Sparse Hypercube
— A Minimal k-Line Broadcast Graph —

Satoshi Fujita
Department of Electrical Engineering
Hiroshima University
Email: fujita@se.hiroshima-u.ac.jp

Arthur M. Farley
Department of Computer Science
University of Oregon
Email: art@cs.oregon.edu

Abstract

This paper proposes a method for reducing the maximum degree of vertices in graphs that maintain optimal broadcast time when a vertex can call a vertex at distance at most k during any time unit. In the proposed method, we eliminate edges from binary n-cubes. We show that, by this approach, the maximum degree of a vertex can be reduced from n to at most $(2k-1)\lceil \sqrt[k]{n-k} \rceil$*, where* $2 \leq k < n$*, which asymptotically achieves the lower bound for any constant k.*

1. Introduction

The study of communication networks and communication algorithms is key to the design of high performance parallel and distributed computer systems. Over the past several decades, a number of efforts have focused on finding effective topologies to serve as the underlying structure of communication networks. Results have included the design and analysis of cycles, k-ary n-cubes, star graphs, and de Bruijn graphs [6, 8, 9]. The availability of efficient algorithms for various communication tasks is an important requirement of a good network design. In particular, the capability of efficient **broadcasting**, i.e., the one-to-all communication task, is a desirable feature. Let us model the topology of a communication network by a graph $G = (V(G), E(G))$, where vertices $V(G)$ represent sites and edges $E(G)$ represent lines of the network. If each vertex can call at most one other vertex during any time unit, then a broadcast in graph G takes at least $\lceil \log_2 |V(G)| \rceil$ time units (i.e., the number of informed vertices can at most double during each time unit). We refer to graphs in which broadcast can be completed in this minimum time as minimal broadcast graphs [3].

Selection of a communication model is an important factor when defining communication networks. The simplest model is the *store-and-forward* model, where a vertex can communicate with a particular neighbor at any point. In [4], Farley proposed the line communication model, which is a simplified model of *circuit switching* or *wormhole routing* [2]. The line communication model can be described as follows: for any $u, v \in V(G)$, vertex u transmits a message by calling any other vertex v during a given time unit; the call succeeds if it shares no edge with other calls during the same time unit and if v is not the intended recipient of another call at that time. Note that a vertex can switch through several calls that do not collide on edges. It was shown that under this line communication model, broadcast can be completed in the minimum, $\lceil \log_2 |V(G)| \rceil$ time units in any connected graph G, regardless of the location of originating vertex [4].

The line communication model suggests an interesting new direction for research, namely limiting the length of a call (i.e., the number of edges occupied by a call) by some constant k (≥ 1) and characterizing networks in which broadcast can be completed in minimum time regardless of the location of originating vertex. Such networks will be referred to as **minimal k-line broadcast graphs** (k**-mlbg**, for short). This call-length constraint is reasonable from an implementation standpoint; long-distance calls utilize more network resources, can be more difficult to complete, are more likely to fail, and can cause bottlenecks for other competing communication processes. In this paper, we consider the problem of finding small, in terms of maximum vertex degree, k-mlbg's. We propose methods for constructing such small graphs for any number of vertices N such that $N = 2^n$ for some $n \geq 0$, and for any given constant k of call length.

The paper is organized as follows. In Section 2, we provide some basic definitions and derive basic properties of our k-line communication model and corresponding minimal broadcast graphs. In Section 3, we consider the n-cube as a basis for k-line broadcasting, removing edges to create *sparse* cubes that allow minimum-time broadcasting for the case when $k = 2$. Section 4 extends the construction and routing to the case of $k \geq 3$. It is shown that for any $n > k \geq 1$, there is a k-mlbg G of order $N = 2^n$ such that $\Delta(G) \leq (2k-1)\left\lceil \sqrt[k]{\log_2 N - k}\,\right\rceil$. Section 5 concludes the paper.

2. Preliminaries

Let $G = (V(G), E(G))$ be an (undirected) graph with N vertices, where $V(G)$ and $E(G)$ represent the set of vertices and the set of edges in graph G, respectively. For any $u, v \in V(G)$, let $dist_G(u, v)$ denote the length of the shortest path connecting u and v in graph G, where $dist_G(u, v)$ is called the **distance** between u and v in G. Throughout the paper, we assume for any distinct vertices $u, v \in V(G)$, $dist_G(u, v) \leq N - 1$; i.e., G is connected.

Definition 1 (k-line communication) Let k be a positive integer. A **k-line communication** is a communication model defined as follows: (1) all communications proceed step by step according to the global clock; (2) at any given time, each vertex can call at most one other vertex at distance no more than k; and (3) a call succeeds if it shares no edge nor receiver with another call placed at the same time. □

Note that 1-line communication is equivalent to the standard store-and-forward communication model and an $(N-1)$-line communication is equivalent to the general line communication model with calls of (essentially) unbounded length [4].

Definition 2 (minimum-time k-line broadcast) A broadcast scheme in graph G is said to be a **minimum-time k-line broadcast scheme** for G if it requires $\lceil \log_2 |V(G)| \rceil$ time units under the k-line communication. □

Remark 1 *For any graph G and for any integer $k \geq 1$, a minimum-time k-line broadcast scheme for G is a minimum-time $(k+1)$-line broadcast scheme for G.* □

Definition 3 (k-mlbg) A graph G is said to be a **minimal k-line broadcast graph** (**k-mlbg**, for short) if, for any vertex $v \in V(G)$, there exists a minimum-time k-line broadcast scheme for G from vertex v. Let $\mathcal{G}_k$ denote the class of k-mlbg's. □

Remark 2 $\forall k \geq 1$, $\mathcal{G}_k \subseteq \mathcal{G}_{k+1}$. □

In this paper, we consider the class $\mathcal{G}_k$ for $1 \leq k \leq N-1$. In the literature, several researchers have studied class $\mathcal{G}_1$ [5, 3, 7], and obtained results including methods for constructing graphs that realize minimum number of edges for certain values of number of vertices N, and approximate the minimum for others. On the other end of the scale, the basic result on line broadcasting obtained in [4] implies that all connected graphs are members of the class $\mathcal{G}_{N-1}$. These results define the limits for the problem we discuss here.

If we measure the goodness of a k-mlbg G by overall number of edges in the graph, then a rather simple graph attains the lower bound. Consider the tree consisting of one central vertex of degree $N-1$ and $N-1$ leaves (such a tree is commonly referred to as a star). As can be easily verified, it is a graph with fewest edges that is a member of $\mathcal{G}_k$ for any $k \geq 2$. Therefore, throughout this paper, we measure the goodness of the constructed graphs G in terms of the maximum degree over all vertices in G, denoted by $\Delta(G)$. The smaller $\Delta(G)$ is, the better the graph is considered to be. As such, we are concerned with minimizing $\Delta(G)$.

Before concluding this section, we want to show several theorems concerned with an upper and lower bounds on the maximum degree of certain k-mlbg's (proofs are omitted here).

Theorem 1 (An upper bound for large k) *For any $k \geq 2\left\lceil \log_2\left(\frac{N+2}{3}\right)\right\rceil$, there is a k-mlbg G with N vertices and $\Delta(G) \leq 3$.*

Theorem 2 (A lower bound) *Let $G \in \mathcal{G}_k$ be a graph with $N = 2^n$ vertices. For any $k = 2, 3, 4$, $\Delta(G) \geq \left\lceil \sqrt[k]{\log_2 N}\,\right\rceil$, and for any $5 \leq k < n$, $\Delta(G) \geq \left\lceil \sqrt[k]{\left(\frac{1}{3}\right)\log_2 N + 1}\,\right\rceil + 1$.*

3. Sparse Hypercubes for $k = 2$

In this and next sections, we propose a method for constructing small k-mlbg's with $N = 2^n$ vertices for any $n \geq k+1$. We do this by "eliminating" edges from binary n-cubes (or simply n-cube), which is known to have a minimum-time broadcasting property under the 1-line communication model. Given two bit strings $u = u_n u_{n-1} \ldots u_1$ and $v = v_n v_{n-1} \ldots v_1$ of length n each and an integer $i \in \{1, 2, \ldots, n\}$, let us denote $v = \oplus_i u$ if $u_j = v_j$ for all $j \neq i$ and $u_j = 1 - v_j$ for $j = i$. Binary n-cube Q_n is a Cayley graph defined as $V(Q_n) = \{0,1\}^n$ and $E(Q_n) = \{\{u, v\} \mid v = \oplus_i u,\ 1 \leq i \leq n\}$. Note that by definition, $\Delta(Q_n) = n$ and $|E(Q_n)| = n \cdot 2^{n-1}$. In what follows, for any $1 \leq i \leq n$ and for any $u \in \{0,1\}^n$, we call $\{u, \oplus_i u\}$ the **i-dimensional edge** of u, where we refer to the least significant bit as dimension one and the most significant bit as dimension n.

In this section, we consider 2-line broadcasting (i.e., $k = 2$) and construct graphs of order $N = 2^n$ that we call **sparse hypercubes**. An extension to the case of $k \geq 3$ is discussed in the next section. As will be shown in this and next sections, sparse hypercubes, as subgraphs of n-cubes, significantly reduce the maximum degree from n $(= \Delta(Q_n))$ to at most $(2k-1)\lceil \sqrt[k]{n-k}\,\rceil$, for any given n $(= \log_2 N) > k \geq 2$. Recall that in Theorem 1, we have proved that for any $k \geq 2\lceil \log_2 (\frac{N+2}{3})\rceil$, $\Delta(G) \leq 3$.

Let m be a positive integer less than n. Consider an m-cube Q_m with vertex set $V(Q_m) = \{0,1\}^m$. The basic step of our construction is to "label" vertices in $V(Q_m)$ by a set C of labels to satisfy the following requirement (we call it Condition A): $\forall u \in V(Q_m)$,

$$\{f(u)\} \cup \{f(v) \mid \{u,v\} \in E(Q_m)\} \;=\; C, \qquad (1)$$

where $f(u)$ denotes the label assigned to vertex u by f. In other words, Condition A requests that, for any label $c \in C$, the subset of vertices which is assigned label c by f forms a **dominating set** for G.[1]

Example 1 Vertices in $V(Q_2)$ can be labeled by $\{c_1, c_2\}$ as $f(00) = f(11) = c_1$ and $f(01) = f(10) = c_2$, to satisfy Condition A, and vertices in $V(Q_3)$ can be labeled by $\{c_1, c_2, c_3, c_4\}$ as $f(000) = f(111) = c_1$, $f(001) = f(110) = c_2$, $f(010) = f(101) = c_3$, and $f(011) = f(100) = c_4$ to satisfy Condition A. □

For any $m \geq 1$, there is at least one such labeling f of $V(Q_m)$ satisfying Condition A. In fact, a trivial labeling which labels all vertices by the same label fulfills the condition. Let f^* be a labeling of $V(Q_m)$ which assigns as many labels as possible under Condition A, and γ_m denote the number of labels assigned by f^* to $V(Q_m)$. In the following, we refer to f^* as an *optimal labeling* of $V(Q_m)$. Observe that two labelings shown in Example 1 are both optimal, and that $\gamma_2 = 2$ and $\gamma_3 = 4$. Given an optimal labeling f^* of $V(Q_m)$, we can proceed to construct a sparse hypercube G of order $N = 2^n$, as is described in the following procedure.

procedure Construct_BASE(n, m) $\{\ n > m \geq 1\ \}$

Step 1: Let $V(G) \triangleq \{0,1\}^n$, and let f^* be an optimal labeling of $V(Q_m)$ with a set $C = \{c_1, \ldots, c_{\gamma_m}\}$ of γ_m labels. By using f^*, define a labeling g of $V(G)$ as follows: $\forall u = u_n \ldots u_1 \in V(G)$,

$$g(u) \triangleq f^*(u_m u_{m-1} \ldots u_1).$$

Step 2: Let $S = \{n, n-1, \ldots, m+1\}$. Note that S is not empty since $n > m$. Partition S into γ_m subsets $S_1, S_2, \ldots, S_{\gamma_m}$ in such a way that $||S_i| - |S_j|| \leq 1$ for any $1 \leq i < j \leq \gamma_m$.

Step 3: According to the labeling g of $V(G)$ and the partition of S, define set $E(G)$ of edges as follows: for each $u \in V(G)$,

Rule 1: $\{u, \oplus_i u\} \in E(G)$ for $1 \leq i \leq m$; and

Rule 2: $\{u, \oplus_i u\} \in E(G)$ for $m+1 \leq i \leq n \Leftrightarrow g(u) = c_j$ and $i \in S_j$. □

The basic notion of our construction is that we create 2^{n-m} copies of Q_m and connect them by as few edges as possible to guarantee that the resulting graph is a 2-mlbg (the correctness will be verified below). Rule 1 interconnects the vertices in each subcube by using the i-dimensional edges for $i = 1, 2, \ldots, m$, and Rule 2 interconnects these subcubes in such a way that, for $i = m+1, m+2, \ldots, n$, a vertex u with label c_j is connected with vertex $\oplus_i u$ through the i-dimensional edge iff "i" is a member of subset S_j which has the same subscript "j" with the label of u. In what follows, we denote by $G_{n,m}$ a graph generated by calling Construct_BASE(n, m). Note that the execution of Steps 1 and 2 is nondeterministic, since, in general, there are plural selections of f^* and the partition of S.

In order to verify that $G_{n,m}$ is in fact a 2-mlbg, we present a minimum-time 2-line broadcast scheme for $G_{n,m}$. In the description of the scheme, we use symbol s to denote the source of a broadcast and variable U to denote the set of informed vertices.

scheme Broadcast_2(s)

Phase 1: The objective of the first phase is to disseminate the message between subcubes using prefix of length $n-m$. Let $U = \{s\}$. For $i = n$ (down)to $m+1$, execute the following operation, step by step: $\forall w \in U$, (i) if $\{w, \oplus_i w\} \in E(G_{n,m})$, then w calls vertex $\oplus_i w$ directly by a call of length 1, otherwise, w calls vertex $\oplus_i(\oplus_j w)$ by a call of length 2 passing through vertex $\oplus_j w$, where $\oplus_j w$ is a neighbor of w such that $1 \leq j \leq m$ and $\{\oplus_j w, \oplus_i(\oplus_j w)\} \in E(G_{n,m})$; (ii) if $\{w, \oplus_i w\} \in E(G_{n,m})$, then add $\oplus_i w$ to set U; otherwise, add $\oplus_i(\oplus_j w)$ to set U.

Phase 2: The objective of the second phase is to disseminate the message in subcubes using suffix of length m. For $i = m$ (down)to 1, execute the following operation, step by step: $\forall w \in U$, (i) w calls vertex $\oplus_i w$ by a call of length 1, and (ii) add $\oplus_i w$ to set U. □

Theorem 3 *Scheme* Broadcast_2 *is a minimum-time 2-line broadcast scheme for the graph constructed by* Construct_BASE(n, m). □

We now estimate the minimum of the maximum degree over all the generated graphs. By description,

[1] A dominating set for G is a subset U of $V(G)$ such that for any $u \in V(G)$, either $u \in U$ or there is a vertex $v \in U$ such that $\{u, v\} \in E(G)$.

any graph generated by Construct_BASE(n,m) has the same maximum degree, say $\Delta_{n,m}$. The following two lemmas hold.

Lemma 1 *For any $n > m \geq 1$, $\Delta_{n,m} \leq \left\lceil \frac{n-m}{\gamma_m} \right\rceil + m$.*

Lemma 2 *For any $m \geq 1$, $\lceil \frac{m}{2} \rceil + 1 \leq \gamma_m \leq m+1$.*

Note that the lower bound in the above lemma could not be improved in general. In fact, for $m = 2$, $\gamma_2 = 2 = \lceil \frac{2}{2} \rceil + 1 < 2 + 1$. By using the above lemmas, we have the following theorem (proof is omitted here).

Theorem 4 *For any $n \geq 1$, there is a 2-mlbg G of order $N = 2^n$ such that $\Delta(G) \leq 2\lceil \sqrt{2\log_2 N + 4}\,\rceil - 4$.*

4. Sparse Hypercubes for Larger k

In this section, we extend the ideas of the last section to the case of $k \geq 3$. The idea of an extension is to apply the construction in Section 3, recursively.

To clarify the exposition of our basic idea, we first restrict our attention to the case of $k = 3$. Let G be a graph with vertex set $V(G) = \{0,1\}^n$, and let a, b be integers such that $1 \leq b < a < n$. The following procedure is used to label vertices in $V(G)$.

procedure LABEL(n,a,b) $\{ n > a > b \geq 1 \}$
$\forall u = u_n \ldots u_1 \in V(G)$, label u by $f^*(u_a \ldots u_{b+1})$, where f^* is an optimal labeling of $V(Q_{a-b})$. □

Example 2 Let $n = 7$, $a = 4$, and $b = 2$. Let f^* be the labeling of $V(Q_2)$ $(= V(Q_{a-b}))$ given in Example 1, and let g denote the labeling of $V(G)$ $(= \{0,1\}^7)$ to be obtained by calling procedure LABEL(n,a,b). Then, labeling g is determined as $g(x00y) = g(x11y) = c_1$ and $g(x01y) = g(x10y) = c_2$ for any $x \in \{0,1\}^3$ and $y \in \{0,1\}^2$. □

The idea of our recursive construction of a graph with $N = 2^n$ vertices for $k = 3$, is to connect 2^{n-a} copies of $G_{a,b}$ with respect to the labeling of $V(G)$ obtained by calling LABEL(n,a,b), in an equivalent way to the connection of copies of Q_m with respect to the labeling of $V(G)$ as in Construct_BASE. A formal description of the procedure is given as follows:

procedure Construct_REC(n,a,b) $\{ n > a > b \geq 1 \}$
Step 1: Let $V(G) = \{0,1\}^n$, and label it by a set $C = \{c_1, c_2, \ldots, c_{\gamma_{a-b}}\}$ of γ_{a-b} labels by calling procedure LABEL(n,a,b). For each $u \in V(G)$, let $g(u)$ denote the label assigned to vertex u.
Step 2: Let $S = \{n, n-1, \ldots, a+1\}$. Partition S into γ_{a-b} subsets $S_1, S_2, \ldots, S_{\gamma_{a-b}}$ in such a way that $||S_i| - |S_j|| \leq 1$ for any $1 \leq i < j \leq \gamma_{a-b}$.
Step 3: According to the labeling of $V(G)$ and the partition of S, define set $E(G)$ of edges as follows: for each $u = u_n \ldots u_1 \in V(G)$,

Rule 1: $\{u, \oplus_i u\} \in E(G)$ for $1 \leq i \leq a \Leftrightarrow \{u_a u_{a-1} \ldots u_1, \oplus_i(u_a u_{a-1} \ldots u_1)\} \in E(G_{a,b})$; and
Rule 2: $\{u, \oplus_i u\} \in E(G)$ for $a+1 \leq i \leq n \Leftrightarrow g(u) = c_j$ and $i \in S_j$. □

Note that in Rule 1, two vertices u and $\oplus_i u$ $(\in V(G))$ are connected by the i-dimensional edge, for $1 \leq i \leq a$, if suffices of them of length a are connected by the i-dimensional edge in graph $G_{a,b}$. In other words, Rule 1 makes 2^{n-a} copies of $G_{a,b}$, which has been shown to be a 2-mlbg (see Theorem 3). On the other hand, Rule 2 specifies a way of connecting those copies by as few edges as possible in such a way to guarantee that the resultant graph G is a 3-mlbg.

Example 3 Let us consider an execution of Construct_REC$(7,4,2)$. Suppose that the vertex set $V(G) = \{0,1\}^7$ of G is labeled by g with a set $\{c_1, c_2\}$ of two labels as in Example 2, and that in Step 2, set $S = \{7,6,5\}$ $(= \{n, \ldots, a+1\})$ is partitioned into 2 $(= \gamma_2)$ subsets $S_1 = \{7,6\}$ and $S_2 = \{5\}$. Then, in Step 3, Rule 1 connects 0000000 with three vertices 0000100, 0000010, and 0000001 since in $G_{4,2}$, 0000 is connected with 0100, 0010 and 0001; and, by Rule 2, 0000000 is connected with two vertices 1000000 and 0100000 since $g(0000000) = c_1$ and $S_1 = \{7,6\}$. □

The idea of the recursion can be generalized for larger k as follows. Assume $n \geq k$. We may use $k-1$ parameters $n_1, n_2, \ldots, n_{k-1}$ such that $n > n_{k-1} > \cdots > n_1 \geq 1$ instead of two parameters a, b in Construct_REC. The algorithm is described as follows.

procedure Construct$(k, (n, n_{k-1}, \ldots, n_1))$
$\{ n\ (= n_k) > n_{k-1} > n_{k-2} > \cdots > n_1 \geq 1 \}$
Step 1: If $k = 2$, then call Construct_BASE(n, n_1), and return the resultant graph G as the output.
Step 2: Let $V(G) = \{0,1\}^n$, and label it with a set $C = \{c_1, c_2, \ldots, c_{\gamma_{n_{k-1}-n_{k-2}}}\}$ of $\gamma_{n_{k-1}-n_{k-2}}$ labels by calling LABEL(n, n_{k-1}, n_{k-2}). For each $u \in V(G)$, let $g(u)$ denote the label assigned to vertex u.
Step 3: Let $S = \{n_k, n_k - 1, \ldots, n_{k-1}+1\}$. Partition S into $\gamma_{n_{k-1}-n_{k-2}}$ subsets $S_1, S_2, \ldots, S_{\gamma_{n_{k-1}-n_{k-2}}}$, in such a way that $||S_i| - |S_j|| \leq 1$ for any $1 \leq i < j \leq \gamma_{n_{k-1}-n_{k-2}}$.
Step 4: According to the labeling of $V(G)$ and the partition of S, define set $E(G)$ of edges as follows: for each $u = u_n u_{n-1} \ldots u_1 \in V(G)$,

Rule 1: $\{u, \oplus_i u\} \in E(G)$ for $1 \leq i \leq n_{k-1} \Leftrightarrow \{u_{n_{k-1}} u_{n_{k-1}-1} \ldots u_1, \oplus_i(u_{n_{k-1}} u_{n_{k-1}-1} \ldots u_1)\} \in E(\hat{G})$, where $\hat{G}$ is the graph generated by calling Construct$(k-1, (n_{k-1}, n_{k-2}, \ldots, n_1))$; and
Rule 2: $\{u, \oplus_i u\} \in E(G)$ for $n_{k-1}+1 \leq i \leq n \Leftrightarrow g(u) = c_j$ and $i \in S_j$. □

Note that in Rule 1, two vertices u and $\oplus_i u \in V(G)$ are connected by the i-dimensional edge, for $1 \leq i \leq n_{k-1}$, if suffices of them of length n_{k-1} are connected by the i-dimensional edge in graph $\hat{G}$, which is the graph generated by (recursively) calling procedure Construct($k-1, (n_{k-1}, \ldots, n_1)$). In other words, Rule 1 makes $2^{n-n_{k-1}}$ copies of $\hat{G}$. On the other hand, Rule 2 specifies a way of connecting those copies by as few edges as possible in such a way to guarantee that the resultant graph G is a k-mlbg. In order to verify that the resultant graph G is in fact a k-mlbg, we propose a minimum-time k-line broadcast scheme for graph G below.

scheme Broadcast_k(s)
Phase 1: Let $U = \{s\}$. For $i = n$ (down)to $n_{k-1}+1$, execute the following operation, step by step: $\forall w \in U$, (i) if $\{w, \oplus_i w\} \in E(\hat{G})$, where $\hat{G}$ is the graph generated by calling Construct($k-1, (n_{k-1}, n_{k-2}, \ldots, n_1)$), then w calls vertex $\oplus_i w$ directly by a call of length 1, otherwise, w calls vertex $\oplus_i(\oplus_j w)$ by a call of length at most k passing through vertex $\oplus_j w$, where $\oplus_j w$ is a neighbor of w such that $1 \leq j \leq m$ and $dist_{\hat{G}}(\oplus_j w, \oplus_i(\oplus_j w)) \leq k-1$; (ii) if $\{w, \oplus_i w\} \in E(\hat{G})$ then add $\oplus_i w$ to U; otherwise, add $\oplus_i(\oplus_j w)$ to U.
Phase 2: For $i = n_{k-1}$ (down)to 1, recursively execute a similar operation to Phase 1. □

By a similar argument to Theorem 3 and by induction on k, we can easily prove the following theorem (note that the base case has already been proved in Theorem 3).

Theorem 5 *Scheme* Broadcast_k *is a minimum-time k-line broadcast scheme for the graph constructed by procedure* Construct($k, (n, n_{k-1}, n_{k-2}, \ldots, n_1)$), *where* $n > n_{k-1} > n_{k-2} > \cdots > n_1 \geq 1$. □

Theorem 6 (Upper Bound for General k) *For any $n > k \geq 2$, there is a k-mlbg G of order $N = 2^n$ such that $\Delta(G) \leq (2k-1)\left\lceil \sqrt[k]{\log_2 N - k}\, \right\rceil$.* □

Corollary 1 *For any $k \geq \lceil \log_2 n \rceil$, there is a k-mlbg G of order $N = 2^n$ such that $\Delta(G) \leq 4\lceil \log_2 \log_2 N \rceil - 2$.*

Corollary 2 (Tight Bound for Constant k) *For any constant $k \geq 1$, there is a graph $G \in \mathcal{G}_k$ of order $N = 2^n$ such that $\Delta(G) = \Theta(\sqrt[k]{n})$ which asymptotically attains the lower bound in Theorem 2.*

5. Conclusion

In this paper, we have introduced the k-line communication model and explored the design of efficient broadcast networks and algorithms for this model. The proposed approach starts with a core complete hypercube of lesser order and then interconnects a number of these together in an efficient fashion to realize a sufficient graph. It constructs graphs G of order $N = 2^n$ with $\Delta(G) \leq (2k-1)\left\lceil \sqrt[k]{\log_2 N - k}\, \right\rceil$ for any $n > k \geq 2$, which (asymptotically) attains the lower bound for constant k.

As a side effect of limiting vertex degree and allowing longer calls, we potentially increase communication congestion with competing tasks over the edges of the network. A related, future research problem is that of dealing with accommodating congestion at each edge. Note that our line communication model assumes that calls must be edge-disjoint during each time unit. In practice, each edge could be designed to accommodate several calls simultaneously. A reasonable idea to overcome heavy congestion due to sparseness of the graph is to increase the bandwidth of each link by applying multiedges, as has been explored in dilated networks [1] and fat-trees [10]. Limiting the bandwidth to m concurrent messages per line suggests another dimension of the problem for future research.

References

[1] B. Aiello and T. Leighton. Coding theory, hypercube embeddings, and fault tolerance. In *Proc. 3rd SPAA*, pages 125–136. ACM, 1991.

[2] W. J. Dally and C. L. Seitz. Deadlock–free message routing in multiprocessor interconnection network. *IEEE Trans. Comput.*, 36(5):547–553, May 1987.

[3] A. M. Farley. Minimal broadcast network. *Networks*, 9:313–332, 1979.

[4] A. M. Farley. Minimum-time line broadcast networks. *Networks*, 10:59–70, 1980.

[5] A. M. Farley, S. T. Hedetniemi, S. L. Mitchell, and A. Proskurowski. Minimum broadcast graphs. *Discrete Math.*, 25:189–193, 1979. (not own).

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

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

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

[9] F. T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes.* Morgan Kaufmann Publishers, Inc., 1992.

[10] C. E. Leiserson. Fat-trees: Universal networks for hardware-efficient supercomputing. *IEEE Trans. Comput.*, C-34(10):892–901, Oct. 1985.