Sparse broadcast graphs

Jean-Claude Bermond, Pavol Hell, Arthur Liestman, Joseph Peters

Sparse broadcast graphs

Jean-Claude Bermond

I3S, CNRS, Bât 4, rue A. Einstein, Sophia-Antipolis, 06560 Valbonne, France

Pavol Hell, Arthur L. Liestman and Joseph G. Peters

School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada V5A 1S6

Abstract

Broadcasting is an information dissemination process in which a message is to be sent from a single originator to all members of a network by placing calls over the communication lines of the network. Several previous papers have investigated ways to construct sparse graphs (networks) on n vertices in which this process can be completed in minimum time from any originator. In this paper, we describe four techniques to construct graphs of this type and show that they produce the sparsest known graphs for several values of n. For $n=18$, $n=19$, $n=30$ and $n=31$ we also show that our new graphs are *minimum broadcast graphs* (i.e., that no graph with fewer edges is possible). These new graphs can be used with other techniques to improve the best known results for many larger values of n.

Keywords. Broadcasting, graphs, networks.

1. Definitions

Broadcasting refers to the process of message dissemination in a communication network whereby a message, originated by one member, is transmitted to all members of the network. Broadcasting is accomplished by placing a series of calls over the communication lines of the network. This is to be completed as quickly as possible subject to the constraints that each call involves only two vertices, each call requires one unit of time, a vertex can participate in only one call per unit of time, and a vertex can only call a vertex to which it is adjacent.

* This article was written while the first author was visiting the School of Computing Science, Simon Fraser University for which the support of Advanced Systems Institute and Simon Fraser University is gratefully acknowledged. This research was also supported by the Natural Sciences and Engineering Research Council of Canada under Grants No. A-5057, A-1734 and A-0322.

Given a connected graph G and a message originator, vertex u, we define the *broadcast time of vertex u*, $b(u)$, to be the minimum number of time units required to complete broadcasting from vertex u. It is easy to see that for any vertex u in a connected graph G with n vertices, $b(u) \geq \lceil \log_2 n \rceil$, since during each time unit the number of informed vertices can at most double.

We define the *broadcast time of a graph G*, $b(G)$, to be the maximum broadcast time of any vertex u in G, i.e., $b(G) = \max\{b(u) \mid u \in V(G)\}$. For the complete graph K_n with $n \geq 2$ vertices, $b(K_n) = \lceil \log_2 n \rceil$, yet K_n is not minimal with respect to this property. 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 use the term *broadcast graph* to refer to any graph G on n vertices with $b(G) = \lceil \log_2 n \rceil$.

We define the *broadcast function*, $B(n)$, to be the minimum number of edges in any broadcast graph on n vertices. A *minimum broadcast graph* (mbg) is a 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 theoretically possible.

2. Previous results

In [5], Farley, Hedetniemi, Mitchell and Proskurowski studied the broadcast function $B(n)$, i.e., the number of edges in a minimum broadcast graph with n vertices. In particular, they determined the values of $B(n)$ for $n \leq 15$ and showed that $B(2^k) = k2^{k-1}$. Mitchell and Hedetniemi [9] determined the value for $B(17)$. For $1 \leq n \leq 17$, the values of the broadcast function $B(n)$ are shown in Table 1.

The results of these studies suggest that mbg's are extremely difficult to find; in fact, no mbg with n vertices is known for any value of $n \geq 17$, except for the values $n = 2^k$, where mbg's are easy to construct, and for $n = 18$, $n = 19$, $n = 30$, and $n = 31$ which are presented in this paper. (The value of $B(18)$ and the mbg on 18 vertices are from an unpublished report by Wang [10].)

Since mbg's seem to be difficult to find, several authors have devised methods to construct broadcast graphs with small numbers of edges. We use the term *sparse broadcast graph* when referring to such graphs. Note that we use this term only in an intuitive sense. The number of edges in any broadcast graph on n vertices gives an upper bound on $B(n)$. In some cases, we are able to show that the new sparse broadcast graphs that we construct are mbg's.

In [4], Farley designed several techniques for constructing broadcast graphs with

Table 1. The broadcast function $B(n)$.

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$B(n)$	0	1	2	4	5	6	8	12	10	12	13	15	18	21	24	32	22

n vertices and approximately $(n/2)\log_2 n$ edges. Chau and Liestman [3] presented constructions based on Farley's techniques which yield somewhat sparser graphs for most values of n. In [6], Grigni and Peleg showed that $B(n) \in \Theta(L(n-1)n)$ for $n \geq 1$ where $L(k)$ denotes the exact number of consecutive leading 1's in the binary representation of k. Asymptotically, Grigni and Peleg's construction (which established their upper bound) produces the best known graphs for most values of n. Our interest is to improve the best known graphs for "practical" values of n.

Liestman and Peters [8] investigated the problem of broadcasting in time $O(\log_2 n)$ in graphs with bounded degree. The work contained in this paper was motivated, in part, by our efforts to improve these results. Our results on broadcasting in bounded degree graphs are included in [2]. For a survey of results on this problem and related problems, see Hedetniemi, Hedetniemi and Liestman [7].

3. Minimum broadcast graphs

In this section, we establish the value of $B(n)$ for $n = 18$, 19, 30, and 31. In doing so, we present several new minimum broadcast graphs. The graph on 18 vertices is closely related to the minimum broadcast graph on 17 vertices presented in [9]. The graphs on 19 and 30 vertices are constructed by general techniques which are described in Section 4. The graph on 31 vertices is an ad hoc construction which is similar to graphs constructed by these techniques.

In determining the value of $B(18)$, Wang [10] proved the following lemma which is useful in establishing lower bounds on $B(n)$. This result is unpublished and the proof is included here for the first time.

Lemma 1. *If a broadcast graph on n vertices, $2^{k-1}+1 < n \leq 2^k$, with e edges has a vertex of degree d, then $B(n-1) \leq e + \frac{1}{2}d(d-3)$.*

Proof. Let G be a broadcast graph on n vertices with e edges where $2^{k-1}+1 < n \leq 2^k$ and a vertex v of degree d. Let v's neighbours be $v_0, v_1, \ldots, v_{d-1}$. Construct a graph G' on $n-1$ vertices by deleting the vertex v (and the d edges incident on it) from G and adding any edges that are necessary to form a clique among the vertices $v_0, v_1, \ldots, v_{d-1}$. We claim that G' is a broadcast graph on $n-1$ vertices. If this is true, it follows that $B(n-1) \leq e + \frac{1}{2}d(d-1) - d = e + \frac{1}{2}d(d-3)$.

In order to show that G' is a broadcast graph, we need only show that each vertex in G' can broadcast in $\lceil \log_2(n-1) \rceil$ time. Note that due to the restriction on n, $\lceil \log_2(n-1) \rceil = \lceil \log_2 n \rceil$.

Consider any broadcast scheme for any originator in G and let u be the vertex that informs vertex v. Without loss of generality, assume that $u = v_0$ and that this call occurs at time t_0. In this scheme, v may subsequently call some of its neighbours, say $v_1, v_2, \ldots, v_{k-1}$ at times $t_1, t_2, \ldots, t_{k-1}$, respectively, where $t_0 < t_1 < \cdots < t_{k-1}$.

A broadcast scheme for G' can easily be obtained from the scheme for G by deleting the calls involving vertex v and adding calls from v_i to v_{i+1} at time t_i where $0 \leq i < k-1$. Thus, each vertex v_{i+1} is informed at time t_i and is ready to continue with other calls by time $t_i + 1 \leq t_{i+1}$ as in the scheme for G. □

Theorem 2. $B(18) = 23$.

Proof. The graph in Fig. 1 is a broadcast graph on 18 vertices with 23 edges from [10]. Six labelled spanning subtrees of this graph are shown in Fig. 2. Each tree describes a broadcast scheme for originators isomorphic to those vertices indicated by "+". Only those edges on which calls are made are shown. At time 1, the originator calls the other vertex marked with a "+". The other labels indicate the times at which the message is received by the remaining vertices. From this, we conclude that $B(18) \leq 23$.

Suppose that there is a broadcast graph on 18 vertices with fewer than 23 edges. Clearly such a graph must contain a vertex of degree 2. If such a graph exists, then a broadcast graph with 17 vertices and no more than 21 edges could be constructed by deleting a vertex of degree 2 and adding an edge between its two neighbours (as in the proof of Lemma 1). This would contradict the fact that $B(17) = 22$. Thus, no such graph can exist, so the graph in Fig. 1 is a minimum broadcast graph, and $B(18) = 23$. □

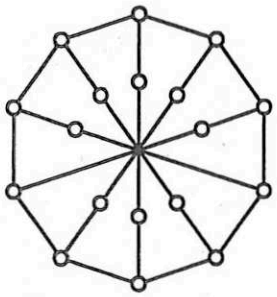


Fig. 1. Mbg on 18 vertices.

Theorem 3. $B(30) = 60$.

Proof. $B(30) \geq 60$ since a minimum broadcast graph on 30 vertices can contain no vertices of degree less than 4. Suppose, to the contrary, that u is a vertex of degree 3 in a minimum broadcast graph G on 30 vertices. Since G is a minimum broadcast graph, each vertex in G must be able to broadcast in $\lceil \log_2 30 \rceil = 5$ time units. If u is the originator of a message to be broadcast, u can inform it's three neighbours no sooner than at times 1, 2 and 3. Each of these vertices can be considered to be the originator of the message in some subgraph of G. In any of these subgraphs, the number of informed vertices can at most double in each time unit so these subgraphs can contain no more than 16, 8 and 4 vertices, respectively, by time 5. Since u can inform no additional vertices directly, at most 29 vertices can be inform-

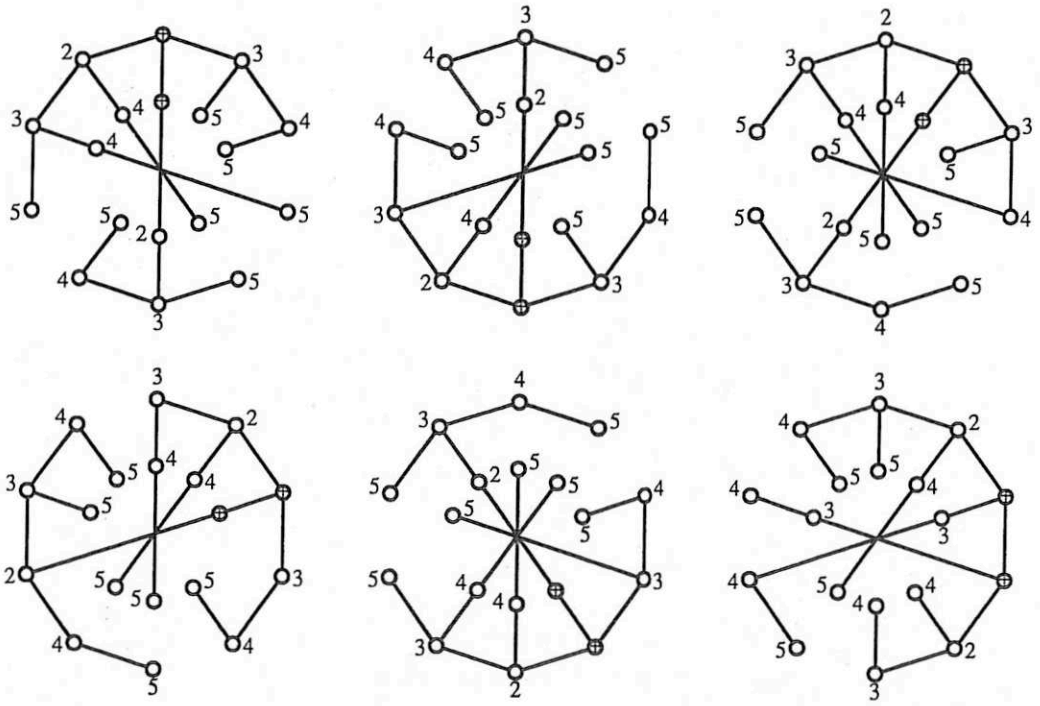


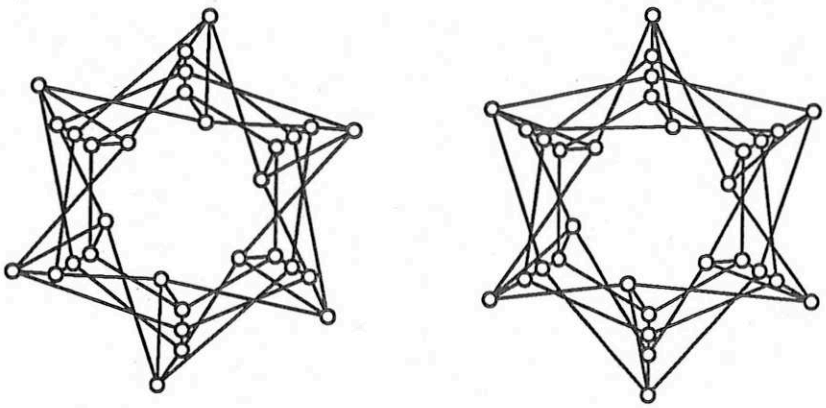
Fig. 2. Broadcast schemes for the mbg on 18 vertices.

ed in 5 time units by any broadcast scheme originating at u. Thus, $B(30) \geq 60$.

The graphs in Figs. 3 and 4(a) show that $B(30) \leq 60$. Each graph has 60 edges and allows any vertex to broadcast in 5 time units.

The graphs of Fig. 3 are formed by arranging six cycles of length 5 in a circle and adding a matching between the vertices of each pair of "adjacent" cycles. (These graphs are constructed by the technique described in Section 4.2.) It is interesting to note that for both of these graphs, the subgraph induced by any pair of "adjacent" cycles is a Petersen graph.

The graph of Fig. 4(a) is constructed by adding chords of lengths 5 and 7 to a



(a) (b)

Fig. 3. Mbg's on 30 vertices.

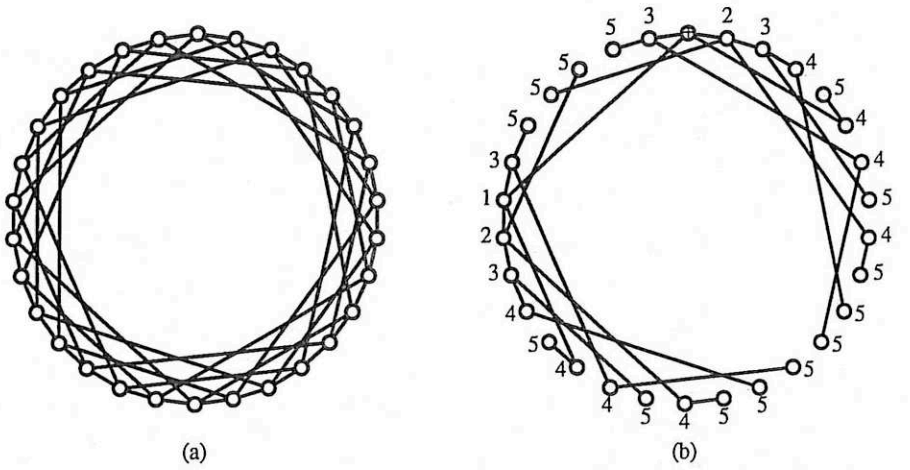


Fig. 4. (a) Mbg on 30 vertices; and (b) broadcast scheme.

cycle of length 30 (in particular, we connect vertices $2i$ to $2i+5$ and $2i-7$). (This graph is constructed by the technique described in Section 4.1.)

The vertices of Fig. 3(a) can be partitioned into five classes and a broadcast scheme can be described for each type of vertex. The details of the broadcast schemes for this graph appear in the appendix. In the appendix, it is assumed that the vertices are numbered clockwise within each cycle starting with the outermost vertex and that the cycles are also considered in clockwise order.

The graph of Fig. 3(b), suggested by Alspach [1], is vertex transitive. The details of the broadcast scheme for this graph also appear in the appendix where the vertices are numbered in the same way as the vertices of Fig. 3(a).

The graph of Fig. 4(a) is also vertex transitive. Fig. 4(b) is a broadcasting scheme for this graph. The originator is marked with "+". The labels on the other vertices indicate the times at which the message is received.

Theorem 4. $B(31)=65$.

Proof. The graph in Fig. 5 is a minimum broadcast graph with 65 edges on 31 vertices. Four labelled spanning subtrees of this graph are shown in Fig. 6. Each tree consists of the edges used in a broadcast scheme for originators isomorphic to those vertices indicated by "+". At time 1, the originator calls the other vertex marked with a "+". The other labels indicate the times at which the message is received by the remaining vertices.

It is easy to show that $B(31)\geq 65$. As in the proof of Theorem 3, no vertex of degree less than 4 can exist in any broadcast graph on 31 vertices. Suppose that the originator of a message in such a graph is a vertex of degree 4 which informs its neighbours at times 1, 2, 3 and 4. Each of these neighbours can be considered to be the originator of a message to be broadcast in a subgraph of the broadcast graph. By time $\lceil \log_2 31 \rceil = 5$, the numbers of informed vertices in these four subgraphs are

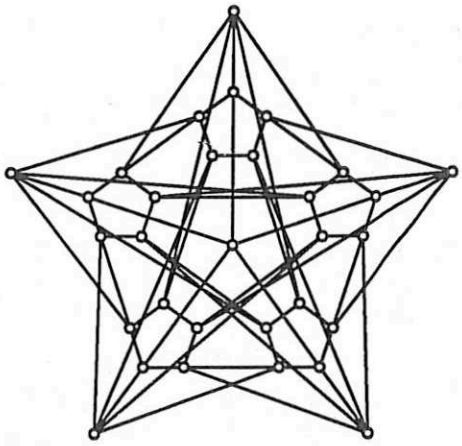


Fig. 5. Mbg on 31 vertices.

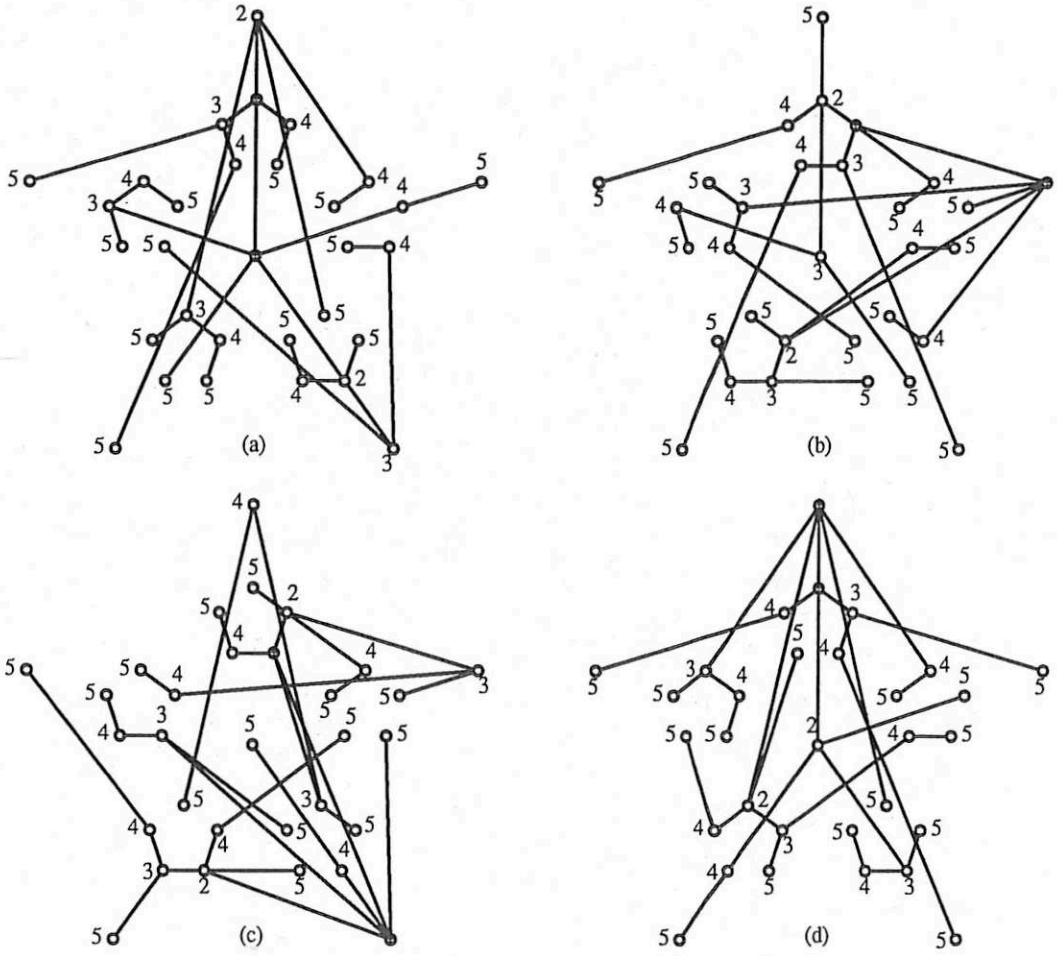


Fig. 6. Broadcast schemes for mbg on 31 vertices.

at most 16, 8, 4 and 2, respectively. Furthermore, the numbers of informed vertices must be at least 16, 8, 4 and 2 to guarantee that a total of 31 vertices are informed in 5 time units. The neighbour that is informed first must have degree greater than

4 since otherwise it could not inform 15 other vertices in the remaining 4 time units. Thus, every degree 4 vertex must be adjacent to a vertex of degree at least 5. A simple counting argument shows that this cannot be done with fewer than 65 edges. Thus, the graph in Fig. 5 is a minimum broadcast graph and $B(31)=65$. □

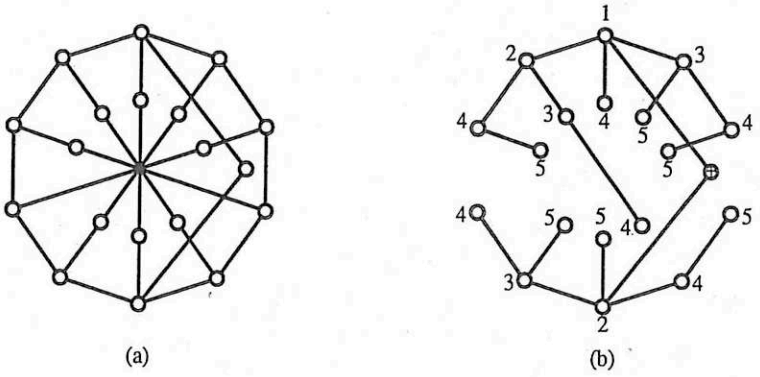
Theorem 5. $B(19)=25$.

Proof. In each of the broadcast schemes of Fig. 2 for the minimum broadcast graph on 18 vertices, either the uppermost or the lowermost vertex is idle at time 5. A new vertex, connected only to these two vertices, can obviously be informed at time 5 by a simple extension of these schemes. The graph formed by adding this new vertex is shown in Fig. 7(a). The scheme in Fig. 7(b) shows that the new vertex can also broadcast in 5 time units. Thus, this graph is a broadcast graph on 19 vertices. (Note that this graph is constructed by the method of vertex addition which is described in Section 4.3.)

A nontrivial case analysis based on the numbers of vertices of various degrees shows that 25 edges are necessary in any broadcast graph on 19 vertices. A sketch of the proof follows.

Suppose that there are broadcast graphs with 19 vertices and fewer than 25 edges. Let G be such a graph with 24 edges, and let n_i denote the number of vertices of degree i in G. Thus, $\sum_i n_i=19$ and $\sum_i in_i=48$. Since every vertex must be able to inform 18 other vertices in 5 time units, we cannot have vertices of degree less than 2, i.e., $n_0=n_1=0$. It now follows that $\sum_i in_{i+2}=10$.

There are further degree restrictions: a vertex of degree 2 must have at least one neighbour of degree greater than 2 – otherwise, it can inform at most 14 other vertices in 5 time units. To summarize the remaining restrictions, we introduce the following notation. Let G^* be the multigraph obtained from G by replacing each path with all inner vertices of degree 2 by an edge joining its endpoints. To keep track of the structure of G, we mark each edge of G^* with small strokes indicating the degree 2 vertices of G that were removed from it. It follows from above that each edge of G^* is marked by at most 2 strokes. It is also easy to see that vertex



(a) (b)

Fig. 7. (a) Mbg on 19 vertices; and (b) broadcast scheme.

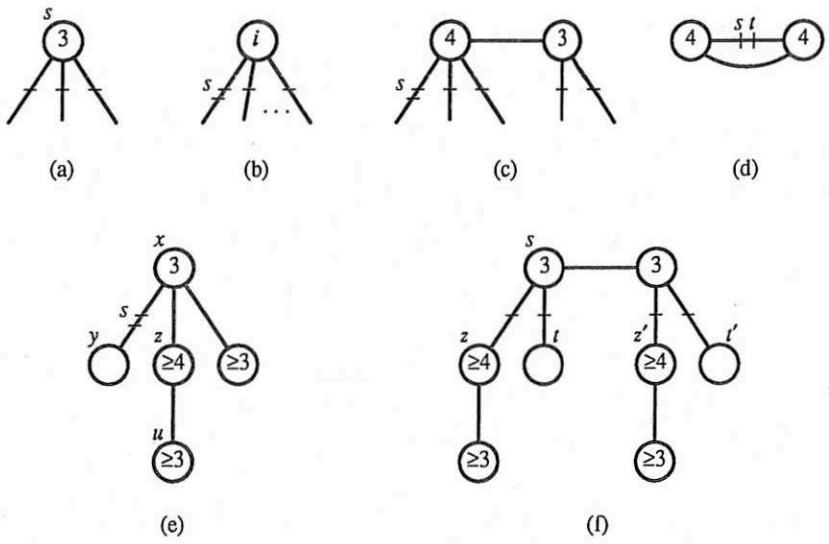


Fig. 8. Configurations.

s in configuration (a) of Fig. 8 cannot broadcast to 18 other vertices in 5 time units. Similar arguments show that broadcasts from the degree 2 vertices labelled s in configurations (b) and (c) are impossible in 5 time units. To prove that configuration (d) is impossible, we verify that the only way broadcasts from s and t can both reach all 19 vertices in 5 time units requires G to have at least 4 vertices of degree 4 or greater and at least 3 more vertices of degree 3 or greater. This contradicts $\sum_i in_{i+2} = 10$. Configurations (e) and (f) illustrate further restrictions on vertices of degree 3. Configuration (e) shows the only possible context for a vertex x of degree 3 incident with an edge marked by two strokes. An attempt to broadcast from vertex s in configuration (e) also shows that all of the vertices in the figure are distinct. Configuration (f) shows the only possible context for an adjacent pair of degree 3 vertices each with two degree 2 neighbours. Broadcasting from vertex s in configuration (f) also shows that vertices z and t are distinct from z' and t' when the degree of z or z' is exactly 4.

Denote by $q(x)$ the total number of strokes on all edges of G^* incident with x. Note that configurations (a) and (e) imply that $q(x) \leq 2$ for a vertex x of degree 3. Furthermore, (b) implies that $q(x) \leq 2d-2$ for a vertex x of any degree d because if some edge has two marks, then some other edge has no marks. Therefore, $2n_2 = \sum_x q(x) \leq 2n_3 + 6n_4 + 8n_5 + \cdots$, or (dividing by 2 and substituting $\sum_i n_i = 19$) $19 \leq 2n_3 + 4n_4 + 5n_5 + \cdots$. If we subtract this inequality from $2\sum_i in_{i+2} = 20$, we obtain $n_5 + 2n_6 + 3n_7 + \cdots \leq 1$. Thus, if G (and G^*) were to exist, they would have no vertex of degree greater than 5, and either zero or one vertex of degree 5. We consider these two cases separately using the following notation. Let α be the number of edges of G^* marked by 2 strokes and β the number of edges of G^* not marked by any strokes. Counting, in two ways, the number of edges of G between a vertex of degree 2 and a vertex of degree higher than 2, produces the equation $2(n_2 - \alpha) = \sum_{i \geq 3} in_i - 2\beta$, which can be rewritten (using $\sum_i in_i = 48$) as $2(n_2 - 12) = \alpha - \beta$.

In the case that $n_5 = 1$, all of the inequalities derived in the preceding paragraph are, in fact, equalities. In particular, $q(x) = 2$ for each vertex of degree 3, $q(x) = 6$ for each vertex of degree 4, and $q(x) = 8$ for the unique vertex of degree 5. It is easy to see from Fig. 8(b) and (e) that this can only be accomplished if each vertex of degree 4 or 5 is incident with precisely one unmarked edge while all other incident edges are marked by two strokes, and if each vertex of degree 3 is incident with one unmarked edge and two edges marked by one stroke each. Thus $\alpha = \frac{1}{2}2(3n_4 + 4n_5)$, and n_4 must be even. Because $n_3 + 2n_4 = 7$ (from the equality $\sum_i i n_{i+2} = 10$), it follows that n_4 is either 0 or 2. Since there are at least four edges incident to the degree 5 vertex that are marked by two strokes, $\alpha \geq 4$ and $n_4 = 0$ is impossible. Thus $n_4 = 2$ and $n_3 = 3$, $n_2 = 13$, $\alpha = 5$ and $\beta = 3$. Each vertex of G^* is incident with at most one unmarked edge, so one of the three unmarked edges must join two vertices of degree 3 or a vertex of degree 3 with a vertex of degree 4. It can be verified that this is impossible by examining configurations (c) and (f).

In the case that $n_5 = 0$, we have $n_3 + 2n_4 = \sum_i i n_{i+2} = 10$. Furthermore, the inequality $n_5 + 2n_6 + 3n_7 + \cdots = 0 \leq 1$ has a slack of 1, so the inequality $\sum_x q(x) \leq 2n_3 + 6n_4 + 8n_5 + \cdots$ from which it was derived has a slack of 2, and it follows that $2n_2 = \sum_x q(x) = 2n_3 + 6n_4 - 2$. If there is a vertex of degree 3 incident with an edge marked by 2 strokes, then by configuration (e), there is a vertex z of degree $d = 4$ with $q(z) \leq 2d - 4$. Since $\sum_x q(x) = 2n_3 + 6n_4 - 2 = 2n_3 + 4 + 6(n_4 - 1)$, it must be the case that $q(x) = 2d - 2 = 6$ for any other vertex x of degree $d = 4$, and $q(x) = 2$ for any x of degree 3. It can be seen from configuration (e) that vertex u is the only possible candidate for a second degree 3 vertex incident with an edge marked by two strokes, since any other location implies a second degree 4 vertex z' with $q(z') \leq 2d - 4$. Thus, $\alpha = \frac{1}{2}\lfloor (4 + 3(n_4 - 1)) \rfloor$, and the situation can be reduced to the three subcases: $n_4 = 2$ ($n_3 = 6$, $n_2 = 11$, $\alpha = 3$, $\beta = 5$), $n_4 = 3$ ($n_3 = 4$, $n_2 = 12$, $\alpha = 5$, $\beta = 5$), and $n_4 = 4$ ($n_3 = 2$, $n_2 = 13$, $\alpha = 6$, $\beta = 4$). In each of these cases we obtain a contradiction with Fig. 8.

If $n_5 = 0$ and no vertex of degree 3 is incident with a doubly marked edge, then $\sum_x q(x) = 2n_3 + 6n_4 - 2$ implies that, among the vertices of degree 4, at most one can have $q(x) = 4$ or at most two can have $q(x) = 5$. Therefore, 2α is between $3n_4 - 2$ and $3n_4$. Furthermore, if $2\alpha = 3n_4$, then $q(x) = 6$ for every vertex x of degree 4. We obtain a list of cases for $n_4 = 0, 1, \ldots, 5$. In each case, a contradiction with Fig. 8 can be obtained. For example, if $n_4 = 0$ ($n_3 = 10$, $n_2 = 9$, $\alpha = 0$, $\beta = 6$), then each of the ten degree 3 vertices is incident with at least one of the six unmarked edges by configuration (a). This implies that there is a pair of adjacent degree 3 vertices each with two degree 2 neighbours in contradiction with configuration (f). Of the remaining cases, the two most involved cases are $n_4 = 4$ and $\beta = 4$ ($n_3 = 2$, $n_2 = 13$, $\alpha = 6$) and $n_4 = 5$ ($n_3 = 0$, $n_2 = 14$, $\alpha = 7$, $\beta = 3$). In the case $n_4 = 4$ and $\beta = 4$, $2\alpha = 3n_4$ so $q(x) = 6$ for every vertex x of degree 4. This fact and configuration (b) imply that each of the degree 4 vertices is incident with one unmarked edge and three doubly marked edges, that the single edge with one stroke must connect the two degree 3 vertices, and that each degree 3 vertex is adjacent to two degree 4 vertices. It is impossible to add the six edges with two strokes to this configuration in a way

that allows every degree 2 vertex to broadcast in 5 time units. In the case $n_4 = 5$, each degree 4 vertex is incident to at least one unmarked edge by configuration (b) and there are no doubly marked edges between adjacent degree 4 vertices by configuration (d). The only possible configuration is a path of two degree 4 vertices and a path of three degree 4 vertices with six doubly marked edges between the two paths. The seventh doubly marked edge must be between the end vertices of the longer path and it can be shown that the degree 2 vertices on this edge cannot broadcast in 5 time units. □

4. Construction of sparse broadcast graphs

Several techniques have proven valuable in attempting to construct improved sparse broadcast graphs for small n. Since these techniques result in highly symmetric graphs, only a few broadcast schemes need to be exhibited to prove that the graph is a broadcast graph. We feel that some of these techniques can be exploited to make further improvements for larger n. We have shown complete proofs (including broadcast schemes) for at least one graph constructed by each technique. Additional sparse broadcast graphs that we have constructed by these techniques are included in Section 4.5 without complete proofs. The proofs are included in the appendix.

4.1. Cycles with chords

One type of construction which yields several new sparse broadcast graphs can be described as a cycle with appropriately chosen chords. Let C_n be the cycle on n vertices $0, 1, \ldots, n-1$ with edges $(i, i+1)$, $0 \leq i \leq n-1$. (Note that addition is assumed to be performed modulo n throughout this section.) We have used three methods to add chords to this cycle.

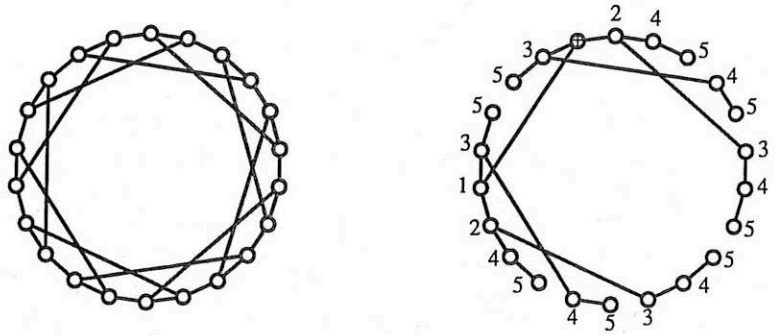
For even n, we can construct a 3-regular graph by choosing odd α and adding edges $(i, i+\alpha)$ for all even i, $0 \leq i \leq n-2$. With this technique, we can construct the following new sparse broadcast graphs.

Theorem 6. $B(22) \leq 33$.

Proof. The graph in Fig. 9(a) is a broadcast graph, constructed by choosing $\alpha = 5$. All of the vertices of the graph are isomorphic. A broadcast scheme for a vertex of the graph is shown in Fig. 9(b). □

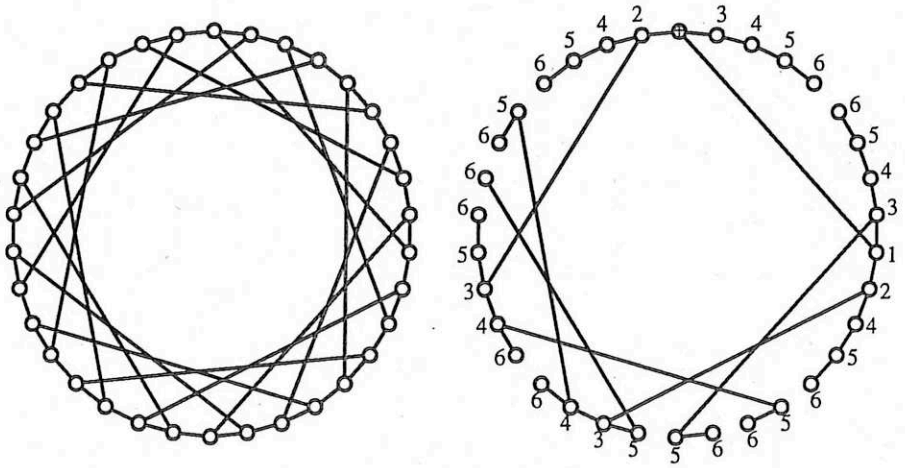
Theorem 7. $B(34) \leq 51$.

Proof. The graph in Fig. 10(a) is a broadcast graph, constructed by choosing $\alpha = 9$. All of the vertices of the graph are isomorphic. A broadcast scheme for a vertex of the graph is shown in Fig. 10(b). □



(a) (b)

Fig. 9. (a) Broadcast graph on 22 vertices; and (b) broadcast scheme.



(a) (b)

Fig. 10. (a) Broadcast graph on 34 vertices; and (b) broadcast scheme.

Another way to construct a 3-regular graph by adding chords to the cycle C_n when $n \equiv 0 \bmod 6$ is as follows. Choose integer $\alpha \equiv 1 \bmod 3$. Add edges $(i, i+\frac{1}{2}n)$ for all $i \equiv 0 \bmod 3$ and edges $(i, i+\alpha)$ for all $i \equiv 1 \bmod 3$. With this variation, we can construct the following new sparse broadcast graph.

Theorem 8. $B(24) \leq 36$.

Proof. The graph in Fig. 11(a) is a broadcast graph, constructed by choosing $\alpha = 7$. The vertices of the graph can be partitioned into two isomorphism classes. The broadcast scheme shown in Fig. 11(b) begins with a call between two nonisomorphic vertices (indicated by "+" signs). Subsequent calls are made as indicated. (For a different broadcast graph with 24 vertices and 36 edges, see Theorem 11 below.) □

For even n, we can construct a 4-regular graph by choosing odd integers α and β and adding the chords $(i, i+\alpha)$ and $(i, i-\beta)$ to C_n for all even i, $0 \leq i \leq n-2$. Note that one of the minimum broadcast graphs on 30 vertices, Fig. 4(a), was constructed

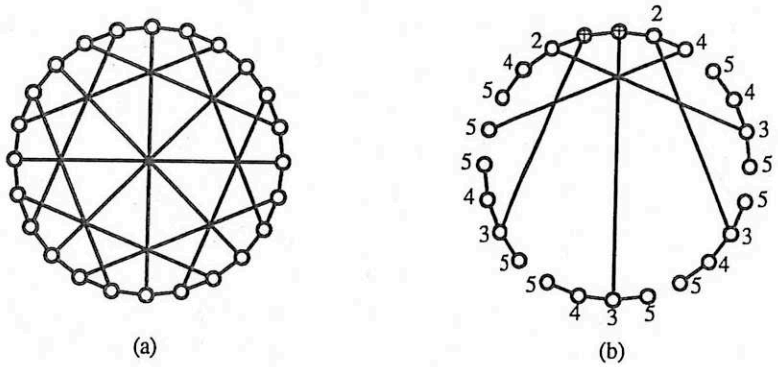


Fig. 11. (a) Broadcast graph on 24 vertices; and (b) broadcast scheme.

with this technique. In this manner, we can construct the following new sparse broadcast graph.

Theorem 9. $B(56) \leq 112$.

Proof. The graph in Fig. 12(a) is a broadcast graph, constructed by choosing $\alpha = 7$ and $\beta = 11$. All of the vertices of the graph are isomorphic. A broadcast scheme for a vertex of the graph is shown in Fig. 12(b). □

4.2. Interconnection of cycles

Another type of construction which yields several new sparse broadcast graphs can be described as the interconnection of cycles. In particular, when n can be factored as $n = ab$, we can construct a graph on n vertices by interconnecting b cycles

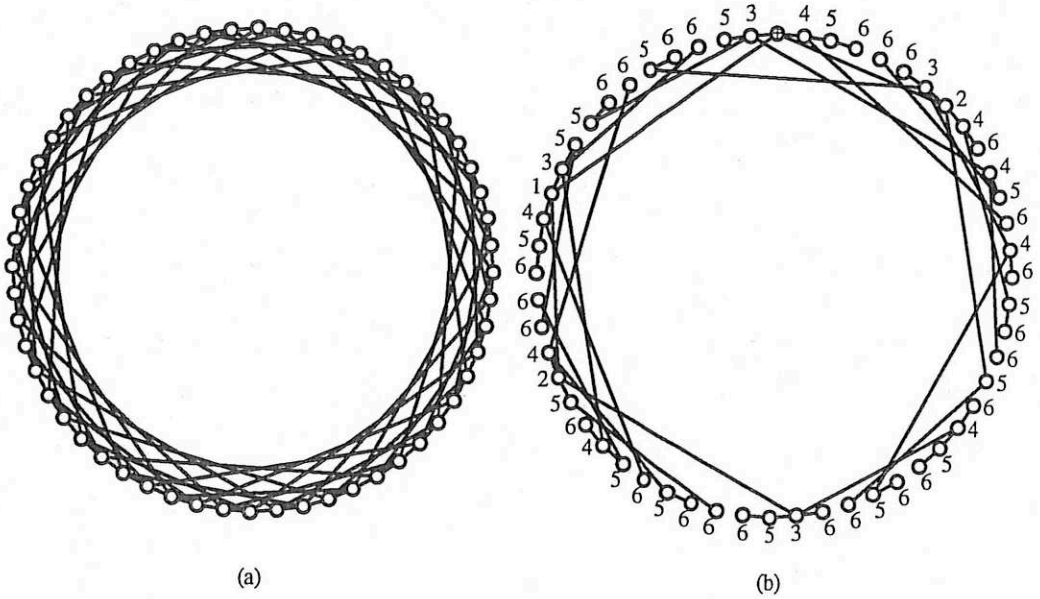


Fig. 12. (a) Broadcast graph on 56 vertices; and (b) broadcast scheme.

of length a. Start by labelling the n vertices with the pairs (i,j) where $0 \le i \le a-1$ and $0 \le j \le b-1$ and add the edges $((i,j),((i+1) \bmod a, j))$ for all i and j. This creates b cycles of length a where (i,j) is the label of the vertex in position i of cycle j.

The interconnections among the cycles can be described by two mappings $f: \{0,1,\ldots,a-1\} \to \{0,1,\ldots,a-1\}$ and $g: \{0,1,\ldots,a-1\} \to \{0,1,\ldots,b-1\}$ so that vertex (i,j) is adjacent to vertex $((i+f(i)) \bmod a, (j+g(i)) \bmod b)$.

The 20 vertex graph in Fig. 13(a) is an interconnection of four cycles of length 5. If the positions of each cycle are labelled clockwise, starting with the degree 4 vertex in position 0, and the cycles are also labelled clockwise, then the interconnection functions are $f(0)=0, f(2)=1, f(3)=4$, and $g(0)=1, g(2)=3, g(3)=1$. For completeness, we also specify $f(1)=0, f(4)=0, g(1)=0$, and $g(4)=0$, so that the vertices in positions 1 and 4 of each cycle are "adjacent to themselves".

For the purposes of this paper, the interconnections among cycles can be more succinctly described by a *template* with a columns corresponding to the positions on cycles and two rows which list the values of the interconnection functions f and g. For example, the template for the graph in Fig. 13(a) is

$$\begin{matrix} 00140 \\ 10310 \end{matrix}.$$

One of the known minimum broadcast graphs on 15 vertices (see [9]) is an interconnection of three cycles of length 5 with template

$$\begin{matrix} 33223 \\ 21211 \end{matrix}.$$

As mentioned earlier, the 30 vertex mbg's in Fig. 3 are both formed from six cycles of length 5 by adding a matching between each pair of "adjacent" cycles. If the positions of each cycle in Fig. 3(a) are labelled clockwise with the outermost vertices labelled 0, then the template for this graph is

$$\begin{matrix} 13024 \\ 11111 \end{matrix}.$$

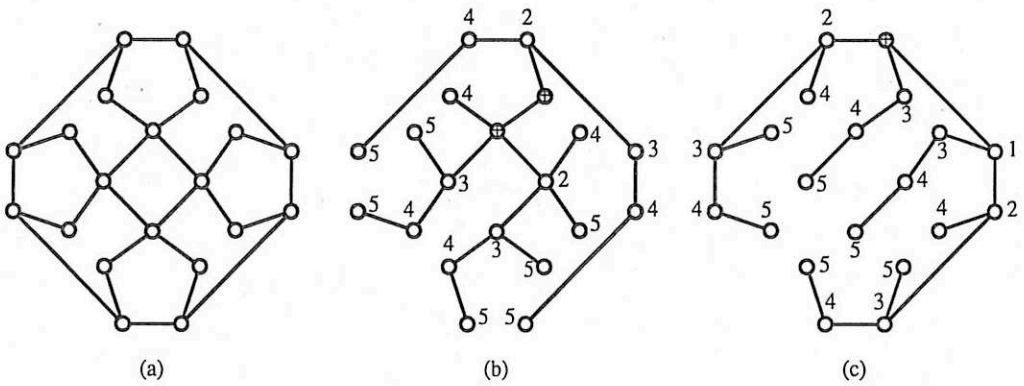


Fig. 13. (a) Broadcast graph on 20 vertices; and (b-c) broadcast schemes.

However, the specification of the graph in Fig. 3(b) requires two templates. Label the positions on the cycles as in Fig.3(a) and label the cycles clockwise with the top cycle labelled 0. The template for cycles 0, 2, and 4 is

$$\begin{matrix} 13024 \\ 11111 \end{matrix}',$$

which is the same as for the graph in Fig. 3(a). The template for cycles 1, 3, and 5 is

$$\begin{matrix} 34012 \\ 11111 \end{matrix}',$$

which specifies the inverse of the first template.

The following new sparse broadcast graphs can also be constructed in this manner.

Theorem 10. $B(20) \leq 28$.

Proof. The graph in Fig. 13(a) is a broadcast graph. It has template

$$\begin{matrix} 00140 \\ 10310 \end{matrix}$$

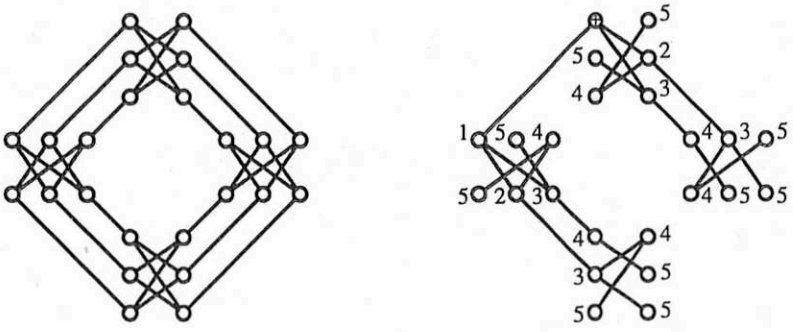
as explained above.

The vertices of the graph can be partitioned into three classes. Fig. 13(b) and (c) describe broadcast schemes for the three classes of vertices. Each tree describes a broadcast scheme for originators isomorphic to those vertices indicated by "+". □

Theorem 11. $B(24) \leq 36$.

Proof. The graph in Fig. 14(a) is a broadcast graph. The template is

$$\begin{matrix} 333333 \\ 131313 \end{matrix}'.$$



(a) (b)

Fig. 14. (a) Broadcast graph on 24 vertices; and (b) broadcast scheme.

The details of a broadcast scheme are shown in Fig. 14(b). (Also see Theorem 8.) □

Theorem 12. $B(25) \leq 40$.

Proof. The graph in Fig. 15(a) is a broadcast graph. The template is

14314

23123

when the vertices of the cycles are labelled clockwise starting with the degree 4 vertex.

Broadcast schemes are shown in Fig. 15(b-d). Each tree describes a broadcast scheme for originators isomorphic to those vertices indicated by "+". □

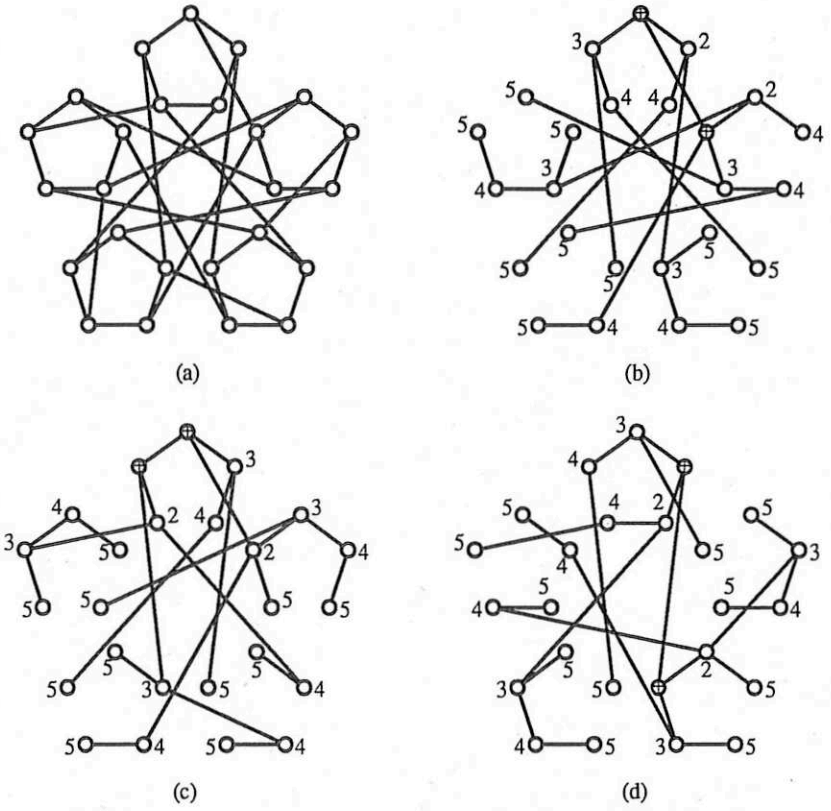


Fig. 15. (a) Broadcast graph on 25 vertices; and (b-d) broadcast schemes.

For $n = 28$, we can construct a good graph by a variation of this technique. In this case, instead of interconnecting cycles, we have interconnected minimum broadcast graphs. This may also be seen as a generalization of Farley's construction (see [4]).

Theorem 13. $B(28) \leq 52$.

Proof. The graph in Fig. 16 is a broadcast graph. The vertices of this graph can be

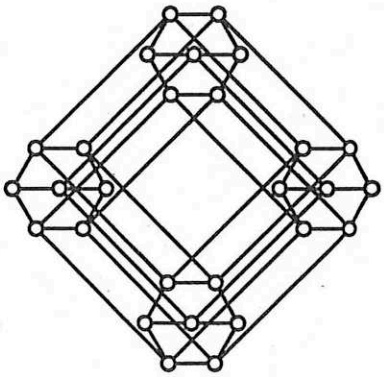


Fig. 16. Broadcast graph on 28 vertices.

partitioned into three classes. The details of the broadcast schemes are shown in Fig. 17. □

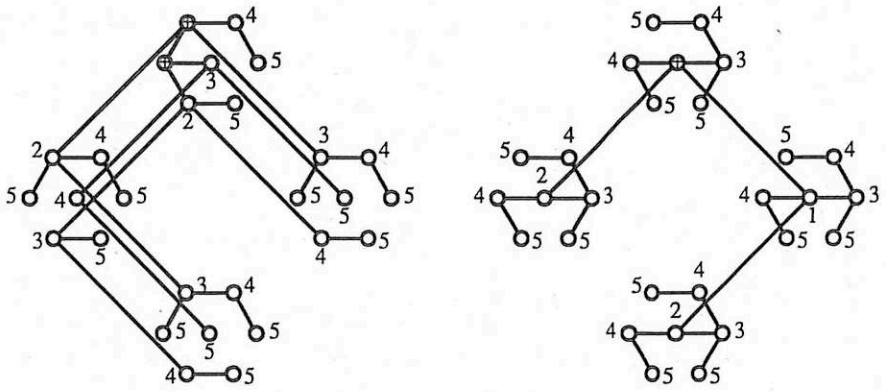


Fig. 17. Broadcast schemes for broadcast graph on 28 vertices.

4.3. Vertex addition

Many of the broadcast schemes above leave vertices idle during the last time unit. These vertices could, potentially, be used to inform additional vertices. It is sometimes possible to add new vertices and edges such that the existing broadcast schemes can be extended to include the new vertices, and the new vertices can originate their own minimum time broadcasts. In some cases, the numbers of edges added are small enough to give broadcast graphs with fewer edges than the best previously known graphs. This technique was used to produce the new minimum broadcast graph on 19 vertices (see Fig. 7(a)) as well as the following new sparse broadcast graph.

Theorem 14. $B(26) \leq 48$.

Proof. The graph in Fig. 18(a) is a broadcast graph which was constructed by ad-

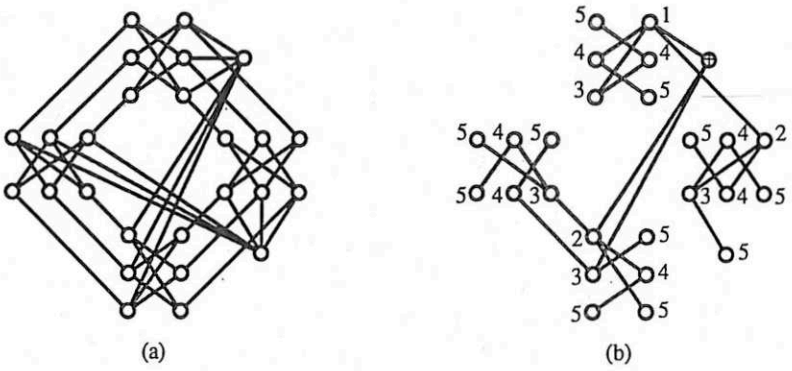


Fig. 18. (a) Broadcast graph on 26 vertices; and (b) broadcast scheme.

ding two vertices and 12 edges to the graph of Fig. 14(a). Note that in the broadcast scheme in Fig. 14(b), the vertices which are labelled with "+", "1" and "2" are idle at time 5. Since all of the vertices of this graph are isomorphic, there will always be two adjacent vertices in two "adjacent" 6-cycles which are not involved in calls at time 5. Thus, we know that we can easily extend the broadcast scheme of Fig. 14(b) to inform the two new vertices by time 5. Since the two new vertices are isomorphic, the broadcast scheme given in Fig. 18(b) completes the proof. □

4.4. Vertex deletion

For many values of n, our best result has come from applying the construction contained in the proof of Lemma 1 to graphs constructed by other techniques.

Theorem 15. $B(33) \leq 51$.

Proof. $B(33) \leq 51$ follows from $B(34) \leq 51$ using Lemma 1 to replace any of the vertices in the graph of Fig. 10(a) to obtain a graph such as Fig. 19. The broadcast schemes can be derived from the broadcast scheme of Fig. 10(b) as described in the proof of Lemma 1. □

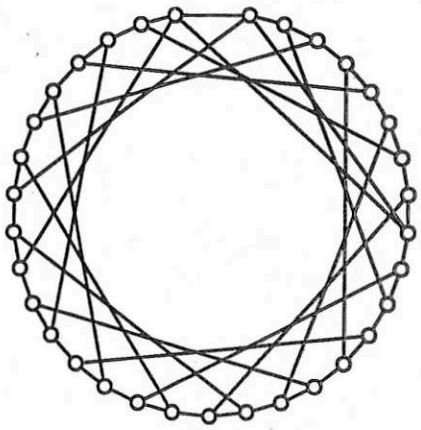


Fig. 19. Broadcast graph on 33 vertices.

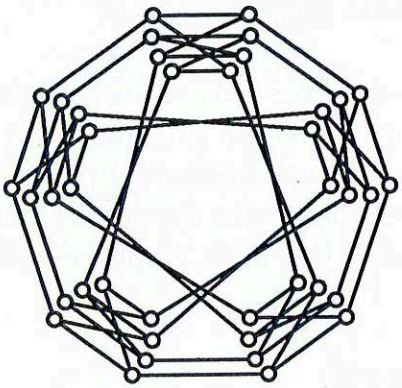


Fig. 20. Broadcast graph on 40 vertices.

The following sparse broadcast graphs on 27 vertices are constructed with a variation of the technique of Section 4.2. In this case, in addition to the edges corresponding to the templates and to the cycles, some extra edges are included.

Theorem 20. $B(27) \leq 51$.

Proof. The graphs in Fig. 21 are broadcast graphs formed by interconnecting nine

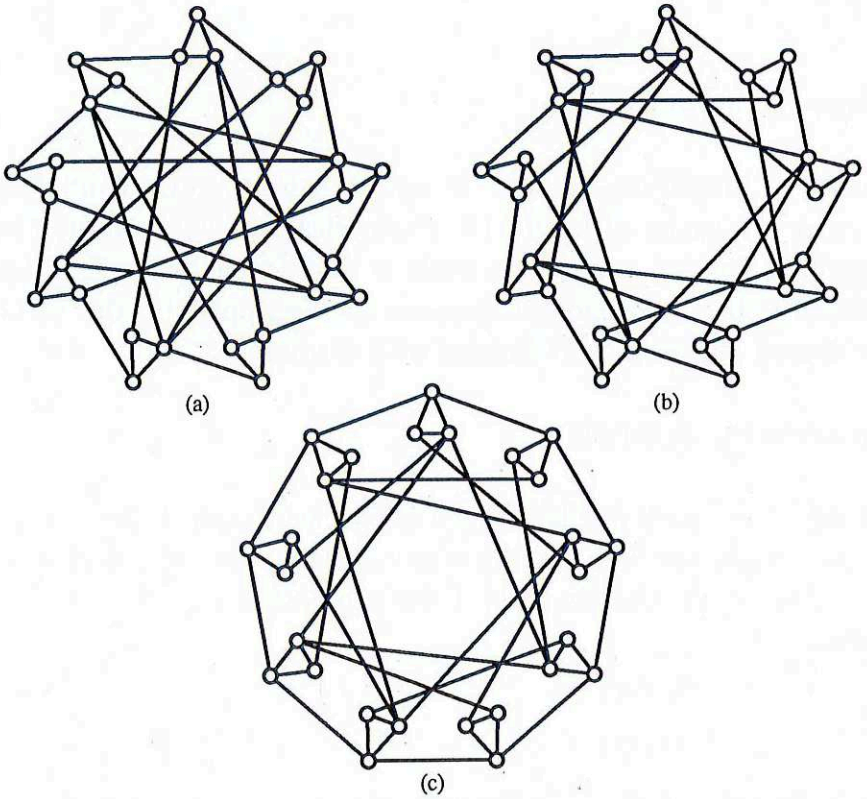


Fig. 21. Broadcast graphs on 27 vertices.

4.5. Additional sparse broadcast graphs

In Sections 4.1–4.4, we have shown complete proofs (including broadcast schemes) for at least one graph constructed by each technique. Additional sparse broadcast graphs can be constructed by these techniques and are briefly described here. The broadcast schemes for the graphs in this section appear in the appendix.

The following sparse broadcast graphs are obtained by the techniques of Section 4.1.

Theorem 16. $B(36) \leq 54$.

Proof. A broadcast graph on 36 vertices can be constructed by the same technique as the graph on 24 vertices of Fig. 3(a) by choosing $\alpha = 10$. □

Theorem 17. *For even* n, $44 \leq n \leq 54$, $B(n) \leq 2n$.

Proof. We can construct a broadcast graph for each even n in this range using the same technique as for the graph on 56 vertices of Fig. 12(a) by choosing $\alpha = 7$ and $\beta = 11$. The broadcast schemes in the appendix assume that vertices are numbered clockwise around the cycle C_n. □

The following sparse broadcast graphs are obtained by the technique of Section 4.2.

Theorem 18. $B(38) \leq 57$.

Proof. A broadcast graph on 38 vertices with 57 edges can be formed by interconnecting two cycles of length 19. The cycles are interconnected by adding a perfect matching so that vertex i in cycle 0 is adjacent to vertex $5i \bmod 19$ in cycle 1, $0 \leq i \leq 18$. In the broadcast schemes in the appendix, the vertices of cycle 1 are numbered 19 through 37 instead of 0 through 18. □

Theorem 19. $B(40) \leq 60$.

Proof. The graph in Fig. 20 is a broadcast graph. Label the positions of the top 8-cycle beginning with 0 at the vertex at the upper right of that cycle and continuing "clockwise" around the cycle. Label the positions of the other cycles similarly. The template is

75317531
13241324

In the broadcast scheme in the appendix, the cycles are numbered in clockwise order starting with cycle 0, so vertex i on cycle j is numbered $8j + i$. □

cycles of length 3. The graphs in Fig. 21(a), (b) and (c) have edges corresponding to the templates

$$\begin{matrix}212\\145\end{matrix}', \quad \begin{matrix}212\\172\end{matrix}', \quad \text{and} \quad \begin{matrix}012\\172\end{matrix}',$$

respectively, where position 0 for all three graphs is the outermost vertex of each cycle as depicted in the figure. In addition to these edges, each of the graphs has six additional edges forming two triangles. The appendix contains the broadcast schemes only for the graph of Fig. 21(a). In the appendix, each vertex (i,j) is numbered $3j+i$. □

The following sparse broadcast graphs are obtained by the vertex addition technique of Section 4.3. The only broadcast schemes included in the appendix are the schemes for the added vertices since the schemes for the remaining vertices are easily derived from the schemes for the graphs to which the vertices were added.

Theorem 21. $B(21) \leq 30$.

Proof. A broadcast graph on 21 vertices with 30 edges can be formed by adding a new vertex (vertex 20) and two new edges to the broadcast graph on 20 vertices shown in Fig. 13(a). In particular, we connect vertex 20 to vertices (1,0) and (1,2) of the 20 vertex graph. In the broadcast scheme in the appendix, each vertex (i,j) from the 20 vertex graph is numbered $5j+i$. □

Theorem 22. $B(23) \leq 35$.

Proof. A broadcast graph on 23 vertices with 35 edges can be formed by adding a new vertex and two new edges to the broadcast graph on 22 vertices shown in Fig. 9(a). In particular, we add vertex 22 and connect it to vertices 0 and 11 of the 22 vertex graph. □

Theorem 23. $B(41) \leq 70$ *and* $B(42) \leq 80$.

Proof. A broadcast graph on 41 vertices with 70 edges can be formed by adding a new vertex numbered 40 and ten new edges to the broadcast graph on 40 vertices shown in Fig. 20. In particular, connect the new vertex to vertices 2 and 6 in each of the 8-cycles of the 40 vertex graph.

A broadcast graph on 42 vertices with 80 edges can be formed by adding two new vertices numbered 40 and 41 and twenty new edges to the broadcast graph on 40 vertices shown in Fig. 20. In particular, connect both new vertices to vertices 2 and 6 in each of the 8-cycles of the 40 vertex graph.

The broadcast schemes for these two graphs use the same vertex numbering scheme as the one described in Theorem 19 for the 40 vertex graph. □

Theorem 24. $B(49) \leq 99$, $B(51) \leq 103$, $B(53) \leq 107$, *and* $B(55) \leq 112$.

Proof. A broadcast graph on 49 vertices with 99 edges can be formed by adding a new vertex (vertex 48) and three new edges to the broadcast graph on 48 vertices of Theorem 17. In particular, we can connect vertex 48 to vertices 0, 30 and 38 of the 48 vertex graph. A broadcast scheme for the new vertex as originator is given in the appendix.

To complete the proof for the 49 vertex graph, we show how to amend the scheme used for the 48 vertex graph to include the new vertex. In the 48 vertex broadcast scheme, the first call is between vertex 0 and vertex 37. In fact, the scheme can clearly be used for any originator in the 48 vertex graph by renumbering the vertices in the scheme. Without loss of generality, assume that the originator in the 49 vertex graph is even (and is not vertex 48).

Vertex 0 is idle at time 6 in the 48 vertex broadcast scheme and could call the new vertex 48 at that time since 0 is adjacent to 48 by our construction. Thus, if vertex 0 is the originator, the previously given scheme could easily be amended to inform all 49 vertices in 6 time units. Since we have also connected vertices 30 and 38 to the new vertex, this same amended scheme could be used if the originator was vertex 30 (or 38).

Vertex 2 is also idle at time 6 in the 48 vertex broadcast scheme. Note that if the originator is vertex 46 in the 49 vertex graph, vertex 2 of the scheme corresponds to vertex 0 of the 49 vertex graph and the scheme can easily be amended to inform vertex 48. Similarly, since vertices 30 and 38 are connected to the new vertex, this same amended scheme could be used if the originator is 28 or 36. Other vertices which are idle at time 6 in the 48 vertex broadcast scheme (6, 18 and 30) allow us to complete the 49 vertex scheme for originators 8, 12, 18, 20, 24, 32 or 42 by the same argument.

To this point, we have shown how to amend the original 48 vertex broadcast scheme to include the new vertex if the originator is 0, 8, 12, 18, 20, 24, 28, 30, 32, 36, 38, 42 or 46 (or the corresponding odd vertices). To complete the proof, we construct several alternate broadcast schemes for the 48 vertex graph which leave other vertices idle at time 6. One of these schemes is obtained by changing only one call in the original scheme to leave vertex 4 idle at time 6 rather than vertex 2. This scheme is amendable to include the new vertex if the originator is 26, 34 or 44. Other alternate broadcast schemes leave vertices 4, 8, 16, 26, 28, 32 and 36 idle at time 6. The list of potentially idle vertices (from all of the schemes) is now 0, 2, 4, 6, 8, 16, 18, 26, 28, 30, 32 and 36, and this provides at least one scheme for any originator to reach vertex 48 at time 6. All of the broadcast schemes for the 48 vertex graph are included in the appendix.

A broadcast graph on 51 vertices with 103 edges can be formed by adding a new vertex and three new edges to the broadcast graph on 50 vertices of Theorem 17. In particular, we can connect a new vertex numbered 50 to vertices 0, 32 and 40 of the 50 vertex graph. The proof is similar to that for the 49 vertex graph described

above and the corresponding broadcast schemes are included in the appendix.

A broadcast graph on 53 vertices with 107 edges can be formed by adding a new vertex and three new edges to the broadcast graph on 52 vertices of Theorem 17. In particular, we can connect a new vertex numbered 52 to vertices 0, 34 and 42 of the 52 vertex graph. The proof is similar to that for the 49 vertex graph described above and the corresponding broadcast schemes are included in the appendix.

A broadcast graph on 55 vertices with 112 edges can be formed by adding a new vertex and four new edges to the broadcast graph on 54 vertices of Theorem 17. In particular, we can connect a new vertex numbered 54 to vertices 0, 32, 36 and 44 of the 54 vertex graph. The proof is similar to that for the 49 vertex graph described above and the corresponding broadcast schemes are included in the appendix. □

Theorem 25. $B(57) \leq 126$ *and* $B(58) \leq 140$.

Proof. A broadcast graph on 57 vertices with 126 edges can be formed by adding a new vertex and fourteen new edges to the broadcast graph on 56 vertices of Fig. 12(a). In particular, we can connect the new vertex (vertex 56) to all vertices $i \equiv 0 \bmod 4$.

A broadcast graph on 58 vertices with 140 edges can be formed by adding two new vertices, numbered 56 and 57, and twenty-eight new edges to the broadcast graph on 56 vertices of Fig. 12(a). In particular, we can connect vertex 56 to all vertices $i \equiv 0 \bmod 4$ and vertex 57 to all vertices $j \equiv 1 \bmod 4$. □

The following sparse broadcast graphs are obtained by the technique of Section 4.4.

Theorem 26. $B(35) \leq 54$, $B(37) \leq 57$, *and* $B(39) \leq 60$.

Proof. A broadcast graph on 35 vertices with 54 edges can be obtained by using the construction of Lemma 1 to replace any vertex in the broadcast graph on 36 vertices (see Theorem 16) with a clique among its neighbours. Broadcast schemes for the new graph can be obtained from the schemes for the broadcast graph on 36 vertices using the method in Lemma 1. Thus, $B(35) \leq 54$.

$B(37) \leq 57$ follows from $B(38) \leq 57$ (Theorem 18) using the construction of Lemma 1 to replace any of the vertices in the broadcast graph on 38 vertices.

We see that $B(39) \leq 60$ by using the construction of Lemma 1 to delete a vertex from the broadcast graph on 40 vertices of Fig. 20. □

The following sparse broadcast graph is constructed using a variation of the vertex deletion technique of Section 4.4. In this particular instance, a vertex is removed and some (but not all) of its former neighbours are joined by new edges.

Theorem 27. $B(29) \leq 58$.

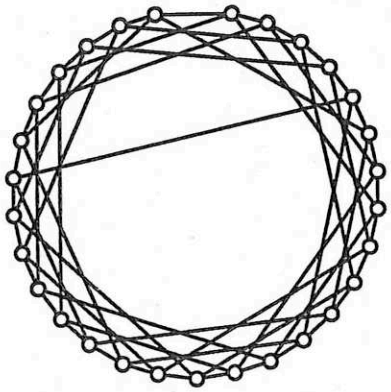


Fig. 22. Broadcast graph on 29 vertices.

Proof. Figure 22 is obtained from the graph of Fig. 4(a) by deleting a vertex v and adding two new edges: one between v's former neighbours on the cycle and the other joining v's other former neighbours. The broadcast schemes for this graph are obtained by modifying four schemes for the graph of Fig. 4(a). Figure 4(b) shows a scheme in which the originator first broadcasts to its "7-neighbour". The three additional schemes in the appendix show that any of the other three neighbours of the originator could be the first informed. The schemes in the appendix assume that vertices are numbered clockwise around the cycle C_{30}. A broadcast scheme for each possible originator in the graph of Fig. 22 is easily obtained by modifying one of these four schemes. □

5. Improvements to previous constructions

The method of Farley [4] and of Chau and Liestman [3] for constructing sparse broadcast graphs use minimum broadcast graphs (or broadcast graphs) for small n to build sparse broadcast graphs for larger values of n. Thus, the discovery of additional values of $B(n)$, or the improvement of the upper bound on $B(n)$ for small values of n, is likely to lead to improvements in the upper bounds on $B(n)$ for larger values of n. The method of Grigni and Peleg [6] is not affected by these improvements.

The method of Chau and Liestman [3] requires that the vertices of the small graphs be partitioned in a particular manner. Given a graph $G=(V,E)$, they define a partition of V into V_a and V_b to be an *even adjacency split* of V if $||V_a|-|V_b||\leq 1$ and each vertex in each of the two subsets is adjacent to at least one vertex in the other subset. If a broadcast graph has an even adjacency split, it can be used in the constructions of Chau and Liestman to create broadcast graphs for larger values of n. Chau and Liestman showed [3] that at least one mbg for each $n\leq 17$ has an even adjacency split and that any broadcast graph constructed by the methods of Farley or of Chau and Liestman (from smaller graph with this property) also has an even

adjacency split. It is a simple matter to verify that each of the graphs presented above can be thus partitioned.

We have calculated, for n in the range $18 \leq n \leq 1023$, the number of edges of the broadcast graphs constructed by the three schemes (Farley, Chau and Liestman, and Peleg) using only the mbg's and sparse broadcast graphs known prior to this paper. When the calculations are repeated using the new minimum broadcast graphs and sparse broadcast graphs of this paper, 60% of the resulting upper bounds on $B(n)$ (for those values of n for which $B(n)$ is not known exactly) are improved. In fact, in the range $18 \leq n \leq 511$, 85% of these bounds are improved. The current best known upper bounds on $B(n)$ for $1 \leq n \leq 64$ are shown in Table 2.

Table 2. Upper bound on $B(n)$ (* indicates optimality).

n	new	old	n	new	old	n	new	old	n	new	old
1		0*	17		22*	33	51	56	49	99	107
2		1*	18	23*	27	34	51	58	50	100	111
3		2*	19	25*	30	35	54	61	51	103	116
4		4*	20	28	32	36	54	63	52	104	121
5		5*	21	30	35	37	57	66	53	107	126
6		6*	22	33	37	38	57	69	54	108	131
7		8*	23	35	39	39	60	72	55	112	135
8		12*	24	36	42	40	60	75	56	112	140
9		10*	25	40	45	41	70	78	57	126	143
10		12*	26	48	49	42	80	81	58	140	147
11		13*	27	51	52	43	84	86	59	147	151
12		15*	28	52	56	44	88	90	60	150	156
13		18*	29	58	59	45	90	94	61	155	164
14		21*	30	60*	63	46	92	97	62	161	173
15		24*	31	65*	71	47	94	100	63	176	182
16		32*	32		80*	48	96	103	64		192*

6. Open problems

It is natural to guess that $B(n) \leq B(n+1)$ for $n \neq 2^k$. Our confidence in this has increased as a result of our recent work in this area.

Let $B_t(n)$ denote the minimum number of edges in a graph on n vertices such that every vertex can broadcast in t time units. We believe that $B_t(n) \leq B_t(n+1)$ for any fixed t.

We believe that $B(2^{k+1}-2) = \frac{1}{2}k(2^{k+1}-2)$ for $k \geq 3$. We know that this is true for $k=3$ and for $k=4$. If we can construct a k-regular broadcast graph on $2^{k+1}-2$ vertices, it will be a minimum broadcast graph since at most $2^{k+1}-3$ vertices can be informed in $k+1$ time units from a vertex of degree $k-1$.

Acknowledgement

We would like to thank Brian Alspach and Xiao Wang for their contributions.

Note added in proof

$B(2^{k+1}-2)=\frac{1}{2}k(2^{k+1}-2)$ for $k\geq 3$ has been proved by Dineen, Fellows and Faber [11].

Appendix

Table 3 indicates the locations of the broadcast schemes corresponding to the current best bounds on $B(n)$ for $1\leq n\leq 64$. The column headed "reference" gives a

Table 3.

n	Reference	Schemes
1–16	[5]	
17	[8]	
18	Theorem 2	Fig. 2
19	Theorem 5	Fig. 7
20	Theorem 10	Fig. 13
21	Theorem 21	Scheme 1
22	Theorem 6	Fig. 9
23	Theorem 22	Scheme 2
24	Theorems 8, 11	Figs. 11, 14
25	Theorem 12	Fig. 15
26	Theorem 14	Fig. 18
27	Theorem 20	Schemes 3, 4, 5, 6, 7, 8, 9
28	Theorem 13	Fig. 17
29	Theorem 27	Fig. 4, Schemes 10, 11, 12
30	Theorem 3	Fig. 3(a), Schemes 13, 14, 15
	Theorem 3	Fig. 3(b), Scheme 16
	Theorem 3	Fig. 4
31	Theorem 4	Fig. 6
32	[5]	
33	Theorem 15	Lemma 1
34	Theorem 7	Fig. 10
35	Theorem 26	Lemma 1
36	Theorem 16	Scheme 17
37	Theorem 26	Lemma 1
38	Theorem 18	Schemes 18, 19
39	Theorem 26	Lemma 1
40	Theorem 19	Scheme 20
41	Theorem 23	Scheme 21

Table 4 (*cont.*)

42	Theorem 23	Scheme 22
43	[4]	
44	Theorem 17	Scheme 23
45	[4]	
46	Theorem 17	Scheme 24
47	[4]	
48	Theorem 17	Scheme 25
49	Theorem 24	Schemes 25–32
50	Theorem 17	Scheme 33
51	Theorem 24	Schemes 33–39
52	Theorem 17	Scheme 40
53	Theorem 24	Schemes 40–46
54	Theorem 17	Scheme 47
55	Theorem 24	Schemes 47–56
56	Theorem 9	Fig. 12
57	Theorem 25	Scheme 57
58	Theorem 25	Scheme 58
59–63	[4]	
64	[5]	

reference to the proof of the bound on $B(n)$. For results proved in this paper, the column headed "Schemes" indicates the location of the broadcast schemes. Some of the schemes are described in the main body of the paper. The remaining schemes follow in Table 4. Each scheme below has three rows: $p(i)$ is the parent of vertex i in the broadcast scheme, and $t(i)$ is the time at which $p(i)$ informs vertex i.

Table 4. Schemes.

1	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20						
	$p(i)$	1	20	1	2	0	0	5	6	12	5	11	20	11	12	10	0	15	13	2	18	*						
	$t(i)$	2	1	3	5	5	3	4	5	5	5	4	2	3	4	5	4	5	5	4	5	*						
2	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22				
	$p(i)$	22	0	1	2	5	0	11	2	7	10	11	22	11	12	15	10	21	18	1	20	21	0	*				
	$t(i)$	1	2	3	5	5	4	5	4	5	5	3	2	4	5	5	4	5	5	4	5	4	3	*				
3	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
	$p(i)$	*	*	0	5	5	0	11	20	3	10	1	10	17	14	1	16	2	16	19	1	19	23	23	10	2	24	21
	$t(i)$	*	*	2	4	5	3	5	5	5	5	2	4	5	5	4	5	3	4	5	3	4	4	5	3	4	5	5
4	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
	$p(i)$	2	2	*	5	5	0	11	20	3	10	1	10	17	14	1	16	2	16	19	1	19	23	23	10	2	24	21
	$t(i)$	2	1	*	4	5	3	5	5	5	5	2	4	5	5	4	5	3	4	5	3	4	4	5	3	4	5	5
5	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
	$p(i)$	5	0	0	*	3	3	8	8	3	14	23	25	13	26	13	17	2	4	19	5	7	22	8	22	26	26	8
	$t(i)$	3	5	4	*	3	2	5	4	1	5	5	5	5	3	4	5	5	4	5	4	5	5	3	4	5	4	2

Table 4 (*cont.*)

6	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26		
	$p(i)$	5	0	16	4	*	4	8	20	3	14	23	25	17	12	12	17	17	4	19	5	19	26	21	21	26	26	17		
	$t(i)$	4	5	5	3	*	2	5	5	4	5	5	5	3	5	4	5	4	1	5	3	4	3	5	4	5	4	2		
7	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26		
	$p(i)$	5	19	0	5	5	*	11	20	3	10	19	10	17	12	1	20	15	4	19	5	19	23	8	10	25	11	17		
	$t(i)$	4	4	5	3	2	*	5	5	4	5	2	3	4	5	5	4	5	3	5	1	3	5	5	4	5	4	5		
8	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26		
	$p(i)$	5	10	16	8	3	3	*	6	6	11	11	6	17	12	9	16	17	8	23	20	7	22	8	10	2	11	17		
	$t(i)$	5	5	4	3	5	4	*	3	1	4	3	2	4	5	5	5	3	2	5	5	4	5	4	4	5	5	5		
9	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26		
	$p(i)$	5	19	1	8	3	3	7	*	*	14	19	6	17	12	12	20	17	8	20	20	7	22	8	18	26	11	17		
	$t(i)$	5	4	5	3	5	4	3	*	*	5	5	4	3	5	4	5	5	2	4	3	2	5	4	5	5	5	4		
10	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	4	5	0	5	6	9	4	15	12	5	12	13	22	17	12	23	18	13	22	23	0	23	20	1	26	29	0
	$t(i)$	*	3	5	5	3	1	4	5	5	4	5	5	2	3	5	4	5	4	4	5	4	5	3	2	5	5	4	5	5	4
11	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	3	0	29	2	1	8	3	10	5	8	7	22	9	24	23	26	21	28	23	0	29	2	1	28	29	0
	$t(i)$	*	1	2	3	5	4	5	4	3	4	4	5	5	5	5	5	5	5	5	5	5	4	4	3	4	5	4	5	3	2
12	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	28	3	0	29	6	1	8	11	6	5	8	13	16	21	24	23	18	21	28	23	0	29	26	1	28	29	0
	$t(i)$	*	2	5	4	5	4	3	5	3	5	5	4	5	4	5	5	4	5	4	5	5	3	5	3	4	5	4	5	2	1
13	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	5	0	6	*	6	7	5	11	5	7	9	6	19	24	12	14	11	24	20	27	25	28	1	4	28	0	28
	$t(i)$	*	3	5	5	4	2	*	3	5	4	5	3	4	5	4	5	5	5	5	4	4	5	5	5	3	4	5	4	2	5
14	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	1	2	*	2	26	6	7	2	7	8	11	12	7	12	6	13	10	12	14	11	26	20	27	25	28	1	27	2	27	3
	$t(i)$	5	3	*	4	5	5	3	1	4	5	4	3	2	4	4	5	5	5	5	5	4	5	5	5	5	4	3	2	4	5
15	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	4	2	3	4	*	6	7	8	*	8	8	10	7	9	10	16	10	16	17	11	26	20	21	25	20	26	4	26	0	3
	$t(i)$	4	5	4	3	*	5	4	3	*	4	2	4	5	5	5	5	3	4	5	5	3	4	5	5	5	4	2	5	5	5
16	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	0	0	1	25	0	6	0	6	7	5	6	10	11	5	10	13	10	12	14	11	26	29	27	25	23	26	0	26	4	1
	$t(i)$	0	3	5	5	4	3	1	4	5	5	2	3	4	4	4	5	5	5	5	5	5	5	5	4	5	3	2	4	5	4
17	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	*	1	2	3	4	7	17	34	10	11	1	11	14	4	33	17	18	0	18	19	3	21	24	25	35	27	28	2	28
	$t(i)$	*	*	2	3	4	6	6	5	5	5	4	3	5	6	5	6	6	4	3	5	6	5	6	6	5	4	6	5	4	6
	i	30	31	32	33	34	35																								
	$p(i)$	12	32	33	34	35	0																								
	$t(i)$	6	6	5	4	3	2																								
18	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	4	20	4	5	8	21	8	11	36	22	27	15	37	17	18	0	0	19	20	21	16	1	24	9	28	17	2
	$t(i)$	*	3	4	6	4	5	6	6	4	5	6	5	6	6	6	5	5	3	2	1	2	3	5	6	5	6	6	5	4	5

Table 4 (*cont.*)

	i	30	31	32	33	34	35	36	37																						
	$p(i)$	29	32	33	18	33	36	37	19																						
	$t(i)$	6	6	5	4	6	6	4	3																						
19	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	19	0	1	34	20	4	7	35	21	8	11	36	11	14	15	37	15	18	0	*	19	20	21	16	1	24	25	13	17	2
	$t(i)$	2	3	5	6	5	6	5	4	5	6	5	4	6	5	4	3	5	5	4	*	3	4	6	6	4	5	6	6	6	6
	i	30	31	32	33	34	35	36	37																						
	$p(i)$	6	10	14	18	35	36	37	19																						
	$t(i)$	6	6	6	6	5	3	2	1																						
20	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	3	6	7	0	15	8	29	4	13	14	15	0	17	6	17	20	21	2	21	16	31	14	5	28	29	30
	$t(i)$	*	1	2	3	5	5	3	2	5	6	6	6	6	5	4	3	5	4	6	6	5	4	6	6	6	6	6	6	5	4
	i	30	31	32	33	34	35	36	37	38	39																				
	$p(i)$	1	30	7	32	33	36	3	36	37	32																				
	$t(i)$	3	5	4	5	6	6	4	5	6	6																				
21	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	1	2	40	2	3	6	40	6	15	10	40	4	13	14	40	0	17	6	40	20	21	2	21	8	25	26	5	26	29	30
	$t(i)$	3	2	1	3	4	3	2	4	5	4	3	5	6	5	4	4	6	5	5	6	5	4	6	6	6	5	4	6	6	5
	i	30	31	32	33	34	35	36	37	38	39	40																			
	$p(i)$	1	30	7	32	40	36	3	18	9	38	*																			
	$t(i)$	4	6	5	6	6	6	5	6	5	6	*																			
22	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	1	2	41	2	3	6	41	6	15	10	41	4	13	14	41	0	17	6	41	20	21	2	21	8	25	26	5	26	29	30
	$t(i)$	3	2	1	3	4	3	2	4	5	4	3	5	6	5	4	4	6	5	5	6	5	4	6	6	6	5	4	6	6	5
	i	30	31	32	33	34	35	36	37	38	39	40	41																		
	$p(i)$	1	30	7	32	40	36	3	18	9	38	2	*																		
	$t(i)$	4	6	5	6	6	6	5	6	5	6	5	*																		
23	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	4	5	6	7	0	7	10	43	10	1	6	13	26	15	28	7	18	27	32	23	34	23	26	33	26	27	28
	$t(i)$	*	4	5	6	5	4	3	2	5	6	4	5	6	5	6	5	6	6	4	5	6	6	6	4	5	4	2	3	4	5
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43																
	$p(i)$	19	32	33	0	33	34	43	36	27	38	29	8	43	0																
	$t(i)$	6	5	4	1	3	5	5	6	5	6	6	6	6	3																
24	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	4	5	6	7	0	7	10	45	10	1	6	13	26	17	28	7	30	27	20	29	34	25	36	25	28	35	28
	$t(i)$	*	4	5	6	5	4	3	2	5	6	4	5	6	5	6	6	6	5	4	6	5	6	6	6	6	4	5	4	2	3
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45														
	$p(i)$	29	30	33	34	35	0	35	36	45	38	29	40	31	8	45	0														
	$t(i)$	4	5	6	5	4	1	3	5	5	6	5	6	6	6	6	3														
25	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	4	5	6	7	0	7	10	47	10	1	6	13	8	5	28	7	30	19	32	29	22	31	36	27	38	27	30
	$t(i)$	*	4	5	6	5	4	3	2	5	6	4	5	6	5	6	6	6	6	4	5	6	6	5	6	6	6	6	4	5	4
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47												
	$p(i)$	37	30	31	32	35	36	37	0	37	38	47	40	31	42	33	38	47	0												
	$t(i)$	2	3	4	5	6	5	4	1	3	6	5	6	5	6	6	5	6	3												

Table 4 (*cont.*)

26	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	5	6	7	0	7	10	47	10	1	6	13	8	5	28	7	30	19	32	29	22	31	36	27	38	27	30
	$t(i)$	*	4	5	6	5	4	3	2	5	6	4	5	6	5	6	6	6	6	4	5	6	6	5	6	6	6	6	4	5	4
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47												
	$p(i)$	37	30	31	32	35	36	37	0	37	38	47	40	31	42	33	38	47	0												
	$t(i)$	2	3	4	5	6	5	4	1	3	6	5	6	5	6	6	5	6	3												
27	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	5	6	7	0	7	10	47	10	1	6	13	8	5	28	7	30	19	32	29	22	31	36	27	38	27	30
	$t(i)$	*	4	5	6	6	4	3	2	5	6	4	5	6	5	6	6	5	6	4	5	6	6	5	6	6	6	6	4	5	4
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47												
	$p(i)$	37	30	31	32	35	36	37	0	37	38	47	40	31	42	33	38	47	0												
	$t(i)$	2	3	4	5	6	5	4	1	3	6	5	6	5	6	6	5	6	3												
28	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	4	5	6	7	0	7	10	47	10	1	6	13	8	5	28	7	30	19	32	29	22	31	36	27	38	27	30
	$t(i)$	*	4	5	6	5	4	3	2	4	6	4	5	6	5	6	5	6	6	5	5	6	6	5	6	6	6	6	4	5	4
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47												
	$p(i)$	37	30	31	32	35	36	37	0	37	38	47	40	31	42	33	38	47	0												
	$t(i)$	2	3	4	5	6	5	4	1	3	6	5	6	5	6	6	5	6	3												
29	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	5	6	7	0	7	10	47	10	1	6	13	8	5	28	7	30	19	22	29	16	31	36	27	38	27	30
	$t(i)$	*	4	5	6	6	4	3	2	5	6	4	5	6	5	6	6	5	6	4	5	6	6	5	6	6	6	6	4	5	4
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47												
	$p(i)$	37	30	31	32	35	36	37	0	37	38	47	40	31	42	33	38	47	0												
	$t(i)$	2	3	4	5	6	5	4	1	3	6	5	6	5	6	6	5	6	3												
30	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	4	5	6	7	0	7	10	47	10	1	6	13	8	5	18	7	30	19	32	29	22	31	18	27	38	27	30
	$t(i)$	*	4	5	6	5	4	3	2	5	6	4	5	6	5	6	6	6	6	4	5	6	6	5	6	6	5	6	4	5	4
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47												
	$p(i)$	37	30	31	32	35	36	37	0	37	38	47	40	31	42	33	38	47	0												
	$t(i)$	2	3	4	5	6	5	4	1	3	6	5	6	5	6	6	5	6	3												
31	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	4	5	6	7	0	7	10	47	10	1	6	13	8	5	18	7	30	19	32	29	22	31	18	27	38	27	30
	$t(i)$	*	4	5	6	5	4	3	2	5	6	4	5	6	5	6	6	6	6	4	5	6	6	5	6	6	5	5	4	6	4
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47												
	$p(i)$	37	30	31	32	35	36	37	0	37	38	47	40	31	42	33	38	47	0												
	$t(i)$	2	3	4	5	6	5	4	1	3	6	5	6	5	6	6	5	6	3												
32	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	48	0	1	4	5	6	7	0	7	10	47	10	1	6	13	8	5	18	7	30	19	32	29	22	31	18	27	38	27	30
	$t(i)$	1	4	5	6	5	4	3	2	5	6	4	5	6	5	6	6	6	5	4	5	6	6	5	6	6	6	5	4	6	4
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48											
	$p(i)$	48	30	31	32	33	26	37	0	48	38	47	40	31	42	45	38	47	0	*											
	$t(i)$	2	3	4	5	6	6	6	5	3	6	5	6	5	6	6	5	6	3	*											
33	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	5	6	7	0	7	8	49	10	1	6	21	26	5	10	7	18	21	32	29	34	13	18	33	28	29	40
	$t(i)$	*	4	5	6	6	4	3	2	5	6	4	6	6	5	5	6	5	5	4	5	6	4	6	6	6	6	5	6	5	4

Table 4 (*cont.*)

	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
	$p(i)$	31	32	39	32	33	34	35	38	39	0	39	40	49	42	33	6	47	40	49	0
	$t(i)$	6	5	2	3	4	5	6	5	4	1	3	6	5	6	6	6	6	5	6	3
34	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	5	6	7	0	7	8	49	10	1	6	21	26	5	10	7	18	21	32	29	34	13	18	33	28	29	40
	$t(i)$	*	4	5	6	5	4	3	2	5	6	4	6	6	5	6	6	6	5	4	5	5	4	6	6	6	6	5	6	5	4
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49										
	$p(i)$	31	32	39	32	33	34	35	38	39	0	39	40	49	42	33	6	47	40	49	0										
	$t(i)$	6	5	2	3	4	5	6	5	4	1	3	6	5	6	6	6	6	5	6	3										
35	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	5	6	7	0	7	8	49	10	1	6	21	16	5	10	7	18	21	32	29	34	13	18	33	28	29	40
	$t(i)$	*	4	5	6	6	4	3	2	5	6	4	6	6	5	5	6	5	5	4	5	6	4	6	6	6	6	5	6	5	4
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49										
	$p(i)$	31	32	39	32	33	34	35	38	39	0	39	40	49	42	33	6	47	40	49	0										
	$t(i)$	6	5	2	3	4	5	6	5	4	1	3	6	5	6	6	6	6	5	6	3										
36	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	5	6	7	0	7	8	49	10	1	6	21	16	5	10	7	18	21	32	29	34	13	18	33	26	29	40
	$t(i)$	*	4	5	6	6	4	3	2	5	6	4	6	6	5	5	6	5	5	4	5	6	4	6	6	6	6	5	6	5	4
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49										
	$p(i)$	31	32	39	32	33	34	35	38	39	0	39	40	49	42	33	6	47	40	49	0										
	$t(i)$	6	5	2	3	4	5	6	5	4	1	3	6	5	6	6	6	6	5	6	3										
37	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	5	6	7	0	7	8	49	10	1	6	21	16	5	10	7	18	21	32	29	34	13	18	33	26	29	40
	$t(i)$	*	4	5	6	6	4	3	2	5	6	4	6	6	5	5	6	5	5	4	5	6	4	5	6	6	6	5	6	6	4
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49										
	$p(i)$	31	32	39	32	33	34	35	38	39	0	39	40	49	42	33	6	47	40	49	0										
	$t(i)$	6	5	2	3	4	5	6	5	4	1	3	6	5	6	6	6	6	5	6	3										
38	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	4	5	6	7	0	7	20	49	10	1	6	21	26	5	10	7	18	21	32	29	34	13	18	33	28	29	40
	$t(i)$	*	4	5	6	5	4	3	2	5	6	4	6	6	5	6	6	6	5	4	5	5	4	6	6	6	6	5	6	5	4
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49										
	$p(i)$	31	32	39	32	33	34	35	38	39	0	39	40	49	42	33	6	47	40	49	0										
	$t(i)$	6	5	2	3	4	5	6	5	4	1	3	6	5	6	6	6	6	5	6	3										
39	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	50	0	1	2	5	6	7	0	7	8	49	10	1	6	21	26	5	10	7	18	19	32	29	34	13	18	33	28	29	40
	$t(i)$	1	4	5	6	6	4	3	2	5	6	4	6	6	5	6	6	5	5	4	5	6	5	6	6	6	6	5	6	5	4
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50									
	$p(i)$	31	32	50	32	33	34	35	30	39	0	50	40	49	42	33	6	47	40	49	0	*									
	$t(i)$	5	4	2	3	4	5	6	6	6	5	3	6	5	6	6	6	6	5	6	3	*									
40	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	5	6	7	0	7	10	51	10	1	6	13	8	5	18	7	18	19	32	23	34	23	36	33	28	35	40
	$t(i)$	*	4	5	6	5	4	3	2	5	5	4	6	6	5	6	6	6	6	4	5	6	6	6	4	5	6	6	6	5	6
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51								
	$p(i)$	31	42	31	34	41	34	35	36	39	40	41	0	41	42	51	44	35	6	49	42	51	0								
	$t(i)$	6	4	5	5	2	3	4	5	6	5	4	1	3	6	5	6	6	6	6	5	6	3								

Table 4 (*cont.*)

41	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	4	5	6	7	0	7	2	51	10	1	6	13	8	5	18	7	18	19	22	23	34	23	36	33	28	35	40
	$t(i)$	*	4	5	6	5	4	3	2	5	6	5	6	6	5	6	6	6	6	4	5	6	6	5	4	6	6	6	6	5	5
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51								
	$p(i)$	37	42	39	34	41	34	35	36	39	40	41	0	41	44	51	44	35	6	49	42	51	0								
	$t(i)$	6	6	6	5	2	3	4	5	5	4	3	1	4	5	4	6	6	6	6	5	6	3								
42	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	5	6	7	0	7	10	51	10	1	6	13	8	5	18	7	18	19	22	23	34	23	36	33	28	35	40
	$t(i)$	*	4	5	6	5	4	3	2	5	5	4	6	6	5	6	6	6	6	4	5	6	6	5	4	6	6	6	6	5	6
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51								
	$p(i)$	37	42	31	34	41	34	35	36	39	40	41	0	41	42	51	44	35	6	49	42	51	0								
	$t(i)$	6	5	6	5	2	3	4	5	6	5	4	1	3	6	5	6	6	6	5	4	6	3								
43	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	5	6	7	0	7	10	51	22	1	6	13	8	5	18	7	18	19	32	23	34	23	36	33	28	35	40
	$t(i)$	*	4	5	6	5	4	3	2	5	5	4	6	6	5	6	6	6	6	4	5	6	6	5	4	6	6	6	6	5	6
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51								
	$p(i)$	31	42	31	34	41	34	35	36	39	40	41	0	41	42	51	4	35	6	49	42	51	0								
	$t(i)$	6	4	5	5	2	3	4	5	6	5	4	1	3	6	6	6	6	6	6	5	5	3								
44	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	4	5	6	7	0	7	10	51	10	1	6	13	8	5	18	7	18	19	32	23	34	23	24	33	28	35	40
	$t(i)$	*	4	6	6	5	4	3	2	5	5	4	6	5	5	6	6	6	6	4	5	6	6	6	4	5	6	6	6	5	6
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51								
	$p(i)$	31	42	31	34	41	34	35	36	39	40	41	0	41	42	51	44	35	6	49	42	51	0								
	$t(i)$	6	4	5	5	2	3	4	5	6	5	4	1	3	6	5	6	6	6	6	5	6	3								
45	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	5	6	7	0	7	10	51	10	1	6	13	8	5	18	7	18	19	32	23	34	23	36	33	28	35	40
	$t(i)$	*	4	5	6	6	4	3	2	5	5	4	6	6	5	6	6	5	6	4	5	6	6	5	4	6	6	6	6	5	6
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51								
	$p(i)$	31	42	31	34	41	34	35	36	39	40	41	0	41	42	51	44	35	6	49	42	51	0								
	$t(i)$	6	4	5	5	2	3	4	5	6	5	4	1	3	6	5	6	6	6	6	5	6	3								
46	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	52	0	9	4	5	6	7	0	7	10	51	10	1	6	13	8	5	18	7	18	19	32	23	34	23	36	33	28	35	22
	$t(i)$	1	5	6	6	5	4	3	2	5	5	4	6	6	5	6	6	6	6	4	5	6	6	5	4	6	6	6	6	5	6
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52							
	$p(i)$	31	42	31	34	52	34	35	36	37	40	41	0	52	42	51	44	35	6	49	42	51	0	*							
	$t(i)$	6	4	5	5	3	3	4	5	6	6	5	4	3	6	5	6	6	6	6	5	6	3	*							
47	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	5	6	7	0	7	10	53	10	1	6	13	8	5	18	7	18	19	32	11	24	25	36	25	38	35	30
	$t(i)$	*	4	5	6	5	4	3	2	5	6	4	5	6	5	6	6	6	6	4	5	6	6	6	6	5	4	6	6	6	6
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53						
	$p(i)$	37	42	33	44	33	36	43	36	37	38	39	42	43	0	43	44	53	46	37	6	51	44	53	0						
	$t(i)$	5	6	5	4	6	5	2	3	4	5	6	5	4	1	3	6	5	6	6	6	6	5	6	3						
48	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	4	5	6	7	0	7	10	53	10	1	6	13	8	5	18	7	18	19	32	11	24	25	36	25	38	35	30
	$t(i)$	*	4	5	6	5	4	3	2	5	6	4	5	6	5	6	6	6	6	4	5	6	6	6	6	5	4	6	6	6	6

Table 4 (*cont.*)

	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53					
	$p(i)$	37	42	33	44	33	36	43	36	37	38	39	42	43	0	43	44	53	46	37	6	51	44	53	0					
	$t(i)$	5	6	5	4	6	5	2	3	4	5	6	5	4	1	3	6	5	6	6	6	6	5	6	3					
49	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	5	6	7	0	7	10	53	10	1	6	13	16	5	18	7	18	19	32	11	24	25	36	25	38	35	30
	$t(i)$	*	4	5	6	6	4	3	2	5	6	4	5	6	5	6	6	5	6	4	5	6	6	6	6	5	4	6	6	6	6
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53						
	$p(i)$	37	42	33	44	33	36	43	36	37	38	39	42	43	0	43	44	53	46	37	6	51	44	53	0						
	$t(i)$	5	6	5	4	6	5	2	3	4	5	6	5	4	1	3	6	5	6	6	6	6	5	6	3						
50	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	5	6	7	0	7	8	53	10	1	6	13	16	5	18	7	18	19	32	11	24	25	36	25	38	35	30
	$t(i)$	*	4	5	6	6	4	3	2	5	6	4	5	6	5	6	6	5	6	4	5	6	6	6	6	5	4	6	6	6	6
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53						
	$p(i)$	37	42	33	44	33	36	43	36	37	38	39	42	43	0	43	44	53	46	37	6	51	44	53	0						
	$t(i)$	5	6	5	4	6	5	2	3	4	5	6	5	4	1	3	6	5	6	6	6	6	5	6	3						
51	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	5	6	7	0	7	10	53	10	1	6	13	8	5	16	7	18	19	32	11	24	25	36	25	38	35	30
	$t(i)$	*	4	5	6	6	4	3	2	5	6	4	5	6	5	6	6	5	6	4	5	6	6	6	6	5	4	6	6	6	6
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53						
	$p(i)$	37	42	33	44	33	36	43	36	37	38	39	42	43	0	43	44	53	46	37	6	51	44	53	0						
	$t(i)$	5	6	5	4	6	5	2	3	4	5	6	5	4	1	3	6	5	6	6	6	6	5	6	3						
52	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	5	6	7	0	7	10	53	10	1	6	13	8	5	18	7	18	19	32	11	24	25	36	25	38	35	30
	$t(i)$	*	4	5	6	5	4	3	2	5	6	4	5	6	5	6	6	6	6	4	5	6	6	6	6	5	4	6	6	6	6
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53						
	$p(i)$	37	42	33	44	33	36	43	36	37	38	39	42	43	0	43	44	53	4	37	6	51	44	53	0						
	$t(i)$	5	6	5	4	6	5	2	3	4	5	6	5	4	1	3	6	6	6	6	6	6	5	5	3						
53	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	5	6	7	0	7	10	53	10	1	6	13	8	5	18	7	18	19	32	11	16	25	36	27	38	35	30
	$t(i)$	*	4	5	6	6	4	3	2	5	6	4	5	6	5	6	6	5	6	4	5	6	6	6	6	5	4	6	5	6	6
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53						
	$p(i)$	37	42	33	44	33	36	43	36	37	38	41	42	43	0	43	44	53	46	37	6	51	44	53	0						
	$t(i)$	5	6	5	4	6	5	2	3	4	6	6	5	4	1	3	6	5	6	6	6	6	5	6	3						
54	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	5	6	7	0	7	10	53	10	1	6	13	8	5	18	7	18	19	32	11	16	25	36	27	38	35	30
	$t(i)$	*	4	5	6	6	4	3	2	5	6	4	5	6	5	6	6	5	6	4	5	6	6	6	6	6	5	6	5	5	6
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53						
	$p(i)$	37	42	33	44	33	36	43	36	37	38	41	42	43	0	43	44	53	46	37	6	51	44	53	0						
	$t(i)$	5	6	5	4	6	4	2	3	4	6	6	5	4	1	3	6	5	6	6	6	6	5	6	3						
55	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	*	0	1	2	5	6	7	0	7	10	53	10	1	6	13	8	5	18	7	18	19	32	11	16	25	36	27	38	35	28
	$t(i)$	*	4	5	6	6	4	3	2	5	6	4	5	6	5	6	6	5	6	4	5	6	6	6	6	6	5	6	5	5	6
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53						
	$p(i)$	37	30	33	44	33	36	43	36	37	38	41	42	43	0	43	44	53	46	37	42	51	44	53	0						
	$t(i)$	5	6	5	4	6	4	2	3	4	6	6	5	4	1	3	6	5	6	6	6	6	5	6	3						

Table 4 (*cont.*)

56	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	54	0	1	2	5	6	7	0	7	10	53	10	1	6	13	8	5	18	7	18	19	32	11	24	25	36	25	38	35	30
	$t(i)$	1	4	5	6	5	4	3	2	5	6	4	5	6	5	6	6	6	6	4	5	6	6	6	6	5	4	6	6	6	6
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54					
	$p(i)$	37	32	54	44	33	36	54	36	37	38	39	34	43	0	54	44	53	46	37	6	51	44	53	0	*					
	$t(i)$	5	5	4	4	5	5	2	3	4	5	6	6	6	5	3	6	5	6	6	6	6	5	6	3	*					
57	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	56	0	1	4	5	16	7	0	15	8	55	10	56	6	15	16	56	16	7	18	27	14	21	16	23	36	15	38	35	40
	$t(i)$	1	5	6	6	5	4	4	3	5	6	5	6	6	6	4	3	2	6	5	6	6	5	6	5	6	5	6	5	6	5
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56			
	$p(i)$	29	32	39	44	35	36	56	38	45	38	56	34	43	44	45	0	45	40	56	48	39	6	51	46	55	0	*			
	$t(i)$	6	6	5	6	5	4	3	6	3	4	4	6	6	5	4	2	5	6	5	6	6	5	6	6	6	4	*			
58	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	$p(i)$	56	0	1	4	5	16	7	0	15	8	55	10	56	6	15	16	56	16	7	18	27	14	21	16	23	36	15	38	35	40
	$t(i)$	1	5	6	6	5	4	4	3	5	6	5	6	6	6	4	3	2	6	5	6	6	5	6	5	6	5	6	5	6	5
	i	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57		
	$p(i)$	29	32	39	44	35	36	56	38	45	38	56	34	43	44	45	0	45	40	56	48	39	6	51	46	55	0	*	45		
	$t(i)$	6	6	5	6	5	4	3	6	3	4	4	6	6	5	4	2	5	6	5	6	6	5	6	6	6	4	*	6		

Originators are indicated by *. (If two originators are indicated, then the first communication is between them.) For example, Scheme 1 is one of the broadcast schemes for the graph with 21 vertices described in Theorem 21. The originator is vertex 20 which starts the broadcast by informing vertex 1 at time 1.

References

[1] B. Alspach, Private communication, December, 1987.

[2] J.-C. Bermond, P. Hell, A.L. Liestman and J.G. Peters, Broadcasting in bounded degree graphs, SIAM J. Discrete Math., to appear.

[3] S.C. Chau and A.L. Liestman, Constructing minimal broadcast networks, J. Combin. Inform. System Sci. 10 (1985) 110–122.

[4] A. Farley, Minimal broadcast networks, Networks 9 (1979) 313–332.

[5] A. Farley, S. Hedetniemi, S. Mitchell and A. Proskurowski, Minimum broadcast graphs, Discrete Math. 25 (1979) 189–193.

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

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

[8] A.L. Liestman and J.G. Peters, Broadcast networks of bounded degree, SIAM J. Discrete Math. 1 (1988) 531–540.

[9] S. Mitchell and S. Hedetniemi, A census of minimum broadcast graphs, J. Combin. Inform. Systems Sci. 5 (1980) 141–151.

[10] X. Wang, Manuscript (1986).

[11] M.J. Dineen, M.R. Fellows and V. Faber, Algebraic constructions of efficient broadcast networks, Tech. Rept. DCS-153-IR, University of Victoria, Victoria, B.C. (1991).