{(k+1)^{s-1}}$ *-cube with the label* $0, 1, \ldots, k$. *That is,* $v_{(0x_{s-1})}, v_{(1x_{s-1})}, \ldots, v_{(kx_{s-1})}$, *where* $0 \le x_{s-1} \le (k+1)^{s-1} - 1$, x_{s-1} *is a (k + 1)-ary string of length s – 1.*

Vertices with the same x_{s-1} *label are connected in the* $Q_{(k+1)^s}$ *-cube.*

Observe that by the above construction, two vertices are connected by an edge if and only if they differ by one bit. For example, in Q_{k+1} all vertices are connected.

Figures 2 and 3 show examples of $Q_{(2)^s}$-cubes and $Q_{(3)^s}$-cubes. For simplicity, the vertices are labeled by x, where $0 \le x \le (k+1)^s - 1$, x is a $(k+1)$-ary string, and the length of x is s. Digits in bold represent the addition of transferring from a k-cube to a $(k+1)$-cube.

2.3. *Properties of* $Q_{(k+1)^s}$*-Cubes*

To simplify the notation $Q_{(k+1)^s}$, we shall denote the cube $Q_{(k+1)^s}$ by Q^s.

1. Q^s-cubes are *ks-regular* graphs. Thus, $|E(Q^s)| = \frac{s}{2}(k+1)^s k$ for $k \geq 1$.
2. $\Delta(Q^s) = d_{Q^s}(v_x) = ks$, where $0 \leq x \leq (k+1)^s - 1$, $k \geq 1$.
   The Q^s-cube is constructed by connecting ***k*** copies of a Q^{s-1}-cube via vertices with the same labels, so that $d_{Q^s}(v_x) = d_{Q^{s-1}}(v_x) + k$ where v_x denotes any vertex in the Q^s-cube. Therefore, the *degree of each vertex* in Q^s is ks.
3. $\text{diam}(Q^s) = s$.
   The Q^1-cube is a polygon with $k+1$ vertices in which all the vertices are connected. Therefore, $\text{diam}(Q^1) = 1$.
   The construction of Q^s yields $\text{diam}(Q^s) = \text{diam}(Q^{s-1}) + 1$. Therefore, the *diameter* of (Q^s) is s.

2.4. *k-Broadcasting in* Q^s*-Cubes*

In the next algorithm, we demonstrate a k-broadcasting scheme in the Q^s-cube. It shows that $b(v_{(0\ldots0)}) = \log_{k+1} n$.

1. The root, $v_{(0\ldots0)}$, at $t=1$, broadcasts to the vertices $v_{(10\ldots0)}, v_{(20\ldots0)}, \ldots, v_{(k0\ldots0)}$.
   Recall that vertices are labeled by a $(k+1)$-ary string of length s.
2. At $t=2$, each of the vertices informed at $t \leq 1$, $v_{(x0\ldots0)}$, broadcasts to the vertices $v_{(x10\ldots0)}, v_{(x20\ldots0)}, \ldots, v_{(xk0\ldots0)}$, $0 \leq x \leq k$.
3. The vertices $v_{(x_1 10\ldots0)}, v_{(x_1 20\ldots0)}, \ldots, v_{(x_1 k0\ldots0)}$ at $t=s$ broadcast to the vertices $v_{(x_2 1)}, v_{(x_2 2)}, \ldots, v_{(x_2 k)}$ where
   x_1 denotes all $k+1$ base numbers between 0, and $(k+1)^{s-2} - 1$.
   x_2 denotes all $k+1$ base numbers between 0 and $(k+1)^{s-1} - 1$.
   The labeling ensures that all of the vertices are informed.

As Q^s-cubes are symmetric, broadcasting can start from any vertex.

The number of informed vertices at each time unit is multiplied by k. From this it follows that the *k-broadcasting time* from any vertex in a Q^s-cube is $\log_{k+1} n$.

Figures 4–6 show examples of k-broadcasting for $k=1$, 2, 3. For simplicity, the vertices are not labeled. For $k=1$, 2, the labeling is like that shown in Figures 2 and 3.

3. PROOF OF THEOREM 1.2

The structure used in the proof of Theorem 1.2 is a m-Time-Relaxed Minimal k-Broadcast Graph (m-RkBG) for $k > 1$. The case $k=1$ was dealt with already in [3].

The additional number of time units is denoted by m and $\lceil \log_{k+1} n \rceil$ is denoted by t.

Proof. The proof is done by construction of an m-RkBG in each of the cases and demonstrating the appropriate broadcast scheme in each case. ∎

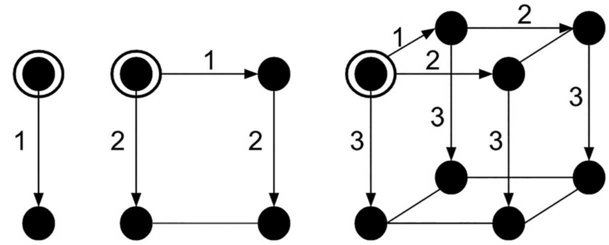

FIG. 4. Broadcasting in Q^s-cubes, $k=1$, $1 \leq s \leq 3$.

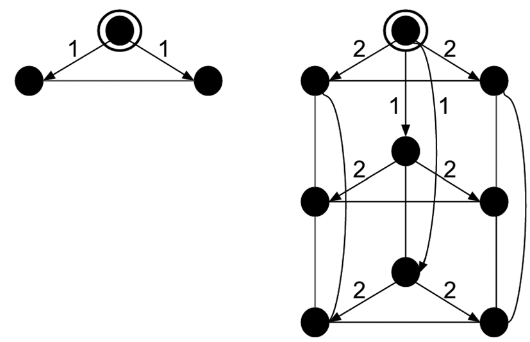

FIG. 5. Broadcasting in Q^s-cubes, $k=2$, $1 \leq s \leq 2$.

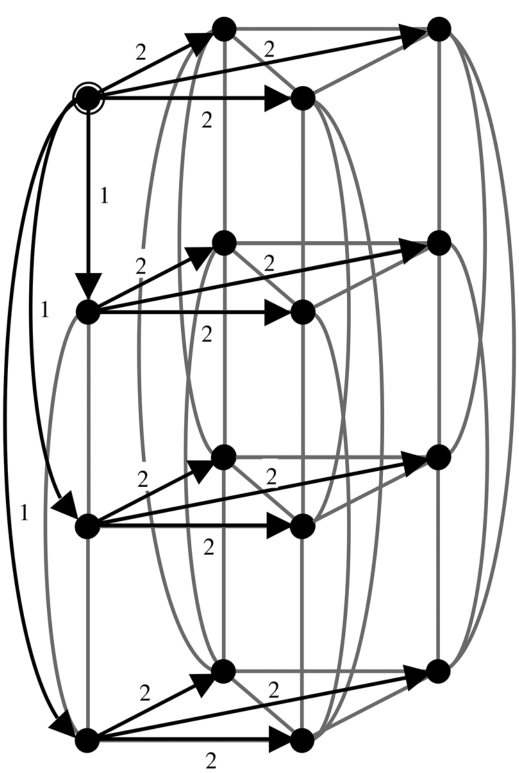

FIG. 6. Broadcasting in Q^2-cubes, $k=3$, $s=2$.

Construction

Case 1. $n > (k+1)^{m+1}$, $n = (k+1)^t$

The m-RkBG is constructed as follows:

Take $(k+1)^c$ copies of the T^{t-c} tree denoted $T^{(i)}$, $0 \leq i \leq (k+1)^c - 1$. Their vertices are labeled as explained in Section 2, using the base 10 notation, with an additional index preceding the vertex label which denotes the tree. Therefore, their originators are $v_{(0)(0)}$, $x_{(1)(0)}$,..., $x_{((k+1)^c)(0)}$.

Create a Q^c-cube using their originators.

The value of c will be determined according to the optimal construction. Now additional edges need to be added to the trees:

Decompose each $T^{(i)}$ into vertex disjoint isomorphic copies of a T^{m-1} tree (assuming $(m-1 \leq t-c)$).

There are exactly $\frac{(k+1)^{t-c}}{(k+1)^{m-1}}$ such copies.

Each originator $v_{(i)(0)}$ is joined to one of the originators in each T^{m-1} subtree which is at distance at least 2 from $v_{(i)(0)}$.

Hence, by this process there are $\frac{(k+1)^{t-c}}{(k+1)^{m-1}} - [k(t-c) - k(m-1)] - 1$ additional edges added per tree.

Since $(k+1)^c$ is the number of trees, $[(k+1)^{t-c} - 1]$ is the number of edges in each tree and $\frac{c}{2}(k+1)^c k$ is the number of edges in a Q^c- cube we obtain the following upper bound

$$\begin{aligned}
B_k^m(n) &\leq (k+1)^c\Bigg[(k+1)^{t-c} - 1 + \frac{(k+1)^{t-c}}{(k+1)^{m-1}} \\
&\qquad - [k(t-c) - k(m-1)] - 1\Bigg] + \frac{c}{2}(k+1)^c k \\
&= n\left(1 + \frac{1}{(k+1)^{m-1}}\right) \\
&\qquad - (k+1)^c\left[k(t-m+1) + 2 - \frac{3}{2}ck\right] \qquad (1)
\end{aligned}$$

Whence, the derivative of the right side of (1), with respect to c, is

$$-(k+1)^c\left[\ln(k+1)\left[k(t-m+1) + 2 - \frac{3}{2}ck\right] - \frac{3}{2}k\right]$$

Therefore, the minimum is achieved when

$$c = \frac{2}{3}(t-m+1) + \frac{4}{3k} - \frac{1}{\ln(k+1)}$$

$\frac{4}{3k} - \frac{1}{\ln(k+1)}$ converges to 0.1, therefore it follows that

$$c = \left\lfloor \frac{2}{3}(t-m+1) \right\rfloor,$$

where

$$m - 1 \leq t - c \qquad (2)$$

Thus, substituting the value of c obtained in (2) into (1) yields (i) of Theorem 1.2, as required.

Case 2. $n > (k+1)^{m+1},\ (k+1)^{t-1} < n < (k+1)^t$

The m-RkBG is constructed as follows:

Take $(k+1)^c$ copies of the T^{t-c} tree (labeled $T^{(i)}, 0 \leq i \leq (k+1)^c - 1$) and create a Q^c- cube using their originators in the same way as Case 1, with $n = (k+1)^t$ and c obtained in the first case.

Now delete vertices as needed to obtain the exact number of vertices which is denoted by n.

First delete whole trees $T^{(i)}$ without their originators $v_{(i)(0)}$, as needed, as long as $(k+1)^t - n \geq |T^{(i)}| - 1$.

Next delete vertices from the same tree, starting with the leaves.

Hence, the total number of edges removed from the bound obtained in (1) is

$$\begin{aligned}
&\left\lfloor \frac{(k+1)^t - n}{(k+1)^{t-c} - 1} \right\rfloor \Bigg[(k+1)^{t-c} - 1 + \frac{(k+1)^{t-c}}{(k+1)^{m-1}} \\
&\qquad - (k(t-c) - k(m-1) + 1)\Bigg] \\
&\qquad + (k+1)^t - n - [(k+1)^{t-c} - 1]\left\lfloor \frac{(k+1)^t - n}{(k+1)^{t-c} - 1} \right\rfloor \\
&= \left\lfloor \frac{(k+1)^t - n}{(k+1)^{t-c} - 1} \right\rfloor \Bigg[\frac{(k+1)^{t-c}}{(k+1)^{m-1}} \\
&\qquad - (k(t-c) - k(m-1) + 1)\Bigg] + (k+1)^t - n \qquad (3)
\end{aligned}$$

where $\left\lfloor \frac{(k+1)^t - n}{(k+1)^{t-c} - 1} \right\rfloor \left((k+1)^{t-c} - 1 + \frac{(k+1)^{t-c}}{(k+1)^{m-1}} - (k(t-c) - k(m-1) + 1)\right)$ is the total number of edges deleted with the deletion of whole trees, and $(k+1)^t - n - [(k+1)^{t-c} - 1]\left\lfloor \frac{(k+1)^t - n}{(k+1)^{t-c} - 1} \right\rfloor$ is the number of edges deleted due to the remaining vertices left to be deleted.

Case 3. $n \leq (k+1)^{m+1}$

The m-RkBG is a T^m tree and has $n-1$ edges. To conform with the vertex labelling in Case 1 and Case 2, the vertices are labeled $v_{(0)(i)},\ 0 \leq i \leq (k+1)^m - 1$.

Broadcast Scheme in the various cases:

Cases 1,2

Case i: The originator is $v \notin \left\{v_{(i)(0)} | 0 \leq i \leq (k+1)^c - 1\right\}$ Assume $v \in T^{(j)}$.

1. v transmits to $v_{(j)(0)}$.
   This takes at most m time units due to the additional edges in each tree.
2. $v_{(j)(0)}$ k-broadcasts to all of the vertices in the Q^c-cube using the first vertex label.
   This takes another c time units.

3. $v_{(i)(0)}$, $0 \le i \le (k+1)^c - 1$, k-broadcasts to all of the vertices in each $T^{(i)}$ as explained in Section 2.
   This takes at most another $t-c$ time units.

Therefore, in total, broadcasting is accomplished within at most $t+m$ time units.

Case ii: The originator is $v \in \left\{v_{(i)(0)} | 0 \le i \le (k+1)^c - 1\right\}$ Assume $v \in T^{(j)}$.

1. $v_{(j)(0)}$ k-broadcasts to all of the vertices in the Q^c-cube using the first vertex label.
   This takes c time units.
2. $v_{(i)(0)}$, $0 \le i \le (k+1)^c - 1$, k-broadcasts to all of the vertices in each $T^{(i)}$ as explained in Section 2.
   This takes at most another $t-c$ time units.

Therefore, in total, broadcasting is accomplished within at most t time units.

Case 3. Case i: The originator is $v \neq v_{(0)(0)}$

1. v transmits to $v_{(0)(0)}$.
   As the depth of the tree is m this takes at most m time units.
2. $v_{(0)(0)}$ k-broadcasts to the vertices in T^m according to the broadcast scheme in Section 2.
   This is completed within at most t additional time units.

Therefore, in total, broadcasting is accomplished within at most $t+m$ time units.

Case ii: The originator is $v = v_{(0)(0)}$

1. $v_{(0)(0)}$ k-broadcasts to the vertices in T^m according to the broadcast scheme in Section 2.
   This is completed within at most t time units.

4. CONCLUDING REMARKS

1. In their article, Harutyunyan and Liestman [16] use a constant r, $r = \lceil \log_{k+1} n \rceil - m$, to determine the ratio between the Q-cube and the $(k+1)$-nomial trees and uses conventional $(k+1)$-nomial trees (without additional edges). Since we determine the ratio by calculating the optimal construction, $c = \left\lfloor \frac{2}{3}(t - m + 1) \right\rfloor$, our result, Theorem 1.2, is an improvement for all n, $n > (k+1)^{m+2}$, and is equal to these in [16] for n in the range: $(k+1)^{m-1} \le n \le (k+1)^{m+2}$.
   For $m = 1$, it is difficult to accurately compare the two bounds results. Specific results are: (1) $k = 2$, $2 \le n \le 4000$, 60% of the results are improved; (2) $k = 3$, $4 \le n \le 5000$, only 40% of the results are improved.
2. One can see from Theorem 1.2 that $B_k^m(n) < 2n$, regarding the values of m and k. Conversely, obviously $n \le B_k^m(n)$.

ACKNOWLEDGMENT

The authors would like to thank the referees for their valuable remarks which enabled them to improve the presentation of the paper.

REFERENCES

[1] A. Averbuch, R. Hollander- Shabtai, and Y. Roditty, Efficient construction of broadcast graphs, Discrete Appl Math 171 (2014), 9-14.

[2] A. Averbuch, Y. Roditty, and B. Shoham, Computation of broadcasting multiple messages in a positive weighted tree, J Comb Math Comb Comput 35 (2000), 161–184.

[3] A. Averbuch, Y. Rodity, and B. Shoham, Construction of minimum time-relaxed broadcasting communication networks, J Comb Math Comb Comput 43 (2002), 123–134.

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

[5] J.C. Bermond, P. Hell, L. Liestman, and J. Peters, Sparse broadcast graphs, Discrete Appl Math 36 (1992), 97–130.

[6] P. Chinn, S. Hedetniemi, and S. Mitchell, Multiple message broadcasting in complete graphs, Proc 10th SE Conf Combinatorics, Graph Theory and Computing, Utilitas Math, Winnipeg, 1979, 251–260.

[7] E.J. Cockayne and A. Thomason, Optimal multi-message broadcasting in complete graphs, Proceedings of the 11th SE Conference on Combinatorics, Graph Theory and Computing, Utilitas Math., Winnipeg, 1980, pp. 181–199.

[8] M. Grigni and D. Peleg, Tight bounds on minimum broadcast networks, SIAM J Discrete Math 4 (1991), 207–222.

[9] F. Harary, Graph theory, Addison-Wesley, London, U.K., 1969.

[10] H.A. Harutyunyan, Minimum multiple message broadcast graphs, Networks 47 (2000), 1–15.

[11] H.A. Harutyunyan, Multiple message broadcasting in modified Knödel graph, SOROCCO, Laquila, Italy, 2000, pp. 157–165.

[12] H.A. Harutyunyan and S. Kamali, Optimum broadcasting in complete weighted-vertex graphs, SOFSEM, Špindlerův Mlýn, Czech Republic, 2010, pp. 489–502.

[13] H.A. Harutyunyan and S. Kamali, Efficient broadcast trees for weighted vertices, Optimum broadcasting in complete weighted-vertex graphs, Discrete Appl Math 216 (2017), 598–608.

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

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

[16] H.A. Harutyunyan and A.L. Liestman, On the monotonicity of the broadcast function, Discrete Math 262 (2003), 149–157.

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

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

[19] P. Hell and K. Seyffarth, Broadcasting in planar graphs, Australas J Comb 17 (1998), 309–318.

[20] S. Khuller and Y.A. Kim, Broadcasting in heterogeneous networks, Algorithmica 48 (2007), 1–21.

[21] S. Khuller, Y.A. Kim, and Y. Wan, Broadcasting on networks of workstations, Algorithmica 57 (2010), 848–868.

[22] Oh-Heum Kwon and Kyung-Yong Chwa, Multiple messages broadcasting in communication networks, Networks 26 (1995), 253–261.

[23] E. Lazard, Broadcasting in DMA-bounded degree graphs, Discrete Appl Math 37/38 (1992), 387–400.

[24] A.L. Liestman, T. Shermer, and M. Suderman, Broadcasting multiple messages in hypercubes (preliminary version), ISPAN (2000), 274–281.

[25] A. Proskurowski, Minimum broadcast trees, IEEE Trans Comput 5 (1981), 363–366.

[26] Y. Roditty and B. Shoham, On broadcasting multiple messages in a d-dimensional grid, Discrete Appl Math 75 (1997), 277–284.

[27] A. Shastri, Time-relaxed broadcasting in communication networks, Discrete Appl Math 83 (1998), 263–278.

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

[29] F.L. Van Scoy and J.A. Brooks, Broadcasting multiple messages in a grid, Discrete Appl Math 53 (1994), 321–336.

[30] D. West, Introduction to graph theory, Simon & Schuster, Upper Saddle River, NJ, 1996.