Networks Immune to Isolated Line Failures

Arthur M. Farley and Andrzej Proskurowski
Department of Computer and Information Science, University of Oregon, Eugene, Oregon 97403

A network is immune to a set of failures if all message transfers between operative sites can be completed in the presence of such failures. A set of line failures is isolated if no two failing lines are incident to the same site. Several classes of isolated line failure immune (ILFI) networks are defined, including a class with fewest lines for a given number of sites. An algorithm is presented which turns an arbitrary tree into one of these minimum ILFI networks and computes routing tables for the new network.

I. INTRODUCTION

We model a communication network by a graph $G = (V, E)$ consisting of a set V of vertices, or communication *sites*, and a set E of edges, or communication *lines*. Each line (u, v) connects a pair of sites u and v. Information is disseminated within such networks in the form of *messages*. Each message has a sender and a receiver (or set of receivers). Messages are transmitted by calls which are placed between directly connected (i.e., neighbor) sites over lines of the network. We assume that a *call* consists of message transmission and an associated verification protocol. As such, a call may succeed or fail.

A message is transferred between distant (i.e., non-neighboring) sites by being forwarded along a path of lines in the network which connects the sending and receiving sites (i.e., by the store and forward method). The routing of messages is an important activity in such communication networks. *Message routing* refers to the process of determining a path between two sites of a network. We assume that message routing is based upon *routing tables*, one for each site. A site's routing table contains entries indicating which neighbor that site is to call for each other site as message receiver.

An important class of network properties is concerned with network's capability of withstanding failures of its components by rerouting message transfers. For example, a network is k-site (k-line) connected iff at least k sites (lines) must be removed (i.e., be inoperative or fail) to disconnect the network. A network is *immune* to a set of failures iff all message transfers between operative sites can be completed in the presence of such failures (i.e., the operative sites remain connected). For instance, a k-connected (site or line) network is immune to sets of failures of limited size.

The notion of networks which are immune to isolated failures, where both site and

NETWORKS, Vol. 12 (1982) 393–403

CCC 0028-3045/82/040393-11$02.10

line failures are possible has been introduced in Farley [2]. A set of failures is *isolated* iff they are pairwise isolated. Two line failures are isolated if they are not incident to a common site (i.e., do not share a common end). Two site failures are isolated if the sites are not neighbors. A line and a site failure are isolated if the line is not incident to a neighbor of the site. In [2], several classes of isolated failure immune networks are presented, including a class with minimum number of lines for a given number of sites.

In this article, we concentrate on networks which are immune to isolated line failures (i.e., we assume absence of site failures). We shall refer to such networks as *isolated line failure immune* or *ILFI* networks. The study of networks immune to isolated line failures is motivated by both theoretical and practical concerns. Connectivity properties of such networks differ from those of globally or locally k-line or k-site connected networks. For an arbitrary k, there exist k-connected networks which are not immune to isolated failures. Conversely, an ILFI network does not even have to be 3-connected. However, assuming that line failures are independent random events, we may expect an ILFI network to be disconnected by a pair of line failures less frequently than, for instance, a minimally 2-connected network. Furthermore, global connectivity of an ILFI network can be determined by combining the results of each site polling the status of its adjacent lines. This property allows also local determination of a schedule for line maintenance that does not disconnect the network. Finally, message rerouting in an ILFI network can be done very efficiently, by a mere detour, rather than backtracking from an encountered failure.

We first define several classes of ILFI networks, including a class requiring the fewest number of lines for a given number of sites. We then present an algorithm which transforms an arbitrary tree into one of these minimum ILFI networks. The algorithm also determines routing tables with preferred and alternate calls for each site. These tables allow a simple calling protocol to complete calls between any two sites as long as line failures are isolated. We conclude with a brief discussion of further research questions.

II. CLASSES OF ILFI NETWORKS

Every effective communication network is connected. A *connected* network contains at least one path between every pair of distinct vertices. A *tree* network is a connected acylic network. It has certain advantages when selected as the topology for a communication network. A tree network requires only $|V| - 1$ communication lines, which is the fewest lines possible in a connected network with $|V|$ sites. Furthermore, one and only one path exists between any two sites. This can simplify the construction and maintenance of routing tables within the network. Tree networks are also advantageous for routing messages during the information dissemination process known as broadcasting [5]. In broadcasting, one site is the originator of a message which is to be transferred to all other sites as receivers. When broadcasting is a tree network, a site receiving the message simply calls all other sites to which it is connected. This completes broadcasting after only $|V| - 1$ calls, which is the minimum number required. Such a "flooding" procedure in more general networks can lead to much redundant calling and the attendant degradation of network performance.

Other properties of tree networks are disadvantageous for communication applications. Traffic bottlenecks can develop, with no alternate routing available for relief.

Most damaging is the fact that tree networks are only immune to failure of leaf sites, where a *leaf* site is a site having only one neighbor. Any other line or site failure disconnects the network. Thus, a tree network is not immune to a single line failure. To improve the immunity of a tree network, lines must be added providing the redundancy necessary to overcome failures. One obvious way to add redundancy is to duplicate the lines of a given tree, as demonstrated in Figure 1(a). The resultant *double-tree* network is still relatively cheap, requiring only $2|V| - 2$ lines, and maintains the agreeable routing characteristics of a tree. Furthermore, it is trival to see that double-trees constitute a subclass of ILFI networks. At most one of each pair of duplicate lines can fail when line failures are isolated.

Proposition 2.1. If a network G is a double tree, then G is an ILFI network.

A double tree is a recursive network in that a double tree with n sites can be constructed from a double tree with $n - 1$ sites by addition of a new site connected to an existing site by two lines. We can generalize the notion of a double tree to obtain a broader class of ILFI networks by eliminating the restriction that both new lines connect to the same existing site. A *2-recursive* network is a single site or, given any 2-recursive network on n sites, a 2-recursive network on $n + 1$ sites is obtained by directly connecting a new site by a total of two lines to one or two sites of the given network. Figure 1(b) presents a 2-recursive network, with sites numbered according to a possible recursive construction.

Proposition 2.2. If G is a 2-recursive network, then G is an ILFI network.

Proof (by induction on number of sites). When $|V| = 1$ or 2, the property is immediate. Assume that any 2-recursive network having less than n sites is an ILFI network and consider G having n sites. Because of the recursive construction rule, there is a site

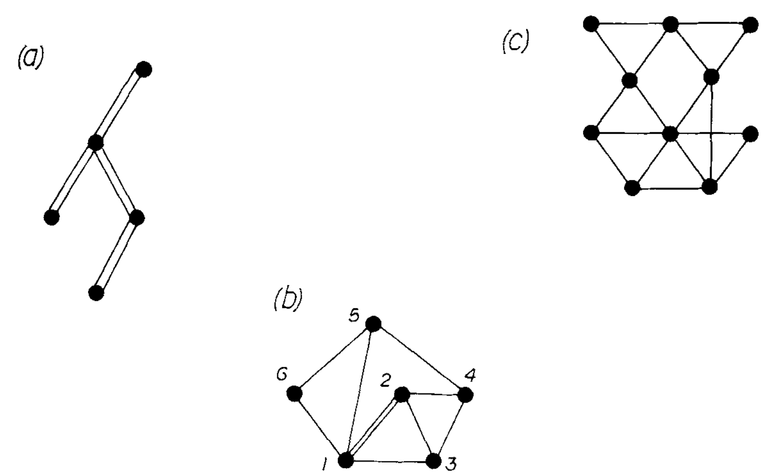

FIG. 1. Three subclasses of ILFI networks: (a) a double tree, (b) another 2-recursive network, and (c) triangled network.

v of degree 2 in G. The subnetwork induced in G by $V - \{v\}$ is 2-recursive, has $n - 1$ sites, and thus is ILFI. Only one line connecting site v to that subnetwork can be down if line failures are isolated. As the subnetwork remains connected under isolated line failures, so does G. ∎

The above proposition indicates that ILFI networks can easily be expanded to include new sites. To add a new site, simply connect it to any one or two sites by a total of two lines. The decision to which sites it will be connected can be made according to other criteria, such as minimizing lengths of the new lines or distance within the new network.

We consider one more general class of ILFI networks before turning our attention to reducing the number of lines required. A *triangle* is a cycle of three sites and three lines. A *triangled* network is a network such that every line is part of a triangle. An example is given in Figure 1(c).

Proposition 2.3. If G is a triangled network, then G is an ILFI network.

Proof. Suppose that a message transfer is to be made from site u to site v. Let $u = S_0, S_1, \ldots, S_k = v$ be the sequence of sites on a path between u and v in G. For every line connecting sites S_i and S_{i+1} $(0 \leqslant i < k)$ there exists a site W_i adjacent to both of the sites, as every line is part of a triangle. If the line connecting S_i and S_{i+1} is down then the message can be rerouted from S_i to site W_i and from W_i to S_{i+1}. Both lines used in rerouting must be operative if failures are isolated. Thus, G is an ILFI network. ∎

Though 2-recursive networks and double tree networks only require $2|V| - 2$ lines, this is not the minimum number of lines in an ILFI network with a given number of sites (see, for instance, the triangled network in Fig. 1(c)). A network G is a *minimum ILFI* network iff G has the fewest lines possible (for its given number of sites). We define the following class of ILFI networks, which are minimum ILFI networks. An *abc-network* is a single site, or, given any *abc*-network G, another *abc*-network can be realized by applying one of the following three rules:

(a) if G has an odd number of sites, connect a new site to one or two sites of G by a total of two lines;
(b) connect two new sites, themselves connected by a line, to a site of G, forming a triangle;

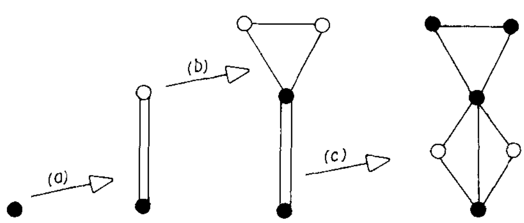

FIG. 2. Application of rules (a), (b), and (c) results in an *abc* network (new sites indicated by light circles).

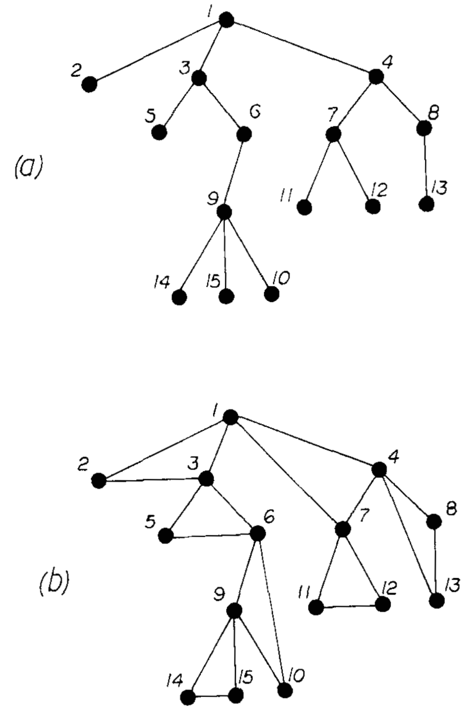

FIG. 3. (a) An input for the algorithm IMMUNIZE and (b) the resulting triangle cactus.

(c) replace a line of G by adding two new sites, each connected to both end-sites of the removed line, forming a four-cycle.

Figure 2 illustrates application of these three rules in a construction of an *abc* network. Note that rule (a) may be applied at most once in any construction. An interesting subclass of *abc* networks is obtained if only rule (b) is applied during construction. We call the resultant network a *triangle cactus*; it is a cactus network [1] in which every line belongs to exactly one triangle. Thus a triangle cactus is obviously triangled. An example is given in Figure 3(b).

Theorem 2.4. If G is an *abc* network then G is an ILFI network.

Proof (by induction on number of sites). A single site is an ILFI network. Assume that the propostion holds for networks with less than $n > 1$ sites, and consider an *abc*-network G with n sites. By definition, G has been constructed from an *abc*-network G' of less than n sites by application of rule (a), (b), or (c). By induction, G' is an ILFI network. If rule (a) has been applied, by our proof for 2-recursive networks (Theorem 2), we know that G is an ILFI network. Similarly, sites forming a triangle cannot be disconnected by any isolated line failure. Thus, if rule (b) has been applied,

G is an ILFI network. After rule (c) is applied, it takes two line failures to disconnect the new four-cycle. Such isolated failures always leave each new site connected to one of the ends of the removed line. The two failures also ensure that other lines incident to the two end sites do not fail, just as failure of the removed line did in G'. Therefore, G is an ILFI network. ∎

The following statement results from a straightforward consideration of the number of sites and lines added by each rule.

Proposition 2.5. If G is an *abc* network with n sites, then G has $\lceil 3(n - 1)/2 \rceil$ lines.

In [3], we show that this is the fewest lines possible in an ILFI network with n sites by defining reduction operations which are effectively inverses of the rules for forming *abc* networks. When applied to an ILFI network, each operation results in another ILFI network with either one site and two lines, or two sites and three lines less than the original network. For an arbitrary ILFI network G, we perform a sequence of applicable reductions yielding eventually an irreducible network G'. If G' contains only a single site, then our proof is completed due to the ratio of lines to sites removed by the reductions. If G' has more than one site, then by virtue of its irreducibility and by the ILFI property, it has no site of degree 2 adjacent to a site of degree less than 4. Furthermore, no site of degree 4 in G' is adjacent to more than two sites of degree 2. By the degree count over the sites of G' subject to these constraints, we get that the ratio of lines to sites in G' is not smaller than 3/2. This, together with the ratio guaranteed by the reduction operations proves that G has at least $\lceil 3(n - 1)/2 \rceil$ lines. We therefore have the following theorem.

Theorem 2.6. If G is an *abc* network then G is a minimum ILFI network.

It remains an open question as to whether *abc* networks completely characterize the class of minimum ILFI networks.

III. MINIMUM ILFI NETWORKS FROM TREE NETWORKS

In this section we present an algorithm which transforms a given tree network with an odd number of sites into an ILFI network by adding lines to produce a triangle cactus as output. The algorithm also determines routing tables for the output network, allowing a simple calling protocol to complete all message transfers as long as the line failures are isolated.

The input tree network T is assumed to be presented in a *father array* [4] reflecting a recursive construction of T (i.e., father$[i] < i$ for site i, $1 < i \leqslant |V|$). Site 1 is the root site in such a representation and may be selected arbitrarily from sites of an unrooted network. Our algorithm processes sites from $|V|$ down to 2 using *marks* to record the state of computation. As site i is processed, it either marks its father if the father is unmarked or it becomes connected by a new line to the site currently marking its father. If site i is itself marked, a new line is added between the marking site and father$[i]$. When a line is entered into a triangle by the addition of a line, we say the line has been (locally) *immunized*.

A routing table named *call* is first established for the input tree T. Its entries indicate which neighbor a site is to call for another site as message receiver. As there is a single path between two sites of a tree, such entries are unique. During processing of a site i, entries in the routing tables of i and its neighbors are made if a new triangle is created. Two routing tables are created for sites of G, named *pref* and *alt*, indicating the preferred (potentially yielding a shorter path) and alternate (in the case of a failure) calls from a site for each other site as message receiver.

Algorithm 3.1 IMMUNIZE

Input: A tree $T = (V, E)$ with odd number of sites $|V| = n$, represented by array *father* $[2 \ldots n]$ according to a recursive numbering of sites, i.e., such that father $[i] = j$ iff $j < i$ and (i, j) in E.

Output: A minimum ILFI network $G = (V, E')$ and routing tables indicating preferred and alternate calls for each site of G.

Data structures: Routing tables as n by n arrays *call*, *pref*, and *alt*; an array *mark*, such that mark$[i] = j$ implies that a line (i, j) of T is yet to be immunized.

```
Method: {0.Initialize}
                InitialTables;
                RouteTree;
                E':=E;
        {1.Process sites by pruning the leaves}
                for i:=n downto 2 do
        {1.1}   begin k:=father[i];
                        if mark[i]=0 {lines to all sons of i are immunized}
        {1.2}           then if mark[k]=0 {lines to all other pruned sons of k are}
        {1.2.1}                 then mark[k]:=i                    {immunized}
        {1.2.2}                 else {there is another line to be immunized}
                                begin j:=mark[k]; mark[k]:=0;
                                        {connect the two sons}
                                        E':=E ∪ {(i,j)};
                                        RouteSites(i,j,k);
                                        end {of two-son triangle}
        {1.3}           else {a line to a son of i is yet to be immunized}
                                begin j:=mark[i];
                                        {connect the son and its grandfather}
                                        E':=E ∪ {(j,k)};
                                        RouteSites(j,k,i);
                                        end {of son-grandfather triangle}
                                end {of processing};
```

```
procedure InitialTables;
begin for i:=1 to n do
      begin mark [i]:=0;
            for j:=1 to n do call[i,j]:=0;
            call[i,i]:=i
            end
      end;
```

```
procedure RouteTree;
begin for i:=n downto 2 do
      begin k:=father[i];
            for j:=1 to n do
            if call[i,j]=0 then call[i,j]:=k
                           else call[k,j]:=i
            end
      end;

procedure RouteSites(base1,base2,top);
{routes a triangle newly created by a line added between sites
base1 and base2, with site top adjacent to both of the sites}
begin for i:=1 to n do
        begin j:=call[top,i];
              case j of
              base1: begin {shortcut from base2}
                        pref[base2,i]:=base1; alt[base2,i]:=top;
                        pref[top,i]:=base1; alt[top,i]:=base2
                        end;
              base2: begin {shortcut from base1}
                        pref[base1,i]:=base2; alt[base1,i]:=top;
                        pref[top,i]:=base2; alt[top,i]:=base1
                        end;
              others: begin pref[base1,i]:=top; alt[base1,i]:=base2;
                              pref[base2,i]:=top; alt[base2,i]:=base1
                              end
              end {cases}
        end; {of routing}.
```

Figure 3 illustrates the execution of IMMUNIZE on a given tree network. The sites are numbered as in the recursive father array and are processed by IMMUNIZE in decreasing order of those numbers.

In proving correctness of algorithm IMMUNIZE, we first establish that the output network G is a minimum ILFI network.

Theorem 3.2. The network G output by IMMUNIZE is a triangle cactus.

Proof. A triangle cactus is a network such that every site is in at least one triangle and every line is part of one and only one triangle. As IMMUNIZE starts with a tree, G is connected and we need only prove that every line of G is part of one and only one triangle. That every line of T is immunized exactly once follows from the one-to-one correspondence between lines of a rooted tree network and the nonroot sites of the network. A site of T is processed exactly once (step 1.1 of IMMUNIZE) either being immunized immediately (steps 1.2.2 or 1.3), or leaving a mark (step 1.2.1). As this mark is considered, the corresponding line is immunized (steps 1.2.2 or 1.3) and the mark removed explicitly (step 1.2.2) or implicitly (step 1.3), never to be considered again. It is easily verified that any processed site cannot have an outstanding mark. Thus, after processing site 2, a line not yet in a triangle must be marking site 1. A case analysis indicates that when the number of sites is odd this situation cannot occur. New lines added by IMMUNIZE are obviously part of one and only one triangle as the other two lines are from the input tree. Lines of the tree are in at most one triangle, as established above. Thus, G is a triangle cactus. ∎

Corollary 3.3. The network G output by IMMUNIZE is a minimum ILFI network.

We now consider correctness of the routing tables produced by IMMUNIZE. First we verify that the table *call* is correctly established for T. Then, we establish that the tables *pref* and *alt* are completely determined by IMMUNIZE. A simple calling protocol is introduced. Finally we verify that, as long as line failures are isolated, message transfers are completed by the protocol using *pref* and *alt*.

Lemma 3.4. The routing table *call* for the input tree T is correct as determined by IMMUNIZE.

Proof (by induction on number of sites). Only procedure InitialTables and RouteTree affect entries in *call*. If $|V| = 2$, then these procedures correctly establish entries of *call*. Assume that these entries are correctly established for all trees with $|V| < n$ and consider an input tree T with n sites. In a recursive numbering of T, site n is a leaf of T and thus is correctly routed in RouteTree as all corresponding nondiagonal entries of *call* are set to father$[n]$. Processing of site n does not affect entries call$[i,j]$, for $1 \leqslant i, j < n$. Thus, during subsequent processing, sites 1 through $n - 1$ are mutually correctly routed by our inductive hypothesis. It remains to show that these sites have correct entries in their routing tables for n as message receiver. When site n is considered in RouteTree, the corresponding entry in *call* for father$[n]$ is set to n (as InitialTables sets call$[n, n]$ to n). Accordingly, for each site i on the unique path from n to the root site 1, the corresponding entry in the routing table is established as follows. When such i is routed, call$[i, n] \neq 0$ and call[father$[i]$, n] is set to i. When any site j not on this path is routed, call$[j, n] = 0$ and call$[j, n]$ becomes father$[j]$, which is correct. ∎

Using table *call*, algorithm IMMUNIZE establishes routing tables *pref* and *alt* for G. Entries in these tables are set when a new line is added to G. Entries are made for sites of the new triangle created by the added line.

Lemma 3.5. The routing tables *pref* and *alt* are completely determined by IMMUNIZE.

Proof. Consider a site v of a newly formed triangle, and as message receivers, the other two sites of the triangle and all sites separated from v by those two sites. Inspection of procedure RouteSites indicates that the corresponding entries in *pref* and *alt* are made correctly. In Theorem 3.2, we established that every line of T becomes part of a triangle. Therefore, every site of G has all entries in pref and alt determined in IMMUNIZE. ∎

The routing tables *pref* and *alt* are used to complete all message transfers between sites as long as line failures are isolated. The following calling protocol P, written in terms of *pref* and *alt*, will ensure this desired performance.

Protocol P. Assume that a site S has to forward a message with site R as message receiver:

(i) If $S = R$, the site S accepts the message;
(ii) Otherwise, site S transmits the message to site $S' = \text{pref}[S, R]$;
(iii) If the call in (ii) fails, site S transmits the message to site $S'' = \text{alt}[S, R]$.

Lemma 3.6. Protocol P completes all message transfers in G as long as line failures are isolated.

Proof. Consider a message transfer from site S to site R as receiver. Let $S = s_0, \ldots, s_k = R$ be the sequence of sites on the unique path between S and R in the input tree T. Our proof will be by induction on the length k of the path. Suppose $k = 1$. Then s_0 and s_1 are sites in a triangle of G, as the line between them is part of a triangle. Site s_0 will try to call s_1 with s_1 as message receiver (pref $[s_0, s_1] = s_1$ in such a situation). If the call fails, s_0 will call the other site t in the triangle, which will forward the message to s_1. At most one of these calls will fail if line failures are isolated. Now assume that all transfers succeed for paths of length less than $k > 1$. Two cases must be considered. Either s_0, s_1, and s_2 form a triangle in G or s_0, s_1, and a site t not on the path form a triangle. In the first case, the preferred call from s_0 will be to s_2, while in the second it will be s_1. In either case, after at most two calls the message will reach at least site s_1. By our inductive assumption, the message transfer to s_k will succeed after a series of calls. ∎

From the above proof we observe that not only does IMMUNIZE add lines to make the input tree into an ILFI network, but it also places entries in the routing tables to take advantage of shorter routes created by the new lines.

Algorithm IMMUNIZE is only applicable to tree networks with an odd number of sites. This simplifies its definition and verification. A tree network with an even number of sites can be dealt with as follows. At first, simply ignore a leaf site of the input tree, allowing IMMUNIZE to process the resultant subtree having an odd number of sites. Then, add a line connecting the ignored site to the father site of its father in the input tree. The ignored site "inherits" the routing information from the father of its father in T, with some obvious local modifications. Entries in *pref* and *alt* for all other sites with the ignored site as message receiver are identical with those of its grandfather. The resulting graph G is not a triangle cactus, as those always have an odd number of sites. Nevertheless, G is a minimum ILFI network. Thus, IMMUNIZE is easily extended to correctly process an arbitrary input tree.

IV. CONCLUSION

In this article we have presented several classes of networks which are immune to isolated line failures. This included a constructive definition of a class of ILFI networks requiring the fewest lines for a given number of sites. We then describe an algorithm which transforms an arbitrary tree into one of these minimum ILFI networks. The algorithm also provides routing tables, which, in conjunction with a simple calling protocol, are capable of completing all message transfers as long as line failures are isolated. Further research will consider generalizations of the problem presented here, including the description of networks immune to sets of line failures having a bounded number of nonisolated pairs of failures.

This research was supported in part by a grant from the National Science Foundation #ENG-79-02960.

References

[1] C. Berge, *Graphs and Hypergraphs*, 2nd ed., North-Holland, New York (1976).
[2] A. M. Farley, Networks immune to isolated failures. *Networks* 11 (1981) 255–268.
[3] A. M. Farley and A. Proskurowski, Extremal graphs with no disconnecting independent vertex set or matching (UO-CIS-TR-80-15).
[4] D. E. Knuth, *Art of Computer Programming,* Vol. I, 2nd ed., Addison-Wesley, Reading, MA (1973).
[5] P. J. Slater, E. J. Cockayne, and S. T. Hedetniemi, Information dissemination in Trees. *SIAM J. Comput.* 10(4) (1981) 692–701.

Received October 10, 1980
Accepted October 26, 1981