On Broadcasting in Radio Networks—Problem Analysis and Protocol Design

IMRICH CHLAMTAC AND SHAY KUTTEN

Abstract—**In this paper we develop a graph-oriented model for dealing with broadcasting in radio networks. Using this model, optimality in broadcasting protocols is defined, and it is shown that the problem of finding an optimal protocol is NP-hard. A polynomial time algorithm is proposed under which a channel is assigned to nodes from global, multiple-source broadcasting considerations. In particular, nodes participating in the broadcast do not interfere with each other's transmissions, but otherwise simultaneous channel reuse is permitted. Protocol implementations of this approach by frequency division and by time division are given. It is shown that, using these protocols, bounded delay for broadcasted messages can be guaranteed.**

I. Introduction

Broadcasting a message to all network nodes is an important activity in computer networks [2], [10], [11]. Broadcasting protocols are typically found at the network layer of the protocol architectures [17]. The use of broadcasting in radio networks is further motivated by the broadcasting nature of the medium.

Existing broadcasting protocols in multihop radio networks are mostly implementations of solutions that worked well in point-to-point networks. While, in the later case, reliable data link access mechanisms are assumed [5], [12], [17], in multihop radio the situation is drastically different, since in most cases actions taken by the broadcasting protocol will influence the success of transmissions made by the data link protocol.

To demonstrate these observations, let us consider a sample network. In Fig. 1 the edges represent the ability of two connected nodes to hear each other's transmission, subject to the absence of collisions. Using shortest path routing, node v initiates a broadcast and eventually a collision in d will occur. Alternatively, a routing policy which prevents collisions can be obtained by using node b to relay the message to nodes b' and d, after which node d relays the message to nodes c' and e. Although shortest routes are not used here, this policy is better in terms of the total number of transmissions and the time needed to accomplish the broadcast.

This simple example demonstrates that in radio networks, the activities at the network layer and the data link layer are mutually strongly dependent. Consequently, we propose construction of broadcasting algorithms derived from the following global view provided by the broadcasting activity: 1) at the data link layer the transmitting node is not aware of the intentions of others; 2) at the routing layer a node can "anticipate" the transmission intentions of nodes on a given path; 3) ultimately, at the broadcasting layer, the transmission intentions of all network nodes are known. The example of Fig. 1 demonstrates how this information can be used to avoid collisions "enforced" when shortest path routing is used.

Paper approved by the Editor for Computer Communication of the IEEE Communications Society. Manuscript received December 14, 1983; revised March 8, 1985.

The authors are with the Department of Computer Science, Technion—Israel Institute of Technology, Haifa 32000, Israel.

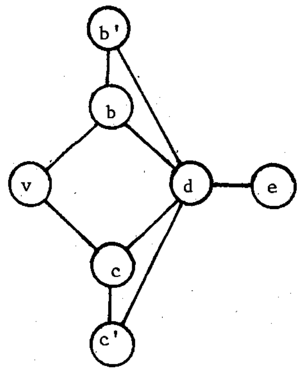

Fig. 1. Alternative broadcast routing procedures.

In the following sections we investigate the properties of broadcasting algorithms in multihop radio networks. We then derive broadcasting and routing protocols based on collision-free channel allocation with spatial reuse. By providing a deterministic allocation approach, the proposed protocols can guarantee a finite upper bound on the time needed for broadcast and ensure that passive acknowledgments are always received with no collisions. Bounded delay is not only critical to most real-time applications, such as voice transmission, but is also beneficial to other network functions, e.g., flow control [17].

Finally, we observe that under the presented radio network model, the problems and algorithms for achieving time division or frequency division channel assignments with spatial reuse can be uniformly treated.

II. Existing Solutions

Radio networks are often constructed in a hierarchical manner and several existing broadcasting algorithms are based on hierarchical routing and flooding [17]. Collisions created by the random access data link protocol are resolved by using acknowledgments and retransmissions [6], [7], [9], [13].

1) A Protocol for Simple Flooding: Each node which accepts a message, relays it to all its neighbors. To prevent nodes from relaying the duplicated message again and again, a hop counter is carried in each message and decremented by every receiving node. When its value becomes zero, the copy of the message is destroyed.

2) A Hierarchical End-to-End Routing and Broadcasting Protocol: In this protocol the information flows over a tree created at network initialization. For routing, a message from the source node is relayed up the tree, and then down to its destination. In graph terminology, each node knows its father, while the father recognizes the son by the label carried in the (son's) message. Broadcast initiated by the head of the hierarchy (root) can be performed easily by having every node send the message to all its sons in the tree. Broadcasts initiated by nodes other than the root may be routed via the head of the hierarchy and be broadcasted as before.

3) A Directional Flooding Protocol: In this protocol, every node must know its distance, in hops, from every other

0090-6778/85/1200-1240$01.00 © 1985 IEEE

node. Contrary to 1), an intermediate node u will repeat a message received from v only if u is relatively closer to the message's destination than v is. Notice that although this solution was originally proposed for end-to-end routing [6], one can suggest a variation to obtain a more efficient broadcast protocol, as follows: if a broadcast was initiated at node w, then node u, which received the message from v, will repeat it only if u is more distant from w than v is.

The above protocols differ in the amount of global information they use, the number of retransmissions they invoke, and the space and bandwidth they consume. While, with the simple flooding protocol, a node must only know the identities of its neighbors and maintain counters, this protocol will generally cause a very large number of retransmissions. At the other end of the scale, the hierarchical protocol requires additional memory for a spanning tree but the number of retransmissions and the number of duplicates will be smaller. Consequently, a tradeoff exists in their efficiency and use of resources.

In general, since in these protocols any transmission can result in collision, no time bound on message delivery can be given. Second, the broadcasting property cannot be utilized for passive acknowledgments, since a node is not guaranteed to receive the message when forwarded by its successor. Thus, a need for acknowledgments from all neighbors remains. Since these acknowledgments cannot be accepted simultaneously, the transmission process becomes similar to sending copies to all the neighbors on different and unreliable (collision prone) lines. Further performance degradation can thus result [1].

III. Medium and Model Description

We deal with multihop connected radio networks in which a node wishing to broadcast a message can use time and bandwidth, considered to be the shared network resources. Resource sharing is accomplished by allocation of basic units we term *phases*. E.g., if time division is used, phases correspond to time slots; using frequency division, a phase corresponds to a band. Without defining exactly at this point the way phases can be constructed, we make the following model-related assumption which must always hold: transmissions in different phases will not collide. The collision of transmissions is defined below.

We assume that not all nodes are in line of sight and in range, but the network is *source connected,* i.e., there is a possibility to pass a broadcast from the originating source node to all nodes in the network.

For formal representation of the network, we define a directed graph $G(V, E)$ where the vertices in V are the network nodes, and an undirected edge $e \in E$ is interpreted as two antiparallel directed edges. Each directed edge $u \rightarrow v$ corresponds to the following $\rightarrow$ relation between u and v: v receives every signal transmitted by u. In other words, $u \rightarrow v$ denotes that v is in line of sight and within range of u.

To ensure that v not only receives but also *correctly* interprets a transmission initiated by u, the following conditions must *also* be satisfied.

Condition 3.1: v does not transmit in the same phase u does.

Definition: A neighbor w of v is an *incoming neighbor* if there is an edge directed from w to v.

Condition 3.2: u is the only incoming neighbor of v transmitting in the same phase. Clearly, by definition, Conditions 3.1 and 3.2 are necessary and sufficient for collision-free reception of a message transmitted to v by an incoming neighbor u.

If Condition 3.1 does not hold, we say that u's transmission collided in v. In Condition 3.2 does not hold, we say that the transmissions of the incoming neighbors u and w have collided

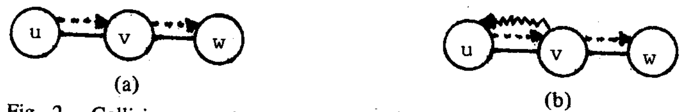

Fig. 2. Collision occurring in v in violation of Condition 3.1. (a) Intended message transmission. (b) Actual signal propagation.

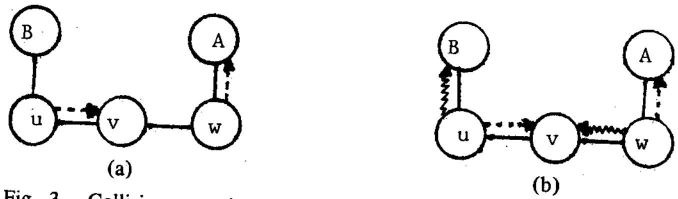

Fig. 3. Collision occurring in v in violation of Condition 3.2. (a) Intended message transmission. (b) Actual signal propagation.

in v. For subsequent use, we note that if a collision in v occurs due to violation of Condition 3.2, then only v is aware of the collision without knowing the origin of the colliding transmissions. When a collision in v occurs due to violation of Condition 3.1, then neither v nor u is aware of the collision. These conditions are demonstrated in Figs. 2 and 3, denoting by $\rightarrow$ the desired transmission of the message (signal propagation) by the node appearing at the tail of the arrow, and denoting by $\rightsquigarrow$ the (not desired) signal propagation resulting from the broadcast property of the radio.

IV. Constructing Optimal Broadcasting Protocols

We define an *optimal* protocol to be one which minimizes one or more of the following measures:

D.max—the maximum time needed for the broadcasted message to reach all nodes;

D.avg—the average time, over all the nodes, needed for the broadcasted message to arrive at each node.

In the following theorems we show that the complexity of constructing optimal protocols is too high for practical considerations if, as commonly postulated, there are nondeterministic polynomial (NP) problems whose complexity is larger than any deterministic polynomial (P) problem [4]. For reasons of clarity and without loss of generality, we shall refer below to the time division method, such that each phase corresponds to a time slot.

Definitions: We say that the source node s is *covered* (by the broadcast) at the broadcast initiation, and node v becomes *covered* after a covered neighbor of it has transmitted, without collision occurring in v. We say that "the broadcast has covered the network" when all nodes are covered.

What is the minimum number of time slots required for a broadcast originated in s to cover the network?

Theorem 4.1: The problem of D.max minimization is NP-hard.

What is the lowest average number of time slots needed for a broadcast to cover the network?

Theorem 4.2: The problem of D.avg minimization is NP-hard.

For proofs of Theorems 4.1 and 4.2, see the Appendix.

V. Investigation of Candidate Heuristics

Since the construction of optimal algorithms is not practical, we resort to approximate solutions. We observe that the high complexity of exact solutions arises from the fact that actions taken by each node are a function of the actions taken by all other nodes in the network. Consequently, we examine heuristics which simplify the relations between successive nodes' actions regarding transmission and reception of the broadcasted message.

A. Heuristics for Reducing the Delay

Notice that a broadcast is an iterative process starting at the source, in which in every iteration (time slot), some of the nodes which have already received the message transmit it to those which have not received it yet. For each broadcast, an order between nodes can be defined according to the order in which they receive the broadcasted message. Since establishing the optimal order in all iterations is too complex, we look at a heuristic in which the propagation of the broadcast in different iterations is optimized independently.

Heuristic H1: Perform the broadcast so that in every slot (given the set of nodes already covered), the number of nodes which *receive* the message for the first time is maximized. (Notice that Heuristic H1 resembles the "hot potato" forwarding broadcast policy used in point-to-point networks [2].)

Theorem 5.1: The problem of maximizing the number of nodes which receive the message in the next time slot is NP-hard.

For the proof of Theorem 5.1, see the Appendix.

The next approach attempts to simplify the "complementing" order, the order in which the nodes *transmit,* by limiting the number of transmissions performed by each node. (Notice that if frequency division is used, i.e., each phase corresponds to a frequency band, then the following heuristic limits the number of transmitters required at each node to one.)

Heuristic H2: Perform the broadcast minimizing $D.\max$ under the restriction that each node is permitted to transmit only once.

Theorem 5.2: The problem of minimizing $D.\max$ under the restriction that each node may transmit only once is NP-hard.

For the proof of Theorem 5.2, see the Appendix.

VI. Network Division into Mutually Noninterfering Areas: Complexity of Optimal Solution and a Polynomial Heuristic

To simplify complex dependency relations, we attempt to partition the network into areas, not necessarily disjoint, such that actions undertaken by nodes outside a given area can be ignored. A good division into areas should then enable nodes to perform simultaneous transmissions without collisions, and reduce complexity by limiting the number of nodes and links to be considered in the partitioning process.

A. Optimal Network Partitioning

Given the iterative view of the broadcast process given at the beginning of Section V-A, we define:

Belt_i—the set of nodes which transmit in the same iteration, i;

CLIENTS (Belt_i)—the set of nodes which receive the broadcasted message from, and only from, a node in Belt_i prior to their own transmission.

Fig. 4 shows a simple example of a network division into belts. Notice that nodes u and v have two common neighbors (s and w) and still can be in the same belt. Notice also that Belt_2 and Belt_5 may transmit in the same phase (thus, e.g., they may transmit simultaneously consecutive packets of a broadcasted message or broadcast two different messages).

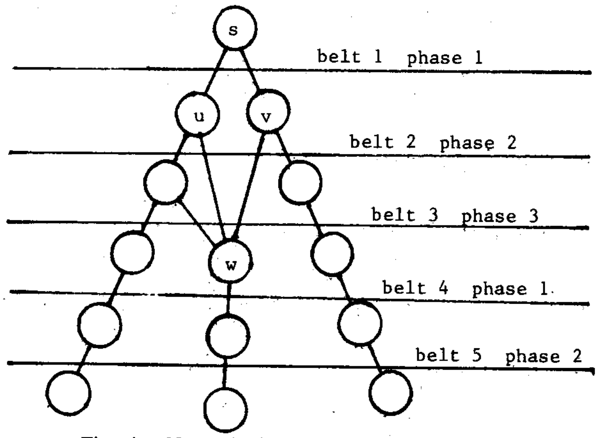

Fig. 4. Network division into belts and phases.

Question: Can a given network be divided into belts in such a way that the number of required phases is minimized?

Theorem 6.1: The problem of dividing the network into belts minimizing the number of phases required is NP-hard.

For the proof of Theorem 6.1, see the Appendix.

B. Solution by a Polynomial Algorithm

We trace the high complexity of the last problem to the fact that too many subsets of nodes are candidates for belts. It is well known that the total number of subsets of a given set is exponential in the number of its members (2^n). We thus look for a polynomial heuristic which limits the number of candidate belts. This can be accomplished by building the belts incrementally over a broadcast spanning tree rooted at the source. The proposed algorithm builds belts by observing each time the area which includes a node, its neighbors, and their neighbors, thus constructing this tree incrementally.

A Greedy Heuristic for Generating a Collision-Free Broadcast Spanning Tree in Polynomial Time: This algorithm constructs a broadcast spanning tree and assigns transmission phases to all nodes in such a way that no node's transmission will collide in its father or sons in the tree. The tree construction can start at any node wishing to broadcast a message; however, the constructed tree can subsequently be used for broadcast from any other node.

Definitions:

The root—any given node in the network (the algorithm constructs the tree starting with the root).

K—the maximum number of neighbors any node in the given graph has.

A phased node—a node (graph vertex) which has already been assigned a transmission phase number. (Phase numbers are taken from the set of positive integers.)

1) A node v *hears* (or is hearing) *from* node u *in phase* i iff

a) u is v's only neighbor which has been assigned phase number i and

b) phase i has not been assigned to v.

2) A node other than the root *hears* iff it has at least one phase for which 1) holds. (The root always *hears* by definition.)

3) Father → son relation—u is defined as the father of all nodes which hear (or "start hearing") by the virtue of the phasing of u. (Notice that the father → son relation is computed by the algorithm, rather than being a static property of the graph.)

1) The Phase Assignment Algorithm:

1) Assign phase number 1 to the root.

While

there is a node v which hears but is not phased

do begin

2) Let v be the first node to become hearing among the unphased nodes (i.e., the first node to have a father by definition 3) among the unphased nodes;

3) Assign to v the lowest sequential phase number i which meets the following conditions:

c.1) i is not the phase number of v's father

c.2) i has not been assigned to another neighbor of v's father

c.3) i has not been assigned to a father of v's neighbor.

end (of while statement)

4) **Stop**

2) Example: To illustrate the phase assignment algorithm, we apply it to a sample network given in Fig. 5. The network

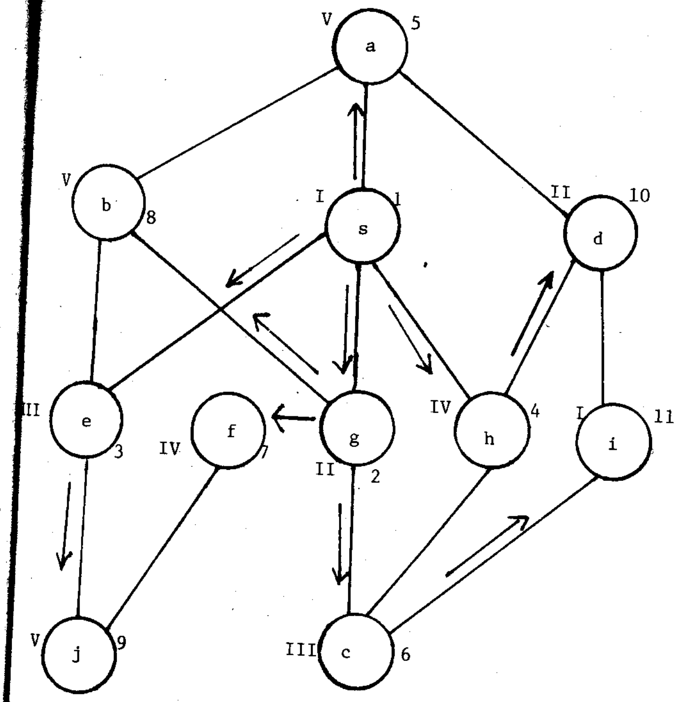

Fig. 5. Phase allocation in an arbitrary network. Roman numerals give the phase numbers assigned by the algorithm. Numbers adjacent to nodes give the order in which phase assignment is performed. Arrows denote the father-son relations.

node identities are denoted by $a, b, c, \cdots, i$ and nodes connected by edges are assumed to be within range and in line of sight, consistent with the model introduced in Section III. The execution of the algorithm for the given network model is as follows.

0) {algorithm initialization}. Node i (the chosen root) is assigned phase number 1. As a consequence, nodes g, e, h, and a become hearing (i.e., they become node s–s sons).

1) {iteration 1 of the WHILE loop}. Nodes g, e, h, and a can each be chosen for the next phase number assignment. Let us assume node g is chosen (i.e., $v = g$ in the algorithm). Condition c.1) prevents the assignment of phase number 1 to node g; therefore, node g will be assigned phase number 2. (Nodes c, f, and b become hearing nodes.)

2) Any one of nodes e, h, and a can be chosen next. Assume node e is chosen. Condition c.1) prevents the assignment of phase number 1 to node e. Both conditions c.2) and c.3) prevent the assignment of phase number 2. Thus, node e is assigned phase number 3. (Node j becomes hearing.)

3) Nodes h and a can now be chosen. Assume node h is chosen. This node cannot be assigned phase number 1 [condition c.1)] or phase number 2 [conditions c.2) and c.3)], or phase number 3 [condition c.2)]. Therefore, node h is assigned slot number 4. (Node d becomes hearing.)

4) At this point the first node to become hearing among the remaining unphased nodes is node a. Node a cannot be assigned phase number 1 [condition c.1)] or phases 2 or 4 [conditions c.2) and c.3)] or phase 3 [condition c.2)] and is thus assigned phase number 5.

5) Nodes c, f, and b can now be chosen. Assume $v = c$. Phase numbers 1 and 2 cannot be assigned to node c [due to conditions c.1) and c.3), respectively]. Thus, node c is assigned phase number 3 and consequently node i becomes hearing (and node c's son).

In a similar manner, nodes f, b, j, d, and i are assigned time slots 4, 5, 5, 2, and 1 in iterations 6–10, respectively. The final time slot allocations as well as the allocation order are shown in Fig. 5.

3) *Algorithm properties:* The following properties hold for the phase assignment algorithm.

a) The algorithm does not assign more than Min $\{2K, K + 2 \cdot SQRT(|E|)\}$ (where $|E|$ defines the number of members in the set E).

b) The algorithm always stops successfully.

c) The father–son relation, defined by the algorithm, constructs a spanning tree such that every broadcast can cover the network by each node in the tree transmitting in its phase.

d) The algorithm is polynomial in execution time.

Proof of Theorem 6.2:

Lemma 1: When there is a hearing node v which is not phased, there always exists a phase with a sequential number lower than Min $\{2K, K + 2 \cdot SQRT(|E|)\}$ which meets the conditions in step 3) and, thus, can be assigned to v.

Proof: There are two subcases:

1) Min $\{2K, K + 2 \cdot SQRT(|E|)\} = 2K$

2) Min $\{2K, K + 2 \cdot SQRT(|E|)\} = K + 2 \cdot SQRT(|E|)$.

1) Let $f(v)$ be v's father. Conditions c.1) and c.3) eliminate no more than $K + 1$ phases. No more than $K - 2$ additional phases are eliminated by condition c.2), since $f(v)$ has at most $K - 2$ neighbors other than $f(v)$'s father and v. Thus, the total number of unassignable phases does not exceed $2K - 1$ and v can be assigned the $2K$th phase.

2) Since the algorithm always assigns the phase with the lowest sequential number possible, the lemma follows if we prove that there is always an assignable phase number out of the first $K + 2 \cdot SQRT(|E|)$ phase numbers. Let us assume (in contradiction to the claim of the lemma) that no phase can be assigned to v. For each phase which cannot be assigned to v, let us consider the reason why. Exactly one phase cannot be assigned to v due to condition c.1). Additional phases cannot be assigned strictly due to condition c.2), i.e., for each one of these phases there exists a neighbor of $f(v)$ which was assigned this phase. We term this neighbor a forbidding neighbor for this phase. Let the set $N(f(v))$ include exactly one forbidding neighbor for each phase which cannot be assigned strictly due to condition c.2).

Notice that so far we have accounted for phases unassignable to v due to conditions c.1) and c.2). Every other phase i is unassignable to v due to condition c.3). Therefore, there must exist at least one node which is a neighbor of v, and which hears only in phase i; otherwise phase i could be assigned to v. We denote one of these nodes as z_i. Let $Z = \bigcup_i z_i$ and let F be the set of nodes from which the nodes in Z hear. Let $P = F \cup N(f(v))$. Notice that F and $N(f(v))$ are disjoint. Recall that except for $f(v)$'s phase [condition c.1)], P now includes exactly one node for each phase which cannot be assigned to v. [The phase of any node in F cannot be assigned to v due to condition c.3), and the phase of any node in $N(f(v))$ cannot be assigned to v due to condition c.2)]. As we have assumed that no phase out of the $K + 2 \cdot SQRT(|E|)$ can be assigned to v, we obtain $|P| = K + 2 \cdot SQRT(|E|) - 1$. Let us number each $p \in P$ with the sequential number of the phase assigned to p. Note that any two nodes in P were assigned a different phase.

Now let us count the edges in the network's graph. For each p_i there are at least $i - K - 1$ edges emerging from p_i to its neighbors; otherwise a phase with a lower sequential number could be assigned to p_i. [c.1) and c.2) are responsible for the elimination of no more than K phases—$f(v)$'s phase and the phases of $f(v)$'s other neighbors except for v]. In addition, there are $|Z|$ edges from v to the members of Z, and $|Z|$ edges of the kind $z_i - f_i$. Since each edge is counted no more than twice, the total number of edges is greater than

$$\frac{1}{2}(1+2+3+\cdots+|Z|)=\frac{1}{2}|Z|\frac{[|Z|+1]}{2}$$

which is greater than $|E|$. This contradicts the assumption that no phase can be assigned to v. Q.E.D.

Lemma 2: Phase assignment to a new node does not cause a hearing node to become nonhearing.

Proof: Immediate from conditions c.1) and c.3).

Lemma 3: The algorithm does not assign a phase (to v) previously assigned to a son of v's neighbor.

Proof: Let u be any father which is a neighbor of v. If the assignment is to a son of u, the lemma follows from condition c.2). Otherwise, v became hearing before any son of u [by step 2)] and thus, when it is assigned a phase, no son of u has previously been phased. Q.E.D.

Lemma 4: When the algorithm stops, all nodes hear and no phase is assigned both to a son and to another neighbor of a given father.

Proof: The lemma follows from Lemmas 2 and 3 and from the following argument. Suppose that there is a node v which does not hear when the algorithm stops. Let N_v be the set of vertices in the maximal connected subgraph which includes v, such that no node in N_v hears. As there is at least one hearing node in the graph (the root), and according to the connectivity assumption (see Section III), there must be a node u in N_v that has a hearing neighbor.

When the while loop is finished, there is no unphased hearing node with a neighbor which does not hear. Thus, u has a phased neighbor. Let w be the first neighbor of u to become phased. On phasing of w, u became a hearing node, and according to Lemma 2, u is still hearing, contrary to the assumption that u is in N_v. Thus, there cannot be a node v which is not hearing at termination. Q.E.D.

Lemma 5: The relation father $\rightarrow$ son between hearing nodes defines a rooted tree. Given that every node transmits only in its phase, no father's transmission will collide in its sons and no son's transmission will collide in its father.

Proof: The lemma follows from Conditions 3.1 and 3.2, from the definition of hearing nodes, from Lemma 4, and from conditions c.1) and c.2).

Proof of Theorem:

a) Immediate from Lemma 1.

b) In every iteration one more node becomes phased. Since the number of nodes is finite, the algorithm must stop.

c) Immediate from Lemmas 4 and 5.

d) In each iteration of the while loop, another node v becomes phased—so that the number of iterations is bounded by $|V|$. In step 3), each of the phases (no more than $2K$) may be examined for interference with phase assignments to the fathers of v's neighbors and the fathers of the neighbors of v's father (both bounded by K). Thus, the complexity of the algorithm is no more than $O(K^2 \cdot |V|)$. Q.E.D.

VII. Broadcast "Spanning" Protocols for Radio Networks

By equating time slots or frequencies with phases, time division and frequency division broadcast protocols can be obtained using the node-phase assignment algorithm given in Section VI. Every node can initiate its own broadcast or forward a broadcasted message by simply transmitting a message in its assigned phase. Broadcasted messages propagate on the broadcast spanning tree created by the phase assignment algorithm in both directions.

The protocol implementation at each node is computationally very simple and places minimal memory requirements on each node. A broadcasted message carries the following information, replaced in each transmission step, in its header: I—the index of the transmitting (forwarding) node, and J—the index of the node from which the message was received by I. A node k will forward the message in its next assigned phase iff $J \neq k$ (to prevent backward radiation) and I is the index of k's neighbor on the spanning tree (to propagate the broadcast on the tree only).

Passive acknowledgments can also be utilized in the protocol, to aid in recovery from nonbroadcast-related transmission errors, as follows. A node k learns of the correct reception of the forwarded message by its neighbors when receiving their transmission of the broadcasted message with $J = k$ (excluding the neighbor from which k has received the message).

A. Time Division Protocols

In the time division version of the protocol, we assume that messages of constant size are transmitted in time slots (phases) and that nodes are synchronized as in [15]. Each slot equals the message transmission time and the maximum propagation delay between neighboring nodes. The collection of the assigned time slots defines a cycle, such that nodes can transmit successively in their own slot in consecutive cycles. Every network node can initiate a broadcast which can be forwarded in any direction on the tree without collisions. The algorithm further guarantees that every node will receive its neighbor's transmission forwarded along the spanning tree. This provides for passive acknowledgments so that the protocol has a mechanism to detect faults due to channel disturbance, node failures, or changes in topology.

With this implementation of the protocol, by giving priority to a broadcasted message (in other words, when assuming that the message is not delayed in intermediate nodes), the protocol guarantees that the maximum delay in the network is bounded by

(message-transmission-time) multiplied by (maximum-path-length-in-the-spanning-tree) multiplied by (cycle-length-in-slots).

It is easy to see that for regular network structures such as linear networks or trees, the protocol thus guarantees optimal delays. For general graphs, a bound is given as above.

B. Frequency Division Protocols

In the frequency division implementation of the algorithm, we divide the total bandwidth into N frequency bands (where N is the total number of phases assigned by the algorithm). Thus, each band corresponds to a phase. The frequency division implementation differs from time division in several significant ways. First, the need for clocks and node synchronization is eliminated so that messages of arbitrary length can be broadcasted. Second, assuming that a node can transmit on a given band while receiving on another, message cut-through switching can be implemented, i.e., a message can be transmitted while being received. In this way store-and-forward delays, and the associated buffer requirements, are reduced. On the other hand, additional transmit/receive hardware may be needed for the frequency division implementation, and the date rates on each band are reduced proportionally to the bandwidth division.

Since the same collision-free properties hold for frequency division as for time division, the total delay involved in broadcasting the message can again be computed. Let D be the length, in hops, of the longest path on the spanning tree generated by the algorithm; T—the message transmission time over the total frequency; a—the maximum propagation delay between any two neighbors on the tree; and b—the node turnaround time (between the reception and the transmission of each bit). The total delay is then given by

$$D \cdot (a+b) + N \cdot T.$$

In the above implementation of phases in frequency division, a number of receivers per node is needed, given by the number of each node's neighbors on the spanning tree. Alternatively, the number of receivers per node can be reduced to only two, by employing the following homing mechanism: a node will sense the incoming signal and adapt itself to the proper band [8]. The penalty of this approach is the possibility to lose all but one son's transmissions at the father when several sons transmit simultaneously. Passive acknowledgments to the sons are, however, still provided.

Lastly, partial neutralizing of the narrowing of the total bandwidth due to frequency division is possible when using frequency-hopping spread spectrum designed to deal with multipath fading [10]. The existing frequency hopping can be used in the protocol by defining each phase as a wave pattern, so that nodes assigned the same phase will use the same key.

VIII. Discussion

The use of classical graph theory to construct and analyze efficient broadcast protocols in radio networks is hampered by the difficulty of introducing the mutual node interference into existing models. Thus, broadcasting over minimum spanning trees or shortest path routes may lead to collisions and reduced performance. Using a graph theory approach, it has been shown that the construction of optimal broadcast protocols, or heuristics previously employed in point-to-point networks, results in NP-hard problems.

From considerations relating to the broadcasting process in radio networks, a polynomial time algorithm has been proposed. Frequency and time division based protocol implementations of this algorithm were shown to have provable properties and bounded delays. The protocols also utilize the broadcasting nature of the radio network in several ways: a single transmission of the broadcasted message by any node is sufficient for it to be received correctly by all its neighbors on the broadcast spanning tree. Second, passive acknowledgments are provided for all transmissions involved in the broadcast. Third, since in a radio network a transmitting node is not free to choose on "which outgoing link or links to place the broadcast," the responsibility of forwarding one copy of each message in the "correct" direction is placed with the receiving nodes.

Several additional observations deserve attention or further research. The number of required phases and, thus, the total delay involved in the broadcast can be reduced by a factor of 2 by using directed broadcasting. In this approach, guaranteed transmissions are made in the direction away from the source, while passive acknowledgments are provided in the other direction [3]. Such implementation may prove attractive for broadcasting and routing in centralized or hierarchical networks [6].

The last observation concerns the use of a single broadcast spanning tree. Clearly, a single tree will not necessarily be equally good for all sources. Similar observations have been made with respect to spanning tree based broadcast algorithms in point-to-point networks, and reasons were given to motivate the use of a single tree [14]. Those reasons concern the dependencies between routes and, thus, are even more strongly applicable to radio networks.

Appendix

Proof of Theorem 4.1

We prove by reducing the 3XC problem (the problem of exact cover by sets of three members) [4] to the following problem.

Input: An undirected graph $G(V, E)$ (representing a network), a vertex s in V (we shall call s the "source"). The network is source connected as defined in Section III.

Question: What is the minimum number of time slots required for a broadcast to cover the network?

We show that the problem remains NPC, even if we only ask whether the minimum number of time slots is less than or equal to a given number of time slots I, $I = 2$ (D.max2 problem). Given an input to the 3XC problem, let us transform

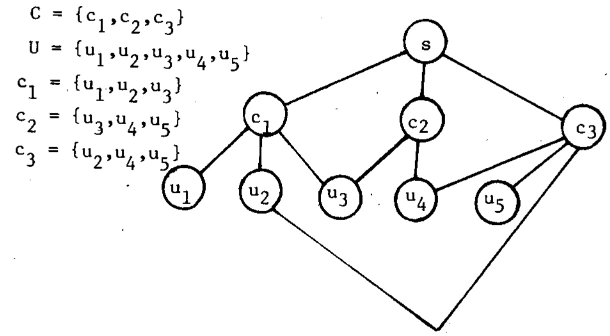

Fig. 6. The reduction from 3XC to the broadcast delay problem (two time slots delay).

it to be a case of the D.max2 problem in the following way. Let $G(V, E)$ be the following graph—V includes the source s, a vertex c for every c in C, and a vertex u for every u in U. E includes edges connecting s to every c in C, and each c in C is connected to every u in c. See Fig. 6.

Given a solution to the original 3XC problem, we can perform a broadcast in G in the following way: in the first time slot s transmits, and all the sets hear, and in the second time slot, all the sets participating in the cover C' transmit. As each member u is a member of exactly one set in the cover, it is obvious that each member hears the broadcast, and no collision occurs. Given a solution to the D.max2 problem, we take the c vertices which have to transmit. Apparently, the union of the sets they represent includes all the u vertices, since the broadcast has to reach all the vertices. Also, every u vertex is a neighbor (son) only to one c vertex, since it can hear only in the second time slot (the first being dedicated to the source), and if two of its neighbors transmit, then a collision occurs.

Q.E.D.

Proof of Theorem 4.2

We use the same construction as for Theorem 4.1. If there is an exact cover, then the average number of time slots is

$$\frac{(2 \cdot |U| + 1 \cdot |C| + 0 \cdot 1)}{|U| + |C| + 1}.$$

If there is no exact cover, the values of the denominator and the second term in the numerator remain as before, but the value of the first term in the numerator increases, according to the number of u's which do not hear in the second time slot.

Q.E.D.

Proof of Theorem 5.1

We prove by reducing an NPC problem to the following "hot-potato" problem.

Input: A graph G, an integer I, a set A of covered vertices.

Question: Can we choose a subset of A, such that if vertices in the subset transmit in the next slot, then more than I vertices will become covered (i.e., will enter the set A)?

In [16] we find the NP completeness proof for the following problem: check the edges of the network (the links), so that a check in one direction is done by having a vertex connected to the edge transmit, and having the vertex on the other end of the edge hear the transmission without collision. Some of the vertices of the network can only receive, and the others can only transmit. The question: is it possible to divide vertices which are able to transmit into two sets—a silent (transmitting in the next time slot) set, and a transmitting set, such that in the next time slot, at least I edges will be checked? It is easy to find a reduction from this problem to the "hot-potato" problem: let us add a source vertex, connected to all those vertices which can only transmit, and only to them.

Clearly, there is a positive answer to the case of the "hot-potato" problem we have defined, if and only if this is an answer to the original question. Q.E.D.

Proof of Theorem 5.2

We prove by reducing 3XC to the following problem.

Input: An undirected graph $G(V, E)$, a vertex s (the source) in V, an integer I. The network is source connected.

Question: Can a broadcast cover the network in less than I slots using each vertex only once for transmission?

A reduction from the problem in Theorem 4.1: given a special case $G = (V, E)$ for the problem of Theorem 4.1, let us construct $G' = (V', E')$ as follows: for every v in V we generate $|V|$ vertices—$v_1, v_2, \cdots, v_{|V|}$. If the edge u-v is in E, then E' includes the edges v_i-u_j for each $1 \leq i, j \leq |V|$. Given a solution to the problem in Theorem 4.1, it is obvious that each v in V is used no more than $|V|$ times, as, due to the source connectivity assumption, we can perform the broadcast in $|V|$ time slots. We transform the given solution to a solution of the new problem in the following way: suppose v, in the given solution, transmits L times ($L = <|V|$), and $v_1, v_2, \cdots, v_L$ in G' transmit, one time each, in the time slots v transmits in G. From the definition of E', it follows that a transmission of each of the v_i's can be received by all the u_j's. Given a solution to the new problem, we can transform it to a solution of the original problem, by letting v transmit, in every time slot in which one of the v_i's in G' transmits. Q.E.D.

Proof of Theorem 6.1

The proof is very similar to the proof of Theorem 4.1. If an exact cover exists, then we can divide the transmitting nodes into two belts (using two phases). The first belt will include only the source node and the second belt will include the nodes corresponding to the sets of the exact cover. If no exact cover exists, then clearly more than two belts and phases will be needed. Q.E.D.

REFERENCES

[1] N. Abramson, "The ALOHA system," in *Computer Communication Networks,* N. Abramson and F. Kuo, Eds. Englewood Cliffs, NJ: Prentice-Hall, 1973, pp. 501–518.

[2] Y. K. Dalal and R. M. Metcalfe, "Reverse path forwarding," *Commun. ACM,* vol. 21, Dec. 1978.

[3] I. Chlamtac and S. Kutten, "On broadcasting and spatial reuse in radio networks—Problem analysis and design of protocols with provable properties," Dep. Comput. Sci., Technion—Israel Inst. Technol., Haifa, Tech. Rep. 273, May 1983.

[4] M. R. Garey and D. S. Johnson, *Computers and Intractibility.* San Francisco, CA: Freeman, 1979.

[5] W. R. Franta and I. Chlamtac, *Local Networks, Motivation, Technology and Performance.* Lexington, MA: Lexington Books, 1981.

[6] I. Gitman, R. M. Van Slyke, and H. Frank, "Routing in packet-switching broadcast radio networks," *IEEE Trans. Commun.,* vol. COM-24, pp. 926–930, Aug. 1976.

[7] I. Gitman, H. Frank, and R. Van Slyke, "Performance evaluation of packet radio systems by simulation—A case study," in *Proc. Winter Simulation Conf.,* Miami Beach, FL, Dec. 1978, pp. 737–748, part ii.

[8] Special Issue on Synchronization, *IEEE Trans. Commun.,* vol. COM-28, Aug. 1980.

[9] R. C. Kunzelman, "Overview of the ARPA packet radio experimental network," in *Compcon Conf. Proc.,* Spring 1978, pp. 157–160.

[10] R. E. Kahn, S. A. Gronemeyer, J. Burchfiel, and R. C. Kunzelman, "Advances in packet radio technology," *Proc. IEEE,* vol. 66, Nov. 1978.

[11] J. K. Peacock, E. Maning, and J. W. Wong, "Synchronization of distributed simulations using broadcast algorithms," *Comput. Networks,* vol. 4, pp. 3–10, Feb. 1980.

[12] A. Segal, "Distributed network protocols," *IEEE Trans. Inform. Theory,* submitted for publication.

[13] N. Shacham and E. J. Craighill, "Dynamic routing for real-time transport in packet radio networks," in *Proc. INFOCOM,* 1982.

[14] D. W. Wall and S. S. Owicki, "Center based broadcasting," Dep. Elec. Eng. Comput. Sci., Stanford Univ., Stanford, CA, Tech. Rep. 189, June 1980.

[15] W. R. Fried, "Principles and simulation of JTIDS relative navigation," *IEEE Trans. Aerosp. Electron. Syst.,* vol. AES-14, Jan. 1978.

[16] S. Even, O. Goldriech, and P. Tong, "On the NP completeness of certain network testing problems," Technion—Israel Inst. Technol., Haifa, Tech. Rep. 230, Dec. 1981; also, *Networks,* to be published.

[17] A. S. Tanenbaum, *Computer Networks.* Englewood Cliffs, NJ: Prentice-Hall, 1981.

★

Imrich Chlamtac, for a photograph and biography, see p. 1075 of the October 1985 issue of this TRANSACTIONS.

★

Shay Kutten received the B.Sc. (cum laude) and M.Sc. degrees from the Technion—Israel Institute of Technology, Haifa, in 1981 and 1984, respectively.

He is currently a Ph.D. student in the Department of Computer Science at the Technion, where he worked as a Teaching Assistant from 1980 to 1984 and is now an Instructor.