CS-TR-77-3

Minimal Time "Line Broadcast" Networks

Arthur M. Farley
Computer Science Dept.
University of Oregon
Eugene, OR 97403

ABSTRACT

Broadcast refers to the process of message dissemination in a communication network. It is shown that line broadcast can be completed in any connected network from any member in minimal time. An algorithm is present which produces a calling schedule completing broadcast in minimal time from any member of any tree. A discussion of broadcast completes the paper.

ACKNOWLEDGEMENT

I wish to acknowledge the contributions of Drs. S. T. Hedetniemi and A. J. Proskurowski in producing the results reported here.

Broadcast 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. We model a communication network with a graph $G=(V,E)$ consisting of a set V of vertices, or *members*, and a set E of edges, or *communication lines*. We assume that all lines have equal cost (length). A member can transmit a message to another member by making a *call*. We assume that each call requires one unit of time, and that a member can make only one call during any time unit.

Restrictions as to what calls can be placed during a given time unit differentiate three types of broadcasting. In "local broadcasting" an informed member can only call an adjacent vertex. In "path broadcasting" an informed member can call any other member in the network, provided there is an *open path* between the two members. An open path is a path such that every edge vertex of the path is not involved in a call. In "line broadcasting" a member can call any other member, provided there is an *open line* between the members. An open line is a path such that every edge of the path is not involved in a call.

The minimum number of time units necessary to complete broadcast in a network of n members is $\lceil \log_2 n \rceil$ [1]. A *minimal time broadcast network* is a communication network such that broadcast can be completed in the minimum number of time units regardless of message originator. Each type of broadcasting imposes its own constraints upon the architecture of associated minimal time broadcast networks. In the case of "line broadcasting" we find the constraints are the least possible.

after time unit t, m mebers are to be informed, it is obvious that at most $\lfloor m/2 \rfloor$ calls can be made during time unit t. MLB determines a set of $\lfloor m/2 \rfloor$ calls which do not involve any line-use conflicts. These calls are scheduled between closest elements of the set of m to-be-informed members of the network. By scheduling calls between such members, all line-use conflicts are avoided. Thus, $m - \lfloor m/2 \rfloor$ members are to be informed after time unit t-1. The algorithm successively considers each preceding time unit, through time unit 1.

The semantics of two control structures used in the formal algorithm presentation deserve explanation. The first has the form -

```
REPEAT  k  TIMES
   :
   :  body
   :
TAEPER
```

The statements of the body are cyclically executed k times. The other control structure has the form -

```
CASE

  C1:( condition 1):
     :
     : body 1
  C2:( condition 2):
     :
     : body 2

  Cn:( condition n ):
     :
     : body n

ESAC
```

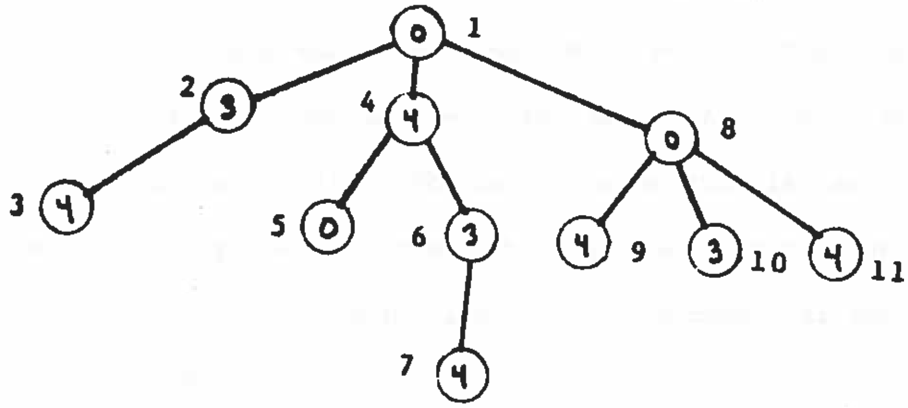
The conditions are evaluated in the natural order until one is found to be true, at which point the associated body is executed. At most one body is executed. If no condition is true, no body is executed. One last definition is also necessary. The remote member of a leaf member of a tree is the one member to which the leaf is connected.

```
STEP #
3.3.2          CASE
3.3.2.1           C1:(time[lf]=0 and mark[lf]=0 and mark[r]=0):
3.3.2.1.1            set mark[r] to lf
3.3.2.2           C2:(time[lf]=0 and mark[lf]=0 and mark[r]=mo):
3.3.2.2.1            make-call(mark[r],lf,unit)
3.3.2.2.2            set mark[r] to 0
3.3.2.3           C3:(time[lf]=0 and mark[lf]=0 and mark[r]>0):
3.3.2.3.1            make-call(lf,mark[r],unit)
3.3.2.3.2            set mark[r] to 0
3.3.2.4           C4:(time[lf]=0 and mark[lf]=mo):
3.3.2.4.1            make-call(mark[lf],lf,unit)
3.3.2.5           C5:(time[lf]=0 and mark[lf]>0):
3.3.2.5.1            make-call(lf,mark[lf],unit)
3.3.2.6           C6:(time[lf]>0 and mark[lf]>0 and mark[r]=0:
3.3.2.6.1            set mark[r] to mark[lf]
3.3.2.7           C7:(time[lf]>0 and mark[lf]>0 and mark[r]=mo):
3.3.2.7.1            make-call(mark[r],mark[lf],unit)
3.3.2.7.2            set mark[r] to 0
3.3.2.8           C8:(time[lf]>0 and mark[lf]>0 and mark[r]>0):
3.3.2.8.1            make-call(mark[lf],mark[r],unit)
3.3.2.8.2            set mark[r] to 0
               ESAC
            TAEPER
3.4         prune the last member lf from tr
3.5         CASE
3.5.1          C1:(time[lf]=0 and mark[lf]=mo):
3.5.1.1           make-call(mark[lf],lf,unit)
3.5.2          C2:(time[lf]=0 and mark[lf]>0):
3.5.2.1           make-call(lf,mark[lf],unit)
           ESAC
3.6        set unit to unit - 1
        TAEPER
     END
```

Where make-call is a function defined as follows:

```
STEP #
0      make-call(sender,receiver,time-unit)
1          output("member "sender "calls member "receiver "during time unit" time-unit)
2          set time(receiver) to unit
   end
```

Figure 3

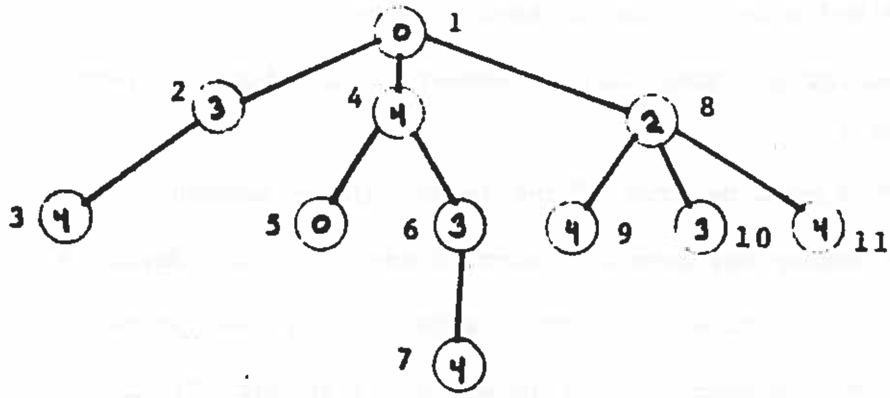


Member 8 calls member 10 during time unit 3.

Member 5 calls member 6 during time unit 3.

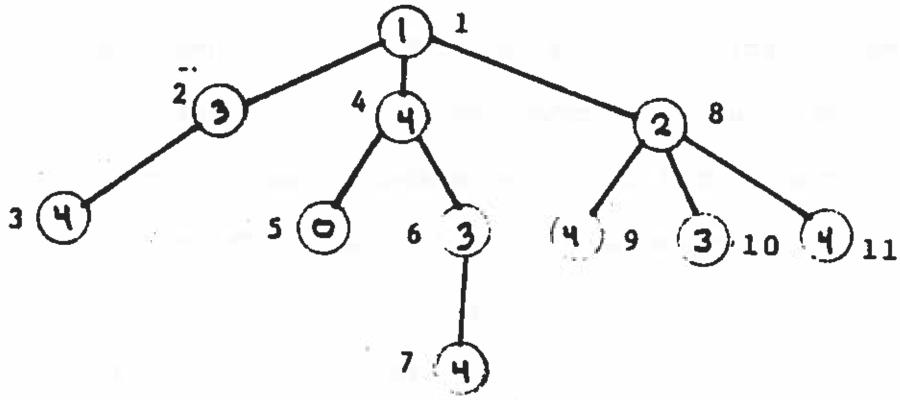
Member 1 calls member 2 during time unit 3.

Figure 4



Member 5 calls member 8 during time unit 2.

Figure 5



Member 5 calls member 1 during time unit 1.

this point, eliminating the member in question from further consideration during that time unit. ||

Lemma 2: *During any time unit, the scheduled calls do not create a line-use conflict.*

Proof: *During scheduling activity for any time unit, MLB traverses each line of the tree only once. This transversal can be said to occur when the first of a line's two end members is pruned from tr. At most one member's label (number) is passed over the line (to mark the other end element) as a possible participant in a call during that time unit. This occurs in cases C1, C2, C3, C6 , C7, and C8 of Step 3.3.2. In all other cases, no member label is passed over the line at all. In such cases, the line will be inactive during that time unit.* ||

Lemma 3: *Assuming that m members are to be informed after time unit t, MBL schedules exactly $\lfloor m/2 \rfloor$ calls during time unit t.*

Proof: *We shall first determine that the number of calls is less than or equal to $\lfloor M/2 \rfloor$. When beginning to schedule calls for time unit t, it is those members j such that time[j]=0 which are to be informed after time unit t. During any time unit, if member i is a participant in a call, time[i]=0 as scheduling begins for that time unit. This can be verified by inspection of MLB. Only case C1 of Step 3.3.2 can generate a new mark, that mark being a member lf such that time [lf]=0. An inspection of all cases which generate a call indicates that only a mark or a member lf such that time[lf]=0 can become a participant in a call. Since each such member participates in at most one call (Lemma 1) and each call requires two participants, at most $\lfloor m/2 \rfloor$ calls are scheduled.*

It remains to prove that the number of calls scheduled cannot be less than $\lfloor m/2 \rfloor$. The proof shall be by contradiction. Suppose that fewer than $\lfloor m/2 \rfloor$ calls are scheduled. If this is so, there must be at least two members i and j such that time[i]=0, time[j]=0, and both members are not scheduled participants in a call during time unit t (after executing Step 3.5 for unit=t). Consider the path between j and i, represented as ($j\ n_1\ n_2 \ldots n_m \ldots n_{k-1}\ n_k\ i$).

Since there cannot be two such members after Step 3.5 is executed for time unit t, there cannot be less than $\lfloor m/2 \rfloor$ calls scheduled. Therefore, exactly $\lfloor m/2 \rfloor$ calls are scheduled by MLB during time unit t. ||

Lemma 4: *Given integer n and the sequence defined as:*

$$a_0=n,\ a_i=a_{i-1} - \lfloor a_{i-1}/2 \rfloor\ ,\quad i=1,2,...$$

If k is the least integer such that $a_k=1$, then $k=\lceil \log_2 n \rceil$ for all $n>0$.

Proof: *Obviously, k cannot be less than $\lceil \log_2 n \rceil$ since $a_0=n$ and $a_i \geq a_{i-1}/2$ (as $a_i = \lceil a_{i-1}/2 \rceil$).*

Thus, it remains to prove that k cannot be greater than $\lceil \log_2 n \rceil$. The integer k would take on a maximal value when the decreases are minimal. Thus, suppose the decreases are minimal. In other words, let

$$a_i=a_{i-1} + 1/2 \quad (\text{or } a_{i-1} = 2a_i-1)$$

Let us consider the sequence in reverse, letting $b_j = a_{k-i}$. Thus, $b_0=1$, $b_1 = 2$, and $b_j = 2b_{j-1}-1$, for $j\geq 2$. This latter recurrence relation can be solved, yielding as a result: $b_j = 2^{j-1} + 1$, for $j\geq 1$. Now, suppose that k is greater than $\lceil \log_2 n \rceil$. Then, $k\geq \lceil \log_2 n \rceil +1$. If $k = \lceil \log_2 n \rceil + 1$ then $b_k = n + 1$. This is the least b_k can ever be made to be due to previous assumption as to the minimal size of the increment (decrement in the a_i sequence). But, $b_k = a_0 = n$, which is a contradiction. Therefore, k cannot be greater than $\lceil \log_2 n \rceil$. ||

Lemma 5: *If member i is scheduled to make a call during time unit t, member i is either the message originator or receives a call during time unit j, $1\leq j<t$.*

Proof: *In legal broadcasting schedules, any member making a call must be informed of the message.*

In case C2 or C7 of Step 3.3.2, mark[r] makes the call and mark[r] is the message originator. In case C4 of Step 3.3.2 or case C1 of Step 3.5, mark[lf] makes the call and is the message originator. In those other cases where calls

Further Conclusions

The result for minimal time "line broadcasting" networks is in contrast to the following result concerning minimal time "local broadcasting" networks.

Theorem: *No tree of more than three members is a minimal time "local broadcast" network.*

Proof: *Any tree which has more than three members has at least two members with degree equal to one. Consider message broadcast when such a member is the message originator. During time unit 1, that member calls his one adjacent member. Since the message originator can place no further calls, the situation is now equivalent to message broadcast originated by that adjacent member in a network of n-1 members. If n is not equal to 2^k+1 for some k, the overall message broadcast can, at best, be completed in $\lceil \log_2 n \rceil +1$ time units, which is not minimal.*

If n does equal 2^k+1, it is possible that broadcast from a member of degree one can be completed in minimal time. That member could call its one adjacent member, from which broadcast could be completed in $\lceil \log_2 n \rceil -1$ time units. However, if broadcast is possible in minimal time from that adjacent member, each member informed during a later time unit must be able to place a call during each successive time unit. Consider the member called by the adjacent member during the next to the last time unit of broadcasting. That member must call a new member during the last time unit. Broadcast originated by that new member can not be completed in $\lceil \log_2 n \rceil$ time units. During time unit 1, it can call its only adjacent member. During the next time unit that member can call its only other adjacent member. A broadcast throughout n-2 members cannot be completed in $\lceil \log_2 n \rceil -2$ time units for $n>3$. ||

Minimal time "line broadcasting" networks require strictly fewer lines than minimal time "local broadcasting" networks for networks of more than three members. Several recent reports provide a closer look at the number of lines required in minimal time "local broadcasting" networks [1,2].

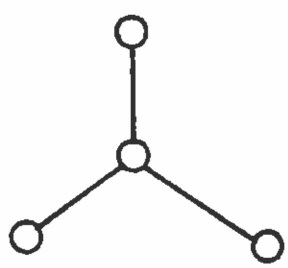
message originator call the most distant local broadcast center of the tree. A broadcast center [4] is defined to be the subset of members of the tree from whom "local broadcast" can be completed in minimal time. This distance can be at most $\lceil \log_2 n \rceil$. That center can complete the broadcast for its "half" of the tree in $\lceil \log_2 n \rceil - 1$ time units employing only local calls, each of length one. Eliminate that broadcast center and its "half" of the tree and repeat the same process. This time the distance to the center can be at most $\lceil \log_2 n \rceil - 1$. If one continues this process, the maximum broadcast cost is bounded by:

$$\underbrace{\sum_{1}^{\lceil \log_2 n \rceil} i}_{\text{cost from message originator}} + \underbrace{n-1-\lceil \log_2 n \rceil}_{\text{cost from others}} = n + \frac{(\lceil \log_2 n \rceil - 2)(\lceil \log_2 n \rceil + 1)}{2}.$$

||

Characteristics of minimal time "path broadcasting" networks lie between the extremes established for the local and line broadcasting cases. Every tree is not a minimal time "path broadcasting" network, as evidenced by the tree network of Figure 6. In order that broadcasting be completed throughout four members in minimal time, two calls must be made during the second and final time unit. Since the central vertex is an element of every non-null path in the network, two calls can never be placed during a given time unit.

Figure 6.



REFERENCES

1. Farley, A.M., "Minimal broadcast networks", submitted to Networks, July, 1977.

2. Farley,A., Hedetniemi,S., Mitchell,S., and Proskurowsi,A., "Minimum broadcast graphs", Univ. of Oregon, Computer Science Report, 1977.

3. Proskurowsi,A.J., "Minimum broadcast trees", in preparation.

4. Slater,P.J., Hedetniemi,S.T., and Cockayne, E.J., "The gossip center of a tree", submitted.