Bounded-Call Broadcasting

Arthur Farley and Andrzej Proskurowski

CIS-TR-91-22
August 19, 1991

Department of Computer and Information Science
University of Oregon

Bounded-Call Broadcasting

Arthur M. Farley and Andrzej Proskurowski
Computer and Information Science, University of Oregon

August 19, 1991

Abstract

Broadcasting is the information dissemination task whereby a message from one site of a network (the sender) is transmitted to all other sites (receivers). In this paper, we initiate the study of broadcasting under operational protocols that bound the number of calls made by any site to be less than or equal to a predetermined constant, c. Specifically, we: (i) investigate the c-call broadcast time function, being the minimum possible time required to inform all vertices in a network when the number of calls made by each site is bounded by c; (ii) define a general class of sparse minimum time c-call broadcast graphs and an associated broadcast protocol; (iii) characterize the structure of minimum broadcast trees with call bound c; (iv) discuss the complexity of the recognition problem for minimum time c-call broadcast graphs, and (v) present a catalog of minimum 2-call broadcast graphs, for small values of n.

1 Introduction

A communication network provides the means for dissemination of information among a set of *sites* by transmission of *messages* embodied in *calls* placed over *lines* interconnecting the sites. Definition of a communication network consists of two aspects: *topological* and *operational.* A network's topological specification describes relatively static aspects of the network: the set of sites and their interconnection by lines, together with relevant properties of the components, such as a line's length, cost, and bandwidth or a site's buffer capacity.

We model some of the topological aspects of a network by an undirected graph $G = (V, E)$, consisting of a set V of *vertices*, corresponding to network sites, and a set E of *edges*, where each edge e connects a distinct pair of vertices (v_1, v_2), corresponding to lines of the network. The *degree* of a vertex is the number of edges incident with that vertex. A *path* between two vertices, x and y, is a sequence of edges $(v_1, v_2), (v_2, v_3), ..., (v_{n-1}, v_n)$ such that $v_1 = x$ and $v_n = y$. The *length* of a path is the number of its edges. The *distance* between two vertices is equal to the length of a shortest path between them. Two vertices at a distance of 1 (*i.e.*, directly connected by an edge) are said to be *adjacent* or *neighbors*. A graph is *connected* if there exists a path between every pair of vertices. The *diameter* of a connected graph is equal to the maximum distance between vertices of the graph.

A network's operational specification indicates functional properties of specific components, such as whether a site can receive messages on separate lines concurrently, and general network properties, such as whether it is a store-and-forward or common-access network and particulars of the communication protocols followed.

We model the operational aspects of a network by communication processes executed by sites of the network and by the data structures upon which those processes depend, including routing tables and calling sequences for various information dissemination tasks. In our research we are not concerned with the details of lower level communication protocols such as the encoding of routing information and message content.

Several information dissemination tasks have been considered in communications research. These include message transfer, broadcasting, gossiping, and polling. The basic communication task is that of a *message transfer* between one site, the *sender*, and another site, the *receiver* (*i.e.*, a one-to-one process). Of particular interest in this paper will be the task of broadcasting. *Broadcasting* is the information dissemination task whereby a message from a broadcast *originator* is transmitted to all other sites as receivers (*i.e.*, a one-to-all process). *Gossiping* is the all-to-all process, while *polling* is the one-to-all-to-one process. Broadcasting and gossiping have been studied as important subprocesses within distributed algorithms that require information sharing.

We will consider broadcasting in store-and-forward networks. In such networks, broadcasting is accomplished over a sequence of time units. During each time unit, a set of calls is made, each call involving two sites that are directly connected by a line; a subset of previously informed sites places calls during a given time unit, each site calling a different, uninformed receiver.

Given any broadcast, the message follows paths in the network from the originating vertex to all other vertices; for a non-redundant broadcast, the set of those paths forms a *broadcast tree*. Two observations immediately follow: (i) for broadcasting to be possible in a given network, the graph representing its topology must be connected; (ii) the broadcasting process must take at least as many time units as the diameter of that graph.

Given a connected graph G, the *broadcast time* of vertex $u, b(u,G)$, is the minimum number of time units required to complete broadcasting from vertex u in G. It is easy to see that for any vertex u in a connected graph G with n vertices, $b(u,G) \geq \lceil \log_2 n \rceil$, since the number of informed vertices can at most double during any time unit. The *broadcast time of a graph* $G, b(G)$, is defined to be the maximum broadcast time of any vertex u in G, *i.e.* $b(G) = \max\{b(u) : u \in V(G)\}$. For the complete graph K_n with $n \geq 2$ vertices, $b(K_n) = \lceil \log_2 n \rceil$. However, K_n is not minimal (in number of edges) with respect to this property for any $n \geq 3$. That is, we can remove edges from K_n and still have a graph G with n vertices such that $b(G) = \lceil \log_2 n \rceil$. We refer to the class of graphs in which broadcast can be completed in the minimum possible time (*i.e.*, $\lceil \log_2 n \rceil$) as *minimum-time broadcast graphs*, or *mtbgs* (see [10]).

The *broadcast function*, $B(n)$, is the minimum number of edges in any *mtbg*. A *minimum broadcast graph* (*mbg*) is a minimum-time broadcast graph on n vertices having $B(n)$ edges. From an applications perspective, minimum broadcast graphs represent the cheapest possible communication networks (having the fewest communication lines) in which broadcasting can be accomplished from any vertex as fast as is theoretically possible.

In [11], Farley, Hedetniemi, Mitchell and Proskurowski began the study of $B(n)$. In particular, they determined the values of $B(n)$ for $n \leq 15$ and noted that $B(2^k) = k2^{k-1}$ (*i.e.*, the k-cube is an *mbg* on $n = 2^k$ vertices). Mitchell and Hedetniemi [13] determined the value for $B(17)$, while Bermond, Hell, Liestman and Peters [1] found the values of $B(19), B(30)$, and $B(31)$. Otherwise, $B(n)$ is not known for any value of $n > 32$, except for $n = 2^k$, where the k-cube can be used as an *mbg* ([1]).

The limited results above suggests that *mbg*s are difficult to find. In fact, Slater, Cockayne, and Hedetniemi [15] have shown that, given an arbitrary graph G, vertex v, and $t \geq 4$ as input, deciding whether $b(v, G) \leq t$ is $\mathcal{NP}$-complete. Thus, determining whether a particular graph is an *mtbg*, let alone an *mbg*, is a difficult problem. As a result, several authors have devised methods to construct sparse graphs which allow minimum-time broadcasting from each vertex. We use the term *sparse broadcast graph* to denote an *mtbg* on n vertices having "close to" $B(n)$ edges (*i.e.*, $\mathcal{O}(n \log_2 n)$ edges).

Farley [10] designed several techniques for constructing sparse broadcast graphs with n vertices and approximately $\frac{n}{2} \log_2 n$ edges, for arbitrary values of n. Chau and Liestman [6] presented constructions based on Farley's technique which yield somewhat sparser graphs for most values of n. Recently, Gargano and Vaccaro [7] gave constructions which produce the best of the known graphs for some large values of n. Grigni and Peleg's [9] construction produces the best of the known graphs for most values of n.

In all of the above research, it is assumed that once a site becomes informed of a broadcast message, it can place calls to its neighbors during any number of calls, until the broadcast is completed. If optimal time is to be realized, this commitment can range from $\lceil \log_2 n \rceil$ calls for the originator down to 1 or 0 calls for the sites informed during the next to last or last time units, respectively. Since most sites are informed during the last two time units in an optimal scheme, it would appear we may be able to limit a site's maximum operational commitment to a predetermined, small number of calls and not suffer great loss in performance. Such a scheme would better distribute the operational costs of a broadcast throughout the network, guaranteeing a limited time commitment for broadcast procedures at each site, regardless of originator.

In this paper, we begin the study of broadcasting under operational protocols that bound the number of calls made by any site during broadcasting to be less than or equal to a predetermined constant, c. (We will indicate the call bound c as a subscript of pertinent identifiers,) Specifically, we will: (i) investigate the c-call broadcast time function $b_c t(n)$, being the minimum possible time required to inform n vertices in any network when calls at each site are bounded by a constant c; (ii) define a general class of sparse $mtb_c g$s (*i.e.*, minimum time broadcast graphs with call bound c) and an associated broadcast protocol; (iii) characterize the structure of mb_cts (*i.e.*, minimum broadcast trees with call bound c); (iv) discuss the complexity of the recognition problem for minimal c-call broadcast graphs, and (v) present a catalog of mb_2gs, determining values for $B_2(n)$, for small values of n. Throughout the remainder of this paper, we will assume that n refers to the number of sites in a network and c refers to the maximum number of calls that can be made by any site during a broadcast.

Other researchers have considered the definition of a graphs with bounded maximum degree in which every vertex can broadcast "quickly" (see [2]). We will use the term *bounded-degree broadcast graph* (b_dbg) to describe a graph G on n vertices with maximum degree d such that $b(G)$ is "close to" the minimum time required to broadcast in any network with n vertices, $bt(n)$. In a recent paper, Liestman and Peters [12] investigate bounded-degree broadcast graphs with maximum degrees 3 and 4 (i.e., b_3bgs and b_4bgs). They give lower bounds on the time required to broadcast in such graphs and present several constructions that produce good bounded degree broadcast graphs. Liestman and Peters show that $b(b_3bg) \geq 1.440 log_2 n - 1.769$, and that if n is a power of 2, then $b(b_3bg) < 2\log_2 n + 1$; the upper bound is achieved by constructing folded-shuttle-exchange graphs [5]. They also show that $b(b_4bg) \geq 1.137\log_2 n - 0.637$ and that, if n is a power of 4, then $b(b_4bg) \leq 1.625\log_2 n + 2.25$. More recently, Bermond and Peyrat [3] considered broadcasting in de Bruijn and Kautz graphs. They were able to improve on the upper bounds of Liestman and Peters, showing, in particular, that $b(b_4bg) \leq 1.5\log_2 n + 1$, if n is either a power or 3 times a power of 2.

In our discussions to follow, we will note the differences in results that serve to distinguish the bounded-call broadcast problem discussed here from the bounded-degree broadcast problem discussed elsewhere.

2 Broadcast Time Function $b_ct(n)$

In this section, we investigate the function $b_ct(n)$, being the minimum time (over all possible graphs and calling sequences, *i.e.*, without any topological constraints) required to broadcast a message to n sites, when we limit each vertex to making at most c calls. We will assume that every informed vertex makes all its calls as soon as possible.

We can characterize the minimum time required to broadcast to n vertices in terms of the following functions indicating the maximum number of sites informed after t time units or newly informed during time unit t.

We define the function *informed*, $i_c(t)$, to be the maximum possible number of vertices informed after t time units, given each site is limited to at most c calls. By our earlier observations, $i_c(t) \leq 2^t$ for $c, t > 0$, with $i_c(t) = 2^t$ for $t \leq c$. By definition, $b_ct(n)$ is greater than or equal to the minimum t such that $n \leq i_c(t)$.

We define the function *newly-informed*, $ni_c(t)$, to be the maximum number of sites informed during time unit t, given each site is limited to at most c calls. Since only those informed in the immediately previous c time units can place calls during a current time unit, $ni_c(t) = \sum_{1 \leq i \leq c} ni_c(t-i)$.

By definition, $i_c(t) = \sum_{1 \leq i \leq t} ni_c(i)$. Table 1 give values of $ni_c(t)$ and $i_c(t)$, for small c and t, to illustrate behavior of the functions.

c	1		2		3		4		5	
t	$ni_1(t)$	$i_1(t)$	$ni_2(t)$	$i_2(t)$	$ni_3(t)$	$i_3(t)$	$ni_4(t)$	$i_4(t)$	$ni_5(t)$	$i_5(t)$
0	1	1	1	1	1	1	1	1	1	1
1	1	2	1	2	1	2	1	2	1	2
2	1	3	2	4	2	4	2	4	2	4
3	1	4	3	7	4	8	4	8	4	8
4	1	5	5	12	7	15	8	16	8	16
5	1	6	8	20	13	28	15	31	16	32
6	1	7	13	33	24	52	29	60	31	63
7	1	8	21	54	44	96	46	106	61	124
8	1	9	34	88	81	177	98	204	120	244

We note that the function $i_c(t)$ can also be expressed by the following recurrence relation:

$$i_c(t) = \begin{cases} 0, & \text{for } t < 0; \\ 2^t, & \text{for } 0 < t \leq c; \\ 2i_c(t-1) - i_c(t-c-1), & \text{for } t > c. \end{cases}$$

This can be compared to the recurrence relation derived in [12] for broadcasting in graphs with vertices having degrees bounded by d:

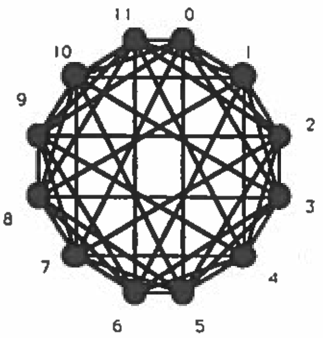
$$i_d(t) = \begin{cases} 0, & \text{for } t < 0; \\ 2^t, & \text{for } 0 < t \leq d; \\ 2i_d(t-1) - i_d(t-d), & \text{for } t > d. \end{cases}$$

The subtle difference arises as follows. In bounded-degree graphs, the originator can place at most d calls and then all other sites can place at most $d-1$ calls. If we let $c = d$, we see that all non-originating sites are handicapped by being able to place one less call. Thus, when $c = d$, $i_c(t) > i_d(t)$, for large t. If we consider $c = d - 1$, we see that the originator can participate in one extra call, thereby starting one more subtree of calls and allowing the doubling of informed sites to continue for one extra time unit. In this case, when $c = d - 1$, $i_c(t) < i_d(t)$, for large t.

In [2], Bermond *et al.* present a table that shows the best lower-bound, asymptotic behavior for $b_d t(n)$, based on the recurrence for $i_d(t)$, as a function of the form: $df_d \cdot \log_2 n$. The lower bound values of the *degree delay factor* df_d for small values of d are as follows: $df_3 = 1.1803; df_4 = 1.0901; df_5 = 1.0450; df_6 = 1.0225$. As discussed above, the value for the analogous *call delay factor* cf_c lies between df_c and df_{c+1}, for any given c. As an example, one can easily see that the minimum broadcast trees for $c = 2$ are subtrees of Fibonacci trees (see section 4). Fibonacci trees have an asymptotic depth (thus, broadcast time) of $1.441 \log_2 n$ (see, for instance, [4]) which is greater than df_3. Limiting calls to at most two per site degrades optimal broadcast performance by less than 50%. By the results quoted above, we see that if we allow at most 4 calls per site, the time penalty for optimal broadcasting will be less than 10% in an mtb_4g, for large n.

3 A Class of Sparse mtb_cgs

In this section, we define a class of sparse mtb_cgs and an associated calling protocol that realizes optimal time broadcasting. Our design is based upon the general class of graphs known as star polygons. A *star polygon* on n vertices is determined by a finite offset list $(s_1, ..., s_m)$ and is constructed by numbering the vertices uniquely from 0 to $n-1$, arranging them in a circle with vertex numbers increasing sequentially, and connecting each vertex i to the set of neighbors $\{i + s_j\}$ for $1 \leq j \leq m$ as computed from the offset list.



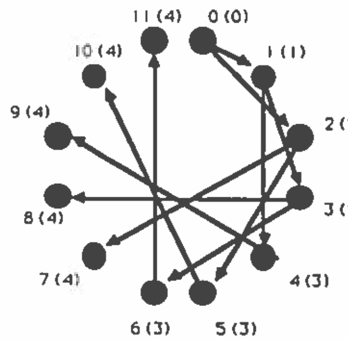


Figure 1: Sparse minimum time 2-broadcast graph and its calling scheme

If $c = 1$, the mtb_cg is a cycle on n vertices. If $n = 1$, no edges are needed (*i.e.*, broadcast is immediate). For any given pair of values $c, n > 1$, we construct a sparse mtb_cg by creating a star polygon on n vertices with offset list being $ni_c(t)$, for all t such that $i_c(t) \leq n$; any duplicated edges are consolidated into a single edge. As an example, Figure 1 presents the mtb_2g on 12 vertices determined by this method. When $c > \log_2 n$, the graphs produced are identical to those described in Farley [10]. As such, the construction technique presented here is a generalization of that earlier method.

To complete an optimal broadcast in these graphs, each vertex places calls during at most c time units directly following the time at which it becomes informed. Broadcasting is completed by having each vertex that can call at time t place a call the vertex offset by $ni_c(t)$ (unless this would call a previously informed site, due to wrapping around the circle). As an example, consider the broadcast scheme from site 0 in the 12 vertex network with call limit 2 shown in Figure 1. The broadcast progresses clockwise from the originator, adding $ni_c(t)$ sites each time until the broadcast is complete. Thus, the broadcast time is optimal.

Our design of mb_ctgs and calling sequences can accomodate all positive values of c and n. While the method is general and produces sparse graphs relative to complete graphs, the smaller the c the more edges required for a graph on n vertices. This would seem a bit counterintuitive, due to our discussion of the relationship of bounded-call broadcasting to broadcasting in bounded-degree graphs and the fact that broadcasts are also taking longer with smaller c. We can probably do better than the graphs generated above; but it will not be easy to get optimal mb_cgs, as we shall see.

4 Characterization of mb_cts

As noted earlier, each non-redundant broadcast in a graph G induces a broadcast tree in G. If the broadcast time is to be minimum, the tree must be a member the class of minimum time c-call broadcast trees. These can be constructed (and counted) in a manner similar to that of the general minimum time broadcast trees (*cf.* Proskurowski [14]). We will indicate the construction algorithm through a recurrence counting the different minimum time c-call broadcast trees with n vertices.

Let $N_c(n,t)$ be defined as the number of rooted, ordered trees with n vertices in which the root can c-call broadcast a message in time t. We require that every vertex forward the message (making all its calls) as soon as possible.

$$\begin{aligned} N_c(0,t) &= 1 \text{ for al } t \\ N_c(1,t) &= 1 \text{ for } t \geq 0 \\ N_c(1,t) &= 0 \text{ for } t < 0 \\ N_c(n,0) &= 0 \text{ for } n > 1 \\ N_c(n,t) &= \sum_{\mathbf{n}} \prod_{1 \leq i \leq c} N_c(n_i, t-i) \text{ otherwise} \end{aligned}$$

where the summation is over all partitions of $n-1$, $\mathbf{n}=(n_1, n_2, \ldots, n_c)$ such that $\sum_{1 \leq i \leq c} n_i = n-1$ and if $n_i = 0$ then $n_j = 0$, for all $i,j : 1 \leq i \leq j \leq c$.

The recurrence for $N_c(n,t)$ reflects construction of broadcast trees through the distribution of $n-1$ descendants of the root into the c subtrees represented by $N_c(n_i, t-i)$ above. The special cases of the recurrence with value 0 capture the impossibility of informing too many vertices in too little time (when $n > i_c(t)$). The latter contraint represents our assumption regarding timely forwarding of messages. As an illustration of the above counting and construction method, we give all mb_2ts with at most 7 vertices in Figure 2.

5 Complexity of mb_cg problems

In this section we will discuss the complexity status of determining the minimum time bounded-call broadcast from a specified vertex of a given graph and, more generally, of determining the membership of a given graph in the class of mtb_cgs.

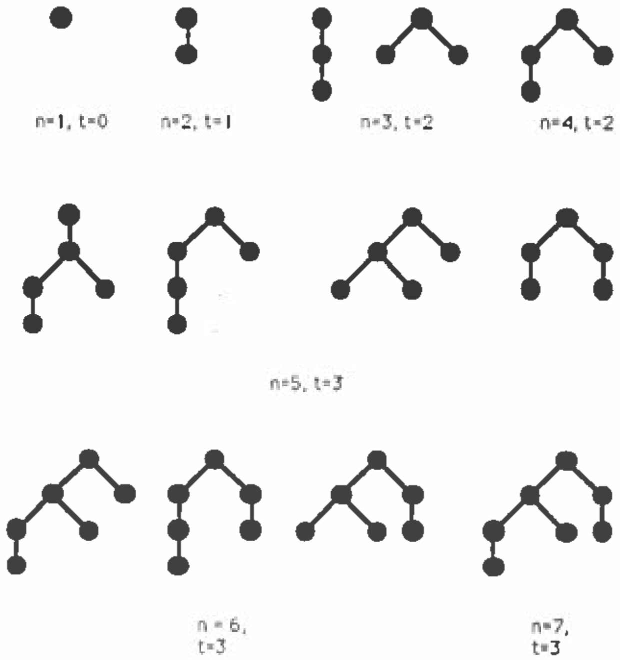


Figure 2: Minimum 2-call broadcast trees with $n = 1..7$ vertices

The former question bears a strong resemblance to the similar question for unrestricted broadcast (*cf.* [15]) and can be answered in a similar manner. Following the pattern of the proof in [15], we show that the Multiple Source c-Call Broadcast problem is $\mathcal{NP}$-complete by reduction from the 3-dimensional matching problem (*3DM*, *cf.* [8]). Since our proof requires the call limit $c > 1$, we show that in the case of $c = 1$ the latter question is closely related to the *Hamiltonian Path* problem.

Let us first define the corresponding decision problems:

Multiple Source c-Call Broadcast (MSB_c)

Instance: Graph G, subset of vertices V_0, integer $t \geq 5$.

Question: Is there a c-bounded call broadcast from V_0 in G that takes no more than t time units?

3-Dimensional Matching

Instance: Three sets X, Y, Z, each of cardinality m and a set of triples $M \subseteq X \times Y \times Z$.

Question: Is there an m-subset of M covering all elements in X, Y, Z?

Hamiltonian Path

Instance: Graph G.

Question: Is there a simple path in G containing all vertices?

Single Source c-Call Broadcast (SSB_c)

Instance: Graph G and a vertex u, integer t.

Question: Is there a c-bounded call broadcast from u in G that takes no more than t time units?

mtb_cg **Membership**

Instance: Graph G.

Question: Is G an mtb_cg?

First we will show that the MSB_1 problem is $\mathcal{NP}$-complete. For any particular broadcast source, the question is roughly equivalent to the existence of a Hamiltonian path. Below, we indicate a construction allowing a precise reduction between the two problems.

Theorem 1. The SSB_1 problem is $\mathcal{NP}$-complete.

Proof: Let G be an instance of the Hamiltonian Path problem. Construct an instance of SSB_1, H, by adding to G a new vertex, u, adjacent to all vertices of G. A Hamiltonian path in G between some v_i and v_j implies the following 1-bounded call broadcast from u by first calling v_i and then completing the broadcast along the Hamiltonian path to v_j. Conversely, if H admits a 1-bounded call broadcast from u, it defines a Hamiltonian path in G. ∎

By a trivial reduction from SSB_1, one can see that MSB_1 is also $\mathcal{NP}$-complete. We now prove a similar result for larger values of c.

Theorem 2. The MSB_2 problem is $\mathcal{NP}$-complete.

Proof: Put more formally, the question asks if there is a sequence of t subsets of vertices, $V_i, 0 < i \leq t$, such that: (i) $\cup_{0\leq i\leq t} V_i = V(G)$, (ii) for all $i, 0 < i \leq t$, each vertex in V_i is adjacent to a different vertex in $\cup_{0\leq j<i} V_j$, and (iii) each vertex in $V(G)$ is so distinguished at most c times. Obviously, the

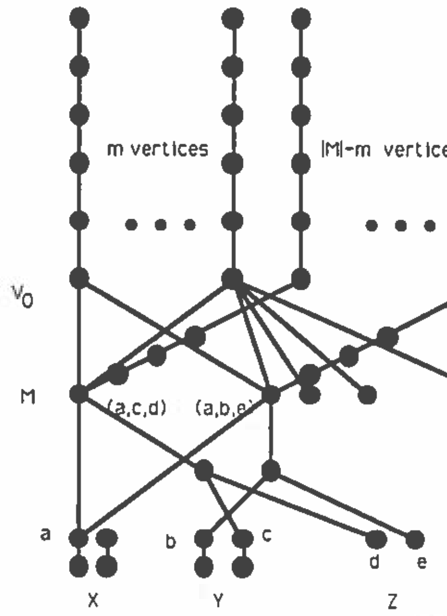


Figure 3: The instance of MSB_2 in the reduction from *3DM*

problem is in $\mathcal{NP}$, since one can verify in polynomial time the conditions of a broadcast. To prove completeness, we will show a reduction from the $\mathcal{NP}$-complete *3DM* problem.

Given an instance of *3DM*, we construct an instance of MSB_2, G (the graph G is indicated in Figure 3). Its vertex set, $V(G)$, consists of a number of disjoint subsets. There are "basic" sets of vertices representing elements of M, X, Y, Z and $|M|$ vertices of the originator set V_0. The other vertices will ensure proper 2-broadcasting conditions. These vertices include a set of "intermediate" vertices between elements of M and the corresponding vertices of Y and Z, vertices arranged in paths pending from the vertices of V_0, X, and Y, and vertices of paths between $|M| - m$ vertices of V_0 and all vertices of M, as indicated. The special m vertices of V_0 and vertices of M induce a complete bipartite subgraph of G and each vertex of M is adjacent to elements of X and (through an intermediate vertex) Y and Z the corresponding to the triple the vertex represents. The 2-broadcast time t is equal to 5.

It is fairly easy to see a 2-call broadcast scheme in G originating in V_0

and informing all vertices within 5 time units if there is a matching S of M. Namely, all vertices of V_0 call their pending path neighbors first, and then the m special vertices call the vertices in S, that can complete the 2-call broadcast in the remaining 4 time units by first calling their intermediate vertices and then the vertices of X. Also, this is the only type of 2-call broadcast that succeds in 5 time units, since calls of the other $|M| - m$ vertices of V_0 cannot reach M before time 5, and the m vertices of M called in time unit 2 have to call as described above. Thus, an affirmative answer to the *3DM* problem is equivalent to an affirmative answer to MSB_2, completing the proof. ∎

The above construction can be easily modified to treat the case of the general $c > 2$. This would require a construction that would force $c - 1$ calls "up" from all vertices of V_0. Thus, the following theorem could be proved.

Theorem 3. The MSB_c problem is $\mathcal{NP}$-complete. ∎

The single source broadcast problem has the same complexity status.

Theorem 4. The SSB_c problem is $\mathcal{NP}$-complete.

Proof: Since it is a special case of the MSB_c problem, it is in $\mathcal{NP}$. To prove completeness, we will follow again the proof of [15] in reducing the *3DM* problem to our problem, for $c = 2$. The construction involves some additional vertices and edges in the graph from the previous proof (Figure 3). Namely, the originator vertex u will be the root of a tree with leaves connected in one-to-one fashion with the vertices of V_0. The structure of the tree will ensure that all vertices of V_0 will become informed at the same time or else the broadcast will not be completed in the required time $|M| + 5$. ∎

The complexity status of mtb_cg Membership problem is not certain. The problem is equivalent to the existence of spanning trees rooted at every vertex of the instance graph and belonging to the set of mb_cts. In the case of $c = 1$, the reduction from the Hamiltonian Path problem is given below.

Theorem 5. The mtb_1g Membership problem is $\mathcal{NP}$-complete.

Proof: Let G be an instance of the Hamiltonian Path problem. Construct an instance of mtb_1g Membership problem, H, by creating two disjoint copies of G, H' and H'' (with vertices labeled, say, v_i' and v_i'') and two new vertices h', h'' adjacent to all other vertices (but not to each other). A Hamiltonian path in G between some v_i and v_j implies the following 1-bounded call broadcast from a vertex v'_k in H: call along the Hamiltonian path in H' up to v'_i, then call h', v''_j, along the Hamiltonian path in H'' up to v''_i, h'', v'_j, and complete the broadcast along the Hamiltonian path in in H'. (As broadcast originators, vertices h', h'' call the corresponding starting points of Hamiltonian paths first.) Conversely, if H is an mtb_1g, then it admits a 1-bounded call broadcast from h'. Such a broadcast defines a Hamiltonian path in G, since all vertices in an isomorphic copy of G have to be informed before calling h''. ■

While we have not proven that the mtb_cg Membership problem (for general c) is $\mathcal{NP}$-complete, we have shown related problems to be so. Given that the problem of determining whether a graph is an mtb_cg appears difficult, if not $\mathcal{NP}$-complete, the investigation and enumeration of mb_cgs for restricted classes of c and n is well motivated.

6 Instances of mb_2gs for small n

Below, we will construct mb_2gs with 1 through 12 vertices. We will use the standard notation C_i for the cycle with i vertices. A graph is *unicyclic* if it has exactly one cycle. "Degree 2 path" and "degree 2 loop" denote a path and a cycle of degree 2 vertices, respectively. Degree sequence analysis will require the matching "half-edges", *i.e.*, considering vertices with unassigned ("uncommitted") vertex degrees as adjacent.

In our proofs, we will make use of the structure of minimum time c-broadcast trees. Based on the shape of these trees, we will derive certain necessary or impossible neighborhood configurations for the broadcast originator. We will be able to eliminate certain graphs by discovering that (i) their diameter exceeds the purported broadcast time, (ii) more than one vertex is at the diameter distance from a broadcast originator, or (iii) two vertices have to make a call and share a potential receiver of the call (a "choking" situation).

Our presentation consists of a terse description of the mb_2gs for a given number of vertices, n, with proof based upon the above considerations. We leave the task of finding minimum time broadcast schemes from every vertex as an exercise for the reader.

$n = 1 - 4$. The optimum graphs are unique (paths and C_4).

$n = 5$. The optimum graphs are unicyclic. A tree will not work because of a combination of "choking" on a degree 3 vertex and the diameter; this argument applies also to $n = 4$. Each of the three possible graphs derived from C_5, C_4, C_3, where degree 1 vertices are adjacent to distinct vertices of the cycle, succeeds.

$n = 6$. The unique optimum graph is the C_6, since any degree 1 vertex as the originator would require broadcasting in $t = 2$ among 5 vertices.

$n = 7$. The optimum graphs have at least 9 edges. Since there must be no degree 1 vertex, C_7 does not work; every vertex needs a degree ≥ 3 neighbor, therefore a C_7 with a chord fails. Vertex degree sequence analysis for graphs with 9 edges that have at least two adjacent vertices of higher degree follows:

$n_2 = 5, n_3 = 1, n_5 = 1$: unavoidable C_4 off the degree 5 vertex fails to broadcast in minimum time.

$n_2 = 4, n_3 = 2, n_4 = 1$: case analysis based on mutual adjacency of degree 3 vertices gives two graphs.

$n_2 = 5, n_4 = 2$: since there must not be an articulation point, case analysis over the lengths of the three degree 2 paths gives one graph.

$n_2 = 3, n_3 = 4$: case analysis based on subgraphs induced by degree 3 vertices shows that two pairs of adjacent degree 3 vertices fail, as do all other connected configurations, except for a path and a delta that yield two graphs (each isomorphic to C_7 with two chords).

The five minimum 2-broadcast graphs on 7 vertices are given in Figure 4.

$n = 8$. C_8 works, while no tree does. The only other unicyclic graph that works is C_4 with four leaves, each adjacent to a different vertex of the

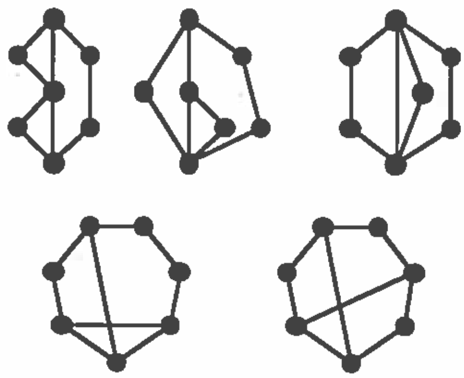


Figure 4: All minimum 2-call broadcast graphs with $n = 7$ vertices

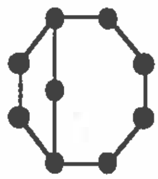


Figure 5: The unique minimum 2-call broadcast graph with $n = 9$ vertices

cycle (because a degree 1 vertex must be adjacent to the root of the unique mb_2t with 7 vertices).

$n = 9$. There can not be a degree 1 vertex, since 8 vertices also need 4 time units to broadcast. Consideration of minimum 2-call broadcast trees for 9 vertices indicates that every vertex must be within distance 2 of a *big* vertex (*i.e.*, with degree≥ 3), so C_9 fails. Also, adding one chord is not sufficient (the "almost diagonal" is the only possibility).

However, adding a path with 1 vertex as a "subdiagonal" to the C_8 works (*cf.* Figure 5). The only other configurations yielding 9 vertices and 10 edges are the octagon with a vertex on a "diagonal" path, or degree 2 loops off the two big vertices, which both fail.

$n = 10$. Examination of minimum time 2-call broadcast trees with 10 vertices (see Figure 6) indicates that all vertices must be of degree at least 2 and have a big vertex as neighbor. This rules out 11 edges. Three graphs with 12 edges are successful. We discover these graphs by considering

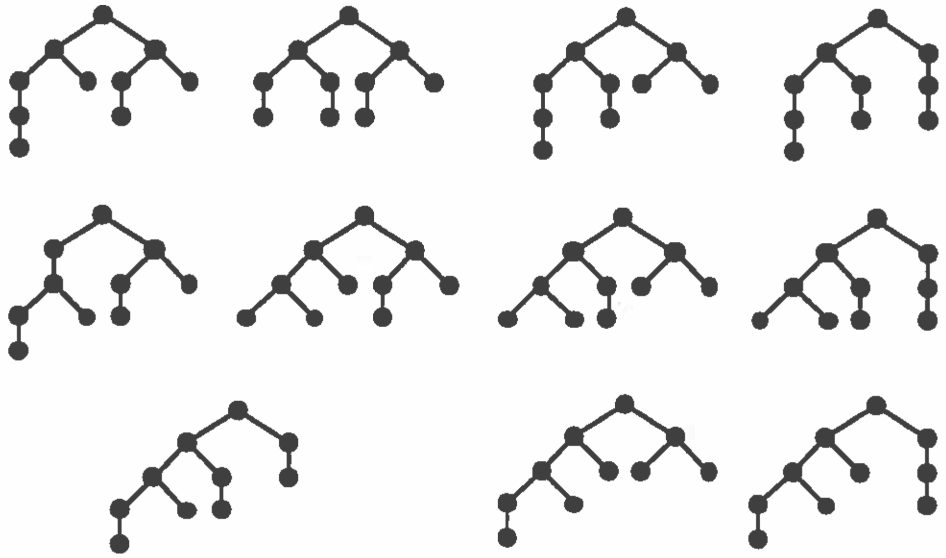


Figure 6: All minimum 2-call broadcast trees with $n = 10$ vertices

degree sequences:

$n_2 = 9, n_6 = 1$: no successful graphs, as the only big vertex does not have a big neighbor.

$n_2 = 8, n_3 = 1, n_5 = 1$: no successful graph, as the two big vertices must be neighbors, leaving only six half-edges to accommodate eight degree 2 vertices.

$n_2 = 8, n_4 = 2$: the same as the preceding case.

$n_2 = 7, n_3 = 2, n_4 = 1$: no successful graph, as the three big vertices must be connected through at least two edges using four half-edges and leaving only six half-edges to accommodate the seven degree 2 vertices.

$n_2 = 6, n_3 = 4$: the big vertices must induce two edges, a path, or a star (other induced subgraphs use too many half-edges). Each of these cases leads to exactly one successful graph (see Figure 7).

$n = 11$. Examination of the minimum time 2-call broadcast trees with 11 vertices (see Figure 8) indicates that the originator must induce a path

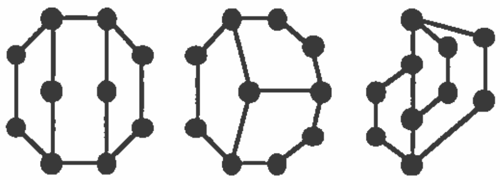


Figure 7: All minimum 2-call broadcast graphs with $n = 10$ vertices

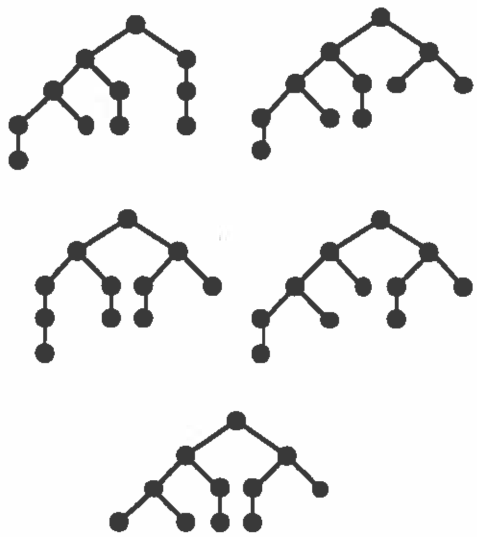


Figure 8: All minimum 2-call broadcast trees with $n = 11$ vertices

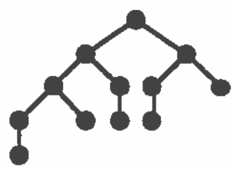


Figure 9: The minimum 2-call broadcast tree with $n = 12$ vertices

with two big vertices. Since the originator can be a big vertex itself, we must allow for three big vertices in a successful graph. The minimum possible number of edges in such a graph would be 13, since three big vertices must have degrees 4, 3, and 3, or 3, 3, 3, and 3. In either case there is not enough half-edges from the big vertices (4 or 6 half-edges reserved for connections between big vertices) to accommodate the small vertices as neighbors. Thus, at least 14 edges are necessary in an optimal graph with 11 vertices. Indeed, 14 edges are sufficient, as shown by the example graph in Figure 10.

$n = 12$. The broadcast originator must induce a four-vertex path with three big vertices, itself an interior vertex of this path (see Figure 9 for the minimum time 2-call broadcast tree). 14 edges are not sufficient, since four degree 3 vertices have too few half-edges to accomodate the degree 2 neighbors (at least 8 half-edges reserved for the connections between the big vertices).

A graph with 15 edges could be achieved by six degree 3 vertices, connected by at least 6 edges; this leaves too few half-edges to accomodate small vertices. With a degree 4 vertex and four degree 3 vertices (at least 5 connecting edges) or with a degree 5 vertex and three degree vertices, we have the same deficiency as before.

Let us consider graphs with 16 edges. Since every big vertex has to be connected to at least two others, k edges connecting k big vertices are required. This leaves eight degree 3 vertices as the only possibility in the vertex degree sequence (the presence of a degree 4 or 5 vertex leaving too few half-edges to accommodate the small vertices). The single cycle, C_8, connecting the big vertices provides only two feasible connection schemes for small vertices: along diagonals or along sub-diagonals of the octagon. (This is because a small vertex in a cycle of

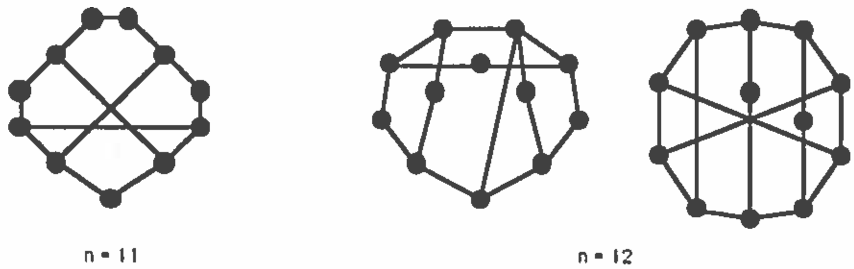


Figure 10: Minimum 2-call broadcast graphs with $n = 11, 12$ vertices

length 3 or 4 with degree 3 neighbors necessarily chokes the broadcast.) By inspection, both graphs fail. Similarly, degree 3 vertices inducing two C_4's choke the broadcast.

Thus, at least 17 edges are necessary. Figure 10 presents two graphs with 12 vertices and 17 edges in which minimum time 2-call broadcast is possible.

To summarize our results from this section, we compare the traditional broadcast function $B(n)$, indicating the minimum number of edges required in a minimum-time broadcast graph on n vertices, with the bounded-call broadcast function $B_2(n)$, for n from 1 to 12.

n	1	2	3	4	5	6	7	8	9	10	11	12
$B(n)$	0	1	2	4	5	6	9	12	10	12	13	15
$B_2(n)$	0	1	2	4	5	6	9	8	10	12	14	17

The difference at $n = 8$ is due to the difference in broadcast times: $b_2t(n)$ is 4, while $bt(n)$ is 3. Of particular interest are the results for n =11 and 12. They indicate that, though broadcast times are the same, bounding the number of calls a site can make necessitates more edges in the corresponding minimum broadcast graph. We noted this was a property of our general class of sparse minimum-time broadcast graphs defined in Section 3.

The result for $n = 12$ indicates there exists a minimum broadcast graph for $c = 2$ that contains a vertex with degree greater than 3, thus further distinguishing bounded-call broadcasting from the bounded graph degree constraints discussed in [12,2]. The other mb_2g with 12 vertices does have maximum vertex degree bounded by 3.

It remains an open question whether there exists an mb_cg with maximum vertex degree bounded by $c+1$, for all $c, n > 1$.

References

[1] J-C. Bermond, P. Hell, A.L. Liestman, and J.G. Peters, Sparse broadcast graphs, *Discrete Applied Mathematics*

[2] J-C. Bermond, P. Hell, A.L. Liestman, and J.G. Peters, Broadcasting in bounded degree graphs, *SIAM Journal of Discrete Mathematics*

[3] J-C. Bermond and C. Peyrat, Broadcasting in DeBruijn networks, Proceedings of the 19th South-Eastern Conference on Combinatorics, Graph Theopry and Computing, *Congressus Numerantium* (1988), 283-292;

[4] Brown and Sedgewick, A dichromatic framework for balanced trees, in *Proceedings of 19th STOC* (1978), 8-21;

[5] F.R.K. Chung, Diameters of graphs: old problems and new results, Proceedings of the 18th South-Eastern Conference on Combinatorics, Graph Theory and Computing, *Congressus Numerantium* (1987), 295-317;

[6] S.C. Chau and A.L. Liestman, Constructing minimal broadcast networks, *J. Combinatorics, Information and System Science 10* (1985), 110-122;

[7] L. Gargano and U. Vaccaro, On the construction of minimal broadcast networks, *Networks 19* (1989), 673-689;

[8] Garey and Johnson, *Computers and Intractability*, Freeman (1978);

[9] M. Grigni and D. Peleg, Tight bounds on minimum broadcast networks, *TR MIT/LCS/TM-374* (1988);

[10] A.M. Farley, Minimal broadcast networks, *Networks 9* (1979), 313-332;

[11] A.M. Farley, S.T. Hedetniemi, S.L. Mitchell, and A. Proskurowski, Minimum broadcast graphs, *Discrete Mathematics 25* (1979), 189-193;

[12] A.L. Liestman and J.G. Peters, Broadcast networks of bounded degree, *SIAM J. of Dicrete Mathematics 1* (1988), 531-540;

[13] S.L. Mitchell and S.T. Hedetniemi, A census of minimum broadcast graphs, *J. Combinatorics, Information and System Science 5* (1980), 141-151;

[14] A. Proskurowski, Minimum broadcast trees, *IEEE Transactions on Computers C-30*, 5(1981), 363-366;

[15] P.J. Slater, E. Cockayne, and S.T. Hedetniemi, Information dissemination in trees, *SIAM J. Computing 10* (1981), 692-701.