INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

Université d'Ottawa • University of Ottawa

Broadcasting in Wireless Networks

Mahtab Seddigh

A thesis submitted to the Computer Science Department
in partial fulfillment of the requirements
for the degree of

Master of Computer Science

April 2001

Ottawa-Carleton Institute for Computer Science
The School of Information Technology & Engineering
University of Ottawa
Ottawa, Ontario
K1N 6N5 Canada

0-612-67210-7

Acknowledgements

I would like to thank Dr. Ivan Stojmenovic for his valuable guidance, expertise, and support in supervising my thesis work. I owe a debt of gratitude to my parents for the many sacrifices they have made and for their love and support all through my career. I also want to thank my dear husband Nabil for being a constant source of inspiration, guidance and patience.

Abstract

In a multihop wireless network, each node has a transmission radius and is able to send a message to its neighbours (nodes located within its transmission radius). A message broadcasted by a source node will be received by all the nodes in the network either directly or indirectly via other intermediate nodes. Typically, broadcasting has been studied in literature in the context of flooding which consumes considerable network resources. Other techniques suggested for broadcasting include clustering, which reduces the number of rebroadcasts but involves substantial communication overhead for maintenance of the clustered structure of nodes.

This work examines the problem of providing reliable broadcast delivery of messages with minimal communication overhead in wireless networks. We consider two types of communication for broadcasting: one-to-all (the classical form of broadcasting where each node forwards the message to all its neighbours with one transmission) and one-to-one (each transmission forwards the message to only one neighbour at a time). We propose to reduce the communication overhead involved in both these forms of broadcasting by applying the concept of internal nodes. We simulate and analyze the performance of a family of three algorithms we propose for the one-to-all model using internal nodes. Neighbour elimination is used to reduce redundant rebroadcasts, and reliability is almost guaranteed with the introduction of RANA (Retransmission after negative acknowledgement). For the one-to-one model, in addition to applying the internal nodes concept, three additional algorithms P-broadcast, IP-broadcast and PI-broadcast are proposed based on the concept of planar subgraphs such as RNG (relative neighbourhood graphs). The one-to-one experimental results show a reduction in the number of retransmissions to about 50% for sparse networks and about 5% for dense networks. We compare the relative performance of these proposed approaches against those presented in literature. The criteria used for the performance comparison includes: reliability (reaching all nodes in the absence of message collisions) and savings in rebroadcasting. The other features of the proposed algorithms are their localized and parameterless nature. The one-to-all experimental results show improvements in reachability of up to 20% for sparse networks with corresponding savings in rebroadcasting.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Thesis Background and Motivation

Wireless networks consist of static or mobile hosts (or nodes) which can communicate with each other over wireless links without any static network interaction. Each mobile host has the capability to communicate directly with other mobile hosts located within its transmission radius. However, if two hosts who want to communicate are outside each other's transmission radii, they can only do so if other intermediate hosts forward packets for them. Examples of such networks include ad-hoc local area networks, packet radio networks, and sensor networks. These networks find applications in situations such as disaster rescues, wireless conferences in the hall, battlefields, or monitoring objects in a possibly remote or dangerous environment.

In such wireless networks, there's often a need for a node to broadcast a message to every other node in the network. Two models of communication can be used for broadcasting: *one-to-all* and *one-to-one*. In the *one-to-all* model, the message transmitted by a node can reach *all* the nodes located within the radius of the sending node. In the *one-to-one model*, a sender directs its message toward only *one* of its neighbours. This form of broadcasting is used when narrow beam directional antennas are used, or when a node receives messages on its own frequency, and each node is aware of the frequency of each of its neighbours.

Most of the studies on broadcasting focus on the one-to-all model [BMSU, PR, NTCS, PL, QVL]. The most basic means of broadcasting is to use a flooding technique. In flooding, the same message is retransmitted by *all* nodes that receive it, and on *all* outgoing edges (except the edges on which the same message was already received, possibly several times). Hence, a node always forwards the packet to its neighbours, regardless of whether or not the neighbours have already received the data from another source. Flooding is used for route discovery, for example (see [BMJHJ]). Other applications in the context of wireless networks include paging a particular host or sending an alarm signal.

While Flooding guarantees delivery of the message to all the nodes in the network with shortest delay (as the message is propagated along the shortest path from source to each destination), it does so at a high cost since it introduces a large number of duplicate messages. Flooding consumes considerable network resources which are already scarce in wireless networks eg. battery power and bandwidth. Further, it leads to serious congestion, redundancy, contention and collision in mobile wireless networks referred to as the *broadcast storm problem* [NTCS]. As a result, flooding cannot be used in its "pure" form to broadcast a message. Efficient broadcast techniques that address reliability, redundancy, bandwidth and energy-efficiency need to be developed.

1.2 Existing solutions

Several broadcasting methods exist that propose different approaches to reduce the redundancy of rebroadcasts and the communication overhead associated with flooding. Broadcasting in the one-to-all model is studied by [L, BMSU, PR, NTCS, PL, QVL] to name a few, while [BMSU, HKB, SK] study the one-to-one model.

Pagani and Rossi [PR] proposed to replace flooding by a method where each clusterhead and gateway (or border, as renamed in this work) node in a clustered wireless network forwards the message exactly once. There is considerable communication overhead involved in creating and maintaining a clustered structure. In the broadcasting algorithm of [HKB], the full message is sent exactly once to each node, while short messages (which inform nodes about existence of full messages) are flooded.

The broadcasting algorithms in [NTCS] achieve high ratio of nodes receiving the message with reduced amount of rebroadcasting in the one-to-all model. However, they do not guarantee delivery. Each of the methods from [NTCS] has a parameter whose best value may depend on particular network conditions, which is global information.

The current algorithms reviewed attempt to address the different issues faced with broadcasting in wireless networks. However, it is unlikely that any one broadcasting method is the best approach for all the networking scenarios. In chapter 2, we further review some relevant existing algorithms and in chapters 5 and 6 compare these results with that obtained for algorithms proposed in this work.

1.3 Thesis Objectives and contributions

The objective of this work is to develop efficient broadcasting algorithms that will improve broadcasting for wireless networks. Specific criteria include:

- Guaranteed delivery &
- Minimal communication overhead.

This thesis proposes to restrict broadcasting to internal nodes only, since any other node is directly linked to an internal node. We will show that internal nodes offer more savings in terms of retransmission over clustering. In addition, the communication overhead of maintaining internal nodes is much smaller than that for clustered nodes which has demanding maintenance procedures.

In addition to using internal nodes, the broadcasting algorithm for the one-to-one case proposes to reduce the number of message retransmissions by limiting the retransmissions to edges that belong to the relative neighbourhood graph (RNG), or applying both these concepts.

We also introduce the concept of *neighbour elimination* to reduce the number of unnecessary rebroadcasts. An internal node will retransmit a message only if it has a non-eliminated neighbouring node who is either a non-*internal* node assigned to it or an internal node.

The broadcasting algorithm for the one-to-all model almost guarantees delivery with introduction of RANA (Retransmission after negative acknowledgement). If the receiving node does not receive the message in full (due to collision), it requests the sender to resend the message. This scheme ensures that nodes who did not receive the message in the first pass, still stand a chance of receiving it.

The proposed broadcasting algorithms are localized, since each node only needs the information from its one-hop neighbours, and possibly two-hop neighbours. They are also degree independent in two senses: there is no parameter in the algorithms that is set according to the network average degree *d*, and the performance of the proposed algorithms appears to be relatively stable with respect to *d*.

In short, the broadcasting algorithms proposed in this thesis have the following features:

✓ Restrict broadcasting to internal nodes

✓ Reduce number of unnecessary re-broadcasts

✓ Guarantee delivery of message to all nodes

✓ Are localized and parameterless in nature

The two joint papers [SS], [SSZ] with the above contributions have been presented and published in conference proceedings. An additional two papers on one-to-one and one-to-all broadcasting have been submitted for publication in Journals.

1.4 Analysis

The broadcasting algorithms presented in literature are, in most cases, evaluated by using a discrete event simulator on certain kinds of graphs, with particular values (eg. location based parameters, map size, packet size, slot time). To be able to perform a fair comparison with these results, we often modified our algorithms to generate results in terms of these values.

The topology of a multihop network can be represented using a graph $G = (V,E)$ where V is the set of hosts and E the edges connecting these hosts. For our simulations, we use *Unit graphs* as they best model ad-hoc networks. Hence, all nodes are assumed to have the same transmission range. There is an edge between two nodes, and they are called neighbours, if they are within each others transmission radii. To reach a node K hops away, we assume that the intermediate nodes will forward the message. The degree of a node u in such a graph is the total number of its one-hop neighbours.

The concept of Gabriel Graphs(*GG*) is also used for the simulations in the one-to-one model. To define a GG, consider a graph S, containing nodes u and v. Let $disk(u,v)$ be the disk with diameter (u,v). Then, the Gabriel graph $GG(S)$ is a graph in which the edge (u,v) is present if and only if the $disk(u,v)$ contains no other points of S. $GG(S)$ is a planar graph (that is, no two edges cross each other).

For one-to-all broadcasting algorithms, we have developed a simulator using the C-language. A simplified version of the MAC specification in IEEE 802.11 standard is implemented to simulate carrier sense multiple access with collision avoidance (CSMA/CA) behavior among hosts. Since the speed of broadcasting a message is significantly larger than the node mobility, we assume that nodes are static while broadcasting is in progress. Detailed analysis can be found in chapter 6.

1.5 Assumptions and Limitations

- One broadcasting task at a time is in the network, and there is none other message traffic while broadcasting is in progress.
- The transmission radius is the same for each node and is fixed.
- Every node can detect any duplicate broadcast messages it receives from its neighbours.
- In the one-to-all model, the wireless channel is shared by all nodes: when a node transmits a message, all its neighbours will "hear" the transmission.
- There are no obstacles between any two nodes: a message is received if and only if the distance between sending and receiving node is less than the transmission radius.
- In the one-to-all model, a collision occurs if a node is receiving a message simultaneously from several neighbors: collision also occurs at the transmitting node if the node receives a message while transmitting.
- In one-to-one model, each node receives messages on its own frequency or by using a directional antenna, therefore no message collisions occur.
- To apply the dominating set concept in both one-to-one and one-to-all models, each node either knows the exact location of each of its neighbors or knows the list of neighbors of each of its neighbors.

- To apply the RNG concept for the one-to-one model, each node needs to know the exact location of each of its neighbors, which is available from either GPS or by finding relative coordinates by measuring signal strengths.
- The algorithms are localized, meaning that each node does not know about the global network configuration beyond its one or two-hop neighbors.
- Since the speed of broadcasting a message is significantly larger than that at which nodes move, it is assumed that nodes are static while broadcasting is in progress.

1.6 Thesis Organization

This remainder of this thesis is organized as follows:

Chapter 2 is a literature review of related work. Chapter 3 describes the proposed broadcasting algorithms for the one-to-one model. The proposed broadcasting algorithms for the one-to-all model are presented in Chapter 4. Chapter 5 contains performance evaluations for the one-to-one algorithms proposed in chapter 3. Chapter 6 contains performance evaluations for the one-to-all algorithms proposed in chapter 4 and Chapter 7 draws conclusions and discusses future work

Chapter 2 Related Work / Literature Review

This chapter reviews recent wireless broadcasting algorithms proposed in the literature.

2.1 One-to-one broadcasting algorithms

The *SPIN* broadcasting algorithm described by Heinzelman, Kulik, and Balakrishnan [HKB] is used for sending a message from a node in a sensor network to all other nodes. *SPIN* uses the concept of long (*datum*) and short (*meta-datum*) messages. The long message is the actual *datum*, which could be all data that a particular sensor collects. The short message is the *meta-datum*, which has a considerably shorter length (e.g. 16 bytes instead of 500). The contents of the meta-datum could possibly be the sensor's *ID*, or a combination of variable names (e.g. (x,y)-coordinates of sensor, possibly rounded). When a node receives the datum that is being broadcast, it will forward a corresponding *meta-datum*. Any neighbouring nodes that did not receive the *meta-datum* yet will reply with a request to get the actual *datum*. Consequently, the original node will respond by sending the actual datum to all nodes that requested it. Suppose that node A has d neighbours and that it forwards the *meta-datum* (or actual *datum*) to k of these neighbours. The energy spent by node A to transmit is k* *bitsize* * 600mW (*bitsize* is 16 or 500). The energy needed by any node to receive such a message is *bitsize** 200mW. The experiments carried on a particular static sensor network and particularly selected parameters show 60% energy savings for *SPIN*, in one-to-one networks, and 80% for one-to-all case, with performance very close to the theoretically optimal one (which is the case when the actual data are sent without previous meta-data).

The power consumed by the SPIN protocol [HKB] is *(n-1)E + 2E'* where *n* is number of nodes, *E* and *E'* are mean powers consumed for sending long and short messages along one hop, respectively. In any broadcast scenario, the energy *(n-1)E* consumed is inevitable and is a lower bound that needs to be utilized.

Subramaniam and Katz [SK] proposed to construct broadcast graphs or broadcast trees, and use a spanning tree subgraph for broadcasting. The power consumed for sending messages using this approach is considerably reduced. However, the maintenance cost for the case of moving nodes appears to be significant, and the method may only be considered for static networks such as several kinds of sensor networks.

It was proved in [BMSU] that the intersection of a connected unit graph and a Gabriel graph (defined over the same set of points) is a connected planar graph. Bose, Morin, Stojmenovic and Urrutia [BMSU] proposed a broadcasting algorithm for unit graph G, which is based on finding a planar subgraph G″ of G (by intersecting G with Gabriel graph of the same nodes) and applying the algorithm of [BKOO] which enumerates all the faces, edges, and vertices of a connected embedded planar graph G. The algorithm requires no memory at the nodes of the graph and uses only O(1) additional memory in the packet that is traveling around the network. The algorithm works by defining a spanning tree on the faces of G and performing depth first search on this spanning tree in $O(n^2)$ time, where n is the number of vertices of G. Although no memory requirement at nodes is a desirable property for a fully distributed algorithm, the message complexity makes this algorithm quite expensive.

2.2 One-to-all broadcasting algorithms

Broadcasting is sometimes studied in the context of address serving [L] for hierarchically clustered packet radio networks. Each node has a hierarchical address used in routing the packet to the node. This hierarchical address is the sequence of encoding clusters, starting with the highest level and ending with the lowest level, in which the node resides. Address servers located in each bottom-level cluster (thus associated with each clusterhead) keep track of a node's hierarchical address. When a node wants to send traffic to another node, it queries the address server present in its cluster for the address of the destination node. To answer the query, the address server in turn may query other address servers. Address servers may also update each other when a particular node has joined or left its cluster. These searches and updates can use any of a variety of algorithms, including flooding, multicast along a spanning tree, and sending a packet directly to each address server. Flooding is reliable, but is vulnerable to clusterhead failures; searching by sending a request to each address server is reliable and fairly efficient[L].

Broadcasting has traditionally been performed using flooding. Ho et all [HOTV] argued that flooding can be a viable candidate for multicast and routing protocols in very dynamic ad hoc networks. Their simulation results show that even flooding is insufficient for reliable multicast when mobility is very high. By reducing the broadcast redundancy [NTCS, PL], one can reduce the packet loss due to contention or collision, and potentially enhance the reliability of broadcasting.

Pagani and Rossi [PR] described a broadcasting protocol for ad-hoc networks. It is based on a clustered organization of nodes. Nodes are divided into clusters, with one of them serving as the *clusterhead* in each cluster. Each clusterhead has a direct link to each of the nodes in its cluster. Thus, two nodes in the same cluster have a hop distance of at most two. In the broadcasting protocol [PR], the source node forwards the message to its clusterhead (CH), which then initiates the construction of a virtual spanning tree of all CHs (two neighbouring CHs are at hop distance two) by forwarding the message to all of them. More precisely, the message is sent to all neighbouring CHs which in turn forward it to their neighbouring CHs. Any node (ordinary node or a CH) does not forward duplicate copies of the same message. All CHs broadcast the message (with one transmission) to the nodes in its own cluster. The protocol [PR] is therefore similar to the address searching algorithm [L] that sends a request to each address server (that is, CH).

The broadcasting algorithms in [PR, L] use the *Lowest ID* clustering algorithm proposed in [EWB, GT] to create their clusters. The *Lowest ID* algorithm starts creating clusters by assigning every node in the network a distinct ID. Periodically each node broadcasts the list of nodes that it can hear (including itself). The procedure continues as follows:

- A node that only hears nodes with IDs higher than itself is a *clusterhead* (CH).
- The lowest-ID node that a node hears is its clusterhead, unless the lowest-ID node specifically gives up its role as a clusterhead (deferring to a yet lower ID node).

In Fig. 1, the Lowest ID algorithm assigns nodes 1, 2 and 4 as CHs.

A sophisticated maintenance procedure for cluster formation when nodes move are also described in [LG].

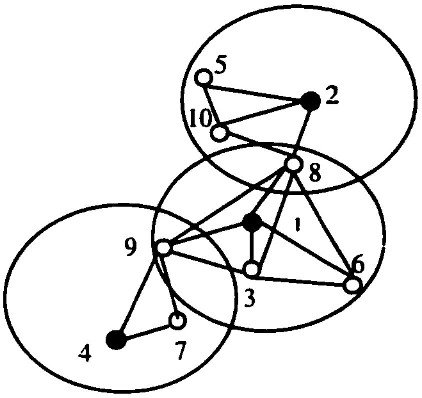

Figure 1 Clustering using *lowest ID* algorithm

The *lowest ID* algorithm is not always successful in fully clustering a given set. Gerla and Tsai [GT] describe a modified version of the algorithm proposed in [P] namely *Highest-connectivity* algorithm, in which the highest degree node in a neighbourhood becomes the *clusterhead*. These nodes are elected as CHs, and their neighbours are thus covered. The process then continues for the remaining uncovered nodes. An uncovered node is elected as a clusterhead if it has the highest degree among all its uncovered neighbours. Although the algorithm is expected to perform well on many randomly defined graphs (according to [GT]), it may not produce any CH for graphs which do not have any node with the highest number of neighbours (interval and triangular graphs, where almost all nodes have degrees two and three respectively). The *ConID* algorithm [SS] modifies the *Highest-connectivity* algorithm to address this case. It uses a combination of highest node degree and lowest ID (in case of a tie) to decide the clusterhead priority. The algorithm is described later in this section. The modification is based on the clustering algorithm [LG] which is described briefly below.

Lin & Gerla[LG] propose a distributed clustering algorithm which is initiated by each node whose ID is the lowest among all its neighbours (local lowest id nodes). Each of these nodes broadcasts its decision to create clusters (with them as CHs) to all its neighbours. Every node *Y* in the graph who *may* hear broadcasts by several of its neighbours (with lower IDs) proposing themselves as CHs, selects the node with the lowest ID, if any, from among these nodes. If there is a node *Y* such that none of its neighbours with lower IDs declared themselves as a CH, then *Y* assigns itself as the CH and broadcasts its ID as cluster ID, otherwise it chooses the neighbouring CH with the lowest ID and broadcasts its decision. Thus, every node broadcasts its clustering decision after all its neighbours with lower IDs have already done so. Hence, every node can determine its cluster and belongs to only one cluster. Each node transmits exactly one message during the algorithm.

ConID proposes a combined higher connectivity and lower *ID* clustering algorithm. Each node is assigned a pair $did=(d, ID)$, where d is node degree and ID is its ID, which will also be called clusterhead priority. Let $did'= (d', ID')$ and $did''= (d'', ID'')$. Then $did'>did''$ if $d'> d''$ or ($d'=d''$ and $ID' < ID''$). That is, a node has clusterhead priority over the other if it has higher connectivity or, in case of equal connectivity, has lower *ID*. The algorithm then follows the algorithm [LG], where lower *ID* clusterhead priority is replaced with higher *did* clusterhead priority described here.

The *ConID* algorithm produces reduced number of CHs and border nodes. Border nodes are nodes that have neighbours that belong to more than one cluster. In a cluster-based broadcasting algorithm, each CH and border node will transmit the message exactly once. Therefore, their total number needs to be optimized. For example, in Fig. 2 there are two CH nodes (*U* and *B*) and three border nodes (*P*, *Q*, and *R*) when *ConID* (with *x*-coordinate serving as node *ID*) is applied. The lowest-ID algorithm [LT] produces four CHs (*V, R, P* and the rightmost node) and five border nodes (all except top three). Note that the count applies only if broadcasting is initiated by a node that is supposed to retransmit by the method. Otherwise (that is, if non-internal node or a node that is not a CH or border node is the source for a broadcasting task), the count should be increased by one, for transmitting the message from the source to one of its neighbours that is selected for broadcasting by the method. However, to simplify the comparison, we ignore the possible count difference in our experiments.

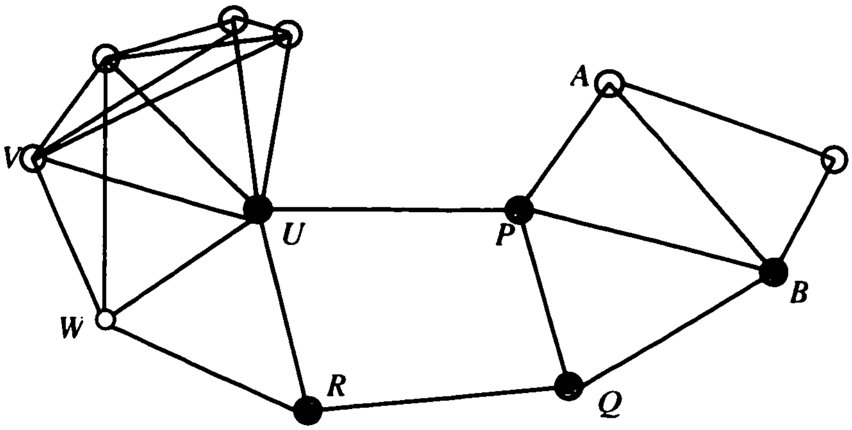

Figure 2 Internal Nodes- *Intermediate* nodes AVWBUPQR, *inter-gateway* and *gateway* nodes BUPQR

Ni, Tseng, Chen and Sheu [NTCS] studied the *broadcast storm* problem. A straightforward broadcasting by flooding is usually very costly and will result in serious redundancy, contention, and collision. Through analysis and simulations, [NTCS] proves the seriousness of the broadcast storm problem. [NTCS] proposes several schemes (probabilistic, counter-based, distance-based, location-based, and cluster-based) to reduce redundant rebroadcasts and differentiate the timing of rebroadcasts to alleviate this problem. These schemes achieve a high delivery rate with low number of retransmissions. However, they do not guarantee delivery.

In the probabilistic scheme, each node rebroadcasts the first copy of a received message with a given probability p. In the counter-based scheme, each node rebroadcasts the message if and only if it received the message from less than C neighbours. In the distance-based scheme, the message is retransmitted if and only if the distance to each neighbour that already retransmitted the message is $>D$ (the distance may be measured by signal power or GPS). In the location-based scheme (which requires GPS for its application), the message is retransmitted if and only if the additional area that can be covered if the node rebroadcasts the message (divided by the area of the circle with transmission radius) is greater than the threshold A. A simplified version of the method is to rebroadcast the message if the node is not located inside the convex hull of neighbouring nodes that already retransmitted the message. In the cluster-based scheme, *lowest ID* clustering algorithm [LG] is applied, and one of the above four methods is then applied on clusterhead and border nodes (that is, on the dominating set created by clustering the nodes).

Experiments in [NTCS] have measured reachability *RE* (that is, delivery rate, the ratio of nodes receiving the message), the saved rebroadcasts *SRB* (the ratio of nodes that do not rebroadcast the message), and average latency until the last host receives the message. One hundred nodes are placed in an area of varying size and fixed transmission radius R, resulting in graphs with varying average degrees. The optimal values for the parameters used for each scheme was dependent on the average node degree, which is an information of global nature and is not locally available to nodes. The location-based method, according [NTCS], is the best performing method that takes advantage of GPS. However, [NTCS] observed that the reachability in sparse graphs is unacceptable, even after clustering is applied first. *SRB* for low degree graphs seems to be low (compared to data obtained in this thesis), for *RE* in the range 80-90%. In this work, a higher SRB is obtained using reliable broadcasting methods.

Recently, Qayyum, Viennot and Laouiti [QVL] proposed a multipoint relaying method for efficient flooding in mobile wireless networks. They first define relay points for a given node (source or retransmitting node) *S*. A node is considered 'covered' if it receives (directly or via retransmissions by other nodes) messages originating at *S*. Relay points of *S* are those 1-hop neighbours of *S* that cover all 2-hop neighbours of *S*. That is, after all relay points of *S* retransmit the message, all 2-hop neighbours of *S* will receive the message. The goal is to minimize the number of relay points of *S*. The computation of a multipoint relay set with minimal size is a NP-complete problem [QVL]. The multipoint relay method [QVL] is a heuristic that works as follows. Let *N1(S)* and *N2(S)* be sets of one-hop and two-hop neighbours of *S* respectively. Further, let the relay points of *S* form the set *MPR(S)*. A small size set *MPR(S)* is computed by the procedure *relays(S)*.

Procedure *relays(S)*

1. Start with an empty multipoint relay set *MPR(S)*.

2. First select those one-hop neighbours of *S* which are the only neighbours of some node in *N2(S)*, and add them to *MPR(S)*.

3. While there still exist some node in *N2(S)* which is not covered by *MPR(S)* **do**

    a) For each node in *N1(S)* which is not in *MPR(S)* compute the number of nodes that it covers among the uncovered nodes in the set *N2(S)*.

    b) Add to *MPR(S)* the node for which the number if maximal (ties are broken arbitrarily).

Broadcasting from a node *S* is then performed as follows:

1. S broadcasts to all its neighbours, and computes *MPR(S)*. Initialize *retransmit(S) = MPR(S)*.

2. For each node *X* in *retransmit(S)* **do**

    - *X* retransmits the message and computes *MPR(X)* by calling procedure *relays(X)*.

- Add nodes from ***MPR(X)*** to the set *retransmit(S)*.

Note that each node *X* in *retransmit(S)* performs the computation (including retransmission) exactly once, by memorizing past traffic. Next, the list of nodes from ***MPR(X)*** may be added to the message, or alternatively, each node *X* informs each of its neighbours *Y* whether or not *Y* is in the relay set of *X*. The latter update is performed whenever node mobility causes topological changes in the network. In order to decide its set *MPR(X)*, node *X* needs to know the list of neighbours for each of its neighbours. In addition, node *X* reacts to any update from any neighbour by recomputing its relay set and informing all neighbours about their new status. Thus, each location update by node *Y* is followed by message updates by each neighbour *X* of *Y*, which is significant (but nevertheless local) overhead to the algorithm. The algorithm is reliable. It achieves high reachability and a high percentage of savings in terms of number of message retransmissions (see section 6.1)

Gerla, Kwon and Pei [GKP] proposed a combined clustering and broadcasting algorithm which involves no communication overhead for both the maintainance of the cluster structure and for updating the neighbourhood information. In their passive clustering algorithm, the cluster structure is updated with existing traffic by adding two bits to each outgoing message. We review the method for the case of a broadcasting algorithm initiated from a source. The source *S* will transmit the message to all its neighbours. *S* will declare itself a clusterhead (for the timeout period which is a parameter in the method) if it has no neighbouring active clusterhead. Each node *A* will act as follows upon receiving the message to be flooded to the whole network. *A* will declare itself a clusterhead using the same criterion as the source *S*. Otherwise, *A* will check the ratio of neighbouring clusterheads and neighbouring gateway nodes and declare itself a gateway if that ratio is above a certain threshold, which is also a parameter used by the method. If *A* decides to be a gateway, it will retransmit the message. Otherwise, *A* decides to be an ordinary node and does not retransmit the message. The performance of the algorithm depends on the two parameters mentioned whose best values are in accordance to network density and traffic load, which are generally information not available to nodes. Further, the method is not reliable. Moreover, there are pathological cases of poor delivery ratio. The authors report 35% savings in message retransmissions for flooding under certain network conditions. The absence of control overhead makes it a reasonable choice in case of significant node mobility, while in the case of low or no mobility the methods analyzed in this work have superior performance.

Peng and Lu [PL], independently with the conference version of this work [SS] (within ten days of each other) suggested the *neighbour elimination* scheme to reduce the number of rebroadcasts. In this scheme, a node does not need to rebroadcast a message if all its neighbours have been covered by previous transmissions. [PL] propose to let nodes with more neighbours rebroadcast earlier, so that more nodes can be covered by one transmission. They propose the following formula to calculate the rebroadcast delay. For node *U*, the delay time *Tmax* is calculated as follows: $T0 = \frac{1 + D\max(U)}{1 + D(U)}$, $Tmax = random(\Delta * T0)$, where *Dmax(U)* is the maximum degree of a neighbour of *U* (*U* is not taken into account), *D(U)* is the degree of node *U*, and is a small constant. *Random(x)* is a function that returns a random integer an interval between 0 and *x*.

The special case of broadcasting when all nodes are positioned on a straight line is studied in [SFLYO] in the context of inter-vehicle communications on a highway. In the TRADE protocol [SFLYO], each vehicle categorizes neighbouring vehicles into moving in the same and opposite moving, and vehicles not on the same road. A node (vehicle) that receives a broadcast message selects the furthest vehicle among its neighbours and includes the ID of that node in the retransmitted message. The vehicle that recognizes its own ID in the message is the next one to retransmit the message. In the DDT protocol [SFLYO], each vehicle that (re)transmits the message will include its own coordinates (available via GPS) in the message. When a vehicle receives the broadcast message, it computes a defer time inversely proportional to the distance from the node that transmitted the message. During defer time, each vehicle records the position of the neighbouring vehicle that transmits the same message. When the defer time expires, if most of its neighbours are covered by previous transmissions then retransmission is redundant and is dropped. Otherwise, the message is retransmitted. Interestingly, the (existing and new) two-dimensional broadcasting algorithms described in this work do not reduce to any of these simple algorithms when nodes are positioned on a straight line.

2.3 Internal nodes and Dominating sets

Let G be the graph that corresponds to a given wireless network. A set is dominating if all the nodes in G are either in the set or neighbours of nodes in the set. For example, clusterheads and border nodes in a clustered structure define a dominating set. Nodes that belong to a dominating set will be called *internal* nodes for G (of course, a different definition for dominating set leads to different set of internal nodes). Routing based on a connected dominating set is a frequently used approach [WL], where the searching space for a route is reduced to corresponding internal nodes. The routing process, in this approach, is divided into three steps:

- If the source node is not an internal node, it forwards the packets to one of its adjacent internal nodes. This internal node then acts as a new source to route the packets in the reduced graph consisting of internal nodes only.

- Eventually, the packets reach the destination internal node which is either the destination node itself or neighbour of the destination node.

In the latter case, the destination internal node forwards the packets directly to the destination node.

Such routing is suggested for shortest path and dynamic source routing [WL], which do not use location information in routing decisions. If the dominating set concept is applied, routing tables need to contain only information about internal nodes, and their size is therefore reduced. It is desirable, in this context, to create a dominating set with minimal possible ratio of internal nodes. However, the savings in the communication overhead in maintaining routing tables is not overshadowed by the overhead imposed in maintaining dominating set structure.

We observe that the clustering process is an example of creating a dominating set. The dominating set consists of all clusterhead nodes and border nodes, which are nodes that have neighbours from at least two clusters (that is, border nodes connect two CHs). The main drawback of using the cluster structure as a dominating set is its significant communication overhead for maintaining the structure in a moving environment. [WL] reviewed several existing dominating set definitions which have significant overhead in maintaining the structure and do not produce better ratios of internal nodes than the simple definitions given in [WL].

Wu and Li [WL] proposed a simple and efficient distributed algorithm for calculating the connected dominating set in a ad-hoc wireless network. They introduced the concept of an *intermediate* node. A node A is an *intermediate* node if there exist two neighbours B and C of A that are not direct neighbours themselves. [WL] also introduced two rules that considerably reduce the number of internal nodes in the network. Let $N(u)$ be the (open) set of all neighbours of node u, and let $N[u]=N(u) \cup \{u\}$ be the corresponding closed neighbour set, that is the set of all neighbours and *u* itself. Suppose that each node has a unique *id* (it may be obtained by generating a random number in [0,1], or their *x*-coordinate may serve the purpose). Let us define *inter-gateway* nodes as intermediate nodes that are not eliminated by Rule 1. Next, let the *gateway* nodes be those intermediate nodes that are not eliminated by both rules. Rule 1 [WL] is as follows:

- Consider two intermediate nodes v and u. If

  $$N[v] \subseteq N[u] \text{ in } G \text{ and } id(v) < id(u),$$

  then node v is not an *inter-gateway* node. In other words, if any neighbour of v is also a neighbour of u, and v is connected to u and has lower *id*, then any path via v can be replaced by a path via u, thus node v is not needed as internal node. We may also say that node v is 'covered' by node u. In Fig. 3, since $N[v] \subseteq N[u]$, vertex V is not an *inter-gateway* node if $id(v) < id(u)$.

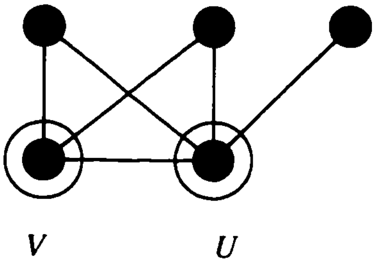

Figure 3 Rule 1 [WL] generates *inter-gateway* nodes

The number of internal nodes can be further reduced by applying Rule 2 [WL], as follows:

- Assume that, after applying Rule 1, u and w are two *inter-gateway* neighbours of a *inter-gateway* node v. If

  $$N(v) \subseteq N(u) \cup N(w) \text{ in } G \text{ and } id(v) = \min\{id(v), id(u), id(w)\}$$

  then node v is declared a non-gateway node. In other words, if each neighbour of v is a neighbour of u or w, where u and w are two connected neighbours of v, then v can be eliminated from the list of *gateway* nodes (when, in addition, v has lowest *id* among the three). The hop count between a source and destination node may increase by one in this process, since a segment *pvq* of the path between them is replaced by a segment *puwq* which is one hop longer. In Fig. 4, since $N(v) \subseteq N(u) \cup N(w))$, node *V* is not a *gateway* node.

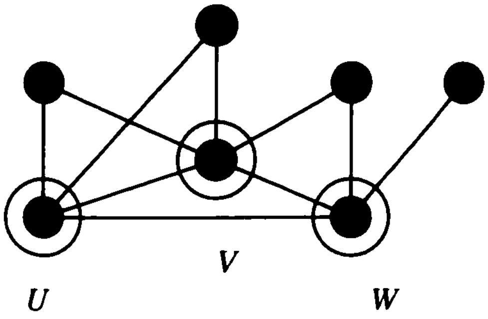

Figure 4 Rule 2 [WL] generates *gateway* nodes

If GPS is available, enabling nodes to know the location of all their neighbours, each node can determine whether or not it is an *intermediate*, *inter-gateway* or *gateway* node in $O(k^2)$ computation time (where k is the number of its neighbours), and without any message exchanged with its neighbours for that purpose. If GPS is not available, then the maintenance of internal node status requires each node to know the list of neighbours for each of its neighbours.

2.4 Relative Neighbourhood Graphs

The Relative Neighbourhood Graph (*RNG*) is a geometric and graph theoretic concept proposed by Toussaint [T]. The *RNG* can be defined as follows. An edge (u,v) exists between vertices u and v if the distance between them, $d(u,v)$, is less than or equal to the distance between every other vertex w, and whichever of u and v is farther from w. In other words,

$\forall w \neq u,v\colon d(u,v) \leq \max(d(u,w), d(v,w))$.

Thus, for an edge *(u,v)* to be included, the lune (intersection of two circles centered at *u* and *v* and with diameter *uv*) should contain no vertex w. In Fig. 5, *uv* is not in *RNG* because of the witness node w. Fig. 6 shows the *RNG* on a set of six nodes (in this case the *RNG* is a spanning tree, which is not always the case).

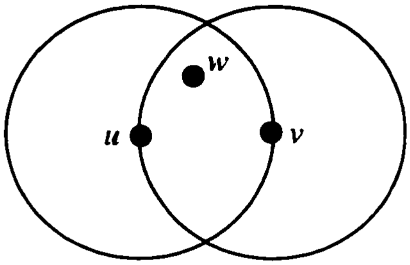

Figure 5 RNG graph

Edge *(u,v)* is not in *RNG* graph because of a witness w

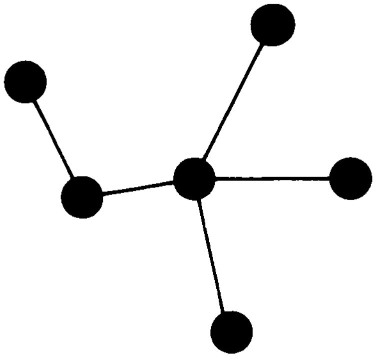

Figure 6 RNG graph

Toussaint [T] proved several important properties of *RNG* graphs, which are necessary for their application in broadcasting. *RNG* is a connected graph, and is a planar graph. The planarity of the graph assures that it is a sparse graph. A planar graph with *n* nodes can have at most *3n-6* edges[T].

Chapter 3 Proposed Broadcasting Algorithms for one-to-one model

In some applications, such as route discovery or paging, the message to be broadcast can be considered to be a 'short' one, whereas file sharing is an example of a 'long' message. Heinzelman et al [HKB] proposed the concept of short and long messages. Each node receives a long message exactly once, which is optimal. Thus, we will only try to optimize the number of short messages. In the flooding algorithm [HKB], the number of short messages is equal to the total number of edges in the network. We propose five broadcasting algorithms, for the one-to-one model, that use *RNG* and *internal dominating set* concepts to reduce the number of short messages.

3.1 Broadcasting using internal nodes

We propose new broadcasting algorithms separately for each of the one-to-one and one-to-all network (presented in chapter 4) models using internal nodes. In both cases, the use of internal nodes reduces the overhead of the broadcasting algorithms.

We first propose three internal node types *SP-intermediate*, *SP-inter-gateway* and *SP-gateway*, as an extension to the three types in [WL] namely *intermediate*, *inter-gateway* and *gateway*. A node is an *SP-intermediate* node if it is an intermediate node (connects two neighbours who are not directly connected) and in addition *also* provides the shortest path in terms of distance between its two unconnected neighbours. In Fig. 7, *A* and *B* are both intermediate nodes, but only node *A* is an *SP-intermediate* node because it provides a shorter path between neighbours *C* and *D* who are not connected directly. The *SP-inter-gateway* and *SP-gateway* nodes apply the same principle as *SP-intermediate* nodes on *inter-gateway* and *gateway* nodes respectively.

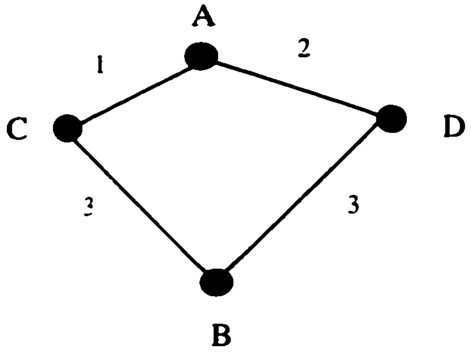

Figure 7 SP-intermediate nodes

Since any node in the network has an internal node neighbour, it suffices that only internal nodes retransmit the message. In *one-to-one* broadcasting, a node transmits a message to only *one* of its neighbours at a time. Messages are only sent on edges connecting two internal nodes (one message per edge). The number of short messages is then equal to the number of edges in the subgraph of internal nodes. Each non-internal node, knowing all its internal node neighbours, will choose one of them and inform that internal node to send all broadcast messages to it. The number of non-internal nodes is therefore added to the number of edges connecting internal nodes. This algorithm (algorithm 1) is referred to as the *I-broadcast* algorithm (that is, broadcasting restricted to Internal nodes). The additional (insignificant) overhead (in this and the four other algorithms proposed below in sections 3.2, 3.3 and 3.4) when the broadcasting source node is not an internal node is ignored in order to simplify the discussion and experiments.

For example, Fig. 2 shows a unit graph with 12 nodes. Eight of these nodes (those that are re-shaded and named) are *intermediate*, while five of them are *inter-gateway* nodes (they are also *gateway* nodes). The subgraph of five *gateway* nodes has six edges. Hence, the number of transmissions required for one-to-one broadcasting would be six plus one edge for each of the remaining seven non-*gateway* nodes. Thus, broadcasting based on the internal nodes concept, for the *one-to-one* model, will require 16, 13 and 13 message transmissions for the case of *intermediate*, *inter-gateway* and *gateway* nodes (respectively), compared to 23 in case of the full flooding algorithm [HKB].

3.2 Broadcasting using RNG

In case of unit graphs, the test for a given edge (whether or not it belongs to the *RNG*) may be applied to neighbouring nodes only. Since location update between neighbouring nodes is performed regularly for other reasons (i.e. routing), the selection of edges for the planar graph is local and requires no additional communication overhead. Edge (u,v) belongs to the constructed planar subgraph if and only if none of the common neighbours of u and v is located inside the mentioned lune. Let $U(S)$ and $RNG(S)$ be unit graph and relative neighbourhood graphs defined on the set of nodes S. [BMSU] proves that if $U(S)$ is connected then $U(S) \cap RNG(S)$ is connected.

We propose a second algorithm *P-broadcast*, a Planar subgraph based algorithm which constructs a planar subgraph (for instance, *RNG*) for all the nodes. The number of short messages in the algorithm is therefore equal to the total number of edges in the planar subgraph which is at most $3n-6$ for a network with n nodes. We propose to use *RNG* here because, to the best of our knowledge, it is the sparsest planar subgraph that can be locally defined. *RNG* is a subgraph of Gabriel Graph [BMSU] which is another planar locally defined subgraph that is more appropriate for use in routing with guaranteed delivery [BMSU], which prefers more edges. Gabriel graphs can also be used for broadcasting, but will result in fewer retransmission savings.

3.3 Broadcasting using both RNG and internal nodes

We propose two additional broadcasting algorithms *PI-broadcast* (algorithm 3) and *IP-broadcast* (algorithm 4) that combine *RNG* and internal node concepts as follows. The *PI-broadcast* algorithm applies the planar subgraph construction first, followed by the internal nodes concept on the resulting subgraph. The result is different from the internal nodes applied on the whole graph. The *IP-broadcast* algorithm changes the order of application compared to the previous algorithm. Internal nodes are first identified in the whole graph, following which the obtained subgraph (containing only internal nodes) is further reduced to a planar subgraph by the *RNG* construction.

3.4 Broadcasting reduction using Neighbour Elimination

A neighbour elimination scheme which was proposed in [SSZ, PL] for *one-to-all* networks may also be applied here. After node *A* receives a message from node *B*, *A* does not need to forward the message to any node *C* that is a common neighbour of *A* and *B* (except a non-internal node *C* that is associated with *A*). This will effectively eliminate one edge from any 'communication triangle'. In order to apply this to *RNG*, node *A* needs to know the location of all neighbours of its neighbours. Experiments show insignificant improvements when neighbour elimination is applied on *RNG* graphs, thus the neighbour elimination scheme is only considered for the internal node concept which we call *IN-broadcast* (algorithm 5). In this case, no additional information is needed, and the results for *IN-broadcast* shows results comparable to that for the *RNG*-only based scheme.

Chapter 4 Proposed Broadcasting algorithms for one-to-all model

4.1 Broadcasting using internal nodes

The concept of internal nodes is also used in the one-to-all broadcasting model. Although the concept of short and long messages from [HKB] can be applied to *the one-to-all* model, we believe that the overhead introduced by several neighbours acknowledging short messages, and the collision effect involved, is higher than if each internal node simply forwards the long message to all its neighbours once.

Since any node in the network has a neighbour that is an internal node, broadcasting can be performed by each internal node transmitting the message to all it's neighbours. In *one-to-all* broadcasting, when an internal node forwards a message, *all* it's neighbours receive the message. This is counted as one transmission. The comparison with the full flooding method (where each node forwards the message exactly once) is simplified, since it suffices to find the ratio of internal nodes in a network which represents the number of broadcasts in the network. Comparison with algorithm [PR] is also simplified, since the number of messages in algorithm [PR] is equal to the number of clusterheads and border nodes in the clustered graph. For example, Fig. 2 shows a unit graph with 12 nodes. Eight of these nodes (those that re-shaded and named) are *intermediate*, while five of them are *inter-gateway* nodes (they are also *gateway* nodes). Thus, broadcasting based on the internal nodes concept will require 8, 5 and 5 message transmissions for the case of *intermediate*, *inter-gateway* and *gateway* nodes (respectively), compared to 12 in the case of full flooding.

4.1.1 Improved broadcasting by neighbour elimination

A possible problem with the location-based [NTCS] and possibly all the other methods reviewed from literature is shown in Fig. 8. Node *A* receives a broadcast message from its only neighbour *B*. For the probabilistic, counter, distance or location-based methods proposed in [NTCS], *A* will rebroadcast the message even though it has no other neighbour that needs to receive the message. Thus, a good additional coverage area may often be empty. Even *lowest ID* clustering may treat *A* as a clusterhead (with 50% chance), thus imposing rebroadcast at *A*. Note that *A* is not an internal node, or a relay point for any other point (which may not always be the case). The neighbour elimination scheme proposed in this work is designed to prevent such nodes from rebroadcasting.

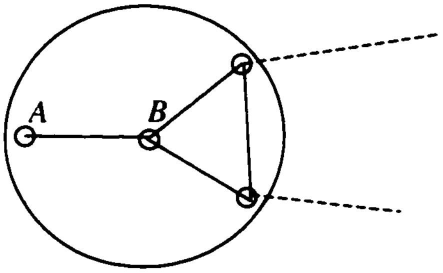

Figure 8 Problem with existing methods - Node *A* receives message from its only neighbour *B*.

The improvement caused by the neighbour elimination scheme is based on the observation given in Fig. 9. A node will re-broadcast the message only if it has a neighbour that might need the message. Thus, some neighbours are eliminated for re-broadcasting (they will, however, receive the message again if it is sent because of other neighbours). The neighbour elimination scheme works as follows:

- First, each node that is not supposed by the method to rebroadcast (ie non-internal nodes) at all, will assign itself to one of its possibly re-transmitting neighbours. In the internal node structure we propose to assign the neighbour as follows: each non-internal node *A* will assign itself to neighbouring internal node *B* which has the largest degree. In case of ties, the lowest ID (say, x-coordinate) is used among candidate neighbours. This rule attaches more neighbours to higher degree nodes thus possibly 'emptying' the assigned list of low degree internal nodes.

- Second, neighbours that received one of the copies of the message that arrived at a node *A* are also eliminated from the list of neighbours that might need the message. Consider, for example, node *A* in Fig. 9, which received the message being broadcast twice, from neighbours *B* and *C*. Neighbours *E* and *F* are eliminated from the broadcast list, since they received the same broadcast message from neighbours *B* and *C*, respectively. However, node *A* will, in this example, still re-broadcast the message because of neighbour *G* which is not 'covered' by *B* or *C*. The three circles around *A*, *B* and *C*, respectively, indicate the transmission ranges for these three internal nodes. Nodes *E*, *F* and *G* are either internal nodes (that is, belonging to dominating set), or non-internal nodes which are assigned to *A*. This scheme will further reduce the number of re-transmissions in broadcasting. Its efficiency may also somewhat depend on the MAC scheme that is being used.

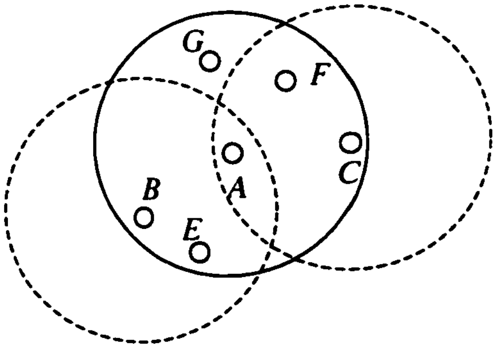

Figure 9 Neighbour Elimination - Node *A* eliminates neighbours *E* and *F* from its broadcast list

4.1.2 Retransmission after negative acknowledgements

In order to approach 100% RE, we have designed the *RANA* (Retransmission After Negative Acknowledgements) broadcasting algorithm. When a message is retransmitted by a node *A*, if a collision at receiving node *B* occurs within the first two slots of receiving the message, no retransmission request is issued. It is assumed that two slots are sufficient to identify the sender *A* and the message *ID*. Thus, if a collision occurs between the third and $p=127^{th}$ slot, the receiving node *B* can send a retransmit request to sending node *A*. We assume the retransmit request to be 4 slots long (it should suffice to include requested message *id* and the identity of the node *A* to whom the request is made, to avoid retransmissions from several neighbours). Node *A* will retransmit the message only if it received, without collision, all the four slots. The MAC protocol used is the same for both retransmit requests and message retransmissions. The retransmission request is cancelled if, while waiting to send a retransmission request, node *B* receives the same message without collision from another neighbour.

Chapter 5 Performance Evaluation for the one-to-one model

The performance of the five proposed broadcasting algorithms for the *one-to-one* network model was evaluated using random connected unit graphs with n=100 nodes. The algorithms attempt to minimize the number of edges used for (re)broadcasting. The percentage of retransmissions R is represented by the ratio $R = \frac{T}{E}$ where

T = Total # of edges used for message (re)broadcasting and
E = Total # of edges in the graph

Since our algorithms use only a portion of the original messages (short messages), without adding any new message, rebroadcast savings can be claimed without using the MAC layer in our experiments. The amount of collision is further reduced, and even better comparison (if the MAC layer were used) would have been obtained.

Table 1 shows the percentages of retransmissions for the proposed algorithms. The simulation was performed on each of the 6 internal node types. Due to the similarity of results obtained and space constraints, we only present here the results applying the *gateway* internal node. The results in Table 1 when compared with the worst possible solution, *flooding*, indicate that the proposed algorithms clearly outperform flooding which uses 100% of the edges in the graph vs. those presented in Table 1-which are just over 50%. For higher degree graphs, the planar graph concept (*P-broadcast* algorithm) alone leads to a significant reduction in the number of message retransmissions, e.g. only 23% of retransmissions were performed for degree 10 graphs compared to the flooding algorithm [HKB]. This improvement is obtained since *RNG* is a sparse graph (the average node degree was in the range 2.14-2.38 for n=100 nodes and for degrees as indicated in Table 1). The results indicate that the internal node concept is up to 10% less efficient than the *RNG* concept.

PI-broadcast showed no improvements over *P-broadcast*, while *IP-broadcast* lead to a merely 1% improvement over *PI-broadcast*. When the neighbour elimination scheme was applied to the internal nodes, the resulting *IN-broadcast* method showed performance comparable to the *P-broadcast* method, as indicated in Table 1.

Method/degree	4	5	6	7	8	9	10	15	20	25	30	35	40
P-broadcast	53	45	37	33	29	26	23	16	12	10	8	7	6
PI-broadcast	53	45	37	33	29	26	23	16	12	10	8	7	6
IP-broadcast	52	44	36	32	28	25	23	15	11	9	7	6	5
I-broadcast	61	52	47	41	38	35	32	21	15	12	9	7	6
IN-broadcast	53	45	37	31	27	24	22	21	10	8	7	6	5
SP-broadcast	50	40	33	28	25	22	20	13	10	8	7	6	5

Table 1 Percentages of retransmissions (n=100 nodes) for gateway nodes

It is obvious that the proposed algorithms perform much better than flooding (which is the worst solution). We now evaluate them against the best possible ideal solution for a given graph. For a graph with n nodes and an arbitrary node degree, the minimum number of edges needed to broadcast a message from one node to all the other nodes is n-1 which can be achieved by any spanning tree *(SP)* in the graph. The ideal algorithm is therefore referred to as *SP-broadcast* algorithm. The average degree in such a subgraph is $\frac{2(n-1)}{n}$ which is slightly under 2. The ratio of retransmissions for the *SP-broadcast* method with respect to flooding is given by $\frac{2n-2}{nd} \cong \frac{2}{d}$ which are shown in the last row of Table 1.

The average degree achieved by our methods is *pd*, where p is the percentage of retransmitting edges (expressed in [0,1]). Thus, the ratio of our method against the ideal *SP-broadcast method* can be obtained as $\frac{pdn}{2n-2}$. When compared to the ideal solution:

- The *P-broadcast* algorithm for instance, shows fairly good results. The percentage of additional messages, compared to the ideal solution, is 6-8%, 4-12%, 5-15%, 4-16%, 8-17%, 8-20% and 12-22% for n=10, 20, 30, 40, 50, 100 and 200 nodes respectively, and degrees d between 3 and 40. The higher number appears to be the limit for arbitrary degree, since the same percentage was obtained for any degree d between 10 and 40.
- The two best methods appear to be the *IP-broadcast* and *IN-broadcast* followed by *P-broadcast* and *PI-broadcast*.

Chapter 6 Performance Evaluation for the one-to-all model

The experiments were carried out in several phases. In the first phase (section 6.1), an ideal MAC protocol is assumed, which provides for collision free broadcasting. The number of nodes that re-broadcast the message is counted, and compared to the total number of nodes, to evaluate the savings. The second phase involves a real simulation using a MAC protocol. Three algorithms are proposed here. Algorithm 1 implements broadcasting alone, while algorithm 2 adds the concept of neighbour elimination to reduce the number of redundant broadcasts. Algorithm 3 further enhances Algorithm 2 with the introduction of RANA or negative acknowledgement which almost guarantees delivery of the message to all nodes.

6.1 Internal nodes vs cluster based and multipoint broadcasting

The results for clustering, multipoint relay and internal node methods are presented in Table 2 in terms of percentage of nodes that broadcast the message. The *intermediate* nodes concept does not achieve satisfactory gains. Gains for the *inter-gateway* method are comparable to the *LowestID* method. The *Gateway* nodes method is comparable to *ConID;* more precisely, it is better for degrees over 6 and somewhat worse for lower degrees. Interestingly, the ratios appear to be relatively stable with respect to degree *d*, for most methods. More precisely, ratios are quite stable for the clustering and *inter-gateway* methods, somewhat increasing (with increasing *d*) for the *intermediate* method, and notably decreasing for the *gateway* method. The *multipoint relay* method performs well, with savings between the *inter-gateway* and *gateway* methods. Thus, the *gateway* method appears to be the best overall method among those proposed in this work.

Method/degree	4	5	6	7	8	9	10
LowestID	67	66	61	63	67	59	64
ConID	50	46	51	55	51	52	54
Multipoint Relay	60	59	60	60	61	64	63
Intermediate	80	84	88	91	92	94	95
Inter-gateway	65	66	66	67	69	69	70
Gateway	60	57	54	51	50	48	45

Table 2 Percentage of broadcast nodes for n=100 nodes.

Recall that this comparison involves only retransmissions involved in direct broadcasting, and for the formation of the initial clustering structure. The communication overhead involved in maintaining clustered or internal node structures is not included. As noted before, the maintenance of a clustered structure (in the presence of moving nodes) is a non-trivial operation and may involve significant amount of message traffic. On the other hand, an internal node structure only requires communication with its neighbours when the topology changes, and only when GPS is not available. If GPS is available, no additional overhead (except for location updates, also needed for the clustering task) is incurred. Moving nodes pose an additional problem for clustered structure, by loosening the highest degree CH property, which effectively moves the performance of the *ConID* toward the *LowestID* algorithm unless a global re-clustering (which also means additional overhead) occurs. Therefore, the internal nodes concept does seem to perform broadcasting with significantly lower communication overhead compared to the existing methods based on clustering, even if the clustering process is optimized (*ConID*). Clustering-based broadcasting methods are therefore not evaluated further in this work.

6.2 Internal nodes vs location based broadcasting

We have developed a simulator using the C language. A simplified version of the MAC specification in IEEE 802.11 standard is referenced to simulate carrier sense multiple access with collision avoidance (CSMA/CA) behavior among hosts. Since the speed of broadcasting a message is significantly larger than that at which nodes move, we assume that nodes are static while broadcasting is in progress. The experiments were carried on random unit graphs, defined as follows. Each of n nodes is chosen by selecting its x and y coordinates at random in the interval $[0,m)$. We experimented with n=10, 20, 30, 40, 50, 100, 200 nodes, as in [NTCS]. The average node degree d, the transmission radius R, and the map size m are related to each other. In order to compare our results with those from [NTCS], and to comply with the IEEE 802.11 standard, we decided to fix the radius R to 500 meters. The map sizes from [NTCS] are equal to $s*R$, for s=1, 3, 5, 7, 9, 10 and 11. The corresponding average node degrees d are 96.5, 25.4, 10.4, 5.6, 3.5, 2.9, and 2.4, respectively.

It is assumed that one broadcasting task at a time is in the network, and there is none other message traffic while broadcasting is in progress (this is a fair assumption for comparing various broadcasting methods). We used the same parameters as in [NTCS]: the bit rate is 1M per second, the slot time is 20us (microseconds), the packet size is 280 bytes which, with required overhead, took 2536us in their simulation. Global synchronization can be achieved by adding some dummy bits so that the transmission takes an integer p=2540/20=127 number of slots. Acknowledgements are not sent. In this protocol, when a node A receives a packet to be transmitted, it first waits for an interframe spacing $DIFS$ period ($DIFS=2$ in our experiments). Node A then chooses a random integer BC (backoff counter) in the interval [0..31]. The backoff counter determines the number of transmission free slots as sensed by A. During periods in which the channel is clear, A decrements BC. When BC reaches 0, A transmits the packet. Once a node starts a transmission, it transmits continuously for p slots until the packet is fully transmitted. Thus, a neighbouring node receives the packet if it receives collision free transmissions for the duration of p consecutive slots.

[NTCS] uses performance metrics *RE* and *SRB*. We use the same metrics to facilitate the comparison of results:

- Reachability (*RE*):

$$RE = \frac{num\_recd}{num\_reachable}$$

where *num_recd* is the number of nodes receiving the broadcast message and *num_reachable* is the total number of nodes that are reachable from source (the graph may not be connected);

- Saved ReBroadcast (*SRB*):

$$\text{SRB} = \frac{(r-t)}{r}$$

where r is the number of nodes receiving the broadcast message, and t is the number of nodes that actually transmitted the message.

Method/map size	1 x 1	3 x 3	5 x 5	7 x 7	9 x 9	11 x 11
Intermediate	100	100	100	97	96	98
Inter-gateway	100	100	99	96	96	98
Gateway	100	99	97	94	95	98
Neighbor elimination	100	100	99	97	96	98
Location 0.1871	100	100	97	90	72	72
Location 0.0913	100	100	98	96	87	79
Location 0.0469	99	99	99	100	88	81
Location 0.0251	98	99	99	98	90	83
Location 0.0134	97	99	99	99	94	83

Table 3 RE for algorithm 1 and location [1] based methods

1: source [NTCS]

The location-based method is the best method among those presented in [NTCS], according to their experimental data. We will therefore compare our proposed internal node based broadcasting and neighbour elimination methods *only* with the location based method (see tables 3 and 4. The average number of reachable (*RE*) nodes from source were 100, 100, 99.8, 85.7, 36.1 and 11.3, respectively. We have listed numerical data from [NTCS] in our tables.

Method/map size	1 x 1	3 x 3	5 x 5	7 x 7	9 x 9	11 x 11
Intermediate	1	1	5	14	22	32
Inter-gateway	97	32	29	34	37	43
Gateway	98	76	54	45	41	45
Neighbour elimination	57	11	14	22	30	45
Location 0.1871	90	78	52	37	30	22
Location 0.0913	78	55	36	26	17	12
Location 0.0469	67	42	26	17	12	8
Location 0.0251	61	36	21	12	8	5
Location 0.0134	56	32	28	10	8	6

Table 4 SRB for algorithm 1 and location [2] based methods

2: source [NTCS]

We have also implemented a location based method from [NTCS] which eliminates from rebroadcasting, nodes contained inside a convex hull of neighbouring nodes that have already transmitted the same message. However, the results obtained are not competitive. Comparing the results obtained for algorithm 1 (tables 3 & 4) to those from [NTCS]:

- Among the internal node based methods, the *gateway* nodes method seems to provide the best combined values for *RE* and *SRB*.
- The *RE* values for the [NTCS] location-based method are comparable to the *gateway* nodes method for s=1, 3, 5, and 7, but are lower for s= 9 and 11.
- The *SRB* data are comparable to the *gateway* nodes method only for A=0.1871 and only for s= 3 and 5, and are lower otherwise.

Therefore, since a method is to be selected for arbitrary network density, the *gateway* nodes based broadcasting method is better than any of the methods proposed in [NTCS], and has no parameter associated with it. This is especially valid for networks with average degrees of nodes under 10, which is the range that includes practically all throughput efficient networks. The *RE* for the neighbour elimination-only method is comparable to that for the internal node based methods. However, it has lower *SRB* results than the *(inter)gateway* node concept.

The latencies of the various methods were also compared. The *gateway* based method, for instance, has lower latencies than any of the location based methods (for the various parameter values used). The differences were small for A=0.1871 but were increasing with increased value of threshold A.

6.3 Adding the neighbour elimination scheme

We now evaluate the performance of algorithm 2, which applies the neighbour elimination scheme to the broadcasting algorithm in Section 6.2 (Algorithm 1). Compared to Algorithm 1, the differences in the results for reachability were negligible (the *gateway* node concept, for instance, had improvements up to 1%). The *Intermediate* method benefitted significantly from the elimination scheme in terms of *SRB* and latencies (more precisely, the results became similar to those obtained for the neighbour elimination scheme alone), while the other methods had improvements in the 1-11% range. Table 5 shows the *SRB* results for the *(inter)gateway* node concepts enhanced by the neighbour elimination scheme. The latencies were very close to those obtained for the method where nodes retransmit upon receiving negative acknowledgements, presented next.

Method/map size	1x1	3x3	5x5	7x7	9x9	11x11
Inter-gateway	99	39	36	40	43	54
Gateway	99	81	60	49	47	55

Table 5 *SRB* for algorithm 2

6.4 Retransmission after negative acknowledgements

Algorithm 3, enhances Algorithm 2 by applying the concept of RANA. Here, the calculation of $SRB = \frac{(r-t)}{r}$ includes *all* transmissions; more precisely, t is the total number of message retransmissions. The retransmission requests are not counted since they are much shorter in length than the message itself. Compared to Algorithm 2, the reachability *RE* for the *(inter)gateway* node-based method has improved and was measured to be over 98.3% for all map sizes. The Neighbour elimination scheme alone also performed well, with over 97.3% *RE* for all map sizes. The *SRB* results have reduced by up to 10% as a trade-off for enhanced *(inter)-gateway* methods, as indicated in Table 6, while the neighbour elimination scheme alone did not give significant retransmission savings for half the map sizes.

Method/map size	1x1	3x3	5x5	7x7	9x9	11x11
Inter-gateway	99	28	26	34	41	53
Gateway	99	75	52	44	44	54
Neighbour elimination	57	1	4	16	27	44

Table 6 *SRB* for algorithm 3

Table 7 presents the measured latencies for Algorithm 3 for the *(inter)gateway* schemes, and for the location-based method with two extreme parameter values (latencies for other parameters are between the indicated values). The latencies are expressed in terms of number of p slot sequences, that is the number of slots needed for the last node to receive the message divided by p. This measure is chosen in order to compare latency with the message length, which is more illustrative than the time, which is proportional to the indicated values. Table 7 shows that the latencies for the *RANA* algorithm are still lower than latencies for the location based methods [NTCS].

Method/map size	1x1	3x3	5x5	7x7	9x9	11x11
Int.gateway	2.12	19.85	14.73	16.64	12.04	5.03
Gateway	2.21	9.18	12.20	15.85	11.66	4.97
Location A=0.1871	6.63	11.75	11.8	15.5	11.66	6.9
Location A=0.0134	25.8	20.2	16.1	18.4	16.9	8.0

Table 7 Latencies in terms of number of message lengths [3]

3: Source for location results : [NTCS]

6.5 Impact of variable maximal backoff, message size, traffic and mobility

Recall that the backoff counter *BC* is a random number in the interval [1.. *Tmax*]. The main difference between the neighbour elimination schemes independently proposed by us and by [PL] is that *Tmax* in [PL] depends on the node density relative to its neighbours, while we proposed to use a fixed value for *Tmax* =31. We compared these two *Tmax* formulas using both the *RANA* algorithm and the previous version without negative acknowledgements. The difference in performance was negligible. The difference in the *RE* values for *Tmax* =31 and *Tmax* from [PL] was in the range –0.4 to +0.3, the *SRB* data did not change, while the change in latency was upto –0.001 seconds.

We then experimented with the idea of a variable *Tmax* formula such that it would reflect the dynamic nature of the node degree after eliminating some neighbours. If a node *A* decides to retransmit the message, it calculates a random number x in the range [0,1). Then the backoff counter is calculated as *BC=x*Tmax*, where *Tmax* depends on the number of non-eliminated neighbours. We have tested the following two formulas for $Tmax = \frac{127}{NNEN}$ and $Tmax = 20 + \frac{31}{NNEN}$, where *NNEN* is the number of internal and non-internal neighbours of *A* who have not been eliminated yet. Note that x is fixed while NNEN (and consequently *Tmax* and *BC*) are recalculated each time node A receives a new copy of the same message. As a result, *BC* may increase, resulting in longer delay due to the reduced number of non-eliminated neighbours. The change in *RE* for both $Tmax = \frac{127}{NNEN}$ and $Tmax = 20 + \frac{31}{NNEN}$ was in the range –3.8 and +0.5, with no differences in *SRB* and latencies. Therefore, we were not able to find any improvements with variable *Tmax* formulas.

The impact of message size *p* was then experimentally verified by repeating the experiments for other sizes. In particular, measurements were done for p=64 and p=254 (that is, half the size and double the size). The impact of varying message size on *RE* and *SRB* results was negligible (except for values of p under *Tmax* which reduces the probability of a collision). Latency was naturally affected by the message size.

All the experiments described so far assume that there is only one message being broadcast at a given time. This assumption is reasonable for many applications, such as alarm signal. Moreover, if the transmission speed is high, in networks with small and medium number of nodes, one can expect that the current broadcasting task will finish before the next one is initiated. In order to verify the impact of several concurrent broadcasting tasks in the network, we assumed that each node may initiate a new broadcasting task at the beginning of any time slot with a given small probability. The experimental results confirm the small differences for *RE* and *SRB* data. More precisely, the mutual differences are increasing in favor of better methods. The reason is obvious: excess transmissions generated by weaker methods create even more collisions when more broadcasting tasks are in the network. That is, more unnecessary retransmissions cause more collisions.

The impact of node mobility is then verified. Each node moves with a given probability, to a randomly chosen destination position on a straight line with a speed that is also chosen at random between two threshold values. We have implemented only location updates between neighbouring nodes, using a method described in [S]. Each node stores the location of its neighbours, which suffices to apply almost all the described methods. In addition, it stores the location of its two-hop neighbours, using information obtained from its neighbours (this information is also needed to apply the multi-point relay method). In addition to location, each node decides to send a message containing its new location to all its neighbours whenever it detects that an existing edge will be broken, or that a two-hop neighbour is becoming a direct neighbour. The distance from a node A to a node B is measured by using an estimated position B' for node B, calculated by using the last known position of B and its reported speed and direction of movement. This kind of location updates significantly reduced the number of such messages. For instance, two nodes moving with the same speed and in the same direction do not need to repeatedly report to each other their new positions.

Because of locality of broadcasting procedures, and the efficiency of the described location update technique, the measured differences in *RE* and *SRB* data from the case of static nodes were, in all cases, under 1%. The difference is mostly due to adding location update messages in the system, and partially by occasional incorrect information about neighbour positions. Thus, the proposed broadcasting algorithms perform equally well in case of moving nodes. Note that a similar conclusion is reported also by Peng and Lu [PL] for the neighbour elimination scheme. In their experiments, there was also about a 1% difference between static scenario and the case where all nodes were in a state of continuous movement.

Chapter 7 Conclusion and Future Work

This work examined the problem of providing reliable broadcasts in wireless networks in both the one-to-all and one-to-one models. We proposed several algorithms that almost guarantee delivery of the message in one-to-all networks as well as reduce the redundancy in broadcasting that in turn reduces contention and collision in the network. The experimental results have demonstrated the efficiency of the proposed broadcasting algorithms. The internal and *RNG* (for one-to-one) concepts used do not introduce any communication overhead in the network, since the maintenance that they require is provided by regular message exchange between neighbours.

For the one-to-one model, the performance of the proposed algorithms is close to optimal. For future work, the multipoint relay scheme proposed for the one-to-all case in [QVL] may be tested in one-to-one networks as well. Also, some of the methods suggested in [NTCS] may be adapted.

For the one-to-all model, we believe that further savings and perhaps guaranteed delivery may be achieved by improving the proposed algorithms in various ways. Schemes that will use the location of the source may be applied, if the source location is part of the broadcast message. Generally, each receiving node will forward the message only to neighbours that are further than itself from the source. However, if a broadcast message does not arrive within a reasonable time from a neighbour closer to the source, that neighbour will receive the message as well. Next, neighbours that are further will receive no message if there is evidence that another neighbour will forward the same message to such neighbours.

References

[BKOO] M. de Berg, M. van Kreveld, R. van Oostrum, and M. Overmars, "*Simple traversal of a subdivision without extra storage*", Int. J. of Geographic Information Systems, 11, 1997, 359-373.

[BGI] R. Bar-Yehuda, O. Goldreich, A. Itai, "*On the time complexity of broadcast in radio networks: an exponential gap between determinism and randomization*", Proc. ACM Symp. Principles of Distributed Computing, 1987, 98-108.

[BMJHJ] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, J. Jetcheva, "*A performance comparison of multi-hop wireless ad hoc network routing protocols*", Proc. MOBICOM, 1998, 85-97.

[BMSU] P. Bose, P. Morin, I. Stojmenovic and J. Urrutia, "*Routing with guaranteed delivery in ad hoc wireless networks*", 3rd int. Workshop on Discrete Algorithms and methods for mobile computing and communications, Seattle, August 20, 1999, 48-55.

[CGGP] B.S. Chlebus, L. Gasienec, A. Gibbons, A. Pelc, "*Deterministic broadcasting in unknown radio networks*", Proc. SODA, 2000, to appear.

[DSW] S. Datta, I. Stojmenovic, J. Wu, "*Location and internal nodes based routing in wireless networks*", SITE, University of Ottawa, TR-99-08, September 1999

[EGHK] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, "*Next century challenges: Scalable coordination in sensor networks*, Proc. MOBICOM, 1999, Seattle, 263-270.

[EWB] A. Ephremides, J.E. Wieselthier and D.J. Baker, "*A design concept for reliable mobile radio networks with frequency hoping signaling*", Proc. IEEE 75, 1987, 56-73.

[GKP] M. Gerla, T.J.Kwon, and G. Pei, "*On demand routing in large ad-hoc wireless networks with passive clustering*", Proc. IEEE WCNC, September 2000

[GT] M. Gerla and J.T.C. Tsai, "*Multicluster, mobile, multimedia radio network, Wireless networks*", 1, 1995, 255-265.

[HKB] Heinzelman, J. Kulik, H. Balakrishnan, "*Adaptive protocols for information dissemination in wireless sensor networks*", Proc. MOBICOM, Seattle, 1999, 174-185.

[HOTV] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath, "*Flooding for reliable multicast in multihop ad hoc networks*", 3rd Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications DIAL'M, August 1999.

[L] G. Lauer, "*Address servers in hierarchical networks*", Proc. ICC, 1988, 443-451.

[LG] C.R. Lin and M. Gerla, "*Adaptive clustering for mobile wireless networks*", IEEE J.

Selected Areas in Communications, 15, 7, 1997, 1265-1275.

[NTCS] S.Y. Ni, Y.C. Tseng, Y.S. Chen, J.P. Sheu, "*The broadcast storm problem in a mobile ad hoc network*", Proc. MOBICOM, Seattle, Aug. 1999, 151-162.

[P] A.K. Parekh, "*Selecting routers in ad hoc wireless networks*", ITS, 1994.

[PL] Wei Peng, Xi-Cheng Lu, "*On the reduction of broadcast redundancy in mobile ad hoc networks*", Proc. First Annual Workshop on Mobile and Ad Hoc Networking and Computing, Boston, USA, August 11, 2000, 129-130.

[PR] E. Pagani and G.P. Rossi, "*Providing reliable and fault tolerant broadcast delivery in mobile ad-hoc networks*", Mobile Networks and Applications, 4, 1999, 175-192.

[SFLYO] M.T. Sun, W.C. Feng, T.H. Lai, K. Yamada, H. Okada, "*GPS based message broadcast for adaptive inter-vehicle communications*", Proc. Int. Conf. on Parallel Processing, Toronto, August 2000.

[SK] L. Subramanian and R. H. Katz, "*An architecture for building self-configurable systems*", Proc. First Annual Workshop on Mobile and Ad Hoc Networking and Computing, Boston, USA, August 11, 2000, 63-73.

[SL1] I. Stojmenovic and X. Lin, "*GEDIR: Loop-free location based routing in wireless networks*", IASTED Int. Conf. on Parallel and Distributed Computing and Systems, Nov. 3-6, 1999, Boston, MA, USA, 1025-1028.

[SL2] I. Stojmenovic and Xu Lin, "*Power-aware localized routing in wireless networks*", IEEE Int. Parallel and Distributed Processing Symp., Cancun, Mexico, May 1-5, 2000, to appear.

[SS] I. Stojmenovic, M. Seddigh, "*Broadcasting algorithms in wireless networks*", Proc. Int. Conf. on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet SSGRR, L'Aquila, Italy, July 31-Aug. 6, 2000.

[SSZ] I. Stojmenovic, M. Seddigh, J. Zunic, "*Internal node based broadcasting in wireless networks*", Proc. IEEE Int. Conf. on System Sciences, January 2001, to appear.

[T] G. Toussaint, "*The relative neighbourhood graph of a finite planar set, Pattern Recognition*", 12, 4, 1980, 261-268.

[WL] J. Wu and H. Li, "*On calculating connected dominating set for efficient routing in ad hoc wireless networks*", Proc. DIAL M, Seattle, Aug. 1999, 7-14.

[WNE] J.E. Wieselthier, G.D. Nguyen and A. Ephremides, "*On the construction of energy-efficient broadcast and multicast trees in wireless networks*", IEEE INFOCOM, March 2000.