ELSEVIER

Discrete Applied Mathematics 83 (1998) 207–228

DISCRETE APPLIED MATHEMATICS

Line broadcasting in cycles

Jave O. Kane[1], Joseph G. Peters[2,*]

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

Received 31 October 1995; received in revised form 17 December 1996; accepted 7 April 1997

Abstract

Broadcasting is the process of transmitting information from an originating node (processor) in a network to all other nodes in the network. A *local* broadcast scheme only allows a node to send information along single communication links to adjacent nodes, while a *line* broadcast scheme allows nodes to use paths of several communication links to call distant nodes. The minimum time possible for broadcasting in a network of n nodes when no node is involved in more than one communication at any given time is $\lceil \log n \rceil$ phases. Local broadcasting is not sufficient, in general, for broadcasting to be completed in minimum time; line broadcasting is always sufficient. An optimal line broadcast is a minimum-time broadcast that uses the smallest possible total number of communication links. In this paper, we give a complete characterization of optimal line broadcasting in cycles, and we develop efficient methods for constructing optimal line broadcast schemes. © 1998 Elsevier Science B.V. all rights reserved.

1. Introduction

In broadcasting, information known by one informed processor, the *originator*, is transmitted to all other nodes (processors) in a communication network. In *local* broadcasting, an informed node may use one of its communication links to *call* an adjacent node during any given time unit, or *phase*. In *line* broadcasting, an informed node may call any other node using the communication links of any simple path between the two nodes with the restriction that no link is used in more than one call in a given phase.

When no node is involved in more than one communication in any given phase, and each communication can be completed during one phase, the number of informed

* Corresponding author. E-mail: peters@cs.sfu.ca.

[1] Supported by NSERC of Canada Postgraduate Scholarships. Current address: Lawrence Livermore National Laboratory, Livermore, CA 94550, USA; E-mail:jave@llnl.gov.

[2] Supported by NSERC of Canada.

PII S0166-218X(97)00111-X

nodes can at most double during each phase, so at least $\lceil \log n \rceil$ phases[3] are needed to broadcast in a network of n nodes. It is not possible, in general, to inform all nodes in a network in minimum time (i.e., $\lceil \log n \rceil$ phases) using local broadcasting, but Farley [3] has shown that there is a *minimum-time line broadcast scheme* for any originator in any connected network. The question that we address in this paper is *how much* line broadcasting is needed to complete a minimum-time broadcast from an arbitrary originator in a given graph?

A *broadcast scheme* for a network is a specification of which calls are made during each phase and which communication paths are used to make the calls. A broadcast scheme for a network of n nodes requires $n-1$ calls. Furthermore, $n-1$ calls are sufficient because each node only needs to receive the information once. If minimum-time broadcasting using $n-1$ *local calls* is possible from any originator in a network, then the network is a *broadcast graph*. During the last 15 years, considerable effort has been devoted to the discovery of *minimum broadcast graphs* (broadcast networks with the fewest possible links) and to the construction of sparse broadcast graphs. (See [2] for a comprehensive study of this subject.) Unfortunately, situations in which a network can be designed to be optimal for a particular communication pattern such as broadcasting are rare. Usually, the topology of the network is determined by other factors and the task is to use the network as 'efficiently' as possible.

One approach to designing broadcast schemes in fixed networks is to use only local calls and then try to minimize time (e.g., the number of phases) or some other measure of cost. If the network uses *store-and-forward* routing, then this is the only possible approach since all communications are local. (See [5] for a recent survey of research in this area.) If the network supports some form of *circuit-switched* routing, then a second possible approach is to insist that one of the parameters, such as the number of phases, is optimized, and then try to minimize some other measure of cost. Usually, this other measure is total time to complete the broadcast taking into account other factors such as switching time at intermediate nodes and transmission rates of links. (A recent example of this approach is [10].)

In this paper, we will take the somewhat different approach of minimizing the total amount of 'equipment' used to complete broadcasting in the minimum number of phases using circuit-switched routing. In particular, we will minimize the total number of communication links used (i.e., the sum over all phases of the number of links used in each phase). A simple example for which this approach could be useful is the distribution of electronic news on the Internet. At one time, most of the network used telephone lines and most sites were only willing to devote one modem to the net-news. The cost of distributing news depended on the amount of data and on the distance that it was sent. The elapsed time of a phone call to send a particular piece of news is essentially independent of distance travelled, so it makes sense to talk about phases of a broadcast. Assuming that network news readers want their news as quickly as possible, the cost of providing news over a telephone network depends on the total

[3] All logarithms in this paper are base 2.

amount of equipment used. In other words, it depends on the long distance telephone charges and these are proportional to the total distance travelled. While the current technology of the Internet involving high-speed trunks and dedicated lines is much more sophisticated, the model still has validity.

There have not been many papers on the subject of line broadcasting. Farley [3] introduced the topic and gave a constructive proof that minimum-time line broadcasting is possible in any tree (and hence any connected network). Farley's construction gives an upper bound of $(n-1)\lceil \log n \rceil$ on the minimum *total length* (total number of links) needed to line broadcast in minimum time in any network on n nodes. In this paper, we determine the exact value of the minimum total length for minimum-time line broadcasting in any cycle on n nodes. For cycles of $n = 2^k$ nodes, we give the exact value explicitly; for other values of n, the exact values are given indirectly. For all values of n, we show that the exact value is only about $\frac{1}{3}$ of Farley's upper bound. Farley [3] also introduced a more restricted form of line broadcasting called *path broadcasting* in which calls in any given phase must use node-disjoint paths. In this paper, we show that optimal line broadcasting in cycles satisfies the stronger restriction of path broadcasting. Almstrom [1] studied a restricted type of line broadcasting on networks that consist of a single path of processors (i.e., a one-dimensional grid). Almstrom determined the number of nodes reachable in k phases when there is a constant upper bound on the length of line calls. Fujita and Farley [6] have extended our results from cycles to paths. In particular, they have derived bounds on the total length needed to line broadcast from any originator in a path of length n in terms of the total length required to line broadcast in a cycle of length n, and they have derived the exact total length for originators close to the ends of paths of length $n = 2^k$. A line broadcasting model in which all intermediate nodes involved in a line call also receive the message is studied in [4].

A problem closely related to ours, *embedding*, has received considerable attention. (See [9] for a survey.) Our problem is to find a *constrained* embedding of a *broadcast tree* (which describes the logical structure of a broadcast) into a graph representing the interconnections of a network. The vertices of the broadcast tree are mapped one-to-one to network nodes and edges of the broadcast tree are mapped to paths in the network. The reason our problem is a 'constrained' embedding is because the calls in any phase must use link-disjoint paths. An embedding of a broadcast tree into a cycle can also be specified by numbering the vertices of the tree with the integers in $[1, n]$. Consecutive integers correspond to consecutive nodes of the cycle. The total length of the embedding is the sum of the differences of the labels assigned to the endpoints of the edges of the tree. A minimum-sum numbering gives an optimal embedding. Iordanskiǐ [7] has investigated minimum-sum numberings of trees with fixed degree bounds using some concepts similar to the concepts of *layers* and *nestedness* that we use in this paper.

In this paper, we investigate line broadcasting in cycles using a model in which broadcasting must be completed in $\lceil \log n \rceil$ phases and the optimization measure (or cost) is the total number of links used during the broadcast. We determine the cost of minimum-time line broadcasting in cycles, give a complete characterization of optimal line broadcast schemes in cycles, and develop efficient methods for constructing optimal line broadcast schemes. The basis of our results is three independent properties of broadcast schemes – *nestedness*, *flatness*, and *fullness*. We prove that these three properties are both necessary and sufficient for optimality. We determine the exact cost of flat, nested, full broadcast schemes for cycles with 2^k nodes, $k \geqslant 1$ and then adapt our results to cycles of other lengths. This leads to an upper bound of approximately $n\lceil \log n \rceil/3$ for all $n \geqslant 2$. The proofs of necessity and sufficiency appear in Section 3 of this paper. The cost analysis and methods for constructing optimal schemes are in Section 4. In Section 2, we discuss several examples of optimal schemes to introduce terminology and to give informal and intuitive definitions of the main concepts used in Sections 3 and 4. Section 5 contains a brief discussion of open problems.

2. Properties and examples

In this paper we will model communication networks as graphs in which nodes represent processors and edges represent communication links. The networks studied in this paper are n-cycles – n processors connected into a simple cycle by n communication links. We will assume a *unit-cost single-port* model of communication (see [5]) in which no node is involved in more than one communication at any given time and each communication can be completed in one time unit or *phase*. With these assumptions, the minimum number of phases to complete broadcasting is $\lceil \log n \rceil$.

A *broadcast tree* describes the logical structure of a broadcast. A broadcast tree which describes a minimum-time broadcast is a *minimum broadcast tree* or MBT. In the unit-cost single-port model, an MBT with 2^k nodes is a binomial tree, and any MBT with n nodes, $2^{k-1} < n < 2^k$, is a subgraph of the binomial tree on 2^k nodes. The root of an MBT (and the binomial tree that contains it) is called the *originator*. Associated with each node of a broadcast tree is the phase during which it receives the message. The phase of the originator is 0 and the 'deepest' phase is $\lceil \log n \rceil$ if the tree is an MBT. If an MBT is a complete binomial tree, then the assignment of phases to nodes is unique. If the number of nodes is not a power of 2, then the MBT is not necessarily unique, and there also may be some flexibility in the assignment of phases to some of the nodes.

A *broadcast scheme* S is a pair (T,E), where T is a broadcast tree with n nodes, and E is an embedding of T into a network G with n processors. To simplify notation and avoid awkward phrasing, we will often use the term *scheme* and the notation S when one of the terms *broadcast tree* or *embedding* or symbols T or E would be more precise. For example, we will usually talk about properties of a scheme S even when they are actually properties of the embedding E or of the broadcast tree T. We will always use the more precise terminology when defining new concepts or when the context does not eliminate ambiguity. Since the embedding E is a one-to-one mapping of the nodes of T to the processors of G, we will often use the term *node* when referring to a processor. Each edge of T is mapped by E to a simple path in G. Since the correspondence between edges of T and communication links of G is not necessarily one-to-one, we will use *link* when referring to physical links in a network and *edge* when referring to broadcast trees.

Each path P of T is mapped by E to a connected path in G which we denote $S(P)$ (even though $E(P)$ would be more precise). Similarly, we use $S(T)$ to denote the embedding of T into G. $S(P)$ is not necessarily simple; it may fold on itself, and it may 'wrap around' the cycle and overlap itself. Associated with $S(P)$, there is a connected, undirected, path in the cycle G which contains all of the nodes on $S(P)$ and exactly all of the links on $S(P)$. We call this path the *segment* of $S(P)$ and denote it $\sigma(S(P))$. If $S(P)$ wraps around the cycle, then $\sigma(S(P))$ is exactly the cycle. We will see that $\sigma(S(P))$ is always a simple path in an optimal scheme. If T_u is a subtree of T rooted at node u, then the *subscheme* $S(T_u)$ is the scheme $S(T)$ restricted to T_u. Since S preserves the connectedness of T, we can extend the definition of *segment* to any subscheme $S(T_u)$ in the obvious way.

A line broadcast scheme S for n nodes always uses a total of at least $n-1$ links. If S maps an edge of a broadcast tree to a single link of the network, then the embedded edge is a *local call*. If the edge is mapped to a path P of $\lambda > 1$ links, then $S(P)$ is a *line call* which contributes *extra length* $\lambda - 1$ to the *total length* of a broadcast scheme. Thus, the total length of a scheme is always *total extra length* plus $n-1$. A *minimum-time broadcast scheme* is a scheme that has $\lceil \log n \rceil$ phases. An *optimal line broadcast scheme* for a particular originator is a minimum-time scheme rooted at the originator with minimum total extra length. Since cycles are vertex-transitive, we are free to specify any processor as the originator of the broadcast.

Fig. 1 shows several schemes on cycles. Parts (a), (b), (c), (d), and (f) of the figure show schemes for 4-, 8-, 16-, 32- and 64-cycles, respectively. Parts (e) and (g) show schemes for 22- and 55-cycles. Nodes are shown as black dots, and calls as arrows or short lines. Links of the cycle are not shown. In particular, for each cycle, the link connecting the leftmost and rightmost nodes is not shown. That link is not used by any call in any of our examples (although nothing in the definition of line broadcast schemes prohibits the use of that link). The number under a node is the phase during which the node is informed; the originator is informed at phase 0. An arrowhead on a call, if present, shows the direction of the call. The 4-cycle scheme in part (a) appears repeatedly in the other schemes as a subscheme, and when it does, the arrowheads are omitted to reduce clutter.

Each scheme is shown in two ways; the first shows all of the nodes on one line, and the second shows one path of the scheme laid out flat and the rest of the scheme hanging below that path. The phases and positions of nodes on the cycle are the same in the two representations. The total extra length of any call is exactly the number of nodes which are under the call. (This is easier to see in the first representation.)

The schemes in Fig. 1 are all minimum-time schemes. This can be verified by examining the phases at which nodes are informed. They are also all optimal, as we will

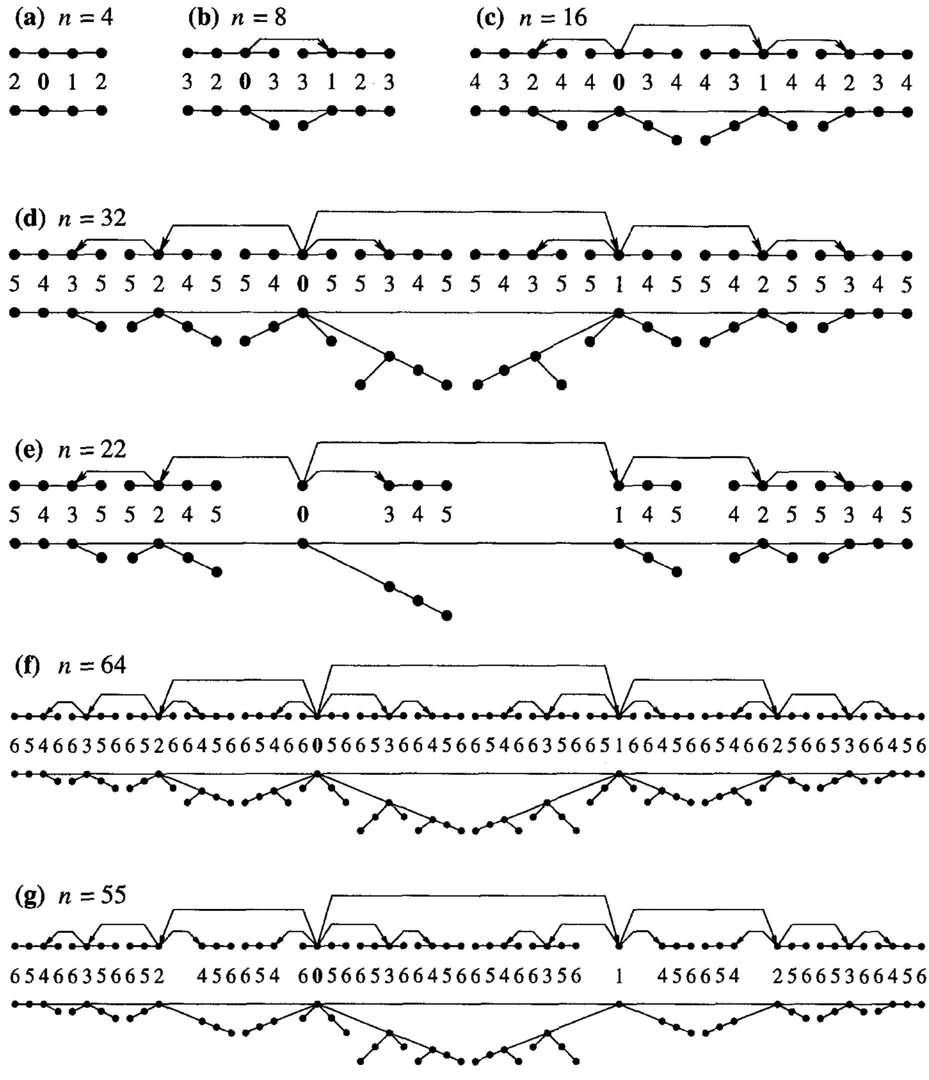

Fig. 1. Cycle schemes.

prove in later sections of this paper. The proof of optimality is based on a demonstration that three independent properties of broadcast schemes, which we call *nestedness*, *flatness*, and *fullness* are necessary and sufficient for optimality. Fullness is a property of a broadcast tree; nestedness and flatness are properties of an embedding of a broadcast tree.

From the schemes in Fig. 1, we note that calls are *nested*; later calls are shorter and stay under earlier calls, and calls never cross. Thus, for any pair of calls, either one of the calls is completely under the other, or the calls do not share any links. We will see that a scheme that is not nested can always be modified to reduce its total length.

In the second representation, a *top path* of calls is laid out *flat*. It can be seen that all but one link of the cycle is on the top path, and that the rest of the scheme is completely under the top path. We say that the top path is *layer* 0 of a scheme S. Further examination reveals that removal of the top path leaves a set of *subschemes* and that each of these subschemes has a flat top path. The collection of the top paths of these subschemes is *layer* 1 of S. Removal of layer 1 gives sub-subschemes, and so on. (Examine the subscheme structure of the originator in parts (f) and (g) for example.) We will prove that any optimal scheme can be decomposed into layers by repeatedly removing the flat top paths of subschemes. We give the name *flatness* to the property of a scheme that all of its layers are embedded flat.

In all of the examples in Fig. 1, the shallower layers of a scheme are as *full* as possible (i.e., there are as many calls as possible at layer 0, then layer 1 is filled, and so on). We will prove that, if the shallow layers of a scheme are not full, then there is another scheme with smaller total length in which shallow layers *are* full. We use the term *fullness* to refer to the property that all but the deepest layer contains as many calls as possible.

Further examination of the first representation in parts (a)–(f) of Fig. 1 reveals that an optimal scheme for a 2^k-cycle can be produced from a 2^{k-1}-cycle scheme by placing two mirror image 2^{k-1}-cycle schemes beside each other and joining their originators with a line call. A second recursive method for creating a 2^k-cycle scheme is to start with a scheme for a 2^{k-1} cycle, add two new nodes to the center of the top path, and let each of the two new nodes be the root of a subscheme which looks exactly like an optimal 2^{k-2}-cycle scheme. This construction is best seen in the second representation; the nodes with phases labelled 0 and 1 are the two added nodes.

The 22-cycle scheme shown in Fig. 1(e) is an adaptation of the 32-cycle scheme, with the deepest layer entirely removed and some of the calls in the next layer removed. The nodes which the removed calls informed in the 32-cycle are also removed from the cycle, thus shortening some of the remaining calls. The 32- and 22-cycle schemes are deliberately drawn to emphasize the correspondence between a node, call, or phase in the 22-cycle scheme with the node, call, or phase directly above it in the 32-cycle scheme. Similarly, the 55-cycle scheme shown in part (g) is an adaptation of the 64-cycle scheme with some calls removed from the deepest layer. We will prove that this *elimination method* produces optimal schemes when the number of nodes is not a power of 2.

3. Nestedness, flatness, and fullness

In the first three subsections of this section, we prove that nestedness, flatness, and fullness are necessary properties of optimal line broadcast schemes on cycles. In the fourth subsection, we show that these three properties are sufficient for optimality.

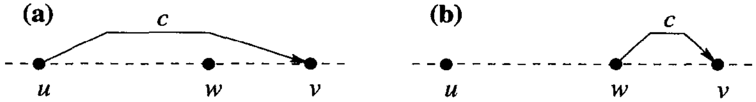

Fig. 2. Nestedness.

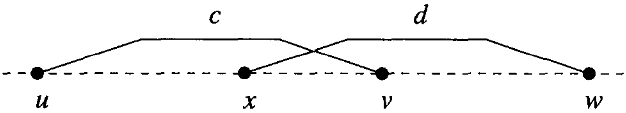

Fig. 3. Crossed calls.

3.1. Nestedness

Definition 1. A broadcast scheme S is *nested* if no call of S passes through an informed node.

Lemma 2. *Every optimal cycle scheme is nested.*

Proof. Assume S is an optimal scheme that is not nested. Then some call c in S goes through an informed node w, as shown in Fig. 2(a). (In the figures in this section, dashed lines indicate paths of one or more links.) Since every link between u and v is used by c, w cannot originate a call while c is being made. A cheaper scheme is possible by letting w, instead of u, inform v, as shown in Fig. 2(b). Thus, a non-nested scheme cannot be an optimal scheme. □

We note that nestedness is a property of the embedding.

Definition 3. A call d in a cycle scheme is *under* a call c if every link which is used by d is also used by c. A node v is *under* call c if v is not the sender or receiver of c but c goes through v. A link ℓ is *under* call c if c uses ℓ.

It follows immediately from Definition 3 that there can be a node under a call only if the call has total length greater than 1 and, in a nested scheme, only if the node is informed after the sender and receiver of the call have been informed.

Using Lemma 2, it is easy to show that the situation shown in Fig. 3 cannot occur in an optimal scheme. Since c and d share an edge, they cannot occur in the same phase. Furthermore, the endpoint x of d is under c, and the endpoint v of c is under d. If d occurs first, then x is informed before c occurs and nestedness prohibits c. Similarly, if c occurs first, then nestedness prohibits d. We have established the following property.

Property 4. *In a nested scheme, two calls c and d can never* cross; *either c and d use disjoint sets of links (but may share one node if it is the sender of both calls) or one call is under the other.*

One consequence of Property 4 is that optimal line broadcasting in the cycle is actually *path broadcasting*; calls in the same phase cannot share any links and cannot share either sender or receiver. Thus, they must be link- *and* node-disjoint.

Nestedness implies some other useful properties. In the following discussion, let P_{uv} be a directed path from node u to node v in a broadcast tree T and let S be a nested cycle scheme for T. Since u is the first informed node of $S(P_{uv})$, nestedness implies that no call of $S(P_{uv})$ either informs u or goes through u. It follows that u is one of the endpoints of $\sigma(S(P_{uv}))$, the segment of $S(P_{uv})$, and one of the links incident on u is not on $\sigma(S(P_{uv}))$. The following property is true by extension of this reasoning.

Property 5. *Let S be a nested cycle scheme for a broadcast tree T and let P_{uv} be a directed path from node u to a node v in T. Then the segment $\sigma(S(P_{uv}))$ is a simple path in the cycle and either u or v is the first informed node on the segment.*

We can also conclude that each node on $\sigma(S(P_{uv}))$ is either u or a descendant of u in S. To establish this, suppose to the contrary that some node x is on $\sigma(S(P_{uv}))$ but is neither u nor a descendant of u in S. In particular, x is not on $S(P_{uv})$, so it can only be on $\sigma(S(P_{uv}))$ because a call of $S(P_{uv})$ goes through x. Any call through x must use both links incident on x (because a call is always mapped to a simple path in the network), so x cannot be an endpoint of $\sigma(S(P_{uv}))$. By nestedness, x is informed during a later phase than u so x cannot be an ancestor of u. Thus, u and x have a common ancestor w in S that is not on $\sigma(S(P_{uv}))$. Since x is not an endpoint of $\sigma(S(P_{uv}))$, some call on the path $S(P_{wx})$ must go through an endpoint of $\sigma(S(P_{uv}))$ to reach x, and must therefore cross a call of $S(P_{uv})$ that goes through x. This contradicts Property 4 and proves the following property.

Property 6. *Let S be a nested cycle scheme for a broadcast tree T and let P_{uv} be a directed path from node u to a node v in T. Then either u or v is the ancestor in S of all other nodes on $\sigma(S(P_{uv}))$.*

We can use Property 6 to generalize Property 4 from calls to paths and then to subschemes. First consider two paths $S(P_{uv})$ and $S(P_{wx})$ in a nested cycle scheme. If w is not a descendant of u in S, then no node on $S(P_{wx})$ can be a descendant of u and Property 6 implies that no node on $S(P_{wx})$ can be on $\sigma(S(P_{uv}))$. Furthermore, $\sigma(S(P_{uv}))$ and $\sigma(S(P_{wx}))$ have no common links. If $w=u$, then $\sigma(S(P_{uv}))$ and $\sigma(S(P_{wx}))$ are disjoint except for their common endpoint. It is now straightforward to generalize this property from paths to subschemes.

Property 7. *Let S be a nested cycle scheme for a broadcast tree T and let T_u and T_w be subtrees of T. If u is not in T_w, and w is not in T_u, except that u can be w, then the segments of $\sigma(S(T_u))$ and $\sigma(S(T_w))$ share no links in the cycle.*

Now, consider a nested cycle scheme S with originator s. By Property 6, the segment $\sigma(S(P_{sx}))$ of every path $S(P_{sx})$ has s as an endpoint. If we arbitrarily assign the directions 'left' and 'right' in the cycle, then we can unambiguously say that $\sigma(S(P_{sx}))$ is to the left or to the right of s. Let $\sigma(S(P_{sL}))$ be the longest segment to the left of s and let $\sigma(S(P_{sR}))$ be the longest segment to the right of s. Then every link used by S is in either $\sigma(S(P_{sL}))$ or $\sigma(S(P_{sR}))$. By Property 7, s is the only node shared by $\sigma(S(P_{sL}))$ and $\sigma(S(P_{sR}))$, and the two segments share no links. It follows that there is at least one link on the cycle which is in neither segment, so there is a link which S does not use. However, there is at most one such link since there would be an uninformed node otherwise.

Property 8. *There is exactly one* unused link *in any nested cycle scheme.*

3.2. Flatness

Let T be a broadcast tree with n nodes. We will say that a node is a *top node* of T if it is the originator, one of the first two nodes called by the originator, or the first node called by another top node. The *top path* of T is the simple path consisting of the top nodes of T and the edges of T representing calls to top nodes. The top path of T is *layer* 0 of T. The subtrees that remain after the edges of layer 0 are removed from T are called the *layer* 1 *trees* of T. Since each layer 1 tree is a broadcast tree, we can use our definitions recursively. Thus, the top path of a layer 1 tree is a *layer* 1 path of T, and the union of all layer 1 paths is *layer* 1 of T. When the edges of layer 1 are also removed from T, the *layer* 2 trees of T remain. Continuing, we see that each edge of T belongs to exactly one layer and that the non-empty layers of T are numbered consecutively starting at 0.

A layer structure is a property of a broadcast tree. The layers of a broadcast scheme are determined by the layers of its broadcast tree, so we use terms like *top call* and *layer p call* in the obvious way. We will use the term *bottom scheme* to refer to a subscheme for a layer 1 tree of a broadcast tree and *bottom call* to refer to any call in a bottom scheme (i.e., any call that is not a top call).

The next property, flatness, is a property of an embedding.

Definition 9. A simple path P of a broadcast tree is embedded *flat* by a cycle scheme S if every node of P satisfies one of the following conditions.

- If a node on P makes two calls on P, then S embeds the two calls in opposite directions into the cycle.
- If a node on P makes a call on P and receives a call on P, then S embeds the two calls in the same direction into the cycle.

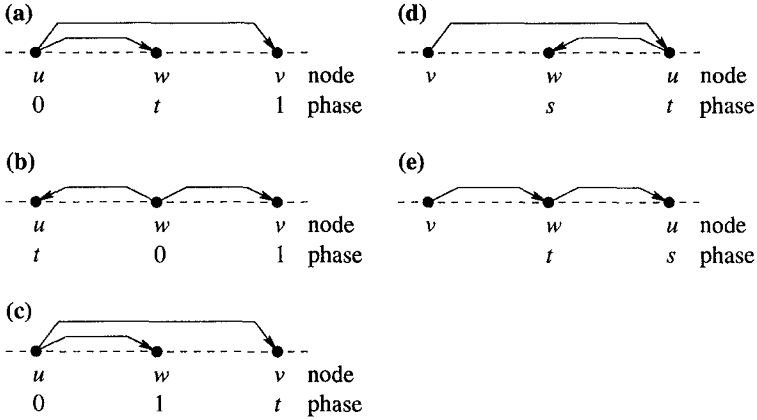

Fig. 4. Directions of top calls.

Definition 10. A cycle scheme S is flat if each layer p path in each non-empty layer p of S is embedded flat.

Flatness and nestedness are independent properties, despite the fact that the proof of Lemma 11 below uses the nestedness of an optimal scheme. A top path can fold on itself (so it is not embedded flat) and yet still be nested. On the other hand, two bottom schemes in a flat scheme can cross and thus violate nesting.

Lemma 11. *The top path of an optimal cycle scheme is embedded flat.*

Proof. Let S be an optimal cycle scheme and u a node on the top path of S. If u is the root of the top path, then u is the originator of S. Suppose that in S, u calls v at phase 1 and that the next call by u is at phase $t>1$ to w, such that w is under the call from u to v. The situation is shown in Fig. 4(a). Then we can obtain a cheaper scheme by making w the originator and having w call v at phase 1 and then u at phase t as illustrated in part (b). (Note that nestedness prohibits the situation shown in part (c).)

Now suppose that u is not the root of S. Then u is called by some other node v at some phase t. Suppose that u's first call in S is at phase $s>t$ to w, and that w is under the call from v to u as shown in Fig. 4(d). (Note that since S is an optimal scheme, u cannot call *through* v by nestedness.) As shown in part (e), we can obtain a cheaper scheme by having v call w at phase t and then having w call u at phase s.

In both of the cheaper schemes, the only calls of u, v, and w that are different are the ones shown explicitly, and the calls are made during the same phases as they were in S. □

Property 12. *If S is a scheme, u is a top node of S, and the bottom scheme of u in S is non-empty, then u has two neighbours on the top path of S.*

If u is the root of the top path, then Property 12 is true because the first two edges used by the originator are on the top path by definition. If u is not the root of the top path, then u is informed by a top call and the first call made by u is also a top call by definition. In both cases, both top calls involving u are made before any bottom calls.

Definition 13. A subscheme $S(T_u)$ is *contiguous* if its segment $\sigma(S(T_u))$ contains only nodes of $S(T_u)$.

Lemma 14. *If S is a nested scheme on a cycle, and S' is a bottom scheme of S, then S' is contiguous.*

Proof. Let S be a nested scheme for a broadcast tree T. By Property 12, if u is the root of a bottom scheme of S, then u has a neighbour v and a neighbour w on the top path of S. Either u calls v and w, or v (say) calls u and u calls w. In the first case, shown in Fig. 5(a), u is the originator of S, T_v and T_w are the largest subtrees of T rooted at v and w, and $S(T_v)$ and $S(T_w)$ are the associated subschemes. Since u is the originator, it cannot be in either $S(T_v)$ or $S(T_w)$. Neither v nor w can be in the bottom scheme $S(T_u)$ because they are informed by top calls. By Property 7, S does not map any node in T_v or T_w closer on the cycle to u than any node in T_u. Since the three subschemes account for all nodes in S, $S(T_u)$ must be contiguous. The second case can be argued similarly, but there are two sub-cases since either v is the originator of S or v is between the originator and u on the top path of S. Fig. 5(b) shows the second sub-case in which v is not the originator. In both sub-cases, $S(T_u)$ is the bottom scheme of u, T_w is the largest subtree of T rooted at w, and we define T_v to be the subtree containing all nodes not in T_u or T_w. In either sub-case, u is not in either $S(T_v)$ or $S(T_w)$ and neither v nor w can be in the bottom scheme $S(T_u)$, so we can apply Property 7 as before. $\square$

Let S be a nested scheme whose top path P is embedded flat. By Lemma 11 and Property 5, the embedded path $S(P)$ is simple. Let u be any top node of S and let $S(T_u)$ be the bottom scheme of u in S. By Property 12, if $S(T_u)$ is non-empty, u has two neighbours on the top path, and these neighbours and u are all informed before any descendant of u in $S(T_u)$. By nestedness, no call in $S(T_u)$ goes through u or its neighbours on the top path, so we get the following property.

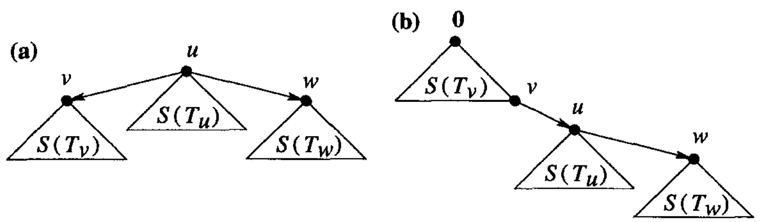

Fig. 5. Contiguous bottom schemes.

Property 15. *If S is a nested scheme and the top path of S is embedded flat, then no top call of S is under any other call of S and each bottom call of S is under exactly one top call of S.*

Lemma 16. *Every optimal cycle scheme is flat.*

Proof. Let S be an optimal cycle scheme. By Lemmas 2 and 11, and Property 15, the top path of S is embedded flat and each bottom call of S is under exactly one top call of S. Since S uses every link of the cycle except one (Property 8), its segment is $n-1$ links long, and the contribution of the top path to the total cost of the scheme is independent of any details of the bottom schemes including the positions of their roots. By Lemma 14 each bottom scheme is contiguous, so no detail of one bottom scheme affects the cost of any other bottom scheme. Thus, each bottom scheme must be an optimal scheme for its segment, and the position of its root will be a position that minimizes the cost of the bottom scheme. It follows that an optimal bottom scheme for a segment will look exactly like an optimal scheme for a cycle with the same number of nodes and it will share all of the properties of an optimal cycle scheme. In particular, the top path of each bottom scheme must be embedded flat. Repeating the argument recursively, we see that S must satisfy Definition 10. □

3.3. *Fullness*

The nestedness and flatness properties provide sufficient structure to enable us to calculate the total extra length of a scheme. Let S be a flat, nested cycle scheme on n nodes. By Property 15, each bottom call of S is under exactly one top call of S. By Lemma 14, the bottom schemes are contiguous, and no call in a bottom scheme is under any call in another bottom scheme. By Lemmas 2 and 16, each bottom scheme is also nested and flat, so, each layer 1 call is under exactly one layer 0 call, and is under no other calls. We can repeat this argument recursively to show that any layer p call is under exactly one layer r call, $0 \leqslant r \leqslant p-1$, and no other calls. If $q-1$ is the deepest layer of S, there are no calls under the layer $q-1$ calls, so all layer $q-1$ calls are local calls.

Now, we can describe a procedure to determine the total extra length of a flat, nested scheme S for a broadcast tree T on n nodes. Start by removing any leaf u from layer $q-1$ of S to obtain a scheme for $n-1$ nodes. More precisely, remove u and the edge that informs u from T, and remove u from the cycle by merging the links incident on u. Since u was a local call, it contributed 1 to the total length. However, u was under $q-1$ calls in lower numbered layers, and each of these calls will be shorter by one link. Thus, the new cycle scheme for $n-1$ nodes has $q-1$ less extra length than the original scheme S. The removal of u preserves nestedness and flatness: no call now goes through any different nodes than it did in the original scheme, so nesting is preserved, and the flatness of the layers is not affected by making the paths shorter.

We can repeat the procedure until all calls have been removed and each unit of extra length has been counted once. Letting $N(p)$ denote the number of calls in layer p of S, $0 \leqslant p \leqslant q-1$, the total extra length of S is

$$L(n) = \sum_{p=0}^{q-1} N(p) \cdot p. \tag{1}$$

The sum $L(n)$ is minimized by an optimal scheme on n nodes. It is clear from the procedure for determining $L(n)$ that $L(n)$ will be minimized if there are as many calls as possible in lower layers and the fewest possible number of calls in layer $q-1$. We will show that this property, which we call *fullness*, is a property of all optimal schemes.

Definition 17. Let T_k denote the complete binomial tree on 2^k nodes, and let $\Lambda(k)$ be the number of layers in T_k. Define $M(k,p)$ to be the number of calls in layer p of T_k, $k \geqslant 0$, $0 \leqslant p \leqslant \Lambda(k)-1$. A broadcast scheme on a tree T with n nodes, $2^{k-1} < n \leqslant 2^k$, and q layers is *full* if T is an MBT, each layer r of T has $M(k,r)$ calls, $0 \leqslant r < q-1$, and layer $q-1$ has at most $M(k,q-1)$ calls.

Lemma 18. *Every optimal scheme for a cycle with 2^k nodes, $k \geqslant 0$, is full.*

Proof. Any optimal scheme must be nested and flat by Lemmas 2 and 16. Create a flat, nested, full scheme S for a binomial tree T_k as follows. First, lay out the top path P flat and without overlap into a cycle which has as many nodes as there are top nodes. Next, create the bottom schemes of S by replacing each node of $S(P)$ by a contiguous embedding of the layer 1 tree rooted by that node. In the process, we stretch out the initial path $S(P)$ by inserting nodes in the cycle to accommodate the nodes in the bottom schemes of S. The details of the bottom schemes are determined recursively. It is clear that this construction gives a scheme S that is flat and nested, and S is full by definition. Note that T_k is the unique MBT on 2^k nodes and that the number of nodes in each of its layers is fixed. Increasing the number of nodes in any layer would increase the number of phases and the resulting tree would not represent a minimum-time broadcast. Decreasing the number of nodes in any layer would require the addition of a new layer and this would also increase the number of phases. Since S has the maximum possible number of calls in each layer, the sum $L(n)$ is minimized. $\square$

We can create a flat, nested, full scheme for any n-cycle by starting with a flat, nested, full scheme for the binomial tree T_k with 2^k nodes, $k = \lceil \log n \rceil$. We then remove $2^k - n$ calls from $S(T_k)$ using the same procedure that we used to calculate $L(n)$. That is, we repeatedly delete a leaf from the deepest non-empty layer of $S(T_k)$ until n nodes remain. Let T be the broadcast tree on n nodes that results. The procedure preserves nestedness and flatness and the order in which nodes are eliminated ensures that $S(T)$ is also full. Also, T is an MBT since it has the correct number of phases, $k = \lceil \log n \rceil$.

Before proving that $S(T)$ is optimal, we need to show that it is not possible by some other method to find an MBT T' with n nodes which has more calls at lower layers than T_k.

Definition 19. The *capacity* $C(n, p)$ is the maximum possible number of edges in layer p of any MBT with n nodes.

Lemma 20. $C(n, p) \leqslant M(k, p)$, $k = \lceil \log n \rceil$.

Proof. We prove the result using strong induction on k. The statement of the lemma clearly holds for the base cases $k=0$ and $k=1$ in which $n=1$ and $n=2$. Now assume that the result holds for every $j<k=\lceil \log n \rceil$ and consider the complete binomial tree T_k with 2^k nodes. T_k is an MBT and satisfies the statement of the lemma by definition. T_k has a root r which makes k calls $c_1, c_2, \ldots, c_i, \ldots, c_k$ to the roots of its subtrees $T_{k-1}, T_{k-2}, \ldots, T_{k-i}, \ldots, T_0$ which are complete binomial trees with $2^{k-1}, 2^{k-2}, \ldots, 2^{k-i}, \ldots, 2^0$ nodes, respectively. Any MBT T' with n nodes, $2^{k-1} < n < 2^k$, can be constructed from T_k by eliminating some of the calls in T_k, and the subtrees of T_k below those calls, and then perhaps moving some calls and their subtrees to different layers. By the induction assumption, no modifications within the subtrees $T_0, T_1, \ldots, T_{k-1}$ can violate the capacity restrictions of binomial trees with $2^0, 2^1, \ldots, 2^{k-1}$ nodes, respectively. So, no modifications of this type can result in more than $M(k, p)$, $k = \lceil \log n \rceil$ edges in any layer p of T'. The only remaining cases to consider are modifications involving the calls c_i. A call c_j can only be moved to an earlier phase (so that c_j and the calls in its subtree could move to lower layers) if some call c_i, $i<j$ in that earlier phase is eliminated or moved to a later phase. However, by the induction assumption, the capacity of each layer of T_{k-j} is less than the capacity of the corresponding layer in T_{k-i}, so this type of modification cannot result in more than $M(k, p)$, $k = \lceil \log n \rceil$ edges in any layer p of T'. □

Lemma 21. *Every optimal cycle scheme is full.*

Proof. The broadcast tree of an optimal cycle scheme must be an MBT, so the first part of Definition 17 is satisfied. Our procedure for finding a flat, nested, full scheme for any n-cycle by eliminating calls from a flat, nested, full scheme for a binomial tree proves that there is a flat, nested MBT satisfying the second part of Definition 17 for any n. The result now follows by Lemma 20 and the fact that Eq. (1) is minimized when the numbers of calls in all lower layers of a scheme are maximized. □

3.4. Sufficiency of nestedness, flatness and fullness

Suppose that S is a minimum-time scheme for the n-cycle which has q layers numbered $0, 1, \ldots, q-1$, and $\omega(n)$ calls in layer $q-1$. If S is flat, nested, and full, then by Eq. (1) and Lemma 21 the total extra length of S is exactly

$$L(n) = \sum_{p=0}^{q-2} M(\lceil \log n \rceil, p) \cdot p + \omega(n) \cdot (q-1). \tag{2}$$

We can now state our main result.

Theorem 22. *A cycle scheme is optimal if and only if it is flat, nested, and full.*

Proof. Lemmas 2, 16, and 21 establish that nestedness, flatness, and fullness are necessary conditions for optimality. To establish sufficiency, suppose that there is a minimum-time scheme S with n nodes that has a cost different than $L(n)$ given in Eq. (2). Since every flat, nested, and full scheme with n nodes has cost $L(n)$, S must lack at least one of these properties, and S cannot be optimal by one of Lemmas 2, 16, and 21. Therefore, $L(n)$ is the only possible cost for an optimal scheme on n nodes and all flat, nested, and full schemes are optimal. This establishes the sufficiency of nestedness, flatness, and fullness. □

4. Construction and analysis of optimal cycle schemes

In this section, we will determine the exact cost of optimal line broadcast schemes for cycles with $n = 2^k$ nodes, $k \geqslant 1$. In the process, we will obtain enough information to describe the cost of optimal cycle schemes for all other values of n. In the first subsection, we describe a method for constructing optimal schemes for 2^k-cycles and analyze the construction to determine the exact cost. In the second subsection, we briefly describe an alternative construction method. In the third subsection, we explain how to derive an optimal scheme for an n-cycle, $2^k - 1 < n < 2^k$, from an optimal scheme for a 2^k-cycle.

4.1. Analysis of optimal 2^k-cycle schemes

If a cycle has $n = 2^k$ nodes, then a complete binomial tree with 2^k nodes is the only possible MBT. So, we can find the total extra length of an optimal cycle scheme with 2^k nodes by determining the number of layers $\Lambda(k)$ in the broadcast tree and the size $M(k, p)$ of each layer p, $p = 0, 1, \ldots, \Lambda(k) - 1$. We showed in Section 3.3 that the total extra length is $L(2^k) = \sum_{p=0}^{\Lambda(k)-1} M(k, p) \cdot p$. We will determine $\Lambda(k)$ and the sizes $M(k, p)$ shortly, but it is easier to first derive $L(2^k)$ directly.

Our direct determination of $L(2^k)$ is based on a recursive construction of optimal broadcast schemes with 2^k nodes. First note that a complete binomial tree with 2^k nodes can be constructed by adding two new nodes to the center of layer 0 of a complete binomial tree with 2^{k-1} nodes and then attaching a complete binomial tree with 2^{k-2} nodes to each of the new nodes in layer 0. The case $k = 5$ is shown in Fig. 6 and in parts (c) and (d) of Fig. 1. We know that in an optimal scheme, each

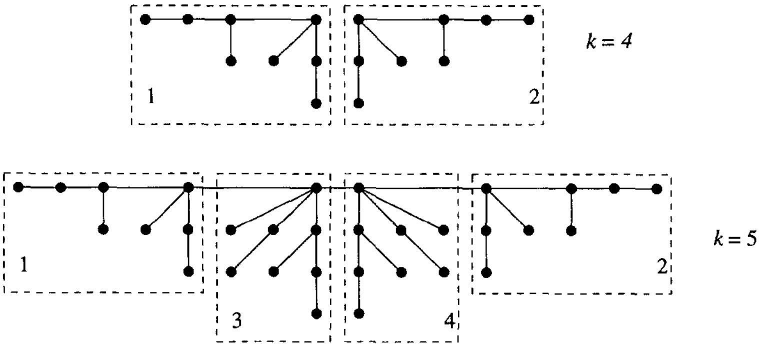

Fig. 6. A recursive construction of optimal cycle schemes.

bottom scheme is flat, nested, and full. It follows that an optimal scheme for 2^k nodes can be derived from an optimal scheme with 2^{k-1} nodes by adding two new nodes to the center of the top path, and then making each of the two new nodes the root of a bottom scheme with 2^{k-2} nodes, such that each of the two new 2^{k-2}-schemes is contiguous. This procedure is illustrated in Fig. 6 for $k=5$. (Also see Fig. 1 for more examples.)

It is easy to see that the calls in dashed boxes labelled 1 and 2 in Fig. 6 together contribute extra length $L(2^{k-1})$. The calls in boxes 3 and 4 each contribute extra length $L(2^{k-2})+2^{k-2}-1$. The term $2^{k-2}-1$ is due to the number of new calls under the top path. This gives the following recurrence relation:

$$L(2^1)=0,$$
$$L(2^2)=0,$$
$$L(2^k)=L(2^{k-1})+2\cdot L(2^{k-2})+2^{k-1}-2, \quad k>2.$$

The solution to the recurrence relation is

$$L(2^k)=\tfrac{1}{9}[2^k(3k-8)-(-1)^k]+1, \tag{3}$$

which is easily verified by direct substitution. The total length of an optimal scheme on 2^k nodes is then $L(2^k)+(2^k-1)$, which is

$$F(2^k)=\tfrac{1}{9}[2^k(3k+1)-(-1)^k]. \tag{4}$$

It is not difficult to show that $\Lambda(k)=\lceil k/2\rceil$, for $k>0$. The case $k=1$ is trivial, since an optimal scheme for 2 nodes has 1 call and 1 layer. To determine $\Lambda(k)$ in general, note that the originator makes its first two calls on layer 0, its next two calls on layer 1, and so on. Since the originator makes one call in each of the k phases, there are at least $\lceil k/2\rceil$ layers. Each layer, except possibly the deepest layer, contains two calls made by the originator in two phases. Thus, all calls of the originator are contained in exactly $\lceil k/2\rceil$ layers. Every other node makes fewer calls in fewer phases than the originator, so $\lceil k/2\rceil$ layers contain all calls of the broadcast scheme.

We can determine $M(k, p)$ by using the construction in Fig. 6, and arguing in much the same way as we did for $L(2^k)$, to derive the following recurrence relation:

$$M(k,p)=0, \quad k\geqslant 1, p\geqslant \left\lceil \frac{k}{2} \right\rceil,$$

$$M(k,0)=2k-1, \quad k\geqslant 1,$$

$$M(k,p)=M(k-1,p)+2\cdot M(k-2,p-1), \quad p\geqslant 1, k\geqslant 3.$$

The solution to this recurrence relation is

$$\begin{aligned} M(k,p) &= 2^p \cdot \left[2\cdot \binom{k-p-1}{p+1} + \binom{k-p-1}{p}\right] \\ &= 2^p \cdot \left[2\cdot \left(\frac{k-p}{p+1}\right) - 1\right] \cdot \binom{k-p-1}{p}, \end{aligned} \tag{5}$$

which can be confirmed by substitution. Substituting Eq. (5) and $A(k)=\lceil k/2 \rceil$ into the equation $L(2^k)=\sum_{p=0}^{A(k)-1} M(k,p)\cdot p$ gives Eq. (3).

4.2. Alternate procedure for creating cycle schemes

In Section 2 we mentioned another recursive procedure for creating optimal schemes for the 2^k-cycle that joins the originators of two mirror-image schemes for 2^{k-1}-cycles by a line call. Fig. 7 shows a few steps of the recursive construction procedure. The schemes in parts (a), (c), and (d) for cycles with 4, 8, and 16 nodes are similar to the schemes shown earlier in Fig. 1. The only difference is that the nodes of the schemes in Fig. 7 are labelled with bit strings. When nodes are visited in order around the cycles, these sequences of bit strings are *binary reflected Gray codes* as described in [11]. A scheme for 2^k nodes is created recursively from a scheme for 2^{k-1} nodes by first placing two mirror-image copies of the scheme for 2^{k-1} nodes side by side with the originators as close to each other as possible. Then, a 0 is prepended to each label in the left copy and a 1 to each label in the right copy, and a line call is added from the originator $00\ldots0$ of the left copy to the originator $100\ldots0$ of the right copy. During a broadcast, this new line call is the call made during the first phase. Parts (a), (b), and (c) of Fig. 7 show the construction of a scheme for 8 nodes from a scheme for 4 nodes. Part (d) shows the scheme for 16 nodes.

The construction procedure produces a scheme with k phases for the 2^k-cycle, since the first phase of the scheme uses one call to begin two subschemes with $k-1$ phases. Thus, the schemes are minimum-time schemes. Several properties of the schemes are immediate consequences of the construction procedure. In each phase i, $1\leqslant i\leqslant k$, exactly 2^{i-1} calls are made and all of the calls in phase i have the same length. Furthermore, the labels of the sender and receiver of a call in phase i are Hamming distance 1 apart, and the bit position in which they differ is the ith position from the left. Thus, a node that is informed by a call in phase i will have a 1 in the ith position from the left, 0's in all positions to the right of this 1, and the label of the node that it calls in each

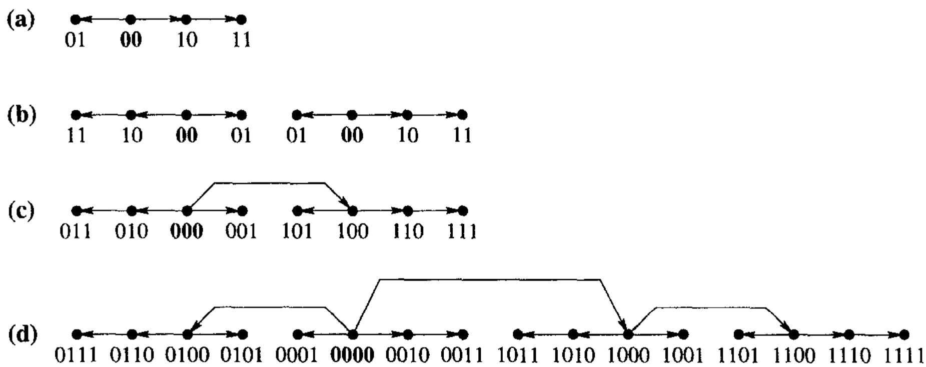

Fig. 7. Recursive construction with Gray code labelling.

phase $j>i$ is obtained by complementing the bit in the jth position from the left. We omit the proof that this procedure produces optimal schemes. The proof can be found in the first author's M.Sc. thesis [8].

4.3. The elimination method for cycles

In Section 3.3, we described a method for creating a broadcast scheme for any n-cycle. First, we construct a flat, nested, full scheme on 2^k nodes, where $k=\lceil \log n \rceil$ using, for example, one of the recursive procedures described earlier in this section, and then eliminate the most expensive calls in the scheme until $n-1$ calls are left. We have shown that this method, which we call the *elimination method*, preserves nestedness and flatness, and by Lemma 20 the resulting scheme is full, so the scheme is optimal by Theorem 22. The total cost, $F(n)$, of the scheme can be obtained by using Eqs. (2) and (5) and noting that there are $\lceil k/2 \rceil$ layers in the flat, nested, full scheme with 2^k nodes. We have only found a closed form solution for $F(n)$ when n is a power of 2 (i.e., Eq. (4)), but there are some interesting observations that we can make about the behavior of $F(n)$. First, we note that each application of the elimination method reduces total cost, so $F(n)$ is monotonically increasing between 2^k+1 and 2^{k+1}. In fact, for any $n \geqslant 6$, all calls removed by the elimination method to get an optimal scheme for n nodes have extra cost at least one. In contrast, the function $F(n)$ decreases between $n=2^k$ and $n=2^k+1$ for $k \geqslant 5$. For example, $F(64)=135$ and $F(65)=116$.

Property 23. $F(n)<F(n+1)$ *for* $n \leqslant 8$ *and for* $n \neq 2^k$ *when* $n>8$.

Property 24. $F(2^k)>F(2^k+1)$ *for* $k \geqslant 5$ *and* $F(2^k)=F(2^k+1)$ *for* $k=3$ *and* $k=4$.

Proof. The cases $k=3$ and $k=4$ can be verified directly. For $k \geqslant 5$, note from Eq. (5) that the capacity of each non-empty layer of an optimal scheme for 2^k+1 nodes is greater than the capacity of the corresponding layer in the scheme for 2^k nodes. In particular, layer 0 in a scheme for 2^k+1 nodes can contain two more nodes than a scheme for 2^k nodes. For $k \geqslant 5$, we can construct a scheme for $2^k + 1$ nodes from a scheme for 2^k nodes by adding two nodes to the ends of the top path (layer 0), and removing one node from the deepest layer. For $k \geqslant 5$, the deepest layer is layer 2 or greater, so the construction shows that $F(2^k) \geqslant F(2^k+1)+1$. Note that the schemes for $2^k + 1$ nodes constructed in this way are not optimal since they do not take advantage of the extra unused capacity in layer 2 (and possibly deeper layers). □

We state our final observation as a theorem.

Theorem 25. *The total length of an optimal scheme for a cycle with $n \geqslant 2$ nodes is*

$$F(n) \leqslant \tfrac{1}{9}[n(3\lceil \log n \rceil + 1) - (-1)^{\lceil \log n \rceil}].$$

Proof. For $n \leqslant 8$, the bound can be verified directly by using the elimination method on the schemes in Fig. 1, and the result is true for powers of 2 from Eq. (4). From Property 24, we know that $F(2^k) \geqslant F(2^k + 1)$ for $k \geqslant 3$, so, using Eq. (4), we get

$$\begin{aligned}
&F(2^{k+1}) - F(2^k + 1) \geqslant F(2^{k+1}) - F(2^k) \\
&\quad = \frac{1}{9}[2^{k+1}(3(k+1)+1) - (-1)^{k+1}] - \frac{1}{9}[2^k(3k+1) - (-1)^k] \\
&\quad = \frac{1}{9}[(2^{k+1} - 2^k)(3k+1) + 3 \cdot 2^{k+1} + 2(-1)^k] \\
&\quad = 2^k\left(\frac{3k+7}{9}\right) + \frac{2}{9}(-1)^k \\
&\quad = (2^k - 1)\left(\frac{3k+7}{9}\right) + \frac{3k+7+2(-1)^k}{9} \\
&\quad > (2^k - 1)\left(\frac{3k+7}{9}\right).
\end{aligned}$$

Therefore, the average decrease in cost due to one application of the elimination method is greater than $(3k+7)/9$. The elimination method always removes a call with largest extra cost, so when the elimination method is applied $\ell < 2^k$ times to obtain a scheme with $n = 2^{k+1} - \ell$ nodes, the largest decreases in cost occur first. It follows that

$$\begin{aligned}
F(n) &< F(2^{k+1}) - \ell\left(\frac{3k+7}{9}\right) \\
&= \frac{1}{9}[2^{k+1}(3(k+1)+1) - (-1)^{k+1}] - \frac{\ell(3(k+1)+1)}{9} - \frac{\ell}{3} \\
&= \frac{1}{9}[(2^{k+1} - \ell)(3(k+1)+1) - (-1)^{k+1}] - \frac{\ell}{3} \\
&< \frac{1}{9}[n(3\lceil \log n \rceil + 1) - (-1)^{\lceil \log n \rceil}]
\end{aligned}$$

for $n \neq 2^k$. □

Farley's upper bound [3] for total length in any network on n nodes is $F(n) \leqslant (n-1)\lceil \log n \rceil$; asymptotically in n, our result is $\frac{1}{3}$ of this bound.

5. Further work

We have generalized the cycle schemes described in this paper to produce schemes for *multi-dimensional cycles* (or *toroidal meshes*). We believe that these schemes are optimal, but the proofs appear to be much more difficult than the proof of optimality for the cycle. For two-dimensional cycles, we have found generalizations of nestedness, flatness, and fullness that are necessary for optimality, but they are not sufficient. For arbitrary higher-degree networks, we know that nestedness is not a property of all optimal schemes because we have found networks for which some originator *must* make a call through an informed node in any optimal scheme. However, it appears that nestedness is a necessary condition for optimality in any vertex-transitive network. We refer the reader to [8] for more details.

In the cycle, we proved that optimal line broadcasting is also path broadcasting, and we relied on this simplifying property in our proofs of optimality. This property is not true for higher-degree networks since nodes can switch through multiple calls. The loss of this simplification appears to be the source of the difficulties in the proofs for higher-degree networks. An interesting line of further work might be line broadcasting in degree-3-regular or degree-3-bounded graphs. A degree 3 node can switch through at most one call, but can be the sender or receiver of another call at the same time.

We mentioned in the Introduction that Iordanskiĭ [7] has used concepts similar to *layers* and *nestedness* to study minimum-cost *unconstrained* embeddings of undirected (unrooted) trees into linear networks (cycles and paths). The embeddings that we have studied are constrained because calls that are made in the same phase must be mapped to link-disjoint paths. We believe that our optimal schemes are also optimal unconstrained embeddings.

Conjecture 26. *An optimal cycle scheme with n nodes has the same total length as an optimal unconstrained embedding of the undirected version of a broadcast tree with n nodes into the cycle with n nodes.*

Acknowledgements

We would like to thank the referees for comments that improved the presentation and for observations that led to improvements in the proofs in Section 3.3.

References

[1] C. Almstrom, Limited line broadcasting in the infinite 1-dimensional grid graph, manuscript, 1988.
[2] J.-C. Bermond, P. Fraigniaud, J.G. Peters, Antepenultimate broadcasting, Networks 26 (1995) 125–137.
[3] A.M. Farley, Minimum-time line broadcast networks, Networks 10 (1980) 59–70.

[4] R. Feldmann, J. Hromkovič, S. Madhavapeddy, B. Monien, P. Mysliwietz, Optimal algorithms for dissemination of information in generalized communication modes, Discrete Appl. Math. 53 (1994) 55–78.

[5] P. Fraigniaud, E. Lazard, Methods and problems of communication in usual networks, Discrete Appl. Math. 53 (1994) 79–133.

[6] S. Fujita, A.M. Farley, Minimum-cost line broadcasting in paths, manuscript, 1994.

[7] M.A. Iordanskiĭ, Minimal numberings of the vertices of trees, Sov. Math. Dokl. 15 (1974) 1311–1315.

[8] J.O. Kane, Line broadcasting, M.Sc. Thesis, School of Computing Science, Simon Fraser University, 1993.

[9] B. Monien, H. Sudborough, Embedding one interconnection network into another, Comput. Suppl. 7 (1990) 257–282.

[10] J.G. Peters, M. Syska, Circuit-switched broadcasting in torus networks, IEEE Trans. Parallel Distributed Systems 7 (1996) 246–255.

[11] E.M. Reingold, J. Nievergelt, N. Deo, Combinatorial Algorithms: Theory and Practice, Prentice-Hall, Englewood Cliffs, NJ, 1977, pp. 173–176.