ON BROADCASTING AND SPATIAL REUSE
IN RADIO NETWORKS- PROBLEM ANALYSIS AND
DESIGN OF PROTOCOLS WITH PROVABLE PROPERTIES

by

Imre Chlamtac and Shay Kutten

Technical Report #274

May 1983

* This paper is part of the second author’s M.Sc. Thesis, Dept. of
Computer Science, Technion.
ABSTRACT

Broadcasting a message means sending that message to all nodes in a network.

A multihop radio network is a radio network in which not all nodes are in range and in line of sight.

Spatial reuse (SR) means utilizing the fact that a node cannot receive signals from some other nodes, to enable simultaneous transmissions on the same band with no collisions.

Existing solutions for a broadcast in multihop radio networks are implementations of solutions which worked well in point to point (p.t.p) networks. They leave the task of dealing with the differences in the channel type, multiaccess versus p.t.p, to the underlying link layer protocol. Such solutions do not fully take into account the special characteristics of the radio. The absence of coordination between the two protocol layers (the link and the routing) may cause collisions and loss of packets, since the link and the broadcasting or routing protocols perform mutually dependent time calculations, each ignoring the transmission decisions made by the other.

Using a graph theory model we show that the problem of finding an optimal broadcasting protocol (according to various definitions of optimality) is NPH. We also show some natural heuristics and conventional approaches in point to point networks to be NPH when applied to radio networks due to the aforementioned mutual dependencies.

We propose algorithms that provide a guaranteed amount of parallelism by using time division multiaccess with spatial reuse-TDMA/SR and frequency division multiaccess with spatial reuse-FDMA/SR. The algorithms are based on polynomial heuristics which by an iterative approach succeed to reduce the dependencies associated with the global optimal approach. The broadcasting protocols derived from these algorithms guarantee a bounded delay for broadcasted messages and due to the parallelism enabled by spatial reuse, these protocols reduce
the load on the channel and the nodes involved in the broadcast. The TDMA/SR and FDMA/SR algorithms can be further used as datalink protocols for radio networks with hidden nodes. Finally the tree-like construction built for the broadcasting protocols can be used for end-to-end routing.

1. INTRODUCTION

While routing and broadcasting in point-to-point networks received much attention these problems in radio networks have received considerably less treatment.

Broadcasting a message means sending that message to all nodes in the network.

Broadcast is an important activity in computer networks in general [DM78], [PMW79], [KGBK78]. In radio networks its use is further motivated by the broadcasting nature of the radio.

Radio networks are usually constructed in a hierarchical manner, containing nodes of less importance (repeaters) and greater importance (stations) [GVF76], [GYF78]. The hierarchical approach makes it easier to cope with the complications presented by the medium. Consequently existing solutions often use hierarchical approach, and even point to point routing is often implemented by embedding it in a broadcast (see p. 198 in [T81]). Existing solutions for broadcast in multihop radio networks are mostly implementations of solutions which worked well in point to point networks. They leave the task of dealing with the differences in the channel type (multiaccess versus p.t.p.), to the underlying link layer protocol. Such a solution does not fully take into account the special characteristics of the radio, since no effort is made to coordinate the two protocols (the link and the routing) In p.t.p. computer networks, routing and broadcasting protocols assume a reliable transmission mechanism [T81], [S80]. In other words it is assumed that if transmission between two network nodes is requested, the request is honored by the underlying data link protocol, and that basically the
data link protocol may transmit any message with the same success. In multihop radio networks, the situation is drastically different since in this case actions taken by the routing protocol also influence the outcome, or the success of the data link protocol. In most radio networks the data link layer is based on random access protocols which assume that transmission requests arrive at random. If a node’s transmission of a message $M_1$ has collided (with some other message $M_2$ and thus has been lost), it will be rescheduled after a RANDOM time interval. In reality, however, message transmissions at data link layer are requested by higher level protocols, such as routing or broadcasting, so that these protocols influence the successful transmission or loss of messages through collisions and thus influence the time needed to pass them successfully over the channel. Consequently, the efficient use of radio frequencies for passing a message over one or more hops in a multihop radio network can be obtained by global network considerations to provide collision free channel use with spatial reuse.

To demonstrate these observations let us consider the sample network of figure 1, where edges represent the ability of two connected nodes to hear each other’s transmission subject to the absence of collision. Suppose node $N$ initiates a broadcast — i.e. sends a message to all the other nodes in the network. A shortest path, p.t.p. routing policy, would be to let node $B$ relay the message to nodes $B'$ and $D$, and let $C$ relay the message to $C'$ followed by $D$ relaying the message to $E$. However, if $B$ and $C$ transmit at the same time a collision in $D$ will occur. Consequently, it is now up to the data link layer to resolve the collision caused by the higher routing layer protocol, and the maximum message delay in the network, i.e. the time $E$ receives the message, is not the shortest possible. There is however another routing policy which prevents collisions obtained by using only $B$ (and not $C$) in relaying the transmission from $A$, and having $D$ relay the message to both $C'$ and $E$. This policy suggests itself to be a better routing strategy in the radio network in terms of link utilization (# transmissions) and the time till termination of the broadcast.
It should be evident by now, even from this very simple example, that both routing and datalink layers are mutually strongly dependent, as opposed to the situation in p.t.p. networks. Consequently it is of interest to investigate algorithms which account for these dependencies to reduce total delay and improve channel and nodes' utilization, which are all negatively affected by message collisions, and by the following related considerations as follows.

In the shortest path routing policy given above nodes B, C cannot notice the collision in D, and so, it is necessary to add acknowledgements to the scheme. This makes the delays even longer. Note furthermore that acknowledgements may collide as well and must be scheduled randomly for retransmission [A73]. Hence a sending node cannot know for sure whether its message has been received, and the acknowledgement is on its way, or the message has been lost, and needs retransmission.

Notice further that in the example of figure 1.1, if we choose the shortest path routing strategy, node A cannot hear B and C repeating its message and
thus implied acknowledgements can not be used. If, however, we choose the collision free routing strategy then the repetition of the message by \( B \) may serve instead of an explicit acknowledgement.

With random access scheme for datalink control we can not guarantee a finite upper bound on the time needed for a message to get to its destination. Unfortunately most routing algorithms demand such information, since if the time needed for passing a message on every link is unknown, there is not much sense in asking "What is the best route?" In other words without coordination the datalink layer and the routing layer perform dependent time-calculations, each ignoring the transmission decisions made by the other.

Upper bound guarantees on the time in which the message arrives (at least for messages with high priority) are also needed by other network functions, such as flow control [T81], fault detection and handling, and topology mapping. An example for an application which needs a bounded time (but not necessarily very short) is a daily newscast, somebody may want to broadcast over the network, and to be sure it is arrived while it is still hot. Another example is voice passing.

In summary both the datalink layer and higher protocols serviced by it suffer from lack of coordination.

In searching for collision free protocols based on considerations related to global (i.e. end to end) network activity, we notice that collision free radio channel allocation to network nodes can be obtained by dividing it into fixed division of time between the nodes (TDMA), fixed division of frequency (FDMA) or both; or by using reservation [CF79],[PC81]. The known way to perform the time division is to give each node a different timeslot [SI82]. A more attractive way would be one which allows several nodes which "do not disturb each other" to transmit in the same timeslot. In other words a division with spatial reuse of radio frequencies is sought. In this work we investigate some division problems, and show that solutions can be found by approaching the channel allocation problem by taking a global view of network transmission activity. This global view is provided by the
broadcast activity

We shall try to explain this point as follows:

Note that part of the collision problem arises from the following fact: transmissions collide since the transmitting nodes are not aware of the others’ intention to transmit. This information cannot be known to the datalink function in advance, as the datalink protocol is concerned only with the two endpoints of the link. On the other hand, routing protocols can “anticipate” transmission intentions of some nodes on a given path, while broadcast protocols may know about the intentions of all the nodes (the example of fig 1.1 demonstrates how such information may be used).

In the following section we investigate the various aspects of time-slot and frequency assignments. From this investigation we derive algorithms which can serve for a collision-free channel allocation with spatial reuse from which efficient datalink, routing and broadcast protocols can be derived. We shall first prove that the complexity of the optimization of radio network performance is high (if NPH problems have high complexity) and we show that several natural approximation approaches have high complexity as well. We shall suggest, and justify, an approach with reasonable complexity (i.e. a polynomial approximation) based on a tree-like construction for broadcast routing which permits parallel collision free transmissions. We suggest a way to use this construction to derive broadcast, end-to-end routing and data link protocols with spatial reuse. We show that the bigger the dependency between the transmission decisions of different nodes, the bigger becomes the dependency between the routes, and the less sensible it becomes to work with multiple constructions i.e. a different routing tree for every possible source in the network.

Finally we show that for the discussed radio network model the problems and algorithms for achieving time division and frequency division channel assignments with spatial reuse are equivalent.
2. EXISTING SOLUTIONS

Algorithms for broadcasting and routing in radio networks are based on
flooding [T81] and on hierarchical routing. Collisions are solved by the data-link
protocol using random access, acknowledgments and retransmissions. [GVP76],
[GVP78], [K78], [SHG82]

a) A protocol for simple flooding:

Each node which accepts a message, relays it to all its neighbors. Thus a
node is bound to receive duplicates of every message at least one copy per
neighbor. To prevent the node from relaying the message again and again (once
per each duplicate), causing the number of duplicates to grow to infinity, a hop
counter and/or a packet identification memory are used. The hop counter is car-
ried in each message and decremented by every receiving node. When its value
becomes zero, the copy of the message is destroyed. The identification memory is
used by every node to remember the identity of each message it has recently
transmitted so that subsequent retransmissions can be prevented.

This broadcasting protocol is used also for embedding end to end routing.

b) A hierarchical end to end routing protocol which may be used for broadcasting;

In this protocol, the information flows over a tree (in graph terminology),
created at network initialization time. A message from a node is routed up the
tree, then down to the destination. Each node knows its father. The father
recognizes the son by the label carried in the son's message. The labels of the
sons include the label of the father. Sometimes the network can recover from
breakdowns, by finding another father to a son which lost his father. To avoid
loops, the new father must be higher in the hierarchy then the son.

Broadcasting initiated by the head of the hierarchy can be performed easily
by having every node send the message to all its sons in the tree. Notice that
although the solution is for a single source broadcast, broadcast initiated by
other nodes may be routed, as end to end messages, to the head of the hierarchy.
c) A directional flooding protocol:

In this protocol every node must know its distance in hops from every other node. It is different from flooding in that an intermediary node, U, will repeat a message received from V only if, relative to the message's destination, W, U is closer than V.

Notice that although the solution is suggested for end-to-end routing, a variation of it can be suggested to obtain a flooding broadcast protocol which is more efficient than the simple flooding. With this variation if a given node N has initiated a broadcast then node U, which received the message from V, will repeat it only if U is farther from N than V.

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. So a tradeoff exists in their efficiency and use of resources.

With the simple flooding protocol a node must know the identities of its neighbors. This difficulty is introduced by the radio medium property since in contrast to PTP a node is not aware of the outgoing "links." This protocol causes generally a larger number of retransmissions than the other two. The protocol requires space and bandwidth for the packet identification memory, for the hops counter and for the duplicates created.

Under the more efficient directional flooding protocol, every node must further know the number of hops to every other node. Intuitively this protocol causes less retransmissions and thus needs less space and bandwidth for duplicates, while on the other hand additional space is required for the distances table. Also, the distances carried in the messages consumes bandwidth.

In the hierarchical protocol additional memory is needed for a spanning tree to be built, the number of retransmissions should be still smaller however. Space and bandwidth are needed only for recognition of father and son relations.
general one can assume that the more global information is required in every node, the more expensive is the protocol initialization.

Since any transmission can result in accumulation none of these protocols gives a time bound on message delivery. The broadcasting property is not fully taken account of in the sense that collisions prevent the use of implied acknowledgments. In addition a commonly noticed distinction between radio and p.t.p. routing, lies in the placement of the responsibility for the decision "on which link should the message be transmitted". In radio networks this decision belongs naturally to the receiving node which decides whether to accept the message or not, as a message is often received by neighbors of the transmitting node no matter if it is needed for further transmissions. With the protocols discussed the transmitting node cannot be relieved from the responsibility for routing as there is a need for acknowledgments from all its neighbors. These acknowledgments cannot be accepted simultaneously and so the transmission process becomes similar to sending copies to all the neighbors on different lines like in p.t.p. Those "lines" are however not reliable (unlike the p.t.p. lines) and no parallel message deliveries can be done on them (because we must consider the acknowledgement as a part of the message delivery process).

3. MEDIUM AND MODEL DESCRIPTION

To formally analyze broadcasting protocols in radio networks we introduce the following model. A node having a message to broadcast will transmit it in a single transmission interval or in several shorter intervals, or time slots. The broadcasting node can engage the whole bandwidth available in the network or the bandwidth can be divided into narrower bands. Both time and bandwidth can be considered basic network resources which have to be shared. The sharing is done in basic units we term "phases," so that time division only is used (i.e., each transmission employs the whole bandwidth) a phase corresponds to a basic time slot. If frequency division alone is used a phase corresponds to a band.
We deal with multihop connected radio networks, i.e., not all nodes are in line of sight and in range but there is a possibility to pass a broadcast from the originating source node (whenever such a node is defined) to all nodes in the network. We shall term these networks 'source connected'. Note that source connectivity is a weaker definition than 'strong connectivity' and is stronger than 'connectivity' in directed graphs. We shall assume that if a node N1 is in line of sight and in range of node N2 then, when N1 transmits, N2 always receives a signal. However, collisions may prevent N2 from receiving information stored in the signal, i.e., two or more transmissions over the same band, which are received simultaneously in node N2, are interpreted by N2 as mere noise. While transmitting on a certain band, node N2 cannot recognize other transmissions over the same band. We say that node N2 hears the transmission from N1 if the transmission is received correctly, i.e., without collision.

To represent a radio network, we use a directed graph where the vertices are the network nodes, and an undirected edge is interpreted as two anti parallel directed edges. With this representation, we can define the above concepts formally. A directed edge $u \rightarrow v$ means that $v$ can hear $u$ subjected to the conditions 3.1 and 3.2:

**Condition 3.1:** $v$ does not transmit at the same time $u$ does or $v$ does not transmit on the same band $u$ does. (i.e., $v$ can not hear anything in the same phase it is transmitting).

**Definition:** Let us call a neighbor, $w$, of $v$, 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 at that time on that band (i.e., $v$ can not hear simultaneously two transmissions on the same band).

If condition 3.1 does not hold we say that $u$'s transmission collided in $v$. If $u$ and $v$ have transmitted but condition 3.1 held we say that $u$ and $v$ have transmitted in different phases. If condition 3.2 does not hold, we say that the
transmissions of the incoming neighbors, \( u \) and some \( w \), have collided in \( v \), if \( u \) and \( w \) have transmitted but their transmissions have not collided we say that their transmissions were in different phases. For subsequent use we note that if collisions in \( v \) occur due to violation of condition 3.2 then only \( v \) is aware of the collision but cannot know the origin of the collided transmissions. When collisions in \( v \) occur due to violation of condition 3.1 then even \( v \) is not aware of the collision. These conditions are demonstrated in figure 3.1 and 3.2.

4. OPTIMAL BROADCASTING PROTOCOLS ARE NOT PRACTICAL

We look for a "good" protocol which performs a broadcast from a given source. In this section we show that there are theoretical difficulties in constructing such a protocol. We define a "good" protocol to be one which minimizes one (or more) of the following measures:

- \( D_{\text{max}} \): the maximum time needed for the broadcasted message to reach in its entirety all nodes.
- \( D_{\text{avg}} \): the average time, over all the nodes, needed for the broadcasted message to arrive entirely at each node.
- \( M.r. \): (average memory requirements) the average number of buffers needed, multiplied by the time each buffer is needed.

\[
\frac{1}{n} \sum_{i \in V} B_i t_i
\]

where \( B_i \) is the number of buffers needed in node \( i \), and may equal 0 or 1, \( n \) is the number of nodes (i.e. \( \text{SIZE}(V) \)). A node, \( v \), has to keep a broadcasted message in its memory till the message is no longer needed for \( v \) to transmit it, in doing so part in the broadcast.

- \( \text{NC.avg} \): The average node consumption, or put differently, the normalized average time a node is required to dedicate its attention to the broadcast.
To obtain $N C_{\text{avg}}$ we define the following:

$t_{ocu}(i,b)$ - the number of timeslots during which node $i$ can not accept messages on band $b$ (other than the broadcasted message, because it is transmitting on that band, or because at least one of its neighbors is transmitting on it).

$|b|$ - the bandwidth of band $b$ divided by the total bandwidth.
We then define $NC$ (node consumption in $n$ timeslots) as

$$NC = \frac{1}{n} \sum_{i \in V} \sum_{b \in \text{BANDS}} |t_{\text{gcu}}(i,b)|$$

and define $NC_{\text{avg}}$ (average node consumption in $n$ timeslots) as

$$NC_{\text{avg}} = \frac{1}{n} \cdot NC$$

Clearly we want to minimize the time, per broadcasted message, each node and the channel are occupied by the broadcast activity in the network so that they can be relieved for other transmissions.

The following discussion is given to motivate and clarify the use of $NC_{\text{avg}}$ as a measure in radio networks. In p.p.p-network and in networks using buses we try to minimize the occupation of the lines by the broadcast. In multihop radio networks the links overlap (for example in figure 4.1 D can not send messages to A while E is sending a message to C). Still links from a node should lead, of course, to other nodes. Therefore measuring the occupation of the nodes (the time other messages can not be accepted by them) seems to capture also the notion of

![Diagram of a two buses network](image1)

![Diagram of a radio network](image2)

Figure 4.1: Consumptions in different networks
occupation of the channel.

For convenience we normalize NC by n - the number of nodes in the network, as we know for any network how to perform a broadcast in n slots (provided that there is no competition from other activities). When transmissions do not occupy the whole bandwidth we normalize also by |b| which is the fraction of the total bandwidth on which the node can not accept other messages.

The following examples are given to show that NC.avg does capture the intuitive notion of resources (nodes and channel) utilization. In the case of the network given in figure 4.2, we want to perform a broadcast initiated by node s. There are two branches emerging from s. Two possible strategies are

1. have the broadcast propagate along the broken line in figure 4.2a.
2. have the broadcast propagate by parallel transmissions along the broken lines in figure 4.2b (the broadcast propagate separately in each branch).

The two strategies differ in D.max and D.avg but the average node consumption converges to the same amount as the branches get longer, as each node (except for the endpoints of the branches) is occupied 3 times by the broadcast under the two strategies.

In the case of a single common bus there are two possible situations: If nobody transmits then NC.avg = 0. If there is exactly one timeslot in which at least one node transmits then NC.avg = \frac{x}{n}. If there are x timeslots (out of arbitrary n) in which at least one node transmits then NC.avg = \frac{x}{n}. In other words NC.avg will correspond to channel utilization which also corresponds to the fraction of time (out of n slots) during which the nodes attention is occupied.

In the example of figure 4.1 there are a radio network and a two buses network. These networks look very similar but NC should be different, assuming that N can ignore the transmission on one bus and choose to accept the transmission on the other. In the radio network if the clique which includes B, includes K1 nodes (including N) and the other clique includes K2 nodes (including N), then if
Figure 4.2: Two broadcast strategies with the same consumption.

B sends a message to C then $NC = \frac{1}{n} \cdot \frac{k1}{K1 + K2 - 1} = \frac{1}{n} \cdot \frac{K1}{n}$. In the two burst network $NC = \frac{1}{n} \cdot \frac{k1 - 1}{n}$. 
In the following theorems we show that the complexity of optimal protocols is high (if $P = NP$) when a message is broadcasted in one piece. Clearly, an algorithm which divides the messages to more basic units and sends them one by one is even more impractical, assuming that it must know how to optimally send the basic unit.

**Definitions:**

We say that the source node 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.

**Theorem 4.1:** The problem of $\text{D max minimization (problem P4.1)}$ is $\text{NP}$-hard [Wi79].

**Problem P4.1:**

**Input** an undirected graph $G(V,E)$ (representing a network), a vertex $s$ in $V$ (we shall call $s$ the "source"), an integer $f$. The network is source connected as defined in section 3.

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

**Proof** (of theorem 4.1). We prove by reducing an $\text{NPC}$ problem to the following problem: "can a broadcast be performed (in the way defined in problem P4.1) in $f$ timeslots or less?" (we shall refer to this problem as Problem P4.1). while $f$ is given in the input. We shall show this by showing that the problem remains $\text{NPC}$, even if $f$ stays fixed, and is equal to two (Problem P4.12). The reductions from P4.1 to P4.11 and from P4.1 to P4.12 are obvious. The problem we shall reduce problem P4.12 to, is called $3\text{XC (3 exact cover)}$ [E79].

**Input** a collection $C$ of sets, three members in each $U$ is the union of the sets.

**Question** does there exist a subcollection $C'$ included in $C$, such that every member $u$ in $U$ is a member in exactly one set $c$ of $C'$?
**Lemma 1A.1:** problem P4.1.2 is NP-complete.

**Proof of the lemma.** Given an input to the 3XC problem, let us transform it to be a case of problem P4.1.2 in the following way: Let $G$ 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 Figure 4.3. Given a solution to the original 3XC problem, we can perform a broadcast in $G$ in the following way: in the first timeslot $s$ transmits, and all the sets hear, and in the second timeslot, all the sets participating in the cover $C_i$ 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 problem P4.1.2, we take the $c$ vertices which have to transmit. Apparently, the union of the sets they represent, include all the $u$ vertices.

---

Figure 4.3: The reduction from 3XC to the broadcast delay problem (2 timeslots delay)

---

*We do not prove for problems to be in NP.
vertices; as the broadcast has to get to all the vertices. Also, every $u$ vertex is a
neighbor (son) only to one $c$ vertex, as it can hear only in the second timeslot; as
the first must be dedicated to the source, and if two of its neighbors transmit
then a collision occurs in it, and it can't hear.

Q.E.D.

Hence problem $P4.1.1$ is NPC, and problem $P4.1$ is NPH.

Q.E.D.

Remark: given any constant $I$ (not only for $I=2$), is it possible to perform the
broadcast in $D_{\max}$ less then or equal to $I$ (problem $P4.1.3$)

Claim: problem $P4.1.3$ is NPC.

Proof: We use the following construction: $I-1$ vertices $s_1, s_2, \ldots, s_{I-1}$, while $s_i$ is
connected by an edge to $s_i$ and $s_i$ is connected to $|C|$ vertices, one per each $c$ in $C$.
There is also a vertex for every $u$ in $U$, connected to all the c vertices, in it's set it
participates. The source is $s_1$. See figure 4.4.

Q.E.D.

The proof from here on is very similar to the proof for problem $P4.1.2$, as,
apparently, the solution must use the first $I-1$ timeslots on passing the broadcast
on the s's, and only one-timeslot is left for transmission, by the c's, to the u's.

Q.E.D.

Theorem $T4.2$ The problem of $D_{\text{avg}}$ minimization (problem $P4.2$) is NPH.

Proof: We prove by reducing an NPC problem to the following problem (problem
$P4.2.1$).

Problem $P4.2.1$

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

Question "can a broadcast cover the network in an average time less or equal to
$I$?"
Figure 4.4: The reduction from 3×C to the broadcast delay problem (l timeslots delay)

We use the construction of problem P4.2.2. If there is an exact cover, then the average number of timeslots is 
\[
\frac{(2 \cdot |U| + 1 - |C| + 0)}{|U| + |C| + 1}
\]
and if there is no exact cover, the values of the denominator and the second term in the numerator stay 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 timeslot.

Q.E.D.

Theorem 4.3 The problem of NC.avg minimization (problem P4.3) is NP-H.

Proof: We prove by reducing an NPC problem to the following problem:

Problem P4.3.1

Input: an undirected graph G(V,E), a vertex s (the source) in V, an integer I. The network is source connected. Nodes' transmissions occupy the full bandwidth.
Question: can a broadcast cover the network using NC.avg which is less or equal to $I$?

We use the same construction as the one in problem P4.1.2, but we shall add an edge between every two c's in C. If there is an 3XC, then the sets are busy twice, and the members once. Otherwise, the sets are busy more then twice, and part of the members - more then once.

Q.E.D.

Theorem T4.4: The problem of minimizing $M.r$ (problem P4.4) is NP-hard.

Proof: We prove by reducing an NPC problem to the following problem:

Problem P4.4.1:

Input: an undirected graph $G(V,E)$, (representing a network, as explained in section 2), a vertex $s$ in $V$ (we shall call $s$ the "source"), an integer $I$. The network is source connected (as defined in section 3).

Question: can a broadcast cover the network using $M.r$ which is less or equal to $I$?

We use the same construction as in problem P4.3.1. If there is an exact cover, then the buffer in the source node is needed for one timeslot, and there is also a need for one buffer timeslot in exactly $\frac{1}{3} \text{SIZE}(U)$ vertices corresponding to sets. Otherwise the number of set vertices having to keep a buffer would be bigger, or the source node would have to keep the message longer.

Q.E.D.

5. INVESTIGATION OF CANDIDATE HEURISTICS

Since optimal solutions are not practical, we attempt to design approximate solutions to achieve certain degree of optimization for the measures defined in the preceding section. We observe that the high complexity of exact solutions arises from the fact that actions taken by each node are a function of the action taken by all the other nodes in the network. It is easy to see that this relation
function is highly complex as the events near a given node $N$ depend on the events near its neighbors which in turn depend on the events near their neighbors (including $N$ itself) and so on. The heuristics come from trying to loosen those requirements that seems to lead to the most complicated relations.

5.1 A heuristics for reducing Average Node Consumption ($NC_{avg}$)

Recall that node consumption is induced by a given broadcasting protocol in those timeslots in which the node, or a neighbor of that node transmit the broadcasted message. Since minimizing node consumption itself is an NPH problem we try first to approximate node consumption minimization at only those instances in which node's attention is dedicated to it's neighbors.

Let us define the "disturbance quota" of node $i$ ($disturbance^{\text{quota}}_i$), as the number of timeslots in which at least one of node $i$'s neighbors is transmitting the broadcasted message.

**Heuristic H1:** perform the broadcast minimizing the average disturbance quota.

$$\frac{1}{n} \sum_{v \in V} disturbance^{\text{quota}}_v$$

**Theorem T5.1:** The problem of minimizing the average disturbance quota (problem $P5.1$) is NPH.

**Proof:** We prove by reducing an NPC problem to the following problem:

**Problem P5.1 1**

**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 with $\frac{1}{n} \sum_{v \in V} disturbance^{\text{quota}}_v = 1$?

We use the construction of problem $P4.1$ If there is a solution to the 3XC problem, then the disturbance quota of every vertex equals 1. If there is no 3XC then the disturbance quota of the source and some of the vertices representing members, gets higher as some of the vertices representing members has to receive transmission more than once because of collisions.
The minimization of the average disturbance quota is potentially complicated by the following relation between the transmission emanating from neighbors of node i: if the neighbors' transmissions are simultaneous, then a single disturbance unit is counted at node i. Otherwise more than one disturbance units are counted. Of course these neighbors participate with other nodes in the creation of disturbance units at other nodes, which in turn contribute to disturbance units at nodes other than i, etc. We wish to loosen this complexity of counting (and subsequently minimizing) the disturbance quota by ignoring this transitive relation and observing instead only the total number of copies sent. We count a transmission of a node with x neighbors as x transmissions, since during every transmission the node "disturbs" all its neighbors.

**Heuristic H2:** Perform the broadcast while minimizing the total number of transmitted copies.

**Theorem P5.2:** The problem of minimizing the number of copies sent in performing a broadcast (problem P5.2) is NPH.

**proof:** We prove by reducing an NPC problem to the following problem:

**Problem P5.2.1**

**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 with the number of copies sent less than I?

We use the construction of problem 4:1. If there is a 3XC then each member vertex accepts only one copy, and the source accepts copies by the number of members divided by 3. Else, some members receive more than one copy, and extra copies are sent also to the source which is a neighbor of the sets.

**Q.E.D.**

To further reduce the complexity we attempt to simplify the problem by looking at a more loosely defined approximation to the optimal average node
consumption obtained by counting each transmission only once, (instead of once per neighbor of the transmitting node).

**Heuristic H3:** Perform the broadcast minimizing the total number of transmissions.

Note that heuristics H2 and H3 resembles the minimum spanning tree broadcasting policy in p.p.p. networks [DM78]. This policy tries to minimize the cost of utilizing the links. Since no edges length is defined in our model, this cost corresponds to the number of transmissions or number of copies.

**Theorem T5.3:** The problem of minimizing the total number of transmissions for a broadcast (problem P5.3) is NPH.

**proof:** We prove by reducing an NPC problem to the following problem:

**Problem P5.3.1**

**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 with the number of transmissions less or equal to $I$?

We use the construction of problem P4.1. If there is an exact cover then the source transmits once, and there are exactly $\frac{\text{number of members}}{3}$ transmissions of set vertices. Else the number of transmissions of set vertices is bigger.

Q.E.D.

Note that heuristic H3 also attempts to optimize node consumption contributed by condition 3.1 i.e. a node can not accept messages on a band it is transmitting on.

**5.2 A heuristic for reducing the delay**

Notice that a broadcast of a message from a given source is an iterative process in which in every iteration (time slot) some of the nodes which have already received the message transmit it to some of their neighbors that 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. It is easy to observe that this order relation potentially complicates the delay minimization. Accordingly we can attempt to simplify the problem by looking not at the whole sequence of ordered iterations which defines the broadcast, but rather at each iteration or height independently.

**Heuristic H4:** 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.

Note that heuristic H4 resembles the hot potato forwarding broadcasting policy, used in p.t.p. networks [DM78] whereby the policy of each node is to forward the message as soon as possible, no matter which direction.

**Theorem T5.4:** The problem of maximizing the number of nodes which receives the message in the next timeslot (problem P5.4) is NPH.

**Proof:** We prove by reducing an NPC problem to the following problem:

**Problem P5.4.1**

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

**Question** can we pick up a subset of A, such that if they transmit in the next slot then more then I vertices will become covered (i.e. will enter the set A).

In [EGTB81] there is NP completness proof for the following problem: It is necessary to check the edges of the network (the links). Such a check, in one direction, is done by having a vertex, the edge is connected to, transmit, and having the vertex in the other end hear, under conditions 3.1, and 3.2: A part of the vertices of the network can ONLY receive, and the others are able ONLY to transmit. The question is whether it is possible to divide those who are able to transmit, to two sets: a silent, (in the next timeslot) set, and a transmitting set, such that in the next timeslot, at least I edges will be checked. It is easy to find a reduction, from this problem to problem P5.4.1: Let us add a source vertex, connected to all those vertices which can only transmit, and only to them. Apparently there is a positive answer to the case of problem P5.4.1 we have created, if and
only if this is the answer to the original question.

Q.E.D.

6.3 A heuristic for minimizing delay and node consumption

Heuristic H5: Perform the broadcast minimizing the total delay under the restriction that each node is permitted to transmit only once.

This heuristic is attractive since it demands low effort on the part of each node and consequently should lead to low node consumption. The heuristic is also attractive for delay minimization. While with H4 we attempted to simplify the order in which messages were received, heuristic H5 attempts to simplify the "complementing" time order—the order in which the nodes transmit. Since with previous heuristics a node may be required to transmit the broadcasted message several times, it may appear several times in this order. Heuristics H5 limits the number of such appearances to one (Notice also that if we divide the channel by frequency division, instead of time division, then in this approach each node has to have only one transmitter. This point will be discussed further in section 9).

Theorem T5.5: The problem of minimizing the delay under the restriction that each node may transmit only once (problem P5.5) is NPH.

proof: We prove for the maximum delay by reducing an NPC problem to the following problem:

Problem P5.5:1

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 problem 4.1.1 given a special case \( \text{G}=(V,E) \), for problem 4.1.1, let us create \( \text{G}'=(V,E') \) as below— for every \( v \) in \( V \) we create \( |V| \) vertices \( v_1, v_2, \ldots, v_{|V|} \). If the edge \( u-v \) is in \( E \), then \( E' \) includes the edges \( u_i-v_j \) for each \( 1 \leq i, j \leq |V| \). Given a solution to problem 4.1.1 it is obvious that each \( v \) in \( V \) is used.
no more than \(|V|\) times, as, because of the connectivity assumption (see section 2), we can perform the broadcast in \(|V|\) timeslots. 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 \((L < |V|)\), in \(G\), \(v, \hat{v}, \hat{v}_1, \hat{v}_2, \hat{v}_L\) transmit one time each, in the same timeslots \(v\) transmits in \(G\). From the definition of \(E\) it follows, that a transmission of each of the \(v\)'s \(i\)'s can be received by all the \(u\)'s \(j\)'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 timeslot one of the \(u_j\)'s transmits.

Q.E.D.

5.4 A heuristic for minimizing the memory requirements (M.r)

Notice that a node must keep a copy of a broadcasted message in its buffers from the timeslot it accepts it and until this copy is successfully sent to all of those \(i\)'s neighbors which depend on that copy to receive the broadcasted message, i.e. those neighbors covered by node \(i\) before node \(i\) has the chance to receive another copy. To cover all these neighbors \(i\) may be required to transmit the message in several timeslots, since the reception of \(i\)'s transmission depends also on the activities of the neighbors and their neighbors in turn. To minimize the time a node is required to store a copy of the message, all subsequent transmissions must therefore be coordinated. This yields an NPH problem. We simplify the requirement for total coordination as follows. The idea is to enable each node to cover all of it's uncovered neighbors by a single transmission (not necessarily in the slot following slot in which the message was received). To ensure such covering we require that two nodes, with a common uncovered neighbor, do not transmit in the same timeslot. This requirement guarantees that if a node transmits a message to a neighbor simultaneous transmissions of that neighbor's neighbors will not interfere with it. By introducing this restriction we hope to reduce memory requirement by choosing a policy from a more limited number of potential transmission sequences.

Heuristic H6: Perform the broadcast minimizing M.r under the restriction that no two nodes, having a common uncovered neighbor, may transmit in the same
timeslot.

**Theorem T5.6:** The problem of minimizing $M_r$ under the restriction that no two
nodes, having a common uncovered neighbor, may transmit in the same timeslot
(problem P5.6) is NPH.

**Proof:** We prove by reducing an NPC problem to the following problem.

**Problem P5.6.1**

*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 using $M_r$ which is less or equal to $I$, under the restriction that no two nodes, having a common uncovered node may transmit in the same timeslot.

The reduction from an NPC problem is the same as for P4.4.1.

Q.E.D.

6. **DIVISION OF THE NETWORK INTO MUTUAL NON INTERFERING AREAS**

Another way to simplify complex dependency relations is to limit the number
of nodes and links to be considered in the relations. We attempt to partition the
network into areas (not necessarily disjoint) in a way which permits us to ignore
actions undertaken by nodes outside a given area. A good division into areas
should then enable nodes to perform simultaneous transmissions without collisions. This Spatial Reuse property of protocols thus obtained, seems intuitively to optimize node consumption. It also seems helpful in reducing memory require-
ments since with interference avoided the time nodes have to keep the 'message'
in their buffers is bounded. In this section we show that the search for optimal
partitioning with the above properties (maximum parallel transmissions) results
in an NPH problem. However a more limited search for partitions will be showed to
yield polynomial algorithms.
6.1 Optimal partitions of networks

Given the iterative view of the broadcast process starting at the source (see 5.2) whereby in each iteration some of the nodes which have already received the broadcasted message relay it to some of their neighbors, we define:

Definition

Belt- the set of nodes which transmit in the same iteration.

CLIENTS of belt Belt₁ CLIENTS(Belt₁)- the set of those nodes which depend on belt Belt₁ to receive the broadcasted message.

Put differently CLIENTS(Belt₁) receive the broadcasted message from, and only from, a node in Belt₁ prior to their own transmission. Figure 6.1 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 Belt₂ and Belt₅ may transmit in the same phase (for example they may transmit simultaneously consecutive packets of a broadcasted message).

On the basis of the preceding discussion we can define the following heuristic for minimizing average node consumption.

Heuristic H7: Perform the broadcast using a division into belts that minimizes the total number of phases needed (there is a need for different phases for certain belts to ensure that the transmissions of any belt do not collide in it's CLIENTS).

Theorem T6.1: The problem of dividing the network into belts, minimizing the number of phases required (problem P6.1) is NPH.

Proof: We prove by reducing an NPC problem to the following problem:

Problem P6.1.1

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

Question can the graph be divided into belts in such a way that the number of phases required is less than I?
Proof Like in problem P4.1 we reduce to the subproblem of fixed $l$ equal to 2 (we denote the subproblem as problem P6.1.2). Likewise we reduce problem P6.1.2 to $2XC$, and the reduction uses the same construction as that of the proof of problem P4.1.2.

If there is an exact cover, then the graph could be partitioned into two belts, using each a one phase, in the following way: The first belt includes only the source. The second belt includes those sets which participate in the cover. Obviously, all the vertices in the second belt accept the message together, and when they transmit together, in the second phase each member receives exactly one transmission from the one set which covers that member in the exact-cover.

If there is a partition of the graph into belts using only two phases then there is an exact cover. The first phase must be assigned to a belt including only the source—so there is left only one phase in which the sets may transmit. As the
partitioning into belts is done in a way which enables every vertex to receive the broadcasted message. It comes out that in the second phase transmit set vertices which each member vertex is a neighbor of exactly one of them. Hence an exact cover.

Q.E.D.

6.2 Partition of networks by polynomial algorithms

The high complexity of the last problem is likely to result from the fact that too many subsets of vertices are candidates for belts since it is well known that the total number of subsets of a given set is exponential in the number of its members \(2^n\). This suggests that a polynomial heuristic should be searched by limiting the number of candidate belts, so that the procedure will not check all the possible sets which are candidates for belts, but builds the belts incrementally. Two basic approaches we use in constructing such heuristics are the good neighbor approach and the greedy approach.

The basic idea is derived from the fact that a broadcast flows over a tree rooted at the source. The approximating algorithms construct this tree by building the belts and assigning phases. We look, for each node, at the area which includes the node, its neighbors and their neighbors. The good neighbor approach says that a phased assignment must guarantee that a node should not prevent its neighbors from hearing their father in the tree. The greedy approach says furthermore that each time a phase is assigned to a node, it should be the phase with the lower sequential number (of phases) among those that can be

![Diagram of network with nodes and phases](image)

**Figure 6.2:** Nodes far apart are assigned the same phase.
assigned. Note that the "areas" for the approximating algorithms are the neighbors in the previous belts as are probably the areas for the NPH problem.

Definitions:

A phased node: a node (vertex) which has already been assigned a transmission phase (say, phase $j$), by the algorithm.

A vertex, $v$, hears from $u$ in phase $i$ iff $u$ is the only neighbor which is phased by phase $i$.

A vertex "hears" or "is hearing": iff it has at least one phase in which it hears.

$K$: the maximum number of neighbors a vertex in the given graph has. When the graph is directed, $K$ is the maximum outbound degree (i.e., number of outgoing edges a node has).
Algorithm good neighbor (for constructing the broadcast tree and assigning phases)

(0) divide the channel into $K+1$ phases.
(1) assign phase 1 to the source.
while there is a hearing vertex, $v$, which is not phased and which has a neighbor that does not hear
  do begin
    (2) pick such vertex, $v$.
    (3) choose one of the vertices $v$ hears from, as $v$'s "father". ($v$ is a "son" of it's father).
    (4) assign to $v$ a phase (out of those created in step (0)) which meets the following conditions:
      (a) no phased vertex may stop hearing it's father (e.g. if $v$ is a neighbor of $u$, and $u$ has a father, the algorithm should not assign $v$ the phase of $u$'s father)
      (b) no unphased vertex, which is, however, hearing, may stop hearing (i.e. if $v$ is a neighbor of $w$, and $w$ hears only from one vertex, $z$, which is, thus, the only candidate to be $w$'s father, $v$ may not be assigned the phase of $z$)
      (c) $v$ may not be assigned the phase of it's father.
  end (of while statement)
(6) for every hearing vertex, $v$, which is not phased, choose as $v$'s father the first neighbor that became phased among those that $v$ hears from.
(7) stop.

Theorem T6.2: (properties of the good neighbor algorithm)
(a) The good neighbor algorithm does not assign more than $K+1$ phases.
(b) The algorithm always stops successfully.
(c) The algorithm constructs a spanning tree such that the broadcast can cover
the network by hearing each father in the tree transmit in its phase.
(d) The algorithm is polynomial (in execution time).

Proof

(a) In (0), only \( K+1 \) phases are defined, and the assignment (which is done only
in (4)) is only out of those \( K+1 \) phases.

(b) Lemma 16.1: when there is a hearing node, \( v \), which is not phased and has an
unhearing neighbor, there is always a phase (out of the existing \( K+1 \)) which
meets the condition in (4) and thus can be assigned to \( v \).

Proof of the lemma. Suppose \( u \) is \( v \)'s father and \( u \) is forbidden to transmit in the
following phases: the phase in which \( u \) transmits; if \( u \) is not the source, also
the phase of \( u \)'s father, and maybe, the fathers of \( v \)'s other neighbors. \( v \) has
at most \( K-1 \) neighbors (except \( u \)). If all of them hear, already, there is
no need for \( v \) to transmit. Otherwise, at least one of them does not hear, so,
at most \( K-2 \) of them hear, in at most \( K-2 \) other phases, which may be for­
bidden for \( v \). So, \( v \) is forbidden to transmit in at most \( K \) phases, so we can
assign it the \( K+1 \)-th phase.

Q.E.D. (lemma)

It follows from the lemma and step (2) that in each iteration of the algorithm
another vertex becomes phased. As the number of vertices is finite, the algorithm
must stop.

(c) Lemma 16.2: when the algorithm assigns a phase, no hearing vertex stops to
be a hearing vertex.

Proof of the lemma. Immediate from the conditions in (4).

Lemma 16.3: when the algorithm stops all the vertices hear.

Proof of the lemma. Suppose that there is a vertex, \( v \), that does not hear
when the algorithm stops. Let \( N_v \) be the set of vertices in the maximal con-
nected subgraph which includes v, such that no vertex in \( N_v \) hears. As there is at least one hearing node in the graph (the source) and according to the connectivity assumption (see section 3) there must be a node \( u \) in \( N_v \) that has a hearing neighbor.

When the loop at (2) is finished, there was no unphased hearing vertex with a neighbor which does not hear. Thus \( u \) has a phased neighbor. Suppose \( w \) was the first neighbor of \( u \) to become phased. When \( w \) became phased, \( u \) became a hearing vertex and according to the previous lemma \( u \) is still hearing. Contrast to the assumption that \( u \) is in \( N_v \). Thus there can not be such a vertex, \( v \), which is not hearing at the termination of the algorithm.

Q.E.D. (lemma)

**Lemma 1.6.4:** The father-son relation between the hearing vertices defines a tree, rooted in the source. If vertices transmit only in their phases then the transmission of a father can be heard by the son and can never collide in the son.

**Proof** (by induction on the order of the entrance of vertices to the tree).

First we show that the phased vertices construct such a tree. When the source is the only phased vertex the claim holds. Suppose the claim holds till the \( j \)-th iteration. In the \( j+1 \)-th iteration we pick an unphased vertex, \( v \), and thus it is not part of the tree. In (3) we choose, as \( v \)'s father, a phased vertex, and thus a vertex in the tree. An edge from a vertex in the tree, to a vertex which was not in the tree, can not close a circuit. The choice of the phase for \( v \) (in (4)) is done in such a way, that if vertices transmit only in their phases then

(i) every vertex that becomes a hearing vertex can really hear \( v \) with no collision, and

(ii) if a vertex has previously been a hearing vertex then it still can really hear with no collision at least one of those it is defined as hearing from.
When (6) is executed (it happens only once, and it is before the termination), no circuit is closed as only new leaves are added. Also from the previous lemma and from (i) and (ii) we conclude that the added leaves can hear their fathers with no collision if all the vertices transmit only in their phases.

Q.E.D. (lemma).

From the lemma we conclude that the broadcast can cover the network in the following way: the source transmits in its phase and then, every phased vertex which receives a transmission from its father, starts to transmit in its own phase.

(d) In each iteration of the loop of steps (2) to (5) another vertex becomes phased so the number of iterations is bounded by $\text{SIZE}(V)$ where $V$ is the set of vertices in the graph, and $\text{SIZE}(V)$ is the number of vertices in $V$. Every hearing node would have to be examined to see whether it has an unhearing neighbor (bounded by $K$) but if a hearing node is examined in one iteration and has no unhearing neighbor, then there is no need to examine it again in the other iterations. In step (4) each of the phases $(K+1)$ may be examined if it does not interfere with each of the neighbors (bounded by $K$). The complexity of step (6) is no more than $O(\text{SIZE}(V))$. Thus the complexity of the good neighbor algorithm is no more than $O(K^2 \cdot \text{SIZE}(V))$.

Q.E.D.

Note that in the case of a directed graph the only neighbors, of node $N$, to be considered in step (6) of the algorithm are those to which an edge from $N$ exists. Thus in a directed graph the bound is $K+1$ where $K$ is the maximum outgoing degree. The proof for this case is exactly the same as for the undirected case.

As demonstrated above the bound provided by the good neighbor algorithm depends on the largest number of neighbors a node in the networks has. The addition of the greedy approach provides another bound which depends on the total number of edges. This bound $(2 \cdot \text{SQR}T(\text{SIZE}(E)))$ is thus more promising for networks in which the number of neighbors varies greatly from node to node.
Recall that the average number of neighbors is $\frac{2\cdot \text{SIZE}(E)}{n}$ and in graphs with many edges $\text{SIZE}(E) = O(n^2)$.

The greedy algorithm is similar to the good neighbor except that the phase with the lowest number which meets the conditions is the one to be assigned.
The greedy algorithm (for constructing the broadcast tree and assigning phases):

(0) divide the channel into $K+1$ phases.

(1) assign phase 1 to the source.

while

there is at least one hearing vertex, $v$, which is not phased and which has a neighbor that does not hear

do begin

(2) pick such vertex, $v$

(3) choose one of the vertices $v$ hears from, as $v$'s "father". ($v$ is a "son" of its father)

(4) assign to $v$ a phase (out of those created in step (0)) with the lowest number (of phases) which meets the following conditions:

(a) no phased vertex may stop hearing its father (e.g. if $v$ is a neighbor of $u$, and $u$ has a father, the algorithm should not assign $v$ the phase of $u$'s father)

(b) no unphased vertex, which is, however, hearing, may stop hearing (i.e. if $v$ is a neighbor of $w$, and $w$ hears only from one vertex, $z$, which is, thus, the only candidate to be $w$'s father, $v$ may not be assigned the phase of $z$)

(c) $v$ may not be assigned the phase of its father.

end (of while statement)

(6) for every hearing vertex, $v$, which is not phased, choose as $v$'s father the first neighbor that became phased among those that $v$ hears from.

(7) stop.

Theorem 16.3: (properties of the greedy algorithm)

(a) The greedy algorithm does not assign more than $K+1$ phases, where $K$ is the maximum number of neighbors a vertex in the graph may have, (same as
(b) The algorithm always stops successfully (same as theorem T6.2 (b)).

(c) The algorithm constructs a spanning tree such that the broadcast can cover the network by having each father in the tree transmit in its phase (same as theorem T6.2 (c))

(d) The algorithm is polynomial in execution time (same as theorem T6.2 (d)).

(e) The number of phases assigned is less or equal to $2 \cdot \sqrt{\text{SIZE}(E)}$ where $E$ is the set of edges in the graph and $\text{SIZE}(E)$ is the number of edges in $E$

**Proof**. Assume without loss of generality that we run the algorithm with the following change to step (0)

(0) divide the channel into $\text{Min}\{(K+1),(2 \cdot \sqrt{\text{SIZE}(E)})\}$.

All the proofs for theorem T6.2 remain unchanged, except lemma L6.1 which we replace by lemma L6.5.

**Lemma L6.5**: When there is a hearing node, $v$, which is not phased and which has a neighbor that does not hear, there is always a phase which meets the condition in (4) and thus can be assigned to $v$.

**Proof**

If $\text{Min}\{(K+1),(2 \cdot \sqrt{\text{SIZE}(E)})\} = K+1$, then the proof is the same as the proof for the lemma in theorem T6.2. Suppose that $\text{Min}\{(K+1),(2 \cdot \sqrt{\text{SIZE}(E)})\} = 2 \cdot \sqrt{\text{SIZE}(E)}$ Suppose also that there is a hearing node, $v$, which is not phased, and there is no phase that could be assigned to $v$ (no phase meets the condition in step (4)) Thus $v$ has a set $Z$ of at least $|Z| = 2 \cdot \sqrt{\text{SIZE}(E)}$ neighbors, hearing each in a different phase.

Let $z$ be a vertex in $Z$, define $f(z)$ as the vertex from which $z$ hears. Let $F$ be the set $\cup z \in Z f(z)$. Clearly, $\text{SIZE}(F) = \text{SIZE}(Z) = 2 \cdot \sqrt{\text{SIZE}(E)})$, and each vertex in $F$ transmits in another phase. Assign every vertex $f$ in $F$ the number of the phase in which $f$ transmits.
For $0 \leq i < |Z|-1$, $f_{|Z|-i}$ has at least $|Z|-i-1$ neighbors that when the algorithm assigned the phase $i$ to $f$, were hearing each in another phase from $i$ to $|Z|-i-1$ that is why the algorithm did not assign to $f_i$, a phase with a number lower than $|Z|-i$. Let us count the edges in the graph. We count $|z|$ for the $|z|$ neighbors (the set $F$) of $v_i$, and we count $(|z|-i-1)$ for the edge emanating from $f_{|Z|-i-1}$. Each edge is counted no more then twice (once for each endpoint of the edge) so the total number of edges is at least 

$$\frac{1}{2} \cdot \frac{1}{2} |z| (|z| + 1),$$

which is greater than $\text{SIZE}(E)$ the total number of edges in the graph. Thus a contradiction to the assumption that no phase can be assigned to $v_i$. 

Q.E.D.

7. A TIME DIVISION SINGLE SOURCE BROADCASTING PROTOCOL

The following broadcasting protocol is based on the results of the previous section. The time is divided into fixed size slots. Slots are assigned to nodes by the greedy algorithm. The slots are collected into cycles so that a transmitting node may transmit again on the same slot (assigned to it by the greedy algorithm) in the consecutive cycle. Figure 7.1 gives an example to the slot assignment for the broadcasting protocol. The protocol assumes that the assignment information is calculated centrally at the time of initialization, and distributed between the nodes. Broadcast is done from a given source which is at the top of the hierarchy- the tree created by the greedy algorithm, created by the greedy algorithm.
Figure 7.1: Parallel transmissions using Spatial Reuse
Box number i (1 or 2) - packet number i.

Protocol Single Source Broadcast (*Node i knows it's father's id*)

(*Source node transmits when needed*)
Message received from father;

(*A message for broadcast has arrived*)

Slot assigned to me;

(*The assignment as done by the greedy algorithm*)

procedure Transmit (message)

(*Insert Node i identification in the header and transmit*)

begin

on Message arrived from father do

begin

wait until Slot assigned to me;

transmit (Broadcast message)

end

end

The following properties hold for the Single Source Broadcast (TD) protocol

1. The broadcasted message propagates on a tree, rooted at the source. The transmission of a father in the tree never collides in its sons.

   See theorem T6.2.

2. The number of slots, $S_i$, is bounded by $S_i \leq \min \{ K + 1, 2 \cdot \sqrt{\text{size}(E)} \}$

   See theorem T6.2.

3. Let $R$ be the radius of the tree, i.e. the number of hops (the number of relays), that every bit of the message from the head of the hierarchy has to pass, on the longest pass in the tree (clearly $R < |V|$). Let $T$ be the length of timeslot. Assuming broadcast in this direction is given highest priority, the total delay is bounded by

   $R \cdot S_i \leq T$
Comment: Sender address in message is not needed if father's timeslot is known to sons. With the father's addresses being part of the message, the slot assignment does not have to be known, and furthermore when the source node receives a message to broadcast it can initiate it's broadcast in the timeslot following the message reception.

8. MULTIPLE SOURCE BROADCASTING

The Single Source Broadcast (TD) protocol discussed in the previous section guarantees that a broadcast initiated at the root node of the network will be propagated to all nodes with father's transmissions not colliding in their sons, and within a bounded time delay, over a tree rooted at the source. The Single Source Broadcasting (TD) protocol does not guarantee that transmissions by other network nodes not related to forwarding the root node initiated broadcast will not collide. In this section we generalize the greedy algorithm to provide phase assignments which allow every node in the network to broadcast it's own message so that multiple source broadcast is obtained.

We consider three basically different approaches to providing the broadcast capability to all nodes. In the first and simplest generalization of the Single Source Broadcasting (TD), section 8.1, all nodes are granted broadcast rights, but the tree remains directional as before so that successful transmissions are guaranteed only in the direction away from the tree root. This solution is the most economic in terms of the number of additional phases needed while being still suitable for centralized network configurations.

In the second approach, section 8.2, we remove the tree directionality so that the broadcast of any node can be relayed over the tree with no collisions, broadcasted message of any node can be guaranteed a bounded delay. However, since a single tree is used the routes chosen by the broadcast are not necessarily the best for all sources. We consequently close this section by the investigation of a broadcast protocol based on $N$ directed trees each rooted at one of the network
nodes. This solution is shown to be prohibitively expensive in terms of the number of phases needed, and consequently inefficient in terms of the delay required to accomplish a given broadcast.

8.1 Multiple Source Directed broadcasting under Single Source Broadcasting (TD)

For property 1 of the Single Source Broadcast to hold it is sufficient that nodes transmit only in the slots assigned to them by the greedy algorithm, while still of course, transmissions in the "son to father" direction may collide in the father. With this observation in mind the multiple source broadcast protocol is a combination of up tree transmissions (which are not necessarily free from collisions), and a collision free way down. The idea is to make broadcast initiated by a node other than the root make it's way up to the root and be broadcasted by it using the Single Source Broadcast (TD) protocol. Several attempts may be necessary to pass a message from son to father in the up tree stage.

To enable all nodes to initiate a broadcast, those nodes not assigned phases by the greedy algorithm the leaves of the tree must also be included in the phase allocation. The resulting change to the greedy algorithm:

- replace step (0) by:

(0) divide the channel into \( \text{Min}\{K+2, (2 \times \text{SQRT} (\text{SIZE}(E)))\} \)

and

- replace the condition in the while statement of the greedy algorithm (section 6) by:

there is a vertex, \( v \), which is not phased.

We shall denote the resulting algorithm as "generous greedy algorithm".

Theorem 8B.1 (properties of the generous greedy algorithm)

(a) The generous greedy algorithm does not assign more than \( K+2 \) phases.

(b) The algorithm always stops successfully.

(c) The algorithm constructs a spanning tree such that the broadcast can cover the network by having each father in the tree transmit in it's phase.
The algorithm is polynomial (in execution time).

(e) The number of phases assigned is less or equal to \(2 \cdot \sqrt{\text{SIZE}(E)}\) where \(E\) is the set of edges in the graph and \(\text{SIZE}(E)\) is the number of edges in \(E\).

**Proof:** The only change in the proof (from that of 18.3) is in lemma 18.5. We replace it with lemma 18.1.

**Lemma 18.1:**

When there is a hearing node, \(v\), which is not phased, there is always a phase (out of the existing \(\min \{\lfloor K/2 \rfloor, 2 \cdot \sqrt{\text{SIZE}(E)}\}\)) which meets the condition in (4) and thus can be assigned to \(v\).

**Proof of lemma:** Suppose \(u\) is \(v\)'s father. \(v\) is forbidden to transmit in the following phases: the phase in which \(u\) transmits, if \(u\) is not the source - also the phase of \(u\)'s father, and maybe, the fathers of \(v\)'s other neighbors; \(v\) has, at most, other \(K-1\) neighbors (except \(u\)) - those neighbors hear, in at most \(K-1\) other phases, which may be forbidden for \(v\). So, \(v\) is forbidden to transmit in, at most, \(K+1\) phases, so we can assign it the \(K+2\)th phase. Q.E.D.

The following properties hold for a protocol implementing this approach:

1. Every node can send messages and initiate broadcasts.
2. Broadcast of nodes which are higher up in the hierarchy will usually have a lower delay by having to pass a shorter way up the tree which is not collision free.
3. Messages are passed down the tree from father to son with no collisions in the sons. If the broadcasted message has the highest priority then the delay of messages on the way down the tree is bounded by (almost) the same bounds \((K+2\) instead of \(K+1\)) given by the Single Source Broadcasting (TD).

Clearly the Directed Multiple Source Broadcasting (TD) protocol is not fair and guarantees delivery to the tree root node only. The protocol can be useful for strongly centralized networks and also presents a transitional step between...
the Single Source Broadcasting protocol of section 7, and the fair multiple source broadcasting protocol which guarantees delivery from all nodes, presented next.

6.2 Multiple Source Undirected Broadcasting (TD)

The Multiple Source Undirected Broadcasting protocol gives every node an equal opportunity to broadcast a message. The basic idea is to construct a tree in which the transmissions of sons do not collide in their fathers in addition to the property that transmissions of a father do not collide in its sons. The following Undirected Generous Greedy algorithm constructs that tree. The price to be paid is the addition of phases which in the case of time division increases the length of the cycle.

The Undirected Generous Greedy algorithm is derived from the generous greedy algorithm by the following change:

Replace step (0) by

(0)' divide the channel into \( \min\{2K, (2 \cdot \sqrt{\text{SIZE}(E)})\} \) phases.

Replace step (4) by

(4) assign to \( v \) the phase (out of those created in step (0)) with the lowest sequential number (of phase) which meets the following conditions:

(a) no phased vertex will stop hearing it's father (e.g., if \( v \) is a neighbor of \( u \), and \( u \) has a father, the algorithm should not assign the phase of \( u \)'s father to \( v \))

(b) no unphased vertex, which is, however, hearing, will stop hearing (i.e. if \( v \) is a neighbor of \( w \), and \( w \) hears only from one vertex, \( z \), which is, thus, the only candidate to be \( w \)'s father, \( v \) will not be assigned the phase of \( z \))

(c) \( v \) will not be assigned the phase of it's father

(d) \( v \) may not be assigned the phase of another son of \( v \)'s father

Notice that the change from the generous greedy algorithm is the addition of (d) in (4).
Theorem T8.2 (properties of the Undirected Generous Greedy algorithm):

(a) The Undirected generous Greedy algorithm does not assign more than \(2K\) phases, where \(K\) is the maximum number of neighbors a vertex in the graph may have.

(b) The algorithm always stops successfully.

(c) The algorithm constructs a spanning tree such that the broadcast can cover the network by having each father in the tree transmit in its phase.

(d) The algorithm is polynomial (in execution time).

(e) the number of phases assigned is less or equal to \(K+2\cdot \text{SQRT}(\text{SIZE}(E))\) where \(E\) is the set of edges in the graph and \(\text{SIZE}(E)\) is the number of edges in \(E\).

Proof: The only change in the proof from that of theorems T8.1 and T8.3 is in lemma L8.1 (or lemma L6.1). We replace lemma L6.1 by lemma L8.2:

Lemma L8.2:

when there is a hearing node, \(v\), which is not phased there is always a phase \(i\) out of the existing \(\text{Min}\{2K, (K+2\cdot \text{SQRT}(\text{SIZE}(E)))\}\) which meets the condition in (4) and thus can be assigned to \(v\).

Proof of the lemma: Suppose, \(u\) is \(v\)'s father. \(v\) is forbidden to transmit in the following phases: the phase in which \(u\) transmits, if \(u\) is not the source - also the phase of \(u\)'s father, and maybe, the fathers of \(v\)'s other neighbors: \(v\) has, at most, other \(K-1\) neighbors (except \(u\)) those neighbors hear, in at most \(K-1\) other phases, which may be forbidden for \(v\) if \(v\) is also forbidden to transmit on the bands the other neighbors of \(v\)'s father transmit. \(v\)'s father has at most \(K-2\) neighbors except \(v\) itself and except the father of \(v\)'s father (we have already counted the father of \(v\)'s father). So, \(v\) is forbidden to transmit in, at most, \(K+1+K-2=K-1\) phases, so we can assign it the \(2K\)-th phase.

Q.E.D.
**Protocol Multiple Source Undirected Broadcasting (TD):**

```plaintext
var Assigned_to_me: integer;

(*Assigned_to_me contains at node i the number of slot
 assigned to node i by the generous greedy algorithm*)

procedure Transmit(from the queue)

procedure Enqueue(buffer)

(*puts the message in the queue
 of messages to be sent in node i’s
 slot in this cycle or the following
 cycles. The rule of service in the
 queue is not specified*)

function Not.Empty(queue)

buffer My.message, Broadcasted.message

queue of buffers Queue_of_broadcasted_messages

event Slot(number_of_slot);

(*slot (number_of_slot) has arrived*)

I have a message to broadcast,

(*node i wants to initiate a broadcast*)

begin

  on slot(Assigned_to_me)
  
    if(Not.Empty(Queue_of_broadcasted_messages))
    then Transmit(Broadcasted_message)

  on Message.arrived

    Enqueue(broadcasted_message)

  on I have a message to broadcast

    Enqueue(My.message)

end
```
The following properties hold for the Multiple Source Undirected Broadcasting (TD) protocol

1. Every node can send messages and initiate collision-free broadcasts.

2. (a) The Undirected Generous Greedy algorithm constructs a tree-rooted in the source. The father of a node (decided in (3) in the algorithm) is its father in the tree. Each node in the tree is able to hear its father (i.e., it has no neighbors which transmit on the same band as its father, and neither will the father transmit on that band). (b) No two sons of the same father transmit over the same band so that the father will always hear a transmitting son.

Proof: (by induction on the order of the entrance of vertices to the tree)
When the tree includes only the source, the claim holds. Suppose till the j-th iteration, the phased vertices construct a tree-rooted in the source, and the father (according to (3) in the algorithm) of any vertex in the tree, is its father in the tree, and the father is, of course, a phased vertex too. Suppose, also, that no son of a vertex transmits over the same band as another neighbor of the father. Vertex transmits over the same band. In the j+1-th iteration we pick an unphased vertex, v, and thus, it is not part of the tree. In (3) we choose, as v's father, a phased vertex, and thus - a vertex in the tree. An edge from a vertex in the tree to a vertex which was not in the tree, can not close a circuit. The choice of the band for v is in that way, that a vertex which has a father still can hear its father, and a vertex is not assigned the same band as one of its father's neighbors (so the father can choose to hear it).

Q.E.D.

3. Let R be the radius of the tree, i.e., the number of hops (the number of relays), that every bit of the message from the head of the hierarchy has to pass, on the longest path in the tree (clearly \( R < |V| \)). Let T be the length of timeslot (and the length of a message). Assuming broadcast in this direction is given highest priority, the total delay is bounded by...
Where $\# \leq \text{Min}'\{(2K),(K+2\sqrt{\text{SIZE}(E)})\}$

It is important to realize that while all nodes are given a guaranteed service to broadcast a message these broadcasts are spread over a single tree. This tree will not necessarily be equally "good" for all source. Similar observations have been made with respect to spanning tree based broadcast algorithms in p.t.p. networks and reasons were given to motivate the use of only one tree\cite{WQ80}. Those reasons concern the dependencies between routes, and thus are even more strongly applicable to radio networks. We do not pursue the question of the "best" tree further.

### 8.3 Multiple directed broadcasting trees

In preceding sections we discussed broadcast protocols based on a single spanning tree. Since a single "best for everybody" tree may not exist, we investigate the penalty involved in constructing a tree for each vertex in the network. When there are $l$ trees, $l < K$, then according to the method we used before, we should forbid each vertex to use the phases of its brothers in each one of the trees: $L(K-2)$, the phase of its father in every tree $L$, the phases of the fathers of all its neighbors (their fathers in all the trees). $L_K$: The sum is given by the following expression:

$$L(2K-1)+1$$ \hspace{1cm} (8.2)

In order to construct a tree for each vertex in the network it is sufficient that we guarantee that whenever a vertex transmits, its neighbors are free to hear it. Thus, when $v$ transmits, its $K$ neighbors must not transmit, and the other $K-1$ neighbors of each of them (the $K$-th is $v$ itself), too must not transmit. The sum is given by:

$$K^2-K+1$$ \hspace{1cm} (8.3)

expression 8.3 is smaller than expression 8.2 for $l$, for which expression 8.4 holds:

$$l \geq \frac{K(K-1)}{2K-1}$$ \hspace{1cm} (8.4)
still, the value of expression \( 8.3 \) is \( O(k^2) \), e.g. in expression \( 8.1 \) reducing \( R \) half, may cost us the squaring of \( S \).

9. A FREQUENCY DIVISION SINGLE SOURCE BROADCASTING PROTOCOL

The Single Source Broadcasting (SSB) protocol can be revised by using frequency division instead of time division in a way which guarantees the same level of parallelism (i.e., spatial reuse) while lowering the maximum and average delay. The approach is analogous to reducing the store-and-forward delay in p.t.p. networks by cut-through switching. In other words, the frequency division is used to achieve the effect of packets of one bit length (and thus avoid store and forward delay) and also to avoid the need to wait for the assigned timeslot. The price to be paid instead is the addition of radio hardware. We assume that a node can begin relaying a message on band \( b_1 \) while receiving the same message on band \( b_2 \). The effect is that of cut-through switching; instead of store and forward since the beginning of the message can be send forward, while the rest of the message is still being received. In fact, several nodes on the same path of the message may be transmitting parts of it simultaneously, see figure 9.1.

In addition to reducing delay the frequency division approach also omits the need for clocks and time synchronization.

The following \( SSB(FD) \) protocol is aimed for broadcasting (sending messages to all the nodes), from a single source. That source is at the top of a hierarchy. For this protocol, only one receiver is needed per node. Bands are assigned by the greedy algorithm (see section 6) which is used to assign bands and not timeslots. We assume that the assignment information is calculated centrally and spread between the nodes at initialization time.
a) before initiation of the broadcast

b) the leading edge of the broadcast has reached v.

c) A cut through transmission

transmitter on incoming message without storing it.

Figure 9.1: Using frequency division to achieve relaying of messages without store and forward delay.

protocol Single Source Broadcast (Frequency Division) for phased node i:

("Source node transmits when needed") (*Node i knows it's father's id*)

event

Message received from father;

("A message for broadcast has arrived")
procedure Transmit(message)

(*Insert Node i identification in
the header and transmit on band
assigned to node i by the greedy
algorithm*)

begin

on Message.arrived.from.father do

transmit(Broadcast.message)

end

The following properties hold for the SSB(FD) protocol

1. The broadcasted message propagates on a tree, rooted at the source. The
   transmissions of a father in the tree never collide in the sons.
   (For proof see theorem T6.3)

2. The number of bands, \(b\#\) is bounded by\( b\# < \min\{ \frac{K+1}{2} \cdot \text{SQRT}(\text{SIZE(E)}) \} \)
   (See theorem T6.2)

3. Let \(R\) be the radius of the tree, i.e. the number of hops (the number of
   relays), that every bit of the message from the head of the hierarchy has to
   pass, on the longest pass in the tree (clearly \( R < |V| \)). Let \(\alpha\) be the max-
   imum propagation delay time between two adjacent nodes. Let \(T\) be the time
   that would have been enough for the source to transmit the message, had it
   used the whole channel (and not only part of it). The total delay is then
   given by:

\[
R \cdot \alpha + k \cdot T
\]  \hspace{1cm} (9.1)

\( R \cdot \alpha \) as each bit is send immediately and not delayed in a node. \( k \cdot T \) as infor-
   mation that could be transmitted (by the source) in \( T \) time units, (using the
   whole channel), requires \( k \cdot T \) time units to be transmitted on a band with
   width \( \frac{1}{k} \) of that of the full channel. \( R \cdot \alpha \) is the time passes till the whole net-
   work is participating simultaneously in the broadcast (see figure 9.1) and it
   can be looked upon as a payment for constructing a virtual circuit. In radio,
it is reasonable to assume that $T \gg a$, and so, the right term overrides the left term in the delay expression.

4. If the broadcast is given the highest priority then it is guaranteed to be received by all nodes. If the message is started to be passed on (from those which hear it) immediately, then it is guaranteed that the leading edge (the first bits) of every broadcast covers, in maximum speed, the tree rooted in the source node.

Proof
(by induction on the passing of the message in the tree) When a node transmits, it's sons can hear it (see theorem 7.6.3 (c)) and they start immediately transmitting it (see (a) in the protocol) if they have sons.

Q.E.D.

Comment: Sender address in message is not needed if father's band is known to sons. With the father's addresses being part of the message, the band assignment does not have to be known in advance, and furthermore when the source node has a message to broadcast it can thus initiate the broadcast on any band.

10. MULTIPLE SOURCE BROADCASTING- FREQUENCY DIVISION.

As with Single Source Broadcasting (FD), the replacement of time division by frequency division suggests a shortening of the delay. Again, the immediate translation of the multiple source broadcasting protocols to frequency division is the assignment of bands instead of timeslots. Here, however, nodes are required to listen simultaneously over the bands of all their neighbors in the tree (father and sons). If this requirement can be met, then multiple broadcasting trees (one per node, see 5.3) can be used more efficiently than in time division. The reason is that in contrast to the situation in time division, with frequency division arrived message can be sent on one tree immediately without waiting for the end of another message on another tree. If the requirement of listening on the bands of all the neighbors is not cost effective, it can be replaced by the
following two requirements:

**Requirement 1**: Each node, $i$, is able to choose to receive on its father's band even when the father's transmission starts while $i$ is busy receiving on a different band. (this can be accomplished, for example, by having two receivers—one of them reserved for reception on the father's band).

**Requirement 2**: A node, when not receiving any transmission, does not have to know in advance on which band it will receive a signal. That is, when a node starts to receive a signal it can sense it and adapt itself to the band. This operation (sensing and adaptation) can be performed within a few bits [1E80]. We call the sensing and adaptation mechanism the "homing mechanism."

In 10.1 we use these two requirements to construct multiple source broadcasting protocols in which the messages flow over a tree in such a way that the way down the tree (i.e., away from the root) is done with no collisions of father's transmissions in its sons, but messages can be lost on the way up the tree. In 10.2 we achieve a multiple source broadcasting protocol in which no collisions of sons' messages occur in their father; however when several sons compete on the attention of the father only one of them wins it.

10.1 Multiple source broadcasting protocols with one way guaranteed

In this section two protocols are discussed. One is the adaptation of Multiple Source under the Single Source Broadcasting protocol given in 8.1 and the other is the adaptation of the Multiple Source Undirected Broadcasting protocol given in 8.2. As we shall see both adaptations yield a directed protocol.

In the adaptation of the Multiple Source under Single Source Broadcasting protocol we use the Generous Greedy algorithm of 9.1. The way down the tree is the same as in the Single Source Broadcasting (F8) protocol, and the way up uses a channel access protocol which resolves collisions. The repetition of a broadcasted message by the father can be used as an acknowledgement by one of the sons, and to warn the other sons not to transmit. If however two, sons start to transmit together (within a time interval which is shorter than the propagation
delay from one son to the father plus the propagation delay from the father to the other son) a collision occurs. There may be a collision even if the two sons transmit over different bands, as requirement 2 demands only that a node be able to adjust its receiver to a band when it is not engaged in reception. In other words, there is no requirement of a node being able to choose between two signals which their leading edges arriving simultaneously.

This last observation is the reason why the adaptation to frequency division of the Multiple Source Undirected Broadcasting protocol does not guarantee freedom from collisions of sons' transmissions in fathers provided they do not have enough receivers— one per neighbor. However, such adaptation can limit the number of collisions to the resolution of the homing mechanism (see requirement 2).

The following properties hold for a local implementing this approach:

1. Every node can initiate broadcasts.
2. (a) The initialization algorithm creates a tree rooted in the source. The father of a vertex (decided in (3) in the algorithm) is its father in the tree. Each vertex in the tree is able to hear its father (i.e. it has no neighbors which transmit on the same band as its father, and neither does the vertex itself transmit on that band).

   **Proof:** See Theorem 7.5.3 and the proofs in property 2 of the Multiple Source Undirected Broadcasting protocol.

3. If a broadcast has the highest priority then the message sent down is guaranteed that its leading edge (the first bits) to cover, in maximum speed, all the subtree rooted in the node that initiated the broadcast.

   **Proof** Same as for property 4 in SSB(FD).

4. Let $R$ be the radius of the tree, i.e. the number of hops (the number of relays), that every bit of the message from the head of the hierarchy has to pass, on the longest path in the tree (clearly $R < |V|$). Let a be the
maximum propagation delay time, between two adjacent nodes. Let $T$ be the
time that would have been enough for the source to transmit the message,
if it had used the whole channel, and not only a band. The total delay is

$$R \alpha + k T$$

(10.1)

$R \alpha$ as each bit is sent immediately and not delayed in a node. $k T$ as infor-
mination that could be transmitted (by the source) in $T$ time units (using the
whole channel), requires $k T$ time units to be transmitted on a band with
width $\frac{1}{k}$ of that of the full channel. $R \alpha$ is the time that passes until the
whole network is participating simultaneously in the broadcast (see figure,
9.1) and it can be looked upon as a payment for constructing a virtual cir-
cuit. As was mentioned, in radio it is reasonable to assume that $T \gg \alpha$, and
so, the right term overrides the left term in the expression for the delay.

Same as property for 4 in of SSB(FD)

10.2 Multiple source broadcasting protocol with bidirectional freedom from colli-
sions

We start by adding the following requirement:

- **Requirement 3**: Each node is able to choose, and hear at least one of several
  simultaneous transmissions, even when their leading edges coincide.

This requirement solves the problem of adaptation of the Multiple Source
Undirected Broadcasting protocol (mentioned in 10.1). When several sons of the
same father transmit on the band assigned to them by the Undirected Generous
Greedy algorithm, their father can always choose to hear one of them and no col-
lossions occurs. However, the transmissions of the sons not chosen will be lost. If
the father always immediately repeats the broadcasted message then only small
parts of the sons' messages are lost as they can hear the father transmitting
another message, and thus can know their message is not accepted.

The following properties hold for a broadcast protocol implementing this
approach.
1. Every node can initiate broadcasts.

2. (a) The Undirected Generous Greedy algorithm creates a tree, rooted in the source. The father of a vertex (decided in (3) in the algorithm) is its father in the tree. Each vertex in the tree is able to hear its father (i.e., it has no neighbors which transmit on the same band as its father and neither does the vertex itself transmit on that band).

(b) No two sons of the same father transmit over the same band, and so the father can hear the son it chooses to hear (if there is no transmission of the father's father).

**Proof** same as property 2 of the Multiple Source Undirected Broadcasting protocol.

3. If a broadcast has the highest priority then the message sent down is guaranteed that it's leading edge covers, in maximum speed, all the subtree rooted in the node that initiated the broadcast.

**Proof** Same as property 4 in of Single Source Broadcast (FD).

4. On the way up the tree the arbitration of the attention of the fathers does not use control messages. When several sons are competing then there is always a winner and no part of the transmission of the winner is lost due to the competition.

5. If a broadcasted message is given the highest priority and each node can choose always to hear the highest priority message among those transmitted to it (can be regarded as requirement 4) then the message is guaranteed that its leading edge (the first bits) covers all nodes in maximum speed.

6. Let $R$ be the radius of the tree, i.e., the number of hops (the number of relays), that every bit of the message from the head of the hierarchy has to pass, on the longest path in the tree (clearly, $R < |V|$). Let $a$ be the maximum propagation delay time, between two adjacent vertices. Let $T$ be the time that would have been enough for the source to transmit the message, had it used the whole channel, and not only a subchannel. We now calculate
the total delay under one of the following conditions: 1. A message is given highest priority and it is the message of the head of the hierarchy, or 2. A node can choose always to hear the message with the highest priority (among simultaneously received transmissions), or 3. The message do not meet another message on its way Under any one of the above conditions, the total delay is $R \cdot n$, as no bit is delayed in a vertex, but it is send immediately $k \cdot T$ as information that could be transmitted (by the source) in $T$ time units (using the whole channel), requires $x \cdot T$-time units to be transmitted on a subchannel with width $\frac{1}{x}$ of that of the full channel. $R \cdot n$ is the time that passes until the whole network is participating simultaneously in the broadcast, and it can be looked upon as a payment of constructing a virtual circuit. As was mentioned, in radio it is reasonable to assume that $T \gg a$, and so, the right term overrides the left term in the expression for the delay.

11. TDMA/SR, FDMA/SR AND HIERARCHICAL ROUTING

Notice that the undirected generous greedy algorithm imposes on the network a spanning tree-like construction. We can consider this spanning tree as representing an imposed network - the reliable link allocation over which simultaneous and collision free transmissions can be achieved. To this end we define the neighbors of a node in the network to be the neighbors given by the spanning tree (father and sons). With this definition, a reliable link exists between any pair of neighboring nodes. Obviously, the tree structure obtained from intuitive broadcast considerations exists independently of the broadcast activity. The network protocols can thus be divided to layers, as is common in communication networks.

The link layer protocol as obtained from the tree derived from the Undirected Generous Greedy algorithm provides TDMA/SR or FDMA/SR i.e. time or frequency allocation so that the same timeslot (or the same band) can be simultaneously assigned to different nodes.
The routing is done by distributing the information over the tree using only those links required for reaching the destination node. Since the tree is rooted at a node which is not necessarily the source of the message, a basically hierarchical routing is achieved in which the message is passed up the tree via the first common father of the source-destination pair. Recall that hierarchical routing is a common method in existing radio networks [GVF76], [K78]. The unreliable links out of the tree can be used for a limited auxiliary mechanism.

An alternative way is to use the generous greedy algorithm and get a mixed network with unreliable and unreliable links. The hierarchical structure can still be useful for the routing as it uses reliable links at least one way (down the tree).

12. DISCUSSION

We have looked at radio-oriented solutions to the datalink and network layers protocols which overcome the difficulties caused by the lack of coordination and resulting competition between the link layer, the routing and the broadcasting protocols. We used a global view supplied by the broadcast activity. We have shown that an optimal solution, and also several natural heuristics are NPH, but approaches which yield polynomial algorithms can be had. We have widened the solution step by step to suggest an integrated approach to organize a radio network using channel division with spatial reuse, hierarchical routing and broadcasting. The proposed approach takes into account the broadcasting property of the radio, to obtain reliable links and to use their broadcasting property. It also enables the use of implied acknowledgments whereby the sender knows its message has been correctly received by hearing the neighbor’s relaying it. The responsibility of the routing is laid upon the receivers—whether to receive or reject, so that for example in the broadcasting protocols a node does not have to know who are its sons, or whether and who is currently listening to it. In broadcasting when the assignment is known, addresses of neighbors do not have to be used; thus additional bandwidth can be saved. The notion of priority can be imple-
mented easily in order to achieve bounded delay.

Under frequency division the delay is short, especially appropriate for short and urgent messages as the leading edge of the message reaches its destination with no store and forward delay (provided it has been granted highest priority).

An elegant way is suggested to use more than one receiver and one transmitter. The effect of narrowing the bandwidth due to the division (the bandwidth of a sub-band is, of course narrower than the full bandwidth) can be neutralized under spread spectrum techniques, such as frequency hop. This physical layer technique (intended, among other things, to deal with multipath [KGBK78]) narrows the bandwidth in order to overcome interference. A network using both the proposed protocols and frequency hop can be constructed in such a way that the bandwidth is narrowed only by the spread spectrum technique or only by the proposed protocols. In the frequency hop technique, each node changes frequency (subband) at short intervals, so that the effective bandwidth is only that of the subbands. To use spread spectrum techniques together with the proposed protocol we define phase as the waveform so that all nodes will use the same bands, but nodes assigned different phase will not use the same band at the same time.

The presented solutions need initialization, and thus are vulnerable to changes in the network topology. This however is part of existing solutions as well (see section 2).

Lastly note that the implementation of the suggested protocols in any protocol layer, does not depend on the implementations chosen for other layers. TDMA/SR (or FDMA/SR) can be implemented by themselves at data link level. Similarly Single Source Broadcasting protocol can be implemented on top of unreliable data link protocol. In this case of course bounded time delivery can not be guaranteed, but it at least ensures that the broadcast does not compete with itself, and, and the protocol thus remains valid as a heuristic for minimizing the average node consumption by the broadcast.
Random access methods are generally fit for networks with low traffic. The fixed divisions methods, TDMA/SR, or FDMA/SR provide stability the random access methods generally lack, and so do not deteriorate for high traffic situations. While a constant bound on how far from optimal does no exist, we can show that for trees the assignment is optimal.
APPENDIX

In this appendix we bring some additional results to the subjects discussed.

Section 5 and 6 (attempted approximations).

Claim CA.1: If $P = NP$, then there can not be a polynomial approximation algorithm $A$, to problem 4.1.1, which guarantees, for every special case, $G$:

$$\frac{A(G)}{OPT(G)} < \frac{3}{2}$$

Proof: If there were such an algorithm, we could use it to problem 4.1.2 exactly. If $A$ shows us a way to perform a broadcast in two phases, then of course there is such a way in the other direction. If there is a way to perform the broadcast in two phases, $A$ must give a solution of 2, as a larger solution, $B$ (even $B = 3$, has $B \geq \frac{3}{2}$). This contradicts the assumption that $P = NP$.

Q.E.D.

Corollary: If $P = NP$, then there is no polynomial approximation scheme for problem 4.

The good neighbor algorithm and part (a) in theorems T6.2, T6.3, T6.1 and T6.2

Claim CA.2: For every $K$, there exists a graph $G$ in which the maximal degree of any vertex is $K$, but $K$ phases are not enough to perform a broadcast in it with no collisions of fathers' transmissions in their sons.

Proof: We use the following construction: a source vertex, $s$, connected only to one vertex, $f$. $K-2$ vertices: $v_1, v_2, \ldots, v(K-2)$, are connected to $f$. Each $v(i)$ has a vertex $u(i)$, connected only to it. Every two vertices, $v(i)$ and $v(j)$, $i \neq j$, have a vertex $w(i,j)$, connected only to $v(i)$ and to $v(j)$.

Apparently each $v(i)$ must transmit, as it has a vertex $u(i)$, which will not hear otherwise. Each $v(i)$ is forbidden to transmit in $s$'s phase, $f$'s phase, and other $K-2$ phases of the other $K-2$ $v(i)$'s.

Q.E.D.

Notice that $K$—the maximum outgoing degree of a node in a network can be reduced by reducing the transmission power, and thus the transmission range.
Such a reduction may invalidate the source connectivity property of a network. Resuming connectivity can be done by placing repeaters (often used to increase connectivity [GVP76]) arranged in a tree. The branches of the tree can lead from the source connected subnet, to the other components of the network. It is easy to see that three phases are enough to cover an undirected tree. Alternatively, if the constructors of a network are free in positioning the nodes, the positioning can be done in a grid-like geometrical arrangement. In such an arrangement as long as the transmission radius is greater than the distance between grid pixels, the network stays source connected.

Incoming-degree bounded directed graph.

Claim CA.3: For every $C>0$ and for every $K\geq 2$, there exists a directed graph, $G$, in which the incoming-degree does not exceed $K$, and still it cannot be covered by less than $C+1$ phases, if a vertex is permitted to transmit only in one phase. Given $C$, let us construct the following construction. A source, $s$, is connected by edges directed from it to $C+1$ vertices: $v_1,v_2,...,v_{(C+1)}$, and every such $v(i)$ is connected to a vertex $u(i)$ (different $u(i)$ for each $v(i)$), by an edge $v(i)\rightarrow u(i)$. This edge is the only one entering $u(i)$. Every two $v$'s $v(i)$ and $v(j)$, $i<j$, have a vertex $w(i,j)$, two edges are entering it - one from $v(i)$, and the other from $v(j)$. Apparently the incoming degree is less or equal to two.

We prove the claim by induction on $C$. The proof for $C=2$, is clear from the construction. Assume the claim is true for $C=N$, and assume that every two $v$'s transmit in different phases. We take the construction for $N$, and add new vertices $v(N+2)$, $u(N+2)$, and the needed $w$'s, and we connect them as described. If we assign $v(N+2)$ the phase $v(i)$ is using, then $w(i,N+2)$ can not hear. If we try to solve it by changing $v(i)$'s phase, we have the same situation as before. We can not find a phase for $v(i)$, just as we could not find for $v(N+2)$. 

Q.E.D.

Claim CA.4: Given a directed graph, in which the incoming degree of each vertex is no more than 1 (except the source), it is possible to cover it by 3 phases.
Proof: According to the connectivity assumption, the above defines a directed tree which obviously can be covered by three phases.

Q.E.D.
REFERENCES


[E79] Even, S. Graph Algorithms, Computer Science press. 1979


Simulation Conference, 737-48 part ii, 4-6 December, 1978, Miami Beach, FL, USA.


